УДК: 004.93

РОЗВ'ЯЗАННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТІВ МЕТОДАМИ ОБРОБКИ ПРИРОДНЬОЇ МОВИ ТА МАШИННОГО НАВЧАННЯ

кандидат технічних наук, доцент, Сперкач М. О., Юзьвак Д. Ю.
Національний технічний університет України Київський політехнічний інститут імені Ігоря Сікорського

У статті розглядається практичне застосування методів природньої обробки мови та машинного навчання для вирішення задачі класифікації текстів. Описано процес діяльності, що автоматизується в рамках розроблення системи класифікації текстів, сформульовано постановку задачі та описано методи її вирішення. Зроблено висновки щодо застосування алгоритмів машинного навчання для вирішення поставленої задачі. Описано результати щодо ефективності використання моделей машинного навчання на основі різних алгоритмів. Встановлено, що поєднання методів обробки природньої мови та машинного навчання є ефективним способом вирішення поставленої задачі.

Ключові слова: машинне навчання, обробка текстів, обробка природньої мови, модель, класифікація текстів

Sperkach M., PhD of Technical Sciences, Associate Professor; Yuzvak D. Solving the text classification problem using the natural language processing and machine learning methods / National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

This article deals with the practical methods of applying the methods of natural language processing and machine learning to solve the problem of text classification. The activity process that is automated is described during the developing of the system for text classification. The main problem was formulated. Conclusions are made on the application of machine learning algorithms for solving the problem. The results of using different algorithms for the creation of machine learning models were discussed. It was concluded that the combination of the natural language processing and machine learning methods is the effective way for solving the text classification problem.

Key words: machine learning, text processing, natural language processing, model, text classification.

Вступ. Сучасний світ просто потопає у великих обсягах інформації, кількість яких стрімко зростає. Для людини все складніше стає аналізувати, обробляти та класифікувати дані за категоріями.

При цьому зростання інформації і одночасне зростання доступної обчислювальної потужності комп'ютерів дозволяють використовувати сучасні методи для вирішення задачі класифікації.

Процес класифікації текстів займає досить багато часу, адже для вирішення цієї задачі необхідно не тільки прочитати текст, а й проаналізувати його зміст та обрати підходящу категорію із множини доступних. Коли ж мова йде про глобальні задачі та велику кількість даних, то часто виникають проблеми із браком людських ресурсів для їх оброблення.

Пропонується автоматизувати процес класифікації текстів.

Автоматизована класифікація текстів розглядається як життєво важливий метод для управління та обробки великої кількості цифрових документів, які широко поширені та постійно зростають. Загалом, класифікація текстів відіграє важливу роль у отриманні та підсумовуванні інформації, пошуку тексту та відповідей на запитання.

Пропонується розв'язати задачу класифікації текстів методами машинного навчання із застосуванням методів обробки природньої мови.

Мета та задачі роботи. Метою даного дослідження є спрощення процесу класифікації текстів за рахунок створення моделей машинного навчання із застосуванням методів обробки природної мови.

Завдяки цьому користувачі можуть створювати власні моделі на обраних масивах інформації (дата сетах із раніше класифікованими текстами), тестувати їх при різних умовах, а також оцінювати ефективність кожної із моделей.

Процес класифікації текстів. Для того щоб класифікувати тексти необхідно пройти такі етапи:

- ознайомитися із доступними категоріями, на які можна розділити тексти;
- ознайомитися із переліком текстів та прочитати кожен текст;
- проаналізувати текст та віднести його до конкретної категорії.

Для автоматизації цього процесу необхідно вирішити задачу машинного навчання. Задачі машинного навчання можна поділити на 2 типи: навчання з учителем (supervised learning) та навчання без учителя (unsupervised learning).

При навчанні з учителем машина знає результати роботу алгоритму ще до того як почне працювати з ним або вивчати його. При навчанні без учителя машина сама намагається зрозуміти суть алгоритму. Було вирішено вирішувати цю задачу саме методом з учителем, адже необхідно не просто поділити текстові документи на категорії, а й вказати конкретну категорію кожного документу. Для вирішення будь-якої задачі машинного навчання методом з учителем необхідно мати початковий набір даних (дата сет). У процесі автоматизації система працює з документами та аналізує їх зміст.

У кінці автоматизації ми отримаємо систему, що дозволятиме користувачеві самостійно створювати моделі машинного навчання на основі власних дата сетів, навчати та аналізувати їх ефективність.

Програмний продукт, що пропонується дозволяє пройти процес навчання моделі самостійно, отримати оцінку кожної із моделей машинного навчання та обрати алгоритм, що найкраще підходить для вирішення конкретної задачі.

Постановка задачі. Введемо позначення, які будуть зустрічатися при розв'язанні задачі класифікації тестів методами обробки природньої мови та машинного навчання.

Текст – загалом зв'язна і повністю послідовна кінцева множина слів.

Слово – найменша самостійна і вільно відтворювана в мовленні відокремлено оформлена значима одиниця мови, набір символів [1].

Нехай маємо набір текстів, що були класифікованими за певними категоріями. Такі тексти для зручності наводяться у вигляді таблиці 1.

Таблиця 1

Представлення текстів

Тексти	Категорії
Текст 1	Категорія 1
…	…
Текст $k-1$	Категорія 1
Текст k	Категорія 2
…	…
Текст l	Категорія 2
…	…
Текст t	Категорія m
…	…
Текст n	Категорія m

Задача полягає у тому, аби однозначно визначити, до якої категорії відноситься текст, що не належить набору текстів, тобто не був класифікованим раніше.

Нехай маємо множину текстів *D*, що складається із довільних текстів:

$$D = \{d_1, d_2, ...d_n\}, \quad (1)$$

де $d_i, i = \overline{1,n}$ – конкретний текст, n – кількість текстів.

Нехай маємо множину категорій C, на які можна розділити дані тексти:

$$C = \{c_1, c_2, ...c_m\} \quad (2)$$

де $c_i, i = \overline{1,m}$ конкретний текст, m – кількість категорій.

Множина текстів D, кожен елемент якої класифікований по заданим категоріям із множини C називається початковий дата сетом, або просто дата сетом, тобто набором даних. Дата сет позначатимемо як Ds.

$$Ds = \begin{pmatrix} d_1 & c_1 \\ \vdots & \vdots \\ d_{k-1} & c_1 \\ d_k & c_2 \\ \vdots & \vdots \\ d_l & c_2 \\ \vdots & \vdots \\ d_t & c_m \\ \vdots & \vdots \\ d_n & c_m \end{pmatrix} \quad (3)$$

Дата сет побудований за допомогою невідомої цільової функції *F*:

$$F : C \times D \to \{0,1\} \quad (4)$$

Задача класифікації текстів методами машинного навчання полягає у тому, аби побудувати класифікатор $F`$, максимально наближений до F.

Класифікатором $F`$ зветься визначена на множині текстів D, функція яка однозначно зіставляє кожен текст із множини D конкретній категорії із множини C.

Розглянемо приклад такої задачі. Зауважимо, що даний приклад є незрівнянно малим у порівнянні із реальними задачами, які будемо розглядати у рамках дослідження.

Нехай маємо 10 текстів, що відповідають відгукам про ресторан. Тексти класифіковані за такою ознакою позитивний відгук, негативний відгук. Дата сет наведено у таблиці 2.

Таблиця 2

Приклад задачі для класифікації текстів

Текст	Категорія
On the up side, their cafe serves really good food.	Позитивний відгук
The only good thing was our waiter, he was very helpful and kept the bloody Mary's coming.	Позитивний відгук
Good prices.	Позитивний відгук
Although I very much liked the look and sound of this place, the actual experience was a bit disappointing.	Негативний відгук

Worst service to boot, but that is the least of their worries.	Негативний відгук
For a self-proclaimed coffee cafe, I was wildly disappointed.	Негативний відгук
Overall, I was very disappointed with the quality of food.	Негативний відгук
At first glance it is a lovely bakery cafe - nice ambiance, clean, friendly staff.	Позитивний відгук
Anyway, I do not think i will go back there.	Негативний відгук
Those burgers were amazing.	Позитивний відгук

Задача полягає у тому, аби класифікувати будь-які відгуки, що не належать даному дата сету за однією із двох категорій: позитивний відгук, негативний відгук.

Для вирішення цієї задачі ми використовуватимемо методи машинного навчання у поєднанні із обробкою природних мов. Планується розглядати такі алгоритми машинного навчання: Наївний Баєсів класифікатор (Naive Bayes Classifier) [6], Метод логістичної регресії (Logistic Regression) [3], Дерево ухвалення рішень (Decision Tree) [3] та Метод опорних векторів (Support Vector Machine) [4].

Опис методів розв'язання задачі. Для розв'язання задачі було використано 4 методи машинного навчання, які базуються на таких алгоритмах: логістична регресія, алгоритм наївного Байєса, опорних векторів та дерево ухвалення рішень.

Логістична регресія є методом підбору лінії регресії $y = f(x)$, у випадку коли y складається із даних, що можна представити у двійковому вигляді. Якщо відповідь, що необхідно отримати є двійковою (дихотомічною) змінною, а x є числовим значенням, то логістична регресія відповідає логістичний кривій функціонального відношення між x та y [3].

Перевагою класифікатора є простота реалізації.

Недоліками є невисока якість класифікації, значні обчислювальні витрати та факт того, що додавання текстів у тренувальну вибірку може значно вплинути на результат обчислення вагових коефіцієнтів, а значить і змінити характеристики моделі.

Класифікатор наївного Байєса є простим імовірнісним класифікатором, що заснований на застосуванні теореми Байєса з сильними (наївними) припущеннями про незалежність [6]. Перевага наївного Байєсівого класифікатора полягає в тому, що він вимагає лише невеликої кількості навчальних даних для оцінки необхідних параметрів класифікації.

Класифікатор Naive Bayes є, мабуть, найпростішим і найбільш широко використовуваним класифікатором. Він моделює розподіл документів у кожному класі з використанням імовірнісної моделі, припускаючи, що розподіл різних термінів незалежний один від одного.

Перевагами класифікатора є простота реалізації та низькі обчислювальні витрати. Недоліками є невисока якість класифікації.

Метод опорних векторів – це алгоритм, який визначає у якому місці необхідно розмежувати класи векторів, які належать певним групам. Він може бути застосований до будь-яких векторів, які кодують будь які дані. Найпростіше розглянути цей алгоритм у Декартовому просторі. Вирішуючи задачу класифікації текстів та перетворюючи тексти у вектори ми стикаємося із багатовимірними просторами, з якими легко справляється комп'ютер, але які важко зрозуміти людині [4].

Перевагами класифікатора є його ефективність на великих об’ємах даних та простота реалізації. До недоліків слід віднести високі обчислювальні витрати.

Дерево рішень є ієрархічним деревом навчальних екземплярів, в якому умова значення атрибуту використовується для поділу даних ієрархічно. Іншими словами, дерево рішень рекурсивно розділяє набір навчальних даних на менші підрозділи на основі набору тестів, визначених у кожному вузлі або гілці. Кожен вузол дерева є тестом, і кожна гілка, що спускається з вузла, відповідає одному значенню цього атрибута. Екземпляр класифікується, починаючи з кореневого вузла, перевіряючи атрибут цим вузлом і рухаючись вниз по гілці дерева відповідає значенню атрибута в даному екземплярі. І цей процес рекурсивно повторюється. У випадку текстових даних, умови на вершині дерева вершин зазвичай визначаються термінами в текстових документах. Наприклад, вузол може бути поділений на своїх дітей, спираючись на наявність або відсутність конкретного терміну в документі.

Крім цього, варто зазначити, що існує багато методів, які як окремо так і в поєднанні один з одним можуть значно покращувати ефективність роботи алгоритму.

Переваги класифікатора: ефективний на великих об’ємах даних, не потребує параметрів для навчання, інтуїтивно зрозумілий алгоритм, може бути представлений у вигляді скінченного набору правил.

Недоліками є: алгоритм легко “перевчити” на тестових даних. Оскільки дерева рішень використовують метод «поділяй і володарюй», вони, як правило, працюють добре, якщо існують декілька високо релевантних атрибутів, але менше, якщо присутні

багато складних взаємодій. Тобто із зростанням кількості класів ефективність алгоритму погіршуватиметься [5].

Результати дослідження та оцінка ефективності. Вирішуватимемо задачу класифікації текстів на прикладі дата сету із текстами зі статтями BBC. Дата сет містить 2225 текстів, розділених на 5 категорій. Гістограму розділення кількості текстів зображено на рисунку 1.

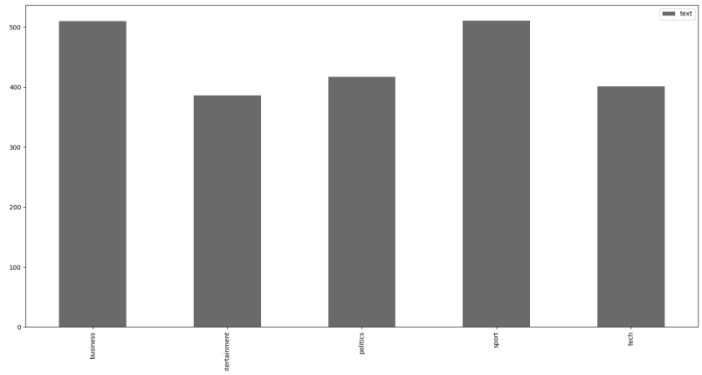


Рис. 1. Розподіл текстів між категоріями

Із рисунку видно, що дата сет містить такі категорії: business, politics, entertainment, sport, tech.

Розглянемо процес діяльності класифікації текстів після процесу автоматизації:

КРОК 1. Завантажити дата сет в систему.

КРОК 2. Вибрати алгоритми машинного навчання для створення моделей.

КРОК 3. Створити моделі та протестувати їх.

КРОК 4. Проаналізувати ефективність створених моделей.

КРОК 5. Класифікувати нові тексти.

Результат роботи програми у вигляді точностей моделей машинного навчання зображено на рисунку 2.

```
LR model created
The accuracy of Logistic Regression Algorithm is 0.9550561797752809
NB model created
The accuracy of Naive Bayes Algorithm is 0.9011235955056179
SVM model created
The accuracy of Support Vector Machine Algorithm is 0.9438202247191011
DT model created
The accuracy of Desicion Tree Algorithm is 0.7910112359550562
```

Рис. 2. Результат роботи програми

Є різні методи оцінки ефективності моделей машинного навчання. В дані статті використовуватимемо метод матриці невідповідностей. Матриця невідповідностей розміру $n \times n$ для моделі машинного навчання показує кількість текстів які були віднесенні до

конкретної категорії та кількість текстів які справді належать цій категорії. У даному випадку n – кількість категорії. [7]

Структура матриці невідповідностей для 2-х категорій представлено у таблиці 3.

Таблиця 3

Структура матриці невідповідностей для двох категорій.

	Передбачено, що текст не належить до категорії	Передбачено, що текст належить до категорії
Текст не належить до категорії	A	B
Текст належить до категорії	C	D

A – кількість передбачень, що текст не належить до категорії серед текстів які не належать до категорії;

B – кількість передбачень, що текст належить до категорії серед текстів які не належать до категорії;

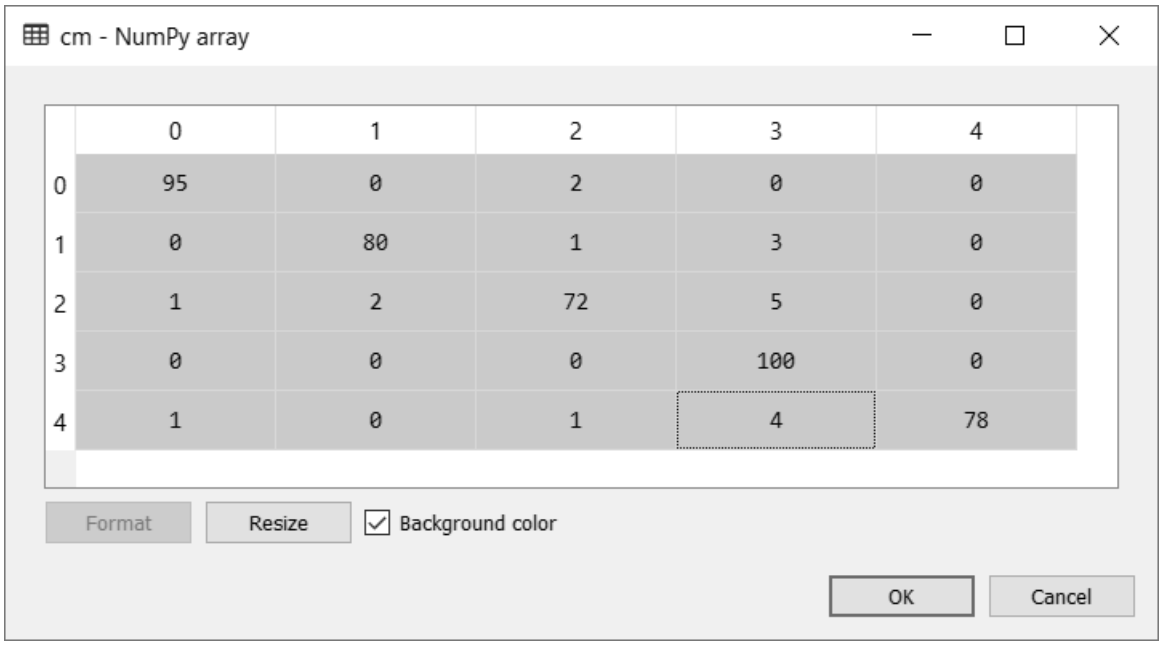
C – кількість передбачень, що текст належить до категорії серед текстів які не лежать до категорії;

D – кількість передбачень, що текст належить до категорії серед текстів які належать до категорії.

Розглянемо матрицю невідповідностей, що була отримана внаслідок класифікації текстів із дата сету, що розглядається моделями машинного навчання.

Дата сет містить 2225 текстів. Навчальна вибірка складається із 1780, а тренувальна вибірка складається із 445 текстів.

На рисунку 3 зображено сформовану матрицю невідповідностей для моделі машинного навчання створеної на основі алгоритму логістичної регресії.



	0	1	2	3	4
0	95	0	2	0	0
1	0	80	1	3	0
2	1	2	72	5	0
3	0	0	0	100	0
4	1	0	1	4	78

Рис. 3. Матриця невідповідностей для моделі машинного навчання створеної на основі алгоритму логістичної регресії

Колонка із номером 0 відповідає категорії ‘business’, колонка із номером 1 відповідає категорії ‘entertainment, колонка із номером 2 відповідає категорії ‘politics’, колонка із номером 3 відповідає категорії ‘sport’, колонка із номером 4 відповідає категорії ‘tech’.

Розглянемо перший рядок матриці невідповідностей. Сума елементів цього рядка є кількістю текстів із категорії ‘business’. Таким чином, можна зробити висновок, що створена на основі алгоритму логістичної регресії модель машинного навчання із 97 текстів, що відносяться до категорії ‘business’ правильно класифікувала 95 текстів. 2 тексти було віднесено до категорії ‘politics’.

Аналізуючи матрицю невідповідностей можна порахувати точність моделі машинного навчання як оцінку ефективності. Очевидно, що для того, аби порахувати точність, необхідно знайти відношення правильно класифікованих текстів до їх загальної кількості. Із матриці невідповідностей, що зображена на рисунку 3 легко побачити, що кількість правильно класифікованих текстів складає $95 + 80 + 72 + 100 + 78 = 425$. Загальна кількість текстів із тренувальної вибірки становить 445. Таким чином обчислюємо точність:

$$a = \frac{425}{445} = 0.95.$$

Із рисунка 2 видно, що точність обчислена програмно теж складає 0.95.

Аналогічно обчислюємо точність для інших алгоритмів. Результат представлено у таблиці 4.

Таблиця 4

Результат обчислення похибок

Алгоритм	Точність
Логістична регресія	0.95
Наївного байєса	0.90
Опорних векторів	0.94
Дерево ухвалення рішень	0.79

Отже, можемо зробити висновок, що для даного дата сету, доцільно використовувати алгоритм логістичної регресії. Таким чином користувачі системи можуть самі завантажувати дата сети та оцінювати моделі машинного навчання для того щоб обрати найбільш ефективну модель.

Висновки. У сучасному світі більшість інформації представлена у вигляді текстів. Тому зростає необхідність у створенні автоматизованих систем обробки текстових даних. Для того, щоб вирішити конкретну задачу класифікації текстів потрібно витратити багато ресурсів, що не є прийнятим особливо коли розглядаються комплексні задачі. Тому виникає необхідність у створенні універсального методу вирішення поставленої задачі. Використовуючи розроблену систему класифікації текстів кожен може створити моделі машинного навчання на основі обраних алгоритмів, оцінити ефективності моделей і обрати конкретну модель для вирішення задачі. Для створення такої системи було проаналізовано алгоритми машинного навчання та описано їх переваги та недоліки. На прикладі дата сету, що складається із текстів статей розділених по категорія було побудовано моделі машинного навчання, показано як оцінюється їх ефективність.

Література:

1. *Словник української мови: в 11 томах. — Том 9, 1978.*
2. *Daniel Jurafsky & James H. Martin. Copyright (2015) «Speech and Language Processing».*
3. *J Korean Acad Nurs Vol.43 No.2, 154-164 «An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain»*
4. *Manabu Sassano Fujitsu Laboratories, Japan «Virtual Examples for Text Classification with Support Vector Machines»*
5. *Barry de Ville, «Decision Trees for Business Intelligence and Data Mining: Using SAS Enterprise Miner», SAS Institute Inc., Cary, NC, USA, 2006.*
6. *Kevin P. Murphy, «Naïve Bayes classifier», Department of Computer Science, University of British Columbia, 2006.*

7. *Sofia Visa Computer Science Department College of Wooster Wooster, OH, USA «Confusion Matrix-based Feature Selection».*