

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ**
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

«До захисту допущено»

Завідувач кафедри

_____ В. В. Койбічук

«__» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр
(бакалавр / магістр)

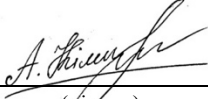
зі спеціальності 051 «Економіка»,
(код та назва)

освітньо-професійної програми «Економічна кібернетика»
(освітньо-професійної / освітньо-наукової) (назва програми)

на тему: «Розробка модуля для автоматизованого аналізу, класифікації та управління відгуками на товари за допомогою штучного інтелекту»

Здобувача (ки) групи ЕК.м-21 Кільдей Анастасії Денисівни
(шифр групи) (прізвище, ім'я, по батькові)

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



(підпис)

Анастасія Кільдей
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник
доцентка, к.е.н., В.В. Койбічук
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Суми – 2023

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувачка кафедри
доцентка, к.е.н.
_____ В. В. Койбічук
“ ___ ” _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА
спеціальність 051 Економіка (Економічна кібернетика)
студенту 2 курсу, групи ЕК.м-21

Кільдей Анастасії Денисівни

1. Тема роботи «Розробка модуля для автоматизованого аналізу, класифікації та управління відгуками на товари за допомогою штучного інтелекту» затверджена наказом Про затвердження тем і керівників кваліфікаційних робіт наказ №1331-VI від 22.11.2023 року.
 2. Термін подання студентом закінченої роботи «15» грудня 2023 року
 3. Мета кваліфікаційної роботи: полягає вивченні, розробці та вдосконаленні методів класифікації та аналізу текстової інформації з використанням штучного інтелекту
 4. Об'єкт дослідження це процеси аналізу та класифікації тексту, зокрема у відношенні до відгуків на товари.
 5. Предмет дослідження є використання нейронних мереж для задач обробки природномовних текстів.
 6. Кваліфікаційна робота виконується на наукових публікацій іноземних та вітчизняних дослідників, існуючих методологій у сфері аналізу та класифікації тексту, а також враховує новітні досягнення в галузі штучного інтелекту.
 7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети
- Розділ 1. Огляд теоретичних основ задачі класифікації тексту
У розділі 1: 1.1 Загальна характеристика аналізу та класифікації тексту.

1.2 Огляд існуючих методів аналізу та класифікації тексту.

1.3 Переваги та недоліки використання нейромереж для задач класифікації тексту.

Розділ 2. Підготовка та обробка датасету для класифікації відгуку на товар

У розділі 2: 2.1 Роль штучного інтелекту у вдосконаленні аналізу, класифікації та управління відгуками на товари.

2.2 Опис основних методів збору датасету.

2.3 Збір та обробка даних для навчання моделі.

Розділ 3. Розробка модуля для автоматизованого аналізу, класифікації та управління відгуками на товари.

У розділі 3: 3.1 Технічне завдання до вирішення поставленої проблем за допомогою машинного навчання.

3.2 Опис структури і функціоналу модуля автоматизованого аналізу, класифікації та управління відгуками на товари.

3.3 Оцінка очікуваного ефекту від впровадження модуля автоматизованої класифікації відгуку на товар.

8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Койбічук В.В., к.е.н., доцентка	30.10.2023	06.11.2023
2	Койбічук В.В., к.е.н., доцентка	09.11.2023	23.11.2023
3	Койбічук В.В., к.е.н., доцентка	24.11.2023	08.12.2023

9. Дата видачі завдання: «30» жовтня 2023 року

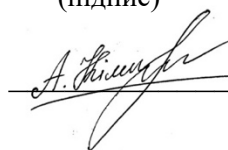
Керівник кваліфікаційної роботи _____

В. В. Койбічук

(підпис)

(ініціали, прізвище)

Завдання до виконання одержав _____



А.Д. Кільдей

АНОТАЦІЯ
кваліфікаційної роботи на тему
«РОЗРОБКА МОДУЛЯ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ,
КЛАСИФІКАЦІЇ ТА УПРАВЛІННЯ ВІДГУКАМИ НА ТОВАРИ ЗА
ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ»
студента Кільдей Анастасії Денисівни

Актуальність теми, обраної для дослідження. Зростання обсягу текстової інформації в сучасному інформаційному суспільстві ставить перед науковцями невідкладне завдання розробки ефективних методів аналізу та класифікації тексту. Проблеми, пов'язані з витягненням суттєвої інформації з текстових джерел, є важливими для розвитку сучасних технологій, особливо з використанням штучного інтелекту.

Мета кваліфікаційної магістерської роботи полягає у вивченні, розробці та вдосконаленні методів класифікації та аналізу текстової інформації з використанням штучного інтелекту, а також створення ефективної системи, яка забезпечить точні та швидкі результати в аналізі відгуків для прийняття підприємницьких рішень.

Об'єктом дослідження є процеси аналізу та класифікації тексту, зокрема у відношенні до відгуків на товари, що є значущими в сфері сучасного електронного бізнесу та споживчого ринку.

Предметом дослідження є використання нейронних мереж для задач обробки природномовних текстів. Для реалізації проекту була обрана мова програмування Python, оскільки вона найбільш зручна та найкраще пристосована для роботи із штучним інтелектом.

Основним завданням є визначення оптимальних стратегій та інструментів для ефективного аналізу тексту, забезпечуючи точну та інформативну класифікацію відгуків на товари.

У процесі дослідження були використані методи аналізу, моделювання та реалізації алгоритмів машинного навчання, зокрема нейромереж, для досягнення поставлених цілей.

Дослідження базується на використанні наукових публікацій іноземних та вітчизняних дослідників, існуючих методологій у сфері аналізу та класифікації тексту, а також враховує новітні досягнення в галузі штучного інтелекту.

Проведене дослідження передбачає створення нового підходу до класифікації та аналізу тексту, ґрунтованого на використанні штучного інтелекту.

Для досягнення поставленої мети та вирішення поставлених завдань дослідження були використані такі методи: аналіз існуючих підходів до класифікації тексту, моделювання алгоритмів машинного навчання, зокрема використання неймереж для автоматизованого аналізу тексту.

Інформаційною базою кваліфікаційної магістерської роботи є наукові публікації, методології у сфері аналізу та класифікації тексту, а також передові досягнення в галузі штучного інтелекту.

Основний науковий результат кваліфікаційної магістерської роботи полягає у вдосконаленні методів класифікації та аналізу текстової інформації з використанням штучного інтелекту, зокрема у розробці ефективної системи для аналізу відгуків на товари.

Одержані результати можуть бути використані в розробці інноваційних інструментів для аналізу великих обсягів текстової інформації, забезпечуючи точні та швидкі результати в аналізі великої кількості документів.

Результати апробації основних положень кваліфікаційної магістерської роботи розглядалися на Міжнародній науково-практичній конференції «Проблеми та перспективи розвитку фінансово-кредитної системи України» 4-5 грудня 2023 року. За результатами, було опубліковано тезу в збірнику конференції: «Кільдей А. Д., Кушнерьов О. С., Койбічук В. В. Роль та ефективність використання нейронної мережі у вдосконаленні аналізу, класифікації та управління відгуками на товари. Проблеми та перспективи розвитку фінансово-кредитної системи України: Матеріали Міжнародної

науково-практичної конференції / за заг. ред.: Л. Л. Гриценко, І. В. Тютюнник.
– Суми : Сумський державний університет, 2023. С. 36–39».

Ключові слова: аналіз відгуків, аналіз тональності, класифікація тексту, машинне навчання, нейромережі, штучний інтелект.

Зміст кваліфікаційної магістерської роботи викладено на 41 сторінках (3- 44).

Список використаних джерел із 62 найменувань, розміщений на 45-50 сторінках. Робота містить 2 таблиці, 14 рисунків, а також 4 додатки, розміщених на 51-65 сторінках.

Рік виконання кваліфікаційної роботи – 2023 рік.

Рік захисту роботи – 2023 рік

ЗМІСТ

ВСТУП.....	3
1 ОГЛЯД ТЕОРЕТИЧНИХ ОСНОВ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТУ.....	6
1.1 Загальна характеристика аналізу та класифікації тексту	6
1.2 Огляд існуючих методів аналізу та класифікації тексту	10
1.3 Переваги та недоліки використання нейромереж для задач класифікації тексту	14
2 ПІДГОТОВКА ТА ОБРОБКА ДАТАСЕТУ ДЛЯ КЛАСИФІКАЦІЇ ВІДГУКУ НА ТОВАР	17
2.1 Роль штучного інтелекту у вдосконаленні аналізу, класифікації та управління відгуками на товари.....	17
2.2 Опис основних методів збору датасету.....	23
2.3 Збір та обробка даних для навчання моделі	24
3 РОЗРОБКА МОДУЛЯ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ, КЛАСИФІКАЦІЇ ТА УПРАВЛІННЯ ВІДГУКАМИ НА ТОВАРИ.....	27
3.1 Технічне завдання до вирішення поставленої проблем за допомогою машинного навчання.....	27
3.2 Опис структури і функціоналу модуля автоматизованого аналізу, класифікації та управління відгуками на товари.....	29
3.3 Оцінка очікуваного ефекту від впровадження модуля автоматизованої класифікації відгуку на товар.....	40
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТКИ	51

ВСТУП

У сучасному інформаційному суспільстві, насиченому великим обсягом текстової інформації, виникає необхідність в ефективних методах її аналізу та класифікації. Особливо це стосується розпізнавання та систематизації текстів, що визначаються як ключовий аспект у забезпеченні успіху багатьох галузей, від бізнесу до науки. У цьому контексті, дослідження з питань класифікації та аналізу тексту за допомогою нейронних мереж стає важливим завданням, що визначає актуальність даної роботи.

Зростаючий обсяг текстової інформації в мережі Інтернет, соціальних мережах, документах та інших джерелах створює потребу у вдосконалених інструментах для автоматизованого аналізу та розподілу цієї інформації. В умовах інформаційного перенасичення, важливо мати засоби, які допомагають розмежувати текстові дані за їхнім змістом, тематикою та іншими параметрами.

Застосування нейронних мереж для класифікації текстів визнається одним із найбільш перспективних напрямків у цьому контексті. Ці інтелектуальні системи машинного навчання можуть виявити високу ефективність у розпізнаванні та категоризації текстової інформації, здійснюючи це з урахуванням складних структурних та семантичних зв'язків.

Метою даного дослідження є розробка та вдосконалення методів класифікації та аналізу текстової інформації з використанням штучного інтелекту, а також створення ефективної системи, яка забезпечить точні та швидкі результати в аналізі відгуків для прийняття підприємницьких рішень.

Об'єктом дослідження є процеси аналізу та класифікації тексту, зокрема у відношенні до відгуків на товари, що є значущими в сфері сучасного електронного бізнесу та споживчого ринку.

Предметом дослідження є розробка та вдосконалення методів підготовки вихідної інформації, обробки даних, та реалізації модуля для автоматизованого аналізу, класифікації та управління відгуками на товари за допомогою штучного інтелекту. Для реалізації проекту була обрана мова програмування Python, оскільки вона найбільш зручна та найкраще пристосована для роботи із штучним інтелектом.

Основним завданням є визначення оптимальних стратегій та інструментів для ефективного аналізу тексту, забезпечуючи точну та інформативну класифікацію відгуків на товари. Для досягнення поставленої мети також необхідно виконати такі завдання:

- дати загальну характеристику та оглянути існуючі методи аналізу та класифікації тексту;
- визначити переваги та недоліки використання нейромереж для задач категоризації тексту;
- описати існуючі методи збору вихідних даних та обрати найбільш доцільний метод для подальшого дослідження;
- поставити технічне завдання та розробити веб-додаток для автоматизації класифікації та аналізу тексту;
- оцінити економічний ефект від впровадження розробленого додатку.

У процесі дослідження були використані методи аналізу, моделювання та реалізації алгоритмів машинного навчання, зокрема нейромереж, для досягнення поставлених цілей.

Дослідження базується на використанні наукових публікацій, існуючих методологій у сфері аналізу та класифікації тексту, а також враховує новітні досягнення в галузі штучного інтелекту.

Проведене дослідження передбачає створення нового підходу до класифікації та аналізу тексту, ґрунтованого на використанні штучного інтелекту.

Результати апробації основних положень кваліфікаційної магістерської роботи розглядалися на Міжнародній науково-практичній конференції

«Проблеми та перспективи розвитку фінансово-кредитної системи України» 4-5 грудня 2023 року. За результатами, було опубліковано тезу в збірнику конференції: «Кільдей А. Д., Кушнерьов О. С., Койбічук В. В. Роль та ефективність використання нейронної мережі у вдосконаленні аналізу, класифікації та управління відгуками на товари. Проблеми та перспективи розвитку фінансово-кредитної системи України: Матеріали Міжнародної науково-практичної конференції / за заг. ред.: Л. Л. Гриценко, І. В. Тютюнник. – Суми : Сумський державний університет, 2023. С. 36–39».

1 ОГЛЯД ТЕОРЕТИЧНИХ ОСНОВ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТУ

1.1 Загальна характеристика аналізу та класифікації тексту

Автоматизована класифікація тексту за заздалегідь визначеними категоріями завжди вважалася життєво необхідним методом управління та обробки великої кількості документів у цифрових формах, які широко поширені і кількість яких постійно зростає.

Аналіз тексту – це процес застосування комп'ютерних систем для зчитування та інтерпретації написаного людиною тексту з метою витягнення бізнес-інформації. Програмне забезпечення для аналізу тексту може автоматично класифікувати, сортувати та витягувати інформацію з тексту для виявлення закономірностей, взаємозв'язків, тенденцій та інших практично корисних відомостей. Аналіз тексту можна використовувати для ефективної та точної обробки численних текстових джерел (електронних листів, документів, вмісту соціальних мереж та відгуків про товари) так само, як це зробила б людина [19].

Класифікація тексту — це завдання віднести документ до попередньо визначеної категорії. більше формально, якщо d_i є документом усієї множини документів D , а $\{c_1, c_2, c_3, \dots, c_n\}$ є множиною усі категорії, тоді класифікація тексту призначає одну категорію c_j документу d_i . Документи в залежності від їх характеристик можуть бути позначені як для одного класу, так і для більше одного класу. Якщо документ віднесено лише до одного класу, він називається «одномітковим» і якщо документ віднесено до кількох класів, це називається «багатомітковим».

Проблему класифікації тексту з «одною міткою» можна далі класифікувати як проблему «бінарного класу», якщо документу призначено лише один із двох класів, і ця проблема класифікації тексту з «одною міткою» стає проблемою «багато класовості», якщо документу призначено лише N взаємовиключних класів призначено документу. Текстова класифікація

складається з подання документа, вибір або перетворення ознак, застосування алгоритму інтелектуального аналізу даних і, нарешті оцінка застосованого алгоритму інтелектуального аналізу даних [8].

Класифікація тексту — це проблема побудови моделей, які можуть класифікувати нові документи за заздалегідь визначеними класами. Зараз це складний процес, який включає не лише навчання моделей, а й численні додаткові процедури, наприклад попередня обробка даних, перетворення та зменшення розмірності. Класифікація тексту залишається важливою темою дослідження, використовуючи різні методи та їх комбінації в складних системах. Крім того, дослідники або розробляють нові системи класифікації, або вдосконалюють існуючі, включаючи їх елементи, щоб отримати кращі результати, тобто вищу обчислювальну ефективність.

Доступні огляди здебільшого описують і зосереджуються на наступних п'яти елементах процесу класифікації тексту:

- попередня обробка документа, тобто токенізація, видалення стоп-слова, а також формування основи або лематизація;
- моделювання документа, тобто представлення документа у відповідній формі, яка буде оброблена алгоритмом машинного навчання;
- вибір і проектування ознак;
- використання алгоритму машинного навчання для побудови моделі або функції класифікації;
- визначення показників якості та методів оцінки.

К. Аас і Л. Ейквіл [4] раніше перерахували й описали етапи класифікації тексту, а саме попередню обробку, створення векторної просторової моделі, зменшення розмірності (вибір ознак і проектування), навчання функції класифікації та вимірювання ефективності. У їхній роботі класифікація тексту використовувалася для представлення кількох схем зважування ознак, наприклад, логічне значення, частота терміну, зворотна частота документа та ентропія.

Крім того, автори пояснили три методи відбору ознак, а саме: порогове значення частоти документа, приріст інформації та χ^2 -статистику, а також один метод проєкції ознак, а саме латентне семантичне індексування (LSI).

Крім того, вони узагальнили та пояснили шість методів машинного навчання: алгоритм Роккіо, наївний Байєс, k-найближчий сусід, дерево рішень, опорний векторний механізм (SVM) і ансамблеве навчання, включаючи алгоритми пакетування та підвищення. Крім того, вони описали показники продуктивності для завдань класифікації з двома, кількома класами та кількома мітками. Вони також надали набір даних Reuters-21578, придатний для різних класифікаційних експериментів.

Базовий процес класифікації тексту включає такі елементи (рис. 1.1):



Рисунок 1.1 – Базовий процес класифікації тексту

З рисунку 1.1 маємо, що процес класифікації тексту за допомогою нейронних мереж можна розділити на кілька етапів:

1. Збір даних.

Перший етап полягає в зборі даних для навчання нейронної мережі. Дані можуть бути отримані з різних джерел, таких як внутрішні набори даних, Інтернет і відкриті бази даних. Дані повинні представляти різні типи текстів, які потрібно класифікувати.

2. Попередня обробка даних

Після того, як дані зібрані, вони повинні бути попередньо оброблені для створення представлення, необхідного для навчання нейронної мережі. Цей процес може включати такі кроки, як:

- Вилучення характеристик – вибір характеристик тексту, які будуть використовуватися для класифікації.
- Нормалізація даних – приведення даних до одного масштабу.
- Видалення шуму – видалення непотрібної інформації з даних.

3. Навчання нейронної мережі

На цьому етапі нейронна мережа навчається класифікувати тексти. Цей процес може бути виконаний за допомогою різних методів навчання, таких як зворотне поширення помилки або навчання без учителя.

4. Класифікація тексту

Після того, як нейронна мережа навчена, вона може використовуватися для класифікації тексту. Цей процес включає в себе подачу тексту нейронній мережі та отримання її прогнозу класу тексту.

5. Оцінка результатів.

Розглянемо приклад класифікації тексту за допомогою нейронних мереж. Нехай потрібно розробити тест для оцінки знань студентів з історії. Для цього необхідно зібрати дані для навчання нейронної мережі, використовуючи внутрішні набори даних, Інтернет і відкриті бази даних. Дані повинні включати різні типи історичних текстів, такі як статті, книги та документи.

Після того, як дані зібрані, можемо попередньо обробити їх, щоб створити представлення, необхідне для навчання нейронної мережі. Наприклад, можна вибрати такі характеристики тексту, як частота слів, довжина речень і складність граматики. Також необхідно нормалізувати дані, щоб привести їх до одного масштабу, і видалити шум, наприклад, пунктуацію і стоп-слова.

Наступним етапом виступає навчання нейронної мережі класифікації текстів за допомогою методу зворотного поширення помилки. Цей метод дозволяє нейронній мережі навчитися прогнозувати клас тексту, використовуючи дані, на яких вона була навчена.

Після того, як нейронна мережа навчена, можна використовувати її для класифікації тексту з тесту. Даний приклад наочно відображає суть класифікації тексту та його практичне застосування у сучасному світі.

1.2 Огляд існуючих методів аналізу та класифікації тексту

Одним із методів аналізу та класифікації тексту є дерево ухвалення рішень (також можуть називатися деревами класифікацій або регресійними деревами) — використовується в галузі статистики та аналізу даних для прогнозних моделей. Структура дерева містить такі елементи: «листя» і «гілки». На ребрах («гілках») дерева ухвалення рішення записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах — атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка прогнозує значення цільової змінної на основі декількох змінних на вході [1].

Під класифікацією текстів розуміється розподіл текстових документів за задалегідь визначеними категоріями. Методи класифікації машинного

навчання (machine learning) і текстів лежать на межі двох областей інформаційного пошуку (information retrieval). Відповідно, автоматична класифікація може здійснюватися на основі заздалегідь заданої схеми класифікації і вже наявної множини класифікованих документів або бути повністю автоматизованою. При використанні підходів машинного навчання, класифікаційне правило будується на основі тренувальної колекції текстів.

Класифікація з використанням дерева рішень виникає з розбиття даних на групи, ґрунтуючись на значеннях змінних простору ознак. Це призводить до створення ієрархії операторів "якщо-то", які класифікують дані. Для визначення категорії, до якої відноситься конкретний документ, потрібно відповісти на питання, що розташовані в вузлах дерева, починаючи з його кореня. Ці питання мають форму "чи значення змінної і більше порога і b?". Позитивна відповідь веде до переходу до правого вузла, а негативна - до лівого. Наступне питання пов'язане з відповідним вузлом.

Для автоматичної побудови дерев рішень використовується алгоритм, розроблений за допомогою навчання на прикладах. Розглянемо один з таких алгоритмів, який циклічно розбиває навчальні приклади на класи згідно з змінною, що має найбільшу класифікуючу силу. Кожна підмножина прикладів, визначена такою змінною, знову розбивається на класи з використанням змінної з найбільшою класифікуючою здатністю, і так далі. Процес розбиття завершується, коли в підмножині залишаються елементи лише з одного класу, і утворюється дерево рішень.

Для визначення змінної з найбільшою класифікуючою силою використовується критерій інформаційної ваги слова в рубриці. Після побудови "точного" дерева рішень до нього застосовуються процедури усічення і перетворення для досягнення балансу між складністю (кількістю вузлів) і якістю навчання. Класичний метод перетворення - алгоритм C4.5.

Одним з недоліків методу дерев рішень для класифікації текстів є те, що алгоритм надає однакову вагу "позитивним" і "негативним" розгалуженням в вузлах. Це може спричинити "перенавчання" алгоритму. Проте перевагою є те,

що побудоване дерево легко піддається аналізу, і результати алгоритму можна інтерпретувати наглядно. Для візуалізації дерев рішень існують програми графічного відображення.

Метод Байєсової (наївної) класифікації (Naive Bayes). Наївний Байєсівський класифікатор традиційно використовується в задачах класифікації текстів, таких як фільтрація спаму, автоматична рубрикація або визначення тональності документа. Набули поширеного розвитку два його різновиди: багатомірний (multivariate) та мультиноміальний (multinomial).

Зазначений метод використовує ймовірнісну модель, в якій класифікація та включення у відповідну категорію документів проводиться шляхом оцінювання ймовірності появи слів у документі. Ймовірності можуть бути використані для оцінки найбільш близьких категорій тестового документа.

Основні переваги наївного байєсівського класифікатора обчислювальні витрати при навчанні та класифікації. У тих рідкісних випадках, коли ознаки дійсно незалежні (або майже незалежні), наївний байєсівський класифікатор (майже) оптимальний. Основним недоліком методу є відносно невисока якість класифікації в більшості реальних завдань. Зазначений метод часто використовується в якості базового методу при порівнянні різних методів машинного навчання.

Метод опорних векторів (Support Vector Machine, SVM) пошуку площини вирішення, яка може розділити позитивні і негативні приклади в багатовимірному просторі функції, в якому навчальні документи представлені як вектори. У своєму первинному вигляді алгоритм вирішував завдання розрізнення об'єктів двох класів. Метод набув величезну популярність завдяки своїй високій ефективності. Багато дослідників використовують його в роботах, присвячених класифікації текстів. Підхід, запропонований В. Вапником для визначення того, до якого з двох заздалегідь визначених класів повинен належати аналізований зразок, заснований на принципі структурної мінімізації ризику. Результати класифікації текстів за допомогою методу опорних векторів є одними з найкращих, у порівнянні з іншими методами

машинного навчання. Однак, швидкість навчання даного алгоритму одна з найнижчих. Метод опорних векторів вимагає великого обсягу пам'яті і значних витрат машинного часу на навчання, що знижує його масштабованість. Проте даний алгоритм можна використовувати як еталон з точки зору якості класифікації. Метод буде працювати ефективно, якщо опорних векторів буде порівняно не багато, якщо ж їх кількість зростає, метод стає малоефективний через значно збільшену складність.

Метод k-найближчого сусіда (k-nearest neighbor). Цей метод є одним з найбільш вивчених і високоточних алгоритмів, що використовуються при створенні автоматичних класифікаторів. Вперше він був запропонований ще в 1952 році для вирішення завдань дискримінантного аналізу. У дослідженнях, присвячених аналізу роботи різних алгоритмів машинного навчання для задачі класифікації текстів, цей метод демонструє одні з найкращих результатів. В основі методу лежить досить проста ідея: знаходити та відрубрикувати колекції найбільш схожі на аналізований текст документи і, опираючись на знання про їх категоріальну приналежність, класифікувати невідомий документ. З метою визначення рубрики, релевантні документу d , цей документ порівнюється з усіма документами з навчальної вибірки.

Головною особливістю, що виділяє зазначений метод серед інших, є відсутність у нього стадії навчання. Іншими словами, належність документа категоріям визначається без побудови класифікуючої функції. Основною перевагою такого підходу є можливість оновлювати навчальну вибірку без перенавчання класифікатора. Ця властивість може бути корисною, наприклад, у випадках, коли навчальна колекція часто поповнюється новими документами, а перенавчання займає надто багато часу. Класичний алгоритм пропонує порівнювати аналізований документ з усіма документами з навчальної вибірки і тому головний недолік описаного методу полягає в тривалості часу роботи рубрикатора на етапі класифікації [2].

Штучні нейронні мережі (Artificial Neural Networks) вже понад тридцять років знаходять широке застосування в галузі штучного інтелекту та аналізу

даних. Вони представляють собою математичну модель, існуючу в програмній або апаратній формі, засновану на принципах взаємодії нейронів, аналогічно організації нервових клітин у живих організмах. Цей метод машинного навчання визнаний одним з найстаріших та найбільш відомих.

Штучні нейронні мережі є адаптивною системою, побудованою з групи з'єднаних штучних нейронів. Ця система може навчатися змінювати свої внутрішні стани, виявляти зв'язки між документами та їх категоріями. Для успішної класифікації текстів важливо визначити оптимальну структуру та топологію нейронної мережі.

Основні топології класифікуючих нейронних мереж включають одно- і багатошаровий перцептрон, нейромережевий Гаусів класифікатор, мережу Кохонена, мережу вбудованого розповсюдження та каскадну мережу [15]. Усі ці топології проявляють високу точність у роботі з лінійними та нелінійними прикладами. Однак прийняття рішень щодо класифікації ускладнюється через природу організації нейронної мережі, ставлячи перед вченими виклик з урахуванням масштабованості при обмежених обчислювальних ресурсах.

Штучні нейронні мережі виявляються досить потужним інструментом для розв'язання завдань аналізу даних, але їх використання також вимагає ретельного підходу до визначення параметрів та оптимізації структури для досягнення найкращих результатів у конкретному контексті.

1.3 Переваги та недоліки використання нейромереж для задач класифікації тексту

У сучасному світі штучний інтелект знаходить широке застосування у багатьох галузях, зокрема в аналізі та класифікації текстової інформації. Це надає численні переваги, але також супроводжується відповідними викликами, які варто враховувати при прийнятті рішення.

Переваги використання даного методу для класифікації тексту включають в себе:

1. Швидкість та підвищену ефективність. Алгоритми штучного інтелекту можуть миттєво обробляти великі обсяги текстової інформації, дозволяючи автоматизувати завдання класифікації в реальному часі.

2. Автоматизація. Штучний інтелект дозволяє автоматизувати процес класифікації тексту, звільняючи від необхідності ручного аналізу, особливо при роботі з великими обсягами даних. В рамках підприємницької діяльності це означає обов'язкове зменшення трудовитрат, а отже економію та вивільнення коштів.

3. Об'єктивність. ШІ-моделі не піддаються людським упередженням, що робить їх більш об'єктивними для задач класифікації тексту. Це важливо для задач, де потрібна неупереджена класифікація, наприклад, для судових справ або медичних досліджень.

4. Висока точність. Деякі моделі машинного навчання, зокрема нейронні мережі, досягають високого рівня точності в класифікації тексту, особливо після налагодження та оптимізації. Звичайно точність досягається шляхом підбору оптимальних параметрів та використання широкого датасету для навчання.

5. Спроможність враховувати контекст тексту. Використання контекстуальних моделей, таких як BERT чи GPT, дозволяє враховувати більше контексту та взаємозв'язків між словами, що покращує якість класифікації.

Недоліки використання штучного інтелекту наведені нижче:

1. Необхідність великої кількості тренувальних даних. Більшість моделей вимагають значної кількості даних для ефективного тренування, що може ускладнити завдання у випадках, коли дані обмежені. Натомість, велика вибірка суттєво підвищує точність класифікатора.

2. Складна інтерпретація. Деякі складні моделі можуть бути важкими для розуміння, що ускладнює їхню інтерпретацію. Даний недолік переважно вирішується візуалізацією даних.

3. Проблема загального моделювання. Деякі моделі можуть важко адаптуватися до нових чи специфічних для конкретного домену даних. Таким чином рекомендується донавчання моделі, що веде за собою збільшення витрат на використання.

4. Чутливість до якості даних. Моделі, створені штучним інтелектом зазвичай чутливі до якості та збалансованості даних, що впливає на якість класифікації.

5. Питання етики та прозорості. Використання штучного інтелекту вимагає уваги до етичних аспектів, таких як конфіденційність та вплив на приватне життя, а також вимагає створення прозорих та зрозумілих моделей.

Загалом, не дивлячись на виклики цієї технології, використання штучного інтелекту для класифікації тексту надає значні переваги у забезпеченні ефективного та швидкого аналізу великих обсягів інформації.

Однак для досягнення максимальної результативності, важливо прийняти до уваги переваги та недоліки цієї технології, враховуючи специфіку конкретного завдання та його контексту.

2 ПІДГОТОВКА ТА ОБРОБКА ДАТАСЕТУ ДЛЯ КЛАСИФІКАЦІЇ ВІДГУКУ НА ТОВАР

2.1 Роль штучного інтелекту у вдосконаленні аналізу, класифікації та управління відгуками на товари

Штучний інтелект (ШІ) активно впроваджується в різні сфери людського життя, змінюючи традиційні підходи до бізнес-процесів. Використання технологій штучного інтелекту і технологій в бізнес-операціях має потенціал для підвищення продуктивності праці на організаційному рівні [21]. Однією з сфер, де він виявляє великий вплив, є аналіз, класифікація та управління відгуками на товари.

Штучний інтелект – це здатність машин виконувати завдання, які зазвичай вимагають людського інтелекту, такі як розуміння мови, розпізнавання моделей та прийняття рішень. ШІ працює за допомогою алгоритмів, які є наборами правил або інструкцій, для обробки великих обсягів даних та навчання з них. AI може використовувати різні методи, такі як обробка природної мови (NLP), машинне навчання (ML) та глибоке навчання (DL), щоб аналізувати відгуки клієнтів та генерувати результати.

Нейронні мережі також можуть допомогти автоматизувати аналіз відгуків клієнтів для контролю якості, економії часу та ресурсів, одночасно підвищуючи точність та послідовність результатів. ШІ здатен розкривати приховані шаблони, кореляції та деталі з відгуків клієнтів, які люди можуть не помітити або пропустити. Крім того, ця технологія допомагає визначити потреби клієнтів, очікування, уподобання та больові точки, щоб запропонувати дії або рішення для покращення продуктів та послуг.

Щоб використовувати такі технології для аналізу відгуків клієнтів для контролю якості, потрібно збирати релевантні, чіткі та повні відгуки клієнтів з різних джерел, таких як опитування, огляди, рейтинги, соціальні мережі,

електронні листи, чати, дзвінки тощо. Потім необхідно очистити, організувати та відформатувати дані зворотного зв'язку, щоб полегшити обробку алгоритмів ШІ. Після цього можна застосувати відповідну методологію – NLP (Natural Language Processing), ML (Machine Learning) або DL (Deep Learning) - для створення балів, міток, категорій, резюме або рекомендацій. Нарешті, необхідно зрозуміти і оцінити результати алгоритмів ШІ і використовувати їх для поліпшення процесів контролю якості. Це може включати візуалізацію результатів або впровадження змін на основі відгуків клієнтів [22].

Необхідно розглянути, які плюси та мінуси приносить ця трансформація:

1. Аналіз та Класифікація.

Статистика показує, що кожного дня в інтернеті публікується мільйони відгуків на товари. За допомогою алгоритмів штучного інтелекту, компанії можуть ефективно обробляти цей великий обсяг даних та використовувати цю інформацію для покращення маркетингу, виробництва товарів тощо.

Згідно з дослідженням Gartner [23], використання нейронних мереж для аналізу відгуків може призвести до збільшення швидкості обробки даних на 60%, що дозволяє бізнесу більш оперативно реагувати на зміни в споживацьких попитах.

2. Управління та Відгуки.

Однією з ключових переваг використання ШІ є автоматизоване управління відгуками. За даними Reputation Institute [24], компанії, які ефективно взаємодіють з відгуками та реагують на них, можуть побачити збільшення лояльності споживачів на 12-15%. Також, дослідження показує, що кожен додатковий позитивний відгук може призвести до збільшення прибутку в середньому на 10%.

3. Етичні та Приватнісіні Ризики.

Згідно з опитуванням Pew Research Center [25], 64% споживачів висловлюють занепокоєння щодо того, як компанії використовують їхні особисті дані для аналізу відгуків. Це викликає питання етичності та

прозорості в застосуванні ШІ. Більшість споживачів (71%) також відзначають, що важливо мати контроль над тим, як їхні дані використовуються.

Усе це підкреслює важливість балансу між використанням ШІ для поліпшення бізнес-процесів підприємства.

Штучний інтелект, який використовується для аналізу тональності тексту (Sentiment analysis), стає все більш популярним інструментом для підприємств, що прагнуть отримати максимум інформації від клієнтського відгуку на товар або послугу. Ось декілька прикладів успішних вивчених випадків імплементації штучного інтелекту в бізнес-процеси компаній, які демонструють потужність технології у різних галузях (табл. 2.1).

Таблиця 2.1 – Приклади компаній: що використовують аналіз тональності тексту зі штучним інтелектом.

Назва компанії	Спосіб використання технології
Airbnb – сервіс для бронювання житла	Використовує аналіз тональності тексту на основі штучного інтелекту для аналізу відгуків клієнтів та визначення областей для вдосконалення. Аналізуючи тисячі відгуків, Airbnb зміг виявити конкретні проблеми для клієнтів, такі як чистота арендованого приміщення та комунікація з власником, і вносити зміни до своєї платформи та послуг для їх вирішення.
Coca-Cola – компанія харчової галузі	Компанія Coca-Cola використовувала аналіз тональності тексту для аналізу обговорень свого бренду в соціальних мережах під час Чемпіонату світу 2018 року. Аналізуючи настрої та ключові теми в режимі реального часу, Coca-Cola змогла створити цільові рекламні кампанії та взаємодіяти з фанатами більш ефективним способом.
T-Mobile – постачальних стільникового зв'язку	Компанія T-Mobile використовувала аналіз тональності тексту для аналізу відгуків клієнтів та виявлення загальних проблем. З використанням аналізу настроїв на основі штучного інтелекту T-Mobile швидко виявила ключові проблеми та вирішила їх, що призвело до зменшення кількості скарг клієнтів на 73%.

Продовження таблиці 2.1

Ford – виробник автомобілів	Ford використовує аналіз для аналізу відгуків клієнтів та виявлення тенденцій у продуктивності транспортних засобів та задоволенні клієнтів. Аналізуючи відгуки в реальному часі, Ford зміг виявляти проблеми та вносити поліпшення до своїх автомобілів та послуг.
Sephora – косметична мережа магазинів	Sephora використовує аналіз тональності тесту на основі штучного інтелекту для аналізу відгуків клієнтів в соціальних мережах та виявлення тенденцій у виборі та поведінці клієнтів. Аналізуючи настрої та ключові теми, Sephora змогла розробити цільові маркетингові кампанії та покращити взаємодію з клієнтами.

Джерело: [26]

Реальні вивчені випадки сучасних компаній (табл. 2.1) демонструють потужність штучного інтелекту в аналізі тональності тексту.

При класифікації тексту програмне забезпечення для аналізу тексту вчиться асоціювати певні ключові слова з конкретними темами, намірами користувачів або тональностями тексту. При цьому використовуються наступні методи.

Класифікація на основі правил присвоює тексту теги на основі заздалегідь визначених правил для семантичних компонентів або синтаксичних шаблонів.

Системи на основі машинного навчання працюють шляхом навчання програм аналізу тексту на прикладах і підвищення їх точності у присвоєнні тексту тегів. Для обробки структурованих даних, категоризації слів і формування семантичного розуміння між ними ці системи використовують такі лінгвістичні моделі, як наївний баєсівський класифікатор, метод опорних векторів і глибоке навчання.

Наприклад, позитивний відгук часто містить такі слова, як "добре", "швидко" і "відмінно". Тим часом негативні відгуки можуть містити такі слова, як "незадоволений", "повільний" і "поганий". Спеціалісти з аналізу даних

навчають програмне забезпечення аналізу тексту шукати такі терміни та класифікувати відгуки як позитивні чи негативні. Таким чином, служба підтримки клієнтів може легко відстежувати настрої клієнтів за відгуками.

Класифікація тексту має тисячі варіантів застосування та використовується для широкого спектру завдань. У деяких випадках інструменти класифікації даних працюють непомітно, покращуючи функції додатків, якими ми користуємося щодня (наприклад, фільтрація спаму в електронній пошті). В інших випадках класифікатори використовуються маркетологами, менеджерами з продукту, інженерами та продавцями для автоматизації бізнес-процесів та економії сотень годин ручної обробки даних.

Нижче наведені деякі з найпопулярніших прикладів використання класифікації тексту.

Виявлення термінових проблем. У Twitter, користувачі надсилають 500 мільйонів твітів [61] кожного дня. Опитування показують, що 83% клієнтів, які коментують або скаржаться в соціальних мережах, очікують відповіді того ж дня, а 18% - негайної реакції.

За допомогою класифікації тексту, бізнес може отримувати цінні деталі з великих обсягів даних. Такі технології як аспектний аналіз тональності тексту, допомагають зрозуміти, про що говорять користувачі та як вони говорять про той чи інший аспект продукту. Наприклад, можна виявити потенційну кризу PR, клієнта, який збирається припинити співпрацю, скарги на помилки, що зачіпають більшість клієнтів.

Автоматизація процесів підтримки клієнтів. Створення позитивного досвіду для клієнтів – один із основних факторів для сталого та зростаючого бізнесу. Згідно з Hubspot, у компаній з відмінним обслуговування клієнтів, 93% клієнтів є повторними покупцями [62]. Дослідження також показало, що 80% респондентів припинили співпрацю з компанією через негативний досвід обслуговування.

Класифікація тексту може допомогти командам підтримки клієнтів автоматизуючи завдання, які краще залишити комп'ютерам. Це дозволить заощадити цінний час, який можна витратити на більш важливі речі.

Наприклад, класифікація тексту часто використовується для автоматичного розподілу та сортування тикетів (завдань). Вона дозволяє автоматично направляти тикети підтримки до співробітника з відповідними знаннями про продукт та рівнем доступу. Якщо клієнт пише про повернення коштів, можна автоматично призначити тикет співробітнику з дозволом на їх виконання. Це забезпечить швидшу відповідь клієнту.

Команди підтримки також можуть використовувати аналіз тональності для автоматичного визначення терміновості тикету та пріоритезації тих, що містять негативні настрої. Це може допомогти знизити відтік клієнтів і навіть покращити погану ситуацію.

Голос клієнта (VoC). Компанії використовують такі опитування, як Net Promoter Score, щоб прислухатися до голосу своїх клієнтів на кожному етапі їх взаємодії з продуктом.

Зібрана інформація є як якісною, так і кількісною. І хоча бали NPS легко аналізувати, але відкриті запитання потребують більш глибокого аналізу з використанням технік класифікації тексту. Замість того, щоб покладатися на людей для аналізу даних VoC, можна швидко обробляти відкриті відгуки клієнтів за допомогою машинного навчання. Моделі класифікації можуть допомогти проаналізувати результати опитування, щоб виявити тенденції та отримати такі знання:

- Що подобається людям у нашому продукті чи послугі?
- Що нам потрібно покращити?
- Що нам потрібно змінити?

Поєднання кількісних результатів та якісних аналізів дозволяє командам приймати більш обґрунтовані рішення, не витрачаючи годин на ручний аналіз кожної відповіді.

2.2 Опис основних методів збору датасету

Основні методи збору датасету для аналізу відгуків клієнтів за допомогою штучного інтелекту можна розділити на два типи:

Прямі методи, при яких відгуки клієнтів збираються безпосередньо від користувачів або покупців товарів та сервісів:

1. Опитування – один з найпоширеніших методів збору відгуків клієнтів. Опитування можуть бути проведені в різних формах, наприклад, онлайн, по телефону, пошті тощо.

2. Фокус-групи – це метод збору відгуків клієнтів, при якому невелика група людей обговорює певну тему під керівництвом модератора. Вони можуть бути корисними для отримання глибоких і всебічних відгуків про продукти або послуги.

3. Аналіз соціальних мереж – це метод збору відгуків клієнтів, при якому аналізуються публікації в соціальних мережах, тобто відгуки, коментарі та хештеги.

Непрямі методи, при яких відгуки клієнтів збираються з інших джерел, наприклад, з даних продажів, даних обслуговування клієнтів або з веб-сайту. До таких методів належать:

1. Аналіз даних обслуговування клієнтів – це метод збору відгуків клієнтів, при якому аналізуються дані про обслуговування клієнтів, такі як кількість скарг, час їх вирішення і рівень задоволеності клієнтів. Цей метод може бути корисним для виявлення проблем з обслуговуванням клієнтів, які негативно впливають на задоволеність клієнтів.

2. Аналіз даних веб-сайту – це метод збору відгуків клієнтів, при якому аналізуються дані про веб-сайт, такі як кількість відвідувань, час перебування на веб-сайті і поведінка користувачів. Аналіз даних веб-сайту може бути корисним для виявлення проблем з веб-сайтом, які негативно впливають на досвід клієнтів.

3. Використання готових відкритих даних для навчання, доступних у вільному доступі в мережі. Наприклад, Kaggle чи інші платформи для аналізу даних можуть містити реальні або штучні відгуки клієнтів на товари чи послуги.

4. Генерування вихідних даних самостійно за допомогою спеціальних нейромереж. Таким чином можливо згенерувати одразу ідеальний масив для подальшого навчання моделі.

Вибір методу збору датасету залежить від цілей аналізу, доступності даних та вартості доступу до даних.

2.3 Збір та обробка даних для навчання моделі

На основі попереднього пункту даної роботи було обрано такий метод збирання вихідних даних як їх генерація в ChatGPT [27]. Звичайно, наразі доступна велика кількість готових датасетів у таких сервісах як Kaggle [28], але рішення ґрунтувалося на таких параметрах:

- чистота даних, оскільки допомогою генерації даних ми можемо згенерувати повністю чистий від шуму датасет, який буде містити найбільш емоційно забарвлені слова для ефективного та результативного навчання;
- можливість одразу класифікувати відгуки як позитивний та негативний, що полегшує подальший процес підготовки датасету;
- безумовно такий метод збору даних вважається легким, а дані можна генерувати нескінченно;
- генерування бази даних можливо на будь-якій мові;
- відсутність додаткових грошових витрат.

Після збору вихідних даних все одно потрібно провести належну обробку тексту, оскільки цей процес вважається важливим етапом у процесі класифікації тексту. Етап підготовки домагає зменшити вірогідність помилок та підвищити точність класифікації [36, 30]. Основною метою цього етапу є видалення шумових та незначущих слів [30].

Кожен відгук мав такий процес обробки:

1. Очистка даних:

- видалення дублікатів, цифр, розділових знаків та чисел;
- вилучення стоп-слів та непотрібних слів, таких як: займенники та прийменники.

2. Трансформація та токенизація.

Токенизація – це процес розбиття відгуків на частини, які називають токенами.

3. Вилучення стоп-слів.

Слова, такі як сполучники та займенники, які не мають відношення до концепції тексту, називають стоп-словами. Цей процес включає вилучення деяких загальних слів, які часто зустрічаються в усіх документах. Важливо вилучати ці слова високої частоти, оскільки вони можуть спричинити помилкову класифікацію документів [37].

4. Стемінг (Stemming).

Стемінг - це процес зведення словоформ до однієї форми (стебла або кореня) за вилученням префіксів, суфіксів та інфіксів [30]. Процес стемінгу залишає лише кореневі форми слів. Таким чином, можна визначити терміни, які мають однаковий корінь, але виглядають як різні слова через їх афікси. Наприклад, "комп'ютер" і "комп'ютеризація" мають однаковий корінь "комп'ют" [37].

5. Вага термінів (Term weightning).

Після того як слова перетворюються в терміни, необхідно визначити їхню форму вираження у документі. Цей процес називається вагою термінів. Таким чином, кожен документ може бути записаний у векторній формі залежно від термінів, які він містить. Для отримання вектора ваги використовується алгоритм ваги ознак term frequency–inverse document frequency (TF-IDF). N-грама вказує на послідовність n токенів на основі слів.

6. Обрізка слів.

Мета процесу обрізання полягає у відсіченні менш часто вживаних ознак у наборі документів.

Після проведення усіх вищезазначених етапів нарешті можна приступати до створення та навчання моделі. За результатами цього пункту, було створено датасет у розмірі 2 500 відгуків з такими колонками, як: ПБ, контактні дані, відгук.

3 РОЗРОБКА МОДУЛЯ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ, КЛАСИФІКАЦІЇ ТА УПРАВЛІННЯ ВІДГУКАМИ НА ТОВАРИ

3.1 Технічне завдання до вирішення поставленої проблем за допомогою машинного навчання

Після створення модуля для автоматизованого аналізу та класифікації відгуків передбачається, що ця система буде використовуватися у багатьох галузях для підвищення ефективності та зменшення трудовитрат і часу на обробку вхідних відгуків від клієнтів. Звичайно, така система найкраще буде працювати та клієнтів B2C (business to customer), але її також можна імплементувати для B2B (business to business) компаній.

Передбачається, що компанія має своє середовище для взаємодії з клієнтами, наприклад, CRM (customer relations system), куди можна інтегрувати наш майбутній додаток.

Пропонується поділити процес створення системи на декілька етапів для наочності процесу:

Етап 1. Збіг вихідних даних та навчання моделі.

Як вже зазначалося в попередньому розділі, було прийнято рішення використовувати Chat GPT для генерації тренувального набору даних відгуків на товари, оскільки сервіс є безкоштовним, генерує дані українською мовою та дає відносно вільні від шуму відгуки.

Для навчання моделі було обрано середовище Google Colab [32], оскільки воно безкоштовне та має необхідну потужність для створення та навчання моделі.

Щодо самої моделі, порогові значення для точності класифікації відгуків мають складати не менше ніж 90%.

Етап 2. Формування бекенду додатку.

Система має працювати на мові програмування Python, оскільки вважається, що ця мова найкраще працює для нейронних мереж.

Алгоритм має приймати вихідні дані для класифікації у декількох форматах – csv, json, xlsx. Дана умова ґрунтується на тому, що потенційні користувачі можуть брати відгуки на товари з різних джерел у декількох форматах.

Надсилання готової класифікації буде відбуватися у нереляційну базу даних MongoDB [33], що забезпечує гнучкість, швидкий доступ до даних та зручність у роботі з великими обсягами інформації.

Етап 3. Формування фронденту додатку.

З урахуванням сучасних вимог до побудови додатків та необхідними функціями для зручної взаємодії, необхідно створити систему з такими функціями:

- на головній сторінці має бути інструмент для завантаження зовнішніх файлів (відгуків) в систему;
- після завантаження система має відображати результат аналізу відгуків та виводити їх на головній сторінці з контактними даними, змістом відгуку і відповідною класифікацією – «позитивний» або «негативний»;
- також необхідна діаграма на головному екрані зі співвідношенням кількості позитивних відгуків до негативних;
- для поліпшення роботи з відгуками необхідно виводити загальну кількість завантажених документів та скільки було опрацьовано на сьогоднішній день;
- додатково, потрібно вивести усі негативні відгуки в окрему секцію, де менеджер по роботі з клієнтами може наочно бачити відгук, контактні дані клієнта та опрацьовувати його (дати зворотній зв'язок та перенести відгук у категорію опрацьованих).

У результаті виконання усіх вищезазначених завдань будемо мати готовий до роботи з реальними даними додаток, що суттєво спростить роботу

менеджерам по роботі з клієнтами, а керівництву дасть можливість бачити загальну картину настроїв клієнтів щодо продукції/сервісів.

3.2 Опис структури і функціоналу модуля автоматизованого аналізу, класифікації та управління відгуками на товари

Згідно до попереднього розділу, розпочнімо розробку з навчання моделі. Для цього використовувався сервіс Google Colab [32].

У додатку Б наведено реалізацію нейронної мережі для обробки текстових даних за допомогою бібліотек TensorFlow [34] та scikit-learn. Для реалізації були імпортовані такі бібліотеки (рис. 3.1):

```
import json
import pandas as pd
import pickle
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
```

Рисунок 3.1 – Імпорт бібліотек

З них виокремимо основні для побудови та навчання моделі:

- Matplotlib.pyplot – використовується для побудови графіків та візуалізації даних;
- Tensorflow – бібліотека для розвитку та навчання нейронних мереж;
- Sequential, Embedding, LSTM, Dense, Dropout – класи та шари для визначення архітектури нейронної мережі;
- Tokenizer – використовується для перетворення тексту на послідовності токенів.

Наступним етапом завантажуюмо датасет для навчання моделі, який складається з 2 500 відгуків, які попередньо класифіковані на позитивні (1) та

негативні (0). На рисунку 3.2 можемо побачити код для завантаження та частину вихідного масиву даних.

```
df = pd.read_excel('/content/reviews_dataset.xlsx')
df = df[df['reviews'].notna()]

[ ] df.head()
```

	reviews	is_positive
0	Я люблю цей продукт! Це дуже просто у використ...	1
1	Цей продукт є порятунком! Я не уявляю свого жи...	1
2	Цей продукт чудовий! Це найкраще, що я коли-не...	1
3	Я наполегливо рекомендую цей продукт усім, хто...	1
4	Цей продукт трохи підвів. Віє не такий зручни...	0

Рисунок 3.2 – Завантаження та швидкий перегляд вихідних даних

У рамках підготовки даних необхідно розділити датасет на тренувальні (80%) та валідаційні (20%) дані (рис. 3.3).

```
# Split the data into training and testing sets
train_texts, test_texts, train_labels, test_labels = train_test_split(
    df['reviews'].values,
    df['is_positive'].values,
    test_size=0.2,
    random_state=42
)
```

Рисунок 3.3 – Розділення вихідних даних

Наступним етапом відбувається токенізація та створення самої моделі (3.4). Цей код створює модель із вбудованим Embedding-шаром для роботи з текстовими даними, LSTM-шаром для управління довгостроковою залежністю та Dense-шаром для виведення бінарного класифікаційного результату. Компіляція включає в себе налаштування параметрів оптимізатора та визначення метрик для вимірювання ефективності моделі під час навчання.

```
[ ] # Tokenize and pad the text sequences
max_words = 32000 # Adjust based on your dataset size
max_len = 100 # Adjust based on the desired sequence length
tokenizer = Tokenizer(num_words=max_words, oov_token='<OOV>')
tokenizer.fit_on_texts(train_texts)

[ ] train_sequences = tokenizer.texts_to_sequences(train_texts)
train_padded = pad_sequences(train_sequences, maxlen=max_len, truncating='post')

test_sequences = tokenizer.texts_to_sequences(test_texts)
test_padded = pad_sequences(test_sequences, maxlen=max_len, truncating='post')

[ ] # Build the model
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=max_words, output_dim=32, input_length=max_len),
    tf.keras.layers.LSTM(64),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model with optimizer
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
```

Рисунок 3.4 – Токенізація та побудова моделі

Навчання моделі відбувається за допомогою наступного фрагменту коду, зображеного на рисунку 3.5. Використовується бібліотека TensorFlow для тренування попередньо визначеної нейронної мережі з використанням даних тренування та валідації. Також імплементуємо засіб ранньої зупинки (early stopping) для автоматичної зупинки навчання моделі, якщо покращення на валідаційному наборі даних не спостерігається певну кількість епох (в даному випадку 5 епох).

```
[ ] # Train the model
early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

history = model.fit(
    train_padded,
    train_labels,
    epochs=100,
    validation_data=(test_padded, test_labels),
    batch_size=64,
    callbacks=[early_stopping]
)

Epoch 22/100
31/31 [=====] - 1s 17ms/step - loss: 0.2077 - accuracy: 0.9460 - val_loss: 0.2890 - val_accuracy: 0.9133
Epoch 23/100
31/31 [=====] - 1s 27ms/step - loss: 0.1954 - accuracy: 0.9510 - val_loss: 0.2586 - val_accuracy: 0.9032
Epoch 24/100
31/31 [=====] - 1s 21ms/step - loss: 0.1834 - accuracy: 0.9500 - val_loss: 0.2463 - val_accuracy: 0.9153
Epoch 25/100
31/31 [=====] - 0s 12ms/step - loss: 0.1680 - accuracy: 0.9571 - val_loss: 0.2678 - val_accuracy: 0.9093
Epoch 26/100
31/31 [=====] - 1s 26ms/step - loss: 0.1618 - accuracy: 0.9611 - val_loss: 0.2308 - val_accuracy: 0.9194
Epoch 27/100
31/31 [=====] - 0s 12ms/step - loss: 0.1586 - accuracy: 0.9581 - val_loss: 0.2288 - val_accuracy: 0.9234
Epoch 28/100
31/31 [=====] - 1s 17ms/step - loss: 0.1466 - accuracy: 0.9616 - val_loss: 0.2205 - val_accuracy: 0.9254
Epoch 29/100
31/31 [=====] - 0s 11ms/step - loss: 0.1362 - accuracy: 0.9707 - val_loss: 0.2139 - val_accuracy: 0.9294
```

Рисунок 3.5 – Навчання моделі

Після завершення навчання моделі виведемо графік точності (Model accuracy) та втрат (Model loss). Графік показує динаміку точності та втрат моделі машинного навчання протягом процесу навчання. Результат наведено на рисунку 3.6.

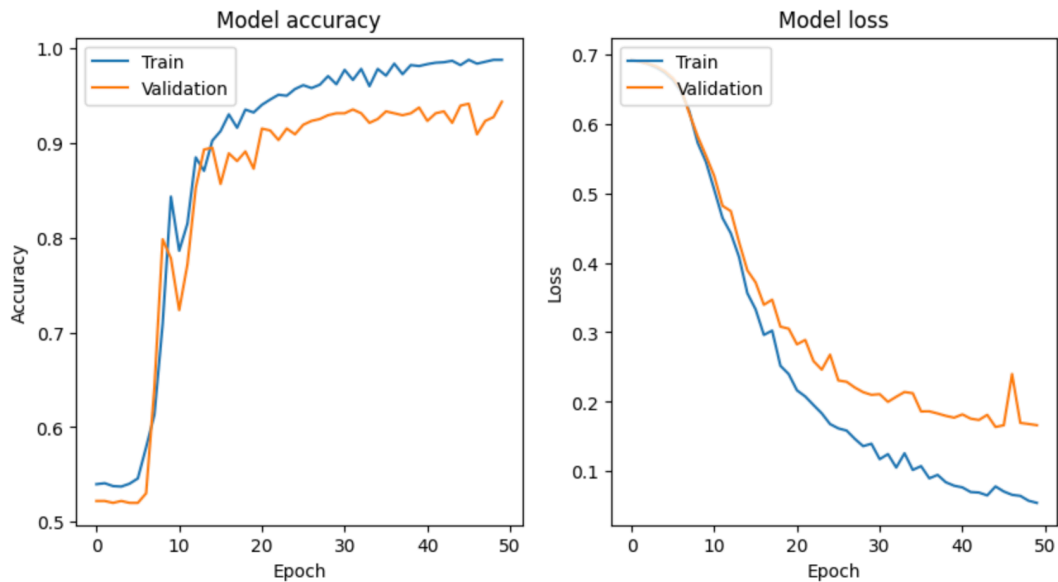


Рисунок 3.6 – Графік точності та втрат моделі

З рисунку 3.6 бачимо, що точність моделі на тренувальному наборі (синя лінія) поступово зростає, досягаючи приблизно 96% після 20 епох. Точність на наборі для перевірки (помаранчева лінія) також збільшується, досягаючи приблизно 94% після 20 епох. Втрати моделі на обох наборах зменшуються протягом процесу навчання.

На графіку також видно, що втрати моделі на наборі для перевірки починають зростати після певної кількості епох, що означає можливість перенавчання моделі при збільшенні кількості епох. Перенавчання – це проблема, яка може виникнути в машинному навчанні, коли модель навчена на надмірно великому тренувальному наборі. У цьому випадку модель стає занадто точною для тренувального набору, але має низьку точність на новому наборі даних.

Таким чином, графік свідчить про те, що LSTM-модель вдається навчитися виконувати завдання бінарної класифікації тексту. Це означає, що

модель може бути використана для прогнозування класу даних з високою точністю.

```
# Evaluate the model
loss, accuracy = model.evaluate(test_padded, test_labels)
print(f'Validation loss: {loss * 100:.2f}, Validation accuracy: {accuracy * 100:.2f}')

16/16 [=====] - 0s 6ms/step - loss: 0.1638 - accuracy: 0.9395
Validation loss: 16.38, Validation accuracy: 93.95
```

Рисунок 3.7 – Оцінка моделі

У цьому коді функція `evaluate()` моделі використовується для розрахунку втрат та точності моделі на наборі тестів. Втрати - це міра того, наскільки добре модель відповідає дійсності. Точність - це міра того, наскільки часто модель дає правильні відповіді.

Результати, виведені кодом, такі:

- Втрати: 16,38. Це означає, що модель у середньому помилялася на 0,1638 одиниці на одиницю даних у наборі тестів.
- Точність: 93,95%. Це означає, що модель правильно відповідала на 93,95% запитів у наборі тестів.

Оскільки точність моделі становить 93,95%, можна сказати, що модель є досить точною. Однак, оскільки втрати становлять 16,38% можна дійти до висновку, що модель все ще може бути покращена за допомогою підбору параметрів, даванням шарів для навчання, методами `transfer learning` та покращенням вихідного масиву даних.

Для подальшої розробки додатку виведена точність моделі співпадає з тією, що була визначена технічним завданням, тобто ми приймаємо цю модель для подальшої розробки.

Наступним кроком починаємо розробку серверної частини застосунку для автоматизації класифікації та аналізу відгуків. Повний код наведено у додатку В. У розробці було використано фреймворк веб-додатків `Flask` для мови програмування `Python`, оскільки він часто використовується для

створення онлайн-додатків та пропонує простий та гнучкий підхід до розробки веб-застосунків та API.

Наведений нижче код є частиною Flask-додатка, який реалізує API для імпорту та аналізу відгуків, зберігання їх у базі даних MongoDB, а також надає можливість отримання та оновлення відгуків через HTTP-запити. Основні функції включають:

1. Підготовка Flask-додатка та з'єднання з базою даних MongoDB.
2. Завантаження моделі та токенизатора (рис. 3.8):

```
loaded_model = tf.keras.models.load_model('tf_model/sentiment_model.h5')
with open('tf_model/best/tokenizer.pkl', 'rb') as handle:
    tokenizer = pickle.load(handle)
```

Рисунок 3.8 – Завантаження натренованої моделі нейронної мережі та токенизатора з файлів

3. Функції для обробки та аналізу відгуків (рис. 3.9):

```

usage
def analyze_sentiment_text(text):
    new_text = [text]
    new_text_sequence = tokenizer.texts_to_sequences(new_text)
    new_text_sequence = pad_sequences(new_text_sequence, maxlen=100)

    prediction = loaded_model.predict(new_text_sequence)

    is_positive = prediction[0][0] >= 0.5 # Assuming 0.5 is the threshold
    return 1 if is_positive else 0

@app.route(rule: '/api/reviews/import', methods=['POST'])
def import_reviews():
    try:
        file = request.files['file']
        print("file", file)
        if not file:
            return jsonify({'error': 'No file provided.'}), 400

        if file.filename.endswith('.csv') or file.filename.endswith('.xlsx'):
            if file.filename.endswith('.csv'):
                reviews_df = pd.read_csv(file)
            elif file.filename.endswith('.xlsx'):
                reviews_df = pd.read_excel(file)

            # Drop the column with label '№' if present
            reviews_df = reviews_df.drop(columns=['№'], errors='ignore')

            # Adjust field mapping based on actual column names in DataFrame
            field_mapping = {'ПІБ': 'full_name', 'Контакти': 'contact', 'Відгук': 'review'}
            reviews_df = reviews_df.rename(columns=field_mapping)
        elif file.filename.endswith('.json'):
            reviews_data = json.load(file)

```

Рисунок 3.9 – Код для реалізації функції обробки та аналізу відгуків

– `analyze_sentiment_text(text)` – функція для аналізу емоційного забарвлення відгуку. На даному етапі ми припускаємо, що якщо ймовірність позитивності відгуку більше, ніж 50%, то система буде класифікувати його як позитивний та навпаки.

API-маршрути:

- `/api/reviews/import` (POST) – завантаження та імпорт відгуків з файлу (CSV, Excel або JSON) до бази даних;
- `/api/reviews` (GET) – отримання всіх відгуків з можливістю фільтрації за позитивною або негативною ознакою;
- `/api/reviews/update_positive/<review_id>` (PUT) – оновлення статусу відгуку на позитивний.

Вищенаведений код реалізує основний функціонал веб-додатка, який може взаємодіяти з базою даних, нейронною мережею та надавати API для обробки відгуків.

Під час конфігурації бекенду додатку ми вже встановили зв'язок з нашою базою даних MongoDB [33]. Внаслідок цього відгуки, які були оброблені додатком, фіксуються у базі даних у вигляді індивідуальних об'єктів з такою структурою (рис. 3.10):

- `id` – індектифікатор відгуку, що заповнюється автоінкрементом;
- `contact` – номер телефону або електронна поштова адреса;
- `full name` – повне ім'я;
- `is_positive` – класифікація відгуку в форматі «1» - позитивний, «0» - негативний;
- `review` – текст відгуку;
- `timestamp` – час завантаження відгуку в базу даних;
- `timeUpdated` – параметр, що виводить час, коли відгук редагувався.

```
_id: ObjectId('6571c9193752e8cdcf2c95c4')
contact: "+380501234567"
full_name: "Коваль Ірина Ігорівна"
is_positive: "1"
review: "Задоволена покупкою, рекомендую!"
timeUpdated: ""
timestamp: "'01 Dec 2023 14:06:23 GMT'"
```

Рисунок 3.10 – Запис відгуку клієнта з проведеною класифікацією у базі даних MongoDB

У додатку Г наведено частину коду для побудови інтерфейсу додатку з використанням таких технологій, як:

- Node.js представляє собою платформу, яка надає можливість виконання JavaScript-коду на сервері. Такі додатки вважаються більш ефективними та масштабованими порівняно з традиційними технологіями.

– React, з іншого боку, є фреймворком для створення інтерфейсів користувача, побудованим на мові програмування JavaScript. Він використовує концепцію "компонентів", що представляють собою повторювані одиниці інтерфейсу користувача.

На представленому зображенні можна оглянути інтерфейс веб-додатку. Додаток складається із двох основних сторінок:

1. Головна сторінка (рис. 3.11):

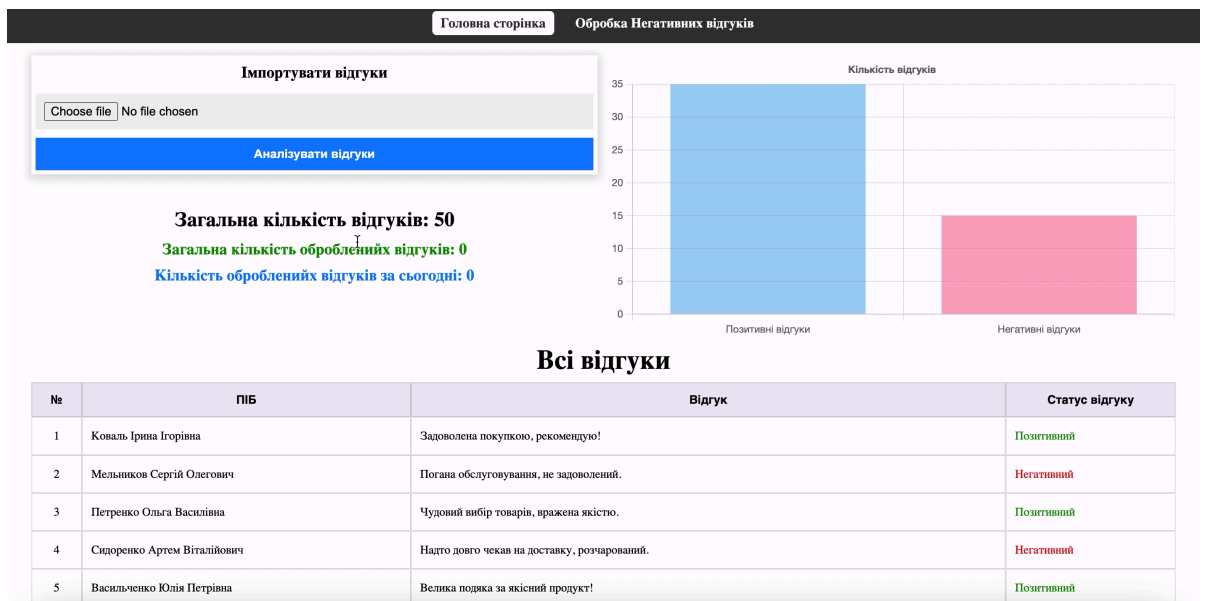


Рисунок 3.11 – Головна сторінка веб-додатку

Головна сторінка включає у себе список відгуків, які були завантажені в систему та класифіковані. Відгуки впорядковані за датою створення, починаючи з найновіших. На головній сторінці розташовані такі елементи:

- кнопка "Імпортувати відгуки" для завантаження нових відгуків з файлу;
- кнопка "Аналізувати відгуки" для запуску процесу аналізу відгуків після їх завантаження.

Частина сторінку для аналізу включає результати обробки відгуків, які містять такі відомості:

- кількість відгуків, які пройшли обробку.

- кількість відгуків, класифікованих як позитивні або негативні.

Сторінка аналізу містить також такі компоненти:

- таблиця зі списком всіх оброблених відгуків;
- графік, який ілюструє розподіл відгуків за категоріями (рис. 3.12).

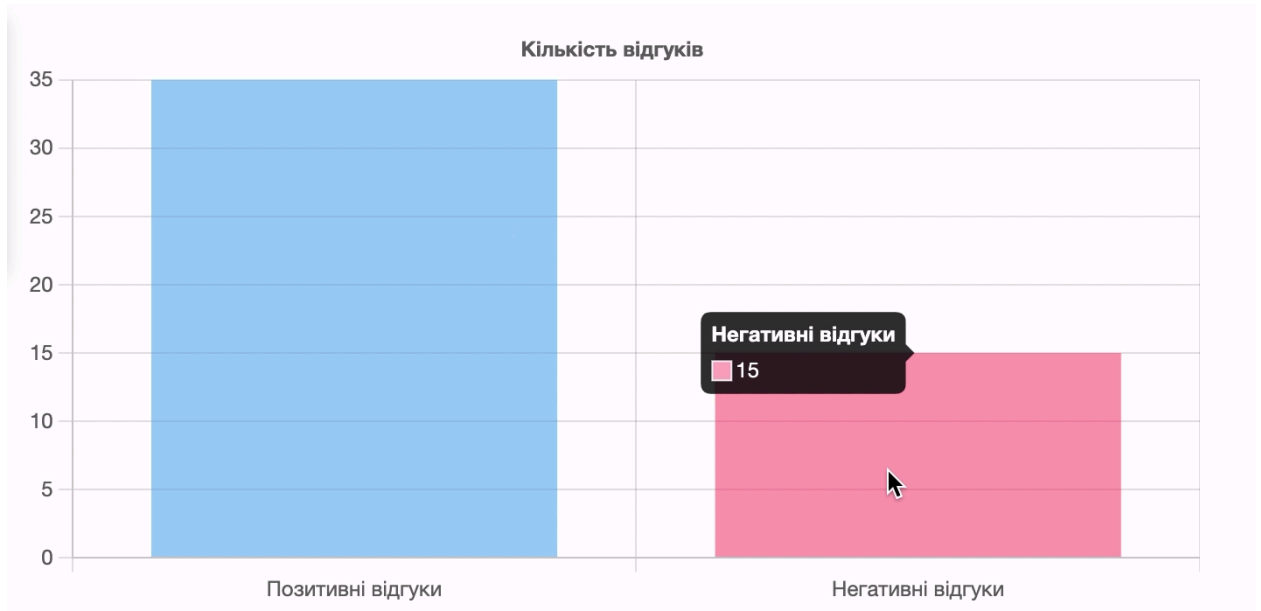


Рисунок 3.12 – Графік розподілу завантажених відгуків на позитивні та негативні

За допомогою графіку користувач має можливість зробити швидку оцінку відгуків одразу після завантаження та класифікації системою. Вісь y відображає шкалу кількості відгуків, що регулюється автоматично в залежності від кількості завантажених даних. По вісі x представлено дві категорії відгуків – негативні та позитивні. При наведенні курсором на відповідну частину графіку відображається мітка даних – кількість відгуків, що були класифіковані як позитивні чи негативні.

2. Друга сторінка має назву "Обробка негативних відгуків" (рис. 3.13).

Список містить перелік тих відгуків, які були класифіковані як негативні, впорядкованих за датою створення. Таблиця має такі колонки, як: порядковий номер, ПІБ клієнта, зміст відгуку, статус та кнопка перенесення коментаря в інший статус.

Головна сторінка		Обробка Негативних відгуків		
Відгуки				
№	ПІБ	Відгук	Статус	Дія
1	Кравченко Ігор Ігорович	Продукт не відповідає очікуванням, низька якість.	Негативний	<input type="button" value="Перемістити в оброблені"/>
2	Ткаченко Андрій Вікторович	Товар виявився несправним, розчарований.	Негативний	<input type="button" value="Перемістити в оброблені"/>
3	Литвиненко Катерина Сергіївна	Хороший вибір, але ціни трошки високі.	Негативний	<input type="button" value="Перемістити в оброблені"/>
4	Онопріско Вікторія Олегівна	Продукт виявився не таким, як на фото.	Негативний	<input type="button" value="Перемістити в оброблені"/>
5	Соловйова Аліна Павлівна	Проблеми з оплатою, не дуже зручно.	Негативний	<input type="button" value="Перемістити в оброблені"/>
6	Власенко Олександр Ігорович	Товар виявився кращим, ніж я сподівався.	Негативний	<input type="button" value="Перемістити в оброблені"/>
7	Трофімова Вікторія Андріївна	Проблеми з доставкою, трошки затрималася в дорозі.	Негативний	<input type="button" value="Перемістити в оброблені"/>
8	Ігнатенко Олександр Петрович	Товар поганої якості, не рекомендую.	Негативний	<input type="button" value="Перемістити в оброблені"/>
9	Шульга Анастасія Ігорівна	Замовлення відправлене з неправильним наповненням, розчарована.	Негативний	<input type="button" value="Перемістити в оброблені"/>
10	Демиденко Ілля Олегович	Хороший сервіс, дякую за професіоналізм.	Негативний	<input type="button" value="Перемістити в оброблені"/>
11	Кравець Ігор Олександрович	Недоліки в роботі сайту, не зручно замовляти.	Негативний	<input type="button" value="Перемістити в оброблені"/>

Рисунок 3.13 – Сторінка для обробки негативних відгуків

Таким чином користувачі додатку мають можливість переглядати усі негативні відгуки в окремому місці та обробляти їх. Після натискання на кнопку «Перемістити в оброблені» система надає відгукам новий статус – «Оброблений», а в базу даних додається запис про час оновлення відгуку. Цей функціонал дозволяє відслідковувати процес обробки відгуків, оскільки на головному екрані показується статистика того, скільки відгуків було оброблено загалом та скільки за сьогодні.

Звичайно, що при додаванні нових відгуків або обробці статистика з кількості та графік оновлюються автоматично за декілька секунд, тобто є можливість зробити процес повністю автоматизованим за умови інтеграції додатку в реальному часі до декількох джерел, наприклад, відгуки з сайту, Instagram та Google.

У підсумку, інтерфейс веб-додатку відзначається чіткістю, зручністю в користуванні та повністю відповідає вимогам технічного завдання, забезпечуючи користувачам необхідні інструменти для ефективної роботи з негативними відгуками.

3.3 Оцінка очікуваного ефекту від впровадження модуля автоматизованої класифікації відгуку на товар

Для проведення детальної оцінки ефективності введення автоматизованої системи для класифікації та аналізу відгуків необхідні детальні дані підприємства та порівняльна характеристика. Наприклад, статистичні дані ймовірності покупки після належного опрацювання негативного відгуку до введення автоматизованої системи та після. При цьому допускається, що при належній реакції керівництва та менеджерів підприємства ще можна повторно залучити клієнта до покупки продукту або сервісу.

З цього випливає, що ефективність впровадження автоматизованої системи для класифікації та аналізу відгуків можна визначити як співвідношення між отриманими результатами та витратами на впровадження. Отримані результати можна оцінити за такими критеріями:

- кількість негативних відгуків, які були успішно класифіковані;
- точність класифікації;
- ймовірність повторної покупки клієнтів, які залишили негативний відгук.

Витрати на впровадження можна оцінити за такими статтями:

- заробітна плата програмістів;
- інші витрати, пов'язані з розробкою та впровадженням системи.

Для цієї мети було побудовано наступну таблицю, що вміщує в собі оплату праці для програмістів та супутні витрати (табл. 3.1).

Апаратне забезпечення. При розробці додатку використовувався лише локальний сервер, що не несе за собою додаткових витрат. Однак для подальшої роботи та доступу до веб-додатку необхідний VPS-сервер, що буде коштувати 441 грн/місяць.

Програмне забезпечення. Для конфігурації додатку використовувалися безкоштовні сервіси.

Таблиця 3.1 – Оцінка економічних витрат

Статті витрат	Вартість в гривнях
Апаратне забезпечення	
Локальний сервер	Наявний
VPS-сервер для подальшого використання (NVMe VPS 2G) [35]	441 грн/місяць
Домен [35]	Залежить від конкретного обраного домену клієнтом
Програмне забезпечення	
Безкоштовно	
Витрати на оплату праці	
Заробітна плата розробника (Middle Javascript Software Engineer) [36]	91 773 грн/місяць
Заробітна плата розробника (Middle Python Software Engineer) [36]	99 114 грн/місяць
Тривалість розробки	1 місяць
Інші витрати	
Відсутні	

Витрати на оплату праці. Розробка такого додатку потребує наймання двох розробників. Таким чином, заробітна плата для Middle Javascript Software Engineer складає \$2500/місяць за даними dou.ua, що становить 91773 грн/місяць за курсом 36,71. Для Middle Python Software Engineer середня заробітна плата на ринку складає \$2700/місяць, що дорівнює 99114 грн/місяць за курсом 36,71.

Таким чином, у процесі розробки використовувалась локальна інфраструктура та безкоштовні програмні додатки, що суттєво знижує

загальну вартість проекту. Загальні витрати на заробітну плату розробникам та використання VPS сервера оцінюються у розмірі 190 887 грн + 441 грн/місяць. Дані результати свідчать про економічну обґрунтованість проекту, а отриманий веб-додаток як ефективний інструмент для класифікації та аналізу відгуків клієнтів.

ВИСНОВКИ

У ході дослідження, присвяченого аналізу та класифікації тексту з використанням штучного інтелекту, було виявлено значущі та перспективні напрями в цій області.

Загальна характеристика аналізу тексту в розділі 1 розкрила важливі аспекти його застосування в сучасному інформаційному суспільстві. Огляд існуючих методів аналізу та класифікації тексту дозволив визначити різноманітні підходи та їх ефективність у різноманітних сценаріях застосування.

Особливу увагу приділено перевагам та недолікам використання нейромереж для завдань класифікації тексту. За результатами аналізу можна зробити висновок, що нейромережі стали потужним інструментом для розв'язання складних завдань в цій області, здатними автоматично визначати закономірності та тенденції у текстових даних. Проте важливо враховувати їхні обмеження, зокрема, необхідність великого обсягу даних для ефективного навчання та необхідність уважного налаштування параметрів.

Розділ 2, присвячений підготовці та обробці датасету для класифікації відгуків на товар, детально розглядає роль штучного інтелекту у вдосконаленні аналізу відгуків та методологію збору та обробки даних для побудови моделі. Зазначається, що ретельний збір та належна обробка вихідного масиву даних значно підвищує точність моделі. Також, було розглянуто основні методи збору вихідних даних, такі як: опитування, готові датасети, генерація даних та інші.

У розділі 3 розглядається розробка модуля для автоматизованого аналізу, класифікації та управління відгуками на товари. У ньому описано технічне завдання до вирішення проблеми за допомогою машинного навчання, структура та функціонал модуля, а також проведена оцінка очікуваного ефекту на покращення процесу аналізу відгуків.

За результатами дослідження, було створено та навчено нейронну мережу для класифікації відгуків з точністю результатів 93%. Покращити модель, тобто показник точності, можна за допомогою оптимізації параметрів, додаванням шарів, методами transfer learning та використанням більш якісного вихідного масиву даних для навчання.

Також, для подальшої практичної реалізації було розроблено веб-додаток, що може класифікувати відгуки у режимі реального часу на основі раніше розробленої моделі.

Застосунок розроблений з використанням сучасних технологій та мов програмування, що дає можливість покращувати його, нарощуючи додатковий функціонал. Загальні витрати на розробку становлять 190 887 грн (заробітна плата) + 441 грн/місяць (VPS сервер для подальшого використання).

Отже, дане дослідження висвітлило не лише актуальність проблеми аналізу та класифікації тексту, але й потенційні можливості використання штучного інтелекту, зокрема нейромереж, для вирішення цього завдання.

Отримані висновки та результати можуть служити основою для подальших наукових досліджень та практичних застосувань в області обробки та аналізу текстової інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дерево ухвалення рішень: веб-сайт. URL: https://uk.wikipedia.org/wiki/Дерево_ухвалення_рішень (дата звернення 21.11.2023).
2. Методи класифікації текстових документів в задачах text mining. Волосюк Ю. В., канд. техн. наук. (Миколаївський філіал ПВНЗ «Європейський Університет», м. Миколаїв)
3. Типологія текстів: веб-сайт. URL: https://uk.wikipedia.org/wiki/Типологія_текстів (дата звернення 21.11.2023).
4. Aas, K., & Eikvil, L. (1999). Classification of text: An overview of the stages and methods. *Expert Systems with Applications*, 17(3), 237-247.
5. AI Text Classifier vs Human: Who Gets It Right More Often? URL: <https://tinyurl.com/5n8md3uy> (дата звернення 21.11.2023).
6. Cambria E., D. Das, S. Bandyopadhyay, and A. Feraco. 2017. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis. Socio-Affective Computing vol 5.*, ed. Cambria E., Das D., Bandyopadhyay S., Feraco A., vol 5. Cham: Springer. https://doi.org/10.1007/978-3-319-55394-8_1.
7. How do you use AI to automate customer feedback analysis for quality control? URL: <https://tinyurl.com/4xn49u73>.
8. Jindal, Rajni, Malhotra, Ruchika, & Jain, Abha (2015). "Techniques for text classification: Literature review and current trends." *Webology*, 12(2), Article 139.
9. Latent semantic analysis for text categorization using neural network: веб-сайт. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705108000993?via%3Dihub> (дата звернення 21.11.2023).
10. Marcin Michał Mirończuk, Jarosław Protasiewicz. A recent overview of the state-of-the-art elements of text classification. National Information Processing Institute, al. Niepodległości 188 b, 00-608 Warsaw, Poland

11. Multi-level graph neural network for text sentiment analysis: веб-сайт. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0045790621001051>
12. Paweł Cichosz. Naive Bayes classifier 2015 URL: https://www.researchgate.net/publication/313991281_Naive_Bayes_classifier .
13. Pollyanna Gonçalves, Matheus Araújo. Comparing and Combining Sentiment Analysis Methods 2014 URL: <https://arxiv.org/pdf/1406.0032.pdf>.
14. Sánchez-Franco, M., A. Navarro-García, and F. Rondán-Cataluña. 2019. A naive Bayes strategy for classifying customer satisfaction: A study based on online reviews of hospitality services.
15. Sebastiani F. Machine learning in automated text categorization / F. Sebastiani // ACM Comput. Surv. – March 2010. – Vol. 34, No. 1. – P. 1-47.
16. Trilochan Tripathy. Bayes and Naive Bayes Classifier 2014 URL: https://www.researchgate.net/publication/261368939_Bayes_and_Naive_Bayes_Classifier (дата звернення 21.11.2023).
17. Understanding Evolution and Future of Natural Language Processing. URL: <https://www.xenonstack.com/blog/evolution-of-nlp>.
18. V Umarani. Sentiment Analysis using various Machine Learning and Deep Learning Techniques 2021 URL: https://www.researchgate.net/publication/356624355_Sentiment_Analysis_using_various_Machine_Learning_and_Deep_Learning_Techniques.
19. What is Text Analysis? URL: <https://aws.amazon.com/ru/what-is/text-analysis/> (дата звернення 21.11.2023).
20. Zhao Jianqiang, Gui Xiaolin, Zhang Xuejun. Deep Convolution Neural Networks for Twitter Sentiment Analysis 2018 URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8244338>.
21. Czarnitzki, D.; Fernández, G.P.; Rammer, C. Artificial Intelligence and Firm-Level Productivity. J. Econ. Behav. Organ.

22. How do you use AI to automate customer feedback analysis for quality control? URL: <https://www.linkedin.com/advice/0/how-do-you-use-ai-automate-customer-feedback-analysis> (дата звернення 21.11.2023).
23. What Is Artificial Intelligence? URL: <https://www.gartner.com/en/topics/artificial-intelligence>
24. RepTrack. URL: <https://www.reprtrak.com/blog/>.
25. PewResearch. URL: <https://www.pewresearch.org/internet/2023/10/18/how-americans-view-data-privacy/>.
26. The role of AI in content sentiment analysis for customer feedback. URL: <https://aicontentfy.com/en/blog/role-of-ai-in-content-sentiment-analysis-for-customer-feedback>.
27. OpenAI. URL: <https://openai.com/> (дата звернення 21.11.2023).
28. Kaggle. URL: <https://www.kaggle.com/> (дата звернення 21.11.2023).
29. Behrooz Noori (2021) Classification of Customer Reviews Using Machine Learning Algorithms, Applied Artificial Intelligence, 35:8, 567-588, DOI: 10.1080/08839514.2021.1922843Jjj.
30. Bahassine, S., A. Madani, M. Al-Sarem, and M. Kissi. 2018. Feature selection using an improved Chi-square for Arabic text classification. Journal of King Saud University-Computer and Information Sciences 32(2):225-231. <https://doi.org/10.1016/j.jksuci.2018.05.010>
31. Google Colab. URL: <https://colab.google/> (дата звернення 21.11.2023).
32. MongoDB. URL: <https://www.mongodb.com/> (дата звернення 21.11.2023).
33. Tensorflow. URL: <https://www.tensorflow.org/> (дата звернення 21.11.2023).
34. Hosting Ukraine. URL: <https://www.ukraine.com.ua/uk/> (дата звернення 21.11.2023).

35. DOU.UA. URL: <https://jobs.dou.ua/salaries/?period=2023-06&position=Middle%20SE>.
36. Uysal, A., and S. Gunal. 2014. The impact of preprocessing on text classification. *Information Processing & Management* 50:104–12. doi:10.1016/j.ipm.2013.08.006.
37. Uğuz, H. 2011. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems* 24:1024–32. doi:10.1016/j.knosys.2011.04.014.
38. Ali, F., K. Kwak, and Y. Kim. 2016. Opinion mining based on fuzzy domain ontology and support vector machine: A proposal to automate online review classification. *Applied Soft Computing* 47:235–50. doi:10.1016/j.asoc.2016.06.003.
39. Bi, J., Y. Liu, Z. Fan, and E. Cambria. 2019. Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model. *International Journal of Production Research* 57(22):1–21. <https://doi.org/10.1080/00207543.2019.1574989>
40. Cambria E., D. Das, S. Bandyopadhyay, and A. Feraco. 2017. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis. Socio-Affective Computing vol 5.*, ed. Cambria E., Das D., Bandyopadhyay S., Feraco A., vol 5. Cham: Springer. https://doi.org/10.1007/978-3-319-55394-8_1
41. Cao, Q., W. Duan, and Q. Gan. 2011. Exploring determinants of voting for the “helpfulness” of online user reviews: A text mining approach. *Decision Support Systems* 50:511–21. doi:10.1016/j.dss.2010.11.009.
42. Cardot, H., and D. Degras. 2018. Online principal component analysis in high dimension: Which algorithm to choose? *International Statistical Review* 86:29–50. doi:10.1111/insr.12220.
43. Dadgar, S., M. Araghi, and M. Farahani 2016. A novel text mining approach based on TF-IDF and support vector machine for news classification. In *A novel text mining approach based on TF-IDF and support vector machine for news*

classification. 2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE, Coimbatore, India. 112–16.

44. Duan, W., Y. Yu, Q. Cao, and S. Levy. 2016. Exploring the impact of social media on hotel service performance: A sentimental analysis approach. *Cornell Hospitality Quarterly* 57:282–96. doi:10.1177/1938965515620483.

45. Gulsoy, N., and S. Kulluk. 2019. A data mining application in credit scoring processes of small and medium enterprises commercial corporate customers. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9:e1299.

46. Hu, Y., and K. Chen. 2016. Predicting hotel review helpfulness: The impact of review visibility, and interaction between hotel stars and review ratings. *International Journal of Information Management* 36:929–44. doi:10.1016/j.ijinfomgt.2016.06.003.

47. Moro, S., P. Rita, and J. Coelho. 2017. Stripping customers' feedback on hotels through data mining: The case of Las Vegas Strip. *Tourism Management Perspectives* 23:41–52. doi:10.1016/j.tmp.2017.04.003.

48. Manek, A., P. Shenoy, M. Mohan, and K. Venugopal. 2017. Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. *World Wide Web* 20:135–54. doi:10.1007/s11280-015-0381-x.

49. Tripathy, A., A. Agrawal, and S. Rath. 2016. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications* 57:117–26. doi:10.1016/j.eswa.2016.03.028.

50. Ye, Q., Z. Zhang, and R. Law. 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications* 36:6527–35. doi:10.1016/j.eswa.2008.07.035.

51. Zainuddin, N., A. Selamat, and R. Ibrahim. 2018. Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence* 48:1218–1232.

52. Zavala R., Martínez P., Segura-Bedmar I. A Hybrid Bi-LSTM-CRF model for Knowledge Recognition from eHealth documents, 2018. SEPLN. P.65-70.
53. Sagot B. A multilingual collection of CoNLL-U-compatible morphological lexicons.: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 2018. P.1861.
54. Bodyanskiy E., Ryabova N., Zolotukhin O. Multilayer Adaptive Fuzzy Probabilistic Neural Network In Classification Problems Of Text Documents – Radio Electronics, Computer Science, Control Volume: 1(32), 2015, P. 39-45.
55. Shatalov O., Ryabova N. Named Entity Recognition Problem for Long Entities in English Texts, 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), 2021, Volume 1, pp. 76-79, doi: 10.1109/CSIT52700.2021.9648768.
56. Yadav V., Bethard S., A survey on recent advances in named entity recognition from deep learning models, 2018. Proc. 27th ICCL. P.2145-2158.
57. Understanding Evolution and Future of Natural Language Processing. URL: <https://www.xenonstack.com/blog/evolution-of-nlp>.
58. Julian Szymański julian.szymanski@eti.pg.gda.pl (2014) Comparative Analysis of Text Representation Methods Using Classification, Cybernetics and Systems, 45:2, 180-199, DOI: 10.1080/01969722.2014.874828
59. Wang, H., P. Yin, J. Yao, and J. Liu. 2013. Text feature selection for sentiment classification of Chinese online reviews. Journal of Experimental & Theoretical Artificial Intelligence 25:425–39. doi:10.1080/0952813X.2012.721139.
60. Schuckert, M., X. Liu, and R. Law. 2015. Hospitality and tourism online reviews: Recent trends and future directions. Journal of Travel & Tourism Marketing 32(5): 608–621. <https://doi.org/10.1080/10548408.2014>
61. Twitter Usage Statistics. URL: <https://www.internetlivestats.com/twitter-statistics/>. (дата звернення 21.11.2023)
62. The Hard Truth About Acquisition Costs (and How Your Customers Can Save You). URL: <https://blog.hubspot.com/service/customer-acquisition-study>.

ДОДАТКИ

ДОДАТОК А

SUMMARY

Kildei A. D. Development of a module for automated analysis, classification and management of product reviews using artificial intelligence – Masters-level Qualification Thesis. Sumy State University, Sumy, 2023.

This master's thesis delves into the realm of text classification and analysis, specifically employing artificial intelligence techniques. The research centers around the application of machine learning algorithms, particularly neural networks, to develop a web-based application to analyze and classify textual information effectively. The study's findings open new avenues for further exploration and development in this rapidly evolving field.

Keywords: artificial intelligence, text classification, sentiment analysis, machine learning, neural networks, product reviews analysis.

АНОТАЦІЯ

Кільдей А. Д. Розробка модуля для автоматизованого аналізу, класифікації та управління відгуками на товари за допомогою штучного інтелекту – Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2023 р.

Ця магістерська робота глибоко досліджує сферу класифікації та аналізу тексту, зокрема використовуючи техніки штучного інтелекту. Дослідження спрямоване на застосування алгоритмів машинного навчання, зокрема нейромереж, для ефективного побудови додатку для аналізу та класифікації текстової інформації. Результати дослідження відкривають нові можливості для подальших досліджень та розвитку в цьому сегменті.

Ключові слова: аналіз відгуків, аналіз тональності, класифікація тексту, машинне навчання, нейромережі, штучний інтелект.

ДОДАТОК Б

Код для створення та навчання моделі штучного інтелекту.

```

import json
import pandas as pd
import pickle
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

df = pd.read_excel('/content/reviews_dataset.xlsx')
df = df[df['reviews'].notna()]

df.head()
df["is_positive"].value_counts().plot(kind="bar");
# Split the data into training and testing sets
train_texts, test_texts, train_labels, test_labels = train_test_split(
    df['reviews'].values,
    df['is_positive'].values,
    test_size=0.2,
    random_state=42
)
# Tokenize and pad the text sequences
max_words = 32000 # Adjust based on your dataset size
max_len = 100 # Adjust based on the desired sequence length
tokenizer = Tokenizer(num_words=max_words, oov_token='<OOV>')
tokenizer.fit_on_texts(train_texts)
train_sequences = tokenizer.texts_to_sequences(train_texts)
train_padded = pad_sequences(train_sequences, maxlen=max_len, truncating='post')

test_sequences = tokenizer.texts_to_sequences(test_texts)
test_padded = pad_sequences(test_sequences, maxlen=max_len, truncating='post')
# Build the model
model = tf.keras.Sequential([

```

```

tf.keras.layers.Embedding(input_dim=max_words, output_dim=32, input_length=max_len),
tf.keras.layers.LSTM(64),
tf.keras.layers.Dense(1, activation='sigmoid')
])
# Compile the model with optimizer
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
# Train the model
early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)
history = model.fit(
    train_padded,
    train_labels,
    epochs=100,
    validation_data=(test_padded, test_labels),
    batch_size=64,
    callbacks=[early_stopping]
)
# Plot training history
def plot_training_history(history):
    # Plot training & validation accuracy values
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend(['Train', 'Validation'], loc='upper left')

    # Plot training & validation loss values
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend(['Train', 'Validation'], loc='upper left')

```

```
plt.show()
plot_training_history(history)
# Evaluate the model
loss, accuracy = model.evaluate(test_padded, test_labels)
print(f'Validation loss: {loss * 100:.2f}, Validation accuracy: {accuracy * 100:.2f}')
# Save the model
model.save('sentiment_model.h5')
# Save the tokenizer
with open('tokenizer.pkl', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
loaded_model = tf.keras.models.load_model('sentiment_model.h5')
with open('tokenizer.pkl', 'rb') as handle:
    tokenizer = pickle.load(handle)
def predict_sentiment(new_text):
    new_text = [new_text]
    new_text_sequence = tokenizer.texts_to_sequences(new_text)
    new_text_sequence = pad_sequences(new_text_sequence, maxlen=max_len, truncating='post')
    prediction = loaded_model.predict(new_text_sequence)
    is_positive = prediction[0][0] >= 0.6 # Assuming 0.5 is the threshold
    print(f'Sentiment: {is_positive}, Confidence: {prediction}')
```

ДОДАТОК В

Код для створення серверної частини веб-застосунку.

```

from flask import Flask, request, jsonify
from flask_cors import CORS
from pymongo import MongoClient
from bson import ObjectId
from dotenv import load_dotenv
import os
import json
import pickle
import pandas as pd
from datetime import datetime
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences

load_dotenv()
app = Flask(__name__)
CORS(app)
client = MongoClient(os.getenv('MONGO_URL'))
db = client['reviews']
loaded_model = tf.keras.models.load_model('tf_model/sentiment_model.h5')
with open('tf_model/tokenizer.pkl', 'rb') as handle: tokenizer = pickle.load(handle)

def map_fields(review, field_mapping):
    return {field_mapping.get(k, k): v for k, v in review.items()}

def analyze_sentiment_text(text):
    new_text = [text]
    new_text_sequence = tokenizer.texts_to_sequences(new_text)
    new_text_sequence = pad_sequences(new_text_sequence, maxlen=100)
    prediction = loaded_model.predict(new_text_sequence)
    is_positive = prediction[0][0] >= 0.5 # Assuming 0.5 is the threshold
    return 1 if is_positive else 0

@app.route('/api/reviews/import', methods=['POST'])
def import_reviews():
    try:
        file = request.files['file']

```

```

print("file", file)
if not file:
    return jsonify({'error': 'No file provided.'}), 400

if file.filename.endswith('.csv') or file.filename.endswith('.xlsx'):
    if file.filename.endswith('.csv'):
        reviews_df = pd.read_csv(file)
    elif file.filename.endswith('.xlsx'):
        reviews_df = pd.read_excel(file)

    # Drop the column with label '№' if present
    reviews_df = reviews_df.drop(columns=['№'], errors='ignore')

    # Adjust field mapping based on actual column names in DataFrame
    field_mapping = {'ПІБ': 'full_name', 'Контакти': 'contact', 'Відгук': 'review'}
    reviews_df = reviews_df.rename(columns=field_mapping)
elif file.filename.endswith('.json'):
    reviews_data = json.load(file)
    if not isinstance(reviews_data, list):
        return jsonify({'error': 'Invalid JSON format. Expecting a list of reviews.'}), 400

    # Standardize field names using a mapping
    field_mapping = {'ПІБ': 'full_name', 'Контакти': 'contact', 'Відгук': 'review'}
    reviews_df = pd.DataFrame([map_fields(review, field_mapping) for review in reviews_data])
else:
    return jsonify({'error': 'Unsupported file format. Only CSV, Excel (xlsx), and JSON are
supported.'}), 400

# Add a timestamp column to the DataFrame
reviews_df['timestamp'] = datetime.utcnow()
reviews_df['timeUpdated'] = ""

# Analyze sentiment for each review and add is_positive to the DataFrame
reviews_df['is_positive'] = reviews_df['review'].apply(lambda text: analyze_sentiment_text(text))
# This is where you would insert data into MongoDB if needed
reviews_to_insert = reviews_df.to_dict(orient='records')
db.reviews.insert_many(reviews_to_insert)

return jsonify(

```

```

        {'message': 'Reviews imported successfully', 'result': reviews_df.to_dict(orient='records')}, 201
    except Exception as e:

```

```

        return jsonify({'error': f'Error during review import: {str(e)}'}, 500

```

```

@app.route('/api/reviews', methods=['GET'])

```

```

def get_all_reviews():

```

```

    is_positive_param = request.args.get('is_positive')

```

```

    # Query to filter based on is_positive parameter

```

```

    if is_positive_param is not None:

```

```

        is_positive_value = int(is_positive_param)

```

```

        reviews = db.reviews.find({"is_positive": is_positive_value}).sort("timestamp", -1)

```

```

    else: reviews = db.reviews.find().sort("timestamp", -1)

```

```

    total_reviews = db.reviews.count_documents({})

```

```

    # Count positive and negative reviews

```

```

    positive_reviews_count = db.reviews.count_documents({"is_positive": 1})

```

```

    negative_reviews_count = db.reviews.count_documents({"is_positive": 0})

```

```

    # Simplify the format of _id field

```

```

    simplified_reviews = []

```

```

    for review in reviews:

```

```

        simplified_review = review

```

```

        simplified_review["_id"] = str(review["_id"])

```

```

        simplified_reviews.append(simplified_review)

```

```

    return jsonify({

```

```

        'totalReviews': total_reviews,

```

```

        'positiveReviews': positive_reviews_count,

```

```

        'negativeReviews': negative_reviews_count,

```

```

        'reviews': simplified_reviews,

```

```

    })

```

```

@app.route('/api/reviews/update_positive/<review_id>', methods=['PUT'])

```

```

def update_positive(review_id):

```

```

    try:

```

```

        review_id_object = ObjectId(review_id)

```

```

        review = db.reviews.find_one({'_id': review_id_object})

```

```

        print("review", review)

```

```

    if not review:

```

```
    return jsonify({'error': f'No review with id exists: {review_id}'}), 404

# Update the is_positive field to 1 and add timeUpdated field
current_time = datetime.utcnow()
updated_review = db.reviews.find_one_and_update(
    {'_id': review_id_object},
    {'$set': {'is_positive': 1, 'timeUpdated': current_time}},
    new=True
)

# Convert ObjectId to string before returning in JSON
updated_review['_id'] = str(updated_review['_id'])
return jsonify({'updatedReview': updated_review})

except Exception as e:
    return jsonify({'error': f'Error during update_positive: {str(e)}'}, 500

if __name__ == '__main__':
    app.run(debug=True, port=os.getenv('PORT'))
```


ДОДАТОК Г

Код для створення інтерфейсу веб-застосунку.

```
import React, {useEffect, useState} from 'react';
import axios from 'axios';
import {
  Chart as ChartJS,
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip,
  // Legend,
} from 'chart.js';
import {Bar} from 'react-chartjs-2';
import Table from "../components/Table.jsx";

const Home = () => {
  ChartJS.register(
    CategoryScale,
    LinearScale,
    BarElement,
    Title,
    Tooltip,
    // Legend
  );
  const [payload, setPayload] = useState({
    file: "",
  });

  const [reviewsList, setReviewsList] = useState(null);
  const [loading, setLoading] = useState(true);
  const [loadingText, setLoadingText] = useState("Завантаження відгуків...");
  const [icon, setIcon] = useState("loading");
  const [reviewsWithTimeUpdated, setReviewsWithTimeUpdated] = useState(0);
  const [reviewsUpdatedToday, setReviewsUpdatedToday] = useState(0);
```

```

const getReviewsList = () => {
  axios
    .get('http://localhost:8000/api/reviews')
    .then(function (response) {
      setReviewsList(response.data);
    })
    .catch(function (error) {
      alert(error.response.data.msg);
    })
    .finally(() => {
      setLoading(false)
      setLoadingText("Завантаження відгуків...")
      setIcon("loading")
    });
};

useEffect(() => {
  getReviewsList();
}, []);

const handleFileChange = (event) => {
  const file = event.target.files[0];
  setPayload((prev) => ({...prev, file}));
};

const onSubmit = async (event) => {
  event.preventDefault();
  try {
    setLoading(true);
    setLoadingText("Аналіз відгуків...")
    setIcon("analysing")
    const formData = new FormData();
    formData.append('file', payload.file);

    await axios.post('http://localhost:8000/api/reviews/import', formData, {
      headers: {

```

```

        'Content-Type': 'multipart/form-data',
      },
    });
    getReviewsList();
  } catch (error) {
    alert(error.response.data.error || 'An error occurred. Please try again.');
```

```

  } finally {
    setLoading(false); // Set loading state to false after file upload completes
  }
};

const options = {
  responsive: true,
  plugins: {
    legend: {
      position: 'top',
    },
    title: {
      display: true,
      text: 'Кількість відгуків',
    },
  },
};

const chartData = {
  labels: ['Позитивні відгуки', 'Негативні відгуки'],
  datasets: [
    {
      data: [
        reviewsList && reviewsList.positiveReviews ? reviewsList.positiveReviews : 0,
        reviewsList && reviewsList.negativeReviews ? reviewsList.negativeReviews : 0,
      ],
      backgroundColor: ['rgba(53, 162, 235, 0.5)', 'rgba(255, 99, 132, 0.5)']
    },
  ],
};
```

```
const columns = [
```

```

{
  accessor: 'index',
  label: '№',
  format: (id, value, index) => {
    return (<div className="index"><span>{index + 1}</span></div>)
  }
},
{accessor: 'full_name', label: 'ПІБ'},
{
  accessor: 'contact', label: 'Контакт',
  format: (record, value, index) => {
    let linkElement;
    if (value.includes('@')) {
      linkElement = <a href={`mailto:${value}`}>{value}</a>;
    } else {
      linkElement = <a href={`tel:${value}`}>{value}</a>;
    }
    return linkElement;
  }
},
{accessor: 'review', label: 'Відгук'},
{
  accessor: 'is_positive', label: 'Статус відгуку',
  format: (record) => {
    return (
      <div>
        {record.is_positive === 1 && record.timeUpdated === "" ? (
          <span className="green">Позитивний</span>
        ) : record.is_positive === 1 && record.timeUpdated !== "" ? (
          <span className="progressing">Оброблені</span>
        ) : record.is_positive === 0 ? (
          <span className="red">Негативний</span>
        ) : null}
      </div>
    );
  }
},
]

```

```

const currentDate = new Date();
const todayDateString = currentDate.toDateString();

useEffect(() => {
  if (reviewsList && reviewsList.reviews) {
    const withTimeUpdated = reviewsList.reviews.filter(review => review.timeUpdated !== "").length;
    const updatedToday = reviewsList.reviews.filter(review => {
      const reviewDate = new Date(review.timeUpdated);
      return reviewDate.toDateString() === todayDateString;
    }).length;

    setReviewsWithTimeUpdated(withTimeUpdated);
    setReviewsUpdatedToday(updatedToday);
  }
}, [reviewsList, todayDateString]);

return (
  <div className="container reviews">
    { /*<div className="title">*/ }
    { /* <h1>Головна сторінка</h1>*/ }
    { /*</div>*/ }
    { loading ? (
      <div className="loader">
        <img src={` ${icon}.svg` } alt="loader"/>
        <p>{loadingText}</p>
      </div>
    ) : (
      <div className="reviews__block">
        <div className="reviews__title">
          <form className="reviews__form form" onSubmit={onSubmit}>
            <div className="form__group">
              <h3>Імпортувати відгуки</h3>
            </div>
            <div className="form__group">
              <input

```

```

        className="form__control"
        type="file"
        name="file"
        id="file"
        onChange={handleFileChange}
        placeholder="File"
    />
</div>
<div className="form__group">
    <button className={`btn form__control btn-add ${payload.file === "" ? "disabled" :
""}` `}
        type="submit">
        Аналізувати відгуки
    </button>
</div>
</form>
<div className="reviews__total">
    <h2>Загальна кількість відгуків: <span
        className="bold">{reviewsList && reviewsList.totalReviews ?
reviewsList.totalReviews : 0}</span></h2>
    <h3 className="green">Загальна кількість оброблених відгуків: <span
        className="bold">{reviewsWithTimeUpdated}</span></h3>
    <h3 className="progressing">Кількість оброблених відгуків за сьогодні: <span
        className="bold">{reviewsUpdatedToday}</span>
    </h3>
</div>
</div>
{reviewsList &&
    <div>
        <Bar options={options} data={chartData}/>
    </div>
}
</div>
<div className="reviews__table">
    {reviewsList?.totalReviews === 0 ?
    <div className="reviews__table-title">
        <h1>Інформація по відгукам відсутня</h1>
    </div>

```

```
:
<
<div className="reviews__table-title">
  <h1>Всі відгуки</h1>
</div>
{
  reviewsList !== null ?
    <Table rows={reviewsList?.reviews} columns={columns} cleanRows={true}
      rowStyle="border" className="product__list w-100"/>
    :
    ""
  }
</>
}
</div>
</>
)
}
</div>
)
;
};

export default Home;
```