Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2024 р.

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «**МЕТОД ФІЛЬТРАЦІЇ СПАМ-ПОВІДОМЛЕНЬ НА ОСНОВІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ**»

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.06.2024 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**
(підпис)

Виконав:
студент 4 курсу, групи КІ – 41
Спеціальність: 123 – «Комп'ютерна інженерія»
ЯКОВЛЕВ Данило В'ячеславович _____

Керівник:
к.т.н., доцент кафедри ЗВО теоретичної та прикладної системотехніки
СТРІЛЕЦЬ Вікторія Євгенівна _____

Рецензент:
к.т.н., доцент ЗВО, в.о. завідувача кафедри теоретичної та прикладної інформатики
МЕНЯЙЛОВ Євген Сергійович _____

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновок, списку використаних джерел і додатків. Загальний обсяг роботи складає 51 сторінку, із яких 37 сторінки основної частини, 3 рисунка, 3 таблиці та 14 найменувань використаних джерел та три додатки.

Метою кваліфікаційної роботи є підвищення точності моделей виявлення спам-повідомлень та мінімізація хибних позитивний результатів, що є критично важливим для підтримки ефективності сучасної комунікації у інформаційному середовищі

Об’єкт дослідження – процес виявлення спам повідомлень серед текстових даних.

Предмет дослідження – моделі і методи виявлення і фільтрації спам повідомлень.

Область застосування – класифікація спам-повідомлень систем електронних листів.

Ключові слова: спам, фільтрація спаму, машинне навчання, алгоритм наївного Баєйса, методи класифікації.

ABSTRACT

The explanatory note for the bachelor's qualification work consists of an introduction, three chapters, a conclusion, a list of references, and appendices. The total length of the work is 51 pages, of which 37 pages make up the main part, 3 figures,3 tables, 14 references, and four appendices.

The goal of the qualification work is to ensure high accuracy of spam detection models and minimize false positive results, which is critically important for maintaining the efficiency of modern communication in the information environment.

The object of the research is the process of detecting spam messages among text data.

The subject of the research is the models and methods for detecting and filtering spam messages.

The area of application is the classification of spam messages in email systems.

Keywords: Spam, spam filtering, machine learning, Naive Bayes algorithm, classification methods.

ЗМІСТ

ВСТУП

У сучасному цифровому просторі спам-повідомлення стали серйозною загрозою для ефективності та безпеки електронної комунікації. Кожного року кількість спаму невпинно зростає, приводячи до значної шкоди як великим організаціям, так і окремим користувачам. Традиційні методи боротьби зі спамом, такі як фільтрація за ключовими словами, використання чорних або сірих списків часто призводять до малої ефективності у сучасному світі у зв’язку з постійним винаходом нових способів розповсюдження спаму. Через це виникає необхідність впровадження нових, інтелектуальніших методів, які мають можливість адаптуватись до постійних спроб їх обходів. Вони повинні забезпечувати високий рівень захисту.

Кожен день зловмисники намагаються удосконалити свої програми для спам розсилки, знайти нові методи розповсюдження спаму та фішингу тож виникає питання: Яким чином боротись з цією шкодою та запобігати її?

Одним із перспективних напрямів для протидії спаму є використання методів машинного навчання для класифікації спам-повідомлень. Завдяки машинному навчанню вдається створювати моделі, які мають змогу автоматично виявляти спам з високою точністю, адаптуючись до нових типів загроз водночас знижуючи випадки помилково позитивних опрацювань. Застосування алгоритмів машинного навчання при вирішенні такої глобальної задачі має на меті підвищення ефективності боротьби з небажаними повідомленнями, які можуть завдавати непоправної шкоди усьому інформаційному середовищу.

Кваліфікаційна робота присвячена розробці та дослідженню методів фільтрації спам повідомлень на основі алгоритмів машинного навчання. У даній роботі будуть розглянуті різні підходи до побудови моделей машинного навчання, проаналізовані їхні переваги та недоліки, а також проведені експерименти з використанням реальної вибірки даних для оцінки ефективності запропонованих методів.

Об’єкт дослідження – процес виявлення спам повідомлень серед текстових даних.

Предмет дослідження – моделі і методи виявлення і фільтрації спам повідомлень.

Метою даної кваліфікаційної роботи є забезпечення високої точності моделей виявлення спам-повідомлень та мінімізація хибних позитивний результатів, що є критично важливим для підтримки ефективності сучасної комунікації у інформаційному середовищі

Для досягнення мети необхідно виконати:

1) аналіз сучасних засобів і методів фільтрації спам-повідомлень.

2) розробку методу фільтрації спам-повідомлень на основі алгоритмів машинного навчання.

3) програмну реалізацію методу фільтрації спам-повідомлень на основі алгоритмів машинного навчання та її тестування.

Актуальність теми "Метод фільтрації спам повідомлень на основі алгоритмів машинного навчання" обумовлена швидким зростанням кількості спам-повідомлень та їх постійним ускладненням. Алгоритми машинного навчання пропонують потужні інструменти для побудови систем фільтрації, здатних адаптуватись та автоматично навчатись на основі великої кількості вже існуючих даних, а також удосконалювати свої моделі класифікації спам-повідомлень. Використання алгоритмів машинного навчання дозволяє значно підвищити точність та надійність фільтрації, знижуючи кількість хибних позитивних класифікацій а також забезпечують краще виявлення нових, раніше невідомих типів спаму.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ФІЛЬТРАЦІЇ СПАМ ПОВІДОМЛЕНЬ

Спам [14] – це небажані повідомлення у будь-якій формі, які надсилаються у великій кількості. Зазвичай зі словом спам асоціюють листи, надіслані на електронну пошту, проте сьогодні спам може з'явитись у вигляді повідомлень у соціальних мережах, коментарів на форумах, повідомлень у месенджерах, текстових повідомлень (SMS) тощо.

Метою спаму може бути реклама товарів або послуг, фішинг (отримання конфіденційної інформації), розповсюдження шкідливого програмного забезпечення або просто створення інформаційного шуму.

Термін «спам» став використовуватися з 1993 року, коли рекламні компанії почали публікувати в групах новин Usenet, дискусійних листах і гостьових книгах повідомлення, які не відповідали тематиці або були відвертою рекламою.

1.1 Походження терміну «Спам»

Світ вперше дізнався про «спам» 5 липня 1937 року, коли американська компанія Hormel Foods Corporation презентувала консервовану свинину під брендом Spam (читається як «спем»). Назву для нового продукту запропонував Кен Дайно, брат тодішнього голови Hormel Foods Corporation Джея Сі Хормела. Розшифровка цього слова, за словами Хормела, відома лише вузькому колу керівників Hormel Foods, але вважається, що Spam (SPAM) — це скорочення від «шинка зі спеціями» (spiced ham), основного інгредієнта цих консервів.

Коли продукт вийшов на ринок, компанія приділила велику увагу рекламній компанії. У ній дуже часто згадувалось слово SPAM. Після початку другої світової війни та вступу до неї США про «спем» почув увесь світ. Свинячі консерви Spam стали частиною раціону не тільки американських солдат, а й її союзників. У регіонах, де проводились бойові дії, консервована свинина стала

незамінним продуктом, але після закінчення війни, великі запаси Spam та прагнення компанії позбутися їх, зіграло з ними злий жарт.

Компанія розгорнула вкрай агресивну рекламну кампанію для просування нового бренду, використовуючи всі доступні засоби. Завдяки цьому, у 1959 році було продано мільярдну банку Spam, а через 10 років консерви почали асоціюватися з настирливою рекламою.

У 1970 році було випущено скетч під такою ж назвою Spam британською комік-групою «Монті Пайтон». Сюжет цього скетчу доволі тривіальний: головні герої потрапляють до кафе, де кожна позиція з меню містись слово «Спем», а відвідувачі у свою чергу співають рекламну пісню продукту Spam. Навіть через 15 років після виходу цього скетчу, він залишився у пам’яті багатьох глядачів.

Першою асоціацією набридливих повідомлень зі словом «Spam» відбулося лише у 1986 році, коли, в популярній у ті часи комп’ютерній мережі Usenet, коли користувач Дейв Родес наполегливо рекламував фінансову піраміду, публікуючи один й той самий текст з інструкцією збагачення на різних форумах. Таким чином ці повідомлення досить швидко набридли користувачам. Вони побачили аналогію між цими повідомленнями та популярним скетчем і почали називати усю настирливу рекламу назвою на честь консервів «Spam».

Пізніше, коли слово спам остаточно закріпилось за агресивною рекламою, він вже встиг перетворитись на спосіб заробітку як чесним, так і відверто шахрайським шляхом. Компанія Hormel Foods неодноразово намагалась заборонити використовувати назву її марки як слово для набридливих повідомленнь у судовому порядку, проте у них нічого не вийшло. У результаті, слово спам у наші дні асоціюється не з м’ясними консервами, котрі випускаються досі у різних варіантах, а з набридливою рекламою, повідомленнями, дзвінками, тощо.

1.2 Види спаму

Розрізняють кілька видів спаму, який користувачі можуть отримувати.

1. Реклама. Найпоширеніший вид спаму, який містить рекламні повідомлення про різні товари або послуги. Ці повідомлення часто надсилаються без згоди отримувачів і можуть бути сумнівної якості або походження.

2. Фішинг. Спам, спрямований на викрадення особистої інформації, такої як паролі, номери кредитних карток або інші конфіденційні дані. Фішингові повідомлення часто маскуються під офіційні листи від банків, платіжних систем або інших авторитетних організацій.

3. Шкідливе програмне забезпечення (Malware). Спам, який містить посилання або вкладення з шкідливим програмним забезпеченням, наприклад, вірусами, троянами або шпигунськими програмами. Відкриття таких повідомлень може призвести до зараження комп'ютера та втрати даних.

4. «Листи щастя» та шахрайські схеми. Спам, що містить пропозиції взяти участь у фінансових пірамідах, лотереях, "листи щастя" або інші шахрайські схеми, які обіцяють швидке збагачення. Метою таких повідомлень є виманювання грошей або особистих даних у довірливих користувачів.

5. Політичний спам. Спам, який містить політичну пропаганду або заклики до участі у політичних заходах. Такі повідомлення можуть бути спрямовані на вплив на громадську думку або підтримку певних кандидатів чи партій.

1.3 Основні риси спам-повідомлення

Не дивлячись на різні типи і напрямки спаму, виділяють характерні риси, властиві усім спам-повідомленням:

1. Масовість. Спам надсилається великій кількості людей. Це можуть бути сотні, тисячі або навіть мільйони адресатів. Такий обсяг розсилок дозволяє спамерам досягати широкої аудиторії з мінімальними витратами.

2. Небажаність. Одержувачі не дали згоди на отримання цих повідомлень. Спам є небажаним вторгненням у персональний простір користувачів, що часто викликає роздратування. Відсутність згоди від одержувачів робить спам

незаконним у багатьох країнах, де діють закони про захист персональних даних та електронні комунікації.

3. Рекламний або шахрайський характер. Спам часто містить рекламу, шахрайські пропозиції або посилання на шкідливі веб-сайти. Рекламний спам може пропонувати різні товари та послуги, часто сумнівної якості або походження. Шахрайський спам може містити фішингові повідомлення, метою яких є викрадення особистих даних, таких як паролі або номери кредитних карток. Шкідливий спам може включати посилання на веб-сайти, які заражають комп'ютери вірусами та іншими видами шкідливого програмного забезпечення.

1.4 Способи поширення спам-повідомлень

Розрізняють також і способи розповсюдження спаму. Серед них є відомими:

1. Електронна пошта (E-mail спам). Зловмисники використовують спеціальні програми, котрі допомагають надсилати листи на велику кількість адресів одночасно. Щоб уникнути блокування, використовуються тимчасові електронні адреси, або адреси, котрі були вкрадені.

2. Соціальні мережі. У соціальних мережах, спам повідомлення можуть надсилатись у особисті повідомлення, у коментарях. Для цього методу використовують підроблені акаунти.

3. Текстові повідомлення (SMS спам). Маючи ваш номер телефона, вам можуть надходити спам повідомлення у вигляді SMS. Іноді одержувач повідомлення повинен платити за його відправу.

4. Телефонні дзвінки. Деякі компанії можуть масово здійснювати велику кількість телефонних дзвінків. Для цього методу не обов’язково мати ваш номер телефону, розсилка здійснюється за допомогою перебору телефонних номерів, однак для поширення ефективності частіше використовують готові бази даних з номерами абонентів конкретного оператора.

5. Спливаючі вікна. За допомогою вірусних програм, зловмисники можуть виводити вам на екран вікна з рекламою. Частіше спливаючі вікна зустрічаються на веб-сайтах і деякі з них не мають можливість бути закритими

6. Мережеві ігри. Чати в онлайн-іграх стають місцем для спаму, де спамери розсилають не лише рекламні повідомлення, але і шахрайські пропозиції. Користувачі можуть зустріти надокучливі рекламні оголошення, які відволікають увагу від гри, а також спроби обману або шахрайства через запропоновані у чатах продукти чи послуги.

1.5 Наслідки спаму

Спам може призвести до певних проблем або неприємностей. Наприклад:

1. Втрата часу та ресурсів. Спам забирає час і ресурси як у користувачів, які змушені видаляти небажані повідомлення, так і у компаній, які витрачають значні кошти на захист від спаму та боротьбу з ним.

2. Зниження продуктивності. Велика кількість спаму може суттєво знизити продуктивність працівників, особливо якщо вони отримують спам на робочі електронні пошти.

3. Особисті фінансові втрати і проблеми, які можуть отримати довірливі або необізнані про спам користувачі.

1.6 Методи боротьби зі спамом

Якщо розглядати спам-листи на електронну пошту, то найнадійніший метод боротьби – не дозволяти зловмисникам отримати вашу електронну адресу. Як виявилось, ця задача не з простих, проте є декілька запобіжних заходів, які підвищать шанси:

- не публікувати електронну адресу на веб-сайтах чи в групах Usenet без необхідності;

- не використовувати власну адресу електронної пошти для реєстрації на підозрілих сайтах. Якщо цей сайт корисний для користувача – використовувати іншу електрону пошту, спеціально створену для цього;
- ні в якому разі не відповідати на спам і не переходити за посиланнями у спамі. Таким чином користувач дає зрозуміти, що використовує цю електронну пошту і дає зловмисникам «зелений» сигнал відправляти ще більше спаму;
- при створенні електронної пошти, використовувати довге та незручне ім’я. Таким чином ускладняється вгадування імені для розсилки спаму.

Для фільтрації спаму існує спеціальне програмне забезпечення, яке застосовується для виявлення небажаних повідомлень. Воно може використовуватись кінцевим користувачем, або на серверах (рис. 1.1) [1]. Таке ПЗ розподіляють на два підходи:

1. Перший підхід полягає в аналізі самого повідомлення і подальшої класифікації спам це чи ні. У такому випадку, якщо електронний лист класифіковано як спам, то він може позначатися міткою, переміщуватись у окрему папку, або навіть видалятись. Таке ПЗ може працювати як на сервері, так і на комп’ютері користувача. При використанні даного підходу, коритувач може не бачити відфільтрованого спаму, проте буде продовжувати платити гроші за використання та фільтрацію.

2. Другий підхід полягає в аналізі не повідомлення, а користувача, чи вважається даний користувач, який надсилає листи, спамером. Для такого визначення використовуються різні методи та підходи. Таке ПЗ може працювати виключно на сервері, де здійснюється прийом електронного листа. При цьому підході вдається зменшити витрати – серверу треба тільки перевірити пошту відправника та відмовити приймати листа. Однак перевага такого підходу не велика, після відмови приймати листа, спамерська програма буде намагатися обійти захист і відправляти його іншим способом. Через це кількість перевірки пошти відправника зростатиме.

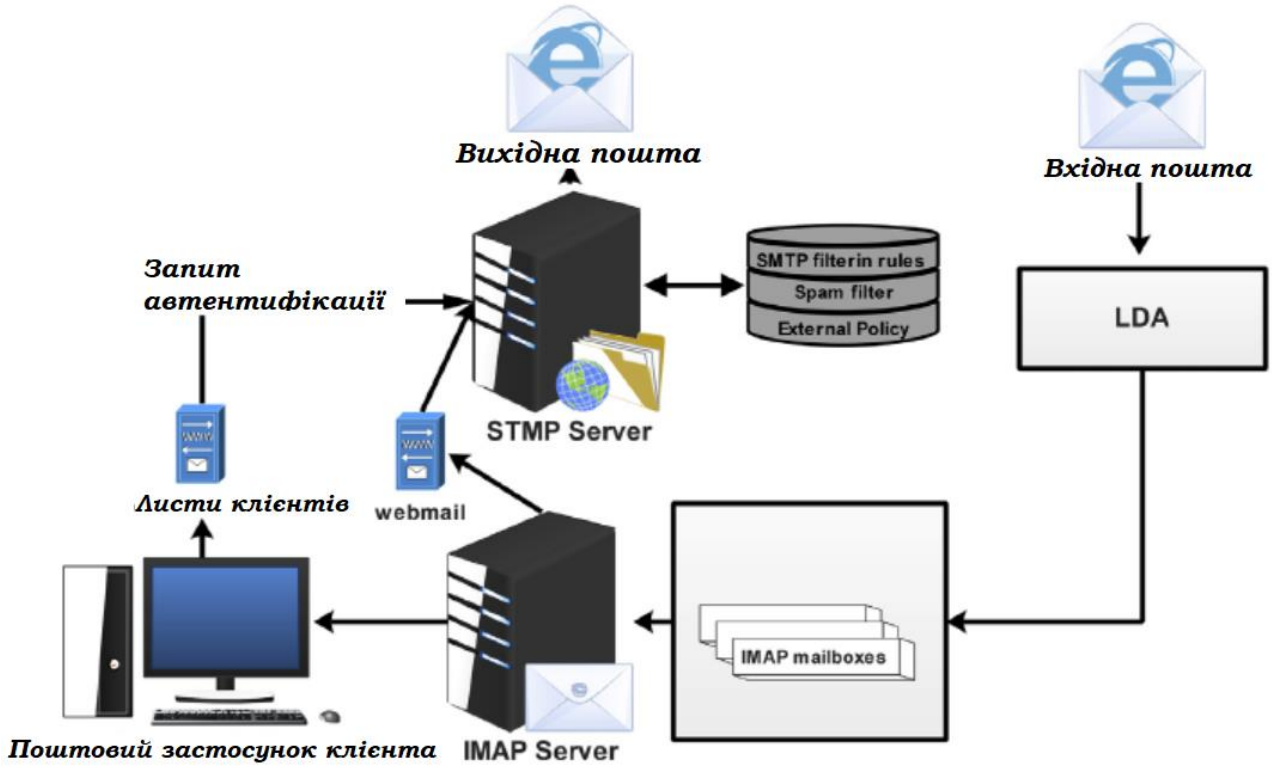

Рисунок 1.1 – Архітектура фільтрації спаму сервера електронної пошти

Спам-фільтри розгортаються багатьма постачальниками Інтернет-послуг на кожному рівні мережі, перед сервером електронної пошти або на поштовому ретрансляторі, де присутній брандмауер. Брандмауер — це система безпеки мережі, яка відстежує та керує вхідним і вихідним мережевим трафіком на основі заздалегідь визначених правил безпеки. Сервер електронної пошти служить вбудованим рішенням для захисту від спаму та вірусів, що забезпечує комплексну безпеку електронної пошти на периметрі мережі. Фільтри можуть бути реалізовані в клієнтах, де їх можна монтувати як надбудови на комп’ютерах, щоб служити посередником між деякими кінцевими пристроями [1].

Фільтри блокують небажані або підозрілі електронні листи, які становлять загрозу безпеці мережі, від потрапляння до комп’ютерної системи. Крім того, на рівні електронної пошти користувач може мати налаштований спам-фільтр, який блокуватиме спам-лист відповідно до певних умов.

Кожен підхід має сенс на існування, однак у наш час найбільш поширенішим є перший, через можливість навчання комп'ютера самостійно класифікувати повідомлення.

1.7 Технічні методи боротьби зі спамом

Сьогодні існує декілька основних способів боротьби зі спамерами та їх спробами надсилати небажані, або навіть фішингові листи. У таблиці 1.1 розглянуто такі способи та проаналізовані їх переваги та недоліки.

Таблиця 1.1

Технічні методи боротьби зі спамом

ВИД	ПЕРЕВАГИ	НЕДОЛІКИ
Спам-фільтри	Можливість самостійного налаштування та визначення ключових слів для ідентифікації спам-повідомлень.	Постійне оновлення програм для відправлення спаму та зміна слів на слова синоніми для обходу фільтрів.
Чорні списки	100% блокування повідомлень з підозрілих або небажаних IP-адрес.	Можливість помилкової ідентифікації законослухняних користувачів як зловмисників та заборона на відправку буд-яких повідомлень цим користувачам
Сірі списки	Можливість попередньої перевірки листів, відправлених з невідомих серверів з наступним додаванням його в білий	Затримка при доставці першого листа. Вона може досягати 30 хвилин, а іноді – більше.

	список, якщо це не спам, або чорний – якщо цей лист належить до спаму.	
Виклик-відповідь	Перед відправкою повідомлення одразу дають можливість перевірити чи відправляє цей лист реальна людина.	Створюють додаткове навантаження на сервери. Неможливість відрізняти ботів, котрі надсилають спам від ботів, котрі надсилають новини тощо.

1.8 Процес фільтрації спам-повідомлень на основі алгоритмів машинного навчання

Не зважаючи на усі існуючі методи боротьби зі спамом, будь то зі сторони користувача або сервера, існуючі методи, у сучасному світі, працюють не так ефективно. Ці підходи тільки дозволять на незначний відсоток уникнути спам-повідомлень. Через це виникає питання: як можна боротись зі спамом ефективніше? Відповіддю на це питання є використання нових технологій, підходів тощо.

Одним з підходів до вирішення цієї проблеми стали моделі машинного навчання. Вони можуть навчатись на вже існуючих у великій кількості спам-повідомлень, та ще більшої кількості легітимних повідомленнях, створюючи ефективну модель котра за короткий час може класифікувати нові повідомлення, котрі вона не бачила раніше. Такий підхід є дуже ефективним, тому що усі повідомлення, котрі надходять, можуть бути використанні для навчання та удосконалення ускладнюючи зловмисникам процес обходу його.

На рисунку 1.2 зображено послідовність дій при отриманні нового повідомлення, яка використовує моделі машинного навчання для фільтрації спам-повідомлень.

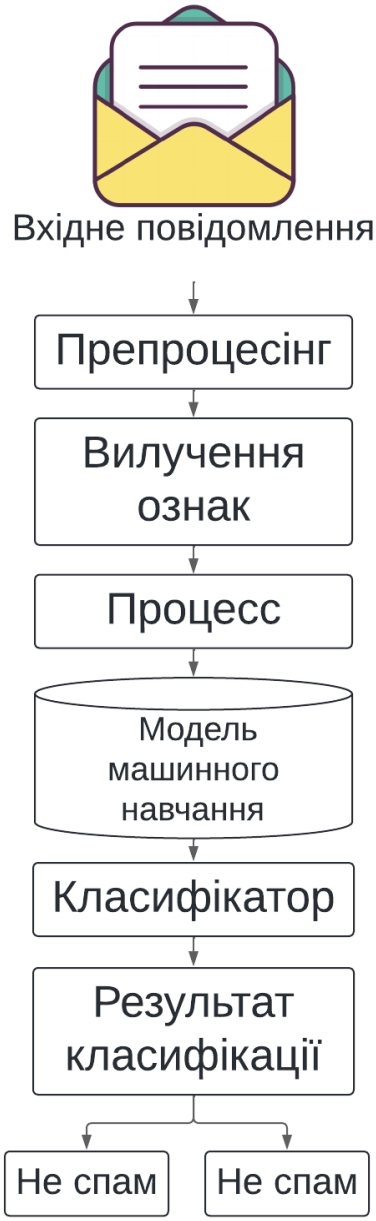

Рисунок 1.2 – Алгоритм класифікації спам-повідомлення за допомогою машинного навчання

Висновки за розділом 1

За результатами аналізу, проведеному у розділі 1, можна сказати, що існуючі методи фільтрації спам-повідомлень мають довели свою корисність, проте постійна еволюція спам-програм та обхід усіх цих методів не залишає вибору окрім створювати все досконаліші методи боротьби із спамом.

Ефективна боротьба зі спамом потребує комплексного підходу, що включає адміністративні заходи, підвищення обізнаності користувачів та використання нових технологічних методів. Сучасні технічні рішення, зокрема алгоритми машинного навчання, показують високу ефективність у виявлення та запобіганні спаму. Постійний прогрес цих методів є ключовим для забезпечення безпеки та надійності електронної комунікації в умовах постійної еволюції загроз.

РОЗДІЛ 2
МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ СПАМУ

Машинне навчання [10] – галузь штучного інтелекту, яка зосереджена на розробці та вивченні статистичних алгоритмів, здатних навчатися на відомих даних і узагальнювати нові, незнайомі дані, й відтак виконувати завдання без явних інструкцій.

Підходи машинного навчання використовуються в багатьох галузях. Їх впровадження значно підвищує ефективність процесів, точність прогнозування та якість ухвалення рішень.

Машинне навчання відіграє ключову роль у фільтрації спам-повідомлень, дозволяючи створювати системи, які автоматично ідентифікують та видаляють небажані повідомлення. Такі системи можуть використовувати різні методи та алгоритми для аналізу тексту, виявлення закономірностей та ухвалення рішень чи є повідомлення спамом, або воно легітимне.

2.1 Задача фільтрації спаму як задача машинного навчання

Перед побудовою моделі класифікації спам-повідомлень треба спочатку зрозуміти, як вирішити цю задачу за допомогою машинного навчання. Об’єкт, який необхідно класифікувати – повідомлення, тобто це звичайні строкові змінні у мовах програмування.

На початку вирішення розглядуваної задачі одразу стикаємось з тим, що більшість алгоритмів машинного навчання працюють та можуть класифікувати тільки числові значення. Якщо використовуються строкові дані, вони повинні бути представлені у числовому вигляді, наприклад, можна перетворити повідомлення у вектори чисел (вектори ознак), а потім вже класифікувати ці вектори. Таким чином можна представити повідомлення як вектор частоти або використовувати інші методи векторизації початкових даних.

При вилучені ознак, зазвичай губиться інформація і треба обережно підходити до вибору яким саме чином ми будемо це робити. Якщо ознаки будуть обрані таким чином, що для спам-повідомлення та легітимного повідомлення

вони можуть бути однаковими, тоді звісно, що модель буде допускати помилки при класифікації повідомлення. Однак, якщо зробити класифікація більш тривіальною, тобто встановити основну ознаку – бути повідомленню спамом або ні, тоді це поліпшить та полегшить класифікацію.

2.2 Підходи машинного навчання

Підходи машинного навчання прийнято розділяти на три великі категорії, які відповідають тій чи іншій парадигмі навчання, залежно від природи «сигналу» або «зворотного зв’язку»:

1. Кероване навчання (навчання з вчителем, supervised learning) [8] – парадигма машинного навчання, в якій модель тренують об'єкти входу (наприклад, вектор змінних-передбачувачів) та бажане значення виходу (також відоме як мічений людиною керівний сигнал, англ. supervisory signal).

2. Некероване навчання (навчання без вчителя, unsupervised learning) [12] – парадигма машинного навчання, в якому, на відміну від керованого або напівкерованого, алгоритми навчаються образів виключно з немічених даних.

3. Напівкероване навчання (навчання з підкріпленням, reinforcement learning) [11] – парадигма машинного навчання, в якій використовується невелика кількість даних з мітками разом з великою вибіркою даних без міток.

2.3 Методи керованого навчання

При виборі керованого навчання розкривається широкий спектр алгоритмів, кожен із яких має свої сильні та слабкі сторони. Єдиного алгоритму навчання, який працює краще з усіма видами задач не існує. При вирішенні задачі фільтрації спам повідомлень існують наступні алгоритми керованого навчання:

2.3.1 Наївний Байєсівський метод

Це один з простих та ефективних алгоритмів класифікації, заснований на теоремі Байєса з припущенням незалежності між ознаками. Завдяки його

простоті у задачах з обробки тексту, наприклад такі як спам-фільтри, він широко використовується на практиці.

В основі Наївного байєсівського методу лежить теорема Байєса – визначення ймовірності події, спираючись на обставини, які могли бути пов'язані з цією подією, з припущенням, що ознаки незалежні.

Теорема Байєса описується рівнянням:

$$\mathrm{P}(A \mid x) = \frac{P(x \mid A)\, P(A)}{P(x)},$$

де $\mathrm{P}(A \mid x)$ – апостеріорна ймовірність класу – умовна ймовірність, ймовірність події А за умови даних *x*;

$P(x \mid A)$ – ймовірність спостереження даних *x* за умови істинності А;

P(A) – ймовірність події А;

P(B) – загальна ймовірність даних *x*.

На практиці використовують тільки чисельник, оскільки знаменник не залежить від А та значення ознак. Через це модель можна виразити за допомогою формули:

$$P(x \mid C_k) = P(x_1, x_2, x_3, \dots, x_n \mid C_k) = \prod_{i=1}^{n} P(x_i \mid C_k),$$

де *x* – вектор спостереження ознак, $\prod_{i=1}^{n}$ - добуток усіх умовних ймовірностей $P(x_i \mid C_k)$ для кожної окремої ознаки x_i.

У залежності від того, який закон розподілу випадкових величин використовується в алгоритмі, розроблено кілька варіантів наївного байєсівського методу:

– **Гаусівський наївний Байєс.** Цей різновид наївного Байєсівського методу, який припускає, що ознаки у векторі спостережень мають нормальний розподіл. Хоча він і передбачає незалежність ознак, однак гаусовський наївний байєсівський класифікатор може використовувати коваріаційну матрицю для моделювання залежності між ознаками.

– **Поліноміальний наївний Баєйс**. Реалізує наївний Байєсівський алгоритм та є одним з алгоритмів класифікації, котрий використовується для задач класифікації тексту. Така модель ефективно працює з даними, котрі представлені як вектор частоти або кількості, наприклад слів.

– **Наївний байєсівський метод на основі розподілу Бернуллі**. Метод класифікації даних на основі наївного Байєсівського алгоритму. Для цієї моделі потребуються дані, які розподіляються відповідно до багатовимірних розподілів Бернуллі, тобто ознаки повинні бути представлені у двійковому вигляді.

– **Категоріальний наївний Баєйс.** Різновид наївного Байєсівського методу, котра ефективно працює з категоріальними ознаками, тобто з ознаками, котрі можуть приймати одне з декількох можливих значень. Числове або частотне представлення ознак у даному випадку не є обов'язковим.

2.3.2 Лінійна регресія

Регресія [9] – метод визначення зв'язку однієї змінної (y) за іншими змінними (x). У машинному навчанні лінійна регресія використовується як контрольований алгоритм машинного навчання.

Загальна формула лінійної регресії має вигляд:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_K + \varepsilon,$$

де Y – залежна змінна, яку намагаються передбачити;

$\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ – ваги, які множаться на відповідні предиктори $X_1, X_2, \ldots, X_K$. Ваги предикторів визначають, на скільки кожен окремий предиктор впливає на залежну змінну;

ε – нормально розподілена випадкова величина (розбіжність між передбаченими значеннями та фактичними).

Загальний вигляд моделі лінійної регресії:

$$\mathrm{a}(x) = \omega_0 + \sum_{j=1}^{d} \omega_j x_j \text{, де}$$

ω_0- вільний коефіцієнт або зсув.

ω_j- ваги або коефіцієнти.

2.3.3 Метод K найближчих сусідів

Це контрольований алгоритм навчання, який працює за принципом, згідно якого кожна точка даних, розташована поруч одна з одною, відноситься до одного класу. Різниця між звичайним методом найближчого сусіда полягає у тому, що замість одного об'єкта, що знаходиться найближче до нового, обирають певну кількість k-сусідів.

Через легкість та короткий час обчислення, метод k-найближчих сусідів використовується для задач класифікації, в тому числі і тексту.

Основний принцип класифікації методу k-найближчих сусідів полягає у наступному. Розглянемо набір відомих даних як точки у просторі. Усі елементи з ознакою першого класу будуть знаходитись поруч, наприклад в правому куті, а інші елементи, з ознаками другого класу – в лівому. Тепер уявимо що у нас є некласифікований об'єкт, який ми хочемо ідентифікувати як той чи інший клас, для цього треба обчислити відстань до всіх зразків у навчальному наборі даних, обрати k-зразків з найменшими відстанями та класифікувати новий об'єкт шляхом голосування: клас, кількість котрого переважає у k-найближчих сусідів, призначається новому об'єкту.

Для визначення кількості сусідів, яких хочемо розглядати вказує число K, однак з цим треба бути обережно. Вибір оптимальної кількості k-сусідів є критично важливим.

Відстань між точками зазвичай обчислюється за допомогою метрики відстані. Метрики використовуються такі:

- Евклідова: $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$;
- Мангеттенська: $d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$;
- Чебишова: $d(x,y) = \max_i|x_i - y_i|$;
- Мінковського: $d(x,y) = \sqrt[p]{\sum_{i=1}^{n} w(x_i - y_i)^p}$;
- Махаланобіса: $d(x,y) = \sqrt{(x-y)^T V^{-1}(x-y)}$.

2.3.4 Дерева прийняття рішень

Один з алгоритмів машинного навчання, який використовують для задач класифікації. Цей алгоритм імітує процес прийняття рішень людиною. Структура дерева складається з:

– основної вершини дерева, її ще називають корінь дерева. Це початок дерева до якого не веде ні одна дуга;

– вузлів дерева (гілки) – пункт прийняття рішень.

– листя дерева – вузли дерева, які не мають нащадків. Вони вказують на кінцеву класифікацію.

При побудові дерева прийняття рішень виконуються наступні етапи:

1. Вибір ознаки для розгалуження. Для кожного вузла обирається ознака, яка найкраще розділяє дані. Вибір ознаки може бути виконаний за допомогою критеріїв: індекс Джині, ентропія, середньоквадратична похибка.

2. Після вибору ознаки, дані розділяються на підмножини, що відповідають різним значенням цієї ознаки.

3. Рекурсивне повторення перших двох пунктів, поки не буде досягнуто кінцева умова зупинки (максимальна глибина дерева або мінімальна кількість зразків у вузлі).

2.4 Попередня обробка навчальних даних

Перед тим, як використовувати деякий набір даних для навчання моделі тим чи іншим методом, цей набір даних треба попередньо обробити. Це є одним з ключових етапів при створені буд-якої моделі, у тому ж числі і моделі класифікації.

Основна ціль – підготувати дані для надання у створювану модель, для підвищення якості класифікації.

При фільтрації спам-повідомлень навчальні дані складаються з множини текстів, тому попередня обробка буде включати набір обов’язкових кроків:

1. Очищення даних.

Під фразою «Очищення даних» мається на увазі видалення дублікатів, відсутніх значень, тощо. Це робиться для збільшення якості даних, та сприяє уникненню перенавчання.

2. Токенізація.

Токенізація – розбиття тексту на менші об’єкти, які називаються токенами. На цьому етапі рядки, абзаци розбиваються на токени. Зазвичай токеном є окреме слово, але це може бути ще символ, або частина слова. Це робиться з метою відокремити слова (токени) від інших синтаксичних знаків, цифр, спец-символів, інтернет адрес, тощо.

3. Видалення стоп-слів.

Стоп-слово – це поширені слова, які можна часто зустріти у тексті. Вони майже або взагалі не несуть змістовного значення для розуміння змісту тексту. Наприклад, слова: The, is, at, which, тощо. За допомогою видалення стоп-слів зменшується кількість унікальних слів у тексті або реченні, це дозволяє моделі зосередитись на більш важливих словах. Також стоп-слова можуть зустрічатись дуже часто у тексті, через що вони будуть створювати зайвий шум, заважаючи пошуку шаблонів (патернів) в наборі даних.

4. Лематизація та стемінг.

Лематизація – це процес групування відмінюваних форм слова так, щоб їх можна було проаналізувати як єдиний елемент, ідентифікований за лемою слова або словниковою формою.

Стемінг – це процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс. Результати стемінгу іноді дуже схожі на визначення кореня слова, але його алгоритми базуються на інших принципах.

Ці два методи використовуються для приведення схожих слів до одного, щоб зменшити кількість токенів. Проте, хоч вони і роблять одне й те саме, вони все ж таки відрізняються. Лематизація потребує словника конкретної мови, на якій був написаний текст для приведення до його кореня. В свою чергу стемінг просто відсікає зайві елементи слова, щоб привести його до кореня слова.

Якщо при тренуванні моделі важлива точність та контекст, тоді краще обрати лематизацію для обробки тексту, це може використовуватись якраз для задач обробки природної мови. Стемінг треба обирати, якщо необхідна швидкість, а не точність.

5. Векторизація.

Це процес перетворення тексту у числові вектори для подальшої обробки алгоритмами машинного навчання. Через те, що моделі машинного навчання працюють виключно з числовими даними, це робить даний етап одним з ключових для попередньої обробки тексту.

Розглянемо декілька методів векторизації тексту:

- Bag of Words (BoW) – метод векторизації тексту, при якому створюється словник унікальних слів зі всього набору даних. Кожен запис – документ, а кожен документ це вектор фіксованої довжини. Якщо у документі є слова, які відсутні у словнику, відповідна компонента вектору буде дорівнювати 0. Цей метод є простим в реалізації та принцип його роботи легко зрозуміти, однак основним недоліком є те, що порядок слів ігнорується, тобто семантичне значення речення може бути втрачене.

- Term Frequency-Inverse Document Frequency (TF-IDF) – метод векторизації тексту, який визначає частоту входження терміну у документі. Першим чином визначається наскільки часто термін зустрічається у документі, тобто обчислюється кількість його входжень до документу по відношенню кількості слів у документі. Другим етапом є визначити наскільки термін є унікальним у всьому копусі тексту, обчислюється логарифм відношення загальної кількості документів до кількості документів, в яких існує даний термін. Останній етап – добуток першого та другого кроку. Серед переваг цього методу можна виділити підвищення ваги рідкісних слів, які мають велике значення для класифікації та вирахування наскільки даний термін є важливим у корпусі тексту. Проте такий підхід має недолік у великій кількості обчислювальних ресурсів, котрі потрібні для обробки великого обсягу тексту.

6. Балансування класів.

Це процес коригування при дисбалансі між кількістю класів у наборі даних. При створені моделі для класифікації може виникнути ситуація, коли один клас переважає інший (кількість одного класу значно більша за інший). У такому випадку це може призвести до зменшення продуктивності моделі.

Для балансування використовуються такі підходи для балансування:

- Class Weights – підхід при якому кожному класу надається вага. Таким чином ми компенсуємо нерівновагу у класах.
- Data Sampling – підхід при якому можуть випадкові екземпляри з класу у котрого запису більше, або множинне додавання екземплярів до класу, котрий має меншу чисельність.

Висновок за розділом 2

Сучасний світ пропонує різноманітні методи машинного навчання, які використовуються для розв’язання задач класифікації, кластеризації, прогнозування даних. У розділі особливу увагу приділено методам керованого навчання. Цей підхід дозволяє ефективно тренувати моделі класифікації даних на основі навчальних даних (вибірок). Проте варто зауважити, що ефективність різних методів керованого навчання може відрізнятися, і вибір кращого підходу залежить від конкретних умов і властивостей даних. Тому важливо провести уважну оцінку та порівняти різні алгоритми, щоб визначити найкращий спосіб класифікації спам-повідомлень. Використання методів машинного навчання для боротьби зі спамом є перспективним напрямом, який дозволяє створювати більш адаптивні та точні системи захисту електронної пошти.

Також можна зазначити, що створити ефективну модель класифікації спам-повідомлень можна тільки за попередньою обробкою вибірки даних, які будуть використовуватись у навчанні моделі тим чи іншим алгоритмом. Попередня обробка даних, особливо текстових даних, є одним з основних етапів при створені моделі класифікації.

РОЗДІЛ 3
РОЗОРБКА МОДЕЛІ КЛАСИФІКАЦІЇ СПАМ-ПОВІДОМЛЕНЬ

3.1 Вибір інструментів для розробки моделі

3.1.1 Вибір мови програмування

При виборі мови програмування для побудови моделі за допомогою машинного навчання, треба спиратись на декілька факторів. Треба проаналізувати наявність бібліотек для машинного навчання, легкість у використанні та продуктивність. Серед запропонованих мов програмування, мною було обрано використовувати Python.

По-перше, Python – мова програмування, котра має легкий для зрозуміння синтаксис, що робить розробку коду швидкою та зрозумілою. За допомогою цього зменшується кількість часу на зрозуміння коду, бібліотек, тощо.

По-друге, Python містить став домінуючою мовою програмування у задачах машинного навчання, через свої технічні характеристики, архітектуру та екосистему. Він має усі бібліотеки, які потрібні не тільки для навчання моделі тим чи іншим методом, але й для попередньої обробки тексту.

Таким чином, можна сказати, що Python займає лідируюче місце серед розробників при вирішенні задач машинного навчання.

Основні переваги Python:

- Простота та зрозумілість синтаксису. Він має надзвичайно простий та інтуітивно зрозумілий синтаксис для користувача, це зменшує поріг входу для людей та дозволяє зосередитись на вирішенні задач, а не на те, як воно реалізовано.
- Продуктивність та оптимізація. Python не є найшвидшою мовою програмування, проте багато бібліотек у Python написані на низькорівнених мовах програмування C або C++.
- Багато бібліотек. Для Python кожен рік створюються нові бібліотек та оновлюються старі для підтримки продуктивності та вирішенні нових задач.

Наприклад він містить такі бібліотеки як NLTK (Natural Language Toolkit), ця бібліотека дозволяє обробляти текст, перед його використанням для навчання.

- Інтеграція з іншими мовами програмування. Python можна легко інтегрувати з іншими мовами програмування, що дозволяє створити мультимедійні середовища з використанням переваг кожної мови.

3.1.2 Бібліотека NLTK

Серед існуючих бібліотек для оброки природної мови, мною було обрано бібліотеку NLTK (Natural Language Toolkit). Ця бібліотека є одною з найпопулярніших для вирішення задач розпізнавання та обробки природної мови. Вона забезпечує широкий інструментарій для роботи з текстом, включаючи такі моменти як токенізація тексту, стемінг тексту, лематизація тексту, аналіз синтаксису та семантики тексту. У роботі був використаний наступний функціонал:

- Функція word_tokenize(). Ця функція використовується для одного з основних етапів попередньої обробки тексту. Вона дозволяє розділити текст, речення на окремі слова. За допомогою потужного токенізатора TreebankWordTokenizer, ця функція враховує багато особливостей англійської мови: пунктуація, лапки, скорочення, тощо.

- Модуль stopwords. Цей модуль містить у собі списки стоп-слів для різних мов. За допомогою цього модуля можна завантажити власні стоп-слова, або використовувати вже існуючи для конкретної мови. За допомогою вже існуючої колекції стоп-слів можна зосередитись на навчанні та удосконаленні моделі класифікації.

3.1.3 Бібліотека scikit-learn (sklearn)

У мові програмування Python існує багато бібліотек для машинного навчання. Серед вже існуючих бібліотек було вирішено взяти бібліотеку scikit-learn. Вона є однією з найпопулярніших бібліотек для вирішення задач машинного навчання. Бібліотека scikit-learn є універсальною та простою у

використанні за допомогою чого це робить її привабливішою серед інших бібліотек.

У роботі були використані наступні можливості даної бібліотеки:

- Модуль sklearn.metrics. Цей модуль створений для оцінки якості моделей машинного навчання. Він має усі інструменти для якісного та швидкого проведення алгоритмів оцінки моделі. За допомогою цього можна визначити наскільки добре наша модель виконує поставлену задачу, використовуючи різні метрики для цього. З цієї бібліотеки було використані метрики:
  - accuracy_score – метрика, за допомогою якої можна вимірювати точність класифікаційної моделі. Вона має наступну формулу: кількість правильних прогнозів поділена на загальну кількість зразків. Ця функція приймає 2 аргументи список з правильними мітками та список з прогнозованими мітками.
  - classification_report – метрика, за допомогою якої ми можемо отримати розширену оцінку продуктивності нашої моделі класифікації. Вона включає у собі такі метрики як точність, повнота, та підтримка. Розглянемо детальніше кожну з метрик:
  - точність (Precision) – надає можливість визначити часту правильних позитивних прогнозів серед усіх прогнозованих позитивних випадках. Визначається наступною формулою $Precision = \frac{TP}{TP+FP}$, де TP (True Positive) – кількість істино позитивних прогнозів, FP (False Positive) – кількість хибно позитивних прогнозів. При високій точності можна казати, що наша модель рідко робить хибні позитивні прогнози.
  - повнота (Recall) – надає можливість визначити частку правильних позитивних прогнозів по відношенню до усіх реальних позитивних випадків. Визначається наступною формулою $Recall = \frac{TP}{TP+FN}$, де TP (True Positive) – кількість істино позитивних прогнозів, FN (False Negative) – кількість хибно негативних прогнозів. При високій повноті можна казати, що наша модель рідко пропускає реальні позитивні випадки.

- F1-міра (F1-score) – дозволяє визначити середнє гармонійне між Precision та Recall. Визначається наступною формулою $F1\,Score = \frac{Precision*Recall}{Precision+Recall}$. Якщо значення F1-score близька до одиниці, тоді метрика вказує що модель добре збалансована між її точністю та повнотою. Вона є вкрай корисна, якщо важливо уникнути хибнопозитивних та хибнонегативних прогнозів.
- підтримка (Support) – метрика, котра дозволяє подивитись кількість реальних випадків кожного класу в тестовому наборі.

- Модуль sklearn.feature_extraction.text. Цей модуль створений для надання інструментів для перетворення текстових даних у числові ознаки. Цей модуль використовується для попередньої обробки тексту. У роботі були використані 2 найпоширеніших інструменти даного модуля:
  - CountVectorizer – функція, за допомогою якої здійснюється перетворення колекції текстових даних у матрицю частот слів.
  - TfidfVectorizer – функція, за допомогою якої здійснюється перетворення текстових даних у числові ознаки, використовуючи схему TF-IDF, котра була описана у другому розділі.
- Модуль sklearn.naive_bayes. Цей модуль створений для реалізації різниїх наївних байєсівських класифікаторів. У роботі були використані та створені 2 моделі наївного байєсівського класифікатору:
  - MultinomialNB – реалізація наївного байєсівського класифікатора, який підходить для задач з дискретними ознаками, такими як частоти або кількості певних подій. Особливо підходить до задач з класифікації тексту.
  - GaussianNB – реалізація наївного байєсівського класифікатора, яка передбачає що ознаки мають нормальний (гаусівський) розподіл. Цей метод машинного навчання також підходить до задач класифікації тексту, але не є таким ефективним.
- Модуль sklearn.model_selection. Цей модуль створений для вирішення задач з розділення даних. Він надає широкий спектр інструментів для розподілу

даних та оцінки і вдосконалення моделей машинного навчання. У роботі була використана одна з основних функцій даного модуля:

- Функція train_test_split – ця функція використовується для розділення набору даних на навчальну та тестову вибірки. За допомогою цього ми зможемо в подальшому оцінити ефективність нашої моделі для нових даних, яких вона не бачила при навчанні. Вона також дозволяє змінювати розміри тестових та навчальних наборів даних у відсотках. Ця функція приймає наступні аргументи:

- arrays – набір даних, котрі ми плануємо розділити. Вони можуть бути як матрицею даних, так і складатися з декількох масивів;
- test_size – Відсоток даних, які треба відібрати як тестувальний набір даних. Воно приймає число від 0 до 1, де 0 – порожня тестова вибірка, а 1 – складається з усіх даних;
- train_size – аналогічно як і test_size, тільки для тренувального набору даних;
- random_state – Випадковий стан для розділення даних на тестову та навчальну вибірки;
- shuffle – параметр, який приймає значення True або False. Призначений для перемішування даних перед розділенням. За замовчуванням значення цього параметру дорівнює Ture.

3.2 Опис набору даних для моделювання

При виборі набору даних, вирішив використати готову колекцію повідомлень (SMSSpamCollection, посилання на колекцію - https://archive.ics.uci.edu/dataset/228/sms+spam+collection), яка вже містить мітки "ham" для неспаму та "spam" для спаму. Ця колекція даних складається з повідомлень, які можуть бути електронними листами, текстовими повідомленнями або іншими формами текстового спілкування. Використовуючи цей набір даних, мої моделі навчалися відрізняти нормальні повідомлення від

спаму на основі їх змісту. Поза цією основною метою, цей набір даних був використаний для дослідження різних алгоритмів класифікації спаму та розробки ефективної моделі для фільтрації спам-повідомлень. Загалом, колекція складається з 5574 записів, що дозволяє проводити ретельні аналізи та експерименти з машинним навчанням.

3.3 Розробка моделей класифікації спам-повідомлень

При розробці програмної реалізації моделей було створено 4 файли з розширенням .py (main, file_reader, naive_bayes_spam_classifier, gaussian_bayes_spam_classifier). Перед початком написання програмного коду було прийнято рішення розділити його на декілька етапів.

3.3.1 Завантаження колекції та попередня обробка тексту

Перед початком налаштування та навчання моделей класифікації треба завантажити та попередньо обробити дані, які будуть використовуватись для навчання та перевірці вашої моделі. За допомогою функції load_data(file_path) виконується завантаження даних у списки з нашого файлу, який знаходиться за посиланням file_path. Ця функція дозволяє повертати декілька списків одразу, що є дуже корисним функціоналом через те, що наші дані поділені на мітки та речення.

3.3.2 Попередня обробка тексту

Наступним етапом є створення функції та використання вже існуючих інструментів для обробки вже завантаженого тексту. Мною було створено функцію preprocess_text(text), яка приймає один параметр – список з повідомленнями. У цій функції увесь набір даних проходить обробку для збільшення якості навчання та тестування моделей. Як описано у розділі 2, першим чином текст було переведено до нижнього регістру, після чого було проведено видалення знаків пунктуації з тексту.

Далі відбувається токенізація тексту з подальшим видаленням стоп-слів. Закінченням обробки стає об’єднання токенів назад у текст та повернення списку записів з функції. За допомогою цих нескладних операцій був зменшений розмір

тексту, прибрана пунктуація, видалені стоп-слова. На цьому етапі попередня обробка закінчується, текст готовий для тренування моделі.

3.3.3 Розділення даних на тестову та тренувальну

Це етап робиться з метою розділити набір даних на 2 частини. Тренувальний набір даних буде використаний для навчання певного алгоритму класифікувати текст. Оскільки були обрані методи навчання з вчителем, які потребують навчальні початкові дані.

3.3.4 Тренування моделей

Один з заключних етапів програми, в якому моделі передається навчальний набір даних з їх мітками. Цей етап треба розібрати детальніше, тоскільки для двох обраних моделей машинного навчання існує різниця між їх навчанням.

Для розділення моїх моделей були створені окремі файли з їх назвами, які були описані на початку розділу. Таким чином маємо 2 моделі машинного навчання: Multinomial та Gaussian naive bayes. Для обох моделей створені 3 функції. Розглянемо ці функції окремо:

- Функція train_model(X_train, y_train). Ця функція приймає 2 аргументи. Першим аргументом є список з даними, а другий – мітки до цих даних. Наступним чином дані ще треба перетворити у числовий формат, але вже з урахуванням яку модель будемо навчати. Для MultinomialNB використовуємо векторизацію за допомогою об’єкту типу CountVectoricer, який використовується для перетворення текстових даний у числові ознаки. Тоді як для GaussianNB використовуємо об’єкт типу TfidfVectorizer для перетворення текстових даних у числові ознаки за допомогою TF-IDF. Після векторизації для GaussianNB ще треба перетворити розріджену матрицю у щільну. Наступний крок – створити об’єкти MultinomialNB та GaussianNB, таким чином ми створюємо сиру модель. Останній етап – передача даних до моделі за допомогою функції fit(), яка є у обох методів машинного навчання. У цю функцію передаються тренувальні дані та їх мітки, після чого з функції повертається модель та вектор даних.

- Функція classify_spam(). Згідно з її назви можна зрозуміти, що ця функція буде класифікувати дані на основі вже навченої моделі, вектору та тренувального набору даних з її мітками. Першим етапом векторизуємо тестові дані так само як і у попередній функції. Після чого для GaussianNB треба ще ці дані перетворити у щільну матрицю. Наступним етапом модель класифікує дані на основі свого навчання за допомогою функції predict(). Заключним етапом цієї функції є вивід у консоль оцінки моделі, щоб можна було побачити ці показники та зробити висновки стосовно цієї моделі. Ця інформація допоможе детальніше зрозуміти чи модель навчена добре, або треба її перенавчати.
- Функція check_message(). Ця функція була створена для обробки окремого повідомлення і може використовуватись вже для моделей, в яких необхідно фільтрувати спам-повідомлення. Повідомлення, яке передаємо до створеної моделі також проходить веторизацію та за допомогою функції predict(), модель класифікує його як спам, або легітимне повідомлення. Треба зазначити, що для більш якісної класифікації окремого повідомлення його також треба попередньо обробити, інакше модель може неправильно класифікувати його.

3.4 Перевірка результатів та порівняння моделей

Згідно з програмної реалізації у консоль нам виводять деякі показники для кожної моделі. Пропоную розглянути ці показники та зробити висновок для кожної моделі а також перевірити модель на якість класифікації спам повідомлень при виникненні нового повідомлення.

```
Enter message for classificationIs that seriously how you spell his name?
The message is classified as ham.
The message is classified as ham.
```

Рисунок 3.1 – Перевірка моделей на прикладі легітимного повідомлення

```
Enter message for classificationHi! Do you want to earn money? Just call me and claim your reward!
The message is classified as spam.
The message is classified as spam.
```

Рисунок 3.2 – Перевірка моделей на прикладі спам повідомлення

3.4.1 Аналіз поліноміальної байєсівської моделі

Далі буде проведений аналіз на основі результатів роботи поліноміальної байєсівської моделі. Усі дані були отримані в результаті роботи моделі. У таблиці 3.1 наведені результати тренування і тестування створеної моделі.

Таблиця 3.1

Результати для моделі поліноміального наївного байєсу

	PRECISION	RECALL	F1-SCORE	SUPPORT
HAM	0.98	1.00	0.99	2403
SPAM	0.97	0.90	0.93	383
ACCURACY	-	-	0.98	2786
MACRO AVG	0.98	0.95	0.96	2786
WEIGHTED AVG	0.98	0.95	0.98	2786

Аналіз метрик для класу «ham»

- Persition (0.98): Точність становить 98%, що означає, що з усіх повідомлень, які модель класифікувала як "ham", 98% дійсно є "ham".
- Recall (1.00): Повнота становить 100%, що означає, що модель виявила всі реальні випадки "ham" у тестовій вибірці. Жодного не спам-повідомлення не було пропущено.
- F1-score (0.99): F1-оцінка є середнім гармонійним точності і повноти і становить 99%, що вказує на відмінний баланс між точністю і повнотою для цього класу.
- Support (2403): Кількість зразків класу "ham" у тестовій вибірці.

Аналіз метрик для класу «spam»

- Precision (0.97): Точність становить 97%, що означає, що з усіх повідомлень, які модель класифікувала як "spam", 97% дійсно є спамом. Є певна кількість помилкових позитивів (неспам-повідомлень, класифікованих як спам).
- Recall (0.90): Повнота становить 90%, що означає, що модель виявила 90% реальних випадків спаму. Деякі спам-повідомлення були пропущені (класифіковані як "ham").
- F1-score (0.93): F1-оцінка для спаму становить 93%, що вказує на хороший баланс між точністю і повнотою для цього класу.
- Support (383): Кількість зразків класу "spam" у тестовій вибірці.

Аналіз загальних метрик:

- Accuracy (0.98): Загальна точність моделі становить 98%, що вказує на дуже високий рівень правильно класифікованих зразків.
- Macro avg:
  - Precision (0.98): Середня точність по всіх класах.
  - Recall (0.95): Середня повнота по всіх класах.
  - F1-score (0.96): Середня F1-оцінка по всіх класах.
- Weighted avg:
  - Precision (0.98): Зважена середня точність з урахуванням кількості зразків у кожному класі.
  - Recall (0.98): Зважена середня повнота з урахуванням кількості зразків у кожному класі.
  - F1-score (0.98): Зважена середня F1-оцінка з урахуванням кількості зразків у кожному класі.

Таким чином, модель машинного навчання MultinomialNB є дуже ефективною для класифікації спам-повідомлень. Класифікація не спам-повідомлень майже безпомилкова, з дуже високими показниками точності та повноти. У моделі є невеликі проблеми з виявленням спам-повідомлень, іноді вона може класифікувати не спам як спам. Загалом, метрики вказують на те, що

наша модель дуже збалансована та ефективно класифікує класи а середні значення вказують на те, що модель не упереджена щодо класу з більшою кількістю зразків «ham».

3.4.2 Аналіз гаусівської байєсівської моделі

Далі буде проведений аналіз на основі результатів роботи гаусівської байєсівської моделі. Усі дані були отримані в результаті роботи моделі. У таблиці 3.2 наведені результати тренування і тестування гаусівської байєсівської моделі.

Таблиця 3.2

Результати для моделі гаусівського наївного байєсу

	PRECISION	RECALL	F1-SCORE	SUPPORT
HAM	0.97	0.89	0.93	2403
SPAM	0.56	0.85	0.67	383
ACCURACY	-	-	0.89	2786
MACRO AVG	0.77	0.87	0.80	2786
WEIGHTED AVG	0.92	0.89	0.90	2786

Аналіз метрик для класу «ham»

- Precision (0.97): Точність становить 97%, що означає, що з усіх повідомлень, які модель класифікувала як "ham", 97% дійсно є "ham". Однак 3% прогнозів "ham" є помилковими (це спам-повідомлення).
- Recall (0.89): Повнота становить 89%, що означає, що модель виявила 89% реальних випадків "ham". 11% неспам-повідомлень були пропущені (класифіковані як спам).
- F1-score (0.93): F1-оцінка є середнім гармонійним точності і повноти і становить 93%, що вказує на хороший баланс між точністю і повнотою для цього класу.
- Support (2403): Кількість зразків класу "ham" у тестовій вибірці.

Аналіз метрик для класу «spam»

- Precision (0.56): Точність становить 56%, що означає, що з усіх повідомлень, які модель класифікувала як "spam", лише 56% дійсно є спамом. 44% прогнозів "spam" є помилковими (це неспам-повідомлення).
- Recall (0.85): Повнота становить 85%, що означає, що модель виявила 85% реальних випадків спаму. 15% спам-повідомлень були пропущені (класифіковані як "ham").
- F1-score (0.67): F1-оцінка для спаму становить 67%, що вказує на значний дисбаланс між точністю і повнотою для цього класу.
- Support (383): Кількість зразків класу "spam" у тестовій вибірці.

Аналіз загальних метрик:

- Accuracy (0.89): Загальна точність моделі становить 89%, що є нижчим показником порівняно з MultinomialNB..
- Macro avg:
  - Precision (0.77): Середня точність по всіх класах.
  - Recall (0.87): Середня повнота по всіх класах.
  - F1-score (0.80): Середня F1-оцінка по всіх класах.
- Weighted avg:
  - Precision (0.92): Зважена середня точність з урахуванням кількості зразків у кожному класі.
  - Recall (0.89): Зважена середня повнота з урахуванням кількості зразків у кожному класі.
  - F1-score (0.90): Зважена середня F1-оцінка з урахуванням кількості зразків у кожному класі.

Отже, модель машинного навчання GaussianNB показує добрі результати при класифікації не спам-повідомлень. Однак ця модель має суттєві проблеми з точністю для класифікації спам-повідомлень, що в свою чергу призводить до великої кількості помилкових позитивних результатів. Згідно з аналізу загальної

точності можна сказати, що модель є високою, проте вона потребує покращення для збільшення ефективності класифікації спам-повідомлень.

Висновок за розділом 3

За результатами, отриманими у третьому розділу, можна зробити наступні висновки.

На сьогодні створити модель класифікації спам-повідомлень на основі машинного навчання не займає великого часу та має невеликий поріг входження. Існуючі бібліотеки для вирішення даної задачі має об’ємний спектр інструментів, які допоможуть з цим.

Порівнюючи два алгоритми машинного навчання можна сказати, що мультиноміальний байєсвіський класифікатор більш ефективний при побудові моделі класифікації спам-повідомлень, ніж гаусівська байєсівська модель. Це може бути обумовлено різними підходами векторизації даних при навчанні моделей.

ВИСНОВОК

Під час виконання кваліфікаційної роботи було виконаний аналіз наявних стратегій протидії спаму. В рамках цього аналізу були розглянуті різноманітні види спаму та методи їх запобігання, які можуть бути використані користувачами. Крім того, були досліджені основні характеристики, що відрізняють спам-повідомлення від легітимних, щоб краще розуміти їхню природу та знаходити ефективні методи боротьби з ними.

У другому розділі проведено детальний огляд наявних методів машинного навчання, що дозволило визначити та усвідомити, які саме методи є найбільш ефективними у розв'язанні завдань класифікації спам-повідомлень. В результаті аналізу було вибрано моделі та підходи, які демонстрували найвищу точність та надійність у розпізнаванні та фільтрації спаму.

Заключним етапом стало розроблення двох моделей класифікації спаму на основі різних методів машинного навчання. Цей процес включав у собі підбір оптимальних параметрів моделей, їхню тренування та тестування на відповідному наборі даних. Після цього було проведено аналіз результатів роботи цих моделей, щоб визначити їхню ефективність та порівняти їхню продуктивність. Такий підхід дозволяє виявити найкращі методи класифікації спаму та визначити оптимальну стратегію боротьби з цією проблемою. У цьому розділі також була проведена попередня обробка даних перед їх подальшим використанням для навчання моделей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa. Machine learning for email spam filtering: review, approaches and open research problems, 2019. 23 p.

2. K-Nearest Neighbor(KNN) Algorithm for Machine Learning – режим доступу URL: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (дата звернення 12.01.2024)

3. Natural Language Toolkit Documentation – режим доступу URL: https://www.nltk.org/index.html (дата звернення 02.03.2024)

4. Python Developer's Guide – режим доступу URL: https://devguide.python.org/ (дата звернення 15.02.2024)

5. Skitit-learn. Machine Learning in Python – режим доступу URL: https://scikit-learn.org/stable/index.html (дата звернення 01.03.2024)

6. Teodorescu M.H. Machine Learning Methods for Strategy Research. HBS Working Paper 18-011. Harvard Business School, 2017. 59 p.

7. Дерево ухвалення рішень – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D1%83%D1%85%D0%B2%D0%B0%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%80%D1%96%D1%88%D0%B5%D0%BD%D1%8C (дата звернення 13.01.2024)

8. Кероване навчання – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення 15.01.2024)

9. Лінійні регресії – режим доступу URL: https://w3schoolsua.github.io/ai/ai_regressions.html#gsc.tab=0 (дата звернення 15.01.2024)

10. Машинне навчання – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення 10.01.2024)

11. Напівкероване навчання – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%BA%D0%B5_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (15.01.2024)

12. Некероване навчання – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BA%D0%B5%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F (15.01.2024)

13. Олена Мусієнко. Чому спам називається спамом: несподівана історія популярного терміна – режим доступу URL: https://www.imena.ua/blog/why-spam-is-called-spam/

14. Спам – режим доступу URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B0%D0%BC (дата звернення 10.01.2024)

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
_____д.т.н., проф. Шматков С. І.
«21» грудня 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Яковлева Данила В'ячеславовича
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання»**

керівник роботи Стрілець Вікторія Євгенівна, к.т.н. доцент кафедри ТПС
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024року № 4101-5/909

2.Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити:

1) Аналіз сучасних засобів і методів фільтрації спам-повідомлень.
2) Розробка методу фільтрації спам-повідомлень на основі алгоритмів машинного навчання.
3) Програмна реалізація методу фільтрації спам-повідомлень на основі алгоритмів машинного навчання та її тестування.

4.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Огляд наукової літератури та аналіз сучасних засобів і підходів до фільтрації спам-повідомлень	21.12.2023 - 10.01.2024
2	Аналіз існуючих методів фільтрації спам-повідомлень	11.01.2023 - 21.01.2024
3	Розробка методу фільтрації спам-повідомлень із використанням алгоритмів машинного навчання	22.01.2024 - 10.03.2024
4	Первинне тестування та удосконалення методу фільтрації спам-повідомлень	11.03.2024 - 20.03.2024
5	Оформлення рекомендацій для використання методу	21.03.2024 - 30.03.2024
6	Оформлення пояснювальної записки	31.03.2024 - 10.05.2024
7	Надання кваліфікаційної роботи науковому керівнику та рецензенту	11.05.2024 - 19.05.2024
8	Оформлення супроводжувальної документації	20.05.2024 – 27.05.2024

5. Дата видачі завдання 21.12.2023 р.

Студент Яковлев Д.В. _____
ініціали, прізвище підпис

Керівник роботи Стрілець В.Є. _____
ініціали, прізвище підпис

Додаток Б

Затверджую

_____

«_____»_____2024 р.

Технічне завдання

на розробку програмного виробу «Метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання».

1.	Введення	1.1 Назва програмного виробу – Метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання. 1.2. Галузь застосування: Інформаційні технології
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра №_____від «__»_____2024 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: створення методу для фільтрації спам повідомлень на основі алгоритмів машинного навчання. 3.2. Призначення розробки: можливість використання для класифікації та запобігання розповсюдженню спам-повідомлень. 3.3. Вхідні дані: набір повідомлень, спам-повідомлення, легітимні повідомлення. 3.4. Вихідні дані розробки: отримання класифікації повідомлення
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: покращення класифікації повідомлень. 4.2. Вимоги до надійності: забезпечення безпомилкової фільтрації даних у вигляді тексту 4.3. Вимоги до умов експлуатації: наявність встановленої мови програмування, наявність інтегрованого середовища розробки. 4.4. Вимоги до складу і параметрів технічних засобів: Для виконання програми повинен підходити ПК з будь-якою операційною системо. 4.5. Вимоги до інформаційної та програмної сумісності: Підтримка операційних систем Windows або Linux, підтримка пожливостей мови програмування Python.

<table>
<tr><td></td><td></td><td colspan="2">4.6. Вимоги до маркування та упаковки: дані вимоги не представляються.
4.7. Вимоги до транспортування і зберігання: дані вимоги не представляються.
4.8. Спеціальні вимоги: дані вимоги не представляються</td></tr>
<tr><td>5.</td><td>Вимоги до програмної документації</td><td colspan="2">Програмною документацією до виробу «метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання» вважати:
1) Ознайомчу сторінку з правилами користування програми
2) Програму і методику випробувань розробленої програми (представити як додаток В до пояснювальної записки до кваліфікаційної роботи).
3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).</td></tr>
<tr><td>6.</td><td>Вимоги до техніко-економічних показників</td><td colspan="2">Програмною документацією до виробу «метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання» вважати:
1) Справжнє Технічне завдання на розробку методу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).
2) Опис програмної реалізації методу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).
3) Джерела базової інформації.</td></tr>
<tr><td rowspan="6">7.</td><td rowspan="6">Стадії і етапи розробки</td><td>Дата</td><td>Назва етапу</td></tr>
<tr><td>21.12.2023 – 10.01.2024</td><td>Огляд наукової літератури та аналіз сучасних засобів і підходів до фільтрації спам-повідомлень</td></tr>
<tr><td>11.01.2024 – 21.01.2024</td><td>Аналіз існуючих методів фільтрації спам-повідомлень</td></tr>
<tr><td>22.01.2024 – 10.03.2024</td><td>Розробка методу фільтрації спам-повідомлень із використанням алгоритмів машинного навчання</td></tr>
<tr><td>11.03.2024 – 20.03.2024</td><td>Первинне тестування та удосконалення методу фільтрації спам-повідомлень</td></tr>
<tr><td>21.03.2024 – 30.03.2024</td><td>Оформлення рекомендацій для використання методу</td></tr>
</table>

		31.03.2024 – 10.05.2024 11.05.2024 – 19.05.2024 20.05.2024 – 27.05.2024	Оформлення пояснювальної записки Надання кваліфікаційної роботи науковому керівнику та рецензенту Оформлення супроводжувальної документації
8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки методу виконувати раз в 2 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в примірнику на CD-R компакт-диску.	

Виконавець
Студент групи КІ-41
Яковлев Д.В,

Замовник
канд. техн. наук
Стрілець В. Є.

Додаток В

Програма і методика випробувань програмного виробу

«Метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання»

1. Об’єкт випробувань

1. Назва програмного виробу: «Метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання».

2. Галузь застосування: Інформаційні технології

3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності програмної реалізації методу із заявленим ифункціональними можливостями в технічному завдання (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп’ютерного класу кафедри в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавці та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

4.1. Вимоги до функціональних характеристик: покращення класифікації повідомлень.

4.2. Вимоги до надійності: забезпечення безпомилкової фільтрації даних у вигляді тексту

4.3. Вимоги до умов експлуатації: наявність встановленої мови програмування, наявність інтегрованого середовища розробки.

4.4. Вимоги до складу і параметрів технічних засобів: Для виконання програми повинен підходити ПК з будь-якою операційною системо.

4.5. Вимоги до інформаційної та програмної сумісності: Підтримка операційних систем Windows або Linux, підтримка пожливостей мови програмування Python.

4.6. Вимоги до маркування та упаковки: дані вимоги не представляються.

4.7. Вимоги до транспортування і зберігання: дані вимоги не представляються.

4.8. Спеціальні вимоги: дані вимоги не представляються

5. Вимоги до програмної документації

Програмною документацією до виробу «метод фільтрації спам-повідомлень на основі алгоритмів машинного навчання» вважати:

1) Справжнє Технічне завдання на розробку методу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) Опис програмної реалізації методу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

3) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні засоби: мова програмування Python, інтегроване середовище розробки PyCharm.

6.2. Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

– Ознайомчий (1-й етап);

– Власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірку комплектності складу програмної документації здійснюються за критерієм наявності зазначеної в ТЗ документації

2) Перевірка якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 19.301-79 ЕСПД «Програма і методика випробувань».

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.

2) Перевірку ступеня виконання функціональних вимог до програми.

3) Методику проведення перевірок:

a) Запустити проект у PyCharm.

b) Порядок проведення випробувань:

– у файлі з назвою main.py знайти зміну під назвою message_to_check.

– Записати у цю змінну власне повідомлення на англійській мові.

– Запустити програму та перевірити чи правильно класифікує власне повідомлення навчані моделі класифікації.

Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Для проведення випробувань пропонується тест 1, тест 2

Тест 1

1. Перевірка виконання програми;

2. Перевірка правильності класифікації повідомлення, яке вважається спамом;
3. Перевірка власноруч чи дійсно це повідомлення вважається спамом.

```
Spam message: Congratulations! You've won a free trip to Hawaii. Click here to claim your prize!
The message is classified as spam.
```

Рис. В.1 Тест 1

Тест вважається успішним якщо наше повідомлення класифікується як spam.

Тест 2

1. Перевірка виконання програми;
2. Перевірка правильності класифікації повідомлення, яке вважається легітимним;
3. Перевірка власноруч чи дійсно це повідомлення вважається легітимним.

```
Spam message: Is that seriously how you spell his name?
The message is classified as ham.
```

Рис. В.2 Тест2

Тест вважається успішним якщо наше повідомлення класифікується як легітимне (ham)

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ41, Яковлев Д.В.