

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ АНАЛІЗУ
ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ПРО РИНОК АКЦІЙ.
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
ANALYZING THE TONE OF STOCK MARKET MESSAGES

Виконав: студент(ка) 2 курсу, групи КНТ-120м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Чорнобель Д.С.

(прізвище та ініціали)

Керівник Льовкін В.М.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Чорнобеля Данііла Сергійовича
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів аналізу тональності текстових повідомлень про ринок акцій. Research and Software Implementation of Methods for Analyzing the Tone of Stock Market Messages
 керівник проєкту (роботи) Льовкін Валерій Миколайович, к.т.н., доцент
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- затверджені наказом закладу вищої освіти від “ 5 ” листопада 2021 року № 435
2. Строк подання студентом проєкту (роботи) 13 грудня 2021 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Основні рішення щодо реалізації компонентів системи. 5. Експлуатація та експериментальне дослідження програми.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Льовкін В.М., доцент		
Нормоконтроль	Камінська Ж.К., асистент		

7. Дата видачі завдання 7 вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Експлуатація та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	10-11 тиждень	Додатки
8	Нормоконтроль та рецензування.	12–13 тижні	
9	Захист роботи.	14 тиждень	

Студент (ка) _____ Чорнобель Д.С.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ Льовкін В.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
122 с., 56 рис., 15 табл., 3 дод., 60 джерел.

НАУКА ПРО ДАНІ, МАШИННЕ НАВЧАННЯ, АНАЛІЗ
ТОНАЛЬНОСТІ, КЛАСИФІКАЦІЯ ТЕКСТІВ, РИНОК АКЦІЙ, ІНВЕСТОР.

Об'єкт дослідження – процес аналізу тональності текстових повідомлень.
Предмет дослідження – методи аналізу тональності текстових повідомлень про ринок акцій. Мета роботи – розроблення вебзастосунку для проведення аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

Матеріали, методи та технічні засоби: функціональне та об'єктно-орієнтоване програмування; мова програмування Python; бібліотеки Pandas, Plotly, FastText, Keras; вебфреймворк Flask; система MongoDB; середовище Visual Studio Code; комп'ютер з процесором AMD Ryzen 5 4600H.

Результати. Досліджено існуючі методи аналізу тональності. Розроблено вебзастосунок, в якому застосовується запропонований метод та за допомогою якого можна проводити аналіз тональності заголовків новин ринку акцій для дослідження ситуації на ринку та обґрунтування інвестиційних рішень. Наукова новизна дипломної роботи полягає у розробленому методі визначення тенденцій зміни цін акцій, який характеризується використанням моделей FastText та LSTM і дозволяє розв'язати задачу категоризації акцій і виконати аналіз тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

Висновки. Розроблено вебзастосунок для проведення аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень, що застосовує запропонований метод на практиці.

Галузь використання – інвестування на ринку акцій.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 122 pages, 56 figures, 15 tables, 3 appendixes, 60 sources.

DATA SCIENCE, MACHINE LEARNING, SENTIMENT ANALYSIS, TEXT CLASSIFICATION, STOCK MARKET, INVESTOR.

Object of the study is the process of text messages sentiment analysis. Subject of the study is the stock market text messages sentiment analysis methods. The purpose of the work is to develop the web application that provides stock market text messages sentiment analysis for investment decisions justification.

Materials, methods, and tools: functional and object-oriented programming; Python programming language; Pandas, Plotly, FastText, Keras libraries; Flask web framework; MongoDB system; Visual Studio Code IDE; computer with AMD Ryzen 5 4600H processor.

Results. The existing sentiment analysis methods are investigated. The web application that uses proposed method and provides stock market text messages sentiment analysis for stock market analysis and investment decisions justification is developed. The scientific novelty of the thesis lies in the developed method of determining stock price trends, which is characterized by the use of FastText and LSTM models and allows to solve the problem of stock categorization and perform stock market text messages sentiment analysis for investment decisions justification.

Conclusions. The web application that provides stock market text messages sentiment analysis for investment decisions justification and uses the proposed method is developed.

The field of use is the stock market investment.

ЗМІСТ

Перелік скорочень та умовних позначок	9
Вступ	10
1 Аналіз проблеми та постановка завдань дослідження	12
1.1 Огляд ключових понять.....	12
1.2 Огляд програмних засобів, що вирішують аналогічні завдання.....	13
1.2.1 Sentdex	13
1.2.2 StockSnips	14
1.2.3 CryptoMood.....	15
1.2.4 Market Prices Forecast.....	17
1.3 Огляд застосування попередньої обробки даних та методів створення вкладень слів	18
1.3.1 Огляд застосування попередньої обробки даних	18
1.3.2 Огляд методів створення вкладень слів.....	19
1.4 Постановка завдань дослідження	23
1.5 Висновки з розділу 1	24
2 Матеріали і методи	25
2.1 Задача класифікації текстових повідомлень про ринок акцій за їх тональностями	26
2.2 Метод аналізу тональності, заснований на словнику	27
2.3 Метод аналізу тональності, заснований на правилах	29
2.4 Методи аналізу тональності, які засновані на машинному навчанні з вчителем.....	30
2.4.1 Наївний баєсів класифікатор.....	31
2.4.2 Метод опорних векторів	34
2.4.3 Метод випадкового лісу	36
2.4.4 FastText класифікатор.....	38
2.5 Метод аналізу тональності, заснований на машинному навчанні без вчителя	40

2.6	Методи аналізу тональності, засновані на глибинному навчанні.....	41
2.6.1	CNN	41
2.6.2	RNN LSTM	42
2.6.3	BERT.....	45
2.7	Порівняння розглянутих методів	47
2.8	Запропонований метод визначення тенденцій зміни цін акцій.....	49
2.9	Висновки з розділу 2	53
3	Розробка архітектури програми.....	53
3.1	Вибір мови програмування	54
3.2	Вибір засобів Python для роботи з даними	55
3.3	Вибір засобів Python для реалізації класифікації та прогнозування.....	56
3.4	Вибір вебфреймворку	58
3.5	Вибір бази даних.....	60
3.6	Структура програми	62
3.7	База даних	64
3.8	Діаграма компонентів.....	68
3.9	Діаграми прецедентів	69
3.10	Висновки з розділу 3.....	70
4	Основні рішення щодо реалізації компонентів системи	70
4.1	Алгоритм функціонування програмного забезпечення.....	71
4.2	Опис модулів реалізованої програми	72
4.2.1	Опис модулів пакету db	72
4.2.2	Опис модулів пакету fasttext_analysis	73
4.2.3	Опис модулю stock_categorization.....	74
4.2.4	Опис модулю plots	75
4.2.5	Опис модулю app	76
4.3	Висновки з розділу 4	77
5	Експлуатація та експериментальне дослідження програми.....	78
5.1	Призначення програми	78
5.2	Умови виконання програми	78

5.3 Виконання програми	78
5.3.1 Виконання програми в режимі перегляду графіків ціни акцій та оцінок тональності новин однієї компанії.....	79
5.3.2 Виконання програми в режимі перегляду загальної гістограми оцінок тональності новин декількох компаній	82
5.3.3 Виконання програми в режимі перегляду таблиці розподілу акцій за тенденціями зміни їх цін	83
5.3.4 Повідомлення про помилки.....	84
5.4 Експериментальне дослідження запропонованого методу	85
5.5 Висновки з розділу 5	88
Висновки	89
Перелік джерел посилань.....	91
ДОДАТОК А Технічне завдання	97
ДОДАТОК Б Текст програми	102
ДОДАТОК В Слайди презентації.....	116

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API (application programming interface) – прикладний програмний інтерфейс;

CSV (comma-separated values) – текстовий файл зі списком даних;

HTML (HyperText Markup Language) – мова розмітки гіпертексту;

JSON (JavaScript Object Notation) – текстовий формат обміну даними;

URL (Uniform Resource Locator) – єдиний вказівник на ресурс;

БД – база даних;

СУБД – система управління базами даних.

ВСТУП

Аналіз тональності тексту – клас методів якісно-кількісного вивчення документів в комп'ютерній лінгвістиці, який призначений для автоматизованого виявлення емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті, щоб оцінити загальний настрій тексту, чи є він позитивним, негативним або нейтральним, та в якій мірі [1].

Сучасні інструменти аналізу тональності на основі алгоритмів можуть послідовно і з прийнятною точністю обробляти дані та визначати відповідно оцінки настроїв заданих текстів.

В наш час, враховуючи величезні простори інформації в мережі Інтернет, а також підвищення довіри до інформації з соціомедіаресурсів, актуальність технології аналізу тональності росте і дана технологія знаходить застосування в багатьох сферах.

Наприклад, аналіз тональності здатний допомогти розібратися в законах людської природної мови і навчити комп'ютер сприймати його на рівні, наближеному до людського, вводячи для цього крім абстрактних лексем ще й тональність тексту [2].

Таким чином, добре навчена аналізу тональності машина може використовуватися для підвищення якості машинного перекладу, моніторингу суспільної думки, проведення маркетингових кампаній, прогнозування думок на основі текстів, які аналізуються, аналізу соціальних медіа, оцінки новин [3].

Як раз дослідження доцільності застосування аналізу тональності текстових повідомлень про ринок акцій, а саме заголовків новин ринку акцій, проводиться в даній дипломній кваліфікаційній роботі.

З роками список наборів даних, що використовуються інвесторами та трейдерами для дослідження ситуації на ринку, розширився від наборів даних, заснованих на історичних цінах, до статистичних кореляцій між активами, впливу унікальних факторів ризику і т. д. І щоб отримати перевагу розшуковуються нові, нетрадиційні підходи, які мають сенс.

Новини є багатим джерелом наборів даних для удосконалення торгових стратегій. Новини в реальному часі, хоча і менш структуровані, ніж звичайні фундаментальні методи оцінки, зазвичай містять перші підказки про серйозні зміни на ринку [4].

Отже, використання технології аналізу тональності новин разом з традиційними способами дослідження ситуації на ринку може позитивно впливати на якість досліджень і розробка засобу, за допомогою якого можна проводити аналіз тональності заголовків новин ринку акцій для обґрунтування інвестиційних рішень, є актуальним.

Об'єкт дослідження – процес аналізу тональності текстових повідомлень.

Предмет дослідження – методи аналізу тональності текстових повідомлень про ринок акцій.

Мета роботи – розроблення вебзастосунку для проведення аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд ключових понять

Предметною областю дипломної роботи є ринок акцій. Ринок акцій – це ринок, на якому випускаються та торгуються акції або через біржі, або на позабіржових ринках; надає компаніям доступ до капіталу та інвесторам частину власності в компанії з потенціалом реалізувати прибутки, виходячи з її майбутніх результатів [5].

До ключових понять відносяться:

– фондова біржа – постійно діючий організований ринок цінних паперів, на якому відбувається купівля-продаж акцій, облігацій, векселів тощо та укладаються угоди [6];

– компанія – об'єднання юридичних і фізичних осіб для проведення економічної діяльності (виробничої, торговельної, посередницької, фінансової, страхової тощо); суб'єкт господарювання, який має різні організаційно-правові форми [7];

– акція – це титул власності, цінний папір без установленого строку обігу, що засвідчує дольову участь у статутному фонді акціонерного товариства, підтверджує членство в акціонерному товаристві та право на участь в управлінні ним, дає право його власникові на одержання частини прибутку у вигляді дивіденду [8];

– трейдер – учасник біржової торгівлі, який здійснює короткострокові або середньострокові торговельні операції з цінними паперами (облігаціями, акціями) з метою отримання прибутку [9];

– інвестор – людина чи комерційна установа, яка здійснює довгострокове вкладення капіталу в якусь справу, підприємство з метою збереження коштів та подальшого їх примноження [10];

– аналіз ринку – комплекс дій, спрямованих на вивчення всіх умов, факторів, ситуацій, які впливають на стан, зміни обсягів, структури, масштабів і розвиток ринку [11].

1.2 Огляд програмних засобів, що вирішують аналогічні завдання

Було виявлено існуючі мобільні та вебзастосунки, які вирішують подібні завдання, а саме надають інструменти для аналізу тональності інформаційних ресурсів, в тому числі і фінансових новин, для дослідження ситуації на ринку.

До програмних засобів, які будуть розглянуті, належать Sentdex, StockSnips, CryptoMood та Market Prices Forecast.

1.2.1 Sentdex

Вебзастосунок Sentdex базується на власному алгоритмі аналізу тональності, який розуміє емоції, які люди використовують у своєму онлайн-спілкуванні, і перекладає їх на комп'ютерну мову.

Отримані алгоритмом дані можна використовувати для глибшого розуміння світу і дослідження ставлення людей по всьому світу до різних тем (компанії, політика, війна та багато іншого). Але нічого на вебсайті Sentdex не слід сприймати як інвестиційну пораду, бо весь його вміст призначений виключно для освітніх цілей [12].

На рисунку 1.1 представлено екранну копію вебзастосунку Sentdex з оглядом результатів аналізу тональності новин компанії Apple.

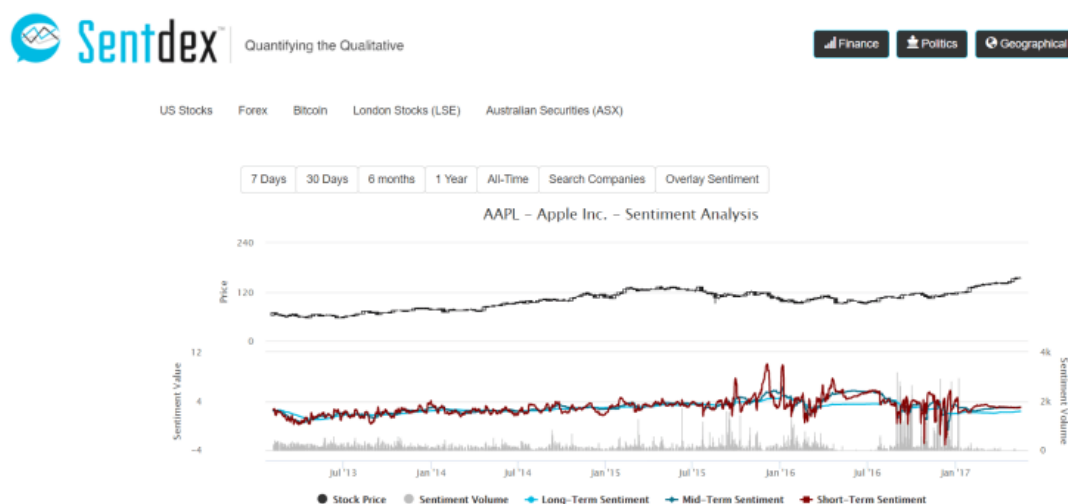


Рисунок 1.1 – Вебзастосунок Sentdex

Список функцій Sentdex:

- аналіз тональності новин ринку акцій США, Лондону, австралійських цінних паперів, ринку Forex та Bitcoin;
- аналіз тональності політичних новин;
- географічний аналіз тональності;
- побудова графіків зміни ціни та значень тональності;
- вибір періоду за який час відображаються дані.

Перевагами застосунку є простий дизайн, аналіз тональності різних тем та вільний доступ до користування ним як для освітніх цілей.

1.2.2 StockSnips

Компанія StockSnips надає мобільний та вебзастосунок з потужними інструментами аналізу тональності для дослідження ситуації на ринку акцій.

Технологія StockSnips базується на власному текстовому аналізі та моделі машинного навчання, які були натреновані на основі розмічених даних, створених фінансовими експертами [13].

На рисунку 1.2 представлено екранну копію мобільного застосунку StockSnips при перегляді графіків ціни та оцінок тональності новин.

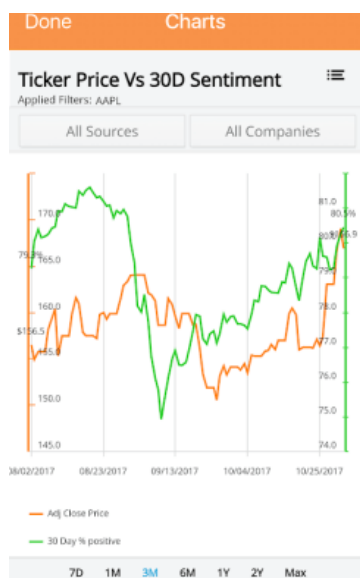


Рисунок 1.2 – Перегляд графіків ціни та оцінок тональності новин в StockSnips

На рисунку 1.3 представлено екранну копію мобільного застосунку StockSnips при перегляді проаналізованих новин.

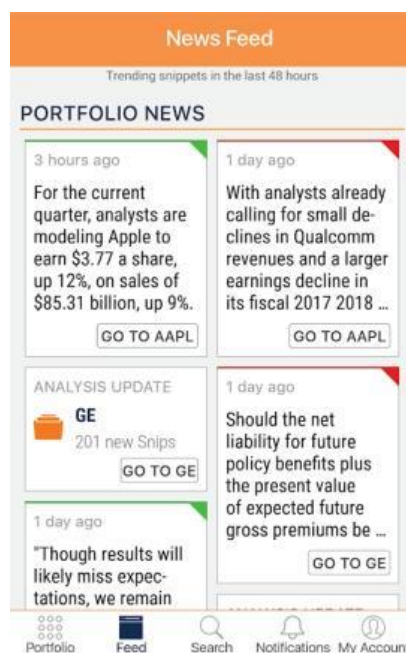


Рисунок 1.3 – Перегляд новин в StockSnips

Список функцій StockSnips:

- аналіз тональності новин ринку акцій з можливістю обирати сектори;
- фільтрація новин;
- відображення індикаторів тональності у кожного джерела новин;
- побудова графіків зміни ціни та індикаторів тональності;
- вибір періоду за який час відображаються дані.

Перевагами застосунку є прогресивність використовуваних технологій для аналізу тональності, інтуїтивно зрозумілий дизайн, різні функції для просунутого дослідження ситуації на ринку.

1.2.3 CryptoMood

Мобільний застосунок CryptoMood забезпечує зручним оглядом поточної тональності новин про криптовалюту. Використовувана застосунком технологія

базується на штучному інтелекті та даних з понад 50 000 джерел для аналізу настроїв ринку та визначення торгових сигналів [14].

На рисунках 1.4-1.5 представлено екранні копії мобільного застосунку.



Рисунок 1.4 – Перегляд графіку ціни та тональності новин в CryptoMood

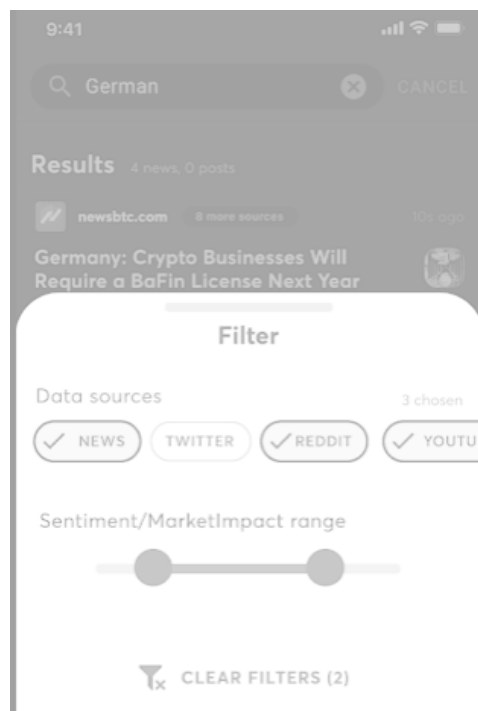


Рисунок 1.5 – Фільтр новин в CryptoMood

Список функцій CryptoMood:

- аналіз тональності новин криптовалютного ринку;
- фільтрація новин;
- відображення індикаторів тональності у кожного джерела новин;
- показ новин про криптовалюти та публікацій в соціальних мережах від відповідних впливових осіб у криптовалютному просторі;
- побудова графіків зміни ціни та індикаторів тональності новин;
- вибір періоду за який час відображаються дані.

1.2.4 Market Prices Forecast

Вебзастосунок Market Prices Forecast аналізує історії цін 514 цінних паперів ETF та на їх основі надає прогнози.

Крім цього, надає можливість завантаження CSV-файлу з історією цін будь-якої іншої акцій, якої немає в списку застосунку. Для вдалого оброблення та подальшого прогнозу на основі завантажених даних файл повинен мати визначені вимогами стовпці та назву [15].

На рисунку 1.6 представлено екранну копію вебзастосунку.

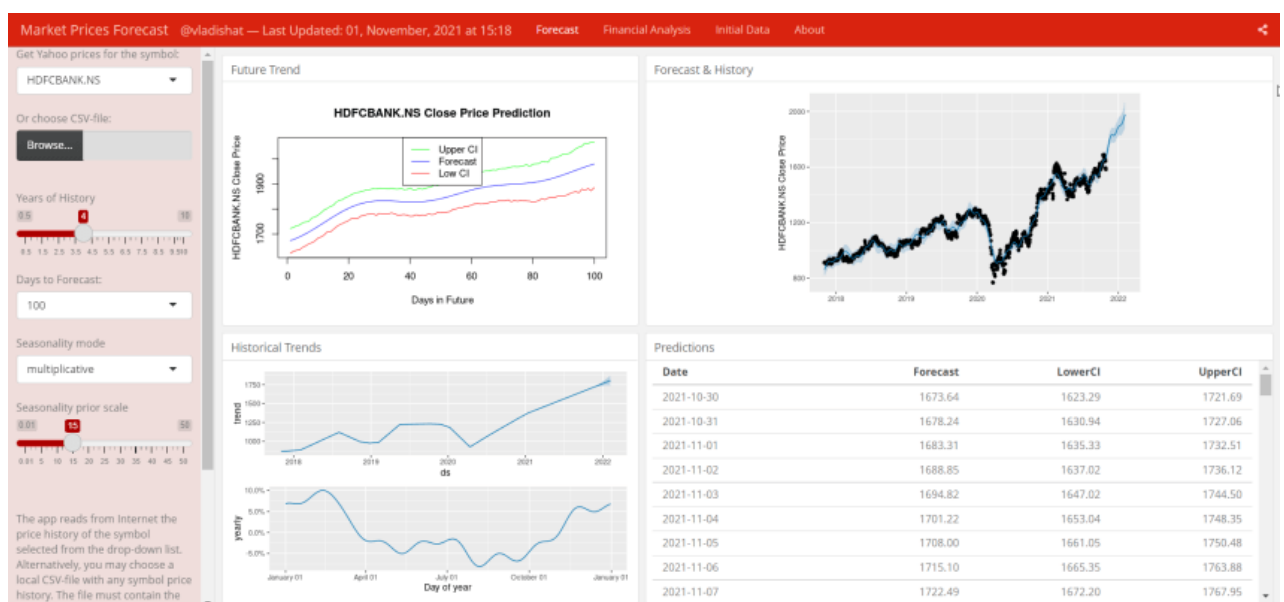


Рисунок 1.6 – Вебзастосунок Market Prices Forecast

Список функцій Market Prices Forecast:

- аналіз історії цін на цінні папери;
- прогноз цін на цінні папери;
- налаштування на скільки днів робити прогноз;
- завантаження історії цін на цінні папери з власного файлу.

1.3 Огляд застосування попередньої обробки даних та методів створення вкладень слів

Розглянуті вище програмні засоби базуються на результатах використовуваних ними певних методів машинного або глибинного навчання.

А невід’ємними та важливими складовими для отримання якісних результатів при застосуванні будь-якого методу машинного або глибинного навчання у задачах обробки природньої мови, в тому числі і у задачах аналізу тональності текстів, є попередня обробка даних та створення вкладень слів.

1.3.1 Огляд застосування попередньої обробки даних

Попередня обробка даних є важливим етапом, після якого починається сам процес аналізу даних, в тому числі і аналіз тональності текстів. Якість результатів аналізу залежить як від поданих на вхід даних, так і від якості їх попередньої обробки.

Цей процес можна розділити на такі етапи:

- а) видалення дублікатів (текстів, які вже зустрічалися в наборі);
- б) переведення тексту в нижній регістр;
- в) розбиття тексту на речення та слова (токенізація);
- г) видалення розділових знаків, спеціальних символів, символів нової строки та табуляції;
- г) видалення посилань;
- д) видалення HTML тегів;

е) видалення стоп-слів (слів, які поширені в мові, але не несуть якоїсь цінності для аналізу, як наприклад артиклі в англійській мові);

є) приведення слів до нормальної форми – процес, під час якого слова приводяться до єдиного представлення (наприклад, «writes», «writing», «wrote» після нормалізації перетворюються у «write»). Існують два підходи для нормалізації:

1) стемінг – в цьому випадку виділяється основа слова, доповнивши яку можна отримати слова-нащадки. Такий підхід є наївним, бо він просто обрізає рядок, відкидаючи закінчення, тому таким чином не виявить зв'язок між наприклад «sing» і «sung»;

2) лематизація – вирішує зазначену дану проблему більш комплексно, використовуючи структуру даних, в якій всі форми слова пов'язуються з його найпростішою формою – лемою. Лематизація зазвичай приносить більше користі, ніж стемінг [16].

1.3.2 Огляд методів створення вкладень слів

Вкладання слів (англ. word embedding) – це методики мовного моделювання та навчання ознак в обробці природної мови, в яких слова або фрази зі словника перетворюються у вектори дійсних чисел [17]. Вкладання слів дають математичне вкладення з багатовимірною простору, по одному на слово, до неперервного векторного простору з розмірністю, яка є набагато нижчою.

Застосування вкладень слів та фраз як базового представлення вхідних даних підсилюють продуктивність в задачах обробки природної мови, таких як: синтаксичний аналіз, аналіз тональності тексту.

Слова, що зустрічаються в подібних контекстах, мають, як правило, схожі значення. Подібність слів обчислюється косинусом подібності та евклідовою відстанню.

До найбільш популярних представників алгоритмів вкладень слів належать Word2Vec, GloVe та FastText.

Word2Vec пропонує два підходи для векторних представлень слів (обидва із залученням нейронних мереж; порядок слів контексту не впливає на результат ні в одному з них): Skip-gram та CboW.

Skip-gram – використовує поточне слово, щоб передбачати слова, які його оточують; добре працює з невеликою кількістю навчальних даних і відображає навіть рідкісні слова чи фрази; цільову функцію алгоритму можна підрахувати за виразом (1.1):

$$J = \frac{1}{T} \sum_{t=1}^T \sum_{-n \leq j \leq n} \log p(w_{t+j} | w_t); \quad (1.1)$$

CboW (англ. Continuous Bag of Words, «безперервний мішок зі словами») – архітектура, яка передбачає поточне слово, виходячи контексту, який його оточує; на тренування витрачається у кілька разів менше часу, ніж на skip-gram і забезпечує дещо кращу точність для слів, які часто зустрічаються в тексті; цільову функцію алгоритму можна підрахувати за формулою (1.2):

$$J = -\log \hat{P}(w_t | w_{t-n+1} \dots w_{t-1}). \quad (1.2)$$

Перевагами Word2Vec є: проста архітектура, швидке навчання моделі, вкладання слів змістовні, а спірні моменти піддаються розшифровці.

До недоліків можна віднести те, що модель не надає інформації про речення та загальний контекст, не розпізнає випадків, коли одне й те саме слово може мати різні значення в залежності від контексту і не дуже якісно обробляє невідомі та рідкісні слова [18].

GloVe (англ. Global Vectors, «глобальні вектори») є алгоритмом спонтанного навчання для отримання векторних представлень слів, що досягається відображенням слів до змістовного простору, в якому відстань між словами пов'язана з семантичною подібністю.

Цільову функцію GloVe алгоритму можна підрахувати за виразом (1.3):

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^W f(P_{ij})(u_i^T v_j - \log P_{ij})^2. \quad (1.3)$$

Для кожної пари слів i та j , які можуть зустрічатися одночасно проводиться мінімізація різниці добутку їх вкладень слів та логарифмічним підрахунком i та j . Завдяки $f(P_{ij})$ виходить зменшити вагу дуже частих співпадінь та обмежити важливість дуже частих слів.

Вектори u та v є взаємозамінними. Два різних набори векторів забезпечують кращу стабільність при оптимізації.

Як і Word2Vec, GloVe має просту архітектуру та швидко навчається. Крім цього, на відміну від Word2Vec, метод не покладається лише на локальну статистику (локальну контекстну інформацію про слова), а враховує ще й глобальну статистику (спільне входження слів на основі підрахунку) для отримання векторів слів. Також на відміну від віконних методів, де проводиться оптимізація одного вікна за раз, GloVe оптимізує один підрахунок за раз та дозволяє фіксувати семантику незалежно від кількості входжень слів.

До недоліків можна віднести те, що модель не надає інформації про загальний контекст речення і з її допомогою не можуть бути представлені слова, що не зустрічаються в навчальній вибірці [19].

Модель FastText можна назвати модифікацією Word2Vec, яка має кращу ефективність та продуктивність у зв'язку з тим, що її можна ще швидше натренувати, вона може навчитись розуміти суфікси та префікси, на менших тренувальних наборах даних та з незнайомими словами показує кращі результати.

Продуктивність FastText завдяки застосуванню subword-моделі, яка полягає у представленні слів ланцюжками символів визначеної довжини. Наприклад, слово «настрій» в залежності від гіперпараметрів може складатися з

n-грам «нас», «аст», «стрі», «трій» і т. п. Тобто вектор слова є сумою всіх його n-грам.

Після того, як слова були представлені за допомогою n-грам застосовується модель Skip-gram з негативним семпліруванням – це спосіб створити для навчання векторної моделі негативні приклади, тобто показати пари слів, які не є сусідами по контексту.

Для кожного позитивного прикладу (коли слова в тексті стоять поруч, наприклад, «рожеве фламінго») підбирається кілька негативних («рожевий аналіз», «рожевий вітер» і т. п.). Тобто цей підхід дозволяє максимізувати ймовірність зустрічі потрібного слова в контексті, який є для нього типовим, і мінімізувати – у рідкісному/нетиповому контексті.

Цільову функцію Skip-gram з негативним семпліруванням можна виразити формулою (1.4):

$$J_t(\theta) = \log \sigma(u_o^T v_c) + \sum_{j \sim P(w)} \left[\log \sigma(-u_j^T v_c) \right], \quad (1.4)$$

де $\sigma = \frac{1}{e^x}$;

θ – різні змінні при ітерації;

t – ітерація;

u та v – вектори.

Ознаки, отримані за допомогою розбиття на n-грами, мають величезну розмірність, що може сповільнити роботу моделі. Для фіксування розмірності ознак застосовується хешування ознак, тобто ознаки отримують хеш-індекси, що допомагає зчитувати їх швидше.

Недоліком FastText є те, що при його застосуванні ігнорується спільна поява слів, тому модель не так якісно навчається різним значенням слів у різних контекстах. В цьому виграє GloVe зі своїм статистичним підходом [20].

Якщо узагальнити, то Word2Vec зосереджується на тому, чи з'являються

слова у подібних контекстах; GloVe зосереджується на спільних появах слів у всьому тексті, адже його вкладання стосуються ймовірності того, що два слова з'являться разом; FastText покращує Word2Vec, враховуючи також частини слів, що дозволяє навчати моделі вкладень слів на менших наборах даних та узагальнювати невідомі слова.

Таким чином, всі ці алгоритми мають багато спільного, та для отримання кращих результатів насамперед потрібно використовувати їх у відповідних контекстах.

1.4 Постановка завдань дослідження

Розглянуті в пункті 1.2 програмні засоби схожі між собою за своїми цілями та функціями, які вони надають. Sentdex, StockSnips та CryptoMood застосовують машинне або глибинне навчання для аналізу тональності, а Market Prices Forecast для прогнозування цін.

Існують різні алгоритми машинного або глибинного навчання, які можна застосувати для визначеної задачі і є методи, які не відносяться до машинного або глибинного навчання, але також можуть використовуватися для цих цілей.

Завданням дослідження є порівняти актуальні способи для проведення аналізу тональності, визначити який з них можна застосувати для отримання найбільш якісних результатів.

Щоб дослідити на практиці один зі способів буде розроблятися система для проведення аналізу тональності заголовків новин ринку акцій для обґрунтування інвестиційних рішень. До функцій системи належать:

- аналіз тональності заголовків новин ринку акцій;
- категоризація акцій за тенденціями зміни їх цін.
- побудова графіків ціни акцій та оцінок тональності новин;
- побудова гістограми кількості новин проаналізованих за день;
- побудова загальної гістограми оцінок тональності декількох акцій;
- побудова таблиці розподілу акцій за тенденціями зміни їх цін.

1.5 Висновки з розділу 1

У першому розділі було проведено аналіз предметної області та огляд програмних засобів, які вирішують аналогічні завдання; описано процес попередньої обробки даних та популярні алгоритми вкладень слів Word2Vec, GloVe та FastText; визначено завдання дослідження та функції системи, що розроблюється.

У дипломній кваліфікаційній роботі мають бути розв’язані наступні завдання:

- вибір засобів розробки (мови програмування, потрібних бібліотек та фреймворків, системи управління базою даних);
- проєктування програмного забезпечення (структури програми, схеми бази даних, схеми функціонування програми, діаграми прецедентів та діаграми компонентів);
- розробка програмного забезпечення;
- експериментальне дослідження програми.

2 МАТЕРІАЛИ І МЕТОДИ

Враховуючи актуальність задачі аналізу тональності, над удосконаленням її рішень багато працюють спеціалісти науки про дані, яка в наш час стрімко розвивається. Тому є доступним простір розроблених методів та інструментів, підготовлених наборів даних для реалізації систем аналізу тональності.

Одним зі способів визначення настрою тексту є його визначення за бінарною шкалою, що є найбільш поширеним та простим способом. При цьому способі використовуються дві оцінки: позитивна та негативна. До недоліків такого підходу можна віднести те, що емоційну складову тексту можна точно визначити не завжди, адже текст може мати ознаки як позитивної оцінки, так і негативної.

Також можна класифікувати настрій тексту за багатосмуговою шкалою, як було здійснено Пангом і Снайдером наприклад. Вони розширили основну задачу класифікації відгуків на фільми за настроєм в бік прогнозування рейтингу фільму за трьох або чотирьохбальною шкалою. Крім цього, Снайдер здійснив поглиблений аналіз відгуків на ресторани і зробив прогноз рейтингів за п'ятибальною шкалою їх властивостей, таких як їжа і атмосфера.

Ще одним методом аналізу тональності є застосування систем шкалювання. Після дослідження тексту алгоритмами обробки природньої мови виділені з тексту слова та вирази аналізуються з ціллю зрозуміти їх значення, а потім словам, зазвичай пов'язаними з негативними або нейтральними або позитивними настроями, проставляються числа за шкалою від -10 до 10 (від самого негативного до самого позитивного) відповідно. Шкали можуть відрізнятися в залежності від застосовуваного інструменту.

Крім класифікації текстів за настроєм існує ще напрямок, який полягає у визначенні суб'єктивності/об'єктивності тексту, що полягає у віднесенні тексту в один з двох класів – суб'єктивний або об'єктивний. Іноді така задача може бути більш складною, ніж звичайна класифікація настроїв, адже суб'єктивність

виразів може залежати від контексту, а об'єктивний текст може мати суб'єктивні речення (наприклад, стаття з новин, цитування) [21].

2.1 Задача класифікації текстових повідомлень про ринок акцій за їх тональностями

В даному дослідженні для виконання аналізу тональності текстових повідомлень (а саме заголовків новин) про ринок акцій вирішується задача бінарної класифікації – формалізована задача, що полягає в розподілі множини об'єктів на дві апіорно задані групи, які називаються класами, всередині кожного з яких об'єкти мають приблизно однакові ознаки [22].

Крім цього, повинна застосовуватися система шкалювання, за якою визначається, що повідомлення з оцінкою тональності -1 має найбільш негативну тональність, а повідомлення з оцінкою 1 – відповідно найбільш позитивну.

Тобто для текстів, які аналізуються, задача полягає у підборі відповідного класу за настроєм тексту: позитивний або негативний, а також у визначенні числового значення від -1 до 1, яке показує наскільки текст є позитивним або негативним.

На основі отриманих результатів класифікації повинно виконуватись прогнозування тенденцій зміни цін акцій на ринку для їх категоризації відповідно за їх тенденціями. Категоризація акцій полягає у їх розподілі на такі задані категорії:

- ціна майже не зміниться;
- незначне падіння ціни;
- значне падіння ціни;
- незначне зростання ціни;
- значне зростання ціни.

Далі розглядатимуться існуючі методи та підходи для реалізації аналізу тональності тексту:

- метод, заснований на словнику;

- метод, заснований на правилах;
- методи, засновані на машинному навчанні з вчителем;
- метод, заснований на машинному навчанні без вчителя;
- методи, засновані на глибинному навчанні.

Крім цього, методи, засновані на машинному та глибинному навчанні можуть бути використані для реалізації методу визначення тенденцій зміни цін акцій, а саме для потрібного для нього прогнозування тенденцій зміни цін.

2.2 Метод аналізу тональності, заснований на словнику

Даний метод є найбільш простим. Він полягає у визначенні тональності тексту базуючись на тональностях слів або фраз з тексту. Ці тональності слів або фраз беруться із заздалегідь підготовлених тональних словників. Звичайно, дані словники не покривають визначенням тональності всіх слів мови (в даному випадку англійської).

Приклад такого словника представлений в таблиці 2.1.

Таблиця 2.1 – Приклад тонального словника

Слово	Тональність
Чудовий	0.9
Відмінний	0.7
Хороший	0.5
Нормальний	0.0
Поганий	-0.6
Огидний	-0.8

В процесі виконання цього методу спочатку підраховується кількість позитивних та негативних слів в тексті, потім рахується різниця між кількістю цих позитивних та негативних слів. Отримане число і визначає тональність тексту. Якщо кількість позитивних слів в тексті більша, ніж кількість негативних,

то тональність тексту визначається як позитивна, і навпаки. Крім цього, може додатково підраховуватися різниця проценту позитивної та негативної тональності тексту.

Метод може бути адаптований для підрахування валентності будь-якого настрою (наприклад, радості або смутку) настільки, наскільки тональний словник, який використовується для аналізу тональності, забезпечує відповідними словами та виразами. Наприклад, кількість слів, які асоціюються з радістю може бути використана для визначення саме «радісності» тексту.

Також замість визначення у словнику слів як позитивних та негативних можна визначати їх тональність числовими значеннями починаючи з -1 та закінчуючи 1, де оцінка тональності -1 показує, що слово має найбільш негативний окрас, а оцінка 1 – відповідно найбільш позитивний. І потім брати середнє арифметичне значення отриманих оцінок слів як загальну оцінку тональності тексту.

Продуктивність методу дуже залежить від тонального словника, який використовується. І саме трудомісткість процесу його складання є основною проблемою. Щоб проводити аналіз тональності з високою точністю терміни словника повинні відповідати предметній області тексту, який аналізується, і за кількістю і за оцінками тональності, бо одне й те саме слово в різних предметних областях може мати різні настрої.

Також обмеженням методу є те, що він не може обробляти заперечення в реченнях. Наприклад, хоч речення «я не радісний» має негативний окрас, метод визначить його як позитивний, захопившись тільки за слово «радісний» і проігнорувавши слово-заперечення [23].

Таким чином, метод аналізу тональності, заснований на тональному словнику, хоч і є простим у застосуванні, але не є універсальним. Використовуючи цей метод рано чи пізно доведеться зіткнутися з проблемою неможливості підвищення якості аналізу тональності тексту.

Тим не менш, приклад вдалого застосування даного методу для аналізу тональності новин ринку акцій можна побачити в статтях [24], [25].

2.3 Метод аналізу тональності, заснований на правилах

Даний метод полягає у застосуванні набору правил для визначення тональності тексту.

Одне й те саме слово в різних предметних областях може мати різні настрої (як було сказано в п. 2.1), крім цього, є слова, які самі по собі не відносяться до конкретно визначеної тональності, але в сполученні з іншими словами набувають певного настрою. Тож метод може мати правила для аналізу тональності двох типів: незалежні від контексту та залежні від контексту.

Незалежні від контексту правила враховують лише настрої слів та фраз, які є уніфікованими.

Наприклад, на рівні уніфікації прислівників та прикметників такі правила можна сформулювати таким чином:

- INTENSIFIER + POSITIVE → VERY POSITIVE;
- INTENSIFIER + NEGATIVE → VERY NEGATIVE;
- INVERTOR + POSITIVE → NEGATIVE;
- INVERTOR + NEGATIVE → POSITIVE;
- INTENSIFIER + NEUTRAL → POSITIVE;
- INVERTOR + NEUTRAL → NEGATIVE;
- POSITIVE + NEUTRAL → POSITIVE;
- NEGATIVE + NEUTRAL → NEGATIVE;
- POSITIVE + POSITIVE → POSITIVE;
- NEGATIVE + NEGATIVE → NEGATIVE;
- NEUTRAL + POSITIVE → POSITIVE;
- NEUTRAL + NEUTRAL → NEUTRAL;
- NEUTRAL + NEGATIVE → NEGATIVE;
- POSITIVE + NEGATIVE → NEGATIVE;
- NEGATIVE + POSITIVE → NEGATIVE.

Тож для виразу «дуже гарний» може діяти правило виду INTENSIFIER + POSITIVE → VERY POSITIVE, яке визначає вираз як дуже позитивний.

А для виразу «не подобається» може діяти правило виду INVERTOR + POSITIVE → NEGATIVE, яке визначає вираз як негативний.

Залежні від контексту правила враховують загальний настрій фрази, не дивлячись на те, що окремо слова фрази не мають визначеної тональності.

Наприклад:

- («high», NEUTRAL) + («price», NEUTRAL) → NEGATIVE;
- («high», NEUTRAL) + («quality», NEUTRAL) → POSITIVE;
- («small», NEUTRAL) + («portion», NEUTRAL) → NEGATIVE;
- INTENSIFIER + («big», NEUTRAL) + («portion», NEUTRAL) → VERY POSITIVE.

Такі правила міцно прив'язані до предметної області тексту, який аналізується, тому якщо вона змінюється, правила повинні бути максимально адаптовані до нової предметної області.

Багато комерційних систем використовують метод аналізу тональності, заснований на правилах, хоча даний метод і потребує багато витрат, адже для того, щоб метод надавав якісні результати потрібно розробити велику кількість правил [26].

Тим не менш, метод може працювати доволі точно, якщо є якісна база правил і приклади його вдалого застосування для аналізу тональності новин ринку акцій можна побачити в статтях [27], [28].

2.4 Методи аналізу тональності, які засновані на машинному навчанні з вчителем

Методи аналізу тональності, засновані на машинному навчанні з вчителем полягають в обробці алгоритмами машинного навчання тренувальних наборів даних, в яких є заздалегідь розмічені тексти за відповідними настроями (позитивний, негативний та нейтральний, якщо є) і використанні навченої моделі, яка отримується на виході алгоритмів для проведення класифікації нових текстів за їх тональностями.

Процес створення системи для аналізу тональності текстів не дуже відрізняється від створення інших систем, які застосовують машинне навчання:

- формування колекції текстів для навчання класифікатора;
- визначення тональності для кожного тексту (розмітка даних);
- створення вкладень слів (представлення текстів як векторів ознак);
- вибір алгоритму класифікації та подальше навчання класифікатора;
- застосування навченої моделі для аналізу тональності [29].

Важливим застереженням, про яке варто згадати, є те, що машинне навчання з вчителем вимагає маркування даних, що може означати, що будь-яка суб'єктивність даних відобразиться в моделі.

Було виявлено, що для задачі аналізу тональності текстів, зокрема і для аналізу тональності текстових повідомлень про ринок акцій, популярними є: наївний баєсів класифікатор, метод опорних векторів, метод випадкового лісу та FastText класифікатор.

2.4.1 Наївний баєсів класифікатор

Наївний баєсів класифікатор – це ймовірнісний класифікатор, що використовує теорему Баєса для визначення ймовірності приналежності елемента вибірки до одного з класів при припущенні незалежності змінних [30].

Якщо на основі значень змінних можна однозначно визначити клас якогось елемента вибірки, байєсів класифікатор повідомить про відповідну ймовірність приналежності даного елемента до цього класу [30].

У проміжних випадках, коли елемент вибірки може належати до різних класів з різною ймовірністю, результатом роботи класифікатора є вектор, компоненти якого є ймовірностями приналежності до того чи іншого класу [30].

Теоремі Баєса відповідає формула (2.1):

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \quad (2.1)$$

де A та B – події;

$P(A)$ та $P(B)$ – ймовірності A та B безвідносно одна до одної;

$P(A|B)$ – умовна ймовірність A за умови істинності B ;

$P(B|A)$ – ймовірність спостереження події B за умови істинності A .

Конкретно для задачі аналізу тональності проводиться прогнозування приналежності тексту t до класу c , тому формула виглядатиме так (2.2):

$$P(c|t) = \frac{P(c)P(t|c)}{P(t)}, \quad (2.2)$$

де $t = \{w_1, w_2, \dots, w_n\}$ – вектор ознак;

w_i – вага i -ої ознаки;

n – кількість ознак.

За цією формулою обчислюються умовні ймовірності приналежності текстів для кожного з класів.

Якщо умовна ймовірність максимальна, то відповідний клас c^* виявляється найбільш ймовірним класом, до якого належить текст (2.3):

$$c^* = \arg \max_c P(c|t) = \arg \max_c \frac{P(c)P(t|c)}{P(t)}. \quad (2.3)$$

Формула може бути спрощена шляхом відкидання знаменника, адже $P(c|t)$ обчислюється для кожного можливого класу і ймовірність $P(t)$ не змінюється для одного й того ж тексту (2.4):

$$c^* = \arg \max_c P(c)P(t|c). \quad (2.4)$$

Байєсів класифікатор припускає, що всі ознаки текстів не залежать один

від одного (класифікатор названий наївним саме через це припущення). Це надзвичайно серйозне припущення і, у загальному випадку, є невірним, але не дивлячись на це, класифікатор видає досить задовільні результати.

Крім цього, припускається, що позиція слів у реченнях не так важлива, тому умовну ймовірність для ознак тексту можна представити так (2.5):

$$P(w_1 | c) * P(w_2 | c) * \dots * P(w_n | c) = \prod_i P(w_i | c). \quad (2.5)$$

Таким чином, для знаходження найбільш ймовірного класу для заданого тексту, необхідно порахувати умовні ймовірності приналежності тексту для кожного з представлених класів окремо і обрати клас, який має максимальну ймовірність (2.6):

$$c^* = \arg \max_c \left[P(c) * \prod_i P(w_i | c) \right]. \quad (2.6)$$

Оцінка ймовірності класу $P(c)$, яка є відношенням кількості текстів цього класу в тренувальному наборі даних (Q_c) до загальної кількості текстів (Q), визначається за формулою (2.7):

$$P(c) = \frac{Q_c}{Q}. \quad (2.7)$$

А оцінка умовних ймовірностей для ознак $\hat{P}(w_i | c)$ формулою (2.8):

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c)) + |V|}, \quad (2.8)$$

де $\hat{P}(w_i | c_j)$ – відношення кількості слів w_i в класі c_j до загальної кількості слів;
 V – кількість слів у наборі даних.

Як видно, додається одиниця в чисельник і знаменник для обробки випадків, коли в тестовому наборі даних зустрічаються слова, які не були в тренувальному, і отримують нульові ймовірності, а через це і ймовірності приналежності текстів до будь-якого з класів також будуть дорівнювати нулю, що є неправильним. Таким чином, такі слова отримують незначну, але не нульову ймовірність і з'являється можливість визначити текст в один із класів.

Наївний байєсів класифікатор є одним з найчастіше використовуваних класифікаторів, відомий своєю простотою у розробці та тестуванні, високою швидкістю роботи, помірними вимогами до пам'яті.

Але як ймовірнісний класифікатор, він дуже залежить від заданих на вхід даних, які повинні бути вичерпними та репрезентативними. Відсутність якісних даних для навчання призводить до незадовільних результатів аналізу тональності текстів з великою кількістю незнайомих для класифікатора слів [30].

Приклади вдалого застосування класифікатора для аналізу тональності новин ринку акцій можна побачити в статтях [31], [32].

2.4.2 Метод опорних векторів

Метод опорних векторів – це метод аналізу даних для реалізації регресійного аналізу та класифікації за допомогою моделей навчання з вчителем, які пов'язані з алгоритмами навчання – опорно-векторними машинами (ОВМ, англ. support vector machines, SVM); входить до сімейства лінійних класифікаторів [33].

В даному методі точка в просторі розглядається як вектор розмірності p і ведеться перевірка на те, чи є можливість розділити дані точки гіперплощиною розмірністю $(p - 1)$, яка зветься лінією класифікатора.

Дані можуть бути класифіковані з допомогою різних гіперплощин. Краща

гіперплощина – це та, при побудові якої поділ і різниця між класами максимальна, і вона є гіперплощиною максимальної різниці.

Опорні вектори – це точки, для яких відстанню до гіперплощини є $\frac{1}{\|w\|}$.

Вони є ефективними елементами на навчальній вибірці. Якщо всі точки, крім опорних векторів, видалити, алгоритм методу залишиться яким і був. Ця властивість робить метод опорних векторів унікальним, несхожим на всі інші методи, де всі точки тренувального набору даних використовуються для оптимізації функції.

Тренувальний набір даних можна представити як набір D , який складається з n об'єктів (2.9):

$$D = \{(x_i, y_i) \mid x_i \in R^p, y_i \in (-1, 1)\}, \quad (2.9)$$

де y_i приймає значення від -1 до 1, визначаючи до якого класу належить точка x_i .

Кожна точка x_i є вектором розмірності p . Ведеться пошук гіперплощини максимальної різниці, яка розділяє об'єкти з $y_i = 1$ від об'єктів з $y_i = -1$.

Будь-яку гіперплощину можна описати як множину точок x , які задовольняють $w * x - b = 0$, де $*$ є скалярним твором нормалі до гіперплощини на вектор x .

Якщо навчальні дані є лінійно роздільними, можна обрати дві паралельні гіперплощини таким чином, що вони розділять множину точок на два класи, і точок між ними не буде. А далі максимізувати дистанцію між ними, одночасно роблячи поворот і паралельно зсув паралельних прямих.

Ці гіперплощини можна описати рівняннями (2.10, 2.11):

$$w * x - b = 1, \quad (2.10)$$

$$w * x - b = -1. \quad (2.11)$$

Відстанню між цими гіперплощинами є $\frac{-2}{\|w\|}$.

Для отримання максимальної відстані треба мінімізувати $\|w\|$, при цьому для точок x_i з першого класу повинно бути справедливо $w * x_i - b \geq 1$, а для x_i з другого класу – $w * x_i - b \leq -1$, і $y_i(w * x_i - b) \geq 1$ для $0 \leq i \leq n$.

Ефективність методу опорних векторів залежить від вибору ядра, параметрів ядра і параметра C для геометричної різниці. Загальним вибором є Гаусове ядро, яке має один параметр γ . Кращу комбінацію C і γ зазвичай вибирають використовуючи пошук по сітці з ростом частоти C і γ .

Фінальна модель, яка використовується для тестування і класифікації нових даних, навчається потім на всій множині з використанням обраних параметрів.

Метод має два суттєвих недоліки: він не працює, якщо класи лінійно нероздільні; якщо в тренувальному наборі даних колекції є помилка – неправильно класифікований один або кілька елементів, то через цих елементів лінія класифікатора може дуже відрізнятись від тієї, яка вийшла б у випадку з коректним набором даних.

Перевага ОВМ полягає в тому, що для класифікації йому може бути достатньо досить невеликого набору даних. При правильній роботі моделі, побудованої на тестовому наборі даних, цілком можливе застосування даного методу на реальних даних. Також метод знаходить лінію поділу з максимальною шириною, завдяки чому і здійснюється більш впевнена класифікація [33].

Метод впевнено застосовується і для аналізу тональності новин ринку акцій, як можна побачити в статтях [31], [32], [34].

2.4.3 Метод випадкового лісу

Випадковий ліс – це алгоритм машинного навчання, який полягає у використанні ансамблю дерев прийняття рішень, який поєднує в собі основні дві

ідеї: метод Бреймана (незалежне навчання алгоритмів класифікації, при якому результат визначається голосуванням) та метод випадкових підпросторів.

Основна ідея методу є наступною: багато некорельованих дерев прийняття рішень (тобто тих, які діють спільно та знаходять рішення незалежно одне від одного) може перевершити рішення, яке було отримано одним деревом.

Некорельовані моделі мають можливість створювати ансамблеві прогнози, які є точнішими, ніж окремо взяті прогнози, бо кожне дерево рішень «захищає» від індивідуальних помилок одне одного. Не зважаючи на те, що деякі дерева можуть бути неправильними, більшість виявлятимуться правильними, адже загалом група дерев рухатиметься в правильному напрямку [35].

Найбільш поширений спосіб побудови дерев ансамблю полягає в тому, що для навчального набору, який складається з N прикладів, з розмірністю простору ознак M , і заданим параметром $m \approx \sqrt{M}$ здійснюється таким чином:

- генерується підвибірка випадковим чином з повторюванням розміром n навчального набору (при цьому деякі об'єкти потраплять в неї декілька разів, а приблизно третина з них взагалі не ввійде до неї);
- випадково обираються m ознак із M ;
- будується дерево рішень, яке класифікує об'єкти підвибірки, і при створенні вузла дерева обирається ознака, на основі якої виконується поділ, не з усіх M ознак, а тільки з m ознак, обраних випадково. Вибір кращої з цих m ознак може здійснюватися за критерієм Джині, за критерієм приросту інформації або якимсь іншим способом;
- ознака X розділяється на два класи: $X_i \geq S_i$ і $X_i \leq S_i$;
- у двох нових класах вимірюється гомогенність за критерієм Джині;
- обирається значення S_i ознаки X , при якому наявна максимальна гомогенність класу;
- будовання дерева відбувається до повного вичерпання підвибірки;
- повернення до початку – генерація нової вибірки, потім повтор наступних пунктів, будуючи наступне дерево.

Чим більше дерев побудовано, тим меншою буде помилка класифікатора на тестовій вибірці.

Класифікація об'єктів виконується голосуванням: кожне дерево ансамблю відносить об'єкт до одного з класів, за який проголосувала більша кількість дерев.

Перевагами методу є те, що метод може ефективно обробляти дані, які мають багато ознак, а також його висока масштабованість. Недоліками є схильність до перенавчання та великий розмір моделей, адже чим більше модель важить, тим більше вона потребує обчислювальних ресурсів та часу для знаходження рішень [35].

Метод можна застосовувати для аналізу тональності новин ринку акцій, як можна побачити в статтях [32], [34].

2.4.4 FastText класифікатор

FastText є розробкою командою Facebook для класифікації текстів, але може бути використана і для створення вкладень слів. З того моменту, коли FastText став продуктом доступним для всіх (2016 рік), він отримав широке застосування через хорошу швидкість тренування та відмінну працездатність.

Модель FastText є нейронною мережею з одним прихованим шаром. На рисунку 2.1 представлено узагальнену архітектуру моделі для речення з кількістю n -грам N .

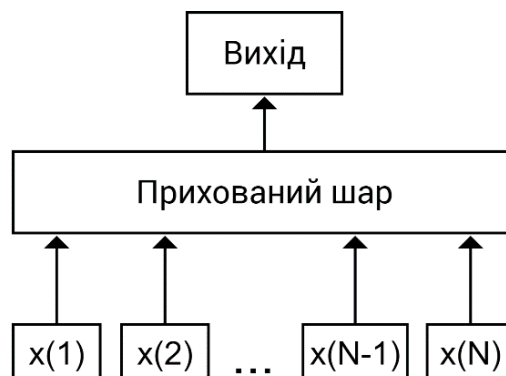


Рисунок 2.1 – Архітектура моделі FastText

На початку текст представляється як «мішок слів» – спрощене подання тексту, що використовується для обробки природних мов та інформаційного пошуку, яке полягає у представленні тексту як мультимножини його слів без урахування граматики та порядку слів, але зі збереженням інформації про їх кількість.

«Мішок слів» пропускається через перший шар, в якому він трансформується у вкладання слів. Надалі отримані вкладання усереднюються по всьому тексту і зводяться до одного єдиного, який застосовується по всьому тексту.

У прихованому шарі йде робота з $n_words * dim$ числом параметрів, де dim – це розмір вкладання, а n_words – розмір словника, що використовується для тексту.

Після усереднення отримується один єдиний вектор, який пропускається через класифікатор на основі мультиноміальної логістичної регресії (що означає, що FastText підтримує багатокласову класифікацію).

На цьому етапі застосовується функція втрат Softmax – нормована експоненційна функція для лінійного відображення вхідних даних першого шару на останній.

За лінійне перетворення виступає матриця розмірністю $dim * n_output$, де n_output є числом дійсних класів.

Кінцева ймовірність L належності тексту до одного з класів підраховується за формулою (2.12):

$$L = -\frac{1}{N} \sum_{n=1}^N y_n \log(f(BAx_n)), \quad (2.12)$$

де x_n – представлення слова у вигляді n -грам;

A – матриця вкладень слів;

B – лінійне перетворення вихідних даних;

f – безпосередньо експоненційна функція.

Для різних завдань класифікації тексту моделі FastText за своєю швидкістю можуть конкурувати з моделями глибинного навчання та за своєю точністю можуть знаходитись з ними на одному рівні. Крім цього, моделі FastText невимогливі до обчислювальних ресурсів та можуть бути навчені на невеликих наборах навчальних даних [36].

Як і вище розглянуті класифікатори, даний метод може використовуватися і для аналізу тональності новин ринку акцій, що можна побачити в статті [37].

2.5 Метод аналізу тональності, заснований на машинному навчанні без вчителя

Застосування машинного навчання без вчителя дозволяє уникнути залежності від розмітки навчальних даних, адже для таких алгоритмів достатньо тільки одних даних. Але для забезпечення задовільної якості результатів даних потрібно дуже багато.

Для визначення тональності тексту при цьому підході можуть виявлятися ймовірісно-статистичні закономірності тексту та вже на їх основі здійснювати виявлення та визначення їх тональності визначених ключових термінів, або може розроблятися алгоритм, який проводить розмітку даних самостійно.

Наприклад, в дослідженні [38] спочатку виконується ініціалізація: проводиться класифікація за тональностями із застосуванням вже натренованої для схожих задач моделі, вибираються найбільш вірогідні результати. Потім результати отримані на етапі ініціалізації застосовуються для класифікації у моделі, яка навчається, а далі вибираються найбільш вірогідні результати вже цієї моделі і вже вони застосовуються для класифікації у моделі. Це повторюється поки модель не опрацює весь набір навчальних даних або не виконається задана умова зупинки.

Однак такі методи вимагають складних налаштувань і не факт, що забезпечать такою високою точністю результатів аналізу тональності як методи

машинного або глибинного навчання з вчителем.

А приклад спроби застосування машинного навчання без вчителя для аналізу тональності саме новин ринку акцій можна побачити в статті [39], де в підсумку зазначається, що запропонований метод аналізу потребує ще покращення для конкурування з іншими методами.

2.6 Методи аналізу тональності, засновані на глибинному навчанні

Глибинне навчання – це клас алгоритмів машинного навчання, які базуються на застосуванні нейронних мереж (обчислювальні системи, які розроблені за принципом функціонування та організації мереж нервових клітин живого організму) та використовують каскад багатьох шарів вузлів нелінійної обробки для виявлення ознак та перетворень, при цьому кожний наступний шар використовує вихід із попереднього як вхід [40].

Якщо машинне навчання – це навчання комп'ютера здатності мислити та діяти з меншим втручанням людини, то глибинне навчання – це навчання комп'ютера мислити, використовуючи структури, змодельовані на основі людського мозку. При цьому глибинне навчання вимагає дуже багато часу та даних для навчання.

Було виявлено, що для задачі аналізу тональності текстів, зокрема і для аналізу тональності текстових повідомлень про ринок акцій, популярними методами заснованими на глибинному навчанні є CNN, LSTM та BERT.

2.6.1 CNN

Згортова нейронна мережа (ЗНМ, англ. convolutional neural network, CNN) відрізняється від звичайної нейронної мережі тим, що вона працює з великим об'ємом вхідних даних і розроблена так, щоб вимагати використання мінімального обсягу попередньої обробки.

Вхідними даними CNN є матриця з фіксованою висотою n , кожний рядок

в якій є векторним відображенням токена в просторі ознак розмірності k . Для формування простору ознак часто застосовуються вкладання слів.

На першому етапі вхідна матриця обробляється шарами згортки. Згортка – це математична комбінація двох відносин для створення третьої. Вона об'єднує два набори інформації.

Як правило, фільтри мають фіксовану ширину, рівну розмірності простору ознак, а для підбору розмірів у фільтрів налаштовується тільки один параметр – висота h . Виходить, що h – це висота суміжних рядків, що розглядаються фільтром спільно. Відповідно, розмірність матриці ознак на виході для кожного фільтра варіюється в залежності від висоти цього фільтра h і висоти вихідної матриці n .

Далі карта ознак, отримана на виході кожного фільтра, обробляється шаром субдискретизації з певною функцією ущільнення, яка зменшує розмірність сформованої карти ознак. Так витягується найбільш важлива інформація для кожної згортки незалежно від її положення в тексті.

Після цього карти ознак, розраховані на виході кожного шару субдискретизації, об'єднуються в один загальний вектор ознак. Він подається на вхід прихованому повнозв'язному шару, а потім надходить на вихідний шар нейронної мережі, де і розраховуються підсумкові мітки класів [41].

Згорткові нейронні мережі знаходять своє застосування в розпізнаванні зображень та відео, системах рекомендації та обробці природньої мови, в тому числі в аналізі тональності текстів.

Приклад вдалого застосування даного методу для аналізу тональності новин ринку акцій можна побачити в статті [42].

2.6.2 RNN LSTM

Рекурентні нейронні мережі (PHM, англ. recurrent neural networks, RNN) – це клас штучних нейронних мереж, у яких з'єднання між вузлами створюють граф орієнтований у часі, завдяки чому створюється внутрішній стан мережі, що

дозволяє їй проявляти динамічну поведінку в часі [43].

В нейронних мережах цього типу використовується принцип, заснований на властивості думок накопичуватися і впливати одні на одних. Тому на відміну від простих нейронних мереж, РНМ використовують свою внутрішню пам'ять для обробки даних, які подаються на вхід. Для цього в них є цикли, які зберігають інформацію та дозволяють передавати її з одного кроку мережі в інший.

Рекурентні нейронні мережі використовують отриману раніше інформацію для вирішення подальших задач, але коли потрібно ще більше контексту (наприклад, якраз для визначення тональності речення), то розрив між необхідною для обліку інформацією і точкою, в якій вона потрібна, стає більшим і по мірі збільшення цього розриву РНМ втрачає зв'язок між інформацією.

Тому для таких ситуацій існує LSTM (long short-term memory) – нейронна мережа з довгою короткостроковою пам'яттю, це тип рекурентної нейронної мережі, яка здатна навчатися довгостроковим залежностям.

Стан комірки в LSTM є ключовим поняттям. Це ніби конвеєрна стрічка, що проходить через весь ланцюжок, отримуючи при цьому незначні лінійні перетворення.

В залежності від потреб, LSTM збільшує або зменшує кількість інформації в стані комірки. Для здійснення цього застосовуються структури, які не пропускають або пропускають інформацію.

Складовими цих структур є сигмовидний шар нейронної мережі та операції поточкового множення. На виході сигмовидного шару видаються числа від нуля до одиниці, визначаючи, який об'єм кожної одиниці інформації пропускати далі, де значення 0 означає «не пропускати нічого», а значення 1 – «пропускати все».

LSTM має три такі структури для контролю стану комірки: втрати, збереження та нового стану.

На етапі втрати вирішується, яка інформація зберігатиметься для обчислення стану комірки, а яка не має значення і може бути відкинута. Це рішення приймається сигмовидним шаром, званим «шаром втрати». Він отримує

на вхід h та x і для кожного номера в стані комірки C видає значення від 0 до 1, де 0 означає «видаляти повністю», а 1 – «зберігати повністю» (2.13):

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (2.13)$$

де h_{t-1} – інформація з попередньої комірки;

x_t – інформація з поточної комірки.

На етапі збереження вирішується, яка нова інформація зберігатиметься в стані комірки. Спочатку сигмовидний шар, що називається «шаром входу», вирішує, які значення потребують оновлення. Далі створюється вектор нових значень-кандидатів C шаром $\tanh$, які додаються до стану. Потім ці два значення об'єднуються для оновлення стану (2.14, 2.15):

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (2.14)$$

$$C'_t = \tanh(W_C[h_{t-1}, x_t] + b_C). \quad (2.15)$$

На етапі нового стану оновлюється попередній стан комірки для отримання нового стану C . Для цього попередній стан помножується на f і відкидається вже непотрібна інформація. Потім додаються $i * C$. Ці нові значення кандидатів, масштабовані в залежності від того, як вирішено оновлювати кожне значення стану (2.16):

$$C_t = f_t * C_{t-1} + i_t * C'_t. \quad (2.16)$$

Результатом на виході є відфільтрований стан комірки. Спочатку запускається сигмовидний шар, який вирішує, які частини стану комірки виводити. Потім стан комірки пропускається через $\tanh$ (щоб розмістити всі

значення в інтервалі $[-1, 1]$) і помножується на вихідний сигнал сигмовидного шару (2.17, 2.18):

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (2.17)$$

$$h_t = o_t * \tanh(C_t). \quad (2.18)$$

LSTM є потужним інструментом для прогнозування часових рядів, розпізнавання мовлення та рукописного тексту, навчання граматики, аналізу тональності текстів [43]. Приклад вдалого застосування даного методу для аналізу тональності новин ринку акцій можна побачити в статтях [42], [44].

2.6.3 BERT

BERT (англ. Bidirectional Encoder Representations from Transformers, двоспрямовані кодувальні представлення з Трансформерів) – нейромережева модель-трансформер (модель глибинного навчання, яка переймає механізм уваги, роздільно зважуючи важливість кожної частини даних входу).

Розробник моделі Google оголосила, що BERT використовується в якості основної частини їх пошукового алгоритму для кращого розуміння запитів на різних мовах.

За замовчуванням BERT можна застосовувати для попереднього тренування обробки природньої мови, створення вкладень слів, але крім цього, модель також навчена на двох завданнях без вчителя – моделювання мовних масок і передбачення наступного речення, що в свою чергу дозволяє використовувати попередньо навчену модель BERT, налаштовуючи її під конкретні завдання, в тому числі і для визначення емоційного забарвлення тексту.

Новизна моделі полягає в способі попереднього навчання моделі. Більш ранні архітектури (а саме односпрямовані), щоб навчитися, виконували завдання

генерації тексту – тобто передбачали, яке слово найімовірніше буде стояти наступним, враховуючи всі слова до нього і на рішення нейромережі впливали тільки слова.

Більш розвинуті двонаправлені нейромережі для навчання працюють паралельно, при цьому одна передбачає слова зліва направо, а інша – справа наліво, і потім результат обох мереж об'єднується (наприклад, ця ідея лежить в основі моделі ELMo). А попереднє навчання BERT проводиться на «маскованій мовній моделі», яка полягає в тому, що потрібно передбачити слово не в кінці речення, а десь посередині. Маскованою модель називається тому, що шуканий токен замінюється токеном [MASK].

Такий підхід до навчання дозволив зробити те, чого не можна було зробити із застосуванням ранніх методів: подати на вхід (і врахувати) всі частини фрази, а не тільки слова зліва чи справа. З цим новим методом навчання досягається «глибока двоспрямованість», коли модель насправді дивиться в обидві сторони, а не об'єднує окремі уявлення про контекст зліва і справа.

У випадку проведення аналізу тональності з використанням BERT додається шар класифікації над виходом трансформатора, щоб отримати мітку класу CLS, який модель навчилася передбачати в процесі попереднього навчання. Як і зазвичай, попередньою обробкою є токенизація тексту на вході та створення вкладень слів.

Технологія створення контекстуалізованих вкладень слів трансформера BERT є його найбільшою перевагою, але також і найбільшим недоліком, оскільки воно потребує ще більших ресурсів для обчислень [45].

Тим не менш, завдяки якісному вивченню двостороннього контексту, BERT допомагає моделям краще розуміти багатозначні слова, що є важливим для задач обробки природньої мови, в тому числі і для проведення аналізу тональності новин ринку акцій [46], [47].

2.7 Порівняння розглянутих методів

Загальне порівняння розглянутих методів представлено в таблиці 2.2.

Таблиця 2.2 – Загальне порівняння методів аналізу тональності

Підхід	Переваги	Недоліки
Метод, заснований на словнику	простота у застосуванні; мінімальні вимоги до обчислювальних ресурсів	трудомісткість процесу створення словника; неможливість обробляти заперечення, граматичні та друкарські помилки в реченнях; ігнорує контекст
Метод, заснований на правилах	задовільна точність при наявності хорошої бази правил	трудомісткість процесу створення набору правил
Наївний баєс	простота у розробці та тестуванні; висока швидкість тренування; помірні вимоги до пам'яті	дуже залежить від заданих на вхід даних
Метод опорних векторів	потребує досить невеликого набору даних	не працює, якщо класи лінійно нероздільні; помилка в тренувальному наборі даних дуже відображається на результатах
Метод випадкового лісу	ефективна обробка даних з великою кількістю ознак; висока масштабованість	схильність до перенавчання; великий розмір моделей
FastText класифікатор	простота у застосуванні; мінімальні вимоги до обчислювальних ресурсів; точність результатів на рівні моделей глибокого навчання	помилка в тренувальному наборі даних може відобразитися на результатах

Продовження таблиці 2.2.

Підхід	Переваги	Недоліки
CNN	має потенціал видавати дуже точні результати; звертає увагу на контекст тексту	для якісного навчання вимагає великого тренувального набору даних, багато часу та обчислювальних ресурсів
RNN LSTM		
BERT		
Машинне навчання без вчителя	не потребує попередньо розмічених даних; може помічати якісь приховані шаблони, що можуть в подальшому мати велике значення	для якісного навчання вимагає великого обсягу даних, багато часу та обчислювальних ресурсів; часто менша точність результатів

Враховуючи те, що метод FastText є продуктивним та не вимагає багато часу, великого набору даних та багато обчислювальних ресурсів для навчання моделі і при цьому забезпечує якість результатів, яка є на одному рівні з якістю результатів моделей глибинного навчання, було вирішено обрати його застосування для створення вкладень слів та навчання моделі класифікації при реалізації методу визначення тенденцій зміни ціни акцій

Розглянуті методи, засновані на машинному навчанні (наївний баєс, метод опорних векторів, метод випадкового лісу) та методи, засновані на глибинному навчанні (CNN та RNN LSTM) можуть застосовуватися і для прогнозування майбутніх значень на основі минулих даних, в тому числі і для прогнозування подальшої тенденції зміни цін акцій, як можна побачити в дослідженнях [44], [48], [49].

В залежності від наборів даних, які використовуються, та від налаштувань моделей, кожний метод по-своєму добре може себе показати. Але загалом метод глибинного навчання LSTM є найбільш популярним для вирішення цієї задачі (прогнозування часових рядів), адже показує якісні результати при опрацюванні

нестационарних даних – даних, статистичні властивості яких (наприклад, середнє значення і стандартне відхилення) не є постійними в часі, а змінюються з часом.

Моделі LSTM здатні зберігати інформацію за певний період часу, інакше кажучи, мають ємність пам'яті, а це дуже корисно при аналізі послідовних даних. Тому застосування саме LSTM моделі, як моделі прогнозування, було обрано для реалізації методу визначення тенденцій зміни ціни акцій.

2.8 Запропонований метод визначення тенденцій зміни цін акцій

На рисунку 2.2 представлено схему роботи запропонованого методу.

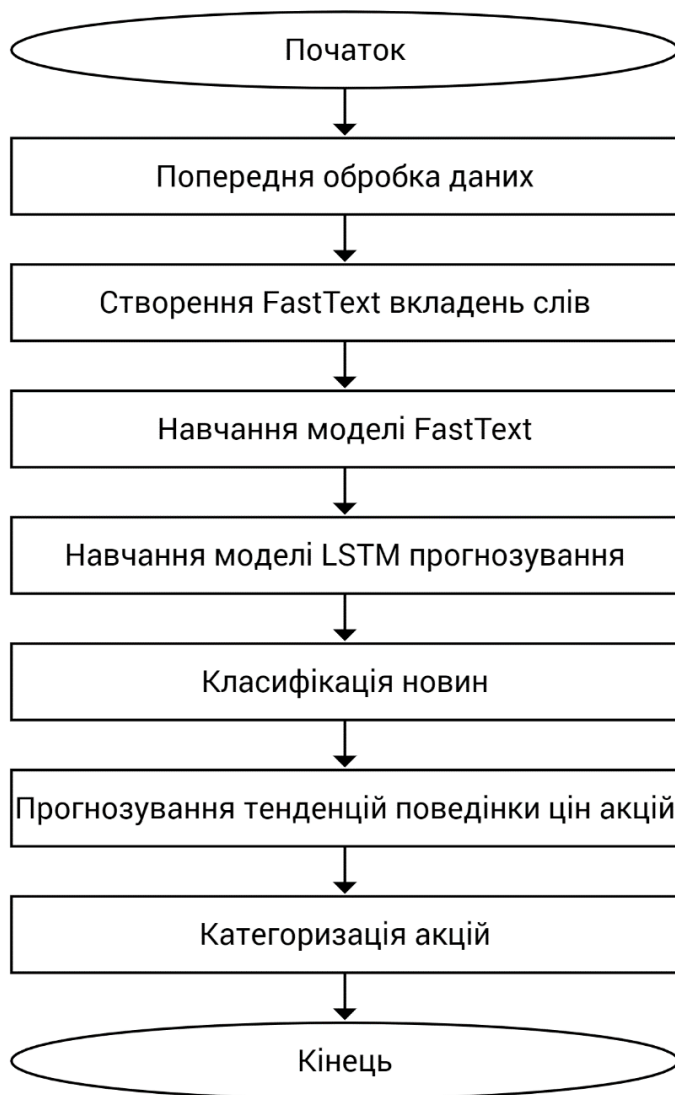


Рисунок 2.2 – Метод визначення тенденцій зміни цін акцій

Спочатку подані на вхід дані для навчання моделі FastText класифікації текстів попередньо обробляються в процесі виконання відповідних кроків, зазначених у пункті 1.3.1. А для приведення слів до єдиного представлення використовується саме лематизація, адже при цьому перетворенні враховується контекст і слово перетворюється на його значущу базову форму.

На рисунку 2.3 представлено CSV-файл з навчальним розміченим набором даних, який обробляється.

```
> training > data.csv
1 __label__,text
2 negative,"The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility
3 positive,With the new production plant the company would increase its capacity to meet the expected increase in demand ar
4 positive,"According to the company 's updated strategy for the years 2009-2012 , Basware targets a long-term net sales gr
5 positive,FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is aggressively pursuing its growth strategy by increasingly focusing c
6 positive,"For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same period a yea
7 positive,"In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5 mn , and operating profit by 34.9 % to
8 positive,Operating profit rose to EUR 13.1 mn from EUR 8.7 mn in the corresponding period in 2007 representing 7.7 % of r
9 positive,"Operating profit totalled EUR 21.1 mn , up from EUR 18.6 mn in 2007 , representing 9.7 % of net sales ."
10 positive,TeliaSonera TLSN said the offer is in line with its strategy to increase its ownership in core business holdings
11 positive,"STORA ENSO , NORSKE SKOG , M-REAL , UPM-KYMMENE Credit Suisse First Boston ( CFSB ) raised the fair value for s
12 positive,"A purchase agreement for 7,200 tons of gasoline with delivery at the Hamina terminal , Finland , was signed wit
13 positive,"Finnish Talentum reports its operating profit increased to EUR 20.5 mn in 2005 from EUR 9.3 mn in 2004 , and ne
```

Рисунок 2.3 – Файл з навчальними даними

На рисунку 2.4 представлено текстовий файл, який отримується в результаті обробки.

```
app > training > train.txt
272 __label__positive cost saving ronweader group currently process implement streamline restructure measure r
273 __label__negative restriction apply subject payment condition term individual instrument payment 2010 fisc
274 __label__positive nordea see return positive growth baltic country 2011
275 __label__positive adp news nov 3 2008 finnish cargo handling system service company cargotec oyj omx c
276 __label__negative rupee drop 82 paise 7450 dollar coronavirus spook market
277 __label__negative change compare period previous year free cash flow benefit strong inflow customer adva
278 __label__negative result negative extraordinary item amount around chr 13 million push earning red expec
279 __label__negative much top action short key 890 short cat 95 stop 956 ike trade ot
280 __label__negative effectiveness withdrawal appeal still require acceptance court
281 __label__negative employment contract 14 employee end march 2009 result company previous employeremployee
282 __label__positive truly take efficiency new height mr metso add
283 __label__negative hnz another american institution sell foreign interest sa gutte buffett 0 management p
```

Рисунок 2.4 – Файл з обробленими навчальними даними

Далі для навчання моделі FastText класифікації їй на вхід подається текстовий файл з обробленими заголовками новин ринку акцій та мітками виду `__label__<class>`, які показують до якого класу відноситься відповідний текст.

Для найбільш можливого якісного навчання моделі були підібрані такі гіперпараметри:

- кількість n-грам – 3 (безпосередньо є налаштуванням для створення вкладень слів);
- кількість епох (кількість повторів навчання моделі) – 50;
- швидкість навчання (наскільки сильно змінюється модель) – 0.5;
- функція втрат – softmax (нормована експоненційна функція).

Навчання виконується за алгоритмом FastText для аналізу тональності (п. 2.4.4), в результаті роботи якого отримується навчена модель класифікації.

На етапі навчання моделі LSTM прогнозування їй на вхід подається Pandas структура даних зі стовпцями date з датами днів проаналізованих новин та probability з середніми значеннями оцінок тональності новин за кожний день (рис. 2.5).

1	date,probability
2	2021-07-20,0.0915067978203296
3	2021-07-21,0.4109258446842432
4	2021-07-22,0.022047879234437
5	2021-07-23,0.4122969632347424
6	2021-07-24,0.3385294675827026
7	2021-07-25,-0.7673095166683197
8	2021-07-26,0.1886839008912807
9	2021-07-27,0.0402067347270686
10	2021-07-28,0.1157297106350169
11	2021-07-29,0.4583203338441394
12	2021-07-30,0.2900192870034112
13	2021-07-31,0.2159851619175502
14	2021-08-01,-0.5475750863552094
15	2021-08-02,0.3781603760406619
16	2021-08-03,0.2448739798532592
17	2021-08-04,0.2071606507668128
18	2021-08-05,0.2974852224191029
19	2021-08-06,0.4120411760402175
20	2021-08-07,0.3692331314086914
21	2021-08-08,-0.1425388385267818
22	2021-08-09,0.2398634219453448

Рисунок 2.5 – Вхідні дані для LSTM моделі

Для найбільш можливого якісного навчання моделі були підібрані такі гіперпараметри:

- кількість комірок: на вході – 100, на виході – 1;

- кількість часових кроків – 7;
- кількість епох – 200;
- функція активації – $\tanh$ (функція гіперболічного тангенса).

Кількість часових кроків було встановлено саме таким чином, щоб модель аналізувала послідовності з 7 кроків, тобто використовувала дані за останні 7 днів, щоб прогнозувати подальше значення. При використанні більшої або меншої кількості кроків прогнози ставали б нестабільними.

Навчання моделі виконується за алгоритмом LSTM (п. 2.6.2), в результаті роботи якого отримується навчена модель прогнозування.

На етапі класифікації зібрані новини передаються до навченої FastText моделі. Модель визначає тональності новин, підраховує відповідні оцінки від -1 до 1, де -1 показує, що текст має найбільш негативний окрас, а оцінка 1 – відповідно найбільш позитивний. В кінці середні значення оцінок тональності новин за кожний день зберігаються.

Далі кожна навчена на послідовностях середніх значень оцінок тональності новин компаній за попередні дати LSTM модель повертає числове значення, яке стає основою для прогнозування тенденції зміни цін акцій відповідної компанії в наступні 7 днів.

Останній етап – категоризація акцій – полягає у їх розподілі на категорії:

- незначне падіння ціни (slight drop);
- значне падіння ціни (significant drop);
- ціна майже не зміниться (almost intact);
- незначне зростання ціни (slight rise);
- значне зростання ціни (significant rise).

Нехай S – середнє значення оцінок тональності за перший з 7 днів, які аналізуються, а P – числове значення прогнозу LSTM моделі на наступні 7 днів.

Підраховується різниця $D = P - S$ і якщо:

- $D \leq -0.7$ – визначається, що ціна може значно впасти;
- $-0.2 < D < -0.7$ – визначається, що ціна може впасти незначно;
- $-0.2 \leq D \leq 0.2$ – визначається, що ціна може майже не змінитись;

- $0.2 < D < 0.7$ – визначається, що ціна може зрости незначно;
- $D \geq 0.7$ – визначається, що ціна може значно зрости.

На рисунку 2.6 представлено схематично як відбувається категоризація акцій за тенденціями зміни їх цін у майбутньому.

-1			-0.7					-0.2				0.2					0.7			1
significant drop				slight drop				almost intact				slight rise				significant rise				

Рисунок 2.6 – Категоризація акцій

Таким чином, на основі результатів категоризації можуть бути прийняті рішення щодо купівлі або продажу певних акцій.

2.9 Висновки з розділу 2

У другому розділі було розглянуто методи аналізу тональності, засновані на словниках та на правилах, методи машинного навчання з вчителем (наївний баєсів класифікатор, метод опорних векторів, метод випадкового лісу, FastText класифікатор), метод машинного навчання без вчителя та методи глибинного навчання (CNN, RNN LSTM, BERT), які можна застосувати для аналізу тональності текстових повідомлень.

Було запропоновано власний метод визначення тенденцій зміни цін акцій, який використовує результати аналізу тональності заголовків новин класифікатором FastText та результати LSTM моделі прогнозування тенденцій зміни ціни акцій для визначення категорії кожної акції відповідно до тенденції зміни її ціни, що може стати основою для прийняття рішень щодо купівлі або продажу акцій.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

Разом зі стрімким розвитком науки про дані не відстають і мови програмування, з'являються та удосконалюються зручні у користуванні бібліотеки для роботи з даними. Тож вирішення задачі аналізу тональності тексту є доступним. Але також важливим є доцільний підбір мови програмування та супутніх інструментів, які б задовольнили потреби при розробці системи.

3.1 Вибір мови програмування

Найбільш підходящими мовами програмування для системи, що розроблюється, є ті, які пов'язані з наукою про дані. До таких можна віднести Python, R, MATLAB, JavaScript, Java. В таблиці 3.1 проведено порівняння цих мов програмування, враховуючи потреби під час розробки проєкту.

Таблиця 3.1 – Порівняльна таблиця мов програмування

Мова програмування / Критерії	Python	R	MATLAB	JavaScript	Java
Активна спільнота	+	—	—	+	+
Докладна документація	+	+	+	+	+
Зручні засоби для аналізу даних	+	—	—	+/-	+/-
Зручні засоби для візуалізації аналізу даних	+	+/-	+/-	+	—
Бібліотеки для науки про дані	+	+	+	+	+
Зручні засоби для аналізу тональності текстів	+	—	+/-	+	+/-

Отже, вибір мови програмування для розробки системи, що розглядається, не є проблемою, адже розглянуті популярні мови програмування надають потужні засоби для аналізу та візуалізації даних. Крім цього,

враховуючи що система, яка розроблюється, є вебзастосунком, перевагу мають мови програмування, які, крім засобів для роботи з даними, надають ще й засоби для веброзробки.

Таким чином, враховуючи розглянуті варіанти було обрано Python як мову програмування для розробки системи для проведення аналізу тональності текстових повідомлень про ринок акцій для дослідження ситуації на ринку та обґрунтування інвестиційних рішень.

Python – це мова програмування високого рівня, орієнтована на підвищення продуктивності розробника і читаності коду [50]. Python є мовою загального призначення і може використовуватися в різних сферах розробки, але найбільшої поширеності набув у науці про дані та веброзробці.

3.2 Вибір засобів Python для роботи з даними

Для обробки даних було обрано бібліотеку Pandas – це інструмент з відкритим вихідним кодом, який надає високопродуктивні, прості у використанні структури даних та інструменти аналізу та обробки даних.

Бібліотека призначена для швидкого та легкого маніпулювання даними, зчитування, зведення та візуалізації. Pandas може зчитувати дані з файлів форматів CSV та TSV або з бази даних SQL і створювати об'єкти Python, які являють собою двовимірні структури даних із рядками та стовпцями, які дуже схожі на таблиці в статистичному програмному забезпеченні (наприклад, Excel).

Крім Pandas, справді популярним є PySpark – дуже просунутий інструмент, який відмінно підходить для обробки величезних масивів даних, відомий своєю швидкістю та можливістю інтеграції з різними файловими системами (як Hadoop HDFS, AWS S3).

Але для системи, що розроблюється наданих бібліотекою Pandas можливостей виявляється більше, ніж достатньо, тому використання PySpark було б зайвим.

До популярних Python засобів для візуалізації даних належать Matplotlib,

Seaborn та Plotly.

Matplotlib – це бібліотека, яка надає об’єктно-орієнтований API для візуалізації даних у вигляді 2D-фігур. Matplotlib може зобразити широкий діапазон візуалізацій: лінійні діаграми, гістограми, кругові діаграми, діаграми розсіювання, спектрограми, ізолінії і т. д.

Seaborn є розширенням Matplotlib, яке має різноманітні шаблони візуалізації з менш складним синтаксисом. Використовуючи Seaborn можна визначати кореляції, проаналізовувати одновимірні чи двовимірні розподіли та порівнювати їх між різними підмножинами даних, будувати моделі лінійної регресії для залежних змінних і т. д.

Plotly – це просунута бібліотека для побудови графіків, які при цьому можна робити інтерактивними (наприклад, змінювати масштаб перегляду графіків, відображати деталі при наведенні на фігури). Як і Matplotlib, Plotly також надає широкий діапазон візуалізацій, але крім тих, що можна побудувати засобами Matplotlib, використовуючи Plotly можна будувати більш специфічні статистичні та наукові графіки та діаграми [51].

Для візуалізації даних було обрано саме Plotly, адже за допомогою цієї бібліотеки можна робити інтерактивні графіки та гістограми з невеликою кількістю коду.

3.3 Вибір засобів Python для реалізації класифікації та прогнозування

Для реалізації класифікації текстів за їх тональностями обраним методом FastText існує однойменна Python бібліотека, яка, як і метод (п. 2.4.4), є розробкою команди Facebook.

Бібліотека FastText є безкоштовною, легковісною та з відкритим вихідним кодом. Вона надає попередньо навчені вкладання слів та класифікатори тексту.

Бібліотека надає вкладання слів англійської мови і ще для 157 мов, які створені на базі великої кількості сторінок Вікіпедії та кількох порталів новин, а

також попередньо навчені моделі для класифікації та розпізнавання мови. Не є вимогливою до обчислювальних ресурсів

Засобами FastText можна швидко провести тренування моделі аналізу тональності текстових повідомлень на своєму наборі даних, а потім кардинально зменшити розмір моделі [52].

Тож враховуючи доступні обчислювальні ресурси при розробці системи, було вирішено обрати саме цю бібліотеку для реалізації аналізу тональності текстових повідомлень про ринок акцій.

До засобів Python, які можна застосувати для реалізації прогнозування тенденцій зміни цін акцій обраним методом LSTM, відносяться бібліотеки Keras та PyTorch.

Keras – це відкрита Python-бібліотека, яка забезпечує взаємодію зі штучними нейронними мережами, містить численні втілення часто використовуваних будівельних блоків для нейромереж (цільові функції, передавальні функції, шари, оптимізатори та інші інструменти для зручної обробки тексту із застосуванням нейромереж).

Крім стандартних нейронних мереж, Keras підтримує згорткові та рекурентні нейронні мережі, в тому числі і LSTM [53].

PyTorch – це відкрита Python бібліотека машинного та глибинного навчання. PyTorch перепроєктує і впроваджує Torch (бібліотека для машинного навчання, система для наукових обчислень та мова сценаріїв на основі мови програмування Lua) в Python, спільно використовуючи ті ж основні бібліотеки C для внутрішнього коду.

Розробники PyTorch налаштували цей внутрішній код для ефективної роботи з Python. Вони також зберегли апаратне прискорення на основі графічного процесора, а також функції розширюваності.

Як і Keras, PyTorch можна використовувати для розробки моделі LSTM прогнозування [54].

Якщо порівнювати ці дві бібліотеки, то Keras має API високого рівня, просту архітектуру, яка дозволяє швидко створювати, навчати й оцінювати

моделі, в основному використовується для невеликих наборів даних через його повільну швидкість. А PyTorch має низькорівневе API, складну архітектуру, кращу швидкість і можливості тонкого налагодження.

Обидві бібліотеки зручні у використанні і забезпечують потрібними засобами для вирішення поставленої задачі, та враховуючи задовільну потужність Keras при його простій архітектурі, для системи, що розроблюється, було обрано саме цю бібліотеку для створення LSTM моделі прогнозування тенденцій зміни ціни акцій, результати якої застосовуватимуться для категоризації акцій.

3.4 Вибір вебфреймворку

Вебфреймворки широко застосовуються при розробці більшості сучасних вебзастосунків. При використанні цих інструментів процес проєктування архітектури, розробка та подальша підтримка вебзастосунку спрощується.

До популярних вебфреймворків на Python можна віднести: Django, Flask, Tornado, Web2py, Pyramid, Turbo Gears, Dash. Але вже довгий час лідерами є Django та Flask.

Django – це повноцінний фул-стек вебфреймворк з відкритим сирцевим кодом. Цей інструмент надає розробникам:

- сервер для розробки;
- готові архітектурні рішення (архітектура Django схожа на «Модель-Представлення-Контролер» (Model-View-Controller, MVC));
- заздалегідь підготовлені для використання компоненти, які є стандартними для вебзастосунків (аутентифікація, адмін-панелі, контактні форми, підтримка завантаження файлів та ін.);
- URL-маршрутизацію;
- Django ORM як зручний інструмент для роботи з базами даних;
- розширювану систему шаблонів;
- інструменти для зручної роботи з формами.

Вебфреймворк Django використовується в таких відомих вебсайтах, як: lamoda, Instagram, The Washington Times, Mozilla, Pinterest.

Flask – це мікрофреймворк, який надає базові можливості для розробки вебзастосунків.

Flask не обмежує розробника своїми рішеннями у проєктуванні архітектури системи та залишає вибір потрібних інструментів для розробки на бажання розробника.

Завдяки активній спільноті Flask з'являється багато розширень для фреймворка, які поліпшують інструмент та допомагають розробникам додавати нову функціональність розроблюваним вебзастосункам.

Крім цього, Flask надає сервер для розробки, відлагоджувач, підтримку модульних тестів і безпечних куків.

Flask використовується такими компаніями, як: Airbnb, Reddit, Patreon, Netflix, Uber.

І Django, і Flask мають вбудований сервер, зручні утиліти командного рядка, інструменти для тестування. Також обидва фреймворки забезпечені зрозумілою документацією та активною спільнотою.

Обидва вебфреймворки надають зручну функціональність для розробки вебзастосунків, але мають свої відмінності при наданні потрібних інструментів, наприклад:

- Django надає власний ORM для керування базою даних (БД), а Flask навпаки в себе не включає ніяких модулів для цього, залишаючи керування БД на розсуд розробника;

- Django надає вбудоване рішення для аутентифікації, а Flask – ні, але є варіанти для розробки такої функціональності (наприклад, використання безпечних куків);

- Django надає інструменти для обробки форм, а Flask у «чистому вигляді» – ні, але для цього можна скористатися розширенням WTForms;

- Django надає панель адміністратора, а Flask – ні, але щоб додати таку функціональність можна скористатись розширенням Flask-Admin [55].

Таким чином, вибір між Django та Flask залежить від потреб під час розробки, спроектованої архітектури системи та поставлених до неї задач. Тож враховуючи, що просунуті інструменти для роботи з формами, шаблонами, адмін-панелі та інші засоби вебфреймворку Django при розробці системи були б зайвими, як вебфреймворк було обрано Flask.

3.5 Вибір бази даних

Так як робота з даними в системі, що розроблюється, організована з використанням бібліотеки Pandas, для зручного зберігання даних доцільно використовувати нереляційну базу даних, модель якої базується на форматі JSON, адже створені засобами Pandas структури даних легко конвертувати в цей формат.

Крім цього, дані, які зберігатимуться, не є об'єктами реального світу, як зазвичай у реляційних базах даних. І немає потреби проєктувати строгу структуру зберігання даних, бо від самого початку вона визначається засобами Pandas. Нереляційна база даних повинна бути оптимізована для продуктивності та масштабованості.

До популярних нереляційних документоорієнтованих баз даних належать CouchDB, DynamoDB та MongoDB.

CouchDB – це нереляційна документо-орієнтована база даних із відкритим кодом, яка для зберігання полів документів застосовує карти ключових значень.

Поля можуть бути як простими парами «ключ-значення», так і картами або списками. Кожний документ, який зберігається, отримує унікальний ідентифікатор, а кожний документ, в який вносяться зміни – номер редагування.

База даних CouchDB здатна здійснювати перетворення документів на ходу та в режимі реального часу оповіщати про зміни.

До особливостей CouchDB відносяться:

- JSON-формат документів;

- бібліотеки для різних мов;
- висока швидкість індексування;
- проста реплікація через декілька екземплярів сервера;
- REST-подібний інтерфейс для вставки, оновлення, видалення та пошуку документів [56].

DynamoDB – це NoSQL база даних, запропонована компанією Amazon Web Services (AWS).

До особливостей DynamoDB відносяться:

- забезпечує надійною роботою навіть під час масштабування;
- не потребує налаштувань серверів;
- зашифровує всі дані за замовчуванням і забезпечує детальну ідентифікацію та управління доступом до всіх таблиць;
- надає невеликий, простий API, що забезпечує простий доступ до даних виду «ключ-значення», а також більш вдосконалені шаблони запитів [57].

MongoDB – це крос-платформна, потужна, гнучка, документно-орієнтована база даних, що легко масштабується, яка використовує JSON як формат даних та ідеально підходить для зберігання даних, керування вмістом та кешування.

Однією з ключових особливостей MongoDB є те, що ця база даних використовує формат BSON (формат для зберігання даних, подібний до JSON). BSON означає Binary JSON, що є двійковою кодованою серіалізацією JSON-подібних документів, які MongoDB використовує при зберіганні документів у колекціях.

MongoDB використовує свій метод розподілу даних на декількох машинах для підтримки розгортання з дуже великими наборами даних та високою пропускнуою здатністю. Це дає можливість забезпечити масштабування по горизонталі.

MongoDB має власний Python драйвер PyMongo і команду інженерів, які працюють над тим, щоб драйвер відповідав потребам спільноти Python, забезпечуючи бездоганну роботу MongoDB з Python [58].

Всі розглянуті бази даних є сучасними, не вимагають багато часу на встановлення та налаштування, враховуючи величезну кількість навчального матеріалу до кожної з технологій.

Враховуючи популярність, потужність, простоту інтеграції MongoDB з проектом на Python та зручність інтерфейс застосунку керування базою даних MongoDB Compass для системи, що розроблюється, було вирішено обрати саме MongoDB, а PyMongo для зручної взаємодії з базою даних при розробці [59].

3.6 Структура програми

На рисунку 3.1 представлено файлову структуру системи, що розроблюється.

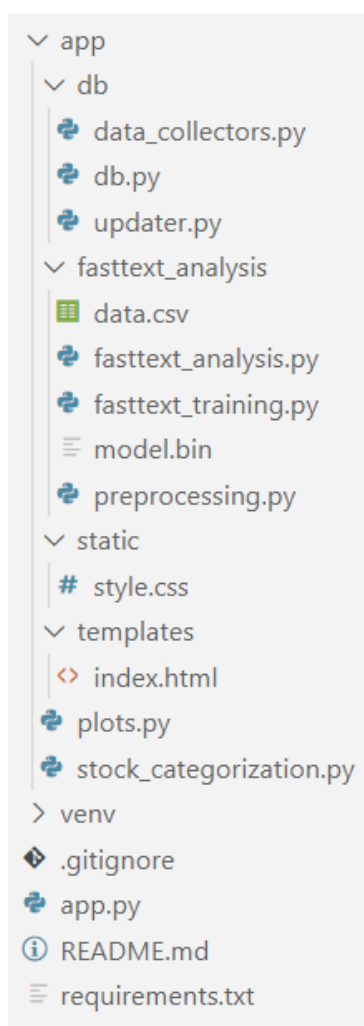


Рисунок 3.1 – Файлова структура проєкту

Каталог `app` містить у собі файли та пакети з модулями системи, що розроблюється.

Пакет `db` містить у собі модулі для збирання, оновлення та зберігання даних у базу даних:

- `data_collectors.py` – модуль, що містить функції збирання заголовків фінансових новин та історичних цін акцій;

- `db.py` – модуль з визначенням PyMongo клієнту бази даних та функцією її оновлення даними з `updater.py`;

- `updater.py` – модуль з функцією оновлення даних, яка об'єднує дані, отримані з `data_collectors.py` та `fasttext_analysis.py`, і передає їх в `db.py`.

Пакет `fasttext_analysis` містить у собі модулі для тренування моделі аналізу тональності та її подальшого застосування:

- `preprocessing.py` – модуль для попередньої обробки даних;

- `fasttext_training.py` – модуль, що містить функцію тренування моделі аналізу тональності;

- `fasttext_analysis.py` – модуль, що містить функцію аналізу тональності текстових повідомлень.

Крім модулів, цей пакет містить файл з розширенням `CSV` з набором даних для тренування моделі та файл натренованої моделі `model.bin`, яка використовується при аналізі тональності заголовків новин ринку акцій.

Модуль `stock_categorization.py` містить у собі функції для навчання LSTM моделі прогнозування тенденцій зміни ціни акцій, результати якої застосовуються для категоризації акцій.

Модуль `plots.py` містить у собі функції будування графіків цін акцій та оцінок тональності новин.

Каталог `static` містить у собі файл стилів `style.css`, а каталог `templates` – HTML-шаблон сторінки вебзастосунку `index.html`.

Каталог `venv` – це службовий каталог з віртуальним середовищем проєкту, який містить в собі інтерпретатор Python та встановлені бібліотеки, які використовуються у розробці.

А до решти файлів відносяться:

- модуль `app.py` – головний файл проекту, який відповідає за запуск всього застосунку та має Flask маршрути;
- `readme.md` – файл для загального опису проекту;
- `requirements.txt` – файл, який містить список потрібних для роботи проекту бібліотек та інструментів.

3.7 База даних

База даних MongoDB складається з колекцій. Кожна колекція має своє унікальне ім'я – довільний ідентифікатор, що складається з не більше ніж 128 різних алфавітно-цифрових символів і знака підкреслення. На відміну від реляційних баз даних, MongoDB не використовує табличну організацію даних з чітко заданою кількістю стовпців і типів даних.

MongoDB є документо-орієнтованою системою, в якій центральним поняттям є документ. При додаванні документа в колекцію для нього автоматично визначається унікальний ідентифікатор.

Отримані оцінки тональностей новин ринку акцій зберігаються в колекції `stock_sentiment_db` в документах з назвами виду `найменування_компанії_sentiment`.

Дати днів проаналізованих новин зберігаються у поля типу `Date` під назвою `date`, а оцінки тональностей зберігаються у поля типу `Double` під назвою `probability` (рис. 3.2).

	☒ <code>date</code> Date ▼	☒ <code>probability</code> Double ▼
1	2021-07-16	0.1034038811922073
2	2021-07-17	0.5146479407946268
3	2021-07-18	0.0201036880413691

Рисунок 3.2 – Визначення типів полів документів колекції `stock_sentiment_db`

На рисунку 3.3 представлено екранну копію інтерфейсу MongoDB Compass з відкритим документом збережених щоденних оцінок тональностей новин компанії Amazon.

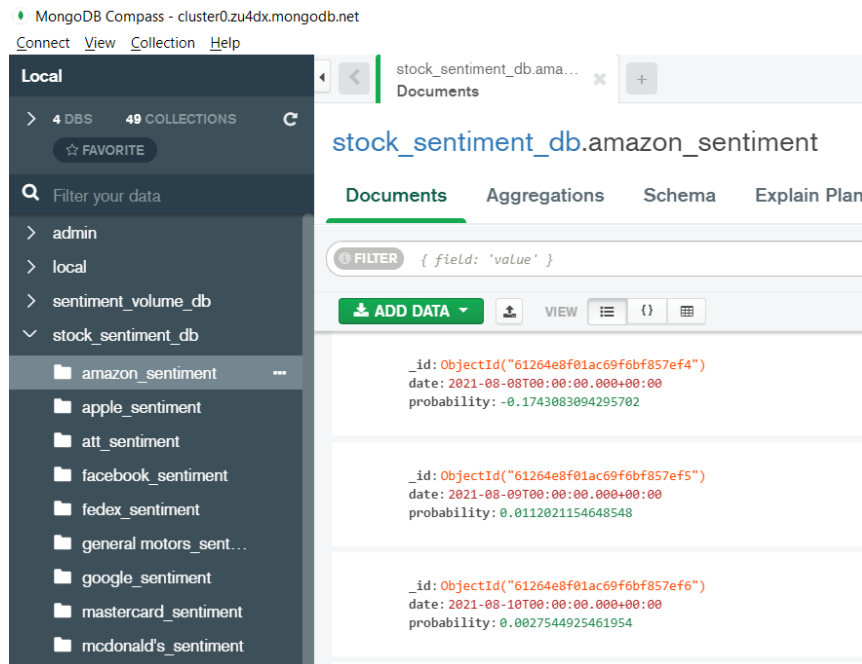


Рисунок 3.3 – Збережені оцінки тональностей новин компанії Amazon

Кількості проаналізованих новин, визначених як позитивні та негативні, зберігаються в колекції sentiment_volume_db в документах з назвами виду найменування_компанії_volumes.

Дати днів проаналізованих новин зберігаються у поля типу Date під назвою date, а кількості новин зберігаються у поля типу Int32 під назвами positive та negative відповідно (рис. 3.4).

	☒ date Date ▼	☒ positive Int32 ▼	☒ negative Int32 ▼
1	2021-07-16	14	10
2	2021-07-17	7	2
3	2021-07-18	6	6

Рисунок 3.4 – Визначення типів полів документів колекції sentiment_volume_db

На рисунку 3.5 представлено екранну копію MongoDB Compass з відкритим документом збережених кількостей проаналізованих фінансових новин, визначених як позитивні та негативні, компанії Facebook.

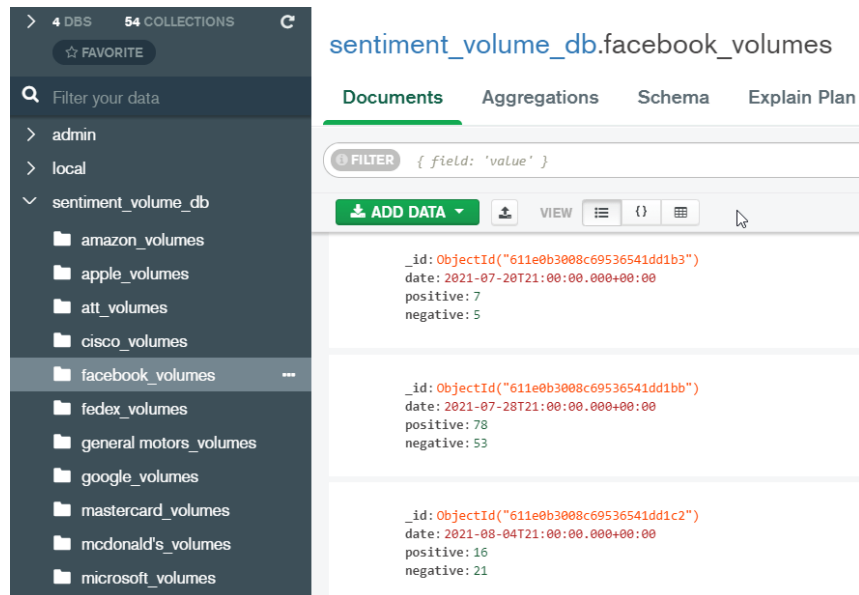


Рисунок 3.5 – Кількості проаналізованих фінансових новин компанії Facebook

Результати категоризації акцій (прогнози тенденцій зміни цін акцій) зберігаються в колекції `stock_categorization_db` в документі `stock_categorization`.

Назви компаній та прогнози зберігаються у поля типу String під назвами `company` та `forecast` відповідно (рис. 3.6).

	company String	forecast String
1	amazon	slight drop
2	apple	slight drop
3	att	slight rise
4	cisco	slight rise
5	facebook	significant drop
6	fedex	slight drop
7	general motors	slight rise
8	google	significant rise
9	mastercard	slight drop
10	mcdonald's	almost intact

Рисунок 3.6 – Визначення типів полів документів колекції `stock_categorization_db`

На рисунку 3.7 представлено екранну копію графічного інтерфейсу MongoDB Compass з відкритим документом збережених результатів категоризації акцій.

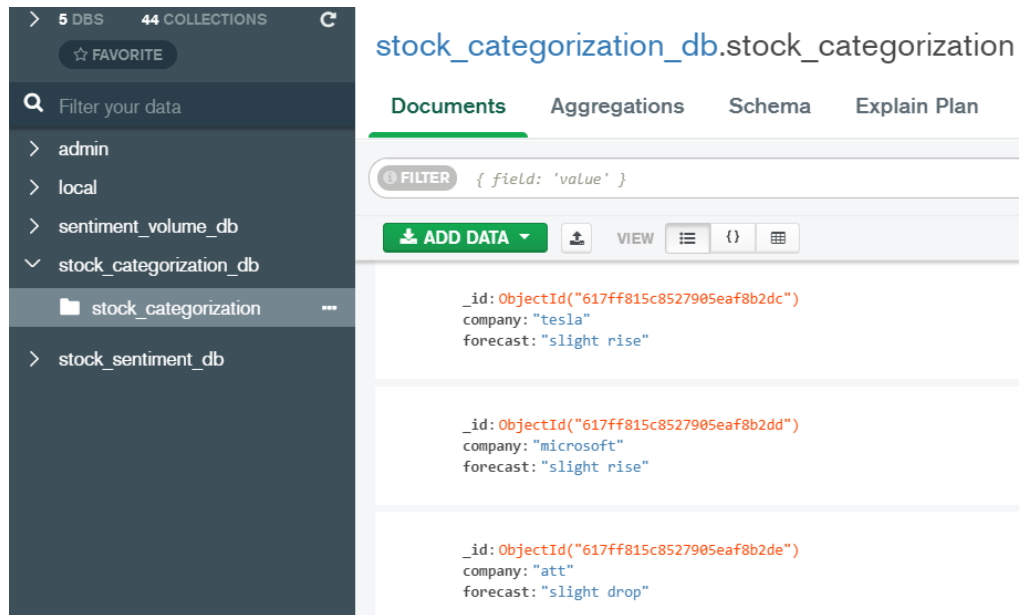


Рисунок 3.7 – Результати категоризації акцій

Таким чином, загальну схему використовуваної бази даних MongoDB можна представити як на рисунку 3.8.

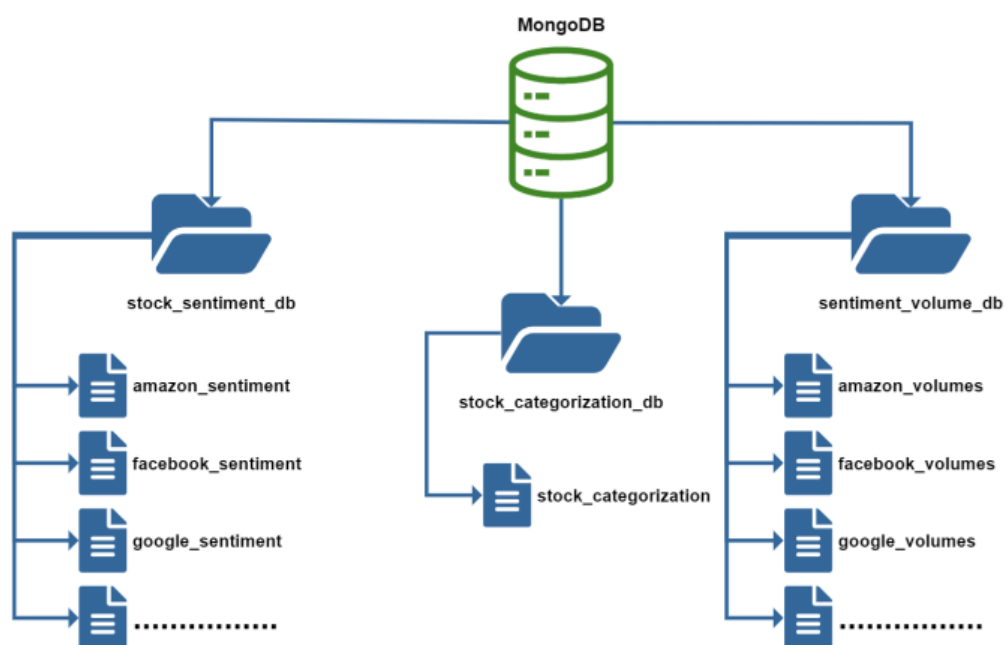


Рисунок 3.8 – Схема бази даних

3.8 Діаграма компонентів

На рисунку 3.9 зображено діаграму компонентів системи, що розроблюється.

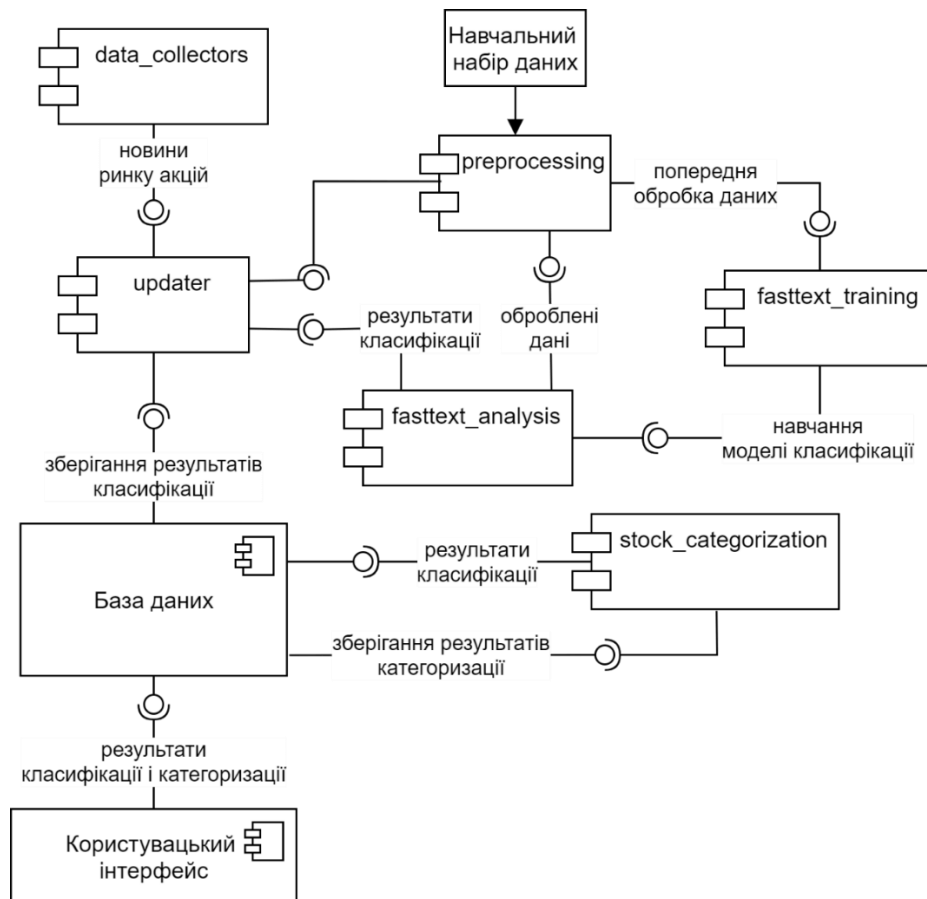


Рисунок 3.9 – Діаграма компонентів системи, що розроблюється

Тобто для навчання моделі класифікації розмічений набір даних попередньо обробляється в модулі `preprocessing.py` і потім використовується для навчання моделі в модулі `fasttext_training.py`. Натренована модель зберігається в файл `model.bin` та використовується при аналізі тональності в модулі `fasttext_analysis.py`.

Для проведення аналізу тональності новин ринку акцій спочатку функції з модулю `data_collectors.py` збирають заголовки фінансових новин з трьох призначених для цього API. Модуль `updater.py` звертається до `preprocessing.py` для підготовки даних і до `fasttext_analysis.py` для проведення аналізу тональності,

а потім зберігає отримані оцінки тональностей та підраховані кількості проаналізованих новин у базу даних за допомогою модуля db.py.

Функції з модулю stock_categorization.py завантажують оцінки тональностей з бази даних, використовують їх в LSTM моделі прогнозування тенденцій зміни цін акцій на ринку та категоризують акції. Результати категоризації зберігаються в базу даних.

Під час запуску головного файлу вебзастосунку app.py і відкритті сторінки застосунку в браузері завантажуються файл шаблону index.html зі стилями з файлу styles.css

Під час першого відкриття сторінки вебзастосунку та під час обирання режимів для перегляду результатів класифікації новин та категоризації акцій викликається функція з модулю plots.py, яка використовує збережені в MongoDB дані для побудови графіків.

3.9 Діаграми прецедентів

На рисунку 3.10 зображено діаграму прецедентів, яка показує варіанти використання системи користувачем (інвестором або трейдером).

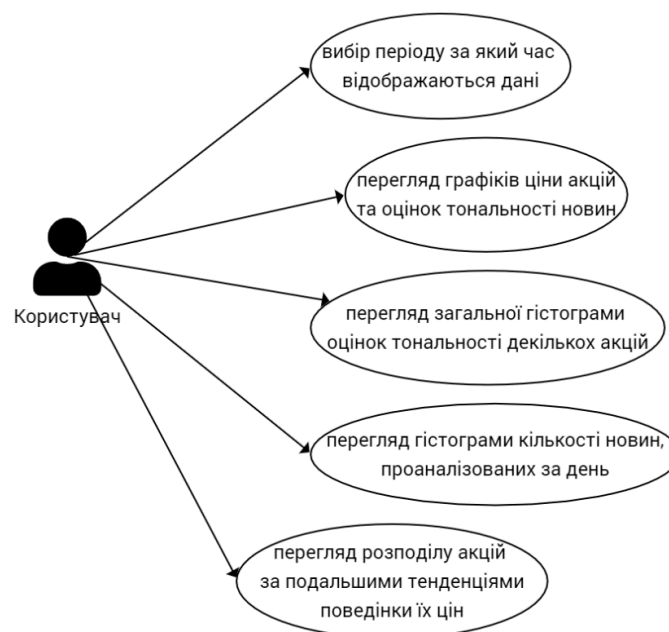


Рисунок 3.10 – Діаграма прецедентів користувача

На рисунку 3.11 зображено діаграму прецедентів, яка показує варіанти використання системи оператором.

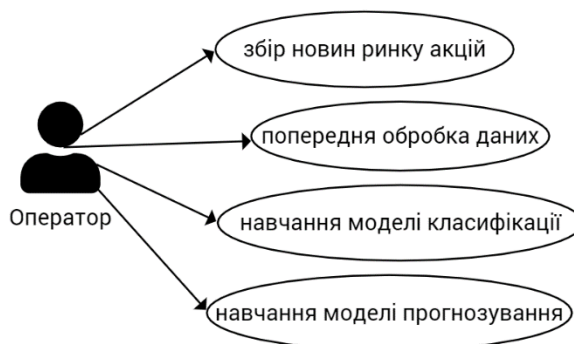


Рисунок 3.11 – Діаграма прецедентів оператора

3.10 Висновки з розділу 3

У третьому розділі було розглянуто, які мови програмування, бібліотеки, фреймворки та бази даних можна застосувати при розробці системи для проведення аналізу тональності заголовків новин ринку акцій для обґрунтування інвестиційних рішень.

Проаналізувавши розглянуті варіанти, остаточно було вирішено обрати для розробки мову програмування Python, бібліотеки для роботи з даними та їх візуалізації Pandas та Plotly, бібліотеки для проведення аналізу тональності текстових повідомлень та прогнозування тенденцій зміни цін акцій FastText та Keras відповідно, вебфреймворк Flask, нереляційну базу даних MongoDB та драйвер PyMongo для взаємодії з нею.

Також було описано структуру програми та бази даних, побудовано схему бази даних та схему роботи вебзастосунку у вигляді діаграми компонентів, побудовано діаграми прецедентів, які показують варіанти використання системи користувачем та оператором.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Алгоритм функціонування програмного забезпечення

Вебзастосунок надає користувачеві три режими:

- перегляд графіків ціни акцій та оцінок тональності новин однієї компанії;
- перегляд загальної гістограми оцінок тональності новин декількох компаній;
- перегляд таблиці розподілу акцій за тенденціями зміни їх цін.

Також надає вибір компаній, відомості яких продивляться. Тож як саме вебзастосунок виконується залежить від дій користувача. Таким чином, було визначено і побудовано схему, яка описує алгоритм функціонування системи в залежності від дій користувача (рис. 4.1).

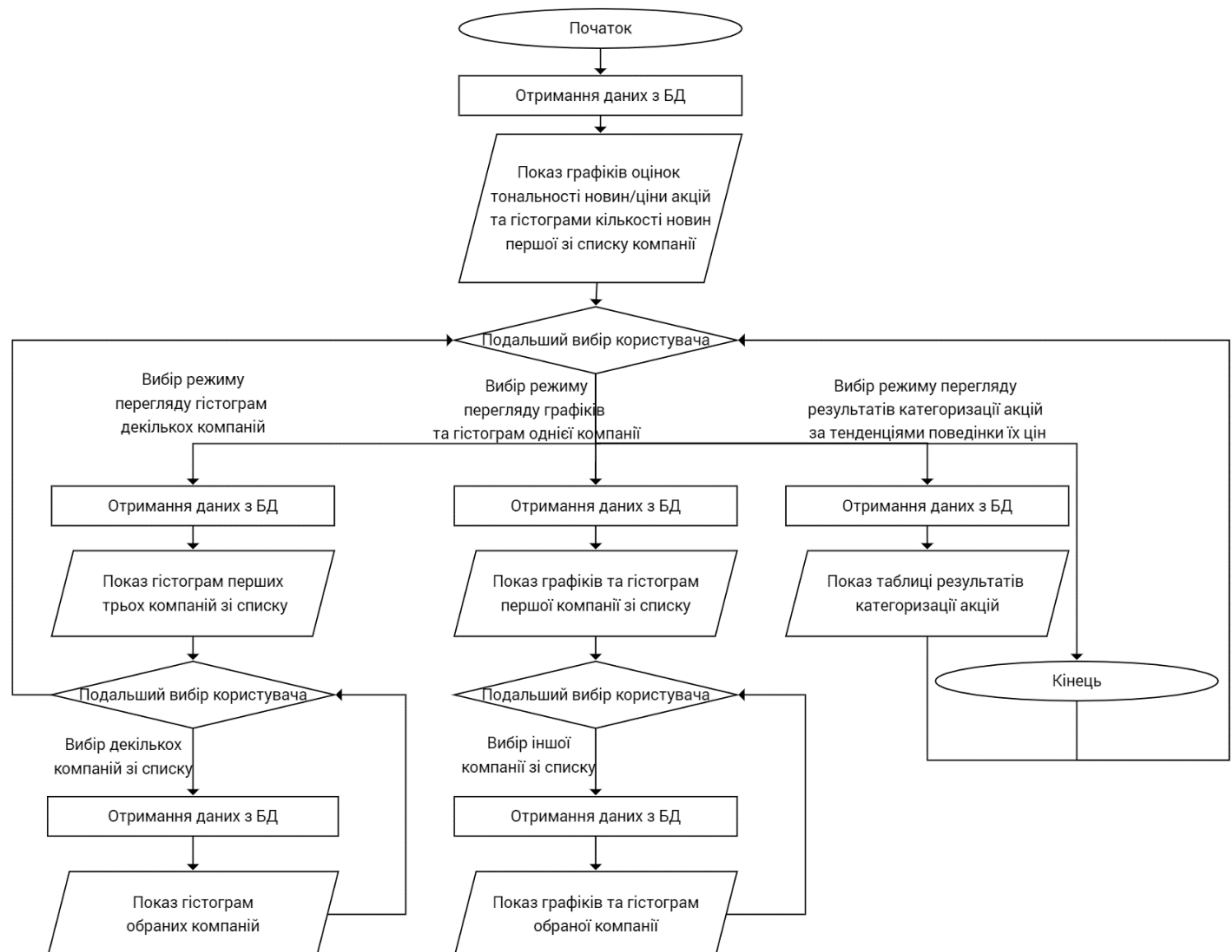


Рисунок 4.1 – Схема функціонування програми

4.2 Опис модулів реалізованої програми

4.2.1 Опис модулів пакету db

В таблиці 4.1 наведено опис функцій модулю data_collectors.

Вхідні дані: назва компанії, поточна дата, кількість днів.

Вихідні дані: зібрані заголовки новин, отримані ціни акцій.

Таблиця 4.1 – Опис функцій модулю data_collectors

№	Найменування функції	Опис функції
1	<code>collect_news(stock)</code>	Функція збирає заголовки новин вказаної компанії, викликаючи наступні функції.
2	<code>get_news_from_polygon(stock, date)</code>	Функція збирає заголовки новин за останні 30 днів від поточної дати за допомогою Polygon.io API.
3	<code>get_news_from_newsapi(stock, date)</code>	Функція збирає заголовки новин за останні 30 днів від поточної дати за допомогою News API.
4	<code>get_news_from_finnhub(stock, date)</code>	Функція збирає заголовки новин за останні 30 днів від поточної дати за допомогою Finnhub API.
5	<code>get_stock_prices_from_yfinance(stock, days_quan)</code>	Функція отримує ціни акцій за кількість днів, переданих у параметр days_quan.

В таблиці 4.2 наведено опис функцій модулю db.

Вхідні дані: нові дані, назва колекції, назва документу.

Вихідні дані: оновлені в БД дані.

Таблиця 4.2 – Опис функцій модулю db

№	Найменування функції	Опис функції
1	<code>update_db(new_data, collection_name, document_name)</code>	Функція оновлює документи колекцій в MongoDB.

В таблиці 4.3 наведено опис функцій модулю `updater`

Вхідні дані: заголовки новин, оцінки тональностей.

Вихідні дані: підготовлені для зберігання в БД дані.

Таблиця 4.3 – Опис функцій модулю `updater`

№	Найменування функції	Опис функції
1	<code>updater()</code>	Функція завантажує зібрані з <code>data_collectors</code> заголовки новин, запитує для них аналіз тональності з <code>fasttext_analysis</code> та передає сформовані дані в <code>db</code> для збереження.

4.2.2 Опис модулів пакету `fasttext_analysis`

В таблиці 4.4 наведено опис функцій модулю `preprocessing`.

Вхідні дані: навчальний набір даних або заголовки новин.

Вихідні дані: підготовлені для аналізу тональності дані.

Таблиця 4.4 – Опис функцій модулю `preprocessing`

№	Найменування функції	Опис функції
1	<code>preprocess_data(data)</code>	Функція здійснює попередню обробку даних, які передаються в параметр <code>data</code> .

В таблиці 4.5 наведено опис функцій модулю `fasttext_training`.

Вхідні дані: розмічений для навчання набір даних.

Вихідні дані: навчена модель аналізу тональності.

Таблиця 4.5 – Опис функцій модулю `fasttext_training`

№	Найменування функції	Опис функції
1	<code>fasttext_training(data)</code>	Функція здійснює навчання моделі аналізу тональності, використовуючи розмічений набір даних, який передається в параметр <code>data</code> .

Код, яким відбувається навчання моделі FastText класифікації та її зберігання у файл:

```
model = fasttext.train_supervised(
    input="data_train.txt",
    wordNgrams = 3,
    epoch=50,
    loss="softmax",
    lr=1.0
)
model.save_model("model.bin")
```

В таблиці 4.6 наведено опис функцій модулю `fasttext_analysis`.

Вхідні дані: заголовки новин.

Вихідні дані: оцінки тональностей.

Таблиця 4.6 – Опис функцій модулю `fasttext_analysis`

№	Найменування функції	Опис функції
1	<code>fasttext_analysis(data)</code>	Функція проводить аналіз тональності заголовків новин, які передаються в параметр <code>data</code> .

Код, яким відбувається класифікація заголовків новин за їх оцінками тональності:

```
model_prediction = model.predict(headline)
sentiment_class = model_prediction[0][0]
sentiment_score = model_prediction[1][0]
```

4.2.3 Опис модулю `stock_categorization`

В таблиці 4.7 наведено опис функцій модулю `stock_categorization`.

Вхідні дані: оцінки тональностей.

Вихідні дані: результати категоризації акцій за тенденціями зміни їх цін.

Таблиця 4.7 – Опис функцій модулю stock_categorization

№	Найменування функції	Опис функції
1	<code>predict_stock_attitude(data)</code>	Функція здійснює LSTM прогнозування тенденції зміни ціни акції на основі переданих в параметр <code>data</code> результатів аналізу тональності.
2	<code>categorize_stocks()</code>	Функція здійснює категоризацію акцій за тенденціями зміни їх цін на основі отриманих результатів LSTM прогнозування.

Код, яким відбувається створення та навчання моделі LSTM прогнозування:

```
model = Sequential()
model.add(
    LSTM(
        units=100,
        activation="tanh",
        input_shape=(7, 1),
        return_sequences=True
    )
)
model.add(LSTM(units=100, activation="tanh"))
model.add(Dense(units=1))
model.compile(loss="mse", metrics=["mean_squared_error"])
model.fit(X, y, epochs=200)
```

Код, яким модель застосовується для визначення прогнозу:

```
stock_prediction = model.predict(previous_data)
```

4.2.4 Опис модулю plots

В таблиці 4.8 наведено опис функцій модулю plots.

Вхідні дані: назви компаній.

Вихідні дані: побудовані графіки та гістограми.

Таблиця 4.8 – Опис функцій модулю plots

№	Найменування функції	Опис функції
1	<code>plot_company_sentiment(company)</code>	Функція будує графік оцінок аналізу тональності/ціни акцій та гістограму кількості проаналізованих заголовків новин, переданої в параметр <code>company</code> компанії.
2	<code>plot_companies_sentiment(companies)</code>	Функція будує гістограму оцінок аналізу тональності та кількості проаналізованих заголовків новин, переданих в параметр <code>companies</code> компаній.

4.2.5 Опис модулю app

В таблиці 4.9 наведено опис функцій модулю app.

Вхідні дані: назви компаній.

Вихідні дані: згенеровані шаблони з графіками та гістограмами.

Таблиця 4.9 – Опис функцій модулю app

№	Найменування функції	Опис функції
1	<code>@app.route('/') @app.route('/single_company_analysis') show_single_company_analysis()</code>	Функція відповідає на запит URL вказаних в декораторах <code>route</code> та генерує шаблон з графіком та гістограмою компанії, переданої в параметр URL.
2	<code>@app.route('/several_companies_analysis', methods=['GET', 'POST']) show_several_companies_analysis()</code>	Функція відповідає на запит URL вказаного в декораторі <code>route</code> та генерує шаблон з гістограмами трьох перших зі списку компаній при GET запиті та гістограми компаній, обраних зі списку в шаблоні при POST запиті.

Продовження таблиці 4.9.

№	Найменування функції	Опис функції
3	<code>@app.route('/stock_categorization_results')</code> <code>show_stock_categorization_results()</code>	Функція відповідає на запит URL вказаного в декораторі route та генерує шаблон з таблицею результатів категоризації всіх акцій.

4.3 Висновки з розділу 4

У четвертому розділі було визначено і побудовано схему, яка описує алгоритм функціонування системи в залежності від дій користувача, описано модулі програми з наведенням розроблених функцій та їх вхідних та вихідних даних.

5 ЕКСПЛУАТАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Призначення програми

Розроблений вебзастосунок призначений для проведення аналізу тональності текстових повідомлень (заголовків новин) про ринок акцій для дослідження ситуації на ринку і обґрунтування інвестиційних рішень.

Областю застосування вебзастосунку є аналіз ситуації на ринку акцій, тож вебзастосунок надає користувачеві (інвестору або трейдеру) можливість переглядати графіки цін акцій та оцінок тональності новин, гістограми кількості проаналізованих новин обраної однієї компанії або декількох, таблицю результатів категоризації акцій за прогнозами тенденцій зміни їх цін.

5.2 Умови виконання програми

Умовами виконання програми є встановлений Python інтерпретатор версії 3.8 та потрібні для роботи програми бібліотеки (Flask, PyMongo, Pandas, Plotly, FastText, Keras та інші вказані у файлі requirements.txt), 512 мегабайтів дискового простору задля безпроблемного розміщення потрібних засобів та тимчасових файлів. Вебзастосунок призначений для роботи тільки в браузерях на персональних настільних комп'ютерах або ноутбуках.

5.3 Виконання програми

Вебзастосунок надає користувачеві три режими:

- перегляд графіків ціни акцій та оцінок тональності новин однієї обраної компанії;
- перегляд загальної гістограми оцінок тональності новин декількох обраних компаній;
- перегляд таблиці розподілу акцій за тенденціями зміни їх цін.

Користувач може обрати один із режимів, використовуючи кнопки вибору режимів (рис. 5.1).



Рисунок 5.1 – Режими перегляду

5.3.1 Виконання програми в режимі перегляду графіків ціни акцій та оцінок тональності новин однієї компанії

В даному режимі користувач може обрати компанію з випадającego списку (рис. 5.2).

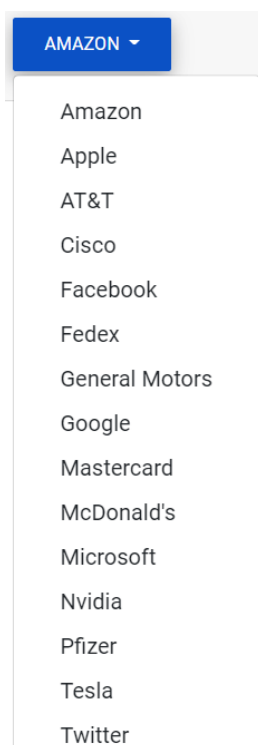


Рисунок 5.2 – Список компаній для вибору однієї

Після вибору компанії завантажуються відповідні графіки та гістограми результатів аналізу тональності. Наприклад, графіки та гістограми результатів аналізу тональності новин обраної зі списку компанії Amazon представлені на

рисунку 5.3, де синій графік – це графік цін акцій за період від 15.07.2021 до 10.11.2021, а зелений – графік результатів аналізу тональності новин. Зелені та червоні кола на графіку результатів аналізу тональності показують, який загальний настрій новин домінує, позитивний чи негативний відповідно.

Нижче графіків розміщені гістограми, які показують кількості проаналізованих за день новин та виділяють аналогічно зеленим та червоним кольором, скільки було позитивних новин, а скільки негативних.

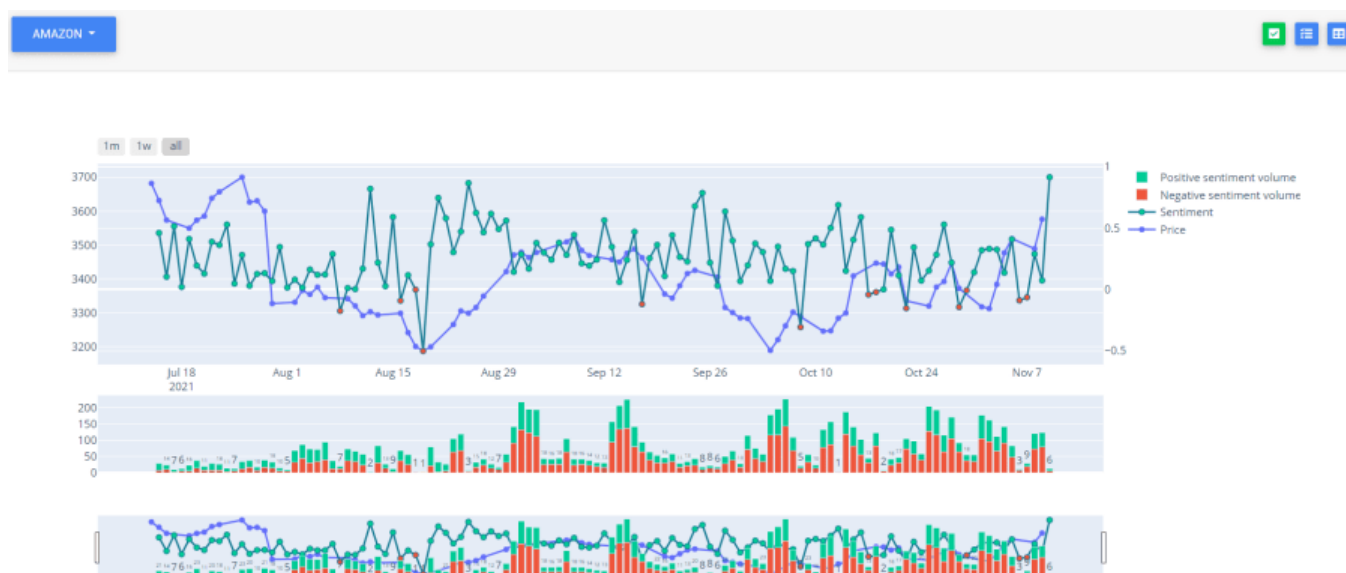


Рисунок 5.3 – Графіки та гістограми результатів аналізу тональності новин компанії Amazon

Наводячи курсором на графіки, користувач може побачити спливаючі повідомлення з інформацією. Наприклад, якщо навести на одну з точок графіку результатів аналізу тональності, то з'явиться повідомлення, яке показує дату та загальну оцінку тональності новин (рис. 5.4).



Рисунок 5.4 – Спливаюче повідомлення про оцінку тональності новин

А якщо навести на одну з колонок гістограми, то з'явиться повідомлення, яке показує дату та кількість проаналізованих новин (рис. 5.5).

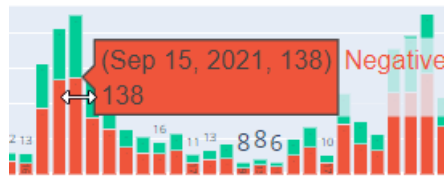


Рисунок 5.5 – Спливаюче повідомлення про кількість проаналізованих новин

Користувач може обирати за який період переглядати графіки та гістограми. Обравши відповідну опцію, можна продивлятися дані за останній місяць (1m) або за останній тиждень (1w), або за весь доступний період (all), що представлено на рисунку 5.6.



Рисунок 5.6 – Опції вибору за який період продивляться дані

Крім цього, період для перегляду можна налаштовувати більш гнучко, двигави повзунок, який знаходиться під графіками та гістограмами (рис. 5.7).

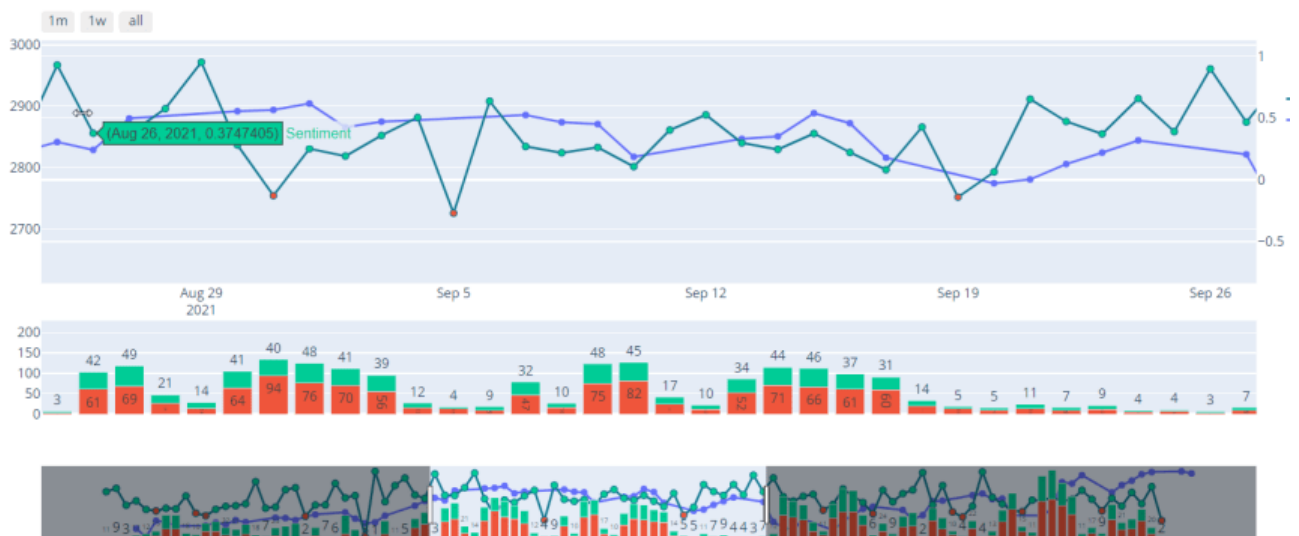


Рисунок 5.7 – Налаштування перегляду графіків повзунком

5.3.2 Виконання програми в режимі перегляду загальної гістограми оцінок тональності новин декількох компаній

В даному режимі користувач може обрати декілька компаній з випадającego списку та натиснути «Show» (рис. 5.8) для перегляду гістограм результатів аналізу тональності (рис. 5.9).

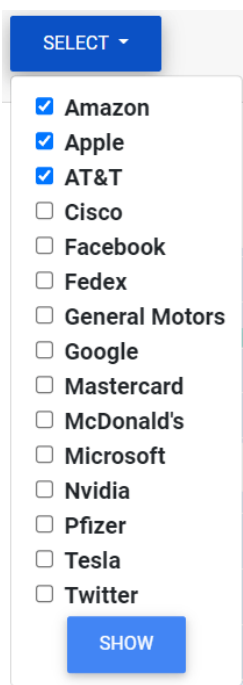


Рисунок 5.8 – Список компаній для вибору декількох



Рисунок 5.9 – Гістограми результатів аналізу тональності новин Facebook, Google та Microsoft

Так само як і в попередньому режимі, користувач може наводити на колонки курсором та отримувати спливаючі повідомлення.

5.3.3 Виконання програми в режимі перегляду таблиці розподілу акцій за тенденціями зміни їх цін

В даному режимі користувач може продивитись таблицю розподілу акцій за тенденціями зміни їх цін за прогнозом на наступні 7 днів від визначеної дати.

Наприклад, на рисунку 5.10 представлено таблицю розподілу акцій за тенденціями зміни їх цін за прогнозом від 10.11.2021.

Forecast from 10.11.2021

significant drop	slight drop	almost intact	slight rise	significant rise
AT&T	Mastercard	General Motors	Google	Tesla
Cisco	Facebook	McDonald's	Nvidia	Microsoft
	Fedex		Pfizer	Twitter
			Amazon	Apple

Рисунок 5.10 – Таблиця розподілу акцій за тенденціями зміни їх цін

5.3.4 Повідомлення про помилки

Вебзастосунок оповіщає користувача, коли в режимі перегляду результатів аналізу тональності декількох компаній не було обрано жодної і що потрібно обрати хоча б одну (рис. 5.11).

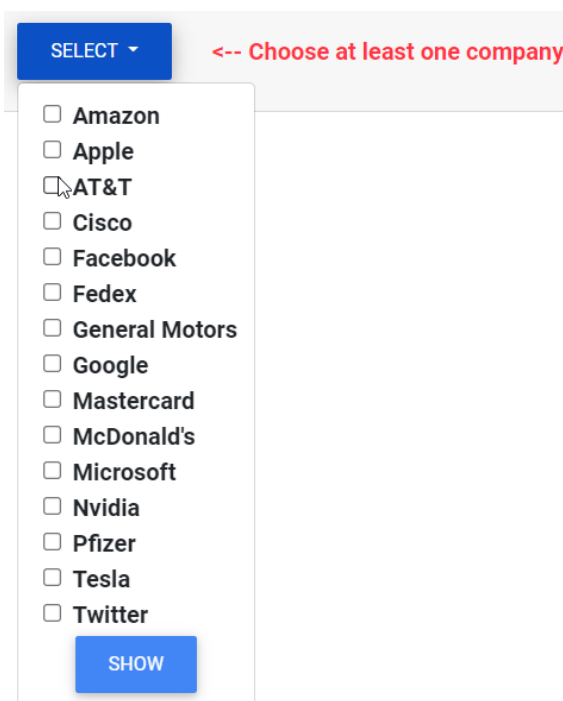


Рисунок 5.11 – Оповіщення про не обрані зі списку компанії

А якщо виникає якась помилка при завантаженні даних і через це графіки та гістограми не можуть бути відображені, то з'являється відповідне про це повідомлення (рис. 5.12).

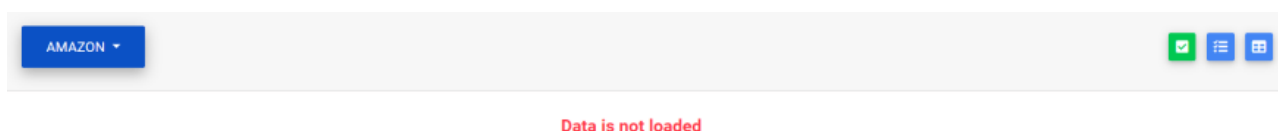


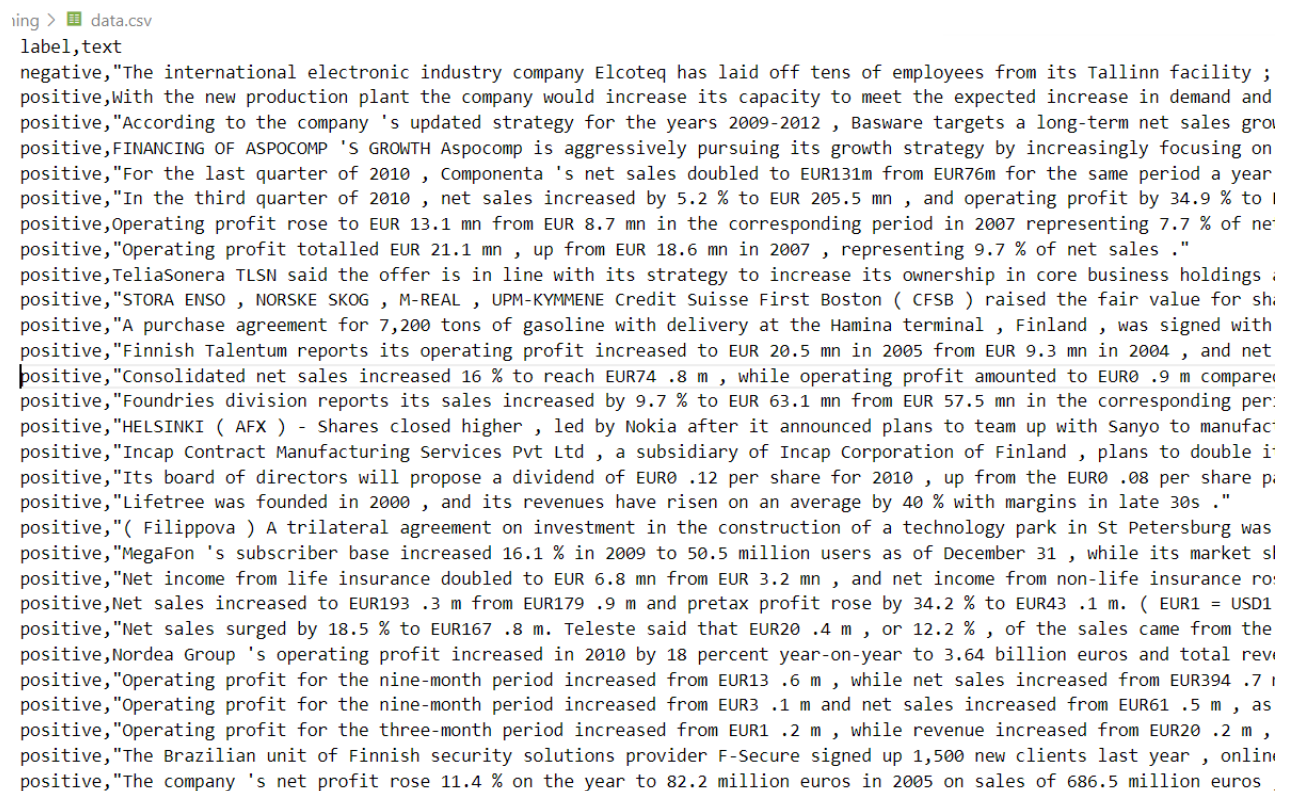
Рисунок 5.12 – Оповіщення про не завантажені дані

5.4 Експериментальне дослідження запропонованого методу

Для проведення навчання моделі FastText класифікації текстових повідомлень за їх тональностями було знайдено три набори навчальних даних, які підходять для поставленої задачі, тобто мають фінансові новини з відповідними мітками їх тональності [60].

З цих трьох наборів було сформовано один набір і було визначено кількості новин за їх тональностями: позитивних – 6052, негативних – 3710, нейтральних – 2879.

Так як поставлена задача бінарної класифікації, нейтральні новини було видалено. А щоб при навчанні моделі не було переваги позитивних новин над негативними, їх кількість було зменшено до 3710 (рис. 5.13).



```

ing > data.csv
label,text
negative,"The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility ;
positive,With the new production plant the company would increase its capacity to meet the expected increase in demand and
positive,"According to the company 's updated strategy for the years 2009-2012 , Basware targets a long-term net sales gro
positive,FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is aggressively pursuing its growth strategy by increasingly focusing on
positive,"For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year
positive,"In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5 mn , and operating profit by 34.9 % to 1
positive,Operating profit rose to EUR 13.1 mn from EUR 8.7 mn in the corresponding period in 2007 representing 7.7 % of ne
positive,"Operating profit totalled EUR 21.1 mn , up from EUR 18.6 mn in 2007 , representing 9.7 % of net sales ."
positive,TeliaSonera TLSN said the offer is in line with its strategy to increase its ownership in core business holdings i
positive,"STORA ENSO , NORSKE SKOG , M-REAL , UPM-KYMMENE Credit Suisse First Boston ( CFSB ) raised the fair value for sh
positive,"A purchase agreement for 7,200 tons of gasoline with delivery at the Hamina terminal , Finland , was signed with
positive,"Finnish Talentum reports its operating profit increased to EUR 20.5 mn in 2005 from EUR 9.3 mn in 2004 , and net
positive,"Consolidated net sales increased 16 % to reach EUR74 .8 m , while operating profit amounted to EUR0 .9 m compare
positive,"Foundries division reports its sales increased by 9.7 % to EUR 63.1 mn from EUR 57.5 mn in the corresponding per
positive,"HELSINKI ( AFX ) - Shares closed higher , led by Nokia after it announced plans to team up with Sanyo to manufac
positive,"Incap Contract Manufacturing Services Pvt Ltd , a subsidiary of Incap Corporation of Finland , plans to double i
positive,"Its board of directors will propose a dividend of EUR0 .12 per share for 2010 , up from the EUR0 .08 per share p
positive,"Lifetree was founded in 2000 , and its revenues have risen on an average by 40 % with margins in late 30s ."
positive,"( Filippova ) A trilateral agreement on investment in the construction of a technology park in St Petersburg was
positive,"MegaFon 's subscriber base increased 16.1 % in 2009 to 50.5 million users as of December 31 , while its market sl
positive,"Net income from life insurance doubled to EUR 6.8 mn from EUR 3.2 mn , and net income from non-life insurance ro
positive,Net sales increased to EUR193 .3 m from EUR179 .9 m and pretax profit rose by 34.2 % to EUR43 .1 m. ( EUR1 = USD1
positive,"Net sales surged by 18.5 % to EUR167 .8 m. Teleste said that EUR20 .4 m , or 12.2 % , of the sales came from the
positive,Nordea Group 's operating profit increased in 2010 by 18 percent year-on-year to 3.64 billion euros and total rev
positive,"Operating profit for the nine-month period increased from EUR13 .6 m , while net sales increased from EUR394 .7
positive,"Operating profit for the nine-month period increased from EUR3 .1 m and net sales increased from EUR61 .5 m , as
positive,"Operating profit for the three-month period increased from EUR1 .2 m , while revenue increased from EUR20 .2 m ,
positive,"The Brazilian unit of Finnish security solutions provider F-Secure signed up 1,500 new clients last year , onlin
positive,"The company 's net profit rose 11.4 % on the year to 82.2 million euros in 2005 on sales of 686.5 million euros

```

Рисунок 5.13 – Набір даних для навчання моделі класифікації

Сформований набір було розділено на тренувальний (3210 позитивних та 3210 негативних новин) та тестовий (500 позитивних та 500 негативних новин).

Модель FastText класифікації було навчено на тренувальному наборі даних та протестовано на тестовому наборі.

На основі результатів тестування моделі класифікації було побудовано матрицю невідповідностей (табл. 5.1) та обчислено показники якості роботи моделі: влучність (precision), повнота (recall) та f1-міра (f1-score) (табл. 5.2).

Таблиця 5.1 – Матриця невідповідностей

		Фактично	
		Клас «позитивний»	Клас «негативний»
Класифіковано	Клас «позитивний»	414	87
	Клас «негативний»	86	413

Таблиця 5.2 – Показники якості роботи моделі

Клас	Влучність	Повнота	F1-міра
Клас «позитивний»	0.827	0.826	0.826
Клас «негативний»	0.826	0.827	0.827

Як видно, модель класифікації майже однаково розподілила новини на позитивні та негативні, тобто не має переваги в одну чи іншу сторону при визначенні настроїв новин. Загальна точність (ассигасу) роботи моделі – 0.83.

В таблиці 5.3 наведено порівняння точності роботи використовуваної моделі FastText класифікації з іншими моделями машинного навчання, які розглядалися в пункті 2.4.

Таблиця 5.3 – Порівняння моделей

Модель	Точність
Модель наївного баєса	0.7
Модель опорних векторів	0.8
Модель випадкового лісу	0.79
Використовувана FastText модель	0.83

Отже, використовувана FastText модель забезпечила точність результатів більшу на 18.57%, ніж модель наївного баєса, більшу на 3.75%, ніж модель опорних векторів, та більшу на 5.06%, ніж модель випадкового лісу.

Далі навчена модель застосовувалась для аналізу тональності заголовків новин, зібраних про кожну компанію зі списку. Результати аналізу використовувались для здійснення LSTM прогнозування тенденцій зміни цін акцій кожної компанії зі списку.

Наприклад, за весь доступний період від 20.07.2021 до 10.11.2021 було проведено аналіз тональності новин компанії Fedex і ці дані були подані до LSTM моделі для навчання. На рисунку 5.14 зображено графік середньоквадратичної похибки під час навчання моделі.

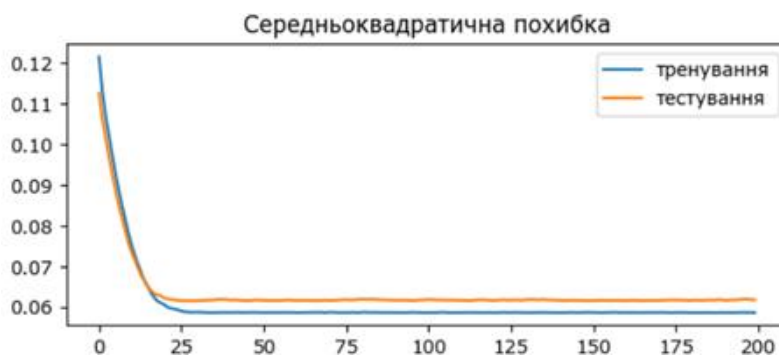


Рисунок 5.14 – Середньоквадратична похибка LSTM моделі

LSTM моделі будуються окремо для кожної акції, відповідно і результати навчання виявляються різні. Але загалом для більш якісного навчання моделі потребують даних за довший період.

Навчена модель на основі оцінок тональності останніх 7 днів видало число прогнозу 0.18 і далі методом категоризації, враховуючи отриманий результат, для акцій компанії Fedex було визначено категорію «slight drop», що означає, що в наступні 7 днів ціна цих акцій може незначно впасти.

Так само відбувається з акціями всіх інших компаній зі списку і згодом формується таблиця результатів категоризації акцій (п. 5.3.3), яка може бути основою для прийняття рішень щодо купівлі або продажу акції.

Серед розглянутих в пункті 1.2 вебзастосунків зі схожим напрямом та дослідницьких статей подібної тематики, на які є посилання в пунктах 2.1-2.6, не було виявлено аналогів, які б застосовували або розглядали запропонований метод визначення тенденцій зміни цін акцій.

Загалом способи вирішення задачі аналізу тональності текстових повідомлень про ринок акцій для прогнозування цін акцій вимагають більше досліджень для застосування цих напрацювань у комерційному програмному забезпеченні.

Результати роботи методу можуть використовуватися для подальших досліджень, але як інвестиційна рекомендація розцінюватися не можуть.

Покращити результати роботи запропонованого методу можна шляхом:

- забезпечення навчання моделі класифікації якіснішим та більшим набором даних, щоб модель могла краще розуміти контекст текстових повідомлень та розпізнавати їх настрої з меншою кількістю помилок;

- забезпечення навчання моделі прогнозування набором оцінок тональності новин за набагато довший період часу, щоб модель могла краще зчитувати закономірності у послідовностях значень;

- додаткового використання відомостей про кількості новин у моделі прогнозування, щоб модель розпізнавала періоди збільшення або зменшення зацікавленості до певної акції і відповідно формувала прогноз, спираючись і на цю інформацію.

5.5 Висновки з розділу 5

У п'ятому розділі було визначено призначення програми та умови її виконання, наведено екранні копії виконання програми та повідомлення про помилки, які можуть з'являтися під час користування, наведено експериментальне дослідження запропонованого методу визначення тенденцій зміни цін акцій.

ВИСНОВКИ

В даній дипломній кваліфікаційній роботі було вирішено актуальне завдання розробки програмного засобу, а саме вебзастосунку, для проведення аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

В процесі виконання поставленого завдання було проаналізовано предметну область – ринок акцій, розглянуто програмні засоби, що вирішують аналогічні завдання, описано процес попередньої обробки даних та популярні алгоритми вкладень слів Word2Vec, GloVe та FastText, проаналізовано методи машинного та глибинного навчання, які можуть застосовуватися для вирішення поставленої задачі, обрано метод FastText для проведення аналізу тональності текстових повідомлень та метод LSTM для прогнозування тенденцій зміни цін акцій, запропоновано власний метод визначення тенденцій зміни цін акцій та описано його алгоритм.

Було проведено аналіз інструментів для розробки вебзастосунку та обрано мову програмування Python, бібліотеки для роботи з даними та їх візуалізації Pandas та Plotly, бібліотеки для проведення аналізу тональності текстових повідомлень та прогнозування тенденцій зміни цін акцій FastText і Keras відповідно, вебфреймворк Flask, нереляційну базу даних MongoDB та драйвер PyMongo для взаємодії з нею.

Було описано структуру програми, побудовано схему бази даних, діаграму компонентів системи та діаграми прецедентів, що показують варіанти використання застосунку користувачем та оператором.

Було побудовано схему, яка описує алгоритм функціонування програми в залежності від дій користувача, та описано модулі програми з розробленими функціями.

В результаті проведеної роботи було розроблено вебзастосунок для проведення аналізу тональності новин ринку акцій для обґрунтування інвестиційних рішень.

Вебзастосунок надає користувачеві три режими: перегляд графіків ціни акцій та оцінок тональності новин однієї обраної компанії, перегляд загальної гістограми оцінок тональності новин декількох обраних компаній, перегляд таблиці розподілу акцій за тенденціями зміни їх цін.

Інтерфейс застосунку є простим і зрозумілим, що дозволить без особливих складностей працювати з ним користувачам.

Наукова новизна дипломної роботи полягає у розробленому методі визначення тенденцій зміни цін акцій, який характеризується використанням моделей FastText та LSTM і дозволяє розв'язати задачу категоризації акцій і виконати аналіз тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень. Етапами методу є попередня обробка даних, створення вкладень слів, навчання моделі класифікації та подальша класифікація новин, навчання моделі прогнозування та подальше прогнозування тенденцій зміни цін акцій, категоризація акцій за тенденціями зміни їх цін.

Розроблений метод дозволяє розв'язати задачу аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Pang, B. Opinion Mining and Sentiment Analysis [Text] / B. Pang, L. Lillian // Foundations and Trends in Information Retrieval. – 2008. – Vol. 2, Issue 2. – P. 1–135.
2. Анализ тональности текста: концепция, методы, области применения [Электрон. ресурс]. – Режим доступа <http://datareview.info/article/analiz-tonalnosti-teksta-kontseptsiya-metodyi-oblasti-primeneniya/>
3. О сентимент-анализе и перспективах его применения [Электрон. ресурс]. – Режим доступа : <https://cyberleninka.ru/article/n/o-sentiment-analize-i-perspektivah-ego-primeneniya>
4. Trading the news: Use machine-readable data to find alpha [Electronic resource]. – Access mode : <https://www.bloomberg.com/professional/blog/trading-news-use-machine-readable-data-find-alpha/>
5. Ринок акцій [Електрон. ресурс]. – Режим доступу : <https://uk.templeprotestant.org/equitymarket-6482>
6. Біржа фондова [Електрон. ресурс]. – Режим доступу : https://leksika.com.ua/10050728/legal/birzha_fondova
7. Компанія [Електрон. ресурс]. – Режим доступу : <http://slovopedia.org.ua/38/53402/380686.html>
8. Ринок акцій та перспективи розвитку [Електрон. ресурс]. – Режим доступу : <https://buklib.net/books/25540/>
9. Трейдер [Електрон. ресурс]. – Режим доступу : <https://alpari.com/ua/beginner/glossary/trader/>
10. Інвестор [Електрон. ресурс]. – Режим доступу : <https://www.jnsm.com.ua/cgi-bin/u/book/sis.pl?Qry=Інвестор>
11. Introduction to Stock Market Analysis [Electronic resource]. – Access mode : <https://www.analyticssteps.com/blogs/introduction-stock-market-analysis>
12. How Sentdex and Sentiment Analysis works [Electronic resource]. – Access mode : <http://sentdex.com/how-sentdex-works/>

13. Stocksnips [Electronic resource]. – Access mode : <https://stocksnips.net/financial-sentiment/analytics-platform/>
14. Cryptomood [Electronic resource]. – Access mode : <https://cryptomood.com/personal/products/sentiment-analysis-app/>
15. Market Prices Forecast [Electronic resource]. – Access mode : https://skipper.shinyapps.io/shiny_market_prices_forecast/#section-forecast
16. Getting started with Text Preprocessing [Electronic resource]. – Access mode : <https://www.kaggle.com/sudalairajkumar/getting-started-with-text-preprocessing>
17. A Guide to Word Embedding [Electronic resource]. – Access mode : <https://towardsdatascience.com/a-guide-to-word-embeddings-8a23817ab60f>
18. Short technical information about Word2Vec, GloVe and Fasttext [Electronic resource]. – Access mode : <https://towardsdatascience.com/short-technical-information-about-word2vec-glove-and-fasttext-d38e4f529ca8>
19. Negative Sampling and GloVe [Electronic resource]. – Access mode : <https://towardsdatascience.com/nlp-101-negative-sampling-and-glove-936c88f3bc68>
20. Как работает FastText и где ее применять [Электрон. ресурс]. – Режим доступа : <https://sysblok.ru/nlp/kak-rabotaet-fasttext-i-gde-ee-primenjat/>
21. Types of Sentiment Analysis and Its Uses [Electronic resource]. – Access mode : <https://medium.com/swlh/types-of-sentiment-analysis-and-its-uses-ad733535c895>
22. Завдання класифікації [Електрон. ресурс]. – Режим доступу : https://znaimo.com.ua/Завдання_класифікації
23. Shi, F. Learn About Dictionary-Based Sentiment Analysis in Python With Data From the Economic News Article Tone Dataset [Text] / F. Shi // SAGE Research Methods Datasets. – 2019. – Vol 2. – P. 1–9.
24. Shah, D. Predicting the Effects of News Sentiments on the Stock Market [Text] / D. Shah, H. Isah, F. Zulkernine // IEEE International Conference on Big Data : Seattle, 10–13 December 2018 : proceedings. – New York : IEEE, 2018. – P. 4705–4708.

25. Turner, Z. Lexicon-based sentiment analysis for stock movement prediction [Text] / Z. Turner, K. Labille, S. Gauch // Journal of Construction Materials. – 2021. – Vol. 14, Issue 5. – P. 185–191.

26. Romanyshyn, M. Rule-Based Sentiment Analysis of Ukrainian Reviews [Text] / M. Romanyshyn // International Journal of Artificial Intelligence & Applications. – 2013. – Vol. 4, Issue 2. – P. 103–111.

27. Selimi, M. A Proposed Model for Stock Price Prediction Based on Financial News [Text] / M. Selimi, A. Besimi // Enterprise Research Inovation Conference : Rovinj, 12–14 September 2019 : proceedings. – Zagreb: IRENET, 2019. – Vol. 5. – P. 100–107.

28. Stock Prediction using Sentiment analysis and Long Short Term Memory [Text] / [H. Panday, V. Vijayarajan, A. Mahendran et al.] // European Journal of Molecular & Clinical Medicine. – 2020. – Vol. 7, Issue 2. – P. 5060–5069.

29. Обучаем компьютер чувствам [Электрон. ресурс]. – Режим доступа : <https://habr.com/ru/post/149605/>

30. Naive Bayes and Sentiment Classification [Electronic resource]. – Access mode : <https://web.stanford.edu/~jurafsky/slp3/4>

31. Financial Context News Sentiment Analysis for the Lithuanian Language [Text] / [R. Itrimaitis, P. Stefanovic, S. Ramanauskaite et al.] // Applied Sciences. – 2021. – Vol. 11, Issue 10. – P. 1–13.

32. Joshi, K. Stock trend prediction using news sentiment analysis [Text] / K. Joshi, H. N. Bharathi, R. Jyothi // International Journal of Computer Science & Information Technology. – 2016. – Vol. 8, Issue 3. – P. 67–76.

33. Метод опорных векторов [Электрон. ресурс]. – Режим доступа : <http://statistica.ru/local-portals/data-mining/metod-opornykh-vektorov/>

34. Sridhar, S. Analysis of the effect of news sentiment on stock market prices through event embedding [Text] / S. Sridhar, S. Sanagavarapu // Federated Conference on Computer Science and Information Systems : 16th Conference on Computer Science and Intelligence Systems, Sofia, 2–5 September 2021 : proceedings. – New York : IEEE, 2021. – P. 147–150.

35. Вступ до машинного навчання [Електрон. ресурс]. – Режим доступу : <http://specials.kunsht.com.ua/machinelearning3>

36. FastText [Electronic resource] : Under the Hood. – Access mode : <https://towardsdatascience.com/fasttext-under-the-hood-11efc57b2b3>

37. Gumus, A. Stock market prediction in Istanbul stock exchange by combining stock price information and sentiment analysis [Text] / A. Gumus, O. Sakar // International journal of advances in engineering and pure sciences. – 2021. – Vol. 33, Issue 1. – P. 18–27.

38. Unsupervised self-training for sentiment analysis of code-switched data [Text] / [A. Gupta, S. Menghani, S. Rallabandi et al.] // Workshop on Computational Approaches to Linguistic Code-Switching : Stroudsburg, June 2021 : proceedings. – Stroudsburg: Association for Computational Linguistics, 2021. – Vol. 5. – P. 103–112.

39. Stock market trading based on market sentiments and reinforcement learning [Text] / [K. Suhail, S. Sankar, A. Kumar et al.] // Computers, Materials & Continua. – 2021. – Vol. 70, Issue 1. – P. 935–950.

40. Введення у глибинне навчання [Електрон. ресурс]. – Режим доступу : <https://sites.google.com/a/kubg.edu.ua/integrovanii-sistemi/masinne-navcanna-machine-learning-glibinne-navcanna-deep-learning/vvedenna-u-glibinne-navcanna>

41. Анализ тональности текстов с помощью сверточных нейронных сетей [Электрон. ресурс]. – Режим доступа : <https://habr.com/ru/company/mailru/blog/417767/>

42. Zeynep, H. Sentiment Analysis Based Direction Prediction in Bitcoin using Deep Learning Algorithms and Word Embedding Models [Text] / H. Zeynep // International Journal of Intelligent Systems and Applications. – 2020. – Vol. 8, Issue 2. – P. 60–65.

43. LSTM – нейронная сеть с долгой краткосрочной памятью [Электрон. ресурс]. – Режим доступа : <https://neurohive.io/ru/osnovy-data-science/lstm-nejronnaja-set/>

44. Jiawei, X. Stock market trend prediction with sentiment analysis based on LSTM neural network [Text] / X. Jiawei, T. Murata // International MultiConference

of Engineers and Computer Scientists : Hong Kong, 13–15 March 2019 : proceedings.
– Hong Kong: International Association of Engineers, 2019. – P. 475–479.

45. Как устроена нейросеть BERT от Google [Электрон. ресурс]. – Режим доступа : <https://sysblok.ru/knowhow/kak-ustroena-nejroset-bert-ot-google/>

46. Ko, C. LSTM-based sentiment analysis for stock price forecast [Text] / C. Ko, H. Chang // PeerJ Computer Science. – 2021. – Vol. 7, Issue 408. – P. 1–23.

47. FinBERT [Text] : a pre-trained financial language representation model for financial text mining / [D. Huang, K. Huang, Z. Liu et al.] // International Joint Conference on Artificial Intelligence : Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, 11–17 July 2020 : proceedings. – California: IJCAI, 2020. – P. 4513–4519.

48. Wiranata, R. The stock exchange prediction using machine learning techniques [Text] / R. Wiranata, A. Djunaidy // Journal of Computer Science and Information. – 2021. – Vol. 14, Issue 2. – P. 91–112.

49. Pabuccu, H. Forecasting the movements of Bitcoin prices: an application of machine learning algorithms [Text] / H. Pabuccu, S. Ongan, A. Ongan // Quantitative Finance and Economics. – 2020. – Vol. 4, Issue 4. – P. 679–692.

50. Язык программирования Python [Электрон. ресурс]. – Режим доступа : <https://web-creator.ru/articles/python>

51. Top 10 Python Libraries for Data Science [Electronic resource]. – Access mode : <https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266>

52. What is fastText? [Electronic resource]. – Access mode : <https://fasttext.cc/docs/en/support.html>

53. Keras [Electronic resource]. – Access mode : <https://keras.io/>

54. PyTorch [Electronic resource]. – Access mode : <https://pytorch.org/>

55. Flask vs Django: What's the Difference Between Flask & Django? [Electronic resource]. – Access mode : <https://www.guru99.com/flask-vs-django.html>

56. Что такое CouchDB? [Электрон. ресурс]. – Режим доступа : <https://uk.theastrologypage.com/couchdb>

57. Amazon DynamoDB [Электрон. ресурс]. – Режим доступа : <https://aws.amazon.com/ru/dynamodb/>

58. What are the key features of MongoDB? [Electronic resource]. – Access mode : <https://www.tutorialsjar.com/key-features-of-mongodb/>

59. Getting Started with Python and MongoDB [Electronic resource]. – Access mode : <https://www.mongodb.com/blog/post/getting-started-with-python-mongodb>

60. Sentiment Analysis for Financial News [Electronic resource]. – Access mode : <https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news>

ДОДАТОК А
Технічне завдання

Вступ

Програма, що розроблюється, є вебзастосунком для проведення аналізу тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень. Областю застосування застосунку є аналіз ситуації на ринку акцій.

A.1 Підстава для розробки

Підставою для розробки є затверджене на засіданні кафедри програмних засобів за протоколом № 435 від 5 листопада 2021 року завдання на дипломний проєкт на тему «Дослідження та програмна реалізація методів аналізу тональності текстових повідомлень про ринок акцій», запропоновану студентом та затверджену з керівником дипломного проєкту.

A.2 Призначення розробки

Вебзастосунок, що розроблюється, призначений для проведення аналізу тональності заголовків текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

Експлуатаційне призначення для використання клієнтами:

- вибір періоду за який час відображаються дані;
- перегляд графіків ціни акцій та оцінок тональності новин однієї обраної компанії;
- перегляд гістограми кількості новин, проаналізованих за день;
- перегляд загальної гістограми оцінок тональності новин декількох обраних компаній;
- перегляд розподілу акцій за тенденціями зміни їх цін.

Функціональне призначення:

- організація взаємодії користувача з системою;

- аналіз тональності заголовків новин ринку акцій;
- категоризація акцій за тенденціями зміни їх цін;
- побудова графіків ціни акцій та оцінок тональності новин;
- побудова гістограми кількості новин проаналізованих за день;
- побудова загальної гістограми оцінок тональності новин;
- побудова таблиці розподілу акцій за тенденціями зміни їх цін.

А.3 Вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Кожна функція повинна бути працездатною і виконувати своє призначення.

До функцій системи належать:

- аналіз тональності заголовків новин ринку акцій;
- категоризація акцій за тенденціями зміни їх цін;
- побудова графіків ціни акцій та оцінок тональності новин однієї обраної компанії;
- побудова гістограми кількості новин проаналізованих за день;
- побудова загальної гістограми оцінок тональності новин декількох обраних компаній;
- побудова таблиці розподілу акцій за тенденціями зміни їх цін.

Вхідною інформацією в залежності від функції та дій користувача є дані, які представляють собою заголовки новин про ринок акції, обрані компанії. А вихідною інформацією є занесені в базу даних результати проведеного аналізу тональності заголовків новин, кількості проаналізованих новин та результати категоризації акцій.

Інтерфейс користувача повинен забезпечити ефективну і просту взаємодію користувача з вебзастосунком.

А.3.2 Вимоги до надійності

Програма повинна розпізнавати ситуації нерозуміння програмою користувача (або користувача програми), при виборі компаній зі списку програма повинна попередньо перевіряти коректність звершеної дії.

Програма повинна здійснювати обробку збійних ситуацій, і таким чином забезпечувати стійке функціонування та максимально швидке відновлення після відмови.

А.3.3 Умови експлуатації

Умовами використання програми користувачем є наявність браузеру на персональному настільному комп'ютері або ноутбуці.

А.3.4 Вимоги до складу і параметрів технічних засобів

Вебзастосунок призначений для роботи тільки в браузерах на персональних настільних комп'ютерах або ноутбуках.

Як апаратно-технічні засоби для виконання програми повинен використовуватися комп'ютер або ноутбук з характеристиками не нижче: процесор з тактовою частотою 1 ГГц або вище, RAM 1 ГБ, 512 МБ дискового простору.

А.3.5 Вимоги до інформаційної та програмної сумісності

Обов'язковими вимогами без яких програма не працюватиме є: Python інтерпретатор версії 3.8 та потрібні для роботи програми бібліотеки (Flask, PyMongo, Pandas, Plotly, FastText, Keras та інші вказані у файлі requirements.txt).

А.3.6 Вимоги до програмної документації

Програмну документацію складає технічне завдання. Технічне завдання (ТЗ) – це затверджений документ, на основі якого виконується розробка проєкту. ТЗ повинно бути зрозумілим як розробнику, так і замовнику, і повинно відображати відповіді на всі питання щодо планування розробки проєкту.

А.3.7 Стадії та етапи розробки

Стадіями розробки є:

- розробка і утвердження технічного завдання;
- вибір засобів розробки (мови програмування, потрібних бібліотек та фреймворків, системи управління базою даних);
- проєктування архітектури системи;
- моделювання схеми бази даних;
- розробка програмного забезпечення.

А.3.8 Порядок контролю та приймання

Дипломна кваліфікаційна робота магістра допускається до захисту керівником і нормоконтролером, затверджується завідувачем кафедрою і захищається в присутності членів екзаменаційної комісії.

ДОДАТОК Б
Текст програми

Б.1 Модуль app.py

```

from flask import Flask, render_template, request

from data_collectors import COMPANIES_LIST
from plots import plot_company_sentiment, plot_companies_sentiment
from stock_categorization import categorize_stocks

app = Flask(__name__)

@app.route('/')
@app.route('/single_company')
def index():
    ticker = request.args.get('company') if 'company' in request.args
else COMPANIES_LIST[0]

    plot_company_sentiment(company=ticker)

    return render_template(
        "index.html",
        sentiment_graph=f"{ticker}_daily_sent_stock.html",
        ticker=ticker,
        companies=COMPANIES_LIST,
        single_ticker=True
    )

@app.route('/several_companies', methods=['GET', 'POST'])
def several_companies():
    if request.method == 'POST':
        tickers = request.form.getlist('checkbox')
    else:
        tickers = COMPANIES_LIST[:3]

    plot_companies_sentiment(companies=tickers)

    return render_template(
        "index.html",
        companies=COMPANIES_LIST,
        several_companies=True,
        tickers=tickers
    )

@app.route('/classification')
def categorization():
    categorization_results, forecast_date = categorize_stocks()

    return render_template(
        "index.html",
        categorization = True,
        forecast_date = forecast_date,
        categorization_results = categorization_results,
    )

if __name__ == "__main__":
    app.run(port=5000, debug=True)

```

Б.2 Модуль db.py

```
import pymongo
import pandas as pd

db_client = pymongo.MongoClient(
    'mongodb+srv://admin:admin@cluster0.zu4dx.mongodb.net/test'
)

def update_db(new_data, db_name, collection_name):
    db = db_client[db_name]
    collection = db[collection_name]
    prev_data = pd.DataFrame(list(collection.find({})))

    new_dates_list = []
    for date in list(new_data['date']):
        if date not in list(prev_data['date']):
            new_dates_list.append(date)

    new_data = new_data[new_data['date'].isin(new_dates_list)]
```

Б.3 Модуль updater.py

```
import time
import functools

import pandas as pd

from db import update_db
from data_collectors import COMPANIES_LIST, collect_news
from fasttext_analysis import fasttext_analysis

def updater():
    merged_sentiments = []
    merged_volumes = []

    for company in COMPANIES_LIST:
        merged_headlines_daily = collect_news(company)
        df_daily_sentiment, df_volumes =
fasttext_analysis(preprocess_data(merged_headlines_daily))
        df_daily_sentiment["date"] =
df_daily_sentiment["date"].apply(lambda x: x.strftime('%Y-%m-%d'))

        df = df_daily_sentiment.rename(columns =
{'probability':f'{company}'})
        merged_sentiments.append(df)
        df = df_volumes.rename(columns = {'positive':
f'{company}_positive', 'negative': f'{company}_negative'})
        merged_volumes.append(df)

        update_db(new_data=df_daily_sentiment,
db_name='stock_sentiment_db', collection_name=f"{company}_daily_sentiment")
        update_db(new_data=df_volumes, db_name='sentiment_volume_db',
collection_name=f"{company}_volumes")

    time.sleep(60)
```

```

        df_merged_sentiments = functools.reduce(
            lambda left, right: pd.merge(left, right, on='date',
            how='inner'),
            merged_sentiments
        )
        update_db(new_data=df_merged_sentiments,
        db_name='stock_sentiment_db', collection_name="merged_sentiments")

        df_merged_volumes = functools.reduce(
            lambda left, right: pd.merge(left, right, on='date',
            how='inner'),
            merged_volumes
        )
        update_db(new_data=df_merged_volumes, db_name='sentiment_volume_db',
        collection_name="merged_volumes")

if __name__ == "__main__":
    updater()

```

Б.4 Модуль data_collectors.py

```

import json
from collections import defaultdict
from datetime import datetime, timedelta
from urllib.request import urlopen, Request

import finnhub
import pandas as pd
import yfinance as yf
from newsapi import NewsApiClient

COMPANIES_DICT = {
    "amazon": "amzn",
    "tesla": "tsla",
    'microsoft': 'msft',
    'att': 't',
    'mastercard': "ma",
    'google': "GOOGL",
    'cisco': 'csc',
    'general motors': "gm",
    'twitter': "twtr",
    "apple": "aapl",
    'mcdonald\'s': 'mcd',
    'nvidia': 'nvda',
    'pfizer': "pfe",
    'facebook': "fb",
    "fedex": "fdx",
}

COMPANIES_LIST = COMPANIES_DICT.keys()

def merge_data(data):
    merged_headlines = defaultdict(list)

    for headlines in data:
        for k, v in headlines.items():
            merged_headlines[k].extend(v)

    return merged_headlines

```

```

def collect_news(stock):
    date = datetime.now()

    polygon_headlines_daily = get_news_from_polygon(stock, date)
    newsapi_headlines_daily = get_news_from_newsapi(stock, date)
    finnhub_headlines_daily = get_news_from_finnhub(stock, date)

    return merge_data(
        (
            newsapi_headlines_daily,
            finnhub_headlines_daily,
            polygon_headlines_daily,
        )
    )

def get_stock_prices_yfinance(stock, days_quan):
    date = datetime.now()

    ticker = yf.Ticker(COMPANIES_DICT[company])

    ticker_stock_daily = ticker.history(
        start=date - timedelta(days=days_quan),
        end=date,
    ).reset_index()
    ticker_stock_daily.rename(columns = {'Date':'date'}, inplace = True)
    ticker_stock_daily["date"] = ticker_stock_daily["date"].apply(lambda
x: x.strftime('%Y-%m-%d'))

    return ticker_stock_daily

def get_news_from_polygon(stock, date):
    key = ""

    start_date = (date - timedelta(days=31)).strftime("%Y-%m-%d")
    end_date = date.strftime("%Y-%m-%d")

    url = (
        f'https://api.polygon.io/v2/reference/news?limit=1000&ticker='
        f'{COMPANIES_DICT[company].upper()}&published_utc.gt={start_date}'
        f'&published_utc.lt={end_date}&apiKey={key}'
    )

    req = Request(url=url)
    response = urlopen(req).read()

    polygon_headlines_daily = defaultdict(list)

    for res in json.loads(response.decode('utf-8'))['results']:
        date_key = pd.to_datetime(res["published_utc"]).strftime("%Y-%m-
%d")
        polygon_headlines_daily[date_key].append(f'{res["title"]}')

    return polygon_headlines_daily

def get_news_from_newsapi(stock, date):
    newsapi = NewsApiClient(api_key="")

    all_articles = newsapi.get_everything(
        q=company,
        qintitle=company,
        from_param=date - timedelta(days=31),
        to=date,

```

```

        language="en",
    )

    newsapi_headlines_daily = defaultdict(list)

    for n in all_articles["articles"]:
        date_key = pd.to_datetime(n["publishedAt"]).strftime("%Y-%m-%d")
        newsapi_headlines_daily[date_key].append(f"{n['title']}")

    return newsapi_headlines_daily

def get_news_from_finnhub(stock, date):
    finnhub_client = finnhub.Client(api_key="")

    news = finnhub_client.company_news(
        COMPANIES_DICT[company],
        _from=(date - timedelta(days=31)).strftime("%Y-%m-%d"),
        to=date.strftime("%Y-%m-%d")
    )

    finnhub_headlines_daily = defaultdict(list)

    for n in news:
        date_key = date.fromtimestamp(n["datetime"]).strftime("%Y-%m-%d")
        finnhub_headlines_daily[date_key].append(f"{n['headline']}")

    return finnhub_headlines_daily

```

Б.5 Модуль fasttext_analysis.py

```

import re
from string import punctuation

import spacy
import fasttext
import pandas as pd

nlp = spacy.load("en_core_web_sm", disable=["parser", "ner"])
stop = nlp.Defaults.stop_words

def fasttext_analysis(df):
    model = fasttext.load_model("model.bin")
    df_scores = []
    pos = []
    neg = []

    for new_headlines_list, d in zip(df["new_headlines"], df['date']):
        sentence_scores = []

        for new_hl in new_headlines_list:
            score = model.predict(new_hl)[1][0]
            sentiment = model.predict(new_hl)[0][0]

            if score:
                if "negative" in sentiment:
                    score = -1 * score
                sentence_scores.append(score)

        pos.append(len([s for s in sentence_scores if s > 0]))
        neg.append(len([s for s in sentence_scores if s < 1]))

```

```

df_scores.append(sum(sentence_scores) / len(sentence_scores))

df_volumes = pd.DataFrame({'date': list(df['date']), 'positive': pos,
'negative': neg})

df["probability"] = df_scores

df.drop('headlines', axis=1, inplace=True)
df.drop('new_headlines', axis=1, inplace=True)

return df, df_volumes

```

Б.6 Модуль fasttext_training.py

```

import csv
import spacy
import fasttext
import pandas as pd
from nltk.corpus import stopwords

nlp = spacy.load("en_core_web_sm", disable=["parser", "ner"])
stop = stopwords.words("english")

def fasttext_training(data):
    data[['__label__', 'text']].to_csv('data_train.txt')
    model = fasttext.train_supervised(
        input='data_train.txt',
        wordNgrams = 3,
        epoch=50,
        bucket=2000000,
        loss='softmax',
        lr=1.0
    )
    model.save_model("model.bin")

    return model

if __name__ == "__main__":
    data = pd.read_csv("data_train.csv")
    fasttext_training(data)

```

Б.7 Модуль preprocessing.py

```

def preprocess_data(data):
    data = pd.read_csv(data)

    data["text"] = data["text"].apply(lambda hl: " ".join(hl.lower() for
hl in hl.split()))
    data["text"] = data["text"].apply(lambda hl: re.sub(r'http\S+', '',
hl))
    data["text"] = data["text"].str.replace("[^\w\s]", "") #punctuation
    data["text"] = data["text"].apply(lambda hl: " ".join(hl for hl in
hl.split() if hl not in stop))
    data["text"] = data["text"].apply(lambda hl: " ".join([token.lemma_
for token in nlp(hl)]))

    return data

```

Б.8 Модуль stock_categorization.py

```

from keras.layers import LSTM
from keras.layers import Dense
from keras.models import Sequential

import pandas as pd
from data_collectors import COMPANIES_LIST
from matplotlib import pyplot

from numpy import array

def predict_stock_attitude(sentiment):
    N = len(sentiment) - 7
    sentiments = array(list(sentiment["probability"][:N]))

    n_steps = 7
    X, y = split_sequence(sentiments, n_steps)
    n_features = 1
    X = X.reshape((X.shape[0], X.shape[1], n_features))

    model = Sequential()
    model.add(
        LSTM(
            units=100,
            activation="tanh",
            input_shape=(n_steps, n_features),
            return_sequences=True,
        )
    )
    model.add(LSTM(100, activation="tanh"))
    model.add(Dense(1))

    model.compile(loss="mse", metrics=["mean_squared_error"])
    history = model.fit(X, y, validation_split=0.33, epochs=200,
verbose=0)

    pyplot.title('Середньоквадратична похибка')
    pyplot.plot(history.history['mean_squared_error'],
label='тренування')
    pyplot.plot(history.history['val_mean_squared_error'],
label='тестування')
    pyplot.legend()
    pyplot.show()

    sentiments_val = list(sentiment["probability"])[N:]
    x_input = array(sentiments_val)
    x_input = x_input.reshape((1, len(sentiments_val), n_features))
    sentiment_prediction = model.predict(x_input, verbose=0)

    return sentiment_prediction[0][0]

def categorize_stocks():
    stock_categorization = {}

    for company in COMPANIES_LIST:
        sentiment = pd.read_csv(f"csv/{company}_daily_sentiment.csv")
        sentiment = array(list(sentiment["probability"])[-7])
        prediction = predict_stock_attitude(sentiment)

        if prediction > sentiment:

```

```

        attitude = "rise"
    else:
        attitude = "drop"

    diff = abs(prediction - sentiment)

    if diff >= 0.7:
        grade = "significant"
    elif diff < 0.7 or diff >= 0.2:
        grade = "slight"
    else:
        grade = "almost intact"

    stock_categorization[company] = f"{grade}-{attitude}"

    return stock_categorization

if __name__ == "__main__":
    categorize_stocks()

```

Б.9 Модуль plots.py

```

import plotly.graph_objects as go
from plotly.subplots import make_subplots

def plot_company_sentiment(df_daily_sentiment, ticker_stock, company,
time, df_volumes):
    price_line = go.Scatter(
        x=ticker_stock["date"],
        y=list(ticker_stock["Close"]),
        name="Price",
        mode='lines+markers',
        yaxis="y1"
    )
    sentiment_line = go.Scatter(
        x=df_daily_sentiment['date'],
        y=list(df_daily_sentiment['probability']),
        name='Sentiment',
        mode='lines+markers',

marker=dict(color=get_colors(list(df_daily_sentiment['probability'])),
            line=dict(width=1,
                    color='#117791')),
        yaxis="y2",
        line=dict(color='#117791')
    )
    pos_sentiment_vol_bar = go.Bar(
        x=df_daily_sentiment['date'],
        y=list(df_volumes['positive']),
        text=list(df_volumes['positive']),
        name='Positive sentiment volume',
        yaxis="y3",
        marker_color='#00cc96'
    )
    neg_sentiment_vol_bar = go.Bar(
        x=df_daily_sentiment['date'],
        y=list(df_volumes['negative']),
        text=list(df_volumes['negative']),
        name='Negative sentiment volume',
        yaxis="y3",
        marker_color='#ef553b'

```

```

)

fig = make_subplots()

fig.add_trace(price_line)
fig.add_trace(sentiment_line)
fig.add_trace(neg_sentiment_vol_bar)
fig.add_trace(pos_sentiment_vol_bar)

fig.update_layout(
    barmode='stack',
    height=630, width=1400,
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1,
                    label="1m",
                    step="month",
                    stepmode="backward"),
                dict(count=7,
                    label="1w",
                    step="day",
                    stepmode="backward"),
                dict(step="all")
            ])
        ),
        rangeslider=dict(
            visible=True,
        ),
        type="date"
    ),
    yaxis1=dict(domain=[0.35, 1], side='left', overlaying='free'),
    yaxis2=dict(domain=[0.35, 1], side='right', overlaying='y1'),
    yaxis3=dict(domain=[0, 0.25], side="left", overlaying="y3"),
)

fig.write_html(f"./templates/{company}_{time}_sent_stock.html")

def plot_companies_sentiment(df_merged_sentiments, df_merged_volumes,
companies):
    fig = make_subplots()

    data = []
    for i, company in enumerate(companies):
        data.append(go.Bar(
            x=df_merged_sentiments['date'],
            y=list(df_merged_sentiments[f'{company}']),
            name=f'{company}',
            offsetgroup=i+1,
            yaxis="y1",
        ))

    for i, company in enumerate(companies):
        data.append(go.Bar(
            x=df_merged_volumes['date'],
            y=list(df_merged_volumes[f'{company}_positive']),
            name=f'{company.title()} positive news count',
            offsetgroup=i+1,
            yaxis="y2",
        ))
        data.append(go.Bar(
            x=df_merged_volumes['date'],

```

```

        y=list(df_merged_volumes[f'{company}_negative']),
        name=f'{company.title()} negative news count',
        offsetgroup=i+1,
        yaxis="y2",
    ))

fig = go.Figure(data)

fig.update_layout(
    height=630, width=1400,
    bargap=0.4,
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1,
                    label="1m",
                    step="month",
                    stepmode="backward"),
                dict(count=7,
                    label="1w",
                    step="day",
                    stepmode="backward"),
                dict(step="all")
            ])
        ),
        rangeslider=dict(
            visible=True,
        ),
        type="date"
    ),
    yaxis1=dict(domain=[0.6, 1], side='left'),
    yaxis2=dict(domain=[0, 0.5], side='left'),
)

fig.write_html("./templates/graph.html")

```

Б.10 Файл index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title> </title>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta http-equiv="x-ua-compatible" content="ie=edge">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
        integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous">
    </script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.
js"
        integrity="sha384-
LtrjvnR4Twt/qOuYxE721u19sVFLVSA4hf/rRt6PrZTmiPltdZcI7q7PXQBYTKyf"
crossorigin="anonymous">
    </script>

```

```

<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.2/css/all.css">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap">
<link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">
<link
href="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.1/css/mdb.min.css"
rel="stylesheet">
</head>
<body>
<div class="card w-100">
<div class="card-header font-weight-bold row m-0 justify-content-between">
<div>
{% if single_ticker %}
<div class="dropdown">
<button class="btn btn-primary dropdown-toggle"
type="button" id="dropdownMenuButton"
data-mdb-toggle="dropdown" aria-
expanded="false">
{{ticker}}
</button>
<ul class="dropdown-menu" aria-
labelledby="dropdownMenuButton">
{% for company in companies %}
<li><a class="dropdown-item"
href="/single_company?company={{ company }}"
{{ company|title }}
</a></li>
{% endfor %}
</ul>
</div>
{% elif several_companies %}
<div class="dropdown">
<button class="btn btn-primary dropdown-toggle"
type="button" id="dropdownMenuButton"
data-mdb-toggle="dropdown" aria-
expanded="false">
Select
</button>
<form class="checkbox-form dropdown-menu" aria-
labelledby="dropdownMenuButton" method="POST"
action="/several_companies">
{% for company in companies %}
<div class="form-check">
{% if company in tickers %}
<input class="form-check-input"
type="checkbox" id="inlineCheckbox1" value="{{company}}"
name="checkbox" checked>
{% else %}
<input class="form-check-input"
type="checkbox" id="inlineCheckbox1" value="{{company}}"
name="checkbox">
{% endif %}

<label class="form-check-label"
for="inlineCheckbox1">
{{ company|title }}
</label>
</div>
{% endfor %}

```

```

                                <div class="text-center">
                                    <input class="btn btn-primary submit-btn"
type="submit" value="Show">
                                </div>
                            </form>
                        </div>
                    {% else %}

                        {% endif %}
                    </div>
                <div class="menu">
                    <div class="col m-0 p-1 justify-content-end">
                        <a class="btn btn-{% if single_ticker %}success{%
else %}primary{% endif %} single-company"
                            href="/single_company">
                            <i class="fas fa-check-square"></i>
                        </a>
                    </div>
                    <div class="col m-0 p-1 justify-content-end">
                        <a class="btn btn-{% if several_companies %}success{%
else %}primary{% endif %} several-companies"
                            href="/several_companies">
                            <i class="fas fa-tasks"></i>
                        </a>
                    </div>
                    <div class="col m-0 p-1 justify-content-end">
                        <a class="btn btn-{% if classification %}success{%
else %}primary{% endif %} classification"
                            href="/classification">
                            <i class="fas fa-table"></i>
                        </a>
                    </div>
                </div>
            </div>
            <div class="card-body">
                {% if single_ticker %}
                    {% include sentiment_graph %}
                {% elif several_companies %}
                    {% include 'graph.html' %}
                {% else %}
                    <div>
                        <p class="table-name">
                            <b>Forecast from {{ forecast_date }}</b>
                        </p>
                        <div class="classification-table-head">
                            {% for type in types %}
                                <p>
                                    <b>{{ type|replace('-', ' ') }}</b>
                                </p>
                            {% endfor %}
                        </div>
                        <div class="classification-table">
                            {% for type in types %}
                                <div>
                                    {% for res in
classification_results.items() %}
                                        {% if res[1] == type %}
                                            <p class="{{ type }}">
                                                <b>{{ res[0].title() }}</b>
                                            </p>
                                        {% endif %}
                                    {% endfor %}
                                </div>
                            {% endfor %}
                        </div>
                    </div>
                </div>
            </div>

```

```

                                {% endfor %}
                            </div>
                        </div>
                    {% endif %}
                </div>
            </div>
        </div>
    </body>

    <script>
        $('[data-toggle="tooltip"]').tooltip({
            container: 'body',
            trigger: 'hover',
            delay: { "show": 500, "hide": 100 },
            html: true
        });

        $('.dropdown-toggle').dropdown()
        $(document).click(function () {
            $(".dropdown-menu").removeClass('show');
        });
    </script>

</html>

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Титульний слайд

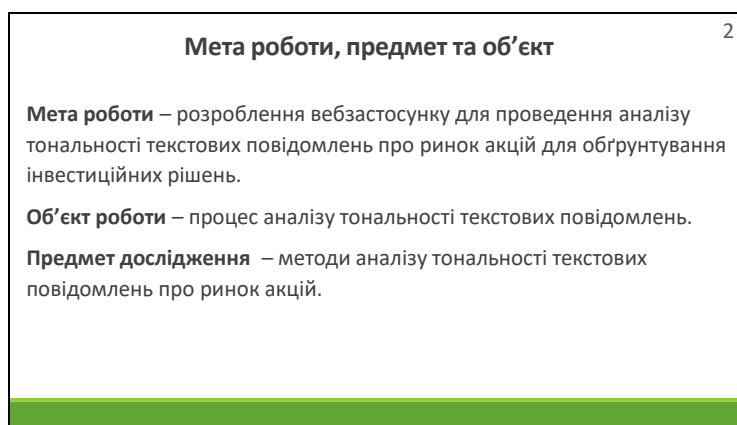


Рисунок В.2 – Слайд «Мета роботи, предмет та об'єкт»

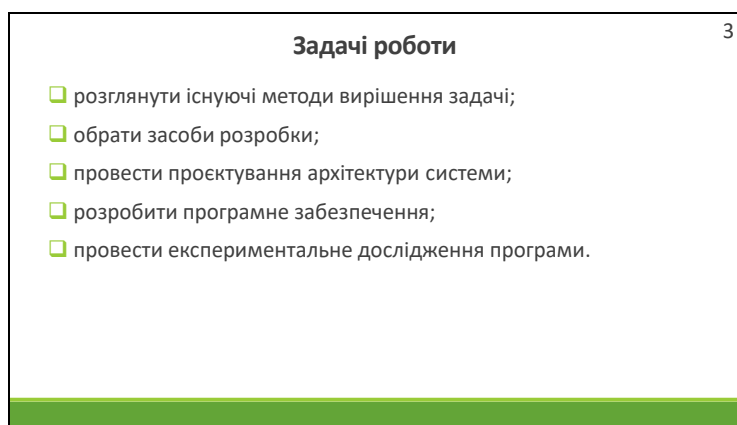


Рисунок В.3 – Слайд «Задачі роботи»

Огляд існуючих підходів вирішення задачі (методи, засновані на словниках та правилах)			4
Підхід	Переваги	Недоліки	
Метод, заснований на словнику	простота у застосуванні; мінімальні вимоги до обчислювальних ресурсів	трудомісткість процесу створення словника; неможливість обробляти заперечення, помилки в реченнях; ігнорує контекст	
Метод, заснований на правилах	задовільна точність при наявності хорошої бази правил	трудомісткість процесу створення набору правил	

Рисунок В.4 – Слайд «Огляд існуючих підходів вирішення задачі
(методи, засновані на словниках та правилах)»

Огляд існуючих підходів вирішення задачі (методи машинного навчання з вчителем)			5
Підхід	Переваги	Недоліки	
Наївний баєс	простота у розробці та тестуванні; висока швидкість тренування; помірні вимоги до пам'яті	дуже залежить від заданих на вхід даних	
Метод опорних векторів	потребує досить невеликого набору даних	не працює, якщо класи лінійно нероздільні; помилка в тренувальному наборі даних дуже відображається на результатах	
Метод випадкового лісу	можливість ефективно обробляти дані з великою кількістю ознак і класів та висока масштабованість	скхильність до перенавчання; великий розмір моделей	
FastText класифікатор	простота у застосуванні; мінімальні вимоги до обчислювальних ресурсів; точність результатів на рівні моделей глибинного навчання	помилка в тренувальному наборі даних може відобразитися на результатах	

Рисунок В.5 – Слайд «Огляд існуючих підходів вирішення задачі (методи
машинного навчання з вчителем)»

Огляд існуючих підходів вирішення задачі (методи глибинного навчання та машинного навчання без вчителя)			6
Підхід	Переваги	Недоліки	
CNN	має потенціал видавати дуже точні результати; звертає увагу на контекст тексту	для якісного навчання вимагає великого тренувального набору даних, багато часу та обчислювальних ресурсів	
RNN LSTM			
BERT			
Машинне навчання без вчителя	не потребує попередньо розмічених даних; може помічати якісь приховані шаблони, що можуть в подальшому мати велике значення	для якісного навчання вимагає великого обсягу даних, багато часу та обчислювальних ресурсів; часто менша точність результатів	

Рисунок В.6 – Слайд «Огляд існуючих підходів вирішення задачі (методи
глибинного навчання та машинного навчання без вчителя)»



Рисунок В.7 – Слайд «Метод визначення тенденцій зміни цін акцій»

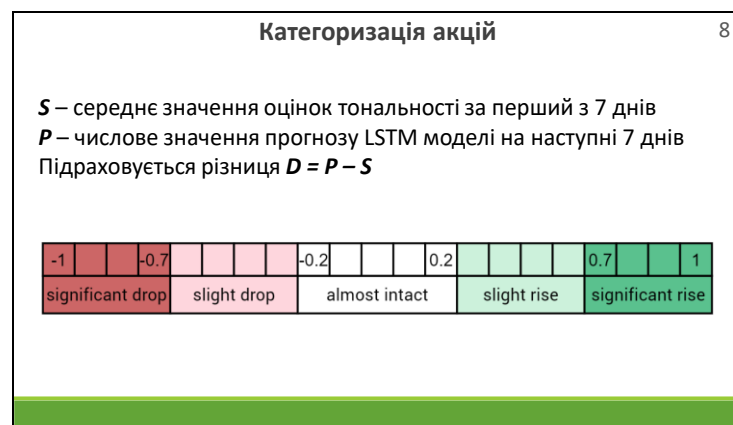


Рисунок В.8 – Слайд «Категоризація акцій»

Вибір мови програмування 9

Мова програмування / Критерії	Python	R	MATLAB	JavaScript	Java
Активна спільнота	+	–	–	+	+
Докладна документація	+	+	+	+	+
Зручні засоби для аналізу даних	+	–	–	+/-	+/-
Зручні засоби для візуалізації аналізу даних	+	+/-	+/-	+	–
Бібліотеки для науки про дані	+	+	+	+	+
Зручні засоби для аналізу тональності текстів	+	–	+/-	+	+/-

Рисунок В.9 – Слайд «Вибір мови програмування»



Рисунок В.10 – Слайд «Структура програмної системи»

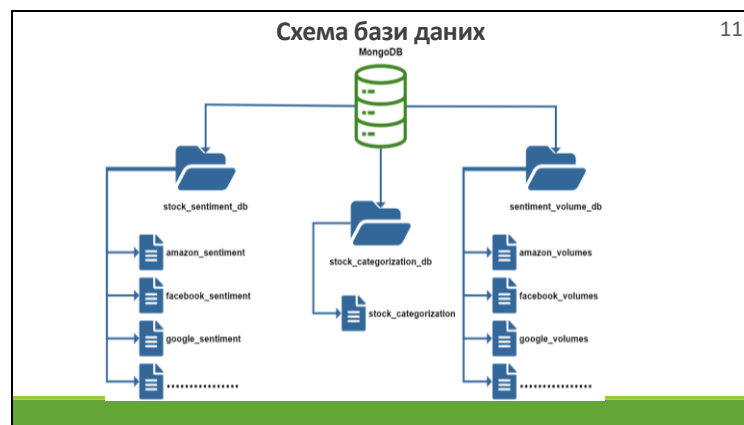


Рисунок В.11 – Слайд «Схема бази даних»



Рисунок В.12 – Слайд «Схема функціонування програми»

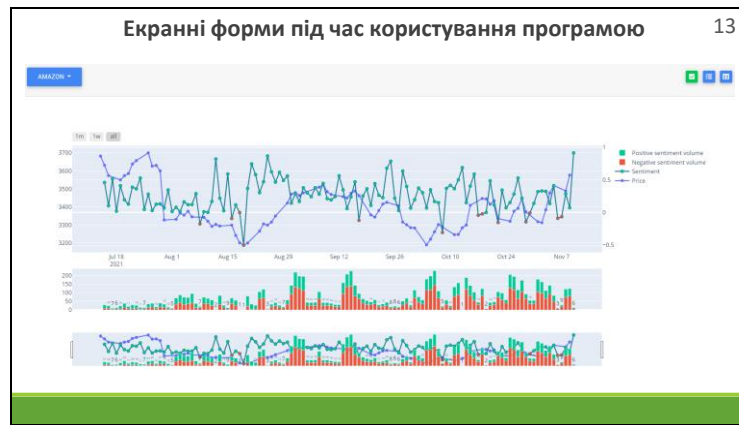


Рисунок В.13 – Слайд «Екранні форми під час користування програмою 1»



Рисунок В.14 – Слайд «Екранні форми під час користування програмою 2»



Рисунок В.15 – Слайд «Екранні форми під час користування програмою 3»

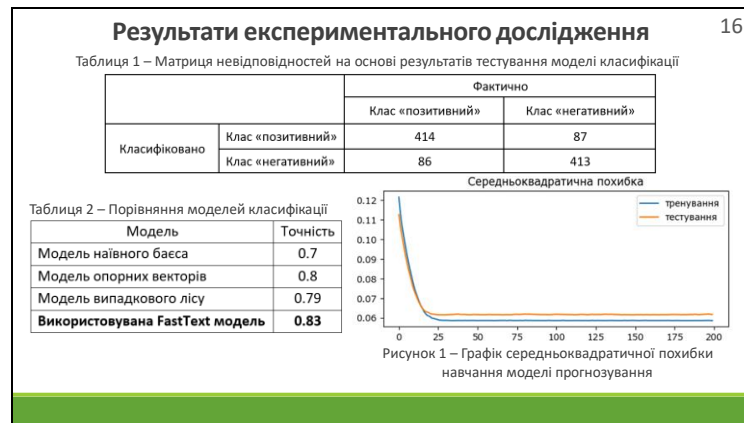


Рисунок В.16 – Слайд «Результати експериментального дослідження»

17

Висновки

За результатами роботи:

- проаналізовано існуючі підходи вирішення задачі;
- запропоновано метод визначення тенденцій зміни цін акцій;
- обрано засоби розробки;
- проведено проектування архітектури системи;
- розроблено програмне забезпечення;
- проведено експериментальне дослідження програми.

Наукова новизна дипломної роботи полягає у розробленому методі визначення тенденцій зміни цін акцій, який характеризується використанням моделей FastText та LSTM і дозволяє розв'язати задачу категоризації акцій і виконати аналіз тональності текстових повідомлень про ринок акцій для обґрунтування інвестиційних рішень.

Рисунок В.17 – Слайд «Висновки»

18

Перелік публікацій

- Чорнобель Д.С. Розробка концепту «Розумний будинок» / Д.С. Чорнобель, О.М. Дроздов, О.Є. Мяло // Тиждень науки : Науково-практична конференція, Запоріжжя, квітень 2018 р. : тези доповідей / Редкол. : В. В. Наумик (відпов. ред.) Електрон. дані. – Запоріжжя : ЗНТУ, 2018.
- Льовкін В.М. Програмна реалізація telegram-бота для підтримки студентів факультету комп'ютерних наук / В.М. Льовкін, Д.С. Чорнобель // Тиждень науки : Науково-практична конференція, Запоріжжя, квітень 2020 р. : тези доповідей / Редкол. : В. В. Наумик (відпов. ред.) Електрон. дані. – Запоріжжя : НУ "Запорізька політехніка", 2020.
- Чорнобель, Д.С. Програмна реалізація Telegram-бота для підтримки студентів / Д.С. Чорнобель, В.М. Льовкін // Матеріали XII міжнародної науково-практичної конференції «Free and Open Source Software», Харків, 17-19 листопада 2020 р. – Харків : ХНУБА, 2020. – С. 36.

Рисунок В.18 – Слайд «Перелік публікацій»