КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Економічний факультет
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

«Розробка програмного забезпечення для прогнозування курсу криптовалют»

студента 4 курсу
спеціальності 051 «Економіка»
ОПП «Економічна кібернетика»
денної форми навчання
Мариненка Дениса Вікторовича

Науковий керівник:
к.е.н., асистент
Наумова Марія Олександрівна

Засвідчую, що у цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань
Студент_____
(підпис)

Роботу допущено до захисту перед ЕК
рішенням кафедри економічної кібернетики
від 12 червня 2023 р., протокол № 17
Завідувач кафедри:
доктор економічних наук, професор
Ляшенко Олена Ігорівна

_____
(підпис)

КИЇВ – 2023

РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 44 с., 34 рис., 33 джерела, 1 додаток.

Ключові слова: прогнозування криптовалюти, машинне навчання, інтелектуальний аналіз даних, Bitcoin, Ethereum, мова програмування Python.

Об’єкт дослідження: інтелектуальний аналіз даних.

Предмет дослідження: створення програмного забезпечення, що проводить аналіз даних з подальшим прогнозуванням та графічним відображенням результатів.

Мета дослідження: реалізація програмного забезпечення для отримання даних з прогнозування курсу криптовалют у вигляді графічного відображення результатів.

Методи дослідження: аналіз предметної області, розгляд існуючих рішень, методів проектування та розробки програмного забезпечення для прогнозування курсу криптовалют, інтелектуальний аналіз даних.

Наукова новизна, теоретична значимість дослідження: дане рішення використовує сучасні технології машинного навчання, в основі якого лежить ідея, що комп'ютерні системи можуть аналізувати дані, виявляти патерни і навчатися на основі цих патернів, щоб здійснювати передбачення або приймати рішення у майбутньому на прикладі прогнозування криптовалют.

Практична цінність: розроблене програмне забезпечення з прогнозування курсу криптовалют допоможе інвесторам прийняти раціональні рішення щодо купівлі, продажу або утримання цифрових активів.

RESUME

Taras Shevchenko National University of Kyiv,

Faculty of Economics, Department of Economic Cybernetics

Key words: cryptocurrency forecasting, machine learning, intellectual data analysis, Bitcoin, Ethereum, Python programming language.

Object of research: data mining.

Subject of research: creation of software that analyzes data with subsequent forecasting and graphical display of results.

The purpose of the study: implementation of software for obtaining data on predicting the exchange rate of cryptocurrencies in the form of a graphical display of the results.

Research methods: subject area Analysis, Analysis of existing solutions, methods of designing and developing software for predicting the cryptocurrency exchange rate.

Scientific novelty, theoretical significance of the study: this solution uses modern machine learning technologies, which is based on the idea that computer systems can analyze data, identify patterns and learn from these patterns in order to make predictions or make decisions in the future using the example of cryptocurrency forecasting.

Practical value: the developed cryptocurrency exchange rate forecasting software will help investors make rational decisions about buying, selling or holding digital assets.

44 pages, 34 pictures 33 sources, 1 append.

ЗМІСТ

ВСТУП

На сьогоднішній день світ фінансів інтенсивно переживає еру криптовалют, що змінює уявлення про традиційні фінансові ринки. Розмірковуючи про інвестування та торгівлю криптовалютами, одним з ключових аспектів, який залучає увагу інвесторів, є прогнозування курсу цих цифрових активів. Прогнозування курсу криптовалют є складною задачею, оскільки ці ринки характеризуються високою волатильністю та швидкими змінами ціни.

У зв'язку з цим, розробка програмного забезпечення для прогнозування курсу криптовалют стає надзвичайно актуальною. Такі програмні продукти можуть надати інвесторам, трейдерам та іншим учасникам ринку цінну аналітичну інформацію та інструменти для прийняття обґрунтованих рішень щодо купівлі, продажу або утримання криптовалют.

Об'єктом дослідження є інтелектуальний аналіз даних.

Предмет дослідження – створення програмного забезпечення, що проводить аналіз даних з подальшим прогнозуванням та графічним відображенням результатів.

Мета даної роботи – реалізація програмного забезпечення для отримання даних з прогнозування курсу криптовалют у вигляді графічного відображення результатів.

Методи дослідження: аналіз предметної області, розгляд існуючих рішень, методів проектування та розробки програмного забезпечення для прогнозування курсу криптовалют, інтелектуальний аналіз даних.

Для досягнення зазначеної мети було поставлено такі завдання:

1. проведення аналізу наукових та літературних джерел з тематики дослідження;
2. розгляд сучасних існуючих рішень, що використовуються в галузі прогнозування курсу криптовалют;
3. проектування етапів розробки додатку;
4. проведення аналізу сучасних інструментів та програмних засобів реалізації програмного забезпечення;

5. написання програмного коду для розв’язування поставленого завдання;
6. описання архітектуру та функціоналу додатку;
7. проведення функціонального тестування розробленої системи;

Наукова новизна полягає у тому, що розроблене програмне забезпечення використовує сучасні технології машинного навчання, в основі якого лежить ідея, що комп'ютерні системи можуть аналізувати дані, виявляти патерни і навчатися на основі цих патернів, щоб здійснювати передбачення або приймати рішення у майбутньому на прикладі прогнозування криптовалют.

Практична значущість результатів дослідження може бути застосовано для допомоги інвесторам прийняти раціональні рішення щодо купівлі, продажу або утримання цифрових активів, отримавши аналітичну інформацію з візуальним відображенням у вигляді графіків. Таким чином компанія або людина, яка хоче торгувати на ринку криптовалют, це рішення заощадить час та кошти на витратах аналітичних послуг.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Основні положення криптовалюти

Криптовалюта - це цифровий або віртуальний актив, який використовує криптографію для безпечного проведення фінансових операцій, контролю створення нових одиниць і підтвердження транзакцій. Вона базується на технології блокчейн, що забезпечує безпеку, децентралізацію і недубльованість транзакцій [1].

Криптовалюти не потребують централізованого посередництва банків або урядів, оскільки транзакції підтверджуються мережею користувачів. Вони також можуть забезпечувати анонімність та глобальну доступність. Кількість криптовалют зазвичай обмежена, що може впливати на їхню вартість.

Одиниця криптовалюти - це унікальний код, який формується шляхом виконання складних обчислень на комп'ютері за допомогою математичних алгоритмів [34].

Суть криптовалюти полягає в тому, що, хоч вона електронна, її не можна скопіювати як звичайний файл. І начебто дуже зручна і надійна для миттєвих фінансових розрахунків по світу. Колись, можливо, вона замінить звичайні гроші. І ціни на криптовалюти формуватимуться потужностями економік країн чи активами крупних компаній [1].

Проте у наших непередбачуваних реаліях не віриться у надійність криптовалюти й того ж біткойну. Важно зазначити, що фактично за математичні розрахунки коштує значну кількість грошей. Ціна виникає внаслідок складних обчислень, які вимагають великих обчислювальних ресурсів та спеціалізованого обладнання. Незважаючи на обмежену кількість криптовалют (наприклад, біткоїн обмежений 21 мільйоном монет), існує безліч різних криптовалют, і біткоїн є лише першим з них. Важливо відзначити, що біткоїн не є найкращим з точки зору зручності, швидкості та функціональності, оскільки на сьогоднішній

день розроблено багато інших криптовалют, які можуть мати покращені характеристики і функціонал.

Згідно з експертами, ціна на біткоїн висока завдяки хайпу, існує думка, що це надута бульбашка, яка може стати найбільшою в історії людства. Прихильники біткоїна вірять в нього настільки ж, як колись вірили в тюльпани під час бульбашки на ринку в Голландії у XVII столітті. Так само, як інвестори вірили в інтернет-компанії під час "дотком буму" в 2000 році, аргументуючи це майбутнім розвитком світової економіки, і що їхні акції треба купувати за будь-якою ціною. Бо "буде тільки дорожчати!" Як усім відомо, більшість тих інтернет-компаній зазнали краху, хоч ера інтернет-компаній все-таки настає [1].

Її основні переваги у тому, що вона не прив’язана до жодної країни і уряди не можуть на них впливати, не залежить від банків (можна розраховуватись в інтернеті без посередників), це анонімно та конфіденційно, безпечно і стійке до взлому, тут блискавичні та швидкі операції транзакції.

Але є й такі недоліки: висока волатильність і швидкі зміни ціни, поки що небагато гравців пропонують криптовалюту як засіб платежу, легше вести кримінальний бізнес, також не обов’язково повністю анонімно, якщо власник не подбав про це [2].

Основні риси криптовалюти:

1) Децентралізація: Криптовалюти працюють на основі технології блокчейн, яка дозволяє уникнути централізованого контролю та посередництва банків або урядів. Замість цього, транзакції підтверджуються мережею користувачів, що забезпечує більшу гнучкість і безпеку.

2) Криптографічна безпека: Криптовалюти використовують криптографію для захисту транзакцій та контролю створення нових одиниць. Це означає, що транзакції є безпечними і неможливими до фальсифікації.

3) Анонімність: В деяких криптовалютах, використання псевдонімів та шифрування дозволяє користувачам зберігати конфіденційність та захищати особисті дані.

4) Обмежене постачання: Більшість криптовалют мають обмежену

загальну кількість одиниць, яка може бути створена. Це створює штучний обмежений ресурс, що може впливати на ціну та вартість криптовалюти.

5) Глобальна доступність: Криптовалюти можуть бути переведені або отримані будь-де в світі без обмежень часу або географії. Вони не залежать від банків або традиційних фінансових інституцій, що робить їх доступними для будь-якого користувача з Інтернетом. Це відкриває нові можливості для людей, які не мають доступу до традиційних банківських послуг, або живуть у регіонах з обмеженим фінансовим доступом.

6) Відкритість та прозорість: Багато криптовалют працюють на відкритих блокчейн-платформах, що означає, що всі транзакції і дані є публічно доступними. Це забезпечує прозорість у системі, дозволяючи користувачам перевіряти та переглядати транзакції.

7) Можливість дроблення: Багато криптовалют можуть бути поділені на менші одиниці, що дозволяє здійснювати мікроплатежі або забезпечує більшу гнучкість у використанні.

8) Міжнародні транзакції: Криптовалюти можуть бути використані для здійснення глобальних транзакцій без необхідності обміну валют та впливу валютних курсів. Це спрощує міжнародну торгівлю та забезпечує швидкість операцій.

9) Інноваційність та потенціал розвитку: Криптовалюти представляють нову форму фінансових активів, що відкриває широкі можливості для інновацій у фінансовій та технологічній галузях. Вони стимулюють розвиток нових додатків, технологій та екосистем, які можуть змінити спосіб функціонування фінансової системи.

10) Ризики та волатильність: Варто зазначити, що криптовалюти також мають свої ризики та волатильність. Ціни криптовалют можуть швидко змінюватися, інвестори піддаються ризику втрати коштів, а також можливості шахрайства та кібератак [2].

1.2. Принцип роботи криптовалют

Принцип роботи криптовалют базується на технології блокчейн і включає такі основні етапи:

I. Створення нових блоків: Криптовалюти використовують механізм, відомий як "доказ роботи" (Proof of Work) або інші консенсус-протоколи, для створення нових блоків. Учасники мережі, які називаються "майнерами", вирішують складну обчислювальну задачу, і перший, хто вирішить її, отримує право створити новий блок.

II. Підтвердження транзакцій: Коли новий блок створюється, він містить набір транзакцій, які були здійснені між користувачами криптовалюти. Ці транзакції перевіряються і підтверджуються мережею, щоб переконатися в їхній правомірності та недублюваності.

III. Розповсюдження блоку: Після підтвердження, новий блок розповсюджується по всій мережі, і кожен учасник оновлює свою копію блокчейну.

IV. Забезпечення безпеки: Криптовалюти використовують криптографічні методи для захисту транзакцій та забезпечення безпеки мережі. Кожен блок містить хеш-функцію, яка унікально ідентифікує блок і посилається на попередній блок у послідовності. Це створює ланцюжок блоків, відомий як блокчейн, який надійно захищає дані і забезпечує недублюваність транзакцій.

V. Децентралізація: Криптовалюти працюють на основі децентралізованої мережі, що означає, що немає центрального органу або влади, яка контролює всі транзакції. Замість цього, вони базуються на спільному згоді і участі учасників мережі, які підтверджують та підтримують правильну роботу системи.

VI. Криптовалютний гаманець: Користувачі криптовалют мають свої особисті гаманці, які зберігають їхні криптовалютні ключі. Ключі використовуються для підпису транзакцій і доступу до власних активів. Гаманці можуть бути онлайн, офлайн або апаратними, залежно від рівня безпеки, який користувач бажає мати.

Це загальний принцип роботи криптовалют, проте кожна конкретна криптовалюта може мати свої особливості та деталі реалізації.

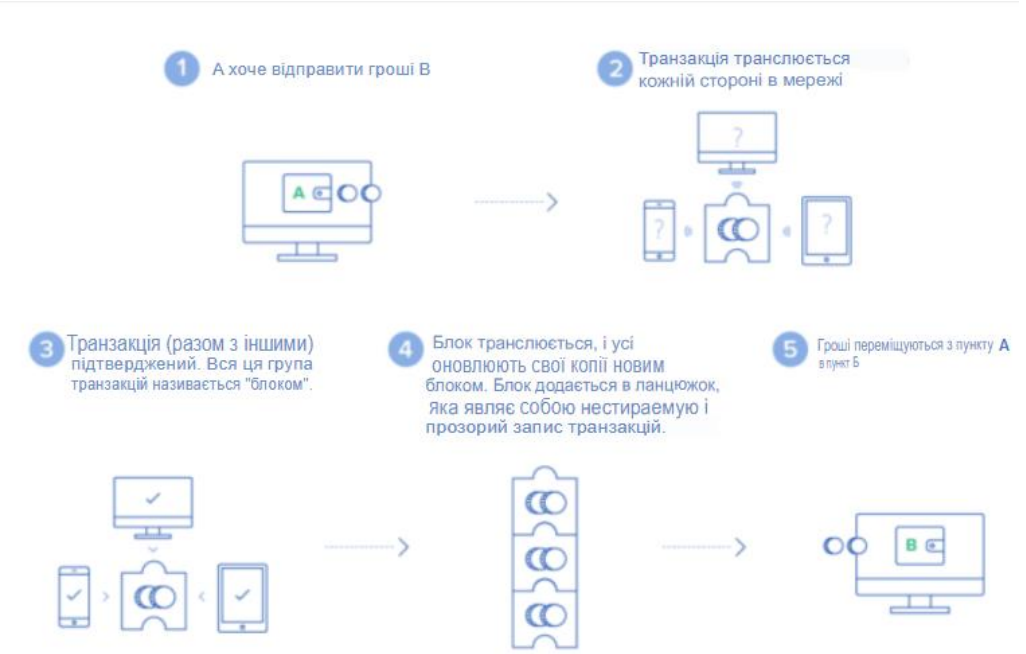

Рис. 1.1. Принцип роботи блокчейну

1.3. Основні види криптовалют

Існує велика кількість криптовалют, і кожна з них має свої унікальні особливості.

Bitcoin (BTC): Bitcoin є першою і найпопулярнішою криптовалютою. Він був запущений у 2009 році і пропонує децентралізовану систему електронних платежів, що базується на технології блокчейн.

Ethereum (ETH): Ethereum є платформою для розробки децентралізованих додатків та смарт-контрактів. Він має власну криптовалюту під назвою Ether, яка використовується для здійснення транзакцій на платформі Ethereum.

Ripple (XRP): Ripple є протоколом для міжбанківських платежів та переказу грошей. Криптовалюта XRP використовується як мостик між різними валютами для швидкого та ефективного переказу коштів.

Litecoin (LTC): Litecoin є відгалуженням від Bitcoin і пропонує швидші часи блоків та більш ефективний механізм видобутку. Він використовує алгоритм Scrypt замість SHA-256, який використовується Bitcoin.

Bitcoin Cash (BCH): Bitcoin Cash є відгалуженням від оригінального Bitcoin, яке було створене з метою поліпшення масштабованості та швидкості транзакцій. Він має більший розмір блоків, що дозволяє обробляти більше транзакцій за один блок.

Binance Coin – це нативна криптовалюта біржі Binance. Вона використовується для оплати комісій на біржі, доступу до ексклюзивних функцій та послуг, а також як форма оплати за товари та послуги. За останні роки Binance Coin значно зросла в ціні і вважається однією з найпопулярніших криптовалютів.

Binance USD – це стейблкоїн, випущений Binance та прив'язаний до долара США. Він забезпечує стабільну інвестиційну можливість на високоволатильному ринку та широко використовується на біржі Binance.

Dogecoin - це криптовалюта, яка була створена на основі популярного інтернет-мему, що зображує сміючого собаку породи ши-тцу з назвою Doge. Хоча Dogecoin спочатку був створений як жартівлива криптовалюта. Незважаючи на своє походження, Dogecoin увійшов до топ-10 криптовалюту за ринковою капіталізацією.

Також відомо, що Ілон Маск, американський підприємець та інвестор, проявляє активний інтерес до криптовалют і має великий вплив на їх ринок своїми заявами та діями. Він часто згадує про криптовалюту Dogecoin у своїх твіттер-публікаціях, що призводить до значних коливань ціни цієї криптовалюти. Ілон Маск також висловлював підтримку для ідеї використання криптовалют, зокрема Bitcoin, у світі фінансів та трансакцій. Його вплив на криптовалютні ринки став предметом обговорення та контроверсій [18].

1.4. Методи прогнозування

Прогнозування курсу криптовалют є складним завданням, оскільки ціни криптовалют піддаються значним коливанням і можуть бути сильно впливані різноманітними факторами. Існує кілька методів і підходів до прогнозування курсу криптовалют, і ось декілька з них:

1) Аналіз технічних показників: Цей метод використовує історичні дані цін

криптовалют та технічні показники, такі як графіки цін, обсяг торгів, індикатори технічного аналізу (наприклад, середня ковзна, RSI, MACD і т.д.), для визначення тенденцій та патернів, які можуть допомогти у прогнозуванні майбутньої ціни.

2) Фундаментальний аналіз: Цей підхід полягає в оцінці фундаментальних факторів, таких як новини, події, законодавство, розвиток технологій, партнерства та інші фактори, що можуть впливати на ціну криптовалют. Аналізуючи ці фактори, трейдери та інвестори можуть намагатися прогнозувати, як вони впливатимуть на курс криптовалюти.

3) Моделі машинного навчання: Використання моделей машинного навчання для прогнозування курсу криптовалют стає все популярнішим. Ці моделі використовуються для аналізу великого обсягу даних і виявлення складних залежностей та патернів, які можуть допомогти у прогнозуванні майбутнього руху цін.

4) Соціальний аналіз: Цей метод використовує аналіз соціальних мереж, форумів та інших джерел інформації, щоб виявити настрої та думки громадськості щодо криптовалют. Він базується на припущенні, що громадська думка може впливати на ціну криптовалют.

Важливо зауважити, що жоден з цих методів не може гарантувати точних прогнозів. Ринок криптовалют вкрай непередбачуваний і піддається впливу багатьох факторів, включаючи новини, регуляторні зміни, поведінку інвесторів та інші фактори. Тому, при прогнозуванні курсу криптовалют, рекомендується використовувати комбінацію різних методів та бути свідомим ризиків, пов'язаних з такими прогнозами.

1.5. Огляд існуючих рішень

Walletinvestor

На даному веб-сайті компанія надає послуги прогнозування курсу криптовалют на короткостроковий або довгостроковий період. Методологія

їхніх прогнозів є комерційною і не розголошується. Нижче зображення ілюструє дизайн сайту і його вигляд (рис. 1.2 і 1.3) [5].

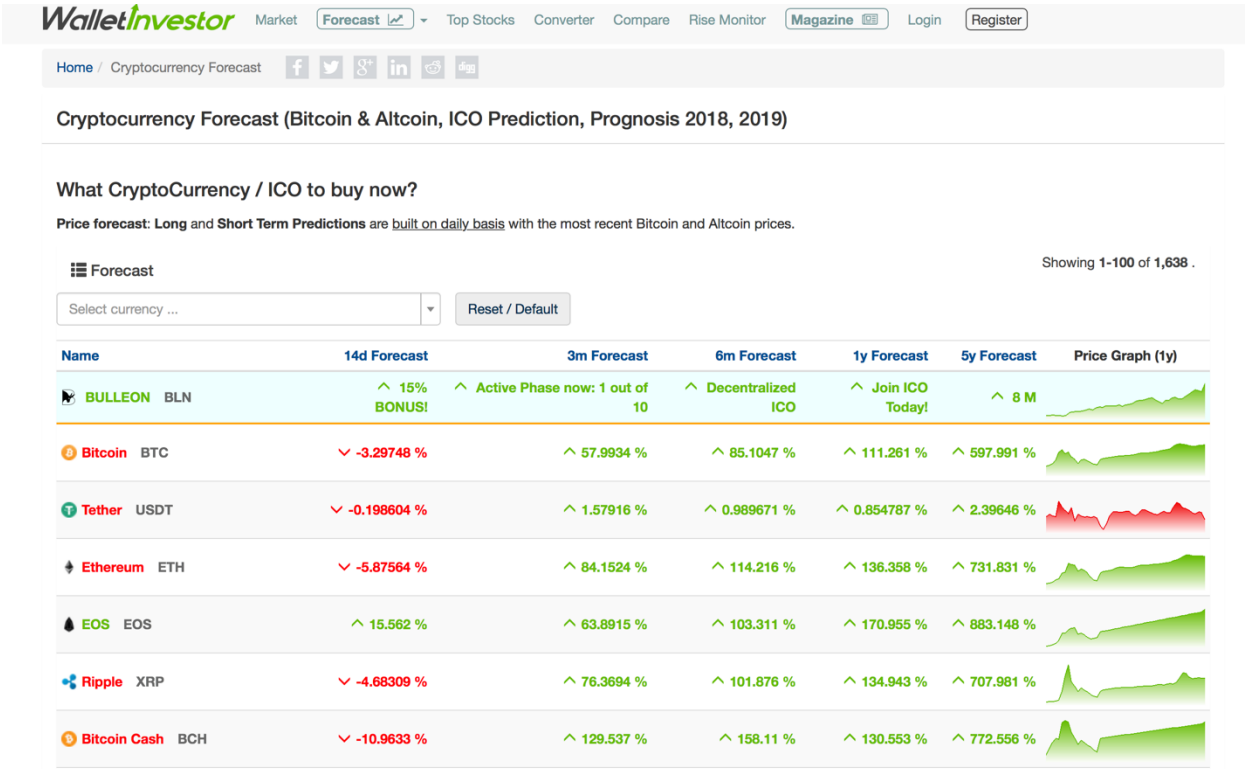

Рис.1.2. Вигляд сайту с прогнозами Walletinvestor

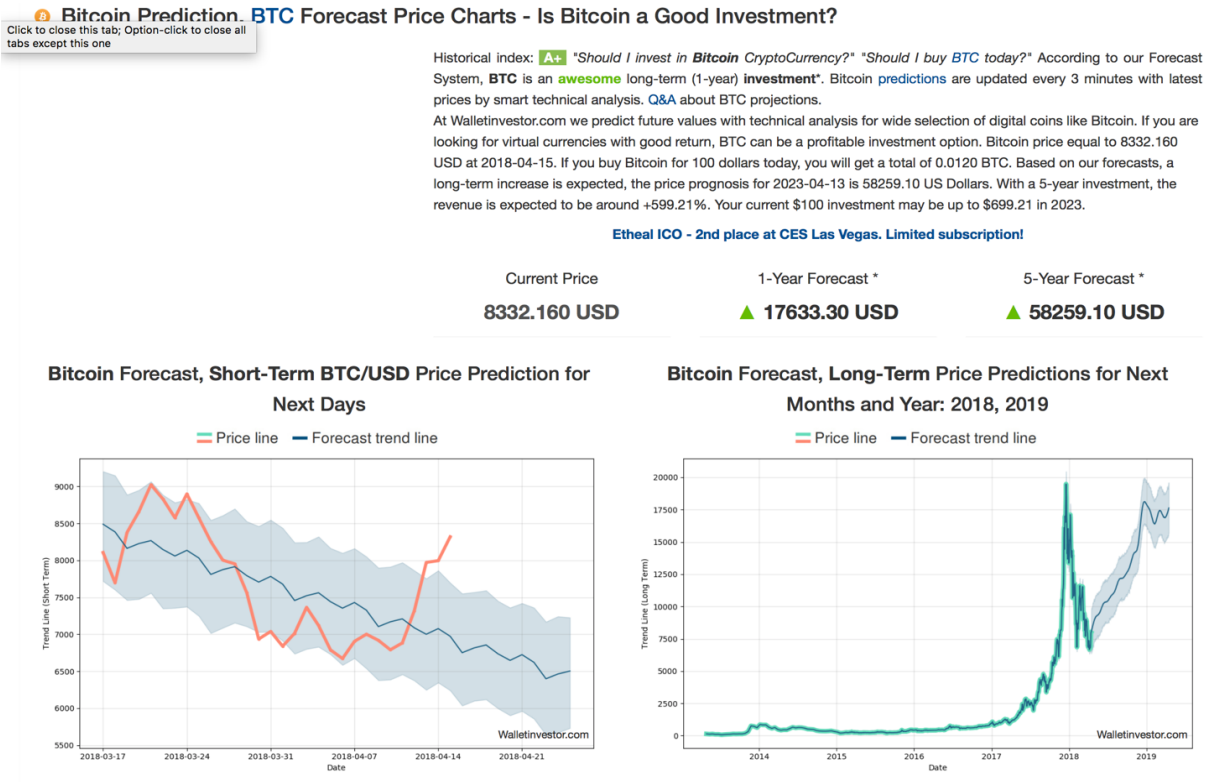

Рис. 1.3. Вигляд сайту Walletinvestor з прогнозами щодо Bitcoin

Belinvestor

а даному веб-сайті компанія надає послуги прогнозування курсу криптовалют з використанням конфіденційних комерційних методів. Зображення, які представлені на сайті, демонструють його дизайн та зовнішній вигляд (рис. 1.4 та 1.5) [6].

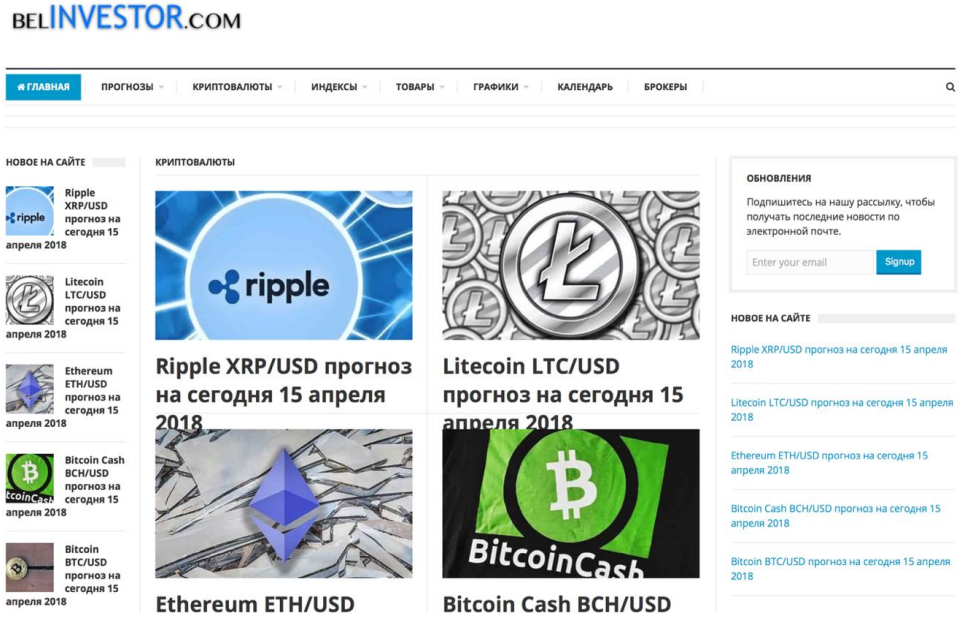

Рис. 1.4. Вигляд сайту с прогнозами Belinvestor

Рис.1.5. Вигляд сайту Belinvestor с прогнозами щодо Bitcoin

NeuroShell.

NeuroShell Day Trader - це спеціалізований нейропакет, розроблений для прогнозування фінансових ринків. Розробники звернули особливу увагу на простоту використання та доступний інтерфейс, щоб користувачам не потрібно було мати програмувальні навички для роботи з нейронними мережами. Однією з переваг NeuroShell Day Trader є можливість оптимізації з використанням генетичних алгоритмів, що дозволяє значно економити час, який раніше витрачався на підбір параметрів та аналіз входів нейронної мережі.

NeuroShell Day Trader фокусується на розробці торгової системи, яка може використовувати як індикатори, так і прогнозовані значення, отримані від нейронних мереж. Процес побудови нейронних мереж досить простий, але деякі ключові етапи є конфіденційними. Основна архітектура, використовувана NeuroShell Day Trader, - це багатошарова штучна нейронна мережа (рис. 1.6 та 1.7) [7].

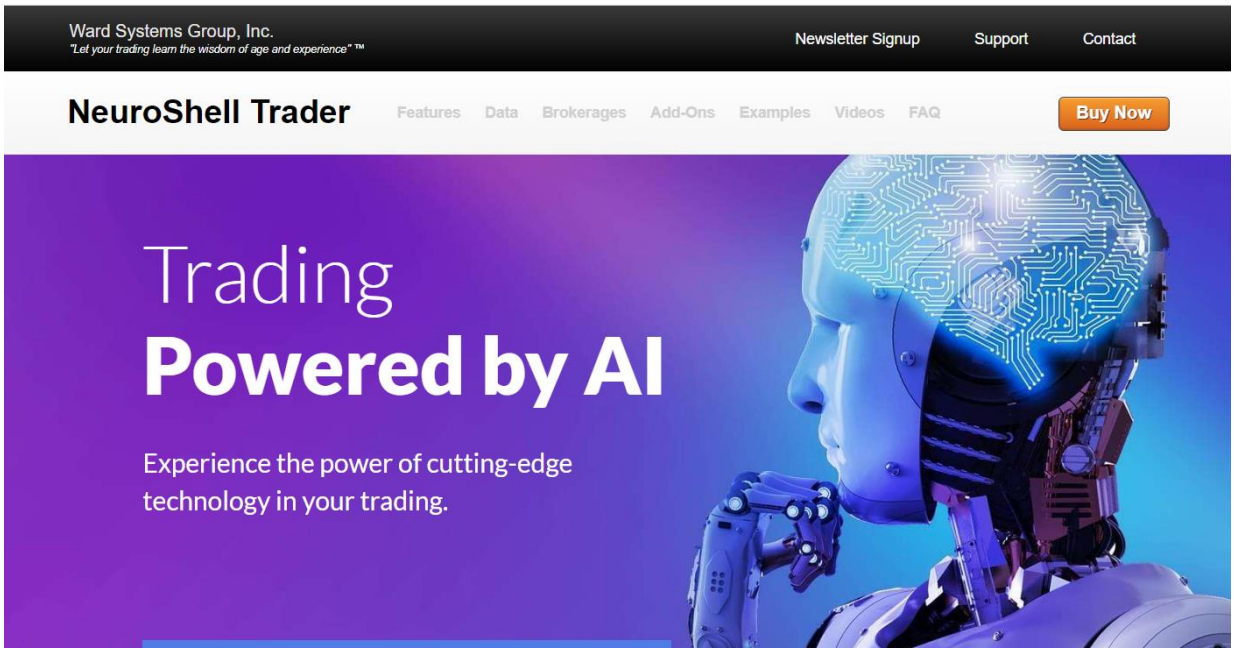

Рис. 1.6. Головна сторінка NeuroShell

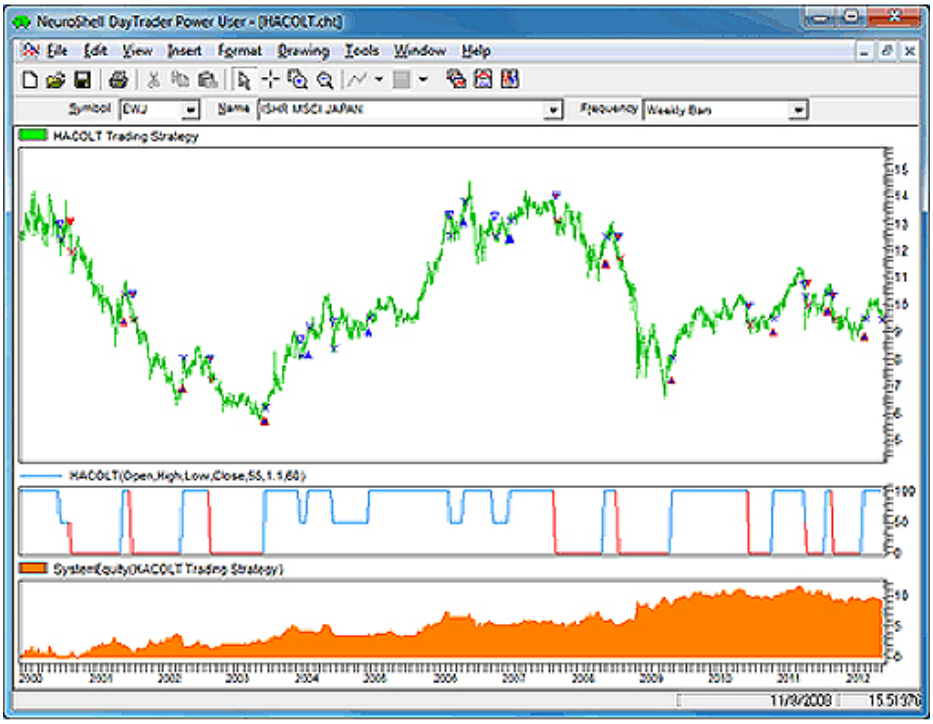

Рис. 1.7. Прогнозування курсу криптовалют в NeuroShell Day Trader

Trader

- Даний веб-додаток пропонує передбачення змін курсів валют на валютній біржі. Для аналізу даних використовуються попередні результати торгів у вигляді часового ряду, який містить інформацію про максимальну та мінімальну ціну, ціну закриття та обсяг угод за кожен день. У системі використовуються наступні алгоритми для аналізу цих даних:

- MACD (Moving Average Convergence Divergence) - це гістограми для виявлення змін в трендах та потенційних сигналів для купівлі або продажу.

- RSI (Relative Strength Index), OBV (On-Balance Volume), Williams R%, CandleSticks, Point & Figure. Ці індикатори є популярними і надають різноманітну інформацію про потенційні перекуплені або перепродані ринки, обсяги угод, цінові паттерни та інші показники. В системі користувач також може створювати власні формули для аналізу даних. До переваг системи також належить можливість застосовувати індикатор до вже побудованого індикатору. Наприклад, для побудови MACD-гістограми можна обчислити ковзне середнє для різниці двох ковзних середніх.

- Лінійне ковзне середнє (Simple Moving Average, SMA): це середнє значення цін або обсягів за певний період, де кожне значення має однакову вагу. Експоненціальне ковзне середнє (Exponential Moving Average, EMA) - це середнє значення, яке надає більшу вагу останнім даним, зменшуючи вагу старіших даних за допомогою експоненційної функції. Ковзне середнє з вагами (Weighted Moving Average, WMA - це середнє значення, де кожне значення має вагу, що залежить від його позиції в періоді. Зазвичай, найновіші значення мають більшу вагу [8].

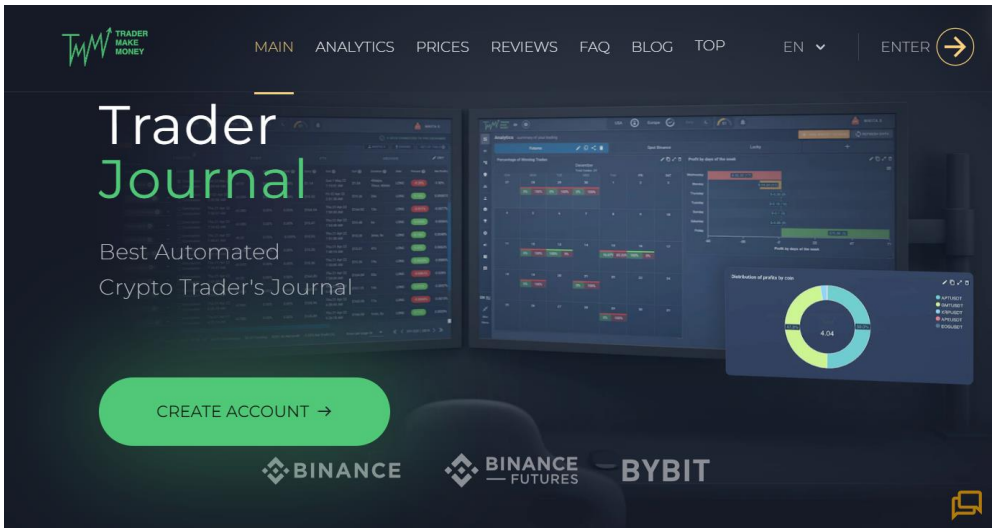

Рис.1.8. Головна сторінка Trader

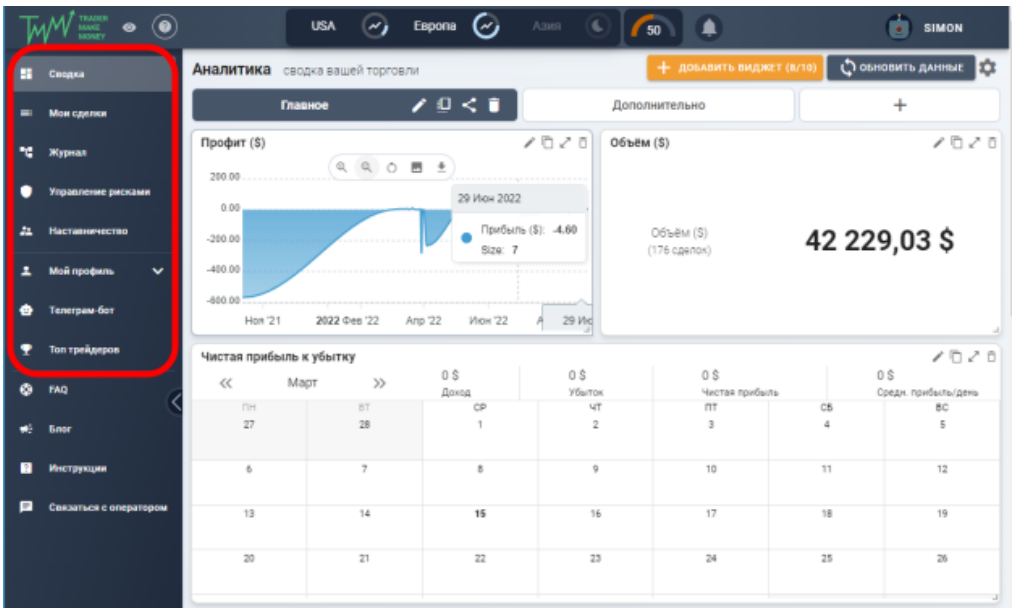

Рис. 1.9. Вигляд інтерфейсу після реєстрації акаунту

Висновки до розділу 1

Отже, під час аналізу переметної області були визначені основні поняття та положення криптовалюти, її принцип роботи, було розглянуто основні методи прогнозування та існуючі рішення.

Прогнозування курсу криптовалют є складним завданням, оскільки ціни криптовалют піддаються значним коливанням і можуть бути сильно впливані різноманітними факторами. Існують такі рішення як: Walletinvestor, Belinvestor, NeuroShell Day Trader, Trader тощо.

Головні мінуси існуючих рішень – зазвичай всі веб-додатки платні, а безплатні – не призначені для довгострокового прогнозування.

РОЗДІЛ 2
АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Мова програмування Python

Python - це мова програмування загального призначення, яка є високорівневою та інтерпретованою. Вона була розроблена у 1991 році з акцентом на спрощення синтаксису та покращення читабельності коду. Python є об'єктно-орієнтованою мовою програмування, що дозволяє створювати об'єкти, включаючи класи, спадкування та поліморфізм. Вона також підтримує багато інших парадигм програмування, включаючи функціональне програмування та структурне програмування. Python став популярним завдяки своїй простоті, зручності використання та багатому екосистемі бібліотек та фреймворків, що розширюють його можливості.

Python є однією з найпопулярніших мов програмування, завдяки своїй простоті, ефективності та широким можливостям в різних сферах.

Основні риси мови Python:

1) Простота використання: Python має простий та зрозумілий синтаксис, що дозволяє швидко вчитися та розробляти програми. Вона підтримує динамічну типізацію, що означає, що не потрібно оголошувати типи змінних перед використанням.

2) Розширюваність: Python має велику кількість стандартних бібліотек, які надають різноманітні функції та інструменти для роботи зі звичайними завданнями програмування. Крім того, Python також має активне спільноту, яка розробляє та підтримує сторонні бібліотеки для різних потреб.

3) Портативність: Python може працювати на різних операційних системах, таких як Windows, macOS та Linux. Це дозволяє розробникам створювати програми, які можуть бути легко перенесені з однієї платформи на іншу.

4) Інтерпретованість: Python є інтерпретованою мовою, що означає, що програмний код виконується рядок за рядком без необхідності компіляції. Це полегшує тестування та розробку, але може знизити швидкість виконання

програми порівняно з компільованими мовами.

5) Широке застосування: Python використовується у багатьох областях, таких як веб-розробка, наукові обчислення, штучний інтелект, аналіз даних, автоматизація, ігри та багато іншого. Вона надає різноманітні інструменти і бібліотеки для вирішення різних завдань.

6) Спрощена синтаксична структура: Python використовує відступи (пробіли або табуляцію) для оформлення блоків коду, що полегшує читання і структурування програми. Це також сприяє читабельності коду та зменшенню кількості помилок [10].

2.2. Бібліотека pandas

Pandas - це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та обробки даних із відкритим кодом, створений на основі мови програмування Python. Воно надає структури даних та функції, які спрощують маніпулювання табличними даними, обробку часових рядів, роботу зі структурованими даними та багато іншого.

Ця бібліотека допомагає завантажувати фрейм даних у форматі 2D-масиву та має кілька функцій для виконання завдань аналізу за один раз.

Основні риси та функціональність pandas включають:

- DataFrame: Основною структурою даних в pandas є DataFrame, який представляє собою двовимірну таблицю з індексами та стовпцями. DataFrame дозволяє легко виконувати операції над даними, такі як фільтрація, сортування, групування, об'єднання тощо.
- Статистичні функції: pandas надає широкий набір статистичних функцій, які дозволяють виконувати обчислення статистичних показників, таких як середнє значення, медіана, стандартне відхилення та інші.
- Часові ряди: pandas має підтримку для обробки та аналізу часових рядів. Вона дозволяє працювати з датами, виконувати ресемплінг, інтерполяцію та інші операції, що пов'язані з часовими даними.
- Введення/виведення даних: pandas надає функції для завантаження та

збереження даних у різних форматах, таких як CSV, Excel, SQL, JSON, HDF5 та інші.

- Обробка пропущених значень: pandas надає інструменти для виявлення та обробки пропущених значень у даних, такі як заповнення пропусків, вилучення рядків з пропущеними значеннями та інші операції.
- Об'єднання та злиття даних: pandas дозволяє об'єднувати та зливати дані з різних джерел, виконувати злиття на основі спільних стовпців або індексів [11].

2.3. Бібліотека NumPy

NumPy - це бібліотека для мови Python, яка надає підтримку для маніпулювання великими багатовимірними масивами і матрицями. Вона також надає широкий набір математичних функцій для виконання операцій з цими масивами. Основним об'єктом у NumPy є багатовимірний масив однорідних елементів, які зазвичай є числами та мають однаковий тип даних.

Найбільш важливі атрибути об'єктів ndarray:

ndarray.ndim: Це атрибут, який показує кількість вимірів або осей у масиві. Він використовується для визначення розміру масиву та доступу до його елементів.

ndarray.shape: Це атрибут, що містить розміри масиву або його форму. Він представляє собою кортеж натуральних чисел, які показують довжину масиву по кожній з його осей. Наприклад, для матриці з n рядків та m стовпців, shape буде (n, m). Кількість елементів у кортежі shape дорівнює значенню атрибуту ndarray.ndim.

ndarray.size: Це атрибут, який вказує загальну кількість елементів у масиві. Він обчислюється як добуток довжини по кожній з осей, вказаних у ndarray.shape. Це число вказує на загальний обсяг даних, що займаються масивом.

ndarray.dtype: Це атрибут, що описує тип елементів у масиві. Він може бути визначений за допомогою стандартних типів даних Python, таких як int, float, або використовувати спеціальні типи даних NumPy. ndarray.dtype вказує на тип

даних, що використовується для зберігання значень у масиві.

ndarray.itemsize: Це атрибут, який показує розмір (кількість байтів) кожного елемента у масиві. Він використовується для визначення обсягу пам'яті, який займає масив в загальному.

ndarray.data: Це атрибут, який містить фактичні елементи масиву у вигляді буфера. Він зазвичай не використовується безпосередньо, оскільки доступ до елементів масиву зазвичай здійснюється за допомогою індексів.

У NumPy є різноманітні методи для створення масивів, і один з простих способів - використання функції numpy.array(). Ця функція дозволяє створити масив звичайних списків або кортежів Python. Важливо запам'ятати, що array є функцією, яка створює об'єкт типу ndarray [12].

2.4. Бібліотека Matplotlib

Matplotlib є потужною бібліотекою для створення різноманітних візуалізацій на Python, включаючи статичні, анімовані та інтерактивні. Вона спрощує створення простих графіків та візуалізацій і надає можливості для реалізації більш складних завдань. Завдяки Matplotlib навіть складні візуалізації стають можливими та доступними. Вона дозволяє наступне:

- Створювати якісні сюжети для публікації .
- Створювати інтерактивні фігури, які можна масштабувати, панорамувати, оновлювати.
- Налаштувати візуальний стиль і макет .
- Експортувати у багато форматів файлів .
- Вставляти в JupyterLab і графічний інтерфейс користувача .
- Використовувати багатий набір сторонніх пакетів, побудованих на Matplotlib [13].

Нижче представлено приклад застосування даної бібліотеки.

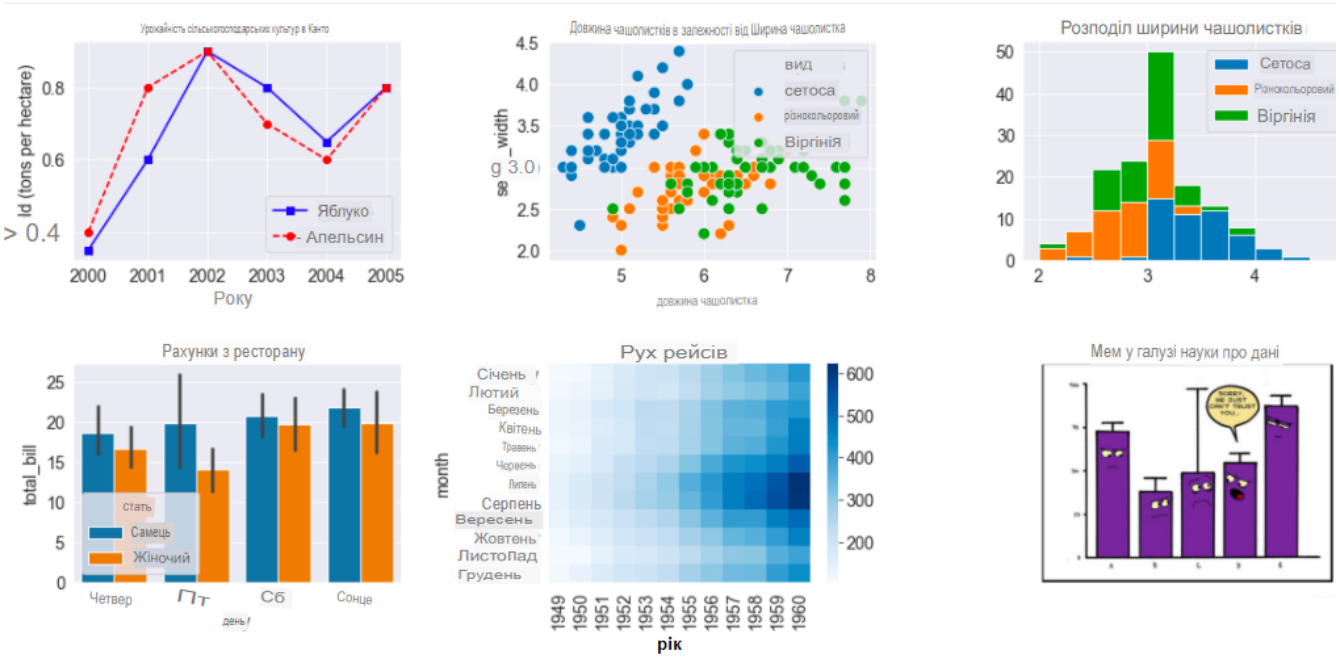

Рис. 2.1. Matplotlib: Візуалізація за допомогою Python

2.5. Бібліотека Seaborn

Seaborn є бібліотекою візуалізації даних для Python, яка базується на Matplotlib. Вона надає зручний інтерфейс для створення привабливих та інформативних статистичних графіків. Seaborn розширює можливості Matplotlib, додаючи нові типи графіків, стилізацію та покращену роботу з даними. Завдяки Seaborn візуалізація даних стає більш простою та естетично задовольняючою [21].

Для короткого ознайомлення з ідеями бібліотеки ви можете прочитати вступні примітки або статтю . Відвідайте сторінку встановлення , щоб дізнатися, як завантажити пакет і почати з ним. Ви можете переглянути галерею прикладів, щоб побачити деякі з речей, які ви можете зробити з seaborn, а потім перегляньте навчальні посібники або посилання на API , щоб дізнатися, як це зробити.

Щоб переглянути код або повідомити про помилку, відвідайте репозиторій GitHub . Загальні питання підтримки найбільш актуальні на stackoverflow , який має спеціальний канал для seaborn [14].

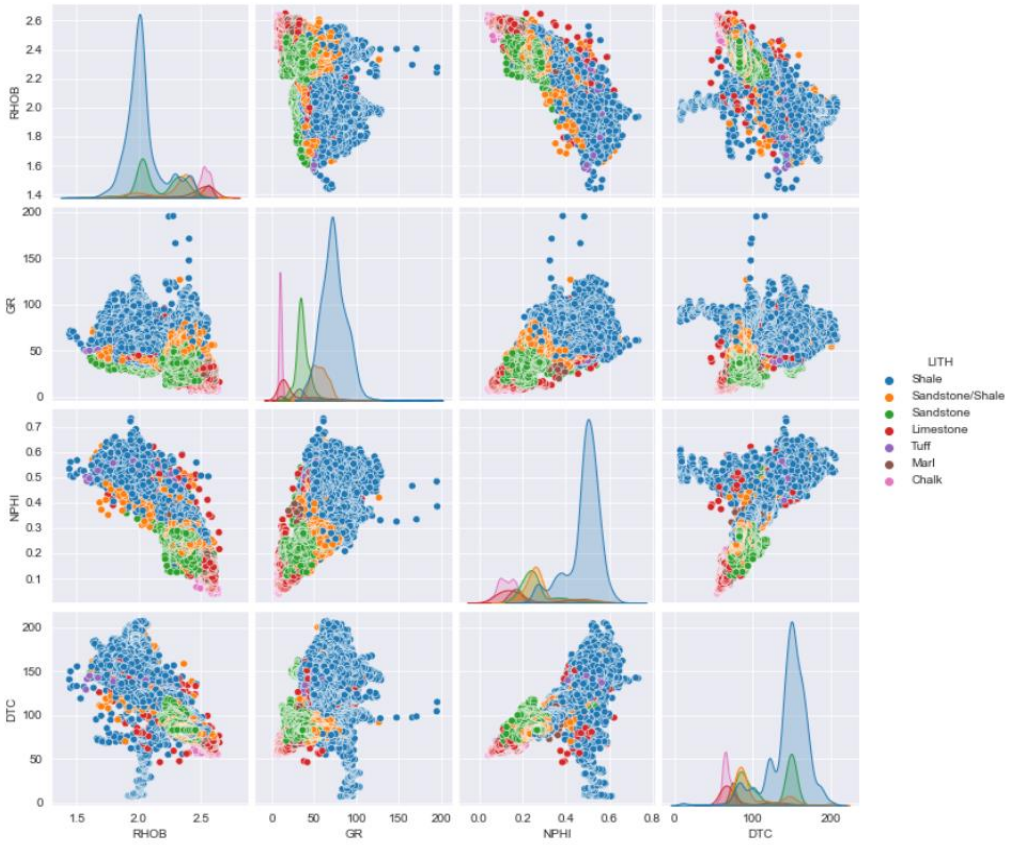

Рис.2.2. Seaborn: Візуалізація за допомогою Python

2.6. Бібліотека Sklearn

Scikit-learn — це бібліотека аналізу даних із відкритим кодом і золотий стандарт машинного навчання (ML) в екосистемі Python [20].

Основні поняття та функції включають:

- Алгоритмічні методи прийняття рішень, включаючи:
- Класифікація: процес ідентифікації та розподілу даних на категорії на основі заданих шаблонів чи характеристик.
- Регресія: метод передбачення чи моделювання значень даних, використовуючи статистичні методи та середні значення наявних та планованих даних.
- Алгоритми прогнозного аналізу, які варіюються від простих методів лінійної регресії до складніших моделей розпізнавання образів, таких як нейронні мережі.

Кластеризація - це процес автоматичного групування схожих даних у наборі

даних без заздалегідь визначених категорій або міток. Метою кластеризації є знаходження внутрішніх структур або патернів у даних, які допомагають визначити подібність між об'єктами. Кластеризація може виявляти групи даних, які можуть бути невидимі або невідомі при попередньому аналізі. Вона може бути використана для виявлення прихованих залежностей, згрупування подібних об'єктів або розділення даних на підмножини зі спільними характеристиками. Кластеризація є важливим інструментом у багатьох галузях, таких як машинне навчання, аналіз даних, обробка зображень, біоінформатика та багато інших.

Взаємодія з популярними бібліотеками NumPy, pandas і matplotlib, що дозволяє легко обробляти та візуалізувати дані.

ML — це технологія, яка дозволяє комп’ютерам навчатися на вхідних даних і створювати/навчати прогнозну модель без явного програмування. ML є підмножиною штучного інтелекту (AI) [15].

2.7. Бібліотека XGBoost

XGBoost — це оптимізована розподілена бібліотека посилення градієнта, розроблена як високоефективна, гнучка та портативна. Він реалізує алгоритми машинного навчання під інфраструктурою Gradient Boosting[19].

XGBoost забезпечує паралельне прискорення дерева (також відоме як GBDT, GBM), яке швидко й точно вирішує багато проблем із наукою про дані. Той самий код працює в основному розподіленому середовищі (Hadoop, SGE, MPI) і може вирішу XGBoost — це алгоритм посилення градієнта, який широко використовується в науці про дані. Це реалізація посилення градієнта, яка розроблена як високоефективна, гнучка та портативна [16].

XGBoost був спочатку розроблений *Tianqi Chen* як вдосконалення алгоритму GBM. Алгоритм був розроблений з урахуванням наступних цілей:

- бути високоефективним;
- бути гнучким;
- бути портативним.

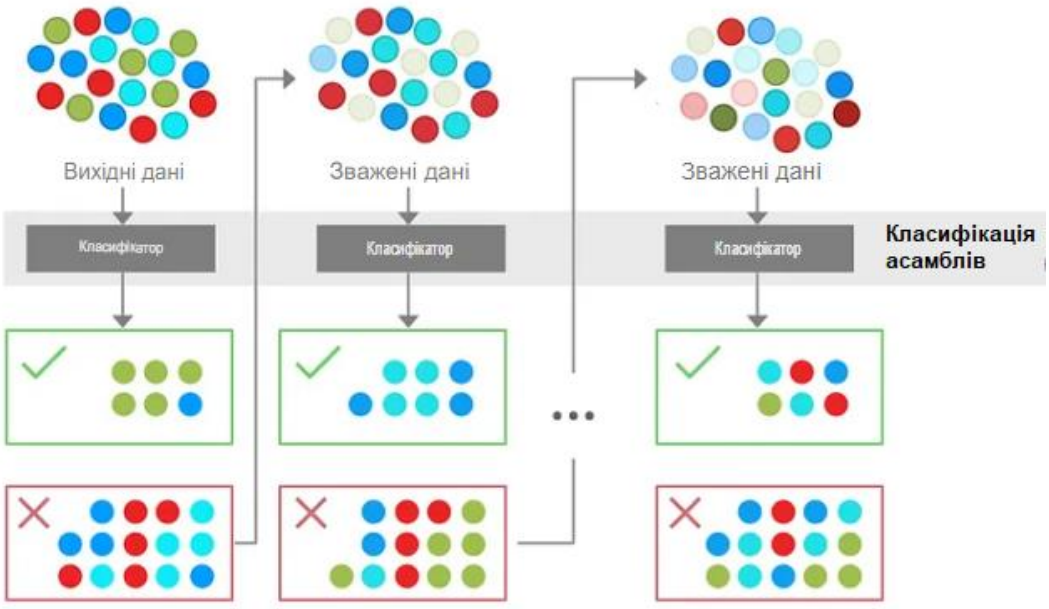

Рис. 2.3. Пояснення алгоритму XGBoost

Доведено, що XGBoost перевершує інші алгоритми машинного навчання в різних завданнях, включаючи класифікацію, регресію та ранжування (рис. 2.3):

- бути високоефективним;
- бути гнучким;
- бути портативним.

Доведено, що XGBoost перевершує інші алгоритми машинного навчання в різних завданнях, включаючи класифікацію, регресію та ранжування.

XGBoost має низку параметрів, які можна налаштувати для покращення продуктивності алгоритму. Найбільш важливими параметрами є:

- max_depth: максимальна глибина дерев рішень;
- eta: швидкість навчання;
- гамма: мінімальне зменшення втрат, необхідне для поділу;
- підвибірка: Частка навчальних даних, яка використовується для навчання кожного дерева.

XGBoost має ряд переваг порівняно з іншими алгоритмами машинного навчання:

- високоефективний;

- гнучкий;
- портативний;
- точний [17].

2.8. Графіки OHLC

Графіки OHLC (Open-High-Low-Close) використовуються в фінансовому аналізі та торгівлі на фінансових ринках, зокрема на ринку акцій, ф'ючерсних ринках та ринку валют (Forex). Цей тип графіків надає інформацію про цінову динаміку за певний період часу [24].

Основні компоненти графіків OHLC включають:

Відкриття (Open): Це перше значення ціни на початку періоду. Відкриття відображається як горизонтальна лінія, яка з'єднує пункти графіку.

Максимум (High): Це найвища ціна, досягнута протягом даного періоду. Максимум позначається вертикальною лінією, яка виступає вгору від відкриття.

Мінімум (Low): Це найнижча ціна, досягнута протягом даного періоду. Мінімум позначається вертикальною лінією, яка виступає вниз від відкриття.

Закриття (Close): Це останнє значення ціни на кінець періоду. Закриття відображається як горизонтальна лінія, яка з'єднує пункти графіку.

Графіки OHLC дозволяють візуалізувати коливання цін протягом певного часового періоду, надаючи торговцям та аналітикам важливу інформацію про тенденції ринку. Вони часто використовуються для визначення рівнів підтримки та опору, виявлення моделей свічок (candlestick patterns) та проведення технічного аналізу для прийняття рішень щодо входу чи виходу з позиції на ринку [25].

На рис. 2.4 представлена будова графіків OHLC.

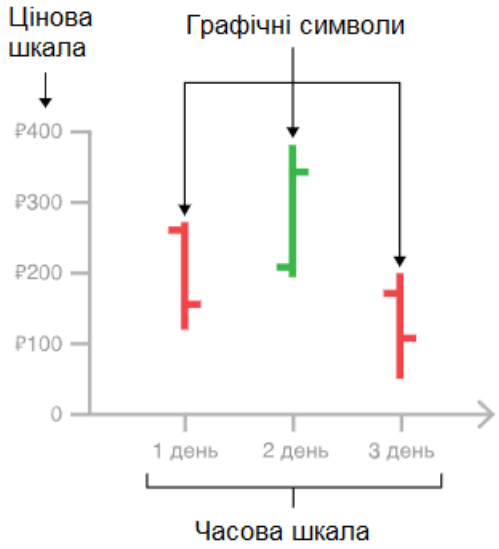

Рис. 2.4. Будова графіків OHLC

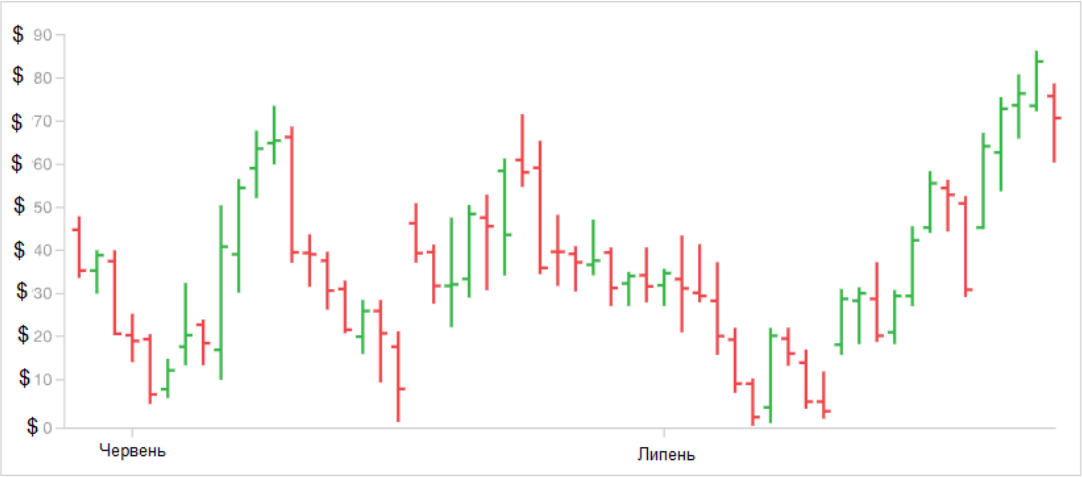

Рис. 2.5. Приклад графіку OHLC

2.8. Метод аналізу даних EDA

EDA (Exploratory Data Analysis) є процесом дослідження та аналізу даних з метою виявлення основних характеристик, закономірностей та взаємозв'язків між змінними. Вона допомагає розуміти структуру даних, виявляти аномалії, співвідношення, тренди та кореляції, що можуть бути корисними для подальшого аналізу та моделювання [26].

Основні кроки EDA включають:

1. Завантаження та огляд даних: Починаючи з отримання набору даних, важливо завантажити його та ознайомитись зі структурою та форматом. Це може включати перегляд перших декількох рядків даних, вивчення описових

статистик та перевірку наявності пропущених значень.

2. Візуалізація даних: Використовуючи графіки та діаграми, можна візуалізувати дані для отримання першого враження про їх розподіл та взаємозв'язки. Графіки можуть включати стовпчасті діаграми, гістограми, розсіювальні графіки, ящики з вусами тощо.

3. Обробка та очищення даних: На цьому етапі проводиться обробка даних для видалення аномальних значень, вирівнювання пропущених даних, перетворення змінних, якщо потрібно, та інші операції для покращення якості даних.

4. Аналіз залежностей: Використовуючи кореляційні матриці, теплові карти та інші методи, можна досліджувати залежності між змінними. Це дозволяє виявити потенційні впливи однієї змінної на інші та встановити фактори, які варто враховувати в аналізі.

5. Виявлення трендів та патернів: Аналізуючи дані зі зростанням часу або іншими змінними, можна виявити тренди, сезонність або інші патерни, що допоможуть зрозуміти динаміку даних та зробити висновки.

EDA є важливим етапом в аналізі даних, оскільки вона допомагає отримати перші відомості про дані, виявити проблеми та спрямувати подальший аналіз. Вона може виконуватись за допомогою різноманітних інструментів та програм, таких як Python (з використанням бібліотек, наприклад, Pandas, Matplotlib, Seaborn) або R (з використанням пакетів, таких як ggplot2, dplyr, tidyr) [27].

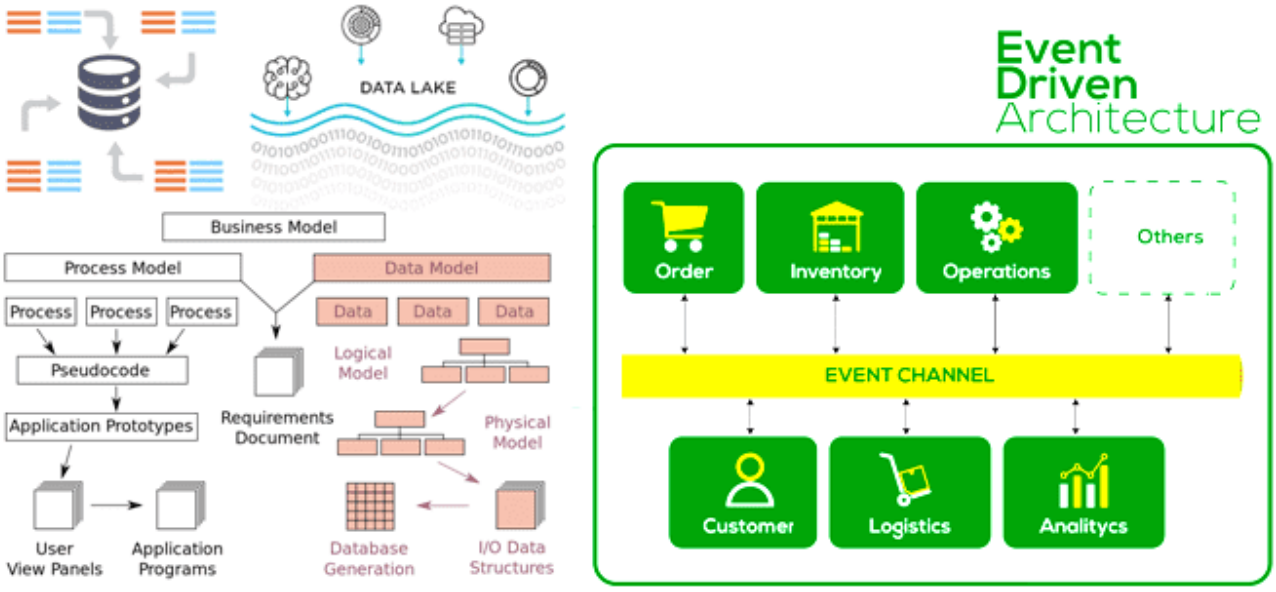

Рис. 2.6. Принцип роботи методу аналізу даних EDA

2.9. Крива ROC-AUC

Крива ROC (Receiver Operating Characteristic) та площа під кривою AUC (Area Under the Curve) є метриками для оцінки якості бінарних класифікаторів[28].

Крива ROC візуалізує залежність між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) класифікатора при зміні порогу прийняття рішення. Вона представляє собою графік, де по осі абсцис зображається специфічність, а по осі ординат - чутливість. Крива ROC може допомогти знаходити оптимальний поріг класифікації залежно від потреб дослідника.

Площа під кривою AUC вимірює площу під кривою ROC і використовується як числова метрика для оцінки ефективності класифікатора. Значення AUC може бути в діапазоні від 0 до 1, де значення 1 вказує на ідеальну класифікацію, а значення 0.5 означає випадковий класифікатор. Чим більше значення AUC, тим кращий класифікатор.

Крива ROC та AUC особливо корисні при роботі з незбалансованими наборами даних або в ситуаціях, коли важно виявити чутливість класифікатора до помилкових позитивних результатів.

Ці метрики широко використовуються в області машинного навчання та

оцінки якості моделей класифікації [29].

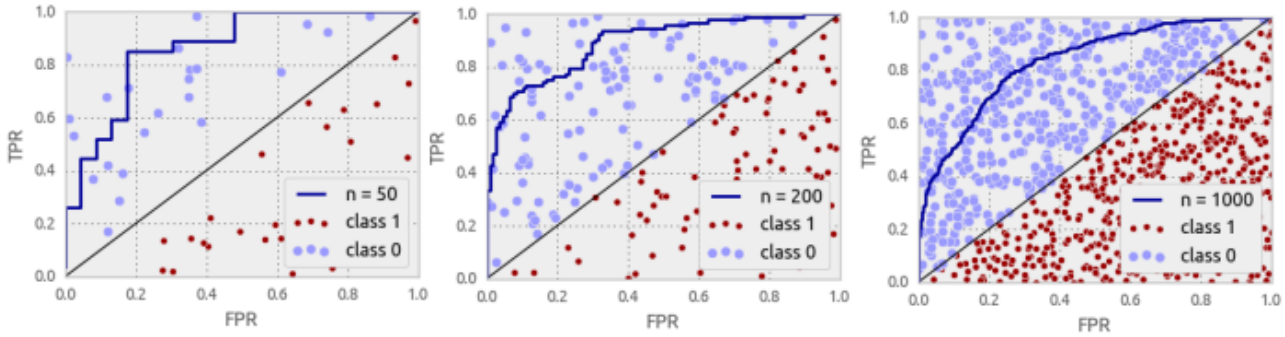

Рис. 2.7. Приклад кривої ROC-AUC

2.10. Boxplot

Boxplot (або діаграма «ящик з вусами») є графічним методом візуалізації статистичних даних, який надає інформацію про розподіл значень змінної та виявляє потенційні викиди (аномалії) в даних. В описовій статистиці коробкова діаграма — це зручний спосіб графічного зображення груп числових даних через їхні квартилі. Коробчастий графік відображає медіану, вищий/нижній квартиль і максимум/мінімум [30].

Основні компоненти boxplot включають:

1. Медіану (Q2): Це центральне значення розподілу даних. Вона позначається як лінія, яка розділяє ящик на дві половини.

2. Міжквартильний розмах (IQR): Це розмах між верхнім квартилем (Q3) та нижнім квартилем (Q1). Квартилі визначаються таким чином, що 25% значень знаходяться нижче Q1, 50% значень - між Q1 та Q3, і 25% значень - вище Q3.

3. Верхній вус (Upper Whisker): Це лінія, яка виходить вище ящика і відображає максимальне значення в діапазоні, не враховуючи потенційні викиди.

4. Нижній вус (Lower Whisker): Це лінія, яка виходить нижче ящика і відображає мінімальне значення в діапазоні, не враховуючи потенційні викиди.

5. Викиди (Outliers): Це окремі значення, які відхиляються від основного розподілу даних і відображаються як окремі точки або випадкові великі значення на вусах [31].

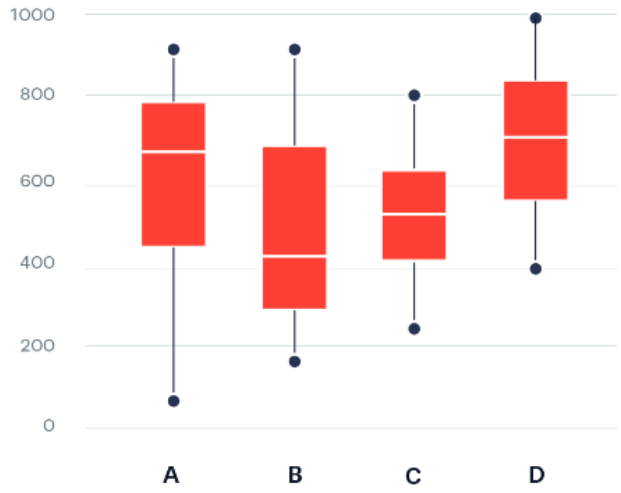

Рис. 2.8. Приклад застосування методу Boxplot

Висновки до розділу 2

В даному розділі були розглянуті та проаналізовані основні інструментальні засоби для реалізації програмного забезпечення, що буде прогнозувати курс криптовалюти: мова програмування Python, допоміжні бібліотеки (pandas, NumPy, Matplotlib, Sklearn, XGBoost тощо), метод аналізу даних – EDA.

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка програмного забезпечення (ПЗ) - це процес створення програмного продукту, який включає в себе проектування, розробку, тестування, впровадження та підтримку програмного забезпечення [22].

Машинне навчання (ML) виявилося надзвичайно корисним у багатьох галузях промисловості в автоматизації завдань, які раніше вимагали людської праці. Одним із таких застосувань ML є прогнозування того, чи буде певна торгівля прибутковою чи ні.

Ми дізнаємося, як передбачити сигнал, який вказує, чи буде покупка певної акції корисною чи ні за допомогою ML.

Давайте почнемо з імпорту деяких бібліотек, які використовуватимуться для різних цілей, які будуть пояснені далі в цій роботі.

У програмному забезпеченні буде реалізовано наступну функціональність, що включає в себе:

- створення вибірки даних з сховища;
- інтелектуальний аналіз даних;
- використання декількох моделей прогнозування даних;
- прогнозування перспектив криптовалюти;
- прогнозування факторів впливу на зміну криптовалюти;
- графічне відображення отриманих результатів та їх аналіз.

Формалізація постановки задачі дослідження

Задача, яка розглядається, може бути формалізована у вигляді математичної моделі з наступними складовими, представлена формулою 3.1 [3]:

$$R^* = R^*\left(d, R\left(d - d_f\right), R\left(d - d_f - 1\right), \dots, R\left(d - d_f - d_{mr}\right), T\left(d - d_f\right), T\left(d - d_f - 1\right), \dots, T\left(d - d_f - d_{mt}\right)\right), \quad (3.1)$$

де R^* – прогнозований курс біткоіна,

d – дата прогнозу в днях,

R – реальний курс біткоіна,

T – кількість постів у $Twitter$ зі згадуванням $bitcoin$ за день,

d_f – день прогнозу,

d_{mr} – кількість днів за які подаються дані реального курсу,

d_{mt} – кількість днів за які подаються дані з Twitter.

3.1. Імпорт бібліотек

Бібліотеки Python дуже полегшують нам обробку даних і виконання типових і складних завдань за допомогою одного рядка коду.

На цьому етапі були завантажені та імпортовані основні бібліотеки: Pandas, Numpy, Matplotlib, Seaborn, Sklearn.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn import metrics

import warnings
warnings.filterwarnings('ignore')
```

Рис. 3.1. Імпорт бібліотек

3.2. Імпорт набору даних

Набір даних, який ми будемо використовувати тут для виконання аналізу та створення прогнозної моделі, є даними про ціну біткойна. Ми використовуватимемо дані OHLC («Open», «High», «Low», «Close») з 17 липня 2014 року по 29 грудня 2022 року, тобто за 8 років для ціни біткойна.

```
df = pd.read_csv('bitcoin.csv')
df.head()
```

Рис. 3.2. Імпорт набору даних

Виведення даних показано на рис. 3.3 нижче:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100

Рис. 3.3. Вихід даних - перші п'ять рядків даних

Про розмір таблиці з даними повідомляє її атрибут shape. У результаті виходить кортеж (незмінний список) із двох чисел: перше – кількість рядків, друге – кількість стовпців. А кортеж – одномірна незмінна послідовність. Це структура даних, схожа на список, її також можна зберігати у змінній.

Допишемо рядок df.shape, та отримує результат нижче (рис. 3.4):

```
(2904, 7)
```

Рис.3.4. Вихід отриманих даних

З цього ми дізналися, що доступно 2904 рядки даних, і для кожного рядка ми маємо 7 різних функцій або стовпців.

Якщо допишемо рядок df.describe(), буде наступне:

	Open	High	Low	Close	Adj Close	Volume
count	2904.000000	2904.000000	2904.000000	2904.000000	2904.000000	2.904000e+03
mean	12615.330584	12941.782231	12250.477381	12620.375694	12620.375694	1.569188e+10
std	16485.373722	16910.146022	15993.681472	16480.704501	16480.704501	1.984987e+10
min	176.897003	211.731003	171.509995	178.102997	178.102997	5.914570e+06
25%	650.070252	657.745758	637.294510	650.874771	650.874771	9.073677e+07
50%	6660.895019	6794.083252	6547.864990	6674.425049	6674.425049	6.259845e+09
75%	11977.260254	12528.664063	11710.292237	12032.553222	12032.553222	2.688600e+10
max	67549.734375	68789.625000	66382.062500	67566.828125	67566.828125	3.509679e+11

Рис.3.5. Описові статистичні вимірювання даних

Допишемо df.info(), отримаємо вихід:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2904 entries, 0 to 2903
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2904 non-null   object
 1   Open       2904 non-null   float64
 2   High       2904 non-null   float64
 3   Low        2904 non-null   float64
 4   Close      2904 non-null   float64
 5   Adj Close  2904 non-null   float64
 6   Volume     2904 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 158.9+ KB
```

Рис. 3.6. Вихід даних

3.3. Дослідницький аналіз даних

EDA — це підхід до аналізу даних за допомогою візуальних методів. Він використовується для виявлення тенденцій і закономірностей або для перевірки припущень за допомогою статистичних підсумків і графічних зображень [23].

Виконуючи EDA даних про ціну біткойна, ми проаналізуємо, як змінилися ціни на криптовалюту протягом періоду часу та як кінець кварталу впливає на ціни валюти.

Додамо наступнірядки:

```
plt.figure(figsize=(15, 5))
plt.plot(df['Close'])
plt.title('Bitcoin Close price.', fontsize=15)
plt.ylabel('Price in dollars.')
plt.show()
```

Отримаємо графік як на рис.3.7:

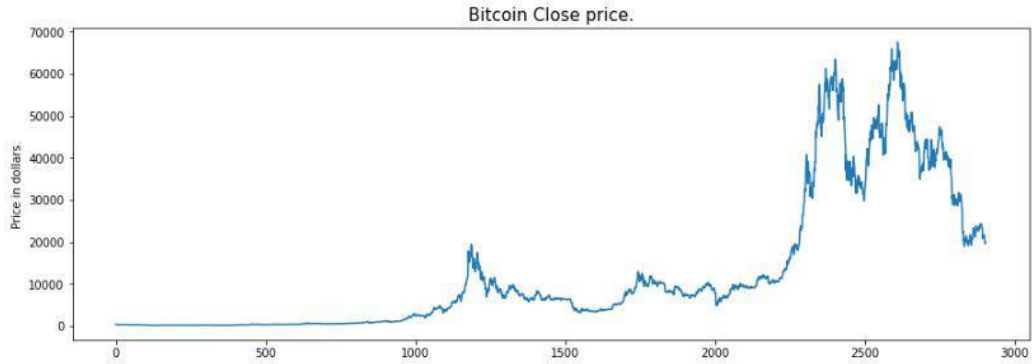

Рис.3.7. Графік зміни ціни криптовалюти

Ціни біткойн-акцій демонструють висхідну тенденцію, як показано на графіку ціни закриття акцій.

Дописали наступнийрядок:

```
df[df['Close'] == df['Adj Close']].shape, df.shape
```

Отримали вихід ((2904,7), (2904, 7)).

Звідси можна зробити висновок, що всі рядки стовпців «Close» і «Adj Close» мають однакові дані. Таким чином, наявність надлишкових даних у наборі даних не допоможе, тому ми видалимо цей стовпець перед подальшим аналізом.

Записали такий рядок:

```
df = df.drop(['Adj Close'], axis=1)
```

Тепер намалюємо графік розподілу для безперервних функцій, наданих у наборі даних, але перш ніж рухатися далі, давайте перевіримо нульові значення, якщо вони присутні у кадрі даних.

Нижче додамо рядок, отримуємо суму нульових значень у стовпцці.

```
df.isnull().sum()
```

```
Date         0
Open         0
High         0
Low          0
Close        0
Volume       0
Adj Close    0
dtype: int64
```

Рис. 3.8. Сума нульових значень у стовпці

Це означає, що в наданому наборі даних немає нульових значень.

Далі пропишемо рядок:

```
features = ['Open', 'High', 'Low', 'Close']

plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,2,i+1)
  sb.distplot(df[col])
plt.show()
```

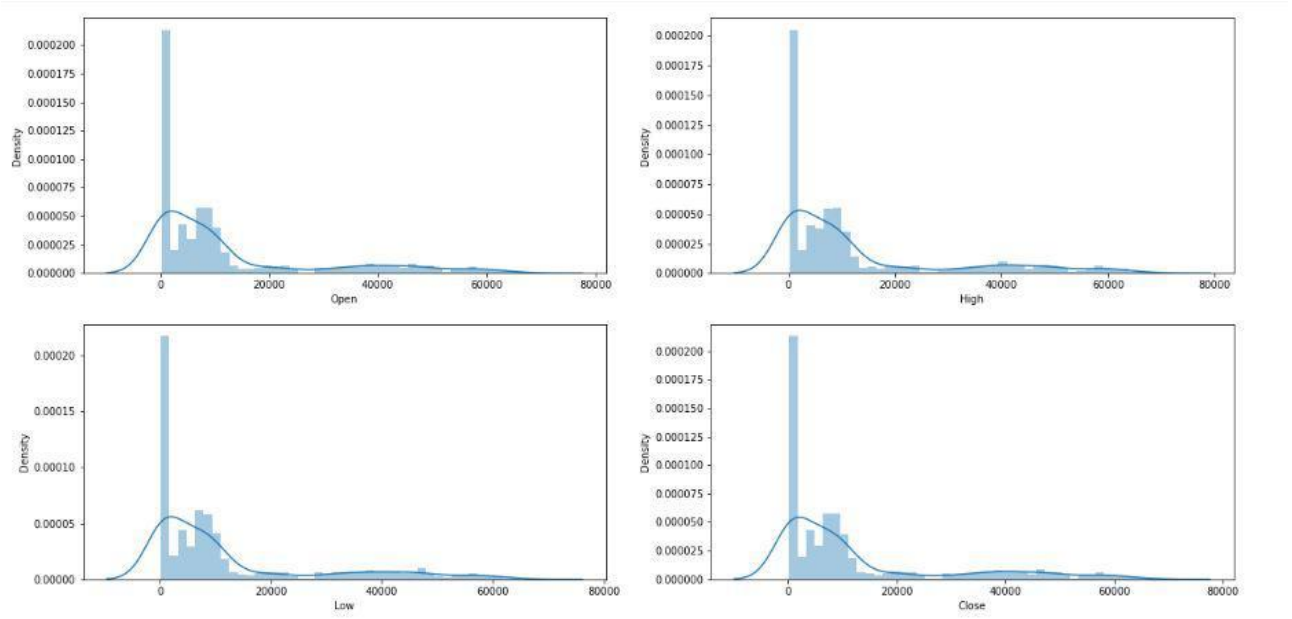

Рис. 3.9. Діаграма розподілу даних OHLC

У результаті отримали діаграму даних OHLC.

Графіки OHLC (Open-High-Low-Close), використовуються як трейдинговий інструмент для візуалізації та аналізу змін цін на облігації, валюти, акції, цінні папери тощо за певний період часу. Графіки OHLC зручні для регулярного аналізу настроїв ринку та прогнозування цінових змін у майбутньому на основі виявлених патернів [24].

Додали ще такі рядки:

```
plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,2,i+1)
  sb.boxplot(df[col])
plt.show()
```

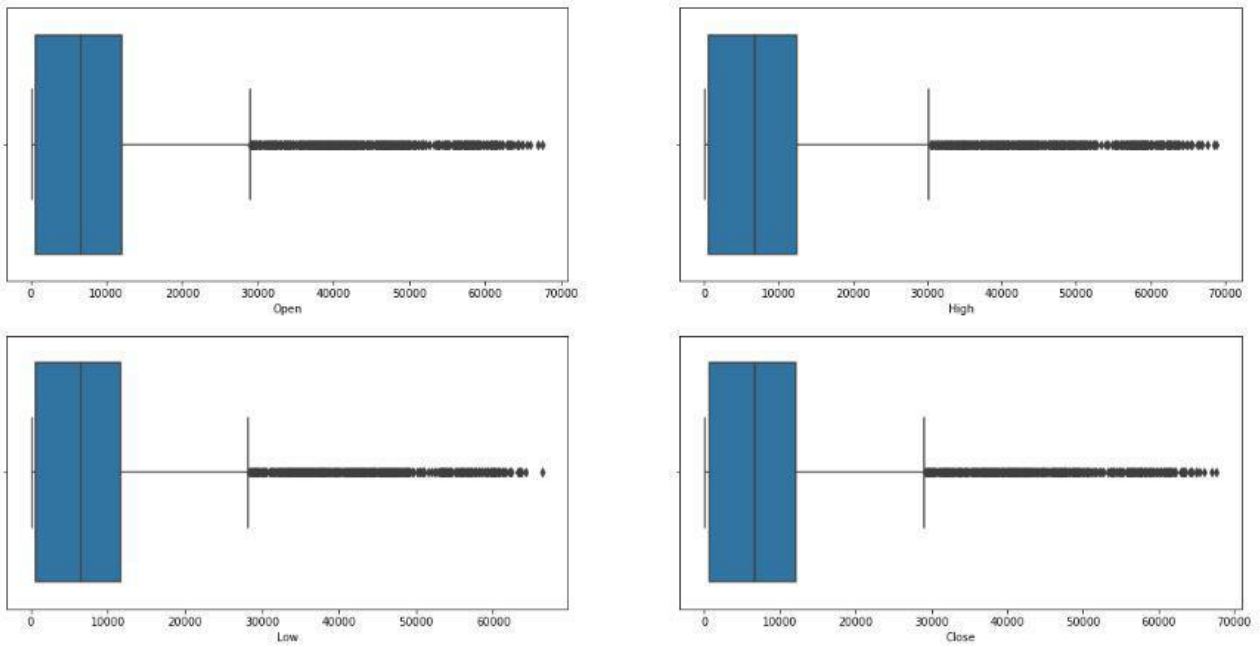

Рис. 3.10. Boxplot даних OHLC

У даних так багато викидів, що означає, що ціни на акції сильно змінювалися за дуже короткий період часу. Перевіримо це за допомогою бардіаграми.

3.4. Розробка функцій

Розробка функцій допомагає отримати деякі цінні функції з існуючих. Ці додаткові функції іноді допомагають значно підвищити продуктивність моделі та, звичайно, допомагають отримати глибше розуміння даних.

```
splited = df['Date'].str.split('-', expand=True)
df['year'] = splited[0].astype('int')
df['month'] = splited[1].astype('int')
df['day'] = splited[2].astype('int')
 df.head()
```

Вихід показано на рис.3.11:

	Date	Open	High	Low	Close	Volume	year	month	day
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	2014	9	17
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	2014	9	18
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	2014	9	19
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	2014	9	20
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	2014	9	21

Рис. 3.11. Перші п'ять рядків даних

Тепер у нас є ще три стовпці, а саме «день», «місяць» і «рік», усі ці три були отримані зі стовпця «Дата», який спочатку був наданий у даних.

Наступні рядки додамо:

```
data_grouped = df.groupby('year').mean()
plt.subplots(figsize=(20,10))
for i, col in enumerate(['Open', 'High', 'Low', 'Close']):
  plt.subplot(2,2,i+1)
  data_grouped[col].plot.bar()
plt.show()
```

Вихід:

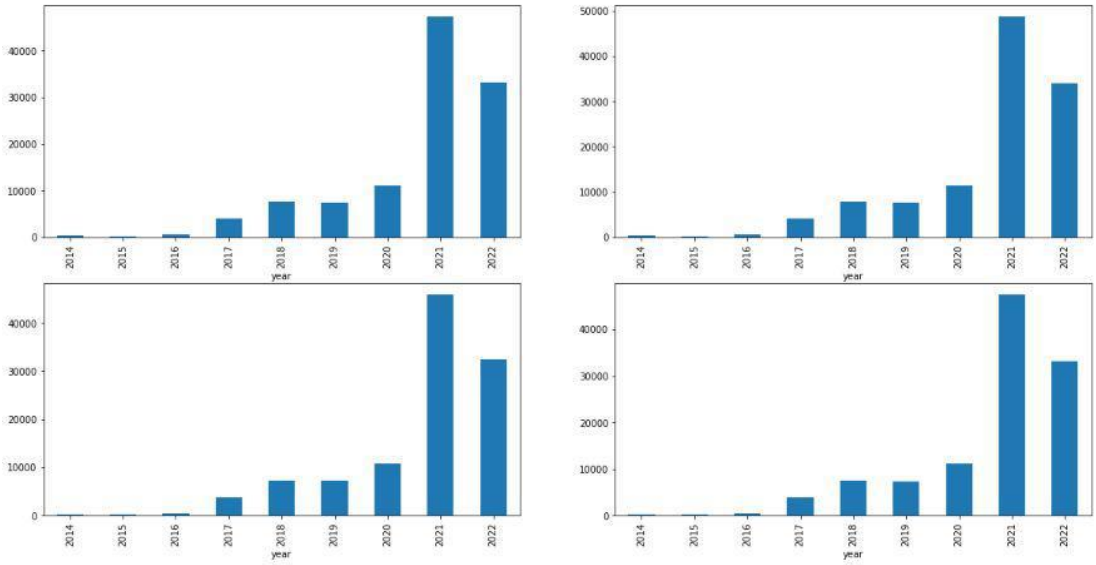

Рис. 3.12. Бар діаграма середньої ціни біткойна за рік

Тут ми можемо спостерігати, чому в даних так багато викидів, оскільки ціни на біткойни вибухнули в 2021 році.

Дописано рядок:

```
df['is_quarter_end'] = np.where(df['month']%3==0,1,0)
df.head()
```

Вихід представлено на рис.3.13:

	Date	Open	High	Low	Close	Volume	year	month	day	is_quarter_end
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	2014	9	17	1
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	2014	9	18	1
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	2014	9	19	1
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	2014	9	20	1
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	2014	9	21	1

Рис. 3.13. Перші п'ять рядків даних

Додамо ще кілька стовпців, які допоможуть у навчанні нашої моделі:

```
df['open-close']  = df['Open'] - df['Close']
```

```
df['low-high']  = df['Low'] - df['High']
df['target'] = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)
```

Ми додали цільову функцію, яка є сигналом, купувати чи ні. Ми навчимо нашу модель передбачати лише це.

Але перш ніж продовжити, давайте перевіримо, чи ціль збалансована чи ні, використовуючи секторну діаграму.

Кругова (секторна) діаграма— це графічне зображення статистичних даних у вигляді циклічного діаграми. У такій діаграмі представлений лише один ряд даних, а площа кожного сектора відповідає відсотковому співвідношенню цієї частини даних до загальної суми [33].

Нижче код для її створення:

```
plt.pie(df['target'].value_counts().values,
        labels=[0, 1], autopct='%1.1f%%')
plt.show()
```

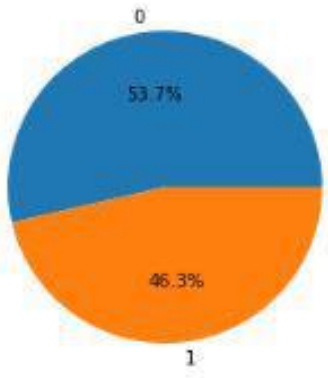

Рис. 3.14. Секторна діаграма для розподілу даних між двома мітками

Коли ми додаємо функції до нашого набору даних, ми повинні переконатися, що немає сильно корельованих функцій, оскільки вони не допомагають у процесі навчання алгоритму.

Теплові карти в Seaborn можна побудувати за допомогою функції **seaborn.heatmap().**

```
plt.figure(figsize=(10, 10))
# As our concern is with the highly
# correlated features only so, we will visualize
# our heatmap as per that criteria only.
sb.heatmap(df.corr() > 0.9, annot=True, cbar=False)
plt.show()
```

Теплова карта визначається як графічне представлення даних із використанням кольорів для візуалізації значення матриці [32].

У цьому випадку для представлення більш поширених цінностей або вищої активності використовуються більш яскраві кольори, в основному червонуваті кольори, а для представлення менш поширених значень або цінностей активності перевага віддається темнішим кольорам. Теплова карта також визначається назвою матриці затінення.

Вихід:

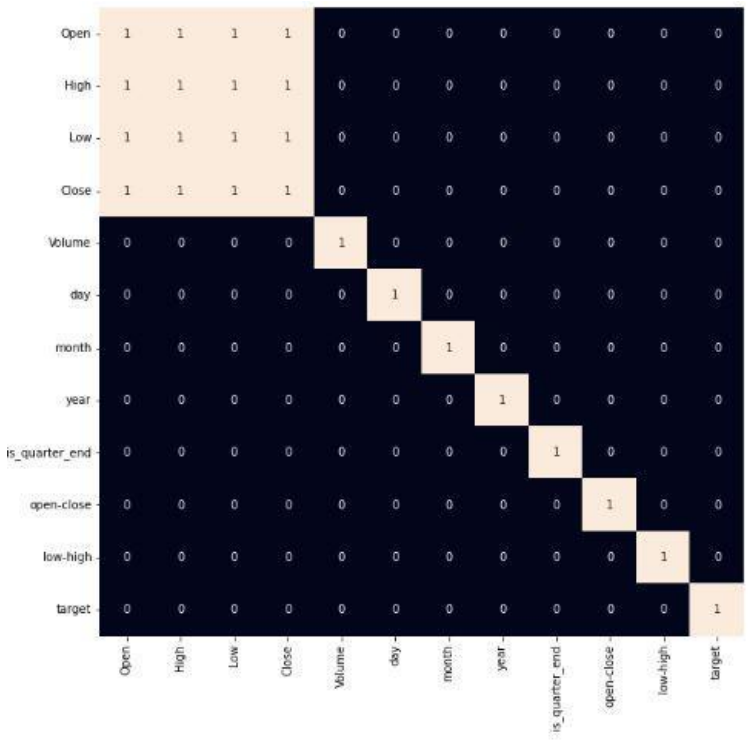

Рис. 3.15. Теплова карта для пошуку висококорельованих функцій

З наведеної вище карти тепла ми можемо сказати, що існує висока кореляція між OHLC, яка є досить очевидною, і додані функції не сильно корелюють одна з одною або раніше наданими функціями, що означає, що ми готові створювати нашу модель.

```
features = df[['open-close', 'low-high', 'is_quarter_end']]
target = df['target']
scaler = StandardScaler()
features = scaler.fit_transform(features)
 X_train, X_valid, Y_train, Y_valid = train_test_split(
 features, target, test_size=0.1, random_state=2022)
print(X_train.shape, X_valid.shape)
```

Вихід:

```
(2613, 3) (291, 3)
```

Після вибору функцій для навчання моделі ми повинні нормалізувати дані, оскільки нормалізовані дані призводять до стабільного та швидкого навчання моделі. Після цього всі дані були розділені на дві частини у співвідношенні 90/10, щоб ми могли оцінити продуктивність нашої моделі на невидимих даних.

3.5. Розробка та оцінка моделі

Зараз настав час навчити деякі найсучасніші моделі машинного навчання (логістична регресія, опорна векторна машина, XGBClassifier), а потім, ґрунтуючись на їх ефективності на основі даних навчання та перевірки, ми виберемо, яка модель ML служить меті під рукою краще.

Для метрики оцінки ми будемо використовувати криву ROC-AUC, але чому - це тому, що замість прогнозування твердої ймовірності, яка дорівнює 0 або 1, ми хотіли б, щоб вона передбачала м’які ймовірності, які є безперервними

значеннями від 0 до 1. А з м’якими ймовірностями крива ROC-AUC зазвичай використовується для вимірювання точності прогнозів.

```
models = [LogisticRegression(), SVC(kernel='poly', probability=True), XGBClassifier()]
for i in range(3):
models[i].fit(X_train, Y_train)
 print(f'{models[i]} : ')
print('Training Accuracy : ', metrics.roc_auc_score(Y_train, models[i].predict_proba(X_train)[:,1]))
print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid, models[i].predict_proba(X_valid)[:,1]))
print()
```

Вихід представлено на рис. 3.16:

```
LogisticRegression() :
Training Accuracy :  0.5221235185617062
Validation Accuracy :  0.5595588934309346

SVC(kernel='poly', probability=True) :
Training Accuracy :  0.48147598485758647
Validation Accuracy :  0.47471242513546913

XGBClassifier() :
Training Accuracy :  0.7109072256262028
Validation Accuracy :  0.49588839243274074
```

Рис. 3.16. Продуктивність різних найсучасніших моделей.

Серед трьох моделей навчена нами XGBClassifier має найвищу продуктивність, але її скорочено до переобладнання , оскільки різниця між точністю навчання та перевірки надто висока. Але у випадку логістичної регресії це не так.

Тепер давайте побудуємо матрицю плутанини для даних перевірки.

```
metrics.plot_confusion_matrix(models[0], X_valid, Y_valid)
```

```
plt.show()
```

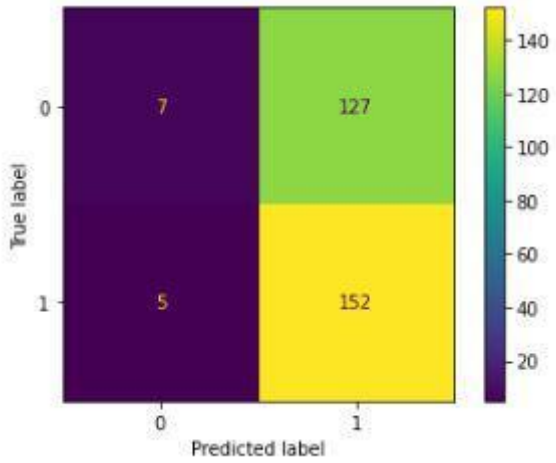

Рис. 3.17. Матриця помилок для даних перевірки

Отже, в результаті даної роботи можемо помітити, що точність, досягнута найсучаснішою моделлю ML, не краща, ніж просто вгадування з імовірністю 50%. Можливими причинами цього може бути відсутність даних або використання дуже простої моделі для виконання такого складного завдання, як прогноз фондового ринку.

Висновки до розділу 3

Отже, в результаті роботи було поетапно розібрано процес розробки програмного забезпечення (імпорт бібліотек, дослідницький аналіз даних, розробка функцій, розробка та оцінка моделі і т. д.), описано частини коду та приклад їх функціонування.

Точність, досягнута найсучаснішою моделлю ML, не краща, ніж просто вгадування з імовірністю 50%. Можливими причинами цього може бути відсутність даних або використання дуже простої моделі для виконання такого складного завдання, як прогноз фондового ринку.

ВИСНОВКИ

У результаті виконання наукової роботи було проаналізовано предметну область, визначені основні поняття, концепції криптовалюти, блокчейну. В процесі роботи було розглянуто та проаналізовано існуючі рішення з прогнозування курсу криптовалют, виконано аналіз інструментальних засобів, на основі яких було безпосередньо розроблено програмне забезпечення мовою програмуванн Pyton для прогнозування зміни курсу криптовалюти Bitcoin.

Наукова новизна даної роботи полягає у тому, що розроблене програмне забезпечення використовує сучасні технології машинного навчання, в основі якого лежить ідея, що комп'ютерні системи можуть аналізувати дані, виявляти патерни і навчатися на основі цих патернів, щоб здійснювати передбачення або приймати рішення у майбутньому на прикладі прогнозування криптовалют.

Практична значущість результатів дослідження може бути застосовано для допомоги інвесторам прийняти раціональні рішення щодо купівлі, продажу або утримання цифрових активів, отримавши аналітичну інформацію з візуальним відображенням у вигляді графіків. Таким чином компанія або людина, яка хоче торгувати на ринку криптовалют, це рішення заощадить час та кошти на витратах аналітичних послуг.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Криптовалюта – наше майбутнє чи чергова афера? URL: https://galychyna.if.ua/analytic/kriptovalyuta-nashe-maybutnye-chi-chergova-afera/
2. Переваги та недоліки криптовалюти. URL: https://kryptovaluta.info/
3. Bitcoin. BTC URL: https://coinmarketcap.com/currencies/bitcoin/
4. Ethereum. ETH URL: https://coinmarketcap.com/currencies/ethereum/
5. Walletinvestor. URL: https://walletinvestor.com/forecast
6. Belinvestor. URL: https://belinvestor.com/cryptocurrencies/
7. NeuroShell. URL: https://try.neuroshell.com/index/
8. Trader. URL: https://tradermake.money
9. Python Programming Language. URL: https://www.python.org/
10. A computer science portal for geeks. URL: https://www.geeksforgeeks.org/
11. Pandas. URL: https://pandas.pydata.org/
12. NumPy. URL: https://pythonworld/numpy/1.html
13. Matplotlib. URL: https://matplotlib.org/
14. Seaborn. URL: https://seaborn.pydata.org/
15. Scikit-learn. URL: https://www.activestate.com/resources/quick-reads/what-is-scikit-learn-in-python/#:~:text=Scikit%2Dlearn%20is%20an%20open,categorizing%20data%20based%20on%20patterns.
16. XGBoost. URL: https://xgboost.readthedocs.io/en/stable/
17. XGBoost Algorithm Explained in Less Than 5 Minutes. URL:https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee
18. ТОП-10 найпопулярніших криптовалют у 2023 році. URL: https://tribun.com.ua/ru/98941-kak-uxazhivat-za-kozhej-vokrug-glaz-posle-30-let
19. XGBoost. URL: https://uk.wikipedia.org/wiki/XGBoost
20. Scikit-learn. URL: https://uk.wikipedia.org/wiki/Scikit-learn

21. Seaborn для візуалізації даних у Python. URL: https://pythonru.com/biblioteki/seaborn-plot
22. Розробка програмного забезпечення. URL: http://surl.li/uuof
23. ВІЗУАЛІЗАЦІЯ ДАНИХ ДЛЯ DATA SCIENCE. URL: http://surl.li/hutrf
24. OHLC. URL: https://datavizcatalogue.com/RU/metody/grafik_barov_ohlc.html.
25. Що таке OHLC. URL: http://surl.li/huuff
26. Розвідковий аналіз даних. URL: https://www.wikiwand.com/uk/Exploratory_Data_Analysis
27. EDA під іншим кутом. URL: https://habr.com
28. ROC-крива. URL: https://uk.wikipedia.org/wiki/ROC-%D0%BA%D1%80%D0%B8%D0%B2%D0%B0
29. ROC-криві. Оглядова стаття. URL: https://dou.ua/forums/topic/33858/#:~:text=%D0%9F%D0%BB%D0%BE%D1%89%D1%83%20%D0%BF%D1%96%D0%B4%20ROC%20%D0%BA%D1%80%D0%B8%D0%B2%D0%BE%D1%8E%20%D0%BD%D0%B0%D0%B7%D0%B8%D0%B2%D0%B0%D1%8E%D1%82%D1%8C,%D1%89%D0%BE%20%D0%BE%D0%BF%D0%B8%D1%81%D1%83%D1%94%20%C2%AB%D1%96%D0%B4%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D1%83%C2%BB%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C.
30. Коробковий графік. URL: httpshttps://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BE%D0%B1%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BABoxplot.
31. Boxplot. URL: https://datavizproject.com/data-type/box-plot/
32. Теплові карти. URL: https://www.visitor-analytics.io/ua/glosarii/kh/teplovi-karti/
33. Секторна діаграма. URL: http://surl.li/huwls
34. Що ж таке крипта простими словами. URL: https://life.fakty.com.ua

ДОДАТКИ

Додаток А

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn import metrics

import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('bitcoin.csv')
df.head()
df.shape
df.describe()
df.info()
plt.figure(figsize=(15, 5))
plt.plot(df['Close'])
plt.title('Bitcoin Close price.', fontsize=15)
plt.ylabel('Price in dollars.')
plt.show()

df[df['Close'] == df['Adj Close']].shape, df.shape
df = df.drop(['Adj Close'], axis=1)
```

```
df.isnull().sum()

features = ['Open', 'High', 'Low', 'Close']

plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,2,i+1)
  sb.distplot(df[col])
plt.show()

plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,2,i+1)
  sb.boxplot(df[col])
plt.show()


splited = df['Date'].str.split('-', expand=True)

df['year'] = splited[0].astype('int')
df['month'] = splited[1].astype('int')
df['day'] = splited[2].astype('int')

df.head()


data_grouped = df.groupby('year').mean()
plt.subplots(figsize=(20,10))
for i, col in enumerate(['Open', 'High', 'Low', 'Close']):
  plt.subplot(2,2,i+1)
  data_grouped[col].plot.bar()
plt.show()
```

```
df['is_quarter_end'] = np.where(df['month']%3==0,1,0)
df.head()

df['open-close']  = df['Open'] - df['Close']
df['low-high']  = df['Low'] - df['High']
df['target'] = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)

plt.pie(df['target'].value_counts().values,
        labels=[0, 1], autopct='%1.1f%%')
plt.show()

plt.figure(figsize=(10, 10))

# As our concern is with the highly
# correlated features only so, we will visualize
# our heatmap as per that criteria only.
sb.heatmap(df.corr() > 0.9, annot=True, cbar=False)
plt.show()

features = df[['open-close', 'low-high', 'is_quarter_end']]
target = df['target']

scaler = StandardScaler()
features = scaler.fit_transform(features)

X_train, X_valid, Y_train, Y_valid = train_test_split(
    features, target, test_size=0.1, random_state=2022)
print(X_train.shape, X_valid.shape)

models = [LogisticRegression(), SVC(kernel='poly', probability=True),
XGBClassifier()]
```

```
for i in range(3):
  models[i].fit(X_train, Y_train)


  print(f'{models[i]} : ')
  print('Training Accuracy : ', metrics.roc_auc_score(Y_train,
models[i].predict_proba(X_train)[:,1]))
  print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid,
models[i].predict_proba(X_valid)[:,1]))
  print()


metrics.plot_confusion_matrix(models[0], X_valid, Y_valid)
plt.show()
```