

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

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

**«Розробка симулятора для роботи з облігаціями
внутрішньої державної позики. Аналіз фінансової
грамотності та прогнозування поведінки користувача.»**

Виконав студент групи ПЗм-22-1
ЛАБЕНСЬКИЙ Владислав Анатолійович

Керівник роботи:
АНТОНЮК Дмитро Сергійович

Рецензент:
МОРОЗОВ Андрій Васильович

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

«ЗАТВЕРДЖУЮ»

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

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛАБЕНСЬКИЙ Владислав Анатолійович**

Керівник роботи: **АНТОНЮК Дмитро Сергійович**

Тема роботи: **«Розробка симулятора для роботи з облігаціями внутрішньої державної позики. Аналіз фінансової грамотності та прогнозування поведінки користувача.»**,

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

Вихідні дані для роботи: предметом дослідження є методи обробки даних і моделі машинного навчання для прогнозування поведінки користувачів; об'єктом дослідження є використання моделей машинного навчання у проведенні аналізу взаємодії користувачів з симулятором та прогнозуванням їхньої поведінки у роботі з облігаціями.

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

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

Дата видачі завдання: **«13» вересня 2023 р.**

РЕФЕРАТ

Завдання кваліфікаційної магістерської роботи полягало у відборі моделей машинного навчання, проведенні аналізу, дослідженні та обґрунтуванні вибору моделей, що були використані під час розробки додаткового модуля задля розширення функціоналу симулятора роботи з облігаціями внутрішньої державної позики. Програмний комплекс було розроблено та вдосконалено так, щоб він працював з використанням технологій проведення аналізу даних та методів машинного навчання, а саме методу класифікації, що використовується для прогнозування поведінки користувачів у роботі з облігаціями внутрішньої державної позики.

Під час написання кваліфікаційної роботи було визначено та проаналізовано найбільш популярні алгоритми машинного навчання. Зроблено опис їх роботи для розуміння, як вони працюють і для яких задач вони найбільше підходять.

Для обраних алгоритмів машинного навчання було проведено дослідження їх якості роботи на власному наборі даних. Унаслідок чого, було визначено найбільш відповідний алгоритм для обраної теми кваліфікаційної роботи. Обґрунтовано, чому надано перевагу алгоритмові випадкових лісів серед решти алгоритмів.

Проведено тестування розробленого комплексу, що в свою чергу дало можливість підтвердити правильність та доскональність вибору алгоритму машинного навчання, щоб уникнути непередбачуваної поведінки додатку.

Кваліфікаційна робота складається з програмного та теоретичного комплексу системи та пояснювальної записки. Пояснювальна записка до кваліфікаційної магістерської роботи нараховує 70 сторінок друкованого тексту, 36 ілюстрацій, 5 таблиць, 43 джерела інформації.

КЛЮЧОВІ СЛОВА: СИМУЛЯТОР, ОБЛІГАЦІЇ, ІНВЕСТИЦІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, МОДЕЛЬ, АНАЛІТИКА ДАНИХ, ПРОГНОЗУВАННЯ.

ABSTRACT

The task of the master's thesis involved selecting machine learning models, conducting analysis, researching and justifying the choice of models used in the development of an additional module to expand the functionality of a simulator for working with domestic government bond obligations. The software complex was developed and improved to operate using data analysis technologies and machine learning methods, specifically a classification method used for predicting user behavior in working with domestic government bond obligations.

During the preparation of the thesis, the most popular machine learning algorithms were identified and analyzed. Descriptions of their workings were provided to understand how they operate and for which tasks they are most suited.

For the selected machine learning algorithms, a study of their performance quality was conducted using a proprietary dataset. As a result, the most suitable algorithm for the chosen thesis topic was identified. A justification was provided for the preference given to the random forest algorithm over other algorithms.

Testing of the developed complex was conducted, which in turn allowed for the confirmation of the correctness and comprehensiveness of the chosen machine learning algorithm, to avoid unpredictable behavior of the application.

The thesis consists of both a programmatic and theoretical system complex, along with an explanatory note. The explanatory note for the master's thesis comprises 70 pages of printed text, 36 illustrations, 5 tables, and references 43 sources of information.

KEYWORDS: SIMULATOR, BONDS, INVESTMENTS, ARTIFICIAL INTELLIGENCE, MODEL, DATA ANALYTICS, PREDICTIONS.

ЗМІСТ

РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1 Постановка завдання	10
1.2 Аналіз аналогів програмного продукту	10
1.3 Аналіз методів машинного навчання.....	14
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	23
Висновки до першого розділу	25
РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ КОРИСТУВАЧІВ	27
2.1 Варіанти використання системи	27
2.2 Обробка даних перед проведенням аналізу	29
2.3 Аналіз вхідних даних	33
2.4 Вибір, розробка та навчання моделі для прогнозування	42
Висновки до другого розділу	47
РОЗДІЛ 3 ДОСЛІДЖЕННЯ ПРАЦЕЗДАТНОСТІ ТА ОЦІНКИ ЕФЕКТИВНОСТІ МОДЕЛІ ПРОГНОЗУВАННЯ	49
3.1 Оцінка та аналіз якості моделі	49
3.2 Тестування роботи моделі машинного навчання в процесі прогнозування	51
Висновки до третього розділу	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	58
ДОДАТКИ.....	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ОВДП – облігації внутрішньої державної позики

ШІ – Штучний інтелект

МН – Машинне навчання

API – Application Programming Interface

JSON – JavaScript Object Notation

CSV – Comma-separated values

ВСТУП

Актуальність теми.

Зараз Інтернет став однією із потреб у нашому повсякденному житті, сприяючи легкості, швидкості та простоті. Використання Інтернету надає нам доступ до фактів, цифр, інформації та знань, що сприяє особистому, соціальному, розумовому та економічному зростанню. Останнім часом веб-додатки пройшли процес значної трансформації, а їх розробка розвивається навіть швидше, ніж очікувалося. Потреба користувачів також зростає з кожним днем.

В свою чергу – це свідчить про те, що користувачі шукають в інтернеті інтелектуальні та інноваційні веб-додатки, які не лише надають управління даними та контентом, але й пропонують розвиток, навчання та вдосконалення власних знань та вмінь.

Також сучасний світ зазнає значних змін і викликів, пов'язаних з швидким розвитком технологій та постійно зростаючим обсягом інформації про фінанси та інвестиції. Фінансова грамотність стає критично важливою для всіх верств суспільства. Особливо актуальним стає забезпечення належного рівня фінансової грамотності у роботі з таким інструментом, як облігації внутрішньої державної позики.

Однією із невирішених проблем є необхідність створення ефективного інструменту, що допоможе звичайним людям, які ніколи не працювали з ОВДП, інвесторам, фахівцям з фінансової сфери розуміти складність процесів управління облігаціями внутрішньої державної позики, аналізувати ризики і можливості. Симулятор може стати потужним інструментом для досягнення цієї мети, сприяючи розвитку фінансової грамотності та зростанню інтересу до інвестування в облігації.

Також, із збільшенням обсягу доступної інформації про фінанси та зростанням ролі штучного інтелекту, можливості аналізу даних стають все більш значущими. Аналітика дозволяє здійснювати індивідуальний підхід до

кожного із користувачів, надаючи персоналізовані рекомендації, прогнози індивідуально. Використання штучного інтелекту та аналітики стає необхідним інструментом у створенні персоналізованих навчальних програм, підвищенні ефективності процесу навчання та передбачення поведінки користувачів у різних часових проміжках.

Відштовхуючись від вищезазначених потреб розвитку інноваційних підходів до розвитку фінансової грамотності громадян та прогнозування їхньої поведінки, метою даної кваліфікаційної роботи є розробка симулятора для роботи з облігаціями внутрішньої державної позики, а також створення системи аналізу даних та моделей машинного навчання для прогнозування поведінки користувачів на основі даних. Цей проєкт буде спрямований на створення інструменту який дасть змогу збирати, зберігати, обробляти дані засобами розробленого симулятора та за допомогою моделей машинного навчання і штучного інтелекту прогнозувати поведінку користувачів і як вони себе поведуть у відповідний момент.

Для досягнення поставленої мети, будуть використані сучасні методи розробки програмного забезпечення, аналізу даних, моделювання та прогнозування.

Мета дослідження полягає в розробці рішення, яке буде використане для аналізу та прогнозування поведінки користувачів з використанням моделей штучного інтелекту.

Завдання кваліфікаційної роботи:

- Провести відбір моделей машинного навчання, які є найперспективнішими для використання у розроблюваному модулі;
- Провести аналіз та дослідження характеристик кожної моделі для визначення їхньої точності та швидкості роботи;
- Розробити функціонал для аналітичного процесу, який дозволить зрозуміти поведінку користувачів при взаємодії із симулятором;

- Розробити прогнозувальну модель для побудови прогнозів майбутніх фінансових рішень користувачів.

Об'єкт дослідження – це використання моделей машинного навчання у проведенні аналізу взаємодії користувачів з симулятором та прогнозуванням їхньої поведінки у роботі з облігаціями.

Предмет дослідження – це методи обробки даних і моделі машинного навчання для прогнозування поведінки користувачів з облігаціями внутрішньої державної позики.

Практичне значення кваліфікаційної роботи:

- вдосконалення наявного веб-симулятор;
- розробка нового функціоналу для симулятора, що дозволить прогнозувати поведінку користувачів з використанням машинного навчання;
- вдосконалення наявного функціоналу симулятора можливістю проведення аналізу, що дасть змогу бути більш обізнаними про знання користувачів та їх поведінку у певних часових проміжках;
- використання розробленого функціоналу в сфері економіки.

Апробація результатів роботи: Матеріали дослідження за тематикою кваліфікаційної роботи було опубліковано у Всеукраїнській науково-практичній конференції молодих вчених «Наукова молодь-2022» 15 листопада 2022р. [1], у V Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», 01–02 грудня 2022 р. [2], а також опублікована стаття у «Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки»[3].

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Постановка завдання

У відповідь на сучасні виклики поточного фінансового ринку та його доступність, є важливим створити умови здобуття нових знань та практичного досвіду у сфері облігацій. Освіченість людей у цій сфері та вміння навчатися на основі реальних даних є ключовими можливостями для успішного управління власним інвестиційним портфелем з метою прибутку.

Саме тому розробка модуля, що буде вдосконаленням симулятора для роботи з облігаціями внутрішньої державної позики включає в себе завдання:

1. **Вибір моделей машинного навчання:** необхідно визначити та відібрати алгоритми, які є максимально ефективними для прогнозування поведінки користувачів.
2. **Порівняльний аналіз моделей:** необхідно провести аналіз обраних моделей для оцінки їх точності та ефективності. Цей процес включає вивчення їх здатності адаптуватися до нових даних, реагувати на зміни в даних і бути здатними до роботи без перенавчання.
3. **Проектування функціоналу аналізу поведінки користувачів:** потрібно розробити інструменти, які зможуть збирати, обробляти та аналізувати дані про дії користувачів у симуляторі з метою розуміння їхньої взаємодії з ОВДП.
4. **Реалізація систему прогнозування поведінки користувачів:** після вибору моделі машинного навчання необхідно розробити систему, яка буде використовувати обрану модель для автоматизованого прогнозування поведінки користувачів.

1.2 Аналіз аналогів програмного продукту

Симуляція – це процес імітація реального процесу або ситуації. Існує досить велика різноманітність симуляторів, які використовуються з різними цілями та потребами:

- навчання;
- тренування;
- тестування певних процесів;
- розважальних цілей

Симулятори найчастіше використовуються в ситуаціях, де взаємодія з реальним об'єктом є неможливою, ризикованою, або пов'язана з великими затратами.

Перейдемо до порівняння аналогів і перший аналог, який ми будемо розглядати – це «Investopedia Stock Simulator». Онлайн-платформа, що дозволяє користувачам експериментувати з віртуальною торгівлею акціями без реальних фінансових ризиків. Основний принцип, що цей симулятор надає користувачам можливість купувати і продавати акції з використанням віртуальних грошей на основі реальних котирувань ринку. Цільовою аудиторією є як новачки, так і досвідчені інвестори, які хочуть випробувати нові стратегії або отримати початковий досвід без реальних витрат.

Наявний функціонал:

1. Реальні котирування: Ціни акцій у симуляторі відображають реальний стан ринку.
2. Конкурси: Користувачі можуть брати участь у віртуальних торгових конкурсах, що робить взаємодію більш захоплюючою.
3. Групи: Користувачі можуть створювати і приєднуватися до груп, щоб конкурувати з друзями або колегами.

Переваги:

1. Безпечне середовище для навчання: Користувачі можуть вільно експериментувати без ризику втрати реальних коштів.
2. Інформація про акцію: Гравці мають необхідну інформацію про акції, бачать графіки росту та падіння акцій.
3. Велика кількість освітніх ресурсів: Завдяки великій кількості статей та навчальних матеріалів на «Investopedia», користувачі можуть глибоко досліджувати та вивчати теми і практикуватися на платформі.

Недоліки:

1. Обмеженість інструментів: Основний акцент зроблений на акційному ринку, інші інвестиційні інструменти відсутні або мало представлені.
2. Не повністю імітує реальність: Хоча ціни акцій відображають реальний ринок, деякі реальні витрати або податки є відсутніми в симуляції.

Підсумовуючи, цей аналог і має ряд переваг, але він має також ряд недоліків, а саме що цей симулятор розроблений для навчання працювати з торгівлею акціями. В нашому випадку нам потрібно, щоб це була робота із ОВДП. Також ця платформа не має змогу для аналізу даних про користувачів та прогнозування поведінки користувачів на ринку, а в даному випадку ми робимо акцент саме на такий функціонал у нашому симуляторі.

Можливо, для адміністраторів платформи наявний такий функціонал, але про це нам не відомо і такий функціонал для користувачів ніде не використано в цьому аналогові.

В свою чергу цей додаток було взято до уваги і також в майбутньому, у нашому симуляторі додатково до облігацій можуть з'явитись акції, навчальні матеріали та групи і для них також буде наявний аналіз та побудова прогнозів.

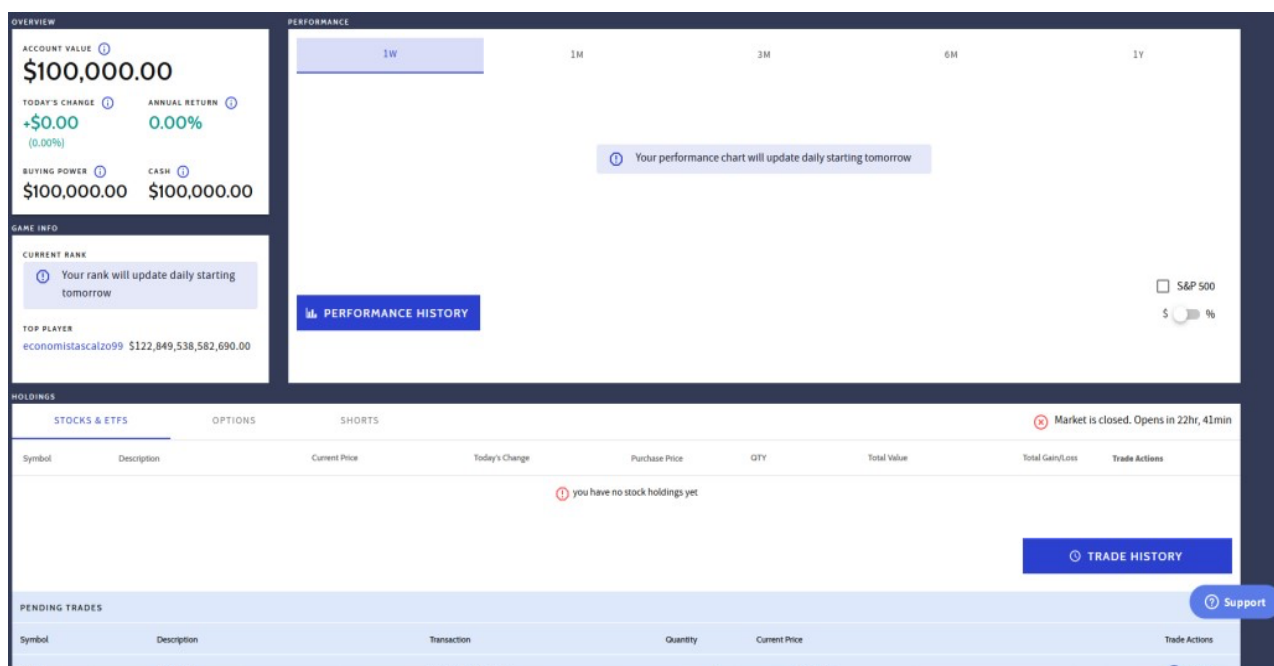


Рис. 1.2.1 Головний екран «Investopedia Stock Simulator»

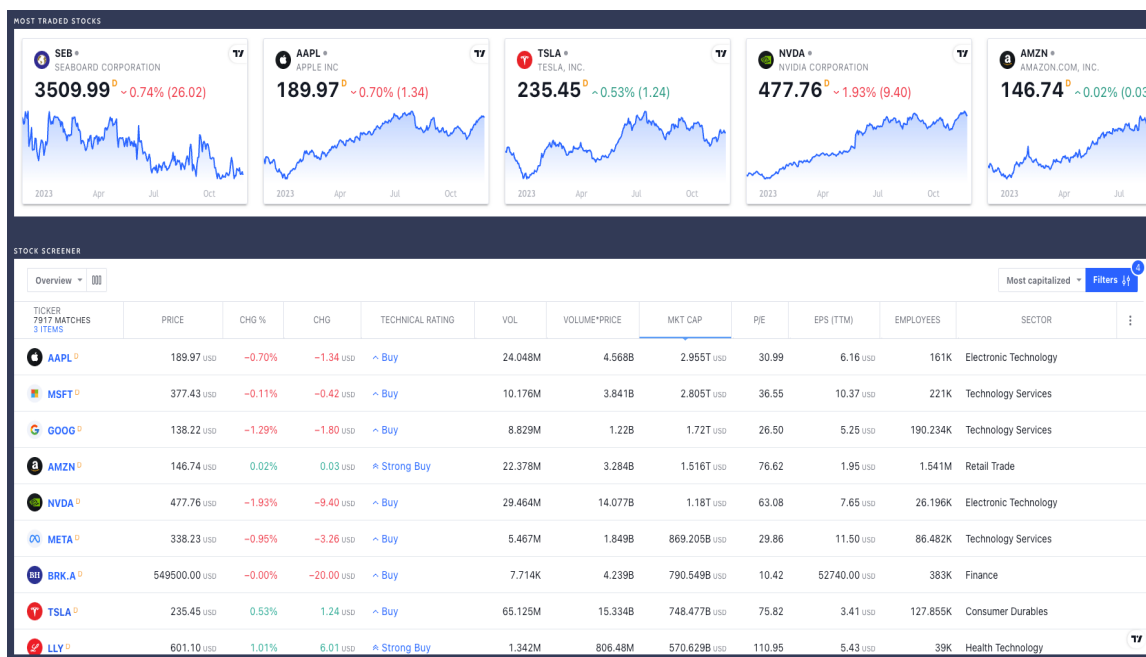


Рис. 1.2.2 Сторінка переліку акцій

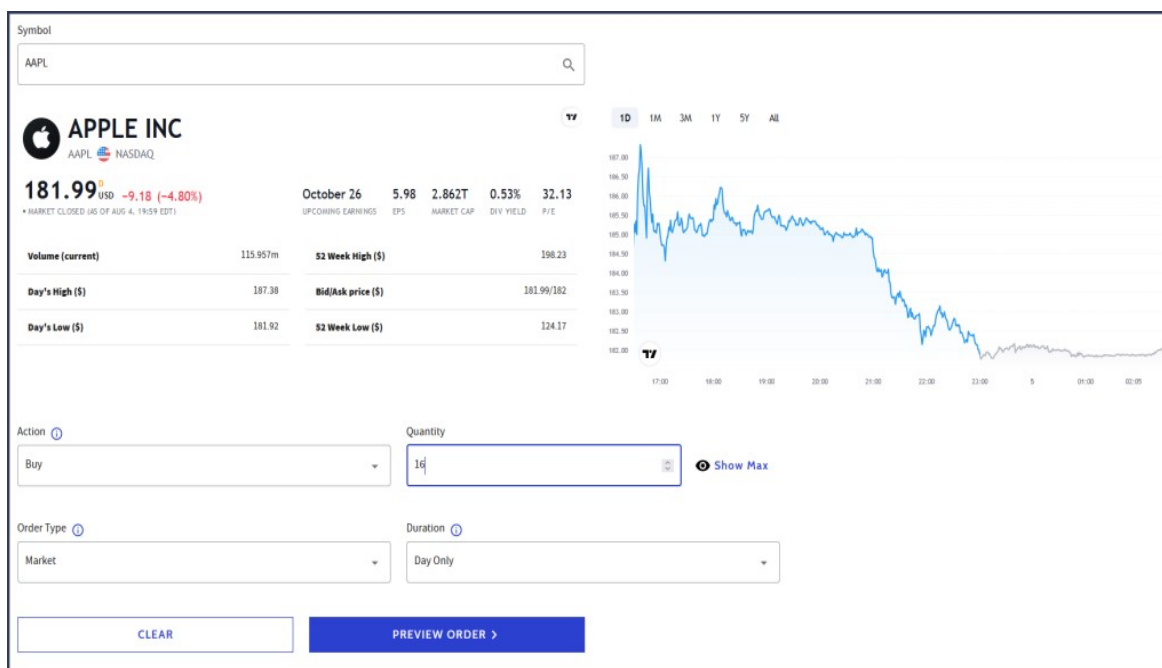


Рис. 1.2.3 Інформаційна панель про акцію

Перейдемо до наступного аналогу «Wall Street Survivor» який є однією із найпопулярніших платформ для віртуальної торгівлі акціями. Вона спрямована на навчання людей основам інвестування та фінансової грамотності за допомогою інтерактивних курсів та віртуальної торгівлі. Цей аналог є дуже подібний до попереднього, але має деякі переваги над попереднім, а саме:

- Інтерактивність – платформа використовує гейміфікацію, що робить процес навчання більш захоплюючим;
- Широкий спектр ресурсів – починаючи від текстових статей і до відеоуроків, «Wall Street Survivor» намагається покрити якнайбільше аспектів інвестування.

Серед недоліків – це рекламні матеріали, які мають комерційний характер та обмежений доступ до інструментів віртуальної торгівлі акціями, але основним недоліком є також відсутність функціоналу аналізу даних та побудови прогнозів.

Отже, порівнюючи із обраною темою, то обидві платформи зосереджені на акційному ринку, тоді як наша тема більше зосереджена на ОВДП.

Також основна ідея нашого симулятора полягає в тому, що окрім взаємодії з облігаціями, він має можливості для проведення аналітики та прогнозування, що в свою чергу відкриває ряд можливостей, як для користувачів так і розробників та фахівців в сфері економіки. Тоді як платформи аналоги не надають таких можливостей. Тут користувачі можуть тільки купувати, продавати акції, спостерігати за власним портфелем.

1.3 Аналіз методів машинного навчання

Машинне навчання (ML) – це метод аналізу даних у якому додатки використовують наявні дані для виявлення закономірностей і прийняття рішень без чітко запрограмованих вказівок. Іншими словами, застосунки навчаються самі, без втручання людини. Будучи надійним методом аналізу даних, ML використовується в багатьох галузях, включно з класифікацією, кластеризацією, передбачувальною аналітикою, асоціативним навчанням, виявленням аномалій, аналізом зображень та обробкою природної мови.[38]

Існує декілька типів машинного навчання, але основні три – це навчання з вчителем, без вчителя та з підкріпленням. Вони визначаються різними критеріями, але насамперед вони відображають різні підходи до передачі та опрацювання інформації у процесі навчання.

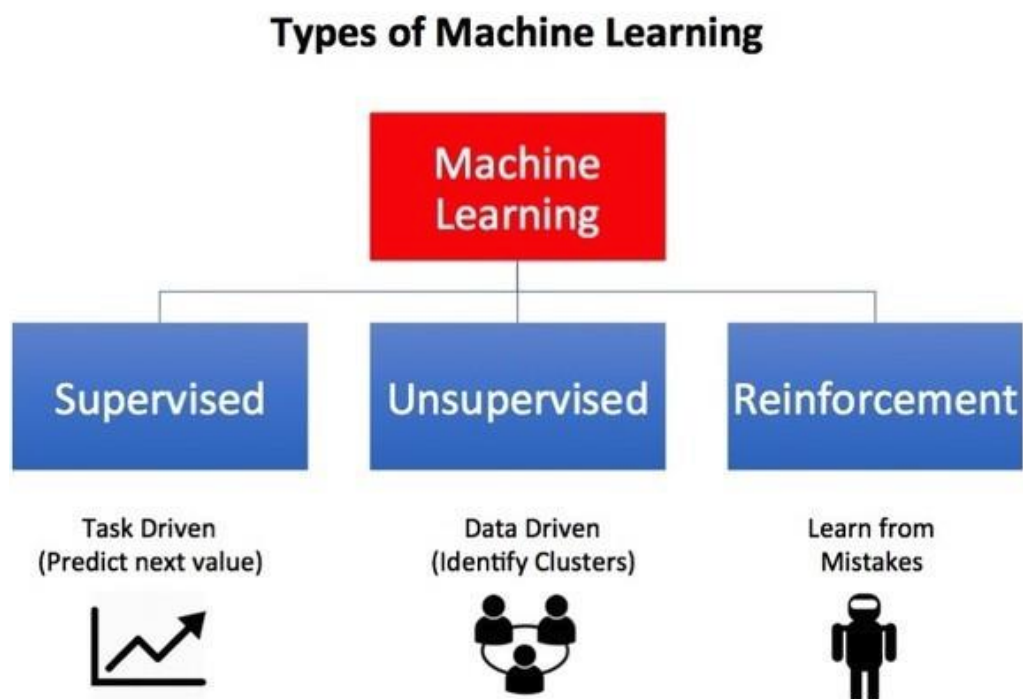


Рис. 1.3.1 Види машинного навчання

Машинне навчання є винятково ефективним у розв'язанні складних проблем та задач, для яких традиційні методи вимагають великих витрат та складного ручного налаштування, або ж у випадках, коли взагалі відсутні готові рішення.

Однією з ключових переваг ML в порівнянні з традиційними алгоритмами є автоматизація процесів рішень. Це дозволяє уникнути витрат на розробку функціоналу, який забезпечуватиме можливість знаходження аномалій, прогнозування майбутніх сценаріїв, адаптації до зміни оточення, а також висвітлення складних проблем у великих обсягах даних і виявлення в цих даних наявних закономірностей, які можуть бути непомітними для людини.[9]

Метод контрольованого навчання в машинному навчанні – це ефективний підхід, що здатний успішно впоратися з простими завданнями та вважається легким для реалізації. Основна концепція полягає в тому, що вхідні дані мають мітки, які вказують на очікуваний вихідний результат алгоритму. В свою чергу – це допомагає системі розпізнавати закономірності в даних та удосконалювати свою здатність до класифікації та прогнозування.

Контрольоване навчання поділяється на дві основні категорії алгоритмів:

- класифікацію;
- регресію.

У класифікації вихідна змінна призначена для конкретної категорії, в той час як для регресії вихідна змінна представляється реальним числовим значенням.

Порівнюючи цей метод, можна прирівняти контрольоване навчання до ситуації, коли людина навчає собаку новим вмінням, використовуючи систему похвали та винагород. Коли власник навчає свого собаку, він показує, як правильно виконати потрібну команду і дає за це нагороду, коли той правильно виконує завдання.

В цьому випадку власник собаки використовує заздалегідь відомий ключ – правильне виконання команди для навчання нових вмінь собаки. Пес вивчає асоціації між певною задачею і отриманням винагороди за це. З часом, за допомогою такого контрольованого навчання, собака розуміє, як виконувати кожен із команд без прямого показу власника, подібно до того, як система навчання може вдосконалювати свої навички на основі заздалегідь відомих відповідей.

Щодо регресійних моделей машинного навчання, вони широко використовуються в прогнозуванні для передбачення тенденцій та результатів. Ці моделі навчаються розуміти залежності між різними факторами та бажаним результатом, що дозволяє їм працювати в різних умовах та з різними видами даних.

Використання регресійних моделей дозволяє прогнозувати результати на основі появи нових даних, а також враховувати коливання ринку та тестування в різних умовах. Здобуті знання можуть бути використані для аналізу та прийняття ефективних рішень в різних галузях.

Класифікація – є одним із розділів контрольованого машинного навчання, що зосереджений на розв'язанні задач в яких існує багато об'єктів

або ситуацій, які поділені на різні класи. Є визначений набір об'єктів з відомою класовою приналежністю, який відомий як тренувальний датасет. Класи інших об'єктів невідомі.[25]

Завдання полягає у створенні алгоритму, який зможе визначити клас будь-якого об'єкта з первинної множини. Класифікувати об'єкт – означає встановити номер або назву для класу до якого він належить. Класифікація об'єкта визначається як номер або назва класу, яку алгоритм присвоює об'єкту при його застосуванні.

Алгоритми бінарної класифікації:

- Метод опорних векторів;
- Наївний баєвський класифікатор;
- К-найближчих сусідів;
- Дерева прийняття рішень;
- Випадковий ліс;
- Логістична регресія.

Навчання без вчителя визначається, як категорія алгоритмів машинного навчання, що проводять пошук шаблонів у наборах даних, які не містять заздалегідь встановлених міток. Відповідно до своєї назви, цей підхід не вимагає контролю з боку людини чи обробки – він працює самостійно.

Алгоритми навчання без вчителя можна застосовувати у наступних сферах:

Кластеризація: процес полягає у виявленні подібних немаркованих значень і їх подальшому групуванні.[10]

Пошук асоціацій: використовується для виявлення взаємозв'язків між різними елементами у наборі даних.

Зменшення розмірності: даний підхід допомагає у випадках, коли набір даних містить надто велику кількість змінних, спрощуючи його до більш зручного розміру, але при цьому зберігаючи важливу інформацію.

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

Наївний баєсів класифікатор (Naive Bayes) – це класифікатор, що є одним із великої сукупності баєсових методів. Алгоритмів, які засновані на принципах максимуму апостеріорної вірогідності. Для об'єкта, що піддають класифікації обраховують так звані функції правдоподібності кожного з класів, по ним обраховують апостеріорні вірогідності цих класів. Після чого об'єкт буде віднесено до того класу, для якого апостеріорна вірогідність максимальна.[30]

Наївний баєсовий алгоритм – це алгоритм класифікації, що побудований на теоремі Баєса з припущеннями незалежності ознак. Якщо казати іншими словами, то даний алгоритм передбачає, що наявність будь-якої іншої ознаки в класі не є пов'язано з наявністю будь-якої іншої ознаки.

Наприклад, фрукт може рахуватись яблуком, якщо він червоний, круглий і його діаметр є близько 8 сантиметрів. Попри те що ці ознаки є залежними один від одного або від інших ознак. В будь-якому випадку вони вносять незалежний внесок у імовірність, що цей фрукт буде яблуком. Із-за таких припущень алгоритм називається «наївним».

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

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

Рис. 1.3.1 Формула розрахунку апостеріальної ймовірності $P(y|x)$

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

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

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

$P(x)$ – predictor prior probability – безумовна (апріорна) ймовірність даного

значення ознаки.

Простіше для розуміння буде переписати теорему з іншими позначеннями:

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{P(D)} \quad (1.2)$$

Якщо зіставити це з типовим завданням машинного навчання, то тут D – це ті дані що ми маємо, а θ – це параметри моделі, які ми хочемо навчити. Кожна з ймовірностей теж має свій сенс.

Отже, $P(\theta|D)$ – це те, що ми будемо шукати, розподіл ймовірностей параметрів моделі після того, як ми взяли до уваги дані. Це називається апостеріорною ймовірністю. Така ймовірність, як правило, просто так не знайти, і тут як раз на поміч приходить нам теорема Баєса.

Значення $P(D|\theta)$ – це правдоподібність ймовірності даних за умови зафіксованих параметрів моделі. Знайти її у більшості випадків легко, адже конструкція моделі зазвичай в тому і полягає, щоб задати функцію правдоподібності. Попередня ймовірність $P(\theta)$ (prior probability) – це математична формалізація нашої інтуїції про предмет, формалізацією того, що ми знали раніше, ще до різного роду експериментів.

Сьогодні баєсів підхід вважається швидше як розгляд ймовірностей з позиції «ступеня довіри». В результаті кожен тестовий приклад x приймає значення із множини категорій V та описується атрибутами $\langle a_1, a_2, \dots, a_n \rangle$. Нам же необхідно знайти найбільш вірогідне значення даного атрибута:

Основні кроки алгоритму:

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

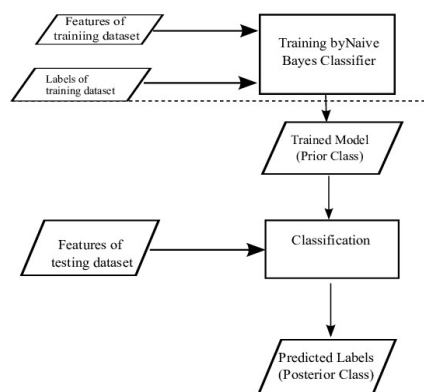


Рис. 1.5.1 Блок-схема навчання та ідентифікації за Баєсовим класифікатором

Наступним є алгоритм метод випадкового лісу (**Random Forest Classifier**), що використовується для машинного навчання та прогнозування на основі наявних даних. Метод випадкового лісу є потужним алгоритмом машинного навчання, що використовує концепцію "випадкових лісів" для класифікації даних. Він належить до категорії ансамблевих методів, що об'єднує декілька дерев рішень для створення більш точної та стабільної моделі класифікації. [13]

Основна ідея алгоритму полягає в тому, що він будує кілька дерев рішень, використовуючи підвибірки навчального набору даних і випадково вибрану підмножину ознак для кожного дерева. Кожне дерево рішень вирішує задачу класифікації незалежно від інших дерев. При класифікації нових об'єктів, алгоритм об'єднує результати всіх дерев, використовуючи голосування або вагове голосування, щоб прийняти остаточне рішення.

Один з ключових аспектів алгоритму випадкових лісів – це випадковість. Алгоритм використовує випадкову вибірку замість повного навчального набору даних для побудови кожного дерева. Це йому дозволяє забезпечити різноманітність дерев, оскільки кожне дерево навчається на іншій підмножині даних. Крім того, для кожного дерева випадково вибираються ознаки, що додають ще більшої варіативності. Цей підхід зменшує ризик перенавчання і допомагає підвищити стійкість моделі до шуму в даних.

Кожне дерево рішень в алгоритмі є розгалуженою структурою «дерева». Воно рекурсивно розгалужується на основі наявних ознак, шукаючи найкращі розділювачі, які найбільше зменшують нев'язку (наприклад, за допомогою

критерію Джині або ентропії). Кожне розгалуження вирішується на основі ваги ознак та порогового значення. У листових вузлах дерева знаходяться класи, до яких належать об'єкти, що потрапляють у цей листок.

Алгоритм демонструє багато переваг. Він ефективно працює з великими наборами даних, включаючи дані з багатьма ознаками і може обробляти пропущені значення. Також, він може оцінювати важливість ознак, що дозволяє отримати уявлення про те, які ознаки найбільше сприяють класифікації. Даний алгоритм відносно стійкий до перенавчання і має високу точність.

Підводячи підсумки, цей алгоритм застосовується в багатьох сферах, таких як обробка природних мов, фінанси та медицина. Його зручно використовувати і в задачах класифікації, коли потрібно отримати надійну та точну модель, здатну працювати з різними типами даних.

Основні кроки алгоритму:

1. Вибір підмножини тренувальних даних та випадкова побудова дерев рішень
2. Тренування дерев рішень: для кожного дерева рішень в ансамблі виконується тренування, де використовуються підмножини тренувальних даних та випадкові підмножини ознак.
3. Прогнозування: класифікація нових вхідних даних шляхом ансамблювання прогнозів окремих дерев рішень. Клас, який набирає найбільшу кількість голосів від окремих дерев, стає прогнозованим класом.

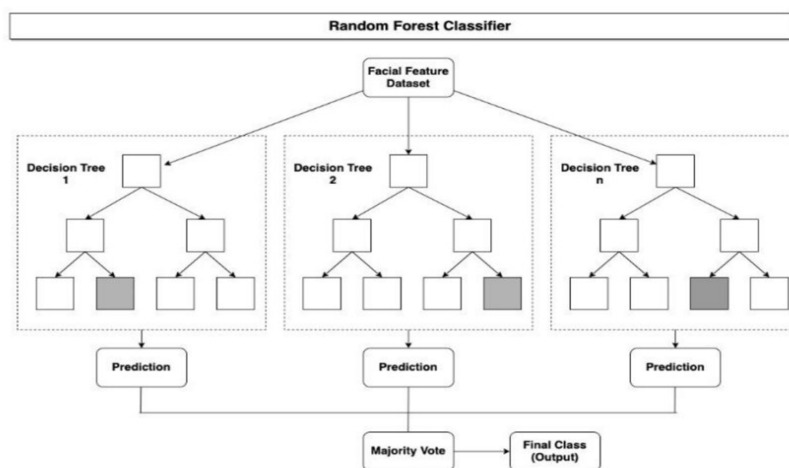


Рис. 1.5.2 Блок-схема роботи алгоритму

Перейдемо до наступного алгоритму нейронної мережі для розв'язування задач машинного навчання – **Sequential**. Це нейронна мережа, яка є однією із найпоширеніших нейронних мереж для розв'язання задач машинного навчання. Вона відноситься до класу штучних нейронних мереж, які намагаються змодельовати роботу людського мозку. Даний алгоритм – це проста лінійна модель в якій нейрони організовані в послідовному порядку шарів.[31]

В Sequential кожен шар мережі має вхідні дані та передає їх до наступного шару для обробки. Кожен шар може мати різну кількість нейронів і функцію активації, яка визначає, як нейрон активується при отриманні вхідних даних. Sequential може містити різні типи шарів, такі як повнозв'язні (dense), звукові (convolutional) або з'єднувальні (recurrent), залежно від вимог завдання.

Один з основних етапів використання цього алгоритму є процес навчання. Зазвичай це включає в себе поділ навчальних даних на тренувальний та набори, визначення функції втрати та оптимізатора, ітерацію через тренувальні дані та оновлення вагових коефіцієнтів мережі для покращення результатів.

Даний алгоритм є потужним і гнучким інструментом для розв'язання різних задач ML, таких як класифікація, регресія або генерація тексту. Він добре знаходить застосування в багатьох сферах, включаючи обробку природньої мови, комп'ютерний зору, фінансів та біоінформатики. Завдячуючи своїй простоті використання та ефективності, Sequential є одним із популярних у виборі для розробників та дослідників у галузі машинного навчання.

Кроки створення нейронної мережі:

1. Ініціалізація моделі;
2. Додавання шарів: додавання складових шарів з вказаною кількістю нейронів та активаційною функцією;
3. Компіляція моделі: визначення функції втрат, оптимізатора та метрик для оцінки моделі;
4. Тренування моделі: тренування моделі з використанням тренувальних даних і вказаною кількістю епох та розміром пакету;
5. Прогнозування результатів для нових вхідних даних.

Отже, було проведено опис роботи основних алгоритмів машинного навчання, що буде використано в кваліфікаційній роботі для аналізу та прогнозування на основі наявних наборів даних із симулятора.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для нашого програмного комплексу. ми з моїм колегою вибрали основної мовою програмування – JavaScript та мікросервісну архітектуру.

Мікросервісна архітектура чудово підходить для нашого симулятора, адже ми маємо можливість для створення нових модулів, які є незалежними і майже не впливають на роботу один-одного, але мають високий потенціал для масштабування.[6]

JavaScript – це мова програмування, яка є універсальною, адже вона може бути використана як для розробки клієнтського додатку так і для серверної частини симулятора. Для побудови клієнтської частини колега використовував JavaScript із використанням бібліотеки React.js, а я буду використовувати для розробки серверної частини Node.js. Для проведення аналізу даних та прогнозування буде використано мову програмування Python та наступні модулі: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Statsmodels, TensorFlow, Keras.

React.js – це одна з найпопулярніших JavaScript бібліотек, яка використовується для побудови інтерфейсів користувача, зокрема для SPA застосунків. Ця бібліотека була розроблена команда Facebook. Вона швидко здобула популярність серед розробників завдяки своїй гнучкості, ефективності та простоті використання, адже однією з ключових особливостей React.js є використання компонентів. Компоненти дозволяють розробникам розділяти інтерфейс на незалежні частини, які можуть повторно використовуватись без дублювання коду. В результаті чого це спрощує управління складними інтерфейсами, оскільки зміни в одному компоненті не впливають на інші..

Node.js – це потужна платформа, яка змінила спосіб, яким розробники

дивляться на JavaScript. Раніше JavaScript використовувався виключно для клієнтської частини веб-додатків, працюючи в браузері для додання інтерактивності на веб-сторінки. Але з появою Node.js, JavaScript почав використовуватись на серверній стороні. В основі Node.js лежить використання подієвої моделі та неблокуючих операцій введення та виведення, що робить його особливим для розробки масштабованих мережевих додатків. Він ідеально підходить для обробки великої кількості одночасних з'єднань з мінімальними затратами ресурсів, що є великою перевагою у сучасній веб-розробці.

Express.js – це один із найпопулярніших фреймворків для веб-розробки на стороні сервера, створений для Node.js. Його головною метою є спрощення процесу створення веб-додатків та API. Він забезпечує мінімалістичний підхід, але в той же час є досить потужним, щоб розробляти складні веб-додатки. Основною перевагою Express є його гнучкість та простота використання. Ще має простий, але потужний механізм маршрутизації, який дозволяє легко керувати запитами від клієнтів.[4]

MongoDB – це документо-орієнтована система керування базами даних. В якій замість використання таблиць та рядків, використовуються колекції та документи. Документи – це пара ключ-значення, що є основною одиницею даних в цій СУБД.

Основні переваги MongoDB – це висока швидкість роботи, гнучкість та можливість швидкого масштабування, кросплатформеність, реплікація – можливість працювати на декількох серверах надаючи ще більшої надійності, дані зберігаються у вигляді JSON документів.

Недоліком MongoDB – є труднощі, що виникають під час роботи із жорстко пов'язаними даними. Хоч і наявний інструмент агрегація, але він завжди є простим, а особливо для новачків.

Mongoose – це бібліотека ODM, яка використовується, щоб створити зв'язок між MongoDB та Node.js. Надає великий набір функцій для створення та

роботи зі сховищем Mongo.[5]

Python – це одна із найпопулярніших мов програмування для роботи з машинним навчанням та аналізом даних. Це пов'язано із великою кількістю доступних бібліотек та різних навчальних матеріалів. Ця мова є високорівневою, інтерпретованою з динамічною типізацією та великим набором стандартних бібліотек. Що в результаті робить її чудовим вибором для програмістів початківців, а також для розвинутих розробників. []

Розвинута підтримка бібліотек таких як NumPy – для наукових обчислень, Pandas – для обробки даних, Matplotlib – для візуалізації, Scikit-learn – для машинного навчання та TensorFlow – для глибокого навчання перетворює Python в потужний інструмент для аналізу даних та розробки складних алгоритмічних моделей.

TensorFlow та Keras – відомі бібліотеки для глибокого навчання, які підтримують широкий спектр алгоритмів і моделей.

Scikit-learn – це бібліотека, яка має широкий набір інструментів для аналізу даних та моделювання. Вона підтримує багато популярних алгоритмів машинного навчання.

Pandas – бібліотека для обробки та аналізу даних, яка дозволяє легко працювати з табличними даними в Python.

Висновки до першого розділу

В процесі написання першого розділу було виконано постановку задачі вибору та проведення дослідження моделей машинного навчання, що будуть використані під час розробки програмного забезпечення. Це надасть можливості проведення аналізу та прогнозування поведінки користувачів під час взаємодії із системою симулятора. Адже в результаті розроблений модуль може стати інструментом, що буде використовуватись у підвищенні фінансової грамотності та підготовці користувачів до викликів у світі фінансів та інвестицій.

Проведення аналізу існуючих аналогів дозволило нам визначити їхні сильні та слабкі сторони, сформулювати основні вимоги до розроблюваного

модуля, а також виділити функціонал для вдосконалення та розширення, який є фундаментальним у кваліфікаційній роботі.

В розділі було наведено теоретичну основу та опис алгоритмів, що будуть використані для подальшого проектування і спрямовані на прогнозування та аналіз даних.

Також було проведено вибір інструментальних засобів та вимог до апаратного забезпечення, що дає нам не тільки технічну можливість розширення функціоналу симулятора, але й оптимальне використання ресурсів з максимальною ефективністю роботи симулятора.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ КОРИСТУВАЧІВ

2.1 Варіанти використання системи

Новий модуль розробляється з метою розширення наявного функціоналу симулятора та створенням підґрунтям для ще більших майбутніх вдосконалення додатка на основі розробленого модуля.

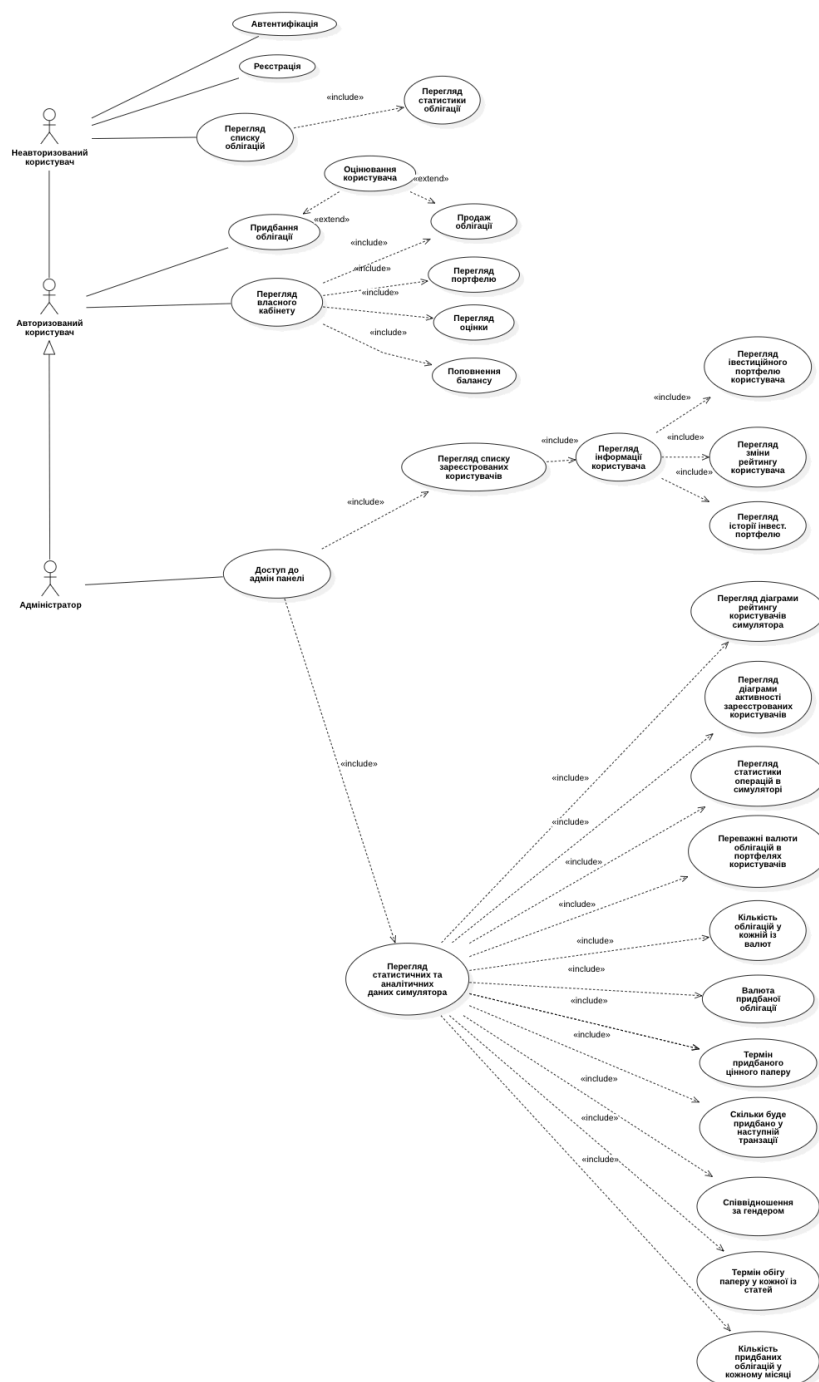


Рис. 2.1.1 Діаграма використання симулятора

Вимоги не автентифікованих користувачів:

1. Перегляд списку облігацій;
2. Перегляд інформації про облігацію.

Автентифікований користувач розширюватиме можливості не автентифікованого користувача.

Вимоги до автентифікованих користувачів:

1. Придбання облігацій;
2. Перегляд кабінету користувача;
3. Перегляд власного інвестиційного портфеля;
4. Поповнення балансу;
5. Перегляд власного рейтингу.
6. Продаж облігацій;

Адміністратор буде мати всі ті ж можливості, що й автентифікований користувач, але ще для нього буде доступний функціонал, який є обмежений для простого користувача.

Вимоги до адміністратора:

1. Перегляд діаграми кількості користувачів із відповідним рейтингом;
2. Перегляд активних та неактивних зареєстрованих користувачів;
3. Діаграма кількості облігацій у різних валютах;
4. Діаграма валют, які переважають в портфелі користувачів;
5. Перегляд інформації про співвідношення кількості чоловіків до жінок в симуляторів;
6. Розподіл за терміном обігу цінного паперу для кожного полу;
7. Графік щільності відношення вартості до терміну обігу;
8. Діаграма сумарної кількості придбаних ОВДП за кожен місяць;
9. Прогноз валюти наступної облігації, з можливістю розподілу за статтю, віком, освітою;
10. Прогноз кількості облігацій за наступну транзакцію, як для одного користувача так і загалом;
11. Прогноз терміну цінного паперу, як для одного користувача так і

загально;

12. Прогноз скільки користувач буде тримати цінний папір у власності.

2.2 Обробка даних перед проведенням аналізу

В цьому розділі ми зосередимося на процесі збору, зберігання та обробки даних про користувачів, які отримуємо з симулятора і будемо використовувати в новому розробленому сервісі. Вся ця інформація зберігається в базі даних MongoDB, як було зазначено раніше відомій своєю гнучкістю та масштабованістю, що робить її ідеальною для роботи з великими обсягами даних та їх структуруванням.

Для ефективної обробки та взаємодії, дані будуть перетворюватись у формат CSV. В свою чергу, дані в такому форматі будуть передаватись у Python для подальшої роботи з ними.

Для зберігання даних було використано 3 основних колекції: bonds, userbonds, userinformations.

Колекція «bonds» відповідає за збереження актуальних облігацій України з якими можна проводити операції у додатку.

Таблиця №2.2.1 Опис документів колекції «bonds»

Назва	Тип	Опис
_id	ObjectID	Ідентифікатор облігації
buyPrice	Number	Вартість для придбання
code	String	ISIN код облігації
bondType	Number	Тип облігації
sellPrice	Number	Вартість продажу
termType	String	Термін обігу
currency	String	Валюта
emitDate	Date	Дата випуску
repayDate	Date	Дата погашення
paymentSchedule	Array	Графік виплати

Продовження таблиці №2.2.1

nearestPayDate	Date	Найближча дата виплати
nominal	Number	Номінал облігації
YTM	Number	Прибутковість до продажу
yearCoupon	Number	Номінальний рівень дохідності (%)

Колекція «userbonds» зберігає інформацію про придбані облігації користувачами симулятора, тобто їхній інвестиційний портфель.

Таблиця №2.2.2 Опис документів колекції «userbonds»

Назва	Тип	Посилання на іншу колекцію	Опис
_id	ObjectID	—	Ідентифікатор
buyDate	Date	—	Дата придбання
userID	ObjectID	user	Ідентифікатор користувача
bondID	ObjectID	bonds	Ідентифікатор облігації
qty	Number	—	Придбана кількість
buyPrice	Number	—	Вартість купівлі

Колекція «userinformations» є однією із найважливіших, адже на основі цієї інформації ми можемо багато зрозуміти про користувача, що дасть нам змогу зробити більш точну аналітику та прогнозування його поведінки у майбутній взаємодії з симулятором та ОВДП.

Таблиця №2.2.3 Опис документів колекції «userinformations»

Назва	Тип	Посилання на іншу колекцію	Опис
_id	ObjectID	—	Ідентифікатор
dateOfBirth	ObjectID	user	Ідентифікатор користувача
gender	Enum	—	Стать
educationLevel	String	—	Освіта
profession	String	—	Професія

Продовження таблиці №2.2.3

location	Object	–	Місце проживання
maritalStatus	String	–	Сімейний стан
investmentExperience	Number	–	Кількість років досвіду в інвестиціях
riskTolerance	String	–	Відношення до ризику
invesmentKnowledge	String	–	Знання у інвестиціях
tradingFrequency	String	–	Періодичність торгівлі

Для збору даних із бази даних було розроблений необхідний функціонал, адже інформація зберігається в різних колекціях. В результаті для цього було використано інструмент – агрегації.

Агрегація – це механізм обробки даних в MongoDB, що дозволяє виконувати складні операції на даних, такі як фільтрація, групування, сортування та виконання інших трансформацій з даними.

Лістинг функції збору даних про цінні папери в портфелях користувача(ів) та інформації про цінні папери

```

async getUsersBondsAndBondsInfo(userID) {
  const data = await userBondsModel.aggregate([
    {
      $match: userID ? {userID: ObjectId(userID)} : {},
    }, {
      $lookup: {
        from: "bonds", localField: "bondID", foreignField:
        "_id", as: "bond",
      },
    }, {
      $unwind: "$bond", {
        $project: {
          _id: true,
          userID: true,
          qty: true,
          buyPrice: true,
          buyDate: true,
          monthName: true,
          monthNumber: true,
          code: "$bond.code",
          currency: "$bond.currency",
          termType: "$bond.termType",
          yearCoupon: "$bond.yearCoupon",
          Y: "$bond.Y",
          YTM: "$bond.YTM",
        }
      }
    }
  ])
}

```

```

    },
  ]]);

  if (!!data.length) {
    await this.writeInCSV(data, "userBonds_bondsInfo")
    return data
  }
  return [];
}

```

Лістинг функції збору даних про цінні папери в портфелях користувача(ів) з інформацією про цінні папери та інформацією про користувача

```

async getUsersAndBonds(userID) {
  const data = await userBondsModel.aggregate([
    {
      $match: userID ? {userID: ObjectId(userID)} : {},
    }, {
      $lookup: {
        from: "bonds", localField: "bondID", foreignField:
"_id", as: "bond",
      },
    }, {
      $lookup: {
        from: "userinformations", localField: "userID",
foreignField: "userID", as: "userInformation",
      },
    },
    {
      $unwind: "$bond", { $unwind: "$userInformation", {
        $project: {
          _id: true,
          userID: true,
          qty: true,
          buyPrice: true,
          buyDate: true,
          monthName: true,
          monthNumber: true,
          code: "$bond.code",
          currency: "$bond.currency",
          termType: "$bond.termType",
          yearCoupon: "$bond.yearCoupon",
          Y: "$bond.Y",
          YTM: "$bond.YTM",
          gender: "$userInformation.gender",
          educationLevel: "$userInformation.educationLevel",
          investmentExperience:
"$userInformation.investmentExperience",
          profession: "$userInformation.profession",
          riskTolerance: "$userInformation.riskTolerance",
          tradingFrequency:
"$userInformation.tradingFrequency",
        },
      },
    },
  ]),
}

```

```

    }]);

    if (!!data.length) {
        await this.writeInCSV(data,
            "userBonds_bondsInfo_userInfo")
        return data
    }
    return [];
}

```

Отже, ми зібрали необхідні для нас колекції та розробили функціонал для збору потрібних нам наборів даних, які будуть оброблятися та записуватись в CSV файли. Кожного разу при викликові цих функцій дані завжди будуть актуальними.

2.3 Аналіз вхідних даних

В цьому розділі ми зосередимося на процесі аналізу вхідних даних про користувачів, які отримуємо з симулятора для роботи нашого модуля.

Опис полів, які наявні в наборах даних:

Поле «qty»: кількість придбаних цінних паперів.

Поле «buyPrice»: вартість купівлі цінного паперу

Поле «monthNumber»: місяць у якому було здійснено транзакцію

Поле «code»: номер цінного паперу.

Поле «currency»: валюта облігації.

Поле «termType»: термін обігу (короткострокові, середньострокові, довгострокові).

Поле «yearCoupon»: номінальний рівень прибутковості (%)

Поле «YTM»: Прибутковість до продажу

Поле «dateOfBirth»: Дата народження.

Поле «gender»: Стать користувача

Поле «educationLevel»: Рівень освіти (Початкова, Основна середня, Старша середня, Професійно-технічна, Неповна вища, Бакалавр, Магістр, Спеціаліст, Аспірант, Докторант, Доктор філософії, Доктор наук, Доцент, Професор).

Поле «maritalStatus»: Сімейний стан (неодружений, одружений, розведений).

Поле «location»: Локація, що включає країну та місто.

Поле «profession»: Професія.

Поле «investmentExperience»: Досвід в інвестиціях.

Поле «investmentKnowledge»: Знання в інвестуванні(початковий, середній, високий).

Поле «riskTolerance»: Відношення до ризику (низький, середній, високий).

Поле «tradingFrequency»: Частота торгівлі (рідко, щомісяця, щотижня, щодня)

```
dataset = read_csv('userBondsAndBondsInfo.csv')
print(dataset.head(10))
```

	_ID	USERID	QTY	BUYPRICE	\
0	655a2093243cfd0fa01da8c6	655a2092243cfd0fa01da8b2	24	1054	
1	655a2093243cfd0fa01da8d9	655a2092243cfd0fa01da8b2	19	1283	
2	655a2094243cfd0fa01da8ee	655a2092243cfd0fa01da8b2	37	1253	
3	655a2095243cfd0fa01da903	655a2092243cfd0fa01da8b2	9	932	
4	655a2096243cfd0fa01da919	655a2092243cfd0fa01da8b2	9	1150	
5	655a2097243cfd0fa01da930	655a2092243cfd0fa01da8b2	37	791	
6	655a2099243cfd0fa01da963	655a2098243cfd0fa01da94f	25	767	
7	655a2099243cfd0fa01da976	655a2098243cfd0fa01da94f	24	778	
8	655a209a243cfd0fa01da98a	655a2098243cfd0fa01da94f	35	1069	
9	655a209b243cfd0fa01da99f	655a2098243cfd0fa01da94f	26	923	

	MONTHNAME	MONTHNUMBER	BUYDATE	\
0	October	10	Thu Oct 13 2022 05:33:33 GMT+0300	(Eastern Eur...
1	March	3	Fri Mar 11 2022 16:05:24 GMT+0200	(Eastern Eur...
2	February	2	Sat Feb 19 2022 15:36:50 GMT+0200	(Eastern Eur...
3	March	3	Thu Mar 02 2023 00:46:20 GMT+0200	(Eastern Eur...
4	January	1	Fri Jan 21 2022 13:10:48 GMT+0200	(Eastern Eur...
5	October	10	Mon Oct 02 2023 06:38:48 GMT+0300	(Eastern Eur...
6	February	2	Thu Feb 17 2022 17:46:34 GMT+0200	(Eastern Eur...
7	January	1	Tue Jan 17 2023 11:32:34 GMT+0200	(Eastern Eur...
8	November	11	Fri Nov 18 2022 16:55:36 GMT+0200	(Eastern Eur...
9	January	1	Mon Jan 31 2022 00:57:30 GMT+0200	(Eastern Eur...

	CODE	CURRENCY	TERMTYPE	YEARCOUPON	Y	YTM
0	UA4000173371	UAH	Medium	12.09	11.995041	-302.307508
1	UA4000204150	UAH	Medium	15.85	14.861455	-39.622340
2	UA4000218333	USD	Medium	3.50	3.341363	624.320125
3	UA4000218747	EUR	Medium	2.50	0.838812	-0.068788
4	UA4000216774	UAH	Medium	11.60	11.149976	-699.145691
5	UA4000206460	UAH	Medium	11.35	12.001582	-54.614976
6	UA4000218531	UAH	Medium	12.59	13.093709	-591.722809
7	UA4000223861	UAH	Medium	10.80	7.789447	4917.708781
8	UA4000216923	USD	Medium	3.90	4.653535	-180.345331
9	UA4000207880	UAH	Medium	12.85	12.586317	1045.794115

Рис. 2.3.1 Перших 10 записів набору даних

```
dataset = dataset.drop(['_ID', 'USERID', 'BUYDATE'], axis=1)
print(dataset.head(10))
```

	QTY	BUYPRICE	MONTHNAME	MONTHNUMBER	CODE	CURRENCY	TERMTYPE	\
0	24	1054	October	10	UA4000173371	UAH	Medium	
1	19	1283	March	3	UA4000204150	UAH	Long	
2	37	1253	February	2	UA4000218333	USD	Long	
3	9	932	March	3	UA4000218747	EUR	Medium	
4	9	1150	January	1	UA4000216774	UAH	Short	
5	37	791	October	10	UA4000206460	UAH	Long	
6	25	767	February	2	UA4000218531	UAH	Short	
7	24	778	January	1	UA4000223861	UAH	Medium	
8	35	1069	November	11	UA4000216923	USD	Short	
9	26	923	January	1	UA4000207880	UAH	Medium	

	YEARCOUPON	Y	YTM
0	12.09	11.995041	-302.307508
1	15.85	14.861455	-39.622340
2	3.50	3.341363	624.320125
3	2.50	0.838812	-0.068788
4	11.60	11.149976	-699.145691
5	11.35	12.001582	-54.614976
6	12.59	13.093709	-591.722809
7	10.80	7.789447	4917.708781
8	3.90	4.653535	-180.345331
9	12.85	12.586317	1045.794115

Рис. 2.3.2 Набір даних після очистки не потрібних стовпців

Наш набір містить стовпці, які нам є не потрібними, а саме `buydate`, адже тут зберігається точна дата купівлі цінного паперу в симуляторі. Також `_id` – це ідентифікатор транзакції купівлі в симуляторів та поле `userid`, але для проведення подальшого аналізу ці поля нам не потрібні і ми виконаємо очистку цих даних.

	QTY	BUYPRICE	MONTHNUMBER	CODE	CURRENCY	TERMTYPE	YEARCOUNPON	\
0	24	1054	10	UA4000173371	UAH	Medium	12.09	
1	19	1283	3	UA4000204150	UAH	Long	15.85	
2	37	1253	2	UA4000218333	USD	Long	3.50	
3	9	932	3	UA4000218747	EUR	Medium	2.50	
4	9	1150	1	UA4000216774	UAH	Short	11.60	
5	37	791	10	UA4000206460	UAH	Long	11.35	
6	25	767	2	UA4000218531	UAH	Short	12.59	
7	24	778	1	UA4000223861	UAH	Medium	10.80	
8	35	1069	11	UA4000216923	USD	Short	3.90	
9	26	923	1	UA4000207880	UAH	Medium	12.85	

	Y	YTM	GENDER	EDUCATIONLEVEL	INVESTMENTEXPERIENCE	\
0	11.995041	-302.307508	female	Аспірант	26	
1	14.861455	-39.622340	female	Аспірант	26	
2	3.341363	624.320125	female	Аспірант	26	
3	0.838812	-0.068788	female	Аспірант	26	
4	11.149976	-699.145691	female	Аспірант	26	
5	12.001582	-54.614976	female	Аспірант	26	
6	13.093709	-591.722809	female	Докторант	9	
7	7.789447	4917.708781	female	Докторант	9	
8	4.653535	-180.345331	female	Докторант	9	
9	12.586317	1045.794115	female	Докторант	9	

	PROFESSION	RISKTOLERANCE	TRADINGFREQUENCY
0	Ерготерапевт	high	monthly
1	Ерготерапевт	high	monthly
2	Ерготерапевт	high	monthly
3	Ерготерапевт	high	monthly
4	Ерготерапевт	high	monthly
5	Ерготерапевт	high	monthly
6	Фахівець з публічних відносин	low	monthly
7	Фахівець з публічних відносин	low	monthly
8	Фахівець з публічних відносин	low	monthly
9	Фахівець з публічних відносин	low	monthly

Рис. 2.3.3 Набір даних з інформацією про користувача після очищення

На даному етапі ми не будемо видаляти стовпець `code`, адже, ми можемо робити аналітику по цьому полю і розуміти, яким облігаціям за полем `code` користувачі надають перевагу.

Було зроблено очистку набору, далі зробимо аналіз загальних характеристик в наборі даних.

Використовуючи функцію `describe` з модуля `pandas` виведемо опис кожного стовпця.[19] Цей метод дасть нам змогу отримати описову статистику про наш набір даних, а саме:

- Кількість: Вказує кількість ненульових спостережень у кожному стовпці. Це може бути корисним для визначення чи є в даних пропущені значення.
- Середнє (mean): Середнє значення для кожного числового стовпця.

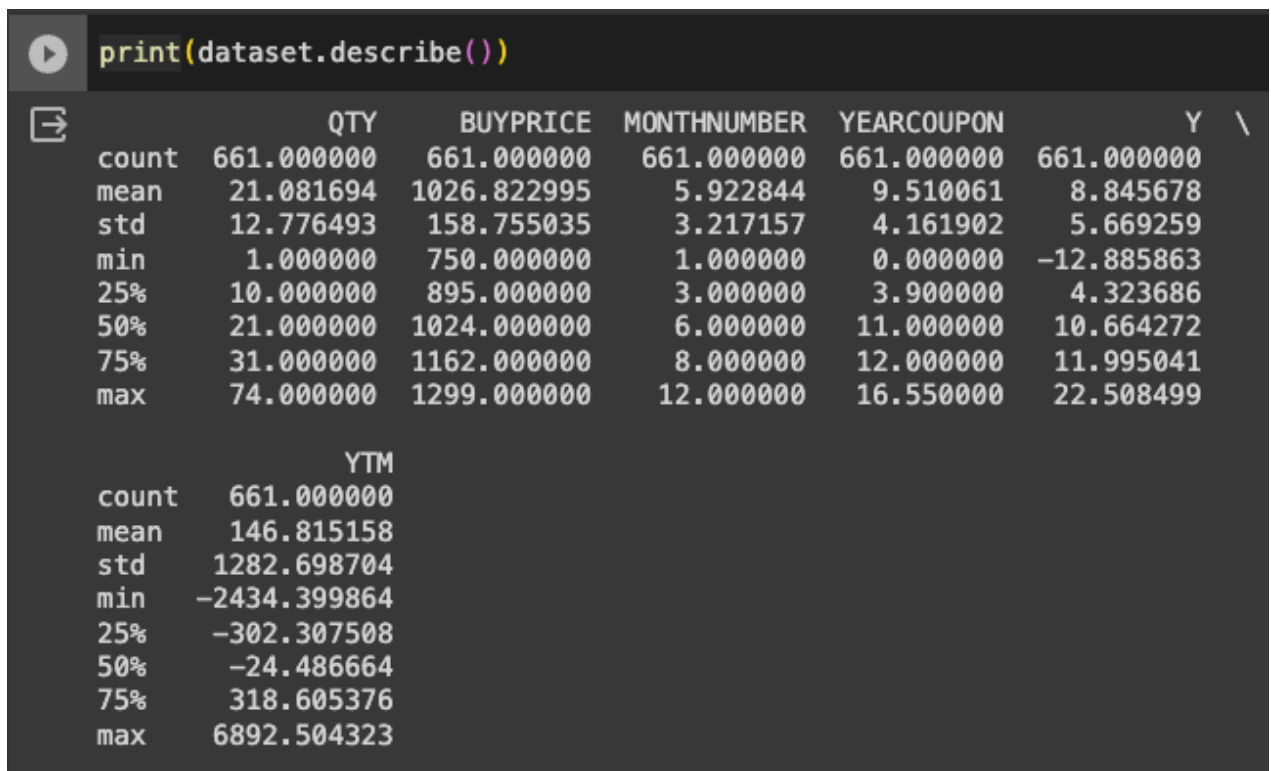
Це дасть нам уявлення про центральну тенденцію даних у кожному стовпці.

- Стандартне відхилення (std): Показує розкид даних навколо середнього значення. Велике стандартне відхилення вказує на більший розкид даних.[22]

- Мінімальне (min) та максимальне (max) значення: Вказує найнижче та найвище значення у кожному стовпці.

- Квартилі: 25%, 50% – медіана та 75% квартилі. Квартилі дають нам уявлення про розподіл наявних даних. 25% кварталі вказує на значення, які знаходяться нижче 25% даних. Медіана показує середню точку даних, а 75% кварталі показує ту точку, нижче якої знаходиться 75% даних.[19]

Отже, ця інформація буде нам корисною для попереднього аналізу набору даних. Це дасть нам змогу швидко оцінити ключові статистичні характеристики кожного із стовпців. В свою чергу це також допоможе нам побачити аномалії, якщо вони у нас є. Такі як дуже великі або навпаки малі значення, що можуть вказувати на помилки або якісь важливі особливості в даних.



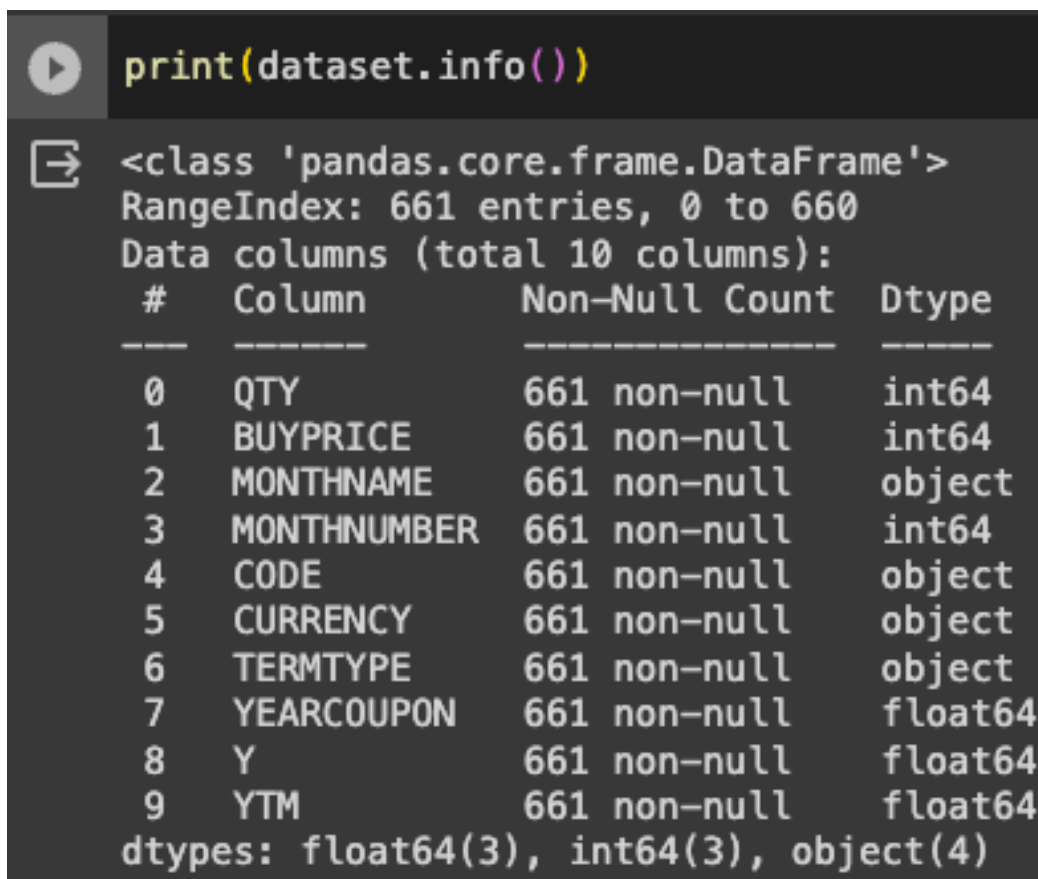
```
print(dataset.describe())
```

	QTY	BUYPRICE	MONTHNUMBER	YEARCoupon	Y \
count	661.000000	661.000000	661.000000	661.000000	661.000000
mean	21.081694	1026.822995	5.922844	9.510061	8.845678
std	12.776493	158.755035	3.217157	4.161902	5.669259
min	1.000000	750.000000	1.000000	0.000000	-12.885863
25%	10.000000	895.000000	3.000000	3.900000	4.323686
50%	21.000000	1024.000000	6.000000	11.000000	10.664272
75%	31.000000	1162.000000	8.000000	12.000000	11.995041
max	74.000000	1299.000000	12.000000	16.550000	22.508499

	YTM
count	661.000000
mean	146.815158
std	1282.698704
min	-2434.399864
25%	-302.307508
50%	-24.486664
75%	318.605376
max	6892.504323

Рис. 2.3.4 Результат описової статистики даних

Виконавши аналіз загальних характеристик в наборі даних. Ми можемо побачити тип даних для кожного стовпця, скільки нульових значень є в стовпці. Це дасть нам змогу зрозуміти чи є пропущені значення в якомусь із стовпців та кількість записів. Всі ці характеристики для стовпців, окрім типу даних, будуть кожного разу відрізнятись, адже симулятор збирає дані на постійній основі, записує їх до бази даних. В результаті чого потім ми їх використовуємо для потрібних нам цілей.



```
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 661 entries, 0 to 660
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  -
0   QTY                  661 non-null   int64
1   BUYPRICE             661 non-null   int64
2   MONTHNAME            661 non-null   object
3   MONTHNUMBER          661 non-null   int64
4   CODE                 661 non-null   object
5   CURRENCY              661 non-null   object
6   TERMTYPE              661 non-null   object
7   YEARCoupon           661 non-null   float64
8   Y                    661 non-null   float64
9   YTM                  661 non-null   float64
dtypes: float64(3), int64(3), object(4)
```

Рис. 2.3.5 Результат типу даних кожного значення в наборі даних

В нашому наборі даних немає нулевих значень, що є дуже хорошим показником для проведення аналітики та прогнозування. Також ми бачимо, що наш набір містить поля, які визначено як об'єкт, але це є специфічність мови програмування Python. Рядки мають властивості та методи, які є характерними для об'єктів. Рядки мають методи для зміни регістру, розділення, інверсії, тощо. Це робить їх гнучкими та потужними для обробки тексту.

Проведемо аналіз, який відсоток цінних паперів купували за терміном обігу. Даний показник може відображати рівень довіри або недовіри у інвесторів,

якщо переважна більшість купує короткострокові цінні папери, то це є індикатором про недовіру до економіки країни та небажання втратити власні інвестиції.

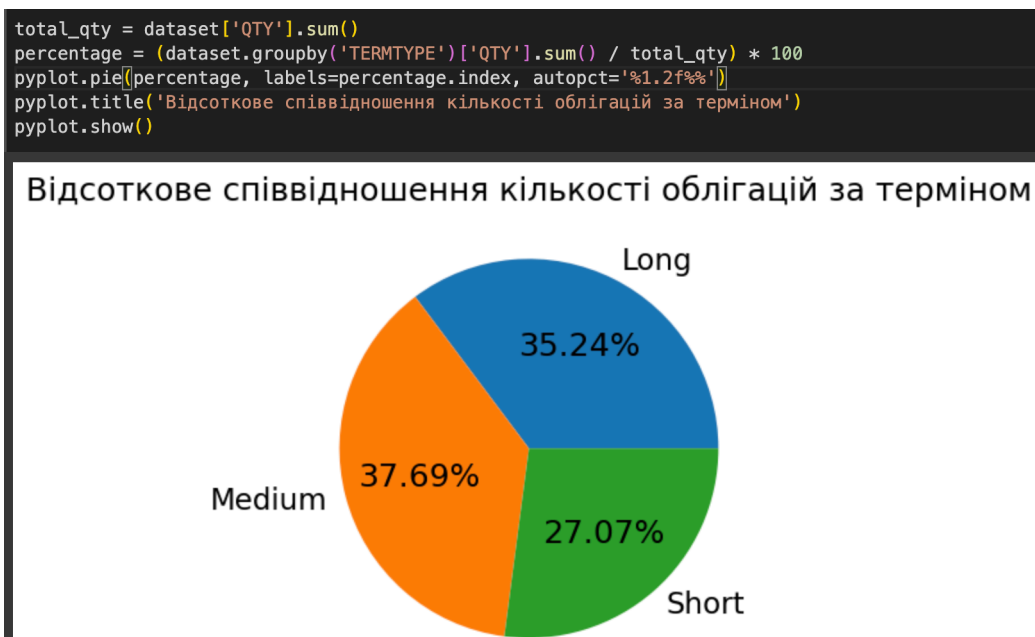


Рис. 2.3.6 Кругова діаграма поділу за терміном обігу цінних паперів у портфелях користувачів

Перейдемо далі і дізнаємось, яка кількість цінних паперів сумарно придбали чоловіки та жінки. Проводячи такий аналіз поведінки користувачів ми можемо адаптувати власні маркетингові стратегії, щоб залучити потрібну демографічну групу. В свою чергу наявна інформація може вказувати на зміни в соціальних та економічних трендах, наприклад, якщо більша кількість жінок, то на зростаючу фінансову незалежність та участь жінок у ринку інвестицій.

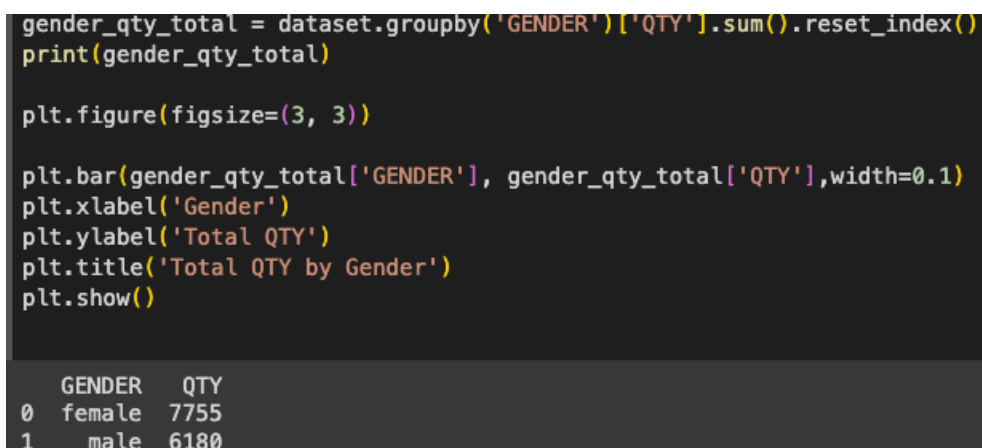


Рис. 2.3.7 Результат відношення кількості жінок та чоловіків у нашому наборі даних

В результаті такого аналізу, ми бачимо, що більшу кількість цінних паперів купили жінки.

Маючи поле «gender» дізнаємось, яким цінним паперам за терміном обігу надає кожна стать. Ця інформація дасть можливість зрозуміти відношення до ризиковості в інвестиціях, перевагу у типах цінних паперів або загальну відмінність у стратегіях інвестування між жінками та чоловіками. Якщо будуть виникати аномалії, то така інформація може спонукати до запровадження подальшого аналізу щодо факторів, які впливають на інвестиційні рішення різних статей та пошуку впливу за для підвищення цього показника. Наприклад, вдосконаленням програми навчання.



Рис. 2.3.8 Результат за терміном обігу для кожної із статей

В нашому результаті ніяких великих аномалій не виявлено. Чоловіки купили менше облігацій різних термів, адже з попереднього показника ми знаємо, що кількість транзакцій з придбання ОВДП у жінок переважають.

Далі перейдемо до побудови діаграми щільності. Діаграма щільності є одним із ключових інструментів в аналізі даних, що дозволяє чіткіше побачити та зрозуміти характеристики розподілу в наявних даних. Даний тип діаграми на відміну від гістограм, забезпечує гладке відображення, що допомагає візуально оцінити форму розподілу даних.[15] Ми можемо зрозуміти чи є вони нормально розподілені або скошені. Це в свою чергу дозволяє легко виявити відмінності в розподілах, а також допомагає у виявленні викидів або аномалій у даних.

Ця діаграма дасть нам можливість побачити залежність вартості цінного паперу від терміну обігу. Наприклад, цей графік може бути використаний для ефективного планування боргової стратегії країни, включаючи визначення

оптимального терміну випуску нових облігацій та управління держборгом.

Лістинг побудови діаграми:

```
sns.kdeplot(data=dataset, x="BUYPRICE", hue="TERMTYPE", fill=True)
pyplot.xlabel('Варість')
pyplot.ylabel('щільність')
pyplot.title('Графік щільності вартості до типу погашення')

pyplot.show()
```

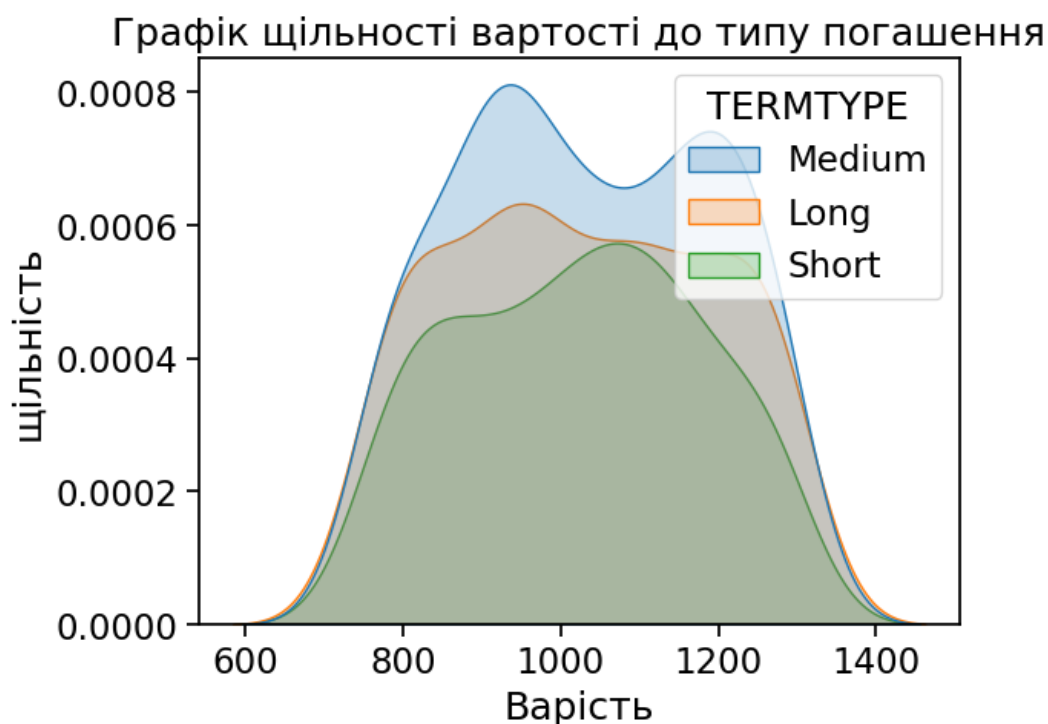


Рис. 2.3.9 Графік щільності відношення вартості до терміну обігу

Стовпчаста діаграма[15] – скільки було придбано цінних паперів у кожному місяці. Ця інформація дозволить ідентифікувати сезонні та місячні тренди інвестування в облігації внутрішньої державної позики. Певні місяці можуть виявитися більш активними, це в свою чергу може бути пов'язано з фінансовими циклами, податковими сезонами, політичними або навіть військовими змінами. Такий аналіз допоможе зрозуміти, як різні зовнішні фактори впливають на ринок облігацій.

Мати такі дані також цінно, наприклад для планування випуску нових облігацій та побудови прогнозів доходу від їх продажу. Далі ця інформація може бути використана для прийняття обґрунтованих рішень щодо управління державним боргом.

І врешті-решт, відслідковуючи зміни кількості придбаних ОВДП від

місяця до місяця, можна розуміти загальні тенденції впевненості та настроїв інвесторів, що може бути одним із показником економічного клімату країни.

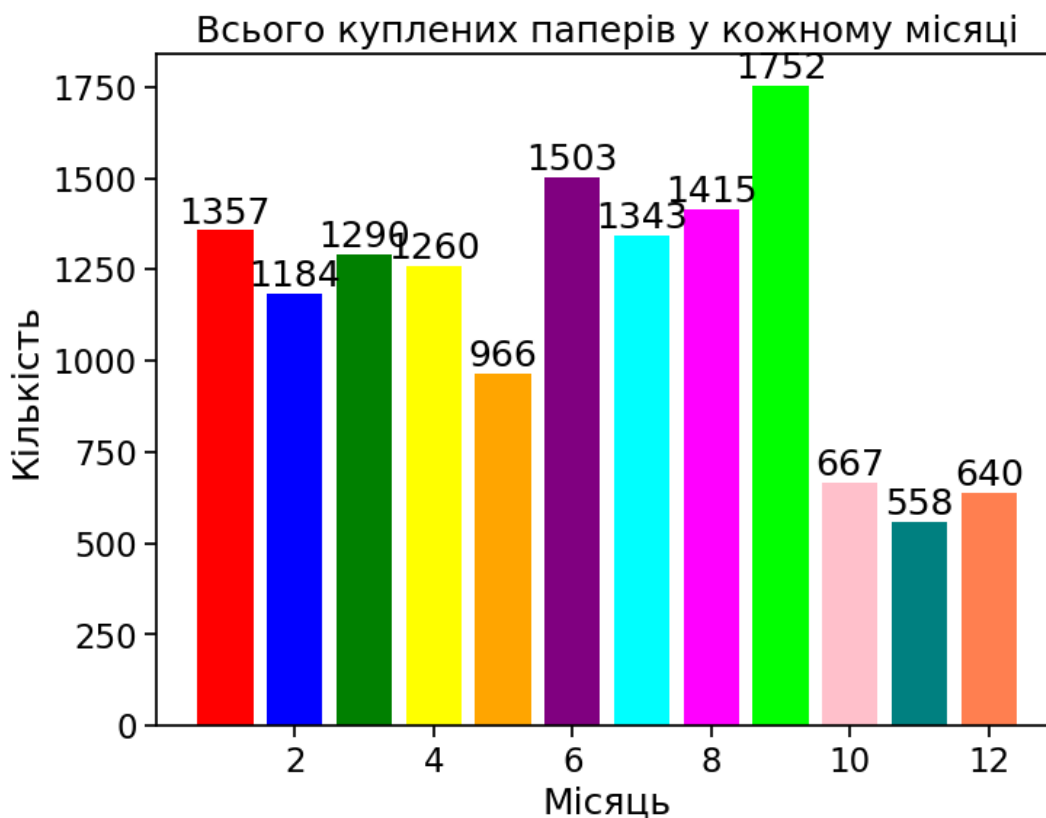


Рис. 2.3.10 Стовпчаста діаграма сумарної к-ті придбаних ОВДП у кожному місяці

Лістинг побудови графіка:

```
grouped_data = dataset.groupby('MONTHNUMBER')['QTY'].sum().reset_index()
monthNumber = grouped_data['MONTHNUMBER']
qty_sum = grouped_data['QTY']

plt.figure(figsize=(8, 6))
bars = plt.bar(monthNumber, qty_sum, color=colors)

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, int(yval),
             va='bottom', ha='center')

plt.xlabel('Місяць')
plt.ylabel('Кількість')
plt.title('Всього куплених паперів у кожному місяці')
plt.show()
```

2.4 Вибір, розробка та навчання моделі для прогнозування

На даному етапі ми перейдемо до процесу вибору оптимальної моделі, що буде використана для нашого нового модуля та її побудови з використанням у наших задачах прогнозування. Для розробки точної моделі прогнозування потрібно не лише ретельно обрати алгоритми, але й чітко розуміти набір даних із яким ми працюємо.

В процесі буде розглянуто декілька популярних алгоритмів машинного навчання, а саме:

- **Логістична Регресія (LR):** Здатність моделювати вірогідність подій.
- **Лінійний Дискримінантний Аналіз (LDA):** ефективний у знаходженні лінійної комбінації ознак, яка найкраще розділяє два або більше класи об'єктів або подій.
- **Метод Найближчих Сусідів (KNN):** використовується для інтуїтивно зрозумілого підходу до класифікації, базуючись на найближчих відомих даних.
- **Дерева Рішень (CART):** Здатність до моделювання нелінійних взаємозв'язків і легкість інтерпретації.
- **Наївний Баєсів Класифікатор (NB):** Ефективний у роботі з великими наборами даних і здатність до швидкого навчання.
- **Метод Опорних Векторів (SVM):** Гнучкий та ефективний у вирішенні як лінійних, так і нелінійних задач.
- **Метод Випадкового Лісу (Random Forest):** Висока точність і здатність до обробки великої кількості ознак без перенавчання.

Кожен з цих алгоритмів має ряд своїх переваг та недоліків, сценаріїв застосування. Наша мета полягає в тому, щоб порівняти та зрозуміти, який з них найкраще підійде для наших потреб. Буде проведено оцінку їх за декількома ключовими показниками, включаючи точність, складність моделі, час навчання та здатність до узагальнення.

В результаті такого аналізу ми визначимо найбільш ефективні алгоритми,

що підійдуть для наших задач пов'язаних із прогнозуванням.

Перейдемо до визначення точності алгоритмів машинного навчання на наших даних. Перш ніж виконати розподіл даних, потрібно їх перетворити із рядків у числовий тип даних.

```
TermType Mapping:
{'Long': 0, 'Medium': 1, 'Short': 2}

Currency Mapping:
{'EUR': 0, 'UAH': 1, 'USD': 2}
```

Рис. 2.4.1 Результат перетворення із рядків в числа

Виконаємо розподіл даних на навчальний та тестові набори даних. Після чого виконаємо перевірку точності кожного алгоритму і порівняємо їхні результати[30].

Лістинг порівняння точності обраних алгоритмів машинного навчання:

```
features = train_data.drop('TERMTYPE', axis=1)
target = train_data['TERMTYPE']

models = [
    ('LR', LogisticRegression(solver='liblinear', multi_class='ovr')),
    ('LDA', LinearDiscriminantAnalysis()),
    ('KNN', KNeighborsClassifier()),
    ('CART', DecisionTreeClassifier()),
    ('NB', GaussianNB()),
    ('SVM', SVC(gamma='auto')),
    ('RandomForest', RandomForestClassifier())
]

results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, features, target, cv=kfold,
    scoring='accuracy', error_score='raise')
    results.append(cv_results)
    training_time = time.time() - start_time
    names.append(name)
    print(f'{name}: {cv_results.mean():.3f} ({cv_results.std():.3f}) -
    Training Time: {training_time:.3f} sec')

fig, ax = plt.subplots()
colors = ['b', 'g', 'r', 'c', 'm', 'y', 'r']
```

```
ax.bar(names, [result.mean() for result in results], color=colors)

ax.set_xlabel('Алгоритм')
ax.set_ylabel('Точність')
ax.set_title('Порівняння алгоритмів ML для TERMTYPE')

plt.show()
```

Таблиця 2.4.1 Результати порівняння алгоритмів

Назва алгоритму	Результат точності (%)	Час навчання (с)	Складність
Логістична регресія	35%	0.100	Середня
Лінійний дискримінантний аналіз (LDA)	36%	0.057	Середня
Метод найближчих сусідів (KNN)	73%	0.086	Висока
Дерева рішень (CART)	100%	0.058	Середня
Наївний Бассів класифікатор (NB)	50%	0.048	Низька
Метод опорних векторів (SVM)	40%	0.205	Висока
Метод випадкового лісу (Random Forest)	99%	1.870	Висока

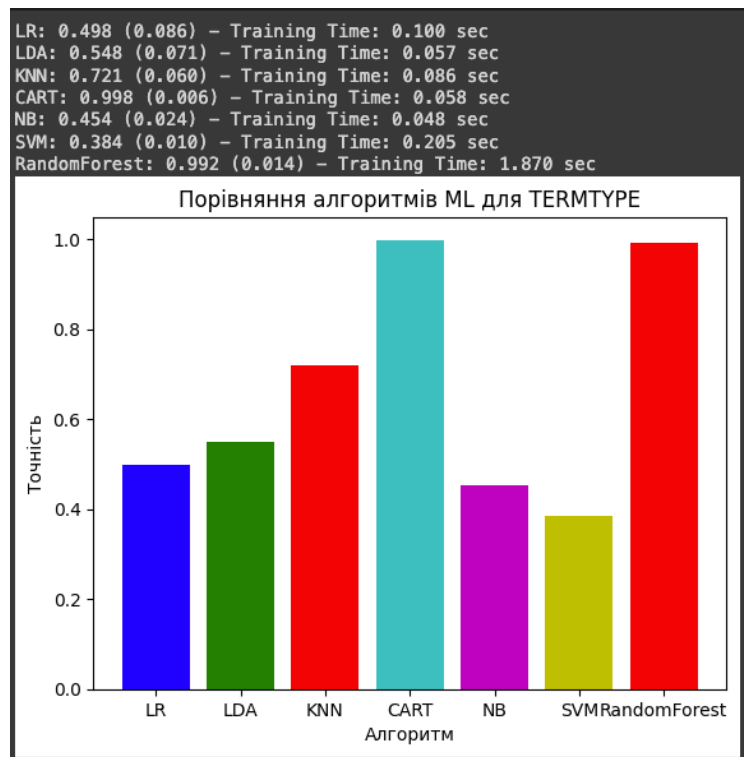


Рис. 2.4.2 Результати порівняння відображені точності на стовпчастій діаграмі

Отже, під час проведення аналізу ми використали різноманітні алгоритми машинного навчання для подальшого прогнозування терміну обігу цінного паперу, який буде придбано в наступній транзакції на основі даних. Серед моделей ML, які ми порівнювали – модель CART показала ідеальну точність 100%. Однак, не дивлячись на це, є важливіші аспекти, які слід врахувати перед вибором моделі для реального застосування в симуляторі.[33]

Одна із проблем CART полягає в перенавчанні моделі, особливо коли вона навчається на шумових або дуже специфічних даних. Це означає, що хоч і модель демонструє чудові результати на тренувальних даних, її продуктивність може значно погіршуватися на нових, раніше невідомих даних для цієї моделі даних.[13]

Не дивлячись на довгий час навчання, але алгоритм випадкового лісу (Random Forest) виявився менш схильним до перенавчання та забезпечує також високу точність. На противагу точність є трішечки нижча ніж у CART, але ця різниця є маленькою.

Алгоритм випадкового лісу використовує багато дерев прийняття рішень, кожне з яких навчається на випадковій підвибірці ознак і даних. В результаті чого забезпечує більш стабільні прогнози. Цей метод знижує ризик перенавчання, оскільки кожне із дерев в ансамблі має менший шанс навчатися на нерепрезентативних або випадкових шаблонах, що підвищує загальну надійність та точність прогнозування. В свою чергу – це впливає на час навчання, але точність прогнозу не буде різнитись кожного разу на різних наборах даних.[35]

Підсумовуючи, хоч модель CART і показала надзвичайно високі результати, але перевагу було надано саме алгоритмові випадкового лісу. Це є більш доцільним рішенням для довгострокової перспективи та надійного прогнозування. Особливо коли мова йде про ухилення від перенавчання та забезпечення стабільності роботи функціоналу прогнозування на різноманітних даних.[39]

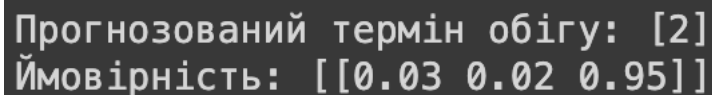
Лістинг прогнозування терміну обігу цінного паперу в наступній транзакції

```
label_encoder = LabelEncoder()
dataset['TERMTYPE'] = label_encoder.fit_transform(dataset['TERMTYPE'])

dataset['CURRENCY'] = label_encoder.fit_transform(dataset['CURRENCY'])
features_termType = dataset.drop(['TERMTYPE'], axis=1)
target_termType = dataset['TERMTYPE']
features_currency = dataset.drop(['CURRENCY'], axis=1)
target_currency = dataset['CURRENCY']

model_termType = RandomForestClassifier()
model_termType.fit(features_termType, target_termType)

next_data_termType = features_termType.tail(1)
predicted_termType = model_termType.predict(next_data_termType)
probability_termType = model_termType.predict_proba(next_data_termType)
print("Прогнозований термін обігу:", predicted_termType)
print("Ймовірність:", probability_termType)
print()
```



```
Прогнозований термін обігу: [2]
Ймовірність: [[0.03 0.02 0.95]]
```

Рис. 2.4.3 Отриманий результат прогнозу

Лістинг прогнозування терміну обігу з розподілом за періодичністю торгівлі користувача:

```
trading_frequencies = data_cleaned['TRADINGFREQUENCY'].unique()

def predict_termType_for_trading_frequency(data, frequency):
    frequency_data = data[data['TRADINGFREQUENCY'] == frequency]

    if frequency_data.empty or len(frequency_data) < 10:
        return f"TRADINGFREQUENCY {frequency}: Недостатньо даних"

    features = frequency_data.drop(['TERMTYPE'], axis=1)
    target = frequency_data['TERMTYPE']

    model = RandomForestClassifier(random_state=42)
    model.fit(features, target)

    predictions = model.predict(features)
    most_common_termType = max(set(predictions),
key=list(predictions).count)

    termType_label =
label_encoder.inverse_transform([most_common_termType])[0]
```

```

return f"TRADINGFREQUENCY {frequency}: {termType_label}"
trading_frequency_predictions =
[predict_termType_for_trading_frequency(data_cleaned, frequency) for
frequency in trading_frequencies]
trading_frequency_predictions

```

```

['TRADINGFREQUENCY 1: Long',
 'TRADINGFREQUENCY 2: Medium',
 'TRADINGFREQUENCY 3: Medium',
 'TRADINGFREQUENCY 0: Long']

```

Рис. 2.4.4 Результат прогнозування з розподілом за частотою торгівлі

Опис числових полів частоти торгівлі: 0 – рідко; 1 – кожного місяця; 2 – щотижня; 3 – кожного дня.

Отже, ми отримали результат, що користувачі, які будуть займатись торгівлею кожного дня, або кожного тижня нададуть перевагу середньостроковим облігаціям. В той час, як ті хто буде щомісячно або рідко виберуть цінні папери з довгостроковим терміном обігу.

Підсумовуючи, було порівняння моделей для нашого майбутнього прогнозування. Отримано результати й обрано найкращі моделі машинного навчання, які будуть використані. Було зроблено також опис, чому надано перевагу алгоритмові випадкових лісів, а не методу CART.

Висновки до другого розділу

В цьому розділі було представлено способи використання нашого модуля у системі симуляції для максимального реалізму процесу роботи з ОВДП у віртуальному світі з проведенням аналітики взаємодії користувачів та прогнозуванням їхньої поведінки у симуляторі.

Розроблено функціонал для обробки даних, їх потрібно було правильно сформувати та відфільтрувати. Визначено 3 основні колекції, які будуть використовуватись для аналізу та проведення прогнозувань.

Далі було проведено аналіз зібраних даних, використовуючи сучасні методи мови програмування Python з використанням необхідних додаткових модулів для спрощення роботи.

На основі цього ми зробили вибір моделі машинного навчання та навчили моделі для прогнозування, яку буде використано у нашому додатку. Цей процес включав тестування різних алгоритмів машинного навчання. Після того, як було обрано необхідний алгоритм, то було проведено його налаштування.

РОЗДІЛ 3 ДОСЛІДЖЕННЯ ПРАЦЕЗДАТНОСТІ ТА ОЦІНКИ ЕФЕКТИВНОСТІ МОДЕЛІ ПРОГНОЗУВАННЯ

3.1 Оцінка та аналіз якості моделі

Розробка моделі машинного навчання зазвичай включає три основних кроки. На першому етапі виконується тренування моделі за допомогою спеціально відібраного набору даних, що відповідає задачі для якої вона розробляється. Наступним етапом є процес навчання, модель регулярно перевіряється на вибірці з цих же даних, щоб можна було оцінити її ефективність у роботі з невідомими даними. На останньому етапі, після завершення навчання, модель піддається випробуванню на окремому тестовому наборі даних, де і визначаються остаточні показники її продуктивності. Ці метрики, незалежно від їх типу, надають інформацію про те, наскільки ефективно модель справляється з класифікацією на тестових даних.

Метрики, які будуть розглянуті під час аналізу якості моделі:

1. Акуратність – зазвичай використовується, щоб розрахувати частку правильних прогнозів від загального числа;
2. Точність вимірювань – вказує, яка частка із усіх позитивних прогнозів була правильна;
3. Оцінка F1 – вимірювання точності моделі. Цей показник балансує між точністю та чутливістю. Високий рівень F1 означає, що модель точно ідентифікує більшість позитивних випадків, роблячи при цьому малу кількість помилок;
4. Повнота – вказує на частку коректних позитивних прогнозів від всіх позитивних результатів, що зробила б наша модель.

Спочатку ми почнемо із проведення аналізу для прогнозу терміну обігу в цілому між усіма користувачами.

```
TermType Mapping:  
{ 'Long': 0, 'Medium': 1, 'Short': 2 }
```

Рис. 3.1.1 Трансформування рядкових значень у числові

Після проведення прогнозу, ми отримали результат, що з ймовірністю у 77% наступним буде придбано довгостроковий цінний папір

Прогнозований термін обігу: [0]
Ймовірність: [[0.77670322 0.16245019 0.06084659]]

Рис. 3.1.2 Результат прогнозу

Оцінка прогнозу за терміном дії:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	220
1	1.00	1.00	1.00	142
2	1.00	1.00	1.00	199
accuracy			1.00	561
macro avg	1.00	1.00	1.00	561
weighted avg	1.00	1.00	1.00	561
Confusion Matrix для прогнозу терміну дії:				
[[220 0 0]				
[0 142 0]				
[0 0 199]]				

Рис. 3.1.3 Результати метрик точності прогнозу та матриці плутанини

Аналізуючи матрицю плутанини можемо сказати, що не було жодної помилкової класифікації між класами, адже значення поза діагоналями рівні нулю. Також значення на діагоналі (220, 142, 199) вказують на кількість правильно класифікованих випадків для кожного із класів.[33]

Далі перейдемо до проведення аналізу прогнозування цінного паперу з використанням інформації про користувача, а саме його досвід в інвестиціях.

Виконавши цей прогноз – ми можемо побачити обрані діапазони за роками, тип, а далі бачимо метрики аналізу про які розповідалось раніше.

```
Prediction for termType by user experience
1 - 10: Short, Accuracy: 0.9242424242424242, Recall: 0.9242424242424242, Precision: 0.9242424242424242, F1 Score: 0.9242424242424242
10 - 20: Long, Accuracy: 0.9310344827586207, Recall: 0.9310344827586207, Precision: 0.9310344827586207, F1 Score: 0.9310344827586207
20 - 30: Long, Accuracy: 0.8695652173913043, Recall: 0.8695652173913043, Precision: 0.8695652173913043, F1 Score: 0.8695652173913043
30 - 50: Medium, Accuracy: 0.6842105263157895, Recall: 0.6842105263157895, Precision: 0.6842105263157895, F1 Score: 0.6842105263157895
```

Рис. 3.1.4 Прогноз та результат його точності

Перейдемо до перевірки точності прогнозу для терміну цінного паперу для людей за різною частотою торгівлі: 0 – рідко; 1 – кожного місяця; 2 – щотижня; 3 – кожного дня.

```
['TRADINGFREQUENCY 2: Short, Accuracy: 0.7291666666666666, Recall: 0.7291666666666666, Precision: 0.7291666666666666, F1 Score: 0.7291666666666666',
'TRADINGFREQUENCY 1: Short, Accuracy: 0.8611111111111112, Recall: 0.8611111111111112, Precision: 0.8611111111111112, F1 Score: 0.8611111111111112',
'TRADINGFREQUENCY 0: Short, Accuracy: 0.9166666666666666, Recall: 0.9166666666666666, Precision: 0.9166666666666666, F1 Score: 0.9166666666666666',
'TRADINGFREQUENCY 3: Long, Accuracy: 0.8421052631578947, Recall: 0.8421052631578947, Precision: 0.8421052631578947, F1 Score: 0.8421052631578947']
```

Рис. 3.1.5 Результат прогнозу та його точності

В результаті перевірки точність нашої моделі прогнозування, ми отримали

досить високі показники за обраними метрикам, що засвідчує її надійність у передбаченні.

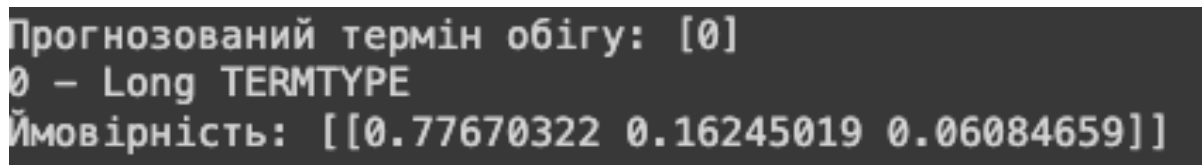
В цьому розділі було проведено оцінку якості прогнозів. В результаті чого, варто відмітити, що всі розглянуті прогнози показали досить високу точність. Це свідчить про ефективність обраних методів машинного навчання. Також ці результати вказують на значний потенціал для майбутнього розвитку та розширення функціоналу нашого симулятора.

3.2 Тестування роботи моделі машинного навчання в процесі прогнозування

На даному етапі ми зосередимося на проведенні тестування раніше розроблених моделей машинного навчання. Буде проведено різні експерименти для перевірки, наскільки точно моделі передбачають результати у різних умовах і як ці прогнози корелюються із реальними даними. В результаті таких підходів ми зможемо визначити сильні сторони та обмеження кожної моделі, а також надати глибший аналіз їх практичної застосовності.

В даний момент наші моделі розроблені і також було проведено аналіз точності цих моделей, але потрібно їх також перевірити і провести видозміни в даних. Ми будемо видозмінювати наші дані задля перевірки точності наших прогнозів та стабільності роботи обраних моделей.

Перший прогноз із якого ми розпочнемо перевірку – це прогнозування терміну обігу наступного цінного паперу.



```
Прогнозований термін обігу: [0]
0 - Long TERMType
Ймовірність: [[0.77670322 0.16245019 0.06084659]]
```

Рис. 3.2.1 Результат прогнозу

Ми отримали результат, що наступним буде придбано цінний папір із довгостроковим терміном обігу. Так, дійсно, цей прогноз є цілком правдивим. Адже після проведення аналізу даних і підрахунку скільки відсотків є кожного із типів терміну обігу в наших даних, переважно більша кількість цінних паперів була придбана юзерами з довгостроковими типом (Рис.3.2.2).



Рис. 3.2.2. Розподіл за терміном придатності

Далі перейдемо до тестування нашого прогнозу з урахуванням досвіду в інвестиціях. Такий прогноз дасть нам можливість розуміння, яким облігаціям надають інвестори з різною кількістю років досвіду в сфері інвестування.

```
['1 - 10: Short', '10 - 20: Long', '20 - 30: Long', '30 - 50: Medium']
```

Рис. 3.2.1 Результат прогнозу

Ми отримали прогноз з розподілом за 4 діапазонами. Задля перевірки цієї точності дані були сфабриковані, щоб ми могли точно оцінити якість та точність прогнозу.

Так цей прогноз є дійсно коректним, адже в даному випадку було запрограмовано генератор, щоб він для кожного із діапазонів купував переважно один певний тип обігу ОВДП.

Також, ми бачимо, що для перших двох діапазонів точність прогнозу є більше 90%, а для 2 та 3 діапазону вона є менше – це пов'язано із тим, що серед наших даних переважна кількість користувачів із досвідом в сфері інвестицій від 1 до 20 років.

На наступному Рис. 3.2.2 буде продемонстровано найбільше популярний термін обігу для кожного діапазону та його кількість.

```
range  termtype  qty
['1 - 10: Short, 185',
 '10 - 20: Long, 133',
 '20 - 30: Long, 36',
 '30 - 50: Medium, 50']
```

Рис. 3.2.2 Результат підрахунку кількості цінних паперів для кожного із діапазону

Перейдемо до проведення тестування наступного прогнозу, а саме прогнозу, яку кількість цінних паперів буде придбано у наступній транзакції. Цей показник є досить цікавим, адже можна виконувати фіксацію цього показника у різні моменти часу і відслідковувати зміни.

Наприклад, в якийсь момент часу кількість почне різко зростати або навпаки падати, то це буде сигналом для проведення детального аналізу та розуміння чому саме так.

Кількість паперів у наступній покупці: 15 шт.

Рис. 3.2.3 Результат прогнозу

Наступним у процесі тестування є прогноз із розподілом за рівнем до ризику. Такий прогноз надає розуміння про відношення різних рівнів ризику до ринкових трендів, інвестиційних рішень. Користувачі з різною толерантністю до ризику мають різні інвестиційні стратегії, які в свою чергу й формують загальну динаміку інвестиційного ринку.

Рівні схильності до ризику:

0. Низький
1. Середній
2. Високий

```
['RISKTOLERANCE 0: Short, Accuracy: 0.8507462686567164, Recall: 0.8507462686567164, Precision: 0.8507462686567164, F1 Score: 0.8507462686567164',
 'RISKTOLERANCE 1: Short, Accuracy: 0.813953488372093, Recall: 0.813953488372093, Precision: 0.813953488372093, F1 Score: 0.8139534883720931',
 'RISKTOLERANCE 2: Short, Accuracy: 0.8833333333333333, Recall: 0.8833333333333333, Precision: 0.8833333333333333, F1 Score: 0.8833333333333333']
```

Рис. 3.2.4 Результат прогнозу з розподілом за рівнем до ризику.

Перейдемо до прогнозу з розподілом за статтю. Такий прогноз дасть можливість оцінити, яким цінним паперам віддають перевагу чоловіки та жінки

та як кожен гендер відносяться до інвестування в цілому.

```
['GENDER 1: Short, Accuracy: 0.90625, Recall: 0.90625, Precision: 0.90625, F1 Score: 0.90625',
 'GENDER 0: Long, Accuracy: 0.8648648648648649, Recall: 0.8648648648648649, Precision: 0.8648648648648649, F1 Score: 0.8648648648648649']
```

Рис. 3.2.5 Результат прогнозування типу цінного паперу за терміном обігу

Цей прогноз має досить високі показники точності, але все-таки було виконано перевірку використовуючи аналіз даних. І зроблено розподіл який цінним паперам надає перевагу кожна із статей.

	GENDER	TERMTYPE	counts
0	female	Long	88
1	female	Medium	72
2	female	Short	84
3	male	Long	93
4	male	Medium	89
5	male	Short	135

Рис. 3.2.6 Результат розподілу за статтю

В результаті такого аналізу, ми дійсно бачимо, що ніяких аномалій у нас не виникає і жінки надають трішки більшу перевагу для довгострокового інвестування в той час, як чоловіки надають значно більшу перевагу короткостроковим інвестицій. І це є досить цікавий прогноз, адже такий прогноз є поштовхом до глибокого вивчення ситуації з подальшим розумінням, який фактор впливає на кожную із статей – обізнаність в сфері цінних паперів чи суто психологічний чинник.

Висновки до третього розділу

Під час написання третього розділу кваліфікаційної роботи – було проведено оцінку розроблених моделей машинного навчання та тестування зроблених прогнозів у розробленому модулі.

Для проведення оцінювання було обрано метрики, а саме акуратність, повноту, точність та оцінку F1. Результати аналізів показали досить високий рівень, що демонструється високими показниками точності та оцінки F1. В свою чергу це свідчить про те, що використані алгоритми машинного навчання було

добре навчено та налаштовано, щоб виявляти складні закономірності в даних. Це дає високі можливості для подальшого використання та вдосконалення програмного комплексу для роботи з облігаціями внутрішньої державної позики.

Провівши всі аналізи над розробленими моделями, ми перейшли до перевірки наших прогнозів задля уникнення нереалістичності та виявлення аномалій, які могли б відбутись в процесі прогнозування.

Прогнози було проведено на різноманітних даних. І це в свою чергу виявило, що моделі здатні ефективно адаптуватися до нових ситуацій та появи нових записів, що свідчить про їхню універсальність, гнучкість. Така гнучкість та адаптивність є досить цінною в умовах швидкозмін ринку та даних про взаємодію із симулятором.

Підсумовуючи, дослідження підтвердили наявний значний потенціал машинного навчання у сфері облігацій внутрішньої державної позики. Одночасно підкреслюючи важливість подальших досліджень та розвитку в цій області для досягнення ще більшої точності та надійності прогнозів.

ВИСНОВКИ

В процесі написання кваліфікаційної магістерської роботи було вирішено актуальне завдання із розробки модуля для можливості проведення аналізу даних та прогнозування поведінки користувачів.

Відштовхуючись від цього було зроблено обґрунтування актуальності теми, що дало змогу зрозуміти цінність, основні аспекти розробки нового модуля із використанням засобів аналізу даних та штучного інтелекту. Адже цей модуль в майбутньому має шанс стати особливим інструментом, який буде використовуватись для підвищення фінансової грамотності та підготовки користувачів до викликів у сфері фінансів та інвестицій.

Проведено огляд аналогів і виявлено їх переваги та недоліки. Це дало нам розуміння, що серед наявного функціоналу має бути виконання прогнозів та проведення аналізу поведінки користувачів. В свою чергу, це відкрило нам додаткові можливості для вдосконалення нашого нового сервісу.

Відібрано 7 найбільш перспективних моделей машинного навчання, а саме: логістична регресія – здатна моделювати вірогідність подій; лінійний дискримінантний аналіз – ефективний у знаходженні лінійної комбінації ознак; метод найближчих сусідів – базується на найближчих відомих даних; метод дерева рішень – здатний до моделювання нелінійних взаємозв'язків; наївний Баєсів класифікатор – ефективний у роботі з великими наборами даних; метод опорних векторів – гнучкий та ефективний у вирішенні як лінійних, так і нелінійних задач; метод випадкового лісу – має високу точність і здатність до обробки великої кількості ознак без перенавчання.

Проведено теоретичний опис основних алгоритмів машинного навчання, які були відібрані для подальшого процесу аналізу та використані в новому модулі під час розробки функціоналу прогнозування поведінки та аналізу даних.

Зроблено вибір інструментів, засобів та вимог до ПЗ, що дало можливість розробки і впровадження нового модуля, але й оптимального використання ресурсів з максимальною ефективністю роботи програмного комплексу.

Використовуючи сучасні методи мови програмування Python із необхідними модулями як: Matplotlib, Seaborn, Pandas ми перейшли до процесу аналізу даних, що дав можливість зрозуміти, яку кількість цінних паперів придбали окремо чоловіки та жінки, скільки цінних паперів придбано у кожному місяці, відсотковий розподіл за кожним терміном обігу та кількість придбаних цінних паперів у кожній валюті.

Провівши аналіз відібраних моделей машинного навчання ми отримали результати й побачили, що логістична регресія, лінійний дискримінантний аналіз, наївний баєсів класифікатор та метод опорних векторів показали низький рівень точності – до 50%, що є досить низьким показником.

В свою чергу, алгоритм дерева прийняття рішень та випадкового лісу показали найвищі показники точності, а саме: 100% для дерева ухвалення рішень і 99% для випадкового лісу. Внаслідок наявного недоліку у перенавчанні, що присутній в дереві рішень, було надано перевагу алгоритмові випадкових лісів з огляду на його стійкість до перенавчання та досить високу точність.

Наступним етапом було розроблено функціонал для прогнозування поведінки користувачів з використанням методу випадкових лісів. Побудовані прогнози відкрили додаткові можливості для розуміння, якому терміну обігу та кількості цінних паперів буде надано перевагу у наступній транзакції. Також прогнози терміну обігу цінних паперів із розподілом за досвідом в інвестиціях, частоті трейдингу та відношенні до ризику.

На завершення проведено тестування розробленого модуля аналізу даних та прогнозування поведінки на різних наборах даних задля виявлення проблем, помилок. Тестування дало розуміння, що розроблені моделі досить ефективно адаптуються до нових даних, що свідчить про їхню універсальність.

Отже, в результаті написання магістерської кваліфікаційної роботи ми отримали працездатний модуль, який надає нові можливості для проведення аналізу та прогнозування поведінки користувачів. Розроблений модуль є розширенням наявного функціоналу симулятора для роботи з облігаціями внутрішньої державної позики.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вакалюк, Т.А., Лабенський, В.А., 2022. Використання симуляторів для навчання людей новим навичкам або вдосконалення наявних навичок. У: Наукова молодь-2022, 15 листопада 2022, Київ, с. 44-45.
2. Вакалюк, Т.А., Лабенський, В.А., 2022. Використання симулятора для навчання процесу інвестування. У: Тези V Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 01–02 грудня 2022 р. Житомир: Житомирська політехніка, с. 300-301.
3. Антонюк, Д.С., Лабенський, В.А., Мєдведєв, В.В., Новачук, Р.О., 2022. Проектування симулятора управління персональним портфелем облігацій внутрішньої державної позики. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, с. 43-52.
4. Документація JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
5. Офіційна документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>.
6. Syed Fahad Ahmed, 2022. Evolution of web development via machine learning. URL: <https://itchronicles.com/artificial-intelligence/evolution-of-webdevelopment-with-machine-learning/>
7. Web Development Technologies. URL: <https://www.orientsoftware.com/technologies/web-technologies/>
8. Seldon, 2021. Machine Learning Regression Explained. URL: <https://www.seldon.io/machine-learning-regression-explained>
9. Müller, A.C. & Guido, S., 2016. Introduction to Machine Learning with Python. O'Reilly Media.
10. Kelleher, J.D., Mac Namee, B. & D'Arcy, A., 2015. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. The MIT Press.

11. Witten, I.H., Frank, E., Hall, M.A. & Pal, C.J., 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
12. Conway, D. & White, J.M., 2012. Machine Learning for Hackers. O'Reilly Media.
13. Moroney, L., 2020. AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence. O'Reilly Media.
14. Harrington, P., 2012. Machine Learning in Action. Manning Publications.
15. Bruce, P. & Bruce, A., 2017. Practical Statistics for Data Scientists: 50 Essential Concepts. O'Reilly Media.
16. Burkov, A., 2019. The Hundred-Page Machine Learning Book. Self-published.
17. Cairo, A., 2013. The Functional Art: An Introduction to Information Graphics and Visualization. New Riders.
18. Few, S., 2013. Information Dashboard Design: Displaying Data for At-a-glance Monitoring. Analytics Press.
19. Dykes, B., 2020. Effective Data Storytelling: How to Drive Change with Data, Narrative, and Visuals. Wiley.
20. Rendgen, S. & Wiedemann, J., 2012. Information Graphics. Taschen.
21. O'Neil, C., 2016. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. Crown.
22. James, G., Witten, D., Hastie, T. & Tibshirani, R., 2013. An Introduction to Statistical Learning: with Applications in R. Springer.
23. Topol, E., 2019. Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. Basic Books.
24. Bird, S., Klein, E. & Loper, E., 2009. Natural Language Processing with Python. O'Reilly Media.
25. Russell, S. & Norvig, P., 2016. Artificial Intelligence: A Modern Approach. Pearson.
26. Heaton, J., 2015. Artificial Intelligence for Humans, Volume 1: Fundamental Algorithms. Heaton Research, Inc.

27. Nilsson, N.J., 1998. Artificial Intelligence: A New Synthesis. Morgan Kaufmann.
28. Bostrom, N., 2014. Superintelligence: Paths, Dangers, Strategies. Oxford University Press.
29. Theobald, O., 2021. Machine Learning for Absolute Beginners. Scatterplot Press.
30. Russel, S.J. & Norvig, P., 2015. Artificial Intelligence: A Modern Approach. Pearson.
31. Mitchell, T.M., 1997. Machine Learning. McGraw-Hill Education.
32. Chollet, F., 2017. Deep Learning with Python. Manning Publications.
33. Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
34. Kelleher, J.D., Mac Namee, B. & D'Arcy, A., 2015. Fundamentals of Machine Learning for Predictive Data Analytics. The MIT Press.
35. Witten, I.H., Frank, E., Hall, M.A. & Pal, C.J., 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
36. Conway, D. & White, J.M., 2012. Machine Learning for Hackers. O'Reilly Media.
37. Moroney, L., 2020. AI and Machine Learning For Coders: A Programmer's Guide to Artificial Intelligence. O'Reilly Media.
38. Harrington, P., 2012. Machine Learning in Action. Manning Publications.
39. Wachter-Boettcher, S., 2017. Technically Wrong: Sexist Apps, Biased Algorithms, and Other Threats of Toxic Tech. W. W. Norton & Company.
40. Molnar, C., 2020. Interpretable Machine Learning. Lulu.com.
41. Domingos, P., 2015. The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books.
42. Marcus, G. & Davis, E., 2019. Rebooting AI: Building Artificial Intelligence We Can Trust. Pantheon Books.

ДОДАТКИ

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score,
precision_score, f1_score, roc_auc_score
from sklearn.metrics import confusion_matrix

file_path = 'termtype_with_experience.csv'
data = pd.read_csv(file_path)

data_cleaned = data.drop(['EDUCATIONLEVEL', 'PROFESSION', '_ID',
                          'USERID', 'MONTHNAME', 'BUYDATE', 'CODE', 'CURRENCY'],
                          axis=1)

label_encoder = LabelEncoder()
for col in ['RISKTOLERANCE', 'TRADINGFREQUENCY', 'GENDER', 'TERMTYPE']:
    data_cleaned[col] = label_encoder.fit_transform(data_cleaned[col])

print(data_cleaned)

def predict_termType_for_range(data, min_exp, max_exp):
    range_data = data[(data['INVESTMENTEXPERIENCE'] >= min_exp) &
                      (data['INVESTMENTEXPERIENCE'] < max_exp)]

    if range_data.empty or len(range_data) < 10:
        return f"{min_exp} - {max_exp}: Not enough data length"

    target = range_data['TERMTYPE']
    features = range_data.drop(['TERMTYPE'], axis=1)

    X_train, X_test, y_train, y_test = train_test_split(features, target,
                                                          test_size=0.3, random_state=42)

    model = RandomForestClassifier(random_state=13)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    y_pred_proba = model.predict_proba(X_test)[:, 1]

    accuracy = accuracy_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred, average='micro')
    precision = precision_score(y_test, y_pred, average='micro')
    f1 = f1_score(y_test, y_pred, average='micro')

    most_common_termType = max(set(y_pred), key=list(y_pred).count)
    termType_label =
label_encoder.inverse_transform([most_common_termType])[0]

    return (f"{min_exp} - {max_exp}: {termType_label}, "
            f"Accuracy: {accuracy}, Recall: {recall}, Precision:
{precision}, "
            f"F1 Score: {f1}")

```

```

investment_ranges = [(1, 10), (10, 20), (20, 30), (30, 50)]
predictions = [predict_termType_for_range(data_cleaned, min_exp, max_exp)
for min_exp, max_exp in investment_ranges]
predictions

def aggregate_data(data, min_exp, max_exp):
    filtered_data = data[(data['INVESTMENTEXPERIENCE'] >= min_exp) &
(data['INVESTMENTEXPERIENCE'] < max_exp)]

    if filtered_data.empty:
        return f"{min_exp} - {max_exp}: No data"

    termType_counts = filtered_data['TERMTYPE'].value_counts()
    if not termType_counts.empty:
        most_common_termType = termType_counts.idxmax()
        termType_count = termType_counts.max()
        return f"{min_exp} - {max_exp}: {most_common_termType},
{termType_count}"
    else:
        return f"{min_exp} - {max_exp}: No common termType"

investment_ranges = [(1, 10), (10, 20), (20, 30), (30, 50)]
results = [aggregate_data(data, min_exp, max_exp) for min_exp, max_exp in
investment_ranges]

print("range  termtype  qty")
results

trading_frequencies = data_cleaned['TRADINGFREQUENCY'].unique()

def predict_termType_for_trading_frequency(data, frequency):
    frequency_data = data[data['TRADINGFREQUENCY'] == frequency]

    if frequency_data.empty or len(frequency_data) < 10:
        return f"TRADINGFREQUENCY {frequency}: Недостатньо даних"

    features = frequency_data.drop(['TERMTYPE'], axis=1)
    target = frequency_data['TERMTYPE']

    X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.3, random_state=50)

    model = RandomForestClassifier(random_state=50)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    y_pred_proba = model.predict_proba(X_test)[:, 1]  # Для розрахунку
ROC AUC

    accuracy = accuracy_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred, average='micro')
    precision = precision_score(y_test, y_pred, average='micro')
    f1 = f1_score(y_test, y_pred, average='micro')

```

```

# roc_auc = roc_auc_score(y_test, y_pred_proba)

most_common_termType = max(set(y_pred), key=list(y_pred).count)
termType_label =
label_encoder.inverse_transform([most_common_termType])[0]

conf_matrix = confusion_matrix(y_test, y_pred)

return (f"TRADINGFREQUENCY {frequency}: {termType_label}, "
        f"Accuracy: {accuracy}, Recall: {recall}, Precision:
{precision}, "
        f"F1 Score: {f1}")

trading_frequency_predictions =
[predict_termType_for_trading_frequency(data_cleaned, frequency) for
frequency in
                                trading_frequencies]

trading_frequency_predictions

trading_frequencies = data_cleaned['RISKTOLERANCE'].unique()

def predict_termType_for_trading_frequency(data, frequency):
    frequency_data = data[data['RISKTOLERANCE'] == frequency]

    if frequency_data.empty or len(frequency_data) < 10:
        return f"RISKTOLERANCE {frequency}: Недостатньо даних"

    features = frequency_data.drop(['TERMTYPE'], axis=1)
    target = frequency_data['TERMTYPE']

    X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.3, random_state=50)

    model = RandomForestClassifier(random_state=50)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    y_pred_proba = model.predict_proba(X_test)[:, 1] # Для розрахунку
ROC AUC

    accuracy = accuracy_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred, average='micro')
    precision = precision_score(y_test, y_pred, average='micro')
    f1 = f1_score(y_test, y_pred, average='micro')
    # roc_auc = roc_auc_score(y_test, y_pred_proba)

    most_common_termType = max(set(y_pred), key=list(y_pred).count)
    termType_label =
label_encoder.inverse_transform([most_common_termType])[0]

    conf_matrix = confusion_matrix(y_test, y_pred)

    return (f"RISKTOLERANCE {frequency}: {termType_label}, "
            f"Accuracy: {accuracy}, Recall: {recall}, Precision:
{precision}, "

```

```

        f"F1 Score: {f1}")

trading_frequency_predictions =
[predict_termType_for_trading_frequency(data_cleaned, frequency) for
frequency in
                                trading_frequencies]
trading_frequency_predictions

trading_frequencies = data_cleaned['GENDER'].unique()

def predict_termType_for_trading_frequency(data, frequency):
    frequency_data = data[data['GENDER'] == frequency]

    if frequency_data.empty or len(frequency_data) < 10:
        return f"GENDER {frequency}: Недостатньо даних"

    features = frequency_data.drop(['TERMTYPE'], axis=1)
    target = frequency_data['TERMTYPE']

    X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.3, random_state=50)

    model = RandomForestClassifier(random_state=50)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    y_pred_proba = model.predict_proba(X_test)[:, 1] # Для розрахунку
ROC AUC

    accuracy = accuracy_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred, average='micro')
    precision = precision_score(y_test, y_pred, average='micro')
    f1 = f1_score(y_test, y_pred, average='micro')

    most_common_termType = max(set(y_pred), key=list(y_pred).count)
    termType_label =
label_encoder.inverse_transform([most_common_termType])[0]

    conf_matrix = confusion_matrix(y_test, y_pred)

    return (f"GENDER {frequency}: {termType_label}, "
            f"Accuracy: {accuracy}, Recall: {recall}, Precision:
{precision}, "
            f"F1 Score: {f1}")

trading_frequency_predictions =
[predict_termType_for_trading_frequency(data_cleaned, frequency) for
frequency in
                                trading_frequencies]

trading_frequency_predictions

print(data.groupby(['GENDER',
'TERMTYPE']).size().reset_index(name='counts'))

```

ДОДАТОК Б

```

dataset = read_csv('main.csv')
print(dataset.head(10))

dataset = dataset.drop(['_ID', 'USERID', 'BUYDATE'], axis=1)
print(dataset.head(10))

print(dataset.info())

print(dataset.describe())

print(dataset.groupby('TERMTYPE')['QTY'].sum())

print(dataset.groupby('CURRENCY')['QTY'].sum())

sns.kdeplot(data=dataset, x="BUYPRICE", hue="TERMTYPE", fill=True)

pyplot.xlabel('Варість')
pyplot.ylabel('Щільність')
pyplot.title('Графік щільності вартості до типу погашення')

pyplot.show()

selected_columns = ['QTY', 'BUYPRICE', 'MONTHNUMBER', 'CURRENCY',
                    'TERMTYPE', 'YEARCOUPON', 'Y', 'YTM']
selected_data = dataset[selected_columns]

numeric_columns = ['QTY', 'BUYPRICE', 'MONTHNUMBER', 'YEARCOUPON', 'Y',
                  'YTM']
selected_data[numeric_columns] =
selected_data[numeric_columns].apply(pd.to_numeric, errors='coerce')

corr_matrix = selected_data.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

total_qty = dataset['QTY'].sum()
percentage = (dataset.groupby('TERMTYPE')['QTY'].sum() / total_qty) * 100

pyplot.pie(percentage, labels=percentage.index, autopct='%1.2f%%')
pyplot.title('Відсоткове співвідношення кількості облігацій за терміном')
pyplot.show()

total_qty = dataset['QTY'].sum()
percentage = (dataset.groupby('CURRENCY')['QTY'].sum() / total_qty) * 100
pyplot.pie(percentage, labels=percentage.index, autopct='%1.1f%%')
pyplot.title('Відсоткове співвідношення кількості облігацій за валютою паперів')
pyplot.show()

print(dataset.groupby(['MONTHNUMBER'])['QTY'].sum().reset_index(name='TOTAL_QTY'))

```

```

colors = ['red', 'blue', 'green', 'yellow', 'orange', 'purple', 'cyan',
'magenta', 'lime', 'pink', 'teal', 'coral']

grouped_data = dataset.groupby('MONTHNUMBER')['QTY'].sum().reset_index()
monthNumber = grouped_data['MONTHNUMBER']
qty_sum = grouped_data['QTY']

plt.figure(figsize=(8, 6))
bars = plt.bar(monthNumber, qty_sum, color=colors)

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval, int(yval),
va='bottom', ha='center')

plt.xlabel('Місяць')
plt.ylabel('Кількість')
plt.title('Всього куплених паперів у кожному місяці')
plt.show()

grouped_data = dataset.groupby(['CURRENCY',
'MONTHNUMBER'])['QTY'].sum().reset_index()

currency = grouped_data['CURRENCY'].unique()
monthNumber = grouped_data['MONTHNUMBER'].unique()

qty_sum = []
for cur in currency:
    cur_qty = []
    for month in monthNumber:
        qty = grouped_data[(grouped_data['CURRENCY'] == cur) &
(grouped_data['MONTHNUMBER'] == month)]['QTY'].values
        cur_qty.append(qty[0] if len(qty) > 0 else 0)
    qty_sum.append(cur_qty)

width = 0.15
x = np.arange(len(monthNumber))
fig, ax = plt.subplots()
bars = []
for i, cur in enumerate(currency):
    bars.append(ax.bar(x + i * width, qty_sum[i], width, label=cur))

ax.set_xlabel('Місяць')
ax.set_ylabel('Сумарна к-ть куплених паперів')
ax.set_title('Сумарна к-ть куплених паперів за валютою кожному місяці')
ax.set_xticks(x)
ax.set_xticklabels(monthNumber)
ax.legend()

for bar in bars:
    for rect in bar:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom')

```

```

plt.tight_layout()
plt.show()

label_encoder = LabelEncoder()
dataset['TERMTYPE'] = label_encoder.fit_transform(dataset['TERMTYPE'])
termType_mapping = dict(zip(label_encoder.classes_,
label_encoder.transform(label_encoder.classes_)))
dataset['CURRENCY'] = label_encoder.fit_transform(dataset['CURRENCY'])
currency_mapping = dict(zip(label_encoder.classes_,
label_encoder.transform(label_encoder.classes_)))

print("TermType Mapping:")
print(termType_mapping)
print("Currency Mapping:")
print(currency_mapping)

dataset = dataset.drop(['CODE', 'MONTHNAME'], axis=1)

train_data, test_data = train_test_split(dataset, test_size=0.15,
random_state=42)

print(train_data.tail())

print(test_data.head())

import time

features = train_data.drop('TERMTYPE', axis=1)
target = train_data['TERMTYPE']

models = [
    ('LR', LogisticRegression(solver='liblinear', multi_class='ovr')),
    ('LDA', LinearDiscriminantAnalysis()),
    ('KNN', KNeighborsClassifier()),
    ('CART', DecisionTreeClassifier()),
    ('NB', GaussianNB()),
    ('SVM', SVC(gamma='auto')),
    ('RandomForest', RandomForestClassifier())
]

results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    start_time = time.time()
    cv_results = cross_val_score(model, features, target, cv=kfold,
scoring='accuracy', error_score='raise')
    results.append(cv_results)
    training_time = time.time() - start_time
    names.append(name)
    print(f'{name}: {cv_results.mean():.3f} ({cv_results.std():.3f}) -
Training Time: {training_time:.3f} sec')

fig, ax = plt.subplots()
colors = ['b', 'g', 'r', 'c', 'm', 'y', 'r']

```

```

ax.bar(names, [result.mean() for result in results], color=colors)

ax.set_xlabel('Алгоритм')
ax.set_ylabel('Точність')
ax.set_title('Порівняння алгоритмів ML для TERMType')

plt.show()

label_encoder = LabelEncoder()
dataset['TERMType'] = label_encoder.fit_transform(dataset['TERMType'])

dataset['CURRENCY'] = label_encoder.fit_transform(dataset['CURRENCY'])
features_termType = dataset.drop(['TERMType'], axis=1)
target_termType = dataset['TERMType']
features_currency = dataset.drop(['CURRENCY'], axis=1)
target_currency = dataset['CURRENCY']

model_termType = RandomForestRegressor()
model_termType.fit(features_termType, target_termType)

next_data_termType = features_termType.tail(1)
predicted_termType = model_termType.predict(next_data_termType)
probability_termType = model_termType.predict_proba(next_data_termType)

print("Прогнозований термін обігу:", predicted_termType)
print("Ймовірність:", probability_termType)

model_currency = GaussianNB()
model_currency.fit(features_currency, target_currency)
next_data_currency = features_currency.tail(1)
predicted_currency = model_currency.predict(next_data_currency)
probability_currency = model_currency.predict_proba(next_data_currency)

print("Номер прогнозованої валюти:", predicted_currency)
print("Ймовірність:", probability_currency)

probabilities_df = pd.DataFrame(probability_termType,
                                columns=label_encoder.classes_)
print(probabilities_df)

features_qty = dataset.drop(['QTY'], axis=1)
target_qty = dataset['QTY']

model_qty = RandomForestRegressor()
model_qty.fit(features_qty, target_qty)

next_data_qty = features_qty.tail(1)
predicted_qty = model_qty.predict(next_data_qty)

print("Кількість паперів у наступній покупці:", int(predicted_qty),
      "шт.")

predicted_termType_all = model_termType.predict(features_termType)
predicted_currency_all = model_currency.predict(features_currency)

print("Точність прогнозу за терміном дії: ",

```

```
accuracy_score(target_termType, predicted_termType_all))
print("Точність прогнозу валюти: ", accuracy_score(target_currency,
predicted_currency_all))

print("Оцінка прогнозу за терміном дії:")
print(classification_report(target_termType, predicted_termType_all))

print("Confusion Matrix для прогнозу терміну дії:")
print(confusion_matrix(target_termType, predicted_termType_all))

print("Оцінка прогнозу за валютою:")
print(classification_report(target_currency, predicted_currency_all))

print("Confusion Matrix для прогнозу валюти:")
print(confusion_matrix(target_currency, predicted_currency_all))
```