УДК 004.42:004.8
DOI: https://doi.org/10.36994/2788-5518-2024-01-07-06

ПОРІВНЯННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ ФЕЙКОВИХ НОВИН

Джоші А., *Відкритий міжнародний університет розвитку людини «Україна», Інститут комп'ютерних технологій, Київ, Україна. https://orcid.org/0009-0003-7487-2354, arturjoshi.mailbox@gmail.com*

Павленко В. І.*, к.ф.-м.н., доц., Відкритий міжнародний університет розвитку людини «Україна», Інститут комп'ютерних технологій, Київ, Україна. https://orcid.org/0000-0002-3958-0415, pavlenko.v@i.ua*

Анотація. *У статті досліджено ефективність різних моделей машинного навчання для класифікації фейкових текстових новин. Розглянуто такі моделі: наївний Баєс, модель k-найближчих сусідів (kNN), модель опорних векторів (SVM) та модель випадкового лісу. Особливу увагу приділено впливу попередньої обробки даних, зокрема оверсемплінгу та використанню різних векторизаторів, на продуктивність моделей. У контексті наївного Баєса досліджується ефект згладжування на результати класифікації, визначаючи оптимальні параметри для поліпшення точності моделі. Для моделі kNN аналізується вплив кількості сусідів (k) на точність моделі, що дає змогу визначити найбільш відповідний параметр для даного завдання. Для моделі опорних векторів (SVM) випробувано різні перетворення, такі як лінійне, радіально-базисне функціональне (РБФ) та сигмоїдальне, із метою оцінки їхнього впливу на класифікацію текстових новин. Для випадкового лісу проведено аналіз того, як кількість дерев впливає на загальну ефективність моделі, що дає змогу знайти баланс між точністю та швидкістю навчання. Також розглянуто вплив різних методів векторизації тексту, таких як TF-IDF та CountVectorizer, на продуктивність моделей. Зокрема, проведено аналіз, як оверсемплінг впливає на точність моделей, даючи змогу краще справлятися з дисбалансом класів у даних. Результати дослідження надають важливу інформацію про оптимальні налаштування та підходи до попередньої обробки даних для поліпшення точності класифікації новин на достовірність. Отримані висновки можуть бути корисними для дослідників та практиків у галузі обробки природної мови та машинного навчання, які займаються розробленням систем для виявлення фейкових новин. Робота також підкреслює важливість ретельного вибору моделей та їхніх параметрів залежно від специфіки даних та поставлених завдань.*

Ключові слова: *класифікація текстових новин, моделі машинного навчання, попередня обробка даних, ефективність моделей машинного навчання.*

COMPARISON OF MACHINE LEARNING METHODS FOR FAKE NEWS DETECTION

Artur Joshi, *Open International University of Human Development «Ukraine» Kyiv, Ukraine. https://orcid.org/0009-0003-7487-2354, arturjoshi.mailbox@gmail.com*

Volodymyr Pavlenko*, Ph.D., Ass. Prof., Open University of Human Development «Ukraine», Kyiv, Ukraine. https://orcid.org/0000-0002-3958-0415, pavlenko.v@i.ua*

Abstract. *The paper examines the effectiveness of various machine learning models for the classification of fake text news. The following models are reviewed: Naive Bayes, k-nearest neighbors (kNN), support vector model (SVM), and random forest model. Special attention is paid to the impact of data preprocessing on the performance of models, in particular oversampling and the usage of various vectorizers. In the context of naive Bayes, the effect of smoothing on the classification effectiveness is investigated, determining the optimal parameters to improve the accuracy of the model. For kNN, the influence of the number of neighbors (k) on the accuracy of the model is analyzed, which allows determining the most appropriate parameter for this task. For the support vector model (SVM), various kernel functions such as linear, radial basis functional (RBF) and sigmoidal have been tested in order to evaluate their impact on text news classification. For a random forest, an analysis of how the number of trees affects the overall performance of the model is executed, which allows finding a balance between accuracy and learning speed. The impact of different text vectorization methods, such as TF-IDF and CountVectorizer, on the performance of the models is also evaluated. In particular, an analysis was made to investigate how oversampling affects the accuracy of the models, allowing to better cope with the imbalance of*

classes in the data. The research results provide valueable information about optimal settings and data preprocessing approaches to improve the accuracy of news classification for authenticity. The obtained conclusions can be useful for researchers and practitioners in the field of natural language processing and machine learning, who are engaged in the development of systems to detect fake news. The work also emphasizes the importance of careful selection of models and their parameters depending on the specifics of the data and tasks.

Key words: *text news classification, machine learning models, data preprocessing, machine learning models efficiency.*

Вступ

У сучасному інформаційному просторі фейкові новини стали серйозною проблемою, особливо в Україні, де дезінформація може мати далекосяжні наслідки як для суспільства, так і для держави. Вплив фейкових новин на громадську думку та політичні процеси потребує ефективних інструментів для їх виявлення та класифікації. У цьому контексті машинне навчання виступає потужним засобом для автоматизованого аналізу та розпізнавання неправдивої інформації.

Для поліпшення ефективності автоматизованого аналізу та розпізнавання фейкових новин важливо порівняти різні методи машинного навчання для класифікації, зокрема наївного Баєсового класифікатора, алгоритму k-найближчих сусідів (KNN), методу опорних векторів (SVM) та випадкового лісу (Random Forest). Варто проаналізувати ефективність кожного фз цих методів на основі низки критеріїв, таких як точність, F-міра, чутливість.

На ефективність вищезгаданих алгоритмів можуть впливати різні чинники, такі як формат та попередня обробка вхідних даних, а також власні параметри алгоритмів. Варто зазначити, що однакова зміна до попередньої обробки вхідних даних може позитивно впливати на одні алгоритми і негативно – на інші, тому важливо проаналізувати цей вплив, а також дати йому кількісну оцінку. Також кожен з алгоритмів має власні параметри, підбір яких впливає на ефективність виконання конкретно взятого завдання.

Виконання досліджень

Машинне навчання – це галузь штучного інтелекту, яка займається розробленням алгоритмів і моделей, що дають змогу комп'ютерам навчатися з даних і робити прогнози або приймати рішення без явного програмування для виконання конкретних завдань. Основна ідея машинного навчання полягає у тому, що система може автоматично покращувати свою продуктивність шляхом аналізу великої кількості даних, знаходження закономірностей та використання цих закономірностей для майбутніх прогнозів або класифікацій. Основними видами машинного навчання є кероване та некероване навчання. За умови наявності якісного набору даних кероване навчання є потужним інструментом для аналізу як тексту в цілому, так і для класифікації тексту за певними критеріями.

До керованих методів машинного навчання відносять наївний Баєсовий класифікатор (Naive Bayes), алгоритм k-найближчих сусідів (kNN), метод опорних векторів (SVM) та випадковий ліс (Random Forest). Ці алгоритми відіграють ключову роль у задачах класифікації та прогнозування. Наївний Баєс базується на теоремі Байєса і передбачає незалежність ознак, що робить його швидким та ефективним для великих наборів даних, проте він може бути менш точним у випадках складних залежностей між ознаками. Алгоритм k-найближчих сусідів здійснює класифікацію, порівнюючи тестовий зразок із k найближчими зразками з навчального набору, що робить його простим у реалізації, по суті, він не вимагає навчання як такого, проте є ресурсоємним для великих даних. Метод опорних векторів (SVM) використовує гіперплощини для розділення класів у багатовимірному просторі, що дає йому змогу ефективно працювати з багатомірними даними та складними межами класифікації. Випадковий ліс, який складається з множини дерев рішень, є потужним методом завдяки своїй здатності знижувати вплив перенавчання, однак він може вимагати значних обчислювальних ресурсів.

Під час розгляду алгоритмів класифікації текстових даних у моделях машинного навчання важливо враховувати те, як репрезентується текст у вектори, що дають змогу алгоритмам обробляти ці дані. Текстові дані у найпростішому вигляді уявляються у вигляді n-вимірного вектору X, де n – довжина словника, Xi – кількість появ і-го слова у даному тексті. Словник формується під час тренування моделі і складається з усіх слів, що зустрічалися в тренувальному наборі даних. Очевидно, що набір даних потребує попередньої обробки перед тренуванням, оскільки необхідно виключити вплив лінгвістичних особливостей певної природньої мови на результати класифікації. Зазвичай для цих цілей увесь текст зводиться до нижнього регістру, прибираються знаки пунктуації та проводиться стемінг – процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс. Таким чином, вектор X складається з n цілочисельних елементів xi, які є кількістю разів, з якою і-е слово зустрічається в тексті. У разі керованих методів машинного навчання кожен елемент набору даних містить у собі клас Yk, до якого належить даний елемент.

Баєсів класифікатор базується на теоремі Баєса, що визначає ймовірність певної події за умови істинності іншої взаємопов'язаної події.

$$P(A|B)=\frac{P(B|A)*P(A)}{P(B)}\ , \tag{1}$$

де $P(B|A)$ – імовірність події *B* за умови істинності події *A, P(A), P(B)* – імовірності настання подій *A* та *B* відповідно.

Нехай Y_k – подія, що визначає приналежність елементу набору даних до класу *k*. Застосувавши теорему Баєса для побудови класифікатора, отримуємо:

$$P(Y_k|x_1,x_2,\ldots,x_n)=\frac{P(x_1,x_2,...,x_n|Y_k)*P(Y_k)}{P(x_1,x_2,...,x_n)} \tag{2},$$

$$P(Y_k|x_1,x_2,\ldots,x_n)\propto P(x_1,x_2,...,x_n|Y_k)*P(Y_k) \tag{3},$$

$$P(Y_k|x_1,x_2,\ldots,x_n)\propto P(x_1|Y_k)*P(x_2|Y_k)*\ldots P(x_n\,|\,Y_k)*P(Y_k) \tag{4}$$

Отже, імовірність того, що даний вектор належить класу Y_k, пропорційна ймовірності появи цього класу в цілому і помножена на умовну ймовірність приналежності кожного елементу вектору до цього класу. Із цього одразу випливає особливість Баєсового класифікатора, яка полягає у тому, що нерівномірний розподіл класів у тренувальному наборі даних може впливати на результат класифікації, оскільки чим частіше зустрічається певний клас у тренувальному наборі даних, тим частіше класифікатор буде тяжіти до цього класу в тестовому наборі [2]. Це особливо важливо для класифікації текстових фейкових новин, оскільки для коректної класифікації фейкових новин необхідно, щоб кількість фейкових новин була приблизно рівною кількості правдивих новин. Якщо набір даних не є однорідним, існують техніки, за допомогою яких можна зменшити вплив цього ефекту, котрі будуть описані далі.

Метод *k* найближчих сусідів ґрунтується на ідеї, що приналежність даного елемента набору даних до певного класу визначається дистанцією цього елемента до елементів із тренувального набору. Як дистанція зазвичай береться евклідова відстань.

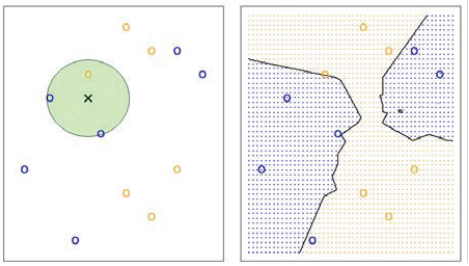

Рис. 1. Двовимірна модель *k* найближчих сусідів

На рис. 1 зображено приклад роботи класифікатора для 2-вимірного випадку. Аналогічно можна побудувати n-вимірний класифікатор тексту, де n – довжина словника. Очевидно, що дана модель на відміну від Баєсового класифікатора не має залежності ефективності класифікації від розподілу класів у тренувальному наборі даних [3].

Метод опорних векторів (SVM) полягає у пошуку рівняння гіперплощини, що максимізує евклідову відстань до елементів різних класів. На рис. 2 зображено приклад класифікатора для 2-вимірного простору.

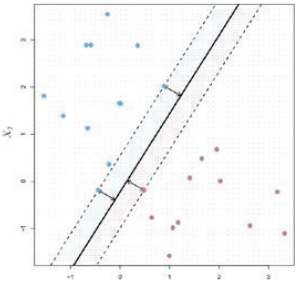

Рис. 2. Двовимірна модель опорних векторів

Варто зауважити, що метод опорних векторів потребує додаткової попередньої обробки даних, якщо межі класів не є лінійними [4].

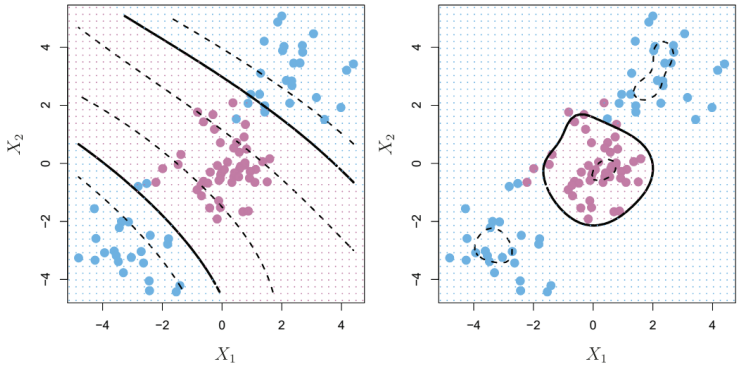

Рис. 3. Попередня обробка набору даних для моделі опорних векторів

У такому разі необхідно здійснити перехід із n-мірного простору до *n+1* мірного простору за допомогою певного відображення $K(x_i, x_i')$.

Метод випадкового лісу – це ансамблевий алгоритм машинного навчання, який використовується для задач класифікації та регресії. Він складається з множини дерев рішень і базується на принципі побудови кількох дерев рішень та об'єднання їх результатів. Для кожного вузла дерева вибирається випадкова підмножина ознак замість використання всіх ознак. Це додатково знижує кореляцію між деревами і допомагає уникнути перенавчання. Для задачі класифікації новий зразок проходить через кожне дерево, і кожне дерево дає свій прогноз класу. Кінцевий прогноз визначається голосуванням більшості.

Рис. 4. Метод випадкового лісу

Для порівняння ефективності різних моделей машинного навчання використовуються різноманітні метрики. Перш ніж розглянути їх, необхідно ввести поняття матриці помилок.

Таблиця 1

Можливі результати бінарної класифікації

	y = 1	y = 0
$y_0 = 1$	Істинно позитивні (True Positive, TP)	Хибно позитивні (False Positive, FP)
$y_0 = 0$	Хибно негативні (False Negative, FN)	Істинно негативні (True Negative, TN)

Найпростішою метрикою є точність – частка правильних відповідей алгоритму в цілому. Потрібно зазначити, що дана метрика не є репрезентативною у разі тренувальних наборів даних із нерівними класами.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Більш репрезентативними метриками є прецизійність (precision) по кожному класу окремо та повнота.

$$precision = \frac{TP}{TP + FP} \tag{6},$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

Прецизійність є часткою елементів, що були класифіковані до певного класу і при цьому дійсно є елементами цього класу. Повнота показує, яку частку елементів класу знайшла модель.

Для об'єднання обох метрик в одну агреговану метрику використовується f-міра.

$$F_\beta = \left(1+\beta^2\right) * \frac{precision * recall}{\left(\beta^2 * precision\right) + recall} \quad (8)$$

Для поліпшення ефективності класифікаційних моделей варто дослідити вплив як попередньої обробки вхідних даних, так і власних параметрів моделей. У своїй роботі Ahmed H. Aliwy та Esraa H. Abdul Ameer запропонували низку поліпшень для збільшення точності моделей, таких як використання різних перетворень для моделі SVM, використання різних мір дистанції для моделі KNN, а також видалення дублікатів слів для наївного Баєсового класифікатора [1]. Для більш репрезентативної оцінки впливу вищезазначених чинників необхідно також дослідити кількісні характеристики й інших метрик, таких як прицизійнсть, повнота та f-міра.

Завданням даного дослідження є проведення експерименту задля оцінки впливу попередньої обробки вхідних даних і власних параметрів на ефективність моделей класифікації текстових новин українською мовою. Як вхідні дані взято 10 735 текстових новин з інформаційних телеграм-каналів, які розміщені на вебресурсі Kaggle [5]. Даний набір даних містить 8 237 достовірних і 2 498 фейкових новин.

Вхідний набір даних містить у собі необроблені дані в оригінальному вигляді. Задля поліпшення ефективності класифікаційних моделей необхідно виконати попередню обробку вхідних даних. До основних технік обробки вхідних даних відносять фільтрацію стоп-слів, лематизацію та стемінг. Фільтрація стоп-слів є процесом, який видаляє з вхідного набору даних певні слова, що містяться у заданому словнику. Лематизація – це комплексний підхід, що базується на визначенні основи слова. Першим кроком цього алгоритму є визначення частин мови у реченні. На другому кроці до слова застосовуються певні правила відповідно до частини мови. У результаті лематизації вхідне слово зводиться до слова у його базовій формі, наприклад інфінітив для дієслова або називний відмінок для іменника. Стемінг є процесом відкидання суфіксів та префіксів, завдяки чому є швидшим за лематизацію. Як попередню обробку вхідних даних було вибрано фільтрацію стоп-слів [6] та стемінг. На рисунках зображено частотні розподіли слів до та після попередньої обробки. Попередня обробка дала змогу вирівняти розподіл частот різних слів, а також прибрати несуттєві частини мови, що могли значно спотворити результати класифікації.

Рис. 5. Діаграма частотності слів до фільтрації

Рис. 6. Діаграма частотності слів після фільтрації

Важливою технікою попередньої обробки даних є оверсемплінг. Оверсемплінг – це метод боротьби з дисбалансом класів у наборах даних для машинного навчання. Коли одна з категорій

або класів у тренувальному наборі даних значно переважає інші, це може призвести до того, що модель навчиться ігнорувати менш представлені класи. Така модель буде добре працювати на переважаючому класі, але погано на рідкісних класах. Оверсемплінг передбачає збільшення кількості елементів менш представлених класів, щоб збалансувати кількість елементів для кожного класу. Це може бути досягнуто шляхом дублювання наявних елементів або створення нових синтетичних елементів.

Наступним кроком попередньої обробки вхідних текстових даних є векторизація. Векторизація є ключовим процесом у машинному навчанні та обробці природної мови, який перетворює текстові дані на числові вектори, що можуть бути оброблені алгоритмами машинного навчання. Оскільки алгоритми машинного навчання працюють із числовими даними, векторизація дає змогу представити текст у вигляді числових значень, зберігаючи інформацію про структуру та зміст тексту. Основна мета векторизації полягає у тому, щоб створити числові вектори, які відображають зміст та контекст тексту. Це досягається шляхом кодування кожного слова або фрази у вигляді вектора фіксованої довжини. Ці вектори можуть бути дуже простими, наприклад такими, що відображають наявність або відсутність слова у документі, або складнішими, такими як контекстуальні вектори, що враховують взаємозв'язки між словами у тексті. Векторизація тексту вирішує кілька важливих завдань. По-перше, вона дає змогу обробляти текстові дані на основі їхніх частотних характеристик, таких як частота появи слів у документі або зворотна частота документа, що знижує вплив загальних слів і підвищує значущість специфічних термінів. По-друге, вона дає змогу алгоритмам машинного навчання виявляти семантичні та синтаксичні патерни у тексті, що важливо для задач, пов'язаних із розумінням природної мови, таких як класифікація текстів, аналіз настроїв та пошук інформації.

Існує багато методів векторизації тексту, основним із яких є CountVectorizer, який перетворює колекцію текстових документів на матрицю термін-документ. У цій матриці кожен рядок відповідає одному документу, а кожен стовпець відповідає одному слову зі словника, створеного на основі всіх документів. Значення в матриці вказують кількість разів, коли слово з'являється у відповідному документі. Term Frequency-Inverse Document Frequency Vectorizer – це вдосконалений метод векторизації тексту, який ураховує не лише кількість появ слова у документі, а і його частотність у наборі даних у цілому. Це досягається шляхом обчислення TF-IDF значень для кожного слова. TF вимірює частоту появи слова в документі. Вона розраховується для кожного слова у кожному документі окремо. Особливістю TFIDF векторизації є зменшення ваги слів, що часто зустрічаються у тексті, та зменшення впливу шуму.

$$TF(t,d) = \frac{Кількість\ появ\ слова\ t\ в\ елементі\ d}{Загальна\ кількість\ слів\ в\ елементі\ d} \tag{9}$$

IDF оцінює рідкість слова в корпусі документів. Слова, які часто зустрічаються у багатьох документах, отримують меншу вагу.

$$IDF(t,d) = \log\left(\frac{N}{|d \in D : t \in d|}\right) \tag{10}$$

N – загальна кількість документів у наборі даних.

Розглянемо вплив оверсемплінгу та вибору векторизатора на ефективність вищезгаданих моделей.

Таблиця 2

Вплив попередньої обробки даних на ефективність моделей

		З оверсемпінгом	**Без оверсемплінгу**	**CountVectorizer**	**TFIDF Vectorizer**
Наївний Баєс	Точність	78,80%	72,60%	77,80%	77,80%
	Прецизійність (фейкові)	53,80%	44,20%	52,20%	52,20%
	Повнота (фейкові)	54,60%	71,80%	49,80%	50,80%
	F-міра (фейкові)	54,20%	54,80%	51,00%	51,40%
	Прецизійність (достовірні)	86,40%	89,60%	85,00%	85,00%
	Повнота (достовірні)	85,60%	72,60%	86,40%	85,80%
	F-міра (достовірні)	86,20%	80,20%	86,00%	85,60%

Продовження таблиці 2

Метод k найближчих сусідів	Точність	74,00%	77,20%	58,60%	74,60%
	Прецизійність (фейкові)	35,40%	58,40%	32,80%	35,80%
	Повнота (фейкові)	14,60%	0,80%	44,60%	14,40%
	F-міра (фейкові)	21,00%	1,60%	29,20%	20,60%
	Прецизійність (достовірні)	78,00%	77,20%	80,20%	79,00%
	Повнота (достовірні)	91,80%	100,00%	63,60%	92,00%
	F-міра (достовірні)	84,40%	87,00%	64,40%	85,00%
Метод опорних векторів	Точність	83,20%	80,00%	84,40%	83,80%
	Прецизійність (фейкові)	88,60%	98,60%	92,20%	89,00%
	Повнота (фейкові)	33,80%	15,40%	37,60%	35,80%
	F-міра (фейкові)	49,20%	26,80%	53,20%	50,60%
	Прецизійність (достовірні)	82,80%	79,20%	83,60%	83,20%
	Повнота (достовірні)	98,60%	100,00%	99,00%	98,60%
	F-міра (достовірні)	89,80%	88,20%	90,60%	90,40%
Випадковий ліс	Точність	91,20%	90,80%	90,80%	91,20%
	Прецизійність (фейкові)	90,20%	95,00%	91,40%	89,20%
	Повнота (фейкові)	70,20%	63,00%	68,60%	68,80%
	F-міра (фейкові)	79,00%	75,80%	78,40%	77,80%
	Прецизійність (достовірні)	91,40%	89,80%	91,00%	91,40%
	Повнота (достовірні)	97,80%	99,00%	98,00%	97,60%
	F-міра (достовірні)	94,40%	93,80%	94,60%	94,40%

Із результатів, поданих у таблиці, можна зробити такі висновки:

1. Для всіх моделей, окрім методу k найближчих сусідів, оверсемплінг підвищує загальну точність на 0,4–6,2%. Значний вплив оверсемплінгу на модель наївного Баєса пояснюється, передусім, більш рівномірним розподілом елементів обох класів у тренувальному наборі та, як наслідок –збільшенням апріорної ймовірності меншого класу.

2. Оверсемплінг має позитивний вплив на прецизійність та повноту моделі наївного Баєса.

3. Модель k найближчих сусідів практично не здатна знаходити фейкові новини за умови відсутності оверсемплінгу. Це може пояснюватися тим, що оверсемплінг, дублюючи точки, що репрезентують елементи з класу фейкових новин, збільшує вагу цієї точки під час класифікації.

4. Оверсемплінг позитивно впливає на пошук фейкових новин, використовуючи модель опорних векторів. Збалансування кількості зразків кожного класу допомагає моделі знайти більш точну гіперплощину, збільшує кількість підтримуючих векторів і знижує ризик перенавчання.

5. Прецизійність та повнота моделі випадкового лісу практично не зазнають впливу оверсемплінгу. Це може пояснюватися тим, що під час побудови кожного дерева випадковий ліс випадково вибирає підмножину ознак для розділення вузлів. Також рішення випадкового лісу базується на голосуванні багатьох дерев. Навіть якщо окремі дерева можуть бути схильні до домінування одного класу, загальний результат голосування буде менш упередженим завдяки різноманітності дерев.

6. Обидва векторизатори мають приблизно однакові результати для моделі наївного Баєса, опорних векторів та випадкового лісу. Це може буде зумовлено тим, що ці моделі відносно стабільні до різних методів векторизації, оскільки їхні алгоритми базуються на відносних частотах або вагових пропорціях слів у документах. TF-IDF фактично масштабує частоти слів, але це не змінює відносні пропорції настільки, щоб суттєво вплинути на кінцеві результати моделі.

7. Використання TFIDF-векторизатору для методу k найближчих сусідів значно підвищує точність та повноту пошуку саму достовірних новин. Це може бути зумовлено тим, що TFIDF-векторизатор ураховує не лише частоту слів, але й їх важливість, зменшуючи вагу часто вживаних слів і підвищуючи вагу рідкісних. Це змінює відстані між текстовими векторами і може суттєво вплинути на результати моделі kNN, покращуючи її здатність до класифікації.

Також варто проаналізувати вплив власних параметрів моделей на їх ефективність. Наприклад, параметр α в наївному Баєсі використовується для додаткової регуляризації (зглажування). Його основна мета – уникнути проблеми нульової ймовірності для слів, які не зустрічаються в тренувальних даних для певного класу. Без регуляризації, якщо слово ніколи не з'являється в тренувальних даних для певного класу, його ймовірність буде оцінюватися як нуль, що призведе до того, що обчислювана ймовірність належності нового елементу до цього класу також буде нульовою, навіть якщо всі інші ознаки сильно вказують на цей клас.

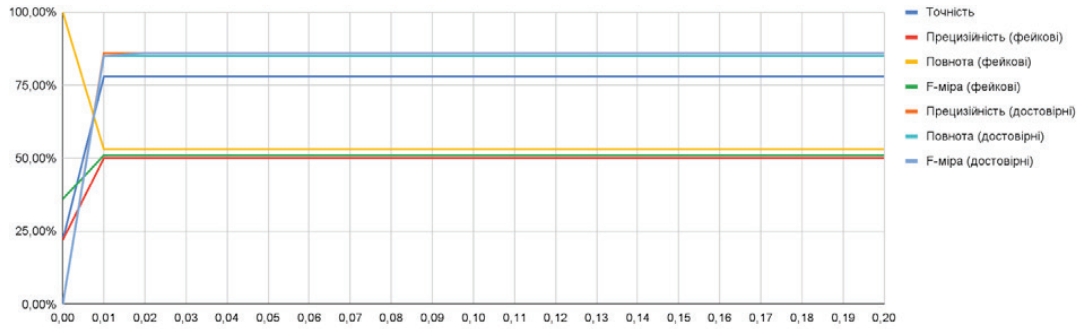

Рис. 7. Вплив регуляризації на ефективність моделі наївного Баєса

На діаграмі показана залежність метрик ефективності моделі наївного Баєса від параметру альфа. Із графіку випливає, що за значення альфа, що дорівнює 0, модель класифікує всі елементи набору даних як фейкові. Використавши навіть невелике ненульове значення alpha, можна привести модель до робочого стану. Незважаючи на те що за дуже великих значень альфа-метрики трохи покращуються, не рекомендується використовувати великі значення, оскільки це може зробити модель надто загальною, оскільки ймовірності будуть надто згладжені, і різниця між частотами слів для різних класів буде зменшена. У результаті модель може втратити здатність ефективно розрізняти класи на основі важливих слів.

Модель k найближчих сусідів має низку параметрів, що здатні впливати на ефективність моделі: k – кількість найближчих точок, до яких рахується відстань; однакові вагові коефіцієнти, або пропорційні відстані; використання манхетенської абревклідової відстані.

Із графіку можна помітити, що після k = 5 відзначається різке зменшення якості класифікації фейкових новин. Також експерименти не показали, що використання манхетенської відстані замість евклідової або пропорційні вагові коефіцієнти суттєво позитивно впливають на якість класифікації.

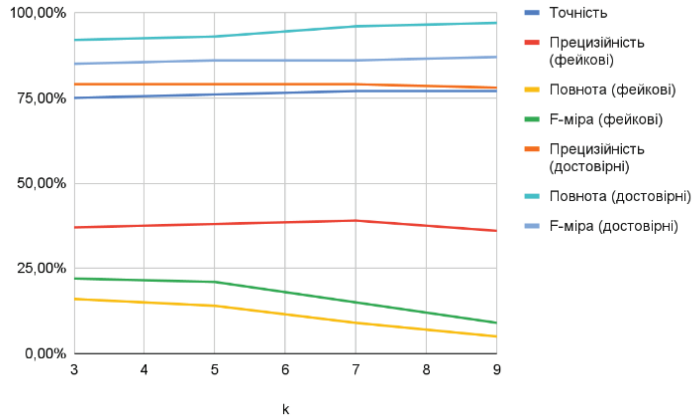

Рис. 8. Вплив параметра k на ефективність моделі k найближчих сусідів

Ефективність моделі опорних векторів має високу залежність від відображення, що використовується за переходу між векторними просторами. Із наступного графіку можна помітити, що для задач класифікації текстових новин найкраще підходить сигмоїдна функція. Зменшення прецизійності класифікації фейкових новин компенсується повнотою, що відображено в агрегованій метриці F-міра.

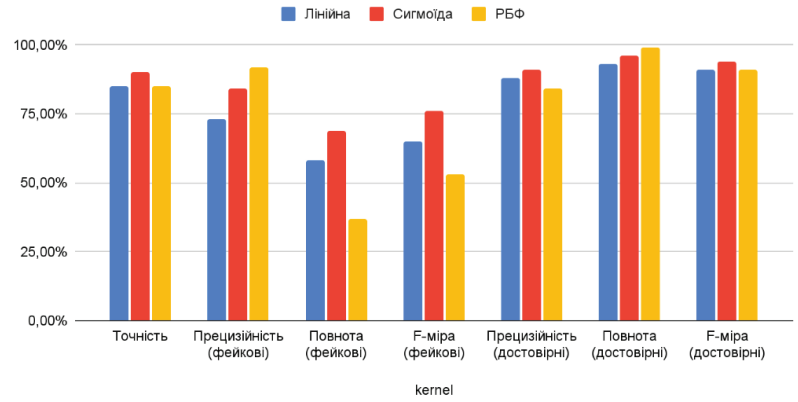

Рис. 9. Залежність ефективності моделі опорних векторів від вибраного перетворення

Ефективність моделі випадкового лісу значною мірою залежить від кількості дерев, що входять до його складу. Слід зауважити, що зі збільшенням кількості дерев збільшується час тренування та класифікації. Із графіку нижче можна зробити висновок, що 50 дерев є оптимальною кількістю для даної задачі. Ураховуючи, що час тренування моделі лінійно зростає зі збільшенням кількості дерев, можна зробити висновок про недоцільність подальшого збільшення кількості дерев у випадковому лісі.

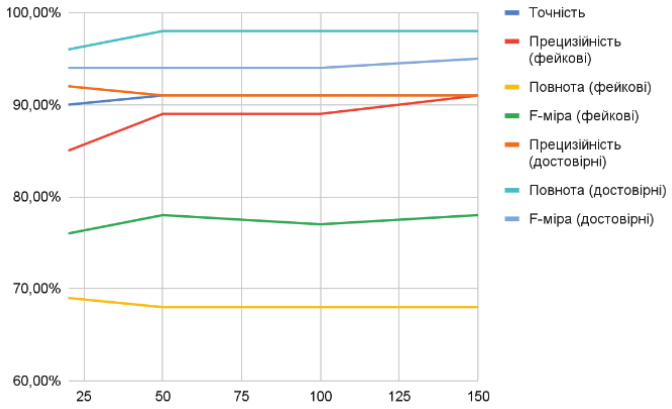

Рис. 10. Залежність ефективності моделей від кількості дерев у моделі випадкового лісу

Отже, додавання нових дерев зменшує варіативність результатів за рахунок агрегації, але після досягнення певної кількості дерев цей ефект стає незначним, оскільки більша частина варіативності вже знижена. Кожне додаткове дерево приносить дедалі менше нової інформації, оскільки більшість важливих патернів вже було враховано попередніми деревами. Після досягнення певного порогу додавання нових дерев майже не впливає на точність, оскільки нові дерева вносять мінімум нової інформації. Зі збільшенням кількості дерев зростають обчислювальні витрати для тренування та прогнозування. Із певного моменту додавання нових дерев не виправдовується через незначне поліпшення точності та збільшення часу обробки.

Висновки

У рамках дослідження було проведено огляд моделей машинного навчання для класифікації текстових новин на предмет достовірності. Розглянуто наївний Баєсів класифікатор, модель k найближчих сусідів, модель опорних векторів і модель випадкового лісу. Розглянуто методи попередньої обробки даних, а саме: фільтрацію, стемінг, оверсемплінг та векторизацію. Проведено аналіз впливу оверсемплінгу та векторизації на якість класифікації, а також вплив власних параметрів моделей на їх ефективність. Було визначено таке:

1. Найефективнішою моделлю для визначення фейкових новин є модель випадкового лісу. Дана модель показала ефективність визначення фейкових новин рівну 79% за метрикою F-міра та 94,4% для достовірних новин. Дещо меншу ефективність, на 15% за метрикою F-міра для достовірних та фейкових новин, показала модель наївного Баєса. Модель k найближчих сусідів має достатньо низьку ефективність класифікації фейкових новин – 21% за метрикою F-міра.

2. Найбільш значний вплив оверсемплінгу прослідковується для моделі наївного Баєса за рахунок його високої залежності від апріорної ймовірності класів та моделі опорних векторів за рахунок пошуку більш точної гіперплощини. Вплив оверсемплінгу на модель випадкового лісу менше 1% за метрикою F-міра для достовірних та фейкових новин.

3. Вплив використання TFIDF-векторизатора порівняно з простим CountVectorizer є не більше 2% за метрикою F-міра для достовірних та фейкових новин. Використання TFIDF-векторизатора покращує визначення достовірних новин на 20% за метрикою F-міра для моделі k найближчих сусідів за рахунок зменшення ваги часто вживаних слів, проте погіршує визначення фейкових новин на 8%.

4. Додаткова регуляризація є обов'язковою для моделі наївного Баєса, хоча і не вимагаються високі значення цього параметра: використання значень більших за 0,01 не призводять до підвищення ефективності класифікації.

5. Модель k найближчих сусідів не демонструє росту ефективності за збільшення кількості точок, що беруть участь у класифікації. Найвищу ефективність дана модель показує за значень k від 3 до 5.

6. Сигмоїда показала найкращі показники ефективності як відображення для моделі опорних векторів.

7. Модель випадкового лісу демонструє оптимальні показники ефективності за досить невисоких значень кількості дерев. Так, за класифікації набору даних обсягом близько 10 000 новин випадковий ліс із 50 дерев показує ефективність 80% за метрикою F-міра для визначення фейкових новин та 94% – для визначення достовірних новин.

Література

1. Ahmed H. Aliwy, Esraa H. Abdul Ameer. Comparative Study of Five Text Classification Algorithms with their Improvements. *International Journal of Applied Engineering Research* Vol. 12, 2017, Number 14, ISSN 0973-4562. pp. 4309-4319

2. Jason D.M. Rennie. Improving Multi-class Text Classification with Naive Bayes. *Massachusetts institute of technology – artificial intelligence laboratory.* 2001.

3. Songbo Tan. An effective refinement strategy for KNN text classifier. *Institute of Computing Technology, Chinese Academy of Sciences.* 2006.

4. Simon Tong, Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Computer Science Department Stanford University.* 2001.

5. Набір даних фейкових та достовірних українських новин. URL: https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news

6. Набір стоп-слів української мови. URL: https://github.com/skupriienko/Ukrainian-Stopwords/blob/master/stopwords_ua.txt

References

1. Ahmed H. Aliwy, Esraa H. Abdul Ameer. Comparative Study of Five Text Classification Algorithms with their Improvements. *International Journal of Applied Engineering Research* Vol. 12, 2017, Number 14, ISSN 0973-4562. pp. 4309-4319

2. Jason D.M. Rennie. Improving Multi-class Text Classification with Naive Bayes. *Massachusetts institute of technology – artificial intelligence laboratory.* 2001.

3. Songbo Tan. An effective refinement strategy for KNN text classifier. *Institute of Computing Technology, Chinese Academy of Sciences.* 2006.

4. Simon Tong, Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Computer Science Department Stanford University.* 2001.

5. Ukrainian fake and true news dataset. URL: https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news

6. Ukrainian stopwords dataset. URL: https://github.com/skupriienko/Ukrainian-Stopwords/blob/master/stopwords_ua.txt