

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка методів та програмних засобів для формування портфелю інвестицій в
облігації

Виконав: студент II курсу
групи 1ПІ- 23м спеціальності
121 – Інженерія програмного забезпечення
Костюк Руслан Олександрович

Керівник: к.т.н., доц. каф. ПЗ Ткаченко О.М.

« 09 » грудня 2024р.

Опонент: к.т.н., доц. каф. ОТ Тарновський М.Г.

« 09 » грудня 2024р.

Допущено до захисту

Завідувач кафедри ПЗ

д.т.н., проф. Романюк О. Н.

(прізвище та ініціали)

« 09 » грудня 2024 р.

ВНТУ – 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти II-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор Романюк О.Н.

« 18 » вересня 2024 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Костюку Руслану Олександровичу

1. Тема роботи – Розробка методів та програмних засобів для формування портфелю інвестицій в облігації.

Керівник роботи: Ткаченко Олександр Миколайович, д.т.н., завідувач кафедри ПЗ, затверджені наказом вищого навчального закладу від « 17 » вересня 2024 р. № 310.

2. Строк подання студентом роботи

3 грудня 2024 р.

3. Вихідні дані до роботи: PyCharm, MongoDB, мови програмування Python, операційна система – Linux, методи машинного навчання: Random Forest

4. Зміст текстової частини: вступ; аналіз методів прогнозування ринкових цін облігацій внутрішньої державної позики (ОВДП); теоретичні основи прогнозування ринкових цін ОВДП за допомогою машинного навчання; розробка програмних засобів для прогнозування ринкових тенденцій цін ОВДП; впровадження технологій машинного навчання в прогнозуванні ринкових тенденцій цін ОВДП; економічна частина; висновки; список літератури; додатки.

5. Перелік ілюстративної частини: титульний слайд, наукова новизна, мета і задачі роботи, об'єкт та предмет дослідження, актуальність теми, порівняльний аналіз аналогів, архітектура системи, UML-діаграма станів системи, головна UML-діаграма послідовності, складові навчальної вибірки, UML-діаграма послідовності завантаження даних, тестування системи, порівняння графіків поточних та прогнозованих значень, висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Ткаченко О.М., к.т.н., доцент кафедри ПЗ	19.09.2024 <i>Алс</i>	02.12.2024 <i>Алс</i>
5	Причепя І. В., к.е.н., доцент кафедри ЕПВМ	24.11.2024 <i>Алс</i>	30.11.2024 <i>Алс</i>

7. Дата видачі завдання 19 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз програмних засобів та методів прогнозування тенденцій цін ОВДП	20.09. 2024 – 30.09. 2024	<i>Вик.</i>
2	Розробка логіки побудови вибірки навчальних даних та методів машинного навчання	01.10. 2024 – 08.10. 2024	<i>Вик.</i>
3	Побудова архітектури системи для поєднання оптимальної навчальної вибірки з обраною моделлю	09.10. 2024 – 02.11. 2024	<i>Вик.</i>
4	Програмна реалізація системи та експериментальний аналіз моделі прогнозування цінових тенденцій	03.11. 2024 – 24.11. 2024	<i>Вик.</i>
5.	Економічна частина	24.11. 2024 - 30.11.2024	<i>Вик.</i>

Студент

Алс
(підпис)

Костюк Р. О.

(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи

Алс
(підпис)

Ткаченко О. М.

(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.8

Костюк Р.О. Розробка методів та програмних засобів для формування портфелю інвестицій в облігації: магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2024. 107 с.

На укр. мові. Бібліогр.: 53 назв; рис.: 28; табл. 12.

Ця магістерська кваліфікаційна робота присвячена дослідженню процесів прогнозування ринкових тенденцій цін ОВДП в комп'ютерних системах з використанням технологій машинного навчання. Основну увагу приділено дослідженню сучасних методів машинного навчання для прогнозування, а також формуванню вибірки навчальних даних з використанням макроекономічних показників та даних ОВДП.

Дослідження включає проектування, розробку та тестування алгоритмів для прогнозування ринкових цін ОВДП, спрямованих на підвищення точності прогнозування. Також дослідження охоплює розробку архітектури веб-системи.

Практична цінність роботи полягає у покращенні прийняття рішень українськими інвесторами завдяки підвищенню прогнозованості ринку цінних паперів.

Ключові слова: машинне навчання, аналіз даних, фінансові інструменти, веб-архітектура, фінансові API.

ABSTRACT

Ruslan Kostiuk. Development of methods and software tools for the formation of a portfolio of investments in government bonds: master's qualification work in specialty 121 - software engineering, educational program - software engineering. Vinnytsia: VNTU, 2024. 107 p.

In Ukrainian. Bibliogr.: 53 titles; 28 figs, 12 tables.

This master's qualification work is devoted to the study of the processes of forecasting market trends in the prices of government bonds in computer systems using machine learning technologies. The main focus is on the study of modern machine learning methods for forecasting, as well as the formation of a sample of training data using macroeconomic indicators and government bonds data.

The research includes the design, development, and testing of algorithms for predicting the market prices of domestic government bonds aimed at improving the accuracy of the prediction. The research also includes the development of a web-based system architecture.

The practical value of the work is to improve decision-making by Ukrainian investors by increasing the predictability of the securities market.

Keywords: machine learning, data analysis, financial instruments, web architecture, financial APIs.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ РИНКОВИХ ЦІН ОВДП	8
1.1 Аналіз сучасних підходів до прогнозування цінових тенденцій ОВДП....	8
1.2 Аналіз методів прогнозування цінових тенденцій ОВДП	10
1.3 Аналітичний огляд програмних засобів прогнозування цінових тенденцій ОВДП.....	12
1.4 Аналіз джерел навчальних даних для прогнозування цінових тенденцій ОВДП.....	17
1.5 Формування вимог до програмного забезпечення для прогнозування ринкових тенденцій цін ОВДП	22
1.6 Висновки	23
2 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ.....	25
2.1 Прогнозування ринкових тенденцій цін ОВДП.....	25
2.2 Оптимізація вибірки навчальних даних	26
2.3 Вибір методу машинного навчання	29
2.4 Поєднання методу машинного навчання та вибірки навчальних даних..	36
2.5 Висновки	38
3. РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП.....	39
3.1 Побудова архітектури застосунку	39
3.2 Вибір мови програмування	42
3.3 Вибір програмних засобів розробки	46
3.4 Вибір системи управління базами даних.....	50

	3
3.5 Реалізація логіки побудови вибірки навчальних даних	52
3.6 Реалізація логіки алгоритму машинного навчання	58
3.7 Висновки	62
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП	63
4.1 Підготовка середовища та налаштування розробленої системи.....	63
4.2 Тестування розробленого програмного засобу	64
4.3 Аналіз результатів прогнозування цінових тенденцій ОВДП.....	70
4.4 Висновки	78
5 ЕКОНОМІЧНА ЧАСТИНА.....	80
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	80
5.2 Розрахунок витрат на проведення науково-дослідної роботи	85
5.2.1 Витрати на оплату праці.....	85
5.2.2 Відрахування на соціальні заходи	87
5.2.3 Сировина та матеріали.....	88
5.2.4 Розрахунок витрат на комплектуючі	89
5.2.5 Спецустаткування для наукових (експериментальних) робіт	89
5.2.6 Програмне забезпечення для наукових (експериментальних) робіт	89
5.2.7 Амортизація обладнання, програмних засобів та приміщень	90
5.2.8 Паливо та енергія для науково-виробничих цілей	91
5.2.9 Службові відрядження.....	92
5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	93
5.2.11 Інші витрати.....	93
5.2.12 Накладні (загальновиробничі) витрати.....	93
5.3 Висновки	99

ВИСНОВКИ.....	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	101
ДОДАТКИ.....	106
Додаток А - Технічне завдання.....	107
Додаток Б - Протокол перевірки навчальної роботи.....	110
Додаток В – Лістинг програми	111
Додаток Г – Ілюстрована частин	121

ВСТУП

Обґрунтування вибору теми дослідження. Актуальність дослідження зумовлена необхідністю вдосконалення методів прогнозування ринкових тенденцій цінних паперів, зокрема цін облігацій внутрішньої державної позики (ОВДП). Зростаюча доступність та популярність облігацій внутрішньої державної позики призвела до загострення проблематики ризиків, пов'язаних з коливанням процентних ставок, інфляційними очікуваннями та іншими макроекономічними факторами. Незважаючи на державну гарантію, існує ряд ризиків, що впливають на фактичний прибуток, серед яких значну роль відіграє ризик коливання цін [1].

Міністерство фінансів, відповідно до покладених на нього обов'язків Бюджетним кодексом, здійснює випуск (емісію) державних облігацій з метою фінансування дефіциту державного бюджету та рефінансування державного боргу [2, 3].

Вдале формування портфеля інвестицій постає важливим для мінімізації ризиків та максимізації прибутку [4]. З огляду на складність ринку і кількість факторів, що впливають на дохідність ОВДП, стає очевидною потреба у передбаченні майбутніх змін цін та доходностей.

Серед існуючих методів прогнозування ринкових тенденцій особливої уваги заслуговують методи, що базуються на машинному навчанні. Використання алгоритмів машинного навчання пропонує широкі можливості для аналізу історичних даних та побудови моделей, здатних ефективно прогнозувати поведінку ринку ОВДП. Завдяки таким підходам, інвестори можуть приймати більш обґрунтовані рішення щодо управління своїм портфелем, підвищуючи його стабільність і прибутковість у довгостроковій перспективі. Таким чином, для підвищення точності прогнозування ринкових тенденцій, буде вдосконалено існуючі алгоритми машинного навчання. Отже, тема дослідження є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення точності прогнозування тенденцій змін ринкових цін ОВДП на основі машинного навчання за рахунок додавання до вибірки навчальних даних макроекономічних показників.

Основними задачами дослідження є:

- провести аналіз існуючих методів і засобів машинного навчання для прогнозування тенденцій ринкових цін ОВДП;
- вдосконалити методи машинного навчання для прогнозування тенденцій ринкових цін ОВДП за наявності вибірки даних, що включає ринкову вартість ОВДП та макроекономічні показники за певний період часу;
- розробити програмні засоби для побудови вибірки навчальних даних, що включатиме як дані ринкової вартості ОВДП, так і макроекономічні показники;
- розробити програмні компоненти та систему візуалізації на основі запропонованих методів;
- провести експериментальні дослідження розроблених засобів прогнозування вартості ОВДП.

Об'єкт дослідження – процес прогнозування ринкових тенденцій цін ОВДП в комп'ютерних системах.

Предмет дослідження – методи та засоби прогнозування ринкових тенденцій цін ОВДП.

Методи дослідження. У процесі досліджень використовувались: методи математичної статистики для обробки й аналізу навчальних даних; методи машинного навчання для створення прогнозних моделей; методи проектування програмного забезпечення для розробки архітектури системи та програмного коду застосунку; комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Наукова новизна отриманих результатів.

Удосконалено метод машинного навчання для прогнозування цінових тенденцій ОВДП, в якому, на відміну від існуючих методів, було використано дані макроекономічних показників в поєднанні з даними цін ОВДП за певний період часу для навчання, що дозволило підвищити достовірність прогнозу ринкових цін ОВДП.

Отримала подальшого розвитку архітектура програмного застосунку, в яку, на відміну від існуючих, введено модуль побудови навчальної вибірки даних, що дозволило прискорити процес машинного навчання.

Практичне значення отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби прогнозування ринкових цін ОВДП для формування портфелю інвестицій.

Особистий внесок здобувача.

Всі наведені у магістерській кваліфікаційній роботі наукові результати, були отримані автором особисто. Наукову працю опубліковано особисто.

Апробація матеріалів магістерської кваліфікаційної роботи. Основні положення магістерської кваліфікаційної роботи, представлені та обговорені на Міжнародній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)».

Публікації. Основні результати досліджень були опубліковані в одній науковій праці, яка входила у матеріали Міжнародної науково-практичної інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» [1].

1 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ РИНКОВИХ ЦІН ОВДП

1.1 Аналіз сучасних підходів до прогнозування цінових тенденцій ОВДП

З розвитком фінансового ринку України та підвищенням інтересу до державних облігацій, все більше інвесторів звертають увагу на можливості, що відкриваються перед ними. Облігації внутрішньої державної позики (ОВДП) є одним з найнадійніших фінансових інструментів в Україні, оскільки забезпечуються державними гарантіями, що робить їх популярними серед інвесторів, особливо в умовах економічної нестабільності.

Зростаюча доступність та популярність облігацій внутрішньої державної позики значною мірою підвищила потребу українців у інструментах, націлених на полегшення формування портфелю інвестицій. Основною метою таких інструментів є відстеження динаміки цін на ринку цінних паперів задля підвищення прогнозованості та зменшення потенційних ризиків, що призводять до коливання цін [1].

Прогнозування цінових тенденцій на фінансових ринках, зокрема на ринку облігацій внутрішньої державної позики (ОВДП), є важливим завданням для учасників ринку, аналітиків та дослідників, оскільки дає змогу приймати обґрунтовані інвестиційні рішення і керувати потенційними ризиками.

Серед сучасних підходів до прогнозування часто виділяють такі, що базуються на поєднанні традиційних статистичних методів з інноваційними алгоритмами машинного навчання, що дозволяє враховувати складні взаємозв'язки між ринковими та макроекономічними показниками. Це підвищує точність прогнозів і дає змогу ефективніше реагувати на зміни у фінансовому середовищі. При цьому важливу роль відіграють як обрані методи прогнозування, так і вибірка ринкових даних та даних макроекономічних показників, на основі якої базуватимуться ці методи.

Також важливим є той факт, що різні методи обчислення здатні працювати з різними наборами даних, побудова яких вимагає значних зусиль та часу. Отже, для прогнозування ринкових тенденцій необхідно обрати оптимальний метод прогнозування та вибірку даних, з якою цей метод буде ефективним.

Інформаційні технології стали невід'ємним інструментом у сучасному прогнозуванні цінових тенденцій, забезпечуючи швидкий доступ до великих обсягів даних, автоматизацію обробки інформації та підвищення точності прогнозів. Сучасні інвестори стикаються з проблемами аналізу складних економічних факторів, високої швидкості змін на ринку та обмеженого часу на обробку даних, що ускладнює прогнозування та прийняття рішень.

Інформаційні технології, завдяки машинному навчанню та аналізу великих обсягів даних, допомагають враховувати широкий спектр факторів і швидко адаптувати прогнози до змінних умов ринку. При цьому програмні засоби часто націлені на аналіз існуючих даних та їх графічне представлення, спрощуючи щоденну роботу інвесторів.

Програмні засоби для інвестування мають низку особливостей, які роблять їх корисними інструментами для інвесторів різного рівня досвіду. Такі технології надають доступ до аналітики та ринкових даних у реальному часі, що дозволяє оперативно реагувати на зміни на ринку. Багато з них мають вбудовані інструменти технічного та фундаментального аналізу, зокрема графіки, індикатори, а також рекомендації аналітиків, що спрощує прийняття рішень.

Водночас використання таких програмних засобів потребує розуміння їхньої функціональності та можливих ризиків, таких як збої в роботі, кібератаки чи помилки в алгоритмах, що може вплинути на точність прогнозів та призвести до втрат. В той час як інвестиційне програмне забезпечення (ІПЗ) дозволяє відстежувати динаміку ринків та швидко реагувати на її зміни, наявні ризики можуть негативно вплинути на якість прогнозів і призвести до фінансових втрат.

Отже, рішенням перелічених проблем може бути розробка програмного

забезпечення, націленого на збір макроекономічних та ринкових даних та застосування оптимального алгоритму для прогнозування цінових тенденцій. Також програмне забезпечення для прогнозування динаміки ринкових цін ОВДП повинно враховувати наведені особливості, забезпечуючи інвесторам зручність і функціональність у прийнятті рішень. Програмна система, що розробляється, повинна автоматизувати процеси збору та аналізу даних, з метою прогнозування динаміки цін ОВДП на ринку.

1.2 Аналіз методів прогнозування цінових тенденцій ОВДП

Методи прогнозування цінових тенденцій ОВДП варіюються від традиційних економетричних моделей, що дозволяють виявити нові, не відомі раніше зв'язки, уточнити чи спростувати гіпотези про існування визначених зв'язків між економічними показниками [5], до сучасних підходів на основі машинного навчання, що дозволяє виявляти закономірності, передбачати поведінку та прогнозувати майбутні події [6]. Кожен із методів має свої переваги, недоліки та особливості застосування, тому для досягнення максимальної точності часто застосовують комбіновані підходи.

Першими слід згадати економетричні моделі, які довгий час були основними інструментами в аналізі цінових тенденцій ОВДП. Серед них можна виділити лінійну регресію, авторегресивно-інтегровану модель ковзного середнього (ARIMA) та векторну авторегресію (VAR).

Лінійна регресія дозволяє знаходити залежності між залежними та незалежними змінними, у випадку прогнозування цін ОВДП – між ціновими показниками та певними макроекономічними факторами, такими як рівень інфляції, облікова ставка та курс національної валюти. Лінійна регресія – один із базових методів, що дозволяє моделювати лінійні залежності між ціновими показниками та впливовими факторами, такими як процентні ставки, рівень інфляції, економічне зростання тощо [7].

Моделі ARIMA та VAR, у свою чергу, враховують тимчасову структуру даних, що дозволяє краще прогнозувати динаміку цінових змін на основі минулих значень. Проте, основним обмеженням економетричних моделей є їхня чутливість до вибору змінних і припущень про лінійні зв'язки між ними, що може призвести до неточностей у прогнозуванні в умовах високої волатильності ринку [8, 9].

З розвитком технологій аналізу даних все більшої популярності набувають методи машинного навчання, які здатні знаходити складні, нелінійні зв'язки між різними факторами. Серед таких методів можна виділити машинне навчання та методи на основі дерев рішень – зокрема, алгоритми випадковий ліс (Random Forest) і градієнтний бустинг (Gradient Boosting Trees), що добре працюють з великим обсягом даних та складними взаємозв'язками між змінними [10, 11]. Ці моделі добре підходять для обробки великих масивів даних і здатні самостійно визначати найбільш значущі фактори для прогнозування. Наприклад, нейронні мережі можуть ідентифікувати патерни в історичних даних цін ОВДП, враховуючи макроекономічні індикатори, глобальні ринкові тренди та інші чинники, що сприяє більш точному прогнозу. Основним недоліком методів машинного навчання є їхня вимога до обсягу даних для навчання та високих обчислювальних ресурсів, а також складність інтерпретації отриманих результатів [12, 13].

Важливо також згадати комбіновані методи прогнозування, які поєднують переваги економетричних моделей і машинного навчання. Наприклад, можна використовувати ARIMA для прогнозування трендів на короткостроковий період, а нейронні мережі – для довгострокового прогнозу з урахуванням нелінійних залежностей. Такий підхід дозволяє знизити похибки та покращити точність прогнозів, адже він об'єднує сильні сторони обох методів [14].

Варто зазначити, що прогнозування цінових тенденцій ОВДП стикається з низкою викликів та ризиків. Зокрема, ціни ОВДП можуть змінюватися під впливом непередбачуваних подій, таких як економічні кризи, зміни у грошово-кредитній політиці чи політична нестабільність.

Таким чином, кожен із методів прогнозування цінових тенденцій ОВДП має свої переваги і недоліки, тому вибір конкретного підходу залежить від завдань, обсягів наявних даних та поточних умов ринку. Дана робота фокусується саме на використанні методів машинного навчання, оскільки усунення їх недоліків можливе завдяки побудові широкої вибірки навчальних даних та спрощенню кінцевих результатів для простішого аналізу кінцевим користувачем.

1.3 Аналітичний огляд програмних засобів прогнозування цінових тенденцій ОВДП

На ринку існує широкий вибір інструментів для формування портфеля інвестицій в ОВДП, серед яких найбільш популярними є iBonds, BondIT, Bloomberg Terminal, ОTR Bank ОВДП та Універ Капітал. Ці платформи використовують різні підходи до управління інвестиціями, забезпечуючи користувачів аналітичними інструментами, автоматизованими системами та інтерфейсами для відстеження прибутковості облігацій і ризиків.

Ці платформи постійно вдосконалюються, адаптуючи новітні технології для покращення користувацького досвіду, що дозволяє інвесторам ефективніше управляти своїми портфелями ОВДП. Тому важливо щоб програмне забезпечення для формування портфеля інвестицій відповідало сучасним тенденціям та могло конкурувати з нововведеннями та покращеннями у сфері.

Платформа BondIT — це алгоритмічна платформа, яка використовує штучний інтелект (ШІ) та машинне навчання для підтримки в прийнятті рішень на ринку облігацій при формуванні інвестиційного портфеля, надаючи рекомендації щодо інвестицій. Платформа пропонує інструменти для інвесторів та фінансових аналітиків та забезпечує високу швидкість пошуку, аналізу та оптимізації портфелів облігацій. Вона забезпечує автоматизований аналіз великих обсягів даних, включаючи оцінку ризиків, прогнози прибутковості та оптимізацію інвестиційних стратегій. Приклад інтерфейсу зображений на рис. 1.1. Серед переваг зазначеної

платформи виділяють потужні аналітичні інструменти, автоматизація процесу підбору активів і рекомендації, основані на AI, що дозволяє ефективно управляти інвестиціями. Проте платформу супроводжують недоліки, зокрема висока вартість для користувачів з невеликими портфелями та значна складність інтерфейсу, що може стати перепорою у використанні для новачків [15].

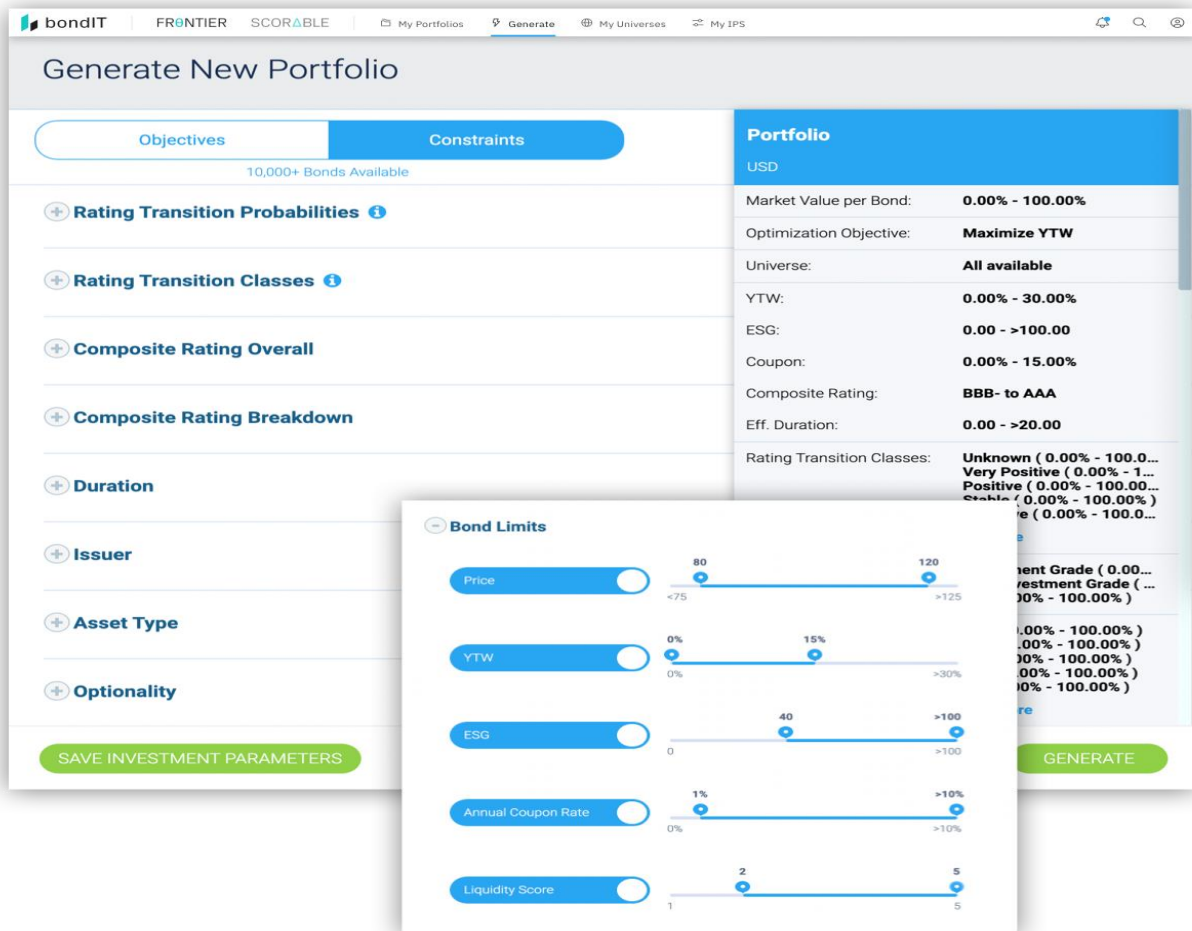


Рисунок 1.1 – Інтерфейс платформи BondIT

Програмне забезпечення Bloomberg Terminal — це глобальна платформа для управління фінансовими активами, включно з облігаціями внутрішньої державної позики (ОВДП), яка надає широкий спектр фінансових даних і аналітики. Серед її основних переваг варто відзначити найповніший набір фінансової інформації, розширену аналітику та рекомендації щодо інвестицій на основі складних моделей,

що робить її незамінною для професіоналів у сфері фінансів. Однак висока вартість користування та складність освоєння для непрофесіоналів є суттєвими недоліками, які можуть обмежити доступність платформи для широкого кола інвесторів [16]. Приклад інтерфейсу зображений на рис. 1.2.

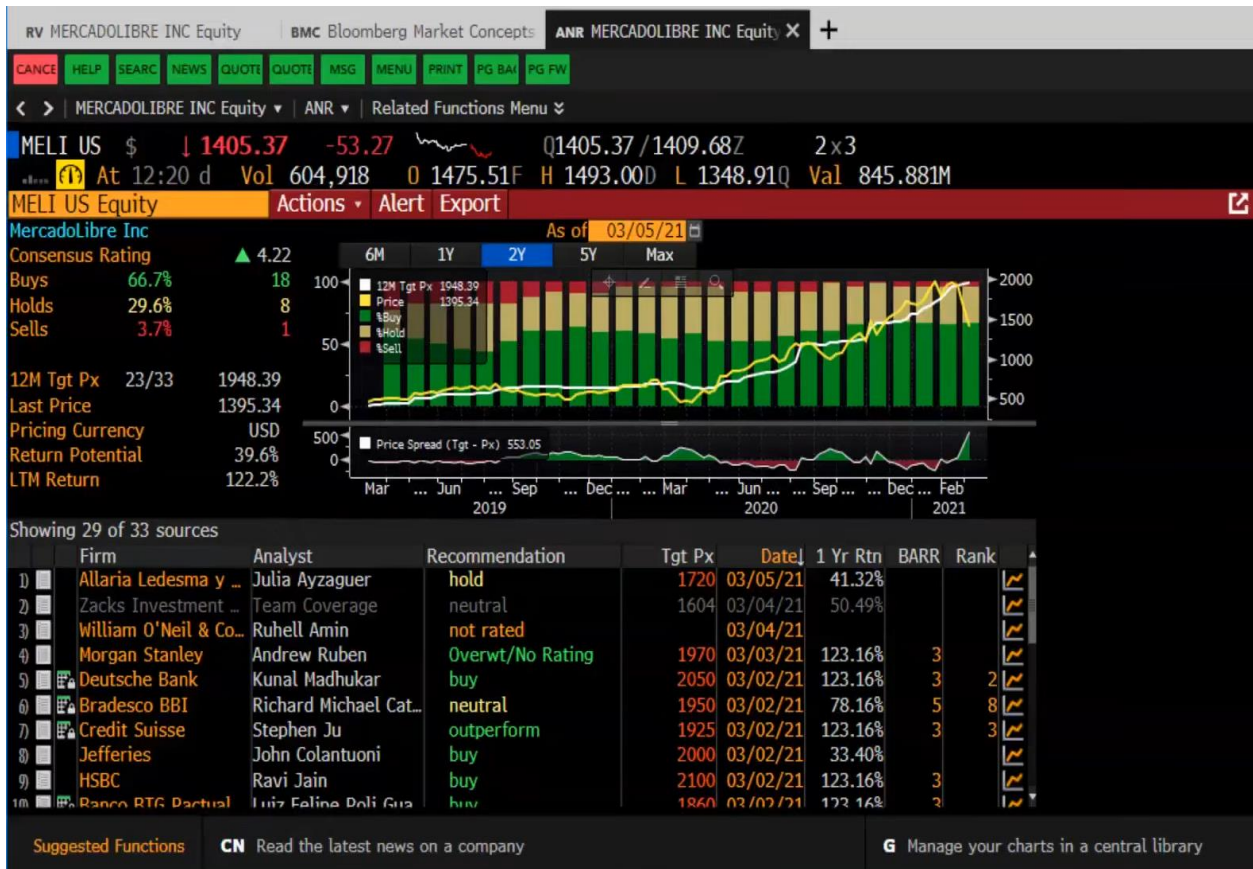
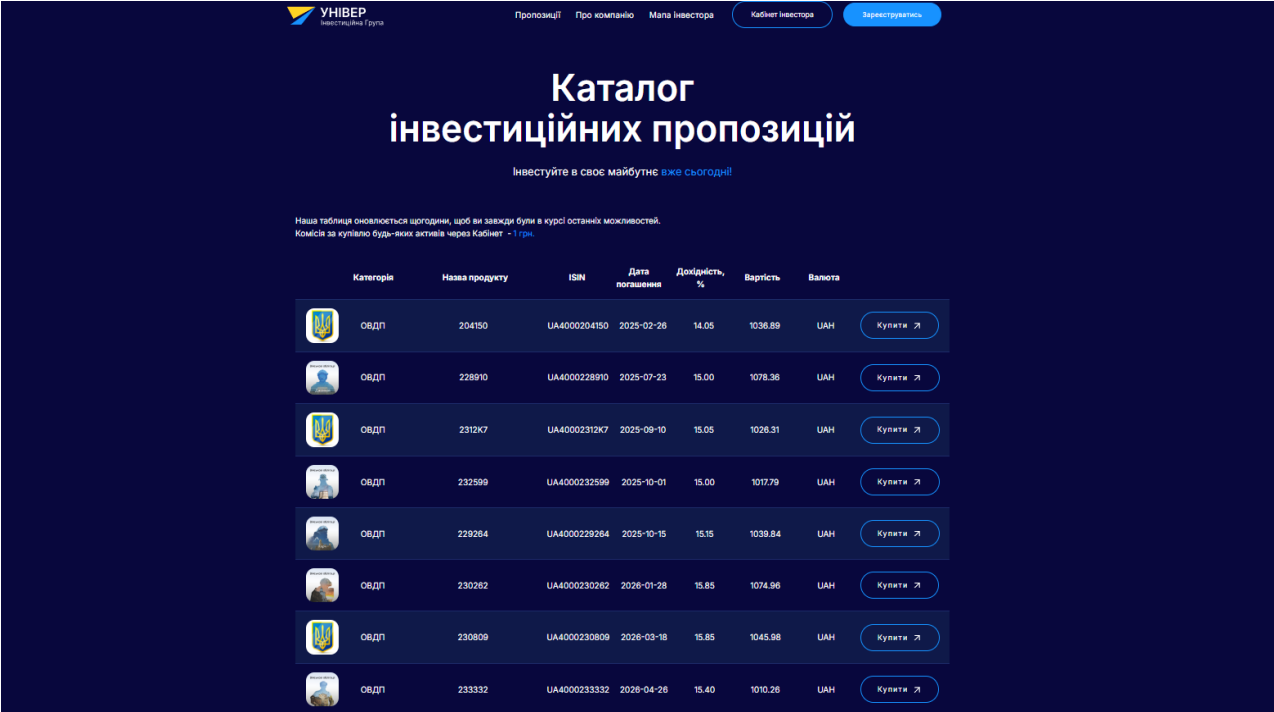


Рисунок 1.2 – Інтерфейс програмного забезпечення Bloomberg Terminal

Платформа ОTR Bank ОВДП — це застосунок для інвестицій в облігації внутрішньої державної позики, який дозволяє користувачам купувати облігації безпосередньо через свій смартфон, моніторити ринкову вартість та управляти своїм портфелем. Серед переваг виділяється простий і інтуїтивно зрозумілий інтерфейс, що забезпечує зручність управління інвестиціями в будь-який час і в будь-якому місці. Проте додаток має й недоліки, зокрема обмежені можливості для аналізу ринку та прогнозування ризиків, що може ускладнити прийняття

обґрунтованих інвестиційних рішень [17].

Універ Капітал — це українська платформа для інвестицій в державні та корпоративні облігації, яка забезпечує доступ до актуальної інформації про ринок облігацій внутрішньої державної позики (ОВДП). Приклад інтерфейсу зображений на рис. 1.3. Серед основних переваг платформи можна відзначити локалізовану підтримку, орієнтацію на український ринок та можливість отримання консультацій, що робить її привабливою для місцевих інвесторів. Однак одним із недоліків є менш потужні аналітичні інструменти в порівнянні з міжнародними платформами, що може обмежити можливості користувачів у глибокому аналізі ринкових тенденцій та ризиків [18].



Категорія	Назва продукту	ISIN	Дата погашення	Доходність, %	Вартість	Валюта	
ОВДП	204150	UA4000204150	2025-02-26	14.05	1036.89	UAH	Купити
ОВДП	228910	UA4000228910	2025-07-23	15.00	1078.36	UAH	Купити
ОВДП	2312K7	UA40002312K7	2025-09-10	15.05	1026.31	UAH	Купити
ОВДП	232599	UA4000232599	2025-10-01	15.00	1017.79	UAH	Купити
ОВДП	229264	UA4000229264	2025-10-15	15.15	1039.84	UAH	Купити
ОВДП	230262	UA4000230262	2026-01-28	15.85	1074.96	UAH	Купити
ОВДП	230809	UA4000230809	2026-03-18	15.85	1045.88	UAH	Купити
ОВДП	233332	UA4000233332	2026-04-26	15.40	1010.26	UAH	Купити

Рисунок 1.3 – Інтерфейс програмного забезпечення Універ Капітал

Проаналізувавши існуючі програмні засоби, було представлено порівняльну таблицю 1.1, що демонструє ключові аспекти різних інструментів для формування портфельів інвестицій в облігації внутрішньої державної позики (ОВДП). В

порівняльній таблиці наведено переваги та недоліки кожного з рішень, їхні функціональні можливості, точність аналізу ризиків, доступ до фінансових даних та інтеграційні можливості з іншими фінансовими сервісами.

Таблиця 2.1 – Порівняння програмних засобів для формування портфеля інвестицій в ОВДП

Назва платформи	Простота використання	Аналітичні інструменти	Вартість	Автоматизація	Мультиплатформність
iBonds	Висока	Обмежені	Середня	Ні	Так
BondIT	Середня	Розширені	Висока	Так	Так
Bloomberg Terminal	Низька	Найпотужніші	Дуже висока	Так	Так
ОТР Bank ОВДП	Висока	Обмежені	Низька	Ні	Так
Універ Капітал	Висока	Обмежені	Низька	Ні	Ні

Відповідно до наведених даних, BondIT та Bloomberg Terminal пропонують найбільш потужні аналітичні інструменти, використовуючи алгоритми і моделі прогнозування ринкових коливань. У той час як ОТР Bank ОВДП і Універ Капітал більше орієнтовані на користувачів, яким потрібен базовий доступ до ринку. Сервіси iBonds і ОТР Bank ОВДП вирізняються інтуїтивно зрозумілим інтерфейсом, що робить їх актуальними для новачків. BondIT і Bloomberg Terminal,

хоча і надають більш глибокий функціонал, є складнішими в освоєнні. Найдорожчим рішенням є Bloomberg Terminal, що орієнтоване на професійних інвесторів. У той час як ОTR Bank ОВДП та Універ Капітал надають більш доступні варіанти для локального ринку.

На основі проведеного аналізу, конкурентоспроможне програмне забезпечення для формування портфеля інвестицій в ОВДП може відповідати наступним характеристикам:

- простота у використанні;
- автоматизація процесів;
- прогнозування цінових тенденцій ринку;
- мультиплатформність;
- наявність аналітичних інструментів;
- доступність.

Отже, було проведено аналіз сучасних програмних засобів для формування портфеля інвестицій. Існуючі програмні засоби для формування портфеля інвестицій в ОВДП пропонують широкий спектр функцій, проте мають свої переваги та недоліки. Нова система має забезпечити поєднання простоти, автоматизації, прогнозування цінових тенденцій та якісної аналітики, що дозволить інвесторам ефективніше управляти своїм портфелем.

1.4 Аналіз джерел навчальних даних для прогнозування цінових тенденцій ОВДП

У сучасному фінансовому середовищі прогнозування цінових тенденцій облігацій, зокрема облігацій внутрішньої державної позики, стає дедалі складнішим завданням, яке вимагає застосування передових методів аналізу, таких як штучний інтелект та машинне навчання. Проте успіх прогнозних моделей значною мірою залежить від якості навчальних даних, які використовуються для їхнього навчання.

Навчальні дані – це початковий вибірка даних, яка використовується для навчання алгоритму або моделі машинного навчання чи комп'ютерного зору для обробки інформації [19].

Якісні дані не лише збільшують точність прогнозів, а й забезпечують більш глибоке розуміння динаміки цін на ринку облігацій. Відсутність якісних навчальних даних призводить до того, що навіть найефективніші алгоритми не здатні надати коректний результат. Дійсним є твердження, що надійні моделі машинного навчання можуть надавати хибні результати, якщо для їх навчання було використано некоректні, неточні або нерелевантні дані [20].

Машинне навчання, яке застосовується для фінансового прогнозування, здебільшого базується на великих обсягах даних про попередню цінову динаміку, макроекономічних показниках, даних про попит і пропозицію на ринку та інших факторах, що впливають на ціни ОВДП.

Навчальні дані допомагають моделям розпізнавати шаблони та тренди, що виникають на ринку, забезпечуючи основу для прогнозування. Якщо дані містять значні похибки чи не відображають реальної ринкової динаміки, натренована модель буде не здатна точно розпізнати закономірності, що веде до помилкових прогнозів.

Якісні навчальні дані повинні бути точними, повними та надійними [21]. Точність даних особливо важлива у випадку з ОВДП, де навіть невеликі коливання можуть впливати на загальну дохідність та ризики для інвесторів. Помилки в даних призводять до хибних висновків, що може спонукати інвесторів до прийняття невірних рішень, які здатні призвести до фінансових втрат. Крім того, для моделювання ОВДП потрібно враховувати специфічні економічні показники, включаючи процентні ставки, інфляцію, валютні курси тощо. Тому важливо забезпечити достовірність цих даних, аби модель могла реагувати на зміни, враховуючи реальні макроекономічні умови.

Сучасний ринок облігацій є динамічним, через що ціни на ОВДП можуть швидко змінюватися залежно від внутрішньої та зовнішньої політичної ситуації, економічних реформ, коливань попиту та пропозиції. Для відповідності прогнозів моделі реальній ситуації її навчальні дані повинні бути актуальними на момент побудови прогнозів. Оновлення навчальних даних дозволяє моделі залишатися релевантною та забезпечує вищу точність прогнозів [22].

Крім того, для прогнозування цінових тенденцій ОВДП важливим є забезпечення повноти та різноманітності даних. Навчальні дані повинні мати достатній набір значущих характеристик, що можуть вплинути на ціну облігацій. Поєднання різних джерел даних та інформації дозволяє моделі виявляти складні взаємозв'язки та робити більш точні прогнозування.

Окрім зазначеного, ще одним важливим аспектом у формуванні тренувальної вибірки даних є уникнення надмірного навчання моделі, оскільки надмірне навчання моделі може призвести до врахування нею випадкових коливань як значущих закономірностей. Така модель може виявитися менш ефективною при прогнозуванні наступних тенденцій [23]. Це особливо важливо у випадку з фінансовими ринками, де вплив зовнішніх факторів може змінюватися з часом. Отже, вибірка навчальних даних повинна включати мінімальний набір характеристик, що впливають на зміни ціни ОВДП.

Враховуючи описані особливості формування навчальної вибірки даних, можна виділити наступні вимоги до навчальних даних для прогнозування цінових тенденцій ОВДП:

- точність та надійність;
- актуальність;
- різноманітність;
- мінімально повний обсяг даних.

Оскільки макроекономічні показники значною мірою впливають на формування цін облігацій, саме на пошуку їх закономірностей базуватиметься прогнозування цінових тенденцій ОВДП.

Відповідно до описаних характеристик, можна виділити ключові навчальні дані, що формуватимуть мінімально повну вибірку за певний проміжок часу. До такої вибірки можна віднести наступні показники:

- цінові показники конкретних облігацій;
- характеристики конкретних облігацій;
- валютні курси гривні;
- облікові ставки;
- індекс споживчих цін.

Отже, було сформовано вимоги до навчальної вибірки даних для прогнозування цінових тенденцій ОВДП. Для підвищення точності прогнозування дані вибірки повинні бути точними та надійними, актуальними, різноманітними та складати мінімально повний набір. При цьому слід врахувати як характеристики самих облігацій, так і макроекономічні показники, такі як валютні курси, облікові ставки та індекс споживчих цін.

Як вже було зазначено, для формування мінімально повної навчальної вибірки має бути використано точні, надійні, різноманітні та актуальні дані. При цьому вибірка даних включатиме як характеристики цінних паперів, так і макроекономічні показники за певний період часу. Для забезпечення наведених якісних характеристик даних, має бути використано надійні джерела. Оскільки навчальні дані відповідатимуть певному часовому проміжку та складатимуть значний об'єм, джерела даних повинні надавати можливості автоматизації процесу завантаження даних за допомогою програмних інтерфейсів застосунку (API).

Враховуючи описані особливості, для отримання макроекономічних показників може бути використано API для розробників, надане Національним банком України та API популярних банків.

Інтерфейси API Національного банку України дозволяють отримувати широкий перелік макроекономічних показників, серед яких було обрано:

- офіційний курс гривні до іноземних валют;
- щоденні дані про оголошені процентні ставки за операціями Національного банку;
- результати розміщення облігацій внутрішніх державних позик;
- інформація про цінні папери, включно з характеристиками випущених ОВДП;
- індекси споживчих цін.

Перелічені дані можуть бути отримані шляхом виконання GET-запитів за протоколом HTTP у форматі JSON або XML. Що не менш важливо, Національний банк України надає можливість отримувати ці дані на конкретний момент часу [24].

Окремо варто виділити важливість отримання ринкового курсу гривні, який більш точно відображатиме тенденції змін на валютному ринку, аніж державний. Для цього може бути використано API популярних банків, що працюють на території України. Прикладом може слугувати API ПриватБанк для розробників. Наведене API надає можливості отримання валютних курсів на конкретний момент часу за допомогою протоколу HTTP. Інтерфейс також повертає дані в форматі JSON, що дозволяє зручно автоматизувати процес одержання потрібних даних.

Також для формування навчальної вибірки необхідним є збір даних ринкових цінових змін для конкретних цінних паперів. Для побудови такої вибірки може бути обрано дані одного або декількох брокерів. При цьому виникає складність виконання цієї задачі програмно, оскільки у відкритому доступі може бути важко знайти API надійного брокера. Рішенням може бути формування такої вибірки шляхом спостережень та збереження цінових показників протягом певного проміжку часу.

Отже, для формування мінімально повної навчальної вибірки, що включатиме точні, надійні, різноманітні та актуальні дані за певний проміжок часу, може бути

використано API для розробників. API, що надаються Національним банком України та API ПриватБанк може бути використано для отримання даних макроекономічних показників. Вибірку даних цінових тенденцій може бути сформовано шляхом спостережень та збереження цін конкретного брокера протягом певного періоду.

1.5 Формування вимог до програмного забезпечення для прогнозування ринкових тенденцій цін ОВДП

Беручи до уваги актуальність та затребуваність програмного забезпечення для формування портфеля інвестицій ОВДП, а також наведені особливості методів прогнозування цінових тенденцій ОВДП, переваги та недоліки існуючих програмних рішень для формування портфеля інвестицій, може бути сформовано вимоги до системи, що розробляється.

Конкурентоспроможне програмне забезпечення для формування портфеля інвестицій в ОВДП повинне відповідати наступним характеристикам:

- простота у використанні;
- автоматизація процесів;
- прогнозування цінових тенденцій ринку;
- мультиплатформність;
- наявність аналітичних інструментів;
- доступність.

З метою підвищення точності прогнозування, система, що розробляється, використовуватиме методи машинного навчання в поєднанні з оптимальною вибіркою макроекономічних показників та цінових характеристик ОВДП.

Необхідною є автоматизація процесів формування мінімально повної навчальної вибірки, що включає точні, надійні, різноманітні та актуальні дані за певний період часу. До таких даних слід віднести:

- цінові показники конкретних облігацій;

- характеристики конкретних облігацій;
- валютні курси гривні;
- облікові ставки;
- індекс споживчих цін.

Отже, було сформовано вимоги до програмного забезпечення для прогнозування цінової динаміки ОВДП. Система, що розробляється, має забезпечити поєднання простоти, автоматизації, прогнозування цінових тенденцій та якісної аналітики, що дозволить інвесторам ефективніше управляти своїм портфелем.

1.6 Висновки

Було розглянуто роль інформаційних технологій у формуванні портфеля інвестицій, зокрема роль машинного навчання у прогнозуванні цінових тенденцій ОВДП. Рішенням існуючих проблем прогнозування цін може бути розробка програмного забезпечення, націленого на збір макроекономічних та ринкових даних та застосування оптимального алгоритму для прогнозування цінових тенденцій.

Було розглянуто переваги та недоліки методів машинного навчання для прогнозування ринкових цін ОВДП.

Було проведено аналіз сучасних програмних засобів для формування портфеля інвестицій. Існуючі програмні засоби для формування портфеля інвестицій в ОВДП пропонують широкий спектр функцій, проте мають свої переваги та недоліки.

Крім того, було сформовано вимоги до навчальної вибірки даних для прогнозування цінових тенденцій ОВДП.

Для формування навчальної вибірки може бути використано API для розробників. API, що надаються Національним банком України та API ПриватБанк може бути використано для отримання даних макроекономічних показників.

Вибірку даних цінових тенденцій може бути сформовано шляхом спостережень та збереження цін конкретного брокера протягом певного періоду.

Також було сформовано вимоги до програмного забезпечення для прогнозування цінової динаміки ОВДП. Система, що розробляється, має забезпечити поєднання простоти, автоматизації, прогнозування цінових тенденцій та якісної аналітики, що дозволить інвесторам ефективніше управляти своїм портфелем.

2 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1 Прогнозування ринкових тенденцій цін ОВДП

Прогнозування цінової динаміки облігацій внутрішньої державної позики є важливою складовою ефективного управління інвестиційним портфелем, що потребує застосування сучасних підходів та інструментів аналізу даних.

Алгоритм має використовувати навчальні дані для створення моделей, аналізуючи взаємозв'язки між різними ознаками (наприклад, між обліковими ставками та цінами ОВДП). Під час навчання модель має визначати, які з ознак є найбільш важливими для прогнозування цінових тенденцій, і будувати на їх основі складні рішення.

Модель прогнозування повинна бути стійкою до перенавчання, що є особливо важливим для фінансових ринків, де наявність шуму і змінність даних може вплинути на точність результатів. Вона повинна мати здатність обробляти складні, нелінійні взаємозв'язки в даних, що відповідає характеру ринкових процесів ОВДП. Крім того, модель повинна автоматично оцінювати важливість ознак, що дозволяє аналізувати вплив різних факторів на зміну цін. Підготовка даних для навчання не повинна вимагати значних витрат часу на попередню обробку, оскільки ефективність алгоритму залежить від здатності працювати з неповними даними та з різними форматами інформації.

Для побудови прогностичних моделей необхідна система, яка дозволяє працювати з даними, що включають числові та категоріальні ознаки. Важливо, щоб алгоритм міг ефективно обробляти великі обсяги даних і забезпечувати надійні результати без значних обчислювальних витрат. Всі ці аспекти створюють основу для побудови високоякісної моделі прогнозування цін ОВДП.

Отже, для прогнозування цінової динаміки ОВДП необхідно враховувати довгострокові тенденції ринку та мати здатність обробляти складні взаємозв'язки

та неповні дані, забезпечуючи точність і стійкість прогнозів. Ефективна система повинна автоматично оцінювати важливість ознак, бути стійкою до перенавчання та обробляти великі обсяги даних з мінімальними витратами часу та ресурсів.

2.2 Оптимізація вибірки навчальних даних

Програмні інтерфейси для розробників, що надаються банками, а також API Національного банку України надають можливості отримувати актуальну інформацію про макроекономічні показники та цінні папери. Проте, дані, що надаються публічними API часто мають значну кількість деталей та є представленими багатовимірними об'єктами. Прикладом можуть слугувати дані процентних ставок, які надаються API Національного банку України, наведені на рис. 2.1.

Проаналізувавши наведений приклад, можна зробити висновок, що такі дані є надмірними та складаються зі складних об'єктів, що може призвести до збільшення складності обробки даних та помилкових результатів в роботі моделі. При цьому серед наведених даних лише облікова ставка Національного банку України є необхідною для прогнозування цінових тенденцій ОВДП, а саме її відсоткове значення.

Описана проблематика складності та надмірності даних є актуальною й для решти API – усі згадані раніше способи отримання навчальної вибірки надають надмірні дані. Враховуючи важливість побудови повної та мінімальної вибірки даних, що відповідатимуть вимогам якості, необхідною є попередня обробка кожного типу даних, що призведе до спрощення об'єктів та видалення нерелевантної інформації з результуючої вибірки.

Така обробка включатиме наступні кроки:

- обчислення та спрощення значень індексів споживчих цін;
- отримання значення облікової ставки зі складного об'єкту;
- отримання значень курсу валют зі складного об'єкту;

- завантаження даних ОВДП;
- побудова кортежа навчальних даних.

```

▼ KEY_RateOvernightCD : Object
  dt : "20230703"
  txt : "Процентна ставка за депозитними сертифікатами овернайт Національного б..."
  txten : "Rate on overnight certificates of deposit of the NBU"
  id_api : "KEY_RateOvernightCD"
  level_i : 1
  parent : "Key_Rates_NBU"
  freq : "D"
  tzep : "Rate on overnight certificates of deposit of the NBU"
  value : 20
▼ KEY_RateOvernightLoans : Object
  dt : "20230703"
  txt : "Процентна ставка за кредитами овернайт Національного банку України"
  txten : "Rate on overnight loans of the NBU"
  id_api : "KEY_RateOvernightLoans"
  level_i : 1
  parent : "Key_Rates_NBU"
  freq : "D"
  tzep : "Rate on overnight loans of the NBU"
  value : 27
▼ KEY_PolicyRate : Object
  dt : "20230703"
  txt : "Облікова ставка Національного банку України "
  txten : "NBU key policy rate "
  id_api : "KEY_PolicyRate"
  level_i : 1
  parent : "Key_Rates_NBU"
  freq : "D"
  tzep : "NBU key policy rate "
  value : 25

```

Рисунок 2.1 – Дані процентних ставок Національного банку України

Описану логіку може бути виконано як на етапі збереження даних до БД, так і під час формування навчальної вибірки безпосередньо перед початком процесу навчання. Для досягнення поставленої мети буде послідовно виконано обробку кожного об'єкту даних у вибірці та поєднано отримані корисні дані за датою, що зберігається як властивість об'єкту. Описаний алгоритм попередньої обробки даних наведено на рис. 2.2.



Рисунок 2.2 – UML-діаграма діяльності побудови навчальної вибірки

Результатом виконання алгоритму побудови навчальної вибірки є одновимірний масив об'єктів, що відображають поєднання макроекономічних показників та цінових характеристик ОВДП за певний проміжок часу, де кожен

об'єкт зберігає дані, що відповідають конкретній даті. Схематичне зображення об'єкту вибірки навчальних даних наведено на рис. 2.3.

Дата
Індекс споживчих цін
Курс купівлі долару США
Курс продажу долару США
Облікова ставка НБУ
Курс купівлі ОВДП 1
Курс продажу ОВДП 1
...
Курс купівлі ОВДП n
Курс продажу ОВДП n

Рисунок 2.3 – Складові об'єкту навчальної вибірки

Отже, було виконано оптимізацію навчальних даних. Було розроблено алгоритм побудови вибірки навчальних даних. Розроблений алгоритм виконує спрощення навчальних даних задля забезпечення моделі машинного навчання мінімальною та повною вибіркою даних.

2.3 Вибір методу машинного навчання

Машинне навчання є інструментом інформаційних технологій, який використовується для аналізу великих обсягів даних та розв'язання складних задач,

що базуються на пошуку закономірностей, та може бути застосоване у сфері прогнозування цінових тенденцій ОВДП. Вибір оптимального методу машинного навчання є ключовим етапом розробки моделей, оскільки від нього залежать точність, ефективність і швидкість їх роботи. Вдалий підхід до вибору методу дозволяє отримати результати високої якості, оптимально використовуючи наявні ресурси і дані.

В ході аналізу основних методів машинного навчання було зосереджено увагу на таких методах як Random Forest, Gradient Boosting, Vector Autoregressive Model, ARIMA, лінійна регресія

Метод машинного навчання Random Forest — це ансамблевий метод машинного навчання, який застосовується для задач класифікації і регресії. Він побудований на основі дерев рішень, які працюють незалежно один від одного, а результат визначається голосуванням (для класифікації) або усередненням (для регресії). Основні переваги методу включають його стійкість до перенавчання завдяки усередненню результатів численних дерев і здатність обробляти складні, нелінійні взаємозв'язки у великих наборах даних. Також слід зазначити особливості моделі при роботі з даними: Random Forest може автоматично оцінювати важливість кожної ознаки при визначенні залежностей, може працювати як з числовими, так і з категоріальними даними, а також з даними, що містять пропущені значення. При цьому модель не потребує складної попередньої обробки даних та не вимагає значних зусиль для налаштування. Однак, модель потребує значних обчислювальних ресурсів і може бути складною для інтерпретації. Вона підходить для завдань, де дані мають складні нелінійні взаємозв'язки.

Алгоритм навчання Random Forest застосовує загальну техніку bootstrap-агрегування, або пакування, в ході якої для навчальної множини $X = x_1, \dots, x_n$ з відповідями $Y = y_1, \dots, y_n$ багаторазово (B разів) обирається випадкова вибірка із заміною навчальної множини та підбираються дерева до цих вибірок: для $b = 1, \dots, B$ з виконанням наступних кроків:

1. Вибрати, з заміною, n навчальних прикладів з X, Y ; назовемо їх X_b, Y_b .
2. Виконати навчання дерева класифікації або регресії f_b на X_b, Y_b .

Після навчання прогнози для невидимих вибірок x' можуть бути зроблені за допомогою множинного голосування у випадку з деревами класифікації або шляхом усереднення прогнозів з усіх окремих дерев регресії на x' :

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (2.1)$$

де:

$\hat{f}$ — результат прогнозування;

B — кількість дерев у лісі;

$f_b(x')$ — прогноз i -го дерева для вхідних даних x .

Описана процедура бутстрапінгу призводить до кращої роботи моделі, за рахунок зменшення дисперсії моделі при відсутності збільшення зміщення. Як наслідок, в той час, як прогнози одного дерева дуже чутливі до шуму в навчальній вибірці, середнє значення для багатьох дерев не є таким, якщо дерева не корельовані між собою. Просте навчання багатьох дерев на одній навчальній вибірці призведе до сильної кореляції дерев (або навіть одного дерева, якщо алгоритм навчання детермінований); бутстрап вибірка - це спосіб усунення кореляції між деревами, показуючи їм різні навчальні вибірки.

Крім того, оцінка невизначеності прогнозу може бути обчислена як середньоквадратичне відхилення прогнозів від усіх окремих дерев регресії на x' :

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (f_b(x') - \hat{f})^2}{B-1}} \quad (2.2)$$

де:

B — кількість дерев у лісі;

$f_b(x')$ — прогноз i -го дерева для вхідних даних x ;

$\hat{f}$ — результат прогнозування.

Кількість вибірок B є вільним параметром. Розповсюдження є використання від кількох сотень до кількох тисяч дерев, залежно від розміру та характеру навчальної вибірки [25].

Приклад візуалізації роботи методу Random Forest наведено на рис. 2.4.

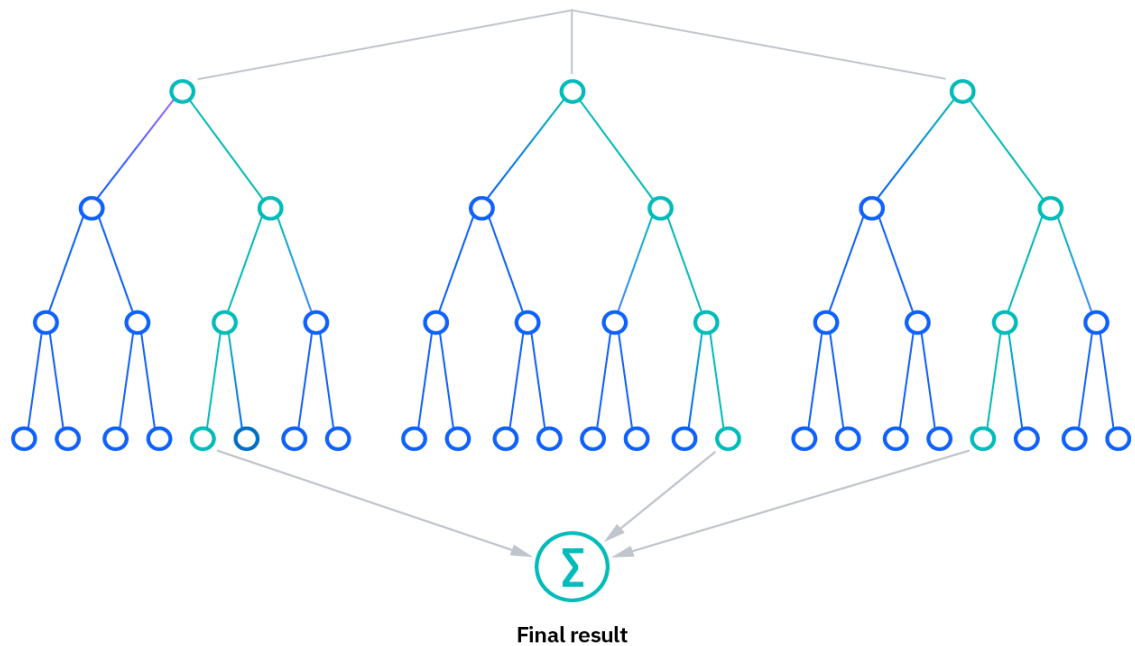


Рисунок 2.4 – Візуалізація методу Random Forest

Метод Gradient Boosting — ансамблевий метод, що базується на деревоподібних моделях і використовується для задач класифікації та регресії. Цей метод базується на послідовності дерев, де кожне дерево спрямоване на виправлення помилок попереднього. Вага кожного дерева визначається його здатністю зменшити похибку. Gradient Boosting відомий високою точністю, особливо для задач з нелінійними даними, і є вдалим рішенням задач зі складними взаємозв'язками. Водночас цей метод потребує багато обчислювальних ресурсів і може бути схильним до перенавчання, за відсутності контролю кількості дерев та

їхньої глибини. Gradient Boosting часто застосовується в задачах, де пріоритет надається точності над швидкістю.

Принцип прогнозування методу Gradient Boosting полягає у мінімізації середньоквадратичної похибки, що описується рівнянням:

$$L = \frac{1}{n} \sum_{i=0}^n (y_i - \gamma_i)^2 \quad (2.3)$$

де:

i - індекс кортежу навчальних даних навчальної множини розміру n ;

y_i - значення, що спостерігається;

γ_i - значення, отримане в результаті прогнозування [26].

Якщо алгоритм має M етапів, то на кожному етапі m ($1 \leq m \leq M$) для покращення недосконалої моделі F_m (для малих m ця модель може просто передбачати y^i як y^- , середнє значення y), необхідним є додавання деякої нової оцінки, $h_m(x)$, що може бути описане рівнянням

$$F_{m+1}(x_i) = F_n(x_i) + h_m(x_i) = y_i \quad (2.4)$$

або

$$h_m(x_i) = y_i - F_n(x_i) \quad (2.5)$$

де:

$F_m(x)$ — поточна модель на етапі m ;

$h_m(x)$ — нова оцінка, яку додають на етапі m ;

y_i - значення, що спостерігається.

При цьому кожен наступний метод F_{m+1} намагається виправити помилки свого попередника F_m .

Модель Vector Autoregressive Model (VAR) є статистичною моделлю для аналізу багатовимірних часових рядів. VAR використовується для прогнозування

та аналізу взаємозалежних часових рядів, де кожна змінна у системі є функцією попередніх значень усіх змінних. Модель дозволяє ефективно враховувати залежності між кількома часовими рядами. Її перевагою є здатність моделювання складних взаємозв'язків між змінними, однак VAR застосовний тільки для лінійних залежностей і може бути складним для інтерпретації у випадках великої кількості змінних. Основна сфера застосування VAR — це економіка, фінансовий аналіз, макроекономічні прогнози та інші галузі, де важливо вивчати взаємозв'язок між декількома часовими рядами.

$$Y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \epsilon_t \quad (2.6)$$

де:

α — відрізок, константа;

β_1, β_2 до β_p — коефіцієнти лагу Y до порядку p [27].

Модель ARIMA (AutoRegressive Integrated Moving Average) є статистичною моделлю для аналізу одномірних часових рядів. Вона складається з трьох компонентів: авторегресії, інтеграції і ковзного середнього, що дозволяє їй працювати з сезонними та нестационарними рядами. ARIMA здатна ефективно передбачати тренди та сезонні коливання, але потребує перетворення даних у стаціонарний ряд, що може бути складним і трудомістким процесом. Цей метод підходить для прогнозування одномірних часових рядів, таких як фінансові та економічні показники та описується наступним рівнянням [28]:

$$Y_t = \beta + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \dots + \alpha_p y_{t-p} = \beta + \sum_{i=1}^p \alpha_i Y_{t-i} \quad (2.7)$$

,

де:

Y_t — наступне значення часової мітки, яке буде передбачене;

β — інтерсепт або константа;

α_i – коефіцієнт для лагового спостереження;

Y_{t-i} – лагове спостереження;

p – кількість лагових порядків.

Лінійна регресія є одним із найпростіших та найбільш інтерпретованих методів для аналізу регресійних залежностей. Ця модель оцінює лінійні зв'язки між незалежними змінними та залежною змінною, підбираючи пряму, що мінімізує суму квадратів похибок. Лінійна регресія є простою для використання та потребує відносно небагато обчислювальних ресурсів, але її використання є доцільним лише для задач, що мають лінійну залежність між змінними. Обмеження лінійної регресії включають низьку точність у випадках існування нелінійної залежності даних. Лінійна регресія широко застосовується у соціальних науках та економіці [29].

Загальна лінійна регресійна модель має вигляд:

$$Y_t = \beta + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \dots + \alpha_p y_{t-p} = \beta + \sum_{i=1}^p \alpha_i Y_{t-i} \quad (2.8)$$

де:

y - залежна змінна;

x_i - незалежні змінні;

u - випадкова похибка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне сподівання якої дорівнює нулеві.

Згідно з цією моделлю, математичне сподівання залежної змінної є лінійною функцією незалежних змінних:

$$E y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u \quad (2.9)$$

де:

$E y$ — математичне сподівання y у при заданих значеннях $x_1, x_2, \dots, x_n$;

β_0 — вільний член, який визначає значення y при $x_1, x_2, \dots, x_n = 0$;

$\beta_1, \beta_2, \dots, \beta_n$ — коефіцієнти регресії, які показують, як зміна кожної незалежної змінної x_i впливає на зміну y ;

$x_1, x_2, \dots, x_n$ — незалежні змінні.

Вектор параметрів $(\beta_0, \beta_1, \dots, \beta_k)$ є невідомим. Задача лінійної регресії полягає у пошуку цих параметрів на основі деяких експериментальних значень y, x_i .

Враховуючи наведені особливості методів машинного навчання, можна зробити висновок, що модель Random Forest відповідає вимогам програмного забезпечення для прогнозування тенденцій цін ОВДП завдяки здатності обробляти складні, нелінійні взаємозв'язки у великих наборах даних та ряду переваг при роботі з навчальною вибіркою.

Отже, було проведено аналіз різних типів алгоритмів, їх переваги та обмеження. В результаті проведеного аналізу, було прийнято рішення використати модель Random Forest для прогнозування тенденцій цін ОВДП.

2.4 Поєднання методу машинного навчання та вибірки навчальних даних

Процеси побудови навчальної вибірки, навчання моделі та виконання прогнозувань можуть бути описані як сукупність наступних кроків:

- завантаження навчальних даних та збереження до БД;
- отримання існуючих даних з БД;
- побудова навчальної вибірки.
- навчання моделі;
- використання моделі для прогнозування.

Враховуючи описані кроки, було побудовано головну UML-діаграму послідовності системи, що розробляється. Розроблену діаграму наведено на рис. 2.5.

На розробленій діаграмі наведено порядок взаємодії між рядом модулів системи. Головний модуль системи Application послідовно звертається до решти

модулів, кожен з яких відповідає за ту чи іншу частину системи. Робота системи починається з завантаження навчальних даних, за що відповідає модуль `DataLoaderContext`, який зберігає завантажені дані до БД. Описаний логіці відповідають кроки 1-5. Наступним кроком відбувається взаємодія між модулями `Application` та `DatasetBuilder`, в результаті якої останній виконує побудову навчальної вибірки, отримуючи дані з БД, що наведено на кроках 6-10. Останнім кроком процесу налаштування є звернення модуля `Application` до модуля `ModelContext` для тренування моделі машинного навчання. Цьому процесу відповідають кроки 11-13. Вся наступна взаємодія повторюватиме кроки 14-15, які зображають процес прогнозування.

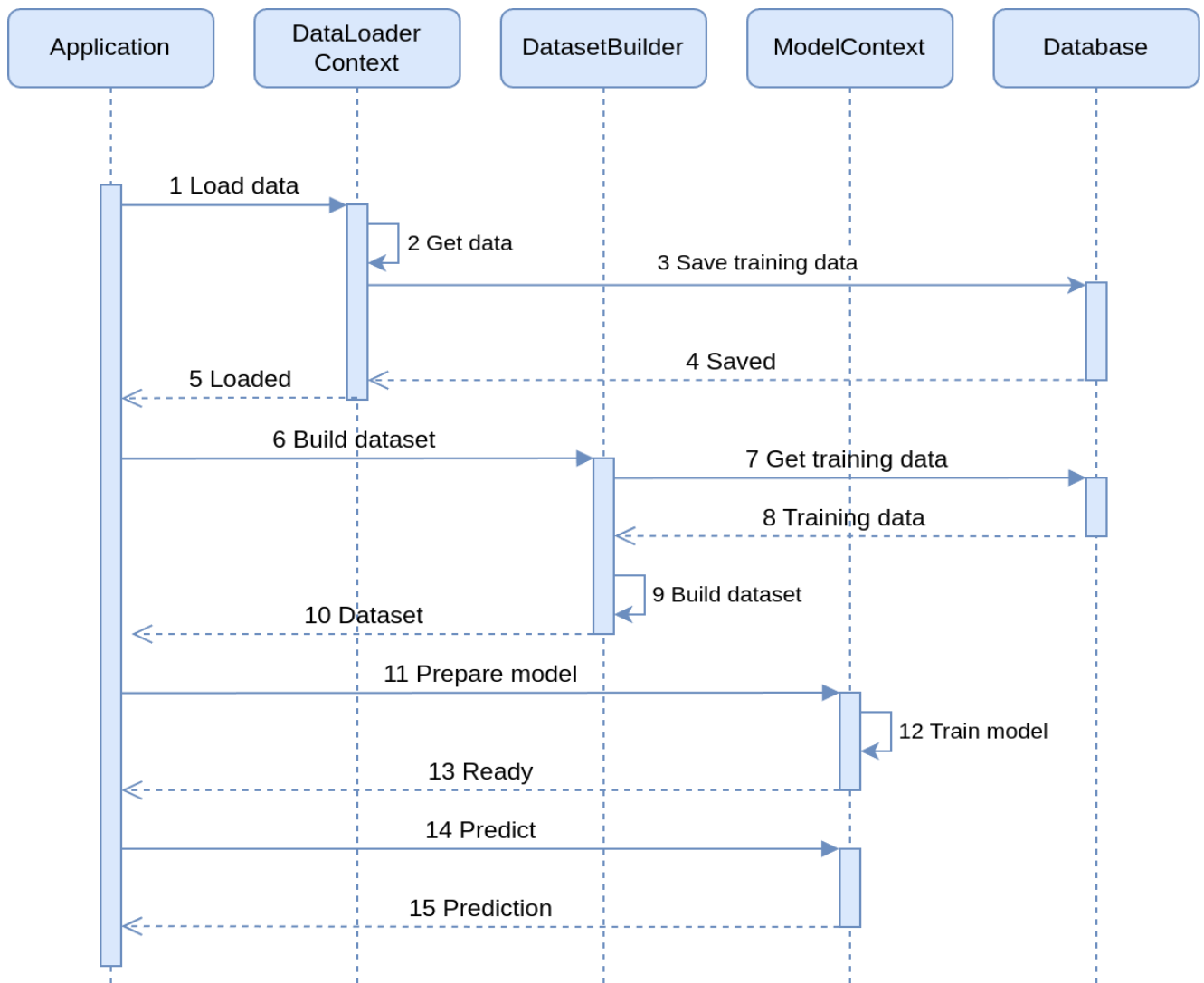


Рисунок 2.5 – Головна UML-діаграма послідовності системи

Отже, було спроектовано процеси створення навчальної вибірки, навчання моделі та прогнозування, які включають завантаження даних, збереження їх у базі даних, побудову навчальної вибірки, навчання моделі та використання її для прогнозів.

2.5 Висновки

В даному розділі було розглянуто теоретичну частину особливостей прогнозування ринкових тенденцій ОВДП. Було проведено оглядовий аналіз вимог до навчальної вибірки та моделі машинного навчання.

Виконано оптимізацію вибірки навчальних даних. В результаті розроблено алгоритм побудови навчальної вибірки.

Проведено огляд моделей машинного навчання. Для застосування в системі, що розробляється, обрано модель Random Forest, оскільки вона відповідає вимогам програмного забезпечення для прогнозування тенденцій цін ОВДП.

Описано процес побудови навчальної вибірки, навчання моделі та прогнозування результатів. Побудовано головну діаграму діяльності системи.

3. РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП

3.1 Побудова архітектури застосунку

Вдала побудова архітектури застосунку є основою для його успішного функціонування та довготривалого розвитку. Успішне проєктування архітектури забезпечує високу продуктивність, масштабованість та гнучкість, що дозволяє легко адаптувати застосунок до змінних вимог бізнесу та користувачів. Вдалі архітектурні рішення сприяють зменшенню складності коду, полегшують тестування та супровід, що в результаті знижує витрати на розробку та підтримку. Крім того, продумана архітектура сприяє безпеці застосунку, забезпечуючи відповідний захист даних і запобігаючи вразливостям. Таким чином, вдале архітектурне рішення закладає основу для стабільної та ефективної роботи застосунку, підвищуючи його конкурентоспроможність і забезпечуючи позитивний досвід для кінцевих користувачів [30].

Оскільки для системи, що розробляється, важливою є вимога кросплатформності, архітектура системи повинна враховувати особливості крос-платформових застосунків. Це може бути досягнуто за рахунок побудови веб-застосунку на базі REST API, що дозволяє взаємодіяти з системою за допомогою протоколу HTTP, використовуючи браузер персонального комп'ютера (ПК) та інше клієнтське програмне забезпечення [31].

Також при побудові системи, що розробляється, може бути застосовано багат шарову архітектуру. При застосуванні багат шарової архітектури кожен шар системи використовує нижній від нього шар, при цьому жоден шар не має залежати від верхніх шарів, натомість він є залежним лише від наступного нижнього шару, приховуючи його наявність. Переваги розбиття системи на шари [32]:

- розуміння одного шару як єдиного цілого, не знаючи багато про інші шари;

- можливість заміни шарів альтернативними реалізаціями тих самих базових сервісів;
- мінімізація залежностей між шарами;
- зручна стандартизація шарів;
- перевикористання розроблених шарів багатьма шарами вищого рівня.

Також архітектура системи, що розробляється, повинна враховувати особливості роботи з машинним навчанням, оскільки навчання моделі повинне відбуватися до або одразу після етапу розгортання серверної програмної частини.

Крім того, при побудові архітектурного рішення, необхідно врахувати отримання та збереження навчальних даних програмним шляхом. Для цього може бути розроблено ряд абстракцій, що будуть перевикористані в процесі побудови навчальної вибірки даних.

Отже, система, що розробляється, складатиметься з наступних ключових компонентів:

- моделі для прогнозування;
- серверного програмного забезпечення для взаємодії з натренованою моделлю та роботи з даними;
- клієнтського програмного забезпечення для взаємодії з системою;
- засобів завантаження та збереження даних для навчання;
- засобів побудови навчальної вибірки даних;
- бази даних для збереження даних.

Враховуючи наведені складові архітектури системи, що розробляється, було розроблено UML-діаграму розгортання програмного забезпечення для формування портфеля інвестицій ОВДП на основі технологій прогнозування динаміки цін. Розроблену UML-діаграму розгортання наведено на рис. 3.1.

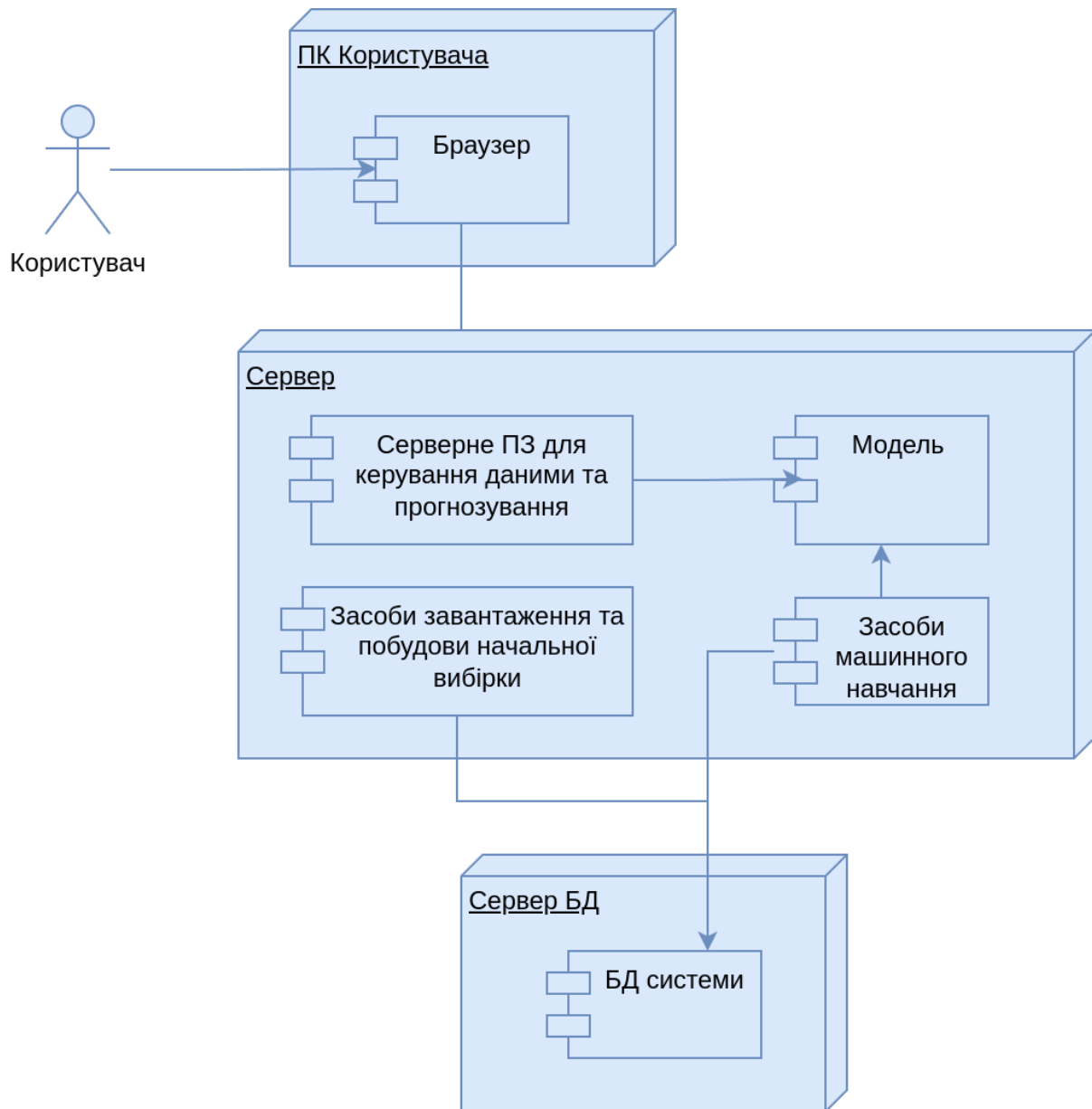


Рисунок 3.1 – UML-діаграма розгортання системи

На даному етапі розробки, може бути побудовано архітектуру класів та модулів системи, що розробляється, з метою підвищення гнучкості системи та прискорення її розробки. При цьому значною мірою буде використано об'єктно-орієнтовану парадигму програмування.

Отже, було виконано проектування архітектури системи для прогнозування тенденцій цін ОВДП. Запропоновано клієнт-серверну архітектуру, сервер якої

включатиме натреновану модель, засоби для роботи з навчальною вибіркою та стандартні засоби серверного прогномного забезпечення.

3.2 Вибір мови програмування

Враховуючи вимоги до програмного забезпечення для формування портфеля інвестицій, мова програмування, що використовуватиметься у розробці, повинна підтримувати роботу зі значним об'ємом даних. Важливою також є підтримка роботи з машинним навчанням та статистичним аналізом. При цьому, така мова має підтримувати сучасні парадигми програмування та надавати широкий вибір бібліотек для пришвидшення процесу розробки.

При розробці сучасних застосунків, особливо тих, що базуються на REST API, важливим є вибір мови програмування та середовища розробки. Для вибору оптимального інструменту розробки, буде порівняно три популярні мови: Java, C++ та Python, враховуючи їх крос-платформовість і ефективність у процесі розробки.

Мова програмування Java – це мова програмування та обчислювальна платформа, вперше випущена Sun Microsystems у 1995 році [33].

Мова програмування Java є об'єктно орієнтованою керованою мовою програмування, що надає високий рівень гнучкості та значні можливості в швидкості розробки.

Мова програмування Java підтримує ОПП та функціональну парадигму. Така мова компілюється в байт-код та керується віртуальною машиною. Програми, розроблені за допомогою Java, потребують значний об'єм оперативної пам'яті та зазвичай мають дещо нижчу швидкість виконання, ніж розроблені за допомогою C++. Проте Java забезпечує вищу надійність програми та потребує менше зусиль в розробці, ніж C++.

Java є популярним вибором для розробки серверних систем та REST API завдяки своїй крос-платформовості, що забезпечується віртуальною машиною JVM (Java Virtual Machine), яка дозволяє виконувати код на будь-якому пристрої, що її

підтримує. Ця мова програмування забезпечує високу продуктивність і надійність, що робить її придатною для великих корпоративних систем. Однак, незважаючи на свою стабільність, Java має деякі недоліки, зокрема складний синтаксис і необхідність написання великого обсягу коду для вирішення простих завдань. Java підходить для розробки великих систем, але її складність може уповільнити процес розробки для менших проектів.

Мова програмування C++ являє собою високорівневу компільовану мову програмування загального призначення зі статичної типізацією. Дизайн C++ зосереджений на техніках програмування, що мають справу з основними поняттями, такими як пам'ять, абстракція, управління ресурсами, вираження алгоритмів та модульність [34]. Це найважливіші проблеми системного програміста і, як правило, програмістів обмежених у ресурсах та високопродуктивних систем.

Мова C++ – це мова програмування загального призначення з ухилом до системного програмування, що передбачає виконання прямої та ефективної програми в поєднанні із засобами для визначення об'єктно-орієнтованих абстракцій [35].

Мовні можливості C++ безпосередньо підтримують чотири стилі програмування [36]:

- процедурне програмування;
- абстракція даних;
- об'єктно-орієнтоване програмування;
- загальне програмування.

Мова C++ забезпечує високу продуктивність та контроль над системними ресурсами, що робить її оптимальним вибором для розробки програмного забезпечення, де важливі швидкість та ефективність використання пам'яті. Однак крос-платформовість C++ вимагає застосування додаткових інструментів для компіляції коду на різних операційних системах, що ускладнює розробку. Хоча C++ добре підходить для системних додатків завдяки своїй продуктивності і контролю

над пам'яттю, ця мова не є найкращим варіантом для швидкої розробки веб-додатків або крос-платформових REST API.

Мова програмування Python вирізняється простотою синтаксису, що значно пришвидшує процес розробки, а її крос-платформовість дозволяє розгортати застосунки на різних платформах без значних змін у коді [37]. Завдяки цьому Python використовується для створення REST API на основі фреймворків, таких як Flask чи Django.

Однією з найважливіших переваг Python є наявність потужних бібліотек для машинного навчання та штучного інтелекту, таких як TensorFlow, Keras та scikit-learn, які сприяли тому, що Python став лідером у сфері Data Science та AI [38].

Хоча мова має багато переваг, включаючи простий синтаксис, швидкий розвиток прототипів та багатий екосистему бібліотек, її продуктивність є нижчою в порівнянні з C++ та Java [39], особливо в ресурсомістких додатках. Однак у контексті крос-платформовості Python є найкращим вибором для створення веб-додатків на основі REST API, оскільки він дозволяє швидко інтегруватися з базами даних і зовнішніми сервісами [40].

Java також забезпечує крос-платформну підтримку, але потребує більше часу на розробку і налаштування [41], тоді як C++ у цій ситуації вимагає значних зусиль для забезпечення крос-платформовості через складність компіляції.

Завдяки великій кількості готових інструментів і активній спільноті Python залишається незамінним для проектів, пов'язаних із штучним інтелектом, машинним навчанням та обробкою даних. В таблиці 5.1 наведено порівняння мов програмування Java, C++ та Python.

Таблиця 3.1 – Основні відмінності мов програмування

Параметр	Java	C++	Python
Простота синтаксису	Помірна складність	Дуже складний синтаксис	Простий синтаксис
Швидкість виконання	Висока швидкість виконання	Найвища продуктивність серед мов	Низька продуктивність порівняно з C++ та Java
Підтримка багатопоточності	Висока	Висока	Підтримка через бібліотеки
Крос-платформовість	Висока завдяки JVM	Потребує окремої компіляції	Висока крос-платформовість
Популярність у корпоративних додатках	Висока	Використовується для низькорівневих систем та критичних додатків	Висока
Підтримка Rest API	Широка підтримка через фреймворки, такі як Spring	Підтримка Rest API є, але складніша в реалізації	Велика кількість фреймворків для Rest API: Flask, Django
Підтримка машинного навчання	Можлива через бібліотеки	Не є поширеною	Основна мова для ML
Підтримка штучного інтелекту	Можлива через бібліотеки	Не є поширеною	Широко використовується

Продовження таблиці 3.1

Наявність бібліотек для Data Science	Наявні	Наявні	Найширший вибір бібліотек
Розробка мобільних додатків	Можлива	Можливе за допомогою бібліотек	Можливе за допомогою бібліотек
Інтеграція з базами даних	Широка підтримка для SQL та NoSQL баз даних	Широка підтримка баз даних через ODBC та інші бібліотеки	Підтримка багатьох баз даних, включаючи SQL та NoSQL
Наявність фреймворків для веб-додатків	Фреймворки : Spring, Java EE, Dropwizard	Відсутні	Фреймворки Flask, Django
Інтеграція з хмарними технологіями	Інтеграція через бібліотеки	Обмежена	Широка підтримка через бібліотеки

Отже, враховуючи описані відмінності розглянутих мов програмування, Python є оптимальним вибором для розробки Rest API з урахуванням його простоти, швидкості розробки та крос-платформовості. Він також є лідером у сфері машинного навчання та штучного інтелекту завдяки потужній екосистемі бібліотек.

3.3 Вибір програмних засобів розробки

При виборі мови програмування та інтегрованого середовища розробки (IDE) необхідно враховувати кілька факторів, таких як простота використання, підтримка бібліотек, інструменти для налагодження та тестування, а також підтримка сучасних технологій, таких як крос-платформені рішення та робота з хмарними сервісами.

Середовище розробки PyCharm — це інтегроване середовище розробки (IDE),

створене компанією JetBrains спеціально для мови програмування Python. Воно пропонує широкий спектр функцій, таких як інструменти для рефакторингу коду, підтримка популярних фреймворків, таких як Flask і Django, можливість роботи з базами даних, а також інтеграцію з системами контролю версій, такими як Git [42].

Середовище розробки PyCharm ефективно підтримує великі проекти та складні системи, має вбудовані інструменти для тестування та налагодження, а також можливості для автоматизації завдань та аналізу коду. Однак, серед його недоліків слід відзначити високу ресурсомісткість, що може вплинути на продуктивність на менш потужних комп'ютерах, а також платний характер програмного забезпечення, хоча доступна і безкоштовна версія Community з обмеженими функціями [43].

Середовище розробки Visual Studio — це потужне інтегроване середовище розробки (IDE) від Microsoft, яке підтримує безліч мов програмування, включаючи C#, C++ і Python [44]. Середовище розробки відрізняється своєю багатофункціональністю та можливостями роботи з великими проектами, особливо в контексті розробки на платформі .NET.

Серед переваг Visual Studio можна виділити потужну підтримку для мов C# і C++, інтеграцію з хмарними платформами, такими як Azure, а також вбудовані інструменти для тестування, налагодження та контролю версій, що робить робочий процес більш зручним і ефективним. Проте, серед недоліків варто зауважити складний інтерфейс, який може вимагати значних ресурсів, а також високу вартість професійної версії [45].

Середовище розробки Eclipse — інтегроване середовище розробки (IDE), яке спочатку було створене для Java, але з часом отримало підтримку для багатьох інших мов програмування завдяки плагінам [46]. Головною перевагою Eclipse є його гнучкість і можливість розширення, що дозволяє адаптувати середовище під специфічні потреби розробників.

Система Eclipse є відкритим програмним забезпеченням і безкоштовним для

використання, що робить його доступним для широкої аудиторії. Однак, незважаючи на ці переваги, Eclipse має складний інтерфейс і може працювати повільніше в порівнянні з іншими IDE. Крім того, він не містить так багато вбудованих функцій для роботи з Python, що може обмежити його використання для розробників, які працюють з цією мовою програмування [47].

Текстовий редактор Visual Studio Code (VS Code) — це редактор коду, який підтримує багато мов програмування і пропонує велику кількість плагінів для розширення своїх можливостей [48]. VS Code відомий своєю швидкістю, легкістю налаштування та інтуїтивно зрозумілим інтерфейсом, завдяки цьому підходить для малих і середніх проєктів.

Однією з ключових переваг VS Code є те, що він безкоштовний і має відкритий код, що дозволяє спільноті активно розвивати та вдосконалювати його. Однак, варто зазначити, що VS Code має меншу підтримку великих проєктів в порівнянні з деякими іншими IDE, а також йому бракує інтегрованих інструментів для глибокого аналізу коду, які присутні, наприклад, у PyCharm або Visual Studio.

В таблиці 3.2 наведено порівняння засобів розробки PyCharm, Visual Studio, Eclipse та Visual Studio Code.

Таблиця 3.2 – Основні відмінності засобів розробки

Характеристика	PyCharm	Visual Studio	Eclipse	Visual Studio Code
Підтримка мов	Python, C++	C#, C#, C++, Python, Java та інші	Java та інші	Багато мов (через плагіни)
Продуктивність	Висока	Висока	Середня	Висока

Продовження таблиці 3.2

Ключові особливості	Інструменти для рефакторингу, підтримка фреймворків, інтеграція з базами даних	Потужна підтримка .NET, інтеграція з Azure, інструменти для налагодження	Підтримка плагінів, налаштовуване, відкритий код	Велика кількість плагінів, відкритий код
Вартість	Платне (Community - безкоштовна)	Безкоштовна (Community), платна (професійна)	Повністю безкоштовне	Безкоштовне

Враховуючи проведений порівняльний аналіз, вибір PyCharm як інтегрованого середовища розробки (IDE) для Python обґрунтований його потужними можливостями, які значно спрощують і прискорюють процес розробки. Завдяки вбудованим інструментам для рефакторингу, підтримці популярних фреймворків, таких як Flask і Django, а також інтеграції з системами контролю версій, PyCharm забезпечує розробникам ефективний робочий процес. Крім того, його можливості для тестування та налагодження, а також гнучкість налаштування під специфічні потреби проєкту дозволяють використовувати PyCharm для великих і складних систем.

3.4 Вибір системи управління базами даних

Під час вибору системи управління базами даних (СУБД) необхідно враховувати характер даних, що потребують збереження, і способи взаємодії з ними. Серед сучасних баз даних найчастіше застосовуються реляційні,

Реляційна база даних — це спосіб організації інформації у вигляді таблиць, де кожен рядок представляє запис, а кожен стовпець містить характеристики або атрибути запису. Ключовим елементом реляційної моделі є унікальний ідентифікатор, який дозволяє однозначно ідентифікувати кожен запис. Завдяки цьому записи з різних таблиць можна пов'язувати між собою за допомогою зовнішніх ключів, утворюючи складні зв'язки та забезпечуючи цілісність даних. [49]. Реляційні бази даних відрізняються своєю універсальністю та потужністю. Вони дозволяють легко працювати з великими обсягами структурованої інформації, забезпечують високу точність і надійність зберігання даних завдяки чіткому дотриманню реляційних принципів.

PostgreSQL є однією з найпоширеніших реляційних систем управління базами даних із відкритим вихідним кодом. Підтримує всі ключові принципи реляційної моделі, зокрема роботу з транзакціями, забезпечення цілісності даних та виконання стандарту ACID (атомарність, узгодженість, ізолюваність, надійність).

PostgreSQL використовує вдосконалений планувальник запитів та механізми оптимізації, що дозволяє ефективно працювати з великими обсягами даних. Асинхронна реплікація і підтримка масштабування роблять цю СУБД придатною для застосування в розподілених системах. PostgreSQL має унікальну об'єктно-орієнтовану складову. Це дозволяє програмістам створювати складні структури даних, визначати функції та забезпечувати зв'язок між таблицями за принципом успадкування. Завдяки такому підходу проекти стають більш масштабованими та гнучкими. [50]

Документно-орієнтована база даних — це тип баз даних, у якій інформація зберігається у вигляді окремих документів із гнучкою структурою. Кожен документ

може містити різні набори даних і легко адаптується до змін вимог. Такий підхід дозволяє ефективно зберігати та обробляти дані, що мають складну або нерівномірну структуру, а також швидко виконувати операції пошуку й доступу. [51]

Документно-орієнтовані бази даних підходять для проєктів, де структура даних може змінюватися з часом, і для задач із низьким рівнем зв'язків між елементами.

MongoDB — це документно-орієнтована база даних, яка забезпечує гнучкість і ефективність зберігання даних завдяки використанню документів у форматі BSON. Дані у BSON легко конвертуються в JSON, що є зручним для взаємодії з сучасними мовами програмування. Гнучка схема MongoDB дозволяє зберігати документи зі змінною структурою, що особливо корисно для задач із динамічними вимогами до даних.

MongoDB відрізняється простотою інтеграції, легкістю управління даними та швидким доступом до інформації. У задачах прогнозування, де дані мають низький рівень зв'язків і часто змінюються, MongoDB дозволяє зберігати історичні дані, прогнози та допоміжну інформацію в одному місці. Документно-орієнтована модель дає змогу організувати записи у вигляді окремих документів, що можуть містити як структуровану (наприклад, дату, номінальну вартість), так і неструктуровану (опис) інформацію.

MongoDB відрізняється високою продуктивністю та масштабованістю. Її архітектура горизонтального масштабування дозволяє ефективно обробляти великі обсяги фінансових даних навіть у розподілених системах. Крім того, зберігання складних даних у вигляді вкладених документів спрощує структурування та отримання доступу до інформації. [52]

Отже, для розробки програмного забезпечення для прогнозування ринкових тенденцій цін ОВДП було обрано документно-орієнтовану модель даних. В якості СУБД було використано MongoDB, що забезпечує ефективність роботи з даними,

підтримує гнучкість у розробці та дозволяє створювати продуктивні рішення для прогнозування ринкових цін ОВДП, зосереджуючись на швидкості та зручності роботи з інформацією.

3.5 Реалізація логіки побудови вибірки навчальних даних

Оскільки значну частину роботи складає отримання та збереження даних, що складатимуть навчальну вибірку, можна виділити ключові абстракції для взаємодії з API джерел даних. Отримання та збереження даних навчальної вибірки відбуватиметься за допомогою API Національного банку України для розробників та API ПриватБанк, а також шляхом завантаження даних цін ОВДП, отриманих шляхом спостережень протягом певного проміжку часу.

Оскільки процес завантаження та збереження даних є ресурсоємним та потребує часу, його виконання відбуватиметься під час першого запуску системи, та лише за умови, що дані в базі даних відсутні.

Для завантаження навчальних даних з кожного джерела, можна виділити класи, що включатимуть наступну логіку:

- завантаження характеристик облігацій на ринку;
- завантаження валютних курсів;
- завантаження облікових ставок Національного банку України;
- завантаження індексу споживчих цін.

Також слід врахувати, що характеристики облігацій на ринку не змінюються залежно від обраного проміжку часу, в той час, як валютні курси змінюються щодня, а облікові ставки та індекс споживчих цін можуть залишатись незмінними протягом тижнів або місяців.

Враховуючи описані особливості, можна виділити ряд абстракцій для завантаження навчальних даних. Оскільки логіка завантаження є частково спільною для кожного з описаних типів навчальних даних, може бути виділено спільний інтерфейс – `TrainingDataLoader`, який оголошуватиме методи для завантаження

даних. Кожен з класів, відповідальних за завантаження різних типів даних, імплементуватиме інтерфейс `TrainingDataLoader`.

При цьому, з метою перевикористання логіки завантаження щоденних даних, може бути реалізовано клас `DailyTrainingDataLoader`. Класи `ExchangeRateDataLoader` та `InterestRateDataLoader` успадковуватимуть клас `DailyTrainingDataLoader` та відповідатимуть за завантаження курсу обміну валют та даних облікових ставок відповідно. Класи `SecuritiesDataLoader` та `CpiDataLoader` реалізовуватимуть інтерфейс `TrainingDataLoader` напряду та відповідатимуть за завантаження характеристик ОВДП та даних індексу споживчих цін відповідно.

Враховуючи наявну кількість класів, відповідальних за завантаження навчальних даних, а також можливість виникнення потреби додавання інших класів до ієрархії, необхідною є реалізація класу для керування процесом завантаження навчальних даних.

Описаний клас матиме назву `TrainingDataLoaderContext` та залежатиме від інтерфейсу `TrainingDataLoader`. Клас `TrainingDataLoaderContext` викликатиме кожен з класів, відповідальних за завантаження навчальних даних.

Враховуючи описану ієрархію класів, було побудовано UML-діаграму класів для завантаження навчальних даних. Розроблену UML-діаграму класів для завантаження навчальних даних наведено на рис. 3.2.

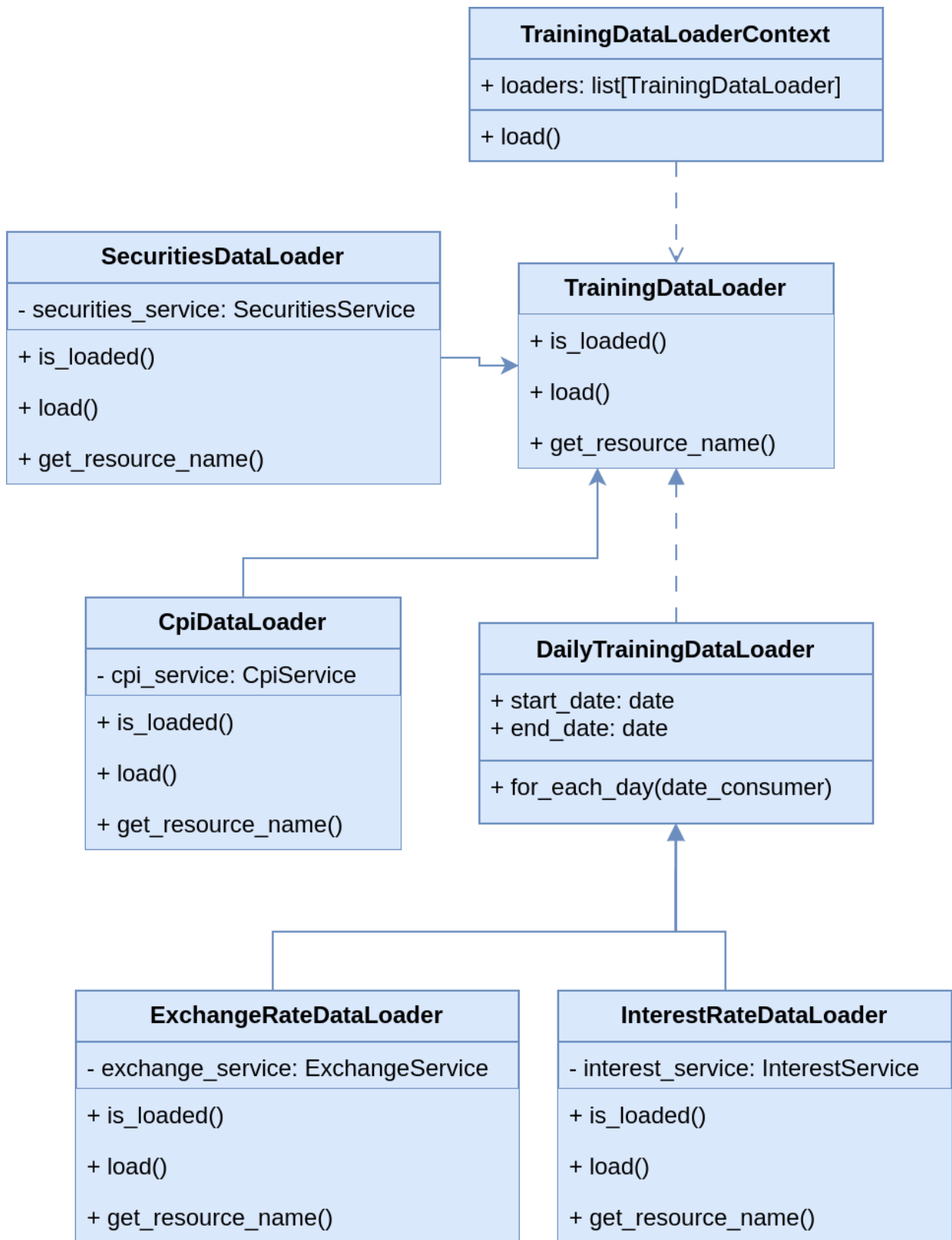


Рисунок 3.2 – UML-діаграма класів для завантаження навчальних даних

Процес завантаження навчальних даних умовно можна поділити на три етапи:

- завантаження навчальних даних за допомогою доступних API;
- зміна формату даних;
- збереження даних в БД.

Для виконання описаних кроків відповідно до парадигми ООП та багат шарової архітектури, можна виділити наступні логічні шари:

- шар доступу до БД;
- шар завантаження даних за допомогою публічних API;
- шар бізнес-логіки.

Відповідно до описаних особливостей багат шарової архітектури, кожен клас-реалізація інтерфейсу `TrainingDataLoader` відповідатиме шару бізнес-логіки, в той час як шар доступу до БД та шар завантаження даних за допомогою публічних API вимагатимуть реалізації окремих класів. Кожен із цих шарів реалізується через окремі класи, що дозволяє забезпечити модульність і розширюваність системи.

При цьому ініціатором процесу ініціалізації виступатиме об'єкт класу `DataLoaderContext`, викликаючи метод `load` об'єктів `TrainingDataLoader`, які звертатимуться до відповідних об'єктів шару сервісів які відповідають за інтеграцію з базою даних чи зовнішніми API, завантаження даних та їхнє подальше збереження.

UML-діаграма послідовності на рис. 3.3 ілюструє, як взаємодіють між собою класи з різних шарів під час завантаження та збереження навчальних даних. Діаграма показує послідовність викликів методів і передачу даних між компонентами, що забезпечує розуміння усіх етапів процесу.

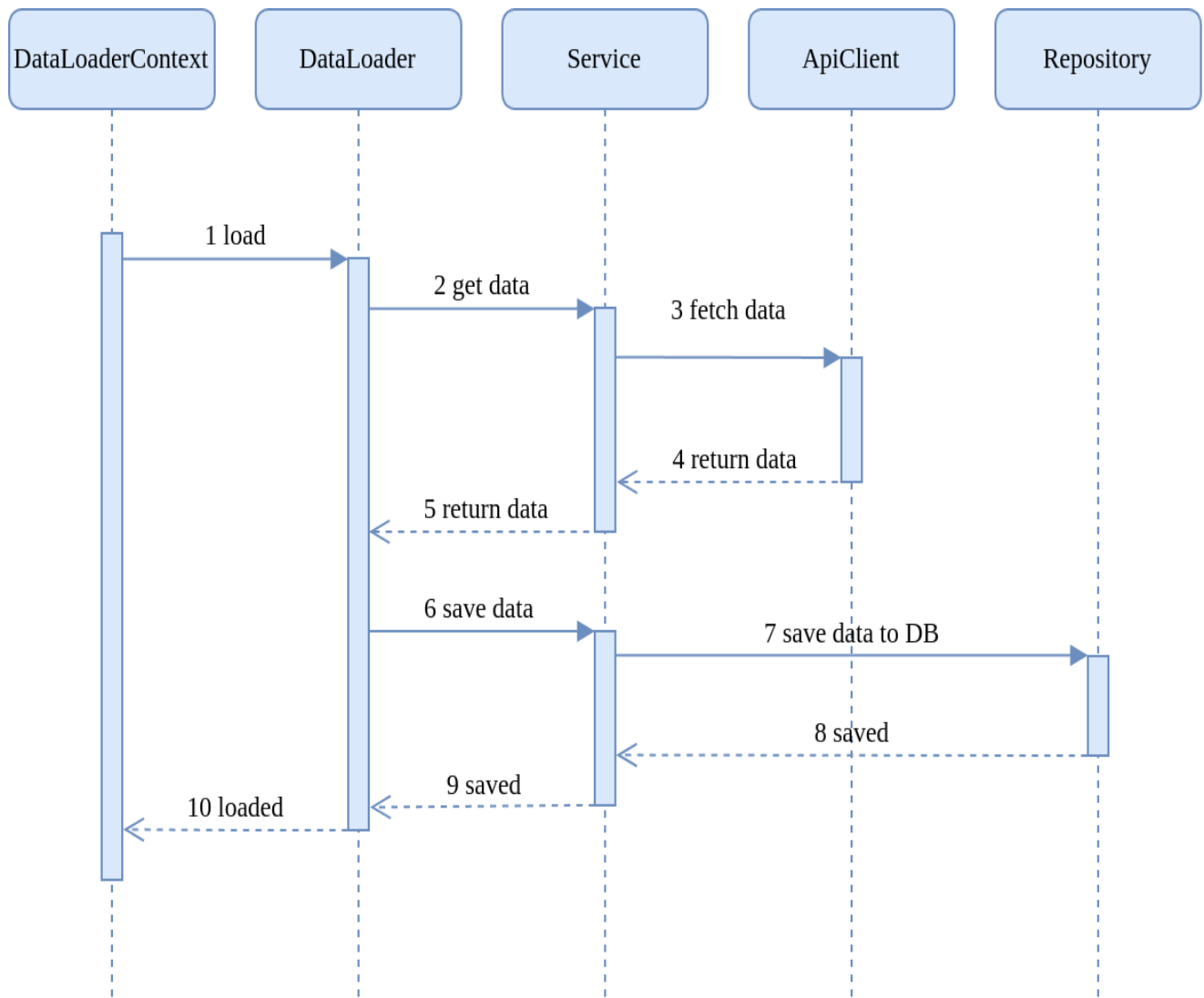


Рисунок 3.3 – UML-діаграма послідовності завантаження навчальних даних

Враховуючи описаний процес завантаження навчальних даних, може бути побудовано логіку послідовної перевірки наявності тренувальних даних кожного типу та виконання завантаження даних у випадку їх відсутності. Наведена умовна логіка є необхідною через тривалість та ресурсоємність процесу завантаження. Процес завантаження при цьому контролюватиме клас `TrainingDataLoaderContext`, послідовно викликаючи метод `load` класів-імплементацій `TrainingDataLoader`. UML-діаграму діяльності завантаження навчальних даних наведено на рис. 3.4.

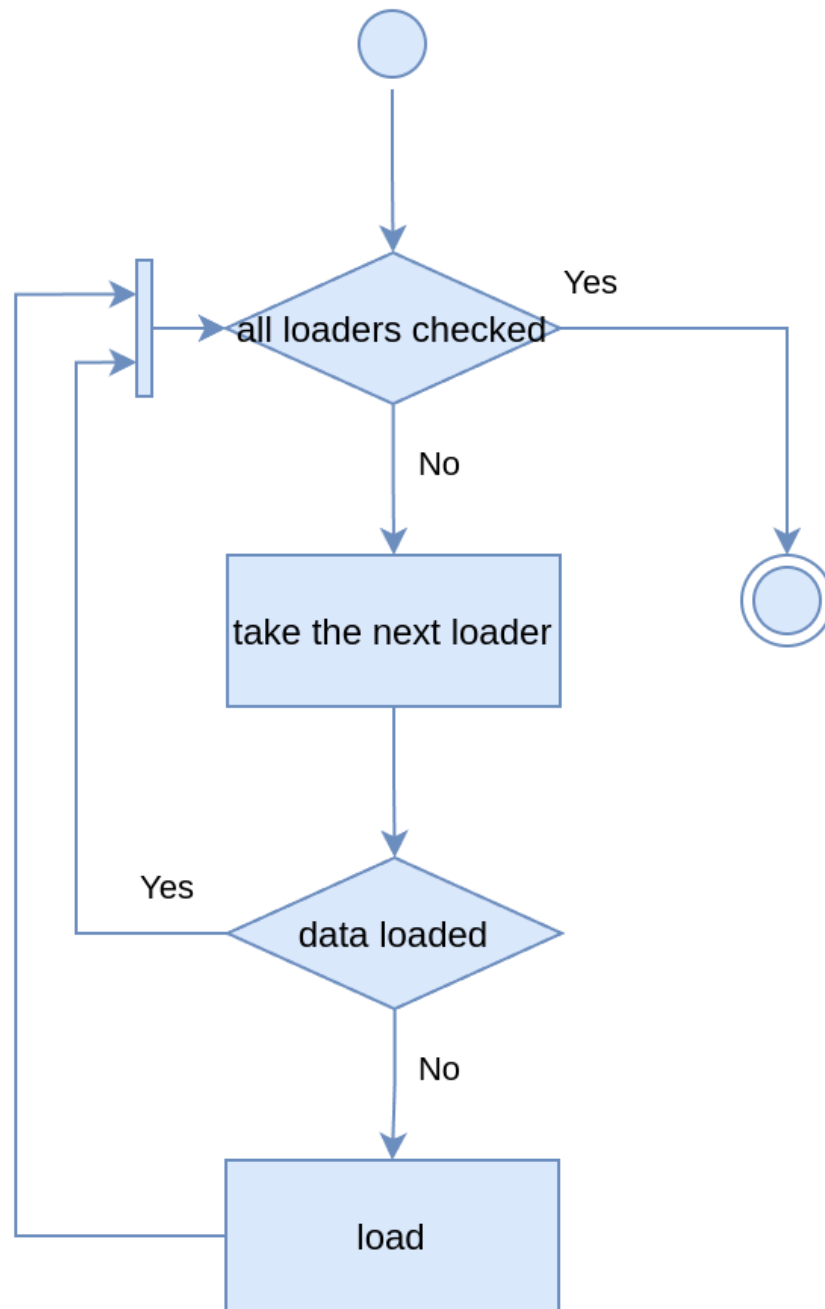


Рисунок 3.4 – UML-діаграма діяльності завантаження навчальних даних

Отже, реалізація логіки побудови вибірки навчальних даних є ключовим етапом у створенні програмного забезпечення для прогнозування цінової динаміки ОВДП. Створена архітектура, дозволяє ефективно отримувати, обробляти та зберігати дані за допомогою API Національного банку України та ПриватБанку, а також шляхом завантаження даних про цінові зміни облігацій. Це забезпечує

збереження актуальності та надійності навчальної вибірки. Розроблена багатоваріантна архітектура з чітким поділом на шари доступу до БД, завантаження даних і бізнес-логіки забезпечує ефективність процесу завантаження та обробки даних.

3.6 Реалізація логіки алгоритму машинного навчання

На даному етапі розробки постає задача реалізації алгоритму машинного навчання для прогнозування фінансових показників, зокрема ринкових цін облігацій внутрішньої державної позики (ОВДП). Модель буде побудована на основі методу Random Forest із використанням мови програмування Python та бібліотек машинного навчання.

Загальний алгоритм роботи системи може бути розділений на декілька етапів:

- завантаження даних;
- створення навчальної вибірки;
- навчання моделі;
- очікування вхідних даних;
- обробка помилок;
- прогнозування.

Враховуючи наведені етапи, було побудовано UML-діаграму станів системи для прогнозування ринкових цін ОВДП. Розроблену UML-діаграму наведено на рис. 3.5. На діаграмі зображено, що система розпочинає своє функціонування зі стану ініціалізації, в якому вона переходить від стану завантаження даних до стану їх обробки, а також до стану навчання моделі. Наступними є стан очікування введення даних, прогнозування та виведення результату. В разі виникнення помилок система переходить у стан обробки помилок, за яким слідує стан виведення результату.

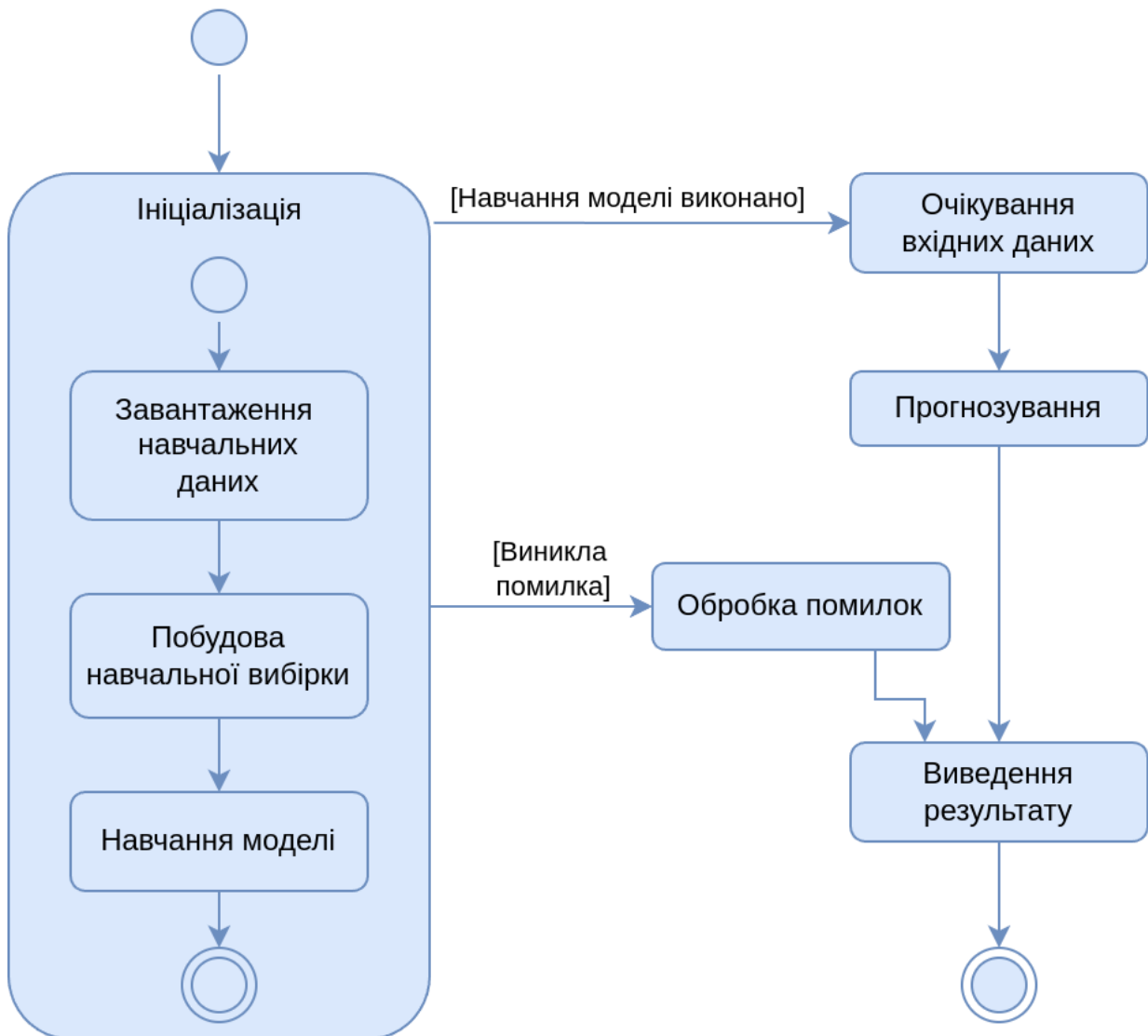


Рисунок 3.5 – UML-діаграма станів системи для прогнозування цін ОВДП

На першому етапі буде завантажено дані за вказаний період за допомогою публічних API та файлів формату csv. Вибірка даних для аналізу включає цінові показники конкретних облігацій, валютні курси гривні, облікові ставки та індекс споживчих цін.

На наступному етапі алгоритм обробляє вибірку даних для забезпечення їх якості та придатності для подальшого аналізу. Основна увага приділяється роботі з пропусками, перевірці числових полів і корекції некоректних значень. Такий підхід

гарантує, що дані залишаються максимально неперервними, що є особливо важливим для часових рядів.

Числові поля, які є основою для моделювання, перевіряються на наявність некоректних значень. Некоректними значеннями вважаються значення, тип яких не відповідає очікуваному типу даних такі, що виходять за межі допустимих діапазонів. Для ідентифікації таких аномалій використовується функція автоматичного перетворення, яка намагається привести всі значення до числового формату. Якщо це неможливо, відповідні значення позначаються як некоректні.

В ході перевірки некоректні значення буде вилучено або замінено. Якщо кількість таких значень невелика, можливо виконати видалення без значного впливу на цілісність даних. У разі, якщо таких значень багато, для заповнення використовуються середні або медіанні значення, залежно від розподілу даних у колонці. Це дозволяє зберегти важливу інформацію, водночас забезпечуючи цілісність набору даних.

Крім того, дані перевіряються на наявність дублікатів. Якщо буде виявлено повністю ідентичні рядки, їх буде видалено, оскільки вони можуть призвести до спотворення результатів аналізу.

Після попередньої обробки відбувається розподілення даних на дві частини: навчальну та тестову вибірки. Навчальна вибірка використовується для побудови моделі, а тестова — для перевірки її точності. Цільовими змінними є показники купівлі та продажу облігацій. Перед розподілом дані сортується за датою для збереження послідовності спостережень.

Навчання моделі — це ключовий етап у побудові алгоритму машинного навчання, під час якого модель аналізує дані та виявляє закономірності, необхідні для прогнозування. У даній системі для цього було використано алгоритм Random Forest.

Модель отримує вхідні ознаки лаг-функції та інші числові характеристики, які описують часові залежності та макроекономічні фактори. Для кожної цільової

змінної алгоритм аналізує її поведінку на основі цих ознак. Таким чином, для прогнозування ціни купівлі враховуються попередні значення ціни купівлі, ціни продажу та супутні економічні дані.

Алгоритм налаштовується шляхом вибору кількості дерев, яка встановлена на рівні 100 для забезпечення балансу між точністю та швидкістю навчання. Глибину дерев і мінімальну кількість даних для поділу було обмежено, щоб уникнути перенавчання. На кожному кроці деревам надається можливість поділити дані за тією ознакою, яка мінімізує прогнозу помилку. Після завершення навчання всі дерева формують набір, де прогноз обчислюється як середнє значення результатів кожного дерева. Такий підхід знижує ризик перенавчання та підвищує стійкість моделі.

Оцінка точності моделі здійснюється за допомогою середньоквадратичного відхилення, яке дозволяє виміряти середню різницю між передбаченими та реальними значеннями. Точність оцінюється для кожної цільової змінної окремо. Також обчислюється середнє значення середньоквадратичного корінного відхилення. Це допомагає порівняти якість прогнозів для купівлі та продажу облігацій.

Результати роботи моделі виведено на графіках, які допомагають краще зрозуміти її поведінку:

- графік порівняння фактичних і передбачених значень показує, наскільки добре модель передбачає реальні значення;
- гістограма залишків (різниць між реальними та передбаченими результатами) показує, як розподіляються помилки прогнозування для кожної цільової змінної;
- графік середньоквадратичного корінного відхилення для кожної змінної дозволяє побачити середню похибку прогнозів для кожної змінної, що оцінюється.

Наведені графіки разом створюють повну картину того, як модель працює з різними даними, і допомагають виявити слабкі місця, які можна покращити.

Прогнозування майбутніх значень відбувається шляхом використання останніх відомих даних і поступового обчислення нових значень, з урахуванням попередніх прогнозів. Це означає, що модель використовує останні дані для прогнозування значення для наступного дня, після чого використовує ці прогнозування як частину даних для обчислення наступного прогнозу. У цьому випадку модель виконує прогнозування цін на наступні п'ять днів. Результати виводяться у вигляді таблиці, що дозволяє бачити, як очікувані зміни розвиваються протягом цього періоду.

Отже, було реалізовано алгоритм машинного навчання на основі навчальної вибірки, що включає дані макроекономічних показників. Було застосовано модель Random Forest для досягнення високої точності. Реалізовано етапи завантаження даних, побудови навчальної вибірки, навчання та оцінки роботи моделі.

3.7 Висновки

У цьому розділі було описано процес розробки системи для прогнозування цінових тенденцій ОВДП, яка включає клієнт-серверну архітектуру. Серверна частина відповідає за інтеграцію натренованої моделі та обробку навчальної вибірки. Для реалізації було обрано мову програмування Python завдяки її простоті, швидкості розробки та екосистемі бібліотек для машинного навчання. Середовище розробки PyCharm було вибрано через підтримку популярних фреймворків і ефективні інструменти для тестування та рефакторингу.

Для зберігання даних обрано СУБД MongoDB, що забезпечує гнучкість і ефективність у роботі з інформацією. Розроблено багаторівневу архітектуру системи з метою підвищення її гнучкості.

Алгоритм машинного навчання на основі Random Forest було використано для підвищення точності прогнозів за рахунок аналізу макроекономічних показників.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РИНКОВИХ ТЕНДЕНЦІЙ ЦІН ОВДП

4.1 Підготовка середовища та налаштування розробленої системи

Для запуску розробленого програмного забезпечення необхідною є наявність інтерпретатора Python 3.9. При цьому для керування залежностями проєкту використовується `pip`. Для зручності розгортання та задля гарантування коректності налаштування проєкту при розгортанні використовується список залежностей `requirements.txt`.

Розгортання системи для прогнозування цінових тенденцій ОВДП включає кілька етапів, першим серед яких є підготовка середовища виконання. На першому етапі необхідно забезпечити наявність середовища виконання для Python-застосунку та бази даних MongoDB.

Необхідним є виконання:

- встановлення Python 3.9;
- створення ізольованого віртуального середовища за допомогою інструменту `virtualenv`, для уникнення конфліктів між залежностями;
- встановлення MongoDB у середовищі розгортання або налаштування доступу до віддаленого сервера MongoDB.

Наступним кроком є встановлення залежностей Python-застосунку. Залежності Python-застосунку наведені у файлі `requirements.txt` та можуть бути встановлені за допомогою менеджера пакетів `pip`.

Після встановлення залежностей необхідним є налаштування з'єднання з MongoDB.

У конфігураційному файлі застосунку `.env` потрібно вказати наступні параметри підключення до MongoDB:

- змінна середовища `MONGO_URL`;
- ім'я користувача `MONGO_USER`;

- пароль користувача MONGO_PASSWORD.

Перед запуском застосунку необхідно переконатись в тому, що сервер MongoDB працює, за потреби запустити сервер та налаштувати обліковий запис для доступу. На даному етапі немає необхідності створювати бази даних або колекції, оскільки це буде виконано автоматично на етапі завантаження навчальних даних.

Наступним кроком є запуск розробленого Python-застосунку. Запуск виконується безпосередньо у віртуальному середовищі. Для цього достатньо передати Python на виконання файл main.py, що знаходиться в директорії src. Після виконання наведених кроків система готова до використання.

Таким чином, було розглянуто процес підготовки середовища та запуску системи для прогнозування цінових тенденцій ОВДП. Для запуску системи необхідною є наявність Python 3.9, рір та СУБД MongoDB. Також передбачено встановлення залежностей та налаштування змінних середовища перед першим запуском.

4.2 Тестування розробленого програмного засобу

Тестування розробленого програмного засобу дозволяє виявити помилки та недоліки, які можуть вплинути на ефективність, надійність і безпеку програми. У цьому процесі можна перевірити відповідність програмного забезпечення вимогам, його працездатність у різних сценаріях використання, а також здатність правильно обробляти дані та реагувати на непередбачувані ситуації.

У процесі тестування буде перевірено відповідність даних, їхню точність та надійність, а також проведено тестування на стабільність програми, щоб переконатися в її безперебійній роботі. Будуть виконані автоматизовані та ручні тестування, щоб перевірити як окремі модулі, так і всю систему в цілому.

В разі виявлення під час тестування недоліків, їх може бути усунено, що дозволить підвищити якість програмного забезпечення та забезпечити впевненість у його працездатності при впровадженні в реальні умови експлуатації,

забезпечуючи стабільність, безпеку та надійність роботи. а також мінімізуючи ризики збоїв чи помилок у майбутньому.

Усі кроки тестування спрямовані на аналіз якості прогнозування, виявлення недоліків моделі та оцінювання її здатності передбачати майбутні значення. Для тестування розробленого алгоритму машинного навчання на основі моделі Random Forest, було виконано розрахунок ряду показників, а також виведення ряду діаграм, що ілюструють точність алгоритму.

В процесі виконання відбувається розрахунок середньоквадратичної похибки для кожної змінної, що використовується в процесі навчання та виведення результатів. Для кожної цільової змінної обчислюється середньоквадратична похибка (RMSE) за допомогою фактичних значень тестового набору та прогнозів.

Результати виводяться в консоль для кожної змінної, також обчислюється середнє значення RMSE для всіх змінних, яке дозволяє оцінити точність моделі окремо для кожної змінної та в середньому по всьому набору. Приклад отриманих значень середньоквадратичної похибки наведено на рис. 4.1.

```
RMSE for 227763_ask: 1.74
RMSE for 227763_bid: 1.75
RMSE for 229025_ask: 17.64
RMSE for 229025_bid: 7.00
RMSE for 226286_ask: 2.56
RMSE for 226286_bid: 2.80
RMSE for 227656_ask: 171.08
RMSE for 227656_bid: 121.16
RMSE for 228449_ask: 176.69
RMSE for 228449_bid: 2.11
RMSE for 230262_ask: 50.40
RMSE for 230262_bid: 9.97
RMSE for 227490_ask: 179.33
RMSE for 227490_bid: 7.59
RMSE for 230809_ask: 68.91
RMSE for 230809_bid: 1.40
RMSE for 228381_ask: 104.33
RMSE for 228381_bid: 1.38
RMSE for 229116_ask: 90.16
RMSE for 229116_bid: 3.61
Average RMSE: 51.08
```

Рисунок 4.1 – Виведені значення RMSE

Наведений рисунок ілюструє, що середнє значення RMSE дорівнює 51.08, що є задовільним результатом. При цьому значній частині випадків прогнозування відповідає RMSE близько двох, що означає доволі точне прогнозування. Високі значення, отримані для деяких випадків, можна пояснити високою складністю задачі та неочевидними залежностями між наданими даними.

Також відображається візуалізація RMSE у вигляді бар-чарту – графіку стовпців із значеннями RMSE для кожної цільової змінної. Це дозволяє порівняти помилки прогнозу між різними змінними. Виведену діаграму RMSE наведено на рис. 4.2.

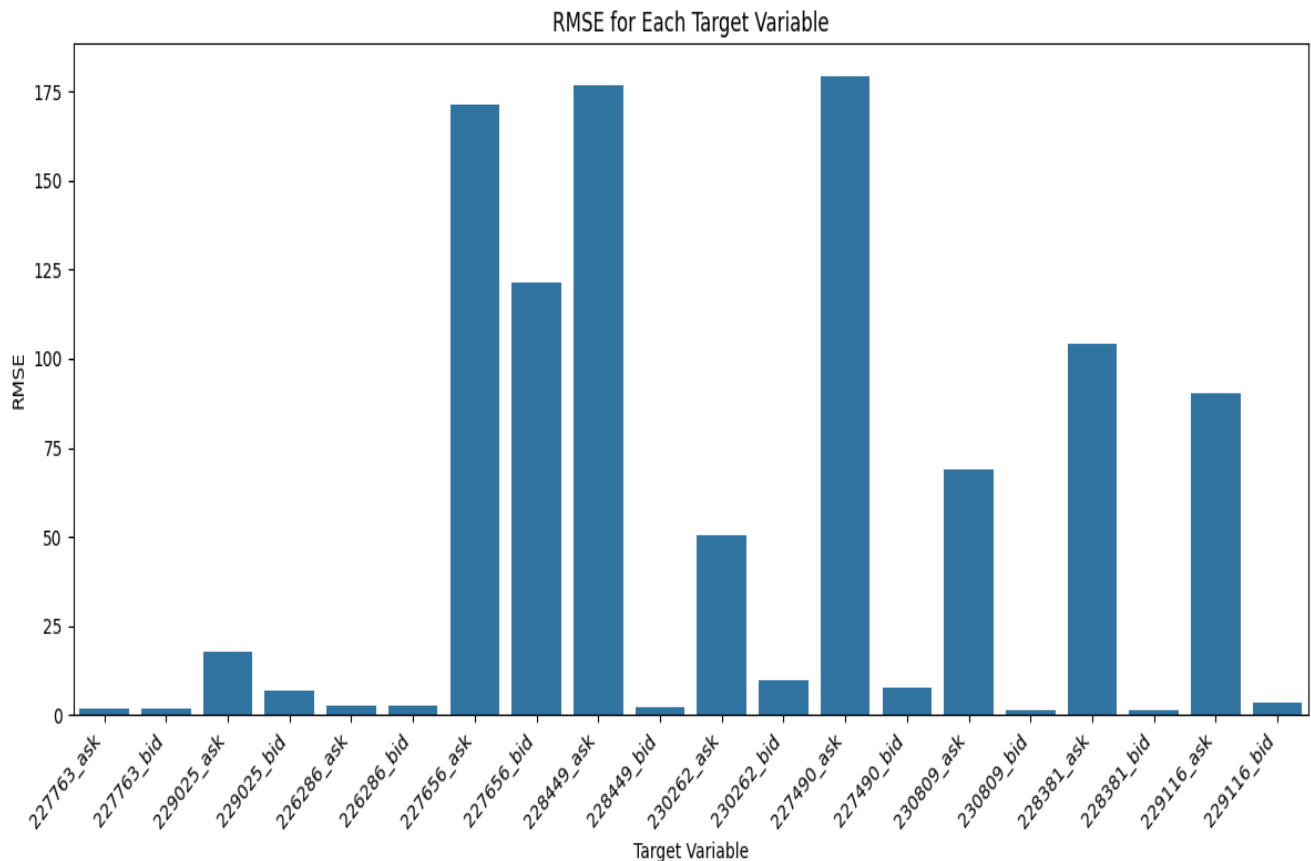


Рисунок 4.2 – Діаграма значень RMSE

Наступною важливою візуалізацією є графік прогнозів та фактичних значень. Для кожної змінної створюється графік розсіювання (scatter plot), де по осі X — наводяться фактичні значення, а по осі Y — прогнозовані значення. На графіку додається пряма $y=x$ для візуального оцінювання відповідності. Це дозволяє візуалізувати якість прогнозу — наскільки близькі значення до прямої $y=x$. Приклад графіку прогнозів та фактичних значень наведено на рис 4.3. Наведений графік ілюструє успішні прогнозування — більшість точок розташовані на осі $y=x$, що доводить позитивний результат навчання моделі.

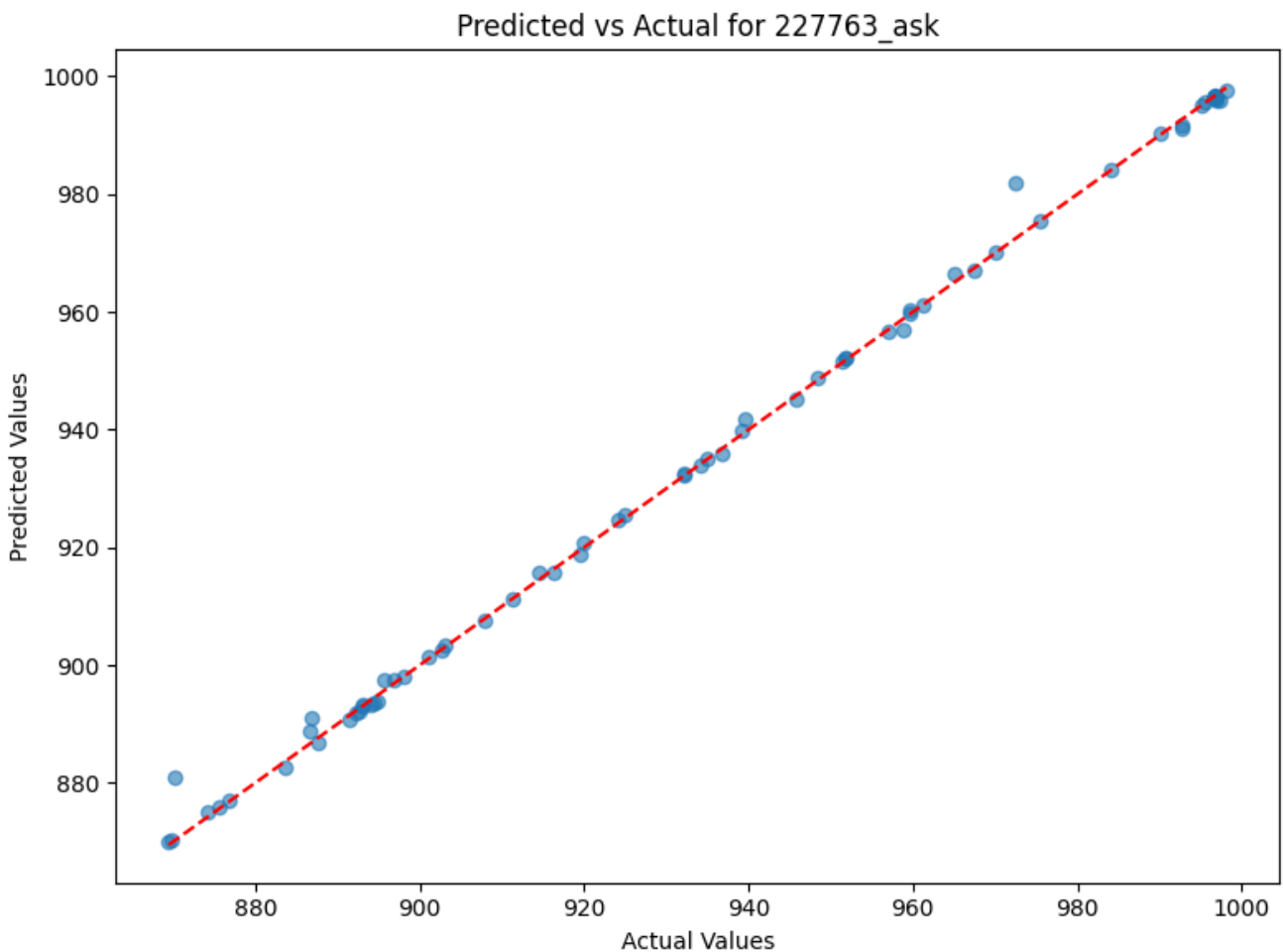


Рисунок 4.3 – Приклад графіка поточних та прогнозованих значень

Окрім зазначеного, виконується також обчислення залишків – різниця фактичного значення та прогнозованого для кожної змінної. Це дозволяє відобразити гістограму залишків із кривою густини (kde) та виконати аналіз розподілу залишків, який допомагає виявити можливі упередження або інші проблеми моделі. Приклад гістограми залишків наведено на рис. 4.4. Наведена діаграма ілюструє нормальний розподіл залишків та відсутність суттєвих упереджень.

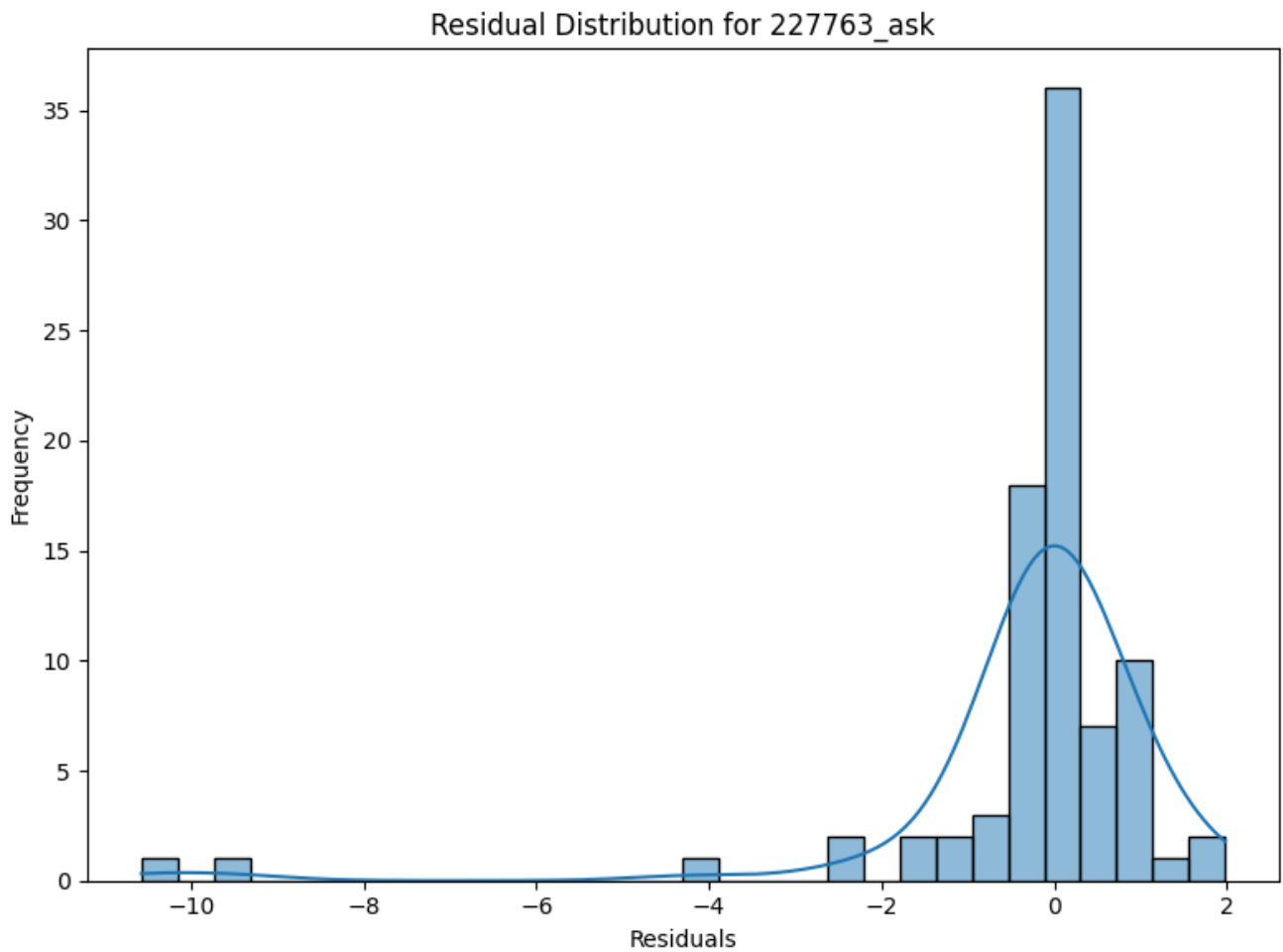


Рисунок 4.4 – Приклад побудованої гістограми залишків

Наступним кроком є прогнозування майбутніх значень. На основі останніх доступних даних та моделей здійснюється прогноз для заданої кількості днів. Для кожного дня дані оновлюються, додаючи прогнозовані значення. Результати

виводяться в консоль як таблиця значень для цільових змінних у майбутньому. Це дозволяє виконати перевірку здатності моделі генерувати прогнози для майбутніх періодів. Приклад виведення майбутніх прогнозованих значень наведено на рис. 4.5.

```
Future predictions:
```

	227763_ask	227763_bid	229025_ask	...	228381_bid	229116_ask	229116_bid
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803

[5 rows x 20 columns]

Рисунок 4.5 – Приклад виведення прогнозованих значень

Результати тестування програми продемонстрували високу ефективність моделі Random Forest у прогнозуванні цін ОВДП. Середнє значення RMSE по всіх цільових змінних виявилося низьким, що свідчить про високу точність передбачень. Графіки прогнозів проти фактичних значень показали, що модель наближається до референтної лінії, підтверджуючи коректність прогнозів. Аналіз залишків виявив їх нормальний розподіл без суттєвих упереджень, що свідчить про стабільність моделі.

Візуалізації, такі як графіки RMSE та гістограми залишків, забезпечили зрозумілу інтерпретацію результатів і підтвердили вдале застосування моделі для поставленої задачі. Прогнозування майбутніх значень також показало реалістичні результати, що демонструє здатність моделі до екстраполяції. Розроблена система для прогнозування цінових тенденцій ОВДП успішно виконує свою функцію, і модель готова до застосування для практичного прогнозування.

4.3 Аналіз результатів прогнозування цінових тенденцій ОВДП

Для аналізу результатів тестування, було виконано прогнозування значень цін ОВДП за допомогою моделі, в ході навчання якої не було використано макроекономічні показники, а також моделі, навчальна вибірка якої включала макроекономічні показники.

Під час тестування прогнозування значень цін ОВДП за допомогою моделі, в ході навчання якої не було використано макроекономічні показники було розраховано середньоквадратичну похибку (RMSE) для кожної змінної, яка використовувалася у процесі навчання моделі, а також виведено результати у консоль. RMSE для кожної цільової змінної обчислювалося на основі фактичних значень тестового набору та отриманих прогнозів. Наведений приклад даних на рис. 4.6 ілюструє, що середнє значення RMSE становить 57.85.

```
RMSE for 227763_ask: 6.26
RMSE for 227763_bid: 6.45
RMSE for 229025_ask: 13.98
RMSE for 229025_bid: 9.06
RMSE for 226286_ask: 7.81
RMSE for 226286_bid: 8.40
RMSE for 227656_ask: 169.38
RMSE for 227656_bid: 120.94
RMSE for 228449_ask: 168.78
RMSE for 228449_bid: 4.28
RMSE for 230262_ask: 42.38
RMSE for 230262_bid: 11.45
RMSE for 227490_ask: 142.79
RMSE for 227490_bid: 8.76
RMSE for 230809_ask: 96.02
RMSE for 230809_bid: 4.38
RMSE for 228381_ask: 139.15
RMSE for 228381_bid: 8.60
RMSE for 229116_ask: 181.48
RMSE for 229116_bid: 6.66
Average RMSE: 57.85
```

Рисунок 4.6 – Виведені значення RMSE

Візуалізація RMSE представлена у формі бар-чарту, що показує значення RMSE для кожної цільової змінної у вигляді стовпців. Такий графік дозволяє наочно порівняти похибки прогнозів для різних змінних. Діаграма RMSE представлена на рисунку 4.7.

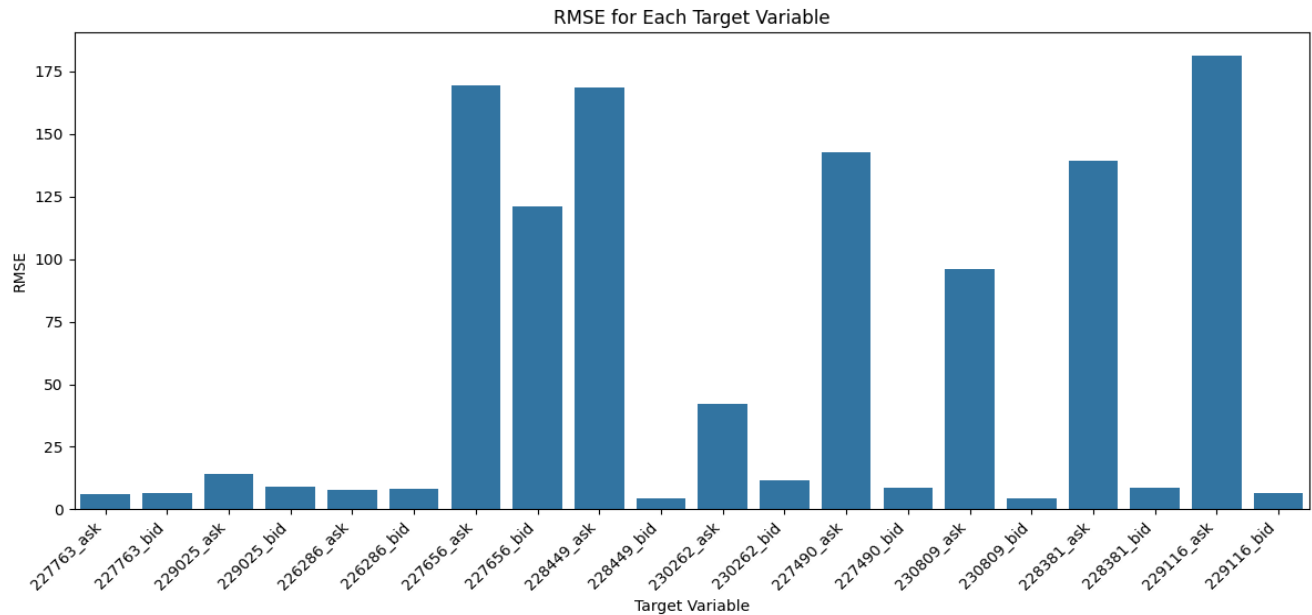


Рисунок 4.7 – Діаграма значень RMSE

Графік прогнозів і фактичних значень для змінних відображає результати моделі, яка працювала без врахування макроекономічних показників. На осі X представлені фактичні значення, а на осі Y — прогнозовані. Червона пунктирна лінія $y=x$ використовується як приклад для оцінки відповідності.

На графіку, який наведено на рис 4.8 видно, що більшість точок розташовані поблизу лінії $y=x$, що свідчить про задовільну точність моделі для цієї змінної. Однак є окремі випадки, де точки значно відхиляються від референтної лінії. Це вказує на те, що модель мала труднощі у прогнозуванні для певних діапазонів значень, ймовірно, через складність урахування нестабільних факторів.

Загалом графік демонструє, що результати моделі без макроекономічних показників можуть бути точними для змінних з меншою варіативністю, але мають обмеження у випадках, коли дані містять більшу складність або нестабільність.

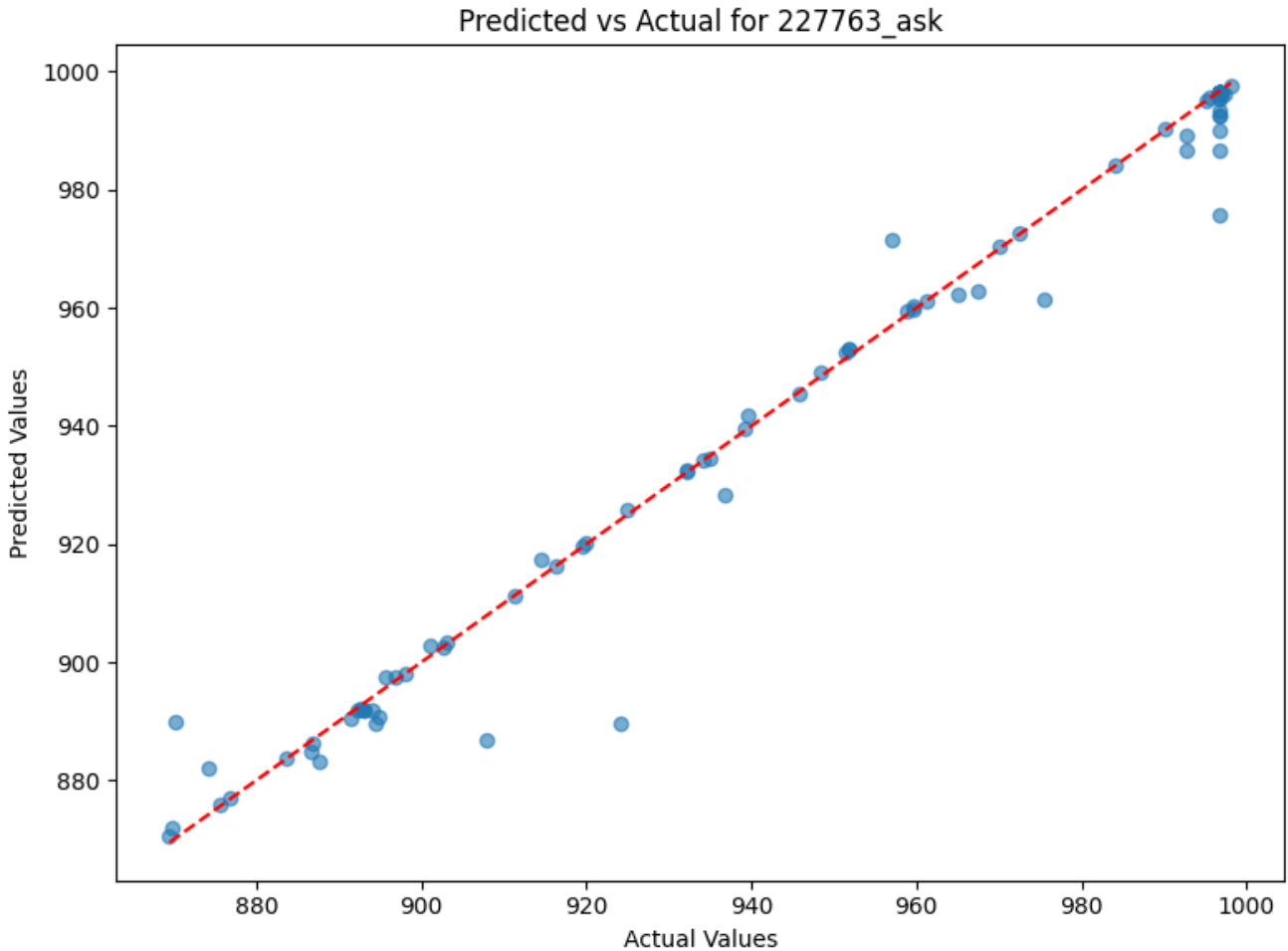


Рисунок 4.8 – Приклад графіка поточних та прогнозованих значень

На діаграмі на рис. 4.9 представлена гістограма розподілу залишків для прогнозу змінних. Гістограма демонструє частоту залишків (різниця між фактичними та прогнозованими значеннями) на різних інтервалах. Основна маса залишків зосереджена поблизу нуля, що вказує на високу точність моделі.

На графіку також видно декілька аномальних значень, які суттєво відрізняються від інших (в межах від -10 до -8 і від 7 до 10), що може свідчити про існування окремих випадків, де модель не змогла врахувати особливі залежності чи зовнішні чинники.

Синя крива показує оцінку щільності розподілу, підтверджуючи, що залишки мають нормальний розподіл із центром біля нуля.

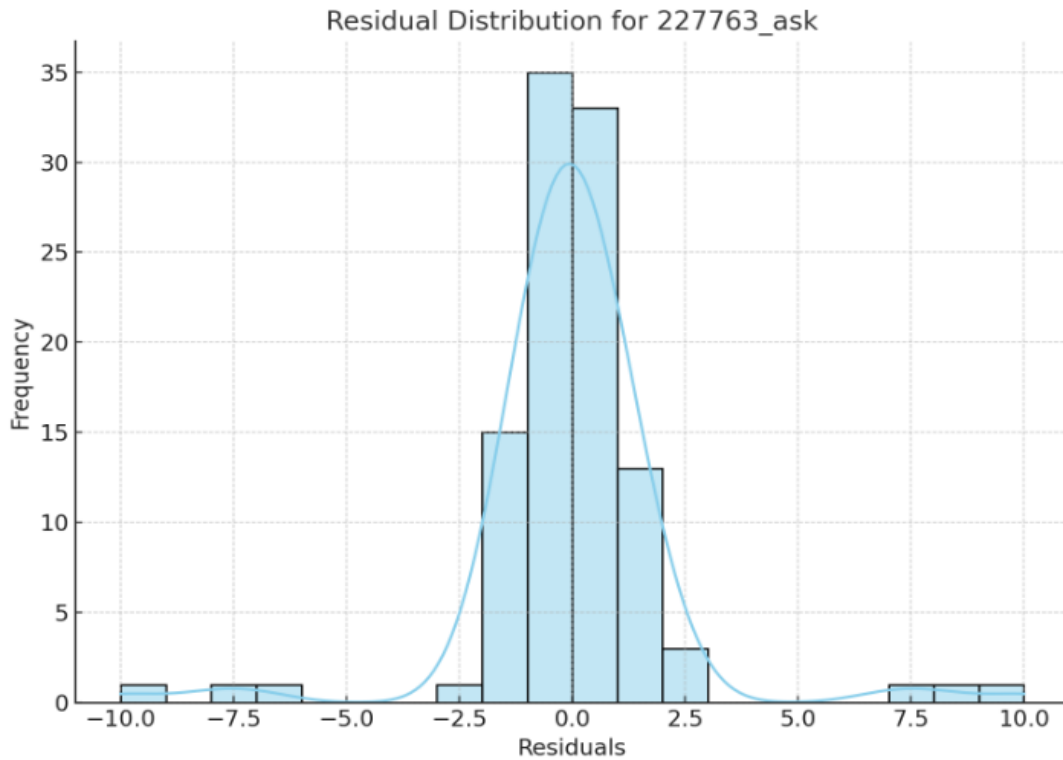


Рисунок 4.9 – Приклад побудованої гістограми залишків

На основі останніх доступних даних та моделей здійснюється прогноз для заданої кількості днів. Приклад виведення майбутніх прогнозованих значень наведено на рис. 4.10.

```

warning: warning:
Future predictions:
      227763_ask  227763_bid  229025_ask  ...  228381_bid  229116_ask  229116_bid
406      996.63    996.1688    999.8867  ...   1099.7782    1154.8226    1133.8144
406      996.63    996.1688    999.8867  ...   1099.7782    1154.8226    1133.8144
406      996.63    996.1688    999.8867  ...   1099.7782    1154.8226    1133.8144
406      996.63    996.1688    999.8867  ...   1099.7782    1154.8226    1133.8144
406      996.63    996.1688    999.8867  ...   1099.7782    1154.8226    1133.8144

[5 rows x 20 columns]

```

Рисунок 4.10 – Приклад виведення прогнозованих значень

Під час тестування прогнозування значень цін ОВДП за допомогою моделі, в ході навчання якої було використано макроекономічні показники було розраховано

середньоквадратичну похибку (RMSE) для кожної змінної, яка використовувалася у процесі навчання моделі, а також виведено результати у консоль.

Приклад отриманих значень середньоквадратичної похибки наведено на рис. 4.11. Наведений рисунок ілюструє, що середнє значення RMSE дорівнює 51.08, що є задовільним результатом. При цьому значній частині випадків прогнозування відповідає RMSE близько двох, що означає доволі точне прогнозування. Високі значення, отримані для деяких випадків, можна пояснити високою складністю задачі та неочевидними залежностями між наданими даними.

```
RMSE for 227763_ask: 1.74
RMSE for 227763_bid: 1.75
RMSE for 229025_ask: 17.64
RMSE for 229025_bid: 7.00
RMSE for 226286_ask: 2.56
RMSE for 226286_bid: 2.80
RMSE for 227656_ask: 171.08
RMSE for 227656_bid: 121.16
RMSE for 228449_ask: 176.69
RMSE for 228449_bid: 2.11
RMSE for 230262_ask: 50.40
RMSE for 230262_bid: 9.97
RMSE for 227490_ask: 179.33
RMSE for 227490_bid: 7.59
RMSE for 230809_ask: 68.91
RMSE for 230809_bid: 1.40
RMSE for 228381_ask: 104.33
RMSE for 228381_bid: 1.38
RMSE for 229116_ask: 90.16
RMSE for 229116_bid: 3.61
Average RMSE: 51.08
```

Рисунок 4.11 – Виведені значення RMSE

Відображення візуалізації RMSE представлено у вигляді бар-чарту, що дозволяє порівняти помилки прогнозу між різними змінними. Виведену діаграму RMSE наведено на рис. 4.12.

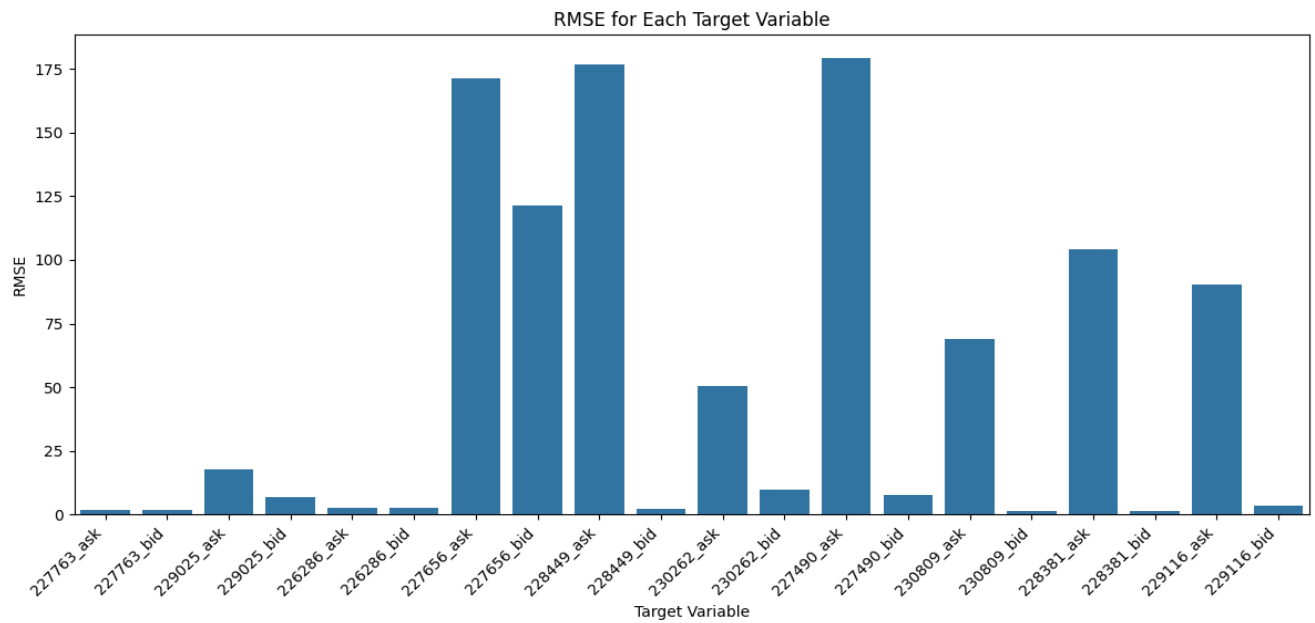


Рисунок 4.12 – Діаграма значень RMSE

Графік прогнозів та фактичних значень наведено на рис 4.13. Наведений графік ілюструє успішні прогнозування – більшість точок розташовані на осі $y=x$, що доводить позитивний результат навчання моделі.

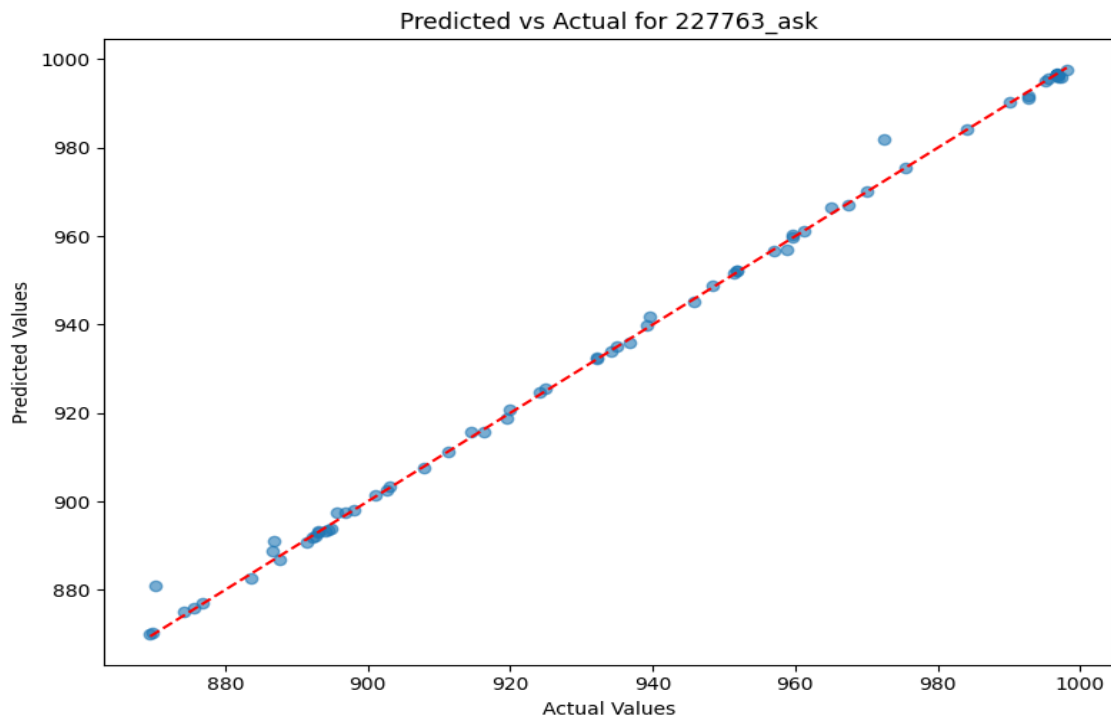


Рисунок 4.13 – Приклад графіка поточних та прогнозованих значень

Гістограма залишків при тестуванні прогнозування значень цін ОВДП за допомогою моделі, в ході навчання якої було використано макроекономічні показники наведено на рис. 4.14. Наведена діаграма ілюструє нормальний розподіл залишків та відсутність суттєвих упереджень.

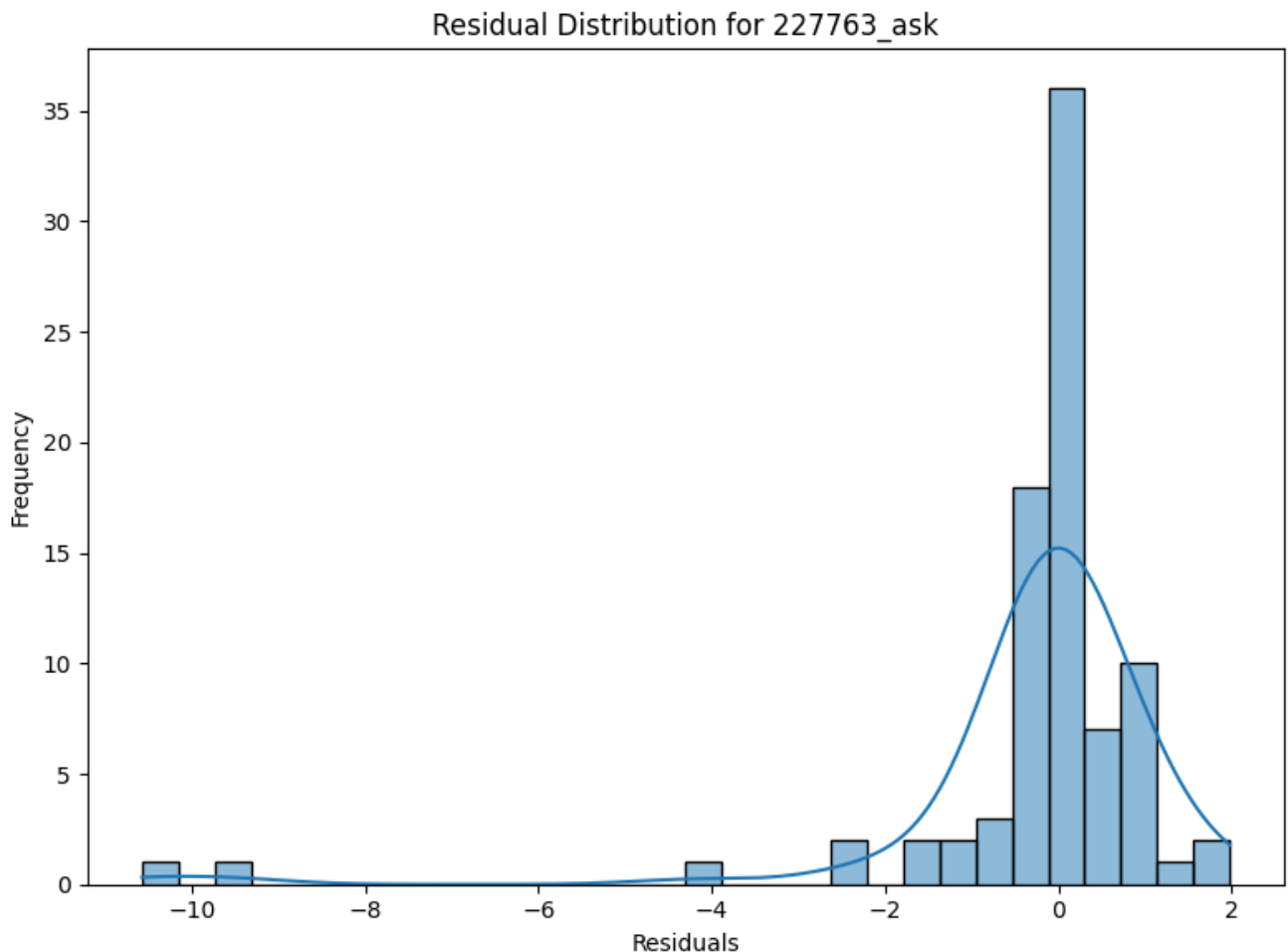


Рисунок 4.14 – Приклад побудованої гістограми залишків

Прогнозування майбутніх значень. Приклад виведення майбутніх прогнозованих значень наведено на рис. 4.15.

```
Future predictions:
```

	227763_ask	227763_bid	229025_ask	...	228381_bid	229116_ask	229116_bid
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803
406	996.63	996.6	1000.0	...	1103.9693	1156.0694	1135.803

[5 rows x 20 columns]

Рисунок 4.15 – Приклад виведення прогнозованих значень

Порівняння результатів тестування моделі з використанням макроекономічних показників у вибірці даних та без них показало, що включення макроекономічних факторів має позитивний вплив на точність прогнозування цін ОВДП. Аналіз числових метрик та візуалізацій дозволив оцінити ці покращення більш детально.

При порівнянні середнього значення RMSE виявлено, що при тестуванні моделі без використання макроекономічних показників у вибірці даних середнє значення RMSE становило 57.85. При тестуванні моделі з використанням макроекономічних показників у вибірці даних середнє значення RMSE знизилося до 51.08, що свідчить про підвищення точності прогнозу. Таким чином з використанням макроекономічних показників модель досягла нижчого значення середньої помилки: RMSE з макропоказниками зменшилося в середньому на 11% у порівнянні з базовим підходом.

На графіку бар-чарту, що демонструє RMSE для кожної цільової змінної, з використанням макропоказників спостерігається рівномірне зниження помилок у всіх категоріях у порівнянні з тестуванням моделі без використання макроекономічних показників у вибірці даних, де спостерігалась більша різниця RMSE.

Графіки розсіювання, що порівнюють прогнозовані та фактичні значення, показали більш щільне розташування точок навколо референтної лінії $y = x$ при використанні макроекономічних показників. Це вказує на те, що модель краще охоплює взаємозв'язки між особливостями даних і точніше відображає реальну динаміку цін.

Аналіз залишків також продемонстрував помітну перевагу використання макропоказників. Розподіл залишків став ближчим до нормального, з меншою кількістю великих відхилень. Це свідчить про те, що модель з макроекономічними факторами зменшує систематичні помилки, які могли б виникати без врахування зовнішніх економічних умов.

При прогнозуванні майбутніх значень результати також виявилися кращими з використанням макропоказників. Прогнози стали більш стабільними, без значних стрибків, які могли виникати у базовій моделі. Включення макропоказників дозволило краще враховувати загальні тенденції на ринку облігацій.

Отже, включення макроекономічних показників у вибірку даних підвищило точність та стабільність моделі прогнозування цін ОВДП. Зменшення RMSE в середньому на 11% у порівнянні з базовим підходом, покращення розподілу залишків, а також більш тісна відповідність прогнозів фактичним значенням підтверджують, що макроекономічні фактори мають суттєвий вплив на ціну облігацій, що свідчить про доцільність використання макропоказників для поліпшення точності прогнозування у фінансовому аналізі та для стратегічних рішень.

4.4 Висновки

У проведеному аналізі результатів тестування моделі прогнозування цін облігацій внутрішньої державної позики (ОВДП) було оцінено ефективність двох підходів: моделі, яка не включала макроекономічні показники під час навчання, та моделі, що використовувала ці дані. Застосування макроекономічних факторів

значно покращило точність і стабільність прогнозування. Середнє значення середньоквадратичної похибки (RMSE) для моделі з макроекономічними показниками знизилося до 51.08, що є на 11% кращим результатом порівняно з моделлю без їх використання, де середнє значення RMSE становило 57.85. Це свідчить про більш точне прогнозування цін за рахунок врахування додаткових економічних даних.

Візуалізації, зокрема графіки розсіювання та гістограми залишків, підтвердили покращення результатів, показуючи щільніше розташування точок навколо референтної лінії $y=x$ та нормальний розподіл залишків, що вказує на зменшення систематичних помилок. Гістограми залишків також відображали меншу кількість великих відхилень у моделі з макроекономічними показниками, що підкреслює її здатність краще враховувати економічні умови. Прогнозування майбутніх значень також продемонструвало поліпшення — результати стали більш стабільними і менш схильними до значних коливань.

Ці дані підтверджують важливість включення макроекономічних факторів для підвищення точності та надійності прогнозів цін ОВДП, що робить їх застосування доцільним у фінансовому аналізі та стратегічному плануванні.

5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» є оцінювання науково-технічного рівня та рівня

комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [53].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 5.1

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	3	4	3

Продовження таблиці 5.2

7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	3	3	3
9. Практична здійсненність (наявність фінансів)	3	3	4
10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	3	3	3
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	38	38	41
Середньоарифметична сума балів $СБ_c$	39		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [53].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» становить 39 бали, що, відповідно до таблиці 5.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

5.2 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_o$) розраховуємо у відповідності до посадових окладів працівників, за формулою [53]:

$$З_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дні;

T_p – середнє число робочих днів в місяці, $T_p=21$ день.

$$З_o = 24000,00 \cdot 91 / 21 = 99272,73 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	24000	1090,91	91	99272,73
Інженер-розробник програмного забезпечення	20000	909,09	91	82727,27
Консультант в сфері машинного навчання	20000	909,09	30	27272,73
Консультант в сфері інвестицій	20000	909,09	30	27272,73
Всього				236545,45

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт НДР розраховуємо за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.3)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [53];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ день;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000,00 \cdot 1,10 \cdot 1,65 / (21 \cdot 8) = 82,50 \text{ грн.}$$

$$З_{р1} = 82,50 \cdot 8,00 = 660,00 \text{ грн.}$$

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення обладнання	8	2	1,1	82,50	660,00
Підготовка робочого місця	1	2	1,1	82,50	82,50
Інсталяція програмного забезпечення	5	5	1,7	127,50	637,50
Всього					1380,00

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_o + З_p) \cdot \frac{H_{дод}}{100\%}, \quad (5.4)$$

де $H_{дод}$ – норма нарахування додаткової заробітної плати. Прийmemo 12%.

$$З_{дод} = (236545,45 + 1380,00) \cdot 12 / 100\% = 26171,80 \text{ грн.}$$

5.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{дод}) \cdot \frac{H_{zn}}{100\%} \quad (5.5)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (236545,45 + 1380,00 + 26171,80) \cdot 22 / 100\% = 58101,4 \text{ грн.}$$

5.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень.

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{ej}, \quad (5.6)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$Ц_{ej}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3 \cdot 180,00 \cdot 1,1 - 0,000 \cdot 0,00 = 594 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од, грн	Норма витрат, од	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір	180	3	0	0	594
Папір для записів Calipso Parers Light A5	110	1	0	0	121
Канцелярське приладдя (набір офісного працівника)	150	2	0	0	330
Flesh накопичувач	350	1	0	0	385
Всього					1430

5.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» відсутні.

5.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення в даному дослідженні не використовувалось.

5.2.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{прог}} \cdot C_{\text{прог},i} \cdot K_i, \quad (5.7)$$

де $C_{\text{прог}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог},i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 406 \cdot 3 \cdot 1,12 = 1364,16 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.7– Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Інтегроване середовище розробки PyCharm Professional	3	406	1364,16
Всього			1364,16

5.2.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{C_{\text{б}}}{T_{\text{б}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (5.8)$$

де $C_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_e – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (60000,00 \cdot 3) / (5 \cdot 12) = 3000 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук Lenovo, 2шт	60 000	5	3	3000,00
Маршрутизатор Asus	6 000	5	3	300,00
Монітор DELL	6 000	2	3	750,00
Робоче місце дослідника	20000	5	3	1000,00
Оргтехніка	7000	4	3	437,50
Приміщення лабораторії	250000	20	3	3125,00
Хмарна інфраструктура AWS EC2	2463	3	3	205,25
Всього				8817,75

5.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (5.9)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 11$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,012 \cdot 728,0 \cdot 11 \cdot 0,95 / 0,97 = 94,11 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук Lenovo, 2шт	0,17	728*2	2666,58
Маршрутизатор Asus	0,012	728	94,11
Монітор DELL, 1 шт	0,019	728	149,01
Робоче місце дослідника	0,15	1456	2352,87
Оргтехніка	0,45	16	77,57
Всього			5340,14

5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (5.10)$$

де H_{ce} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{ce} = 20\%$.

$$B_{ce} = (236545,45 + 1380,00) \cdot 20 / 100\% = 47585,09 \text{ грн.}$$

5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.11)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 35\%$.

$$B_{cn} = (236545,45 + 1380,00) \cdot 35 / 100\% = 83273,91 \text{ грн.}$$

5.2.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\varepsilon} = (Z_o + Z_p) \cdot \frac{H_{\varepsilon}}{100\%}, \quad (5.12)$$

де H_{ε} – норма нарахування за статтею «Інші витрати», прийmemo $H_{\varepsilon} = 50\%$.

$$I_{\varepsilon} = (236545,45 + 1380,00) \cdot 50 / 100\% = 118962,73 \text{ грн.}$$

5.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з

освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.13)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 120\%$.

$$B_{нзв} = (236545,45 + 1380,00) \cdot 120 / 100\% = 285\,510,55 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = 3_o + 3_p + 3_{доо} + 3_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (5.14)$$

$$B_{заг} = 969082,98 \text{ грн.}$$

Загальні витрати $3B$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$3B = \frac{B_{заг}}{\eta}, \quad (5.15)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,7$.

$$3B = 969082,98 / 0,7 = 1384404,25 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

1-й рік – 100000 користувачів;

2-й рік – 200000 користувачів;

3-й рік – 150000 користувачів.

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 42000000 користувачів;

C_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 2000 грн;

$\pm \Delta C_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo (50,00) грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [53]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.16)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту.

Прийmemo $\rho = 30\%$;

$\mathcal{Q}$ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\mathcal{Q} = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (50 \cdot 42000000,00 + 2050 \cdot 100000) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 472335990 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (50 \cdot 42000000,00 + 2050 \cdot (100000 + 200000)) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 556352370 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (50 \cdot 42000000,00 + 2050 \cdot (100000 + 200000 + 150000)) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 619364655 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.17)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,25$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 472335990/(1+0,25)^1 + 556352370/(1+0,25)^2 + 619364655/(1+0,25)^3 = 1051049012,16 \text{ грн.}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (5.18)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 5$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 1384404,25 грн.

$$PV = k_{инв} \cdot 3B = 5 \cdot 1384404,25 = 6922021,266 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (5.19)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1051049012,16 грн;

PV – теперішня вартість початкових інвестицій, 6922021,266 грн.

$$E_{абс} = ПП - PV = 1051049012,16 - 6922021,266 = 1044126990,89 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_{\epsilon} = T_{ж} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.20)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 1044126990,89 грн;

PV – теперішня вартість початкових інвестицій, 6922021,266 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_{\epsilon} = T_{ж} \sqrt{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 1044126990,89 / 6922021,266)^{1/3} - 1 = 4,33.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (5.21)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{мін} = 0,11 + 0,25 = 0,36 < 4,33$ свідчить про те, що внутрішня економічна дохідність інвестицій E_{ϵ} , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_{\epsilon}}, \quad (5.22)$$

де E_{ϵ} – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 4,33 = 0,23 \text{ року.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

5.3 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації» становить 39 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу вище середнього).

Також термін окупності становить 0,23 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації».

ВИСНОВКИ

В результаті виконання магістерської кваліфікаційної роботи на тему "Розробка методів та програмних засобів для формування портфелю інвестицій в облігації", було підвищено точність прогнозування цін ОВДП, за рахунок додавання до вибірки навчальних даних макроекономічних показників.

В ході дослідження було проведено аналіз програмних засобів для формування портфелю інвестицій в ОВДП.

Було сформовано вимоги до програмного забезпечення для формування портфелю інвестицій. Також було сформовано вимоги до вибірки навчальних даних. Було проведено аналіз методів машинного навчання для прогнозування ринкових тенденцій ОВДП.

Розроблено програмні засоби для побудови вибірки навчальних даних, навчання та використання моделі. Для збору навчальних даних використовувалися API, надані Національним банком України та ПриватБанком. Розроблені програмні засоби дозволили поєднати різні типи даних у єдину вибірку.

Отримала подальшого розвитку архітектура програмного застосунку, в яку, на відміну від існуючих, введено модуль побудови навчальної вибірки даних, що дозволило прискорити процес машинного навчання.

Було виконано тестування розробленої системи. Розроблена система відповідає поставленим функціональним вимогам. Результати тестування підтверджують високу ефективність системи, зокрема значне покращення точності прогнозування завдяки включенню макроекономічних показників. Використання моделі Random Forest у поєднанні з цими даними підвищило точність прогнозування на 11%, що свідчить про поліпшення прогнозної здатності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Молодь в науці: дослідження, проблеми, перспективи (МН-2025) [Електронний ресурс].
<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22283> (дата звернення: 01.10.2024).
2. ОВДП для населення [Електронний ресурс]. URL:
https://mof.gov.ua/uk/domestic_government_bonds_for_population-360.
3. Облігації внутрішньої державної позики України [Електронний ресурс]. URL: <https://privatbank.ua/ovdp>.
4. Інвестиції в Державні Облігації: Ризики та Вигоди [Електронний ресурс]. URL: <https://www.assets.com.ua/posts-afg/investiciyi-v-derzhavni-obligaciyi-riziki-ta-vigodi>.
5. Луцький національний технічний університет. Факультет бізнесу та права. Кафедра обліку і аудиту. Електронний посібник з дисципліни: економіко-статистичні методи та інформаційні технології підтримки управлінських рішень: Лекція 1. «Загальні поняття фінансових економетричних моделей». Луцьк, 2024.
6. What is predictive AI? [Електронний ресурс]. URL:
<https://www.ibm.com/think/topics/predictive-ai>
7. Що таке лінійна регресія? [Електронний ресурс]. URL:
<https://www.unite.ai/uk/what-is-linear-regression/>
8. What are ARIMA models? [Електронний ресурс]. URL:
<https://www.ibm.com/topics/arima-model>
9. Why Is the Vector Autoregressive Model One Of The Best For Time-series Forecasting? [Електронний ресурс]. URL: <https://medium.com/@aliyujimoh58/why-is-the-vector-autoregressive-model-one-of-the-best-for-time-series-forecasting-317649670ea1>

10. What is random forest? [Електронний ресурс]. URL: <https://www.ibm.com/topics/random-forest>
11. Gradient Boosting Explained – The Coolest Kid on The Machine Learning Block [Електронний ресурс]. URL: <https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/>
12. 8 Popular Machine Learning Algorithms for Predictive Modeling [Електронний ресурс]. URL: <https://vmsoftwarehouse.com/8-machine-learning-algorithms-for-predictions>
13. The Top 10 Machine Learning Algorithms to Know [Електронний ресурс]. URL: <https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies>
14. When Econometrics Meets Machine Learning [Електронний ресурс]. URL: <https://www.sciencedirect.com/science/article/pii/S2543925122000900>
15. BondIT: Reimagined Fixed Income Investing [Електронний ресурс]. URL: <https://bonditglobal.com/>.
16. Bloomberg Terminal [Електронний ресурс]. URL: <https://www.bloomberg.com/professional/products/bloomberg-terminal/>.
17. Інвестування в ОВДП [Електронний ресурс]. URL: <https://www.otpbank.com.ua/privateclients/investments/ovdp/>.
18. УНІВЕР Капітал: Інвестиційна Група УНІВЕР [Електронний ресурс]. URL: <https://www.univer.ua/>.
19. The Full Guide to Training Datasets for Machine Learning [Електронний ресурс]. URL: <https://encord.com/blog/an-introduction-to-data-labelling-and-training-data/>
20. The role of accuracy of data in advanced forecast system: An investigative approach [Електронний ресурс]. URL: https://www.researchgate.net/publication/384442100_The_role_of_accuracy_of_data_in_advanced_forecast_system_An_investigative_approach

21. Datasets: Data characteristics [Електронний ресурс]. URL: <https://developers.google.com/machine-learning/crash-course/overfitting/data-characteristics>
22. Ensuring Data Freshness: Why It Matters and How to Achieve It [Електронний ресурс]. URL: <https://datahubproject.io/learn/data-freshness/>
23. Cracking the Code: Understanding and Avoiding Model Overtraining [Електронний ресурс]. URL: <https://www.c-sharpcorner.com/article/cracking-the-code-understanding-and-avoiding-model-overtraining/>
24. Національний банк України. API для розробників [Електронний ресурс]. URL: <https://bank.gov.ua/ua/open-data/api-dev>
25. RandomForestClassifier [Електронний ресурс]. URL: <https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
26. Gradient Boosting for Regression -Math & Intuition [Електронний ресурс]. URL: <https://medium.com/geekculture/gradient-boosting-for-regression-math-intuition-71f9a9826035>
27. Vector Autoregressive models VAR(p) models [Електронний ресурс]. URL: <https://online.stat.psu.edu/stat510/lesson/11/11.2>
28. ARIMA for Time Series Forecasting: A Complete Guide [Електронний ресурс]. URL: <https://www.datacamp.com/tutorial/arima>
29. Linear Regression in Machine learning [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/ml-linear-regression/>
30. Software Architecture Guide [Електронний ресурс]. URL: <https://martinfowler.com/architecture/>
31. What Is a REST API? Examples, Uses, and Challenges [Електронний ресурс]. URL: <https://blog.postman.com/rest-api-examples/>
32. Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. Patterns of Enterprise Application Architecture. Addison Wesley, 2002 p. 560 с.

33. About java. Oracle [Электронный ресурс].
URL: https://www.java.com/en/download/help/whatis_java.html].
34. Bjarne Stroustrup. The C++ programming language. Fourth edition. Addison Wesley, 2013 p. 1346 с.
35. [DUPLICATE Bjarne Stroustrup. The C++ programming language. Fourth edition. Addison Wesley, 2013 p. 1346 с.].
36. C++ Glossary [Электронный ресурс]. URL: <https://stroustrup.com/glossary.html>
37. Popular Python Libraries Boost Testing Efficiency [Электронный ресурс]. URL: <https://testomat.io/blog/popular-python-libraries-to-make-python-testing-more-efficient/>
38. Why Python is the go-to language for machine learning [Электронный ресурс]. URL: <https://onesource.be/why-python-is-the-go-to-language-for-machine-learning/>
39. Python vs C++: Detailed Comparison of Technical Differences and Use Cases [Электронный ресурс]. URL: <https://llego.dev/posts/python-vs-c-plus-plus-comparison/>
40. An Introduction to REST API with Python [Электронный ресурс]. URL: <https://www.integrate.io/blog/an-introduction-to-rest-api-with-python/>
41. 10 Best Cross-Platform Programming Languages [Электронный ресурс]. URL: <https://binarapps.com/10-best-cross-platform-programming-languages/>
42. The Python IDE for data science and web development [Электронный ресурс]. URL: <https://www.jetbrains.com/pycharm/>
43. What is PyCharm? Features, Advantages & Disadvantages [Электронный ресурс]. URL: <https://www.geeksforgeeks.org/what-is-pycharm-features-advantages-disadvantages/>
44. What is Visual Studio? [Электронный ресурс]. URL: [https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-](https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022)

45. Visual Studio vs Visual Studio Code: What's the Key Difference? [Електронний ресурс]. URL: <https://distantjob.com/blog/visual-studio-vs-visual-studio-code/>

46. Eclipse Java Review: Custom Development Environment [Електронний ресурс]. URL: <https://javarush.com/en/groups/posts/en.2359.eclipse-java-review-custom-development-environment>

47. Eclipse ide Reviews [Електронний ресурс]. URL: <https://www.joinsecret.com/eclipse-ide/reviews>

48. Visual Studio Code [Електронний ресурс]. URL: <https://code.visualstudio.com/>

49. What is a Relational Database (RDBMS)? [Електронний ресурс]. URL: <https://www.oracle.com/database/what-is-a-relational-database/>

50. What is PostgreSQL? [Електронний ресурс]. URL: <https://aws.amazon.com/ru/rds/postgresql/what-is-postgresql/>

51. Document-oriented database [Електронний ресурс]. URL: <https://www.techtarget.com/whatis/definition/document-oriented-database>

52. Why Use MongoDB and When to Use It? [Електронний ресурс]. URL: <https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb>

53. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

ДОДАТКИ

Додаток А - Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
" 19 " 09 2024 р.

Технічне завдання
на магістерську кваліфікаційну роботу
«Розробка методів та програмних засобів для формування портфелю
інвестицій в облігації. »
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

к.т.н., доц. каф. ПЗ . О.М. Ткаченко

« 19 » 09 2024 р.

Виконав:

студент гр. ІПІ-23м Р.О. Костюк

« 19 » 09 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації».

Галузь застосування – формування портфеля інвестицій в облігації.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ №___ від 18 вересня 2024 року ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення точності прогнозування тенденцій змін ринкових цін ОВДП на основі машинного навчання за рахунок додавання до вибірки навчальних даних макроекономічних показників. Призначення роботи – розробка методів і засобів прогнозування тенденцій цін облігацій.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Інвестиції в Державні Облігації: Ризики та Вигоди [Електронний ресурс]. URL: <https://www.assets.com.ua/posts-afg/investiciyi-v-derzhavni-obligaciyi-riziki-ta-vigodi>.

2. Луцький національний технічний університет. Факультет бізнесу та права. Кафедра обліку і аудиту. Електронний посібник з дисципліни: економіко-статистичні методи та інформаційні технології підтримки управлінських рішень: Лекція 1. «Загальні поняття фінансових економетричних моделей». Луцьк, 2024.

3. What is predictive AI? [Електронний ресурс]. URL: <https://www.ibm.com/think/topics/predictive-ai>

4. What is random forest? [Електронний ресурс]. URL: <https://www.ibm.com/topics/random-forest>

5. The Full Guide to Training Datasets for Machine Learning [Електронний ресурс]. URL: <https://encord.com/blog/an-introduction-to-data-labelling-and-training-data/>

6. The role of accuracy of data in advanced forecast system: An investigative approach [Електронний ресурс]. URL: https://www.researchgate.net/publication/384442100_The_role_of_accuracy_of_data_in_advanced_forecast_system_An_investigative_approach

5. Технічні вимоги

Вихідні дані до роботи: модель розробки – водоспадна; метод передачі повідомлень між сервісами – HTTP; система управління базами даних – MongoDB; вхідні дані – база даних з кешованими навчальними даними; вихідні дані – макроекономічні показники, цінові показники облігацій; середовище розробки – PyCharm; мова програмування – Python.

6. Конструктивні вимоги.

Розроблене ПЗ повинне бути реалізоване відповідно до сучайних парадигм. Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

Додаток Б - Протокол перевірки навчальної роботи

Протокол перевірки МКР на плагіат

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: **Розробка методів та програмних засобів для формування портфелю інвестицій в облігації.**

Тип роботи: кваліфікаційна робота

Підрозділ : кафедра програмного забезпечення, ФІТКІ, ІПП – 23м

Науковий керівник: к.т.н., доц. каф. ПЗ Ткаченко. О. М.

Turnitin	
Оригінальність	95,0 %
Схожість	5,0%

Аналіз звіту подібності

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою щодо роботи «Розробка методів та програмних засобів для формування портфелю інвестицій в облігації».

Автор Костюк Руслан Олександрович

Костюк Руслан Олександрович

Опис прийнятого рішення: **допустити до захисту**

Особа, відповідальна за перевірку

(підпис)

Г. О. Черноволик

(ініціали, прізвище)

Експерт

(за потреби) (підпис)

(ініціали, прізвище)

Додаток В – Лістинг програми

Лістинг основних алгоритмів програми

```
def training_demo():
    logging.basicConfig()
    logging.getLogger().setLevel(logging.INFO)

    start_date = date(2023, 7, 3)
    end_date = date(2024, 9, 4)
    data_loader = TrainingDataLoaderContext(start_date, end_date)
    data_loader.load_training_data()

    random_forest_prediction_demo(start_date, end_date)

def random_forest_prediction_demo(start_date, end_date):
    # Example: Load your training data into a DataFrame (replace with actual data
loading logic)
    dataset_builder = TrainingDatasetBuilder()
    data = dataset_builder.build(start_date, end_date)

    df = pd.DataFrame(data)

    # Preprocessing
    # Convert numeric columns to floats and handle missing values
    def preprocess_dataframe(df):
        for col in df.columns:
            if col.endswith("_ask") or col.endswith("_bid"):
```

```

df[col] = pd.to_numeric(df[col], errors='coerce')

# Fill missing bond prices with forward fill (or use other imputation strategies)
df.fillna(method='ffill', inplace=True)
df.fillna(method='bfill', inplace=True)

# Convert date to datetime and sort
df['Date'] = pd.to_datetime(df['Date'])
df.sort_values('Date', inplace=True)

return df

df = preprocess_dataframe(df)

# Feature Engineering
# Create lag features for bond prices
lag_days = 3 # Number of past days to consider

def create_lag_features(df, lag_days):
    lagged_df = df.copy()
    for lag in range(1, lag_days + 1):
        for col in df.columns:
            if col.endswith("_ask") or col.endswith("_bid"):
                lagged_df[f"{col}_lag{lag}"] = df[col].shift(lag)

# Drop rows with NaN values introduced by lagging
lagged_df.dropna(inplace=True)
return lagged_df

```

```

df = create_lag_features(df, lag_days)

# Define features and target
feature_cols = [col for col in df.columns if
                 not col.endswith("_ask") and not col.endswith("_bid") and col !=
                 "Date"]

target_cols = [col for col in df.columns if col.endswith("_ask") or
               col.endswith("_bid")]

# Remove datetime column from features
feature_cols = [col for col in feature_cols if df[col].dtype in ['int64', 'float64']]

X = df[feature_cols]
y = df[target_cols]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train Random Forest model
models = {}
for target in target_cols:
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X_train, y_train[target])
    models[target] = model

# Predict on test data

```

```

predictions = {}
for target, model in models.items():
    predictions[target] = model.predict(X_test)

# Evaluate model performance
rmse_results = {}
for target in target_cols:
    rmse = np.sqrt(mean_squared_error(y_test[target], predictions[target]))
    rmse_results[target] = rmse
    print(f"RMSE for {target}: {rmse:.2f}")

# Calculate and print average RMSE
average_rmse = np.mean(list(rmse_results.values()))
print(f"Average RMSE: {average_rmse:.2f}")

# Visualizations
# 1. RMSE Bar Plot
plt.figure(figsize=(12, 6))
sns.barplot(x=list(rmse_results.keys()), y=list(rmse_results.values()))
plt.title("RMSE for Each Target Variable")
plt.xlabel("Target Variable")
plt.ylabel("RMSE")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# 2. Predicted vs Actual Scatter Plots
for target in target_cols:

```

```

plt.figure(figsize=(8, 6))
plt.scatter(y_test[target], predictions[target], alpha=0.6)
plt.title(f"Predicted vs Actual for {target}")
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.plot([min(y_test[target]),      max(y_test[target])],      [min(y_test[target]),
max(y_test[target])],
         'r--') # Reference line
plt.tight_layout()
plt.show()

```

3. Residuals Distribution

```

for target in target_cols:
    residuals = y_test[target] - predictions[target]
    plt.figure(figsize=(8, 6))
    sns.histplot(residuals, kde=True, bins=30)
    plt.title(f"Residual Distribution for {target}")
    plt.xlabel("Residuals")
    plt.ylabel("Frequency")
    plt.tight_layout()
    plt.show()

```

Forecast future days

```

def forecast_future(df, models, days_to_forecast):
    future_predictions = []
    last_known_data = df.iloc[-1]

    for _ in range(days_to_forecast):

```

```

new_data = last_known_data.copy()

for target, model in models.items():
    features = new_data[feature_cols].values.reshape(1, -1)
    new_data[target] = model.predict(features)[0]

future_predictions.append(new_data[target_cols])
last_known_data = new_data

return pd.DataFrame(future_predictions)

forecast_days = 5
future_df = forecast_future(df, models, forecast_days)
print("Future predictions:")
print(future_df)

def compute_average_cpi(price_indices):
    values = [cpi['value'] for cpi in price_indices]
    return fmean(values)

def map_cpi(cpi):
    return {
        'date': cpi['date'],
        'avg_cpi': compute_average_cpi(cpi['price_indices'])
    }

def map_interest_rate(interest_rate):
    return {

```

```

    "date": interest_rate['date'],
    "interest_rate": interest_rate['KEY_PolicyRate']['value'],
}

```

```

def combine(cpis, exchange_rates, interest_rates, bond_prices):
    if len(exchange_rates) != len(interest_rates):
        raise ValueError("exchange_rates and interest_rates must have the same
number of records.")

```

```

    cpi_map = {
        (cpi['date'].year, cpi['date'].month): cpi for cpi in cpis
    }

```

```

    exchange_rate_map = {exchange_rate['date']: exchange_rate for exchange_rate
in exchange_rates}

```

```

    interest_rate_map = {interest_rate['date']: interest_rate for interest_rate in
interest_rates}

```

```

    bond_price_map = {format_bond_price_date(bond_price['Date']): bond_price
for bond_price in bond_prices}

```

```

    # return bond_price_map

```

```

combined_data = []

```

```

for date in bond_price_map:

```

```

    exchange_rate = exchange_rate_map[date]

```

```

    interest_rate = interest_rate_map.get(date)

```

```

    bond_price = bond_price_map.get(date)

```

```

    if not interest_rate:

```

```

        raise ValueError(f"No interest rate record found for date {date}.")

    year = date.year
    month = date.month

    cpi = cpi_map.get((year, month))
    if not cpi:
        raise ValueError(f"No CPI record found for year {year} and month
{month}.")

    combined_record = cpi | exchange_rate | interest_rate | bond_price
    combined_data.append(combined_record)

    return combined_data

def format_bond_price_date(bond_price_date):
    date_parts = bond_price_date.split('/')
    day, month, year = map(int, date_parts)
    return datetime.combine(date(2000 + year, month, day), datetime.min.time())

class TrainingDatasetBuilder:

    def __init__(self):
        self.bond_prices = []
        self.cpi_repository = CpiRepository()
        self.exchange_rate_repository = ExchangeRateMongoRepository()
        self.interest_rate_repository = InterestRateMongoRepository()
        self.bond_prices_reader = BondPricesReader()

```

```

self.logger = logging.getLogger(type(self).__name__)

def build(self, start_date, end_date):
    start_datetime = date_to_datetime(start_date)
    end_datetime = date_to_datetime(end_date)

    cpis = list(self.cpi_repository.get_within_date_range(start_datetime,
end_datetime))
    avg_cpis = [map_cpi(cpi) for cpi in cpis]

    exchange_rates = list(self.exchange_rate_repository.get_within_date_range(start_datetime, end_datetime))
    interest_rates = [map_interest_rate(rate) for rate in
self.interest_rate_repository.get_within_date_range(start_datetime, end_datetime)]
    bond_prices = self.bond_prices_reader.read_bond_prices()

    return combine(avg_cpis, exchange_rates, interest_rates, bond_prices)

class BondPricesReader:

    def __init__(self):
        pass

    def read_bond_prices(self):
        # path = "bonds_with_changes.csv"
        path = "bonds.csv"
        with open(path, "r", encoding="utf-8") as f:
            reader = csv.DictReader(f)

```

```
return list(reader)
```

```
class TrainingDataLoaderContext:
```

```
    def __init__(self, start_date: date, end_date: date):  
        self.training_data_loaders = [  
            ExchangeRateDataLoader(start_date, end_date),  
            InterestRateDataLoader(start_date, end_date),  
            SecuritiesDataLoader(),  
            CpiDataLoader(start_date, end_date)  
        ]  
  
    def load_training_data(self):  
        for loader in self.training_data_loaders:  
            loader.load_if_absent()
```

Додаток Г – Ілюстрована частина

**Магістерська кваліфікаційна робота
на тему:**

**«Розробка методів та програмних засобів для формування
портфелю інвестицій в облігації»**

Виконав:

Студент групи 1ПІ-23м

Костюк Р. О.

Науковий керівник:

к. т. н., доц. каф. ПЗ

Ткаченко О. М.

Рисунок Г.1 – Титульний слайд

Наукова новизна

Удосконалено метод машинного навчання для прогнозування цінових тенденцій ОВДП, в якому, на відміну від існуючих методів, було використано дані макроекономічних показників в поєднанні з даними цін ОВДП за певний період часу для навчання, що дозволило підвищити достовірність прогнозу ринкових цін ОВДП.

Отримала подальшого розвитку архітектура програмного застосунку, в яку, на відміну від існуючих, введено модуль побудови навчальної вибірки даних, що дозволило прискорити процес машинного навчання.

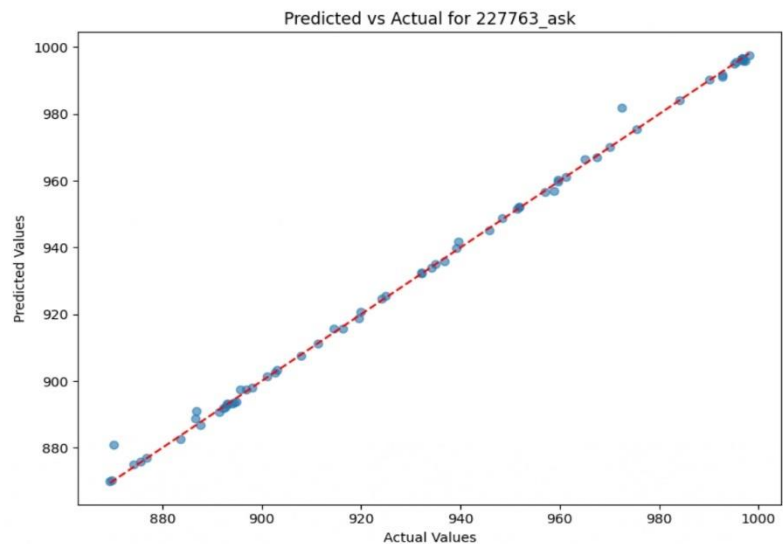


Рисунок Г.2 – Наукова новизна

Мета і задачі роботи

Метою роботи є підвищення точності прогнозування тенденцій змін ринкових цін облігацій на основі машинного навчання за рахунок додавання до вибірки навчальних даних макроекономічних показників.

Основними задачами дослідження є:

- провести аналіз існуючих методів і засобів машинного навчання для прогнозування тенденцій ринкових цін ОВДП;
- вдосконалити методи машинного навчання для прогнозування тенденцій ринкових цін ОВДП за наявності вибірки даних, що включає ринкову вартість ОВДП та макроекономічні показники за певний період часу;
- розробити програмні засоби для побудови вибірки навчальних даних, що включатиме як дані ринкової вартості ОВДП, так і макроекономічні показники;
- розробити програмні компоненти та систему візуалізації на основі запропонованих методів;
- провести експериментальні дослідження розроблених засобів прогнозування вартості ОВДП.

Рисунок Г.3 – Мета і задачі роботи

Об'єкт та предмет дослідження

Об'єкт дослідження – процес прогнозування ринкових тенденцій цін ОВДП в комп'ютерних системах.

Предмет дослідження – методи та засоби прогнозування ринкових тенденцій цін ОВДП.

Практична цінність

Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби прогнозування ринкових цін ОВДП для формування портфелю інвестицій.

Рисунок Г.4 – Об'єкт та предмет дослідження

Актуальність теми

Актуальність дослідження зумовлена необхідністю вдосконалення методів прогнозування ринкових тенденцій цінних паперів, зокрема цін облігацій внутрішньої державної позики (ОВДП).

Вдале формування портфеля інвестицій постає важливим для мінімізації ризиків та максимізації прибутку. З огляду на складність ринку і кількість факторів, що впливають на дохідність ОВДП, стає постає потреба у передбаченні майбутніх змін цін.

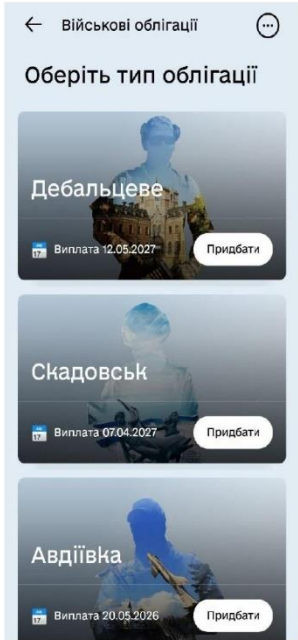


Рисунок Г.5 – Актуальність теми

Порівняльний аналіз аналогів

Конкурентоспроможне програмне забезпечення для формування портфеля інвестицій в ОВДП може відповідати наступним характеристикам:

- простота у використанні
- автоматизація процесів
- прогнозування цінових тенденцій ринку
- мультиплатформність
- наявність аналітичних інструментів
- доступність

Назва платформи	Простота використання	Аналітичні інструменти	Вартість	Автоматизація	Мультиплатформність
iBonds	Висока	Обмежені	Середня	Ні	Так
BondIT	Середня	Розширені	Висока	Так	Так
Bloomberg Terminal	Низька	Найпотужніші	Дуже висока	Так	Так
OTP Bank ОВДП	Висока	Обмежені	Низька	Ні	Так
Універ Капітал	Висока	Обмежені	Низька	Ні	Ні

Рисунок Г.6 – Порівняльний аналіз аналогів

Архітектура системи

Отримала подальшого розвитку архітектура програмного застосунку, в яку, на відміну від існуючих, введено модуль побудови навчальної вибірки даних, що дозволило прискорити процес машинного навчання.

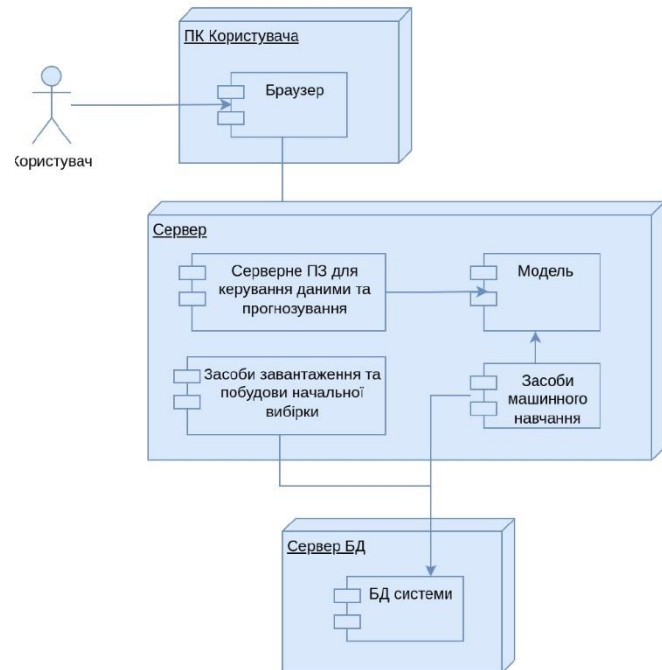


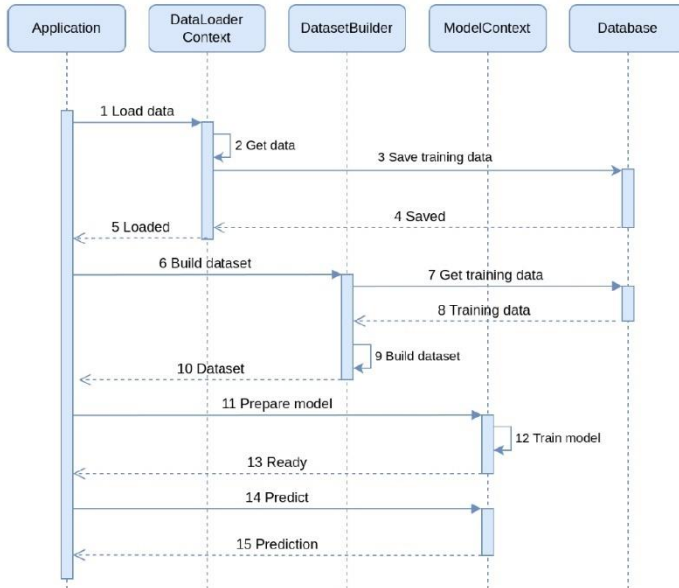
Рисунок Г.7 – Архітектура системи



UML-діаграма станів системи

- Завантаження навчальних даних
- Побудова навчальної вибірки
- Навчання моделі
- Очікування вхідних даних
- Прогнозування
- Виведення результату
- Обробка помилок

Рисунок Г.8 – UML-діаграма станів системи



Головна UML-діаграма послідовності функціонування системи

Рисунок Г.9 – Головна UML-діаграма послідовності

Складові навчальної вибірки

```

{
  "date": "2023-07-03 00:00:00",
  "avg_cpi": 10.38409090909091,
  "saleRateNB": 36.5686,
  "purchaseRateNB": 36.5686,
  "interest_rate": 25,
  "Date": "03/07/23",
  "227763_ask": "868.75",
  "227763_bid": "862.53",
  "226286_ask": "970.02",
  "226286_bid": "962.31"
  ...
}

```

Дата
Індекс споживчих цін
Курс купівлі долару США
Курс продажу долару США
Облікова ставка НБУ
Курс купівлі ОВДП 1
Курс продажу ОВДП 1
...
Курс купівлі ОВДП n
Курс продажу ОВДП n

Рисунок Г.10 – Складові навчальної вибірки

UML-діаграма послідовності завантаження навчальних даних

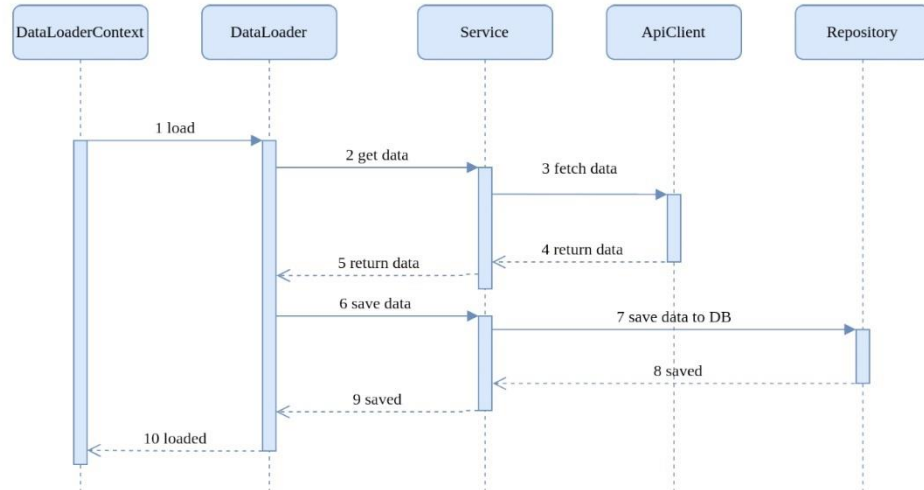
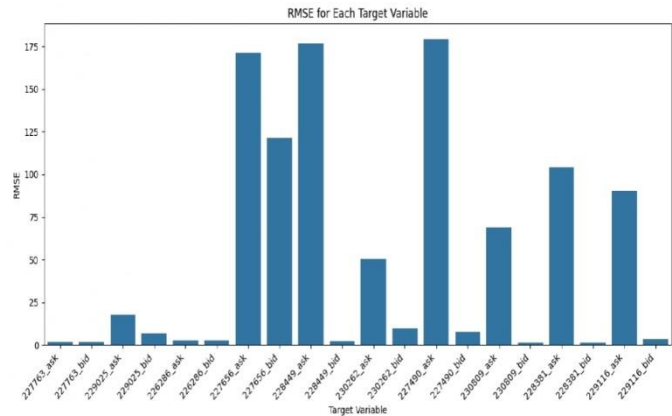
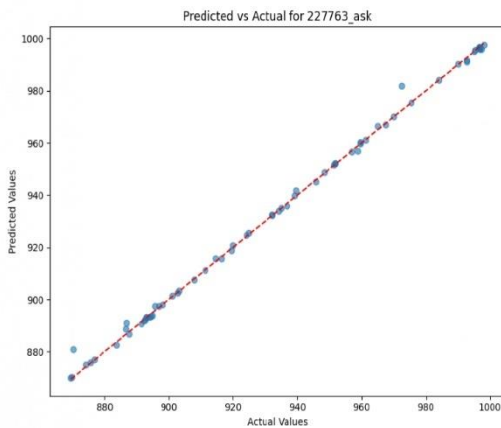


Рисунок Г.11 – UML-діаграма послідовності завантаження даних

Тестування системи



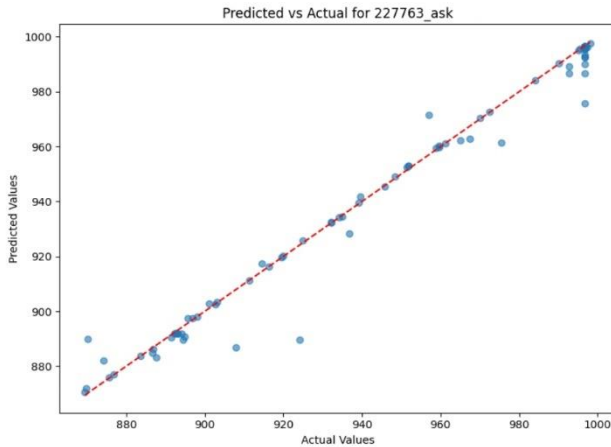
```

Future predictions:
227763_ask 227763_bid 229025_ask ... 228381_bid 229116_ask 229116_bid
406 996.63 996.6 1000.0 ... 1103.9693 1156.0694 1135.803
406 996.63 996.6 1000.0 ... 1103.9693 1156.0694 1135.803
406 996.63 996.6 1000.0 ... 1103.9693 1156.0694 1135.803
406 996.63 996.6 1000.0 ... 1103.9693 1156.0694 1135.803
406 996.63 996.6 1000.0 ... 1103.9693 1156.0694 1135.803

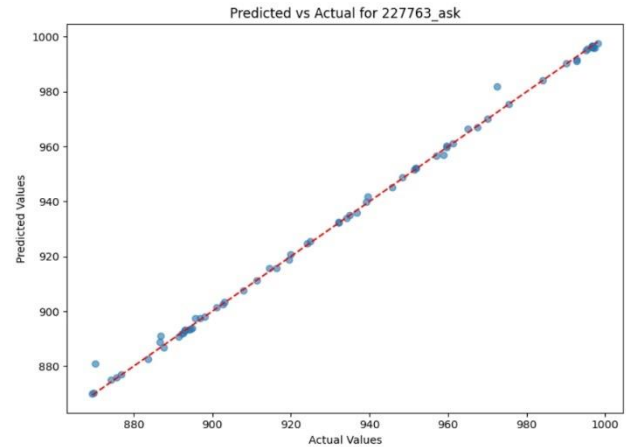
[5 rows x 20 columns]
  
```

Рисунок Г.12 – Тестування системи

Порівняння графіків поточних та прогнозованих значень



За відсутності макроекономічних даних



За наявності макроекономічних даних

Середнє значення середньоквадратичної похибки (RMSE) для моделі з макроекономічними показниками знизилося з 57.85 до 51.08, що є на 11% кращим результатом порівняно з моделлю без їх використання.

Рисунок Г.13 – Порівняння графіків поточних та прогнозованих значень

Висновки

В результаті виконання магістерської кваліфікаційної роботи, було підвищено точність прогнозування цін ОВДП на 11%, за рахунок додавання до вибірки навчальних даних макроекономічних показників.

В ході дослідження було проведено аналіз програмних засобів для формування портфеля інвестицій в ОВДП.

Було сформовано вимоги до програмного забезпечення для формування портфеля інвестицій. Також було сформовано вимоги до вибірки навчальних даних. Було проведено аналіз методів машинного навчання для прогнозування ринкових тенденцій ОВДП.

Розроблено програмні засоби для побудови вибірки навчальних даних, навчання та використання моделі. Для збору навчальних даних використовувалися API, надані Національним банком України та ПриватБанком.

Отримала подальшого розвитку архітектура програмного застосунку, в яку, на відміну від існуючих, введено модуль побудови навчальної вибірки даних, що дозволило прискорити процес машинного навчання.

Було виконано тестування розробленої системи. Розроблена система відповідає поставленим функціональним вимогам.

Рисунок Г.14 – Висновки