Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

КУРСОВА РОБОТА

з дисципліни: «Комп'ютерні технології»
на тему: програмний засіб для перевірки email повідомлень на спам

Виконав студент: II курсу 1ПКТ-23б групи
спеціальності: 122 – Комп'ютерні науки,
Ярослав ДЗЮБЕНКО _____

Керівник: к.т.н., доцент каф. АІІТ
Костянтин ОВЧИННИКОВ

Кількість балів: _____ Оцінка: ECTS _____

Члени комісі: _____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

Вінниця ВНТУ – 2024 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
зав. кафедри АІІТ, проф., д.т.н.
_____ О. В. Бісікало
(підпис)
«13» серпня 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни «Комп'ютерні технології»
студенту Дзюбенко Ярославу Валентиновичу групи 1ПКТ-23б

Розробити програмний засіб для перевірки email повідомлень на спам
Вимоги до розробки:

1. Передбачити використання моделі штучного інтелекту для класифікації повідомлень.
2. Налаштувати програму для перевірки повідомлень з електронної пошти
3. _____

Примітка:

1. Файл курсової роботи підготувати з використанням комп'ютерної типографії LaTeX
2. _____
3. _____

Затверджено на засіданні кафедри 13 серпня 2024 р. протокол №1

Орієнтовний зміст ПЗ до курсової роботи:

Зміст
Анотація
Вступ
Огляд методів вирішення поставленої задачі
Розробка алгоритмічного забезпечення
Розробка програмного забезпечення
Висновки
Список використаних джерел
Додатки

Дата видачі «23» серпня 2024 р. керівник _____
(підпис)

завдання отримав _____
(підпис)

АНОТАЦІЯ

Автор: Дзюбенко Я.В.

Тема: Розробка програмного засобу для класифікації електронних повідомлень на основі нейронних мереж

У даній курсовій роботі досліджується проблема класифікації електронних повідомлень з метою автоматичної ідентифікації спам і важливих (ham) листів. Використовуючи сучасні методи машинного навчання, зокрема нейронні мережі, розроблено програмний засіб, що дозволяє покращити ефективність обробки електронної пошти.

На початку роботи проведено аналіз існуючих підходів до класифікації електронних листів, що включає традиційні алгоритми та новітні архітектури нейронних мереж. Описано процес підготовки даних, вибору моделі, навчання нейронної мережі та оцінки її продуктивності.

Результати тестування показали, що розроблена модель досягає високих показників точності класифікації, проте виявлено обмеження, пов’язані з доступом до текстів повідомлень через політику конфіденційності поштових сервісів. Це вказує на необхідність подальшої роботи в напрямку інтеграції з API поштових клієнтів.

Ключові слова: класифікація, електронна пошта, спам, нейронні мережі, машинне навчання, програмний засіб.

ЗМІСТ

ВСТУП

Наразі, в часи перенасичення інтернету інформацією, спам email повідомленнями є досить поширеною проблемою. Такі повідомлення можуть нести як суто рекламний характер, так і використовуватися для злоякісних хакерських атак. Проблема є досить серйозною, тому майже всі провідні компанії, що надають послуги онлайн-скриньок, вже розробили потужні інструменти для фільтрації спаму. Однак, популярних сторонніх сервісів для фільтрації спаму немає.

Для розв'язання цієї проблеми буде розглянуто варіант моделі штучного інтелекту, що прийматиме на вхід текст і на виході класифікуватиме його як спам або не спам.

Таким чином, спам як комплексна проблема потребує комплексного вирішення, що включає наступні елементи:

- Просвіта (освітня діяльність);
- Організаційна діяльність;
- Технологічні заходи;
- Законодавство.

Мета: розглянути методи і можливості боротьби зі спамом, створити нейронну мережу для розпізнавання спаму, та застосунокобгортку навколо мережі.

Завдання: оцінити ефективність методів боротьби зі спамом, виявити тенденції, які намічаються в методах боротьби зі спамом.

Отже, що ж таке спам. Спам – це, незаконно розповсюджена шляхом масових розсилок, інформація рекламного характеру, отримання якої не узгоджено з користувачем. Іноді можна зустріти написання СПАМ або SPAM, оскільки слово приймають за черговий відомий нам акронім. І це дійсно акронім, за походженням є складноскороченим словом, але не абревіатурою, тому писати його треба «в нижньому регістрі», малими літерами. Це «складне» словосполучення утворилося з усіченого spiced ham. У буквальному перекладі spiced ham - «шинка зі спеціями», але насправді в

консервних банках з написом «Spam», наскільки відомо з історії, перебував ковбасний фарш. В інтернеті слово прижилось і стало позначенням для наглої, безпардонного і нахабної непрошеної реклами.

Дослідження, що будуть проведені в межах курсової роботи, можуть бути використані для вивчення методів класифікації тексту за допомогою штучного інтелекту.

1 АНАЛІЗ СПАМУ ТА ОГЛЯД МЕТОДІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Види спаму

Спам у електронних повідомленнях можна класифікувати на кілька основних типів залежно від змісту та мети повідомлення. Нижче наведено основні види спаму з коротким поясненням кожного.

- **Комерційний спам** — повідомлення, які рекламують товари чи послуги без згоди отримувача. Часто включає посилання на вебсайти, що рекламують продукти, акції або знижки.
- **Фішинговий спам** — повідомлення, які видають себе за офіційні листи від банків, соціальних мереж або інших організацій. Метою є отримання конфіденційної інформації (логінів, паролів, даних кредитних карт) під виглядом авторизованого джерела.
- **Шахрайський спам** — містить обіцянки великих прибутків або виграшів. Наприклад, пропозиції від нібито багатих спадкоємців або лотерей, які вимагають оплатити «адміністративний внесок» для отримання виграшу.
- **Медичний спам** — рекламні повідомлення, що пропонують медичні товари або послуги, наприклад, «чудодійні» ліки або препарати для схуднення. Часто спрямований на вразливі категорії населення.
- **Соціальний спам** — масові розсилки, що використовуються для просування облікових записів у соціальних мережах, або повідомлення, які запрошують до спільнот, не пов’язаних з інтересами отримувача.
- **Спам, що поширює шкідливе ПЗ** — повідомлення, що містять шкідливі файли або посилання, що завантажують віруси, трояни чи інше шкідливе програмне забезпечення на комп’ютер отримувача.
- **Політичний спам** — повідомлення з політичним змістом, які мо-

жуть включати пропаганду або заклики до певних дій, з метою впливу на громадську думку або залучення до політичної діяльності.

– **Ланцюговий спам** — повідомлення, які просять переслати їх іншим людям, часто з обіцянкою щастя, удачі або грошей, або погрозою нещасть, якщо цього не зробити.

Кожен з наведених типів спаму має свою специфіку та вимагає індивідуального підходу для ефективного виявлення та фільтрації.

1.2 Машинне навчання. Класифікація

Машинне навчання (МН) — це підгалузь штучного інтелекту, яка займається створенням алгоритмів та моделей, що дозволяють комп'ютерам навчатися на основі даних без явного програмування для кожної конкретної задачі. У процесі машинного навчання модель аналізує вхідні дані та автоматично знаходить закономірності, які дозволяють прогнозувати або приймати рішення на основі нових даних. Основною ідеєю МН є здатність системи до самоудосконалення на основі накопиченого досвіду та великих обсягів інформації [1]. Машинне навчання (МН) відіграє ключову роль у створенні ефективних алгоритмів для автоматичного виявлення та фільтрації спаму. Мета застосування МН у цьому контексті — навчити модель розпізнавати шаблони, що відрізняють спам від звичайних (легітимних) повідомлень. Використовуючи великі набори даних, що містять як спам, так і легітимні повідомлення, МН-модель може ідентифікувати специфічні характеристики, такі як частота певних слів, загальна структура, наявність посилань або інших елементів, які часто зустрічаються в спам-повідомленнях. Класифікація текстів — це процес автоматичного визначення категорії або класу для заданого текстового фрагмента. Основна мета класифікації тексту — це спрощення роботи з великими обсягами даних, що дозволяє швидко і точно сортувати та структурувати інформацію для подальшого аналізу. Алгоритми

класифікації тексту навчаються розпізнавати специфічні шаблони і особливості у текстах, які допомагають віднести їх до конкретних категорій [2].

1.3 Базові методи машинного навчання для класифікації спаму

Існує декілька базових методів машинного навчання, які широко використовуються для класифікації спаму. До них належать:

- **Наївний байєсівський класифікатор** (*Naive Bayes*) — один із найбільш простих та ефективних методів для класифікації текстів. Він ґрунтується на обчисленні ймовірності того, що певний текст є спамом, на основі частоти окремих слів. Цей алгоритм вважає всі ознаки незалежними одна від одної, що спрощує обчислення та підвищує його продуктивність [3].
- **Метод опорних векторів** (*Support Vector Machines, SVM*) — ще один потужний алгоритм, який використовує гіперплощини для розділення класів. SVM намагається знайти оптимальну гіперплощину, що максимально розділяє спам та легітимні повідомлення. Цей метод добре працює для високорозмірних даних, таких як тексти [4].
- **Логістична регресія** — метод, який використовує логістичну функцію для оцінки ймовірності належності повідомлення до певного класу. Логістична регресія широко застосовується для двокласової класифікації і дозволяє легко інтерпретувати результати у вигляді ймовірностей [5].

Ці базові методи є ефективними та мають невеликі вимоги до обчислювальних ресурсів, що робить їх популярними для задачі класифікації спаму, особливо при обробці великих обсягів текстових даних.

2 РОЗРОБКА І НАЛАШТУВАННЯ МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ

В данному розділі буде описаний алгоритм розробки моделі штучного інтелекту, з описанням усіх пунктів.

2.1 Підбір навчальних данних

Для вирішення задачі класифікації email повідомлень на спам та не спам було використано набір даних, взятий з платформи Kaggle, який є у вільному доступі. Набір даних містив інформацію про текстові повідомлення, включаючи марковані спам і не спам листи. Основними критеріями для вибору цього набору даних стали такі фактори:

- Якість даних
- Обсяг даних
- Актуальність

Перед використанням датасету для навчання нейронної мережі, потрібно попередньо обробити його. З текстових данних потрібно отримати числові вектори, які можна використовувати як вхідні дані для алгоритмів машинного навчання. Для цього буде використовуватись метод CountVectorizer з бібліотеки scikit-learn. Він працює за таким алгоритмом:

а) **Токенізація:** Текст розбивається на окремі слова або токени. Наприклад, речення "The cat sat on the mat"буде перетворено на список слів: [‘‘The’’, ‘‘cat’’, ‘‘sat’’, ‘‘on’’, ‘‘the’’, ‘‘mat’’, ‘‘dog’’, ‘‘log’’].

б) **Створення словника:** Після токенізації `CountVectorizer` створює словник усіх унікальних слів у всіх документах. Наприклад, для двох текстів:

- "The cat sat on the mat"
- "The dog sat on the log"

Словник буде складатися з наступних унікальних слів:
[‘‘The’’, ‘‘cat’’, ‘‘sat’’, ‘‘on’’, ‘‘the’’, ‘‘mat’’, ‘‘dog’’, ‘‘log’’].

в) **Перетворення на вектори:** Для кожного документа створюється вектор, що представляє кількість поява кожного слова зі словника у документі. Наприклад, для двох текстів отримаємо такі вектори:
   - "The cat sat on the mat": `[2, 1, 1, 1, 1, 1, 0, 0]`
   - "The dog sat on the log": `[2, 0, 1, 1, 1, 0, 1, 1]`

   Кожен елемент вектора відповідає кількості поява слова зі словника у відповідному документі.

Цих операцій буде достатньо для подальшої роботи. Тепер потрбіно визначитись з архітектурою нейронної мережі.

2.2 Опис алгоритму машинного навчання: наївний баєсівський класифікатор

Наївний байєсівський класифікатор (Naive Bayes) є одним з найпростіших і водночас потужних алгоритмів машинного навчання, який часто застосовується для класифікації текстових даних, зокрема, для задачі виявлення спаму [6]. Основою цього алгоритму є теорема Байєса, яка дозволяє обчислювати ймовірність належності певного об’єкта до класу за наявними ознаками.

2.2.1 Основні принципи

Основна ідея наївного баєсівського класифікатора полягає у застосуванні теореми Байєса, яка описується рівнянням:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{2.1}$$

де $P(C|X)$ — ймовірність того, що зразок X належить до класу C; $P(X|C)$ — ймовірність спостереження X за умови, що об'єкт належить до класу C; $P(C)$ — апріорна ймовірність класу C; $P(X)$ — ймовірність спостереження X. У випадку текстової класифікації, наприклад для виявлення спаму, кожне повідомлення розглядається як сукупність слів (ознаки), де кожне слово є незалежним, що є головним припущенням "наївності"в алгоритмі. Отже, ймовірність $P(X|C)$ можна розкласти як добуток ймовірностей окремих слів у повідомленні:

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C) \tag{2.2}$$

де x_i — окреме слово в повідомленні, а n — кількість слів у повідомленні.

2.2.2 Апріорні та умовні ймовірності

Наївний байєсівський класифікатор потребує двох типів ймовірностей для кожного класу: апріорної ймовірності $P(C)$ і умовної ймовірності $P(x_i|C)$ для кожного слова x_i. Апріорна ймовірність $P(C)$ зазвичай обчислюється як частка повідомлень, що належать до класу C (спам або не-спам), у всій навчальній вибірці:

$$P(C) = \frac{\text{кількість повідомлень класу } C}{\text{загальна кількість повідомлень}} \tag{2.3}$$

Для оцінки умовних ймовірностей $P(x_i|C)$ найчастіше використовуються обмеження гладкості, такі як згладжування Лапласа, щоб уникнути нульових ймовірностей:

$$P(x_i|C) = \frac{\text{кількість появ } x_i \text{ в } C + 1}{\text{загальна кількість слів у } C + |V|} \tag{2.4}$$

де $|V|$ — розмір словника, тобто загальна кількість унікальних слів у навчальній вибірці.

2.2.3 Класифікація нового зразка

Для класифікації нового зразка X, алгоритм обчислює значення $P(C|X)$ для кожного класу C і вибирає клас із максимальною ймовірністю:

$$\hat{C} = \arg\max_{C} P(C|X) \tag{2.5}$$

Це дає змогу алгоритму призначати кожне повідомлення до одного з класів (наприклад, "спам"або "не-спам"), залежно від того, яка ймовірність є більшою.

2.2.4 Переваги та обмеження

Наївний байєсівський класифікатор має кілька важливих переваг, таких як ефективність у використанні пам’яті, швидке навчання та висока точність для багатьох задач класифікації. Однак основне припущення про незалежність ознак може бути проблематичним у випадках, коли ознаки не є незалежними, що може негативно впливати на точність класифікації [7].

2.2.5 Висновок

Таким чином, наївний байєсівський алгоритм є простим та потужним інструментом для класифікації тексту, зокрема виявлення спаму. Він надає можливість швидкого й ефективного навчання на великих вибірках текстових даних, забезпечуючи надійні результати класифікації.

2.3 Параметри навчання

У моделі **Multinomial Naive Bayes** є декілька ключових параметрів, які можна налаштовувати для поліпшення якості класифікації. Ось основні параметри, які використовуються під час навчання та оптимізації цієї моделі:

а) **alpha (згладжування Лапласа):**
   - **Опис:** Параметр згладжування Лапласа запобігає нульовим ймовірностям для слів, які не з'являються у певному класі під час навчання. Це важливо, оскільки такі нульові значення можуть сильно знижувати точність моделі.
   - **Типово:** $\alpha = 1.0$
   - **Значення:** Можна обирати значення від 0 до 1. Менше значення наближається до реальних даних, а більше згладжує ймовірності сильніше.
   - **Оптимізація:** Найчастіше підбирається за допомогою перехресної перевірки (GridSearchCV або RandomizedSearchCV) для знаходження оптимального значення.

б) **fit_prior**
   - **Опис:** Вказує, чи слід враховувати апріорні ймовірності класів $P(C_k)$. Якщо встановити `fit_prior = False`, модель передбачатиме класи з рівною ймовірністю для всіх.
   - **Типово:** `True`
   - **Оптимізація:** Цей параметр може бути корисним у випадку незбалансованих класів; під час навчання на таких даних часто краще залишити `fit_prior = True`.

в) **class_prior:**
   - **Опис:** Дозволяє встановити власні апріорні ймовірності класів. Якщо цей параметр заданий, `fit_prior` ігнорується.
   - **Типово:** `None` (тобто апріорні ймовірності визначаються з даних)

- **Оптимізація:** Рідко використовується в оптимізації, оскільки апріорні ймовірності зазвичай найкраще визначаються автоматично.

Оптимізацію параметра α можна виконати за допомогою перехресної перевірки з інструментами, такими як `GridSearchCV` або `RandomizedSearchCV`

3 ЗАСТОСУНОК ДЛЯ ВИЯВЛЕННЯ СПАМУ

Тепер перейдемо до практичної частини моєї курсової роботи, а саме створення додатку для виявлення спаму.

3.1 Розробка моделі

Модель розроблялась з використанням мовим програмування Python разом із популярними бібліотеками для обробки даних та машинного навчання. Зокрема, *pandas* використовується для ефективного маніпулювання та аналізу табличних даних [8], *NumPy* забезпечує оптимізовані обчислення з масивами та математичними операціями [9], а *scikit-learn* використовується для реалізації та налаштування алгоритмів машинного навчання [10]. Комбінація цих інструментів дозволяє створювати комплексні та високопродуктивні рішення для задач класифікації, зокрема для виявлення спаму. У додатку А подано лістинг коду для навчання моделі. У наведеному коді створюється створюється конвеєр (*Pipeline*), який складається з двох етапів: векторизація тексту (*CountVectorizer*) для перетворення тексту в числовий формат і застосування *MultinomialNB* для навчання класифікації.

3.2 Впровадження моделі в програмний засіб

Для збереження налаштованої моделі машинного навчання з можливістю подальшого використання була застосована бібліотека *joblib*. Ця бібліотека дозволяє ефективно серіалізувати та десеріалізувати великі об’єкти Python, включно з моделями машинного навчання, що значно спрощує їхнє збереження та завантаження без втрати точності [11]. За допомогою *joblib* налаштована модель була збережена у вигляді файлу, який можна легко імпортувати до програмного засобу з графічним інтерфейсом, розробленого

для зручної взаємодії з користувачем [10]. Для реалізації простого графічного інтерфейсу було використано бібліотеку *Tkinter*, яка є стандартною бібліотекою для створення GUI на Python [12]. Інтерфейс включає текстове поле для введення повідомлень, кнопку для запуску класифікації, а також індикатор, який відображає результат класифікації як спам або не-спам. Застосована модель була підключена до цього інтерфейсу, що дозволяє користувачам у режимі реального часу перевіряти повідомлення на наявність спаму. У додатку Б подано лістинг коду для розробленого графічного інтерфейсу з підключеною моделлю.

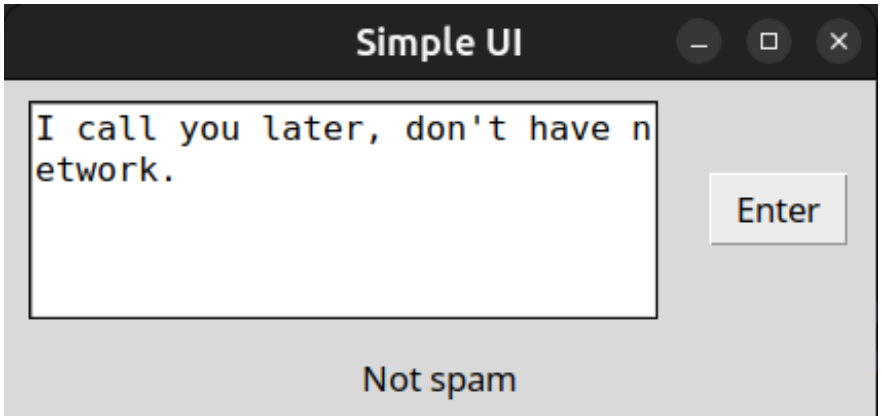

Рисунок 3.1 – Приклад програми з не-спамом

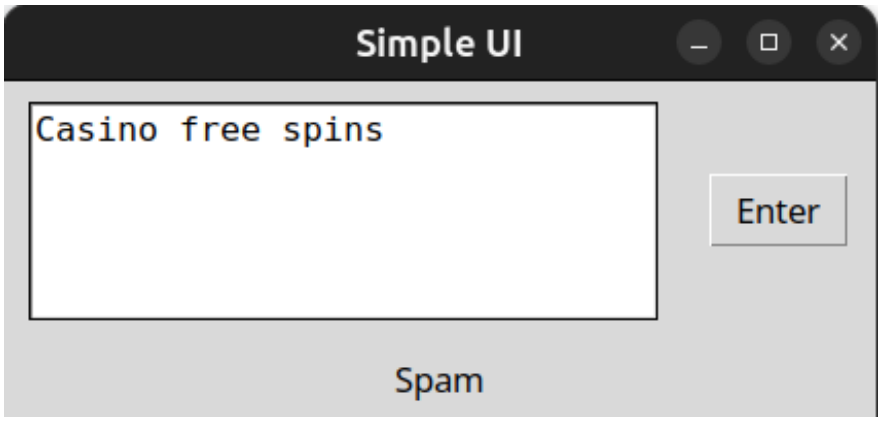

Рисунок 3.2 – Приклад програми зі спамом

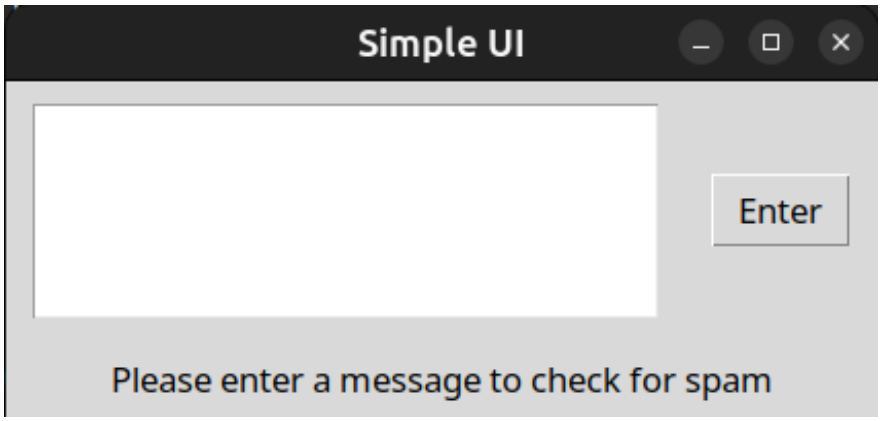

Рисунок 3.3 – Приклад тільки відкритої програми

3.3 Проблеми класифікації email повідомлень

На етапі підключення до програми імпорту даних з електронної пошти були виявлені наступні складнощі. Поштові сервіси не надають можливості вільного завантаження текстів повідомлень через обмеження в API або через політику конфіденційності. Це означає, що програма не може використовуватися для класифікації повідомлень безпосередньо з поштової скриньки. Натомість, її функціональність обмежується класифікацією вже завантажених даних, що знижує ефективність автоматизованого аналізу в реальному часі.

ВИСНОВКИ

У процесі виконання курсової роботи було розроблено програмний засіб для класифікації електронних повідомлень на основі нейронної мережі. Основною метою проекту було створення системи, яка здатна автоматично розпізнавати спам і важливі повідомлення (ham) з високою точністю. Під час роботи були вирішені такі завдання:

- Здійснено аналіз існуючих методів класифікації та обрано оптимальну архітектуру нейронної мережі для поставленого завдання.
- Реалізовано модель класифікації, яка використовує попередньо підготовлені дані, а також виконано її навчання на відповідному наборі даних.
- Проведено тестування моделі, що дозволило досягти прийнятних результатів за ключовими метриками.

Проте в процесі розробки було виявлено низку обмежень. Однією з ключових проблем стало обмеження доступу до тексту повідомлень з поштових скриньок через захисні механізми поштових сервісів, що унеможливлює автоматичну обробку в реальному часі без попереднього завантаження даних. Це вказує на необхідність подальших досліджень у напрямку інтеграції з API поштових клієнтів або альтернативних способів збору даних.

Отже, розроблений програмний засіб може ефективно використовуватися для класифікації завантажених повідомлень, а також може стати основою для подальших удосконалень у сфері автоматизації фільтрації електронної пошти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Bishop Christopher M*. Pattern recognition and machine learning. Springer, 2006.
2. *Manning Christopher D, Raghavan Prabhakar, Schütze Hinrich*. Introduction to Information Retrieval. Cambridge University Press, 2008.
3. *McCallum Andrew, Nigam Kamal*. A comparison of event models for naive bayes text classification // *AAAI-98 workshop on learning for text categorization*. 1998. Vol. 752. P. 41–48.
4. *Joachims Thorsten*. Text categorization with support vector machines: Learning with many relevant features // European conference on machine learning / Springer. 1998. P. 137–142.
5. *Hosmer David W, Lemeshow Stanley, Sturdivant Rodney X*. Applied logistic regression. John Wiley & Sons, 2013.
6. *Russell Stuart, Norvig Peter*. Artificial Intelligence: A Modern Approach. Pearson Education, 2003.
7. *Goodfellow Ian, Bengio Yoshua, Courville Aaron*. Deep Learning. MIT Press, 2016.
8. *McKinney Wes*. Data structures for statistical computing in python. SciPy, 2010.
9. Array programming with NumPy / Charles R Harris, K Jarrod Millman, Stéfan J van der Walt et al. // *Nature*. 2020. Vol. 585, no. 7825. P. 357–362.
10. Scikit-learn: Machine learning in python / Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. // *Journal of machine learning research*. 2011. Vol. 12. P. 2825–2830.
11. *Varoquaux Gael et al.* Joblib: running python functions as pipeline jobs. https://joblib.readthedocs.io. 2023.
12. *Welch Brent B*. Tcl/Tk 8.5 Programming. O'Reilly Media, Inc., 2003.

Додатки

Додаток А

Лістинг коду навчання моделі

model.py

```
import pandas as pd
import numpy as np

data=pd.read_csv('data/spam.csv')
data['Spam']=data['Category'].apply(lambda x:1 if x=='spam'
 ↪  else 0)
data.head(5)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train,
    y_test=train_test_split(data.Message, data.Spam,
 ↪
 ↪  test_size=0.25)

from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

model=Pipeline([
    ('vectorizer', CountVectorizer()),
    ('nb', MultinomialNB())
])

model.fit(X_train, y_train)

import joblib
joblib.dump(model, 'model1.pkl')
```

Додаток Б

Лістинг коду графічного інтерфейсу

main.py

```
import joblib
import tkinter as tk
from tkinter import messagebox

class SimpleUI:
    def __init__(self, root):
        self.model = joblib.load(filename="model1.pkl")

        self.root = root
        self.root.title("Simple UI")

        self.entry = tk.Text(self.root, width=30, height=5)
        self.entry.grid(row=0, column=0, padx=10, pady=10)

        self.enter_button = tk.Button(self.root,
        ↪  text="Enter", command=self.on_enter)
        self.enter_button.grid(row=0, column=1, padx=10,
        ↪  pady=10)

        self.indicator1 = tk.Label(self.root, text="Please
        ↪  enter a message to check for spam", width=40)
        self.indicator1.grid(row=1, column=0, columnspan=2,
        ↪  padx=10, pady=5)


    def on_enter(self):
        user_input = self.entry.get("1.0", tk.END)
        if user_input:

            self.indicator1.config(text="Spam" if
            ↪  self.model.predict([user_input])[0] else "Not
            ↪  spam")
        else:
            messagebox.showwarning("Input Error", "Please
            ↪  enter some text!")


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