

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

**«Система прогнозування стану фінансових ринків на
основі лінгвістичної моделі мережі інтернет.»**

Виконав студент групи ПЗм-22-1
МЄДВЄДСВ Віталій Володимирович

Керівник роботи:
АНТОНЮК Дмитро Сергійович

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«18» вересня 2023 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **МЄДВЄДЄВ Віталій Володимирович**

Керівник роботи: **АНТОНЮК Дмитро Сергійович**

Тема роботи: **«Система прогнозування стану фінансових ринків на основі лінгвістичної моделі мережі інтернет.»**,

затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**

Вихідні дані для роботи: **предметом дослідження є засоби розробки системи прогнозування станів фінансових ринків на основі лінгвістичної моделі мережі з використанням методів машинного навчання та впровадження отриманої системи в процес навчання (навчальний симулятор); об'єктом дослідження є методи використання ШІ та машинного навчання в прогнозуванні та інтеграція їх з навчальними програмами.**

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Антонюк Д.С.	12.09.2023р.	12.09.2023р.
2	Антонюк Д.С..	11.10.2023р.	11.10.2023р.
3	Антонюк Д.С.	31.11.2023р.	31.11.2023р.

Дата видачі завдання: **«12» вересня 2023 р.**

РЕФЕРАТ

Завдання кваліфікаційної магістерської роботи полягає в науковому дослідженні та обґрунтуванні вибору моделей та методів, що будуть використані для розробки та вдосконалення системи прогнозування станів фінансових ринків та впровадження її в симулятор фінансового середовища, з метою надання максимально наближених до реальності умов навчання для збільшення ступеню засвоєння навичок з подальшим їх використанням в умовах реальних ринків. Через складність аналізу великих масивів даних (англ. BigData) систему розроблено з використанням технологій машинного навчання, а саме методу класифікації для прогнозування на основі лінгвістичних моделей.

Під час написання кваліфікаційної роботи було визначено та проаналізовано найбільш перспективні алгоритми машинного навчання згідно до теми кваліфікаційної роботи. Зроблено опис їх роботи.

Для обраних алгоритмів було проведено дослідження їх якості роботи на даних отриманих через мережу Інтернет. Унаслідок чого було визначено найбільш відповідні алгоритми для обраної теми кваліфікаційної роботи. Обґрунтовано вибір алгоритмів Naive Bayes, RNN, RandomForestClassifier

Кваліфікаційна робота складається з програмного та теоретичного комплексу системи та пояснювальної записки. Пояснювальна записка до кваліфікаційної магістерської роботи нараховує 55 сторінок, 22 ілюстрації, 1 таблицю, 1 формулу та список використаних джерел.

КЛЮЧОВІ СЛОВА: СИМУЛЯТОР, ОБЛІГАЦІЇ, ІНВЕСТИЦІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, МОДЕЛЬ, АНАЛІТИКА ДАНИХ. ФІНАНСОВІ РИНКИ, ЛІНГВІСТИЧНА МОДЕЛЬ, ВЕЛИКІ ДАНІ, ІНТЕРНЕТ.

ABSTRACT

The task of the qualifying master's thesis consists in the scientific research and justification of the choice of models and methods that will be used to develop and improve the system of forecasting the state of financial markets and its implementation in the simulator of the financial environment, with the aim of providing training conditions as close as possible to reality to increase the degree of assimilation of skills from their further use in the conditions of real markets. Due to the complexity of analyzing large data sets (English: BigData), the system was developed using machine learning technologies, namely the classification method for prediction based on linguistic models.

During the writing of the qualification work, the most promising machine learning algorithms were identified and analyzed according to the topic of the qualification work. A description of their work is made.

For the selected algorithms, a study of their quality of work was conducted on data received via the Internet. As a result, the most appropriate algorithms for the chosen topic of the qualification work were determined. The selection of Naive Bayes, RNN, RandomForestClassifier algorithms is justified

The qualification work consists of the programmatic and theoretical complex of the system and an explanatory note. The explanatory note to the qualifying master's thesis has 55 pages, 22 illustrations, 1 table, 1 formula and a list of used sources.

KEY WORDS: SIMULATOR, BONDS, INVESTMENT, ARTIFICIAL INTELLIGENCE, MODEL, DATA ANALYTICS. FINANCIAL MARKETS, LINGUISTIC MODEL, BIG DATA, INTERNET.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ	9
1.1 Перелік задач, що підлягають автоматизації згідно теми	9
1.2 Аналіз аналогів програмного продукту	10
1.3 Архітектура та узагальнена структура майбутнього програмного комплексу.	14
1.4 Інструментальні засоби реалізації програмного продукту.....	15
1.5 Інструменти проектування потоків даних за стандартами DFD та IDEF0	16
1.6 Аналіз методів машинного навчання.....	19
Висновки до першого розділу	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ КОРИСТУВАЧІВ. ІМПЛЕМЕНТАЦІЯ ПРОГНОЗУВАННЯ	24
2.1 Формування варіантів використання системи.....	24
2.2 Проектування варіантів використання системи.	27
2.3 Програмна реалізація компонентів підсистем.....	30
Висновки до другого розділу	36
РОЗДІЛ 3. ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ МОДЕЛІ. ФОРМУВАННЯ АНАЛІТИЧНИХ ЗВІТІВ ТА АНАЛІЗ ВІДХИЛЕНЬ.....	37
3.1 Аналіз відхилень даних.....	37
3.2 Проектування алгоритму аналізу лінгвістичної моделі мережі інтернет	39
3.3 Проектування алгоритму математичної системи	43
3.4 Проектування алгоритму роботи інтелектуального асистенту.....	45
3.5 Проектування алгоритму визначення тенденції на ринку.....	47
Висновки до третього розділу	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ОВДП – облігації внутрішньої державної позики

ІІІ – Штучний інтелект

МН – Машинне навчання

API – Application Programming Interface

JSON – JavaScript Object Notation

CSV – Comma-separated values

BD - BigData

ВСТУП

Актуальність теми полягає в тому, що в умовах сучасного світу ми спостерігаємо швидкий розвиток технологій та постійно зростаючий обсяг інформації на фоні кризових умов. В таких умовах фінансова грамотність стає критично важливою для всіх верств суспільства для збереження власних статків та мінімізації втрат при кризових умовах. Разом з цим підвищується темп життя та обсяги інформації, які необхідно враховувати для формування ефективної стратегії. Все це призводить до потреби в механізмах розрахунку потенційної вигоди та ризиків задля збереження власного капіталу особливо для інвесторів-початківців, які не мають достатнього досвіду та компетенцій для формування ефективної інвестиційної стратегії.

Проблемою яка розглянута в кваліфікаційній роботі є необхідність створення ефективного інструменту, що допоможе, як новачкам, які ніколи не працювали з інвестиціями так і професійним гравцям на ринку розуміти ризики, які можуть їх спіткати в умовах багатовекторної системи, що залежна від часу, станів інших ринків та соціогуманітарного фону. Таким чином симулятор, який реалізовує не тільки базові механізми розрахунків за формулами, але і проводить корегування моделі на основі узагальненого коефіцієнту прогнозування, може стати потужним інструментом, що допоможе формувати більш ефективні стратегії.

Одночасно з цим аналітика даних дозволяє реалізовувати ad-hoc підхід до кожного користувача, надаючи персоналізовані рекомендації та максимально точні та коректні на даний момент дані про ризики, кращі та гірші сценарії тощо.

Метою дослідження є проектування системи для аналізу лінгвістичної моделі мережі Інтернет, узагальнення її на основі реальних даних та подальша інтеграції у навчальний симулятор з використанням механізмів ІІІ.

Завданням кваліфікаційної роботи є:

- Формалізувати задачу прогнозування станів фінансових ринків на основі лінгвістичної моделі мережі Інтернет;
- Розробити алгоритм прогнозування станів фінансових ринків з виведенням допоміжних коефіцієнтів;
- Розробити алгоритм математичного розрахунку статистичних показників з урахуванням допоміжних коефіцієнтів;
- Розробити алгоритм програми асистента здатного надавати підказки з урахуванням допоміжних коефіцієнтів розрахованих алгоритмом прогнозування.

Об'єктом дослідження є методи аналізу лінгвістичних моделей та використання ШІ та машинного навчання в прогнозуванні. Інтеграція їх з навчальними програмами.

Предметом дослідження є засоби розробки системи прогнозування станів фінансових ринків на основі лінгвістичної моделі мережі Інтернет з використанням методів машинного навчання та впровадження отриманої системи в процес навчання (навчальний симулятор).

Практичне значення кваліфікаційної роботи полягає у:

- Прогнозуванні станів реальних фінансових ринків;
- Контролю за якістю навчання користувача та надання наближених до реальності умов;
- вдосконалення наявного веб-симулятора;

Апробація результатів роботи: Матеріали дослідження за тематикою кваліфікаційної роботи було опубліковано у V Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», 01–02 грудня 2022 р. [22], V Всеукраїнській науково-практичній інтернет-конференції студентів, аспірантів та молодих вчених «Сучасні інформаційні системи і технології.» 30 листопада 2022 р. [23], а також опублікована стаття у «Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки» 2022 р. [21].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ

1.1 Перелік задач, що підлягають автоматизації згідно теми

До переліку задач, що підлягають автоматизації у системі входять:

1. **Збір та обробка даних:** Автоматизування процесу збору даних з доступних джерел мережі Інтернет з залученням медіа вертикального та горизонтального типу, та виведення загального коефіцієнту між станом ринку і лінгвістичною моделлю, яка є домінуючою на даний час з використанням слів-маркерів;
2. **Прогнозування стану ринку:** На основі отриманих шляхом аналізу мережі Інтернет даних утворити коригуючий коефіцієнт для прогнозування поточних тенденцій ринку;
3. **Сегментація типів ринку:** Використання машинного навчання для групування ринків за спільними рисами для прогнозування їх станів окремо та виведення коефіцієнту впливу ринків один на одного;
4. **Аналіз ризиків:** Розробка алгоритмів для оцінки ризиків, пов'язаних із певними інвестиційними стратегіями на основі прогнозованої моделі;
5. **Розробка рекомендаційної системи та індивідуальний підхід до навчання:** На основі наявних даних можна буде автоматизувати створення персоналізованих порад та рекомендацій, які враховують індивідуальні особливості та потреби, що може бути використано для формування відгуків для користувачів симулятора;
6. **Звітність та аналітика:** Автоматизація генерування звітів та аналітичних дошок, графіків, діаграма для користувачів та адміністраторів системи;

1.2 Аналіз аналогів програмного продукту

В силу особливостей кваліфікаційної роботи порівняння аналогів проводиться в двох площинах: Порівняння симуляторів та реалізацій систем прогнозування.

Симуляція – імітація реального процесу чи ситуації. В більшості випадків симулятори використовують, коли взаємодія з реальним об'єктом зв'язана з недосяжністю, небезпекою чи високою вартістю такої взаємодії.[10]

Симулятори поділяють на два види:

- Навчальні або тренувальні симулятори;
- Розважальні симулятори.

В ході дослідження аналогів було виявлено що існує обмежена кількість симуляторів згідно до теми кваліфікаційної роботи, які би покривали різні аспекти процесу і надавали би можливість роботи з реальними даними. З існуючих аналогів наближених до теми в ході дослідження було виділено наступні:

«STONKS-9800» - ігровий симулятор японської фондової біржі 80-90х років розроблений українськими розробниками. Надає достатньо глибоке розуміння процесів інвестування у розважальній формі, але концентрує увагу на ринці Японії в обмеженому часовому просторі.

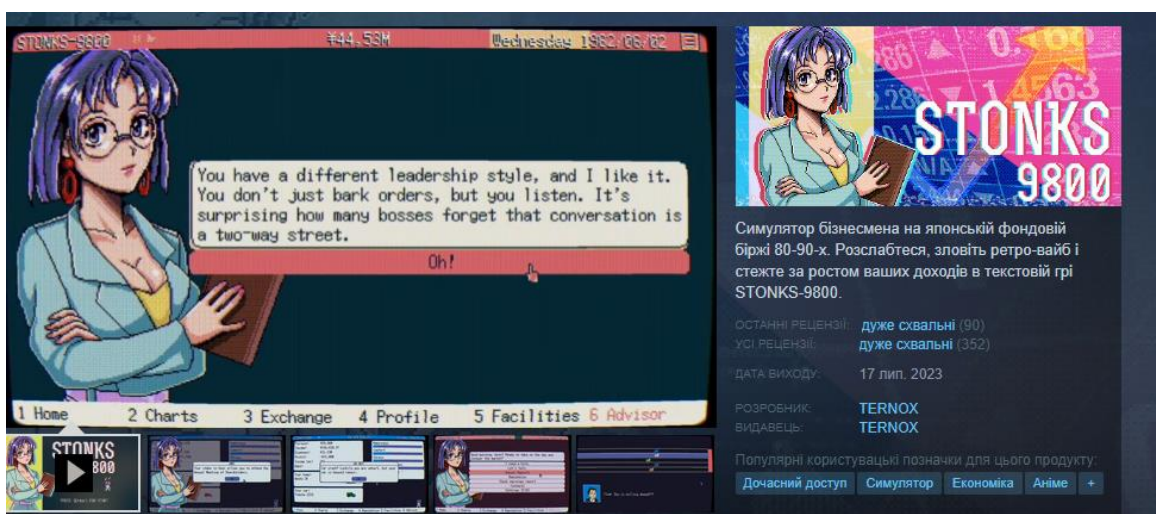


Рис. 1.2.1 «STONKS-9800»

«**BondEvalue**» – платформа для роботи з облігаціями, що використовує реальні гроші. Платформа надає тестовий баланс для навчання, але має обмежений функціонал що суттєво обмежує використання системи для освітніх цілей. Система не використовує методи прогнозування та не має вбудованих інструментів ШІ. Не дає інструментів статистики використовуючи лише базові економічні формули ізольовано від соціального середовища.

«**Build Your Stax**» – Симулятор, що був розроблений некомерційною організацією "Next Gen Personal Finance» та поєднує є гейміфікованою середою, яка дозволяє сформувати базове уявлення про інвестування. Симулятор надає безкоштовну навчальну програму з особистих фінансів. Ціллю симуляції є сформувати інвестиційну стратегію за обмежений період часу. Система має елементи симуляції, але не використовує реальні дані для прогнозування, а самі алгоритми побудовані на спрощеній системі «пропозиція/попит» і не відтворює соціогуманітарну ситуації та її вплив на економічні процеси. В результаті наявних механізмів гра показує, який результат можна отримати слідуючи обраній гравцем інвестиційній стратегії, але цей результат через сильну економічну ізольованість (відсутність суттєвих факторів зовнішнього впливу) дає дуже базове знання про економіку ігноруючи соціальний та поведінковий аспект.



Рис. 1.2.2 «Build Your Stax»

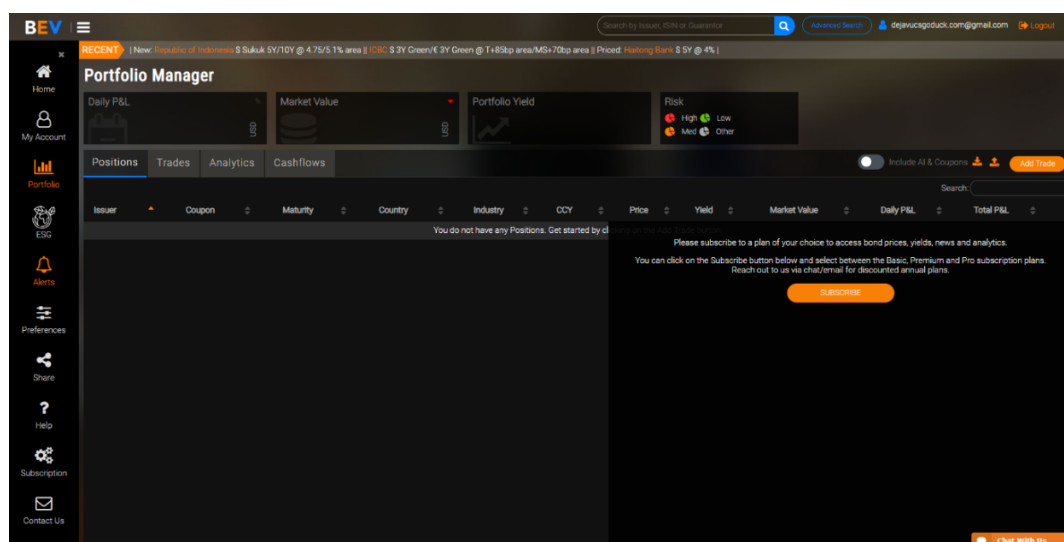


Рис. 1.2.3 Сторінка інвестиційного портфелю користувача в «BondValue»

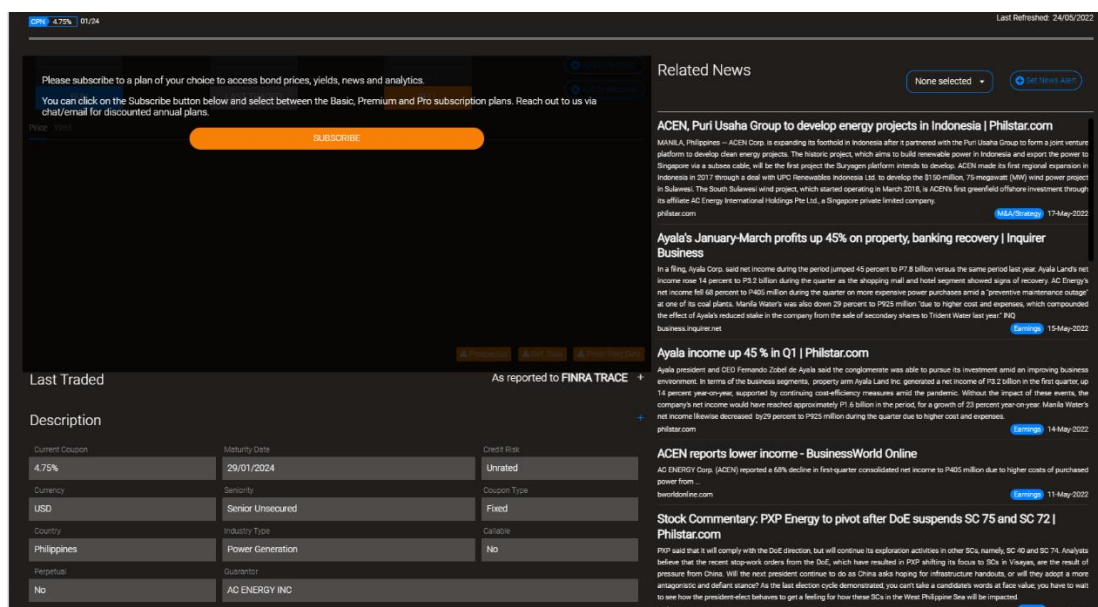


Рис. 1.2.4 Статистика облігації в «BondValue»

«Wall Street Survivor» – є однією з найпопулярніших платформ для віртуальної торгівлі акціями. Система спрямована на навчання людей основам інвестування та фінансової грамотності. Особливістю платформи є використання гейміфікації процесу, що робить процес навчання більш захоплюючим та покращує рівень засвоєння навичок через емоційне залучення. Надає доступ до широкого спектру навчальних матеріалів – починаючи від текстових статей і закінчуючи відеоуроками.

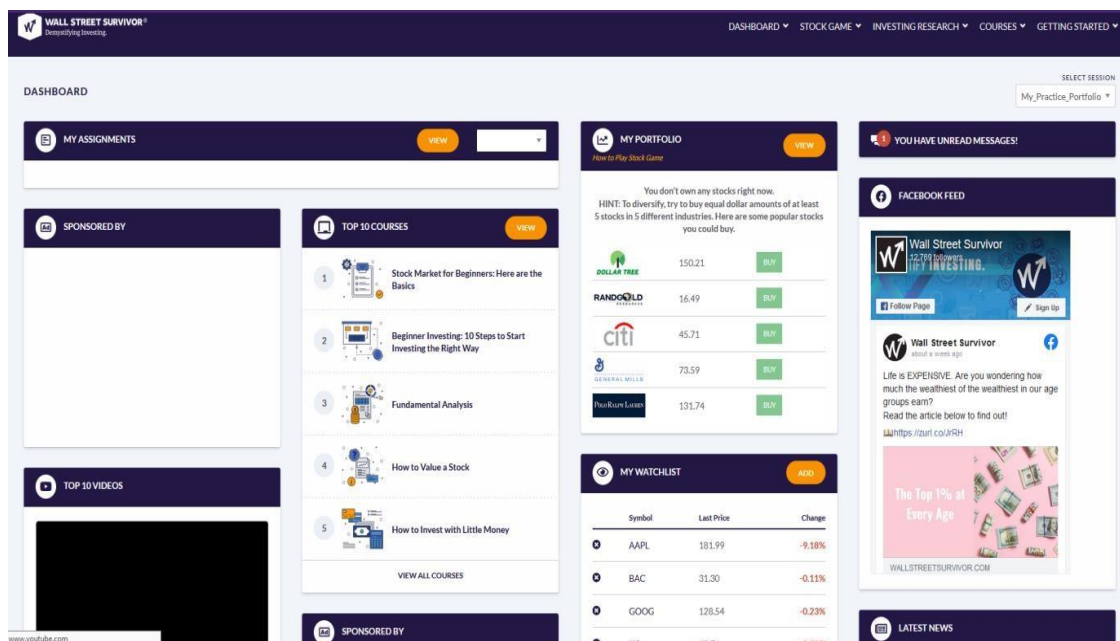


Рис 1.2.5 Головний екран «Wall Street Survivor»

Отже, порівнюючи із темою кваліфікаційної роботи можна з розглянутими аналогами симуляторів можна зробити висновок, що основна увага платформ зосереджена на акційному ринку або на самий базових знаннях про інвестування вилучаючи проміжну ланку прогнозування, при цьому не надаючи систем прогнозування змін та/або не використовуючи реальні дані.

Окремо потрібно оглянути наявні системи прогнозування. На даний момент не було виявлено готової та відкритої системи прогнозування станів фінансових ринків хоча роботи в цьому напрямку активно ведуться [24]. Основними напрямками цих досліджень на даний момент є:

- Прогнозування та превенція утворення бульбашок.
- Визначення шахрайства.
- Впровадження систем-асистентів.
- Прогнозування станів фінансових ринків.

Таким чином наявні симулятори, аби надають недостатньо достовірні дані, або концентруються на власному середовищі для навчання. В більшості випадків в них відсутні системи прогнозування та, системи-асистенти. В цей час моделі прогнозування, які є на даний момент є вузькоспеціалізованими, пропріетарними або знаходяться на ранній стадії розвитку.

1.3 Архітектура та узагальнена структура майбутнього програмного комплексу.

Для розробки підсистем симулятора було обрано клієнт-серверну архітектуру в якості основи з використанням мікросервісів на стороні сервера реалізуючи *Enterprise Service Bus* підхід до архітектури. Сам сервер виконує операції або сам, або передає виклик на виконання паралельно запущеному процесу, який прив'язаний до відповідного порту. В нашому випадку такими системами є система прогнозування та системи математичних розрахунків.

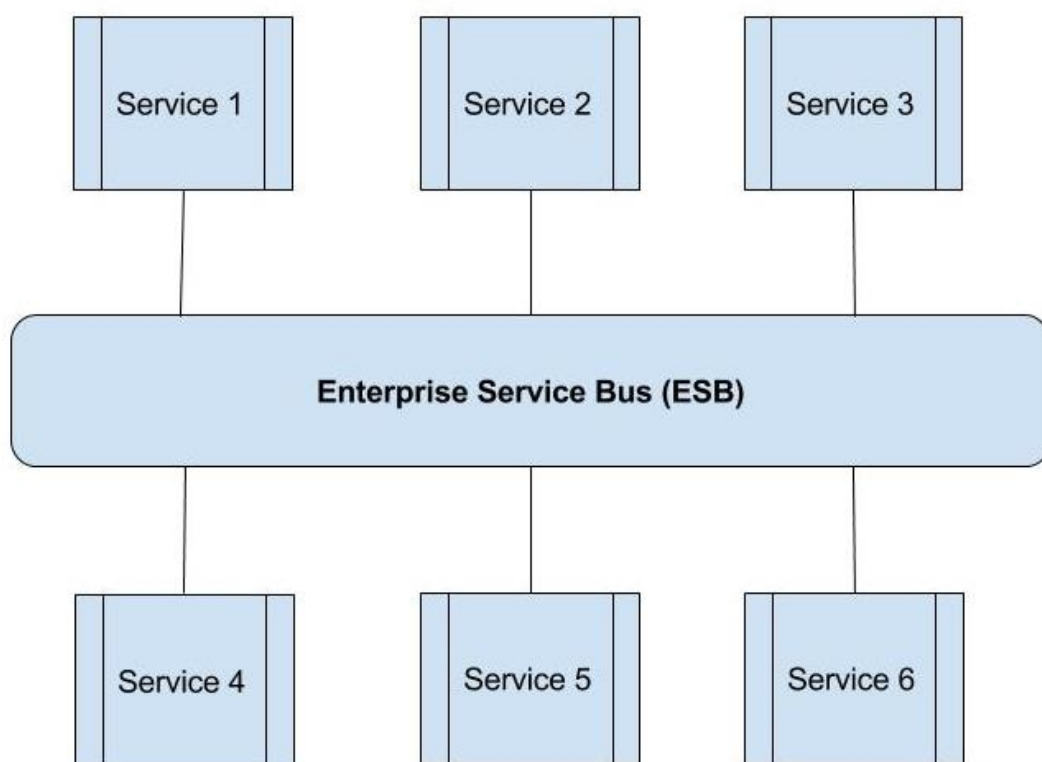


Рис. 1.3.1 Архітектура Enterprise Service Bus

Функціонування системи прогнозування засноване на використанні мови *Python* та бібліотек III. Після проведення аналізу даних виконавча середовище мови *Python* передає необхідні дані на збереження в базу даних і повертає коефіцієнти, які в подальшому можуть бути передані в підсистему розрахунків на основі платформи .NET для розрахунків точних моделей з урахуванням коефіцієнтів.

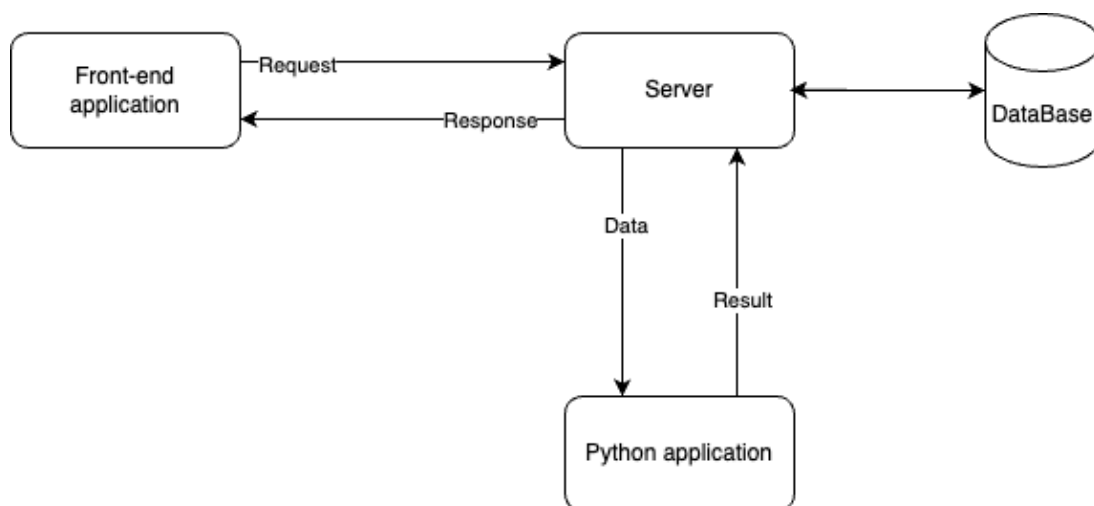


Рис. 1.3.2 Загальна діаграма архітектури додатку

Система математичних розрахунків побудована на базі платформи .NET ASP.CORE та побудована на основі патерну «Команда» (англ. *Command*)

1.4 Інструментальні засоби реалізації програмного продукту.

В якості основи веб частини інформаційної системи було обрано мову *JavaScript*[2] та бібліотеки написані для неї. Так, як розробка системи проводилася командою реалізація системи була поділена на декілька основних напрямків:

- Клієнтська частина – те з чим взаємодіє користувач. Клієнтська частина була реалізована з використанням мови програмування *JavaScript*.
- Серверна частина, що складається з підсистем математичних на базі мови мову *C#* та платформу *.NET* на якій будуть проводитись основні математичні розрахунки. розрахунків та підсистеми прогнозування на базі мови *Python* для створення моделей аналізу лінгвістичної моделі мережі Інтернет.

Розділивши напрямки для кожного з них було обрано необхідний набір інструментів. Для клієнтської частини було обрано наступні інструменти:

- Компоненти користувацького інтерфейсу будуть реалізовані

механізмами бібліотеки **React.js**[6]

- Для функціонування серверної частини було обрано платформу **Node.js** в основі якої лежить модель подій (*англ. Event*) з неблокуючими операціями введення-виводу. Платформа використовує модель подій на основі «циклу подій» (*англ. Event Loop*).
- Для координації потоків даних буде використано відкритий фреймворк **Express.js**[3].
- Для роботи з базами даних було вирішено використати систему керування базами даних **MongoDB**[4].

На стороні серверу будуть використані наступні інструменти:

- В якості основи для машинного навчання будуть використані бібліотеки **TensorFlow**, **Keras**, **Pandas** та **Scikit-learn**.
- Для реалізації контролерів математичної підсистеми буде використано бібліотеку **Mediatr** – Бібліотека, що активно використовується для сучасних веб-систем написаних на основі мови C#, що дозволяє зменшити зв'язність різних компонентів системи шляхом впровадження систем команд, які пишуться ізольовано, як одиниця роботи на відміну від *XxxService* підходить коли різні команди прописуються, як методи деякого сервісу.
- В якості основи майбутньої системи розрахунку буде задіяна платформа **.NET** – Це фреймворк створений для мови C#, яка надає широкий спектр базових команд та підсистем для полегшення процесу написання програмних систем.

1.5 Інструменти проектування потоків даних за стандартами DFD та IDEF0

Для вирішення завдань прогнозування змін фінансових ринків на основі соціальних явищ потрібно оперувати не лише множиною інформаційних даних, а й мати можливість прослідкувати історію їх взаємодії у часі щоб мати змогу

провести лінію впливу змінних одна на одну.

Для вирішення цієї задачі було вирішено скористатися діаграми потоків даних за стандартом DFD, «що застосовуються для описування документообігу і обробки інформації як додаток до IDEF0. На відміну від IDEF0, де система розглядається як пов'язані між собою роботи і стрілки представляють собою жорсткі взаємозв'язки, стрілки в DFD показують лише те, як об'єкти (включаючи дані) рухаються від однієї роботи до іншої. DFD окреслює функціональні залежності значень, які вираховуються в системі, включаючи вхідні, вихідні значення і внутрішні сховища даних. DFD - це графік, який відображає рух значень даних від їх джерел»[11].

Діаграми потоків даних забезпечують графічне відображення системи. Дана модель дозволяє розробникам програмного забезпечення, клієнтам та користувачам ефективно співпрацювати під час аналізу технічних вимог, що висуваються до системи.

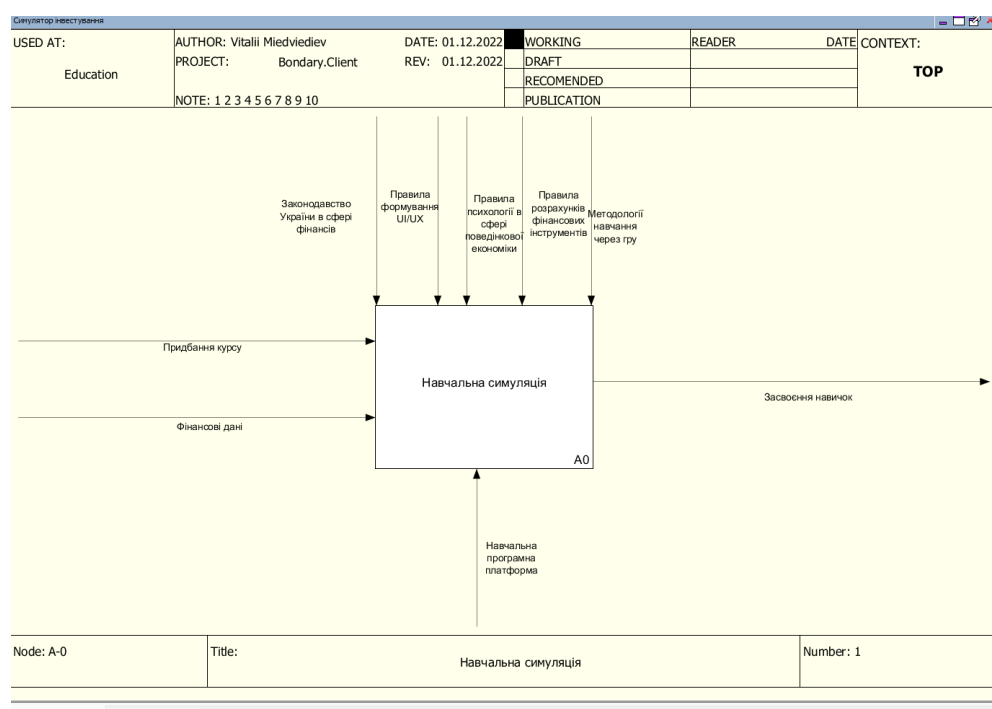


Рис. 1.4.1 Діаграма потоків симулятора

Переваги використання діаграм потоків даних полягає в наступних властивостях:

- Може зобразити схему організації чи системи.

- Може скласти схему поточної або запропонованої ситуації.
- Може полегшити аналіз або проектування.
- Забезпечує хорошу роботу від аналізу до проектування.
- Полегшує спілкування з користувачем на всіх етапах.

В моделюванні виділяють наступні види DFD діаграм:

Поточні – як зараз надходять дані:

- Запропоновані – як ми хочемо, щоб дані надходили.
- Логічні - «сутність» процесу.
- Фізичні - здійснення того чи іншого процесу.
- Розділені фізичні - архітектура системи або дизайн високого рівня.

Потоки даних є механізмами, що використовуються для моделювання передачі інформації (або фізичних компонентів) з однієї частини системи в іншу. При-значення процесу полягає в продукуванні вихідних потоків із вхідних відповідно до дії, що задається ім'ям процесу. Кожен процес повинен мати унікальний номер для посилань на нього всередині діаграми.

Ефективність та продуктивність праці розробників функціональних моделей можуть бути підвищені за рахунок застосування типових моделей та окремих діаграм, орієнтованих на застосування у конкретних предметних галузях

Дані з різних джерел містяться в сховищі, а їх описи – в репозиторії метаданих. Кінцевий користувач, використовуючи різні інструменти (засоби візуалізації, побудови звітів, статистичної обробки, тощо) та вміст репозиторію, аналізує дані у сховищі. Результатом є інформація у вигляді готових звітів, знайдених прихованих закономірностей, будь-яких прогнозів. Оскільки засоби роботи кінцевого користувача зі сховищем даних можуть бути найрізноманітнішими, теоретично їх вибір не повинен впливати на структуру сховища та функції його підтримки в актуальному стані.

1.6 Аналіз методів машинного навчання

Машинне навчання (*англ. Machine Learning, ML*) – це загальний термін стосовно розв'язування задач, для яких розробка алгоритмів програмістами-людьми була би занадто витратною, і які натомість розв'язують, допомагаючи машинам «відкривати» свої «власні» алгоритми, без потреби явно вказувати їм, що робити, за допомогою алгоритмів, розроблених людьми. Будучи методом аналізу великих даних (*англ. BigData*) ML використовується в дедалі більшій кількості галузей, включно з класифікацією, кластеризацією, аналітичному прогнозуванні, асоціативному навчанні, виявленні аномалій, аналізі зображень і обробленні природної мови [34]. Особливістю ML в порівнянні з традиційними алгоритмами є здатність до автоматичного знаходження функцій, що зменшує вартість розробки нових програмних продуктів та допомагає будувати системи здатні адаптуватися до оточення.

Далі буде наведено опис алгоритмів, що буде використано в кваліфікаційній роботі з коротким описом:

Naïve Bayes (наївний баєсів класифікатор) [37]– це статистичний метод класифікації, який ґрунтується на теоремі Баєса. Цей алгоритм вважається "наївним", оскільки він припускає, що всі властивості класифікованих об'єктів незалежні між собою.

Теорема Байєса дозволяє розрахувати апостеріорну ймовірність $P(y|x)$ на основі $P(y)$, $P(x)$ і $P(x|y)$:

Формула 1.6.1

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)},$$

де $P(y|x)$ – posterior probability – апостеріорна ймовірність, що дане значення ознаки x говорить про належність до класу y , саме її нам треба розрахувати;

$P(x|y)$ – likelihood – ймовірність зустріти ознаку x серед всіх ознак що належать класу y (правдоподібність, тобто ймовірність даного значення ознаки при даному класі);

$P(y)$ – class prior probability – безумовна (априорна) ймовірність зустріти документ класу y в корпусі документів;

$P(x)$ – predictor prior probability – безумовна (априорна) ймовірність даного значення ознаки.

Основні кроки алгоритму:

1. *Побудова моделі*: Обчислення ймовірностей залежності до кожного класу на основі тренувальних даних.
2. *Прогнозування*: Обчислення ймовірності належності нових вхідних даних до кожного класу і вибір класу з найвищою ймовірністю.

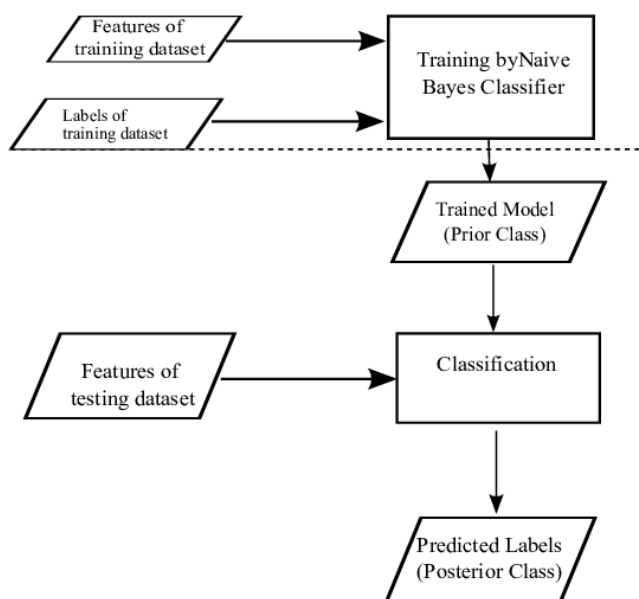


Рис. 1.6.1. Блок-схема навчання та ідентифікації за допомогою Байєсового наївного класифікатора

Наївний Баєсів класифікатор широко використовується в задачах текстової класифікації, фільтрації спаму, а також в інших областях машинного навчання.

Таким чином даний алгоритм можливо використати для формування вірогідностей та встановлення вірогідного типу ринку для поділення його за типом.

Наступним є алгоритм **RandomForestClassifier** [36], Random Forest є ансамблевим методом машинного навчання, який використовується для класифікації та регресії. Алгоритм *RandomForestClassifier* входить до складу бібліотеки *scikit-learn*, що буде використано в межах кваліфікаційної роботи і використовується для класифікації даних за допомогою великої кількості різних рішень, які об'єднуються в "лісі" (forest).

Основна ідея Random Forest полягає в тому, щоб побудувати кілька рішень (дерев рішень) під час тренування та об'єднати їх, щоб отримати більш точний та стійкий класифікатор. Кожне дерево рішень навчається на випадковому підмножині даних і випадковому підмножині ознак.

Основні етапи алгоритму RandomForestClassifier:

1. *Випадковий вибір підмножини даних*: З набору даних випадковим чином вибирається підмножина (з поверненням), яка використовується для тренування кожного дерева рішень.
2. *Випадковий вибір ознак*: При побудові кожного дерева рішень випадковим чином вибирається підмножина ознак для використання у кожному вузлі дерева.
3. *Побудова дерева рішень*: Кожне дерево рішень тренується на вибраній підмножині даних і вибраній підмножині ознак за допомогою рекурсивного поділу вузлів відповідно до значень ознак.
4. *Голосування або середнє значення*: Для класифікації нового об'єкта кожне дерево рішень вносить свій внесок, і результат обчислюється на основі голосування (у випадку класифікації) або середнього значення (у випадку регресії) виходів дерев.

Перевагами алгоритму є висока стійкість до перенавчання і здатність працювати з різноманітними типами даних. Цей метод також може надавати інформацію про важливість ознак, що робить його корисним інструментом для вибору ознак при класифікації.

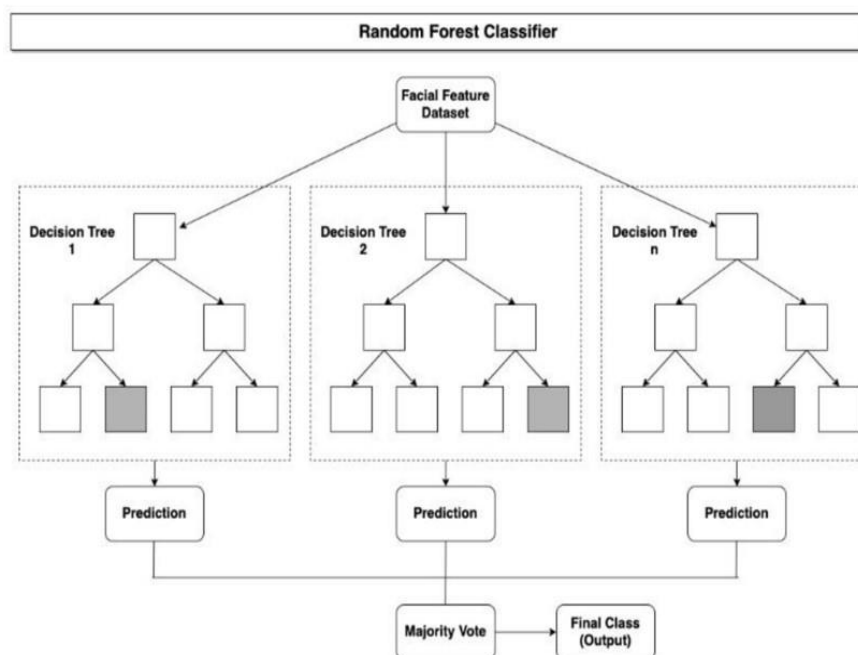


Рис. 1.6.2 Блок-схема роботи алгоритму

Рекурентні нейронні мережі (англ. *RNN*) є типом нейронних мереж, які призначені для роботи з послідовнісними даними та враховують часові залежності між елементами послідовності. Основною ідеєю RNN є використання зв'язків між вузлами для передачі інформації від попередніх кроків часу до поточного.

Таким чином RNN можуть використовуватися для задач розпізнавання текстів написаних природніми мовами та формування підсистем аналізу семантики.

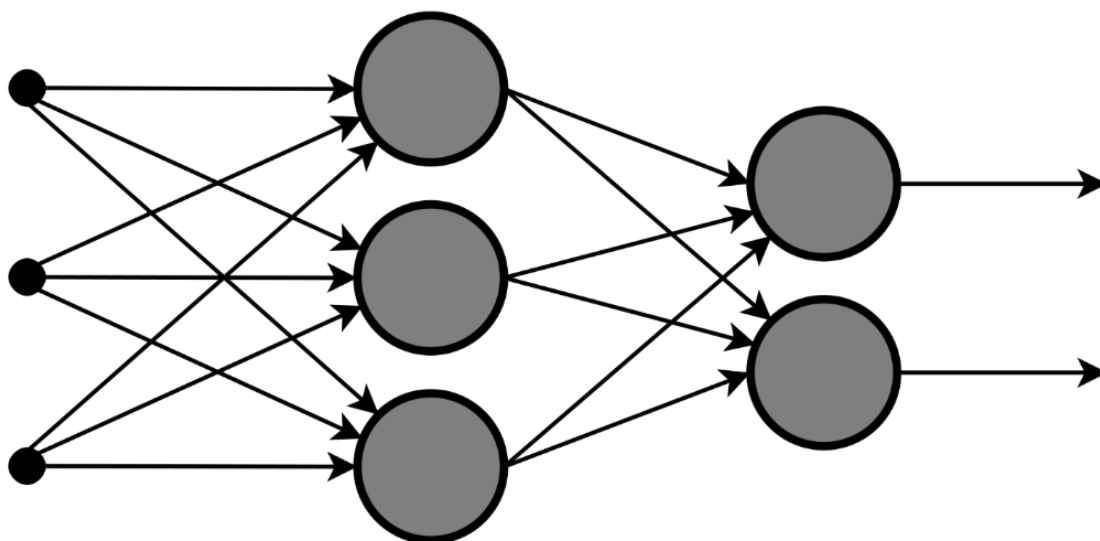


Рис. 1.6.3 Загальне зображення рекурентної нейромережі

Висновки до першого розділу

Під час написання основної частини було проведено аналіз предметної області, що дозволило визначити основні аспекти системи прогнозування стану фінансових ринків.

Провели аналіз існуючих аналогів та систем, що знаходяться у розробці. Також було вирішено, що система буде використовувати клієнт-серверну архітектуру, для роботи з моделями ШІ – мову Python, для роботи з числовими обчисленнями – мову C#.

Було обрано моделі штучного інтелекту, машинного навчання та розроблено алгоритми, які будуть використовуватись в системі та проведено їх детальний опис такі, як Naive Bayes та формування припущень на основі отриманих даних з метою розбиття даних на категорії та моделі RandomForestClassifier і RNN для аналізу природніх мов, та прогнозування станів фінансових ринків через потужність, та можливість глибокого аналізу тексту написаного природньою мовою.

В результаті було обрано базову архітекту системи та набір інструментів для реалізації системи. Для розробки моделей та алгоритмів штучного інтелекту буде використано – Python з бібліотеками TensorFlow, Keras, Scikit-learn, Pandas.

Для реалізації системи обчислення було використано мову програмування C# та платформу .NET. Окремо було проведено вибір способів графічного опису потоків-даних в системі.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ КОРИСТУВАЧІВ. ІМПЛЕМЕНТАЦІЯ ПРОГНОЗУВАННЯ

2.1 Формування варіантів використання системи.

Розроблена система має об'єднувати декілька основних компонентів, а саме підсистеми клієнтської частини симулятора, які будуть забезпечувати людино-машину взаємодію та серверної частини, які будуть забезпечувати математичні розрахунки, підсистему асистента та прогнозування. Таким чином ми можемо сформулювати вимоги до всіх системи симулятора, як для систем графічного інтерфейсу та варіантів його використання, так і до підсистем прогнозування:

Вимоги не автентифікованих користувачів:

1. Реєстрація;
2. Автентифікація;
3. Перегляд списку облігацій;
4. Перегляд інформації про облігацію з можливістю отримати прогноз та аналіз тенденцій по ній.

Вимоги автентифікованих користувачів:

1. Придбання облігацій з оцінюванням;
2. Продаж облігацій з оцінюванням;
3. Перегляд кабінету користувача;
4. Перегляд власного інвестиційного портфеля з можливістю отримати консультацію по ньому;
5. Поповнення балансу;
6. Перегляд власного рейтингу та співвідношення його дій з рішеннями, які би прийняв комп'ютер та інші інвестори, що зареєстровані в системі.
7. Виклик інтелектуального-асистента для підказки і прогнозу різних операцій.

Функціональні вимоги:

1. Симулятор повинен мати можливість реєстрації користувача з присвоєнням йому відповідної ролі (користувач, адміністратор) та можливості автентифікації для розширення можливостей у роботі з симулятором;
2. Відображення актуального списку цінних паперів (в поточній версії облігацій ОВДП України);
3. Можливість збереження інформації: Система повинна зберігати інформацію про клієнта, портфель цінних паперів, історію портфеля, рейтинг, історію змін рейтингу, новини;
4. Доступ до Інтернету: користувач не зможе користуватись симулятором без доступу до Інтернету;
5. Проводити активний збір інформації про вікову, професійну тощо структури людей, що проходять курс для формування навчальних програм;
6. Інтеграція підсистеми на рівні серверу, яка буде забезпечувати надання аналітичної інформації про можливі зміни в економіці;
7. Інтеграція інтелектуальної підсистеми асистента на рівні серверу, яка буде виконувати функції консультанта та давати оцінку дій;
8. Інтеграція підсистеми математичних розрахунків на рівні серверу, яка буде виконувати розрахунки математичних моделей та оцінювати успішність операції користувачем;
9. Можливість створити запит на формування тенденції ринку для прогнозування змін станів на обраному відрізці часу.

Нефункціональні вимоги:

1. Сприйняття:
 - Час, потрібний для навчання роботи з веб-додаток для звичайних користувачів — 30 хвилин, для досвідчених — 5 хвилин.
 - Інтерфейс симулятора має бути зрозумілим та зручним для користувача, не вимагати від нього додаткової підготовки. При проектуванні інтерфейсу потрібно уникати великої кількості компонентів на одному екрані, щоб збільшити час до настання

когнітивної втоми користувача;

2. Продуктивність. Симулятор повинен підтримувати, як найменше 100 одночасно працюючих користувачів;
3. Масштабування – система повинна мати можливість збільшувати продуктивність, зі збільшенням користувачів таким чином, щоб це ніяк негативно не відобразилося на роботі;
4. Система прогнозування має задіяти в аналізі лінгвістичної моделі мережі Інтернет, соціальні медіа вертикального (ті які працюють за ієрархічною моделлю, наприклад онлайн видавництва, державні медіа) та горизонтального (мережі, що реалізують підхід «рівний-до-рівного», наприклад соціальні мережі твітер (*англ. Twitter*) типів. Ураховувати динаміку не фінансових ринків, та формувати загальний коефіцієнт на основі їх співвідношення до реальної ситуації.
5. Математична підсистема має надавати широкий спектр контролю за типами даних, для уникнення некоректних математичних операцій та накопичення похибки.

Антивимоги:

- Не використовує інформацію, заборонену нормами моралі та законодавством;
- Не містить реклами, принаймні тієї, що не стосується тематики сайту;
- Не використовує інформацію, що є недійсною або може зашкодити психіці користувачів.

Аналіз функціональних вимог дав можливість виділити наступні сутності, що забезпечать реалізацію програмного комплексу системи.

Далі буде наведено специфікацію прецедентів: купівлю та продаж облігацій, виклик детальної інформації про цінний папір, отримання підказки асистентом.

2.2 Проектування варіантів використання системи.

Після формування вимог до системи потрібно детальніше описати необхідні варіанти використання системи, які доступні користувачеві. Загальна схема варіантів використання наведена нижче.

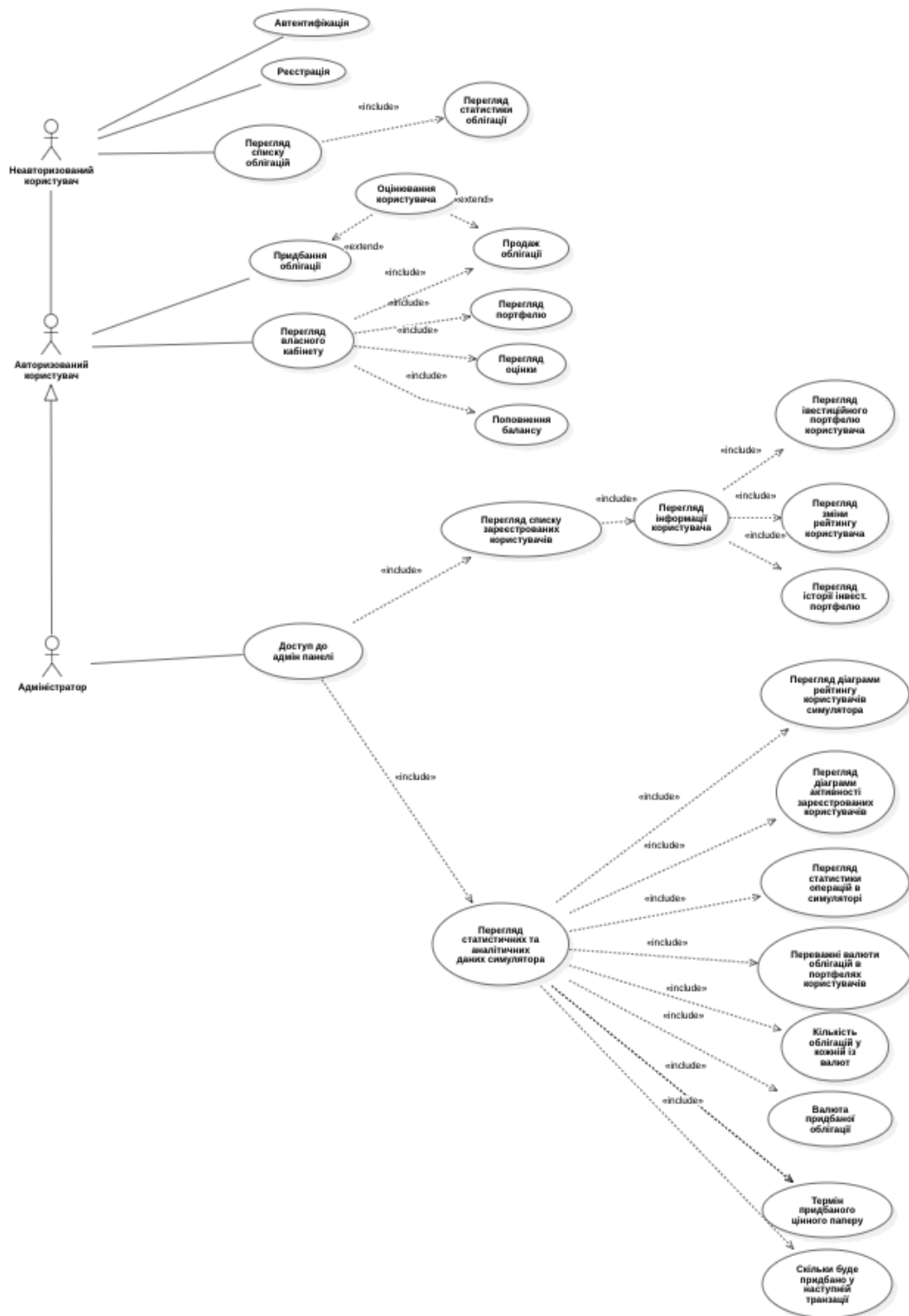


Рис. 2.2.1 Діаграма варіантів використання симулятора

На даному етапі кваліфікаційної роботи відносно до теми представлено декілька основних типів варіантів використання, які потребують більш детального уточнення їх прецедентів, а саме:

- Придбання/продаж цінних паперів (в поточній реалізації облігацій) з використанням підсистеми оцінки вибору користувача на основі поточної ситуації;
- Виклик інтелектуального агента, що використовує для формування підказки дані з системи прогнозування та математичної системи. Може працювати в двох можливих варіаціях:
 - Користувач отримує деталізовано інформацію, щодо конкретного цінного паперу, з оцінкою різних операцій над ним;
 - Користувач отримує деталізовану інформацію, щодо поточної тенденції на ринку з описом показників, які були використані для встановлення поточної тенденції та з пропозицією використання інвестиційних стратегій, що краще відповідають поточному положення на ринку.
- Запит до математичної підсистеми, яка відповідає за базові математичні обрахунки, а також використовує коефіцієнти відхилення розраховані системою прогнозування для формування моделей, щоб відповідали різним тенденціям. В межах цієї підсистеми виконується формування оцінки операції на основі значення типу операції, а також поточної ситуації на ринку;
- Запит на отримання інформації щодо тенденції ринку: Формування тенденції виступає, як окрема операція, що є надбудовою над системою прогнозування та математичною системою і формує інформацію, щодо тенденції за різними сценаріями з можливістю визначити діапазон дат для аналізу уперед чи у минуле, для визначення поточної тенденції ринку, а також його відхилення від «позитивного», «негативного» та «нейтрального» сценаріїв.

Таблиця 2.2.1

Опис варіанту використання: Виклик інтелектуального асистента

Ім'я варіанту використання: Виклик асистенту		Рівень важливості: Високий
Основний актор: Користувач		
Короткий опис:		Користувач викликає асистента для отримання порад щодо конкретного цінного паперу
Передумови:		Вхід у систему.
Пост-умови:		Отримання інформації з оцінюванням різних операцій і їх наслідків
Відносини:		
	Асоціація:	
	Включення:	
	Розширення:	
	Узагальнення:	
Основний потік:		Користувач обирає цінний папір та робить запит на отримання підказки, щодо нього.
Альтернативні/виключні потоки:		

2.3 Програмна реалізація компонентів підсистем

В межах даної кваліфікаційної роботи було створено декілька програмних алгоритмів для вирішення питань збору та первинного аналізу даних, що необхідні для формування кінцевого алгоритму.

Таким чином на першому етапі уточнення та підрахунків кількості вжитих слів відповідно до словника потрібно створити підпрограму, яка здатна проходити по деякому джерелу інформації розташованому у мережі Інтернет, та виконувати аналіз та підрахунок найбільш розповсюджених та вживаних слів в тексті з використанням словника слів-маркерів, що зустрічаються у тексті.

```
import requests
from bs4 import BeautifulSoup
import re
import random
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
from collections import Counter
import nltk
import pandas as pd
import pymorphy2

url = 'https://www.pravda.com.ua/news/2023/12/10/7432405/'
response = requests.get(url)

soup = BeautifulSoup(response.content, 'html.parser')
news = soup.body.get_text()
# remove tags
news = re.sub(r'<.*?>', '', news)

# to token
news = re.sub(r'""[0-9]""', r" ", news)
news = re.sub(r'""[^A-ЩЬЮЯҐЄІїа-щьюяґєіі'\n]""', ' ', news)
rows = news.lower().split("\n")
rows = [row for row in rows if row]
rows_words = [[word for word in row.split(" ") if word] for row in rows]
words = [word for words in rows_words for word in words if word]

word_counts = Counter(words)
most_common_words = word_counts.most_common(20)

for word, count in most_common_words:
    print(f"{word}: {count}")

stop_words = ""

stop_words = set(stop_words.split(", "))
morph = pymorphy2.MorphAnalyzer(lang="uk")
series = pd.Series(words)
series = [s for s in series if s not in stop_words]
```

```

series = [morph.parse(s)[0].normal_form for s in series]
print('Кількість символів: ', len(series))
series = nltk.ngrams(series, 1)
series = nltk.FreqDist(series)
print('Кількість токенів: ', series.N())
print('Кількість унікальних токенів: ', series.B())
print('Найбільш уживані токени: ', series.most_common(20))
series.plot(20, cumulative=True)

```

Даний фрагмент коду виконує одну з базових операцій алгоритму прогнозування, а саме підрахунок слів маркерів згідно зі словника.

```

C:\Users\ViMi\PycharmProjects\pythonProject\.venv\Scripts\python.
v: 16
з: 13
на: 13
що: 9
уп: 8
у: 8
та: 8
україні: 7
правда: 6
від: 6
вже: 5
новини: 4
українська: 4
зеленський: 4
як: 4
кузін: 4
фото: 4
за: 4
реклама: 4
і: 4

```

Рис 2.3.1 Приклад виконання підрахунку слів-маркерів з використанням розширеного словника

Окремим етапом було винесено порівняння кількості входжень слів маркерів у різних джерелах інформації.

```

baseUrl = 'https://www.pravda.com.ua/news/';
test = seriestget(baseUrl, 2022, 2022)
test2 = seriestget(baseUrl, 2023, 2023)
test.most_common(20)
test2.most_common(20)

```

```

common = [word for word in test.most_common(20) if word in
test2.most_common(20)]

coeff = 0
for word in test.most_common(20):
    for word2 in test2.most_common(20):
        if word2[0] == word[0]:
            coeff += word2[1] / word[1]

```

Другою основною підсистемою, яка необхідна для функціонування систем прогнозування – математична підсистема, яка має вираховувати базову статистику з подальшим використанням коефіцієнтів створених системою прогнозування. Частина коду розрахунку статистики наведена нижче.

```

...

public double HY

{

    get

    {

        return ( (

                                ((DirtyMarketPriceFixedCoupon
                                (ClearMarketPriceFixedCoupon / 100)) / (OwnDays + YearCouponIncome))
                                /

                                (Bond.Nominal / 2)

                                ) * 100 ) * Bond.ShiftCoeff;

        }

    }

...

```

Даний фрагмент коду виконує операції обрахунку базової статистики згідно з заздалегідь відомими формулами на основі моделі предметної області. (В поточній реалізації модель облігації (*англ. Bond*) [Додаток А].

Для взаємодії з сервером NodeJS використовується підхід команд, де на вхід команді підсистемі передається набір даних отриманих з зовнішнього

джерела модель, що описує необхідний мінімум інформації який отримує система на вхід наведена нижче.

```
public class BondCommandBase
{
    public DateTime EmitDate { set; get; }
    public DateTime RepayDate { set; get; }
    public DateTime BuyDate { set; get; } = DateTime.Now;
    public string? BondType { set; get; }
    public string? Code { set; get; }
    public string? Currency { set; get; }
    public int Nominal { set; get; }

    public double YearCoupon { set; get; }
    public double Y { set; get; }
    public double YTM { set; get; }
    public string? TermType { set; get; }
    public double BuyPrice { set; get; }
    public double SellPrice { set; get; }
    public DateTime NearestPayDate { set; get; }
    public DateTime[]? PaymentSchedule { set; get; }
}
```

Складовою частиною математичною підсистеми виступає підсистема оцінювання, яка проводить багато факторну оцінку операцій користувача.

```
public async Task < EstimationRates > GetRate(Domain.Entities.Bondbond)
{
    var price = bond.Statistic.DirtyMarketPriceFixedCoupon;
    var yeild = bond.Statistic.Y;

    if (yeild >= 10)
        return EstimationRates.High;
    if (yeild < 10 & & yeild >= 5) return EstimationRates.Good;
    if (yeild <= 5 & & yeild >= 1) return EstimationRates.Average;
    if (yeild < 1 & & yeild >= 0) return EstimationRates.Low;
    if (yeild < 0) return EstimationRates.Awfull;

    return EstimationRates.Average;
}
```

Для проміжної оцінки всередині системи використовується перерахування *EstimationRates*, яка пізніше транслюється в оцінку користувача у вигляді числа від 0.5 до 1.5 наступним кодом

```

public enum EstimationRates
{
    High      = 2,
    Good      = 1,
    Average   = 0,
    Low       = -1,
    Awfull    = -2
}

public double TranslateRate(EstimationRates rates)
{
    switch(rates)
    {
        case EstimationRates.High: return 1.5;
        case EstimationRates.Good:  return 1;
        case EstimationRates.Average: return 1;
        case EstimationRates.Low:   return 0.5;
        case EstimationRates.Awfull: return 0.5;
        default: return 1;
    }
}

public double applyRateCoeff(double rawRate, double coeff)
{
    return rawRate * coeff;
}

```

Математична підсистема працює на основі обміну повідомлень між різними компонентами системи, і реалізована у вигляді окремо функціонуючого веб сервісу, з яким можливе спілкування шляхом виклику відповідного шляху (route), що дозволяє компонентам системи обмінюватися інформацією один з одним та зменшує фактичну зв'язність програмної системи.

Наразі математична підсистема реалізує базові компоненти для розрахунку базової статистики, підготовку цінових екстремумів на основі базової статистики та надає базовий функціонал для вираховування статистичної інформації з можливістю накладання коефіцієнтів відхилення [Додаток А].

Система асистента реалізована з використанням допоміжних коефіцієнтів розрахованих системою прогнозування, які разом з прапором сценарію та прапором дії, яку виконав користувач проводить внутрішню оцінку рішення, яке було обрано. Наприклад при операції купівлі в негативному сценарії, на зростанні повертається позитивна чи нейтральна оцінка так, як на падінні ринку цінні папери з більшою вірогідністю втрачають в ціні, на відміну від операцій коли продажу, коли користувач з більшою вірогідністю втрачає різницю між поточною ціною, та ціною на момент купівлі цінного паперу.

Фрагмент коду, що ілюструє формування базової оцінки на основі правил наведено нижче.

```
NEGATIVE_ACTION_RATE = 0.5
NORMAL_ACTION_RATE = 1
POSITIVE_ACTION_RATE = 1.5

def GetActionRateScenarion(scenario, operation, coef):
    if scenario == NEGATIVE_SCENARIO:
        if coef == NORMAL or coef == POSITIVE and operation == SELL:
            return NEGATIVE_ACTION_RATE
        if coef == NEGATIVE and operation == SELL:
            return POSITIVE_ACTION_RATE
        if coef == NORMAL or coef == POSITIVE and operation == BUY:
            return POSITIVE_ACTION_RATE
        if coef == NEGATIVE and operation == BUY:
            return NORMAL_ACTION_RATE

    if scenario == POSITIVE_SCENARIO:
        if coef == NORMAL or coef == POSITIVE and operation == SELL:
            return POSITIVE_ACTION_RATE
        if coef == NEGATIVE and operation == SELL:
            return NORMAL_ACTION_RATE
        if coef == NORMAL or coef == POSITIVE and operation == BUY:
            return NEGATIVE_ACTION_RATE
        if coef == NEGATIVE and operation == BUY:
            return NEGATIVE_ACTION_RATE

    if scenario == NORMAL_SCENARIO:
        return NORMAL_ACTION_RATE
```

Висновки до другого розділу

В межах другого розділу провели формування функціональний та не функціональних вимог до симулятивної частини інформаційної системи, які були використані при формуванні інтерфейсу користувача для забезпечення людинно-машиної взаємодії та виведенням інформації прогнозування. Були сформовані варіанти використання системи на даний момент часу, сформовані детальні описи циркуляції інформаційних потоків в межах системи, що дозволило нам визначити, як дані обробляються та передаються в системі, що є критичним для ефективності та надійності системи.

Описані базові вимоги до системи прогнозування, системи математичних розрахунків, та інтелектуального-асистента. Визначили напрямки для розвитку цих компонентів врахувавши потреби у точності, швидкості та надійності при визначенні вимог до математичних моделей та алгоритмів прогнозування.

У процесі формування вимог ми також врахували різноманітні сценарії використання системи. Це дозволило визначити, як користувачі будуть взаємодіяти з системою на даний момент часу та як система може ефективно використовуватися в реальних умовах.

Провели проектування основних прецедентів, згідно до теми кваліфікаційної роботи, а саме. Продаж/купівлю облігації, які підпадають під оцінку математичною системою. Виклик системи асистента для підказки та отримання довідки щодо поточного стану ринків. Виклик математичною системою для отримання розрахованої моделі та виклик системи прогнозування тенденцій для отримання інформації, яка може бути використана для графічного зображення поточних трендів ринку та того, наскільки вони не співпадають з «позитивними» та «негативними» сценаріями.

Провели програмну реалізацію базових алгоритмів системи прогнозування та математичної системи.

РОЗДІЛ 3. ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ МОДЕЛІ. ФОРМУВАННЯ АНАЛІТИЧНИХ ЗВІТІВ ТА АНАЛІЗ ВІДХИЛЕНЬ.

3.1 Аналіз відхилень даних

В ході первинного збору інформації було за допомогою алгоритмів аналізу текстових масивів мережі інтернет було виявлено наявність значних відхилень при прямому перенесені алгоритму коефіцієнтів, які унеможливають формування моделі високої точності без залучення додаткових джерел інформації для уточнення коефіцієнту корегування.

На першому етапі основним джерелом для аналізу лінгвістичної моделі виступали джерела інформації вертикального типу такі як новинні сайти (наприклад видання «Українська Правда») однак аналіз їх моделі не завжди дає вичерпний результат і не може бути використаний в прямому вигляді, що зумовило необхідність залучення для коригування коефіцієнту, як додаткових джерел вертикального типу, так і джерел горизонтального типу.

```
1. 13
що: 14
та: 12
російські: 12
до: 11
не: 11
для: 10
війська: 9
уп: 8
як: 8
ініціативу: 8
про: 7
за: 7
операцій: 7
правда: 6
перехопити: 6
Кількість символів: 2
Кількість токенів: 2
Кількість унікальних токенів: 1
Найбільш уживані токени: [(['окупанти',), 2]]
```

Рис 3.2.1 Результат виконання підрахунку слів найбільш уживані токени з звуженого словника «окупант»

Прикладом впливу джерел горизонтального типу на ринки може виступати наприклад пости Ілона Маска в твітері [46] в момент часу можуть створити екстремум на графіці цін криптовалют чи акції компаній. Переносячи це визначення ширше ми приходимо до розуміння необхідності залучення в аналітичну модель інформації що розповсюджується в соціальних медіа горизонтального типу таких, як соціальні мережі для корегування моделі.

Разом з тим перенесення початкової моделі пошуку по словам маркерам стає ще менш надійним так, як в мережах горизонтального типу ширше розповсюджена проблема фейків та ботів, які можуть створювати інформаційний шум і провокувати хибно-позитивне спрацювання маркерів через приймання системою аналізу віртуальної кризи за реальну. Разом з тим, так як частина представників мережі горизонтального типу є людьми, тобто акторами реальних ринків віртуальна криза може провокувати сплески на реальних ринках і навпаки, що призводить до необхідності формування додаткових підсистем для аналізу джерел горизонтального типу:

1. Бази даних профілів так званих лідерів-думок (людей, що мають широкий вплив на велику кількість людей в окремій сфері) для більш детального аналізу їх дописів. Така підсистема необхідна через високу залежність реальних котирувань від слів та заяв публічний персон [46];

2. Розвиток системи, яка би вносила дані з профілів, які визначені, як боти з меншим коефіцієнтом корегування. Така система аналізу необхідна через тенденцію систем вертикального типу (наприклад урядові організації) маскуватися під системи горизонтального (маскуватися під групу псевдонезалежних віртуальних особистостей), для розповсюдження необхідних їм наративів, що може вести до викривлення реальної моделі, але так, як ці дописи можуть мати вплив на свідомість реальних людей повністю ігнорувати їх вплив неможливо. Тим не менше в такій підсистемі зберігається фактор невизначеності так як не завжди можливо визначити за зовнішніми чинниками чи є профіль – бот-профілем;

3. Аналіз профілів користувачів для встановлення їх соціального

положення, так, як більш молоді користувачі в середньому більш схильні до агресивних та незважених дій та експресивного характеру ведена дискусії, що може теж вести до хибно-позитивного спрацювання алгоритму аналізу моделі. Тобто профілі в яких можливо визначити вік дописувача, вводяться в систему впливу з меншим коефіцієнтом. Також робиться імперичне припущення, що більш молоді люди ще не мають достатнього впливу на ринки, а отже мають менший кінцевий коефіцієнт впливу [9][47].

Ще один виклик під час аналізу лінгвістичної моделі мережі інтернет: хибне-позитивні спрацювання на події не фінансових ринків. Так, якщо книга з назвою «Війна» стане бестселером цілком природньо, що буде спостерігатися сплеск використання цього слова в мережах усіх типів. Для вирішення цього питання було вирішено впровадити додаткові підсистеми, які будуть:

- 1 – Враховувати контекст тексту з використанням алгоритмів з пам'яттю;
- 2 – Враховувати інформацію з інших галузей, щоб на їх основі формувати додаткові коефіцієнти ваги слів-маркерів.

Третя проблема була виявлена під час використання мереж вертикального типу та пов'язана з використанням слів-замінників для пом'якшення інформаційного поля (наприклад підміна слів: «дим» на «димок» тд). Такий підхід може зменшувати спекулятивну складову на ринках, але довгостроково може говорити про систему кризи, яка рано чи пізно може привести до системного колапсу, який буде прихований від аналізу. Для вирішення питань такого характеру пропонується створити окрему підсистему словників і розширити функціонування моделі з використанням зворотної пам'яті для більш чіткого розуміння контексту тексту.

3.2 Проектування алгоритму аналізу лінгвістичної моделі мережі інтернет

Таким чином система реалізує декілька основних етапів: Навчання та реального аналізу.

Навчання в свою чергу передбачає декілька етапів:

- На першому етапі, формуються словники слів маркерів, сталих виразів та розповсюджених слів-замінників. Це необхідно для покращення функціонування лінгвістичної моделі. Наступним кроком визначаються списки основних лідерів-думок, лідерів та їх профілі в соціальних мережах, як такі що мають підвищений вплив на інформаційне поле. Окремо створюється база ненадійних профілів, які потенційно визначаються, як профілі-боти: При першому старті цей список може порожнім, але він необхідний для більш швидкої роботи алгоритму аналізу, для зменшення часу на аналіз підозрілих профілів. І останнім створюється перелік джерел вертикального типу, до яких робиться імперичне припущення, що менші за обсягом та відомістю джерела мають менший коефіцієнт впливу аніж більше та відоміші;
- Другий етап передбачає запуск алгоритму для формування коефіцієнтів «позитивного» сценарію, «нормального» та «негативного» сценарію:
 - «Позитивний» - формується, як середнє серії наборів даних по заздалегідь прописаним найменш кризовим рокам. Середнє значення цих даних приймається за «1.5» і вважається позитивним сценарієм;
 - «Негативний» - формується, як середнє серії наборів даних по найбільш кризовим рокам. Середнє значення, яких береться за коефіцієнт «0.5» і вважається негативним;
 - «Нормальний» - формується, як середнє значення по рокам з виключенням «позитивних» та «негативних» періодів та приймається за «1».

Таким чином сам алгоритм формування коефіцієнту, який буде використовуватися для аналізу, буде приймати наступний вигляд:

- Формування базового словнику слів-маркерів;
- Проведення аналізу джерел вертикального типу та визначення частоти входу слід маркерів;
- Прийняття заздалегідь визначених вагових коефіцієнтів з урахуванням

хибно-позитивного спрацювання на слова-маркери;

- Проведення аналізу джерел горизонтального типу;
- Прийняття вагових коефіцієнтів на слова з урахуванням їх кількості відносно до профілю, який їх публікував (для профілів-ботів менший коефіцієнт, для лідерів-думок більший) та накладання на вагових коефіцієнтів хибно-позитивного спрацювання;
- Порівняння різниці результатів аналізу медіа вертикального та горизонтального. При виявленні значного відхилення провести поглиблений аналіз джерел вертикального типу, як таких, що схильні до використання слів-замінників, при близькому результаті з першим аналізом по словам маркерам провести більш глибокий аналіз мереж горизонтального типу, як таких, які схильні до сильного викривлення;
- Накладання отриманих даних на дані заздалегідь розрахованих коефіцієнтів «позитивного», «нормального», «негативного»;
- Отримані відхилення від «негативного», «нормального» та «позитивного» приймається за очікувані зміни і може використовуватися надалі;

Після отримання всіх необхідних коефіцієнтів вони зберігаються і в подальшому використовуються математичною підсистемою, яка проводить базові розрахунки.

```
новини: 4
як: 4
утримують: 4
лівому: 4
березі: 4
Кількість символів: 3
Кількість токенів: 3
Кількість унікальних токенів: 1
Найбільш уживані токени: [(['окупанти',), 3)]
Current coefficient by word: occupant
1.5
```

Рис 3.2.1 Розрахунок коефіцієнту за словами маркерами для слова «окупант»

```

GET SERIES FOR -> https://www.pravda.com.ua/news/
READED NEWS DAY-> 2023
READED NEWS MONTH-> 12
READED NEWS DAY-> 10
READED NEWS IDX-> 7432467
READED -> TRUE
READED NEWS IDX-> 7432468
READED NEWS IDX-> 7432469
READED NEWS IDX-> 7432470
SERIES -> [(['окупанти',),), 1]]
PROCEEDING NEWS DATA []
CALCULATE FOR POSITIVE PERIOD TO POSITIVE ->
FILTER BY COMMON FOR
CALCULATE COEF NOW to BEFORE FOR
Current coefficient by words:
[]
CALCULATE MEDIUM COEFICIENT
RAW SHIFT COEF FOR CURRENT to POSITIVE
1
NORMALIZED POSITIVE
1
CALCULATE FOR NEGATIVE PERIOD TO NEGATIVE ->
FILTER BY COMMON FOR
CALCULATE COEF NOW to BEFORE FOR
Current coefficient by words:
[]
CALCULATE MEDIUM COEFICIENT
RAW SHIFT COEF FOR CURRENT to NEGATIVE
1
NORMALIZED Negative

```

Рис 3.2.2 Результат розрахунків коефіцієнту та нормалізація за словами маркерами

Окремо виконується кластеризація отриманих даних за типами (ринку, сфери і тд.) на основі даних збережених в локальній базі даних системи для уникнення додаткового ускладнення алгоритму аналізу лінгвістичних моделей.

Для реалізації цього підходу використано *Naive Bayes*. Отримані групи зберігаються для подальшого виокристання математичною моделлю для формування оцінки користувача на основі налаштувань користувача, а також для покращення моделі прогнозування з виведенням коефіцієнтів впливу одних ринків на інші.

Після проведення етапу групування отриманих даних статистичних даних за типами відбувається процес формування коефіцієнтів впливу. На цьому етапі проводиться аналіз усіх згрупованих даних, та спроба визначити їх вплив один до одного на основі різних критерії наприклад дані, які відносяться до різних країн, чи дані, які відносяться до різних ринків.

Отримані коефіцієнти впливу зберігаються парами «тип»-«коефіцієнт» окремо та використовуються для додаткової перевірки моделі прогнозування.

3.3 Проектування алгоритму математичної системи

Після проведення розрахунків усіх коефіцієнтів та групування даних за типами система передає виконання математичній підсистема, яка отримавши запит на розрахунки буде виконувати наступні кроки:

- Отримання збережених в системі даних отриманих засобами відкритого API серверів Національного Банку України (далі НБУ) чи іншого джерела визначеного типом розрахунку;
- Розрахунок необхідних змінних за формулами, який показує те, як себе буде поводити одиниця аналізу (Одиниця аналізу: облігація, акція тощо) ізольовано, що утворює базову модель (модель, без накладених змін);
- На основі отриманої базової моделі створюється по одній моделі для кожного коефіцієнта на кожну з яких накладаються отримані під час аналізу коефіцієнти. Таким чином ми отримуємо моделі для гіршого, кращого, та очікуваного сценаріїв;
- Після проведення усіх трансформацій моделі для користувача формується оцінка дії в залежності від типу операції, яка разом з моделями повертається на сторону клієнта для відображення.

```

C:\Users\ViMi\Desktop\ X + v
0.5393728222996516
1000
2
53.937282229965156
UAH
4.566210045662101E-12
720
10/12/2023 19:51:39
0
3.376054139602902
133.48837209302326
1334.8837209302326
1334.8837209302326

360
0.072
1
1
0
1.0397741553529891

0.3083665338645418
1000
2
30.83665338645418
UAH
1.1415525114155252E-12
720
10/12/2023 20:42:17
0
1.7112072921504191
29.18604651162791
583.7209302325582
583.7209302325582

180
0.072
1
1
0
0.0964115198354385

```

Рис 3.3.1. Результат виконання математичної підсистеми з використанням коефіцієнту зміщення на тестових даних для «нормального» та «поганого» сценаріїв в консолі.

Окремо засобами математичної підсистеми окрім базових моделей облігацій формується оцінка користувача відповідно до типу виконаної ним операції.

Алгоритм розрахунку оцінки користувача наступний:

- Користувач провів операцію з одиницею аналізу;
- Формується статистика по всіх документах цього типу (наприклад облігації бувають короткотривалі/середньотривалі та довготривалі) з яких визначається найкраща в поточних умовах. Якщо користувач вибрав найкращий варіант він отримує 1.5 бали, якщо обрав найгірший отримує 0.5 бали. За проміжні рішення користувач отримує 1 бал;
- Наступним кроком утворена оцінка перевіряється на відповідність обраній стратегії користувача, якщо така стратегія наявна;
- Якщо стратегія користувача наявна формується коефіцієнт стратегії який

відповідає тому наскільки виконана дія відповідає обраному підходу. Якщо обрана стратегія є, і операція відповідає їй то коефіцієнт складає «1.5», якщо операція визнана нейтральною або стратегія не визначена коефіцієнт складає «1», якщо операція визнана такою, що не відповідає стратегії «0.5», якщо операція визнана ефективною, але не відповідає стратегії коефіцієнт складає «1.5», а в систему заносить повідомлення про те, що рішення було вірним, але не відповідає обраному підходу.

```
Customer Rate -- Noraml Scenarion
0.5
```

```
Customer Rate -- Bad Scenarion
0.75
```

Рис 3.3.2 Оцінка при позитивному та негативному сценаріях

В якості проміжної ланки між викликами підсистем виступає сервер на основі платформи NodeJs, який відповідає за комунікацію між системами та повернення відповіді клієнту.

3.4 Проектування алгоритму роботи інтелектуального асистенту

Після проведення усіх попередніх розрахунків дані передаються на бік клієнта для їх виведення та позначення моментів, які варті уваги. Наприклад, якщо користувач обрав негативну дію система має проінформувати його, що операція, яку він виконав не відповідає позитивному сценарію.

Наступним кроком було реалізовано модель асистента, яка може виконуватись за запитом надаючи більш детальний опис того чому якась конкретна дія є небажаною. Через потребу великої кількості ресурсів для такої моделі, було вирішено викликати її за запитом, а не показувати її.

Алгоритм дій моделі асистента наступний:

- Після того, як користувач запросив деталі по конкретній операції дані цієї операції передаються інтелектуальному агенту;
- Інтелектуальний агент завантажує в себе всі заздалегідь збережені дані,

щодо одиниць аналізу та їх груп;

- Робить запит математичній підсистемі для додаткової інформації, якщо така необхідна;
- Отримавши усю інформацію формує відповідь відповідно до типу запиту який було виконано. Наприклад, якщо цінний папір є найгіршим який можна купити в поточних умовах то система повертає пояснення оцінки, а також варіант, який системою вважався би найкращим з розрахованою математичною моделлю для «найкращого» кандидата;

```
GET ACTION RATE ->
Current ACTION RATE -> POSITIVE_ACTION_RATE
1.5
```

Рис. 3.4.1 Результат оцінки дії на основі операції купівлі при зростанні ринку для негативного сценарію

Таким чином орієнтовний потік виконання системи набуває наступного вигляду:

- Користувач заходить у систему і може зробити запит або одразу на деталі, якогось цінного паперу, з прогнозом, або, якщо він наявний в системі зробити запит на операцію купівлі/продажу;
- Система прогнозування існує ізольовано від інших і виконує функцію формування даних та коефіцієнтів;
- Математична підсистема виконується по запиту деталей облігацій та формує декілька основних математичних моделей, якщо актуальних коефіцієнтів немає, вона може зробити запит до системи прогнозування, яка повинна їх сформувати для неї;
- Система асистента може взаємодіяти з математичною підсистемою на виклик користувача, з якої вона отримає інформацію про всі розраховані моделі, а також приблизну оцінку на основі яких вона формує відповідь для користувача, щодо того наскільки та чи інша операція є доцільною.



.Рис 3.4.1 Взаємодія підсистем

Таким чином підсистема асистента маючи доступ до математичної системи та до кешованої бази даних може формувати відповіді на доцільність продажу цінного паперу зараз, чи доцільність купівлі цінного паперу пізніше.

3.5 Проектування алгоритму визначення тенденції на ринку

Окремою системою, яка використовує усі попередні підсистеми є система визначення тенденцій на ринку. Через велику кількість необхідних ресурсів для обрахування ця операція, як і виклик системи-асистента, в подальшому буде виконуватися за запитом користувача, якому надається можливість визначити тип ринків з наявних в системі та встановити часовий проміжок який він хоче проаналізувати.

Загальний алгоритм прогнозування тенденцій набуває наступного вигляду:

- Користувач визначає тип ринку та часові межі;
- Виконання передається системі прогнозування, яка послідовно розраховує коефіцієнти для кожного періоду, для періодів у

майбутньому визначається середня тенденція відносно попередній років при схожих умовах;

- Отримані значення коефіцієнтів зберігаються;
- Виконання передається математичній системі, яка формує список для кожної одиниці аналізу зі значеннями коефіцієнтів різних періодів (по 5 на один коефіцієнт);
- Отримані дані повертаються користувачу.

Отримані користувачем дані використовуються для графічного зображення тенденцій після чого користувач отримує можливість побачити поточний тренд ринку, а також оцінити наскільки сильне відхилення присутнє відносно «негативного», «позитивного», «нормального» та базового сценаріїв.

Для отримання додаткової інформації користувач має можливість зробити запит до системи асистента, яка сформує відповідь відносно базового алгоритму з урахуванням часових меж, а саме:

- Система асистент отримує дані за всі періоди, разом з коефіцієнтами для аналізу;
- Формує відношення отриманих коефіцієнтів, та коефіцієнтів впливу для визначення чи був присутній системний вплив чи це ізольована подія;
- Якщо було визначено, що мав місце системний вплив проводиться аналіз з метою визначення причини цього системного впливу. Наприклад, якщо рік визначено, як кризовий то формується базова відповідь: «Криза» на основі якої запускається більше детальний аналіз по звуженим словникам та джерелам з метою визначення, який тип «Кризи» був присутній в цей період;
- Якщо системного впливу не визначено, але зберігається значне відхилення між очікуванням та фактичним сценарієм, робить спроба простежити відкладені наслідки попередній криз. Робиться припущення, що це наслідки кризових явищ, що розтягнуті у часі;
- Після розрахування коефіцієнтів повертає відповідь користувачеві.

Таким чином користувач отримує можливість проведення більш ґрунтового аналізу та підбору інвестиційної стратегії на основі не тільки оцінки конкретної операції, а і оцінки ринку та його тенденцій в найближчий час, що відкриває можливості для більш точної оцінки дій користувача та більш надання більш якісного навчального матеріалу, оцінювання проходження якого буде враховувати дотримання обраної стратегії, та її ефективність додаючи до алгоритму оцінювання додаткові проміжні кроки, які будуть враховувати обрану користувачем стратегію, а не тільки тенденцію ринку та обраний тип операції.

```

0
y
LAST WORDS LOGITS
tensor(0.)
WORD IDX
0
y
LAST WORDS LOGITS
tensor(0.)
WORD IDX
0
y
PREDICTED
None
START READED NEWS DATA
GET SERIES FOR -> https://www.pravda.com.ua/news/
READED -> TRUE
READED -> TRUE
WORDS TO SAVE IN CSV ->
['uci', 'розділи', 'укр', 'рус', 'підтримати', 'уп', 'укр', 'рус', 'розділи', 'нов
SERIES -> []
PROCEEDING NEWS DATA []
GET SERIES FOR -> https://www.pravda.com.ua/news/
SERIES -> []
PROCEEDING NEWS DATA []
GET SERIES FOR -> https://www.pravda.com.ua/news/
READED -> TRUE
READED -> TRUE
WORDS TO SAVE IN CSV ->
['uci', 'розділи', 'укр', 'рус', 'підтримати', 'уп', 'укр', 'рус', 'розділи', 'нов

```

Рис 3.5.1 Прогнозування слів на основі лінгвістичної моделі та збереження словника інформаційного джерела в якості csv файлу

Висновки до третього розділу

Аналіз утвореної моделі дав можливість глибоко осмислити ключові аспекти проекту та визначити перспективні напрямки подальшого розвитку.

Сформували критерії за якими можливо формувати аналітичні звіти шляхом залучення аналізу профілів користувачів за категоріями: несправжні акаунти, вікова група, гендерна, расова, етнічна приналежність тощо.

Під час аналізу лінгвістичної моделі були виявлені ряд проблем, які потребували корекції утвореної моделі для подолання отриманих аномальних відхилень. Зокрема, виявлено викривлення словників, яке може впливати на точність та надійність моделі.

Наявна тенденція до хибно-позитивного спрацювання також визначена як недолік, що потребує уточнення та вдосконалення механізмів оцінювання.

Схильність систем вертикального типу до маскуванню маркерів, яке може ускладнювати розпізнавання патернів та шаблонів характерних для кризових ситуацій. Крім того, було визначено, що системи вертикального типу можуть створювати віртуальні кризи, використовуючи для цього ресурси мереж горизонтального типу.

Незважаючи на переваги мереж горизонтального типу, високий вплив окремих представників може викликати дисбаланс у функціонуванні системи прогнозування створюючи короткочасні екстремуми, які важко передбачити.

Для подолання негативних впливів, які несуть горизонтальні та вертикальні мережі на точність моделі введено декілька імперичних припущень про корегування коефіцієнту впливу. Для мереж вертикального типу слова маркери вводяться з меншою вагою відповідно до цитованості даного джерела (чим менше цитоване джерело тим менший воно має вплив). Також виведено необхідність створення словників слів-замінників та розширення використання моделей з пам'яттю для випадків коли необхідно встановити сенс великого об'єму тексту.

Для мереж горизонтального типу виведено необхідність створити списки

профілів лідерів-думок (профілів з дуже великим охопленням), які потребуються більш детального аналізу і вносять в систему з більшою вагою, а також профілі користувачів ботів, які вводяться в систему з меншою вагою, як не справжні актори ринку, але такі, що здатні впливати на справжніх акторів ринку.

Виведено необхідність врахування соціального статусу дописувача для корегування ваги слів маркерів. Наприклад для профілів молодих людей передбачається зменшення ваги маркера так, як молоді користувачі більше схильні до агресивного листування і при цьому мають в середньому невеликий вплив на ринки.

Спроектували систему формування коефіцієнтів впливу ринків один на одного та коефіцієнтів тенденцій (позитивна, негативна, нормальна).

Створили математичну підсистему, яка надає користувачеві розраховані дані, щодо обраної ним операції відповідно до 4 основних сценаріїв (поточний та з накладеними нормальним/негативним/позитивним коефіцієнтами) та повертає їх моделі (нормальну, негативну, позитивну та базову без накладання зовнішніх змін).

Окремо сформували алгоритм оцінювання успішності інвестора, та інтегрували його функціонування в навчальний-симулятор та підсистему асистента, яка здатна дати відповідь на питання доцільності виконання тієї чи іншої операції в даний момент часу та дати її очікувану оцінку.

Інтегрували в систему алгоритм прогнозування тенденцій ринку для надання користувачеві інструментів для формування власної стратегії з урахуванням фази ринку та тенденцій на ньому.

ВИСНОВКИ

В ході написання кваліфікаційної магістерської роботи було вирішено завдання із розробки системи прогнозування зміни станів фінансових ринків на основі лінгвістичної моделі мережі інтернет та інтеграції отриманої інформації з математичною підсистемою та підсистемою асистентом.

На основі цього було зроблено обґрунтування актуальності теми, що дало змогу формалізувати, основні аспекти розробки нових підсистем із використанням засобів аналізу даних та штучного інтелекту. Що дає можливість подальшої експлуатації системи окремо від будь якої середи, як самодостатній інструмент економічного аналізу на основі поточного інформаційного фону.

Проведено огляд аналогів програмного забезпечення і виявлено їх переваги та недоліки, що дало змогу сформулювати додаткові вимоги для покращення системи прогнозування та використання її в навчальних цілях.

Відібрано основні інструменти та алгоритми для розробки системи та аналізу даних. Проведено відбір найбільш актуальних в межах завдання кваліфікаційної роботи алгоритмів машинного навчання. Обрано наступні алгоритми:

- Наївний Баєсів класифікатор, як ефективний інструмент обробки та класифікації великих об'ємів даних для первинної класифікації;
- Random Forest Classifier – як алгоритм високої точності, що здатен до обробки великої кількості ознак;
- Рекурентні нейромережі (RNN) – як алгоритм, який здатен оброблювати природні мови зі зворотною пам'яттю, що важливо для обробки сенсів виразів та речень.

Проведено теоретичний опис основних алгоритмів машинного навчання, які були відібрані для реалізації в підсистемах в межах завдання кваліфікаційної роботи.

Проведено формування вимог до систем прогнозування, математичної підсистеми та системи асистента.

Спроектовано алгоритми системи прогнозування, математичної підсистеми та підсистеми асистента.

Використовуючи методи мови програмування Python із такими модулями як: `tourch`, `Pandas`, `request` перейшли до процесу аналізу даних, що дало можливість спроектувати алгоритм прогнозування відхилення на основі поточної лінгвістичної моделі мережі інтернет, для подальшого їх використання в математичній підсистемі, для формування можливих сценаріїв та формування прогнозу тенденцій та системі асистенти, для формування підказки та оцінки дій користувача на основі коефіцієнтів.

В результаті написання магістерської кваліфікаційної було отримано працездатний модуль, який надає нові можливості для проведення аналізу та прогнозування відхилень на фінансових ринках. Розроблений модуль є розширенням наявного функціоналу навчального-симулятора для роботи з облігаціями внутрішньої державної позики.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Документація JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
2. Офіційна документація Express.js [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/en/starter/installing.html>
3. Офіційна документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>.
4. Офіційна документація Mongoose [Електронний ресурс] – Режим доступу до ресурсу: <https://mongoosejs.com/docs/guides.html>.
5. Документація React.js [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>
6. Smith, J. (2019). "Machine Learning for Finance", Packed Publishing.
7. Brown, A. (2018). "Personalized Financial Literacy Training: A Comprehensive Guide", Springer.
8. Thompson, L., & Green, M. (2020). "Data Analytics in Financial Markets", Oxford University Press.
9. O'Reilly, K., & Goss, J. (2017). "Utilizing Simulation Platforms for Enhanced Investment Strategies", Journal of Financial Computing, 12(4), 295-310.
10. Patel, R. (2018). "Investment Simulators and Modern Portfolio Theory", Cambridge Finance Review, 14(3), 45-59.
11. Jackson, E. (2019). "The Role of Artificial Intelligence in Financial Education", AI & Finance Journal, 5(2), 80-91.
12. Wagner, T., & Lee, H. (2020). "Interactive Learning in Financial Literacy: A Comparative Study", Journal of Interactive Finance, 8(1), 22-38.
13. Core [Електронне Джерело] - <https://core.ac.uk/download/pdf/304295518.pdf>
14. Um [Електронне Джерело] - <http://um.co.ua/8/8-2/8-218941.html>
15. Learn [Електронне Джерело] - https://learn.ztu.edu.ua/pluginfile.php/177172/mod_resource/content/1/%D0

%9B%D0%B0%D0%B1_%D1%80%D0%B0%D0%B12_%D0%9C%D1%82%D0%90%D0%86%D0%9F%D1%82%D0%A1.pdf

16. IOP [Електронне Джерело] -
https://portal.tpu.ru/departments/otdel/publish/izdaniya_razrabotanye_v_ramkah_IOP/Tab1/hranilicha_dannyh_zac_zikl_lab_rab_zac.pdf
17. Warehouse [Електронне Джерело] -
https://www.ucg.ac.me/skladiste/blog_10838/objava_107239/fajlovi/data%20warehouse.pdf
18. OLAP [Електронне Джерело] -
https://www.academia.edu/40430628/Concepts_and_Fundaments_of_Data_Warehousing_and_OLAP_Concepts_and_Fundaments_of_Data_Warehousing_and_OLAP_2017
19. Azure Synapse [Електронне Джерело] -
https://www.lantana.be/uploads/1/2/1/5/121561526/azure_white_book_synaps_2.pdf
20. Azure - [Електронне Джерело] -
https://www.lantana.be/uploads/1/2/1/5/121561526/azure_white_book_synaps_2.pdf
21. Антонюк, Д.С., Лабенський, В.А., Медведєв, В.В., Новачук, Р.О., 2022. Проектування симулятора управління персональним портфелем облігацій внутрішньої державної позики. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, с. 43-52.
22. Вакалюк Т.А., Медведєв В.В. 2022 Використання інформаційних технологій в економіці при кризових умовах. У: Комп'ютерні технології: інновації, проблеми, рішення, 01-02 грудня 2022, Житомир, с. 44-45.
23. Медведєв В.В., Вакалюк Т.А 2022 Прогнозування фінансових ринків в умовах невизначеності. У: Сучасні інформаційні системи і технології. 30 листопада 2022, Хмельницький, с 146-148.

24. <https://corporatefinanceinstitute.com/resources/data-science/machine-learning-in-finance/>
25. Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." Springer.
26. Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.
27. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). "An Introduction to Statistical Learning." Springer.
28. Murphy, K. P. (2012). "Machine Learning: A Probabilistic Perspective." MIT Press.
29. Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." MIT Press.
30. Shalev-Shwartz, S., & Ben-David, S. (2014). "Understanding Machine Learning: From Theory to Algorithms." Cambridge University Press.
31. Caruana, R., & Niculescu-Mizil, A. (2006). "An Empirical Comparison of Supervised Learning Algorithms." Proceedings of the 23rd International Conference on Machine Learning.
32. Breiman, L. (2001). "Random Forests." Machine learning, 45(1), 5-32.
33. Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
34. Rasmussen, C. E., & Williams, C. K. I. (2006). "Gaussian Processes for Machine Learning." MIT Press. Kim, H. (2018). "Deep Learning for Image Recognition: A Comprehensive Review." Neural Computation, 30(6), 1239-1293.
35. Hastie, T., & Tibshirani, R. (1990). "Generalized Additive Models." Chapman and Hall.
36. Chen, M., Hao, Y., & Liu, Q. (2020). "A Survey of Machine Learning for Internet of Things: Applications and Techniques." IEEE Access, 8, 55043-55065.

37. Murphy, K. P. (2012). "Machine Learning: A Probabilistic Perspective." MIT Press.
38. Chollet, F. (2017). "Deep Learning with Python." Manning Publications.
39. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you? Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
40. Breiman, L. (1996). "Bagging Predictors." Machine Learning, 24(2), 123-140.
41. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). "Interpretable Representations of Data with Predictive Complexity." arXiv preprint arXiv:1610.07581.
42. Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine." Annals of Statistics, 29(5), 1189-1232.
43. Russell, S. J., & Norvig, P. (2009). "Artificial Intelligence: A Modern Approach." Pearson.
44. Георгій Почепцов. «ФЕЙК. Технології спотворення реальності» (2020), ISBN 978-966-518-746-2.
45. Деніел Еверетт. «Походження мови. Як ми навчилися говорити» (2019) ISBN 978-617-7730-49-0.
46. Wassima Lakhchini, Rachid Wahabi, Mounime El Kabbouri "Artificial Intelligence & Machine Learning in Finance: A literature review" (2022)

ДОДАТКИ

Лістинг сутності *StatsPrognose*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.Design;

using Bondary.Domain.Helpers;

namespace Bondary.Domain.Entities;

public enum ScenarioTypes
{
    POSITIVE = 1,
    NORMAL = 2,
    NEGATIVE = 3,
    CURRENT = 4,
}

public class StatsPrognose
{
    private Bond? Bond { set; get; }

    public ScenarioTypes Scenario { get; init; }

    public StatsPrognose(ScenarioTypes scenario)
    {
        Scenario = scenario;
    }

    public double HY { set; get; }
    public int OwnDays { set; get; }
    public double DaysToPayDate { set; get; }
    public double YearsInWork { set; get; }
    public double DaysBetweenBuyPayDate { set; get; }
    public double YearsToPayDate { set; get; }
    public double ACY { set; get; }
    public double YMTSimpleApproximation { set; get; }
    public double AllFutureCouponsCount { set; get; }
    public double SimpleYTM { set; get; }
    public double YTMWithReinvest { set; get; }
    public double CouponPercentage { set; get; }
    public double BEY { set; get; }
    public double EAY { set; get; }
    public double CurrentIncome { set; get; }
    public double DaysFromStartCouponPeriod { set; get; }
    public double AllCouponIncome { set; get; }
    public double Y { set; get; }
    public double CouponY { set; get; }
    public double CurrentY { set; get; }
    public double ModifiedCurrentY { set; get; }
    public double BondCourse { set; get; }
    public double Selly { set; get; }
    public int PaysDuringYear
    {
        get
    }

```

```

    {
        return 2;
    }
}

public double CouponIncome
{
    get
    {
        return YearCouponIncome / PaysDuringYear;
    }
}

public double ClearMarketPriceFixedCoupon
{
    get
    {
        return PriceWithFixedCoupon;
    }
}

public double DirtyMarketPriceFixedCoupon
{
    get
    {
        return ClearMarketPriceFixedCoupon + ACY;
    }
}

public double BoughtPrice
{
    get
    {
        return ClearMarketPriceFixedCoupon;

        // (
        // ((Bond.Nominal * (ClearMarketPriceFixedCoupon/100))) + ACY );

        // )
    }
}

public double PriceWithFixedCoupon
{
    get
    {
        var coupon = CouponIncome;
        var nominal = Bond.Nominal;

        double price = 0;
        int i = 0;

        for (; i < Bond.PayDates.Count(); ++i)
            price += coupon / Math.Pow(1 + Bank.MarketPercentage, i + 1);

        price += nominal / Math.Pow(1 + Bank.MarketPercentage, i - 1);
        return price;
    }
}

public double YearCouponIncome
{

```

```

        get
        {
            return ((Bond.YearCoupon * Bond.Nominal) / 100);
        }
    }
}

```

Лістинг сервісу BondService

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using Bondary.Application.Common.Interfaces;
using Bondary.Application.Bond.Models;

using Bondary.Application.Bond.Commands.Chart;
using Bondary.Application.Bond.Commands.Stats;

using Bondary.Domain.Entities;
using Bondary.Domain.Helpers;
using Bondary.Domain.Enums;

using Bondary.Application.Bond.Models.Enums;
using System.Runtime.InteropServices;

namespace Bondary.Application.Bond.Service;

public class BondService
    : IBondService
{

    public async Task<EstimationRates> GetRate(Domain.Entities.Bond bond)
    {

        var price = bond.Statistic.DirtyMarketPriceFixedCoupon;
        var yeild = bond.Statistic.Y;
    }
}

```

```

        if (yeild >= 10) return EstimationRates.High;
        if(yeild < 10 && yeild >= 5) return EstimationRates.Good;
        if (yeild <= 5 && yeild >= 1) return EstimationRates.Average;
        if (yeild < 1 && yeild >= 0) return EstimationRates.Low;
        if (yeild < 0) return EstimationRates.Awfull;

        return EstimationRates.Average;
    }

    public double TranslateRate(EstimationRates rates)
    {
        switch(rates)
        {
            case EstimationRates.High: return 1.5;
            case EstimationRates.Good: return 1;
            case EstimationRates.Average: return 1;
            case EstimationRates.Low: return 0.5;
            case EstimationRates.Awfull: return 0.5;
            default: return 1;
        }
    }

    public double applyRateCoeff(double rawRate, double coeff)
    {
        return rawRate * coeff;
    }

    public Domain.Entities.StatsPrognose CreateScenarios(Domain.Entities.Bond
bond, ScenarioTypes scenario, double shiftCoeff)
    {
        var prognose = new StatsPrognose(scenario);

        prognose.EAY = bond.Statistic.EAY * shiftCoeff;
        prognose.Y = bond.Statistic.Y * shiftCoeff;
        prognose.ACY = bond.Statistic.ACY * shiftCoeff;
        prognose.AllCouponIncome = bond.Statistic.AllCouponIncome * shiftCoeff;
    }

```

```

        prognose.AllFutureCouponsCount = bond.Statistic.AllFutureCouponsCount *
shiftCoeff;

        prognose.CouponY = bond.Statistic.CouponY * shiftCoeff;

        prognose.CurrentIncome = bond.Statistic.CurrentIncome * shiftCoeff;

        prognose.ModifiedCurrentY = bond.Statistic.ModifiedCurrentY * shiftCoeff;

        prognose.YTMWithReinvest = bond.Statistic.YTMWithReinvest * shiftCoeff;

        prognose.YMTSimpleApproximation = bond.Statistic.YMTSimpleApproximation *
shiftCoeff;

        return prognose;
    }

```

```

public async Task<BondStatDto> GetBondStats (Domain.Entities.Bond bond)
{
    var priceInEvenryDate = new CalculatePriceInEveryDate().Build(new
Domain.Entities.Bond(bond));

    var chartACY = new CalculateACYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var acyForAllPeriods = new CalculateACYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var beyForAllPeriods = new CalculateBEYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var eayForAllPeriods = new CalculateEAYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var hyForAllPeriods = new CalculateHYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var paymentsAllPeriod = new CalculatePaymentsYield().Build(new
Domain.Entities.Bond(bond));

    var yForAllPeriods = new CalculateYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var ytmForAllPeriods = new CalculateYTMForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var sellYForAllPeriods = new CalculateSellYForAllPeriods().Build(new
Domain.Entities.Bond(bond));

    var whenPriceProfitable = new
CalculateWhenPriceProfitably().Calculate(priceInEvenryDate,
bond.Statistic.BoughPrice);

    var whenYieldGreaterBuyPrice = new
CalculateWhenYieldGreaterBuyPrice().Calculate(new Domain.Entities.Bond(bond));

    var forAllCouponFromOwnPeriod = new
CalculateChartForAllCouponFromOwnPeriod().Calculate(new
Domain.Entities.Bond(bond));

    var allPriceInEveryDate = new CalculatePriceInEveryDate().Build(new
Domain.Entities.Bond(bond));

```

```

var allPriceInEveryDateACY = new CalculatePriceInEveryDateACY().Build(new
Domain.Entities.Bond(bond), chartACY);

var nominal = bond.Nominal;
var buyDate = bond.BuyDate;
var code = bond.Code;
var currency = bond.Currency.ToString();
var currentDate = bond.CurrentDate;
var payDates = bond.PayDates;
var prevPayDates = bond.PrevPayDate;
var repayDate = bond.RepayDate;
var type = bond.Type.ToString();
var yearCoupon = bond.YearCoupon;
var acy = bond.Statistic.ACY;

var allCouponIncome = bond.Statistic.AllCouponIncome;
var allFutureCouponsCount = (int)bond.Statistic.AllFutureCouponsCount;
var ownDays = bond.Statistic.OwnDays;
var paysDuringYear = bond.Statistic.PaysDuringYear;
var sellY = bond.Statistic.SellY;
var bondCourse = bond.Statistic.BondCourse;
var bey = bond.Statistic.BEY;
var buyPrice = bond.Statistic.BoughPrice;
var sellPrice = bond.Statistic.ClearMarketPriceFixedCoupon;
var clearMarketPriceFixedCoupon =
bond.Statistic.ClearMarketPriceFixedCoupon;
var couponIncome = bond.Statistic.CouponIncome;
var couponPercentage = bond.Statistic.CouponPercentage;
var couponY = bond.Statistic.CouponY;
var currentIncome = bond.Statistic.CurrentIncome;
var currentY = bond.Statistic.CurrentY;
var daysBetweenBuyPayDate = (int)bond.Statistic.DaysBetweenBuyPayDate;
var daysFromStartCouponPeriod =
(int)bond.Statistic.DaysFromStartCouponPeriod;
var daysToPayDate = (int)bond.Statistic.DaysToPayDate;
var dirtyMarketPriceFixedCoupon =
bond.Statistic.DirtyMarketPriceFixedCoupon;
var eay = bond.Statistic.EAY;

```

```

var hy = bond.Statistic.HY;
var modifiedCurrentY = bond.Statistic.ModifiedCurrentY;
var priceWithFixedCoupon = bond.Statistic.PriceWithFixedCoupon;
var YTM = bond.Statistic.SimpleYTM;
var Y = bond.Statistic.Y;
var yearCouponIncome = bond.Statistic.YearCouponIncome;
var yearsInWork = (int)bond.Statistic.YearsInWork;
var yearsToPayDate = (int)bond.Statistic.YearsToPayDate;
var ymtSimpleApproximation = bond.Statistic.YMTSimpleApproximation;
var ytmWithReinvest = bond.Statistic.YTMWithReinvest;

```

```

var statDto = new BondStatDto()
{
    Nominal                = nominal,
    BuyDate                 = buyDate,
    Code                    = code,
    Currency                = currency,
    CurrentDate             = currentDate,
    PayDates                = payDates,
    PrevPayDate             = prevPayDates,
    RepayDate               = repayDate,
    Type                    = type,
    YearCoupon              = yearCoupon,
    ACY                     = acy,
    AllCouponIncome         = allCouponIncome,
    AllFutureCouponsCount   = allFutureCouponsCount,
    OwnDays                 = ownDays,
    PaysDuringYear          = paysDuringYear,
    Selly                   = sellY,
    BondCourse              = bondCourse,
    BEY                     = bey,
    BuyPrice                = buyPrice,
    SellPrice               = clearMarketPriceFixedCoupon,
    ClearMarketPriceFixedCoupon = clearMarketPriceFixedCoupon,
    CouponIncome            = couponIncome,

```

```

CouponPercentage      = couponPercentage,
CouponY               = couponY,
CurrentIncome         = currentIncome,
CurrentY              = currentY,
DaysBetweenBuyPayDate = daysBetweenBuyPayDate,
DaysFromStartCouponPeriod = daysFromStartCouponPeriod,
DaysToPayDate         = daysToPayDate,
DirtyMarketPriceFixedCoupon = dirtyMarketPriceFixedCoupon,
EAY                   = eay,
HY                     = hy,
ModifiedCurrentY      = modifiedCurrentY,
PriceWithFixedCoupon  = priceWithFixedCoupon,
YTM                    = YTM,
Y                      = Y,
YearCouponIncome      = yearCouponIncome,
YearsInWork           = yearsInWork,
YearsToPayDate        = yearsToPayDate,
YMTSimpleApproximation = ymtSimpleApproximation,
YTMWithReinvest       = ytmWithReinvest,

ACYForAllPeriods      = acyForAllPeriods,
BEYForAllPeriods      = beyForAllPeriods,
EAYForAllPeriods      = eayForAllPeriods,
HYForAllPeriods       = hyForAllPeriods,
PaymentsAllPeriod     = paymentsAllPeriod,
YForAllPeriods        = yForAllPeriods,
YTMForAllPeriods      = ytmForAllPeriods,
SellyForAllPeriods    = sellYForAllPeriods,
WhenPriceProfitable   = whenPriceProfitable,
WhenYieldGreaterBuyPrice = whenYieldGreaterBuyPrice,
    ForAllCouponFromOwnPeriod = forAllCouponFromOwnPeriod,
    AllPriceInEveryDate       = allPriceInEveryDate,
    AllPriceInEveryDateACY    = allPriceInEveryDateACY,

};

```

```

        return statdDto;
    }

    public async Task<BondDto> PrepareBond(BondDto bond)
    {
        var createdBond = await this.Map(bond);
        createdBond.Statistic = new Domain.Entities.BondStatistic(createdBond);
        var dto = new BondDto()
        {
            RepayDate = createdBond.RepayDate,
            EmitDate = createdBond.EmitDate,
            Y = createdBond.Statistic.Y,
            YTM = createdBond.Statistic.YMTSimpleApproximation,
            BondType = createdBond.Type.ToString(),
            TermType = Domain.Helpers.BondTerms.GetBondTermString(createdBond),
            BuyPrice = createdBond.Statistic.BoughPrice,
            Code = bond.Code,
            Currency = bond.Currency,
            NearestPayDate = createdBond.NearesPayDate,
            PaymentSchedule = createdBond.PayDates.ToArray(),
            Nominal = createdBond.Nominal,
            SellPrice = createdBond.Statistic.DirtyMarketPriceFixedCoupon,
            YearCoupon = createdBond.YearCoupon,
        };
        return dto;
    }

    public async Task<Domain.Entities.Bond> Map(BondDto bond)
    {
        var payDates = Bank.CalculateApproximatelyPayDates(bond.EmitDate,
bond.RepayDate);
        var nearest = payDates.FirstOrDefault();
        var currency = Currency.Map(bond.Currency);

```

```
return new()  
{  
    BuyDate      = DateTime.Now,  
    PayDates     = payDates,  
    EmitDate     = bond.EmitDate,  
    YearCoupon   = bond.YearCoupon,  
    RepayDate    = bond.RepayDate,  
    NearesPayDate = nearest,  
    Nominal      = Bank.DefaultNominal,  
    Currency     = currency,  
    Type         = BondType.Coupon,  
    Code         = bond.Code,  
};  
}  
}
```

Додаток Б Лістинг системи прогнозування та асистенту

Read_news_utils.py

```

import requests
from bs4 import BeautifulSoup
import re
import random
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
from collections import Counter
import nltk
import pandas as pd
import pymorphy2
import numpy as np
import seaborn
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # data visualization
import seaborn as sns # statistical data visualization
import csv
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB

import category_encoders as ce

import coff_constants
import rnn_constants
import LM_types

def get_vertical_series_wage():
    wage_list = []

    # Extend to calculation on wage news
    for origin in coff_constants.VERTICAL_NEWS:
        news_tuple = (origin[0], coff_constants.DEFAULT_NEWS_WAGE)
        wage_list.append(news_tuple)

    return wage_list

def clear_base_csv():
    f = open('csv_words.csv', 'a')
    # create the csv writer
    f.truncate()

def write_words_as_csv(words):
    # open the file in the write mode
    f = open('csv_words.csv', 'a')
    # create the csv writer

```

```

writer = csv.writer(f)
# write a row to the csv file
writer.writerow(words)

def read_words_from_origin(url):
    response = requests.get(url)
    if response.status_code == 200:
        print("READED -> TRUE")
        soup = BeautifulSoup(response.content, 'html.parser')
        news = soup.body.get_text()
        # remove tags

        news = re.sub(coff_constants.FILTER_TAG_PATTERN, '', news)

        # to token
        news = re.sub(coff_constants.LOCALE_PATTE_R_NUMBERS, r" ", news)
        news = re.sub(coff_constants.LOCALE_PATTE_R, ' ', news)
        rows = news.lower().split("\n")
        rows = [row for row in rows if row]
        rows_words = [[word for word in row.split(" ") if word] for row in rows]
        words = [word for words in rows_words for word in words if word]

        print("READED -> TRUE")

        return words

    else:
        return None

# Calculate word by step by web scratching
def series_get(url_path, year_from, year_to, month_from, month_to, day_from,
day_to, index_from=0,
                index_to=coff_constants.MAX_TEST_IDX, write_words_to_csv = True):
    completed_series = []
    origin_urls = get_origin_urls(url_path, year_from, year_to, month_from,
month_to, day_from, day_to, index_from,
                                index_to)

    for origin in origin_urls:
        words = read_words_from_origin(origin)

        if write_words_to_csv == True and words is not None:
            write_words_as_csv(words)

        if words is not None:
            stop_words = coff_constants.STOP_MARKERS
            stop_words =
set(stop_words.split(coff_constants.STOP_MARKERS_SEPARATOR))
            morph = pymorphy2.MorphAnalyzer(lang=coff_constants.LANG)
            series = pd.Series(words)
            series = [s for s in series if s in stop_words]

            series = [morph.parse(s)[0].normal_form for s in series]
            series = nltk.ngrams(series, 1)
            series = nltk.FreqDist(series)

            for word, freq in series.most_common():
                new_tuple = (word, freq)
                completed_series.append(new_tuple)

    return completed_series

```

```

def get_base_calculation(year_from, year_to, month_from, month_to, day_from,
day_to, index_from=0,
                        index_to=coeff_constants.MAX_TEST_IDX):
    proceeded = []

    vertical_wages = get_vertical_series_wage()

    for vertical_origin in coeff_constants.VERTICAL_NEWS:
        print("GET SERIES FOR -> ", vertical_origin)
        news_series = series_get(vertical_origin, year_from, year_to,
month_from, month_to, day_from, day_to,
                                index_from, index_to)

        print("SERIES -> ", news_series)

        if len(news_series) == 0:
            continue

        if len(proceeded) == 0:
            for news in news_series:
                for wages in vertical_wages:
                    if wages[0] == news[0]:
                        proceeded.append((news[0], news[1] * wages[1]))
        else:
            for news in news_series:
                for word in proceeded:
                    if word[0] == news[0]:
                        for wages in vertical_wages:
                            if wages[0] == word[0]:
                                new_proceeded = (word[0], (word[1] + news[1]) *
wages[1])
                                proceeded.remove(word)
                                proceeded.append(new_proceeded)

            print("PROCEEDING NEWS DATA ", proceeded)
            return proceeded

def get_common_from(now, before):
    print("FILTER BY COMMON FOR")
    common = [word for word in now if word in before]

def calculate_shift_cof_linked_with(now, before):
    print("CALCULATE COF NOW to BEFORE FOR ")
    proceeded_data_shift_coff = []
    for word in now:
        for word2 in before:
            if word2[0] == word[0]:
                current_tuple = (word[0], float(word2[1] / word[1]))
                proceeded_data_shift_coff.append(current_tuple)

    print("Current coefficient by words:")
    print(proceeded_data_shift_coff)

    print("CALCULATE MEDIUM COFFICIENT")
    coeff = 0
    for word in proceeded_data_shift_coff:
        coeff += word[1]

    raw_coff = 1

    if len(proceeded_data_shift_coff) > 0:

```

```

        raw_coff = coeff / len(proceeded_data_shift_coff)

    return raw_coff

def normalize_coff(coff, scenario):
    if scenario == coff_constants.POSITIVE_SCENARIO:
        if coff > 1:
            return coff_constants.NEGATIVE
        if coff <= 1:
            return coff_constants.POSITIVE

    if scenario == coff_constants.NEGATIVE_SCENARIO:
        if coff >= 1:
            return coff_constants.NEGATIVE
        if coff < 1:
            return coff_constants.POSITIVE

    if scenario == coff_constants.NORMAL_SCENARIO:
        if coff < 1:
            return coff_constants.POSITIVE
        if coff == 1:
            return coff_constants.NORMAL
        if coff > 1:
            return coff_constants.NEGATIVE

def save_coff_txt(positive_shift, normal_shift, negative_shift):
    f = open(coff_constants.COFF_STORAGE_TXT_NAME, "a")
    f.truncate()
    coff_string = str(positive_shift) + ',' + str(normal_shift) + ',' +
str(negative_shift) + "\n"
    f.write(coff_string)
    f.close()

def read_coff_txt():
    f = open(coff_constants.COFF_STORAGE_TXT_NAME, "r")
    return f.read()

def get_origin_urls(url_path, year_from, year_to, month_from, month_to,
day_from, day_to, index_from=0,
                    index_to=coff_constants.MAX_TEST_IDX):
    url = url_path
    current_year = year_from
    current_month = month_from
    current_day = day_from
    current_news_idx = index_from

    origin_urls = []

    while current_year <= year_to:
        url += str(current_year) + '/'
        while current_month <= month_to:
            url += str(current_month) + '/'
            while current_day <= day_to:
                url += str(current_day) + '/'
                while current_news_idx <= index_to:
                    url += str(current_news_idx) + "/"
                    origin_urls.append(url)
                    current_news_idx += 1
                current_day += 1
            current_month += 1
        current_year += 1

```

```

        current_year += 1

    return origin_urls

def predict_origins_vertical(rnn_mode):
    for vertical_origin in coff_constants.VERTICAL_NEWS:
        predict_by_origin(vertical_origin, rnn_mode)

def predict_by_origin(url, rnn_mode):
    words = read_words_from_origin(url)

    if words is not None:
        words = list(words)[: 30000]
        print("PREDICT TEST ->")
        print(words)

        dataset = LM_types.LanguageModelDataset(words,
rnn_constants.SEQUENCE_LENGTH)
        data_loader = DataLoader(dataset, batch_size=rnn_constants.BATCH_SIZE,
shuffle=True)
        device = torch.device(rnn_mode)
        model = LM_types.LanguageModel(len(dataset.uniq_words)).to(device)
        optimizer = optim.Adam(model.parameters()),
lr=rnn_constants.LEARNING_RATE)
        criterion = nn.CrossEntropyLoss()
        predict(dataset, model, words)

def predict(dataset, model, text, next_words=10):
    model.eval()
    words = text.split(' ')
    state_h, state_c = model.init_state(len(words))

    for i in range(0, next_words):
        x = torch.tensor([[dataset.word_to_index[w] for w in words[i:]])
        y_pred, (state_h, state_c) = model(x, (state_h, state_c))
        last_word_logits = y_pred[0][-1]
        p = torch.nn.functional.softmax(last_word_logits,
dim=0).detach().numpy()
        word_index = np.random.choice(len(last_word_logits), p=p)
        words.append(dataset.index_to_word[word_index])

    return words

```

assistant_utils.py

```

import coff_constants
import assist_constants

def get_action_rate_scenarion(scenario, operation, coef):
    if scenario == coff_constants.NEGATIVE_SCENARIO:
        if coef == coff_constants.NORMAL or coef == coff_constants.POSITIVE and
operation == assist_constants.SELL:
            return assist_constants.NEGATIVE_ACTION_RATE
        elif coef == coff_constants.NEGATIVE and operation ==
assist_constants.SELL:
            return assist_constants.POSITIVE_ACTION_RATE
        elif coef == coff_constants.NORMAL or coef == coff_constants.POSITIVE
and operation == assist_constants.BUY:
            return assist_constants.POSITIVE_ACTION_RATE
        elif coef == coff_constants.NEGATIVE and operation ==
assist_constants.BUY:

```

```

        return assist_constants.NORMAL_ACTION_RATE

    elif scenario == coff_constants.POSITIVE_SCENARIO:
        if coef == coff_constants.NORMAL or coef == coff_constants.POSITIVE and
operation == assist_constants.SELL:
            return assist_constants.POSITIVE_ACTION_RATE
        elif coef == coff_constants.NEGATIVE and operation ==
assist_constants.SELL:
            return assist_constants.NORMAL_ACTION_RATE
        elif coef == coff_constants.NORMAL or coef == coff_constants.POSITIVE
and operation == assist_constants.BUY:
            return assist_constants.NEGATIVE_ACTION_RATE
        elif coef == coff_constants.NEGATIVE and operation ==
assist_constants.BUY:
            return assist_constants.NEGATIVE_ACTION_RATE

    if scenario == coff_constants.NORMAL_SCENARIO:
        return assist_constants.NORMAL_ACTION_RATE

```

LM_types.py

```

import torch
import torch.nn as nn
from torch import device
from torch.utils.data import Dataset, DataLoader
from collections import Counter

import rnn_constants

#RNN language model
class LanguageModel(nn.Module):
    def __init__(self, vocab_size):
        super(LanguageModel, self).__init__()
        self.lstm_size = rnn_constants.LSTM_SIZE
        self.embedding_dim = rnn_constants.EMBEDDING_DIM
        self.num_layers = rnn_constants.NUM_LAYERS
        self.embedding = nn.Embedding(
            num_embeddings=vocab_size,
            embedding_dim=self.embedding_dim,
        )
        self.lstm = nn.LSTM(
            input_size=self.lstm_size,
            hidden_size=self.lstm_size,
            num_layers=self.num_layers,
            dropout=rnn_constants.DROPOUT_RATE,
        )
        self.fc = nn.Linear(self.lstm_size, vocab_size)

    def forward(self, x, prev_state):
        self.model.to(rnn_constants.CUDA_MODEL)
        embed = self.embedding(x)
        output, state = self.lstm(embed, prev_state)
        logits = self.fc(output)
        return logits, state

    def init_state(self, sequence_length):
        return (
            torch.zeros(self.num_layers, sequence_length,
self.lstm_size).to(device),
            torch.zeros(self.num_layers, sequence_length,
self.lstm_size).to(device)
        )

```

```

class LanguageModelDataset(torch.utils.data.Dataset):
    def __init__(self, words, sequence_length):
        self.sequence_length = sequence_length
        self.words = words
        self.uniq_words = self.get_uniq_words()
        self.index_to_word = {index: word for index, word in
            enumerate(self.uniq_words)}
        self.word_to_index = {word: index for index, word in
            enumerate(self.uniq_words)}
        self.words_indexes = [self.word_to_index[w] for w in self.words]

    def get_uniq_words(self):
        word_counts = Counter(self.words)
        return sorted(word_counts, key=word_counts.get, reverse=True)

    def __len__(self):
        return len(self.words_indexes) - self.sequence_length

    def __getitem__(self, index):
        return (
            torch.tensor(self.words_indexes[index:index +
            self.sequence_length]),
            torch.tensor(self.words_indexes[index + 1:index +
            self.sequence_length + 1]),
        )

```

Read_news.py

```

import read_news_utils
import coff_constants
import rnn_constants

waged = read_news_utils.get_vertical_series_wage()
print("START READED NEWS DATA")

normal = read_news_utils.get_base_calculation(2022, 2022, 12, 12, 16, 16,
7381133, 7381136)
negative = read_news_utils.get_base_calculation(2022, 2022, 2, 2, 24, 26,
7432467, 7432507)
positive = read_news_utils.get_base_calculation(2021, 2021, 11, 11, 10, 12,
7313561, 7313601)

current = read_news_utils.get_base_calculation(2023, 2023, 12, 12, 10, 10,
7432467, 7432470)

print("CALCULATE FOR POSITIVE PERIOD TO POSITIVE -> ")
raw_coef_positive = read_news_utils.calculate_shift_cof_linked_with(current,
positive)
print("RAW SHIFT COEF FOR CURRENT to POSITIVE")
print(raw_coef_positive)

normalize_positive = read_news_utils.normalize_coff(raw_coef_positive,
coff_constants.POSITIVE_SCENARIO)
print("NORMALIZED POSITIVE")
print(normalize_positive)

print("CALCULATE FOR NEGATIVE PERIOD TO NEGATIVE -> ")
raw_coef_negative = read_news_utils.calculate_shift_cof_linked_with(current,
negative)
print("RAW SHIFT COEF FOR CURRENT to NEGATIVE")

```

```

print(raw_coef_negative)

normalize_negative = read_news_utils.normalize_coff(raw_coef_positive,
coff_constants.NEGATIVE_SCENARIO)
print("NORMALIZED Negative")
print(normalize_negative)

print("CALCULATE FOR NORMAL PERIOD TO NORMAL -> ")
raw_coef_normal = read_news_utils.calculate_shift_cof_linked_with(current,
positive)
print("RAW SHIFT COEF FOR CURRENT to NORMAL")
print(raw_coef_normal)

normalize_normal = read_news_utils.normalize_coff(raw_coef_positive,
coff_constants.POSITIVE_SCENARIO)
print("NORMALIZED NORMAL")
print(normalize_normal)

read_news_utils.save_coff_txt(normalize_positive, normalize_normal,
normalize_negative)

print("PREDICT ORIGINS VERTICAL RNN")
read_news_utils.predict_origins_vertical(rnn_constants.CPU_MODEL)

```