

Н. І. БОЙКО

кандидат економічних наук, доцент,  
доцент кафедри систем штучного інтелекту  
Національний університет «Львівська політехніка»  
ORCID: 0000-0002-6962-9363

В. Ю. МИХАЙЛИШИН

асистент кафедри систем штучного інтелекту  
Національний університет «Львівська політехніка»  
ORCID: 0000-0003-1889-9053

## АЛГОРИТМ КЛАСИФІКАЦІЇ ТЕКСТОВОГО КОНТЕНТУ СОЦІАЛЬНИХ МЕРЕЖ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО ТОНУ

В роботі наведено результати досліджень та порівняння результатів застосування наївного класифікатора Баєса з використанням простих словесних ознак та векторних моделей слів. Було проаналізовано методи та середовище дослідження, визначились з набором вхідних даних. Було проведено тренування класифікатора на обраному датасеті та оцінка його точності за допомогою функції `classify.accuracy` з бібліотеки `nlTK`. Також протестовано класифікатор на власному тексті та визначали правильність класифікації. Було побудовано гістограму, яка наглядно показала кількість правильно класифікованих позитивних та негативних прикладів. Була виведена `confusion matrix`, яка дозволила оцінити точність класифікації для кожного класу. В експериментальній частині використовувалась векторна модель слів `Word2Vec` з бібліотеки `gensim`. Проводилось тренування класифікатора та оцінювалася його точність. Здійснено значне покращення точності в порівнянні зі спрощеним підходом. В роботі обговорювались питання використання векторних моделей слів для покращення результатів класифікації тексту. Вони дозволяють краще врахувати семантику та контекст тексту, що призводить до більш точних результатів. Проаналізовано, що точність класифікації залежить від датасету, особливостей текстів та використаних методів обробки даних. В дослідженні наведений оптимальний вибір методів та підхід до класифікації, які повинні враховувати конкретну задачу та контекст застосування. В роботі розглядалися векторні моделі слів, та використання більш складних моделей класифікації. Наводились фактори, що впливають на емоційний стан тексту. Було проведено оптимізацію параметрів моделей для досягнення кращих результатів. Було підтверджено в результаті проведених експериментів, ефективність наївного класифікатора Баєса та векторних моделей слів у задачі класифікації емоційного стану тексту.

**Ключові слова:** алгоритм, емоційний тон, контент, класифікація, соціальна мережа.

N. I. BOYKO

Candidate of Economical Sciences, Associate Professor,  
Associate Professor at the Department of Artificial Intelligence  
Lviv Polytechnic National University  
ORCID: 0000-0002-6962-9363

V. YU. MYKHAILYSHYN

Assistant at the Department of Artificial Intelligence  
Lviv Polytechnic National University  
ORCID: 0000-0002-0060-2381

## ALGORITHM FOR CLASSIFICATION OF TEXT CONTENT OF SOCIAL NETWORKS FOR DETERMINING EMOTIONAL TONE

The article presents the results of the research and a comparison of the results of the application of the naive Bayes classifier using simple verbal signs and vector word models. The research methods and environment were analyzed, and a set of input data was determined. A classifier is trained on the selected dataset and its accuracy is evaluated using the `classify.accuracy` function from the `nlTK` library. The classifier was also checked on its own text and the correctness of the classification was determined. A histogram was constructed that visually showed the number of correctly classified positive and negative examples. A confusion matrix was derived, which made it possible to evaluate the classification accuracy for each class. In the experimental part, the vector word model `Word2Vec` from the `gensim` library was used. The classifier was trained and its accuracy was evaluated. A significant increase in accuracy is achieved compared to the simplified approach. The paper discussed the issue of using vector models of words to improve text classification results. They allow you to better take into account the semantics and context of the text, which leads to more accurate results.

*It was analyzed that the accuracy of the classification depends on the data set, the features of the texts, and the data processing methods used. Research provides an optimal choice of methods and an approach to classification, which must take into account the specific task and context of the application. The work considered vector models of words and the use of more complex classification models. Factors affecting the emotional state of the text are given. The model parameters were optimized to achieve better results. As a result of the conducted experiments, the effectiveness of the naive Bayesian classifier and vector word models in the task of classifying the emotional state of the text was confirmed.*

**Key words:** algorithm, emotional tone, content, classification, social network..

### Постановка проблеми

На сьогоднішній день соціальні мережі є невід'ємною частиною сучасного суспільства. Вони також є великою платформою для вираження своїх емоцій та думок, комунікації з іншими людьми, для реклами, маркетингових кампаній, розвитку бізнесу, для формування спільнот та підтримки громадянських позицій, організації заходів, політичних дискусій тощо. Через це необхідний аналіз цієї системи, і це можливо зробити через класифікацію текстового контенту від користувачів, визначення їх емоційного тону [2, 6].

Неможливо недооцінити важливість задачі класифікації емоційного тону контенту, який публікують користувачі в соціальних мережах [3]. Це знаходить своє призначення у сфері реклами та маркетингу, соціальному моніторингу, аналізі громадської думки, тощо.

Таким чином, аналіз настроїв користувачів в соціальних мережах є досить важливою темою сьогодні, тому як очевидним є те, що соціальні мережі тісно пов'язані з бізнесом, висловлюванням своїх думок та позицій [1]. Відповідно, актуальним буде розгляд методів, який допоможе в аналізі соціальних мереж.

В рамках цієї роботи розглядалися методи машинного навчання. Для дослідження цієї теми обиралися різноманітні рішення:

- Метод опорних векторів (Support Vector Machines, SVM), який є потужним методом класифікації, що шукає оптимальну гіперплощину для розділення двох класів даних. Використання ядерних функцій дозволяє SVM працювати з нелінійно розділними даними.

- Дерева рішень (Decision Trees) будуються на основі серії розділень, щоб класифікувати дані. Дерева можуть бути розширені до ансамблів, таких як випадковий ліс (Random Forests) або градієнтний бустинг (Gradient Boosting), для поліпшення точності класифікації.

- Ансамблеві методи поєднують декілька моделей, щоб отримати кращу загальну прогнозу точність. Наприклад, можна застосувати ансамблі, такі як голосування багатьох класифікаторів (Voting Classifier) для аналізу настроїв користувачів.

- Наївний баєсів класифікатор (Naive Bayes classifier) є популярним методом для сентимент-аналізу, який полягає в класифікації текстових документів (у цьому випадку повідомлень з соціальних мереж) на позитивний, негативний або нейтральний сентимент. Його застосовують для класифікації тексту, фільтрації спаму, класифікації зображень тощо.

Для подальшого дослідження було обрано наївний баєсів класифікатор, так як він найбільше підходить для роботи з великим обсягом даних, простий в реалізації та ідеально інтегрується з метою дослідження.

Головна ідея наївного баєсового класифікатора полягає в тому, що він вважає всі ознаки незалежними одна від одної. Це називається «наївним» припущенням. Навіть якщо це припущення не завжди відповідає реальності, наївний баєсів класифікатор часто показує задовільні результати в практичних застосуваннях, особливо коли маємо великі обсяги даних [2, 7].

Задача наївного баєсового класифікатора полягає в призначенні категорії або класу до нового прикладу на основі ймовірностей. Конкретно, він використовує ймовірності атрибутів для класифікації даних.

Основна ідея полягає в обчисленні ймовірностей, що відображають те, наскільки ймовірно приклад належить до певного класу, враховуючи значення його атрибутів. Класифікатор обчислює ці ймовірності на основі навчального набору даних, де відомі мітки класів для тренувальних прикладів.

### Аналіз останніх досліджень і публікацій

Дане дослідження є актуальним та має новизну в декількох аспектах. Враховуючи, що наївний баєсів класифікатор є одним з найпростіших та ефективних методів машинного навчання, новизна полягатиме в його застосуванні саме до аналізу настроїв користувачів у соціальних мережах. Про що наголошують автори [7] в своїх дослідженнях.

В роботі [1] автором наголошується на важливості виконаного дослідження. Ця робота є вступним посібником у галузі аналізу соціальних медіа. Дане джерело пропонує читачам основні концепції, методи та інструменти для аналізу даних з соціальних медіа. В роботі [1] можна знайти розділи, які присвячені аналізу настроїв у соціальних медіа. Ця книга описує застосування наївного баєсового класифікатора у контексті аналізу настроїв. Автори [1] обґрунтовують застосування наївного баєсового класифікатора, описують алгоритм його роботи, наводять переваги та недоліки його застосування для аналізу настроїв у текстових даних з соціальних медіа.

В дослідженні [2] наведено практичне застосування аналізу даних з соціальних мереж. У цій книзі [2] автор ділиться знаннями про збір, обробку та аналіз даних з популярних соціальних мереж, таких як Facebook, Twitter, LinkedIn та інших. Дослідження присвячене аналізу настроїв і сентименту в текстових даних соціальних мереж. У цих розділах описуються методи машинного навчання, включно наївний баєсів класифікатор, для виконання завдань аналізу настроїв користувачів.

В роботі [3] охоплюється широкий спектр тем і методів аналізу настроїв. В її змісті можна знайти докладний опис наївного баєсового класифікатора та його застосування для аналізу настроїв. Автор [3] пояснює принципи роботи наївного баєсового класифікатора, його передумови та особливості. Деякі розділи книги включають приклади коду або практичні рекомендації щодо використання наївного баєсового класифікатора для аналізу настроїв у текстових даних.

У книзі [4] автор охоплює широкий спектр методів та підходів до аналізу настроїв у тексті, включаючи такі теми, як аналіз емоцій, класифікація суб'єктивних висловів, аналіз думок та деякі приклади застосування цих методів. Джерело містить важливі концепції, методи та рекомендації для проведення аналізу настроїв, а також розглядає реальні застосування в цій галузі. Також описані різні методи машинного навчання, які використовуються для аналізу настроїв. Хоча конкретний метод наївного баєсового класифікатора не є центральною темою цієї книги, автор знайомить читачів з концепцією наївного баєсового класифікатора та включає його як один із можливих методів для виконання завдань аналізу настроїв.

Отож, **метою дослідження** є вивчення методу машинного навчання, відомого як наївний баєсів класифікатор та використати його для аналізу настроїв користувачів в соціальних мережах. Завдання дослідження є застосування набору даних для проведення класифікації за емоційним тоном текстових повідомлень користувачів з соціальних мереж.

Об'єктом нашого дослідження є текстовий контент, що генерують користувачі в соціальних мережах. Це можуть бути пости, коментарі, повідомлення, відгуки або будь-які інші форми текстових висловлювань.

#### Викладення основного матеріалу дослідження

Математична постановка задачі аналізу соціальних мереж для визначення настроїв користувачів може бути представлена у вигляді множини користувачів  $N$ , множини можливих настроїв  $M$ , вектор та матриця ознак для кожного користувача [5, 8].

Припустимо, що ми маємо набір даних з  $N$  користувачів, де кожен користувач має асоційований з ним вектор ознак  $X = (x_1, x_2, \dots, x_k)$ , де кожна ознака  $x_i$  може відображати, наприклад, ключові слова, емоційні індикатори або інші деталі повідомлень користувача. Ми також маємо  $M$  можливих класів настроїв, позначимо їх як  $C_1, C_2, \dots, C_m$ .

У математичній постановці задачі визначаються наступні змінні [9]:

- $N$  – кількість користувачів у соціальній мережі.
- $M$  – кількість можливих настроїв, які можуть бути визначені, наприклад, позитивний, негативний, нейтральний і т.д.
- $X$  – матриця розміром  $N \times M$ , де  $X[i, j]$  представляє настрої  $j$ -го користувача  $i$ -тої соціальної мережі. Кожен елемент матриці  $X$  відповідає настрою користувача на певному соціальному медіа.

Для розв'язання цієї задачі можна використовувати наївний баєсівський класифікатор, який є статистичним методом машинного навчання. Цей класифікатор базується на Баєсовій теоремі та припущенні про незалежність ознак [9].

Для використання наївного баєсівського класифікатора вводяться наступні ймовірності:

- $P(C|X)$  – ймовірність того, що користувач належить класу  $C$  (настрою) при заданому векторі ознак  $X$  (повідомлення користувача).
- $P(X|C)$  – ймовірність зустрічі вектора ознак  $X$  (повідомлення користувача) при заданому класі  $C$  (настрою).
- $P(C)$  – апіорна ймовірність класу  $C$  (настрою) без урахування жодних ознак.
- $P(X)$  – апіорна ймовірність вектора ознак  $X$  (повідомлення користувача) без урахування класів.

Ймовірність  $P(C|X)$  обчислюється за формулою Баєса (формула 1):

$$P(C|X) = (P(X|C) \times P(C)) : P(X). \quad (1)$$

Ймовірність  $P(X|C)$  визначається як добуток ймовірностей  $P(x_i|C)$  для кожної ознаки  $x_i$  вектора ознак  $X$  (повідомлення користувача).

Спектр настрою визначається на основі значень ймовірностей  $P(C|X)$  для кожного класу  $C$ . Якщо ймовірність  $P(\text{позитивний}|X)$  більша за ймовірності  $P(\text{негативний}|X)$  і  $P(\text{нейтральний}|X)$ , то настрої користувача вважається позитивним.

Наївний баєсів класифікатор припускає, що ознаки вхідних даних незалежні між собою, що дає можливість спростити обчислення умовних ймовірностей [4].

Основа методу наївного баєсівського класифікатора полягає в припущенні, що ознаки є незалежними одна від одної, за винятком залежності від класу. Це спрощення дозволяє ефективно моделювати залежності між ознаками та класами, зокрема у випадку великої кількості ознак.

Припустимо, ми маємо набір даних, що складається з вхідних ознак  $X = (x_1, x_2, \dots, x_n)$ , де кожна ознака представляє числове або категоріальне значення, і маємо цільову змінну  $y$ , яку потрібно класифікувати на основі цих ознак.

Метод наївного баєсового класифікатора використовує умовну ймовірність  $P(y|X)$ , щоб призначити клас  $y$  для нових вхідних даних  $X$ . За допомогою Баєсової теореми, ця умовна ймовірність може бути виражена як (Формула 2):

$$P(y|X) = (P(X|y) \times P(y)) : P(X), \quad (2)$$

де  $P(X|y)$  – умовна ймовірність вхідних ознак  $X$  при заданому класі  $y$ ,  $P(y)$  – апіорна ймовірність класу  $y$ ,  $P(X)$  – маргінальна ймовірність вхідних ознак  $X$  [13, 5].

Припущення про наївність полягає в тому, що всі ознаки  $X$  вважаються незалежними між собою при заданому класі  $y$ . Це припущення спрощує обчислення умовної ймовірності  $P(X|y)$  і дозволяє представити його у вигляді добутку умовних ймовірностей кожної окремої ознаки.

Застосовуючи цей метод, ми можемо обчислити ймовірності для кожного класу  $y$  і призначити новим вхідним даним клас з найвищою ймовірністю.

Щоб покращити роботу наївного баєсового класифікатора, можна застосувати деякі з способів:

– Використання незалежних наївних баєсових класифікаторів: замість припущення про незалежність ознак, слід використовувати незалежні наївні баєсові класифікатори, де кожна ознака розглядається окремо.

– Присвоєння ваг або значимості кожній ознаці на основі її впливу на класифікацію. Це може бути зроблено, наприклад, за допомогою використання *TF-IDF* (Term Frequency-Inverse Document Frequency), де ознаки, які часто зустрічаються в документі, але не часто в інших документах, отримують вищі ваги. Формула для обчислення ваги ознаки має вигляд (формула 3):

$$weight(x_i) = TF(x_i) \times IDF(x_i), \quad (3)$$

де  $TF(x_i)$  – частота ознаки  $x_i$  в документі,  $IDF(x_i)$  – зворотна частота ознаки  $x_i$  в корпусі документів.

Згладжування Лапласа дозволяє більш реалістично оцінити ймовірності ознак і допомагає покращити роботу наївного баєсового класифікатора, особливо в випадках, коли маємо обмежену кількість даних. Уявимо, що деяка ознака не з'явилася у тренувальному наборі даних для певного класу, то умовна ймовірність цієї ознаки для цього класу буде рівна нулю. Це може спричинити проблеми при подальшому обчисленні ймовірностей. Згладжування Лапласа вводить маленьку константу (зазвичай 1) до кількості спостережень кожної ознаки для кожного класу, забезпечуючи, що нульових ймовірностей не виникає.

Формула для обчислення умовної ймовірності ознаки  $x_i$  для певного класу  $y$  з використанням згладжування Лапласа виглядає наступним чином (формула 4):

$$P(x_i|y) = (N_i + 1) / (N + K), \quad (4)$$

де  $N_i$  – кількість спостережень ознаки  $x_i$  у класі  $y$ ,  $N$  – загальна кількість спостережень у класі  $y$ ,  $K$  – загальна кількість можливих значень ознаки  $x_i$  [2, 5].

Відповідно, математична модель для аналізу соціальних мереж з метою визначення настроїв користувачів може бути наступною:

1. Множина користувачів: позначимо множину користувачів як  $U = \{u_1, u_2, \dots, u_n\}$ , де  $n$  – загальна кількість користувачів.

2. Множина повідомлень: позначимо множину повідомлень, які розміщують користувачі в соціальній мережі, як  $M = \{m_1, m_2, \dots, m_m\}$ , де  $m$  – загальна кількість повідомлень.

3. Настрої користувачів: кожен користувач  $u_i$  має свій настрій, який може бути позитивним, негативним або нейтральним. Позначимо множину настроїв як  $S = \{positive, negative, neutral\}$ .

4. Ознаки повідомлень: для кожного повідомлення  $m_j$  визначимо набір ознак, які можуть бути використані для визначення настрою. Нехай  $F = \{f_1, f_2, \dots, f_k\}$  буде множина ознак, де  $k$  – загальна кількість ознак.

5. Матриця ознак: створимо матрицю  $X$  розміром  $n \times k$ , де кожен рядок відповідає користувачу, а кожний стовпчик відповідає ознаці. Кожен елемент  $X_{ij}$  матриці представляє значення ознаки  $f_j$  для користувача  $u_i$ .

6. Класифікатор: Використовуємо наївний баєсів класифікатор для класифікації настроїв користувачів на основі їх повідомлень. Припустимо, що  $P(y)$  є апіорною ймовірністю настрою  $y$ , а  $P(x_j|y)$  є умовною ймовірністю ознаки  $f_j$  при настрої  $y$ . Класифікація здійснюється на основі формули Байеса (формула 5):

$$P(y|X) = P(y) \times P(x_j|y) / P(X), \quad (5)$$

де  $P(y|X)$  – умовна ймовірність настрою  $y$  при використанні набору ознак  $X$ ,  $P(y)$  – апіорна ймовірність настрою  $y$ ,  $P(x_j|y)$  – умовна ймовірність ознак,  $P(X)$  – ймовірність набору ознак  $X$ .

7. Класифікація настроїв: на основі отриманої умовної ймовірності  $P(y|X)$ , класифікуємо настрої користувачів на позитивний, негативний або нейтральний.

8. Оцінка точності: для оцінки точності класифікації можна використовувати метрики, такі як точність, відновлення та F-міру.

Ця математична модель дозволяє аналізувати соціальні мережі з метою визначення настроїв користувачів на основі їх повідомлень. Застосування наївного баєсового класифікатора дозволяє здійснити класифікацію настроїв з використанням ймовірнісних моделей.

Для аналізу було обрано набір даних Sentiment Analysis on Twitter. Цьому вибору посприяло ряд причин:

- Доступність.

Sentiment Analysis on Twitter є широко відомим та легко доступним датасетом, який можна знайти онлайн. Він використовується у багатьох дослідженнях та проектах з аналізу настроїв, що дозволяє отримати доступ до різноманітних ресурсів та підходів, пов'язаних з цим датасетом.

- Розмічені дані.

Датасет містить твіти, які були розмічені на позитивні, негативні або нейтральні. Це значно полегшує розробку та тренування моделей аналізу настроїв, оскільки існують вже готові мітки настрою для навчання та оцінки ефективності моделі.

- Репрезентативність.

Twitter є популярною соціальною мережею, де користувачі активно висловлюють свої думки, враження та емоції. Тому аналіз настроїв на основі твітів дозволяє отримати значимі інсайти щодо думок та настрою групи користувачів.

Цей датасет містить в собі 10000 унікальних твітів та 3 категорії. Він має інформацію, який з настроїв (позитивний, негативний або нейтральний) притаманний цьому твіту, та його ID.

Таблиця 1

Категорії твітів датасету Sentiment Analysis on Twitter

| Назва поля    | Тип     | Опис  |
|---------------|---------|---|
| Tweet ID      | Integer | Містить цифровий ідентифікатор поста в соціальній мережі Твіттер. |
| Sentiment     | String  | Оцінка емоційного тону тексту, що написав користувач.             |
| Tweet content | String  | Зміст самого текстового повідомлення від користувача.             |

Для подальшого аналізу потрібно здійснити перетворення тексту в числовий вектор, який можна використовувати для навчання моделей машинного навчання. Це може бути досягнуто за допомогою методів, таких як *TF-IDF* (term frequency-inverse document frequency) або векторизації слів на основі Word2Vec чи GloVe.

Загалом, наведені методи попередньої обробки даних можуть збільшити якість та достовірність даних, покращити ефективність моделей та зробити їх більш стійкими до шуму та випадкових варіацій у вхідних даних.

Для початку, натреноємо наш класифікатор, на обраному нами датасеті, та дізнаємось наскільки він точний. Для оцінки точності використаємо `classify.accuracy` з бібліотеки `nlTK`, виведемо цю точність в консоль (рис. 1), та для наглядності прокласифікуємо текст, написаний нами. Результат представлений на рис. 2.

```
Accuracy: 0.7346666666666667
I love this product! : Positive
```

Рис. 1. Точність класифікатора

```
custom_tweet = "I love this product!"
custom_tokens = remove_noise(word_tokenize(custom_tweet))
print(custom_tweet, ":", classifier.classify(dict([token, True] for token in custom_tokens)))
```

Рис. 2. Приклад коду до представлених результатів

Результати досить точні, можна помітити, що коментар "I love this product!", був визначений як позитивний, що є правильним.

Далі, на основі тестувальних даних, побудуємо гістограму, в якій покажемо, скільки результатів було правильно визначені як позитивні, та скільки результатів було правильно визначено як негативні, покажемо гістограму за допомогою `matplotlib`, а дані для неї підготуємо за допомогою написаної функції, функцію наведемо нижче (рис. 3):

```
def calculate_accuracy(reference_set, predicted_set, sentiment):
    correct = [tweet for tweet in predicted_set if tweet[1] == sentiment]
    accuracy = len(correct) / len(reference_set) * 100
    return accuracy
```

Рис. 3. Код для пошуку результатів виконання відносно емоцій

Далі виведемо саму гістограму, побудовану за допомогою matplotlib (рис. 4):

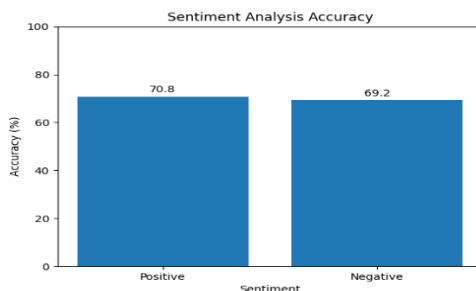


Рис. 4. Гістограма точності класифікатора відносно емоції, яку він передбачає

Виведемо також confusion matrix (рис. 5), для наших даних, ця матриця є інструментом для оцінки точності моделі класифікації. Вона допомагає визначити, як добре модель працює у відношенні розпізнавання різних класів.

У матриці переконань можна побачити чотири основні значення:

- True Positive (TP): Кількість прикладів, для яких модель правильно передбачила позитивний клас.
- True Negative (TN): Кількість прикладів, для яких модель правильно передбачила негативний клас.
- False Positive (FP): Кількість прикладів, для яких модель неправильно передбачила позитивний клас (тобто вказала наявність позитивного класу, хоча його немає).
- False Negative (FN): Кількість прикладів, для яких модель неправильно передбачила негативний клас (тобто вказала наявність негативного класу, хоча його немає).

Тому важливо також показати її, для більшого розуміння отримання результатів.

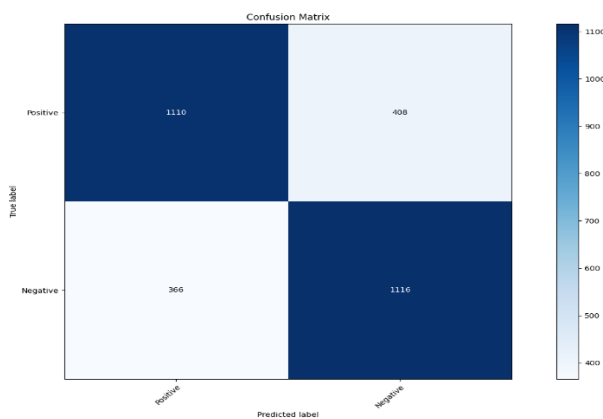


Рис. 5. Матриця переконань для нашого класифікатора

Спробуємо покращити результати нашої моделі, для цього використаємо векторну модель слів (Word Embeddings) замість простого підходу з використанням словесних ознак. В цьому випадку, ми будемо використовувати модель слів Word2Vec, з бібліотеки gensim.

Натронуємо знову наш класифікатор, та дізнаємось наскільки наш класифікатор точний (рис. 6).

```
Accuracy: 0.8275
I love this product! : Positive
```

Рис. 6. Точність класифікатора та написаний нами текст, навпроти нього його зміст

Можна помітити, що точність значно зросла, що може вказувати на те, що використання векторних моделей слів, значно покращує результати. Виведемо ще гістограму, для візуалізації точності відносно класів (рис. 7).

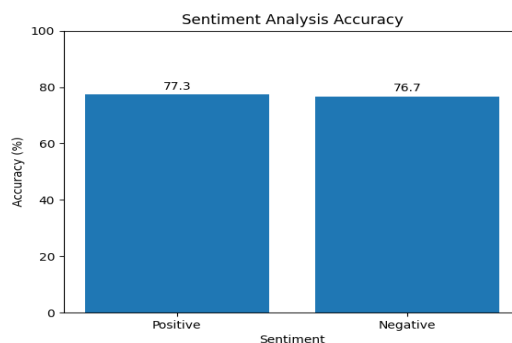


Рис. 7. Гістограма точності класифікатора відносно емоції, яку він передбачає

### Висновки

Наївний класифікатор Баєса є простим і ефективним методом для класифікації тексту, оснований на застосуванні теореми Баєса. Він припускає, що кожна ознака (у випадку тексту – слово) незалежна від інших ознак при заданому класі. Наївний класифікатор Баєса заснований на ймовірності того, що документ належить до певного класу, враховуючи наявність певних ознак (слів) у документі.

Під час тренування нашого класифікатора з використанням наївного класифікатора Баєса, ми використали датасет з мітками класів (позитивний або негативний) для навчання моделі. Кожен текст (коментар) у датасеті розбивається на окремі слова, і знаходяться частоти появи кожного слова у позитивних і негативних текстах. Для класифікації нового тексту модель використовує отримані ймовірності та формулу теореми Баєса для обчислення ймовірності належності тексту до позитивного або негативного класу.

Після навчання класифікатора, ми оцінюємо його точність, використовуючи функцію `classify.accuracy` з бібліотеки NLTK. Ця функція порівнює передбачені мітки класу з фактичними мітками у тестовому наборі даних і обчислює відсоток правильно класифікованих прикладів.

Для візуалізації кількості правильно класифікованих позитивних і негативних прикладів побудували гістограму. Для цього ми використовуємо бібліотеку Matplotlib та готуємо дані за допомогою власної написаної функції.

Confusion matrix є інструментом для оцінки точності моделі класифікації. Вона надає інформацію про кількість правильно і неправильно класифікованих прикладів для кожного класу. У матриці переконань є чотири значення: TP, TN, FP і FN, які дозволяють оцінити ефективність моделі у розпізнаванні різних класів.

В наступній частині експерименту ми використовуємо векторну модель слів (Word Embeddings), зокрема модель Word2Vec з бібліотеки Gensim. Векторна модель слів дозволяє представити слова у векторному просторі, враховуючи семантичні залежності між ними. Ми повторюємо той самий процес тренування та оцінки точності, використовуючи цю векторну модель слів.

Точність класифікатора з використанням векторної моделі слів значно зросла порівняно з попереднім підходом. Це може свідчити про те, що використання векторних моделей слів дозволяє краще врахувати семантику та контекст тексту, що сприяє покращенню точності класифікації.

Загалом, експеримент показує, що використання наївного класифікатора Баєса в поєднанні з векторними моделями слів може покращити результати класифікації тексту. Однак, для досягнення ще кращих результатів можуть бути застосовані інші методи або оптимізації, наприклад, використання більш складних моделей класифікації або врахування інших факторів, які можуть впливати на емоційний стан тексту.

### Список використаної літератури

1. Pang B., Lee L., Vaithyanathan S. Thumbs up: sentiment classification using machine learning techniques. Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing. Vol. 10. Association for Computational Linguistics. 2002. Pp. 321–342.
2. Maas A.L., Daly R.E., Pham P.T., Huang D., Ng A.Y., Potts C. Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics. ACL 2011. 2011. Pp. 23–36.
3. Rennie J.D. Tackling the poor assumptions of naive bayes text classifiers. Machine Learning-International Workshop then Conference. 2003. Vol. 20(2). Pp. 56–62.
4. Tseng C., Patel N., Paranjape H., Lin T. Y., Teoh S. Classifying twitter data with naive bayes classifier. IEEE International Conference on Granular Computing. 2012. Pp. 89–101.

5. Estivill-Castro V., Lee I. Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram. 9th Intern. Symp. on spatial data handling, Beijing, China. 2000. Pp. 26–41.
6. Guo D., Peuquet D.J., Gahegan M. ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata. *Geoinformatica*. 2003. Vol. 3. N. 7. Pp. 229–253.
7. Harel D., Koren Y. Clustering spatial data using random walks. Proc. of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, San Francisco, California. 2000. Pp. 281–286.
8. Boyko N., Pylypiv O., Peleshchak Yu., Kryvenchuk Yu., Campos J. Automated Document Analysis for Quick Personal Health Record Creation. The 2 nd International Workshop on Informatics & Data-Driven Medicine (IDDM 2019). Lviv, Ukraine, November 11-13. 2019. Vol. 1. Pp. 208–221.
9. Yakovyna V., Peleshchyshyn A., Albota S. Discussions of wikipedia talk pages: Manipulations detected by lingual-psychological analysis, CEUR Workshop Proceedings. 2019. Vol. 2392. Pp. 309–320.

#### References

1. Pang, B., Lee, L., Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing. Vol. 10. Association for Computational Linguistics, 321–342 [in English].
2. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics. ACL 2011, 23–36 [in English].
3. Rennie, J.D. (2003). Tackling the poor assumptions of naive bayes text classifiers. Machine Learning-International Workshop then Conference, 20(2), 56–62 [in English].
4. Tseng, C., Patel, N., Paranjape, H., Lin, T. Y., Teoh, S. (2012). Classifying twitter data with naive bayes classifier. IEEE International Conference on Granular Computing, 89–101 [in English].
5. Estivill-Castro, V., Lee, I. (2000). Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram. 9th Intern. Symp. on spatial data handling, Beijing, China, 26–41 [in English].
6. Guo, D., Peuquet, D.J., Gahegan, M. (2003). ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata. *Geoinformatica*, 3(7), 229–253 [in English].
7. Harel, D., Koren, Y. (2000). Clustering spatial data using random walks. Proc. of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, San Francisco, California, 281–286 [in English].
8. Boyko, N., Pylypiv, O., Peleshchak, Yu., Kryvenchuk, Yu., Campos, J. (2019). Automated Document Analysis for Quick Personal Health Record Creation. The 2 nd International Workshop on Informatics & Data-Driven Medicine (IDDM 2019), Lviv, Ukraine, November 11–13, 1, 208–221 [in English].
9. Yakovyna, V., Peleshchyshyn, A., Albota, S. (2019). Discussions of wikipedia talk pages: Manipulations detected by lingual-psychological analysis, CEUR Workshop Proceedings, 2392, 309–320 [in English].