**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ**
**ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**
**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

# КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи визначення точок привабливості на основі відгуків користувачі соціальних мереж»

на здобуття освітнього ступеня магістра
зі спеціальності 124 Системний аналіз
освітньо-професійної програми «Інтелектуальні системи управління»

*Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело*

_____ Дмитро МАРТИНЕНКО
*(підпис)*

Виконав: здобувач вищої освіти групи САДМ-61

Дмитро МАРТИНЕНКО

Керівник: Ігор ПАТРАКЄЄВ
*к.т.н., доцент*

Рецензент: _____
*науковий ступінь, вчене звання* *Ім'я, ПРІЗВИЩЕ*

**Київ 2024**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра інформаційних систем та технологій

Ступінь вищої освіти магістр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Інтелектуальні системи управління

**ЗАТВЕРДЖУЮ**
Завідувач кафедрою ІСТ
_____Каміла СТОРЧАК
«_____» ___20___р.

# ЗАВДАННЯ
# НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мартиненко Дмитро Олегович
*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Методи визначення точок привабливості на основі
1. відгуків користувачів соціальних мереж

керівник кваліфікаційної роботи Ігор ПАТРАКЕЄВ, к.т.н., доцент
затверджені наказом Державного університету інформаційно-комунікаційних технологій

від «____» _____20___р. № ____

2. Строк подання кваліфікаційної роботи «___» _____20___р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з питань збору та аналізу великих обсягів даних. Інформація про сучасні алгоритми машинного навчання та методи класифікації текстових даних. Практичні рекомендації щодо проєктування серверних та клієнтських частин систем управління даними.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   1. Попередня обробка текстових даних: очищення, нормалізація, видалення шуму та визначення релевантної інформації;
   2. Застосування алгоритмів аналізу настроїв для класифікації відгуків на позитивні, негативні та нейтральні; визначення ключових слів і тем для формування точок привабливості;
   3. Використання методів машинного навчання та кластеризації для автоматичного виявлення закономірностей у відгуках і групування схожих за змістом повідомлень
   4. Розробка механізмів оцінки якості аналізу та візуалізації результатів для демонстрації популярних точок привабливості

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «_____» _____20____р.

# КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Аналіз літературних джерел, пов'язаних з темою роботи | | |
| 2 | Розробка архітектури та створення прототипу системи | | |
| 3 | Визначення методів аналізу та обробки зібраних даних | | |
| 4 | Реалізація прототипу, інтеграція модулів системи | | |
| 5 | Оцінка ефективності запропонованого методу | | |
| 6 | Аналіз переваг і недоліків розробленого рішення | | |
| 7 | Написання висновків по роботі | | |
| 8 | Підготовка демонстраційних матеріалів | | |
| 9 | Підготовка доповіді для захисту роботи | | |

Здобувач вищої освіти _____ Дмитро МАРТИНЕНКО

*(підпис)*

Керівник
кваліфікаційної роботи _____ Ігор ПАТРАКЕЄВ

*(підпис)*

# РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 72 стор., 34 рис., 2 табл., 23 джерела.

*Мета роботи – розробити систему для збору, обробки та аналізу даних з соціальних мереж для надання персоналізованих рекомендацій користувачам на основі текстових відгуків та геолокації.*

*Об'єкт дослідження – соціальні мережі як джерело даних для аналізу популярності локацій.*

*Предмет дослідження – методи визначення точок привабливості за допомогою текстових відгуків та геолокаційної інформації.*

*Короткий зміст роботи*: У роботі аналізуються методи збору та обробки даних з соціальних мереж, зокрема текстових відгуків та геолокаційної інформації, для створення персоналізованих рекомендацій для користувачів. Застосовуються сучасні підходи в машинному навчанні та аналізі тексту, зокрема для прогнозування популярності місць і визначення точок привабливості.

КЛЮЧОВІ СЛОВА: СОЦІАЛЬНІ МЕРЕЖІ, БІЗНЕС-СТРАТЕГІЇ ПЕРСОНАЛІЗАЦІЯ КОНТЕНТУ, ТАРГЕТИНГ, ГЕОЛОКАЦІЙНІ СЕРВІСИ АРІ ДЛЯ ЗБОРУ ДАНИХ, МАШИННЕ НАВЧАННЯ, ОБРОБКА ДАНИХ, АНАЛІЗ НАСТРОЮ ТЕКСТУ

# ABSTRACT

The textual part of the qualification work for obtaining the master's degree: 72 pages, 34 pictures, 2 tables, 23 sources.

*The aim of the study* is to develop a system for collecting, processing, and analyzing data from social networks to provide personalized recommendations to users based on textual reviews and geolocation.
*The object of the study* is social networks as a data source for analyzing the popularity of locations.
*The subject of the study* is methods for identifying points of attraction using textual reviews and geolocation information.
*Summary:*
The work analyzes methods for collecting and processing data from social networks, particularly textual reviews and geolocation information, to create personalized recommendations for users. Modern approaches in machine learning and text analysis are applied, including predicting location popularity and identifying points of attraction.



KEYWORDS: SOCIAL NETWORKS, BUSINESS STRATEGIES, CONTENT PERSONALIZATION, TARGETING, GEOLOCATION SERVICES, DATA COLLECTION API, MACHINE LEARNING, DATA PROCESSING, TEXT SENTIMENT ANALYSIS.

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

**ПОДАННЯ**
**ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ**
**ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ**
**на здобуття освітнього ступеня магістра**

Направляється здобувач *Мартиненко Д. О.* до захисту кваліфікаційної роботи

за спеціальністю *124 Системний аналіз*

освітньо-професійної програми Інтелектуальні системи управління

на тему: «Методи визначення точок привабливості на основі відгуків користувачі соціальних мереж».

Кваліфікаційна робота і рецензія додаються.

Директор ННІІТ _____ Катерина НЕСТЕРЕНКО
*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

**Висновок керівника кваліфікаційної роботи**

Здобувач Дмитро МАРТИНЕНКО продемонстрував ґрунтовні знання в обраній темі, ефективно застосував сучасні методи аналізу даних та успішно вирішив поставлені завдання. Робота вирізняється актуальністю теми, високим рівнем практичного опрацювання та значущістю отриманих результатів для подальших досліджень і впровадження в реальних умовах. Здобувач продемонстрував здатність до самостійного наукового пошуку, творчого підходу та глибокого аналізу інформації, що свідчить про високий рівень професійної підготовки.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Дмитра МАРТИНЕНКА на оцінку «_____» та присвоїти йому кваліфікацію магістр з системного аналізу.

Керівник кваліфікаційної роботи _____ Ігор ПАТРАКЄЄВ
*(підпис)*

«___»_____20___року

**Висновок кафедри про кваліфікаційну роботу**

Кваліфікаційна робота розглянута. Здобувач *Мартиненко Д. О.* допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою ІСТ _____ Каміла СТОРЧАК
*(назва)* *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

**ВІДГУК РЕЦЕНЗЕНТА**
**на кваліфікаційну роботу**
**на здобуття освітнього ступеня магістра**

здобувача вищої освіти Мартиненко Дмитро Олегович
на тему «Методи визначення точок привабливості на основі відгуків користувачі соціальних мереж»

**Актуальність.** Тема дослідження є актуальною у зв'язку зі стрімким розвитком цифрових технологій та мобільних додатків, що зумовлює зростання попиту на персоналізовані рекомендаційні системи. Використання геолокації та аналізу відгуків для надання релевантних рекомендацій є важливим завданням, особливо в умовах роботи з великими обсягами даних. Це сприяє підвищенню ефективності сервісів та забезпечує користувачів точними й своєчасними рекомендаціями, що відповідають їхнім індивідуальним потребам.

**Позитивні сторони.**

1. Використання сучасних алгоритмів машинного навчання забезпечує глибокий аналіз текстів, зокрема настроїв, що дозволяє отримувати якісні результати.
2. Система інтегрує автоматизовані етапи збору, очищення та обробки даних, мінімізуючи ручну працю і знижуючи ймовірність помилок.
3. Архітектура дозволяє легко адаптувати систему до нових потреб і зростання обсягів даних, зберігаючи її продуктивність.

**Недоліки.**

1. Стабільність роботи системи значною мірою залежить від доступності сторонніх API, що може викликати перебої в роботі.
2. Низька якість вхідних даних, таких як спам або текст із помилками, може призвести до некоректних результатів аналізу.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи магістерської.

**Висновок:** *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "_____", а здобувач Дмитро МАРТИНЕНКО заслуговує присвоєння кваліфікації магістр з системного аналізу.*

Рецензент:
*науковий ступінь, вчене звання* _____ *підпис* _____ *Ім'я, ПРІЗВИЩЕ*

**ЗМІСТ**

**ВСТУП** ..... 10

**РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ** ..... 13

1.1 Визначення точок привабливості ..... 13

1.2 Огляд соціальних мереж ..... 13

1.3 Визначення соціальних мереж ..... 14

1.4 Роль соціальних мереж у створенні популярності місць ..... 15

1.5 Методи збору інформації з соціальних мереж ..... 16

1.6 Типи даних ..... 16

1.7 Методи збору даних ..... 17

1.8 Виклики та обмеження ..... 17

1.9 Способи використання даних соціальних мереж ..... 18

1.9.1 Таргетинг та персоналізація контенту в соціальних мережах ..... 18

1.9.2 Інтеграція соціальних мереж із геолокаційними сервісами ..... 19

1.9.4 Використання даних соціальних мереж у бізнес-стратегіях компаній ..... 20

**РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ ДЛЯ ВИЗНАЧЕННЯ ТОЧОК ПРИВАБЛИВОСТІ** ..... 24

2.1 Використання API для збору даних ..... 24

2.2 API Foursquare ..... 24

2.3 Типи даних ..... 25

2.4 Формат даних ..... 26

2.5 Зберігання даних ..... 28

2.6 Ключові API та їх обмеження ..... 29

2.7 Первинна очистка та фільтрація отриманих даних ..... 30

2.8 Використання моделей на основі трансформерів у розпізнаванні настрою тексту ..... 31

2.8.1 Word2Vec ..... 31

2.8.1 BERT ..... 32

2.8.3 DistilBERT ..... 35

2.9 Інструменти розробки ..... 37
2.9.1 Збір даних для навчання ..... 37
2.9.2 Обробка даних ..... 39
2.9.3 Навчання моделі класифікації та її оцінка ..... 40
2.9.4 Серверна частина ..... 41
2.9.5 Збір метрик та їх відображення ..... 41
**2.9.6 Клієнтська частина** ..... 42
2.9.7 Інструменти розробки та середовища ..... 42
**РОЗДІЛ 3. ОПИС РЕАЛІЗАЦІЇ ТА ПРОТОТИПУ** ..... 44
3.1 Архітектура системи ..... 44
3.2 Модульна структура прототипу: реалізація та інтеграція ..... 47
3.2.1 Модуль збору місць та відгуків (Tip Collector) ..... 48
3.2.2 Модуль створення вибірок (Requester) ..... 48
3.2.3 Модуль обробки даних (Data Pre-Processor) ..... 48
3.2.4 Модуль тренування моделі (ModelTrainer) ..... 49
3.2.5 Система пошуку і взаємодії з користувачем ..... 49
3.3 Процес підготовки датасетів для тренування моделі ..... 49
3.4 Використання DistilBERT для формування векторних уявлень та подальшого моделювання ..... 52
3.5 Система пошуку і взаємодії з користувачем ..... 59
3.6 Збір аналітичних даних ..... 62
3.7 Візуалізація зібраних даних ..... 64
**РОЗДІЛ 4. ПЕРЕВАГИ ТА НЕДОЛІКИ РОЗРОБЛЕНОГО МЕТОДУ** ..... 69
4.1 Переваги ..... 69
4.2 Обмеження та недоліки ..... 70
**ВИСНОВКИ** ..... 72
**ПЕРЕЛІК ПОСИЛАНЬ** ..... 73
**ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)** ..... 75

# ВСТУП

**Актуальність теми.** У сучасних умовах швидкого розвитку цифрових технологій та мобільних додатків стає надзвичайно важливим надання користувачам персоналізованих і точних рекомендацій щодо різноманітних локацій (ресторанів, клубів, кафе тощо) на основі їхніх вподобань і геолокації. Особливу актуальність це набуває для сервісів, які працюють з великими обсягами відгуків і даних, де необхідно враховувати не лише відстань до місця, але й якість та актуальність відгуків. Створення системи, яка забезпечує швидкий і точний аналіз відгуків, враховує статистику популярності місць і дозволяє користувачам отримувати найкращі рекомендації, є важливим завданням для розробників сучасних мобільних додатків і веб-сервісів.

**Об’єкт дослідження:** соціальні мережі як джерело даних для аналізу популярності локацій та створення рекомендацій.

**Предмет дослідження:** методи визначення точок привабливості на основі текстових відгуків та геолокаційної інформації, зібраної із соціальних мереж.

**Мета роботи.** Розробка ефективної системи для збору, обробки та аналізу великих обсягів даних, яка забезпечує точну оцінку популярності об'єктів на основі відгуків користувачів і їх геолокації. Система повинна надавати персоналізовані рекомендації, використовуючи сучасні методи обробки даних, машинного навчання та аналізу текстів, а також інтегрувати ці процеси в зручні інтерфейси для мобільних та веб-додатків.

Для досягнення мети в роботі поставлено та виконано такі завдання:

1. Провести аналіз предметної області: – вивчити основні аспекти соціальних мереж і їхній вплив на популярність місць; – виявити ключові характеристики даних для оцінки точок привабливості.

2. Обрати методи та технології для реалізації проекту: – вибір інструментів збору, обробки та аналізу даних; – обґрунтування використання моделей машинного навчання для аналізу текстів.

3. Реалізувати збір даних для навчання: – зібрати дані з соціальних мереж, включаючи текстові відгуки та геолокаційні метадані; – забезпечити збереження даних у зручному форматі для подальшої обробки.

4. Розробити алгоритми попередньої обробки даних: – виконати токенізацію тексту, видалення стоп-слів та нормалізацію; – використовувати моделі машинного навчання для аналізу настроїв у текстах.

5. Виконати моделювання та прогнозування: – застосувати нейронні мережі для прогнозування популярності місць; – створити рекомендаційні системи, які поєднують текстову та геолокаційну інформацію.

6. Розробити серверну частину системи: – реалізувати серверну архітектуру для обробки запитів користувачів; – забезпечити масштабованість та продуктивність системи; – підготувати можливість переходу на мікросервісну архітектуру в майбутньому.

7. Розробити користувацький інтерфейс: – створити веб- та мобільний інтерфейси для відображення аналізу і рекомендацій; – забезпечити зручність використання для користувачів.

8. Оцінити ефективність системи: – провести тестування точності рекомендацій; – оцінити продуктивність і швидкість обробки запитів.

**Практичне значення**. Розроблена система автоматично визначає точки привабливості на основі відгуків користувачів у соціальних мережах. Зокрема:

1. Збір даних: – реалізація модулів для отримання даних із соціальних мереж за допомогою API (наприклад, Foursquare, Twitter, Instagram); – збір текстових відгуків, геолокаційних даних та метаінформації (рейтинги, кількість відгуків тощо).

2. Обробка даних: – розробка алгоритмів для первинної обробки даних, включаючи очищення, нормалізацію та токенізацію текстів; – застосування моделей аналізу настроїв для визначення емоційного тону відгуків.

3. Моделювання: – використання трансформерних моделей (BERT, DistilBERT) для створення векторних представлень текстів; – навчання моделі машинного навчання для прогнозування популярності локацій.

4. Інтеграція та розробка: – розробка серверної частини для обробки запитів користувачів; – реалізація клієнтської частини у вигляді веб- або мобільного додатку, що забезпечує доступ до рейтингу місць та персоналізованих рекомендацій.

5. Тестування та оцінка: – проведення тестування системи на реальних даних; – оцінка точності моделі та швидкості обробки запитів; – аналіз ефективності рекомендацій, створених системою.

6. Візуалізація результатів: – створення зручного інтерфейсу для відображення рейтингу місць, настроїв відгуків та трендів популярності.

Практичне завдання спрямоване на створення працюючого прототипу системи, який може бути адаптований для використання у різних галузях.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Визначення точок привабливості

У цьому розділі розглядається поняття точок привабливості, їх характеристика та значення в аналізі соціальних мереж.

Точки привабливості — це місця або локації, які викликають підвищений інтерес серед користувачів. Ці локації можуть мати культурну, історичну, комерційну чи соціальну значущість. Визначення таких точок базується на аналізі даних, отриманих із соціальних мереж, таких як коментарі, лайки, позначки, чекини, хештеги, фотографії тощо.

Основними характеристиками точок привабливості є:

- Високий рівень взаємодії користувачів. Наприклад, велика кількість фотографій, зроблених у певному місці, або регулярне згадування його в дописах.
- Емоційна привабливість. Місце викликає позитивні емоції, які стимулюють людей ділитися враженнями з іншими.
- Доступність. Це фактор, який впливає на частоту відвідувань. Сюди входять зручне розташування, транспортна доступність або популярність серед місцевого населення.
- Унікальність. Локація може виділятися своєю архітектурою, природними пейзажами чи пропонувати незвичні види активностей.

## 1.2 Огляд соціальних мереж

Соціальні мережі є важливим елементом сучасного інформаційного простору, об'єднуючи мільйони користувачів з усього світу. Вони сприяють формуванню колективної думки, поширенню інформації та впливають на поведінку людей. Завдяки можливостям взаємодії, обміну контентом і розповсюдженню рекомендацій, соціальні мережі стали потужним інструментом для аналізу даних та вивчення поведінкових моделей.

Користувачі соціальних мереж публікують текст, фото, відео, обговорюють актуальні теми, залишають відгуки та формують репутацію місць або послуг. Ця

інформація дозволяє прогнозувати популярність локацій, створювати персоналізовані рекомендації та виявляти сучасні тренди.

Соціальні мережі також є важливим засобом маркетингу. Завдяки таргетованій рекламі бренди можуть досягати аудиторії з високою точністю, впливаючи на їхні вподобання.

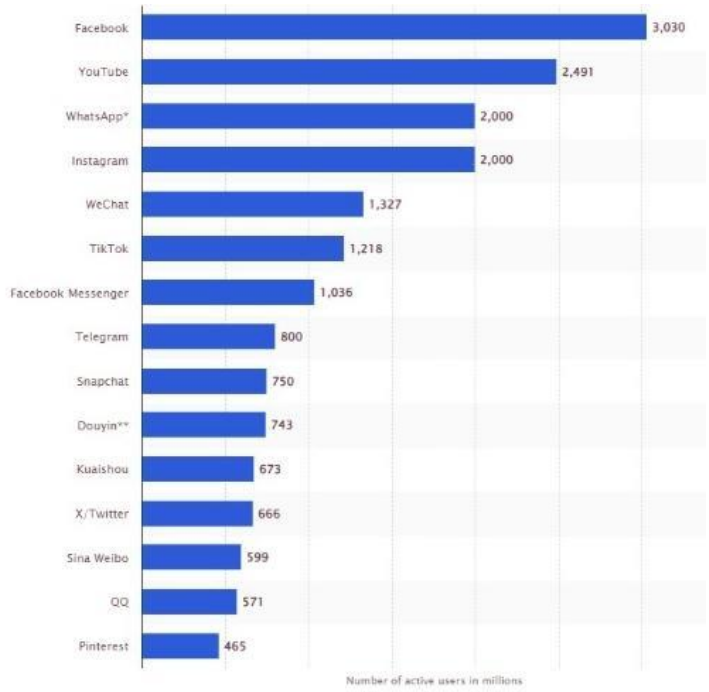



Рис. 1.1 − Популярність соціальних мереж у світі

## 1.3 Визначення соціальних мереж

Соціальні мережі — це онлайн-платформи для створення профілів, спілкування з іншими користувачами та обміну контентом. Найпопулярнішими серед них є Facebook, Instagram, Twitter, TikTok, які забезпечують широкий спектр функцій для користувачів.

Окрім загальних платформ, існують спеціалізовані соціальні мережі, такі як TripAdvisor, Yelp, Foursquare. Вони орієнтовані на відгуки про місця, заклади чи події. Ці мережі відіграють важливу роль у формуванні рекомендацій для подорожей, харчування чи відпочинку.

Соціальні мережі об’єднують можливості обміну інформацією, створення груп за інтересами, просування брендів і поширення відгуків. Це робить їх багатим джерелом даних для аналізу та прогнозування.

Аналіз даних із соціальних мереж відкриває широкі перспективи у сфері маркетингу, планування та дослідження ринку. Наступні розділи детальніше розглядають методи збору й обробки даних, а також виклики, які виникають у цьому процесі.

### 1.4 Роль соціальних мереж у створенні популярності місць

Соціальні мережі значно впливають на популярність різних локацій, таких як ресторани, кафе, туристичні об’єкти, магазини та інші заклади. Вони виступають важливим каналом обміну інформацією, дозволяючи користувачам ділитися враженнями, оцінками та відгуками. Це формує колективну думку, яка має вирішальне значення для репутації та відвідуваності місць.

Одним із ключових способів впливу є публікація відгуків та рейтингових оцінок. Наприклад, дослідження Nielsen Global Online Consumer Survey (2015) показало, що 70% користувачів довіряють відгукам у соціальних мережах більше, ніж офіційній рекламі.

Візуальний контент, такий як фотографії й відео, також грає важливу роль. Яскраві та емоційні зображення місць здатні створювати сильний емоційний зв’язок між потенційними відвідувачами та брендом, заохочуючи їх до відвідування.

Геолокаційні сервіси й теги полегшують пошук цікавих місць поблизу. Водночас рекомендації від інфлюенсерів, популярних блогерів чи лідерів думок сприяють різкому зростанню популярності. Зокрема, співпраця з інфлюенсерами може збільшити відвідуваність місця на 25%.

Прямі активності в соціальних мережах, як-от конкурси, промо-акції або знижки для підписників, активно залучають нових користувачів. Соціальні групи та особисті зв’язки забезпечують ефект "сарафанного радіо", поширюючи інформацію в рази швидше, ніж традиційні методи.

Соціальні мережі є не лише інструментом реклами, а й платформою для реальних відгуків та рекомендацій. Це дає користувачам змогу приймати

обґрунтовані рішення щодо вибору місць, які вони хочуть відвідати, а закладам — ефективно формувати свій імідж і залучати нову аудиторію.

### 1.5 Методи збору інформації з соціальних мереж

Соціальні мережі є багатим джерелом даних, які дозволяють досліджувати інтереси, поведінку та уподобання користувачів. Отримана інформація знаходить застосування у створенні персоналізованого контенту, аналізі популярності місць, розробці маркетингових стратегій і прогнозуванні трендів.

Перед аналізом даних важливо чітко визначити, які саме дані збираються із соціальних мереж. Класифікація даних на категорії допомагає ефективніше їх обробляти та використовувати відповідно до цілей дослідження.

### 1.6 Типи даних

- Текстові дані**.** Текстові дані включають коментарі, пости, твіти, приватні повідомлення та хештеги. Вони містять ключову інформацію про думки, емоції та оцінки користувачів щодо певних місць або подій. Аналіз тексту дозволяє ідентифікувати популярні теми, проблеми та загальні настрої.
- Мультимедійні дані**.** До мультимедійних даних належать фотографії, відео та GIF-файли, які публікують користувачі. Цей тип контенту дає змогу оцінити візуальну привабливість місця, а також ідентифікувати елементи, які привертають увагу аудиторії.
- Поведінкові дані**.** Цей тип даних базується на активності користувачів, зокрема лайках, шерінгах, підписках, коментарях і часу, проведеному на сторінках. Поведінкові дані допомагають зрозуміти, що саме приваблює аудиторію, і дозволяють прогнозувати майбутню активність користувачів.
- Геолокаційні дані**.** Ці дані відображають фізичне місцезнаходження користувачів, що дозволяє визначити, які місця найчастіше відвідуються. Геолокаційні дані корисні для аналізу популярності локацій, планування маркетингових кампаній і прогнозування потенційного попиту.

Ці типи даних із соціальних мереж надають унікальні можливості для

розуміння поведінки користувачів та створення стратегій, спрямованих на залучення нових аудиторій і підвищення популярності місць.

## 1.7 Методи збору даних

Збір даних із соціальних мереж здійснюється за допомогою різних підходів, що залежать від цілей дослідження та типу потрібної інформації. Основні методи, які використовують:

- API платформ соціальних мереж

Багато соціальних мереж надають доступ до своїх даних через API (Application Programming Interface). Це дозволяє автоматизовано збирати інформацію про пости, коментарі, рейтинги, хештеги та інші параметри.

- Веб-скрейпінг

Технологія автоматичного збирання даних з відкритих веб-сторінок соціальних мереж. Вона використовується, коли API не надає доступу до потрібної інформації.

- Аналітика взаємодій

Збирання поведінкових даних через внутрішні аналітичні інструменти платформ. Наприклад, Facebook Insights або Instagram Analytics надають інформацію про взаємодії, демографічний склад аудиторії та географічне охоплення.

- Опитування та форми зворотного зв’язку

Крім автоматичних методів, корисною може бути інформація, яку користувачі надають добровільно. Анкети або відгуки на платформах, таких як TripAdvisor, дозволяють глибше зрозуміти сприйняття місця.

## 1.8 Виклики та обмеження

Аналіз даних з соціальних мереж відкриває нові можливості для розуміння уподобань та поведінки користувачів, але водночас супроводжується низкою викликів. Ці виклики пов’язані як із технічними, так і з етичними аспектами роботи з даними. З одного боку, соціальні мережі надають величезний обсяг інформації,

але з іншого — якість та доступність цих даних, а також дотримання норм конфіденційності, залишаються ключовими проблемами:

- Конфіденційність даних: Збір та аналіз даних має відповідати законодавчим нормам, таким як GDPR (General Data Protection Regulation), щоб уникнути порушення прав користувачів.
- Надійність даних: Дані, зібрані з соціальних мереж, можуть бути суб'єктивними та спотвореними, оскільки базуються на особистій думці користувачів.
- Технічні бар'єри: Не всі платформи надають відкритий доступ до своїх даних, що може обмежувати можливості аналізу.

## 1.9 Способи використання даних соціальних мереж

Дані, зібрані з соціальних мереж, є потужним інструментом для бізнесу, маркетингу, соціальних досліджень та управління репутацією. Вони дозволяють розробляти персоналізовані стратегії, прогнозувати поведінку споживачів та покращувати комунікацію з аудиторією. У цьому розділі розглядаються основні способи застосування цих даних у різних сферах.

### 1.9.1 Таргетинг та персоналізація контенту в соціальних мережах

Таргетинг дозволяє показувати рекламу та контент виключно тим користувачам, які з найбільшою ймовірністю зацікавляться пропозицією. Завдяки аналізу даних, таких як вік, стать, інтереси, поведінка в інтернеті та геолокація, компанії можуть:

1. Сегментувати аудиторію:
   - Розділяти користувачів на групи залежно від їхніх інтересів, звичок та демографічних показників.
   - Формувати різні рекламні кампанії для кожного сегмента.
2. Персоналізувати контент:
   - Пропонувати релевантні продукти чи послуги залежно від попередньої активності користувача.

- Наприклад, показувати оголошення про туристичні тури людям, які раніше переглядали відповідні сторінки або публікували контент на цю тему.

3. Динамічний ретаргетинг:

- Повторно залучати користувачів, які взаємодіяли з брендом, але не завершили покупку.

- Наприклад, показувати оголошення про товари, які залишилися у кошику на сайті.

4. Використовувати психологічні тригери:

- Використання слів, зображень або кольорів, які відповідають інтересам та емоційному стану цільової аудиторії.

Приклад використання:

Сервіс потокової музики Spotify активно застосовує персоналізацію, пропонуючи плейлисти, складені на основі музичних уподобань користувача, його активності в додатку та популярних треків серед схожих профілів.

Вплив на ефективність:

Таргетинг та персоналізація значно підвищують ефективність маркетингових кампаній, дозволяючи зменшити витрати на рекламу, залучати нових клієнтів та утримувати постійних.

### 1.9.2 Інтеграція соціальних мереж із геолокаційними сервісами

Інтеграція даних із соціальних мереж та геолокаційних сервісів відкриває широкі можливості для аналізу поведінки користувачів, прогнозування попиту та підвищення точності маркетингових стратегій. Завдяки використанню даних про місцезнаходження та активність у соціальних мережах, компанії можуть краще розуміти, як, де та коли взаємодіяти зі своєю аудиторією.

Основні аспекти інтеграції:

- Моніторинг популярності локацій:

Геолокаційні теги та «чекіни» користувачів у соціальних мережах допомагають визначити, які місця користуються популярністю серед аудиторії.

Наприклад, ресторани чи туристичні пам’ятки можуть аналізувати кількість

відміток та фотографій із їхньої локації.

- Персоналізовані рекомендації:

Використовуючи геолокаційні дані, платформи можуть запропонувати користувачам місця, які відповідають їхнім інтересам і знаходяться поблизу.

Наприклад, додатки на кшталт TripAdvisor чи Google Maps рекомендують заклади залежно від активності користувачів у соціальних мережах.

- Організація локальних акцій:

Підприємства можуть проводити акції або знижки для людей, які знаходяться у певній географічній зоні.

Наприклад, пропонувати спеціальні пропозиції тим, хто «чекіниться» у закладі через Facebook чи Instagram.

- Аналіз туристичних потоків:

За допомогою даних соціальних мереж можна аналізувати маршрути туристів, відвідуваність визначних місць і навіть передбачати, які місця стануть трендовими у майбутньому.

- Візуалізація даних:

Створення теплових карт на основі даних про геолокацію та активність у соціальних мережах. Це дозволяє бізнесу зрозуміти, які райони міста є найбільш активними з точки зору потенційних клієнтів.

### 1.9.4 Використання даних соціальних мереж у бізнес-стратегіях компаній

Крупні компанії активно використовують дані, зібрані у соціальних мережах, для підвищення ефективності бізнес-процесів, оптимізації маркетингових стратегій та формування більш персоналізованого підходу до клієнтів. Дані, отримані з відгуків, візуального контенту, хештегів, геолокаційних тегів та поведінкових взаємодій, забезпечують компаніям ключові інсайти для ухвалення рішень.

**1. Spotify: Персоналізація музичних рекомендацій**

Spotify, провідний сервіс потокового аудіо, активно використовує дані соціальних мереж та поведінки користувачів для формування персоналізованих рекомендацій. Компанія аналізує відгуки користувачів у соціальних мережах, їхню

активність на платформі та зібрані з інших джерел дані про смаки та вподобання, щоб рекомендувати музику, яка найбільше відповідає інтересам кожного окремого слухача.

Spotify використовує алгоритми машинного навчання для оцінки кожної пісні, базуючись на таких параметрах, як кількість прослуховувань, популярність серед певних груп слухачів і взаємодії з іншими користувачами. Завдяки цьому підходу, компанія змогла збільшити час прослуховування музики на платформі та поліпшити користувацький досвід, адже рекомендації стали точнішими і більш персоналізованими.

**2. Amazon (Dress the Population)**

Компанія Dress the Population, що продає високоякісний одяг, використала технології машинного навчання (ML) для персоналізації покупок на платформі Amazon Web Services (AWS). Завдяки співпраці з компанією Obviyo, яка є партнером AWS, Dress the Population змогла впровадити персоналізовані рекомендації на основі алгоритмів Amazon Personalize, що дозволяють швидко адаптувати пропозиції до поведінки користувачів у реальному часі. Це дозволило підвищити коефіцієнт конверсії на 28%.

Рішення, яке було реалізовано за лічені дні, використовує Growth Bots для персоналізації рекомендацій товарів на вебсайті компанії. Протягом перших 72 годин після запуску понад 14% відвідувачів взаємодіяли з персоналізованими рекомендаціями, а 48% всіх продажів включали хоча б одну таку взаємодію. Результатом впровадження персоналізації стало значне зростання доходів, а також покращення досвіду користувачів.

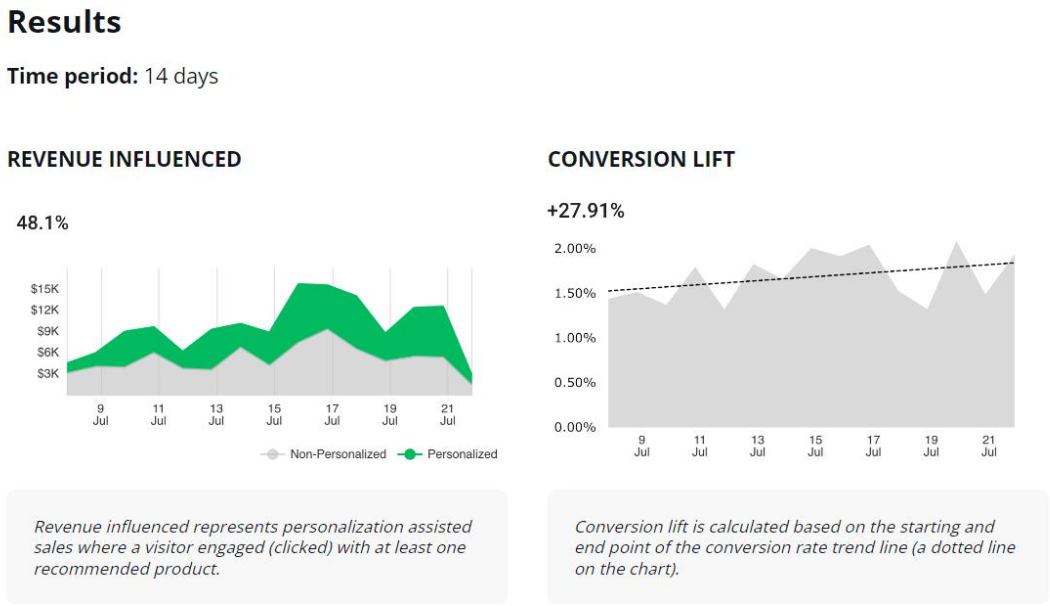



Рис. 1.2 − Графік коефіцієнту конверсії “Dress the Population”

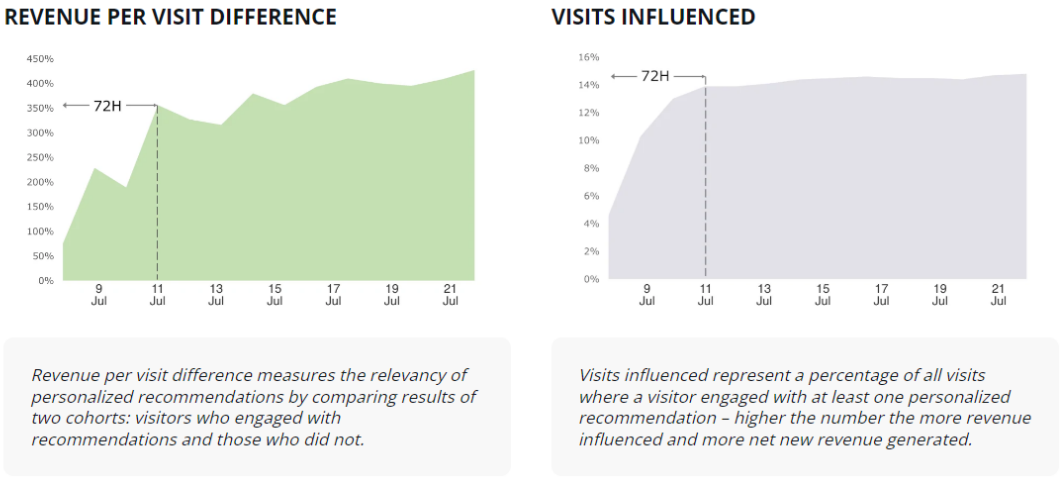



Рис. 1.3 − Графік відвідувань “Dress the Population

## 3. Coca-Cola: Взаємодія з користувачами через соціальні мережі

Coca-Cola активно використовує соціальні мережі для взаємодії з аудиторією та побудови маркетингових стратегій. Один з найбільших успіхів компанії був досягнутий через кампанію “Share a Coke” (Поділись Кока-Колею), де на пляшках Coca-Cola були зображені популярні імена людей. Кампанія сприяла активному

поширенню фото у соціальних мережах, коли люди ділилися зображеннями своїх пляшок з іменами друзів і родичів.

Coca-Cola також використовує дані з соціальних мереж для моніторингу реакцій на свої маркетингові кампанії та оцінки настроїв споживачів. Компанія аналітично обробляє відгуки та пости в соціальних мережах, щоб своєчасно реагувати на потенційні кризи або покращувати маркетингові стратегії. За допомогою таких даних Coca-Cola змогла створити більш персоналізовані кампанії, що сприяло збільшенню продажів та покращенню іміджу бренду на світовому ринку.

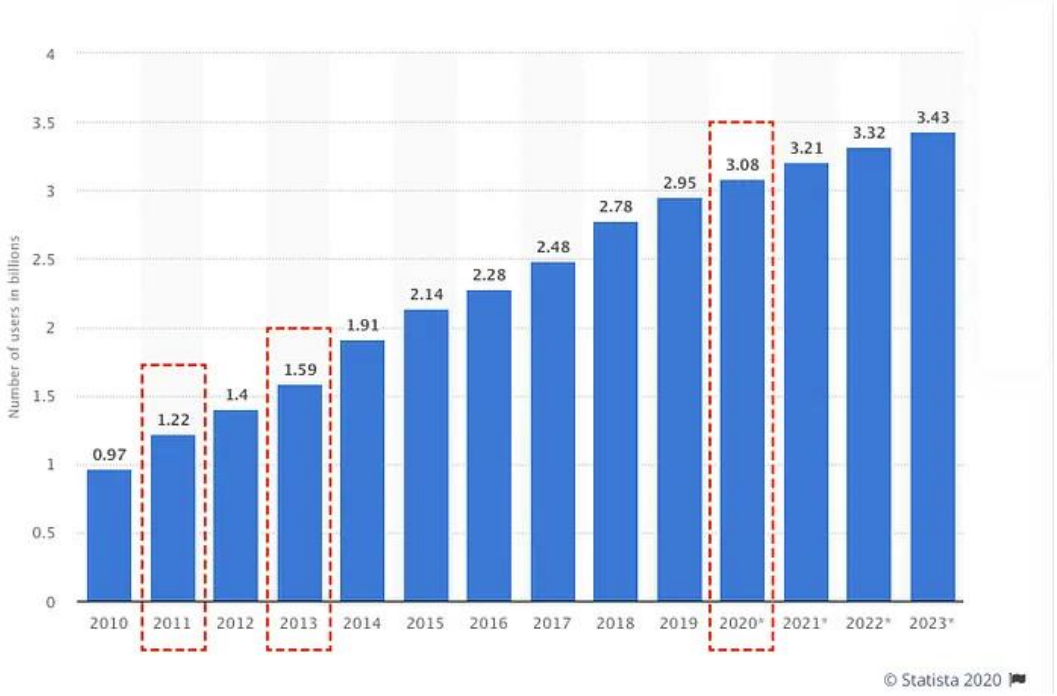



Рис. 1.4 − Графік успіху кампанії “Share a Coke”

# РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ ДЛЯ ВИЗНАЧЕННЯ ТОЧОК ПРИВАБЛИВОСТІ

## 2.1 Використання API для збору даних

У розробці сучасних додатків часто використовуються сторонні API для збору, обробки та інтеграції даних. Це дозволяє значно спростити розробку та знизити витрати часу на створення власних рішень для збору та обробки даних. API (Application Programming Interface) є потужним інструментом для доступу до зовнішніх джерел даних, таких як веб-сервіси, бази даних чи інші платформи, які пропонують функціональність, необхідну для виконання конкретних завдань.

У випадку нашої системи для аналізу настроїв відгуків про місця, для збору даних використовується сторонній API Foursquare. Цей API дозволяє отримувати інформацію про різноманітні місця (ресторани, кафе, магазини тощо), відгуки користувачів, рейтинги та інші характеристики, що необхідні для подальшого аналізу. Використання Foursquare API дозволяє уникнути необхідності збирати дані вручну або будувати власну інфраструктуру для збору інформації, що значно прискорює процес розробки.

Завдяки стороннім API, додатки можуть отримувати актуальну та точну інформацію в реальному часі, що підвищує ефективність системи та дозволяє її швидше адаптувати до змін у зовнішньому середовищі. Проте, слід зазначити, що залежність від сторонніх сервісів може також створювати певні ризики, наприклад, недоступність API через технічні проблеми або зміни в його роботі. Тому важливо ретельно вибирати надійні та стабільні джерела даних, а також мати механізми для обробки помилок і відновлення роботи системи при неполадках з зовнішніми сервісами.

## 2.2 API Foursquare

Для збору даних було використано  Foursquare API, яке надає доступ до великої кількості інформації про місця, включаючи їхні геолокаційні координати, відгуки користувачів, фото та іншу інформацію. API підтримує різні кінцеві точки

(endpoints), що дозволяють отримувати інформацію про локації, рекомендації, події та інші аспекти. Для роботи з цим API необхідно отримати ключ доступу (API ключ), який дозволяє ідентифікувати ваш додаток і контролювати використання запитів.

Структура API Foursquare:

API Foursquare забезпечує доступ до великої кількості інформації через різноманітні кінцеві точки (endpoints). Найбільш важливими з них є:

- /places/search — Використовується для пошуку місць на основі геолокації (широта, довгота) та різних фільтрів (категорії, радіус пошуку, популярність тощо).
- /places/{fsq id} — Дозволяє отримати детальну інформацію про конкретне місце, таке як адреса, тип, опис, фотографії та інші метадані.
- /places/{fsq id}/tips — Повертає відгуки користувачів про конкретне місце, що є важливим для аналізу настроїв.
- /places/{fsq id}/photos — Підтримує доступ до фотографій місця, що можуть допомогти в аналізі контексту або візуальних характеристик.
- /places/categories — Повертає список категорій місць, що дозволяє класифікувати місця за типом (ресторани, магазини, розваги тощо).

Кожна з цих кінцевих точок має свої параметри запиту, що дозволяє гнучко налаштовувати пошук та отримувати релевантну інформацію. Дані зазвичай надаються у форматі JSON, що є зручним для подальшої обробки.

## 2.3 Типи даних

Foursquare API дозволяє отримувати широкий спектр даних, що охоплює:

- Геолокаційні дані: координати місць, адреси, регіони.
- Соціальні дані: відгуки, рейтинги, популярні часи відвідування.
- Візуальні дані: зображення, додані користувачами, що надають контекст до локацій.
- Описова інформація: категорії місць, години роботи, пов’язані заходи.

Ця інформація дозволяє формувати комплексний портрет локацій для подальшого

аналізу та інтеграції.

## 2.4 Формат даних

Для пошуку місць ми відправляємо GET-запити до ендпоінтів API Foursquare, де вказуємо параметри. Наприклад:

**Запит:**

“*https://api.foursquare.com/v3/places/search?query={query}&ll={ll}&radius={radius}&limit={limit}”*

{query} = Ключове слово для пошуку, наприклад - coffee

{ll} = Широта/довгота (точка для пошуку), наприклад -50.450001,-30.523333

{radius} = Радіус пошуку в метрах, наприклад -1000

{limit} = Кількість отриманих місць у списку, максимум за раз - 50

У відповідь ми отримаємо:

Список “results”, це масив місць, які підходять під наші запити.

```
"results": [
    {
        "fsq_id": "6344336adf00a123209dd752",
        "categories": [
            {
                "id": 13035,
                "name": "Coffee Shop",
                "short_name": "Coffee Shop",
                "plural_name": "Coffee Shops",
                "icon": {
                    "prefix": "https://ss3.4sqi.net/img/categories_v2/food/coffeeshop_",
                    "suffix": ".png"
                }
            }
        ],
        "chains": [],
        "closed_bucket": "LikelyOpen",
        "distance": 176,
        "geocodes": {
            "drop_off": {
                "latitude": 40.711419,
                "longitude": -74.006219
            },
            "main": {
                "latitude": 40.711286,
                "longitude": -74.006123
            },
            "roof": {
                "latitude": 40.711286,
                "longitude": -74.006123
            }
        },
        "link": "/v3/places/6344336adf00a123209dd752",
        "location": {
            "address": "140 Nassau St",
            "census_block": "360610015011012",
            "country": "US",
            "cross_street": "Beekman St",
            "dma": "New York",
            "formatted_address": "140 Nassau St (Beekman St), New York, NY 10038",
            "locality": "New York",
            "postcode": "10038",
            "region": "NY"
        },
        "name": "Variety Coffee Roasters",
        "related_places": {},
        "timezone": "America/New_York"
    },
```


Рис. 2.1 − Структура відповіді ендпоінту “/places” від API Foursquare

Після отримання місці, ми проводимо додаткову фільтрацію по кількості відгуків, після чого робимо наступний GET-запит, для отримання відгуків.

**Запит**

https://api.foursquare.com/v3/places/{fsq_id}/tips?limit={limit}

{fsq_id}= Унікальній ідентифікатор місця, наприклад 4ea0afbf9adf1e334e4cc0e6

{limit} = Кількість відгуків у відповіді

В результаті отримуємо список відгуків у форматі:

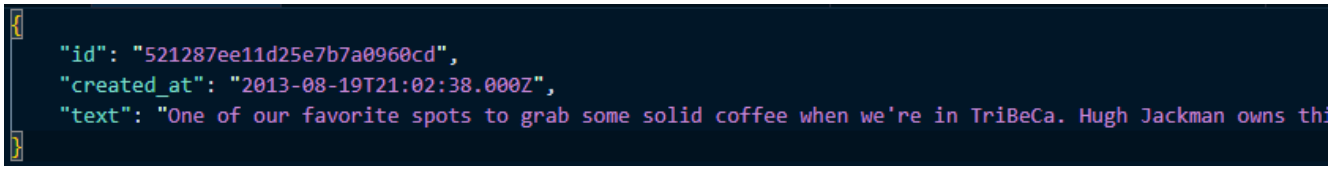



Рисунок 2.2 − Структура відповіді ендпоінту “/places/tips” від API Foursquare

Для зберігання отриманих даних були створені чотири основні сутності, які використовуються для зберігання інформації про місця, категорії та відгуки з API Foursquare в базі даних:

1. PlaceEntity — це сутність, яка представляє місце, отримане з Foursquare. Вона містить:

- fsqId — унікальний ідентифікатор місця з Foursquare.
- name — назва місця.
- location — вбудована сутність LocationEmbeddable, що містить адресу, форматовану адресу, місцевість, поштовий індекс, регіон та країну.
- distance — відстань до місця від заданої точки пошуку.
- categories — список категорій місця (наприклад, "кафе", "ресторан"), зберігається у вигляді списку сутностей CategoryEntity.

2. LocationEmbeddable — це вбудована сутність, яка містить інформацію про місцезнаходження, таку як адреса, форматована адреса, місцевість, поштовий індекс, регіон та країна. Вона використовується у сутності

3. PlaceEntity для зберігання даних про розташування місця.

4. CategoryEntity — ця сутність описує категорію місця. Вона містить:

- id — унікальний ідентифікатор категорії в базі даних.
- categoryId — ідентифікатор категорії з Foursquare.
- name — повне ім'я категорії.
- shortName — скорочене ім'я категорії.
- iconPrefix та iconSuffix — префікс і суфікс для отримання іконки категорії.

5. TipEntity — це сутність для зберігання відгуків про місця. Вона містить:

- id — унікальний ідентифікатор відгуку.
- text — текст відгуку.
- created_at — дата створення відгуку.
- place — зв'язок з сутністю PlaceEntity, що дозволяє асоціювати відгук з конкретним місцем.

Ці сутності разом дозволяють ефективно зберігати і організовувати дані, отримані з Foursquare, включаючи місця, їх категорії та відгуки, в реляційній базі даних.

## 2.5 Зберігання даних

Дані, отримані через API Foursquare, зберігаються у реляційній базі даних PostgreSQL. Структура бази забезпечує логічну організацію інформації про місця, категорії та відгуки.

Опис таблиць:

**places** (Місця):

- Зберігає основну інформацію про місця.
- fsq_id: Унікальний ідентифікатор місця (первинний ключ).
- distance: Відстань від заданої точки до місця.
- address: Адреса місця.
- country, locality, region: Інформація про локацію місця.
- formatted_address: Відформатована адреса.
- name: Назва місця.

**categories** (Категорії):

- Відображає категорії, до яких належать місця.
- id: Унікальний ідентифікатор (первинний ключ).
- category_id: Ідентифікатор категорії.
- icon_prefix, icon_suffix: Інформація про іконку категорії.
- name, short_name: Назва та скорочена назва категорії.
- place_id: Зв'язок із таблицею places через зовнішній ключ.

**tip_entity** (Відгуки):

- Містить інформацію про текстові відгуки користувачів для місць.
- id: Унікальний ідентифікатор відгуку (первинний ключ).
- created_at: Час створення відгуку.
- text: Текст відгуку.
- place_id: Зв'язок із таблицею places через зовнішній ключ.

## 2.6 Ключові API та їх обмеження

У процесі збору даних з API Foursquare виникла необхідність застосування кількох ключів API та проксі-серверів для обходу обмежень, накладених на кількість запитів. API Foursquare має певні обмеження на кількість запитів, які можна зробити з одного ключа API протягом певного часу. Це обмеження включає ліміти на кількість запитів на годину або на день, що може суттєво ускладнити збір великої кількості даних, особливо якщо необхідно отримати інформацію по великій кількості місць.

Щоб обійти ці обмеження, були прийняті наступні рішення:

1. **Використання списку API ключів**: Для забезпечення безперервного збору даних був створений список API ключів. Кожен ключ використовувався по черзі, що дозволяло обійти обмеження, пов'язані з кількістю запитів. У разі досягнення ліміту для одного ключа, система автоматично переходила до наступного ключа в списку. Це дозволяє підтримувати стабільну роботу системи збору даних без перебоїв.

2. **Паралельні запити через проксі**: Для збільшення швидкості збору даних і подолання обмежень на кількість одночасних запитів з одного ключа використовувалася технологія паралельних запитів через проксі-сервери. Проксі дозволяють направляти запити через різні IP-адреси, що також допомагає обійти обмеження, накладені на одну IP-адресу. Таким чином, запити до API Foursquare відправляються через різні проксі-сервери, що дозволяє значно збільшити обсяг запитів за одиницю часу, не порушуючи обмеження API.

Завдяки цьому підходу вдалося зібрати велику кількість даних за короткий час, обійшовши обмеження API Foursquare, а також забезпечити стабільну роботу системи збору даних навіть у разі великих обсягів запитів.

## 2.7 Первинна очистка та фільтрація отриманих даних

Процес обробки даних після їх збору є ключовим етапом для подальшого використання отриманих даних у системах аналізу чи рекомендацій. Оскільки дані, отримані з API, можуть бути неповними або мати різні формати, вони потребують ретельної обробки.

Кроки, виконані для обробки даних:

1. **Видалення дублікатів**: Для запобігання дублюванню даних, що можуть з'являтися при кількох запитах до API, була здійснена перевірка унікальності кожного запису (наприклад, за ідентифікатором місця). Дублікатні записи були вилучені з даних для збереження їхньої цілісності.

2. **Фільтрація за релевантністю**: Деякі дані можуть не відповідати заданим критеріям. Наприклад, місця без визначеної категорії або з некоректною адресою були відфільтровані, щоб забезпечити точність і відповідність даних до необхідних вимог. Цей крок допоміг залишити лише релевантні записи, що мають значення для подальшої обробки.

3. **Обробка пропущених значень**: У разі відсутності деяких важливих полів, таких як адреса чи координати, для таких записів було застосовано дефолтні значення або їх було виключено з подальшої обробки. Це дозволило зберегти якість

даних, уникнути невизначеності в результатах та забезпечити коректність під час аналізу.

Ці етапи обробки допомогли значно покращити якість зібраних даних, усунути потенційні помилки та підготувати їх до подальших етапів, таких як нормалізація, аналіз чи інтеграція в систему рекомендацій.

## 2.8 Використання моделей на основі трансформерів у розпізнаванні настрою тексту

Моделі на основі трансформерів, такі як BERT та DistilBERT, стали основою сучасного аналізу тексту завдяки їх здатності враховувати контекст слова в реченні. Ці моделі застосовуються для визначення настрою тексту, класифікації відгуків, аналізу тональності повідомлень у соціальних мережах та інших задач.

### 2.8.1 Word2Vec

Word2Vec — це одна з найпопулярніших моделей для створення вбудовувань слів, розроблена Google у 2013 році. Її мета — відображати слова у вигляді векторів, що дозволяє зберігати семантичні зв’язки між словами. Це стало значним кроком вперед у порівнянні з традиційними методами, що використовували представлення слів через однакові або часткові ознаки (наприклад, "ручка" і "перо" були б представлені як однакові).

Word2Vec працює за допомогою двох основних архітектур: CBOW (Continuous Bag of Words) і Skip-gram. CBOW намагається передбачити поточне слово на основі його контексту, тоді як Skip-gram виконує зворотне завдання — передбачає контекст для заданого слова. Обидві архітектури використовують нейронні мережі, щоб оптимізувати вектори слів, а сам процес навчання зводиться до максимізації ймовірності передбачення слова або контексту.

Word2Vec є дуже швидким для обробки великих корпусів тексту, що дозволяє моделі ефективно працювати з величезними обсягами даних, такими як тексти з інтернету. Модель створює контекстно-вільні представлення слів, що означає, що кожне слово має своє фіксоване векторне представлення, незалежно

від контексту, в якому воно використовується. Це може бути корисно, коли важливо зберігати семантичні зв’язки між словами, але одночасно може призвести до проблем при роботі з омонімами (наприклад, слово "bank" у контексті "фінансова установа" та "річкова берега" буде мати однакове представлення).

Word2Vec широко використовується в багатьох завданнях обробки природної мови, таких як класифікація тексту, машинний переклад, пошукові системи, та інші завдання, де важливо враховувати семантичні зв’язки між словами. Однак, оскільки модель є контекстно-вільною, вона не завжди здатна враховувати значення слова в залежності від контексту, що обмежує її застосування у складніших задачах, де значення слова змінюється залежно від його оточення.

### 2.8.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) — це модель, розроблена компанією Google у 2018 році, яка стала революційним підходом до попереднього навчання мовних представлень. Вона використовує двонапрямний механізм трансформерів, що дозволяє краще враховувати контекст слів як зліва, так і справа в тексті. Завдяки цьому підходу, BERT встановила нові стандарти точності для широкого спектра завдань обробки природної мови (NLP), таких як класифікація тексту, заповнення пропусків, відповіді на запитання та аналіз тональності. Модель стала базою для багатьох подальших досліджень і розробок у сфері NLP, істотно прискоривши прогрес у цій галузі.

BERT — це метод попереднього навчання мовних представлень, що передбачає створення універсальної моделі "розуміння мови" на основі великого корпусу тексту (наприклад, Вікіпедії). Ця модель згодом використовується для вирішення конкретних задач обробки природної мови (NLP), таких як відповіді на запитання чи аналіз тональності. BERT перевершує попередні підходи, адже є першим глибоко двонапрямним і повністю неконтрольованим методом попереднього навчання для NLP.

Неконтрольований підхід означає, що BERT навчався виключно на "сирому" тексті, без міток, що є важливим, адже величезна кількість таких даних доступна в інтернеті для багатьох мов.

Представлення слів може бути або контекстно-вільним, або контекстуальним, причому контекстуальні представлення можуть бути одно- чи двонапрямними. Контекстно-вільні моделі, як-от word2vec чи GloVe, створюють одне "вбудовування" для кожного слова в словнику, тож слово bank матиме однакове представлення в контекстах “bank deposit” і “river bank”. Контекстуальні моделі, навпаки, формують представлення слова залежно від інших слів у реченні.

BERT побудований на основі попередніх досліджень у сфері контекстуальних представлень, таких як Semi-supervised Sequence Learning, Generative Pre-Training, ELMo та ULMFit. Проте ці моделі були лише однонапрямними або поверхнево двонапрямними: слово враховувало лише контекст зліва або справа. Наприклад, у реченні “I made a bank deposit” однонапрямна модель для слова “bank” враховує лише “I made a”, але не “deposit”. Деякі моделі комбінували представлення з лівого та правого контекстів, але лише поверхово. Натомість BERT формує представлення слова на основі як лівого, так і правого контексту (I made a ... deposit), починаючи з самого низу глибокої нейронної мережі, що робить його глибоко двонапрямним.

BERT використовує простий підхід: замінює 15% слів у тексті на [MASK], пропускає послідовність через глибоку двонапрямну модель трансформера та прогнозує пропущені слова. Наприклад:

Вхідні дані: the man went to the [MASK1]. he bought a [MASK2] of milk.

Мітки: [MASK1] = store; [MASK2] = gallon

Для вивчення зв’язків між реченнями BERT також навчається на завданні передбачення послідовності речень: чи є речення B логічним продовженням речення A.

- Речення A: *the man went to the store.*
- Речення B: *he bought a gallon of milk.*
- Мітка: *IsNextSentence*

- Речення B: *penguins are flightless.*
- Мітка: *NotNextSentence*

Модель має велику архітектуру (12-24 шари трансформера) та навчається на корпусах, таких як Wikipedia і BookCorpus, протягом тривалого часу (до 1 млн кроків).

BERT має дві фази використання: попереднє навчання та донавчання.

- Попереднє навчання — це дорогий процес, (компанія Google витратила 4 дні на 4–16 Cloud TPU), але він виконується лише один раз для кожної мови. Google вже випустив кілька попередньо навчених моделей. Більшість дослідників NLP не потребуватимуть самостійного навчання BERT з нуля.
- Донавчання — простий і швидкий процес. Наприклад, завдання SQuAD можна натренувати за 30 хвилин на Cloud TPU, досягнувши Dev F1-результату 91.0%, що є найкращим результатом у своєму класі.

BERT легко адаптується до різних завдань NLP, включно з аналізом речень (SST-2), пар речень (MultiNLI), слів (NER) та текстових діапазонів (SQuAD), з мінімальними модифікаціями.

Google також випустили натреновані моделі BERT-Base та BERT-Large, інформація про ці моделі описані у них у статті посвяченій цій моделі. Uncased означає, що текст (у процесі навчання моделі) було приведено до нижнього регістру перед токенізацією WordPiece (наприклад, John Smith стає john smith). Модель Uncased також видаляє будь-які акцентні знаки. Cased означає, що збережено справжній регістр літер і акцентні знаки. Зазвичай модель Uncased дає кращі результати, якщо тільки ви точно не знаєте, що інформація про регістр важлива для вашого завдання (наприклад, розпізнавання іменованих сутностей або тегування частин мови).

Таблиця 3.1.

Представлені Google BERT моделеі

| **Назва моделі** | **Характеристики** |
|---|---|
| BERT-Large, Uncased (Whole Word Masking) | 24-layer, 1024-hidden, 16-heads, 340M parameters |
| BERT-Large, Cased (Whole Word Masking) | 24-layer, 1024-hidden, 16-heads, 340M parameters |
| BERT-Base, Uncased | 12-layer, 768-hidden, 12-heads, 110M parameters |
| BERT-Large, Uncased | 24-layer, 1024-hidden, 16-heads, 340M parameters |
| BERT-Base, Cased | 12-layer, 768-hidden, 12-heads , 110M parameters |
| BERT-Large, Cased | 24-layer, 1024-hidden, 16-heads, 340M parameters |
| BERT-Base, Multilingual Cased (New, recommended) | 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters |
| BERT-Base, Multilingual Uncased (Orig, not recommended) | 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters |
| BERT-Base, Chinese | Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters |

**Проблеми з нестачею пам'яті**

Через величезні вимоги до пам’яті, навіть без урахування ресурсомісткого оптимізатора Adam, використання оригінальної моделі BERT є надто затратним.

### 2.8.3 DistilBERT

DistilBERT — це оптимізована версія BERT, створена з використанням техніки knowledge distillation (дистиляція знань). Модель була розроблена, щоб зменшити вимоги до ресурсів, зберігаючи при цьому продуктивність, близьку до оригінальної BERT, для багатьох задач обробки природної мови. Вона використовує менше пам’яті, швидше працює в режимі інференсу та підходить навіть для пристроїв із обмеженими обчислювальними можливостями, таких як мобільні пристрої.

**Архітектура DistilBERT:**

Структура: Загальна архітектура схожа на BERT, але:

- Кількість шарів зменшена вдвічі для зниження обчислювальної складності.
- Видалено token-type embeddings та pooler, що спрощує модель.
- Лінійні операції та нормалізація шарів залишаються незмінними, оскільки вони ефективно оптимізовані у сучасних бібліотеках алгебри.

Ініціалізація: Студентська модель ініціалізується на основі вчителя (BERT) шляхом копіювання кожного другого шару, що забезпечує швидку й стабільну збіжність під час навчання

**Алгоритми оптимізації:**

Дистиляція знань – Це техніка стиснення, де менша модель (студент) навчається відтворювати поведінку великої моделі (вчителя):

1. М'які мітки: Замість використання однозначних міток (one-hot) використовується розподіл ймовірностей, передбачений вчителем. Це зберігає більше інформації про "уявлення" вчителя щодо задачі.
2. Температура Softmax: Розподіл ймовірностей згладжується за допомогою параметра температури TTT, який застосовується до вчителя та студента під час навчання. Це допомагає краще передавати знання між моделями.

Формула: $\rho i = \frac{exp(zi/T)}{\sum jexp(zj/T)}$

Де $zi$ — вихід моделі для класу i, а T контролює згладжування. Під час інференсу T встановлюється на 1.

3. Функція втрат:

- Distillation loss (Lce): Враховує відмінності між розподілами ймовірностей студента та вчителя.
- Masked Language Modeling loss (Lmlm): Як у BERT, використовується для передбачення замаскованих слів.
- Cosine embedding loss (Lcos): Максимізує схожість прихованих станів студента і вчителя, щоб моделі краще узгоджувалися. Підсумкова функція втрат є лінійною комбінацією цих компонентів:

$$L = \alpha Lce + \beta Lmlm + \gamma Lcos$$

**Переваги DistilBERT**

- Зменшена кількість параметрів: На 40% менша модель порівняно з BERT.
- Швидкість інференсу: На 60% швидша, що робить її придатною для використання в реальному часі.
- Гнучкість: Може бути доопрацьована для багатьох задач обробки природної мови без втрати продуктивності.
- Можливість запуску на edge-пристроях: Підходить для мобільних пристроїв та інших середовищ із обмеженими ресурсами.

DistilBERT демонструє, як зменшення розмірів моделей може забезпечити ефективність і продуктивність без значного компромісу в точності.

Таблиця 3.2.

Продуктивність DistilBERT у порівнянні з BERT

| **Модель** | **Score** | **CoLA** | **MNLI** | **MRPC** | **QNLI** | **QQP** | **RTE** | **SST-2** | **STS-B** | **WNLI** |
|---|---|---|---|---|---|---|---|---|---|---|
| BERT-base | 79.5 | 56.3 | 86.7 | 88.6 | 91.8 | 89.6 | 69.3 | 92.7 | 89.0 | 53.5 |
| DistilBERT | 77.0 | 51.3 | 82.2 | 87.5 | 89.2 | 88.5 | 59.9 | 91.3 | 86.9 | 56.3 |

## 2.9 Інструменти розробки

Для реалізації даного проекту було обрано ряд сучасних інструментів та технологій, що забезпечують високу продуктивність, зручність у використанні та ефективність. Кожен з вибраних інструментів відповідає певним вимогам проекту та допомагає досягти необхідної функціональності.

### 2.9.1 Збір даних для навчання

Для збору даних з різних джерел та обробки даних для навчання та тестування моделі були використані наступні інструменти та технології:

1. Foursquare API:

API Foursquare дозволяє отримувати інформацію про місця, події та інші об'єкти на основі геолокації. Це включає дані про популярні локації, рейтинги, відгуки користувачів та іншу метаінформацію, яка корисна для створення рекомендаційної системи. За допомогою API можна робити запити до бази даних Foursquare, отримувати відгуки та іншу метаінформацію про місця, зберігати ці дані для подальшого аналізу.

Типи запитів:

- GET *places/search* – для отримання локацій в певному радіусі від вказаної геолокації.
- GET *places/{fsq id}*/tips – для отримання відгуків (текстових та метаінформації) про певне місце.

2. Spring WebFlux (Kotlin Reactor):

Використовується для асинхронної обробки запитів у реальному часі. Оскільки збирання даних з кількох джерел, таких як Foursquare, може бути об'ємним і потребує паралельної обробки запитів, WebFlux забезпечує асинхронність і ефективну обробку потокових даних. Завдяки реактивному підходу, кожен запит не блокує виконання інших запитів, що дозволяє обробляти великий обсяг даних в реальному часі.

За допомогою WebClient, який є частиною WebFlux, відправляються асинхронні HTTP запити до API Foursquare.

Кожен запит обробляється паралельно і результат надсилається через реактивні потоки (Flux/Mono), що дозволяє ефективно працювати з даними в реальному часі.

Наприклад, запити для отримання відгуків та інформації про місця можуть бути виконані одночасно для кількох локацій, не чекаючи на завершення попереднього запиту.

3. PostgreSQL:

Зібрані дані, які включають текстові відгуки, геолокаційні координати, рейтинги і т.д., зберігаються в реляційній базі даних PostgreSQL. Це дозволяє зберігати великі обсяги даних в структурованому вигляді і забезпечує ефективне

збереження і подальшу обробку даних.

- Для кожного місця створюється запис, який містить інформацію про локацію, відгуки користувачів, фото та інші мета-дані.
- У базі даних також зберігається інформація про геолокаційні координати кожної локації, що дозволяє пізніше виконувати аналіз на основі географічної близькості.

4. JSON (формат даних):

Дані, що надходять з API Foursquare, мають формат JSON, що є зручним для обміну інформацією через HTTP і для подальшої обробки в додатку. JSON забезпечує легку інтеграцію даних з різними системами та дозволяє зберігати як структуровану, так і метаінформацію.

- Отримані дані з API Foursquare (відгуки, рейтинги, фото) будуть парситись з JSON-формату і зберігатись в PostgreSQL.
- Після цього ці дані можуть бути використані для аналізу настроїв, прогнозування популярності місць та для генерації рекомендацій.

### 2.9.2 Обробка даних

Для підготовки датасетів для навчання моделі використовуються наступні інструменти та бібліотеки:

1. Python: Основна мова програмування для обробки даних, що надає багатий екосистему бібліотек та інструментів для аналізу, маніпуляції та візуалізації даних. Python також інтегрується з іншими компонентами системи, такими як бази даних та моделі машинного навчання.

2. nltk (Natural Language Toolkit): Бібліотека для обробки природної мови, яка дозволяє виконувати задачі, пов'язані з текстовими даними. Використовується для:

- Токенізації тексту
- Видалення стоп-слів
- Аналізу лексичних і синтаксичних структур тексту
- Підготовки тексту для подальшого навчання моделей

3. pandas: Потужна бібліотека для маніпуляції структурованими даними, як-от табличні дані. Основні функціональні можливості включають:

- Роботу з DataFrame для обробки табличних даних
- Фільтрацію, групування та агрегування даних
- Інтеграцію з різними форматами файлів (CSV, Excel тощо)

4. sqlalchemy: ORM (Object-Relational Mapping) інструмент для Python, який використовується для:

- Ефективної інтеграції з базами даних, такими як PostgreSQL
- Автоматизації запитів до бази даних
- Створення моделей для зберігання структурованих даних
- Забезпечення зручного зв’язку між Python-кодом та базою даних

### 2.9.3 Навчання моделі класифікації та її оцінка

Для навчання моделі були використані наступні технології:

**1. Підготовка текстових даних**

- Pandas (pd): Для завантаження та попередньої обробки табличних даних з файлу CSV.
- Numpy (np): Для обробки числових даних і маніпуляцій з масивами (підготовка масок і паддінгу для токенізованих даних).

**2. Обробка тексту**

- Transformers (transformers від Hugging Face):

Завантаження попередньо нетренованої моделі DistilBERT і її токенізатора.

DistilBERT — це легша версія BERT, яка використовується для обробки тексту, зокрема отримання векторних уявлень тексту.

**3. Tensor Operations**

- PyTorch (torch):

Використовується для обчислень на GPU/CPU і роботи з тензорами.

DistilBERT також реалізований у PyTorch.

**4. Машинне навчання**

Scikit-learn (sklearn):

Train-test split: Для розділення даних на тренувальну та тестову вибірки.

Logistic Regression: Алгоритм класифікації для навчання на отриманих векторах.

GridSearchCV: Пошук оптимальних параметрів логістичної регресії.

Cross-validation: Перевірка моделі DummyClassifier для встановлення базового результату.

### 2.9.4 Серверна частина

Для реалізації бекенду користувацького додатку використано наступні технології:

Spring WebFlux (Kotlin Reactor):

Реактивний фреймворк для роботи з асинхронними запитами, що забезпечує високу продуктивність та масштабованість.

Забезпечує:

- Обробку HTTP-запитів від клієнтського додатку.
- Виклик навченої нейронної мережі для генерації рекомендацій.
- Збереження та управління кешованими результатами у базі даних.

**Функціональність серверної частини:**

- Приймає запити з параметрами локації, заданої користувачем(наприклад, ключове слово, локація, радіус).
- Отримує список локацій, та порівнює їх з використанням складних алгоритмів, які базується на параметрах
- Є можливість повертати актуальні дані з кешу(PostgreSQL).
- У випадку відсутності даних у кеші, запит обробляється через модель, а результати зберігаються в базі для подальшого використання.
- Виде запис статистики записів для аналітики

### 2.9.5 Збір метрик та їх відображення

Prometheus:

- Збирає метрики запитів до бекенду, включаючи кількість оброблених

запитів, час відповіді, успішність кешування.

- Надає дані у форматі, придатному для візуалізації.

Grafana:

- Використовується для створення інтерактивних дашбордів.
- Відображає статистику запитів у вигляді графіків і таблиць:
- Кількість запитів у реальному часі.
- Відсоток успішно оброблених запитів із кешу.
- Час обробки запитів нейронною мережею.

### 2.9.6 Клієнтська частина

Для реалізації інтерфейсу користувача використано:

Flutter:

- Багатоплатформний фреймворк для створення мобільних додатків.
- Забезпечує сучасний, інтерактивний дизайн для мобільних пристроїв.

Vue.js:

- Фреймворк для створення веб-додатків.
- Використовується для реалізації веб-версії додатку з інтуїтивно зрозумілим інтерфейсом.

### 2.9.7 Інструменти розробки та середовища

Для розробки, тестування та підтримки проекту було використано наступні інструменти:

GitHub

- Використовується як система контролю версій і платформа для співпраці.
- Основні функції:

1. Зберігання вихідного коду проєкту.
2. Ведення історії змін та відстеження версій.
3. Організація командної роботи через Pull Requests, Issues, та Projects.

Visual Studio Code

- Легкий текстовий редактор для швидкого редагування файлів і роботи з фронтендом.
- Основні функції:

1. Налаштування розширень для роботи з Flutter і Vue.js.
2. Інтеграція з GitHub для управління репозиторієм.

IntelliJ IDEA

- Використовується для розробки бекенд-додатку на Kotlin із використанням Spring WebFlux.
- Основні функції:

1. Інтеграція з фреймворками Spring та бібліотекою Reactor.
2. Підтримка розробки REST API.

PyCharm

- Середовище для роботи з Python під час обробки даних і навчання моделі.
- Основні функції:

1. Вбудована підтримка роботи з бібліотеками nltk, pandas, та PyTorch.
2. Зручне налаштування віртуальних середовищ і інтеграція з SQLAlchemy для обробки даних.

PgAdmin 4

- Інструмент для управління базою даних PostgreSQL.
- Основні функції:

1. Створення таблиць і запитів.
2. Аналіз і оптимізація структури бази даних.

Postman

- Використовується для тестування REST API бекенду.
- Основні функції:

1. Тестування запитів і перевірка відповідей від серверу.
2. Збереження колекцій запитів для повторного використання.

# РОЗДІЛ 3. ОПИС РЕАЛІЗАЦІЇ ТА ПРОТОТИПУ

## 3.1 Архітектура системи

Система була розроблена з урахуванням принципу модульності, що дозволяє розділити загальну задачу на окремі, логічно взаємопов’язані частини. Це не лише спрощує процес розробки, але й забезпечує легкість тестування, інтеграції та масштабування. У рамках реалізації система складається з п’яти ключових модулів, кожен з яких відповідає за виконання конкретного підзавдання.

**Модуль збору місць та відгуків (Tip Collector)** – Модуль збору місць та відгуків, реалізований через WebClient, є REST-сервісом, який відповідає за здійснення запитів до Foursquare API для отримання даних про місця та відгуки користувачів. Сервіс працює за допомогою списку ключів доступу (API keys) та проксі-серверів, що дозволяє забезпечити стабільний доступ до зовнішнього API навіть за умов високих навантажень або обмежень по кількості запитів. Сервіс додатково надає можливість моніторингу швидкості запитів, ведучи логування часу виконання запитів та частоти запитів на секунду, що дозволяє оперативно виявляти проблеми з доступом до API. Логи допомагають аналізувати ефективність роботи системи та забезпечують прозорість обробки запитів. У випадку виникнення помилок, таких як неполадки з API чи непередбачені виключення, система забезпечує обробку помилок, а відповідні дані фіксуються в логах, що дозволяє швидко виявляти та виправляти проблеми, не впливаючи на стабільність роботи сервісу.

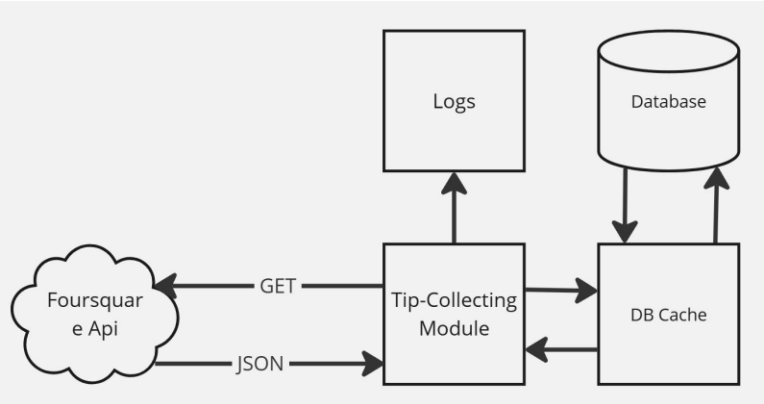



Рис. 3.1 − Архітектура модулю “Tip Collector”

**Модуль створення вибірок (Requester)** – відповідає за отримання та підготовку збалансованої вибірки з зібраних (за допомогою Tip Collector) даних для подальшої обробки, інформації про місця та відгуки. Він інтегрується з базами даних та REST-сервером для автоматичного завантаження відгуків і метаінформації про місця (категорії, підкатегорії).

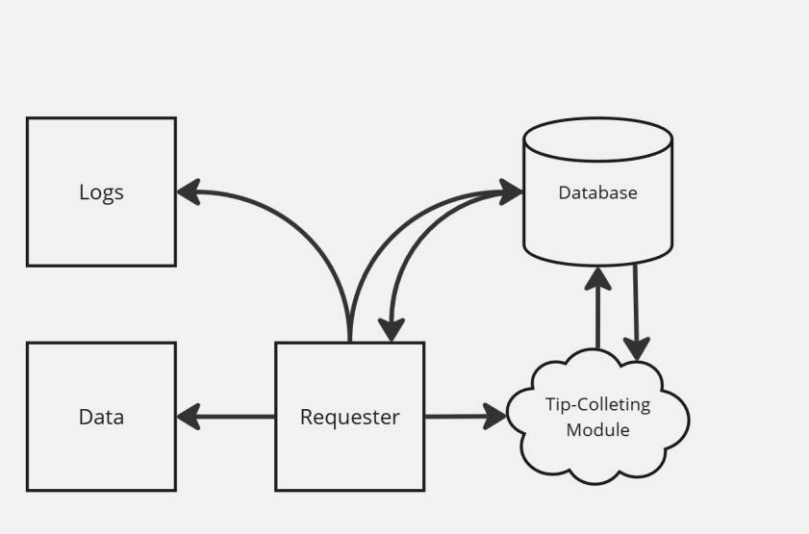



Рис. 3.2 − Архітектура модулю “Requester”

**Модуль обробки даних (Data Pre-Processor)** – виконує очищення текстів відгуків, усуваючи зайві символи та стоп-слова, готує дані для навчання моделі, а також аналізує їх настрої. Результати обробки зберігаються у базі даних, що дозволяє швидко повторно використовувати підготовлені дані без необхідності їх повторного оброблення.

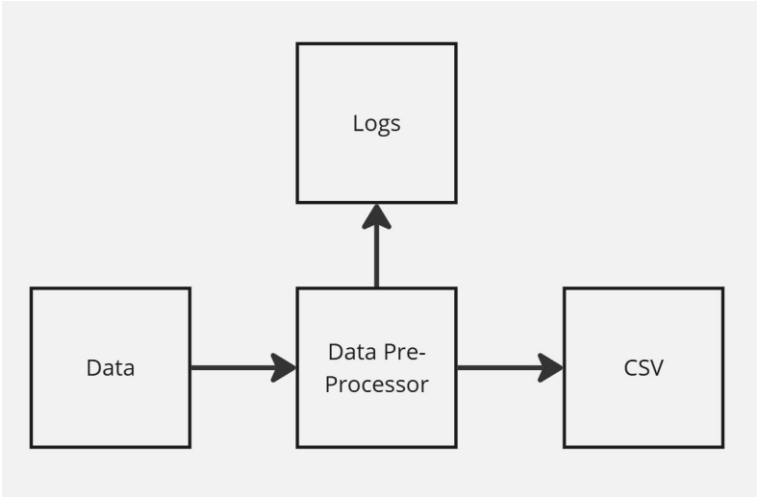



Рисунок 3.3 − Процес обробки даних за допомогою модулю “Data Pre-Processor”

**Модуль тренування моделі (ModelTrainer)**– модуль, побудований для створення моделі для аналізу настрою відгуку яка приймає одне речення (наприклад, з нашого набору даних) і видає або 1 (що буде вказувати на позитивну тональність речення), або 0 (що буде вказувати на негативну тональність).

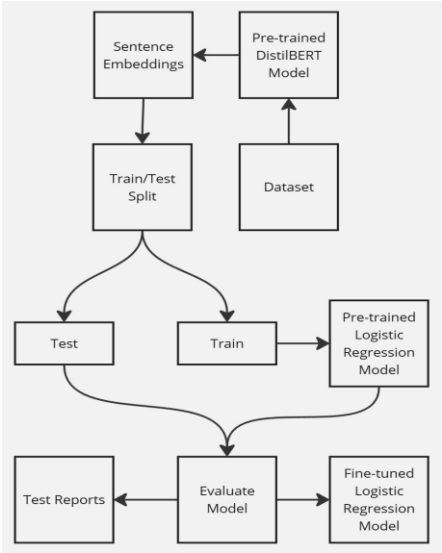



Рисунок 3.4 − Кроки навчання моделі

**Система пошуку і взаємодії з користувачем** – це остаточний компонент, який дозволяє користувачам взаємодіяти з натренованную моделлю через клієнтський інтерфейс. Запити можуть надходити через клієнтські додатки на

основі Vue.js або Flutter. Користувачі мають можливість швидко знаходити найкращі місця за ключовими словами чи категоріями, отримуючи результат, заснований на аналізі настроїв реальних відгуків.

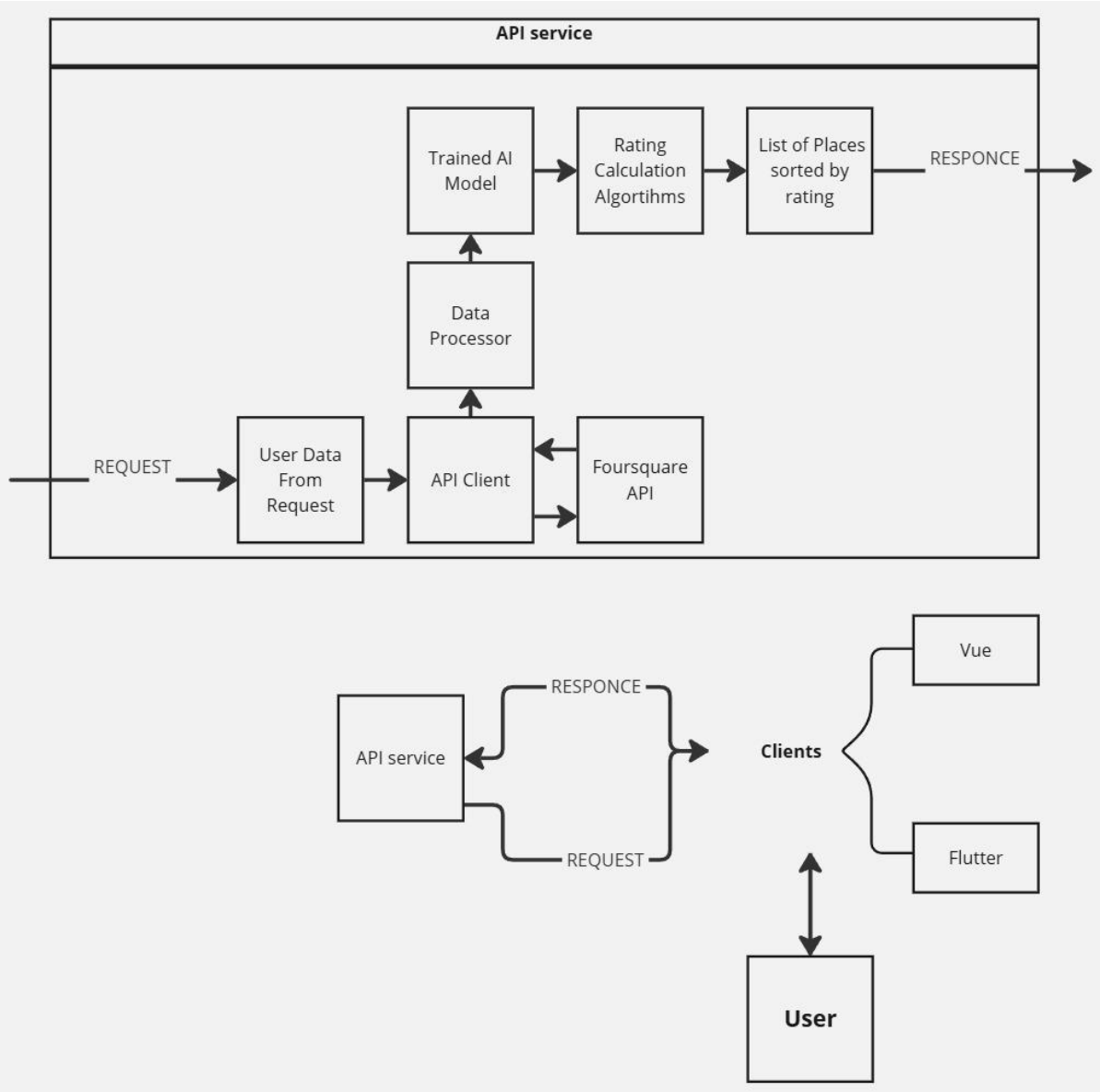



Рисунок 3.5 − Архітектура системи пошуку і взаємодії з користувачем

### 3.2 Модульна структура прототипу: реалізація та інтеграція

Реалізація прототипу базується на попередньо описаній архітектурі системи, яка складається з п’яти ключових компонентів. Основна мета прототипу – перевірити функціональність кожного з модулів і забезпечити їх інтеграцію в єдину систему.

### 3.2.1 Модуль збору місць та відгуків (Tip Collector)

На першому етапі реалізації було розроблено REST-сервіс для отримання даних із Foursquare API. Модуль забезпечує:

- Використання WebClient для здійснення запитів до API, керування ключами доступу та проксі-серверами.
- Моніторинг запитів, включаючи час їх виконання та частоту на секунду, для забезпечення стабільного доступу навіть за умов обмежень.
- Обробку помилок та логування всіх виключень для спрощення відлагодження та підвищення надійності.

На етапі прототипування основну увагу приділено налаштуванню стабільної інтеграції з Foursquare API і збереженню даних у базі даних.

### 3.2.2 Модуль створення вибірок (Requester)

Другий крок передбачає підготовку збалансованих вибірок з даних, зібраних Tip Collector. Реалізація включає:

- Інтеграцію з базою даних для автоматичного завантаження даних про місця, категорії та відгуки.
- Побудову REST-інтерфейсу для отримання готових вибірок, зокрема з врахуванням категорій та інших критеріїв.

Для спрощення тестування функціональності було створено базові API-запити, що повертають структуровані набори даних.

#### 3.2.3 Модуль обробки даних (Data Pre-Processor)

На цьому етапі модуль виконує очищення даних і підготовку їх для моделі. Реалізовано:

- Текстове очищення, включаючи видалення зайвих символів, стоп-слів та переведення тексту у нижній регістр.
- Початковий аналіз настроїв (тональності) з використанням бібліотек на основі NLP (Natural Language Processing).
- Збереження результатів обробки у базі даних для подальшого

використання.

Прототипування модуля дозволило перевірити коректність обробки текстів відгуків і швидкість виконання операцій.

### 3.2.4 Модуль тренування моделі (ModelTrainer)

Для аналізу настроїв відгуків було реалізовано прототип системи машинного навчання. Основні етапи:

1. Попередня підготовка даних, отриманих від Data Pre-Processor.
2. Навчання моделі на збалансованій вибірці, використовуючи алгоритми класифікації.
3. Оцінка точності моделі на тестовому наборі даних із застосуванням метрик, таких як Precision, Recall, F1-score.

Результатом є модель, яка успішно класифікує текст відгуків за настроєм (позитивний або негативний).

### 3.2.5 Система пошуку і взаємодії з користувачем

Для взаємодії з користувачем було реалізовано прототип інтерфейсу, що базується на Vue.js. Прототип забезпечує:

- Пошук місць за ключовими словами або категоріями.
- Відображення результатів аналізу настроїв у вигляді рейтингу місць.
- Інтеграцію із REST-інтерфейсом для отримання даних від модулів системи.

Інтерфейс дозволяє протестувати функціональність всіх компонентів системи та їх інтеграцію.

## 3.3 Процес підготовки датасетів для тренування моделі

Процес підготовки датасету є інтеграційним завданням, яке вимагає взаємодії кількох модулів системи. Основні етапи та відповідні модулі взаємодії описані нижче:

1. Підготовка середовища

Використовувані модулі:

- Модуль збору місць та відгуків (Tip Collector): виконує початкове збирання даних через Foursquare API та зберігає їх у базі даних.
- База даних: надає доступ до зібраних даних через SQL-запити.

Опис:

- Завантажуються бібліотеки для обробки текстів (pandas, nltk, transformers) та ресурси для очищення й токенізації (наприклад, stopwords і punkt).
- Налаштовується підключення до бази даних через SQLAlchemy.

2. Завантаження даних з бази даних

Використовувані модулі:

- База даних: зберігає таблиці places і tip_entity, з яких завантажуються текстові відгуки та метаінформація.
- Requester: виконує запити для отримання додаткових даних, якщо поточна вибірка є неповною.

Опис:

- Виконується SQL-запит для об’єднання таблиць places і tip_entity.
- Отримані дані перетворюються у DataFrame для обробки та перевірки на повноту вибірки.

3. Попередня обробка текстів

Використовувані модулі:

- Модуль обробки даних (Data Pre-Processor): забезпечує очищення текстів, видалення стоп-слів, токенізацію та формування готових текстів для аналізу.

Опис:

- Тексти розбиваються на слова за допомогою токенізатора.
- Видаляються зайві символи, стоп-слова, всі слова переводяться в нижній регістр.
- Оброблені дані повертаються у вигляді очищених текстів.

4. Аналіз настроїв

Використовувані модулі:

- Модуль тренування моделі (ModelTrainer): виконує аналіз настроїв за допомогою попередньо натренованої моделі.
- Data Pre-Processor: підготовлені тексти передаються для аналізу настроїв.

Опис:

- Для кожного тексту виконується аналіз настроїв (класифікація на позитивний або негативний із рівнем впевненості).
- Результати (мітки настроїв і впевненість) додаються до очищеного датасету.

5. Балансування даних та збір додаткових відгуків

Використовувані модулі:

- Requester: отримує додаткові дані через REST-запити, якщо вибірка є неповною або незбалансованою.
- Tip Collector: виконує збирання відгуків у випадку додаткових запитів.

Опис:

- Вибірка перевіряється на повноту за двома критеріями: категорії місць та збалансованість настроїв (позитивний/негативний).
- У разі недостатньої кількості даних для окремих категорій, автоматично запускається додаткове збирання через REST-запити.

6. Збереження результатів

Використовувані модулі:

- База даних: зберігає оброблені тексти та мітки настроїв у новій таблиці.
- Requester: створює запити для передачі оброблених даних до бази.

Опис:

- Очищені тексти та результати аналізу зберігаються у CSV-файл для подальшого використання.
- Підготовлені дані дублюються в таблиці оброблених відгуків, що дозволяє швидко збирати готові вибірки без необхідності повторного аналізу.

## 3.4 Використання DistilBERT для формування векторних уявлень та подальшого моделювання

Для тренування моделі, був створений датасет, який схожий на SST-2 але містить в собі відгуки підготовлені модулями Requester та Data-Pre-Processor розмічену вибірку, що базується на попередньо зібраних відгуках про популярні місця (наприклад, кафе, ресторани або туристичні локації). Кожен відгук був проаналізований і класифікований за тими ж категоріями: позитивний або негативний.

Для створення фінальної моделі, ми будемо використовувати дві моделі:

1. DistilBERT:

Роль: Обробка вхідних пропозицій та виділення ознак.

Опис: DistilBERT — це зменшена версія моделі BERT, розроблена командою HuggingFace та викладена в відкритий доступ. Вона швидша та легша за свого попередника, але при цьому зберігає високу точність у вирішенні задач, таких як класифікація тональності тексту.

Вхідні дані: Пропозиція, представлена текстом.

Вихідні дані: Вектор ознак розмірністю 768, який є ембеддингом пропозиції, на основі якого здійснюється класифікація.

2. Логістична регресія:

Роль: Класифікація тональності пропозиції.

Опис: Після того, як DistilBERT обробить пропозицію та виділить з неї ембеддинг, цей вектор передається до базової моделі логістичної регресії, що використовує бібліотеку scikit-learn. Логістична регресія класифікує ембеддинг як позитивний (1) або негативний (0) залежно від тональності пропозиції.

Вхідні дані: Вектор ембеддингу розмірністю 768, отриманий від DistilBERT.

Вихідні дані: Класифікація — або 1 (позитивне пропозиція), або 0 (негативне пропозиція).

Процес передачі даних між моделями

Після обробки пропозиції моделлю DistilBERT, вона перетворює текст у вектор ознак розмірністю 768.

Цей вектор передається в модель логістичної регресії, яка виконує класифікацію на основі ембеддингу та повертає кінцеве значення — 1 або 0, що вказує на тональність пропозиції.

Таким чином, DistilBERT та логістична регресія працюють у зв'язці для ефективної та швидкої класифікації текстів.

**Навчання моделі**

Незважаючи на те, що ми використовуємо дві моделі, навчанням безпосередньо займається лише логістична регресія. Що стосується DistilBERT, то ми використовуємо вже переднавчену модель, що була створена для загальних мовних завдань. Хоча ця модель не була спеціально навчена або налаштована для завдання класифікації тональності, вона володіє певними властивостями, які можна застосувати до нашої задачі.

Одним із основних переваг DistilBERT є те, що він використовує інформацію з токена [CLS] (перший токен у вхідному тексті), щоб представити загальний зміст всього речення. Цей підхід був, ймовірно, розроблений для задачі класифікації наступного речення, що є однією з цілей, для яких BERT був спочатку навчений. Саме тому DistilBERT генерує вектор, що відображає сенс всього введеного тексту, виводячи його в першу позицію (на виході з моделі).

Враховуючи ці характеристики, ми можемо використовувати вихід по токену [CLS] для отримання корисної інформації про загальний зміст речення, що дозволяє виконувати класифікацію без необхідності додаткового тренування самого DistilBERT.

**Використання бібліотеки transformers для роботи з DistilBERT**

Бібліотека transformers надає як реалізацію DistilBERT, так і переднавчені версії цієї моделі, що дозволяє легко інтегрувати для наших задач.

**План дій виглядає наступним чином:**

Спершу ми скористаємося переднавченим DistilBERT, щоб створити ембеддинги для 2 тисяч речень.

Далі робота з DistilBERT завершується, і ми переходимо до використання Scikit Learn. На цьому етапі набір даних розбивається на навчальну та тестову вибірки:

- Розділення результатів роботи DistilBERT (модель №1) створює набори даних для тренування і оцінки логістичної регресії (модель №2). Важливо зазначити, що Scikit Learn автоматично перемішує дані перед поділом, а не просто бере перші 75% прикладів у порядку їхнього зберігання в початковому наборі.
- На навчальній вибірці ми навчаємо логістичну регресію:

**Як обчислюється прогнозоване значення?**

Наприклад, спробуємо класифікувати речення:

«great place to drink some coffee».

- Крок 1: Токенізація

Спершу використовуємо токенайзер BERT, щоб розбити речення на токени. Потім додаємо спеціальні токени, необхідні для класифікації: [CLS] на початку і [SEP] в кінці речення.

- Крок 2: Перетворення токенів у ідентифікатори

На цьому етапі кожен токен замінюється на його ідентифікатор з таблиці ембеддингів, що надається разом із переднавченою моделлю.

**Токенізація в одній команді**

Усі три дії токенізатор виконує однією стрічкою коду:

*tokenizer.encode("great place to drink some coffee", add_special_tokens=True)*

Тепер наше вхідне речення має потрібну форму для обробки DistilBERT'ом.

**Обробка в DistilBERT**

При проходженні через DistilBERT порядок обробки вхідного вектора ідентичний до стандартного BERT. На виході кожен токен матиме відповідний вектор, який складається з 768 чисел із плаваючою точкою.

**Використання [CLS]-токена**

Оскільки наша задача — класифікація речень, ми ігноруємо всі інші вектори, крім першого (асоційованого з токеном [CLS]). Саме цей вектор передається на вхід моделі логістичної регресії.

**Завершальна класифікація**

На цьому етапі роботу завершує логістична регресія, яка отримує вектор і на основі здобутих під час навчання знань класифікує його як позитивний (1) або негативний (0).

Процес класифікації можна уявити так:


Рисунок 3.6 – Процес класифікації відгуку

**Реалізація**

Цей блок імпортує необхідні бібліотеки, зокрема Transformers для роботи з моделями, scikit-learn для класифікації та pandas для обробки даних.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV, cross_val_score
import torch
import transformers as ppb
import warnings
warnings.filterwarnings('ignore')
```

Рисунок 3.7 – Список імпортів

**Завантаження датасету**

Використовуємо pandas для завантаження нашого датасету:

```
df = pd.read_csv('train-data/place-tip-9d87y-wejdi-ew9js-vidjr-wewq2', delimiter='\t', header=None)
```

Рисунок 3.8 – Завантаження датасету

Для продуктивності беремо лише перші 2,000 речень:

```
batch_1 = df[:2000]
```

Рисунок 3.9 – Вибір 2000 відгуків

**Розподіл міток у вибірці:**

```
batch_1[1].value_counts()
```

Рисунок 3.10 – Розподіл міток у вибірці

**Вихід:**

1 (позитивні): 1041

0 (негативні): 959

Завантаження попередньо налаштованих DistilBERT моделі та токенайзеру

```
# Load DistilBERT:
model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

Рисунок 3.11 – Ініціалізація моделі і токенайзеру

**Попередня обробка даних**

Перед подачею тексту в BERT необхідно підготувати дані.

- Токенізація

Розбиваємо речення на токени та додаємо спеціальні токени [CLS] і [SEP]:

```
# Tokenize
tokenized = batch_1[0].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))
```

Рисунок 3.12 – Розбиття тексту на токени

- Паддінг

Приводимо всі речення до однакової довжини шляхом додавання нулів:

```
# Padding
max_len = max([len(i) for i in tokenized.values])
padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
```

Рисунок 3.13 – Приведення речення до однакової довжини

- Маскування

Додаємо attention mask, щоб модель ігнорувала додані нулі:

```
# Masking
attention_mask = np.where(padded != 0, 1, 0)
```

Рисунок 3.14 − Додавання attention-mask

**Запуск моделі DistilBERT**

Тепер, коли модель та вхідні дані готові, можна виконати обчислення.

Конвертуємо дані в тензори:

```
input_ids = torch.tensor(padded)
attention_mask = torch.tensor(attention_mask)
```

Рисунок 3.15 − Перетворення даних у тензори

Обробка даних моделлю

Запускаємо модель у режимі без градієнтів:

```
with torch.no_grad():
    last_hidden_states = model(input_ids, attention_mask=attention_mask)
```

Рисунок 3.16 − Запуск моделі у режимі “без градієнтів”

Результат: last_hidden_states містить вихідні значення для кожного токена.

**Отримання ознак для класифікації**

Для класифікації використовується вектор, що відповідає спеціальному токену [CLS].

Виділення векторів [CLS]

Зберігаємо перший вектор для кожного речення:

```
features = last_hidden_states[0][:, 0, :].numpy()
```

Рисунок 3.17 − Збереження першого (CLS) вектору

Результат: features — матриця ознак, яку використовуватиме модель логістичної регресії.

**Збереження міток класів**

Зберігаємо мітки (позитивне/негативне):

```
labels = batch_1[1]
```

Рисунок 3.18 − Збереження міток настрою

**Навчання моделі логістичної регресії**

Розділення даних на тренувальну та тестову вибірки

Щоб оцінити якість моделі, спершу ділимо дані:

```
train_features, test_features, train_labels, test_labels = train_test_split(features, labels)
```

Рисунок 3.19 − Розділення даних на вибірки

**Пошук оптимальних параметрів моделі (опціонально)**

Можна використовувати метод Grid Search, щоб знайти найкраще значення параметра регулярізації C, який визначає силу регулювання.

Приклад налаштування:

```
parameters = {'C': np.linspace(0.0001, 100, 20)}
grid_search = GridSearchCV(LogisticRegression(), parameters)
grid_search.fit(train_features, train_labels)

print('best parameters: ', grid_search.best_params_)
print('best scores: ', grid_search.best_score_)
```

Рисунок 3.20 − Налаштування параметру регулярізації

**Результат**: знаходиться оптимальне значення C.

**Навчання моделі логістичної регресії**

Навчання виконується за замовчуванням або із заданим значенням C:

```
lr_clf = LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, l1_ratio=None, max_iter=100,
                            multi_class='warn', n_jobs=None, penalty='l2',
                            random_state=None, solver='warn', tol=0.0001, verbose=0,
                            warm_start=False)

lr_clf.fit(train_features, train_labels)
```

Рисунок 3.21 − Ініціалізація та навчання моделі логістичної регресії

**Оцінка моделі логістичної регресії**

Точність моделі на тестовому наборі

1. Перевіряємо, наскільки модель правильно класифікує речення:

```
accuracy = lr_clf.score(test_features, test_labels)
print("Model accuracy:", accuracy)
```

Рисунок 3.22 − Показник точності навченої моделі

Результат: Точність моделі: 0.824 (82.4%).

2. Порівняння з базовим (Dummy) класифікатором

Для оцінки моделі порівнюємо її з простим Dummy-класифікатором:

```
clf = DummyClassifier()

scores = cross_val_score(clf, train_features, train_labels)
print("Dummy classifier score: %0.3f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Рисунок 3.23 − Показник точності Dummy-класифікатора

Результат Dummy-класифікатора: Середня точність: 0.527 (52.7%).

Модель логістичної регресії значно перевершує базовий класифікатор.

3. Порівняння з найкращими моделями

Для набору даних SST2 (наш датасет майже ідентичний по структурі) найкращі моделі показують наступні результати:

Fine-tuned DistilBERT: 90.7% точності.

Повнорозмірний BERT: 94.9% точності.

Максимальна точність для SST2: 96.8%.

## 3.5 Система пошуку і взаємодії з користувачем

Система пошуку і взаємодії з користувачем є ключовим компонентом, що забезпечує ефективний зв’язок між користувачами та функціональністю системи. Головна мета цього модуля – надати зручний інтерфейс для отримання релевантної інформації про місця, їхні характеристики та аналіз настроїв відгуків. Інтеграція

системи з попередньо натренованою моделлю дозволяє отримувати персоналізовані результати у реальному часі.

Опис роботи системи

Система пошуку включає кілька послідовних етапів:

1. Введення параметрів пошуку

- Користувач починає роботу із введення запиту через інтерактивний клієнтський інтерфейс. Функціональність включає:

2. Вибір місця пошуку за допомогою інтерактивної мапи.

3. Введення ключових слів для деталізації запиту.

4. Налаштування радіусу пошуку для визначення просторових меж.

5. Обробка запиту на бекенді

- Коли користувач завершує формування запиту, фронтенд передає його параметри на бекенд через API. Бекенд виконує такі дії:

6. Перевіряє наявність відповідних даних у кеші для швидкого доступу.

- У разі відсутності інформації звертається до моделі аналізу настроїв для формування нових результатів.

- Модуль аналізу популярності місць використовує алгоритми для обробки відгуків та визначення рейтингу місця. Включає такі етапи:

7. Збір відгуків: отримує всі доступні відгуки користувачів для кожного місця, зокрема кількість та зміст відгуків.

8. Аналіз настроїв: за допомогою моделей обробки природної мови визначається позитивність чи негативність відгуку.

9. Визначення релевантності відгуків: зважаючи на контекст кожного відгуку, видаляються спам або накручені коментарі. Для цього використовуються спеціальні моделі машинного навчання, що аналізують поведінку користувачів (наприклад, частота та характер відгуків).

10. Обчислення рейтингу: кожному місцю надається рейтинг, що залежить від позитивних/негативних відгуків, їх релевантності та кількості. Рейтинг можна порахувати як зважену суму позитивних і негативних відгуків, скориговану на фактор накрутки.

11. Обчислення впевненості в рейтингу: враховується кількість відгуків (чим більше відгуків, тим вище впевненість), а також їхня різноманітність і баланс між позитивними та негативними.

12. Отримані результати зберігаються у базу даних для майбутнього використання, що підвищує продуктивність.

Повернення результатів

Бекенд передає оброблену інформацію на фронтенд, де результати представлені у зручному вигляді:

- Список місць з основними характеристиками.
- Інтерактивна мапа для візуального аналізу знайдених об’єктів.

Моніторинг використання системи

Усі запити користувачів логуються для подальшого аналізу. Статистичні дані дозволяють:

- Визначити популярні категорії запитів.
- Оптимізувати роботу системи, враховуючи особливості поведінки користувачів.

**Фронтенд-інтерфейс**

Інтерфейс користувача побудований на основі сучасних веб-технологій і забезпечує інтуїтивно зрозумілу взаємодію. Основні елементи:

- Поле пошуку для введення запиту.
- Інтерактивна мапа з відображенням результатів пошуку.
- Панель фільтрів для уточнення пошуку за категоріями.

*Рисунок 4.23* демонструє прототип клієнтського інтерфейсу із зазначеними елементами.

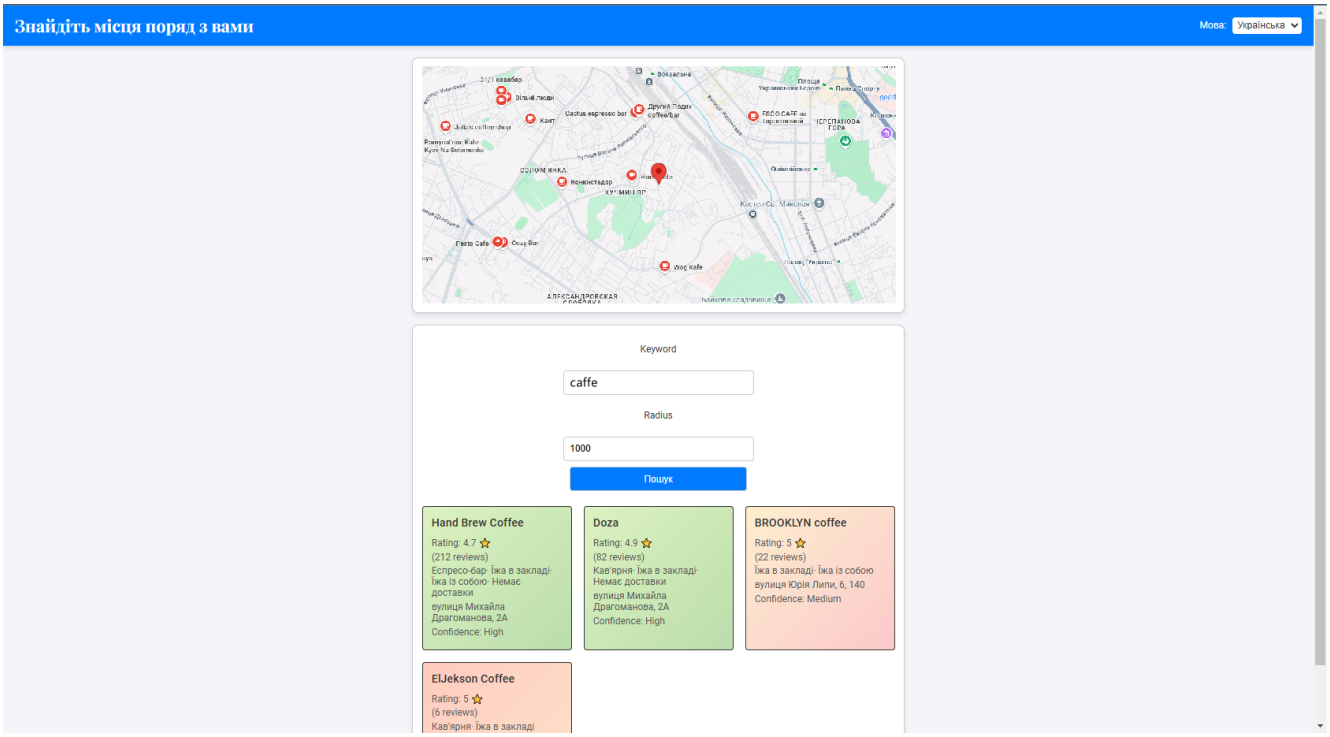



Рисунок 3.24 – Прототип клієнтського інтерфейсу

## 3.6 Збір аналітичних даних

Система забезпечує можливість автоматичного збору, обробки та аналізу аналітичних даних для підвищення ефективності пошуку і персоналізації взаємодії з користувачем. Основна мета збору даних — моніторинг активності користувачів, оцінка продуктивності системи та покращення моделі аналізу настроїв.

Джерела аналітичних даних:

1. Користувацька активність

- Логування параметрів запитів, зокрема ключових слів, радіусу пошуку та вибраних категорій.

- Аналіз популярності різних типів місць (наприклад, ресторанів чи розважальних закладів).

2. Результати пошуку

- Кількість знайдених об’єктів та їх відповідність запитам.

- Частота використання інтерактивної мапи для уточнення результатів.

3. Взаємодія з результатами

- Натискання на об'єкти в списку результатів або на мапі.

- Оцінка релевантності отриманих даних через механізм зворотного зв’язку.

4. Продуктивність моделі

- Відслідковування точності аналізу настроїв на основі зібраних відгуків користувачів.
- Моніторинг часу обробки запитів і навантаження на систему.

Методи збору та обробки:

- Логування запитів

Усі параметри запитів та результати пошуку записуються у спеціалізовану базу даних для аналітики.

- Кешування даних

Використання кешу для збереження результатів частих запитів, що прискорює їх обробку.

- Інструменти візуалізації

Зібрані дані аналізуються за допомогою дашбордів, що дозволяють визначати ключові тренди у використанні системи.

Використання аналітичних даних:

1. Оптимізація пошукової системи

- Покращення алгоритмів пошуку для підвищення точності та релевантності результатів.
- Балансування навантаження на сервер шляхом передбачення пікових періодів активності.

2. Адаптація моделі настроїв

- Тренування моделі на оновлених даних для підвищення точності аналізу.
- Автоматичне виявлення нових шаблонів у текстах відгуків.

3. Підтримка бізнес-рішень

- Оцінка популярності закладів і категорій для визначення ключових напрямків розвитку.
- Надання статистичних даних для маркетингових досліджень.

Збір і аналіз аналітичних даних дозволяє створювати динамічну систему, яка адаптується до потреб користувачів, забезпечуючи їх актуальною інформацією та високою якістю сервісу.

### 3.7 Візуалізація зібраних даних

Зібрані дані про місця та відгуки відкривають широкі можливості для різних видів аналізу. Існує безліч способів обробки і вивчення цих даних — від базових статистичних розподілів до складних геопросторових моделей. Ви можете зосередитися на аналізі популярності місць, якості відгуків, їхній залежності від географічного розташування або часу.

Враховуючи різноманітність доступних підходів, я вирішив продемонструвати лише кілька основних методів візуалізації, які, на мою думку, надають найбільше інформації для подальшого розуміння поведінки користувачів і популярності різних локацій. Зокрема, це теплові карти для візуалізації концентрації відгуків і скаттер-плоти для розподілу відгуків за категоріями місць.

Ці методи дозволяють виокремити основні тренди, виявити зони з найбільшим інтересом та зрозуміти, як різні параметри впливають на популярність місць. Хоча існує багато інших способів вивчення цих даних, я обрав саме ці для більш наочної демонстрації можливостей візуалізації.

Графік на *рисунку 4.24* демонструє розподіл кількості слів у відгуках на основі зібраних даних. Використовуючи зібрані дані, можна аналізувати різноманітні аспекти відгуків, зокрема їх довжину, що дає змогу робити висновки щодо детальності та глибини відгуків.

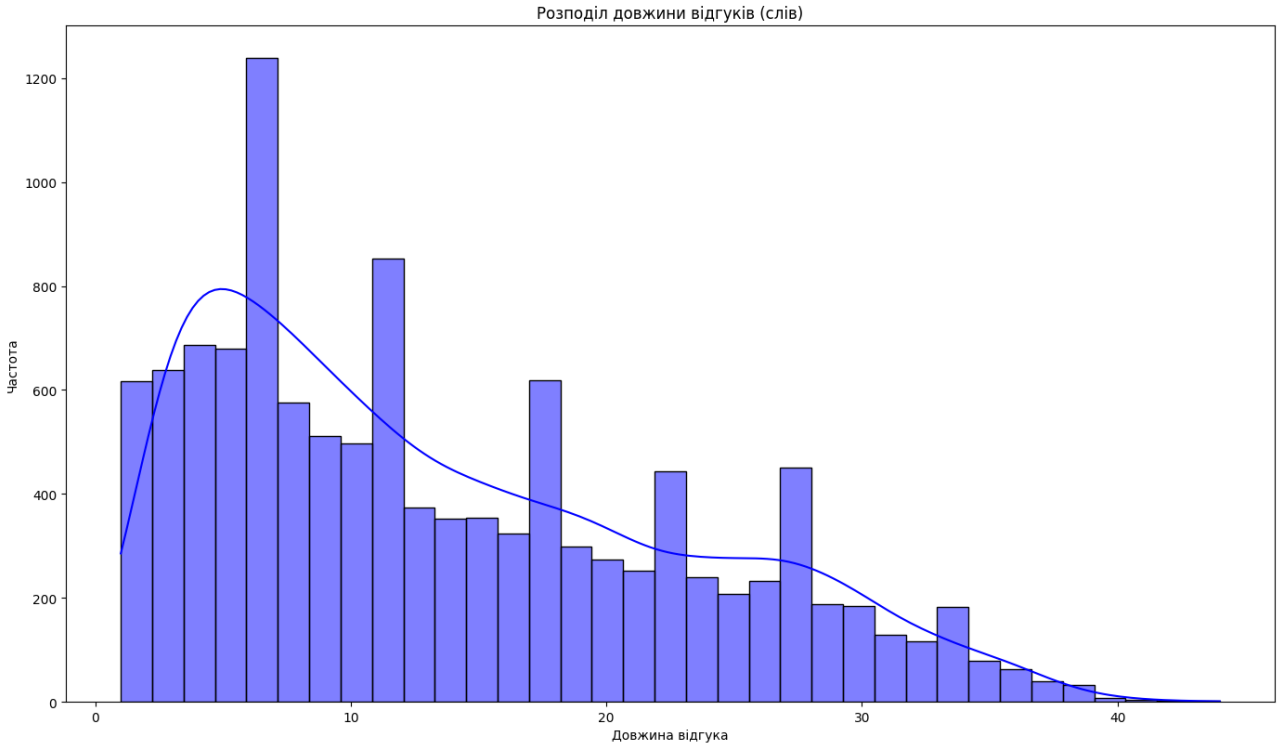



Рисунок 3.25 − Графік розподілу кількості слів у відгуках

Розподіл кількості слів у відгуках може показувати, що певна довжина зустрічається частіше через стандартні формати відгуків, обмеження на кількість слів у платформах або через звичку користувачів писати короткі та чіткі коментарі. Така закономірність може також залежати від типу відгуку — позитивні можуть бути коротшими, тоді як детальні негативні і нейтральні відгуки зазвичай довші.

Теплова мапа *рисунку 4.25* показує розподіл популярних місць у радіусі 2 км від конкретних точок. На основі зібраних даних можна створювати різноманітні карти, де інтенсивність кольорів на мапі відповідає кількості відгуків або їх якості, що дозволяє вивчати популярність різних локацій. Відгуки на *рисунку 4.25* були відфільтровані за останній тиждень.

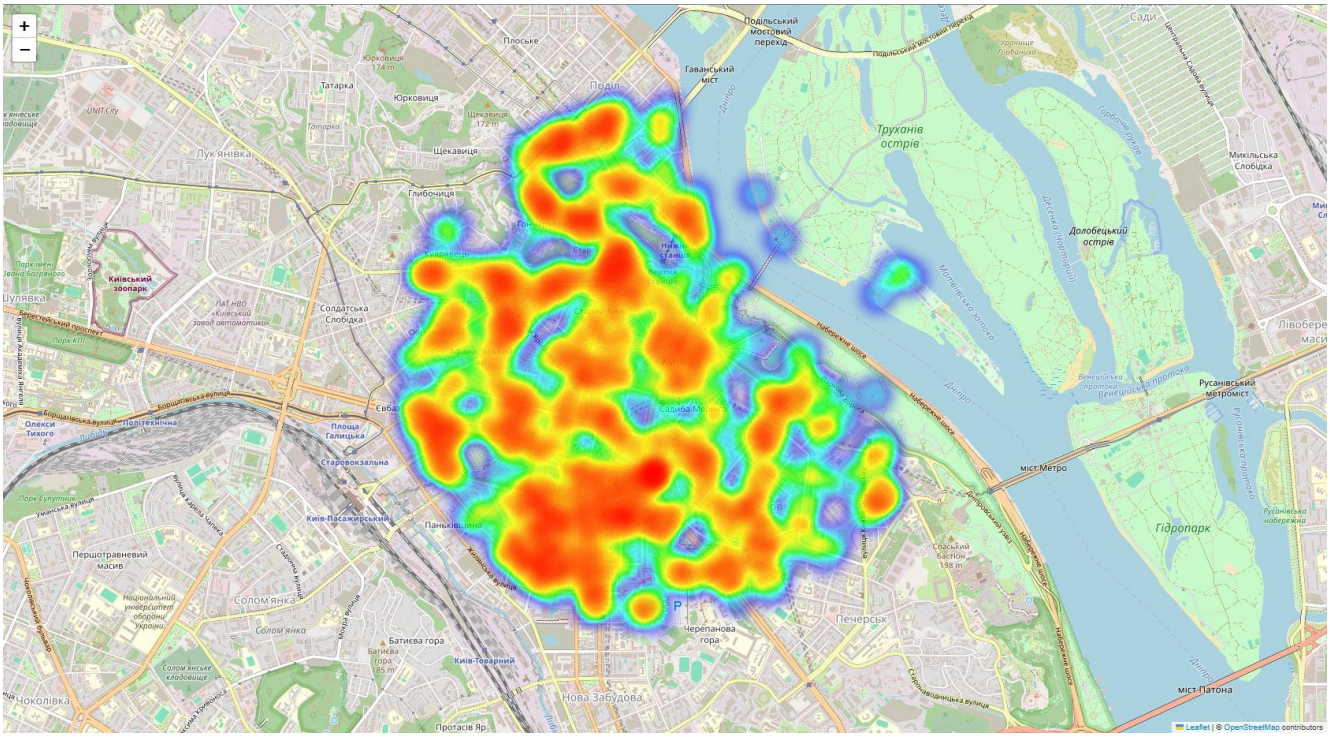



Рисунок 3.26 − Теплова мапа популярних місць радіусом 2 км.

Це наближена версія попередньої теплової мапи, що дозволяє краще побачити точне розташування популярних місць в межах 2 км. Зібрані дані можуть бути використані для подальшого аналізу з різними параметрами, такими як відстань, якість відгуків та частота їх написання.

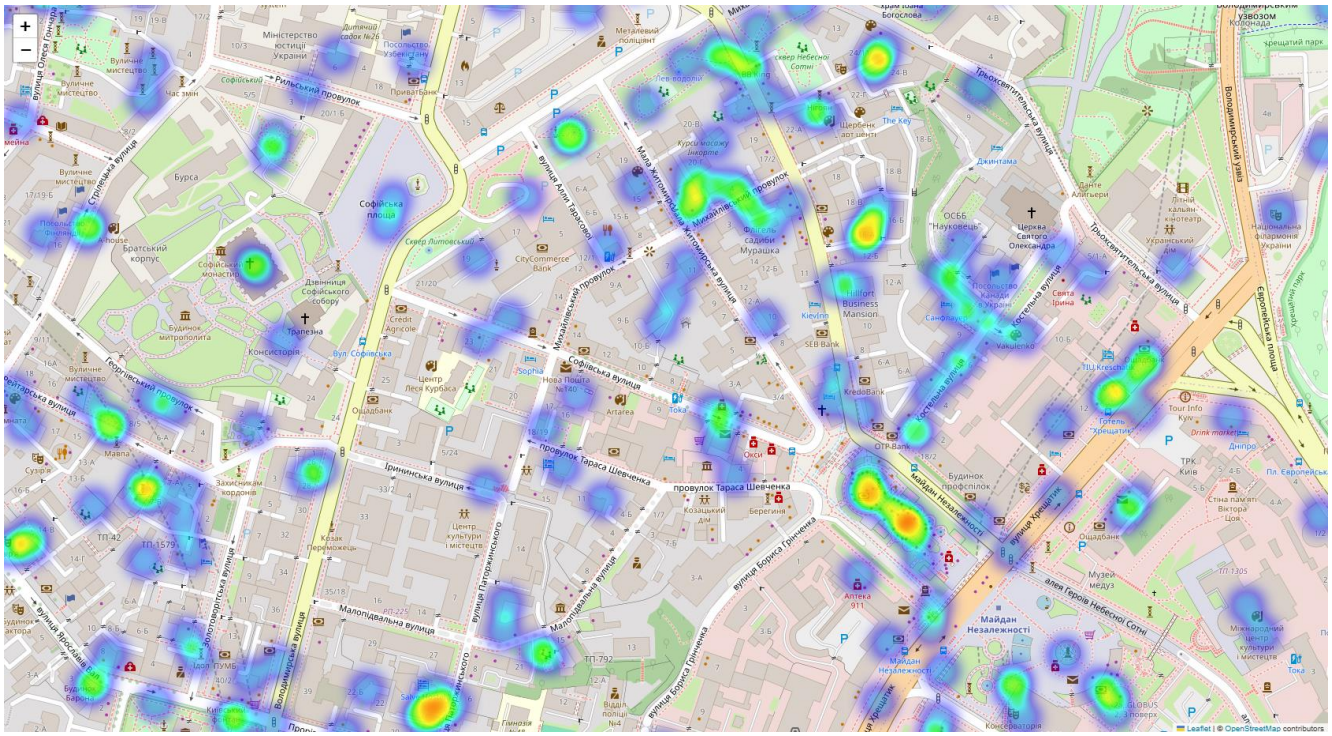



Рисунок 4.27 − Теплова мапа популярних місць радіусом 2 км (приближено).

Висновок, що популярні та відомі туристичні місця отримують більше відгуків, можна зробити на основі зібраних даних. Зазвичай такі місця привертають більше уваги від туристів, що призводить до більшої кількості відгуків. Це може бути обумовлено високою відвідуваністю, великою кількістю туристичних ресурсів, що рекламують ці місця, а також зростаючим інтересом до них у соціальних мережах та інших платформах. Відгуки можуть служити важливим індикатором популярності і допомагати іншим туристам у виборі місць для відвідування.

На основі зібраних даних можна побудувати граф, де вузли представляють туристичні місця, а ребра відображають взаємодії між ними, наприклад, схожість у відгуках чи географічну близькість. Така структура дозволяє візуалізувати популярність місць та їх взаємозв'язки. Важливим є те, що на основі цього графа можна застосовувати різні моделі аналізу даних для виявлення найбільш бажаних місць, зокрема за кількістю відгуків, рейтингами або іншими метриками популярності. Моделі можуть порівнювати ці місця, оцінюючи, які з них приваблюють більше уваги користувачів та мають більший потенціал для залучення нових відвідувачів. Таким чином, граф може стати потужним інструментом для аналізу та прогнозування популярності туристичних місць.

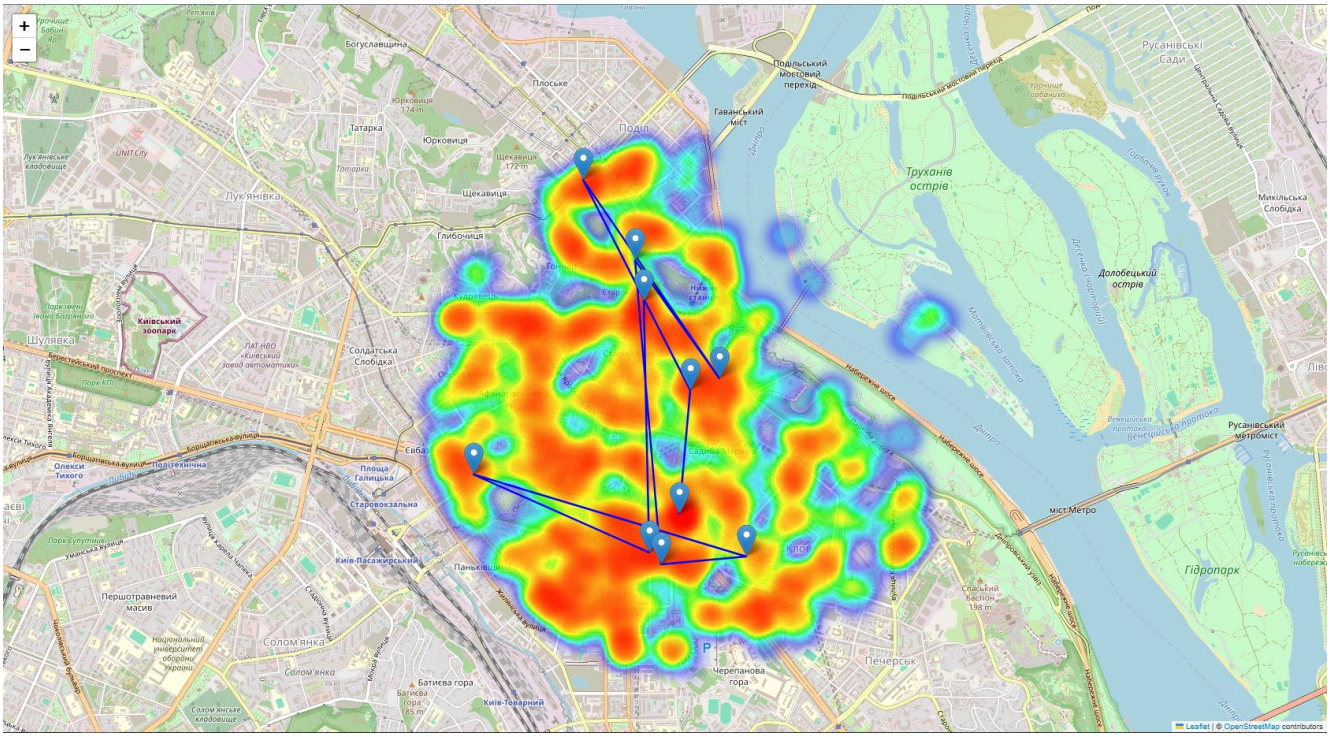



Рисунок 4.27 − Граф найпопулярніших місць.

Аналогічно, саме таким чином наразі працює алгоритм визначення найпопулярніших місць у розділі 3.1. Використовуючи зібрані дані, алгоритм створює граф, де кожне туристичне місце є вузлом, а взаємодії між ними (наприклад, схожість за відгуками або географічна близькість) формують ребра. Алгоритм аналізує цей граф, виводячи найбільш популярні місця за кількістю відгуків, рейтингами або іншими характеристиками, що вказують на їх привабливість для відвідувачів. Цей підхід дозволяє автоматично визначати найбільш бажані місця, використовуючи модель для порівняння різних туристичних точок, оцінюючи їх на основі числових показників та зв'язків між ними в графі.

# РОЗДІЛ 4. ПЕРЕВАГИ ТА НЕДОЛІКИ РОЗРОБЛЕНОГО МЕТОДУ

Розробка запропонованої системи зіткнулася з низкою труднощів, зокрема потребою у високій точності аналізу даних та забезпеченні ефективної взаємодії між її компонентами. Однак у процесі створення вдалося закласти гнучку архітектуру, яка дозволяє легко адаптувати та масштабувати систему відповідно до зростання її функціональних вимог або обсягів оброблюваних даних.

## 4.1 Переваги

1. Висока точність аналізу настроїв**.** Система використовує сучасні алгоритми машинного навчання, включаючи методи глибокого навчання та обробки природної мови (NLP), які дозволяють ефективно ідентифікувати емоційний стан користувачів навіть у великих масивах текстових даних.

2. Автоматизація збору та обробки даних**.** Система самостійно виконує всі етапи обробки даних:

- Збір даних з джерел для навчання
- Попередня обробка, включаючи фільтрацію шуму та видалення дублікатів.
- Аналіз настроїв і збереження результатів в базі даних.

3. Швидкість роботи через кешування**.** Завдяки використанню кешу система зберігає результати для популярних або повторюваних запитів, що дозволяє миттєво видавати відповіді без повторного аналізу. Це особливо корисно для даних з високою частотою використання.

4. Гнучкість та масштабованість**.** Система створена таким чином, щоб легко адаптуватися до змін у бізнес-логіці чи зростання обсягів даних. Наприклад, можна додавати нові модулі для аналізу специфічних типів даних або збільшувати кількість серверів для обробки.

5. Інтеграція з різними платформами**.** Система підтримує API, які забезпечують інтеграцію з іншими сервісами, такими як CRM-системи, платформи аналітики чи мобільні додатки. Це дозволяє використовувати результати аналізу

для прийняття рішень у реальному часі.

6. Рекомендаційна система**.** На основі аналізу настроїв і поведінкових даних система може пропонувати персоналізовані рекомендації, які підвищують залученість користувачів і поліпшують їхній досвід.

## 4.2 Обмеження та недоліки

1. Залежність від зовнішніх сервісів. Система може використовувати сторонні API для збору даних або обробки, наприклад, перекладачі чи сервіси текстової аналітики. Якщо ці сервіси недоступні або працюють нестабільно, це може вплинути на продуктивність системи.

2. Обмеження моделі аналізу настроїв. Незважаючи на сучасність моделей, вони можуть помилятися у випадках:

- Складних метафор або сарказму.
- Контексту, який залежить від поза текстової інформації.

3. Чутливість до якості даних. Неякісні або неповні дані, такі як спам, орфографічні помилки чи неоднозначний текст, можуть призвести до неправильного аналізу.

4. Затримки при великому обсязі даних. Під час обробки дуже великих наборів даних, наприклад, мільйонів твітів за короткий проміжок часу, можуть виникати затримки через обмеження ресурсів або недостатню оптимізацію алгоритмів.

5. Високі вимоги до ресурсів. Для роботи системи потрібні потужні сервери або хмарні обчислювальні платформи, що може збільшити витрати на інфраструктуру.

## 4.3 Шляхи вдосконалення

**1.** Оптимізація моделей

- Покращення існуючих моделей за рахунок їх донавчання на спеціалізованих наборах даних.
- Використання методів трансферного навчання для зменшення вимог до

обсягу даних для навчання.

**2.** Розширення підтримки мов. Інтеграція нових мов дозволить системі працювати з більшою аудиторією. Це може включати використання багатомовних моделей, таких як mBERT чи GPT-4.

3. Покращення механізму кешування**.** Інтелектуальне кешування дозволить передбачати, які дані будуть найбільш затребуваними, і зберігати їх наперед. Це зменшить кількість запитів до основної системи обробки.

4. Децентралізація обробки даних**.** Розподіл обчислювальних задач між декількома серверами або використання хмарних обчислювальних сервісів для забезпечення швидкості роботи навіть при великих обсягах даних.

5. Покращення інтерфейсу користувача**.** Створення адаптивного дизайну, який дозволяє легко знаходити потрібну інформацію та взаємодіяти з системою. Наприклад, інтерактивні дашборди для візуалізації результатів аналізу настроїв.

Ці заходи допоможуть не тільки усунути недоліки, але й значно розширити функціональність та ефективність системи.

# ВИСНОВКИ

У цій роботі було розроблено комплексний підхід до збору, обробки, аналізу та візуалізації даних про місця, базуючись на настроях користувацьких відгуків. Ключові етапи включають отримання даних через Foursquare API, формування збалансованих вибірок, очищення текстів, їх підготовку до аналізу та використання моделей машинного навчання для класифікації відгуків як позитивних чи негативних. Уся система спрямована на забезпечення точності результатів, стабільності та гнучкості у використанні.

Завдяки застосованим методам вдалося досягти високої ефективності у виявленні настроїв, що дозволяє отримувати інсайти для різноманітних бізнес-задач. Інтеграція з іншими платформами, зручний користувацький інтерфейс та адаптивність до різних сценаріїв використання розширити сферу застосування розробленого рішення.

Для подальшого вдосконалення існують перспективи оптимізації обробки даних, що дозволить прискорити роботу з великими обсягами інформації, а також інтеграції багатомовних моделей аналізу, що розширить аудиторію системи на глобальному рівні. Покращення користувацького досвіду через створення більш інтуїтивного інтерфейсу та впровадження розподілених обчислювальних систем дозволить масштабувати роботу для великих проектів.

# ПЕРЕЛІК ПОСИЛАНЬ


1. *Kotlin. [Електронний ресурс]. – Режим доступу: https://kotlinlang.org/.*
2. *Kotlin Coroutines: kotlinx-coroutines-reactor.*
3. *Foursquare API. [Електронний ресурс]. – Режим доступу: https://foursquare.com/.*
4. *Google Places API. Overview | Places API | Google for Developers.*
5. *DistilBERT: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.*
6. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*
7. *DistilBERT base-uncased. [Електронний ресурс]. – Режим доступу: https://huggingface.co/distilbert-base-uncased.*
8. *Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.*
9. *Spring WebFlux Documentation. [Електронний ресурс]. – Режим доступу: https://docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html.*
10. *PostgreSQL Documentation. [Електронний ресурс]. – Режим доступу: https://www.postgresql.org/docs/.*
11. *JSON Format Overview. [Електронний ресурс]. – Режим доступу: https://www.json.org/.*
12. *Python Official Documentation. [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/.*
13. *Natural Language Toolkit (NLTK) Documentation. [Електронний ресурс]. – Режим доступу: https://www.nltk.org/.*
14. *pandas Documentation. [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org/.*
15. *SQLAlchemy Documentation. [Електронний ресурс]. – Режим доступу: https://docs.sqlalchemy.org/.*

16. *Hugging Face Transformers Library. [Електронний ресурс]. – Режим доступу: https://huggingface.co/docs/transformers/.*

17. *PyTorch Official Documentation. [Електронний ресурс]. – Режим доступу: https://pytorch.org/docs/.*

18. *Scikit-learn Documentation. [Електронний ресурс]. – Режим доступу: https://scikit-learn.org/stable/.*

19. *Prometheus Documentation. [Електронний ресурс]. – Режим доступу: https://prometheus.io/docs/.*

20. *Grafana Documentation. [Електронний ресурс]. – Режим доступу: https://grafana.com/docs/.*

21. *Flutter Documentation. [Електронний ресурс]. – Режим доступу: https://docs.flutter.dev/.*

22. *Vue.js Documentation. [Електронний ресурс]. – Режим доступу: https://vuejs.org/.*

23. *Nielsen Global Online Consumer Survey (2015).*

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій
Кафедра Системного Аналізу

## КВАЛІФІКАЦІЙНА РОБОТА

на тему

Методи визначення точок привабливості на основі відгуків користувачів соціальних мереж

Виконав: Студент групи САДМ-61
Мартиненко Д. О.

Керівник: Кандидат технічних наук, Доцент.
Патракеєв І. М.

### АКТУАЛЬНІСТЬ ТЕМИ

Сучасні користувачі потребують персоналізованих рекомендацій щодо популярних місць, які враховують їхні вподобання, геолокацію та актуальність відгуків. Це робить розробку таких систем важливим завданням.

### МЕТА РОБОТИ

Розробка ефективної системи для збору, аналізу даних з використанням методів визначення точок привабливості і надання рекомендацій користувачам.

02

### ОБ'ЄКТ ДОСЛІДЖЕННЯ

Соціальні мережі як джерело даних для аналізу популярності локацій та створення рекомендацій.

### ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи визначення точок привабливості на основі текстових відгуків та геолокаційної інформації, зібраної із соціальних мереж.

# ЗАВДАННЯ ДОСЛІДЖЕННЯ

01 **АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**
Визначити ключові аспекти даних для оцінки популярності.

02 **ВИБІР МЕТОДІВ ТА ТЕХНОЛОГІЙ**
Обрати інструменти для збору та обробки інформації.

03 **ОБРОБКА ДАНИХ**
Реалізувати алгоритми обробки текстів та геолокаційних даних.

04 **ЗБІР ДАНИХ**
Зібрати дані з соціальних мереж та інших джерел для тренування моделі.

05 **ТРЕНУВАННЯ МОДЕЛІ**
Виконати тренування моделі машинного навчання на зібраних даних.

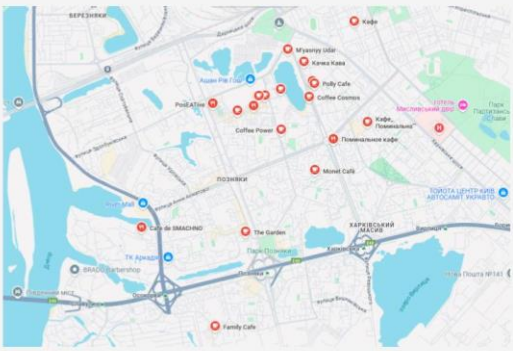
06 **АНАЛІЗ НАСТРОЇВ ТА ПРОГНОЗУВАННЯ**
Використати модель для аналізу настроїв і прогнозування популярності.

07 **РОЗРОБКА СИСТЕМИ**
Інтегрувати результати в систему для візуалізації та рекомендацій.

08 **ОЦІНКА ЕФЕКТИВНОСТІ**
Оцінити точність та продуктивність системи



# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ




**Точки привабливості**
Оцінка місць за відгуками та геолокацією.

**Вплив соціальних мереж**
Як соціальні мережі впливають на популярність місць.

**Збір даних**
Використання API для отримання даних із соціальних мереж

## ВИКОРИСТАННЯ ДАНИХ СОЦІАЛЬНИХ МЕРЕЖ

Приклади застосування

Таргетинг реклами, персоналізовані рекомендації, інтеграція з картами та геосервісами.

Роль даних у бізнес-стратегіях

Використання даних для підвищення лояльності клієнтів, покращення сервісу та розробки персоналізованих пропозицій.

Аналіз та моніторинг

Збір і використання даних для моніторингу ефективності кампаній і популярності місць.

05

## МЕТОДИ ТА ІНСТРУМЕНТИ ДОСЛІДЖЕННЯ

06

API для збору даних

01

Моделі на основі трансформерів, регресії

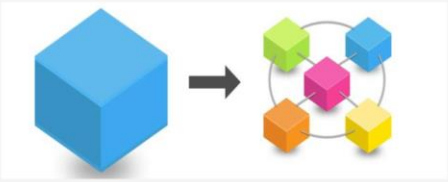
02

Інструменти розробки

03

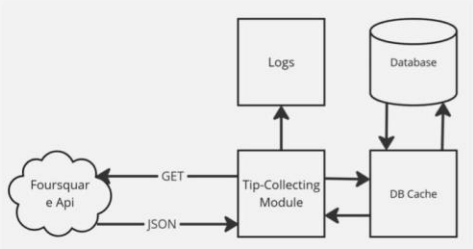
# АРХІТЕКТУРА СИСТЕМИ ТА ПРОТОТИП

Архітектура системи побудована на принципі модульності, що дозволяє розділити задачу на окремі компоненти, кожен з яких відповідає за конкретну функцію. Це забезпечує гнучкість, зручність тестування та масштабування системи.



Прототип системи реалізовано з п’яти основних модулів, кожен з яких відповідає за певний етап роботи.

07



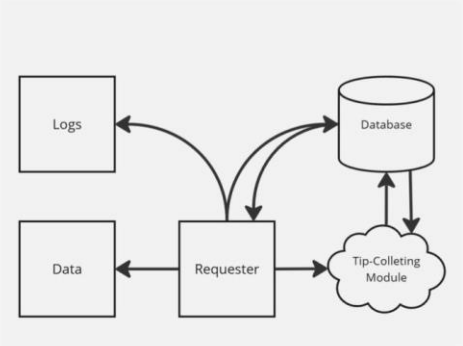

# МОДУЛЬ ЗБОРУ МІСЦЬ ТА ВІДГУКІВ (TIP COLLECTOR)

Здійснює запити до Foursquare API для отримання даних про місця та відгуки користувачів, забезпечуючи стабільний доступ до API та ведення логів для моніторингу швидкості запитів.

08

## МОДУЛЬ СТВОРЕННЯ ВИБІРОК (REQUESTER)

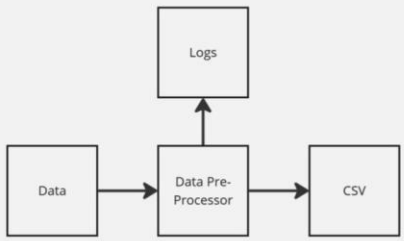
Отримує та підготовлює збалансовані вибірки з даних для подальшої обробки та аналізу, інтегрується з базами даних та REST-сервером.




09

## МОДУЛЬ ОБРОБКИ ДАНИХ (DATA PRE-PROCESSOR)

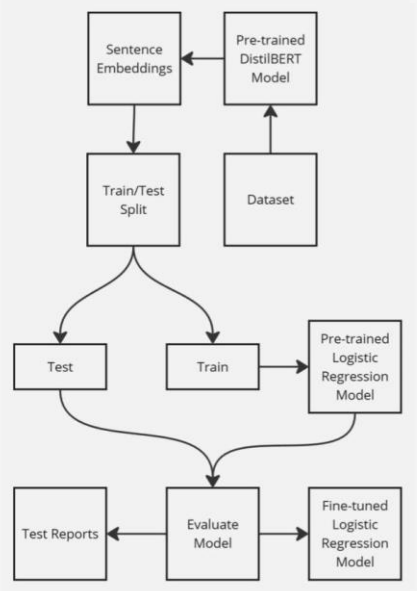
Очищає текстові відгуки, видаляючи зайві символи та стоп-слова, готує дані для навчання моделі та аналізує їх настрої.




10

## МОДУЛЬ ТРЕНУВАННЯ МОДЕЛІ (MODELTRAINER)

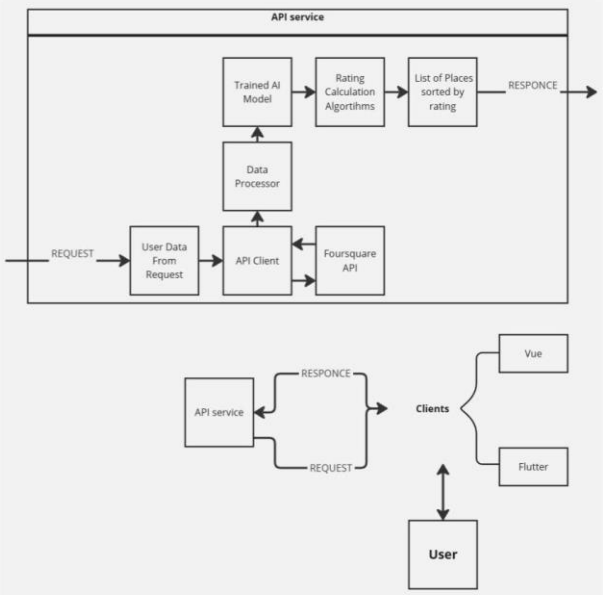
Створює модель для аналізу настроїв відгуків, визначаючи позитивну чи негативну тональність.




11

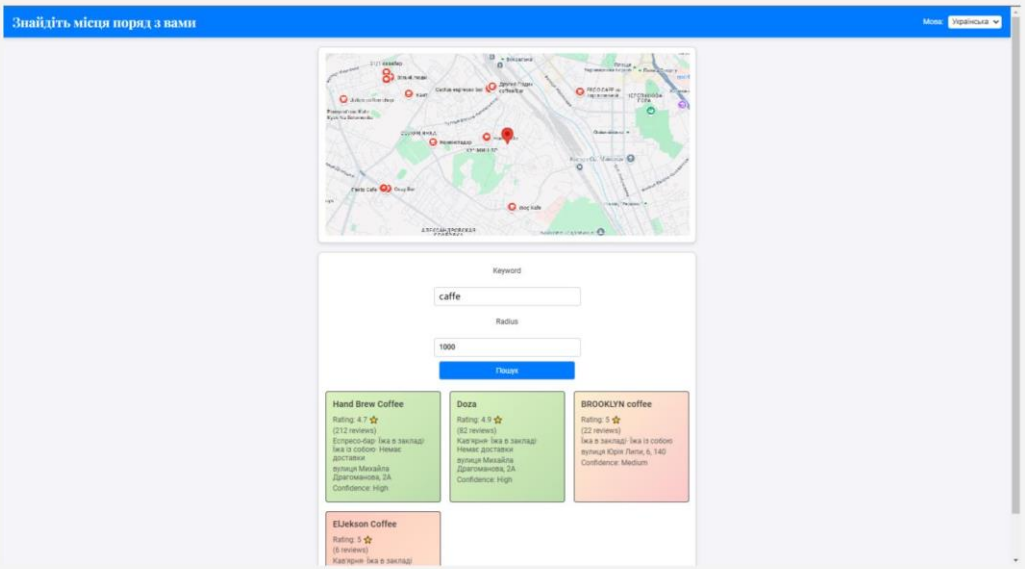
## СИСТЕМА ПОШУКУ І ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ

Надання користувачам можливості взаємодіяти з натренованою моделлю через інтерфейси на Vue.js або Flutter для пошуку найкращих місць за категоріями чи ключовими словами.



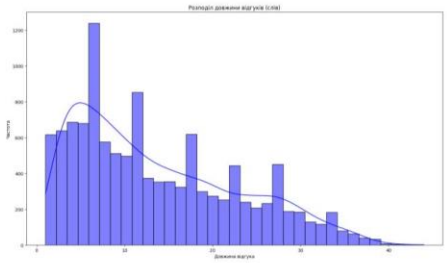

12

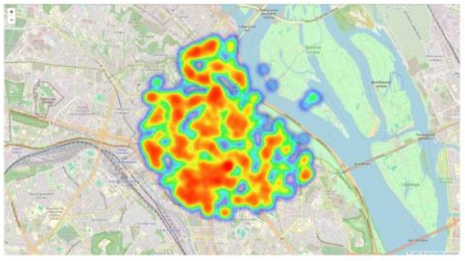
## Прототип користувацького інтерфейсу




13

## Візуалізація даних





Зібрані дані можна представляти та застосовувати багатьма способами

14

# ВИСНОВКИ

У результаті дослідження була розроблена система для збору та аналізу даних про місця на основі користувацьких відгуків. Ключовими компонентами є модулі збору даних через Foursquare API, обробки текстів, створення вибірок, тренування моделей для аналізу настроїв і взаємодія з користувачем через сучасні клієнтські технології.

Система демонструє високу точність, гнучкість і стабільність, а також здатність інтегруватися з іншими платформами та адаптуватися до різних бізнес-сценаріїв.

Для подальшого вдосконалення пропонується оптимізувати обробку даних, додати підтримку багатьох мов, поліпшити користувацький інтерфейс і розглянути можливість масштабування системи для більшої продуктивності при роботі з великими обсягами даних.