

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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Індивідуальний дослідницький проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Застосунок для виявлення фейкових новин за допомогою машинного
навчання»**

Виконав:

студент IV курсу, групи ІА-82

Зевальов Артур Андрійович _____

Керівник:

Професор кафедри ІСТ, д.т.н, проф.,

Онищенко Вікторія Валеріївна _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАВДАННЯ

на індивідуальний дослідницький проєкт студенту

Зевальову Артуру Андрійовичу

1. Тема проєкту «Застосунок для виявлення фейкових новин за допомогою машинного навчання», керівник проєкту Оніщенко Вікторія Валеріївна д.т.н, проф.
2. Термін подання студентом проєкту 15-червня 2022 року
3. Вихідні дані до проєкту код на мові програмування Python
4. Зміст пояснювальної записки аналіз предметної області, огляд існуючих рішень, вибір технологій та особливості реалізації, розробка ПЗ, керівництво користувача, тестування
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) діаграми варіантів використання, класів, розгортання та активності
6. Дата видачі завдання 1 грудня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Аналіз предметної області	25.04.2022	Виконано
2.	Огляд існуючих рішень	02.05.2022	Виконано
3.	Вибір технології та особливості реалізації	07.05.2022	Виконано
4.	Розробка застосунку	15.05.2022	Виконано
5.	Навчання нейронної мережі	20.05.2022	Виконано
6.	Тестування	23.05.2022	Виконано
7.	Виконання графічних документів	28.05.2022	Виконано
8.	Оформлення пояснювальної записки	31.05.2022	Виконано
9.	Подання ІДП на основний захист	15.06.2022	Виконано

Студент

Артур ЗЕВАЛЬОВ

Керівник проекту

Вікторія ОНИЩЕНКО

АНОТАЦІЯ

Зевальов А.А. Застосунок для виявлення фейкових новин за допомогою машинного навчання. КПІ ім. Ігоря Сікорського, Київ, 2022.

Проект містить 62 с. тексту, 18 рисунків, 6 таблиці, посилання на 41 літературні джерела, додатки та 4 конструкторських документів.

Ключові слова: фейкові новини, машинне навчання, класифікація тексту, графічний застосунок, Python.

Об'єктом розробки є програма для виявлення фейкових новин.

Мета розробки – дати можливість людям самостійно підтверджувати або спростовувати правдивість новин.

В результаті виконання індивідуального дослідницького проекту поставлена мета була досягнута, застосунок може використовуватись людьми з різними рівнями підготовки для аналізу новин. Додаток має простий інтерфейс, що здатен трансформуватися та надавати більше можливостей за потреби.

Отримані результати можуть бути корисними при створенні подібного програмного забезпечення чи покращенні існуючого застосунку.

SUMMARY

Zevalov A. A. An application for detecting fake news using machine learning. Igor Sikorsky, KPI, Kyiv, 2022.

The project contains 62 pages of text, 18 figures, 6 tables, references to 41 literature sources, appendices and 4 design documents.

Keywords: fake news, machine learning, text classification, graphical application, Python.

The object of development is a program for detecting fake news.

The purpose of the development is to enable people to confirm or refute the veracity of the news.

As a result of the implementation of an individual research project, the goal was achieved, the application can be used by people with different levels of training for news analysis. The application has a simple interface that can be transformed and provide more features as needed.

The results can be useful in creating similar software or improving the existing application.

Номер рядка	Формат	Позначення	Найменування	Кільк. листів	Номер екзем.	Приміт.
1			<u>Документація загальна</u>			
2						
3			Заново розроблена			
4						
5	A4	IA82.080БАК.003 ПЗ	Пояснювальна записка	62		
6						
7	A3	IA82.080БАК.003 Д1	Застосунок для виявлення	1		
8			фейкових новин за допомогою			
9			машинного навчання			
10			Діаграма класів			
11						
12	A3	IA82.080БАК.003 Д2	Застосунок для виявлення	1		
13			фейкових новин за допомогою			
14			машинного навчання			
15			Діаграма розгортання			
16						
17	A3	IA82.080БАК.003 Д3	Застосунок для виявлення	1		
18			фейкових новин за допомогою			
19			машинного навчання			
20			Діаграма варіантів використання			
21						
22	A3	IA82.080БАК.003 Д4	Застосунок для виявлення	1		
23			фейкових новин за допомогою			
24			машинного навчання			
25			Діаграма активності			
26						
27						
28						

					IA82.080БАК.003 ТП			
Зм.	Лист	№ докум.	Підпис	Дата				
Розроб.	Зевальов				Застосунок для виявлення фейкових новин за допомогою машинного навчання Відомість технічного проекту	Літ.	Лист	Листів
Керівн.	Онищенко							1
Реценз.						НТУУ «КПІ» ФІОТ Група IA-82		
Н. Контр.								
Затв.								

Пояснювальна записка
до індивідуального дослідницького проєкту
на тему: «Застосунок для виявлення фейкових новин за
допомогою машинного навчання»

Київ – 2022 року

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	5
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Ввід поняття фейкових новин	9
1.1.1 Визначення поняття фейкових новин	9
1.1.2 Загроза фейкових новин	10
1.1.3 Шляхи вирішення проблеми фейкових новин	11
1.2 Підходи до задачі розпізнавання фейкових новин	11
1.2.1 Вибір правильного інструменту	12
1.3 Опис алгоритмів класифікації.....	13
1.3.1 Алгоритм лінійної моделі зі стохастичним градієнтним спуском.....	13
1.3.2 Методи апроксимації ядра	17
1.3.3 Методи лінійних опорних векторів	17
1.3.4 Метод наївного Байеса.....	19
1.3.5 Метод найближчих сусідів.....	22
1.3.6 Ансамблеві методи.....	22
Висновки до розділу 1	23
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	24
2.1 False, Misleading, Clickbait-y, and/or Satirical “News” Sources	24

					IA82.080BAK.003 ПЗ			
		№ докум.	Підпис		Застосунок для виявлення фейкових новин за допомогою машинного навчання. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Зевальов А. А.			Т		2	66	
Перевірив	Онищенко В. В.							
Затв.								
						КПІ ім. Ігоря Сікорського ФІОТ група IA-82		

2.2 Fact-checking on Facebook.....	25
2.3 Fake news debunker by InVID & WeVerify	25
Висновки до розділу 2	26
3 ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ.....	27
3.1 Вибір нейронної мережі(алгоритму).....	27
3.2 Засоби розробки	27
3.2.1 Мова програмування Python	28
3.2.2 Середовище розробки Visual Studio Code	28
3.2.3 Графічний інтерфейс Tcl/Tk за допомогою Tkinter	29
3.2.4 Інструментарій дистрибуції коду – PyInstaller.....	30
3.2.5 Бібліотеки та модулі Python	31
Висновки до розділу 3	32
4 РОЗРОБКА ПЗ.....	34
4.1 Опис внутрішньої роботи	35
4.2 Розробка діаграми класів.....	37
4.3 Розробка діаграми розгортання	38
4.4 Розробка діаграми варіантів використання	39
4.5 Розробка діаграми активності.....	40
4.6 Опис створених класів та їх функціоналу	41
Висновки до розділу 4	45
5 КЕРІВНИЦТВО КОРИСТУВАЧА.....	46
Висновки до розділу 5	51

6 ТЕСТУВАННЯ	52
6.1 Мета тестування	52
6.2 Тестування інтерфейсу	52
6.3 Тестування точності моделі	55
Висновки до розділу 6	55
ВИСНОВКИ.....	57
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТОК А.....	63

					ІА82.080БАК.003 ПЗ	Аркуш
						4
Зм.	Арку	№ докум.	Підпис	Дата		

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Клікбейт (англ. Clickbait) – заголовок, який змушує користувача інтернету натиснути на нього; зневажливий термін, що описує вебконтент, метою якого є отримання доходу від онлайн-реклами, особливо на шкоду якості або точності інформації.

Обробка природної мови (англ. Natural-language processing, NLP) — загальний напрям інформатики, штучного інтелекту та математичної лінгвістики. Він вивчає проблеми комп'ютерного аналізу та синтезу природної мови.

					ІА82.080БАК.003 ПЗ	Аркуш
						5
Зм.	Арку	№ докум.	Підпис	Дата		

ВСТУП

Фейкові новини почали своє існування з тих же часів, що і звичайні новини. Протягом сотень років, вони існували пліч-о-пліч, але у цей самий час також була своєрідна система стримувань і переваг. Розповсюдження фейкових новин могло швидко призвести до втрачання соціальної довіри до людини чи організації, яка цим займалася. Саме цей соціальний тиск стримував розповсюдження фейкових новин, адже відбудувати свій соціальний статус було б досить важко.

Сьогодні технології, що дозволяють залишатися в інтернеті анонімним, дали нове життя фейковим новинам. Відтепер будь-хто здатен створити та розповсюдити таку інформацію з метою заробітку грошей, дестабілізації ситуації та інше. Це підтверджується тисячами скарг у соціальних мережах, що переповнилися фейковими новинами. Рівень довіри до достовірних джерел новин також похитнувся, адже людина більше не може бути впевненою, чи вона читає реальні новини.

Компанії, волонтерські організації, індивідууми та навіть уряди – усі спрямовують ресурси на подолання тотальної епідемії фейкових новин, але результати говорять самі за себе, адже використані дії не дають бажаного результату.

Країни, що приймають закони проти розповсюдження фейкових новин, погрожуючи штрафами та навіть ув'язненням стикаються з проблемою неможливості знайти відповідального за створення самої фейкової новини, а тільки можуть дістатися до пересічних людей, що повірили у цю новину, та розповсюдили її не бажаючи зла. Це призводить до того, що від таких законів страждають ті, кого вони мали б захищати, що доводить те, що держава не має можливості подолати фейкові новини простою заборною.

Деякі компанії наймають великі штати так званих «перевірячів фактів», що продивляються кожну новину опубліковану у соціальній мережі, та оцінюють її. Цей

					IA82.080BAK.003 ПЗ	Аркуш
						6
Зм.	Арку	№ докум.	Підпис	Дата		

підхід є незадовільним через великі витрати на підтримку такої кількості спеціалістів та не захищає від упередженості, або персональних, політичних переконань особи, що може призвести до класифікації реальних, але неподобаючихся, новин як фейк.

Волонтери організуються, щоб створити бази даних сайтів з фейковими новинами, що в теорії можу допомогти відбракувати такі новини. Але цей підхід стикається з проблемою відносною легкості створення сотень таких сайтів, тому такі бази даних завжди будуть застарілими на момент публікації.

Дивлячись на цю проблему, було прийнято рішення розробити застосунок, який би міг надійно, швидко та легко оцінювати новини на предмет фейковості. Цей застосунок мав би бути легким в використанні, але з додатковими функціями для користувачів з більшими потребами. Ця програма мала б бути кросплатформеною та працювати повністю офлайн, що б дозволило її використання якнайбільшої кількості людей. Також, однією з важливих характеристик програми мала б бути легкість додавання нових способів перевірки новин.

Отриманий на виході застосунок з відкритим кодом має бути доступним для кожного, що дозволить поступово вирішити, або суттєво послабити проблему фейкових новин та стати базою для подальшої розробки і покращення програми.

Створений в результаті виконання цього індивідуального дослідницького проекту застосунок має відповідати усім поставленим завданням. Окремо від своєї основної функції – виявлення фейкових новин, надання користувачам вибору типу інтерфейсу має дозволити розробити одночасно простий у використанні, але з великою кількістю додаткових функцій застосунок. Використання об'єктно орієнтованого програмування у поєднанні зі створеними базовими класами для різних типів класифікаторів, має дозволити додання нового функціоналу за лічені хвилини, що дасть можливість масштабувати та доробляти застосунок для інших розробників також.

					IA82.080BAK.003 ПЗ	Аркуш
						7
Зм.	Арку	№ докум.	Підпис	Дата		

У ході розробки мають бути використані найновітніші технології розробки програмного забезпечення. Таким чином, застосунок має використовувати многопоточність, що підвищить його швидкість та надійність роботи, надавати можливість зберігати налаштування користувача та пропонувати легкий в опануванні функціонал для створення, навчання, модифікації та експортування моделей класифікаторів.

Індивідуальний дослідницький проєкт складається з таких розділів: вступ, аналіз предметної області, огляд існуючих рішень, вибір технологій та особливості реалізації, розробка ПЗ, керівництво користувача, тестування, висновки, перелік використаних джерел із 41 найменувань, графічна частина складається з 4 креслеників. Загальний обсяг 66 сторінок.

					IA82.080BAK.003 ПЗ	Аркуш
						8
Зм.	Арку	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Ввід поняття фейкових новин

Багато пересічних громадян вважають, що знають, що таке фейкові новини та як їх відрізнити від рекламних або гумористичних статей, упередженої або відверто поганої журналістики, та новин, з якими людина не погоджується в силу своїх особистих переконань. Але як показує практика, це зовсім не відповідає дійсності, адже більшість людей, на всіх рівнях освіти, навіть отримавши безпосереднє завдання – знайти рекламу у виборці із статей, не в змозі цього зробити, що каже про низький рівень інформаційної грамотності [1].

1.1.1 Визначення поняття фейкових новин

У чистому вигляді фейкові новини – це повністю вигадані статті та/або інші об'єкти передачі інформації, створені з метою маніпулювання та нагадування надійної журналістики, щоб привернути максимальну увагу, а разом з цим і прибуток від реклами. Часто такі новини розміщуються на веб-сайтах, які дотримуються конвенцій щодо дизайну відомих онлайн-медіа новин та із звичайними назвами, що надають видимість легітимності. Ці історії орієнтовані на публікування в соціальних мережах, що набагато поживляє їхнє проникнення у медіа-простір [2].

З переходами за посиланням приходять прибуток, наприклад людина, яка керує серією фейкових новинних сайтів може заробляти приблизно 30 000 доларів США на місяць від однієї тільки реклами, яка винагороджує високий трафік, що у випадку таких новин є майже гарантованим [3].

					ІА82.080БАК.003 ПЗ	Аркуш
						9
Зм.	Арку	№ докум.	Підпис	Дата		

Фейкові новини можуть використовуватися не тільки задля заробітку, а також для повідомлення людям неприємної або шокуючої особистої інформації про когось з метою дискредитувати політичного опонента та дестабілізувати ситуацію у громаді або країні [4].

Існує багато сайтів, контент яких може виглядати як фейкові новини, наприклад існують:

— веб-сайти, які поширюють викривлену, деконтекстуалізовану чи сумнівну інформацію, наприклад, клікбейтні заголовки; не відображають факти історії, або мають неоголошену упередженість;

— гумористичні сайти, особливо якщо вони торкаються поточних подій чи політики, а також якщо вони з’являються без контексту в соціальних мережах;

— сайти, які спеціалізуються на реальних історіях, які здаються такими іронічними або смішними, що на перший погляд видаються фальшивими;

— колонка, огляд чи звіт новин, з якими переглядач не згоден.

Усі наведені у переліку вище типи сайтів не прийнято вважати фейковими новинами [2].

1.1.2 Загроза фейкових новин

Фейкові новини здатні створити реальну загрозу фізичному і психологічному здоров’ю людей, та навіть життю. Особливо ця загроза стає помітною, коли відбувається концентрація таких історій направлена на одну тематику. Це призводить до появи конспіраційних теорій, що підтримують ненависть та недовіру у суспільстві. Люди, вмотивовані конспіраційними теоріями, можуть стати небезпекою для себе та оточуючих [5-6].

					ІА82.080БАК.003 ПЗ	Аркуш
						10
Зм.	Арку	№ докум.	Підпис	Дата		

Загроза фейкових новин може бути більш витонченою. Наприклад, країна-агресор має можливість застосувати фейкові новини задля втручання у демократичні процеси та дестабілізації ситуації у країні-жертві; інші зацікавлені групи також можуть таким чином оказувати вплив зсередини [4].

1.1.3 Шляхи вирішення проблеми фейкових новин

Існує декілька дієвих способів щодо вирішення цієї проблеми. Але кожен з них має якісь недоліки, як розглянуто у розділі 2.

Можливо ввести цензурування новин у соціальних мережах, або найняти спеціалістів, що будуть слідкувати за відповідністю фактам.

Також існує можливість перекрити дохід сайтам, що розповсюджують фейкові новини, тим самим вбиваючи мотивацію їх створювати та підтримувати.

Громада може створити список сайтів, які є відомими розповсюджувачами несправжньої інформації.

Країна має можливість запустити кампанію щодо критичного мислення або почати викладати предмети такої тематики у школі, таким чином люди почнуть перевіряти інформацію перед тим як поширювати її або приймати на віру.

1.2 Підходи до задачі розпізнавання фейкових новин

Так, як фейкові новини – це здебільшого текст, засоби розпізнавання та класифікації тексту можуть бути застосовані для завдання виявлення таких новин.

Обробка природної мови (NLP) є однією з найскладніших областей машинного навчання, оскільки вона вимагає вивчення людського спілкування і, якимось, людської свідомості. Більше того, на відміну від того, як відносно легко можна закодувати

					IA82.080BAK.003 ПЗ	Аркуш
						11
Зм.	Арку	№ докум.	Підпис	Дата		

зображення, з точки зору даних (тобто двовимірної матриці) або фізичного експерименту (це в основному файл .csv), дуже складно закодувати текст у вигляді числа або вектора [7].

1.2.1 Вибір правильного інструменту

Часто найскладнішою частиною вирішення проблеми машинного навчання може бути пошук правильного інструменту оцінки для роботи. Різні оцінювачі краще підходять для різних типів даних і різних проблем. На рисунку 1 наведено блок-схему, яка допомагає обрати потрібний інструмент [8].

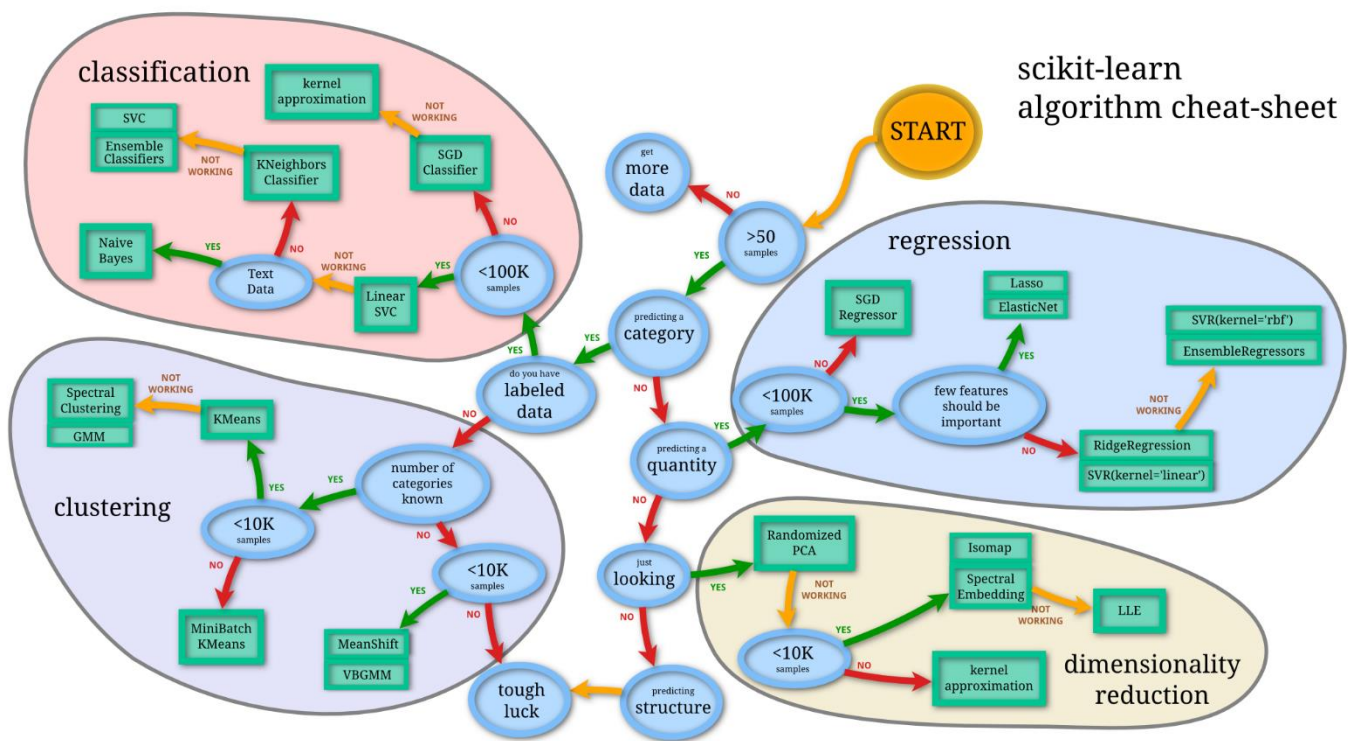


Рисунок 1 – Блок-схема шляху вибору алгоритму[8]

					IA82.080BAK.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		12

Простежимо шлях вибору алгоритму:

Задача – виявлення фейкових новин, отож маємо мати вибірку з більше ніж 50 зразків, що у нашому випадку є досить легким в отриманні, адже існує велика кількість потрібних вибірок у відкритому доступі.

При виявленні фейкових новин у нас зазвичай є дві категорії позначень: «Справжні» чи «Фейкові» новини. Також, зазвичай вибірки даних теж несуть відповідні лейбли до кожного зразка, задля розрізнення категорій.

Таким чином бачимо, що для нашого завдання більш всього підходять алгоритми типу класифікації.

Далі, залежно від розміру вибірки має обрати конкретний алгоритм.

При розмірі вибірки більше ніж 100,000 зразків, маємо обрати алгоритм лінійної моделі зі стохастичним градієнтним спуском, у випадку, коли цей метод не дає потрібних результатів – використаємо метод апроксимації ядра.

При розмірі вибірки менше ніж 100,000 зразків, слід обрати алгоритм класифікації методом лінійних опорних векторів, а у випадку, коли він не працює – маємо, залежно від типу даних вибірки, використати або метод наївного Байєса, коли працюємо з текстовими даними, або метод найближчих сусідів при роботі з, наприклад, зображеннями. У випадку, коли метод найближчих сусідів нас не задовольняє, можемо використати або метод С-підтримки векторної класифікації, або ансамблеві методи.

1.3 Опис алгоритмів класифікації

1.3.1 Алгоритм лінійної моделі зі стохастичним градієнтним спуском

Історія градієнтного спуску та алгоритмів оптимізації почалася у Франції 19 століття.

					IA82.080BAK.003 ПЗ	Аркуш
						13
Зм.	Арку	№ докум.	Підпис	Дата		

Французький математик, інженер і фізик Огюстен Коші був піонером у галузі комплексного аналізу та абстрактної алгебри. І його вплив назавжди відображено в математиці та фізиці, з кількома теоремами та концепціями, названими на його честь, як-от Розподіл Коші, Теорема Коші або Пружні матеріали Коші.

Серед багатьох ключових внесків Коші також відомим є те, що він розробив алгоритм градієнтного спуску.

Мотивація Коші полягала в тому, що він хотів обчислити орбіту небесного тіла без розв'язування диференціальних рівнянь, а замість цього вирішив алгебраїчні рівняння, що представляють рух цього тіла [9].

Ідея алгоритму полягає в тому, щоб знайти ефективний спосіб досягнення мінімального значення функції.

Градієнт функції — це сукупність усіх її часткових похідних, організованих у вектор [10], і представлений перевернутим трикутником, який називається набла. У моделі машинного навчання градієнт можна розглядати як сукупність часткових похідних, кожна з яких стосується однієї з ознак моделі, що показано на рисунку 2.

$$\nabla f(x_0, x_1, x_2, \dots, x_n) = \begin{bmatrix} \frac{\partial f}{\partial x_0}(x_0, x_1, x_2, \dots, x_n) \\ \frac{\partial f}{\partial x_1}(x_0, x_1, x_2, \dots, x_n) \\ \frac{\partial f}{\partial x_2}(x_0, x_1, x_2, \dots, x_n) \\ \dots \\ \frac{\partial f}{\partial x_n}(x_0, x_1, x_2, \dots, x_n) \end{bmatrix}$$

Градієнт функції F

похідна від f до x_0
похідна від f до x_1
похідна від f до x_2
похідна від f до x_n

Рисунок 2 – Градієнт функції

					IA82.080BAK.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		14

Градiєнт функції завжди буде вказувати в напрямку, де є найбільше збільшення функції.

Алгоритм градієнтного спуску виконується за такими кроками:

а) вибираємо випадкову точку у функції, що стає початковою точкою;

б) поки градієнт сходиться:

а. обчислюємо від’ємний градієнт у початковій точці, градієнт рухається в протилежному напрямку;

б. оновлюємо початкову точку результатом кроку а. і повертаємося до кроку а.

в) успішно знаходимо мінімум [11].

Вищенаведений алгоритм можна побачити на рисунку 3.

$$\underbrace{W^{(t+1)}}_{\text{позиція наступної ітерації}} = \underbrace{W^{(t)}}_{\text{позиція попереднього кроку}} - \underbrace{\alpha}_{\substack{\text{швидкість} \\ \text{навчання}}} \underbrace{\nabla f(W^{(t)})}_{\text{крок}}$$

Рисунок 3 – Формула градієнтного спуску

З алгоритму можна побачити основні недоліки цього методу.

					IA82.080BAK.003 ПЗ	Аркуш
						15
Зм.	Арку	№ докум.	Підпис	Дата		

Навіть якщо градієнтний спуск — це потужний алгоритм, в реальних сценаріях із постійно зростаючими вибірками даних він має два основних обмеження:

- розрахунок похідних для всього набору даних займає багато часу;
- необхідна пам'ять пропорційна розміру набору даних.

Зі збільшенням набору даних обчислення стають набагато повільнішими, тому час до збіжності буде збільшуватися.

Ці обчислення зазвичай виконуються в пам'яті, тому чим більше набір даних, тим більші вимоги до пам'яті [12].

Ось тут на допомогу приходить стохастичний градієнтний спуск.

З огляду на обмеження градієнтного спуску, стохастичний градієнтний спуск з'явився як спосіб вирішення проблем продуктивності та прискорення збіжності у великих наборах даних.

Стохастичний градієнтний спуск є ймовірнісним наближенням градієнтного спуску. Це наближення, оскільки на кожному кроці алгоритм обчислює градієнт для одного спостереження, вибраного випадковим чином, замість того, щоб обчислювати градієнт для всього набору даних. Деталі можна побачити на рисунку 4.

$$\begin{array}{c}
 \begin{array}{ccc}
 & \text{швидкість} & \text{спостереження } i \\
 & \text{навчання} & \\
 & \swarrow & \searrow \\
 & \alpha & \nabla f_i(w^{(t)}) \\
 & \swarrow & \searrow \\
 \text{позиція} & & \text{крок} \\
 \text{наступної} & & \\
 \text{ітерації} & & \\
 & \downarrow & \\
 & \text{позиція} & \\
 & \text{попереднього} & \\
 & \text{кроку} &
 \end{array} \\
 \hline
 w^{(t+1)} = w - \alpha \nabla f_i(w^{(t)})
 \end{array}$$

Рисунок 4 – Формула стохастичного градієнтного спуску

					IA82.080BAK.003 ПЗ	Аркуш
						16
Зм.	Арку	№ докум.	Підпис	Дата		

Порівняно з градієнтним спуском, стохастичний градієнтний спуск є набагато швидшим і більше підходить для масштабних наборів даних.

1.3.2 Методи апроксимації ядра

Існує декілька способів апроксимації ядра. Найбільш відомі: метод Ністррома для апроксимації ядра, метод ядра радіальної базисної функції, метод адитивного ядра χ^2 -квадрат, асиметричний метод ядра χ^2 -квадрат та метод апроксимації поліноміального ядра за допомогою тензорного ескізу [13].

Метод Ністррома є загальним методом для низькорангових апроксимацій ядер. Це досягається, по суті, шляхом підвибірки даних, на яких оцінюється ядро. Метод Ністрьрома використовує залежні від даних базисні функції, що призводить до покращеного аналізу підходів до навчання ядра [14].

Метод ядра радіальної базисної функції є популярною функцією ядра у машинному навчанні, яка використовується в різних алгоритмах ядрового навчання. Зокрема, цей метод зазвичай використовується в класифікації за допомогою опорних векторів. Цей метод особливо популярний через його схожість з алгоритмом найближчих сусідів, але він має перевагу у долатті проблеми невистачення простору, оскільки методи ядра радіальної базисної функції просто повинні зберігати опорні вектори під час навчання, а не весь набір даних. [15]

1.3.3 Методи лінійних опорних векторів

Цей метод є одним із найпопулярніших алгоритмів навчання з вчителем, який використовується для задач класифікації та регресії. Його основною метою є створення найкращої лінії або межі рішення, яка може розділити n -вимірний простір

					IA82.080BAK.003 ПЗ	Аркуш
						17
Зм.	Арку	№ докум.	Підпис	Дата		

на класи, щоб ми могли легко помістити нову точку даних у правильну категорію в майбутньому. Ця межа найкращого рішення називається гіперплощиною. Метод лінійних опорних векторів вибирає крайні точки/вектори, які допомагають створити гіперплощину. Ці екстремальні випадки називаються опорними векторами. На рисунку 5 наведено діаграму, яка пояснює це на практиці [16].

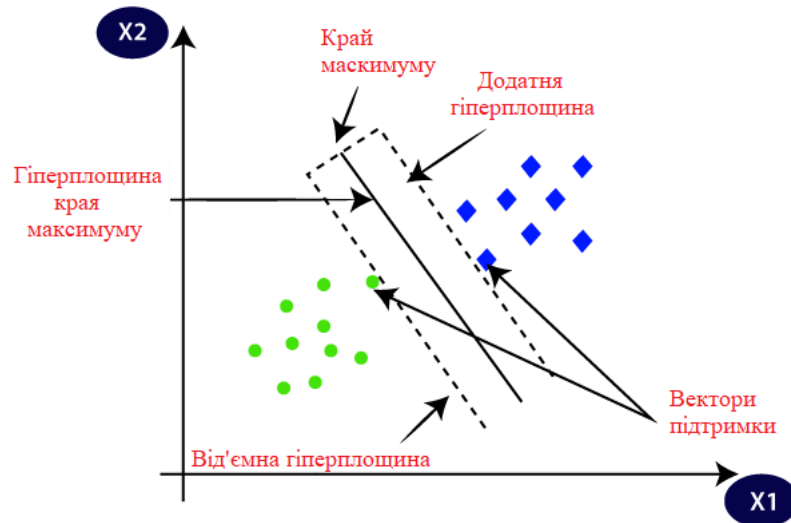


Рисунок 5 – Діаграма методу лінійних опорних векторів

Пояснення до рисунку 5:

Гіперплощина: може бути кілька ліній/меж рішень для відокремлення класів у n -вимірному просторі, але нам потрібно знайти найкращу межу рішення, яка допомагає класифікувати точки даних. Ця найкраща межа відома як гіперплощина. Розміри гіперплощини залежать від функцій, присутніх у наборі даних, що означає, що якщо є 2 об'єкти (як показано на зображенні), то гіперплощина буде прямою лінією. А якщо є 3 ознаки, то гіперплощина буде 2-вимірною площиною. Ми завжди створюємо гіперплощину з максимальним запасом, що означає максимальну відстань між точками даних.

					IA82.080BAK.003 ПЗ	Аркуш
						18
Зм.	Арку	№ докум.	Підпис	Дата		

Точки даних або вектори, які є найближчими до гіперплощини і які впливають на положення гіперплощини, називаються опорними векторами (векторами підтримки) оскільки ці вектори підтримують гіперплощину [16].

1.3.4 Метод наївного Байєса

Існує декілька алгоритмів, що утилізують метод наївного Байєса, а саме: гаусівський наївний Байєс, багаточленний наївний Байєс, доповнюючий наївний Байєс та наївний Байєс Бернуллі [17].

Наївний Байєс — це група контрольованих алгоритмів класифікації машинного навчання на основі теореми Байєса, формулу якої можна побачити нижче (формула 1.1). Це проста методика класифікації, але має високу функціональність. Вони знаходять застосування, коли розмірність вхідних даних висока. Складні задачі класифікації також можуть бути реалізовані за допомогою наївного байєсового класифікатора [18].

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}, \quad (1.1)$$

де A та B – це події та $P(B) \neq 0$.

$P(A | B)$ є умовною ймовірністю: ймовірність настання події A , якщо B є істинною. Її також називають апостеріорною ймовірністю A даного B .

$P(B | A)$ також є умовною ймовірністю: ймовірність настання події B , якщо A є істинною. Його також можна інтерпретувати як ймовірність A за умови фіксованого B , оскільки $P(B | A) = L(A | B)$.

$P(A)$ та $P(B)$ – це ймовірності спостереження A та B відповідно без будь-яких умов; вони відомі як гранична ймовірність або попередня ймовірність.

					IA82.080BAK.003 ПЗ	Аркуш
						19
Зм.	Арку	№ докум.	Підпис	Дата		

Під час роботи з безперервними даними часто прийнято вважати, що безперервні значення, пов'язані з кожним класом, розподіляються відповідно до нормального (або гаусового) розподілу. Гаусівський наївний Байєс підтримує безперервні ознаки та моделі, кожна з яких відповідає гауссовому (нормальному) розподілу. Підхід до створення простої моделі полягає в припущенні, що дані описуються гаусівським розподілом без коваріації (незалежних розмірів) між вимірами. Ця модель може бути підібрана, просто знайшовши середнє значення та стандартне відхилення точок у кожній мітці, що є все, що потрібно для визначення такого розподілу [19].

На рисунку 6 показано, як працює класифікатор Гаусса наївного Байєса. У кожній точці даних обчислюється відстань z-оцінки між цією точкою та кожним середнім класом, а саме відстань від середнього класу, поділена на стандартне відхилення цього класу.

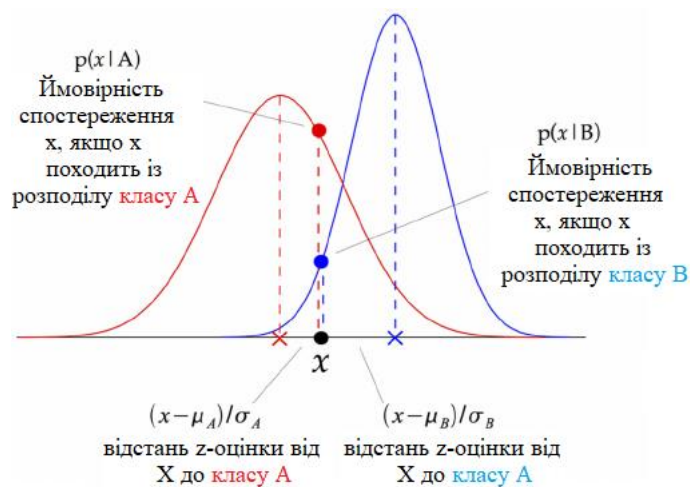


Рисунок 6 – Схема роботи класифікатора Гаусса наївного Байєса [19]

Багаточленний наївний Байєс — це спеціалізована версія наївного Байєса, призначена для роботи з текстовими документами з використанням кількості слів як

					IA82.080BAK.003 ПЗ	Аркуш
						20
Зм.	Арку	№ докум.	Підпис	Дата		

основного методу обчислення ймовірності. Це проста, але елегантна модель для обробки класифікації, яка включає прості класи, які не передбачають аналіз настроїв (складні вирази емоцій, такі як сарказм). Навіть маючи невелику кількість тестових даних, ми все одно можемо створити точну або майже точну модель класифікації [20].

Доповнюючий наївний Байєс є адаптацією стандартного алгоритму багаточленного наївного Байєса, який не дуже добре працює на незбалансованих наборах даних. Незбалансовані набори даних — це набори даних, у яких кількість прикладів одного класу перевищує кількість прикладів, що належать іншим класам. Це означає, що розподіл прикладів нерівномірний. З цим типом наборів даних може бути важко працювати, оскільки модель може легко переповнювати ці дані на користь класу з більшою кількістю прикладів. Доповнюючий наївний Байєс замість того, щоб обчислювати ймовірність належності предмета до певного класу, обчислює ймовірність належності предмета до всіх класів.

Алгоритм доповнюючого наївного Байєса виглядає так:

— для кожного класу обчислюється ймовірність того, що даний екземпляр йому не належить;

— після обчислення для всіх класів перевіряються всі обчислені значення і вибирається найменше значення;

— вибирається найменше значення (найменша ймовірність), оскільки це найменша ймовірність того, що це НЕ той конкретний клас. Це означає, що він має найвищу ймовірність належати до цього класу [21].

Наївний Байєс Бернуллі зазвичай використовується для дискретних даних і працює за допомогою розподілу Бернуллі. Основною особливістю цього алгоритму є те, що він приймає функції лише як двійкові значення, такі як «true» чи «false», так чи ні, успіх чи невдача, 0 чи 1 тощо. Отже, коли значення ознак двійкові, ми знаємо, що ми повинні використовувати наївний байєсовий класифікатор Бернуллі [22].

					IA82.080BAK.003 ПЗ	Аркуш
						21
Зм.	Арку	№ докум.	Підпис	Дата		

1.3.5 Метод найближчих сусідів

Метод найближчих сусідів це тип алгоритму навчання з вчителем, який використовується як для регресії, так і для класифікації. Алгоритм цього методу такий:

— передбачити правильний клас для даних тесту, обчислюючи відстань між тестовими даними та всіма навчальними точками;

— вибрати «К» кількість точок, яка близька до даних тесту;

— обчислити ймовірність того, що тестові дані належать до класів навчальних даних «К», і буде обрано клас із найвищою ймовірністю.

У разі регресії значенням є середнє значення «К» вибраних навчальних точок [23].

1.3.6 Ансамблеві методи

Ансамблеві методи – це методи, які спрямовані на підвищення точності результатів у моделях шляхом комбінування кількох моделей замість використання однієї моделі. Комбіновані моделі значно підвищують точність результатів.

Найпопулярнішими ансамблевими методами є беггінг, бустінг та стакінг. Методи ансамблю ідеально підходять для регресії та класифікації, де вони зменшують зміщення та дисперсію, щоб підвищити точність моделей [24].

Беггінг в основному використовується в класифікації та регресії. Це підвищує точність моделей за допомогою дерев рішень, що значною мірою зменшує дисперсію. Цей тип навчання означає: багато разів навчати ансамбль на випадкових вибірках даних. І в кінцевому підсумку усереднити відповіді. Це виглядає як голосування за найбільш популярну відповідь, де багато моделей працюють паралельно.

					ІА82.080БАК.003 ПЗ	Аркуш
						22
Зм.	Арку	№ докум.	Підпис	Дата		

Стаккінг Спершу навчають кілька алгоритмів, потім результати їх роботи показують останньому алгоритму. Саме він і приймає остаточне рішення. Стекінг – хороший, але найменш точний ансамбль серед інших методів.

Буустінг Цей спосіб включає послідовне навчання алгоритмів. Тобто спершу навчаємо перший і відзначаємо місця, де він помилився. Потім навчаємо другий, особливу увагу приділяючи місцям на яких помилявся перший. І так далі. До необхідного результату [25].

Висновки до розділу 1

В розділі було досліджено предметну область розпізнавання тексту, засобів та інструментів класифікації та загальної методики вибору потрібного класифікатора.

Аналіз дозволив вибрати найефективніші методи для розробки програмного застосунку. Так як більшість відкритих баз даних з навчальними даними надають значно менше ніж 100,000 записів у одній базі та з огляду на те, що класифікація тексту включає в себе існування помічених даних, було прийнято рішення використовувати класифікатори лінійного типу та наївного Байєса.

					ІА82.080БАК.003 ПЗ	Аркуш
						23
Зм.	Арку	№ докум.	Підпис	Дата		

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Більшість існуючих рішень мають декілька недоліків, що не дозволяють або роблять незручним їх повсякденне використання для різних типів користувачів. Деякі рішення можливо використовувати тільки на самій платформі, тобто не можливо перевіряти новини з різних джерел. Інші варіанти покладаються на волонтерську діяльність людей та їх власний розсуд, що може призвести до упереджених та помилкових результатів.

2.1 False, Misleading, Clickbait-y, and/or Satirical “News” Sources

Мелісса Зімдарс, професор медіа в коледжі Меррімак у Массачусетсі, склала список «підроблених, неправдивих або регулярно вводячих в оману веб-сайтів», які цілеспрямовано публікують підроблену інформацію або є абсолютно ненадійними. Список складається з 176 сайтів, що має дозволяти перевірити більшість сайтів на надійність або використовувати цей список у сторонньому застосунку як базу даних.

Але у цього підходу є мінуси. Для власників сайту з фейковими новинами набагато простіше змінити доменне ім'я, ніж для потенційних волонтерів відслідковувати такі зміни. Таким чином, цей список стає застарілим дуже швидко і не існує мотивації для достатньої кількості людей, щоб підтримувати його в оптимальному стані [26]. Отже будь-які системи, по вимушені покладатися на ручну обробку новин людьми є безнадійною справою.

					ІА82.080БАК.003 ПЗ	Аркуш
						24
Зм.	Арку	№ докум.	Підпис	Дата		

2.2 Fact-checking on Facebook

У соціальній мережі Facebook існує програма для виявлення фейкових новин. Вона працює наступним чином: на базі зворотного зв'язку від користувачів, новини потрапляють у поле зору так званих «перевірячів фактів», що є спеціалістами третіх сторін на окладі. Якщо ці спеціалісти маркують новини як неправду, то ці новини будуть скриті з мережі, а групи, що їх розповсюджують взяті на контроль [26].

Головними недоліками цього підходу є те, що система вимушена покладатися на зворотній зв'язок користувачів, та має підтримувати штат спеціалістів, для перевірки великої кількості новин, що зростає паралельно з ростом популярності сервісу. Також ця система працює тільки для однієї соціальної мережі, що не дозволяє її широко використовувати [27].

2.3 Fake news debunker by InVID & WeVerify

Це розширення для браузеру Chrome дозволяє швидко отримувати контекстну інформацію про зображення та відео у Facebook і Twitter, а також у відео YouTube, щоб виконувати зворотний пошук зображень у пошукових системах, запитувати базу даних відомих підробок, фрагментувати відео на ключові кадри, покращувати й досліджувати ключові кадри та зображення та застосувати розширений інструментарій криміналістичної експертизи до зображень, підозрюваних у маніпуляціях [28]. На рисунку 7 наведено інтерфейс цього розширення.

					ІА82.080БАК.003 ПЗ	Аркуш
						25
Зм.	Арку	№ докум.	Підпис	Дата		

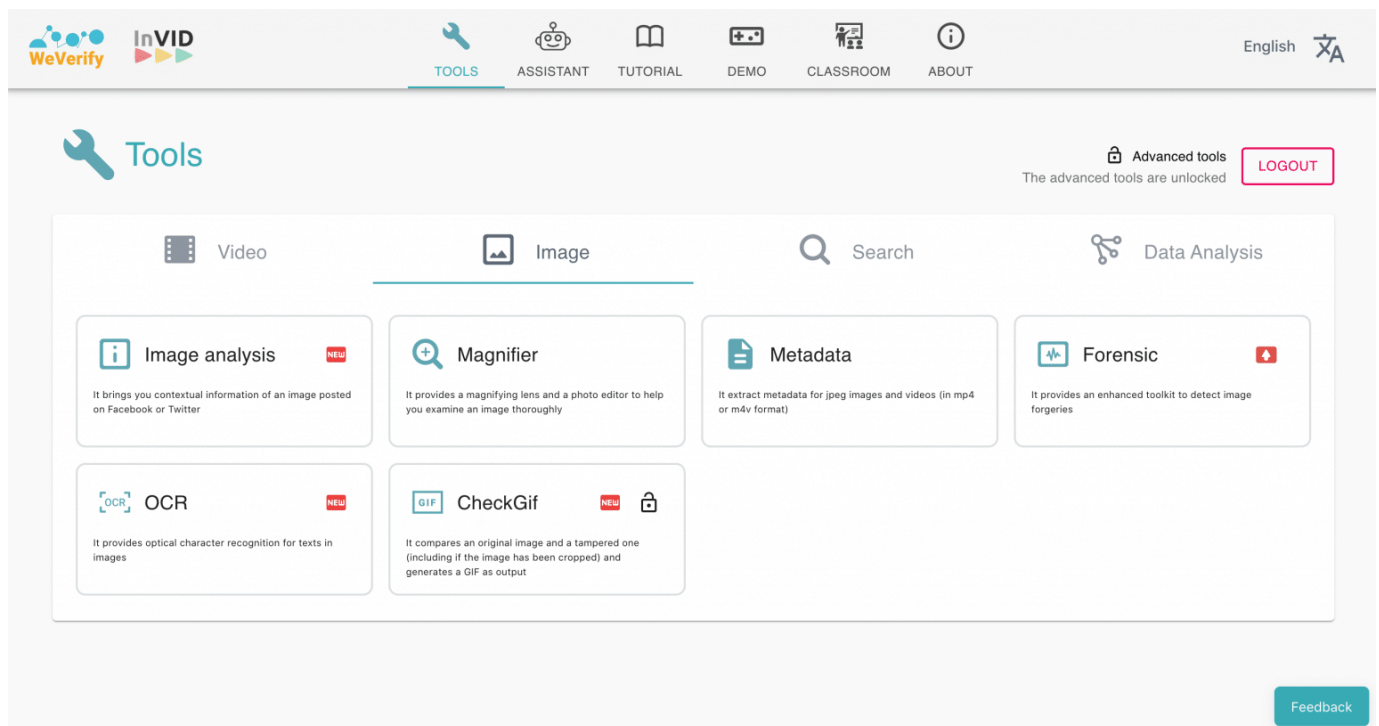


Рисунок 7 – Інтерфейс розширення Fake news debunker by InVID & WeVerify [30]

Одним із головних недоліків цього розширення є те, що воно доступне тільки для користувачів браузерів Chrome, та не дозволяє перевіряти текстові новини на правдивість, а тільки аналізувати зображення та кадри відео [29].

Висновки до розділу 2

Як можна побачити, усі приведені рішення дійсно мають якісь недоліки, що унеможлиблює їх широке використання. Застосунок, розроблений в рамках цього проєкта буде готовий узяти на себе багато завдань, з якими не можуть впоратися існуючі рішення, що доволі часто покладаються на реальних людей для того, щоб аналізувати кожну новину, тоді, коли було б набагато ефективніше навчити це робити машину, яка б могла вчитися на неправильних результатах та не була б упередженою.

					IA82.080BAK.003 ПЗ	Аркуш
						26
Зм.	Арку	№ докум.	Підпис	Дата		

3 ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ

Для чіткого розуміння причин та аргументації вибору технологій розробки необхідно розглянути доступні варіанти для побудова системи.

3.1 Вибір нейронної мережі(алгоритму)

У розділі 1 було прийнято рішення використовувати лінійні класифікатори та наївного Байєса.

Існує багато класифікаторів побудованих на основі цих типів, тому реалізовувати будемо чотири з них – два лінійного типу, та два типу наївного Байєса.

Бібліотека `scikit-learn` надає доступ до алгоритмів навчання моделей. У роботі буде використано такий функціонал:

- `sklearn.linear_model.PassiveAggressiveClassifier`;
- `sklearn.linear_model.Perceptron`;
- `sklearn.naive_bayes.MultinomialNB`;
- `sklearn.naive_bayes.ComplementNB`.

3.2 Засоби розробки

Так як розробка застосунку для аналізу новин на фейковість передбачає використання засобів обробки та аналізу текстових даних, постає необхідність використання специфічних інструментів.

					IA82.080BAK.003 ПЗ	Аркуш
						27
Зм.	Арку	№ докум.	Підпис	Дата		

3.2.1 Мова програмування Python

Python — це потужна мова програмування, яка легко вивчається. Вона має ефективні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованою природою роблять його ідеальною мовою для написання скриптів і швидкої розробки додатків у багатьох областях на більшості платформ [31].

Окрім переваг у вигляді легкості вивчення та зрозумілого синтаксису, Python також володіє великою кількістю бібліотек і фреймворків, що значно спрощують створення нейронних мереж та їх навчання.

Також однією з сильних сторін Python є його кросплатформеність, що дозволяє запускати застосунки на усіх широко-застосованих операційних системах без або з мінімальними модифікаціями.

3.2.2 Середовище розробки Visual Studio Code

У якості середовища розробки було обрано Visual Studio Code. Visual Studio Code — це легкий, але потужний редактор вихідного коду, який працює на Windows, macOS та Linux. Він поставляється з вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов (наприклад, C++, C#, Java, Python, PHP, Go) і середовищ виконання (наприклад, .NET і Unity) [32].

Саме багатий вибір розширень є основною перевагою цього редактору. Для цього проекту було використано розширення для підтримки Python, автоматичної перевірки коду, та розширення для взаємодії з GitHub.

					IA82.080BAK.003 ПЗ	Аркуш
						28
Зм.	Арку	№ докум.	Підпис	Дата		

3.2.3 Графічний інтерфейс Tcl/Tk за допомогою Tkinter

Взагалі існує декілька популярних бібліотек для створення графічного інтерфейсу, і переважна більшість з них може бути використана з Python. При підготовці до створення програмного застосунку були проаналізовані наступні модулі Python: PyGObject, PySide, wxPython та Tkinter [33].

PyGObject надає доступ до всієї програмної платформи GNOME. Він повністю сумісний з GTK+ 3, що надало б йому багато переваг, якби GTK не був так зафіксований на Linux, що робить адаптацію застосунків, використовуючих GTK важкою на Windows та MacOS. Так як однією з цілей створення програмного застосунку є його кросплатформеність, PyGObject не відповідає вимогам.

PySide розроблений The QT Company, розробником бібліотеки графічного інтерфейсу QT часто називається лідером індустрії. PySide дозволяє створювати застосунки з нативними компонентами інтерфейсу та ізолює програміста від великої кількості деталей імплементації. Плата за це – велике місце, що займають створені застосунки. Клієнт має завантажити усю платформу QT для того, щоб запустити застосунок, що не є доречним [34].

wxPython – це набір інструментів для мови програмування Python, що дозволяє програмістам створювати програми з надійним, високофункціональним графічним інтерфейсом користувача, просто та легко. Він реалізований як модуль розширення Python, який обгортає популярну міжплатформену бібліотеку wxWidgets, написану на C++. Нажаль, використання C++ позначається на загальній швидкості застосунку. [33][34]

					IA82.080BAK.003 ПЗ	Аркуш
						29
Зм.	Арку	№ докум.	Підпис	Дата		

Tkinter — це тонкий об'єктно-орієнтований шар поверх Tcl/Tk. Він має перевагу в тому, що входить до стандартної бібліотеки Python, що робить його найбільш зручним і платформи-сумісним набором інструментів для програмування графічного інтерфейсу. І Tk, і Tkinter доступні на більшості платформ Unix, а також у системах Windows і Macintosh. Починаючи з випуску 8.0, Tk пропонує оригінальний вигляд віджетів на всіх платформах. Tk використовує C, що робить його швидким та легким у роботі. [35]

Таким чином, було обрано Tkinter для створення графічного інтерфейсу через його швидкість, простоту та кросплатформеність.

3.2.4 Інструментарій дистрибуції коду – PyInstaller

Однією з цілей застосунку є необхідність бути легкою у використанні навіть для тих користувачів, що мають тільки поверхнєве знайомство за комп'ютерами, тому існує необхідність зробити установку застосунку як можна простіше.

Одним з способів це зробити – це використання різних інструментів для дистрибуції коду. Ці інструменти “заморожують” код, автоматично додають необхідні файли, бібліотеки та інтерпретатор Python. Завдяки такому підходу, користувачам не потрібно мати на комп'ютері встановлений Python та встановлювати додаткові бібліотеки вручну. Користувачу для початку роботу з програмою достатньо запустити “.exe” файл [36].

У таблиці 3.1 перераховано основні засоби для “заморожування” коду та їхня характеристика: чи підтримуються окремі операційні системи та Python третьої версії, та ліцензія.

					IA82.080BAK.003 ПЗ	Аркуш
						30
Зм.	Арку	№ докум.	Підпис	Дата		

Таблиця 3.1 – Характеристика інструментів дистрибуції коду

Назва	Windows	Linux	OsX	Python3	Ліцензія
bbFreeze	Так	Так	Так	Ні	MIT
py2exe	Так	Ні	Ні	Так	MIT
pyInstaller	Так	Так	Так	Так	GPL
cx_Freeze	Так	Так	Так	Так	PSF
py2app	Ні	Ні	Так	Так	MIT

Як можна побачити з таблиці 3.1, тільки PyInstaller та cx_Freeze одночасно підтримують усі три найбільш розповсюджені операційні системи та найновішу версію Python.

cx_Freeze є відомим своїми широкими можливостями конфігурації, коли як PyInstaller є біль простим інструментом, функціоналу якого більш ніж достатньо для дистрибуції створеного застосунку, тому було обрано рішення використовувати саме його.

3.2.5 Бібліотеки та модулі Python

Створений програмний застосунок використовує декілька додаткових бібліотек для виконання своїх функцій.

По-перше, для аналізу новин на предмет фейковості, було прийнято рішення використовувати бібліотеку scikit-learn. scikit-learn – це бібліотека машинного навчання для Python, що включає у себе прості та ефективні інструменти для прогнозного аналізу даних доступні кожному та з можливістю багаторазового використання в різних контекстах. Ця бібліотека створена на основі NumPy, SciPy та matplotlib та розповсюджується у вигляді відкритого вихідного коду під ліцензією

					IA82.080BAK.003 ПЗ	Аркуш
						31
Зм.	Арку	№ докум.	Підпис	Дата		

BSD. Ця бібліотека може бути використана для задач класифікації, регресії, кластеризації, зменшення розмірності, вибору моделі та попередньої обробки [37].

По-друге, для перетворення необроблених табличних даних у вид придатний для аналізу scikit-learn було використано бібліотеку pandas. pandas – це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та маніпулювання даними з відкритим вихідним кодом, побудований на основі мови програмування Python [38].

По-третє, для забезпечення можливості зберігання та експортування створених моделей, було використано вбудований модуль pickle. Модуль pickle реалізує двійкові протоколи для серіалізації та десеріалізації об'єктної структури Python. «Піклінг» — це процес, за допомогою якого ієрархія об'єктів Python перетворюється на потік байтів, а «Анпіклінг» — це зворотна операція, за допомогою якої потік байтів (з двійкового файлу або байтоподібного об'єкта) перетворюється назад в ієрархію об'єктів. Цей модуль дозволяє зберігати більшість підтримуваних Python структур на жорсткому диску з можливістю подальшого використання та розповсюдження [39].

По-четверте, для забезпечення безперервної роботи застосунку, було використано модуль threading. Цей модуль дозволяє створювати потоки виконання, що дає можливість виділити графічний інтерфейс та функції, потребуючі довгого обчислення, у окремі потоки, таким чином гарантуючи чуйність графічного інтерфейсу до вводу користувача, при цьому продовжуючи виконання функцій [40].

Висновки до розділу 3

У розділі були приведені чотири моделі нейронних мереж до яких надає доступ scikit-learn.

					ІА82.080БАК.003 ПЗ	Аркуш
						32
Зм.	Арку	№ докум.	Підпис	Дата		

Також були обрані засоби розробки відповідно до поставленої задачі. Для реалізації застосунку було обрано мову програмування Python та середовище розробки Visual Studio Code. У якості інструменту дистрибуції коду – PyInstaller. Графічний інтерфейс було прийнято рішення виконувати за допомогою Tkinter. Для створення моделей нейронних мереж було обрано бібліотеку scikit-learn та pandas, що надає сервіс обробки даних перед тим, як їх можна використати у scikit-learn. Для забезпечення можливості збереження моделей було обрано модуль pickle, а для безперебійного виконання – threading, для створення потоків.

					IA82.080BAK.003 ПЗ	Аркуш
						33
Зм.	Арку	№ докум.	Підпис	Дата		

4 РОЗРОБКА ПЗ

В ході виконання індивідуального дослідницького проєкту був розроблений застосунок для виявлення фейкових новин.

Розроблена система здатна вирішити проблему фейкових новин, що є головною частиною її функціоналу, але на відміну від інших існуючих рішень (дивися розділ 2), ця програма не потребує доступу до інтернету, є кросплатформеною за рахунок використання мови програмування Python та прозорою у свої діяльності через відкритість вихідного коду.

Також, програма вирішує проблему недоліку функціоналу, що не дає бажаних результатів просунутим користувачам, або навпаки надлишку цього функціоналу, що робить користування програмою та ознайомлення з програмою набагато складнішою задачею для пересічних користувачів, які не бажають витратити багато часу на вивчення усього доступного функціоналу, а тільки потребують існуючого готового рішення для виявлення фейкових новин. Досягається це шляхом розділення програмного застосунку на два типи користувацького інтерфейсу.

Перший простий тип інтерфейсу тільки надає можливість перевіряти новини на фейковість та змінювати теми інтерфейсу. Тоді як другий тип інтерфейсу, просунутий, дає повний контроль над створенням, навчанням та поведінкою моделей. Також він надає більше інформації користувачу про точність створеної моделі та результативність навчання.

Усі налаштування користувача такі як вибір теми та встановлення типу інтерфейсу зберігаються між перезапусками застосунку.

Система є готовою для повсякденного використання, та впровадження у закладах, які цього потребують.

					IA82.080BAK.003 ПЗ	Аркуш
						34
Зм.	Арку	№ докум.	Підпис	Дата		

Потрібно окрему відмітити, що за наявності більшої кількості обчислювальних ресурсів та всеосяжної вибірки даних, можна значно підвищити точність здійснюваних розрахунків, тим самим даючи можливість отримувати точніші результати аналізу новин на предмет фейковості.

Для використання у країнах, де більшість новин публікується не англійською, а місцевою мовою, програма потребує надання їй підбірки навчальних даних на цій самій місцевій мові, що має бути створена окремо та не є однією з цілей цього індивідуального дослідницького проєкту.

Програмний застосунок складається з наступних модулів:

— `fnd.py` – модуль класифікації тексту, що відповідає за створення та навчання моделей, а також за аналіз тексту цими моделями;

— `fndGUI.py` – модуль, що відповідає за створення користувацького інтерфейсу, створення потоків виконання, нормалізацію вхідних даних та реалізує логіку модулів класифікації тексту та створення файлів конфігурації;

— `configurator.py` – модуль, що відповідає за логіку створення файлів конфігурації, що дозволяє запам'ятовувати стан програми після внесених користувачем змін.

4.1 Опис внутрішньої роботи

Для того, щоб застосунок був корисним для будь-яких користувачів, було прийнято рішення «пакувати» одну створену та навчену модель разом з програмою то поширювати її саме у такому форматі.

Ця базова модель має назву «basic» та зберігається у папці «models» із розширенням «.pickle» як і будь-яка інша, що користувачі можуть створити. Для створення цієї моделі використовувався клас `PAClassifier`, що класифікує вхідний

					IA82.080BAK.003 ПЗ	Аркуш
						35
Зм.	Арку	№ докум.	Підпис	Дата		

текст за допомогою пасивно-агресивного алгоритму, що є надбудовою лінійних методів описаних у розділі 1. Цей алгоритм залишається пасивним для правильного результату класифікації, а в разі прорахунку, оновлення та коригування стає агресивним.

Параметри цієї моделі були обчислені методом підбору, тобто була обрана та комбінація параметрів, що повторно призводила до створення моделі з більшою точністю. Точні значення параметрів наступні:

- тренувальний розмір склав 75% від розміру набору даних;
- тестувальний розмір склав 25% від розміру набору даних;
- кількість ітерацій 1000000.

Набір даних, на основі якого проходило навчання моделі, є заново створеною копією відкрито доступної бази фейкових новин, що публікується Університетом Вікторії [41]. Університет надає дані двома великими файлами, один з фейковими новинами, та інший з реальними. Для підвищення швидкості розробки ці файли були зменшені у розмірі шляхом видалення частини даних та об'єднані в один файл, де кожна новина має позначку: «Fake» чи «Real».

Отримана в результаті навчальна вибірка включає новини тільки англійською мовою, тому створена на її основі модель не в змозі об'єктивно аналізувати інші новини на інших мовах, що є одним з потенційних покращень, висвітлених у висновках до цього дослідницького проєкту.

Ця навчальна вибірка, як і усі інші, що можуть бути доданими користувачами, зазнає конвертації у матрицю, що складається з числових значень кількості разів кожне слово зустрічається у тексті кожної комірки даних та числового значення того, на скільки це слово є важливим. Важливість слова корелює з частотою його появи в інших, не пов'язаних текстах, де популярніше слово має меншу важливість. Таким

					IA82.080BAK.003 ПЗ	Аркуш
						36
Зм.	Арку	№ докум.	Підпис	Дата		

чином, класифікатор отримує можливість звертати більшу увагу на ті слова, що з більшою долею вірогідності несуть сенсове навантаження.

Наступним кроком пасивно-агресивний класифікатор отримує на вхід отриману матрицю, та за допомогою алгоритму побудованого на основі лінійних моделей, розглянутих у розділі 1, проводить навчання. У конкретному випадку базової моделі результатом навчання стала точність у розмірі 94%, що каже про високу ефективність самого алгоритму, адже інші використані у роботі класифікатори принесли гірші результати, а саме: ~92% у випадку перцептрона, та від ~82-86% у випадках методів наївного Байєса. Ця точність не означає, що модель буде розпізнавати новини з вказаною ефективністю, а тільки вказує на точність у навчанні. Для оцінки точності на прикладах реальних новин див. розділ 6.

Навчена модель зберігається на диску користувача та може бути використана на реальних прикладах новин.

4.2 Розробка діаграми класів

Діаграма класів — це тип діаграми статичної структури, яка описує структуру системи, показуючи її класи, їх атрибути, операції (або методи) і зв'язки між об'єктами. Класи на діаграмі класів представляють як основні елементи взаємодії в програмі, так і класи, які будуть створені.

Розроблена діаграма класів розміщена в документі IA82.080БАК.003 Д1.

Як можна побачити з цієї діаграми, одним з головних елементів системи є клас FakeDetector, що знаходиться у модулі fndGUI.py. Саме він відповідає за користувацький інтерфейс та використання логіки класів з інших модулів. Цей клас може бути тільки одним під час виконання програми, так як створення нового класу означає запуск програми повторно.

					IA82.080БАК.003 ПЗ	Аркуш
						37
Зм.	Арку	№ докум.	Підпис	Дата		

Клас `Detector`, що знаходиться у модулі `find.py` містить у собі більшість змінних та методів, які є загальними для усіх наслідуючих класів.

Класи `LinearDetector` та `BayesDetector`, що наслідують від класу `Detector` є двома більш спеціалізованими версіями цього класу. Вони не передбачені для прямого використання, але можуть бути корисними під час створення додаткових класифікаторів, виступаючи платформою для них.

Нарешті в нас є класи `PAClassifier` та `Perceptron`, що є нащадками `LinearDetector` та `ComplNB` з `MultiNB`, що у свою чергу є нащадками `BayesDetector`.

Ці класи передбачені для використання як класифікатори тексту. Кожен клас реалізує свій вид класифікатора.

Окремо потрібно відмітити, що у програмі у кожен окремий момент часу може бути тільки один ініціалізований клас з перерахованих, або жодного. Це відбувається через те, що при зміні моделі, її клас зберігається на диск, та зникає з робочої пам'яті програми та заміщується класом нової моделі.

`Configurator` клас з модуля `configurator.py` існує задля того, щоб зробити можливим зберігання налаштувань користувача на диск. Цей клас реалізує усю потрібну логіку та керується двома простими для споживача його функціоналу методами: `get_prefs()` та `update_prefs()`. `update_prefs()` використовується споживачами при зміні налаштувань, а `get_prefs()` при потребі зчитати інформацію про налаштування. Цей клас завжди створюється під час запуску застосунку у єдиному екземплярі.

4.3 Розробка діаграми розгортання

Діаграма розгортання моделює фізичний розвиток артефактів на вузлах.

Розроблена діаграма розгортання розміщена в документі IA82.080БАК.003 Д2.

					IA82.080БАК.003 ПЗ	Аркуш
						38
Зм.	Арку	№ докум.	Підпис	Дата		

Як можна побачити з діаграми, головним вузлом системи є комп'ютер користувача, який виступає базою для програми, що виконується інтерпретатором мови програмування Python та для двох артефактів сховища даних, а саме файлів конфігурації, що зберігають налаштування користувача, та файлів збережених моделей. У цей самий час, програма володіє завантаженими у пам'ять копіями цих файлів.

Тренувальні дані можуть зазвичай можуть бути завантажені двома шляхами, або з будь-якого фізичного носія, або з веб-серверу через вказування URL до файлу.

4.4 Розробка діаграми варіантів використання

Діаграма варіантів використання (також відома як діаграма прецедентів) — це графічне зображення можливої взаємодії користувача з системою. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів, яких має система, і часто супроводжується діаграмами інших типів. Варіанти використання представлені або колами, або еліпсами. Актори часто зображені у вигляді фігурок.

Розроблена діаграма варіантів використання розміщена в документі IA82.080БАК.003 ДЗ.

Логіка роботи програми поділяє користувачів на два типи: просунутий та пересічний. Саме це обумовило використання двох акторів на діаграмі прецедентів. При цьому просунутий користувач отримує доступ до усіх варіантів використання, що і простий користувач, але має доступ до більшої кількості варіантів.

Деякі варіанти використання супроводжуються іншими варіантами, наприклад при зміні налаштувань користувач може змінити тему, тип інтерфейсу і так далі. Зміна налаштувань супроводжується обов'язковим збереженням змін на диск задля запам'ятовування преференцій користувача.

					IA82.080БАК.003 ПЗ	Аркуш
						39
Зм.	Арку	№ докум.	Підпис	Дата		

4.5 Розробка діаграми активності

Діаграми активності — це графічне зображення робочих процесів поетапних дій і дій з підтримкою вибору, ітерації та паралельності.

Розроблена діаграма діяльності розміщена в документі ІА82.080БАК.003 Д4.

Дана діаграма діяльності була побудована від лица програми і являє собою загальний алгоритм дій застосунку.

Під час кожного запуску програми, виконується декілька підготовчих дій. Спочатку перевіряється чи існує у досяжності застосунку папка з моделями, якщо так, то виконання продовжується, якщо ні, то папка створюється і виконання також продовжується.

Далі, перевіряється конфігураційний файл з налаштуваннями користувача, якщо такий існує, то програма продовжує завантаження з оглядом на збережені налаштування, якщо ні, то використовуються значення за замовчуванням.

Наступним кроком йде завантаження однієї з моделей, якщо існує хоча б одна модель, то виконується її завантаження, якщо ні, то виводиться повідомлення, що користувач має самостійно створити хоча б одну модель, для використання застосунку.

Далі, відбувається перевірка доступних тем користувача та ініціалізація інтерфейсу, і програма є повністю завантаженою, таким чином починається її стандартний робочий цикл.

Кожні декілька моментів, програма оновлює графічний інтерфейс та перевіряє наявність команд користувача у черзі команд. При відсутності команд, цикл повторюється, при наявності відбувається одна з дій, залежно від типу команди.

Якщо отримана команда є командою закриття застосунку, то програма закривається.

					ІА82.080БАК.003 ПЗ	Аркуш
						40
Зм.	Арку	№ докум.	Підпис	Дата		

Якщо отримана команда є командою створення нової моделі, то спочатку створюється новий потік даних, що дозволяє виконувати створення моделі і під цей самий час, продовжувати користування програмою. Інакше графічний інтерфейс не реагував би на команди користувача доти, доки йде виконання попередньої команди.

У випадку, коли команда користувача є командою зміни налаштувань, то спочатку змінюються обрані налаштування, а потім вони зберігаються поперх старих у файлі конфігурації. Після цього цикл повторюється.

В усіх інших випадках відбувається виконання команди та повернення до початку циклу.

4.6 Опис створених класів та їх функціоналу

У таблиці 4.1 можна побачити опис кожного зі створених класів у модулі `fnd.py`, що відповідає за логіку створення моделей, їх навчання та аналізу тексту.

Таблиця 4.1 – Опис класів у модулі `fnd.py`

Назва класу	Наслідує від	Опис
Detector	-	Є суперкласом усіх класів нижче. Реалізує увесь загальний функціонал
LinearDetector	Detector	Є суперкласом для усіх класів, що базуються на класифікаторах лінійного типу
BayesDetector	Detector	Є суперкласом для усіх класів, що базуються на класифікаторах типу наївного Байєса
PAClassifer	LinearDetector	Заснован на пасивно-агресивному класифікаторі
Percept	LinearDetector	Заснован на класифікаторі типу Перцептрон
MultiNB	BayesDetector	Заснован на класифікаторі типу многочленного наївного Байєса

					IA82.080BAK.003 ПЗ	Аркуш
						41
Зм.	Арку	№ докум.	Підпис	Дата		

CompINB	BayesDetector	Заснован на класифікаторі типу доповнюючого найвного Байєса
---------	---------------	---

У таблиці 4.2 можна побачити опис функцій кожного з класів, що були описані вище.

Таблиця 4.2 – Опис функцій класів у модулі fnd.py

Назва функції	Клас	Опис
__init__	Detector	Конструктор класу, отримує велику кількість параметрів та створює і навчає модель
predict	Detector	Отримує текст на вхід, аналізує його та повертає результат «FAKE» або «REAL»
__getattr__	Detector	Перезавантаження методу задля можливості використовувати pickle модуль
__init__	LinearDetector	Конструктор класу, отримує параметри, проводить попередню обробку
Detector.__init__	LinearDetector	Визиває конструктор суперкласу, передаючи оброблені параметри
__init__	PAClassifer	Конструктор класу, отримує параметри, проводить попередню обробку
LinearDetector.__init__	PAClassifer	Визиває конструктор суперкласу, передаючи оброблені параметри
__str__	PAClassifer	Перезавантаження методу задля можливості «надрукувати» клас
__init__	Percept	Аналогічно попередньому

					IA82.080BAK.003 ПЗ	Аркуш
						42
Зм.	Арку	№ докум.	Підпис	Дата		

LinearDetector.__init__	Percept	Аналогічно попередньому
__str__	Percept	Аналогічно попередньому
__init__	BayesDetector	Конструктор класу, отримає параметри, проводить попередню обробку
Detector.__init__	BayesDetector	Визиває конструктор суперкласу, передаючи оброблені параметри
__init__	MultiNB	Конструктор класу, отримає параметри, проводить попередню обробку
BayesDetector.__init__	MultiNB	Визиває конструктор суперкласу, передаючи оброблені параметри
__str__	MultiNB	Перезавантаження методу задля можливості «надрукувати» клас
__init__	ComplNB	Аналогічно попередньому
BayesDetector.__init__	ComplNB	Аналогічно попередньому
__str__	ComplNB	Аналогічно попередньому

Для збереження налаштувань користувача було створено клас Configurator у модулі configurator.py.

У таблиці 4.3 можна побачити опис кожної функції цього класу.

Таблиця 4.3 – Опис функцій класу Configurator

Назва	Параметри	Опис
__init__	self, folder='./settings', file_name='preferences.ini'	Конструктор класу; перевіряє чи існує вказана папка, якщо ні, то створює її, та файл конфігурації зі змінними за замовчуванням

					ІА82.080БАК.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		43

update_prefs	self, **kwargs	На вхід отримує будь-яку кількість параметрів типу «ключ: значення»; в залежності від отриманих оновлює конфігураційний файл, створюючи нові записи, або поверх старих
get_prefs	self	Повертає усі дані конфігурації для використання

Також існує головний модуль програми fndGUI.py, який містить клас FakeDetector, що відповідає за графічний функціонал, оперування класифікаторами, конфігуратором, створює потоки виконання та оперує можливістю збереженням моделей на диск. У таблиці 4.4 можна побачити опис функції цього класу.

Таблиця 4.4 – Опис функцій класу FakeDetector

Назва функції	Опис
__init__	Конструктор класу; відповідає за попередню ініціалізацію графічного інтерфейсу, завантажує налаштування користувача, визиває частину інших функцій
analyse	Спрацьовує при натисканні на кнопку «Analyse»; отримує текст з текстового поля, направляє його до моделі на аналіз та встановлює луйбл з результатом
set_interface	Встановлює просунутий чи простий інтерфейс користувача
get_models	Повертає усі назви моделей, що зберігаються на диску; використовується для створення можливості вибору моделей у графічному інтерфейсі
disable_widgets	Робить деякі частини інтерфейсу неактивними, якщо обраний класифікатор не підтримує деякий функціонал

					IA82.080BAK.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		44

export_model	Зберігає модель до вказаної директорії на диску
update_theme	Змінює тему застосунку
get_filename	Отримує ім'я обраного файлу з навчальними даними та намагається передбачити ім'я моделі, що користувач вибере
thread_helper	Використовується для створення нових потоків виконання
update_after_create	Оновлює деякі записи, після створення моделі
create_model	Створює модель з вказаними параметрами
pickle_model	Зберігає модель на диск
unpickle_model	Завантажує модель з диску у пам'ять
set_labels	Встановлює надписи на графічному інтерфейсі залежно від об'єкту
show_about	Створює додаткове вікно з інформацією про програму

Висновки до розділу 4

У розділі були викладені основні відомості про структуру створеного застосунку, а саме: модулі, класи та функції.

Кожний клас та функція супроводжувалися пояснюючими коментарями, що може стати у нагоді як посібник з документацією при подальшій роботі над застосунком для інших розробників та науковців.

Також було пояснено технічний аспект роботи застосунку на прикладі створення та роботи з базовою моделлю.

					ІА82.080БАК.003 ПЗ	Аркуш
						45
Зм.	Арку	№ докум.	Підпис	Дата		

5 КЕРІВНИЦТВО КОРИСТУВАЧА

Програмний застосунок було створено з урахуванням двох можливих робочих процесів націлених на дві різні категорії користувачів, а саме:

— на користувачів, які бажають, щоб програма “просто” працювала, тобто не потребувала додаткового налагодження чи спеціальних знань;

— на користувачів, які мають спеціальні знання та бажають отримати доступ до більш детального рішення, тобто до можливості тренувати свої моделі на своїх наборах даних та\або змінювати налаштування окремих моделей, бачити ефективність навчання моделей та маніпулювати ними іншими способами.

На рисунку 8 можна побачити, як виглядає головний екран застосунку. Головний екран ПЗ має простий та зрозумілий інтерфейс, саме те, що потрібно першому типу користувача, який бажає, щоб застосунок був легким у роботі, та не потребував додаткових умінь.

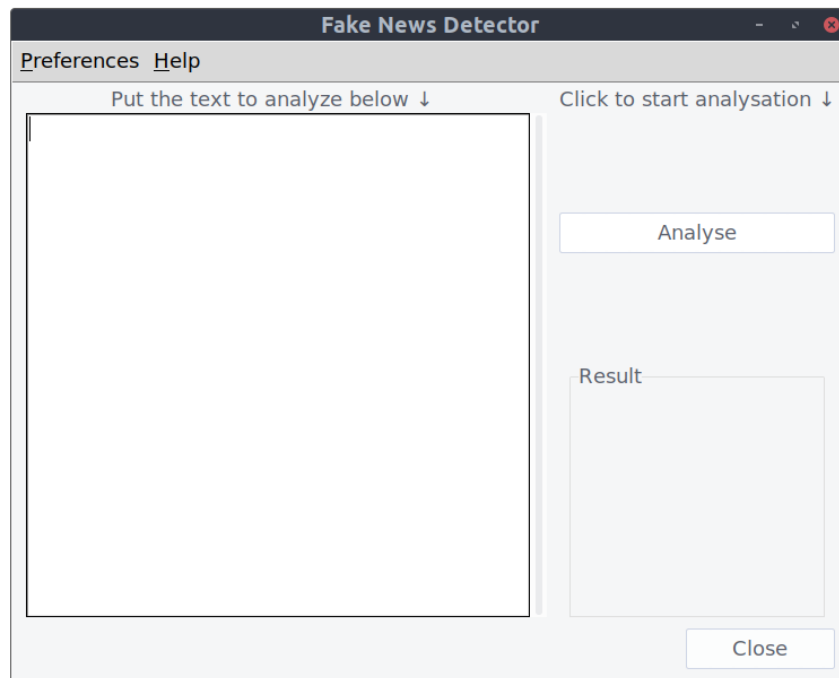


Рисунок 8 – Головний екран ПЗ

					ІА82.080БАК.003 ПЗ	Аркуш
						46
Зм.	Арку	№ докум.	Підпис	Дата		

Використання застосунку є досить легким, адже кожне поле супроводжується підказкою.

Звідси, користувач може проаналізувати текст методом копіювання чи прямого вводу у текстове поле, натиснути кнопку «Analyse» та отримати результат: «REAL» або «FAKE», як це можна побачити на рисунку 9, або закрити застосунок.

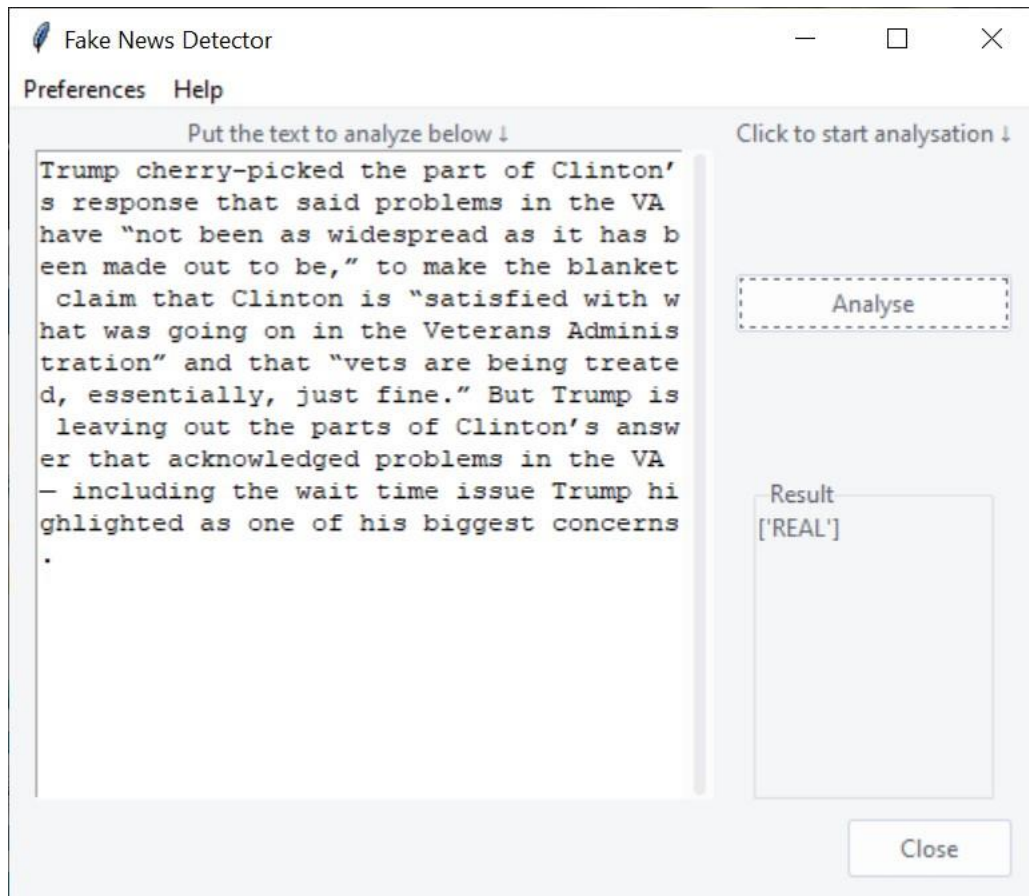


Рисунок 9 – Загальна схема роботи ПЗ

У верхній панелі знаходяться дві кнопки «Preferences» та «Help», при натисканні на останню, можна відкрити секцію «About», що надає інформацію про розробника, головні використані технології, а також ліцензію на програмний застосунок. Цей функціонал зображено на рисунку 10.

					IA82.080BAK.003 ПЗ	Аркуш
						47
Зм.	Арку	№ докум.	Підпис	Дата		

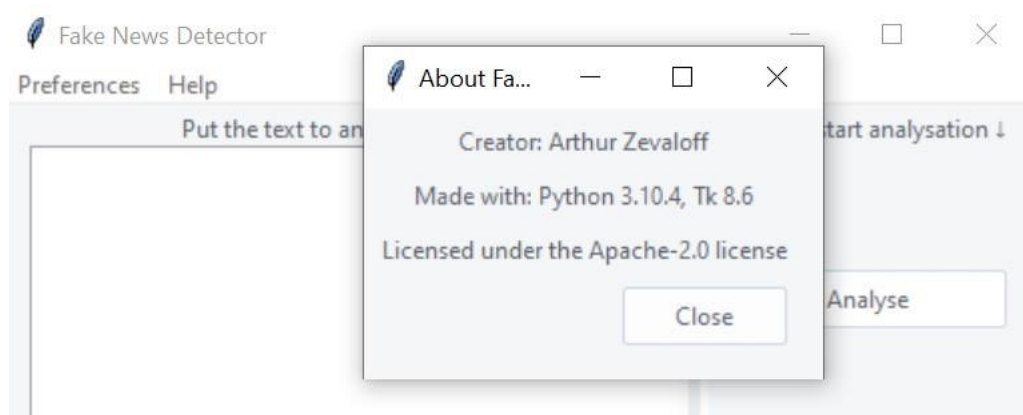


Рисунок 10 – Секція «About»

При натисканні на кнопку «Preferences» користувач отримує доступ до двох секцій «Theme» та «Activate advanced mode».

Як можна побачити на рисунку 11, секція «Theme» дає можливість обрати одну з декількох графічних тем користувача.

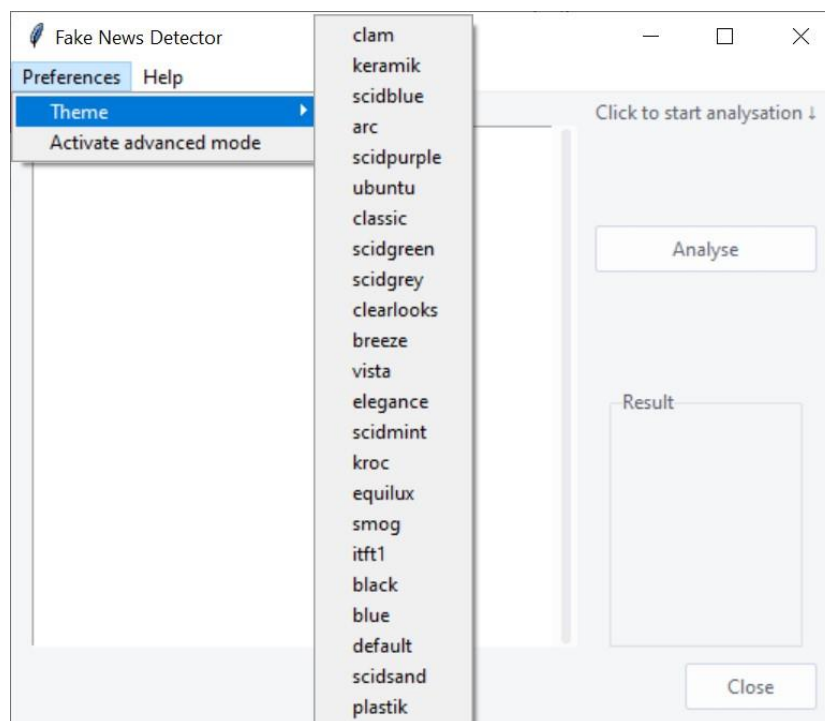


Рисунок 11 – Обирання теми користувача

					IA82.080BAK.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		48

Користувачі другої категорії, ті хто мають спеціальні знання та бажають мати більше контролю над їхніми моделями можуть активувати просунутий користувацький інтерфейс за допомогою натискання на секцію «Activate advanced mode». Як видно з рисунка 12, це відкриває додаткове поле з параметрами та можливістю створення нових моделей, їх зміни та експортування.

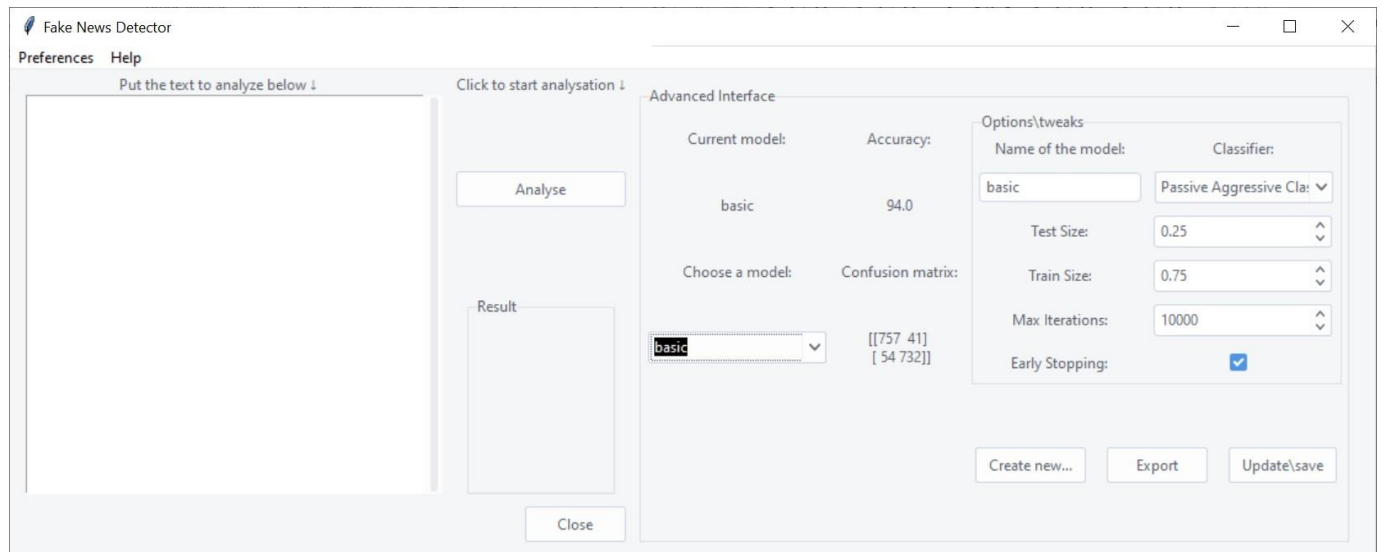


Рисунок 12 – Розвернутий інтерфейс

Просунутий користувацький інтерфейс можна умовно поділити на три частини.

У першій частині знаходяться дані про точність, назву та результатів навчання обраної моделі, з можливістю зміни поточної моделі, яка буде оцінювати текст.

У другій частині розташовано вікно «Options\Tweaks». Воно дозволяє змінити налаштування існуючої моделі чи внести корективи під час створення нової. Можна змінити ім'я, обрати інший класифікатор (у наявності пасивно-агресивний класифікатор, перцептрон, многочленний наївний Байєс та доповнюючий наївний Байєс). Далі можна обрати пропорцію даних, яка буде використана для тренування, а яка для тестування. Далі існує можливість ввести кількість бажаних ітерацій, що може

					ІА82.080БАК.003 ПЗ	Аркуш
						49
Зм.	Арку	№ докум.	Підпис	Дата		

підвищити точність створеної моделі, та указати, чи бажає користувач перервати навчання, коли точність перестає зростати.

У третій частині інтерфейсу розташовані три кнопки «Create new...», «Export» та «Update\save».

Кнопка «Export» дозволяє зберегти існуючу модель у будь-якому місці на диску, для зберігання чи для поширення з іншими людьми.

Кнопка «Update\save» дозволяє зберегти зміни до існуючої моделі, або зберегти нову модель, для якої вже були завантажені дані та обрані бажані опції.

Кнопка «Create new...» дозволяє створити нову модель на основі якоїсь бази даних. На рисунку 13 можна побачити процес обирання даних, а на рисунку 14 стан інтерфейсу перед зберіганням на диск.

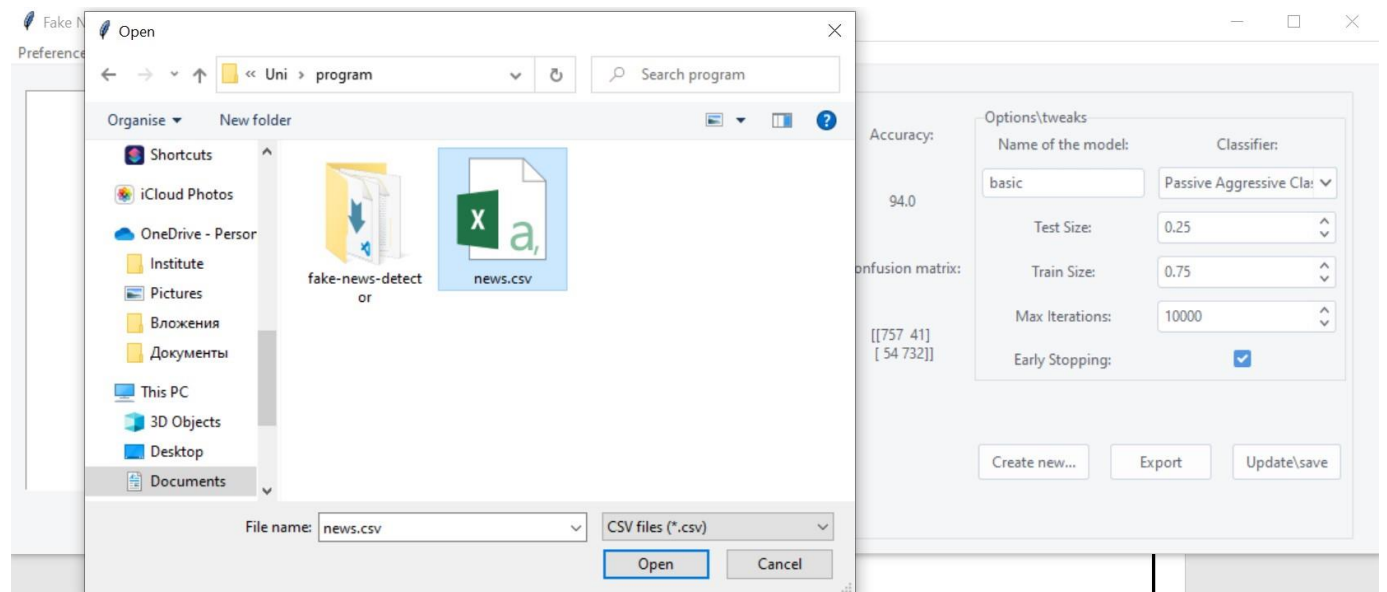


Рисунок 13 – Обирання навчальних даних

					IA82.080BAK.003 ПЗ	Аркуш
						50
Зм.	Арку	№ докум.	Підпис	Дата		

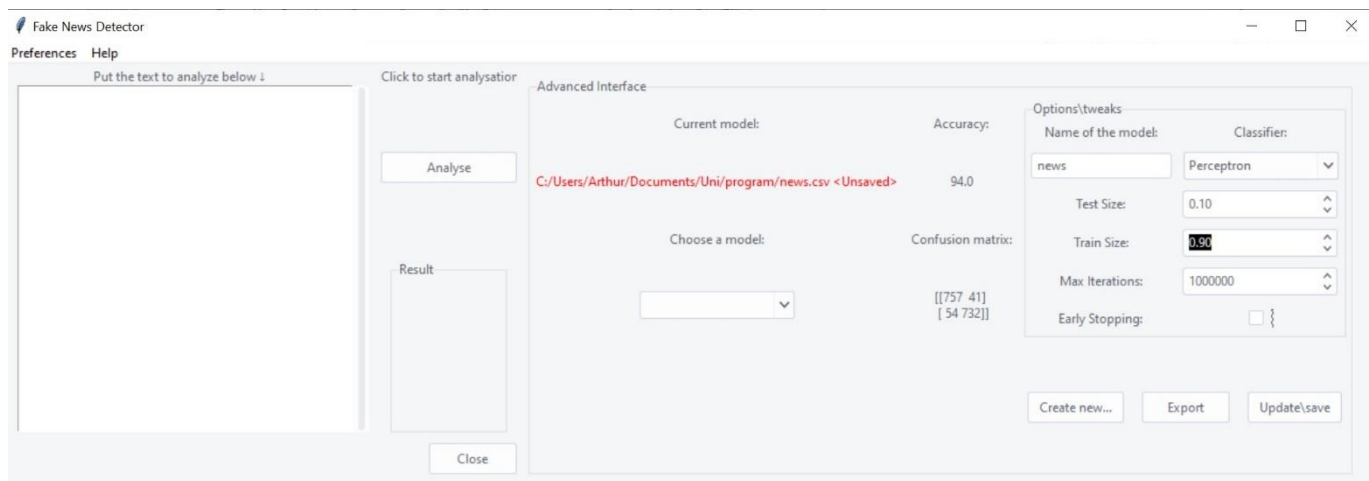


Рисунок 14 – Стан інтерфейсу під час створення моделі

Усі обрані налаштування користувача зберігаються після закриття застосунку. Наприклад, якщо користувач обрав іншу тему та встановив просунутий інтерфейс, то саме у такому вигляді буде запускатися програма.

Висновки до розділу 5

В розділі було наведено керівництво користувача, а саме: описано головні можливості системи, прослідковано процес створення моделі, її навчання та використання на тексті з реального життя. Було перераховано класифікатори, які можливо використовувати для створення моделей, а також зроблено наголос на том, що налаштування користувача зберігаються після закриття застосунку

					ІА82.080БАК.003 ПЗ	Аркуш
						51
Зм.	Арку	№ докум.	Підпис	Дата		

6 ТЕСТУВАННЯ

Цей розділ описує мету здійснюваного тестування, спосіб проведення та результати для перевірки роботи створеного програмного застосунку та відповідність функціоналу заявленим вимогам.

6.1 Мета тестування

У першу чергу, перевірка правильності роботи застосунку дозволяє побачити, чи відповідає застосунок усім заявленим вимогам. Також тестування допомагає знайти області, які потребують подальшої роботи, або невеликої доробки.

6.2 Тестування інтерфейсу

Графічний інтерфейс застосунку є саме тією стороною програмного забезпечення, яку бачать та з якою взаємодіють користувачі, що робить її одним з головних елементів. Інтерфейс має бути зрозумілим, не перевантаженим зайвими функціями та, як найголовніше, – давати чіткий зворотній зв'язок про дії користувача та стан системи в теперішньому часі.

У таблиці 6.1 нижче наведені основні елементи та функції інтерфейсу, очікуваний результат та підсумки проходження тесту.

					ІА82.080БАК.003 ПЗ	Аркуш
						52
Зм.	Арку	№ докум.	Підпис	Дата		

Таблиця 6.1 – Тести інтерфейсу застосунку

Дія	Очікуваний результат	Відгук інтерфейсу	Підсумок
Зберегти модель (кнопка «Update\save»)	Модель з вказаними параметрами зберігається на диск	При спробі зберегти модель з потенційно неправильними параметрами з'являється вікно, яке запитує користувача чи дійсно він впевнений у операції	Тест пройдено
Відкрити застосунок	Застосунок відкривається та стає готовим до роботи	При відсутності створених моделей, з'являється вікно з попередженням, що програма не може знайти жодної моделі та рекомендацією створити модель	Тест пройдено
Завантажити набір даних (кнопка «Create new...»)	Програма завантажує набір даних для навчання	При спробі завантажити файл, який не може бути прочитаний програмою, з'являється вікно з помилкою, що говорить користувачу про неможливість виконання операції та рекомендує, який тип файлу бажано використовувати	Тест пройдено

На рисунках 15-17 наведені сповіщення, які виникають у разі помилки.

					IA82.080BAK.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		53

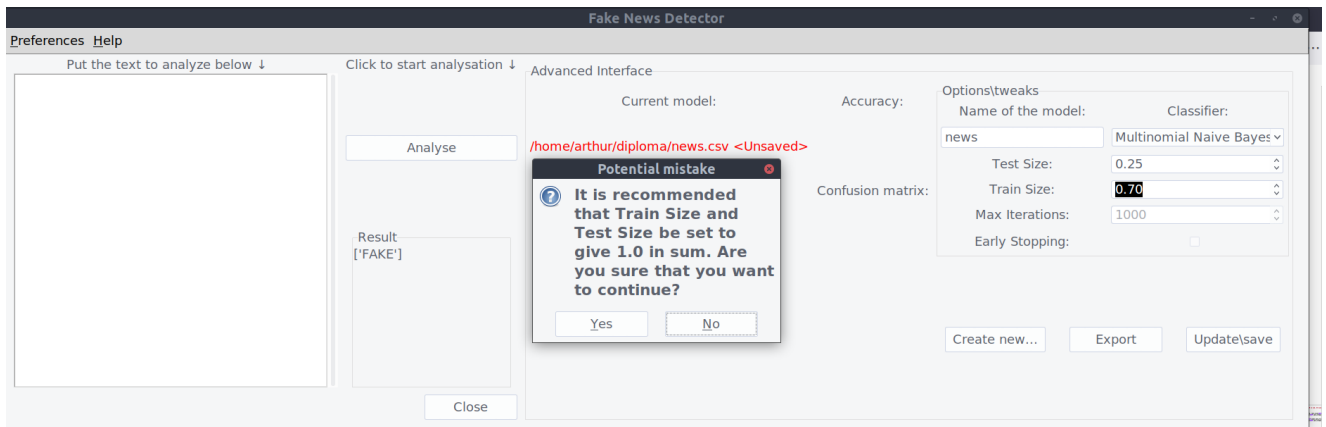


Рисунок 15 – Інформаційне повідомлення

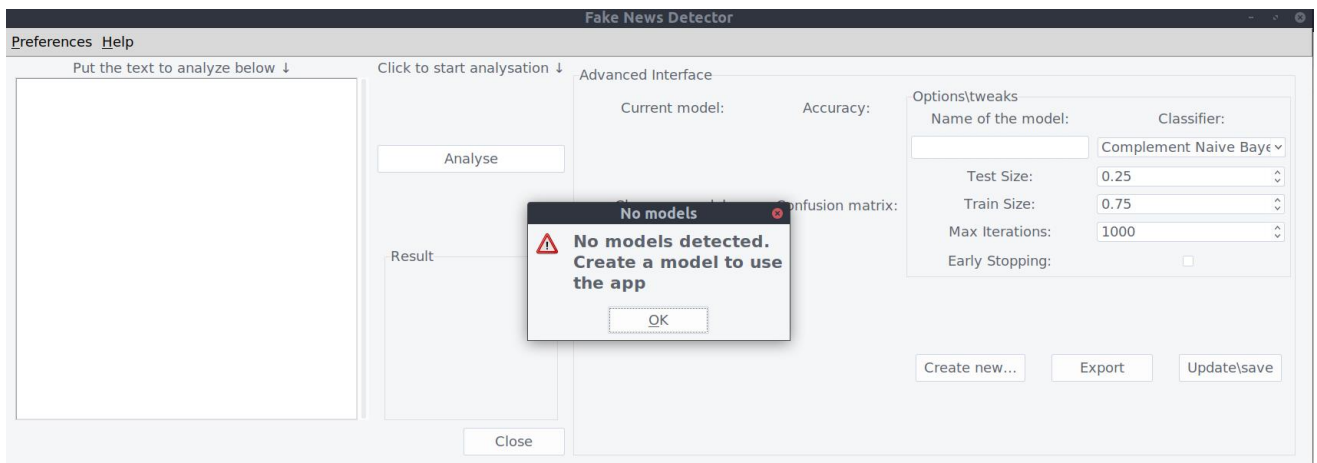


Рисунок 16 – Попередження

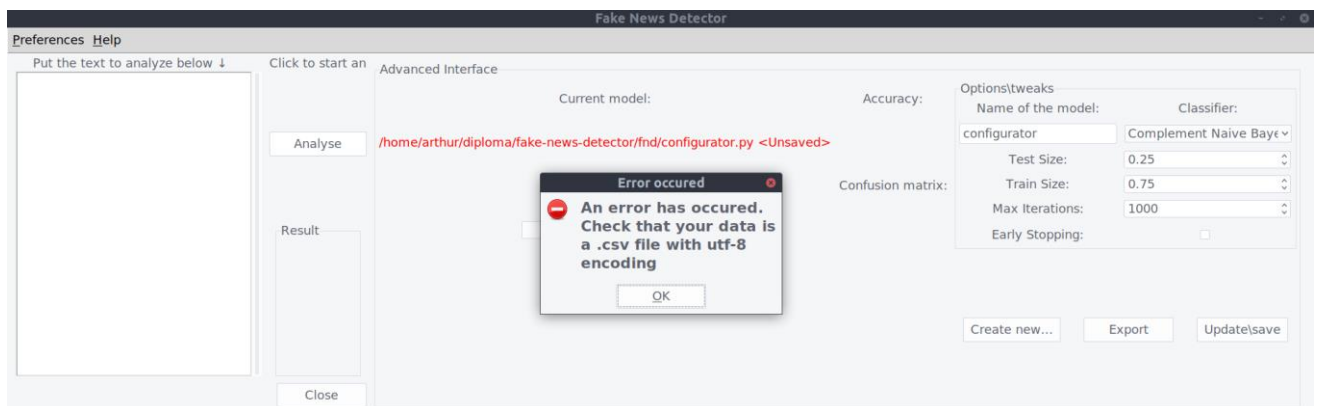


Рисунок 17 – Помилка

					ІА82.080БАК.003 ПЗ	Аркуш
Зм.	Арку	№ докум.	Підпис	Дата		54

6.3 Тестування точності моделі

Для тестування точності створеної базової моделі було прийнято рішення використати її в польових умовах, а саме на вже промаркованих новинах з тієї частини даних від Університету Вікторії (див. розділ 4), що були виключені з набору навчальних даних. Усього було обрано 100 новин з яких 41 – фейкові і 59 – реальні. Результат тестування можна побачити на рисунку 18.

```
PS C:\Users\Arthur> & C:/Users/Arthur/AppData/Local/Programs/Python/Python310/python.exe c:/Users/Arthur/Documents/Uni/program/testing/testing.py
Набір даних "test_news.csv" склав: 100 записів
З них промарковані: 59 -- REAL; 41 -- FAKE
Тестуєма модель: basic
* ----- *
Результат: правильно вгаданих фейкових -- 36, правильно вгаданих реальних -- 40, неправильно вгаданих фейкових -- 5, неправильно вгаданих реальних -- 19
Загальна точність склала: 0.76
```

Рисунок 18 – Тестування точності

Отже, при загальній точності у розмірі 76% можна казати про достатню точність моделі для повсякденного використання. Як згадується у висновках до цієї роботи, використання більшого навчального набору даних з більшою кількістю системних ресурсів здатна призвести до значного покращення результату.

Висновки до розділу 6

У розділі було викладено причини тестування, а також обґрунтовано сфери тестування, на яких варто зосередитися. У графічному застосунку такою сферою видався саме графічний інтерфейс.

					ІА82.080БАК.003 ПЗ	Аркуш
						55
Зм.	Арку	№ докум.	Підпис	Дата		

Графічний інтерфейс був протестован і результати записані і викладені у розділі. Кожна передбачувана ситуація, у якій може опинитися користувач під час використання застосунку супроводжується графічним зворотнім зв'язком у вигляді повідомлення, що ранжується від інформативних до попереджень та помилок.

Створена базова модель була протестована на вибірці даних, що не приймали участі у тренуванні і принесла результат у вигляді 76%, що є очікуваним для використаного тренувального набору даних та доступних обчислювальних ресурсів.

					ІА82.080БАК.003 ПЗ	Аркуш
						56
Зм.	Арку	№ докум.	Підпис	Дата		

ВИСНОВКИ

В ході написання індивідуальної дослідницької роботи було з нуля спроектовано та розроблено програмний застосунок для виявлення фейкових новин за допомогою машинного навчання.

Розробка застосунку почалася з аналізу предметної області, що включило у себе ввід поняття фейкових новин, де було дано визначення поняттю «фейкові новини» обґрунтовано загрозу, яку вони несуть для суспільства та для індивідуумів та надано шляхи вирішення вказаної проблеми. Наступним кроком було проведено огляд підходів до задачі розпізнавання фейкових новин за наукової точки зору. Далі були описані алгоритми класифікації тексту, які існують та широко використовуються в індустрії.

Перед переходом до активної фази розробки застосунку, було проаналізовано існуючі рішення, які усі мали великі недоліки у їх реалізації, що залишило розроблене програмне забезпечення майже без аналогів у сфері розпізнавання фейкових новин.

Звернувши увагу на неповноцінність існуючих рішень та переконавшись у необхідності створення вище зазначеного програмного застосунку, почалася фаза вибору технологій та безпосередньої реалізації. Було обрані алгоритми розроблені sciki-learn, які надали можливість швидкої розробки остаточної системи. При виборі засобів розробки було приділено багато уваги вибору найновітніших технологій та інструментів, щоб задовільнити усі поставлені задачі для створеного програмного забезпечення.

Після розробки ПЗ, було проведено широке тестування на предмет існування невіправлених помилок у роботі, з особливою увагою приділеною саме графічному інтерфейсу, з яким більше всього взаємодіють користувачі.

					IA82.080BAK.003 ПЗ	Аркуш
						57
Зм.	Арку	№ докум.	Підпис	Дата		

Загалом, усі поставлені на меті цілі та завдання були виконані в повному обсязі. В результаті роботи над індивідуальним дослідницьким проектом був побудован застосунок для виявлення фейкових новин.

Навіть звертаючи увагу на те, що фінальний продукт майже не має аналогів у своїй сфері, все ще залишаються моменти, які можуть бути доопрацьовані та покращені:

— однією з найпростіших модифікацій, яка тільки може бути, але у той самий час надавати багато корисного функціоналу є збільшення кількості доступних класифікаторів, що дуже легко зробити, адже потрібні базові класи вже побудовані;

— застосунок є кросплатформеним, тому має працювати на будь-якій операційній системі, але ще був протестований на MacOS, тому тестування та виправлення можливих недоліків спеціально для цієї ОС може зробити програму доступною для більшої кількості людей;

— існують люди з обмеженими можливостями, тому адаптація застосунку до програм зчитування з екрану та підтримка нестандартних способів вводу дозволить використання застосунку цим категоріям користувачів;

— можливо додати більше комбінацій клавіш для швидшого робочого процесу для тих, хто звик використовувати клавіатуру з незначним покладанням на мишу.

					IA82.080BAK.003 ПЗ	Аркуш
						58
Зм.	Арку	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wineburg, Sam and McGrew, Sarah and Breakstone, Joel and Ortega, Teresa. (2016). Evaluating Information: The Cornerstone of Civic Online Reasoning. Stanford Digital Repository. Available at: <http://purl.stanford.edu/fv751yt5934>
2. The Guardian. ‘What is fake news? How to spot it and what you can do to stop it’. URL: <https://www.theguardian.com/media/2016/dec/18/what-is-fake-news-pizzagate>
3. NPR. ‘We tracked down a fake-news creator in the suburbs. Here’s what we learned’. URL: <https://www.npr.org/sections/alltechconsidered/2016/11/23/503146770/npr-finds-the-head-of-a-covert-fake-news-operation-in-the-suburbs>
4. U.S.News. ‘Fake news in reality’. URL: <https://www.usnews.com/opinion/thomas-jefferson-street/articles/2017-04-14/what-is-fake-news-maybe-not-what-you-think>
5. The Guardian. ‘Pizzagate is a lie. But what it says about our society is real’. URL: <https://www.theguardian.com/commentisfree/2016/dec/05/pizzagate-lie-what-it-says-about-society-real>
6. The Guardian. ‘Pizzagate conspiracy’ gunman: ‘I regret how I handled the situation’. URL: <https://www.theguardian.com/us-news/2016/dec/08/pizzagate-conspiracy-gunman-i-regret-how-i-handled-the-situation>
7. Medium. ‘Fake News Detection With Machine Learning Using Python’. URL: <https://towardsdatascience.com/fake-news-detection-with-machine-learning-using-python-3347d9899ad1>
8. Pedregosa et al., JMLR 12, pp 2825-2830, 2011.
9. Lemaréchal, C. (2012). *Cauchy and the Gradient Method. Doc Math Extra*: 251–254).
10. [The Gradient](#), Khan Academy

					IA82.080БАК.003 ПЗ	Аркуш
						59
Зм.	Арку	№ докум.	Підпис	Дата		

11. scikit-learn. 'Classification'. URL: <https://scikit-learn.org/stable/modules/sgd.html#classification>
12. Ruder, S. (2017). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747
13. scikit-learn. 'Kernel Approximation'. URL: https://scikit-learn.org/stable/modules/kernel_approximation.html#id2
14. Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin and Zhi-Hua Zhou (2012). "Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison"
15. Ali Rahimi and Benjamin Recht (2007). "Random features for large-scale kernel machines"
16. JavaPoint. 'Support Vector Machine Algorithm'. URL: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
17. scikit-learn. 'Naïve Bayes'. URL: https://scikit-learn.org/stable/modules/naive_bayes.html
18. Wikipedia. 'Bayes's theorem'. URL: https://en.wikipedia.org/wiki/Bayes%27_theorem
19. OpenGenus. 'Gaussian Naive Bayes'. URL: <https://iq.opengenus.org/gaussian-naive-bayes/>
20. OpenGenus. 'Multinomial Naïve Bayes'. URL: <https://iq.opengenus.org/multinomial-naive-bayes/>
21. GeeksForGeeks. 'Complement Naïve Bayes Algorithm'. URL: <https://www.geeksforgeeks.org/complement-naive-bayes-cnb-algorithm/>
22. OpenGenus. 'Bernoulli Naïve Bayes'. URL: <https://iq.opengenus.org/bernoulli-naive-bayes/>

					IA82.080БАК.003 ПЗ	Аркуш
						60
Зм.	Арку	№ докум.	Підпис	Дата		

23. Medium. 'K-Nearest Neighbor'. URL: <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4>
24. Corporate Finance Institute. 'Ensemble Methods'. URL: <https://corporatefinanceinstitute.com/resources/knowledge/other/ensemble-methods/>
25. Evergreen. 'Ансамблеві моделі машинного навчання'. URL: <https://evergreens.com.ua/ua/articles/ensembles.html>
26. Daily Dot. 'Here are all the 'fake news' sites to watch out for on Facebook'. URL: <https://www.dailydot.com/debug/fake-news-sites-list-facebook/>
27. Facebook. 'About fact checking on Facebook'. URL: <https://www.facebook.com/business/help/2593586717571940>
28. The Guardian. 'Facebook's plan to tackle fake news raises questions over limitations'. URL: <https://www.theguardian.com/technology/2016/dec/16/facebook-fake-news-system-problems-fact-checking>
29. Fake news Debunker. URL: <https://chrome.google.com/webstore/detail/fake-news-debunker-by-inv/mhccpoafgdgbhjhkcmgknndkeenfhe>
30. Verification Plugin. URL: <https://weverify.eu/verification-plugin/>
31. Python. 'The Python Tutorial'. URL: <https://docs.python.org/3/tutorial/index.html>
32. Visual Studio Code. 'Getting Started'. URL: <https://code.visualstudio.com/docs>
33. The Hitchhiker's Guide to Python. 'GUI Applications'. URL: <https://docs.python-guide.org/scenarios/gui/>
34. OpenSource. 'Qt versus Wx: How do two of the most popular Python frameworks compare?'. URL: <https://opensource.com/article/17/4/pyqt-versus-wxpython>
35. ActiveState. 'Python GUI Programming: wxPython vs. tkinter'. URL: <https://www.activestate.com/blog/python-gui-programming-wxpython-vs-tkinter/>

					IA82.080BAK.003 ПЗ	Аркуш
						61
Зм.	Арку	№ докум.	Підпис	Дата		

36. The Hitchhiker's Guide to Python. 'Freezing Your Code'. URL: <https://docs.python-guide.org/shipping/freezing/>
37. scikit-learn. 'User Guide'. URL: <https://scikit-learn.org/stable/>
38. pandas. URL: <https://pandas.pydata.org/>
39. Python. 'Python object serialization'. URL: <https://docs.python.org/3/library/pickle.html>
40. Python. 'Thread-based parallelism'. URL: <https://docs.python.org/3/library/threading.html>
41. University of Victoria. 'Fake News Detection Datasets' URL: <https://www.uvic.ca/ecs/ece/isot/datasets/fake-news/index.php>

					IA82.080БАК.003 ПЗ	Аркуш
						62
Зм.	Арку	№ докум.	Підпис	Дата		

ДОДАТОК А

Код програми

Посилання на репозиторій: <https://github.com/P1car00n/fake-news-detector>

SCAN ME

