Міністерство освіти і науки України
Національний технічний університет України
Київський політехнічний інститут імені Ігоря Сікорського

Методи аналізу даних та машинного навчання в інформаційно-управляючих системах

Курс лекцій
Частина 2

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за спеціальністю 126 – «Інформаційні системи та технології»

Укладачі: В.В. Онищенко, О.В. Гавриленко, Мягкий М.Ю.

Електронне мережеве навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2025

УДК 004.896 (075)
А64

Укладачі: *Онищенко Вікторія Валеріївна*, доктор техн. наук, проф., проф. кафедри інформаційних систем та технологій, факультет інформатики та обчислювальної техніки,
Гавриленко Олена Валеріївна, канд. фіз.-мат. наук, доц., доц. кафедри інформаційних систем та технологій, факультет інформатики та обчислювальної техніки,
Мягкий Михайло Юрійович, асистент кафедри інформаційних систем та технологій, факультет інформатики та обчислювальної техніки.

Рецензент *Олійник Юрій Олександрович*, канд. техн. наук, доц. кафедри інформатики та програмної інженерії, факультет інформатики та обчислювальної техніки, КПІ ім. Ігоря Сікорського

Відповідальний редактор *Жураковська Оксана Сергіївна*, канд. техн. наук, доц., доц. кафедри інформаційних систем та технологій, факультет інформатики та обчислювальної техніки, КПІ ім. Ігоря Сікорського

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(*протокол* № 8 *від* 07.06.2025 *р.*)
за поданням вченої ради факультету інформатики та обчислювальної техніки
(*протокол* № 13 *від* 26.05.2025 *р.*)

А64 **Методи аналізу даних та машинного навчання в інформаційно-управляючих системах** [Електронний ресурс]: Курс лекцій. Частина 2: навч. посіб. для здобувачів ступеня бакалавра за спец. 126 – «Інформаційні системи та технології» / КПІ ім. Ігоря Сікорського; уклад.: В.В. Онищенко, О.В. Гавриленко, М.Ю. Мягкий – Електрон. текст. дані (1 файл). – Київ: КПІ ім. Ігоря Сікорського, 2025. – 318 с.

У посібнику систематизовано викладено курс лекцій з дисциплін: «Аналіз даних в інформаційно-управляючих системах», «Обробка та аналіз текстових даних на Python» «Технології Машинного навчання» та «Прикладні задачі Машинного навчання». Курс лекцій покликаний розвинути теоретичні навички студентів та навички до розв’язання основних задач з вказаних дисциплін для їх застосування в інженерній практиці. Навчальний посібник призначений для здобувачів ступеня бакалавра зі спеціальності 126 – «Інформаційні системи та технології», буде також корисним для студентів інших технічних спеціальностей.

УДК 004.896 (075)
Реєстр. № НП 23/24-165. Обсяг 13,25 авт. арк.
Національний технічний університет України
Київський політехнічний інститут імені Ігоря Сікорського
проспект Берестейський, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

ЗМІСТ

ВСТУП

Сучасні інформаційно-управляючі системи (ІУС) відіграють ключову роль у процесах прийняття рішень, автоматизації бізнес-процесів, аналізу великих обсягів даних та прогнозування подій. Розвиток технологій штучного інтелекту (ШІ) та Машинного навчання (ML) значно розширив можливості таких систем, дозволяючи підвищити їхню ефективність, гнучкість і точність.

Методи аналізу даних у поєднанні з алгоритмами Машинного навчання дають змогу обробляти складні масиви інформації, знаходити приховані закономірності та адаптивно реагувати на зміни у середовищі. Використання таких підходів особливо важливе в сферах управління підприємствами, фінансової аналітики, медицини, кібербезпеки та інших галузях, де оперативність і точність рішень є критичними.

У цьому дослідженні розглянуто основні методи Аналізу даних, зокрема кластеризацію, регресійний аналіз, нейронні мережі та глибоке навчання, а також їхнє застосування в ІУС. Особливу увагу приділено питанням оптимізації алгоритмів, їхньої інтеграції у сучасні програмні рішення та перспективам розвитку цієї галузі.

Даний посібник призначений для здобувачів ступеня бакалавра зі спеціальності 126 – «Інформаційні системи та технології», в рамках дисциплін «Аналіз даних в інформаційно-управляючих системах», «Технології машинного навчання» та «Прикладні задачі машинного навчання».

Мета курсу лекцій: набуття теоретичних знань і практичних навичок розв’язування задач Інтелектуального аналізу даних та Машинного навчання у різних сферах професійної діяльності.

Предметом курсу лекцій є технології, методи та засоби Інтелектуального аналізу даних та Машинного навчання.

Задачі вивчення дисципліни: навчити студентів застосовувати апарат Інтелектуального аналізу даних та Машинного при створенні інформаційно-управляючих систем.

ЛЕКЦІЯ 11

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТЕКСТУ

11.1. Поняття, етапи та методи Text Mining

Інтелектуальний аналіз тексту (*Text Mining*, *Text Analytics*) – це набір методів і технологій для автоматизованої обробки текстової інформації з призначенням патернів, знань та схованих закономірностей.

Ключові завдання Text Mining

1. Категоризація текстів;
2. Пошук інформації;
3. Обробка змін в колекціях текстів;
4. Розробка засобів представлення інформації для користувача.

Основні етапи аналізу Text Mining

Етап 1. Збір даних;
Етап 2. Попередня обробка;
Етап 3. Формування ознак;
Етап 4. Аналіз та моделювання;
Етап 5. Візуалізація та інтерпретація результатів.

Зупинимося докладніше на кожному з цих етапів.

11.2. Збір даних

Збір текстових даних (*Text Data Collection*) – це перший і важливий етап в інтелектуальному аналізі тексту. Джерела текстових даних можуть бути додатково: веб-сайти, соцмережі, документи, книги тощо.

На даному етапі відбувається отримання текстових даних з різних джерел (статті, соцмережі, документи).

11.2.1. Основні методи збору текстових даних

До основних методів збору даних належать:

1. Веб скрейпінг;
2. Використання API;
3. Збір текстів із документів;
4. Використання корпусів текстів.

Зупинимось докладніше на кожному з них.

11.2.1.1. Веб-скрейпінг

Веб-скрейпінг (*Web Scraping*) – процес зберігання текстових даних із веб-сторінок за допомогою бібліотек для автоматичного зчитування $HTML$-коду та вилучення тексту.

Приклади:
- Статті з новинних сайтів;
- Коментарі;
- Форуми.

Обмеження:

- Деякі сайти блокують автоматичний збір даних;
- Необхідно дотримуватися $robots.txt$ та юридичних правил.

Приклад реалізації веб-скрепінгу Python

```
import requests
from bs4 import BeautifulSoup

url = "https://example.com"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

# Витягуємо текст із тегу <p>
text = " ".join([p.text for p in soup.find_all("p")])
print(text)
```

11.2.1.2. Використання *API*

Використання API – це процес створення запит до офіційних сервісів для отримання текстових даних. Використовуються *REST API* або *GraphQL API* для отримання структурованих даних.

Приклади:
- *YouTube API* (коментарі, описи відео);
- *Twitter API* (твіти);
- *Reddit API* (дискусії).

Плюси:
- Легальний спосіб збору даних;
- Часто вже оброблені структури $(JSON, XML)$.

Мінуси:
- Обмежити кількість запитів;
- Потрібна реєстрація та *API*-ключі.

Приклад використання *API* в Python

```
import tweepy

api_key = "your_api_key"
api_secret = "your_api_secret"
access_token = "your_access_token"
access_secret = "your_access_secret"

auth = tweepy.OAuth1UserHandler(api_key, api_secret,
access_token, access_secret)
api = tweepy.API(auth)
```

```
tweets = api.search_tweets(q="OpenAI", lang="en",
count=10)
for tweet in tweets:
    print(tweet.text)
```

11.2.1.3. Збір текстів із документів

Збір текстів із документів ($PDF, DOCX, TXT, JSON$) – це процес зчитування тексту з файлів. Використовуються бібліотеки для роботи з документами.

Приклади:

- Книги;
- Статті;
- Звіти.

Плюси:

- Повний контроль над джерелами;
- Підходити для обробки наукових текстів.

Мінуси:

- Деякі PDF мають захищений текст (потрібне OCR).

Приклад збору тексту в Python

```
import fitz  # PyMuPDF

with fitz.open("document.pdf") as doc:
    text = "\n".join([page.get_text() for page in
doc])
print(text)
```

11.2.1.4. Використання корпусів текстів

Використання корпусів текстів (*готових наборів даних*) – це процес використання готових баз текстів за допомогою завантаження з відкритих ресурсів.

Приклади:

- $NLTK\ Corpora$ (різні корпуси текстів);
- $Wikipedia\ Dumps$ (статті Вікіпедії);
- $Hugging\ Face\ Datasets$ (готові корпуси для NLP).

Плюси:

- Безкоштовні й доступні;
- Висока якість текстів.

Мінуси:

- Обмежена тематика.

Приклад використання корпусу *NLTK* в Python

```
import nltk
```

```
from nltk.corpus import reuters

nltk.download("reuters")
text = reuters.raw("test/14826")
print(text)
```

11.2.2. Порівняння методів збору текстових даних

1. Веб-скрейпінг використовується для збору даних із сайтів (парсінгу);
2. *API* використовуються для легального отримання структурованих даних;
3. Документи $(PDF, DOCX)$ використовуються для для обробки статей, звітів;
4. Готові корпуси забезпечують швидкий доступ перевірених текстів.

11.3. Попередня обробка

Попередня обробка текстових даних (*Text Preprocessing*) – це ключовий етап в інтелектуальному аналізі тексту (Text Mining), який допоможе зробити текст придатним для аналізу.

На даному етапі відбувається очищення тексту від зайвих символів, токенізація, лематизація/стемінг.

Основні етапи попередньої обробки тексту

До основних етапів попередньої обробки текстів належать:

Етап 1. Очищення тексту;
Етап 2. Токенізація;
Етап 3. Видалення стоп-слів;
Етап 4. Лематизація та стемінг.

Зупинимося докладніше на кожному з них.

11.3.1. Очищення тексту

Очищення тексту – видалення непотрібних символів. В процесі очистки видаляються *HTML*-теги, спецсимволи, зайві проби, цифри.

Приклад реалізації процедури очищення тексту в Python

```
import re

def clean_text(text):
    text = re.sub(r"<.*?>", "", text)  # Видалення HTML
    text = re.sub(r"\d+", "", text)  # Видалення чисел
    text = re.sub(r"[^\w\s]", "", text)  # Видалення пунктуації
    text = text.lower().strip()  # Перетворення в нижній регістр
    return text
```

```
text = "Привіт! Це тестовий текст з HTML
<b>тегами</b> і цифрами 123."
print(clean_text(text))  # → "привіт це тестовий
текст з html тегами і цифрами"
```

11.3.2. Токенізація

Токенізація – процес розбиття тексту на окремі слова або речення. Ці елементарні одиниці називаються *токени*.

Під *нормалізацією* мається на увазі обробка отриманих токенів і видалення їх із результуючого масиву, якщо вони не є лемами, що несуть семантичний зміст.

Приклад результатів токенізації наведено на рис. 11.1.

liveart, груден, 2014, листопад, 2015, посад, front-end, розробник, опис, проект, веб-дизайнер, створен, макет, різн, товар, замовлен, друк, технології, typescript, knockout.js, php5, javascript, bootstrap3, javascript, jquery, обов'язк, постійн, підтримк, рефакторинг, розробк, клієнт.

Рис. 11.1. Приклад токенів

Приклад реалізації процедури токенізації в Python

```
import nltk
from nltk.tokenize import word_tokenize,
sent_tokenize

nltk.download("punkt")
text = "Привіт! Це текст для тестування токенізації.
Як справи?"

words = word_tokenize(text)  # Токенізація слів
sentences = sent_tokenize(text)  # Токенізація речень

print(words)  # ['Привіт', '!', 'Це', 'текст', 'для',
'тестування', 'токенізації', '.']
print(sentences)  # ['Привіт!', 'Це текст для
тестування токенізації.', 'Як справи?']
```

11.3.3. Видалення стоп-слів

Видалення стоп-слів – усунення частин, але малозначущих слів («і», «це», «що»). Можна використовувати власний список українських стоп-слів.

Приклад реалізації процедури видалення стоп-слів в Python

```
import nltk
```

```
from nltk.corpus import stopwords

nltk.download("stopwords")
ukrainian_stopwords =
set(stopwords.words("ukrainian"))  # Українських
вбудованих немає, можна додати власні

text = "Це є приклад речення, яке містить зайві
слова."
words = word_tokenize(text)
filtered_words = [word for word in words if
word.lower() not in ukrainian_stopwords]

print(filtered_words)  # ['Це', 'приклад', 'речення',
',', 'містить', 'зайві', 'слова', '.']
```

11.3.4. Лематизація або стемінг

Лематизація або *стемінг* – це два основних методи зведення слів до їхньої базової форми, які зменшують варіативність слів і покращують їх.

1. *Стемінг* – обрізає слово до його кореня, не враховуючи граматичні правила. Це швидкий, але грубий метод, після чого можна створити неіснуючі слова.

Багато пошукових систем використовують стемінг для встановлення синонімічних зв'язків, якщо в них збігаються форми після стематизації. Цей процес називають *злиттям*.

Для стемінгу використовується алгоритм стемінгу Портера.

Стемінг Портера (*Porter Stemming Algorithm*) – один із найпопулярніших алгоритмів для приведення слів до їхньої основи (стему). Він був розроблений Мартіном Портером у 1980 році і широко використовується для обробки тексту, зокрема у пошукових системах та *NLP*.

Основна ідея алгоритму стемінгу Портера:

- Видалення суфіксів та префіксів для скорочення слів до кореневої форми;
- Використання набору правил для поступового перетворення слова;
- Поділ на п'ять етапів (поетапне скорочення);
- Перевірка умов перед кожним скороченням, щоб уникнути спотворення значення слова.

Приклади:

- «бігав», «бігти», «бігатиму» → «біг»;
- «граєш», «грати», «грався» → «грат» (може бути некоректним).

Плюси:

- Швидкий;

- Не потребує великих мовних ресурсів;
- Виправляє помилки, пов’язані з морфологією..

Мінуси:

- Може давати неправильні корені (спотворювати слова);
- Не враховує контекст;
- Не підходить для всіх мов.

Основні етапи стемінгу Портера

Етап 1. Видалення суфіксів множини та часу.
Етап 2. Обробка суфіксів дієслівних форм.
Етап 3. Обробка суфіксів прикметників та іменників.
Етап 4. Видалення залишкових суфіксів.
Етап 5. Завершальні перетворення:

- Виявлення подвійних літер;
- Перевірка голосних, щоб слово залишилося зрозумілим.

Приклад реалізації стемінгу в Python

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()
words = ["running", "flies", "easily", "denied"]
stems = [stemmer.stem(word) for word in words]

print(stems)  # ['run', 'fli', 'easili', 'deni']
```

2 *Лематизація* – приводити слово до його початкової форми. Вона використовує словники морфологічного аналізу та враховує граматичні правила, щоб повернути слово до його базової форми.

Приклад:

- «бігав», «бігти», «біжу» → «бігти»;
- «граєш», «грати», «грався» → «грати».

Плюси:

- Точніша, ніж стемінг;
- Враховує граматику;
- Підходить для мовних моделей.

Мінуси:

- Повільніша, ніж стемінг;
- Потребує мовного словника.

Приклад реалізації лематизації в Python

```
import spacy

nlp = spacy.load("uk_core_news_sm")  # Завантаження моделі для української мови
```

```
text = "Я бігав у парку і бачив, як діти гралися."
doc = nlp(text)
lemmatized_text = " ".join([token.lemma_ for token in
doc])
print(lemmatized_text)  # "я бігти у парк і бачити ,
як дитина гратися ."
```

Застосування.

Рекомендації щодо застосування наведено в табл. 11.1.

Таблиця 11.1. Особливості застосування лематизації та стемінгу

Метод	Використання
Стемінг	Коли потрібно швидке, але приблизне нормалізування слів (пошукові системи, базові NLP-завдання).
Лематизація	Коли потрібна точна обробка тексту, що враховує граматику (машинний переклад, аналіз тональності, класифікація текстів).

11.4. Формування ознак

Формування ознак – перетворення тексту в числові представлення.

Перетворення тексту в числові представлення – процес конвертування текстових даних у числовий формат. Даний процес здійснюється, оскільки моделі машинного навчання працюють лише з числовими даними.

Основні методи перетворення тексту в числові представлення

1. Bag of Words;
2. TF-IDF;
3. Word Embeddings.

Докладніше зупинимося на кожному з них.

11.4.1. Bag of Words

Bag of Words (*BoW*, *мішок слів*) – один із найпростіших методів перетворення тексту в числові представлення. Його широко використовують у обробці природної мови (NLP) для аналізу текстів, класифікації документів, виявлення тем і побудови моделей машинного навчання.

Основна ідея:

- Ігнорується порядок слів – текст розглядається як набір (мішок) слів;
- Рахується частота слів у документі – створюється матриця, де кожен рядок відповідає документу, а кожен стовпець – окремому слову;
- Слова перетворюються в числові вектори, що можна використовувати для аналізу.

Приклад. Нехай на вхід подаються два документи:

$Doc1$: «Кіт спить на сонці. »

$Doc2$: «Собака грає на вулиці. »

Словник (унікальні слова в корпусі) для цих документів представляється у вигляді (11.1):

[«кіт», «спить», «на», «сонці», «собака», «грає», «вулиці»]. (11.1)

Матриця BoW для словника (11.1) представлена в табл. 11.2:

Таблиця 11.2. Матриця BoW

	кіт	спить	на	сонці	собака	грає	вулиці
$Doc1$	1	1	1	1	0	0	0
$Doc2$	0	0	1	0	1	1	1

Слово «на» зустрічається в обох документах, тому має значення 1 у кожному.

Приклад реалізації BoW у Python

Використання $CountVectorizer$ $(sklearn)$

```
from sklearn.feature_extraction.text import
CountVectorizer

# Вхідні тексти
documents = ["Кіт спить на сонці.", "Собака грає на
вулиці."]

# Ініціалізація BoW
vectorizer = CountVectorizer()

# Перетворення текстів у векторну форму
X = vectorizer.fit_transform(documents)

# Вивід результатів
print("Словник:", vectorizer.get_feature_names_out())
print("Матриця частот:\n", X.toarray())
```

Вивід

```
Словник: ['грає' 'кіт' 'на' 'сонці' 'собака' 'спить'
'вулиці']
Матриця частот:
 [[0 1 1 1 0 1 0]
  [1 0 1 0 1 0 1]]
```

Недоліки BoW:

- Ігнорує порядок слів. Наприклад, «не їжте це» та «їжте це не» будуть однаковими;

- Не враховує семантику (значення слів). Наприклад, «кіт» та «кошеня» вважаються різними словами;
- Велика розмірність (Sparse Matrix Problem).

Якщо словник великий (наприклад, 100000 слів), то матриця буде величезною.

11.4.2. Метод TF-IDF

Метод TF-IDF (*Term Frequency – Inverse Document Frequency*) – це статистичний метод, який використовується для оцінки важливості слів у документі відносно всього корпусу текстів. Він часто застосовується в пошукових системах, класифікації текстів, видобутку інформації та обробці природної мови (NLP).

Основна ідея TF-IDF:

- Метод TF-IDF визначає, наскільки значущим є слово в документі, враховуючи частоту його появи і розповсюдженість у всьому корпусі.

Основні кроки методу TF-IDF

Крок 1. Обчислення метрики TF.

TF (*Term Frequency*) – частота слова у документі, що визначає, як часто слово зустрічається у конкретному документі. Обчислюється за формулою (11.2):

$$TF = \frac{\text{Кількість появ слова у документі}}{\text{Загальнакількість слів у документі}}. \quad (11.2)$$

Наприклад, якщо слово «кіт» з'являється 3 рази в тексті з 100 слів:

$$TF(\text{«кіт»}) = \frac{3}{100} = 0{,}03.$$

Крок 2. Обчислення метрики IDF.

IDF (*Inverse Document Frequency*) – зворотна частота документа. IDF визначає, наскільки рідкісним є слово в усьому корпусі. Обчислюється за формулою (11.3):

$$IDF = log\left(\frac{\text{Загальна кількість документів}}{\text{Кількість документів,де є слово}}\right), \quad (11.3)$$

де log – зазвичай натуральний логарифм (ln). Іноді використовують десятковий логарифм (lg) (рідше застосовується) або двійковий логарифм (використовується в деяких інформаційно-пошукових системах).

З формули (11.3) очевидно, що:

- Якщо слово зустрічається у багатьох документах, його значення зменшується;
- Якщо слово рідкісне, його значення збільшується.

Крок 3. Обчислення метрики $TF - IDF$

$TF - IDF$ – визначає значущість слова у документі. Обчислюється за формулою (11.4):

$$TF - IDF = TF \cdot IDF. \quad (11.4)$$

Чим вище $TF - IDF$, тим важливіше слово. Загальні слова («це», «і», «у», «що») отримають низькі значення, бо вони часто зустрічаються в багатьох текстах.

Приклад розрахунку $TF - IDF$

Нехай на вхід подаються три документи:

$Doc1$: «Кіт спить на сонці»;

$Doc2$: «Собака грає на вулиці»;

$Doc3$: «Кіт і собака сплять».

Виконаємо послідовно всі кроки методу TF-IDF.

Крок 1. TF розраховується за формулою (11.2). представлено в табл. 11.3.

Таблиця 11.3. Розрахунок TF

Слово	TF у $Doc1$	TF у $Doc2$	TF у $Doc3$
кіт	0,25	0	0,25
спить	0,25	0	0
на	0,25	0,25	0
сонці	0,25	0	0
собака	0	0,25	0,25
грає	0	0,25	0
вулиці	0	0,25	0
сплять	0	0	0,25

Крок 2. IDF розраховується за формулою (11.3). Результати представлено в табл. 11.4.

Таблиця 11.4. Розрахунок IDF

Слово	Кількість документів, де зустрічається слово	IDF
кіт	2	$log(3/2) = 0{,}176$
спить	1	$log(3/1) = 0{,}477$
на	2	$log(3/2) = 0{,}176$
сонці	1	$log(3/1) = 0{,}477$
собака	2	$log(3/2) = 0{,}176$
грає	1	$log(3/1) = 0{,}477$
вулиці	1	$log(3/1) = 0{,}477$
сплять	1	$log(3/1) = 0{,}477$

Крок 3. $TF - IDF$ розраховується за формулою (11.4). Результати представлено в табл. 11.5.

Таблиця 11.5. Розрахунок $TF - IDF$

Слово	$TF - IDF$ у $Doc1$	$TF - IDF$ у $Doc2$	$TF - IDF$ у $Doc3$
кіт	0,044	0	0,044
спить	0,119	0	0

на	0,044	0,044	0
сонці	0,119	0	0
собака	0	0,044	0,044
грає	0	0,119	0
вулиці	0	0,119	0
сплять	0	0	0,119

З табл. 11.5 можна зробити наступні висновки:

- Слово «спить» має вище значення $TF - IDF$, бо воно унікальне для $Doc1$;
- Слова «на» та «кіт» мають нижчі значення, бо вони є у кількох документах.

Приклад реалізації методу TF-IDF у Python

Використання бібліотеки sklearn

```
from sklearn.feature_extraction.text import
TfidfVectorizer

# Вхідні тексти
documents = ["Кіт спить на сонці", "Собака грає на
вулиці", "Кіт і собака сплять"]

# Ініціалізація TF-IDF
vectorizer = TfidfVectorizer()

# Перетворення текстів у TF-IDF матрицю
X = vectorizer.fit_transform(documents)

# Вивід словника та значень TF-IDF
print("Слова:", vectorizer.get_feature_names_out())
print("TF-IDF:\n", X.toarray())
```

Вивід

```
Слова: ['грає' 'кіт' 'на' 'собака' 'сонці' 'спить'
'сплять' 'вулиці']
TF-IDF:
 [[0.    0.5   0.5   0.    0.5   0.5   0.    0.   ]
  [0.5   0.    0.5   0.5   0.    0.    0.    0.5  ]
  [0.    0.5   0.    0.5   0.    0.    0.5   0.   ]]
```

11.4.3. Word Embeddings

Word Embeddings – це метод перетворення слів у числові вектори, що зберігають їхній семантичний зміст. На відміну від Bag of Words (BoW) та

TF-IDF, які не враховують значення слів, Word Embeddings дозволяють моделі розуміти зв'язки між словами.

Основна ідея:

- Кожне слово кодується як багатовимірний вектор;
- Слова з подібним значенням мають схожі вектори (наприклад, «король» та «королева» матимуть подібні координати);
- Вектори можна використовувати для аналізу тексту, класифікації, перекладу тощо.

Приклад: Модель може навчитися такій залежності між словами:

«король» − «чоловік» + «жінка» ≈ «королева».

Це означає, що векторне представлення «королева» знаходиться близько до результату цього векторного віднімання/додавання.

11.4.3.1. Основні методи Word Embeddings

1. *Word2Vec* (*слово у вектор*) (Google, 2013) — це техніка, яка використовується для перетворення слів у вектори, таким чином фіксуючи їх значення, семантичну подібність і зв'язок із навколишнім текстом. Цей метод допомагає комп'ютерам вивчати контекст і значення виразів і ключових слів із великих текстових колекцій, таких як новинні статті та книги.

Основна ідея Word2Vec:

- Кожне слово представляється як багатовимірний вектор, де положення вектора в цьому багатовимірному просторі відображає значення слова .
- Для навчання векторів слів в методі використовується нейронна мережа.
- Метод підходить для великих корпусів текстів.
- Метод має дві архітектури:

CBOW (*Continuous Bag of Words*) – архітектура, що прогнозує слово за контекстом;

Skip-gram – архітектура, що прогнозує контекст за словом.

Навчання Word2Vec на прикладі простого корпусу у Python

```
from gensim.models import Word2Vec

# Простий корпус текстів
sentences = [
    ["кіт", "спить", "на", "вулиці"],
    ["собака", "грає", "у", "парку"],
    ["кіт", "і", "собака", "друзі"]
]

# Створення та навчання моделі
```

```
model = Word2Vec(sentences, vector_size=10, window=2,
min_count=1, workers=4)

# Отримання вектора для слова "кіт"
print("Вектор для слова 'кіт':", model.wv["кіт"])

# Знаходження найближчих слів до "кіт"
print("Слова, найближчі до 'кіт':",
model.wv.most_similar("кіт"))
```

2. *GloVe* (*Global Vectors for Word Representation*) (Stanford, 2014) – це метод представлення слів у векторному просторі (Word Embeddings). На відміну від Word2Vec, який навчається на основі контекстних вікон, GloVe використовує статистику спільної зустрічальності слів.

Принцип роботи GloVe

1. Будується матриця спільної зустрічності слів:

- Записується, як часто слово w_i зустрічається поряд із словом w_j;
- Наприклад, слово «король» частіше зустрічається поруч із «принц» або «королева» , ніж із «автомобіль».

2. GloVe використовує цю матрицю для побудови векторних зображень слів.

Оптимізується функція (11.5):

$$J = \sum_{i,j}\left(f(X_{ij}) \cdot \left(w_i^{\mathrm{T}} \cdot w_j + b_i + b_j - log(X_{ij})\right)^2\right), \qquad (11.5)$$

де:

X_{ij} – частота спільного використання слів;

w_i, w_j – вектори слів;

b_i, b_j – зміщення.

Приклад реалізації GloVe в Python

Зразок коду

```
import gensim.downloader as api

# Завантаження готової моделі GloVe
glove_model = api.load("glove-wiki-gigaword-50")  #
50-вимірний вектор

# Приклад роботи
word = "king"
print(glove_model[word])

# Аналіз подібності слів
print(glove_model.most_similar("king"))
```

Очікуваний результат

```
[ 0.5043  0.3421 -0.4012 ...  0.2312  0.9874 -0.6723]
# Вектор слова "king"

[('queen', 0.834), ('prince', 0.812), ('monarch',
0.805)]  # Подібні слова
```

3. *FastText* (Facebook, 2016) – це метод вбудовування слів, який є покращеною версією Word2Vec. Основна відмінність – робота з підсловами (subword embeddings), що дає кращі результати для морфологічно багатих мов (наприклад, українська).

Принцип роботи FastText

1. Розбиває слово на підслова (n-грами):
- Наприклад, слово

«програмування» →

[« < пр», «про», «рог», «огр», «грам», … , »ня > »,

« < програмування > »].

- Це дозволяє отримати морфологічні особливості слів.

2. Вектор кожного слова – це середнє значення векторів його n-грам:
- Це дає кращу генералізацію , особливо для рідкісних слів.

3. Підтримується як CBOW, так і Skip-gram:
- CBOW швидкий, Skip-gram точніший для малих корпусів.

4. FastText добре працює з рідкісними словами та новими словоформами!

Принцип реалізації FastText на Python

```
import fasttext.util
fasttext.util.download_model('uk',
if_exists='ignore')  # Українська модель
ft = fasttext.load_model('cc.uk.300.bin')

# Вектор слова "програмування"
print(ft.get_word_vector("програмування"))

# Пошук схожих слів
print(ft.get_nearest_neighbors("комп'ютер"))
```

4. *BERT* (*Bidirectional Encoder Representations from Transformers*) (Google, 2018) – це модель обробки природної мови (NLP), яка базується на архітектурі Transformer . Основна особливість BERT – двонаправлена увага (Bidirectional Attention), що дозволяє краще розуміти контекст слова у реченні. Дана модель буде описана нижче.

11.4.3.2. Порівняння методів перетворення тексту в числове представлення

Результати порівняння наведено в табл. 11.6.

Таблиця 11.6. Порівняння методів перетворення тексту в числове представлення

Метод	Враховує порядок слів?	Враховує значення?	Підходить для великих корпусів?
BoW	Ні	Ні	Так
TF-IDF	Ні	Частково	Так
Word2Vec	Так	Так	Так
GloVe	Так	Так	Так
FastText	Так	Так	Так
BERT	Так	Так (контекстний)	Ні (вимогливий до ресурсів)

11.5. Аналіз та моделювання текстів

Аналіз та моделювання текстів (*Text Mining & Modeling*) — це процес обробки, аналізу та отримання значущої інформації з текстових даних. Він використовується в таких сферах, як машинне навчання, штучний інтелект, лінгвістика, маркетинг, соціальні науки тощо [39-41, 45, 47].

На даному етапі відбувається класифікація, кластеризація, аналіз тональності, виявлення тем.

11.5.1. Аналіз тональності текстів

Аналіз тональності (*Sentiment Analysis*) – визначення емоційного забарвлення тексту (позитивний, нейтральний, негативний).

Методи аналізу тональності текстів

1. Словникові методи;
2. Методи на основі машинного навчання;
3. Глибоке навчання та нейромережі.

Докладніше зупинимось на кожному з них.

11.5.1.1. Словникові методи

Словникові методи (*Lexicon-based methods*) – це методи визначення тональності тексту за допомогою спеціальних словників.

Принцип словникового методу

Крок 1. Вхідний текст розбити на речення. Нехай m – кількість речень в тексті.

Крок 2. Представити кожне речення у вигляді множини A_i $(i = \overline{1,m})$.

Крок 3. Як відомо, будь-який текст може викликати в людини три типи емоцій: позитивні, негативні та нейтральні. Для їх визначення у текстах обрати спеціальні словники:

- B – словник позитивних емоцій;

- C – словник негативних емоцій;
- D – словник нейтральних емоцій.

Крок 4. Розрахувати метрики (11.6):

$$K_{pi} = \frac{|A_i \cap B|}{|A_i|}, K_{ni} = \frac{|A_i \cap C|}{|A_i|}, K_{Ni} = \frac{|A_i \cap D|}{|A_i|}, \quad (11.6)$$

де:

K_{pi} – коефіцієнт позитивних емоцій від речення i;

K_{ni} – коефіцієнт негативних емоцій від речення i;

K_N – коефіцієнт нейтральних емоцій від речення i.

Крок 5. Для кожної множини A_i $(i = \overline{1,m})$ брати найбільше значення розрахованих метрик:

- Якщо K_{pi} – найбільше, то речення вважати позитивним;
- Якщо K_{ni} – найбільше, то речення вважати негативним;
- Якщо K_{Ni} – найбільше, то речення вважати нейтральним.

Крок 6. За отриманими оцінками кожного речення, зробити висновок про емоційне забарвлення всього тексту загалом (встановити яких речень, позитивних, негативних чи нейтральних, більше).

Переваги та недоліки.

- Використовують спеціальні словники з оцінками слів (позитивні, негативні, нейтральні);
- Простий, не потребує навчання;
- Погано працює з сарказмом, контекстом, багатозначністю слів.

Приклади словникових методів:

- SentiWordNet;
- VADER (Valence Aware Dictionary and sEntiment Reasoner);
- LIWC (Linguistic Inquiry and Word Count);
- NRC Emotion Lexicon.

Інтерпритація результатів:

- Позитивний ($score > 0$);
- Нейтральний ($score \approx 0$);
- Негативний ($score < 0$).

Приклад реалізації словникового методу в Python

```
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()
text = "Цей фільм просто чудовий! Я в захваті."

score = analyzer.polarity_scores(text)
print("Оцінка тональності:", score)
```

11.5.1.2. Методи на основі машинного навчання

Методи на основі машинного навчання (*ML-based methods*) – це методи визначення тональності тексту за допомогою алгоритмів машинного навчання.

Переваги та недоліки:

- Використовують моделі ML;
- Навчаються на розмічених даних;
- Потребують великих наборів даних для ефективної роботи.

Популярні ML-алгоритми:

- Naive Bayes;
- Logistic Regression;
- Support Vector Machines (SVM).

Про ці алгоритми говорилося в попередніх лекціях.

11.5.1.3. Методи на основі глибинного навчання

Глибоке навчання (*Deep Learning-based methods*) – це методи визначення тональності тексту за допомогою мовних моделей, що основані на використанні нейромереж.

Переваги та недоліки:

- Використовують Word Embeddings (Word2Vec, GloVe, FastText);
- Працюють із контекстом;
- Вимогливі до обчислювальних ресурсів.

Популярні моделі:

- Модель LSTM;
- Модель BEART;
- Модель GPT.

Зупинимося докладно на кожній з них.

11.5.1.3.1. Модель LSTM

LSTM (*Long Short-Term Memory*) — це тип рекурентної нейронної мережі (RNN), який дозволяє ефективно працювати з послідовними даними (текст, аудіо, часові ряди) та уникати проблеми зникнення градієнта.

Архітектура LSTM

LSTM складається з трьох основних «воріт», які керують потоком інформації (рис. 11.2, рис. 11.3).

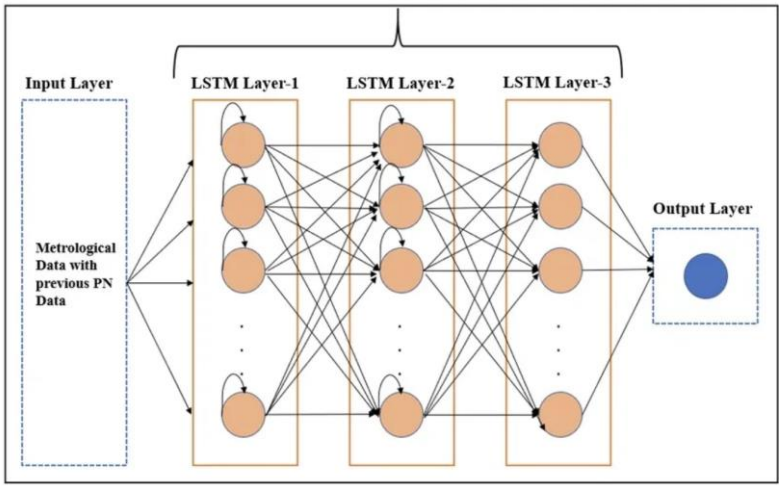


Рис. 11.2. Приклад 1 архітектури LSTM

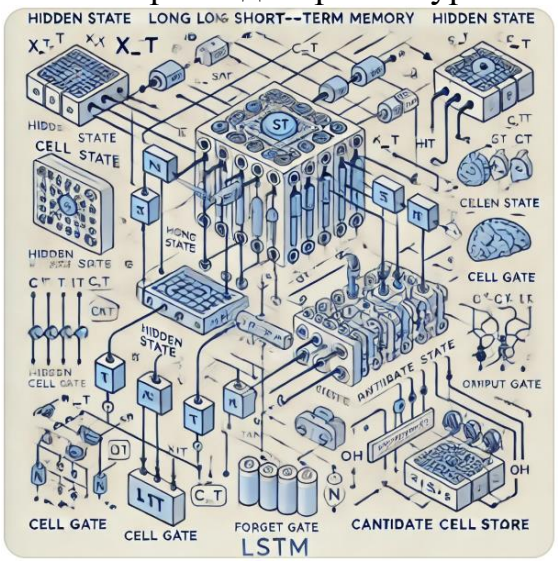


Рис. 11.3. Приклад 2 архітектури LSTM

«*Ворота*» (*gates*) – це спеціальні механізми, які контролюють потік інформації всередині комірок пам'яті. Вони використовують сигмоїдну активацію (σ), щоб вирішити, яку інформацію пропустити, а яку – заблокувати.

1. *Вхідні «ворота»* (*Input Gate*, i_t) – механізм, що визначає, яка нова інформація буде збережена в пам’яті. Обчислюється за формулою (11.7):

$$i_t = \sigma \cdot (W_i \cdot [h_{t-1}, x_t] + b_i), \qquad (11.7)$$

де:

i_t — значення вхідних «воріт» у момент t;

σ — сигмоїдна функція активації, яка обмежує значення в інтервалі (0,1);

W_i — матриця матриця ваг вхідних «воріт»;

h_{t-1} — прихований стан з попереднього кроку $(t-1)$;

x_t — вхідні дані на поточному кроці t (наприклад, слово у реченні);

b_i — зсув (ухил), який допомагає навчати коректніше.

Приклад. Нове слово в реченні – чи має воно значення для контексту?

2. *Забуттєві «ворота»* (*Forget Gate*, f_t) – механізм, що визначає, яка стара інформація буде забута. Обчислюється за формулою (11.8):

$$f_t = \sigma \cdot (W_f \cdot [h_{t-1}, x_t] + b_f), \quad (11.8)$$

де:

f_t — вихідне значення забутєвих «воріт» у момент часу t. Визначає, яку частину попереднього стану пам'яті потрібно зберегти, а яку забути;

σ — сигмоїдна функція активації , яка повертає значення в інтервалі (0,1);

W_f — матриця ваг забуттєвих «воріт»;

h_{t-1} — прихований стан (вихід) із попереднього кроку $(t-1)$;

x_t — входні дані на поточному кроці t;

b_f — зсув для забуттєвих воріт.

Приклад. У попередньому реченні згадувалася тема – чи потрібно її пам'ятати далі?

3. *Вихідні ворота* (*Output Gate*, o_t) – механізм, що визначає, що буде передано в наступний часовий крок. Обчислюється за формулою (11.9):

$$o_t = \sigma \cdot (W_o \cdot [h_{t-1}, x_t] + b_o), \quad (11.9)$$

де:

o_t — вихідне значення вихідних воріт у момент часу t;

σ — сигмоїдна функція активації , яка повертає значення в інтервалі (0,1);

W_0 — матриця ваг вихідних «воріт»;

h_{t-1} — прихований стан (вихід) із попереднього кроку $(t-1)$;

x_t — входні дані на поточному кроці t;

b_o — зсув для вихідних воріт.

Приклад. Що важливого передати в наступний стан мережі?

Оновлення стану пам’яті

Обчислюється за формулою (11.10):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}, \quad (11.10)$$

де:

C_{t-1} — попередній стан пам'яті;

C_t — оновлений стан пам'яті;

i_t — значення вхідних «воріт» у момент t;

f_t — вихідне значення забутєвих «воріт» у момент часу t.

$\widetilde{C_t}$ — нова інформація, яка додається до пам’яті.

Оновлення вихідного стану

Обчислюється за формулою (11.11):

$$h_t = o_t \cdot tanh(C_t) \quad (11.11)$$

де:

h_t — вихідний стан (що передається далі);

o_t — вихідне значення вихідних воріт у момент часу t;

$tanh(C_t)$ — необроблений вихідний сигнал.

Візуалізація процесу

🔵 Забуттєві ворота вирішують, що з пам'яті варто стерти;
 Вхідні ворота додають нову інформацію;
 Вихідні ворота визначають, що передати далі.

Реалізація LSTM у Python (*TensorFlow/Keras*)

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Створюємо модель LSTM
model = Sequential([
    LSTM(50, return_sequences=True, input_shape=(10,
1)),  # 50 нейронів, 10 кроків у послідовності
    LSTM(50),
    Dense(1)  # Вихідний шар для регресії
])

# Компільовуємо модель
model.compile(optimizer='adam', loss='mse')

# Виводимо структуру
model.summary()
```

11.5.1.3.2. Модель BERT

BERT (*Bidirectional Encoder Representations from Transformers*) – це трансформерна модель, розроблена Google у 2018 році, яка дозволяє розуміти контекст слів у реченні бінаправлено (одночасно зліва направо та справа наліво).

Основні особливості BERT:

- На відміну від інших моделей (наприклад, GPT, який читає текст зліва направо), BERT аналізує весь контекст навколо кожного слова, тобто враховує залежності слів у двох напрямках (ліворуч і праворуч);
- Попереднє навчання здійснюється на величезних текстових корпусах;
- Використовує маскування слів ($Masked\ Language\ Model, MLM$);
- Використовує передбачення наступного речення ($Next\ Sentence\ Prediction, NSP$);
- Використовується для глибокого розуміння тексту та завдань NLP NLP, тобто перевершує традиційні підходи у багатьох NLP-задачах (аналіз тональності, розпізнавання сутностей, машинний переклад, питання-відповіді тощо).

Архітектура BERT

BERT базується на трансформерах, а саме на їх енкодерній частині для аналізу контексту слів у реченні (рис. 11.4 та рис. 11.5).

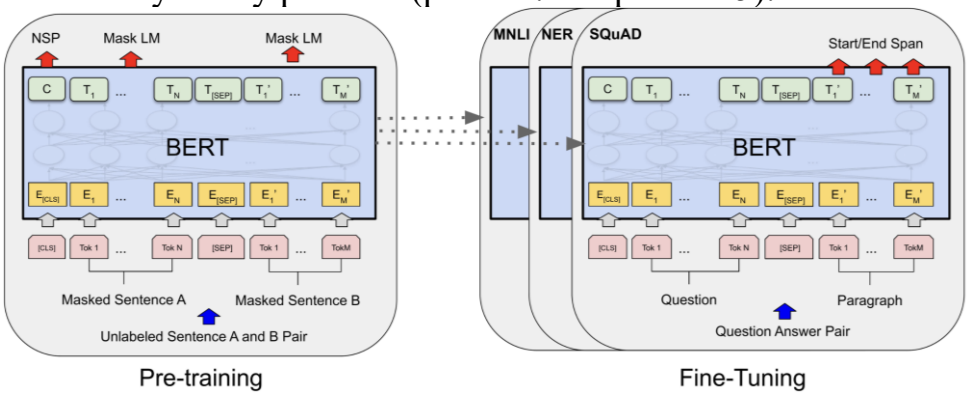


Рис. 11.4. Приклад 1 архітектури BERT

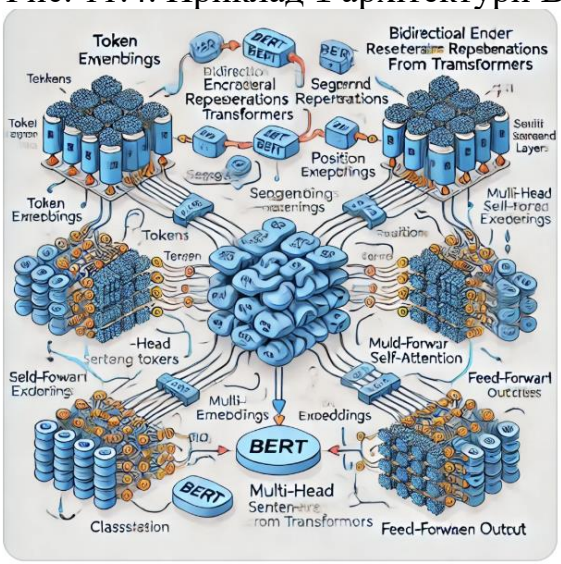


Рис. 11.5. Приклад 2 архітектури BERT

Структура BERT:

- Складається з L шарів енкодера (наприклад, 12 для BERT-Base, 24 для BERT-Large);
- Розмір ембедингів (H): 768 (BERT-Base) або 1024 (BERT-Large);
- Кількість багатоголових механізмів уваги: 12 (BERT-Base) або 16 (BERT-Large).

Два ключові методи попереднього навчання

1. *Масковане мовне моделювання (MLM).*

Принцип роботи:

- У реченні випадково замінюють 15% слів на $[MASK]$;
- Модель повинна відновити ці слова на основі контексту.

Приклад. Нехай *на вхід* подається речення: «Я люблю $[MASK]$ вранці.»

Очікуваний *вихід*: «Я люблю каву вранці.»

Це змушує BERT навчитися контексту в обох напрямках.

2. *Передбачення наступного речення (NSP).*

Принцип роботи:

- Подається дві фрази:
  «Правильне» речення, яке справді йде після першого;
  «Хибне» речення, яке випадково вибране з іншого документа.
- Модель має вирішити, чи друге речення є логічним продовженням першого.

Приклад: *Реальна послідовність*:

«Я прийшов у магазин.»
«Я купив молоко.»

(Модель каже: «Так»).

Випадкове продовження:

«Я прийшов у магазин.»
«Сонце світить яскраво.»

(Модель каже: «Ні»).

Використання BERT у задачах *NLP*

BERT можна тонко налаштовувати (fine-tuning) для конкретних задач:

1. Аналіз тональності (Sentiment Analysis);
2. Відповіді на запитання (Question Answering, QA);
3. Класифікація текстів;
4. Розпізнавання іменованих сутностей (NER);
5. Машинний переклад.

Реалізація BERT у Python (Hugging Face Transformers)

```
from transformers import BertTokenizer, BertModel

# Завантажуємо токенізатор BERT
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

# Токенізуємо текст
text = "Hello, how are you?"
tokens = tokenizer(text, return_tensors="pt")

# Завантажуємо модель BERT
model = BertModel.from_pretrained("bert-base-uncased")

# Пропускаємо через модель
outputs = model(**tokens)

# Витягуємо приховані стани
hidden_states = outputs.last_hidden_state
```

```
print(hidden_states.shape)  # (1, кількість токенів, 768)
```

11.5.1.3.3. Модель GPT

Модель GPT (*Generative Pre-trained Transformer*) — це нейромережева модель, яка генерує текст на основі вхідного запиту. Вона базується на архітектурі трансформерів і працює в авторегресійному режимі, тобто передбачає наступне слово, використовуючи попередній контекст.

Основні характеристики GPT

1. Однобічний трансформер – обробляє текст зліва направо, на відміну від BERT, який є двонаправленим.
2. Навчання у два етапи:

- *Pre-training* (*Попереднє навчання*) – вивчає структуру мови на великому корпусі текстів;
- *Fine-tuning* (*Додаткове навчання*) – адаптується до конкретних завдань, таких як діалогові системи або кодування.

3. Використовує механізм самоуваги (Self-Attention), щоб визначати, які слова в контексті найважливіші.

Архітектура GPT

Архітектура *GPT* представлена на рис. 11.6. та рис. 11.7.

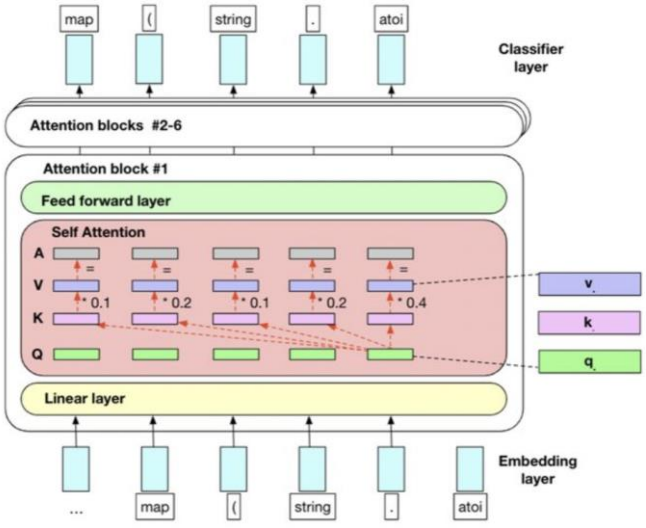


Рис. 11.6. Приклад 2 архітектури моделі GPT

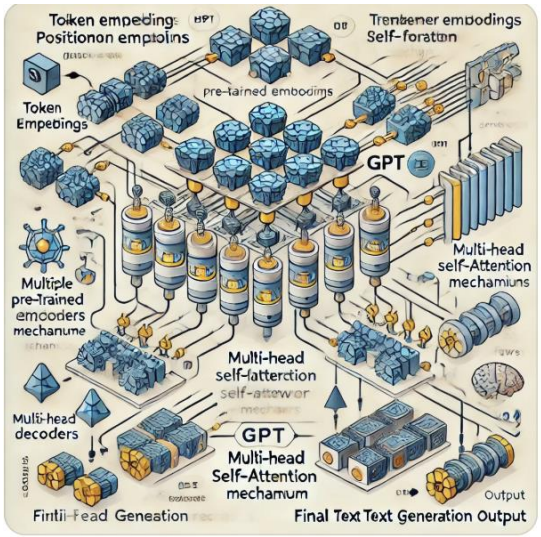


Рис. 11.7. Приклад 2 архітектури моделі GPT

Головні компоненти:

- *Токенізація (Byte-Pair Encoding, BPE)* – розбиває слова на підслова;
- *Ембеддинги токенів* та *позиційні ембеддинги* – кодують слова та їх порядок у реченні;
- *Багатоголовий механізм самоуваги (Multi-Head Self-Attention)* – визначає контекст слова, враховуючи попередні слова;
- *Нормалізація (Layer Normalization)* та *Feed-Forward мережі* – покращують навчання;
- *Логістична регресія (Softmax layer)* – перетворює вектори у ймовірності наступного слова.

Використання GPT

Дана модель широко використовується у наступних задачах:

- Чат-боти (ChatGPT);
- Генерація текстів;
- Автоматичне написання коду;
- Резюмування та перефразування.

Приклад використання GPT у Python

```
from transformers import GPT2LMHeadModel,
GPT2Tokenizer

# Завантажуємо модель GPT-2 та токенізатор
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
model = GPT2LMHeadModel.from_pretrained("gpt2")

# Вхідний текст
input_text = "Artificial intelligence is"

# Токенізація
input_ids = tokenizer.encode(input_text,
return_tensors="pt")
```

```
# Генерація тексту
output = model.generate(input_ids, max_length=50,
num_return_sequences=1)

# Розкодування вихідного тексту
generated_text = tokenizer.decode(output[0],
skip_special_tokens=True)
print(generated_text)
```

11.5.1.3.4. Порівняння LSTM, GPT та BERT

Порівняння моделей LSTM, GPT та BERT представлено в табл. 11.7.

Таблиця 11.7. Порівняння мовних моделей

Модель	Тип	Основна ідея	Переваги	Недоліки
LSTM	Рекурентна нейромережа (RNN)	Обробляє послідовності, пам'ятаючи довгострокові залежності	Добре працює з послідовними даними, наприклад, перекладом	Послідовна обробка, важко масштабувати
GPT	Трансформер (авторегресійний)	Прогнозує наступне слово, обробляє текст зліва направо	Добре генерує зв'язний текст	Погано враховує контекст задніх слів
BERT	Трансформер (масковане навчання)	Використовує двонаправлений контекст	Чудово розуміє контекст, ефективний у класифікації текстів	Не підходить для генерації тексту

11.5.2. Кластеризація текстів

Кластеризація текстів – групування схожих текстів (наприклад, тематичний аналіз новин).

Основні методи кластеризації тексту

1. K-Means;
2. DBSCAN (просторова кластеризація на основі щільності);
3. Ієрархічна кластеризація.

Етапи кластеризації тексту

Етап 1. Збір текстових даних (статті, коментарі, відгуки).
Етап 2. Попередня обробка (лематизація, видалення стоп-слів).
Етап 3. Перетворення в числовий формат (TF-IDF, Word Embeddings).
Етап 4. Застосування алгоритму кластеризації.

Області використання кластеризації текстів

1. Групування новин (на основі теми);
2. Автоматична класифікація документів;

3. Аналіз відгуків клієнтів;
4. Виявлення аномалій (наприклад, спам).

Приклад кластеризації текстів за допомогою *k*-середнє та TF-IDF у Python

Приклад коду

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.cluster import KMeans

# Зразки текстів
texts = [
    "Штучний інтелект змінює світ технологій",
    "Машинне навчання використовується у фінансах",
    "Література формує культуру суспільства",
    "Глибоке навчання допомагає у комп'ютерному
зорі",
    "Мистецтво впливає на творчість людей",
    "Алгоритми класифікації використовуються в
аналізі даних",
    "Філософія досліджує сенс життя",
    "Комп'ютерні науки змінюють індустрії",
    "Поезія розкриває емоції через слова"
]

# Векторизація текстів (TF-IDF)
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)

# Кластеризація (K-Means)
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters,
random_state=42)
labels = kmeans.fit_predict(X)

# Виведення результатів
for i, text in enumerate(texts):
    print(f"Кластер {labels[i]}: {text}")
```

Приклад виводу

```
Кластер 0: Література формує культуру суспільства
Кластер 0: Мистецтво впливає на творчість людей
```

```
Кластер 0: Поезія розкриває емоції через слова

Кластер 1: Штучний інтелект змінює світ технологій
Кластер 1: Машинне навчання використовується у
фінансах
Кластер 1: Глибоке навчання допомагає у комп'ютерному
зорі
Кластер 1: Комп'ютерні науки змінюють індустрії

Кластер 2: Алгоритми класифікації використовуються в
аналізі даних
Кластер 2: Філософія досліджує сенс життя
```

11.5.3. Класифікація текстів

Класифікація текстів – визначення категорії тексту (наприклад, новини, спорт, політика).

Основні етапи класифікації текстів

Етап 1. Збір і підготовка даних.

1.1. Отримання текстів (відгуки, статті, коментарі);

1.2. Попередня обробка:

- Видалення стоп-слів;
- Лематизація / стем;
- Перетворення тексту у вектор.

Етап 2. Перетворення тексту в числовий вигляд.

Популярні методи:

- Bag of Words (BoW);
- TF-IDF;
- Вбудовані слова Word Embeddings.

Етап 3. Вибір моделі.

3.1. Моделі машинного навчання:

- Логістична регресія;
- Naive Bayes;
- RANDOM FOREST;
- SVM.

3.2. Глибоке навчання:

- LSTM;
- CNN для тексту;
- BERT, GPT (трансформери).

Області використання класифікації текстів

1. Фільтрація спаму;
2. Аналіз тональності (Sentiment Analysis);
3. Класифікація новин за темами (спорт, політика, наука);

4. Категоризація запитів у чат-ботах.

Приклад реалізації класифікації текстів у Python за допомогою Naive Bayes

Приклад коду

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline

# Дані (відгуки)
texts = ["Цей фільм чудовий!", "Жахливий сервіс,
більше не прийду.",
         "Дуже сподобався ресторан!", "Гроші на
вітер, не рекомендую."]
labels = [1, 0, 1, 0]  # 1 - позитивний, 0 -
негативний

# Створення пайплайну: TF-IDF + Naive Bayes
model = Pipeline([
    ("vectorizer", TfidfVectorizer()),
    ("classifier", MultinomialNB())
])

# Навчання моделі
model.fit(texts, labels)

# Передбачення нового тексту
new_texts = ["Фільм був цікавий і захоплюючий!",
"Жахлива гра акторів."]
predictions = model.predict(new_texts)

# Вивід результатів
for text, label in zip(new_texts, predictions):
    print(f"{text} → {'Позитивний' if label == 1 else
'Негативний'}")
```

Приклад виводу

```
Фільм був цікавий і захоплюючий! → Позитивний
Жахлива гра акторів. → Негативний
```

11.5.4. Topic Modeling

Topic Modeling (*Latent Dirichlet Allocation*, *LDA*) – виділення основних тем у великому корпусі текстів. Це неконтрольований метод навчання, який автоматично виводить закриті теми в текстах.

Він особливо корисний для кластеризації статей, аналізу соцмереж, дослідження великих текстових корпусів.

Основні алгоритми для тематичного моделювання

1. *Латентний розподіл Діріхле* (*LDA*)

- Один із найпопулярніших методів;
- Вважає, що кожен документ – це набір тем, а кожна тема – набір слів;
- Визначає ймовірність належності слів до тем.

Приклад тем:

- Тема 1: [«наука», «дослідження», «експеримент», «технології»];
- Тема 2: [«спорт», «футбол», «команда», «матч»].

Мінус:

- Чутливий до гіперпараметрів, потребує підбору числа тем.

2. *Факторизація невід'ємної матриці* (*NMF*).

- Використовує матрицю частотних слів;
- Декомпозує її на матрицю слів і матрицю тем;
- Дає чіткіші результати для коротких текстів.

Мінус:

- Не підходить для коротких документів із малою кількістю слів.

3. *BERTopic* (*на основі BERT*).

- Використовує Word Embeddings (BERT, RoBERTa);
- Ви відображає теми семантично , а не просто за частотою;
- Відмінний для аналізу новин, соціальних мереж.

Мінус:

- Потрібно більше обчислювальних ресурсів.

Приклад LDA в Python

Приклад коду

```
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn.decomposition import
LatentDirichletAllocation

# Зразки текстів
texts = [
    "Штучний інтелект та машинне навчання змінюють
світ",
    "Футбол – найпопулярніший вид спорту у світі",
    "Глибоке навчання – це основа комп'ютерного
зору",
```

```
    "Команди змагаються у фіналі чемпіонату",
    "Нейронні мережі покращують технології обробки
мови"
]

# Векторизація текстів (Bag of Words)
vectorizer = CountVectorizer(stop_words="english")
X = vectorizer.fit_transform(texts)

# Застосування LDA
lda = LatentDirichletAllocation(n_components=2,
random_state=42)
lda.fit(X)

# Виведення слів для кожної теми
words = vectorizer.get_feature_names_out()
for topic_idx, topic in enumerate(lda.components_):
    print(f"Тема {topic_idx + 1}: {[words[i] for i in
topic.argsort()[-5:]]}")
```

Приклад виводу

```
Тема 1: ['інтелект', 'навчання', 'нейронні',
'глибоке', 'технології']
Тема 2: ['спорт', 'команди', 'футбол', 'чемпіонату',
'змагаються']
```

11.5.5. Named Entity Recognition

Named Entity Recognition (*NER*) – розпізнавання іменованих сутностей (імена, міста, організації). Це завдання обробки природної мови (NLP), яке визначає іменовані сутності в тексті та класифікує їх за категоріями.

Основні категорії сутностей у NER

- PER $(Person)$ – ім'я людини (наприклад: Ілон Маск, Тарас Шевченко);
- ORG $(Organization)$ – компанія, установа (наприклад: ООН, Google, NASA);
- LOC $(Location)$ – географічні назви (наприклад: Київ, Амазонка, Альпи);
- $DATE$ (Дата) – дата, час (наприклад: 10 липня 2024, 21:00);
- $MONEY$ $(Money)$ – суми грошей (наприклад: $100, 200 грн);
- GPE $(Geopolitical\ Entity)$ – країни, міста, регіони (наприклад, Україна, ЄС).

NER можна навчити на корпусі спеціальних даних або використовувати вже готові моделі.

Методи для NER

1. Ручне створення правил (регулярні вирази, словники);
2. Класичні ML-моделі (SVM, Decision Trees);
3. Глибоке навчання (LSTM, BERT).

11.5.6. Візуалізація та інтерпретація результатів

Аналіз тексту (NLP) генерує велику кількість числових даних, які важливо правильно візуалізувати та інтерпретувати.

Візуалізація результатів аналізу текстів – це процес представлення текстових даних у графічній формі для полегшення розуміння, аналізу та інтерпретації їхніх характеристик, структури та змісту.

Інтерпретація результатів аналізу текстів – це процес розуміння та пояснення отриманих даних після застосування методів обробки, аналізу та моделювання текстової інформації.

Основні методи візуалізації

1. *Словесна хмара (Word Cloud).* Візуалізує найчастіші слова в тексті (більший розмір – частіше слово) (рис. 11.8). Добре підходить для аналізу тематики статей, коментарів, новин.

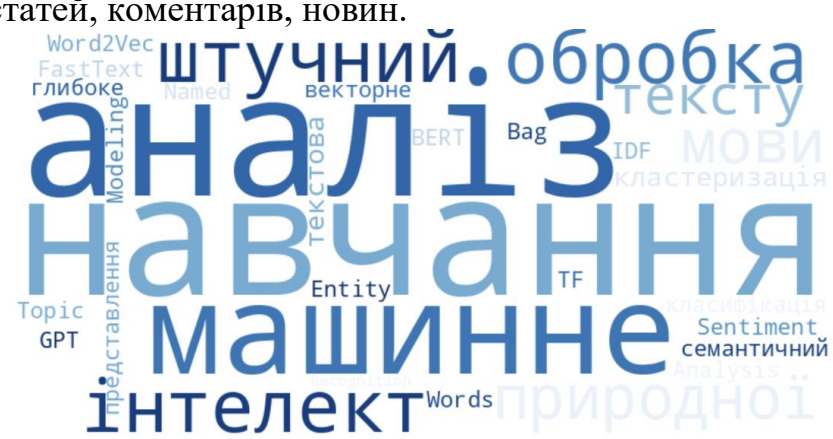


Рис. 11.8. Приклад словесної хмари, створеної на основі термінів, пов’язаних з аналізом текстів і машинним навчанням

2. *Візуалізація частоти слів.* Побудова гістограми найбільш популярних слів у тексті (рис. 11.9).

Рис. 11.9. Приклад візуалізації частоти слів у тексті

3. *Візуалізація тематичного моделювання (Topic Modeling).* Використання LDA (Latent Dirichlet Allocation) для виявлення тем у тексті (рис. 11.10).

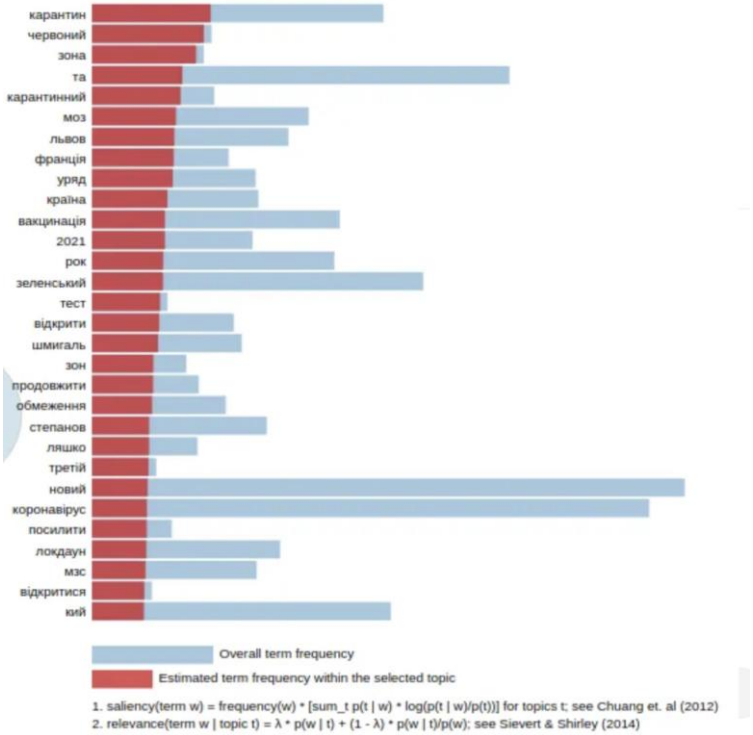


Рис. 11.10. Приклад топ-30 найбільш релевантних тем

4. *Візуалізація векторних зображень (Word Embeddings).* Використання t-SNE або UMAP для зменшення розмірів Word2Vec, FastText, BERT.

t-SNE (*t-Distributed Stochastic Neighbor Embedding*) — це метод зменшення розмірності, який використовується для візуалізації високорозмірних даних у двовимірному або тривимірному просторі. Його зазвичай застосовують для аналізу великих наборів даних, де важливо зберігати подібність між об'єктами.

UMAP (*Uniform Manifold Approximation and Projection*) — це ще один популярний метод зменшення розмірності, схожий на t-SNE, але з деякими ключовими перевагами. UMAP швидший та здатний працювати з великими

наборами даних, а також зберігає глобальні структури даних краще, ніж t-SNE.

Приклад векторного представлення слів наведено на рис 11.11.

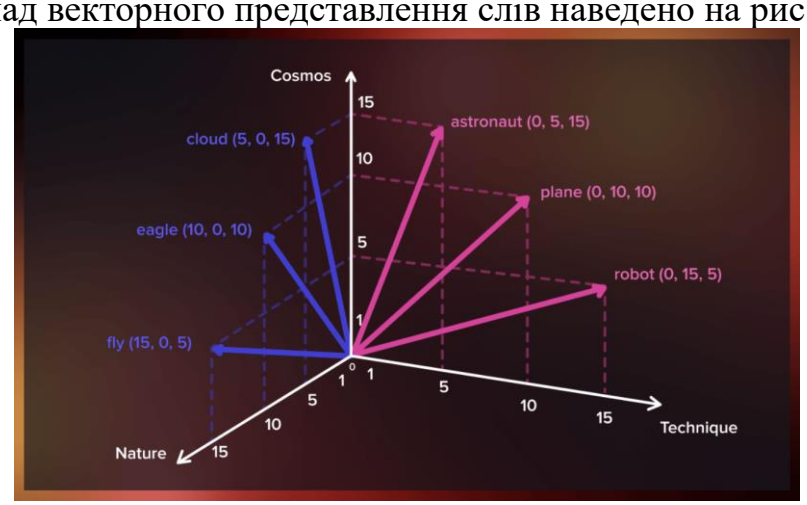


Рис. 11.11. Представлення слова у векторній формі

5. *Аналіз зв'язків між словами (Graph Representation)* – використовується для аналізу співзгаданих слів та побудови семантичних графів (рис. 11.12).

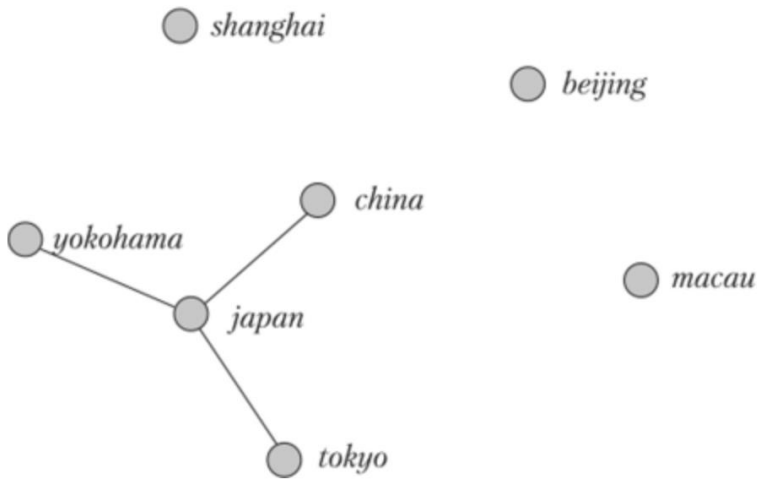


Рис. 11.12. Граф семантичних зв'язків між термінами

11.6. Задача генерації віршів

Генерація віршів є складним завданням у сфері обробки природної мови (NLP), яке передбачає створення текстів із дотриманням ритмічних, стилістичних і семантичних закономірностей. Це одна з форм творчого текстового моделювання, яка вимагає розуміння структури віршів, рими та змістового зв’язку між рядками [51].

11.6.1. Формалізація задачі генерації віршів

Завдання можна сформулювати так:

Нехай є вхідний контекст (тема, перші рядки або стиль) – нехай ми маємо n віршів, що були зібрані з різних електронних джерел. Така інформація для зручності наводяться у таблиці 12.8.

Таблиця 12.8. Представлення текстів

Вірш	Автор і Назва

Вірш 1	Автор 1, Назва 1
...	...
Вірш n	Автор n, Назва n

Потрібно згенерувати послідовність слів, що відповідає певному віршованому формату:

- Віршовий розмір (ямб, хорей, дактиль, тощо).
- Римування (парне, перехресне, кільцеве тощо).
- Семантична узгодженість (логічний розвиток теми, контекст).

Для зручності розглянемо критерії детальніше.

Контекст – у вірші моделюється послідовний розвиток взаємопов'язаних подій.

Рима – останні склади слів у кінці строфи є ідентичними або вимовляються однаково.

Ритм – виконується чергування складів для легкості читання віршів.

Іншими словами ми повинні створити модель нейронної мережі, яка буде створювати словник слів, який буде враховувати положення слова в строфі відносно інших сліві і у різних контектсах. Крім цього модель повинна навчитися правил граматики, римування та слідкувати за метрикою.

Приклади застосування

1. Генерація віршів на задану тему (наприклад, «осінь», «кохання»).
2. Імітація стилю конкретного поета (Шевченко, Ліна Костенко, Байрон тощо).
3. Допомога письменникам у створенні поетичних творів.
4. Автоматичне римування слів для пісень та репу.

11.6.2. Математична постановка задачі

Нехай маємо множину слів V (словник), де кожне слово позначимо як w_i. Завдання генерації тексту можна подати як пошук оптимальної послідовності слів $W = (w_1, w_2, \ldots, w_n)$, яка:

- відповідає заданій темі або контексту;
- зберігає ритмічні обмеження;
- дотримується правил римування.

Формально, необхідно знайти послідовність слів W^*, яка максимізує ймовірність (11.12):

$$W^* = arg\left(\max_W P(W \mid C)\right), \tag{11.12}$$

де C — початковий контекст (наприклад, задана тема, перший рядок або стиль).

Оскільки ймовірність повної послідовності слів можна розкласти за правилом ланцюга (11.13):

$$P(W \mid C) = P(w_1 \mid C) \cdot P(w_2 \mid w_1, C) \cdot P(w_3 \mid w_1, w_2, C) \cdot \ldots \cdot \\ \cdot P(w_n \mid w_1, \ldots, w_{n-1}, C), \tag{11.13}$$

то задача генерації віршів зводиться до покрокового прогнозування наступного слова на основі попередніх.

11.6.3. Додавання поетичних обмежень

Щоб отримати коректний вірш, додаються поетичні обмеження:

Ритмічні обмеження

Ритм у віршах визначається схемою наголосів у словах. Формально, ритм можна описати як послідовність наголошених (S) і ненаголошених (U) складів (11.14):

$$R = (r_1, r_2, \ldots, r_n), r_i \in \{S, U\}. \quad (11.14)$$

Генероване речення має відповідати заданій ритмічній схемі. Для цього можна ввести функцію штрафу (11.15):

$$L_{\text{ритм}} = \sum_{i=1}^{n} \delta(r_i \neq \hat{r}_i), \quad (11.15)$$

де:

$\hat{r}_i$ — бажаний наголос для i-го слова;

δ — індикаторна функція, яка дає 1, якщо наголос не збігається, і 0, якщо збігається.

Римування

Рими можна формалізувати через функцію схожості звуків між кінцівками рядків (11.16):

$$S_{\text{рифа}}(w_i, w_j) = cos(\phi(w_i), \phi(w_j)), \quad (11.16)$$

де $\phi(w)$ — акустичне представлення слова (наприклад, фонетичний вектор). Якщо використовується певна схема римування (наприклад, $AABB$ або $ABAB$), можна додати обмеження (11.17):

$$L_{\text{рифа}} = \sum_{\text{пар}(wi,wj)\in R}(1 - S_{\text{рифа}}(w_i, w_j)), \quad (11.17)$$

де R — множина пар слів, які повинні римуватися.

Семантична узгодженість (контекст)

Щоб текст був змістовним, кожен рядок має логічно продовжувати попередній. Це можна моделювати через косинусну подібність векторів слів (на основі BERT, Word2Vec, GPT тощо) (11.18):

$$S_{\text{сем}}(W) = \sum_{i=1}^{n} cos(f(w_i), f(w_{i-1})), \quad (11.18)$$

де $f(w)$ — семантичне представлення слова або рядка.

Фінальна функція втрат для оптимізації генерації віршів (11.19):

$$L_{\text{заг}} = -ln(P(W \mid C)) + \lambda_1 \cdot L_{\text{ритм}} + \lambda_2 \cdot L_{\text{рифа}} + \lambda_3 \cdot (1 - S_{\text{сем}}(W)), \quad (11.19)$$

де $\lambda_1, \lambda_2, \lambda_3$ — вагові коефіцієнти для контролю впливу ритму, рими та семантики.

Логарифм перед ймовірністю використовується з метою покращення числової стабільність і зручність оптимізації.

Таким чином, ключова формула для обчислення матиме вигляд (11.20):

$$W^* = arg\left(\max_W\left(P(W \mid C) + ln\left(P(W \mid C) - \lambda_1 \cdot L_{\text{ритм}} - \lambda_2 \cdot L_{\text{рифа}} - \lambda_3 \cdot (1 - S_{\text{сем}}(W))\right)\right)\right). \quad (11.20)$$

де:

$ln(P(W \mid C))$ — правдоподібність послідовності вірша;
$L_{\text{ритм}}$ — штраф за неправильний ритм;
$L_{\text{рифа}}$ — штраф за відсутність рими;
$S_{\text{сем}}(W)$ — семантична узгодженість (чим ближче до 1, тим краще).

11.6.4. Методи генерації віршів

Статистичні методи

N-грамні моделі:

- Генерують текст на основі ймовірності появи слів після попередніх;
- Недолік: не зберігають довготривалу семантику, можуть давати безглузді результати.

Марковські ланцюги:

- Кожне слово або рядок вибирається на основі ймовірностей переходу між станами;
- Підходить для простих римованих текстів, але не враховує контекст глибше ніж 1-2 рядки.

Машинне навчання та нейромережі

LSTM та GRU:

- Генерація тексту на основі пам'яті про попередні слова;
- Навчання на великих корпусах віршів певного автора або стилю;
- Можуть враховувати ритм та стиль, але іноді повторюють фрази.

GPT:

- Використовує глибоку трансформерну архітектуру для створення осмислених текстів;
- GPT-3, GPT-4 або спеціалізовані моделі можуть генерувати поетичні тексти з римою та стилем;
- Мінус: потребує додаткової обробки для підтримки певного розміру рядка та рими.

Fine-tuning BERT для віршів:

- Використання BERT для вибору найбільш ймовірного рядка, що відповідає заданій римі або стилю;
- Може працювати в поєднанні з іншими моделями, але самостійно менш ефективний для генерації.

RL + Poetic Constraints (Посилене навчання + обмеження поезії):

- Моделі навчаються не тільки прогнозувати наступні слова, а й оптимізувати структуру вірша за заданими правилами (рими, складовий розмір тощо);
- Використовується, наприклад, для створення хайку або сонетів.

Приклад генерації вірша за допомогою ланцюгів Маркова наведено на рис. 11.13.

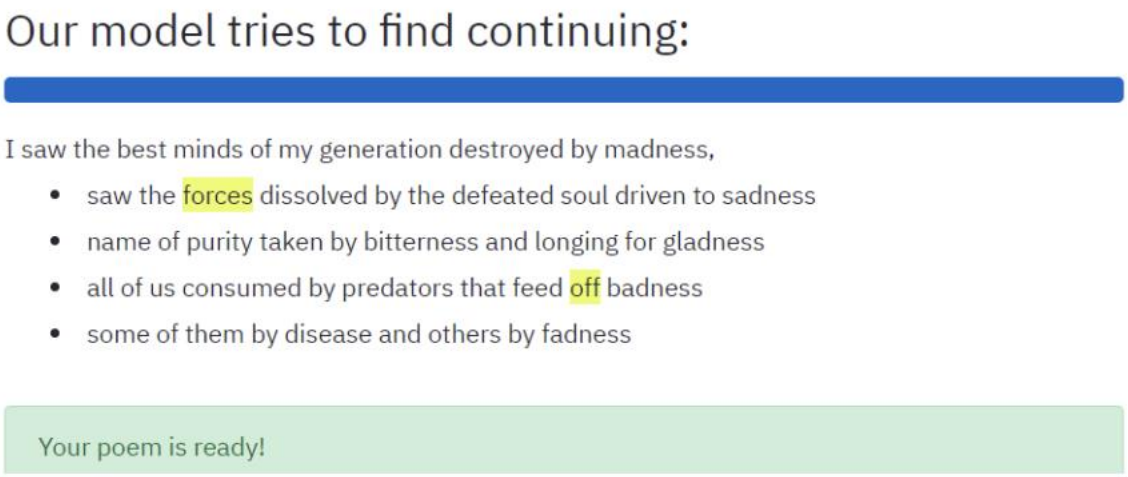


Рис. 11.13. Згенеровані вірші

Приклад вірша згенерованого за допомогою мовної моделі BEART наведено на рис. 11.14.

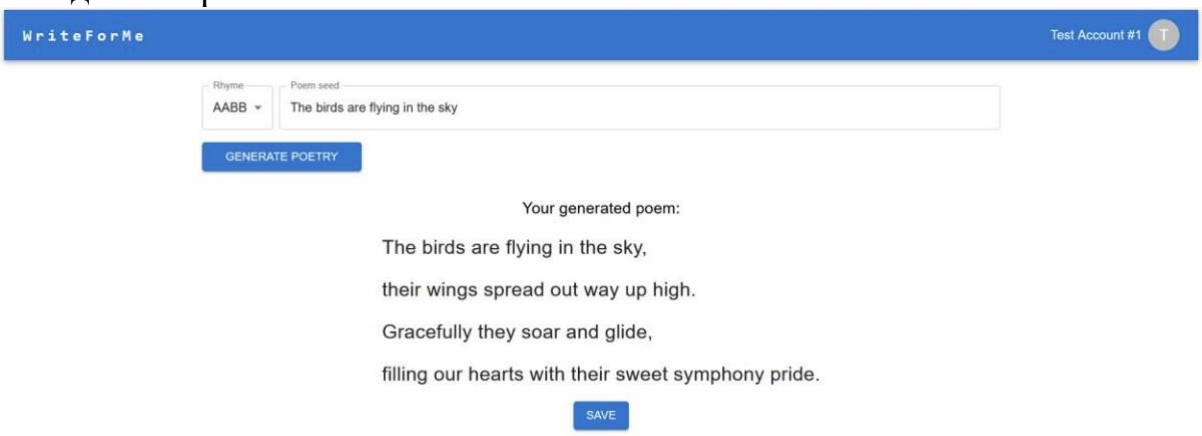


Рис. 11.14. Згенеровані вірші

11.6.5. Приклад генерації вірша в Python

Реалізація

```
import torch
from transformers import GPT2LMHeadModel,
GPT2Tokenizer
import random

# Завантаження моделі GPT-2
model_name = "sberbank-ai/rugpt3small_based_on_gpt2"
# Для українських віршів можна спробувати ruGPT-3
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)

def generate_poem(prompt, max_length=50,
temperature=0.7, num_return_sequences=1):
```

```
    input_ids = tokenizer.encode(prompt,
return_tensors="pt")

    with torch.no_grad():
        output = model.generate(
            input_ids,
            max_length=max_length,
            temperature=temperature,

num_return_sequences=num_return_sequences,
            no_repeat_ngram_size=2,
            top_k=50,
            top_p=0.95
        )

    poem = tokenizer.decode(output[0],
skip_special_tokens=True)
    return poem

# Початковий рядок (можна змінити)
prompt = "Осінній вітер шепоче стиха"
generated_poem = generate_poem(prompt)

print(" Згенерований вірш:\n")
print(generated_poem)
```

Приклад згенерованого вірша

```
Осінній вітер шепоче стиха,
Листя кружляє, співає душа.
Дощ упаде, залишивши на вікнах
Рядки розлуки, де світ неживий.
```

ЛЕКЦІЯ 12

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТЕКСТУ (ПРОДОВЖЕННЯ)

12.1. Виявлення плагіату в тексті

12.1.1. Означення та види плагіату

Плагіат — це використання чужих ідей, текстів, досліджень, робіт або творів без належного посилання на оригінальні джерела або без дозволу на їх використання. Це порушення авторських прав і етики, яке часто трапляється в академічній, творчій та науковій діяльності [50].

Основні види плагіату

1. *Прямий плагіат* – це використання чужого тексту, ідей або роботи без змін і без належного посилання на джерело.

Наприклад, коли студент копіює абзац із книги або статті і вставляє його у свою роботу, не вказуючи автора чи джерело.

2. *Перефразування* (*переписування*) – це використання чужих ідей або інформації, але переписаних своїми словами без належного посилання на оригінальний джерело.

Цей вид плагіату може включати невеликі зміни в словах, але зберігається основна ідея чи структура.

3. *Плагіат через неправильне цитування* – це коли використовуються цитати або джерела, але вони оформлені неправильно або без посилання на конкретне джерело.

Наприклад, не вказано точне місце цитати, або не наведено належну бібліографію.

4. *Самоплагіат* – це використання власних раніше опублікованих робіт без вказівки, що цей матеріал вже був опублікований.

Наприклад, якщо автор публікує свою наукову статтю в іншому журналі, не повідомивши, що вона була опублікована раніше.

5. *Мозаїчний плагіат* – це техніка, коли частини тексту з різних джерел збираються разом без належного оформлення, створюючи враження, що текст є оригінальним.

Наприклад, коли автор бере кілька фрагментів із різних статей або книг і об'єднує їх у нову роботу, не цитуючи джерела.

6. *Ідеї та концептуальний плагіат* – це коли запозичуються не тільки конкретні слова або фрази, але й самі ідеї, концепції чи теорії без посилання на оригінальні джерела.

Наприклад, запозичення ідеї, концепції або наукового підходу без вказівки на першоджерело, навіть якщо словесно виразити це по-іншому.

7. *Плагіат з використанням відомих ідей або відкритих джерел* – це використання відомих ідей, концепцій або даних без належного вказування джерела, навіть якщо це загальнодоступні матеріали.

Важливо вказати, що джерело є відкритим, якщо ви користуєтесь даними з відкритих баз, онлайн-ресурсів чи інших публікацій.

Способи уникнення плагіату

1. *Цитування правильним чином.* Завжди слід наводити точне посилання на джерело, коли використовуються чужі ідеї, дані чи цитати.
2. *Переписування своїми словами.* Якщо необхідно передати чиюсь ідею, краще перефразувати її та все одно посилатися на оригінал.
3. *Перевірка правила цитування.* Різні види робіт (академічні, творчі тощо) вимагають різного формату посилань (APA, MLA, Chicago, тощо).
4. *Використання інструментів перевірки плагіату.* Вони допоможуть виявити потенційні порушення і уникнути недопустимих збігів з іншими роботами.

Запобігання плагіату є важливим аспектом академічної та професійної чесності. Правильне посилання на джерела і дотримання етики дозволяє уникнути порушення авторських прав і зберегти довіру у науковому середовищі.

12.1.2. Постановка задачі пошуку співпадінь в текстах

Задача пошуку співпадінь у наборі текстів полягає у пошуку спільних фрагментів тексту.

На вхід подається текст X, який необхідно проаналізувати, $X = \{x_1, x_2, \dots, x_n\}$, x_i – слова тексту.

Також має бути подано текст Y для порівняння, за умови, якщо у користувача є підозра на плагіат $Y = \{y_1, y_2, \dots, y_m\}$, y_j – слова тексту. В разі, якщо такої інформації немає, то необхідно знайти подібний текст в мережі Інтернет.

На виході отримуються значення числових метрик подібності (зокрема коефіцієнт Жаккара та коефіцієнт косинусної подібності), які показують рівень співпадіння в текстах X та Y (або з текстом, знайдений в мережі Інтернет). На основі визначених співпадінь формуються рекомендації щодо того, чи є текст, який аналізується, плагіатом чи ні. Проте остаточне рішення щодо цього має приймати особа, що приймає рішення (ОПР), – користувач.

Критерії оцінки результатів:

- *Точність* $(Precision, Recall)$. Визначити наскільки коректно знаходяться співпадіння.
- *Продуктивність.* Визначити швидкість роботи алгоритму.
- *Масштабованість* Визначити, чи можна застосувати алгоритм для великої кількості текстів.

12.1.3. Алгоритми перевірки на плагіат

12.1.3.1. Алгоритм шинглів

Алгоритм шинглів (*Shingling*) є одним із методів для виявлення схожості між текстами, що широко використовується для виявлення плагіату або в задачах пошуку дубльованого контенту.

Шингли — це послідовності з n елементів (зазвичай слів чи символів), що виділяються з тексту і можуть бути використані для порівняння та виявлення схожості між різними текстами.

Переваги:

- Простота і ефективність для порівняння великих обсягів тексту.
- Може виявляти плагіат на рівні фрагментів, а не лише за допомогою точних копій.

Недоліки:

- Може не виявляти схожість при великій зміні структури тексту.

Основні етапи алгоритму шинглів

1. *Розбиття тексту на шингли.* Текст поділяється на послідовності слів або символів (шингли). Зазвичай використовуються n-грами для формування шинглів.

Наприклад, для тексту «$The\ economy\ is\ growing\ fast$» можна побудувати шингли з 2 слів (бі-грами), 3 слів (три-грами) і т.д.

2. *Створення множини шинглів.* Для кожного тексту створюється множина всіх можливих шинглів, що утворюються з нього.

Наприклад, для тексту «$The\ economy\ is\ growing\ fast$» бі-грам (2-грам):

«$The\ economy$», «$economy\ is$», «$is\ growing$», «$growing\ fast$».

3. *Порівняння шинглів між текстами.* Порівнюються множини шинглів з різних текстів. Схожість між текстами можна оцінити за допомогою методу подібності множин, наприклад, за допомогою коефіцієнту подібності Жаккара (Jaccard similarity) (12.1):

$$J(X,Y) = Jaccard\ similarity = \frac{|X \cap Y|}{|X \cup Y|}, \qquad (12.1)$$

де:

$| X \cap Y |$ — кількість спільних елементів між множинами слів X і Y;

$| X \cup Y |$ — кількість всіх елементів у обох множинах слів.

Слід зазначити, що замість формули (12.1) може бути використано будь-який коефіцієнт подібності, про які йшлося в лекції 6.

Покроковий опис алгоритму

Крок 1. Попередня обробка тексту:

- Перетворення тексту в нижній регістр.
- Видалення розділових знаків і зайвих символів.
- Розбиття на слова (токенізація).

Крок 2. Виділення шинглів:

- Для кожного тексту обирається розмір шинглів (n). Наприклад, для бі-грам $(n = 2)$ це будуть двоелементні послідовності.
- Формуються шингли з n елементів (зазвичай слів).

Крок 3. Обчислення подібності:

- Після того, як для кожного тексту створено множини шинглів, порівнюються ці множини між собою за допомогою метрики схожості, такої як коефіцієнт подібності Жаккара.

Крок 4. Інтерпретація результату:

- Алгоритм виводить подібність між двома текстами на основі співпадіння їх шинглів. Якщо схожість велика (наприклад, понад 0,7), це може свідчити про те, що тексти подібні.

Загалом, для визначення міри подібності варто застосувати наступну шкалу (табл. 12.1) за аналогією зі шкалою Чеддока.

Таблиця 12.1. Шкала виявлення сили подібності

Значення коефіцієнта подібності	0,0 – 0,3	0,3 – 0,5	0,5 – 0,7	0,7 – 0,9	0,9 – 1,0
Характеристика сили подібності	слабка	помірна	помітна	сильна	дуже сильна

Приклад реалізації алгоритму шинглів в Python

```
import re

# Функція для попередньої обробки тексту
def preprocess(text):
    # Перетворюємо текст в нижній регістр та
видаляємо небажані символи
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    return text

# Функція для побудови шинглів
def generate_shingles(text, n=2):
    words = text.split()  # Розбиваємо текст на слова
    shingles = set()
    for i in range(len(words) - n + 1):
        shingle = tuple(words[i:i + n])
        shingles.add(shingle)  # Додаємо шингл у
множину
    return shingles

# Приклад текстів
text1 = "The economy is growing fast."
text2 = "The financial sector is growing quickly."

# Попередня обробка
text1 = preprocess(text1)
```

```
text2 = preprocess(text2)

# Генерація бі-грам (шинглів з 2 слів)
shingles1 = generate_shingles(text1, 2)
shingles2 = generate_shingles(text2, 2)

# Обчислення подібності Jaccard
intersection = len(shingles1.intersection(shingles2))
union = len(shingles1.union(shingles2))
similarity = intersection / union

print(f"Jaccard Similarity: {similarity:.2f}")
```

Застосування

Виявлення плагіату: Алгоритм шинглів дозволяє виявляти схожість між текстами, що може бути використано для перевірки на плагіат.

Пошук дубльованого контенту: Це може бути корисним для виявлення дубльованого контенту в Інтернеті.

Аналіз тексту: Застосовується для порівняння текстів у різних контекстах, таких як машинний переклад, автоматичне підсумовування тощо.

12.1.3.2. Метод на основі векторів та косинусної подібності

Метод на основі векторів та косинусної подібності (*Vector Space Model*) – це метод, що полягає в перетворенні тексту у векторну форму за допомогою таких моделей, як TF-IDF, після чого обчислюється косинусна подібність (Cosine Similarity) між векторами (12.2):

$$cos(\theta) = Cosine\ Similarity(X, Y) = \frac{(X \cdot Y)}{\|X\| \cdot \|Y\|}, \qquad (12.2)$$

де:

$(X \cdot Y)$ — скалярний добуток між векторами X і Y;

$\| X \|$ і $\| Y \|$ — норми (довжини) векторів X і Y.

Переваги:

- Виявляє схожість за змістом, навіть якщо текст переписаний;
- Зручний для порівняння довгих текстів.

Недоліки:

- Не завжди точний при порівнянні текстів з різною структурою.

Кроки алгоритму

Крок 1. Перетворити текст у вектор з використанням TF-IDF.
Крок 2. Обчислити косинусну подібність між векторами двох текстів.
Крок 3. Якщо косинусна подібність висока, це вказує на наявність плагіату.

Приклад реалізації в Python

Реалізація

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.metrics.pairwise import
cosine_similarity

# Приклад текстів
documents = [
    "The economy is growing fast, there are many
opportunities in the financial sector.",
    "Healthcare is one of the most important sectors
in modern economies.",
    "Technology is revolutionizing the world,
especially the IT and AI sectors.",
    "The education system needs reforms to meet the
growing demands of society.",
    "Climate change is a major global issue that
requires immediate action."
]

# 1. Перетворення текстів у TF-IDF вектори
vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = vectorizer.fit_transform(documents)

# 2. Обчислення косинусної подібності між документами
cosine_similarities = cosine_similarity(tfidf_matrix,
tfidf_matrix)

# 3. Виведення результату (матриця косинусної
подібності)
print(cosine_similarities)
```

Результат

Виведена матриця косинусної подібності буде виглядати, наприклад, так:

```
[[1.         0.44404339 0.32501741 0.23201147
0.22476388]
 [0.44404339 1.         0.51977801 0.34552289
0.29055012]
 [0.32501741 0.51977801 1.         0.38307211
0.40225085]
 [0.23201147 0.34552289 0.38307211 1.
0.24812944]
```

```
 [0.22476388 0.29055012 0.40225085 0.24812944 1.
]]
```

- 1 на головній діагоналі означає, що кожен документ має максимальну подібність до самого себе.
- Інші значення вказують на подібність між різними документами.

Наприклад, значення 0,44404339 показує, як подібні перший і другий документи.

Інтерпретація результатів:

- Вищі значення в матриці свідчать про більшу схожість між документами.
- Нижчі значення вказують на меншу схожість між текстами.

12.1.3.3. Locality Sensitive Hashing

Locality Sensitive Hashing (*LSH*) — це метод хешування, який використовується для пошуку схожих елементів у великій множині даних. Основна ідея LSH полягає в тому, щоб перетворювати схожі об'єкти в однакові або близькі хеші, що дозволяє ефективно знаходити подібні об'єкти без повного порівняння всіх можливих пар.

LSH застосовується в багатьох сферах, зокрема:

- Виявлення плагіату (шинглування + LSH);
- Пошук схожих зображень;
- Пошук найближчих сусідів у великих наборах даних;
- Пошук дублікатів у великих текстових колекціях.

Переваги LSH:

- *Швидкість.* LSH дозволяє знайти схожі документи швидше, ніж повне порівняння всіх пар;
- *Масштабованість.* Підходить для великих наборів даних;
- *Гнучкість.* Можна змінювати параметри (кількість хеш-функцій, бандів) для кращої точності або швидкості.

Недоліки LSH:

- *Точність залежить від параметрів.* Потрібно налаштовувати кількість хеш-функцій і бандів;
- *Неможливість 100% точного пошуку.* LSH знаходить «ймовірні» збіги, а не всі можливі.

Принцип роботи LSH

1. *Перетворення тексту в вектори* (наприклад, за допомогою TF-IDF, шинглів або іншого методу);
2. *Обчислення хешів.* Використовується кілька хеш-функцій, які зберігають схожі об'єкти в тих самих або сусідніх «відрах» (buckets);
3. *Пошук схожих елементів.* Замість порівняння всіх можливих пар, шукаємо об'єкти в однакових відрах.

Реалізація LSH у Python

```
import numpy as np
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.metrics.pairwise import
cosine_similarity
import random
from collections import defaultdict

# Зразки текстів
documents = [
    "The economy is growing fast, there are many
opportunities in the financial sector.",
    "Healthcare is one of the most important sectors
in modern economies.",
    "Technology is revolutionizing the world,
especially the IT and AI sectors.",
    "The education system needs reforms to meet the
growing demands of society.",
    "Climate change is a major global issue that
requires immediate action."
]

# 1. Векторизація текстів за допомогою TF-IDF
vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix =
vectorizer.fit_transform(documents).toarray()

# 2. Функція для генерації випадкових векторів (хеш-
функцій)
def generate_random_vectors(dim, num_vectors):
    return np.random.randn(num_vectors, dim)

# 3. Функція для обчислення хешів LSH
def compute_lsh_hashes(vectors, random_vectors):
    return (np.dot(vectors, random_vectors.T) >
0).astype(int)

# Кількість хеш-функцій та сегментів (змінюйте для
точності)
num_hash_functions = 10
num_bands = 5

# Генеруємо випадкові вектори
```

```
random_vectors =
generate_random_vectors(tfidf_matrix.shape[1],
num_hash_functions)

# Обчислюємо LSH хеші
hashes = compute_lsh_hashes(tfidf_matrix,
random_vectors)

# 4. Створення "відер" (buckets)
buckets = defaultdict(set)
for doc_id, hash_vector in enumerate(hashes):
    for band in range(num_bands):
        band_hash = tuple(hash_vector[band *
(num_hash_functions // num_bands):(band + 1) *
(num_hash_functions // num_bands)])
        buckets[band_hash].add(doc_id)

# 5. Пошук схожих документів
def find_similar_documents(doc_id, buckets):
    similar_docs = set()
    hash_vector = hashes[doc_id]
    for band in range(num_bands):
        band_hash = tuple(hash_vector[band *
(num_hash_functions // num_bands):(band + 1) *
(num_hash_functions // num_bands)])
        similar_docs.update(buckets[band_hash])
    similar_docs.discard(doc_id)  # Видаляємо сам
документ
    return similar_docs

# Приклад: знаходимо схожі документи для першого
документа
similar_docs = find_similar_documents(0, buckets)
print(f"Документи, схожі на документ 0:
{similar_docs}")
```

12.1.3.4. Метод лінійних хеш-функцій

Метод лінійних хеш-функцій (*MinHash*) — це метод, який використовується для швидкого обчислення схожості між множинами шинглів, часто використовується разом з алгоритмом Locality Sensitive Hashing (LSH) для знаходження схожих документів у великих наборах даних.

Кроки алгоритму MinHash

Крок 1. Створити набір шинглів для обох текстів.
Крок 2. Для кожного набору шинглів обчислити MinHash (вибираючи мінімальний хеш серед усіх можливих хешів для шинглів).
Крок 3. Використовувати LSH для порівняння схожості хешів.

Переваги:

- Дуже швидко обчислюється для великих наборів даних;
- Підходить для роботи з великими документами або базами даних.

Недоліки:

- Може не забезпечити високу точність при низькому значенні схожості.

Реалізація MinHash у Python

```
import random
import numpy as np
from sklearn.feature_extraction.text import
TfidfVectorizer

# Функція для створення шинглів (n-грам)
def get_shingles(text, k=3):
    """Функція розбиває текст на множину шинглів
довжиною k"""
    return set([text[i:i + k] for i in
range(len(text) - k + 1)])

# Функція для генерації випадкових хеш-функцій
def generate_hash_functions(num_hashes, max_shingle):
    """Генерує випадкові коефіцієнти a і b для хеш-
функцій"""
    return [(random.randint(1, max_shingle),
random.randint(0, max_shingle)) for _ in
range(num_hashes)]

# Функція обчислення MinHash підпису
def compute_minhash_signature(shingles,
hash_functions, max_shingle):
    """Обчислює MinHash підпис для даного набору
шинглів"""
    signature = []
    for a, b in hash_functions:
        min_hash = min(((a * shingle + b) %
max_shingle) for shingle in shingles)
        signature.append(min_hash)
```

```
    return signature

# Функція обчислення подібності між підписами MinHash
def jaccard_similarity_minhash(sig1, sig2):
    """Обчислює подібність між двома MinHash-підписами"""
    return sum(1 for i in range(len(sig1)) if sig1[i] == sig2[i]) / len(sig1)

# Текстові документи для порівняння
doc1 = "The economy is growing fast, there are many opportunities in the financial sector."
doc2 = "The financial sector is full of opportunities as the economy is expanding rapidly."

# Налаштування параметрів MinHash
num_hashes = 100   # Кількість хеш-функцій
shingle_size = 5   # Довжина шинглів
max_shingle = 2**32 - 1  # Велике просте число для хешування

# Отримання шинглів
shingles1 = get_shingles(doc1, shingle_size)
shingles2 = get_shingles(doc2, shingle_size)

# Генерація випадкових хеш-функцій
hash_functions = generate_hash_functions(num_hashes, max_shingle)

# Обчислення MinHash-підписів
signature1 = compute_minhash_signature(shingles1, hash_functions, max_shingle)
signature2 = compute_minhash_signature(shingles2, hash_functions, max_shingle)

# Обчислення подібності
similarity = jaccard_similarity_minhash(signature1, signature2)

print(f"Оцінена схожість між документами (MinHash Jaccard Similarity): {similarity:.4f}")
```

12.1.3.5. Порівняння Shingling, Vector Space Model, Locality Sensitive Hashing, MinHash

Порівняння алгоритмів виявлення плагіату наведено в табл. 12.2.

Таблиця 12.2. Порівняння Shingling, Vector Space Model, Locality Sensitive Hashing, MinHash

Алгоритм	Метрика подібності	Швидкість	Точність	Підходить для великих даних?	Основні використання
Shingling	Jaccard Similarity	Низька	Висока	Ні	Пошук дублікатів, плагіат
VSM	Косинусна подібність	Середня	Висока	Так	Пошукові системи, NLP
LSH	Jaccard / Косинус	Висока	Середня	Так	Пошук схожих документів
MinHash	Jaccard Similarity	Висока	Середня	Так	Виявлення дублікатів

12.1.4. Сервіси для перевірки тексту на плагіат

Сервіс для перевірки на плагіат — це програмне забезпечення або онлайн-інструмент, що дозволяє перевіряти тексти на наявність схожих або запозичених фрагментів з інших джерел. Такі сервіси порівнюють наданий текст з великою кількістю баз даних, публікацій, статей або Інтернет-ресурсів, щоб виявити можливі випадки плагіату.

Основні функції таких сервісів включають:

- Аналіз тексту на схожість з іншими публікаціями;
- Визначення конкретних частин тексту, які є запозиченими;
- Посилання на джерела, з яких був взятий матеріал (якщо це доступно);
- Інтерфейс для завантаження файлів або вставки тексту для перевірки.

Основні сервіси для пошуку плагіату в тексті

1. *Використання інструментів для перевірки плагіату.* Багато інструментів, доступних онлайн, перевіряють текст на схожість із іншими джерелами:

- *Turnitin* – це одна з найпопулярніших систем перевірки академічних текстів на плагіат. Вона використовується університетами, школами та науковими установами для аналізу унікальності студентських робіт, дисертацій і наукових публікацій.

Основні функції Turnitin:

1. *Перевірка на плагіат* – порівняння тексту з базами даних (академічні роботи, веб-сторінки, журнали тощо);
2. *Генерація звіту про унікальність* – відсоток збігів з іншими джерелами;
3. *Аналіз стилю письма* – виявлення аномалій у тексті, які можуть свідчити про використання *AI*;
4. *Збереження робіт у базі даних* – щоб уникнути повторного використання тієї ж роботи;

5. *Функція Feedback Studio* – дозволяє викладачам залишати коментарі та оцінки в межах платформи.

Схематичне зображення звіту Turnitin представлено на рис. 12.1. Воно відображає основні елементи: підсвічені збіги в тексті, загальний відсоток подібності та список джерел.

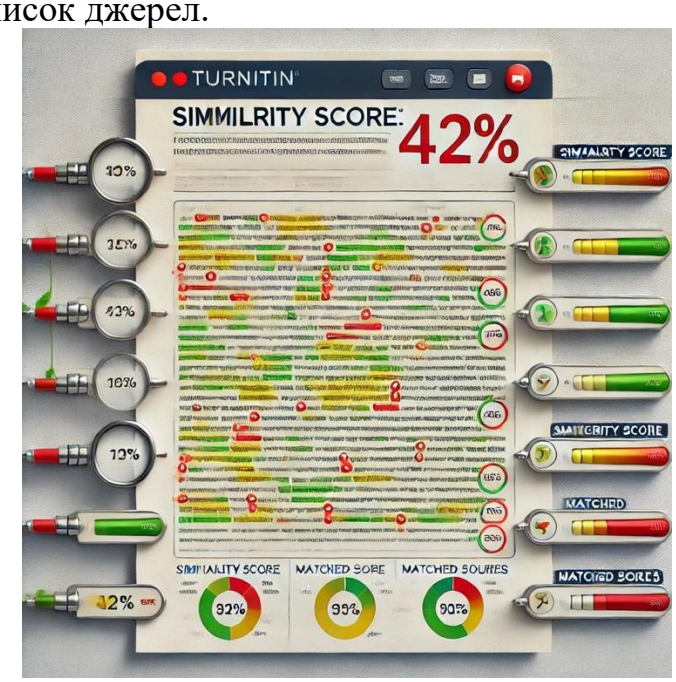


Рис. 12.1. Схематичне зображення звіту Turnitin

- *Copyscape* – це онлайн-сервіс для перевірки текстів на унікальність, який допомагає виявити плагіат або копії контенту в Інтернеті.

Основні функції Copyscape:

1. *Перевірка веб-сторінок.* Ввести URL, і сервіс знайде схожі тексти в Інтернеті;
2. *Перевірка документів та тексту.* Вставте текст або завантажте файл для пошуку збігів;
3. *Copyscape Premium.* Глибший пошук (детальніший аналіз, перевірка PDF, TXT, DOC);
4. *Copysentry.* Автоматичний моніторинг веб-сторінок на предмет несанкціонованого копіювання;
5. *API для інтеграції.* Дозволяє автоматизувати перевірку контенту.

Зразок звіту в Copyscape наведено на рис. 12.2 та рис. 12.3.

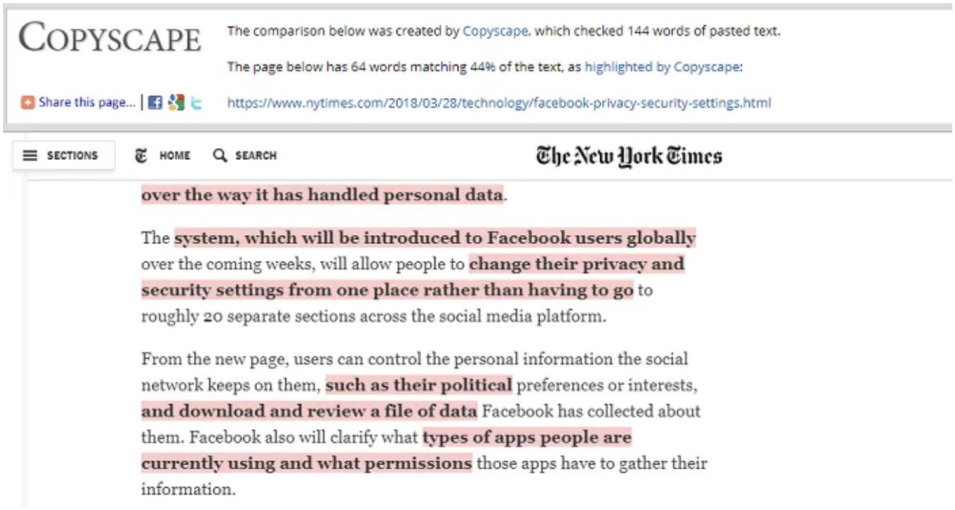


Рис. 12.2. Детальний звіт в Copyscape (частина 1)

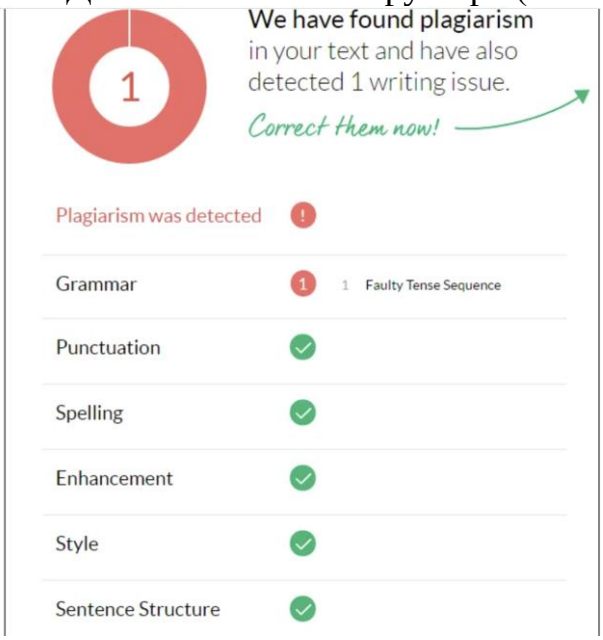


Рис. 12.3. Детальний звіт в Copyscape (частина 2)

- *Grammarly* (*Premium*) — це онлайн-сервіс для перевірки граматики, орфографії, стилю та плагіату в текстах. Він використовується письменниками, студентами, професіоналами та всіма, хто хоче покращити якість написаного тексту.

Основні функції Grammarly:

1. *Перевірка граматики та орфографії* – виявлення помилок у тексті;
2. *Редагування стилю та тону* – рекомендації щодо покращення читабельності;
3. *Перевірка на плагіат* (лише у платній версії) – пошук збігів з онлайн-джерелами;
4. *Аналіз читабельності* – оцінка складності тексту, довжини речень тощо;
5. *Адаптація під цільову аудиторію* – вибір формального чи неформального стилю;

6. *Інтеграція з різними платформами* – доступний як плагін для браузерів, Word, Google Docs, а також як мобільний додаток.

Зразок звіту в Grammarly наведено на рис. 12.4 та рис. 12.5.

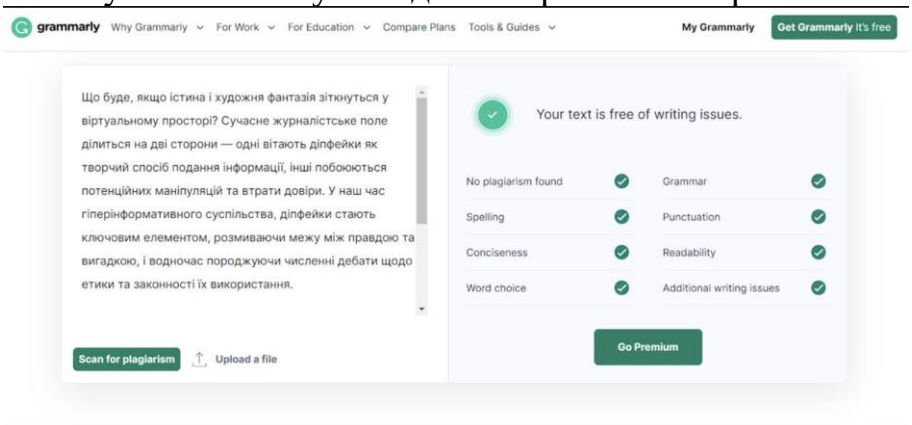


Рис. 12.4. Детальний звіт в Grammarly (частина 1)

Рис. 12.5. Детальний звіт в Grammarly (частина 2)

- *Unicheck* — це онлайн-сервіс для перевірки текстів на плагіат. Його використовують університети, школи, науковці та бізнес для виявлення збігів у документах.

Основні функції Unicheck:

1. *Перевірка на плагіат* – аналіз тексту на унікальність порівняно з Інтернет-джерелами, базами університетів і власними архівами;
2. *Швидка обробка* – система перевіряє мільйони джерел за лічені секунди;
4. *Детальний звіт про збіги* – текст підсвічується різними кольорами відповідно до знайдених джерел;
5. *Інтеграція з LMS* (Moodle, Google Classroom, Canvas, Blackboard) – зручно для навчальних закладів;
6. *Визначення перефразування та прихованого плагіату* – знаходить замасковані запозичення;

7. *Перевірка стилю та граматики* (не основна функція, але є базовий аналіз).

Зразок звіту в Unicheck наведено на рис. 12.6 та рис. 12.7.

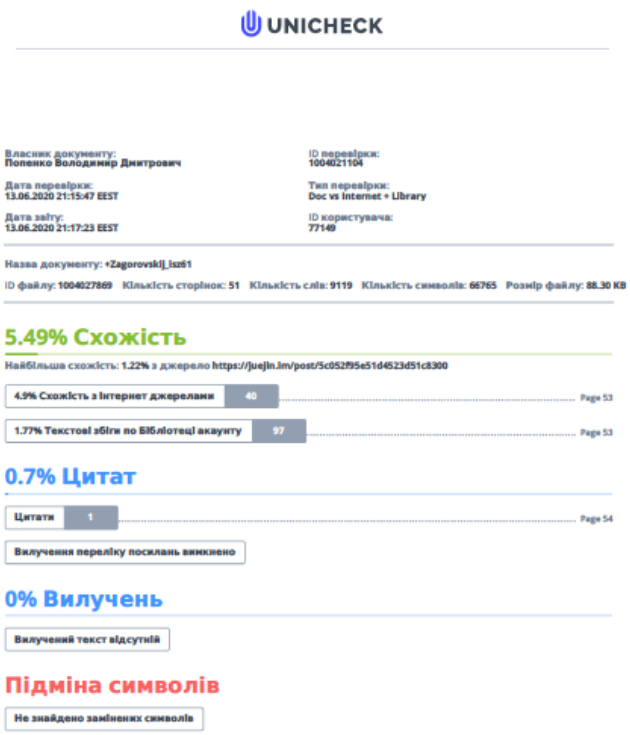


Рис. 12.6. Детальний звіт в Unicheck (частина 1)

Рис. 12.7. Детальний звіт в Unicheck (частина 2)

Порівняння сервісів пошуку плагіату в текстах

Порівняння розглянутих сервісів наведено в табл. 12.3.

Таблиця 12.2. Порівняння Turnitin, Grammarly, Copyscape, Unichek

Сервіс	Цільова аудиторія	Особливості	Плюси	Мінуси
Turnitin	Академії, студенти, викладачі	Велика база академічних робіт, інтеграція з LMS, детальні звіти	Найкраще для академічних робіт, точний	Висока ціна, лише для академії
Grammarly	Студенти, блогери, бізнесмени	Перевірка граматики, стилістики та плагіату, інтеграція в браузер	Простий у використанні, доступний для всіх	Не такий точний, як Turnitin
Copyscape	Веб-розробники, контент-менеджери	Перевірка плагіату на веб-сайтах, індексація Інтернету	Дешевий, спеціалізований на веб-контенті	Не підходить для академічних робіт
Unicheck	Академії, студенти, викладачі	Інтеграція з навчальними платформами, точні звіти	Простий, точний, доступний для навчальних установ	Платний, не так популярний як Turnitin

2. *Використання бібліотек Python.* Для виявлення схожих текстів можна використовувати деякі бібліотеки Python, такі як $difflib$, щоб порівнювати тексти або знаходити схожість між ними.

3. *Порівняння з базами даних.* Існує кілька відкритих баз даних, таких як Google Scholar або архіви наукових публікацій, з якими можна порівнювати тексти.

Google Scholar — це безкоштовний пошуковий сервіс, що спеціалізується на наукових публікаціях, академічних статтях, книгах, дисертаціях, конференційних матеріалах та інших наукових джерелах. Цей сервіс дозволяє користувачам знаходити різноманітні академічні ресурси в Інтернеті, перевіряти цитування та зручніше орієнтуватися в науковому контексті (рис. 12.8).

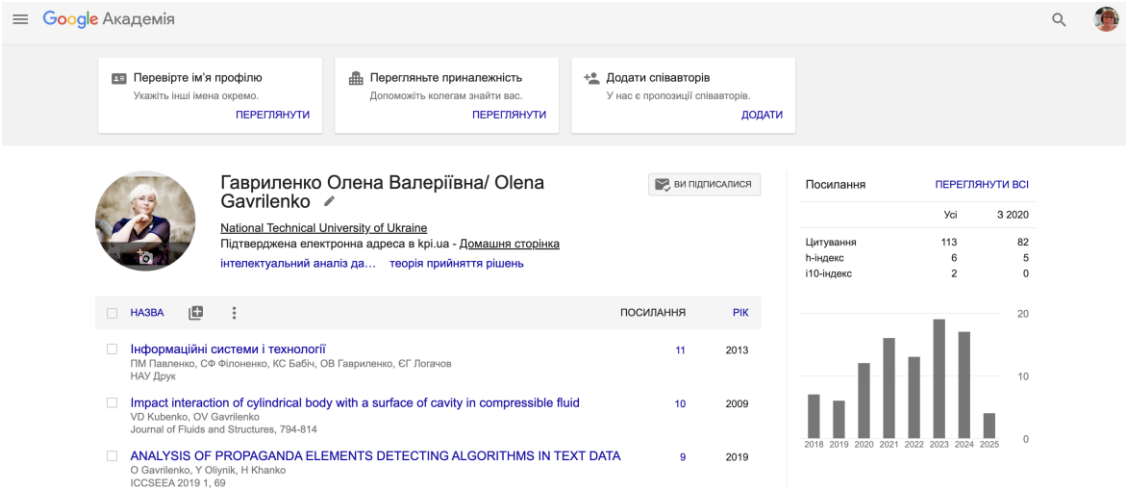


Рис. 12.8. Зразок профілю в Google Scholar

Основні можливості Google Scholar:

1. *Пошук наукових статей*: Ви можете знайти статті, книги, тези та інші наукові роботи по різних темах;
2. *Цитування*: Google Scholar показує, скільки разів була процитована певна публікація, що дозволяє оцінити її наукову значущість;
3. *Профілі авторів*: Автори можуть створювати профілі, щоб управляти своїми публікаціями та цитуваннями;
4. *Слідкування за новими публікаціями*: Ви можете підписатися на певні теми, авторів або журнали, щоб отримувати сповіщення про нові публікації;
5. *Пошук в рецензованих журналах*: Інтерфейс дозволяє фільтрувати результати по рецензованих джерелах, що є важливим для науковців;
6. *Безкоштовний доступ*: Більшість публікацій доступні безкоштовно, або через відкриті репозиторії.

12.2. Визначення пропаганди в тексті

Пропаганда — це систематичне поширення інформації, ідей або поглядів з метою впливу на суспільну думку, поведінку або емоції людей. Вона може використовуватися для підтримки певної ідеології, політичної позиції, релігії або соціального руху [49].

12.2.1. Постановка задачі

Вхідні дані:

1. *Текстові документи* (новини, статті, соцмережі, виступи політиків тощо);
2. *Анотації експертів* (якщо є позначені дані про наявність пропагандистських технік);
3. *Лінгвістичні та семантичні ознаки* (емоційно забарвлені слова, гіперболи, узагальнення, заклики до дії).

Вихідні дані:

1. *Класифікація тексту*:
- Пропагандистський / Нейтральний;
- Тип пропаганди (емоційна маніпуляція, спотворення фактів, тощо);

2. *Оцінка рівня пропаганди* (відсотковий показник ймовірності маніпуляції);
3. *Виділення конкретних фраз, що містять пропагандистські техніки*.

Формалізація задачі

Нехай є множина документів $P = \{P_1, P_2, \dots, P_n\}$.

Потрібно знайти функцію (12.3):

$$f(P_i) \to \{0,1\}, \quad (12.3)$$

яка визначає, чи містить документ P_i $(i = \overline{1,n})$ пропаганду (1 — так, 0 — ні). Або багатокласову функцію (12.4):

$$f(P_i) \to \{\text{нейтральний}, \text{пропаганда} - \text{страху}, \text{пропаганда} - \text{впливу}, \text{тощо}\}. \quad (12.4)$$

Критерії оцінки результатів:

- *Точність класифікації* $(Precision, Recall, F_1)$;
- *Інтерпретованість* – можливість пояснити, чому текст віднесено до пропаганди;
- *Масштабованість* – застосування до великих масивів тексту.

12.2.1. Алгоритм визначення пропаганди в публікації

12.2.1.1. Формування датасету з публікацій

На вхід подається набір публікацій $P_1, \dots, P_l$, які відповідатимуть наступним критеріям:

- Публікації відбираються з різноманітних новосних видань світу чи постів в соціальних мережах;
- Публікації мають відноситися до різної тематики;
- Тексти публікацій повинні бути написані різними мовами;
- Публікації повинні містити статистичні дані (орієнтовний перелік наведено в табл. 12.4).

На виході отримується множина публікацій $P = \{P_1, \dots, P_l\}$ зі статистичними даними щодо них (табл. 12.4) [49].

Таблиця 12.4. Зразок вхідних статистичних даних

	Назва публікації	Час публікації	Першоджерело публікації	Кількість перепостів в публікації	Джерела перепостів	Кількість слів у публікації	Кількість речень у публікації	Кількість складів у публікації
1	P_1	t_1	L_1	m_1	$Q_1^1; Q_2^1; \dots; Q_m^1$	a_1	b_1	c_1
2	P_2	t_2	L_2	m_2	$Q_1^1; Q_2^1; \dots; Q_m^1$	a_2	b_2	c_2
..	…	…	…	…	…	…	…	…
l	P_l	t_l	L_l	m_l	$Q_1^1; Q_2^1; \dots; Q_m^1$	a_l	b_l	c_l

У табл. 12.4. у стовпці 2 наведено P_j – назва публікацій, у стовпці 3 t_j – час, коли було зроблено дану публікацію, у стовпці 4 L_j – назви її першоджерела, у стовпці 5 m_j – кількості її перепостів в інших новосних джерелах чи соціальних мережах, у стовпці 6 $Q_1^j; Q_2^j; \dots; Q_{m_j}^j$ – перелік джерел, в яких було перепощено публікацію, у стовпці 7 a_j – кількість слів у публікації, у стовпці 8 b_j – кількість речень у публікації, у стовпці 9 c_j – кількість складів у публікації $(j = 1,2, \dots, l)$.

Сформований таким чином датасет дає змогу охопити більшу кількість різноманітних інформаційних джерел світу, що, в свою чергу, зробить більш якісним експеримент з розпізнання елементів пропаганди. Зазначені статистичні дані будуть враховані при обчислень деяких значень показників відповідності ознакам пропаганди.

12.2.1.2. Визначення ознак, які є характерними при визначенні пропаганди в публікаціях

На вхід подаються наступні ознаки пропаганди:

x_1 ={спроби маніпулювати аудиторією};
x_2 ={спрямованість публікації викликати емоції};
x_3 ={часті повторення певної думки в публікації};
x_4 ={часті перепости публікації};
x_5 ={простота читання тексту публікації};
x_6 ={високий рівень пропагандиських публікацій в першоджерелі};
x_7 ={приналежність публікації до певної тематики, яка є особливо підвладною пропаганді};
x_8 ={дана публікація може мати вплив на певну подію};
x_9 ={наявність осіб, яким вигідне розповсюдження даної публікації};
x_{10} ={наявність клікабельних виразів в заголовках чи тексті публікації};

На виході отримується множина ознак $X = (x_1, \dots, x_{10})$.

12.2.1.3. Побудова багатофакторної моделі для обчислення рівня пропаганди в публікаціях

Процес побудови багатофакторної моделі для обчислення рівня пропаганди в публікаціях, згідно методу лінійної згортки (рис. 12.9).

Кроки процесу побудови моделі

Крок 0: Передобробка тексту публікації.
Крок 1: Обчислення числових показників моделі.
Крок 2: Обчислення коефіцієнтів важливості кожного показника.
Крок 3: Обчислення функції цінності.

Рис. 12.9. Процес побудови багатофакторної моделі

КРОК 0

На вхід подаються: множина публікацій $P = \{P_1, \dots, P_l\}$.

На виході отримуються множини $A_1, A_2, \dots, A_l$ – множини вживаних слів в публікаціях $P_1, \dots, P_l$ відповідно.

Для формування кожної з множини A_j $(j = 1, \dots, l)$ рекомендується провести передобробку тексту публікації за допомогою процесів лематизації та стемінгу з метою зменшення потужності даної множини за рахунок відкидання спільнокореневих слів та допоміжних частин мови.

КРОК 1

На вхід подаються: множина публікацій $P = \{P_1, \dots, P_l\}$, множина ознак $X = (x_1, \dots, x_{10})$ та статистичні дані про публікації (табл. 12.4).

Необхідно розрахувати рівні пропаганди за кожним з наведених факторів.

На виході отримується множина $K^j = \{K_1^j, \dots, K_{10}^j\}$, де $K_i^j, i = 1, \dots, 10;\ j = 1, \dots, l$ – значення метрик, які показують рівень пропаганди в публікації P_j згідно ознаки x_i.

1. *Обчислення метрики* K_1^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$, в якій йдеться про деякі факти (події), які необхідно попередньо виділити з публікації. Множину фактів F^j представимо у вигляді:

$$F^j = F_1^j \cup F_2^j,\ F_1^j \cap F_2^j = \emptyset,$$

де F_1^j, F_2^j – підмножини фактів, відповідно підтверджених та непідтвердженими достовірними джерелами, статистичними даними та нормативними актами.

Метрика K_1^j публікації P_j вказує на те, яка частка фактів викладених в публікації лишаються без підтвердження з достовірних джерел, статистичних даних чи нормативних актів. Обчислюється за формулою (12.5):

$$K_1^j = \frac{\left|F_2^j\right|}{\left|F^j\right|}, \qquad (12.5)$$

де:

$\left|F_2^j\right|$ – кількість непідтверджених фактів у публікації P_j;

$\left|F^j\right| = \left|F_1^j\right| + \left|F_2^j\right|$ – загальна кількість фактів у публікації P_j.

Очевидно, що $0 \le K_1^j \le 1$ і чим ближче його значення до одиниці, тим більше непідтверджених фактів має дана публікація. Отже, за ознакою x_1 її можна рекомендувати як пропагандистську.

2. *Обчислення метрики* K_2^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$. Як відомо, будь-який текст може викликати в людини три типи емоцій: позитивні, негативні та нейтральні. Для їх визначення у текстах створено спеціальні словники.

Метрика K_2^j вказує на рівень емоційного забарвлення публікації, тобто позитивних або негативних емоцій. Нейтральні емоції не слід розглядати, оскільки пропагандистські публікації не повинні лишати байдужим аудиторію. За аналогією з лекцією 11, обчислюється за формулою (12.6):

$$K_2^j = \frac{K_{2p}^j + K_{2n}^j}{2}, K_{2p}^j = \frac{|A_j \cap B|}{|A_j|}, K_{2n}^j = \frac{|A_j \cap C|}{|A_j|}, \quad (12.6)$$

де:

A_j – множина вживаних слів в публікації P_j, а $|A_j| = a_j$ (табл. 12.4);

B – множина слів у словнику для визначення позитивних емоцій;

C – множина слів у словнику для визначення негативних емоцій;

K_{2p}^j – рівень позитивного забарвлення публікації;

K_{2n}^j – рівень негативного забарвлення публікації.

Очевидно, що $0 \le K_2^j \le 1$ і чим ближче його значення до одиниці, тим вищій рівень емоційного забарвлення має дана публікація. Отже, за ознакою x_2 її можна рекомендувати як пропагандистську.

3. *Обчислення метрики* K_3^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$. Нехай A_j – множина вживаних слів. Ці слова формують певну думку в даній публікації, має бути попередньо визначена.

Метрика K_3^j вказує на частоту повторів цієї думки в публікації (12.7).

$$K_3^j = \frac{m_{A_j}}{n_j}, \quad (12.7)$$

де:

m_{A_j} – кількість повторів слів, що входять в множину A_j;

n_j – загальна кількість слів в публікації P_j.

Очевидно, що $0 \le K_3^j \le 1$ і чим ближче його значення до одиниці, тим частіше повторюється певна думка в даній публікації. Отже, за ознакою x_3 її можна рекомендувати як пропагандистську.

4. *Обчислення метрики* K_4^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$ та множина $Q_j = \left\{Q_1^j; Q_2^j; \dots; Q_{m_j}^j\right\}$ джерел, в яких зроблено перепости публікації P_j (табл. 12.4).

Метрика K_4^j вказує на частоту перепостів даної публікації (12.8).

$$K_4^j = \frac{m_j}{n_{Q_j}}, \quad (12.8)$$

де:

m_j – кількість перепостів публікації P_j (табл.12.4);

n_{Q_j} – середня кількість перепостів публікацій в множині Q_j – має бути визначена додатково.

Очевидно, що $0 \le K_4^j \le 1$ і чим ближче його значення до одиниці, тим частіше перепощується дана публікація. Отже, за ознакою x_4 її можна рекомендувати як пропагандистську.

5. *Обчислення метрики* K_5^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$.

Метрика K_5^j вказує на легкість читання тексту даної публікації. Обчислюється за формулою (12.9):

$$K_5^j = \left(206{,}835 - 1{,}015 \cdot \frac{a_j}{b_j} - 84{,}6 \cdot \frac{c_j}{a_j}\right) \cdot 0{,}01, \qquad (12.9)$$

де:

кількість слів – a_j;

кількість речень – b_j;

кількість складів – c_j (табл. 12.4).

Метрика K_5^j називається *індексом читабельності Флеша*.

Значення даної метрики можна інтерпретувати, як показано в табл. 12.5

Таблиця 12.5. Інтерпретація значень індексу читабельності Флеша

Оцінка	Шкільний рівень	Примітки
0,9–1,0	5 клас	Дуже легко читається. Легко зрозумілий середньому 11-річному учневі.
0,9–0,8	6 клас	Легко читається. Розмовна мова для споживачів.
0,8–0,7	7 клас	Досить легко читається.
0,7–0,6	8 і 9 клас	Звичайна мова. Легко сприймається учнями 13-15 років.
0,6–0,5	10-12 клас	Досить важко читати.
0,5–0,3	Коледж	Важко читати.
0,3–0,1	Випускник технікуму	Дуже важко читати. Найкраще розуміють випускники університетів.
0,1–0,0	Професійний	Надзвичайно важко читати. Найкраще розуміють випускники університетів.

Очевидно, що $0 \le K_5^j \le 1$ і чим ближче його значення до одиниці, тим більш простою для читання є дана публікація. Отже, за ознакою x_5 її можна рекомендувати як пропагандистську.

6. *Обчислення метрики* K_6^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$ та її першоджерело L_j.

Метрика K_6^j вказує на рівень недовіри читачів до джерела L_j. Обчислюється за формулою (12.10):

$$K_6^j = \frac{m_{L_j}}{n_{L_j}}, \quad (12.10)$$

де:

m_{L_j} – кількість публікацій пропагандиського характеру в інформаційному джерелі L_j;

n_{L_j} – загальна кількість публікацій джерела L_j.

Очевидно, що $0 \leq K_6^j \leq 1$ і чим ближче його значення до одиниці, тим вищою є ймовірність того, що дана публікація носитиме пропагандистський характер. Отже, за ознакою x_6 її можна рекомендувати як пропагандистську.

Слід зазначити, що з джерел L_j, публікації яких було проаналізовано на наявність пропаганди, необхідно скласти ранжований список L щодо рівня пропаганди, який буде весь час оновлюватися. Цей список міститиме всю необхідну статистичну інформацію для подальших досліджень (табл. 12.6).

Таблиця 12.6. Зразок списку L

№	Назва інформаційного джерела	Кількість пропагандистських публікацій m_{L_j}	Загальна кількість публікацій n_{L_j}
1	L_1	m_{L_1}	n_{L_1}
2	L_2	m_{L_2}	n_{L_2}
…	…	…	…
l	L_l	m_{L_l}	n_{L_l}

Окрім того, в разі відсутності такого списку можна скористатися рейтинговими списками ЗМІ, складеними журналістами, зокрема видання Forbes. Це вирішить проблему «холодного старту» на початкових етапах дослідження. Але в подальшому слід використовувати список L, що унеможливить суб'єктивний вплив людини (журналіста) на прийняття рішення щодо рівня довіри тому чи іншому інформаційному джерелу.

7. Обчислення метрики K_7^j. Розглянемо:

- Деяку публікацію P_j, $j = 1, \dots, l$;
- Множину вживаних слів у публікації A_j;
- Множину тем $S = (s_1; s_2; \dots; s_r)$, в яких найчастіше зустрічаються пропагандистські публікації;
- Словники характерних слів для даних тем, $T_1; T_2; \dots; T_r$.

Теми і відповідні словники мають бути сформовані попередньо.

Метрика K_7^j вказує на те, чи належить публікація P_j до якоїсь із тем з множини S.

Основні етапи обчислення метрика K_7^j

Етап 1. Обчислюються коефіцієнти подібності Жаккара між множиною A_j та кожним з словників $T_k, k = 1,2, \dots, r$ (12.11).

$$J(A_j, T_k) = \frac{|A_j \cap T_k|}{|A_j \cup T_k|}; \quad (12.11)$$

Етап 2. Обирається максимальне значення коефіцієнту Жаккара:

$$J_{max}(A_j, T_k) = \max_{k=1,2,\dots,r} J(A_j, T_k), j = 1, \dots, l$$

та встановлюється відповідність цього коефіцієнту темі $s_k \in S$, якій відповідає дане значення;

Етап 3. Метрика K_7^j дорівнює (12.12):

$$K_7^j = J_{max}(A_j, T_k). \quad (12.12)$$

Очевидно, що $0 \le K_7^j \le 1$ і чим ближче його значення до одиниці, тим більш близькою до тем, найбільш підвладних пропаганді, є дана публікація. Отже, за ознакою x_7 її можна вважати пропагандистською.

8. *Обчислення метрики* K_8^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$ та множину вживаних слів у публікації A_j.

Метрика K_8^j вказує на те, чи матиме вплив публікація P_j на настання деякої події D.

Під впливом публікації P_j на подію D розуміються прямі або опосередковані згадки про неї в тексті публікації.

Таким чином, з множини A_j необхідно виділити підмножину $D_j \subset A_j$ слів, які описують подію D. Тоді справедлива формула (12.13):

$$K_8^j = \begin{cases} 1, & \text{якщо } D_j \neq \{\emptyset\}; \\ 0, & \text{якщо } D_j = \{\emptyset\}. \end{cases} \quad (12.13)$$

Очевидно, що якщо для публікації P_j значення $K_8^j = 1$, то за ознакою x_8 її можна рекомендувати як пропагандистську.

9. *Обчислення метрики* K_9^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$ та множину вживаних слів у публікації A_j.

Метрика K_9^j вказує на те, чи існують особи (група осіб M), які зацікавлені у публікація P_j.

Під зацікавленістю особи чи групи осіб M у публікації P_j розуміються прямі або опосередковані згадки про неї (них) в тексті публікації.

Таким чином, з множини A_j необхідно виділити підмножину $M_j \subset A_j$ слів, які описують осіб M, що згадуються в публікації P_j. Тобто справедлива рівність (12.14):

$$K_9^j = \begin{cases} 1, & \text{якщо } M_j \neq \{\emptyset\}; \\ 0, & \text{якщо } M_j = \{\emptyset\}. \end{cases} \quad (12.14)$$

Очевидно, що якщо для публікації P_j значення $K_9^j = 1$, то за ознакою x_9 її можна рекомендувати як пропагандистську.

10. *Обчислення метрики* K_{10}^j. Розглянемо деяку публікацію $P_j, j = 1, \dots, l$; множину вживаних слів у публікації A_j та словник клікабельних слів N, який також необхідно створювати попередньо.

Метрика K_{10}^j вказує на те, чи використовує публікація P_j слова з множини N. Тоді справедлива формула (12.15):

$$K_{10}^j = \begin{cases} 1, \text{якщо } A_j \cap N \neq \emptyset; \\ 0, \text{якщо } A_j \cap N = \emptyset. \end{cases} \quad (12.15)$$

Очевидно, що якщо для публікації P_j значення $K_{10}^j = 1$, то за ознакою x_{10} її можна рекомендувати як пропагандистську.

Слід зазначити, що розглянуті метрики можна інтерпретувати як ймовірності того, що публікація P_j буде пропагандою за ознакою x_i.

Значення K_i^j розраховувалися статистичними методами та за допомогою створення різноманітних бібліотек, але також можуть бути розраховані за допомогою засобів машинного навчання (див. лекцію 11).

КРОК 2

На вхід подаються: показники $K_i^j, i = 1, \dots, 10;\ j = 1, \dots, l$ (формули (12.5)-(12.15)).

Необхідно для кожного критерія розрахувати коефіцієнти їх важливості для обчислення функції цінності.

На виході коефіцієнти ω_i.

Основні етапи обчислення вагових коефіцієнтів ω_i

Етап 1. З показників K_i^j сформувати статистичні вибірки з аналогічною назвою K_i^j.

Етап 2. Вибрати порогове значення, при перевищені якого текст публікацію можна вважати пропагандиською.

За аналогією зі шкалою Чеддока, яка визначає силу кореляційного зв'язку між двома випадковими величинами, використаємо наступне шкалування (табл. 12.7):

Таблиця 12.7. Шкала виявлення рівня пропаганди

Значення коефіцієнта K_i^j	$0,0 - 0,1$	$0,1 - 0,3$	$0,3 - 0,5$	$0,5 - 0,7$	$0,7 - 0,9$	$0,9 - 1,0$
Характеристика рівня пропаганди	відсутня пропаганда	слабкий рівень пропаганди	помітний рівень пропаганди	помірний рівень пропаганди	високий рівень пропаганди	вельми високий рівень пропаганди

Необхідно враховувати всі рівні пропаганди, починаючи з помітного. Таким чином, в якості порогового значення встановлюється $\bar{K}_\iota = 0{,}3$.

Дане порогове значення було введене з метою проведення подальших статистичних розрахунків та зручності подальшого порівняння отриманих результатів.

Етап 3. Якщо $K_i^j \geq \bar{K}_\iota$, то вважаємо дану публікацію P_j пропагандистською за ознакою x_i. В противному разі – непропагандистською. Публікації ставиться у відповідність значення «1», якщо вона є пропагандою за цією ознакою і значення «0» – в противному випадку. Тобто, справедлива рівність (12.16):

$$P_j \to \widetilde{K_\iota^j} = \begin{cases} 1, \text{якщо } K_i^j \geq \bar{K}_\iota; \\ 0, \text{якщо } K_i^j < \bar{K}_\iota. \end{cases} \quad (12.16)$$

Перехід від кількісних значень K_i^j до булевих функцій $\widetilde{K_\iota^j}$ було зроблено з міркувань зручності подальшого порівняння отриманих результатів.

Етап 4. Обчислити відносну частоту появи пропагандистських публікацій за кожною з ознак x_i за формулою (12.17):

$$w_i = \frac{m_i}{n}, \quad (12.17)$$

де m_i – кількість пропагандистських публікацій за ознакою x_i; n – загальна кількість публікацій в датасеті.

Етап 5. Нормувати відносні частоти w_i за формулою (12.18):

$$\omega_i = \frac{w_i}{w_1 + w_2 + \cdots + w_8}. \quad (12.18)$$

КРОК 3

На вхід подаються: множина публікацій $P = \{P_1, \ldots, P_l\}$, показники $K_i^j, i = 1, \ldots, 10$; $j = 1, \ldots, l$ та коефіцієнти ω_i^j.

Необхідно для кожної публікації розрахувати значення функції цінності з точки зору наявності в ній ознак пропаганди.

На виході значення функції V_j.

Значення функції цінності V_j, згідно методу лінійної згортки, обчислюються за формулою (12.19):

$$V_j = \sum_{i=1}^{10} (\omega_i \cdot K_i^j). \quad (12.19)$$

Результати розрахунків можна представити у вигляді таблиці (табл. 12.8):

Таблиця 12.8. Розрахункова таблиця

	K_1^j	K_2^j	K_3^j	K_4^j	K_5^j	K_6^j	K_7^j	K_8^j	K_9^j	K_{10}^j	V_j
P_1	K_1^1	K_2^1	K_3^1	K_4^1	K_5^1	K_6^1	K_7^1	K_8^1	K_9^1	K_{10}^1	V_1
P_2	K_1^2	K_2^2	K_3^2	K_4^2	K_5^2	K_6^2	K_7^2	K_8^2	K_9^2	K_{10}^2	V_2
…	…	…	…	…	…	…	…	…	…	…	…
P_l	K_1^l	K_2^l	K_3^l	K_4^l	K_5^l	K_6^l	K_7^l	K_8^l	K_9^l	K_{10}^l	V_l

Сформуємо статистичну вибірку зі значень функції цінності (12.19):

$$V = (V_1, V_{2,}, \dots, V_l).$$

12.2.1.4. Формування рекомендації, щодо того, чи є публікація пропагандистською чи ні

На вхід подається набір публікацій $P = \{P_1, \dots, P_l\}$ та вибірка $V = \{V_1, V_{2,}, \dots, V_l\}$ (див. пункт 12.2.1.3).

На виході формуються рекомендації щодо того, які з публікацій $P = (P_1, \dots, P_l)$ є пропагандою.

Рекомендації формуються за наступним правилом:

- Якщо $V_j \geq \bar{V}$, $(j = 1, \dots, l)$, то публікація P_j рекомендується як пропагандистська;
- Якщо $V_j < \bar{V}$, $(j = 1, \dots, l)$, то публікація P_j не рекомендується як пропагандистська.

В даному правилі $\bar{V} = 0{,}3$ – порогове значення для вибірки V (за аналогією до етапу 2 пункту 12.2.1.3). Тобто справедлива формула (12.20):

$$P_j \to \widetilde{V_j} = \begin{cases} 1, \text{якщо } V_j \geq \bar{V}; \\ 0, \text{якщо } V_j < \bar{V}. \end{cases} \quad (12.20)$$

Тобто, публікації ставиться у відповідність значення «1», якщо вона є пропагандою і значення «0» – в противному випадку.

Релевантність наданих рекомендацій оцінюється за допомогою метрик $Recall$ та $Precision$.

12.3. Задача реферування текстів

Реферування текстів — це процес автоматичного або напівавтоматичного створення короткого змістовного викладу тексту (реферату), що містить основну інформацію вихідного документа [50].

Метою реферування є зменшення обсягу тексту при збереженні його ключових ідей, що дозволяє користувачеві швидко ознайомитися зі змістом документа без необхідності його повного прочитання.

12.3.1. Постановка задачі реферування текстів

На вхід подається текст T, який представлений у вигляді впорядкованої множини речень: (12.21)

$$T = \{s_1, s_2, \dots, s_n\}, \quad (12.21)$$

де:

s_i – окреме речення тексту;

n – загальна кількість речень.

На виході система повинна формувати реферат R, який є підмножиною T (12.22):

$$R \subseteq T, \quad (12.22)$$

та задовольняє такі умови:

- реферат R має містити меншу кількість речень, ніж оригінальний текст;
- речення реферату повинні відображати основний зміст тексту T;
- речення реферату повинні зберігати початковий порядок слідування з тексту T.

Основні кроки процесу формування реферату

Крок 1. Токенізація тексту на речення.
Крок 2. Виявлення ключових слів та фраз.
Крок 3. Нормалізація тексту шляхом приведення слів до початкової форми.
Крок 4. Для кожного речення обчислюється його вагова оцінка $W(s_i)$, яка визначає інформативність речення.
Крок 5. Оцінка $W(s_i)$ (ймовірності переходів між реченнями та важливості ключових слів речення).
Крок 6. Речення сортуються за значенням їхньої ваги $W(s_i)$ у спадному порядку.
Крок 7. Вибираються k речень із найбільшими значеннями $W(s_i)$ для формування реферату R.

Математична інтерпретація задачі реферування тексту

Математично задача автоматизованого реферування формулюється як задача вибору множини R, яка максимізує загальну інформативність, що визначається як (12.23):

$$\sum_{s_i \in R} W(s_i), \qquad (12.23)$$

де $W(s_i)$ – вагова оцінка речення s_i.

Вагова оцінка речення визначається формулою: (12.24):

$$W(s_i) = \alpha \cdot M(s_i) + \beta \cdot A(s_i), \qquad (12.24)$$

де:

α, β – вагові коефіцієнти, які регулюють вплив факторів;

$M(s_i)$ – оцінка зв'язності речення s_i у контексті тексту, отримана на основі матриці переходів;

$A(s_i)$ – оцінка важливості речення s_i з точки зору ключових слів.

Існуючі рішення систем для реферування текстів.

12.3.2. Існуючі рішення систем для реферування текстів

Системи автоматичного узагальнення текстів є важливим кроком у розумінні сучасних тенденцій і технологій в цій галузі. З огляду на стрімке зростання обсягу текстової інформації в різних галузях, таких як новини, наукові публікації, соціальні мережі та корпоративна документація, виникає гостра необхідність у розробці систем, здатних автоматично аналізувати та узагальнювати великі масиви текстових даних. Це дозволяє не лише економити час на обробку інформації, але й підвищувати якість прийняття рішень на основі стислих і релевантних даних.

На сьогоднішній день існує широкий спектр рішень для автоматичного узагальнення текстів, які базуються на різних підходах, таких як

статистичні методи, методи обробки природної мови (NLP), а також сучасні моделі на основі машинного навчання і глибокого навчання. Кожен з цих підходів має свої переваги і недоліки, залежно від специфіки застосування та типу текстових даних.

12.3.2.1. Система реферування текстів QuillBot

QuillBot є одним із провідних інструментів для автоматизації роботи з текстами, що базується на сучасних методах обробки природної мови (NLP) та глибокого навчання. Цей інструмент використовується для перефразування текстів, їх узагальнення, перевірки граматики та стилю, а також автоматизації цитування. QuillBot підтримує інтеграції з популярними платформами, такими як Google Docs та Microsoft Word, що дозволяє його безперебійно використовувати у процесі створення текстових документів.

Основні можливості:

- Перефразування тексту;
- Узагальнення текстів;
- Перевірка граматики та стилю;
- Генерація цитат.

На рис 12.10 можна побачити інтерфейс QuillBot.

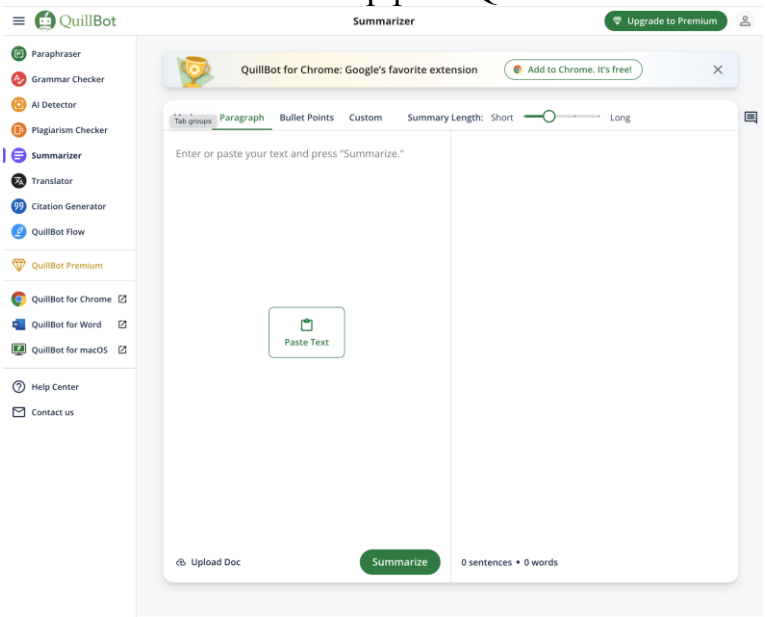


Рис. 12.10. Інтерфейс QuillBot

12.3.2.2. Система реферування текстів Scholarcy

Scholarcy – це інструмент на основі штучного інтелекту, призначений для автоматичного узагальнення та аналізу академічних текстів. Він розроблений спеціально для студентів, дослідників і науковців, щоб допомогти в обробці складних наукових публікацій, розділів книг, звітів та інших документів великого обсягу. Система автоматично виділяє ключову інформацію з текстів, полегшуючи сприйняття великих наукових робіт та економлячи час на їх аналіз.

Основні можливості:

- Автоматичне узагальнення наукових статей;
- Екстракція цитат та посилань;
- Підсвічування ключових термінів та концепцій;
- Інтеграція з іншими інструментами.

Технічно Scholarcy використовує методи обробки природної мови (NLP) і поєднує екстрактивні та абстрактивні методи узагальнення. Екстрактивні методи відбирають найважливіші речення з тексту, тоді як абстрактивні здатні переформулювати речення для стислого відображення основного змісту.

На рис 12.11 можна побачити інтерфейс Scholarcy.

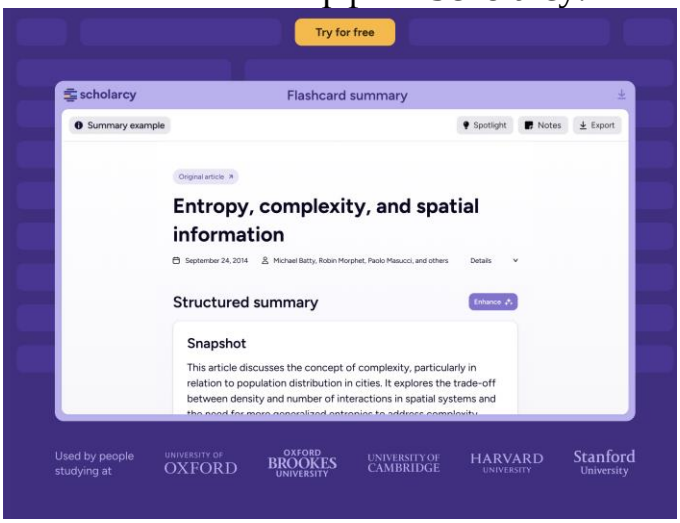


Рис. 12.11. Інтерфейс Scholarcy

12.3.2.3. Сервіс для скорочення тексту SMMRY

Простий і ефективний інструмент для автоматичного узагальнення текстів, що фокусується на екстракції найважливіших речень із введеного матеріалу. Основне завдання *SMMRY* – скорочення тексту до ключових моментів, забезпечуючи користувачів короткими і чіткими резюме. Інструмент використовує екстрактивний підхід до узагальнення, що означає вибір найбільш значущих речень без переформулювання оригінального змісту.

Основні можливості:

- Можливість регулювати довжину резюме, надаючи користувачам контроль над кількістю речень у підсумковому тексті;
- Інструмент підтримує вставку тексту вручну або через URL-адресу вебсторінки, інтерфейс системи можна побачити на рис 12.12;
- Гибкий набір способів надання тексту дає змогу легко створювати резюме для статей в Інтернеті;
- SMMRY пропонує *API* для інтеграції в сторонні додатки, що робить його корисним для розробників.

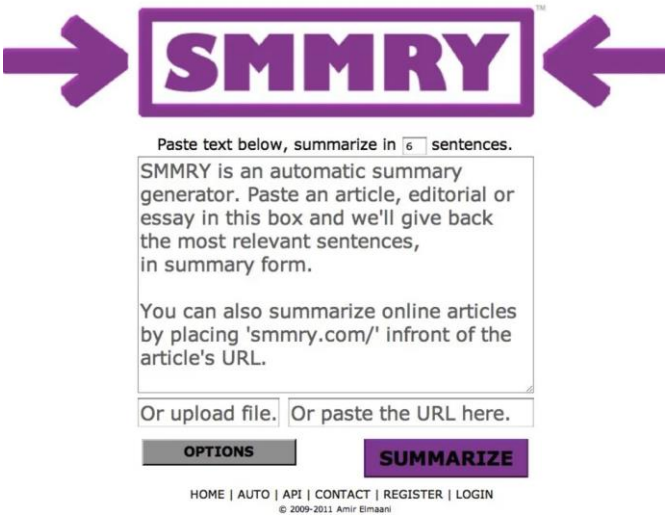


Рис. 12.12. Інтерфейс SMMRY

12.3.2.4. Інструмент для створення рефератів Resoomer

Інструмент Resoomer призначений для автоматичного узагальнення текстів, допомагаючи спрощувати складні документи шляхом виділення головних ідей та фактів. Він став популярним серед студентів, дослідників і журналістів, які потребують швидкого доступу до ключових тез у великих текстах.

Основні можливості:

- Вибір найважливіших речень чи фраз з оригінального тексту без зміни;
- Знаходження ключових слів та зв'язків між ними для визначення найважливіших частин тексту.їх структури;
- Визначення важливості певних слів і фраз на основі їх частоти та розташування в тексті;
- Аналізу синтаксичної структури речень, щоб відкидати незначні елементи, такі як перехідні фрази, пояснення або приклади, які не несуть основного смислового навантаження;
- Розуміння контексту тексту і вибору речень, які найбільше відповідають загальній темі документа.

Інтерфейс можна побачити на рис 12.13.

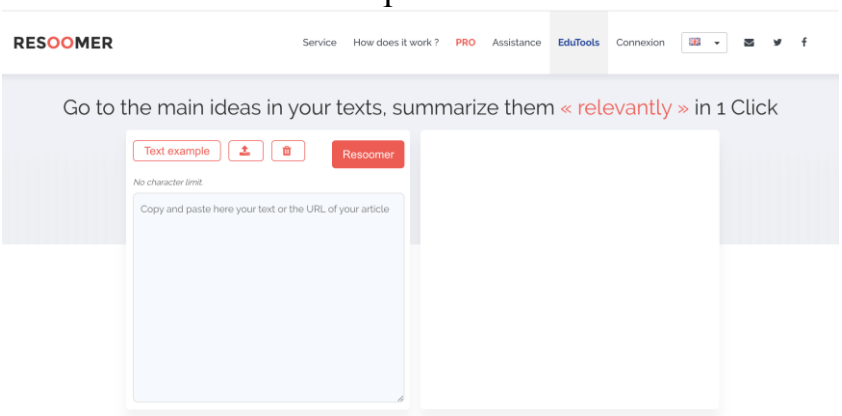


Рис.12.13. Інтерфейс Resoomer

12.3.2.5. Порівняння розглянутих систем

Порівняння систем QuillBot, Scholarcy, SMMRY та Resoomer наведено в табл. 12.9.

Таблиця 12.9. Переваги та недоліки розглянутих систем

Сервіс	Переваги	Недоліки	Проблеми з українською мовою
QuillBot	Універсальність, простота використання, доступ до широких функцій	Можливі втрати семантичних нюансів і спотворення змісту, потреба в перевірці результатів	Алгоритми QuillBot не завжди коректно обробляють українські тексти, що призводить до помилок у синтаксисі, стилі та втрати сенсу
Scholarcy	Оптимізація для наукових публікацій, зручність роботи з великими обсягами текстів, підтримка різних форматів	Обмеження в обробці надто великих документів, можливість втрати важливих деталей, необхідність перевірки результатів	Алгоритми недостатньо адаптовані до української академічної термінології, що ускладнює точне виділення ключових елементів
SMMRY	Швидкість, простота використання, можливість інтеграції через API	Відсутність узагальнень у повному сенсі, проблеми зі зв’язністю тексту, обмежена підтримка форматів документів	Екстрактивний підхід не завжди ефективно працює з українськими текстами, що призводить до втрати важливих контекстів і деталей
Resoomer	Висока швидкість обробки, ефективність для наукових та історичних текстів, зручний інтерфейс	Текст резюме може бути недостатньо зв’язним, можливість втрати важливих деталей, особливо у складних документах	Алгоритми не повністю враховують синтаксичну специфіку української мови, що може призводити до втрати змісту і когерентності тексту

12.3.3. Існуючі алгоритми реферування текстів

12.3.3.1. Абстрактивні алгоритми

Абстрактивні алгоритми є одним із найпросунутіших підходів у автоматичному реферуванні текстів, що відрізняються від екстрактивних методів своєю здатністю не лише вилучати готові фрагменти з тексту, але й створювати нові речення. Це дозволяє таким алгоритмам генерувати резюме, яке виглядає більш природно і зв'язно, оскільки текст узагальнюється на основі розуміння його змісту, а не просто шляхом механічного відбору речень.

Принцип роботи абстрактивних алгоритмів полягає в застосуванні глибоких нейронних мереж та сучасних моделей обробки природної мови, таких як архітектури трансформерів, зокрема, моделі на зразок BERT, GPT або T5. Ці моделі аналізують текст як єдину структуру, враховуючи

семантичні зв’язки між словами, реченнями та навіть абзацами. Такий підхід дозволяє алгоритмам краще розуміти контекст і генерувати нові фрази, що чітко відображають основну думку тексту, забезпечуючи зв'язність і логічність узагальнень.

Переваги:

- *Здатність до глибшого аналізу тексту.* Дозволяє створювати резюме, яке не тільки стисло передає зміст, але й робить це у природній мовній формі, що наближається до людського реферування;
- *Здатність генерувати зв'язні та логічно послідовні тексти.* Оскільки такі алгоритми не обмежуються копіюванням речень з оригіналу, вони можуть синтезувати нові, більш стисливі формулювання, що відображають головні ідеї документа;
- *Забезпечення високої якості узагальнення*, особливо у випадках, коли важливо зберегти логіку та послідовність подання інформації;
- *Гнучкість*. Вони можуть переформулювати текст, виділяючи ключові ідеї навіть у випадках, коли екстрактивні методи не дають чіткої інформації через невідповідність фрагментів оригіналу.

Недоліки:

- *Вони вимагають значних обчислювальних ресурсів для навчання і роботи*, оскільки їхня складна архітектура потребує потужних процесорів або графічних процесорів, а також великих обсягів оперативної пам’яті. Це робить абстрактивні методи більш затратними в порівнянні з екстрактивними підходами, які є значно простішими і швидшими в реалізації;
- *Навчання таких моделей вимагає великих наборів даних*, які повинні містити достатньо прикладів різних текстових структур для досягнення високої точності;
- *Складність їх реалізації.* Розробка таких моделей вимагає значних зусиль з боку інженерів і дослідників, оскільки потрібно не тільки правильно налаштувати модель, але й забезпечити адекватне навчання на відповідних даних;
- Навіть *після тренування абстрактивні алгоритми можуть генерувати текст із помилками*, особливо коли стикаються зі складними або неоднозначними формулюваннями в оригінальному тексті. Це може призвести до того, що резюме буде містити семантичні неточності або невірно передавати основну ідею.

Основні методи абстрактивного реферування

1. *Нейронні мережі на основі Seq2Seq (Sequence-to-Sequence, RNN, LSTM, GRU).* Цей підхід використовує рекурентні нейронні мережі (RNN) для обробки тексту у вигляді послідовності токенів. Зазвичай складається з:

- Енкодера, який аналізує вхідний текст;
- Декодера, який генерує реферат на основі вхідної інформації.

Приклад:

- Алгоритм Pointer-Generator Networks (See et al., 2017) комбінує екстрактивний і абстрактивний підходи, дозволяючи мережі вибирати слова безпосередньо з вихідного тексту або генерувати їх заново.

2. *Transformer-архітектури (BERT, GPT, T5, Pegasus).* Новіші моделі використовують трансформери, що значно покращують якість узагальнення.

Популярні моделі:

- BERTSUM (2019) – модифікація BERT для реферування;
- T5 (Text-to-Text Transfer Transformer, 2020) – модель, яка виконує перетворення тексту в узагальнену версію;
- PEGASUS (2020) – спеціально створена для абстрактивного реферування, тренується на вилученні важливих речень і відновленні їх змісту.

Архітектура моделей абстрактивного реферування

Більшість сучасних моделей використовують архітектуру Transformer:

- *Механізм уваги (Attention)* – дозволяє фокусуватися на найважливіших частинах тексту;
- *Маскований механізм уваги (Masked Attention)* – запобігає витоку інформації під час генерації тексту;
- *Механізм копіювання (Copy Mechanism)* – дозволяє алгоритму використовувати частину оригінального тексту для підвищення точності.

Схема роботи:

- Текст подається в енкодер, який переводить слова в векторне представлення;
- Декодер прогнозує наступні слова, генеруючи новий текст;
- Модель уваги визначає, які частини тексту є найважливішими.

Оцінка якості абстрактивного реферування

Для оцінки якості рефератів використовують автоматичні метрики:

1. *ROUGE* – порівнює збіги між згенерованим рефератом і еталонним текстом.

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) — це набір метрик для оцінки якості автоматично згенерованих текстів, особливо у задачах реферування, машинного перекладу та автоматичного узагальнення тексту.

Метрика стала стандартом для оцінки моделей у задачах абстрактивного та екстрактивного реферування.

Основна ідея *ROUGE*. Метрика *ROUGE* порівнює автоматично згенерований текст із *еталонним* (*референсним*) текстом, використовуючи різні підходи до вимірювання схожості:

- Обчислення точних збігів слів або фраз;
- Порівняння порядку слів у тексті;
- Вимірювання покриття вихідного змісту;

- Основний принцип: чим більше збігів між згенерованим та референсним текстом, тим краща якість реферату.

Види метрик $ROUGE$.

1. $ROUGE-N$ *(Unigram, Bigram, N-gram Overlap)* – метрика, що оцінює точність збігів n-грам між згенерованим текстом і еталонним рефератом. Обчислення за формулою (12.25):

$$ROUGE-N=\frac{\sum_{S\in Reference\ Summaries}(\sum_{gram\in S}(match\ Count(gram)))}{\sum_{S\in Reference\ Summaries}(\sum_{gram\in S}(total\ Count(gram)))}. \quad (12.25)$$

$ROUGE-1$ – оцінює точність збігів окремих слів (уніграм).

$ROUGE-2$ – оцінює точність збігів біграм (2-грам).

$ROUGE-3$, $ROUGE-4$ – оцінюють триграми, чотириграми тощо (рідко використовуються).

Приклад:

- *Оригінальний текст*:

```
Кішка сиділа на вікні.
```

- *Згенерований реферат*:

```
Кішка на вікні.
```

$ROUGE-1=\frac{3}{4}$ (75%) — три з чотирьох слів збіглися.

2. $ROUGE-L$ – вимірює найдовшу спільну підпослідовність (LCS) між референсним і згенерованим текстом. Обчислення за формулою (12.26):

$$ROUGE-L=\frac{LCS(\text{генерований текст,референсний текст})}{\text{довжина референсного тексту}}. \quad (12.26)$$

Перевага $ROUGE-L$:

- Враховує порядок слів у тексті.
- Не потребує вибору фіксованого розміру n-грам.

Приклад:

Оригінальний текст:

```
Кішка сиділа на вікні і дивилася на вулицю.
```

Згенерований реферат:

```
Кішка дивилася на вікно.
```

Найдовша спільна підпослідовність: «Кішка дивилася на вікно».

$$ROUGE-L=\frac{4}{8}=50\%.$$

Слід зазначити, що «вікно» та «вікні» вважалися одним словом, що є цілком логічним.

3. $ROUGE-W$ (*Weighted Longest Common Subsequence*) – це модифікація $ROUGE-L$, яка враховує вагу довгих послідовностей збігів. Якщо довші фрагменти тексту збігаються, вони мають більшу вагу у підрахунку, ніж окремі слова.
4. $ROUGE-S$ (*Skip-Bigram Co-occurrence Statistics*) – це метрика, що враховує біграми з пропусками, тобто дозволяє враховувати слова, які можуть бути розділені іншими словами у реченні.
Формула:

$$ROUGE-S = \frac{\text{Кількість збігів біграм із пропусками}}{\text{Загальна кількість біграм}}. \quad (12.27)$$

Приклад:
Оригінальний текст:

```
Кішка сиділа на вікні і дивилася на вулицю.
```

Згенерований реферат:

```
Кішка дивилася у вікно.
```

$ROUGE-S$ дозволяє врахувати зв'язок між словами «кішка – дивилася» та «дивилася – вікно», навіть якщо вони не стоять поряд.
2. *BLEU* – оцінює схожість між вихідним текстом і рефератом.

BLEU (*Bilingual Evaluation Understudy*) — це метрика, яка використовується для оцінки якості автоматично згенерованого тексту, особливо у задачах машинного перекладу та реферування.

Метрика стала однією з найпопулярніших у сфері оцінювання моделей Natural Language Processing (NLP).

BLEU обчислюється через усереднену прецизійність n-грам з пеналізацією довжини (brevity penalty, BP) за формулою (12.28):

$$BLEU = BP \cdot e^{\left(\sum_{n=1}^{N}(w_n \cdot log(P_n))\right)}, \quad (12.28)$$

де:

P_n – точність n-грам ($Precision$) обчислюється за формулою (12.29):

$$P_n = \frac{\sum_{\text{усі збіги } n-\text{грам}}}{\sum_{\text{всі } n-\text{грам у кандидаті}}}; \quad (12.29)$$

w_n – ваги (зазвичай рівні для всіх n);

log – логарифм за основою 2, 10 або натуральний логарифм (використовується у більшості випадків;

BP – штраф за короткі тексти обчислюється за формулою (12.30):

$$BP = \begin{cases} 1, & \text{якщо } c > r; \\ e^{(1-r/c)}, & \text{якщо } c \leq r, \end{cases} \quad (12.30)$$

де:

c – довжина згенерованого тексту;

r – довжина референсного тексту.

$BLEU$ зазвичай використовує біграми, триграми та чотириграми ($N = 4$).

Приклад розрахунку $BLEU$.

Референсний текст:

```
Кішка сидить на вікні і дивиться на вулицю.
```

Згенерований текст:

```
Кішка дивиться на вікно.
```

$BLEU - 1$ (*уніграм*):

- У референсі:

[«Кішка», «сидить», «на», «вікні», «і», «дивиться», «на», «вулицю»];

- У кандидаті: [«Кішка», «дивиться», «на», «вікно»];
- Спільні слова: [«Кішка», «дивиться», «на»].

За формулою (12.29) при $n = 1$:

$$P_1 = \frac{3}{4} = 0{,}75\ (75\%).$$

За формулою (12.30), враховуючи, що довжина референсу дорівнює 8, довжина кандидата дорівнює 4

$$BP = e^{\left(1-\frac{8}{4}\right)} = e^{-1} \approx 0{,}3679.$$

За формулою (12.28), враховуючи, що $w_1 = 1$:

$$BLEU - 1 = 0{,}3679 \cdot e^{ln(0{,}75)} = 0{,}3679 \cdot 0{,}75 \approx 0{,}276.$$

$BLEU - 2$ (*біграми*):

- У референсі:

[(«Кішка сидить»), («сидить на»), («на вікні»), («вікні і»), . . .];

- У кандидаті: [(«Кішка дивиться»), («дивиться на»), («на вікно»)];
- Спільні біграми: [«дивиться на»].

За формулою (12.29) при $n = 2$:

$$P_2 = \frac{1}{3} = 0{,}33\ (33\%).$$

$$BP \approx 0{,}3679.$$

За формулою (12.28), враховуючи, що $w_1 = w_1 = 0{,}5$:

$$BLEU - 2 = 0{,}3679 \cdot e^{\frac{1}{2}\cdot(ln(0{,}75)+ln(0{,}33))} \approx 0{,}183\ (18.30\%).$$

$BLEU - 3$:

- Триграми у референсі:

(«Кішка», «сидить», «на»);
(«сидить», «на», «вікні»);
(«на», «вікні», «і»);
(«вікні», «і», «дивиться»);

(«і», «дивиться», «на»);
(«дивиться», «на», «вулицю»).

- Триграми у кандидата:

(«Кішка», «дивиться», «на»);
(«дивиться», «на», «вікно»).

- Спільні триграми між кандидатом і референсом: Жодна триграма не збігається
- Кількість спільних триграм: 0;
- Загальна кількість триграм у кандидатові: 2.

За формулою (12.29) при $n = 3$ отримуємо, що $P_3 = 0$.

Довжина пенальті $BP = 0{,}3679$.

Оскільки $P_3 = 0$, то $ln(P_3) = -\infty$, що призведе до того, що

$$BLEU - 3 = 0.$$

$BLEU - 4$:

- Чотириграми у референсі:

(«Кішка», «сидить», «на», «вікні»);
(«сидить», «на», «вікні», «і»);
(«на», «вікні», «і», «дивиться»);
(«вікні», «і», «дивиться», «на»);
(«і», «дивиться», «на», «вулицю»).

- Чотириграми у кандидатові:

(«Кішка», «дивиться», «на», «вікно»);

- Спільні чотириграми між кандидатом і референсом: Жодна чотириграма не збігається;
- Кількість спільних чотириграм: 0;
- Загальна кількість чотириграм у кандидатові: 1.

За формулою (12.29) при $n = 4$ отримуємо, що $P_4 = 0$.

Довжина пенальті $BP = 0{,}3679$.

Оскільки $P_3 = P_4 = 0$, то $ln(P_3) = ln(P_4) = -\infty$, що призведе до того, що

$$BLEU - 4 = 0.$$

3. $BERTScore$ – використовує семантичне порівняння текстів за допомогою BERT.

$BERTScore$ — це метрика для оцінки якості згенерованого тексту, яка використовує глибокі нейронні мережі, зокрема модель BERT. На відміну від $BLEU$ або $ROUGE$, $BERTScore$ враховує семантичну схожість слів, а не просто збіги n-грам.

Основна ідея $BERTScore$:

- Замість порівняння n-грам, як у $BLEU$ або $ROUGE$, $BERTScore$ працює з векторами значень слів (ембедингами), отриманими з $BERT$;

- Метрика вимірює косинусну схожість між векторами слів у згенерованому тексті та референсі.

Приклад:

Референс:

```
Автомобіль їде дорогою.
```

Кандидат:

```
Машина рухається трасою.
```

$BLEU$ та $ROUGE$ оцінили б схожість як низьку через різні слова. Але $BERTScore$ врахує, що «автомобіль» ≈ «машина», «їде» ≈ «рухається», і дасть вищий бал.

Основні кроки обчислення $BERTScore$

Крок 1. Отримуємо векторні представлення слів з моделі BERT.

Крок 2. Обчислюємо косинусну подібність між словами у згенерованому та референсному текстах.

Крок 3. Використовуємо $Precision$ (12.31), $Recall$ (12.32) та F_1 (12.33) для підсумкової оцінки.

$$Precision = \frac{1}{m} \cdot \sum_{x \in Candidate} \left(\max_{y \in Reference} s(x,y) \right), \quad (12.31)$$

$$Recall = \frac{1}{n} \cdot \sum_{y \in Reference} \max_{x \in Candidate} s(x,y), \quad (12.32)$$

$$BERTScore = F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (12.33)$$

де:

m – кількість слів у кандидаті;

n – кількість слів у референсі;

$s(x,y)$ – косинусна подібність між словами x та y.

$BERTScore$ враховує і $Precision$, і $Recall$, тому найчастіше використовується як фінальна метрика.

Порівняння метрик оцінка якості абстрактивного реферування

Порівняння метрик $BLEU$, $ROUGE$ та $BERTScore$ наведено в табл. 12.10.

Таблиця 12.10. Порівняння метрик $BLEU$, $ROUGE$ та $BERTScore$ між собою

Метрика	Опис	Враховує семантику?	Чутливість до порядку	Найкраще підходить для
$BLEU$	Порівнює точність n-грам	Ні	Так	Машинний переклад
$ROUGE$	Рахує збіги слів	Ні	Так	Реферування (екстрактивне)

BERTScore	Косинусна подібність BERT-ембедингів	Так	Ні	Абстрактивне реферування, генерація тексту

12.3.3.2. Екстрактивні алгоритми

Екстрактивні алгоритми ґрунтуються на виборі найважливіших фрагментів або речень із вихідного тексту для створення резюме, зберігаючи при цьому їхню первісну форму та структуру. На відміну від абстрактивних методів, екстрактивні алгоритми не генерують нові речення, а використовують готові елементи тексту. Це робить їх швидкими та простими в реалізації, що особливо корисно для завдань, де потрібна висока продуктивність і швидкість обробки.

Екстрактивні методи зазвичай базуються на статистичних показниках, таких як частота слів або позиція речень у тексті. Наприклад, чим частіше певне слово зустрічається в документі, тим важливішим воно може бути для тексту, тому речення, яке його містить, може бути обране для резюме. Також часто використовується метод позиційного аналізу, коли важливими вважаються речення на початку та наприкінці абзаців, оскільки вони зазвичай містять узагальнення або ключові висновки. Одним з ефективних інструментів для оцінки важливості слів є TF-IDF, який дозволяє визначати унікальні терміни для конкретного тексту.

Перевагою даних алгоритмів є *простота та ефективність*. Вони не потребують глибоких мовних моделей або складних нейронних мереж, тому вимагають значно менших обчислювальних ресурсів ніж розглянутів вище алгоритми. Це робить їх придатними для використання в реальних умовах, де швидкість та економічність є важливими факторами. Крім того, екстрактивні алгоритми не вимагають значного навчання або налаштування, що спрощує їхнє впровадження у різні системи реферування текстів.

Недоліки:

- *Резюме, створені за допомогою екстрактивних методів, можуть втрачати логічну зв'язність*. Оскільки алгоритм просто витягує окремі речення з тексту, вони можуть бути вирвані з контексту, що іноді призводить до фрагментації резюме або втрати загальної смислової структури. Також екстрактивні алгоритми не враховують семантичні зв'язки між реченнями, що може ускладнити узагальнення складних текстів;
- *Обмежена адаптивність* цих методів до різних типів текстів. Алгоритми часто базуються на поверхневих ознаках, як-от частота слів або позиція речення, вони можуть не забезпечити належного узагальнення для текстів із складною або специфічною структурою, як, наприклад, художня література чи тексти з великою кількістю контекстуальних деталей.

Методи екстрактивного реферування

1. *Статистичні методи* (*без машинного навчання*).

TF-IDF:

- Оцінює важливість слів у тексті (див. лекцію 11) та обчислюється за формулою (12.34):

$$TF - IDF(w) = TF(w) \cdot IDF(w); \quad (12.34)$$

- Речення з найбільш вагомими словами включаються в реферат.

TextRank (графовий метод, подібний до PageRank):

- Будується граф, де вузли — це речення, а ребра — їхня семантична подібність;
- Визначаються найважливіші вузли (речення) за допомогою ітеративного алгоритму ранжування.

LexRank (модифікація TextRank):

- Використовує косинусну подібність між реченнями для побудови графа;
- Найбільш центральні речення в графі потрапляють у реферат.

Luhn's Algorithm (на основі частоти слів):

- Визначає «інформативні» слова та їхні скупчення;
- Речення з високою концентрацією важливих слів додаються до реферату.

2. *Методи з використанням машинного навчання*.

Супервізовані методи (*Neural Networks, Random Forest, SVM, LSTM*):

- Використовують моделі, навчені на великих наборах текстових пар («оригінал» → «резюме»);
- Ознаки: довжина речення, частота слів, розташування в тексті, TF-IDF, семантична подібність тощо.

Нейронні мережі (*Transformers, BERT, BART, T5*):

- Використовують контекстні ембеддинги (наприклад, BERT) для визначення важливих речень;
- Деякі моделі (BART, T5) можуть працювати як у екстрактивному, так і в абстрактивному режимі.

Порівняння екстрактивних методів реферування текстів

Порівняння екстактивних методів наведено в табл. 12.11.

Таблиця 12.11. Порівняння екстрактивних методів реферування текстів

Метод	Переваги	Недоліки
TF-IDF	Простий у реалізації, швидкий	Не враховує семантику тексту
TextRank	Без учителя, добре працює з довгими текстами	Чутливий до параметрів графа
LexRank	Враховує подібність між реченнями	Може виділяти надто схожі речення
Luhn's Algorithm	Підходить для технічних текстів	Вибирає речення за локальною частотою слів

ML-моделі (SVM, LSTM)	Можуть навчитися виділяти ключові речення	Вимагають міткованих даних

12.4. Машинний переклад текстів

Машинний переклад текстів (*Machine Translation*, *MT*) — це галузь штучного інтелекту та обробки природної мови (NLP), що займається автоматичним перекладом текстів з однієї мови на іншу за допомогою комп'ютерних алгоритмів.

12.4.1. Постановка задачі машинного перекладу тексту

Змістовна постановка задачі

Загальна мета задачі. Переклад тексту з однієї природної мови (мова-джерело) на іншу (мова-призначення) автоматично, зберігаючи смисл, граматику та стилістику.

Вхідні дані. Текст або речення мовою-джерелом (наприклад, українською (рис. 12.14)):

Guten Tag! Mein Name ist Alex. Ich bin neunzehn Jahre alt. Ich wohne in Kyjiw, der Hauptstadt der Ukraine und es ist fantastische Stadt. Heute ist kalt und 13 Grad. Bei uns in Kyjiw ist bewölkt.

Ich habe keine Geschwister, ich bin Einzelkind. Ich bin Student. Meine Spezialität ist Programmieren. Ich mache gerade ein Praktikum als Entwickler.

Meine Hobbys sind Lesen und Filme schauen. Jetzt lese ich "Im Westen nichts Neues" von Remarque. Ich spiele auch gerne Schach und Dame. Ich möchte. Mein Traum ist Gitarre spielen zu lernen und durch Europe zu reisen.

Ich habe keine Haustiere, aber Oma hat einen Hund und eine Katze.

Рис. 12.14. Вхідний текст німецькою мовою

Очікуваний вихід. Той самий текст, перекладений мовою-призначенням (наприклад, англійською (рис. 12.15 та рис.12.16)):

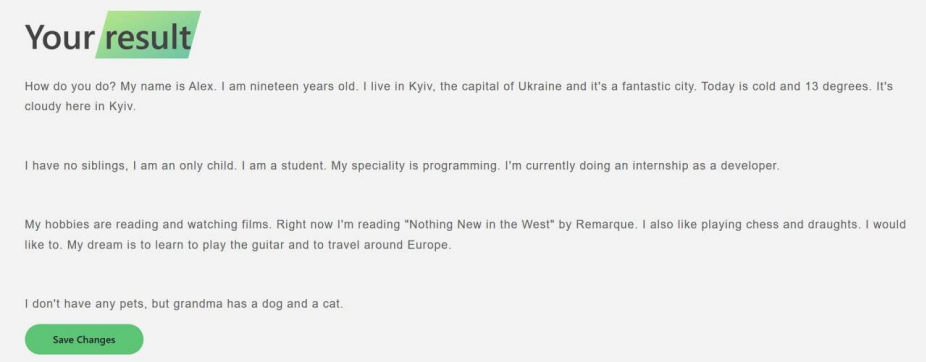
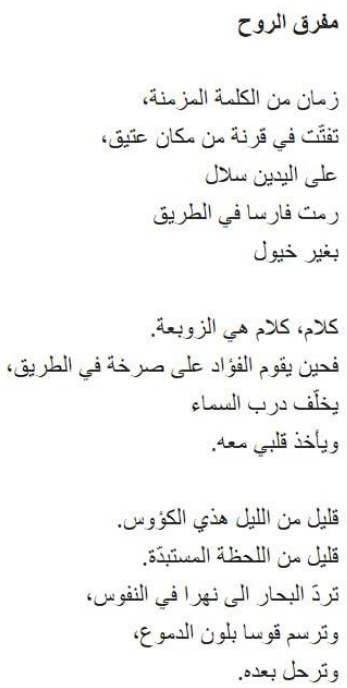


Рис. 12.15. Переклад тексту англійською мовою


مفرق الروح

زمان من الكلمة المزمنة،
تفتّت في قرنة من مكان عتيق،
على اليدين سلال
رمت فارسا في الطريق
بغير خيول

كلام، كلام هي الزوبعة.
فحين يقوم الفؤاد على صرخة في الطريق،
يخلّف درب السماء
ويأخذ قلبي معه.

قليل من الليل هذي الكؤوس.
قليل من اللحظة المستبدّة.
تردّ البحار الى نهرا في النفوس،
وترسم قوسا بلون الدموع،
وترحل بعده.

Рис. 12.16. Переклад тексту арабською мовою

Формальна постановка задачі

Задано:

- Мова-джерело S;
- Мова-призначення T;
- Множина парних речень $D = \{(s_i, t_i)\}_{i=1}^{N}, s_i \in S, t_i \in T$.

Потрібно побудувати функцію перекладу (12.34):

$$f: S \to T, \tag{12.34}$$

яка мінімізує помилку перекладу, тобто:

$$f(s) = arg\left(\max_{t \in T}(P(t \mid s))\right), \tag{12.35}$$

де $P(t \mid s)$ — умовна ймовірність, що речення ttt є правильним перекладом речення s.

Ключові вимоги до функції перекладу:

- *Семантична точність* — збереження значення оригінального тексту;
- *Граматична правильність* — відповідність правилам мови-призначення;
- *Стилістична відповідність* — передача формального або неформального стилю;
- *Контекстна узгодженість* — врахування попередніх речень або абзаців.

12.4.2. Типи машинного перекладу

Основні типи машинного перекладу наведено в табл. 12.12.

Таблиця 12.12. Типи машинного перекладу текстів

Тип	Опис	Приклади

Статистичний (SMT)	Переклад базується на ймовірностях появи слів/фраз	Google Translate (до 2016)
Нейронний (NMT)	Використовує глибокі нейронні мережі (зазвичай seq2seq)	Google Translate, DeepL, OpenNMT
Правило-орієнтований (RBMT)	Побудований на лінгвістичних правилах	Apertium
Гібридний	Поєднує кілька підходів (правила + статистика або нейронка)	Modern CAT tools

12.4.3. Популярні інструменти та бібліотеки для автоматичного перекладу тексту

Популярні інструменти та бібліотеки для автоматичного перекладу тексту наведено в табл. 12.13.

Таблиця 12.13. Популярні інструменти автоматичного перекладу

Інструмент / Бібліотека	Опис
Google Translate API	Один із найточніших хмарних сервісів перекладу. Підтримує 100 + мов.
DeepL Translator	Відомий якістю перекладу, особливо для європейських мов. Має безкоштовну і Pro-версію.
Amazon Translate	Хмарний сервіс від AWS для масштабного перекладу в реальному часі.
Microsoft Translator (Azure)	Частина Azure Cognitive Services, зручний для інтеграції в бізнес-додатки.
Meta No Language Left Behind (NLLB)	Open-source модель від Meta, орієнтована на малорозповсюджені мови.
HuggingFace Transformers (OpenNMT, MarianMT)	Бібліотека з відкритим кодом для використання передових моделей обробки природної мови (NLP)

Приклад на Python з HuggingFace Transformers:

```
from transformers import pipeline
```

```
translator = pipeline("translation", model="Helsinki-NLP/opus-mt-en-uk")
result = translator("This is a test sentence.", max_length=40)
print(result[0]['translation_text'])  # → Це тестове речення.
```

12.4.4. Числові метрики для оцінки якості машинного перекладу

Для оцінки якості машинного перекладу використовуються різні автоматичні метрики. Вони порівнюють машинний переклад із референсними (еталонними) перекладами або оцінюють текст безпосередньо.

Метрики на основі порівняння з референсним перекладом

Ці метрики оцінюють схожість між машинним перекладом та еталонним перекладом, виконаним людиною.

1. *BLEU.*

BLEU (*Bilingual Evaluation Understudy*) – найпопулярніша метрика для оцінки машинного перекладу. Дана метрика:

- Обчислює n-грамну схожість між машинним і референсним перекладами;
- Використовує модифіковану точність n-грам та пенальті за довжину ($brevity\ penalty$, BP);
- Обчислюється за формулою (12.28).

Приклад у Python:

```
from nltk.translate.bleu_score import sentence_bleu

reference = [["Це", "тестовий", "переклад"]]
candidate = ["Це", "пробний", "переклад"]

score = sentence_bleu(reference, candidate)
print(f"BLEU Score: {score}")
```

Недоліки:

- Чутливий до точної відповідності слів (не враховує синоніми);
- Погано працює для коротких текстів.

2. *METEOR.*

METEOR (*Metric for Evaluation of Translation with Explicit ORdering*) – метрика, яка:

- Враховує не лише точну відповідність слів, але й синоніми, морфологію та порядок слів;
- Використовує F-міру (гармонійне середнє точності та повноти) і штрафи за зміни порядку слів;

- Обчислюється за формулою (12.36):

$$METEOR = F_{mean} \cdot (1 - Penalty), \quad (12.36)$$

де:

F_{mean} – середнє значення $Precision$ та $Recall$ (F-міра) (див. формулу (3.10) лекції 3);

$Penalty$ – штраф за зміну порядку слів.

Приклад у Python:

```
from nltk.translate.meteor_score import meteor_score

reference = ["Це тестовий переклад"]
candidate = "Це пробний переклад"

score = meteor_score([reference], candidate)
print(f"METEOR Score: {score}")
```

$METEOR$ краще враховує контекст, ніж $BLEU$.

3. *TER.*

TER (*Translation Edit Rate*) – це метрика, яка:

- Вимірює кількість редагувань (вставка, видалення, заміна, перестановка), необхідних для перетворення машинного перекладу на референсний;
- Обчислюється за формулою (12.37):

$$TER = \frac{\text{кількість редагувань}}{\text{кількість слів у референсі}}. \quad (12.37)$$

Чим менший TER, тим кращий переклад.

Ї

Приклад у Python:

```
from sacrebleu.metrics import TER

ter = TER()
score = ter.sentence_score("Це пробний переклад",
["Це тестовий переклад"])
print(f"TER Score: {score.score}")
```

Недоліки:

- TER не враховує синоніми та граматику;
- Чутливий до перестановок.

Метрики без референсного перекладу

Ці метрики оцінюють якість перекладу без порівняння з еталонним перекладом.

1. *COMET.*

COMET (*Crosslingual Optimized Metric for Evaluation of Translation*) – це метрика, яка:

- Використовує нейронну мережу для оцінки якості перекладу.
- Враховує значення слів у контексті та семантику.
- Показує кращу кореляцію з людськими оцінками, ніж *BLEU* або *METEOR*.

Приклад у Python:

```
from comet import download_model,
load_from_checkpoint

model_path = download_model("wmt20-comet-da")
model = load_from_checkpoint(model_path)

data = [{"src": "This is a test translation", "mt":
"Це пробний переклад", "ref": "Це пробний переклад"}]
score = model.predict(data)
print(f"COMET Score: {score}")
```

COMET є найсучаснішою метрикою, але потребує ресурсів для роботи.

2. *BERTScore*.

BERTScore – це метрика, яка:

- Використовує BERT для порівняння глибинних векторних представлень (word embeddings) перекладу та оригіналу;
- Враховує синоніми, контекст, граматику;
- Обчислюється за формулою (12.33).

Приклад у Python:

```
from bert_score import score

candidates = ["Це пробний переклад"]
references = ["Це тестовий переклад"]

P, R, F1 = score(candidates, references, lang="uk",
rescale_with_baseline=True)
print(f"BERTScore F1: {F1.mean().item()}")
```

BERTScore добре працює на довгих текстах і складних мовах.

Порівняння метрик якості перекладу

Порівняння перерахованих метрик якості перекладу наведено в табл. 12.14.

Таблиця 12.14. Порівняння метрик якості перекладу

Метрика	Враховує контекст?	Підходить для довгих текстів?	Недоліки
$BLEU$	Ні	Обмежено	Чутливий до точного збігу слів
$METEOR$	Так	Так	Вимагає більше обчислень
TER	Ні	Так	Ігнорує семантику
$COMET$	Так	Так	Потребує нейромережі
$BERTScore$	Так	Так	Вимагає потужних обчислень

Рекомендації:

- Використовуй $BLEU$ для швидкої оцінки;
- $METEOR$ кращий для врахування синонімів;
- $BERTScore$ / $COMET$ підходять для сучасних моделей перекладу.

12.4.5. Порівняння Google Translate API та DeepL Translator

На рис. 12.17 та 12.18 наведено порівняння Google Translate API та DeepL Translator на основі перерахованих метрик.

```
BLEU score deepl: 0.59
BLEU score google: 0.09
METEOR Score Deepl: 0.17
METEOR Score Google: 0.00
BERTScore Deepl (F1): 0.93
BERTScore Google (F1): 0.75
```

Рис. 12.17. Значення метрик якості перекладу для Google Translate API та DeepL Translator

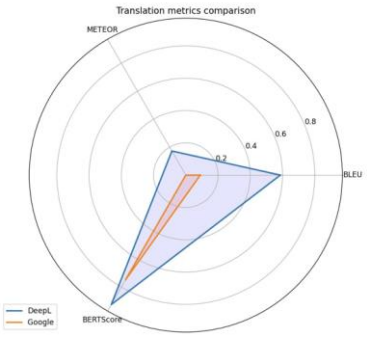


Рис. 12.18. Radio Cart метрик

З рис. 12.17 та 12.18 видно, що переклад, зроблений DeepL Translator, є якіснішим у порівнянні з перекладом, зробленим Google Translate API.

ЛЕКЦІЯ 13

ПРИНЦИПИ КОНТЕН ТА ІНТЕНТ-АНАЛІЗІВ. РЕКОМЕНДАЦІЙНІ СИСТЕМИ

13.1. Контент-аналіз

13.1.1. Поняття контент-аналізу

Контент-аналіз (*Contents* – *зміст*, *вміст*) або аналіз змісту – стандартний метод дослідження в галузі суспільних наук, предметом аналізу якого є зміст текстових масивів та продуктів комунікативної кореспонденції.

У вітчизняній дослідницької традиції контент-аналіз визначається як кількісний аналіз текстів і текстових масивів з метою подальшої змістовної інтерпретації виявлених числових закономірностей. Контент-аналіз застосовується при вивченні джерел, інваріантних щодо структури або суті змісту, але зовні існують, як несистематизований, безладно організований текстовий матеріал. Філософський сенс контент-аналізу як дослідницького методу полягає в сходженні від різноманіття текстового матеріалу до абстрактної моделі змісту тексту (понятійно-категоріальний апарат, двозначне, колізії, парадокси). У зазначеному сенсі, контент-аналіз є однією з номотетіческіх дослідних процедур, що використовуються в сфері застосування идиографический методів.

Основні особливості контент-аналізу

1. *Об'єктивність* – аналізує матеріали без суб'єктивних суджень;
2. *Кількісність та якісність* – може визначати частоту появи термінів або проводити глибоку інтерпретацію змісту;
3. *Автоматизованість* – сучасні методи дозволяють використовувати NLP (Natural Language Processing) для автоматичного аналізу.

Сфера застосування

Коло дисциплін, в яких застосовується контент-аналіз, досить широкий. Крім соціології і політології дана методика застосовується в антропології, управлінні персоналом, психології, літературознавстві, історії, історії філософії. Оле Холсти подає таке розподіл досліджень в галузі контент-аналізу по наукам: соціологія, антропологія – 27,7%, теорія комунікації – 25,9%, політична наука – 21,5%. Слід також відзначити застосування контент-аналізу в області історичних досліджень і зв'язків з громадськістю.

За допомогою контент-аналізу можна аналізувати такі різні типи текстів, як повідомлення ЗМІ, заяви політичних діячів, програми партій, правові акти, рекламні та пропагандистські матеріали, історичні джерела, літературні твори.

Методи контент-аналізу

1. *Ручний аналіз* – людина переглядає текст і кодує вручну;
2. *Автоматизований аналіз* – за допомогою Python, R, GPT, TF-IDF, LDA;

Інструменти контент-аналізу

1. *NLTK, SpaCy (аналіз тексту).*

NLTK – це бібліотека Python для обробки природної мови (NLP). Вона містить інструменти для токенізації, стемінгу, лематизації, аналізу частин мови (POS-tagging) та інших задач NLP.

SpaCy – це швидка та ефективна бібліотека Python для обробки природної мови (NLP). Вона використовується для токенізації, лематизації, визначення частин мови (POS-тегування), розпізнавання сутностей (NER) та інших NLP-завдань.

2. *WordStat (статистичний аналіз текстів).*

WordStat — це програмне забезпечення для контент-аналізу та текстової аналітики, яке використовується для аналізу великих обсягів тексту, виявлення тенденцій та закономірностей у документах. В основному використовується в дослідженнях ринку, соціології, політології та криміналістиці.

Основні можливості WordStat:

- *Частотний аналіз* – підрахунок частоти слів і фраз у текстах;
- *Аналіз ключових слів* – виявлення важливих термінів;
- *Тематика та кластеризація* – групування текстів за темами;
- *Семантичний аналіз* – виявлення зв'язків між словами та поняттями;
- *Графічна візуалізація* – побудова діаграм, карт асоціацій, дендограм;
- *Імпорт із різних джерел* – підтримка аналізу текстів з PDF, $DOCX$, $ELSX$, соціальних мереж тощо.

Застосування WordStat:

- Аналіз соціальних медіа та відгуків;
- Політичні дослідження та аналіз громадської думки;
- Юридична експертиза та криміналістика;
- Аналіз наукових статей та публікацій.

3. *LIWC (аналіз психологічних характеристик тексту).*

LIWC (*Linguistic Inquiry and Word Count*) – це програмне забезпечення для аналізу тексту, яке дозволяє визначити психологічні, емоційні та когнітивні характеристики автора на основі використання слів.

Основні можливості LIWC:

- *Аналіз емоційного стану* – визначення позитивних та негативних емоцій у тексті;
- *Психолінгвістичний аналіз* – аналіз когнітивних процесів, соціальних взаємодій, особистісних рис;
- *Категоризація слів* – LIWC містить словники з десятками категорій, включаючи емоції, мислення, сприйняття, соціальні зв'язки тощо;
- *Оцінка рівня стресу та депресії* – аналізує використання слів, пов'язаних із тривожністю, депресією та агресією;

- *Соціальні аспекти* – аналізує частоту вживання займенників («я», «ми»), що може вказувати на рівень егоцентризму чи колективізму автора;
- *Аналіз стилю мовлення* – визначає формальність, складність тексту та рівень абстрактності.

Приклад роботи LIWC:

Вхідний текст:

«Я почуваюся дуже втомленим,
весь день був жахливий,
ніхто мене не розуміє».

Аналіз LIWC:

- Негативні емоції: 80%;
- Слова, що вказують на втому: 50%;
- Слова, що вказують на самотність: 30%;
- Слова, пов'язані зі стресом: 40%.

Таким чином, LIWC може показати, що автор переживає емоційний стрес і почувається ізольованим.

Застосування LIWC:

- *Психологія та психіатрія* – аналіз стану людини на основі її письмового мовлення;
- *Соціальні науки* – вивчення поведінки людей у соцмережах, блогах, новинах;
- *Криміналістика* – аналіз текстів підозрюваних для виявлення намірів чи емоційного стану;
- *Аналіз політичного дискурсу* – дослідження риторики політиків та їхнього впливу на аудиторію;
- *HR та бізнес* – оцінка психологічного стану працівників через аналіз їхніх звітів, листів.

13.1.2. Історія методу

Методика контент-аналізу широко застосовується в інформаційну епоху, проте історія методу не обмежується епохою автоматичної обробки тексту. Так перші приклади використання контент-аналізу датовані XVIII століттям, коли в Швеції частота появи в тексті книги певних тем служила критерієм її єретичні. Однак, всерйоз говорити про застосування контент-аналізу як повноцінної методики можна лише починаючи з 30-х років XX століття в США. Термін content analysis вперше почали застосовувати в кінці XIX – поч. XX ст. американські журналісти Б. Меттью, А. Тенні, Д. Спіід, Д. Уіпкінс. Біля витоків становлення методології контент-аналізу стояв також французький журналіст Ж. Кайзер.

Використовувався контент-аналіз переважно в соціологічних дослідженнях, в тому числі при вивченні рекламних і пропагандистських матеріалів.

У сфері політичних досліджень початок використання методики контент-аналізу поклав Г. Лассуел, який вдався до аналізу

пропагандистських матеріалів періоду Другої світової війни. У 1960-і роки, під час так званого методологічного вибуху дослідження із застосуванням методики контент-аналізу особливо активізувалися. Це сприяло розвитку методики, урізноманітнило її варіанти. Саме в цей період починається активне використання комп'ютерної техніки в дослідженнях.

13.1.3. Основні типи контент-аналізу

13.1.3.1. Кількісний контент-аналіз

13.1.3.1.1. Описання кількісного контент-аналізу

Кількісний контент-аналіз (також називається *змістовним*) грунтується на дослідженні слів, тим і повідомлень, зосереджуючи увагу дослідника на змісті повідомлення. Таким чином, збираючись піддати аналізу обрані елементи, потрібно вміти передбачати їх зміст і визначати кожен можливий результат спостереження відповідно до очікувань дослідника.

Переваги:

- Об’єктивність і точність;
- Можливість аналізу великих обсягів текстів;
- Використання статистичних методів.

Обмеження:

- Неможливість аналізу глибоких смислів;
- Вразливість до помилок у кодуванні;
- Відсутність контексту (наприклад, іронія не враховується).

Кроки проведення кількісного контент-аналізу

Крок 1. Визначення мети дослідження. Визначення сукупності досліджуваних джерел або повідомлень за допомогою набору заданих критеріїв, яким має відповідати кожне повідомлення:

- Заданий тип джерела (преса, телебачення, радіо, рекламні або пропагандистські матеріали);
- Один тип повідомлень (статті, замітки, плакати);
- Участь в процесі комунікації (відправник, отримувач (реципієнт));
- Розмір повідомлень (мінімальний обсяг або довжина);
- Частота появи повідомлень;
- Спосіб поширення повідомлень;
- Місце поширення повідомлень;
- Час появи повідомлень.

При необхідності можна використовувати й інші критерії, проте перераховані вище зустрічаються найчастіше.

Наприклад:

- «Які теми найчастіше зустрічаються в політичних промовах?»

Крок 2. Вибір матеріалу. Формування вибіркової сукупності повідомлень. У деяких випадках можна вивчати всю визначену на першому етапі сукупність джерел, оскільки підлягають аналізу випадки (повідомлення) часто обмежені за кількістю і добре доступні. Однак іноді контент-аналіз

повинен спиратися на обмежену вибірку, взяту з більшого масиву інформації.

Наприклад:

- Новини, статті, соцмережі, рекламні тексти, промови тощо.

Крок 3. Кодування та категоризація. Виявлення одиниць аналізу. Ними можуть бути слова або теми. Правильний вибір одиниць аналізу – важлива складова всієї роботи. Найпростішим елементом повідомлення є слово. Тема – це інша одиниця, що представляє собою окреме висловлювання про який-небудь предмет. Існують досить чіткі вимоги до вибору можливої одиниці аналізу:

- Вона повинна бути досить великою, щоб висловлювати значення;
- Вона повинна бути досить малою, щоб не висловлювати багато значень;
- Вона повинна легко ідентифікуватися;
- Число одиниць має бути настільки великим, щоб з них можна було робити вибірку.

Якщо в якості одиниці аналізу обирається тема, то вона також виділяється відповідно до деякими правилами:

- Тема не може виходити за межі абзацу;
- Нова тема виникає, якщо відбувається зміна: сприймання, дії, мети, категорії.

Існують також і спеціальні методики контент-аналізу, адаптовані до потреб історичних та історико-філософських досліджень:

- Виділення ключових слів, тем або понять;

Наприклад:

- У політичному аналізі можна виділити категорії: економіка, соціальна політика, безпека.

Крок 4. Підрахунок частоти появи елементів. Виділення одиниць обрахунку, які можуть збігатися зі смисловими одиницями або носити специфічний характер. У першому випадку процедура аналізу зводиться до підрахунку частоти згадки виділеної смислової одиниці, в другому дослідник на основі аналізованого матеріалу і цілей дослідження сам висуває одиниці рахунку, якими можуть бути:

- Фізична протяжність текстів;
- Площа тексту, заповнена смисловими одиницями;
- Число рядків (абзаців, знаків, колонок тексту);
- Тривалість трансляції по радіо або ТБ;
- Метраж плівки при аудіо- і відеозаписах,
- Кількість малюнків з певним змістом, сюжетом і ін.

У деяких випадках дослідники використовують і інші елементи рахунку. Принципове значення на цьому етапі контент-аналізу має суворе дефінірованія його операторів.

Крок 5. Обрахунок та аналіз результатів. Безпосередньо процедура підрахунку та виявлення тенденцій, закономірностей, порівняння різних текстів. Вона в загальному вигляді аналогічна стандартним прийомам класифікації по виділеним угрупованням. На даному кроці застосовуються складання спеціальних таблиць (табл. 13.1), використання комп'ютерних програм, спеціальних формул, статистичних розрахунків.

Таблиця 13.1. Процедура підрахунків

Одиниці аналізу		Одиниця обрахунку	
Категорія	Підкатегорія	Частота згадування	Частота згадування у %
1 категорія	01 підкатегорія	15	32
	02 підкатегорія	7	15
	03 підкатегорія	25	53
Сума		47	100

Крок 6. Висновки та інтерпретація. Інтерпретація отриманих результатів відповідно до цілей і завдань конкретного дослідження. Зазвичай на цьому етапі виявляються і оцінюються такі характеристики текстового матеріалу, які дозволяють робити висновки про те, що хотів підкреслити або приховати його автор. Можливо виявлення відсотка поширеності в суспільстві суб'єктивних смислів об'єкта або явища.

Принцип кількісного контент-аналізу

При проведенні контент-аналізу цього типу дослідник повинен створити свого роду словник, в якому кожне спостереження отримає визначення і буде віднесена до відповідного класу. В цьому випадку дослідника більше цікавить, про що йдеться.

При цьому можуть виникнути наступні проблеми:

1. *Перша проблема* полягає в тому, що дослідник повинен передбачити не тільки згадки, які можуть зустрітися, а й елементи їх контекстуального вживання, а для цього повинна бути розроблена детальна система правил оцінки кожного випадку вживання.

Це завдання полягає у аналізі сукупності повідомлень (тобто за допомогою виявлення на матеріалі невеликої вибірки повідомлень тих типів ключових згадок, які з найбільшою ймовірністю можуть зустрітися в наступному, більш повному аналізі) в поєднанні з арбітражними оцінками контекстів і способів вживання термінів. Переважно може мати справу зі спостереженнями не одного, а декількох дослідників.

2. Більш важкою є *друга проблема*, яка полягає в необхідності приписування ключовим згадкам конкретних оцінок, – коли ми повинні вирішити, чи наводиться згадка об'єкта, що нас цікавить, в позитивному або негативному сенсі, за або проти, і т. д., а також коли нам треба ранжувати ряд згадок відповідно силі їх оцінок (тобто відповідно до того, яка з них найпозитивніше, яке наступне за ним по позитивності і т. д.). При цьому

дослідник потребує досить тонкі показники, якими можна було б вимірювати не тільки настрій політичних суб'єктів, а й силу цих настроїв.

Особливо важким виконання цього завдання є в історичних, історико-філософських і психологічних дослідженнях, оскільки передбачає високий рівень гуманітарної підготовки фахівців, що використовують методику контент-аналізу. Існує багато методів, що полегшують прийняття такого рішення. У деяких випадках вони спираються на судження групи арбітрів (експертів) про значення або силі (інтенсивності) деякого терміну.

Як приклад таких прийомів можна привести:

- Метод «Q-сортування»;
- Метод «Парного порівняння».

Докладно зупинимось на цих методах нижче.

13.1.3.1.2. Методи шкалування

Шкалування методом «Q-сортування»

В методі *«Q-сортування»* використовується шкала жорсткого розподілу з дев'яти пунктів: пункт 1 відповідає мінімальному ступені інтенсивності вимірюваного ознаки (наприклад, найменше схвалення), а пункт 9 – максимальному ступені інтенсивності (наприклад, найвищого ступеня схвалення).

Мета полягає в тому, щоб просто ранжувати (упорядкувати) все судження уздовж єдиної оцінної осі.

Основний принцип шкалування методом «Q-сортування» полягає в наступному:

1. Арбітру дається певна жорстка квота на кожну категорію шкали (тобто очікуване число слів або фраз, які повинні бути їм віднесені до даної категорії), а потім йому пропонується розподілити заданий набір термінів так, щоб встановлені квоти не порушувалися. Квоти засновані на припущенні (не обов'язково вірному), що коливання в інтенсивності слів і фраз повинні укладатися в рамки нормального розподілу (коли досліджувані випадки максимально зосереджені в середній частині шкали, а у міру просування до її полюсів їх число рівномірно убуває). Арбітри, таким чином, змушені давати відносні оцінки конкретних слів і фраз (випадків), відносячи їх до певних категорій шкали;
2. Після того як арбітри завершили свою роботу, обчислюється середня арифметична оцінка шкали для кожного випадку, а потім отримані середні оцінки відповідним чином ранжуються;
3. Далі результати цього ранжирування випадків за інтенсивністю використовуються для приписування аналізованих текстів кодів, зумовлених встречаемостью в них слів або тим, які отримали нашу оцінку. Довільність оцінки одного дослідника компенсується, таким чином, наявністю інших думок.

Основні кроки «Q-сортування».

Крок 1. Провести підготовку тверджень.

Крок 2. На основі опитування респондентів, провести групування тверджень.
Крок 3. Провести аналіз даних (сформувати групи респондентів зі схожими поглядами).

Приклад. Припустимо, що ми хочемо дослідити ставлення двох людей людей r_1 та r_2 до соціальних реформ.

1. Підготовка тверджень. Ми створюємо 15 тверджень про соціальні реформи.

«Соціальна рівність – головний пріоритет держави.»
«Підвищення податків необхідне для розвитку країни.»
«Держава повинна забезпечувати безкоштовну медицину.»
… (до 15 тверджень)

2. Групування тверджень. Респонденти отримують картки з цими твердженнями та повинні розкласти їх у групи за ступенем згоди:

- $+3$ (цілком згоден);
- $+2$ (скоріше згоден);
- $+1$ (нейтрально);
- 0 (байдужий);
- -1 (скоріше не згоден);
- -2 (незгоден);
- -3 (категорично не згоден).

Нехай отримано наступні відповіді респондентів:

$$r_1 = [3, 2, 3, 0, 2, 1, 2, 1, 0, 1, 1, -1, -1, -2, -3];$$
$$r_2 = [3, 3, 2, 1, 1, 0, 2, 1, -1, 1, 0, -2, -1, -2, -3].$$

3. Аналіз даних. Всі відповіді кодуються у матрицю Q-сортування (табл. 13.2 та табл. 13.3).

Таблиця 13.2. Приклад візуалізації Q-сортування для респондента r_1

Категорія оцінки	Кількість місць	Твердження респондента r_1
$+3$ (Цілком згоден)	2	(1), (3)
$+2$ (Скоріше згоден)	3	(2), (5), (7)
$+1$ (Нейтрально)	4	(6), (8), (10), (11)
0 (Байдужий)	2	(4), (9)
-1 (Скоріше не згоден)	2	(12), (13)
-2 (Незгоден)	1	(14)
-3 (Категорично не згоден)	1	(15)

Таблиця 13.3. Приклад візуалізації Q-сортування для респондента r_2

Категорія оцінки	Кількість місць	Твердження респондента r_2
$+3$ (Цілком згоден)	2	(1), (2)
$+2$ (Скоріше згоден)	2	(3), (7)

+1 (Нейтрально)	4	(4), (5), (8), (10)
0 (Байдужий)	2	(6), (11)
−1 (Скоріше не згоден)	2	(9), (13)
−2 (Незгоден)	2	(12), (14)
−3 (Категорично не згоден)	1	(15)

Далі використовується кореляційний аналіз (зокрема обчислення коефіцієнту рангової кореляції Спірмена) для виявлення рівня схожості поглядів респондентів.

4. Приклад реалізації Q-сортування в Python.

Реалізація коду

```
import numpy as np
from scipy.stats import spearmanr

# Відповіді двох респондентів
r1 = [3, 2, 3, 0, 2, 1, 2, 1, 0, 1, 1, -1, -1, -2, -
3]
r2 = [3, 3, 2, 1, 1, 0, 2, 1, -1, 1, 0, -2, -1, -2, -
3]

# Кореляція Спірмена (визначає схожість)
corr, _ = spearmanr(r1, r2)
print(f"Коефіцієнт Спірмена: {corr:.2f}")
```

Результат

```
Коефіцієнт Спірмена: 0.89 (дуже схожі погляди)
```

Шкалування методом «Парного порівняння»

Шкалування *методом «Парного порівняння»* має ті ж цілі, що і попередній метод, але техніка його дещо інша.

Основний принцип шкалування методом «Парного порівняння» полягає у наступному:

1. Кожен випадок, що підлягає оцінці, послідовно порівнюється попарно з усіма іншими випадками, при цьому кожен арбітр повинен вирішити, яке з слів (або фраз) в кожній парі сильніше (або інтенсивніше) іншого.

Так, якщо треба порівняти п'ять тверджень (випадків), то кожен арбітр буде послідовно порівнювати спочатку 1-е з них з 2-м, з 3-м, 4-м, 5-м, потім 2-е з 3-м, 4-м, 5-м і т. д., всякий раз при цьому відзначаючи, яке з двох більш інтенсивно.

2. Підрахувавши, скільки разів кожен випадок виявився згідно оцінки всіх арбітрів, сильнішим інших і, розділивши отримане число на кількість

арбітрів (тобто обчисливши середню оцінку, винесену групою арбітрів кожному твердженню), ми отримуємо можливість здійснити кількісне ранжирування всіх випадків за ступенем їх інтенсивності. Чим вище середня оцінка деякого твердження, тим воно, на думку арбітрів, сильніше.

Основні кроки методу «Парних порівнянь».

Крок 1. Скласти всі можливі пари для порівняння.
Крок 2. Провести оцінювання.
Крок 3. Підрахувати бали.
Крок 4. Ранжувати варіанти.

Приклад. Припустимо, що компанія хоче визначити найкращий логотип із 4 варіантів (A, B, C, D).

1. Складаємо всі можливі пари для порівняння. Отримаємо 8 пар:

$$(A\ vs. B);$$
$$(A\ vs. C);$$
$$(A\ vs. D);$$
$$(B\ vs. C);$$
$$(B\ vs. D);$$
$$(C\ vs. D).$$

2. Проводимо оцінювання. Респондентам пропонують обрати кращий логотип у кожній парі. Наприклад, голоси розподілилися так (табл. 13.4):

Таблиця 13.4. Розподіл голосів

Пари	Обраний варіант
$A\ vs. B$	A
$A\ vs. C$	C
$A\ vs. D$	A
$B\ vs. C$	B
$B\ vs. D$	B
$C\ vs. D$	C

3. Підрахунок балів:

- Логотип A: 2 перемоги;
- Логотип B: 2 перемоги;
- Логотип C: 2 перемоги;
- Логотип D: 0 перемог.

4. Ранжування варіантів:

- Логотипи A, B, C мають однакову кількість перемог, отже, необхідне додаткове оцінювання.
- Якщо необхідно, можна використовувати більше респондентів або додати вагові коефіцієнти для різних порівнянь.

5. Реалізація методу «Парного порівняння» в Python.

Реалізація коду

```
import numpy as np
```

```
# Матриця парних порівнянь (1 – переможець у парі)
comparisons = np.array([
    [0, 1, 0, 1],  # A vs. (B, C, D)
    [0, 0, 1, 1],  # B vs. (C, D)
    [1, 0, 0, 1],  # C vs. (D)
    [0, 0, 0, 0]   # D vs. (інших)
])

# Підрахунок перемог
scores = np.sum(comparisons, axis=1)
labels = ['A', 'B', 'C', 'D']

# Вивід рейтингу
ranking = sorted(zip(labels, scores), key=lambda x:
x[1], reverse=True)
print("Рейтинг логотипів:", ranking)
```

Результат

```
Рейтинг логотипів: [('A', 2), ('B', 2), ('C', 2),
('D', 0)]
```

Порівняння методів шкалування

Порівняння методів «Q-сортування» та «Парного порівняння» наведено в табл. 13.3.

Таблиця 13.3. Порівняння методів «Q-сортування» та «Парного порівняння»

Критерій	Q-сортування	Попарні порівняння
Підхід	Групування об’єктів у категорії	Порівняння кожного об’єкта з усіма іншими
Кількість об'єктів	Велика (10–50+)	Невелика (до 10–15)
Результат	Класифікація об'єктів	Чіткий рейтинг
Складність для респондента	Відносно проста	Може бути складною при великій кількості об'єктів
Застосування	Психологія, маркетинг, соціологія	Спорт, UX-дизайн, вибір стратегій

Метод Q-сортування підходить, коли потрібно розподілити об’єкти за рівнем важливості чи згоди.

Метод попарних порівнянь корисний, коли потрібно чітко визначити рейтинг невеликої кількості об’єктів.

13.1.3.1.3. Приклад кількісного контент-аналізу в Python

Приклад коду

```
from collections import Counter
import nltk
nltk.download('punkt')

text = "Економіка стабільна. Політика країни
змінюється. Економічні показники зростають."
words = nltk.word_tokenize(text.lower())

word_counts = Counter(words)
print(word_counts)
```

Вивід

```
{'економіка': 1, 'стабільна': 1, 'політика': 1,
'країни': 1, 'змінюється': 1, 'економічні': 1,
'показники': 1, 'зростають': 1}
```

13.1.3.2. Якісний контент-аналіз

Якісний контент-аналіз — це метод дослідження текстових, аудіо- та візуальних даних, який передбачає інтерпретацію змісту, смислів, контексту та прихованих значень.

В цьому випадку дослідника цікавить не стільки про що йдеться, скільки як про це кажуть.

Вимірювання в параметрах, досліджуваних в ході якісного контент-аналізу, поверхнево зачіпають сам зміст кожного повідомлення на відміну від детального й уважного обстеження, необхідного при кількісному аналізі. В результаті якісний контент-аналіз зазвичай більш простий в розробці і проведенні, а тому і більш дешевий і надійний, ніж змістовний контент-аналіз. І хоча його результати, можливо, задовольнять в меншій мірі, бо вони дають швидше начерк, ніж закінчену картину повідомлення, але при відповіді на конкретний дослідний питання вони можуть часто виявитися цілком адекватними.

Основні характеристики:

- Глибоке розуміння контексту — аналіз не лише частоти, а й значення слів та висловлювань;
- Суб'єктивність та інтерпретація — дослідник враховує приховані значення, тональність, стиль;
- Менший обсяг даних — на відміну від кількісного аналізу, тут важливіше змістове наповнення тексту.

Переваги:

- Виявлення прихованих значень і підтексту;

- Аналіз емоцій, наративів, соціальних норм;
- Використовується для глибокого вивчення окремих кейсів.

Обмеження:

- Висока суб'єктивність (інтерпретація залежить від дослідника);
- Неможливість аналізу великих обсягів тексту;
- Важко формалізувати критерії оцінки.

Кроки проведення якісного контент-аналізу

Крок 1. Формулювання дослідницького питання.: Формулювання питання типу:

«Які значення та смисли приховані в тексті?»

Наприклад:

- «Як формується образ політичного лідера у ЗМІ?»

Крок 2. Вибір джерел. Визначення джерел аналізу:

- Аналіз медіа, інтерв'ю, книг, соцмереж тощо.

Крок 3. Кодування та категоризація. Виділення ключових тем, понять, тональності тексту.

Наприклад:

- У новинах можна аналізувати емоційне забарвлення, використані метафори.

Крок 4. Аналіз смислів та контексту. Визначення наративів, підтексту, цінностей, ідеологій.

Наприклад:

- Як змінюється риторика політика в кризових ситуаціях.

Крок 5. Формулювання висновків:

- Які патерни та тенденції виявлені?
- Як автори текстів намагаються впливати на аудиторію?

Приклад аналізу якісного контент-аналізу в Python

Приклад коду

```
from textblob import TextBlob

text = "Ця книга неймовірна, вона змінила моє життя!"
blob = TextBlob(text)

print(blob.sentiment.polarity)  # Виведе значення від -1 (негатив) до 1 (позитив)
```

Вивід

```
0.8 (позитивний тон)
```

13.1.3.3. Автоматизований контент-аналіз

Автоматизований контент-аналіз — це метод аналізу текстових, аудіо- та візуальних даних, який використовує програмні алгоритми та штучний інтелект для обробки великих обсягів інформації.

Основні особливості:

- *Швидкість і масштабованість* — аналізується велика кількість текстів за короткий час;
- *Об'єктивність* — зменшується суб'єктивний вплив дослідника;
- *Можливість аналізу тональності, тем, частоти слів, стилістики*;
- *Використання NLP* — машинне навчання для розуміння змісту тексту.

Застосування автоматизованого контент-аналізу:

- *Журналістика* — моніторинг ЗМІ, аналіз тенденцій;
- *Маркетинг* — аналіз відгуків про бренди;
- *Соціологія* — дослідження суспільних настроїв;
- *Політологія* — виявлення пропаганди та маніпуляцій.

Переваги:

- Висока швидкість аналізу;
- Велика точність при правильному налаштуванні алгоритмів;
- Об'єктивність та відсутність людського фактора.

Обмеження:

- Важко аналізувати іронію, сарказм;
- Потрібне якісне налаштування алгоритмів;
- Проблеми з омонімією та контекстом.

Методи автоматизованого контент-аналізу

1. *Аналіз частоти слів та фраз*:
   - Визначення найуживаніших слів у тексті;
   - Приклад: які слова найчастіше зустрічаються в новинах про політику?
2. *Аналіз тональності* (*Sentiment Analysis*):
   - Визначення позитивного, негативного чи нейтрального тону тексту;
   - Використовується у маркетингу, соціальних мережах, політиці.
3. *Тематичне моделювання* (*Topic Modeling*)
   - Автоматичне виявлення тем у великому масиві текстів;
   - Приклад: які основні теми обговорюють у коментарях до новин?
4. *Семантичний аналіз*:
   - Визначення смислових зв'язків між словами та поняттями.
5. *Виявлення маніпуляцій і пропаганди*:
   - Аналіз стилістичних і риторичних прийомів у медіа.

Всі перераховані методи було описано в лекціях 11-12.

Приклад автоматизованого контент-аналізу тексту в Python

1. *Аналіз частоти слів*

```
from collections import Counter
```

```
import nltk
nltk.download('punkt')

text = "Політика змінюється. Політичні реформи впливають на економіку."
words = nltk.word_tokenize(text.lower())

word_counts = Counter(words)
print(word_counts)
```

Результат

```
{'політика': 1, 'змінюється': 1, 'політичні': 1, 'реформи': 1, 'впливають': 1, 'економіку': 1}
```

2. *Аналіз тональності* (*Sentiment Analysis*)

```
from textblob import TextBlob

text = "Це була чудова реформа, яка покращила життя людей!"
blob = TextBlob(text)

print(blob.sentiment.polarity)  # Від -1 (негатив) до +1 (позитив)
```

Результат

```
0.85 (позитивний тон)
```

3. *Тематичне моделювання* (*Topic Modeling*)

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

texts = [
    "Економічні реформи важливі для зростання країни.",
    "Політика впливає на соціальні реформи.",
    "Економіка та політика взаємопов'язані."
]
```

```
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(texts)

lda = LatentDirichletAllocation(n_components=2,
random_state=42)
lda.fit(X)

words = vectorizer.get_feature_names_out()
for topic_idx, topic in enumerate(lda.components_):
    print(f"Тема {topic_idx+1}: {', '.join([words[i]
for i in topic.argsort()[:-5:-1]])}")
```

Результат

```
Тема 1: політика, соціальні, реформи
Тема 2: економічні, зростання, країни
```

13.1.3.4. Порівняння кількісного, якісного та автоматизованого контент-аналізів

Порівняння видів контент-аналізу наведено в табл. 13.4.

Таблиця 13.4. Порівняння кількісного, якісного та автоматизованого контент-аналізів

Критерій	Кількісний контент-аналіз	Якісний контент-аналіз	Автоматизований контент-аналіз
Підхід	Об'єктивний, статистичний	Суб'єктивний, інтерпретативний	Автоматизований, алгоритмічний
Мета	Підрахунок частотності слів, тем, категорій	Глибоке розуміння змісту та контексту	Масштабний аналіз текстів із використанням ШІ
Інструменти	Кодування, таблиці частотності	Глибокий аналіз змісту, тематичне моделювання	NLP, машинне навчання, текстові алгоритми
Точність	Висока, але залежить від коректності кодування	Суб'єктивна, залежить від дослідника	Висока, але може помилятися в складних контекстах

Масштабованість	Середня, потребує ручного аналізу	Низька, складно аналізувати великі масиви даних	Висока, може аналізувати мільйони текстів
Застосування	Журналістика, соціологія, маркетинг	Гуманітарні науки, політичний аналіз, культурологія	Big Data, медіа-аналітика, маркетинг, аналіз

13.1.3.5. Як у західних ЗМІ змінювалося ставлення до України? Контент-аналіз 10000 заголовків

Цю інформацію нам надав проєкт «Око». «Око» за допомогою автоматизованих інструментів щодня збирає, сортує і обробляє матеріали сотень ЗМІ, в яких згадується Україна. Такі дані мають велику цінність для вчених, журналістів і всіх хто має відношення до інформаційної безпеки України. Контент-аналіз даних дозволяє визначити:

- В якому контексті ЗМІ говорять про Україну, як змінюється частота і тональність повідомлень,
- Які джерела і канали поширення новин найбільш впливові і ефективні.

Говорячи прямо – неоціненний інструмент в умовах інформаційної війни, в яку виявилася втягнутою Україна.

За останні 2,5 року в базі «Око» накопичилося близько мільйона статей на 8 мовах. Реалістично оцінюючи власні ресурси, ми обмежили вибірку даних наступними параметрами:

- Аналізувалися дані з кінця жовтня 2014 го до середини серпня 2016-го;
- З найбільш авторитетних і часто цитованих західних медіа[1];
- Аналізувалися тільки заголовки англійською мовою;
- Заголовки, що містять слово «Ukraine» або «Ukrainian».

Фрагмент даних представлено на рис. 13.1.

МЕДІА	САЙТ	КРАЇНА	ЗАГОЛОВКІВ
Reuters	reuters.com	UK	4875
Bloomberg	bloomberg.com	USA	1030
Wall Street Journal	wsj.com	USA	638
New York Times	nytimes.com	USA	558
ABC News	abcnews.go.com	USA	501
Guardian	theguardian.com	UK	492
Washington Post	washingtonpost.com	USA	450
Fox News	foxnews.com	USA	434
British Broadcasting Corporation	bbc.com	UK	423
Financial Times	ft.com	UK	415
Associated Press	ap.org	USA	278
Cable News Network	cnn.com	USA	255
USA Today	usatoday.com	USA	199
Time	time.com	USA	79

Рис. 13.1. Фрагмент вхідного датасету

Деякі результати дослідження наведено на рис. 13.2-13.4.

Рис. 13.2. Частота згадування слів «Ukraine» та «Ukrainian» в заголовках західних медіа

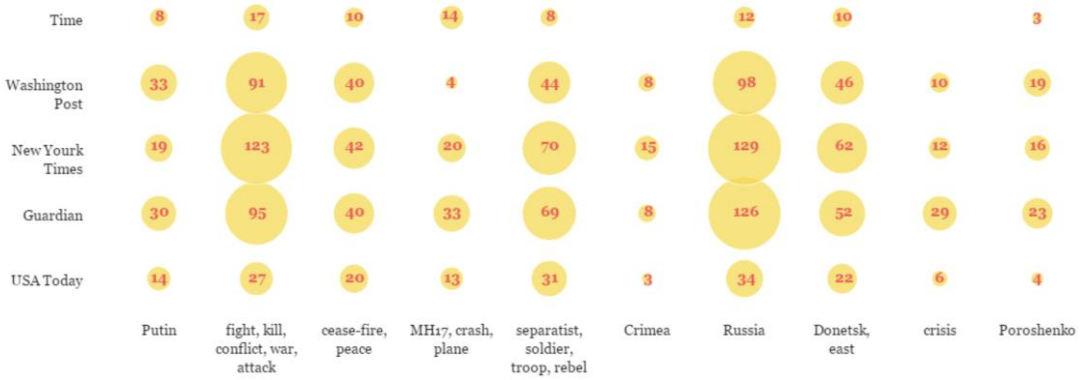


Рис. 13.3. Частота згадування тем, які стосуються України, в заголовках західних медіа

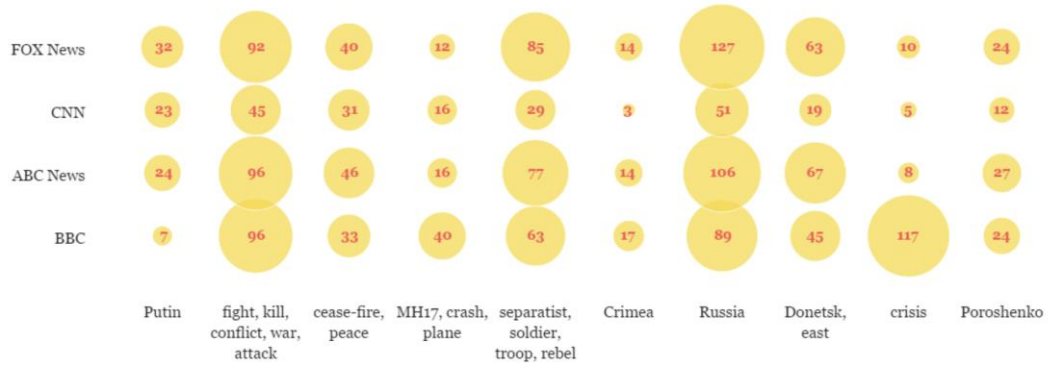


Рис. 13.4. Частота згадування тем, які стосуються України, в заголовках західних медіа (продовження)

13.2. Інтент-аналіз

13.2.1. Основні поняття

Інтенція – суб'єктивна спрямованість на певний об'єкт, активність свідомості суб'єкта. Фактично інтенція - деякий намір (приховане або явне, усвідомлюване або не усвідомлювала самим мовцем і його слухачами).

Інтент-аналіз (*Intention* – намір, мета) або аналіз намірів – теоретико-експериментальний підхід, що дозволяє шляхом вивчення публічної промови говорить виявити недоступний при використанні інших видів аналізу прихований сенс його виступів, намірів і цілей, які впливають на дискурс.

Локутівний акт – орієнтований на інформування про деяке дії, події, наприклад: Під час зустрічі зі своїм американським колегою президент сказав, що ....

Іллокутівний акт – фіксує наявність нібито певного наміри, цілі у описуваного об'єкта, яке відоме автору тексту і про який він повідомляє читачеві або слухачеві: Під час зустрічі зі своїм американським колегою президент прагнув показати, що....

Перлокутивний акт – фіксує нібито реальний результат дії описуваного об'єкта: Під час зустрічі зі своїм американським колегою президент блискуче довів, що....

Предикат – елемент простого атрибутивного судження, що позначає будь-якої ознака (властивість) його суб'єкта, або те, що говориться про суб'єкта. позначається латинською літерою *P*.

13.2.2. Загальний опис інтент-аналізу

Інтент-аналіз спрямований на інтенціональних характеристики мови, які безпосередньо співвідносяться з ходом комунікації. Метод дає досліднику можливість описати як типові, так і інші інтенції, включаючи неусвідомлювані, які сприймаються учасниками спілкування і є психологічною реальністю комунікації. Інтент-аналіз передбачає вивчення природних мовних матеріалів, які безпосередньо беруть із навколишнього життя. Наприклад, методом прихованого магнітофона, коли звучить під час розмови мова записується на плівку або інший носій інформації, а слідом за

особливою системою стенографуються: враховуються паузи, накладення реплік і інші особливості усного мовлення.

Мета Інтент-аналізу – виявлення кола інтенцій (намірів) авторів тексту (постів, промов, публіцистичних статей). Оцінювання інтенцій має проводитися кваліфікованими експертами. Оскільки стоїть завдання об'єктивно оцінити внутрішню, суб'єктивну установку людини за непрямими ознаками, необхідно залучення не менше двох експертів, в ідеалі – трьох: психолога, філолога або лінгвіста і соціолога.

Метод Інтент-аналізу по його суті є психосемантичним, розрахованим на суб'єктивне оцінювання сприймаються висловлювань. Загальна методична організація роботи полягає в послідовному, крок за кроком, оцінюванні експертом або груповий експертів авторських висловлювань обраного тексту. Оцінювання проводиться за одним і тим же критеріям, а саме: чим викликане це висловлювання, яка його цільова спрямованість, навіщо воно потрібне говорить?

Основні типи інтенцій:

- *Транзакційний* (*Transactional Intent*) – користувач хоче здійснити дію (купити, зареєструватися, завантажити);
- *Інформаційний* (*Informational Intent*) – шукає інформацію («Як зробити ... », «Що таке ... »);
- *Навігаційний* (*Navigational Intent*) – шукає конкретний сайт або сервіс («*Facebook*», «Вхід в *Gmail*»);
- *Комерційний* (*Commercial Investigation*) – досліджує перед покупкою («Найкращі смартфони 2025»).

На рис. 13.5 наведено діаграму аналізу інтенцій у NLP. Вона показує основні категорії намірів користувача: інформаційний, навігаційний, транзакційний і комерційний, разом із прикладами.

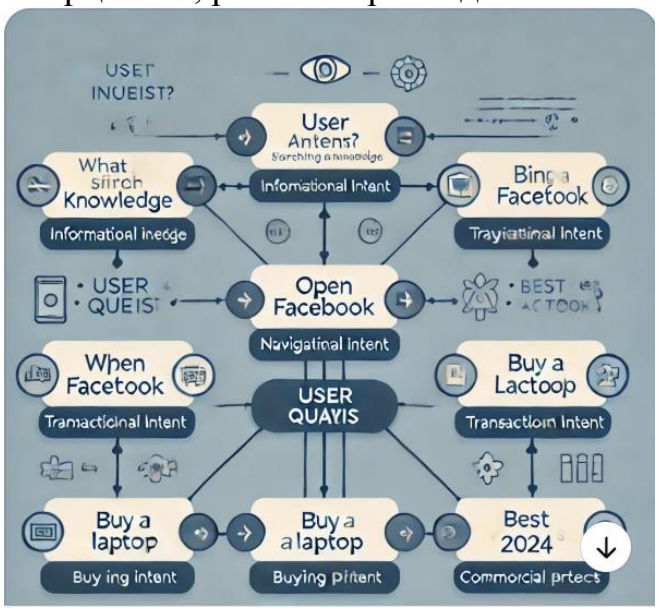


Рис. 13.5. Графічна діаграма інтенцій

Плюси методу:

- Варіативність інтерпретацій тексту в залежності від мети аналізу і характеру документів;
- Дозволяє максимально повно розкривати систему суб'єктивних смислів, виражених в різних видах оповідання (наративах), зокрема, в текстах;
- Підвищує якість відбору інформації і документів;
- Дозволяє дивитися на смисловий зміст тексту під різними кутами.

Мінуси методу:

- Складність кваліфікації інтенцій;
- Суб'єктивізм аналізу зважаючи на присутність людського чинника;
- Суб'єктивність методики Інтент-аналізу також призводить до необхідності створення множини словників інтенцій і великої кількості варіацій уявлення вивідного знання, а це ускладнює процес уніфікації методики;
- Неправильне розуміння і трактування змісту.

Використання інтент-аналізу:

- *SEO та реклама* – для розуміння намірів користувачів у пошуку;
- *Чат-боти* – для правильної відповіді на запити;
- *Клієнтська підтримка* – визначення проблем клієнтів;
- *Соціальні мережі* – моніторинг настроїв і потреб.

13.2.3. Історія і зміст методу

Метод є досить молодим і був розроблений Лабораторією психології мови і психолінгвістики і застосовується для з'ясування внутрішнього сенсу політичних агітаційних текстів.

Метод заснований на ідеї, що дорослі люди пройшли соціалізацію, при вступі в спілкування зазвичай не стільки простодушно висловлюють те, що думають, скільки вибудовують свою промову відповідним чином і прагнуть до досягнення цілей, які часом бувають складними, іноді ховаються мовцем і для надання бажаного впливу ховаються за показними намірами (інтенціями). Інтенціональних спрямованості говорить складаються в досить складну структуру і спрямовані на здійснення впливу на слухачів. Причому інтенції мають різноманітні види прояви, що не заважає їм бути влаштованими за законами мови так, щоб слухачі змогли сприйняти їх загальну спрямованість, а також відчути в висловлюваннях різний емоційний настрій – схвалення, осуд, загрозу і т.п. Вивчення інтенціональних форм надає можливість знайти ключ до розуміння способів словесних впливів і мотивів. Тому, особливо у випадках важливих усних або письмових виступів, добре знання правил интенционального побудови вербальних висловлювань надасть значну допомогу автору тексту і дискурсу. Метод Інтент-аналізу за своєю природою є психосемантичних, а значить розрахованим на суб'єктивну оцінку з боку сприймаються висловлювань. Методична організація роботи полягає в покроковому оцінюванні експертами (одним або декількома) авторських висловлювань,

взятих з деякого обраного тексту. Критерії оцінювання на всьому протязі дослідження є незмінними: що викликало те чи інше висловлювання, в чому полягає його цільова спрямованість, навіщо воно взагалі необхідно говорить? У той же час потрібно розуміти, що методика не позбавлена суб'єктивності, тому потребує ретельного опрацювання. Так найкраще вести дослідження групою, до складу якої входять три-чотири людини-експерта. Це дозволить в разі якихось розбіжностей проводити узгоджені поправки їх суджень, а також отримувати додаткові факти про зонах нерозрізненості або важкою розрізнення інтенцій. Гарною підмогою може стати оперативний словник інтенцій, який складається на основі ясних випадків. У деяких випадках корисно переформулювати аналізовані висловлювання з обов'язковим збереженням сенсу оригіналу.

13.2.4. Процедури інтент-аналізу

Послідовність операцій при виконанні Інтент-аналізу строго конкретна.

Основні кроки інтент-аналізу

Крок 1. Експерти повинні розділити аналізований текст на фрагменти, в кожному з яких міститься інтенція. Якщо інтенція не очевидна, експерти Переформулюю відрізок тексту у відповідності з наступними правилами: при максимальному збереженні сенсу вихідний текст викладається гранично коротко. Повинна бути максимально збережена лексика оригіналу, не можна вводити асоціативні, що розширюють зміст фрагмента вираження (ніяких узагальнень!); окремі мовні фрагменти, що містять найбільш важливі ідеї, виводяться для окремої оцінки; другорядні уточнюючі вираження і характеристики не враховуються; при повторі в тексті думки в протоколі просто згадується повторення: лат. – id $(Idem)$ або англ. – ib $(Ibidem)$.

Крок 2. Експерти повинні кваліфікувати інтенції, визначити їх вид. Існує багато варіантів інтенцій: звинувачення, критика, протистояння, демонстрація сили, похвала, схвалення, відведення звинувачень, кооперація, самопрезентація, самокритика, самовиправдання, відмова в проханні і т.д.

Крок 3. В рамках цього кроку визначаються об'єкти, зазначені в тексті, і інтенції, що відносяться до них; оцінюється структура інтенціональних блоків, що відносяться до згадуваним об'єктів; розраховується частота інтенцій в тексті (у відсотках від усіх виявлених інтенцій); оцінюються форми вираження інтенцій в мові (зокрема, при активному використанні в тексті дієслів визначають тип мовного акту – локутівний, іллокутівний або перлокутивно); оцінюється зміна позитивних і негативних інтенцій в тексті в цілому (у вигляді графіка, який фіксує зміну позитивних, нейтральних і негативних інтенцій в тексті).

Крок 4. При порівнянні декількох текстів одного автора дається уявлення інтенціональних складових тексту у вигляді ментальних карт (проводиться

так зване згортання тексту). На цьому етапі проводять визначення кола обговорюваних об'єктів на основі їх значущості. В аналіз включають ті об'єкти, до яких відносилося не менше двох дескрипторів (ознак, що описують об'єкт). В окремих випадках можна враховувати і об'єкти, до яких відноситься один дескриптор; експертну кодифікацію дескрипторів за двома параметрами – за оцінкою (хороший-поганий, позитивна-негативна; приписуються оцінки $+1, 0, -1$), по динамізму (сильний, енергійний – слабкий, пасивний; приписуються оцінки $+1, 0, -1$) . Потім визначаються інтегральні значення кожного об'єкта за вказаними параметрами.

Крок 5. На завершальному етапі проводиться вербальна інтерпретація отриманих результатів.

13.2.5. Методики Інтент-аналізу

Оскільки метод Інтент-аналізу в більшій мірі заснований на експертній оцінці тексту, його повна автоматизація не здійснено, однак, існують методики, де автоматизовані перші етапи Інтент-аналізу, що дозволяють здійснювати операції виділення тим, кодування і пошуку.

Методи інтент-аналізу тексту

1. *Методика Ethnograph.*

Ethnograph – методика призначена для якісного (змістовного) аналізу даних інтерв'ю, фокус-груп, щоденників тощо.

Ethnograph – це програмне забезпечення для якісного аналізу текстових даних, що використовується у соціології, антропології, психології та маркетингових дослідженнях. Програма допомагає дослідникам кодувати, категоризувати та аналізувати великі обсяги текстової інформації, отриманої з інтерв'ю, відкритих відповідей у опитуваннях, польових нотаток тощо.

Процес роботи з даними побудований на основі того, що проводиться пошук необхідних даних, відбір результатів пошуку і аналіз всього відібраного матеріалу. Для кожного з ланок цього процесу передбачений відповідний набір процедур.

Основні функції Ethnograph:

- *Кодування тексту* – автоматичне чи ручне маркування ключових слів і фраз;
- *Пошук і фільтрація* – знаходження шаблонів і тенденцій у тексті;
- *Візуалізація* – створення графіків і таблиць для аналізу даних;
- *Експорт даних* – інтеграція з іншими програмами (наприклад, SPSS, NVivo).

2. *Методика Leximancer.*

Leximancer – методика, спрямована на виявлення ключових тем (key-themes), концептів (concepts) в електронних документах. На відміну від інших аналогічних методик, в своїй основі з'єднує багато підходів, таких як обчислювальна лінгвістика, контент-аналіз, інформатика, фізика, теорія

мереж тощо. Дозволяє наочно представляти результати аналізу і обчислень у вигляді карти концептів і різноманітних таблиць і діаграм.

Leximancer – це програмне забезпечення для автоматизованого семантичного аналізу тексту, яке використовується у наукових, маркетингових та бізнес-дослідженнях.

Ця система використовує машинне навчання та статистичний аналіз, щоб автоматично ідентифікувати ключові теми та концепції у текстових даних без попереднього кодування вручну.

Основні можливості Leximancer:

- Автоматичне виявлення тем і зв'язків між ними;
- Візуалізація даних у вигляді концептуальних карт;
- Аналіз тональності та контексту тексту;
- Порівняння різних текстових корпусів;
- Імпорт даних з різних джерел (PDF, $DOCX$, CSV тощо).

3. *Методика Minnesota Contextual Content Analysis.*

Minnesota Contextual Content Analysis (*MCCA*) – дозволяє здійснювати контекстуальний аналіз (аналіз слів в просторі 4 соціальних контекстів: практика, традиції, емоції, аналіз) і аналіз основних думок і образів, розкритих в тексті. Для кожного виду аналізу розроблені і стандартизовані норми. Крім зазначених основних функцій дозволяє виводити статистику, проводити аналіз слів, частотний аналіз слів і категорій і т.д.

Minnesota Contextual Content Analysis – це метод кількісного та якісного аналізу тексту, розроблений в Університеті Міннесоти. Його основна мета – ідентифікація контекстуального значення слів і фраз у текстах, що використовуються в соціальних науках, психології, політичному аналізі та дослідженнях медіа.

Основні особливості MCCA:

- *Аналіз контексту слів* – вивчення не лише частоти, а й семантичного значення;
- *Кодування тексту* – автоматизований розподіл текстових фрагментів за категоріями;
- *Структурний аналіз* – визначення зв'язків між поняттями та темами;
- *Статистична обробка* – використання методів кількісного аналізу для оцінки тексту.

Порівняння методів інтент-аналізу тексту

Порівняння розглянутих методів наведено в табл. 13.5.

Таблиця 13.5. Порівняння MCCA, Leximancer та Ethnograph

Критерій	MCCA (Multiple Correspondence Cluster Analysis)	Leximancer	Ethnograph
Основне призначення	Кластерний аналіз тексту на основі множинної	Автоматизований семантичний аналіз тексту	Кодування та якісний аналіз тексту

	кореспонденційної класифікації		
Метод аналізу	Статистичний підхід: кореспонденційний аналіз + кластеризація	Виявлення концептів та їх зв'язків	Ручне кодування та тематичний аналіз
Автоматизація	Частково автоматизований	Повністю автоматизований	Переважно ручний
Візуалізація	Двовимірні карти кластерів	Концептуальні мапи зв’язків між поняттями	Текстові категоризації
Глибина аналізу	Виявляє приховані закономірності у великих текстових масивах	Визначає взаємозв’язки між концептами та їхню вагу	Поглиблений аналіз змісту через ручне кодування
Область застосування	Соціальні науки, маркетинг, аналіз відгуків	Бізнес-аналітика, соціальні науки, автоматизований аналіз текстів	Етнографічні дослідження, психологія, антропологія
Гнучкість	Гнучке налаштування параметрів кластеризації	Автоматичне створення концептів без потреби налаштування	Дуже гнучкий, але вимагає більше часу
Вимоги до користувача	Потребує знань у статистичному аналізі	Інтуїтивно зрозумілий інтерфейс	Вимагає якісного розуміння якісних методів дослідження
Переваги	Висока точність статистичного аналізу, добре працює з великими масивами даних	Автоматизований аналіз, швидка обробка текстів	Дає найглибше розуміння контексту, оскільки аналіз проводиться вручну
Недоліки	Потребує глибоких знань статистики та кластерного аналізу	Обмежені можливості редагування результатів	Трудомісткий процес, потребує

			більше часу для аналізу

MCCA – найкращий для контекстуального аналізу у соціологічних, політичних та медіа-дослідженнях.

Leximancer – найзручніший для автоматизованого аналізу тем у великих текстових масивах, зокрема у маркетингових дослідженнях.

Ethnograph – підходить для глибокого якісного аналізу в соціальних науках, але потребує багато ручної роботи.

Якщо потрібно швидко знайти основні теми у тексті – Leximancer.

Якщо важливий контекст використання слів – MCCA.

Якщо потрібно глибоко аналізувати текст вручну – Ethnograph.

13.2.6. Приклад реалізації інтент-аналізу в Python

Приклад коду

```
import spacy

# Завантажуємо модель NLP (англійська)
nlp = spacy.load("en_core_web_sm")

# Функція аналізу намірів
def detect_intent(text):
    doc = nlp(text.lower())

    if "buy" in text or "order" in text:
        return "Транзакційний"
    elif "how" in text or "what" in text:
        return "Інформаційний"
    elif "site" in text or "official" in text:
        return "Навігаційний"
    else:
        return "Невизначений"

# Приклад запитів
queries = ["Where to buy an iPhone?", "How to reset
Instagram password?", "Official Apple website"]
for query in queries:
    print(f"'{query}' → {detect_intent(query)}")
```

Результат

```
'Where to buy an iPhone?' → Транзакційний
'How to reset Instagram password?' → Інформаційний
'Official Apple website' → Навігаційний
```

13.3. Рекомендаційні системи

Рекомендаційні системи – це алгоритми, що аналізують дані про користувачів і надають персоналізовані рекомендації. Вони широко використовуються в онлайн-магазинах (Amazon), стрімінгових сервісах (Netflix, Spotify), соцмережах (YouTube, TikTok), новинних платформах тощо.

13.3.1. Основні типи рекомендаційних систем

Рекамендаційні системи базуються на наступних методах:

1. Колаборативна фільтрація;
2. Контентна фільтрація;
3. Тегова фільтрація.

Зупинимося докладніше на кожному з них.

13.3.1.1. Колаборативна фільтрація

Колаборативна фільтрація (*Collaborative Filtering*, *CF*) – це метод рекомендацій, який базується на поведінці користувачів та їхніх уподобаннях. Вона передбачає, що якщо двоє людей мають схожі інтереси в минулому, то вони, ймовірно, матимуть їх і в майбутньому.

Де використовується?

- Netflix, YouTube, Spotify – рекомендує контент на основі переглядів інших користувачів;
- Amazon, eBay, AliExpress – радить товари, які купували схожі користувачі. LinkedIn, Facebook, TikTok – рекомендує друзів, сторінки та контент.

Проблеми колаборативної фільтрації:

- *Проблема холодного старту* – якщо користувач або об'єкт новий, немає достатньо даних для рекомендацій;
- *Розріджена матриця* – більшість користувачів оцінюють лише невелику частину об'єктів;
- *Популярність vs. Новизна* – алгоритми можуть перевагу надавати популярним об’єктам, ігноруючи нові.

Проблема холодного старту

Проблема холодного старту (*Cold Start Problem*) – це ситуація, коли рекомендаційна система не має достатньо даних про нових користувачів або нові об'єкти (фільми, товари, музику тощо), через що не може робити точні рекомендації.

Види холодного старту:

- *Новий користувач* – немає історії оцінок, тому система не знає його вподобань;
- *Новий об'єкт (товар, фільм, пісня)* – немає оцінок від користувачів, тому система не знає, кому його рекомендувати;
- *Нова система* – якщо система тільки запущена, у ній немає ні оцінок, ні взаємодій.

Методи вирішення проблеми холодного старту.

1. *Гібридні рекомендаційні системи*:

- Поєднання колаборативної фільтрації (аналіз схожих користувачів) та контентного підходу (аналіз характеристик товарів – контент аналіз).

Приклад:

- Якщо у нових користувачів немає оцінок, система аналізує вік, стать, місцезнаходження, інтереси;
- Якщо з'явився новий фільм, система рекомендує його тим, хто переглядав подібні жанри.

Netflix використовує гібридну модель: якщо немає оцінок, вона аналізує метаінформацію про фільм (режисера, жанр тощо).

2. *Використання демографічних даних*:

- Запит початкової інформації про користувача (вік, стать, улюблений жанр) під час реєстрації;
- Групування користувачів на основі демографічних характеристик.

Spotify на старті запитує у користувача улюблених виконавців, щоб сформувати первинні рекомендації.

3. *Популярні об'єкти для нових користувачів*:

- Рекомендація найпопулярніших товарів/фільмів/книг для нових користувачів;
- Використовується в e-commerce (Amazon, AliExpress), де рекомендують бестселери.

YouTube для нових користувачів показує трендові відео, доки вони не почнуть взаємодіяти з контентом.

4. *Прогнозування через контентні атрибути*:

- Якщо є новий об'єкт (фільм, книга), його характеристики (жанр, автор, актори) можна використовувати для рекомендацій;
- Використовується у контентних фільтраційних моделях.

Goodreads рекомендує книги на основі жанру, автора, схожих попередніх виборів.

5. *Active Learning (запит первинних оцінок)*:

- Користувачеві дають оцінити кілька об'єктів на старті;
- Це допомагає системі швидко отримати первинні дані.

Netflix на початку просить оцінити кілька фільмів, щоб покращити персоналізацію.

Основні підходи колаборативної фільтрації

1. *User-Based Collaborative Filtering (Фільтрація на основі користувачів)*:

- Порівнює користувачів, щоб знайти тих, хто має подібні оцінки або вподобання;
- Якщо користувач A і B обидва люблять певні фільми, і A також подобається інший фільм, можливо, він сподобається і B.

Розглянемо наступний приклад:

- Якщо Петро і Марина поставили високі оцінки однаковим серіалам, а Петро ще дивився «$Médicos, línea\ de\ vida$», то Марині цей серіал також можуть рекомендувати.

Формула обчислення схожості між користувачами.

Подібність між користувачами визначається за допомогою косинусної відстані (13.1):

$$cos(A,B) = \frac{(A \cdot B)}{||A|| \cdot ||B||}, \quad (13.1)$$

яка також була представлена у лекції 6.

2. *Item-Based Collaborative Filtering* (*Фільтрація на основі об'єктів*):

- Аналізує схожість між об'єктами (товарами, фільмами, книгами);
- Якщо два товари отримали схожі оцінки від багатьох користувачів, вони вважаються схожими.
- Розглянемо наступний приклад:
- Якщо багато людей, які купили «$iPhone\ 16$», також купили «$AirPods$», то наступному користувачеві, який шукає iPhone 16, можуть запропонувати AirPods.

Формула схожості між об'єктами.

Подібність між об’єктами також визначається за допомогою косинусної відстані (13.1)

Методи реалізації

1. *Методи на основі пам'яті* (*Memory-based*):

- Використовують усю матрицю оцінок користувачів та об'єктів;
- Вимагають багато ресурсів для обчислення.

Матриця оцінок

Матриця оцінок A (*rating matrix*) – це таблиця, яка містить оцінки користувачів для різних об'єктів (товарів, фільмів, книг тощо). Вона є основою для алгоритмів колаборативної фільтрації.

Матриця оцінок (табл. 13.6) складається з:

- Рядків – користувачі ($User$);
- Стовпців – об'єкти ($Item$);
- Значень – оцінки ($Rating$), які користувачі ставлять об'єктам.

Таблиця 13.6. Приклад матриці оцінок

$User/Item$	Фільм A	Фільм B	Фільм C	Фільм D
Анна	5	?	3	?
Богдан	4	2	?	5
Олег	?	3	4	2
Катя	3	4	2	?

В табл. 13.6 «?» – відсутні значення, тобто користувач не поставив оцінку.

2. *Методи на основі моделі* (*Model-based*):

- Використовують машинне навчання (SVD, матричну факторизацію, нейронні мережі);

- Менш чутливі до «холодного старту» та масштабуються краще.

Метод SVD

SVD (*Singular Value Decomposition*, *сингулярне розкладання матриці*) – це метод лінійної алгебри, який розкладає матрицю A на три інші матриці:

$$A = U \cdot \Sigma \cdot V^T, \quad (13.2)$$

де:

A – вихідна матриця (наприклад, матриця оцінок у рекомендаційній системі);

U – матриця лівих сингулярних векторів (представляє користувачів);

Σ – діагональна матриця сингулярних значень (ваги важливості);

V^T – транспонована матриця правих сингулярних векторів (представляє об'єкти).

Матрична факторизація

Матрична факторизація – це метод у машинному навчанні та лінійній алгебрі, який розкладає велику матрицю на добуток менших матриць, щоб знайти приховані зв’язки між даними. Вона широко використовується в рекомендаційних системах для прогнозування відсутніх оцінок у матриці користувачів та об’єктів.

Основна ідея.

Припустимо, що є матриця оцінок A (табл. 13.6), де користувачі ставлять оцінки фільмам. Але ця матриця розріджена (багато пропущених значень).

Завдання: передбачити пропущені оцінки.

Матрична факторизація розкладає цю матрицю на дві менші матриці:

- *Матриця користувачів U* (*user embeddings*) – матриця, що представляє переваги кожного користувача.
- *Матриця об'єктів V* (*item embeddings*) – матриця, що описує характеристики об'єктів (фільмів, товарів тощо).

Формула факторизації має вигляд (13.3):

$$A \approx U \cdot V^T, \quad (13.3)$$

де:

A – вихідна матриця оцінок (користувач × об'єкт);

U – матриця характеристик користувачів $(m \times k)$;

V – матриця характеристик об'єктів (n×k);

k – кількість прихованих факторів (гіперпараметр).

Приклад факторизації. Розклад матриці A:

$$\begin{pmatrix} 5 & ? & 3 & ? \\ 4 & 2 & ? & 5 \\ ? & 3 & 4 & 2 \\ 3 & 4 & 2 & ? \end{pmatrix} \approx \begin{pmatrix} 0{,}8 & 0{,}1 \\ 0{,}5 & 0{,}7 \\ 0{,}2 & 0{,}9 \\ 0{,}4 & 0{,}6 \end{pmatrix} \cdot \begin{pmatrix} 0{,}9 & 0{,}3 & 0{,}5 & 0{,}2 \\ 0{,}4 & 0{,}8 & 0{,}7 & 0{,}6 \end{pmatrix}^T.$$

Після множення цих двох матриць отримуємо наближену матрицю A' із заповненими пропущеними оцінками.

Метод SVD є одним з методів матричної факторизації.

Реалізація матричної факторизації в Python:

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate

# Завантажуємо стандартний датасет (MovieLens)
data = Dataset.load_builtin('ml-100k')

# Ініціалізуємо SVD-модель
model = SVD()

# Оцінюємо якість (RMSE, MAE)
cross_validate(model, data, cv=5, verbose=True)
```

13.3.1.2. Контентна фільтрація

Контентна фільтрація (*Content-Based Filtering*) – це метод персоналізованих рекомендацій, який аналізує характеристики об'єктів (фільмів, книг, товарів) та порівнює їх із вподобаннями користувача.

Плюси:

- Не залежить від інших користувачів (тільки вподобання одного);
- Добре працює, якщо користувач має історію взаємодій;
- Можна рекомендувати малопопулярні об'єкти.

Мінуси:

- Проблема холодного старту (якщо користувач новий, немає інформації про його вподобання);
- Потрібна деталізована інформація про об'єкти;
- Не враховує поведінку інших користувачів.

Основні кроки контентної фільтрації

Крок 1. Збір інформації:

- Користувач взаємодіє з товарами (ставить оцінки, дивиться фільми, слухає музику);
- У кожного товару є атрибути (жанр, актори, автор, опис, ключові слова).

Крок 2. Створення профілю користувача:

- Аналізуються раніше вподобані об'єкти;
- Формується вектор вподобань (які жанри, ключові слова, характеристики користувач частіше обирає).

Крок 3. Обчислення схожості:

- Використовуються алгоритми на основі косинусної подібності (13.1), метод TF-IDF (аналіз ключових слів) або розширене семантичне порівняння, де враховує не лише жанри, а й актора, режисера, стиль тощо;

- Порівнюється профіль користувача з характеристиками нових об'єктів.

Крок 4. Рекомендація:

- Об'єкти з найбільшою подібністю рекомендуються користувачеві.

Приклад роботи

Нехай необхідно сформувати рекомендації фільмів. Обчислимо схожість між жанрами (табл. 13.7).

Таблиця 13.7. Схожість за допомогою TF-IDF

Фільм	Жанри	TF-IDF
Фільм *A*	Драма, Мелодрама	(0,7; 0,8; 0,1)
Фільм *B*	Бойовик, Трилер	(0,2; 0,9; 0,6)
Фільм *C*	Драма, Історичний	(0,8,0,6; ,0,3)

Якщо користувач раніше дивився драми, система знайде найбільш схожий фільм за жанрами.

Реалізація контентної фільтрації в Python (TF-IDF + Косинусна подібність)

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.metrics.pairwise import
cosine_similarity

# Опис фільмів
descriptions = [
    "Драма про кохання та втрати",
    "Напружений бойовик з погонями",
    "Історична драма з битвами"
]

# Векторизація TF-IDF
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(descriptions)

# Обчислення подібності
similarity_matrix = cosine_similarity(tfidf_matrix)

print(similarity_matrix)  # Видасть схожість між
фільмами
```

Порівняння методів колаборативної та контентної фільтрації

Порівняння методів наведено в табл. 13.8.

Таблиця 13.8. Переваги та недоліки колаборативної та контентної фільтрації

Метод	Переваги	Недоліки
Колаборативна фільтрація	Добре працює без знання характеристик товару	Проблема холодного старту (немає даних для нових користувачів)
Контентна фільтрація	Не залежить від інших користувачів	Потрібна детальна інформація про товари

13.3.1.3. Тегова фільтрація

Тегова фільтрація – частковий випадок контентної фільтрації, де система використовує ключові теги (жанр, тематика, автор тощо) для пошуку схожих об'єктів.

Отже, тегова фільтрація – це один із можливих методів контентної фільтрації.

Основні кроки тегової фільтрації

Крок 1. Кожен об'єкт має набір тегів (наприклад, жанри, характеристики, ключові слова).
Крок 2. Визначається, які теги користувач обирав або оцінював найчастіше.
Крок 3. Система рекомендує об'єкти з найбільш схожим набором тегів.

Приклад рекомендації фільмів

Нехай необхідно сформувати рекомендації фільмів. Визначимо жанрові теги для фільмів (табл. 13.9).

Таблиця 13.9. Жанрові теги до фільмів

Фільм	Теги
Фільм A	Драма, Кохання
Фільм B	Бойовик, Пригоди
Фільм C	Драма, Історія

Якщо користувач дивився Фільм A, йому можуть рекомендувати Фільм C, бо він має схожий тег «Драма».

Приклад реалізації тегової фільтрації в Python

```
from collections import defaultdict

# База даних фільмів та їх тегів
movies = {
    "Фільм A": {"драма", "кохання", "історія"},
    "Фільм B": {"бойовик", "пригоди", "екшн"},
    "Фільм C": {"драма", "історія", "біографія"},
    "Фільм D": {"фантастика", "бойовик", "пригоди"},
    "Фільм E": {"драма", "кохання", "музика"}
}
```

```
# Функція обчислення схожості за тегами
def get_similar_movies(target_movie, movies_db):
    target_tags = movies_db.get(target_movie, set())
    similarity_scores = {}

    for movie, tags in movies_db.items():
        if movie == target_movie:
            continue  # Не порівнюємо з самим собою

        # Обчислення коефіцієнта Жаккара (Jaccard
Similarity)
        intersection = len(target_tags & tags)
        union = len(target_tags | tags)
        similarity = intersection / union if union !=
0 else 0
        similarity_scores[movie] = similarity

    # Сортування за схожістю
    sorted_movies = sorted(similarity_scores.items(),
key=lambda x: x[1], reverse=True)

    return sorted_movies

# Обираємо фільм для рекомендацій
target_movie = "Фільм A"
recommendations = get_similar_movies(target_movie,
movies)

# Виводимо результат
print(f"Рекомендовані фільми для '{target_movie}':")
for movie, score in recommendations:
    print(f"{movie}: схожість {score:.2f}")
```

Відмінності контентної та тегової фільтрації

Порівняння контекстної та тегової фільтрації наведено в табл. 13.10.

Таблиця 13.10. Відмінності контентної та тегової фільтрації

Характеристика	Контентна фільтрація	Тегова фільтрація
Джерело даних	Опис об'єкта, ключові слова, семантичний аналіз	Наперед визначені теги (жанри, автори, теми)

Методи	TF-IDF, Word Embeddings, косинусна подібність	Просте порівняння збігу тегів
Гнучкість	Аналізує контент на глибшому рівні	Обмежена вибором тегів
Ширина охоплення	Може знаходити схожість навіть між об'єктами без однакових тегів	Тільки за збігом тегів

Тегова фільтрація працює швидше, але менш гнучка, ніж контентна фільтрація, яка може аналізувати глибший зміст.

ЛЕКЦІЯ 14

АНАЛІЗ ДАНИХ В СОЦІАЛЬНИХ МЕРЕЖАХ

14.1. Маркетинг в соціальних мережах

14.1.1. Загальний опис

Маркетинг в соціальних мережах (*Social Media Marketing, SMM*) – це повноцінний маркетинг, а не тільки просування через різні соціальні платформи. Social Management є частиною маркетингової і комунікаційної стратегії.

Це комплекс заходів щодо використання соціальних медіа в якості каналів для просування компаній або бренду і рішення інших бізнес-завдань. Marketing в абревіатурі недостатньо точне слово, так як під ним мається на увазі просування, яке входить в комплекс маркетингу. Тобто, більш точну назву – просування в соціальних мережах від англ. Social media promotion (SMP). По-простому – це комунікація з майбутнім споживачем через соціальні мережі.

Основний упор робиться на створенні повідомлення (текстового або візуального), яке люди будуть поширювати через соціальні мережі самостійно, вже без участі організатора. Вважається, що повідомлення, що передаються по соціальним мережам, викликають більше довіри у потенційних споживачів товару або послуги. Це пов'язується з рекомендаційної схемою поширення в соціальних медіа за рахунок соціальних зв'язків, що лежать в основі взаємодії.

Просування в соціальних мережах дозволяє точково впливати на цільову аудиторію, вибирати майданчики, де ця аудиторія більшою мірою представлена, і найбільш підходящі способи комунікації з нею, при цьому в найменшій мірі зачіпаючи незацікавлених в цій рекламі людей.

Важливо відзначити, що просування в соціальних мережах застосовується не тільки на товари і послуги. Активно використовують дану технологію засоби масової інформації. Вони створюють свої облікові записи в соціальних мережах, розміщують свій контент і тим самим збирають передплатників (читачів свого продукту).

Основними завданнями маркетингу в соціальних мережах вважаються брендинг (просування бренду), підвищення лояльності аудиторії і популярності, PR і збільшення відвідуваності сайтів різних компаній.

Інструментами SMM є ведення блогу в соціальних мережах, інформаційні повідомлення в різних спільнотах, спілкування в коментарях, робота з форумами, прихований маркетинг, пряма реклама і вірусний маркетинг, моніторинг позитивного і негативного фону, оптимізація медіапростору.

Не слід плутати SMM з соціальним маркетингом.

SMM може починатися в офлайн, а потім закріплюватися і поширюватися онлайн. Маркетинг в соціальних мережах включає в себе багато методів роботи. Найпопулярніші з них – це побудова спільнот бренду (створення представництв компанії в соціальних медіа), робота з блогосферою, репутаційний менеджмент, персональний брендинг і нестандартні інструменти.

SMM – процес дуже динамічний, тому потрібно постійно стежити за мінливими інтересами аудиторії і появою нових трендів.

14.1.2. Платформи для SMM

Основні платформи для SMM наведено на рис. 14.1.

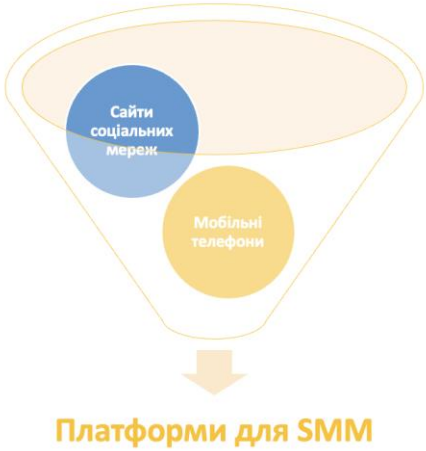


Рис. 14.1. Основні платформи для SMM

14.1.2.1. Сайти соціальних мереж

Веб-сайти соціальних мереж дозволяють окремим особам, підприємствам та іншим організаціям взаємодіяти один з одним і будувати відносини і спільноти в Інтернеті. Коли компанії приєднуються до цих соціальних каналах, споживачі можуть взаємодіяти з ними безпосередньо. Ця взаємодія може бути більш особистим для користувачів, ніж традиційні методи зовнішнього маркетингу та реклами. Сайти соціальних мереж діють як сарафанне радіо або, точніше, електронне сарафанне радіо. Здатність Інтернету охопити мільярди у всьому світі дала онлайн-уст потужний голос і далеке поширення. Здатність швидко змінювати моделі покупок, а також придбання та діяльність продуктів або послуг для зростаючого числа споживачів визначається як мережа впливу. Соціальні мережі та блоги дозволяють передплатникам ретвітів або публікувати коментарі, зроблені іншими людьми про продукт, що просувається, що досить часто зустрічається на деяких сайтах соціальних мереж. Повторюючи повідомлення, призначені для користувача підключення можуть бачити лист, що дозволяє охопити більше людей. Оскільки інформація про продукт розміщується там, отримує повторення і більше трафіку надходить на продукт / компанію.

Соціальні мережі засновані на створенні віртуальних спільнот, які дозволяють споживачам виражати свої потреби, бажання і цінності в

Інтернеті. Маркетинг в соціальних мережах потім пов'язує цих споживачів і аудиторію з підприємствами, які мають однакові потреби, бажання і цінності. Через сайти соціальних мереж компанії можуть підтримувати зв'язок з окремими передплатниками. Це особиста взаємодія може прищепити почуття лояльності послідовникам і потенційним клієнтам. Крім того, вибираючи, за ким слідувати на цих сайтах, продукти можуть досягти дуже вузької цільової аудиторії. Сайти соціальних мереж також містять багато інформації про те, які продукти і послуги можуть зацікавити потенційних клієнтів. За допомогою нових технологій семантичного аналізу маркетологи можуть виявляти сигнали про покупку, такі як контент, яким діляться люди, і питання, розміщені в Інтернеті. Розуміння сигналів покупки може допомогти продавцям орієнтуватися на релевантних потенційних клієнтів, а маркетологи проводити мікро-цільові кампанії.

У 2014 році більше 80% керівників підприємств вказали, що соціальні мережі є невід'ємною частиною їхнього бізнесу. Бізнес-рітейлери збільшили свої доходи від маркетингу в соціальних мережах на 133%.

14.1.2.2. Мобільні телефони

Понад три мільярди осіб у світі активні в Інтернеті. За минулі роки Інтернет постійно набирав все більше і більше користувачів, збільшившись з 738 мільйонів гривень на 2000 року до 3,2 мільярда у 2015 році. Приблизно 81% нинішнього населення в Сполучених Штатах мають профіль в будь-якої соціальної мережі, який вони часто використовують. Застосування мобільних телефонів дуже вигідно для маркетингу в соціальних мережах, тому що ці пристрої дозволяють людям миттєво переглядати веб-сторінки і отримувати доступ до сайтів соціальних мереж. Використання мобільних телефонів росло швидкими темпами, що докорінно змінило процес покупки, дозволяючи споживачам легко отримувати інформацію про ціни і продуктах в режимі реального часу і дозволяючи компаніям постійно оновлювати своїх передплатників і нагадувати їм про вигідні пропозиції та новинки. Багато компаній в даний час розміщують QR-коди (Quick Response) разом з продуктами для приватних осіб, щоб отримати доступ до веб-сайту компанії або онлайн-сервісів з своїх смартфонів. Роздрібні продавці використовують QR-коди для полегшення взаємодії споживачів з брендами, пов'язуючи код з веб-сайтами брендів, рекламними акціями, інформацією про продукти або будь-яким іншим контентом з підтримкою мобільних пристроїв. Крім того, використання торгу в реальному часі в індустрії мобільної реклами є високим і зростаючим через його цінності для перегляду веб-сторінок на ходу. У 2012 році Nexage, постачальник торгів в реальному часі для мобільної реклами, повідомляв про 37%-ве збільшення виручки щомісяця. Adfonic, ще одна платформа для публікації мобільної реклами, повідомила про збільшення запитів на 22 мільярди в тому ж році.

Мобільні пристрої стають все більш популярними, коли їх використовують 5,7 мільярда осіб по всьому світу, і це зіграло свою роль в тому, як споживачі взаємодіють із засобами масової інформації, і має багато інших наслідків для телевізійних рейтингів, реклами, мобільного комерції і багато чого іншого. Використання мобільних медіа, таких як потокове аудіо або мобільне відео, зростає – в Сполучених Штатах, за прогнозами, більше 100 мільйонів користувачів отримують доступ до онлайн-контенту через мобільний пристрій. Дохід від мобільного відео складається з завантажень з оплатою за перегляд, реклами і підписок. Станом на 2013 рік кількість користувачів мобільного телефону в усьому світі склало 73,4%. У 2017 році цифри свідчать про те, що більше 90% користувачів Інтернету будуть отримувати доступ до онлайн-контенту через свої телефони.

14.1.3. Стратегії SMM

14.1.3.1. Стратегії залучення соціальних мереж

Існує дві основні стратегії залучення соціальних мереж в якості інструментів маркетингу (рис. 14.2).

Рис. 14.2. Стратегії залучення соціальних мереж

14.1.3.1.1. Пасивний підхід

Соціальні мережі можуть стати корисним джерелом ринкової інформації та дізнатися думку клієнтів. Блоги, контент-спільноти та форуми – це платформи, на яких люди діляться своїми відгуками та рекомендаціями щодо брендам, продуктам і послугам. Підприємства можуть використовувати і аналізувати голосу і відгуки клієнтів в соціальних мережах для маркетингових цілей; в цьому сенсі соціальні мережі є відносно недорогим джерелом інформації про ринок, яку маркетологи і менеджери можуть використовувати для відстеження та реагування на них. Наприклад, Інтернет вибухнув відео і фотографіями iPhone 6 тест на вигин, який показав, що бажаний телефон можна зігнути руками. Це створило плутанину серед клієнтів, які місяцями чекали запуску останньої версії iPhone. Проте, Apple негайно виступила із заявою, в якому говорилося, що проблема зустрічається вкрай рідко, і що компанія зробила кілька кроків, щоб зробити корпус мобільного пристрою більш сильним і надійним. На відміну від традиційних методів дослідження ринку, таких як опитування, фокус-групи та збір даних, які забирають багато часу і вимагають великих витрат і займають тижні або навіть місяці для аналізу,

маркетологи можуть використовувати соціальні мережі для отримання живої або реальної інформації про поведінку споживачів і поглядів на бренд або продукцію компанії.

14.1.3.1.2. Активний підхід

Соціальні мережі можуть використовуватися не тільки в якості інструментів зв'язків з громадськістю та прямого маркетингу, але і в якості каналів зв'язку, призначених для дуже специфічної аудиторії з впливовими особами в соціальних мережах і особистостями соціальних мереж, а також в якості ефективних інструментів залучення клієнтів. Технології, що передують соціальним мережам, такі як широковещательное телебачення і газети, також можуть надати рекламодавцям досить точно цільову аудиторію, враховуючи, що реклама, що розміщується під час трансляції спортивних ігор або в спортивній секції газети, скоріше за все буде прочитана любителями спорту. Проте, сайти соціальних мереж можуть більш точно орієнтуватися на спеціалізовані ринки. Використовуючи цифрові інструменти, такі як Google Adsense, рекламодавці можуть орієнтувати свої оголошення на дуже специфічні демографічні дані, такі як люди, які зацікавлені в соціальному підприємництві, політичної активності, пов'язаної з певною політичною партією, або відеоіграх. Google Adsense робить це, знаходячи ключові слова в постах і коментарях користувачів соціальних мереж. Телеканалу або паперової газеті було б важко надавати рекламу, спрямовану на цю мету.

Соціальні мережі в багатьох випадках розглядаються як відмінний інструмент, що дозволяє уникнути дорогих досліджень ринку. Вони відомі тим, що вони надають короткий, швидкий і прямий спосіб досягти аудиторії через людину, яка широко відомий. Наприклад, спортсмен, якого підтримує компанія, що займається спортивними товарами, також приносить свою підтримку мільйонам людей, які зацікавлені в тому, що вони роблять або як вони грають, і тепер вони хочуть стати частиною цього спортсмена завдяки своїм схваленням до цих конкретних компаніям. У якийсь момент покупці відвідували магазини, щоб подивитися свою продукцію у знаменитих спортсменів, але тепер ви можете переглянути новітню одяг знаменитостей, таких як Кріштіану Роналду, одним натисканням кнопки. Він рекламує їх вам прямо через свої акаунти в Twitter, Instagram і Facebook.

Facebook і LinkedIn є провідними соціальними мережами, де користувачі можуть націлювати свої оголошення на гіперпосилання. Гіпертаргетінг використовує не тільки загальнодоступну інформацію профілю, а й інформацію, яку користувачі представляють, але приховують від інших. Є кілька прикладів того, як фірми починають діалог з громадськістю в тій чи іншій формі, щоб поліпшити відносини з клієнтами. Фахівці в сфері маркетингу помітили, що, керівники бізнесу, такі як Джонатан Шварц, президент і головний виконавчий директор Sun Microsystems і віце-президент McDonalds Боб Лангерт, регулярно

публікують повідомлення в своїх блогах для керівників, заохочуючи клієнтів взаємодіяти і вільно висловлювати свої почуття, ідеї , пропозиції або зауваження з приводу своїх повідомлень, компанії або її продукції. Використання впливових клієнтів (наприклад, популярних блогерів) може бути дуже ефективним способом запуску нових продуктів або послуг. Серед політичних лідерів на своєму посту прем'єр-міністр Індії Нарендра Моді має найбільше число передплатників – 40 мільйонів, а президент США Дональд Трамп займає друге місце з 25 мільйонами послідовників. Моді використовував соціальні медіа-платформи, для обходу традиційних медіа-каналів, щоб охопити молоде і міське населення Індії, яке оцінюється в 200 мільйонів чоловік.

14.1.3.1.3. Ключові помилки

Фахівцями були визначені деякі основні помилки в застосуванні SMM для ЗМІ:

- Публікації без аналізу ніші, конкурентів і розробки контент-плану
- Відсутність взаємодії в коментарях (обговореннях) з передплатниками;
- Неетичну поведінку адміністраторів у відповідь на негативні коментарі передплатників;
- Видалення негативних коментарів. На всі коментарі необхідно відповідати, навіть на негативні. Таким чином, ви показуєте користувачам, що вам не все одно;
- Недотримання принципів і законів;
- Витримування невірної пропорції інформативний контент: розважальний контент. Загальноприйнята норма 80/20, де 20 – промо пости а 80 – розважальні / захоплюючі;
- Використання одного типу контенту.

14.1.3.2. Ключові стратегії в рамках SMM

Ключові стратегії SMM представлено на рис. 14.3.

Рис. 14.3. Стратегії SMM

14.1.3.2.1. Робота з побудови спільноти навколо продукту, що просувається бренду

Побудова спільноти (*Community Building*) – це стратегія залучення користувачів до активної взаємодії з брендом через спільні інтереси,

цінності та досвід. Це не лише маркетинг, а й створення емоційного зв'язку між брендом і його споживачами.

Основні етапи побудови спільноти

1. *Визначення місії та цінностей бренду.*

Для визначення місії бренду необхідно:

- Розуміння унікальності продукту;
- Визначення головної проблеми, яку він вирішує;
- Формування чітких меседжів.

Приклад:

- Apple створює спільноту навколо ідеї інноваційності та мінімалізму.

2. *Залучення перших учасників.*

Де шукати аудиторію?

- Соцмережі (Instagram, Facebook, TikTok, Twitter);
- Форумні платформи (Reddit, Discord, Telegram);
- Подкасти, блоги, вебінари.

Тактики:

- Використання інфлюенсерів для поширення інформації;
- Запуск реферальних програм (нагородження за запрошення друзів);
- Гейміфікація (бейджі, статуси, рівні активності).

3. *Створення контенту для спільноти.*

Контент має бути:

- Корисним (гайди, поради, лайфхаки);
- Розважальним (меми, інтерактиви, конкурси);
- Надихаючим (історії успіху користувачів);
- Ексклюзивним (ранній доступ до нових продуктів).

Приклад:

- LEGO має фанатський клуб, де користувачі діляться власними моделями, беруть участь у конкурсах і голосують за нові набори.

4. *Взаємодія та підтримка активності.*

Як підтримувати інтерес?

- Відповідати на коментарі та запитання користувачів;
- Створювати закриті групи для активних учасників;
- Робити персоналізовані пропозиції та бонуси;
- Влаштовувати опитування, голосування.

Приклад:

- Tesla використовує форуми та соцмережі для залучення фанатів до тестування та вдосконалення своїх автомобілів.

5. *Перетворення спільноти на бренд-амбасадорів.*

Що працює?

- Визнання активних учасників (публікація їхніх історій);
- Запуск програм лояльності;
- Організація заходів (онлайн-мітинги, конференції, зустрічі).

Приклад:

- Starbucks створив програму «My Starbucks Idea», де клієнти могли пропонувати свої ідеї щодо покращення продукту.

14.1.3.2.2. Використання рекламних інструментів і посіви на майданчиках з метою прямого продажу продукції і підвищення впізнаваності

Використання рекламних інструментів і посівів на майданчиках є ефективними стратегіями для прямого продажу продукції та підвищення впізнаваності бренду.

Основні підходи

1. *Реклама через соціальні мережі*:

- Контекстна реклама на платформах, таких як Facebook, Instagram або TikTok, допомагає націлити рекламні кампанії на конкретну аудиторію за інтересами, демографічними характеристиками або поведінковими патернами;
- Influencer marketing (робота з лідерами думок) може сприяти підвищенню впізнаваності бренду та безпосередньому продажу, оскільки рекомендації від популярних осіб мають значний вплив на рішення споживачів.

2. *Таргетована реклама на вебсайтах та в додатках*:

- Display advertising (банерна реклама) допомагає привернути увагу до продукції безпосередньо на популярних платформах чи сайтах. Це може бути особливо ефективно для підвищення впізнаваності;
- Ремаркетинг дозволяє знову звертатися до користувачів, які вже взаємодіяли з вашим брендом, що збільшує шанси на продаж.

3. *SEO і контент-маркетинг*:

- Використання SEO (оптимізація для пошукових систем) дозволяє забезпечити високі позиції в пошукових системах, що робить ваш сайт більш помітним для потенційних покупців;
- Блогінг, створення інформативних статей, відео чи інфографік можуть допомогти підвищити впізнаваність бренду та взаємодію з клієнтами.

5. *Пряма реклама та $e-mail$-маркетинг*:

- Рекламні акції, спеціальні пропозиції через $e-mail$-розсилки можуть безпосередньо стимулювати продажі, пропонуючи цільовій аудиторії вигідні умови.

6. *Реклама на популярних платформах та вебмайданчиках*:

- Платформи для реклами на спеціалізованих майданчиках (наприклад, на сайтах новин, форумах або в інтернет-магазинах) також сприяють збільшенню впізнаваності та просуванню товарів;
- Завдяки поєднанню цих інструментів, компанії можуть досягти як короткострокових продажів, так і більш стійкої впізнаваності бренду в довгостроковій перспективі.

14.1.3.2.3. Комбінована стратегія

Комбінована стратегія – це підхід, який об'єднує різні методи та інструменти для досягнення бізнес-цілей. Вона дозволяє збалансувати короткострокові та довгострокові завдання, мінімізувати ризики та адаптуватися до змін у ринку.

Основні види комбінованих стратегій

1. *Комбінація стратегії зростання та оптимізації витрат.*

Мета:

- Збільшити дохід знижки, водночас зараховуючи операційні витрати.

Методи:

- Автоматизація бізнес-процесів (CRM-системи, чат-боти);
- Використання цифрового маркетингу замість традиційної реклами;
- Масштабування бізнесу через франчайзинг або партнерства.

Приклад:

- Компанія впроваджує AI-аналітику для прогнозування попиту, що знижує витрати на логістику, водночас запускаючи нові продукти на ринок.

2. *Поєднання онлайн- та офлайн-стратегії.*

Мета:

- Залучити широку аудиторію через багатоканальний підхід.

Методи:

- Взаємопідсилення онлайн- і офлайн-кампаній;
- Використовуйте O2O (online-to-offline) для залучення клієнтів у фізичні магазини через онлайн-рекламу;
- Лояльні програми з бонусами як в інтернет-магазині, так і в офлайн-точках.

Приклад:

- Рітейлер запускає таргетовану рекламу у Facebook із купонами на знижку, які можна використовувати лише в офлайн-магазині.

3. *Злиття диференціації та цінової конкуренції.*

Мета:

- Одночасно пропонувати унікальну цінність і конкурентні ціни.

Методи:

- Використовуйте інновації для створення унікального продукту;
- Оптимізація виробничих процесів для зниження собівартості;
- Формування гібридного підходу – преміальний сегмент із базовими доступними опціями.

Приклад:

- Автовиробник пропонує електромобіль із базовою комплектацією за низькою ціною та преміальні пакети з додатковими функціями.

4. *Поєднання традиційного маркетингу та performance-маркетингу.*

Мета:

- Створити сильний бренд і водночас отримувати вимірювальні результати від реклами.

Методи:

- Контент-маркетинг + таргетована реклама;
- Спонсорство великих заходів + ремаркетинг у соцмережах;
- Робота з інфлюенсерами + PPC (оплата за клік) у пошукових системах.

Приклад:

- Бренд проводить *PR*-кампанію в медіа, а паралельно запускає рекламні кампанії з чіткими *KPI* (буде описано нижче).

При робіт по просуванню в соціальних мережах ключовим пунктом є розробка стратегії роботи.

14.1.3.3. Основні пункти SMM-стратегії

В даний час класична SMM-стратегія трансформується і соціальні мережі вбудовуються в інтегровані маркетингові комунікації. Це пов'язано з появою нових каналів комунікації, які забирають на себе частину аудиторії і з якими також доводиться працювати. Для того, щоб розробити єдину комунікаційну стратегію для просування по всіх каналах, існує ряд моделей, одна з яких – модель PESO. В її назві зашифровані чотири основні групи каналів поширення інформації – Paid, Earned, Shared, Owned. У моделі PESO SMM розглядається в рамках Shared каналів.

Класична SMM-стратегія складається з наступних пунктів (рис. 14.4).

1. Визначення цілей і завдань.
2. Аналіз поточного стану бренду в соціальних мережах, а також аналіз конкурентів.
3. Аналіз цільової аудиторії бренду в соціальних мережах.
4. Вибір ключових майданчиків для просування.
5. Tone of voice бренду в соціальних мережах.
6. Створення рубрикатора.
7. Візуальна добірка.
8. Стратегія просування і використання платних інструментів.
9. Розрахунок KPI.
10. Аналіз проведених робіт і звітність.

Рис. 14.4. Основні пункти SMM-стратегії

14.1.3.3.1. Визначення цілей і завдань

1. Визначення цілей

Ціль – це загальний результат, до якого прагне компанія, команда або окрема людина.

Ціль повинна бути:

- Конкретною (що саме потрібно досягти);
- Вимірюваною (як можна оцінити успіх);
- Досяжною (реалістичною, але амбітною);
- Релевантною (відповідати місії та стратегії);
- Обмеженою в часі (мати дедлайни).

Приклад цілей:

- Збільшити прибуток компанії на 20% за рік;
- Підвищити впізнаваність бренду на 30% у соціальних мережах за 6 місяців.

2. Визначення завдань

Завдання – це конкретні дії, необхідні для досягнення цілі. Вони повинні бути чіткими, зрозумілими та вимірюваними.

Приклад завдань:

- Оптимізувати сайт для SEO та підвищити трафік на 40%;
- Запустити таргетовану рекламу у Facebook та Instagram;
- Впровадити програму лояльності для постійних клієнтів;
- Підключити чат-бот для швидкої відповіді на запити клієнтів.

3. Методи визначення цілей і завдань

Метод SMART

Метод SMART (*Specific, Measurable, Achievable, Relevant, Time-bound*) – це ефективний підхід до постановки цілей, який допомагає зробити їх більш конкретними, реалістичними та досяжними.

SMART розшифровується як:

S (Specific) – Конкретність. Ціль повинна бути чіткою та зрозумілою. Слід уникати загальних формулювань.

- Погано: «Покращити продажі»;
- Добре: «Збільшити продажі на 20% у категорії електроніки».

M (Measurable) – Вимірюваність. Ціль має містити кількісні або якісні показники.

- Погано: «Підвищити продуктивність»;
- Добре: «Збільшити кількість виконаних завдань на 15% за квартал».

A (Achievable) – Досяжність. Ціль має бути реальною, але водночас достатньо амбітною.

- Погано: «Збільшити прибуток у 10 разів за місяць»;
- Добре: «Збільшити прибуток на 15% за рік за рахунок нових каналів продажу».

R (Relevant) – Актуальність. Ціль має відповідати загальній стратегії компанії або особистому розвитку.

- Погано: «Запустити блог про подорожі у компанії, що займається фінансами»;
- Добре: «Розробити блог про фінансову грамотність для залучення нової аудиторії».

T (Time-bound) – Обмеженість у часі. Ціль має чіткі дедлайни.

- Погано: «Розширити клієнтську базу»;
- Добре: «Збільшити кількість клієнтів на 25% за 6 місяців»

Метод OKR

Метод OKR (*Objectives & Key Results*) — це підхід до управління цілями, які використовують компанії, команди та окремі працівники для досягнення амбітних результатів.

Структура OKR:

- *Objective (Ціль)* – чітке та амбітне формулювання того, чого потрібно досягти;
- *Key Results (Ключові результати)* – вимірюванні показники, які були досягнуті.

Принципи OKR:

- *Амбітність* – цілі потрібно мотивувати на проривні досягнення;
- *Вимірюваність* – ключові результати мають бути конкретними та кількісними;
- *Прозорість* – OKR доступний для всіх команд, щоб синхронізувати зусилля;
- *Обмежений термін* – поточний OKR встановлюються на квартал або рік.

Приклад OKR для маркетингової команди:

- Ціль: Підвищити впізнаваність бренду в соціальних мережах;
- Ключові результати: Збільшення попиту на товари даного бренду (підвищення прибутку з їх продажу; збільшення кількості покупців).
- Метод PDCA (Plan-Do-Check-Act) – циклічний процес планування та коригування.

14.1.3.3.2. Аналіз поточного стану бренду в соціальних мережах та аналіз конкурентів

Основні етапи аналізу поточного стану бренду в соцмережах

Етап 1. Аналіз присутності в соцмережах. На даному етапі необхідно:

- Дати відповідь на питання: «Які платформи використовуються (Facebook, Instagram, TikTok, LinkedIn тощо)?»;
- Дати відповідь на питання: «Яка частота та формат публікацій (пости, сторіс, відео, стріми)?»;
- Дати відповідь на питання: «Чи є єдиний візуальний стиль і tone of voice?».

Етап 2. Оцінка ефективності контенту. На даному етапі необхідно:

- Дати відповідь на питання: «Які типи контенту отримують найбільше залучення (фото, відео, каруселі, тексти)?»;
- Отримати оцінку взаємодії аудиторії (лайки, коментарі, репости, збереження);
- Отримати оцінку частоти публікацій та їхнього впливу на охоплення.

Етап 3. Аналіз аудиторії. На даному етапі необхідно:

- Дати відповідь на питання «Який вік, стать, географія та інтереси підписників?»;
- Визначити рівень залученості (ER – Engagement Rate);
- Визначити відсоток активної та пасивної аудиторії.

Етап 4. Аналіз трафіку та конверсій. На даному етапі необхідно:

- Дати відповідь на питання: «Скільки людей переходять із соцмереж на сайт або в магазин?»;
- Дати відповідь на питання: «Який коефіцієнт конверсії (покупки, підписки, заявки)?»;
- Дати відповідь на питання: «Які рекламні кампанії дали найкращі результати?».

Етап 5. Аналіз репутації та зворотного зв'язку. На даному етапі необхідно:
- Дати відповідь на питання: «Які відгуки залишають користувачі?»;
- Дати відповідь на питання: «Чи є негативні коментарі та як вони обробляються?»;
- Дати відповідь на питання: «Чи є брендові згадки та хештеги в інших акаунтах?».

Основні етапи аналізу конкурентів у соціальних мережах

Етап 1. Визначення головних конкурентів. На даному етапі необхідно розбити конкурентів на наступні класи:
- Прямі (схожі продукти, аудиторія);
- Непрямі (схожий контент або стиль комунікації);
- Лідери ринку (бенчмарк для порівняння).

Етап 2. Оцінка контент-стратегії конкурентів. На даному етапі необхідно:
- Дати відповідь на питання: «Які теми вони висвітлюють?»;
- Дати відповідь на питання: «Як вони взаємодіють із підписниками?»;
- Дати відповідь на питання: «Які формати використовують (відео, GIF, інфографіка, меми тощо)?».

Етап 3. Аналіз залученості аудиторії. На даному етапі необхідно:
- Визначити середню кількість лайків, коментарів та репостів;
- Визначити, який тип контенту викликає найбільшу активність;
- Визначити чи працюють конкуренти з інфлюенсерами.

Етап 4. Аналіз рекламних кампаній. На даному етапі необхідно:
- Визначити, чи запускають вони таргетовану рекламу;
- Визначити, які пропозиції просувають (акції, знижки, новинки);
- Визначити, чи використовують ретаргетинг.

Етап 5. Оцінка сильних і слабких сторін. . На даному етапі необхідно:
- Визначити сильні сторони конкурентів: що у них працює краще?
- Визначити слабкі сторони: що можна використати як можливість?

Інструменти для аналізу

- *Google Analytics, Facebook Insights, Instagram Analytics* – аналіз трафіку та активності;
- *Hootsuite, Sprout Social* – моніторинг соцмереж;
- *Popsters, Socialbakers* – порівняльний аналіз контенту конкурентів;
- *SEMrush, Ahrefs* – аналіз трафіку з соцмереж та ключових запитів.

14.1.3.3.3. Аналіз цільової аудиторії бренду в соціальних мережах

Аналіз цільової аудиторії (*ЦА*) у соціальних мережах конкуренції, хто є вашими основними підписниками, що їх цікавить і як ефективніше взаємодіяти з ними.

1. Збір та аналіз даних про аудиторію

Джерела інформації:

- Аналітика соціальної мережі (Facebook Insights, Instagram Analytics, YouTube Studio, Twitter Analytics);
- Google Analytics (якщо є сайт, пов'язаний із соцмережами);
- Соціальне опитування (анкетування, коментарі, відгуки);
- Конкурентний аналіз (аудиторія конкурентів).

Аналітика соціальних мереж допомагає брендам розуміти ефективність контенту, залученість аудиторії та покращувати маркетингову діяльність.

Google Analytics (*GA*) – потужний інструмент для аналізу трафіку, який підсумовує ефективність соцмереж у захопленні відвідувачів сайту та їх поведінку.

Ключові метрики:

- Демографія (вік, стаття, сайт);
- Інтереси та поведінка (який контент взаємодіють, які теми популярні);
- Активність (час і час найбільшої взаємодії);
- Пристрої та платформи (мобільні/десктопні користувачі).

2. Сегментація аудиторії

Після збору даних необхідно розділити аудиторію на сегменти:

- За віком (Gen Z, Millennials, Boomers);
- За інтересами (спорт, мода, технології);
- За поведінкою (активні коментатори, пасивні споживачі контенту, оцінки клієнтів);
- За рівнем залученості (нові підписники, лояльні фанати, покупці).

3. Аналіз конкурентів

Предмети дослідження:

- Який контент вибір конкурентів?
- Як взаємодіє їхня аудиторія (коментарі, лайки, репости)?
- Які платформи у них працюють краще?
- Які рекламні стратегії вони застосовують?

Інструменти дослідження:

- *Brandwatch, Sprout Social* – для аналізу згадувань у соцмережах;
- *Social Blade* – для аналізу зростання підписників;
- *SEMrush, SimilarWeb* – для оцінки трафіку.

4. Визначення ідеального портрета ЦА (Customer Persona)

Приклад профілю ЦА:

Ім'я: Анна, 28 років;

Локація: Київ;

Платформи: Instagram, TikTok;

Інтереси: Мода, б'юті-продукти, лайфстайл;

Поводження: Часто коментує, зберігає пости, купує рекомендації за блогерами

Створення кількох таких портретів допоможе точніше таргетувати контент і рекламу.

5. Використання результатів аналізу

Оптимізація контенту – створення постів, що інтересам аудиторії;
Час публікацій – публікації в роки найбільшої активності;
Персоналізація реклами – таргетинг на потрібні сегменти аудиторії;
Підвищення залученості – інтерактиви, опитування, прямі ефіри.

14.1.3.3.4. Вибір ключових майданчиків для просування

Вибір платформи залежить від цільової аудиторії, специфіки бізнесу та формату контенту.

Критерії вибору платформи

Цільова аудиторія – вік, інтереси, поведінка користувачів;
Формати контенту – відео, текст, зображення, прямі ефіри;
Алгоритми платформи – органічне охоплення, рекламні можливості;
Конкуренція – активність конкурентів у соцмережі.

Популярні соцмережі та їх особливості

Instagram.

Аудиторія: 18–44 роки;

Контент: фото, відео, сторіз, Reels;

Підходить для: моди, краси, лайфстайлу, e-commerce;

Просування: таргетована реклама, блогери, взаємодія з підписниками.

Facebook.

Аудиторія: 25–55 років;

Контент: статті, відео, події, групи;

Підходить для: B2B, локального бізнесу, ком'юніти;

Просування: таргетована реклама, групи, ретаргетинг.

TikTok.

Аудиторія: 13–35 років;

Контент: короткі відео, челенджі;

Підходити для: розваг, навчання, брендів із молодою ЦА.

Метод відбору та оцінки соціальних мереж

Постановка задачі

Нехай задано:

- Множину допустимих альтернатив $A = \{a_1, a_2, \dots, a_n\}$ (розглянутих соціальних мереж), які оцінюються за m критеріями;
- Кожній альтернативі $a_i \in A$, $i = \overline{1,n}$ поставлено у відповідність m числових показників $x_{i1}, \dots, x_{im}$, де $x_{ij}, j = \overline{1,m}$ – значення оцінки альтернативи a_i за j –м критерієм.

Завдання особи, що приймає рішення – вибрати альтернативу $a_i \in$A так, щоб максимізувати функцію цінності v_i.

Слід зазначити, що для визначення показника x_{ij} необхідно обрати відповідну шкалу оцінювання. Зазвичай вона обирається на розсуд особи, що приймає рішення. Зокрема, в даній роботі використовувалася наступна шкала оцінювання (14.1):

$$x_{ij} = \begin{cases} 0, \text{ низький рівень відповідності критерію } j, \\ 0{,}5, \text{достатній рівень відповідності критерію } j, \\ 1, \text{високий рівень відповідності критерію } j; \end{cases} \quad (14.1)$$

- Кожному з l факторів, які описують певний критерій ставиться у відповідність числовий показник $x_{ij}^k, k = \overline{1,l}$ – значення оцінки альтернативи a_i за $k-$ м фактором j –го критерію.
- Для визначення показника x_{ij}^k також необхідно обрати відповідну шкалу оцінювання (14.2):

$$x_{ij}^k = \begin{cases} 0, \text{ низький рівень відповідності фактору } k, \\ 0{,}5, \text{достатній рівень відповідності фактору } k, \\ 1, \text{високий рівень відповідності фактору } k; \end{cases} \quad (14.2)$$

- Набір l чисел $(\omega_j^1, \dots, \omega_j^l)$, які відповідають критеріям оцінки $k-$ го фактору j –го критерію і визначають за кожним критерієм переваги особи, що приймає рішення (далі ваговий коефіцієнт важливості фактору ω_{ij}^k);
- Набір m чисел $(\omega_1, \dots, \omega_m)$, які відповідають критеріям оцінки альтернатив і визначають за кожним критерієм переваги особи, що приймає рішення (далі ваговий коефіцієнт важливості критеріїв ω_j).

Функція цінності v_i, представляється виразами (14.3)-(14.7):

$$v_i = \sum_{j=1}^{m}\left(\sum_{k=1}^{l}\left(x_{ik}^k \cdot \omega_{ij}^k\right)\right) \cdot \omega_j, \quad (14.3)$$

або

$$v_i = \sum_{j=1}^{m}\left(x_{ij} \cdot \omega_j\right), \quad (14.4)$$

де

$$x_{ij} = \sum_{k=1}^{l}\left(x_{ik}^k \cdot \omega_{ij}^k\right); \quad (14.5)$$

$$\sum_{k=1}^{l} \omega_{ij}^k = 1; \quad (14.6)$$

$$\sum_{j=1}^{m} \omega_j = 1. \quad (14.7)$$

В формулах (14.6) та (14.7) $\omega_{ij}^k \in [0;1]$, $\omega_j \in [0;1]$.

На основі отриманих значень функцій цінності v_i необхідно провести ранжування альтернатив з метою визначення соціальної мережі, яка найбільше відповідатиме потребам дослідження.

Алгоритм оцінки соціальних мереж

Крок 1. Визначити перелік соціальних мереж, зібрати та проаналізувати дані з них.

Крок 2. Визначити перелік необхідних критеріїв для подальшої оцінки обраних соціальних мереж.
Крок 3. На основі зібраних статистичних даних поставити оцінки x_{ij}^k кожному фактору критеріїв, що розглядалися, згідно визначеної шкали оцінювання (див. формулу (14.2)).
Крок 4. Обчислити вагові коефіцієнти важливості кожного фактору ω_{ij}^k (див. формулу (14.6)) на основі попереднього аналізу статистичних даних.
Крок 5. Обчислити вагові коефіцієнти важливості критеріїв ω_j (див. формулу (14.7)).
Крок 6. Обчислити для кожної соціальної мережі функцію цінності v_i (див. формулу (14.3)).
Крок 7. Ранжувати соціальні мережі згідно отриманих значень відповідної функції цінності.
Крок 8. Обрати соціальну мережу з найбільшим значенням функції цінності в якості найбільш підходящої для досліджень.

Запропонований підхід представлено на рис. 14.5 [41].

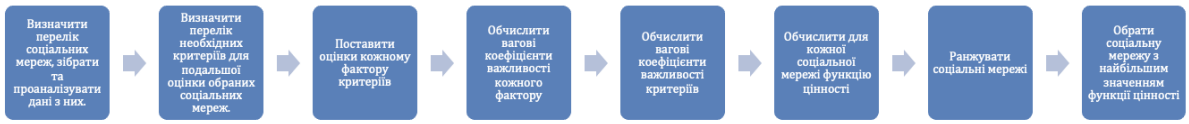


Рис. 14.5. Схема запропонованого алгоритму

14.1.3.3.5. Tone of voice бренду в соціальних мережах

Tone of Voice (*ToV*) — це унікальний стиль спілкування бренду з аудиторією, який хочеться, як компанія звучить у соцмережах. Він формує імідж бренду, закінчує впізнаваність і хоче встановити емоційний зв'язок із підписниками.

Чому важливий Tone of Voice?

- Виділяєте бренд серед конкурентів;
- Формує довіру та лояльність аудиторії;
- Підвищує впізнаваність і запам'ятовуваність;
- Допомагає утримувати єдиний стиль у комунікаціях.

Основні характеристики Tone of Voice:

- *Формальність.* Офіційний чи неформальний стиль спілкування;
- *Емоційність.* Сухий, нейтральний чи емоційно насичений контент;
- *Гумор.* Використання жартів, мемів, сарказму;
- *Проста мова.* Складні конструкції чи легкі, доступні тексти.

Приклад:

- *Офіційний стиль.* «Наш продукт допоможе оптимізувати бізнес-процеси та підвищити ефективність роботи» .
- *Дружній стиль.* «Хочеш прокачати свій бізнес? Наш сервіс зробить усе за тебе!»

14.1.3.3.6. Створення рубрикатора

Рубрикатор — це структура контенту, яка допомагає систематизувати публікації, підтримувати єдиний стиль комунікації та ефективно залучати аудиторію.

Чому потрібен рубрикатор?

- Полегшує планування контенту;
- Підвищує впізнаваність бренду;
- Балансує інформаційний, розважальний та комерційний контент;
- Допомагає уникнути хаотичних публікацій.

Основні категорії контенту.

Інформаційний – дає корисні знання:

- Лайфхаки, експертні поради;
- Огляди трендів;
- Кейси та дослідження.

Розважальний – залучає аудиторію:

- Мем-контент, інтерактиви;
- Опитування, вікторини;
- Бекстейджи, історії з життя команди.

Рекламний (комерційний) – мотивує до покупки:

- Оголошення акцій;
- Презентація продуктів;
- Відгуки клієнтів.

Залучаючий (Engagement) – стимулює комунікацію:

- Опитування та дискусії;
- Челенджі, флешмоби;
- Конкурси та гейміфікація.

Іміджевий – формує бренд:

- Місія, цінності компанії;
- Соціальна відповідальність;
- Культура та команда.

Як створити рубрикатор?

- *Аналіз ЦА* – що цікавить підписників?
- *Визначення ключових тем* – про що писати?
- *Форматування* – пости, відео, сторіз, каруселі
- *Частота публікацій* – який контент публікувати щодня, щотижня?
- *Гнучкість* – коригування стратегії залежно від реакції аудиторії.

Інструменти для планування контенту:

- Google Sheets / Notion – створення контент-плану;
- Trello / Asana – управління рубриками;
- Hootsuite / Planoly – автопостинг і аналітика.

Приклад рубрикатора для Instagram:

- Понеділок – корисні поради;
- Вівторок – закулісне життя компанії;
- Середа – розбір кейсу;

- Четвер – мем або інтерактив;
- П’ятниця – відгуки клієнтів;
- Субота – історії команди;
- Неділя – особистий контент або цінності бренду.

14.1.3.3.7. Візуальна добірка

Візуальна добірка – це узгоджена система оформлення контенту, яка допомагає створити впізнаваний стиль бренду в соцмережах. Вона включає кольорову гаму, типографіку, шаблони для постів, фото- та відеоконтент.

Чому важлива візуальна добірка?
- Підвищує впізнаваність бренду;
- Формує єдиний стиль комунікації;
- Виділяє серед конкурентів;
- Покращує залученість та довіру аудиторії.

Основні елементи візуальної добірки:
- Кольорова гама;
- Типографіка (Шрифти);
- Графічні елементи;
- Фотостиль;
- Шаблони для постів;
- Відеостиль.

Як створити візуальну добірку?
- Аналіз бренду – визначення, які асоціації має викликати контент;
- Розробка Moodboard – зразки кольорів, шрифтів, фото;
- Тестування та адаптація – реакція аудиторії на візуал;
- Єдність контенту – всі платформи мають мати однаковий стиль.

Інструменти для створення візуальної добірки:
- Canva, Figma – створення шаблонів;
- Adobe Photoshop, Lightroom – обробка фото;
- Pinterest, Milanote – збереження референсів.

14.1.3.3.8. Стратегія просування і використання платних інструментів

Ефективне просування бренду в соцмережах поєднує органічний контент та платні інструменти, такі як таргетована реклама, співпраця з блогерами та просування постів.

Основні етапи стратегії просування було наведено вище.

Основні етапи стратегії просування

1. *Аналіз цільової аудиторії (ЦА).* Як зазначалося вище, основним етапом є визначення цільової аудиторії для ефективності просування бренду. Зокрема, аналізуються наступні параметри:
- Вік, стать, геолокація, інтереси;
- Поведінка у соцмережах (які сторінки відвідують, на що реагують);
- Основні проблеми та потреби.

2. *Вибір платформи для просування.* Раніше було зазначено, що найбільш затребуваними платформами є:

- *Facebook, Instagram* – для широкої аудиторії та e-commerce;
- *TikTok* – для молодої ЦА, відеоконтенту;
- *LinkedIn* – для B2B та корпоративних рішень;
- *YouTube* – для довгострокового відеопросування.

3. *Визначення KPI (ключові показники ефективності).* *KPI* є одним з ключових параметрів для оцінки ефективності просування бренду. Він може включати наступні показники:
- Охоплення, залученість (лайки, коментарі, репости);
- Конверсія (підписки, заявки, продажі);
- Вартість залучення клієнта (CPA).

Більш буде докладно описано нижче.

Платні інструменти для просування

1. *Таргетована реклама.* До інструментів таргетованої реклами належать:

1.1. *Facebook Ads, Instagram Ads*:
- Налаштування аудиторії (вік, інтереси, поведінка);
- Ретаргетинг (повторне залучення користувачів, що вже взаємодіяли з брендом);
- Lookalike-аудиторії (пошук схожих клієнтів);

1.2. *TikTok Ads*:
- Охоплення відеоконтентом;
- Бюджетний варіант для вірусного просування.

1.3. *Google Ads (YouTube, дисплейна мережа)*:
- Реклама на відео;
- Контекстна реклама для залучення трафіку.

2. *Співпраця з блогерами.* Створення колаборацій з різними типами блогерів з метою охоплення цільової аудиторії:
- Макроінфлюенсери (100К + підписників) – швидке охоплення, підходить для великих брендів;
- Мікроінфлюенсери (10 – 50К підписників) – більше довіри, дешевша взаємодія.

3. *Просування постів та Stories.* Розробка ефективної стратегії залучення цільової аудиторії за допомогою постів в соціальних мережах. Зокрема:
- Пряме просування через Facebook Ads Manager;
- Використання UGC (користувацького контенту) для реклами.

4. *Платні партнерства та колаборації.* Використання різних способів співпраці з брендами. Зокрема:
- Взаємний піар з іншими брендами;
- Спільні конкурси, гівевеї.

Оптимізація рекламних кампаній

1. *Аналіз ефективності.* До засобів аналізу ефективності рекламної компанії належать:
- Google Analytics, Facebook Pixel – відстеження поведінки користувачів;

- A/B тестування різних варіантів реклами;
- Оптимізація ставок і бюджетів.

A/B тестування (*спліт-тест*) — це метод порівняння двох варіантів реклами, контенту або сторінок, щоб визначити, який працює краще.

Більш докладно буде описано в наступних лекціях.

2. *Зниження витрат на рекламу.* З метою зниження витрат на рекламу, використовують:

- Тестування аудиторій;
- Оптимізація контенту (відео, креативи);
- Використання ретаргетингу.

14.1.3.3.9. Розрахунок *KPI*

KPI (*Key Performance Indicators*) або ключові показники ефективності — це метрики, які використовуються для оцінки успіху організації, проекту або окремих процесів у досягненні стратегічних або операційних цілей. KPI допомагають виміряти ефективність роботи і забезпечують орієнтир для прийняття рішень.

Формули для обчислення KPI залежатимуть від конкретного показника.

Формули для популярних *KPI*

1. *Показник конверсії.*

Показник конверсії (*Conversion Rate*) – метрика, що обчислюється за формулою (14.8):

$$Conversion\ Rate = \frac{\text{Кількість виконаних дій}}{\text{Кількість відвідувачів}} \cdot 100\%. \quad (14.8)$$

2. *Вартість залучення клієнта.*

Вартість залучення клієнта (*Customer Acquisition Cost, CAC*) – метрика, що обчислюється за формулою (14.9):

$$CAC = \frac{\text{Загальні витрати на маркетинг і продажі}}{\text{Кількість нових клієнтів}}. \quad (14.9)$$

3. *Маржа прибутку.*

Маржа прибутку (*Profit Margin*) – метрика, що обчислюється за формулою (14.10):

$$Profit\ Margin = \frac{\text{Чистий прибуток}}{\text{Доходи}} \cdot 100\%. \quad (14.10)$$

4. *Індекс задоволеності клієнтів.*

Індекс задоволеності клієнтів (*Customer Satisfaction Score, CSAT*) – метрика, що обчислюється за формулою (14.11):

$$CSAT = \frac{\text{Сума оцінок клієнтів}}{\text{Кількість респондентів}} \cdot 100\%. \quad (14.11)$$

Зазвичай оцінки коливаються від 1 до 5, або від 1 до 10, і середнє значення використовується для підрахунку.

5. *Час виконання замовлення.*

Час виконання замовлення (*Order Fulfillment Time*) – метрика, що обчислюється за формулою (14.12):

$$Order\ Fulfillment\ Time = \text{Дата доставки} - \text{Дата отримання замовлення}. \quad (14.12)$$

6. *Текучість кадрів.*

Текучість кадрів (*Employee Turnover Rate*) – метрика, що обчислюється за формулою (14.13):

$$Turnover\ Rate = \frac{\text{Кількість звільнених співробітників}}{\text{Середня кількість співробітників за період}} \cdot 100\% \quad .(14.13)$$

7. *Залучення співробітників.*

Індекс залученості (*Employee Engagement*) може вимірюватися через різні опитування і відсоткові оцінки рівня залученості співробітників на основі відповідей на питання, наприклад (14.14):

$$Employee\ Engagement\ Score = \frac{\text{Кількість позитивних відповідей}}{\text{Загальна кількість опитаних}} \cdot 100\% \quad (14.14)$$

8. *NPS.*

NPS (*Net Promoter Score*) – метрика, що обчислюється за формулою (14.15):

$$NPS = \%\text{Промоутерів} - \%\text{Критиків}, \quad (14.15)$$

де Промоутери — це ті, хто оцінює на $9-10$, а Критики — на $0-6$ за шкалою від 0 до 10.

14.1.3.3.10. Аналіз проведених робіт і звітність

Аналіз ефективності просування та контент-стратегії є ключовим етапом маркетингу. Регулярна звітність допомагає оцінити результати, оптимізувати кампанії та покращити показники бренду.

Деякі види аналізу (зокрема розрахунок KPI) було описано вище. Розглянемо загальні показники, які дозволяють оцінити ефективність маркетингу в соціальних мережах.

Органічний приріст абонентів (OFG)

OFG (*organic followers growth*) – це користувачі соціальних мереж, які підписалися на сторінку, паблік або групу без рекламної комунікації з ними. Даний показник може свідчити про якість розміщується контенту, про впізнаваності бренду / особистого бренду.

OFG може бути розрахований вручну за формулою (14.16):

$$OFG = N - V, \quad (14.16)$$

де:

N – нові підписники без застосування реклами;

V – відписки без застосування реклами.

Або у відсотках (14.17):

$$OFG\% = \frac{OFG}{R} \cdot 100\%, \quad (14.17)$$

де R – загальна кількість підписників на початок періоду.

В соціальній мережі Facebook його можна подивитися в розділі Статистика – Передплатники.

Приклад розрахунку. На початку місяця було $R = 10000$ підписників. За місяць без реклами додалося $N = 1200$ нових підписників. Відписалося $V = 300$ користувачів

Розраховуємо приріст у кількостях підписників (див. формулу (14.16)):

$$OFG = 1200 - 300 = 900.$$

Розраховуємо приріст у відсотках (див. формулу (14.17)):

$$OFG\% = \frac{900}{10000} \cdot 100\% = 9\%.$$

Отже, органічний приріст за місяць склав 900 підписників (9%).

Органічний приріст лайків (OL)

OL (*organic likes*) – це показник відміток подобається до однієї публікації або публікаціях, розміщених за певний проміжок часу, в разі коли публікація не просувається за допомогою інструментів таргетированной реклами або платного розміщення інформаційного повідомлення в спільнотах або у блогера. Даний показник може свідчити про якість розміщується контенту, а також про лояльність до бренду / особистого бренду. Показник органічного приросту лайків може бути розрахований вручну, з метою порівняти кількість позначок подобається на інший період часу (наприклад, попередній місяць) і зробити висновки про інтереси аудиторії, про найбільш популярному контенті.

Органічний приріст лайків можна розрахувати за формулою (14.18):

$$OL = P - Q, \qquad (14.18)$$

де:

P – загальна кількість лайків за період;

Q – кількість лайків, що отримані через рекламу.

або у відсотках (14.19):

$$OL\% = \frac{OL}{M} \cdot 100\%, \qquad (14.19)$$

де M – загальна кількість лайків на початок періоду.

Приклад розрахунку. Загальна кількість лайків на місяць $P = 5000$. Лайки, отримані через рекламу $Q = 1500$. Загальна кількість лайків на початок періоду – $M = 20000$.

Розрахунок приросту лайків у штуках (див. формулу (14.18)):

$$OL = 5000 - 1500 = 3500.$$

Розрахунок приросту лайків у відсотках (див. формулу (14.19)):

$$OL\% = \frac{3500}{20000} \cdot 100\% = 17{,}5\%.$$

Таким чином, органічний приріст лайків за місяць склав 3500 лайків (17,5%).

Приріст аудиторії спільноти

PA (*приріст аудиторії спільноти*) – це показник, що дозволяє оцінити ефективність контент-плану, якість взаємодії з передплатниками за певний проміжок часу і вимірюється у відсотках. Розраховується як співвідношення нових передплатників до числа колишніх передплатників.

Приріст аудиторії спільноти можна розрахувати за такою формулою (14.20):

$$PA = H - U, \quad (14.20)$$

де:

H –нові підписники за певний період;

U –кількість відписок за той самий період.

Або у відсотках (14.21):

$$PA\% = \frac{N-U}{R} \cdot 100\%, \quad (14.21)$$

де R — загальна кількість підписників на початок періоду.

Приклад розрахунку. На початку місяця в спільноті було 15000 підписників. За місяць додалося 2500. Відписалося 800 нових підписників. Відписалося 800 користувачів

Розраховуємо приріст у кількість (див. формулу (14.20)):

$$PA = 2500 - 800 = 1700.$$

Розраховуємо приріст у відсотках (див. формулу (14.21)):

$$PA\% = \frac{1700}{15000} \cdot 100\% = 11{,}3\%.$$

Приріст аудиторії за місяць склав 1700 підписників (11,3%).

Вартість одного передплатника

CPA (*вартість одного передплатника)* – це вартість одного залученого передплатника за допомогою використання рекламних каналів комунікації в інтернеті. Розраховується як співвідношення рекламного бюджету за певний проміжок часу до числа залучених передплатників.

Вартість одного передплатника додатково за формулою (14.22):

$$CPA = \frac{W}{Z}. \quad (14.22)$$

де:

W – загальні витрати на рекламу;

Z – кількість залучених підписників через рекламу.

Приклад розрахунку. Загальні витрати на рекламу $W = 500\$$. Кількість підписників, отриманих через рекламу $Z = 1250$.

Розраховуємо CPA:

$$CPA = \frac{500}{1250} = 0{,}4.$$

Вартість одного підписника 0,40$ або 40 центів.

Оформлення аналітичних звітів

Як оформити звітність?

- Щотижневі звіти – короткий аналіз результатів;
- Щомісячні звіти – детальний аналіз KPI, динаміки та висновків;
- Квартальні звіти – стратегічний аналіз трендів та ефективності.

Інструменти для аналізу та звітності:

- Facebook Business Suite, Instagram Insights – аналітика соцмереж;
- Google Analytics – аналіз трафіку сайту;

- Hootsuite, Sprout Social – комплексний моніторинг соцмереж;
- Power BI, Google Data Studio – створення візуальних звітів.

14.1.4. SMM-фахівець

Основні вимоги до SMM-фахівця

1. *Розробка стратегії для написання контент-плану*: визначення цільової аудиторії, дослідження інтересів аудиторії, визначення поведінки аудиторії, проведення аналізу клієнтської ніші, розробка бази для аудиторії, підбір місця з високою концентрацією цільової аудиторії, розробка загальної стратегії присутності в соціальних мережах, підбір інструментів і каналів , оптимально вирішують завдання, розробка системи клієнтської лояльності, визначення впливу SMM, інтегрування SMM-активності в загальну маркетингову стратегію компанії. Розміщення публікацій в соціальних мережах, аналіз залученості користувачів, проведення конкурсів, робота з аудиторією.
2. *SMM-навички*: правильно позиціонувати співтовариство, управляти таргетированной рекламою в Facebook і Instagram, розуміти механізми реклами, прогнозувати бюджет таргетированной реклами, оптимізувати його, проводити конкурси і флеш-моби, розробляти програми для соцмереж, створювати і просувати заходи, управляти вірусним маркетингом, працювати з гео-сервісами, працювати з системою пропозицій, проектувати і проводити спецпроекти, професійні соцмережі, розуміти принципи формування рейтингів і топів, створювати канали на відеохостингу, вибудовувати партнерські програми в соцмережах, користуватися сервісами для оптимізації роботи.
3. *Ком'юніті-менеджмент*: направляти обговорення в потрібне русло, нейтралізовувати негативне ставлення користувачів, підвищувати активність користувачів в спільнотах, підвищувати повторні звернення до вмісту, організовувати і проводити онлайн-івенти, організовувати службу підтримки через соцмережі.
4. *Контент-менеджмент*: створювати карту вмісту для різних майданчиків, адаптувати існуючий вміст під формат блогу, відеохостингу, писати тексти під формат соцмережі, блогу, поширювати соціальні релізи, готувати інфоприводи, створювати сценарії для відео.
5. *Робота з інтерфейсами*: брендировать співтовариство, створювати дизайн спільнот, інтегрувати сайт з соцмережами, інтегрувати соцмережі з електронним магазином, впроваджувати стимули для вступу, створювати посадочні сторінки, створювати стартові сторінки, створювати вкладки, працювати зі стенд-елонг блогами.
6. *Робота з лідерами думок*: виділяти лідерів думок цільової аудиторії, організовувати івенти для лідерів думок, проводити акції семплінгів, вести роботу з громадами цивільних маркетологів.
7. *Аналітика*: користуватися моніторингом, моніторити соцмережі і блоги вручну або за допомогою спеціальних сервісів, проводити аналітику

інфоприводів, проводити аналітику тональності згадувань, знаходити джерело негативу в соцмережах і блогах, аналізувати ефективність компанії, працювати з основними системами веб-аналітики, генерувати унікальні посилання, відстежувати джерела і якість трафіку, прораховувати вартість цільового дії, проводити аналітику зміни інформаційного поля, проводити дослідження в соцмережах, визначати природу негативу, користуватися інструментами статистики соцмереж.

8. *PR в інтернеті*: проводити переговори з редакціями онлайн-ЗМІ, розуміти основи соціальної психології, проводити маркетинговий аналіз, займатися медіа плануванням, медіа торгівлею, формулювати публікації матеріалів, розуміти специфіку роботи з брендом, користуватися класичними інтернет-маркетинговими інструментами.

9. *Загальні навички менеджменту*: розуміти ключові потреби аудиторії, вести переговори, відстежувати тенденції в SMM, знаходити партнерів і управляти процесом роботи з ними, формулювати шаблони на основі успішних кампаній, готувати презентації, календарний план компанії і дотримуватися його, готувати звіти, захищати перед керівництвом концепції, організовувати мозковий штурм в проектній групі, керувати проектами.

У зв'язку з постійним ускладненням SMM-стратегій і зростанням вимог до сторінок брендів в соцмережах, спостерігається тренд на поділ обов'язків SMM-фахівців на множину різних вузькопрофільних менеджерів:

- Проєктний менеджер;
- Контент менеджер;
- Копірайтер;
- Дизайнер / motion-дизайнер;
- Модератор;
- Ком'юніті-менеджер;
- Менеджер з орієнтування;
- Фотограф / Відеографія;
- SERM воно ж Управління репутацією.

14.1.4.1. Модель та методи для створення автоматизованої системи підтримки роботи фахівця з маркетингу у соціальних мережах

Модель системи підтримки фахівця з маркетингу у соціальних мережах зображена на рис. 14.6.

Рис. 14.6. Архітектура моделі

Архітектура моделі складається з наступних задач:

- Задача створення цільових аудиторій;
- Задача визначення найвпливовіших факторів успішності;
- Задача прогнозування очікуваної кількості приросту цільової аудиторії;
- Задача обрахунку KPI.

Розглянемо детально кожну з цих задач.

14.1.4.1.1. Задача створення цільових аудиторій

Вхідні дані:

- Люди, їх вік, їх інтереси.

Вихідні дані:

- Цільові аудиторії.

Цільові аудиторії — сегменти розподілені за віком та інтересами людей.

Алгоритм розв'язання задачі створення цільових аудиторій зображений на рис. 14.7.

Рис. 14.7. Алгоритм розв'язання задачі створення цільових аудиторій

14.1.4.1.2. Задача визначення найвпливовіших факторів успішності

Вхідні дані:

- Потрібна початкова множина даних, яка включає в себе статистику впливовості кожного фактору (14.23):

$$F = \{f_1, \dots, f_i, \dots, f_k\}, \quad (14.23)$$

де:

f – фактор;

$i = \overline{1,k}$;

k – кількість факторів.

На рис. 14.8 показано фрагмент початкової множини даних.

№	Фактор	Початкове значення для майбутнього обрахування
1	Популярність теми	1200000
2	Правильне оформлення інформації на сторінці	1150000
3	Зовнішній вигляд сторінки	959000
4	Рівномірне планування постів	910000
5	Поточний стан сторінки	201000
6	Слідування за трендами	941000
7	Стильний дизайн	157000
8	Власний бренд	189000
9	Привертання уваги	1941000
10	Хештеги	121000

Рис. 14.8. Фрагмент початкової множини даних

Вихідні дані:

- Найвпливовіші фактори з множини вхідних даних (14.24):

$$F^{imp} = \{f_1^{imp}, \dots, f_n^{imp}\}, \quad (14.23)$$

де n – кількість важливих факторів.

Для реалізації даної задачі використовувався мемод головних компонент (див. лекцію 8).

На рисунку 14.9 зображено схему дій задачі визначення найвпливовіших факторів успішності.

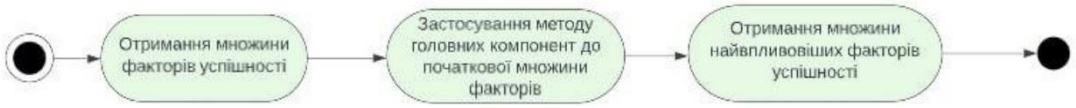


Рис. 14.9. Схема дій задачі визначення найвпливовіших факторів успішності

14.1.4.1.3. Визначення рекомендованих термінів просування контенту

Вхідні дані: Статистика минулих вимирів результатів користувачів (підписники, вподобання контенту, перегляди).

Вихідні дані: Рекомендовані терміни просування, поточний приріст.

Для якісного прогнозування очікуваної кількості приросту, потрібно використати метод часових рядів – ARIMA (див. лекцію 9). Це допоможе обрати правильні терміни для просування контенту, а також для покращення майбутнього значення KPI.

На рис. 14.10 зображена схема використання часових рядів при прогнозуванні термінів просування контенту.

Рис. 14.10. Схема використання методу ARIMA

14.1.4.1.4. Задача прогнозування очікуваного приросту цільової аудиторії

Вхідні дані:

- Найвпливовіші фактори за допомогою методу головних компонент;
- Часовий проміжок із часового ряду;
- Кластери для майбутнього просування.

Вихідні дані:

- Очікувана кількість підписників;
- Очікувана кількість вподобань контенту та переглядів.

Для реалізації даної задачі також використовується алгоритм ARIMA.

Для обрахунку очікуваної кількості, потрібно використати статистику приросту користувача до просування за допомогою часових рядів, а також знайти середнє значення похибки для минулих прогнозів. Якщо ж минулих прогнозів не було, то використаємо стандартну константу – 10%. Це показано у формулі (14.24).

$$\varepsilon_c = \frac{\sum(P_r)}{K_p}, \quad (14.24)$$

де:

ε_c – середня похибка;

P_r – похибка кожного прогнозу r;

K_p – кількість прогнозів.

Обрахуємо загальну похибку для прогнозу. Це показано у формулі (14.25).

$$\varepsilon = \frac{P_m - \varepsilon_c \cdot P_m + P_c}{P_m} \cdot 100\%, \quad (14.25)$$

де:

ε – загальна похибка;

P_m – максимальна кількість приросту цільової аудиторії;

ε_c – середньо-статистична похибка;

P_c – приріст цільової аудиторії до моменту просування.

Отже, знаючи загальну похибку ми зможемо отримати кількість приросту цільової аудиторії. Це показано у формулі (14.26).

$$A_r = P_m - P_m \cdot \varepsilon, \quad (14.26)$$

де A_r – очікувана кількість приросту цільової аудиторії.

Схема алгоритму показана на рис. 14.11.

Рис. 14.11. Архітектура методу прогнозування

14.1.4.1.5. Обрахунок *KPI*

Вхідні дані:

- A_{r1} – очікувана кількість підписників;
- A_{r2} – очікувана кількість вподобань контенту;
- A_{r3} – очікувана кількість переглядів;
- A_{f1} – фінальна кількість підписників;
- A_{f2} – фінальна кількість вподобань контенту;
- A_{f3} – фінальна кількість переглядів.

Вихідні дані:

- Показник KPI.

На основі цього і проходить розрахунок показника. Це показано у формулі (14.27).

$$KPI = \frac{A_{r1} + A_{r2} + A_{r3}}{A_{f1} + A_{f2} + A_{f3}} \cdot 100\%. \quad (14.27)$$

14.2. Соціальний граф

14.2.1. Основні поняття

Соціальний граф (*Social graph*) — це граф (рис. 14.12 та рис. 14.13), вузли якого представлені соціальними об'єктами, такими як профілі користувача з різними атрибутами (наприклад: ім'я, день народження, рідне місто, тощо), співтовариства, медіа-контент , тощо, а ребра — соціальними зв'язками між ними .

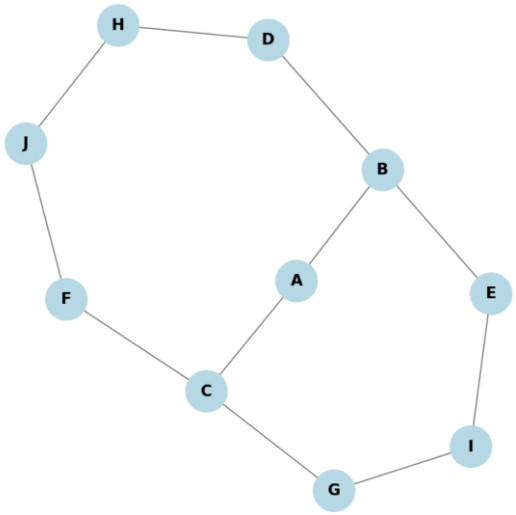


Рис. 14.12. Приклад 1 соціального графу

Соціальний граф, зображений на рис. 14.12, відображає взаємозв'язки між користувачами. Кожен вузол представляє користувача, а ребра показує взаємодію між ними. Це може бути використано для аналізу зв'язків у спільноті, виявлення лідерів думок і центрів впливу.

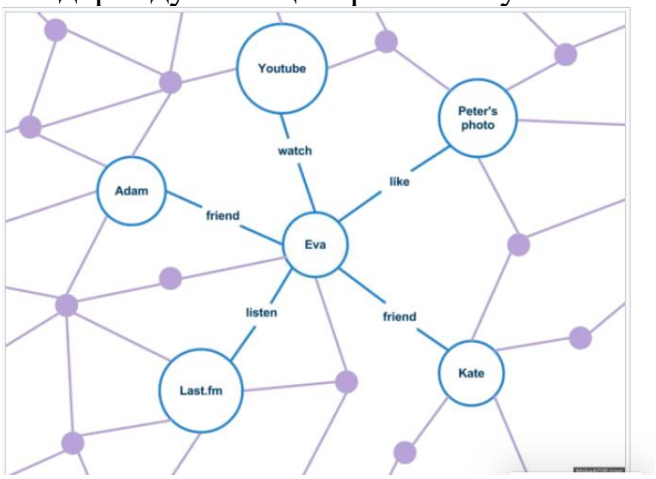


Рис. 14.13. Приклад 2 соціального графу

На даному рис. 14.13 показані в яких стосунках перебувають Різні соціальні об'єкти. Користувач Єва знаходиться в дружніх відносинах з користувачами Адам и Кейт, при цьому Адам та Кейт не є друзями один одному, но у них є спільний друг Єва. Фотографія Пітера булу оцінена багатьма користувачами, в тому числі вона сподобалась и Єві. Також Єва слухає радіо з Last.fm и дивуватися відео з Youtube.

Неявний соціальний граф (*Implicit social graph*) — це такий граф (рис. 14.14), який можна сформувати (вивести, обчислити) на основі взаємодій користувача зі своїми друзями та групами друзів в соціальній мережі. У

цьому графі на відміну від звичайного соціального графа немає явної вказівки друзів, тобто немає явних соціальних зв'язків .

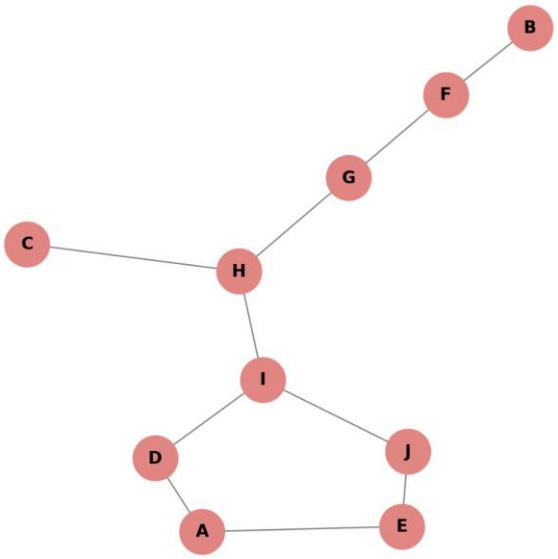


Рис. 14.14. Приклад неявного соціального графу

На рис. 14.14 неявний соціальний граф, що відображає приховані зв'язки між користувачами. Такі зв'язки можуть виникати на основі спільних друзів, інтересів, взаємодій із контентом або непрямих взаємозв'язків у соціальних мережах.

Відмінності між звичайним і неявним соціальним графами

В табл. 14.1 наведено порівняння звичайного та неявного соціального графів.

Таблиця 14.1. Характеристики звичайного та неявного социального графів

Характеристика	Звичайний соціальний граф	Неявний соціальний граф
Тип зв'язків	Прямі взаємодії (дружба, підписка, коментарі)	Непрямі взаємодії (спільні інтереси, поведінкові подібності)
Джерело даних	Видимі дії користувачів	Алгоритмічний аналіз поведінки
Приклад взаємодії	Користувач *A* підписаний на *B*	*A* і *B* лайкають одні й самі ті дописи
Застосування	Візуалізація соціальної мережі, визначення впливових осіб	Рекомендаційні системи, таргетинг реклами, прогнозування поведінки
Тип графа	Читко визначена структура зв'язків	Приховані зв'язки, які виявилися через аналіз даних

Звичайний соціальний граф – застосування для аналізу соціальної структури, впливу лідерів думок, картографування спільнот.

Неявний соціальний граф – використовуйте рекомендаційні алгоритми (Netflix, YouTube, Facebook), де система пропонує контент або друзів на основі поведінки користувачів.

14.2.2. Метрики

Говорячи про завдання на соціальному графі, вживають термін метрики, які в числовій формі відображають характеристики соціальних об'єктів, сегментів/груп об'єктів та їх зв'язків. Ці метрики використовують при проведенні аналізу соціальних мереж.

Особливості соціального графа характеризується такими метриками, як:

- Метрики взаємин;
- Метрики зв'язків;
- Сегментації.

14.2.2.1. Метрики взаємин

Дані метрики подають характер взаємовідносин одного соціального об'єкта з іншими соціальними об'єктами.

14.2.2.1.1. Гомофілія

Гомофілія (*Homophily*) — ступінь, в якій користувач утворює зв'язки з подібними. Подібність може бути визначене за статтю, віком, соціальним станом, освітнім рівнем тощо.

Формула для гомофілії (14.28) в контексті соціальних мереж або взаємодій між індивідами може бути виражена через *коефіцієнт гомофілії* (*homophily coefficient*), що вимірює, наскільки схожі два індивіди за певними характеристиками.

$$H = \frac{\sum_{i,j}\left(A_{ij} \cdot f(x_i, x_j)\right)}{\sum_{i,j} A_{ij}}, \qquad (14.28)$$

де:

H — коефіцієнт гомофілії;

A_{ij} — елемент матриці суміжності графу (1, якщо є зв'язок між індивідами i і j, інакше 0);

$f(x_i, x_j)$ — функція, яка оцінює схожість характеристик індивідів i та j (наприклад, за віком, статтю, ідеологією тощо).

14.2.2.1.2. Множинність

Множинність (*Multiplexity*) — число множинних зв'язків, в яких знаходяться користувачі. Наприклад, два користувача, які товаришують та працюють разом, будуть мати множинність, рівну 2. Множинність пов'язують з силою зв'язку.

Формула (14.29) описує множинність у соціальному графі:

$$M_{ij} = \sum_{k=1}^{n} \left(A_{ik} \cdot A_{kj} \cdot f\left(x_i, x_j, x_k\right) \right), \qquad (14.29)$$

де:

M_{ij} — множинність зв'язку між особами i та j, що враховує наявність серед них проміжних зв'язків;

A_{ik} — елемент матриці суміжності, що вказує на наявність зв'язку між особою i та проміжною особою k;

A_{kj} — елемент матриці суміжності, що вказує на наявність зв'язку між проміжною особою k та особою j;

$f(x_i, x_j, x_k)$ — функція, яка оцінює вплив третіх осіб або контексту на взаємодію між особами i та j через особу k.

14.2.2.1.3. Взаємність

Взаємність (*Mutuality/Reciprocity*) — ступінь, в якій користувачі взаємодіють між собою, відповідають взаємністю на дії один одного.

Формула (14.30) для вимірювання взаємності в соціальному графі може бути виражена через матрицю суміжності A:

$$R_{ij} = \frac{A_{ij} \cdot A_{ji}}{A_{ij} + A_{ji}}, \quad (14.30)$$

де:

R_{ij} — показник взаємності між особами i та j. Значення $R_{ij} = 1$ означає, що зв'язок між i і j є повністю взаємним, а $R_{ij} = 0$ — що зв'язку між ними немає.

A_{ij} — елемент матриці суміжності, який дорівнює 1, якщо є напрямлений зв'язок від i до j, і 0, якщо такого зв'язку немає.

A_{ji} — елемент матриці суміжності, який дорівнює 1, якщо є напрямлений зв'язок від j до i, і 0, якщо такого зв'язку немає.

Інтерпретація:

- Якщо $A_{ij} = A_{ji} = 1$, це означає, що зв'язок між i та j є взаємним, і $R_{ij} = 1$;
- Якщо $A_{ij} = 1$ і $A_{ji} = 0$, то зв'язок односторонній, і $R_{ij} = 0$.

Взаємність може також бути розрахована для всієї мережі як середнє значення взаємності між усіма парами вузлів (14.31):

$$R = \frac{\sum_{i \neq j} R_{ij}}{n \cdot (n-1)}, \quad (14.31)$$

де n — кількість вузлів у графі.

Цей коефіцієнт відображає загальний рівень взаємності в мережі.

14.2.2.1.4. Мережева закритість

Мережева закритість (*Network Closure*) — ступінь, в якій друзі користувача є друзями один одному. Також її називають мірою повноти реляційних тріад. Припущення того, що користувач знаходиться в мережевій закритості, називається *транзитивність*.

Коефіцієнт кластеризації (*Clustering Coefficient*) показує, наскільки вузли схильні утворювати трикутники (тобто чи друзі однієї особи є також друзями між собою).

Формула (14.32) для обчислення локального коефіцієнта кластеризації для вершини i:

$$C_i = \frac{2 \cdot T_i}{k_i \cdot (k_i - 1)}, \quad (14.32)$$

де:

T_i — кількість трикутників, в яких бере участь вузол i;

k_i — ступінь вершини i (кількість її сусідів).

Загальний (середній) коефіцієнт кластеризації для всього графа (14.33):

$$C = \frac{\sum_{i=1}^{n} C_i}{n}, \quad (14.33)$$

де n — загальна кількість вузлів у графі.

14.2.2.1.5. Сусідство

Сусідство (*Propinquity*) — тенденція користувачів мати велику кількість зв'язків з географічно близькими користувачами.

Сусідство в соціальному графі означає набір вузлів, безпосередньо пов'язаних з певним вузлом. У графовій моделі соціальних мереж це відображає людей, з якими конкретний користувач взаємодіє або має зв'язки.

Визначення набору вузлів, пов'язаних з певним вузлом

1. *Множина сусідів вузла*. Якщо задано граф $G = (V, E)$, де:

V — множина вузлів (користувачів);

E — множина зв'язків між ними,

то множина сусідів вузла v визначається як (14.34):

$$N(v) = \{u \in V \mid (v, u) \in E\}. \quad (14.34)$$

Тобто $N(v)$ — це всі вузли, з якими vvv має безпосередній зв'язок.

2. *Ступінь вузла* (*Degree*). Кількість сусідів вузла v називається його ступенем k_v (14.35):

$$k_v = \mid N(v) \mid; \quad (14.35)$$

- *Неорієнтований граф*: k_v— число зв'язків вузла.
- *Орієнтований граф*:

k_v^{in} — кількість вхідних зв'язків (скільки інших вузлів вказують на v);

k_v^{out} — кількість вихідних зв'язків (скільки зв'язків виходить з v).

3. *Спільні сусіди* (*Common Neighbors*). Кількість спільних сусідів для двох вузлів u і v (14.36):

$$CN(u, v) = \mid N(u) \cap N(v) \mid. \quad (14.36)$$

Цей показник використовується в алгоритмах рекомендації друзів у соцмережах.

4. *Міра Жаккара* (*Jaccard Similarity*). Визначає рівень близькості між множинами вузлів. Обчислюється за формулою (6.15) з лекції 6, яка в даному контексті набуде вигляду (14.37).

$$S(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}, \quad (14.37)$$

де:

$N(i)$ — множина сусідів (друзів) вузла i;

$N(j)$ — множина сусідів вузла j.

14.2.2.1.6. Приклад обчислення метрик взаємин в Python

```
import networkx as nx

def compute_relationship_metrics(G, node1, node2):
    """
    Обчислює основні метрики взаємин між двома вузлами в соціальному графі.

    Параметри:
        G (networkx.Graph): Граф (орієнтований або неорієнтований).
        node1, node2: Вузли, між якими обчислюються метрики.

    Повертає:
        dict: Словник із метриками взаємин.
    """

    # Перевірка, чи існує зв'язок між вузлами (Множинність)
    direct_link = G.has_edge(node1, node2) or G.has_edge(node2, node1)

    # Спільні сусіди (Сусідство)
    common_neighbors = list(nx.common_neighbors(G, node1, node2)) if not G.is_directed() else []

    # Міра Жаккара (Сусідство)
    jaccard_coeff = next(nx.jaccard_coefficient(G, [(node1, node2)]), (None, None, 0))[2]

    # Взаємність (для орієнтованих графів)
    reciprocity = 1 if G.has_edge(node1, node2) and G.has_edge(node2, node1) else 0 if G.is_directed() else None

    # Ступені вузлів (Сусідство)
    degree_1 = G.degree(node1)
    degree_2 = G.degree(node2)
```

```
    # Локальний коефіцієнт кластеризації (Закритість)
    clustering_1 = nx.clustering(G, node1)
    clustering_2 = nx.clustering(G, node2)

    return {
        "direct_link": direct_link,
        "common_neighbors": common_neighbors,
        "jaccard_coefficient": jaccard_coeff,
        "reciprocity": reciprocity,
        "degree_node1": degree_1,
        "degree_node2": degree_2,
        "clustering_node1": clustering_1,
        "clustering_node2": clustering_2,
    }

# Приклад використання
graph = nx.Graph()
graph.add_edges_from([(1, 2), (2, 3), (3, 4), (4, 1),
(2, 4)])

metrics = compute_relationship_metrics(graph, 1, 2)
print(metrics)
```

14.2.2.2. Метрики зв'язків

Дані метрики відображають особливості зв'язків, як для окремих соціальних об'єктів, так і для графа в цілому.

14.2.2.2.1. Міст

Міст (*Bridge*) — користувач, чиї слабкі зв'язки заповнюють структурні діри, що забезпечує єдиний зв'язок між іншими користувачами або кластерами (групами користувачів). Також через нього проходитиме найкоротший маршрут.

Міст у соціальному графі — це ребро (зв'язок), яке з'єднує дві підмережі (спільноти) і видалення якого розділити графу або значно зменшити його зв'язок (рис. 14.15).

Для графа:

lua Копировать Редактувати

```
1 -- 2 -- 3 -- 4 -- 5 -- 6
     |    \____/
```

Містами будуть:

- (3 , 4), бо без нього граф розділиться на дві частини.

Рис. 14.15. Приклад визначення мосту

Мости мають велике значення в соціальних мережах, оскільки вони з'єднують окремі групи, сприяючи поширенню інформації та налагодженню зв'язків між спільнотами.

14.2.2.2.2. Центральність

Центральність (*Centrality*) — ступінь, яка показує важливість або вплив певного користувача (кластера користувачів) всередині графа.

Стандартні методи вимірювання центральності

1. *Центральність з посередництва* (*Betweenness Centrality*). Вимірює, як часто вузол знаходиться на найкоротших шляхах між іншими вузлами. Обчислюється за формулою (14.38):

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (14.38)$$

де:

σ_{st} — загальна кількість найкоротших шляхів між вузлами s і t;

$\sigma_{st}(v)$ — кількість таких шляхів, які проходять через вузол v.

Використовується для виявлення «мостів» між спільнотами.

2. *Центральність по близькості* (*Closeness Centrality*). Визначає, наскільки швидко вузол може досягти інших вузлів у мережі. Обчислюється за формулою (14.39):

$$C_C(v) = \frac{1}{\sum_{u \in V} d(v,u)}, \quad (14.39)$$

де $d(v,u)$ — відстань (найкоротший шлях) між вузлами v і u.

Використовується для знаходження інформаційних хабів.

3. *Центральність власного вектора* (*Eigenvector Centrality*). Оцінює вплив вузла, враховуючи вплив його сусідів. Обчислюється за формулою (14.40):

$$C_E(v) = \frac{1}{\lambda} \cdot \sum_{u \in N(v)} C_E(u), \quad (14.40)$$

де λ — найбільше власне значення матриці суміжності графа.

Використовується в алгоритмі PageRank (Google).

4. *Центральність за ступенем* (*Degree Centrality*). Показує, скільки зв'язків має вузол. У неорієнтованому графі обчислюється за формулою (14.41):

$$C_D(v) = \frac{deg(v)}{|V|-1}, \quad (14.41)$$

де:

$deg(v)$ — кількість зв'язків вузла v;

$|V|$ — загальна кількість вузлів у графі.

Для орієнтованого графа розрізняють:

- *Вхідну центральність* (14.42):

$$C_D^{in}(v) = \frac{deg_{in}(v)}{|V|-1}. \quad (14.42)$$

- *Вихідну центральність* (14.43):

$$C_D^{out}(v) = \frac{deg_{out}(v)}{|V|-1}. \quad (14.43)$$

Використовується для пошуку активних або популярнихабо популярних користувачів.

14.2.2.2.3. Густина

Густина (*Density*) — частка прямих зв'язків у мережі по відношенню до загального числа можливих.

Для неорієнтованого графа густина обчислюється за формулою (14.44), яка є аналогічною до формули (14.32):

$$D = \frac{2 \cdot |E|}{|V| \cdot (|V| - 1)}, \quad (14.44)$$

де:

$|E|$ — кількість ребер у графі;

$|V|$ — кількість вузлів у графі.

Для орієнтованого графа (де зв’язки мають напрямок) густина визначається за формулою (14.45):

$$D = \frac{|E|}{|V| \cdot (|V| - 1)}. \quad (14.45)$$

Інтерпретація густини:

- $D \approx 1$ (висока густина) → Граф майже повний (майже всі можливі зв’язки присутні);
- $D \approx 0$ (низька густина) → Граф розріджений (зв’язків дуже мало);
- $D = 1$ → Повний граф (кожен вузол з’єднаний з усіма іншими).

14.2.2.2.4. Відстань

Відстань (*Distance*) — мінімальну кількість зв'язків, необхідних для встановлення наявності взаємозв'язку між двома окремими користувачами.

Відстань між вузлами u і v у графі G позначається як $d(u, v)$ і визначається за формулою (14.46):

$$d(u, v) = min(\text{кількість ребер на шляху від } u \text{ до } v). \quad (14.46)$$

Відстань $d(u, v)$ обчислюється за допомогою алгоритмів:

- Дейкстри;
- Белмана-Форда;
- Флойда-Уоршела.

Докладно про ці алгоритми було розказано в курсі «Дискретна математика».

Інтерпретація відстані:

- Якщо $u = v$, то $d(u, v) = 0$;
- Якщо між u і v немає шляху, то $d(u, v) = +\infty$ (в неорієнтованому графі це означає, що вузли належать до різних компонент зв’язності).

14.2.2.2.5. Структурні діри

Структурні діри (*Structural holes*) — відсутність зв'язків між двома частинами мережі. Тобто, це прогалини в соціальному графі, де деякі групи вузлів (користувачів) погано з'єднані між собою. Вузли, що з’єднують такі групи, можуть отримати перевагу у вигляді контролю інформаційних потоків та впливу на взаємодію між іншими.

Коефіцієнти структурних дір

Деякі метрики допомагають кількісно оцінити структурні діри:

1. *Коефіцієнт обмеженості (Constraint Index).* Вимірює, наскільки вузол залежить від своїх сусідів, і визначається за формулою (14.47):

$$C(v) = \sum_{w \in N(v)} \left(p_{vw} + \sum_{x \in N(v)} (p_{vx} \cdot p_{xw})\right)^2, \quad (14.47)$$

де p_{vw} — відносна вага зв'язку між вузлами v і w.

Інтерпретація коефіцієнту обмеженості:

- Якщо $C(v)$ високе, то вузол залежить від своїх сусідів і не має доступу до структурних дір;
- Якщо $C(v)$ низьке, то вузол займає брокерську позицію і може скористатися перевагами структурних дір.

2. *Ефективний розмір мережі (Effective Size).* Визначає, скільки незалежної інформації отримує вузол v. Обчислюється за формулою (14.48):

$$E(v) = k - \sum_{w \in N(v)} p_{vw}^2, \quad (14.48)$$

де k — кількість зв'язків вузла.

Високе значення $E(v)$ означає, що вузол має доступ до різноманітної інформації.

14.2.2.2.6. Сила зв'язку

Сила зв'язку (*Tie Strength*) визначається лінійною комбінацією часу, близькості та взаємності. Чим більше значення сили зв'язку, тим вона сильніше. Сильні зв'язки визначає гомофілія, сусідство або транзитивність, в той час як слабкі зв'язки визначають мости.

Формалізація сили зв'язку

1. *Класичний підхід.* Визначення сили зв'язку запропонував Марк Грановеттер (1973) у теорії The Strength of Weak Ties. Він розглядав силу зв'язку як функцію (14.49):

$$S(i,j) = f(T, E, I, R), \quad (14.49)$$

де:

T — час, проведений разом;

E — емоційна інтенсивність;

I — інтимність або довіра;

R — взаємна послуга (reciprocal services).

Вищі значення цих параметрів означають міцніші зв'язки.

2. *Графова міра сили зв'язку.* Для кількісного аналізу у графах використовується формула (14.50):

$$S(i,j) = w_{ij}, \quad (14.50)$$

де w_{ij} — вага ребра між вузлами i та j (наприклад, кількість взаємодій, повідомлень, спільних друзів).

3. *Локальна кластеризація. (Принцип триадного замикання).* Якщо у вузлів є багато спільних друзів, їхній зв'язок, ймовірно, міцніший. Значення $S(i,j)$ обчислюється аналогічно до коефіцієнту Жакарра (6.15), яка в даному контексті набуде вигляду (14.37).

14.2.2.1.4. Приклад обчислення метрик зв’язків в Python

```
import networkx as nx

# Створюємо зважений граф (соціальна мережа)
G = nx.Graph()
edges = [
    (1, 2, 0.9), (1, 3, 0.5), (2, 3, 0.7), (2, 4,
0.4),
    (3, 4, 0.8), (3, 5, 0.3), (4, 5, 0.6)
]
G.add_weighted_edges_from(edges)

# Функція для обчислення сили зв’язку між вузлами
(Сила з’язку)
def tie_strength(G, node1, node2):
    return G[node1][node2]['weight'] if
G.has_edge(node1, node2) else 0

# Обчислення коефіцієнтів зв’язку (Сила з’язку)
for u, v in G.edges():
    print(f"Сила зв’язку між {u} і {v}:
{tie_strength(G, u, v)}")

# Коефіцієнт кластеризації для кожного вузла (Сила
зв’язку)
clustering_coeffs = nx.clustering(G, weight="weight")
print("\nКоефіцієнти кластеризації:",
clustering_coeffs)

# Обчислення найкоротших шляхів між вузлами
(Відстані)
shortest_paths =
dict(nx.all_pairs_shortest_path_length(G))
print("\nНайкоротші шляхи між вузлами:")
for source, targets in shortest_paths.items():
    for target, length in targets.items():
        print(f"Відстань між {source} і {target}:
{length}")
```

14.2.2.3. Сегментація

Дані метрики відображають характеристики соціального графа, поділеного на сегменти, які мають відмінні риси.

В якості метрики сегментації знову використовують *коефіцієнт кластеризації* (*Clustering coefficient*) — ймовірність того, що два різних користувача пов'язані з конкретним індивідуумом. Високий коефіцієнт кластеризації вказує на високу замкнутість групи, іншими словами, група може бути клікою.

Дана метрика розглядалася вище. Інші метрики сегментації розглянемо детальніше.

14.2.2.3.1. Кліка

Означення кліки

Кліка (*Cliques*) — група, в якій всі користувачі мають прямі зв'язки (вершини пов'язані (з'єднані) ребром) один до одного.

Формула для кліки в графі залежить від того, як її визначають у контексті теорії графів. У простому вигляді для визначення кліки в графі можна використовувати такі поняття:

Повна підмножина вершин графа $G = (V, E)$, де V — множина вершин, а E — множина ребер.

Кліка — це підмножина вершин $C \subseteq V$, для якої виконується умова: для кожної пари різних вершин $u, v \in C$, існує ребро $(u, v) \in E(u\ v)$, тобто кожна пара вершин у кліці з'єднана прямим ребром (рис. 14.16).

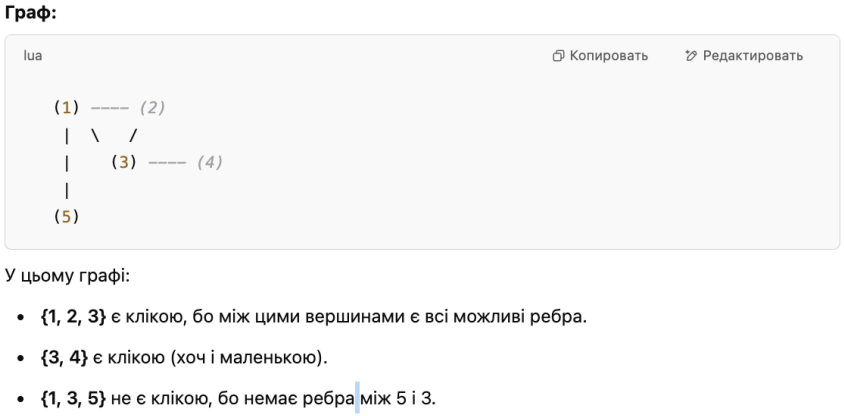


Рис. 14.16. Приклад кліки

Формально, *кліка C з k вершинами* — це підмножина вершин, для якої виконуютєся умова (14.51):

$$\forall u, v \in C, u \neq v \Rightarrow (u, v) \in E. \quad (14.51)$$

Розмір кліки

Якщо кліка містить k вершин, то кількість ребер r у цій кліці буде дорівнювати C_n^2 (14.52).

$$r = C_n^2 = \frac{k \cdot (k-1)}{2}. \quad (14.52)$$

Виявлення максимальної кліки

Щодо пошуку максимальної кліки у графі, це є NP-складною задачею, і для великих графів використовують різні апроксимаційні або евристичні методи.

14.2.2.3.2. Соціальне коло

Соціальне коло (*Social circles*) — група, в якій не обов'язкові прямі зв'язки між користувачами.

Тобто, *соціальне коло* — це концепція, яка описує групу людей, з якими індивід підтримує соціальні зв'язки. Це коло включає в себе людей, з якими індивід регулярно спілкується, взаємодіє або має емоційні зв'язки. Воно може варіюватися за розміром і складом і включати родину, друзів, колег, знайомих і навіть людей, з якими особа взаємодіє в рамках соціальних мереж або спільнот.

Соціальне коло може бути:

- *Маленьким* – це тільки найближчі родичі або друзі;
- *Більшим* – соціальне коло, що включає більше знайомих, колег, однодумців.

Основні аспекти соціального кола:

- *Центр соціального кола* – це особа або група осіб, з якими індивід має найбільш тісні зв'язки;
- *Оточення соціального кола* – це всі інші люди, з якими індивід взаємодіє менш інтенсивно, але все ж підтримує зв'язок.

Формування соціального кола:

- Соціальне коло формуються через спільні інтереси, місце проживання, навчання, роботу або інші соціальні та культурні зв'язки. Це коло може бути стабільним або змінюватися в залежності від обставин у житті людини.

У соціальних мережах:

- У контексті онлайн-платформ (наприклад, Facebook, Instagram) соціальне коло може бути представлене як набір друзів, підписників або контактів. Тут також можна виділити різні рівні взаємодії: найближчі контакти (друзі, сім'я) і більш віддалені (підписники, знайомі).

14.2.2.3.3. Згуртованість

Згуртованість (*Cohesion*) — ступінь, в якій користувачі пов'язані між собою одним, загально-з'єднаним зв'язком, утворюючи соціальну згуртованість. Структурна згуртованість — вказує на таку єдину структуру групи, що видалення невеликої кількості користувачів веде до розриву групи.

Згуртованість в соціальних мережах і графах — це міра того, наскільки добре взаємопов'язані елементи графа або соціальної мережі між собою. Це важливий показник для вивчення соціальних структур, який відображає, наскільки щільно з'єднані вершини графа або учасники соціальної мережі.

Визначення згуртованості

У теорії графів *згуртованість графа* визначається через наявність шляхів між усіма парами вершин:

- Граф вважається зв'язним (згуртованим), якщо для будь-якої пари вершин існує шлях між ними;

- Якщо граф складається з кількох компонентів, які не з'єднані між собою, то граф вважається незгуртованим.

У контексті соціальних мереж *згуртованість* може оцінюватися через рівень взаємодії між членами мережі:

- Якщо велика частина учасників активно взаємодіє між собою, соціальна мережа буде згуртованою;
- Висока згуртованість свідчить про наявність сильних соціальних зв'язків між учасниками, що може бути корисним для вирішення завдань співпраці, організації або колективної дії.

Формула для коефіцієнта згуртованості вершини v є аналогічною до формул (14.32) та (14.44). Вона має вигляд (14.53):

$$C(v) = \frac{2 \cdot E(v)}{k_v \cdot (k_v - 1)}, \quad (14.53)$$

де:

$E(v)$ — кількість реальних ребер між сусідами вершини v;

k_v — кількість сусідів вершини v.

14.2.2.3.4. Приклад обчислення метрик сегментації (згуртованості) в Python

```
import networkx as nx

# Створюємо граф (можна використовувати як неорієнтований граф)
G = nx.Graph()

# Додаємо вузли (користувачів у соціальній мережі)
G.add_edges_from([
    (1, 2), (1, 3), (2, 3),  # трикутник між 1, 2, 3
    (3, 4), (4, 5), (5, 6), (6, 3),  # ще одна структура
    (2, 6)  # зв'язок між двома частинами графа
])

# 1. Обчислення коефіцієнта згуртованості для всього графа
clustering_coeff = nx.average_clustering(G)
print(f"Середній коефіцієнт згуртованості: {clustering_coeff:.4f}")

# 2 Знаходимо всі кліки
cliques = list(nx.find_cliques(G))
print("Знайдені кліки:", cliques)
```

14.3. Моделі соціального графу

Для вирішення завдань на соціальному графі використовуються *спеціальні моделі*, за допомогою яких можна замінити реальні графи. За допомогою моделей соціальних графів вирішують такі завдання, як:

- Ідентифікація користувачів;
- Соціальний пошук;
- Генерація рекомендацій з вибору друзів, медіа-контенту, новин, тощо;
- Виявлення реальних зв'язків або збір відкритої інформації для моделювання графа.

Обробка даних соціальних графів пов'язана з низкою проблем, як наприклад відмінності соціальних мереж, закритість соціальних даних.

14.3.1. Класифікація моделей соціального графу

Моделі соціального графу класифікуються за різними критеріями, зокрема:

- Випадковість;
- Локальна структура;
- Масштабованість;
- Наявність спільнот.

Класифікація з точки зору керованості вузлами графу

Класифікація з точки зору керованості вузлами графу (взаємодії між вузлами, структури зв'язків між ними) представлена на рис. 14.17.

Рис. 14.17. Класифікація моделей соціальних графів з точки зору керованості

1. *Функціонально-керовані моделі.*

Функціонально-керовані моделі соціальних графів будуються на основі певних правил взаємодії між вузлами, що відображають реальні соціальні процеси. Вони відрізняються від класичних математичних моделей тим, що зосереджені на динаміці та поведінкових факторах.

До цього класу моделей належать:

- Модель Барабаші-Альберта;
- Модель «Палаючий ліс».

Дані моделі буде описано нижче.

2. *Структурно-керовані моделі.*

Структурно-керовані моделі будуються на основі заданих правил формування зв’язків між вузлами. Вони відрізняються від випадкових і функціонально-керованих моделей тим, що використовують структурні закономірності для моделювання соціальних мереж.

До цього класу моделей належать:

- DK-графи;
- Графи Кронекера.

DK-графи (*Densification Kronecker Graphs*) – це клас моделей, що використовуються для генерації та аналізу великих соціальних графів, особливо таких, що з часом ущільнюються (densify). Вони ґрунтуються на Kronecker Product Graph Model і враховують емпірично спостережувані закономірності в реальних мережах.

Графи Кронекера є моделями побудови графів, що базуються на добутку Кронекера матриць суміжності, що дозволяє створювати масштабовані та реалістичні соціальні мережі з певними властивостями. Вони широко використовуються в аналізі великих графів, таких як соціальні мережі, комунікаційні мережі, інтернет-графи, оскільки можуть точно моделювати реальні структури з великою кількістю вузлів і зв'язків.

3. *Навмисне-керовані моделі.*

Навмисне-керовані моделі будуються на основі заданих стратегій та цілей, що визначають, як утворюються та змінюються зв’язки в соціальних мережах. Вони застосовуються там, де необхідно контролювати або впливати на структуру графа, наприклад, у пропаганді, маркетингових кампаніях або соціальному управлінні.

До цього класу моделей належать:

- Випадкові блукання;
- Найближчій сусід.

Випадкові блукання є важливою концепцією у теорії графів і моделюванні динамічних систем, що описують рух у середовищі, де кожен крок є випадковим. Це може бути застосовано для моделювання численних процесів, таких як поширення інформації, рух молекул у середовищі, або навіть у аналізі соціальних мереж та комунікаційних мережах.

Найближчий сусід є поняттям, яке широко використовується в теорії графів, машинному навчанні та обробці даних. Це один із основних принципів, що визначає зв'язок між елементами або вузлами в графі чи просторі. Найближчий сусід — це елемент, який знаходиться на мінімальній відстані від заданого елемента, згідно з якимось критерієм.

14.3.2. Опис моделей соціального графу

Перейдемо до більш детального опису деяких з цих моделей.

Основні моделі соціального графа

1. *Випадковий граф* (*Erdős–Rényi Model, ER*).

Опис:

- Усі зв’язки між вузлами утворюються випадково з ймовірністю p;
- Добре підходить для моделювання випадкових взаємодій (наприклад, випадкові знайомства).

Недолік:

- Не відображає реальних соціальних мереж, де зв’язки часто групуються.

Приклад створення:

```
import networkx as nx
import matplotlib.pyplot as plt

# Випадковий граф із 10 вершинами та ймовірністю з'єднання 0.3
G = nx.erdos_renyi_graph(10, 0.3)

# Візуалізація
nx.draw(G, with_labels=True, node_color="skyblue", edge_color="gray")
plt.show()
```

2. *Модель малого світу* (*Watts–Strogatz Model, WS*).

Опис:

- Базується на ідеї, що в соціальних мережах є короткі шляхи між будь-якими двома людьми;
- Створюється із регулярного графа, у якому випадково перемішуються деякі зв’язки;
- Відображає теорію «шість рукостискань», де всі люди пов'язані короткими шляхами.

Приклад створення:

```
#Граф малого світу з 10 вершинами, кожна має 2 сусідів, з ймовірністю перемішування 0.3
G = nx.watts_strogatz_graph(10, 2, 0.3)

nx.draw(G, with_labels=True, node_color="lightgreen", edge_color="gray")
plt.show()
```

3. *Безмасштабна модель* (*Barabási–Albert Model, BA*).

Опис:

- Імітує ефект популярності (чим більше зв’язків у вузла, тим ймовірніше до нього приєднаються нові вузли).

- Добре описує соцмережі, де є «інфлюенсери» (кілька дуже популярних людей).

Приклад створення:

```
# Безмасштабна модель із 10 вузлами, де кожен новий
вузол з'єднується з 2 існуючими
G = nx.barabasi_albert_graph(10, 2)

nx.draw(G, with_labels=True, node_color="orange",
edge_color="gray")
plt.show()
```

4. *Соціальна модель на основі спільнот* (*Stochastic Block Model, SBM*).

Опис:

- Використовується для моделювання кластерів або груп людей у соціальному графі;
- Відповідає ситуаціям, де соцмережа розбивається на окремі спільноти (наприклад, групи друзів у Facebook).

Приклад створення:

```
import numpy as np

# Визначаємо кількість вершин у кожній групі
sizes = [5, 5]  # Дві групи по 5 вузлів
probs = [[0.8, 0.1], [0.1, 0.7]]  # Ймовірності
зв'язків між групами

G = nx.stochastic_block_model(sizes, probs)

nx.draw(G, with_labels=True, node_color="purple",
edge_color="gray")
plt.show()
```

Модель «Палаючий ліс» (*Forest Fire Model, FFM*).

Опис:

- Модель «Палаючий ліс» (була запропонована як спосіб моделювання динаміки росту соціальних мереж. Вона імітує процес поширення вогню в лісі, де нові вузли «запалюють» інші вузли, утворюючи зв’язки, подібно до того, як вогонь поширюється між деревами;
- Ця модель є альтернативою моделі преференційного приєднання (Barabási-Albert) і дозволяє створювати графи з реалістичнішими кластерними структурами та нерівномірним розподілом зв’язків.

Принцип роботи моделі:

1. Додаємо новий вузол у граф;
2. Випадково вибираємо «точку займання» (початковий сусід);
3. Для кожного сусіда з ймовірністю p вузол «запалює» наступні рівні сусідів.
4. Процес триває, поки не припиниться поширення.

Приклад створення:

```
import networkx as nx
import random
import matplotlib.pyplot as plt

def forest_fire_graph(n, p=0.3):
    """Створює граф за моделлю 'Палаючий ліс'"""
    G = nx.DiGraph()  # Орієнтований граф
    G.add_node(0)  # Перший вузол

    for new_node in range(1, n):
        visited = set()
        frontier = [random.choice(list(G.nodes))]  #
Випадковий початковий вузол

        G.add_node(new_node)

        while frontier:
            current = frontier.pop()
            if current not in visited:
                visited.add(current)
                G.add_edge(new_node, current)

                # Ймовірність "спалити" сусідів
                for neighbor in
list(G.successors(current)):
                    if random.random() < p:
                        frontier.append(neighbor)

    return G

# Генеруємо граф із 50 вузлами
G = forest_fire_graph(50, p=0.3)

# Візуалізація
plt.figure(figsize=(8, 6))
```

```
nx.draw(G, with_labels=True, node_color="orange",
edge_color="gray")
plt.show()
```

14.3.3. Вибір моделі для соціального графа

Рекомендації щодо вибору моделі соціального графу наведено в табл. 14.2.

Таблиця 14.2. Порівняння властивостей моделей соціального графу та приклади їх застосування

Тип моделі	Властивості	Приклад застосування
Випадкові графи (ER)	Випадкові зв’язки, без спільнот	Тестування соцмереж, випадкові взаємодії
Моделі малого світу (WS)	Короткі відстані, висока кластеризація	Поширення новин, вірусний маркетинг
Безмасштабні графи (BA)	Є «хаби», степеневий розподіл	Виявлення інфлюенсерів, аналіз соцмереж
Моделі спільнот (SBM)	Чітке групування вузлів у спільноти	Аналіз груп у соцмережах, політичні дослідження
Модель «Палаючий ліс» (FFM)	Цей механізм дозволяє отримати реалістичні соціальні мережі з локальними спільнотами, а також центральними вузлами з високим ступенем.	Аналіз вірусного маркетингу, моделювання епідемій, моделювання атак у комп’ютерних мережах

ЛЕКЦІЯ 15

АНАЛІЗ ДАНИХ В ТЕОРІЇ ПРИЙНЯТТЯ РІШЕНЬ

15.1. Однокритеріальна задача прийняття рішень

15.1.1. Поняття однокритеріальної задачі прийняття рішень

Однокритеріальна задача прийняття рішень – це задача, в якій потрібно вибрати один варіант з множини альтернатив на основі одного критерію чи метрики. Це найпростіший тип задачі прийняття рішень, де відсутня необхідність порівнювати альтернативи за різними аспектами або критеріями.

Наприклад, розглянемо ситуацію, коли ми вибираємо постачальника товарів для нашого підприємства. Якщо є один критерій, який є ключовим для нас, наприклад, ціна, то ми можемо вибрати постачальника, який пропонує найнижчу ціну.

Таким чином, однокритеріальна задача прийняття рішень допомагає спростити процес вибору, оскільки вона базується лише на одному критерії або метриці. Однак в реальному житті часто зустрічаються багатокритеріальні задачі, де потрібно враховувати різні аспекти альтернатив.

15.1.2. Загальна постановка однокритеріальної задачі прийняття рішень

Нехай результат керованого заходу залежить від вибраного рішення (стратегії управління) і деяких невипадкових фіксованих чинників, повністю відомих особі, що приймає рішення. Стратегії управління можуть бути представлені у вигляді значень n-мірного вектора $X = (x_1, x_2, \dots, x_n)$, $i = \overline{1,n}$, на компоненти якою накладені обмеження, обумовлені рядом природних причин і що мають вигляд (15.1):

$$g_j = g_j(A_j, X)\{\leq, =, \geq\} b_j; \quad j = \overline{1,m};\ m\{<, =, >\} n, \tag{15.1}$$

де A_j – деякий масив фіксованих невипадкових параметрів.

Умови (15.1) визначають область Ω_X допустимих значень стратегій X. Ефективність управління характеризується деяким чисельним критерієм оптимальності F (15.2):

$$F = F(X, C), \tag{15.2}$$

де C — масив фіксованих, невипадкових параметрів.

Масиви A_j і C характеризують властивості об'єктів, що беруть участь в управлінні, і умови, за яких відбувається управління. Перед особою, що приймає рішення, коштує завдання вибору такого значення $\overline{X} = (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)$ вектора управління $X = (x_1, x_2, \dots, x_n)$ з області Ω_X його

допустимих значень, яке максимізувало значення критерію оптимальності F, а також значення $\overline{F}$ цього максимуму (15.3)

$$F = F(X, C) = \max_{x \in \Omega_X} F(X, C), \qquad (15.3)$$

де область Ω_X представляється умовою (15.1).

У (15.3) символи $\overline{F}$ і $\overline{X}$ позначають максимально досяжне в умовах (15.1) значення критерію оптимальності F і відповідне йому оптимальне значення вектора управління X. Сукупність співвідношень (15.1), (15.2) і (15.3) є загальним виглядом математичної моделі однокритеріальної статичної детермінованої ЗПР. Задача в такій постановці повністю збігається із загальною постановкою задачі математичного програмування. Тому весь арсенал методів, розроблених для вирішення задач математичного програмування, може бути використаний для вирішення задач прийняття рішень даного класу.

Розглянемо приклад однокритеріальної статичної детермінованої ЗПР.

Хай необхідно відображувати деяку кількість інформаційних моделей (наприклад, картографічну інформацію). Для відображення будь-якої з моделей завжди потрібно вирішити n різних задач $З_1, З_2, \dots, З_n$ (відображення символів, відображення векторів, поворот і переміщення зображень, масштабування і тому подібне). Всі завдання взаємно незалежні. Для вирішення цих задач можуть бути використані m різних мікропроцесорів $M_1, M_2, \dots, M_m$. Протягом часу T мікропроцесор M_j може вирішити a_{ij} задач типу $З_i$ $(i = \overline{1,n}; j = \overline{1,m})$, тобто вирішити задачу $З_i$, кілька разів по одному і тому ж алгоритму, але для різних вихідних даних.

Інформаційну модель можна відображати лише у тому випадку, коли вона містить повний набір результатів вирішення всіх задач $З_1, З_2, \dots, З_n$.

Потрібно розподілити задачі по мікропроцесорам таким чином, щоб кількість інформаційних моделей, синтезованих за час T, була максимальною. Інакше кажучи, необхідно вказати, яку частину часу T мікропроцесор M_j повинен займати вирішенням задачі $З_i$.

Позначимо цю величину через x_{ij} (якщо ця задача не буде вирішуватися на даному мікропроцесорі, то $x_{ij} = 0$).

Очевидно, що загальний час забавною кожного мікропроцесора вирішенням цих задач не повинен перевищувати загального запасу часу T, «доля» — одиниці. Таким чином, маємо наступні обмежувальні умови:

$$\sum_{i=1}^{n} x_{ij} < 1; j = \overline{1,m}.$$

Загальна кількість рішень N_i задачі $З_i$, що отримані всіма мікропроцесорами,

$$N_i = \sum_{j=1}^{m} (a_{ij} \cdot x_{ij});\ i = \overline{1,n}.$$

Оскільки інформаційна модель може бути синтезована лише з повного набору результатів вирішення всіх задач, то кількість інформаційних моделей F буде визначатися мінімальним з чисел N_i.

Отже, розглянемо наступну математичну модель: потрібно знайти такі x_{ij}, щоб оберталася в максимум функція F

$$F = \max_i\left(\sum_{j=1}^{m}\left(a_{ij}\cdot x_{ij}\right)\right);\ i=\overline{1,n},$$

при

$$\sum_{i=1}^{n} x_{ij} \le 1;\ j=\overline{1,m}, \qquad x_{ij}\ge 0.$$

15.1.3. Загальна постановка однокритеріальної статичної задачі прийняття рішень в умовах ризику

Як зазначалося, кожна обрана стратегія управління в умовах ризику пов'язана з множиною можливих результатів, причому кожен результат має певну ймовірність появи, яка заздалегідь відома ОПР.

При оптимізації розв'язку в подібній ситуації стохастичну ЗПР зводять до детермінованої. При цьому широко використовуються два принципи: штучне зведення до детермінованої схеми та оптимізація в середньому.

В першому випадку ймовірнісна картина явища приблизно замінюється детермінованою. Для цього всі випадкові фактори, що враховуються в задачі, приблизно замінюються якимись невипадковими характеристиками цих факторів (як правило, їхнім математичним сподіванням).

Цей підхід використовується в грубих, орієнтовних розрахунках, а також в тих випадках, коли діапазон можливих значень випадкових величин порівняно малий. В тих випадках, коли показник ефективності управління лінійно залежить від випадкових параметрів, цей підхід призводить до того ж результату, що й «оптимізація в середньому».

Підхід «оптимізація в середньому» полягає у переході від вихідного показника ефективності Q, що є випадковою величиною:

$$Q = Q\left(X, A, y_1, y_2, \dots, y_q\right),$$

де:

X — вектор управління;

A — масив детермінованих факторів;

$y_1, y_2, \dots, y_q$ — конкретні реалізації випадкових фіксованих факторів $Y_1, Y_2, \dots, Y_q$ до його усередненої, статистичної характеристики, наприклад до його математичного сподівання $M(Q)$ (15.4):

$$F = M(Q) = \underbrace{\iint \dots \int}_{q}\left(Q\left(X, A, y_1, y_2, \dots, y_q\right)\cdot f\left(y_1, y_2, \dots, y_q\right)\right) dy_1, dy_2, \dots, dy_q = F(X, A, B). \qquad (15.4)$$

Тут B – масив відомих статистичних характеристик випадкових величин $Y_1, Y_2, \dots, Y_q$; $f(y_1, y_2, \dots, y_q)$ – закон розподілу ймовірностей випадкових величин $Y_1, Y_2, \dots, Y_q$.

При оптимізації в середньому по критерію (15.4) в якості оптимальної стратегії $\overline{X}$ буде обрана така стратегія, яка задовольняючи обмеженням на область Ω_X допустимих значень вектора X, максимізує значення математичного сподівання $F = M(Q)$ вихідного показника ефективності Q, тобто справедлива рівність (15.5):

$$\overline{F} = F(\overline{X}, A, B) = max\, F(X, A, B) = \max_{X \in \Omega_X} M\left(Q(X, A, y_1, y_2, \dots, y_q)\right). \quad (15.5)$$

В тому випадку, коли число можливих стратегій i скінченне $i = \overline{1, I}$ і число можливих результатів j скінченне $(j = \overline{1, J})$ то вираз (15.5) переписується у вигляді (15.6):

$$F = F(\overline{X}) = \max_{1 \le i \le I} F(X_i) = \max_{1 \le i \le I} \left(\sum_{j=1}^{J} (P_{ij} \cdot Q_{ij})\right), \quad (15.6)$$

де:

Q_{ij} — значення показника ефективності управління у випадку появи j-го результату при виборі i-ї стратегії управління;

P_{ij} — ймовірність появи j-го результату при реалізації i-ї стратегії.

З виразів (1.5) і (1.6) випливає, що оптимальна стратегія X призводить до гарантованого найкращого результату лише при багатократному повторенні ситуації в однакових умовах. Ефективність кожного окремого вибору пов'язана з ризиком і може відрізнятися від середньої величини як в кращій, так і в гірший бік.

Порівняння двох розглянутих принципів оптимізації в стохастичних ЗПР показує, що вони є детермінізацією вихідного завдання на різних рівнях впливу стохастичних чинників. «Штучне зведення до детермінованої схеми» є детермінізацією на рівні чинників, «оптимізація в середньому» — на рівні показника ефективності.

Після виконання детермінізації можуть бути використані всі методи, застосовні для вирішення однокритеріальних статичних детермінованих ЗПР.

Розглянемо приклад однокритеріальної статичної задачі прийняття рішень в умовах ризику.

Для створення картографічної бази даних необхідно кодувати картографічну інформацію. Використання поелементного кодування призводить до необхідності використання надзвичайно великих об'ємів пам'яті. Відомий ряд методів кодування, що дозволяють істотно скоротити необхідний об'єм пам'яті (наприклад, лінійна інтерполяції, інтерполяція класичними многочленами, кубічні сплайни і т.д). Основним показником ефективності методу кодування є коефіцієнт стискування інформації. Проте значення цього коефіцієнту залежить від вигляду кодованої картографічної

інформації (гідрографія, кордони адміністративних районів, дорожня мережа і т. д). Позначимо через $Q_{ij}(i = \overline{1,n},\ j = \overline{1,m})$ значення коефіцієнту стиснення i-го методу кодування для j-ї інформації. Конкретний район, що підлягає кодуванню, заздалегідь невідомий. Проте попередній аналіз картографічної інформації всього регіону і досвід попередніх розробок дозволяють обчислити ймовірність появи кожного з видів інформації. Позначимо через P_j, ймовірність появи j-го виду,

$$\sum_{j=1}^{m} P_j = 1.$$

Тоді, використовуючи метод оптимізації в середньому, слід обрати такий метод кодування, для якого

$$F = \max_i\left(F(X_i)\right) = \max_i\left(\sum_{j=1}^{m}\left(P_j \cdot Q_{ij}\right)\right);\ i = \overline{1,n}.$$

15.1.4. Критерії прийняття рішень в умовах ризику

Прийняття рішень при відомих апріорних ймовірностях

Прийняття рішень при відомих апріорних ймовірностях — це ситуація в теорії прийняття рішень, коли ймовірності виникнення можливих станів природи (середовища) відомі заздалегідь (тобто апріорно).

Суть методу полягає у використанні кількісних критеріїв оптимальності, зокрема:

1. *Критерій Байєса (максимізації очікуваної вигоди).*

Обирається альтернатива, яка забезпечує максимальне середнє значення виграшу з урахуванням ймовірностей станів.

Отже, будемо позначати ймовірності гіпотез:

$$Q_1 = p(P_1), Q_2 = p(P_2), \ldots, Q_m = p(P_m),$$

які полягають у тому, що природа буде знаходитися в стані S_j $(j = \overline{1,m})$ – це j-й результат при виборі деякої стратегії управління.

Таким чином, ми вважаємо, що ймовірності відомі до того, як ми вирішили прийняти рішення.

Рішення – вибір оптимальної стратегії.

Говорять, що це ситуація *ідеального спостерігача*.

В якості критерію оптимізації обираємо середній виграш, який ми отримаємо, якщо оберемо стратегію A_i (15.7).

$$\overline{a_i}=\sum_{j=1}^{m}\left(a_{ij} \cdot Q_j\right)\ (i = \overline{1,n},\ j = \overline{1,m}), \qquad (15.7)$$

де:

$\overline{a_i}$ — очікувана вигода для стратегії A_i;

Q_j — ймовірність стану природи S_j;

a_{ij} — виграш при застосуванні стратегії A_i в стані природи S_j.

Це аналог математического сподівання. Рішення приймається за критерієм (15.8):

$$\max_i(\overline{a_i}). \qquad (15.8)$$

2. *Мінімізації очікуваного ризику* (як середнього збитку).

Якщо задана матриця ризиків, то можливо для кожного рядка обчислити середній ризик (15.9):

$$\overline{r_i}=\sum_{j=1}^{m}(r_{ij}\cdot Q_j), \quad (15.9)$$

де:

$\overline{r_i}$ — очікуваний ризик для стратегії A_i;

Q_j — ймовірність стану природи S_j;

r_{ij} — програш при застосуванні стратегії A_i в стані природи S_j.

Це аналог середньоквадратичного відхилення. Рішення приймається за критерієм (15.10):

$$\min_i(\overline{r_i}). \quad (15.10)$$

Покажемо, що *оптимальне рішення можна шукати як по* (15.9)*, так і по* (15.10).

$$\overline{a_i}+\overline{r_i}=\sum_{j=1}^{m}(a_{ij}\cdot Q_j)+\sum_{j=1}^{m}(r_{ij}\cdot Q_j)=\sum_{j=1}^{m}(a_{ij}\cdot Q_j)+\sum_{j=1}^{m}(\beta_j-a_{ij})\cdot Q_j=\sum_{j=1}^{m}(\beta_j\cdot Q_j)=c.$$

Тоді:

$$\overline{a_i}+\overline{r_i}=c\Rightarrow\overline{a_i}=c-\overline{r_i},$$

де c – це величина незалежна від i, тобто стала величина для даного рядка.

Таким чином:

$$\max_i(\overline{a_i})=\min_i(\overline{r_i});\ argmax_i(\overline{a_i})=argmin_i(\overline{r_i}).$$

3. *Принцип адаптації до умов.*

В результаті попередніх розв'язків (максимізація прибутки чи мінмізація ризиу) вибирається чиста стратегія A_i, що має очікувану вигоду $\overline{a_i}$.

Чиста стратегія — це стратегія в теорії ігор або прийнятті рішень, яка передбачає вибір певної дії з повною впевненістю, без випадковості. У кожній ситуації гравець або особа приймає рішення обирає одну конкретну дію.

Її основні характеристики:

- *Однозначність.* Завжди обирається одна й та сама дія в певній ситуації.
- *Детермінованість.* Не використовує ймовірностей.
- *Протилежність змішаній стратегії.* Вибір дії не здійснюється випадково відповідно до певного розподілу ймовірностей.

Якщо ми змішаємо наші чисті стратегії з ймовірностями p_i, тоді в результаті отримаємо середній виграш у вигляді:

$$\overline{\overline{a}}=\overline{a_1}\cdot p_1+\overline{a_2}\cdot p_2+\cdots+\overline{a_n}\cdot p_n, \quad (15.11)$$

де $\overline{\overline{a}}$ математичне сподівання.

Очевидно, що якщо математичне сподівання менше максимального значення ($\overline{\overline{a}}<\overline{a_{max}}$), то:

- У грі з природою немає сенсу змішувати стратегії;

- Чиста стратегія забезпечує найкращий результат.

Говорять, що ми цю систему навчаємо. Такий підхід називається *принципом адаптації до умов*.

Прикладом даного принципу є «Задача про секретарку» (буде розглянута тижче).

Принцип недостатності підстави

Якщо апріорну ймовірність вивчити не вдається, то застосовується *принцип недостатності підстави*, що має наступні особливості:

- Якщо невідома ймовірність, то вважаємо, що гіпотези природи рівноймовірні;
- Після цього застосовується критерій ідеального спостерігача;
- Оскільки при рівних ймовірностях ентропія (невизначеність) максимальна, то застосовується принцип песимізму;
- Якщо самі значення ймовірностей невідомі, але є інформація про переваги гіпотез, то використовуються методи обробки переваг і здобуття ймовірностей.

Якщо ймовірності гіпотез відносяться як:

$$m:(m-1):(m-2):\ldots:2:1 = Q_1:Q_2:\ldots:Q_m;\ \sum_{j=1}^{m} Q_j = 1,$$

то можна обчислити саму величину ймовірності у вигляді:

$$Q_j = \frac{2\cdot(m-j+1)}{m\cdot(m+1)};\ j = \overline{1,m.}$$

В деяких випадках враховується не тільки середній виграш, але й середньоквадратичне відхилення, тобто величина розкиду виграшу в кожному рядку (ризик втрати прибутку в результаті застосування стратегії a_i) (15.12):

$$\max_i\left(\bar{a}_i - k\cdot\sigma_{a_i}\right) \qquad (15.12)$$

або (15.13):

$$\sigma_{a_i} = \sqrt{D_{a_i}}, \qquad (15.13)$$

де D_{a_i} – дисперсія.

Тобто з (15.13) маємо (15.14):

$$\min_i(\sigma(a_i)). \qquad (15.14)$$

15.1.5. Приклади розв'язання задач прийняття рішень в умовах ризику

15.1.5.1. Формування рекомендацій щодо доцільності інвестицій в акції фірми

Постановка задачі

Інвестиційний фонд розглядає можливість придбання акцій фірм «A», «B» та «C». Очікуваний прибуток по акціям та відповідні ймовірності наведено в табл. 15.1.

1. Визначити доходність акцій фірм.
2. Визначити ризик по акціям фірм.
3. Дати свої рекомендації щодо доцільності їх придбання.

Таблиця 15.1. Статистичні дані

Фірма «A»		Фірма «B»		Фірма «C»	
Прибутковість, %	Ймовірність	Прибутковість, %	Ймовірність	Прибутковість, %	Ймовірність
4	0,25	4	0,25	5	0,2
9	0,5	8	0,25	15	0,6
11	0,25	11	0,5	20	0,2

Розв'язання

1. Знайдемо математичне сподівання для акцій фірми:

$M_A = 4 \cdot 0{,}25 + 9 \cdot 0{,}5 + 11 \cdot 0{,}25 = 1 + 4{,}5 + 2{,}75 = 8{,}25,$ дохідність 8,25%.

$M_B = 4 \cdot 0{,}25 + 8 \cdot 0{,}25 + 11 \cdot 0{,}5 = 1 = 8{,}5,$ дохідність 8,5%.

$M_C = 5 \cdot 0{,}2 + 15 \cdot 0{,}6 + 20 \cdot 0{,}2 = 14,$ дохідність акцій 14%.

2. Обчислимо середньоквадратичне відхилення для акцій кожної фірми. Для цього обчислимо дисперсію.

$D_A = (4 - 8{,}25)^2 \cdot 0{,}25 + (9 - 8{,}25)^2 \cdot 0{,}5 + (11 - 8{,}25)^2 \cdot 0{,}25 = 6{,}6875,$

$\sigma_A = \sqrt{6{,}6875} \approx 2{,}5860.$

$D_B = (4 - 8{,}5)^2 \cdot 0{,}25 + (8 - 8{,}5)^2 \cdot 0{,}25 + (11 - 8{,}5)^2 \cdot 0{,}5 = 8{,}25,$

$\sigma_B = \sqrt{8{,}25} \approx 2{,}8723.$

$D_C = (5 - 14)^2 \cdot 0{,}2 + (15 - 14)^2 \cdot 0{,}6 + (20 - 14)^2 \cdot 0{,}2 = 24,$

$\sigma_C = \sqrt{24} \approx 4{,}8989.$

3. По акціям фірми C дохідність складає 14%, однак ризик 4,8989. Доцільніше придбати акції фірми A, оскільки ризик мінімальний, або фірми B, оскільки дохідність вища, ніж в фірмі A, але ризик не на багато вищій.

15.1.5.2. Задача про секретарку

Постановка задачі

Директор збирається прийняти на роботу секретарку. Попередній досвід ділить секретарок на три категорії: відмінних (3 бали), хороших (2 бали) і посередніх (1 бал). Аналіз навчальних закладів із підготовки секретарок дає статистику випускниць закладів: імовірність узяти на роботу чудову секретарку — 0,2, хорошу — 0,5, посередню — 0,3. Директор може випробувати тільки трьох претенденток, причому в разі відмови директора кандидат убуває на іншу роботу.

Розв'язання

Побудуємо дерево рішень (рис. 15.1).

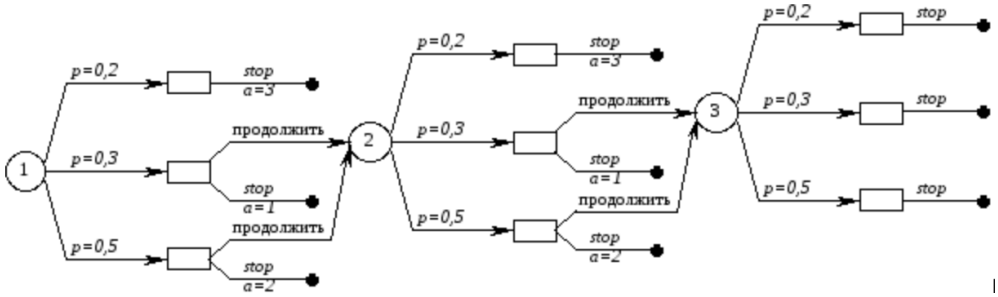


Рис. 15.1. Дерево рішень для «Задачі про секретарку»

Почнемо шукати оптимальне рішення з останнього кроку. Визначимо МО «виграшу» секретарки, якщо ми випробовуємо трьох кандидаток:

$$\overline{\overline{a_{\text{п}}}} = 3 \cdot 0{,}2 + 2 \cdot 0{,}5 + 1 \cdot 0{,}3 = 1{,}9;$$

$$\overline{\overline{a_{\text{х}}}} = 3 \cdot 0{,}2 + 2 \cdot 0{,}5 + 1{,}9 \cdot 0{,}3 = 2{,}17;$$

У другому випробуванні, якщо трапилася хороша секретарка, треба зупинитися, а в першому випробуванні, треба зупинитися тільки якщо трапилася чудова, а в третьому випробуванні беремо будь-яку. Знайдемо середній оптимальний виграш після всіх випробувань:

$$\overline{\overline{a_{\text{в}}}} = 3 \cdot 0{,}2 + 2{,}17 \cdot 0{,}5 + 2{,}17 \cdot 0{,}3 = 2{,}336.$$

Слабким місцем підході з відомими апріорними ймовірностями є те, що треба знати апріорну ймовірність. Якщо вони невідомі, то необхідно їх вивчити. Це можна робити шляхом експериментів, які враховують умови природи.

15.1.6. Приклад реалізації в Python

```
import numpy as np

# Матриця виграшів
payoff = np.array([
    [100, 50],
    [60, 90],
    [70, 70]
])

# Ймовірності станів природи
probabilities = np.array([0.4, 0.6])

# Обчислення очікуваної вигоди для кожної стратегії
expected_values = payoff @ probabilities

# Вибір найкращої стратегії
best_strategy_index = np.argmax(expected_values)

print("Очікувані вигоди:", expected_values)
```

```
print("Найкраща стратегія: A", best_strategy_index + 1)
```

15.2. Прийняття рішень в умовах невизначеності

15.2.1. Особливості прийняття рішень в умовах невизначеності

Процес *прийняття рішень в умовах неопределенності* відрізняється від детермінованого процесу, оскільки у таких умовах немає достатньої інформації для передбачення результатів з достатньою впевненістю.

Підходи та методи, які можна використовувати для прийняття рішень в умовах неопределенності

Метод сценаріїв. Розглядаються різні можливі сценарії подій та їх ймовірність, і для кожного сценарію розробляються стратегії та рішення.

Ймовірнісний підхід. Використовується теорія ймовірностей для оцінки ймовірності виникнення різних подій та їх наслідків. На основі цієї інформації розробляються стратегії з максимізацією очікуваної користі та мінімізацією очікуваних втрат.

Метод аналізу варіантів. Розглядаються різні альтернативи та їх можливі наслідки, а потім вибирається та аналізується найбільш перспективна альтернатива.

Метод інтуїтивного прийняття рішень. Ґрунтується на інтуїції, досвіді та здоровому глузді. Люди приймають рішення на основі своєї інтуїції та досвіду, коли вони мають обмежену інформацію.

Робочі моделі. Використовуються абстрактні моделі або симуляції для аналізу різних варіантів та їх можливих наслідків.

Ці методи можна використовувати окремо або в поєднанні залежно від конкретної ситуації та умов прийняття рішень. Важливою є гнучкість та здатність адаптуватися до змін в умовах неопределенності.

15.2.2. Оцінка складних систем в умовах нестохастичної невизначеності

Особливості оцінки складних систем в умовах невизначеності

1. Наявність у керуючій системі як елементу ОПР (особи що приймає рішення), що здійснює управління на основі суб'єктивних моделей, які призводять до великого розмаїття поведінки системи.
2. Алгоритм управління будує сама система управління, переслідуючи крім цілей старшої системи свої цілі, які завжди збігаються із зовнішніми.
3. На цьому етапі оцінки ситуації у ряді випадків виходять не з фактичної ситуації, а з тієї моделі, яку використовує ОПР.
4. У процесі прийняття рішень велику роль грають логічні міркування ОПР, які не піддаються формалізації класичними методами математики.
5. При виборі керуючого впливу ОПР може оперувати нечіткими поняттями, відношеннями та висловлюваннями.
6. У більшості класів завдань управління АСУ відсутні об'єктивні критерії оцінювання досягнення цільового та поточного стану ОУ, а також

статистичних даних для визначення ймовірнісних законів для конкретного прийнятого рішення.

Таким чином, методи прийняття рішень, які використовуються для детермінованих та ймовірнісних рішень, для даного класу завдань не застосовні.

Тому для оцінки систем в умовах нестохастичної невизначеності використовуються методи, в основі яких лежить *матриця ефективності* (табл. 15.2):

Таблиця 15.2. Матриця ефективності

a_i	n_j				$K(a_i)$
	n_1	n_2	…	n_l	
a_1	k_{11}	k_{21}	…	k_{l1}	
a_2	k_{12}	k_{22}	…	k_{l2}	
…	…	…	…	…	
a_m	k_{1m}	k_{2m}	…	k_{lm}	

де:

a_i — вектор керованих параметрів, що визначає властивості системи;

n_j — вектор некерованих параметрів, що визначає стан ситуації;

k_{ij} — значення ефективності системи a_i для стану ситуації n_j;

$K(a_i)$ — ефективність системи.

Залежно від характеру невизначеності операції поділяються на ігрові та статистичні.

У *ігрових операціях* невизначеність вносить своїми свідомими діями супротивник (теорія ігор).

Умови *статистично невизначених операцій* залежать від об'єктивної дійсності (природи). Природа пасивно стосовно особі, яка приймає рішення – теорія статичних рішень.

15.2.3. Ухвалення рішення в умовах невизначеності

Насамперед відзначимо принципову різницю між стохастичними чинниками, які призводять до ухвалення рішення за умов ризику, і невизначеними чинниками, які призводять до ухвалення рішення за умов невизначеності. І ті, й інші призводять до розкиду можливих результатів управління. Але стохастичні чинники повністю описуються відомою стохастичною інформацією, ця інформація дозволяє вибрати найкраще в середньому рішення. Щодо невизначених факторів, подібна інформація відсутня.

У випадку невизначеність може бути викликана або протидією розумного противника, або недостатньою поінформованістю про умови, у яких здійснюється вибір рішення.

Ухвалення рішень в умовах розумної протидії є об'єктом дослідження теорії ігор.

Розглянемо принципи вибору рішень за наявності недостатньої обізнаності щодо умов, у яких здійснюється вибір. Такі ситуації заведено називати «іграми з природою». У термінах «ігор із природою» завдання прийняття рішень може бути сформульована в такий спосіб. Нехай особа, яка приймає рішення, може вибрати один із m можливих варіантів своїх рішень: $a_1, a_2, \dots, a_m$ та нехай відносно умов, в яких будуть реалізовані можливі варіанти, можна зробити n припущень: $n_1, n_2, \dots, n_l$. Оцінки кожного варіанту розв'язку в умовах (a_i, n_j) відомі та задані у вигляді матриці виграшів ОПР: $A = |k_{ij}|$.

Припустимо спочатку, що апріорна інформація про ймовірності виникнення тієї чи іншої ситуації n_j відсутня.

Теорія статистичних рішень пропонує кілька критеріїв оптимальності вибору рішень. Вибір того чи іншого критерію не можна формалізувати, він здійснюється людиною, яка приймає рішення, суб'єктивно, виходячи з його досвіду, інтуїції і т. д. Розглянемо ці критерії.

15.2.4. Класичні критерії прийняття рішень

Критерій середнього виграшу

Даний критерій передбачає завдання ймовірності стану ситуації P_j. Ефективність системи оцінюється як середнє очікуване значення (МС) оцінок ефективності по всіх станах обстановки. Він обчислюється за формулою (15.15):

$$K(a_i) = \sum_{j=1}^{l} (P_j \cdot k_{ij,}) \; i = \overline{1,m}, j = \overline{1,l}. \qquad (15.15)$$

Оптимальній системі буде відповідати ефективність, яка обчислюється за формулою (15.16):

$$K^{\text{опт}} = \max_{1 \le i \le m} \left(\sum_{j=1}^{l} (P_j \cdot k_{ij,}) \right) i = \overline{1,m}, j = \overline{1,l}. \qquad (15.16)$$

Критерій мінімакса

Цим критерієм слід оцінювати системи по максимальному значенню ефективності і вибирати як оптимальне рішення найменше значення з максимумів. Він обчислюється за формулою (15.17):

$$K(a_i) = \max_{1 \le i \le m} (k_{ij}), i = \overline{1,m}, j = \overline{1,l},$$

$$K^{\text{опт}} = \min_{1 \le i \le m} \left(\max_{1 \le j \le l} (k_{ij,}) \right) i = \overline{1,m}, j = \overline{1,l}. \qquad (15.17)$$

Критерій максимакса

Цим критерієм слід оцінювати системи по максимальному значенню ефективності і вибирати як оптимальне рішення найбільше значення з максимумів. Обчислюється за формулою (15.18):

$$K(a_i) = \max_{1 \le i \le m} (k_{ij}), i = \overline{1,m}, j = \overline{1,l},$$

$$K^{\text{опт}} = \max_{1 \le i \le m} \left(\max_{1 \le j \le l} (k_{ij}) \right), i = \overline{1,m}, j = \overline{1,l}. \qquad (15.18)$$

Це найоптимістичніше рішення. При цьому ризик максимальний.

Критерій Лапласа

Оскільки кожна ймовірність виникнення тієї або іншої ситуації n_j невідома, їх всі вважатимемо рівноймовірними. Тоді для кожного рядка матриці виграшів підраховується середнє арифметичне значення оцінок. Оптимальному рішенню відповідатиме таке рішення, якому відповідає максимальне значення цього середнього арифметичного, тобто справедлива рівність (15.19):

$$La = K^{\text{опт}} = \max_{1 \le i \le m} \left(\frac{1}{l} \cdot \sum_{j=1}^{l} k_{ij} \right). \quad (15.19)$$

Критерій Вальда

У кожному рядку матриці вибираємо мінімальну оцінку. Оптимальному рішенню відповідає таке рішення, якому відповідає максимум цього мінімуму. Тобто справедлива рівність (15.20):

$$Wa = K^{\text{опт}} = \max_{1 \le i \le m} \left(\min_{1 \le j \le l} (k_{ij}) \right). \quad (15.20)$$

Цей критерій дуже обережний. Він орієнтований на найгірші умови, лише серед яких і відшукується найкращий і тепер уже гарантований результат.

Критерий Сэвиджа

В кожному стовпчику матриці знаходиться максимальна оцінка $\max_{1 \le i \le m} (k_{ij})$ і складається нова матриця, елементи якої визначаються співвідношенням (15.21):

$$r_{ij} = \max_{1 \le i \le m} (k_{ij}) - k_{ij}. \quad (15.21)$$

Величину r_{ij} називають *ризиком*, під яким розуміють різницю між максимальним виграшем, який мав би місце, якби було достовірно відомо, що наступить ситуація n_j, з виграшем при виборі рішення a_i в умовах n_j. Ця нова матриця називається *матрицею ризиків*. Далі з матриці ризиків вибирають таке рішення, при якому величина ризику набуває найменшого значення в найсприятливішій ситуації. Тобто, справедлива рівність (15.22):

$$Sa = K^{\text{опт}} = \min_{1 \le i \le m} \left(\max_{1 \le j \le l} (r_{ij}) \right) = \min_{1 \le i \le m} \left(\max_{1 \le j \le l} \left(\max_{1 \le i \le m} (k_{ij}) - k_{ij} \right) \right). \quad (15.22)$$

Суть цього критерію полягає в мінімізації ризику. Як і критерій Вальда, критерій Севіджа дуже обережний. Вони розрізняються різним розумінням гіршої ситуації: у першому випадку — це мінімальний виграш, в другому — максимальна втрата виграшу в порівнянні з тим, чого можна було б досягти в даних умовах.

15.2.5. Похідні критерії прийняття рішень

Критерій Гурвиця

Вводиться деякий коефіцієнт α, який називається «*коефіцієнтом оптимізму*», $0 \le \alpha \le 1$. В кожному рядку матриці виграшів знаходиться найбільша оцінка $\max_{1 \le j \le n} (k_{ij})$ і найменша $\min_{1 \le j \le n} (a_{ij})$. Вони множаться

відповідно на α и $(1-\alpha)$ і потім обчислюється їхня сума. Оптимальному рішенню буде відповідати таке рішення, якому відповідатиме максимум цієї суми. Тобто, справедливий вираз (15.23):

$$Ha = K^{\text{опт}} = \max_{1\le i\le m}[\alpha \cdot \max_{1\le j\le l}(k_{ij}) + (1-\alpha)\cdot \min_{1\le j\le l}(k_{ij})]. \qquad (15.23)$$

При $\alpha = 0$ критерій Гурвиця трансформується в критерій Вальда. Це *випадок крайнього «песимізму»*. При $\alpha = 1$ (випадок крайнього «оптимізму») людина, що приймає рішення, розраховує на те, що йому супроводитиме найсприятливіша ситуація. «Коефіцієнт оптимізму» α призначається суб'єктивно, виходячи з досвіду, інтуїції і так далі. Чим небезпечніша ситуація, тим більше обережним має бути підхід до вибору рішення і тим менше значення коефіцієнту α.

Критерій Ходжа-Лемана

Цей критерій спирається одночасно на ММ-критерій та критерій Баєса-Лапласа. За допомогою параметра α виражається ступінь довіри до використовуваного розподілу ймовірностей. Якщо довіра велика, то домінує критерій Баєса-Лапласа, інакше – ММ-критерій, тобто, справедливий вираз (15.24):

$$HL = K^{\text{опт}} = \max_{1\le i\le m}[\alpha \cdot \sum_{j=1}^{l}(k_{ij}\cdot q_i) + (1-\alpha)\cdot \min_{1\le j\le l} k_{ij}], 0 \le \alpha \le 1. \qquad (15.24)$$

Правило вибору, яке відповідає критерію Ходжа-Лемана формується таким чином: матриця рішень A доповнюється стовпцем, складеним з середніх зважених (з вагою $\alpha = const$) математичними очікуваннями та найменшого результату кожного рядка (15.20). Відбираються варіанти рішень у рядках якого стоїть найбільше значення цього стовпця. При $\alpha = 1$ критерій Ходжа-Лемана перетворюється на критерій Байєса-Лапласа, а за $\alpha = 0$ стає мінімаксним.

Критерій Гермейєра

Цей критерій орієнтований на величину втрат, тобто на відмінні значення всіх k_{ij}. При цьому справджується вираз (15.25):

$$Gr = K^{\text{опт}} = \max_{1\le i\le m}\left(\min_{1\le j\le l}(k_{ij}\cdot q_i)\right). \qquad (15.25)$$

Правило вибору згідно критерію Гермейєра формулюється наступним чином:

- Матриця рішень A доповнюється стовпцем, який містить в кожному рядку найменший з добутків його результатів на відповідний вектор стану;
- Обираються ті варіанти в рядках яких знаходиться найбільше значення цього стовпцю.

15.2.6. Приклади розв'язання задач

Постановка задача

Нехай задана наступна платіжна матриця гри з природою (табл. 15.3)

1. Знайти найкращу стратегію по критерію Гурвіца (коефіцієнт песимізму – 0,6);

2. Знайти найкращу стратегію по критерію Лапласа для даної платіжної матриці гри з природою (табл. 15.3);
3. Знайти найкращу стратегію по критерію Вальда для наступної платіжної матриці гри з природою (табл. 15.3);
4. Знайти найкращу стратегію по критерію Севіджа для наступної платіжної матриці гри з природою (табл. 15.3);
5. Знайти найкращу стратегію по критерію мінімакса (максимакса) для наступної платіжної матриці гри з природою (табл. 15.3);
6. Порівняти отримані результати і зробити висновок щодо найкращої стратегії.

Таблиця 15.3. Платіжна матриця

	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$
A_1	1	7	3	9	15
A_2	2	4	18	2	6
A_3	4	3	1	8	5
A_4	12	1	3	18	19
A_5	1	3	4	2	1

Розв'язання

1. Критерій Гурвиця (15.23), $\alpha = 0{,}6$. Побудуємо матрицю виграшу A (табл. 15.4).

Таблиця 15.4. Матриця виграшу

	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$	$min(k_{ij})$	$max(k_{ij})$	$\alpha \cdot \max\limits_{1\le j\le l}(k_{ij}) + (1-\alpha) \cdot \min\limits_{1\le j\le l}(k_{ij})$	Ha
A_1	1	7	3	9	15	1	15	6,6	
A_2	2	4	18	2	6	2	18	8,4	8,4
A_3	4	3	1	8	5	1	8	3,8	
A_4	12	1	3	18	19	1	19	8,2	
A_5	1	3	4	2	1	1	4	2,2	

Для матриці виграшів критерій Гурвиця $Ha = 8{,}4$ і відповідно, найкраща стратегія A_2.

2. Критерій Лапласа (15.19). Побудуємо матрицю виграшу A (табл. 15.5).

Таблиця 15.5. Матриця виграшу

	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$	$\frac{1}{l} \cdot \sum\limits_{j=1}^{l} k_{ij}.$	La
A_1	1	7	3	9	15	7	
A_2	2	4	18	2	6	6,4	
A_3	4	3	1	8	5	4,2	
A_4	12	1	3	18	19	10,6	10,6

A_5	1	3	4	2	1	2,2	

Для матриці виграшів критерій Лапласа $La = 10{,}6$ і відповідно, найкраща стратегія A_4.

3. Критерій Вальда (15.20). Побудуємо матрицю виграшу A (табл. 15.6).

Таблиця 15.6. Матриця виграшу

	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$	$min(k_{ij})$	Wa
A_1	1	7	3	9	15	1	
A_2	2	4	18	2	6	2	2
A_3	4	3	1	8	5	1	
A_4	12	1	3	18	19	1	
A_5	1	3	4	2	1	1	

Для матриці виграшів критерій Вальда $Wa = 2$ і відповідно, найкраща стратегія A_2.

4. Критерій Севіджа (15.21). Побудуємо матрицю ризику для матриці виграшу A (табл. 15.7).

Таблиця 15.7. Матриця ризику

$max(k_{ij})$	r_{ij}					$\max\limits_{1\le j\le l}(r_{ij})$	Sa
15	14	8	12	6	0	14	
18	16	14	0	16	12	16	
8	4	5	7	0	3	7	
19	7	18	16	1	0	18	
4	3	1	0	2	3	3	3

Для матриці ризику критерій Севіджа $Sa = 3$ і відповідно, найкраща стратегія A_5.

5. Критерій мінімакса (15.17), максимакса (15.18). Побудуємо матрицю виграшу A (табл. 15.8).

Таблиця 15.8. Матриця виграшів

	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$	$max(k_{ij})$
A_1	1	7	3	9	15	15
A_2	2	4	18	2	6	18
A_3	4	3	1	8	5	8
A_4	12	1	3	18	19	19
A_5	1	3	4	2	1	4

Згідно критерію мінімакса, $min(max(k_{ij})) = 4$, отже, найкраща стратегія A_5, а згідно критерію максимакса, $max(max(k_{ij})) = 19$, отже, A_4.

6. Порівняння отриманих результатів наведено в табл. 15.9.

Таблиця 15.9. Порівняння найкращих стратегій згідно кожного критерію

Критерій	Найкраща стратегія
Гурвіца	A_2
Лапласа	A_4

Вальда	A_2
Севіджа	A_5
Мінімакма	A_5
Максиміна	A_4

Серед рекомендованих найкращих стратегій варто розглянути: A_2, A_4, A_5. Для більш точної відповіді варто скористатися додатковими критеріями, оскільки дані стратегії були найкращими за відповідним критерієм однакову кількість разів.

15.2.7. Приклад обчислення значень критеріїв в Python

Припустимо, що у нас є матриця виграшів, де кожен рядок відповідає альтернативі (рішенню), а кожен стовпець – стану природи.

Введення матриці виграшів

```
import numpy as np

# Матриця виграшів (альтернативи x стани природи)
payoff_matrix = np.array([
    [5, 8, 3], # №1
    [7, 2, 6], # №2
    [4, 9, 5] # №3
])
```

Критерій Вальда

```
wald_values = np.min(payoff_matrix, axis=1)  # Мінімум по кожному рядку
wald_criterion = np.max(wald_values)  # Максимальний серед мінімальних значень
best_alternative_wald = np.argmax(wald_values)  # Альтернатива за Вальдом

print(f"Критерій Вальда: {wald_criterion}, Альтернатива: {best_alternative_wald + 1}")
```

Критерій Мінімакса

```
minimax_values = np.max(payoff_matrix, axis=1)  # Максимум по кожному рядку
minimax_criterion = np.min(minimax_values)  # Мінімальний серед максимальних значень
best_alternative_minimax = np.argmin(minimax_values)
print(f"Критерій Мінімакса: {minimax_criterion}, Альтернатива: {best_alternative_minimax + 1}")
```

Критерій Лапласа

```
laplace_values = np.mean(payoff_matrix, axis=1)    # Середнє арифметичне по рядках
laplace_criterion = np.max(laplace_values)
best_alternative_laplace = np.argmax(laplace_values)

print(f"Критерій Лапласа: {laplace_criterion}, Альтернатива: {best_alternative_laplace + 1}")
```

Критерій Севіджа

```
# Обчислення матриці ризиків
max_values = np.max(payoff_matrix, axis=0)      # Максимальне по кожному стовпцю
risk_matrix = max_values - payoff_matrix  # Обчислення ризиків

# Критерій Севіджа
savage_values = np.max(risk_matrix, axis=1)      # Максимальний ризик для кожної альтернативи
savage_criterion = np.min(savage_values)        # Мінімізація максимального ризику
best_alternative_savage = np.argmin(savage_values)

print(f"Критерій Севіджа: {savage_criterion}, Альтернатива: {best_alternative_savage + 1}")
```

Критерій Гермейєра

```
lambda_ = 0.5  # Вага критерію Вальда

hermeyer_values = lambda_ * np.min(payoff_matrix, axis=1) + (1 - lambda_) * np.max(payoff_matrix, axis=1)
hermeyer_criterion = np.max(hermeyer_values)
best_alternative_hermeyer = np.argmax(hermeyer_values)

print(f"Критерій Гермейєра: {hermeyer_criterion}, Альтернатива: {best_alternative_hermeyer + 1}")
```

Аналіз отриманих результатів

Результати наведено в табл. 15.10.

Таблиця 15.10. Порівняння найкращих стратегій згідно кожного критерію

Критерій	Найкраще рішення
Вальда	№ 3
Мінімакса	№ 1
Лапласа	№ 2
Севіджа	№ 3
Ходжа-Лемана	№ 2
Гермейєра	№ 2

З табл. 15.10 видно, що найчастіше критерії рекомендовали альтернативу №2. Її і можна вважати найбільш підходящою.

15.3. Формування множини Парето

15.3.1. Основні поняття

Множина Парето, також відома як Парето-фронт, є поняттям у теорії оптимізації та економіці, яке використовується для визначення набору оптимальних рішень у задачах багатокритеріальної оптимізації. Множина Парето названа так на честь італійського економіста Вільфредо Парето.

Множиною Парето називається підмножина ефективних рішень повної множини альтернатив, яка відповідає наступній умові: об’єкт належить до підмножини ефективних рішень тільки в тому випадку, якщо хоча б по одному з оцінюваних ознак він кращий за всі інші.

Множина Еджворта-Парето є фундаментом численних теоретичних досліджень, а також надійним інструментом при розв’язанні різних прикладних багатокритеріальних задач.

Множина Еджворта-Парето названа так іменами вчених, які вперше звернули увагу на альтернативи, що поступаються одна одній при оцінці за кожним критерієм.

Ідея формування множини Еджворта-Парето у «відсіюванні» альтернатив, у яких оцінки за всіма критеріями гірші, ніж у інших. Реалізація цієї ідеї потребує введення визначення відношення домінування на множині альтернатив.

Альтернативу A будемо називати *домінуючою* відносно альтернативи B, якщо значення оцінки альтернативи A за всіма критеріями не гірше, ніж значення оцінки альтернативи B, а хоча б за одним критерієм значення оцінки альтернативи A – краще.

За таких умов альтернатива В називається *домінованою*.

Альтернативи A і B будемо називати *непорівнюваними* відносно одна одної, якщо значення оцінки альтернативи A за одним із критеріїв краще за значення цього ж критерію альтернативи B, але водночас за одним з інших критеріїв значення альтернативи A гірше за значення цього критерію альтернативи B.

Абсолютно домінуюча альтернатива – це альтернатива, яка є домінуючою відносно кожної з наданих альтернатив.

Отже, процедура формування множини Еджворта–Парето полягає у пошуку всіх домінованих альтернатив та виключення їх із процесу прийняття рішення.

Для множини Еджворта-Парето справедливі такі твердження:

- до множини Еджворта-Парето входять альтернативи, які хоча б за одним критерієм не гірші, ніж інші;
- множина Еджворта-Парето складається із непорівнювальних альтернатив або тільки однієї абсолютно домінуючої альтернативи;
- множину Еджворта-Парето називають множиною непокращуваних рішень.

Задача виділення множини Еджворта-Парето зазвичай розглядається як попередня, і якщо не виявлено абсолютно домінуючої альтернативи, то для прийняття рішень використовують інші методи вже на суттєво меншій множині альтернатив.

У задачах багатокритеріальної оптимізації існує кілька цільових функцій, які потрібно максимізувати або мінімізувати одночасно.

Розв'язок називається *Парето-оптимальним* (або не домінованим), якщо не існує іншого розв'язку, який би покращував значення однієї цільової функції без погіршення значень інших. Множина Парето складається з усіх Парето-оптимальних розв'язків задачі.

15.3.2. Формальне визначення

Нехай $f_1(x), f_2(x), \dots, f_k(x)$ — це цільові функції, які потрібно оптимізувати, де x є вектором змінних. Розв'язок x^* називається *Парето-оптимальним*, якщо для будь-якого іншого розв'язку x виконується хоча б одна з умов (15.26), (15.27):

$$\forall i: f_i(x) \geq f_i(x^*), \quad (15.26)$$

$$\exists j: f_j(x) > f_j(x^*). \quad (15.27)$$

Множина всіх Парето-оптимальних розв'язків називається *множиною Парето* або *Парето-фронтом*.

Інтерпритація: Нехай є дві цільові функції $f_1(x)$ і $f_2(x)$, які потрібно мінімізувати:

- Цільова функція 1: $f_1(x) = x_1 + 2 \cdot x_2$;
- Цільова функція 2: $f_2(x) = 2 \cdot x_1 + x_2$.

Потрібно знайти всі (x_1, x_2), які мінімізують обидві функції. Після обчислення можна побудувати множину Парето, яка складатиметься з точок, де жодна з цільових функцій не може бути покращена без погіршення іншої.

15.3.3. Візуалізація

Візуалізація множини Парето (*Pareto front*) використовується для аналізу результатів багатокритеріальної оптимізації. Вона показує, як розподіляються оптимальні рішення у просторі критеріїв.

Для двовимірної або тривимірної задачі множину Парето можна візуалізувати на графіку (рис. 15.2). Наприклад, якщо є дві цільові функції

f_1 і f_2, то множина Парето складається з точок на площині, де жодна точка не домінує іншу.

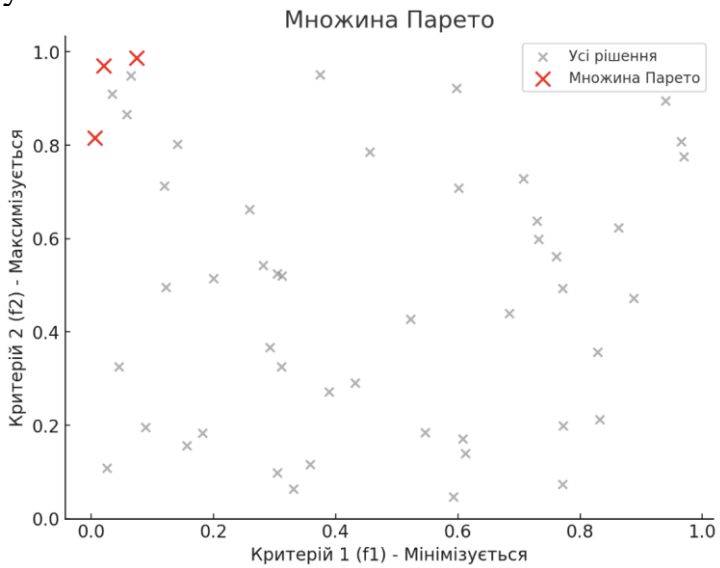


Рис. 15.2. Приклад множини Парето

15.3.4. Використання множини Парето

Множина Парето широко використовується в багатокритеріальній оптимізації, коли потрібно знайти компроміс між кількома дослідними критеріями .

Області застосування множини Парето

1. *Економіка та бізнес*:
- Оптимізація витрат та якість продукту;
- Баланс між прибутком і ризиком в інвестиціях;
- Вибір стратегії маркетингу з урахуванням бюджету та ефективності.

2. *Інженерія та виробництво*:
- Проектування автомобілів: баланс ваги, міцність та витрати пального;
- Оптимізація технологічних процесів (час обробки vs витрати енергії);
- Вибір матеріалів за міцністю і собівартістю.

3. *Штучний інтелект та машинне навчання*:
- Налаштування нейромереж: баланс точності та швидкості вимірювання;
- Підбір параметрів моделей, щоб мінімізувати помилку, але зберегти інтерпретованість.

4. *Логістика та транспорт*:
- Оптимізація маршрутів: мінімізація витрат пального та часу доставки;
- Планування виробництва та складів (баланс між запасами та швидкістю постачання).

5. *Медицина та фармакологія*:
- Вибір оптимального лікування (ефективність vs. побічні ефекти);
- Проектування медичних пристроїв (компактність vs. функціональність).

6. Екологія та енергетика:

- Баланс між екологічною безпекою та економічною вигодою;
- Вибір відновлюваних джерел енергії для зменшення викидів CO_2.

15.3.5. Приклад побудови множини Парето

Постановка задачі

Система оцінюється за трьома критеріями – швидкодія (k_1), вартість (k_2), ефективність (k_3), значення яких лежать в інтервалах відповідно $[100; 200], [10; 50], [0{,}1; 1]$. Значення цих критеріїв для кожної з 10 можливих систем наведені в табл. 15.11. З усіх поданих варіантів систем визначити ту, що є оптимальною.

Таблиця 15.11. Множина об'єктів

№ пор.	k_1	k_2	k_3
1	100	20	0,1
2	200	50	0,5
3	120	20	0,6
4	180	50	0,7
5	100	30	0,6
6	150	40	0,4
7	200	30	0,7
8	140	40	0,8
9	160	40	0,7
10	130	30	0,8

Розв'язання

За визначенням множини Парето, виберемо тільки ті об'єкти, які «не гірші» за інших. Для виключення свідомо неоптимальних рішень здійснюємо попарне порівняння кожного об'єкта з усіма іншими, і, якщо знайдеться об'єкт, який за всіма трьома критеріями виявиться кращим за розглянутий, то цей об'єкт видаляється з розгляду. Наприклад, порівняємо 1-й об'єкт з 2-м: за швидкодією (k_1) і ефективністю (k_3) другий об'єкт кращий за перший; але за вартістю (k_2) першого дешевший, а значить, кращий за другий, отже, на даному етапі перший об'єкт не виключається з області ефективних рішень. Далі, порівняємо 1-й і 3-й об'єкти: за швидкодією та ефективністю 3-й об'єкт кращий за другий, а за ціною вони ідентичні, отже, 3-й об'єкт кращий за перший. На другому кроці 1-й об'єкт виключається з розгляду.

У результаті перебору кожного з кожним варіантом отримаємо таблицю множини Парето чи множину ефективних рішень (табл. 15.12).

Таблиця 15.12. Ефективна множина рішень

№ пор.	k_1	k_2	k_3
3	120	20	0,6
7	200	30	0,7

8	140	40	0,8
10	130	30	0,8

Для подальшого уточнення найкращого варіанта скористаємося функцією корисності для кожного об'єкта, який розраховується за формулою (15.28):

$$F_i = \sum_{j=1}^{n} \left(\frac{k_i \cdot \alpha_i}{k_{i\,max}} \right), \qquad (15.28)$$

де:

k_i – оцінка i-го об'єкта за j-м критерієм ($i = \overline{1,m}$, $j = \overline{1,n}$);

$k_{i\,max}$ – максимально можлива оцінка за даним критерієм;

α_i – ваговий коефіцієнт j-м критерію за i-м об'єктом.

Нехай значення вагових коефіцієнтів критеріїв дорівнюють $\alpha_1 = 0{,}5$; $\alpha_2 = 0{,}3$; $\alpha_3 = 0{,}2$. Для правильного оцінювання варіантів рішень, поданих у табл. 15.12, необхідно помножити ваговий коефіцієнт на відповідний критерій. Результати подані в табл. 15.13.

Таблиця 15.13. Ефективна множина рішень з урахуванням значень вагових коефіцієнтів

№ пор.	k_1	k_2	k_3
3	60	6	0,12
7	100	9	0,14
8	70	12	0,16
10	65	9	0,16

Після цих дій діапазон значень критеріїв k_1, k_2 і k_3 лежить в діапазоні $[60; 100]$, $[6; 12]$, $[0{,}12; 0{,}16]$ відповідно.

Далі, отримавши ефективну множину рішень, ОПР, підсумувавши добуток значень критерію на відповідний ваговий коефіцієнт, набуває значення функції корисності об'єкта. Максимальне значення функції корисності відповідає оптимальному розв'язанню.

Однак, множина Парето, подана в табл. 15.13, не може бути безпосередньо оброблена, тому що містить ненормовані значення критеріїв, що, незалежно від переваг ОПР, надає певної ваги критеріям оптимальності. Нормовані значення критеріїв подані в табл. 15.14. Нормування проводимо поділом поточного значення цього критерію на максимально можливе для нього значення (для $k_1 = 100$, для $k_2 = 12$, для $k_3 = 0{,}16$).

Таблиця 15.14. Нормовані значення ефективної множини рішень

№ пор.	k_1	k_2	k_3
3	0,6	0,5	0,75
7	1	0,75	0,875
8	0,7	1	1
10	0,65	0,75	1

Оскільки множення на вагові коефіцієнти і нормування вже здійснено, залишається тільки підсумувати значення коефіцієнтів по

кожному об’єкту з табл. 15.14. Причому, необхідно врахувати різноспрямованість критеріїв. У першого та третього критерію чим вище значення, тим краще для ОПР, а ось другий критерій має протилежне спрямування, тому його необхідно віднімати від функції корисності:

$$F_3 = 0{,}6 - 0{,}5 + 0{,}75 = 1{,}85;$$

$$F_7 = 2{,}625;\ F_8 = 2{,}7;\ F_{10} = 2{,}4.$$

Таким чином, з усіх поданих варіантів систем оптимальною є восьма, тому що максимальне значення функції корисності відповідає восьмому об’єкту.

15.3.6. Приклад реалізації множини Парето в Python

```
import numpy as np
import matplotlib.pyplot as plt

# Генеруємо випадкові рішення (дві цільові функції)
np.random.seed(42)
n_points = 100
solutions = np.random.rand(n_points, 2)  # Критерії f1
(мінімізуємо), f2 (максимізуємо)

# Функція для пошуку Парето-оптимальних точок
def pareto_front(points):
    pareto = []
    for i, p in enumerate(points):
        dominated = False
        for j, q in enumerate(points):
            if (q[0] <= p[0] and q[1] >= p[1]) and
(q[0] < p[0] or q[1] > p[1]):
                dominated = True
                break
        if not dominated:
            pareto.append(p)
    return np.array(pareto)

# Знаходимо Парето-оптимальні точки
pareto_optimal = pareto_front(solutions)

# Візуалізація
plt.figure(figsize=(8, 6))
plt.scatter(solutions[:,    0],    solutions[:,    1],
color="gray", alpha=0.6, label="Усі рішення")
```

```
plt.scatter(pareto_optimal[:, 0], pareto_optimal[:, 1], color="red", label="Множина Парето", edgecolors='black', s=100)
plt.xlabel("Критерій 1 (f1) - Мінімізується")
plt.ylabel("Критерій 2 (f2) - Максимізується")
plt.title("Множина Парето")
plt.legend()
plt.grid()
plt.show()
```

15.4. Антагоністичні ігри

15.4.1. Визначення антагоністичних ігор

Антагоністичні ігри – це тип ігор, де інтереси учасників протистоять один одному. У таких іграх перемога одного гравця часто означає поразку для інших.

Основні риси антагоністичних ігор:

Протистояння інтересів. Учасники прямо конкурують між собою за досягнення своїх цілей, іноді навіть на шкоду іншим учасникам.

Стратегія і тактика. Гравці намагаються розробити стратегії та тактики, які дозволять їм досягти перемоги або переваги над іншими гравцями.

Нульова сума. Виграш одного гравця при антагоністичній грі зазвичай збігається з втратою для інших гравців, що означає, що сума виграшів і втрат гравців дорівнює нулю або іншій сталій кількості.

Протистояння. Зазвичай в антагоністичних іграх спостерігається пряме протистояння між гравцями, де один гравець може виграти тільки в тому випадку, якщо інший програє.

Деякі приклади антагоністичних ігор включають шахи, покер, футбол, боротьбу, економічні конкуренції тощо. В антагоністичних іграх важливо розуміти стратегії і мотивації інших гравців, щоб досягти успіху.

Формально, ця протилежність (антагоністичність), виявляється в тому, що при переході від однієї ситуації до іншої збільшення (зменшення) виграшу одного гравця, тягне за собою зменшення (збільшення) виграшу іншого. Таким чином, сума виграшів гравців в будь-якій ситуації в антагоністичних іграх стала (як правило, можна вважати, що вона дорівнює нулю). Тому, антагоністичні ігри називають, також, *іграми двох осіб з нульовою сумою* (іноді — *нульовими іграми*).

Математичне визначення поняття антагоністичності (рівність по величині і протилежність по знаку функцій виграшу гравців) є формальним поняттям, яке відрізняється від змістовного філософського поняття, але зберігає його основну рису — непримиренність протиріч.

Антагоністичні ігри в нормальній формі задають системою $\Gamma = < A, B, H >$, де A, B — множини стратегій першого та другого гравців відповідно, H — функція з дійсними значеннями, визначена на всій

множині ситуацій $A \times B$, яка є функцією виграшу першого гравця (за визначенням, функція виграшу другого гравця дорівнює $-H$). Процес розігрування антагоністичних ігор полягає в виборі гравцями деяких своїх стратегій $a \in A, b \in B$, після чого перший гравець отримує від другого суму $H(a,b)$.

15.4.2. Стратегії гравців в антагоністичних іграх

Розумна поведінка гравців в антагоністичних іграх відбувається на основі принципу максиміна, тобто якщо виконується (15.29).

$$\max_{a\in A}\left(\inf_{b\in B}\big(H(a,b)\big)\right) = \min_{b\in B}\left(\sup_{a\in A}\big(H(a,b)\big)\right). \qquad (15.29)$$

Тоді в кожного гравця існують оптимальні стратегії, тобто, стратегії, на яких досягаються в (15.29) зовнішні екстремуми. Однак, навіть в найпростіших випадках рівність (15.29) може не мати місця. Наприклад, в матричній грі з матрицею

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

виявляється

$$\max_i\left(\min_j(a_{ij})\right) = -1, \min_i\left(\max_j(a_{ij})\right) = 1.$$

Для того, щоб забезпечити реалізованість принципу максиміна, множини стратегій гравців розширюють до множини змішаних стратегій, які полягають в випадковому виборі гравцями своїх початкових стратегій, які називаються чистими, а функція виграшу визначається як математичне сподівання виграшу в умовах застосування змішаних стратегій. В наведеному прикладі оптимальними змішаними стратегіями гравців є вибори гравцями обох своїх стратегій з ймовірностями $\frac{1}{2}$, а значення гри дорівнює нулю.

Якщо множини A та B скінченні, то антагоністична гра називається *матричною грою*; для неї завжди існують оптимальні змішані стратегії у обох гравців. Якщо ж одна із множин A або B нескінченна, то антагоністична гра називається *нескінченною*.

Принцип максиміна для нескінченних антагоністичних ігор може здійснюватись (якщо рівність (15.25) не має місця) у вигляді рівності:

$$\sup_{a\in A}\left(\inf_{b\in B}\big(H(a,b)\big)\right) = \inf_{b\in B}\left(\sup_{a\in A}\big(H(a,b)\big)\right).$$

В такому випадку оптимальною стратегією для гравців не існує, однак для будь якого $\varepsilon > 0$ існують ε-оптимальні стратегії (тобто, стратегії, які забезпечують досягнення значення гри з заданою точністю ε) у обох гравців.

Якщо обидві множини A та B нескінченні, то оптимальні змішані стратегії (і навіть ε-оптимальні) не завжди існують. Наприклад, в грі з функцією виграшу

$$H(a,b)=\begin{cases}1, a>b;\\ 0, a=b\\ -1, a<b,\end{cases}$$

де стратегіями гравців є множини натуральних чисел.

15.4.2. Моделі антагоністичних ігор

Існує велика кількість явищ, для яких антагоністичні ігри є задовільною моделлю. До них відносяться деякі (але не всі) військові операції, спортивні і салонні ігри, прийняття ділових рішень в умовах конкуренції.

Прийняття рішень в умовах невизначеності, наприклад, ігри проти природи, можна також моделювати як антагоністичні ігри, припускаючи, що справжня, але невідома закономірність природи призводить до дій, найменш сприятливих для гравця. Це припущення не значить, однак, що природа наділена свідомістю, спрямованою проти людини.

В антагоністичних іграх, за визначенням, неможливі будь які переговори і угоди між гравцями. Дійсно, якщо в результаті будь яких переговорів або домовленостей один із гравців зумів би збільшити свій виграш на деяку величину, то виграш іншого гравця зменшився б на таку ж величину, тобто, для нього такі домовленості були б невигідними.

Природним узагальненням матричних ігор є нескінченні антагоністичні ігри (НАІ), в яких хоч би один з гравців має безконечну кількість можливих стратегій. Ми розглядатимемо ігри двох гравців, що роблять по одному ходу, і після цього відбувається розподіл виграшів.

Велике значення в теорії НАІ має вигляд функції виграшів $M(x,y)$. Так, у відмінності від матричних ігор, не для всякої функції $M(x,y)$ існує рішення. Вважатимемо, що вибір певного числа гравцем означає вживання його чистої стратегії, відповідної цьому числу. По аналогії з матричними іграми назвемо *чистою верхньою ціною гри* величину (формули (15.30) та (15.31)):

$$V_1=\max_x\left(\inf_y(M(x,y))\right) \text{ або } V_1=\max_x\left(\min_y(M(x,y))\right), \qquad (15.30)$$

а *чистою нижньою ціною гри* величину

$$V_2=\min_y\left(\sup_x(M(x,y))\right) \text{ або } V_2=\min_y\left(\max_x(M(x,y))\right). \qquad (15.31)$$

Для матричних ігор величини V_1 і V_2 завжди існують, а в безконечних іграх вони можуть не існувати.

Природно вважати, що, якщо для якої-небудь безконечної гри величини V_1 і V_2 існують і рівні між собою ($V_1=V_2=V$), то така гра має рішення в чистих стратегіях, тобто оптимальною стратегією гравця 1 є

вибір числа $x_0 \in X$ і гравця 2 – числа $y_0 \in Y$, при яких $M(x_0, y_0) = V$, в цьому випадку V називається ціною гри, а (x_0, y_0) – *сідловою точкою в чистих стратегіях*. Тобто, сідлова точка – це такий елемент в платіжній матриці гри, який є мінімальним у своєму рядку і одночасно максимальним у своєму стовпці.

Чисті стратегії – це пара стратегій (одна - для першого гравця, а друга – для другого гравця), які перехрещуються в сідловій точці. Сідлова точка в цьому випадку і визначає ціну гри.

Ігри, які не мають сідлової точки, на практиці зустрічаються частіше. Доведено, що і у цьому випадку рішення завжди є, але воно знаходиться в межах змішаних стратегій. Знайти рішення гри без сідлової точки означає визначення такої стратегії, яка передбачає використання кількох чистих стратегій.

В іграх із сідловою точкою відхилення одного гравця від своєї оптимальної стратегії зменшує його виграш (в найкращому випадку виграш залишається незмінним).

В іграх, які не мають сідлової точки, ситуація інша. Відходячи від своєї оптимальної стратегії гравець має можливість отримати виграш більший за нижню ціну гри. Але така спроба пов'язана з ризиком: якщо другий гравець вгадає, яку стратегію застосував перший, тоді він також відступить від своєї мінімаксної стратегії. В результаті виграш першого гравця буде меншим за нижню ціну гри. Єдина можливість завадити противнику вгадати, яка стратегія використовується – це застосувати декілька чистих стратегій. Звідси з'являється поняття «змішана стратегія».

15.4.3. Приклади дослідження антагоністичних ігор

Постановка задачі

Гравець 1 обирає $x \in X = [0; 1]$, гравець 2 обирає $y \in Y = [0; 1]$. Після цього гравець 1 отримує суму

$$M(x, y) = 2 \cdot x^2 - y^2$$

за рахунок гравця 2.

1. Дослідити дану задачу на наявність сідловок точки;
2. Обчислити ціну гри.

Розв'язання

Як сказано в умові, гравець 1 обирає число x з множини $X = [0; 1]$, гравець 2 обирає число y з множини $Y = [0; 1]$. Після цього гравець 2 платить гравцю 1 суму

$$M(x, y) = 2 \cdot x^2 - y^2.$$

Оскільки гравець 2 хоче мінімізувати виграш гравця 1, то він визначає

$$\min_{y \in Y}(2 \cdot x^2 - y^2) = 2 \cdot x^2 - 1,$$

тобто при цьому $y = 1$.

Гравець 1 бажає максимізувати свій виграш, і тому визначає

$$\max_{x\in X}(\min_{y\in Y}(2\cdot x^2-y^2))=\max_{x\in X}(2\cdot x^2-1)=2-1=1,$$

який досягається при $x=1$.

Отже, нижня ціна гри дорівнює $V_1=1$.

Верхня ціна гри

$$V_2=\min_{y\in Y}(\min_{x\in X}(2\cdot x^2-y^2))=\min_{y\in Y}(2-y^2)=2-1=1,$$

тобто, в цій грі $V_1=V_2=1$. Отже, маємо задачу з сідловою точкою. Тобто, ціна гри $V=1$, а $A(1;1)$ – сідлова точка.

15.4.4. Антагоністичні ігри реалізація в Python

```
import numpy as np
from scipy.optimize import linprog

# Матриця виграшів для гравця A (рядкового)
A = np.array([
    [3, -2, 4],
    [1, 0, -1],
    [2, -3, 5]
])

# Знаходимо мінімаксну стратегію для рядкового гравця
A
m, n = A.shape  # Кількість стратегій у A та B

# Лінійне програмування для гравця A
c = np.ones(m)  # Максимізуємо v
A_ub = -A.T  # Мінімізуємо найгірший виграш
b_ub = -np.ones(n)  # Умови обмежень
bounds = [(0, None) for _ in range(m)]  # Ймовірності
не можуть бути від'ємними

res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=bounds,
method="highs")

# Оптимальні змішані стратегії
p = res.x / np.sum(res.x)

# Аналогічно для гравця B
c = -np.ones(n)
A_ub = A
b_ub = np.ones(m)
bounds = [(0, None) for _ in range(n)]
```

```
res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=bounds,
method="highs")

# Оптимальні змішані стратегії
q = res.x / np.sum(res.x)

# Відображення результатів
print("Оптимальна стратегія гравця A (рядкового):", p)
print("Оптимальна стратегія гравця B (стовпцевого):",
q)
```

15.5. Розв'язання матричних ігор в чистих стратегіях

15.5.1. Поняття матричної гри

Матрична гра двох гравців – це математична модель, яка використовується для аналізу взаємодії двох учасників у конкурентній ситуації, де кожен гравець має деякі альтернативи вибору. Вона представляється у вигляді матриці, де кожен рядок відповідає можливим діям першого гравця, кожен стовпчик – можливим діям другого гравця, а кожний елемент матриці відображає виграш чи втрату першого гравця у випадку, коли обидва гравці вибирають відповідні дії.

Наприклад, розглянемо матричну гру «Затори». Два водії (гравці) мають вибір між двома маршрутами: коротким, але зі значними трафіковими заторами (означається як «пробки»), та довшим, але швидшим маршрутом без пробок. Матриця виглядатиме наступним чином (рис. 15.3):

```
         | Пробки   | Без пробок |
--------|----------|-------------|
Пробки   | (-10, -10) | (-30, 0)    |
--------|----------|-------------|
Без пробок| (0, -30) | (-20, -20)  |
```

Рис. 15.3. Матриця виграшів

У цій матриці кожний елемент вказує виграш (-10) або втрату (-30) першого гравця (водія) та виграш (-10) або втрату (-20) другого гравця у випадку, коли обидва гравці обирають відповідні дії.

Матричні ігри дозволяють моделювати і аналізувати різні стратегії гравців та прогнозувати їхні результати в залежності від вибору кожного гравця. Такі ігри широко використовуються в теорії ігор, економіці, політичних науках та інших галузях для дослідження конкурентних ситуацій.

15.5.2. Матрична гра двох гравців з нульовою сумою

15.5.2.1. Основні поняття

Матрична гра двох гравців з нульовою сумою може розглядатися як наступна абстрактна гра двох гравців.

Перший гравець має m стратегій $i = 1,2,\dots,m$, другий має n стратегій $j = 1,2,\dots,n$. Кожній парі стратегій (i,j) поставлено у відповідність число a_{ij}, що виражає виграш гравця 1 за рахунок гравця 2, якщо перший гравець прийме свою i-ту стратегію, а другий – свою j-ту стратегію.

Кожен з гравців робить один хід: гравець 1 обирає свою i-ту стратегію ($i = \overline{1,m}$), гравець 2 – свою j-ту стратегію ($j = \overline{1,n}$), Після чого гравець 1 отримує виграш a_{ij} за рахунок гравця 2 (якщо $a_{ij} < 0$, то це означає, що гравець 1 платить друга сума $|a_{ij}|$). На цьому гра закінчується.

Кожна стратегія гравця ($i = \overline{1,m}$; $j = \overline{1,n}$) часто називається *чистою стратегією*.

Якщо розглянути матрицю

$$A = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{1j} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix},$$

то проведення кожної партії матричної гри з матрицею A зводиться до вибору гравцем 1 i-го рядка, а гравцем 2 j-го стовпця і одержання гравцем 1 (за рахунок гравця 2) виграшу a_{ij}

Головним у дослідженні ігор є поняття оптимальних стратегій гравців. У це поняття інтуїтивно вкладається такий сенс: стратегія гравця є оптимальною, якщо застосування цієї стратегії забезпечує йому найбільший гарантований виграш при всіляких стратегіях іншого гравця. Виходячи з цих позицій, гравець 1 досліджує матрицю виграшів A таким чином: для кожного значення i $(i = \overline{1,m})$. Визначається мінімальне значення виграшу в залежності від застосовуваних стратегій гравця 2

$$\min_{1 \le j \le n} (a_{ij}), (i = \overline{1,m}),$$

тобто визначається мінімальний виграш для гравця 1 за умови, що він прийме свою i-ту чисту стратегію, потім з цих мінімальних виграшів відшукується така стратегія $i = i_0$ при якій цей мінімальний виграш буде максимальним, тобто виконується умова (15.32):

$$\max_{1 \le i \le m} \left(\min_{1 \le j \le n} (a_{ij}) \right) = a_{i_0 j_0} = \underline{\alpha}. \qquad (15.32).$$

Число α, що визначається за формулою (15.32), називається *нижньою чистою ціною гри* і показує, який мінімальний виграш може гарантувати собі гравець 1, застосовуючи свої чисті стратегії при всіляких діях гравця 2.

Гравець 2 при оптимальному своїй поведінці повинен прагне по можливості за рахунок своїх стратегій максимально зменшити виграш

гравця 1. Тому для гравця 2 відшукується $\max\limits_{1\le i\le m}(a_{ij})$, тобто визначається max виграш гравця 1, за умови, що гравець 2 застосує свою j-ту чисту стратегію, потім гравець 2 відшукує таку свою $j = j_1$ стратегію, при якій гравець 1 отримає min виграш, тобто виконується умова (15.33):

$$\min_{1\le j\le n}\left(\max_{1\le i\le m}(a_{ij})\right) = a_{i_1 j_1} = \bar{\alpha}. \qquad (15.33).$$

Число $\bar{\alpha}$, що визначається за формулою (5.33), називається *чистою верхньою ціною гри* і показує, який максимальний виграш за рахунок своїх стратегій може собі гарантувати гравець 1.

Іншими словами, застосовуючи свої чисті стратегії гравець 1 може забезпечити собі виграш не менше $\underline{\alpha}$, а гравець 2 за рахунок застосування своїх чистих стратегій може не допустити виграш гравця 1 більше, ніж $\bar{\alpha}$.

Якщо в грі з матрицею A $\underline{\alpha} = \bar{\alpha}$, То говорять, що ця гра має *сідлову точку* в чистих стратегіях і *чисту ціну* гри $\upsilon = \underline{\alpha} = \bar{\alpha}$.

Сідлова точка – це пара чистих стратегій (i_0, j_0) відповідно гравців 1 і 2, при яких досягається рівність $\underline{\alpha} = \bar{\alpha}$. У це поняття вкладений наступний сенс: якщо один з гравців дотримується стратегії, відповідної сідловій точці, то інший гравець не зможе вчинити краще, ніж дотримуватися стратегії, відповідної сідловій точці. Математично це можна записати у вигляді (15.34):

$$a_{ij_0} \le a_{i_0 j_0} \le a_{i_0 j}, \qquad (15.34)$$

де:

i, j – будь-які чисті стратегії відповідно гравців 1 і 2;

(i_0, j_0) – стратегії, що утворюють сідлову точку.

Таким чином, виходячи з (15.34), сідловій елемент $a_{i_0 j_0}$ є мінімальним в i-му рядку і максимальним в j-му стовпці в матриці A. Відшукування сідлової точки матриці A відбувається наступним чином: в матриці A послідовно в кожному рядку знаходять мінімальний елемент і перевіряють, чи є цей елемент максимальним у своєму стовпці. Якщо так, то він і є сідловій елемент, а пара стратегій, йому відповідна, утворює сідлову точку. Пара чистих стратегій (i_0, j_0) гравців 1 і 2, твірна сідлову точку і сідловій елемент $a_{i_0 j_0}$, називається *роз'язком гри*. При цьому i_0, та j_0 називаються *оптимальними чистими стратегіями* відповідно гравців 1 і 2.

На рис. 15.4 наведено приклад $3D$ –графіка, що демонструє сідлову точку. Вона розташована в центрі $(0,0)$ і має вигляд гіперболічного параболоїда — мінімум уздовж одного напрямку та максимум уздовж іншого.

Рис. 15.4. Приклад зображення сідлової точки

15.5.2.2. Приклад первірки матричної гри на наявність сідлової точки

Постановка задачі

Необхідно перевірити на наявність сідлової точки матрицю виграшів двох гравців (табл. 15.15).

Таблиця 15.15. Матриця виграшів

	B_1	B_2	B_3
A_1	1	-3	-2
A_2	0	5	4
A_3	2	3	2

Розв'язання

Процес виявлення сідлової точки представимо у вигляді табл. 15.16.

Таблиця 15.16. Розрахункова таблиця

	B_1	B_2	B_3	$\min\limits_j(a_{ij})$	$\max\limits_i\left(\min\limits_j(a_{ij})\right)$
A_1	1	-3	-2	-3	2
A_2	0	5	4	0	
A_3	2	3	2	2	
$\max\limits_i(a_{ij})$	2	3	4		
$\min\limits_j\left(\max\limits_i(a_{ij})\right)$	2				

Отже, сідловою точкою є пара $(i_0 = 3; j_0 = 1)$, при якій $\upsilon = \underline{\alpha} = \bar{\alpha} = 2$.

Зауважимо, що хоча виграш в ситуації (3,3) також дорівнює $\underline{\alpha} = \bar{\alpha} = 2$, вона не є сідловою точкою, тому цей виграш не є максимальним серед виграшів третього стовпця.

15.5.2.3. Приклад реалізації первірки матричної гри на наявність сідлової точки в Python

```
import numpy as np

# Матриця виграшів
A = np.array([
    [3, -2, 4],
    [1, 0, -1],
    [2, -3, 5]
])

# Мінімальні значення в кожному рядку
row_min = np.min(A, axis=1)

# Максимальні значення в кожному стовпці
col_max = np.max(A, axis=0)

# Пошук сідлових точок
saddle_points = []

for i in range(A.shape[0]):
    for j in range(A.shape[1]):
        if A[i, j] == row_min[i] and A[i, j] ==
col_max[j]:
            saddle_points.append((i, j, A[i, j]))

# Вивід результату
if saddle_points:
    for (i, j, value) in saddle_points:
        print(f"Сідлова точка знайдена в позиції
({i+1}, {j+1}) зі значенням {value}")
else:
    print("Сідлової точки немає")
```

15.5.3. Ігри порядку 2x2

У загальному випадку гра 2x2 визначається матрицею (15.35)

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \qquad (15.35)$$

Насамперед необхідно перевірити, чи є у даної гри сідлова точка. Якщо так, то гра має рішення в чистих стратегіях, причому оптимальними стратегіями гравців 1 і 2 відповідно будуть чиста максимін і чиста мінімаксна стратегії. Якщо ж гра з матрицею виграшів A не має чистих стратегій, то обидва гравці мають тільки такі оптимальні стратегії, які використовують всі свої чисті стратегії з позитивними ймовірностями. В іншому випадку один з гравців (наприклад 1) має чисту оптимальну

стратегію, а інший – лише змішані. Не обмежуючи спільності, можна вважати, що оптимальною стратегією гравця 1 є вибір з ймовірністю 1 першого рядка. Далі, по властивості 1 випливає, що $a_{11} = a_{12} = \upsilon$ і матриця має вигляд (15.36):

$$\begin{pmatrix} \upsilon & \upsilon \\ a_{21} & a_{22} \end{pmatrix}. \tag{15.36}$$

Легко бачити, що для матриць такого виду одна з стратегій гравця 2 є домінуючою. Отже, цей гравець має чисту стратегію, що суперечить припущенню.

Нехай $X = (\xi, 1-\xi)$ – оптимальна стратегія гравця 1. Оскільки гравець 2 має змішану оптимальну стратегію, отримаємо систему (15.37):

$$\begin{cases} a_{11} \cdot \xi + a_{21} \cdot (1-\xi) = \upsilon, \\ a_{12} \cdot \xi + a_{22} \cdot (1-\xi) = \upsilon. \end{cases} \tag{15.37}$$

Звідси випливає, що при $\upsilon \neq 0$ стовпці матриці A не можуть бути пропорційні з коефіцієнтом пропорційності, відмінним від одиниці. Якщо ж коефіцієнт пропорційності дорівнює одиниці, то матриця A приймає вид (15.38):

$$\begin{pmatrix} a_{11} & a_{11} \\ a_{12} & a_{12} \end{pmatrix} \tag{15.38}$$

і гравець 1 має чисту оптимальну стратегію (він вибирає з ймовірністю 1 ту з рядків, елементи якої не менше відповідних елементів іншого), що суперечить припущенню. Отже, якщо $\upsilon \neq 0$ і гравці мають тільки змішані оптимальні стратегії, то визначник матриці A відмінний від нуля. З цього випливає, що остання система рівнянь має єдине рішення. Вирішуючи її, знаходимо вирази (15.39) та (15.40) для змінних ξ та υ:

$$\xi = \frac{a_{22} - a_{21}}{a_{11} - a_{12} - a_{21} + a_{22}}; \tag{15.39}$$

$$\upsilon = \frac{a_{11} \cdot a_{22} - a_{21} \cdot a_{21}}{a_{11} - a_{12} - a_{21} + a_{22}}. \tag{15.40}$$

Аналогічні міркування приводять нас до того, що оптимальна стратегія гравця 2 $Y = (\eta, 1-\eta)$ задовольняє системі рівнянь (15.41):

$$\begin{cases} a_{11} \cdot \eta + a_{12} \cdot (1-\eta) = \upsilon, \\ a_{21} \cdot \eta + a_{22} \cdot (1-\eta) = \upsilon. \end{cases} \tag{15.41}$$

Звідки випливає рівність (15.42):

$$\eta = \frac{a_{22} - a_{12}}{a_{11} - a_{12} - a_{21} + a_{22}}. \tag{15.42}$$

15.5.4. Геометрична інтерпритація гри 2×2

Розв'язання гри 2×2 допускає наочну геометричну інтерпретацію. Хай гра задана платіжною матрицею $P = (a_{ij})$, $i, j = 1{,}2$. По осі абсцис відкладемо одиничний відрізок A_1A_2 точка A_1 $(x = 0)$ зображає стратегію A_1, а всі проміжні точки цього відрізання — змішані стратегії S_A першого гравця, причому відстань від S_A до правого кінця відрізку — це ймовірність p_1 стратегії A_1, відстань до лівого кінця — ймовірність p_2 стратегії A_2. На перпендикулярних осях I—I і II—II відкладаємо виграші при стратегіях A_1

і A_2 відповідно. Якщо 2-й гравець прийме стратегію B_1, то вона дає виграші a_{11} і a_{21} на осях I—I і II—II, що відповідають стратегіям A_1 і A_2. Позначимо ці точки на осях I—I і II—II буквою B_1. Середній виграш υ_1, відповідний змішаній стратегії S_A, визначається по формулі математичного сподівання $\upsilon_1 = a_{11} \cdot p_1 + a_{21} \cdot p_2$ і дорівнює ординаті точки M_1, яка лежить на відрізку B_1B_1 і має абсцису S_A (рис. 15.5).

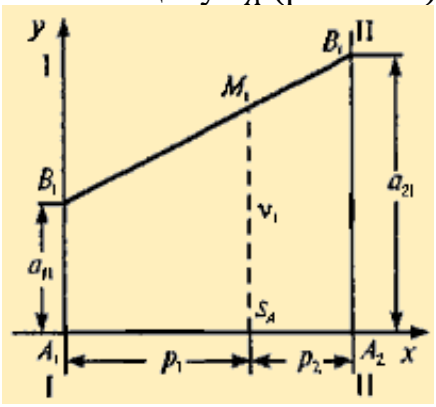


Рис. 15.5. Відрізок B_1B_1

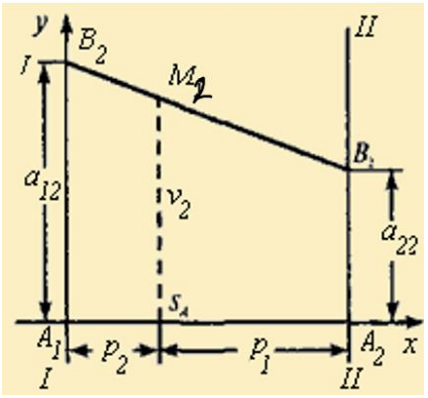


Рис. 15.5. Відрізок B_2B_2

Аналогічно будуємо відрізок B_2B_2, відповідний вживанню другим гравцем стратегії B_2 (рис. 15.6). При цьому середній виграш $\upsilon_2 = a_{12} \cdot p_1 + a_{22} \cdot p_2$ — ордината точки M_2.

Відповідно до принципу мінімакса оптимальна стратегія S_a така, що мінімальний виграш гравця A (при найгіршій поведінці гравця B) звертається в максимум.

Ординати точок, що лежать на ламаній (рис. 15.7), показують мінімальний виграш гравця A при використанні ним будь-якій змішаній стратегії (на ділянці B_1N — проти стратегії B_1, на ділянці NB_2 — проти стратегії B_2). Оптимальну стратегію $S_a^* = (p_1^*, p_2^*)$ визначає точка N, в якій мінімальний виграш досягає максимуму; її ордината дорівнює ціні гри υ.

На рис. 15.7 та рис. 15.8 позначені також верхня і нижня ціни гри α і β. Застосуємо геометричний метод для вирішення наступного прикладу.

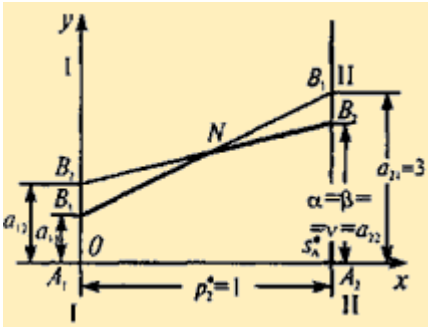


Рис. 15.7. Верхня і нижня ціни гри α і β

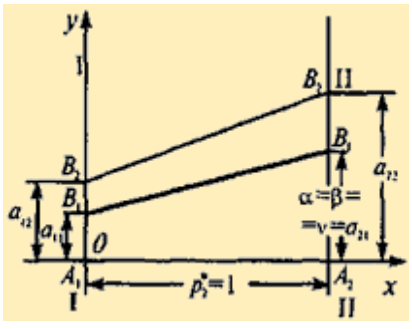


Рис. 15.8. Верхня і нижня ціни гри α і β

15.5.5. Графічний метод розв'язання ігор 2x*n* та *m*x2

Розглянемо метод на прикладах.

Постановка задачі

Деяка гру, задано платіжною матрицею (табл. 15.16). Розв'язати дану матричну гру для обох гравців.

Таблиця 15.16. Платіжна матриця

		Гравець 2		
		B_1	B_2	B_3
Гравець 1	A_1	2	3	11
	A_2	7	5	2

Розв'язання

На площині xOy введемо систему координат і на осі Ox відкладемо відрізок одиничної довжини A_1A_2, кожній точці якого поставимо у відповідність деяку змішану стратегію гравця 1 $(x, 1-x)$. Зокрема, точці $A_1(0;0)$ відповідає стратегія A_1, точці $A_2(1;0)$ — стратегія A_2 і т.д. (рис. 15.9).

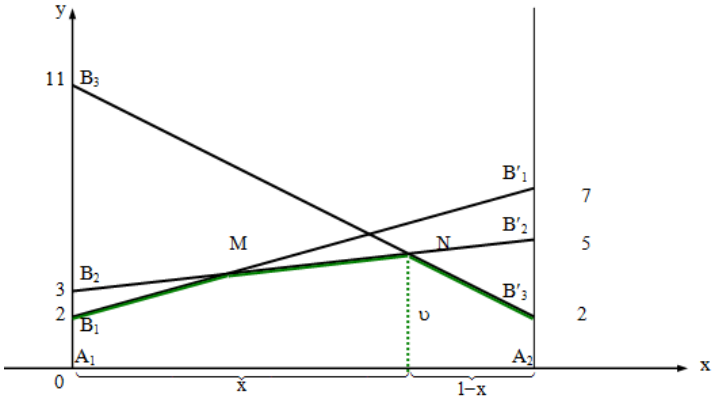


Рис. 15.9. Геометричний метод для гри 2xn

У точках A_1 і A_2 відновимо перпендикуляр і на отриманих прямих будемо відкладати виграш гравців. На першому перпендикулярі (в даному випадку він збігається з віссю Oy) відкладемо виграш гравця 1 при стратегії A_1, а на другому — при стратегії A_2 Якщо гравець 1 застосує стратегію A_1, то виграє при стратегії B_1 гравця 2 — 2, при стратегії B_2 — 3, а при стратегії B_3 — 11. Числам $2, 3, 11$ на осі Ox відповідають точки B_1, B_2 і B_3.

Якщо ж гравець 1 застосує стратегію A_2 то його виграш при стратегії B_1 дорівнює 7, при B_2 — 5, а при B_3 — 2. Ці числа визначають точки B_1', B_2', B_3' на перпендикулярі, відновленому в точці A_2. З'єднуючи між собою точки B_1 і B_1', B_2 і B_2', B_3 і B_3' отримаємо три прямі, відстань до яких від осі Ox визначає середній виграш при будь-якому поєднанні відповідних стратегій. Наприклад, відстань від будь-якої точки відрізка B_1B_1' до осі Ox визначає середній виграш υ_1 при будь-якому поєднанні стратегій A_1A_2 (з частотами x і $1-x$) і стратегією B_1 гравця 2. Це відстань дорівнює

$$2 \cdot x_1 + 6 \cdot (1 - x_2) = \upsilon_1$$

(згадайте планіметрію і розгляньте трапецію $A_1B_1B_1'A_2$). Таким чином, ординати точок, що належать ламаній $B_1MNB_3'A_2$ визначають мінімальний виграш гравця 1 при застосуванні ним будь-яких змішаних стратегій. Ця

мінімальна величина є максимальною в точці N; отже цій точці відповідає оптимальна стратегія $X^* = (x, 1-x)$, а її ордината дорівнює ціні гри υ. Координати точки N знаходимо як точку перетину прямих B_2B_2' і B_3B_3'.

Відповідні два рівняння мають вигляд

$$\begin{cases} 3 \cdot x + 5 \cdot (1-x) = \upsilon, \\ 11 \cdot x + 2 \cdot (1-x) = \upsilon. \end{cases} \Rightarrow x = \frac{3}{11}, \upsilon = \frac{49}{11}.$$

Отже $X = \left(\frac{3}{11}; \frac{8}{11}\right)$, при ціні гри $\upsilon = \frac{49}{11}$. Таким чином ми можемо знайти оптимальну стратегію за допомогою матриці

$$\begin{pmatrix} 3 & 11 \\ 5 & 2 \end{pmatrix}.$$

Оптимальні стратегії для гравця 2 можна знайти з системи

$$\begin{cases} 3 \cdot y + 11 \cdot (1-y) = \upsilon, \\ 5 \cdot y + 2 \cdot (1-y) = \upsilon. \end{cases} \Rightarrow y = \frac{9}{11}.$$

і, відповідно, $Y = \left(0; \frac{9}{11}; \frac{2}{11}\right)$.

15.5.6. Реалізація графічного методу розв'язання ігор 2x2 в Python

```
import numpy as np
import matplotlib.pyplot as plt

# Матриця виграшів (2x2 гра)
A = np.array([[2, 5],  # Виграші для стратегії 1
              [3, 1]])  # Виграші для стратегії 2

# Значення p (ймовірність вибору першої стратегії)
p_values = np.linspace(0, 1, 100)

# Обчислення очікуваних виграшів для обох стратегій
U1 = p_values * A[0, 0] + (1 - p_values) * A[1, 0]  # Стратегія 1
U2 = p_values * A[0, 1] + (1 - p_values) * A[1, 1]  # Стратегія 2

# Візуалізація
plt.plot(p_values, U1, label="Очікуваний виграш (стратегія 1)")
plt.plot(p_values, U2, label="Очікуваний виграш (стратегія 2)")
plt.xlabel("Ймовірність вибору стратегії 1 (p)")
plt.ylabel("Очікуваний виграш")
plt.title("Графічний метод для 2x2 гри")
plt.legend()
```

```
plt.grid()
plt.show()
```

15.5.7. Зведення матричної гри до задачі лінійного програмування

Припустимо, що ціна гри додатна ($\upsilon > 0$). Завжди можна підібрати таке число c, додаток якого до всіх елементів матриці виграшів дає матрицю з позитивними елементами, і отже, з позитивним значенням ціни гри. При цьому оптимальні змішані стратегії обох гравців не змінюються.

Отже, нехай дана матрична гра з матрицею A порядку mxn. Оптимальні змішані стратегії $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_n)$ відповідно гравців 1 і 2 та ціна гри υ повинні задовольняти співвідношенням (15.43) та (15.44):

$$\begin{cases} \sum_{i=1}^{m}(a_{ij} \cdot x_i) \geq \upsilon, (j = \overline{1,n}), \\ \sum_{i=1}^{m} x_i = 1, \\ x_i \geq 0, (i = \overline{1,m}). \end{cases} \quad (15.43)$$

$$\begin{cases} \sum_{j=1}^{n}(a_{ij} \cdot y_j) \leq \upsilon, (i = \overline{1,m}), \\ \sum_{j=1}^{n} y_j = 1, \\ y_j \geq 0, (j = \overline{1,n}). \end{cases} \quad (15.44)$$

Розділимо всі рівняння і нерівності в (15.43) і (15.44) на υ (це можна зробити, тому що по припущенню $\upsilon > 0$) і введемо позначення:

$$\frac{x_i}{\upsilon} = p_i, (i = \overline{1,m}), \frac{y_i}{\upsilon} = q_i, (j = \overline{1,n}).$$

Тоді (15.39) і (15.40) перепишуться у вигляді (15.45) та (15.46):

$$\sum_{i=1}^{m}(a_{ij} \cdot p_i) \geq 1, \sum_{i=1}^{m} p_i = \frac{1}{\upsilon},\ p_i \geq 0, (i = \overline{1,m}), \quad (15.45)$$

$$\sum_{i=1}^{n}(a_{ij} \cdot q_j) \leq 1, \sum_{i=1}^{n} q_{ij} = \frac{1}{\upsilon},\ q_j \geq 0, (j = \overline{1,n}). \quad (15.46)$$

Оскільки перший гравець прагне знайти такі значення x_i та, отже, p_i, щоб ціна гри υ була максимальною, то рішення першої задачі зводиться до знаходження таких невід'ємних значень p_i $(i = \overline{1,m})$, при яких виконується (15.47):

$$\sum_{i=1}^{m} p_i \to min, \sum_{i=1}^{m}(a_{ij} \cdot p_i) \geq 1. \quad (15.47)$$

Оскільки другий гравець прагне знайти такі значення y_j і, отже, q_j, щоб ціна гри υ була найменшою, то рішення другої задачі зводиться до знаходження таких невід'ємних значень q_j, $(j = \overline{1,n})$, при яких справджується (15.48):

$$\sum_{i=1}^{n} q_j \to max, \sum_{i=1}^{n}(a_{ij} \cdot q_j) \leq 1. \quad (15.48)$$

Формули (15.47) і (15.48) описують двоїсті одна одній задачі лінійного програмування (ЛП).

Вирішивши ці завдання, отримаємо значення p_i $(i = \overline{1,m})$, q_j, $(j = \overline{1,n})$ і υ. Тоді змішані стратегії, тобто x_i і y_j обчислюються за формулами (15.49) та (15.50):

$$x_i = \upsilon \cdot p_i, (i = \overline{1,m}), \quad (15.49)$$
$$y_j = \upsilon \cdot q_j, (j = \overline{1,n}). \quad (15.50)$$

15.5.8. Дослідження матричної гри за заданою платіжною матрицею

Розглянемо процес дослідження матричної гри на конкретному прикладі.

Постановка задачі

Для гри $G(2x5)$ заданої платіжної матриці (табл. 15.17):
1. Застосувати принцип домінування.
2. Перевірити, чи є сідлова точка.
3. Розв'язати геометричну задачу для гравця A.
4. Сформулювати задачу лінійного програмування для гравців A та B.

Таблиця 15.17. Платіжна матриця

	B_1	B_2	B_3	B_4	B_5
A_1	40	80	60	102	20
A_2	20	60	40	60	60

Розв'язання

1. Застосуємо принцип домінування, тобто виключимо свідомо невигідні стратегії. Такими стратегіями є: для гравця A-ті, яким відповідають рядки з елементами значно меншими в порівнянні з елементами інших рядків, для гравця B-ті, яким відповідають стовпці з елементами значно більшими в порівнянні з елементами інших стовпців (табл. 15.18).

Таблиця 15.18. Платіжна матриця після застосування принципу домінування

	B_1	B_2
A_1	40	20
A_2	20	60

2. Перевіримо, чи є сідловка точка. Для цього перевіримо, чи виконується умова рівності мінімакса та максиміна (табл. 15.19).

Таблиця 15.19. Матриця виграшів

	B_1	B_2	$\min_j(a_{ij})$	$\max_i\left(\min_j(a_{ij})\right)$
A_1	40	20	20	20
A_2	20	60	20	
$\max_i(a_{ij})$	40	60		
$\min_j\left(\max_i(a_{ij})\right)$	40			

$$\underline{\alpha} = \max_i\left(\min_j(a_{ij})\right) = 20, \bar{\alpha} = \min_j\left(\max_i(a_{ij})\right) = 40, \underline{\alpha} \neq \bar{\alpha}.$$

Тобто ми маємо гру без сідлової точки.

3. Розв'яжемо задачу графічно для гравця A (рис. 15.10).

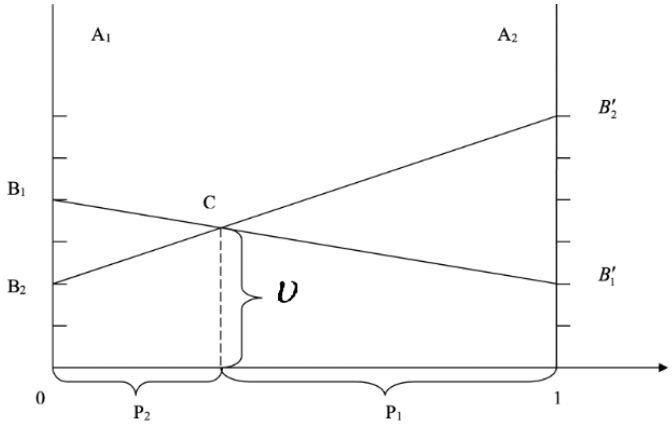


Рис. 15.10. Графічне розв'язання задачі

Точка C – це точка перетину прямих. Складемо рівняння прямої $B_1B_1^{'}, B_1(0;40), B_1^{'}(1;20)$.

$$\frac{x-x_0}{x_1-x_0}=\frac{y-y_0}{y_1-y_0}$$

$$x=\frac{y-40}{20-40}$$

$$y=-20\cdot x+40.$$

Складемо рівняння прямої $B_2B_2^{'}, B_2(0;20), B_2^{'}(1;60)$.

$$\frac{x-0}{1-0}=\frac{y-20}{60-20}$$

$$x=\frac{y-20}{40}$$

$$y=40\cdot x+20.$$

$$\begin{cases} y=-20\cdot x+40 \\ y=40\cdot x+20. \end{cases}$$

$$\begin{cases} y=-20\cdot x+40 \\ x=\frac{1}{3}. \end{cases}$$

$$\begin{cases} y=\frac{100}{3} \\ x=\frac{1}{3}. \end{cases}$$

Точка $C\left(\frac{1}{3};\frac{100}{3}\right)$.

Тобто,

$$P_1=\frac{2}{3}, P_2=\frac{1}{3}, \upsilon=\frac{100}{3}.$$

4. Сформулюємо задачі лінійного програмування для гравців A і B.

Для гравця A:

$$x_i=\frac{p_i}{\upsilon}$$

$$x_1+x_2\to min$$

$$\begin{cases} 40 \cdot x_1 + 20 \cdot x_2 \geq 1 \\ 20 \cdot x_1 + 60 \cdot x_2 \geq 1 \end{cases}$$
$$x_i \geq 0.$$

Для гравця B:

$$y_i = \frac{q_j}{\upsilon}$$
$$y_1 + y_2 \to max$$
$$\begin{cases} 40 \cdot y_1 + 20 \cdot y_2 \leq 1 \\ 20 \cdot y_1 + 60 \cdot y_2 \leq 1 \end{cases}$$
$$y_j \geq 0.$$

15.5.9. Симплекс-метод

15.5.9.1. Опис методу

Симплекс-метод – це числовий алгоритм для розв'язання оптимізаційних задач лінійного програмування. Він широко використовується для пошуку максимуму або мінімуму лінійної функції у випадку обмежень у вигляді лінійних нерівностей.

Основна ідея сімплекс-методу полягає в тому, щоб визначити точку в n-вимірному просторі (називається сімплексом), де значення цільової функції досягає свого максимуму або мінімуму. Потім метод переміщається з точки в точку сімплекса, поки не буде досягнута оптимальна точка.

Основні етапи сімплекс-методу

Етап 1. Стартова точка. Зазвичай це точка, де всі змінні дорівнюють нулю або мають якісь інші початкові значення.

Етап 2. Побудова сисмплексу. Починаючи з початкової точки, побудувати сімплекс:

- Кожна вершина симплексу відповідає можливому базисному рішенню;
- У багатовимірному просторі вершини є точками, де кількість невідомих дорівнює кількості обмежень.

Етап 3. Перевірка умови зупинки. Перевіряється, чи досягнута оптимальна точка або неможливо покращити значення цільової функції.

Етап 4. Вибір напрямку. Вибирається напрямок для руху сімплексу, що призведе до покращення значення цільової функції.

Етап 5. Крок. Обчислюється величина кроку або відстань, на яку слід змінити поточну точку сімплексу.

Етап 6. Оновлення сімплексу. Сімплекс переміщується в нову точку згідно з обчисленим кроком.

Етап 7. Повторення кроків. Процес повторюється до досягнення оптимального рішення або до виявлення неможливості подальшого покращення.

Симплекс-метод є ефективним алгоритмом для розв'язання багатьох практичних задач лінійного програмування, таких як планування виробництва, управління запасами, логістичне планування та багато інших.

15.5.9.2. Приклад симплекс-методу

Постановка задачі

Припустимо, у нас є наступна задача оптимізації. Необхідно максимізувати цільову функцію: $z = 2 \cdot x_1 + 3 \cdot x_2$

З обмеженнями:

$$\begin{cases} x_1 + x_2 \leq 5, \\ 2 \cdot x_1 + x_2 \leq 8, \\ x_1, x_2 \geq 0. \end{cases}$$

Розв'язання

Виконаємо послідовно всі етапи.

Етап 1. Почнемо з початкової точки (0,0).

Етап 2. Побудуємо симплекс, включаючи вершини, що відповідають обмеженням. Одна вершина буде (5,0), друга – (0,8).

Етап 3. Перевіримо, чи досягнута оптимальна точка.

Етап 4. Оберемо напрямок, що призведе до максимального зростання цільової функції. Для цього визначимо оптимальну вершину симплексу.

Етап 5. Обчислимо величину кроку у напрямку до оптимальної вершини.

Етап 6. Перемістимо симплекс до нової точки відповідно до обчисленого кроку.

Етап 7. Повторюємо процес до досягнення оптимального рішення або до виявлення неможливості подальшого покращення.

Це лише загальний опис процесу. Конкретні кроки розв'язку задачі за допомогою симплекс-методу вимагають ретельного обчислення та маніпуляцій з матрицями (рис. 15.11-15.13).

$F(x) = 2x_1 + 3x_2 \rightarrow \max$

$\begin{cases} x_1 + x_2 \leq 5 \\ 2x_1 + x_2 \leq 8 \end{cases}$

$\Rightarrow$

$F(x) = 2x_1 + 3x_2 + 0x_3 + 0x_4 \rightarrow \max$

$\begin{cases} x_1 + x_2 + x_3 = 5 \\ 2x_1 + x_2 + x_4 = 8 \end{cases}$

Рис. 15.11. Перетворення матриць при симплекс-методі

B	Cb	P	x_1	$x_2 \downarrow$	x_3	x_4	Q
			2	3	0	0	
$x_3 \leftarrow$	0	5	1	1	1	0	5
x_4	0	8	2	1	0	1	8
max		0	-2	-3	0	0	

Рис. 15.12. Попередній етап. Ітерація 1

B	Cb	P	x_1	x_2	x_3	x_4	Q
			2	3	0	0	
x_2	3	5	1	1	1	0	
x_4	0	3	1	0	-1	1	
max		15	1	0	3	0	

Рис. 15.13. Обчислення елементів таблиці. Ітерація 2

Отже, $F^* = 15, X^* = (0; 5)$.

15.5.9.3. Приклад сімплекс-методу

```
from scipy.optimize import linprog

# Коефіцієнти цільової функції (мінімізуємо -Z, бо
linprog мінімізує)
c = [-2, -3]  # Мінус, бо лінійне програмування
мінімізує

# Коефіцієнти обмежень (матриця A)
A = [
    [1, 1],   # x1 + x2 ≤ 5
    [2, 1]    # 2x1 + x2 ≤ 8
]

# Вільні члени обмежень (вектор b)
b = [5, 8]

# Обмеження змінних (x1, x2 ≥ 0)
x_bounds = [(0, None), (0, None)]

# Використовуємо симплекс-метод
res = linprog(c, A_ub=A, b_ub=b, bounds=x_bounds,
method="simplex")

# Вивід результату
if res.success:
    print(f"Оптимальне значення Z = {-res.fun:.2f}")
    print(f"x1 = {res.x[0]:.2f}, x2 =
{res.x[1]:.2f}")
else:
    print("Розв'язку немає або метод не збіжний.")
```

Отже, $F^* = 12, X^* = (3; 2)$.

15.6. Властивості бінарних відношень

15.6.1. Бінарне відношення та його властивості

В реальних ситуаціях часто буває важко або неможливо дати характеристику окремого об'єкта $a \in A$ з множини A, який є альтернанивою при виборі (найкращій кандидат на роботу, найбільш підходяща в домашні улюбленці тварина, найбільш зручний для проживання будинок, тощо), у вигляді числового значення. Але якщо розглядати об'єкт не окремо, а в парі з іншими об'єктами цієї множини, то знаходяться підстави сказати, який з них краще (переважає інший).

Таким чином, процес порівняння таких об'єктів можна математично записати у вигляді бінарних відношень (15.51):

$$R_1 = \{\text{Об}'\text{єкт } a \text{ кращій за об}'\text{єкт } b | \ a, b \in A\};$$
$$R_2 = \{\text{Об}'\text{єкт } a \text{ гірний за об}'\text{єкт } b | \ a, b \in A\};$$
$$R_3 = \{\text{Об}'\text{єкт } a \text{ еквівалентний об}'\text{єкту } b | \ a, b \in A\}. \quad (15.51)$$

Бінарне відношення в математиці – це підмножина декартового добутку множини на себе. Його можна представити у вигляді пар елементів з початкової множини. Відношення може мати різні властивості, які характеризують його поведінку та структуру.

Властивості бінарних відношень

Рефлексивність. Відношення R на множині A є *рефлексивним*, якщо для кожного елемента a з A $(a \in A)$ пара (a, a) також належить до R. Це означає, що кожен елемент множини пов'язаний із собою.

Симетричність. Відношення R на множині A є *симетричним*, якщо для кожного пари елементів (a, b) з А, якщо (a, b) належить до R, то (b, a) також належить до R.

Транзитивність. Відношення R на множині A є *транзитивним*, якщо для кожних трьох елементів a, b, c з A, якщо (a, b) і (b, c) належать до R, то (a, c) також належить до R.

Антисиметричність. Відношення R на множині A є *антисиметричним*, якщо для кожного пари елементів (a, b) з A, якщо (a, b) належить до R і (b, a) також належить до R, то $a = b$.

Асиметричність. Відношення R на множині A є асиметричним, якщо для кожної пари елементів (a, b) з A, або тільки (a, b) належить до R, або тільки (b, a) належить до R.

Антирефлексивність. Відношення R на множині A є *антирефлексивним*, якщо для жодного елемента a з A $(a \in A)$, пара (a, a) не належить до R.

Тотальність. Відношення R на множині A є *тотальним*, якщо для будь-яких двох елементів a, b з A, або (a, b) належить до R, або (b, a) належить до R, або і (a, b) , і (b, a) належать до R.

Ці властивості корисні, як взаємодіє множина елементів у бінарному відношенні, що допомагає в аналізі та розумінні його властивостей.

15.6.2. Матриця парних порівнянь

Визначити бінарне відношення R означає тим чи іншим способом вказати всі пари об'єктів, для котрих виконується R. Існує три способи визначення бінарних відношень:

- Переліком (списком) усіх пар;
- За допомогою графа;
- У вигляді матриць парних порівнянь (МПП).

Зупинимось на матриці парних порівнянь.

Матриця парних порівнянь – це інструмент, який використовується для порівняння кількох елементів у відношенні до одного на основі певного критерію. Надалі ці елементи будемо називати *альтернативами*. Ця матриця використовується в аналізі рішень, у багатьох методах оптимізації, при аналізі експертних оцінок та в багатьох інших.

Основна структура матриці парних порівнянь – це квадратна матриця, де рядки і стовпці – це альтернативи, і кожен елемент матриці оцінює ступінь переваги або важливості однієї альтернативи над іншими.

Елементи МПП можуть визначатись, наприклад у вигляді (15.52):

$$a_{ij} = \begin{cases} 0, a_i \cong a_j, \\ 1, a_i \succ a_j, \quad a_i, a_j \in A, \\ -1, a_i \prec a_{j,} \end{cases} \qquad (15.52)$$

де $\succ, \prec, \cong$ – відношення «краще», «гірше», «однакові» відповідно.

Наприклад, якщо ми порівнюємо п'ять альтернативних варіантів A, B, C, D і E щодо їх вартості, то матриця парних порівнянь також може виглядати наступним чином (табл. 15.20):

Таблиця 15.20. Матриця парних порівнянь

	A	B	C	D	E
A	1	3	5	2	4
B	1/3	1	2	1/2	3
C	1/5	1/2	1	1/3	2
D	1/2	2	3	1	4
E	1/4	1/3	1/2	1/4	1

Елементи матриці (табл. 15.20) вказують, на рівень переваги однієї альтернативи над іншими. Наприклад, число 3 у клітинці в рядку B і стовпці A означає, що альтернатива A має утричі більшу перевагу над альтернативою B.

Сполучення зі згаданих властивостей дозволяє визначити різні типи відношень об'єктів з множини A.

Транзитивність відношень особливо важлива у теорії вибору та прийняття рішень, оскільки ця властивість виражає деякі природні взаємозв'язки між об'єктами. Зрозуміло, що циклічна тріада означає непослідовність в твердженнях експертів. Зауважимо, що чим менше

циклічних тріад має МПП тим більш послідовним можна вважати експерта у своїх судженнях.

Матриця називається *узгодженою*, коли в ній повністю відсутні циклічні тріади. Перевірка на транзитивність дозволяє проаналізувати і, у випадку необхідності, скорегувати логіку мислення експерта.

15.6.3. Метод рядкових сум

Ґрунтуючись на бінарних властивостях, які визначені для пар об'єктів $\forall a_i, a_j \in A, i, j \in J$, можна провести ранжування всіх об'єктів на основі різних методів.

Один з підходів до структурування об'єктів ґрунтується на методі рядкових сум.

Метод рядкових сум (*метод Лапласа*) є одним з підходів до прийняття рішень в умовах невизначеності, коли немає чітких ймовірностей для можливих станів природи. Він базується на принципі рівноймовірності всіх можливих станів.

Метод рядкових сум також застосовується в теорії ігор для розв'язку матричних ігор $2 \times n$. Він дозволяє знайти найкращі стратегії гравця, який мінімізує свій максимальний виграш.

Алгоритм методу рядкових сум

Крок 1. Побудувати матрицю парних порівнянь A.

Крок 2. Обчислити ваги для кожної альтернативи матриці A за формулою (15.53):

$$\omega_j = \sum_{i=1}^{n} a_{ij}, \qquad (15.53)$$

де:

$i, j \in J = \{1, .., n\}$;

n – загальна кількість об'єктів;

ω_j – вага альтернативи з матриці порівнянь;

a_{ij} – елементи матриці порівнянь, які визначаються за принципом (15.48).

Крок 3. Вибрати альтернативу з найвищим значенням ваги.

Крок 4. Провести загальне ранжування альтернатив. Альтернативі з максимальним значенням ω_j відводиться перше місце в ранжуванні, альтернативі з мінімальним значенням – останнє місце.

Приклад реалізації методу рядкових сум у Python

Розглянемо задачу з такою матрицею парних порівнянь:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}.$$

Зразок коду

```
import numpy as np
```

```
# Матриця парних порівнянь
A = np.array([[0, 1, 1],
              [-1, 0, 1],
              [-1, -1, 0]])

# Обчислення рядкових сум (середніх значень виграшу)
row_sums = A.mean(axis=1)

# Вибір альтернативи з максимальним середнім виграшем
optimal_strategy = np.argmax(row_sums) + 1  # +1 для нумерації з 1

print(f"Середні виграші для кожної альтернативи: {row_sums}")
print(f"Найкраща альтернатива за методом рядкових сум: Альтернатива {optimal_strategy}")
```

Результат роботи коду

```
Середні виграші для кожної альтернативи: [2.0 0.0 -2.0]
Найкраща альтернатива за методом рядкових сум: Альтернатива 1
```

15.6.4. Застосування методу рядкових сум

Постановка задачі

Розглянемо ЗПР визначення складових здорового способу життя та встановлення пріоритетів між ними.

У табл. 15.21 представлено множину заходів, що складають здоровий спосіб життя:

1. Порівняти заходи між собою за привабливістю (значущістю, важливістю тощо) методом парних порівнянь. Результати порівнянь звести у відповідну МПП.
2. Методом рядкових сум визначити найбільш значущій об'єкт.
3. Перевірити отриману матрицю на узгодженість.
4. Зробити висновок щодо послідовності оцінювання.

Таблиця 15.21. Перелік заходів, що складають здоровий спосіб життя

Захід	Формальна позначка
Харчування	a_1
Фізична активність	a_2
Загартування	a_3
Позитивні емоції	a_4

Відмова від шкідливих звичок	a_5
Розвиток духовності	a_6

Розв'язання

1. Порівняємо заходи між собою за ступенем значущості та результати порівнянь представимо відповідною МПП (табл. 15.22), де значення кожної комірки формується умовами (15.48). Добавляємо стовпчик для підрахунку рядкових сум:

Таблиця 15.22. Підрахунок рядкових сум

	a_1	a_2	a_3	a_4	a_5	a_6	ω_j
a_1	0	-1	-1	1	1	0	0
a_2	1	0	1	0	-1	1	2
a_3	1	-1	0	1	1	-1	1
a_4	-1	0	-1	0	1	-1	-2
a_5	-1	1	-1	-1	0	1	-1
a_6	0	-1	1	1	-1	0	0

Примітка: *заповнюємо лише верхню трикутну частину матриці, а нижню заповнюємо дзеркально-протилежно*.

2. Методом рядкових сум встановлюємо порядок серед об'єктів і виділяємо найбільш значущий об'єкт. У табл. 15.23 його виділено жовтим кольором.

Таблиця 15.23. Порядок серед об'єктів

Порядок	ω_j	Захід
a_2	2	Фізична активність
a_3	1	Загартування
a_1, a_6	0	Харчування, розвиток духовності
a_5	-1	Відмова від шкідливих звичок
a_4	-2	Позитивні емоції

Порядок також можна представити ранжуванням $a_2 \succ a_3 \succ a_1 \cong a_6 \succ a_5 \succ a_4$.

3. Перевіряємо матрицю на узгодженість експертних оцінок, або іншими словами на транзитивність відношень. Для цього перевіряємо послідовно всі можливі комбінації відношень на 3-х об'єктах за означенням транзитивності. Оскільки МПП заповнювалась протилежно-дзеркально, достатньо це зробити лише для однієї трикутної частини. Представимо процес перевірки у вигляді табл. 15.24:

Таблиця 15.24. Перевірка МПП

№	Комбінації 3-х об'єктів	Умова транзитивності	Відношення	Примітка
1	a_1, a_2, a_3	$a_1 \prec a_2, a_2 \succ a_3 \rightarrow$ транзитивність не може бути перевірено		

2	a_1, a_2, a_4	$a_1 \prec a_2, a_2 \cong a_4 \rightarrow$ транзитивність не може бути перевірено		
3	a_1, a_2, a_5	$a_1 \prec a_2, a_2 \prec a_5 \rightarrow a_1 \prec a_5$	$a_1 \succ a_5$	-
4	a_1, a_2, a_6	$a_1 \prec a_2, a_2 \succ a_6 \rightarrow$ транзитивність не може бути перевірено		
5	a_1, a_3, a_4	$a_1 \prec a_3, a_3 \succ a_4 \rightarrow$ транзитивність не може бути перевірено		
6	a_1, a_3, a_5	$a_1 \prec a_3, a_2 \succ a_5 \rightarrow$ транзитивність не може бути перевірено		
7	a_1, a_3, a_6	$a_1 \prec a_3, a_3 \prec a_6 \rightarrow a_1 \prec a_6$	$a_1 \cong a_6$	-
8	a_1, a_4, a_5	$a_1 \succ a_4, a_4 \succ a_5 \rightarrow a_1 \succ a_5$	$a_1 \succ a_5$	+
9	a_1, a_4, a_6	$a_1 \succ a_4, a_4 \prec a_6 \rightarrow$ транзитивність не може бути перевірено		
10	a_1, a_5, a_6	$a_1 \succ a_5, a_5 \succ a_6 \rightarrow a_1 \succ a_6$	$a_1 \cong a_6$	-
11	a_2, a_3, a_4	$a_2 \succ a_3, a_3 \succ a_4 \rightarrow a_2 \succ a_4$	$a_2 \cong a_4$	-
12	a_2, a_3, a_5	$a_2 \succ a_3, a_3 \succ a_5 \rightarrow a_2 \succ a_5$	$a_2 \prec a_5$	-
13	a_2, a_3, a_6	$a_2 \succ a_3, a_3 \prec a_6 \rightarrow$ транзитивність не може бути перевірено		
14	a_2, a_4, a_5	$a_2 \cong a_4, a_4 \succ a_5 \rightarrow$ транзитивність не може бути перевірено		
15	a_2, a_4, a_6	$a_2 \cong a_4, a_2 \prec a_6 \rightarrow$ транзитивність не може бути перевірено		
16	a_2, a_5, a_6	$a_2 \prec a_5, a_5 \succ a_6 \rightarrow$ транзитивність не може бути перевірено		
17	a_3, a_4, a_5	$a_3 \succ a_4, a_4 \succ a_5 \rightarrow a_3 \succ a_5$	$a_3 \succ a_5$	+
18	a_3, a_4, a_6	$a_3 \succ a_4, a_4 \prec a_6 \rightarrow$ транзитивність не може бути перевірено		
19	a_3, a_5, a_6	$a_3 \succ a_5, a_5 \succ a_6 \rightarrow a_3 \succ a_6$	$a_3 \prec a_6$	-
20	a_4, a_5, a_6	$a_4 \succ a_5, a_5 \succ a_6 \rightarrow a_4 \succ a_6$	$a_4 \prec a_6$	-

4. Результат аналізу та висновки: перевірка на транзитивність показала що у 7-ми випадках з 9-ти можливих для перевірки транзитивність була порушена. Отже, експерт у своїх оцінках був непослідовним приблизно на 22%.

ЛЕКЦІЯ 16

ВІРТУАЛІЗАЦІЯ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

16.1. Віртуалізація даних

16.1.1. Опис процесу віртуалізації даних

Віртуалізація даних — це процес створення уніфікованого логічного представлення даних із різних джерел без потреби їх переміщення або копіювання. Це дозволяє користувачам працювати з даними так, ніби вони зберігаються в одному місці, хоча фактично вони можуть бути розподілені по різних базах даних, хмарах, API тощо (рис. 16.1).

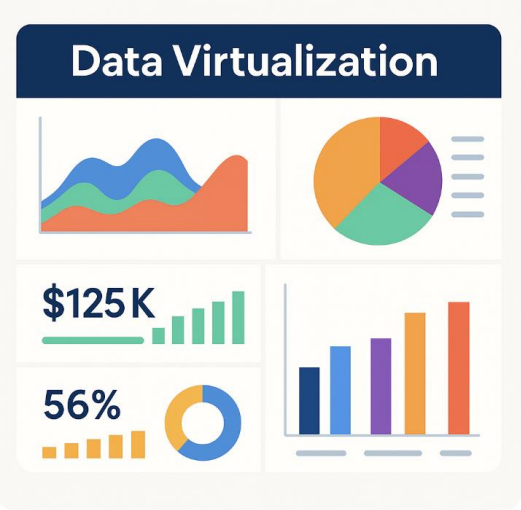


Рис. 16.1. Дашборд Віртуалізації даних

Основні можливості віртуалізації даних:

- Єдиний інтерфейс доступу до даних з різних джерел (SQL, $NoSQL$, $Excel$, API…);
- Об’єднання в реальному часі без ETL ($Extract$, $Transform$, $Load$);
- Покращена продуктивність аналітики;
- Масштабованість, тобто легко інтегрувати нові джерела даних.

Приклад застосування. Уявіть, що у вас є:

- Дані про продажі в $Excel$;
- Дані про клієнтів у SQL;
- Поведінкові метрики в Google Analytics API.

Віртуалізація дозволяє поєднати всі ці дані в одну аналітичну панель без дублювання даних.

На рис. 16.2 та табл. 16.1 наведено порівняння віртуалізації даних і традиційного зберігання (ETL-підходу).

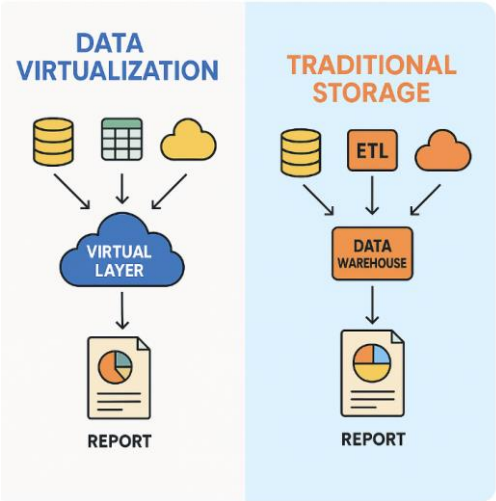


Рис. 16.2. Схема «Традиційне зберігання vs Віртуалізація»

Таблиця 16.1. Порівняння віртуалізації даних з традиційним зберіганням

Критерій	Віртуалізація даних	Традиційне зберігання (ETL)
Доступ до даних	У реальному часі	Після завантаження в хранилище
Фізичне копіювання	Немає копій — працює на логічному рівні	Дані копіюються в окреме сховище
Час оновлення	Миттєве, при запиті	Залежить від графіку завантаження (напр., раз на день)
Гнучкість інтеграції	Висока: легко підключити нові джерела	Менш гнучко: потребує налаштування ETL-процесів
Продуктивність	Може залежати від джерел (високе навантаження)	Висока, бо все зберігається локально
Вартість зберігання	Менша (немає дублювання даних)	Вища (потрібно зберігати копії)
Складність реалізації	Потребує конекторів, але спрощує інфраструктуру	Вимагає складних ETL-конвеєрів і супроводу
Безпека та контроль	Дані залишаються у джерелі	Потрібно захищати окреме сховище
Типові інструменти	Denodo, TIBCO, SAP HANA SDA, Red Hat	Talend, Informatica, Apache Nifi, Microsoft SSIS
Використання	Self-service BI, аналітика в реальному часі	Масова підготовка даних, звітність, історичний аналіз

Рекомендації, щодо використання віртуалізації даних і традиційного зберігання наведено в табл. 16.2.

Таблиця 16.2. Схеми використання та рекомендований підхід

Ситуація	Рекомендований підхід
Потрібно швидко об’єднати дані	Віртуалізація
Аналіз історичних даних	Традиційне зберігання
Складна трансформація даних	Традиційне зберігання
Динамічна, інтерактивна аналітика	Віртуалізація

Ключові етапи процесу

Етап 1. Підключення до джерел даних:
- Бази даних (SQL, $NoSQL$);
- Файли (CSV, $Excel$, XML);
- API, хмари, веб-сервіси.

Етап 2. Створення віртуального рівня:
- Немає копіювання даних;
- Віртуальні представлення (views);
- Об’єднання різних форматів.

Етап 3. Інтеграція та трансформація:
- Фільтрація, агрегація;
- Узгодження схем.

Етап 4. Доступ для користувачів:
- Через BI-інструменти (Tableau, Power BI);
- Через SQL-запити, $REST\ API$.

16.1.2. Інструменти віртуалізації даних

16.1.2.1. Denodo

Denodo — це сучасна платформа для віртуалізації даних, яка дозволяє компаніям отримувати доступ до різноманітних джерел даних (SQL-бази, $Excel$, веб-сервіси, хмарні сервіси тощо) без фізичного переміщення або дублювання даних (рис. 16.3).

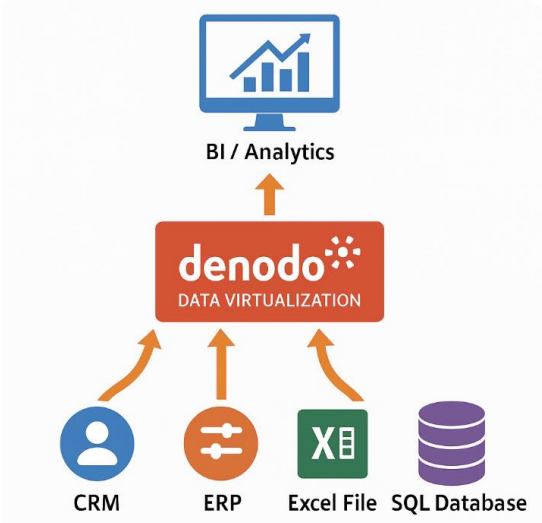


Рис. 16.3. Приклад дашборду з використанням Denodo

Основні можливості Denodo:

- *Data Virtualization* (віртуалізація даних) — інтеграція в режимі реального часу з багатьох джерел.
- *Semantic Layer* — створення єдиного представлення даних для користувачів та аналітиків.
- *Кешування* — підвищення продуктивності при збереженні гнучкості віртуалізації.
- *Безпека та контроль доступу* — централізоване управління політиками доступу до даних.
- *Підтримка BI-інструментів* — Tableau, Power BI, Qlik, і навіть Python або R через *API*.

Tableau — це один з найпопулярніших інструментів для візуалізації даних (буде розглянуто нижче).

Power BI — це потужний інструмент бізнес-аналітики, розроблений корпорацією Майкрософт. Він дає змогу користувачам візуалізувати дані, ділитися аналітикою з усією організацією або вбудовувати їх у програми чи веб-сайти (буде розглянуто нижче).

Qlik — ще один великий гравець у сфері бізнес-аналітики (BI) та аналізу даних, відомий своєю потужною асоціативною моделлю даних та можливостями дослідження, керованими користувачем.

Переваги Denodo та їх опис наведено в табл. 16.3.

Таблиця 16.3. Переваги Denodo

Перевага	Опис
Швидкий доступ до даних	Без *ETL*-процесів, у режимі реального часу
Інтеграція з різними джерелами	Різні бази, файли, хмари, *API* — все в одному місці
Централізоване управління	Один рівень керування даними незалежно від їхнього розташування
Високий рівень безпеки	Автентифікація, авторизація, маскування та логування доступу

Приклад використання. Уявімо, компанія має:

- CRM у Salesforce;
- ERP у SAP;
- Аналітику у Google BigQuery.

Замість об’єднання цих даних в одну базу, Denodo дозволяє створити віртуальну модель, до якої можна звертатися як до єдиної таблиці — все це в реальному часі!

16.1.2.2. Red Hat Data Virtualization

Red Hat JBoss Data Virtualization — це рішення для віртуалізації даних, яке дозволяє об'єднувати розподілені джерела даних (реляційні бази, NoSQL, Hadoop, веб-сервіси тощо) в єдину логічну модель без фізичного

переміщення даних. Це забезпечує централізований доступ до даних для аналітики, BI, додатків і користувачів через стандартні інтерфейси, такі як JDBC, OData REST або ODBC (рис. 16.4).

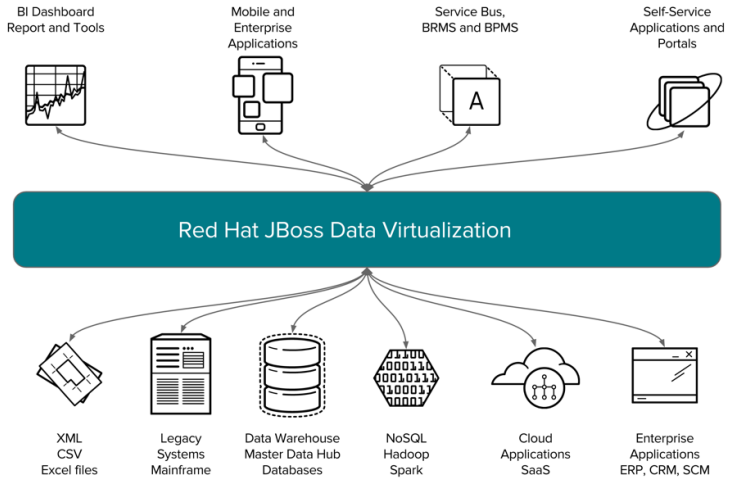


Рис. 16.4. Red Hat JBoss Data Virtualization

Основні компоненти (Data Virtualization Service):

- *Teiid Server* — ядро системи, що відповідає за обробку запитів, оптимізацію та виконання;
- *Віртуальні бази даних (VDB)* — логічні моделі, які поєднують дані з різних джерел у вигляді єдиної структури;
- *Конектори до джерел* — підтримка широкого спектра джерел: від SQL/NoSQL до файлів, API та хмарних сервісів;
- *Teiid Admin Console* — веб-інтерфейс для адміністрування, моніторингу та налаштування.

Схематично Data Virtualization Service наведені на рис. 16.5.

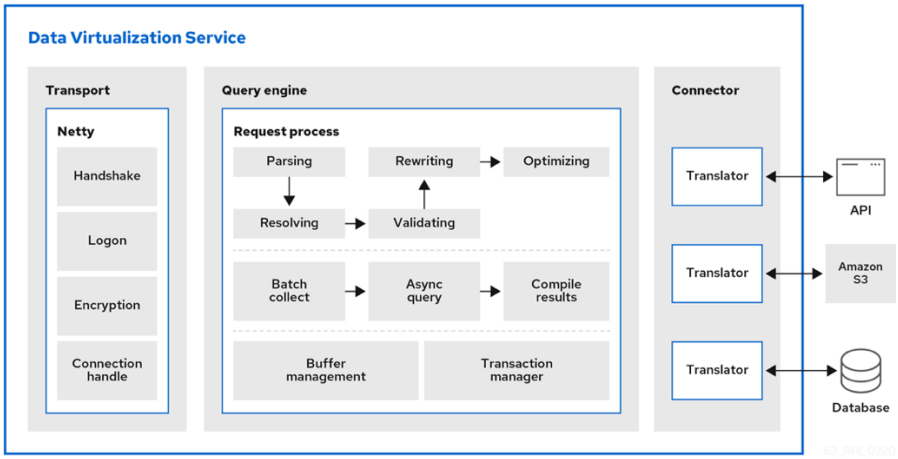


Рис. 16.5. Data Virtualization Service

Ключові переваги:

- *Уніфікований доступ до даних* — користувачі працюють з даними, як з єдиної бази, незалежно від фізичного розташування чи формату;
- *Зниження витрат на інтеграцію* — відсутність потреби в дублюванні або ETL-процесах;

- *Гнучкість* — швидке додавання нових джерел без зміни архітектури;
- *Підтримка масштабування* — працює в контейнеризованих середовищах, таких як OpenShift.

Приклади використання:

- *Бізнес-аналітика* — швидкий доступ до агрегованих даних з різних систем для побудови звітів і дашбордів;
- *Міграція даних* — створення віртуальних шарів для поступового переходу між системами;
- *Інтеграція в реальному часі* — об'єднання даних з різних джерел без затримок для оперативного прийняття рішень.

16.1.2.3. TIBCO Data Virtualization

TIBCO Data Virtualization (*TDV*) — це платформа, яка дозволяє об'єднувати дані з різних джерел у єдину логічну модель без фізичного переміщення даних. Вона забезпечує централізований доступ до даних для аналітики, BI, додатків і користувачів через стандартні інтерфейси, такі як JDBC, OData REST або ODBC (рис. 16.6).

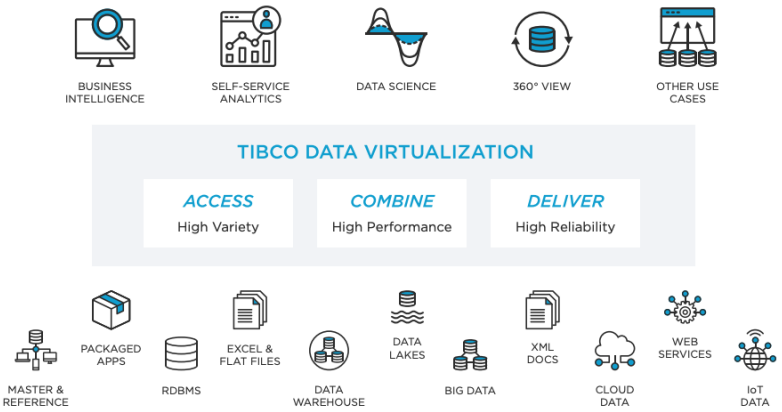


Рис. 16.6. TIBCO Data Virtualization

Основні компоненти:

- *TDV Server* — ядро системи, що відповідає за обробку запитів, оптимізацію та виконання;
- Віртуальні бази даних (*VDB*) — логічні моделі, які поєднують дані з різних джерел у вигляді єдиної структури;
- *Конектори до джерел* — підтримка широкого спектра джерел: від SQL/NoSQL до файлів, API та хмарних сервісів;
- *TDV Studio* — веб-інтерфейс для адміністрування, моніторингу та налаштування.

Ключові переваги:

- *Уніфікований доступ до даних* — користувачі працюють з даними, як з єдиної бази, незалежно від фізичного розташування чи формату;
- *Зниження витрат на інтеграцію* — відсутність потреби в дублюванні або ETL-процесах;
- *Гнучкість* — швидке додавання нових джерел без зміни архітектури;

- *Підтримка масштабування* — працює в контейнеризованих середовищах, таких як OpenShift.

Приклади використання:

- *Бізнес-аналітика* — швидкий доступ до агрегованих даних з різних систем для побудови звітів і дашбордів;
- *Міграція даних* — створення віртуальних шарів для поступового переходу між системами;
- *Інтеграція в реальному часі* — об'єднання даних з різних джерел без затримок для оперативного прийняття рішень.

16.1.2.4. SAP HANA Smart Data Access

SAP HANA Smart Data Access (*SDA*) — це технологія віртуалізації даних, яка дозволяє SAP HANA отримувати доступ до віддалених джерел даних так, ніби вони є локальними таблицями, без необхідності фізичного копіювання або реплікації даних (рис. 16.7).

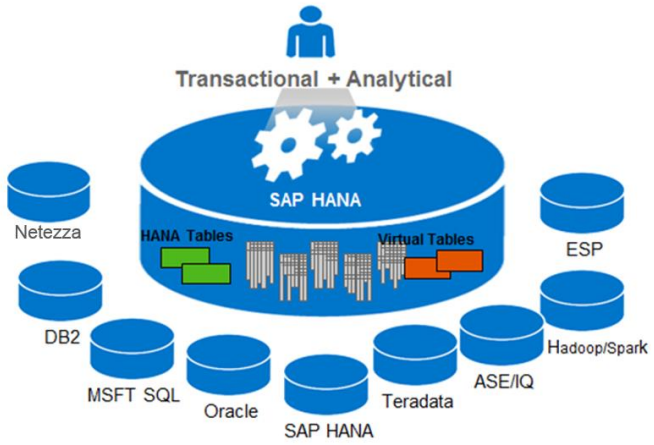


Рис. 16.6. SAP HANA Smart Data Access

Основні можливості SAP HANA Smart Data Access:

- *Віртуальні таблиці.* SDA дозволяє створювати віртуальні таблиці в SAP HANA, які відображаються на віддалені таблиці в інших системах. Це забезпечує прозорий доступ до даних без їхнього дублювання.
- *Підтримка різноманітних джерел даних.* SDA підтримує підключення до різних віддалених систем, таких як SAP HANA, Sybase, Teradata, Apache *Hadoop, Oracle та інші.*
- *Оптимізація* запитів. Запити до віртуальних таблиць оптимізуються та виконуються частково на віддалених джерелах, що зменшує обсяг переданих даних і підвищує продуктивність.
- *Підтримка стандартних інтерфейсів.* Доступ до віртуальних таблиць здійснюється через стандартні інтерфейси, такі як ODBC та JDBC.

Основні компоненти SDA:

- Віртуальні таблиці: Це об'єкти в SAP HANA, які відображаються на таблиці в віддалених джерелах даних. Вони дозволяють виконувати операції читання та запису, такі як SELECT, INSERT та UPDATE;

- Віддалені джерела (Remote Sources): Це конфігурації, які визначають підключення до віддалених систем через відповідні адаптери;
- Адаптери: Спеціальні компоненти, які забезпечують зв'язок між SAP HANA та віддаленими джерелами даних, використовуючи протоколи, такі як ODBC.

Обмеження та зауваження:

- Обмеження на типи даних: Віртуальні таблиці не підтримують деякі типи даних, такі як BLOB/CLOB. Як обхідний шлях, можна створити представлення (view) на віддаленій таблиці, виключивши стовпці з такими типами даних, і потім створити віртуальну таблицю на основі цього представлення;
- Обмеження на операції: Деякі операції, такі як INSERT, UPDATE та DELETE, можуть бути обмежені або не підтримуватися для певних віддалених джерел даних;
- Використання в кластерних середовищах: Віртуальні таблиці не підтримуються в багатовузлових (multi-node) кластерах SAP HANA.

16.2. Візуалізація даних

16.2.1. Означення та цілі візуалізації даних

Візуалізація даних — це процес представлення числових або текстових даних у графічному вигляді. Це можуть бути діаграми, графіки, карти, теплокарти, дашборди та інші форми, що дозволяють швидко й ефективно зрозуміти великі обсяги інформації [43, 44].

Візуалізація даних відкриває приховані тенденції, показує найменші деталі в бізнес-процесах мережі, допомагає отримати важливі інсайти для менеджерів та керівників.

Основні цілі:

- Спрощення сприйняття інформації;
- Виявлення закономірностей, трендів, аномалій;
- Підтримка прийняття рішень;
- Комунікація результатів аналізу.

16.2.2. Популярні інструменти візуалізації даних

16.2.2.1. Tableau

Tableau — потужний інструмент для візуалізації даних, який часто використовують у бізнес-аналітиці, фінансовій звітності, маркетингу та ін.

Ключові можливості Tableau:

1. Інтерактивні дашборди;
2. «Drag & drop» побудова графіків;
3. Підключення до різних джерел даних ($Excel$, SQL, Google Sheets тощо);
4. Фільтри, підсвічування, динамічні поля;
5. Автоматичне оновлення звітів;
6. Можливість публікації на Tableau Server або Tableau Public.

Інтерактивні дашборди

Інтерактивний дашборд — це візуальний інтерфейс, який дозволяє користувачам взаємодіяти з даними в режимі реального часу. Він використовується для аналізу, моніторингу та прийняття рішень на основі динамічної інформації (рис. 16.7).

Рис. 16.7. Приклад інтерактивного дашборду в стилі Tableau

Основні особливості:

- *Фільтри та вибірки* — можна змінювати параметри перегляду (дата, регіон, категорія тощо).
- *Динамічні графіки* — графіки оновлюються в залежності від вибраних параметрів.
- *Інтеграція з джерелами даних* — дашборд підключається до баз даних, Google Sheets, *API* тощо.
- *Оновлення в реальному часі* — наприклад, відображення даних про продажі або трафік на сайт за останню годину.

«Drag & drop» побудова графіків

«Drag & drop» побудова графіків – це підхід, що дозволяє користувачам візуалізувати дані без необхідності писати код: просто перетягують поля у відповідні області для створення графіків, діаграм, карт тощо. Це значно спрощує аналітику для користувачів без технічного бекграунду (рис. 16.8).

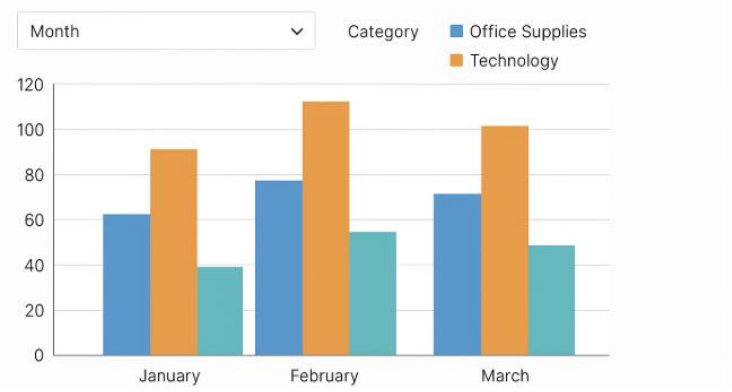


Рис. 16.8. Приклад побудови графіків за допомогою функції «Drag & Drop»

Фільтри, підсвічування, динамічні поля

Фільтри дозволяють користувачам вибирати конкретні дані для перегляду:

- Обрати період: день, тиждень, місяць;
- Обрати категорію: продукт, регіон, канал;
- Обрати статус: активний, завершений.

Наприклад, обравши «Львів» та «Березень», можна побачити лише дані для цього міста й місяця.

Підсвічування (*Highlighting*) автоматично виділяє пов'язані значення на інших елементах дашборду:

- Навівши курсором на стовпчик «Продукт *A*» — виділяються всі пов'язані показники;
- Натискаючи на елемент, можна «засвітити» інші діаграми;
- Це зручно, коли ти хочеш простежити зв'язки між даними.

Динамічні поля (*Calculated Fields* / *Dynamic KPIs*) – це поля, які перераховуються в реальному часі залежно від вибору фільтра:

- Відсоток зростання;
- Середній чек;
- *ROI* для вибраного сегмента.

Наприклад, якщо в дашборді «Аналіз продажів» (рис. 16.9) обрати валюту, прибуток та регіон, дашборд перераховує середній дохід лише для цієї групи.

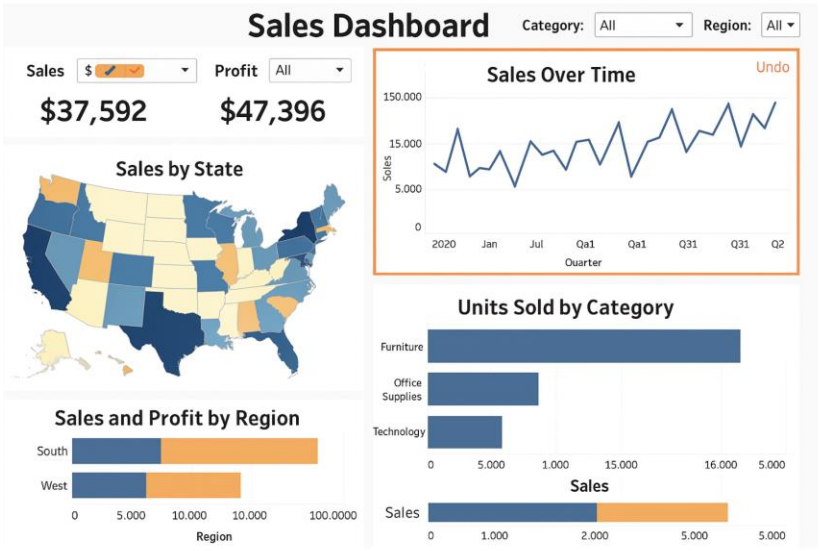


Рис. 16.9. Приклад дашборду з фільтрами, підсвічуванням і динамічними полями

Елементи на рис. 16.9:

- Діаграми продажів по регіонах;
- Фільтри за роками, товарами, категоріями;
- Динамічні показники доходу, прибутку і кількості продажів;
- Підсвічування при виборі фільтра;

- Інтерактивна навігація між сторінками.

Можливість публікації на Tableau Desktop, Tableau Server або Tableau Public

Tableau Desktop — це професійний інструмент для створення інтерактивних візуалізацій, звітів і аналітичних дашбордів (рис. 16.10).

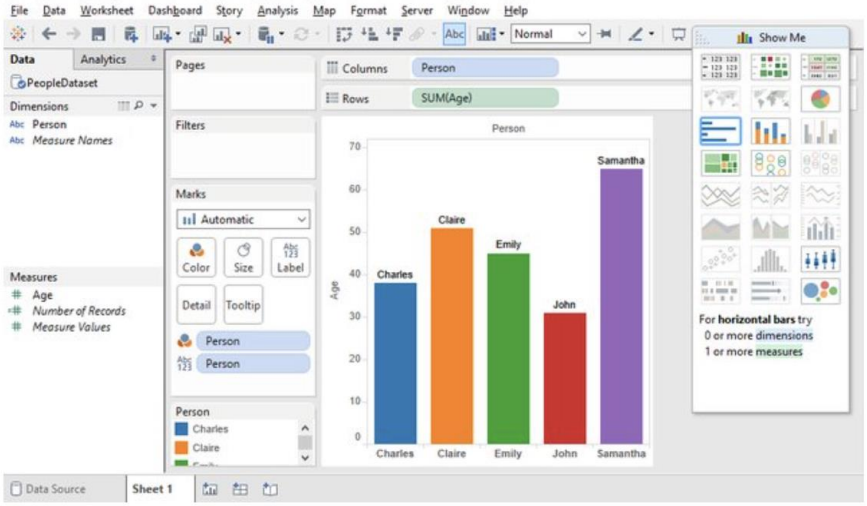


Рис. 16.10. Приклад реального дашборду, створеного в Tableau Desktop

Основні можливості Tableau Desktop:

- *Підключення до даних.* Імпортує дані з різних джерел — баз даних, Excel, CSV, хмарних сервісів тощо.
- *Побудова візуалізацій.* Швидке створення графіків, діаграм, теплокарт, картографічних карт через простий «drag & drop» інтерфейс.
- *Аналіз даних.* Використання фільтрів, агрегатів, обчислень і статистичних моделей.
- *Інтерактивність.* Створення інтерактивних дашбордів із можливістю фільтрації, підсвічування, взаємодії між елементами.
- *Підготовка даних.* Очищення, об'єднання, перетворення даних без потреби у коді.
- *Публікація.* Можливість зберігати проєкти локально або публікувати на Tableau Server чи Tableau Cloud.

Tableau Server — це корпоративна платформа для спільного використання, керування і безпечної публікації дашбордів і аналітичних звітів, створених у Tableau Desktop (рис. 16.11).

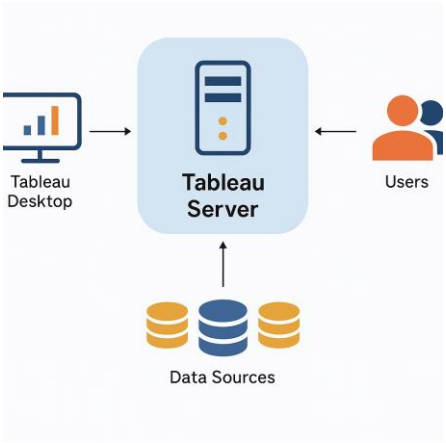


Рис. 16.12. Структурна схема, як працює Tableau Server

Ключові характеристики Tableau Server:

- *Спільний доступ.* Надає змогу користувачам переглядати, редагувати і взаємодіяти з візуалізаціями через браузер або мобільний додаток.
- *Безпека.* Контроль доступу на основі ролей, інтеграція з корпоративними системами автентифікації (Active Directory, LDAP).
- *Автоматизація.* Планування оновлення даних, оповіщення про зміни у звітах.
- *Масштабованість.* Аідтримка великої кількості користувачів та обробки великих обсягів даних.
- *Інтеграція.* Підключення до різних джерел даних у реальному часі або через екстракти.
- *Адміністрування.* Моніторинг продуктивності, керування ліцензіями, аудит активності користувачів.

Tableau Desktop — це місце, де створюють звіти, а Tableau Server — де їх ділять на всю команду

Tableau Public — це безкоштовна версія Tableau, яка дозволяє створювати інтерактивні візуалізації та дашборди й публікувати їх онлайн для широкої аудиторії (рис. 16.12).

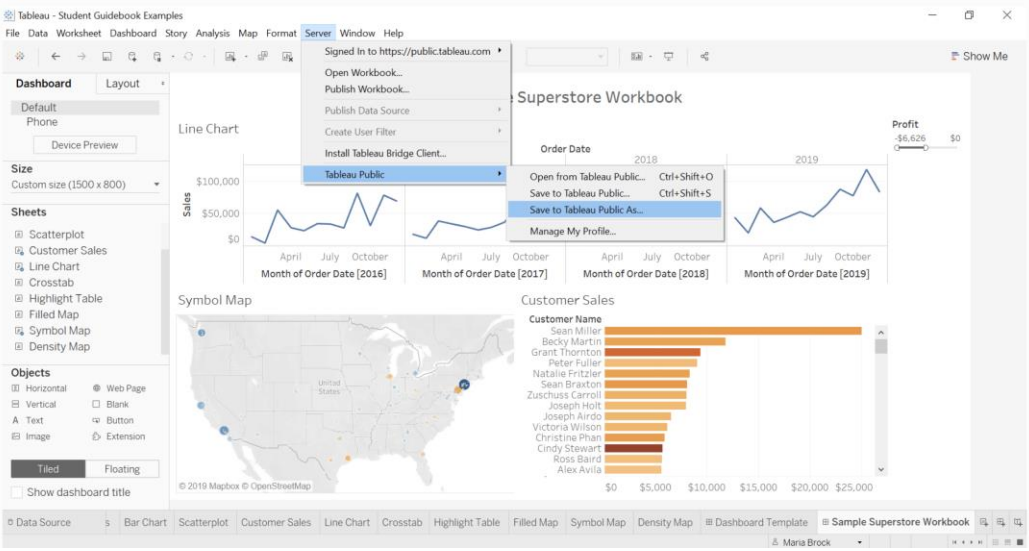


Рис. 16.12. Приклад дашборду, створеного в Tableau Public

Ключові особливості Tableau Public:

- Безкоштовне використання для всіх користувачів;
- Усі опубліковані дані та візуалізації доступні у відкритому доступі;
- Простий інтерфейс «drag-and-drop» для створення графіків та дашбордів;
- Можливість переглядати роботи інших аналітиків у спільноті Tableau Public;
- Ідеально підходить для портфоліо аналітиків, викладачів, студентів.

16.2.2.2. Microsoft Power BI

Microsoft Power BI — потужна платформа для візуалізації, аналітики та бізнес-інтелекту (рис. 16.13).

Рис. 16.13. Приклад дашборду Power BI

Microsoft Power BI – це хмарна служба та набір інструментів від Microsoft для:

- Підключення до різноманітних джерел даних;
- Створення інтерактивних звітів і дашбордів;
- Візуалізації ключових показників у режимі реального часу.

Можливості:

- Підключення до SQL, $Excel$, $Azure$, веб-сервісів тощо;
- Перетворення та об'єднання даних (Power Query);
- Візуалізації: графіки, діаграми, KPI-тайли, карти;
- Спільне використання звітів у Microsoft Teams або на порталі;
- Використання DAX для обчислень та аналітики.

16.2.2.3. Google Data Studio

Google Data Studio — це безкоштовний інструмент для створення інтерактивних дашбордів та звітів, який дозволяє користувачам візуалізувати дані з різних джерел, таких як Google Analytics, Google Sheets, BigQuery, SQL-бази даних, а також інші сторонні джерела (рис. 16.14).

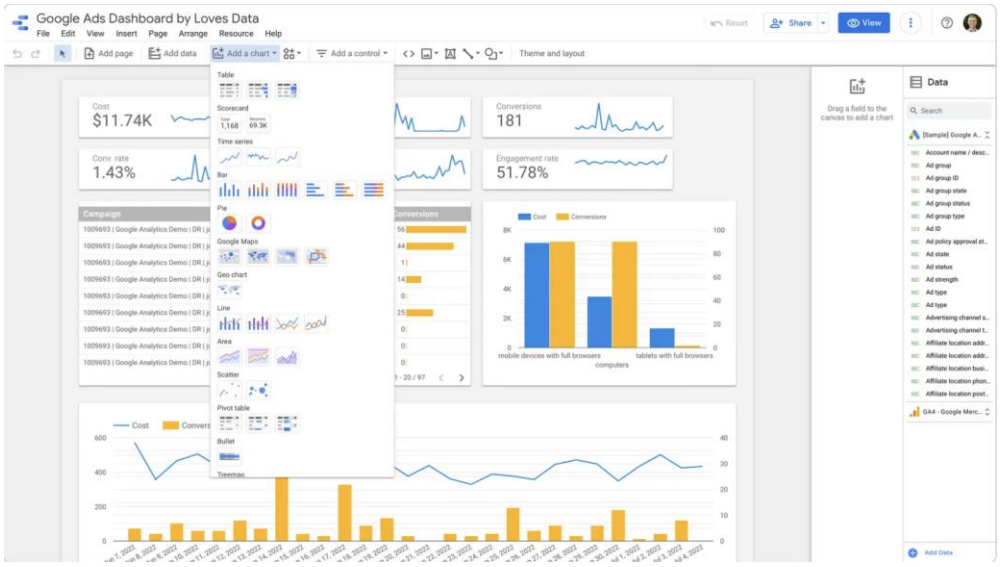


Рис. 16.14. Приклад дашборду, створеного в Google Data Studio

Приклади дашбордів у Google Data Studio:

- *Marketing Dashboard.* Цей дашборд дозволяє відстежувати ефективність маркетингових кампаній, використовуючи метрики з Google Analytics, Google Ads і соціальних мереж. Візуалізації можуть включати трафік на сайті, кількість кліків, конверсії та інші важливі показники;
- *SEO Performance Dashboard.* Використовується для відстеження SEO-метрик, таких як органічний трафік, позиції в пошукових системах, кількість зворотних посилань, і показники взаємодії з контентом. Звіт може включати графіки про зміну позицій, порівняння результатів за різні періоди часу та багато іншого;
- *Sales Dashboard.* Дашборд для відслідковування обсягів продажу, дохідності, маржинальності і продуктивності продавців. Може включати графіки та таблиці, що відображають порівняння результатів за різні періоди, з різними сегментами продуктів чи категорій;
- *Customer Insights Dashboard.* Використовується для аналізу клієнтських даних: вік, географія, поведінка на сайті, зацікавленість у продуктах та інші демографічні або поведінкові показники. Візуалізації можуть включати діаграми про вподобання клієнтів та їхню активність.

Як створити дашборд в Google Data Studio:

- Перейти на сайт Google Data Studio;
- Створити новий звіт або виберіть шаблон, який найбільше підходить для поставлених цілей;
- Підключити джерела даних (Google Analytics, Google Sheets, SQL, тощо);
- Додати різні візуалізації, такі як графіки, таблиці, карти, та налаштуйте їх відповідно до ваших потреб;
- Зберегти звіт і публікуйте або діліться ним з іншими користувачами.

16.2.2.4. Візуалізація даних в Python

Matplotlib

Matplotlib — це основна бібліотека для створення графіків та діаграм у Python. Вона дуже гнучка та дозволяє створювати широкий спектр візуалізацій, таких як лінійні графіки, гістограми, кругові діаграми, графіки розсіяння тощо.

```
import matplotlib.pyplot as plt

# Дані для графіку
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Створення графіку
plt.plot(x, y)
plt.title('Лінійний графік')
plt.xlabel('X')
plt.ylabel('Y')

# Показ графіку
plt.show()
```

Seaborn

Seaborn — це бібліотека для статистичної візуалізації, яка будується на основі Matplotlib. Вона дозволяє створювати складні візуалізації з мінімальними налаштуваннями.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Завантаження вбудованих даних
tips = sns.load_dataset("tips")

# Створення графіку розсіяння
sns.scatterplot(x="total_bill", y="tip", data=tips)

# Показ графіку
plt.show()
```

Plotly

Plotly — це бібліотека для інтерактивної візуалізації. Вона дозволяє створювати інтерактивні графіки, які можна переглядати та маніпулювати, що робить її ідеальною для створення дашбордів.

```
import plotly.express as px
```

```
# Завантаження вбудованих даних
df = px.data.iris()

# Створення інтерактивної діаграми розсіяння
fig = px.scatter(df, x="sepal_width",
y="sepal_length", color="species")
fig.show()
```

Bokeh

Bokeh — це ще одна бібліотека для інтерактивної візуалізації, яка дозволяє створювати динамічні графіки та інтерактивні дашборди.

```
from bokeh.plotting import figure, show

# Створення фігури
p = figure(title="Графік розсіяння", x_axis_label='X',
y_axis_label='Y')

# Дані для графіку
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Додавання точок на графік
p.scatter(x, y)

# Показ графіку
show(p)
```

Altair

Altair — це декларативна бібліотека для візуалізації даних, яка дозволяє створювати прості, але потужні графіки за допомогою опису їх структури.

```
import altair as alt
import pandas as pd

# Створення DataFrame
data = pd.DataFrame({
    'x': [1, 2, 3, 4, 5],
    'y': [2, 3, 5, 7, 11]
})

# Створення графіку
```

```
chart = alt.Chart(data).mark_line().encode(
    x='x',
    y='y'
)

# Показ графіку
chart.show()
```

Pandas Visualization

Pandas також має вбудовані методи для візуалізації даних через Matplotlib, якщо у вас є DataFrame. Це корисно для швидкого побудови графіків без необхідності імпортувати інші бібліотеки.

```
import pandas as pd
import matplotlib.pyplot as plt

# Створення DataFrame
data = pd.DataFrame({
    'x': [1, 2, 3, 4, 5],
    'y': [2, 3, 5, 7, 11]
})

# Побудова графіку
data.plot(x='x', y='y', kind='line')

# Показ графіку
plt.show()
```

Ці бібліотеки та інструменти надають великий вибір варіантів для створення візуалізацій, які допомагають зрозуміти дані та продемонструвати їх іншим (рис. 16.15). Вибір бібліотеки залежить від ваших потреб (статистична візуалізація, інтерактивність, простота тощо).

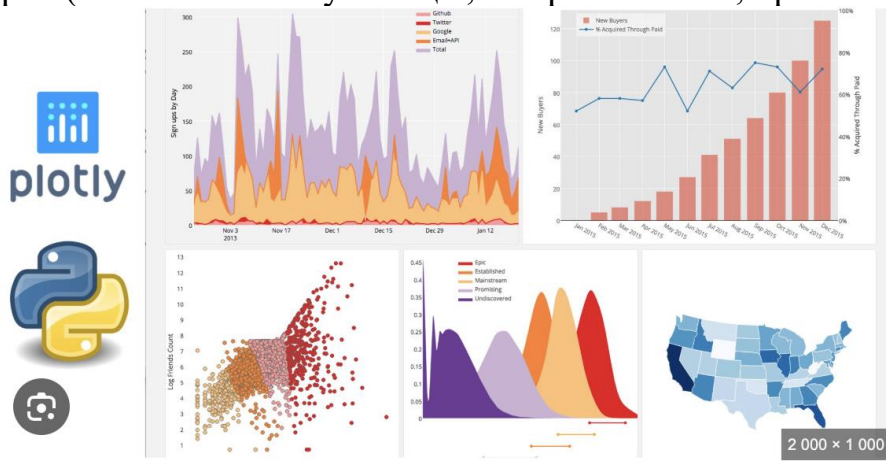


Рис. 16.15. Приклад візуалізації даних в Python

16.2.2.5. Візуалізація даних мовою R

Базова побудова графіка в R

```
# Дані
x <- c(1, 2, 3, 4, 5)
y <- c(2, 3, 5, 7, 11)

# Побудова базового графіка
plot(x, y, type = "o", col = "blue",
     main = "Приклад лінійного графіку",
     xlab = "X", ylab = "Y")
```

В коді *type* = «*o*» означає «*line and points*» (лінії + точки).

Використання *ggplot2* для більш красивої візуалізації

```
# Встановіть бібліотеку ggplot2, якщо ще не
встановлена
# install.packages("ggplot2")

library(ggplot2)

# Дані
data <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(2, 3, 5, 7, 11)
)

# Побудова графіку за допомогою ggplot2
ggplot(data, aes(x = x, y = y)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  ggtitle("Приклад лінійного графіку з ggplot2") +
  xlab("X") +
  ylab("Y")
```

Приклади візуалізації даних мовою R наведено на рис. 16.16 та рис. 16.17.

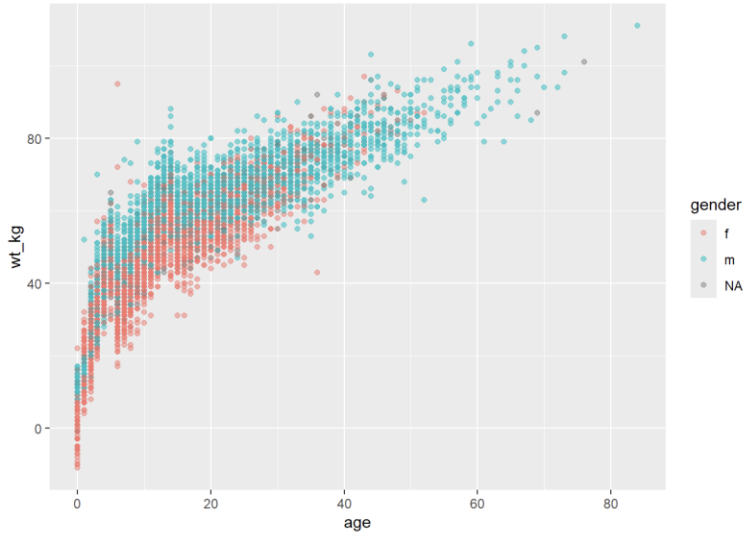


Рис. 16.16. Приклад візуалізації розбиття даних по групам мовою R

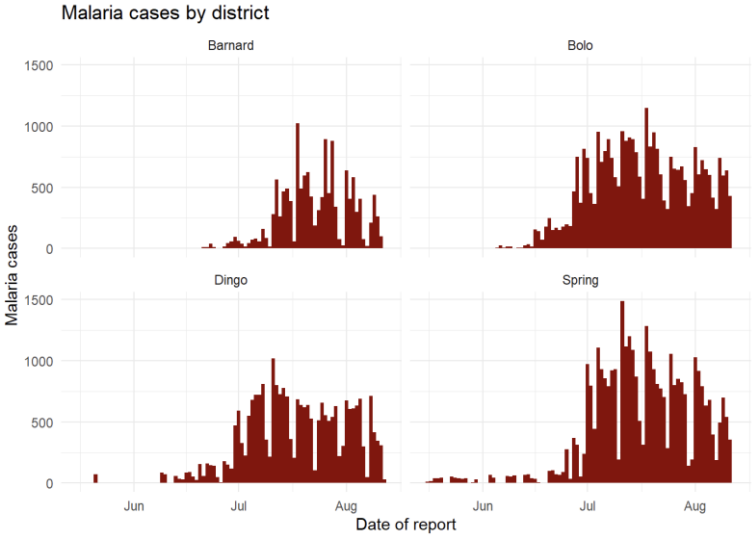


Рис. 16.17. Приклад візуалізації розбиття даних по малим множинам (фісетам) мовою R

16.2.2.5. Порівняння популярних інструментів візуалізації даних

Порівняння популярних інструментів візуалізації даних наведено в табл. 16.4

Таблиця 16.4. Порівняльна таблиця популярних інструментів візуалізації даних

Інструмент	Переваги	Недоліки	Кращий для
Tableau	Інтуїтивний інтерфейс, глибока аналітика, інтерактивні візуалізації	Комерційна ліцензія, може бути складним для початківців	Бізнес-аналітика, дашборди
Power BI	Інтеграція з Microsoft 365,	Менше можливостей	Корпоративна звітність

	доступна ціна, автоматизація	дизайну порівняно з Tableau	
Google Data Studio	Безкоштовний, зручна інтеграція з Google-продуктами	Обмежені типи візуалізацій	Веб-аналітика, маркетинг
Python ($Plotly, Seaborn, \dots$)	Потужна кастомізація, інтеграція з ML	Потребує програмування	Науковий аналіз, R&D
R ($ggplot2$)	Глибокий статистичний аналіз, якісна графіка	Складність для новачків	Академічна сфера, дослідження

16.3. Порівняння візуалізації та віртуалізації даних

Порівняння віртуалізації та візуалізації даних наведено в табл. 16.5.

Теорема 16.5. Порівняння віртуалізації та візуалізації даних

Параметр	Візуалізація даних	Віртуалізація даних
Мета	Представити дані у зручній графічній формі	Об'єднати дані з різних джерел без переміщення
Фокус	Графіки, діаграми, панелі моніторингу (дашборди)	Створення єдиної точки доступу до даних
Інструменти	Tableau, Power BI, Looker	Denodo, Red Hat Data Virtualization, SAP HANA SDA
Користувачі	Аналітики, менеджери, візуалізатори даних	Інженери даних, архітектори даних
Обробка даних	Працює з уже підготовленими даними	Динамічно підтягує дані з різних джерел
Приклади	Дашборд із продажами	Платформа, що об'єднує базу клієнтів із CRM та ERP
Переваги	Краще розуміння трендів, швидке прийняття рішень	Зменшує дублювання даних, економить ресурси

ЛЕКЦІЯ 17

СТАТИСТИЧНИЙ ТА ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗИ В МАРКЕТИНГОВИХ ДОСЛІДЖЕНННЯХ

17.1. A/B-тестування

17.1.1. Опис A/B-тестувань

A/B-тестування (*A/B testing*, *Split testing*) — метод маркетингового дослідження, суть якого полягає в тому, що контрольна група елементів порівнюється з набором тестових груп, в яких один або декілька показників були змінені, для того, щоб з'ясувати, які зі змін покращують цільовий показник. Хороший приклад — це дослідження впливу колірної схеми, розташування та розміру елементів інтерфейсу на конверсію сайту.

Метод часто використовується при оптимізації вебсторінок відповідно до заданої мети. Тестуються 2 дуже схожі сторінки (сторінка A і сторінка B), які відрізняються лише одним елементом або декількома елементами (тоді метод називають *$A/B/N$ Testing*). Сторінки A і B показуються користувачам почергово в рівних пропорціях, при цьому відвідувачі, як правило, не знають про це. По закінченні певного часу або при досягненні певного статистично значимого числа показів порівнюються числові показники мети і визначається найкращий варіант сторінки. До числа компаній, що використовують цей метод, відносяться Amazon і Zynga.

На рис. 17.1 наведено приклад A/B-тестувань для вебсайту. Відвідувачам випадково демонструють дві версії вебсайту, які відрізняються лише дизайном одного елемента — кнопки. Це дозволяє оцінити відносну ефективність кожної з них.

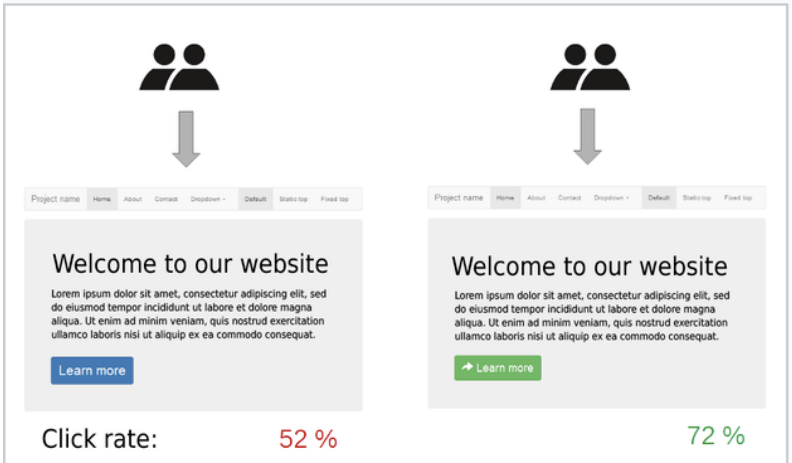


Рис. 17.1. Приклад A/B-тестувань для вебсайту

Суть A/B-тестувань

- Варіант A (контрольна група) — поточна версія (наприклад, старий інтерфейс або кнопка);

- Варіант B (експериментальна група) — нова версія (наприклад, новий дизайн або нове повідомлення);
- Користувачі випадково розподіляються між цими варіантами;
- Збираються метрики: конверсії, кліки, час на сторінці тощо.

Інструменти для A/B-тестувань

Багато інструментів A/B –тестування активно розвиваються. Деякі з них доступні за ліцензією відкритого джерела або безкоштовно:

- Google Analytics Content Experiments (раніше Google Website Optimizer) (на стороні сервера вимагається використання тегів),
- Easy Оптимізатор вебсайтів.

Інші комерційні рішення, як правило, пропонують більш широкий спектр можливостей.

17.1.2. Типові метрики для A/B-тестувань

17.1.2.1. CTR

CTR (*Click-Through Rate*) — це коефіцієнт клікабельності, який показує, як часто користувачі натискають на елемент (наприклад, посилання, рекламу, кнопку), побачивши його. Обчислюється за формулою (17.1):

$$CTR = \frac{m}{n} \cdot 100\%, \qquad (17.1)$$

де:

m – кількість кліків;

n – кількість показів.

На рис. (17.1) наведено приклад візуалізації динаміки CTR для двох рекламних кампаній протягом тижня. Видно, що кампанія A має вищі показники CTR у всі дні, хоча обидві кампанії демонструють тенденцію до зростання.

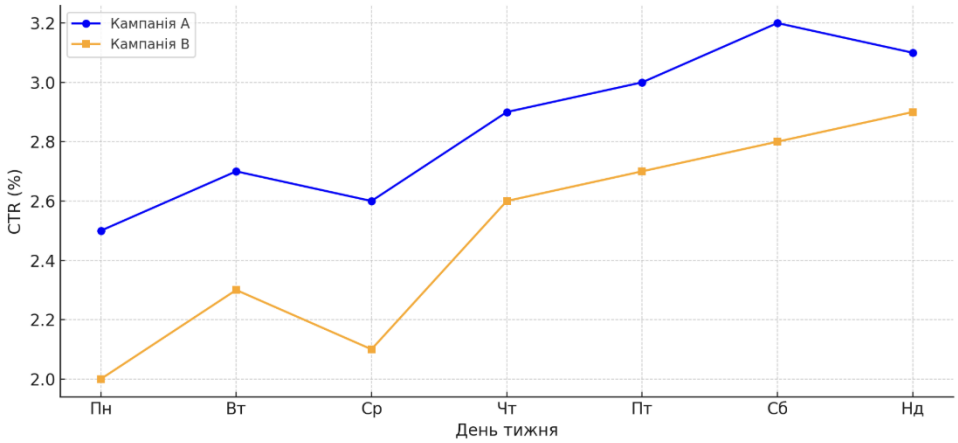


Рис. 17.2. Динаміка CTR протягом тижня

Навіщо вимірювати CTR?

- Для оцінки ефективності реклами;
- Для порівняння A/B варіантів;
- Щоб покращити дизайн або текст оголошення/кнопки;

- Для оптимізації SEO та $e-mail$-маркетингу.

Приклад розрахунку CTR:

- Кількість показів реклами (імпресій): $n = 10000$;
- Кількість кліків: $m = 350$.

За формулою (17.1):

$$CTR = \frac{350}{10000} \cdot 100\% = 3{,}5\%.$$

Приклад обчислення засобами Python:

```
def calculate_ctr(clicks, impressions):
    ctr = (clicks / impressions) * 100
    return round(ctr, 2)

# Приклад
clicks = 350
impressions = 10000

ctr = calculate_ctr(clicks, impressions)
print(f"CTR = {ctr}%")
```

17.1.2.2. CR

CR (*Conversion Rate*) — метрика, яка показує, скільки користувачів виконали цільову дію (наприклад, покупку, підписку, реєстрацію) від загальної кількості відвідувачів. Обчислюється за формулою (17.2):

$$CR = \frac{k}{n} \cdot 100\%, \qquad (17.2)$$

де:

k – кількість конверсій;

n – кількість відвідувань (або кліків).

На рис. 17.3 представлено приклад порівняння ефективності двох сторінок по днях. Цей графік корисний для оцінки результатів A/B тесту.

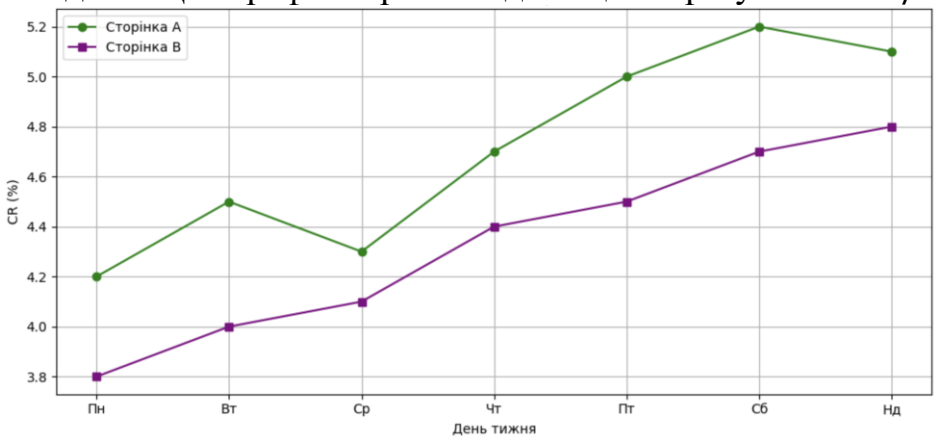


Рис. 17.3. Динаміка CR протягом тижня

На рис. 17.3 можна побачити, що сторінка A має вищий CR майже щодня — вона конвертує краще.

Застосування CR:

- Оцінка ефективності лендінгів;
- Аналіз воронок продажу;
- Розрахунок ROI у маркетингу;
- Основний показник у A/B тестах і CRO (Conversion Rate Optimization).

Приклад розрахунку:

- Кількість кліків: $n = 800$;
- Кількість конверсій (реєстрацій): $k = 120$.

$$CR = \frac{120}{800} \cdot 100\% = 15\%.$$

Приклад обчислення засобами Python:

```
def calculate_cr(conversions, clicks):
    cr = (conversions / clicks) * 100
    return round(cr, 2)

# Приклад
conversions = 120
clicks = 800

cr = calculate_cr(conversions, clicks)
print(f"CR = {cr}%")
```

17.1.2.3. Час взаємодії

Час взаємодії (*Interaction Time*) — це метрика, яка показує, скільки часу користувач витратив на взаємодію з інтерфейсом, елементом сайту чи додатка після його відображення.

На рис. 17.4 наведено приклад гістограми, яка показує розподіл часу взаємодії користувачів. Вона ілюструє, скільки користувачів провели певний час, взаємодіючи з інтерфейсом.

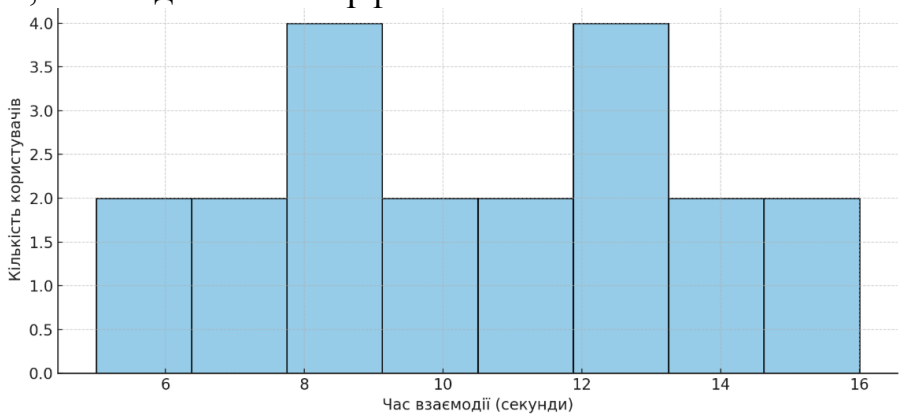


Рис. 17.4. Розподіл часу взаємодії користувачів

Формули для обчислення часу взаємодії

1. *Середній час взаємодії.*

Якщо у нас є набір значень часу взаємодії $t_1, t_2, \ldots, t_n$, то *середній час взаємодії* обчислюється за формулою (17.3):

$$\bar{t} = \frac{1}{n} \cdot \sum_{i=1}^{n} t_i, \quad (17.3)$$

де:

$\bar{t}$ – середнє значення взаємодії;

t_i — час взаємодії i-го користувача;

n — загальна кількість користувачів або сесій.

2. *Максимальний час.*

Максимальний час взаємодії — це метрика, яка визначається співвідношенням (17.4):

$$max(t_1, t_2, \ldots, t_n). \quad (17.4)$$

3. *Мінімальний час.*

Мінімальний час взаємодії — це метрика, яка визначається співвідношенням (17.5):

$$min(t_1, t_2, \ldots, t_n). \quad (17.5)$$

4. *Медіана.*

Медіана – це центральне значення в упорядкованому списку $t_1 \leq t_2 \leq \ldots \leq t_n$.

5. *Дисперсія (варіативність часу).*

Дисперсія часу взаємодії — це метрика, яка визначається за формулою (17.6):

$$\sigma^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (t_i - \bar{t})^2, \quad (17.6)$$

де $\bar{t}$ — середній час взаємодії.

6. *Time to First Interaction (TTFI).*

TTFI – це час від завантаження сторінки до першої дії користувача. Обчислюється за формулою (17.7):

$$TTFI = t_{\text{взаємодії}} - t_{\text{початку завантаження}}, \quad (17.7)$$

де:

$t_{\text{взаємодії}}$— час, коли користувач здійснив першу дію;

$t_{\text{початку завантаження}}$ — момент, коли почалась навігація/завантаження сторінки.

7. *Total Interaction Time (TIT).*

TIT — це метрика, яка вимірює загальний час активної взаємодії користувача з вебсторінкою або додатком. Обчислюється за формулою (17.8):

$$TIT = \sum_{i=1}^{n} \left(t_{\text{Кінець дії}_i} - t_{\text{Початок дії}_i} \right), \quad (17.8)$$

де:

i — індекс активної сесії/взаємодії;

$t_{\text{Кінець дії}_i}; t_{\text{Початок дії}_i}$ — моменти часу між якими користувач щось робив на сторінці під час сесії i.

Основні приклади використання часу взаємодії:

- *UX-дослідження*: оцінка ефективності дизайну (напр., як швидко користувач знаходить потрібний елемент);
- *Аналіз залучення*: скільки часу користувач активно взаємодіє з контентом (прокрутка, кліки, заповнення форм);
- *Оптимізація сайтів та додатків*: виявлення повільних або заплутаних ділянок інтерфейсу.

Приклад обчислення засобами Python:

```
import numpy as np

# Секунди взаємодії 10 користувачів
interaction_times = [5, 12, 9, 15, 8, 14, 7, 6, 10,
13]

avg_time = np.mean(interaction_times)
max_time = np.max(interaction_times)
min_time = np.min(interaction_times)

print(f"Середній час взаємодії: {avg_time:.2f} с")
print(f"Максимальний: {max_time} с, Мінімальний:
{min_time} с")
```

17.1.2.4. *Bounce Rate*

Bounce Rate (*показник відмов*) — це відсоток відвідувачів, які покинули сторінку, не виконавши жодної взаємодії (не натиснули, не перейшли, не прокрутили і т.д.). Обчислюється за формулою (17.9):

$$Bounce\ Rate = \frac{l}{s} \cdot 100\%, \tag{17.9}$$

де:

l – кількість сеансів без взаємодії;

s – загальна кількість сеансів.

На рис. 17.5 наведено приклад розподілу $Bounce\ Rate$ по сторінкам сайту. На даній гістограмі можна побачити, які сторінки мають найвищий рівень відмов. Їх можна використати для оптимізації UX, CTA, контенту.

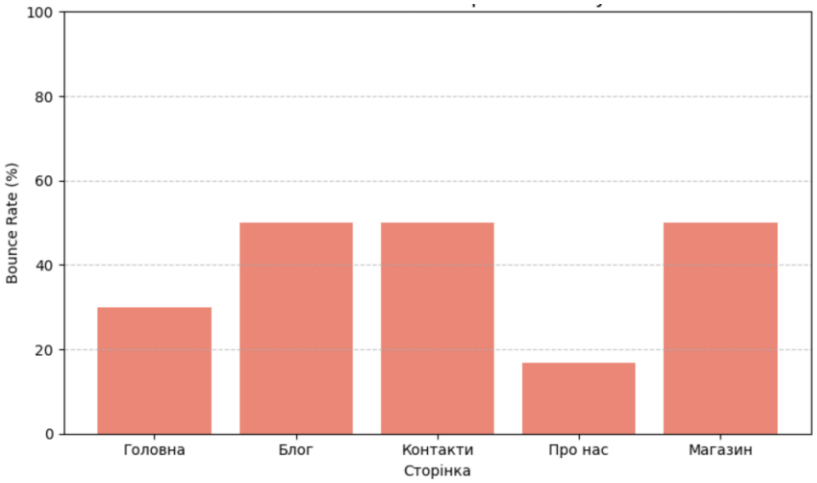


Рис. 17.5. Розподіл $Bounce\ Rate$ по сторінкам сайту

Наприклад:

- $s = 100$ – загальна кількість користувачів, що зайшли на сайт;
- $l = 60$ – кількість користувачів, що одразу вийшли.

$$Bounce\ Rate = \frac{60}{100} \cdot 100\% = 60\%.$$

Як інтерпретувати?

Шкалу для інтерпретації значень $Bounce\ Rate$ наведено в табл. 17.1.

Таблиця 17.1. Інтерпретація значень $Bounce\ Rate$

$Bounce\ Rate$	Оцінка
$< 20\%$	Дуже добре (можливо — помилка у трекінгу)
$20–40\%$	Норма для контенту з цільовою взаємодією
$40–60\%$	Середній показник
$> 60\%$	Можливо, проблема з контентом або UX

Що означає високий Bounce Rate?

- Проблеми з UX або швидкістю завантаження;
- Контент не відповідає очікуванням;
- Немає заклику до дії (CTA);
- Поганий дизайн або мобільна адаптація.

Коли високий $Bounce\ Rate$ — це нормально:

- Користувач шукав конкретну відповідь і знайшов її на першій сторінці;
- Односторінкові сайти (landing page);
- Контактна інформація була ціллю.

Як зменшити $Bounce\ Rate$:

- Оптимізувати швидкість сайту;
- Зробити структуру та CTA більш помітними;
- Поліпшити заголовки та метаопис;
- Адаптувати під мобільні пристрої.

Розрахунок *Bounce Rate* (Python):

```
import pandas as pd
import matplotlib.pyplot as plt

# Імітаційні дані по сторінках сайту
data = {
    "Сторінка": ["Головна", "Блог", "Контакти", "Про нас", "Магазин"],
    "Сеансів": [1000, 800, 400, 600, 1200],
    "Сеансів_без_взаємодії": [300, 400, 200, 100, 600]
}

df = pd.DataFrame(data)

# Додаємо стовпець Bounce Rate
df["Bounce Rate (%)"] = (df["Сеансів_без_взаємодії"] / df["Сеансів"]) * 100

print(df[["Сторінка", "Bounce Rate (%)"]])
```

17.1.2.5. *ARPU*

ARPU (*Average Revenue Per User*) — це середній дохід з одного користувача за певний період (місяць, квартал, рік). Обчислюється за формулою (17.10):

$$ARPU = \frac{q}{p}, \tag{17.10}$$

де:

p – кількість активних користувачів;

q – загальний дохід.

Приклад візуалізації *ARPU* наведено на рис. 17.6.

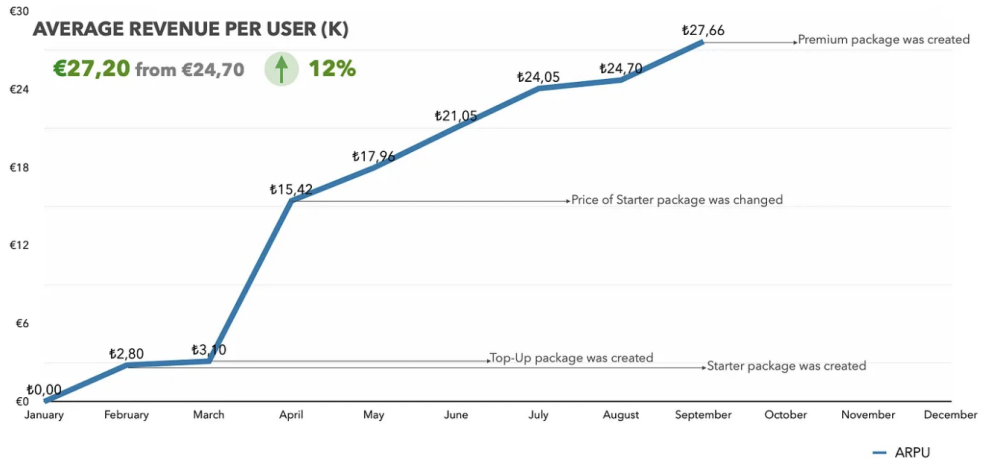


Рис. 17.6. Динаміка *ARPU* протягом року

Що показує $ARPU$:

- Наскільки ефективно монетизується користувач;
- Чи окупається вартість залучення одного користувача (CPA);
- Чи зростає прибутковість бізнесу;
- Важлива метрика для бізнесу, мобільних застосунків, SaaS-сервісів, онлайн-ігор тощо.

Приклад:

- Загальний дохід за місяць: $q = 10000$ грн;
- Кількість активних користувачів: $p = 500$.

$$ARPU = \frac{10000}{500} = 20 \text{ (грн/користувач)}.$$

Приклад розрахунку в Python:

```
# Дані
total_revenue = 10000  # грн
active_users = 500     # користувачів

# Обчислення ARPU
arpu = total_revenue / active_users
print(f"ARPU: {arpu:.2f} грн/користувач")
```

17.1.3. Інтерпритація результатів A/B-тестів

Для інтерпритації результатів A/B-тестів необхідно використати параметр $p - value$, з яким ми вже стикалися у лекції 9 в тесті Діккі-Фулера. Зупинемось на цьому параметрі більш детально в контексті A/B-тестів.

17.1.3.1. Параметр $p - value$

$p - value$ – це статистична міра, яка допомагає визначити значущість результатів при перевірці гіпотез. Воно кількісно виражає ймовірність того, що отримані результати будуть принаймні настільки ж екстремальними, як і отримані, якщо припустити, що нульова гіпотеза є вірною.

17.1.3.1.1. Формулювання гіпотез

Для розрахунку $p - value$ спочатку необхідно сформулювати наступні статистичні гіпотези:

Нульова гіпотеза (H_0): Це твердження про відсутність ефекту або різниці між вибірками з генеральних сукупностей. При перевірці гіпотез, це припущення полягає у тому, що спостережувані дані є випадковими.

Альтернативна гіпотеза (H_1): Це гіпотеза про те, що існує ефект або значуща різниця між спостережуваними даними.

17.1.3.1.2. Обчислення $p - value$

З вищесказаного випливає, що $p - value$ визначається як ймовірність, за умови нульової гіпотези H_0, отримати результат рівний або більш екстремальний ніж той, що фактично спостерігався.

В залежності від того як це розглядати, «більш екстремальний ніж той, що фактично спостерігався», може означати $\{X \geq K_{\text{сп}}\}$ (подія із правого хвоста) або $\{X \leq K_{\text{сп}}\}$ (подія із лівого хвоста) або «менший» із $\{X \leq K_{\text{сп}}\}$ та $\{X \geq K_{\text{сп}}\}$ (подія із обох хвостів) ($K_{\text{сп}}$ – значення відповідної статистики).

Таким чином, $p - value$ визначається умовні як ймовірності (17.11)-(17.13):

$P(X \geq K_{\text{сп}}|H)$ для випадку події із правого хвоста; (17.11)

$P(X \leq K_{\text{сп}}|H)$ для випадку події із лівого хвоста; (17.12)

$2 \cdot min\{P(X \leq K_{\text{сп}}|H), P(X \geq K_{\text{сп}}|H)\}$ для обох хвостів. (17.13)

Приклад розрахунку $p - value$ наведено на рис. 17.7. Вертикальній координатній шкалі відповідає густині ймовірності кожного результату, розрахованого відповідно до нульової гіпотези. $p - value$ це площа під кривою, що знаходиться за точкою даних спостереження.

Рис. 17.7. Приклад розрахунку $p - value$

Таким чином, $p - value$ обчислюється на основі деякої статистики тесту (вона вибирається залежно від потрібного статистичного критерію: критерій Стьюдента, z-критерій, тощо). Для цього порівнюється отримана статистика тесту з розподілом відповідної статистики (наприклад, нормальний розподіл для z-тесту, t-розподіл для t-тесту чи $\mathcal{X}^2$-розподіл для $\mathcal{X}^2$-тесту). Існують таблиці критичних значень для цих розподілів.

Слід зазначити, що для A/B-тестів при обчислені $p - value$ використовуються z-тест для пропорцій та t-тест.

Z-тест для пропорцій

Z-тест — це статистичний метод, який використовується для перевірки гіпотез про середні значення або пропорції, коли:

- вибірка досить велика ($n > 30$);
- відоме стандартне відхилення генеральної сукупності (σ);
- застосовується до долучених пропорцій (наприклад, у A/B тестах).

Пропорція — це частка або відсоток успішних випадків у певній вибірці.

В загальному випадку *пропорція* – це кількість об’єктів з певною ознакою.

Дане поняття є більш широким поняттям, ніж поняття конверсія (17.3).

Тобто, *конверсія* — специфічна вибіркова пропорція, яка описує частоту досягнення цілей.

В z-тесті для пропорцій використовується стандартний нормальний розподіл $N(0,1)$. Це випливає з центральної граничної теореми (ЦГТ): при досить великому об'ємі вибірки, розподіл вибіркових пропорцій (часток) наближається до нормального, навіть якщо початковий розподіл біноміальний (успіх/неуспіх).

Статистика z для двох пропорцій обчислюється за формулою (17.14)

$$K_{\text{сп}} = z_{\text{сп}} = \frac{p_1 - p_2}{\sqrt{\hat{p}\cdot(1-\hat{p})\cdot\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}, \quad (17.14)$$

де:

p_1 і p_2 — вибіркові пропорції;

n_1, n_2 — розміри вибірок;

$\hat{p}$ — зведена (pooled) пропорція, яка обчислюється за формулою (17.15):

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}, \quad (17.15)$$

де:

n_1, n_2 — розміри вибірок;

x_1 — кількість успішних подій (наприклад, конверсій) у групі A;

x_2 — кількість успішних подій у групі B.

Після обчислення z-статистики, ми шукаємо $p - value$ — площу під кривою нормального розподілу (рис. 17.7):

Для *двостороннього тесту* справедлива формула (17.16):

$$p - value = 2 \cdot P(Z > | z_{\text{сп}} |). \quad (17.16)$$

Для *одностороннього тесту* справедлива формула (17.17):

$$p - value = P(Z > z_{\text{сп}}) \text{ або } p - value = P(Z < z_{\text{сп}}). \quad (17.17)$$

Приклад обчислення в Python.

Це можна зробити в Python за допомогою $scipy.stats.norm.cdf()$ або $proportions\_ztest$, який уже враховує це автоматично.

```
from statsmodels.stats.proportion import
proportions_ztest

conversions = [80, 100]
visitors = [1000, 1000]

z_stat, p_value =
proportions_ztest(count=conversions, nobs=visitors)

print(f"Z-статистика: {z_stat:.4f}") # z_stat=z_сп
print(f"P-value: {p_value:.4f}")
```

T-тест для середніх значень

T-тест — це статистичний метод, який використовується для незалежних вибірок використовується при порівнянні середніх значень двох груп в A/B-тесті.

У випадку t-тесту в якості $K_{\text{сп}}$ у формулах (17.16)-(17.17) використовується $t_{\text{сп}}$, що у випадку відомих вибіркових середніх та виправлених дисперсій у групах A і B обчислюється за формулою (17.18):

$$K_{\text{сп}} = t_{\text{сп}} = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \quad (17.18)$$

де:

$\overline{x_1}, \overline{x_2}$ – середні значення у групах A і B;

s_1^2, s_2^2 – вибіркові виправлені дисперсії у групах A і B;

n_1, n_2 – розмір вибірок у кожній групі;

$t_{\text{сп}}$ – t-статистика, яку порівнюють із t-розподілом (розподілом Стьюдента).

Для незалежних вибірок з відомими коефіцієнтами конверсії для обчислення $t_{\text{сп}}$ також справедлива формула (17.19), що є наслідком формули (17.14), якщо замість вибіркових пропорцій взяти значення коефіцієнтів конверсії для кожної групи:

$$K_{\text{сп}} = t_{\text{сп}} = \frac{CR(A) - CR(B)}{\sqrt{\widehat{CR} \cdot (1 - \widehat{CR}) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}, \quad (17.19)$$

де:

$CR(A), CR(B)$ – коефіцієнти конверсії для груп A і B відповідно;

$\widehat{CR}$ – комбінований коефіцієнт конверсії (середній коефіцієнт для обох груп), який обчислюється за формулою (17.20):

$$\widehat{CR} = \frac{CR(A) + CR(B)}{2}; \quad (17.20)$$

де:

n_1, n_2 – розмір вибірок у кожній групі;

$t_{\text{сп}}$ – t-статистика, яку порівнюють із t-розподілом (розподілом Стьюдента).

Приклад обчислення в Python.

```
from scipy.stats import ttest_ind
import numpy as np

group_a = np.random.normal(loc=5.0, scale=1.0,
size=100)
group_b = np.random.normal(loc=5.3, scale=1.0,
size=100)

t_stat, p_value = ttest_ind(group_a, group_b)
```

```
print(f"T-статистика: {t_stat:.4f}") # t_stat=t_сп
print(f"P-value: {p_value:.4f}")
```

Інтерпретація $p-value$

Якщо $p-value$ невелике (зазвичай менше $0{,}05$), це означає, що спостережувані дані не узгоджуються з нульовою гіпотезою, що призводить до її відхилення. Це означає, що результати є статистично значущими.

Більше значення $p-value$ свідчить про те, що докази проти нульової гіпотези є слабкими, і її не слід відкидати.

17.1.3.2. Умови успішного проходження A/B-тестів

1. Коректна постановка гіпотези:
- Нульова гіпотеза (H_0): між групами немає різниці;
- Альтернативна гіпотеза (H_1): є статистично значуща різниця.

Приклад: «Зміна дизайну підвищить конверсію на сайті».

2. Рандомізація:
- Користувачі мають випадковим чином потрапляти в групу A або групу B, щоб уникнути упередженості.

3. Велика та збалансована вибірка:
- У кожній групі має бути достатньо користувачів;
- Розмір вибірки можна розрахувати заздалегідь (потрібно знати очікувану різницю, варіацію, бажаний рівень значущості та силу тесту).

4. Вимірювання однієї метрики:
- Необхідно чітко визнач, яка метрика тестується (наприклад, конверсія, $ARPU$, час на сайті);
- Не слід міряти «все підряд» — інакше зростає ризик хибнопозитивних результатів (p-hacking).

P-hacking — це практика маніпулювання даними або аналізом, щоб отримати статистично значущий результат ($p-value<0{,}05$), навіть якщо насправді ефекту немає.

5. Достатній час тестування:
- Тест має тривати достатньо довго, щоби охопити весь цикл поведінки користувачів (наприклад, усі дні тижня);
- Але не надто довго, щоб не виник вплив сезонності або змін зовнішніх факторів.

6. Незалежність груп:
- Один і той самий користувач не повинен потрапити в обидві групи;
- Дані мають бути незалежними, інакше статистика буде хибною.

7. Фіксований період та кількість користувачів:
- Не можна підглядати в $p-value$ щодня й зупиняти тест, коли «щось гарне» з'явилося;
- Слід використовувати заздалегідь визначений час або розмір вибірки.

8. Контроль зовнішніх факторів:

- Не слід змінювати інші елементи сайту/продукту під час тесту;
- Якщо йдуть інші паралельні експерименти — вони можуть заважати.

9. Перевірка статистичних припущень:
- Якщо метрика — пропорція → z-тест;
- Якщо метрика — середнє → t-тест.

Але перед цим необхідно перевірити:
- Нормальність розподілу (для t-тесту);
- Гомогенність дисперсій, тобто їх рівність (опціонально).

10. Коректна інтерпретація результату:
- $p - value < 0{,}05$ → статистично значуща різниця;
- Але слід пам'ятати, що статистична значущість не завжди означає практичну корисність!

17.1.4. Приклад A/B тестування

Умова задачі

Компанія проводить A/B тестування кнопки «Купити зараз» на своєму вебсайті, щоб визначити, який варіант кнопки має кращу конверсію (тобто більший відсоток користувачів, які натискають на кнопку).

Варіант A (синя кнопка):
- Кількість користувачів, які побачили кнопку: 1000;
- Кількість кліків на кнопку: 50.

Варіант B (червона кнопка):
- Кількість користувачів, які побачили кнопку: 1000
- Кількість кліків на кнопку: 75.

Розрахунок коефіцієнта конверсії

Коефіцієнт конверсії обчислюється за формулою (17.2).

Для варіанту A:

$$CR(A) = \frac{50}{1000} \cdot 100\% = 5\%.$$

Для варіанту B:

$$CR(B) = \frac{75}{1000} \cdot 100\% = 7{,}5\%.$$

Порівняння коефіцієнтів конверсії

Варіант A: 5%;

Варіант B: 7,5%.

Варіант B має вищий коефіцієнт конверсії, що свідчить про те, що червона кнопка привертає більше уваги користувачів і спонукає їх натискати на неї частіше.

Статистичний аналіз (t-тест)

Для перевірки статистичної значущості різниці в коефіцієнтах конверсії між варіантами A та B, можемо застосувати t-тест. Тобто, для обчислення $p - value$ виконаємо наступні кроки:

Крок 1. Обчислення t-статистики. За формулою (17.19) обчислюємо t-статистику, враховуючи, що $CR(A) = 0{,}05$; $CR(B) = 0{,}075$; $\widehat{CR} = \frac{0{,}05+0{,}075}{2} = 0{,}0625$:

$$t_{\text{сп}} = \frac{0{,}05 - 0{,}075}{0{,}0108} = -\frac{0{,}025}{0{,}0108} \approx -2{,}31$$

Крок 2. Розрахунок $p-value$. Для того, щоб знайти $p-value$, ми використовуємо таблицю розподілу t-статистики або програмне забезпечення для статистичних розрахунків. Зважаючи на те, що $t_{\text{сп}} \approx -2{,}31$, ми можемо знайти $p-value$ за формулою (17.16) за допомогою таблиці t-розподілу при рівні значущості $0{,}05$ для великих об'ємів вибірок.

$$p-value = 2 \cdot P(Z > |-2{,}31|) = 2 \cdot P(Z > 2{,}31) =$$
$$= 2 \cdot \big(F(+\infty) - F(2{,}31)\big) = 2 \cdot (1 - 0{,}9896) \approx 0{,}0208.$$

Крок 3. Інтерпретація $p-value$. Аналіз результатів відбуватиметься за наступними правилами:

- Якщо $p-value < 0{,}05$, то різниця між варіантами статистично значуща;
- Якщо $p-value > 0{,}05$, то ми не можемо зробити висновок про статистичну значущість різниці.

У нашому випадку $p-value \approx 0{,}02$, що менше за $0{,}05$, тому різниця між варіантами A та B є статистично значущою.

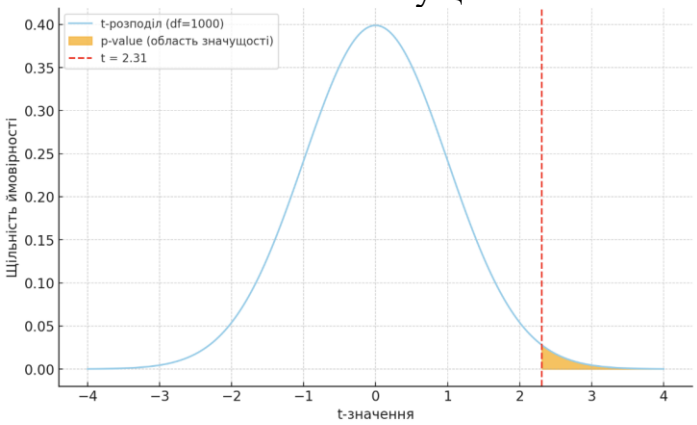


Рис. 17.8. Візуалізація рівня значущості для t-розподілу в прикладі

Висновок

Оскільки варіант B показує кращу конверсію (7,5% порівняно з 5%), і за умови, що t-тест показує статистично значущу різницю, компанія може обрати варіант B як більш ефективний.

17.1.5. Приклад реалізації A/B тесту в Python

```
import scipy.stats as stats

# Дані: кількість успіхів (наприклад, конверсій) і
загальна кількість відвідувань
conversions_A = 200
```

```
visits_A = 2000
conversions_B = 260
visits_B = 2100

# Обчислення пропорцій
p_A = conversions_A / visits_A
p_B = conversions_B / visits_B

# Стандартна похибка
SE = ((p_A * (1 - p_A)) / visits_A + (p_B * (1 -
p_B)) / visits_B) ** 0.5

# z-статистика
z = (p_B - p_A) / SE
p_value = 1 - stats.norm.cdf(z)

print(f"Z-статистика: {z:.2f}")
print(f"P-значення: {p_value:.4f}")
```

17.2. *UX*-дослідження

17.2.1. Опис *UX*-дослідження

UX-дослідження (*User Experience Research*) — це вивчення взаємодії користувачів з продуктом з метою виявлення болей, потреб і шляхів покращення інтерфейсу.

Основні етапи *UX*-дослідження

UX-дослідження складається з наступних етапів, що наведено в табл. 17.2.

Таблиця 17.2. Етапи *UX*-дослідження

№	Етап	Опис
1	Визначення цілей	Що саме потрібно дізнатись? Яку гіпотезу перевіряємо?
2	Вибір аудиторії	Цільові користувачі: хто вони, чим користуються, які в них потреби?
3	Збір даних	Методи: спостереження, інтерв'ю, опитування, аналітика, тести
4	Аналіз поведінки	Що користувачі роблять? Де зупиняються? Що ігнорують?
5	Виведення інсайтів	Які проблеми, що треба покращити, що працює добре
6	Тестування гіпотез	*A/B* тести, прототипування, перевірка рішень
7	Ітерація	Внесення змін, перевірка, нові цикли досліджень

Основні методи UX-дослідження

Найбільш розповсюджені методи UX-дослідження наведено в табл. 17.3.

Таблиця 17.3. Методи UX-дослідження

№	Метод	Тип	Приклад застосування
1	Спостереження	Кваліативний	Як користувачі проходять реєстрацію
2	Інтерв'ю	Кваліативний	Глибинне розуміння мотивацій користувачів
3	Опитування	Кількісний	Збір думок від великої кількості людей
4	Юзабіліті-тестування	Обидва	Спостереження за тим, як користувачі виконують завдання
5	Веб-аналітика (GA)	Кількісний	Виявлення слабких місць у воронці продажів
6	A/B тестування	Кількісний	Порівняння двох варіантів інтерфейсу
7	Карта кліків/теплокарта	Кількісний	Де саме на сторінці клікають користувачі

A/B тестування було описано вище; метод «Веб-аналітика» буде розглянуто нижче.

Докладніше зупинимось на таких методах як:

1. *Спостереження.*

Спостереження — це якісний метод збору даних у UX-дослідженнях, який передбачає аналіз поведінки користувачів без прямого втручання (рис. 17.9).

Час	Крок користувача	Поведінка / Дії користувача	Проблеми / Затримки	Коментарі / Враження користувача
00:01–00:10	Відкрив головну сторінку	Прокручує, не натискає нічого	Виглядає розгубленим	"Я шукаю кнопку, але не бачу її..."
00:11–00:30	Натиснув «Увійти»	Зупинився на полі e-mail	Затримка 15 секунд	«Не впевнений, чи це мій логін»
00:31–01:00	Вводить дані	Видно помилку, повертається до початку	Помилка авторизації	«Можна було б зробити автопідказку...»
01:01–01:20	Завершив авторизацію	Перейшов на дашборд		Виглядає задоволеним

Рис. 17.9. Шаблон аркуша спостереження

Мета:

- Виявити, як люди реально взаємодіють із продуктом;
- Побачити неочевидні проблеми, які користувачі не усвідомлюють або не озвучують.

Види спостереження:

- *Пасивне* – спостереження, за якого спостерігач не втручається, а лише фіксує дії користувача;

- *Активне/модероване* – спостереження, за якого дослідник може задавати питання, просити пояснити дії;
- ***Аудіо/відеозапис*** – спостереження, що дозволяє переглядати реакції в деталях після сесії.

2. *Інтерв'ю*.

Інтерв'ю в *UX*-дослідженні — це якісний метод збору даних, під час якого дослідник ставить користувачу запитання, щоб краще зрозуміти його досвід, потреби, мотивацію, бар'єри та емоції, пов'язані з продуктом або послугою.

Типи інтерв'ю:

- *Структуроване* – тип інтерв'ю, що потребує фіксований список запитань, без відхилень;
- *Напівструктуроване* – тип інтерв'ю, що потребує основні запитання + можливість уточнень;
- *Неструктуроване* – тип інтерв'ю, що передбачає вільну розмову по темі.

Приклад запитань для *UX*-інтерв'ю

Загальні:

- Як часто ви використовуєте подібні сервіси?
- Який був ваш останній досвід користування нашим сайтом/додатком?

Навігація:

- Чи було легко знайти те, що вам потрібно?
- Який розділ викликав найбільше запитань?

Рішення і мотивація:

- Що спонукало вас скористатися цим сервісом?
- Чи були у вас сумніви перед тим, як натиснути «Придбати» / «Зареєструватися»?

Проблеми:

- Що вас розчарувало під час використання?
- Як ви вважаєте, що можна було б покращити?

Очікування:

- Чого вам не вистачило?
- Яка функція, на вашу думку, була б корисною?

3. *Опитування*.

Опитування (*анкетування*) — це кількісний метод *UX*-дослідження, який дозволяє швидко зібрати дані від великої кількості користувачів про їхній досвід, потреби, задоволення, вподобання тощо.

Типи запитань в *UX*-опитуванні

Закриті (кількісні):

- Чи змогли ви легко знайти потрібну функцію? (Так / Ні);
- Оцініть зручність інтерфейсу (від 1 до 5);
- Як часто ви користуєтесь нашим сервісом?

Відкриті (*якісні*):

- Що вам найбільше подобається в нашому продукті?
- Що б ви покращили?

Матричні:

- Оцініть наступні характеристики (1–5) (приклад оцінювання наведено в табл. 17.4):

Таблиця 17.4. Приклад оцінювання характеристик

Критерій	Оцінка
Швидкість роботи	4
Зручність	5
Дизайн	3

Логічні гілки:

- Якщо вибрано «Ні» → показати уточнююче запитання.

Інструменти для створення *UX*-опитувань:

- Google Forms;
- Typeform;
- SurveyMonkey;
- Hotjar (вбудовані опитування на сайті)
- UXtweak / Maze / Usabilla.

Приклад Google Forms наведено на рис. 17.10.

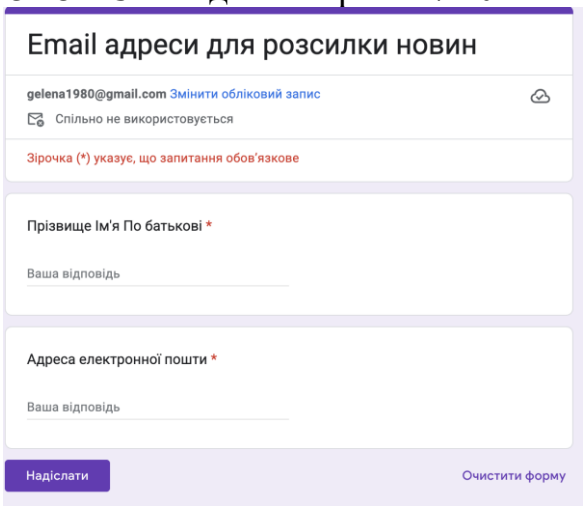


Рис. 17.10. Google Forms опитування $e-mail$ на кафедрі ІСТ

4. *Юзабіліті-тестування.*

Юзабіліті-тестування — це метод *UX*-дослідження, який дозволяє оцінити зручність інтерфейсу продукту (сайту, додатку тощо) шляхом спостереження за реальними користувачами, які виконують типові завдання.

Мета:

- Труднощі в навігації;
- Незрозумілі елементи;
- Проблеми, які заважають досягненню цілей користувача.

Параметри дослідження:

- Успішність виконання;

- Час виконання;
- Кількість помилок;
- Задоволення.

Основні кроки юзабіліті-тестування

Крок 1. Підготовка сценаріїв. Визначаються завдання для користувача.
Крок 2. Відбір учасників. Обираються представники цільової аудиторії.
Крок 3. Проведення тестування. Користувачі виконують завдання в присутності модератора або онлайн.
Крок 4. Збір спостережень. Нотуються помилки, час виконання, коментарі.
Крок 5. Аналіз. формуються рекомендації для покращення інтерфейсу.

5. *Карта кліків (або теплокарта кліків)*

Карта кліків (*або теплокарта кліків*) — це візуалізація дій користувачів на веб-сторінці, яка показує, де саме відвідувачі найчастіше натискали (рис. 17.11).

Рис. 17.11. Візуалізація теплокарти кліків

Вона використовується для:

- Аналізу залученості;
- Виявлення клікабельних зон;
- Виявлення неочевидних місць натискання;
- Оптимізації UX/UI.

17.2.2. Метрики UX-дослідження

17.2.2.1. Середній час оформлення

Середній час оформлення (*Average Checkout Time*) — це метрика, яка показує, скільки в середньому часу користувач витрачає на оформлення покупки (від додавання товару до підтвердження замовлення).

Дана метрика та обчислюється за формулою (17.3), яка також використовується в A/B тестах.

Наприклад:

Користувач A: 2 хв;
Користувач B: 4 хв;
Користувач C: 3 хв.

Тоді

$$\bar{t} = \frac{2+4+3}{3} = 3 \text{ хв.}$$

Приклад візуалізації цієї метрики для даного прикладу на дашборді наведено на рис. 17.12.

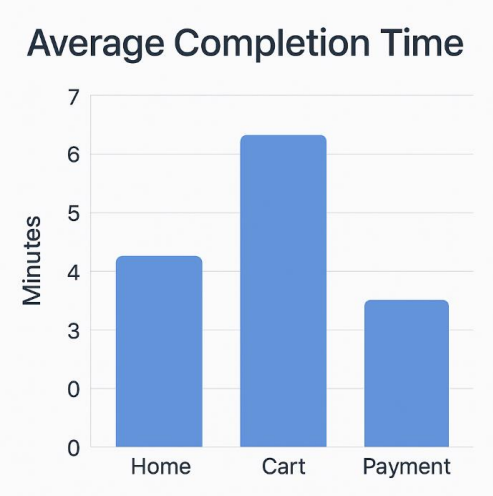


Рис. 17.12. Візуалізація середнього часу оформлення

Як зменшити час оформлення?

- Зменшити кількість полів у формі;
- Додати автозаповнення;
- Дозволити збереження платіжних даних;
- Додати прогрес-бар.

17.2.2.2. $Drop - off$

У контексті UX-аналітики або воронки продажів $drop - off$ означає відтік користувачів між етапами. Його також можна назвати «втратою» або «відмовою» на певному кроці.

$Drop - off$ визначається за формулою (17.22):

$$Drop - off = \left(1 - \frac{n_{\text{нк}}}{n_{\text{пк}}}\right) \cdot 100\%, \qquad (17.21)$$

де:

$n_{\text{нк}}$ – кількість користувачів на наступному кроці;

$n_{\text{пк}}$ – кількість користувачів на попередньому кроці.

Наприклад, розглянемо наступну воронку продажів та розрахуємо $Drop - off$ для кожного кроку (табл. 17.5 та рис.17.13).

Таблиця 17.5. Приклад воронки продажів

Крок	Кількість користувачів	$Drop - off$
Зайшли на сторінку товару	1000	–
Перейшли до кошика	600	40%
Дійшли до оплати	400	33%
Оформили покупку	200	50%

Рис. 17.13. Візуалізація воронки з $Drop-off$

Як відстежувати $Drop-off$?

- Google Analytics (через Funnel visualization або GA4 Exploration Funnel);
- Hotjar / Microsoft Clarity (теплові карти, записи сеансів);
- Mixpanel / Amplitude (гнучкі воронки та сегментація);
- Власна подієва аналітика (наприклад, збереження логів переходів).

Навіщо це потрібно?

- Знайти «вузькі місця» у взаємодії;
- Оптимізувати форму, CTA, екрани;
- Підвищити конверсію;
- Зменшити відтік користувачів.

Як зменшити $Drop-off$?

- Спрощення форм;
- Візуальні підказки / прогрес-бари;
- Збереження введених даних;
- Прозора інформація про оплату та доставку.

17.2.2.3. *Bounce Rate*

Bounce rate (*показник відмов*) — це відсоток користувачів, які залишають сайт після перегляду лише однієї сторінки, не виконавши жодної подальшої взаємодії (наприклад, не клацнули посилання, не заповнили форму тощо).

$Bounce\ Rate$ обчислюється за формулою (17.9) аналогічно до A/B тестів.

17.2.2.4. Кількість скарг

Кількість скарг — це ключова метрика у UX-дослідженнях, підтримці клієнтів та аналізі якості сервісу. Вона показує, скільки користувачів виразили незадоволення щодо продукту, сайту або додатку.

Принципи використання метрики

Перелік метрик, пов'язаних з кількістю скарг користувачів та їх опис наведено в табл. 17.6.

Таблиця 17.6. Метрики, основані на кількості скарг користувачів

№	Метрика	Опис
1	Загальна кількість скарг	Скільки скарг надійшло за певний період

2	Скарги на 1000 користувачів	Нормалізація для коректного порівняння
3	Частка повторних скарг	Показує, чи проблему не вирішено з першого разу
4	Середній час до відповіді	Важливо для швидкості підтримки
5	Тема скарг	Наприклад: UI, баги, повільна робота, оплата і т.д.

Більшість метрик («Загальна кількість скарг», «Частка повторних скарг», «Середній час до відповіді») є інтуїтивно зрозумілими, ґрунтуються на статистичних обрахунках і не потребують додаткових коментарів або розглядалися вище. Метрика «Тема скарг» має бути представлена у формі заяви-скарги (вказана користувачем) чи визначена аналітиком засобами TEXT MINING (див. лекції 11 та 12).

Зупинимося більш докладно на метриці «Скарги на 1000 користувачів».

Формула для нормалізації кількості скарг на 1000 користувачів має вигляд (17.22):

$$NC_{1000} = \frac{NC}{n} \cdot 1000, \qquad (17.22)$$

де:

NC_{1000} – кількість скарг на 1000 користувачів;

NC – загальна кількість скарг;

n – загальна кількість користувачів.

На рис. 17.14 наведено графік, що показує кількість скарг за категоріями.

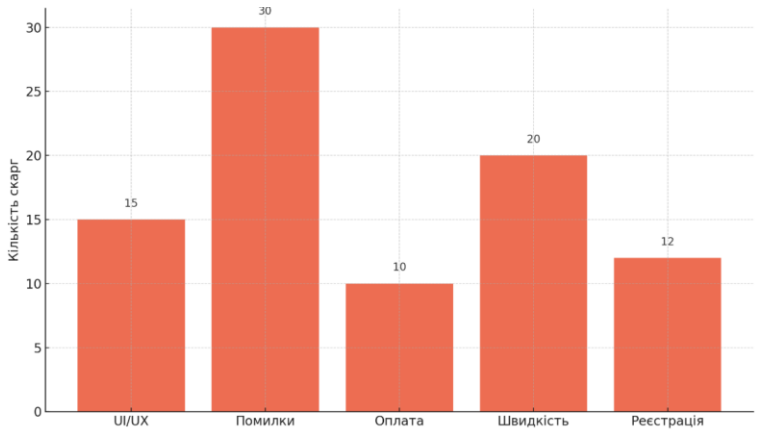


Рис. 17.14. Кількість скарг за категоріями

На даному графіку видно, що найбільше нарікань стосується помилок і швидкості. Це сигнал для команди UX і розробки, що ці зони потребують негайного покращення.

17.2.3. Приклад UX-дослідження

Розглянемо UX-дослідження на прикладі дослідження мобільного додатоку «FoodieGo» (сервіс доставки їжі).

Опис *UX*-дослідження додатоку «FoodieGo»

1. *Мета дослідження.* Покращити зручність використання додатку, скоротити час оформлення замовлення та зменшити кількість відмов на останньому етапі покупки.

2. *Цільова аудиторія.* Обрано осіб за наступними критеріями:

- *Вік*: 18– 45 років;
- *Локація*: міста з населенням $100k+$;
- *Частота використання*: хоча б 1 раз/тиждень;
- *Пристрої*: смартфони Android та iOS.

3. *Методологія.*

Кількісні методи:

- Google Analytics (час на сторінці, bounce rate, drop-off rate);
- A/B-тестування нового процесу оформлення.

Якісні методи:

- Глибинні інтерв'ю;
- Спостереження за поведінкою користувача під час замовлення;
- Опитування через додаток.

Глибинне інтерв'ю — це метод якісного дослідження, який передбачає особисту розмову з респондентом з метою глибокого розуміння його мотивів, думок, емоцій, потреб, поведінки. Особливості глибинного інтерв'ю наведено в табл. 17.7.

Таблиця 17.7. Особливості глибинного інтерв'ю та їх характеристики

№	Ознака	Характеристика
1	Формат	Неформальна, напівструктурована розмова
2	Тривалість	30– 90 хвилин
3	Один-на-один	Зазвичай без присутності інших
4	Сценарій	Є загальний план, але дозволяє імпровізацію
5	Мета	Отримати глибокі інса

4. *Ключові дослідницькі питання.* Наведемо приклади питань для проведення дослідження:

- Що вас мотивувало зайти на сайт?
- Чи виникали труднощі під час пошуку товару?
- Як би ви описали свій досвід оформлення замовлення?
- Які етапи замовлення найбільш заплутані?
- Чи розуміють користувачі, які кроки їм залишились?
- Чи достатньо помітна кнопка «Оформити замовлення»?
- Чи є затримки або баги, які дратують?

5. *Зібрані метрики.* Результати збору та обчислення основних метрик *UX*-дослідження наведено в табл. 17.8.

Таблиця 17.8. Значення основних метрик *UX*-дослідження

Метрика	Значення
Середній час оформлення	3 хв 50 с

$Drop-off$ на етапі оплати	27%
$Bounce\ rate$ з кошика	15%
Кількість скарг	8/1000

6. *Виявлені проблеми.* В результаті дослідження було виявлено наступні проблеми:

- Користувачі плутаються між «Замовити зараз» та «Зберегти на потім»;
- Поле CVV у формі оплати не підписане;
- Додаток «підвисає» при великому замовленні;
- Відсутність прогрес-бару замовлення.

7. *Рекомендації.* Для подолання вказаних проблем було сформовано наступні рекомендації:

- Зробити одну основну кнопку «Оформити замовлення»;
- Додати прогрес-бар (4 етапи: вибір, кошик, адреса, оплата);
- Оптимізувати завантаження даних у кошику;
- Додати підказки у формі оплати (tooltip, іконки).

8. *Подальші кроки.* Для втілення даних рекомендацій необхідно:

- Провести A/B тестування прогрес-бару;
- Дослідити UX новачків у додатку;
- Оновити інтерфейс після 2 тижнів зворотного зв'язку.

Візуалізацію даного UX-дослідження можна побачити на рис. 17.15 та рис. 17.16.

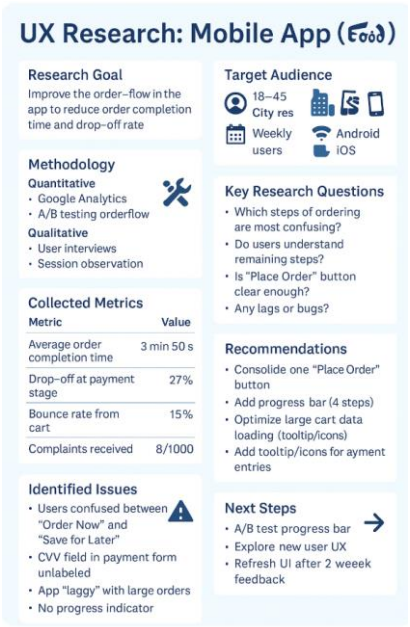


Рис. 17.15. Візуальний дашборд 1 UX-дослідження

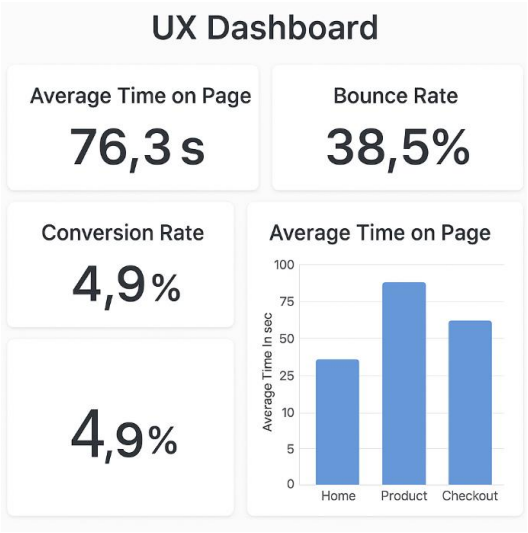


Рис. 17.16. Візуальний дашборд 2 *UX*-дослідження

17.2.4. Реалізація *UX*-дослідження в Python

Вхідні дані

Нехай на вхід подається *CSV*-файл з даними такого вигляду:

```
user_id,page,session_duration,clicked,converted
1,Home,45,1,0
2,Product,120,1,1
3,Home,5,0,0
4,Checkout,200,1,1
5,Home,7,0,0
...
```

UX-аналітика

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Завантаження даних
df = pd.read_csv("user_behavior.csv")

# Середній час взаємодії (session_duration)
avg_time = df["session_duration"].mean()
print(f" Середній час взаємодії: {avg_time:.2f} сек")

# Bounce Rate (користувачі, що не клікнули)
bounces = df[df["clicked"] == 0]
bounce_rate = len(bounces) / len(df) * 100
print(f" Bounce Rate: {bounce_rate:.2f}%")
```

```
# Конверсія
conversion_rate = df["converted"].mean() * 100
print(f" Конверсія: {conversion_rate:.2f}%")

#  Візуалізація часу по сторінках
plt.figure(figsize=(8, 4))
sns.barplot(data=df, x="page", y="session_duration",
estimator="mean", ci=None)
plt.title("Середній час на сторінці")
plt.ylabel("Час (сек)")
plt.show()
```

17.3. Веб-аналітика

Веб-аналітика (*Google Analytics*, *GA*) — це процес збору, вимірювання, аналізу та звітування про дані користувацької активності на сайті чи в додатку. Основна мета — зрозуміти поведінку користувачів та оптимізувати *UX*, маркетинг і бізнес-рішення

Приклад візуалізації *GA* наведено на рис. 17.17.

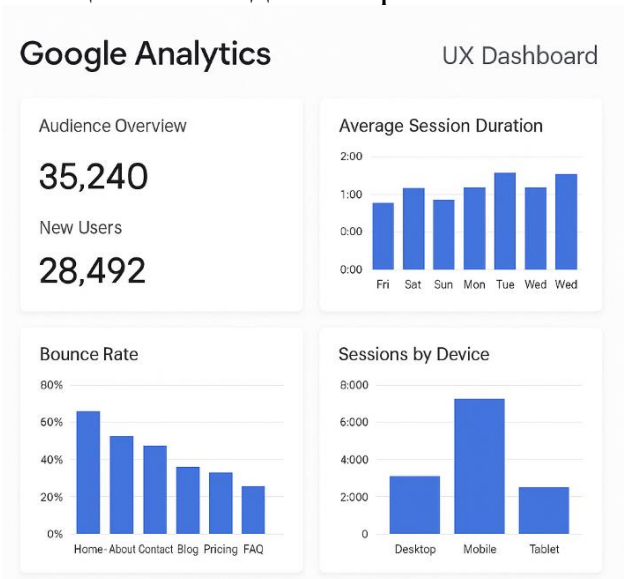


Рис.17.17. Приклад з реальними блоками метрик, що відповідають на *UX*-запити

Функції Google Analytics

1. *Відстеження трафіку.* Хто заходить, звідки, коли, з яких пристроїв;
2. *Поведінка користувачів.* Як рухаються по сайту, які сторінки відвідують, де виходять;
3. *Конверсії та цілі.* Скільки користувачів виконали бажану дію (покупка, підписка тощо);
4. *Аудиторії.* Географія, вік, стать, інтереси;
5. *Події* (*Events*). Кліки по кнопках, відео, завантаження файлів тощо;
6. *Drop-off & Funnels.* Де саме користувачі «випадають» з процесу конверсії.

Ключові метрики UX у Google Analytics

Основні метрики UX у Google Analytics представлено в табл. 17.9.

Таблиця 17.9. Метрики Google Analytics

№	Метрика	Опис
1	$Bounce\ Rate$	Відсоток користувачів, що пішли після першої сторінки
2	$Avg.Session\ Duration$	Середня тривалість взаємодії
3	$Pages/Session$	Скільки сторінок переглянуто за один сеанс
4	CR	Відсоток користувачів, що виконали цільову дію
5	CTR	Клікабельність кнопок/банерів
6	$New\ vs\ Returning\ Users$	Нові чи постійні користувачі
7	$Exit\ Rate$	Де саме користувачі виходять із сайту

Більшість метрик ($Bounce\ Rate$, CR, CTR, $Avg.Session\ Duration$) було розглянуто вище. Метрики $Pages/Session$, $New\ vs\ Returning\ Users$ та $Exit\ Rate$ носять статистичний характер і інтуїтивно зрозумілі.

Google Analytics в UX-дослідженні

Google Analytics широко використовується в UX-дослідженні та A/B-тестах (табл. 17.10).

Таблиця 17.10. Google Analytics в UX-дослідженні

UX-завдання	Як допомагає GA
Виявити «вузькі місця»	Аналіз $Drop-off$, $Exit\ Rate$
Оцінити ефективність редизайну	Порівняти $Bounce\ Rate$ та $Avg.Session\ Duration$
Тестувати A/B варіанти	Порівняти показники конверсії
Налаштувати мікро-цілі	Track події (клік, скрол, фільтр)
Визначити сегменти аудиторії	Аналіз GA, девайсів, типу трафіку

Реалізація Google Analytics в Python

Крок 1. *Встановлення бібліотеки*:

```
pip install google-analytics-data
```

Крок 2. *Авторизація*:

- Перейти на Google Cloud Console;
- Створити проєкт → Увімкнути Google Analytics Data API;
- Створити Service Account (службовий обліковий запис;
- Завантажити $JSON$-файл з ключем;
- Додати $e-mail$ сервіс-акаунта як користувача до $GA4$ ресурсу (в налаштуваннях).

Крок 3. *Код запиту до* $GA4$:

```
from google.analytics.data_v1beta import
BetaAnalyticsDataClient
from google.analytics.data_v1beta.types import
DateRange, Metric, Dimension, RunReportRequest
from google.oauth2 import service_account

# Заміни шлях до свого JSON-файлу
KEY_PATH = "your_service_account_key.json"
PROPERTY_ID = "your_GA4_property_id"  # приклад:
"123456789"

# Авторизація
credentials =
service_account.Credentials.from_service_account_file
(KEY_PATH)
client =
BetaAnalyticsDataClient(credentials=credentials)

# Запит
request = RunReportRequest(
    property=f"properties/{PROPERTY_ID}",
    dimensions=[Dimension(name="country")],
    metrics=[Metric(name="sessions")],
    date_ranges=[DateRange(start_date="2024-01-01",
end_date="2024-12-31")]
)

response = client.run_report(request)

# Вивід результатів
for row in response.rows:
    print(f"{row.dimension_values[0].value}:
{row.metric_values[0].value}")
```

17.4. *UX/UI*-дизайн

UX/UI-дизайн — це процес створення зручних, ефективних і привабливих інтерфейсів для користувачів. Він об'єднує аналітичне мислення *UX*-дизайну з естетикою *UI*-дизайну, щоб користувач легко взаємодіяв із сайтом, додатком чи платформою.

Компоненти *UX/UI*-дизайну

1. *UX-дизайн (User Experience).* Основні компоненти *UX*-дизайну представлено в табл. 17.11. Вони фокусується на логіці та структурі користувацького шляху.

Таблиця 17.11. Компоненти UX-дизайну

№	Компонент	Опис
1	Дослідження	Аналіз поведінки користувачів, опитування, інтерв'ю
2	Архітектура	Побудова логічної структури (інформаційна архітектура)
3	Прототипування	Створення схематичного функціоналу ($wireframes, mockups$)
4	Тестування	Виявлення проблем за допомогою юзабіліті-тестів
5	Аналітика	Аналіз дій користувачів через Google Analytics, теплокарти

2. *UI-дизайн (User Interface).* Основні компоненти UI-дизайну представлено в табл. 17.12. Фокусується на зовнішньому вигляді та взаємодії:

Таблиця 17.12. Компоненти UI-дизайну

№	Компонент	Опис
1	Візуальний стиль	Кольори, типографіка, іконографіка
2	Інтерактивність	Стани кнопок, анімації, переходи
3	Адаптивність	Зручність на мобільних, планшетах, десктопах
4	Дизайн-системи	Використання стандартів ($Material$, $Apple\ HIG$ тощо)
5	Мова інтерфейсу	Текст, мікрокопі, підказки — частина $UX\ writing$

Цілі UX/UI-дизайну:

- Зменшити фрустрацію користувача;
- Збільшити конверсію;
- Покращити зручність навігації;
- Створити позитивний емоційний досвід;
- Вирішити проблеми користувача інтуїтивно.

Інструменти UX/UI-дизайну (табл. 17.13):

Таблиця 17.13. Інструменти UX/UI-дизайну

UX	UI
Figma	Figma
Miro	Adobe XD
Maze	Sketch
Hotjar	Framer
Google Analytics	Zeplin

ЛЕКЦІЯ 18

ЯК ПРОВОДИТИ АНАЛІТИЧНІ ДОСЛІДЖЕННЯ

18.1. Поняття та типи аналітичного дослідження

18.1.1. Поняття аналітичного дослідження

Аналітичне дослідження – це процес збору та аналізу даних в ході оцінки певної проблеми, теми чи наявної політики, з метою формування пропозицій для зміни поточної ситуації.

Зазвичай, *оптимальне рішення* – це замовити аналітику у профільних аналітичних центрів (think tanks).

Наприклад, Інститут аналітики та адвокації завжди готовий реалізувати дослідження на Ваш запит у профільних для ІАА темах. Організація співпрацює з сервісними та правозахисними установами, створює для них стратегічно важливі документи, які потім використовуються в адвокації. Будь-яке дослідження зосереджене над проблемою, яка потребує вирішення. Важливо розуміти, що аналіз проблеми є обов'язковою передумовою ефективної адвокаційної кампанії. Перш ніж змінювати існуючі практики, потрібно досконало проаналізувати, де саме необхідні зміни та якими аргументами «озброїтися» для ефективного переконання, підсиливши позицію фактами та доказами. Пам'ятайте, що усе починається з технічного завдання. Це необхідна передумова якісного дослідження, адже саме на поставленій задачі будується методологія і перевіряється, наскільки фактичний результат відповідає запиту. Ми рекомендуємо залучати дослідників з навичками та досвідом розробки технічного завдання.

18.1.2. Типи аналітичних досліджень

Всі дослідження поділяються на описові та аналітичні (рис. 18.1).

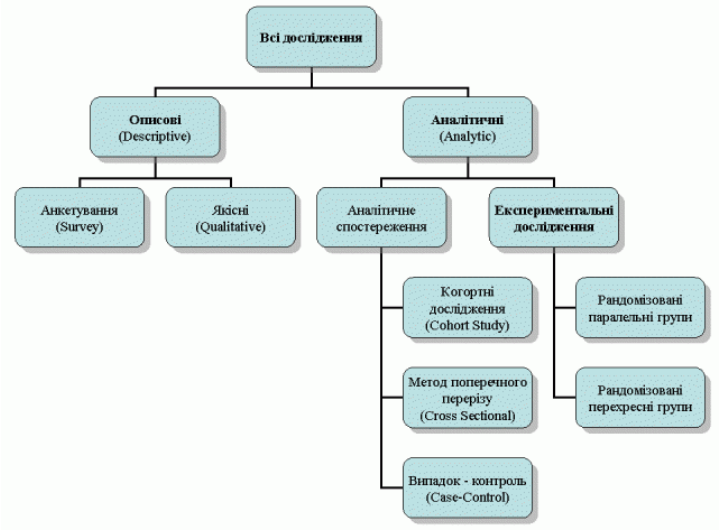


Рис. 18.1. Види досліджень

Зупинимося докладно на цих видах.

18.1.2.1. Описові дослідження

Описові дослідження не ставлять собі за мету вияснити взаємовідносини між різними факторами, а лише охарактеризувати, що відбувається в популяції, наприклад поширеність захворювання.

Вони включають:

- Анкетування;
- Якісні дослідження.

1. Анкетування

Анкетування (*Survey*) — це метод збору даних шляхом заповнення респондентами письмових або електронних опитувальників, які містять запитання для вивчення думок, поведінки, досвіду чи характеристик.

Основні риси анкетування:

- Може бути паперовим або онлайн (Google Forms, SurveyMonkey тощо);
- Містить відкриті, закриті або шкальні запитання;
- Забезпечує масовий збір інформації за короткий час;
- Часто використовується в соціології, педагогіці, маркетингу, психології;
- Результати можна кількісно аналізувати.

Приклад:

Запитання:

- «Як часто ви користуєтесь інтерактивними технологіями під час навчання?»

Відповіді:

- Щодня;
- Кілька разів на тиждень;
- Рідко;
- Ніколи.

2. Якісні дослідження

Якісні дослідження – це тип наукового дослідження, метою якого є глибоке розуміння явищ, мотивацій, поведінки чи досвіду через нечислові (вербальні, візуальні, текстові) дані.

Основні характеристики:

- Орієнтовані на зміст, а не на кількість;
- Застосовуються при вивченні соціальних, культурних, освітніх, психологічних процесів;
- Дані збираються через:

1. Інтерв’ю;
2. Фокус-групи;
3. Спостереження;
4. Аналіз документів;
5. Вільні відповіді на запитання;

- Результати зазвичай оформлюються у вигляді тематичного аналізу, описів, категорій.

Приклад:

- Інтерв'ювання викладачів про досвід використання онлайн-навчання під час війни, з подальшим виділенням основних тем:

1. Виклики;
2. Переваги;
3. Адаптація.

Порівняння анкетування та якісних досліджень

Порівняння методів описового дослідження наведено в табл. 18.1.

Таблиця 18.1. Анкетування *vs* Якісні дослідження

Критерій	Анкетування (кількісний метод)	Якісні дослідження
Мета	Вимірювання, порівняння, узагальнення	Глибоке розуміння досвіду, мотивацій, смислів
Тип даних	Кількісні (цифри, шкали, категорії)	Якісні (слова, наративи, описові відповіді)
Методи збору	Опитувальники, тести, онлайн-форми	Інтерв'ю, фокус-групи, спостереження
Аналіз	Статистичний (середнє, відсотки, кореляції)	Тематичний, контент-аналіз, інтерпретація
Розмір вибірки	Великий (50–1000+ учасників)	Малий (5–30 учасників — "глибина, не кількість")
Час і ресурсозатратність	Швидко, дешево (особливо онлайн)	Повільно, трудомістко (аналіз інтерв'ю тощо)
Стандартизація	Стандартизовані запитання, однакові для всіх	Гнучкі, адаптивні запитання залежно від контексту

18.1.2.2. Аналітичні дослідження

Аналітичні дослідження спрямовані на визначення взаємозв'язків між різними факторами впливу, наприклад певного методу лікування пацієнта на результати. Для визначення впливу необхідно порівняти результати у контрольній та експериментальній групі.

Аналітичні дослідження можуть бути:

- Аналітично-спостережними (при пасивному спостереженні);
- Експериментальними (при активному впливі дослідника).

18.1.2.2.1. Аналітично-спостережні дослідження

При *аналітичному спостереженні* науковець просто вимірює рівень певного впливу (наприклад лікування) на групу осіб.

Метод аналітичного спостереження включає:

- Когортні дослідження;
- Випадок-контроль;
- Метод поперечного перерізу.

Докладніше зупинимося на них.

1. Когорнтне дослідження

Когортне дослідження (*Cohort study*) – це обсерваційне дослідження, у якому виділену групу людей (когорту) спостерігають протягом деякого часу і порівнюються результати у різних підгрупах даної когорти. Ці дослідження можуть бути:

- Ретроспективними;
- Проспективними.

1. *Ретроспективне дослідження.*

Ретроспективне дослідження — це тип наукового дослідження, при якому аналізуються вже наявні дані або події, що вже відбулися у минулому.

Мета:

- Виявлення зв'язків, закономірностей або факторів, які могли вплинути на результати.

Основні характеристики:

- Використовує історичні дані (наприклад, архіви, звіти, анкети, медичні картки, освітні журнали);
- Часто застосовується в медицині, соціології, освіті, криміналістиці, економіці;
- Здебільшого не передбачає експериментального втручання;
- Може бути дешевшим та швидшим за проспективне дослідження (яке стежить за подіями в реальному часі).

Приклад:

- Дослідник вивчає, як успішність студентів за останні 5 років пов’язана з їхнім рівнем відвідуваності, використовуючи архівні журнали викладачів.

2. *Проспективне дослідження.*

Проспективне дослідження — це тип дослідження, при якому спостереження або збір даних відбувається в реальному часі або у майбутньому, після початку дослідження. Дослідник випереджує події, щоб простежити, як певні фактори впливають на результати.

Мета:

- Виявлення причинно-наслідкових зв’язків.

Основні характеристики:

- Дані збираються поступово, у міру того як події відбуваються;
- Часто використовується у клінічних випробуваннях, освітніх експериментах, поведінкових дослідженнях;
- Зазвичай триваліше та дорожче, ніж ретроспективне.

Приклад:

- Дослідник формує групу студентів і впродовж року відстежує, як змінюється їх успішність залежно від застосування різних методик навчання (наприклад, гейміфікація або традиційні лекції).

2. Випадок-контроль

Випадок-контроль (*Case control study*) – це тип ретроспективного дослідження, у якому група людей з певним результатом (наприклад, хворобою, поведінкою, подією) — «випадки» — порівнюється з групою без цього результату — «контроль», щоб виявити, які чинники могли сприяти його виникненню (рис. 18.2).

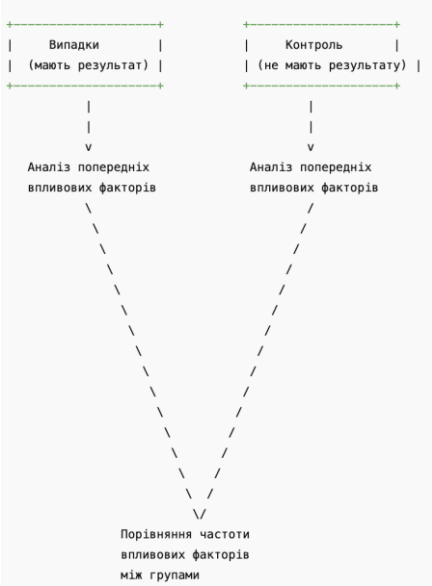


Рис. 18.2. Схема дослідження «випадок-контроль»

Основні риси:

- Починається з наслідку, а не з причини;
- Дослідник шукає зв’язок між результатом і можливими впливовими факторами в минулому;
- Часто застосовується у медицині, психології, соціології.

Приклад:

- Вивчається група студентів, які вибули з університету (випадки), і група, яка успішно продовжує навчання (контроль). Аналізуються попередні чинники: рівень успішності, соціально-економічне становище, участь у позааудиторній діяльності тощо.

3. Метод поперечного перерізу

Метод поперечного перерізу (*cross-sectional study*) — це тип дослідження, у якому дані збираються одночасно в один момент часу для того, щоб описати характеристики, стани або зв’язки між змінними в певній сукупності (рис. 18.3).

Рис. 18.3. Візуалізація поперечного розрізу

Основні риси:

- Дослідження «тут і зараз», без аналізу в динаміці;
- Вивчається одна або кілька груп на один момент часу;
- Часто використовується для:

1. Опису стану речей (наприклад, рівень задоволеності);
2. Виявлення зв'язків між змінними (але не причинно-наслідкових зв'язків).

Приклад:

- У 2024 році проводиться опитування студентів 2 курсу різних ЗВО щодо їхнього ставлення до дистанційного навчання. Аналізуються: стать, спеціальність, регіон, задоволеність, мотивація.

Переваги:

- Швидке й економічне у виконанні;
- Добре підходить для масових опитувань.

Обмеження:

- Не дозволяє оцінити зміни у часі;
- Не встановлює причинність між змінними.

Порівняння методів аналітично-спостережного дослідження

Порівняння методів аналітичного спостереження наведено в табл. 18.2.

Таблиця 18.2. Когортне *vs* Випадок-контроль *vs* Поперечний переріз

Критерій	Когортне	Випадок-контроль	Поперечний переріз
Тип дослідження	Проспективне (або ретроспективне)	Ретроспективне	Описове (одночасне)
Мета	Визначити ризики розвитку результату	Виявити чинники, пов'язані з наслідком	Описати ситуацію «тут і зараз»
Вибірка	Люди без результату, але з/без впливу	Люди з результатом	Випадкова (репрезентативна) вибірка

		(випадки) та без нього	
Часова орієнтація	В майбутнє (або минуле в ретро-когорті)	У минуле	На момент збору даних
Причинно-наслідкові висновки	Так, сильні аргументи	Обмежені	Немає
Тривалість дослідження	Довга (може тривати роки)	Середня	Коротка
Вартість	Висока	Середня	Низька

18.1.2.2.2. Експериментальні дослідження

Експериментальні дослідження – це науковий підхід, у якому дослідник свідомо втручається в умови або змінні, щоб вивчити їхній вплив на інші змінні.

Мета:

- Встановлення причинно-наслідкових зв'язків між факторами.

Основні риси експериментального дослідження:

- Є експериментальна та контрольна групи;
- Одна або кілька змінних маніпулюються (незалежні змінні);
- Інші змінні вимірюються (залежні змінні);
- Проводиться у контрольованих умовах (лабораторія, клас, онлайн-середовище тощо);
- Часто застосовуються рандомізація та блайнд-дизайн (у медичних чи психологічних дослідженнях).

Приклад:

- Дослідник розділяє учнів на дві групи: одній викладають тему за допомогою відео, іншій — традиційним методом. Потім порівнюють результати тестування.

Дизайн експериментального дослідження

Дизайн експериментального дослідження — це структурований план проведення дослідження, який визначає, як буде організовано експеримент, щоб отримати об'єктивні, надійні та відтворювані результати.

Розрізняють наступні два основні види дизайну експериментального дослідження.

1. *Рандомізовані паралельні групи.*

Рандомізовані паралельні групи — це дизайн експериментального дослідження, при якому учасників випадковим чином (рандомізовано) розподіляють на дві або більше незалежних груп, кожна з яких отримує різне втручання (або плацебо/контроль) (рис. 18.4).

Рис. 18.4. Візуалізація дизайну «Рандомізовані праралельні групи»

Основні характеристики наведено в табл. 18.3:

Таблиця 18.3. Характеристики дизайну «Рандомізовані паралельні групи»

Елемент	Опис
Мета	Визначити ефективність втручання або впливу
Рандомізація	Учасники розподіляються випадково, щоб усунути упередження
Паралельність	Групи існують одночасно, але незалежно, без перетину
Втручання	Наприклад, одна група — нова методика навчання, інша — стандартна
Аналіз	Порівняння результатів між групами (ANOVA, t-тест тощо)

ANOVA (*Analysis of Variance*, *аналіз дисперсії*) — це статистичний метод, який дозволяє порівнювати середні значення трьох або більше груп і визначити, чи є між ними статистично значуща різниця (використовується F-критерій).

t-тест — це статистичний метод, який використовується для порівняння середніх значень двох груп, щоб визначити, чи є між ними статистично значуща різниця (використовується t-критерій).

Приклад:

- Вивчається ефективність нової онлайн-платформи. 100 студентів розподілено випадково:

Група 1 — працює через нову платформу;

Група 2 — навчається традиційно.

Через місяць порівнюють рівень знань.

2. *Рандомізовані перехресні групи.*

Рандомізовані перехресні групи (*Randomized Crossover Design*) — це тип експериментального дослідження, в якому кожен учасник послідовно отримує кілька втручань (лікувань, методик і т.д.), але в різному порядку,

причому порядок визначається випадковим чином (рандомізацією) (рис. 18.5).

Рис. 18.5. Візуалізація дизайну «Рандомізовані перехресні групи»

Суть даного дизайну наведено в табл. 18.4.

Таблиця 18.4. Суть дизайну «Рандомізовані перехресні групи»

Учасники	Період 1	Період 2
Група *A*	Інтервенція *X*	Інтервенція *Y*
Група *B*	Інтервенція *Y*	Інтервенція *X*

Після першого періоду йде «період відмивання» (washout), щоб ефект попереднього втручання не вплинув на наступне.

Переваги:

- Усі учасники отримують усі втручання → зменшується вплив індивідуальних відмінностей;
- Менше учасників потрібно, ніж у паралельних групах;
- Збільшує статистичну потужність (точність порівняння).

Недоліки:

- Не підходить, якщо ефект першого втручання довготривалий або незворотний;
- Не можна застосовувати, якщо є потужний ефект переносу (carryover effect);
- Потрібен час на «відмивання» між фазами.

Приклад:

- Дослідження двох методик викладання:

1. Студенти групи *A* спочатку навчаються методом проєктів, потім традиційно.
2. Студенти групи *B* — навпаки.

Порівнюють результати в обох періодах.

Порівняння дизайнів експериментального дослідження

В табл. 18.5 наведено порівняння рандомізованих паралельних та рандомізованих перехресних груп у дослідженні.

Таблиця 18.5. Таблиця порівняння рандомізованих паралельних та рандомізованих перехресних груп у дослідженні

Ознака	Рандомізовані паралельні групи	Рандомізовані перехресні групи
Суть	Учасники розподіляються у дві (або більше) груп і отримують різні втручання одночасно	Кожен учасник послідовно отримує всі втручання в різний час
Кількість груп	Кілька незалежних груп	Одна група або декілька, де учасники міняються ролями
Тип розподілу	Випадковий розподіл учасників у групи	Випадковий порядок отримання втручань кожним учасником
Переваги	Простий дизайн, уникнення ефекту порядку	Менше учасників, кожен є власним контролем
Недоліки	Потрібно більше учасників, можливий вплив варіацій між особами	Ризик впливу попереднього втручання (ефект переносу)
Коли застосовують	Коли очікується стійкий ефект або тривалий вплив	Коли ефекти втручання короткочасні і зворотні
Аналіз результатів	Порівняння між групами	Порівняння між періодами в межах одного учасника

Висновок:

- Паралельний дизайн — кращий для втручань з тривалим або незворотним ефектом;
- Перехресний дизайн — ефективний, якщо можна уникнути перенесення ефекту і якщо дослідження маломасштабне.

18.2. Етапи аналітичного дослідження

Для самостійного провести дослідження необхідно виконати 9 етапів аналітичного дослідження (рис. 18.6).

Рис. 18.6. Етапи аналітичного дослідження

Слід зазначити, що цю інструкцію розроблено на основі можливостей, які створює українське законодавство. У випадку її застосування для інших країн необхідно враховувати місцевий контекст та нормативні особливості.

Докладно зупинимось на кожному з цих етапів.

18.2.1. Підготовка

Підготовка аналітичного дослідження — це системний процес планування, організації і забезпечення всіх умов, необхідних для проведення дослідження з метою отримання достовірної, обґрунтованої інформації для аналізу та прийняття рішень.

Усе починається з гіпотез – припущень, які перевірятимуться протягом дослідження. Це вектор, який має відправну точку для Вашого дослідження та дає інформацію про те, куди рухатися. Підтверджена в результаті дослідження гіпотеза може виявити нову проблему в суспільстві або розширити розуміння вже наявної.

На цьому етапі здійснюється пошук доступної інформації у інтернет-мережі та інших відкритих джерелах щодо проблеми.

У першу чергу слід ознайомитися з веб-сайтами центральних і місцевих органів влади та самоврядування, зокрема Кабміну, міністерств, Держстату, облдержадміністрацій, міських рад тощо.

Також необхідно здійснити пошук нормативно-правових актів, які регулюють обране питання. Наприклад, веб-сайт Верховної Ради України чи сайти обласних рад є доступними джерелами подібної інформації.

Аналітичні центри, зазвичай, користуються заздалегідь сформованою базою інтернет-ресурсів, що дозволяє їм оптимізувати підготовчий етап. Також корисними стануть консультації з профільними спеціалістами та експертами у обраній темі. Їхні знання допоможуть зрозуміти контекст ситуації.

На основі зібраної інформації розробляється *методологія дослідження* – покрокова інструкція майбутніх дій: особливостей аналізу, бажаного результату, необхідних ресурсів для досягнення мети.

Структура методології дослідження

1. *Вибір типу дослідження*:
- Кількісне (статистика, опитування, експерименти);
- Якісне (інтерв'ю, спостереження, аналіз контенту);
- Змішане (комбінація кількісних і якісних методів).

2. *Формування гіпотез і питань дослідження*:
- Що саме хочемо перевірити або дізнатися?

3. *Вибір методів збору даних*:
- Анкетування;
- Інтерв’ю;
- Спостереження;
- Аналіз документів;
- Експерименти.

4. *Вибір вибірки*:
- Хто саме буде брати участь?
- Скільки учасників потрібно?
- Який спосіб відбору (випадковий, цілеспрямований тощо)?

5. *Опис процесу збору даних*:
- Інструменти (анкети, платформи, пристрої);
- Умови та середовище збору.

6. *Опис методів аналізу даних*:
- Статистичний аналіз;
- Контент-аналіз;
- Машинне навчання;
- Інші специфічні техніки.

7. *Етичні питання*:
- Конфіденційність;
- Добровільна участь;
- Інформована згода.

Також необхідно розробити *стратегію протидії ризикам* — це план або набір заходів, які допомагають виявити, оцінити та мінімізувати негативні наслідки можливих ризиків у проєктах, бізнесі чи дослідженнях.

Стратегії протидії ризикам

1. *Визначення ризиків*:
- Ідентифікація можливих загроз і проблем;
- Оцінка ймовірності настання та потенційних збитків.

2. *Оцінка ризиків*:
- Класифікація за рівнем впливу (високий, середній, низький);
- Пріоритизація ризиків.

3. *Вибір стратегії реагування*:
- Уникнення ризику (зміна плану, щоб прибрати загрозу);
- Передача ризику (страхування, аутсорсинг);
- Зменшення ризику (впровадження заходів контролю);
- Прийняття ризику (готовність взяти на себе наслідки).

4. *Розробка плану дій*:
- Що робити у разі настання ризику;
- Відповідальні особи.

5. *Моніторинг і перегляд*:
- Постійне спостереження за ризиками;
- Актуалізація плану за потреби.

Результат та продукт етапу підготовки аналітичного дослідження

Результат:
- Розуміння обраної теми, її нормативного регулювання, контексту.

Продукт:
- База зібраної інформації;
- Методологія дослідження;
- Стратегія протидії ризикам.

18.2.2. Збір інформації

Збір інформації — це перший і один із найважливіших етапів будь-якого аналітичного дослідження чи прийняття рішень. Це систематичний процес пошуку, виявлення та отримання необхідних даних або фактів для вирішення певного завдання чи проблеми.

Основні принципи якісного збору інформації

1. Достовірність;
2. Повнота;
3. Актуальність;
4. Репрезентативність.

Основні етапи збору інформації

1. *Формулювання цілей*:
- Що саме потрібно дізнатися і навіщо?

2. *Визначення джерел інформації*:
- Первинні дані (опитування, експерименти);
- Вторинні дані (звіти, література, статистика, інтернет-ресурси).

3. *Вибір методів збору*:
- Інтерв'ю;
- Анкетування;
- Спостереження;
- Аналіз документів;
- Веб-аналітика.

4. *Фіксація та організація інформації*:
- Збереження у зручному форматі (бази даних, таблиці, звіти).

5. *Перевірка достовірності*:
- Перехресна перевірка з різних джерел.

Рекомендації щодо збору інформації

1. Дослідник має оцінити сформовану базу інформації та виокремити потребу в додаткових даних;

2. У результаті слід сформувати список необхідної додаткової інформації, згрупувавши за критерієм органів влади, які можуть володіти даними;
3. На основі списку необхідно сформувати та розіслати запити на доступ до публічної інформації;
4. Можна використовувати паперову або електронну форми. У випадку паперової форми варто направляти лист-запит із повідомленням про вручення. У другому випадку слід зателефонувати та переконатися, що запит дійшов до адресата, отримавши реєстраційний вхідний номер. Через п'ять робочих днів здійснюється контроль за отриманням інформації (Закон України «Про доступ до публічної інформації», ст. 20, ч. 1);
5. Після збору додаткових даних удосконалюється сформована раніше база зібраної інформації. На збір інформації важливо планувати достатньо часу у випадку продовження строку відповіді на запит до 20 робочих днів. Також може виникати потреба надіслати додаткові уточнюючі запити;
6. Отримана за запитом інформація має неодмінно містити супровідний лист органу влади. Усне спілкування і листування без супровідного листа є ненадійними джерелами даних, які не бажано використовувати в дослідженні;
7. Слід прохати надати інформацію у машиночитному форматі. Це збереже час на етапі обробки даних;
8. У випадку відмови відповідати на запит можливо самостійно навідатися до органу влади та сфотографувати/скопіювати необхідні дані у спеціально відведеному для цього місці (Закон України «Про доступ до публічної інформації», ст. 14, ч. 1, п. 4);
9. Необхідно знайти аналітичні звіти за обраною темою. Це допоможе уникнути помилок попередників та удосконалити поточне дослідження. Цікавим може стати порівняння отриманих даних з більш ранніми зі звітів. Це покаже динаміку процесів;
10. Рекомендується зберігати отримані дані у одному місці, зокрема у хмарному сховищі Google-Disk, на окремих серверах тощо.

Результат та продукт етапу збору інформації для аналітичного дослідження

Результат:
- Написання інформаційних запитів та контроль за отриманням інформації;
- Комунікація та налагодження взаємодії з органами влади;
- Краще розуміння досліджуваної теми.

Продукт:
- Список необхідної додаткової інформації;
- Запити на доступ до публічної інформації;
- Оновлена база зібраної інформації.

18.2.3. Обробка та аналіз отриманих даних

Обробка та аналіз отриманих даних — це етап аналітичного дослідження, що передбачає систематизацію, очищення, перетворення та інтерпретацію зібраної інформації для виявлення закономірностей, тенденцій і формування обґрунтованих висновків.

Рекомендації щодо обробки та рекомендації отриманих даних

1. Зазвичай органи влади надсилають інформацію у вигляді сканованих або паперових копій;
2. Для зручності отримані дані слід перенести до завчасно сформованих аналітичних *Excel*-таблиць або інших аналітичних програм (R-Studio тощо);
3. *Excel*-таблиці мають компонуватися згідно з методологією дослідження та містити усі розрахунки й операції з даними;
4. Формуючи таблицю, важливо прописати усі формули для автоматичного розрахунку значень та додаткові формули для перевірки. У такий спосіб аналітичні центри убезпечують себе від хибних результатів, заощаджуючи час.

Методи обробки та аналізу даних

1. *Статистичний аналіз* – обчислення середніх, медіан, дисперсій, кореляцій для виявлення основних закономірностей.
2. *Машинне навчання* – побудова моделей класифікації, регресії, кластеризації для виявлення прихованих структур або прогнозування.
3. *Кластеризація* – групування об'єктів за схожими характеристиками (методи: *K*-means, DBSCAN, ієрархічна кластеризація).
4. *Регресійний аналіз* – виявлення залежностей між змінними (лінійна, множинна, логістична регресія).
5. *Візуалізація даних* – створення графіків, теплових карт, діаграм для інтуїтивного розуміння великих обсягів інформації.
6. *Аналіз часових рядів* – вивчення даних, змінних у часі (наприклад, прогнозування продажів, зміни трафіку).
7. *Текстовий аналіз (NLP)* – обробка текстової інформації: класифікація документів, виявлення тональності, пошук ключових слів.
8. *Побудова моделей рішень* – створення сценаріїв та моделей для оптимізації прийняття рішень (наприклад, дерево рішень).

Розглянемо таблицю, яка показує, які методи краще підходять для різних типів даних (табл. 18.6).

Таблиця 18.6. Таблиця відповідності методів обробки та аналізу типам даних

Тип даних	Рекомендовані методи
Числові дані	Статистичний аналіз, регресія, кластеризація, машинне навчання, аналіз часових рядів
Категоріальні дані	Класифікація, асоціативні правила, кластеризація
Текстові дані	Текстовий аналіз (NLP), тематичне моделювання (LDA), класифікація текстів

Часові ряди	Аналіз часових рядів, прогнозування (ARIMA, Prophet)
Просторові дані	Геоаналітика, кластеризація, просторовий аналіз
Великі обсяги даних	Машинне навчання, глибинне навчання, паралельна обробка (Hadoop, Spark)

Результат та продукт етапу аналізу та обробки отриманих даних для аналітичного дослідження

Результат:

- Практика електронізації даних;
- Робота з *Microsoft Excel*;
- Математичний аналіз даних.

Продукт:

- Аналітичні *Excel*-таблиці.

18.2.4. Опис аналізу даних

Після електронізації та аналізу даних відбувається текстовий опис результатів.

Опис аналізу даних — це пояснення процесу обробки, перевірки, інтерпретації та представлення даних, який допомагає виявити закономірності, тренди, залежності та підтримати ухвалення обґрунтованих рішень.

Основні моменти опису аналізу даних включають

1. *Мета аналізу* — що саме потрібно дослідити чи з'ясувати.
2. *Джерела даних* — звідки взято дані (опитування, бази даних, експерименти тощо).
3. *Методи обробки* — які техніки використовуються для очищення, структурування, трансформації даних.
4. *Методи аналізу* — які аналітичні чи статистичні методи застосовані (кластеризація, регресія, кореляція тощо).
5. *Інструменти аналізу* — використане програмне забезпечення або платформи (*Excel*, Python, R, Tableau тощо).
6. *Основні висновки* — результати, тренди, рекомендації, які випливають із аналізу.

Рекомендації до створення аналітичного звіту

1. Науковий стиль вважається найвідповіднішим для аналітичних звітів;
2. Слід уникати плагіату та використовуйте цитування;
3. Необхідно робити доступною інформацію з нормативно-правових актів;
4. Необхідно групувати цифрові дані у прості таблиці. Їхній опис відбувається за наступною схемою:
   - Текстова підводка до таблиці;
   - Таблиця;
   - Висновки з табличних даних;
5. Слід уникати оціночних суджень і пасивної форми дієслів (отримано, сформовано, досліджено змінюйте на отримали, сформували, дослідили);

6. Не слід робити довгих та складних речень;
7. Писати абзацами давно не модно;
8. У тексті слід виділяти мінімальні та максимальні значення, простежувати тенденції та закономірності, робити паралелі й порівняння.

Опис аналізу даних для дослідження продажів

Мета:

- Проаналізувати динаміку продажів за останній рік і виявити фактори, що впливають на обсяги продажів.

Джерела даних:

- CRM-система компанії (контакти клієнтів, історія покупок);
- Google Analytics (джерела трафіку, поведінка на сайті);
- Дані про маркетингові кампанії (розсилки, реклама).

Методи обробки:

- Очищення даних від невалідних транзакцій;
- Уніфікація назв товарів та категорій;
- Побудова часових рядів продажів за місяцями.

Методи аналізу:

- Описова статистика (загальний дохід, середній чек);
- Регресійний аналіз для оцінки впливу кампаній на продажі;
- Побудова когортного аналізу для оцінки поведінки нових клієнтів.

Інструменти:

- SQL для витягування даних;
- Python (pandas, seaborn, statsmodels);
- Power BI для візуалізації результатів.

Основні висновки:

- 30% загального доходу забезпечують 15% постійних клієнтів;
- Пряма залежність між кількістю $e-mail$-розсилок і кількістю покупок;
- Період знижок у листопаді-грудні приносить удвічі більше нових клієнтів.

Результат та продукт етапу опису аналізу даних для аналітичного дослідження

Результат:

- Опис результатів аналізу даних.

Продукт:

- Чернетка аналітичного документу.

18.2.5. Формування висновків та пропозицій

На основі здійсненого дослідження, оперуючи отриманими даними та результатами їхнього аналізу, слід сформувати висновки та пропозиції.

Формування висновків та пропозицій — це завершальний етап аналітичного дослідження, де на основі оброблених і проаналізованих даних формуються висновки та пропозиції.

1. *Висновки* — це узагальнення основних результатів дослідження. Це фактичні твердження, що відображають знайдені закономірності, проблеми, сильні сторони або ризики.

Наприклад: «Зростання продажів на 20% пов'язане із запуском нової маркетингової кампанії».

2. *Пропозиції* — це практичні рекомендації, які випливають із висновків і спрямовані на вирішення проблем, покращення процесів чи досягнення цілей.

Наприклад: «Рекомендується збільшити інвестиції в рекламні кампанії в соціальних мережах для подальшого зростання продажів».

Рекомендації щодо формування висновків та рекомендацій

1. Варто менше вживати канцеляризми у тексті;
2. Ці розділи мають бути доступними для читача та водночас підтверджувати експертний рівень аналітика;
3. У висновках рекомендується синтезувати результати дослідження та описати їх, згадати про новизну й значення дослідженого в контексті наявних політик;
4. Висновки є одним із найбільш читаних фрагментів будь-якого тексту;
5. До розділу пропозицій слід внести конкретні та реалістичні практичні кроки. Адресуйте окремі тези конкретним органам влади;
6. Пропозиції є найважливішим компонентом, оскільки кінцева мета дослідження – це не збір та аналіз даних, а пошук вирішення проблем у обраній темі;
7. Важливо розуміти різницю між використанням термінів «рекомендації» та «пропозиції». За своїм змістом вони синонімічні. Одні й ті ж практичні кроки можна назвати як рекомендаціями, так і пропозиціями. Однак рекомендації можуть бути між рівними собі за статусом, а пропозиції – ні.

Наприклад, рекомендації одного органу влади іншому зобов'язують останнього відреагувати. Водночас у комунікації з НУО органи влади не зобов'язані реагувати на сформовані приписи, тому громадським організаціям доцільніше користуватися терміном «пропозиції».

8. У структурі дані розділи мають бути:

- *Висновки*: короткі, чіткі, логічні, відповідають на поставлені завдання дослідження.
- *Пропозиції*: конкретні, реалістичні, засновані на отриманих даних, орієнтовані на дію.

Результат та продукт етапу формування висновків та пропозицій для аналітичного дослідження

Результат:

- Синтез результатів дослідження;
- Виокремлення найважливіших здобутків;
- Бачення необхідних змін для вирішення поточної ситуації.

Продукт:

- Висновки та пропозиції.

18.2.6. Написання аналітичного документу

18.2.6.1. Означення та характеристики аналітичного документу

Продуктом дослідження має стати аналітичний документ.

Аналітичний документ — це письмова робота, у якій:

- Збирають, систематизують і обробляють дані;
- Проводять глибокий аналіз ситуації, проблеми або явища;
- Формують висновки та розробляють практичні пропозиції чи рекомендації.

Основні характеристики аналітичного документа

1. Спирається на факти, дослідження і перевірені дані;
2. Має логічну, структуровану побудову;
3. Орієнтований на прийняття рішень або пропозицію рішень;
4. Подає інформацію об'єктивно, стисло й зрозуміло.

Результат та продукт етапу написання аналітичного документу для аналітичного дослідження

Результат:

- Фіксація мети, ходу, результатів, висновків та пропозицій дослідження у текстовій формі.

Продукт:

- Чернетка аналітичного документу.

18.2.6.3. Структура аналітичного документу та рекомендації щодо написання

Структура аналітичного документу

Рекомендовано наступну структуру аналітичного документу:

1. Назва, зміст і перелік скорочень;
2. Резюме;
3. Вступ;
4. Методологія дослідження;
5. Аналітичні розділи;
6. Висновки;
7. Пропозиції;
8. Додатки та використана література.

Назва має привертати увагу та складатися з ключових слів. Можливо використовувати запитання. Оптимальний обсяг – 3 – 5 слів. Заголовок обов'язково пов'язаний зі змістом дослідження.

Зміст виконує функцію путівника по тексту аналітичного документа. Він складається з нумерованих та ненумерованих розділів. Нумеровані розділи є взаємозалежними один від одного, ненумеровані – автономними частинами, які можуть самостійно існувати за межами аналітичного звіту. Рекомендується нумерувати розділи основної аналітичної частини та залишати без номеру резюме, вступ, методологію, висновки і пропозиції.

Резюме зацікавлює читача та містить стислі ключові тези. Цей розділ є скороченим синтезом висновків та пропозицій. Його обсяг має не перевищувати 1 – 2 сторінок. Інформація з резюме – це те, що читачі мають запам'ятати, навіть якщо вони не прочитають документ.

Вступ пояснює мотиви написання тексту, привертає увагу читача та знайомить з темою. Цей розділ важливо починати з потужного першого речення. Цінність вступу у поясненні наявного контексту. Його розмір не більше 10% від усього звіту. Оптимально – 1 сторінка.

У випадку роботи з даними написання методології дослідження є обов'язковим. У розділі можуть визначатися ключові терміни, описані типи даних та використані методи їх аналізу. Гарно написаною вважається методологія, при виконанні дій якої інша група дослідників приходить до ідентичного результату.

Аналітичні розділи складаються з опису результатів четвертого етапу дослідження. Це нумеровані та взаємозалежні блоки. Будь-який текст складається з абзаців. Класична теорія вчить, що абзац є комплексом з:

1. Першого головного речення;
2. Речення на підтримку (дані, твердження, приклади);
3. Пояснення чи обговорення тверджень;
4. Речення-підсумку (результат, пропозиція).

При написанні аналітичного документу важливо використовувати перші три складники абзацу та ситуативно четвертий. Для ефективного написання аналітичних звітів рекомендуємо читати актуальні монографії, статті з академічних журналів, звіти уряду чи міжнародних організацій, статистичні звіти та блоги науковців. Неможливо написати якісний текст, якщо його не читати.

Висновки й пропозиції вже сформовані на попередньому етапі дослідження.

У *додатки* переміщуються всі громіздкі дані, які не потрапили до аналітичних розділів. Цю інформацію можуть використати інші дослідники у своїй роботі.

Перелік використаної літератури підтверджує експертність автора звіту та свідчить про обсяг опрацьованої інформації.

Рекомендації щодо написання аналітичних документів

1. Формуючи текст, важливо розуміти, що це спосіб комунікації з цільовою аудиторією, платформа для вирішення проблеми, інструмент впливу та альтернатива класичному науковому дослідженню;
2. У звіту має бути короткострокова та довгострокова цілі.

Короткострокова відповідає предмету дослідження.

Довгострокова –закликає діяти.

18.2.6.3. Типові приклади аналітичних документів

Розглянемо основні типи аналітичних документів.

18.2.6.3.1. Аналітичні звіти

Аналітичний звіт (*Policy paper*) — це документ, який містить детальний аналіз певної проблеми, ситуації чи даних. Він може бути використаний для прийняття рішень, виявлення тенденцій, оцінки ефективності різних процесів чи проектів. Такий звіт зазвичай включає зібрану інформацію, її обробку та інтерпретацію, а також рекомендації чи висновки на основі результатів аналізу.

Аналітичний звіт може включати:

- *Опис проблеми чи ситуації.* Визначення мети дослідження та контексту;
- *Збір даних.* Використання різних методів для збору необхідної інформації;
- *Аналіз даних.* Обробка і вивчення отриманих даних, часто із застосуванням статистичних чи математичних методів;
- *Висновки та рекомендації.* На основі аналізу, пропонуються конкретні рішення чи заходи.

Вони можуть бути використані у багатьох сферах, таких як бізнес, економіка, наука, соціологія, політика, маркетинг тощо.

Наведемо приклад аналітичного звіту.

АНАЛІТИЧНИЙ ЗВІТ

Тема: Оцінка ефективності рекламної кампанії для нового продукту.

Дата: 27 квітня 2025 року.

1. Вступ.

Метою цього аналітичного звіту є оцінка ефективності рекламної кампанії, проведеної для запуску нового продукту — смартфона моделі $X-2025$. Звіт включає аналіз зібраних даних, таких як обсяги продажів, ефективність рекламних каналів і вплив на брендову відомість.

2. Збір даних.

Для аналізу були зібрані наступні типи даних:

- Дані про продажі за період з 1 по 20 квітня 2025 року;
- Оцінки ефективності рекламних кампаній (інтернет-реклама, телевізійні ролики, соціальні мережі);
- Відгуки та коментарі користувачів на платформах соціальних мереж;
- Статистика відвідувань вебсайту компанії та конверсії з реклами.

3. Аналіз даних.

3.1. *Обсяги продажів.*

Протягом першого тижня кампанії обсяг продажів збільшився на 15%, що свідчить про позитивний вплив рекламних матеріалів на покупців. У другому тижні збільшення становило 5%, що показує можливу насиченість ринку або зниження ефективності реклами.

3.2. *Аналіз каналів реклами.*

Аналіз основних каналів реклами є наступним:

- *Інтернет-реклама.* Статистика показує високий рівень конверсії через Google Ads і Facebook Ads. Приріст продажів через ці канали становить 30% від загального обсягу;
- *Телевізійні ролики.* Зниження ефективності через високу вартість рекламних роликів. Приріст продажів лише 10%;
- *Соціальні мережі.* Оцінка залученості користувачів через Instagram та Twitter показала високий рівень зацікавленості, однак ефективність конверсії була нижчою, ніж очікувалося (15%).

3.3. *Відгуки користувачів.*

Відгуки на новий продукт в основному позитивні, зокрема підкреслюють високу якість камери та швидкість роботи. Проте, є негативні відгуки щодо ціни та дизайну.

4. Висновки та рекомендації.

Висновки:

- Рекламна кампанія мала загальний позитивний ефект на продажі, але ефективність знизилася через деякі канали (телевізійні ролики);
- Інтернет-реклама та соціальні мережі стали найефективнішими каналами для залучення потенційних покупців;
- Відгуки користувачів вказують на високий рівень задоволення від функціональності продукту, але необхідно працювати над удосконаленням дизайну та зниженням ціни.

Рекомендації:

- Продовжити використання інтернет-реклами, зокрема таргетованої реклами на соціальних платформах;
- Переглянути стратегію рекламних роликів на телебаченні, знизити їх вартість або змінити підхід до творчого контенту;
- Використовувати відгуки користувачів для вдосконалення продукту, зокрема врахувати зауваження щодо дизайну та ціни.

5. Завершення.

Цей звіт надає основу для коригування маркетингових стратегій і допоможе покращити ефективність кампаній в майбутньому.

18.2.6.3.2. Аналітичні записки

Аналітична записка (*Policy brief*) — це документ, що містить короткий, але змістовний аналіз конкретної ситуації, проблеми чи питання, а також надають рекомендації або пропозиції для прийняття рішень. Основною її метою є забезпечення прийняття обґрунтованих рішень на основі зібраної та проаналізованої інформації.

Основні характеристики аналітичної записки:

1. *Короткість.* Записка зазвичай є лаконічною і не містить зайвої інформації. Вона зосереджена лише на суті питання та основних висновках;
2. *Аналіз.* Записка базується на детальному аналізі ситуації, даних, фактів або трендів, які мають важливе значення для організації або осіб, які приймають рішення;

3. *Рекомендації.* На основі аналізу в аналітичній записці зазвичай надаються рекомендації або пропозиції щодо вирішення проблеми чи покращення ситуації;
4. *Мета.* Метою записки може бути інформування керівництва або зацікавлених осіб про важливі аспекти діяльності, оцінка ризиків, розробка стратегії або внесення змін до процесів.

Аналітичні записки часто використовуються в таких сферах, як бізнес, політика, економіка, наука, управління проектами та державне управління.

Аналітичні записки зазвичай мають чітку структуру, орієнтуючись на швидке і зрозуміле донесення інформації до керівників чи інших зацікавлених осіб. Вони часто використовуються для оперативного реагування на ситуацію або для надання рекомендацій у конкретних сферах. Вони можуть містити наступні основні розділи:

1. *Вступ.* Коротке пояснення мети записки та важливості питання;
2. *Аналіз ситуації.* Оцінка поточного стану справ, факторів, що впливають на ситуацію;
3. *Висновки.* Підсумки з аналізу ситуації;
4. *Рекомендації.* Пропозиції щодо подальших дій або змін.

Аналітичні записки можуть бути використані для обґрунтування рішень, розробки стратегій, поліпшення процесів чи управлінських рішень.

Наведемо приклад аналітичної записки.

АНАЛІТИЧНА ЗАПИСКА

Тема: Оцінка ризиків впровадження нової технології в виробництві.

Дата: 27 квітня 2025 року.

Підготовлено: Аналітичний відділ.

1. Мета записки.

Метою цієї записки є оцінка потенційних ризиків, пов'язаних із впровадженням нової автоматизованої технології у виробничому процесі підприємства, а також пропозиції щодо їх мінімізації.

2. Оцінка ризиків.

2.1. *Технічний ризик.* Впровадження нової технології може спричинити проблеми з її інтеграцією у наявні виробничі процеси. Відсутність досвіду в роботі з новим обладнанням може призвести до збоїв у виробництві.

2.2. *Фінансовий ризик.* Високі витрати на закупівлю нового обладнання та навчання персоналу можуть перевищити заплановані бюджетні кошти, що створює фінансове навантаження на підприємство.

2.3. *Ризик для персоналу.* Впровадження нової технології може викликати опір серед працівників, особливо серед тих, хто не має достатньо кваліфікації для роботи з новим обладнанням. Це може спричинити зниження ефективності роботи на початкових етапах.

2.4. *Юридичний ризик.* Можливі правові питання, пов'язані з ліцензіями на нову технологію або із захистом інтелектуальної власності, якщо технологія належить сторонньому постачальнику.

3. Рекомендації.

3.1. *Пілотне тестування.* Рекомендується спочатку провести пілотний запуск технології на обмеженій частині виробництва для виявлення потенційних технічних проблем та коригування процесів.

3.2. *Навчання персоналу.* Необхідно розробити план навчання для працівників, який включатиме як теоретичні заняття, так і практичне освоєння нового обладнання.

3.3. *Фінансове планування.* Врахувати додаткові витрати на тестування, адаптацію та непередбачені ситуації у фінансовому плані, а також зберегти резервний фонд на випадок перевищення бюджетних витрат.

3.4. *Юридичний моніторинг.* Провести консультації з юридичним відділом щодо ліцензій та умов використання технології, а також перевірити, чи є потреба в укладанні додаткових угод з постачальниками.

4. Висновок.

Незважаючи на потенційні ризики, впровадження нової технології може значно підвищити ефективність виробництва. Рекомендується здійснити поетапне впровадження з урахуванням запропонованих заходів для мінімізації ризиків.

Підпис: [Ім'я аналітика]
Аналітичний відділ

18.2.6.3.3. Політичні або бізнес-огляди

Політичні або *бізнес-огляди* — це аналіз актуальних подій, трендів та ситуацій у політичній чи бізнесовій сферах, який спрямований на оцінку їх впливу, прогнозування можливих наслідків і надання рекомендацій для зацікавлених сторін (керівництва, політиків, інвесторів, громадськості).

Залежно від актуальності ситуацій, огляди можуть бути:

- Регулярними;
- Інцидентними.

Політичні огляди

Політичні огляди надають аналіз поточних політичних подій, законодавчих ініціатив, змін у внутрішній та зовнішній політиці, що можуть вплинути на стабільність держави, економіку або соціальні процеси.

Основні характеристики політичного огляду:

- *Аналіз політичної ситуації.* Огляд поточної політичної ситуації в країні чи на міжнародній арені. Це може включати аналіз виборчих процесів, політичних скандалів, реформ або змін у владних структурах.
- *Визначення ключових акторів.* Оцінка ролі основних політичних гравців, партій, лідерів та їхніх стратегій.
- *Оцінка впливу на економіку і суспільство.* Визначення того, як політичні події можуть вплинути на економічну ситуацію, бізнес-клімат або соціальну стабільність.

- *Прогнозування.* Прогнози щодо розвитку політичної ситуації, можливих виборів, реформ, змін у політичних альянсах тощо.

Наведемо приклад політичного огляду.

ПОЛІТИЧНИЙ ОГЛЯД

Тема: Оцінка впливу нових виборчих законів на результат парламентських виборів.

Дата: 27 квітня 2025 року.

1. Вступ.

Новий виборчий закон, що набирає чинності з наступних парламентських виборів, може мати значний вплив на результати виборів та політичну ситуацію в країні.

2. Основні зміни.

В результаті дослідження було сформульовано наступні зміни:

- Перехід до змішаної виборчої системи;
- Зменшення порогу для проходження партій до парламенту;
- Впровадження нових вимог до фінансування виборчих кампаній.

3. Оцінка впливу.

Сформульовано наступні оцінки впливу:

- Зміни можуть призвести до підвищення представництва малих партій;
- Змішана система збільшить конкуренцію між основними політичними силами, що призведе до зниження стабільності в парламенті;
- Фінансові обмеження можуть обмежити можливості для великих партій у проведенні кампаній.

4. Прогноз.

Згідно з новими змінами, можна очікувати більш поляризовані результати виборів, з посиленням ролі популістських партій.

Бізнес-огляди

Бізнес-огляди зосереджуються на аналізі економічних та бізнесових тенденцій, фінансових ринків, конкурентного середовища та інновацій, що можуть впливати на стратегію компаній чи галузей.

Основні характеристики бізнес-огляду:

- *Аналіз ринку.* Оцінка стану ринку, його тенденцій, нових можливостей та загроз;
- *Оцінка фінансових показників.* Аналіз результатів діяльності компаній, рентабельності, прибутковості та ефективності бізнес-процесів;
- *Прогнозування трендів.* Оцінка майбутніх тенденцій, таких як технологічні інновації, зміни у споживчих уподобаннях, зміни в міжнародній торгівлі;
- *Визначення стратегічних можливостей і загроз.* Виявлення можливих шляхів для розвитку бізнесу або компанії на основі поточних трендів та прогнозів.

Наведемо приклад бізнес-огляду.

БІЗНЕС-ОГЛЯД

Тема: Тенденції розвитку ринку електричних автомобілів у 2025 році

Дата: 27 квітня 2025 року

1. Вступ.

Ринок електричних автомобілів (ЕА) продовжує зростати, і в 2025 році очікується значне збільшення обсягів продажів. Підприємства в галузі активно інвестують у нові технології та розширення інфраструктури зарядних станцій.

2. Оцінка ринку.

В рамках оцінки ринку зроблено наступні висновки:

- Попит на електричні автомобілі зростає завдяки урядовим стимулюючим програмам і зниженню вартості технологій;
- Конкуренція серед виробників стає все більш інтенсивною, зокрема з боку нових гравців на ринку, таких як стартапи з Китаю та США.

3. Прогноз тенденцій.

Виявлено наступні тенденції:

- Збільшення долі електричних автомобілів на ринку через покращення технологій акумуляторів;
- Спостерігатиметься зростання попиту на електричні автомобілі серед молодших споживачів, що більше орієнтовані на екологічність;
- Розширення інфраструктури зарядних станцій сприятиме зменшенню бар'єрів для споживачів.

4. Рекомендації для компаній.

Сформульовано наступні аргументації:

- Інвестувати в розробку нових акумуляторних технологій для збільшення дальності поїздок;
- Розвивати партнерства з урядами для зниження вартості електричних автомобілів через субсидії або знижки.

Висновки

1. Політичні огляди допомагають оцінити вплив поточних політичних подій на економіку, стабільність або суспільний розвиток;
2. Бізнес-огляди дозволяють зрозуміти тенденції ринку, виявити конкурентні переваги та приймати обґрунтовані рішення щодо інвестицій, стратегії компаній чи галузей.

18.2.6.3.4. Прогнози і рекомендації для управлінців

Прогнози і рекомендації для управлінців — це елементи аналітичних документів (наприклад, записок, звітів чи оглядів), які допомагають керівникам приймати обґрунтовані рішення в умовах невизначеності, змін або ризиків.

Вони мають <u>дві основні функції</u>:

- *Прогнози* — описують можливі майбутні сценарії розвитку ситуації на основі аналізу трендів, даних або поточних подій;

- *Рекомендації* — пропонують конкретні дії або стратегії для досягнення бажаного результату або мінімізації ризиків.

Як виглядають прогнози:

- Ґрунтуються на аналітичних висновках із зібраної інформації;
- Містять кілька варіантів розвитку подій: оптимістичний, реалістичний і песимістичний сценарії;
- Оцінюють ймовірність настання кожного сценарію;
- Вказують, які чинники можуть змінити ситуацію.

Приклад прогнозу:

- За умови збереження поточних темпів інвестування у цифрову трансформацію, до кінця 2025 року очікується зростання продуктивності компанії на 15–20%. У випадку економічної рецесії цей показник може бути знижений до 5–7%.

Як виглядають рекомендації:

- Чітко орієнтовані на дії (actionable);
- Засновані на прогнозах і враховують потенційні ризики;
- Можуть містити кілька альтернативних варіантів дій;
- Вказують строки реалізації та бажані результати.

Приклад рекомендацій:

- Рекомендується вже зараз розпочати поетапне впровадження нових IT-рішень, починаючи з департаментів з найвищим рівнем навантаження. При цьому передбачити резервний бюджет на випадок непередбачених витрат у розмірі 10% від загальної суми інвестицій.

Структура блоку «Прогнози і рекомендації» в аналітичному документі:

- Короткий виклад ситуації (підстава для прогнозу);
- Прогноз розвитку подій (з варіантами сценаріїв);
- Рекомендовані заходи (з конкретними діями і термінами);
- Можливі ризики і способи їх мінімізації.

Наведемо приклад прогнозів та рекомендацій для управлінців.

Прогнози і рекомендації для управлінців: Вихід на ринок Східної Європи

1. *Короткий виклад ситуації.*

Компанія планує вийти на ринки Східної Європи (Польща, Чехія, Угорщина) з новою лінійкою екологічних товарів. Конкуренція на цих ринках висока, але спостерігається сталий попит на продукцію категорії «eco-friendly».

2. *Прогноз розвитку подій*:

2.1. *Оптимістичний сценарій (ймовірність 40%).*

Активний попит на екологічні товари забезпечить швидке зростання продажів на 25% протягом першого року. Партнерства з локальними рітейлерами зміцнять позиції бренду.

2.2. *Реалістичний сценарій (ймовірність 50%).*

Протягом першого року продажі зростатимуть поступово на рівні 10–15%. Вимагатиметься активна маркетингова підтримка для нарощування довіри до нового бренду.

2.3. *Песимістичний сценарій (ймовірність 10%).*

Через сильну конкуренцію та невідповідність продукту локальним споживчим очікуванням, ринок може виявитися важким для завоювання. Можливі втрати на рівні 5% інвестицій.

3. *Рекомендовані заходи.*

3.1. *Старт із пілотного проекту в Польщі.*

Запуск на одному ринку дозволить адаптувати продукт і маркетинг до локальних реалій перед масштабним розширенням.

3.2. *Проведення локальних маркетингових досліджень.*

Дослідити споживацькі вподобання і коригувати комунікаційні стратегії відповідно до культурних особливостей кожної країни.

3.3. *Побудова партнерств із місцевими мережами.*

Забезпечити розповсюдження продукції через відомі мережі супермаркетів і онлайн-платформи.

3.4. *Гнучкість в ціноутворенні.*

Передбачити можливість варіювання цін залежно від рівня конкуренції в різних країнах.

3.5. Моніторинг результатів кожні 3 місяці.

Відстежувати ключові показники ефективності (KPI) і вчасно коригувати стратегію при необхідності.

4. *Потенційні ризики і шляхи їх мінімізації.*

В результаті проведеного аналізу визначено наступні ризики:

- *Ризик недооцінки локальної конкуренції* — активна конкурентна розвідка і адаптація цінової політики;
- *Ризик репутаційних втрат через невдалий старт* — обмежити масштаби запуску на початковому етапі та працювати з лідерами думок для формування позитивного іміджу.

18.2.6.3.5. Резюме висновків і пропозицій

Резюме висновків і пропозицій (*Policy memo*) — це короткий узагальнений розділ аналітичного документа, у якому підсумовуються основні висновки аналізу та формулюються конкретні пропозиції щодо подальших дій.

Основні риси резюме висновків і пропозицій:

- *Лаконічність.* Стисле викладення головних ідей без зайвих деталей;
- *Чіткість.* Ясне розділення висновків (що встановлено) і пропозицій (що робити);
- *Орієнтація на рішення.* Фокус на практичних кроках і діях.

Наведемо приклад резюме висновків та пропозицій.

РЕЗЮМЕ ВИСНОВКІВ І ПРОПОЗИЦІЙ

Висновки:

- Східноєвропейський ринок демонструє високий потенціал для продукції категорії «eco-friendly», проте рівень конкуренції там суттєвий;
- Шанси на успіх найбільші при поступовому входженні на ринок із локалізацією продукту і маркетингових стратегій;
- Основні ризики пов'язані з нерозумінням локальних споживацьких потреб і сильною конкуренцією.

Пропозиції:

- Розпочати із запуску в Польщі як тестовому ринку;
- Інвестувати в дослідження локальних споживачів перед масштабним розширенням;
- Вибудовувати партнерства з локальними рітейлерами для швидшого виходу на ринок;
- Застосовувати гнучку політику цін залежно від специфіки кожної країни;
- Регулярно моніторити результати і оперативно адаптувати стратегію.

18.2.7. Рецензування аналітичного документу

Чернетка аналітичного документу має пройти рецензування двох видів:

- Внутрішнє;
- Зовнішнє.

Внутрішня рецензія – це рецензія від фахівців та керівництва організації. Важливо бути конструктивними. Рекомендації мають покращити текст. Формуйте їх закликом до дії. Слід обмежити використання таких критичних слів, як: незрозуміло, неправильно тощо.

Зовнішня рецензія – це рецензія, яку слід замовляти у фахівців галузі з багаторічним досвідом, національних експертів та лідерів громадської думки.

Рекомендуємо рецензувати з точки зору:

- Новизни матеріалу;
- Доступності основної ідеї потенційному читачеві;
- Логіки побудови тексту;
- Переконливості аргументації;
- Повноти аналізу;
- Стилістичної доречності (відсутності оціночних суджень, прийомів художнього письма тощо);
- Практичності пропозицій.

Результат та продукт етапу рецензування аналітичного документу для аналітичного дослідження

Результат:

- Оцінка чернетки аналітичного документу;
- Набір пропозицій щодо її удосконалення.

Продукт:

- Внутрішня та зовнішня рецензії аналітичного документу.

18.2.8. Кінцеве доопрацювання аналітичного документу

Кінцеве доопрацювання аналітичного документу — це завершальний етап роботи над аналітичним текстом, який включає перевірку змісту, структури, стилю і оформлення, усунення неточностей та приведення документа у відповідність до вимог замовника або стандартів.

Автори звіту можуть врахувати або прийняти до відома надані рецензії. У першому випадку вносять зміни до звіту, у другому – чернетка звіту вважається готовим продуктом.

Що включає кінцеве доопрацювання:

- Перевірка логіки викладу: чи чітко простежується причинно-наслідковий зв'язок між розділами;
- Уточнення формулювань: усунення двозначностей, заміна складних конструкцій простішими;
- Узгодження висновків і рекомендацій: щоб висновки логічно випливали з аналізу, а рекомендації — з висновків;
- Перевірка фактів і даних: виправлення можливих помилок або неточностей у цифрах, іменах, назвах тощо;
- Редагування стилю і мови: дотримання принципів офіційно-ділового стилю (або іншого, залежно від вимог);
- Оформлення: перевірка заголовків, нумерації, списків, таблиць, цитувань.

Мета кінцевого доопрацювання — зробити документ:

- Професійним і зрозумілим;
- Переконливим для аудиторії;
- Готовим до презентації або передання керівництву/замовнику.

Слід враховувати, що людський мозок розуміє картинку у 65 тис. разів швидше ніж текст. Більшість населення (80%) за своєю природою краще сприймають зображення. Відтак якісно написаний звіт – це лише половина справи. Потрібна візуалізація! Крім того, не стандартна, а саме аналітична.

У аналітичних центрах нерідко є постійні дизайнери. Вони працюють у парі з аналітиками. Тобто мають достатній досвід та знання як виділяти найважливіші дані, який вид графіки обрати для тої чи іншої цифрової інформації.

Після візуалізації аналітичний документ надсилається замовнику.

Результат та продукт етапу кінцеве доопрацювання аналітичного документу для аналітичного дослідження

Результат:

- Удосконалення аналітичного документу.

Продукт:

- Аналітичний документ.

18.2.9. Поширення результатів

Поширення результатів — це процес доведення підготовленого аналітичного матеріалу (висновків, прогнозів, рекомендацій) до цільової

аудиторії з метою інформування, підтримки прийняття рішень або формування громадської думки.

Основні елементи поширення результатів:

- *Вибір цільової аудиторії.* Визначення, хто саме має отримати інформацію (керівники, партнери, громадськість, експерти тощо).
- *Форма подачі.* Вибір формату — аналітична записка, презентація, прес-реліз, стаття, брифінг тощо.
- *Канали розповсюдження.* Офіційні листи, публікації на сайті, розсилки електронною поштою, публічні виступи, ЗМІ.
- *Тон і стиль комунікації.* Адаптація стилю подачі під очікування аудиторії (офіційний, популярний, візуалізований).

Мета поширення результатів — забезпечити правильне розуміння аналітичних висновків і сприяти реалізації запропонованих заходів.

Платформи для розповсюдження результатів аналізу:

- Публічні заходи (круглі столи, обговорення, дискусії, конференції, форуми тощо);
- Ділові зустрічі зі стейкхолдерами;
- Прес-конференції та брифінги;
- Прес-матеріали для ЗМІ;
- Соціальні мережі (поширення інфографіки та відеоінфографіки).

Рекомендації щодо поширення результатів

1. Без ефективного поширення результатів роботи аналітиків не відбудеться інформування цільової аудиторії.
2. Варто ретельно продумати медійний план: яка інформація, для кого і яким чином буде поширена.
3. Слід враховувати, що аналітика – це аргументована форма впливу на суспільство, тому дуже важливо, щоб контент був якісним і адаптованим для цільової аудиторії.
4. Слід залучати всі медіа-інструменти: ЗМІ, роздаткові матеріали, соцмережі, месенджери і візуальні повідомлення для груп впливу.

Результат та продукт етапу поширення результатів для аналітичного дослідження

Результат:

- Цільова аудиторія отримала інформацію про результати аналітичного дослідження, обговорила та використала їх у своїй діяльності.

Продукт:

- Статті в ЗМІ, інфографіка, відеоінфографіка.

18.3. Технічне завдання на аналітику

Отже, якщо потрібна аналітика, необхідно дотримуватися декількох практичних порад та питань, які допоможуть порозумітися з аналітиками. Відповіді на них стануть повноцінним технічним завданням.

1. Яка тема Вашого дослідження? До якої сфери вона належить? Чи аналізували її раніше? Опишіть контекст. Надайте власне розуміння.

2. Які Ваші гіпотези?
3. Яка Ваша мета? Чого ви хочете досягти зараз і в майбутньому? Як плануєте використовувати результати? Яких змін прагнете?
4. Вкажіть часові рамки та географію дослідження.
5. Хто має отримати користь від результатів дослідження? Хто прочитає звіт? Назвіть стейкхолдерів та цільову аудиторію. Хто є Вашим прихильником? Противником?
6. Яким має бути продукт аналітичного дослідження: інфографіка, коротка довідка (policy memo), аналітична презентація, записка (policy brief), звіт (policy paper) тощо. Який бажаний сторінковий обсяг?
7. Хто є лідером громадської думки у обраній сфері? Чия рецензія додасть бонусів аналітичному документу?
8. Зафіксуйте необхідність погодження з Вами чернетки аналітичного продукту та остаточного затвердження готового документу. Пам'ятайте про міру.
9. Як плануєте поширювати результати дослідження? Чи має Ваша цільова аудиторія спеціалізовані ЗМІ?
10. Чи потрібна верстка та візуалізація продуктів дослідження? Як щодо аналітичних статей для ЗМІ?
11. Обговоріть авторські права на продукти дослідження. Домовтеся про передачу Вам усіх отриманих у ході дослідження первинних даних та файлів із розрахунками.
12. Вкажіть дедлайн роботи на місяць раніше, ніж він є насправді.

18.4. Приклад реалізації аналітичного дослідження в Python

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Створюємо прикладовий датафрейм
data = {
    'Години_навчання': [2, 4, 1, 5, 3, 6, 2, 7, 3,
8],
    'Відвідуваність_%': [60, 80, 50, 90, 70, 95, 55,
100, 65, 98],
    'Успішно_склав': [0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
# 1 — здав, 0 — ні
}

df = pd.DataFrame(data)
```

```
# Візуалізація
sns.scatterplot(x='Години_навчання',
y='Відвідуваність_%', hue='Успішно_склав', data=df)
plt.title("Навчання та ймовірність складання іспиту")
plt.show()

# Підготовка до моделювання
X = df[['Години_навчання', 'Відвідуваність_%']]
y = df['Успішно_склав']

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=42)

# Логістична регресія
model = LogisticRegression()
model.fit(X_train, y_train)

# Прогнозування
y_pred = model.predict(X_test)

# Звіт
print("Оцінка моделі:")
print(classification_report(y_test, y_pred))
```

18.5. Українська наука 2019 аналітичне дослідження

Скільки в Україні науковців, у яких сферах вони працюють, як фінансують їхню роботу держава та бізнес, якими є результати їхніх досліджень, зокрема, в міжнародних порівняннях? Цю та багато іншої важливої статистичної інформації про українську науку можна знайти в аналітичній доповіді: «Наукова та науково-технічна діяльність в Україні у 2019 році», що була розміщена на сайті МОН сьогодні, 13 серпня 2020 року.

Зокрема, торік на наукові дослідження і розробки в Україні з усіх джерел було витрачено 17 млрд 254 млн грн. Найбільшу частину профінансувала держава – 38,3%. Частка коштів вітчизняних замовників становила 28,1%, іноземних джерел – 22,3%, власних коштів – 10%, інших джерел 1,3% (рис. 18.7).

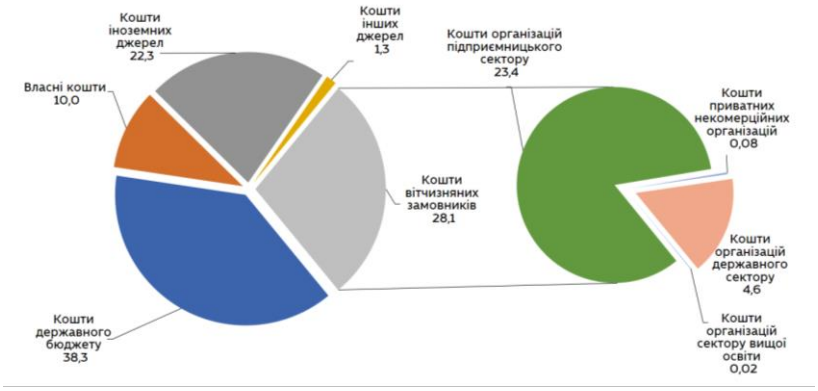


Рис. 18.7. Розподіл загального обсягу витрат на виконання ДІР у 2019 р.

Водночас наукоємність ВВП, як і в попередні роки, залишилася критично низькою. 2019 року цей показник становив 0,43% (рекордний мінімум за останні 10 років), а за рахунок коштів держбюджету – 0,17%.

Для порівняння: за даними 2018 року, наукоємність ВВП країн ЄС-28 у середньому становила 2,12% (рис. 18.8).

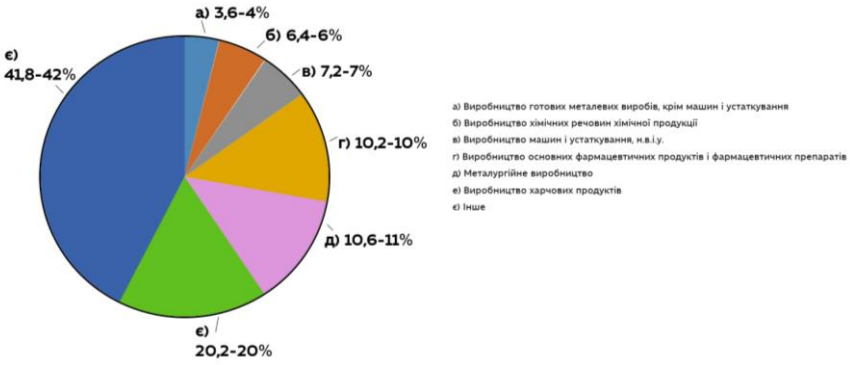


Рис. 18.8. Витрати на інноваційну діяльність

З держбюджету наукова сфера фінансувалася 20-ма головними розпорядниками коштів через 49 програм. У їхніх межах було профінансовано 9 млрд 312 млн грн, з яких понад 70% – це загальний фонд. Більшість коштів загального фонду (понад 80%) були спрямовані на дослідження і розробки в цілому. Ще 7,6% пішло на підтримку розвитку наукової інфраструктури та оновлення матеріально-технічної бази, а 12,2% – на інші напрями (рис. 18.9).

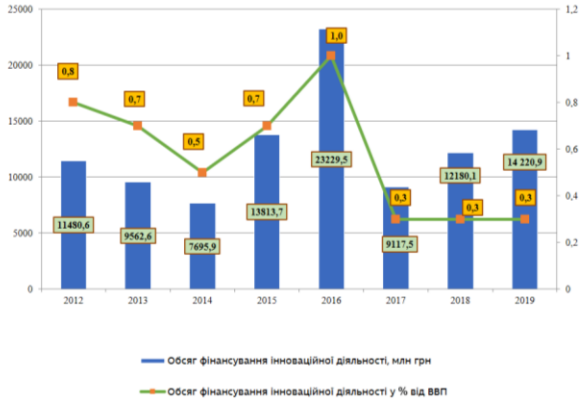


Рис. 18.9. Фінансування інноваційної діяльності

Одним із найважливіших показників ефективності використання бюджетних коштів з фінансування досліджень і розробок є рівень впровадження наукової (науково-технічної) продукції.

Так, з 19453 одиниць продукції, створеної за кошти держбюджету, впроваджено майже 76%. Але при цьому рівень впровадження продукції, створеної за рахунок загального фонду, становить 70,6%, а створеної за рахунок спецфонду – майже 90%. Така ж ситуація спостерігається і за всіма видами продукції. Наприклад, в категорії «Матеріали» різниця між впровадженням за кошти загального і спецфонду становить понад 40% (рис. 18.10).

Рис. 18.10. Фінансування інноваційної діяльності

Докладніша інформація про інноваційну діяльність в Україні 2019 року представлена в окремому дослідженні, що оприлюднено на сайті МОН.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

Основи аналізу даних

1. Що таке аналіз даних?
2. Які основні етапи аналізу даних?
3. У чому різниця між структурованими та неструктурованими даними?
4. Які основні джерела отримання даних?
5. Що таке «брудні дані» і як їх очищувати?
6. Які методи потрібні для обробки пропущених значень у датасеті?
7. Що таке дискретні та неперервні зміни?
8. Які метрики вибираються для оцінки якості даних?
9. Що таке нормалізація та стандартизація даних?
10. Чим відрізняється описова та інтерференційна статистика?

Статистика в аналізі даних

11. Що таке середнє, медіана та мода?
12. Що таке дисперсія та стандартне відхилення?
13. Як розраховується міжквартильний розмах (IQR)?
14. Що таке закон великого чисел?
15. Які основні типи розподілів у статистиці?
16. Що таке нормальний розподіл і чому він важливий?
17. Що таке закон Пуассона і де його застосувати?
18. Як інтерпретувати значеннясторсторстор- вартість?
19. Що таке довірчий інтервал?
20. Що таке коефіцієнт кореляції Пірсона та Спірмена?

Машинне навчання та моделювання

21. Які основні типи машинного навчання?
22. У чому різниця між наглядовим і ненаглядовим навчанням?
23. Що таке моделювання даних?
24. Які метрики вибираються для оцінки якості моделі?
25. Що таке переобучення (overfitting) і як з ним боротися?
26. Як працює крос-валідація?
27. Що таке регулярізація та які її основні типи?
28. Чим відрізняється лінійна регресія від логістичної?
29. Що таке метрика точності, повноти та F_1-міри?
30. Як працює метод опорних векторів (SVM)?

Обробка та візуалізація даних

31. Що таке EDA (Exploratory Data Analysis)?
32. Які основні методи візуалізації даних?
33. Як обрати відповідний тип графіка для даних?
34. Що таке boxplot і для чого він використовується?
35. Як побудувати матрицю кореляції?
36. Що таке гістограма і чим вона відрізняється від стовпчикової діаграми?

37. Алгоритми та методи аналізу даних.
38. Що таке кластеризація і як її візуалізувати?
39. Що таке алгоритм k-means?
40. Як працює алгоритм DBSCAN?
41. Як працює алгоритм OPTICS?
42. Що таке метод головних компонентів (PCA)?
43. Як працює алгоритм APRIORI при пошуку асоціацій?
44. Як працює алгоритм ECLAT при пошуку асоціацій?
45. Як працює алгоритм FP-GROWTH при пошуку асоціацій?
46. Чим відрізняється ієрархічна кластеризація від k-means?
47. Що таке LDA (Latent Dirichlet Allocation)?
48. Як працює метод випадкового лісу (RANDOM FOREST)?
49. Чим відрізняється XGBoost від звичайного градієнтного бустингу?
50. Що таке підсилене навчання (Reinforcement Learning)?
51. Які основні метрики кластеризації?

Аналіз часових рядів

52. Що таке часовий ряд?
53. Які основні компоненти часового ряду?
54. Що таке автокореляція?
55. Як працює метод ковзного середнього?
56. Що таке ARIMA-модель?
57. Що така сезонність у часових рядах?
58. Як працює експоненціальне згладжування?
59. Що таке трендовий аналіз?
60. Як оцінити точність прогнозу для часових рядів?
61. Що таке стохастичні процеси?

Великі дані та робота з ними

62. Що таке Big Data?
63. Які основні характеристики великих даних ($3V$)?
64. Що таке розподілені обчислення?
65. Як працює Hadoop?
66. Що таке MapReduce?
67. Які бази даних резервуються для роботи з Big Data?
68. Як працює Apache Spark?
69. Що бере NoSQL і в чому його переваги?
70. Як вибрати індекси в базі даних?
71. Що таке поточна обробка даних (Stream Processing)?

Аналіз текстових даних (NLP)

72. Що таке NLP (Natural Language Processing)?
73. Як працює токенізація?
74. Що таке лемматизація та стемінг?
75. Як працює модель Word2Vec?
76. Що таке TF-IDF і для чого він використовується?

77. Як обробляти стоп-слова в тексті?
78. Що таке аналіз тональності тексту?
79. Як працюють трансформери у NLP?
80. Що таке неймовірні моделі в аналізі тексту?
81. Як оцінити якість NLP -моделі?

Байєсівська статистика та A/B-тести

82. Що таке байєсівський підхід у статистиці?
83. Як працює Байєсівська теорема?
84. Що таке A/B-тестування?
85. Які основні помилки при проведенні A/B-тестів?
86. Що таке ефект відбору при тестуванні?
87. Що таке MAB (Multi-Armed Bandit)?
88. Як інтерпретувати результати A/B-тесту?
89. Як вибрати розміри для експерименту?
90. Що такий ефект новини в тестах?
91. Як уникнути гібнопозитивних результатів?

Етичні аспекти та бізнес-аналітика

92. Що таке етика аналізу даних?
93. Які основні ризики роботи з персональними даними?
94. Що таке GDPR і як він впливає на аналіз даних?
95. Як дані можуть бути використані в шахрайстві?
96. Як забезпечити анонімність даних?
97. Яка роль держави у сфері Data Science?
98. Що таке BI (Business Intelligence)?
99. Як дані використані для ухвалення бізнес-рішень?
100. Що таке KPI і як їх аналізувати?
101. Як оцінити економічний ефект від аналізу даних?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. David S. Moore, George P. McCabe, Bruce A. Craig // Introduction to the practice of statistics / W. H. Freeman and Company, pp. 710, 2009. ISBN 978-1-4292-1623-4. DOI: 10.1017/S0025557200003399

2. Trevor Hastie, Robert Tibshirani, Jerome Friedman. // Data Mining, Inference, and Prediction, Second Edition / Springer Science+Business Media, LLC, part of Springer Nature 2009. DOI: 10.1007/978-0-387-84858-7.

3. Christopher M. Bishop // Pattern Recognition and Machine Learning / Springer, New York, 2006. DOI: 10.1007/978-0-387-45528-0.

4. Christophe Giraud-Carrier, O. Povel // Characterising Data Mining software / Intelligent Data Analysis, 7(3):181-192, 2003. DOI: 10.3233/IDA-2003-7302.

5. Chickering D, Geiger D., Heckerman D. Learning // Bayesian networks: The combination of knowledge and statistical data Machine Learning. / Machine Learning, 20(3), 197-243, 1995. DOI:10.1007/BF00994016.

6. Heckerman D. // Bayesian Networks for Data Mining / Data Mining and Knowledge Discovery, № 1, P. 79-119. 1997. DOI: 10.1023/A:1009730122752.

7. Christopher Pal, Mark Hall, Eibe Frank, Ian Witten // Data Mining: Practical Machine Learning Tools and Techniques, 4rd ed. / Morgan Kaufmann, 2017. DOI: 10.1016/C2015-0-02071-8.

8. Jennifer Reese, Richard Reese // Java for Data Science / Packt Publishing, P. 336, 2017. ISBN: 1785281240, 9781785281242. URL: https://books.google.com.ua/.

9. Bostjan Kaluza// Machine Learning in Java / Packt Publishing, P. 258, 2016. ISBN: 1784396583, 9781784396589. URL: https://books.google.com.ua/.

10. Jiawei Han, Micheline Kamber and Jian Pei // Data Mining: Concepts and Techniques, 3rd ed / Morgan Kaufmann, 2012. DOI: 10.1016/C2009-0-61819-5.

11. Foster Provost, Tom Fawcett // Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking / O'Reilly Media, Inc., P. 414, 2013. URL: https://books.google.com.ua/.

12. Wes McKinney // Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Ipython / O'Reilly Media, Inc., P. 449, 2012. URL: https://books.google.com.ua/.

13. Hadley Wickham and Garrett Grolemund // R for Data Science / O'Reilly Media, Inc., P. 518, 2017. URL: https://r4ds.had.co.nz/.

14. Aurélien Géron // Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow / O'Reilly Media, Inc., P. 821, 2019. URL: https://books.google.com.ua/.

15. Richard Arnold Johnson, Dean W. Wichern // Applied Multivariate Statistical Analysis / Pearson Prentice Hall, P. 773, 2007. URL: https://books.google.com.ua/.

16. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani // An Introduction to Statistical Learning / Springer New York, NY, P. 607, 2021. DOI: 10.1007/978-1-0716-1418-1.
17. Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, Donald B. Rubin // Bayesian Data Analysis, 3rd ed / CRC Press, P. 645, 2015. URL: https://books.google.com.ua/
18. Гавриленко О.В. // Навчальний посібник з дисциплін «Аналіз даних» та «Аналіз даних в управляючих системах» для студентів спеціальності 126 – Інформаційні системи та технології / Київ: КПІ ім. Ігоря Сікорського, 85 с., 2020. URL: https://ela.kpi.ua/.
19. Гавриленко О.В., Онищенко В.В. // Задачі Аналізу даних в Теорії прийняття рішень. Практикум. Частина 1 [Електронний ресурс]: навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 126 Інформаційні системи та технології / Київ: КПІ ім. Ігоря Сікорського, 2024, 85 с. URL: https://ela.kpi.ua/.
20. Гавриленко О.В. // Аналіз даних в інформаційно-управляючих системах. Курс лекцій [Електронний ресурс]: навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 126 Інформаційні системи та технології / Київ: КПІ ім. Ігоря Сікорського, 2023, 205 с. URL: https://ela.kpi.ua/.
21. Гавриленко О.В., Мягкий М.Ю. // Аналіз даних в інформаційно-управляючих системах. Лабораторний практикум [Електронний ресурс]: навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 126 Інформаційні системи та технології / Київ: КПІ ім. Ігоря Сікорського, 2024, 81 с. URL: https://ela.kpi.ua/.
22. W. J. Ewens, K. Brumberg // Introductory Statistics for Data Analysis / Springer, P. 273, 2023. DOI: 10.1007/978-3-031-28189-1.
23. B. Blaine // Introductory Applied Statistics / Springer, P. 190, 2023, DOI: 10.1007/978-3-031-27741-2.
24. Доценко С.І., Онищенко В.В. // Математичне програмування: Навчально-методичний посібник для студентів денної та заочної форми навчання за напрямком 0502 – менеджмент /Київ: ДУІКТ, 2003, 31 с. URL: https://studfile.net/.
25. Han, J., Kamber, M., & Pei, J. // Data Mining: Concepts and Techniques (3rd ed.) / Morgan Kaufmann, 2011. URL: https://shop.elsevier.com/
26. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. // Data Mining: Practical Machine Learning Tools and Techniques (4th ed.) / Morgan Kaufmann, 2016. URL: https://shop.elsevier.com/.
27. Tan, P.-N., Steinbach, M., & Kumar, V. // Introduction to Data Mining (2nd ed.) / Pearson, 2018. URL: https://www.pearson.com/.
28. Jennifer Reese, Richard Reese // Java for Data Science / Packt Publishing, 2017. URL: https://www.packtpub.com/.
29. Bostjan Kaluza. // Machine Learning in Java / Packt Publishing, 2016. URL: https://github.com/.

30. Leskovec, J., Rajaraman, A., & Ullman, J. D. // Mining of Massive Datasets (2nd ed.) / Cambridge University Press, 2014. URL: https://www.cambridge.org/.
31. Aggarwal, C.C. // Data Mining: The Textbook / Springer, 2015. URL: https://charuaggarwal.net/.
32. Hastie, T., Tibshirani, R., & Friedman, J. // The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.) / Springer, 2009. URL: https://freecomputerbooks.com/.
33. Grus, J. // Data Science from Scratch: First Principles with Python (2nd ed.) / O'Reilly Media, 2019. URL: https://archive.org/.
34. Müller, A. C., & Guido, S. // Introduction to Machine Learning with Python: A Guide for Data Scientists / O'Reilly Media, 2016. URL: https://books.google.com.ua/.
35. Hyndman Rob J., Koehler Anne B., Ord, / Keith, Snyder Ralph D. Forecasting with Exponential Smoothing: The State Space Approach. Springer- Verlag Berlin Heidelberg, 2008. P. 76. URL: https://link.springer.com/.
36. O. Valenzuela, F. Rojas, Luis J. Herrera, H. Pomares and I. Rojas // Theory and Applications of Time Series Analysis and Forecasting / Springer, P. 333, 2023, DOI: 10.1007/978-3-031-14197-3.
37. Альперт М.І., Онищенко В.В. // Пошук найкращого алгоритму найкоротшого шляху для розумної валізи / Управління розвитком складних систем, 55, 2023, 92–97. DOI: https://doi.org/10.32347/2412-9933.2023.55.92-97.
38. Гавриленко О.В., Жураковська О.С., Коган А.В., Матвійчук Р.М., Піскун А.М., Хавікова Ю.І., Халус О.А. // Принцип формування портфелю публічних (адміністративних) послуг на основі аналізу статистичної інформації // Східно-Європейський журнал передових технологій, №3 (117), 2022. DOI: https://doi.org/10.15587/1729-4061.2022.260136.
39. Sergii Telenyk, Grzegorz Nowakowski, Olena Gavrilenko, Mykhailo Miahkyi, Olena Khalus // An analysis of the influence of famous people's posts on social networks on the cryptocurrency exchange rate / Bulletin of the polish academy of sciences technical sciences, Vol. 72(4), 2024. DOI: 10.24425/bpasts.2024.150117.
40. Petro Bidyuk, Olena Gavrilenko, Mykhailo Myagkyi // The algorithm for predicting the cryptocurrency rate taking into account the influence of posts of a group of famous people in social networks // System research and information technologies, No.2, 2023. DOI: https://doi.org/10.20535/SRIT.2308-8893.2023.2.02.
41. Gavrilenko, O., & Myagkyi, M. // Analysis of communities and groups in social networks as a significant factor of influence on cryptocurrency rates. / Innovative technologies and scientific solutions for industries, 1 (27), 2024, pp. 18–25. DOI: https://doi.org/10.30837/ITSSI.2024.27.018.
42. Гавриленко О.В., Новаківська К.Д., Шумейко О.А. // Вибір найбільш впливових економічних факторів для прогнозування курсу долару США. /

Вісник НТУ №54. – К.: НТУ, 2022. с.26-35. DOI: 10.33744/2308-6645-2022-4-54-026-035 .
43. В. Онищенко, Є. Власюк // Автоматизований спосіб визначення користувачів продуктів даних в розподіленій системі сіток даних / Адаптивні системи автоматичного управління, Том 2 №43, 2023. DOI: https://doi.org/10.20535/1560-8956.43.2023.292261.
44. А. Іванов, В. Онищенко // Методи генерації зображень з використанням мереж GAN / Адаптивні системи автоматичного управління, Том 1, №42, 2023. DOI: https://doi.org/10.20535/1560-8956.42.2023.279109.
45. Гавриленко О.В., Процюк Ю.В. // Approaches to the solution to the problem of news-based events forecasting / Адаптивні системи автоматичного управління, 2022, с. 65-70. DOI: https://doi.org/10.20535/1560-8956.40.2022.261651.
46. Olena Gavrilenko; Oleksandr Khomenko; Oksana Zhurakovska; Alla Kohan; Andrii Piskun; Olena Khalus // Application of association rules for formation of public (administrative) services portfolio / «Сучасні інформаційні системи» Том 6, №4, с.63-68, 2022. DOI: https://doi.org/10.20998/2522-9052.2022.4.09.
47. О. Гавриленко, М. Мягкий // Алгоритм прогнозування курсу криптовалюти з урахуванням впливу ранжованої групи експертів в соціальних мережах / Адаптивні системи автоматичного управління, Том 1, №46, 2025, с. 145-159. DOI: https://doi.org/10.20535/1560-8956.46.2025.323713.
48. Gavrilenko, O., & Myagkyi, M. // Forecasting the cryptocurrency exchange rate based on the ranking of expert opinions / Innovative technologies and scientific solutions for industries, 4 (26), 2023, pp. 24–32. DOI: https://doi.org/10.30837/ITSSI.2023.26.024.
49. О. Гавриленко, К. Фещенко // Виявлення пропаганди в потоках новин / Адаптивні системи автоматичного управління, Том 1, №46, 2025, с. 145-159. DOI: https://doi.org/10.20535/1560-8956.46.2025.323759.
50. О. Гавриленко, А. Бистріцький, І. Толкунов, Н. Богданова // Аналіз сучасних методів реферування текстів / Адаптивні системи автоматичного управління, Том 1, №46, 2025, с. 46-54. DOI: https://doi.org/10.20535/1560-8956.46.2025.323679.
51. Д. Митник, О. Гавриленко, Н. Богданова // Архітектура програмного забезпечення для створення психологічного портрету людини на основі активності в соціальних мережах / Адаптивні системи автоматичного управління, Том 1, №46, 2025, с. 83-96. DOI: https://doi.org/10.20535/1560-8956.46.2025.323690.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Moodle https://do.ipo.kpi.ua/course/view.php?id=5470.
2. «Електронний Кампус КПІ» https://ecampus.kpi.ua./