МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Центр післядипломної освіти

Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів класифікації листів українською мовою з метою виявлення спаму

Виконав:
Студент 2 курсу, групи ІПЗздм-19-1
Нечіпор В.О.

Спеціальність 121- Інженерія програмного забезпечення

Тип програми Освітньо-наукова

Керівник проф. Єрохін А.Л.

Допускається до захисту

Зав. кафедри З.В. Дудар

2021 р.

Харківський національний університет радіоелектроніки

Центр післядипломної освіти

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121- Інженерія програмного забезпечення

Тип програми Освітньо-наукова

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« 26 » березня 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Нечіпора Володимира Олександровича

1. Тема роботи Дослідження методів класифікації листів українською мовою з метою виявлення спаму

затверджена наказом університету від 25.01.2021 № 385

2. Термін подання роботи до екзаменаційної комісії 9.05 2021р.
3. Вихідні дані до роботи В результаті дослідження вдалось підняти ефективність прогнозованої класифікації спаму українською мовою наївного класифікатору з 82,7% до 88,3%, що також перевищило ефективність мультиноміального (85,4%) методу без модифікації для розпізнавання апострофів.
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, огляд методів класифікації текстів,огляд наявних аналогів вирішення задачі,розробка програмного забезпечення з використанням методів класифікації Байєса і порівняння результатів, розробка вдосконаленого методу класифікації для роботи з листами українською мовою
5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій

Кі-сть спам-повідомлень на рік (в млрд.), доступні фільтри спаму на ринку, передумови існування спам фільтру української мови, зібраний датасет, машинне навчання з учителем, вибір методу класифікації спаму, реалізація через сервіс Datalore від Jetbrains, розробка наївного класифікатора і порівняння ефективності двох методів, середнє арифметичне ефективності класифікаторів, проблема класифікації повідомлень з апострофами, модифікація наївного методу класифікатора, середнє арифметичне ефективності мільтиноміального і модифікованого наївного класифікаторів, тестування ефективності методів на датасеті з апострофами, висновки

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Єрохін А.Л.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	25 лютого 2021р.	
2.	Огляд методів класифікації текстів	05 березня 2021р.	
3.	Програмна реалізація мультиноміального класифікатора за Байєсом	28 березня 2021 р.	
4.	Програмна реалізація наївного класифікатора за Байєсом	01 квітня 2021 р.	
5.	Порівняння наївного і мультиноміального методів	03 квітня 2021 р.	
6.	Програмна реалізація модифікованого методу класифікації текстів	06 квітня 2021 р.	
7.	Подання статті в журнал «Біоніка Інтелекту»	21 квітня 2021 р.	
8.	Фінальне оформлення роботи	27 квітня 2021 р.	

Дата видачі завдання 25 січня 2021р.

Студент _____
(підпис)

Керівник роботи _____ проф. Єрохін А.Л.
(підпис) (посада, прізвище, ініціали)

Харківський національний університет радіоелектроніки

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 69 с., 8 рис., 6 табл., 21 джер.

МАШИННЕ НАВЧАННЯ, МЕТОД КЛАСИФІКАЦІЇ БАЙЄСА, МУЛЬТИНОМІАЛЬНИЙ МЕТОД, НАЇВНА КЛАСИФІКАЦІЯ, УКРАЇНСЬКА МОВА, АПОСТРОФ, СПАМ, ПРОГНОЗУВАННЯ.

Мовний аналіз текстів з метою їх подальшої класифікації є об’єктом дослідження. Метою роботи є підвищення ефективності класифікації листів українською мовою з метою фільтрування спаму. Методи розробки базуються на таких технологіях, як мова програмування Python, Datalore з використанням бібліотек pandas, nltk, sklearn.

В рамках дослідження було проаналізовано недоліки методу класифікації Байєса в рамках сучасної реалізації цього методу на мові програмування Python для роботи з українською мовою. Основним недоліком програмної реалізації методу Байєса було виявлено некоректний для української мови поділ на слова за умови, що слова містять апостроф. Для виправлення цієї проблеми було розроблено модифікований метод класифікації за Байєсом, який коректно працює з словами української мови, що містять апостроф. В результаті вдалось підняти ефективність прогнозованої класифікації спаму наївного класифікатору з 82,7% до 88,3%, що також перевищило ефективність мультиноміального (85,4%) методу без модифікації для розпізнавання апострофів.

MACHINE LEARNING, BAYES CLASSIFICATION METHOD, MULTINOMIAL METHOD, NAÏVE METHOD, UKRAINIAN LANGUAGE, APOSTROPHE, SPAM, FORECASTING

Modern analysis of texts for the purpose of their further classification in context of spam filtering is the object of the research. The aim of the work is to evaluate the effectiveness of classification of letters written in Ukrainian language in

order to filter spam. Development methods are based on such technologies as Python programming language, Datalore tool and such libraries as: pandas, nltk, sklearn.

As a result of the study, the shortcomings of the Bayesian classification method towards Ukrainian language were analyzed in the context of the modern implementation of this method in Python programming language. The main disadvantage of the software implementation of the Bayesian method was incorrect for the Ukrainian language parsing of sentences into words, provided that the words contain apostrophes. To correct this problem, a modified Bayesian classification method was developed, which correctly splits sentences into words, even if they contain apostrophes. As a result, the efficiency of the predicted spam classification via naïve Bayes method was raised from 82,7% to 88,3%, which is also higher than 85,4% of effectiveness of multinomial classification method without custom modification for apostrophe in the Ukrainian language.

Я, Нечіпор Володимир Олександрович, студент гр. ІПЗздм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «*вказати тему кваліфікаційної роботи*», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Життя в сучасному світі тісно пов’язане з онлайн присутністю – Інтернет є джерелом новин, соціальною мережею, архівом інформації про будь-що з історії, робочою платформою, інструментом для освіти, тощо. У 2020-му році більш ніж 4.5 мільярди людей постійно користувались Інтернетом, а це більша половина населення планети [2].

Невід’ємною частиною онлайн присутності є електронна пошта, яка використовується для ідентифікації користувача в мережі і листування, отримання звітів і чеків про купівлю в мережі. Згідно останньою статистики щоденно відправляється 319 мільярдів листів [1]. Чи всі вони бажані? Однозначно ні.

Електронна пошта є не тільки зручним інструментом, але й небезпекою – величезна кількість небажаних листів щоденно розсилається багатьом користувачам, чия електронна адреса стала відома тим, хто розсилає їх. Саме феномен відправки небажаних листів називається спамом. При цьому зміст листів може бути як просто набридливою рекламою, так і погрозами, спробами обманути і заманити на не доброчесний сайт, перейти за фішинговим посиланням, розсилкою з інформацією про незаконні ресурси: порно, жорстокі веб-сайти, тощо.

Феномен спаму набуває нечуваних масштабів і, напевно, кожен з вищезазначених 4.5 мільярдів Інтернет користувачів хоч раз був жертвою розсилки небажаних листів. Це робить розробку сервісів захисту від спаму надзвичайно актуальним питанням.

В основі будь-якої програми захисту від спаму (далі називатимемо їх спам-фільтрами) лежить мовний аналіз, а саме алгоритм, який зможе класифікувати повідомлення за бінарним критерієм «спам»/«не спам». Необхідність мовного аналізу ускладнює створення спам-фільтрів тим, що різні мови належать до різних груп мов, відрізняються граматичними і семантичним

правилами побудови речень, абеткою, містять унікальні допоміжні символи (як апостроф в українській мові). Мовний аналіз текстів з метою їх подальшої класифікації є об’єктом дослідження.

На сьогодні існують готові рішення для спам-фільтрів найбільш популярних мов, в першу чергу для англійської як для найпоширенішої мови світу. В той же час український сектор лише починає розвиватись в напрямку створення сервісів, які вміють досконало аналізувати українську мову. Відсутність аналогів спам-фільтрів української мови робить їх розробку надзвичайно актуальною. Саме методи класифікації текстів українською мовою і є предметом дослідження.

Спам-фільтр є одним зі складних прикладів програмного забезпечення і вимагає застосування машинного навчання. В рамках дослідження ми розробимо нейронну мережу з вчителем.

Метою даної роботи є визначення ефективного методу класифікації текстів українською мовою на основі проведеного порівняльного аналізу методів класифікації і розробка вдосконаленого методу.

В рамках мети виділимо наступні задачі дослідження:

– проаналізувати ринок на предмет наявності доступних повноцінних аналогів або сервісів, що частково підійдуть для досягнення мети роботи;

– провести аналіз методів класифікації текстів;

– розробити програмне забезпечення спам-фільтру українською мови з використанням різних методів класифікації текстів;

– розробити модифікований метод класифікації текстів української мови на основі одного з розроблених у попередньому розділі;

– проаналізувати результат використання різних методів класифікації;

– підвести підсумки і визначити ефективній метод класифікації текстів українською мовою з метою спам-фільтрації.

Поставлені задачі було виконано в повному обсязі в рамках даного дослідження. Було розроблено мільтиноміальний класифікатор Байєса і наївний класифікатор Байєса. Результатом порівняння двох методів є висновок, що серед доступних на даний момент методів для роботи з українськими текстами, мультиноміальний метод класифікації Байєса дає дещо кращі результати вірних передбачень у порівнянні з результатами класичного наївного класифікатора:

– мультиноміальний метод класифікатору Байєса: 0,854 або 85,4%;
– наївний метод класифікатору Байєса: 0,827 або 82,7%.

В процесі розробки вищевказаних методів було помічено ваду обробки текстів української мови, за умови, що слова містять апостроф. Оскільки апостроф є невід'ємною частиною українського слова, він не є пунктуаційним знаком, яким можна знехтувати. Проте більшість розроблених інструментів для роботи з мовами орієнтована на те, що апостроф можна прибрати без втрати сенсу повідомлення (як, наприклад, в англійській мові).

З метою усунення даної проблеми було внесено модифікації в наївний класифікатор Байєса, які розподіляють речення на слова з урахуванням особливостей української мови. Ефективність модифікованого методу склала 0,883, або 88,3%, що перевищило ефективність і звичайного наївного методу класифікатору Байєса і мультиноміального класифікатора з бібліотеки nltk.

Результати дослідження вважаються задовільними, адже цілі роботи, поставлені перед автором, були повністю виконані:

– ефективність прогнозів двох різновидів класифікатору текстів за Байєсом було порівняно між собою як середнє арифметичне ефективності за 10 спроб для кожного з них;
– проблему обробки українського тексту у вигляді слів з апострофами встановлено;
– модифікований метод наївного класифікатору, який може працювати з українськими словами, розроблено і його ефективність перевищує

таку у мультиноміального методу з бібліотеки nltk і у наївного класифікатору без модифікації.

Практична користь роботи полягає у можливості використання модифікованого методу для виконання задачі класифікації тексту українською мовою з більшою ефективністю, ніж доступні аналоги на мові Python.

В рамках виконання роботи до науково-технічного журналу «Біоніка Інтелекту» було подано статтю під назвою «Модифікація методу класифікації Байєса в задачах виявлення спаму українською мовою»[15].

1 АНАЛІЗ СТАНУ ВИРІШЕННЯ ПРОБЛЕМИ

Станом на 2020 рік близько половини всіх електронних листів є спамом[1], що зумовлює потребу створення відповідного програмного забезпечення для фільтрації спаму.

Можемо описати задачу фільтрації листів як задачу класифікації текстів на дві категорії: «спам» і «не спам». Класифікація тексту є традиційним завданням машинного навчання.

Машинне навчання (Machine learning, ML) – звід методів в області штучного інтелекту, набір алгоритмів, які застосовують, щоб створити машину, яка вчиться на власному досвіді. В якості навчання машина обробляє величезні масиви вхідних даних і знаходить у них закономірності.

Для задачі класифікації текстів, наприклад, після того, як закономірності були знайдені, відбувається розподіл тестових даних згідно конкретних закономірностей і робиться висновок про приналежність тексту до певної категорії.

В рамках нашого дослідження достатньо цього висновку класифікатора для визначення успішності моделі, однак іноді задачі набагато складніші. У роботі [17], наприклад, реалізована регресійна модель на основі LARS з використанням даних про два значимих фактори, які виділялись при навчанні моделі. Ці дані використовувались для визначення ступеню обструкції носоглотки пацієнта, що також свідчить про реальну сферу практичного застосування таких досліджень.

В рамках даного дослідження користуватимемось існуючими надбаннями в області машинного навчання для створення відповідного програмного забезпечення фільтрації спаму.

Визначимо необхідні елементи для створення програмного забезпечення спам-фільтру української мови:

– датасет текстів української мови, класифікованих за принципом «спам»/«не спам»;

– інструменти текстового аналізу (стеммінг, прибирання стоп-слів, тощо.).

Розглянемо їх детальніше.

1.1 Датасет українського спаму

Хоча задача фільтрації спаму не нова, найбільшого розвитку вирішення цієї проблеми досягло саме в рамках англійської мови, як однієї з найбільш популярних мов в Інтернет кореспонденції. На веб-сайті kaggle.com, наприклад, легко знайти чіткі інструкції з налаштування спам-фільтру [11], а також готові датасети текстів, які вже класифіковані за принципом «спам»/«не спам» [12].

Збір, аналіз і класифікація такого дата-сету є окремою масштабною задачею, тому його наявність конкретною мовою є важливим елементом для підготовки в роботі зі спам-фільтром.

В рамках роботи на даним дослідженням нами не було виявлено доступних датасетів спам текстів українською мовою. Це підтверджує актуальність даного дослідження і створює перепони для досягнення мети роботи.

Для того щоб продовжити роботу над аналізом методів класифікації текстів автором даного дослідження буде зібраний власний дата-сет невеликого обсягу розміром в декілька сотень текстів. Частина текстів буде зібрана з поштових ящиків і смс-повідомлень автора дослідження і його рідних. У разі недостатнього обсягу датасету нами буде використаний переклад частики англійського датасету українською мовою. Оскільки мета повідомлень різною мовою приблизно однакова – повідомити небажані новини або спробувати обманути і виманити дані – знехтуємо тим, що оригінал повідомлень було

написано англійською і припустимо, що переклад буде досить точний для досягнення мети дослідження.

1.2 Інструменти текстового аналізу української мови

Після отримання датасету спам текстів українською мовою потрібно виконати задачу спрощення тексту і його уніфікації. Сюди входить видалення стоп-слів: які не тільки не допоможуть в класифікацій, але можуть і заважати (наприклад і, у, в, на, тощо.), стеммінг – приведення різних форм слова до єдиного виду, зазвичай до кореня слова (знижка, знижку, зниження буде приведено до вигляду «зниж» або «ниж» залежно від налаштування фільтрації префіксів), перевірка наявності слова в словнику для уникнення аналізу випадкових буквосполучень без сенсу.

Для виконання цих завдань можна використовувати сервіси з того ж сервісу kaggle.com, які були розроблені для англійської мови, однак це буде не оптимально через особливості української мови. Елементарним прикладом такої особливості можуть бути слова з апострофами, які, на відміну від англійських апострофів, можуть знаходитись посередині смислової частини слова (наприклад: дерев’яний). Для більш точного результату нам потрібні інструменти, які налаштовані на аналіз саме українського тексту.

Такі інструменти вже існують і ми використовуватимемо саме їх в рамках даного дослідження:

– токенізатор, Великий Електронний Словних Української мови (ВЕСУМ), корпуси, алгоритми, а також інструмент векторного представлення слів з сайту https://lang.org.ua/uk/ що є проектом групи «lang-uk» [13];

– GEC-корпус української мови, зібраний командою проекту Grammarly [14].

1.3 Постановка задач дослідження

Проведений аналіз показав, що проблема фільтрації спаму української мови на момент написання даного дослідження не є вирішеною, тому що відсутня підготовлена база даних уже класифікованих текстів на категорії «спам»/«не спам» а також відсутні готові аналоги спам-фільтрів з використанням машинного навчання. Проте доступні інструменти аналізу тексту які можна використати для побудови вищезазначеного програмного забезпечення фільтрації спаму.

Мовний аналіз текстів з метою їх подальшої класифікації є об’єктом дослідження. Методи класифікації текстів українською мовою є предметом дослідження.

Спам-фільтр є одним зі складних прикладів програмного забезпечення і вимагає застосування машинного навчання. В рамках дослідження ми розробимо нейронну мережу з вчителем.

Метою даної роботи є визначення ефективного методу класифікації текстів українською мовою на основі проведеного порівняльного аналізу методів класифікації і розробка вдосконаленого методу.

В рамках мети виділимо наступні задачі дослідження:

– проаналізувати ринок на предмет наявності доступних повноцінних аналогів або сервісів, що частково підійдуть для досягнення мети роботи;

– провести аналіз методів класифікації текстів;

– розробити програмне забезпечення спам-фільтру українською мови з використанням різних методів класифікації текстів;

– розробити модифікований метод класифікації текстів української мови на основі одного з розроблених у попередньому розділі;

– проаналізувати результат використання різних методів класифікації;

– підвести підсумки і визначити ефективній метод класифікації текстів українською мовою з метою спам-фільтрації.

Програмна реалізація буде виконана мовою Python з використанням сервісу Datalore від Jetbrains і бібліотеки nltk.

2 ДОСЛІДЖЕННЯ МЕТОДІВ КЛАСИФІКАЦІЇ ТЕКСТІВ

Проаналізуємо основні методи для класифікації текстів із сфери машинного навчання, розберемо їх особливості і оберемо два з них для порівняльного аналізу ефективності фільтрації спаму українською мовою.

Класифікація текстів - це визначення приналежності текстів до однієї із категорій, що були визначені заздалегідь. Методи класифікації текстів знаходяться на перетині двох наукових областей – інформаційного пошуку і машинного навчання[3]. В рамках даної роботи в першу чергу аналізуватимемо саме автоматичну класифікацію текстів, адже саме вона потрібна для реалізації програмного забезпечення спам-фільтру українською мови. Для успішного функціонування автоматична класифікація потребує розробки на основі машинного навчання, а саме – створення нейронної мережі з вчителем.

Нейронна мережа називається мережою з вчителем, якщо модель алгоритму була натренована на заздалегідь класифікованих екземплярах даних. У випадку створення програмного забезпечення спам-фільтру потрібна наявність бази даних листів чи текстів які вже класифіковані за категорією «спам»/«не спам». Автоматична класифікація складається з двох основних частин: тренування моделі і тестування (або використання)моделі.

Навчання моделі – це опис множини заздалегідь визначених категорій і представлення тренувального набору елементів з уже визначеною категорією. Модель алгоритму побудує власні правила класифікації на основі певної тренувальної вибірки даних, які ми їй запропонуємо. Саме крок підбору даних для тренування моделі є найбільш важливим для правильного функціонування моделі, адже неправильно класифіковані дані або їх дуже мала кількість можуть спричинити збої в роботі моделі.

Використання моделі полягає в визначенні категорій нових, раніше невідомих даних. Модель вважається успішною, якщо кількість правильно здійснених класифікацій перевищує 50%. Формально опишемо задачу

класифікації: вважатимемо, що потрібно класифікувати за допомогою алгоритму деяку множину текстів T= $\{t_i \dots t_n\}$. Вся множина текстів розбивається на непересічні підмножини категорій: $C = \{C_i \dots C_n\}$ [21].

Невідома цільова функція $F: T\ x\ C \rightarrow \{0,1\}$ задається формулою:

$$F\left(t_j, c_i\right) = \begin{cases} 0, \text{якщо } t_l \in c_i \text{ не істина} \\ 1, \text{якщо } t_l \in c_i \text{ істина} \end{cases}$$

Завдання класифікації можна оформити як необхідність визначити множину, до якої належить певний текст, або необхідність знайти функцію F', яка буде максимально наближена до ідеалу F.

Кожен елемент вибірки під назвою t володіє певним набором смислових ознак

t = $\{X_i\}$. Ці ознаки використовуються в алгоритмі класифікації для визначення текстів, що з більшою долею ймовірності можуть бути визначені, як належні до заданого класу[4].

Задача класифікації тексту може бути виконана декількома методами, а не одним. Перші ніж обрати метод, потрібно зважити переваги і недоліки кожного з них у виконанні конкретної задачі.

Більшість методів класифікації засновані на припущенні, що текст, який належить певної категорії, має набір однакових вагомих факторів і ознак. Присутність ознак, які є визначними для певної категорії, є достатньою причиною для класифікації тексту як приналежного до даної категорії. Те ж саме с відсутністю ключових ознак.

Таким чином, для кожної категорії повинна бути множина ознак:

$$F(C) = \bigcup (c_r = \{f_1, f_2 \dots f_n\})$$

Дана множина ознак часто носить назву словника. Така назва обумовлена наявністю лексем, тобто слів або словосполучень, які є визначними для певної категорії. Кожен текст також має ознаки, що дозволяють розподілити його до однієї із категорій з певним ступенем ймовірності:

$$F(t) = \{f_1^i, f_2^i \dots f_n^i\}$$

На відміну вид класифікації об’єктів у сфері Data Mining, що можуть бути характеризовані набором атрибутів, множина ознак усіх тренувальних текстів обов’язково співпадає з множиною смислових ознак, які визначають категорію, тобто:

$$F(C) = F(T) = \bigcup F(t_i)$$

Кінцевий висновок щодо приналежності листа l до категорії k відбувається на основі перетину:

$$F(l_i) \bigcup F(k_r)$$

Головною задачею методів для класифікації текстів є визначення тих ознак сукупності текстів, які дозволять сформулювати правила для майбутнього прийняття рішень щодо належності тексту до певного класу

Головними методами машинного навчання, які можуть бути використані для виконання задачі класифікації текстів, є наступні:

– класифікація через дерево рішень;
– наївна класифікація Байєса;
– класифікація за методом опорних векторів;
– класифікація за методом найближчого сусіда;

– класифікація штучними нейронними мережами.

2.1 Класифікація за допомогою дерева рішень

Алгоритм дерев рішень відрізняється від інших алгоритмів машинного навчання унікальною «вузловою» або «гілковою» структурою аналіз даних. На основі певних змінних і ознак, характерних конкретному датасету створюються залежності характеру “якщо-то” для класифікації документів у наведеному датасеті [5].

Визначення приналежності тексту до одного із заздалегідь визначених класів засноване на сукупності поступових ієрархічних відповідей на сформовані запитання. Прикладом запитання може бути наступне: «чи більше значення кількості прояву суттєвої ознаки о за визначений поріг р?». За умов позитивної чи негативної відповіді відбувається перенаправлення алгоритму до наступного запитання, «вузла», на дереві рішень. Критично важливо відмітити, що вузли будуть різними залежно від відповіді, і повернення назад в ієрархії запитань практично неможливе.

Побудова дерева рішень вимагає набагато більше підготовки, ніж інші алгоритми машинного навчання, адже потрібно чітко визначити запитання для кожного «вузла» дерева. За умов наявності невірного аналізу відповідей запитання, або не релевантного запитання, алгоритм приречений на провал.

Для вирішення цієї проблеми існує ряд алгоритмів для автоматичної побудови дерев рішень на основі визначення характерних ознак, приналежність текстів до яких і буде «вузлами» дерева[6]. Одним з найпопулярніших алгоритмів є CLS (Concept Learning System).

Для його функціонування перш за все потрібно визначити змінну, яка вважається найважливішою для успішної класифікації. За допомогою класифікації по цій категорії виділяється наступна підмножина прикладів з

найбільш вагомими ознаками. Ієрархічний поділ відбувається доти, доки в утвореній підмножині залишаться лише елементи, що належать до однієї категорії, яка і буде однією із заключних категорій дерева рішень. Утворена структура нагадує дерево, адже починається з одного вузла і закінчується потенційною безліччю фінальних категорій, за що алгоритм і отримав свою назву.

Для того, щоб визначити початковий найважливіший критерій потрібно визначити критерій інформаційної значущості елементу класу. Часто цей процес набагато складніший ніж просто знаходження найвагомішого елементу і подальша класифікація. Окрім цього використовуються вирізання несуттєвих гілок дерева і його супутні перетворення.

Слід відмітити, що логічна простота алгоритму дерева рішень приховує потенційну проблему неможливості оцінки вузлового переходу як позитивного чи негативного для даної конкретної задачі. Так, за умови наявності більшої кількості негативних переходів, ніж позитивних, алгоритм може зробити хибне передбачення.

Одним із варіантів даного методу є так званий метод випадкового лісу (Random Forest), який працює наступним чином:

– вибирається підвибірка навчальної вибірки певного розміру - по ній будується дерево (для кожного дерева – своя підвибірка);

– для побудови кожного розщеплення в дереві переглядаємо максимальну кількість випадкових ознак (для кожного нового розщеплення – свої випадкові ознаки);

– вибираємо найкращі ознаки і розщеплення за ним (по заздалегідь заданому критерію). Дерево будується, як правило, до вичерпання вибірки (доки не залишаться представники тільки одного класу), але в сучасних реалізаціях є параметри, які обмежують висоту дерева, число об'єктів в листі і число об'єктів підвибірки, при яких проводиться розщеплення.

Метод випадкового лісу успішно використовувався в роботі [18] для обробки часових рядів біомедицини.

2.2 Метод наївної класифікації Байєса

Специфіка задачі класифікації текстів така, що саме наївний класифікатор Байєса часто успішно використовується для її вирішення. До головних задач цієї категорії можемо віднести визначення тональності документа, фільтрацію спаму, тощо.

Є декілька різновидів класифікаторів текстів за Байєсом: мультиноміальний, класифікатор за Гаусом, класифікатор за Берноулі, тощо. Кожен з них має певні особливості, однак все ще належить до сімейства методів класифікації за Байєсом.

Опишемо головний принцип, що об'єднує усі різновиди Байєсівських класифікаторів, наступною формулою:

$$\mathrm{C} = \arg max\, P(C|o_1, o_2, \dots o_n) = \arg\max P(c)\Pi P(o_i|c)$$

де С – сукупність класів, а $o_1, o_2, \dots o_n$– сукупність ознак. За умов відомого набору незалежних ознак $o_1, o_2, \dots o_n$ класифікація полягає у визначенні максимального значення аргументу[20]. При цьому:

$$P(c)\Pi P(o_i|c) = P(C)P(P(o_1|c)\, P(o_2|c) \dots\, P(o_n|c)$$

Обчислення ймовірності класу P(C) при відомих ознаках $o_1, o_2, \dots o_n$зводиться до наступного:

$$P(C|o_1, o_2, \dots o_n) = \sum o_1, o_2, \dots o_n) + 1/\sum(C|A) + \sum A,$$

де А – набір відомих ознак, отриманих при навчанні класифікатора.

Формула класифікації тексту зводиться до наступного:

$$C(T) = \max \sum (s_1, s_2, \dots s_n | C\ ),$$

де Т – текст, що класифікується, а $s_1, s_2, \dots s_n$– набір речень тексту. Максимальне значення суми коефіцієнтів належності речень визначає належність тексту до однієї із заздалегідь визначених категорій.

Метод наївної класифікації за Байєсом заснований на використанні моделі ймовірності, згідно якої закономірності появи слів у документах заздалегідь визначених тренувальних класів використовується для подальшого розподілення на категорії [7].

Наївний класифікатор Байєса привабливий простотою своєї реалізації і відносно низькими витрати на обчислювальні потужності під час етапу тренування моделі. Недоліком даного алгоритму є недосконала робота в ситуаціях, коли межі категорій визначені нечітко. В такій ситуації робота алгоритму ускладнена і він не може достеменно точно дати відповідь на приналежність до тієї чи іншої категорії. Якщо границі категорій визначені чітко то наївний класифікатор Байєса є оптимальним. В реальному житті не завжди є можливість встановити чіткі границі категорій, що ускладнює використання даного методу. Проте він є досить ефективним в задачах класифікації текстів на чіткі підгрупи.

2.3 Метод опорних векторів

Метод опорних векторів (SVM - Support Vector Machine) полягає в виділенні позитивних і негативних прикладів в багатовимірному просторі функції з векторами, які представлені за допомогою тренувальних текстів.

Даний метод був розроблений в 1995 році В. Вапником, проте для класифікації текстів вперше використаний Торстеном Джохімсом. Схоже до алгоритму класифікації Байєса, даний метод спроектовано для вирішення задачі класифікації певних об’єктів на дві категорії. Метод опорних векторів швидко набув широкої популярності в рамках виконання задачі класифікації текстів завдяки своїй ефективності.

Головною особливістю даного методу є підхід заснований на мінімізації ризику похибки для визначення закономірностей належності до певної категорії.

Перевагою методу опорних векторів є ефективність вірного прогнозування належності до класів - вона одна з найбільших порівняно з іншими методами машинного навчання.

Недоліком методу можна назвати надзвичайно низьку швидкість роботи, пов’язану із значними витратами оперативної пам’яті і, відповідно, часу, за умов її не достатку. Проте якщо є можливість використовувати великі об’єми ресурсів цей метод важно недооцінити.

Якщо відкинути вищевказані недоліки і оцінити виключно якість результату, то метод опорних векторів можна назвати зразковим для виконання задачі класифікації[8].

Даний метод успішно використовується в багатьох сферах життя. Наприклад автори роботи [16] використовують його для знаходження інформаційних риноманометричних сигналів.

2.4 Метод k-найближчого сусіда

Даний метод є одним з найдавніших і, відповідно, найкраще вивчених в сфері точних алгоритмів. Він успішно використовувався ще в середині двадцятого століття для задачі дискримінантного аналізу. Метод k-

найближчого сусіда добре підходить ля задачі класифікації текстів і показує високі результати ефективності в цьому напрямку [9].

Ядром методу є нескладна ідея, суть якої полягає в локалізації колекції, що має найбільш схожих ознак з текстом, який аналізується, і присвоєння їй певної смислової категорії. Інформація про категоріальну приналежність тексту використовується для подальшої категоризації невідомих документів.

Для знаходження категорії, яка визначає текст t, алгоритм порівнює текст з усіма іншими текстами представленого датасету. Для кожного тексту t_t з тренувального датасету буде визначено косинус кута між векторами смислових ознак:

$$p(t, t_t) = \cos(t, t_t)$$

Наступним кроком є пошук n найближчих текстів, які найближчі до визначеної категорії за сукупністю ознак. Підрахунок коректності приналежності до категорії описується наступною формулою:

$$s(c_j, t) = \sum \cos(t, t_t)$$

За умови що формула коректності повернула результат, вищий заздалегідь визначеної норми, текст вважається приналежним до класу, що аналізується. Змінна n традиційно обирається в проміжку від 0 до 100.

Простота алгоритму полягає в тому, що за умов необхідності визначення лише одного класу достатньо обрати категорію з максимальною кількістю співпадінь серед зазначених n-текстів. Якщо ж потрібно визначити більше однієї категорії і один текст може належати до різних категорій, потрібно задати поріг який вважатиметься достатнім для присвоєння тексту класу з найбільш коректним результатом

Слід відмітити таку особливість алгоритму як відсутність етапу тренування. Достатньо просто представити набір текстів і на їх основі динамічно визначити категорії. Тобто тренувальна функція класифікації відсутня.

Це може бути зручно за умов динамічного датасету, який весь час поповнюється новими даними – адже не потрібно кожен раз перенавчати алгоритм на новому датасеті.

Класичний алгоритм пропонує порівнювати аналізований документ з усіма документами з навчальної вибірки і тому головний недолік описаного методу полягає в тривалості часу роботи класифікатора [10].

2.5 Штучні нейронні мережі

Нейронні мережі штучного характеру вивчаються експертами з штучного інтелекту уже, щонайменше, протягом тридцяти п’яти років. За цей час вони набули значного розвитку. Свою назву нейронні мережі отримали через подібність до структури нервових клітин людини.

Головною особливістю нейронних мереж є їх здатність динамічно адаптуватись до нових станів в процесі роботи – саме так як себе мала б вести справжня нейронна система. Динамічній адаптації підлягає зміна внутрішніх станів, внутрішніх зв’язків текстів і визначених категорій.

Перш ніж приступати до класифікації текстів у нейронних мережах потрібно визначити коректну і раціональну топологію нейронних зв’язків. До основних розповсюджених топологій нейронних мереж належать це одно- і багатошарові персептрони, мережа Кохонена, нейромережевий Гаусів класифікатор, мережа вбудованого розповсюдження, каскадна мережа [11].

Дані топології успішно використовуються в задачах лінійного і нелінійного програмування. Однак слід відмітити додаткову складність

формалізації залежностей класифікатора через складність структури самого аналізатора. Через великий масштаб структури і обмежені ресурси для обрахування результатів це є складною задачею.

2.6 Постановка практичної задачі і вибір методу

Предметом даного дослідження є методи класифікації текстів українською мовою. Поставимо більш чіткі цілі, яких намагатимемось досягнути в процесі розробки практичної частини завдання. Оскільки головною метою дослідження є покращення ситуації на ринку програмного забезпечення для роботи з українськими текстами, спробуємо реалізувати метод класифікації текстів, який досягне більшої ефективності, ніж наразі доступний метод класифікації Байєса в бібліотеці nltk мовою програмування Python.

Метод класифікації Байєса є одним з фундаментальних методів класифікації, який часто використовується в дослідженнях, як метод, з яким буде відбуватись порівняння. Проте цей метод має не одну реалізацію, а декілька, які дещо відрізняються за принципами роботи, і, відповідно, ефективністю: мультиноміальний, класифікатор за Гаусом, класифікатор за Берноулі, тощо.

В даній роботі порівнюватимемо між собою найпримітивнішу реалізацію наївного класифікатору і мультиноміальний класифікатор Байєса.

За основу візьмемо мультиноміальний метод з бібліотеки nltk, натренуємо його на листи української мови і визначимо його ефективність передбачення. Після цього розробимо власний метод класифікації Байєса, який буде слідувати основним принципам Байєсівського класифікатора, проте, на відміну від методу з бібліотеки дана реалізація дозволить спостерігати аналіз тексту української мови на різних етапах його виконання. Для простоти

називатимемо створений метод просто «наївним класифікатором Байєса», який протиставлятиметься бібліотечному мультиноміальному методу.

Після реалізації наївного методу класифікації знайдемо в ньому ваду, яка напряму пов’язана з недосконалою роботою з українськими текстами, і спробуємо її усунути, або хоча б покращити ситуацію з обробкою текстів, що містять вищевказану ваду.

3 НАВЧАННЯ МОДЕЛЕЙ КЛАСИФКАЦІЇ БАЙЄСА НА ЗІБРАНОМУ ДАТАСЕТІ

3.1 Збір дата сету

Для того щоб продовжити роботу над аналізом методів класифікації текстів автором даного дослідження було зібрано власний дата-сет відносно невеликого обсягу розміром в декілька сотень текстів. Частина текстів була зібрана з поштових ящиків і смс-повідомлень автора дослідження. Оскільки мета повідомлень різною мовою приблизно однакова – повідомити небажані новини або спробувати обманути і виманити дані – знехтуємо тим, що смс-повідомлення і електронні листи часто дещо відрізняються об’ємами тексту, що передається.

Метадані повідомлення в рамках цього дослідження також ігноруватимуться. Під метаданими (даними про дані) мається на увазі додаткова інформація: поштова скринька відправника, сайт відправника, рейтинг доброчесності відправника, заголовок листу, тощо. Предметом дослідження буде саме тіло листа українською мовою.

Зібраний датасет обсягом 356 текстових повідомлень, що включають 30% спам повідомлень і 70% звичайних повідомлень було оформлено за зразком, який надано у таблиці 1.

Спам повідомлення отримали маркування «spam», а звичайні повідомлення – «ham». У сфері спам фільтрів досить давно ввели поняття «ham» як коротке і співзвучне зі спамом для маркування повідомлень, що не являються спамом. Адже кожного разу використовувати «не спам» досить не зручно. Дотримуватимемось загальноприйнятої тенденції в цьому дослідженні.

Спам повідомлення отримали маркування «spam», а звичайні повідомлення – «ham». У сфері спам фільтрів досить давно ввели поняття

«ham» як коротке і співзвучне зі спамом для маркування повідомлень, що не являються спамом. Адже кожного разу використовувати «не спам» досить не зручно. Дотримуватимемось загальноприйнятої тенденції в цьому дослідженні.

Таблиця 1 – Датасет повідомлень українською

Label	Body
spam	МЕГА РОЗПРОДАЖ! Знижки -70%+додатково -20%: http://bit.ly/**
ham	Добрий день, гадаю, в такому вигляді можна прийняти звіт

Існують більш складні динамічні способи підготовки даних – у роботі [19] був використаний спосіб розмитого розбиття (fuzzy partitioning) який поділяє датасет на класи з нечіткими границями. Однак нам достатньо більш простого рівня підготовки тексту для цього дослідження.

3.2 Реалізація методу мультиноміальної класифікації Байєса

Для практичної реалізації методу Байєса використаємо сервіс Datalore від компанії Jetbrains, який надає безкоштовний доступ до Jupiter ноутбуків, що використовуються в сфері машинного навчання. Jupiter Notebook – це командна оболонка для інтерактивних розрахунків. Вони потрібні через те, що самостійно реалізувати оточення необхідної потужності розрахунків досить важко.

Спочатку спробуємо використати уже розроблений мультиноміальний метод Байєса з бібліотеки nltk.

Першим кроком розіб’ємо датасет на частину для навчання і частину для перевірки, де 80% усіх повідомлень будуть використані для навчання. Розбивання відбувається випадковим чином.

Після цього навчаємо модель на основі тренувального датасету і протестуємо.

```
▷ 0.1s
messages_bow = CountVectorizer(analyzer=process_text).fit_transform(df['BODY'])
X_train, X_test, y_train, y_test = train_test_split(
    messages_bow, df['SPAM'], test_size = 0.20, random_state = 0
)

classifier = MultinomialNB()
classifier.fit(X_train, y_train)

pred = classifier.predict(X_train)
print(classification_report(y_train ,pred ))
print('Confusion Matrix: \n',confusion_matrix(y_train,pred))
print()
print('Accuracy: ', accuracy_score(y_train,pred))
```

Рисунок 1 – Використання мультиноміального методу Байєса з nltk бібліотеки

Як видно на Рисунку 4, для реалізації мультиноміального методу Байєса з бібліотеки nltk на мові програмування Python потрібно мінімум додаткової обробки тексту, адже метод є одразу готовим до роботи і включає в себе наступні кроки:

– прибираємо дублікати повідомлень;

– прибираємо пунктуацію;

– трансформуємо всі літери в нижній регістр;

– розбиваємо речення на слова;

– прибираємо стоп-слова (слова, які не мають важливого значення, коли відірвані від контексту).

– вираховуємо вагу кожного слова відносно категорії (залежить від частоти використання в певній категорії в навчальному датасеті);

– перевіряємо результат на тестовому датасеті.

В даній роботі буде вимірюватись ефективність роботи класифікаторів за Байєсом за шкалою від нуля до одиниці, де 1 це 100% вірно класифікованих повідомлень, а 0 – жодного вірно класифікованого повідомлення.

У результаті фільтрування спаму отримуємо результат в 0.86, що означає, що 86% повідомлень було фільтровано коректно. Це дуже великий показник ефективності, який частково залежить від невеликого об’єму датасету і того, що весь датасет з поштової скриньки однієї людини. В таких умовах відслідкувати залежності для алгоритму має бути досить нескладно. Результат наведено на рисунку 2.

```
              precision    recall  f1-score   support

           0       0.76      1.00      0.87        13
           1       1.00      0.75      0.86        16

    accuracy                           0.86        29
   macro avg       0.88      0.88      0.86        29
weighted avg       0.89      0.86      0.86        29

Confusion Matrix:
 [[13  0]
 [ 4 12]]

Accuracy:  0.8620689655172413
```

Рисунок 2 – Ефективність мультиноміального методу Байєса

Звернемо увагу на те, що всі 100% листів не були класифіковані вірно. Проаналізуймо помилкові приклади і спробуємо знайти якусь закономірність, специфічну українській мові.

Серед хибно класифікованих повідомлень знаходимо декілька прикладів з словами, що містять апострофи і висуваємо теорію, що саме це може бути причиною хибного висновку.

Таблиця 2 – Хибно класифіковані повідомлення

Label	Body	Prediction
spam	Спекотна п’ятниця! 50% на все. Тільки 3 дні.	ham
spam	М'язиста пропозиція! Дев'ять днів тренування за кошт п'яти. ФК Плутон на Широнінців.	ham
ham	Припиню зв'язок з сім'єю на декілька днів доки не з'ясую свою ситуацію із здоров'ям."	spam

Пояснимо причину такого висновку на першому прикладі, відслідкувавши трансформацію повідомлення протягом семи описаних кроків трансформації тексту.

– прибираємо пунктуацію:

«Спекотна п ятниця 50 на все Тільки 3 дні»

– трансформуємо всі літери в нижній регістр:

«спекотна п ятниця 50 на все тільки 3 дні»

– розбиваємо речення на слова:

['спекотна', 'п', 'ятниця', '50', 'на', 'все', 'тільки', '3', 'дні']

– прибираємо стоп-слова (слова, які не мають важливого значення, коли відірвані від контексту):

['спекотна', 'все', 'тільки', 'дні']

– вираховуємо вагу кожного слова відносно категорії (залежить від частоти використання в певній категорії в навчальному датасеті):

Таблиця 3 – Розбір прикладу речення на оцінені слова

Слово	Кількість використань в спамі	Кількість використань не в спамі
'спекотна'	0	0
'все'	21	23
'тільки'	6	4
'дні'	8	10

Кількість слів у реченні, яке ми розглядаємо, невелика: після видалення слова «п’ятниця» як беззмістовного залишилось лише чотири слова. Наступним кроком алгоритм оцінив кількість вживань даних слів у повідомленнях, промаркованих як «спам» і «не спам» у тренувальному датасеті. Як бачимо, кількість вживань у повідомленнях з позитивною конотацією переважає у двох з чотирьох слів, а одне із слів не вживалось в жодному з повідомлень обраного тренувального датасету.

Це лише початкова обробка тексту після якої слідує власне обробка даних алгоритмом Байєса, однак цих даних вже достатньо, щоб зрозуміти, що алгоритму надається некоректна інформація, що веде до хибного висновку.

Як уже було зазначено, слово «п’ятниця» було поділене на «п» і «ятниця», тому що стандартний механізм поділу, розроблений з огляду на англійську мову, замінює апостроф на пробіл, і далі ділить по ньому.

При фінальній класифікації текстів вага кожного слова в категоріях «spam» і «ham» дуже важлива і може схилити класифікатор на одну чи іншу сторону. Якби слово «п’ятниця» було розпізнане вірно, воно мало б високий негативний коефіцієнт ймовірності бути в спамі, тому що лише серед підібраного невеликого датасету було декілька повідомлень про «Чорну п’ятницю» і супутні знижки. Оскільки слово розпізнане не було, то його коефіцієнтом знехтували, а слова, що залишились, були недостатньо переконливими для вірної класифікації.

Схожа ситуація і з іншими прикладами – багато з них були класифіковані невірним чином через те, що аналізатор зміг обробити лише частину слів через невірний поділ речення на слова.

Отже, після аналізу речення *«Спекотна п’ятниця! 50% на все. Тільки 3 дні.»* зробимо висновок, що наявність апострофів в словах української мови є перепоною для роботи алгоритму класифікації Байєса. Серед хибно класифікованих речень більшість містять апострофи, які, якщо і не є головною причиною хибного висновку, є причиною неможливості коректно

класифікувати конкретні слова. Це, в свою чергу, призводить до зменшення кількості слів, які могли б схилити алгоритм на одну із сторін класифікації – «спам»/«не спам».

3.3 Реалізація методу наївної класифікації Байєса

Метод мультиноміальної класифікації Байєса показав результат в 86% ефективності класифікації повідомлень на «spam» і «ham». Оскільки саме варіації методу Байєса в вирішенні задачі класифікації повідомлень на дві основні категорії вважаються найбільш ефективними, порівняємо метод мультиноміальної класифікації з методом наївної класифікації Байєса.

Для подальшого аналізу було реалізовано метод наївної класифікації Байєса власними силами, для того щоб була можливість модифікувати його частини з метою усунення проблеми виявлення слів з апострофами.

Першим кроком нам потрібно завантажити тренувальну базу до алгоритму і виконати приведення до нижнього регістру і прибирання пунктуації. В першій реалізації методу наївної класифікації Байєса жодної оптимізації по роботі з апострофами не очікується. Спочатку порівняймо який із варіантів методу краще впорається з задачею класифікації тексту на «spam» і «ham». Реалізацію вищевказаних кроків разом з прикладами результатів наведено на рисунку 3.

Одразу після виконання частини програмного коду сервіс Datalore, яким ми скористались для реалізації програмного коду, дозволяє вивести частину результату для візуалізації. У наведеному на рисунку 3 прикладі було виведено три останні екземпляри листів з тренувального датасету. Лише серед початків речень трьох екземплярів видно проблемне місце зі словом «кур’єр», яке було трансформовано в «кур» і «єр» і, відповідно, не буде вірно проаналізоване з точки зору використання в листах спаму і бажаних листах.

```
▶ 11.0s
import pandas as pd

spamBase = pd.read_csv('SpamBase.csv', sep=',',
header=None, names=['label', 'body'])
data_randomized = spamBase.sample(frac=1, random_state=1)
training_test_index = round(len(data_randomized) * 0.8)
training_set = data_randomized[:training_test_index].reset_index(drop=True)
test_set = data_randomized[training_test_index:].reset_index(drop=True)
training_set['body'] = training_set['body'].str.replace('\W', ' ')
training_set['body'] = training_set['body'].str.lower()
training_set.tail(3)
```

```
<ipython-input-37-b6fd70b17653>:5: FutureWarning: The default value of regex will c
  training_set['body'] = training_set['body'].str.replace('\W', ' ')
```

	label	body
280	ham	є підозра шо ловити мене треба ...
281	spam	володимире не витрачайте кошт...
282	ham	вітаємо кур єр в дорозі телефон ...

Рисунок 3 – Реалізація первинної обробки тексту в методі Байєса

Наступним кроком реалізації алгоритму є розбиття речення на слова і підрахунок кількості використання слів в конкретних реченнях включаючи асоціацію з відповідним типом повідомлення: «spam» і «ham». Результатом виконання цієї частини програми буде таблиця, зразок якої наведено у таблиці 4.

Таблиця 4 – Розбір прикладу речення на оцінені слова

label	body	підозра	єр	витрачайте
ham	[«є», «підозра», «що», «ловити» …]	1	0	0
spam	[«володимире», «не», «витрачайте», «кошти» …]	0	0	1
ham	[«вітаємо», «кур», «єр», «в», «дорозі» …]	0	1	0

В таблиці 4, яка представлена у вигляді матриці, наведений приклад на основі трьох колонок, а отже трьох унікальних слів. Повна кількість колонок в таблиці 4 дорівнює кількості унікальних слів в повідомленнях. Значення нуля чи одиниці означає відповідно відсутність і використання слова в повідомлені, що вказане в даному рядку.

На даному етапі можна вважати, що навчання моделі виконано, за умови що вищевказану класифікацію окремих слів було пройдено на датасеті великого розміру. Як уже було зазначено в роботі, оскільки готового класифікованого датасету спам повідомлень українською мовою нами знайдено не було, обмежимось кількістю повідомлень в 364 одиниці, які були зібрані власноруч.

Для навчання моделі використаємо приблизно 80% датасету, що складає 293 повідомлення. Повідомлення, що залишились, розміром в 71 одиницю, будуть використані для перевірки тренованої моделі.

```
def classify_test_set(message):
    message = re.sub("\W", ' ', message)
    message = message.lower().split()

    spam_probability = default_spam_probability
    ham_probability = default_ham_probability

    for word in message:
        if word in parameters_spam:
            spam_probability *= parameters_spam[word]

        if word in parameters_ham:
            ham_probability *= parameters_ham[word]

    if ham_probability > spam_probability:
        return 'ham'
    elif spam_probability > ham_probability:
        return 'spam'
    else:
        return 'unknown'

test_set['predicted'] = test_set['body'].apply(classify_test_set)
test_set.head()
```

Рисунок 4 – Реалізація функції класифікації за Байєсом

Спробуємо зробити висновок про ціле повідомлення з тестового датасету на основі менших висновків про окремі слова з тренувального датасету. Повідомлення вважатиметься спамом, якщо кількість окремих слів в ньому, які визначені як ймовірний «spam», перевищуватиме кількість слів, які визначені як «ham». Реалізацію вищевказаного методу, який в циклі перебирає всі слова в повідомленні і робить висновок, зображено на рисунку 4.

Результат виконання цього методу класифікації виведемо у вигляді таблиці з трьох колонок: тип повідомлення, тіло повідомлення, прогноз повідомлення. Приклад було наведено у таблиці 2.

Результат роботи наївного класифікатора Байєса на тому ж самому датасеті, який було використано в реалізації мультиноміального методу, наведено на рисунку 5.

```
Correct: 60
Incorrect: 11
Accuracy: 0.8450704225352113
```

Рисунок 5 – Ефективність наївного класифікатору за Байєсом

Як бачимо, ефективність наївного класифікатору за Байєсом складає 84%, тобто 84% повідомлень були класифіковані вірно. Це на 2% менше ніж за мультиноміальним методом класифікації Байєса. Невеликий відсоток різниці зумовлений, перш за все, відносно невеликим розміром тестового датасету. Наявність різниці дозволяє нам припустити, що мультиноміальний метод класифікації Байєса є більш ефективним для виконання задачі класифікації листів на «spam» і «ham» за умови наявності більшого тренувального датасету. Проте і мультиноміальний і наївний методи підходять для успішного виконання задачі класифікації спаму.

4 СТАТИСТИЧНЕ ПОРІВНЯННЯ ЕФЕКТИВНОСТІ МЕТОДІВ КЛАСИФІКАЦІЇ

4.1 Порівняння наївного і мільтиноміального методів класифікації

Оскільки кожен з методів класифікації було використано лише один раз, ми не можемо бути достеменно впевнені в коректності наведених результатів. Є декілька основних підходів до поділу датасету на тестовий і тренувальний під час навчання моделі:

а) свідомо використовуємо один і той самий набір тренувальних текстів. Не міняємо їх між використаннями алгоритму.

Перевага: дозволяє уникнути непередбачуваності тренувальних даних після першої обробки.

Недолік: за умов неправильно підібраного тренувального датасету, модель приречена на провал, адже у неї немає шансу перевчитись.

б) випадково розбиваємо набір даних на тренувальні і тестові під час кожного навчання моделі у розробці.

Перевага: усуває проблему першого підходу – модель може перенавчитись, якщо в одному з попередніх використань було підібрано невірний датасет.

Недолік: випадковість результату ефективності класифікації викликає недовіру до одного отриманого результату, який може бути «випадково» успішний.

Під час попередніх вимірів ефективності мультиноміального і наївного класифікаторів за Байєсом був використаний підхід б) з випадковістю розподілення даних на тестові і тренувальні. Для того, щоб впевнитись в отриманих результатах і усунути або хоча б мінімізувати вплив випадковості результатів, збережемо результат виконання кожного з алгоритмів щонайменше

десять разів і знайдемо середню ефективність. Результат навчання моделей наведено у таблиці 5.

Таблиця 5 – Середнє арифметичне ефективності мультиноміального і наївного методів класифікації

Спроба	Мультиноміальний	Наївний
1	0,862	0,845
2	0,794	0,816
3	0,884	0,810
4	0,826	0,873
5	0,869	0,788
6	0,783	0,845
7	0,873	0,802
8	0,836	0,760
9	0,921	0,901
10	0,894	0,830
Середнє	0,854	0,827

Середнє арифметичне результатів ефективності використання мультиноміального методу класифікації Байєса складає 0,854, а середнє арифметичне результатів методу наївної класифікації Байєса 0,827. Різниця ефективності збереглась, і навіть збільшилась на користь мультиноміального методу Байєса з 0,017 до 0,027.

На основі проведеного дослідження можемо зробити остаточний висновок, що мультиноміальний метод класифікації Байєса є більш ефективним ніж метод наївної класифікації для задачі класифікації спаму листів українською мовою.

Під час дослідження обох методів класифікації було помічено проблему розподілення слів речення на слова у зв’язку з українськими апострофами, які, на відміну від англійських апострофів, є невід’ємною частиною слова.

З метою подальшого вдосконалення способів вирішення задачі класифікації текстів з метою фільтрування спаму українською мовою внесемо модифікацію, пов’язану з вирішенням проблеми апострофів, в метод наївної класифікації, який вже було реалізовано.

4.2 Реалізація модифікованого наївного методу класифікації Байєса

Спробуємо модифікувати частину коду, яка відповідає за розбиття речення на слова. Для цього потрібно змінити налаштування пунктуаційних символів, щоб апостроф перестав вважатись символом пунктуації і став вважатись частиною слова. Також необхідно впевнитись, що внесені зміни не зменшують ефективність інших частин алгоритму.

Переглянемо символьну таблицю і визначимо символи пунктуації, які необхідно прибрати з речення українською мовою. Використаний програмний код видалення пунктуації:

```
training_set['body'].str.replace("[!\"#$%&()*+,-./:;<=>?@[\]^_{|}~]", ' ')
```

Наступним кроком даного дослідження буде використання модифікованої частини програмного коду для роботи з пунктуаційними символами в методі наївної класифікації Байєса.

Ось що отримаємо при розбитті речень на слова після модифікації:

– ['на', 'ваше', 'ім’я', 'надійшли', 'кошти', 'для', 'отримання', 'дзвоніть', '080****305', 'код', '22'];

– ['спекотна', 'п’ятниця', '50', 'на', 'все', 'тільки', '3', 'дні'].

Як бачимо, слова з апострофами тепер сприймаються вірно і будуть мати коректний коефіцієнт ймовірності спаму. Це особливо суттєво в невеликих повідомленнях, де може бути лише декілька навантажених важливим значенням слів. Якщо якусь частину з них просто прибрати так само як «стоп-слова» то класифікація найбільш ймовірно буде проведена хибно.

Виміряємо ефективність модифікованого методу на тому ж датасеті, що й класичні методи мультиноміальної та наївної класифікації. Результат наведено на рисунку 6.

```
Correct: 65
Incorrect: 6
Accuracy: 0.9154929577464789
```

Рисунок 6 – Ефективність Модифікованого Методу Байєса

Як бачимо, проста модифікація до однієї частини методу класифікації Байєса внесла зміни в фінальний прогноз і дозволила вірно класифікувати слова, специфічні для української мови. Відзначимо, що результат модифікованого методу наївної класифікації вищий, ніж результат мультиноміального класифікатору.

Для повноти аналізу проведемо статистичне дослідження ефективності модифікованого методу і визначимо середнє арифметичне ефективності за 10 тренувань алгоритму. Результат наведено у таблиці 6.

Середнє арифметичне ефективності класифікованого спаму листів українською мови модифікованого методу наївної класифікації Байєса складає 0,883. Це на 0,561 або 5,6% більше, ніж метод наївної класифікації до внесення змін.

Таблиця 6 – Середнє арифметичне ефективності мультиноміального, наївного і модифікованого наївного методів класифікації

Спроба	Мультиноміальний	Наївний	Модифікований Наївний
1	0,862	0,845	0,915
2	0,794	0,816	0,901
3	0,884	0,810	0,830
4	0,826	0,873	0,915
5	0,869	0,788	0,845
6	0,783	0,845	0,887
7	0,873	0,802	0,920
8	0,836	0,760	0,887
9	0,921	0,901	0,901
10	0,894	0,830	0,830
Середнє	0,854	0,827	0,883

Отже, нам вдалось підвищити ефективність наївного класифікатору Байєса для задачі класифікації листів українською мовою на суттєвих 5,6%, що є дуже позитивним показником. Окрім цього, різниця з середнім арифметичним ефективності мультиноміального методу класифікації складає 0,289 або 2.9%.

Таким чином, нам вдалось не лише покращити результат наївного класифікатору за Байєсом, але й досягти ефективності вище такої у мультиноміального методу класифікації, який на початку дослідження вважався більш ефективним.

Для того, щоб остаточно впевнитись в ефективності внесених змін, протестуємо оригінальний і модифікований методи наївного класифікатора на цільовому тестовому наборі повідомлень з великою кількістю апострофів. Було відібрано 24 повідомлення з словами де є один і більше апострофів і протестовано виключно на цій вибірці. Результат приведено на рисунку 7.

```
Correct: 15
Incorrect: 9
Accuracy: 0.625
```

Рисунок 7 – Ефективність Методу Байєса на прикладах з апострофами

Як бачимо, початкова ефективність оригінального методу впала з 0.86 до 0.62. Наведемо результат тестування модифікованого методу на рисунку 8.

```
Correct: 19
Incorrect: 5
Accuracy: 0.7916666666666666
```

Рисунок 8 – Ефективність Модифікованого Методу Байєса на прикладах з апострофами

Модифікований метод наївного класифікатора за Байєсом досяг ефективності в 0,79 на цільовому тестовому датасеті з великою кількістю апострофів. Цей результат є меншим за 0,915, якого було досягнуто на великому тестовому датасеті, однак дане зниження ефективності є закономірним з огляду на невелику кількість в 24 приклади.

На даному датасеті ефективність оригінального наївного методу класифікації на 0,17 менша, ніж ефективність модифікованого, що дозволяє нам стверджувати про можливість модифікованого методу краще розпізнавати слова українською мови.

Отже, внесену нами зміну в метод вважатимемо ефективною в рамках використаного датасету, адже вона дозволила підняти ефективність методу класифікації Байєса на 5,6% і 2,9% в порівнянні з мультиноміальним і наївним методами відповідно, що є суттєвою зміною.

Для повноцінного аналізу потрібно випробовувати цю зміну на більшому датасеті в тисячі, десятки, а краще сотні тисяч повідомлень. Одна перш за все цей датасет потрібно зібрати, а це не входить в рамки нашого дослідження.

ВИСНОВОК

В рамках даного дослідження було проаналізовано ринок наявного програмного забезпечення машинного навчання для аналізу української мови, перш за все для виконання задачі класифікації.

Було виявлено деякі інструменти для роботи з мовою, як то: ВЕСУМ (Великий Електронний Словник Української мови), інструмент лематизації (знаходження кореню слова) тощо.

Датасету класифікованих на «спам» і «не спам» листів українською мовою знайдено не було, тому датасет був підібраний з власної поштової скриньки і колекції смс-повідомлень.

В процесі класифікації датасету було виявлено похибку стандартного nltk Python методу класифікації, розробленого з огляду на англійську мову, який невірно знаходив границі українських слів з апострофами.

Для покращення аналізу тексту українською мовою нами було розроблено модифікований метод наївної класифікації Байєса з покращеною очисткою пунктуації тексту. Було знайдено і виправлено проблему аналізу текстів української мови за допомогою існуючих інструментів в мові програмування Python, яка полягає в словах з апострофами, що повністю ігнорувались програмою.

З метою усунення даної проблеми було внесено модифікації в наївний класифікатор Байєса, які розподіляють речення на слова з урахуванням особливостей української мови. Ефективність модифікованого методу склала 0,883, або 88,3%, що перевищило ефективність і звичайного наївного методу класифікатору Байєса і мультиноміального класифікатора з бібліотеки nltk.

В рамках виконання даного дослідження було написано статтю під назвою «Модифікація методу класифікації Байєса в задачах виявлення спаму українською мовою», яка висвітлює головні зміни, внесені до методу класифікації Байєса з метою підвищення ефективності класифікації

повідомлень українською мовою. Вказана стаття була подана до опублікування в журналі «Біоніка Інтелекту». Повний текст статті приведено в додатку В.

Практична користь роботи полягає у можливості використання модифікованого методу для виконання задачі класифікації тексту українською мовою з більшою ефективністю, ніж доступні аналоги на мові Python.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Joseph Johnson. Number of sent and received e-mails per day worldwide from 2017 to 2025 / Johnson Joseph URL: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/ (дата звернення: 25.03.2021).

2. Simon Kemp. Digital 2020: 3.8 billion people use social media / Simon Kemp URL: https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media (дата звернення: 25.03.2021).

3. Барсегян А. А. Анализ данных и процессов: учеб. пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. – 3-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2009. – 512 с.

4. Yang Y. A re-examination of text categorization methods / Y. Yang, X. Liu // Proc. of Int.ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999. – P. 42-49.

5. Вагин В. Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянская, М. В. Фомина. – Москва : Физматлит, 2004. – 704 с.

6. Quinlan J. R. C4.5 Programs for machine learning / Quinlan J. R. – Morgan Kaufmann Inc. – San Mateo,Californie, 1993. – 240 p.

7. Joachims T. Making large-scale SVM learning practical / T. Joachims // Advances in Kernel Methods Support Vector Learning. – MIT Press, 1999. – 218 p.

8. C.J.C Burges. A Tutorial on Support Vector Machines for Pattern Recognition / Burges C.J.C // Vol 2: Data Mining and Knowledge Discovery. – 1998, P. 121-167.

9. Yang Y. A re-examination of text categorization methods / Y. Yang, X. Liu // Proc. SIGIR’2012, 22nd ACM International Conference on Research and Development in Information Retrieval, 2012. – P. 42-49.

10. Sebastiani F. Machine learning in automated text categorization / F. Sebastiani // ACM Comput. Surv. – March 2010. – Vol. 34, No. 1. – P. 1-47.

11. Jean Dos Santos. Ham or Spam? SMS Text Classification Walkthrough / Santos Dos Jean URL: https://www.kaggle.com/jeandsantos/ham-or-spam-sms-text-classification-walkthrough (дата звернення: 28.03.2021).

12. SMS Spam Collection Dataset / URL: https://www.kaggle.com/uciml/sms-spam-collection-dataset (дата звернення: 01.04.2021).

13. Oles Petriv, Serhii Shekhovtsov Sentiment Dictionary for Ukrainian / Petriv Oles, Shekhovtsov Serhii URL: https://lang.org.ua/en/dictionaries/ (дата звернення: 25.03.2021).

14. Grammarly Team URL: https://ua-gec-dataset.grammarly.ai/ (дата звернення: 01.04.2021)

15. Г. Г. Четвериков, Єрохін А.Л / URL: http://bionica-scimag.com/ua/index (дата звернення: 21.04.2021).

16. Yerokhin, A., Nechyporenko A., Babii A., Turuta A. Usage of F-transform to finding informative parameters of rhinomanometric signals / Proc. of the International Conference on Computer Sciences and Information Technologies, IEEE, Lviv, Ukraine, 2015, 14-17 September. – P.129-132.

17. Yerokhin, A.L., Babii, A.S., Nechyporenko, A.S., Turuta, O.P. / A Lars-Based Method of the Construction of a Fuzzy Regression Model for the Selection of Significant Features // Cybernetics and Systems Analysis. №4, 2016. - P. 167–173. DOI: 10.1007/s10559-016-9867-

18. Andriy Yerokhin, Alina Nechyporenko, Andrii Babii, Oleksii Turuta, Ihor Mahdalina. Usage of Phase Space Diagram to Finding Significant Features of Rhinomanometric Signals // Computer Science & Information Technologies (CSIT'2016), 6-10 Sept. 2016, Lviv, Ukraine. – P. 70 – 72. DOI: 10.1109/STC-CSIT.2016.7589871.

19. Andriy Yerokhin, Valerii Semenets, Alina Nechyporenko, Andrii Babii, Oleksii Turuta. F-transform 3D Point Cloud Filtering Algorithm // Proc. of the 2th

IEEE International Conference on Data Stream Mining & Processing. 21-25 August 2018, Lviv, Ukraine. - P.524-527. DOI: 10.1109/DSMP.2018.8478581

20. Bradley P. Carlin / Bayes and Empirical Bayes Methods for Data Analysis, Second Edition 2nd Edition / P. Bradley Carlin, A. Thomas Louis. – 2000. – 440 p.

21. Richard S. Sutton Reinforcement Learning, second edition: An Introduction (Adaptive Computation and Machine Learning series) / Sutton S. Richard Barto G. Andrew. – 2018. – 552 p.