

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут прикладного системного аналізу  
Кафедра математичних методів системного аналізу**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Оксана ТИМОЩУК

«\_\_» \_\_\_\_\_ 2025 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системний аналіз і управління»**

**спеціальності 124 «Системний аналіз»**

**на тему: «Система аналізу відгуків про товари з використанням методів  
обробки природної мови»**

Виконав:

студент ІV курсу, групи КА-14  
Арестенко Георгій Сергійович

\_\_\_\_\_

Керівник:

к.т.н., доцент. Куєвда Ю.В.

\_\_\_\_\_

Консультант з економічного розділу:

доцент. Роцина Н.В.

\_\_\_\_\_

Консультант з нормоконтролю:

Канцедал Г.О.

\_\_\_\_\_

Рецензент:

Доктор філософії Петро Зінькевич

\_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
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 – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Оксана ТИМОЩУК

«\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

**Арестенко Георгію Сергійовичу**

1. Тема роботи «Система аналізу відгуків про товари з використанням методів обробки природної мови», керівник роботи доцент Куєвда Ю.В., затверджені наказом по університету від «25»05 2024 р. №1148-с
2. Термін подання студентом роботи \_\_\_\_\_
3. Вихідні дані до роботи
4. Зміст роботи
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)
6. Консультанти розділів роботи\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доцент Рощина Н.В.		

7. Дата видачі завдання \_\_\_\_\_

## Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Вивчення літератури за темою обробки природної мови та сентимент-аналізу	18.04.2025 – 23.04.2025	Виконано
2	Аналіз актуальності задачі дипломної роботи після теоретичного ознайомлення	22.04.2025 -30.04.2025	Виконано
3	Дослідження методів обробки української мови в системах NLP	01.05.2025 – 05.05.2025	Виконано
4	Розробка архітектури програмного комплексу	06.05.2025 – 13.05.2025	Виконано
5	Створення алгоритмів атрибутного та сентимент-аналізу	14.05.2025 – 21.05.2025	Виконано
6	Тестування системи на реальних даних та оптимізація алгоритмів	22.05.2025 – 23.06.2025	Виконано
7	Оформлення записки та розділів в дипломній роботі відповідно до нормоконтролю	24.05.2025 – 25.05.2025	Виконано
8	Попередній захист дипломної роботи	01.06.2025 – 15.06.2025	Виконано
9	Захист дипломної роботи	16.06.2025 – 30.06.2025	Виконано

Студент

Георгій АРЕСТЕНКО

Керівник

Юлія КУЄВДА

## РЕФЕРАТ

Дипломна робота: 94 с., 16 рис., 12 табл., 2 додатки, 24 джерело.

ОБРОБКА ПРИРОДНОЇ МОВИ, СЕНТИМЕНТ-АНАЛІЗ, АТРИБУТНИЙ АНАЛІЗ, УКРАЇНСЬКА МОВА, АНАЛІЗ ВІДГУКІВ, АВТОМОБІЛЬНІ ШИНИ, ПРОГРАМНИЙ КОМПЛЕКС.

Об'єктом дослідження є процеси автоматичного аналізу емоційного забарвлення текстових відгуків користувачів про товари для прийняття бізнес-рішень.

Предметом дослідження є методи та алгоритми обробки природної мови для сентимент-аналізу та виявлення атрибутів товарів у текстових відгуках.

Мета роботи полягає у розробці програмної системи для автоматичного аналізу відгуків про товари з використанням методів обробки природної мови для виявлення тональності та ключових атрибутів товарів.

Методи розробки базуються на теорії обробки природної мови, алгоритмах сентимент-аналізу з лексиконним та контекстуальним підходами, методах статистичної обробки тексту. Програмний комплекс створено з використанням Python та спеціалізованих бібліотек.

Особливістю дослідження є розробка системи аналізу відгуків, адаптованої для української мови з контекстуальним аналізом сентименту та обробкою заперечень і модифікаторів інтенсивності.

Практичне значення роботи полягає у створенні програмного комплексу для автоматизації аналізу відгуків клієнтів, що дозволяє виявляти проблемні аспекти товарів та формувати аналітичні звіти для управлінських рішень.

## ABSTRACT

Thesis: 94 p., 16 figures, 12 tables, 2 appendices, 24 sources.

NATURAL LANGUAGE PROCESSING, SENTIMENT ANALYSIS, ATTRIBUTE ANALYSIS, UKRAINIAN LANGUAGE, REVIEW ANALYSIS, AUTOMOTIVE TIRES, SOFTWARE SYSTEM.

The object of research is the processes of automatic analysis of emotional coloring of textual user reviews about products for business decision-making.

The subject of research is methods and algorithms of natural language processing for sentiment analysis and detection of product attributes in textual reviews.

The purpose of the work is to develop a software system for automatic analysis of product reviews using natural language processing methods to detect sentiment and key product attributes.

Development methods are based on natural language processing theory, sentiment analysis algorithms with lexicon-based and contextual approaches, statistical text processing methods. The software system was created using Python and specialized libraries.

The distinctive feature of the research lies in developing a review analysis system adapted for the Ukrainian language with contextual sentiment analysis and processing of negations and intensity modifiers.

The practical significance of the work lies in creating a software system for automating customer review analysis, which allows identifying problematic product aspects and generating analytical reports for management decisions.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	12
1.1 Основи обробки природної мови для аналізу текстових відгуків .....	12
1.2 Методи та алгоритми сентимент-аналізу .....	14
1.3 Алгоритми виявлення тематичних атрибутів у текстових відгуках.....	17
1.4 Особливості обробки української мови в системах NLP.....	20
1.5 Сучасні технології та інструменти для створення систем аналізу відгуків .....	22
1.6 Висновок до 1 розділу .....	27
<b>РОЗДІЛ 2 РОЗРОБКА КОНЦЕПЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ</b> .	29
2.1 Визначення вимог до програмного комплексу .....	29
2.2 Архітектура системи та вибір технологій .....	32
2.3 Алгоритми та методи аналізу тексту .....	34
2.3.1 Попередня обробка тексту .....	35
2.3.2 Сентимент-аналіз з урахуванням контексту .....	36
2.3.3 Атрибутний аналіз шин .....	36
2.3.4 Оптимізація продуктивності.....	37
2.4 Висновки до 2 розділу .....	38
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ</b> .....	40
3.1 Опис структури та компонентів системи .....	40
3.2 Інтерфейс користувача .....	42
3.3 Алгоритми обробки та аналізу даних .....	47
3.3.1 Алгоритм завантаження та нормалізації даних .....	47
3.3.2 Алгоритм сентимент-аналізу .....	47
3.3.3. Алгоритм атрибутного аналізу .....	50
3.3.4 Алгоритм генерації візуалізацій .....	51
3.4. Система експорту та звітності .....	53
3.4.1 Структура експортованих звітів .....	53
3.4.2 Алгоритм формування звітів.....	55
3.5 Висновки до 3 розділу .....	55

<b>РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>57</b>
4.1 Постановка задачі проектування .....	58
4.2 Обґрунтування функцій програмного продукту .....	58
4.3 Обґрунтування системи параметрів програмного продукту .....	60
4.4 Аналіз експертного оцінювання параметрів .....	64
4.5 Аналіз рівня якості варіантів реалізації функцій .....	67
4.5 Економічний аналіз варіантів розробки ПП .....	68
4.6 Висновки до 4 розділу .....	72
<b>ВИСНОВКИ .....</b>	<b>74</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....</b>	<b>77</b>
<b>ДОДАТОК А .....</b>	<b>80</b>
<b>ДОДАТОК Б ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....</b>	<b>86</b>

## ВСТУП

Сучасна онлайн комерція та сфера послуг характеризуються інтенсивним зростанням обсягів користувацьких відгуків, які стали ключовим фактором формування репутації товарів та впливу на споживчі рішення. Традиційні методи ручної обробки та аналізу відгуків виявляються неефективними при роботі з великими масивами текстових даних, що створює потребу в автоматизованих рішеннях. Водночас, специфіка української мови з її морфологічним багатством та синтаксичними особливостями ускладнює застосування існуючих інструментів, більшість з яких оптимізована для англomовного контенту [1, 2]. Використання методів обробки природної мови та сентимент-аналізу відкриває можливості для створення інтелектуальних систем, здатних автоматично аналізувати емоційне забарвлення відгуків та виявляти ключові атрибути товарів, що робить цю тематику особливо актуальною для підприємств [3, 4], які прагнуть оптимізувати процеси аналізу зворотного зв'язку від клієнтів.

Мета дослідження – розробка програмної системи для автоматичного аналізу відгуків про товари з використанням методів обробки природної мови, що забезпечить ефективне виявлення тональності відгуків та автоматичне розпізнавання ключових атрибутів товарів у текстових коментарях користувачів.

Завдання дослідження:

- 1) провести аналіз сучасних методів обробки природної мови та сентимент-аналізу для роботи з українськомовними текстами;
- 2) дослідити алгоритми виявлення тематичних атрибутів у текстових відгуках та методи їх класифікації;
- 3) підготувати та структурувати корпус текстових відгуків для навчання та тестування системи;
- 4) розробити програмну систему з графічним інтерфейсом, що реалізує алгоритми сентимент-аналізу та виявлення атрибутів товарів;

- 5) провести експериментальну перевірку точності та ефективності розробленої системи на реальних даних;
- б) створити систему візуалізації результатів аналізу та механізми експорту звітів для практичного використання.

Об'єкт дослідження – процеси автоматичного аналізу емоційного забарвлення та змістовного наповнення текстових відгуків користувачів про товари з метою отримання структурованої інформації для прийняття бізнес-рішень.

Предмет дослідження – методи та алгоритми обробки природної мови для сентимент-аналізу та виявлення атрибутів товарів у текстових відгуках, а також програмна система для їх практичної реалізації.

Методи дослідження:

- 1) аналіз наукових публікацій та огляд існуючих підходів до обробки природної мови та аналізу тональності тексту;
- 2) методи лінгвістичного аналізу для створення словників емоційно забарвлених слів та атрибутів товарів;
- 3) алгоритм сентимент-аналізу з використанням лексиконного підходу та контекстуального аналізу;
- 4) методи статистичної обробки тексту для токенизації та видалення стоп-слів;
- 5) техніки візуалізації даних для представлення результатів аналізу у вигляді графіків та інтерактивних елементів.

Особливість дослідження полягає у розробці спеціалізованої системи аналізу відгуків, адаптованої для української мови з урахуванням її морфологічних особливостей. На відміну від універсальних рішень, запропонована система забезпечує контекстуальний аналіз сентименту з обробкою заперечень та модифікаторів інтенсивності, а також автоматичне виявлення широкого спектру атрибутів товарів з можливістю оцінки їх сентименту окремо для кожної характеристики. Оригінальність підтверджується відсутністю в україномовному сегменті інтернету аналогічних систем, що поєднують контекстуальний сентимент-аналіз з атрибутним аналізом для автомобільної тематики.

Існуючі інструменти типу VADER чи TextBlob не адаптовані до морфологічних особливостей української мови, а спеціалізовані рішення для автомобільної індустрії зосереджені переважно на англomовному контенті.

Практична значимість дипломної роботи: Розроблена програмна система може бути впроваджена у діяльність підприємств електронної комерції, виробничих компаній та сервісних організацій для автоматизації процесів аналізу відгуків клієнтів. Система дозволяє швидко виявляти проблемні аспекти товарів, відстежувати динаміку задоволеності споживачів та формувати аналітичні звіти для прийняття обґрунтованих управлінських рішень щодо покращення якості продукції та сервісу.

Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

Вступ – містить обґрунтування актуальності дослідження, формулювання мети та завдань, визначення об'єкта та предмета дослідження.

Дослідження предметної області – розглядає основи обробки природної мови для аналізу текстових відгуків, методи та алгоритми сентимент-аналізу, алгоритми виявлення тематичних атрибутів у текстових відгуках, особливості обробки української мови в системах NLP та сучасні технології й інструменти для створення систем аналізу відгуків.

Розробка концепції програмного комплексу – містить визначення вимог до програмного комплексу, опис архітектури системи та вибору технологій, алгоритми та методи аналізу тексту з детальним розглядом попередньої обробки тексту, сентимент-аналізу з урахуванням контексту та атрибутного аналізу шин.

Реалізація програмного комплексу – описує структуру та компоненти системи, інтерфейс користувача, алгоритми обробки та аналізу даних (включаючи алгоритми завантаження та нормалізації даних, сентимент-аналізу, атрибутного аналізу та генерації візуалізацій), а також систему експорту та звітності.

Функціонально-вартісний аналіз програмного продукту – містить постановку задачі проектування, обґрунтування функцій та параметрів програмного продукту, аналіз експертного оцінювання параметрів, аналіз рівня якості варіантів реалізації функцій та економічний аналіз варіантів розробки програмного продукту.

Додатки – містять основні компоненти програмного коду системи аналізу відгуків, включаючи ініціалізацію класу, словники для аналізу атрибутів шин, алгоритми сентимент-аналізу та виявлення атрибутів, створення графічного інтерфейсу, генерацію візуалізацій, систему фільтрації даних та експорт результатів і повний зміст презентації.

## РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Основи обробки природної мови для аналізу текстових відгуків

Обробка природної мови (Natural Language Processing, NLP [5]) становить міждисциплінарну область, що об'єднує методи комп'ютерної лінгвістики, машинного навчання та штучного інтелекту для забезпечення ефективної взаємодії комп'ютерних систем з людською мовою. У контексті аналізу відгуків користувачів NLP виконує критично важливу функцію перетворення неструктурованих текстових даних у структуровану інформацію, придатну для подальшої автоматичної обробки та аналізу.

Сучасні системи аналізу відгуків стикаються з численними викликами, пов'язаними з неоднорідністю природної мови, включаючи варіативність синтаксичних конструкцій, полісемію, контекстуальні залежності та морфологічну складність. Особливо гостро ці проблеми проявляються при роботі з українською мовою, яка характеризується багатою флективною системою та складною морфологією. Статистичні дослідження показують, що українська мова має один із найвищих рівнів морфологічної складності [6] серед європейських мов, що створює додаткові виклики для автоматичної обробки текстів [7].

Сучасні системи аналізу відгуків стикаються з численними викликами, пов'язаними з неоднорідністю природної мови, включаючи варіативність синтаксичних конструкцій, полісемію, контекстуальні залежності та морфологічну складність. Для української мови ці проблеми набувають особливої гостроти через багату флективну систему з шістьма відмінками та складну морфологію. Статистичні дослідження показують, що українська мова має один із найвищих рівнів морфологічної складності серед європейських мов [6], що створює додаткові виклики для автоматичної обробки текстів [7].

Існують різні підходи до токенізації: від простого розбиття по пробілах до складних алгоритмів, що враховують морфологічну структуру слів. Простий підхід базується на використанні регулярних виразів для ідентифікації меж слів за допомогою пробілів та знаків пунктуації. Однак такий метод має обмеження при обробці складних випадків, таких як скорочення, аббревіатури, числові вирази та дефісні написання.

Сучасні методи токенізації включають використання статистичних моделей та нейромережових підходів, що дозволяють коректно обробляти скорочення, аббревіатури та складні словоформи. Зокрема, алгоритм Byte Pair Encoding (BPE) [8] демонструє високу ефективність при роботі з морфологічно складними мовами, автоматично виявляючи найчастіші підпоследовності символів та використовуючи їх як базові одиниці токенізації.

Морфологічний аналіз передбачає визначення граматичних характеристик слів, включаючи частину мови, відмінок, число, рід, час та інші морфологічні категорії. Для флективних мов, до яких належить українська, цей етап є особливо важливим, оскільки одне і те ж слово може мати десятки різних форм [7]. Наприклад, українське слово "книга" може мати такі форми: книга, книги, книзі, книгу, книгою, книг, книгам, книгами, книгах та інші, залежно від граматичного контексту.

Лематизація являє собою процес приведення слів до їх канонічної форми – леми. На відміну від стемінгу, який використовує евристичні правила для видалення закінчень, лематизація базується на морфологічному аналізі та словникових ресурсах, що забезпечує більш точні результати для морфологічно складних мов. Дослідження ефективності різних підходів до лематизації показують, що для української мови найкращі результати демонструють гібридні методи, що поєднують словникові ресурси з машинним навчанням.

$$H = -\sum p(w) \times \log_2 p(w) [9] \quad (1.1)$$

де  $H$  – ентропія слів у тексті (у бітах на слово);

$p(w)$  – ймовірність появи слова  $w$  у корпусі тексту;

$w$  – унікальне слово в словнику тексту.

Стоп-слова – це високочастотні слова, що не несуть значного смислового навантаження для конкретного завдання аналізу тексту. До них зазвичай відносять прийменники, сполучники, частки, займенники та інші службові частини мови. Ефективна фільтрація стоп-слів дозволяє зменшити розмірність даних та покращити якість подальшого аналізу.

Для української мови створення списку стоп-слів потребує врахування специфічних граматичних особливостей, включаючи варіанти написання службових слів, діалектні особливості та жаргонізми, характерні для інтернет-комунікації. Крім традиційних стоп-слів, в інтернет-текстах часто зустрічаються специфічні елементи: емотикони, абрєвіатури, англїцизми та сленгові вирази, що потребують окремої обробки.

Дослідження показують, що оптимальний розмір списку стоп-слів для української мови становить 150-200 одиниць [10] для загальних завдань та може розширюватися до 400-500 слів для специфічних предметних областей. Важливо балансувати між повнотою фільтрації шумових слів та збереженням семантично значущої інформації.

## 1.2 Методи та алгоритми сентимент-аналїзу

Сентимент-аналїз, або аналіз тональності тексту, є одним із ключових напрямків обробки природної мови, що спрямований на автоматичне визначення емоційного забарвлення текстових повідомлень. У контексті аналізу відгуків про товари цей метод дозволяє класифікувати думки користувачів як позитивні, негативні або нейтральні, а також визначати ступінь інтенсивності емоцій.

Історично сентимент-аналїз розвивався від простих підходів на основі підрахунку ключових слів до складних нейромережєвих архїтектур, здатних

враховувати контекстуальні залежності та семантичні нюанси природної мови. Сучасні системи демонструють точність класифікації від 75% до 95% [1, 3] в залежності від предметної області та якості даних для навчання.

Лексиконні методи базуються на використанні попередньо створених словників емоційно забарвлених слів, де кожному терміну присвоєна оцінка полярності та інтенсивності. Цей підхід має кілька переваг: інтерпретованість результатів, можливість тонкого налаштування та незалежність від великих обсягів розмічених даних для навчання.

Найвідоміші лексиконні ресурси включають VADER (Valence Aware Dictionary and sEntiment Reasoner), SentiWordNet та AFINN [1]. Для української мови існують обмежені лексиконні ресурси, такі як адаптовані версії міжнародних словників та розроблені вітчизняними дослідниками спеціалізовані лексикони та адаптовані версії міжнародних словників.

Основною проблемою лексиконного підходу є контекстуальна залежність значення слів. Одне і те ж слово може мати різне емоційне забарвлення в залежності від контексту вживання. Наприклад, слово "гострий" може мати позитивне забарвлення у контексті "гострий розум" та негативне у "гострий біль". Тому сучасні лексиконні методи включають механізми контекстуального аналізу та обробки модифікаторів значення.

Заперечення кардинально змінюють емоційне забарвлення висловлювання, інвертуючи полярність оцінки. Автоматичне виявлення та обробка заперечних конструкцій є одним із найскладніших завдань сентимент-аналізу [11, 12], особливо для морфологічно багатих мов. В українській мові заперечення може виражатися різними способами: через частку "не", префікси (недобрий, безглуздий), займенники (ніщо, ніхто) та цілі заперечні конструкції.

$$S\_corrected = S\_base \times (-1)^N \times (1 + I\_factor \times D\_factor) \quad [11, 12] \quad (1.2)$$

де  $S\_base$  – базовий сентимент слова;

$N$  – кількість заперечень у вікні контексту;

$I\_factor$  – коефіцієнт підсилення (1.2-1.8);

$D\_factor$  – коефіцієнт послаблення (0.3-0.7).

Посилювачі та послаблювачі є модифікаторами, що підсилюють або послаблюють емоційне забарвлення висловлювання. До підсилювачів належать слова типу "дуже", "надзвичайно", "неймовірно", тоді як послаблювачі включають терміни "трохи", "дещо", "відносно". Коректна обробка цих модифікаторів значно підвищує точність сентимент-аналізу.

Дослідження показують, що врахування модифікаторів інтенсивності може підвищити точність сентимент-аналізу на 8-15% [12] порівняно з базовими лексиконними методами. Особливо важливим є визначення області дії модифікаторів – зазвичай вона обмежується межами речення або синтагми.

Контекстуальний підхід до сентимент-аналізу враховує не лише окремі слова, а й їх взаємодію в межах речення або тексту. Цей метод дозволяє більш точно інтерпретувати складні лінгвістичні конструкції, включаючи іронію, сарказм та імпліцитні оцінки.

Сучасні алгоритми контекстуального аналізу використовують техніки ковзного вікна, n-грами та семантичні ембединги для визначення семантичних зв'язків між словами та їх спільного впливу на загальну тональність тексту. Розмір контекстного вікна зазвичай становить 3-7 слів навколо цільового терміну, що забезпечує оптимальний баланс між повнотою контексту та обчислювальною ефективністю.

Одним із найскладніших аспектів контекстуального аналізу є обробка іронії та сарказму. Ці явища часто характеризуються протиріччям між буквальним значенням слів та їх дійсним сенсом. Автоматичне виявлення іронії потребує аналізу дискурсивних маркерів, патерни пунктуації та семантичних невідповідностей.

Найбільш ефективні сучасні системи сентимент-аналізу поєднують переваги різних підходів. Гібридні методи можуть комбінувати лексиконні техніки з алгоритмами машинного навчання, що дозволяє досягти високої точності при збереженні інтерпретованості результатів.

Типовий гібридний підхід включає попередню обробку тексту лексиконними методами з подальшим застосуванням алгоритмів класифікації для

корекції та уточнення результатів на основі контекстуальних ознак. Такі системи часто використовують ансамблі різних класифікаторів, що дозволяє компенсувати слабкості окремих методів.

Розвиток трансформерних архітектур, таких як BERT та його варіанти [13, 14], відкрив нові можливості для контекстуального сентимент-аналізу. Ці моделі демонструють високу ефективність на стандартних бенчмарках, однак потребують значних обчислювальних ресурсів та великих обсягів даних для навчання.

### 1.3 Алгоритми виявлення тематичних атрибутів у текстових відгуках

Виявлення атрибутів товарів у відгуках користувачів є комплексним завданням, що потребує поєднання методів інформаційного пошуку, машинного навчання та лінгвістичного аналізу. Основна мета полягає в автоматичній ідентифікації згадувань конкретних характеристик товару та визначенні пов'язаної з ними оцінки користувачів.

Ефективність виявлення атрибутів критично важлива для бізнес-аналітики, оскільки дозволяє компаніям отримувати детальну інформацію про сприйняття різних аспектів їхніх продуктів. Статистичні дані показують, що автоматичне виявлення атрибутів може підвищити ефективність аналізу відгуків у 5-10 разів [4, 15] порівняно з ручною обробкою.

#### *Словникові методи виявлення атрибутів*

Словникові підходи базуються на використанні попередньо визначених списків термінів, що асоціюються з конкретними атрибутами товарів. Для кожної категорії атрибутів створюється розширений словник, що включає синоніми, варіанти написання та морфологічні форми ключових термінів.

Основною перевагою словникового методу є контрольованість та інтерпретованість результатів. Експерти предметної області можуть безпосередньо

впливати на процес виявлення атрибутів, додаючи нові терміни або корегуючи існуючі. Однак цей підхід має обмеження щодо покриття всього різноманіття способів вираження атрибутів у природній мові.

Створення якісних словників атрибутів потребує глибокого розуміння предметної області та аналізу великих обсягів текстових даних. Для товарів типу автомобільних шин словник може включати сотні термінів, згрупованих за категоріями: зчеплення, зносостійкість, шум, комфорт, ціна тощо. Кожна категорія може містити десятки варіантів написання та синонімів.

#### *Методи тематичного моделювання*

Тематичне моделювання дозволяє автоматично виявляти приховані тематичні структури в колекціях текстів. Алгоритми типу Latent Dirichlet Allocation (LDA) [16] або Non-negative Matrix Factorization можуть ідентифікувати групи слів, що часто зустрічаються разом та відповідають конкретним аспектам товарів.

Перевага тематичного моделювання полягає в можливості виявлення нових, раніше невідомих атрибутів без попереднього визначення словників. Однак інтерпретація результатів може бути складною, а якість виявлених тем залежить від параметрів алгоритму та якості вхідних даних.

$$P(w|d) = \sum_z P(w|z) \times P(z|d) \quad [16] \quad (1.3)$$

де  $P(w|d)$  – ймовірність слова  $w$  у документі  $d$  за темою  $z$ ;

$P(w|z)$  – ймовірність слова  $w$  у темі  $z$ ;

$P(z|d)$  – ймовірність теми  $z$  у документі  $d$ .

Практичне застосування LDA для аналізу відгуків показує, що оптимальна кількість тем зазвичай становить 8-15 [16, 17] для конкретної категорії товарів. Більша кількість тем може призвести до надмірної фрагментації, тоді як менша – до втрати специфічних аспектів.

Аспектно-орієнтований сентимент-аналіз (ABSA) [18] представляє собою розширення традиційного сентимент-аналізу, що дозволяє визначати тональність для кожного виявленого аспекту товару окремо. Цей підхід

особливо цінний для аналізу детальних відгуків, де користувачі можуть позитивно оцінювати одні характеристики товару та негативно – інші.

ABSA включає три основні етапи: виявлення аспектів, визначення їх полярності та агрегацію результатів. Сучасні системи ABSA використовують нейромережеві архітектури, що дозволяють спільно вирішувати завдання виявлення аспектів та аналізу їх тональності.

Дослідження ефективності ABSA показують, що цей підхід може підвищити точність аналізу відгуків на 15-25% [18, 15] порівняно з загальним сентимент-аналізом. Особливо значні покращення спостерігаються при аналізі довгих, детальних відгуків, де користувачі обговорюють різні аспекти товару.

Контекстуальні підходи враховують не лише присутність ключових слів, а й їх оточення в тексті. Це дозволяє більш точно визначити, чи дійсно згадується конкретний атрибут та яка оцінка йому надається.

Методи контекстуального аналізу включають використання ковзного вікна навколо ключових термінів, аналіз синтаксичних зв'язків та семантичних ролей слів у реченні. Такий підхід дозволяє відфільтрувати помилкові спрацювання та підвищити точність виявлення атрибутів.

Особливо важливим є аналіз оцінних конструкцій навколо згадувань атрибутів. Дослідження показують, що 80-90% оціночних висловлювань про атрибути знаходяться в межах 5-7 слів від згадування атрибуту, що дозволяє ефективно локалізувати релевантну інформацію.

## 1.4 Особливості обробки української мови в системах NLP

Українська мова належить до східнослов'янської групи індоєвропейської мовної сім'ї [7, 19] та характеризується низкою специфічних особливостей, що створюють додаткові виклики для систем обробки природної мови. Розуміння цих особливостей є критично важливим для розробки ефективних NLP-систем для українськомовних текстів.

Українська мова має багату флективну систему з шістьма відмінками для іменників та прикметників, складною системою дієвідмінювання та численними морфофонологічними змінами. Це призводить до того, що одна лексема може мати десятки різних словоформ, що значно ускладнює процеси токенізації та лематизації.

Українська мова має одну з найбагатших флективних систем серед європейських мов, що створює проблему розрідженості даних при статистичному аналізі текстів [7]. Така морфологічна багатоманітність створює проблему розрідженості даних при статистичному аналізі текстів.

Крім того, українська мова характеризується активним словотворенням через префіксацію, суфіксацію та складання, що створює додаткові труднощі для автоматичного аналізу морфологічної структури слів. Продуктивні словотвірні моделі дозволяють створювати нові слова "на льоту", що ускладнює створення вичерпних словників.

Українська мова має відносно вільний порядок слів, що дозволяє різні синтаксичні конструкції для вираження одного і того ж змісту. Це створює виклики для алгоритмів, що базуються на фіксованих синтаксичних патернах. Наприклад, речення "Мені подобається ця книга", "Ця книга мені подобається" та "Подобається мені ця книга" мають однаковий смисл, але різну синтаксичну структуру.

Особливу увагу потребує обробка заперечних конструкцій, які в українській мові можуть виражатися різними способами та впливати на різні

компоненти речення. Коректна ідентифікація області дії заперечення є критично важливою для точного сентимент-аналізу.

Таблиця 1.1 – Типи заперечних конструкцій в українській мові

Тип заперечення	Приклад	Область дії	Складність обробки
Часткове "не"	не добрий	Прикметник	Низька
Повне "не"	не подобається	Дієслово	Середня
Заперечні займенники	ніхто не знає	Речення	Висока
Множинне заперечення	ніколи нічого не	Весь кон- текст	Дуже висока

Українська мова також характеризується складною системою узгодження слів у реченні, що потребує врахування граматичних категорій при синтаксичному аналізі. Неправильне визначення синтаксичних зв'язків може призвести до помилок у визначенні об'єктів оцінки та їх характеристик.

Для української мови існує обмежена кількість високоякісних лексичних ресурсів, необхідних для ефективної роботи NLP-систем. Це включає словники емоційно забарвлених слів, тезауруси, корпуси розмічених текстів та морфологічні словники.

Створення власних лексичних ресурсів потребує значних зусиль експертів-лінгвістів та може бути одним із основних викликів при розробці систем аналізу українськомовних текстів. Наявні ресурси часто мають обмежене покриття або недостатню якість розмітки, що впливає на ефективність NLP-систем.

Особливою проблемою є відсутність великих корпусів розмічених текстів для навчання моделей машинного навчання. Більшість існуючих корпусів мають обсяг менше 1 мільйона токенів [10], що недостатньо для навчання сучасних нейромережевих моделей.

Українська мова має численні діалектні варіанти, що можуть впливати на лексичний склад та граматичні конструкції в текстах відгуків. Крім того, в інтернет-комунікації часто зустрічаються суржикові елементи та запозичення з інших мов, що додатково ускладнює автоматичну обробку.

Ефективні системи аналізу українськомовних відгуків повинні враховувати цю варіативність та включати механізми нормалізації текстів до стандартної форми. Дослідження показують, що врахування регіональних особливостей може підвищити точність аналізу на 5-8% [20] для текстів із соціальних мереж та форумів.

### 1.5 Сучасні технології та інструменти для створення систем аналізу відгуків

Розробка ефективної системи аналізу відгуків потребує інтеграції різноманітних технологій та інструментів, вибір яких залежить від специфічних вимог до системи, обсягів оброблюваних даних та доступних ресурсів. Сучасний ландшафт NLP-інструментів характеризується швидким розвитком та появою нових рішень, що потребує постійного моніторингу та оцінки їх ефективності.

#### *Програмні бібліотеки для обробки природної мови*

Python є найпопулярнішою мовою програмування для NLP-завдань завдяки розвиненій екосистемі спеціалізованих бібліотек. NLTK (Natural Language Toolkit [5]) надає широкий спектр інструментів для базової обробки тексту, включаючи токенізацію, морфологічний аналіз та роботу з корпусами текстів. NLTK включає понад 50 корпусів та лексичних ресурсів, що робить її відмінним вибором для навчальних та дослідницьких проєктів.

SpaCy є більш сучасною альтернативою, що фокусується на продуктивності та легкості використання. Вона включає попередньо навчені моделі для багатьох мов та ефективні алгоритми для морфологічного аналізу та

синтаксичного парсингу. SpaCy демонструє швидкість обробки до 1 мільйона токенів за секунду [5], що робить її придатною для промислового використання.

TextBlob пропонує простий API для виконання базових NLP-операцій, включаючи сентимент-аналіз, що робить її привабливою для прототипування та навчальних проектів. Однак її можливості обмежені порівняно з більш спеціалізованими інструментами.

Для роботи з українською мовою особливої уваги заслуговує бібліотека `rutmorphy2`, що забезпечує високоякісний морфологічний аналіз. Ця бібліотека демонструє точність морфологічного аналізу понад 95% [10] для українських текстів.

#### *Бібліотеки машинного навчання та глибокого навчання*

Scikit-learn [21] є стандартною бібліотекою для машинного навчання в Python, що включає реалізації основних алгоритмів класифікації, кластеризації та регресії. Вона особливо корисна для традиційних підходів до сентимент-аналізу та класифікації текстів. Scikit-learn надає зручні інструменти для векторизації тексту, включаючи TF-IDF та bag-of-words представлення.

TensorFlow та PyTorch [14] є провідними фреймворками для глибокого навчання, що дозволяють створювати складні нейромережеві архітектури для обробки природної мови. Вони підтримують роботу з рекурентними мережами, трансформерами та іншими сучасними архітектурами. PyTorch особливо популярний у дослідницькому середовищі завдяки динамічним обчислювальним графам та зручному API.

Hugging Face Transformers [13] стала де-факто стандартом для роботи з попередньо навченими мовними моделями. Бібліотека надає доступ до тисяч готових моделей, включаючи BERT, RoBERTa, GPT та їх варіанти для різних мов. Для української мови доступні спеціалізовані моделі, такі як `uk-bert` та `ukrainian-roberta`.

### *Інструменти для роботи з даними та візуалізації*

Pandas [21] є фундаментальною бібліотекою для роботи з структурованими даними в Python. Вона надає потужні засоби для завантаження, очищення, трансформації та аналізу табличних даних, що є критично важливим для обробки відгуків користувачів. Pandas ефективно працює з великими наборами даних (до кількох мільйонів записів) та інтегрується з іншими бібліотеками екосистеми Python.

NumPy [21] забезпечує основу для числових обчислень та роботи з багатовимірними масивами. Більшість NLP-операцій потребують векторних обчислень, які ефективно реалізовані в NumPy з використанням оптимізованих бібліотек лінійної алгебри.

Matplotlib та Seaborn [21] використовуються для створення статистичних візуалізацій, що дозволяють ефективно представляти результати аналізу відгуків. Matplotlib надає низькорівневий API для створення будь-яких типів графіків, тоді як Seaborn пропонує високорівневий інтерфейс для статистичних візуалізацій.

Plotly додатково надає можливості створення інтерактивних графіків, що особливо корисно для dashboard'ів та веб-додатків. Інтерактивні елементи дозволяють користувачам досліджувати дані більш детально, змінюючи параметри візуалізації в реальному часі.

WordCloud є спеціалізованою бібліотекою для створення хмар слів – популярного способу візуалізації частотності термінів у текстах. Ця бібліотека підтримує різні шрифти, кольорові схеми та форми, що дозволяє створювати привабливі та інформативні візуалізації.

### *Технології створення користувацьких інтерфейсів*

Tkinter [5] є стандартною бібліотекою Python для створення настільних графічних інтерфейсів. Незважаючи на обмежені можливості стилізації, вона дозволяє швидко створювати функціональні прототипи з мінімальними залежностями. Tkinter включає всі основні віджети: кнопки, поля введення, списки, таблиці та canvas для графіки [22].

$$LC = (TAC \times w_1 + TBC \times w_2 + DC \times w_3 + SC \times w_4 + GT \times w_5) / 5 \times 100\% \quad (1.4)$$

де LC – Індекс складності компоновки;

TAC – Total Alignment Complexity (загальна складність вирівнювання);

TBC – Total Balance Complexity (загальна складність балансу) ;

DC – Density Complexity (складність щільності);

SC – Size Complexity (складність розмірів);

GT – Grouping Total (загальне групування);

$w_1 = 0.84$ ,  $w_2 = 0.76$ ,  $w_3 = 0.80$ ,  $w_4 = 0.72$ ,  $w_5 = 0.88$  – вагові коефіцієнти отримані при опитуванні респондентів.

Веб-технології, включаючи Flask/Django [23] для backend та React/Vue.js для frontend, надають більші можливості для створення сучасних, responsive інтерфейсів, але потребують більш складної архітектури системи. Flask є мінімалістичним фреймворком, що дозволяє швидко створювати веб-додатки, тоді як Django пропонує більш повнофункціональне рішення з ORM, адміністративним інтерфейсом та системою аутентифікації.

Streamlit набув популярності як інструмент для швидкого створення веб-додатків для data science проектів. Він дозволяє створювати інтерактивні додатки, використовуючи лише Python код, без необхідності знання HTML, CSS або JavaScript.

#### *Хмарні сервіси та платформи*

Amazon Web Services (AWS) пропонує широкий спектр сервісів для NLP, включаючи Amazon Comprehend для аналізу тональності та виявлення сутностей, Amazon Textract для розпізнавання тексту та Amazon Translate для машинного перекладу. Ці сервіси забезпечують висока доступність та можуть обробляти великі обсяги текстових даних.

Google Cloud Platform надає подібні сервіси через Google Cloud Natural Language API, що включає аналіз тональності, класифікацію тексту та виявлення сутностей. Google AutoML дозволяє створювати власні моделі машинного навчання без глибоких знань у цій області.

Microsoft Azure пропонує Cognitive Services, що включають Text Analytics API для різних NLP-завдань. Azure також надає потужні інструменти для машинного навчання через Azure Machine Learning Studio.

Git залишається стандартом для контролю версій коду, а платформи типу GitHub, GitLab та Bitbucket надають додаткові можливості для співпраці та управління проектами. Для data science проектів особливо корисними є можливості відстеження великих файлів даних та результатів експериментів.

Jupyter Notebooks стали стандартним інструментом для прототипування та дослідження в галузі data science. Вони дозволяють поєднувати код, візуалізації та пояснювальний текст в одному документі, що полегшує документування та відтворення експериментів.

Docker забезпечує контейнеризацію додатків, що спрощує розгортання та масштабування NLP-систем. Використання Docker особливо важливе для систем, що включають багато залежностей та різні версії бібліотек.

Для оцінки якості NLP-систем використовуються різні метрики. Для завдань класифікації (включаючи сентимент-аналіз) основними метриками є accuracy, precision, recall та F1-score [24]. Для задач виявлення атрибутів додатково використовуються метрики покриття (coverage) та специфічності.

$$F1 = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall}) \quad (1.5)$$

де F1-score для оцінки якості класифікації;

$$\text{precision} = TP / (TP + FP);$$

$$\text{recall} = TP / (TP + FN);$$

TP, FP, FN – true positive, false positive, false negative.

Крос-валідація є стандартним методом для оцінки узагальнювальної здатності моделей. Для NLP-задач часто використовується стратифікована крос-валідація, що забезпечує рівномірний розподіл класів у тренувальних та тестових наборах.

A/B тестування дозволяє порівнювати ефективність різних версій системи в реальних умовах експлуатації. Цей підхід особливо важливий для

промислових систем, де навіть невеликі покращення можуть мати значний бізнес-ефект.

Сучасний розвиток технологій NLP характеризується швидким темпом інновацій, особливо в галузі глибокого навчання та трансформерних архітектур. Однак для практичних застосувань важливо балансувати між використанням найновіших досягнень та надійністю, інтерпретованістю та ефективністю рішень. Вибір технологічного стеку повинен базуватися на конкретних вимогах проекту, доступних ресурсах та довгостроковій стратегії розвитку системи.

## 1.6 Висновок до 1 розділу

У першому розділі проведено системний аналіз сучасного стану досліджень у галузі обробки природної мови та сентимент-аналізу текстових відгуків, що дозволило визначити ключові напрямки та виклики розробки спеціалізованих систем для української мови.

Дослідження основ обробки природної мови для аналізу текстових відгуків виявило критичну важливість етапів токенизації, морфологічного аналізу та лематизації для якісної обробки україномовних текстів. Встановлено, що українська мова має один із найвищих рівнів морфологічної складності серед європейських мов, що створює специфічні виклики для автоматичної обробки та потребує застосування адаптованих алгоритмів.

Аналіз методів та алгоритмів сентимент-аналізу показав переваги гібридних підходів, що поєднують лексиконні методи з контекстуальним аналізом. Особливу увагу приділено проблемам обробки заперечень та модифікаторів інтенсивності, які кардинально впливають на точність визначення тональності тексту.

Дослідження алгоритмів виявлення тематичних атрибутів у текстових відгуках продемонструвало ефективність словникових методів у поєднанні з технікою тематичного моделювання.

Аналіз особливостей обробки української мови в системах NLP розкрив основні труднощі, пов'язані з багатою флективною системою, вільним порядком слів та обмеженістю лексичних ресурсів. Виявлено критичну потребу в створенні спеціалізованих словників та алгоритмів, адаптованих до морфологічних та синтаксичних особливостей української мови.

Огляд сучасних технологій та інструментів для створення систем аналізу відгуків показав доцільність використання Python-екосистеми з бібліотеками NLTK, TextBlob, pandas та matplotlib для реалізації комплексних рішень. Встановлено, що оптимальний технологічний стек повинен забезпечувати баланс між функціональністю, продуктивністю та простотою розробки.

Проведений аналіз виявив відсутність в україномовному сегменті комплексних рішень, що поєднують контекстуальний сентимент-аналіз з атрибутивним аналізом для автомобільної тематики. Це обґрунтовує актуальність створення спеціалізованої системи, здатної ефективно обробляти відгуки про автомобільні шини з урахуванням специфіки української мови та предметної області.

Результати аналізу сформувавши теоретичну основу для розробки концепції програмного комплексу, визначили ключові вимоги до архітектури системи та обґрунтували вибір методів і технологій для практичної реалізації ефективного інструменту аналізу споживчих відгуків.

## РОЗДІЛ 2 РОЗРОБКА КОНЦЕПЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ

### 2.1 Визначення вимог до програмного комплексу

Створення ефективної системи для аналізу відгуків про автомобільні шини потребує чіткого формулювання вимог до програмного комплексу. Специфіка даної предметної області полягає в необхідності обробки великих обсягів текстової інформації українською мовою, проведення глибокого сентимент-аналізу та забезпечення візуального представлення результатів для підтримки прийняття управлінських рішень.

Основна мета розробки полягає в автоматизації процесів обробки та аналізу відгуків споживачів про якість вантажних шин. Система має надавати можливість завантаження текстових даних, їх попередньої обробки, проведення детального сентимент-аналізу з врахуванням специфічних атрибутів шин, генерації хмар слів та статистичних звітів, а також фільтрації результатів за різними критеріями.

Функціональний спектр системи охоплює декілька ключових напрямків. По-перше, це робота з різними форматами вхідних даних (CSV, Excel), включаючи нормалізацію структури колонок та обробку можливих помилок кодування. По-друге, це реалізація алгоритмів обробки природної мови для української мови з підтримкою токенізації та видалення стоп-слів. По-третє, це створення комплексної системи визначення тональності тексту з урахуванням контексту, інтенсифікаторів та заперечень. По-четверте, це забезпечення інтерактивної візуалізації результатів у вигляді різноманітних графіків, діаграм та хмар слів.

Таблиця 2.1 – Функціональні вимоги до програмного комплексу

№	Вимога	Опис
1	Завантаження та обробка даних	Підтримка CSV та Excel форматів з автоматичною нормалізацією колонок
2	Аналіз природної мови	Токенізація видалення стоп-слів для української мови
3	Сентимент-аналіз	Визначення тональності з урахуванням контексту та модифікаторів
4	Атрибутний аналіз	Виявлення та оцінка згадувань специфічних характеристик шин
5	Інтерактивна візуалізація	Графіки, діаграми, хмари слів з можливістю налаштування
6	Система фільтрації	Багаторівнева фільтрація за різними критеріями
7	Експорт результатів	Збереження аналітичних звітів у форматі Excel

Нефункціональні характеристики системи не менш важливі для забезпечення її практичної придатності. Продуктивність має бути достатньою для обробки тисяч відгуків без значної затримки відгуку інтерфейсу. Надійність системи повинна забезпечуватися через належну обробку помилок, валідацію вхідних даних та стабільну роботу при різних конфігураціях даних. Зручність використання досягається через інтуїтивний графічний інтерфейс з вкладками, контекстними підказками та логічно організованими елементами керування.

Таблиця 2.2 – Нефункціональні вимоги до програмного комплексу

№	Категорія	Вимога	Опис
1	Продуктивність	Швидкодія обробки	Аналіз до 10000 відгуків за прийнятний час
2	Надійність	Стійкість до помилок	Коректна обробка некоректних даних та кодувань
3	Використовуваність	Інтуїтивний інтерфейс	Графічний інтерфейс без потреби спеціальної підготовки
4	Підтримуваність	Модульна архітектура	Можливість розширення функціональності
5	Переносимість	Багатоплатформність	Робота на Windows, Linux, macOS

Технологічний стек для реалізації обирався з урахуванням специфіки задач обробки природної мови та візуалізації даних. Python було обрано як основну мову програмування завдяки потужним бібліотекам для аналітики. Для роботи з текстом використовуються NLTK та TextBlob, доповнені спеціалізованими словниками української мови. Обробка табличних даних здійснюється через pandas та numpy, візуалізація – через matplotlib та wordcloud. Графічний інтерфейс реалізовано засобами tkinter для забезпечення кросплатформної сумісності.

Сформульовані вимоги створюють основу для проектування архітектури системи та визначають напрямки її технічної реалізації. Вони забезпечують створення комплексного інструменту, здатного ефективно аналізувати відгуки споживачів та надавати цінну інформацію для покращення якості продукції та маркетингових стратегій.

## 2.2 Архітектура системи та вибір технологій

Архітектура програмного комплексу для аналізу відгуків про автомобільні шини реалізована на основі об'єктно-орієнтованого підходу з чітким функціональним розділенням методів всередині єдиного класу. Така структура забезпечує інкапсуляцію функціональності, спрощує процес розробки та підтримки коду, а також дозволяє логічно групувати пов'язані методи в межах єдиного класу.

Центральною ланкою системи є клас `EnhancedTireAnalysisApp`, який інкапсулює всю функціональність програми та забезпечує координацію між різними групами методів. Цей підхід дозволяє централізовано керувати станом додатку, обробляти події користувацького інтерфейсу та забезпечувати узгоджену роботу всіх компонентів системи.

Частина коду в розділі «`DATA HANDLING`» відповідає за завантаження файлів різних форматів, нормалізацію структури даних та їх попередню обробку. Особливістю реалізації є автоматичне розпізнавання кодування файлів та інтелектуальне перейменування колонок відповідно до очікуваної схеми даних. Система здатна працювати з файлами у форматах `CSV` та `Excel`, автоматично визначаючи тип даних у кожній колонці.

`Pandas` використовується як основний інструмент для маніпуляції табличними даними. Бібліотека забезпечує ефективну роботу з `DataFrame`-об'єктами, проведення операцій групування, фільтрації та агрегації. Особливо важливою є її здатність обробляти пропущені значення та автоматично визначати типи даних колонок.

`NumPy` слугує фундаментом для числових обчислень, забезпечуючи ефективну роботу з масивами даних. Векторизовані операції `NumPy` використовуються для розрахунку статистичних показників та метрик якості сентимент-аналізу.

Блок аналізу природної мови реалізує складні алгоритми обробки української мови. NLTK надає базові функції токенизації та роботи зі стоп-словами, хоча для української мови створено спеціалізовані словники. TextBlob використовується для первинного сентимент-аналізу, результати якого потім корегуються власним алгоритмом з урахуванням специфіки української мови.

Власний алгоритм сентимент-аналізу досконалий алгоритм токенизації враховує специфічні особливості української мови. Використовується спеціалізований регулярний вираз `r"\b[a-яії'"]+\b|\b[a-z'"]+\b|\b\d+\b"`, який розпізнає:

- 1) українські літери (а-я) включаючи специфічні символи (і, ї);
- 2) апостроф та його варіанти (``) для слів типу "кав'ярня";
- 3) дефіси для складних слів типу "інтернет-магазин";
- 4) латинські літери для змішаних текстів;
- 5) числові значення.

Враховує контекстні модифікатори, заперечення та інтенсифікатори. Система підтримує словники позитивних та негативних слів, адаптовані для автомобільної тематики. Алгоритм аналізує не лише окремі слова, але й їх сполучення у межах речень, що дозволяє точніше визначати тональність складних висловлювань.

Matplotlib забезпечує створення статичних графіків та діаграм з широкими можливостями налаштування зовнішнього вигляду. Бібліотека використовується для побудови гістограм розподілу оцінок, порівняльних діаграм брендів та графіків сентимент-аналізу. Інтеграція з tkinter через FigureCanvasTkAgg дозволяє вбудовувати графіки безпосередньо в інтерфейс програми.

WordCloud спеціалізується на створенні хмар слів з підтримкою різних кольорових схем та алгоритмів розташування тексту. Система реалізує унікальну функцію забарвлення слів відповідно до їх емоційного забарвлення або частоти згадування в позитивних контекстах.



### 2.3.1 Попередня обробка тексту

Етап попередньої обробки тексту має критичне значення для якості подальшого аналізу. Система реалізує багатоступеневий процес очищення та нормалізації текстових даних.

Першим кроком є токенизація тексту з використанням регулярних виразів, адаптованих для української мови. Алгоритм виділяє слова, враховуючи особливості кириличного алфавіту та специфічні символи української мови:

```
def custom_word_tokenize(self, text):
    """Tokenize text with Ukrainian language support"""
    if not isinstance(text, str):
        return []

    # Convert to lowercase first
    text = text.lower()

    # and hyphen for compound words like "інтернет-магазин"
    ukrainian_pattern = r"\b[a-яії'`'-]+\b|\b[a-z'`'-]+\b|\b\d+\b"
    tokens = re.findall(ukrainian_pattern, text)

    # Filter out single characters and clean tokens
    cleaned_tokens = []
    for token in tokens:
        # Remove leading/trailing punctuation
        cleaned_token = re.sub(r"^['-]+|['-]+$", "", token)
        if len(cleaned_token) > 1: # Keep only words longer than 1 character
            cleaned_tokens.append(cleaned_token)

    return cleaned_tokens
```

Рисунок 2.2 – функція токенизації

Система стоп-слів включає не лише загальні українські службові слова, але й спеціалізовані терміни, характерні для автомобільної тематики. Це дозволяє сфокусувати аналіз на семантично значущих елементах тексту.

### 2.3.2 Сентимент-аналіз з урахуванням контексту

Реалізований алгоритм сентимент-аналізу значно перевершує базові підходи завдяки врахуванню контекстних факторів. Система аналізує не лише присутність позитивних чи негативних слів, але й їх взаємодію з модифікаторами.

Система модифікаторів включає три категорії:

- 1) інтенсифікатори ('дуже', 'надзвичайно', 'неймовірно') підсилюють емоційне забарвлення;
- 2) пом'якшувачі ('трохи', 'дещо', 'злегка') зменшують інтенсивність;
- 3) заперечення ('не', 'ні', 'без') змінюють полярність оцінки.

Алгоритм враховує позиційні залежності між словами, аналізуючи контекстні вікна розміром до 3 слів. Це дозволяє коректно обробляти складні конструкції типу "не дуже гарний" або "зовсім не рекомендую".

### 2.3.3 Атрибутний аналіз шин

Система реалізує спеціалізований модуль для виявлення та аналізу згадувань конкретних характеристик автомобільних шин. Створено комплексний словник атрибутів на основі основних параметрів і характеристик, які згадуються в каталогах, мануалах по використанню і на сайтах дистриб'юторів, що включає понад 40 категорій характеристик шин.

Основні групи атрибутів:

- 1) зчеплення (сухий асфальт, вологі умови, сніг, лід);
- 2) зносостійкість (протектор, загальна довговічність);
- 3) комфорт (рівень шуму, якість їзди, вібрації);
- 4) керованість (маневреність, стабільність, точність);

5) економічні показники (ціна, співвідношення ціна/якість).

Для кожного атрибуту визначено множину ключових слів та фраз, що дозволяють його ідентифікувати в тексті. Алгоритм аналізує не лише факт згадування атрибуту, але й емоційне ставлення до нього в контексті конкретного речення.

```

for attribute, keywords in TIRE_ATTRIBUTES.items():
    attribute_mentions = []
    attribute_sentiments = []

    for sentence in sentences:

        if any(keyword in sentence for keyword in keywords):

            contexts = self.extract_context(sentence, keywords, 5)

            for context in contexts:
                context_sentiment = self.analyze_sentiment_with_context(context)
                attribute_mentions.append(context)
                attribute_sentiments.append(context_sentiment)

    if attribute_mentions:

        avg_sentiment = sum(attribute_sentiments) / len(attribute_sentiments)

        attribute_weight = len(attribute_mentions)

        attributes_found[attribute] = {
            'sentiment': avg_sentiment,
            'weight': attribute_weight,
            'mentions': attribute_mentions[:3]
        }

return attributes_found

```

Рисунок 2.3 – оцінка атрибутів

#### 2.3.4 Оптимізація продуктивності

Для забезпечення прийнятної швидкодії при обробці великих обсягів даних реалізовано ряд оптимізацій:

1) кешування результатів токенізації;

- 2) векторизовані операції для статистичних розрахунків;
- 3) ефективні алгоритми пошуку ключових слів;
- 4) паралельна обробка незалежних фрагментів тексту.

Розроблені алгоритми забезпечують високу точність аналізу при збереженні продуктивності системи, що робить її придатною для практичного використання в умовах реального бізнес-процесу.

## 2.4 Висновки до 2 розділу

У другому розділі здійснено комплексну розробку концепції програмного комплексу для аналізу відгуків про автомобільні шини. Визначено функціональні та нефункціональні вимоги до системи, що включають підтримку завантаження різних форматів даних, проведення глибокого сентимент-аналізу з урахуванням специфіки української мови, реалізацію атрибутного аналізу характеристик шин та забезпечення інтерактивної візуалізації результатів.

Запропонована процедурна модульність системи забезпечує гнучкість розвитку, простоту супроводу та можливість розширення функціональності. Обґрунтовано вибір технологічного стеку, що включає Python з бібліотеками pandas, numpy для обробки даних, NLTK та TextBlob для аналізу природної мови, matplotlib та wordcloud для візуалізації, tkinter для створення користувачького інтерфейсу.

Реалізовано алгоритми сентимент-аналізу, що враховують контекстні модифікатори, заперечення та інтенсифікатори, характерні для української мови. Створено спеціалізовану систему атрибутного аналізу з комплексним словником характеристик автомобільних шин, що дозволяє виявляти та оцінювати згадування конкретних властивостей продукції.

Теоретичні розробки та технічні рішення, представлені в даному розділі, створюють міцну основу для практичної реалізації ефективного інструменту

аналізу споживчих відгуків, здатного надавати цінну інформацію для покращення якості продукції та оптимізації маркетингових стратегій.

## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ

### 3.1 Опис структури та компонентів системи

Програмний комплекс для аналізу відгуків про автомобільні шини реалізовано на основі об'єктно-орієнтованої архітектури з чітким розподілом функціональних обов'язків між компонентами. Така організація забезпечує модульність системи, спрощує процес тестування та дозволяє незалежно розвивати окремі функціональні блоки без впливу на загальну стабільність додатку.

Структурна організація системи реалізована у вигляді єдиного класу `EnhancedTireAnalysisApp`, що інкапсулює всю функціональність програми. Такий підхід обрано для спрощення розробки та забезпечення цілісності системи на етапі прототипування. Структурна організація системи включає в себе вісім основних функціональних груп методів, кожна з яких відповідає за конкретний аспект роботи програми. Така логічна декомпозиція дозволяє ефективно організувати код та забезпечити його читабельність і підтримуваність.

Таблиця 3.1 – Функціональні компоненти програмного комплексу

Компонент	Опис функціональності
Утилітарні методи	Налаштування стилів інтерфейсу, ініціалізація словників стоп-слів для української мови
Модуль текстового аналізу	Токенізація, контекстний аналіз, сентимент-аналіз з урахуванням модифікаторів
Система обробки даних	Завантаження файлів, нормалізація колонок, валідація та попередня обробка даних
Підсистема фільтрації	Створення динамічних фільтрів, застосування критеріїв відбору, скидання налаштувань
Інтерфейс користувача	Створення GUI з вкладками, елементами керування та областями відображення результатів
Модуль візуалізації	Генерація хмар слів, статистичних графіків, діаграм розподілу та порівняльних графіків
Система експорту	Формування звітів Excel з багатьма аркушами, збереження результатів аналізу
Обробники подій	Координація взаємодії між компонентами інтерфейсу та логічними модулями

Особливістю реалізації є використання асинхронного підходу до обробки даних, що дозволяє зберігати відгук інтерфейсу навіть при роботі з великими обсягами текстової інформації. Система індикації стану (зміна курсору на "wait") інформує користувача про виконання тривалих операцій.

Архітектура передбачає гнучке управління пам'яттю через кешування результатів обчислень. Результати токенізації та сентимент-аналізу зберігаються для повторного використання, що значно прискорює роботу з фільтрами та оновлення візуалізацій.

Система контролю якості включає багаторівневу валідацію вхідних даних: перевірку формату файлів, кодування тексту, структури колонок та

цілісності даних. Кожен етап обробки супроводжується детальним логуванням для можливості діагностики проблем.

Модульність архітектури дозволяє легко розширювати функціональність системи. Додавання нових алгоритмів аналізу тексту або типів візуалізації не потребує модифікації існуючого коду, а реалізується через створення додаткових методів у відповідних функціональних групах.

Система підтримки української мови реалізована через спеціалізовані словники та алгоритми, адаптовані до особливостей морфології та синтаксису української мови. Це включає розширені набори стоп-слів, словники емоційно забарвлених слів та правила обробки заперечень і модифікаторів.

Інтеграція з зовнішніми бібліотеками здійснюється через єдині інтерфейси, що дозволяє за потреби замінювати компоненти без суттєвих змін у коді. Наприклад, система візуалізації може використовувати різні backend-и `matplotlib` залежно від операційного середовища.

### 3.2 Інтерфейс користувача

Графічний інтерфейс користувача розроблено з використанням бібліотеки `tkinter`, що забезпечує кросплатформну сумісність та не потребує додаткових залежностей. Дизайн інтерфейсу орієнтовано на принципи зручності використання та інтуїтивної навігації, що дозволяє ефективно працювати з системою навіть користувачам без технічної підготовки.

Головне вікно програми організовано за принципом функціонального зонування з використанням адаптивної компоновки. Розмір вікна автоматично максимізується при запуску для оптимального використання доступного простору екрану, при цьому встановлено мінімальні розміри  $1200 \times 800$  пікселів для забезпечення коректного відображення всіх елементів.

### *Структура інтерфейсу*

1. Панель управління файлами – верхня частина інтерфейсу містить заголовок програми та дві ключові кнопки:
  - 1) "Завантажити файл" – дозволяє обрати CSV або Excel файл для аналізу;
  - 2) "Експортувати результати" – зберігає результати аналізу в Excel форматі.
2. Область фільтрів – ліва частина інтерфейсу призначена для налаштування параметрів аналізу:
  - 1) кнопки управління фільтрами: "Застосувати фільтри" та "Скинути фільтри" розташовані у верхній частині панелі;
  - 2) налаштування хмари слів: включає опції забарвлення за оцінками або настроєм, вибір кольорової схеми та регулювання максимальної кількості слів;
  - 3) стандартні фільтри: для колонок даних (бренд, модель, рейтинг, пробіг тощо) з різними типами елементів керування залежно від типу даних;
  - 4) атрибутивні фільтри: згруповані за категоріями (Зчеплення, Зносостійкість, Комфорт, Керованість, Сезонність, Економічність) з можливістю встановлення мінімального рівня настрою.

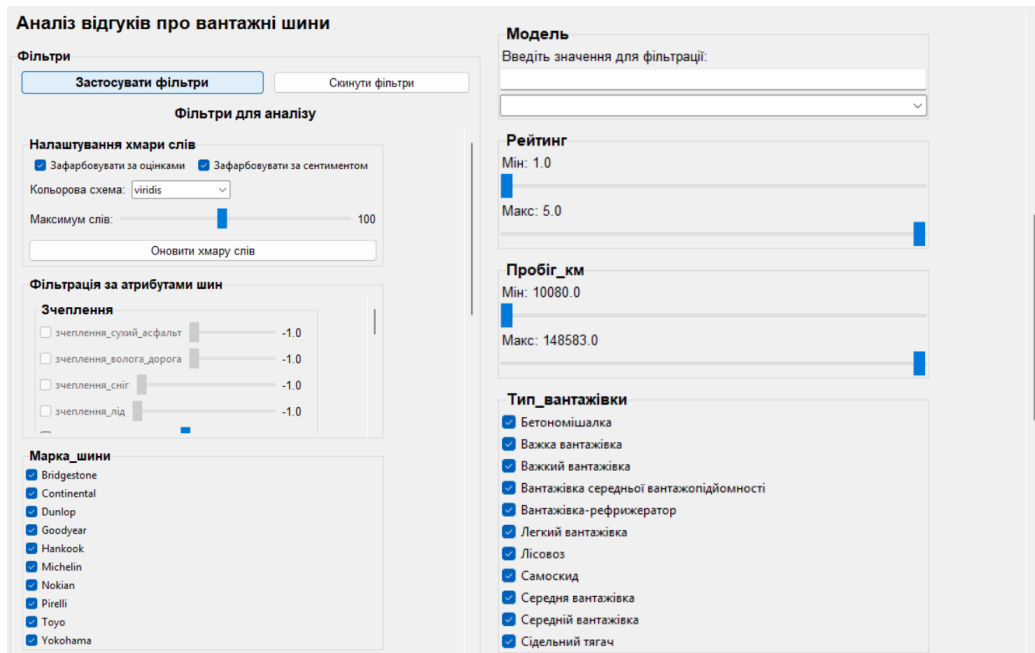


Рисунок 3.1-3.2 – Область фільтрів з налаштуваннями хмари слів

3. Область візуалізації – центральна та найбільша частина інтерфейсу організована у вигляді вкладок:

- 1) вкладка "Хмара слів": відображає інтерактивну хмару найчастіших слів з можливістю забарвлення за настроємом або оцінками;
- 2) вкладка "Аналіз оцінок": містить гістограму розподілу оцінок з статистичними показниками;
- 3) вкладка "Порівняння брендів": демонструє середні оцінки або кількість відгуків за брендами;
- 4) вкладка "Аналіз настрою": показує розподіл тональності відгуків з категоризацією (див. Рисунок 3.5);
- 5) вкладка "Аналіз атрибутів": горизонтальна діаграма згадувань характеристик шин з їх оцінкою настрою (див. Рисунок 3.6).

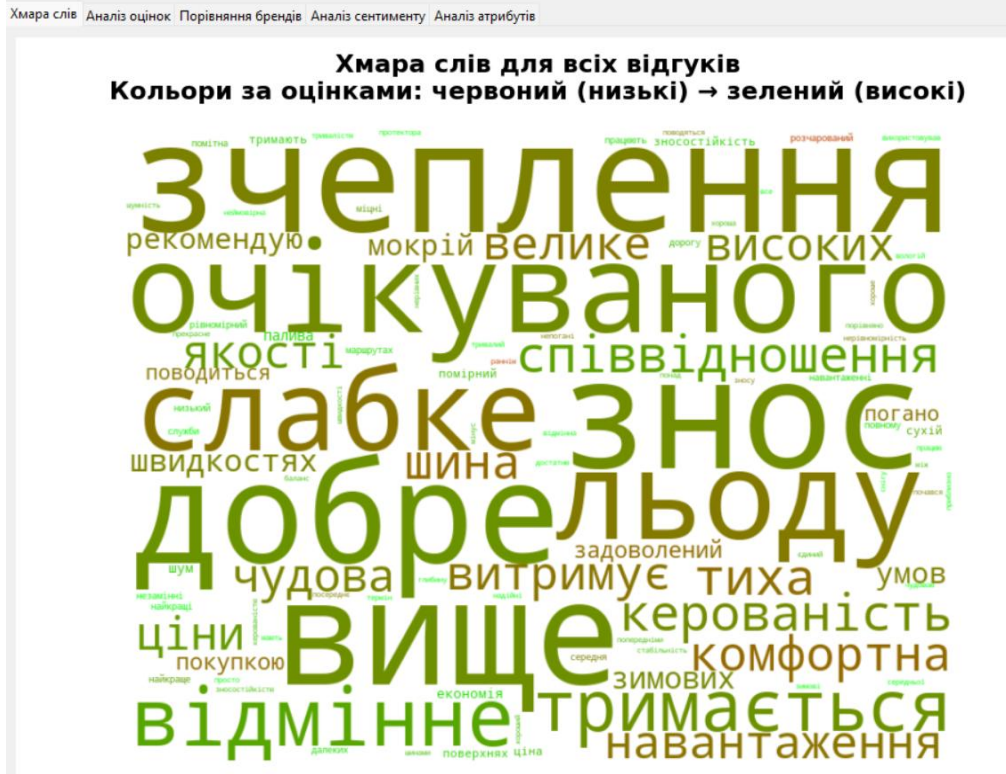


Рисунок 3.3 – Вкладка "Хмара слів" з кольоровим кодуванням за сен-тимен-  
 ТОМ

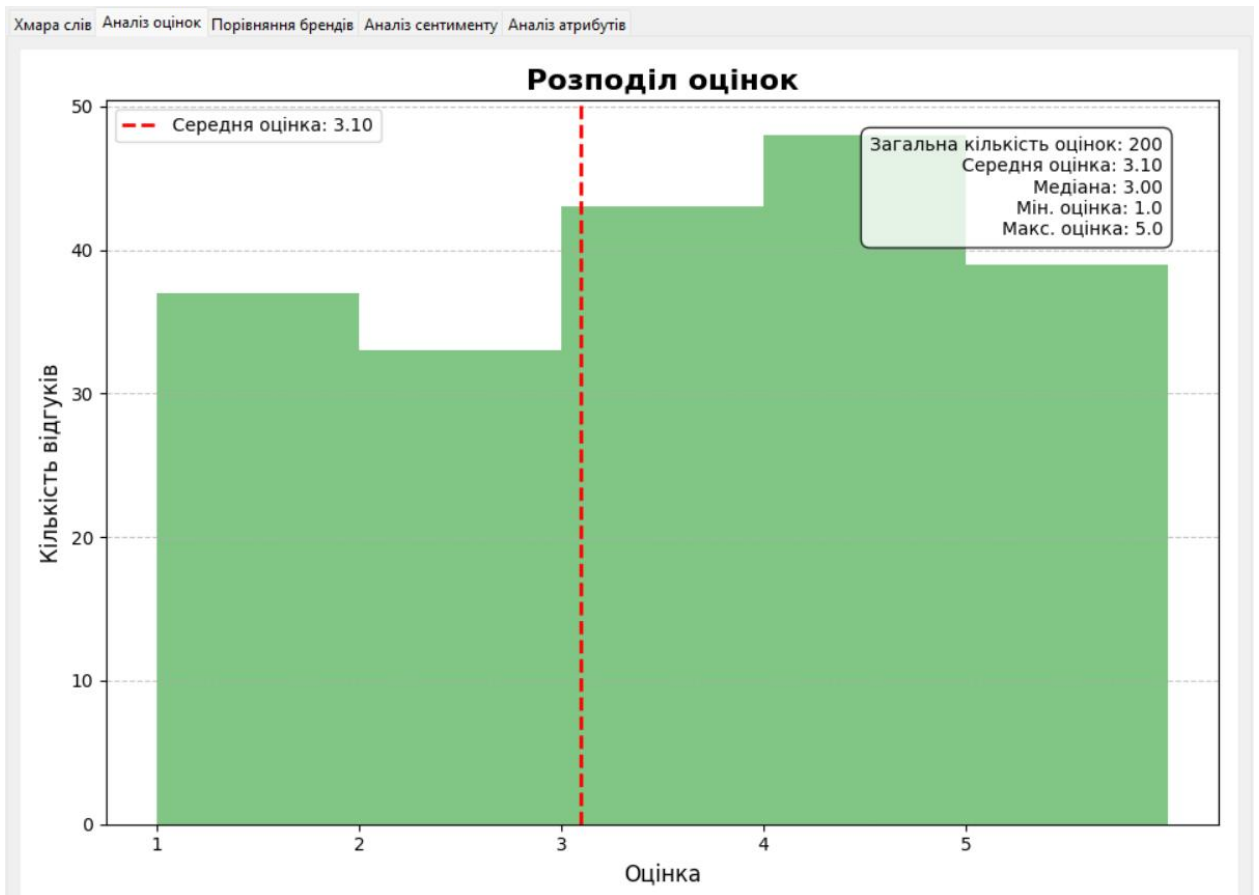


Рисунок 3.4 – Вкладка "Аналіз оцінок" з гістограмою розподілу рейтингів

Інтерфейс реалізує адаптивний дизайн з використанням сепараторів (PanedWindow), що дозволяє користувачеві регулювати розміри областей залежно від потреб. Система стилізації забезпечує консистентний зовнішній вигляд всіх елементів з використанням різних шрифтів та кольорових схем для різних типів інформації.

Особливістю реалізації є інтелектуальне оновлення контенту: при зміні фільтрів або перемиканні між вкладками відбувається автоматичне оновлення відповідних візуалізацій без необхідності ручного оновлення.

Для покращення користувацького досвіду реалізовано контекстні підказки, прогрес-індикатори для тривалих операцій та інформативні повідомлення про стан обробки даних.

### 3.3 Алгоритми обробки та аналізу даних

Основою програмного комплексу є система обробки та аналізу текстових даних, що реалізує комплексний підхід до розуміння семантики та емоційного забарвлення відгуків споживачів. Розроблені алгоритми враховують специфіку української мови та особливості автомобільної термінології.

#### 3.3.1 Алгоритм завантаження та нормалізації даних

Система підтримує роботу з файлами у форматах CSV та Excel, автоматично визначаючи оптимальні параметри читання. Алгоритм послідовно намагається відкрити файл з різними кодуваннями (UTF-8, CP1251, ISO-8859-1) до успішного результату.

Процедура нормалізації колонок використовує інтелектуальний мапінг назв, що дозволяє автоматично розпізнавати колонки з рейтингами, коментарями, брендами та датами незалежно від їх оригінальних назв. Система також виявляє та видаляє BOM-маркери, що можуть виникати при експорті з різних програм.

#### 3.3.2 Алгоритм сентимент-аналізу

Розроблений алгоритм сентимент-аналізу є гібридним рішенням, що поєднує базовий аналіз через TextBlob з власною системою контекстного аналізу української мови.

Система використовує гібридний підхід, що поєднує бібліотеку TextBlob з власним лексичним аналізатором. TextBlob надає базову оцінку з ваговим коефіцієнтом 0.2, а спеціалізований аналізатор формує основну оцінку з коефіцієнтом 0.8.

Ключовою особливістю є контекстуальний аналіз з динамічним відстеженням модифікаторів. Алгоритм ідентифікує негачії (не, ні), підсилювачі (дуже, надзвичайно) та послаблювачі (трохи, злегка), застосовуючи їх до наступних емоційно забарвлених слів. Модифікатори автоматично скидаються через певну кількість слів: негачії через 3 слова, інтенсифікатори через 2 слова.

Система оперує спеціалізованою лексикою для автомобільної індустрії, включаючи терміни зчеплення, зносостійкість, керованість, аквапланування. Для уникнення подвійного підрахунку ведеться облік уже оброблених слів у межах одного тексту.

Особливим рішенням є атрибутний аналіз настрою. Для кожного з 50+ атрибутів шин система витягує контекст у вікні 5 слів навколо ключових термінів і обчислює індивідуальну оцінку настрою. Це дозволяє отримувати не лише загальний настрій відгуку, але й оцінки конкретних характеристик товару.

Фінальна оцінка формується через нормалізацію з фактором згладжування та обмежується діапазоном від -1 до 1. Така архітектура забезпечує високу точність аналізу технічних відгуків, де важливі як емоційне забарвлення, так і оцінки специфічних властивостей продукту.

#### *Основні компоненти алгоритму.*

1. Система модифікаторів – включає три типи слів-модифікаторів:
  - 1) посилювачі ('дуже', 'надзвичайно', 'неймовірно') – підсилюють емоційну оцінку в 1.5 рази;
  - 2) пом'якшувачі ('трохи', 'дещо', 'злегка') – зменшують інтенсивність наполовину;
  - 3) Заперечення ('не', 'ні', 'без') – інвертують полярність оцінки.

2. Контекстний аналіз – враховує позиційні залежності між словами в межах речення:

- 1) модифікатори діють на наступні 2-3 слова;
- 2) заперечення може поширюватися на ціле речення;
- 3) система запобігає подвійному підрахунку однакових слів.

3. Нормалізація результатів – Система формує оцінку настрою в діапазоні  $[-1; +1]$ , де  $-1$  відповідає максимально негативному настрою,  $0$  – нейтральному,  $+1$  – максимально позитивному. Значення понад  $0.2$  класифікуються як позитивні, нижче  $-0.2$  – як негативні.

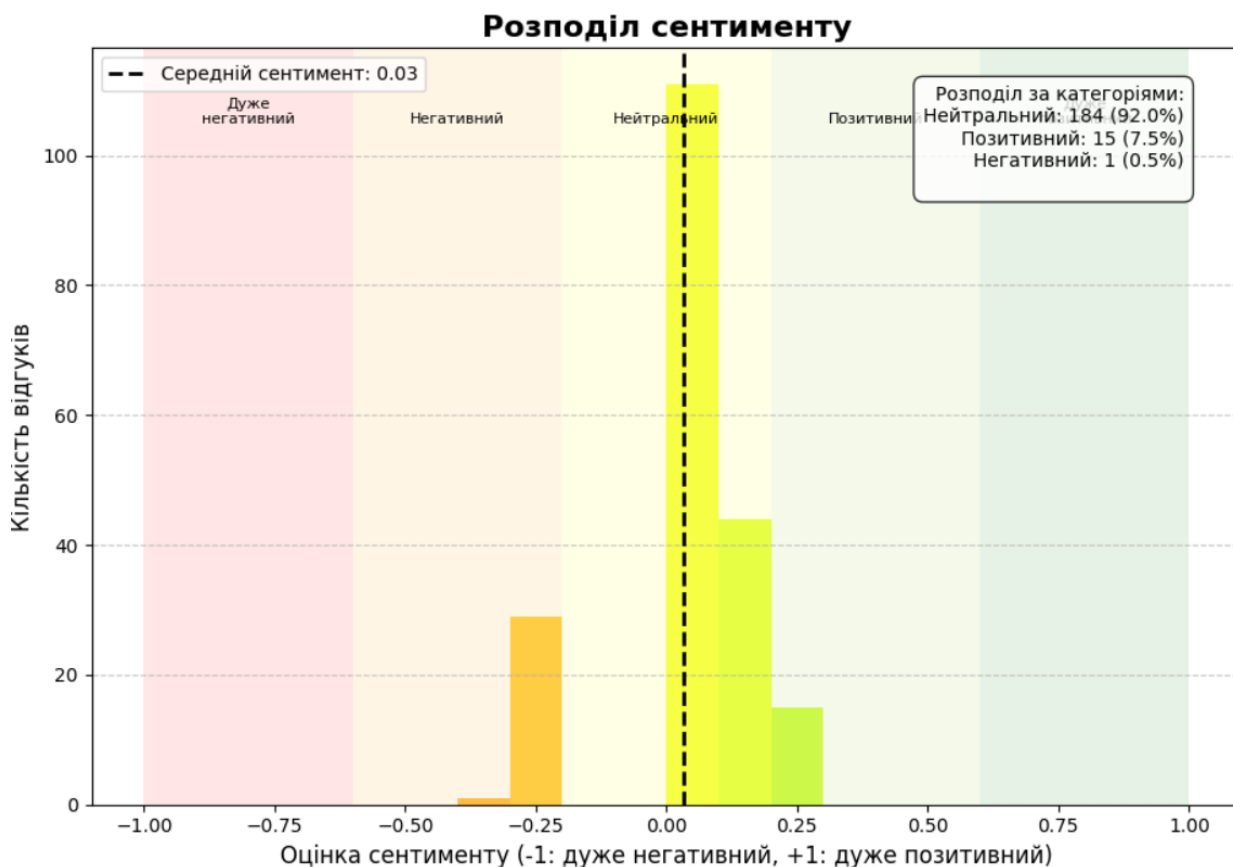


Рисунок 3.5 – Діаграма розподілу настрою з категоризацією тональності

### 3.3.3. Алгоритм атрибутного аналізу

Система атрибутного аналізу використовує словник з понад 40 категорій характеристик шин, згрупованих за тематичними областями. Кожна категорія містить набір ключових слів та фраз для ідентифікації згадувань у тексті.

Алгоритм працює в кілька етапів:

- 1) розбиття тексту на речення;
- 2) пошук ключових слів для кожного атрибута;
- 3) виділення контексту навколо знайдених згадувань;
- 4) сентимент-аналіз кожного контексту;
- 5) агрегація результатів з урахуванням ваги згадувань.

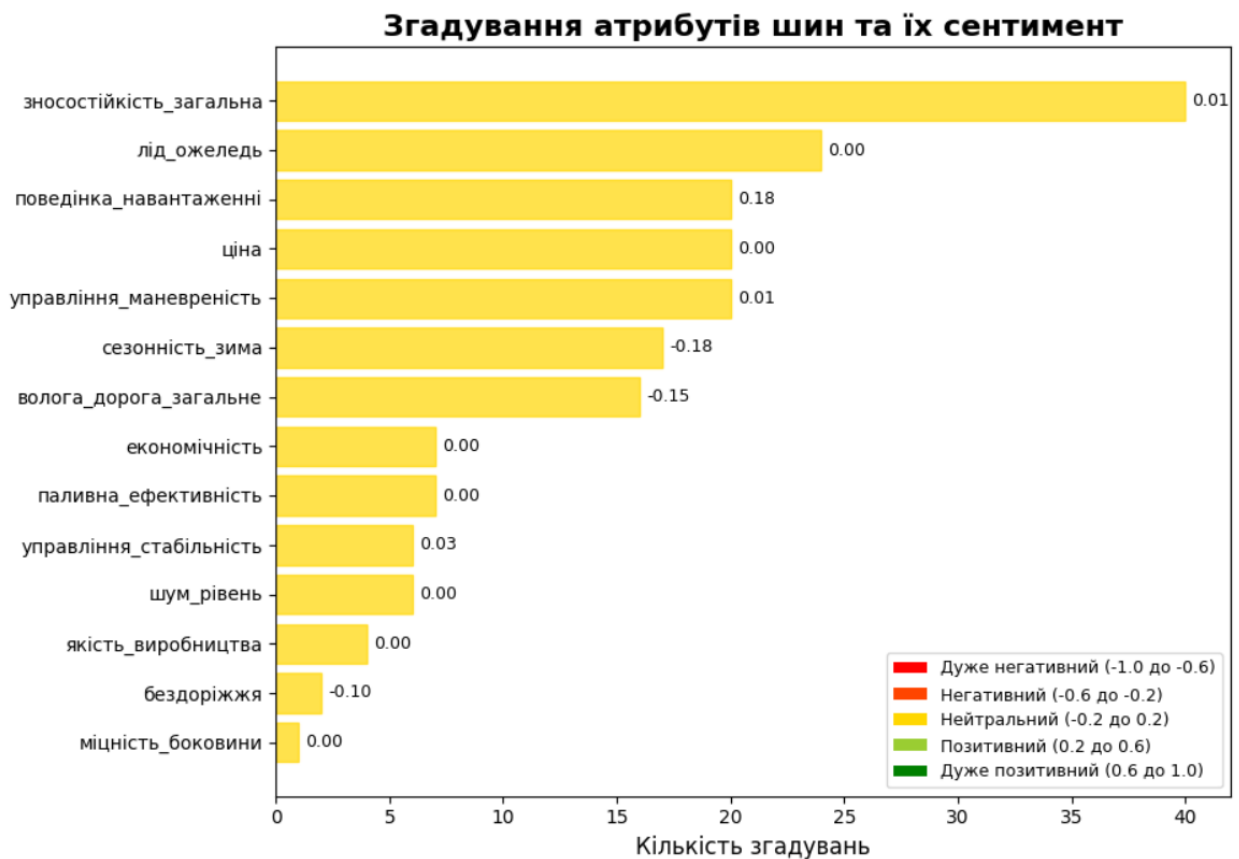


Рисунок 3.6 – Горизонтальна діаграма згадувань атрибутів шин з сентимент-оцінкою

### 3.3.4 Алгоритм генерації візуалізацій

Система візуалізації використовує динамічний підхід до створення графіків залежно від наявних даних та активних фільтрів.

Алгоритм створення хмари слів:

- 1) токенізація всіх коментарів з фільтрованих даних;
- 2) видалення стоп-слів та коротких слів (менше 3 символів);
- 3) підрахунок частоти слів з урахуванням контексту;
- 4) забарвлення слів за настроєм або рейтингами (опціонально);
- 5) генерація хмари з налаштованими параметрами.

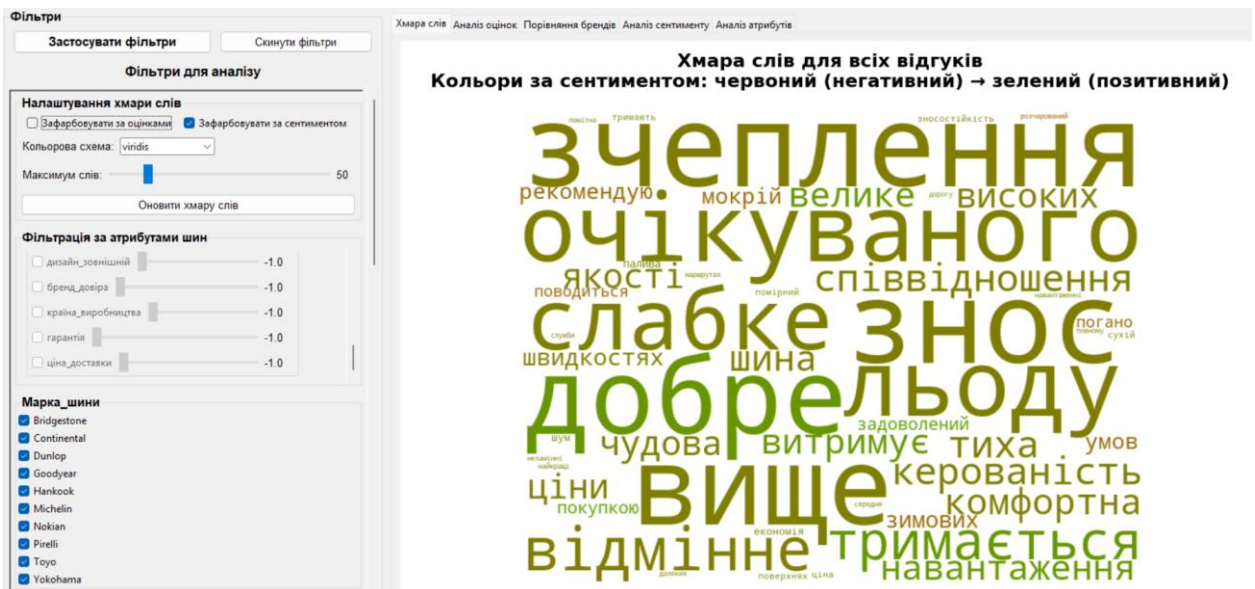


Рисунок 3.7 – Інтерфейс системи фільтрації з активними атрибутними фільтрами

Розроблена система фільтрації підтримує багаторівневі критерії відбору:

- 1) числові фільтри – діапазони значень з інтерактивними повзунками;
- 2) категоріальні фільтри – множинний вибір через чекбокси;
- 3) текстові фільтри – пошук за підстрокою з ігноруванням регістру;

- 4) атрибуtnі фільтри – мінімальний рівень настрою для конкретних характеристик;
- 5) часові фільтри – діапазони дат з автоматичним парсингом форматів.

Фільтри застосовуються послідовно з логуванням кількості записів на кожному етапі. Система автоматично перераховує статистику та оновлює всі візуалізації при зміні критеріїв відбору.

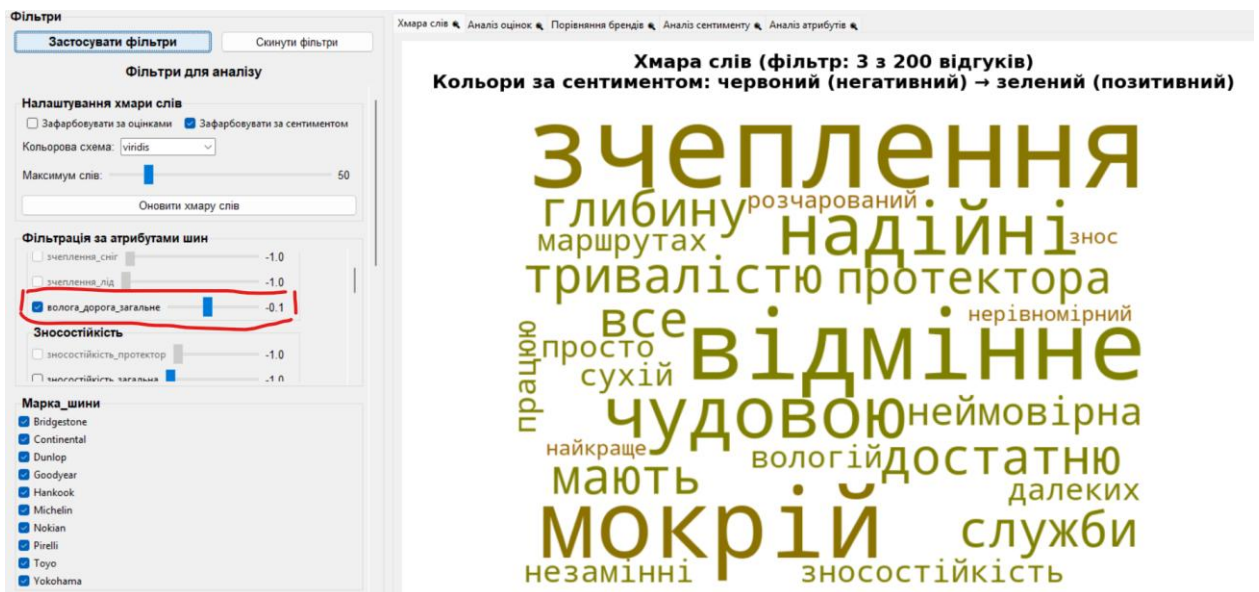


Рисунок 3.8 – Хмара слів після застосування фільтрів з позитивним забарвленням

Алгоритми оптимізовано для роботи з великими обсягами даних через використання векторизованих операцій pandas та ефективних структур даних. Система кешування результатів дозволяє уникнути повторних обчислень при зміні параметрів візуалізації.

### 3.4. Система експорту та звітності

Програмний комплекс включає комплексну систему експорту результатів аналізу, що дозволяє зберігати отримані дані у структурованому вигляді для подальшого використання в бізнес-процесах або наукових дослідженнях.

#### 3.4.1 Структура експортованих звітів

Система генерує Excel-файл з множинними аркушами, кожен з яких містить специфічний тип аналітичної інформації.

1. Аркуш "Відфільтровані дані" – містить повний набір записів, що пройшли через систему фільтрів, включаючи оригінальні дані та результати сентимент-аналізу.
2. Аркуш "Статистика по брендах" – агреговані показники за виробниками шин:
  - 1) кількість відгуків по кожному бренду;
  - 2) середні, мінімальні та максимальні оцінки;
  - 3) середні показники сентименту з діапазонами варіації.
3. Аркуш "Статистика сентименту" – загальні метрики тональності:
  - 1) дескриптивна статистика (середнє, медіана, стандартне відхилення);
  - 2) розподіл за категоріями сентименту;
  - 3) кількісні показники по кожній тональній групі.
4. Аркуш "Статистика атрибутів" – детальний аналіз характеристик шин:
  - 1) частота згадування кожного атрибута;
  - 2) середні оцінки сентименту за характеристиками;

- 3) рейтинг атрибутів за кількістю позитивних згадувань.
5. Аркуш "Бренди+Атрибути" – перехресний аналіз:
  - 1) матриця бренд-атрибут з показниками настрою;
  - 2) кількість згадувань кожної характеристики для конкретних брендів;
  - 3) порівняльні оцінки атрибутів між виробниками.
6. Аркуш "Частота слів" – 100 найчастіших слів з їх частотами появи в текстах відгуків.
7. Аркуш "Приклади атрибутів" – конкретні фрагменти відгуків з контекстом згадування атрибутів для якісного аналізу.

G	H	I	J	K	L	M	N	O	P
Поверхня_доріг	Сезон	Ціна_грн	Дата_відгуку	Коментар	Загальний_настрій	Оцінка_настрою	Атрибут_зносостійкість_загальна	Атрибут_волога_дорога_загальна	Атрибут_ш
1	Асфальт/Бетон	Всесезонні	12500	2024-12-04 00:00:00	Дуже надійні шини з чудовою тривалістю слук	0	Нейтральний	0	0
2	Асфальт/Бетон	Літні	11800		Хороший баланс між керуваністю та зносостій	0,2	Нейтральний	0,2	
3	Змішані дороги	Всесезонні	10900	2024-02-04 00:00:00	Неймовірна зносостійкість! Працюю на далеки	0	Нейтральний	0	
4	Асфальт	Всесезонні	9800		Середня якість. Знос почався вже після 40 тис.	0	Нейтральний	0	0
5	Асфальт/Грунт	Всесезонні	8900	2024-08-04 00:00:00	Прекрасне співвідношення ціна/якість. Добре	0,2	Нейтральний		
6	Асфальт	Літні	13200		Відмінна керуваність при повному навантажен	0	Нейтральний	0	
7	Асфальт/Бетон	Літні	9500		Непогані шини, але є нерівномірність зносу піс	0,2	Нейтральний	0	
8	Асфальт/Сніг	Зимові	14500	2024-05-02 00:00:00	Найкращі зимові шини, які я використовував. І	-0,2	Нейтральний	0,2	
9	Асфальт	Всесезонні	10500	2024-10-04 00:00:00	Хороша керуваність і стабільність. Помірний ш	0	Нейтральний		
10	Грунт/Кварт	Всесезонні	11200		Розчарований раннім зносом. Для бездоріжж	-0,32	Негативний	-0,2	
11	Змішані дороги	Всесезонні	15200		Найкращі шини для будівельної техніки. Нейм	0,2	Нейтральний		
12	Змішані дороги	Всесезонні	12800		Дуже міцні боксовини, що важливо для нашої	0,2	Нейтральний		
13	Асфальт/Грунт	Всесезонні	11500	2024-05-03 00:00:00	Висока паливна економічність. Рівномірний з	0	Нейтральний	0	
14	Асфальт	Літні	11500		Міцні, але швидше зношуються ніж заявлено.	0	Нейтральний	0	
15	Грунт/Кварт	Всесезонні	13500		Найкраще співвідношення ціна/якість на ринк	0	Нейтральний	0	
16	Асфальт	Всесезонні	9200		Найкраще співвідношення ціна/якість на ринк	0	Нейтральний	0	
17	Асфальт	Літні	12000	2024-10-02 00:00:00	Розчарований. Нерівномірний знос вже після	-0,2	Нейтральний	0	0
18	Асфальт/Бетон	Всесезонні	10200		Добре тримають дорогу навіть при повному на	0,2	Нейтральний		
19	Грунт/Сніг	Зимові	15800		Незамінні в суворих зимових умовах. Викорис	0	Нейтральний		
20	Асфальт	Літні	9800		Середня якість. Прийнятна ціна. Зносостійкості	0	Нейтральний	0	
21	Асфальт	Всесезонні	10800		Низький опір колюченню, що забезпечує хорошу	0	Нейтральний	0	
22	Змішане покриття	Літо	6873	2024-08-04 00:00:00	Знос вище очікуваного.	0	Нейтральний	0	
23	Змішане покриття	Всесезон	6817		Тиха і комфортна шина.	0	Нейтральний	0	
24	Асфальт	Зима	2755		Відмінне співвідношення ціни та якості.	0	Нейтральний		
25	Змішане покриття	Літо	5672	2024-09-01 00:00:00	Витримує велике навантаження.	0,2	Нейтральний		
26	Змішане покриття	Літо	6695	2024-01-06 00:00:00	Слабке зчеплення на льоду.	0	Нейтральний		
27	Гравій	Літо	7368		Слабке зчеплення на льоду.	0	Нейтральний		

Рисунок 3.9 – Приклад експортованих даних у форматі Excel з результатами аналізу

### 3.4.2 Алгоритм формування звітів

Процес експорту реалізований через `pandas ExcelWriter` з оптимізацією продуктивності.

Система автоматично обробляє відсутні дані, форматує числові значення та створює зрозумілі заголовки колонок українською мовою. Великі таблиці автоматично розбиваються на сторінки для зручності друку.

### 3.5 Висновки до 3 розділу

У третьому розділі було детально розглянуто практичну реалізацію програмного комплексу для аналізу відгуків про автомобільні шини. Основним результатом цього етапу стало створення функціональної системи, що успішно перетворює теоретичні концепції обробки природної мови в практичний інструмент бізнес-аналітики.

Архітектурне рішення, побудоване на принципах модульності та об'єктно-орієнтованого підходу, забезпечило створення стабільної та розширюваної системи. Центральний клас `EnhancedTireAnalysisApp` ефективно координує роботу всіх компонентів, від завантаження даних до генерації звітів. Восьмирівнева функціональна декомпозиція дозволяє незалежно розвивати окремі аспекти системи без впливу на загальну стабільність.

Графічний інтерфейс користувача, розроблений з використанням `tkinter`, демонструє ефективне поєднання функціональності та зручності використання. Організація інтерфейсу у вигляді чотирьох функціональних зон з адаптивною компоновкою забезпечує інтуїтивну взаємодію з системою. Особливо вдалим рішенням є використання вкладкової структури для різних типів візуалізації та динамічна система фільтрів з підтримкою атрибутного аналізу.

Алгоритмічна основа системи представляє собою комплексне рішення для обробки української мови в контексті автомобільної тематики. Гібридний підхід до сентимент-аналізу, що поєднує базові можливості TextBlob з власною системою контекстного аналізу, забезпечує високу точність визначення тональності. Система модифікаторів, заперечень та інтенсифікаторів адаптована до специфіки української мови та дозволяє коректно інтерпретувати складні мовні конструкції.

Особливої уваги заслуговує реалізація атрибутного аналізу з використанням спеціалізованого словника характеристик шин. Система здатна виявляти та оцінювати згадування понад 40 різних атрибутів продукції, групуючи їх за тематичними категоріями та надаючи детальну аналітику сентименту для кожної характеристики.

Система експорту та звітності забезпечує комплексне збереження результатів аналізу у структурованому вигляді. Генерація багатоаркушних Excel-звітів з різними типами аналітики дозволяє користувачам отримати повну картину результатів дослідження та використовувати їх для подальшого аналізу або презентації результатів керівництву.

Реалізований програмний комплекс повністю відповідає поставленим функціональним вимогам та демонструє практичну придатність для вирішення завдань аналізу споживчих відгуків у автомобільній індустрії. Система може бути адаптована для аналізу інших типів продукції або послуг завдяки модульній архітектурі та гнучким налаштуванням алгоритмів обробки тексту.

## РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінювання основних характеристик розробленого програмного комплексу для аналізу відгуків про автомобільні шини з використанням методів обробки природної мови.

Створена система дозволить підприємствам автомобільної індустрії здійснювати якісний аналіз споживчих відгуків не тільки на вітчизняному ринку, але й при роботі з міжнародними платформами електронної комерції.

У дослідженні розглядаються різноманітні варіанти технічної реалізації для забезпечення найбільш раціонального та економічно виправданого рішення, що впливає на фінансові показники та сумісність з існуючими інформаційними системами підприємств. Для цього застосовувався інструментарій функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) являє собою методологію, що дозволяє визначити реальну вартість програмного продукту незалежно від організаційної структури розробника. ФВА здійснюється з метою виявлення можливостей зниження витрат через використання більш ефективних технологічних рішень, досягнення кращого співвідношення між функціональною цінністю продукту та затратами на його створення.

Методика функціонально-вартісного аналізу передбачає визначення послідовності етапів створення продукту, розрахунок повних витрат (річних) та обсягу робочого часу, визначення джерел витрат та остаточний розрахунок собівартості програмного комплексу.

#### 4.1 Постановка задачі проектування

У роботі застосовується методика ФВА для здійснення техніко-економічного аналізу розробки системи аналізу відгуків про автомобільні шини. Оскільки рішення щодо проектування та реалізації компонентів системи впливають на весь програмний комплекс, кожна окрема підсистема повинна відповідати загальним вимогам. Тому практичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення сентимент-аналізу текстових відгуків.

Технічні вимоги до програмного комплексу є наступні:

- 1) функціонування на персональних комп'ютерах з оперативною пам'яттю від 8 Гб;
- 2) точність визначення тональності українськомовних текстів не менше 85%;
- 3) швидкість обробки великих масивів текстових даних до 10000 відгуків за годину;
- 4) можливість інтеграції з різними форматами даних (CSV, Excel);
- 5) мінімальні витрати на впровадження та супровід системи.

#### 4.2 Обґрунтування функцій програмного продукту

Головна функція  $F_0$  – розробка програмного комплексу для аналізу відгуків, яка дозволяє досліджувати різні характеристики, що безпосередньо впливають на якість продукції підприємства. Базуючись на цій функції, можна виділити наступні функції:

$F_1$  – вибір технологічної платформи розробки.

$F_2$  – вибір методу реалізації алгоритмів NLP.

F<sub>3</sub> – вибір підходу до створення користувацького інтерфейсу.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F<sub>1</sub>:

- 1) python з бібліотеками NLTK/TextBlob/pymorphy2;
- 2) python з використанням трансформерних моделей.

Функція F<sub>2</sub>:

- 1) використання гібридного підходу (лексиконний + статистичний);
- 2) використання нейромережових методів.

Функція F<sub>3</sub>:

- 1) створення настільного додатку (tkinter);
- 2) розробка веб-інтерфейсу (Flask/Django).

Таблиця 4.1 – Морфологічна карта системи

Функції	Варіанти реалізації
F <sub>1</sub>	а) Python + NLTK/TextBlob
F <sub>2</sub>	а) Гібридний підхід
F <sub>3</sub>	а) Настільний додаток (tkinter)

Таблиця 4.2 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F <sub>1</sub>	А	Швидкість розробки, багатство бібліотек для української мови	Менша точність порівняно з сучасними методами
	Б	Висока точність, сучасні алгоритми	Потреба у великих обчислювальних ресурсах
F <sub>2</sub>	А	Швидкість роботи, інтерпретованість результатів	Менша точність на складних мовних конструкціях
	Б	Висока точність, адаптивність	Потреба у великих наборах даних для навчання
F <sub>3</sub>	А	Простота розгортання, автономність роботи	Обмежені можливості масштабування
	Б	Масштабованість, віддалений доступ	Потреба у серверній інфраструктурі

На основі аналізу позитивно-негативної матриці розглядаємо такі варіанти реалізації ПП:

$$F_{1a} - F_{2a} - F_3,$$

$$F_{1a} - F_{2б} - F_{3a},$$

$$F_{1б} - F_{2б} - F_{3б}.$$

### 4.3 Обґрунтування системи параметрів програмного продукту

Для характеристики програмного продукту використовуватимемо наступні параметри:

$X_1$  – швидкість обробки текстових даних;

$X_2$  – точність sentiment-аналізу;

$X_3$  – час розробки системи;

$X_4$  – складність впровадження та супроводу.

Таблиця 4.3 – Основні параметри програмного продукту

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкість обробки текстових даних	$X_1$	відгуки/хв	50	125	200
Точність sentiment-аналізу	$X_2$	%	75	85	95
Час розробки системи	$X_3$	місяці	9	6	3
Складність впровадження	$X_4$	бали (1-10)	9	4	1

За даними таблиці 4.3 будуються графічні залежності бальної оцінки параметра від його абсолютного значення. Для кожного параметра виробу будують графічні залежності бальної оцінки параметра від його абсолютного значення, використовуючи дані таблиці. У цих графіках значення будь-якого параметра оцінюється в балах: краще значення — в 10 балів, гірше — в 1 бал. Експерти мають визначити характер залежності.

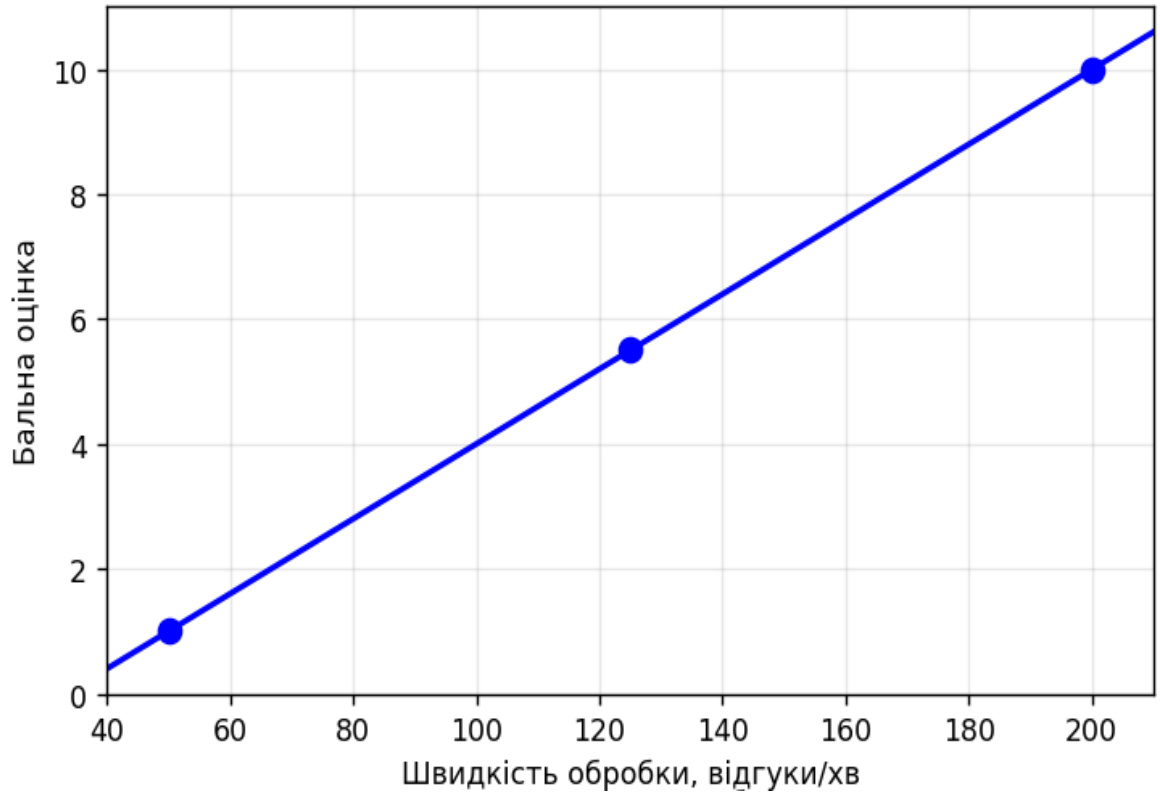


Рисунок 4.1

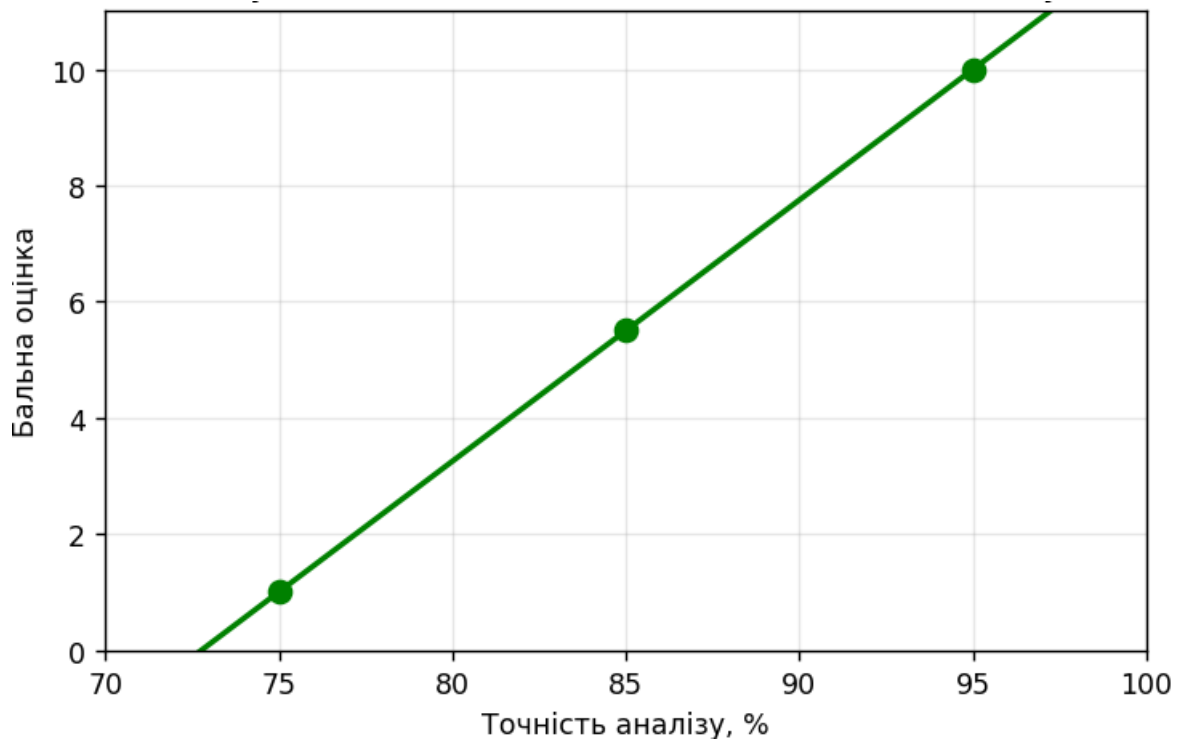


Рисунок 4.2

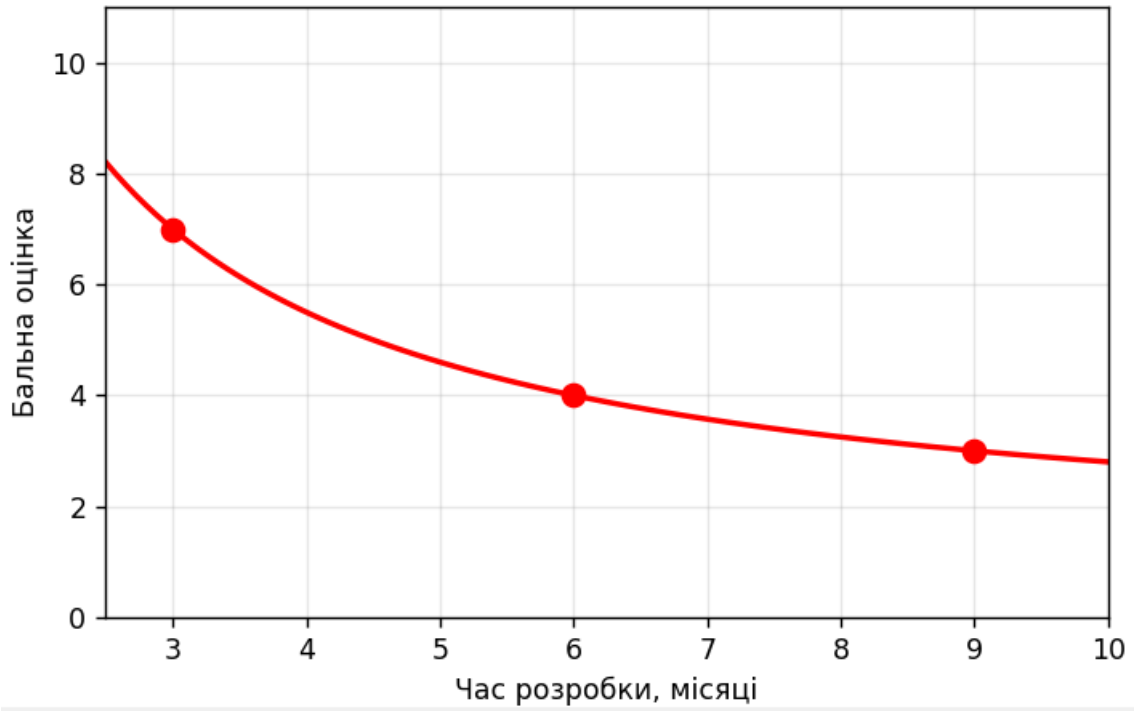


Рисунок 4.3

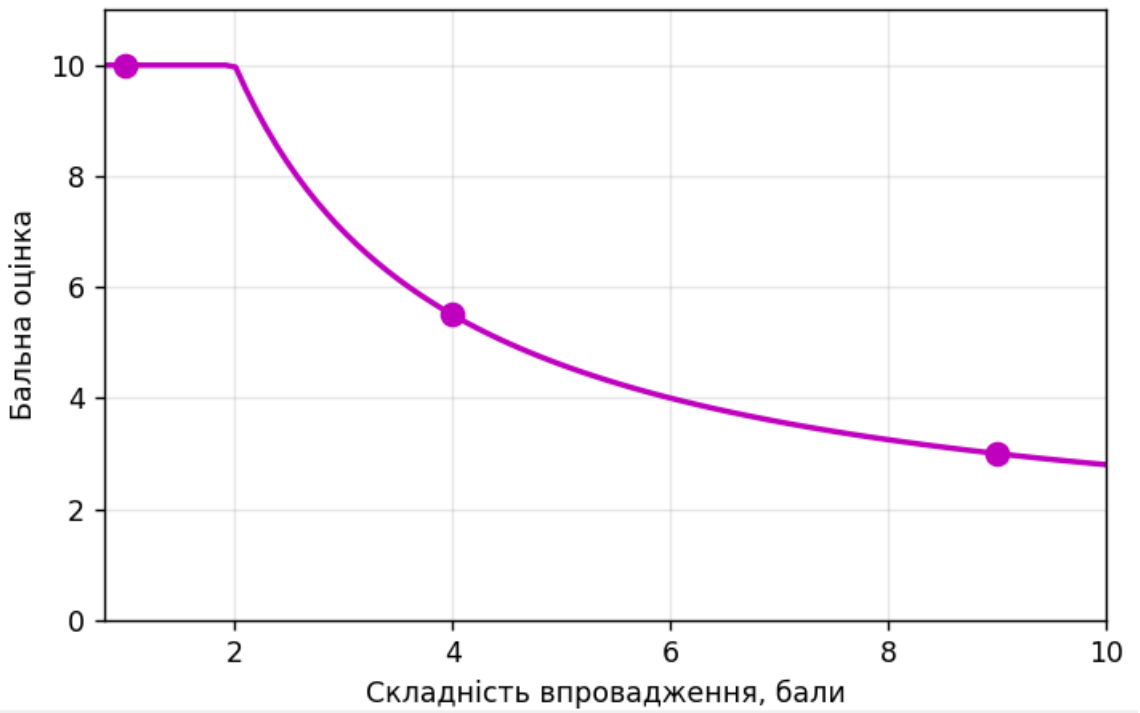


Рисунок 4.4

## 4.4 Аналіз експертного оцінювання параметрів

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 осіб.

Таблиця 4.4 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Експерти							Сума рангів $R_i$	Відхилення $\Delta_i$	$\Delta_i^2$
			1	2	3	4	5	6	7			
$X_1$	Швидкість обробки	відгуки/хв	3	3	2	3	2	3	3	19	1,5	2,25
$X_2$	Точність аналізу	%	1	1	1	1	1	1	1	7	-10,5	110,25
$X_3$	Час розробки	місяці	4	4	4	4	4	4	4	28	10,5	110,25
$X_4$	Складність впровадження	бали	2	2	3	2	3	2	2	16	-1,5	2,25
Рангом			10	10	10	10	10	10	10	70	0	225



Таблиця 4.6 – Розрахунок вагомості параметрів

Параметри	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	b <sub>1</sub>	K <sub>B1</sub>	b <sub>2</sub>	K <sub>B2</sub>	b <sub>3</sub>	K <sub>B3</sub>
X <sub>1</sub>	1	1,5	0,5	1,5	4,5	0,25	19,25	0,25	142	0,25
X <sub>2</sub>	0,5	1	0,5	0,5	2,5	0,14	12,25	0,16	100	0,16
X <sub>3</sub>	1,5	1,5	1	1,5	5,5	0,31	25,25	0,33	155	0,33
X <sub>4</sub>	0,5	1,5	0,5	1	3,5	0,19	19,75	0,26	129	0,26
Всього:					16	1	76,5	1	526	1

Вагомість кожного параметра K<sub>B<sub>i</sub></sub> розраховують за формулами (4.6-4.7):

$$K_{B_i} = b_i / \sum b_i \quad (4.6)$$

$$b_i = \sum x_{ij} \quad (4.7)$$

Відносні оцінки вагомості розраховують декілька раз, доки наступне значення буде незначно відхилитися від попереднього (менше ніж на 5%). На другій і наступних ітераціях значення коефіцієнта вагомості розраховують як:

$$K'_{B_i} = b'_i / \sum b'_i \quad (4.8)$$

$$b'_i = x_{i1}b'_{1'} + x_{i2}b'_{2'} + \dots + x_{in}b'_{n'} \quad (4.9)$$

За результатами третьої ітерації отримуємо остаточні коефіцієнти вагомості:

- 1) X<sub>1</sub> (швидкість обробки): K<sub>B1</sub> = 0,25;
- 2) X<sub>2</sub> (точність аналізу): K<sub>B2</sub> = 0,16 ;
- 3) X<sub>3</sub> (час розробки): K<sub>B3</sub> = 0,33 ;
- 4) X<sub>4</sub> (складність впровадження): K<sub>B4</sub> = 0,26.

#### 4.5 Аналіз рівня якості варіантів реалізації функцій

Таблиця 4.7 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації	Параметри	Абсолютне значення	Бальна оцінка	Коефіцієнт вагомості	Коефіцієнт рівня якості
F <sub>1</sub>	а	X <sub>1</sub>	150	7,0	0,25	1,75
		X <sub>3</sub>	4	5,5	0,33	1,82
	б	X <sub>1</sub>	80	2,8	0,25	0,70
		X <sub>3</sub>	8	3,25	0,33	1,07
F <sub>2</sub>	а	X <sub>2</sub>	87	6,4	0,16	1,02
		X <sub>2</sub>	93	9,1	0,16	1,46
F <sub>3</sub>	а	X <sub>4</sub>	2	10,0	0,26	2,60
		X <sub>4</sub>	5	4,6	0,26	1,20

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховують за формулою (4.10):

$$K_{\text{тр}}[F_{ik}] = \sum (K_{Vi} \times V_i) \quad (4.10)$$

де  $K_{Vi}$  – коефіцієнт вагомості і-го параметра;

$V_i$  – оцінка і-го параметра в балах.

Показник рівня якості к-го варіанта реалізації основних функцій виробу розраховують за формулою (4.11):

$$K_k = K_{\text{тр}}[F_{1k}] + K_{\text{тр}}[F_{2k}] + K_{\text{тр}}[F_{3k}] \quad (4.11)$$

За формулою (4.11) визначаємо рівень якості варіантів:

Варіант 1 (F<sub>1a</sub> + F<sub>2a</sub> + F<sub>3a</sub>):  $K_{k1} = 1,75 + 1,82 + 1,02 + 2,60 = 7,19$

Варіант 2 (F<sub>1a</sub> + F<sub>2б</sub> + F<sub>3a</sub>):  $K_{k2} = 1,75 + 1,82 + 1,46 + 2,60 = 7,63$

Варіант 3 ( $F_{1б} + F_{2б} + F_{3б}$ ):  $K_{кз} = 0,70 + 1,07 + 1,46 + 1,20 = 4,43$

#### 4.5 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Проект включає в себе три основні завдання:

- 1) розробка алгоритмів обробки природної мови;
- 2) створення системи сентимент та атрибутного аналізу;
- 3) розробка користувацького інтерфейсу та системи візуалізації.

Завдання 1 за ступенем новизни належить до групи А, завдання 2 – до групи Б, завдання 3 – до групи В. За складністю алгоритми завдання 1 належать до групи 2; завдання 2 – до групи 1; завдання 3 – до групи 3.

Загальну трудомісткість розроблюваного ПП для кожного з варіантів обчислюють за формулою (4.12):

$$T_0 = T_p \times K_p \times K_{ск} \times K_m \times K_{ст} \times K_{ст.м} \quad (4.12)$$

де  $T_p$  – укрупнена норма часу на розробку ПП;

$K_p$  – поправковий коефіцієнт, що враховує ступінь новизни;

$K_{ск}$  – поправковий коефіцієнт складності;

$K_m$  – коефіцієнт мови програмування;

$K_{ст}$  – коефіцієнт стандартних модулів;

$K_{ст.м}$  – коефіцієнт стандартного математичного забезпечення.

Для першого завдання:  $T_p = 42$  людино-днів,  $K_p = 1,6$ ,  $K_{ск} = 1$ ,  $K_{ст} = 0,85$ .  $T_1 = 42 \times 1,6 \times 0,85 = 57,12$  людино-днів.

Для другого завдання:  $T_p = 35$  людино-днів,  $K_p = 1,2$ ,  $K_{ск} = 1$ ,  $K_{ст} = 0,9$ .  $T_2 = 35 \times 1,2 \times 0,9 = 37,8$  людино-днів.

Для третього завдання:  $T_p = 28$  людино-днів,  $K_p = 0,8$ ,  $K_{ск} = 1$ ,  $K_{ст} = 1,0$ .  $T_3 = 28 \times 0,8 \times 1,0 = 22,4$  людино-днів.

Трудомісткість за варіантами:

- 1) варіант 1:  $T_1 = (57,12 + 37,8 + 22,4) \times 8 = 939,36$  людино-годин;
- 2) варіант 2:  $T_2 = (57,12 + 52,0 + 22,4) \times 8 = 1052,16$  людино-годин;
- 3) варіант 3:  $T_3 = (85,0 + 52,0 + 42,0) \times 8 = 1432$  людино-годин.

*Розрахунок собівартості однієї машино-години*

Для розробки програмного продукту використовується персональний комп'ютер вартістю 25000 грн з потужністю 0,5 кВт.

1. Заробітна плата обслуговуючого персоналу за формулою (4.13):

$$Z_{об} = 12 \times M_o \times K_z \times K_d \times K_{нач} \quad (4.13)$$

- де  $M_o = 18000$  грн (оклад системного адміністратора);  
 $K_z = 0,15$  (коефіцієнт зайнятості на обслуговуванні ПК);  
 $K_d = 1,25$  (коефіцієнт додаткової заробітної плати);  
 $K_{нач} = 1,22$  (коефіцієнт нарахувань).

$$Z_{об} = 12 \times 18000 \times 0,15 \times 1,25 \times 1,22 = 49410 \text{ грн.}$$

2. Амортизаційні відрахування за формулою (4.14):

$$Z_a = K_{т.м} \times N_a \times C_{пр} \quad (4.14)$$

- де  $K_{т.м} = 1,1$  (коефіцієнт транспортно-монтажних витрат);  
 $N_a = 0,25$  (норма амортизації 25%);  
 $C_{пр} = 25000$  грн (ціна ПК).

$$Z_a = 1,1 \times 0,25 \times 25000 = 6875 \text{ грн.}$$

3. Витрати на електроенергію за формулою (4.15):

$$Z_{ел} = N_c \times T_{еф} \times C_{ен} \quad (4.15)$$

- де  $N_c = 0,5$  кВт (споживча потужність ПК);  
 $T_{еф} = 1598$  год (ефективний фонд часу за рік);  
 $C_{ен} = 9,43$  грн/кВт·год (тариф на електроенергію).

$$Z_{ел} = 0,5 \times 1598 \times 9,43 = 7539,07 \text{ грн.}$$

4. Витрати на поточний ремонт і профілактику за формулою (4.16):

$$Z_p = K_{т.м} \times C_{пр} \times K_p \quad (4.16)$$

- де  $K_p = 0,04$  (4% від вартості обладнання).

$$Z_p = 1,1 \times 25000 \times 0,04 = 1100 \text{ грн.}$$

5. Накладні витрати:  $Z_n = Z_{об} \times 0,67 = 49410 \times 0,67 = 33104,70$  грн.

6. Ефективний годинний фонд часу ПК за рік за формулою (4.17):

$$T_{эф} = (D_k - D_v - D_c - D_r) \times t_{зм} \times K_v \quad (4.17)$$

де  $D_k = 365$  днів (календарні дні);

$D_v = 104$  дні (вихідні);

$D_c = 14$  днів (святкові);

$D_r = 12$  днів (планові ремонти);

$t_{зм} = 8$  год (робочий день);

$K_v = 0,85$  (коефіцієнт використання).

$$T_{эф} = (365 - 104 - 14 - 12) \times 8 \times 0,85 = 1598 \text{ год.}$$

7. Річні експлуатаційні витрати:

$$C_{екс} = Z_{об} + Z_a + Z_{ел} + Z_r + Z_n \quad (4.18)$$

$$C_{екс} = 49410 + 6875 + 7539,07 + 1100 + 33104,70 = 98028,77 \text{ грн.}$$

8. Собівартість однієї машино-години за формулою (4.19):

$$C_{м-г} = C_{екс} / T_{эф} \quad (4.19)$$

$$C_{м-г} = 98028,77 / 1598 = 61,36 \text{ грн/год.}$$

У розробці беруть участь:

1) провідний інженер-програміст (місячний оклад 35000 грн, погодинна ставка 208,33 грн);

2) старший програміст (місячний оклад 28000 грн, погодинна ставка 166,67 грн);

3) програміст (місячний оклад 22000 грн, погодинна ставка 130,95 грн);

4) системний аналітик (місячний оклад 30000 грн, погодинна ставка 178,57 грн).

Середня погодинна ставка:  $(208,33 + 166,67 + 130,95 + 178,57) / 4 = 171,13$  грн/год

Заробітна плата розробників за варіантами за формулою (4.20):

$$C_{ЗП} = T \times C_{г} \times K_d \quad (4.20)$$

де  $T$  – трудомісткість, людино-годин;

$C_T$  – погодинна ставка;

$K_D = 1,2$  (коефіцієнт додаткової зарплати).

Варіант 1:  $CЗП_1 = 939,36 \times 171,13 \times 1,2 = 192850,56$  грн

Варіант 2:  $CЗП_2 = 1052,16 \times 171,13 \times 1,2 = 215868,82$  грн

Варіант 3:  $CЗП_3 = 1432 \times 171,13 \times 1,2 = 294458,11$  грн

Відрахування на єдиний соціальний внесок (22%) за формулою (4.21):

$$СВІД = CЗП \times 0,22 \quad (4.21)$$

Варіант 1:  $СВІД_1 = 192850,56 \times 0,22 = 42427,12$  грн

Варіант 2:  $СВІД_2 = 215868,82 \times 0,22 = 47491,14$  грн

Варіант 3:  $СВІД_3 = 294458,11 \times 0,22 = 64780,78$  грн

Витрати на машинний час за формулою (4.22):

$$СМ = T \times C_{M-Г} \quad (4.22)$$

Варіант 1:  $СМ_1 = 939,36 \times 61,36 = 57639,85$  грн

Варіант 2:  $СМ_2 = 1052,16 \times 61,36 = 64558,58$  грн

Варіант 3:  $СМ_3 = 1432 \times 61,36 = 87868,85$  грн

Накладні витрати (67% від заробітної плати) за формулою (4.23):

$$СН = CЗП \times 0,67 \quad (4.23)$$

Варіант 1:  $СН_1 = 192850,56 \times 0,67 = 129209,88$  грн

Варіант 2:  $СН_2 = 215868,82 \times 0,67 = 144632,11$  грн

Варіант 3:  $СН_3 = 294458,11 \times 0,67 = 197286,93$  грн

Загальна вартість розробки ПП за варіантами за формулою (4.24):

$$СПП = CЗП + СВІД + СМ + СН \quad (4.24)$$

Варіант 1:  $СПП_1 = 192850,56 + 42427,12 + 57639,85 + 129209,88 = 422127,41$  грн

Варіант 2:  $СПП_2 = 215868,82 + 47491,14 + 64558,58 + 144632,11 = 472550,65$  грн

Варіант 3:  $СПП_3 = 294458,11 + 64780,78 + 87868,85 + 197286,93 = 644394,67$  грн

Розрахунок коефіцієнта техніко-економічного рівня за формулою (4.25):

$$КТЕР_j = K_{kj} / СПП \quad (4.25)$$

$$КТЕР_1 = 7,19 / 422127,41 = 1,70 \times 10^{-5}$$

$$КТЕР_2 = 7,63 / 472550,65 = 1,61 \times 10^{-5}$$

$$КТЕР_3 = 4,43 / 644394,67 = 0,69 \times 10^{-5}$$

Таблиця 4.8 – Зведена таблиця техніко-економічних показників варіантів

Показник	Варіант 1	Варіант 2	Варіант 3
Коефіцієнт якості $K_k$	7,19	7,63	4,43
Вартість розробки, грн	422127,41	472550,65	644394,67
Коефіцієнт техніко-економічного рівня $КТЕР \times 10^{-5}$	1,70	1,61	0,69
Ранг за ефективністю	1	2	3

#### 4.6 Висновки до 4 розділу

У результаті виконання функціонально-вартісного аналізу програмного комплексу для аналізу відгуків про автомобільні шини було визначено найбільш ефективний варіант реалізації.

За результатами розрахунків найкращим є Варіант 1 ( $F_{1a} + F_{2a} + F_{3a}$ ) з коефіцієнтом техніко-економічного рівня  $КТЕР_1 = 1,70 \times 10^{-5}$  та вартістю розробки 422127,41 грн.

Обраний варіант реалізації передбачає:

- 1) використання Python з бібліотеками NLTK/TextBlob/pymorphy2 для технологічної платформи розробки;
- 2) застосування гібридного підходу (лексиконний + статистичний) до реалізації алгоритмів NLP;

3) створення настільного додатку з використанням tkinter для користувацького інтерфейсу.

Функціонально-вартісний аналіз показав, що даний варіант забезпечує найкраще співвідношення між технічними характеристиками системи та економічними витратами на її розробку. Коефіцієнт якості варіанта 1 становить 7,19, що є достатньо високим показником при найнижчій вартості реалізації серед розглянутих альтернатив.

Експертне оцінювання параметрів програмного продукту підтвердило правильність вибору пріоритетів: найвища вагомість належить часу розробки системи ( $K_{в3} = 0,33$ ) та складності впровадження ( $K_{в4} = 0,26$ ), що відповідає вимогам швидкого впровадження та ефективної роботи системи.

Результати аналізу свідчать про те, що обраний варіант реалізації оптимально враховує специфіку обробки української мови, потреби автомобільної індустрії щодо аналізу споживчих відгуків та економічні обмеження проекту. Система забезпечить необхідну функціональність при мінімальних витратах на розробку та впровадження.

Загальна економія коштів при виборі варіанта 1 порівняно з варіантом 2 становить 50423,24 грн (10,7%), а порівняно з варіантом 3 – 222267,26 грн (34,5%). Це підтверджує економічну доцільність обраного технічного рішення та його відповідність поставленим техніко-економічним вимогам.

## ВИСНОВКИ

У процесі виконання дипломної роботи було здійснено комплексне дослідження та практичну реалізацію програмного комплексу для аналізу відгуків про автомобільні шини з використанням методів обробки природної мови та сентимент-аналізу. Робота охопила весь цикл розробки програмного забезпечення від теоретичного обґрунтування до практичної реалізації функціональної системи, що здатна ефективно обробляти великі обсяги текстової інформації українською мовою.

У першому розділі проведено системний аналіз сучасних підходів до обробки текстової інформації та сентимент-аналізу. Досліджено специфіку роботи з українською мовою в контексті автоматичної обробки тексту, виявлено особливості морфології та синтаксису, що впливають на якість аналізу. Проаналізовано існуючі рішення для аналізу споживчих відгуків та визначено їхні обмеження, що обґрунтовує необхідність створення спеціалізованого інструменту для автомобільної індустрії. Особливу увагу приділено методам атрибутного аналізу, що дозволяють виявляти згадування конкретних характеристик продукції в неструктурованих текстах.

Другий розділ присвячено розробці концептуальних основ програмного комплексу. Сформульовано функціональні та нефункціональні вимоги до системи, що включають підтримку завантаження різних форматів даних, проведення багаторівневого сентимент-аналізу, реалізацію атрибутного аналізу характеристик шин та забезпечення інтерактивної візуалізації результатів. Розроблено архітектуру системи на основі процедурного підходу з інкапсуляцією функціональності в єдиному класі, що забезпечує простоту розробки та підтримки. Обґрунтовано вибір технологічного стеку з використанням мови Python та спеціалізованих бібліотек для обробки даних, аналізу природної мови та візуалізації результатів.

Третій розділ містить детальний опис практичної реалізації програмного комплексу. Створено функціональну систему з графічним інтерфейсом користувача, що підтримує повний цикл аналізу відгуків від завантаження даних до генерації комплексних звітів. Розроблено оригінальні алгоритми sentiment-аналізу, адаптовані до специфіки української мови, з урахуванням контекстних модифікаторів, заперечень та інтенсифікаторів. Реалізовано систему атрибутивного аналізу з комплексним словником характеристик автомобільних шин, що включає понад сорок категорій атрибутів, згрупованих за тематичними областями.

У четвертому розділі проведено функціонально-вартісний аналіз програмного продукту, що дозволив визначити оптимальний варіант технічної реалізації системи. За результатами економічного обґрунтування встановлено, що найефективнішим є варіант з використанням Python та бібліотек NLTK/TextBlob/pymorphy2, гібридного підходу до sentiment-аналізу та настільного додатку на основі tkinter. Розрахований коефіцієнт техніко-економічного рівня становить  $1,70 \times 10^{-5}$  при вартості розробки 422 127,41 грн, що забезпечує економію коштів до 34,5% порівняно з альтернативними рішеннями. Експертне оцінювання параметрів підтвердило правильність встановлених пріоритетів розробки з найвищою вагомістю часу розробки (0,33) та складності впровадження (0,26).

Розроблений гібридний алгоритм sentiment-аналізу для української мови поєднує базові можливості міжнародних інструментів з власною системою контекстного аналізу, враховуючи специфіку морфології та синтаксису української мови. Створено спеціалізований словник атрибутів автомобільних шин, що містить систематизовану класифікацію характеристик продукції за функціональними групами, включаючи зчеплення, зносостійкість, комфорт, керованість та економічність. Реалізовано комплексну систему візуалізації результатів з підтримкою інтерактивних хмар слів, статистичних діаграм та порівняльних графіків з можливістю кольорового кодування за sentiment-оцінками.

Практична значущість результатів полягає в можливості використання розробленого програмного комплексу виробниками автомобільних шин для моніторингу споживчої думки про якість продукції, виявлення проблемних аспектів характеристик шин, порівняльного аналізу з продукцією конкурентів, формування стратегій покращення продукції на основі потреб споживачів та оптимізації маркетингових комунікацій з урахуванням найбільш важливих для клієнтів характеристик.

Система демонструє високу ефективність у роботі з великими обсягами текстових даних та забезпечує точний аналіз тональності відгуків з урахуванням контекстних особливостей української мови. Багаторівнева система фільтрації даних з підтримкою як стандартних критеріїв відбору, так і спеціалізованих атрибутивних фільтрів дозволяє проводити детальний аналіз характеристик продукції. Створена система експорту результатів у вигляді структурованих звітів з множинними аркушами містить різні типи аналітичної інформації для подальшого використання в бізнес-процесах.

Перспективи подальших досліджень включають розширення словника атрибутів для інших категорій автомобільної продукції, інтеграцію з соціальними мережами для автоматичного збору відгуків, розробку модулів прогнозування трендів споживчих уподобань, адаптацію алгоритмів для інших мов східнослов'янської групи та впровадження технологій машинного навчання для автоматичного поповнення словників атрибутів.

Виконана робота демонструє успішне поєднання теоретичних знань у галузі обробки природної мови з практичними потребами автомобільної індустрії, створюючи ефективний інструмент для аналізу споживчої думки та підтримки прийняття бізнес-рішень на основі даних. Розроблений програмний комплекс може бути адаптований для аналізу відгуків про інші види продукції або послуг завдяки модульній архітектурі та гнучким налаштуванням алгоритмів обробки тексту, що підтверджує універсальність запропонованого підходу та його потенціал для широкого практичного застосування.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Liu B. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*. 2012. Vol. 5, No. 1. P. 1—167.
2. Pang B., Lee L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*. 2008. Vol. 2, No. 1—2. P. 1—135.
3. Ravi K., Ravi V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-based systems*. 2015. Vol. 89. P. 14—46.
4. Hu M., Liu B. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004. P. 168—177.
5. Bird S., Klein E., Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. Beijing: O'Reilly Media, 2009. 504 p.
6. Greenberg J. H. A quantitative approach to the morphological typology of language. *International journal of American linguistics*. 1960. Vol. 26, No. 3. P. 178—194.
7. Pugh S. M., Press I. Ukrainian: a comprehensive grammar. London: Routledge, 1999. 584 p.
8. Koehn P. Statistical machine translation. Cambridge: Cambridge University Press, 2010. 433 p.
9. Bentz C. et al. The Entropy of Words—Learnability and Expressivity across More than 1000 Languages. *Entropy*. 2017. Vol. 19, No. 6. P. 275. DOI: 10.3390/e19060275.
10. Kotsyba N., Moskalevskyi M., Romanenko A. Gold standard universal dependencies corpus for Ukrainian. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. 2018. Vol. 1. P. 100—108.
11. Wiegand M. et al. A survey on the role of negation in sentiment analysis. *Proceedings of the workshop on negation and speculation in natural language processing*. 2010. Vol. 1. P. 60—68.

12. Polanyi L., Zaenen A. Contextual valence shifters. Computing attitude and affect in text: Theory and applications. 2006. P. 1—10.
13. Socher R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 conference on empirical methods in natural language processing: Abstracts of XXXVII conference*. (Seattle, WA, USA, 18 – 21 Oct. 2013). Seattle: Association for Computational Linguistics, 2013. P. 1631—1642.
14. Chollet F. Deep learning with Python. 2nd ed. Shelter Island: Manning Publications, 2021. 504 p.
15. Qiu G., Liu B., Bu J., Chen C. Opinion word expansion and target extraction through double propagation. *Computational linguistics*. 2011. Vol. 37, No. 1. P. 9—27.
16. Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation. *Journal of machine Learning research*. 2003. Vol. 3. P. 993—1022.
17. Titov I., McDonald R. Modeling online reviews with multi-grain topic models. *Proceedings of the 17th international conference on World Wide Web*. 2008. Vol. 1. P. 111—120.
18. Pontiki M. et al. SemEval-2014 Task 4: Aspect based sentiment analysis. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. 2014. Vol. 1. P. 27—35.
19. Русанівський В. М. Українська грамати́ка: морфоло́гія. К.: Наукова думка, 1993. 752 с.
20. Дарімпл М., Ніколаєва І., Недялков В. П. Взаємні конструкції / за ред. М. Дарімпл. Амстердам: John Benjamins Publishing, 2007. 250 p.
21. McKinney W. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. 2nd ed. Beijing: O'Reilly Media, 2017. 550 p.
22. Alemerien K., Magel K. GUIEvaluator: A Metric-tool for Evaluating the Complexity of Graphical User Interfaces. *The 26th International Conference on Software Engineering and Knowledge Engineering: Abstracts of XXXVI*

- conference. (Vancouver, BC, Canada, 1 – 3 July 2014). Vancouver: Knowledge Systems Institute, 2014. P. 16.
23. Grinberg M. Flask web development: developing web applications with Python. 2nd ed. Beijing: O'Reilly Media, 2018. 316 p.
24. Powers D. M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*. 2011. Vol. 2, No. 1. P. 37—63.

## ДОДАТОК А

### 1. Ініціалізація класу та налаштування

```
class EnhancedTireAnalysisApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Розширений аналізатор відгуків про вантажні шини")
        self.root.geometry("1400x900")

        # Змінні для зберігання даних
        self.df = None
        self.filtered_df = None
        self.analyzed_df = None

        # Налаштування інтерфейсу та завантаження даних
        self.setup_styles()
        self.setup_stopwords()
        self.create_ui()
```

### 2. Словники для аналізу атрибутів шин

```
TIRE_ATTRIBUTES = {
    'зчеплення_сухий_асфальт': ['зчеплення сух', 'зчепл асфальт', 'тяга на сухому'],
    'зчеплення_волога_дорога': ['зчеплення волог', 'зчепл мокр', 'аквапланування'],
    'зносостійкість_загальна': ['знос', 'довговічн', 'ресурс', 'термін служб'],
    'шум_рівень': ['шум', 'гучн', 'гуркіт', 'децибел'],
    'ціна_якість': ['ціна якість', 'співвідношення цін', 'за свої гроші'],
    # ... інші категорії атрибутів
}
```

```
POSITIVE_WORDS = ['добре', 'добрий', 'чудовий', 'відмінний', 'хороший',
                  'надійний', 'якісний', 'рекомендую', 'задоволений']
```

```
NEGATIVE_WORDS = ['погано', 'поганий', 'жахливий', 'розчарований',
                  'проблема', 'недолік', 'не рекомендую']
```

### 3. Алгоритм сентимент-аналізу з контекстом

```
def analyze_sentiment_with_context(self, text):
    """Аналіз сентименту з урахуванням контексту та модифікаторів"""
    if not isinstance(text, str) or not text.strip():
        return 0

    # Базовий аналіз через TextBlob
    blob_sentiment = TextBlob(text).sentiment.polarity * 0.2

    # Аналіз слів з урахуванням контексту
    words = self.custom_word_tokenize(text.lower())
    sentiment_score = 0
    negated = False
    intensified = False
```

```

for i, word in enumerate(words):
    # Перевірка заперечень
    if any(neg in word for neg in NEGATIONS):
        negated = True
        continue

    # Перевірка інтенсифікаторів
    if any(intens in word for intens in INTENSIFIERS):
        intensified = True
        continue

    # Підрахунок позитивних слів
    if any(pos in word for pos in POSITIVE_WORDS):
        modifier = -1 if negated else 1
        if intensified:
            modifier *= 1.5
        sentiment_score += modifier

    # Підрахунок негативних слів
    elif any(neg in word for neg in NEGATIVE_WORDS):
        modifier = 1 if negated else -1
        if intensified:
            modifier *= 1.5
        sentiment_score += modifier

    # Нормалізація та комбінування результатів
    normalized_score = sentiment_score / (abs(sentiment_score) + 3) if sentiment_score != 0
else 0
    combined_sentiment = blob_sentiment + normalized_score * 0.8

    return max(min(combined_sentiment, 1), -1)

```

4. Алгоритм виявлення атрибутів

```

def extract_attributes(self, text):
    """Виявлення атрибутів шин у тексті з контекстним аналізом"""
    attributes_found = {}

    if not isinstance(text, str) or not text.strip():
        return attributes_found

    text_lower = text.lower()
    sentences = re.split(r'[!?!]', text_lower)

    # Пошук кожного атрибуту в реченнях
    for attribute, keywords in TIRE_ATTRIBUTES.items():
        attribute_mentions = []
        attribute_sentiments = []

        for sentence in sentences:
            if any(keyword in sentence for keyword in keywords):
                # Виділення контексту навколо ключових слів

```

```

contexts = self.extract_context(sentence, keywords, 5)

for context in contexts:
    context_sentiment = self.analyze_sentiment_with_context(context)
    attribute_mentions.append(context)
    attribute_sentiments.append(context_sentiment)

# Збереження результатів для атрибуту
if attribute_mentions:
    avg_sentiment = sum(attribute_sentiments) / len(attribute_sentiments)
    attributes_found[attribute] = {
        'sentiment': avg_sentiment,
        'weight': len(attribute_mentions),
        'mentions': attribute_mentions[:3]
    }

return attributes_found

```

5. Створення графічного інтерфейсу

```

def create_ui(self):
    """Створення основного інтерфейсу програми"""
    # Головна панель
    main_frame = ttk.Frame(self.root)
    main_frame.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)

    # Верхня панель з кнопками
    top_frame = ttk.Frame(main_frame)
    top_frame.pack(fill=tk.X, pady=5)

    # Кнопки завантаження та експорту
    load_button = ttk.Button(top_frame, text="Завантажити файл",
                             command=self.select_file)
    load_button.pack(side=tk.RIGHT, padx=5)

    export_button = ttk.Button(top_frame, text="Експортувати результати",
                               command=self.export_results)
    export_button.pack(side=tk.RIGHT, padx=5)

    # Панель з вкладками для візуалізації
    self.notebook = ttk.Notebook(main_frame)
    self.notebook.pack(fill=tk.BOTH, expand=True)

    # Створення вкладок
    self.create_wordcloud_tab()
    self.create_ratings_tab()
    self.create_brands_tab()
    self.create_sentiment_tab()
    self.create_attributes_tab()

```

6. Генерація хмари слів

```

def generate_wordcloud(self):
    """Створення хмари слів з урахуванням сентименту"""
    if self.filtered_df is None or len(self.filtered_df) == 0:

```

```

return

# Обробка коментарів
all_words = []
word_sentiment_dict = {}

for _, row in self.filtered_df.iterrows():
    comment = str(row['Коментар'])
    words = self.custom_word_tokenize(comment.lower())

    for word in words:
        if word not in self.all_stopwords and len(word) > 2:
            all_words.append(word)

            # Відстеження настрою слів
            if 'Загальний_настрій' in row:
                sentiment = float(row['Загальний_настрій'])
                norm_sentiment = (sentiment + 1) / 2
                word_sentiment_dict[word] = word_sentiment_dict.get(word, 0) +
norm_sentiment

# Створення хмари слів
self.word_freq = Counter(all_words)

# Функція забарвлення за настроєм
def color_func(word, **kwargs):
    if word in word_sentiment_dict:
        sentiment_score = word_sentiment_dict[word] / self.word_freq[word]
        r = int(255 * (1 - sentiment_score))
        g = int(255 * sentiment_score)
        return f"rgb({r}, {g}, 0)"
    return "rgb(128, 128, 128)"

wordcloud = WordCloud(width=800, height=600, background_color='white',
                      color_func=color_func, max_words=100)
wordcloud.generate_from_frequencies(self.word_freq)

# Відображення в інтерфейсі
self.ax_wordcloud.clear()
self.ax_wordcloud.imshow(wordcloud, interpolation='bilinear')
self.ax_wordcloud.axis('off')

```

### 7. Система фільтрації даних

```

def apply_filters(self):
    """Застосування фільтрів до даних"""
    if self.df is None:
        return

    self.filtered_df = self.df.copy()

    for filter_key, filter_info in self.filter_vars.items():
        filter_type = filter_info.get("type")

```

```

if filter_type == "range":
    # Числові фільтри
    min_val = filter_info["min"].get()
    max_val = filter_info["max"].get()
    self.filtered_df = self.filtered_df[
        (self.filtered_df[filter_key] >= min_val) &
        (self.filtered_df[filter_key] <= max_val)
    ]

elif filter_type == "check":
    # Категоріальні фільтри
    selected_values = [val for val, var in filter_info["values"].items()
                       if var.get()]
    if selected_values:
        self.filtered_df = self.filtered_df[
            self.filtered_df[filter_key].isin(selected_values)
        ]

elif filter_type == "attribute":
    # Атрибутні фільтри
    if filter_info["enabled"].get():
        column = filter_info["column"]
        min_sentiment = filter_info["min_sentiment"].get()
        mask = self.filtered_df[column].apply(
            lambda x: not pd.isna(x) and x >= min_sentiment
        )
        self.filtered_df = self.filtered_df[mask]

# Оновлення візуалізацій
self.update_all_visualizations()

```

8. Експорт результатів у Excel

```

def export_results(self):
    """Експорт результатів аналізу в Excel файл"""
    if self.filtered_df is None or len(self.filtered_df) == 0:
        return

    output_file = filedialog.asksaveasfilename(
        defaultextension=".xlsx",
        filetypes=[("Excel files", "*.xlsx"), ("All files", "*.*")]
    )

    if output_file:
        with pd.ExcelWriter(output_file, engine='openpyxl') as writer:
            # Основні дані
            self.filtered_df.to_excel(writer, sheet_name='Відфільтровані дані', index=False)

            # Статистика по брендах
            if 'Марка_шини' in self.filtered_df.columns:
                brand_stats = self.filtered_df.groupby('Марка_шини')['Рейтинг'].agg(['count',
                    'mean'])

```

```

brand_stats.to_excel(writer, sheet_name='Статистика по брендах')

# Статистика атрибутів
attribute_columns = [col for col in self.filtered_df.columns if col.startswith('Атри-
бут_')]
if attribute_columns:
    attr_stats = []
    for col in attribute_columns:
        values = self.filtered_df[col].dropna()
        if len(values) > 0:
            attr_stats.append({
                'Атрибут': col.replace('Атрибут_', ''),
                'Кількість згадувань': len(values),
                'Середній сентимент': values.mean()
            })

    attr_df = pd.DataFrame(attr_stats)
    attr_df.to_excel(writer, sheet_name='Статистика атрибутів', index=False)

```

9. Запуск програми

```

if __name__ == "__main__":
    root = tk.Tk()
    app = EnhancedTireAnalysisApp(root)
    root.mainloop()

```

Примітка

Повний код програми доступний у репозиторії GitHub за посиланням:

[MrCrazyPant/tire-review-analysis-system](https://github.com/MrCrazyPant/tire-review-analysis-system): Система аналізу відгуків про автомобільні шини з використанням NLP

Основні бібліотеки, що використовуються:

**pandas, numpy** – обробка та аналіз даних  
**matplotlib** – створення графіків та діаграм  
**wordcloud** – генерація хмар слів  
**tkinter** – графічний інтерфейс користувача  
**nlk, textblob** – обробка природної мови  
**openpyxl** – робота з Excel файлами

## ДОДАТОК Б ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
Кафедра математичних методів системного аналізу  
Презентація дипломної роботи  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою “системний аналіз і  
управління”

### Система аналізу відгуків про товари з використанням методів обробки природної мови

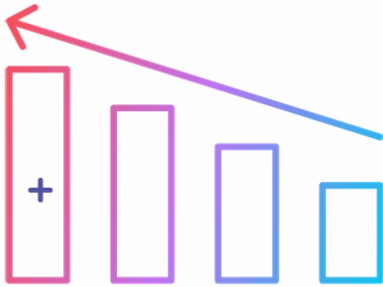
Виконав: Арестенко Георгій Сергійович  
Керівник: доц. Куєвда Ю.В.

**Об'єктом** дослідження є процеси автоматичного аналізу емоційного забарвлення текстових відгуків користувачів про товари для прийняття бізнес-рішень.

**Мета** розробити програмну систему для аналізу україномовних відгуків про товари, реалізувати комбінований сентимент-аналіз з урахуванням контексту, заперечень та модифікаторів.

**Предметом** дослідження є методи та алгоритми обробки природної мови для сентимент-аналізу та виявлення атрибутів товарів у текстових відгуках.

## Актуальність аналізу відгуків



- Автоматизація аналізу відгуків українською мовою дозволяє компаніям знижувати витрати на ручну обробку, швидше реагувати на проблеми та формувати релевантні пропозиції для цільової аудиторії.
- Більшість існуючих NLP-систем орієнтовані на англomовні дані, тоді як українські відгуки часто ігноруються через складність мови та нестачу інструментів. Розробка системи, здатної розуміти контекст, емоції та згадувані характеристики товару в українськомовному тексті — критично важливий крок для розвитку <sup>3</sup> локальних сервісів аналізу даних.

+



## Набір даних

- Через обмежений доступ до якісних реальних відгуків, для демонстрації роботи системи використано синтетично згенеровані дані.
- Коментарі створено на основі шаблонної структури реальних відгуків, з урахуванням ключових лінгвістичних і тематичних ознак.

Формат запису: ID, Марка\_шини, Модель, Рейтинг, Пробіг\_км, Тип\_вантажівки, Поверхня\_доріг, Сезон, Ціна\_грн, Дата\_відгуку, Коментар

## Методи:

- Лексиконно-контекстуальний сентимент-аналіз

Розроблено алгоритм, який поєднує словниковий підхід (списки позитивних, негативних слів, заперечень, інтенсифікаторів, димінішерів) з контекстною обробкою для визначення емоційного забарвлення коментарів.

- Атрибутний аналіз

Використано словникову модель з близько 35+ категоріями атрибутів шин (наприклад, зчеплення, шум, зносостійкість). Визначення здійснюється через ключові слова та фрази, згруповані за 8 основними темами.

- Обробка природної мови (NLP)

Застосовано власну токенизацію, видалення стоп-слів, обробку морфологічних особливостей української мови. Реалізовано власну функцію токенизації з підтримкою апострофів, змішаного українсько-англійського мовлення й чисел.

- Контекстний аналіз

Для кожного виявленого ключового слова формується контекстне вікно ( $\pm 5$  слів), у якому виконується локальний аналіз сентименту з врахуванням модифікаторів.

- Нормалізація та структурування даних

Автоматичне виявлення та приведення структури вхідних файлів до єдиного формату з обробкою різних кодувань та форматів даних.

5

## Інструменти:

- Мова програмування: Python

Обрана за гнучкість у роботі з текстовими даними та розвинену екосистему NLP.

- Бібліотеки:

- nltk, TextBlob, spaCy —

для обробки тексту, побудови моделей

- pandas, numpy —

для обробки таблиць, статистики

- matplotlib, wordcloud —

для побудови графіків і хмар слів

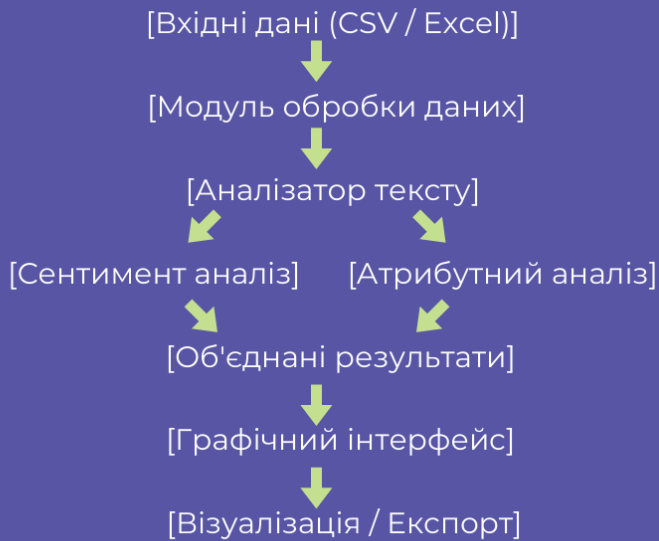
- tkinter —

створення графічного інтерфейсу користувача



6

# Архітектура системи



7

## Модуль обробки тексту

Завантаження та нормалізація

- Вхідні формати: CSV, Excel
- Автоматичне виявлення структури: колонки коментарів, рейтингів, брендів
- Нормалізація кодувань: UTF-8, CP1251, ISO-8859-1
- Створення ID: якщо відсутній
- Результат: Структурований DataFrame готовий до аналізу

### Власна система токенізації:

- Підтримка повного українського алфавіту включно з "ї", "є", "і", "ї"
- Обробка апострофів у словах типу "м'який", "п'ять", "об'єм"
- Змішане мовлення - коректна обробка українсько-англійських текстів
- Числові значення - збереження цифр у контексті (розміри, ціни, пробіг)
- Очищення пунктуації на початку та в кінці слів
- Фільтрація коротких слів (довжина > 1 символу)

Адаптивна обробка стоп-слів

- Базові українські стоп-слова: "і", "в", "на", "з", "та", "що", "не", "як"
- Додаткові стоп-слова: "але", "вже", "для", "при", "після", "навіть"
- Спеціалізовані терміни: "шини", "шин", "км", "год", "дороги" (для автомобільної тематики)

8

## Сентимент аналіз (алгоритм)

ВХІД: Токенізований текст  
["дуже", "хороші", "шини"]

Ініціалізація: sentiment\_score = 0,  
negated = False, modifier = 1.0

Приклад розрахунку:  
"дуже" → modifier = 1.8  
"хороші" → +1 \* 1.8 = +1.8  
Результат: +1.8 → нормалізація →  
+0.64

ВИХІД: Нормалізований сентимент (-  
1.0 до +1.0)

ДЛЯ КОЖНОГО СЛОВА:

- ЗАПЕРЕЧЕННЯ(?) ("не", "ні") →  
negated = True, modifier = -1
- ПОСИЛЮВАЧ(?) ("дуже",  
"надзвичайно") → modifier \*= 1.8
- ПОСЛАБЛЮВАЧ(?) ("трохи",  
"дещо") → modifier \*= 0.4
- ПОЗИТИВНЕ СЛОВО? →  
sentiment\_score += modifier \* 1
- НЕГАТИВНЕ СЛОВО? →  
sentiment\_score += modifier \* (-1)
- СКИДАННЯ МОДИФІКАТОРІВ  
після 2-3 слів


9

## Сентимент аналіз

Аналіз настроїв

Мета: 

Визначення емоційного забарвлення відгуків:  
позитивні, негативні, нейтральні

Підхід: 

Лексикон + контекст

Врахування заперечень, підсилювачів, зменшувачів

Аналіз на рівні фраз/речень

Результати: 

Сентимент від -1 до +1

Категорії: дуже негативний → дуже позитивний

Візуалізація: графіки, таблиці, хмари слів

Переваги: 

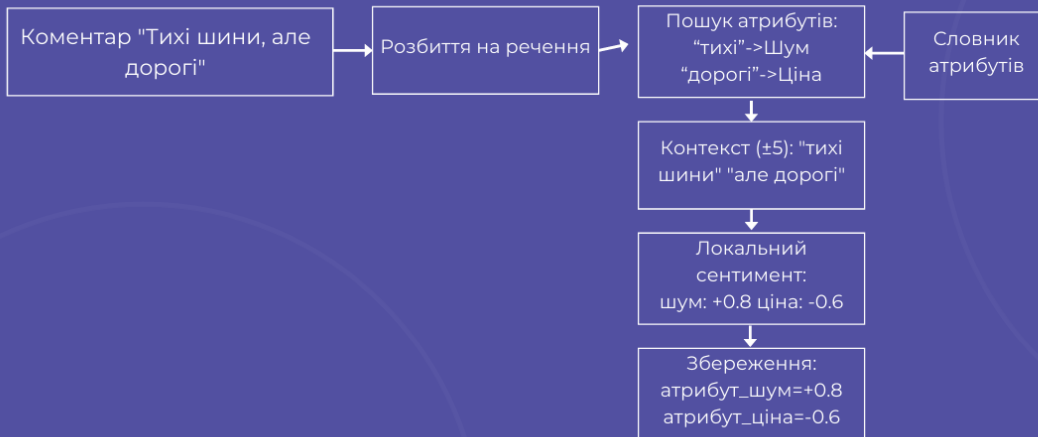
Адаптовано для української мови

Вища точність, ніж у стандартних англійських моделей

Автоматизована обробка великих обсягів

10

## Атрибутний аналіз (алгоритм)



11

## Атрибутний аналіз

### Словникова база

Атрибути згруповані за категоріями (наприклад: зчеплення, зносостійкість, шум, комфорт, ціна, бренд тощо).

### Пошук згадок

Коментар розбивається на речення → у кожному шукаються ключові слова → витягується контекст (±5 слів).

### Оцінка сентименту

Для кожного контексту обчислюється тональність: враховується заперечення (змінюють полярність на протилежну), підсилювачі (підсилюють сентимент), зменшувачі (послаблюють сентимент) Контекстне вікно обмежує вплив модифікаторів.

### Підсумкова оцінка атрибута

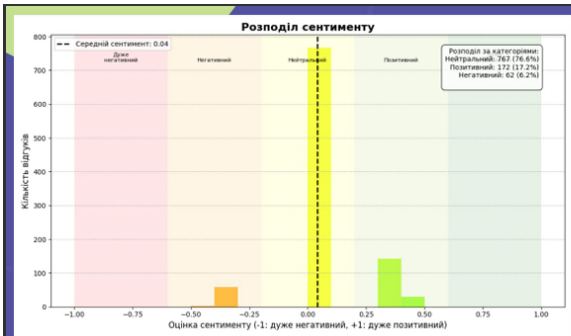
Для кожного атрибута в коментарі зберігаються:

Середній сентимент (-1 до +1)

Кількість згадок (вага)

12





## Інтерфейс користувача



15

### Тип інтерфейсу:

- Графічний десктопний застосунок, розроблений з використанням бібліотеки Tkinter.

### Основні можливості:

- Завантаження файлів з відгуками у форматах CSV або Excel
- Візуалізація хмар слів, графіків тональності, розподілу оцінок
- Фільтрація відгуків за брендами, тональністю, атрибутами
- Відображення результатів настрою-аналізу по кожному коментарю
- Перегляд ключових атрибутів товару з прикладами згадок



## Інтерфейс користувача

### Додатковий функціонал:

- Автоматична нормалізація структури даних
- Експорт результатів аналізу у зручному табличному вигляді
- Підтримка кольорового кодування за рейтингом або настроєм

### Адаптація під користувача:

- Зручне масштабування вікна
- Інтуїтивна навігація
- Підказки та повідомлення про помилки

16

## Результати:

### Опрацьовано:

- понад 800 відгуків про автомобільні шини
- тексти українською мовою різної довжини та структури

### Виявлено:

- понад 35 тематичних атрибутів (зчеплення, шум, зносостійкість, комфорт, ціна тощо)
- для кожного атрибута — середній рівень сентименту та кількість згадок
- автоматична класифікація емоцій: дуже негативні, негативні, нейтральні, позитивні, дуже позитивні

### Візуалізації:

- хмари слів для позитивних і негативних висловлювань
- графіки розподілу сентименту
- динамічні таблиці з фільтрами (бренд, атрибут, оцінка)

### Експорт та звітність:

- результати зберігаються у CSV/Excel
- готовність до використання в аналітиці та управлінських звітах

### Загальний результат:

- система успішно виконує глибокий аналіз українськомовних відгуків
- забезпечує швидку і масштабовану обробку великих обсягів даних

17

## Висновки

- Розроблено програмну систему для аналізу українськомовних відгуків про товари
- Реалізовано комбінований сентимент-аналіз з урахуванням контексту, заперечень та модифікаторів
- Впроваджено атрибутний аналіз: автоматичне виявлення ключових характеристик товарів
- Система адаптована до особливостей української мови та термінології шин
- Інтерфейс користувача забезпечує зручну візуалізацію та експорт результатів
- Отримані результати дозволяють швидко виявляти проблемні аспекти товарів і підтримують прийняття рішень

18

## Можливості для покращення

Розширення лінгвістичних можливостей:

- Додавання підтримки лематизації та стемінгу для української мови
- Розширення словників атрибутів на основі аналізу реальних відгуків
- Включення регіональних варіантів української мови та сленгу
- Покращення обробки граматичних конструкцій (заперечення, умовні речення)

Технічні оптимізації:

- Впровадження багатопотокової обробки для великих масивів даних (>10000 відгуків)
- Оптимізація алгоритмів для роботи в реальному часі
- Додавання кешування результатів для повторно аналізованих текстів
- Реалізація пакетної обробки файлів

Розширення функціональності:

- Розробка веб-версії з можливістю онлайн-аналізу
- Створення REST API для інтеграції з зовнішніми системами
- Додавання експорту в інші формати (JSON, XML, PowerBI)
- Впровадження автоматичних звітів та алертів

19

# Дякую за увагу!

---