

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

«На правах рукопису»  
УДК 004.8+004.9+303.732.4

ДО ЗАХИСТУ ДОПУЩЕНО  
Завідувач кафедри ММСА  
\_\_\_\_\_ Оксана ТИМОЩУК  
«\_\_» \_\_\_\_\_ 2025 р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Системний аналіз фінансового ринку»  
зі спеціальності 124 «Системний аналіз»  
на тему: «Рекомендаційні системи з використанням текстів оглядів  
користувачів»**

Виконав:

Студент II курсу, групи КА-42мп  
Артеменко Євгеній Вячеславович \_\_\_\_\_

Науковий керівник:

Професор кафедри ММСА, доктор технічних наук, доц.  
Недашківська Надія Іванівна \_\_\_\_\_

Консультант з нормоконтролю:

доцент кафедри ММСА, к.ф.-м.н.,  
Статкевич Віталій Михайлович \_\_\_\_\_

Рецензент:

Професор кафедри ШІ, доктор технічних наук, проф.  
Данилов Валерій Якович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ  
Рівень вищої освіти – другий (магістерський)  
Спеціальність, спеціалізація 124 «Системний аналіз»  
Освітньо-професійна програма «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ  
Завідувач кафедри ММСА  
\_\_\_\_\_ Оксана ТИМОЩУК  
«\_\_\_» \_\_\_\_\_ 2025 р.

### **ЗАВДАННЯ**

**на магістерську дисертацію студенту**  
**Артеменку Євгенію Вячеславовичу**

1. Тема дисертації «Рекомендаційні системи з використанням текстів оглядів користувачів»  
науковий керівник дисертації Недашківська Надія Іванівна, професор кафедри ММСА, доктор технічних наук, доц, затверджені наказом по університету від «06» листопада 2025 р. № 4837-с.
2. Строк подання студентом дисертації \_\_12.2025
3. Об'єкт дослідження прогнозування рейтингів користувачів на основі текстових оглядів користувачів та металаних товарів, використовуючи рекомендаційну систему
4. Предмет дослідження моделі та методи SVDpp, CoClustering, KNNBaseline, SlopeOne, TF-IDF, Word2Vec, SBERT, Random Forest, XGBoost, Linear Regression, VADER, TextBlob
5. Перелік завдань, які потрібно розробити
  - 1) огляд предметної області;
  - 2) відбір та аналіз інструментів машинного навчання для практичної реалізації;
  - 3) аналіз та візуалізація дослідницьких даних;
  - 4) розробка моделі вирішення задачі прогнозування рейтингів;
  - 5) порівняльний аналіз отриманих моделей, аналіз результатів, оцінка їх якості та відбір найефективніших
6. Перелік ілюстративного матеріалу
  - 1) рисунки;
  - 2) таблиці;
  - 3) презентація
7. Орієнтовний перелік публікацій  
Артеменко С.В., Недашківська Н.І. Рекомендаційні системи з використанням текстів оглядів користувачів та ансамблювання на

основі стекингу. Системні науки та інформатика: збірник доповідей IV Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 1–5 грудня 2025 року, Київ. Київ: НН ІПСА КПІ ім. Ігоря Сікорського. 2025. 5 с.

8. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 01 вересня 2025 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації (МД)	Термін виконання етапів роботи	Примітка
1	Ознайомлення зі структурою МД згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	01.09.2025 – 20.09.2025	Виконано
2	Ознайомлення з ДСТУ 3008: 2015	20.09.2025 – 05.10.2025	Виконано
3	Проведення дослідження за темою МД під керівництвом керівника	05.10.2025 – 15.10.2025	Виконано
4	Завершення роботи над першим варіантом частини МД	15.10.2025 – 25.10.2025	Виконано
5	Проведення роботи над експериментальною частиною МД	25.10.2025 – 10.11.2025	Виконано
6	Проведення роботи над програмним продуктом	10.11.2025– 30.11.2025	Виконано
7	Оформлення МД та аналіз отриманих результатів	01.12.2025 – 15.12.2025	Виконано

Студент \_\_\_\_\_ Євгеній АРТЕМЕНКО

Науковий керівник дисертації \_\_\_\_\_ Надія НЕДАШКІВСЬКА

## РЕФЕРАТ

Магістерська дисертація: 82 с., 16 рис., 35 табл., 1 додаток, 21 джерело.

РЕКОМЕНДАЦІЙНА СИСТЕМА, ГІБРИДНА ФІЛЬТРАЦІЯ, ВЕКТОРИЗАЦІЯ ТЕКСТІВ, СЕНТИМЕНТ-АНАЛІЗ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ

Об'єкт дослідження – прогнозування рейтингів користувачів на основі текстових оглядів користувачів та метаданих товарів.

Предмет дослідження – моделі колаборативної фільтрації (SVDpp, CoClustering, KNNBaseline, SlopeOne), методи векторизації тексту (TF-IDF, Word2Vec, SBERT), методи sentiment-аналізу (VADER, TextBlob), моделі регресії (Random Forest, XGBoost, Linear Regression), ансамблева гібридна рекомендаційна система на основі стекінгу.

Мета дослідження – розробити ансамблеву гібридну рекомендаційну систему на основі стекінгу, порівняти вплив нових ознак на формування прогнозів, порівняти результати прогнозів базових методів колаборативної фільтрації та ансамблевих гібридних рекомендаційних систем на основі метрик якості (MSE, RMSE, MAE).

Новизна – розробка методу формування ознак, який поєднує sentiment-аналіз, векторні подібності та прогнози методів колаборативної фільтрації, а також застосування цих ознак моделями регресії у задачі прогнозування рейтингу.

## ABSTRACT

Master's thesis: 82 pages, 16 figures, 35 tables, 1 appendix, 21 references.

RECOMMENDER SYSTEM, HYBRID FILTERING, TEXT VECTORIZATION, SENTIMENT ANALYSIS, COLLABORATIVE FILTERING

Object of the research – predicting user ratings based on users' text reviews and product metadata.

Subject of the research – collaborative filtering models (SVDpp, CoClustering, KNNBaseline, SlopeOne), text vectorization methods (TF-IDF, Word2Vec, SBERT), sentiment analysis methods (VADER, TextBlob), regression models (Random Forest, XGBoost, Linear Regression), and an ensemble hybrid recommender system based on stacking.

Purpose of the research – to develop an ensemble hybrid recommender system based on stacking; to compare the impact of new features on prediction generation; and to compare the prediction results of baseline collaborative filtering methods and ensemble hybrid recommender systems using quality metrics (MSE, RMSE, MAE).

Scientific novelty – the development of a feature engineering method that combines sentiment analysis, vector similarity measures, and predictions from collaborative filtering methods, as well as the use of these features by regression models for rating prediction.

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ В ГАЛУЗІ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	9
1.1 Поняття рекомендаційної системи.....	9
1.2 Види гібридних рекомендаційних систем.....	10
1.2.1 Монолітна рекомендаційна система.....	10
1.2.2 Ансамблева гібридна рекомендаційна система.....	12
1.2.3 Змішана гібридна рекомендаційна система.....	13
1.3 Огляд сучасних методів вирішення задачі побудови рекомендаційної системи для задачі прогнозування рейтингів.....	15
1.4 Огляд сервісів з рекомендаційними системами.....	16
1.5 Постановка задачі.....	19
1.6 Висновки до розділу 1.....	19
РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ТЕКСТІВ ОГЛЯДІВ КОРИСТУВАЧІВ...	21
2.1 Методи векторизації текстів та міри подібності.....	21
2.1.1 Term Frequency-Inverse Document Frequency.....	21
2.1.2 Word2Vec.....	22
2.1.3 Sentence-BERT.....	23
2.1.4 Міри подібності.....	25
2.2 Моделі колаборативної фільтрації.....	25
2.2.1 Поняття матричної факторизації.....	25
2.2.2 SVDpp.....	26
2.2.3 CoClustering.....	27
2.2.4 SlopeOne.....	28
2.2.5 KNNBaseline.....	29
2.3 Моделі прогнозування.....	29

	7
2.3.1 Поняття задачі прогнозування.....	29
2.3.2 eXtreme Gradient Boosting.....	30
2.3.3 Лінійна регресія.....	31
2.3.4 Random Forest.....	32
2.4 Методи аналізу настрою текстових корпусів.....	33
2.4.1 VADER.....	33
2.4.2 TextBlob.....	34
2.5 Метрики якості прогнозування.....	35
2.6 Алгоритм побудови ансамблевої гібридної рекомендаційної системи.....	35
2.7 Висновки до розділу 2.....	37
<b>РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....</b>	<b>39</b>
3.1 Опис середовища розробки.....	39
3.2 Опис бібліотек Python.....	40
3.3 Огляд наборів даних.....	42
3.4 Попередня обробка даних.....	47
3.5 Аналіз отриманих результатів.....	47
3.6 Висновки до розділу 3.....	56
<b>РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ.....</b>	<b>59</b>
4.1 Опис ідеї стартап-проекту.....	60
4.2 Технологічний аудит ідеї проекту.....	61
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	63
4.4 Розроблення ринкової стратегії стартап-проекту.....	68
4.5 Розроблення маркетингової програми стартап-проекту.....	69
4.6 Висновки до розділу 4.....	71
<b>ВИСНОВКИ.....</b>	<b>73</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	<b>74</b>
<b>ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ.....</b>	<b>77</b>

## ВСТУП

Рекомендаційні системи є ключовим інструментом електронної комерції, а поєднання текстових даних і колаборативної інформації дає змогу значно підвищити точність прогнозування користувацьких рейтингів. Умови високої розрідженості даних сервісів з рекомендаційними системами та різноманітність текстового контенту створюють потребу в гібридних методах, що інтегрують сучасні підходи обробки природної мови й алгоритми колаборативної фільтрації.

Розділ 1 містить дослідження предметної області в галузі розробки рекомендаційних систем. Розглянуто поняття рекомендаційної системи, види гібридних рекомендаційних систем (монолітна, ансамблева, змішана). А також проаналізовано сучасні методи побудови рекомендаційних систем. Сформульовано постановку задачі.

Розділ 2 містить математичні основи побудови рекомендаційної системи з використанням текстів оглядів користувачів. Розглянуто методи векторизації текстів (TF-IDF, Word2Vec, SBERT), моделі колаборативної фільтрації (SVDpp, CoClustering, SlopeOne, KNNBaseline), моделі прогнозування (XGBoost, Random Forest, Linear Regression), методи сентимент-аналізу (VADER, TextBlob) та метрики якості прогнозування (RMSE, MSE, MAE). Описано алгоритм побудови ансамблевої гібридної рекомендаційної системи.

Розділ 3 містить практичну реалізацію алгоритму побудови рекомендаційної системи. Розробка проводилася у середовищі Google Colab за допомогою мови програмування Python. Проведено порівняльний аналіз результатів прогнозування методів колаборативної фільтрації та гібридних систем.

Розділ 4 містить опис стартап-проекту та його основні етапи.

# РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ В ГАЛУЗІ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

## 1.1 Поняття рекомендаційної системи

Рекомендаційна система – це програмна технологія, призначена для відбору та подання користувачеві інформації, що найбільш відповідає його потребам, характеристикам або контексту взаємодії. Вона функціонує як інтелектуальний фільтр, який зменшує інформаційне перевантаження, автоматично оцінюючи релевантність великої кількості об'єктів і пропонуючи найбільш релевантні. Така система моделює переваги користувачів на основі історії взаємодій, текстових оглядів користувачів, демографічних даних або структурних властивостей текстового контенту, формуючи персоналізовані рекомендації.

Профіль користувача – це формалізоване представлення інформації про користувача, яке використовується рекомендаційною системою для моделювання його інтересів, поведінки та потреб, що може включати історію взаємодій, текстові огляди, демографічні характеристики, а також приховані ознаки, отримані за допомогою алгоритмів навчання (прогнози оцінок щодо товарів без оцінок від даного користувача).

Характеристика товару – це набір описових ознак, які формально представляють властивості кожного об'єкта в рекомендаційній системі. До таких характеристик можуть належати текстові описи, категорії, жанри, технічні параметри, ключові слова, візуальні або аудіо ознаки, а також приховані ознаки, отримані з допомогою алгоритмів машинного навчання (ембединги з тексту чи зображень).

У класичному розумінні, рекомендаційна система ґрунтується на таких трьох групах методів, а саме:

- 1) контентної фільтрації - порівнюють характеристики об'єктів з профілем користувача, який формується на основі проаналізованих вподобаних ним елементів і на основі цього надаються топ  $N$  товарів;
- 2) колаборативної фільтрації - визначають уподобання користувача на основі аналізу поведінки подібних користувачів або подібність між об'єктами у спільних історіях взаємодій, і на основі цього надаються прогнози оцінок користувачів заданим товарам;
- 3) гібридної фільтрації - поєднують різні джерела сигналів для підвищення точності передбачень для пошуку топ  $N$  товарів чи прогнозу оцінок користувачів заданим товарам [1].

## **1.2 Види гібридних рекомендаційних систем**

Як було згадано у пункті 1.1, у рекомендаційних системах виділяють три основні підходи: контентну фільтрацію, колаборативну фільтрацію та гібридну фільтрацію.

Гібридні рекомендаційні системи поєднують кілька методів рекомендації, а саме контентну та колаборативну фільтрацію для усунення слабких сторін кожного з підходів і підвищення точності, стійкості та узагальнювальної здатності моделей.

Гібридні рекомендаційні системи поділяються на три види, а саме:

- 1) монолітні гібридні рекомендаційні системи;
- 2) ансамблеві гібридні рекомендаційні системи;
- 3) змішані гібридні рекомендаційні системи [1].

### **1.2.1 Монолітна рекомендаційна система**

Монолітна рекомендаційна система – це підхід, у якому всі компоненти рекомендаційного процесу об'єднані в одну велику, нероздільну систему. Така система зазвичай містить єдиний модуль, що одночасно

відповідає за збирання даних, їх передобробку, побудову профілів користувачів та об'єктів, навчання моделі, обчислення рекомендацій, оновлення параметрів і подання результатів користувачу.

Переваги монолітної рекомендаційної системи полягають у наступному, а саме:

- 1) простота архітектури – усі компоненти працюють у межах одного застосунку, що полегшує розуміння та підтримку;
- 2) швидкий старт розробки;
- 3) обчислення виконуються локально, без мережових викликів між частинами системи;
- 4) підходять для невеликих проєктів з невеликими обсягами даних.

Недоліки монолітної рекомендаційної системи полягають у наступному, а саме:

- 1) вся система масштабується як одне ціле;
- 2) складність оновлення моделей та обмежена гнучкість;
- 3) підвищений ризик збою, оскільки помилка в одному компоненті може зупинити роботу всієї системи.

Архітектуру монолітної гібридної рекомендаційної системи відображено на рис. 1.1.

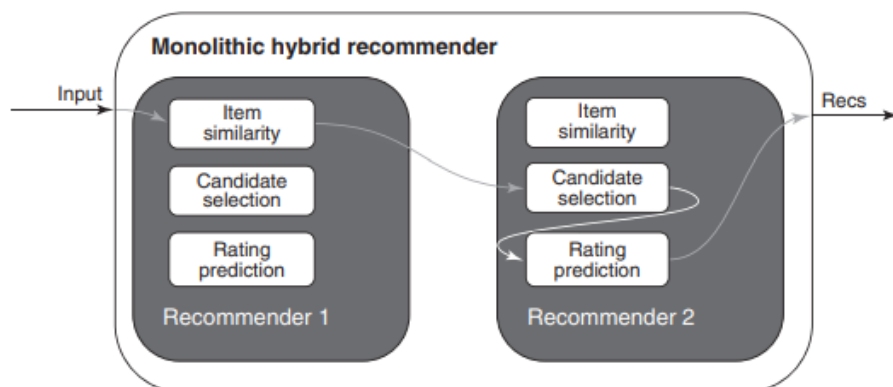


Рисунок 1.1 – Архітектура монолітної гібридної рекомендаційної системи [1]

### 1.2.2 Ансамблева гібридна рекомендаційна система

Ансамблева гібридна рекомендаційна система – це підхід, у якому кілька незалежних моделей використовуються одночасно або послідовно для формування більш точних і стабільних рекомендацій. Ансамбль може поєднувати моделі різної природи (контентні, колаборативні, нейронні, факторизаційні чи контекстні) та об'єднувати їхні результати шляхом усереднення, зважування, голосування або побудови метамоделі, яка навчається на виходах окремих алгоритмів. Основна ідея ансамблю полягає у зменшенні похибок окремих методів шляхом комбінування їхніх сильних сторін та зниженні чутливості до шуму, розрідженості даних і змін поведінки користувачів.

Переваги ансамблевої гібридної рекомендаційної системи полягають у наступному, а саме:

- 1) підвищення точності рекомендацій завдяки комбінуванню різних алгоритмів;
- 2) стійкість до шуму та нерівномірності даних;
- 3) можливість компенсувати слабкі сторони одних моделей сильними сторонами інших;
- 4) гнучкість у налаштуванні та можливість масштабувати ансамбль за рахунок додавання нових моделей без переналаштування всієї системи.

Недоліки ансамблевої гібридної рекомендаційної системи полягають у наступному, а саме:

- 1) значно більша обчислювальна складність порівняно з одиничними моделями;
- 2) потреба у складній інфраструктурі для навчання, об'єднання та керування кількома моделями;
- 3) ускладнення налагодження та інтерпретації результатів, оскільки фінальне рішення формується з багатьох джерел;

- 4) підвищені вимоги до зберігання даних і модульності коду, що може ускладнювати експлуатацію та розгортання в реальних системах.

Архітектуру ансамблевої гібридної рекомендаційної системи відображено на рис. 1.2.

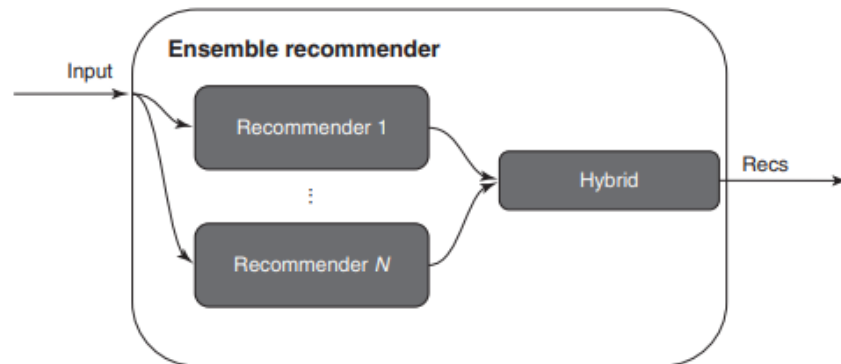


Рисунок 1.2 – Архітектура ансамблевої гібридної рекомендаційної системи [1]

### 1.2.3 Змішана гібридна рекомендаційна система

Змішані рекомендаційні системи – це підхід, у якому результати кількох різних алгоритмів одночасно відображаються користувачеві у вигляді окремих блоків або списків рекомендацій. На відміну від ансамблів, де моделі поєднуються у спільний прогноз, змішані системи подають паралельні набори рекомендацій, сформовані різними методами. Це забезпечує різноманітність і дозволяє користувачу обирати між кількома типами результатів, що підвищує корисність системи та покращує загальне сприйняття рекомендацій.

Переваги змішаних рекомендаційних систем полягають у наступному, а саме:

- 1) забезпечують різноманітність рекомендацій за рахунок одночасного використання різних алгоритмів;
- 2) дають змогу компенсувати обмеження одного методу за допомогою іншого;
- 3) дозволяють масштабувати систему шляхом додавання нових незалежних модулів рекомендацій.

Недоліки змішаних рекомендаційних систем полягають у наступному, а саме:

- 1) потребують більше обчислювальних ресурсів, оскільки кожен алгоритм працює окремо;
- 2) можуть перевантажувати інтерфейс, якщо списків рекомендацій занадто багато;
- 3) не гарантують оптимального загального ранжування, оскільки результати не об'єднуються у спільний список;
- 4) складніше забезпечити узгодженість між блоками, особливо у системах із динамічним оновленням даних.

Архітектуру змішаної гібридної рекомендаційної системи відображено на рис. 1.3.

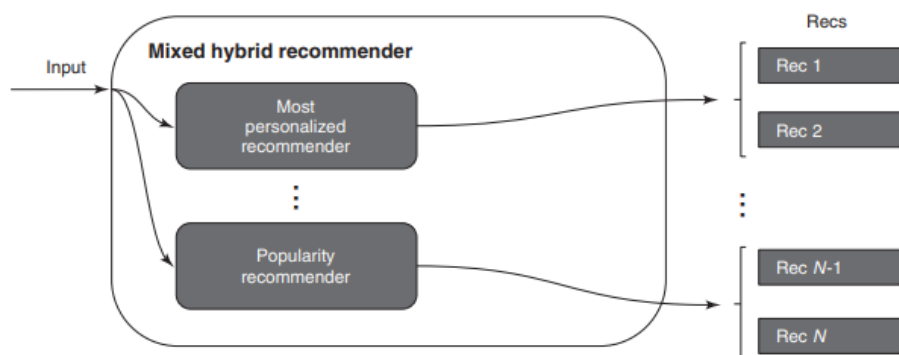


Рисунок 1.3 – Архітектура змішаної гібридної рекомендаційної системи [1]

### **1.3 Огляд сучасних методів вирішення задачі побудови рекомендаційної системи для задачі прогнозування рейтингів**

Сучасні підходи до побудови рекомендаційних систем активно інтегрують текстові дані, зокрема огляди користувачів. Згідно з оглядом сучасних тенденцій у рекомендаційних системах [2], особливе значення мають методи, що використовують контекстуальні ембединги, отримані за допомогою трансформерів. Тексти оглядів дозволяють моделі вловлювати семантичні залежності, тональність та інформативні сигнали, що значно доповнюють класичні рейтингові матриці. Такі підходи дозволяють розв'язувати проблему розрідженості (sparsity) та cold-start, покращуючи точність прогнозування рейтингів завдяки об'єднанню поведінкових і лінгвістичних ознак.

У роботах, орієнтованих на практичні рекомендаційні задачі на основі даних Amazon, зокрема в дослідженні [3], показано ефективність моделей, що поєднують методи матричної факторизації (Matrix Factorization, SVD) з додатковими текстовими ознаками. Огляди користувачів виступають джерелом семантичної інформації, яка перетворюється на вектори за допомогою TF-IDF або простих ембедингів. Такі гібридні моделі зберігають високу швидкість виконання і добре масштабуються, але через обмеженість текстового представлення не повністю враховують складні семантичні структури тексту. Таким чином, традиційні підходи зміцнюють базову точність і залишаються конкурентними завдяки низькій обчислювальній складності.

Новіші методи, орієнтовані на багаті за змістом джерела даних, демонструють можливості глибокого представлення тексту для значного покращення точності rating prediction. У роботі, присвяченій рекомендаційним моделям для категорії Amazon Books [4], автори підкреслюють, що навіть прості текстові представлення (TF-IDF, word2vec) здатні підвищувати якість рекомендацій, але найбільший приріст досягається

при використанні контекстних моделей, таких як BERT або Sentence-BERT. Такі підходи дозволяють виявляти приховані теми, оцінки та емоційні сигнали, формуючи точніші профілі користувачів та товарів.

Загальний напрям сучасних досліджень полягає у переході від класичних моделей колаборативної фільтрації та лінійних методів до гібридних моделей, які інтегрують глибинні мовні представлення. Використання трансформерів, мультимодальних ознак та моделей спільного навчання (joint learning) стає ключовим чинником підвищення точності у задачі прогнозування рейтингу.

#### **1.4 Огляд сервісів з рекомендаційними системами**

Сучасні електронні платформи активно застосовують рекомендаційні системи, інтегруючи поведінкові дані, рейтинги та текстові огляди користувачів. Аналіз текстових оглядів дозволяє враховувати додаткові семантичні сигнали, такі як емоційне забарвлення чи згадки характеристик товарів, що робить їх важливим джерелом даних у сучасних методах моделювання рейтингів та персоналізації.

Amazon є одним із найвідоміших прикладів використання рекомендаційних систем. Платформа впроваджує різні алгоритми персоналізації, зокрема item-to-item collaborative filtering. Окрім цього, тексти оглядів Amazon активно використовуються для задач прогнозування рейтингів та формування пояснених рекомендацій. Аналіз текстів на основі моделей обробки природної мови дає змогу виявляти латентні теми та емоційні сигнали, які покращують точність моделей.

Приклад блоку з текстовими оглядами Reviews на сайті Amazon наведено на рис. 1.4.

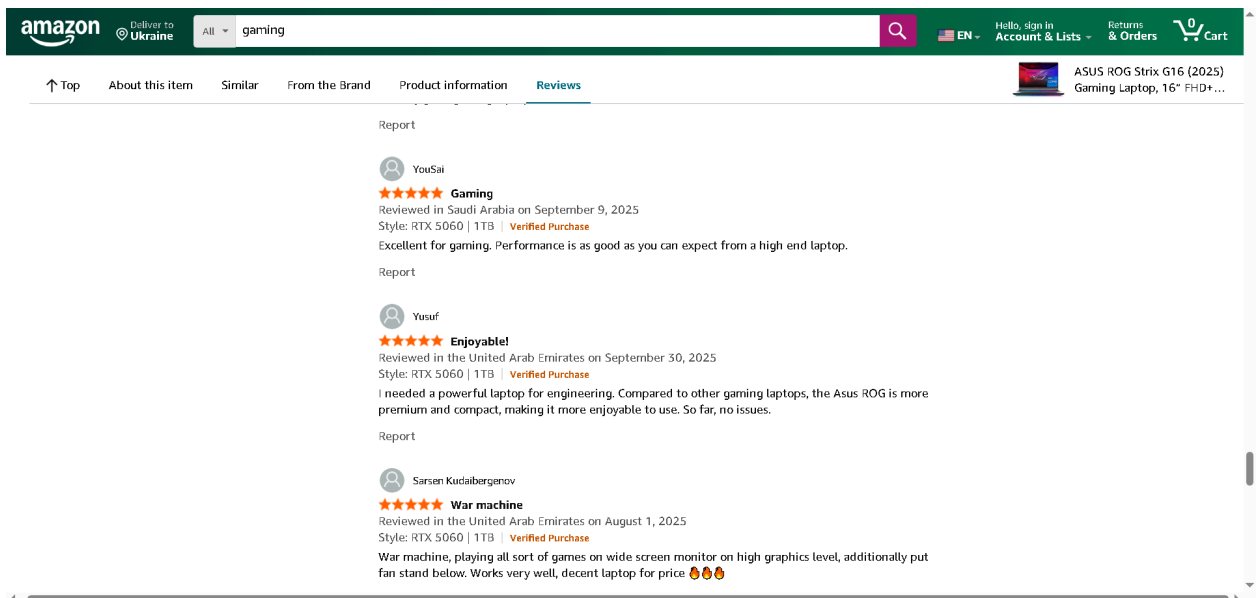


Рисунок 1.4 – Приклад блоку Reviews на сайті Amazon

Rozetka та Prom є великими українськими платформами електронної комерції, що застосовують рекомендаційні механізми, зокрема на основі поведінки користувачів та популярності товарів. Однак їхні внутрішні алгоритми, включно з можливістю використання текстових оглядів у рекомендаційних системах, не оприлюднюються у відкритих джерелах.

Приклад блоку з текстовими оглядами Відгуки та питання на сайті Rozetka наведено на рис. 1.5.

ROZETKA

Каталог

Я шукаю...

Знайти UA

Про товар Характеристики **Відгуки та питання 20/26** Відео 1 Купують разом

### Відгуки та питання

Оцінка користувачів **4.67/5** ★  
на основі 20 відгуків

5 ★ 13  
4 ★ 1  
3 ★ 0  
2 ★ 0  
1 ★ 1

[Написати відгук](#)

**Відгуки (20)** **Питання (26)**

**Олександр Булій** ✓  
Відгук від покупки. Продавець: Rozetka. Колір: Синій. Об'єм встановленої оперативної пам'яті: 16 ГБ. Обсяг SSD: 512 ГБ  
★★★★★  
Купляв синові. Довго не запускався при включенні. При установці ос щось клацало в середині. Горів Кап Лок. На другий день включався на 5 раз. Потім перестав включатися зовсім. При поверненні забрали на діагностику. Синові сподобалися характеристики. Якби замінили на новий ...  
Читати далі ↓  
[Відповісти](#) 👍 0 🗨️ 0 ⋮

**Олександр Фесьоха** ✓  
Відгук від покупки. Продавець: Rozetka. Колір: Синій. Об'єм встановленої оперативної пам'яті: 16 ГБ. Обсяг SSD: 512 ГБ  
★★★★★  
Відмінний по всьому  
**Переваги:** Є...  
Читати далі ↓  
[Відповісти](#) 👍 0 🗨️ 3 ⋮

**Петро Макар** ✓  
Відгук від покупки. Продавець: Rozetka. Колір: Синій. Об'єм встановленої оперативної пам'яті: 16 ГБ. Обсяг SSD: 512 ГБ  
★★★★★  
Читати далі ↓  
[Відповісти](#) 👍 0 🗨️ 0 ⋮

Ноутбук HP Victus Gaming Laptop 15-fa2710ua (BF1H5EA) Performance Blue / 15.6" IPS Full HD 144 Гц / Intel Core i5-13420H / RAM 16 ГБ / SSD 512 ГБ / nVidia GeForce RTX 3050, 6 GB

33 999 ₴ [Купити](#)

Рисунок 1.5 – Приклад блоку з текстовими оглядами Відгуки та питання на сайті Rozetka

Приклад блоку з текстовими оглядами Відгуки про товар на сайті Prom наведено на рис. 1.6.

prom

Каталог

Я шукаю...

Знайти Кабінет Сповіщення Обране Кошик

### Відгуки про товар 1

★★★★★ 5.0

**Аліна Є.** 13.07.2025  
Придбано на Prom.ua  
Все Чудово  
Класний ноутбук. Швидко відправка.  
Рекомендую продавця 👍

[Переглянути всі](#)

### Питання та відповіді

Хочеш дізнатися більше про товар? Запитуй — продавець залюбки підкаже.

[Поставити запитання](#)

[Дивитись все](#)

[Чат](#)

Рисунок 1.6 – Приклад блоку з текстовими оглядами Відгуки про товар на сайті Prom

## 1.5 Постановка задачі

У рамках даної роботи поставлено такі задачі, а саме:

- 1) інтегрувати показники косинусної подібності між текстовими оглядами користувачів та відповідними текстовими описами товарів, сентиментні ознаки (міри позитивності, нейтральності, негативності, compound, суб'єктивність) текстових оглядів користувачів, прогнози базових моделей колаборативної фільтрації у рамках стекінгу у задачі прогнозування рейтингу;
- 2) реалізувати програмний продукт для дослідження у середовищі Google Colab та мови програмування Python;
- 3) оцінити, чи покращує гібридний підхід на основі стекінгу результатів декількох моделей та методів точність прогнозування рейтингу порівняно з окремими моделями колаборативної фільтрації на прикладу даних Amazon Reviews Office Products, Amazon Meta Office Products, Amazon Reviews Automotive, Amazon Meta Automotive [5];
- 4) встановити, які групи ознак (подібність текстів, сентиментні ознаки, прогнози методів колаборативної фільтрації) забезпечують найбільший внесок у підвищення точності;
- 5) сформулювати висновки по виконаній роботі.

## 1.6 Висновки до розділу 1

Рекомендаційні системи у сучасних сервісах значно полегшують пошук релевантних товарів. Проте, досить часто, користувач не розуміє, як ці рекомендації формуються, тобто, самі алгоритми пошуку не надаються у відкритому доступі. Попри це, сучасні сервіси активно залучають сучасні досягнення у сфері штучного інтелекту та машинного навчання задля

формування більш точніших рекомендацій, що у свою чергу покращує користувацький досвід.

В даному розділі розглянуто поняття рекомендаційної системи, типи рекомендаційних систем, види гібридних рекомендаційних систем та, для подальшого дослідження обрано ансамблеві гібридні рекомендаційні системи. Також, до уваги взято можливість використовувати текстові огляди у рекомендаційних системах.

## РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ТЕКСТІВ ОГЛЯДІВ КОРИСТУВАЧІВ

### 2.1 Методи векторизації текстів та міри подібності

Нижче буде проведено опис методів векторизації текстів, а також розглянуто міри подібності.

#### 2.1.1 Term Frequency-Inverse Document Frequency

TF-IDF (Term Frequency-Inverse Document Frequency) є одним із найпоширеніших методів векторизації текстів у задачах обробки природної мови. Він перетворює корпус документів у матрицю числових ваг, що відображають важливість слів у межах окремого документа та всієї колекції. TF-IDF зменшує вагу дуже частих, малозмістовних слів і збільшує вагу термінів, які є інформативними для конкретного документа [6].

Метод складається з таких двох основних частин:

1. TF (Term Frequency) – частотність терміна  $t$  у документі  $d$ . Формула:

$$TF(t, d) = f_{t,d} / \sum_k f_{k,d}, \text{ де } f_{k,d} - \text{кількість появи терміна } t \text{ у документі } d.$$

2. IDF (Inverse Document Frequency) – обернена частотність документа. Формула:  $IDF(t) = \log((1 + n)/(1 + df_t)) + 1$ , де  $n$  – загальна кількість документів у корпусі,  $df_t$  – кількість документів з терміном  $t$ .

Підсумкова вага терміна:  $TFIDF(t, d) = TF(t, d) \times IDF(t)$ .

Переваги методу TF-IDF полягають у наступному, а саме: простота та інтерпретованість, ефективний в операціях обчислення, має високу якість порівняння текстів, реалізований в Python.

Недоліки методу TF-IDF полягають у наступному, а саме: ігнорує семантику, надає розріджені матриці векторизації, чутливий до рідкісних слів.

### 2.1.2 Word2Vec

Word2Vec є одним із найпоширеніших сучасних методів семантичної векторизації текстів, застосовуваних у задачах обробки природної мови. На відміну від класичних статистичних підходів, Word2Vec будує щільні вектори слів, які відображають їхній семантичний зміст та контекст використання у великому корпусі текстів. Основна ідея моделі полягає у тому, що значення слова визначається словами, які його оточують у реальних мовних даних [7].

Метод Word2Vec реалізується у вигляді двох архітектур, а саме: CBOW (Continuous Bag-of-Words), Skip-gram.

У моделі CBOW контекстні слова, що розташовані в певному вікні навколо цільового слова, використовуються для передбачення саме цього цільового слова. Контекст у CBOW розглядається як «мішок слів» (bag-of-words), тобто порядок слів у вікні не враховується. Ідея полягає в тому, що зібрана інформація про оточення дозволяє моделі навчитися типових ситуацій, у яких виникає конкретне слово. Такий підхід забезпечує швидке навчання та добре працює на великих корпусах, де частотні закономірності контекстів легко виявляються. Автори демонструють, що CBOW є обчислювально дешевшим варіантом, особливо на великих словниках.

Skip-gram використовує цільове слово для передбачення слів, що зустрічаються поруч із ним у тексті. Модель намагається вивчити, які контексти є типовими для кожного слова. Skip-gram краще справляється з рідкісними словами, оскільки кожне таке слово під час навчання стає джерелом більшої кількості тренувальних прикладів. Завдяки цьому Skip-gram створює більш якісні векторні представлення для слів, які недостатньо часто з'являються в корпусі.

Схема архітектур CBOW та Skip-gram відображено на рис. 2.1.

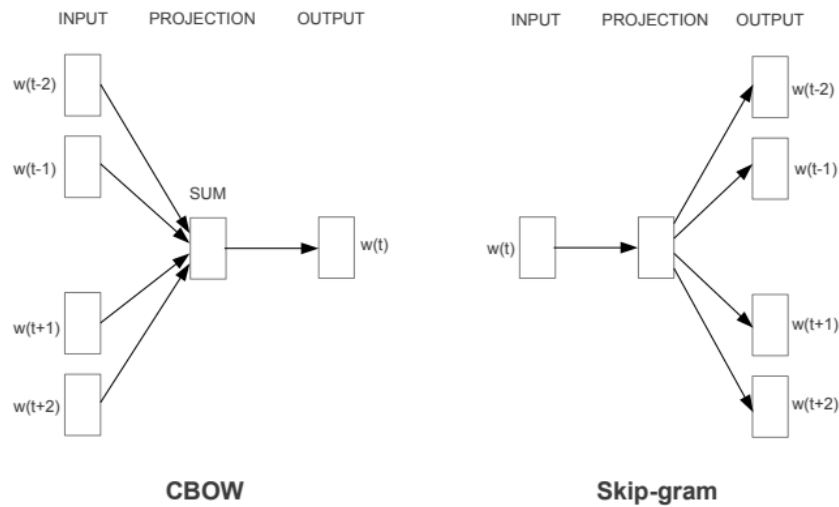


Рисунок 2.1 – Архітектура CBOW (зліва) та Skip-gram (справа) [7]

Переваги методу Word2Vec полягають у наступному, а саме: враховує контекст та семантичні зв'язки між словами, дає інформативні вектори, ефективний для великих текстових корпусів.

Недоліки методу Word2Vec полягають у наступному, а саме: наявність рідкісних слів у корпусі може знизити якість векторного представлення методом, працює лише на рівні слів.

### 2.1.3 Sentence-BERT

Sentence-BERT (SBERT) є модифікацією моделі BERT, розробленою для ефективного отримання векторних представлень речень. На відміну від звичайного BERT, який не призначений для обчислення семантичної подібності між реченнями без значних обчислювальних витрат, SBERT дозволяє генерувати компактні sentence embeddings, що можна безпосередньо порівнювати за допомогою косинусної подібності [8, 21].

Метод будується на основі попередньо натренованих трансформерів BERT і RoBERTa, але доповнюється спеціальними навчальними цілями. Модель навчається на задачах Semantic Textual Similarity та Natural Language Inference, що дозволяє їй краще розрізняти значення речень. На відміну від стандартного BERT, який вимагає повторного пропускання пари речень через модель для кожної операції порівняння, SBERT генерує вектори для речень незалежно. Це забезпечує значне прискорення роботи у великих корпусах текстів і робить модель придатною для задач пошуку, кластеризації та порівняння текстів.

В даній роботі буде використано саме трансформер all-MiniLM-L6-v2 з фреймворку sentence\_transformer. Даний трансформер має такі особливості, а саме: 6 шарів енкодера; розмірність прихованого шару (384,); використовує дистиляцію уваги, що корисно для швидкості енкодування тесту.

Архітектура SBERT відображена на рис. 2.2.

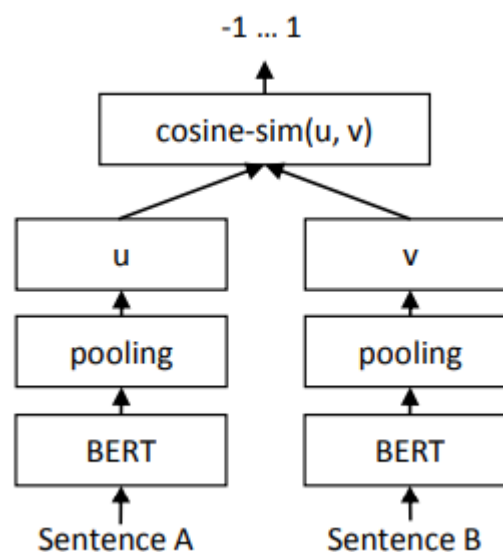


Рисунок 2.2 – Архітектура SBERT [8]

Переваги методу Sentence-BERT полягають у наступному, а саме: швидкі обчислення, якісні семантичні подібності, гнучкість використання.

Недоліки методу Sentence-BERT полягають у наступному, а саме: потребує додаткове донавчання, має обмежений розмір контексту, працює на рівні речення або коротких текстів.

#### 2.1.4 Міри подібності

**Косинусна подібність** оцінює кут між двома векторами [9]. Вона показує, наскільки напрямки векторів подібні, незалежно від їхньої довжини. Це робить косинусну подібність особливо ефективною для текстів, оскільки довжина документа не впливає на значення подібності. Формула косинусної подібності:

$$\text{cosine}(x, y) = (x, y) / (||x|| ||y||),$$

де  $x, y$  – вектори. Діапазон значень косинусної подібності –  $[0, 1]$ .

**Добуток векторів** відображає ступінь подібності векторів з урахуванням ваг окремих координат. Формула добутку векторів:

$$\text{dot}(x, y) = (x, y).$$

Діапазон значень –  $(-\infty, +\infty)$ .

В даній роботі використовуватиметься саме косинусна подібність задля пошуку міри подібності векторних представлень текстових оглядів та відповідних векторних представлень описів товарів.

## 2.2 Моделі колаборативної фільтрації

### 2.2.1 Поняття матричної факторизації

Матрична факторизація у рекомендаційних системах – це апроксимація початкової матриці взаємодій користувач-товар за допомогою добутку двох матриць нижчої розмірності [10]. Кожному товару  $i$  відповідає вектор  $q_i \in \mathbb{R}^f$ , Кожному користувачу  $u$  відповідає вектор  $p_u \in \mathbb{R}^f$ , де  $f$  –

розмірність латентного простору. Приблизна оцінка користувача  $u$  товару  $i$  –

$$\hat{r}_{ui} = q_i^T p_u.$$

Оптимальні латентні вектори знаходяться шляхом мінімізації регуляризованої квадратичної похибки на відомих рейтингах. Формула:

$$\min_{P,Q} \sum_{(u,i) \in K} (\hat{r}_{ui} - r_{ui})^2 + \lambda(\|p_u\|^2 + \|q_i\|^2),$$

де  $P, Q$  – матриці латентних факторів користувачів та товарів відповідно,  $K$  – множина всіх відомих взаємодій пар користувач-товар,  $\lambda$  – коефіцієнт регуляризації.

### 2.2.2 SVDpp

SVDpp – це розширення алгоритму SVD, що враховує не лише явні рейтинги користувача, але й неявний зворотний зв'язок. Модель додає до латентного вектора користувача додатковий компонент, що агрегує інформацію про всі елементи, з якими користувач взаємодіяв [11]. Це покращує якість рекомендацій. Формула прогнозів:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j),$$

де  $\mu$  – глобальне середнє значення рейтингу,  $b_u$  – зсуви користувача,  $b_i$  – зсуви елемента,  $N(u)$  – множина товарів, з якими користувач  $u$  взаємодіяв,  $y_j$  – латентні вектори, що моделюють неявний зворотний зв'язок по товарах  $j$ .

Оптимізація здійснюється стохастичним градієнтним спуском, згідно з правилами оновлень. Формули оновлення параметрів:

$$b_u = b_u + \gamma(e_{ui} - \lambda b_u),$$

$$b_i = b_i + \gamma(e_{ui} - \lambda b_i),$$

$$p_u = p_u + \gamma(e_{ui} q_i - \lambda p_u),$$

$$q_i = q_i + \gamma(e_{ui}p_u - \lambda q_i),$$

$$y_j = y_j + \gamma(e_{ui}|N(u)|^{-1/2}q_i - \lambda y_j),$$

де  $e_{ui} = \hat{r}_{ui} - r_{ui}$ ,  $\gamma$  – швидкість навчання.

Переваги моделі SVDpp полягають у наступному, а саме: враховує неявний фідбек, підвищує точність рекомендацій, покращує моделювання користувачів із малою кількістю рейтингів, добре працює в розріджених матрицях, демонструє високу якість у практичних системах.

Недоліки моделі SVDpp полягають у наступному, а саме: вища обчислювальна складність, потребує більше пам'яті через додаткові вектори, чутливий до підбору гіперпараметрів, повільніший за класичний SVD у навчанні.

### 2.2.3 CoClustering

CoClustering (двовимірне кластеризаційне групування) – це модель одночасного кластерування рядків і стовпців матриці користувач–товар. На відміну від звичайної колаборативної фільтрації, CoClustering намагається знайти латентну структуру не лише серед користувачів або лише серед товарів, а в їхніх перетинах. Оптимізація за допомогою інформаційно-теоретичної функції, що мінімізує втрати взаємної інформації після стиснення матриці у блоки [12].

Користувачам та товарам призначаються відповідні кластери  $C_u$  та  $C_i$  відповідно, та спільні кластери  $C_{ui}$ . Прогноз визначається за формулою:

$$\hat{r}_{ui} = C_{ui} + (\mu_u - C_u) + (\mu_i - C_i).$$

Кластери визначаються за допомогою простого методу оптимізації (наприклад, k найближчих сусідів).

Переваги моделі CoClustering полягають у наступному, а саме: зменшує розмірність одночасно по користувачах і товарах, виявляє

двовимірні латентні структури, добре працює при розріджених даних, інтерпретовані спільні кластери, стійкість до шуму завдяки усередненню в кластерах.

Недоліки моделі CoClustering полягають у наступному, а саме: якість залежить від кількості кластерів, не моделює складні нелінійні взаємодії, може втрачати дрібні індивідуальні сигнали при усередненні, потребує етапу навчання кластеризації, не масштабується до надвеликих даних.

### 2.2.4 SlopeOne

SlopeOne є простою, і водночас точною моделлю. Заснована на різницях між рейтингами. Мета алгоритму – передбачити рейтинг користувача на товар  $j$ , використовуючи той факт, що різниця між рейтингами двох товарів є статистично стабільною у популяції [13].

Формула прогнозу рейтингу:

$$\hat{r}_{ui} = \mu_u + |R_i(u)|^{-1} \sum_{j \in R_i(u)} dev(i, j),$$

де  $R_i(u)$  – набір релевантних елементів,  $dev(i, j) = U_{ij}^{-1} \sum_{u \in U_{ij}} (r_{ui} - r_{uj})$ ,  $\mu_u$  – середнє значення всіх оцінок, наданих користувачем  $u$ ,  $U_{ij}$  – множина всіх користувачів, які оцінили обидва елементи  $i, j$ .

Переваги моделі SlopeOne полягають у наступному, а саме: дуже простий алгоритм, мало параметрів, легка у реалізації та обчисленні, стійка до розрідженості, швидко оновлюється при появі нових рейтингів.

Недоліки моделі SlopeOne полягають у наступному, а саме: працює менш точно на даних із сильною нелінійною структурою, не враховує латентні фактори, менш гнучкий.

## 2.2.5 KNNBaseline

KNNBaseline (або baseline-adjusted k-nearest neighbors) є модифікацією item-based чи user-based KNN, у якій прогнозування рейтингу виконується не напряму, а після корекції на базову модель відхилень (baseline predictors) [14]. Це дозволяє суттєво зменшити зміщення, спричинене індивідуальними схильностями користувачів та популярністю товарів.

Формула прогнозу рейтингу (user-based):

$$\hat{r}_{ui} = b_{ui} + \left( \sum_{v \in N_i^k(u)} \text{sim}(u, v)(r_{vi} - b_{vi}) \right) / \left( \sum_{v \in N_i^k(u)} \text{sim}(u, v) \right).$$

Формула прогнозу рейтингу (item-based):

$$\hat{r}_{ui} = b_{ui} + \left( \sum_{j \in N_u^k(i)} \text{sim}(i, j)(r_{uj} - b_{uj}) \right) / \left( \sum_{j \in N_u^k(i)} \text{sim}(i, j) \right).$$

Переваги моделі KNNBaseline полягають у наступному, а саме: усуває систематичні зміщення, проста й інтерпретована модель, може використовувати будь-яку метрику схожості.

Недоліки моделі KNNBaseline полягають у наступному, а саме: чутливість до вибору метрики, гірша масштабованість при великих  $k$ .

## 2.3 Моделі прогнозування

### 2.3.1 Поняття задачі прогнозування

Нехай  $Y \in \mathbb{R}^n$  – вектор цільової змінної,  $x_i = [x_{i1}, \dots, x_{im}]$  – вектор ознак для спостереження  $i$  ( $i = 1, \dots, n$ ),  $X = [x_1, \dots, x_n]^T$  – матриця ознак,  $m$  – кількість ознак. Тоді задача прогнозування полягає у пошуку такої функції  $f$ , що  $f(X) \approx Y$ .

### 2.3.2 eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) – високопродуктивна реалізація градієнтного бустингу дерев рішень, орієнтованою на масштабованість, регуляризацію та ефективність обчислень. Модель будується як ансамбль дерев, де кожне нове дерево наближає негативний градієнт функції втрат поточного ансамблю [15].

Формула прогнозу  $\hat{y}_i$ :

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i),$$

де  $f_k$  –  $k$ -те дерево,  $x_i$  – ознаки товару  $i$ ,  $K$  – кількість побудованих дерев.

Формула узагальненої цільової функції втрат:

$$L^t = \sum_{i=1}^n (l(y_i, \hat{y}_i^{(t-1)}) + f_t(x_i)) + \Omega(f_t), \quad \Omega(f) = \gamma T + 1/2\lambda \|w\|^2,$$

де  $n$  – кількість прикладів,  $l$  – функція втрат,  $\Omega$  – функція регуляризації,  $\gamma$  – штраф за кожний листок,  $T$  – кількість листків у дереві.,  $\lambda$  – коефіцієнт L2-регуляризації ваг листків,  $w$  – вектор ваг у листках.

Побудова кожного нового дерева базується на другому порядку розкладу функції втрат:

$$L^t = \sum_i (g_i f_t(x_i) + 1/2 h_i f_t^2(x_i) + \Omega(f_t)),$$

$$\text{де } g_i = \frac{\partial l}{\partial y_i^{(t-1)}}, \quad h_i = \frac{\partial^2 l}{\partial y_i^{(t-1)2}}.$$

Градієнтний бустинг –  $L^t = L^{t-1} + \Omega(f_t)$ ,  $t$  – номер ітерації.

Переваги моделі XGBoost полягають у наступному, а саме: висока продуктивність, другий порядок оптимізації, потужна регуляризація, робота з пропущеними значеннями, масштабованість, гнучкість функцій втрат.

Недоліки моделі XGBoost полягають у наступному, а саме: чутливість до гіперпараметрів, повільність великих моделей, низька інтерпретованість, не завжди найкраща продуктивність, слабка підтримка категоріальних ознак.

### 2.3.3 Лінійна регресія

Лінійна регресія (Linear Regression, LR) є фундаментальною моделлю статистичного та машинного навчання, призначеною для встановлення лінійного зв'язку між однією або кількома незалежними змінними та цільовою величиною. Модель передбачає, що залежність має форму лінійної комбінації параметрів, і завдання полягає у знаходженні таких коефіцієнтів, які мінімізують помилку прогнозу [16].

Формула прогнозу:  $\hat{y} = Xw$ , де  $\hat{y}$  – вектор прогнозів розмірності  $n$  (кількість спостережень),  $X$  – матриця ознак спостережень  $(n, p + 1)$  (+1 означає додаткову колонку з одиниць для зсуву),  $p$  – кількість ознак,  $w$  – вектор параметрів моделі розмірності  $p + 1$  (зміщення  $w_0$  і  $p$  коефіцієнтів при ознаках).

Функція втрат:

$$L(w) = \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

Переваги моделі лінійної регресії полягають у наступному, а саме: простота та інтерпретованість, аналітична розв'язність, низька обчислювальна складність, хороша робота на малих наборах ознак, розвинена статистична теорія.

Недоліки моделі лінійної регресії полягають у наступному, а саме: лінійність припущення, чутливість до мультиколінеарності, висока чутливість до викидів, потреба в кодуванні категоріальних ознак, неможливість моделювати складні взаємодії без ручних перетворень.

### 2.3.4 Random Forest

Random Forest (випадковий ліс, RF) – ансамблевий алгоритм, що поєднує багато дерев рішень, створених на різних підвбірках даних і ознак. Модель понижує дисперсію базових дерев за рахунок бутстрепінгу та випадковості у виборі ознак під час розбиття [17].

Нехай маємо вибірку  $D = \{(x_i, y_i)\}_{i=1}^n$ ,  $x_i \in \mathbb{R}^p$ . Для кожного дерева  $T_b$ ,  $b = 1, \dots, B$  генерується бутстреп-репліка обсягом  $n$ . Ід час побудови дерева у кожному вузлі випадково вибирається підмножина ознак розмірності  $m$  значно більшої за  $p$ . І найкраще розбиття обирається лише серед цих ознак, що знижує кореляцію між деревами та збільшує стабільність ансамблю. Дерева ростуть до повної глибини без підрізання, оскільки регуляризація Random Forest досягається саме випадковістю даних і ознак.

Формула прогнозу для задачі регресії:  $\hat{y}(x) = 1/B \sum_{b=1}^B T_b(x)$ .

Переваги моделі Random Forest полягають у наступному, а саме: висока точність без складного тюнінгу, стійкість до переобучення, робота з великою кількістю ознак, нечутливість до масштабування ознак, автоматична оцінка важливості ознак, стійкість до шуму й викидів.

Недоліки моделі Random Forest полягають у наступному, а саме: низька інтерпретованість, великі розміри моделі, чутливість до дисбалансу класів, відсутність екстраполяції за межі навчального діапазону.

## 2.4 Методи аналізу настрою текстових корпусів

У даному пункті будуть розглянуті методи аналізу настрою текстових корпусів, що реалізовані на мові Python.

### 2.4.1 VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) – модель для аналізу тональності, оптимізованою під короткі тексти, типові для соціальних мереж. Метод поєднує лексикон валентностей (побудований за допомогою краудсорсингу) та контекстно-чутливі евристичні правила, що моделюють локальні контекстні явища (коригують емоційну силу слів у природному тексті) [18].

Для роботи моделі потрібні необроблений, розбитий на токени текст, доступ до лексикону валентностей, застосування контекстних правил, нормалізація фінальних score. Лексикон містить слова, сленг, абрєвіатури, емоджі, емотикони, а також графічні та пунктуаційні маркери. Контекстні модифікації включають підсилювачі, зменшувачі, заперечення, пунктуаційні підсилювачі та графічні ефекти. Модифікована валентність кожного токена підсумовується, утворюючи підсумковий sentiment score (pos (діапазон [0,1]), neu (діапазон [0,1]), neg (діапазон [0,1]), compound (діапазон [-1,1])). В сумі  $pos+neu+neg=1$ .

Переваги методу VADER полягають у наступному, а саме: забезпечує обчислення полярності тексту, роботу зі сленгом, емоджі та емоційно маркованими графічними елементами, швидко та інтерпретовану обробку тексту, стійкість до шуму, а також високу кореляцію з людськими оцінками. Модель використовує детерміновані алгоритми контекстного коригування, процедури сумування валентностей та психологічно валідований лексикон.

Недоліки методу VADER полягають у наступному, а саме: слабка здатність моделювати складні семантичні конструкції, труднощі з обробкою сарказму, відсутність механізмів глибокого контексту, потребу в ручному оновленні лексикону та нижчу ефективність на великих або формальних текстах.

## 2.4.2 TextBlob

TextBlob є лінгвістичним інструментом для обробки природної мови на Python, побудованим поверх бібліотеки Pattern, і використовує словниково-орієнтований підхід до визначення емоційної полярності та суб'єктивності тексту [19].

Метод не застосовує машинного навчання, а покладається на статичні лексикони, у яких кожному слову або словосполученню призначено два параметри: *polarity* (у діапазоні  $[-1, 1]$ ), що відображає емоційне забарвлення, та *subjectivity* (у діапазоні  $[0, 1]$ ), що оцінює ступінь суб'єктивного судження.

Для роботи TextBlob необхідний сирий текст, базова токенизація та доступ до лексиконів Pattern, які містять слова та багатослівні вирази з попередньо визначеними оцінками. Алгоритм працює шляхом лексичного аналізу: текст розбивається на речення, кожному елементу речення зі словника зіставляються значення *polarity* і *subjectivity*, після чого вони підсумовуються з урахуванням ваги, зокрема підсилювачів та врахування ступеня впевненості. Підсумкова полярність речення є усередненою зваженою сумою полярностей його складових, а загальна полярність тексту – середнім значенням полярності речень. *Subjectivity* обчислюється аналогічно, як середня зважена суб'єктивність усіх релевантних токенів.

На виході TextBlob надає два значення: *polarity* (визначає орієнтацію тексту від негативної до позитивної), *subjectivity* (відображає частку об'єктивного судження).

Переваги методу TextBlob полягають у наступному, а саме: забезпечує стабільну роботу на загальних англійських текстах, добре підходить для базових аналітичних задач та є інтерпретованим, оскільки кожне значення полярності має чітке словникове походження.

Недоліки методу VADER полягають у наступному, а саме: не моделює контекстуальних залежностей, не враховує складні синтаксичні структури,

працює слабше зі сленгом, сучасними неформальними мовними конструкціями та емодзі, а також не здатний уловлювати сарказм чи іронію.

## 2.5 Метрики якості прогнозування

Найпоширенішими метриками є Mean Squared Error (MSE), Root Mean Squared Error (RMSE) та Mean Absolute Error (MAE). Усі вони базуються на різних способах підрахунку величини помилок і застосовуються залежно від особливостей вибірки та чутливості до викидів.

**MSE** обчислює середнє значення квадратів різниць між фактичними та прогнозованими значеннями [20]. Діапазон значень  $[0, +\infty)$ . Формула:

$$MSE = 1/n \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

**RMSE** є квадратним коренем із MSE. Діапазон значень  $[0, +\infty)$ . Формула:  $RMSE = \sqrt{MSE}$ .

**MAE** обчислює середнє абсолютних різниць між фактичними та прогнозованими значеннями. Діапазон значень  $[0, +\infty)$ . Формула:

$$MAE = 1/n \sum_{i=1}^n |y_i - \hat{y}_i|.$$

## 2.6 Алгоритм побудови ансамблевої гібридної рекомендаційної системи

В даній роботі буде досліджено саме архітектуру ансамблевої гібридної рекомендаційної системи на прикладі стекінгу результатів декількох моделей та методів для задачі прогнозування рейтингів методами XGBoost, LR, та RF.

Дано:

- 1) множина унікальних користувачів  $U$ ,  $|U| = n_{users}$ ;

- 2) множина унікальних товарів  $I$ ,  $|I| = n_{items}$ ;
- 3) множина оглядів  $R$ ,  $|R| = n_{reviews}$ ;
- 4) множина унікальних описів товарів  $D$ ,  $|D| = n_{items}$ ;
- 5) матриця ознак  $X$  розмірності  $n_{reviews} \times 4$ ,  
 $X_{[:,0]} \in U$ ,  $X_{[:,1]} \in I$ ,  $X_{[:,2]} \in R$ ,  $X_{[:,3]} \in D$ ;
- 6) множина рейтингів  $y \in \mathbb{R}^{n_{reviews}}$ .

Потрібно: знайти прогнозовані рейтинги товарів  $\hat{y}$ .

Алгоритм побудови рекомендаційної системи.

1. Збір даних. Для роботи алгоритму потрібні  $X$  та  $y$ .
2. Пошук нових ознак.
  - 2.1. Подібності текстових описів та оглядів. Метод векторизації  $\phi_{vec}$ ,  
 $S^{vec} = \text{CosineSimilarity}(\phi_{vec}(X_{[:,2]}), \phi_{vec}(X_{[:,3]})) \in \mathbb{R}^n$ ,  
 $S = [S^{TFIDF} \ S^{Word2Vec} \ S^{SBERT}]$
  - 2.2. Прогнози моделей колаборативної фільтрації. Модель колаборативної фільтрації  $F^{CF}$ ,  $\hat{y}_{CF} = F^{CF}(X_{[:,\{0,1\}]})$ .  
 $\hat{y} = [\hat{y}_{SVDpp} \ \hat{y}_{SlopeOne} \ \hat{y}_{KNNBaseline} \ \hat{y}_{CoClustering}]$ .
  - 2.3. Сентимент-аналіз та суб'єктивність. Матриця сентимент-ознак  $Sentiment \in \mathbb{R}^{n \times 4}$  (міри позитивності, нейтральності, негативності, compound) методом VADER та вектор суб'єктивності  $Subjectivity \in \mathbb{R}^n$  методом TextBlob.
3. Стекінг. Створюється нова матриця ознак  $\bar{X} = [S \ Sentiment \ Subjectivity \ \hat{y}] \in \mathbb{R}^{n \times 12}$ . Можливе вилучення деяких ознак через високу чи дуже низьку кореляцію з рейтингом.
4. Поділ  $(\bar{X}, y)$  на тренувальну  $(\bar{X}_{train}, y_{train})$  та тестову  $(\bar{X}_{test}, y_{test})$  вибірки.

5. Ансамблева гібридна рекомендаційна система. Нарешті, після навчання моделей прогнозування  $F_{pr}$  ( $pr$  може бути одна з мета моделей RF, LR або XGBoost) на даних  $(\bar{X}_{train}, y_{train})$ , знаходимо фінальні прогнози рейтингів.
6. Тестування. Якість прогнозів визначається метриками MSE, RMSE, MAE на  $(\bar{X}_{test}, y_{test})$ .

## 2.7 Висновки до розділу 2

У другому розділі було систематизовано та проаналізовано моделі, що становлять теоретичну основу розробки ансамблевої гібридної рекомендаційної системи, орієнтованої на обробку текстових даних та прогнозування. Розглянуто декілька класів алгоритмів, кожен з яких виконує окрему функціональну роль у загальній архітектурі системи.

Досліджено сучасні методи векторизації текстів. TF-IDF формує розріджені матриці важливості термінів, Word2Vec дозволяє моделювати лексичні семантичні зв'язки, а Sentence-BERT генерує контекстно-залежні вектори речень, які забезпечують найкращу якість у задачах семантичного пошуку. Додатково було проаналізовано міри подібності, що використовуються для порівняння векторних подань текстів та базових об'єктів.

У межах методів колаборативної фільтрації розглянуто класичні та сучасні підходи: матричну факторизацію як основу багатьох моделей, SVDpp, що враховує неявну активність користувачів, CoClustering, який одночасно кластеризує користувачів і об'єкти, SlopeOne та KNNBaseline як легкі моделі з гармонійним балансом між обчислювальною ефективністю та точністю. Ці моделі забезпечують структурування уподобань користувачів на основі даних взаємодії.

Описано XGBoost як високопродуктивний ансамблевий алгоритм із регуляризацією, лінійну регресію як базову інтерпретовану модель та Random Forest як метод усереднення великої кількості незалежних дерев. Кожен з них може використовуватися як мета-модель ансамблевої гібридної рекомендаційної системи для моделювання рейтингових або поведінкових даних.

У розділі також представлено методи аналізу настрою текстів (VADER, TextBlob), які забезпечують оцінку емоційного забарвлення оглядів і коментарів користувачів. Ця інформація може бути використана як додаткове джерело сигналів щодо задоволеності або незадоволеності продуктами.

Далі розглянуто метрики якості прогнозування (MSE, RMSE, MAE), які відіграють ключову роль у кількісному оцінюванні ефективності моделей, дозволяючи здійснювати порівняння між ними та визначати оптимальні моделі.

## РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

### 3.1 Опис середовища розробки

Дана робота виконана за допомогою середовища Google Colab та мови програмування Python.

Google Colab є хмарним інтерактивним середовищем для виконання Python-коду, орієнтованим на роботу з даними, машинне навчання та наукові обчислення. Воно базується на технології Jupyter Notebook і дозволяє запускати блоки коду без встановлення локального програмного забезпечення. Користувачі працюють у веб-браузері, а обчислення виконуються на віддалених серверах Google, що робить Colab доступним навіть на слабких пристроях. Середовище підтримує інтеграцію з Google Drive, імпорт даних різного формату, використання численних бібліотек Python та можливість швидкого прототипування моделей.

До ключових переваг Google Colab належать безкоштовний доступ до GPU (Graphics Processing Unit) та TPU (Tensor Processing Unit), що значно прискорює навчання моделей глибокого навчання. Середовище містить заздалегідь встановлені популярні бібліотеки (NumPy, Pandas, TensorFlow, PyTorch, scikit-learn), що економить час на конфігурацію. Воно дозволяє легко ділитися ноутбуками, коментувати результати, відтворювати експерименти та забезпечує повну реплікабельність коду. Робота у Colab спрощує командну взаємодію, підтримує інтеграцію з GitHub та забезпечує автоматичний доступ до хмарного сховища Google Drive.

Попри переваги, Google Colab має низку обмежень. Сесії мають часові ліміти, після яких під'єднання переривається, а локальні змінні втрачаються, якщо їх не збережено у Drive. Виділені ресурси (кількість оперативної пам'яті, потужність GPU) нестабільні та залежать від поточного завантаження

серверів Google. Безкоштовна версія може вимикати GPU під великим навантаженням або обмежувати час бездіяльності. Також Colab не підходить для довготривалих або великих виробничих обчислень, адже не гарантує постійну доступність ресурсів.

### 3.2 Опис бібліотек Python

В даній роботі використані нижче описані бібліотеки у даному пункті.

**Gensim** є бібліотекою для обробки природної мови, зосередженою на навчанні тематичних моделей та векторних представлень слів. Вона реалізує алгоритми Word2Vec, Doc2Vec, FastText та LDA, дозволяючи будувати семантичні моделі великих текстових корпусів. Gensim оптимізована для роботи з великими даними завдяки потоковій обробці та низьким вимогам до пам'яті.

**NumPy** є бібліотекою для наукових обчислень у Python, що надає багатовимірні масиви та високопродуктивні математичні операції. Вона слугує основою для багатьох бібліотек з машинного навчання, забезпечуючи швидку обробку числових даних. NumPy підтримує лінійну алгебру, трансформації, генерацію випадкових чисел та різні математичні функції на рівні низькорівневих оптимізацій.

**Pandas** є бібліотекою для роботи з даними у Python, надаючи гнучкі структури даних Series і DataFrame. Бібліотека дозволяє легко виконувати очищення, трансформацію, групування, агрегацію та аналіз табличних наборів даних. Pandas широко використовується у машинному навчанні, аналітиці та ETL-процесах завдяки своїй продуктивності та інтерфейсу, подібному до R.

**NLTK** є бібліотекою для обробки природної мови у Python, що містить інструменти для токенізації, стемінгу, лематизації та аналізу синтаксису. Вона включає численні корпуси та лексичні ресурси, як-от WordNet, а також засоби

для базового аналізу текстів. NLTK часто використовується для навчальних цілей і швидкого прототипування лінгвістичних моделей.

**TextBlob** є бібліотекою для базової обробки англійських текстів. Вона забезпечує API (Application Programming Interface) для токенізації, частиномовного тегування, визначення тональності та перекладу. TextBlob зручний для швидких проєктів, де потрібна базова обробка тексту без налаштування складних моделей обробки природної мови.

**SentenceTransformers** є бібліотекою, що побудована на основі BERT-подібних моделей і надає можливість генерувати семантичні векторні представлення речень. Вона оптимізована для задач семантичного пошуку, кластеризації, порівняння текстів та побудови пошукових систем. Бібліотека містить великий набір попередньо навчених моделей і дозволяє донавчати їх на власних даних.

**Scikit-learn (sklearn)** є бібліотекою для машинного навчання, що містить широкий спектр алгоритмів: класифікації, регресії, кластеризації, методів зниження розмірності та ансамблевих моделей. Вона надає однаковий інтерфейс для всіх моделей, що значно спрощує експерименти. Scikit-learn також включає інструменти для попередньої обробки даних, оцінювання моделей.

**XGBoost** є бібліотекою для градієнтного бустингу, яка використовує оптимізовані алгоритми для роботи з великими та складними наборами даних. Вона забезпечує надзвичайно точні результати завдяки регуляризації та гнучкому налаштуванню гіперпараметрів. XGBoost підтримує GPU-прискорення, ранню зупинку та інтеграцію зі sklearn.

**Surprise** є бібліотекою, що спеціалізується на побудові рекомендаційних систем, особливо колаборативної фільтрації. Бібліотека реалізує багато алгоритмів, включно з SVD, SlopeOne, CoClustering та KNN-методами. Вона надає інструменти для оцінювання якості рекомендацій і роботи з користувачькими рейтингами.

**Seaborn** є бібліотекою для статистичної візуалізації, побудованою поверх Matplotlib, яка пропонує привабливі стилі та високорівневі функції. Вона спрощує побудову складних графіків, таких як heatmap, pairplot, violinplot чи кластерні карти. Seaborn добре підходить для аналізу розподілів та взаємозв'язків між змінними.

**Matplotlib** є бібліотекою для створення графіків у Python і підтримує практично всі типи 2D-візуалізацій. Вона дозволяє повністю контролювати стиль, кольори, підписи та макет графіків. Matplotlib широко використовується як самостійно, так і як основа для інших бібліотек візуалізації.

### 3.3 Огляд наборів даних

Для практичної реалізації рекомендаційної системи було використано відкритий набір даних із проєкту Amazon Product Graph Dataset від Стенфордського університету (SNAP Group). Ця колекція є однією з найвідоміших і найповніших баз оглядів користувачів Amazon, яка застосовується у дослідженнях машинного навчання, аналізу текстів та побудови рекомендаційних систем [5].

У роботі використовувався тематичні піднабори Office Products (OP) та Automotive (AM), які складаються з двох основних файлів, а саме:

- 1) reviews\_Office\_Products\_5.json.gz (reviews\_Automotive\_5.json.gz) – містить користувацькі огляди про товари;
- 2) meta\_Office\_Products.json.gz (meta\_Automotive.json.gz) – містить метадані про кожен товар .

Обидва файли мають формат JSON Lines, тобто кожен рядок – окремий JSON-об'єкт. Архіви зберігаються у форматі gzip і розпаковуються безпосередньо під час зчитування. Вся інформація у наборі англійською мовою.

Структура даних у файлах reviews така, а саме:

- 1) **reviewerID** – унікальний ідентифікатор користувача (тип string);
- 2) **asin** – ідентифікатор товару (Amazon Standard Identification Number) (тип string);
- 3) **reviewText** – повний текст огляду користувача (тип string);
- 4) **overall** – оцінка товару (від 1 до 5) (тип int);
- 5) **summary** – короткий підсумок або заголовок огляду (тип string);
- 6) **unixReviewTime / reviewTime** – дата публікації огляду у форматі UNIX або текстовому вигляді (тип object).

Структура даних у файлах meta така, а саме:

- 1) **asin** – ідентифікатор товару (тип string);
- 2) **title** – назва продукту (тип string);
- 3) **price** – ціна (тип double);
- 4) **brand** – бренд (тип string);
- 5) **imUrl** – шлях до зображення товару (тип string);
- 6) **categories** – ієрархія категорій, до яких належить товар (тип list);
- 7) **description** – опис товару (тип string);
- 8) **related** – словник пов'язаних товарів (тип dict);
- 9) **salesRank** – рейтинг продажів у відповідній категорії (тип dict).

Поля **reviewerID**, **asin** та **overall** з наборів **reviews** використовуються у задачі колаборативної фільтрації. Поле **reviewText** з наборів **reviews** та поле **description** з **meta** використовуються для косинусної подібності після текстової векторизації. Також, поле **reviewText** використовується для сентимент-аналізу та визначення суб'єктивності.

Самі файли **reviews** та **meta** для кожного тематичного піднабору завантажуються у об'єкти **pandas.DataFrame** та з **meta** використовуються лише і тільки лише дані про ті товари, **asin** яких хоч один раз був у наборі **reviews**.

Перші 5 рядків набору даних **reviews\_Automotive\_5.json.gz** завантажених у **pandas.DataFrame** відображено на рис. 3.1.

reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
A3F73SC1LY51OO	B00002243X	Alan Montgomery	[4, 4]	I needed a set of jumper cables for my new car...	5	Work Well - Should Have Bought Longer Ones	1313539200	08 17, 2011
A20S66SKYXULG2	B00002243X	alphonse	[1, 1]	These long cables work fine for my truck, but ...	4	Okay long cables	1315094400	09 4, 2011
A2I8LFSN2IS5EO	B00002243X	Chris	[0, 0]	Can't comment much on these since they have no...	5	Looks and feels heavy Duty	1374710400	07 25, 2013
A3GT2EWQSO45ZG	B00002243X	DeusEx	[19, 19]	I absolutley love Amazon!!! For the price of ...	5	Excellent choice for Jumper Cables!!!	1292889600	12 21, 2010

Рисунок 3.1 – Перші 5 рядків набору даних reviews\_Automotive\_5.json.gz завантажених у pandas.DataFrame

Перші 5 рядків набору даних meta\_Automotive.json.gz завантажених у pandas.DataFrame відображено на рис. 3.2.

	asin	categories	description	title	price	imageUrl	brand	related	salesRank
0	0219400083	[[Automotive, Lights & Lighting Accessories, L...	HID Xenon high and low beam lighting system pr...	Can-Am 219400083 HID Xenon Lighting System	654.99	http://ecx.images-amazon.com/images/I/415rqKdW...	Can-Am	{'also_viewed': ['B006IEOIZO', 'B00AFWJGB2', '...	NaN
1	0715000322	[[Automotive, Motorcycle & Powersports, Parts,...	Keep your hands warm while riding with this ea...	Can-Am 715000322 ATV Heated Hand Grip Kit	72.94	http://ecx.images-amazon.com/images/I/417FMGcl...	Can-Am	{'also_viewed': ['B000GZLKEO', 'B00A8MOZJC', '...	NaN
2	0970408641	[[Automotive, Replacement Parts, Window Regula...	Roll 2 windows up; Automatic temperature Contr...	Scytek ACCWR-8 2 Windows Roll-up Module	22.95	http://ecx.images-amazon.com/images/I/41trxn9p...	ScyTek Electronics	{'also_viewed': ['B0009SWLEQ', 'B004IAC2EA']}	NaN
3	1940825172	[[Automotive, Exterior Accessories, Towing Pro...	Kampflauer IV-A - Jagdluther - Model - Dust T...	Kampflauer Iv-a - Jagdluther	29.53	http://ecx.images-amazon.com/images/I/51kYgec5...	Dust Tactics	{'also_bought': ['1616612231', '1616611642', '...	{'Toys & Games': 371545}
4	2409862403	[[Automotive, Interior Accessories, Antitheft,...	Description:\nThis Keyless Entry System offers...	Car Remote Central Lock Kit Keyless Entry Syst...	13.92	http://ecx.images-amazon.com/images/I/41L9xim...	NaN	{'buy_after_viewing': ['B006QH9C5A', 'B001ANXN...	NaN

Рисунок 3.2 – Перші 5 рядків набору даних meta\_Automotive.json.gz завантажених у pandas.DataFrame

Порівняння піднаборів Amazon Reviews наведено у табл. 3.1.

Таблиця 3.1 – Порівняння піднаборів Amazon Reviews

Характеристика	Automotive	Office Products
Кількість коментарів	20473	53258
Кількість користувачів	2928	4905
Кількість товарів	1835	2420
Розрідженість	0.996	0.995
Пропуски у даних	-	-

Порівняння текстових полей піднаборів (кількості слів) Amazon Reviews та Amazon Meta наведено у табл. 3.2.

Таблиця 3.2 – Порівняння текстових полей (кількості слів) піднаборів Amazon Reviews та Amazon Meta

Характеристика	reviewText (OP)	reviewText (AM)	description (OP)	description (AM)
Кількість унікальних записів	53258	20473	2420	1835
Mean	145	85	106	67
Std	160	99	211	84
Min	0	0	0	0
Квартиль 25%	55	31	0	24
Квартиль 50%	101	52	42	49
Квартиль 75%	177	99	100	88
Max	5494	2239	3538	1862

Порівняння розподілів оцінок у підборах Amazon Reviews наведено у табл. 3.3.

Таблиця 3.3 – Порівняння розподілів оцінок у піднаборах Amazon Reviews

Рейтинг	Automotive	Office Products
5	13928	30327
4	3967	15015
3	1430	5060
2	606	1726
1	542	1130

Порівняння піднаборів Amazon Reviews та метаданих Amazon Meta демонструє суттєві відмінності у структурі, обсязі й характеристиках текстових та рейтингових даних, що безпосередньо впливає на вибір методів попередньої обробки та побудови рекомендаційних моделей. Піднабір Office Products містить у понад два рази більше коментарів, ніж Automotive, що одночасно супроводжується більшою кількістю користувачів та товарів. Незважаючи на це, рівень розрідженості матриці взаємодій у обох піднаборах залишається подібним (0.995–0.996), що підтверджує притаманну маркетплейсам нерівномірність активності користувачів. Подібна структура означає, що більшість користувачів залишають дуже мало оглядів, а більшість товарів отримують невелику кількість оцінок, що створює умови холодного старту як на рівні користувача, так і товару.

Аналіз текстових полів піднаборів виявив, що описи товарів Amazon Meta значно коротші за користувацькі огляди, а їхня варіативність і розкид довжин є нижчими. У категорії Office Products огляди в середньому довші, ніж у Automotive, що свідчить про більш детальну взаємодію користувачів із продукцією цієї групи. Натомість описи товарів мають більшу дисперсію та окремі дуже великі записи, що може впливати на якість моделей, побудованих на основі TF-IDF або векторних представлень слів. Квартильний аналіз демонструє наявність значної кількості коротких описів (мідіана для

деяких піднаборів дорівнює нулю), що підкреслює потребу в методах, стійких до пропусків та коротких текстів, зокрема в моделях типу Sentence-BERT.

Окремо варто відзначити значно зміщені розподіли оцінок у бік високих рейтингів, оскільки оцінка 5 домінує в обох піднаборах, що є типовою властивістю даних Amazon. Така нерівномірність створює виклики для регресійних моделей і може потребувати додаткових технік балансування або альтернативних функцій втрат. Загалом проведене порівняння підтверджує, що піднабори Amazon Reviews є різними за масштабом і структурою, але мають подібні закономірності, властиві даним з маркетплейсів. Це робить їх придатними для тестування гібридних рекомендаційних алгоритмів, які одночасно враховують тексти оглядів, описи товарів та числові рейтинги.

### **3.4 Попередня обробка даних**

На етапі попередньої обробки кожен текст (`reviewText (OP)`, `reviewText (AM)`, `description (OP)`, `description (AM)`) перетворювався на послідовність токенів за допомогою функції `simple_preprocess(str(text))`. Далі видалялися англійські стоп-слова. Завершальним кроком був стемінг, що виконувався через виклик `stemmer.stem(t)`, внаслідок чого зменшився розмір словника. Далі, до текстів застосовувалися методи векторизації, описані в пункті 2.1. Дані було поділено на тренувальну та тестову вибірки.

### **3.5 Аналіз отриманих результатів**

В даній роботі, щоб отримати нові ознаки для мета-моделей ансамблевої гібридної рекомендаційної системи (це для обох тематичних піднаборів) було використано 3 групи моделей, а саме: пошук міри подібності текстів; пошук сентимент-ознак; прогнозів рейтингів.

Налаштування моделей та методів нижче приведені для обох тематичних піднаборів.

Налаштування методу Word2Vec наведено у табл. 3.4.

Таблиця 3.4 – Налаштування методу Word2Vec

Параметр	Опис параметру	Значення
sentences	токенізовані речення	-
vector_size	кількість вимірів векторного представлення слів	100
window	максимальна відстань між цільовим словом і контекстом	5
min_count	мінімальна кількість появ слова в корпусі, щоб воно увійшло в словник	10
workers	кількість потоків для тренування	4
sg	тип архітектури (0 – CBOW, 1 – Skip-gram)	1

Налаштування методу TF-IDF наведено у табл. 3.5.

Таблиця 3.5 – Налаштування методу TF-IDF

Параметр	Опис параметру	Значення
input	вектор текстів	-
lowercase	конвертує весь текст у нижній регістр	True
preprocessor	функція попередньої обробки тексту	None
tokenizer	функція токенізації	None
token_pattern	регулярний вираз, що визначає, які послідовності символів вважаються токенами	r"(?u)\b\w+\b"
ngram_range	діапазон n-грам, які генеруються	(1,1)
min_df	відсікання рідкісних слів	10

Налаштування методу encode SBERT (all-MiniLM-L6-v2) наведено у табл. 3.6.

Таблиця 3.6 – Налаштування методу encode SBERT (all-MiniLM-L6-v2)

Параметр	Опис параметру	Значення
sentences	вектор текстів	-
batch_size	розмір батчу	256
convert_to_numpy	повернути масив Numpy	True

Порівняльна таблиця часу конвертування текстових даних (reviewText+description) для обох тематичних піднаборів наведена у табл. 3.7.

Таблиця 3.7 – Порівняльна таблиця часу в секундах конвертування текстових даних (reviewText+description) для обох тематичних піднаборів

Метод векторизації	Automotive	Office Products
Word2Vec	1.67	6.90
TF-IDF	0.83	3.64
SBERT	28.80	105.60

Далі проаналізуємо параметри налаштування моделей колаборативної фільтрації.

Налаштування SVDpp наведено у табл. 3.8.

Таблиця 3.8 – Налаштування SVDpp

Параметр	Опис параметру	Значення
n_factors	кількість латентних факторів	20
n_epochs	кількість повних проходів по тренувальному набору	20
random_state	фіксований seed для відтворюваності результатів	42

Налаштування CoClustering наведено у табл. 3.9.

Таблиця 3.9 – Налаштування CoClustering

Параметр	Опис параметру	Значення
n_cltr_u	кількість кластерів користувачів	4
n_cltr_i	кількість кластерів товарів	4
n_epochs	кількість ітерацій перевизначення кластерів	39

Налаштування SlopeOne не потребується.

Налаштування KNNBaseline наведено у табл. 3.10.

Таблиця 3.10 – Налаштування KNNBaseline

Параметр	Опис параметру	Значення
k	кількість найближчих сусідів	40
sim_options	опції, що визначають метод обчислення схожості	{"name": "cosine", "user_based": False}

Після навчання моделей, отримання результатів з методів, отримано наступні ознаки (для обох тематичних піднаборів), а саме:

- 1) pos, neu, neg, compound, subj – міри позитивності, нейтральності, негативності, compound показник та міра суб'єктивності текстових оглядів відповідно;
- 2) sim\_tfidf, sim\_w2v, sim\_bert – міри подібності між векторними представленнями reviewerText та відповідними description;
- 3) pred\_SVDpp, pred\_CoClustering, pred\_SlopeOne, pred\_KNNBaseline – спрогнозовані рейтинги (для тренувальної і тестової вибірки) за допомогою методів колаборативної фільтрації.

Описові статистики отриманих ознак з піднабору Automotive наведено на рис. 3.3.

	count	mean	std	min	25%	50%	75%	max
neg	20473.0	0.041331	0.049395	0.000000	0.000000	0.029000	0.066000	0.620000
neu	20473.0	0.800618	0.102738	0.000000	0.747000	0.813000	0.868000	1.000000
pos	20473.0	0.157757	0.101831	0.000000	0.090000	0.141000	0.210000	1.000000
compound	20473.0	0.556315	0.461812	-0.994800	0.361800	0.739100	0.897700	0.999800
subj	20473.0	0.519855	0.164755	0.000000	0.432875	0.519259	0.613333	1.000000
sim_tfidf	20473.0	0.156178	0.148204	0.000000	0.000000	0.127799	0.251429	0.890780
sim_w2v	20473.0	0.685744	0.324717	0.000000	0.708515	0.825241	0.882995	0.993690
sim_bert	20473.0	0.343390	0.201558	-0.155606	0.176569	0.374014	0.497989	0.882025
pred_SVDpp	20473.0	4.469866	0.421077	1.814231	4.263426	4.560487	4.778936	5.000000
pred_CoClustering	20473.0	4.408689	0.807445	1.000000	4.179031	4.714096	5.000000	5.000000
pred_SlopeOne	20473.0	4.449346	0.849745	1.000000	4.000000	4.925926	5.000000	5.000000
pred_KNNBaseline	20473.0	4.462102	0.533020	1.000000	4.213131	4.588552	4.878758	5.000000

Рисунок 3.3 – Описові статистики отриманих ознак з піднабору Automotive

Кореляція отриманих ознак з overall для піднабору Automotive наведена на рис. 3.4.

	overall
neg	-0.187578
neu	-0.133625
pos	0.227293
compound	0.239964
subj	0.099616
sim_tfidf	0.022013
sim_w2v	0.005062
sim_bert	0.029288
pred_SVDpp	0.694866
pred_CoClustering	0.707180
pred_SlopeOne	0.854971
pred_KNNBaseline	0.557594

Рисунок 3.4 – Кореляція отриманих ознак з overall для піднабору Automotive

Описові статистики отриманих ознак з піднабору Office Products наведено на рис. 3.5.

	count	mean	std	min	25%	50%	75%	max
neg	53258.0	0.035733	0.039381	0.000000	0.000000	0.027000	0.054000	1.0000
neu	53258.0	0.809700	0.084493	0.000000	0.767000	0.821000	0.865000	1.0000
pos	53258.0	0.154174	0.083912	0.000000	0.098000	0.141000	0.196000	1.0000
compound	53258.0	0.707401	0.422707	-0.996200	0.659700	0.891000	0.961700	0.9999
subj	53258.0	0.525467	0.125554	0.000000	0.453571	0.520000	0.595000	1.0000
sim_tfidf	53258.0	0.166495	0.170916	0.000000	0.000000	0.133952	0.291666	1.0000
sim_w2v	53258.0	0.572221	0.410258	0.000000	0.000000	0.829929	0.896406	1.0000
sim_bert	53258.0	0.353758	0.262392	-0.138235	0.058135	0.431788	0.580123	1.0000
pred_SVDpp	53258.0	4.342347	0.468485	1.005316	4.088311	4.423997	4.684453	5.0000
pred_CoClustering	53258.0	4.308609	0.668579	1.000000	3.963078	4.445292	4.857925	5.0000
pred_SlopeOne	53258.0	4.328779	0.781982	1.000000	4.000000	4.583366	4.982442	5.0000
pred_KNNBaseline	53258.0	4.339683	0.520718	1.000000	4.059063	4.419278	4.723850	5.0000

Рисунок 3.5 – Описові статистики отриманих ознак з піднабору Office Products

Кореляція отриманих ознак з overall для піднабору Office Products наведена на рис. 3.6.

	overall
neg	-0.240082
neu	-0.170829
pos	0.286424
compound	0.295442
subj	0.116947
sim_tfidf	-0.000800
sim_w2v	-0.002418
sim_bert	0.016367
pred_SVDpp	0.676815
pred_CoClustering	0.648665
pred_SlopeOne	0.850677
pred_KNNBaseline	0.571377

Рисунок 3.6 – Кореляція отриманих ознак з overall для піднабору Office Products

В подальшому, ознаку `pred_SlopeOne` буде прибрано для обох піднаборів, оскільки має високу кореляцію з `overall`. Решта ознак, та `overall` як цільова змінна передаються для тренування та тестування мета-моделей.

Налаштування XGBoost наведено у табл. 3.11.

Таблиця 3.11 – Налаштування XGBoost

Параметр	Опис параметру	Значення
<code>n_estimators</code>	кількість дерев	600
<code>learning_rate</code>	крок навчання	0.05
<code>max_depth</code>	глибина дерев	8
<code>subsample</code>	частка рядків для кожного дерева	0.9
<code>colsample_bytree</code>	частка ознак для кожного дерева	0.8
<code>reg_lambda</code>	L2 регуляризація	2
<code>reg_alpha</code>	L1 регуляризація	1

Налаштування Random Forest наведено у табл. 3.12.

Таблиця 3.12 – Налаштування Random Forest

Параметр	Опис параметру	Значення
<code>n_estimators</code>	кількість дерев	300
<code>min_samples_split</code>	мінімальна кількість прикладів для розбиття вузла	5
<code>min_samples_leaf</code>	мінімальна кількість прикладів у листі	2

Налаштування Linear Regression не потребується.

Порівняння значень метрик якості для моделей натренованих на Automotive наведені у табл. 3.13. Жовтим виділені моделі колаборативної фільтрації, зеленим – моделі гібридної фільтрації.

Таблиця 3.13 – Порівняння значень метрик якості для моделей натренованих на Automotive

Рекомендаційна система	MSE	RMSE	MAE	Time
SVDpp	<b>0.78</b>	<b>0.88</b>	<b>0.63</b>	0.001
CoClustering	1.13	1.06	0.66	0.001
SlopeOne	1.08	1.04	0.68	0.001
KNNBaseline	0.93	0.96	0.64	0.001
STACK_RF	<b>0.94</b>	<b>0.97</b>	<b>0.60</b>	0.170
STACK_XGB	<b>0.94</b>	<b>0.97</b>	<b>0.60</b>	0.080
STACK_LR	1.07	1.03	0.69	<b>0.001</b>

Порівняння значень метрик якості для моделей натренованих на Office Products наведені табл. 3.14. Значення кольорів ті самі, як у табл. 3.13.

Таблиця 3.14 – Порівняння значень метрик якості для моделей натренованих на Office Products

Рекомендаційна система	MSE	RMSE	MAE	Time
SVDpp	<b>0.71</b>	<b>0.85</b>	<b>0.62</b>	0.001
CoClustering	0.84	0.92	0.63	0.001
SlopeOne	0.90	0.94	0.67	0.001
KNNBaseline	0.80	0.90	0.64	0.001
STACK_RF	<b>0.78</b>	<b>0.88</b>	<b>0.61</b>	0.490
STACK_XGB	0.79	0.89	<b>0.61</b>	0.228
STACK_LR	0.84	0.92	0.67	<b>0.002</b>

Ваги ознак у мета-моделей, натренованих на отриманих ознаках та overall з набору Automotive наведено на рис. 3.7.

	STACK_RF	STACK_XGB	STACK_LR
neg	0.014158	0.025703	0.495713
neu	0.016976	0.022475	0.307439
pos	0.023882	0.041265	0.003345
compound	0.025245	0.028587	0.103319
subj	0.022018	0.025813	0.078137
sim_tfidf	0.013537	0.022598	0.051986
sim_w2v	0.015142	0.023668	0.043190
sim_bert	0.018189	0.024044	0.092845
pred_SVDpp	0.188991	0.223258	1.165557
pred_CoClustering	0.621668	0.513227	0.696742
pred_KNNBaseline	0.040193	0.049362	0.589554

Рисунок 3.7 – Ваги ознак у метамоделях стекінгу для набору Automotive

Ваги ознак у мета-моделей, натренованих на отриманих ознаках та overall з набору Office Products наведено на рис. 3.8.

	STACK_RF	STACK_XGB	STACK_LR
neg	0.027197	0.037146	0.703133
neu	0.023054	0.024963	0.058611
pos	0.032333	0.045372	0.586600
compound	0.059314	0.057288	0.206300
subj	0.031551	0.026686	0.085064
sim_tfidf	0.015782	0.026028	0.003318
sim_w2v	0.017023	0.027515	0.004237
sim_bert	0.027369	0.027248	0.014564
pred_SVDpp	0.615443	0.451933	1.496984
pred_CoClustering	0.082945	0.204716	0.575908
pred_KNNBaseline	0.067990	0.071106	0.863666

Рисунок 3.8 – Ваги ознак у метамоделях стекінгу для набору Office Products

Аналіз показує, що для всіх метамоделей найвагомішими ознаками є прогнози оцінок моделей колаборативної фільтрації, зокрема SVDpp та CoClustering. У Random Forest та XGBoost їхні значення важливості суттєво перевищують інші ознаки, що свідчить про сильну залежність метамоделей

від базових рекомендаційних алгоритмів. Це означає, що нелінійні стекінгові моделі будують прогноз здебільшого на основі інтеграції результатів факторизаційних методів, тоді як семантичні і тональні характеристики тексту виконують допоміжну роль. Водночас у лінійній моделі (STACK\_LR) коефіцієнти прогнозів моделей колаборативної фільтрації є найбільшими серед усіх ознак.

Семантичні подібності між описами та оглядами (`sim_tfidf`, `sim_w2v`, `sim_bert`) демонструють стабільно нижчу важливість, однак вони помітно підсилюють моделі XGBoost і RF, особливо у наборі Automotive, де структура текстів більш інформативна. Ознаки тональності (`neg`, `neu`, `pos`, `compound`) також займають другорядне місце, але покращують якість нелінійних моделей через врахування емоційної спрямованості користувачьких оглядів. У лінійній моделі вони мають значно вищі коефіцієнти, що свідчить про її чутливість до слабких кореляцій та неможливість моделювати складні взаємодії між ознаками.

### 3.6 Висновки до розділу 3

У роботі реалізовано гібридну рекомендаційну систему для прогнозування рейтингів товарів на основі даних Amazon Reviews, використовуючи середовище Google Colab та мову Python. Colab забезпечив хмарні обчислення, доступ до GPU та інтеграцію з Google Drive, що спростило експерименти з великими наборами даних, хоча й наклало обмеження на тривалість сесій та стабільність ресурсів.

Використано широкий набір бібліотек: для обробки текстів (`Gensim`, `NLTK`, `TextBlob`, `SentenceTransformers`), для машинного навчання (`scikit-learn`, `XGBoost`, `Surprise`), а також для аналізу та візуалізації даних (`NumPy`, `Pandas`, `Seaborn`, `Matplotlib`). `Surprise` застосовано для побудови моделей колаборативної фільтрації (`SVDpp`, `CoClustering`, `SlopeOne`, `KNNBaseline`), а

XGBoost, Random Forest та Linear Regression виступають як мета-регресори у стекінгу.

Експерименти проведено на двох тематичних піднаборах Amazon Product Graph Dataset – Automotive (20 473 огляди, 2 928 користувачів, 1 835 товарів) та Office Products (53 258 оглядів, 4 905 користувачів, 2 420 товарів) із високою розрідженістю матриці взаємодій (приблизно 0.995). Використано як файли оглядів (reviewerID, asin, reviewText, overall), так і метадані (asin, description). Розподіл рейтингів зміщений у бік оцінки 5, що створює додаткові виклики для регресійних моделей.

Попередня обробка текстів включала токенізацію, видалення англійських стоп-слів та пунктуації, стемінг, після чого застосовано три методи векторизації: Word2Vec, TF-IDF та SBERT. На основі тексту оглядів та описів товарів обчислювалися косинусні подібності `sim_tfidf`, `sim_w2v`, `sim_bert`. Додатково з `reviewText` було отримано сентимент-ознаки (`neg`, `neu`, `pos`, `compound`, `subj`).

З моделей колаборативної фільтрації (SVDpp, CoClustering, SlopeOne, KNNBaseline) отримано прогнозовані рейтинги, які разом із текстовими та сентимент-ознаками стали вхідними ознаками для моделей стекінгу Random Forest, XGBoost та Linear Regression. Ознаку `pred_SlopeOne` вилучено через надто високу кореляцію з цільовим рейтингом.

Порівняння результатів показує, що ансамблеві стекові моделі забезпечують лише часткове покращення якості прогнозування. Зокрема, `STACK_RF` та `STACK_XGB` демонструють найкращі значення MAE, але лише за цією метрикою – для піднабору Automotive MAE=0.60 (проти 0.63 у SVDpp), а для Office Products MAE=0.61 (проти 0.62 у SVDpp).

Водночас за метриками MSE та RMSE базова модель SVDpp чітко залишається лідером. Для Automotive SVDpp досягає MSE=0.78, RMSE=0.88, тоді як `STACK_RF` та `STACK_XGB` мають MSE=0.94, RMSE=0.97, що є гіршими значеннями. Аналогічно для Office Products, SVDpp (MSE=0.71, RMSE=0.85) випереджає стекові моделі (MSE≈0.78–0.79, RMSE≈0.88–0.89).

Крім того, час виконання стекових моделей суттєво більший: наприклад, STACK\_RF потребує 0.170 с для Automotive та 0.490 с для Office Products, тоді як SVDpp працює за 0.001 с.

Отже, ансамблеві гібридні моделі забезпечують локальне покращення лише за MAE, але не перевершують SVDpp за MSE та RMSE і мають більші обчислювальні витрати. Це означає, що їхня доцільність залежить від конкретних вимог системи. Якщо важлива абсолютна точність прогнозу – стекінг має перевагу; якщо ж критичні квадратичні помилки або швидкодія – базові методи колаборативної фільтрації є ефективними.

## РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ

Стартапи сьогодні формують динаміку цифрової економіки, адже саме вони здатні швидко перевіряти гіпотези, впроваджувати нестандартні технологічні підходи та оперативно реагувати на зміни попиту. У межах цього проєкту передбачається створення рекомендаційної системи нового покоління, яка поєднує кілька джерел інформації та різні моделі машинного навчання для забезпечення точної персоналізації.

Зі зростанням кількості онлайн-платформ та масштабуванням даних, що виникають під час взаємодії користувачів із товарами, стає очевидною потреба у гнучких аналітичних інструментах. Традиційні підходи персоналізації вже не забезпечують достатньої якості, тому перспективним напрямом є моделювання, яке поєднує сигнали з текстових оглядів, параметри схожості між оглядами та описами товарів, а також прогнози моделей колаборативної фільтрації.

Основне завдання стартап-проєкту полягає у створенні стекової рекомендаційної системи, яка об'єднує результати кількох моделей: аналізу тональності оглядів, вимірювання текстової подібності та факторизаційних моделей, що прогнозують рейтинги. Завдяки інтеграції цих різномірних ознак метамодель зможе формувати точніші рекомендації та відображати реальні інтереси користувачів.

Електронна комерція продовжує демонструвати стрімке зростання, а разом із цим посилюється потреба бізнесу у високоточних алгоритмах персоналізованого підбору товарів. Запропонований стартап орієнтується на компанії, які прагнуть збільшити конверсію та утримання клієнтів за допомогою інтелектуальних рекомендацій, але не мають можливості інвестувати у власні складні рішення у галузі машинного навчання. Завдяки комплексному підходу система може посісти конкурентну позицію на ринку,

запропонувавши сервіс, що поєднує аналітичну глибину та практичну ефективність для малого та середнього бізнесу.

#### 4.1 Опис ідеї стартап-проекту

Стартап-проект спрямований на створення інтелектуальної гібридної рекомендаційної системи нового покоління, що поєднує колаборативну фільтрацію, семантичний аналіз текстових оглядів, емоційний аналіз, векторні подібності, прогнози моделей колаборативної фільтрації та метарегресійні моделі для формування точних персоналізованих рекомендацій.

Суть продукту полягає у тому, що система опрацьовує історію взаємодії користувачів із товарами, їхні текстові огляди, метадані про продукти та формує прогноз рейтингу або рекомендаційний список з підвищеною точністю завдяки поєднанню гібридних ознак. Запропонована архітектура формує мета-ознаки, які об'єднують різні джерела інформації.

У табл. 4.1 наведена інформаційна карта стартапу.

Таблиця 4.1 – Інформаційна карта стартап-проекту

<b>Назва проєкту</b>	MetaRecomEngine
<b>Автори</b>	Артеменко Євгеній Вячеславович
<b>Анотація</b>	Інтелектуальна рекомендаційна система, що комбінує алгоритми колаборативної фільтрації, семантичного аналізу та метарегресії для точного прогнозування рейтингів і формування персональних рекомендацій.
<b>Термін реалізації</b>	12 місяців
<b>Необхідні ресурси</b>	Обчислювальні сервери або хмарні GPU-інстанси; сховища даних; програмне забезпечення для розробки у сфері машинного навчання; аналітичні інструменти; фінансування на 12 місяців; команда технічних спеціалістів.

## Продовження таблиці 4.1

<b>Опис проблеми, яку вирішує проєкт</b>	Низька точність класичних рекомендаційних систем у разі нестачі даних, відсутність комплексного використання текстових оглядів та емоційного контексту, а також недостатня персоналізація сучасних моделей.
<b>Головні цілі та завдання проєкту</b>	Розробити гібридну рекомендаційну систему, що поєднує факторизаційні, текстові та семантичні моделі; сформувати мета-ознаки; інтегрувати стекінг для підвищення точності прогнозів; забезпечити масштабованість та універсальність системи.
<b>Очікувані результати</b>	Створення функціонального MVP для e-commerce; демонстраційна API-платформа для бізнес-клієнтів; підтвердження переваги над базовими методами CF; можливість комерційного впровадження як SaaS-сервісу.

У наступному підпункті буде наведено технологічний аудит та початковий конкурентний аналіз ідеї стартапу.

#### 4.2 Технологічний аудит ідеї проєкту

Система базується на гібридному підході, що поєднує алгоритми матричної факторизації, сучасні методи обробки природної мови та стекінгові метарегресори. Для обчислювальної частини планується використання Python та фреймворків машинного навчання.

У табл. 4.2 наведений опис ідеї стартапу.

Таблиця 4.2 – Опис ідеї стартапу

<b>Зміст ідеї</b>	<b>Напрямки застосування</b>	<b>Вигоди для користувача</b>
Створення гібридної рекомендаційної системи, що поєднує колаборативні, контентні, семантичні та емоційні ознаки у рамках мета-моделі	Інтернет-магазини та маркетплейси (Amazon, Rozetka, Prom)	Підвищення релевантності рекомендацій
	Аналітичні та маркетингові платформи, що працюють із поведінковими даними	Краща персоналізація, збільшення конверсії та задоволеності користувачів

Порівняльний аналіз конкурентів проекту у табл. 4.3.

Таблиця 4.3 – Порівняльний аналіз конкурентів проекту

№ п/п	Техніко-економічні хар-ки ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Власний проект	Amazon	Rozetka	Prom			
1	Точність прогнозування	Висока завдяки мета-ознакам і стекінгу	Висока, але базується переважно на поведінкових даних	Невідомо	Невідомо		+	
2	Використання текстових оглядів	Глибока інтеграція семантичних та емоційних ознак	Використовується частково, без емоційної складової	Використовується мінімально	Використовується мінімально		+	
3	Гнучкість інтеграції	API, SaaS, кастомізація під клієнта	Висока, оплата за запити	Відсутня	Відсутня			+

Технологічна здійсненність наведена у табл. 4.4.

Таблиця 4.4 – Технологічна здійсненність продукту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Створення гібридної рекомендаційної системи, що поєднує колаборативні, контентні, семантичні та емоційні ознаки у рамках мета-моделі	Використання мови програмування Python	Наявні	Доступні
2		Використання мови програмування Java	Наявні, необхідні допрацювання	Доступні
3		Використання мови програмування C++	Не наявні	Доступні
Обрана технологія реалізації ідеї проекту: Python				

Проаналізувавши таблицю, можна прийти до висновку, що технічна реалізація проекту можлива.

### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Попередній аналіз ринку наведено у табл. 4.5. Характеристика потенційних клієнтів наведена у табл. 4.6.

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3 (Amazon, Rozetka, Prom)
2	Загальний обсяг продаж, грн/ум.од	10 000+ млн грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Технічні бар'єри низькі, конкуренція середня
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні спеціальні регулятивні вимоги
6	Середня норма рентабельності в галузі (або по ринку), %	15–20%

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреби, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Підвищення конверсії та продажів	Інтернет-магазини та маркетплейси	Орієнтовані на ROI, очікують швидкого ефекту	Висока точність рекомендацій, масштабованість
2	Оптимізація процесів аналітики	Малі та середні бізнеси	Потребують готових рішень без складної інтеграції	Низька вартість, просте API
3	Автоматизація роботи	Маркетинг-агентства, CRM	Очікують персоналізацію	Семантичний аналіз, гнучкість моделі

Обрахуємо фактори загроз (табл. 4.7) та можливостей (табл. 4.8). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску продукту на ринок. Фактори можливостей треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 4.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Сильні гравці ринку вже використовують власні алгоритми	Акцент на унікальних мета-ознаках та точності
2	Обмежений доступ до даних	Е-commerce платформи не завжди надають великі датасети	Використання відкритих даних (Amazon Dataset), пропозиція інтеграцій
3	Технічна складність	Підвищені вимоги до обчислювальних ресурсів	Модульна архітектура, використання хмарних рішень

Таблиця 4.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання е-commerce	Потреба в персоналізованих рекомендаціях збільшується	Розробка SaaS-рішення з простим API
2	Попит на аналіз текстів	Маркетинг і аналітика переходять до семантичних моделей	Акцент на SBERT та сентимент-аналізі
3	Низькі бар'єри входу	Можливість зайняти нішу гібридних моделей	Активна маркетингова комунікація та MVP

Щоб оцінити силу ринкових суперників, зрозуміти межі ціноутворення, визначити конкурентні переваги та вибрати ефективну стратегію збуту. розглянемо питання конкуренції, а саме визначимо її тип та рівень (табл. 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія	Домінування одного технологічного лідера	Диференціація через інновації
2. За рівнем конкурентної боротьби - міжнародний	Гравці конкурують глобально за дані	Необхідність міжнародного масштабування швидко
3. За галузевою ознакою - внутрішньогалузева	Боротьба між платформами рекомендацій	Удосконалювати алгоритми швидше за конкурентів
4. Конкуренція за видами товарів - товарно-родова	Змагання між різними типами рекомендаційних систем	Показати вищу точність і персоналізацію
5. За характером конкурентних переваг - нецінова	Переваги формуються якістю, точністю, UX	Інвестувати в сучасні моделі та інтерфейс
6. За інтенсивністю - марочна	Сильні бренди \	Розвиток власного технологічного бренду

Аналіз конкуренції в галузі наведено у табл. 4.10.

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Інші існуючі системи та продукти	Якість, ціни, кількість користувачів, капіталовкладення	Фактори сили постачальників	Контроль якості, порівняння цін	Сила бренду, якість, ціна, масштаби
Висновки	Інтенсивність конкуренції – висока, домінують глобальні гравці	Нові потенційні конкуренти	Постачальники відсутні	Клієнти не диктують умови роботи на ринку	Товарозамінники відсутні

Маючи результати аналізу конкуренції, характеристики ідеї стартап-проекту, характеристики потенційних клієнтів і їх вимоги до продукту та фактори ринкового середовища, було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (табл. 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Універсальність	Працює у різних галузях та форматах даних
2	Простота у використанні	Легка інтеграція, інтуїтивна панель, швидкий запуск
3	Якість та гарантії	Висока точність моделей і стабільність результатів
4	Безкоштовний сервіс при MVP	Знижує ризики клієнта, стимулює тестування продукту

Тепер можна провести порівняльний аналіз сильних та слабких сторін продукту у табл. 4.12.

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін «MetaRecomEngine»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з MetaRecomEngine						
			-3	-2	-1	0	+1	+2	+3
1	Універсальність	18		+					
2	Простота у використанні	16		+					
3	Якість та гарантії	12		+					
4	Безкоштовний сервіс при MVP	18			+				

SWOT-аналіз - це виявлення Strengths (Сильні сторони), Weaknesses (Слабкі сторони), Opportunities (Можливості) та Threats (Загрози). SWOT-аналіз стартап-проекту наведено у табл. 4.13.

Таблиця 4.13 – SWOT-аналіз стартап-проекту

<b>Сильні сторони</b>	<b>Слабкі сторони</b>
Просте впровадження через API та панель керування	Обмежені обчислювальні ресурси
Висока точність рекомендацій у різних галузях	Невелика впізнаваність бренду на глобальному ринку
Гібридна модель	Залежність від сторонніх інфраструктур
<b>Можливості</b>	<b>Загрози</b>
Зростання потреб у персоналізації в e-commerce та SaaS	Конкуренція
Попит на інтерпретовані та етичні рекомендаційні системи	Ризики підвищення вартості хмарних ресурсів та GPU
Швидке масштабування через мультигалузевість продукту	Можливі регуляторні обмеження щодо використання даних

Далі спроектуємо альтернативи ринкового впровадження стартап-проекту у табл. 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап-проекту

<b>№ п/п</b>	<b>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</b>	<b>Ймовірність отримання ресурсів</b>	<b>Строки реалізації</b>
1	Тестування MVP з малими бізнесами	80%	2 місяці
2	Партнерство з маркетплейсами	60%	5 місяців
3	Інтеграція у SaaS-платформи	70%	3 місяці

#### 4.4 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (табл. 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі та середні e-commerce бізнеси	Висока	30%	Середня	Висока
2	SaaS-платформи та онлайн-сервіси	Середня	25%	Середня	Середня
3	Маркетплейси та великі ритейл-мережі	Середня	20%	Висока	Низька
4	Медійні та контентні платформи	Середня	15%	Середня	Середня
Які цільові групи обрано: 1, 2					

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1 та 2	Диференційовані й маркетинг	Простота інтеграції, універсальність, точність моделей	Стратегія диференціації

Базова стратегія та позиціонування відповідно у табл. 4.17, табл. 4.18.

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Так	Ні	Виклику лідера

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
Універсальність, простота, висока точність	Стратегія диференціації	Універсальність, простота, гарантії, MVP безкоштовно	Інтелектуальна точність, легкість інтеграції, повна надійність

#### 4.5 Розроблення маркетингової програми стартап-проекту

Ключові переваги концепції потенційного товару (табл. 4.19).

Таблиця 4.19 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точні персональні рекомендації	Підвищення конверсії та утримання	Гібридна модель високої точності
2	Просте впровадження AI	Швидкий запуск без технічних знань	Інтуїтивний інтерфейс та легка інтеграція
3	Мінімізація ризиків тестування	Оцінка ефекту без витрат	Безкоштовний MVP для перевірки цінності

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.20).

Таблиця 4.20 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Поєднання емоційного аналізу та векторних представлень текстових відгуків та описів товарів для формування релевантних рекомендацій у реальному часі		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Гл/Е/Ор
	1. Семантичний аналіз текстів 2. Емоційна класифікація відгуків 3. Векторні подібності контенту 4. Прогнози методів колаборативної фільтрації	1. Нм 2. Нм 3. Нм 4. Нм	1. Тх 2. Тх 3. Тх 4. Тх
	Якість: тестування точності, стабільності та швидкості роботи		
	Пакування: онлайн-панель, API-документація, технічний гайд		
	MetaRecomEngine by MetaRecom Labs		
III. Товар із підкріпленням	До продажу: технічні консультації, демонстрація MVP, допомога з інтеграцією та оцінкою ефективності системи		
	Після продажу: підтримка 24/7, регулярні оновлення моделей, аналітика ефективності та оптимізація рекомендацій під клієнта		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану

основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.21).

Таблиця 4.21 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Активні у цифрових бізнес-спільнотах	Вебінари, соцмережі, конференції	Простота, точність, економічність	Показати зростання продажів завдяки AI	Демонстрація переваг у реальному часі

#### 4.6 Висновки до розділу 4

У результаті проведеного маркетингового аналізу встановлено, що запропонований стартап-проект MetaRecomEngine має реальні перспективи ринкової комерціалізації. Аналіз стану ринку рекомендаційних систем показав наявність стійкого попиту з боку малих та середніх e-commerce компаній, SaaS-платформ та цифрових сервісів. Динаміка ринку є позитивною, а середня рентабельність галузі перевищує базову дохідність альтернативних інвестицій, що підтверджує економічну доцільність виходу на ринок.

З урахуванням визначених цільових сегментів, бар'єрів входження та умов конкурентного середовища встановлено, що проєкт володіє низкою значущих конкурентних переваг, зокрема: універсальністю застосування, простотою інтеграції, високою якістю результатів і можливістю безкоштовного тестування MVP. SWOT-аналіз продемонстрував, що сильні сторони проєкту дозволяють ефективно використати ринкові можливості, тоді як виявлені загрози можуть бути мінімізовані за рахунок правильно обраної ринкової стратегії.

На основі проведеного аналізу визначено, що найбільш доцільною альтернативою ринкового впровадження є комбінація тестування MVP з малими бізнесами та інтеграції в SaaS-платформи, що забезпечує швидке отримання ресурсів і скорочені строки виходу на ринок. Відповідно сформовано стратегію диференційованого маркетингу, базовану на чіткому позиціонуванні продукту як точного, простого у використанні та економічно вигідного інтелектуального рішення для персоналізації.

Отже, результати дослідження підтверджують доцільність подальшої розробки та впровадження стартап-проєкту MetaRecomEngine, а також його потенційну здатність зайняти конкурентну нішу на зростаючому ринку інтелектуальних рекомендаційних систем.

## ВИСНОВКИ

У даній роботі було досліджено застосування сучасних моделей та методів машинного навчання для розробки рекомендаційної системи прогнозування рейтингів користувачів на основі текстових оглядів користувачів та метаданих товарів.

Досліджено можливості класичних моделей колаборативної фільтрації (SVDpp, CoClustering, KNNBaseline та SlopeOne), методів векторизації текстів (TF-IDF, Word2Vec та SBERT), а також стекінгових метарегресорів (Random Forest, XGBoost, Linear Regression у задачі прогнозування рейтингів).

Розроблено ансамблеву гібридну рекомендаційну систему на основі поєднання нових ознак (сентиментних ознак з текстів оглядів, подібності текстових оглядів та описів товарів, прогнозів методів колаборативної фільтрації) для задачі прогнозування рейтингу. Проведено порівняльний аналіз впливу цих ознак на формування прогнозів гібридними системами, результатів прогнозів базових методів колаборативної фільтрації та ансамблевих гібридних рекомендаційних систем на основі метрик якості (MSE, RMSE, MAE).

Проведений аналіз розробки стартап-проєкту показав, що ансамблева гібридна рекомендаційна система має потенціал бути конкурентноздатною, так як використовує сучасні моделі та методи машинного навчання для точного визначення прогнозів рейтингів. Проведено аналіз потенційних клієнтів та ринкових сегментів, що дає можливість впровадження системи на ринок.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Falk K. Practical Recommender Systems. Shelter Island: Manning Publications. 2019. 462 p. (дата звернення 11.11.2025).
2. Hasan E., Rahman M., Ding C., Huang J. X., Raza S. Review-based Recommender Systems: A Survey of Approaches, Challenges and Future Perspectives. 2024. P. 1-35. URL: <https://arxiv.org/abs/2405.05562> (дата звернення 11.11.2025).
3. Rezaei M. R. Amazon Product Recommender System. 2021. P. 1-5. URL: <https://arxiv.org/abs/2102.04238> (дата звернення 11.11.2025).
4. Lin H. P. Amazon Books Rating Prediction & Recommendation Model. 2023. P. 1-5. URL: <https://arxiv.org/abs/2310.03200> (дата звернення 11.11.2025).
5. McAuley J. Amazon product data. University of California San Diego. URL: <https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html> (дата звернення 11.11.2025).
6. Feature extraction: TF-IDF term weighting. Scikit-learn User Guide. URL: [https://scikit-learn.org/stable/modules/feature\\_extraction.html#tfidf-term-weighting](https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting) (дата звернення 11.11.2025).
7. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space. 2013. P. 1-12. URL: <https://arxiv.org/abs/1301.3781> (дата звернення 11.11.2025).
8. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019. P. 1-11. URL: <https://arxiv.org/abs/1908.10084> (дата звернення 11.11.2025).
9. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press. 2008. 560 p. DOI 10.1017/CBO9780511809071. (дата звернення 11.11.2025).

10. Koren Y., Bell R. and Volinsky C. Matrix Factorization Techniques for Recommender Systems. P. 1-8. URL: [https://datajobs.com/data-science-repo/Recommender-Systems-\[Netflix\].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf) (дата звернення 11.11.2025).
11. Koren Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. New York. ACM. 2008. P. 426–434. DOI 10.1145/1401890.1401944. (дата звернення 11.11.2025).
12. George T., Merugu S.A. Scalable Collaborative Filtering Framework based on Co-clustering. ICDM 2005. Proceedings of the Fifth IEEE International Conference on Data Mining. P. 625–628. DOI: 10.1109/ICDM.2005.14 (дата звернення 11.11.2025).
13. Lemire D., Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering. 2007. P. 1-5. URL: <https://arxiv.org/abs/cs/0702144> (дата звернення 11.11.2025).
14. Koren Y. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. ACM. 2010. Vol. 4. № 1. P. 1-24. DOI 10.1145/1644873.1644874. (дата звернення 11.11.2025).
15. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. 2016. P. 1–13. URL: <https://arxiv.org/abs/1603.02754> (дата звернення 11.11.2025).
16. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2nd ed. New York. Springer. 2009. 764 p. DOI 10.1007/978-0-387-84858-7. (дата звернення 11.11.2025).
17. Breiman L. Random Forests. Machine Learning. 2001. Vol. 45. № 1. P. 5–32. DOI 10.1023/A:1010933404324. (дата звернення 11.11.2025).
18. Hutto C., Gilbert E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Conference: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media. At: Ann Arbor, Michigan, USA. 2014. P. 216–225. URL: <https://www.researchgate.net/publication/275828927> (дата звернення 11.11.2025).

19. TextBlob documentation. URL: <https://textblob.readthedocs.io/en/latest/>  
(дата звернення 11.11.2025)
20. Недашківська Н.І. Інтелектуальний аналіз даних : навч. посіб. для студ. спеціальності 124 «Системний аналіз», освітніх програм «Системний аналіз і управління», «Системний аналіз фінансового ринку», Київ : КПІ ім. Ігоря Сікорського, 2021. 105 с. (дата звернення 11.11.2025).
21. Артеменко Є.В., Недашківська Н.І. Рекомендаційні системи з використанням текстів оглядів користувачів та ансамблювання на основі стекінгу. Системні науки та інформатика: збірник доповідей IV Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 1–5 грудня 2025 року, Київ. Київ: НН ІПСА КПІ ім. Ігоря Сікорського. 2025. 5 с.(дата звернення 11.11.2025).

## ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```

## Завантаження бібліотек

# !pip install -q gensim sentence_transformers surprise nltk
# !pip uninstall numpy
# !pip install numpy==1.26.4

import ast
import time
import json
import gzip
import time
import gensim

import numpy as np
import pandas as pd

import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from textblob import TextBlob
from gensim.models import Word2Vec
from sentence_transformers import SentenceTransformer

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import (
    mean_squared_error,
    mean_absolute_error,
)
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

from xgboost import XGBRegressor

from surprise import Dataset, Reader, accuracy
from surprise import SVDpp, CoClustering, SlopeOne, KNNBaseline

import seaborn as sns
import matplotlib.pyplot as plt

nltk.download("vader_lexicon")
nltk.download("stopwords")

stop_words = set(stopwords.words("english"))

## Завантаження даних з Amazon

# task="Office_Products"
task="Automotive"

!wget -q
https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_{task}
_5.json.gz
!wget -q
http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/meta_{task}.jso
n.gz

```

```

with gzip.open(f"meta_{task}.json.gz", "rt") as f:
    records = [ast.literal_eval(line) for line in f]

meta = pd.DataFrame(records)

with gzip.open(f"reviews_{task}_5.json.gz", "rt") as f:
    reviews = pd.read_json(f, lines=True)

meta.info()

meta.head()

reviews.info()

reviews.head()

n_ratings = len(reviews)
n_users    = reviews['reviewerID'].nunique()
n_items    = reviews['asin'].nunique()

sparsity = 1 - n_ratings / (n_users * n_items)

n_ratings, n_users, n_items, sparsity

reviews['overall'].value_counts()

pd.Series(reviews['reviewText'].astype(str).str.split().str.len()).describe()

meta = meta[meta['asin'].isin(reviews['asin'].unique())].reset_index(drop=True)
meta.fillna({"description": ""}, inplace=True)

pd.Series(meta['description'].astype(str).str.split().str.len()).describe()

## Обработка данных

### Sentiment analysis

sia = SentimentIntensityAnalyzer()

sentiment_scores = [sia.polarity_scores(text) for text in
reviews['reviewText'].values]
reviews = pd.concat([reviews, pd.DataFrame(sentiment_scores)], axis=1)

reviews["subj"] = reviews['reviewText'].apply(
    lambda x: TextBlob(x).sentiment.subjectivity
)

reviews

### Text Vectorization And Cosine Similarity

stemmer = PorterStemmer()

def preprocess(text):
    tokens = gensim.utils.simple_preprocess(str(text))
    tokens = [t for t in tokens if t not in stop_words]
    tokens = [stemmer.stem(t) for t in tokens]
    return tokens

tokenized_desc = [preprocess(text) for text in meta['description'].values]

tokenized_reviews = [preprocess(text) for text in reviews['reviewText'].values]

```

```

w2v_corpus = tokenized_desc + tokenized_reviews

w2v_model = Word2Vec(
    sentences=w2v_corpus,
    vector_size=100,
    window=5,
    min_count=10,
    workers=4,
    sg=1
)

def document_vector(doc):
    vecs = [w2v_model.wv[word] for word in doc if word in w2v_model.wv]
    return np.mean(vecs, axis=0) if vecs else np.zeros(w2v_model.vector_size)

desc_clean = [' '.join(tokens) for tokens in tokenized_desc]
reviews_clean = [' '.join(tokens) for tokens in tokenized_reviews]

vectorizer = TfidfVectorizer(
    ngram_range=(1, 1),
    min_df=10
)

model = SentenceTransformer('all-MiniLM-L6-v2')

embeddings = {
    'descriptions': {},
    'reviews': {}
}
emb_times = {}

t0 = time.time()
embeddings['descriptions']['w2v'] = np.array([document_vector(doc) for doc in
tokenized_desc])
embeddings['reviews']['w2v'] = np.array([document_vector(doc) for doc in
tokenized_reviews])
emb_times['w2v'] = time.time() - t0

t0 = time.time()
embeddings['descriptions']['tfidf'] =
vectorizer.fit_transform(desc_clean).toarray()
embeddings['reviews']['tfidf'] = vectorizer.transform(reviews_clean).toarray()
emb_times['tfidf'] = time.time() - t0

t0 = time.time()
embeddings['descriptions']['bert'] = model.encode(
    meta['description'].tolist(),
    batch_size=256,
    show_progress_bar=True,
    convert_to_numpy=True
)
embeddings['reviews']['bert'] = model.encode(
    reviews['reviewText'].tolist(),
    batch_size=256,
    show_progress_bar=True,
    convert_to_numpy=True
)
emb_times['bert'] = time.time() - t0

def cos_sim(a, b):
    return np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b) + 1e-8)

```

```

for vec_method in ['tfidf', 'w2v', 'bert']:

    reviews[f'sim_{vec_method}'] = np.array([
        cos_sim(
            embeddings['reviews'][vec_method][idx],
            embeddings['descriptions'][vec_method][meta.index[meta['asin']] ==
reviews['asin'].iloc[idx]][0])
        ) for idx in range(len(reviews))
    ])

emb_times

for method in embeddings['descriptions']:
    desc_shape = embeddings['descriptions'][method].shape
    rev_shape = embeddings['reviews'][method].shape
    print(f"\nМетод: {method}")
    print(f"  descriptions → {desc_shape}")
    print(f"  reviews      → {rev_shape}")

reviews

### Matrix Factorization

reader = Reader(rating_scale=(1, 5))

train_idx, test_idx = train_test_split(
    reviews[['reviewerID', 'asin', 'overall']].index,
    test_size=0.2,
    random_state=42,
    stratify = reviews['overall'].values
)

trainset = Dataset.load_from_df(
    reviews[['reviewerID', 'asin', 'overall']].iloc[train_idx],
    reader).build_full_trainset()

testset = list(reviews[['reviewerID', 'asin',
'overall']].iloc[test_idx].itertuples(index=False, name=None))

mf_models = {
    "SVDpp": SVDpp(n_factors=20, n_epochs=20, random_state=42),
    "CoClustering": CoClustering(n_cltr_u=4, n_cltr_i=4, n_epochs=30),
    "SlopeOne": SlopeOne(),
    "KNNBaseline": KNNBaseline(
        k=40,
        sim_options={"name": "cosine", "user_based": False}
    )
}

for name, algo in mf_models.items():
    algo.fit(trainset)

    train_testset = [
        (row.reviewerID, row.asin, row.overall)
        for row in reviews[['reviewerID', 'asin',
'overall']].iloc[train_idx].itertuples(index=False)
    ]

    predict_test = algo.test(testset)

```

```

predict_train = algo.test(train_testset)

reviews[f'pred_{name}'] = 0.0

reviews.loc[train_idx, f'pred_{name}'] = [p.est for p in predict_train]
reviews.loc[test_idx, f'pred_{name}'] = [p.est for p in predict_test]

reviews.columns

## Рекомендаційні системи

### Стекінг

all_features = ['neg', 'neu',
                'pos', 'compound', 'subj', 'sim_tfidf', 'sim_w2v', 'sim_bert',
                'pred_SVDpp', 'pred_CoClustering', 'pred_SlopeOne', 'pred_KNNBaseline']

reviews[all_features].describe().T

reviews[['overall']+all_features].corr()['overall'].drop('overall')

reviews

feature_cols = ['neg', 'neu',
                'pos', 'compound', 'subj', 'sim_tfidf', 'sim_w2v', 'sim_bert',
                'pred_SVDpp', 'pred_CoClustering', 'pred_KNNBaseline']

X_train = reviews[feature_cols].iloc[train_idx].values
y_train = reviews["overall"].iloc[train_idx].values

X_test = reviews[feature_cols].iloc[test_idx].values
y_test = reviews["overall"].iloc[test_idx].values

### Мета-моделі прогнозування

meta_models = {
    "STACK_RF": RandomForestRegressor(
        n_estimators=300,
        min_samples_split=5,
        min_samples_leaf=2,
        random_state=42,
        n_jobs=-1,
    ),
    "STACK_XGB": XGBRegressor(
        n_estimators=600,
        learning_rate=0.05,
        max_depth=8,
        subsample=0.9,
        colsample_bytree=0.8,
        reg_lambda=2,
        reg_alpha=1,
        random_state=42,
        n_jobs=-1,
    ),
    "STACK_LR": LinearRegression(),
}

### Метрики якості

stacking_results = {}

```

```

for name, model in meta_models.items():
    model.fit(X_train, y_train)

    t0 = time.time()
    y_pred = model.predict(X_test)
    pred_time = time.time() - t0

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)

    stacking_results[name] = {
        "MSE": mse,
        "RMSE": rmse,
        "MAE": mae,
        "time": pred_time
    }

def eval_baseline(name: str):
    t0 = time.time()
    y_pred = reviews[f"pred_{name}"].iloc[test_idx].values
    pred_time = time.time() - t0

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)

    return {"MSE": mse, "RMSE": rmse, "MAE": mae, "time": pred_time}

baseline_results = {
    name: eval_baseline(name) for name in mf_models.keys()
}

all_results = {**baseline_results, **stacking_results}
results_df = pd.DataFrame(all_results).T
print(results_df)

importances = {}

for name, model in meta_models.items():
    if hasattr(model, "feature_importances_"):
        importances[name] = model.feature_importances_
    elif hasattr(model, "coef_"):
        importances[name] = np.abs(np.ravel(model.coef_)) # для LR беремо
модуль
    else:
        continue

imp_df = pd.DataFrame(importances, index=feature_cols)
print(imp_df)

```