УДК: 004.8:519.2:663.2

Є. М. Лавренюк
С. О. Жуков

АВТОМАТИЗОВАНЕ ВИЯВЛЕННЯ СПАМ-ПОВІДОМЛЕНЬ У ТЕКСТОВОМУ НАБОРІ ДАНИХ KAGGLE З ВИКОРИСТАННЯМ НАЇВНОГО КЛАСИФІКАТОРА БАЄСА

Вінницький національний технічний університет

Анотація

У роботі досліджено застосування наївного класифікатора Баєса для автоматизованої класифікації текстових повідомлень на два класи: звичайні повідомлення та спам. Для експериментального дослідження використано текстовий набір даних з Kaggle, що містить 83 448 повідомлень із мітками класів. Виконано завантаження даних, створення DataFrame, аналіз структури набору, перевірку пропущених значень, візуалізацію розподілу класів та попередню обробку тексту засобами бібліотеки NLTK. Реалізовано власну версію алгоритму Naive Bayes із використанням згладжування Лапласа. Якість класифікації оцінено за метриками Accuracy, Precision, Recall та F1-score. Отримані результати показали, що наївний класифікатор Баєса забезпечує високу якість розпізнавання спам-повідомлень: Accuracy = 0,9699, Precision = 0,9906, Recall = 0,9517, F1-score = 0,9708.

Ключові слова: наївний класифікатор Баєса, класифікація тексту, спам, машинне навчання, NLTK, Python, аналіз даних.

Abstract

The paper investigates the application of the Naive Bayes classifier for automated text message classification into two classes: regular messages and spam. A text dataset from Kaggle containing 83,448 labeled messages was used for the experimental study. Data loading, DataFrame creation, dataset structure analysis, missing value checking, class distribution visualization, and text preprocessing using the NLTK library were performed. A custom implementation of the Naive Bayes algorithm with Laplace smoothing was developed. Classification quality was evaluated using Accuracy, Precision, Recall, and F1-score metrics. The obtained results showed that the Naive Bayes classifier provides high-quality spam message detection: Accuracy = 0.9699, Precision = 0.9906, Recall = 0.9517, and F1-score = 0.9708.

Keywords: Naive Bayes classifier, text classification, spam, machine learning, NLTK, Python, data analysis.

Вступ

Сучасні інформаційні технології активно використовуються для автоматизованої обробки великих обсягів текстової інформації. Однією з поширених практичних задач є класифікація електронних повідомлень, зокрема виявлення спаму. Спам-повідомлення можуть містити рекламну, шахрайську або небажану інформацію, тому їх автоматичне розпізнавання є важливим елементом захисту користувачів та підвищення ефективності роботи інформаційних систем.

Для розв’язання задач класифікації текстів широко застосовуються методи машинного навчання. Одним із простих, швидких і водночас ефективних алгоритмів є наївний класифікатор Баєса. Його робота ґрунтується на теоремі Баєса та припущенні про умовну незалежність ознак. У задачах аналізу тексту такими ознаками можуть бути окремі слова або токени, що зустрічаються у повідомленнях.

Метою роботи є дослідження ефективності використання наївного класифікатора Баєса для класифікації текстових повідомлень на класи spam та ham, а також оцінювання якості побудованої моделі за стандартними метриками машинного навчання.

Основна частина

Для проведення дослідження було використано текстовий набір даних combined_data.csv, який містить два основні поля: label та text. Поле label визначає клас повідомлення, а поле text містить текст повідомлення. У роботі використано такі позначення класів: 1 — spam, тобто спам-повідомлення, 0 — ham, тобто звичайне повідомлення. Загальний обсяг набору даних становить 83 448 повідомлень, що дозволяє провести достатньо повне навчання та тестування класифікатора.

На початковому етапі було завантажено датасет у середовищі Kaggle та створено DataFrame за допомогою бібліотеки pandas. Перевірка структури даних показала, що набір містить 83 448 рядків і 2

стовпці. Стовпець label має цілочисловий тип, а стовпець text містить текстові дані. Також було перевірено наявність пропущених значень. У результаті встановлено, що пропуски у полях label та text відсутні, тому набір даних є придатним для подальшої обробки та моделювання.

	label	text
0	1	ounce feather bowl hummingbird opec moment ala...
1	1	wulvob get your medircations online qnb ikud v...
2	0	computer connection from cnn com wednesday es...
3	1	university degree obtain a prosperous future m...
4	0	thanks for all your answers guys i know i shou...

Рис. 1. Приклад записів текстового набору даних

Рисунок 1 демонструє фрагмент набору даних, який використовується для навчання класифікатора. У таблиці наведено приклади повідомлень та відповідні мітки класів. Такий формат даних дозволяє розглядати задачу як задачу бінарної класифікації текстів.

Далі було проаналізовано розподіл повідомлень за класами. У наборі даних міститься 39 538 звичайних повідомлень класу ham, що становить 47,38% вибірки, та 43 910 спам-повідомлень класу spam, що становить 52,62% вибірки. Отже, розподіл класів є близьким до збалансованого, що позитивно впливає на процес навчання моделі та дозволяє об'єктивніше оцінювати якість класифікації.

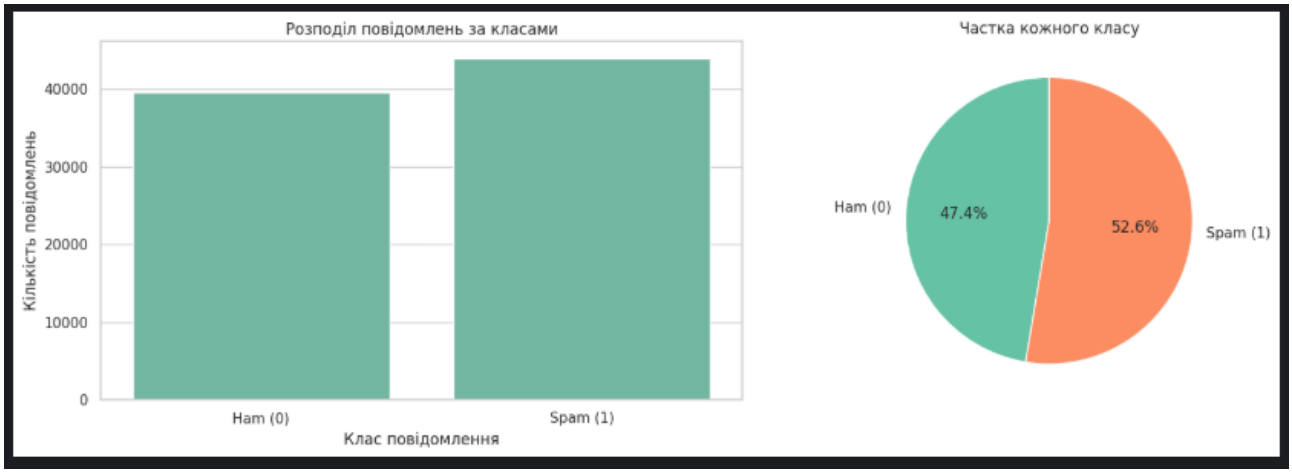

Рис. 2. Розподіл повідомлень за класами spam та ham

Рисунок 2 відображає кількість і частку повідомлень кожного класу. Наявність близького до збалансованого розподілу дозволяє використовувати не лише Accuracy, а й Precision, Recall та F1-score для більш повної оцінки якості класифікатора.

Наступним етапом була попередня обробка текстів за допомогою бібліотеки NLTK. Текстові повідомлення було приведено до нижнього регістру, очищено від URL-адрес, електронних адрес та службового токена escapenumber. Після цього виконано токенізацію, тобто поділ тексту на окремі слова. Для виділення слів застосовано RegexpTokenizer, який дозволяє залишити лише буквені токени. Також було видалено англомовні stop-слова, які не несуть суттєвого змістового навантаження, наприклад службові слова та прийменники. Додатково виконано лематизацію за допомогою WordNetLemmatizer, що дозволило привести слова до словникової форми.

Після попередньої обробки для кожного повідомлення було сформовано список унікальних токенів. Видалення повторів слів у межах одного повідомлення дозволило зменшити надлишковість текстових ознак та спростити подальше обчислення ймовірностей у моделі наївного Баєса.

	label	text	processed_text	token_count
67681	0	accuweather escapenumber day forecast for beve...	accuweather day forecast beverly hill tonight ...	33
61385	1	dear in christ the time has come for christian...	dear christ time come christian worship god sp...	151
41829	1	hallway cosponsor pry reimbursable coat clumsy...	hallway cosponsor pry reimbursable coat clumsy...	327
29172	1	does size matter' escapenumber of women said t...	size matter woman said thay unhappy lover intr...	68
35274	0	along zeng wrote hi all is there levene' test ...	along zeng wrote hi levene test could give adv...	61

Рис. 3. Приклад попередньої обробки текстових повідомлень

На рисунку 3 показано результат перетворення початкового тексту у підготовлений вигляд. Стовпець processed_text містить очищений текст після токенізації, лематизації та видалення зайвих слів, а token_count показує кількість унікальних токенів у повідомленні. Це дає змогу оцінити, наскільки ефективно було підготовлено текстові дані для класифікації.

Для додаткового аналізу було побудовано розподіл кількості унікальних токенів після обробки. Цей етап дозволяє оцінити довжину повідомлень і зрозуміти, наскільки різними є тексти за кількістю інформативних слів.

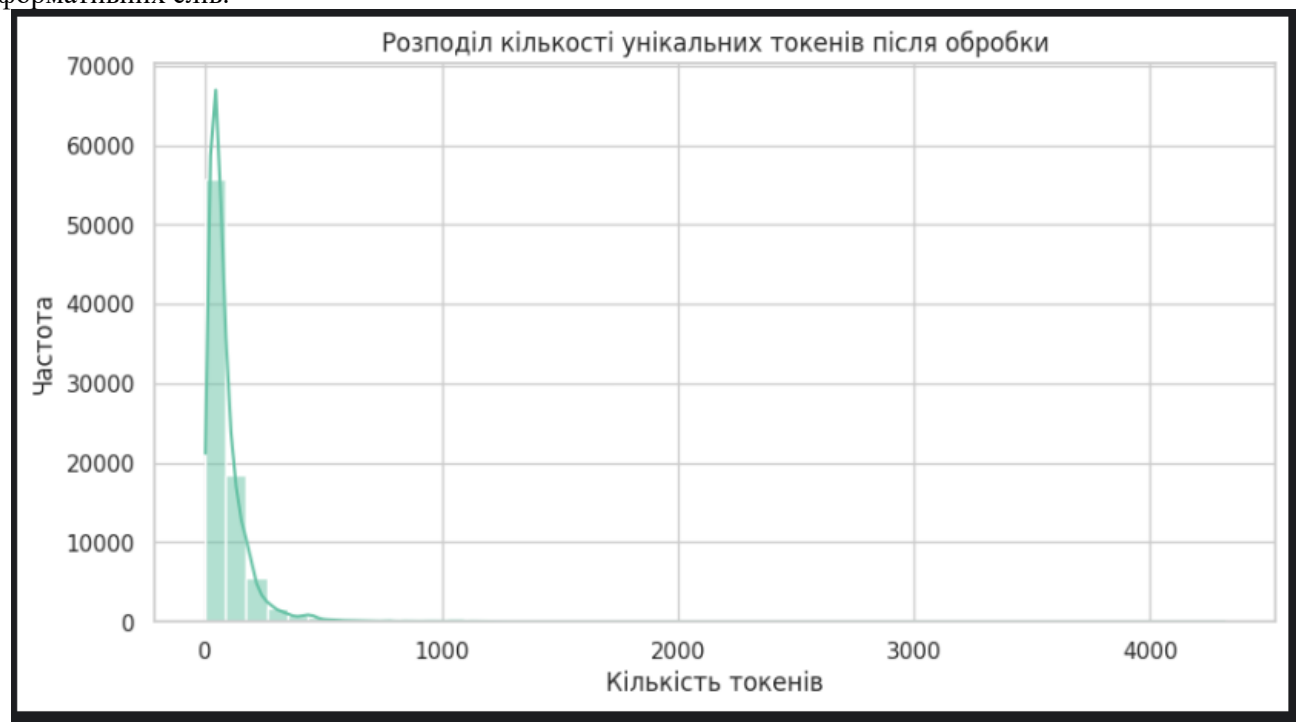

Рис. 4. Розподіл кількості унікальних токенів після попередньої обробки

Рисунок 4 демонструє, що повідомлення мають різну довжину після обробки. Частина текстів містить невелику кількість токенів, тоді як інші повідомлення є значно довшими. Такий аналіз є важливим, оскільки кількість токенів впливає на формування словника моделі та обчислення ймовірностей належності повідомлення до певного класу.

Після попередньої обробки дані було розділено на навчальну та тестову вибірки у співвідношенні 80/20 із використанням стратифікації. Це дозволило зберегти приблизно однакове співвідношення класів у навчальній і тестовій частинах. У результаті було сформовано 35 128 spam-повідомлень для тренування, 31 630 ham-повідомлень для тренування та 16 690 тестових повідомлень.

Для реалізації алгоритму було створено структури train_spam, train_ham та test_emails. Структура train_spam містить токени спам-повідомлень, train_ham — токени звичайних повідомлень, а test_emails — тестові повідомлення, для яких потрібно визначити клас.

У роботі реалізовано власну версію наївного класифікатора Баєса. На етапі навчання для кожного класу підраховується частота появи слів у spam та ham повідомленнях. Далі формується загальний словник усіх токенів і обчислюються апріорні ймовірності класів. Для уникнення нульових імовірностей використано згладжування Лапласа з параметром alpha = 1.0.

Під час класифікації нового повідомлення модель обчислює логарифмічні ймовірності належності тексту до класів spam та ham. Використання логарифмів дозволяє уникнути проблем із дуже малими значеннями ймовірностей, які виникають при множенні великої кількості ймовірностей окремих слів. Повідомлення відноситься до того класу, для якого отримано більше значення логарифмічної ймовірності.

Якість побудованого класифікатора було оцінено за метриками Accuracy, Precision, Recall та F1-score. За результатами тестування модель показала такі значення: Accuracy = 0,9699, Precision = 0,9906, Recall = 0,9517, F1-score = 0,9708. Отримані результати свідчать про високу якість класифікації текстових повідомлень.

	Accuracy	Precision	Recall	F1-score
0	0.9699	0.9906	0.9517	0.9708

Рис. 5. Значення метрик якості наївного класифікатора Баєса

Рисунок 5 відображає основні метрики якості моделі. Високе значення Accuracy показує, що більшість повідомлень класифіковано правильно. Значення Precision = 0,9906 свідчить про те, що серед повідомлень, які модель визначила як спам, дуже мала частка помилкових спрацьовувань. Recall = 0,9517 показує, що модель виявляє більшість реальних спам-повідомлень. F1-score = 0,9708 підтверджує збалансованість моделі за точністю та повнотою.

Для більш детального аналізу було побудовано матрицю помилок. Вона дозволяє оцінити кількість правильно та неправильно класифікованих повідомлень для кожного класу.

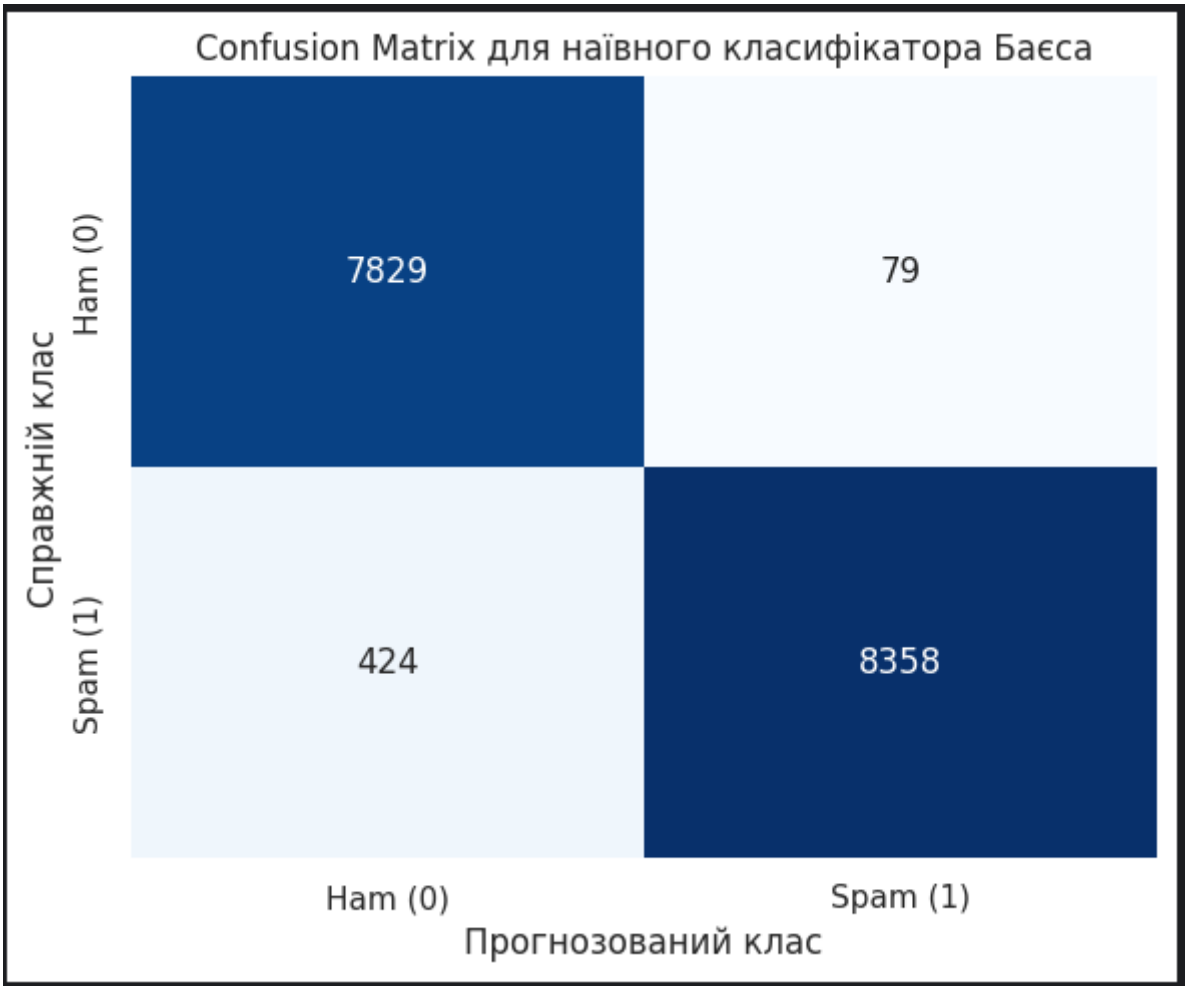

Рис. 6. Матриця помилок для наївного класифікатора Баєса

Рисунок 6 демонструє результати класифікації на тестовій вибірці. Матриця помилок дозволяє побачити, скільки звичайних повідомлень було правильно віднесено до класу ham, скільки спам-

повідомлень було правильно визначено як spam, а також кількість помилкових класифікацій. Такий аналіз є важливим для задач фільтрації спаму, оскільки помилкове віднесення звичайного повідомлення до спаму може бути небажаним для користувача.

Окремо було проведено аналіз слів, які мають найбільшу ймовірність зустрічатися у спам-повідомленнях. Для цього було сформовано таблицю з імовірностями P(word|spam) та P(word|ham), а також розраховано показник переваги слова для спаму як логарифм відношення цих імовірностей. Такий підхід дозволив визначити токени, які найкраще відрізняють spam від ham.

	word	P(word|spam)	P(word|ham)	spam_advantage
0	http	0.004506	0.004308	0.044797
1	com	0.003382	0.002411	0.338645
2	price	0.002972	0.000634	1.545133
3	one	0.002939	0.002387	0.208338
4	time	0.002583	0.002238	0.143322
5	day	0.002488	0.001367	0.599058
6	u	0.002376	0.001498	0.461399
7	get	0.002322	0.002301	0.009000
8	escapelong	0.002146	0.001137	0.635131
9	new	0.002043	0.002406	-0.163605
10	see	0.002039	0.001766	0.143840
11	please	0.002007	0.003847	-0.650807
12	like	0.002004	0.002215	-0.099829
13	offer	0.001915	0.000417	1.525410
14	may	0.001873	0.002169	-0.146658
15	best	0.001818	0.000896	0.707680
16	product	0.001784	0.000429	1.426103
17	need	0.001766	0.002171	-0.206615
18	www	0.001765	0.002666	-0.412379
19	money	0.001723	0.000242	1.963645

Рис. 7. Слова, що найбільше асоціюються зі спам-повідомленнями

Рисунок 7 показує слова з найбільшим значенням показника spam_advantage. Саме ці токени мають значно вищу ймовірність появи у спам-повідомленнях порівняно зі звичайними повідомленнями. Аналіз таких слів дозволяє краще інтерпретувати роботу моделі та зрозуміти, на основі яких ознак класифікатор приймає рішення.

Отримані результати підтверджують, що наївний класифікатор Баєса є ефективним інструментом для розв’язання задачі класифікації текстових повідомлень. Його перевагами є простота реалізації, висока швидкість навчання та достатньо висока якість прогнозування навіть при використанні базової попередньої обробки тексту.

Висновки

У роботі було досліджено застосування наївного класифікатора Баєса для задачі класифікації текстових повідомлень на два класи: spam та ham. Для експерименту використано набір даних combined_data.csv, який містить 83 448 повідомлень. У процесі аналізу встановлено, що дані не містять пропущених значень, а розподіл класів є близьким до збалансованого: 39 538 повідомлень належать до класу ham, а 43 910 — до класу spam.

Було виконано попередню обробку текстів за допомогою бібліотеки NLTK, що включала приведення тексту до нижнього регістру, очищення від зайвих елементів, токенізацію, видалення stop-слів, лематизацію та формування списків унікальних токенів. Після цього дані було поділено на навчальну та тестову вибірки у співвідношенні 80/20. Для навчання використано 35 128 spam-повідомлень і 31 630 ham-повідомлень, а тестова вибірка містила 16 690 повідомлень.

У роботі реалізовано власну версію алгоритму Naive Bayes із використанням згладжування Лапласа. За результатами тестування класифікатор показав високі значення метрик: Accuracy = 0,9699, Precision = 0,9906, Recall = 0,9517 та F1-score = 0,9708. Це свідчить про те, що модель добре розпізнає спам-повідомлення та має низьку кількість помилкових спрацьовувань.

Додатковий аналіз словника моделі дозволив визначити слова, які найчастіше асоціюються зі спамом та найкраще відрізняють спам-повідомлення від звичайних. Отже, теорема Баєса є ефективною основою для побудови простих, швидких та інтерпретованих класифікаторів текстових повідомлень. Отримані результати можуть бути використані для створення базових систем фільтрації спаму та попередньої автоматизованої обробки текстових даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Email Spam Classification Dataset. Kaggle [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/purusinghvi/email-spam-classification-dataset
2. Naive Bayes Classifier. Scikit-learn Documentation [Електронний ресурс]. – Режим доступу: https://scikit-learn.org
3. NLTK Documentation [Електронний ресурс]. – Режим доступу: https://www.nltk.org
4. Pandas Documentation [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org
5. Matplotlib Documentation [Електронний ресурс]. – Режим доступу: https://matplotlib.org
6. Seaborn Documentation [Електронний ресурс]. – Режим доступу: https://seaborn.pydata.org
7. Python Documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org

Лавренюк Євгеній Михайлович – студент групи СА-23б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця. e-mail: lavrenyuk0629@gmail.com

Жуков Сергій Олександрович– к.т.н., доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, e-mail: sazhukov@gmail.com

Lavreniuk Yevhemii – student of Faculty of Intellectual Information Technologies and Automation, SA-23b, Vinnytsia National Technical University, Vinnytsia, e-mail lavrenyuk0629@gmail.com

Zhukov Serhii O. - Ph.D., Assistant Professor of the Department of Systems Analysis and Information Technology, Vinnytsia National Technical University, Vinnytsia, e-mail: sazhukov@gmail.com