Міністерство освіти і науки України

Луцький національний технічний університет
(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій
(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СИСТЕМА ОБРОБКИ ТЕКСТОВИХ ДАНИХ ЗАСОБАМИ МОВИ PYTHON

TEXT DATA PROCESSING SYSTEM USING THE PYTHON LANGUAGE

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Кравчук Юрій Олегович

(підпис)

Керівник:
к.т.н., доцент
Христинець Наталія Анатоліївна

(підпис)

Кваліфікаційну роботу
допущено до захисту
«_____» червня 2024 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та кібербезпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ проф. Н.Чернящук
« 10 » 01 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кравчуку Юрію Олеговичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Система обробки текстових даних засобами мови Python

Керівник роботи к.т.н., доцент Христинець Наталія Анатоліївна

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2024р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях питань обробки текстових даних методами машинного навчання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Аналітичний огляд питань обробки текстових даних
Огляд інструментів розробки та використаних технологій
Розробка системи обробки текстових даних

5. Перелік графічного (ілюстративного) матеріалу:
Аналіз моделі нейронних мереж
Інструменти та бібліотеки Python
Аналіз та підготовка набору даних
Реалізація алгоритму
Тестування роботи програми

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз проблеми за темою роботи та постановка завдань дослідження	*Христинець Н.А., доцент*		
Теоретичне дослідження та практична реалізація	*Христинець Н.А., доцент*		
Програмна реалізація нейронної мережі та аналіз текстових даних	*Христинець Н.А., доцент*		
Нормоконтроль	*Багнюк Н.В., доцент*		
Гарант ОП	*Лавренчук С.В., доцент*		
Показник запозичень тексту	***___%***		
Академічна доброчесність	*Міскевич О.І.,асистент*		

7. Дата видачі завдання 10.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Вивчення літератури*	до 15.02.2024 р.	Виконано
2.	*Огляд предметної області*	до 15.03.2024 р.	Виконано
3.	*Інсталяція програмного забезпечення та встановлення бібліотек*	до 04.05.2024 р.	Виконано
4.	*Створення структури проекту*	до 07.05.2025 р.	Виконано
5.	*Формування списку використаних джерел*	до 10.05.2024 р.	Виконано
6.	*Аналіз наборів даних та тестування програми*	до 15.05.2024 р.	Виконано
7.	*Оформлення ілюстративного матеріалу*	до 20.05.2024 р.	Виконано
8.	*Нормоконтроль*	до 01.06.2024 р.	Виконано
9.	*Інструментальна перевірка на академічний плагіат*	до 04.06.2024 р	Виконано
10.	*Представлення кваліфікаційної роботи бакалавра до захисту*	до 22.06.2024 р.	Виконано

Здобувач вищої освіти _____ Кравчук Ю.О.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ Христинець Н.А.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Кравчук Ю.О. Система обробки текстових даних засобами мови Python. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 40 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячений огляду інструментів обробки даних, аналізу моделей нейронних мереж огляду програмних можливостей для класифікації текстових даних.

В другому розділі здійснено огляд інструментів розробки та використаних технологій.

Третій розділ присвячено опису розроблених програм обробки текстових даних, виконання задач класифікації, векторизації тексту, розробці штучної багатошарової нейронної мережі.

Ключові слова: аналіз тексту, класифікація, нейронна мережа, текстові дані, машинне навчання, обробка природньої мови.

ANNOTATION

Kravchuk Y.O. Text data processing system using the Python language. Manuscript.

Bachelor's qualifying thesis of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 40 p.

The qualification work consists of an introduction, three sections, conclusions, and a list of used sources.

The first chapter is dedicated to the review of data processing tools, analysis of neural network models, review of software capabilities for text data classification.

In the second section, an overview of the development tools and used technologies was carried out.

The third section is devoted to the description of the developed text data processing programs, the implementation of classification tasks, text vectorization, and the development of an artificial multilayer neural network.

Keywords: text analysis, classification, neural network, text data, machine learning, natural language processing.

ЗМІСТ

ВСТУП

Актуальність теми обробки текстових даних на сьогоднішній день важко переоцінити. Інформаційний прогрес, спричинений швидким розвитком технологій, зробив аналіз тексту надзвичайно важливими завданнями у багатьох галузях, від науки та бізнесу до соціальних мереж і медіа. Python, завдяки своїй простоті використання, великій кількості бібліотек та інструментів, є чудовим вибором для реалізації систем обробки текстових даних. Створення системи, яка буде використовувати Python для обробки текстової інформації, не лише відповідає сучасним вимогам, але й відкриває нові можливості в аналізі та розумінні великих обсягів тексту. Наприклад, застосування системи для автоматичної категоризації кінострічок за жанрами може мати велике значення для індустрії розваг, а також для створення персоналізованих рекомендацій для користувачів на основі їхніх індивідуальних вподобань. Такий проект не лише використовує передові методи обробки природної мови, але і відповідає на практичні потреби в сфері аналізу текстових даних, що робить його актуальним та перспективним у контексті сучасних вимог до обробки інформації.

Метою роботи є розробка та реалізація системи обробки текстових даних методами машинного навчання.

Об’єкт дослідження – система обробки текстових даних.

Предмет дослідження – методи, алгоритми та інструменти обробки текстових даних, реалізовані засобами мови програмування Python.

Завдання, які необхідно виконати:

– провести аналітичний огляд з питань систем обробки текстових даних та дослідити, які програмні засоби будуть використані для створення програми;

– провести аналіз моделей нейронних мереж та виявити дії, які включає мета-аналіз текстових даних;

– дослідити інструменти програмного середовища для роботи з нейронними мережами та визначити бібліотеки, необхідні для реалізації машинного навчання;

– повести аналіз та підготовку набору даних та визначити середовище для реалізації тестових експериментів;

– побудувати багатошарову нейронну мережу для аналізу даних, дослідити доцільність кожного шару,

– протестувати роботу програми на різних наборах даних, перевірити достовірність роботи програми.

РОЗДІЛ 1
АНАЛІТИЧНИЙ ОГЛЯД ПИТАНЬ ОБРОБКИ ТЕКСТОВИХ ДАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

1.1 Огляд інтрументів обробки даних

Для обробки даних в мережі Інтернет сьогодні найчастіше використовують методи машинного навчання (Machine learning, ML). Ці методи дозволяють автоматично аналізувати великі обсяги даних, виявляти закономірності та робити прогнози. Машинне навчання є частиною штучного інтелекту і використовує алгоритми для навчання комп'ютерів розпізнавати закономірності в даних і робити передбачення або рішення на основі цих даних.

Сьогодні ML відіграє ключову роль у багатьох сферах, дозволяє автоматизувати складні процеси та аналізувати великі обсяги інформації [1].

Наприклад, в медицині машинне навчання допомагає в діагностиці захворювань та прогнозуванні лікувальних результатів [2-3]. У фінансовій галузі [4] його використовують для виявлення шахрайства та управління ризиками. В роздрібній торгівлі ML застосовують для персоналізації рекомендацій та управління запасами, а в автомобільній індустрії машинне навчання є основою для розвитку автономних транспортних засобів [5]. Також, ML широко використовують в маркетингу для аналізу поведінки клієнтів та оптимізації рекламних кампаній. Інші важливі застосування включають обробку природної мови, розпізнавання зображень та прогнозування кліматичних змін.

Питання обробки даних методами машинного навчання вивчають багато українських та зарубіжних науковців. Так, в роботі [6] зазначається, що для обробки текстових даних найчастіше застосовують «базові моделі лінійної та нелінійної регресії, класифікації (зокрема логістична регресія, методи опорних векторів та k-найближчих сусідів, Байєсова класифікація, дерева рішень і випадковий ліс), кластеризації (ієрархічна і k-середніх), а також методи побудови асоціативних правил». Автори роботи [7] вказують на те, що в процесі аналізу даних самі об'єкти визначаються, як «матеріальний цілісний об'єкт, призначений для виконання певної функції в заданих умовах, технічно реалізований на основі

впорядкованої за номенклатурою, скінченної множини функціонально взаємозалежних, структурно взаємопов’язаних функціональних елементів, які технологічно взаємодіють».

Розглянуто програмні бібліотеки, інструментарії мов програмування для збору і аналізу текстових даних. Виявлено, що одним з основних інструментів для маніпуляції з даними у Python є Pandas [8], який широко використовується для завантаження, очищення та обробки даних. За допомогою Pandas можна завантажувати дані з різних джерел, таких як CSV, Excel, SQL бази даних, використовуючи функції pd.read_csv, pd.read_excel та pd.read_sql. Завантажені дані зберігаються у DataFrame, що дозволяє зручно працювати з таблицями даних.

Для очищення даних Pandas надає потужні інструменти. Можна видаляти або заповнювати відсутні значення за допомогою методів dropna та fillna. Також можна видаляти дублікатні рядки, використовуючи метод drop_duplicates. Pandas дозволяє з легкістю обробляти текстові дані, зокрема, видаляти пробіли за допомогою методу str.strip та змінювати регістр тексту з str.lower або str.upper.

Очищення даних часто включає перетворення типів даних. Метод astype дозволяє змінювати типи колонок, що є корисним для правильного аналізу. Pandas також підтримує роботу з датами, дозволяючи перетворювати текстові строки у datetime об’єкти за допомогою pd.to_datetime. Крім того, можна легко фільтрувати дані, використовуючи умовні вирази.

Агрегування даних здійснюється за допомогою групування з функцією groupby, що дозволяє обчислювати статистичні показники для груп даних. Сортування даних у DataFrame здійснюється за допомогою методу sort_values, що дозволяє впорядковувати дані за однією або декількома колонками. Для більш складних обчислень Pandas надає функцію apply, яка дозволяє застосовувати користувацькі функції до даних.

Зважаючи на це, використання бібліотек Pandas є доцільним у виконанні кваліфікаційної роботи.

В роботах авторів Дубровіна В., Матвєєва Т., Могильної М., Татарникова А. розглядаються інструменти для обробки текстових даних методами машинного навчання. Зазначається, що саме обробка текстів є ключовою складовою багатьох

сучасних додатків, від чат-ботів до аналізу настроїв у соціальних мережах. Першим кроком у цьому процесі є завантаження та очищення даних. Інструментарій Pandas є основним інструментом для маніпуляції з даними у Python, він дозволяє легко завантажувати текстові дані з різних джерел, таких як CSV-файли або бази даних. У роботі [9] зазначено, що нормалізація тексту включає перетворення тексту в нижній регістр за допомогою str.lower для забезпечення консистентності, а токенізація, тобто розбиття тексту на окремі слова або токени, найкраще відбувається за допомогою застосування методу str.split, або використовуючи бібліотеки, такі як NLTK чи spaCy. В дослідженні цих авторів сказано, що по алгоритму далі слідує видалення стоп-слів, які є загальновживаними словами і не несуть значного змістового навантаження.

Видобування цінної текстової інформації з набору різних документів є досить монотонною роботою. В роботі [10] досліджено, що застосування попередньої техніки для аналізу тексту скорочує час і зусилля для пошуку відповідних образів для аналізу та прийняття рішень. Схематично на рисунку 1.1 зображено взаємозв’язки різних сфер інтелектуального аналізу текстових даних.

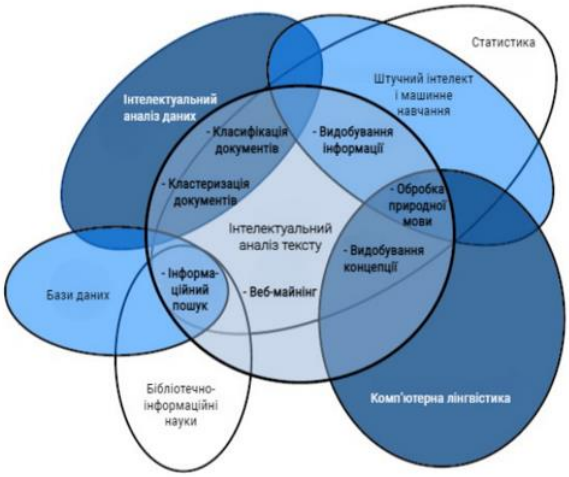

Рисунок 1.1 – Діаграма взаємозв’язків інтелектуального аналізу [10]

З огляду даної схеми, інтелектуальний аналіз тексту загалом виступає, як процес вилучення корисних патернів для дослідження текстових джерел. Це міждисциплінарна галузь, яка поєднує пошук інформації, інтелектуальний аналіз

даних, машинне навчання, статистику та комп'ютерну лінгвістику. Інтелектуальний працює з текстами природною мовою, які зберігаються в напів-структурованому або неструктурованому форматі.

Загалом, застосування інтелектуального аналізу включає пошукові системи, системи управління взаємовідносинами з клієнтами, фільтрацію електронної пошти, аналіз продуктових пропозицій, виявлення шахрайства та аналітику соціальних мереж тощо. Також такий аналіз використовують для аналізу настроїв, виділення ознак, прогнозування та аналізу тенденцій.

1.2 Аналіз моделей нейронних мереж

На початковому етапі важливо правильно підібрати архітектуру мережі з урахуванням конкретного завдання і зазначити, яке ця модель повинна вирішувати питання аналізу текстових даних. Аналіз моделей нейронних мереж – це комплексний процес оцінки та вдосконалення штучних нейронних мереж. Він включає в себе декілька ключових етапів, що охоплюють вибір архітектури мережі, навчання та оптимізацію моделі, оцінку її точності та стійкості, а також інтерпретацію отриманих результатів. Це може бути вибір між різними типами нейронних мереж, такими як перцептрони, конволюційні мережі (CNN), рекурентні мережі (RNN) або трансформери, в залежності від характеру вхідних даних та задачі. Після цього формується сам процес машинного навчання, де модель навчається на тренувальних даних з використанням підібраних параметрів. Важливим є вибір методу оптимізації та гіперпараметрів, таких, як швидкість навчання та розмір міні-пакетів.

Оцінка моделі включає в себе розгляд різних метрик, таких як точність, втрати, precision, recall та F1-score. Ці метрики відображають, наскільки добре модель вирішує своє завдання та як вона впорядковується на нових даних. Наприклад, кількість вхідних шарів для аналізу тексту розділяється на пікселі [11] і коли ми говоримо про рукописні символи у контексті бази даних NIST, то ми маємо на увазі набір даних, який містить зображення рукописних цифр, які були розпізнані та представлені у вигляді дискретних пікселів (рис. 1.2).

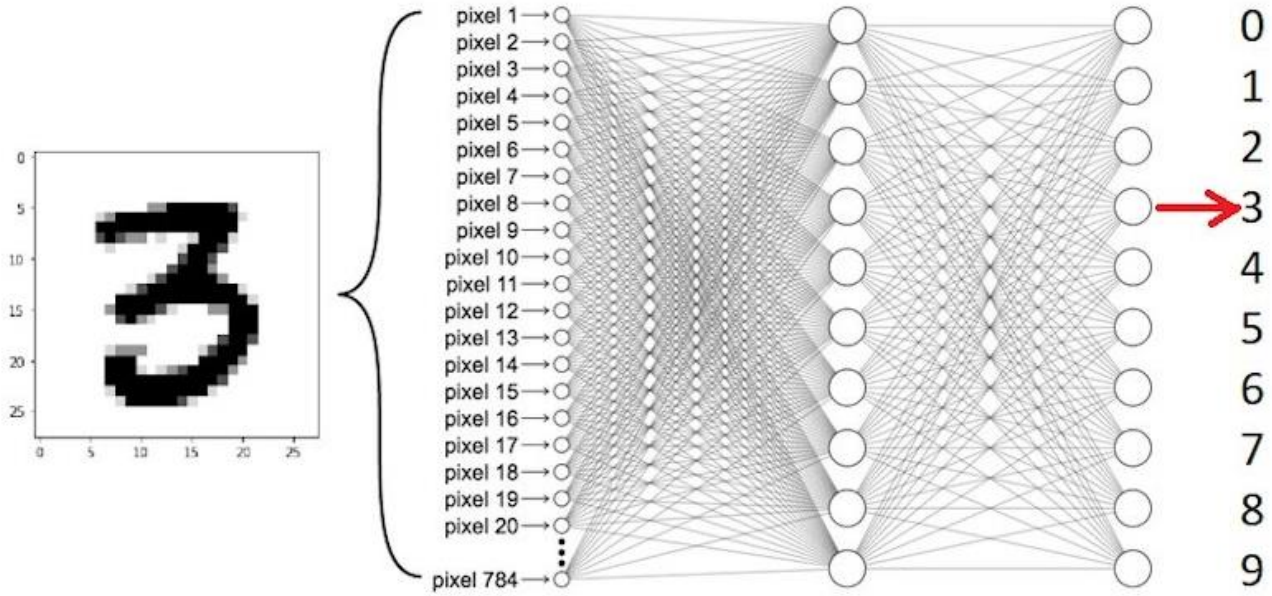

Рисунок 1.2 – Набір даних та вхідні шари для фрагменту рукописного тексту [11]

Крім того, аналіз моделей нейронних мереж включає в себе візуалізацію процесу навчання та результатів, а також інтерпретацію роботи мережі, наприклад, шляхом вивчення впливу вхідних функцій на вихідні прогнози (feature importance).

Тобто, загальний мета-аналіз текстових даних включає такі дії, як крос-валідація для оцінки стабільності моделі, зміна гіперпараметрів для поліпшення її роботи і збір додаткових даних для покращення точності прогнозування [12]. Такий підхід дозволяє досягти оптимальних результатів у розробці та застосуванні нейронних мереж для різноманітних завдань машинного навчання.

1.3 Огляд програмних можливостей для класифікації текстових даних

З аналізу інтернет-джерел, з огляду інструментарію систем побудови штучного інтелекту та за результатами аналізу наукових літературних досліджень, Python є ідеальним для машинного навчання та виконання інших спеціальних програмних завдань без шкоди для надійності системи. Аргументуємо даний вибір.

У Python є вбудована велика стандартна бібліотека, яка підтримує численні загальні завдання програмування, такі як підключення до веб-серверів, пошук тексту за допомогою регулярних виразів, а також читання та редагування файлів,

підтримується понад 125 000 сторонніх бібліотек. Python працює на різних платформах, таких як MacOS, Windows, Linux та Unix, а неофіційні збірки доступні також для Android та iOS.

Для порівняння, розглянемо інші популярні мови серед досліджень у галузі статистики та аналізу даних. Мова R має багато спеціалізованих пакетів для статистичного аналізу, візуалізації даних та машинного навчання, таких як ggplot2, caret, та інші. R відома своїми могутніми засобами візуалізації даних. Пакети, такі як ggplot2, надають зручні інструменти для побудови якісних інформативних графіків, що допомагають аналізувати результати моделей та взаємозв'язки між даними. R також підтримує багато пакетів для обробки текстових даних, таких як tm (Text Mining), quanteda, які включають в себе функції для токенізації, векторизації, аналізу настроїв і виявлення тем у тексті. Також є можливість інтегрувати R з іншими мовами програмування, такими як Python, що дозволяє розширювати функціональність і використовувати обидва середовища для різних аспектів аналізу даних. Проте, одним з головних недоліків R, на нашу думку, є його швидкодія для обробки великих обсягів даних. В порівнянні з іншими мовами, такими як Python або Java, R може бути помітно повільнішим у виконанні складних обчислень або обробці великих даних. Також в R можуть виникати проблеми з обмеженням пам'яті при обробці великих наборів даних. Оскільки R завантажує всі дані в пам'ять, це може обмежувати масштабованість програм і обробки даних.

Java є наступною програмною альтернативою для машинного навчання. Вона відома своєю високою ефективністю і швидкодією, що робить її ідеальним вибором для великих систем машинного навчання. Бібліотека Weka використовується для класифікації, кластеризації, регресії та візуалізації даних, надаючи розширені інструменти для аналізу даних, а бібліотека Deeplearning4j спеціалізується на розробці глибоких нейронних мереж, забезпечуючи підтримку різних алгоритмів навчання та оптимізації. Java також використовується для високопродуктивних систем обробки даних, де важлива робота з великими обсягами і реальним часом обробки. Серед недоліків Java саме в контексті машинного навчання варто зазначити, що порівняно з іншими мовами, такими як Python або R, розробка та прототипування моделей може бути складнішою і вимагати більше коду через

більш формальну синтаксичну структуру Java. Крім того, менша кількість спеціалізованих бібліотек та інструментів для машинного навчання в порівнянні з Python також може ускладнювати розробку в цій галузі.

Аналіз проведених досліджень сформовано в таблиці 1.1.

Таблиця 1.1 – Програмні засоби для реалізації машинного навчання

Мова програмування	R	Java	Python
Вибір бібліотек та інструментів для машинного навчання	обмежено	достатньо	найбільше
Прототипування та розробка моделей	підтримується	підтримується частково	підтримується
Швидкодія	достатня	висока	висока
Інтеграція з іншими мовами та платформами	достатня	достатня	висока

З огляду на дослідження, мова програмування Python є однією з найбільш популярних серед вчених у галузі машинного навчання та аналізу текстових даних завдяки своїй простоті, гнучкості та широкому спектру наявних бібліотек. Крім того, завдяки активній спільноті розробників та дослідників, Python постійно оновлюється та розвивається, що робить його ідеальним вибором для впровадження сучасних методів машинного навчання та аналізу текстових даних. Таким чином, обґрунтовано вибір мови програмування Python для розробки у кваліфікаційній роботі.

РОЗДІЛ 2
ОГЛЯД ІНСТРУМЕНТІВ РОЗРОБКИ ТА ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

«В останні роки спостерігається значне зростання обсягу неструктурованих даних, які представлені у формах, таких як інформаційні пабліки, блоги та соціальні мережі» [13]. Цей великий потік інформації містить у собі безліч текстових матеріалів, від коротких повідомлень до довгих публікацій, що представляють значний інтерес для аналізу та витягнення цінних знань. Відповідно до цього зростає значення технологій обробки природної мови (Natural Language Processing, NLP), які надають найбільш ефективний інструментарій для автоматизованого аналізу і розуміння текстових даних. NLP використовується для виявлення патернів у тексті, класифікації документів, витягнення іменованих сутностей, аналізу сентименту, машинного перекладу та багатьох інших завдань, що допомагають зрозуміти і структурувати масиви неструктурованої інформації. Такий підхід дозволяє компаніям, дослідникам і організаціям ефективно використовувати великі обсяги даних для прийняття інформованих рішень, розробки персоналізованих послуг та покращення стратегій взаємодії зі споживачами.

2.1 Використання бібліотек NLP

Бібліотеки NLP (англ. Natural Language Processing), або обробка природної мови, є окремим напрямком технологій в сфері штучного інтелекту, який зосереджується на тому, як комп’ютер може розуміти, аналізувати та генерувати природну мову, якою користується людина. У науковій спільноті відомий тест Тюрінга, який передбачає, що якщо людина не може відрізнити спілкування з комп’ютером від спілкування з іншою людиною, то комп’ютер можна вважати штучним інтелектом. Саме цей тест став відправною точкою для розвитку NLP.

Дані бібліотеки використані в роботі для токенізації і препроцесингу тексту. На початковому етапі обробки тексту бібліотеки NLP розбивають текст на окремі компоненти (токени), такі як слова або фрази. Це дозволяє ефективніше подальше

його аналізу. Потім відбувається перетворення текстових даних в числові вектори, що в подальшому будуть використані для навчання моделей машинного навчання. Використовуються методи TF-IDF або векторизація на основі вбудованих представлень (наприклад, Word2Vec або GloVe). Даний інструментарій використовується для вилучення іменованих сутностей (NER). Це підхід в NLP, де моделі виявляють і витягують іменовані сутності, такі як імена людей, місця або організації з текстових документів. У випадку розробки кваліфікаційної роботи у якості NER вибираються мітки жанрів з текстових полів (рис. 2.1).

```
25  # Отримання міток
26  labels = data[['crime', 'fantasy', 'history', 'horror', 'psychology', 'romance', 'science', 'sports', 'thriller', 'travel']]
```

Рисунок 2.1 – Інструментарій бібліотеки NLP для полів міток

Вказаними методами NLP можна аналізувати тон і емоційне забарвлення текстів. Це корисно для виявлення настроїв і відгуків користувачів щодо певного продукту чи послуги, а також створювати системи для автоматичного перекладу тексту з однієї мови на іншу, що базуються на статистичних моделях або глибоких нейронних мережах.

До бібліотек NLP включено також NLTK – провідну платформу для створення програми Python для роботи з людською мовою. Вона надає простий у використанні інтерфейс для більш ніж 50 вбудованих семантичних словників, корпусів текстів і навчених моделей, таких як WordNet, а також набір бібліотек для обробки тексту для класифікації, токенізації, формування основ, тегування, синтаксичного аналізу та семантичного вимірювання [14]. NLTK вважають «відмінним засобом для навчання та роботи з комп’ютерною лінгвістикою за допомогою Python» і «потужною бібліотекою для роботи з природною мовою» [15].

Бібліотека NLTK пропонує широкий спектр інструментів для обробки тексту, включаючи:

– лематизацію;
– стемінг;
– виокремлення частин мови;
– аналіз сентименту;

– побудову граматик.

Основними перевагами NLTK є її великий набір інструментів та багаті ресурси бібліотеки. Недоліками є складність (через велику кількість функцій та можливостей бібліотека може здатися складною для новачків, що може вимагати більше часу на освоєння), залежність від іншим бібліотек (деякі функції NLTK залежать від інших бібліотек, що може ускладнювати встановлення та налаштування), та, не зважаючи на це, NLTK залишається потужним інструментом для обробки природньої мови.

Наступним використаним інструментарієм Python є Scikit-learn. Це одна з найпопулярніших бібліотек для машинного навчання, яка пропонує різноманітні інструменти для аналізу даних і створення моделей машинного навчання. Scikit-learn є відкритою та безкоштовною, і це стало також одним із аргументів її використання в кваліфікаційній роботі.

Scikit-learn має високі позиції в Kaggle-змаганнях (рис. 2.2).

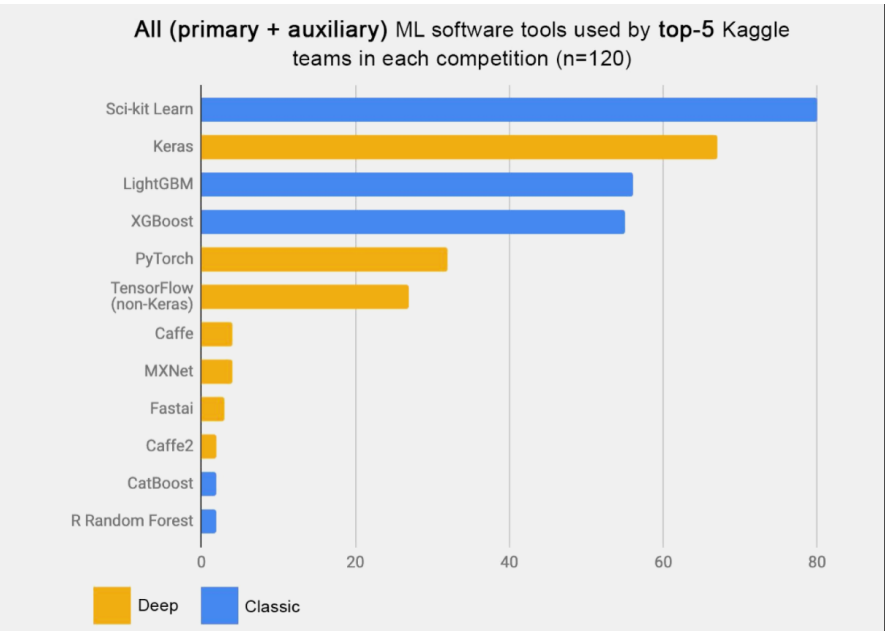

Рисунок 2.2 – Статистика використання бібліотек в машинному навчанні на змаганнях Kaggle [16]

Однією з причин популярності Scikit-learn є її широкий функціонал та зручність використання. Бібліотека містить готові реалізації багатьох алгоритмів машинного навчання, які можна легко застосовувати до різних задач. Крім того,

Scikit-learn має детальну документацію та велику кількість навчальних матеріалів, що дозволяє навіть початківцям швидко освоїти основи роботи з нею. Гнучкість бібліотеки також дозволяє інтегрувати її з іншими інструментами для наукових обчислень та обробки даних, такими як NumPy, Pandas та Matplotlib. Іншою важливою особливістю Scikit-learn є підтримка різних типів задач машинного навчання, таких як класифікація, регресія, кластеризація та зниження розмірності. Завдяки цьому, користувачі можуть легко знаходити оптимальні рішення для своїх конкретних задач. Scikit-learn також підтримує техніки передобробки даних, такі як нормалізація, стандартизація та обробка пропущених значень, що дозволяє підготувати дані для аналізу та моделювання. Усе це робить Scikit-learn незамінним інструментом для дослідників, інженерів та науковців у сфері машинного навчання.

В кваліфікаційній роботі використано метод векторизації через Scikit-learn, фрагмент програмного коду для реалізації цього методу наведено на рисунку 2.3.

```
19  # Ініціалізація векторизатора
20  vectorizer = TfidfVectorizer()
21  # Векторизація тексту
22  X_tfidf = vectorizer.fit_transform(data['X'])  # стовпчик з текстом відгуків називається 'X'
```

Рисунок 2.3 – Фрагмент коду векторизації з допомогою Scikit-learn для обробки тексту

Основною функцією Scikit-learn є класифікація, яка є одним з типів завдань машинного навчання. Класифікація полягає у призначенні кожному об’єкту одного з певного фіксованого набору міток чи класів, на основі вхідних даних. Наприклад, класифікація може використовуватися для передбачення того, чи буде електронний лист спамом чи ні, на основі його вмісту та інших ознак.

Scikit-learn має різноманітні алгоритми класифікації, серед яких є такі методи, як:

– логістична регресія;
– метод опорних векторів;
– «випадковий ліс»;

– Баєсівський класифікатор;

– метод k-найближчих сусідів.

Ці алгоритми можуть бути використані для вирішення різноманітних задач класифікації в залежності від особливостей даних та потреб користувача.

Загалом, етап використання векторизації через Scikit-learn є ключовим етапом в підготовці текстових даних для подальшого аналізу та машинного навчання, що дозволяє ефективно використовувати текст у різних аналітичних завданнях.

2.2 Інструменти для роботи з нейронними мережами

Раніше процес виконання завдань машинного навчання був вкрай трудомістким та заплутаним через необхідність вручну кодувати всі алгоритми та статистичні формули. Це призводило до неефективності процесу. В сучасний період ця задача стала набагато простішою та ефективнішою, завдяки використанню різних бібліотек та модулів Python. Однією з передумов цього є наявність великої кількості вбудованих рішень цієї мови, які значно полегшують роботу з машинним навчанням. До бібліотек реалізації методів машинного навчання мовою Python належать: TensorFlow, PyTorch, Keras, Seaborn, Matplotlib, Pandas, Numpy, Caffe. Переважна більшість цих бібліотек використана в кваліфікаційній роботі (рис. 2.4).

Необхідність і значущість застосування цих бібліотек пояснюється наступним. TensorFlow дозволяє реалізувати програмне забезпечення з відкритим кодом для чисельних розрахунків з використанням графів потоку даних. Вузли графу представлені у вигляді математичних операцій, в той час як ребра графу представляються багатовимірними масивами даних (тензорами), що передаються між вузлами [17].

```
import numpy as np
import pandas as pd
from tensorflow.keras.callbacks import EarlyStopping
from nltk.stem import SnowballStemmer
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.models import Sequential
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.base import BaseEstimator, TransformerMixin
```

Рисунок 2.4 – Бібліотеки проекту для реалізації машинного навчання

Основні переваги бібліотеки TensorFlow перед іншими бібліотеками для ML подані в таблиці 2.1.

Таблиця 2.1 – Порівняльна таблиця бібліотек Python

Критерій	TensorFlow	PyTouch	Scikit-learn
Швидкість розробки	Широкий функціонал може вимагає час на освоєння, проте досвідчені спеціалісти високо оцінюють цей інструмент.	Перевагою є «дружність» до користувачів , що може полегшити розробку на початку.	Перевагою є простий і зручний інтерфейс, що пришвидшує вирішення задач
Гнучкість	Може знаходити застосування у різноманітних завданнях з машинним навчанням	Гнучкість аналогічна до TensorFlow	Має широкий спектр інструментів для ML
Застосування	Популярна в індустрії серед звичайних користувачів та серед великих компаній	Отримавши визнання у дослідників, цей інструмент також стає все більш популярним	Використовується переважно для вирішення простих задач ML

Аналіз сформованих критеріїв дає можливість візуалізації результатів, водночас забезпечуючи гнучкість та підтримку багатьох мов програмування

(TensorFlow спочатку був розроблений для Python, але зараз підтримує також C++, Java, JavaScript, Go та інші мови, що робить його доступним для широкого кола розробників).

TensorFlow використовується для вирішення багатьох задач, включаючи обробку зображень, NLP, рекомендаційні системи, медичній діагностиці та в інших сферах. Компанії, такі як Google, Airbnb, Uber та інші, активно використовують TensorFlow для розробки своїх продуктів та сервісів.

Інструмент Keras, як компонент Python для машинного навчання з відкритим вихідним кодом, забезпечує простий і зручний інтерфейс для створення та навчання нейронних мереж [18]. Keras дозволяє легко створювати різні архітектури нейронних мереж, включаючи звичайні згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), а також комбінаційні моделі.

Однією з головних переваг Keras у порівнянні з TensorFlow (рис. 2.5) є простота використання. Вона має чіткий і стислий API, який дозволяє швидко створювати, налаштовувати та навчати моделі без глибоких знань теорії нейронних мереж. Крім того, Keras надає можливість працювати з різними нейронними мережами в стилі Sequential або функціональному API, що розширює його можливості у створенні складних моделей.

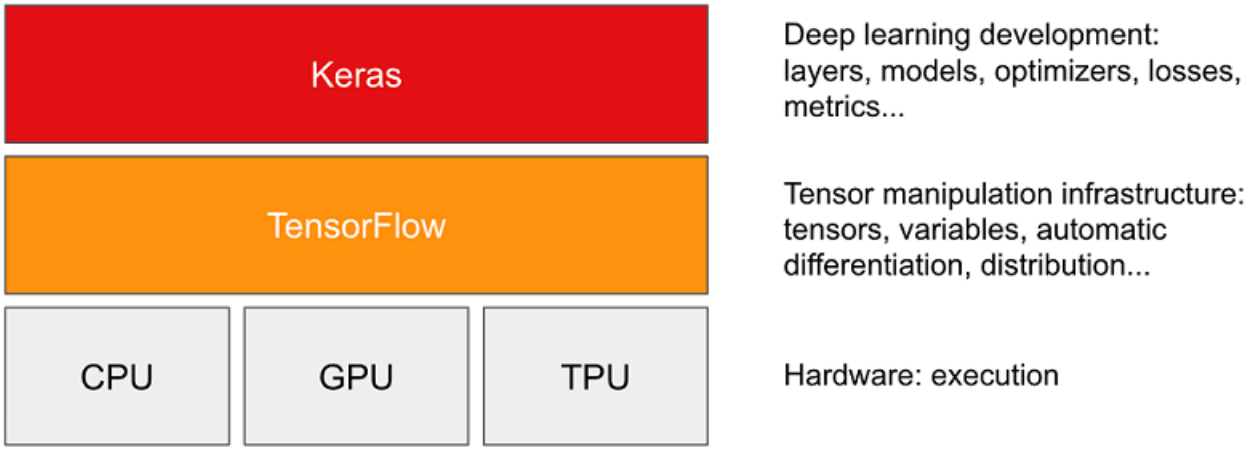

Рисунок 2.5 – Порівняння використання бібліотек Keras і TensorFlow [19]

Важливою особливістю Keras є його вбудована підтримка для різних фреймворків глибинного навчання, зокрема TensorFlow, Theano та Microsoft

Cognitive Toolkit (CNTK). Завдяки цьому Keras є зручним інструментом для застосування як високорівневого інтерфейсу для цих фреймворків та зручного комбінування їх функціональності для розробки потужних моделей.

РОЗДІЛ 3
РОЗРОБКА СИТЕМИ ОБРОБКИ ТЕКСТОВИХ ДАНИХ

3.1 Аналіз та підготовка набору даних

Для навчання штучної нейронної мережі було використано набір даних Book Genre Prediction, розміщений на сайті баз даних kaggle.com [20]. Тобто, в роботі було обрано веб-платформу, яка дозволяє генерувати та завантажувати дані для подальшої обробки. Ця платформа широко використовується як для спеціалістів з даних та дослідників у галузі машинного навчання, так і надає численні можливості для навчання та роботи з базами різних типів. На платформі регулярно організовуються змагання з машинного навчання, де учасники можуть вирішувати різноманітні завдання та вигравати призи, що стимулює розвиток нових підходів та методів. Kaggle пропонує доступ до великої кількості відкритих наборів даних, що дозволяє користувачам тренувати та тестувати свої моделі на реальних даних. Інтегроване середовище Kaggle Kernels дає можливість писати, запускати та ділитися своїми кодами у Jupyter Notebook, що сприяє обміну знаннями та досвідом.

Платформа також містить навчальні курси та туторіали, які покривають основи машинного навчання, обробку даних та інші важливі теми, що робить її цінним ресурсом для новачків та професіоналів. Завдяки активному співтовариству користувачі можуть ставити питання, обговорювати підходи та отримувати підтримку від колег. Kaggle надає інструменти для оцінки продуктивності моделей за допомогою різних метрик, що допомагає користувачам вдосконалювати свої рішення. Крім того, користувачі можуть завантажувати та ділитися власними наборами даних, що збільшує кількість доступних ресурсів та сприяє розвитку спільноти. Платформа дозволяє використовувати та модифікувати попередньо натреновані моделі, що допомагає прискорити розробку рішень. Kaggle надає можливість відстежувати прогрес своїх моделей у реальному часі та порівнювати результати з іншими учасниками змагань, що стимулює досягнення нових висот у сфері машинного навчання.

Вікно веб-інтерфейсу для набору даних, що використовувався у якості тестових показників розробленого проекту, подано на рисунку 3.1.

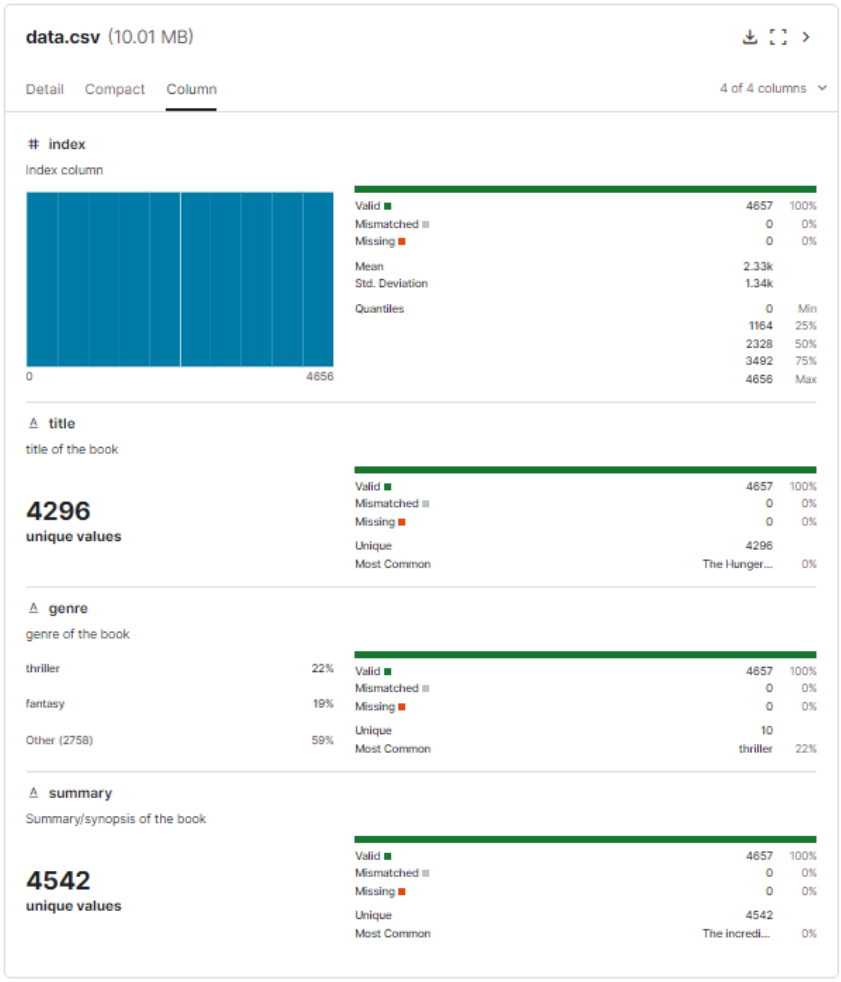

Рисунок 3.1 – Веб-інтерфейс набору даних Book Genre Prediction [20]

Використаний набір має вигляд 4000 записів у 4 колонках: індекс, назва книги, жанри та короткий зміст книги. Для машинного навчання було використано лише два стовпці, які містять жанри книги та їх зміст, тобто, було обрано саме ці дві характеристики. Решта колонок є малоінформативними, але їх було видалено для запобігання перенавчання нейронної мережі.

Під час аналізу документу було виявлено, що не всі поля є заповненими. Тому для проведення обробки даних було видалено рядки із порожніми клітинками. Виведемо перші 20 рядків очищеного набору даних (рис. 3.2).

	genre	summary
0	fantasy	Drowned Wednesday is the first Trustee among ...
1	fantasy	As the book opens, Jason awakens on a school ...
2	fantasy	Cugel is easily persuaded by the merchant Fia...
3	fantasy	The book opens with Herald-Mage Vanyel return...
4	fantasy	Taran and Gurgi have returned to Caer Dallben...
5	fantasy	The novel concerns the dwelling of the Darkov...
6	fantasy	Gen is released from prison by the magus, the...
7	fantasy	The prologue begins with two men who are sear...
8	fantasy	In Luthadel, the capital city of the Final Em...
9	fantasy	A man named Gene finds himself cast into a ne...
10	fantasy	Rhys Mason, former monk of Majere, begins the...
11	fantasy	In the introduction, the current King of the ...
12	fantasy	This book has yet to be published. At this ti...
13	fantasy	The third book sees Alanna through her journe...
14	fantasy	The book is set in a world where colour is th...
15	fantasy	The book weaves around the story of a Kesh wo...
16	fantasy	As the novel begins, a dragon clan receives w...
17	fantasy	The novel and its trilogy use the Moonshae Is...
18	fantasy	The book follows the life of runaway Valerie ...
19	fantasy	Six months after he took control of his own t...

Рисунок 3.2 – Перші 20 записів очищеного набору текстових даних

Такі символи, як зайві пробіли, артиклі тощо можуть вплинути на точність моделі, оскільки однією з важливих ознак для нас буде частота входження слів текст. При аналізі тексту існує ймовірність надмірно підсилити важливість дуже поширених слів, таких, як: a, an, and, are, as, at, be, but, by, for, if, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, this, to, was, will and with і т.д.

Наступним кроком для аналізу текстових даних стало зведення всіх слів до нижнього регістру та проведення операції стемінгу (рис. 3.3). Цей метод має на меті покращити обробку тексту в системах машинного навчання. Для виконання цієї задачі обрано метод SnowballStemmer.

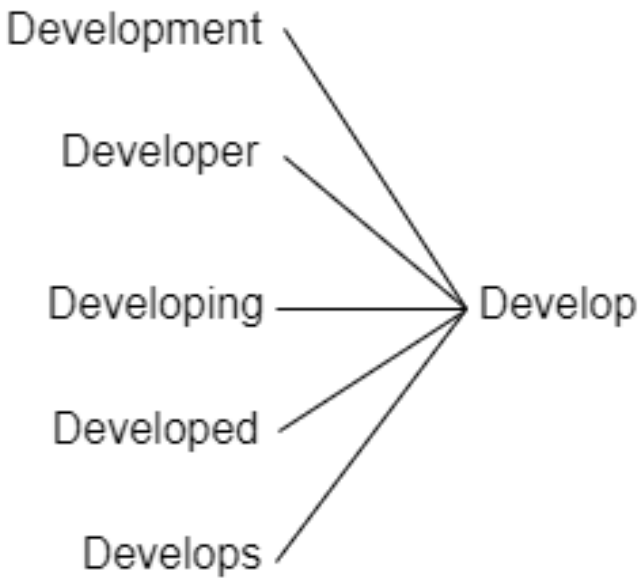

Рисунок 3.3 – Схема проведення стемінгу

На відміну від інших методів для виконання цієї задачі, які використовують вбудовані словники слів, в основі роботи методу SnowballStemmer покладена інформація по конструкціях мови, використання яких забезпечує «зрізання» слів. Через це, у роботі виникають помилки, наприклад видалення частини основи слів. Проте, даний метод не є точним. Результат роботи програми при очищенні представлено на рисунку 3.4.

	crime	fantasy	history	horror	psychology	romance	science	sports	thriller	travel	X
0	0	1	0	0	0	0	0	0	0	0	drown wednesday first truste among morrow day ...
1	0	1	0	0	0	0	0	0	0	0	book open jason awaken school bus unabl rememb...
2	0	1	0	0	0	0	0	0	0	0	cugel easili persuad merchant fianosth attempt...
3	0	1	0	0	0	0	0	0	0	0	book open herald mage vanyel return countri va...
4	0	1	0	0	0	0	0	0	0	0	taran gurgi return caer dallben follow event t...
5	0	1	0	0	0	0	0	0	0	0	novel concern dwell darkovan order renunci als...
6	0	1	0	0	0	0	0	0	0	0	gen releas prison magus king s scholar magus f...
7	0	1	0	0	0	0	0	0	0	0	prologu begin two men search river london thre...
8	0	1	0	0	0	0	0	0	0	0	luthadel capit citi final empir vin scrawni st...
9	0	1	0	0	0	0	0	0	0	0	man name gene find cast new world power godlik...
10	0	1	0	0	0	0	0	0	0	0	rhys mason former monk majer begin novel tri e...
11	0	1	0	0	0	0	0	0	0	0	introduct current king isl valenc iii wizard s...
12	0	1	0	0	0	0	0	0	0	0	book yet publish time expect releas date unknown
13	0	1	0	0	0	0	0	0	0	0	third book see alanna journey bazhir desert ma...
14	0	1	0	0	0	0	0	0	0	0	book set world colour foundat magic sixteen ye...

Рисунок 3.4 – Фрагмент обробки тексту та очищення даних

Оскільки, одним із завдань успішної роботи програми є ще й опція виділення основних слів, важливо не пропустити дійсно суттєві слова, які мають ключове значення для машинного навчання. Тому, треба очистити стовпчик «summary» від зайвих артиклів та пробілів. У результаті текст стає чистішим і більш придатним для подальшого аналізу або застосування алгоритмів машинного навчання. Таким чином, використаний клас забезпечує комплексну підготовку текстових даних, покращуючи якість аналізу.

На рисунку 3.5 представлено згенеровану хмару слів (генерація найбільш вживаних слів у стовпці summary).

Рисунок 3.5 – Хмара слів за частотою згадування у стовпці summary

Усі зазначені вище кроки спрямовані на зменшення розмірності вхідних даних, щоб програма розуміла семантичне значення слів. Різні закінчення в словах одного кореня роблять їх різними для моделі, що ускладнення встановлення зв’язку між ними.

Наступним кроком, стовпчик «Genre» також було очищено від зайвих символів. Частка жанрів у стовпчику були рідкісними, написаними некоректно, помилково або не англійською мовою. Фрагмент коду для побудови діаграми з вибіркою жанрів наведено на рисунку 3.6.

```
# Підрахунок і візуалізація кількості різних жанрів.
plt.bar(y_pos, amount_ofeach_genre, color='dimgrey')  # Використовуємо plt.bar замість plt.barh
# Додавання підписів до стовпців
for index, value in enumerate(amount_ofeach_genre):
    plt.text(index, value, str(value), ha='center', va='bottom', fontweight='bold')  # Вказуємо ha='center' для центрування підписів
plt.xticks(y_pos, top_genres)  # Змінюємо plt.yticks на plt.xticks, оскільки тепер ось x
plt.ylabel('\nЧастота вживання', fontsize=12)  # Змінюємо plt.xlabel на plt.ylabel
plt.xlabel('Жанри\n', fontsize=12)  # Змінюємо plt.ylabel на plt.xlabel
plt.title('\nНайпопулярніші жанри\n', fontsize=14, fontweight='bold')
```

Рисунок 3.6 – Фрагмент коду для побудови діаграми

Для уникнення підвищення складності обчислень та побудови нейронної мережі, було вирішено зменшити розмір вибірки до 10 жанрів. На рисунку 3.7 продемонстровано, як часто вони зустрічаються.

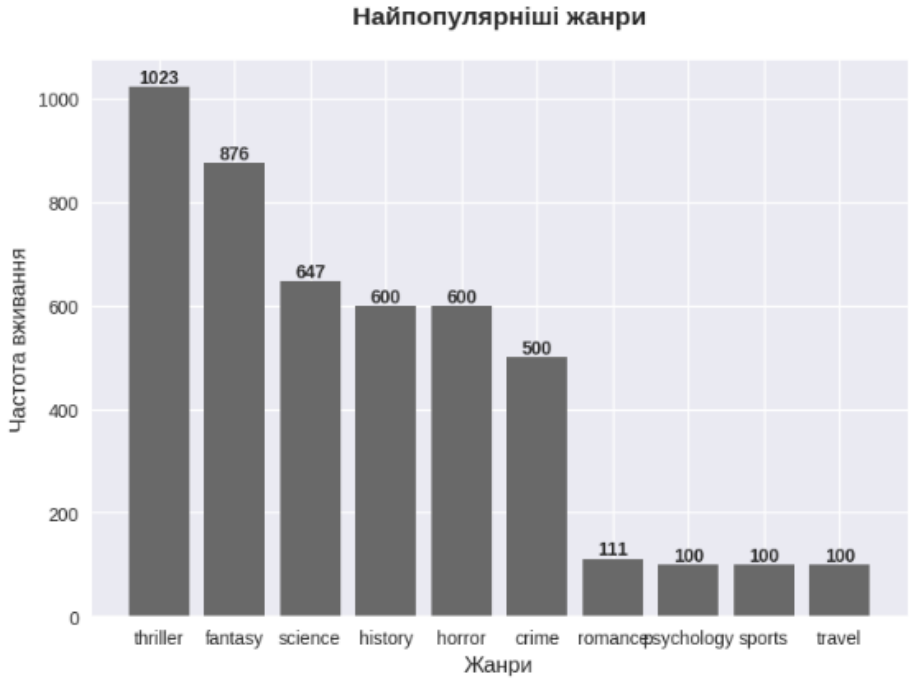

Рисунок 3.7 – Найпопулярніші жанри текстів за частотою вживання

До колонки «Genre» було застосовано метод MultiLabelBinarizer. Цей метод перекодовує мітку на бінарний вектор, кожне значення якого показує, чи належить мітка до певного класу. Наприклад, бінарний вектор книг, які відносяться до жанру «horror», виглядатиме так: (0,0,0,0,1,0,0,0,0)

Алгоритм машинного навчання використовує числові дані, тому для роботи з текстовими даними, ці дані спочатку потрібно перетворити на вектор числових даних за допомогою процесу, відомого як векторизація. Векторизація TF-IDF

передбачає обчислення значення TF-IDF за формулою (3.1) для кожного слова у документі, а потім переміщення цієї інформації у вектор.

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) = \log\left(\frac{N}{nt+1}\right) \times \frac{ni}{\sum_k nk}, \quad (3.1)$$

де ni – кількість входжень терміну t у документ d;

$\sum_k nk$–кількість термінів у документі d;

N - загальна кількість документів у колекції,

nt - кількість документів, що містять термін t.

У таблиці 3.1 представлена схема застосування методу TF-IDF.

Таблиця 3.1 – Демонстрація роботи методу TF-IDF [21]

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log (2/2) = 0	0	0
Car	1/7	0	log (2/1) = 0.3	0.043	0
Truck	0	1/7	log (2/1) = 0.3	0	0.043
Is	1/7	1/7	log (2/2) = 0	0	0
Driven	1/7	1/7	log (2/2) = 0	0	0
On	1/7	1/7	log (2/2) = 0	0	0
The	1/7	1/7	log (2/2) = 0	0	0
Road	1/7	0	log (2/1) = 0.3	0.043	0
Highway	0	1/7	log (2/1) = 0.3	0	0.043

В даному випадку використано два документа – рядок «The car is driven on the road» та «The truck is driven on the highway». Як бачимо з останнього стовпця, найвищу релевантність для «B» мають ті слова, що зустрічаються тільки в документі B. Це слова «Truck» і «Highway» [21].

Таким чином, кожен рядок у наборі даних матиме власний вектор, і вектор матиме оцінку TF-IDF для кожного окремого слова у всій колекції документів.

3.2 Реалізація побудованого алгоритму

Після обробки бази даних було вирішено створити багатошарову нейронну мережу для проведення класифікації тексту. Її програмна реалізація включає шарову архітектуру, яка представлена на рисунку 3.8.

```
46  model.add(Dense(512, activation='relu', input_shape=(X_train.shape[1],)))
47  model.add(Dropout(0.5))
48  model.add(Dense(256, activation='relu'))
49  model.add(Dropout(0.5))
50  model.add(Dense(128, activation='relu'))
51  model.add(Dropout(0.5))
52  model.add(Dense(10, activation='sigmoid'))  # 10 виходів для кожного жанру
```

Рисунок 3.8 – Фрагмент коду з побудовою нейронної мережі

Розроблена мережа буде використовувати для тренування оброблений раніше набір даних.

Пояснення функціоналу кожного шару наступна. Input Layer має 512 нейронів. На цьому рівні використовується функція активації ReLU, яка допомагає уникнути згасання градієнта показників шарів швидше, ніж інші функції активації. Параметр input_shape цього шару визначає форму вхідних даних та встановлює кількість ознак у вхідних даних, які відповідають розміру вектора TF-IDF. Шар Dropout Layer випадковим чином відключає певну кількість нейронів у мережі під час навчання з ймовірністю 50%. Він використовується для регуляризації моделі, щоб уникнути перенавчання і поліпшити загальну здатність. У шарі Hidden Layers додаються два приховані Dense-шари з 256 і 128 нейронами відповідно, кожен з активацією ReLU. Ці шари виконують нелінійні перетворення і дозволяють моделі вчитися складнішим залежностям у вхідних даних. Також для кожного шару додається шар Dropout, використання якого допомагає уникнути перенавчання та покращити загальну здатність моделі. Вихідний шар Output Layer має 10 нейронів, що відповідають 10 можливим класам (жанрам у нашому випадку). Функція активації sigmoid використовується для класифікації і видає ймовірності для кожного класу окремо.

Продуктивність нейронних мереж вимірюється тим, наскільки добре вони можуть передбачати невидимі дані (невидимий набір даних – це той, який не використовувався під час навчання). Проблема узагальнення насправді є однією з головних проблем під час навчання нейронних мереж. Це відоме як тенденція до перевищення навчальних даних, що супроводжується труднощами в прогнозуванні нових даних. Хоча завжди можна точно налаштувати достатньо велику та гнучку нейронну мережу для досягнення ідеальної відповідності (тобто нульової помилки навчання), реальна проблема полягає в тому, як побудувати мережу, яка також здатна передбачати нові дані. Як виявилося, існує зв'язок між переобладнанням навчальних даних і поганим узагальненням. Таким чином, при навчанні нейронних мереж необхідно враховувати питання продуктивності та узагальнення.

Існує кілька методів боротьби з проблемою перенавчання та вирішення проблеми узагальнення. Нами було застосовано метод Early Stopping з пакету keras (рис. 3.9), що дозволяє уникнути перенавчання і зупиняє процес тренування, коли втрати на тестових даних починають зростати.

```
early_stopping = EarlyStopping(monitor='val_loss', patience=1, restore_best_weights=True)
```

Рисунок 3.9 – Фрагмент коду з використанням методу EarlyStopping

Основний параметр методу EarlyStopping – це monitor, який відстежує зміни зазначеної метрики. Було встановлено значення val_loss, цей метод слідкував за динамікою втрат під час тестування нейронної мережі, оскільки втрати на тренувальній вибірці зазвичай менші через ефект перенавчання, що може призвести до переоцінки точності моделі.

Параметр patience був встановлений для обмеження кількості епох, протягом яких відсутні покращення в результатах виконання. Це допомагає уникнути перенавчання моделі, зупиняючи навчання, якщо модель не покращується протягом визначеної кількості епох. Загалом, графік втрат нейронної мережі, поданий на рисунку 3.10, демонструє, як змінюються втрати моделі під час навчання. Спадаючий графік свідчить про те, що з кожною епохою кількість

помилок поступово зменшується, що є ознакою навчання моделі. Зменшення втрат вказує на те, що модель поступово покращує свої передбачення на тренувальних даних. Це важливий показник, який використовується для оцінки ефективності процесу навчання нейронної мережі. Важливо також відслідковувати втрати на валідаційних даних, щоб переконатися, що модель не перенавчається на тренувальних даних.

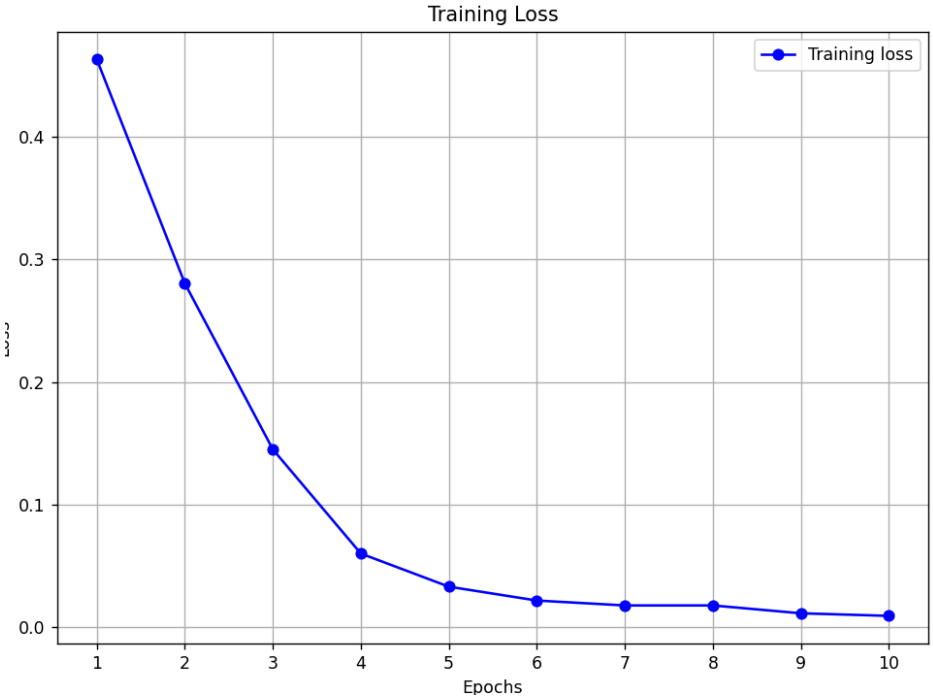

Рисунок 3.10 – Графік функції втрат під час навчання нейронної мережі

Після проведення навчання нейронної мережі, нам потрібно підготувати текст сюжету, який буде вводитися з клавіатури для класифікації. Для цього його необхідно привести до такого ж формату, до якого було приведено нашу базу даних. А саме: видалити стоп-слова, зайві пробіли, розділові знаки, цифри, привести текст до нижнього регістру, а також виконати стемінг слів. Нейронна мережа призначена для роботи лише з векторизованими даними, тому далі цей підготовлений текст потрібно векторизувати. Це означає, що текстовий ввід буде перетворено на числові вектори з використанням таких методів, як TF-IDF або CountVectorizer. Такий підхід забезпечує відповідність формату введених даних

формату, на якому навчалася модель, що є критично важливим для коректної роботи нейронної мережі.

3.3 Тестування роботи та аналіз отриманих результатів

Для тестування нейронної мережі було використано в якості тестових даних сюжет літературного твору, що продемонстрований на рисунку 3.11. Фрагмент сюжету було взято з [22], де розміщені численні резюме, аналізи та обговорення літературних творів. Зокрема, детальний огляд літературного джерела дозволяє нейронній мережі навчатися на даних, які містять багатий лексичний і тематичний матеріал, характерний для фантастики (саме з цього жанру взято джерело).

```
Upon the death of Lord Jon Arryn, the principal advisor to King Robert
Baratheon, Robert recruits his childhood friend Eddard "Ned" Stark, now Warden
of the North, to replace Arryn as Hand of the King, and to betroth his daughter
Sansa to Robert's son Joffrey. Ned accepts the position when he learns that
Arryn's widow Lysa believes he was poisoned by Robert's wife Queen Cersei
Lannister and her family. Shortly thereafter, Ned's son Bran discovers Cersei
having sex with her twin brother Jaime Lannister, who throws Bran from the tower
to conceal their affair, leaving him comatose and paralyzing his legs. Ned
leaves his castle Winterfell and departs for the capital city, King's Landing,
bringing along his daughters Sansa and Arya. Upon arriving in King's Landing to
take his post as Hand, Ned finds that Robert is an ineffective king whose only
interests are hunting, drinking, and womanizing. At Winterfell, an assassin
attempts to kill Bran while he is unconscious, and Ned's wife Catelyn travels to
King's Landing to bring word to Ned. Catelyn's childhood friend, Petyr
"Littlefinger" Baelish, implicates Tyrion Lannister, the dwarf brother of Cersei
and Jaime, in the assassination attempt. On the road back to Winterfell, Catelyn
encounters Tyrion by chance, arrests him, and takes him to the Vale, where her
sister Lysa Arryn is regent, to stand trial for the attempt on Bran's life. In
retaliation for Tyrion's abduction, his father Lord Tywin Lannister sends
soldiers to raid the Riverlands, Catelyn's home region. Tyrion regains his
freedom by recruiting a mercenary named Bronn to defend him in trial by combat.
Ned investigates Jon Arryn's death and eventually discovers that Robert's legal
heirs, including Joffrey, are in fact Cersei's children by Jaime, and that Jon
Arryn was killed to conceal his discovery of their incest. Ned offers Cersei a
chance to flee before he informs Robert, but she uses this chance to arrange
Robert's death in a hunting “accident” and install Joffrey on the throne. Ned
prepares to send his daughters away from King's Landing and enlists
Littlefinger's help to challenge Joffrey's claim; but Littlefinger betrays him,
resulting in Ned's arrest.
```

Рисунок 3.11 – Тестовий набір даних

Кінцевий результат обробки текстових даних представлено у вигляді таблиці, що демонструє результати проведення навчання нейронної мережі (рис. 3.12).

	Genre	Probability
0	crime	0.000000
1	fantasy	99.970001
2	history	0.000000
3	horror	0.000000
4	psychology	0.000000
5	romance	0.000000
6	science	0.000000
7	sports	0.000000
8	thriller	0.000000
9	travel	0.000000

Рисунок 3.12 – Результати тестування нейронної мережі

Тобто, при тестуванні програми виявлено, що жанром досліджуваного тесту є fantasy з відсотком 99,97, що підтверджує валідність і достовірність роботи програми. Виходячи з результатів, можна зробити висновок, що програма правильно провела класифікацію та визначила жанр книги.

Після проведення навчання, користувачу надається ще одна можливість провести класифікацію наступного відгуку. Для виходу із процесу, необхідно натиснути «q». Таким чином, сформований функціонал обробки текстових даних засобами машинного навчання.

ВИСНОВКИ

Дослідження текстових даних за допомогою нейронних мереж є важливим завданням в сучасних науці та технологіях. В роботі показано, що нейронні мережі здатні ефективно аналізувати та розуміти великі обсяги текстової інформації, виявляючи складні патерни та залежності. Такого роду дослідження сприяють удосконаленню систем автоматичного перекладу, категоризації контенту, аналізу відгуків користувачів та інших аспектів обробки природної мови. Застосування нейронних мереж у текстовому аналізі сприяє розвитку інтелектуальних систем, які вміють розпізнавати семантичні зв'язки і вирішувати складні завдання, пов'язані з розумінням мовного контексту та вираженням відтінків значень слів.

У результаті виконання кваліфікаційної роботи було виконано наступні завдання:

– проведено аналітичний огляд з питань систем обробки текстових даних та досліджено, що мова програмування Pyton є ефективною для обробки даних, у порівнянні з іншими мовами, такими як Java та R, а використання Pandas спільно з іншими бібліотеками дозволяє створювати потужні та масштабовані рішення для аналізу текстових даних;

– проведено аналіз моделей нейронних мереж та виявлено, що загальний мета-аналіз текстових даних включає такі дії, як крос-валідація для оцінки стабільності моделі;

– досліджено інструменти програмного середовища для роботи з нейронними мережами та визначено, що для реалізації машинного навчання використовуються бібліотеки TensorFlow, Scikit-learn, Keras, Matplotlib та Numpy, що володіють алгоритмами класифікацій та володіють підтримка різних типів задач машинного навчання, таких як класифікація, регресія, кластеризація та зниження розмірності;

– поведено аналіз та підготовку набору даних та визначено середовище для реалізації тестових експериментів – веб-платформу Kaggle, яка надає численні можливості для навчання та роботи з базами різних типів;

– побудовано семи-шарову нейронну мережу для аналізу даних, досліджено доцільність кожного шару мережі;

– протестовано роботу програми на різних наборах даних, достовірність даних за вибіркою складає 99,97%.

Тобто, усі поставлені завдання виконано. Мета роботи досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Машинне навчання: що це таке, як працює і для чого використовується. *GoIt Global*. URL: http://surl.li/urnjw (дата звернення: 02.03.2024).

2. Руда О., Будко Г., Висоцький А. Машинне навчання для прогнозування й ранньої діагностики серцево-судинних захворювань: методи й перспективи. Перспективи та виклики теоретичної медицини. 2024. № 2. С. 29–37. URL: https://doi.org/10.57125/pmed.2024.01.29.04 (дата звернення: 03.03.2024).

3. Чіома Е. В., Січко Т.В. Машинне навчання в медицині з використанням Power BI EMBEDDED. Прикладні аспекти сучасних міждисциплінарних досліджень: матеріали І всеукр. наук.-практ. конф., м. Вінниця, 2021. С. 124-127.

4. Здір В. А., Ткаченко А. А., Бразілій Н. М. Роль цифрових технологій для ефективного фінансово-облікового управління суб'єктів господарювання. *Проблеми сучасних трансформацій. Серія "Економіка та управління"*. 2024. URL: https://doi.org/10.54929/2786-5738-2024-11-09-02 (дата звернення: 02.03.2024).

5. Ковпак Е., Фролов Ф. Порівняльний аналіз моделей машинного навчання і регресій для прогнозування ціни легкового авто. *Вісник харківського національного університету імені В. Н. Каразіна.* Харків. Т. 9. С. 31–40. URL: https://doi.org/10.26565/2311-2379-2019-97-04 (дата звернення: 05.03.2024).

6. Кононова К. Машинне навчання: методи та моделі. *Харківський національний університет імені В. Н. Каразіна.* 2020. С. 12–21. URL: http://surl.li/uroae (дата звернення: 05.03.2024).

7. Угрюмов М., Черниш С. Методи машинного навчання у задачах системного аналізу і прийняття рішень. Харків : Харків. нац. ун-т ім. В.Н. Каразіна, 2019. 195 с. URL: http://surl.li/urobg (дата звернення: 13.02.2024).

8. Pandas. Version 2.2.2. URL: https://pandas.pydata.org (дата звернення: 12.04.2024).

9. Матвєєв М., Татарников А. Особливості використання машинного навчання в автономних транспортних засобах як методу запобігання

аварій. *Матеріали конференції МНЛ*, м. Вінниця, 3 листопада 2023 р. с. 169–170. URL: http://surl.li/uroij 05.04.2024 (дата звернення: 07.04.2024).

10. Могильна М., Дубровін В. Інтелектуальний аналіз тексту: застосування та безкоштовні програмні засоби. *Прикладні питання математичного моделювання*. 2022. № 5. С. 41–49. URL: http://surl.li/uroka (дата звернення: 08.04.2024).

11. Матвієнко С. Принципи побудови нейронних мереж. *IT-Master*. URL: http://surl.li/urovk (дата звернення: 13.04.2024).

12. Мельник К., Лавренчук С., Христинець Н. Виявлення шахрайства з кредитними картками методами машинного навчання. *Herald of Khmelnytskyi National University. Technical sciences.* 2024. №333(2). с. 189-193. URL: https://doi.org/10.31891/2307-5732-2024-333-2-30 (дата звернення: 13.04.2024)

13. Ткаченко О., Олійник О. Можливості та труднощі використання обробки природної мови. *Практичні та теоретичні питання розвитку науки та освіти (частина І) : Матеріали ІІ Міжнар. науково-практ. конф*., м. Львів, 19–20 груд. 2020 р. URL: http://surl.li/urpka (дата звернення: 05.05.2024).

14. Бази даних NLTK Corpora. URL: https://www.nltk.org/nltk_data/ (дата звернення: 05.05.2024).

15. NLTK URL: https://www.nltk.org/ (дата звернення: 05.05.2024).

16. Ajitesh K. XGBoost Classifier Explained with Python Example. Analytics Yogi. URL: https://vitalflux.com/xgboost-classifier-explained-with-python-example/ (дата звернення: 22.06.2024).

17. TensorFlow. URL: https://www.tensorflow.org/ (дата звернення: 06.05.2024).

18. KerasNLP. URL: https://keras.io/api/keras_nlp/ (дата звернення: 06.05.2024).

19. Donald A. Traffic Sign Detection and Recognition Using Python. 2023. Vol. 9, no. 3. P. 2904–2905. URL: http://surl.li/urqmx (дата звернення: 02.05.2024).

20. Book Genre Prediction. Kaggle. URL: http://surl.li/urqpl (дата звернення: 06.05.2024).

21. Simha A. Understanding TF-IDF for Machine Learning. Capitalone. URL: http://surl.li/ugjsl (дата звернення: 10.05.2024).

22. A Game of Thrones. gradesaver.com. URL: http://surl.li/urrql (дата звернення: 10.05.2024).