НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

«На правах рукопису»
УДК_____

«До захисту допущено»
Завідувач кафедри
_____ Наталія АУШЕВА
«___»_____ 2023 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою
«Цифрові технології в енергетиці»
зі спеціальності 122 «Комп'ютерні науки»

на тему: Класифікація контенту фінансових новин на основі машинного навчання

Виконав: студент 2 курсу, групи ТР-23мп
Алі-Шах Імран Гохарович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к. т. н. Світлана ШАПОВАЛОВА _____
(посада, вчене звання, науковий ступінь, ім'я ПРІЗВИЩЕ) (підпис)

Рецензент доцент кафедри ІПЗЕ, к. т. н. Валерій КУЗЬМІНИХ _____
(посада, вчене звання, науковий ступінь, ім'я ПРІЗВИЩЕ) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ - 2023

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ
Рівень вищої освіти другий (магістерський)
За освітньою програмою «Цифрові технології в енергетиці»
Спеціальності 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Наталія АУШЕВА
(підпис)
«____» _____ 2023 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Алі-Шаху Імрану Гохаровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Класифікація контенту фінансових новин на основі машинного навчання
Науковий керівник Шаповалова Світлана Ігорівна, к. т. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «06» листопада 2023 року №5152-с

2. Строк подання студентом дисертації: 18 грудня 2023 р.

3. Вихідні дані до роботи: мова програмування Python, інтерактивне хмарне середовище Google Colab.

4. Перелік завдань, які потрібно розробити: Провести аналіз існючих рішень за темою роботи, запропонувати власні рішення. Розробити моделі машинного навчання для розпізнавання характеру новин. Обрати набір даних для побудов моделей. Навчити моделі машинного навчання. Провести порівняння моделей машинного навчання за обчисленими метриками оцінювання моделей.

5. Орієнтований перелік ілюстративного матеріалу: Екранні форми, що демонструють побудову моделі, а також результати моделювання.

6. Орієнтований перелік публікацій Ali-Shakh Imran, Shapovalova Svitlana. The classification of financial news content using machine learning. Integration of science as a mechanism of effective development: Proceedings of the XI International Scientific and Practical Conference. Helsinki, Finland. November 28 – December 1, 2023. P. 376-377. DOI – 10.46299/ISG.2023.2.11.

7. Дата видачі завдання «24» жовтня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Розробка змісту магістерської дисертації та видача завдання науковим керівником.	24.10.2022	виконано
2	Підбір літератури по темі дослідження та її аналіз.	31.10.2022 - 11.01.2023	виконано
3	Постановка задачі класифікації контенту фінансових новин на основі машинного навчання.	14.01.2023 - 04.02.2023	виконано
4	Проведення аналізу існуючих рішень .	06.02.2023 - 04.04.2023	виконано
5	Підбір засобів розробки моделей машинного навчання.	06.04.2023 - 08.06.2023	виконано
6	Підбір необхідного набору даних.	13.06.2023 - 18.08.2023	виконано
7	Розробка та навчання моделей.	22.08.2023 - 24.11.2023	виконано
8	Предзахист	06.12.2023	виконано
9	Оформлення магістерської дисертації	01.11.2023 - 18.12.2023	виконано
10	Захист		

Студент _____ Імран АЛІ-ШАХ
(підпис) (ім'я ПРІЗВИЩЕ)

Науковий керівник _____ Світлана ШАПОВАЛОВА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми дослідження. У сучасному швидкоплинному світі фінансів, здатність швидко і точно аналізувати статті і новини має першочергове значення. Фінансові новини містять цінну інформацію, яка може суттєво впливати на ринки, впливати на торгові рішення та формувати настрої інвесторів. Однак величезний обсяг і складність фінансових даних роблять аналіз вручну непрактичним. Машинне навчання, як підмножина штучного інтелекту, дозволяє комп'ютерам вчитися на основі даних, виявляти закономірності та приймати рішення без явного програмування. У сфері класифікації фінансових новин вищезазначені алгоритми можна навчити класифікувати контент новин за попередньо визначеними настроями або темами, надаючи своєчасну інформацію та допомагаючи в процесах прийняття рішень.

Метою дослідження є реалізація методів машинного навчання для класифікації контенту фінансових новин.

Завдання дослідження:

1. Провести аналіз існуючих рішень для класифікації контенту фінансових новин.
2. Обрати набір даних, методи та рішення, які найкраще відповідають меті дослідження, обґрунтувати цей вибір.
3. Виконати навчання моделей на основі обраного набору даних.
4. Виконати порівняльну оцінку моделей, необхідних для виконання цієї роботи.

Об’єкт дослідження – процес класифікації текстових повідомлень за емоційним забарвленням.

Предмет дослідження – моделі машинного навчання для розв’язання задачі класифікації контенту фінансових новин.

Практична цінність отриманих в роботі результатів може бути корисною для широкого кола користувачів, включаючи:

- Потенційному інвестору.
- Підприємства, що мають аналітичний відділ.
- Дослідників в галузі макроекономічних процесів.

Методи дослідження полягають в побудові таких моделей машинного навчання: Logistic Regression, SVM, Random Forest, Naïve Bayes, а також нейронних мереж: DNN, CNN, LSTM, BiLSTM.

Апробація результатів дисертації. Основні положення даної роботи були викладені на: XI International Scientific and Practical Conference «Integration of science as a mechanism of effective development», November 28 – December 01, 2023, Helsinki, Finland.

Дисертація складається зі вступу, п’ятьох розділів та висновків. Повний обсяг дисертації складає 119 сторінки, 23 таблиці, 22 рисунки та 3 сторінки списку використаних джерел у кількості 31 найменувань.

Ключові слова: фінансові новини, NLP, класифікація, машинне навчання, Logistic regression, SVM, Random Forest, Naïve Bayes, DNN, CNN, RNN.

ABSTRACT

Relevance of the research topic. In today's fast-paced world of finance, the ability to analyze articles and news quickly and accurately is of paramount importance. Financial news contains valuable information that can significantly affect markets, influence trading decisions, and shape investor sentiment. However, the sheer volume and complexity of financial data make manual analysis impractical. Machine learning, as a subset of artificial intelligence, allows computers to learn from data, identify patterns, and make decisions without explicit programming. In the area of financial news classification, the above algorithms can be trained to categorize news content according to predefined sentiments or topics, providing timely information and assisting in decision-making processes.

The research aims of this work are to to implement machine learning methods for classifying financial news content.

Research Tasks:

1. Analyze existing solutions for classifying financial news content.
2. Select the dataset, methods, and solutions that best meet the research objective and justify this choice.
3. Train models based on the selected dataset.
4. Perform a comparative evaluation of the models required to perform this work.

Object of research – the process of classifying text messages by emotional coloring.

Subject of research – machine learning models for solving the problem of classification of financial news content.

The practical value of the obtained results in this work can be beneficial for a wide range of users, including:

- Potential investors.
- Enterprises with an analytical department.
- Companies dealing with magnetic materials.
- Researchers in the field of macroeconomic processes.

The research methods consist of building the following machine learning models: Logistic Regression, SVM, Random Forest, Naïve Bayes, as well as neural networks: DNN, CNN, LSTM, BiLSTM.

Approval of Dissertation Results. The main points of this work were presented at the XI International Scientific and Practical Conference «Integration of science as a mechanism of effective development», November 28 – December 01, 2023, Helsinki, Finland.

The dissertation consists of an introduction, five chapters, and conclusions. The complete volume of the dissertation is 119 pages, including 23 tables, 22 figures, and a list of 31 references spanning 3 pages.

Keywords: financial news, NLP, classification, machine learning, Logistic regression, SVM, Random Forest, Naïve Bayes, DNN, CNN, RNN.

ЗМІСТ

ВСТУП

Актуальність роботи полягає в ряді факторів. У швидкозростаючій галузі фінансового аналізу швидка класифікація та інтерпретація контенту новин відіграє ключову роль у формуванні інвестиційних стратегій та ринкових прогнозів. Ця робота присвячена застосуванню машинного навчання (і в тому числі нейронних мереж) для контрольованої класифікації фінансових новин - області, де обсяг і швидкість даних вимагають автоматизованих, але високопродуктивних систем для розбору і розуміння нюансів текстової інформації [1].

Практичність цього дослідження підкреслюється динамічною природою фінансових ринків, де рішення часто є чутливими до часу і базуються на найсвіжішій інформації. Традиційні аналітичні методи випереджають величезну кількість даних, що генеруються щодня, створюючи попит на інноваційні рішення, які можуть не лише аналізувати, але й передбачати ринкові тенденції та настрої інвесторів. Шляхом критичного аналізу ця робота протиставляє існуючі методології найсучаснішим методам машинного та глибокого навчання, підкреслюючи їхню здатність розпізнавати закономірності та настрої на основі текстових даних.

Робота виконана, щоб зробити внесок у цю галузь, представивши порівняльне дослідження різних алгоритмічних підходів, починаючи від звичайних моделей машинного навчання до передових нейромережевих архітектур, таких як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), включаючи мережі з довгою та короткою пам'яттю (LSTM) та його двонаправлені варіанти. Вивчаючи сильні та слабкі сторони кожного методу, дослідження має на меті прокласти шлях до більш надійних, точних та ефективних інструментів для класифікації фінансових новин, тим самим надаючи інвесторам та аналітикам більш поглиблене розуміння постійно мінливого фінансового світу.

Область дослідження знаходиться на перетині комп'ютерної лінгвістики, науки про дані та фінансової аналітики. Воно спрямоване на зменшення розриву між швидким поширенням фінансових новин та здатністю точно і швидко інтерпретувати їхні наслідки [2]. Використовуючи підхід керованого навчання, обрано набір даних, анотованих настроями та іншими відповідними фінансовими показниками, що дозволить розробленим моделям навчатися і прогнозувати з високим ступенем точності.

Загалом, робота є дослідженням, спрямованим на вирішення реальної проблеми. Зазначено визначення нюансів ефективності машинного навчання в класифікації текстів у фінансовій сфері, яке було не тільки теоретично обґрунтованим, але й прагматично цінним. Актуальність полягає щодо створення потенційного аналітичного інструменту, який покращує процеси аналізу фінансових даних, тим самим сприяючи створенню більш проінформованої та ефективної фінансової екосистеми.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Метою даної роботи є розробка моделей, що дозволятиме проаналізувати доцільність у виконані класифікації контенту фінансових новин алгоритмів машинного навчання і показати потребу у вирішенні цього завдання.

Для досягнення мети було сформовано набір завдань, які потрібно виконати протягом роботи.

- Провести аналіз існуючих методів та рішень для класифікації фінансових новин.
- Обрати набір даних, методи та рішення, які найкраще відповідають меті дослідження.
- Побудувати моделі для вирішення завдання.
- Виконати обчислення метрик оцінювання моделей.
- Здійснити порівняльний аналіз результатів обчислення.

Розв'язання цих завдань передбачає розділення роботи на основні етапи.

1.1 Опис завдання

Для досягнення поставленої мети необхідно зробити визначення та окреслити важливість завдання, які потрібно виконати протягом дослідження. Класифікація контенту фінансових новин засобами машинного навчання - це обчислювальний процес виявлення та категоризації думок або емоцій у текстових даних для визначення ставлення автора або спікера до певної теми. Цей метод часто застосовується до фінансових новин, постів у соціальних мережах і фінансових звітів, щоб оцінити загальний настрій (позитивний, негативний або нейтральний) щодо фінансових інструментів або ринків.

Це передбачає використання алгоритмів машинного навчання та обробки природної мови (NLP) для аналізу тональності, контексту та нюансів мови, що використовується у фінансових новинах. Цей процес часто включає синтаксичний аналіз тексту, розпізнавання ключових індикаторів настроїв (наприклад, слів або фраз) і кількісну оцінку настроїв для виконання подальшої класифікації [3].

Аналіз настроїв відіграє вирішальну роль у прогнозуванні динаміки ринку, оскільки колективні настрої в новинах можуть суттєво впливати на поведінку , наприклад, інвесторів і ринкові тенденції. Позитивні новини можуть призвести до "бичачої" поведінки на ринку, тоді як негативні новини можуть викликати "ведмежу" реакцію.

Отже, застосовуючи методи машинного навчання для класифікації настроїв у контенті фінансових новин може допомогти у вирішенні практичних завдань, які мають зв’язок з підвищенням точності ринкових прогнозів. І при цьому, надає додатковий шар даних, що відображає психологічні та суб'єктивні аспекти ринкової поведінки, які часто не враховуються традиційними фінансовими метриками.

1.1.1 Використання моделей

У сфері аналізу фінансових новин традиційні статистичні моделі вже давно стали основою. Такі моделі, як лінійна регресія та аналіз часових рядів, були ключовими в минулому. Ці моделі вирізняються своєю простотою та ефективністю при роботі зі структурованими кількісними даними. Вони широко використовуються для виявлення тенденцій, прогнозування ринку та виявлення історичних закономірностей на фінансових ринках. Однак традиційні статистичні моделі мають певні обмеження. Їх залежність від заздалегідь визначених припущень і робота з великими, неструктурованими наборами даних обмежують їх застосування в сучасному середовищі, багатому на дані. В історичному контексті

вони були високоефективними, але їм не вистачало гнучкості, щоб адаптуватися до нюансів і часто якісної природи фінансових новин.

Впровадження моделей машинного навчання ознаменувало значний прогрес в аналізі фінансових новин. Ці моделі, включаючи нейронні мережі та дерева рішень, принесли зміну парадигми. На відміну від своїх традиційних аналогів, моделі машинного навчання чудово справляються з неструктурованими даними, такими як текст у фінансових новинах, і надають їм сенс. Вони можуть навчатися на даних, виявляти закономірності та робити прогнози без явного програмування. Основні переваги моделей машинного навчання полягають у їхній підвищеній точності, здатності обробляти великі й різноманітні набори даних та адаптивності до мінливих тенденцій у даних. Ці моделі відкрили нові можливості у фінансовому аналізі, уможливлюючи більш тонкий аналіз настроїв, розуміння ринку в реальному часі та прогнозну аналітику, що раніше було складно зробити за допомогою традиційних моделей.

Впровадження моделей машинного навчання ознаменувало значний прогрес в аналізі фінансових новин. Ці моделі, включаючи нейронні мережі та дерева рішень, принесли зміну парадигми. На відміну від своїх традиційних аналогів, моделі машинного навчання чудово справляються з неструктурованими даними, такими як текст у фінансових новинах, і надають їм сенс. Вони можуть навчатися на даних, виявляти закономірності та робити прогнози без явного програмування. Основні переваги моделей машинного навчання полягають у їхній підвищеній точності, здатності обробляти великі й різноманітні набори даних та адаптивності до мінливих тенденцій у даних. Ці моделі відкрили нові можливості у фінансовому аналізі, уможливлюючи більш тонкий аналіз настроїв, розуміння ринку в реальному часі та прогнозну аналітику, що раніше було складно зробити за допомогою традиційних моделей [4].

З точки зору масштабованості, моделі машинного навчання виділяються тим, що здатні обробляти величезні обсяги неструктурованих даних, які стають все більш поширеними в сучасну цифрову епоху. Складність - це ще один аспект, де

моделі машинного навчання, як правило, більш досконалі, що дозволяє їм охоплювати більш нюансовані взаємозв'язки в даних, ніж традиційні моделі. Однак ця складність може бути і недоліком, оскільки вимагає більше обчислювальних ресурсів і досвіду. Адаптивність є сильною стороною моделей машинного навчання, оскільки вони здатні еволюціонувати разом з даними, на відміну від традиційних моделей, які часто є статичними і пов'язані з початковими припущеннями.

1.1.2 Проблеми створення моделей

Створення моделей машинного навчання для аналізу фінансових новин є складним завданням, здебільшого через складну і часто непередбачувану природу фінансових ринків. На ці ринки впливає безліч факторів, включаючи економічні показники, політичні події та настрої інвесторів, які можуть змінюватися швидко і без попередження. Така непередбачуваність створює значні труднощі при створенні моделей. Моделі повинні бути достатньо складними, щоб інтерпретувати ці складнощі та надавати точні прогнози. Моделі повинні розшифровувати тонкі сигнали у фінансових новинах, які можуть вказувати на рух ринку. Необхідність у точності та глибині інтерпретації вимагає тонкого розуміння як фінансових ринків, так і алгоритмів, що лежать в основі цих моделей.

Серйозним викликом також є динамічна природа контенту новин. Вони не є статичними; вони постійно розвиваються, відображаючи мінливий глобальний економічний ландшафт. Ця еволюція стосується не лише фактів, про які повідомляється, але й способу їхньої подачі. Мова, тон і настрій фінансових новин можуть значно змінюватися з часом під впливом поточних подій, культурних зрушень і медіа-ландшафту. Моделі машинного навчання повинні бути достатньо адаптивними, щоб ефективно реагувати на ці зміни. Їх потрібно навчати на різноманітних наборах даних, які охоплюють широкий спектр стилів новин і

настроїв. Адаптація моделей до цих коливань у подачі новин має вирішальне значення для збереження їхньої точності та актуальності. Це вимагає постійного моніторингу та оновлення моделей, щоб забезпечити їхню відповідність останнім тенденціям і змінам у змісті та подачі новин.

При створенні моделей машинного навчання для виконання завдання є якість даних. Ефективність цих моделей значною мірою залежить від точності, неупередженості та загальної якості даних, які до них подаються [5]. Неточні або упереджені дані можуть призвести до помилкових прогнозів та аналізів, що суттєво впливає на процеси прийняття рішень у фінансовому контексті. Тому забезпечення цілісності та надійності даних є критично важливим завданням. Моделі, побудовані на низькоякісних даних, не лише не дають цінної інформації, але й можуть вводити в оману, посилюючи існуючі упередження або помилки. Ця проблема вимагає суворої перевірки та очищення даних для підтримки високих стандартів якості даних.

Стосовно наявності та доступності повних і різноманітних наборів даних. Ефективні моделі аналізу фінансових новин потребують великих наборів даних, які представляють широкий спектр фінансових новин, що охоплюють різні ринки, сектори та геополітичні регіони. Однак доступ до таких різноманітних і всеосяжних наборів даних може бути непростим. Часто доводиться враховувати юридичні та етичні міркування, особливо коли йдеться про власницьку, конфіденційну або особисту інформацію. Правові рамки, що регулюють використання даних, різняться залежно від регіону та типу даних, що додає складнощів у процес збору даних. Забезпечення дотримання правових та етичних норм при отриманні достатньої кількості даних для побудови надійних моделей є делікатним балансуванням.

Надмірна адаптація становить значний ризик при створенні моделей машинного навчання, особливо в контексті аналізу фінансових новин. Моделі, які надмірно пристосовані до певних наборів даних, можуть працювати дуже добре на цих даних, але погано на нових, небачених даних. Ця проблема виникає, коли

моделі є надмірно складними, вловлюючи шум у даних так, ніби він є значущою закономірністю. Ключовий виклик полягає в тому, щоб збалансувати складність моделі зі здатністю узагальнювати різні ринкові умови і типи фінансових новин. Моделі повинні бути достатньо складними, щоб реагувати на нюанси фінансових новин, і в той же час достатньо надійними, щоб ефективно застосовувати отримані знання до нових даних. Для вирішення проблеми надмірної пристосованості потрібні такі методи, як регуляризація, перехресна перевірка та використання ансамблевих методів, що гарантують точність моделей та їхню адаптивність до різних ринкових сценаріїв.

1.1.3 Машинне навчання

Машинне навчання стало трансформаційною силою у сфері аналізу фінансових новин, докорінно змінивши ландшафт обробки та інтерпретації текстових даних. Його роль виходить далеко за межі простої обробки даних, пропонуючи тонке розуміння ринкових тенденцій, настроїв інвесторів і потенційних фінансових результатів на основі новинного контенту.

З появою складних алгоритмів машинного навчання фінансові аналітики та інвестори тепер можуть глибше розуміти динаміку ринку. Моделі машинного навчання вміють виявляти закономірності та кореляції у фінансових новинах, які можуть бути непомітними для людського ока. Ця здатність не лише підвищує точність ринкових прогнозів, але й відкриває нові можливості для фінансових стратегій, заснованих на даних.

Однією з ключових переваг машинного навчання в аналізі фінансових новин є його здатність швидко та ефективно обробляти величезні обсяги даних. У сучасну цифрову епоху, коли фінансові новини генеруються у величезних обсягах, моделі машинного навчання виділяються як важливі інструменти для фільтрації шуму та вилучення значущої інформації [6].

Більше того, алгоритми машинного навчання, особливо ті, що базуються на обробці природної мови, спричинили революцію в аналізі настроїв. Вони можуть аналізувати тон і контекст фінансових новин, надаючи уявлення про те, як певні новини можуть вплинути на поведінку інвесторів і ринкові тенденції. Цей аспект машинного навчання є безцінним у формулюванні інвестиційних стратегій та прийнятті обґрунтованих фінансових рішень.

Переваги машинного навчання в аналізі фінансових новин:

1. Покращена обробка даних: можливість ефективно обробляти та аналізувати великі обсяги новинних даних.
2. Підвищена точність: краще виявлення закономірностей і тенденцій, що призводить до більш точних ринкових прогнозів.
3. Покращений аналіз настроїв: вміння інтерпретувати тон і настрої фінансових новин, пропонуючи більш глибоке розуміння ринку.
4. Аналіз в режимі реального часу: здатність обробляти та аналізувати новини в режимі реального часу, сприяючи своєчасному прийняттю фінансових рішень.
5. Адаптивне навчання: постійно розвивається і адаптується до нових даних, покращуючи свої прогностичні можливості з часом.

Обмеження машинного навчання в аналізі фінансових новин:

1. Залежність від якості даних: значною мірою залежить від якості та цілісності вхідних даних.
2. Складність інтерпретації: складна природа деяких алгоритмів може ускладнити інтерпретацію їхніх результатів.
3. Ризик надмірного пристосування: потенційна можливість надмірного пристосування до навчальних даних, що призводить до низької продуктивності на нових, непередбачуваних даних.
4. Обчислювальні ресурси: вимагає значних обчислювальних ресурсів, особливо для більш складних моделей.

5. Занепокоєння щодо конфіденційності даних: Обробка та опрацювання великих наборів даних може порушувати питання конфіденційності даних та етики.

1.2 Аналіз існуючих рішень

Ринок аналізу настроїв наповнений різноманітними рішеннями, кожне з яких пропонує унікальні функції та можливості. Серед них помітними гравцями є SproutSocial, Brand24 та Qualtrics Clarabridge. Ці платформи задовольняють зростаючий попит на складні інструменти аналізу настроїв, особливо в таких сферах, як дослідження ринку, моніторинг брендів та аналіз відгуків клієнтів. Загальний огляд цих рішень допомагає зрозуміти їхнє місце на ринку та внесок у розвиток галузі.

Платформа для управління соціальними мережами для аналізу настроїв SproutSocial (рисунок 1.1) навчається на широкому спектрі текстових зразків, фіксуючи ідіоми рідної мови, галузеві жаргони та вирази. Воно використовує методи обробки природної мови (NLP), такі як тегування частин мови, лематизація, попередня полярність, заперечення та семантична кластеризація для вилучення нюансованих позитивних, нейтральних та негативних настроїв з різних джерел, використовуючи генеративного штучного інтелекту (ШІ) для вилучення ключових слів з дописів у соціальних мережах.

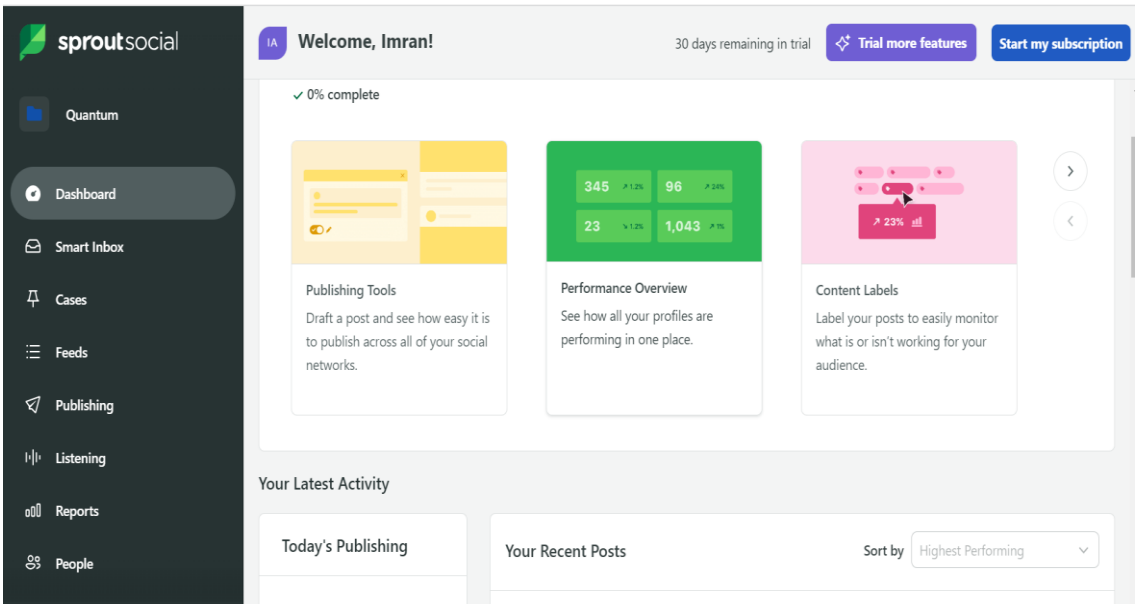


Рисунок 1.1 – Демонстрація використання SproutSocial

Brand24 - це інструмент моніторингу соціальних мереж (рисунок 1.2), який надає доступ до згадок у різних онлайн-джерелах, включаючи соціальні мережі, новини, блоги, відео, форуми, подкасти та огляди, за допомогою штучного інтелекту. Він пропонує розширений аналіз настроїв для сегментування позитивних, негативних або нейтральних згадок, вимірювання та звітування про PR-висвітлення, відстеження пізнаванності та присутності бренду, а також оцінки маркетингових зусиль. Функція аналізу емоцій Brand24 використовує передові алгоритми обробки природної мови (NLP) та машинного навчання для сканування соціальних мереж, блогів, новинних статей та інших онлайн-джерел на наявність згадок, пов'язаних з брендом, продуктом або галуззю, і визначає конкретні емоції, виражені в контенті. Цей інструмент покликаний допомогти бізнесу збирати інформацію, відфільтровувати шум і зосереджуватися на головному, а також приймати обґрунтовані рішення на основі відгуків клієнтів про компанію.

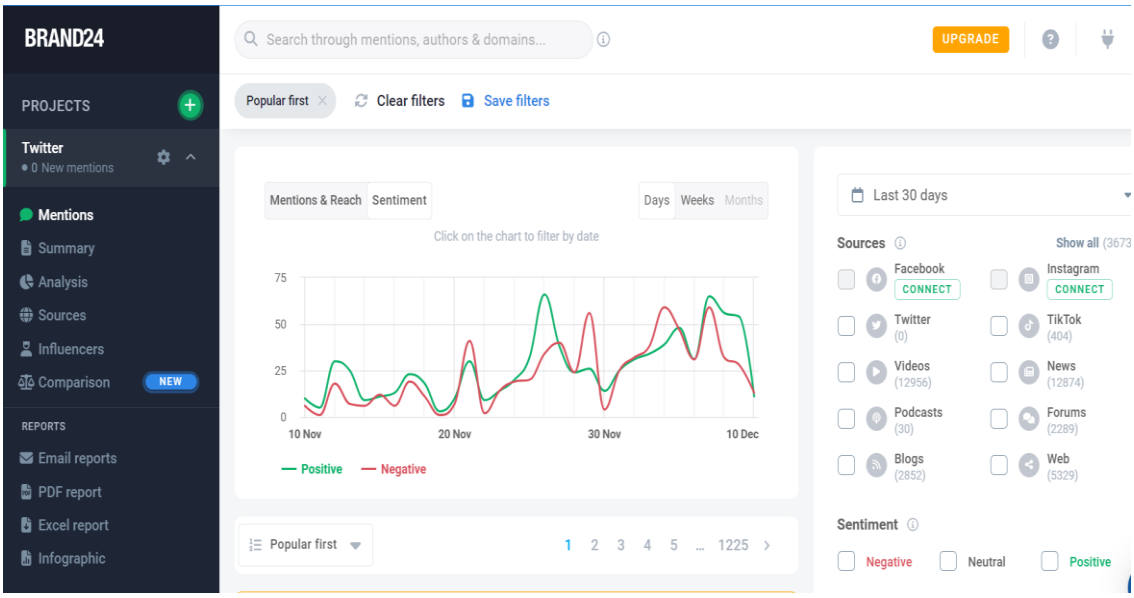


Рисунок 1.2 – Демонстрація використання Brand24

Qualtrics Clarabridge - це рішення для обробки природної мови (NLP), яке використовує методи машинного навчання, засновані на правилах, для вилучення, тегування та оцінки концепцій, які мають відношення до аналізу клієнтського досвіду, таких як емоції, зусилля, наміри та ненормативна лексика (рисунок 1.3). Він автоматично виявляє теми та настрої у відкритому тексті та визначає емоції, наміри та зусилля за допомогою 150+ галузевих моделей NLU. Технологія використовує шестиетапний процес, схожий на робочий процес, щоб ідентифікувати та розуміти фрази, граматику та зв'язки між словами, подібно до того, як люди надають значення тому, що вони читають. У поєднанні з методами аналізу настроїв, обробка природної мови від Qualtrics створює найточніше і найдосконаліше рішення для аналізу тексту з усіх доступних.

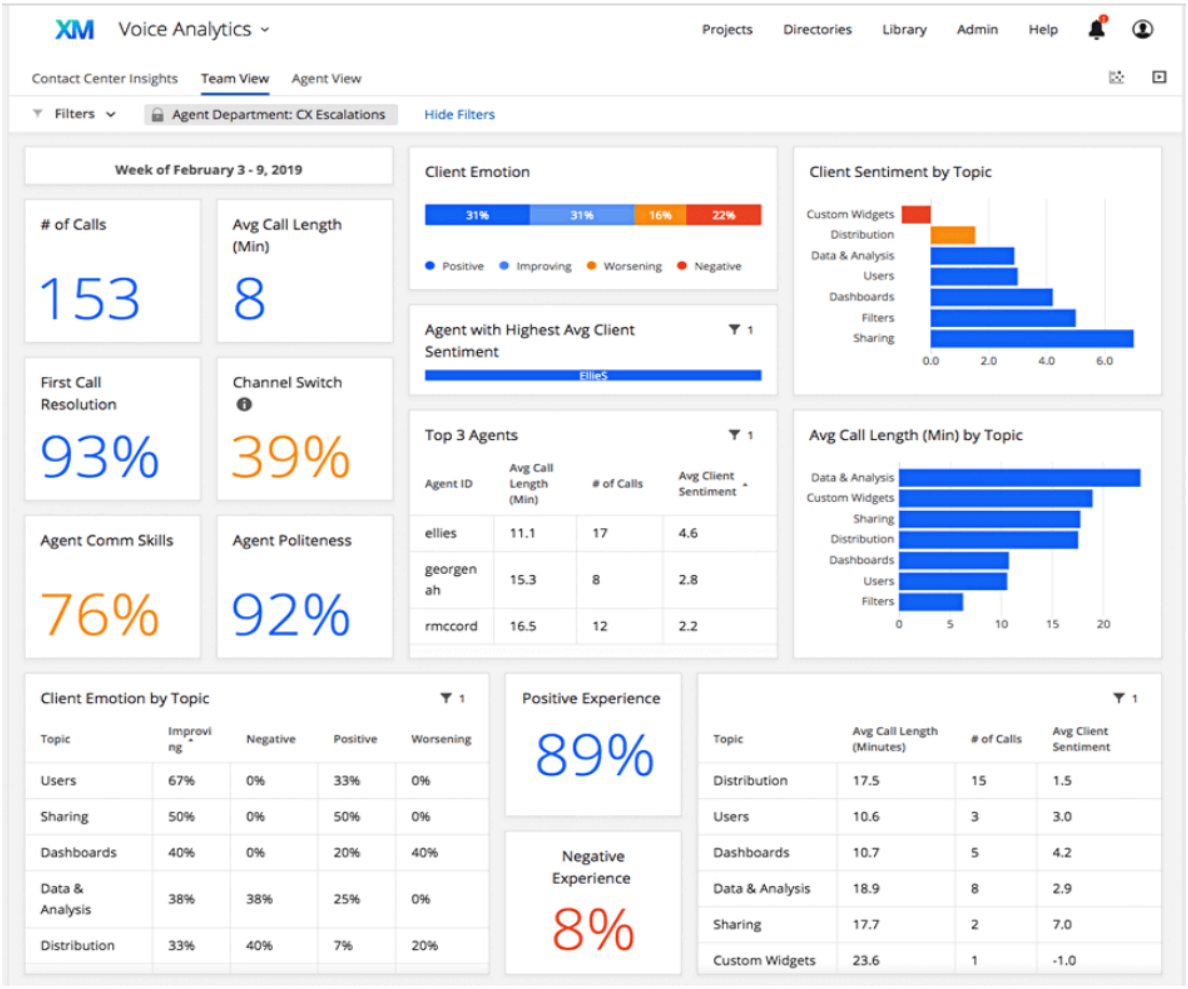


Рисунок 1.3 – Демонстрація використання Qualtrics Clarabridge

Було розглянуто основні характеристики трьох провідних сервісів - SproutSocial, Brand24 та Qualtrics Clarabridge. Кожен сервіс має свої унікальні переваги: від обробки мови та вилучення ключових слів за допомогою штучного інтелекту у SproutSocial до широкого онлайн-моніторингу та аналізу емоцій у Brand24. Тим часом Qualtrics Clarabridge пропонує спеціалізований фокус на клієнтському досвіді з його складними методами машинного навчання на основі правил та аналізу тексту. У наступній таблиці 1.1 представлені ключові особливості цих сервісів, що ілюструють їхню роль у формуванні погляду на сферу аналізу настроїв.

Таблиця 1.1 – Основні характеристики існуючих рішень

Назва	Призначення	Засоби
SproutSocial	Платформа для управління соціальними мережами, яка допомагає компаніям керувати та аналізувати свою присутність у соціальних мережах.	Recurrent Neural Network (LSTM)
Brand24	Інструмент моніторингу соціальних мереж	Convolutional Neural Network
Qualtrics Clarabridge	Рішення для вилучення, тегування та оцінювання концепцій, які мають відношення до аналізу клієнтського досвіду	Recurrent Neural Network (LSTM)

Згідно таблиці 1.1 можна стверджувати, що існуючі сервіси для класифікації настроїв контенту в основному зосереджено на тексти із соціальних мереж та відгуках клієнтів. Але подібної послуги щодо фінансових новин спостерігаються рідко.

Висновок до розділу 1

У цьому розділі було наведено основну мету, яка полягала в тому, щоб представити ретельне дослідження практичності створення рішень на основі машинного навчання для класифікації контенту фінансових новин. Описано різні аспекти, починаючи з аналізу існуючих на ринку рішень і закінчуючи вивченням інструментів і алгоритмів, що використовуються сервісами аналізу настроїв.

Було проведено оцінку ролі машинного навчання в поліпшенні аналізу фінансових новин. Це містить огляд того, як зазначеними засобами революціонізувало інтерпретація новин, підкреслюючи його вплив на аналіз настроїв на фінансових ринках. Надано уявлення про складність і динамічність фінансових ринків та зміст фінансових новин, підкресливши потребу в моделях машинного навчання в цій галузі.

Проведено опис викликів та проблем, пов'язаних зі створенням цих моделей. У цій частині розділу детально розглянув такі перешкоди, як проблеми з якістю даних, труднощі з наявністю та доступністю даних, а також ризики надмірної пристосованості та узагальнення. Вивчення цих проблем підкреслило складнощі, пов'язані з побудовою ефективних і надійних моделей машинного навчання для аналізу фінансових новин.

Виконав аналіз існуючих на ринку рішень для аналізу настроїв, зосередившись на таких платформах, як SproutSocial, Brand24 і Qualtrics Clarabridge. Це містить інструменти та алгоритми обробки природної мови (NLP), що використовуються цими платформами, пропонуючи порівняльну перспективу, яка висвітлює їхні різні підходи та можливості.

2 ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

У цьому розділі було представлено опис існуючих інструментів, алгоритмів машинного навчання та нейронних мережа. Ретельний огляд та аналіз засобів побудови моделей, робить процес розробки надійним та ефективним.

2.1 Алгоритми машинного навчання

У комплексному дослідженні моделей машинного навчання для класифікації контенту фінансових новин опрацьовано різноманітні набори алгоритмів, кожен з яких робить свій унікальний внесок у складне завдання аналізу та інтерпретації відповідних даних. Зосередження на розумінні того, як ці моделі можуть бути ефективно використані для класифікації цільового тексту, що є ключовим аспектом у фінансовому секторі [7].

Набір моделей включає логістичну регресію, випадковий ліс, машину опорних векторів (SVM), мультиноміальний наївний Байєс. Кожна з цих зазначених моделей має свої переваги і спеціалізується на обробці різних аспектів даних щодо фінансових новин.

Логістична регресія та мультиноміальний наївний Байєс особливо корисні в ситуаціях, коли завдання класифікації є відносно простим або коли пріоритетом є інтерпретованість моделі. Вони чудово підходять для класифікації контенту новин на бінарні або категоріальні результати, такі як аналіз настроїв - визначення того, чи має новина позитивний, негативний або нейтральний вплив на ринок.

З іншого боку, машини випадкового лісу та опорних векторів забезпечують більшу складність і надійність. Вони добре справляються з хитросплетіннями фінансових даних, що робить їх придатними для більш складних завдань

класифікації, таких як прогнозування ринкових тенденцій або виявлення тонких закономірностей у фінансових новинах, які можуть вплинути , наприклад, на ціни цінних паперів.

Готуючись до більш детального розгляду кожного з цих алгоритмів, було показано, як їх можна використовувати для перетворення величезної кількості різноманітних фінансових новин на дієві ідеї, відіграючи таким чином ключову роль у прийнятті фінансових рішень і розробці стратегій.

2.1.1 Логістична регресія

Логістична регресія (Logistic regression) - це фундаментальний статистичний метод, який використовується переважно для задач бінарної класифікації [8]. По суті, логістична регресія призначена для моделювання ймовірності існування певного класу або події, наприклад, чи є стаття про фінансові новини позитивною або негативною. На відміну від лінійної регресії, яка прогнозує безперервні результати, логістична регресія використовується, коли результат є бінарним, тобто коли результат потрапляє в одну з двох категорій.

Основний принцип логістичної регресії полягає в її здатності оцінювати ймовірності за допомогою логістичної функції, яка є S-подібною кривою (рисунок 2.1). Ця крива дозволяє моделі впоратися з ситуаціями, коли залежна змінна є категоричною [9]. Наприклад, в контексті аналізу фінансових новин логістична регресія може моделювати ймовірність того, що чи новина класифікована як "позитивна" або "негативна" новина.

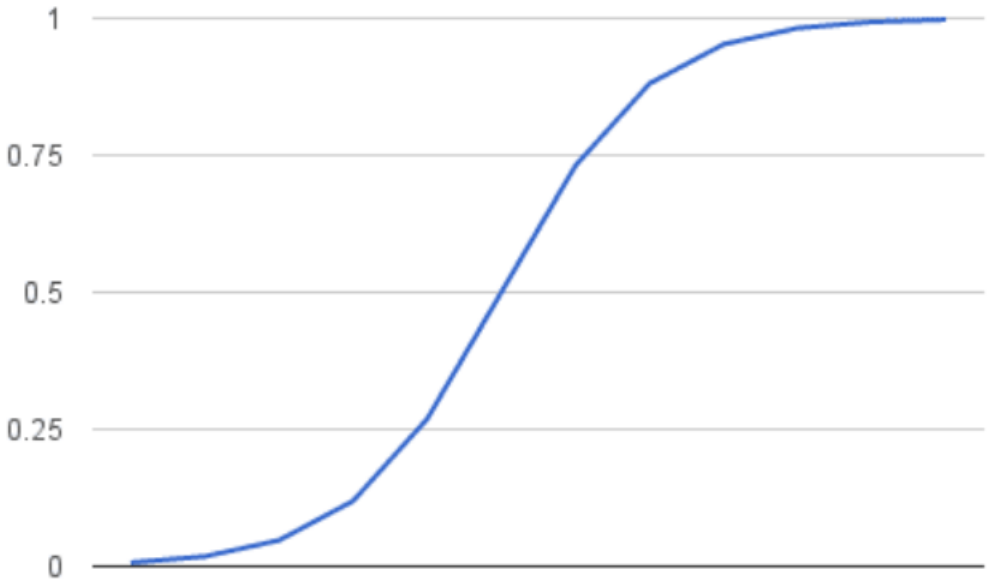


Рисунок 2.1 – Графічне представлення логістичної функції [9]

Логістична функція в логістичній регресії перетворює будь-яку лінійну комбінацію вхідних ознак у значення між 0 і 1, яке можна інтерпретувати як ймовірність. Ця ймовірність потім використовується для прийняття бінарного рішення: якщо ймовірність більша або дорівнює 0,5, результат класифікується в одній категорії, а якщо менша 0,5 - в іншій. Такий підхід робить логістичну регресію особливо придатною для сценаріїв, де результат є дихотомічним, наприклад, в аналізі настроїв щодо фінансових новин.

Математична основа логістичної регресії ґрунтується на вищезазначеній логістичній функції [10], яка є центральною для її здатності моделювати бінарні результати. Вона характеризується S-подібною кривою, відомою як сигмоїдальна функція, яка формально представлена як:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2.1}$$

де e – основа натурального логарифма, і z - лінійна комбінація ознак (вхідних змінних), зазвичай виражається як:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (2.2)$$

де β – представляє коефіцієнти.

Сигмоїдна функція відображає будь-яке дійсне число у значення між 0 і 1, що робить її ідеальною для оцінки ймовірності. Крива починається в точці (0,0), піднімається до точки 0,5 на осі абсцис, а потім продовжується до точки (1,1). Ця S-подібна крива має вирішальне значення для перетворення лінійних залежностей у ймовірності.

У контексті логістичної регресії поняття шансів і лог-шансів (лог-функція) є ключовими. Шанс події O - це відношення ймовірності того, що подія відбудеться, до ймовірності того, що вона не відбудеться. Математично, шанс задається так:

$$O = \frac{P}{1-P} \quad (2.3)$$

де P – ймовірність події.

Логістична регресія безпосередньо моделює лог-шанси або логіт-функцію, яка є натуральним логарифмом формули шансів (2.3). Логіт-функція виражається так:

$$log(O) = log\left(\frac{P}{1-P}\right). \quad (2.4)$$

У логістичній регресії логіт-функція (2.4) дорівнює лінійній комбінації ознак (2.2), що призводить до рівняння моделі:

$$log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (2.5)$$

Це рівняння (2.5) лежить в основі моделі логістичної регресії, що дозволяє оцінити ймовірність бінарного результату шляхом застосування логістичної функції до лінійної комбінації вхідних ознак.

2.1.2 Машина опорних векторів

Машина опорних векторів (SVM) - це потужний і універсальний алгоритм машинного навчання, який використовується переважно для класифікації та регресії. Як інструмент класифікації, SVM відомий своєю здатністю знаходити оптимальний спосіб відокремлення різних класів у наборі даних, що робить його особливо корисним для складних наборів даних, де різниця між різними класами не відразу очевидна [11].

Основна концепція SVM (рисунок 2.2) полягає у знаходженні гіперплощини в N-вимірному просторі (N - кількість ознак), яка чітко класифікує точки даних на різні класи. Щоб зрозуміти це, уявіть, що кожна точка даних у наборі даних представлена у просторі, де кожна ознака є виміром. Мета SVM - знайти плоску межу (або гіперплощину), яка розділяє ці точки на класи .

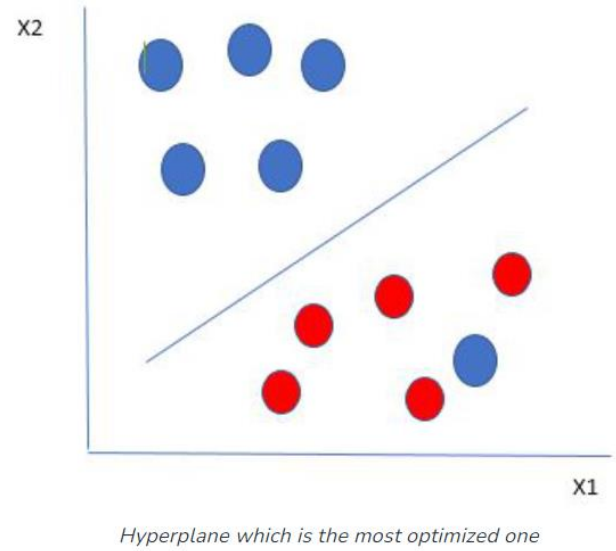


Рисунок 2.2 – Демонстрація концепції SVM [12]

Основне завдання цього алгоритму при класифікації - ідентифікувати цю гіперплощину таким чином, щоб вона мала найбільшу відстань до найближчої

точки навчальних даних будь-якого класу (також відому як маржа), гарантуючи, що поділ між різними класами є максимально чітким і виразним. Це досягається шляхом максимізації поля гіперплощини, тим самим покращуючи його класифікаційні можливості.

По суті, SVM працює шляхом визначення опорних векторів, які є точками даних, найближчими до гіперплощини. Ці точки є критично важливими для визначення положення та орієнтації гіперплощини. Ефективно використовуючи ці опорні вектори, SVM забезпечує надійне та оптимальне розділення класів, навіть у випадках, коли дані не піддаються лінійному розділенню. У ситуаціях, коли набір даних є більш складним, SVM використовує техніку, відому як трюк ядра, для перетворення даних у простір вищої розмірності, де можна успішно ідентифікувати гіперплощину для розділення класів.

Алгоритм працює за принципом пошуку оптимальної гіперплощини, яка ефективно класифікує точки даних за окремими категоріями. Цей процес ґрунтується на двох основних поняттях: поля та опорні вектори [12].

Гіперплощина - це межа прийняття рішень, яка розділяє різні класи в наборі даних. Вона може бути представлена як лінія у двовимірному просторі або як площина у вищих вимірах. Загальне рівняння для цієї гіперплощини має вигляд:

$$\vec{w} * \vec{x} - b = 0 \tag{2.6}$$

де $\vec{w}$ – вектор ваги, перпендикулярний до гіперплощини, $\vec{x}$ - вектор ознак, і b - член зсуву. І завдання полягає в тому, що розташувати цю гіперплощину так, щоб вона максимально розділяла різні класи.

Поле (Margin) в SVM визначається як відстань між гіперплощиною і найближчими точками даних з будь-якого класу. Ці найближчі точки називаються опорними векторами. Вони є ключовими, оскільки безпосередньо впливають на положення та орієнтацію гіперплощини.

Математично, поле r з рівняння (1) має вигляд:

$$r = \frac{2}{\|\vec{w}\|} \tag{2.7}$$

де $\|\vec{w}\|$ – представляє норму, або довжину, вагового вектору $\vec{w}$. Більший запас означає, що модель має потенційно кращі узагальнюючі можливості.

Щоб знайти оптимальну гіперплощину, в SVM розв'язується задача опуклої оптимізації. Мета полягає в тому, щоб максимізувати поле (тобто збільшити відстань між гіперплощиною та опорними векторами), мінімізуючи при цьому помилки класифікації.

Розв'язуючи цю оптимізаційну задачу, SVM визначає найкращі значення вагового вектора $\vec{w}$ та члена зсуву b, які визначають гіперплощину. На рішення значною мірою впливають опорні вектори, які є точками даних, найближчими до гіперплощини. Ці точки є критично важливими, оскільки вони «підтримують» оптимальне положення та орієнтацію гіперплощини, тим самим дозволяючи SVM ефективно класифікувати точки даних.

2.1.3 Випадковий ліс

Випадковий ліс (Random Forest) - це складний алгоритм машинного навчання, який належить до категорії методів ансамблевого навчання. Ансамблеве навчання відноситься до методів, які об'єднують прогнози декількох алгоритмів машинного навчання для отримання більш точних і надійних результатів, ніж будь-яка окрема модель могла б досягти самостійно. У випадку випадкового лісу цей підхід передбачає інтеграцію численних дерев рішень (рисунок 2.3) для формування «лісу» моделей, кожна з яких робить свій внесок у кінцевий прогноз.

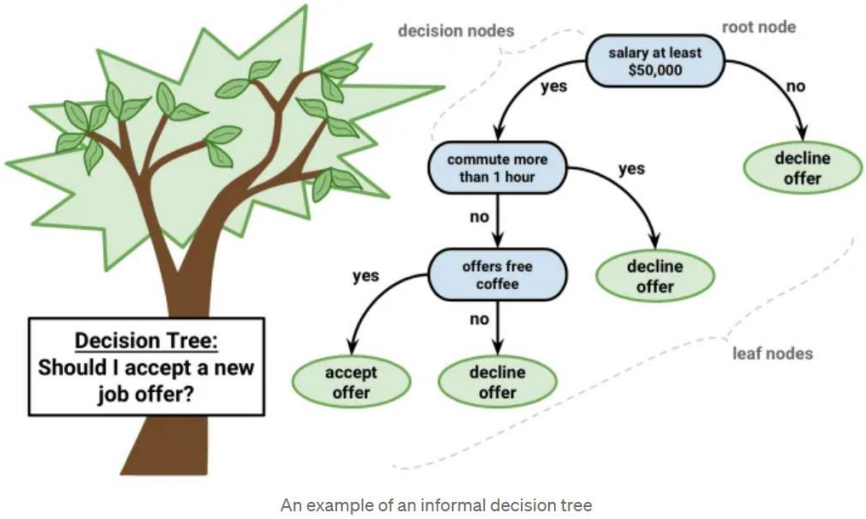


Рисунок 2.3 – Демонстрація прикладу дерева рішень [14]

Цей алгоритм, по суті, будує кілька дерев рішень під час процесу навчання і виводить у вигляді класів (класифікація) або середнє передбачення (регресія) окремих дерев. Кожне дерево у випадковому лісі дає прогноз, і найпоширеніший результат (у випадку класифікації) або середнє значення (у випадку регресії) стає прогнозом випадкового лісу [13].

Основний принцип Random Forest полягає у створенні цих дерев рішень, які генеруються з використанням випадкової підмножини навчальних даних. Ця випадковість гарантує, що окремі дерева є різноманітними, що, в свою чергу, підвищує загальну прогностичну здатність і точність моделі. Алгоритм також випадковим чином вибирає підмножини ознак при розбитті вузлів, додаючи ще один рівень різноманітності до моделі.

Метод Random Forest особливо цінується за його здатність обробляти великі набори даних високої розмірності та за його стійкість до перенавчання завдяки механізму усереднення або «голосування». Однак варто зазначити, що хоча Random Forest може бути високоефективним, хитросплетіння окремих дерев та їхня велика кількість можуть зробити модель складнішою та менш інтерпретованою, ніж простіші одномодельні підходи [14].

Ансамблеве навчання – це фундаментальна концепція машинного навчання, яка передбачає об’єднання прогнозів від декількох моделей для підвищення загальної точності та надійності прогнозного аналізу. Основна передумова ансамблевого навчання полягає в тому, що група слабких учнів може, об’єднавшись, сформувати сильного учня. Цей підхід особливо ефективний для зменшення помилок, оскільки він використовує сильні сторони і компенсує слабкі сторони окремих моделей.

У контексті машинного навчання ансамблеві методи є важливими, оскільки вони дають змогу покращити продуктивність моделі, вирішуючи такі проблеми, як перенавчання, дисперсія та упередженість. Ансамблеві методи навчання особливо корисні при роботі зі складними наборами даних, де окремим моделям може бути складно точно відобразити основні закономірності.

Випадковий ліс є яскравим прикладом ансамблевого навчання. Він належить до категорії ансамблевих методів, відомих як bagging, або бутстреп-агрегування. Random Forest створює «ліс» дерев рішень, кожне з яких навчається на випадковій підмножині навчальних даних. Така рандомізація забезпечує різноманітність дерев, зменшуючи кореляцію між ними і, як наслідок, зменшуючи дисперсію загальної моделі [15].

Випадковий ліс вписується в ширший контекст ансамблевих методів, втілюючи принцип сили в числах. Кожне дерево у випадковому лісі приймає рішення, і ці рішення потім об’єднуються для формування остаточного прогнозу. Це об’єднання може відбуватися у формі голосування (в задачах класифікації) або усереднення (в задачах регресії). Таким чином, сукупний результат часто є більш точним і надійним, ніж той, що може дати кожне окреме дерево.

По суті, Random Forest використовує філософію ансамблевого навчання, будуючи кілька дерев рішень і об’єднуючи їхні результати для підвищення точності прогнозування та зменшення ймовірності надмірного пристосування. Це робить його потужним інструментом в інструментарії машинного навчання,

особливо для складних завдань, таких як аналіз фінансових новин, де дані можуть бути багатовимірними і різноманітними.

2.1.4 Наївний байєсів класифікатор

Мультиноміальний наївний Байєс (Multinomial Naive Bayes) – це імовірнісний метод навчання в машинному навчанні, який переважно використовується для задач класифікації. Цей алгоритм є окремим випадком сімейства наївного Байєса, який застосовує теорему Байєса з «наївним» припущенням про умовну незалежність між кожною парою ознак, враховуючи значення змінної класу. Мультиноміальний наївний Байєс особливо добре підходить для задач класифікації тексту, де ознаками зазвичай є частоти, з якими певні події згенеровані мультиноміальним розподілом (наприклад, частоти слів у документі) [16].

Основна концепція мультиноміального наївного Байєса полягає в обчисленні ймовірності кожного класу на основі вхідних ознак, а потім у виборі класу з найвищою ймовірністю в якості результату. Цей підхід робить його ефективним для обробки великих наборів даних з декількома класами і зазвичай використовується в задачах обробки природної мови.

Механізм роботи мультиноміального наївного Байєса, особливо в контексті текстових даних, ґрунтується на його здатності використовувати підрахунки частот і ймовірнісні моделі для класифікації документів.

Застосування до текстових даних:

- Мультиноміальний наївний Байєс розглядає текстові дані як набір подій, де кожна подія (або ознака) зазвичай є словом або терміном з документа. У контексті класифікації тексту такими подіями є частота слів у документі.

- Для кожного документа алгоритм підраховує частоту кожного слова. Ці підрахунки формують вектори ознак, які представляють документи в числовому форматі, який може обробляти алгоритм.

Multinomial Naive Bayes застосовує теорему Байєса з припущенням, що ознаки (частоти слів) розподілені мультиноміально. Це припущення спрощує обчислення умовної ймовірності належності документа до певного класу [17].

Теорема в цьому контексті використовується для обчислення апостеріорної ймовірності класу за вектором ознак (кількості слів). Формула виглядає наступним чином:

$$P(C|F_1, F_2, \dots, F_n) = \frac{P(C) * P(F_1, F_2, \dots, F_n|C)}{P(F_1, F_2, \dots, F_n)} \quad (2.8)$$

де C – є класом (наприклад, "позитивні" або "негативні" настрої), і $F_1, F_2, \dots, F_n$ - це ознаки (частоти слів).

Обробка підрахунків частот слів:

- Алгоритм обчислює ймовірність належності кожного слова до певного класу, тобто, $P(F_i)$. Це робиться шляхом ділення кількості разів, коли слово F_i з'являється в документах класу C, на загальну кількість слів у всіх документах класу C.
- Для кожного документа алгоритм підраховує частоту кожного слова. Ці підрахунки формують вектори ознак, які представляють документи в числовому форматі, який може обробляти алгоритм.

Щоб впоратися з проблемою нульової ймовірності (коли слово не з'являється в навчальних даних для певного класу), мультиноміальний наївний Байєс зазвичай використовує техніку, яка називається згладжуванням Лапласа. Це передбачає додавання невеликої константи (зазвичай 1) до підрахунку для кожного слова, гарантуючи, що жодна ймовірність ніколи не дорівнюватиме нулю.

Підсумовуючи, можна сказати, що суть механізму роботи Multinomial Naive Bayes у класифікації текстових даних полягає у використанні частот слів як ознак, застосуванні теореми Байєса в припущенні про мультиноміальний розподіл і таких

методів, як згладжування Лапласа, для вирішення потенційних проблем з підрахунком частот. Це робить його високоефективним для таких завдань, як класифікація документів, аналіз настроїв і виявлення спаму в обробці природної мови.

2.2 Алгоритми глибокого навчання

У сфері машинного навчання, що стрімко розвивається, алгоритми глибокого навчання стали потужними інструментами, особливо у вирішенні складних завдань, в тому числі класифікація контенту фінансових новин. Вони відомі своєю здатністю вивчати високорівневі абстракції з даних, значно розширили можливості в різних сферах, включаючи фінанси. В цьому підрозділі ознайомимся в тонкощі алгоритмів глибокого навчання, зосереджуючись на їхній структурі, функціональності та конкретному застосуванні в аналізі фінансових новин.

Глибоке навчання, підвид машинного навчання, включає нейронні мережі з декількома шарами, які дозволяють моделі вивчати представлення даних на різних рівнях абстракції. Ці шари перетворюють вхідні дані за допомогою серії нелінійних етапів обробки, що дозволяє алгоритму розуміти великі і складні набори даних, такі як статті фінансових новин, багаті на контекст і тонкощі.

Було розглянуто три алгоритми глибокого навчання - глибокі нейронні мережі (DNN), згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN) - кожен з яких має унікальні переваги в обробці та аналізі фінансових даних. DNN, відомі своєю глибиною і складністю, вміють виявляти складні закономірності у великих масивах даних. CNN, що використовуються в основному для розпізнавання зображень, знайшли нове застосування в обробці послідовних даних, таких як текст. RNN, завдяки своїй здатності обробляти послідовності даних, особливо добре підходять для завдань, які передбачають розуміння контексту і часової динаміки фінансових новин.

Розглядаючи кожен алгоритм, було висвітлено їхні структурні компоненти, принципи роботи та конкретну роль у покращенні аналізу фінансових новин, пропонуючи розуміння того, як ці технології трансформують область інтерпретації фінансових даних та прийняття рішень.

2.2.1 Глибокі нейронні мережі

Глибокі нейронні мережі (DNN) є значним досягненням у галузі машинного навчання, що характеризується їхньою архітектурою та функціональністю, які відрізняють їх від традиційних нейронних мереж.

DNN складається з вхідного шару, декількох прихованих шарів і вихідного шару (рисунок 2.4). Кожен шар містить ряд взаємопов'язаних вузлів, або нейронів, які обробляють вхідні дані [18].

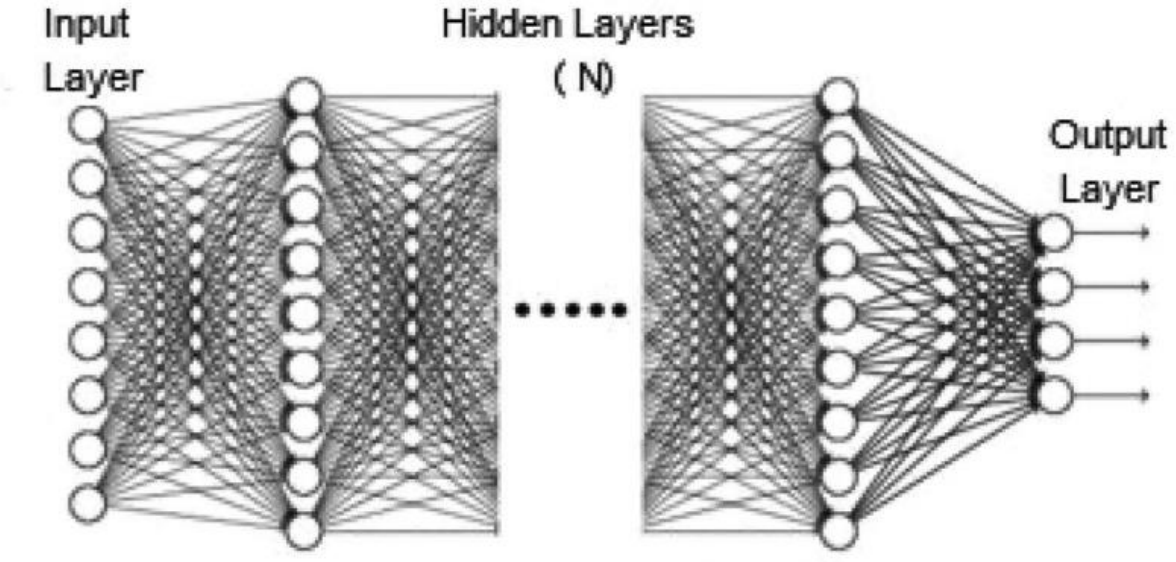


Рисунок 2.4 – Загальна архітектура глибокої нейронної мережі з декількома прихованими шарами [18]

Архітектура та функціональність:

- Шари в DNN виконують низку перетворень вхідних даних. Коли дані проходять через ці шари, мережа вчиться ідентифікувати та витягувати ознаки на різних рівнях складності. Нижчі шари можуть вивчати базові ознаки, тоді як глибші шари можуть розпізнавати більш абстрактні патерни.
- Нейронні мережі використовують функції активації, такі як сигмоїд, гіперболічний тангенс або ReLU (випрямлена лінійна одиниця), для кожного нейрона. Ці функції надають мережі нелінійних властивостей, що дозволяє їй вивчати складні закономірності в даних.

Глибина і складність:

- Ключовою відмінністю DNN від традиційних нейронних мереж є їхня "глибина". Під "глибиною" у DNN розуміють наявність декількох прихованих шарів, яких може бути від декількох до сотень, залежно від складності задачі.
- Така глибина дозволяє DNN моделювати складні та високорівневі абстракції, що є неможливим для дрібних мереж (традиційних нейронних мереж з невеликою кількістю прихованих шарів). Як наслідок, DNN можуть впоратися з більш складними завданнями і наборами даних, що робить їх особливо ефективними для таких додатків, як розпізнавання зображень і мовлення, обробка природної мови і, особливо, аналіз фінансових новин.
- Глибина і складність DNN також означає, що для ефективного навчання їм потрібен більший обсяг даних порівняно з традиційними нейронними мережами. Це пов'язано з тим, що додаткові шари дають мережі більшу здатність до навчання, але також створюють більший ризик надмірної адаптації, якщо їх не навчати на достатньо великому і різноманітному наборі даних.

Процес навчання глибоких нейронних мереж (DNN) є критично важливим аспектом, який дозволяє цим моделям навчатися на основі даних [19]. Цей процес в першу чергу включає алгоритм зворотного поширення та техніку оптимізації градієнтного спуску.

Процес навчання в DNN:

- Навчання DNN передбачає коригування ваг мережі для мінімізації різниці між прогнозованим і фактичним результатом (відомої як помилка). Цей процес відбувається за допомогою набору даних, де правильний результат відомий. Шари виконують низку перетворень вхідних даних.
- Навчання починається з випадкової ініціалізації ваг. Потім мережа робить прогнози на навчальних даних і обчислює похибку для кожного прогнозу.

Алгоритм зворотного розповсюдження:

- Зворотне розповсюдження - це метод, який використовується в нейронних мережах для розповсюдження помилки назад через мережу, що дозволяє їй відповідним чином налаштувати ваги.
- Під час зворотного розповсюдження помилка поширюється у зворотному напрямку від вихідного шару до вхідного. На кожному шарі обчислюється часткова похідна помилки щодо кожної ваги (градієнт), яка показує, як змінюється помилка при незначному збільшенні або зменшенні кожної ваги.
- Ці градієнти потім використовуються для оновлення ваг у мережі, зазвичай переміщуючи їх у напрямку, який зменшує похибку (градієнтний спуск).

Оптимізація градієнтного спуску:

- Градієнтний спуск - це алгоритм оптимізації, який використовується для мінімізації помилки (або втрат) шляхом ітеративного руху до мінімуму функції втрат.

- Основна ідея полягає в оновленні кожної ваги шляхом віднімання частини градієнта. Ця частка визначається параметром, відомим як швидкість навчання.
- Ключовою проблемою у використанні градієнтного спуску є вибір відповідної швидкості навчання. Занадто мала швидкість навчання робить процес навчання дуже повільним, тоді як занадто велика швидкість навчання може призвести до того, що алгоритм перевищить мінімум або навіть розійдеться.

Глибокі нейронні мережі (DNN) знайшли широке застосування в аналізі новин, пропонуючи складні рішення для вилучення значущої інформації зі складних фінансових даних.

Використання DNN у класифікації контенту фінансових новин:

- DNN особливо вправні в обробці та аналізі великих обсягів неструктурованих даних, що є загальною характеристикою контенту фінансових новин. Вони можуть розпізнавати закономірності та взаємозв'язки в даних, які не є очевидними, що робить їх ідеальними для таких завдань, як аналіз настроїв, прогнозування тенденцій та оцінка ризиків на фінансових ринках.
- Ці мережі можуть впоратися з тонкощами і нюансами фінансової мови, включаючи галузевий жаргон і складні семантичні структури, що дозволяє більш точно і глибоко аналізувати фінансові новини, звіти і контент соціальних мереж.
- Ключовою проблемою у використанні градієнтного спуску є вибір відповідної швидкості навчання. Занадто мала швидкість навчання робить процес навчання дуже повільним, тоді як занадто велика швидкість навчання може призвести до того, що алгоритм перевищить мінімум або навіть розійдеться.

Таким чином, DNN відіграють ключову роль в аналізі фінансових новин, надаючи розуміння складних фінансових даних. Їх здатність вчитися на великих

масивах даних і виявляти тонкі закономірності робить їх безцінним інструментом у різних фінансових додатках, від прогнозування ринкових тенденцій до оцінки ризиків і прийняття рішень у режимі реального часу.

2.2.2 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) виділяються серед моделей глибокого навчання насамперед завдяки своїй унікальній архітектурі, яка спеціально розроблена для обробки даних, що мають топологію, подібну до сітки, наприклад, зображень. Однак їх застосування виходить за рамки обробки зображень, що фактично робить їх універсальними інструментами для різних типів даних, включаючи фінансові новини.

CNN складається з низки шарів, кожен з яких виконує певні функції (рисунок 2.5) [20]. Основні шари включають згорткові шари, шари об'єднання (субдискретизації) та повністю з'єднані шари.

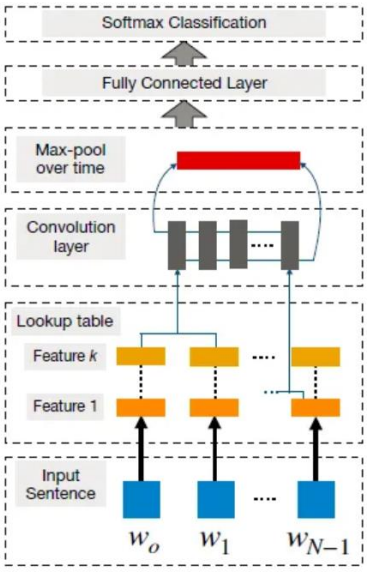


Рисунок 2.5 – Загальна архітектура згорткової нейронної мережі [20]

Архітектура CNN:

- Згорткові шари - це будівельні блоки CNN. Вони виконують операцію згортки, застосовуючи фільтри до вхідних даних для створення карт ознак. Ці фільтри - це невеликі матриці, що легко навчаються, які ковзають по вхідних даних, щоб виявити особливості, такі як краї на зображенні або візерунки в тексті.
- Об’єднувальні шари слідують за згортковими шарами і використовуються для зменшення просторових розмірів (ширини і висоти) вхідного об'єму для наступного згорткового шару. Об'єднання допомагає зменшити обсяг необхідних обчислень, а також робить виявлення об'єктів інваріантним до змін масштабу та орієнтації.
- Нарешті, після кількох шарів згортки та об'єднання, дані згладжуються і подаються в повністю з'єднані шари, подібно до звичайної нейронної мережі, для остаточної класифікації або регресії.

Як CNN обробляє дані:

- На відміну від традиційних нейронних мереж, де кожен вхідний вузол з'єднаний з кожним вихідним вузлом, CNN зберігають просторові зв'язки між елементами, організовуючи вивчені параметри (фільтри) у трьох вимірах (ширина, висота і глибина).
- У контексті обробки зображень це означає, що ранні шари можуть навчитися розпізнавати краї, кольори або текстури, тоді як глибші шари можуть розпізнавати складніші структури, такі як форми або конкретні об'єкти. Для текстових даних CNN можуть ідентифікувати важливі патерни в послідовностях слів або символів.
- Така архітектура дозволяє CNN бути високоефективними в обробці даних із сітчастою структурою. Мережа вчиться розпізнавати шаблони з надзвичайною варіативністю (наприклад, різні шрифти або почерк на зображеннях, синоніми або перефразування в тексті) і менш схильна до

перенавчання, особливо коли має справу з вхідними даними високої розмірності.

Згорткові нейронні мережі (CNN) вміють виокремлювати ознаки, що є критично важливим компонентом їхньої функціональності. Цей процес в першу чергу передбачає використання операцій згортки, які допомагають виокремити релевантні ознаки з вхідних даних.

Процес вилучення ознак в CNN:

- У цієї нейронній мережі вилучення ознак відбувається шляхом згортки, тобто застосування фільтра (або ядра) до вхідних даних. Під час згортки фільтр ковзає по вхідними даними, і в кожній позиції обчислюється точковий добуток між фільтром і частиною вхідних даних, яку він покриває.
- Ця операція створює карту ознак, яка представляє відфільтровану версію вхідних даних. На карті виділяються конкретні ознаки, які фільтр призначений виявляти. Наприклад, певний фільтр може бути налаштований на виявлення країв на зображенні або певних шаблонів слів у тексті.

Роль фільтрів у CNN:

- Фільтри в CNN - це невеликі матриці ваг, які мережа вивчає під час навчання. Ці ваги визначають, які ознаки фільтр буде виокремлювати з вхідних даних.
- Різні фільтри можуть виявляти різні типи об'єктів. При обробці зображень деякі фільтри можуть виявляти вертикальні краї, а інші - горизонтальні краї або певні текстури. В обробці тексту фільтри можуть виявляти певні послідовності символів або слів.
- Кожен фільтр генерує окрему карту ознак, тому застосування декількох фільтрів призводить до створення декількох карт ознак. Кожна карта ознак представляє окремий аспект або особливість вхідних даних.

Роль карт ознак у CNN:

- Карти ознак - це результати, отримані після застосування фільтрів до вхідних даних. Вони слугують стислим представленням вхідних даних, підкреслюючи специфічні особливості, на яких має зосередитися мережа.
- У глибоких CNN перший рівень може генерувати карти ознак, які представляють базові ознаки, такі як краї (на зображеннях) або прості шаблони слів (у тексті). У міру того, як просуваться вглиб мережі, наступні шари комбінують ці базові ознаки для виявлення більш складних патернів.
- Глибина карти особливостей відповідає кількості фільтрів, що використовуються в цьому шарі. Чим більше фільтрів, тим більше карт ознак, що призводить до багатшого представлення вхідних даних.

Згорткові нейронні мережі здобули популярність завдяки своїй винятковій продуктивності в аналізі зображень, але також довели свою ефективність в аналізі текстових даних. Їх здатність виокремлювати та вивчати закономірності робить їх особливо корисними в різних галузях, зокрема, в аналізі фінансових новин.

Застосування в аналізі текстових даних:

- Хоча спочатку CNN були розроблені для обробки зображень, їх застосування в аналізі тексту все частіше досліджується. У цьому контексті CNN розглядають текст як тип послідовних даних, де фільтри можуть вловлювати закономірності в словах або символах.
- В класифікації контенту фінансових новин, CNN може визначати ключові фрази, виявляти настрої або розпізнавати теми, вивчаючи шаблони на основі вкладених слів. Вставки слів перетворюють текст у числову форму, де семантично схожі слова представлені схожими векторами.
- Ця програма особливо корисна для аналізу настроїв у фінансових новинах, де розуміння контексту і тонкощів мови є життєво важливим. CNN може розрізняти позитивні, негативні або нейтральні настрої,

виражені в журналістських статтях, постах у соціальних мережах або аналітичних звітах.

Згорткові нейронні мережі мають низку переваг, особливо в обробці даних високої розмірності, але вони також мають певні обмеження та проблеми.

Переваги:

- Ефективне розпізнавання образів: CNN надзвичайно добре розпізнають шаблони у даних високої розмірності. Ця здатність особливо корисна в задачах розпізнавання зображень, де вхідні дані складаються з великих масивів значень пікселів.
- Виділення ознак: Однією з ключових переваг CNN є їхня здатність автоматично та адаптивно навчатися просторовим ієрархіям ознак на основі вхідних даних. Це робить їх придатними для завдань, які вимагають вилучення складних ознак з великих наборів даних, таких як виявлення тенденцій у фінансових новинах.
- Скорочена попередня обробка: На відміну від традиційних алгоритмів, CNN потребують меншої попередньої обробки даних. Вони можуть навчати фільтри, які в традиційних алгоритмах розроблялися б вручну, тим самим зменшуючи залежність від знань про предметну область.

Обмеження та виклики:

- Потреба у великих наборах даних: Для ефективного навчання CNN і запобігання надмірному пристосуванню часто потрібні великі набори даних. Це може бути значною проблемою в областях, де даних мало або їх отримання є дорогим.
- Обчислювальна інтенсивність: CNN можуть бути обчислювально інтенсивними, як з точки зору пам'яті, так і з точки зору обчислювальної потужності, особливо коли йдеться про дуже глибокі мережі та великі обсяги вхідних даних. Це вимагає потужного апаратного забезпечення, часто з високопродуктивними графічними процесорами.

- Схильні до перенавчання: Незважаючи на свою здатність до узагальнення, CNN все ще можуть бути схильні до надмірного навчання, особливо коли вони навчаються на невеликих наборах даних або коли архітектура мережі є надто складною порівняно зі складністю завдання.
- Відсутність інтерпретованості: CNN, як і багато моделей глибокого навчання, страждають від браку інтерпретованості. Зрозуміти, як мережа приймає рішення, може бути складно, що може бути суттєвим недоліком у сферах, де процеси прийняття рішень повинні бути прозорими, наприклад, у фінансовій сфері.

Таким чином, універсальність CNN в обробці як графічних, так і текстових даних робить їх дуже цінними в широкому спектрі застосувань. Їх здатність навчатися на складних, багатовимірних наборах даних особливо корисна в аналізі фінансових новин, де вони роблять значний внесок у розпізнавання шаблонів, аналіз настроїв і автоматизовану обробку інформації.

2.2.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) - це клас нейронних мереж, спеціально розроблених для обробки послідовних даних, що робить їх особливо придатними для таких завдань, як обробка природної мови, розпізнавання мови та аналіз часових рядів.

RNN характеризується унікальною архітектурою [21], де зв'язки між вузлами утворюють спрямований граф уздовж часової послідовності (рисунок 2.6). Така структура дозволяє йому демонструвати динамічну часову поведінку, на відміну від традиційних нейронних мереж, які припускають незалежні входи і виходи.

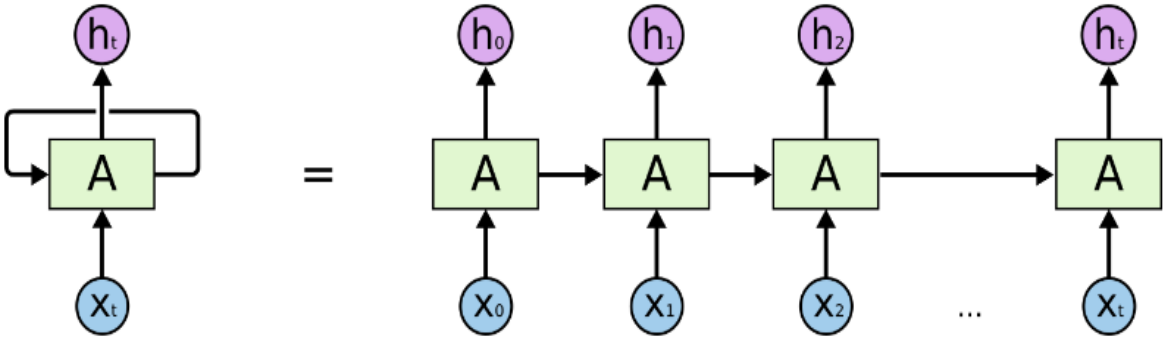


Рисунок 2.6 – Загальна архітектура рекурентної нейронної мережі [21]

Базова структура та функціональність RNN:

- Фундаментальною особливістю RNN є її "пам'ять" або здатність зберігати інформацію з попередніх входів у внутрішньому стані мережі, яка потім використовується для впливу на обробку наступних входів.
- Кожен нейрон у RNN отримує не лише вхідні дані з попереднього шару, але й вихідні дані з власного попереднього кроку. Цей повторюваний зв'язок і дає назву мереж та їхню особливу здатність.

Цикли в RNN та обробка послідовних даних:

- Механізм циклів у RNN має вирішальне значення для обробки послідовних даних. Кожен елемент послідовності обробляється по одному, і мережа підтримує "стан", який фіксує інформацію про те, що було оброблено до цього моменту.
- З практичної точки зору, стан в RNN можна розглядати як "пам'ять" про попередні вхідні дані, що дозволяє мережі приймати рішення на основі як поточних вхідних даних, так і контексту, наданого минулими вхідними даними.

- Ця здатність враховувати "контекст" вхідних даних робить RNN особливо ефективними для задач, де послідовність і порядок точок даних є важливими, наприклад, для прогнозування наступного слова в реченні або аналізу фінансових даних, залежних від часу.

Рекурентні нейронні мережі (RNN) особливо ефективні для аналізу послідовних даних, але вони стикаються зі значними проблемами при роботі з довгими послідовностями. Дві основні проблеми в цьому контексті - це проблеми зникання та вибуху градієнта.

Проблема зникання та вибуху градієнта:

- Проблема зникання градієнта виникає, коли градієнти функції втрат стають дуже малими, оскільки вони поширюються назад через кожен часовий крок мережі під час навчання. Це призводить до того, що навчання відбувається дуже повільно або взагалі не відбувається, оскільки ваги мережі майже не оновлюються.
- І навпаки, проблема вибуху градієнта виникає, коли градієнти стають надмірно великими, що призводить до нестабільних процесів навчання. У цьому випадку ваги можуть змінюватися занадто різко, що призводить до того, що модель розходиться і не збігається до розв'язку.
- Ці проблеми особливо виражені в RNN через їхню структуру та спосіб обробки послідовних даних у часі. Багаторазове множення градієнтів на часових кроках може призвести до цих проблем, особливо коли послідовності довгі.

Вплив на продуктивність довгих послідовностей:

- У випадку довгих послідовностей, градієнти повинні поширюватися через багато часових кроків, що посилює проблему зникання та вибуху градієнтів.
- При зниканні градієнтів RNN стає нездатним вивчати залежності між віддаленими елементами послідовності. Наприклад, при аналізі тексту

мережа може намагатися пов'язати слова в довгому реченні або абзаці, що впливає на такі завдання, як аналіз настроїв або моделювання тем.

- Проблема вибуху градієнтів може призвести до нестабільної поведінки моделі, коли ваги моделі непередбачувано змінюються від однієї ітерації до наступної. Це може зробити процес навчання нестабільним і часто призводить до того, що модель погано узагальнює.
- Ці проблеми можуть суттєво вплинути на продуктивність RNN в задачах, що включають довгі послідовності, обмежуючи їхню здатність ефективно вловлювати довгострокові залежності.

Вдосконалені варіанти рекурентних нейронних мереж, зокрема, з довгою короткочасною пам'яттю (LSTM) і двонаправленою LSTM, були розроблені для подолання обмежень традиційних RNN, особливо при обробці довгих послідовностей і фіксації довготривалих залежностей.

Довга короткочасна пам'ять:

- LSTM - це особливий тип RNN, спеціально розроблений для уникнення проблеми довготривалої залежності, з якою стикаються стандартні RNN. Вони можуть запам'ятовувати інформацію протягом тривалих періодів часу, що робить їх дуже ефективними для цілого ряду завдань моделювання послідовностей.
- Ключ до здатності LSTM зберігати інформацію протягом тривалих періодів часу - це його внутрішня структура, яка включає в себе такі вентилі, як вхідні, вихідні та забування.
- Ці вентилі регулюють потік інформації в комірку і з неї, дозволяючи LSTM зберігати або відкидати дані.
- Вхідний вентиль контролює ступінь надходження нового значення в комірку, вентиль забування контролює ступінь збереження значення в комірці, а вихідний вентиль контролює ступінь використання значення в комірці для обчислення вихідної активації LSTM-блоку.

- Така архітектура робить LSTM вправним у навчанні на важливому досвіді з дуже великими часовими проміками між ними, що дуже корисно для таких додатків, як прогнозування часових рядів, обробка природної мови та складних послідовних завдань, де розуміння контексту протягом тривалих періодів є критично важливим.

Двонаправлені LSTM:

- Двонаправлені LSTM є розширенням традиційних LSTM, які можуть покращити продуктивність моделі в задачах класифікації послідовностей. У цих мережах два LSTM, розташовані один над одним, обробляють вхідну послідовність як у прямому, так і в зворотному напрямках.
- Така структура дозволяє мережам мати як пряму, так і зворотну інформацію про послідовність на кожному часовому кроці. Це особливо корисно в контекстах, де розуміння всієї послідовності покращує продуктивність, наприклад, при розпізнаванні мови або аналізі настрою текстів.
- Обробляючи дані в обох напрямках, двонаправлені LSTM можуть більш ефективно вловлювати контекст, що призводить до кращого розуміння послідовності в цілому.

Порівняльний аналіз:

- Хоча стандартні LSTM мають високі можливості, двонаправлені LSTM пропонують додаткову перевагу в сценаріях, де контекст як минулого, так і майбутнього має вирішальне значення.
- Вибір між LSTM і Bidirectional LSTM залежить від конкретного застосування і характеру даних. Наприклад, при прогнозуванні часових рядів, коли майбутні точки даних недоступні, стандартний LSTM більш доречний. Однак у класифікації тексту або розпізнаванні мови, де вся послідовність відома, двонаправлені LSTM можуть запропонувати кращу продуктивність.

- Як LSTM, так і Bidirectional LSTM успішно застосовуються в різних галузях, демонструючи свою гнучкість і ефективність в обробці складних послідовних даних.

Застосування рекурентних нейронних мереж та їхніх вдосконалених варіантів, таких як довготривала короткочасна пам'ять (LSTM) та двонаправлена LSTM, для класифікації та аналізу фінансових новин є значним кроком вперед у розумінні часової динаміки та контексту фінансових даних.

Застосування в класифікації контенту фінансових новин:

- RNN особливо добре підходять для обробки послідовних даних, таких як фінансові новини, де порядок і контекст слів мають вирішальне значення для точної інтерпретації.
- Ці мережі можуть аналізувати текст фінансових новин у часі, фіксуючи еволюцію ринкових настроїв, відстежуючи зміни економічних показників і визначаючи тенденції в поведінці інвесторів.

Отже, використання RNN та їхніх вдосконалених варіантів для класифікації контенту фінансових новин є потужним інструментом для аналітиків та інвесторів. Використовуючи можливості послідовної та контекстної обробки цих моделей, фінансові фахівці можуть отримати більш глибоке розуміння динаміки ринку, настроїв інвесторів та потенційного впливу фінансових новин на інвестиційні рішення.

2.3 Засоби розробки

Набір засобів розробки моделей машинного навчання для класифікації контенту фінансових новин охоплює низку інструментів і бібліотек, кожна з яких

надає унікальні функціональні можливості для цього процесу. Нижче наведено структурований огляд цих компонентів.

2.3.1 Мова програмування Python

Python зарекомендував себе як провідна мова програмування для науки про дані та машинного навчання завдяки своїй доступності, універсальності та надійній екосистемі [22].

Огляд мови Python для обробки даних та машинного навчання:

- Популярність мови Python у сфері науки про дані та машинного навчання значною мірою зумовлена її зручним синтаксисом та читабельністю, що робить її ідеальним вибором як для початківців, так і для досвідчених програмістів.
- Її простота дозволяє розробникам зосередитися на вирішенні проблем машинного навчання, а не губитися в складнощах програмування.
- Філософія мови по відношенню до високорівневого проектування, яка дозволяє виражати концепції у меншій кількості рядків коду, ніж у таких мовах, як C++ або Java.

Можливості, які роблять Python оптимальним для машинного навчання:

- Широка підтримка бібліотек: Широкий вибір бібліотек та фреймворків Python є однією з його найбільших переваг. Такі бібліотеки, як NumPy та Pandas, спрощують маніпуляції з даними та їх аналіз, а Scikit-Learn надає інструменти для інтелектуального аналізу даних та машинного навчання. Для глибокого навчання такі бібліотеки, як TensorFlow і Keras, пропонують потужні платформи для побудови і навчання складних нейронних мереж.
- Спільнота та екосистема: Python виграє від великої спільноти розробників та дослідників даних. Ця спільнота сприяє створенню

багатої екосистеми бібліотек, інструментів та фреймворків, що робить Python універсальною мовою для широкого спектру застосунків у галузі науки про дані та машинного навчання.

- Інтеграція та гнучкість: Python можна легко інтегрувати з іншими мовами, що дозволяє використовувати її в різноманітних середовищах і для різних додатків. Ця гнучкість є безцінною в проектах машинного навчання, які часто вимагають інтеграції складних алгоритмів у виробничі системи.
- Швидке створення прототипів: Простота Python та багатий набір бібліотек дозволяють швидко створювати прототипи моделей машинного навчання. Розробники можуть швидко експериментувати з різними моделями та алгоритмами, прискорюючи ітеративний процес машинного навчання.
- Візуалізація даних: Такі бібліотеки, як Matplotlib та Seaborn у Python, пропонують потужні інструменти для візуалізації даних, що є критично важливим для аналізу та розуміння складних наборів даних у машинному навчанні.
- Крос-платформна сумісність: Python є кросплатформенною мовою, що означає, що програми на Python можуть працювати на різних операційних системах без необхідності внесення змін до коду.

Таким чином, поєднання простоти, підтримки бібліотек та потужної спільноти робить Python ідеальною мовою програмування для вирішення складних завдань машинного навчання та науки про дані. Її універсальність у роботі з різними аспектами проекту машинного навчання, від попередньої обробки та аналізу даних до побудови та оцінки моделей, підкреслює її позицію як основної мови у цій галузі, що стрімко розвивається.

2.3.2 Бібліотека Scikit-Learn

Scikit-Learn, яку часто називають sklearn, є однією з найбільш універсальних і широко використовуваних бібліотек Python для машинного навчання. Вона відома своєю простотою використання, продуктивністю та широкою сумісністю з екосистемою Python [23].

Огляд Scikit-Learn:

- Scikit-Learn побудований на основі NumPy та SciPy, двох інших бібліотек Python, які є основними в чисельних та наукових обчисленнях відповідно. Ця інтеграція забезпечує високу ефективність та безперешкодну маніпуляцію даними.
- Бібліотека пропонує чистий, уніфікований та впорядкований API, що робить її доступною для початківців, але водночас достатньо надійною та багатофункціональною для досвідчених дослідників даних.
- Вона включає в себе широкий спектр алгоритмів керованого і некерованого навчання, від фундаментальних методів, таких як лінійна регресія і кластеризація за методом k-середніх, до більш просунутих методів, таких як машини опорних векторів і випадкові ліси.

Функціональні можливості побудови, навчання та валідації моделей:

- Побудова моделі: Scikit-Learn надає прості та ефективні інструменти для інтелектуального аналізу даних. Його послідовний інтерфейс для різних алгоритмів дозволяє легко експериментувати з різними моделями. Користувачі можуть переключатися між моделями з мінімальними змінами в коді.
- Навчання: Бібліотека пропонує послідовний спосіб підбору моделей до даних за допомогою методу fit(), що спрощує процес навчання моделей на наборах даних. Вона може працювати з різними форматами і типами даних, забезпечуючи гнучкість у навчанні моделей.

- Оцінка та валідація моделей: Scikit-Learn оснащений безліччю інструментів для оцінки та валідації моделей. Функції для перехресної перевірки, різні методи оцінювання та метрики для класифікації, регресії та кластеризації є легкодоступними. Це допомагає оцінити продуктивність моделей і налаштувати їх для підвищення точності та ефективності.
- Створення конвеєра: Однією з ключових особливостей Scikit-Learn є можливість створення "конвеєра", який спрощує процес перетворення даних і застосування моделей. Це дозволяє автоматизувати багато завдань машинного навчання, таких як масштабування ознак, зменшення розмірності та налаштування моделі, у чіткий і лаконічний спосіб.
- Вилучення та попередня обробка даних: Бібліотека пропонує численні можливості для вилучення та попередньої обробки даних. Сюди входять інструменти для нормалізації, масштабування, обробки відсутніх значень і перетворення категоріальних даних у відповідний формат для моделей машинного навчання.

Таким чином, широкий спектр функціональних можливостей Scikit-Learn у поєднанні зі зручністю використання робить його наріжним каменем у Python для реалізації алгоритмів машинного навчання. Його здатність обробляти різні етапи процесу машинного навчання, від попередньої обробки даних до оцінки моделі, робить його безцінним активом як для початківців, так і для експертів у галузі машинного навчання.

2.3.3 Бібліотека Keras

Keras - це високорівневий нейромережевий API, розроблений для швидкого експериментування з глибоким навчанням, залишаючись при цьому зручним і

модульним. Він функціонує як інтерфейс для бібліотеки TensorFlow, спрощуючи процес створення моделей глибокого навчання [24].

Огляд Keras:

- Keras виступає в ролі високорівневого шару абстракції. Він може працювати поверх TensorFlow, Microsoft Cognitive Toolkit (CNTK) або Theano, забезпечуючи уніфікований інтерфейс для цих платформ. Спочатку розроблений як частина дослідницьких зусиль проекту ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), Keras зосереджується на тому, щоб бути зручним для користувача, модульним і розширюваним.
- Основна мета Keras - уможливити швидке експериментування з глибокими нейронними мережами. Він розроблений для легкого та швидкого створення прототипів, підтримуючи як згорткові мережі, так і рекурентні мережі, а також їх комбінації.
- Keras особливо відомий своєю простотою та легкістю використання, що робить його доступним для широкого кола користувачів, від початківців до експертів у галузі глибокого навчання.

Спрощення створення моделей глибокого навчання:

- Keras спрощує створення моделей глибокого навчання завдяки зручному та модульному інтерфейсу. Він дозволяє розробникам визначати моделі шар за шаром в дуже інтуїтивно зрозумілий спосіб. Кожен шар нейронної мережі доступний як простий, окремий модуль, що дозволяє легко створювати та модифікувати складні архітектури.
- API абстрагує більшу частину складнощів, які зазвичай пов'язані з налаштуванням нейронних мереж, від ініціалізації параметрів до визначення оптимізаторів і функцій втрат. Ця абстракція дозволяє користувачам більше зосередитися на побудові своїх моделей, а не занурюватися в технічні деталі.

- Keras підтримує широкий спектр готових шарів, функцій активації та оптимізаторів, що дозволяє користувачам швидко та ефективно експериментувати з різними архітектурами та рішеннями. Він також включає в себе безліч утиліт для полегшення таких завдань, як попередня обробка даних, оцінка моделі та її точне налаштування.
- Незважаючи на свою простоту, Keras не жертвує гнучкістю та продуктивністю. Досвідчені користувачі можуть легко розширити Keras відповідно до своїх потреб, створюючи власні шари, нові функції активації чи навчальні програми.

Таким чином, Keras виділяється як потужний інструмент для глибокого навчання завдяки своїй орієнтованості на зручність для користувача та можливості швидкого створення прототипів. Він відкриває двері до глибокого навчання, роблячи його більш доступним і знижує бар'єр для тих, хто хоче поекспериментувати з передовими архітектурами нейронних мереж. Як для академічних досліджень, так і для промислових застосувань, Keras надає ефективну та гнучку платформу для побудови та розгортання моделей глибокого навчання.

2.3.4 Бібліотека Pandas

Pandas - це потужна і широко використовувана бібліотека на мові Python, спеціально розроблена для маніпулювання та аналізу даних. Вона стала незамінним інструментом для дослідників даних та аналітиків, які працюють на Python, особливо в контексті машинного навчання [25].

Огляд Pandas:

- Розроблена Весом МакКінні, Pandas - це бібліотека з відкритим вихідним кодом, яка надає високопродуктивні, прості у використанні структури даних та інструменти для аналізу даних. Назва Pandas

походить від терміну "панельні дані", економетричного терміну для багатовимірних структурованих наборів даних.

- В основі Pandas лежать дві основні структури даних: DataFrame і Series. DataFrame - це двовимірна, змінна за розміром і потенційно неоднорідна таблична структура даних з позначеними осями (рядками і стовпчиками). З іншого боку, серія - це одновимірний маркований масив, який може містити дані будь-якого типу.
- Pandas відмінно справляється з обробкою та маніпулюванням структурованими даними, типовими для багатьох форм інформації у реальному світі, включаючи фінансові дані, дані часових рядів і табличні дані з різнотипними стовпчиками.

Можливості в обробці даних:

- Очищення та підготовка даних: Pandas надає широкі можливості для очищення та підготовки даних, що робить його ідеальним інструментом для попередньої обробки даних перед їх подачею в моделі машинного навчання. Він може легко обробляти відсутні дані, дублікати даних і маніпуляції з рядками.
- Аналіз даних: Завдяки широкому набору функцій для описової статистики, він дозволяє користувачам швидко зрозуміти розподіл, кількість, середнє значення та інші статистичні властивості своїх даних.
- Перетворення даних: Pandas дозволяє легко перетворювати, змінювати форму, сортувати та фільтрувати дані. Такі операції, як злиття та приєднання наборів даних, є простими, що полегшує комбінування даних з різних джерел.
- Ефективне зберігання та читання даних: Pandas надає інструменти для ефективного читання та запису даних у різних форматах, таких як CSV, Excel, JSON, HTML та бази даних SQL.

Таким чином, Pandas є наріжним каменем бібліотеки Python для маніпулювання та аналізу даних. Надійні функції для обробки та аналізу структурованих даних роблять її важливим інструментом для будь-якого проекту машинного навчання, особливо в класифікації контенту фінансових новин, де структуровані дані є нормою. Бібліотека не тільки спрощує завдання попередньої обробки даних, але й підвищує загальну ефективність процесу аналізу даних.

2.3.5 Бібліотека NumPy

NumPy, що розшифровується як Numerical Python, є фундаментальним пакетом для наукових обчислень на Python. Це основна бібліотека, яка широко використовується для числових і математичних операцій, необхідних у різних галузях, включаючи машинне навчання, науку про дані та інженерію [26].

Огляд NumPy:

- NumPy забезпечує підтримку великих багатовимірних масивів і матриць, а також набір високорівневих математичних функцій для роботи з цими масивами. Він побудований на мові C, що дозволяє ефективно виконувати числові операції.
- Однією з ключових особливостей NumPy є потужний об'єкт N-вимірного масиву, відомий як ndarray. Цей об'єкт є швидким і гнучким контейнером для великих наборів даних у Python. Масиви в NumPy є більш ефективними для зберігання та маніпулювання даними порівняно зі списками в Python, особливо коли дані зростають у розмірі.
- Здатність бібліотеки виконувати поелементні операції (трансляцію) над масивами є значною перевагою, що дозволяє писати стислий та ефективний код. Це особливо корисно у машинному навчанні, де операції часто потрібно виконувати над великими масивами даних.

Використання для обробки масивів і матриць у машинному навчанні:

- У машинному навчанні масиви NumPy використовуються для обробки числових даних перед застосуванням алгоритмів. Незалежно від того, чи це для зберігання навчальних даних, ваг нейронних мереж або вихідних прогнозів, NumPy забезпечує необхідну структуру для цих завдань.
- NumPy відмінно справляється з матричними операціями, які мають вирішальне значення в багатьох алгоритмах машинного навчання. Такі завдання, як множення матриць, транспонування та операції лінійної алгебри (наприклад, розв'язання лінійних систем, розкладання за власними значеннями) ефективно вирішуються за допомогою NumPy.
- Інтеграція NumPy з іншими бібліотеками, такими як Pandas, Matplotlib та Scikit-Learn, робить його центральним компонентом в екосистемі машинного навчання Python. Наприклад, дані, витягнуті або оброблені за допомогою Pandas, можуть бути легко перетворені в масиви NumPy для задач машинного навчання.
- Модуль випадкових величин NumPy - ще одна корисна функція для машинного навчання, яка використовується для перемішування даних, випадкової вибірки та генерації випадкових чисел для початкових ваг у нейронних мережах.

Таким чином, роль NumPy в машинному навчанні є ключовою. Її потужні операції з масивами та ефективність в обробці числових обчислень роблять її незамінним інструментом для науковців з даних та фахівців з машинного навчання. Бібліотека не тільки надає базові структури для зберігання та маніпулювання даними, але й пропонує набір математичних функцій, критично важливих для розробки та впровадження алгоритмів машинного навчання.

2.3.6 Бібліотека Matplotlib

Matplotlib - це широко використовувана бібліотека Python для створення статичних, анімованих та інтерактивних візуалізацій. Вона особливо відома своєю здатністю створювати 2D-графіки та діаграми, які мають вирішальне значення у сфері аналізу даних та машинного навчання [27].

Огляд Matplotlib:

- Розроблений Джоном Д. Хантером, Matplotlib став однією з перших і найпопулярніших бібліотек Python для візуалізації даних. Вона пропонує широкий спектр функцій для створення різних типів графіків і діаграм, включаючи гістограми, стовпчикові діаграми, діаграми розсіювання і лінійні графіки.
- Matplotlib дуже добре налаштовується і пропонує широкий спектр опцій для пристосування зовнішнього вигляду графіків. Користувачі можуть налаштувати майже кожен елемент діаграми, включаючи розміри, кольори, шрифти та макети, що дозволяє створювати діаграми публікаційної якості.
- Він легко інтегрується з іншими бібліотеками обробки даних, такими як NumPy та Pandas, що робить його зручним вибором для візуалізації складних наборів даних. Графіки Matplotlib можна вбудовувати в додатки з графічним інтерфейсом Python або відображати в блокнотах Jupyter, що робить його універсальним для різних випадків використання.

Роль в аналізі та інтерпретації даних у машинному навчанні:

- Дослідження даних: На початкових етапах проекту машинного навчання Matplotlib є безцінним інструментом для розвідувального аналізу даних (EDA). Він допомагає зрозуміти розподіл, тенденції та закономірності в даних за допомогою візуальних засобів, які можуть бути неочевидними з необроблених даних.

- Оцінка моделі: Візуалізації, створені за допомогою Matplotlib, допомагають інтерпретувати результати моделей машинного навчання. Такі графіки, як матриці плутанини, ROC-криві та криві навчання, дають уявлення про продуктивність алгоритмів.
- Вагомість та вибір ознак: Графічні зображення можуть показати взаємозв'язки між різними ознаками в наборі даних та їхню важливість для прогнозування цільової змінної. Ця інформація має вирішальне значення для відбору ознак та інженерних процесів.
- Поширення результатів: Matplotlib допомагає передавати результати проекту машинного навчання зацікавленим сторонам. Складні результати і тенденції можуть бути ефективно передані за допомогою візуальних зображень, що робить їх більш доступними для нетехнічної аудиторії.

Таким чином, Matplotlib слугує основним інструментом для візуалізації даних у середовищі машинного навчання Python. Його широкі функціональні можливості дозволяють фахівцям ефективно досліджувати, аналізувати, інтерпретувати та передавати дані і результати моделювання, тим самим відіграючи вирішальну роль в успіху проектів машинного навчання.

2.3.7 Бібліотека NLTK

Natural Language Toolkit, широко відомий як NLTK, є провідною платформою для створення програм на мові Python для роботи з даними людської мови. Це комплексний набір бібліотек і програм для символьної та статистичної обробки природної мови (NLP) для мови Python [28].

Огляд NLTK:

- NLTK було створено для підтримки досліджень і викладання в галузі NLP і тісно пов'язаних з нею областях, включаючи емпіричну

лінгвістику, когнітивну науку, штучний інтелект, інформаційний пошук і машинне навчання.

- Він надає прості у використанні інтерфейси до понад 50 корпусів і лексичних ресурсів, таких як WordNet, а також набір бібліотек для обробки тексту для класифікації, токенізації, виокремлення, тегування, синтаксичного та семантичного аналізів.
- Однією з ключових переваг NLTK є його велика документація та навчальні посібники, що робить його ідеальним інструментом навчання для людей, які починають працювати в галузі NLP. Він широко використовується в академічних колах і слугує інструментом навчання, дослідницьким прототипом і середовищем розробки додатків промислового рівня.

Застосування в обробці та аналізі тексту:

- Обробка тексту: NLTK включає бібліотеки для розбиття тексту на токени (слова і речення), що часто є першим кроком в аналізі тексту. Вона також надає інструменти для стемінгу та лематизації - процесів, які зводять слова до їхньої кореневої форми.
- Класифікація тексту: NLTK оснащений алгоритмами для класифікації тексту, такими як визначення жанру тексту або настрою коментаря. Це особливо актуально для класифікації контенту фінансових новин, де визначення настрою може суттєво вплинути на прийняття рішень.
- Тегування за частинами мови: NLTK може призначити частину мови кожному слову в даному тексті, наприклад, іменнику, дієслову, прикметнику тощо. Це корисно для розуміння структури речень і вилучення певних типів інформації з текстів.
- Розпізнавання іменованих об'єктів (Named Entity Recognition): Включає можливості для NER, що передбачає ідентифікацію та категоризацію ключової інформації в тексті, такої як назви компаній, дати та місцезнаходження. У класифікації фінансових новин це допомагає

класифікувати та визначати пріоритетність новин на основі конкретних об'єктів.

- Синтаксичний аналіз: NLTK також надає інструменти для синтаксичного аналізу речень, щоб зрозуміти їхню структуру, що є важливим у більш складних завданнях NLP, які вимагають глибокого розуміння синтаксису мови.

Таким чином, широкий спектр функціональних можливостей NLTK робить його безцінним інструментом для всіх, хто працює з даними людської мови. Її застосовність у різних аспектах NLP, від базової обробки тексту до складних завдань класифікації, підкреслює її корисність у сфері класифікації фінансових новин. Бібліотека не тільки підтримує різні етапи аналізу тексту, але й забезпечує основу для створення складних NLP-додатків.

Висновки до розділу 2

У цьому розділі було детально розглянуто інструменти та методи, які є невід'ємною частиною розробки моделей машинного навчання, зокрема тих, що зосереджені на класифікації контенту фінансових новин. Цей огляд охоплює алгоритми машинного навчання та глибокого навчання, а також вивчення основного набору засобів розробки, що використовується в цих процесах.

Було проведено розгляд різних алгоритмів машинного навчання, включаючи логістичну регресію, машину опорних векторів (SVM), випадковий ліс і мультиноміальний наївний Байєс. Кожен з цих алгоритмів обговорювався з точки зору їхніх фундаментальних принципів, робочих механізмів і конкретних застосувань у контексті аналізу фінансових новин. Це дослідження дало уявлення про те, як різні алгоритми унікально підходять для обробки нюансів і складнощів даних фінансових новин.

Було розглянуто різноманітні алгоритми глибокого навчання, зосереджено увагу надав глибоким нейронним мережам (DNN), згортковим нейронним мережам (CNN) та рекурентним нейронним мережам (RNN). Аналіз цих передових алгоритмів висвітлило їхню архітектуру, функціональність та зростаюче значення у вирішенні складних завдань, таких як аналіз настроїв, прогнозування трендів та розпізнавання шаблонів у фінансових новинах.

Щодо набору засобів розробки, який має вирішальне значення для реалізації цих алгоритмів. Було розглянуто Python, універсальну мову програмування, а також ряд бібліотек, включаючи Scikit-Learn, Keras, Pandas, NumPy, Matplotlib та NLTK. Кожна з цих бібліотек робить унікальний внесок у процес розробки, впровадження та оптимізації моделей машинного навчання.

Аналіз, проведений в цьому розділі, надало розуміння різних інструментів і методологій, які є в розпорядженні для класифікації контенту фінансових новин за допомогою машинного навчання. Від базових методів до складних моделей глибокого навчання, окреслено сильні сторони, обмеження та конкретні сфери

застосування кожного підходу. Ці знання є основою для розробки надійних та ефективних моделей, здатних точно класифікувати та аналізувати новини, тим самим надаючи цінну інформацію про ринкові тенденції та настрої новин.

В розділі запропоновано розгляд технічних аспектів машинного навчання та алгоритмів глибокого навчання, висвітлено практичне значення цих інструментів у контексті аналізу текстових даних. Інтеграція цих алгоритмів з екосистемою Python за допомогою різних бібліотек підкреслює динамічний характер машинного навчання в класифікації контенту фінансових новин.

3 ПОБУДОВА МОДЕЛЕЙ КЛАСИФІКАЦІЇ

У цьому розділі продемонстровано розробка моделей класифікації контенту фінансових новин, розглянуто етапи побудови моделі, такі як: розгляд датасету, попередня обробка, відбір ознак і навчання. Набір даних складається з прикладів, що класифікуються на такі настрої новини: нейтральна, позитивна та негативна. В якості датасету було обрано з платформи Kaggle Sentiment Analysis for Financial News [29].

3.1 Набір даних

Набір даних відіграє ключову роль при створенні моделей машинного навчання, в першу чергу слугуючи основою для розуміння основних закономірностей і нюансів фінансових даних [30]. Також передбачано ознайомлення з набором даних фінансових новин з метою виявлення ключових тенденцій, аномалій та внутрішніх характеристик, які можуть бути неочевидними з першого погляду. Це має вирішальне значення для прийняття обґрунтованих рішень, оскільки надає цінну інформацію про ринкові настрої, поведінку інвесторів та потенційні фінансові результати. Після ознайомлення, можна ефективно готувати та уточнювати дані, гарантуючи, що подальше побудова та аналіз ґрунтуватимуться на глибокому розумінні фундаментальних атрибутів набору даних.

На рисунку 3.1 наведено типовий процес ознайомлення з даними перед обробкою, а саме: кількість, розподіл даних за трьома категоріями, приклди з датасету, наявність відсутніх значень.

```python
import matplotlib.pyplot as plt

sentiment_distribution = data['Sentiment'].value_counts()

plt.figure(figsize=(10,6))
sentiment_distribution.plot(kind='bar', color=['skyblue', 'lightcoral', 'lightgreen'])
plt.title('Distribution of Sentiments')
plt.xlabel('Sentiment')
plt.ylabel('Number of News')
plt.xticks(rotation=45)
plt.show()

sample_news = data.groupby('Sentiment').apply(lambda x: x.sample(2)).reset_index(drop=True)

missing_values = data.isnull().sum()

sentiment_distribution, sample_news, missing_values
```

Рисунок 3.1 – Процес ознайомлення з даними

На рисунку 3.2 наведено розподіл набору даних за категоріями.

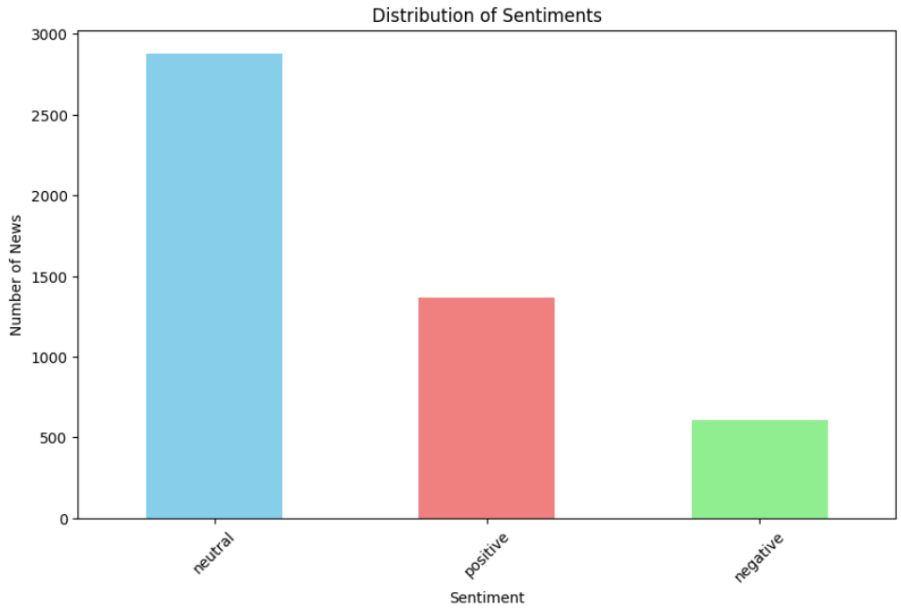


Рисунок 3.2 – Процес ознайомлення з даними

Згідно з розподілу, ключові висновки, які можна підсумувати наступним чином:

- "Нейтральна" категорія має найвищий показник, що свідчить про те, що більшість даних підпадає під цю класифікацію.
- "Негативна" категорія має найнижчий показник, що свідчить про те, що є менше випадків, коли дані позначені як негативні.
- Категорія "Позитивна" має помірну кількість, більше, ніж негативних, але значно менше, ніж нейтральних.

На рисунку 3.3 можна побачити кількість новин, малу вибірку з датасету та відсутність значень.

```
(neutral     2879
 positive    1363
 negative     604
 Name: Sentiment, dtype: int64,
   Sentiment                                               News
 0  negative  In the Baltic states the company reports net s...
 1  negative  According to CEO Matti Perkonoja of the parent...
 2   neutral  Koff 's market share of the volume of the mark...
 3   neutral  According to statistics by the Finnish Maritim...
 4  positive  Suominen Corporation estimates that the cost-c...
 5  positive  Finnish pharmaceuticals company Orion 's net s...,
 Sentiment    0
 News         0
 dtype: int64)
```

Рисунок 3.3 – Характеристики набору даних

Ключові висновки з цих характеристик такі:

- Набір даних не має відсутніх значень.
- Розподіл і його форма можуть вплинути на продуктивність алгоритмів обробки природної мови, оскільки дуже короткі або дуже довгі тексти можуть бути викидами, які можуть вплинути на навчання моделі, якщо їх не обробляти належним чином.

Розуміння цих аспектів даних є фундаментальним для підготовки до попередньої обробки, та побудови, наприклад, для вибору правильних методів нормалізації тексту, прийняття рішення про обмеження кількості слів або довжини

тексту для вхідних даних моделі, а також для виявлення текстів, які можуть потребувати спеціальної обробки.

На етапі ознайомлення з набором даних отримав кількість новин, візуалізував розподіл даних за категоріями та перевірив на предмет відсутності значень. Отримані результати виявили домінуючі тенденції настроїв: концентрацію нейтральних оцінок з меншою кількістю випадків крайніх позитивних чи негативних настроїв. Такий розподіл свідчить про незбалансований набір даних.

3.2 Попередня обробка даних

Попередня обробка даних - важливий крок у процесі машинного навчання. Він передбачає підготовку та перетворення необроблених даних у формат, придатний для побудови та навчання моделей машинного навчання. Необхідність цього процесу виникає через недосконаліть реальних даних, які часто містять шум, пропущені значення, невідповідності та нерелевантну інформацію, що може негативно вплинути на продуктивність прогнозування моделей.

Мета попередньої обробки даних - очистити та організувати дані таким чином, щоб підвищити їхню якість і забезпечити їхню придатність для ефективного використання алгоритмами машинного навчання. Це може включати в себе ряд заходів, таких як:

- Очищення: Видалення або виправлення помилкових даних.
- Нормалізація: Масштабування числових даних до стандартного діапазону.
- Кодування: Перетворення категоріальних змінних на числові значення, які моделі можуть інтерпретувати.
- Інтерполяція: Обробка відсутніх значень шляхом заміни їх статистичним припущенням.

- Вилучення ознак: Зменшення розмірності даних або вилучення значущих атрибутів.
- Вибірка: Усунення дисбалансів у наборі даних шляхом зменшення вибірки класу більшості або збільшення вибірки класу меншості.

На рисунку 3.4 зображено процес виконання зазначеного ряду заходів.

```
from sklearn.utils import resample

neutral_data = data[data['Sentiment'] == 'neutral']
positive_data = data[data['Sentiment'] == 'positive']
negative_data = data[data['Sentiment'] == 'negative']

neutral_downsampled = resample(neutral_data,
                               replace=False,
                               n_samples=len(positive_data),
                               random_state=42)

positive_upsampled = resample(positive_data,
                              replace=True,
                              n_samples=len(neutral_downsampled),
                              random_state=42)

negative_upsampled = resample(negative_data,
                              replace=True,
                              n_samples=len(neutral_downsampled),
                              random_state=42)

resampled_data = pd.concat([neutral_downsampled, positive_upsampled, negative_upsampled])

resampled_data = resampled_data.sample(frac=1, random_state=42).reset_index(drop=True)

resampled_distribution = resampled_data['Sentiment'].value_counts()

resampled_distribution
```

```
neutral     1363
positive    1363
negative    1363
Name: Sentiment, dtype: int64
```

Рисунок 3.4 – Попередня обробка даних

Ці кроки мають вирішальне значення для продуктивності моделей машинного навчання, оскільки вони гарантують, що вхідні дані будуть точно представлені, з мінімальним шумом і максимальною релевантністю, дозволяючи

моделям навчатися на основі справжніх базових закономірностей, а не артефактів процесу збору даних.

3.2.1 Токенізація

Токенізація в контексті обробки текстових даних - це фундаментальний крок, на якому текст розбивається на менші одиниці, які називаються токенами. Це можуть бути слова, числа або розділові знаки. Цей процес (рисунок 3.5) має вирішальне значення для перетворення неструктурованого тексту в структуровану форму, яку алгоритми машинного навчання можуть зрозуміти і проаналізувати.

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

# Define parameters
MAX_NUM_WORDS = 5000  # Top 5000 most frequent words
MAX_SEQUENCE_LENGTH = 250  # Max length of sequences

# Tokenize the text data
tokenizer = Tokenizer(num_words=MAX_NUM_WORDS, lower=True)
tokenizer.fit_on_texts(resampled_data['Processed_News'])
sequences = tokenizer.texts_to_sequences(resampled_data['Processed_News'])

# Pad sequences to ensure consistent length
X_dl = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

# Convert sentiments to numerical labels
sentiment_mapping = {'neutral': 0, 'positive': 1, 'negative': 2}
y_dl = resampled_data['Sentiment'].replace(sentiment_mapping).values
```

Рисунок 3.5 – Токенізація набору даних перед побудової алгоритмів нейронних мереж

У наборах даних фінансових новин токенізація слугує для розбиття складних фінансових наративів на зрозумілі частини, що дозволяє точно виокремити інсайти.

Кожен токен діє як індивідуальна особливість для алгоритмічної інтерпретації, допомагаючи в таких завданнях, як аналіз настроїв, ідентифікація трендів і моделювання тем. Потім токени часто піддаються подальшій обробці за допомогою таких методів, як стеммінг, лематизація та видалення стоп-слів, щоб уточнити дані для додатків машинного навчання.

3.2.2 Видалення стоп-слів

Видалення стоп-слів - загальновживаних слів у мові, які несуть мінімальну інформативність (таких як "the", "is", "at" тощо) - є важливим кроком в аналізі тексту. Він допомагає зосередитися на словах, які мають більше значення і роблять внесок у зміст тексту, що особливо важливо у фінансових новинах, де ключова інформація може бути прихована за великою кількістю шуму.

Процес зазвичай включає фільтрацію стоп-слів з токенізованого тексту. Це робиться за допомогою попередньо визначених списків стоп-слів, що надаються бібліотеками обробки природної мови, такими як NLTK в Python. У контексті фінансових новин видалення стоп-слів зменшує розмірність набору даних і покращує продуктивність обчислювальних моделей, завданням яких є розуміння і прогнозування поведінки ринку на основі текстових даних.

3.2.3 Стемінг

Стеммінг - це метод нормалізації тексту, який використовується в обробці природної мови для зведення слів до їхньої кореневої або базової форми. Він передбачає відсікання кінців слів для пошуку базової форми, яка може не збігатися з морфологічним коренем слова, але є достатньою для об'єднання пов'язаних слів в один токен.

В аналізі даних фінансових новин стеммінг дозволяє об'єднати кілька форм слова в одне представлення, тим самим спрощуючи набір даних. Наприклад, "інвестувати", "інвестує", "інвестування" та "інвестований" будуть утворені від "інвестувати". Таке спрощення полегшує аналіз, гарантуючи, що різні варіації терміну розпізнаються як один і той самий, підвищуючи точність пошуку за ключовими словами та обчислювальних моделей, що використовуються для аналізу настроїв або прогнозування тенденцій у фінансовому контексті.

3.2.4 Лематизація

Лематизація - це лінгвістичний процес, який передбачає скорочення слова до його базової або словникової форми, відомої як лема. На відміну від стеммінгу, який грубо обрубує кінці слів, що часто призводить до неправильного значення і написання, лематизація враховує контекст і трансформує слова до їхньої значущої базової форми. Наприклад, "am", "are" і "is" лематизуються в "be".

При аналізі наборів даних фінансових новин лематизація є корисною, оскільки допомагає точно об'єднати різні форми слова, гарантуючи, що набір даних точно відображає частоту обговорюваних концепцій і тем. Скорочуючи слова до їхніх лем, фінансові тексти можна ефективніше аналізувати на предмет тенденцій, настроїв і ключових тем, підвищуючи якість інсайтів, отриманих за допомогою моделей текстової аналітики.

3.3 Відбір ознак

Відбір ознак у машинному навчанні - це процес вибору підмножини релевантних ознак для використання в побудові моделі. Він підвищує продуктивність моделей машинного навчання, усуваючи нерелевантні або

надлишкові дані, зменшуючи складність і підвищуючи точність. У контексті аналізу тексту вибір ознак має вирішальне значення для визначення того, які слова, фрази або похідні характеристики (наприклад, оцінки TF-IDF) є найбільш інформативними для прогнозування результатів, наприклад, настроїв у статтях фінансових новин. Зосереджуючись лише на найбільш значущих ознаках, моделі можуть навчатися швидше і є більш інтерпретованими, що часто призводить до кращого узагальнення невидимих даних.

3.3.1 Модель bag-of-words

Модель Bag-of-Words (BoW) - це спрощене представлення, яке використовується в обробці природної мови та пошуку інформації. У цій моделі текст (наприклад, речення або документ) представляється у вигляді мішка (множини) слів, без урахування граматики і навіть порядку слів, але зі збереженням множинності. У контексті класифікації тексту модель BoW перетворює текстові документи в числові вектори, де кожен документ представлений вектором, що вказує на наявність і частоту слів із заздалегідь визначеного словника термінів. У застосуванні до аналізу фінансових новин BoW може ефективно інкапсулювати входження фінансово релевантних термінів і фраз, забезпечуючи базовий набір ознак для подальших завдань класифікації, таких як аналіз настроїв або прогнозування трендів.

3.3.2 Статистичний показник TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) - це числова статистика, призначена для відображення важливості слова для документа в колекції або корпусі. Його часто використовують як ваговий коефіцієнт у

пошукових системах для пошуку інформації, текстового аналізу та моделювання користувачів. Значення TF-IDF зростає пропорційно до кількості разів, коли слово з'являється в документі, але компенсується частотою слова в корпусі, що допомагає врахувати той факт, що деякі слова загалом з'являються частіше. У сфері класифікації фінансових новин TF-IDF може бути особливо ефективним у розрізненні документів на певні теми, виділяючи унікальні терміни, які не так часто зустрічаються в інших документах. Це може значно підвищити точність і достовірність алгоритмів класифікації, надаючи більшої ваги більш промовистим термінам, що мають відношення до фінансового контексту.

3.4 Побудова моделей класифікації

При побудові моделей класифікації різні алгоритми навчаються передбачати категоричні результати. Для фінансових новин ці моделі можуть класифікувати статті за такими категоріями, як позитивні, негативні або нейтральні настрої, які потім можуть бути використані для прогнозування ринкових тенденцій або руху акцій. Вибір моделі, чи то логістична регресія, машини опорних векторів, чи більш складні структури глибокого навчання, залежатиме від конкретних вимог аналізу, характеру даних і бажаного результату прогнозування.

Кожен з цих етапів призначені для покращення класифікації контенту фінансових новин, які можуть бути обтяжені галузевим жаргоном та складними структурами речень. Спрощуючи та структуруючи ці дані, стає можливим отримати значущу інформацію, яка може вплинути на фінансові рішення та стратегії.

3.4.1 Моделі машинного навчання

Моделі навчання з навчанням під наглядом - це підмножина алгоритмів машинного навчання, які працюють з маркованими наборами даних. Вони називаються "керованими", тому що процес навчання алгоритму (рисунок 3.6) на навчальному наборі даних можна уявити як вчителя, який контролює процес навчання. І для подальшого обчислювального експерименту щодо метрик оцінювання моделей необхідно виконати розділення новин (рисунок 3.6) на тренеруючий та тестовий набір даних

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

X_train.shape, X_test.shape
```

Рисунок 3.6 – Розділення набору даних

На рисунку 3.7 зображено етап створення та навчання моделей машинного навчання.

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB


logistic_model = LogisticRegression(max_iter=1000, random_state=42)
random_forest_model = RandomForestClassifier(random_state=42)
svm_model = SVC(kernel='linear', probability=True, random_state=42)
naive_bayes_model = MultinomialNB()

logistic_model.fit(X_train, y_train)
random_forest_model.fit(X_train, y_train)
svm_model.fit(X_train, y_train)
naive_bayes_model.fit(X_train, y_train)
```

Рисунок 3.7 – Створення та навчання моделей машинного навчання

Під час створення моделей машинного навчання необхідно було виконати такі кроки:

- Логістична регресія: ініціалізовано максимум 1000 ітерацій (max_iter=1000), щоб алгоритм оптимізації збігався. Параметр random_state=42 забезпечує відтворюваність результатів, надаючи генератору випадкових чисел фіксовану початкову величину.
- Машина опорних векторів (SVM): використано лінійне ядро (kernel='linear'), придатне для лінійно відокремлюваних даних. Ймовірнісні оцінки увімкнено (probability=True), що дозволяє використовувати метод predict_proba для отримання ймовірнісних оцінок для кожного класу. Параметр random_state=42 забезпечує відтворюваність результатів.
- Наївний Байєс: модель використовується з параметрами за замовчуванням такі, як alpha=1.0 (параметр адитивного згладжування), fit_prior=True (параметр вказує, чи дізнаватися попередні ймовірності класів чи ні), class_prior=None (дозволяє класифікатору автоматично визначати ймовірність кожного класу з навчальних даних).
- Випадковий ліс: налаштовано з random_state=42, щоб підтримувати узгодженість результатів під час декількох прогонів, контролюючи випадковість завантаження зразків, що використовуються під час побудови дерев.

Кожна з цих моделей має свої сильні сторони і може бути обрана на основі конкретних характеристик фінансових текстових даних і складності завдання аналізу настроїв.

3.4.2 Моделі нейронних мереж

Нейромережеві моделі представляють потужний клас алгоритмів машинного навчання, які чудово справляються з обробкою та інтерпретацією складних закономірностей у даних. Для аналізу фінансових новин різні архітектури нейронних мереж пропонують різні підходи.

Глибокі нейронні мережі (DNN) - це багатошарові мережі (рисунок 3.8), здатні вловлювати складні структури в даних. Вони підходять для багатофункціональних фінансових новин, дозволяючи здійснювати глибокий аналіз та інтерпретацію.

```
from keras.models import Sequential
from keras.layers import Dense, Embedding, Flatten, Dropout
from keras.optimizers import Adam
from keras.utils import to_categorical

# Convert labels to one-hot encoding
y_train_dl_onehot = to_categorical(y_train_dl)
y_test_dl_onehot = to_categorical(y_test_dl)

# Define DNN model
dnn_model = Sequential()
dnn_model.add(Embedding(input_dim=MAX_NUM_WORDS, output_dim=128, input_length=MAX_SEQUENCE_LENGTH))
dnn_model.add(Flatten())
dnn_model.add(Dense(128, activation='relu'))
dnn_model.add(Dropout(0.5))
dnn_model.add(Dense(3, activation='softmax'))

# Compile the model
dnn_model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
dnn_history = dnn_model.fit(X_train_dl, y_train_dl_onehot, epochs=5, batch_size=32, validation_split=0.2, verbose=1)
```

```
Epoch 1/5
82/82 [==============================] - 8s 82ms/step - loss: 1.1242 - accuracy: 0.3609 - val_loss: 1.0475 - val_accuracy: 0.6321
Epoch 2/5
82/82 [==============================] - 8s 94ms/step - loss: 0.8217 - accuracy: 0.6701 - val_loss: 0.6131 - val_accuracy: 0.7557
Epoch 3/5
82/82 [==============================] - 6s 73ms/step - loss: 0.2629 - accuracy: 0.9258 - val_loss: 0.4482 - val_accuracy: 0.8153
Epoch 4/5
82/82 [==============================] - 7s 86ms/step - loss: 0.0631 - accuracy: 0.9889 - val_loss: 0.4634 - val_accuracy: 0.8107
Epoch 5/5
82/82 [==============================] - 7s 81ms/step - loss: 0.0292 - accuracy: 0.9958 - val_loss: 0.4849 - val_accuracy: 0.8122
```

Рисунок 3.8 – Створення та налаштування DNN

Під час створення та навчання DNN, необхідно було виконати такі кроки:

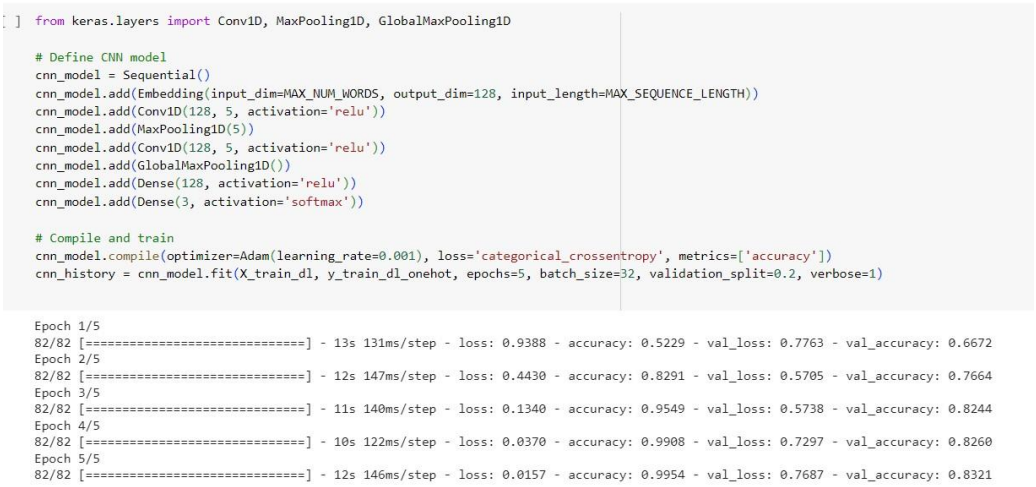
- Імпортування необхідних моделей: Послідовна модель, шари Dense, Embedding, Flatten, Dropout та оптимізатор Adam імпортуються з Keras.

Утиліта to_categorical використовується для одномоментного кодування міток.

- Підготування міток для нейронної мережі: Мітки для навчальних і валідаційних наборів даних перетворюються в однократно кодовану форму для використання з категоріальною функцією крос-ентропійних втрат.
- Визначення архітектури DNN: Послідовна модель ініціалізується для послідовного додавання шарів. Першим додається шар Embedding, який обробляє вхідні послідовності, відображаючи слова у щільні вектори фіксованого розміру. Шар Flatten додається для перетворення двовимірних результатів шару вбудовування в одномірні. Шар Dense зі 128 одиницями та функцією активації ReLU додано для вивчення нелінійних зв'язків. Шар Dropout зі швидкістю 0,5 додано для запобігання надмірному пристосуванню під час навчання. Ще один щільний шар з 3 одиницями і функцією активації softmax використовується як вихідний шар для виведення розподілу ймовірностей за трьома класами.
- Компіляція моделі: Модель скомпільовано за допомогою оптимізатора Adam (зі швидкістю навчання 0.001), функцією втрат categorical_crossentropy та метрикою для моніторингу accuracy.
- Навчання моделі: Модель навчається на тренувальних даних і однократно закодованих мітках, з 5 епохами і розміром партії 32.
- Для моніторингу продуктивності на частині навчальних даних, які не використовуються для оновлення ваг, використовується валідація з коефіцієнтом 0.2.Розгорнутий вивід становлено на 1 для відображення прогресу під час тренування.

Згорткові нейронні мережі (CNN) - це клас глибоких нейронних мереж (рисунок 3.9), які відомі своєю ефективністю в обробці зображень, але вони також дуже ефективні для аналізу тексту. У сфері фінансових новин CNN можуть

обробляти та ідентифікувати ключові текстові шаблони або фрази, які передбачають ринкові настрої та тенденції.



```
[ ] from keras.layers import Conv1D, MaxPooling1D, GlobalMaxPooling1D

    # Define CNN model
    cnn_model = Sequential()
    cnn_model.add(Embedding(input_dim=MAX_NUM_WORDS, output_dim=128, input_length=MAX_SEQUENCE_LENGTH))
    cnn_model.add(Conv1D(128, 5, activation='relu'))
    cnn_model.add(MaxPooling1D(5))
    cnn_model.add(Conv1D(128, 5, activation='relu'))
    cnn_model.add(GlobalMaxPooling1D())
    cnn_model.add(Dense(128, activation='relu'))
    cnn_model.add(Dense(3, activation='softmax'))

    # Compile and train
    cnn_model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    cnn_history = cnn_model.fit(X_train_dl, y_train_dl_onehot, epochs=5, batch_size=32, validation_split=0.2, verbose=1)

    Epoch 1/5
    82/82 [==============================] - 13s 131ms/step - loss: 0.9388 - accuracy: 0.5229 - val_loss: 0.7763 - val_accuracy: 0.6672
    Epoch 2/5
    82/82 [==============================] - 12s 147ms/step - loss: 0.4430 - accuracy: 0.8291 - val_loss: 0.5705 - val_accuracy: 0.7664
    Epoch 3/5
    82/82 [==============================] - 11s 140ms/step - loss: 0.1340 - accuracy: 0.9549 - val_loss: 0.5738 - val_accuracy: 0.8244
    Epoch 4/5
    82/82 [==============================] - 10s 122ms/step - loss: 0.0370 - accuracy: 0.9908 - val_loss: 0.7297 - val_accuracy: 0.8260
    Epoch 5/5
    82/82 [==============================] - 12s 146ms/step - loss: 0.0157 - accuracy: 0.9954 - val_loss: 0.7687 - val_accuracy: 0.8321
```

Рисунок 3.9 – Створення та налаштування моделі CNN

Під час створення та навчання CNN, необхідно було виконати такі кроки:

- Визначення архітектури CNN: Для перетворення вхідних даних у щільні вектори фіксованого розміру використовується шар Embedding. Шар Conv1D з 128 фільтрами та функцією активації relu додається для згортки 1D послідовності. Шар MaxPooling1D використовується для зменшення розмірності вхідних даних, що робить виявлення ознак інваріантним до змін масштабу та орієнтації. Ще один шар Conv1D додається, як правило, для розпізнавання ознак вищого рівня. Шар GlobalMaxPooling1D додається для зменшення дискретизації представлення вхідних даних, беручи максимальне значення за часовим виміром, що корисно для підвищення стійкості моделі до розташування об'єктів. Щільний шар зі 128 одиницями та функцією активації relu використовується для класифікації. Останній щільний шар з 3 одиницями та функцією активації softmax діє як вихідний шар, надаючи ймовірності для кожного з трьох потенційних класів.

- Компіляція моделі: Модель скомпільовано за допомогою оптимізатора Adam зі швидкістю навчання 0.001. В якості функції втрат обрано категоріальну кросентропію, яка підходить для задач багатокласової класифікації. Метрику точності задано для моніторингу роботи моделі.
- Навчання моделі: Для навчання моделі на навчальних даних (X_train_dl) та навчальних мітках (y_train_dl_onehot) використовується метод припасування. Навчання триває 5 епох з розміром партії 32. Використовується валідація з коефіцієнтом 0.2, тобто 20% навчальних даних використовується як валідаційний набір для моніторингу продуктивності моделі та запобігання перенастроюванню. Параметр verbose дорівнює 1, щоб показувати індикатор прогресу під час навчання.

Мережі з довготривалою короткочасною пам'яттю (LSTM) - це особливий тип нейронних мереж (рисунок 3.10), здатних навчатися довготривалим залежностям. Вони особливо добре розуміють контекст фінансових новин, який може охоплювати речення або навіть абзаци.

```
from keras.layers import LSTM

# Define LSTM model
lstm_model = Sequential()
lstm_model.add(Embedding(input_dim=MAX_NUM_WORDS, output_dim=128, input_length=MAX_SEQUENCE_LENGTH))
lstm_model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
lstm_model.add(Dense(3, activation='softmax'))

# Compile and train
lstm_model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
lstm_history = lstm_model.fit(X_train_dl, y_train_dl_onehot, epochs=5, batch_size=32, validation_split=0.2, verbose=1)
```

```
Epoch 1/5
82/82 [==============================] - 76s 876ms/step - loss: 0.9290 - accuracy: 0.5554 - val_loss: 0.7298 - val_accuracy: 0.6687
Epoch 2/5
82/82 [==============================] - 69s 846ms/step - loss: 0.4086 - accuracy: 0.8486 - val_loss: 0.5151 - val_accuracy: 0.7893
Epoch 3/5
82/82 [==============================] - 69s 849ms/step - loss: 0.1695 - accuracy: 0.9423 - val_loss: 0.5247 - val_accuracy: 0.8168
Epoch 4/5
82/82 [==============================] - 70s 849ms/step - loss: 0.0797 - accuracy: 0.9774 - val_loss: 0.5788 - val_accuracy: 0.8015
Epoch 5/5
82/82 [==============================] - 71s 866ms/step - loss: 0.0513 - accuracy: 0.9832 - val_loss: 0.6500 - val_accuracy: 0.8366
```

Рисунок 3.10 – Створення та налаштування моделі LSTM

Під час створення та навчання LSTM, необхідно було виконати такі кроки:

- Визначення LSTM-моделі: Створюється екземпляр моделі Sequential (Послідовна), щоб дозволити додавання шарів у послідовності. До моделі додається шар Embedding для перетворення категоріальних даних, закодованих цілими числами, у щільні вектори фіксованого розміру (output_dim=128). Цей рівень очікує послідовності однакової довжини (input_length=MAX_SEQUENCE_LENGTH), де MAX_SEQUENCE_LENGTH - попередньо визначена константа, що представляє довжину вхідних послідовностей. Додається шар LSTM з 128 одиниць з коефіцієнтом відсіву 0.2, щоб зменшити перенавчання шляхом ігнорування випадково вибраних нейронів під час навчання. Параметр recurrent_dropout використовується для відсіювання зв'язків між рекурентними одиницями. Щільний вихідний шар з 3 одиницями та функцією активації softmax використовується для забезпечення розподілу ймовірностей за трьома класами для класифікації.
- Компіляція моделі: LSTM-модель скомпільовано за допомогою оптимізатора Adam (швидкість навчання 0.001), який є алгоритмом градієнтної оптимізації стохастичних цільових функцій. Параметр categorical_crossentropy вказано як функцію втрат, оскільки вона добре підходить для задач багатокласової класифікації, де мітки надаються в однозначному кодованому форматі. точність включено до списку метрик для оцінювання під час навчання та валідації моделі.
- Навчання моделі: Метод підгонки викликається для навчання моделі з навчальним набором даних (X_train_dl), відповідними одномоментними кодованими мітками (y_train_dl_onehot), вказаною кількістю епох (5), розміром партії (32) та часткою навчальних даних, які будуть використані як валідаційні дані (0.2 або 20%). Параметр verbose=1 вмикає індикатор прогресу під час навчання для відображення статусу кожної епохи.

Двонаправлена LSTM (BiLSTM) розширює можливості LSTM, обробляючи дані як у прямому, так і в зворотному напрямку (рисунок 3.11). Це гарантує, що кожна точка в послідовності отримує інформацію з точок до і після неї, пропонуючи більш повне розуміння тексту.

```
from keras.models import Sequential
from keras.layers import Dense, Embedding, Flatten, Dropout
from keras.optimizers import Adam
from keras.utils import to_categorical

# Convert labels to one-hot encoding
y_train_dl_onehot = to_categorical(y_train_dl)
y_test_dl_onehot = to_categorical(y_test_dl)

# Define DNN model
dnn_model = Sequential()
dnn_model.add(Embedding(input_dim=MAX_NUM_WORDS, output_dim=128, input_length=MAX_SEQUENCE_LENGTH))
dnn_model.add(Flatten())
dnn_model.add(Dense(128, activation='relu'))
dnn_model.add(Dropout(0.5))
dnn_model.add(Dense(3, activation='softmax'))

# Compile the model
dnn_model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
dnn_history = dnn_model.fit(X_train_dl, y_train_dl_onehot, epochs=5, batch_size=32, validation_split=0.2, verbose=1)
```

```
Epoch 1/5
82/82 [==============================] - 8s 82ms/step - loss: 1.1242 - accuracy: 0.3609 - val_loss: 1.0475 - val_accuracy: 0.6321
Epoch 2/5
82/82 [==============================] - 8s 94ms/step - loss: 0.8217 - accuracy: 0.6701 - val_loss: 0.6131 - val_accuracy: 0.7557
Epoch 3/5
82/82 [==============================] - 6s 73ms/step - loss: 0.2629 - accuracy: 0.9258 - val_loss: 0.4482 - val_accuracy: 0.8153
Epoch 4/5
82/82 [==============================] - 7s 86ms/step - loss: 0.0631 - accuracy: 0.9889 - val_loss: 0.4634 - val_accuracy: 0.8107
Epoch 5/5
82/82 [==============================] - 7s 81ms/step - loss: 0.0292 - accuracy: 0.9958 - val_loss: 0.4849 - val_accuracy: 0.8122
```

Рисунок 3.11 – Створення та налаштування моделі BiLSTM

Під час створення та налаштування BiLSTM, необхідно було виконати такі кроки:

- Перетворення міток в однократне кодування: Мітки для навчальних і валідаційних наборів даних конвертуються в формат однократного кодування, щоб підготувати їх до використання з категоріальною функцією втрат крос-ентропії під час навчання.
- Визначення моделі BiLSTM: Послідовну модель визначено для того, щоб шари можна було додавати послідовно. Спочатку додається шар вбудовування для відображення цілочисельної лексики у вектори високої розмірності. Двонаправлене обгортання застосовується до шару LSTM, що дозволяє мережі мати як пряму, так і зворотну

інформацію про послідовність на кожному часовому кроці. Після двонаправленого LSTM можна додати додаткові шари, такі як щільні (повністю з'єднані) шари та шари, що відсікаються, для подальшого навчання та регуляризації. Останнім шаром є шар Dense з функцією активації softmax, який забезпечує розподіл ймовірностей по вихідних класах.

- Компіляція моделі: Модель BiLSTM компілюється за допомогою оптимізатора Adam, який є загальним вибором для навчання моделей глибокого навчання завдяки його ефективним обчисленням і низьким вимогам до пам'яті, зі швидкістю навчання, яка контролює, наскільки вагові коефіцієнти моделі оновлюються під час навчання. Функція втрат має значення categorical_crossentropy, оскільки це задача багатокласової класифікації. Продуктивність моделі моніторилася на основі точності.
- Навчання моделі: Модель навчається за допомогою функції пристосування з навчальними даними та однократно закодованими мітками.
- Параметри навчання включають кількість епох (ітерацій над усім набором даних), розмір партії (кількість вибірок на одне оновлення градієнта) та валідаційний спліт (частка даних, що використовується для валідації). Параметр verbose=1 дозволяє роздруковувати журнали прогресу під час навчання.

Ці нейромережеві моделі пристосовані для розпізнавання складної та послідовної мови фінансових новин, що дозволяє їм передбачати ринкові настрої та тенденції з високим ступенем точності.

Висновки до розділу 3

У цьому розділі проведено створення та навчання моделей для класифікації контенту фінансових новин. За датасет було обрано Sentiment Analysis for Financial News з платформи машинного навчання Kaggle.

Процес починався з огляду набору даних, щоб врахувати особливості набору даних. Після цього виконав попередню обробку даних, яка включає токенізацію для розбиття тексту на менші одиниці, видалення стоп-слів для усунення поширених, але непотрібних слів, стеммінг приведення слів до їх кореневої форми і лематизацію для більш контекстно-орієнтованої конденсації слів.

Відбір ознак передбачав вибір атрибутів, які сприяють результатам прогнозування. Розглядолося два методи: модель Bag-of-Words, яка представляє текстові дані у вигляді набору слів, і статистичний показник TF-IDF, який оцінює, наскільки слово релевантне документу в корпусі. Застосовано векторизацію текстових прикладів датасету методом виділення ознак TF-IDF.

Перед навчанням моделей виконав розділ отриманного набору даних на тренувальну та тестову частину з метою проведення обчислювальних експериментів з метриками оцінювання моделей.

Створення моделей класифікації охоплює моделі керованого навчання, такі як логістична регресія, SVM, наївний Байєс і випадковий ліс, які покладаються на марковані набори даних для навчання і прогнозування. Нейронні мережі, такі як DNN, RNN, LSTM і Bidirectional LSTM, використовуються завдяки їхній здатності обробляти послідовні та часові дані, що притаманно мові.

4 ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ

У даному розділі проведено обчислювальні експерименти з метриками оцінювання моделей. Було розроблено та натреновано набір моделей машинного навчання, основна мета яких розпізнавати характер змісту фінансових новин. Порівняльний аналіз в оцінці моделей має вирішальне значення для розуміння можливостей різних підходів. Він передбачає порівняння різних моделей на одному і тому ж тестовому наборі даних, щоб визначити, яка результативність відповідно до конкретних показників, таких як точність, влучність, повнота та оцінка F1. В аналізі фінансових текстів це може означати оцінку того, наскільки добре кожна модель класифікує контент фінансових новин. Оцінюючи моделі між собою, можна визначити, яка з них є найефективнішою для нюансів фінансових даних, що призводить до прийняття більш обґрунтованих рішень і стратегій.

4.1 Постановка задачі обчислювальних експериментів

Для проведення обчислювальних експериментів з датасету Sentiment Analysis for Financial News в загальному з 4089 прикладів обрано вибірку з 2617 ознак для навчання моделей, 654 – валідації під час налуштування, 818 – для оцінювання.

В експериментах використовувалися моделі таких моделей штучного інтелекту: Logistic regression, Random Forest, SVM, DNN, CNN, LSTM та BiLSTM.

В кожному з експериментів подаволося 818 прикладів та формувався масив відповідей – відповідних класів настроїв.

За цим масивом класів оцінювалася точність поточної моделі на основі метрики оцінювання моделей.

Метрики оцінювання моделей - це кількісні інструменти, що використовуються для оцінювання ефективності моделей класифікації. Ці метрики дають уявлення про різні аспекти передбачень моделі.

Точність (Accuracy) вимірює частку прогнозів, які наша модель зробила правильно, тобто кількість правильних прогнозів, поділену на загальну кількість прогнозів.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

Де TP – істинно позитивні, FP – хибно позитивні, TN – істинно негативні, FN – хибно негативні випадки класифікації тестового набору даних.

Влучність (Precision) оцінює якість позитивних прогнозів, зроблених моделлю. Це кількість правильних позитивних прогнозів, поділена на загальну кількість позитивних прогнозів.

$$Precision = \frac{TP}{TP+FP} \quad (4.2)$$

Повнота (Recall), або чутливість, вимірює здатність моделі правильно ідентифікувати всі релевантні випадки і розраховується як кількість вірних позитивних прогнозів, поділена на кількість фактичних позитивних випадків.

$$Recall = \frac{TP}{TP+FN} \quad (4.3)$$

Оцінка F1 забезпечує баланс між влучністю та повнотою, фіксуючи середнє гармонійне значення цих двох метрик. Це особливо корисно, коли розподіл класів нерівномірний.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4.4)$$

Ці показники є ключовими у визначенні ефективності моделі, особливо в таких галузях, як аналіз фінансових новин, де ціна хибнопозитивного або хибнонегативного результату може бути значною.

4.2 Результати обчислювальних експериментів

У загальному аналізі моделей для класифікації фінансових новин було оцінено ефективність різних архітектур, таких як логістична регресія, випадковий ліс, SVM, наївний Байєс та нейромережеві моделі, такі як DNN, CNN, LSTM та BiLSTM. Кожна модель представляє унікальний набір сильних сторін з точки зору часу навчання і прогнозування, точності, влучності, повноти і оцінки F1.

На рисунку 4.1 зображено розділ даних для обчислювального експерименту моделей машинного навчанння.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

X_train.shape, X_test.shape
```

```
((3271, 5000), (818, 5000))
```

Рисунок 4.1 – Розділ набору даних моделей машинного навчання

Тоді як, на рисунку 4.2 зображено етап розділу даних у випадку нейронних мереж.

```
# Split data into training and testing sets
X_train_dl, X_test_dl, y_train_dl, y_test_dl = train_test_split(X_dl, y_dl, test_size=0.2, random_state=42, stratify=y_dl)

X_train_dl.shape, X_test_dl.shape
```

```
((3271, 250), (818, 250))
```

Рисунок 4.2 – Розділ набору даних моделей нейронних мереж

Дані розділяються на навчальний (80%) і тестовий (20%) набори. Стратифікація гарантує однаковий розподіл класів у розбитті.

Показник F1 особливо корисний, коли вартість хибнопозитивних і хибнонегативних результатів приблизно однакова, або коли ви маєте нерівномірний розподіл класів. Якщо один клас значно недопредставлений, оцінка F1 може дати краще уявлення про якість моделі, ніж лише точність.

У таблиці 4.1 наведено результати обчислення метрик оцінювання моделей за оцінкою F1.

Таблиця 4.1 – Результати метрик оцінювання моделей за оцінкою F1

Модель	Оцінка F1
Логістична регресія (Logistic Regression)	0,824436
Випадковий ліс (Random Forest)	**0,849284**
Метод опорних векторів (SVM)	0,825217
Наївний Байєсівський класифікатор (Naïve Bayes)	0,774799
Глибинна нейронна мережа (DNN)	0,8115896327934093
Згорткова нейронна мережа (CNN)	0,8188956719875496
Мережа довгої короткочасної пам'яті (LSTM)	0,8209197834967876
Двонаправлена LSTM (BiLSTM)	**0,8257299720888208**

Згідно результатів, що моделі на основі випадкового лісу та нейронних мереж, особливо двонаправленої LSTM, демонструють багатообіцяючі результати, завдяки своїй здатності обробляти послідовні дані та вловлювати часові залежності в тексті. Ці моделі, як правило, досягають кращого балансу між точністю та запам'ятовуванням, що свідчить про їхню ефективність у класифікації фінансових новин з вищою надійністю.

Найефективнішими підходами є ті, що можуть точно вловити нюанси та настрої у фінансових текстах, які мають вирішальне значення для прогнозування ринкових рухів [31]. Вибір моделі може залежати від конкретних вимог завдання,

включаючи компроміс між швидкістю і точністю, складністю фінансового тексту і доступними обчислювальними ресурсами.

Висновки до розділу 4

У даному розділі проведено обчислювальні експерименти з метриками оцінювання моделей. Було розроблено та натреновано набір моделей машинного навчання, основна мета яких розпізнавати характер змісту фінансових новин. Виконано порівняльний аналіз в оцінці моделей, бо це має вирішальне значення для розуміння можливостей різних підходів.

На основі обраних архітектурних рішень було створенно надійні моделі з точністю яка складає не менше 80%. Це є підтвердженням здатності алгоритмів машинного навчання до класифікації контенту фінансових новин.

Визначено, що оптимальною за точністю серед моделей машинного навчання є Random Forest, а – серед нейронних мереж – Bidirectional Long-Short Memory.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

5.1 Опис проекту

Використовуючи дипломний проект, як основу, та створену рекурентну мережу типу моделы двонаправленої довготривалої короткочасної пам'яті (BiLSTM), можна реалізувати додаток та API для класифікації контенту фінансових новин, які можуть бути використані як розробниками, так і користувачами після отримання ключа.

У цьому розділі було розглянуто створення стартап-проекту на основі BiLSTM. Опис ідеї стартап-проекту наведений у таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Додаток та API для класифікації контенту фінансових новин	Отримування швидкого розуміння ринкових настроїв	Сприяє прийняттю більш обґрунтованих рішень і потенційно підвищує точність прогнозів інвестора щодо руху ринку
	Співправця з фінтех компаніями	Інтеграція API в існуючі платформи для покращення їхніх послуг, наприклад, можуть запропонувати користувачам аналіз настроїв як додаткову функцію.

Для реалізації ідеї проекту була обрана наступна технологія: побудова програми на мові Python, фреймворку для моделі TensorFlow з використанням бібліотеки keras.

5.2 Технологічний аудит ідеї проекту

Технологічна аудит ідеї проекту наведений у таблиці 5.2.

Таблиця 5.2 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1.	Додаток та АПІ для класифікації контенту фінансових новин	Мова програмування Python	Є в наявності	Доступні безкоштовно
2.		Фреймворк TensorFlow	Є в наявності	Доступні безкоштовно
3.		Бібліотека keras	Є в наявності	Доступні безкоштовно

Для реалізації ідеї проекту було обрано мову програмування Python, фреймворк TensorFlow та бібліотеку Keras. Всі вищеназвані технології є в наявності та доступними.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можуть бути використані під час виведення проекту на ринок, та ринкових загроз, які можуть перешкоджати реалізації проекту на ринок, дозволяє спланувати напрями розвитку проекту з урахуванням стану ринкового середовища, потреб потенційних клієнтів та

пропозицій конкуруючих проектів. У таблиці 5.3 наведено попередню характеристику ринку для розроблюваного стартап-проекту.

Таблиця 5.3 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1.	Кількість головних гравців, од	Не більше 5
2.	Динаміка ринку (якісна оцінка)	Попит на аналітику в реальному часі у фінансовій сфері зростає через стрімку діджиталізацію ринку та збільшення швидкості обміну інформацією.
3.	Наявність обмежень для входу (вказати характер обмежень)	Значна кількість конкурентів серед розробників рішень з використанням штучного інтелекту для сфери аналізу фінансових даних
4.	Специфічні вимоги до стандартизації та сертифікації	Специфічних вимог до стандартизації та сертифікації немає

У таблиці 5.4 наведено характеристику потенційних клієнтів стартап-проекту.

Таблиця 5.4 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Автоматичні торгові сигнали	Алгоритмічні трейдери	Надають перевагу високочастотним оновленням	Надійні сигнали, налаштована частота
2.	Аналіз настроїв для інвестицій	Хедж-фонди, Інвестиційні банки	Потребують інтеграції з існуючими системами	Глибокий аналіз настроїв, інтеграція з API

У таблиці 5.5 наведено фактори загроз для розроблюваного стартап-проекту.

Таблиця 5.5 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Складність виходу моделі на міжнародний ринок	Всі потенційні конкуренти розташовані закордоном. Важкий вихід	Поступово розвиватися на міжнародні арені починаючи розповсюджуватися з найближчих країн.

Кінець таблиці 5.5

2.	Довіра у використанні моделі	Упереджене відношення до використання моделі для додаткових кроків перед прийняттям рішень	Зменшення попиту, необхідні додаткові стимули для довіри у використанні моделі.

У таблиці 5.6 наведено фактори можливостей для розроблюваного стартап-проекту.

Таблиця 5.6 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Залучення інвестицій	Склавши чіткий бізнес-план, можна залучити інвесторів, що дозволить розширити штат та закупити потрібну техніку	Скорочення часу для виходу на міжнародний ринок та покращення системи
2.	Розширення клієнтської бази	Реалізувати масштабну рекламну компанію щоб залучити якомога більше клієнтів	Спланувати рекламну кампанію, проводити презентації на зборах і виставках та семінари на тему BiLSTM, в яких буде демонстрація продукту

Кінець таблиці 5.6

3.	Збільшення клієнтів-підприємців	Складання договорів для подальшого співробітництво з фінтех компаніями	Розширення бази даних інвесторів

У таблиці 5.7 наведено ступеневий аналіз конкуренції на ринку для розроблюваного стартап-проекту.

Таблиця 5.7 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкуренто- спроможною)
Складність старту	На ринку уже є досить успішні конкуренти	Якісна рекламна кампанія та покращення функціоналу
Складність входу на міжнародний ринок	Немає досвіду такого товару на ринку, значні фінансові затрати для створення початкових умов	Проаналізувати необхідну складову базу та створити найбільш сприятливих умов для клієнта

У таблиці 5.8 наведено аналіз конкуренції в галузі за М. Портером для розроблюваного стартап-проекту.

Таблиця 5.8 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти в галузі	Постачальники	Клієнти	Товари-замінники
	фактично прямих конкурентів нема	Brand24	-	Компанії з брендом	Нові стартап-ідеї
Висновки:	Можлива велика кількість різноманітних рішень у конкурентів	-	Постачальники орієнтуються на потреби клієнтів	Компанії з брендом вже можуть співпрацювати з клієнтами	Ентузіасти можуть вийти на ринок с кращим функціоналом

У таблиці 5.9 наведено обґрунтування факторів конкурентоспроможності для розроблюваного стартап-проекту.

Таблиця 5.9 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)

Кінець таблиці 5.9

1.	Досить висока точність	Оскільки проект знаходиться на етапі розробки, можна спиратися на точність товарів-замінників, щоб при виході товару на ринок мати перевагу у точності передбачування
2.	Унікальність надання послуг	Користувач має можливість отримання функцій таких послуг, які на сьогодні отримуються в основному за особистим втручанням
3.	Швидке реагування на відгуки	На етапі розробки неможливо передбачити всі нюанси застосування. Завдяки забезпеченню швидкого відгуку, можна використовувати користувачів у якості бета-тестувальників, що дозволить максимально швидко адаптувати додаток під потреби реальних клієнтів

У таблиці 5.10 наведено порівняльний аналіз сильних та слабких сторін ринку для розроблюваного стартап-проекту.

Таблиця 5.10 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з … (наша підприємства)						
			–3	–2	–1	0	+1	+2	+3

Кінець таблиці 5.10

1.	Точність класифікації	15					+		
2.	Швидкість обробки	13				+			
3.	Підтримка продукту	10				+			
4.	Вдосконалення продукту через реакцію користувачів на бету версію.	11					+		

У таблиці 5.11 наведено SWOT - аналіз стартап-проекту.

Таблиця 5.11 – SWOT - аналіз стартап-проекту

Сильні сторони: Надійність Швидке реагування на побажання клієнтів Легкість експлуатування	Слабкі сторони: Потрібні інвестиції для старту Новий метод, ще не визнаний у світі.
Можливості: Вихід на міжнародний ринок Збільшення функціоналу та покращення моделі	Загрози: Важкість виходу на міжнародний ринок Нестабільна ситуація в країні Поява нового продукту Складність знайти інвестора

У таблиці 5.12 наведено альтернативи ринкового впровадження стартап-проекту.

Таблиця 5.12 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Активна реклама у всіх ресурсах	Висока. Потужна реклама допоможе розповсюдитися серед значної кількості користувачів	Після старту
2.	Вихід на міжнародній рівень	Середній, оскільки більша кількість конкурентів та потреба у відкриті офісів для потреб на міжнародній арені	Через півтора року
3.	Участь в презентаціях перед інвесторами	Середня, складно передбачити потребу у співпраці	Через приблизно пів року, в залежності від зацікавлення продуктом інвесторами

5.4 Розроблення ринкової стратегії проекту

У таблиці 5.13 наведено вибір цільових груп потенційних споживачів для розроблюваного стартап-проекту.

Таблиця 5.13 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Трейдери	Користувачі готові прийняти продукт	Продукт необхідний уже	Конкуренція середня	Вхід помірно-складний
2.	Приватні компанії	Не всі користувачі готові прийняти	Продукт необхідний не всім	Висока конкуренція	Складний вхід
3.	Навчальні та дослідницькі програми	Користувачі готові прийняти продукт	Не всі зацікавлені в продукті	Конкуренція середня	Вхід помірно-складний
Які цільові групи обрано: Потрібно працювати зі всіма цільовими групами, перевагу надати послугам, що найбільш популярні для клієнтів.					

У таблиці 5.14 наведено визначення базової стратегії розвитку для стартап-проекту.

Таблиця 5.14 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкуренто спроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Розвиток шляхом потужної маркетинговій компанії та демонстрації продукту на відкритих конференціях	Реклама та охоплення цільової аудиторії на конференціях допоможе знайти клієнтів	Виступ на конференціях, вільне розповсюдження для навчально-дослідницьких цілей	Прислухатися до побажань клієнтів та оперативно оновлювати продукт під їх стандарти

У таблиці 5.15 наведено визначення базової стратегії конкурентної поведінки для стартап-проекту.

Таблиця 5.15 – Визначення базової стратегії конкурентної поведінки

№	Чи є проект «пер-шопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки

Кінець таблиці 5.15

1.	Ні, існують аналоги	Компанії необхідно буде шукати клієнтів які найбільш зацікавленні у продукті	Через схожість функціоналу, компанія буде копіювати та намагатися покращити основні характеристики у конкурентів	Пропонувати схожий функціонал на кращих умовах, оперативно оновлювати функціонал відповідно до нових стандартів

У таблиці 5.16 наведено визначення стратегії позиціонування для стартап-проекту.

Таблиця 5.16 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Модель повинна мати високу точністьта бути надійною	Залучати все більше нових клієнтів для розширення бази даних та розширятися на міжнародній арені	Оперативне виправлення недоліків та впровадження побажань	Зручна для користувачів, точний у прогнозах

5.5 Розроблення маркетингової програми стартап-проекту

У таблиці 5.17 наведено визначення ключових переваг концепції потенційного товару щодо стартап-проекту.

Таблиця 5.17 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1.	Моделювання мережі високої точності архітектури BiLSTM	Високоточність передбачування	Можливість отримувати чітке точне передбачення
2.	Часте оновлення програми	Врахування побажань клієнтів та виправлення помилок в найкоротші терміни	Створення відділу підтримки для аналізування реагування на відгуки клієнтів
3.	Доступ до API створенної моделі	Можливість використання програми для власних розробок	Відкритий АПІ для використання розробниками

У таблиці 5.18 наведено опис трьох рівнів моделі товару для стартап-проекту.

Таблиця 5.18 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові	
I. Товар за задумом	Програма для передбачення наявності пухлини у пацієнта	
II. Товар у реальному виконанні	Властивості/характеристики	Розмір
	Модуль- BiLSTM моделі	50МБ
	Веб-сторінка для класифікації контенту фінансових новин	60МБ
	Модуль обробки запитів	30МБ
	Якість: логування збоїв та система зворотнього зв’язку для повідомлення про неналежну роботу програми	
	Пакування: продаж електронних ключів-ліцензій	
	Марка: назва організації-розробника + назва товару	

Кінець таблиці 5.18

III. Товар із підкріпленням	До продажу: програмний код
	Після продажу: АПІ/електронна версія додатку з ключем
За рахунок чого потенційний товар буде захищено від копіювання: прив’язка копії програмного продукту до конкретного користувача та активація програми шляхом введення ліцензійного ключа, або шляхом надання послуг без передачі програмного продукту замовнику	

У таблиці 5.19 наведено визначення меж встановлення ціни для продукту.

Таблиця 5.19 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі становлення ціни на товар/послугу
1.	0грн - 100грн	0грн - 300грн	>100000грн/місяць	0грн - 10000грн

У таблиці 5.20 наведено формування системи збуту для продукту.

Таблиця 5.20 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Придбання ліцензії на продукт або договір про надання послуг без передачі ПО до замовника	Розміщення Ключа для доступу до АПІ на веб-сервісі	Канал збуту одноріневий (через розміщення ключів на веб-сервісах)	Вертикальна (право власності залишається у розробника ПЗ)

У таблиці 5.21 наведено концепція маркетингових комунікацій для продукту.

Таблиця 5.21 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Купують товар на вимогу	Тематичні зустрічі, конференції, презентації.	Надання можливості використання мережі	Демонстрація основних функцій продукту	Охоплення аудиторії, пояснення функцій

Висновки до розділу 5

У цьому розділі обговорювалося створення стартап-проекту для класифікації контенту фінансових новин з використанням рекурентної мережі, а саме двонаправленої довготривалої короткочасної пам'яті (BiLSTM). Було представлено концепцію проекту, проаналізовано конкурентів, а також перспективи продукту зайняти свою нішу на міжнародній арені. Розроблено загальний опис концепції проекту для виходу на ринок. Були визначені потенційні можливості стартапу, виявлені основні конкуренти, а також проведено порівняння кращих характеристик конкурентів для визначення переваг стартапу, що розробляється, над обраними конкурентами. Проведено технічний аудит проекту і визначено можливості реалізації програмного продукту.

Проаналізовано ринок у пошуках вигідної можливості для створення фірми. У порівнянні з конкурентами визначено перспективи запуску та розвитку проекту, визначено план та стратегію розвитку продукту, визначено цільові групи споживачів, розроблено засоби просування проекту. При виході проекту на міжнародну арену ключовою метою є запуск масштабної рекламної кампанії для залучення якомога більшої кількості клієнтів.

Таким чином, було побудувано стартап-проект для просування створеного програмного продукту на ринок, навчився формувати стартап-проекти, будувати маркетинговий план та аналізувати обраний ринок.

ВИСНОВКИ

1. Проведено аналіз методів класифікації текстових повідомлень. Запропоновано використання таких моделей штучного інтелекту: Logistic regression, Random Forest, SVM, DNN, CNN, LSTM та BiLSTM. Обґрунтовано викорикистання класифікації емоційних настроїв для аналізу фінансових новин.
2. Представлено обрані алгоритми машинного навчання. Проведено аналіз інструментів розробки моделей навчання. Для реалізації обрано: мову програмування Python та її ключові бібліотеки: Scikit-Learn, Keras, Pandas, NumPy, Matplotlib, NLTK.
3. Проведено векторизацію текстових прикладів датасету Sentiment Analysis for Financial News з платформи машинного навчання Kaggle за такими етапами: токенізація, стеммінг та лематизація, а також застосовано методи виділення ознак TF-IDF.
4. Проведено обчислювальні експерименти на обраних моделях для класифікації контенту фінансових новин. Визначено, що оптимальною за метрикою точності F1 серед моделей машинного навчання є Random Forest, а – серед нейронних мереж – Bidirectional Long-Short Memory.
5. Розроблено стартап-проект, який містить: технологічний аудит ідеї, аналіз ринкових можливостей, ринкової стратегії, маркетингову програму застосування моделі BiLSTM.

Результати досліджень апробовані на XI International Scientific and Practical Conference «Integration of science as a mechanism of effective development», November 28 – December 01, 2023, Helsinki, Finland.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. BASILONE, Ryan. The impact of news on stock market investors. Available at SSRN 3908662, 2021.
2. SANTOSHI, K.; ARCHANA, U.; PRIYANKA, D. Text Classification Using Machine Learning Techniques. Specialusis Ugdymas, 1.43: 8108-8117, 2022.
3. SAXENA, Anshul, et al. Classification of Financial News Articles Using Machine Learning Algorithms. In: Data Science and Security: Proceedings of IDSCS 2020. Springer Singapore, pp. 193-199, 2021.
4. Gurmeet Kaur and Karan Bajaj, "News classification and its techniques: a review", IOSR Journal of Computer Engineering (IOSR-JCE), vol. 18, no. 1, pp. 22-26, 2016.
5. R. Ravi Kumar, M. Babu Reddy, and P. Praveen, "Text classification performance analysis on machine learning", International Journal of Advanced Science and Technology, vol. 28, no. 20, pp. 691-697, 2019.
6. SINGH, Yash Veer, et al. News classification system using machine learning approach. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). IEEE, p. 186-188, 2021.
7. RECHENTHIN, Michael David. Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction. The University of Iowa, 2014.
8. Linear Regression, Logistic Regression, and SVM in 10 Minutes. URL: https://towardsdatascience.com/understanding-3-classical-machine-learning-models-once-and-for-all-part-1-32a1ac52c0fd (дата звернення: 28.10.2023).
9. Logistic Regression for Machine Learning. URL: https://machinelearningmastery.com/logistic-regression-for-machine-learning/ (дата звернення: 28.10.2023).

10. Hosmer, David W., Lemeshow, Stanley. Applied Logistic Regression (2nd ed.). Wiley, 2000.
11. Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.
12. Support Vector Machine (SVM) Algorithm. URL: https://www.geeksforgeeks.org/support-vector-machine-algorithm/ (дата звернення: 28.10.2023).
13. Leo Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.
14. Optimise Your Business with Machine Learning: How Random Forest models can help you make predictions. URL: https://medium.com/@DataCubed/optimise-your-business-with-machine-learning-how-random-forest-models-can-help-you-make-be68ee78180 (дата звернення: 28.10.2023).
15. Bootstrap Aggregation, Random Forests and Boosted Trees. URL: https://www.quantstart.com/articles/bootstrap-aggregation-random-forests-and-boosted-trees/ (дата звернення: 28.10.2023).
16. Multinomial Naïve Bayes' For Documents Classification and Natural Language Processing (NLP). URL: https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6 (дата звернення: 29.10.2023).
17. Classifying data using the Multinomial Naive Bayes algorithm. URL: https://developer.ibm.com/tutorials/awb-classifying-data-multinomial-naive-bayes-algorithm/ (дата звернення: 29.10.2023).
18. SARKER, Iqbal H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2.6: 420. 2021.
19. FATHI, Ehsan; SHOJA, Babak Maleki. Deep neural networks for natural language processing. In: Handbook of Statistics. Elsevier, p. 229-316. 2018.
20. Deep Learning for NLP: An Overview of Recent Trends. URL: https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d (дата звернення: 29.10.2023).

21. Understanding LSTM Networks. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (дата звернення: 29.10.2023).
22. Python 3.12.1 documentation. URL: https://docs.python.org/3/ (дата звернення: 30.10.2023).
23. Scikit-learn documentation. URL: https://scikit-learn.org/stable/index.html (дата звернення: 30.10.2023).
24. Keras 3 API documentation. URL: https://keras.io/api/ (дата звернення: 30.10.2023).
25. Pandas documentation. URL: https://pandas.pydata.org/docs/ (дата звернення: 30.10.2023).
26. NumPy documentation. URL: https://numpy.org/doc/ (дата звернення: 30.10.2023).
27. Using Matplotlib. URL: https://matplotlib.org/stable/contents.html (дата звернення: 30.10.2023).
28. NLTK documentation. URL: https://www.nltk.org/ (дата звернення: 30.10.2023).
29. Sentiment Analysis for Financial News. URL: https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news/data (дата звернення: 29.10.2023).
30. How to Build an End-To-End ML Pipeline. URL: https://neptune.ai/blog/building-end-to-end-ml-pipeline (дата звернення: 29.10.2023).
31. Ali-Shakh Imran, Shapovalova Svitlana. The classification of financial news content using machine learning. Integration of science as a mechanism of effective development: Proceedings of the XI International Scientific and Practical Conference. Helsinki, Finland. November 28 – December 1, P. 376-377. 2023.

Додаток А

Класифікація контенту фінансових новин на основі машинного навчання

Публікації за темою роботи

Аркушів 4

INTEGRATION OF SCIENCE AS A MECHANISM OF EFFECTIVE DEVELOPMENT

Proceedings of the XI International Scientific and Practical Conference

Helsinki, Finland
November 28 - December 01, 2023

THE CLASSIFICATION OF FINANCIAL NEWS CONTENT USING MACHINE LEARNING

Ali-Shakh Imran,
Master student
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Shapovalova Svitlana,
Ph.D., Associate Professor
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

The advent of digital media has brought about an overwhelming influx of economic information, drastically impacting inventory marketplace dynamics. The rapid dissemination of news, specifically through social media, has made the project of reading and interpreting this sizeable amount of data increasingly complicated. Financial news, various from corporate income reviews to worldwide economic modifications, performs an essential position in shaping investor sentiment and market trends [1]. The mission lies in efficient processing and knowledge of the nuances and sentiment embedded in this information.

The conventional methods of economic information evaluation are not viable due to the sheer quantity and complexity of the statistics. Moreover, the subtleties of language and sentiment in economic information require greater sophisticated evaluation than what manual strategies can offer. There is a need for automatic structures that cannot only manage huge volumes of information however additionally correctly classify and interpret the sentiment and the consequences of economic news.

The classification of financial news content using machine learning is a change in thinking from traditional analytical methods, leveraging superior computational strategies to automate and decorate the accuracy of news sentiment evaluation. This approach is vital in decoding the sizable and complicated array of financial news data [2].

Machine learning gives a set of algorithms and strategies tailored for processing and classifying massive units of textual information. The core of this process involves changing unstructured text data into a based layout that gadget-mastering fashions can interpret. This is finished via sophisticated document representation techniques, which include Bag-of-Words or TF-IDF (Term Frequency-Inverse Document Frequency), which translate textual content into numerical values or vectors.

Given the great vocabulary inherent in economic information, function choice and dimensionality reduction are vital. Techniques like stopwords elimination, stemming, and lemmatization are employed to refine the feature set, focusing on applicable phrases that convey massive means. This preprocessing step is critical for reducing computational complexity and improving the version's overall performance.

The unique challenge in financial news classification lies in the sector-specific jargon and the rapidly changing nature of news content. Machine learning models must be adaptive and robust, capable of interpreting not just the lexical content but also the underlying sentiment and the implications of the news.

The fulfillment of these device mastering models in classifying financial news is evaluated the usage of metrics including accuracy, precision, and recall. Continuous optimization and fine-tuning of these models are important to cater to the dynamic nature of financial information, ensuring that the classification remains relevant and accurate over time.

This decision is intended to give an overview of the machine learning workflow employed to classify financial news content, as illustrated in Fig.1.

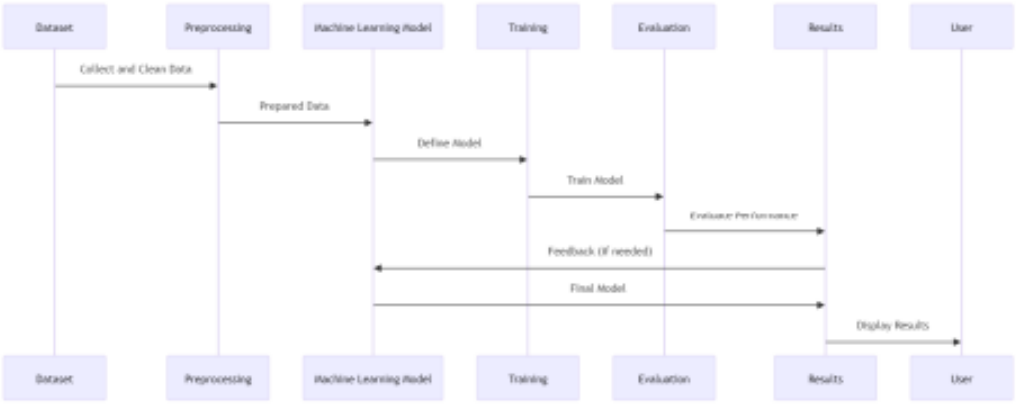


Figure 1. process of classification of financial news content using machine learning.

The integration of this machine learning workflow into the financial analysis ecosystem is both beneficial and necessary. The exponential growth of financial news makes traditional methods of news analysis increasingly unfeasible. Financial news sentiment can be classified accurately and quickly on a scale that manual methods cannot match with the help of machine learning, which offers a scalable and efficient solution.

Using this machine learning workflow, financial institutions can acquire a competitive advantage by harnessing the power of AI to turn vast amounts of data into actionable intelligence. The integration of technology is a significant advancement in financial analytics that provides stakeholders with the tools to navigate the complexity of today's information-rich financial markets.

References:

1. BASILONE, Ryan. The impact of news on stock market investors. *Available at SSRN 3908662, 2021.*

2. SANTOSHI, K.; ARCHANA, U.; PRIYANKA, D. Text Classification Using Machine Learning Techniques. *Specialusis Ugdymas*, 2022, 1.43: 8108-8117.