

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 303.732.4

До захисту допущено
Завідувач кафедри ММСА
_____ Оксана ТИМОЩУК
« ____ » _____ 2024 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Системний аналіз фінансового ринку»
зі спеціальності 124 «Системний аналіз»
на тему: «Аналіз ринку криптовалют з використанням методів машинного
навчання та теорії ігор»

Виконала:
студентка II курсу, групи КА-32мп
Івашова Анастасія Петрівна _____

Науковий керівник:
професор кафедри ММСА, д.т.н., доц.
Зайченко Олена Юріївна _____

Рецензент:
Професор кафедри СП, д.т.н., проф.
Мухін Вадим Євгенович _____

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

Київ – 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти – другий (магістерський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійною програмою «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ

Завідувач кафедри ММСА

_____ Оксана ТИМОЩУК

« ____ » _____ 2024 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Івашовій Анастасії Петрівні

1. Тема дисертації: «Аналіз ринку криптовалют з використанням методів машинного навчання та теорії ігор», науковий керівник дисертації Зайченко Олена Юрійвна, професор, д.т.н., затверджено наказом по університету від «07» листопада 2024 р. № 5001-с.
2. Строк подання студентом дисертації 06.12.2024
3. Об'єкт дослідження: аналіз ринку криптовалют.
4. Предмет дослідження: методи машинного навчання та теорії ігор.
5. Перелік завдань, які потрібно розробити:
 - 1) аналіз джерел інформації за темою роботи;
 - 2) постановка мети та завдань;
 - 3) визначення об'єкта та предмет дослідження;
 - 4) вибір та обґрунтування методів машинного навчання та теорії ігор;
 - 5) збір та підготовка даних;
 - 6) розробка програмного забезпечення, що аналізує поведінку ринку криптовалют;
 - 7) тестування процесу роботи програмного забезпечення;
 - 8) аналіз ринкових можливостей запуску стартап-проекту;
 - 9) розробка відповідних висновків;
 - 10) підготовка ілюстративного матеріалу;

- 11) оформлення пояснювальної записки.
6. Перелік графічного (ілюстративного) матеріалу:
- 1) схема GAN (рис.);
 - 2) приклади функціонування створеного програмного продукту (рис.);
 - 3) таблиці;
 - 4) презентація.
7. Орієнтовний перелік публікацій: Івашова А.П., Зайченко О.Ю. Аналіз ринку криптовалют з використанням методів машинного навчання та теорії ігор // Системні науки та інформатика: збірник доповідей III Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 25–29 листопада 2024 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024, 7 стор.
8. Дата видачі завдання: 02 вересня 2024 року.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1.	Дослідження джерел інформації за темою роботи	02.09.2024 – 15.09.2024	Виконано
2.	Підготовка першого розділу	16.09.2024 – 29.09.2024	Виконано
3.	Підготовка другого розділу	30.09.2024 – 13.10.2024	Виконано
4.	Написання коду програмного забезпечення	14.10.2024 – 27.10.2024	Виконано
5.	Підготовка третього розділу	28.10.2024 – 10.11.2024	Виконано
6.	Підготовка четвертого розділу стартап-проєкту	11.11.2024 – 17.11.2024	Виконано
9.	Висновки та створення презентації	18.11.2024 – 24.11.2024	Виконано
10.	Оформлення пояснювальної записки	25.11.2024 – 01.12.2024	Виконано

Студентка _____ Анастасія ІВАШОВА

Науковий керівник дисертації _____ Олена ЗАЙЧЕНКО

РЕФЕРАТ

Магістерська дисертація: 100 с., 25 рис., 25 табл., 1 додаток, 28 джерел.

АНАЛІЗ КРИПТОВАЛЮТНИХ РИНКІВ, ПРОГНОЗУВАННЯ, ОПТИМІЗАЦІЯ, МЕТОДИ МАШИННОГО НАВЧАННЯ, ТЕОРІЯ ІГОР, ЕФЕКТИВНІСТЬ.

Об'єкт досліджень – аналіз ринку криптовалют.

Предмет досліджень – методи машинного навчання та теорії ігор.

Мета роботи – розробка алгоритму оптимізації стратегії на основі прогнозованих цін купівлі/продажу криптовалюти.

У даній роботі присутній огляд предметної області, застосованих методів машинного навчання та теорії ігор; опис створеного програмного забезпечення, тестування за допомогою оцінок якості та реалізація стартап-проекту.

Було проведено тестування моделі прогнозування ціни на ринку криптовалют та алгоритм оптимізації стратегії. За допомогою даної моделі та алгоритму вдалося покращити процес трейдингу та зробити його більш ефективним і прибутковим.

Програмне забезпечення реалізовано на мові Python, що дозволяє обробляти дані криптовалютного ринку, навчати моделі машинного навчання та аналізувати результати, створюючи обґрунтовану стратегію за допомогою теорії ігор.

ABSTRACT

Master's thesis: 100 p., 25 fig., 25 tabl., 1 appendix, 28 references.

ANALYSIS OF CRYPTOCURRENCY MARKETS, FORECASTING, OPTIMIZATION, MACHINE LEARNING METHODS, GAME THEORY, EFFICIENCY.

The object of research is the analysis of the cryptocurrency market.

The subject of research is machine learning and game theory methods.

The purpose of the work is to develop a strategy optimization algorithm based on predicted cryptocurrency purchase/sale prices.

This work contains an overview of the subject area, applied machine learning and game theory methods; a description of the created software, testing using quality assessments and implementation of a startup project.

The cryptocurrency market price forecasting model and strategy optimization algorithm were tested. Using this model and algorithm, it was possible to improve the trading process and make it more efficient and profitable.

The software is implemented in Python, which allows one to process cryptocurrency market data, train machine learning models, and analyze the results, creating a sound strategy using game theory.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Введення поняття криптовалюти.....	10
1.2 Порівняльний аналіз криптовалюти та фінансових інструментів	12
1.3 Розкриття процесу роботи трейдингу	15
1.4 Застосування методів машинного навчання та теорії ігор.....	20
1.5 Висновки до розділу 1.....	21
РОЗДІЛ 2 ТЕОРЕТИЧНИЙ ОПИС МЕТОДІВ МАШИННОГО НАЧАННЯ ТА ТЕОРІЇ ІГОР	22
2.1 Машинне навчання.....	22
2.1.1 Random Forest	22
2.1.2 AdaBoost.....	23
2.1.3 Extra Trees	26
2.1.4 Gradient Boosting	27
2.1.5 Ансамблевий метод.....	29
2.2 Теорія ігор	29
2.2.1 Рівновага Неша	29
2.2.2 Байєсівські ігри.....	32
2.3 Висновки до розділу 2.....	36
РОЗДІЛ 3 РОЗРОБКА ТА ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
3.1 Опис вхідних даних та їх обробка	37
3.2 Опис функцій програми.....	41
3.3 Процес машинного навчання	47
3.4 Оптимізація результатів за допомогою теорії ігор	48

	7
3.5 Аналіз результатів роботи ПЗ	49
3.6 Висновки до розділу 3	58
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ ЗА ТЕМОЮ МД	60
4.1 Опис ідеї продукту	60
4.2 Технологічний аудит ідеї продукту	62
4.3 Аналіз ринкових можливостей запуску стартап-проекту	64
4.4 Розроблення ринкової стратегії проекту	72
4.5 Розроблення маркетингової програми стартап-проекту	76
4.6 Висновки до розділу 4	80
ВИСНОВКИ	81
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	83
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	87

ВСТУП

Сьогодні фінансові ринки, зокрема ринок криптовалют, що буде досліджуватися у цій роботі, зазвичай характеризуються високою волатильністю та непередбачуваністю. Ці чинники створюють як ризики для трейдерів, так і можуть надати можливість отримати значний прибуток. Криптовалютний ринок є динамічним та складним, котрий постійно змінюється під впливом фінансових змін. Його аналіз виступає як об'єкт дослідження. Ось чому традиційні методи фінансового аналізу, на жаль, є недостатньо ефективними для повного розуміння складних взаємодій та непередбачуваних змін.

У зв'язку з високою динамікою криптовалютного ринку виникає потреба у використанні новітніх технологій, таких як методи машинного навчання та теорія ігор, що можуть допомогти покращити точність прогнозування цін та оптимізації стратегій для трейдерів.

Предметом досліджень є методи машинного навчання та теорії ігор для аналізу ринку криптовалют.

Машинне навчання допомагає знайти приховані принципи у великих наборах даних, що можуть бути більш ефективними під час аналізу ринку криптовалют і не тільки, а теорія ігор є галуззю математики, котра аналізує стратегічну взаємодію між різними гравцями. Їх поєднання надає інструментарій для повноцінного аналізу ринкових ситуацій, визначення оптимальних стратегій, та прийняття оптимальних рішень в умовах невизначеності та високих ризиків. Використання такого комплексного способу дозволяє передбачати майбутні тренди чи зміни в цінах купівлі/продажу, враховуючи стратегічні інтереси трейдерів.

Мета цієї магістерської дисертації є створення моделі для прогнозування цін криптовалют та розробка алгоритму оптимізації торгової стратегії із застосуванням методів машинного навчання та теорію ігор. Робота передбачає

не лише теоретичне дослідження цих методів, але й практичне застосування моделей на реальних даних, що має значення для трейдерів, фінансових аналітиків та інших зацікавлених сторін.

Вивчення та об'єднання цих передових технологій сприятиме підвищенню ефективності торгових стратегій, зниженню ризиків і збільшенню прибутковості операцій на ринку. Таким чином, ця магістерська дисертація має не лише академічне значення, але й важливе практичне застосування, що дозволяє інтегрувати у реальних ринкових умовах.

Наукова новизна роботи полягає у поєднанні методів машинного навчання та теорії ігор для аналізу ринку криптовалют. Це не лише покращить якість прогнозованих значень цін, а й розробить оптимальні стратегії трейдингу.

Відповідно ця робота спрямована на вирішення актуальних проблем аналізу складних моделей ринку та має академічно важливе значення для наукової спільноти, так і практичне для спеціалістів у сфері трейдингу.

Робота складається зі вступу, чотирьох розділів, висновків, переліку джерел посилання і одного додатку.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Введення поняття криптовалюти

Криптовалюта – це форма цифрового фінансового активу, що ґрунтується на блокчейн технології [1].

Блокчейн – це децентралізована база даних, яка функціонує за допомогою великою кількістю комп'ютерів та робить запис про всіх можливі транзакції [2].

Вона не контролюється центральним органом, таким як уряд або центральний банк, і забезпечує анонімність та безпеку транзакцій завдяки використанню криптографічних технологій. Цей фінансовий актив вирізняється здатністю проводити прямі транзакції між користувачами без залучення посередників, або ще відомі як транзакції від особи до особи. На відмінну від традиційних фінансових систем або фізичних гаманців, доступ до криптовалютних активів здійснюється через електронні гаманці або криптовалютні біржі [1].

Важливо зазначити, що криптовалюта, на відміну від інших фінансових активів, зберігається у електронних гаманцях чи на біржах. Доступ до них забезпечують приватні ключі, котрими володіють криптобіржі. Вони дозволяють користувачам отримувати доступ до їхніх коштів у блокчейні [1]. Однак слід розуміти, що даний актив супроводжується певними ризиками [1]:

- 1) висока волатильність;
- 2) загроза не мати доступ до всіх активів під час втрати приватних ключів;
- 3) економічні обмеження (закони).

Крім того, варто зазначити, що цей фінансовий інструмент використовується як засіб обміну, аналогічно до оплати за послуги чи товари. Однак, як вже зазначалося вище, криптовалюта не приймається в якості

офіційного платіжного засобу у більшості країн. На відміну від деяких інших країн, український президент Володимир Зеленський підписав законопроект «Про віртуальні активи», який був схвалений Верховною Радою 17 лютого 2022 року. Цей законодавчий акт надає криптовалюті легальний статус в Україні, встановлюючи її правове положення, класифікацію та права власності. Він також визначає основи для регулювання ринку цифрових активів та впроваджує механізми фінансового контролю в цій сфері [3].

Водночас у Європейському Союзі бажають створити єдину нормативну систему, що включає усі деталі криптовалютної діяльності. Наприклад, у травні 2023 року було затверджено нові правила регулювання криптовалюти, що знаний як Market in Crypto-Assets Regulation (MiCA). Відповідно за ними інвестори та бізнеси мають правову визначеність, а також вони підтримують фінансову стабільність [4].

Далі необхідно зазначити етапи роботи криптовалюти [1]:

1) відправник створює переказ, застосовуючи приватний ключ для створення цифровий підпис, що надає йому безпеку та можливість бути невпізнаним у мережі;

2) транзакція переходить у мережу до нодів (комп'ютерів), які перевіряють її достовірність, перевіряючи підпис чи достатньо грошей у відправника;

3) після підтвердження усіх перевірок, у блокчейні створюється новий блок, де зберігається дана транзакція.

Принцип роботи криптовалюти під час транзакції представлено на рисунку 1.1.

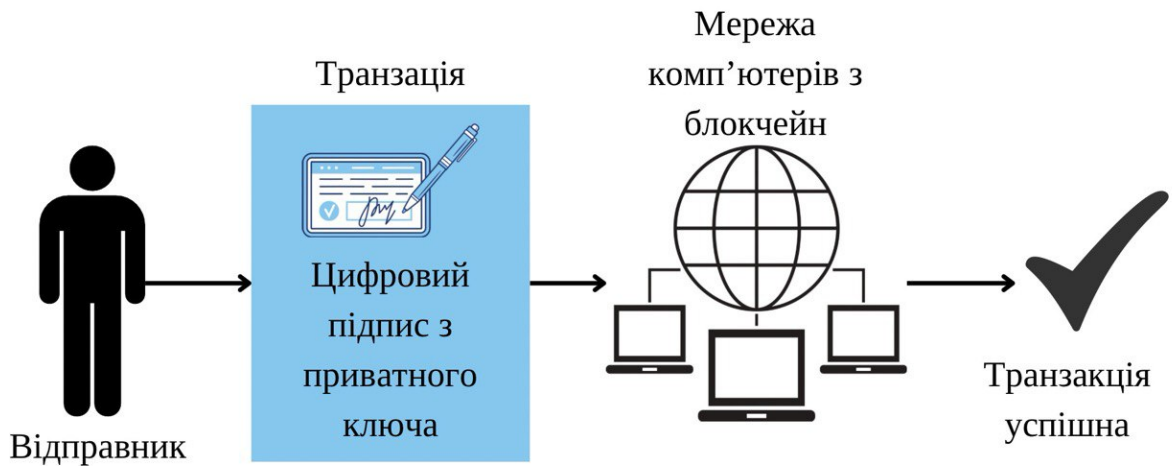


Рисунок 1.1 – Принцип роботи криптовалюти під час транзакції.

Серед найпопулярніших криптовалют виділяються наступні [1]:

- 1) Bitcoin (BTC);
- 2) Ethereum (ETH);
- 3) Tether (USDT);
- 4) BNB.

Отже, криптовалюта забезпечує високий рівень безпеки через відкритість та незмінюваність та не залежить від посередників, оскільки має технологію блокчейн. Тому наступним важливим кроком є порівняння з іншими фінансовими інструментами, щоб виявити переваги та недоліки цього активу.

1.2 Порівняльний аналіз криптовалюти та фінансових інструментів

У порівнянні з іншими, криптовалюта є децентралізованою, що дуже приваблює інвесторів. Оскільки це позбавляє посередників, що додатково беруть комісію за транзакцію, наприклад, як банки. Також, їх кількість обмежена, що може з часом збільшити їх ціну під час зростання попиту. Висока волатильність є досить привабливою, оскільки дозволяє заробляти у короткі

строки, однак це може спричинити і великі втрати, що робить її ризиковою в порівнянні з акціями. Оскільки є багато відмінностей, котрі можна відмітити, тому для зручності було створено таблицю 1.1, що охоплює основні характеристики для порівняння з іншими фінансовими інструментами.

Таблиця 1.1 – Порівняльний аналіз характеристик криптовалюти, акції та фіатної валюти [5, 6].

№	Характеристики	Фіатна валюта	Акція	Криптовалюта
1	Форма	Фізична	Цифрова або фізична (дуже рідко)	Цифрова
2	Представлення	Монети і банкноти	Права власності або участь в компанії	Код (публічний і приватний ключ)
3	Регулюється	Центральними банками або урядом	Фондовим ринком або державним урядом	Децентралізована, намагаються регулювати законами
4	Волатильність	Відносно стабільна, змінюється через економічні чинники	Зазвичай нижча, ніж у криптовалют	Висока, змінюється у короткий проміжок часу
5	Безпека	Залежить лише під час стабільної економіки	Залежить від фінансового стану компанії	Залежить від блокчейн та криптографії

Продовження таблиці 1.1.

№	Характеристики	Фіатна валюта	Акція	Криптовалюта
6	Кількість	Необмежена, бо регулюється урядом	Залежить від кількості випущених акцій	Обмежена, має свій максимум
7	Визначення вартість	Визначається ринковими регуляціями, економічними чинниками	Визначається фінансовою звітністю, економічними чинниками та попитом на ринку	Визначається попитом і пропозицією
8	Прозорість	Висока	Висока	Залежить від конкретної криптовалюти
9	Інвестиційний підхід	Не використовується	Довгострокова інвестиція	Короткострокова інвестиція

Отже, криптовалюта є перспективним та швидким способом для інвесторів. Однак це і пов'язує з достатньо великими ризиками та невизначеністю, що потрібно враховувати під час трейдингу. Тому для вирішення цієї проблеми необхідно дослідити нові варіанти уникнення ризику під час торгівлі криптовалютами.

1.3 Розкриття процесу роботи трейдингу

Торгівля виступає ключовим компонентом економічної діяльності, що охоплює купівлю та продаж активів. Зазвичай на фінансових ринках торгують різними активами такими як криптовалюта чи акції, облігації чи валютні пари, опціони чи ф'ючерси, маржинальні продукти чи інші фінансові інструменти. Для більшості словосполучення "торгівля", особливо коли мова йде про криптовалюту, асоціюється з короткостроковою діяльністю, де трейдери швидко відкривають і закривають позиції. Однак насправді торгівля може охоплювати різні стратегії та аспекти [7].

Торгова стратегія – це сукупність правил та підходів, котрими керуються трейдер під час купівлі/продажу на ринку [7].

На жаль немає універсальної стратегії, яка підходить усім можливим ситуаціям під час торгівлі. Ось чому способи ґрунтуються на особистому досвіді та стилі трейдингу інвестора. Через це необхідно мати добре обдуманний та структурований план торгівлі, щоб визначити чітку мету та уникнути емоційному впливу на рішення. У плані варто відзначати, з якими інструментами працює трейдер, якої позиції притримується, які методи чи стратегії використовує для торгівлі, а також точки входу і виходу з ринку.

Відповідно тому слід визначити які позиції бувають під час торгівлі. Їх існує лише 2 типи: довга та коротка.

Довга позиція – це коли трейдер купує актив, котрий за його власною оцінкою може бути перспективним, оскільки ціна зростає з часом. Тобто цей актив тримають до того моменту поки його вартість не підвищиться до свого пікового значення, а потім продають його за вищою ціною і отримують прибуток [7].

Наприклад, поширеним прикладом довгої позиції є стратегія "купуй і тримай", при якій інвестори купують активи і утримують їх довгий час,

очікуючи на їх зростання вартості. Це вважається базовою стратегією для багатьох, однак особливо для тих, хто бажає менше ризикувати.

Хоча існують спеціальні фінансові інструменти, які мають інший принцип роботи — вони зростають у ціні, коли ринок падає. Такі інструменти називаються токени з кредитним плечем. Наприклад токен BTCDOWN збільшує свою вартість, коли ціна Bitcoin падає, а у випадку зростання Bitcoin – зменшується. Тож, якщо трейдер відкриває довгу позицію на BTCDOWN, то він фактично очікує, що ціна на Bitcoin впаде, адже саме в такому випадку BTCDOWN зросте в ціні [7].

Коротка позиція — це коли трейдер продає актив, якого не має, з надією купити його пізніше за нижчою ціною. Це робиться для того, щоб заробити на падінні його ціни. Відповідно цей процес можна описати по пунктах [7].

1. Спочатку трейдер позичає актив (акції або криптовалюту) у брокера чи іншого суб'єкта.
2. Після чого трейдер продає позичений актив за поточною ціною.
3. Відповідно якщо ціна активу знижується, трейдер купує його назад за нижчою ціною.
4. Трейдер повертає позичений актив і заробляє на різниці між ціною продажу та купівлі.

Наприклад, якщо трейдер позичив і продав 1 Bitcoin за 10,000 доларів, а потім купує його назад, коли ціна впала до 8,000 доларів, його прибуток становитиме 2,000 доларів.

Однак якщо вже говорити не про позиції, слід також звернути увагу на усталені стратегії трейдингу [7].

1. Денна торгівля – це відкриття та закриття торговельних позицій протягом одного дня. Цей термін з'явився від традиційних ринків, які були доступні для торгівлі лише на протязі обмеженого часу в денний період. Поза цим часовим проміжком, денні трейдери зазвичай не залишають відкриті торговельні позиції.

2. Свінг-трейдинг користується рухами ринкових трендів, проте з більш тривалим часовим інтервалом – позиції зазвичай утримуються від кількох днів до кількох місяців. Метою зазвичай є визначення активів, які здаються недооціненими та мають потенціал до зростання вартості. В такому випадку купують цей актив і продають його, коли його ціна підніметься, щоб отримати прибуток. Також, можна шукати переоцінені активи, які ймовірно знизяться в ціні, тоді їх можна продати за високою ціною з наміром купити назад за меншу вартість.

3. Позиційна торгівля або трендова – це стратегія довгострокового інвестування. Трейдери придбають активи з метою збереження їх на тривалий час. Їхня ціль полягає в тому, щоб отримати вигоду, продавши ці активи по вищій ціні в майбутньому.

У трейдингу присутні два типи аналізу активів: фундаментальний та технічний.

Фундаментальний аналіз – це тип аналізу, що спрямований оцінити актив за допомогою дослідження економічних факторів і фінансових звітів для визначення дійсної та справедливої вартості. Зазвичай спеціалісти зосереджуються на макроекономічних умовах, галузевій тенденції та специфіці підприємства в залежності від типу активу. Також, вони застосовують макроекономічні індикатори для того, щоб можна було відслідкувати розвиток ситуації. Їх головною метою є пошук активу, котрий є недооціненим чи переоціненим, що дозволить трейдеру прийняти рішення чи варто купляти/продавати даний актив [7].

Зазвичай даний аналіз застосовується під час торгівлі акціями, оскільки він показує якісну оцінку таких активів. Застосування фундаментального аналізу для криптовалют все ще обмежене через його новизну, оскільки ринок в значній мірі рухається під впливом різних чинників. Тому для використання у ринку криптовалют необхідно провести більше експериментів.

Технічний аналіз – це інший спосіб оцінювання, який базується на ідеї, що історичні цінові рухи можуть давати змогу передбачити майбутню

тенденцію ринку. Такі аналітики не намагаються оцінити внутрішню вартість активу, а замість цього аналізують історичні дані, використовуючи графіки, технічні індикатори та шаблони, щоб виявити потенційні можливості. Технічний аналіз також є важливим інструментом управління ризиками, допомагаючи визначити та контролювати потенційні ризики [7].

Ця практика широко використовується у торгівлі криптовалютами, оскільки цей ринок схильний до спекулятивних впливів, що робить технічний аналіз особливо корисним для прогнозування цін купівлі/продажу на основі технічних чинників. Рідше використовується під час роботи з акціями, оскільки не може включити всі можливі аспекти роботи даного активу та іноді рух акцій може бути не передбачуваним через новини.

Процес трейдингу не є ідеальним, тому тут є свої переваги та недоліки, що представлені у таблиці 1.2.

Таблиця 1.2 – Переваги та недоліки процесу трейдингу [7].

№	Переваги	Недоліки
1	Приносить високі прибутки, особливо при врахування ризиків та правильному підході.	Має завжди ризик втрати грошей чи фінансових активів, особливо новачкам трейдерам.
2	Більшість фінансових активів, такі як акції та валюти, мають високу швидкість оборотів. Через це трейдери можуть швидко купувати чи продавати.	Невизначеність ринку та ризик втрати капіталу може спровокувати стрес та емоційного коливання.
3	Достатньо гнучкий для налаштування графіку та стратегії під себе.	Необхідно мати широкий спектр пізнання ринкових аспектів, аналітичних інструментів та стратегічного мислення.

Продовження таблиці 1.2

4	Є достатньо багато сучасних технологій, котрі дозволяють аналізувати та роботи швидкі операції, навіть з телефону.	Прибутковість залежить від економічної ситуації та ринкових умов.
---	--	---

Торгівля на фінансових ринках є складним процесом, що вимагає глибокого розуміння ринкових механізмів та умілого управління ризиками. Вона включає в себе використання різних стратегій, таких як денна торгівля, свінг-трейдинг, позиційна торгівля, кожна з яких має свої особливості і вимагає специфічних навичок. Фундаментальний та технічний аналіз є ключовими інструментами, що допомагають трейдерам приймати обґрунтовані рішення щодо купівлі чи продажу активів, і вони відіграють важливу роль у формуванні торговельних стратегій.

Незважаючи на потенційно високі прибутки, трейдинг завжди супроводжується ризиком втрати капіталу, особливо в умовах високої невизначеності ринку. Тому, трейдерам необхідно ретельно слідкувати за власними діями. Також, важливо розуміти, що емоційний контроль та дисципліна є критичними чинниками успіху в торгівлі.

Сучасні технології та торгові платформи надають трейдерам доступ до певного спектру інструментів для аналізу та виконання торгів, що може допомогти їм в реалізації їх торгових стратегій більш ефективно та в режимі реального часу. Незалежно від того, яку стратегію обирає трейдер, важливо вчитися і адаптуватися до змін у ринку, щоб підтримувати стійкість і прибутковість свого капіталу.

1.4 Застосування методів машинного навчання та теорії ігор

Машинне навчання та теорія ігор є широко застосовуваними підходами для роботи з різними задачами у фінансовій сфері.

Наприклад, у статті «Штучний інтелект і машинне навчання у фінансових послугах» Вакеріна А. та Коваль О. описують можливість застосування машинного навчання для покращення продуктивності фінансових процедур, враховуючи аналіз торгівлі та інвестицій. Автори визначають, що методи машинного навчання можуть виявляти закономірності та тренди ринку у великих базах даних. Крім того такі алгоритми використовуються для прогнозування цін активів, що дозволяє інвесторам приймати обґрунтовані рішення під час торгівлі, а також оптимізувати стратегії та виключити всі можливі ризики по типу високої волатильності [8].

У статті «Використання штучного інтелекту для прогнозування фінансових показників» Фаріон В., Гомотюк А., Назар Р. та Турчин С. також досліджують застосування методів машинного навчання для прогнозування даних та пошуку закономірностей. Слід зазначити, що у роботі вказано про важливість застосування адаптивних методів, що можуть включати нестабільність та високу волатильність даного ринку [9].

Якщо вже говорити за «Game Theory and Stock Investment» Ullah M., Lakhan G. R., Channa A., і Gul S. досліджують використання теорії ігор для оптимізації стратегії торгівлі. Хоча робота більше описує про фондовий ринок, однак це можна так само застосувати для ринку криптовалют. Також, важливо зазначити, що робота аналізує рівновагу Неша та стратегічну взаємодію, котрі допомагають інвесторам приймати коректні рішення в умовах невизначеності. Автори зазначають, що такі методи можуть моделювати конкурентні та кооперативні взаємодії між інвесторами, включаючи особливості їх запиту, що можна використати і для аналізу ринку криптовалют [10].

Відповідно у статті «Deciphering the Game of Strategy in Bitcoin's Network Through Game Theory» D-Central продемонстровано застосування теорії ігор для оптимізації роботи мережі Bitcoin. Наприклад, рівновага Неша допомагає покращувати взаємодію між майнерами, застосовуючи винагороди для додержування правил, що зменшує кількість атак та захищає з точністю до 51% [11].

Всі вище зазначені статті показують, що методи машинного навчання та теорії ігор є ефективними під час роботи з фінансовими активами. Тому у даній роботі буде поєднано ці два напрямки для прогнозування цін криптовалют та оптимізації стратегії.

1.5 Висновки до розділу 1

У даному розділі було розглянуто криптовалюту як фінансовий актив, котрий є децентралізованим, прозорим, безпечним та працює на основі блокчейн-технології. Проте даний актив має певні законодавчі обмеження, високу волатильність та значні ризики, є перепорою при торгівлі, що провокує створення нових та більш ефективних рішень. Також зроблено порівняльний аналіз криптовалюти з іншими видами активів. Звичайно було згадано всі аспекти торгівлі: визначення, типи, стратегії та методи аналізу. Після чого було розглянуто застосування машинного навчання та теорії ігор, що вказує на актуальність застосування та новизну об'єднання їх разом.

Цей огляд літератури надає значну базу для подальшого дослідження та розуміння практичного застосування.

РОЗДІЛ 2 ТЕОРЕТИЧНИЙ ОПИС МЕТОДІВ МАШИННОГО НАЧАННЯ ТА ТЕОРІЇ ІГОР

2.1 Машинне навчання

2.1.1 Random Forest

Random Forest – це алгоритм машинного навчання, котрий об’єднує велику кількість дерев рішень, кожен з них працює на випадковій вибірці даних [12].

Процес роботи алгоритму описується наступними етапами [12]:

1. Створення угруповання дерев рішень. Вони відрізняються від один одного тим, що спеціалізується на певному типі даних та працюють самостійно один від одного, що дозволяє мінімізувати ризик впливу аспектів одного дерева.

2. Вибір випадкових функцій під час тренування. Випадковий вибір показників уникає схожості дерев та надає різноманітності набору, що робить метод більш стабільним.

3. Робота з різними фрагментами даних. Тобто під час тренування модель навчається не на одній множині даних, а на декількох підмножинах, що обираються за допомогою випадкового вибору даних зі заміною.

4. Голосування та прийняття єдиного рішення. Під час прогнозування дерева рішень починають голосувати. Для задачі класифікації приймають ту відповідь, котра має найбільше голосів, а у випадку регресійної задачі обирають середнє значення усіх рішень.

На рисунку 2.1 представлено етапи роботи алгоритму.

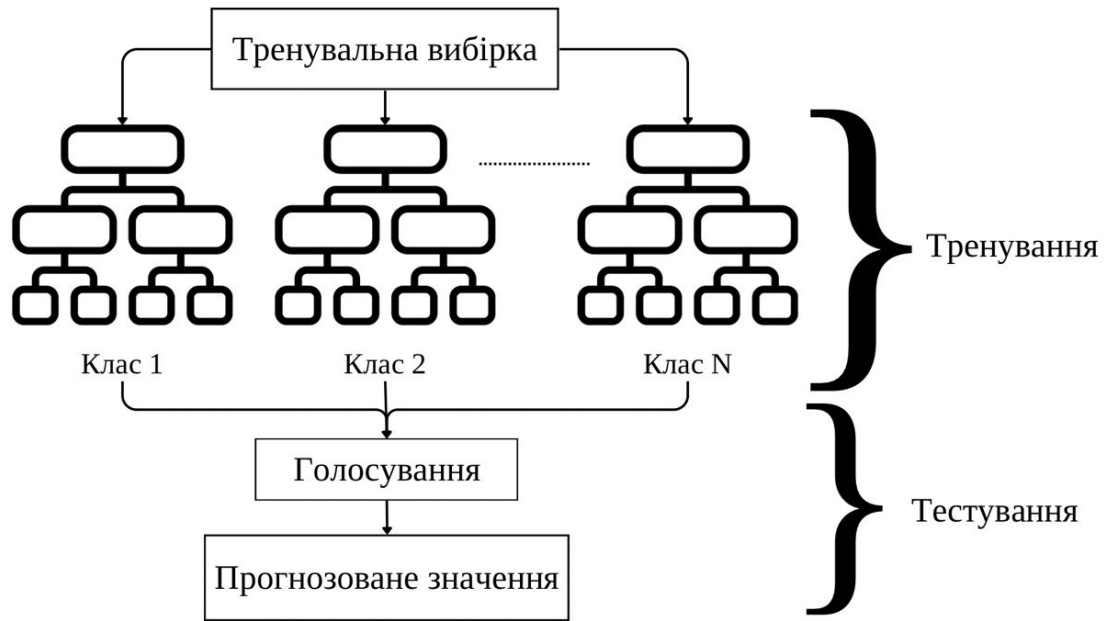


Рисунок 2.1 – Процес роботи алгоритму Random Forest [12].

Перевагами цього алгоритму є стійкість до перенавчання, здатність опрацювати велику вибірку даних та можливість оцінювати важливі показники. Однак звичайно у нього присутні недоліки: вимагає багато часу і ресурсів та важко відтворювати [12].

2.1.2 AdaBoost

AdaBoost – це ансамблевий алгоритм, що складає декілька слабких моделей у більш сильну модель [13].

Алгоритм можна висвітлити у наступні пункти [13]:

1. Вхідні дані: алгоритм слабого навчання, розмір ансамблю T та тренувальна вибірка S за формулою (2.1):

$$S = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_m, Y_m)\}, \quad (2.1)$$

де X_m – характеристики;

Y_m – мітка класу.

2. Початкові налаштування. Задаємо кожному прикладу вагу рівномірно за формулою (2.2):

$$\omega_i^1 = \frac{1}{m}, \text{ де } i = 1, 2, \dots, m. \quad (2.2)$$

3. Робимо для $t = 1, 2, \dots, T$. Алгоритм створює слабкий класифікатор на основі поточних розподіл ваг ω^t . Необхідно обчислити зважену похибку тренування за формулою (2.3):

$$\varepsilon_t = \sum_{i=1}^m \omega_i^t, Y_i \neq h_t(X_i), \quad (2.3)$$

де h_t – слабкий класифікатор на t -й ітерації.

Призначимо вагу для h_t за формулою (2.4):

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (2.4)$$

Оновлюємо ваги за формулою (2.5):

$$\omega_i^{t+1} = \frac{\omega_i^t \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(X_i))}{Z_t}, \quad (2.5)$$

де Z_t – нормалізація розподіл ваг за формулою (2.6):

$$Z_t = \sum_{i=1}^m \omega_i^t \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(X_i)) \quad (2.6)$$

Тобто у кінці кожної ітерації, що має правильну класифікацію отримує меншу вагу, а слабкий – більшу. Це дозволяє алгоритму звернути увагу на більш складні приклади.

4. Вихідні дані. Після закінчення усіх T ітерацій алгоритм поєднує усі слабкі класифікатори у один єдиний за формулою (2.7):

$$f(X) = \text{sign} \left(\sum_{t=1}^T \alpha_t \cdot h_t(X) \right) \quad (2.7)$$

На рисунку 2.2 представлено процес роботи алгоритму.

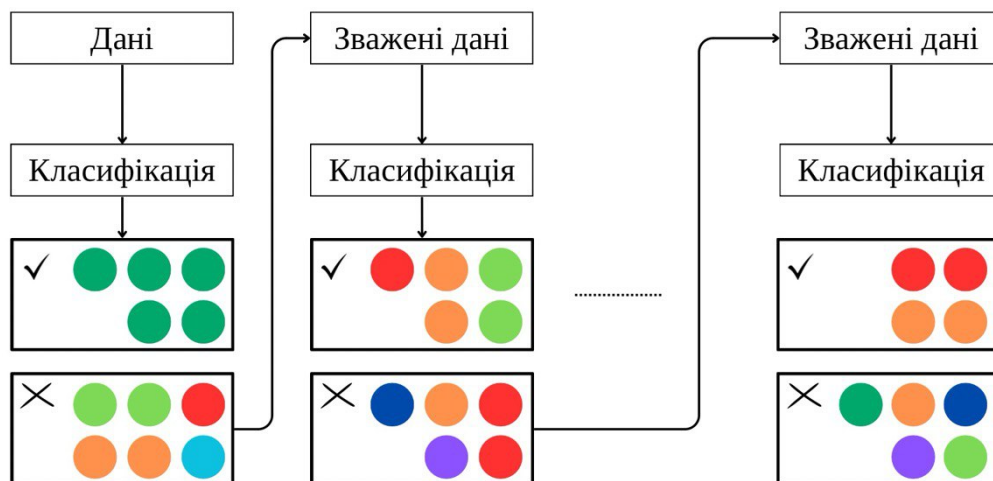


Рисунок 2.2 – Процес роботи алгоритму AdaBoost.

Перевагами цього методу є можливість роботи навіть на невеликих та складних наборах даних та забезпечення високої точності при правильному налаштуванні даних. Серед недоліків є те, що він чутливий до шумів та продуктивність сильно залежить від якості поданих слабких класифікаторів [13].

2.1.3 Extra Trees

Extra Trees є іншим варіантом дерев рішень методу Random Forest, однак з додатковим рівнем випадковості. У даному методі буде легше продемонструвати відмінності роботи з Random Forest, ніж описувати весь процес, оскільки вони мають схожу роботу, однак [14]:

1. Extra Trees обирає розбиття вузлів випадково, а Random Forest – за допомогою обчислення оптимальних критеріїв.
2. Extra Trees застосовує всю тренувальну вибірку, а Random Forest – підмножини даних.

Процес роботи представлено на рисунку 2.3.

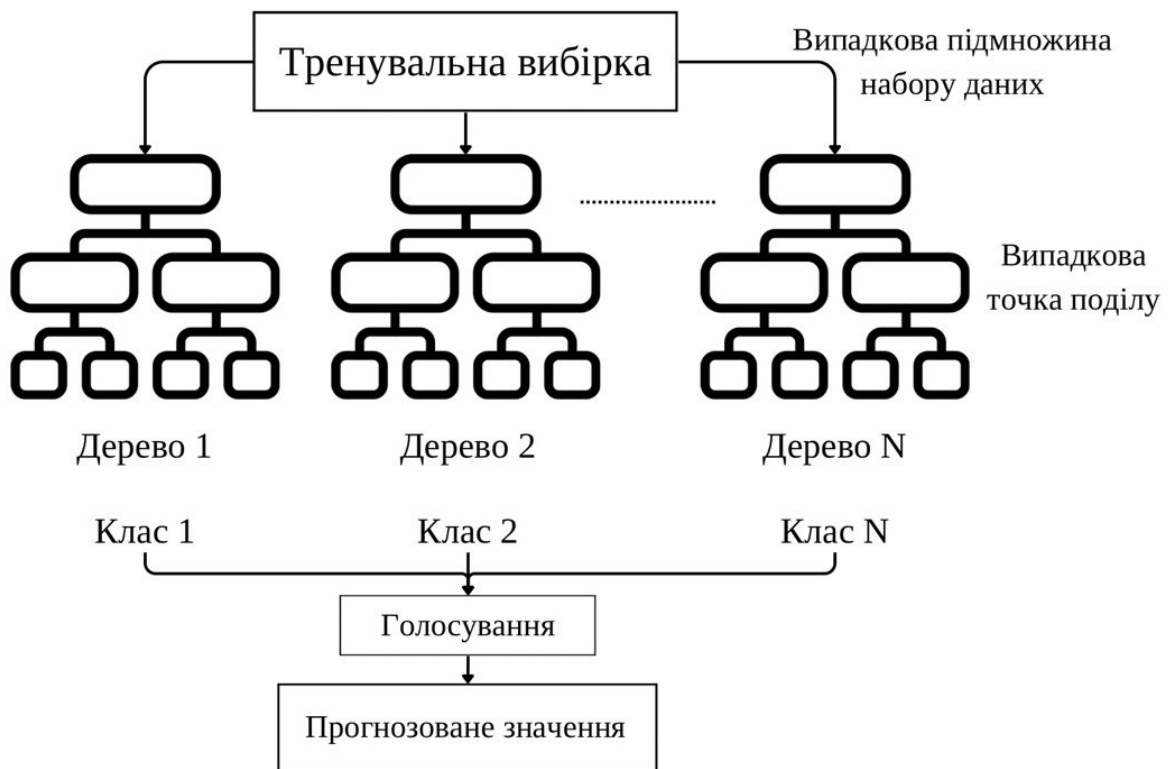


Рисунок 2.3 – Процес роботи алгоритму Extra Trees.

Має такі ж самі переваги як і в Random Forest, проте є швидшим. З недоліків можна виділити чутливість до шуму та потребу у налаштуванні параметрів для запобігання перенавчання [14].

2.1.4 Gradient Boosting

Gradient Boosting – це метод, котрий будує модель поетапно та додає нові, що намагаються виправити попередні помилки [15].

Алгоритм працює за наступними етапами [15]:

1. Вхідні дані:

- а. Вхідні данні $(x, y)_{i=1}^N$.
- б. Кількість ітерацій M .
- в. Функція втрат $\psi(y, f)$.
- г. Модель базова $h(x, \theta)$.

2. Робимо ітерації $t = 1, \dots, M$. Початкова модель – це звичайна константа f_0 .

3. Обрахунок негативного градієнта g_t . На кожній ітерації обраховує наскільки відхиляється модель від свого справжнього вигляду за формулою (2.8)

$$g_t = E_y \left[\frac{\partial \psi(y, f(x))}{\partial f(x)} \Big| x \right]_{f(x)=\hat{f}^{t-1}(x)} \quad (2.8)$$

4. Будуємо нову базову модель $h(x, \theta_t)$. При цьому вона ж і старається виправити ці помилки.

5. Шукаємо найкращий розмір кроку градієнтного спуску p_t за формулою (2.9). Під час цього процесу обирається скільки необхідно додати від нової моделі для покращення прогнозу.

$$p_t = \arg \min_p \sum_{i=1}^N \psi[y_i, f_{t-1}(x_i) + \rho h(x_i, \theta_t)] \quad (2.9)$$

6. Оновлення поточної моделі за формулою (2.10):

$$f_t \leftarrow f_{t-1} + \rho_t h(x, \theta_t) \quad (2.10)$$

7. Перехід до наступної ітерації. Алгоритм повторює минулі кроки до поки модель не буде точною.

На рисунку 2.4 зображено процес даного алгоритму.

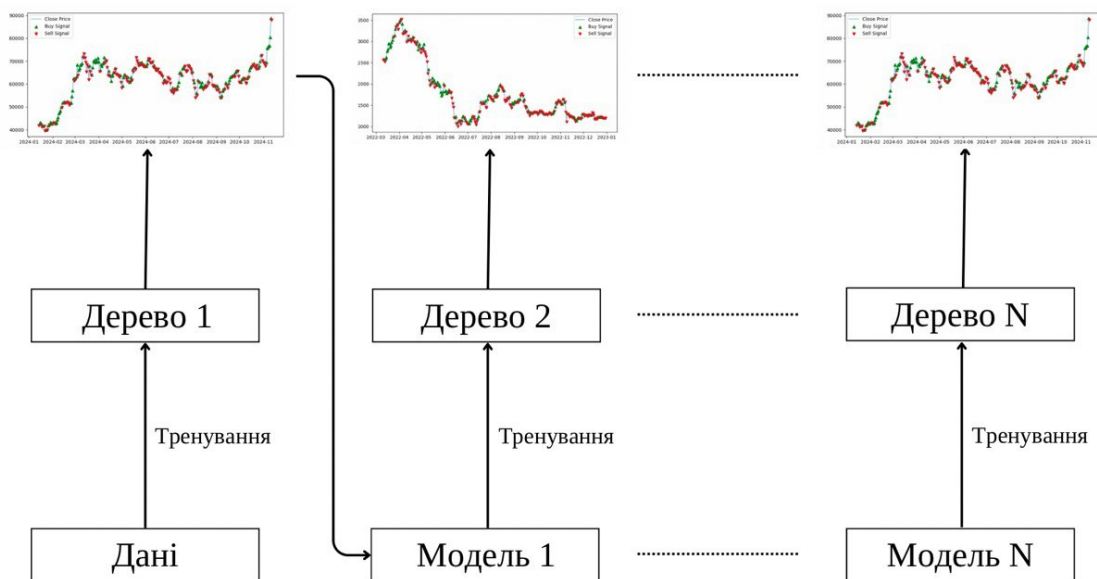


Рисунок 2.4 – Процес роботи алгоритму Gradient Boosting.

Переваги даного алгоритму: поступово покращує модель, зосереджується на більш складних прикладах та адаптується до будь-яких задач. Якщо вже говорити про недоліки, то слід вказати, що метод сприятливий до шуму, вимогливий у ресурсному плані та може перенавчатися під час недостатньої стабілізації [15].

2.1.5 Ансамблевий метод

Ансамблевий метод є цікавим та розповсюдженим способом створення моделі за допомогою об'єднання декількох інших алгоритмів.

У даній роботі буде розглядатися метод голосування – це найлегший серед усіх ансамблевих алгоритмів [16].

Ідея полягає в тому, що кілька моделей об'єднують свої прогнози через голосування у один єдиний кращий. Він ґрунтується на математичному припущенні про незалежність оцінювання, на жаль в реальних умовах такого досягти дуже важко, тому що алгоритми працюють на одних і тих самих даних [16].

2.2 Теорія ігор

2.2.1 Рівновага Неша

Одним з найважливіших концептів у теорії ігор є рівновага Неша. Вона названа на честь Джона Неша, який розробив цей концепт.

Рівновага Неша в теорії ігор є ситуацією, в якій кожен гравець вибрав свою оптимальну стратегію, враховуючи стратегії інших гравців, і жоден гравець не може покращити свій результат, змінивши лише свою стратегію. У такій ситуації жоден гравець не має стимулу відходити від своєї обраної стратегії, оскільки це не призведе до кращого результату для нього [17].

За формальним визначенням нехай є n гравців, кожен з яких має множину можливих стратегій S_i . Нехай $u_i(s_1, s_2, \dots, s_n)$ — це виграш i -го гравця при стратегіях s_1, s_2, \dots, s_n . Профіль стратегій $s^* = (s_1^*, s_2^*, \dots, s_n^*)$ є

рівновагою Неша, якщо для кожного гравця i виконується умова: $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i^*, s_{-i}^*)$ де s_{-i}^* — це профіль стратегій всіх інших гравців, крім i [17].

Розглянемо принцип роботи на прикладі дилеми в'язня.

Дилема в'язня — це класична проблема теорії ігор, яка ілюструє, чому два раціональні індивіди можуть не співпрацювати, навіть якщо це буде вигідніше для обох [18].

Припустимо, двоє злочинців, Аліса і Боб, заарештовані поліцією і підозрюються в скоєнні серйозного злочину. Поліція не має достатніх доказів для засудження їх обох за цей злочин, тому їх розміщують в окремі камери і пропонують угоду кожному [18]:

1) якщо один з них зізнається і свідчить проти іншого, а інший мовчить, той, хто зізнався, буде звільнений, а той, хто мовчав, отримає 10 років ув'язнення;

2) якщо обидва зізнаються, обидва отримають по 5 років ув'язнення;

3) якщо обидва мовчать, кожен отримає по 1 року ув'язнення за менш серйозне правопорушення.

Це можна представити у вигляді наступної матриці виплат (вигоди представлені у вигляді негативних значень, які означають роки ув'язнення) у таблиці 2.1 [18]:

Таблиця 2.1 – Матриця виплат дилеми в'язня.

	Боб мовчить	Боб зізнається
Аліса мовчить	-1, -1	-10, 0
Аліса зізнається	0, -10	-5, -5

Тепер застосуємо рівновагу Неша до цієї задачі [18]:

1) Аліса мовчить, Боб мовчить:

а) якщо Аліса знає, що Боб мовчить, то їй краще зізнатися (отримати 0 років замість 1);

б) якщо Боб знає, що Аліса мовчить, то йому теж краще зізнатися (отримати 0 років замість 1).

2) Аліса зізнатися, Боб мовчить:

а) якщо Аліса знає, що Боб мовчить, то їй краще зізнатися (отримати 0 років замість 10);

б) якщо Боб знає, що Аліса зізнатися, то йому краще також зізнатися (отримати 5 років замість 10).

3) Аліса мовчить, Боб зізнатися:

а) якщо Аліса знає, що Боб зізнатися, то їй краще теж зізнатися (отримати 5 років замість 10);

б) якщо Боб знає, що Аліса мовчить, то йому краще зізнатися (отримати 0 років замість 1).

4) Аліса зізнатися, Боб зізнатися:

а) якщо Аліса знає, що Боб зізнатися, то їй краще теж зізнатися (отримати 5 років замість 10);

б) якщо Боб знає, що Аліса зізнатися, то йому краще зізнатися (отримати 5 років замість 10).

Таким чином, рівновага Неша в даній дилемі — це коли обидва гравці зізнаються, отримуючи по 5 років ув'язнення кожен. Це стабільний стан, оскільки жоден з гравців не може поліпшити свій виграш, змінивши свою стратегію в односторонньому порядку [18].

Рівновага Неша в трейдингу означає, що трейдер діє в рамках, які значно підвищують його шанси на виживання в довгостроковій перспективі. Проте ці рамки є неочевидними та дуже контрінтуїтивними [19].

2.2.2 Байєсівські ігри

Байєсівські ігри є розширеним типом класичних ігор, які включають ймовірнісні елементи, що дозволяє враховувати невизначеність щодо інформації про інших гравців [18].

Байєсівська гра — це модель стратегічної взаємодії між гравцями, чий уподобання можуть залежати від чинників, що виходять за межі стратегій, які вони обирають для гри. Ці чинники часто вважаються характеристиками гравців, такими як їхня ефективність або лінь, їхня енергія чи як вони цінують певні об'єкти. Все дуже залежить від сенсу, наприклад на співбесіді, бійка чи аукціон та інше [18].

Тип гравця часто розглядається як множина характеристик, що описує його самого та на скільки багато йому відомо. Загалом, типи можуть бути розглянуті як сукупність чинників, що можуть в подальшому вплинути на результат. Наприклад, результат битви між двома арміями може залежати не лише від маневрів, які вони виконують, але й від того, наскільки великими чи добре підготовленими вони були спочатку, а також від виду місцевості, на якій вони діють. Ухвалення рішень у такому середовищі вимагає представлення невизначеності гравців щодо цих змінних [18].

Тепер надмо визначення байєсівських ігор, яке є дещо більш загальним, ніж стандартне визначення. Це дозволить легше розробляти розширення для байєсівських ігор з намірами [20].

Зафіксуємо набір гравців, $N = \{1, \dots, n\}$. Байєсова гра (над N) — це кортеж $V = (\Omega, (\Sigma_i, T_i, \tau_i, p_i, u_i)_{i \in N})$, де [20]:

- 1) Ω – вимірний простір природних станів;
- 2) Σ_i — набір стратегій, доступних гравцеві i ;
- 3) T_i — набір типів гравця i ;
- 4) $\tau_i: \Omega \rightarrow T_i$ – сигнальна функція гравця i ;

5) $p_i: T_i \rightarrow \Delta(\Omega)$ пов'язує з кожним типом t_i гравця і ймовірнісну міру $p_i(t_i)$ на Ω з $p_i(t_i) \left(\tau_i^{-1}(t_i) \right) = 1$, представляючи тип t_i гравця i в його віра у природній стан;

6) $u_i: \Sigma \times \Omega \rightarrow \mathbb{R}$ – функція користності гравця i .

Буде корисно коротко розглянути два простих приклади байєсівських ігор [20].

Приклад 1.

Спочатку розглянемо спрощений сценарій аукціону, де кожен учасник $i \in N$ повинен подати ставку $\sigma_i \in \Sigma_i = \mathbb{R}^+$ на даний предмет. Тут типи уявляються як закодовані оцінки вартості предмета для аукціону: для кожного $t_i \in T_i$, нехай $v(t_i) \in \mathbb{R}^+$ представляє, яку гравець i надумав вартість предмету, і визначається користність гравця i , $u_i: \Sigma \times T$. За формулою (2.11) обрахуємо [20]:

$$u_i(\sigma, t) = \begin{cases} v(t_i) - \sigma_i, & \text{якщо } \sigma_i = \max_{j \in N} \sigma_j \\ 0, & \text{в іншому випадку} \end{cases} \quad (2.11)$$

Таким чином, виграш гравця дорівнює 0, якщо він не подає найвищу ставку, котра дорівнює його оцінці вартості предмета мінус його ставка (для простоти, модель припускає, що у разі нічийї кожен гравець із найвищою ставкою отримує предмет). Зазначимо, що тут простір станів неявно вважається тотожним простору профілів типів T , тобто гра представлена в скороченому вигляді. Ось чому тип t_i повідомляє не лише про те, наскільки гравець i вважає цей предмет $v(t_i)$ цінним, але й які переконання $p_i(t_i) \in \Delta(T)$ гравець має про оцінку інших гравців. Умова, що $p_i(t_i) \left(\tau_i^{-1}(t_i) \right) = 1$, просто означає, що кожен гравець впевнений у власній оцінці та у своїх уявленнях про типи інших гравців [20].

Приклад 2.

Далі буде розглядатися, де байєсівська структура використовується для моделювання гравця, чиї переваги залежать від переконань її суперника. Розглянемо гру, де гравці є учнями в класі, при цьому гравець один щойно був викликаний вчителем відповісти на стандартне запитання, що має містити в собі відповідь так або ні. Для спрощення припустимо, що $N = \{1, 2\}$, $\Sigma_1 = \{\text{так, ні, пропустити}\}$, $\Sigma_2 = \{*\}$, де * означає пусту множину ходів, тому лише гравець один має прийняти реальне рішення. Нехай $\Omega = \{w_y, w_n, v_y, v_n\}$, де стани з нижнім індексом y означають, що «так» є правильною відповіддю, а стани з нижнім індексом n означають, що «ні» є правильною відповіддю [20].

Нехай $T_1 = \{t_1, t'_1\}$, $T_2 = \{t_2, t'_2, t''_2\}$ і визначимо сигнал функції за формулами (2.12) та (2.13) як

$$\tau_1(w_y) = \tau_1(w_n) = t_1, \tau_1(v_y) = \tau_1(v_n) = t'_1 \quad (2.12)$$

$$\tau_2(w_y) = \tau_2(w_n) = t_2, \tau_2(v_y) = t'_2, \tau_2(v_n) = t''_2. \quad (2.13)$$

Нарешті, припустимо, що всі суб'єктивні міри ймовірності є результатом через визначення спільного пріоритету $\pi \in \Delta(\Omega)$ від типу гравця. Далі припустимо, що π є рівномірним розподілом. Це свідчить про те, що в кожному стані гравець 1 не впевнений у правильній відповіді. З іншого боку, в станах w_y і w_n гравець 2 також не впевнений у правильній відповіді, а в станах v_y і v_n гравець 2 знає правильну відповідь. Крім того, в станах w_y і w_n гравець 1 впевнений, що гравець 2 не знає правильної відповіді, тоді як в станах v_y і v_n гравець 1 впевнений, що гравець 2 знає правильну відповідь, незважаючи на те, що сам не знає. Таким чином, можна використовувати цю структуру для кодування наступних переваг для гравця 1: якщо гравець 2 не знає відповіді, тоді краще вгадати, а ніж пропустити хід, однак в іншому випадку краще пропустити, ніж вгадати неправильно [20].

Встановимо: $u_1(\text{так}, w_y) = u_1(\text{так}, v_y) = u_1(\text{ні}, w_n) = u_1(\text{ні}, v_n) = 5$, що представляє хорошу винагороду за правильну відповідь [20].

Також встановимо: $u_1(\text{пропустити}, x) = -2$ для всіх $x \in \Omega$, що представляє собою штраф за пропускання ходу, незалежно від того чи правильна відповідь взагалі [20].

У висновку виходить: $u_1(\text{так}, w_n) = u_1(\text{ні}, w_y) = -5$, $u_1(\text{так}, v_n) = u_1(\text{ні}, v_y) = -15$, що представляє штраф за неправильну відповідь, де гравець 2 знає правильну відповідь [20].

Легко перевірити, що якщо гравець 1 вважає w_y і w_n однаково ймовірними, то його очікування корисності від вгадування відповіді дорівнює 0, що краще, ніж пропускання. В іншому випадку, якщо гравець 1 вважає v_y і v_n однаково ймовірними, тоді його очікувана корисність від випадкового вгадування становить -5, що значно гірше, ніж пропускання. Одним словом, рішення гравця 1 залежить від його уявлення про думки гравця 2 [20].

Байєсовські ігри відіграють важливу роль в ситуаціях, де гравці мають різні рівні інформації. У таких іграх кожен гравець знає свій унікальний тип, який впливає на його дії та стратегії, але немає повної картини щодо типів інших учасників. Це змушує гравців реагувати не тільки на очевидні дії опонентів, але й на їх потенційні інформаційні переваги [20].

Метод Байєса в теорії ігор відіграє вирішальну роль у моделюванні ситуацій з неповною інформацією. Цей спосіб дозволяє кожному гравцю використовувати свої уявлення про стан світу та інших гравців для формування своїх стратегій, які максимізують їх очікуваний виграш залежно від їх типів та переконань [20].

2.3 Висновки до розділу 2

У другому розділі було проаналізовано методи машинного навчання та теорії ігор, що будуть застосовуватися для створення програмного забезпечення. Розглянуті методи демонструють свої недоліки та переваги, але є потенціально важливими компонентами для створення програмного забезпечення.

Серед методів машинного навчання досліджені Random Forest, AdaBoost, Extra Trees, Gradient Boosting. Random Forest забезпечує стабільність моделі завдяки використанню ансамблю дерев рішень, що зменшує ризик перенавчання. AdaBoost спрямовує увагу моделі на складні для класифікації випадки, що дозволяє підвищити загальну точність прогнозів. Extra Trees виділяється швидкістю навчання та здатністю зменшувати вплив шуму у даних завдяки випадковим розбиттям вузлів. Gradient Boosting поступово вдосконалює прогнози шляхом корекції помилок попередніх ітерацій, що робить його ефективним для завдань підвищеної складності. Ensemble Method об'єднує результати кількох моделей, що дозволяє досягти збалансованості між точністю та стабільністю прогнозів, роблячи цей підхід одним із найефективніших для аналізу ринку криптовалют.

У рамках теорії ігор розглянуто рівновагу Неша, що дозволяє моделювати взаємодію між учасниками ринку. Як приклад застосування рівноваги Неша розглянуто дилему в'язня, яка ілюструє, як учасники можуть приймати рішення, враховуючи дії один одного. Також, досліджено байєсівські ігри, які базуються на ймовірностях і враховують невизначеність учасників ринку щодо ринкових умов.

Загалом, розглянуті методи є фундаментом для створення програмного забезпечення, що здатне вирішувати задачі прогнозування та оптимізації стратегій на ринку криптовалют.

РОЗДІЛ 3 РОЗРОБКА ТА ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Головна мета третього розділу магістерської дисертації – описати всі важливі моменти практичної реалізації завдання магістерської дисертації, використовуючи згадані у розділі 2 методи машинного навчання та теорії ігор.

Програмне забезпечення (ПЗ) було створено на мові Python, тому що ця мова є гнучкою при створенні складних систем чи моделей, має легкий синтаксис та містить багато необхідних бібліотек таких як sklearn, що дозволяє працювати з різними методами машинного навчання.

Відповідно практична робота відбувалася наступним чином:

- 1) вибір вхідних даних;
- 2) обробка вхідних даних;
- 3) створення навчальної моделі;
- 4) створення оптимізації стратегії;
- 5) тестування програми на різних вибірках;
- 6) аналіз результатів роботи ПЗ.

3.1 Опис вхідних даних та їх обробка

Для аналізу ринку криптовалют використовуються дані, що завантажуються з Yahoo Finance, або користувач може самостійно надати власну вибірку у форматі CSV. На рисунку 3.1 зображено теку «Історичні дані» на сайті Yahoo Finance.

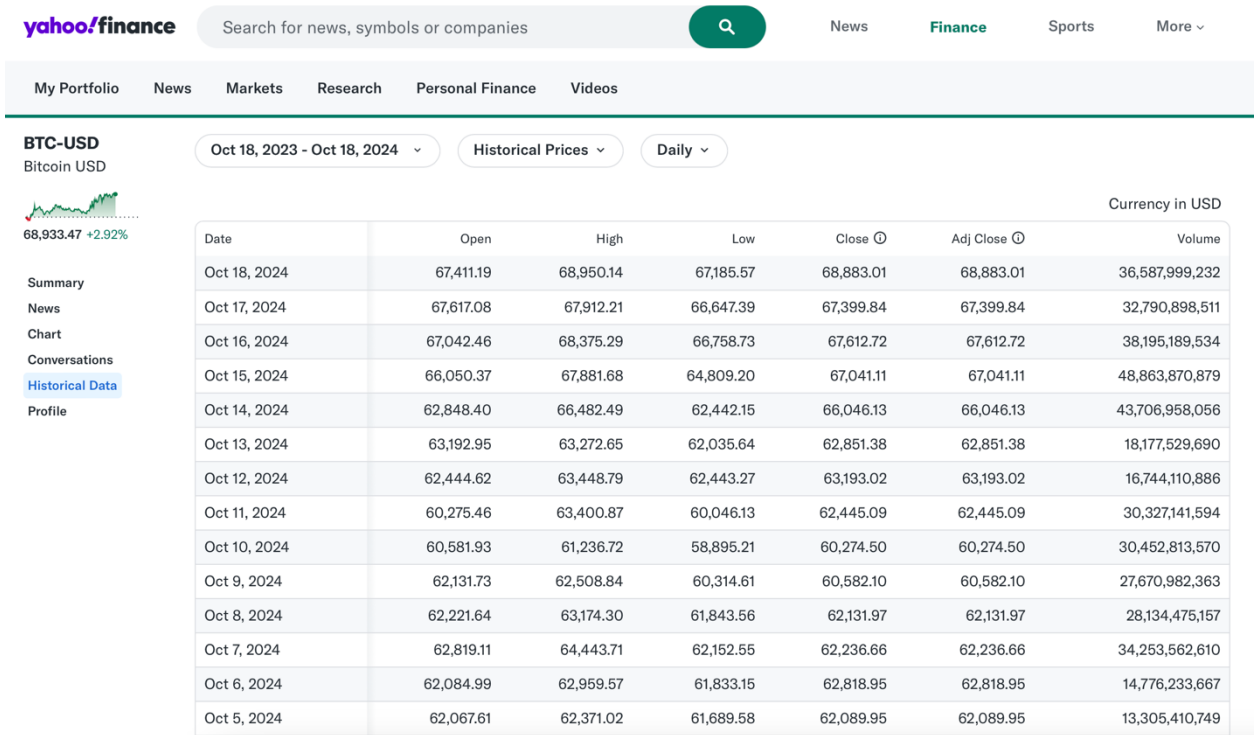


Рисунок 3.1 – Тека «Історичні дані» на сайті Yahoo Finance [21].

Дані містять інформацію про криптовалюту за певний період. Відповідно, дані містять наступні стовпчики:

1. Date – дата запису даних та застосовується як індекс.
2. Open – ціна відкриття на початку торговельного дня.
3. Close – ціна закриття на кінець торговельного дня.
4. High – максимальна ціна протягом торговельного дня.
5. Low – мінімальна ціна протягом торговельного дня.
6. Adj Close – ціна закриття, що враховує дивіденди та інше.
7. Volume – обсяг, котрий було продано за торговельний день.

Дані є базою для технічного аналізу та побудови моделі за допомогою методів машинного навчання, яка прогнозує сигнали купівлі/продажу.

Перед моделюванням необхідно обробити дані, тому можна виділити наступні етапи.

Етап 1. Видаляються всі пусті значення.

Етап 2. До вхідної вибірки додаються технічні індикатори, котрі будуть перераховані нижче:

а. Simple Moving Average (SMA) – це індикатор, котрий обраховується поділом суми крайніх цін фінансового активу на загальну кількість періодів. Є гнучким, оскільки може застосовуватися для будь-якого періоду часу. Однак не враховує важливість новітніх даних порівняно зі старими, бо на думку певних трейдерів новітня інформація може надавати інформацію про тренд, натомість інша частина трейдерів вважає, що можна виявити загальну тенденцію. Індикатор повністю залежить від історичних даних, що і є його ключовим обмеженням і він обчислюється за формулою (3.1) [22].

$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n}, \quad (3.1)$$

де A_n – ціна фінансового активу;

n – загальна кількість періодів [22].

б. Moving Standard Deviation (MSD) – це індикатор, котрий вимірює волатильність активу за рахунок стандартного відхилення за певний проміжок часу. Допомогає приймати обґрунтовані рішення щодо ринкових рухів. Однак може давати хибні сигнали під час низької волатильності, а також є запізнілим, бо базується на старих даних, тому не встигає за швидкими змінами ринку. Індикатор обчислюється за формулою (3.2) [23].

$$MSD = \sqrt{\frac{1}{n} \sum_{i=0}^n (P_i - \mu)^2} \quad (3.2)$$

де P_i – ціна фінансового активу в момент часу i ;

μ – середнє значення цін активу за певний період;

n – кількість періодів [23].

в. Relative Strength Index (RSI) – це технічний індикатор імпульсу, який застосовується для аналізу швидкості та коливання недавніх змін цін фінансового активу. Індикатор зображується у вигляді лінійного графіка на шкалі від 0 до 100. Дозволяє виявити, коли ціна активу є занадто високою (перекупленою) або низькою (перепроданою). Зазвичай значення індикатору вище 70 вказує на перекупленість, а нижче 30 – на перепроданість. Проте може залишатися в одному з цих станів протягом довго часу при значному імпульсі. Також вказує на можливий розворот тренду. Однак сигнали розвороту є рідкими та можуть бути сприйняті навіть хибні. Індикатор обчислюється за формулою (3.3) [24].

$$RSI_{\text{перший крок}} = 100 - \left[\frac{100}{1 + \frac{\text{Середній приріст}}{\text{Середня витрат}}} \right] \quad (3.3)$$

г. Ichimoku Cloud – це набір технічних індикаторів, що відображають рівні підтримки і опору, імпульс та напрямок тренду. Досягається це за рахунок кількох середніх та нанесення їх на графік. Незважаючи на складність набору з першого погляду, деякі трейдери вважають його легким у розумінні через виражені сигнали. Складається з п'яти ліній (двоє з них утворюють хмару): середині за 9 періодів, 26 періодів, середню з двох цих середніх, середню за 52 періоди та лінію відкладеного закриття. Обчислюються за формулами (3.4) - (3.8) [25].

$$\text{Лінія конверсія} = \frac{9 - PH + 9 - PL}{2} \quad (3.4)$$

$$\text{Базова лінія} = \frac{26 - PH + 26 - PL}{2} \quad (3.5)$$

$$\text{Провідний промінь } A = \frac{CL + \text{Базова лінія}}{2} \quad (3.6)$$

$$\text{Провідний промінь } B = \frac{52 - PH + 52 - PL}{2} \quad (3.7)$$

$$\text{Відкладений промінь} = \text{Закриття, нанесене 26 періодів назад} \quad (3.8)$$

де PH – максимум за період;

PL – мінімум за період;

CL – лінія конверсія [25].

Етап 3. Застосування стандартизації. Стандартизація (Standardization) – це метод, котрий забезпечує коректну роботу машинного навчання, оскільки нормалізує розподіл даних так, щоб вони мали середнє значення нуль, а стандартне відхилення один [26].

Етап 4. Застосування PCA. Principal Component Analysis (PCA) – це метод, котрий зменшує розмірність та скорочує кількість змінних, при цьому залишаючи найважливіше [26].

Ця попередня обробка даних дозволить покращити якість аналізу та підвищить точність прогнозування сигналів купівлі/продажу.

3.2 Опис функцій програми

Після обробки даних було створено наступні функції програмного забезпечення:

1. class FinancialAnalysisApp – є основним компонентом програмного забезпечення для фінансового аналізу.

2. `def __init__(self, root)` – ця функція є ініціалізатором класу `FinancialAnalysisApp`, що створює головні атрибути об'єкта програми. Тут ініціалізується змінні для збереження даних, налаштування графічного інтерфейсу та стартові параметри `Back Testing`.

3. `def create_widgets(self)` – ця функція створює головні компоненти інтерфейсу користувача, такі як текстові поля введення даних, дати початку та кінця, також кнопки завантаження даних, запуску аналізу, оптимізація стратегії та завантаження результатів у форматі `xls`. Відповідно є відведені місця для створення графіків та виведення результатів роботи коду. Можна сказати, що вона організовує компоненти у інтерфейсі користувача, котрі розглянуті на рисунках 3.2 - 3.5.

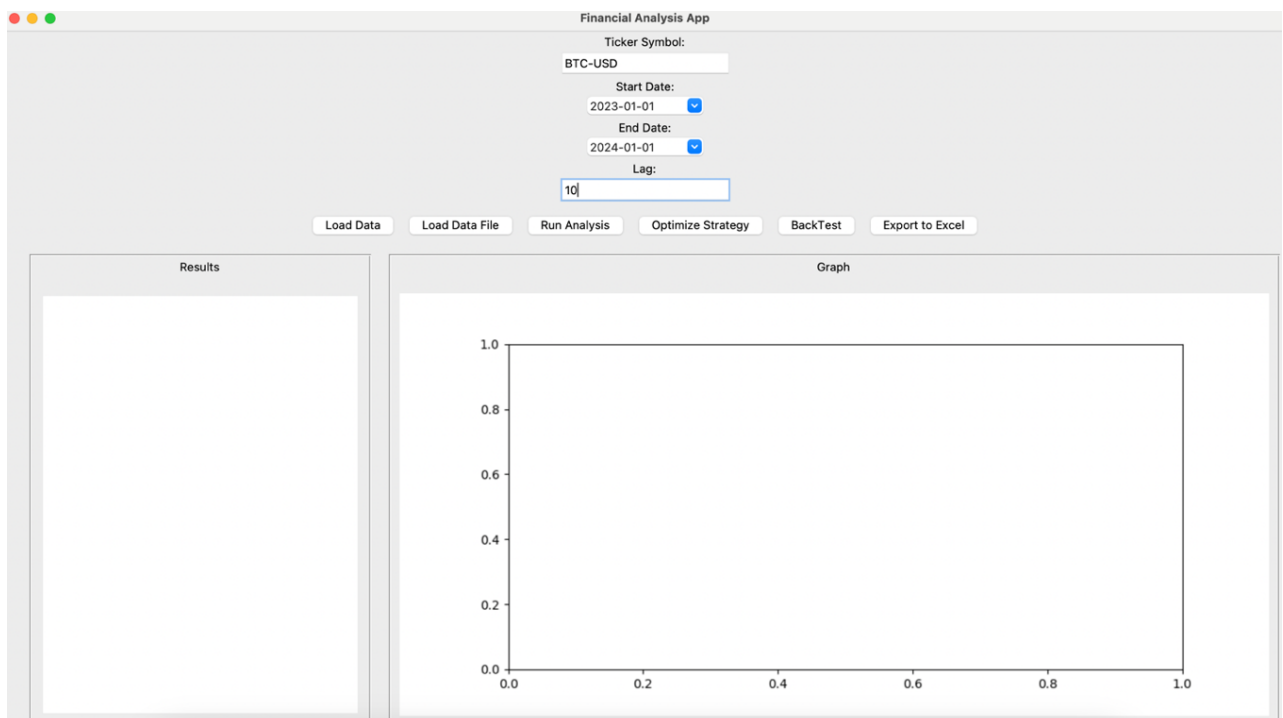


Рисунок 3.2 – Вікно додатку `FinancialAnalysisApp`.

Рисунок 3.3 – Текстові поля.

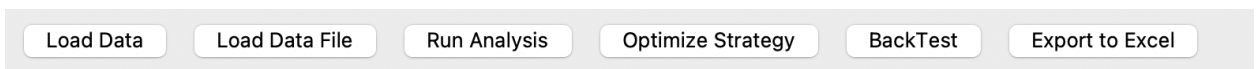


Рисунок 3.4 – Кнопки інтерфейсу.

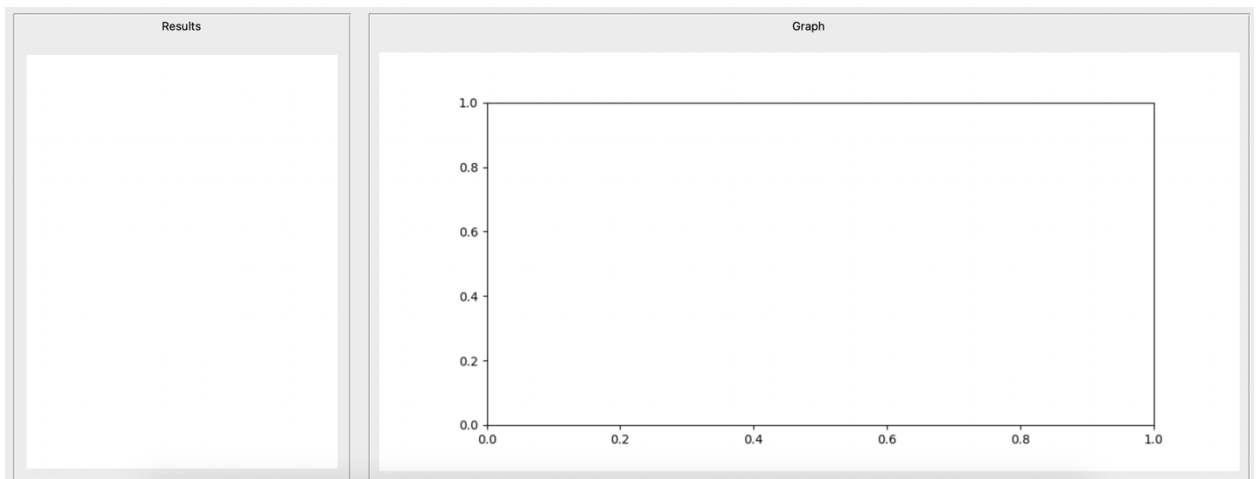


Рисунок 3.5 – Поля виведення результатів та графіків.

4. `def load_data(self)` – ця функція завантажує історичні дані про ціну фінансового активу з Yahoo Finance, застосовуючи введений користувачем тикер, дату початку та кінця. Якщо дані вдало завантажені, то вони зберігаються у змінній `self.df` та виводиться повідомлення про успішність процесу, як на рисунку 3.6. У іншому випадку (неправильно введений тикер чи відсутні дані за той період) виводяться на екрані повідомлення про помилку, як на рисунку 3.7.

```
Results
Data loaded for ticker: BTC-USD from 2023-01-01 to
2024-01-01
```

Рисунок 3.6 – Виведення повідомлення про успішне завантаження даних.

```
Results
No data found for ticker: вв in the selected date
range.
No data found for ticker: BTC-USD in the selected
date range.
```

Рисунок 3.7 – Виведення повідомлення про помилку завантаження даних.

5. `def load_data_from_file(self)` – ця функція дозволяє завантажувати дані з локальної папки у форматі `.csv`. Перевіряє наявність всіх полів у файлі, якщо перевірка успішна, то дані зберігаються у змінній `self.df`.

6. `def error_interface(self)` – ця функція перевіряє правильність введення даних користувачем. Тож він перевіряє наявність завантажених даних, валідність введеного текстового повідомлення та інші можливі помилки. Якщо помилка була знайдена, то все відображається у текстовому полі результатів, як було представлено у пункті 4.

7. `def ensembling_methods(self, df, lag)` – ця функція є відповідальною за технічний аналіз та прогнозування за допомогою машинного навчання. Здійснює введення ознак (технічних індикаторів, які були обговорені у пункті 3.1), формує дані перед машинним навчанням та тренує моделі. Результати роботи функції зберігаються та виводиться звітність про їх точність, як на рисунку 3.8.

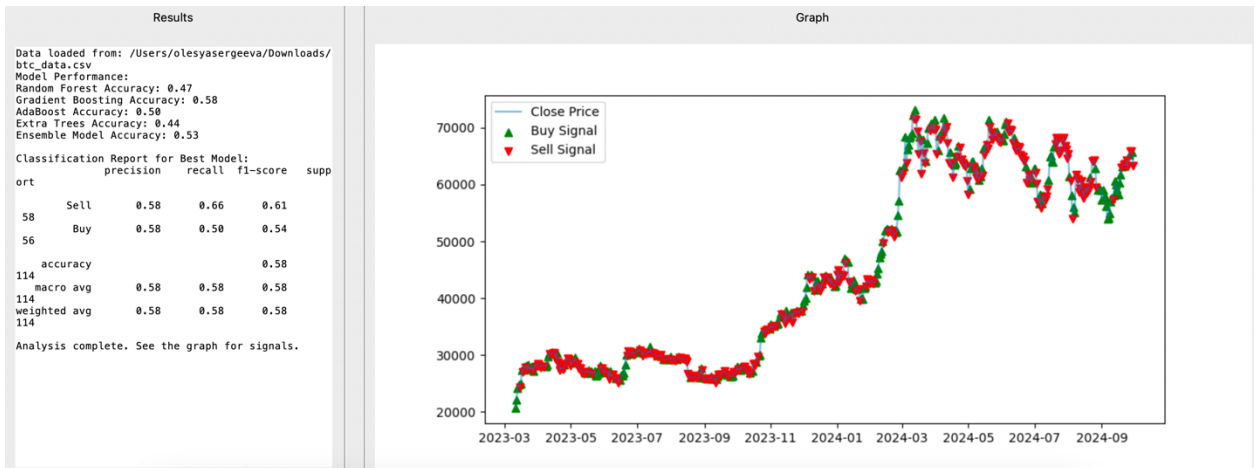


Рисунок 3.8 – Виведені результати та звітність моделей.

8. `def run_analysis(self)` – ця функція викликає основні методи аналізу даних. Спочатку викликає функцію `error_interface` для перевірки даних, а потім запускає `ensembling_methods` для створення прогнозування та застосування технічного аналізу. Результати роботи виводяться у полі результатів та графіку, як на рисунку 3.8.

9. `def optimize_strategy(self)` – ця функція оптимізує стратегію, використовуючи теорію ігор для аналізу сигналі купівлі/продажу. Вона порівнює стратегії на основі очікуваної вигоди і обирає оптимальні торгові сигнали для отримання максимально прибутку. Її результати роботи показані на рисунку 3.9.

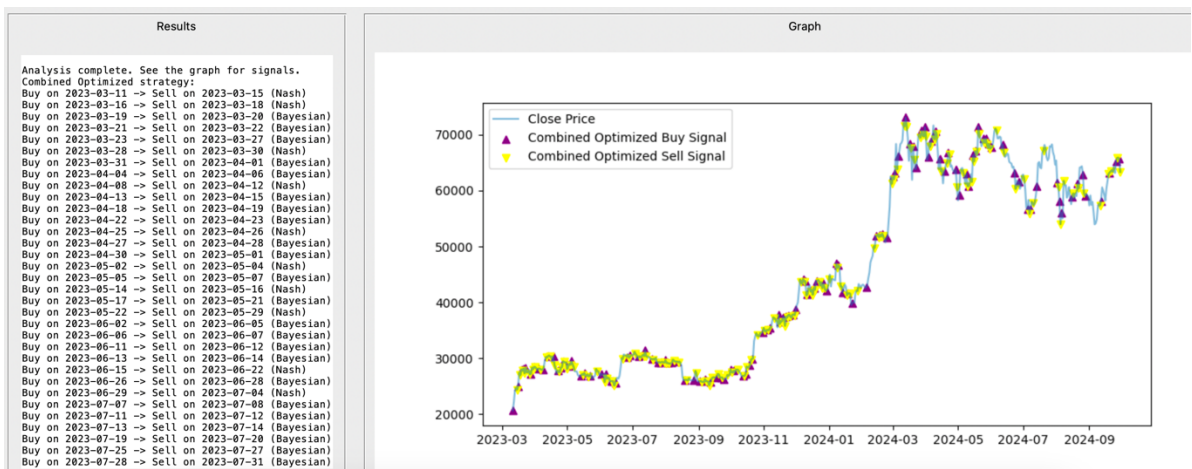


Рисунок 3.9 – Результат роботи оптимізації стратегії.

10. `def export_to_excel(self)` – ця функція дозволяє експортувати результати роботи у форматі xls для подальшого звітування. Приклад звіту представлено на рисунку 3.10.

	A	B	C	D
1		Buy Date	Sell Date	
2	0	2024-03-10 00:00:00	2024-03-12 00:00:00	
3	1	2024-03-13 00:00:00	2024-03-14 00:00:00	
4	2	2024-03-17 00:00:00	2024-03-18 00:00:00	
5	3	2024-03-20 00:00:00	2024-03-21 00:00:00	
6	4	2024-03-23 00:00:00	2024-03-27 00:00:00	
7	5	2024-03-28 00:00:00	2024-03-29 00:00:00	
8	6	2024-03-31 00:00:00	2024-04-01 00:00:00	
9	7	2024-04-03 00:00:00	2024-04-05 00:00:00	
10	8	2024-04-06 00:00:00	2024-04-09 00:00:00	
11	9	2024-04-10 00:00:00	2024-04-11 00:00:00	
12	10	2024-04-14 00:00:00	2024-04-15 00:00:00	
13	11	2024-04-16 00:00:00	2024-04-17 00:00:00	
14	12	2024-04-18 00:00:00	2024-04-21 00:00:00	
15	13	2024-04-22 00:00:00	2024-04-23 00:00:00	
16	14	2024-04-25 00:00:00	2024-04-26 00:00:00	
17	15	2024-04-29 00:00:00	2024-04-30 00:00:00	
18	16	2024-05-02 00:00:00	2024-05-06 00:00:00	
19	17	2024-05-09 00:00:00	2024-05-10 00:00:00	
20	18	2024-05-11 00:00:00	2024-05-14 00:00:00	
21	19	2024-05-15 00:00:00	2024-05-16 00:00:00	
22	20	2024-05-23 00:00:00	2024-05-28 00:00:00	
23				
24				

Рисунок 3.10 – Звіт з результатами роботи коду.

11. `def run_backtest(self)` – ця функція виконує бек-тестування на основі завантажених даних, застосовуючи ціну закриття активу. Вона викликає внутрішню функцію `BackTest` для розрахунку наступних параметрів Sortino Ratio, Beta, Alpha, Maximum Drawdown. Приклад виведення результату буде представлено на рисунку 3.11.

```
Sortino Ratio: 6.983
Beta: 0.042
Alpha: 5.086
Maximum Drawdown: 0.001%
```

Рисунок 3.11 – Виведення коефіцієнтів `BackTest`.

12. `def BackTest(self, serie, annualized_scalar=252 * 8)` – ця функція виконує аналіз історичних даних для визначення ефективної стратегії. Вона

розраховує коефіцієнти Sortino Ratio, Beta, Alpha, Maximum Drawdown на основі порівняння прибутковості активу S&P 500.

У цьому підрозділі було розглянуто всі функції, що утворюють один суцільний додаток, котрий має комфортний інтерфейс для зручності користувача та виконує всі функції, які необхідно мати програмному забезпеченню.

3.3 Процес машинного навчання

Розподіл даних відбувається наступним чином на навчальну та тестову вибірки. Ця процедура є важливою під час введення методів машинного навчання, оскільки це дозволить перевірити як вони узагальнюють інформацію на передчасно невідомих даних. Для навчання моделі виділяється 80% даних, а для тестування – 20%. Такий поділ дозволить уникнути перенавчання моделей.

Для прогнозування сигналів купівлі/продажу застосовуються наступні класифікаційні моделі:

1. Random Forest застосовує велику кількість дерев рішень, де кожне дерево може надавати власний прогноз, а остаточне рішення приймається на основі голосування дерев [12].

2. Gradient Boosting додає нові дерева, щоб виправити помилки попередніх дерев, що підвищує точність прогнозування [15].

3. AdaBoost – це метод ансамблевого навчання, що послідовно тренується декілька слабких моделей, кожна з яких зосереджена на прикладах, котрі були неправильно кваліфіковані іншими моделями [13].

4. Extra Trees є розширенням методу Random Forest, де ще додатково додається випадковість під час вибору розділень вузлів дерев [14].

5. Ensemble Model ґрунтується на тому, що декілька не складних моделей можуть давати високу точність у порівнянні з одним складним. Кожна модель спеціалізується на своїх частинах роботи, а кінцеве рішення відбувається шляхом узгодження моделей [16].

Таким чином, у підрозділі було розібрано процес машинного навчання програмного забезпечення.

3.4 Оптимізація результатів за допомогою теорії ігор

У даному підрозділі застосовано оптимізацію стратегії з використанням теорії ігор, зокрема рівновагу Неша та Байєсівських ігор. Це дозволяє трейдеру визначати оптимальні сигнали купівлі/продажу на основі технічних індикаторів та прогнозів моделі.

Відповідно розглянемо як відбувається цей процес по пунктах:

1. Отримання сигналів купівлі/продажу після машинного навчання.
2. Застосування рівноваги Неша. Відбувається через порівняння між цінами купівлі та продажу, де вигода визначається випадковою змінною.
3. Застосування Байєсівських ігор. Ймовірнісний спосіб, при якому включаються попередні знання про ринкові умови, тобто рішення приймають за допомогою прогнозів та очікувань з урахуванням поведінки ціни у майбутньому.
4. Фільтрація сигналів. Прибирає ті сигнали, що не відповідають чергуванню купівлі/продажу для того, щоб зменшити ризик та покращити оптимізації стратегії.
5. Створення комбінованої стратегії. Об'єднуємо в одну єдину персональну стратегію.

У висновку поєднання теорії ігор та машинного навчання дозволяє створити стабільну та надійну стратегію, що мінімізує ризики.

3.5 Аналіз результатів роботи ПЗ

У цьому підрозділі відбувається аналіз результатів роботи програмного забезпечення за такими пунктами:

- 1) оцінка точності прогнозування;
- 2) BackTest.

Спочатку почнемо з теоретичної частини оцінок точності прогнозування, а вже потім розглянемо у практичній реалізації:

1. Точність (Accuracy Score) – це оцінка, що основана на відношенні кількості правильних передбачень. Обчислюється за формулою (3.9) [27].

$$\text{Точність} = \frac{\text{Кількість правильних передбачень}}{\text{Загальна кількість передбачень}} \quad (3.9)$$

2. Класифікаційний звіт (Classification Report) включає в себе важливі метрики [27]:

а. Точність (Precision) – відсоток адекватних прогнозів серед усіх прогнозів. Обчислюється за формулою (3.10):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.10)$$

де TP – кількість правильних позитивних прогнозувань;

FP – кількість помилкових позитивних прогнозувань.

б. Повнота (Recall) – відсоток адекватних прикладів, котрі було прогнозовано. Обчислюється за формулою (3.11):

$$Recall = \frac{TP}{TP + FN}, \quad (3.11)$$

де TP – кількість правильних позитивних прогнозувань;

FN – кількість помилкових негативних прогнозувань.

в. F1-міра (F1-score) – гармонічне середнє між точністю та повнотою, тобто надає збалансовану оцінку моделі. Обчислюється за формулою (3.12):

$$F1 = \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.12)$$

г. Support – кількість реальних прикладів для кожного класу в тестовій вибірці.

На основі цих оцінок проаналізуємо точність прогнозування криптовалютних даних. Спочатку розглянемо вибірку криптовалюти Bitcoin протяжністю на один рік (від початку 2023 до початку 2024). Відповідно на рисунку 3.12 розглянемо оцінку точності прогнозу та на рисунку 3.13 буде відобразимо результат прогнозування у вигляді графіка.

```

Model Performance:
Optimized Random Forest Accuracy: 0.58
Optimized Gradient Boosting Accuracy: 0.58
Optimized AdaBoost Accuracy: 0.50
Optimized Extra Trees Accuracy: 0.61
Ensemble Model Accuracy: 0.62

Classification Report for Best Model:

```

		precision	recall	f1-score	supp
ort					
35	Sell	0.70	0.54	0.61	
29	Buy	0.57	0.72	0.64	
64	accuracy			0.62	
64	macro avg	0.64	0.63	0.62	
64	weighted avg	0.64	0.62	0.62	
64					

Рисунок 3.12 – Оцінка точності криптовалюти Bitcoin протяжністю в один рік.

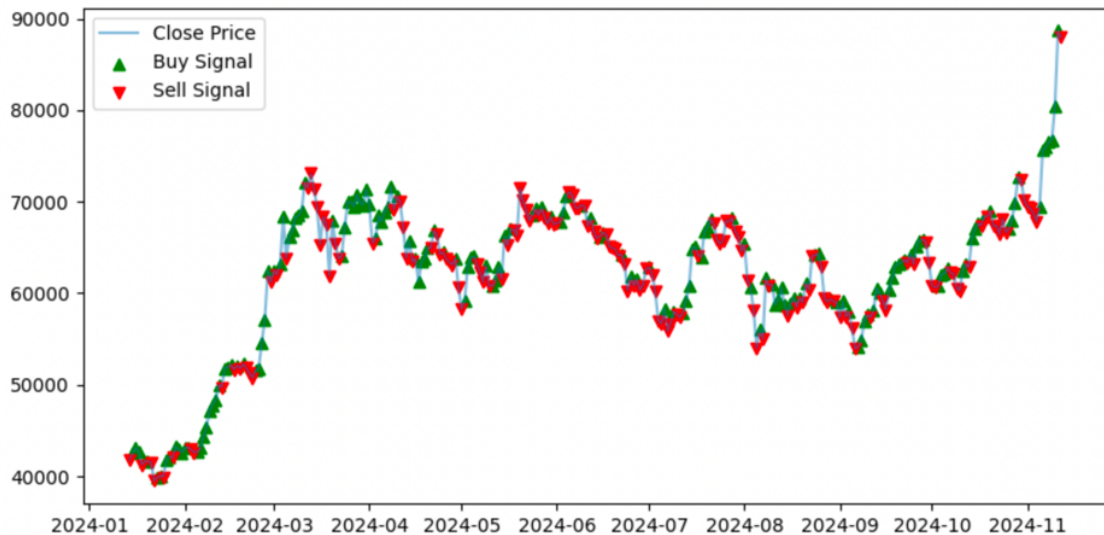


Рисунок 3.13 – Результат прогнозування у вигляді графіка для криптовалюти Bitcoin.

За рисунком 3.12 можна зробити такий висновок:

- 1) найпродуктивнішою моделлю є Ensemble Model з точністю 52% правильних прогнозів;
- 2) сигнал «Sell» (продажів): точність становить 0.7, повнота – 0.54, f1-оцінки – 0.61, тобто це свідчить про те, що з усіх випадків прогнозів були правильними 70% та з усіх реальних прогнозів були вірно прогнозовані 54%;
- 3) сигнал «Buy» (купівля): точність становить 0.57, повнота – 0.72, f1-оцінки – 0.64;
- 4) макро середні показники вказують на збалансовану продуктивність моделі між класами;
- 5) зважене середнє теж стверджує на стабільність моделі.

Висновок: сигнал «Buy» показав кращу повноту порівняно з показником «Sell», що вказує на здатність моделі краще передбачати сигнали купівлі та про можливе зростання цін.

Так само розглянемо іншу криптовалюту, наприклад Ethereum на вибірку продовжністю в один (від початку 2022 до початку 2023). Та подивимося як буде реагувати програма на зменшення вибірки. На рисунку

3.14 буде представлено оцінка точності прогнозу та на рисунку 3.15 – результати у вигляді графіку.

```
Data loaded for ticker: ETH-USD from 2022-01-01 to
2023-01-01
Model Performance:
Random Forest Accuracy: 0.52
Gradient Boosting Accuracy: 0.45
AdaBoost Accuracy: 0.62
Extra Trees Accuracy: 0.48
Ensemble Model Accuracy: 0.47

Classification Report for Best Model:

```

ort		precision	recall	f1-score	supp
33	Sell	0.67	0.61	0.63	
27	Buy	0.57	0.63	0.60	
60	accuracy			0.62	
60	macro avg	0.62	0.62	0.62	
60	weighted avg	0.62	0.62	0.62	

```
Analysis complete. See the graph for signals.
```

Рисунок 3.14 – Оцінка точності криптовалюти Ethereum протяжністю в рік.

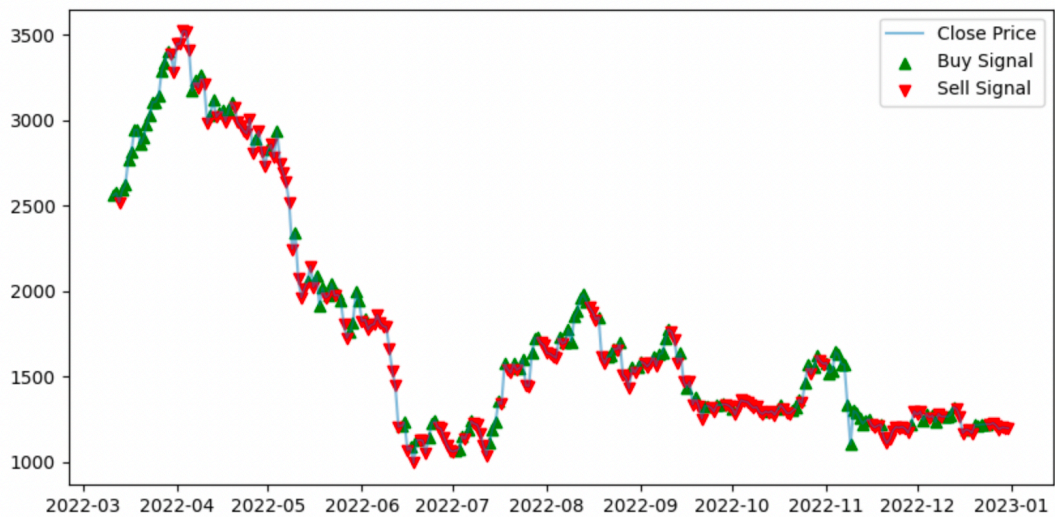


Рисунок 3.15 – Результат прогнозування у вигляді графіка для криптовалюти Ethereum.

За рисунком 3.14 можна зробити такий висновок:

- 1) найпродуктивнішою моделлю є AdaBoost з точністю 62%;

2) сигнал «Sell» (продажів): точність становить 0.67, повнота – 0.61, модель передбачила 67% правильних сигналів продажу та правильно визначила 61% із тих сигналів, котрі дійсно мають бути передбачені;

3) сигнал «Buy» (купівля): точність становить 0.57, повнота – 0.63, що є приблизно такою ж самою як сигнал «Sell»;

4) макро середні показники свідчать про стабільність продуктивності моделі між сигналами;

5) зважене середнє підтверджує узгодженість результатів моделі.

Висновок: модель демонструє дещо кращу повноту для сигналів «Buy» - 63% та «Sell» - 61%, що свідчить про здатність моделі передбачати сигнали купівлі та продажу з досить рівномірною точністю.

Далі було проаналізовано ефективність стратегії кожної моделі, але перед тим як це почати робити розглянемо теоретичну частину коефіцієнтів BackTest [16]:

1. Sortino Ratio застосовуються для оцінки ризику та прибутковості криптовалюти. Обраховується наступним чином за формулою (3.13):

$$\text{Sortino Ratio} = \frac{\mu - rf}{\text{downward}} \quad (3.13)$$

де μ – середнє значення доходності портфеля (за рік);

rf - безризикова ставка;

downward – стандартне відхилення негативних доходів.

Як аналізувати результат роботи:

- а. Sortino Ratio < 0: вкладення є збитковим, при цьому ризик низький.
- б. $0 < \text{Sortino Ratio} < 1$: вкладення хороше та ризик прийнятний.
- в. Sortino Ratio > 1: вкладення відмінне та низький ризик .

2. Beta вказує на скільки чутлива ціна фінансового активу на зміни ринку. Обчислюється за формулою (3.14):

$$Beta = \frac{Cov(rs, rm)}{Var(rm)} \quad (3.14)$$

де rs – прибутковість фінансового активу;

rm – прибутковість ринку.

Як аналізувати результат роботи:

- а. $Beta < 1$: фінансовий актив менш мінливий, ніж ринок.
- б. $Beta = 1$: фінансовий актив рухається з ринком.
- в. $Beta > 1$: фінансовий актив більш мінливий, ніж ринок, тому має високий ризик.

3. Alpha показує перевищення чи недостачу прибутковості фінансового активу відносно очікувань. Обчислюється за формулою (3.15):

$$Alpha = Ers - rf - Beta(Erm - rf) \quad (3.15)$$

де Ers – середня прибутковість фінансового активу;

Erm – середня ринкова прибутковість

rf – безризикова ставка.

Як аналізувати результат роботи:

- а. $Alpha < 0$: низька віддача прибутковості.
- б. $Alpha = 0$: актив відповідає ринковій прибутковості.
- в. $Alpha > 0$: актив перевищує ринкову прибутковість.

4. Maximum Drawdown – це метрика, котра визначає максимальну втрату фінансів від пікової вартості до найнижчої за певний проміжок часу. Чим вище її значення, тим більший ризик. Обчислюється за формулою (3.16):

$$\delta_i = \frac{\varphi_i}{\Phi_i} - 1 \quad (3.16)$$

де φ_i – сукупна продуктивність продукту на момент i ;

Φ_i – максимальна продуктивність на момент i .

Відповідно для тієї самої вибірки Bitcoin протяжністю один рік (від початку 2023 до початку 2024) створимо оптимізовану стратегію, що зображено на рисунку 3.16 та графіку рисунка 3.17, та проаналізуємо за допомогою вище сказаних коефіцієнтів, які покажемо на рисунку 3.18.

```
Combined Optimized strategy:  
Buy on 2023-03-11 -> Sell on 2023-03-15 (Nash)  
Buy on 2023-03-16 -> Sell on 2023-03-18 (Nash)  
Buy on 2023-03-19 -> Sell on 2023-03-20 (Bayesian)  
Buy on 2023-03-21 -> Sell on 2023-03-22 (Bayesian)  
Buy on 2023-03-23 -> Sell on 2023-03-24 (Bayesian)  
Buy on 2023-03-25 -> Sell on 2023-03-27 (Bayesian)  
Buy on 2023-03-28 -> Sell on 2023-04-01 (Nash)  
Buy on 2023-04-04 -> Sell on 2023-04-06 (Bayesian)  
Buy on 2023-04-08 -> Sell on 2023-04-12 (Nash)  
Buy on 2023-04-13 -> Sell on 2023-04-15 (Bayesian)  
Buy on 2023-04-18 -> Sell on 2023-04-19 (Bayesian)  
Buy on 2023-04-22 -> Sell on 2023-04-23 (Bayesian)  
Buy on 2023-04-25 -> Sell on 2023-04-27 (Nash)  
Buy on 2023-04-30 -> Sell on 2023-05-01 (Bayesian)  
Buy on 2023-05-02 -> Sell on 2023-05-04 (Nash)  
Buy on 2023-05-05 -> Sell on 2023-05-06 (Bayesian)  
Buy on 2023-05-14 -> Sell on 2023-05-16 (Nash)  
Buy on 2023-05-17 -> Sell on 2023-05-18 (Bayesian)  
Buy on 2023-05-19 -> Sell on 2023-05-24 (Bayesian)  
Buy on 2023-05-25 -> Sell on 2023-05-29 (Nash)  
Buy on 2023-06-02 -> Sell on 2023-06-03 (Bayesian)  
Buy on 2023-06-04 -> Sell on 2023-06-05 (Bayesian)  
Buy on 2023-06-06 -> Sell on 2023-06-07 (Bayesian)  
Buy on 2023-06-08 -> Sell on 2023-06-10 (Bayesian)  
Buy on 2023-06-11 -> Sell on 2023-06-12 (Bayesian)  
Buy on 2023-06-13 -> Sell on 2023-06-14 (Bayesian)  
Buy on 2023-06-15 -> Sell on 2023-06-24 (Nash)  
Buy on 2023-06-27 -> Sell on 2023-06-28 (Bayesian)  
Buy on 2023-06-29 -> Sell on 2023-07-04 (Nash)  
Buy on 2023-07-07 -> Sell on 2023-07-08 (Bayesian)  
Buy on 2023-07-10 -> Sell on 2023-07-12 (Bayesian)  
Buy on 2023-07-13 -> Sell on 2023-07-14 (Bayesian)  
Buy on 2023-07-19 -> Sell on 2023-07-20 (Bayesian)
```

Рисунок 3.16 – Оптимізована стратегія для криптовалюти Bitcoin в межах одного року.

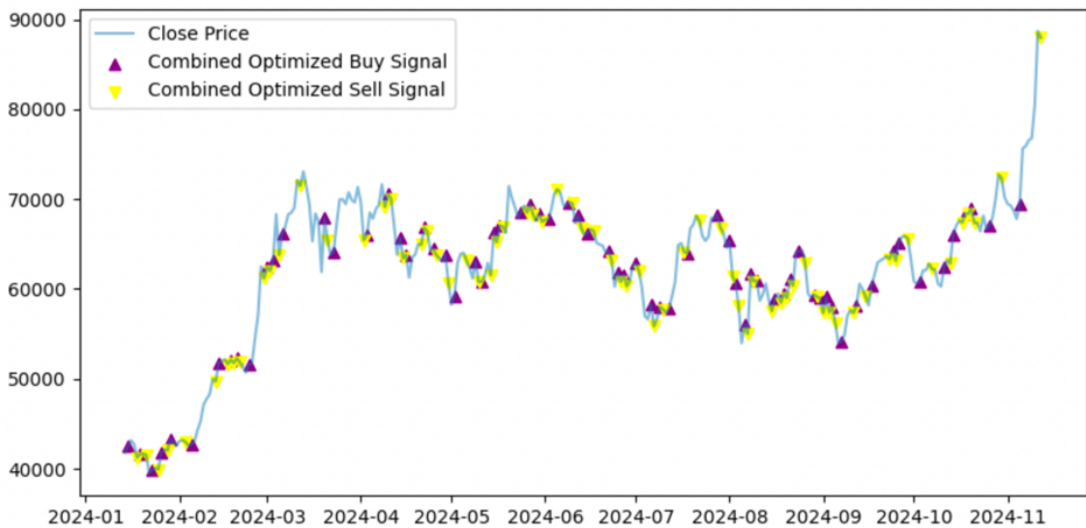


Рисунок 3.17 – Графік оптимізованих сигналів купівлі/продажу Bitcoin в межах одного року.

Sortino Ratio: 8.567
Beta: 0.007
Alpha: 5.775
Maximum Drawdown: 0.001

Рисунок 3.18 – BackTest криптовалюти Bitcoin в межах одного року.

З рисунку 3.18 можна зробити висновок, що за коефіцієнтом:

- 1) Sortino Ratio = $8.567 > 1$: прибуток є дуже високим та має низькі ризики;
- 2) Beta = $0.007 < 1$: фінансовий актив практично не реагує на зміни ринку, що є позитивним з точки зору системних ризиків;
- 3) Alpha = $5.775 > 0$: фінансовий актив перевищує ринкову прибутковість, тобто приносить додаткову прибутковість, що може вказувати на ефективність стратегії;
- 4) Maximum Drawdown = $0.001 = 0\%$: вказує на відсутність великих ризикових втрат.

Висновок: показники свідчать, що інвестиція є прибутковою та підходить інвесторам, котрі хочуть отримати високий дохід з мінімальними ризиками.

Так само зробимо для вибірки нової вибірки з на рік (з початку 2021 до початку 2022) введемо оптимізовану стратегію на рисунку 3.19 і рисунку 3.20 та проаналізуємо коефіцієнти з рисунку 3.21.

```

Analysis complete. See the graph for signals.
Combined Optimized strategy:
Buy on 2021-03-11 -> Sell on 2021-03-12 (Bayesian)
Buy on 2021-03-13 -> Sell on 2021-03-14 (Bayesian)
Buy on 2021-03-16 -> Sell on 2021-03-18 (Nash)
Buy on 2021-03-19 -> Sell on 2021-03-22 (Bayesian)
Buy on 2021-03-23 -> Sell on 2021-03-24 (Bayesian)
Buy on 2021-03-26 -> Sell on 2021-04-03 (Nash)
Buy on 2021-04-04 -> Sell on 2021-04-15 (Nash)
Buy on 2021-04-17 -> Sell on 2021-04-21 (Nash)
Buy on 2021-04-23 -> Sell on 2021-05-01 (Nash)
Buy on 2021-05-02 -> Sell on 2021-05-04 (Bayesian)
Buy on 2021-05-05 -> Sell on 2021-05-06 (Bayesian)
Buy on 2021-05-07 -> Sell on 2021-05-10 (Nash)
Buy on 2021-05-11 -> Sell on 2021-05-12 (Bayesian)
Buy on 2021-05-14 -> Sell on 2021-05-15 (Bayesian)
Buy on 2021-05-20 -> Sell on 2021-05-21 (Bayesian)
Buy on 2021-05-24 -> Sell on 2021-05-25 (Bayesian)
Buy on 2021-05-26 -> Sell on 2021-05-27 (Bayesian)
Buy on 2021-05-30 -> Sell on 2021-06-04 (Nash)
Buy on 2021-06-13 -> Sell on 2021-06-16 (Bayesian)
Buy on 2021-06-17 -> Sell on 2021-06-18 (Bayesian)
Buy on 2021-06-21 -> Sell on 2021-06-22 (Bayesian)
Buy on 2021-06-23 -> Sell on 2021-06-25 (Bayesian)
Buy on 2021-06-27 -> Sell on 2021-06-28 (Bayesian)
Buy on 2021-06-29 -> Sell on 2021-07-01 (Bayesian)
Buy on 2021-07-03 -> Sell on 2021-07-10 (Nash)
Buy on 2021-07-11 -> Sell on 2021-07-12 (Bayesian)
Buy on 2021-07-14 -> Sell on 2021-07-16 (Bayesian)
Buy on 2021-07-18 -> Sell on 2021-07-19 (Bayesian)
Buy on 2021-07-21 -> Sell on 2021-07-28 (Nash)
Buy on 2021-07-29 -> Sell on 2021-07-31 (Nash)

```

Рисунок 3.19 – Оптимізована стратегія для криптовалюти Ethereum в межах одного року.

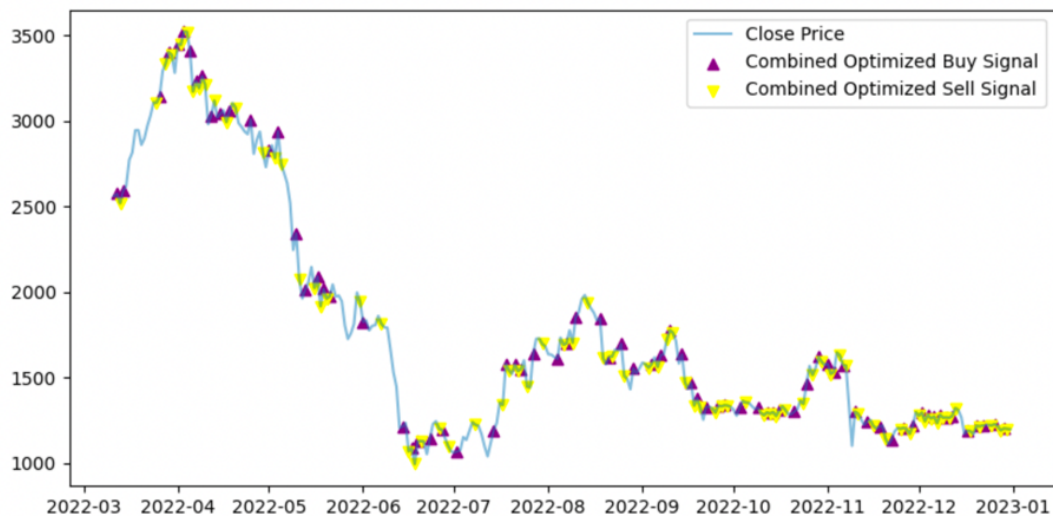


Рисунок 3.20 – Графік оптимізованих сигналів купівлі/продажу Ethereum в межах одного року.

```

Sortino Ratio: 12.125
Beta: -0.091
Alpha: 25.100
Maximum Drawdown: 0.154

```

Рисунок 3.21 – BackTest криптовалюти Ethereum в межах одного року.

З рисунка 3.21 можна зробити висновок, що актив не вигідний:

1) Sortino Ratio = 12.125 > 0: стратегія забезпечує значну дохідність при низьких ризиках;

2) Beta = -0.091 < 1: вказує на те, що актив не реагує на ринкові зміни;

3) Alpha = 25.1 > 1: стратегія надає значну прибутковість та перевищує ринкові очікування;

4) Maximum Drawdown = 0,154 = 15%: відсоток системного ризику є невеликим, тому можна сказати, що він не сильно вплине, оскільки прибутковість висока.

Висновок: стратегія є дуже ефективною та приносить високий рівень дохідності.

Відповідно, у цьому підрозділі проаналізовано прогнозування та оптимізацію з точки зору ефективності роботи. Загалом аналіз показав, що хоча і є певні неточності прогнозування за рахунок високої волатильності даних, розроблені стратегії можуть мінімізувати ризики та за забезпечувати прибуток.

3.6 Висновки до розділу 3

Під час роботи над розділом 3 розроблено програмне забезпечення для прогнозування сигналів купівлі/продажу на основі машинного навчання, а також алгоритм оптимізації стратегії за допомогою теорії ігор.

Спочатку було розглянуто ключові методи обробки даних (стандартизація, додавання ознак та інше) для побудови моделі. Далі основну увагу було зосереджено на процесі машинного навчання та оптимізації. В кінці відбувається тестування програмного забезпечення для оцінювання її продуктивності.

Відповідно до цього у кодї застосовано технічні індикатори (ознаки): SMA, MSD, RSI, Ichimoku Cloud з методами машинного навчання такими як, Random Forest, AdaBoost, Extra Trees, Gradient Boosting, що дозволило отримати точні результати прогнозування. Крім того, застосовано теорію ігор для реалізації алгоритму оптимізації стратегії, що застосовують рівновагу Неша та Байєсівські ігри.

Оцінювання моделей відбувалося за допомогою метрик якості: точність та класифікаційний звіт (Precision, Recall, F1-score), а оцінювання алгоритму оптимізації стратегії – BackTest: Sortino Ratio, Beta, Alpha, Maximum Drawdown, що дозволило підтвердити ефективність оптимізованої стратегії, що зменшило ризики та підвищило стабільність результатів. Дані оцінки підтверджують ефективність роботи моделей та алгоритму.

Згідно з аналізом встановлено, що поєднання машинного навчання та теорії ігор є ефективним підходом для розробки стратегій купівлі/продажу криптовалют, що забезпечує кращу адаптацію до змін на ринку.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї продукту

Проект спрямований на створення аналітичної програми для дослідження та прогнозування сигналів купівлі/продажу на основі методів машинного навчання та теорії ігор. Стартап-проект надаватиме користувачеві прогноз цін криптовалюти та оптимізацію виведеної стратегії.

У межах цього підрозділу буде розглянуто наступні пункти:

- 1) зміст ідеї;
- 2) напрямки застосування;
- 3) вигоди для користувачів;
- 4) чим відрізняється від потенційних конкурентів.

У таблиці 4.1 представлені перші три пункти.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Аналітична програма для ринку криптовалют на основі машинного навчання та теорії ігор, що прогнозує сигнали купівлі/продажу та створює оптимізовану стратегію.	1. Фінансовий/технічний аналіз.	1. Автоматичний прогноз цін купівлі/продажу.
	2. Трейдинг криптовалют.	
	3. Прогнозування цін.	2. Оптимізація стратегії.
	4. Стратегічне планування.	

Ідея стартап-проекту має головні техніко-економічні переваги:

- 1) застосування машинного навчання для прогнозування ринкових сигналів, що дозволяє збільшити точність та якість передбачення;
 - 2) застосування теорії ігор дозволяє оптимізувати стратегію;
 - 3) автоматичні інструменти дозволяють заощадити час користувачів.
- Всі ці переваги представлено у таблиці 4.2.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепція конкурентів				W	N	S
		Даний проект	Coin Market Cap	Crypto Compare	Trading View			
1.	Прогнозування цін купівлі/продажу на основі ML	Застосовує	Не застосовує	Частково застосовує	Не застосовує		+	
2.	Оптимізація стратегії на основі теорії ігор	Застосовує	Не застосовує	Не застосовує	Не застосовує			+
3.	Автоматизація процесів	Присутнє	Немає	Частково	Частково		+	
4.	Точність прогнозів	Висока	Традиційні методи аналізу	Середня точність	Лише технічні індикатори		+	
5.	Підключення до криптовалютних бірж	Використовується	Частково використовується	Частково використовується	Використовується		+	

Такі таблиці дозволяють визначити слабкі чи сильні сторони проекту, сформувавши конкурентоспроможність та виявити можливі шляхи покращення.

Аналіз ідеї проекту показує наглядно, що при застосуванні методів машинного навчання та теорії ігор стартап має домінуючу перевагу над своїми конкурентами.

4.2 Технологічний аудит ідеї продукту

Мета цього підрозділу – провести технологічний аудит для визначення можливості реалізації ідеї проекту. Стартап-проект це аналітичне програмне забезпечення, що застосовує сучасні методи штучного інтелекту, а саме машинне навчання, та теорія ігор для оптимізації стратегії. У межах цього підрозділу аналізуються всі можливі і доступні технології, які застосовуються під час створення самого програмного забезпечення. У таблиці 4.3 є опис всієї технологічної бази.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Прогнозування сигналів ціни купівлі/продажу на основі машинного навчання	Алгоритми машинного навчання, такі як Random Forest, Gradient Boosting, AdaBoost, Extra-Tree та інше.	Існують та застосовуються	Доступні через мову програмування Python та її бібліотек

Продовження таблиці 4.3

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
2.	Оптимізація стратегії на основі теорії ігор	Моделювання стратегій зі застосування рівноваги Неша та Байєсових ігор	Існують, потребують адаптації	Доступні через мову програмування Python
3.	Візуалізація торгових сигналів та результатів аналізу	Застосування бібліотек мови програмування для візуалізації	Існують та застосовуються	Доступні через мову програмування Python та її бібліотек
4.	Реалізація інтерфейсу для користувачів	Tkinter для взаємодії з користувачем	Існують та застосовуються	Доступні через мову програмування Python та її бібліотек
5.	Підключення фінансових даних	Yahoo Finance	Існують та застосовуються	Доступний через відкритий API Yahoo Finance
6.	Back testing торгових стратегій	Застосування історичних даних для аналізу стратегії	Існують та застосовуються	Доступні через мову програмування Python та її бібліотек

У висновку технічна складова проекту можлива, оскільки доступні всі компоненти для відтворення ідеї. Алгоритми машинного навчання, візуалізація даних, реалізація інтерфейсу та Back testing є широко доступними, що дозволяє легко інтегрувати ідею. Оптимізація теорії ігор можлива, але необхідно адаптувати до кодового вигляду. Також, підключення фінансових даних є доступним для користувачів через API Yahoo Finance.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Даний підрозділ розглядає ринкові можливості та загрози, що можуть виникнути під час запуску стартап-проекту. Цей аналіз дозволить визначити чи є цей проект привабливий для користувачів та оцінити переваги та недоліки, що можна включити під час створення маркетингової стратегії.

Для визначення попиту програмного забезпечення у таблиці 4.4 представлено аналіз стану ринку, обсяг продаж, динаміка та обмеження впровадження.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од.	15 платформ
2.	Загальний обсяг продаж, грн/ум.од	500 млн грн. на рік
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Середній рівень, оскільки необхідно сертифікати чи ліцензії з регулювання криптовалюти

Продовження таблиці 4.4

№ п/п	Показники стану ринку (найменування)	Характеристика
5.	Специфічні вимоги до стандартизації та сертифікації	Захист даних
6.	Середня норма рентабельність в галузі (або по ринку), %	15-20%

Ринок криптовалют активно розвивається та обсяг продажів зростає із кожним роком, також попит на продукцію підтверджується не строгими обмеженнями та динамічним зростанням.

У таблиці 4.5 розглянуті потреби та вимоги споживачів до проекту, що дозволить визначити важливий сегмент ринку.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Аналіз ринкових трендів для трейдингу	Приватні інвестори	Ризикові та чутливі до перемін ринку	Гнучкість застосування, точні прогнози
2.	Оптимізація стратегія	Трейдери, професійні інвестори	Короткострокові рішення, автоматизація процесів	Автоматизація, підключення бірж

Продовження таблиці 4.5

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
3.	Аналітика ринку для керування фінансовими інструментами	Інвестиційні фонди	Довгострокові та закостенілі рішення	Стабільність та достовірні прогнози

У таблицях 4.6 та 4.7 розглянуто загрози та можливості проекту для аналізу ринкових дій.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Невизначеність законодавства стосовно криптовалют	Посилення керування, що може обмежити доступ до ринку	Відповідати потребам законодавства (ліцензування та інше)
2.	Конкуренція з подібними платформами	Створення подібного функціонального проекту до випуску самого проекту та під час роботи	Відмінний маркетинговий план, пошук відмінностей від конкурентів

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання зацікавленості до криптовалютного трейдингу	Становлення популярності криптовалюти серед інвесторів/трейдерів	Надавати новим користувачам унікальні рішення
2.	Новітня технологія на ринку	Зацікавленість у використанні машинного навчання та теорії ігор разом	Звернути увагу на високу точність прогнозування

У таблиця 4.8 та 4.9 відбувається аналіз конкуренції на усіх етапах.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія	Кілька гравців контролюють ринок, однак є простір для стартапів	Розробка унікальних рішень, створення відмінностей у продукції
2. За рівнем конкурентної боротьби - національний	Конкуренція зазвичай відбувається на національному рівні з плавним переходом на міжнародний	Пошук міжнародних партнерів, розширення поля присутності

Продовження таблиці 4.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
3. За галузевою ознакою - внутрішньогалузева	Проявляється в конкуренції з подібними платформами	Покращення функціоналу для аналізу ринку та точності прогнозів
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між подібні аналітичні додатки для криптовалют	Розширення функціоналу для аналізу ринку та точності прогнозів
5. За характером конкурентних переваг - нецінова	Основну роль відіграє новизна технології, точність прогнозу та аналітики	Розробка нових ефективних моделей для покращення прогнозу
6. За інтенсивністю - марочна	Має високу інтенсивність, з акцентом на відомість та платформи, що містять базу користувачів та сильну позицію	Звернути увагу на маркетинг та брендинг, розширення бази клієнтів

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінників
Складові аналізу	Coin Market Cap, Crypto Compare, Trading View	Нові стартап-проекти, фінансові компанії	Крипто-біржі та АРІ, що постачать дані для роботи програми	Інвестори та трейдери створюють вимоги до функціоналу та якості прогнозів	Традиційні фінансові ринки, аналітичні інструменти, можуть створювати обмеження по залученню клієнтів
Висновки	Конкуренти достатньо сильні, оскільки мають репутацію та базу користувачів	Має досить легкі умови для входу стартап-проекту, може ускладнити становлення на ринку та викликати велику конкуренцію	Постачальники займають головне місце для надійної роботи програми	Клієнти мають високі вимоги до продукту, що змушує компанію покращувати функціонал	Товари-замінники можуть впливати на користувачів, що становить велику загрозу для проекту, тому потрібно створювати нові функції та покращувати старий функціонал

Ринок криптовалют є стрімким, тому тут є місце для нових стартап проектів, особливо тих, хто пропонує новітні технології.

У таблицях 4.10, 4.11, 4.12 та 4.13 розглядаються сильні та слабкі сторони проекту, які необхідно враховувати під час створення проекту та маркетингової стратегії.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Машинне навчання для прогнозування	Є ознакою, що приваблює користувачів
2.	Теорія ігор для оптимізації	Є ознакою, що приваблює користувачів
3.	Автоматизація процесу	Є ознакою, що приваблює користувачів та робить процес трейдингу легшим
4.	Візуалізація торгових сигналів	Є ознакою зручності для користувачів
5.	Підтримка криптовалютних бірж	Є ознакою зручності для користувачів та налагодженого партнерства
6.	Висока точність прогнозування	Є ознакою зручності для користувачів, створює високу довіру користувачів

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін
«FinancialAnalysisApp»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «FinancialAnalysisApp»							
			-3	-2	-1	0	1	2	3	
1.	Машинне навчання для прогнозування	18								+
2.	Теорія ігор для оптимізації	17							+	
3.	Автоматизація процесу	16							+	
4.	Візуалізація торгових сигналів	14				+				
5.	Підтримка криптовалютних бірж	15						+		
6.	Висока точність прогнозування	19								+

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: Новизна технологій, Висока точність прогнозів	Слабкі сторони: Обмежений бюджет, відсутність ліцензії
Можливості: Розширення на ринок криптовалют	Загрози: Посилення законодавства

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Партнерство з криптобіржами	90%	6 місяців
2.	Співпраця з інвестиційними фондами	50%	8-10 місяців
3.	Маркетингова стратегія	75%	3-5 місяців
4.	Пошук інвесторів для входу на більший ринок	50%	12-18 місяців
5.	Випуск мобільного додатку для збільшення клієнтури	80%	10-12 місяців

У висновку аналіз показує високий потенціал проекту. Незважаючи на ризики конкуренції та законодавчі загрози, проект має значний шанс впровадження через партнерство та маркетингову стратегію.

4.4 Розроблення ринкової стратегії проекту

Цей підрозділ відповідає за створення ринкової стратегії для стартап-проекту. Для цього необхідно визначити цільову групу потенційних споживачів, стратегії розвитку, конкретної поведінки та позиціонування.

На основі аналізу попереднього підрозділу виділю основні цільові групи, що розглянуті у таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Приватні інвестори, що працюють з криптовалютами	Висока	Середня	Середня	Висока
2.	Трейдери та професійні інвестори	Дуже висока	Висока	Висока	Середня
3.	Інвестиційні фонди, що управляють фінансовими активами	Помірно	Середня	Низька	Низька
Які цільові групи обрано: приватним інвесторам, трейдерам та професійним інвесторам є найбільш зацікавленими у такому продукті та готові прийняти його.					

У таблиці 4.15 розглянуто стратегію розвитку проекту в обраних сегментах.

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Партнерство з криптобіржами для розширення клієнтури	Зосередити увагу на одному сегменті	Висока точність прогнозування, автоматизація процесу	Новітнє лідерство
2.	Співпраця з інвестиційними фондами для покращення стратегії	Розробити відмінну характеристику продукту для приваблення користувачів	Точність прогнозів, довгострокові рішення	Продуктова диференціація

У таблиці 4.16 розглянуто створення стратегії конкретної поведінки на ринку, що дозволить ефективно конкурувати.

Таблиця 4.16 – Визначення базової стратегії конкретної поведінки

№ п/п	Чи є проект «першо-хідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні	Компанія буде одночасно шукати нових споживачів та забирати існуючих у конкурентів	Не копіювати, застосовувати новітні та популярні функції	Комбінована (новітнє лідерство, продуктова диференціація та атака на ринок)

У таблиці 4.17 розглянуто стратегію позиціонування, яке дозволить користувачам створити своє сприйняття продукту.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових)
1.	Висока точність прогнозів, автоматизація процесів, партнерство з біржами	Новітнє лідерство	Точність, автоматизація, партнерство	Новизна, технологічна перевага, точність
2.	Стійкість та довгостроковість прогнозів	Продуктова диференціація	Стійкість, стабільність	Довгостроковість, стійкість, стабільність

У висновку, ринкова стратегія ґрунтується на новітньому лідерстві та продуктивній диференціації, що дозволить утриматися на ринку серед конкурентів.

4.5 Розроблення маркетингової програми стартап-проекту

У цьому підрозділі буде створено маркетингову програму для стартап-проекту, він буде включати ключові переваги потенційного товару, три рівні моделі товару, цінове утворення, система збуту та концепція маркетингової комунікації. У таблиці 4.18 на основі попереднього аналізу конкурентоспроможності визначено переваги продукту.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Прогнозування сигналів купівлі/продажу криптовалют	Висока точність прогнозів	Використання декількох методів машинного навчання
2.	Оптимізація стратегії	Знаходження найкращої стратегії	Використання теорії ігор
3.	Автоматизація процесу	Автоматизація традиційного процесу	Партнерство з криптобіржами

У таблиці 4.19 описується три рівні становлення товару: ідея, реальна реалізація та підкріплення.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Задовольняє проблему у точному прогнозі цін купівлі/продажу та оптимізації рішення.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Прогнозування сигналів купівлі/продажу	1. Нм	1. Технологічна
	2. Оптимізація стратегії	2. Нм	2. Технологічна
	Якість: тестування на точність прогнозування та стратегії		
	Пакування: користувацький інтерфейс, що можна завантажити у комп'ютер		
Марка: CryptoAnalytics LLC; FinancialAnalysisApp.			
III. Товар із підкріпленням	До продажу: демо версія додатку, технічна консультація для користувачів перед придбанням		
	Після продажу: цілодобова технічна підтримка, регулярні покращення, адаптація до нових умов на ринку		
За рахунок чого потенційний товар буде захищено від копіювання: буде запатентовано			

У таблиці 4.20 порівнюються ціни конкурентів, дохід користувачів та приблизна ціна товару.

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товар-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	50\$ на місяць	85\$ на місяць	Середній дохід: 1000\$	Нижня межа: 60\$ Верхня межа: 100\$

У таблиці 4.21 для визначення системи збуту аналізуємо особливості клієнтів та вимоги збуту.

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Приватні інвестори/трейдери купують продукти онлайн на веб- платформах	Зручний веб- платформа зі швидкою реєстрацією та здійснення покупок онлайн	Пряма взаємодія з клієнтом	Власна онлайн- платформа з підключенням онлайн платежів
2.	Приватні інвестори/трейдери потребують індивідуальні послуги	Забезпечення індивідуальної послуги	Глибокий канал (індивідуал ьний підхід)	Взаємодія з бізнес партнерами, надання персональних консультацій

Таблиця 4.22 націлена на визначення можливості просування продукту серед цільової клієнтури через канали збуту.

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Приватні інвестори/трейдери шукають додатки для покращення прогнозу	Соціальні мережі, вебінари	Точність прогнозів	Підвищити довіру клієнтів, продемонструвавши точність прогнозу	Збільшити свій пасивний дохід за допомогою точних прогнозів
2.	Приватні інвестори/трейдери цінують швидкість та автоматизацію процесів	Форуми, конференції, пряма реклама	Автоматизація рішень, інтеграція з біржами	Показати переваги автоматизацію та партнерства з біржами	Приймати рішення швидше та точніше з Financial Analysis App

У висновку маркетингова програма створює комплексний підхід для виходу продукту на ринок, включаючи всі аспекти, цінову політику, систему збуту та комунікацію з клієнтурою.

4.6 Висновки до розділу 4

В даних умовах стартап-проект може мати успішну комерціалізацію. Попит на інструменти аналізу ринку криптовалют є стабільним та поступово зростає, тому що цей ринок розвивається та все більше привертає увагу інвесторів, трейдерів, інвестиційних фондів та інших. Динаміка ринку є достатньо позитивною та рентабельність проекту висока через застосування передових та незвичних технологій, таких як машинне навчання та теорія ігор.

Якщо говорити про перспективи впровадження, то з точки аналізу вони достатньо хороші, включаючи широку групу потенційних користувачів. Обмеження для провадження є низькими, оскільки ринок є готовим сприймати стартап-проект, якщо він є новітнім та вирішує потенційні проблеми. Хоча конкуренція в цій галузі достатньо висока, однак конкурентоспроможність проекту забезпечується точним прогнозуванням цін купівлі/продажу, оптимізацією стратегії та автоматизацією процесу.

Для реалізації ринкового впровадження проекту необхідно застосувати стратегію концентрованого маркетингу з акцентом на головну цільову аудиторію. Найоптимальніша альтернатива впровадження передбачає взаємодію з криптобіржами та введення сучасних інструментів трейдингу.

Подальша імплементація проекту є доцільною, оскільки він є актуальним на сьогоднішній день, має високий потенціал масштабування, що допоможе забезпечити рентабельність та розвиток у довгостроковій перспективі.

ВИСНОВКИ

У магістерській дисертації застосовано моделі машинного навчання, що прогнозують цін купівлі/продажу ринку криптовалют, та алгоритм оптимізації стратегії на основі прогнозованих значень за допомогою теорії ігор.

Розроблена програма вирішує проблеми високої волатильності та непередбачуваності досліджуваного ринку, що не дозволяє трейдерам створити обґрунтовану стратегію.

Було застосовано різні методи машинного навчання Random Forest, AdaBoost, Extra Trees, Gradient Boosting та Ensemble Model, що дозволило створити моделі для прогнозування купівлі/продажу криптовалют з різною точністю та стабільністю. Також, було створено алгоритм оптимізації стратегії за допомогою рівноваги Неша та Байєсівських ігор.

У роботі було застосовано такі критерії оцінки якості, як точність (Accuracy) та класифікаційний звіт (Precision, Recall, F1-score) для визначення продуктивності моделей. Крім того, BackTest дозволяє оцінити ефективність стратегії за допомогою фінансових показників Sortino Ratio, Beta, Alpha та Maximum Drawdown, що підтверджують здатність моделей знижувати ризики та покращувати прибутковість.

На прикладі криптовалют Bitcoin та Ethereum було зроблено порівняльний аналіз результатів. Для криптовалюти Bitcoin найкращою виявилася Ensemble Model з точністю 62%, а у випадку Ethereum – AdaBoost з точністю 62%. Оцінка якості сигналів «Buy» та «Sell» вказують на збалансовану ефективність моделі та макро- і зважені значення – на стабільність отриманих результатів. За результатами фінансових метрик BackTest оптимізована стратегія Bitcoin забезпечила доволі високий рівень прибутковості за незначний ризик, а Ethereum – теж високу дохідність та з невеликим ризиком у 15%.

Доведено, що поєднання методів машинного навчання та теорії ігор може значно підвищити ефективність аналізу ринку криптовалют та має практичну цінність для трейдерів.

Подальші дослідження можуть бути спрямовані на розширення моделей для інших фінансових активів, наприклад акцій, або їх вдосконалення під специфічні умови волатильності ринку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What Is Cryptocurrency and How Does It Work? Binance Academy. URL: <https://academy.binance.com/en/articles/what-is-a-cryptocurrency>
2. What Is Blockchain and How Does It Work? Binance Academy. URL: <https://academy.binance.com/en/articles/what-is-blockchain-and-how-does-it-work>
3. Верховна Рада України. (2021). Про віртуальні активи. Закон України № 3637-IX. Офіційний вісник України, 2021, № 79, ст. 5002. URL: <https://zakon.rada.gov.ua/laws/show/2074-20#Text>
4. Digital finance: Council adopts new rules on markets in crypto-assets (MiCA), Council of the EU, Press release, 16 May 2023. URL: <https://www.consilium.europa.eu/en/press/press-releases/2023/05/16/digital-finance-council-adopts-new-rules-on-markets-in-crypto-assets-mica/>
5. Dongyang L., Zequan W. Differentiated Analysis of Cryptocurrencies and Stocks Based on Investor Perspective. Global Economic and Financial Human Resource. 2023. Vol. 21, P. 96-103. URL: <https://drpress.org/ojs/index.php/HBEM/article/view/13671>
6. Setiawan R., Cavaliere L.P.L., Halder S., Koti K. The Concept of the Cryptocurrency and the Downfall of the Banking Sector in Reflecting on the Financial Market. 2021. URL: https://www.researchgate.net/figure/The-comparison-of-Cryptocurrency-Fiat-Currency-Cryptocurrency_tbl2_351871135
7. A Complete Guide to Cryptocurrency Trading for Beginners. (2020, June 22). Binance Academy. URL: <https://academy.binance.com/en/articles/a-complete-guide-to-cryptocurrency-trading-for-beginners>
8. Вакеріна А., Коваль О. Штучний інтелект і машинне навчання у фінансових послугах. Піонерія в науці: збірник наукових праць. 2023. С. 90–92. URL: https://stud.knutd.edu.ua/bitstream/123456789/25866/1/PIONBUG_2023_P090-092.pdf

9. Фаріон В., Гомотюк А., Назар Р., Турчин С. Використання штучного інтелекту для прогнозування фінансових показників. Економічний аналіз. 2024. Том 34. № 2. С. 327–337.
10. Ullah M., Lakhan G. R., Channa A., Gul S. Game Theory and Stock Investment. 2021. Vol. 7, Issue 6. URL: <http://ijdri.com/me/wp-content/uploads/2021/06/5.pdf>
11. D-Central. Deciphering the Game of Strategy in Bitcoin's Network Through Game Theory. 12 January 2024. URL: <https://www.linkedin.com/pulse/deciphering-game-strategy-bitcoins-network-through-theory-ccpce>
12. Random Forest Algorithm in Machine Learning. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>
13. Zhu H., Chen R., Li R. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. Appl. Sci. 2022. Vol. 12, No. 12. Article 5872. URL: <https://www.mdpi.com/2076-3417/12/12/5872>
14. Behzat A., Orhan A., Cemil A. An Extra Tree Regression Model for Discharge Coefficient Prediction: Novel, Practical Applications in the Hydraulic Sector and Future Research Directions. Mathematical Problems in Engineering. 2021, Vol.1, P. 1-19. URL: https://www.researchgate.net/publication/354760259_An_Extra_Tree_Regression_Model_for_Discharge_Coefficient_Prediction_Novel_Practical_Applications_in_the_Hydraulic_Sector_and_Future_Research_Directions
15. Natekin A., Knoll A. Gradient Boosting Machines, A Tutorial. Frontiers in Neurorobotics. 2013. Vol.7. Article 21. P. 1-21. URL: https://www.researchgate.net/publication/259653472_Gradient_Boosting_Machines_A_Tutorial
16. Python for Finance and Algorithmic Trading (2nd edition): Machine Learning, Deep Learning, Time Series Analysis, Risk and Portfolio Management for MetaTrader™5 Live Trading. – Jupyter, 2023.

17. Feng Z. On the Nash Equilibrium in Game Theory: A Learning Report for Optimization Method Class. Specialty in Mathematics 1501, HUST, November 14, 2018. P. 1-29.
18. Барановська, Л. В. Теорія ігор: Курс лекцій. Навчальний посібник. Київ: КПІ ім. Ігоря Сікорського. 2022. 245 с.
19. Kashyap, Dhruv. Earning via Game Theory: Trading and Nash Equilibrium, Effect on the Market's Behaviour. May 4, 2021. URL: <https://medium.com/@frozenflames/trading-and-nash-equilibrium-8621d80d350>
20. Bjorndahl A., Halpern J. Y., Pass R. Bayesian Games with Intentions. Electronic Proceedings in Theoretical Computer Science 215 (Proc. TARK 2015), P. 99-113.
21. Yahoo Finance. Bitcoin to USD Historical. 2024. URL: <https://finance.yahoo.com/quote/BTC-USD/history/>
22. Hayes A. Simple Moving Average (SMA). Reviewed by Charles Potters, Fact checked by Michael Rosenston. 2024. URL: <https://www.investopedia.com/terms/s/sma.asp#toc-limitations-of-simple-moving-average>
23. What is the Moving Standard Deviation (MSD)? 2024. URL: <https://trendspider.com/learning-center/what-is-the-moving-standard-deviation-msd/>
24. Fernando J. Relative Strength Index (RSI) Indicator Explained With Formula. Reviewed by Gordon Scott, Fact checked by Vikki Velasquez. 2024. URL: <https://www.investopedia.com/terms/r/rsi.asp>
25. Mitchell C. What Is the Ichimoku Cloud Technical Analysis Indicator? Reviewed by Khadija Khartit, Fact checked by Melody Kazel. 2024. URL: <https://www.investopedia.com/terms/i/ichimoku-cloud.asp>
26. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., 2019. 856 p.

27. Akshay W. Understanding the Accuracy Score Metric's Limitations in the Data Science Classification Problems. LinkedIn Pulse. 2023. URL: <https://www.linkedin.com/pulse/understanding-accuracy-score-metrics-limitations-data-akshay-w>

28. Івашова А.П., Зайченко О.Ю. Аналіз ринку криптовалют з використанням методів машинного навчання та теорії ігор // Системні науки та інформатика: збірник доповідей III Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 25–29 листопада 2024 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024., 7 стор.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import numpy as np
import pandas as pd
import ta
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier, VotingClassifier
from sklearn.ensemble import AdaBoostClassifier, ExtraTreesClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
import tkinter as tk
from tkinter import filedialog, ttk
from tkcalendar import DateEntry
import matplotlib.dates as mdates
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, classification_report
import yfinance as yf
import random

class FinancialAnalysisApp:
    def __init__(self, root):
        self.df = None
        self.result_data = None
        self.dfc = None
        self.optimized_strategy_df = None

        self.root = root
        self.root.title('Financial Analysis App')

        # Додавання атрибутів для бек-тестування
        self.expected_return_long = 0.1 # приклад значення, можна
змінити
        self.expected_return_short = 0.08 # приклад значення, можна
змінити

```

```

self.risk_long = 0.05 # приклад ризику для довгої позиції
self.risk_short = 0.07 # приклад ризику для короткої позиції

self.create_widgets()

def create_widgets(self):
    # Ticker Symbol Input
    self.ticker_label = tk.Label(self.root, text='Ticker Symbol:')
    self.ticker_label.pack()
    self.ticker_input = tk.Entry(self.root)
    self.ticker_input.pack()

    # Start and End Date Input
    self.start_date_label = tk.Label(self.root, text='Start
Date:')
    self.start_date_label.pack()
    self.start_date_input = DateEntry(self.root,
date_pattern='yyyy-mm-dd')
    self.start_date_input.pack()

    self.end_date_label = tk.Label(self.root, text='End Date:')
    self.end_date_label.pack()
    self.end_date_input = DateEntry(self.root, date_pattern='yyyy-
mm-dd')
    self.end_date_input.pack()

    # Lag
    self.lag_label = tk.Label(self.root, text='Lag:')
    self.lag_label.pack()
    self.lag_input = tk.Entry(self.root)
    self.lag_input.insert(0, '10')
    self.lag_input.pack()

    self.buttons_frame = tk.Frame(self.root)
    self.buttons_frame.pack(pady=(10, 10))

    # Load Data Button
    self.load_button = tk.Button(self.buttons_frame, text='Load
Data', command=self.load_data)

```

```

self.load_button.pack(side=tk.LEFT, padx=(0, 10))

# Load Data Button From File
self.load_button = tk.Button(self.buttons_frame, text='Load
Data File', command=self.load_data_from_file)
self.load_button.pack(side=tk.LEFT, padx=(0, 10))

# Run Analysis Button
self.run_analysis_button = tk.Button(self.buttons_frame,
text='Run Analysis', command=self.run_analysis)
self.run_analysis_button.pack(side=tk.LEFT, padx=(0, 10))

# Optimize Strategy Buttons
self.optimize_strategy_button = tk.Button(self.buttons_frame,
text='Optimize Strategy',
command=self.optimize_strategy)
self.optimize_strategy_button.pack(side=tk.LEFT, padx=(0, 10))

# BackTest Button
self.backtest_button = tk.Button(self.buttons_frame,
text='BackTest', command=self.run_backtest)
self.backtest_button.pack(side=tk.LEFT, padx=(0, 10))

self.result_frame = tk.Frame(self.root, borderwidth=2,
relief='groove')
self.result_frame.pack(side=tk.LEFT, fill=tk.BOTH, padx=10,
pady=10, expand=True)

# Export to Excel
self.export_button = tk.Button(self.buttons_frame,
text='Export to Excel', command=self.export_to_excel)
self.export_button.pack(side=tk.LEFT)

self.result_frame = tk.Frame(self.root, borderwidth=2,
relief='groove')
self.result_frame.pack(side=tk.LEFT, fill=tk.BOTH, padx=10,
pady=10, expand=True)

```

```

        self.result_label = tk.Label(self.result_frame,
text='Results')
        self.result_label.pack(padx=10, pady=(0, 10))

        # Result Text
        self.result_text = tk.Text(self.result_frame, height=10,
width=50)
        self.result_text.pack(padx=10, pady=10, fill=tk.BOTH,
expand=True)

        self.plot_frame = tk.Frame(self.root, borderwidth=2,
relief='groove')
        self.plot_frame.pack(side=tk.LEFT, fill=tk.BOTH, padx=10,
pady=10, expand=True)

        self.plot_label = tk.Label(self.plot_frame, text='Graph')
        self.plot_label.pack(padx=10, pady=(0, 10))

        # Initialize matplotlib figure and canvas
        self.fig, self.ax = plt.subplots(figsize=(15, 6))
        self.canvas = FigureCanvasTkAgg(self.fig,
master=self.plot_frame)
        self.canvas.get_tk_widget().pack(padx=10, pady=10,
fill=tk.BOTH, expand=True)

    def load_data(self):
        ticker = self.ticker_input.get()
        start_date = self.start_date_input.get_date().strftime('%Y-%m-
%d')
        end_date = self.end_date_input.get_date().strftime('%Y-%m-%d')

        if not ticker:
            self.result_text.insert(tk.END, 'Please enter a ticker
symbol.\n')
            return

        if start_date > end_date:
            self.result_text.insert(tk.END, 'Start date cannot be
after end date. Please select valid dates.\n')

```

```

        return

    try:
        df = yf.download(ticker, start=start_date, end=end_date)
        if df.empty:
            self.result_text.insert(tk.END, f'No data found for
ticker: {ticker} in the selected date range.\n')
            return
        df.reset_index(inplace=True)
        df.set_index('Date', inplace=True)
        df.columns = ['open', 'high', 'low', 'close', 'adj_close',
'volume']
        self.df = df
        self.result_text.insert(tk.END, f'Data loaded for ticker:
{ticker} from {start_date} to {end_date}\n')
    except Exception as e:
        self.result_text.insert(tk.END, f'Failed to load data:
{e}\n')
        return

    def load_data_from_file(self):
        file_path = filedialog.askopenfilename(title='Select CSV
File',
                                                filetypes=[('CSV
Files', '*.csv'), ('All Files', '*..*')])

        if file_path:
            try:
                df = pd.read_csv(file_path)
                date_columns = [col for col in df.columns if 'date' in
col.lower() or 'time' in col.lower()]

                if date_columns:
                    df[date_columns[0]] =
pd.to_datetime(df[date_columns[0]], errors='coerce')
                    df.set_index(date_columns[0], inplace=True)
                else:
                    self.result_text.insert(tk.END, 'No valid date
column found in the file.\n')

```

```

        return

        # Renaming columns to standard names if possible
        df.rename(columns=lambda x: x.strip().lower(),
inplace=True)

        column_mapping = {
            'open': 'open',
            'high': 'high',
            'low': 'low',
            'close': 'close',
            'volume': 'volume',
            'adj close': 'adj_close',
            'adjusted close': 'adj_close'
        }
        df.rename(columns=column_mapping, inplace=True)

        required_columns = ['open', 'high', 'low', 'close',
'volume']

        missing_columns = [col for col in required_columns if
col not in df.columns]
        if missing_columns:
            self.result_text.insert(tk.END, f'Missing required
columns: {missing_columns}\n')
            return

        self.df = df
        self.result_text.insert(tk.END, f'Data loaded from:
{file_path}\n')
        except Exception as e:
            self.result_text.insert(tk.END, f'Failed to load data
from file: {e}\n')
        else:
            self.result_text.insert(tk.END, 'No file selected.\n')
            return None

def error_interface(self):
    errors = []

    if self.df is None:

```

```

        errors.append("No data loaded. Please load data from Yahoo
Finance or a file.")

    # Check for valid lag input
    try:
        lag_value = int(self.lag_input.get())
        if lag_value <= 0:
            errors.append("Lag must be a positive integer.")
    except ValueError:
        errors.append("Invalid input for lag. It must be an
integer.")

    if errors:
        for error in errors:
            self.result_text.insert(tk.END, error + '\n')
        return False
    return True

    def ensembling_methods(self, df, lag):
        def feature_engineering(df, lag):
            df_copy = df.copy()
            df_copy["returns"] = df_copy["close"].pct_change(1)
            df_copy["direction"] = np.where(df_copy["returns"] > 0, 1,
0) # Direction: 1 for up, 0 for down
            df_copy["SMA 15"] =
df_copy["close"].rolling(15).mean().shift(lag)
            df_copy["SMA 60"] =
df_copy["close"].rolling(60).mean().shift(lag)
            df_copy["MSD 10"] =
df_copy["returns"].rolling(10).std().shift(lag)
            df_copy["MSD 30"] =
df_copy["returns"].rolling(30).std().shift(lag)
            RSI = ta.momentum.RSIIndicator(df_copy["close"],
window=14, fillna=False)
            df_copy["rsi"] = RSI.rsi().shift(lag)
            IC = ta.trend.IchimokuIndicator(df_copy["high"],
df_copy["low"])
            df_copy["ichimoku_a"] = IC.ichimoku_a().shift(lag)
            df_copy["ichimoku_b"] = IC.ichimoku_b().shift(lag)

```

```

        df_copy["ichimoku_base"] =
IC.ichimoku_base_line().shift(lag)
        df_copy["ichimoku_conversion"] =
IC.ichimoku_conversion_line().shift(lag)
        df_copy.dropna(inplace=True)
        return df_copy

dfc = feature_engineering(df, lag)
X = dfc[["SMA 15", "SMA 60", "MSD 10", "MSD 30", "rsi",
"ichimoku_a", "ichimoku_b", "ichimoku_base",
        "ichimoku_conversion"]]
y = dfc["direction"]

split = int(0.8 * len(dfc))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

sc = StandardScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.transform(X_test)

pca = PCA(n_components=5)
X_train_pca = pca.fit_transform(X_train_sc)
X_test_pca = pca.transform(X_test_sc)

# Voting Classifier Ensemble
rf = RandomForestClassifier(n_estimators=100, random_state=42)
gb = GradientBoostingClassifier(n_estimators=100,
learning_rate=0.1, random_state=42)
ada = AdaBoostClassifier(n_estimators=100, random_state=42)
et = ExtraTreesClassifier(n_estimators=100, random_state=42)

# Using Voting Classifier to create an ensemble
ensemble_model = VotingClassifier(estimators=[('rf', rf),
('gb', gb), ('ada', ada), ('et', et)], voting='soft')
models = [rf, gb, ada, et, ensemble_model]
model_names = ['Random Forest', 'Gradient Boosting',
'AdaBoost', 'Extra Trees', 'Ensemble Model']

```

```

best_model = None
best_accuracy = 0

self.result_text.insert(tk.END, "Model Performance:\n")
for model, name in zip(models, model_names):
    model.fit(X_train_pca, y_train)
    y_pred = model.predict(X_test_pca)
    accuracy = accuracy_score(y_test, y_pred)
    self.result_text.insert(tk.END, f"{name} Accuracy:
{accuracy:.2f}\n")
    if accuracy > best_accuracy:
        best_model = model
        best_accuracy = accuracy

# Generate initial buy/sell signals with the best model
dfc["initial_signal"] =
best_model.predict(pca.transform(sc.transform(X)))
self.dfc = dfc

# Classification Report
y_pred = best_model.predict(X_test_pca)
class_report = classification_report(y_test, y_pred,
target_names=['Sell', 'Buy'])
self.result_text.insert(tk.END, f"\nClassification Report for
Best Model:\n{class_report}\n")

return dfc

def run_analysis(self):
    if not self.error_interface():
        return

    try:
        lag_value = int(self.lag_input.get())
        dfc = self.ensembling_methods(self.df, lag_value)
        self.result_text.insert(tk.END, "Analysis complete. See
the graph for signals.\n")

        # Plot signals

```

```

        self.ax.clear()
        self.ax.plot(dfc.index, dfc["close"], label="Close Price",
alpha=0.5)
        self.ax.scatter(dfc.index[dfc["initial_signal"] == 1],
dfc["close"][dfc["initial_signal"] == 1],
                        color='green', marker='^', label='Buy
Signal')
        self.ax.scatter(dfc.index[dfc["initial_signal"] == 0],
dfc["close"][dfc["initial_signal"] == 0],
                        color='red', marker='v', label='Sell
Signal')

        self.ax.legend()
        self.canvas.draw()

    except Exception as e:
        self.result_text.insert(tk.END, f"Analysis error: {e}\n")

    def optimize_strategy(self):
        if self.dfc is None:
            self.result_text.insert(tk.END, "Please run analysis first
before optimizing strategy.\n")
            return

        try:
            dfc = self.dfc.copy()
            # Optimization using Game Theory - Nash Equilibrium and
Bayesian Games
            buy_signals = dfc[dfc["initial_signal"] == 1].index
            sell_signals = dfc[dfc["initial_signal"] == 0].index

            # Define utility functions
            def utility_nash(buy, sell):
                return (sell - buy) * random.uniform(0.8, 1.2)

            def utility_bayesian(buy, sell, prior_prob):
                return (sell - buy) * prior_prob * random.uniform(0.7,
1.3)

            # Define rules for filtering out suboptimal signals

```

```

optimized_buy_signals = []
optimized_sell_signals = []

# Ensure buy and sell signals alternate correctly
previous_action = None
for date in sorted(buy_signals.union(sell_signals)):
    if date in buy_signals and previous_action != "buy":
        optimized_buy_signals.append(date)
        previous_action = "buy"
    elif date in sell_signals and previous_action ==
"buy":
        optimized_sell_signals.append(date)
        previous_action = "sell"

# Applying Nash Equilibrium to find stable strategies
optimized_strategy = []
for buy_date, sell_date in zip(optimized_buy_signals,
optimized_sell_signals):
    utility_n = utility_nash(dfc["close"].loc[buy_date],
dfc["close"].loc[sell_date])
    utility_b =
utility_bayesian(dfc["close"].loc[buy_date],
dfc["close"].loc[sell_date], prior_prob=0.6)
    if utility_n >= utility_b:
        optimized_strategy.append((buy_date, sell_date,
"Nash"))
    else:
        optimized_strategy.append((buy_date, sell_date,
"Bayesian"))

self.result_text.insert(tk.END, "Combined Optimized
strategy:\n")
combined_buy_signals = []
combined_sell_signals = []
for buy_date, sell_date, strategy_type in
optimized_strategy:
    self.result_text.insert(tk.END, f"Buy on
{buy_date.date()} -> Sell on {sell_date.date()} ({strategy_type})\n")
    combined_buy_signals.append(buy_date)

```

```

        combined_sell_signals.append(sell_date)

        # Plot combined optimized signals
        self.ax.clear()
        self.ax.plot(dfc.index, dfc["close"], label="Close Price",
alpha=0.5)
        self.ax.scatter(combined_buy_signals,
dfc["close"].loc[combined_buy_signals], color='purple', marker='^',
                        label='Combined Optimized Buy Signal')
        self.ax.scatter(combined_sell_signals,
dfc["close"].loc[combined_sell_signals], color='yellow',
                        marker='v', label='Combined Optimized Sell
Signal')

        self.ax.legend()
        self.canvas.draw()

    except Exception as e:
        self.result_text.insert(tk.END, f"Optimization error:
{e}\n")

    def export_to_excel(self):
        file_path =
filedialog.asksaveasfilename(defaultextension=".xlsx",
                                filetypes=[("Excel
Files", "*.xlsx"), ("All Files", "*.*")])
        if file_path:
            try:
                with pd.ExcelWriter(file_path) as writer:
                    self.df.to_excel(writer, sheet_name='Original
Data')

                    if self.optimized_strategy_df is not None:
                        self.optimized_strategy_df.to_excel(writer,
sheet_name='Optimized Strategy')

                self.result_text.insert(tk.END, f>Data exported to
{file_path}\n")
            except Exception as e:
                self.result_text.insert(tk.END, f"Failed to export
data: {e}\n")

```

```

def run_backtest(self):
    if self.df is None:
        self.result_text.insert(tk.END, "Please load data before
running backtest.\n")
        return

    # Використання ціни закриття для бек-тестування
    close_prices = self.df['close']
    self.BackTest(close_prices)

def BackTest(self, serie, annualized_scalar=252 * 8):
    # Check for null values in the input series
    if serie.isnull().any():
        self.result_text.insert(tk.END, "Input series contains
null values and cannot be analyzed.\n")
        return

    # Convert prices to returns if the series represents price
data
    returns = serie.pct_change().dropna()

    # Check if there are enough data points for metrics
calculation
    if len(returns) < 2:
        self.result_text.insert(tk.END, "Not enough data points
for metrics calculation.\n")
        return

    # Download S&P 500 data and calculate daily returns
    sp500 = yf.download("^GSPC", start=returns.index.min(),
end=returns.index.max())["Adj Close"].pct_change()

    # Align the indices of both series to ensure they have the
same length
    returns, sp500 = returns.align(sp500, join='inner')

    # Drop NaN values from both series
    returns = returns.dropna()
    sp500 = sp500.dropna()

```

```

# Additional check to ensure equal lengths after dropping NaNs
if len(returns) != len(sp500):
    min_length = min(len(returns), len(sp500))
    returns = returns.iloc[:min_length]
    sp500 = sp500.iloc[:min_length]

# Check if there are enough data points after alignment
if len(returns) < 2 or len(sp500) < 2:
    self.result_text.insert(tk.END, "Not enough data points
after alignment for correlation calculation.\n")
    return

# Calculate Sortino Ratio
sortino = np.sqrt(annualized_scalar) * returns.mean() /
returns[returns < 0].std()

# Calculate beta using the aligned and length-matched data
beta = np.corrcoef(returns, sp500)[0, 1]

# Calculate alpha
alpha = annualized_scalar * (returns.mean() - beta *
sp500.mean())

# Calculate maximum drawdown
max_drawdown = -np.min((1 + returns).cumprod() - 1)

# Display the metrics
self.result_text.insert(tk.END, f"Sortino Ratio:
{sortino:.3f}\n")
self.result_text.insert(tk.END, f"Beta: {beta:.3f}\n")
self.result_text.insert(tk.END, f"Alpha: {alpha:.3f}\n")
self.result_text.insert(tk.END, f"Maximum Drawdown:
{max_drawdown:.3f}%\n")

if __name__ == '__main__':
    root = tk.Tk()
    app = FinancialAnalysisApp(root)
    root.mainloop()

```