

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системи і методи штучного
інтелекту»**

спеціальності 122 «Комп'ютерні науки»

на тему: «Комп'ютерні науки»

**на тему: «Прогнозування факту лістингу нових криптовалют на біржі
Binance із використанням моделі логістичної регресії»**

Виконав:

студент IV курсу, групи КІ-11

Залізник Ілля Сергійович

Керівник:

доцент кафедри ШІ, д.т.н., доцент,

Терент'єв Олександр Миколайович

Консультант з економічного розділу:

доцент кафедри економічної кібернетики, к.е.н., доцент,

Рощина Надія Василівна

Консультант з нормоконтролю:

фахівець першої категорії кафедри ШІ, к.т.н., доцент,

Комариста Богдана Миколаївна

Рецензент: с.н.с. відділу прикладної інформатики

Інституту телекомунікацій і глобального інформаційного

простору к.е.н., доц. Тетяна Іванівна Просянкін-Жарова

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«15» січня 2025 р.

ЗАВДАННЯ

на дипломну роботу студенту

Залізняка Іллі Сергійовичу

1. Тема роботи «Прогнозування факту лістингу нових криптовалют на біржі Binance із використанням моделі логістичної регресії», керівник роботи доктор. техн. наук Терент'єв Олександр Миколайович, затверджені наказом по НН ІПСА від «26» травня 2025 р. № 1759-с.
2. Термін подання студентом роботи «09» червня 2025 року.
3. Вихідні дані до роботи: зображення, що містять текст.
4. Зміст роботи: теоретичні основи побудови інформаційно-аналітичних систем зберігання та аналітичної обробки інформації, побудови прогнозуючих математичних моделей.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація.
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Надія Василівна, доцент, к. е. н.		

7. Дата видачі завдання «03» лютого 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Вивчення літератури за темою роботи	21.04.2025	Виконано
2	Підготовка першого розділу	28.04.2025	Виконано
3	Підготовка другого розділу	05.05.2025	Виконано
4	Початок розробки програмного продукту	12.05.2025	Виконано
5	Підготовка третього розділу	19.05.2025	Виконано
6	Підготовка економічної частини	26.05.2025	Виконано
7	Підготовка презентації доповіді	02.06.2025	Виконано
8	Оформлення дипломної роботи	09.06.2025	Виконано

Студент

Ілля ЗАЛІЗНЯК

Керівник

Олександр ТЕРЕНТЬЄВ

РЕФЕРАТ

Дипломна робота: 138 с., 24 табл., 25 рис., 5 додатків, 36 джерела.

BINANCE, КРИПТОВАЛЮТА, ЛІСТИНГ, ЛОГІСТИЧНА РЕГРЕСІЯ, ПРОГНОЗУВАННЯ.

Об'єкт дослідження – дані про криптоактиви: технічні характеристики, обсяг торгів та ціни.

Предмет дослідження – математичні методи бінарної класифікації для прогнозування ймовірностей із використанням мови прогнозування Python.

Мета роботи – виявити ключові фактори, що визначають рішення Binance щодо лістингу, та розробити аналітичну модель логістичної регресії, яка за сукупністю цих факторів прогнозує факт лістингу.

Актуальність. Криптовалюти стають все більш популярними у всьому світі. Кількість нових криптоактивів зростає кожного дня з великою швидкістю. Розуміння того, які проекти мають потенціал дозволяє сформувати прибуткові стратегії торгівлі на крипторинку.

Результати роботи. Створено програмний інструмент який демонструє роботу побудованих моделей логістичної регресії, що прогнозують лістинг криптовалюти на біржу Binance. Програмний продукт був реалізований із використанням мови програмування Python у середовищі розробки Visual Studio Code. Для побудови моделей було використано Jupyter Notebook.

Шляхи подальшого розвитку предмету дослідження. Підвищення точності прогнозування за рахунок використання більш складних моделей, таких як ансамблеві моделі або нейронні мережі. Збільшення кейсів у наборах даних для навчання. Прогнозування лістингу на інших біржах. Розширення показників у наборі даних.

ABSTRACT

Bachelor's thesis: 138 pp., 24 tables, 25 figures, 5 appendices, 36 references.

BINANCE, CRYPTOCURRENCY, LISTING, LOGISTIC REGRESSION, PREDICTION.

Object of the study. Data on crypto-assets: their technical characteristics, trading volumes, and prices.

Subject of the study. Binary-classification methods for probability forecasting implemented in the Python programming language.

Purpose of the work – to identify the key factors that influence Binance's listing decisions and to develop a logistic-regression model that predicts the fact of listing based on those factors.

Relevance. Cryptocurrency is growing in popularity worldwide, and the number of new crypto-assets is increasing rapidly every day. Understanding which projects are most likely to be listed enables traders and investors to build profitable strategies in the crypto market.

Results. A software tool was created that showcases logistic-regression models able to forecast whether a cryptocurrency will be listed on Binance. The product was developed in Python within the PyCharm environment, while Jupyter Notebook was used for model building and experimentation.

Directions for further research. Improving prediction accuracy through more sophisticated models such as ensemble methods or neural networks; increasing the number of cases in the training dataset; forecasting listings on other exchanges; expanding the range of indicators in the dataset.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	9
ВСТУП	12
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	14
1.1 Актуальність задачі прогнозування лістингу нових криптовалют	14
1.2 Опис предметної області	15
1.3 Лістинг криптовалют	16
1.3.1 Процес лістинг криптовалют	16
1.3.2 Типи криптобірж	17
1.3.3 Які фактори впливають на прийняття рішення про лістинг	18
1.4 Класифікація криптоактивів	19
1.4.1 Токени протоколів першого рівня	20
1.4.2 Токени протоколів другого рівня	20
1.4.3 Дефі токени	21
1.4.4 Екосистемні токени	21
1.4.5 Стейбкоїни	22
1.5 Вплив лістингу на цінову динаміку криптовалют	22
1.5.1 Вплив лістингу на біржі Binance на ціну криптовалюти	23
1.5.2 Вплив лістингу на інших біржах на ціну криптовалюти	24
1.6 Показники які впливають на рішення про лістинг криптоактиву	24
1.6.1 Продукт і технологія проєкту	25
1.6.2 Команда і екосистема	26
1.6.3 Токеноміка і ринкова динаміка	26
1.6.4 Юридична та безпекова відповідальність	27
1.7 Прогнозування факту лістингу	28
1.7.1 Використання логістичної регресії	28
1.7.2 Обробка показників	29
1.7.3 Побудова та тестування моделей	29

	7
1.8 Постановка задачі	30
Висновки до розділу 1	30
РОЗДІЛ 2 МЕТОДИКА ПРОГНОЗУВАННЯ	32
2.1 Поняття структури математичної моделі авторегресії	30
2.2 Методи регресійного аналізу	33
2.2.1. Метод найменших квадратів	36
2.2.2 Простий приклад	36
2.2.3 Геометричний зміст метода найменших квадратів	38
2.2.4 Вивід методу найменших квадратів	39
2.2.5 Незміщеність оцінки вектору коефіцієнтів	41
2.2.6 Ітераційний метод найменших квадратів	42
2.2.7 Коваріація вектору оцінок коефіцієнтів	45
2.3 Лінійні імовірнісні моделі	47
2.3.1 Пробіт-модель	49
2.3.2 Логіт-модель	52
2.4 Історія виникнення та практика застосування логістичної регресії	53
2.5 Критерії якості	55
2.5.1 Описові характеристики математичної моделі	55
2.5.2 Прогнозуючі характеристики регресійної моделі	57
2.6 Оцінка якості моделі в задачах класифікації	58
Висновки до розділу 2	62
РОЗДІЛ 3 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ	63
3.1 Вступ	63
3.2 Опис джерел для аналізу	64
3.3 Опис змінних моделі	65
3.4 Принцип формування змінної “тижнева зміна ціни”	68
3.5 Послідовність проведення аналізу	69
3.6 Опис алгоритму в загальному вигляді по кроках	72
3.7 Графіки зміни значень коефіцієнтів моделі, по ітераціях.	77
3.8 Аналіз отриманих результатів	81

	8
3.9 Розширена модель логістичної регресії із додатковими ознаками.	83
3.9.1 Змінна сентимент аналіз криптоактиву	83
3.9.2 Текстова ознака - категорія криптоактиву, що аналізується	85
3.9.3 Побудова розширеної моделі	86
Висновки до розділу 3	91
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	93
4.1 Постановка задачі проектування	94
4.2 Обґрунтування функцій програмного продукту	94
4.3 Обґрунтування системи параметрів програмного продукту	98
4.4 Аналіз експертного оцінювання параметрів	101
4.5 Аналіз рівня якості варіантів реалізації функцій	105
4.6 Економічний аналіз варіантів розробки програмного продукту	106
4.7 Вибір кращого варіанту програмного продукту	
техніко-економічного рівня	112
Висновки до розділу 4	113
ВИСНОВКИ	115
ПЕРЕЛІК ПОСИЛАНЬ	117
ДОДАТОК А ЛІСТИНГ КОДУ	121
ДОДАТОК Б ОПИС ПРИНАЛЕЖНОСТІ ПРОЕКТІВ ДО КАТЕГОРІЇ	124
ДОДАТОК В ЛІСТИНГ КОДУ ПРОМПТУ ДЛЯ СЕНТИМЕНТ АНАЛІЗУ	128
ДОДАТОК Д АКТ ВПРОВАДЖЕННЯ	130
ДОДАТОК Е МАТЕРІАЛ ТЕЗ	132

ПЕРЕЛІК СКОРОЧЕНЬ

МНК – метод найменших квадратів, статистичний підхід до оцінювання параметрів регресійної моделі шляхом мінімізації суми квадратів відхилень спостережуваних значень від прогнозованих

ФВА – метод оптимізації вартості продукції або процесу шляхом системного аналізу функцій, які вони виконують, із метою підвищення їх ефективності за найнижчої ціни

ARIMA – AutoRegressive Integrated Moving Average, модель Бокса-Дженкінса

BNB – токен біржі Binance, який застосовується для оплати комісій та отримання різноманітних переваг у межах платформи

BTC – це децентралізована цифрова валюта (криптовалюта), що працює на основі блокчейну без централізованого контролю

DeFi – екосистема фінансових сервісів на базі блокчейну, яка працює без посередників і дозволяє здійснювати кредитування, обмін та інші операції децентралізовано

DW – статистика Дарбіна-Вотсона, яка використовується для перевірки наявності автокореляції в залишках регресійної моделі

FDV – це теоретична ринкова капіталізація токена за умови, що на ринку вже випущений весь його максимальний обсяг. Обчислюється як поточна ціна токена, помножена на загальну кількість tokenів

GARCH – Generalised AutoRegressive Conditional Heteroskedasticity

GUI – Graphic User Interface

KYC/AML-скринінг – це процес перевірки клієнтів фінансової установи на відповідність вимогам «Знай свого клієнта» (KYC) та протидії відмиванню грошей і фінансуванню тероризму (AML)

LLM – це велика мовна модель на основі нейронних мереж, навчена на величезних обсягах тексту для розуміння і генерації природної мови

LSTM – Long Short-Term Memory – це тип рекурентної нейронної мережі (RNN), спеціально розроблений для обробки послідовних даних та довгострокових залежностей у часі

MAE – середня абсолютна помилка

Maximum Drawdown – максимальна просадка

MiCA – регламент ЄС «Markets in Crypto-Assets», що встановлює правову основу для випуску, торгівлі та обігу криптоактивів у межах Європейського Союзу

ML – машинне навчання

MSE – середньоквадратична помилка

MSE – середньоквадратична помилка (Mean Squared Error), яка вимірює середню величину квадрата різниці між прогнозованими та фактичними значеннями

NLP – Natural Language Processing

OI – Open Interest, відкритий інтерес

OKX – міжнародна криптовалютна біржа, що надає трейдерам доступ до торгівлі цифровими активами, ф'ючерсами та деривативами

One-hot Encoding – метод перетворення категоріальних змінних у бінарні вектори, де кожна категорія представлена окремою ознакою (стовпцем) зі значенням 1 або 0

Prompt engineering – процес розробки та оптимізації текстових запитів (prompts) для ефективної взаємодії з великими мовними моделями (LLM), щоб отримати точні, корисні або креативні відповіді

Prompt – набір вказівок для ChatGpt

RMSE – корінь із середньоквадратичної помилки (Root Mean Squared Error), що показує середню величину похибки прогнозу в тих же одиницях вимірювання, що й вихідні дані

RNN – Recurrent Neural Network, це тип нейронної мережі, спеціально розроблений для роботи з послідовними даними, де порядок елементів має значення (наприклад, текст, аудіо, часові ряди)

Sharpe Ratio – коефіцієнт Шарпа

SMA – Simple Moving Average

Target variable – цільова змінна

Total Profit – загальний прибуток

USD – офіційна валюта США, одна з провідних резервних валют у світі

USDT – стейблкоїн від компанії Tether, прив'язаний до курсу долара США 1:1, який використовується для зниження волатильності при торгівлі криптовалютами

Whitepaper – документ, у якому докладно описуються технологічні, економічні та технічні аспекти криптопроєкту

XGBoost – eXtreme Gradient Boosting, градієнтний бустинг

ВСТУП

Криптовалюти¹ останнім часом перетворилися на один із ключових фінансових інструментів глобальної економіки. За останні 5 років, кількість нових проєктів та криптовалют суттєво збільшилася [11]. Як наслідок, виникла необхідність із розробки інструменту, який зможе прогнозувати чи буде певна криптовалюта додана до лістингу на провідних крипто-біржах чи ні. Такий інструмент зможе надавати корисну інформацію, у вигляді ймовірнісних прогнозів, необхідних для торгівлі на біржах та у децентралізованих додатках.

Методи прогнозування лістингу включають фундаментальний аналіз (оцінка команди, фінансування, партнерства), регресійні підходи (зокрема логістичну регресію), ансамблеві моделі (Random Forest, Gradient Boosting), аналіз часових рядів (ARIMA, LSTM), а також інструменти обробки природної мови (NLP) і тональності тексту (Sentiment Analysis), для вилучення характеристик із неструктурованих текстових джерел. Завдяки поєднанню можливостей машинного навчання та глибоких нейронних мереж ці методи здатні опрацьовувати великі обсяги різноманітних даних, виявляти складні залежності та швидко адаптуватися до змін ринкової кон'юнктури, що значно підвищує точність і надійність прогнозів ймовірності лістингу нових криптовалют.

У цій роботі розглянуто різноманітні підходи щодо прогнозування ймовірності лістингу нових криптовалют – від класичної логістичної регресії до сучасних методів машинного та глибокого навчання. В дипломній роботі описано основний принцип роботи, алгоритмічні особливості та вимоги до

¹Тут і нижче використаний такий інструмент штучного інтелекту як чат-бот з генеративним штучним інтелектом ChatGPT, виключно для корегування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПП ім. Ігоря Сікорського (протокол №11 Вченої ради КПП ім. Ігоря Сікорського від 11 грудня 2023 р.).

вхідних даних. Окрім цього наведено практичний приклад застосування, оцінена ефективність та описана перспективність розвитку цього дослідження.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність задачі прогнозування лістингу нових криптовалют

Кожний день з'являються нові криптовалюти, деякі з них мають значний потенціал та стануть широко відомими та дорогими, а деякі лише забезпечать втрату коштів інвесторів.

Прогнозування ймовірності лістингу нових криптовалют є актуальною задачею.

1. Зростання кількості криптовалют (за останній рік кількість доступних на ринку криптоактивів зросла в декілька разів: окрім «класичних» Bitcoin і Ethereum, з'явилися тисячі нових токенів, що представляють проекти у різних криптосферах [11]. Таке різноманіття дозволяє залучати інвесторів із найрізноманітнішими інтересами й апетитами до ризику, але одночасно ускладнює процес відбору найбільш перспективних активів).
2. Вплив лістингів на ціну криптовалют. Коли криптовалюта з'являється на провідній біржі (наприклад, Binance або Coinbase [9]), це часто призводить до різкого зростання її ліквідності та сплеску торгового інтересу зі сторони крипто-спільноти. Перші години після лістингу характеризуються високою волатильністю: інвестори прагнуть швидко купити перспективний актив, що може підштовхнути ціну вгору, а пізніше, за відсутності достатнього обсягу реальних покупців, – привести до корекції. У будь-якому разі вдале прогнозування лістингу відкриває можливість для короткострокових та середньострокових стратегій торгівлі.

3. Аналіз надійності певної криптовалюти. Відсутність перспективи лістингу на великих біржах може свідчити про низький рівень довіри до проекту з боку ринку: недостатню прозорість команди, слабку технічну реалізацію, відсутність партнерств або низький інтерес спільноти. Тому прогнозування ймовірності лістингу працює як фільтр для інвесторів: ті токени, які у моделі отримують високий бал шансів на лістинг, зазвичай мають більш стабільні фундаментальні показники і кращі шанси на довгостроковий успіх. Таке інструментальне оцінювання допомагає знизити ризики та відсіяти безперспективні криптовалюти.

1.2 Опис предметної області

Прогнозування факту лістингу нових криптовалют потребує залучення великої кількості різних факторів та застосування нових методів аналізу даних. Основними методами які можна використовувати для цієї задачі є:

- логістична регресія: дозволяє кількісно оцінити вплив кожного фактору на ймовірність лістингу та інтерпретувати отримані коефіцієнти для виявлення найбільш значущих змінних;
- аналіз часових рядів (ARIMA та LSTM): враховує часову залежність ринкових індикаторів, таких як обсяги торгів і волатильність, та допомагає передбачити майбутні тренди, які можуть сигналізувати про підготовку лістингу;

- методи обробки природної мови (NLP) та аналіз тональності тексту (Sentiment Analysis): застосовуються для автоматизованого збору та перетворення текстових даних з новинних ресурсів і соцмереж у цифрові ознаки, які відображають емоційний фон і суспільний інтерес до проекту.

Практичне використання цих методів потребує залучення моделей у торгову систему для своєчасного прийняття рішення про інвестицію у певну криптовалюту.

1.3 Лістинг криптовалют

Лістинг криптовалюти на біржі це процес додавання нового криптоактиву до списку доступних для торгівлі на певній торговельній платформі [12]. Процедура лістингу передбачає подання заявки, технічний аудит проекту, перевірку відповідності юридичним та регуляторним вимогам, а також оцінку його загального потенціалу й надійності. Лістинг часто супроводжується зростанням торгового обсягу та волатильності ціни.

1.3.1 Процес лістинг криптовалют

Процес лістингу криптовалют зазвичай включає кілька основних етапів: подання проектом офіційної заявки, детальний аналіз криптовалюти фахівцями біржі (технічна експертиза, оцінка безпеки, фінансова та юридична перевірка), а також узгодження та укладання договору між біржею і командою проекту. Остаточне рішення про лістинг ухвалюється на основі

відповідності криптовалюти встановленим критеріям платформи, таким як технічна надійність, прозорість структури проекту, відповідність нормативним вимогам та наявність активної спільноти користувачів [13].

1.3.2 Типи криптобірж

У сегменті централізованих бірж [2] можна виокремити дві умовні категорії за розміром та впливом на ринок.

По-перше, це невеликі біржі: на них з'явлення нового токена майже не позначається на його ціні через низьку ліквідність та обмежений потік трейдерів. Такі біржі часто спеціалізуються на локальних чи тематичних спільнотах і служать для попереднього тестування проектів.

Подруге, це глобальні гіганти, такі як Binance та Coinbase [9]: їхній лістинг здатен викликати миттєвий сплеск попиту та волатильності, оскільки на цих платформах оперує сотні тисяч активних користувачів і великі маркетмейкери. Саме на таких біржах лістинг відіграє ключову роль у загальній динаміці ціни та ліквідності нового токена.

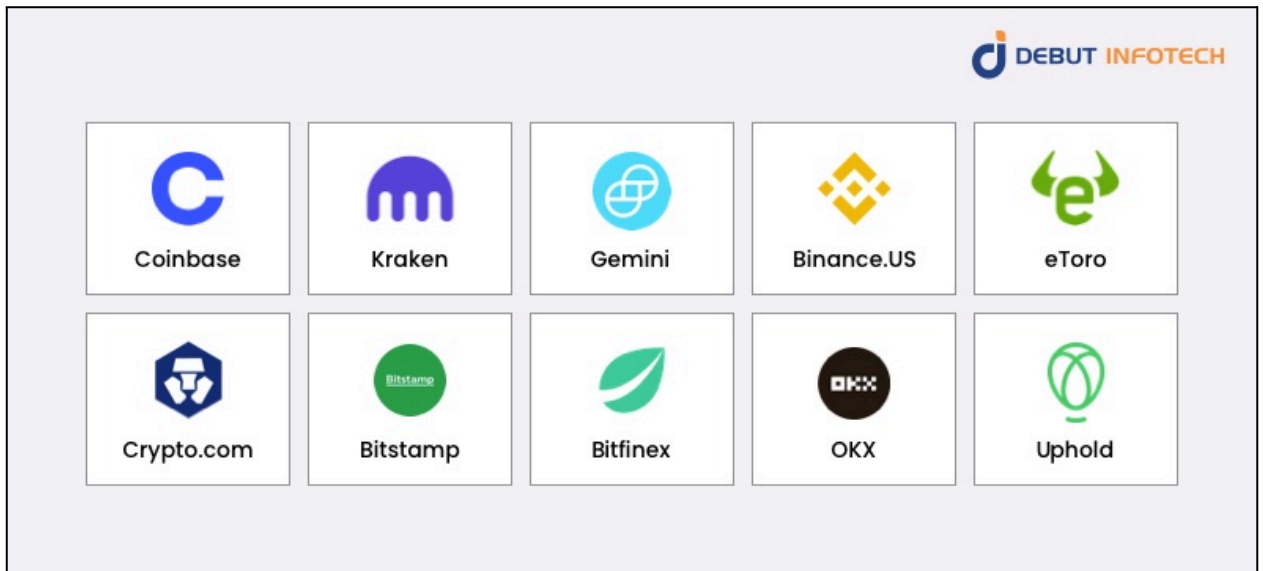


Рисунок 1.1 – Перелік популярних криптобірж [26]

1.3.3 Які фактори впливають на прийняття рішення про лістинг

Рішення про лістинг нового токена на централізованій біржі ґрунтується на сукупності різноманітних факторів. Серед них: команда проекту, яка стоїть за криптоактивом, а саме її досвід та репутація, ринкові показники активу (ліквідність, обсяг торгів, капіталізація), токеноміка проекту (графіки розблокування токенів, концентрація активів серед інсайдерів), інноваційність та відповідність трендам ринку (зазвичай перевага надається проектам, що відповідають актуальним трендам) [14].

1.4 Класифікація криптоактивів

Існує дуже велика кількість різних криптоактивів, більшість з них можна віднести до однієї з наступних категорій:

Таблиця 1.1 – Категорії криптоактивів

Назва категорії	Опис категорії
Токени протоколів першого рівня (Layer 1 Protocol Tokens)	Криптовалюти, які функціонують на власних блокчейнах, забезпечуючи базовий рівень для створення децентралізованих додатків та екосистем
Токени протоколів другого рівня та інфраструктурних рішень (Layer 2 / Scaling / Infrastructure Tokens)	Токени, що використовуються в рішеннях для підвищення швидкості, масштабованості та зниження комісій на базових блокчейнах, таких як Ethereum.
Дефі токени (DeFi Tokens)	Токени, пов'язані з проєктами децентралізованих фінансів, які дають користувачам можливість доступу до фінансових послуг без централізованих посередників
Екосистемні токени (Utility & Ecosystem Tokens)	Токени, призначені для використання всередині конкретної платформи або екосистеми, забезпечуючи доступ до послуг, продуктів чи функцій конкретних проєктів
Стейблкоїни (Stablecoins & Asset-backed Tokens)	Це токени, вартість яких прив'язана до стабільних активів, таких як долар США, золото або інші активи, з метою зменшення волатильності та забезпечення стабільності цін

1.4.1 Токени протоколів першого рівня

Токени протоколів першого рівня (Layer 1 Protocol Tokens) – це криптовалюти, які функціонують на власних незалежних блокчейн-мережах, що виступають фундаментом для побудови різноманітних децентралізованих додатків, екосистем та інших сервісів.

Основною роллю таких tokenів є забезпечення роботи ключових механізмів блокчейну, включаючи досягнення консенсусу, виконання та розгортання смарт-контрактів, а також оплату комісій за транзакції в мережі. Layer 1 токени часто є центральною одиницею цінності в мережі та використовуються для винагороди учасників, які підтримують її стабільність і безпеку (наприклад, майнерів чи валідаторів). До популярних Layer 1 tokenів належать Ethereum (ETH), Solana (SOL), Polkadot (DOT), Cardano (ADA) і Avalanche (AVAX).

1.4.2 Токени протоколів другого рівня

Токени протоколів другого рівня та інфраструктурних рішень (Layer 2 / Scaling / Infrastructure Tokens) – це криптовалюти блокчейнів, що покращують продуктивність та ефективність базових блокчейн-мереж (першого рівня). Вони призначені для підвищення швидкості транзакцій, зменшення витрат та вирішення проблем масштабованості мереж, таких як Ethereum.

Ці рішення зазвичай включають сайдчейни, ролапи (rollups) та інші технології, які дозволяють проводити транзакції за межами основного ланцюга з подальшим підтвердженням на головній мережі [16]. Прикладами таких tokenів є Polygon (MATIC), Optimism (OP) та Arbitrum (ARB).

1.4.3 Дефі токени

Дефі токени (DeFi Tokens) – криптовалюти децентралізованих фінансових протоколів, які забезпечують доступ до фінансових послуг без участі традиційних посередників, таких як банки чи брокери. DeFi-токени можуть виконувати різні функції: слугувати засобом обміну, використовуватись для управління протоколом (через механізми голосування), надавати доступ до певних сервісів або виступати як забезпечення для позик [17]. Прикладами таких токенів є Uniswap (UNI), Aave (AAVE), Maker (MKR) та Compound (COMP).

1.4.4 Екосистемні токени

Екосистемні токени (Utility and Ecosystem Tokens) – це токени, які створені для використання всередині певної платформи або блокчейн-екосистеми. Їх основна функція – надавати користувачам доступ до продуктів, сервісів або функцій проекту. Наприклад, за допомогою такого токена можна оплачувати комісії за транзакції, отримувати знижки, голосувати за зміни в мережі або брати участь в екосистемних ініціативах [18].

Ці токени часто не претендують на роль загального платіжного засобу, як Bitcoin або Ethereum, але відіграють важливу роль у функціонуванні конкретного проекту. Наприклад, токен Binance Coin (BNB) використовується на біржі Binance для оплати комісій та участі в запуску нових токенів (Launchpad), а токен Chainlink (LINK) слугує для оплати послуг оракулів, що

постачають зовнішні дані в смарт-контракти. Такі токени формують економіку проекту та стимулюють активність користувачів.

1.4.5 Стейблкоїни

Стейблкоїни (Stablecoins and Asset-backed Tokens) – це цифрові активи, чия вартість підтримується резервами у реальних активах або алгоритмічною прив'язкою, завдяки чому вони зберігають стабільну ціну порівняно з волатильними криптовалютами.

Стейблкоїни тримають паритет (наприклад 1:1 до долара США) за рахунок фіатних резервів, товарів чи інших форм забезпечення, тоді як asset-backed токени оцифровують частку у конкретному фізичному чи фінансовому активі – від золота та нерухомості до акцій – і дають змогу торгувати цими активами на блокчейні, підвищуючи їхню ліквідність і доступність [19]. Прикладами таких токенів є USDT та USDC.

1.5 Вплив лістингу на цінову динаміку криптовалют

Лістинг криптовалюти на криптобіржі надає трейдерам можливість торгувати новим активом, тому лістинг майже завжди супроводжується великою волатильністю та зміненням ціни активу [4]. Проте лістинг на різних біржах по різному впливає на ціну активу.

1.5.1 Вплив лістингу на біржі Binance на ціну криптовалюти

Лістинг на біржі Binance часто призводить до значного зростання ліквідності та торгового інтересу: поява нової криптовалюти привертає велику кількість трейдерів і маркет-мейкерів [5]. У перші години після лістингу спостерігається підвищена волатильність, ціна може різко вирости через ажіотажний попит, а потім відкоригуватися до більш стабільного рівня. Такий ефект створює можливості для короткострокових прибутків, але водночас вимагає уважного управління ризиками.



Рисунок 1.2 – Приклад впливу оголошення лістингу активу TST на біржу Binance на його ціну [27]

1.5.2 Вплив лістингу на інших біржах на ціну криптовалюти

Лістинг на інших великих біржах, таких як Coinbase та OKX [15] схожим чином впливає на ціну активу. Лістинг на невеликих біржах також привертає додаткову увагу, але його ефект на ціну зазвичай менш виразний через обмежені обсяги торгів та нижчу ліквідність. Поява нової пари на локальному майданчику може стимулювати помірне зростання ціни серед невеликої групи трейдерів, але без масштабного маркет-мейкінгу та широкого охоплення аудиторії цей імпульс швидко згасає. Водночас регулярні лістинги на декількох середніх за розміром біржах поступово підвищують популярність активу, створюючи більш стійкий попит і сприяючи повільному, але тривалому зростанню вартості.

1.6 Показники які впливають на рішення про лістинг криптоактиву

На прийняття рішення про лістинг впливає велика кількість показників. Перед лістингом команда криптобіржи ретельно перевіряє криптоактив та команду проекту.

Існує чотири головні категорії показників які мають суттєвий вплив.

Таблиця 1.2 – Категорії показників, що впливають на прийняття рішення про лістинг

Назва категорії показників	Короткий опис категорії
1) Продукт і технологія проекту	Дорожня карта, Whitepaper, наявність продукту, масштабованість та безпека проекту
2) Команда і екосистема	Досвід команди, партнери, інвестори, спільнота проекту
3) Токеноміка і ринкова динаміка	Розподіл токенів, графік розблокування, ліквідність, обсяг торгів
4) Юридична та безпекова відповідність	Статус активу, результати аудитів, відповідність нормам регуляторів

1.6.1 Продукт і технологія проекту

Ця категорія охоплює все, що свідчить про технічну зрілість і реальну працездатність криптоактиву. Біржа насамперед уважно читає whitepaper [20] та дорожню карту, але цього недостатньо: команда має показати мінімально життєздатний продукт або тестову мережу, аби довести, що проект уже працює, а не існує лише у презентації. Паралельно проводиться технічний аудит: аналіз коду смарт-контрактів, перевірка історії зовнішніх аудитів, оцінка складності системи, ризику централізації та попередніх інцидентів безпеки.

Для біржі це критично, оскільки відсутність вразливостей у коді й наявність чітких механізмів оновлень гарантує, що після лістингу актив не стане джерелом хакерських атак чи втрати коштів користувачів. Окремо оцінюються показники продуктивності продукту і відповідність стандартам токенів, адже продукт повинен бути готовим до підвищених торгових навантажень.

1.6.2 Команда і екосистема

Категорія “Команда і екосистема” охоплює людський фактор і соціальне середовище проекту, які біржа розглядає як показник його життєздатності. Насамперед перевіряється бекграунд команди: публічність засновників, досвід у крипто-індустрії, історія успішних (або провальних) проектів і те, як розробники комунікують із користувачами у складних ситуаціях. Binance, наприклад, прямо зазначає, що уважно спостерігає за тим, «як команда поводить ся зі своєю спільнотою» та які метрики залученості може підтвердити у заявці на лістинг [21].

Окрему увагу приділяють «соціальному капіталу» – органічному зростанню спільноти у Twitter та Discord, а також наявності стратегічних партнерів або децентралізованих застосунків, уже побудованих навколо токена. Актуальні вимоги Binance прямо вимагають надати докладні дані про команду, продукт і спільноту, аби біржа могла оцінити силу мережевого ефекту та довіру користувачів.

1.6.3 Токеноміка і ринкова динаміка

Ця категорія стосується того, наскільки токеноміка [22] та економічна модель токена є життєздатною і чи існує вже реальний попит на ринку. Біржа насамперед дивиться, як розподілений і розблоковується запас монет: чим прозоріший графік емісії й менша концентрація у засновників, тим нижчий ризик раптових викидів на ринок.

Далі оцінюється фактична ліквідність: проект має показати нетривіальні обсяги торгів і глибину ордербуків на інших майданчиках, бо низька ліквідність або сильна волатильність означають проблеми з формуванням ціни після лістингу. Не менш важливі показники ринкової капіталізації та fully-diluted valuation (повністю розведена оцінка) – якщо FDV виглядає завищеним порівняно з фундаментальними метриками, біржа може відкласти лістинг, аби уникнути «надутих» активів. Також важливими показниками є обсяг торгів та динаміка зміни ціни активу.

1.6.4 Юридична та безпекова відповідальність

Юридична та безпекова відповідальність – це та категорія вимог, яку біржа застосовує, аби переконатися, що токен не створить їй регуляторних або технічних проблем уже після лістингу. Передусім проект проходить ретельний KYC/AML-скринінг: біржі, звіряють дані засновників та інвесторів із глобальними санкційними списками. Юридичні відділи перевіряють, чи може токен вважатися цінним папером. У ЄС на додаток діятимуть суворі стандарти MiCA [23].

Безпекова частина включає аудити смарт-контрактів та перевірку технічної інфраструктури: Біржі зобов'язують команди довести, що код не містить критичних вразливостей і здатен безпечно зберігати токени клієнтів. Сукупно ці процедури формують «юридичний та безпековий бар'єр»; якщо проект його не проходить, біржа ризикує штрафами, репутаційними втратами та, що найгірше, прямими проблемами у роботі платформ через зламані смарт-контракти чи заморожені активи.

1.7 Прогнозування факту лістингу

Прогнозування факту лістингу нових криптовалют на біржі Binance складний процес, який має велику кількість етапів, таких як збір та аналіз даних, побудова та аналіз моделі та валідація результатів. Серед основних підходів – застосування логістичної регресії для кількісної оцінки впливу фундаментальних, ринкових та інформаційних факторів. Побудова та тестування моделей на історичних даних Binance дозволяє не лише оцінити їхню ефективність, але й адаптувати їх до реальних умов ринку для оперативної підтримки інвестиційних рішень.

1.7.1 Використання логістичної регресії

Логістична регресія дозволяє кількісно оцінити вплив ключових факторів на ймовірність лістингу криптовалюти на Binance. Модель обчислює ймовірність появи токена в переліку торгів за допомогою логістичної функції, що перетворює суму зважених ознак у значення від 0 до 1. Завдяки простоті інтерпретації коефіцієнтів та високій швидкості навчання логістична регресія є ефективним інструментом для первинного аналізу й підтримки інвестиційних рішень.

1.7.2 Обробка показників

Першим кроком у підготовці даних для прогнозування лістингу є збирання та очищення вхідних показників: видалення дублікатів, заповнення або виключення пропущених значень і перевірка на аномалії в обсягах торгів та цінових даних. Далі числові змінні (ліквідність, середні обсяги, волатильність) нормалізуються або стандартизуються, щоб уникнути домінування великих масштабів над менш вираженими ознаками. Текстові дані, такі як опис проекту обробляються за допомогою сентимент аналізу [6] за допомогою LLM моделі. За потреби застосовується відбір ознак або зведення розмірності, щоб скоротити шум і підвищити стабільність моделі логістичної регресії.

1.7.3 Побудова та тестування моделей

Після підготовки ознак на основі фундаментальних, ринкових та інформаційних даних, моделі логістичної регресії навчаються і тестуються на історичних випадках лістингів з біржі Binance. Це дає змогу оцінити їхню здатність точно передбачати появу нових токенів. Отже, поєднання логістичної регресії і сентимент аналізу забезпечує всебічний підхід до прогнозування лістингу, а результати валідації на реальних даних підтверджують практичну корисність створених моделей.

1.8 Постановка задачі

1. Виявлення ключових факторів, що впливають на прийняття рішення про лістинг активу та вибір моделей для прогнозування.
2. Розробка системи збору та обробки даних з різних джерел для подальшого аналізу.
3. Створення моделей для прогнозування факту лістингу криптоактиву використовуючи зібрані дані.
4. Розробка програми для аналізу та візуалізації результатів прогнозування та побудови метрик.

Висновки до розділу 1

У даному розділі було розглянуто актуальність прогнозування лістингу нових криптовалют на біржі Binance, окреслено предметну область та визначено ключові методи аналізу та прогнозування, серед яких логістична регресія, аналіз часових рядів, обробка природної мови і ансамблеві алгоритми машинного навчання. Застосування цих підходів дозволяє підвищити точність оцінки ймовірності лістингу, забезпечити оперативну реакцію на зміни ринкових та інформаційних сигналів і знизити інвестиційні ризики, пов'язані з вибором нових активів.

Також, було розглянуто особливості лістингу криптовалют на централізованих біржах та його вплив на ринок криптовалют. Зокрема, описано процес додавання токенів на біржі, показники, що впливають на прийняття рішення про лістинг, а також ефекти лістингу на біржах різного масштабу, таких як Binance та менші локальні біржі. Також, було розглянуто, на які категорії поділяються усі криптоактиви та детально описано кожен з

цих категорій. Крім того, розглянуто методи прогнозування факту лістингу з використанням логістичної регресії та обробки даних. Застосування цих методів дає змогу точніше оцінити ймовірність появи токенів на біржах і завчасно реагувати на зміни ринкових умов, підвищуючи ефективність інвестиційних рішень у криптовалютній сфері.

РОЗДІЛ 2 МЕТОДИКА ПРОГНОЗУВАННЯ

2.1 Поняття структури математичної моделі авторегресії

Поняття структури моделі містить у собі такі параметри.

Порядок моделі, тобто порядок диференціального, різницевого чи іншого рівняння, що використовується для опису динаміки процесу чи об'єкта [28]. Наприклад, стохастичне різницеве авторегресійне (АР) рівняння другого порядку має вигляд:

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2) + \varepsilon(k)$$

Тобто, порядок цього різницевого рівняння визначається числом затриманих у часі значень змінної, що використовуються у правій частині рівняння для опису змінної в лівій частині. Стохастичним воно називається тому, що у правій частині присутня випадкова змінна $\varepsilon(k)$, призначення якої ми пояснимо трохи нижче [28].

Вимірність моделі. Вона визначається числом рівнянь, що використовуються для опису об'єкта чи процесу. Процес, котрий описують одним рівнянням, називають одномірним чи скалярним. Процес, котрий описують двома і більш рівняннями, називають багатовимірним [28].

Наявність нелінійностей і їх характер. Визначити наявність нелінійностей - не завжди проста задача. Так, для механічних і деяких інших систем наявність нелінійностей можна визначити шляхом вивчення законів, закономірностей і особливостей їхнього функціонування [28]. Наприклад, відомо, що для механічних систем характерною є наявність нелінійностей типу «люфт», «тертя», гістерезис.

При побудові регресійних моделей частіше зустрічаються нелінійності відносно змінних і нелінійності відносно параметрів [28]. Прикладом нелінійності відносно змінних може бути поліноміальна регресія виду:

$$y(k) = a_0 + a_1 x(k) + a_2 x^2(k) + a_3 x^3(k) + \varepsilon(k)$$

Коефіцієнти цього рівняння можна оцінювати звичайним методом найменших квадратів (МНК) при належній побудові матриці вимірювань (вона буде розглянута нижче) [28]. Нелінійність відносно параметрів зумовлена наявністю в моделі добутоків коефіцієнтів, наприклад, у вигляді:

$$y(k) = a_0 + a_1 a_2 x(k) + a_2 \exp(-bx(k)) + \varepsilon(k)$$

Коефіцієнти (параметри) такої моделі неможливо оцінити за допомогою звичайного МНК, тому для розв'язання цієї задачі використовують нелінійний МНК, метод максимальної правдоподібності чи інші методи нелінійного оцінювання [28].

Час запізнення реакції на виході об'єкта відносно вхідного сигналу. Запізнювання по входу (лаг) досить легко враховується як у неперервних, так і в дискретних моделях [28]. Для моделі з дискретними змінними у вигляді різницевого рівняння:

$$y(k) = a_0 + a_1 y(k-1) + a_2 x(k-d) + \varepsilon(k),$$

де d – час запізнення, що представляє ціле число, що дорівнює числу періодів дискретизації вимірювань, на яке запізнюється вихідний сигнал щодо вхідного [28].

Тривалість періоду дискретизації вимірювань залежить від динаміки конкретного процесу і може змінюватися в межах від декількох мікросекунд у фізико-технічних системах до одного року в макроекономіці [28].

Розглянемо модель неперервного процесу у вигляді передаточної функції із запізненням:

$$W(p) = \frac{K e^{-p\tau}}{1 + Tp},$$

де K – статичний коефіцієнт передачі об'єкта;

p – змінна Лапласа;

T – постійна часу;

τ – час запізнення по входу.

Запізнювання в дискретній формі d і запізнення в неперервній формі τ зв'язані між собою наступним чином:

$$d = \text{int}(\tau / T_s).$$

Тип збурень, що діють на процес, і спосіб їх врахування [28]. Під збуреннями розуміють вхідні впливи процесу, котрі мають, як правило, негативний вплив на його протікання, але не використовуються із тих чи інших причин як керуючі. Збурення поділяють на детерміновані і стохастичні, а враховуються вони в адитивній чи мультиплікативній формі. Вище ми привели різницеві рівняння, у яких збурення $\varepsilon(k)$ входить в адитивній формі [28].

Приклад мультиплікативної форми:

$$h(k) = v(k)[\alpha_0 + \alpha_1 h(k-1)],$$

де $v(k)$ – мультиплікативне збурювання.

Вибір структури моделі, що адекватна процесу, – задача не проста і вирішується, як правило, ітераційно. Спочатку структуру моделі оцінюють наближено на підставі дослідження закономірностей протікання процесу, аналізу кореляційних функцій, візуального аналізу даних. При цьому

вибирають декілька найбільш ймовірних структур (кандидатів). Потім визначають оцінки параметрів моделей-кандидатів і вибирають кращу з них, використовуючи відповідні статистичні характеристики моделей [28].

Якщо жодна з моделей-кандидатів не може вважатися адекватною, то необхідно досліджувати на інформативність експериментальні дані, які можуть бути недостатньо інформативними для оцінювання моделі. У такому випадку може знадобитися повторний чи додатковий збір експериментальних даних [28].

2.2 Методи регресійного аналізу

Регресійний аналіз (лінійний) – являє собою набір статистичних методів дослідження залежності між залежною змінною Y і однією або декількома незалежними змінними $X_1, X_2, \dots, X_N$. Незалежні змінні інакше називають регресорами або предикторами, а залежні змінні – вихідними (от кліку, критеріальними Y -змінними) [29].

Регресійний аналіз має мету.

1. Визначення наявності і характеру (математичного рівняння, що описує залежність) зв'язку між змінними;
2. Визначення ступеня детермінованості варіації критеріальної змінної предикторами;
3. Прогнозування значення залежної змінної за допомогою незалежної;
4. Визначення впливу незалежних змінних у варіацію залежної.

2.2.1 Метод найменших квадратів

Введемо в цьому підрозділі наступні позначення [29]

Y – вектор значень вимірів.

y – вимір (залежна змінна / вихідний сигнал / ендогенна змінна).

$\tilde{y}$ – реальне значення ряду без шуму.

$\hat{y}$ – прогнозне значення виміру (обчислене по математичній моделі).

ξ – шум виміру (вважається білим).

H – матриця вимірів.

β – вектор коефіцієнтів математичної моделі.

β^0 – істинне значення.

$\hat{\beta}_m$ – вектор оцінок коефіцієнтів, обчислений по m вимірам.

$$M[\xi] = 0$$

$$M[\xi \cdot \xi^T] = \delta^2 \cdot I$$

Ідея метода МНК полягає в мінімізації функціоналу:

$$\sum (y_i - \hat{y}_i)^2 \rightarrow \min$$

2.2.2 Простий приклад

Нехай процес описується лінійним рівнянням:

Для мінімізації функціоналу даного виразу, треба знайти точку екстремуму, тобто перша похідна по β (вектор коефіцієнтів) повинна дорівнювати нулю [29].

2.2.3 Геометричний зміст метода найменших квадратів

Розглянемо даний вираз:

$$Q = SSD = \sum (y_i - \hat{y}_i)^2 .$$

Графічно задачу, що вирішується, можна представити як показано на рис. 2.1.

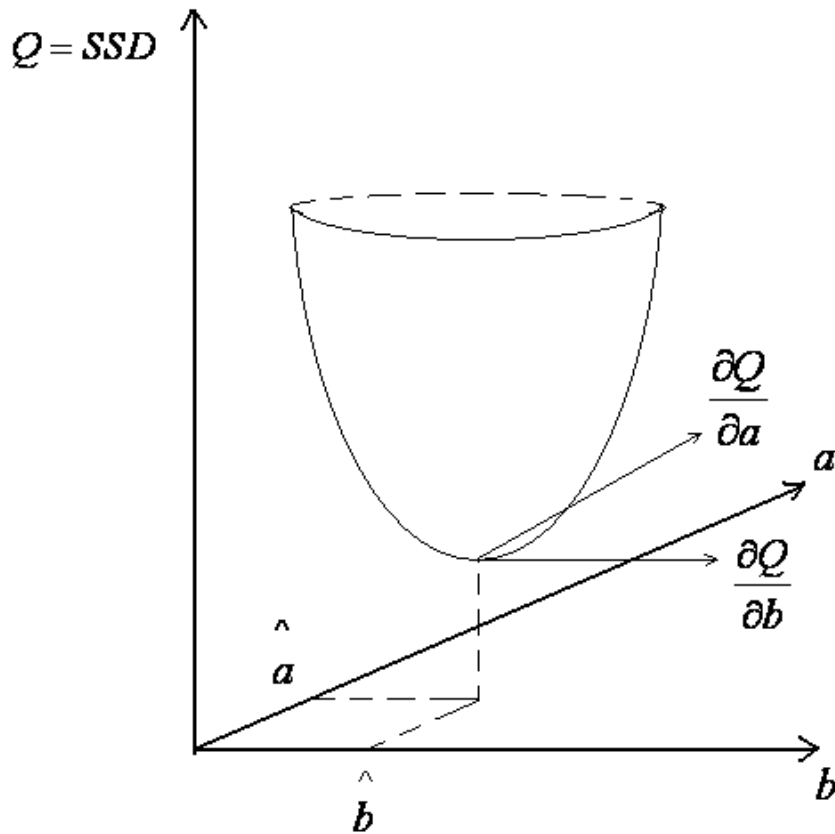


Рисунок 2.1 – Геометричний зміст методу МНК

2.2.4 Вивід методу найменших квадратів

$$\begin{aligned}
 \frac{\partial SSD}{\partial \beta} &= \frac{\left[Y - H \cdot \beta \right]^T \cdot \left[Y - H \cdot \beta \right]}{\partial \beta} = \\
 &= \frac{\partial \left(Y^T \cdot Y - Y^T \cdot H \cdot \beta - \beta^T \cdot H^T \cdot Y + \beta^T \cdot H^T \cdot H \cdot \beta \right)}{\partial \beta} = \\
 &= -H^T \cdot Y - H^T \cdot Y + 2 \cdot H^T \cdot H \cdot \beta = 0 \\
 &\quad -H^T \cdot Y - H^T \cdot Y + 2 \cdot H^T \cdot H \cdot \beta = 0 \\
 &\quad -2 \cdot H^T \cdot Y = -2 \cdot H^T \cdot H \cdot \beta
 \end{aligned}$$

$$H^T \cdot Y = H^T \cdot H \cdot \beta$$

$$\hat{\beta} = (H^T \cdot H)^{-1} \cdot H^T \cdot Y.$$

Так як спочатку в наявності є m вимірів, то отримане рівняння можна представити у вигляді:

$$\hat{\beta}_m = (H^T \cdot H)^{-1} \cdot H^T \cdot Y,$$

де запис $\hat{\beta}_m$ означає вектор оцінок коефіцієнтів, обчислення m вимірам.

Через β^0 позначається точне значення вектору коефіцієнтів, тобто:

$$Y = H \cdot \beta^0 + \xi$$

$$\begin{aligned} \hat{\beta}_m &= (H^T \cdot H)^{-1} \cdot H^T \cdot Y = \\ &= (H^T \cdot H)^{-1} \cdot H^T \cdot \left(H \cdot \beta_m^0 + \xi \right) = \\ &= (H^T \cdot H)^{-1} \cdot H^T \cdot H \cdot \beta_m^0 + (H^T \cdot H)^{-1} \cdot H^T \cdot \xi = \\ &= I \cdot \beta_m^0 + (H^T \cdot H)^{-1} \cdot H^T \cdot \xi = \\ &= \beta_m^0 + (H^T \cdot H)^{-1} \cdot H^T \cdot \xi \end{aligned}$$

При $m \rightarrow \infty$ виконується наступне:

$$\left| \hat{\beta}_m - \beta_m^0 \right| \rightarrow 0.$$

Тобто зі збільшенням кількості вимірів оцінка коефіцієнта наближається до реального значення [29].

2.2.5 Незміщеність оцінки вектору коефіцієнтів

Доведемо незміщеність оцінки вектору коефіцієнтів $\hat{\beta}_m$ відносно β_m^0 , для цього обчислимо математичне сподівання $\hat{\beta}_m$:

$$\begin{aligned}\hat{\beta}_m &= \beta_m^0 + (H^T \cdot H)^{-1} \cdot H^T \cdot \xi \\ M\left[\hat{\beta}_m\right] &= M\left[\beta_m^0 + (H^T \cdot H)^{-1} \cdot H^T \cdot \xi\right] = \\ &= M\left[\beta_m^0\right] + M\left[(H^T \cdot H)^{-1} \cdot H^T \cdot \xi\right] = M\left[\beta_m^0\right].\end{aligned}$$

Даний вираз являє собою константу, тому його можна винести окремо від ξ , за математичне сподівання [29]:

$$(H^T \cdot H)^{-1} \cdot H^T.$$

В свою чергу $M[\xi] = 0$, тому:

$$M\left[(H^T \cdot H)^{-1} \cdot H^T \cdot \xi\right] = (H^T \cdot H)^{-1} \cdot H^T \cdot M[\xi].$$

В результаті:

$$M\left[\hat{\beta}_m\right] = M\left[\beta_m^0\right].$$

Це позначає незміщеність оцінки [29].

2.2.6 Ітераційний метод найменших квадратів

Перейдемо від методу МНК до РМНК. При отриманні наступного виміру робиться нове обчислення оцінок коефіцієнтів. Для наведеного вище прикладу новий $m+1$ вимір має вигляд [29] :

$$h_{m+1}^T = [1 \ x_{m+1}] .$$

$$\begin{aligned} \hat{\beta}_{m+1} &= \left(H^T \boxplus h_{m+1} \cdot \begin{pmatrix} H \\ \boxplus \\ h_{m+1}^T \end{pmatrix} \right)^{-1} \cdot H^T \boxplus h_{m+1} \cdot \begin{pmatrix} Y \\ \boxplus \\ y_{m+1} \end{pmatrix} = \\ &= \left(H^T \cdot H + h_{m+1} \cdot h_{m+1}^T \right)^{-1} \cdot \left(H^T \cdot Y + h_{m+1} \cdot y_{m+1} \right) . \end{aligned}$$

Запис $\boxplus$ означає приєднання вектору в матриці.

Окремо розглянемо вираз:

$$\left(H^T \cdot H + h_{m+1} \cdot h_{m+1}^T \right)^{-1} .$$

Для його перетворення використовується наступне співвідношення:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} .$$

Доказ цього співвідношення робиться шляхом домноження обох частин на матрицю $[\mathbf{A} + \mathbf{BCD}]$.

Якщо припустити дану рівність:

$$A = H^T \cdot H; \quad B = h_{m+1}; \quad C = I; \quad D = h_{m+1}^T.$$

То:

$$\begin{aligned} (H^T \cdot H + h_{m+1} \cdot h_{m+1}^T)^{-1} &= (H^T \cdot H)^{-1} - \\ &- (H^T \cdot H)^{-1} \cdot h_{m+1} \cdot \left[1 + h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1} \right]^{-1} \cdot h_{m+1}^T \cdot (H^T \cdot H)^{-1} = \\ &= (H^T \cdot H)^{-1} - \frac{(H^T \cdot H)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot (H^T \cdot H)^{-1}}{1 + h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1}}. \end{aligned}$$

Отриманий вираз використовується в формулі для обчислення $\hat{\beta}_{m+1}$:

$$\begin{aligned} \hat{\beta}_{m+1} &= (H^T \cdot H + h_{m+1} \cdot h_{m+1}^T)^{-1} \cdot (H^T \cdot Y + h_{m+1} \cdot y_{m+1}) = \\ &= \left((H^T \cdot H)^{-1} - \frac{(H^T \cdot H)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot (H^T \cdot H)^{-1}}{1 + h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1}} \right) \cdot (H^T \cdot Y + h_{m+1} \cdot y_{m+1}) = \\ &= (H^T \cdot H)^{-1} \cdot H^T \cdot Y - \frac{(H^T \cdot H)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot H^T \cdot Y}{1 + h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1}} + \\ &+ (H^T \cdot H)^{-1} \cdot h_{m+1} \cdot y_{m+1} - \frac{(H^T \cdot H)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1} \cdot y_{m+1}}{1 + h_{m+1}^T \cdot (H^T \cdot H)^{-1} \cdot h_{m+1}}. \end{aligned}$$

Зробимо наступну заміну:

$$\hat{\beta}_m = (H^T \cdot H)^{-1} \cdot H^T \cdot Y.$$

Також приведемо всі члени окрім першого до загального знаменника:

$$\begin{aligned}
 \hat{\beta}_{m+1} &= \hat{\beta}_m - \frac{\left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot \hat{\beta}_m}{1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}} + \\
 &+ \frac{\left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot y_{m+1} - \left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot y_{m+1}}{1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}} = \\
 &= \hat{\beta}_m - \left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot \left(\frac{h_{m+1}^T \cdot \hat{\beta}_m - y_{m+1}}{1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}} \right) = \\
 &= \hat{\beta}_m + \left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot \left(\frac{y_{m+1} - h_{m+1}^T \cdot \hat{\beta}_m}{1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}} \right)
 \end{aligned}$$

Отриману формулу можна переписати у вигляді:

$$\begin{aligned}
 \hat{\beta}_{m+1} &= \hat{\beta}_m + \left(H^T \cdot H\right)^{-1} \cdot h_{m+1} \cdot \left(\frac{y_{m+1} - h_{m+1}^T \cdot \hat{\beta}_m}{1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}} \right) = \\
 &= \hat{\beta}_m + \frac{\left(H^T \cdot H\right)^{-1} \cdot h_{m+1}}{\left(1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}\right)} \cdot \left(y_{m+1} - h_{m+1}^T \cdot \hat{\beta}_m \right) = \\
 &= \hat{\beta}_m + \frac{\left(H^T \cdot H\right)^{-1} \cdot h_{m+1}}{\left(1 + h_{m+1}^T \cdot \left(H^T \cdot H\right)^{-1} \cdot h_{m+1}\right)} \cdot \left(y_{m+1} - \hat{y}_{m+1} \right).
 \end{aligned}$$

де $\hat{y}_{m+1} = h_{m+1}^T \cdot \hat{\beta}_m$ – це прогноз [28].

2.2.7 Коваріація вектору оцінок коефіцієнтів

Обчислимо коваріацію вектора оцінок коефіцієнтів $\hat{\beta}$, так як:

$$\hat{\beta} = (H^T \cdot H)^{-1} \cdot H^T \cdot Y.$$

Тому:

$$M \begin{bmatrix} \hat{\beta} \cdot \hat{\beta}^T \end{bmatrix} = M \left[(H^T \cdot H)^{-1} \cdot H^T \cdot Y \cdot Y^T \cdot H \cdot (H^T \cdot H)^{-1} \right] =$$

Позначимо:

$$H^+ = (H^T \cdot H)^{-1} \cdot H^T,$$

$$(H^+)^T = H \cdot (H^T \cdot H)^{-1}.$$

Тоді:

$$= M \left[H^+ \cdot Y \cdot Y^T \cdot (H^+)^T \right] = H^+ \cdot M[Y \cdot Y^T] \cdot (H^+)^T =$$

Введемо заміну:

$$Y = H \cdot \beta + \xi$$

$$\begin{aligned}
&= H^+ \cdot M \left[\left(H \cdot \overset{0}{\beta} + \xi \right) \cdot \left(H \cdot \overset{0}{\beta} + \xi \right)^T \right] \cdot (H^+)^T = \\
&= H^+ \cdot M \left[H \cdot \overset{0}{\beta} \cdot \overset{0}{\beta}^T \cdot H^T + H \cdot \overset{0}{\beta} \cdot \xi^T + \xi \cdot \overset{0}{\beta}^T \cdot H^T + \xi \cdot \xi^T \right] \cdot (H^+)^T =
\end{aligned}$$

Використовуючи властивість лінійності математичного сподівання розкладемо отриманий вираз в суму математичних сподівань [28].

$$\begin{aligned}
&= H^+ \cdot \left(M \left[H \cdot \overset{0}{\beta} \cdot \overset{0}{\beta}^T \cdot H^T \right] + M \left[H \cdot \overset{0}{\beta} \cdot \xi^T \right] + M \left[\xi \cdot \overset{0}{\beta}^T \cdot H^T \right] + M \left[\xi \cdot \xi^T \right] \right) \cdot (H^+)^T = \\
&\quad M \left[H \cdot \overset{0}{\beta} \cdot \overset{0}{\beta}^T \cdot H^T \right] = H \cdot M \left[\overset{0}{\beta} \cdot \overset{0}{\beta}^T \right] \cdot H^T = 0
\end{aligned}$$

Так як коваріація не випадкової величини $\overset{0}{\beta}$ дорівнює нулю, тобто:

$$\begin{aligned}
&M \left[\overset{0}{\beta} \cdot \overset{0}{\beta}^T \right] = 0, \\
&M \left[H \cdot \overset{0}{\beta} \cdot \xi^T \right] = H \cdot \overset{0}{\beta} \cdot M \left[\xi^T \right] = 0, \\
&M \left[\xi \cdot \overset{0}{\beta}^T \cdot H^T \right] = M \left[\xi \right] \cdot \overset{0}{\beta}^T \cdot H^T = 0, \\
&M \left[\xi \cdot \xi^T \right] = \delta^2 \cdot I, \\
&= H^+ \cdot \delta^2 \cdot I \cdot (H^+)^T = \delta^2 \cdot H^+ \cdot (H^+)^T = \delta^2 \cdot (H^T \cdot H)^{-1} \cdot H^T \cdot H \cdot (H^T \cdot H)^{-1} = \\
&= \delta^2 \cdot I \cdot (H^T \cdot H)^{-1} = \delta^2 \cdot (H^T \cdot H)^{-1} \\
&\quad (H^T \cdot H)^{-1} \cdot H^T \cdot H = I.
\end{aligned}$$

В результаті виходить, що:

$$M \begin{bmatrix} \hat{\beta} \cdot \hat{\beta}^T \end{bmatrix} = \sigma^2 \cdot (H^T \cdot H)^{-1}.$$

2.3 Лінійні імовірнісні моделі

Розглянемо регресійну модель виду:

$$y_i = \mathbf{Z}_i \beta + v_i,$$

де $\mathbf{Z}_i$ – це набір характеристик;

β – вектор параметрів.

У такому контексті бінарність значень y_i викликає деякі проблеми, наприклад, при інтерпретації фізичного змісту змінної [29].

Таку систему називають лінійною імовірнісною моделлю (ЛІМ). Якщо залежна змінна неперервна, то зручно перейти до $E[y|Z]$ (математичне сподівання y за умови набору Z). До фіктивних змінних (приймають значення 1 або 0 в залежності від двох можливих отримуваних результатів) у загальному випадку такий підхід не застосовується. Головним недоліком лінійної імовірнісної моделі є те, що вона не забезпечує обмеження значень вихідної змінної в межах від 0 до 1. Оскільки похідна від імовірності по відповідному X дорівнює відповідному β , тоді дані обмеження відсутні. Розглянемо приклад з позитивним коефіцієнтом при пояснювальній змінній. У такому випадку, існує значення пояснювальної змінної, для якого

імовірність більше 1. Лінійні імовірнісні моделі є гетероскедастичними (дисперсія залежить від часу) [29].

Оскільки вихідні значення даного типу моделей не обмежені інтервалом $[0; 1]$, то його застосування неможливе в ряді практичних випадків. Перевагою таких моделей є простота (за умови неперервності) [29].

Оскільки ЛІМ мають внутрішні недоліки для деяких використань, то розглянемо альтернативи. Перейдемо від $X\beta$ до імовірності:

$$prob(y_i = 1) = F(X_i\beta)$$

За функцію F , яка відображає $X\beta$ у інтервал $[0; 1]$, виберемо функцію розподілу. Якщо за F вибрати тотожну функцію, то одержимо:

$$prob(y_i = 1) = X_i\beta$$

Це не підходить [29].

Якщо ж покласти F рівною стандартному нормальному розподілу, то одержимо пробіт-модель:

$$prob(y_i = 1) = \Phi(X_i\beta) = \int_{-\infty}^{X_i\beta} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz$$

Такий варіант підходить, тому що:

$$\lim_{z \longrightarrow +\infty} \Phi(z) = 1$$

$$\lim_{z \longrightarrow -\infty} \Phi(z) = 0$$

Вибираючи логістичний розподіл, одержимо ще одну можливу модель – логіт-модель:

$$\text{prob}(y_i = 1) = \Lambda(X_i \beta) = \frac{\exp(X_i \beta)}{1 + \exp(X_i \beta)}.$$

Її значення також лежать в інтервалі $[0; 1]$.

Ці два види моделей найчастіше використовують на практиці, хоча застосовувати можна будь-яку іншу функцію, що задовольняє умові обмеження виходу інтервалом $[0; 1]$ [29].

2.3.1 Пробіт-модель

Розглянемо пробіт-модель. Нехай ми спостерігаємо змінну y , що приймає значення 0 або 1. Визначимо y^* у такий спосіб:

$$y_i^* = X_i \beta + \varepsilon_i.$$

Передбачається, що:

$$\varepsilon_i \sim N(0, \sigma^2).$$

На відміну від ЛІМ, y_i^* розподілена нормально, хоча y_i – ні.

Нагадаємо, що на відміну від ЛІМ, y_i^* у пробіт-моделі розподілена нормально, а спостереження y_i – ні. Покажемо далі, що умова формує пробіт-модель. Спочатку відзначимо:

$$\begin{aligned} \text{prob}(y_i = 1) &= \text{prob}(y_i^* > 0) = \text{prob}(X_i \beta + \varepsilon_i > 0) = \\ &= \text{prob}(\varepsilon_i > -X_i \beta) = \text{prob}\left(\frac{\varepsilon_i}{\sigma} > -X_i \frac{\beta}{\sigma}\right), \end{aligned}$$

де σ^2 – дисперсія ε .

Ділення на σ дає можливість перейти до величини $\frac{\beta}{\sigma}$, що має стандартний нормальний розподіл, тобто нульове математичне сподівання і одиничну дисперсію. Величина $\frac{\varepsilon}{\sigma}$ є стандартною нормально розподіленою, тому що вона отримана з ε вирахуванням її математичного сподівання (нуля) і діленням на її дисперсію σ . Для пробіт-моделі розподіл є симетричним, тому вираз можна переписати в такий спосіб:

$$\text{prob}(y_i = 1) = \text{prob}\left(\frac{\varepsilon_i}{\sigma} > -X_i \frac{\beta}{\sigma}\right) = \text{prob}\left(\frac{\varepsilon_i}{\sigma} < X_i \frac{\beta}{\sigma}\right) = \Phi\left(X_i \frac{\beta}{\sigma}\right)$$

Далі виведемо функцію правдоподібності. Оскільки:

$$\text{prob}(y_i = 1) = \Phi\left(X_i \frac{\beta}{\sigma}\right)$$

То:

$$\text{prob}(y_i = 0) = 1 - \Phi\left(X_i \frac{\beta}{\sigma}\right)$$

Якщо спостереження незалежні в сукупності, то імовірність усієї вибірки дорівнює добутку ймовірностей кожного спостереження. Нехай

спостереження $1, \dots, m$ мають результат $y_i = 0$, а для $m + 1, \dots, n$ спостережень $y_i = 1$. Тоді:

$$L = \prod_{i=1}^m \text{prob}(y_i = 0) * \prod_{i=m+1}^n \text{prob}(y_i = 1),$$

$$L = \prod_{i=1}^m \left[1 - \Phi\left(X_i \frac{\beta}{\sigma}\right) \right] * \prod_{i=m+1}^n \Phi\left(X_i \frac{\beta}{\sigma}\right),$$

$$L = \prod_{i=1}^n \Phi\left(X_i \frac{\beta}{\sigma}\right)^{y_i} * \left[1 - \Phi\left(X_i \frac{\beta}{\sigma}\right) \right]^{1-y_i}.$$

Зазвичай працюють з логарифмічною функцією правдоподібності:

$$l\left(\frac{\beta}{\sigma}\right) = \ln(L) = \sum_i \left\{ y_i * \ln\left[\Phi\left(X_i \frac{\beta}{\sigma}\right)\right] + (1 - y_i) * \ln\left[1 - \Phi\left(X_i \frac{\beta}{\sigma}\right)\right] \right\}.$$

Логарифмічна функція правдоподібності приймає значення з інтервалу $(-\infty; 0]$, тому що $0 \leq \Phi(\bullet) \leq 1$.

Інший важливий аспект логарифмічної функції правдоподібності полягає в тому, що параметри β і σ завжди з'являються разом. Тому тільки

величина $\frac{\beta}{\sigma}$ має значення. Для зручності σ нормалізують, щоб працювати тільки з β .

Обчислення пробіта виконується безпосередньо, навіть якщо модель є нелінійною чи немає чіткого виразу для Φ . Значення функції Φ обчислюється чисельними методами. Однією з характеристик пробіта (і логіта) є те, що функція правдоподібності є увігнутою. Тому, використовуючи оптимізаційні пакети, не варто побоюватися того, що буде знайдений якийсь локальний максимум [29].

Процедура обчислення полягає у визначенні оцінок для ЛІМ і їх подальшому використанні як початкової точки пошуку. Пошук ведеться доти, поки значення функції правдоподібності зростають. Один з методів пошуку називається методом кількісної оцінки. Оцінки пробіта знаходяться ітераційно:

$$\beta_{f+1} = \beta_f + I^{-1}(\beta_f) * \frac{\partial l}{\partial \beta_f},$$

де індекс вказує номер ітерації. $I(\beta_f)$ – це оцінка інформаційної матриці (квадратної симетричної матриці від’ємних других похідних), обчисленої на останньому кроці.

Коли різниця між β_{f+1} і β_f близька до нуля, пошук припиняється. Відмінності в значеннях коефіцієнтів при двох кращих ітераціях будуть близькі до нуля, якщо значення $\frac{\partial l}{\partial \beta}$ (похідна логарифмічної функції правдоподібності по відповідним параметрах) близька до нуля [29].

2.3.2 Логіт-модель

Побудова логіта ідентична пробіту. З виразу маємо:

$$prob(y_i = 1) = \Lambda(X_i \beta) = \frac{\exp(X_i \beta)}{1 + \exp(X_i \beta)}.$$

Інтерпретація латентної змінної логіта здійснюється аналогічно пробіту з тією відмінністю, що в вираз додане ε , яке називається розподілом екстремальних значень. Як і в пробіті, побудова моделі забезпечує обмеження

вихідних значень інтервалом $[0; 1]$. Головною відмінністю нормального і логістичного розподілу є те, що останній має більшу вагу на «хвостах».

Похідна імовірності по елементу з X змінюється з X , як і в пробіті [29]:

$$\frac{\partial E(y)}{\partial X_k} = \frac{\exp(X\beta)}{(1 + \exp(X\beta))^2} \beta_k$$

Перепишемо похідну так:

$$\frac{\partial E(y)}{\partial X_k} = p(1 - p)\beta_k$$

$$p = \frac{\exp(X\beta)}{1 + \exp(X\beta)}$$

2.4 Історія виникнення та практика застосування логістичної регресії

Логістична регресія аналізує зв'язок між однією або кількома незалежними змінними та залежною змінною [30], яка має категоріальний характер. На відміну від лінійної регресії, яка передбачає безперервний вихід, логістична регресія використовується, коли вихідна змінна є дискретною. Модель оцінює ймовірність належності спостереження до певного класу, використовуючи логістичну функцію (сигмоїду), яка обмежує вихідні значення в інтервалі від 0 до 1 [30].

1. П'єр-Франсуа Вергюльст (1838-1847). Бельгійський математик П'єр-Франсуа Вергюльст вперше запропонував логістичну функцію в серії робіт між 1838 і 1847 роками для моделювання

зростання населення. Він розробив рівняння, яке враховувало обмеження ресурсів, що призводить до насичення популяції. Ця функція стала відома як логістична крива [31].

2. Реймонд Перл і Ловелл Рід (1920). У 1920 році Реймонд Перл і Ловелл Рід незалежно відновили логістичну функцію для моделювання зростання населення США. Вони не були знайомі з роботами Вергюльста, але пізніше дізналися про них через Л. Густава дю Паск'є. Їхні дослідження сприяли подальшому застосуванню логістичної функції в статистиці [31].
3. Честер Іттнер Блісс і Джон Гаддум (1930-ті). У 1930-х роках Честер Іттнер Блісс розробив пробіт-модель для біоаналізу, використовуючи кумулятивну нормальну функцію розподілу. Ця модель дозволяла оцінювати ймовірності настання подій, таких як смертність при різних дозах речовин. Джон Гаддум також зробив внесок у розвиток цієї моделі [31].
4. Джозеф Берксон (1944). Джозеф Берксон ввів термін "логіт" у 1944 році, використовуючи логарифм відношення шансів (log-odds) для моделювання ймовірностей. Він запропонував логістичну модель як альтернативу пробіт-моделі, підкреслюючи її математичну простоту та зручність у застосуванні [31].
5. Сер Девід Кокс (1958). Британський статистик Девід Кокс у 1958 році опублікував роботу, в якій систематизував логістичну регресію як частину узагальнених лінійних моделей (GLM). Його внесок сприяв широкому впровадженню логістичної регресії в статистичний аналіз [31].

2.5 Критерії якості

2.5.1 Описові характеристики математичної моделі

До описових статистик часового ряду відносять наступні показники.

Середнє (mean) – значення математичного сподівання.

Мода (mode) – це значення елемента ряду, яке зустрічається найчастіше.

Медіана (median) – це значення елемента ряду, який є більшим або дорівнює, і одночасно меншим або дорівнює половині інших елементів ряду; наприклад, для ряду значень 4, 4, 5, 7, 10 медіаною є 5, для ряду 5, 4, 11, 4, 7 медіаною також є 5 [28].

Стандартне відхилення (standard deviation) – корінь квадратний з дисперсії:

$$\sigma = \sqrt{D(y)} = \sqrt{\frac{1}{N-1} \sum_{k=1}^N [y(k) - \bar{y}]^2},$$

де N – довжина (потужність) ряду;

$\bar{y}$ – середнє значення ряду,

$D(y)$ – дисперсія.

Коефіцієнт асиметрії (skewness) – характеризує симетричність (хвостів) розподілу і розраховується за виразом [28] :

$$S = \frac{1}{N} \sum_{k=1}^N \left[\frac{y(k) - \bar{y}}{\sigma} \right]^3.$$

Якщо $S > 0$, то правий хвіст розподілу довший, при $S < 0$ довшим є лівий хвіст розподілу.

Якщо $S = 0$, то розподіл симетричний (в реальному житті симетричних хвостів не буває).

Ексцес (kurtosis) – характеризує відмінність форми розподілу від нормального (Гауса) і розраховується за виразом:

$$K = \frac{1}{N} \sum_{k=1}^N \left[\frac{y(k) - \bar{y}}{\sigma} \right]^4.$$

Для нормального розподілу $K = 3$.

Якщо $K > 3$, то форма розподілу буде “гострішою” від нормального.

При $K < 3$ форма розподілу буде “плоскішою” від нормального.

Статистика Жак-Бера (Jarque-Bera) – тестова статистика, яка показує наскільки близьким є ряд до нормального розподілу; фактично це різниця між значеннями S і K для досліджуваного ряду та нормального розподілу [28]:

$$JB = \frac{N-n}{6} \left[S^2 + \frac{1}{4} (K-3)^2 \right],$$

де n – число коефіцієнтів, використаних для побудови моделі ряду.

При нуль-гіпотезі щодо нормальності розподілу статистика Жак-Бера має розподіл χ^2 (хі квадрат) з двома степенями свободи. Ймовірність, зв’язана із статистикою Жак-Бера, показує ймовірність справедливості нуль-гіпотези. Мала ймовірність свідчить про те, що нуль-гіпотезу щодо нормальності розподілу необхідно відхилити [28].

2.5.2 Прогнозуючі характеристики регресійної моделі

Sum square resid.

Сума квадратів похибок моделі (SSE – sum of square error) в загальному випадку обчислюється за формулою:

$$SSE = \sum_{t=1}^N (y - \hat{y})^2 ,$$

де N – розмір вибірки;

y – фактичні значення;

$\hat{y}$ – значення побудовані за моделлю.

Необхідно відмітити, що окрім SSE, ще існує середньоквадратична похибка (MSE – mean square error) яка обчислюється за формулою:

$$MSE = E((y - \hat{y})^2) = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N} .$$

Корінь із середньоквадратичної похибки (RMSE – root mean square error) [28]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} .$$

Durbin-Watson stat.

Коефіцієнт Дурбина-Уотсона (DW – Durbin-Watson) показує адекватність побудованої моделі та обчислюється за формулою [28]:

$$DW = \frac{\sum_{t=2}^N (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^N \varepsilon_t^2},$$

де ε – вектор залишків (різниця між значеннями отриманими за моделлю та фактичними).

При цьому $DW \in [0; 4]$. Для найкращої моделі $DW \rightarrow 2$, це означає що залишки моделі між собою не автокорелюють.

2.6 Оцінка якості моделі в задачах класифікації

Оцінка якості моделі в задачах класифікації полягає у вимірюванні того, наскільки точно й узгоджено алгоритм присвоює об'єктам відповідні категорії.

Крім простого показника точності (ассигасу), що відображає співвідношення всіх правильних передбачень до загальної кількості зразків, важливо аналізувати якість моделі за допомогою різних метрик які дають змогу точніше оцінити модель.

Існують такі метрики для оцінки якості моделі.

Матриця невизначеності (confusion matrix) є зручним інструментом для візуалізації результатів класифікації, оскільки містить інформацію про кількість правильних і неправильних передбачень для кожного з класів. Зазвичай вона має вигляд квадратної таблиці розміром 2×2 у випадку бінарної класифікації, де по рядках зазначено дійсний (істинний) клас об'єктів, а по стовпцях – передбачений клас. Така структура дозволяє швидко оцінити,

скільки прикладів кожного класу модель правильно або помилково класифікувала [32].

У межах матриці невизначеності виділяють чотири основні показники:

True Positive (TP) – кількість об'єктів, які насправді належать до позитивного класу й були правильно класифіковані як позитивні.

True Negative (TN) – кількість об'єктів, які дійсно є негативними й модель правильно віднесла до негативного класу.

False Positive (FP) – кількість об'єктів негативного класу, які модель помилково класифікувала як позитивні (так звані «помилкові тривоги»).

False Negative (FN) – кількість об'єктів позитивного класу, які модель неправильно віднесла до негативного (так звані «пропуски»).

Ці чотири компоненти дають змогу детально зрозуміти, в яких випадках класифікатор помиляється і як часто він правильно ідентифікує кожен із класів.

MISC (Misclassification Rate) показує загальну частку неправильно класифікованих зразків і обчислюється як відношення суми помилкових спрацьовувань до загальної кількості спостережень [33]:

$$MISC = \frac{FP + FN}{TP + TN + FP + FN}.$$

Цей показник відображає, яку частину всіх випадків модель класифікувала некоректно, незалежно від того, чи це були помилкові тривоги (FP), чи пропуски (FN).

Recall (Повнота) визначає здатність моделі виявляти всі справжні позитивні випадки і обчислюється за формулою [34]:

$$Recall = \frac{TP}{TP + FN}.$$

Цей показник показує, яку долю об'єктів позитивного класу модель змогла правильно класифікувати.

Precision (Точність) показує, яку частку об'єктів, передбачених моделлю як позитивні, насправді належать до позитивного класу. Обчислюється вона за формулою [34]:

$$Precision = \frac{TP}{TP + FP}.$$

Цей показник демонструє, яку частку об'єктів, передбачених як позитивні, модель класифікувала правильно.

F-Score є метрикою, що поєднує в собі як точність (Precision), так і повноту (Recall) класифікатора, і розраховується як їхнє гармонійне середнє [35]:

$$F - Score = 2 \frac{Precision \times Recall}{Precision + Recall}.$$

Цей показник особливо корисний у випадках, коли важливо враховувати компроміс між тим, наскільки модель уникає хибних тривог (FP) і наскільки вона не пропускає справжні позитивні випадки (FN). Високе значення F-Score вказує на те, що обидва параметри точність і повнота збалансовані та досягли хорошого рівня.

Крива ROC (Receiver Operating Characteristic) – це графік, на якому по вертикальній осі відкладають рівень істинно позитивних спрацьовувань (True Positive Rate, TPR), а по горизонтальній осі – рівень хибно позитивних спрацьовувань (False Positive Rate, FPR) при різних значеннях порогу класифікації. Її також можна розглядати як залежність статистичної потужності від помилки першого роду для даного правила прийняття рішення. Якщо показники TPR та FPR оцінюються на основі вибірки, крива відображає саме ці оцінки. Коли відомі розподіли ймовірностей для обох

класів, точка на кривій відповідає значенням кумулятивних функцій розподілу (CDF) позитивного та негативного класів до певного порогу [36].

ROC-аналіз дає змогу порівнювати та відбирати моделі класифікації незалежно від конкретних витрат на помилки чи розподілу класів, оскільки показує співвідношення чутливості (справжніх позитивів) та специфічності (справжніх негативів) для всіх можливих порогів. Площа під ROC-кривою (AUC) служить узагальненим критерієм якості моделі: значення 1 відповідає ідеальній класифікації, а 0,5 – випадковому відгадуванню. Застосунок ROC-аналізу охоплюють медичну діагностику, кредитний скоринг, виявлення шахрайства й інші задачі бінарної класифікації [36].

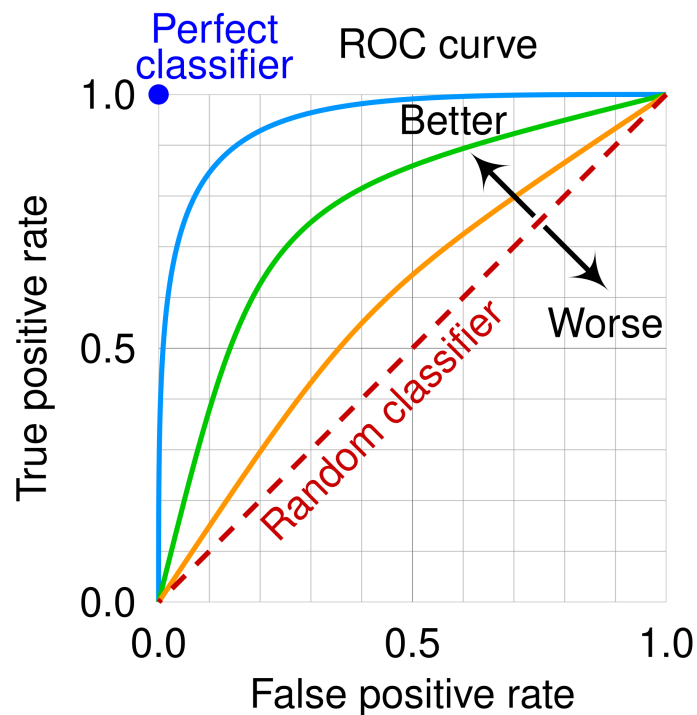


Рисунок 2.2 – Опис ROC-кривої [36]

Коефіцієнт Джині – це показник нерівності розподілу (наприклад, доходів або багатства) в суспільстві. Він вимірюється за площею між лінією ідеальної рівності та фактичною кривою Лоренца, і може змінюватися від 0 (повна рівність) до 1 (максимальна нерівність). Чим вищий коефіцієнт, тим сильніше сконцентровані ресурси в руках невеликої групи людей.

Висновки до розділу 2

У цьому розділі проаналізовано основні підходи до побудови математичних моделей для прогнозування. Розкрито поняття структури моделі. Обґрунтовано застосування методу найменших квадратів, наведено його геометричний зміст, доведено умови незміщеності оцінок і проаналізовано ітераційний підхід для складніших випадків. Докладно розглянуто обчислення коваріації вектора коефіцієнтів, що дає змогу оцінити статистичну значущість параметрів і побудувати довірчі інтервали для прогнозів.

Висвітлено лінійні ймовірнісні моделі, зокрема пробіт- та логіт-підходи, що застосовуються для задач класифікації. Показано історичне становлення логістичної регресії та її практичне використання у фінансових дослідженнях. Особливу увагу приділено характеристикам якості моделей. Для класифікаційних задач роз'яснено, як формуються показники TP, TN, FP, FN та на основі них обчислюються Precision, Recall, F-Score і MISC, що допомагає враховувати баланс між хибними тривогами й пропусками. Також були описані такі показники, як ROC-крива та показник Джинні. Цей теоретичний базис забезпечує фундамент для подальшої практичної реалізації моделей у дослідженні.

РОЗДІЛ 3 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ

3.1 Вступ

В цьому розділі наведений опис розробленої програми. Ця програма побудована для прогнозування факту лістингу нових криптовалют на біржі Binance за допомогою логістичної регресії. Для написання програми застосовувалася мова програмування Python.

На рис. 3.1 наведена високорівнева блок-схема розробленої системи. Система складається з 6 модулів.

Модуль завантаження даних відповідає за завантаження даних з таких джерел, як API криптобірж та веб сайт CoinMarketCap.

Модуль інтеграції з chat-gpt API виконує сентимент аналіз для текстових даних, які формують зміни датасету.

Модуль обробки даних перевіряє дані на пробіли та аномалії та виправляє їх.

Модуль генерації ознак датасету формує датасет за допомогою наданих даних.

Модуль прогнозної моделі надає інтерфейс для навчання, тестування та використання моделі логістичної регресії.

Модуль аналізу результатів та підрахунку метрик отримує на вхід результати прогнозу моделі, будує метрики та ілюструє їх.

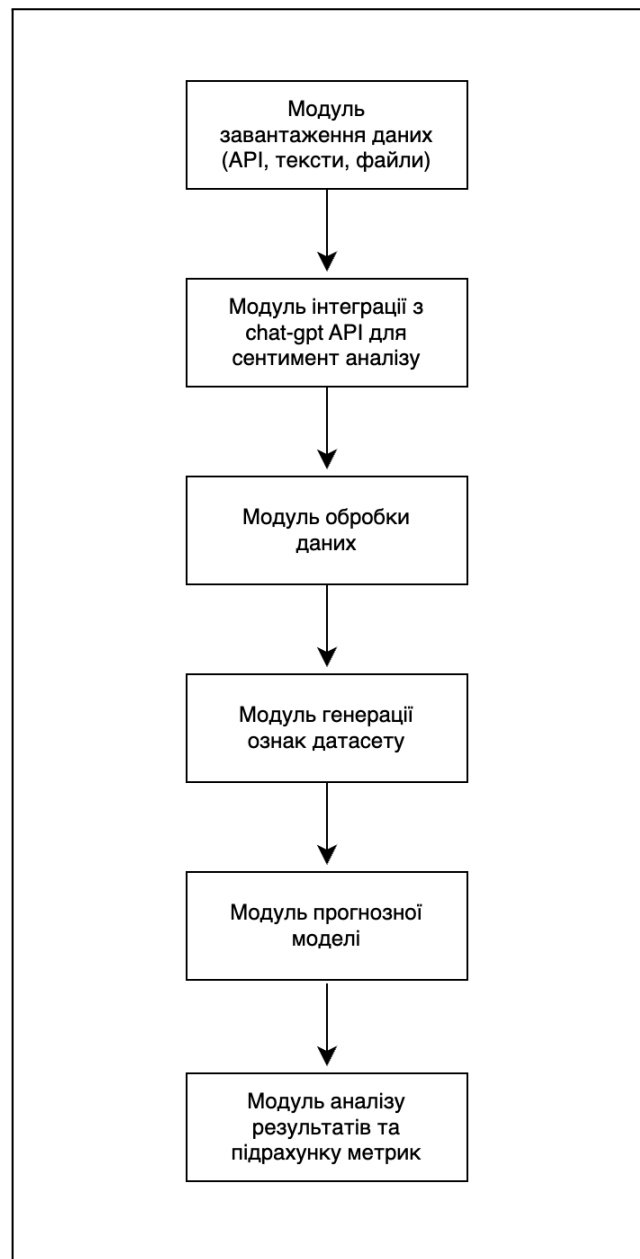


Рисунок 3.1 – Високорівнева система розробленої блок-схеми

3.2 Опис джерел для аналізу

Данні для аналізу, в рамках даної дипломної роботи, було зібрано шляхом вивантаження цін криптоактивів та інформації про них з наступних восьми найкращих криптобірж, за рейтингом від CoinMarketCap [8].

BINANCE – найбільший он-лайн сервіс з обміну криптовалюти, заснований у 2017 році (<https://www.binance.com>). Перше місце за рейтингом від CoinMarketCap [8].

COINBASE – американська публічна компанія, яка керує платформою обміну криптовалюти, заснована 2012 році. Найбільша біржа криптовалют у Сполучених Штатах за обсягом торгів. Друге місце за рейтингом від CoinMarketCap [8].

BYBIT – платформа для деривативів на криптовалюту, заснована у 2018 році. Четверте місце за рейтингом від CoinMarketCap [8].

KUCOIN – крипто-біржа, заснована у 2014 році. П'яте місце за рейтингом від CoinMarketCap [8].

OKX – друга за обігом крипто-біржа, заснована у 2017 році. Шосте місце за рейтингом від CoinMarketCap [8].

HTX – сервіс обміну криптовалют, заснований в 2013 році під назвою HUOBI. 13 вересня 2023 року на честь десятиріччя компанія зробила ребрендинг та змінила назву з Huobi на HTX (H – Huobi, T – TRON, X – eXchange). Займає одинадцяте місце за рейтингом від CoinMarketCap [8].

BITGET – криптовалютна біржа, заснована в 2018 році. Bitget є партнером футбольного клубу Ювентус, гравця Ліонеля Мессі, організатором глобальних кіберспортивних турнірів PGL та всесвітнього турніру з комп'ютерної гри DOTA 2 Bali Major. Шістнадцяте місце за рейтингом від CoinMarketCap [9].

MEXC – централізована криптовалютна біржа, заснована в 2018 році. Вісімнадцяте місце за рейтингом від CoinMarketCap [8].

3.3 Опис змінних моделі

Зауваження, щодо ролей змінних.

Технічні змінні не приймають участі в побудові аналітичної моделі.

Цільова змінна – це змінна, для якої будується прогноз. В нашому випадку – CLASS (1 – факт лістингу криптоактиву, на баржі Бінанс, або 0 – відсутність факту лістингу).

Регресор – вхідна змінна, що впливає на цільову змінну.

Таблиця 3.1 – Опис змінних моделі, що використовуються для аналізу

Номер	Ім'я змінної	Роль в аналізі	Опис змінної
1	listing_date	час (технічна)	Дата лістингу криптоактиву.
2	symbol	технічна	Тікер криптоактиву (позначення на біржі Binance).
3	week_price_change	регресор	Зміна ціни криптоактиву за останній тиждень у відсотках. Зауважимо, що значення ціни у відповідні моменти часу обчислювалися як середні значення цін по всіх біржах, що досліджуються в аналізі.
4	month_price_change	регресор	Зміна ціни криптоактиву за останній місяць у відсотках.
5	week_volume_change	регресор	Зміна обсягу обігу криптоактиву у відсотках, за останній тиждень.
6	month_volume_change	регресор	Зміна обсягу обігу криптоактиву у відсотках, за останній місяць.
7	rank	регресор	Рейтинг за капіталізацією, за 70 діб до моменту лістингу. У випадку, коли немає даних, використовувалася інтерполяція, щоб заповнити відсутні значення, як наслідок змінна може містити нецілі числа.
8	N_of_listings	регресор	Число лістингів даного криптоактиву на інших біржах. Інформація береться за 2 дні до лістингу на біржі Бінанс. Дана змінна формувалася як сумарна кількість бірж, на яких вже був проведений лістинг криптоактиву. Серед усіх бірж, для формування цієї змінної, було обрано наступні сім: HUOBI, OKX, BYBIT, BITGET, MEXC, KUCOIN, COINBASE
9	month_delta_listings	регресор	Приріст за кількістю бірж, доданих за останній місяць. Ця змінна пов'язана, з попередньою змінною N_of_listings, та показує приріст за кількістю бірж, доданих за останній місяць.
10	mean_volume	регресор	Середньодобовий обсяг обігу криптоактиву, за останній місяць.

Номер	Ім'я змінної	Роль в аналізі	Опис змінної
11	class	Цільова змінна	Цільова змінна, що приймає наступні значення: 1 – відбувся лістинг криптоактиву на біржі Binance. 0 – лістинг на біржі Binance не відбувся.

3.4 Принцип формування змінної “тижнева зміна ціни”

Рисунок нижче демонструє принцип формування змінної “тижнева зміна ціни” (week_price_change), аналогічним чином формується змінна “тижнева зміна ціни” (month_price_change).

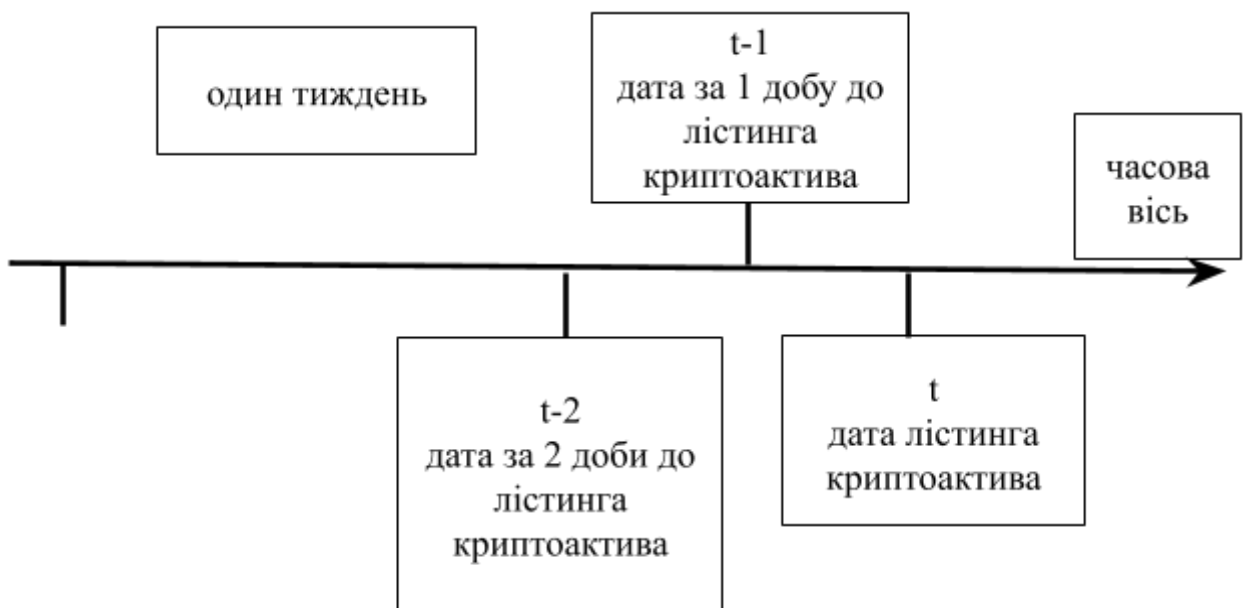


Рисунок 3.2 – Графічна демонстрація змінної “тижнева зміна ціни” (week_price_change), яка обчислюється на основі даних за тиждень, причому ці данні збираються саме за два дні до лістингу.

Загалом вхідний набір даних містить 8 регресорів та 1 цільову змінну. Список відповідних змінних:

- 1) week_price_change – зміна ціни криптоактиву протягом одного тижня у відсотках;
- 2) month_price_change – зміна ціни криптоактиву протягом одного місяця у відсотках;
- 3) week_volume_change – зміна обсягу обігу криптоактива у відсотках, за останній тиждень;
- 4) month_volume_change – зміна обсягу обігу криптоактива у відсотках, за останній місяць;
- 5) rank – рейтинг криптоактива за капіталізацією, до моменту лістингу на біржі Binance;
- 6) N_of_listings – загальна кількість лістингів на інших біржах, за 2 дні до лістингу на біржі Binance;
- 7) month_delta_listings – приріст за кількістю бірж, на яких відбувся факт лістингу криптоактиву, доданих за останній місяць;
- 8) mean_volume – середньодобовий обсяг обігу криптоактиву, за останній місяць;
- 9) class – Цільова змінна, що приймає наступні значення: 1 – відбувся лістинг криптоактиву на біржі Binance; 0 – лістинг на біржі Binance не відбувся;

Всього було накопичено 84 кейсів, що відбулися у період з 2021-11-16 по 2023-08-17. З 84 кейсів у 32 випадках відбувся факт лістингу відповідного криптоактиву на біржі Vinase, а у 52 – не відбувся.

Таблиця 3.2 – Статистика появи відповідних кейсів, по роках

Рік	Class = 0 (кількість кейсів коли лістинг не відбувся)	Class = 1 (кількість кейсів коли лістинг відбувся)	Загальна кількість кейсів
2021	3	5	8
2022	30	16	46
2023	19	11	30

3.5 Послідовність проведення аналізу

Опис процесу створення даних для аналізу.

Було розроблено код програми для створення набору даних, за допомогою коду програми, який можна знайти у додатку до цієї дипломної роботи.

Аналіз якості даних. Зазвичай аналіз якості даних включає в себе процедури перевірки на наявність:

- пропусків в даних;
- аномальних значень.

Нижче, наведені окремим кольором проблеми, наявні у відповідному наборі даних.

listing_date	symbol	week_price_change	month_price_change	week_volume_change	month_volume_change	rank	N_of_listings	month_delta_listings	mean_volume	class
2022-11-07	AURY	-1.66061584	-12.95179528	-46.72748456	-85.06653355	698	2	1	13273.02481	0
2022-11-18	MOON	-20.53010314	-27.47154116	26.00309315	26.51243709	.	1	0	135421.6967	0
2022-11-19	PHB	-23.0890217	-37.48313138	-3.040098966	13.04095893	632	3	0	1045272.107	1
2022-12-01	XYM	-3.048710602	-16.62998127	-8.002395595	-1.832744073	128	4	0	272029.8724	0
2022-12-13	MAGIC	37.21278268	96.68539813	-50.24420586	222.759343	235	4	1	749670.5648	1
2022-12-23	ZEE	-15.03658962	-10.22399509	-44.39395981	-63.23230994	.	2	0	102422.9656	0
2022-12-28	RBN	-6.338496594	14.75984686	-12.88927161	-62.41114473	151	3	0	229851.9581	0
2023-01-14	TIME	21.07554511	-5.703699609	142.7410718	-40.02693011	354	2	0	404998.2475	0
2023-01-19	RPL	30.22026893	56.37723046	200.3267626	384.916939	214	2	0	84472.01998	1
2023-01-21	KOK	-1.643720568	-15.77365663	-11.10562949	-28.46290979	685	4	0	732979.8459	0
2023-02-08	ORAI	45.9761263	172.3745659	534.8032502	243.3738835	703	1	0	687256.081	0
2023-02-14	H2O	-2.941664418	0.258607073	52.47605952	104.7696271	553	2	0	176545.3514	0
2023-02-16	PNK	-6.113309274	5.008996742	60.56642051	300.6915355	657	2	0	249880.773	0
2023-02-18	GNS	-4.25909624	105.2525935	-70.19930669	3397.034231	.	2	1	292135.4016	1
2023-02-20	KLV	1.091055798	62.43693452	-7.572466159	-19.06769208	471.8571429	3	0	660043.5595	0
2023-02-23	SYN	13.64077078	86.88609413	-58.40559221	260.8946746	165	3	0	2452621.465	1

Рисунок 3.3 – Приклад завантажених даних для аналізу. Зверніть увагу, що є пропуски в даних (показано красним), та дані можуть бути не цілими (показано зеленим)

Таблиця 3.3 – Результат виконання програми, щодо перевірки якості даних

Variable	Mean	Std Dev	Minimum	Maximum	N	N Miss
listing_date	22899,93	186,0477	22600	23239	84	0
week_price_change	-1,3771	16	-34	61	84	0
month_price_change	1,1704	39	-52	172	84	0
week_volume_change	33	120	-88	602	84	0
month_volume_change	135	492	-85	3397	84	0
rank	365	224,041	18	975	69	15
N_of_listings	2,6429	1,3047	1	6	84	0
month_delta_listings	0,2143	0,4410	0	2	84	0
mean_volume	2721938	6270547	13273	30643389	84	0
class	0,3810	0,4885	0	1	84	0

Опис стовпчиків, з таблиці наведеної вище, отриманих при виконанні програми:

- 1) Variable – ім'я змінної;
- 2) Mean – математичне очікування;
- 3) Std Dev - стандартне відхилення;
- 4) Minimum – мінімальне значення;
- 5) Maximum – максимальне значення;
- 6) N – кількість записів;
- 7) N Miss – кількість пропусків;

Необхідно звернути увагу, на те що, змінна rank має 15 пропусків. Це пов'язано із тим, що дана змінна показує рейтинг за капіталізацією, до моменту лістингу, і для деяких крипто-монет цей рейтинг дуже низький, нижче за ТОП-1500 список з сайту www.coinbase.com. Тому, в таких випадках, в сформованому наборі даних присутні значення – пропуск.

В подальшому при виконанні аналізу такі значення необхідно замінити на значення 1500. За своєю суттю – це буде означати, що відповідний криптоактив знаходиться на позиції нижче або рівній 1500, за капіталізацією.

Заміна пропусків в даних та приведення до цілих чисел. Необхідно звернути увагу, що за своїм змістом, майже всі змінні повинні бути цілими значеннями, а не дійсними числами (що мають значення після десятинної коми).

Було написано власний макрос для підбору балу відсікання CutOff (значення порогу), для вирішення задачі прогнозуванні. Було реалізовано комп'ютерну програму, що використовувалася для підбору балу відсікання CutOff, для відповідного TRAIN набору даних. Для роботи цього макросу на вхід необхідно подати наступні параметри:

- 1) TRAIN набір даних з прогнозами-оцінками;
- 2) List_CutOff – список CutOff параметрів-гіпотез, які будуть перевірятися.

В якості вихідного результату, макрос видає:

- 1) макро-змінна з ім'ям Best_CutOFF – що містить найкраще значення балу прийняття рішення;
- 2) **CutOff_Best_row** – файл зі статистиками, для підбраного балу відсікання.

3.6 Опис алгоритму в загальному вигляді по кроках

Таблиця 3.4 – Значення оцінок коефіцієнтів моделі, отриманих на перших 30 записах навчального набору даних (TRAIN)

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0,8199	1,4581	0,3162	0,5739
week_price_change	1	-0,0232	0,0466	0,2474	0,6189
month_price_change	1	-0,0428	0,0217	3,8773	0,0489
week_volume_change	1	0,00741	0,00604	1,5071	0,2196
month_volume_change	1	0,000396	0,00187	0,0447	0,8325
rank	1	-0,00081	0,00118	0,469	0,4934
N_of_listings	1	0,0106	0,5195	0,0004	0,9837
mean_volume	1	9,35E-08	6,42E-08	2,1191	0,1455
month_delta_listings	1	0,5976	1,3864	0,1858	0,6664

Таблиця 3.5 – Статистичні характеристики – оцінка співвідношення шансів (Odds Ratio_, оцінок параметрів моделі)

Оцінка співвідношення шансів (Odds Ratio Estimates)			
Параметр моделі	Point Estimate	95% Wald Confidence Limits	
week_price_change	0,977	0,89	1,071
month_price_change	0,958	0,92	1
week_volume_change	1,007	1	1,019
month_volume_change	1	1	1,004
rank	0,999	1	1,002
N_of_listings	1,011	0,37	2,798
mean_volume	1	1	1
month_delta_listings	1,818	0,12	27,52

Таблиця 3.6 – Статистичні характеристики – коефіцієнти конкордації, для прогнозів отриманих на основі побудованої моделі

Асоціація прогнозованих ймовірностей і спостережуваних реакцій (Association of Predicted Probabilities and Observed Responses)			
Percent Concordant	83,9	Somers' D	0,679
Percent Discordant	16,1	Gamma	0,679
Percent Tied	0	Tau-a	0,349
Pairs	224	c	0,839

Значення метрик конкордації можна використовувати для порівняння однієї моделі з іншою. Загалом, більш високий відсоток пар конкордантів і нижчі відсотки розсогласування пар вказують на більш бажану модель.

Чотири індекси рангової кореляції (Somer's D, Gamma, Tau-a, та c) обчислюються на основі кількості конкордантів, розсогласувань та пов'язаних пар спостережень. Взагалі, модель з більш високими значеннями для цих індексів має кращу прогностичну здатність, аніж модель з більш низькими значеннями для цих індексів.

При порівнянні декількох моделей наведені статистики для кращих моделей повинні:

- 1) значення статистики Percent Concordant повинно збільшуватися, прямувати до 100;
- 2) значення статистики Percent Discordant повинно зменшуватися, прямувати до 0.

C-коефіцієнт, ще називають коефіцієнт конкордації повинен збільшуватися. Діапазон можливих значень становить від 0,5 (для моделі з гарною прогностичною силою, аніж модель підкидання чесної монети) до 1 (ідеальний прогноз).

Коефіцієнт конкордації (Percent Concordant), чим вищий, тим кращій. Прагне до 1.

Побудовану модель логістичної регресії застосуємо по відношенню до самого тренінгового набору даних, та отримаємо значення прогнозів-оцінок. Отримані прогнози-оцінки – це значення ймовірностей, позначимо як F_Probability (назва змінної взята як комбінація двох слів: F – forecast (тобто прогноз), а Probability – ймовірність).

Таблиця 3.7 – Приклад результатів, що видає алгоритм

Номер запису	Час (listing_date)	X_1 Перший регресор (week_price_change)	...	X_M Останній регресор (mean_volume)	Цільова змінна (class)	Прогноз-оцінка Значення ймовірності. (F_Probability)
1	2021-11-16	20	...	4204035	0	0.2018
...	...	...	...	...	...	...
30	2022-05-10	-11	...	416058	1	0.5818

У зв'язку із тим, що при виводі теорії логістичної регресії використовувалися збалансовані вибірки даних, то в більшості статистичних пакетів використовується бал прийняття рішення, що дорівнює 0,5.

```
IF F_Probability >= 0,5 THEN F_Decision = 1;
ELSE F_Decision = 0;
```

В наведеному коді F_Decision – це прогноз-рішення, яке отримують на основі прогнозу-оцінки (F_Probability)., але в нашому випадку, вибірка не збалансована, і тому відповідний бал прийняття рішення необхідно визначати окремо, для чого була реалізована окрема програма, що підбирає порогове значення для прийняття рішення, тобто на основі правил обчислюється значення прогнозу-рішення F_Decision.

```
IF F_Probability >= 0,05 THEN F_Decision = 1;
ELSE F_Decision = 0;
```

Таблиця 3.8 – Значення F_Ddecision для гіпотези 1, коли CutOff = 0,05

Номер запису	Час (listing_date)	Цільова змінна (class)	Прогноз-оцінка. Значення ймовірності. (F Probability)	Прогноз-рішення. (F Decision)
1	2021-11-16	0	0.2018	1
...	...	...	...	...
30	2022-05-10	1	0.5818	1

Для розробленого алгоритму, визначення матриці похибок, обчислювалися наступні статистичні характеристики [10]:

- TP – True Positive;
- TN – True Negative;
- FN – False Negative;
- FP – False Positive.

На основі цих статистичних характеристик, також було обчислено, ще деякі додаткові характеристики, а саме:

- Recall – ця статистика більш відома як Sensitivity або TPR (True Positive Rate), та обчислюється за наступною формулою [10]: $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$;
- Precision – ця статистика також відома як PPV (Positive Predictive Value), та обчислюється за наступною формулою [10]: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$;
- F_Score – міра точності тесту. Він розраховується з точності та відкликання тесту, де точність – це кількість правильно ідентифікованих позитивних результатів, поділена на кількість усіх позитивних результатів, включаючи ті, які визначені неправильно, а відгук – це кількість правильно ідентифікованих позитивних результатів, поділена на кількість всіх зразків, які повинні були ідентифіковані як позитивні [10]. $\text{F_Score} = (2 * \text{TP}) / ((2 * \text{TP}) + \text{FP} + \text{FN})$;

- MISC – статистика помилкової класифікації. MISC – це скорочення з англійської від Misclassification Rate [10]. $MISC = 1 - (TP+TN)/(TP+TN+FP+FN)$.

Після того, як для кожної гіпотези, щодо значення CutOff будуть обчислені відповідні статистики, будується загальна таблиця значень похибок, як показано в таблиці нижче.

Таблиця 3.9 – Значення похибок при перевірці гіпотез, щодо різних параметрів CutOff

Номер гіпотези	CutOff	TP	TN	FN	FP	Recall	Precision	F_Score	MISC
1	0,05	15	0	1	14	0,9375	0,5172	0,6667	0,5000
2	0,1	15	2	1	12	0,9375	0,5556	0,6977	0,4333
3	0,15	15	2	1	12	0,9375	0,5556	0,6977	0,4333
4	0,2	15	3	1	11	0,9375	0,5769	0,7143	0,4000
5	0,25	15	6	1	8	0,9375	0,6522	0,7692	0,3000
6	0,3	14	7	2	7	0,8750	0,6667	0,7568	0,3000
7	0,35	14	9	2	5	0,8750	0,7368	0,8000	0,2333
8	0,4	14	9	2	5	0,8750	0,7368	0,8000	0,2333
9	0,45	14	9	2	5	0,8750	0,7368	0,8000	0,2333
10	0,5	14	10	2	4	0,8750	0,7778	0,8235	0,2000
11	0,55	14	10	2	4	0,8750	0,7778	0,8235	0,2000
12	0,6	11	11	5	3	0,6875	0,7857	0,7333	0,2667
13	0,65	10	12	6	2	0,6250	0,8333	0,7143	0,2667
14	0,7	8	14	8	0	0,5000	1,0000	0,6667	0,2667
15	0,75	7	14	9	0	0,4375	1,0000	0,6087	0,3000
16	0,8	6	14	10	0	0,3750	1,0000	0,5455	0,3333
17	0,85	5	14	11	0	0,3125	1,0000	0,4762	0,3667
18	0,9	4	14	12	0	0,2500	1,0000	0,4000	0,4000
19	0,95	1	14	15	0	0,0625	1,0000	0,1176	0,5000

На основі отриманої таблиці робиться сортування, по декільком змінним одночасно.

1. По-перше, за зменшенням F_Score.
2. По-друге, за зменшенням MISC.
3. По-третє, за зменшенням CutOff.

В результаті, перша строчка таблиці містить найкраще значення CutOff = 0,55, що відповідає отриманим результатам прогнозування, для побудованої моделі логістичної регресії, для навчального набору даних розміром $n=30$.

3.7 Графіки зміни значень коефіцієнтів моделі, по ітераціях.

Сенс побудови відповідних графіків, що наведені нижче, полягає в візуальному дослідженні поведінки коефіцієнтів моделі логістичної регресії, по мірі надходження нової інформації, в ідеалі вони не повинні різко змінюватися від ітерації до ітерації

По осі ОХ відкладається номер ітерації, по осі ОУ – значення відповідного коефіцієнта моделі. Відповідні значення оцінок параметрів моделі логістичної регресії, отримані для кожного кроку алгоритму, наведені в таблиці вище – BETAS_MERGE.

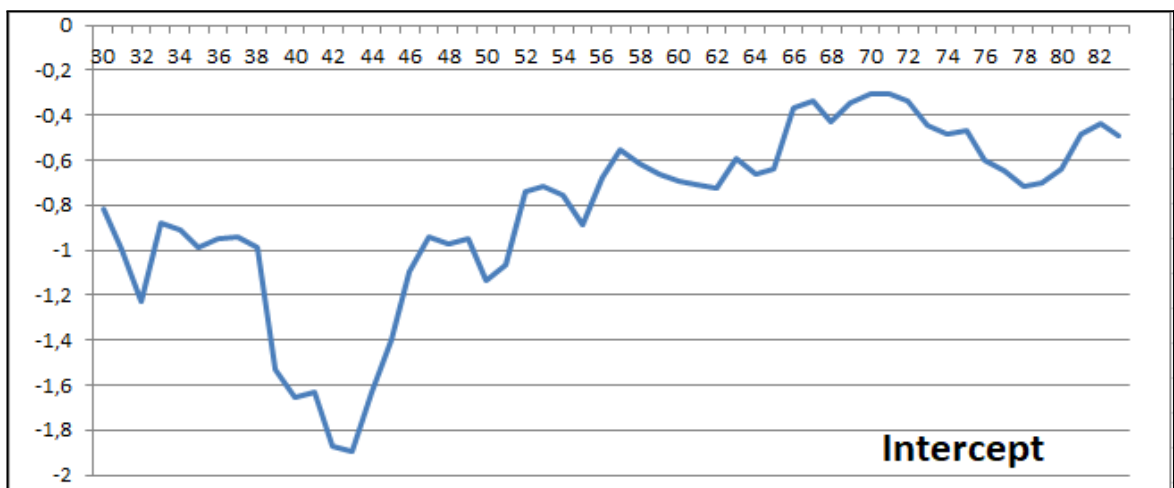


Рисунок 3.4 – Зміна значення коефіцієнта Intercept

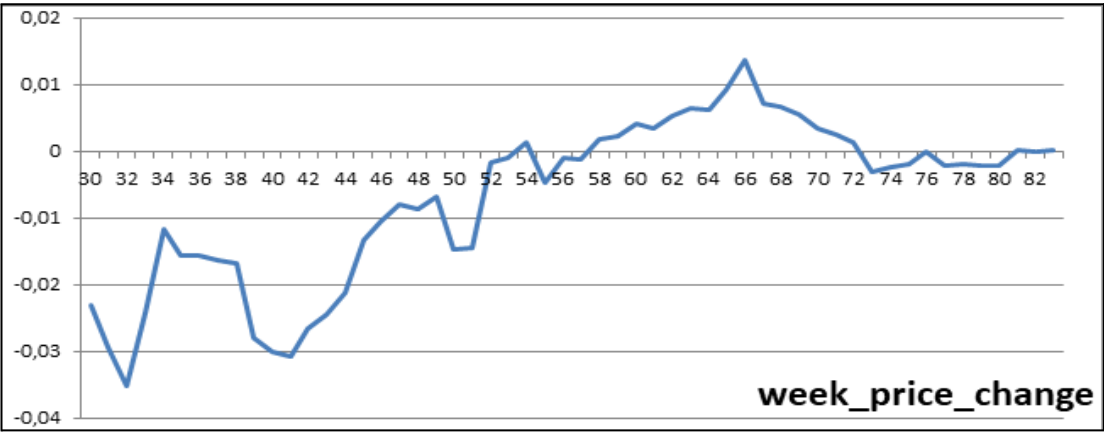


Рисунок 3.5 – Зміна значення коефіцієнта `week_price_change`

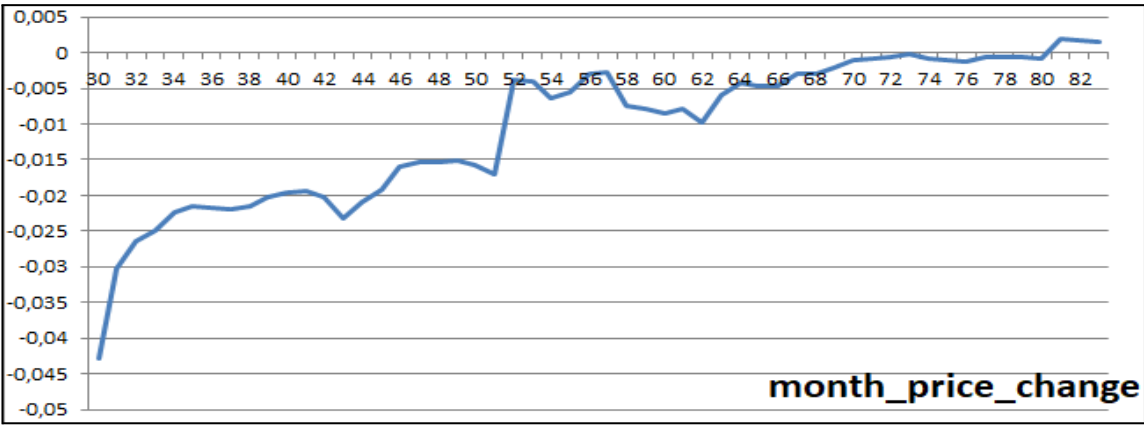


Рисунок 3.6 – Зміна значення коефіцієнта `month_price_change`

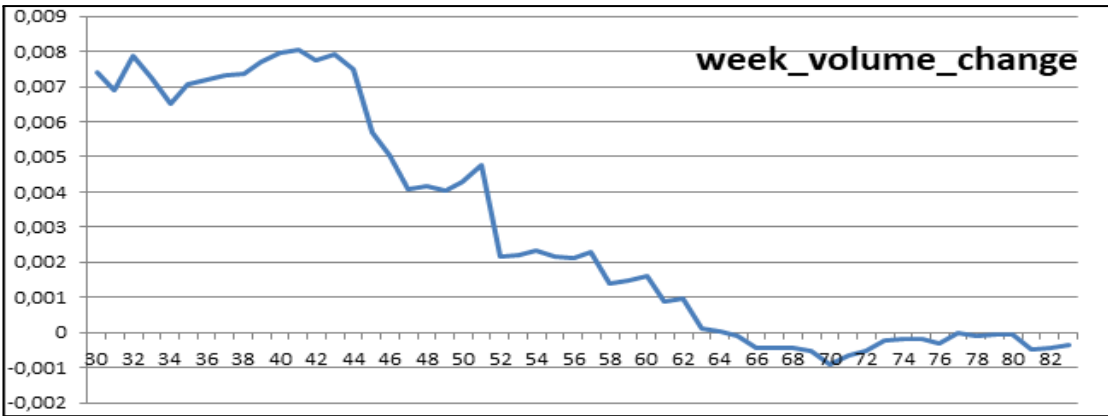


Рисунок 3.7 – Зміна значення коефіцієнта `week_volume_change`

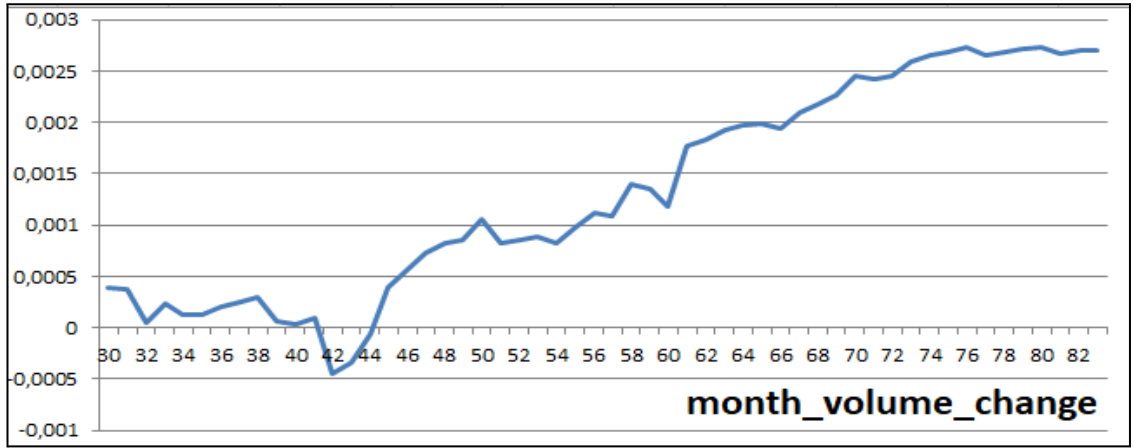


Рисунок 3.8 – Зміна значення коефіцієнта `month_volume_change`

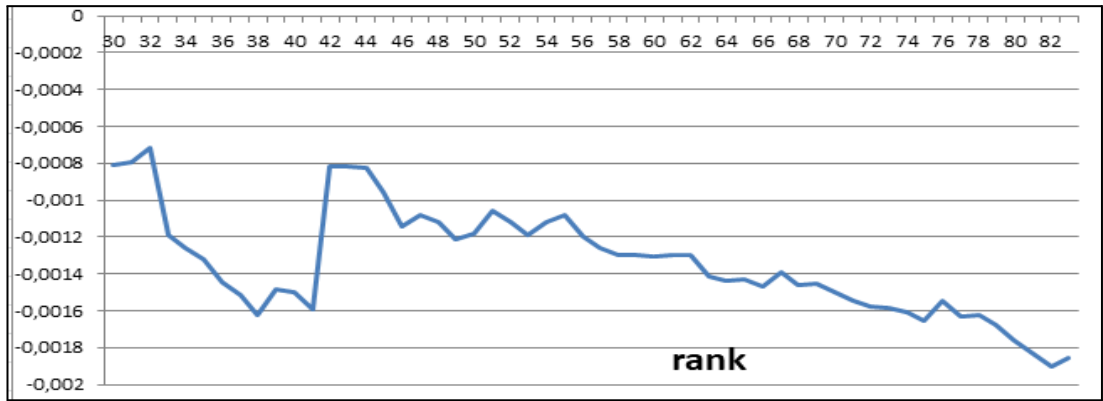


Рисунок 3.9 – Зміна значення коефіцієнта `rank`

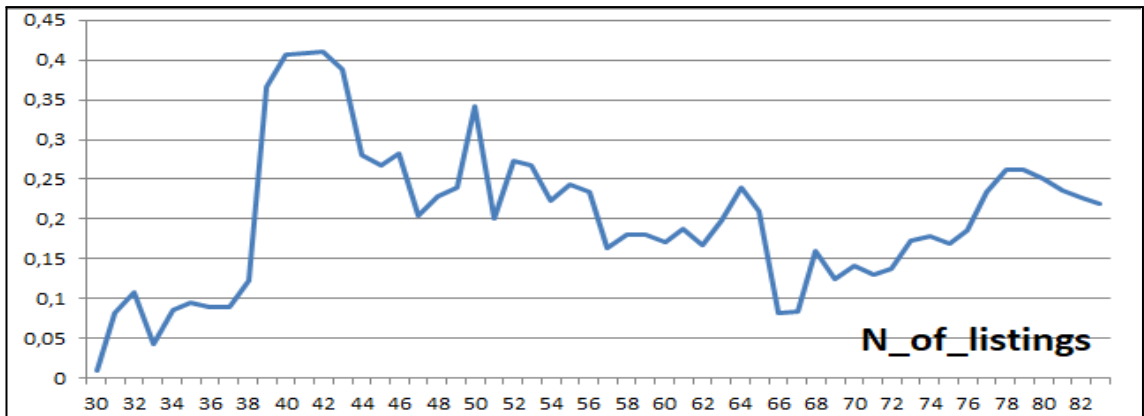


Рисунок 3.10 – Зміна значення коефіцієнта `N_of_listings`

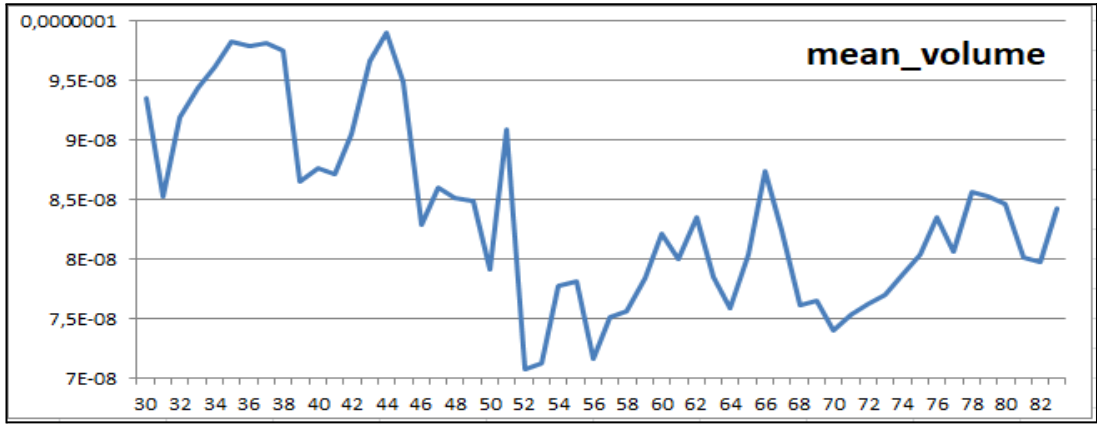


Рисунок 3.11 – Зміна значення коефіцієнта mean_volume

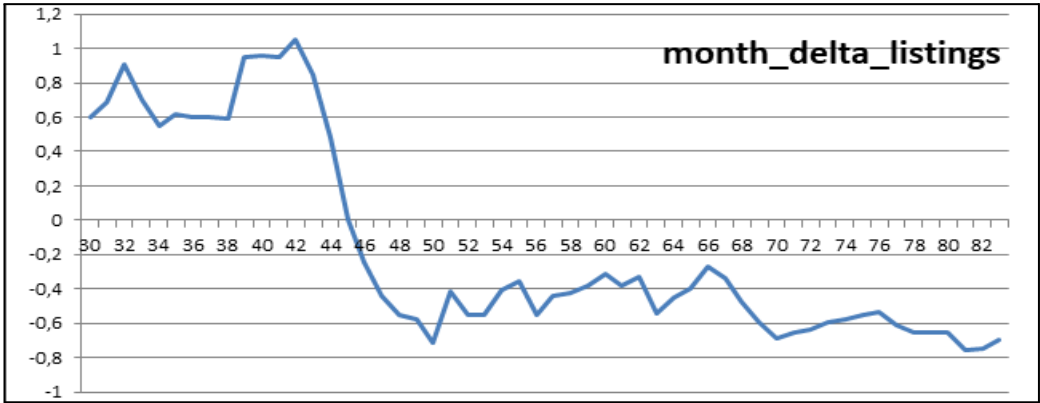


Рисунок 3.12 – Зміна значення коефіцієнта month_delta_listings

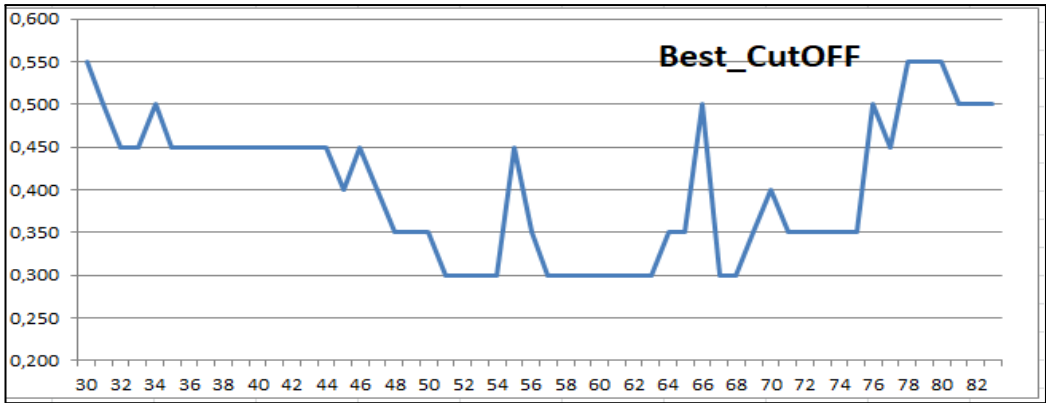


Рисунок 3.13 – Зміна значення коефіцієнта Best_CutOFF

3.8 Аналіз отриманих результатів

Набори вихідних даних.

В результаті виконання програми будуть сформовані наступні вихідні таблиці даних (які більш детально описані нижче):

- CUTOFF_BEST_ROW_MERGE – таблиця значень найкращих підібраних значень CutOff, на кожній ітерації алгоритму, з відповідними статистиками якості, отриманими на навчальному наборі даних.
- BETAS_MERGE – таблиця значень оцінок параметрів моделі логістичної регресії, отримані для кожного кроку алгоритму.
- SCORE_TEST_MERGE – таблиця значень прогнозів на один крок вперед, для кожної ітерації алгоритму. Прогнози отримані на основі відповідних побудованих моделей.

Рівняння математичної моделі логістичної регресії має наступний вигляд:

$$\begin{aligned} class = & - 0,8199 - 0,0232 \cdot x_1 - 0,0428 \cdot x_2 + 0,00741 \cdot x_3 + \\ & + 0,000396 \cdot x_4 - 0,00081 \cdot x_5 + 0,0106 \cdot x_6 - 9,35 \cdot 10^{-8} \cdot x_7 + 0,5976 \cdot x_8 \end{aligned}$$

де class – цільова змінна (1 – відбувся лістинг криптоактиву на біржі Binance, 0 – лістинг на біржі Binance не відбувся.);

x_1 – week_price_change, зміна ціни криптоактиву за останній тиждень у відсотках;

x_2 – month_price_change, зміна ціни криптоактиву за останній місяць у відсотках;

x_3 – week_volume_change, зміна обсягу обігу криптоактива у відсотках, за останній тиждень;

x_4 – month_volume_change, зміна обсягу обігу криптоактива у відсотках, за останній місяць;

x_5 – rank, рейтинг криптоактиву за капіталізацією;

x_6 – N_of_listings, число лістингів даного криптоактиву на інших біржах;

x_7 – mean_volume, середньодобовий обсяг обігу криптоактиву, за останній місяць;

x_8 – month_delta_listings, приріст за кількістю бірж, доданих за останній місяць.

Таблиця 3.10 – Прогнозуючі статистики моделі логістичної регресії

Статистика	Значення
TP	8
TN	22
FN	8
FP	16
Recall	0.5
Precision	0.3333
F_Score	0.4
MISC	0.4444

Вербальний опис змісту таблиці статистичних характеристик.

У 44,44% випадків відбувається невірна класифікація.

TP = 8 – у восьми випадках вірно прогнозується лістинг монети.

TN = 22 – у двадцяти двох випадках вірно прогнозується відсутність лістингу монети.

FN = 8 – у восьми випадках невірно прогнозується відсутність лістингу монети (втрачений потенційний прибуток).

FP = 16 – у шістнадцяти випадках невірно прогнозується лістинг монети (прямий збиток).

3.9 Розширена модель логістичної регресії із додатковими ознаками

Feature engineering (інженерія ознак) – це процес створення, перетворення або вибору релевантних характеристик (ознак) з даних для покращення роботи моделей машинного навчання.

В рамках нашого дослідження в якості такої ознаки, для поліпшення прогнозуючих якостей моделі логістичної регресії запропоновано використати наступні:

- 1) приналежність криптоактиву до однієї з п'яти базових категорій;
- 2) сантмент, емоційна забарвленість криптоактиву.

3.9.1 Змінна сантмент аналіз криптоактиву

Першою додатковою змінною є результат аналізу сантменту криптоактиву. Для отримання цієї змінної будемо також використовувати опис критпоактиву з джерела CoinMarketCap [24] та LLM модель Chat-gpt. Для цього був написаний промпт, який використовується для аналізу сантменту опису криптоактиву. Результатом аналізу буде значення від 0 до 1. Значення, близьке до 0, означає, що опис містить здебільшого нейтральні або критичні формулювання щодо самого токена, а близьке до 1 – що опис передає позитивне сприйняття характеристик та потенціалу активу.

Промпт для ChatGPT:

You are a crypto market analyst.

Analyze the sentiment of the following crypto project description.

Assign a sentiment score from 0 to 1, where:

0 = very negative sentiment (skeptical, pessimistic, uncertain)

0.5 = neutral or balanced (fact-based, no strong emotion)

1 = very positive sentiment (visionary, confident, optimistic)

Consider tone, confidence, visionary language, ecosystem strength, and the presence of innovation or hype.

Do not validate the claims – focus only on how positively or negatively the project is presented in the description.

Respond with decimal value only.

Отримана нова змінна – `sentiment_score`, що описує сантмент сприйняття криптоактиву, також включена в розширену модель логістичної регресії.

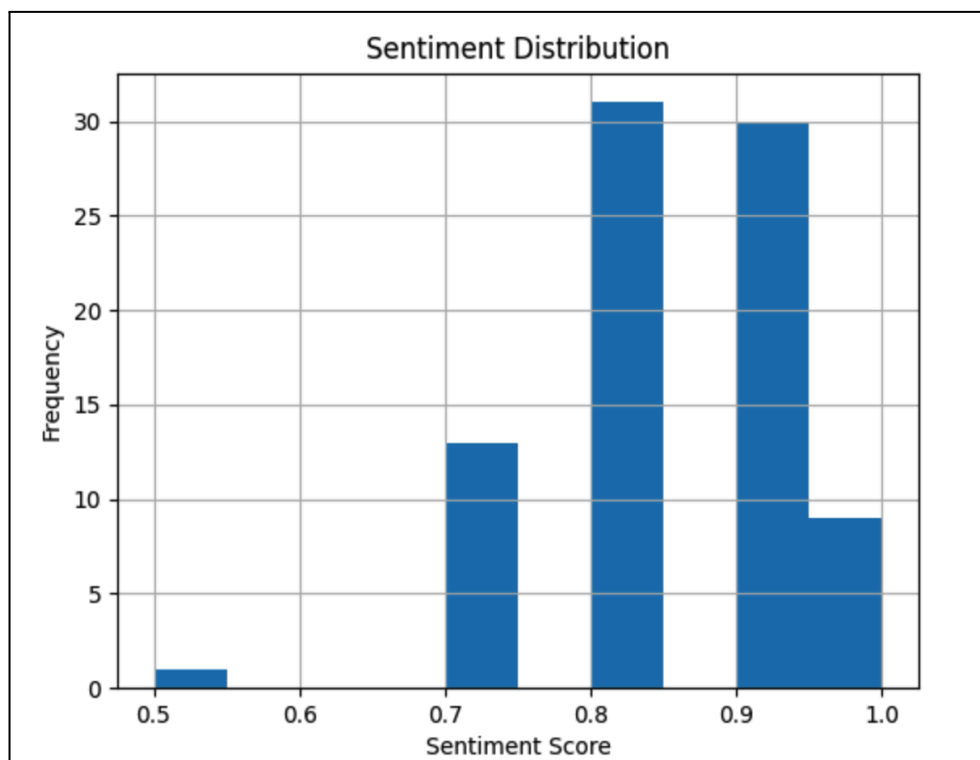


Рисунок 3.14 – Розподіл змінної `sentiment_score` по всій вибірці

3.9.2 Текстова ознака - категорія криптоактиву, що аналізується

По-друге, додамо до вхідних даних змінну яка буде характеризувати категорію до якої належить криптоактив. Для цього усі криптоактиви були розподілені на 5 базових категорій.

1. Layer 1 Protocol Tokens – криптоактив, що відноситься до протоколу блокчейна першого рівня.
2. Layer 2 / Scaling / Infrastructure Tokens – криптоактив, що відноситься до протоколу блокчейна другого рівня.
3. DeFi Tokens – криптоактив, що забезпечує функціонування DeFi платформ.
4. Utility & Ecosystem Tokens – утилітарні токени, що підтримують функціонування цілої блокчейн-екосистеми або мережі.
5. Stablecoins & Asset-backed Tokens – криптоактиви, вартість яких прив'язана до реальних активів або стабільних одиниць вартості (наприклад, долара США, золота чи товарів).

Для класифікації активів будемо використовувати опис критпоактивів, що взято з джерела CoinMarketCap [24].

Приклад опису криптовалюти PENDLE з сайту CoinMarketCap, що було взято за посиланням

<https://coinmarketcap.com/currencies/pendle/#About> :

Pendle is a protocol that enables the tokenization and trading of future yield. With the creation of a novel AMM that supports assets with time decay, Pendle gives users more control over future yield by providing optionality and opportunities for its utilization.

Для класифікації використовується LLM модель Chat-GPT. Для цього був написаний промпт (набір вказівок) та використано API Chat-gpt. З повним результатом класифікації можна ознайомитися у Додатку Б. Маючи

класифікацію, створимо 5 додаткових змінних у датасеті використовуючи технологію One-hot Encoding [25]:

- is_layer_1 – приймає два стани 0 – ні, 1- так;
- is_layer_2 – приймає два стани 0 – ні, 1- так;
- is_defi – приймає два стани 0 – ні, 1- так;
- is_utility – приймає два стани 0 – ні, 1- так;
- is_stablecoin – приймає два стани 0 – ні, 1- так.

Для кожного криптоактиву у датасеті змінна категорії до якої відноситься цей актив є 1, а всі інші дорівнюють 0.

3.9.3 Побудова розширеної моделі

В табл. 3.11 наведені отримані оцінки коефіцієнтів розширеної моделі логістичної регресії, що включає ознаки – приналежність криптоактиву до однієї з п'яти базових категорій та сантмент, емоційна забарвленість криптоактиву.

Таблиця 3.11 – Значення оцінок коефіцієнтів розширеної моделі

Параметр моделі		DF	Estimate	Standard	Wald	Pr > ChiSq
				Error	Chi-Square	
Intercept		1	4,7586	153,4	0,001	0,9753
week_price_change		1	0,000704	0,0301	0,0005	0,9813
month_price_change		1	-0,0013	0,0103	0,0159	0,8998
week_volume_change		1	-0,00051	0,00363	0,02	0,8875
month_volume_change		1	-0,00336	0,00192	3,055	0,0805
rank		1	0,00159	0,000825	3,716	0,0539
N_of_listings		1	-0,2735	0,2527	1,171	0,2792
mean_volume		1	-1,33E-07	7,24E-08	3,3828	0,0659
month_delta_listings		1	0,6148	0,8645	0,5057	0,477
sentiment		1	-1,6724	3,2316	0,2678	0,6048
categories	1	1	-1,153	153,4	0,0001	0,994
categories	2	1	-3,1001	153,4	0,0004	0,9839
categories	3	1	-3,8233	153,4	0,0006	0,9801
categories	4	1	-1,7629	153,4	0,0001	0,9908

Таблиця 3.12 – Схема формування проектних змінних на основі параметра моделі Categories, категорія криптоактиву

Class Level Information					
Class	Value	Design Variables			
categories	1	1	0	0	0
	2	0	1	0	0
	3	0	0	1	0
	4	0	0	0	1
	5	-1	-1	-1	-1

Рівняння математичної моделі логістичної регресії має наступний вигляд:

$$\begin{aligned}
 class = & 4,7586 + 0,000704 \cdot x_1 - 0,0013 \cdot x_2 - 0,00051 \cdot x_3 - \\
 & - 0,00336 \cdot x_4 + 0,00159 \cdot x_5 - 0,2735 \cdot x_6 - 1,34 \cdot 10^{-7} \cdot x_7 + 0,6148 \cdot x_8 - \\
 & - 1,6724 \cdot x_9 - 1,153 \cdot x_{10} - 3,1001 \cdot x_{11} - 3,8233 \cdot x_{12} - 1,7629 \cdot x_{13}
 \end{aligned}$$

де class – цільова змінна (1 – відбувся лістинг криптоактиву на біржі Binance, 0 – лістинг на біржі Binance не відбувся.);

x_1 – week_price_change, зміна ціни криптоактиву за останній тиждень у відсотках;

x_2 – month_price_change, зміна ціни криптоактиву за останній місяць у відсотках;

x_3 – week_volume_change, зміна обсягу обігу криптоактива у відсотках, за останній тиждень;

x_4 – month_volume_change, зміна обсягу обігу криптоактива у відсотках, за останній місяць;

x_5 – rank, рейтинг криптоактиву за капіталізацією;

x_6 – N_of_listings, число лістингів даного криптоактиву на інших біржах;

x_7 – mean_volume, середньодобовий обсяг обігу криптоактиву, за останній місяць;

x_8 – month_delta_listings, приріст за кількістю бірж, доданих за останній місяць;

x_9 – сентименту криптоактиву;

x_{10} – це проектна зміна categories = 1, що означає приналежність криптоактиву до протоколу блокчейна першого рівня (Layer 1 Protocol Tokens);

x_{11} – це проектна зміна categories = 2, що означає приналежність криптоактиву до протоколу блокчейна другого рівня (Layer 2 / Scaling / Infrastructure Tokens);

x_{12} – це проектна зміна categories = 3, означає що токен забезпечує функціонування DeFi платформ (DeFi Tokens);

x_{13} – це проектна зміна categories = 4, означає утилітарний токен, що підтримують функціонування цілої блокчейн-екосистеми або мережі (Utility & Ecosystem Tokens).

Таблиця 3.13 – Статистичні характеристики – оцінка співвідношення шансів (Odds Ratio_, оцінок параметрів розширеної моделі

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
week_price_change	1,001	0,943	1,062
month_price_change	0,999	0,979	1,019
week_volume_change	0,999	0,992	1,007
month_volume_change	0,997	0,993	1
rank	1,002	1	1,003
N_of_listings	0,761	0,464	1,248
month_delta_listings	1,849	0,34	10,067
mean_volume	1	1	1
sentiment	0,188	<0,001	105,785
categories 1 vs 5	<0,001	<0,001	>999,999
categories 2 vs 5	<0,001	<0,001	>999,999
categories 3 vs 5	<0,001	<0,001	>999,999
categories 4 vs 5	<0,001	<0,001	>999,999

Таблиця 3.14 – Статистичні характеристики – коефіцієнти конкордації, для прогнозів отриманих на основі побудованої моделі

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	86,1	Somers' D	0,721
Percent Discordant	13,9	Gamma	0,721
Percent Tied	0	Tau-a	0,344
Pairs	1664	c	0,861

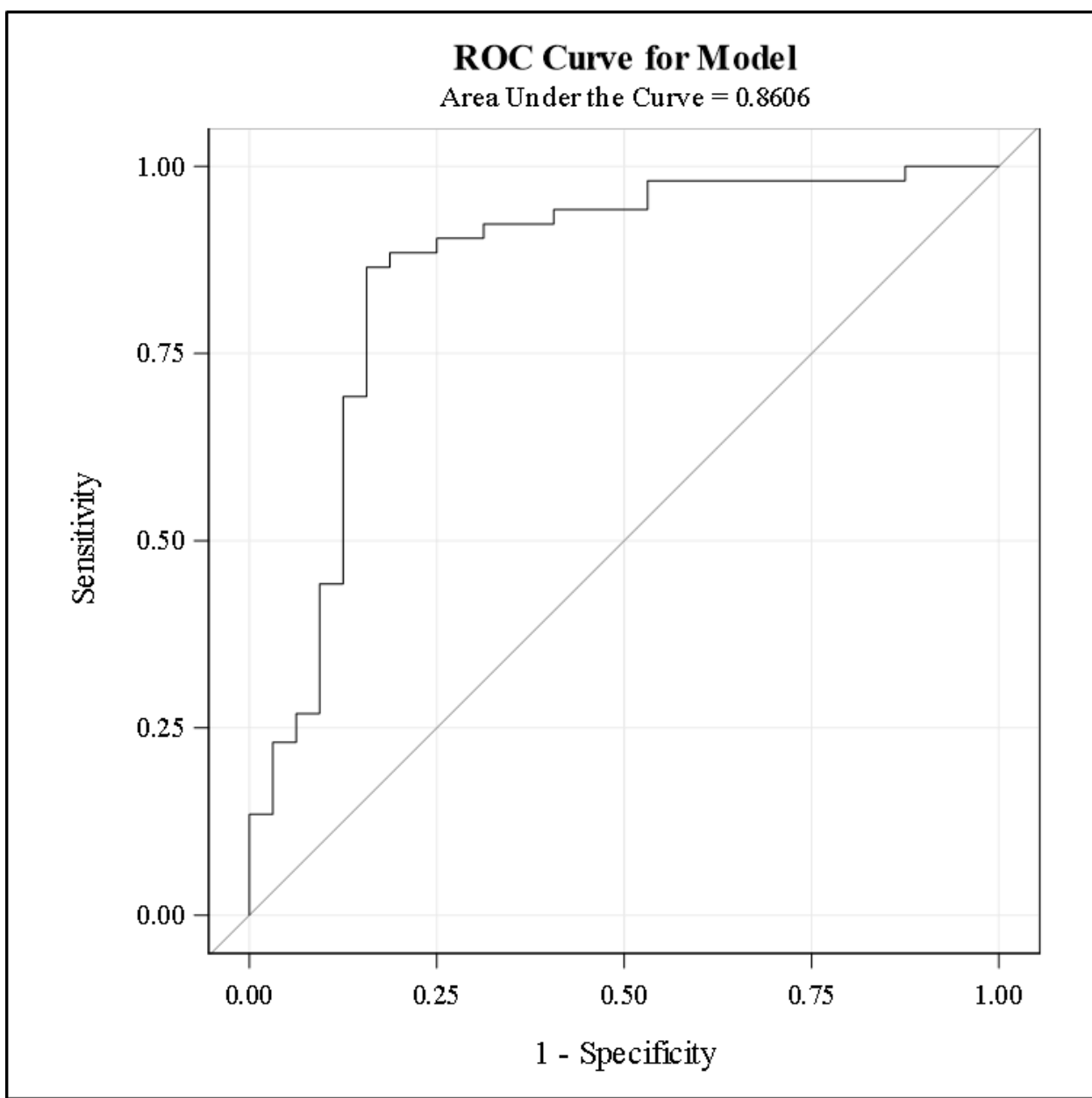


Рисунок 3.15 – Графік ROC-кривої для розширеної моделі логістичної регресії

В табл. 3.15 наведені отримані прогнозуючі статистичні характеристики побудованої розширеної моделі логістичної регресії, а також характеристики раніше побудованої базової моделі.

З отриманих результатів можна зробити висновок, що додавання додаткових ознак в модель суттєво підвищує її прогнозуючі якості.

Таблиця 3.15 – Прогнозуючі статистики моделі логістичної регресії

Статистика	Модель логістично регресії	Розширена модель логістичної регресії
TP	8	14
TN	22	28
FN	8	4
FP	16	6
Recall	0.5	0,78
Precision	0.3333	0,7
F_Score	0.4	0,74
MISC	0.4444	0,22

Вербальний опис змісту таблиці статистичних характеристик.

В розширеній моделі у 22% випадків відбувається невірна класифікація, в той час як в базовій цей показник складав 44,44%.

$TP = 14$ – у чотирнадцяти випадках вірно прогнозується лістинг монети, в той час як попередня модель лише у 8 випадках.

$TN = 28$ – у двадцяти восьми випадках вірно прогнозується відсутність лістингу монети, для попередньої моделі цей показник дорівнював 22.

$FN = 4$ – у чотирьох випадках невірно прогнозується відсутність лістингу монети (втрачений потенційний прибуток), для попередньої моделі цей показник дорівнював 8.

$FP = 6$ – у шести випадках невірно прогнозується лістинг монети (прямий збиток), для попередньої моделі цей показник дорівнював 16.

Висновки до розділу 3

У цьому розділі виконано практичну частину дослідження, що включала збір даних із різних джерел, побудову змінних, навчання моделі та аналіз результатів. Було деталізовано вибір даних і описано кожну змінну,

зокрема особливості формування ключової ознаки “тижнева зміна ціни” та додаткових змінних, як-от категорія криптоактиву й результати сентимент-аналізу. Було наведено покроковий опис алгоритму моделювання, що враховує підготовку даних, налаштування параметрів ітераційного процесу, а також методику оцінювання коефіцієнтів на кожному кроці. Для візуалізації було побудовано графіки, які демонструють зміну значень параметрів моделі в ході ітерацій, що дозволило простежити вплив різних факторів на кінцевий результат.

Було побудовано 2 моделі, перша базова модель, та друга розширена додатковими параметрами.

Аналіз результатів показав, що обидві моделі змогли дати гарний рівень точності прогнозування. Перша модель демонструвала достатньо точні прогнози, а саме 56% точності. Друга модель, яка мала у вхідних даних додаткові показники видала значно кращі результати, а саме 78% точності. Її результати прогнозування були більш точними і будувалися на більшій кількості факторів, що впливають на прийняття рішення про лістинг криптовалюти.

Отримані результати доводять, що додавання нових змінних, а саме даних про криптовалюту, до моделі сприяє покращенню точності прогнозування. Модифікована модель, яка враховувала ці додаткові показники, виявилась більш ефективною та надійною для передбачення лістингу нових криптовалют на Binance.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі буде проведено оцінювання основних характеристик для майбутнього програмного продукту для прогнозування факту лістингу криптовалют на біржі Binance.

Дана реалізація буде сприяти проведенню усіх необхідних досліджень, що дасть змогу якісно дослідити питання не лише в Україні, проте у всьому світі.

Також в даному дослідженні показано різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, що має вплив на економічні фактори та сумісність з майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих годин, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки програми прогнозування факту лістингу криптовалют на біржі Binance. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по компанії.

Технічні вимоги до програмного продукту є наступними:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка можливого програмного продукту, яка дозволяє зчитувати, перекладати текст з зображень і вставляти переклад на оригінальне зображення. Беручи за основу цю функцію, можна виділити наступні:

- F_1 – вибір мови програмування;

- F_2 – вибір основної бібліотеки розпізнавання тексту з зображення;
- F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації.

Функція F_1 :

- а) Python;
- б) C++.

Функція F_2 :

- а) Застосування готових бібліотек;
- б) Написання власних функцій.

Функція F_3 :

- а) Visual Studio Code;
- б) Jupyter Notebook.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

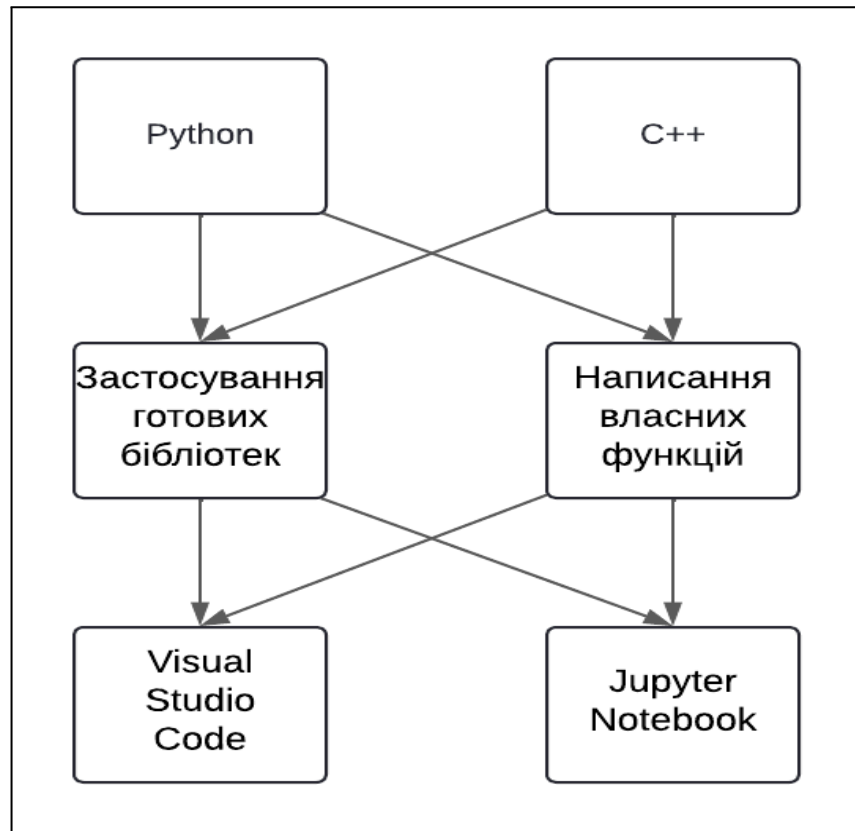


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в табл. 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	<i>Python</i>	Велика кількість бібліотек, висока швидкість і простота розробки програми	Низька швидкість роботи програми
	<i>C++</i>	Висока швидкість роботи програми	Мала кількість готових бібліотек для роботи з зображеннями та нейронними мережами, складний синтаксис і великий час розробки
F_2	Застосування готових бібліотек	Спрощення процесу написання та підтримки Широкий вибір функцій і методів	Можлива невідповідність поставленим цілям
	Написання власних функцій	Можливість персоналізації -Гнучкість та адаптивність	Додаткове використання часового ресурсу Більший ризик помилок
F_3	<i>Visual Studio Code</i>	Велика кількість доповнень, гнучкість у налаштуваннях, інтелектуальне автозаповнення, підтримка багатьох мов програмування	Вибагливість до комп'ютера користувача
	<i>Jupyter Notebook</i>	Підтримка багатьох мов програмування, легкість використання	Мала кількість додаткових інструментів для полегшення розробки

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 . Перевагу надаємо швидкості розробки та простоті. Python має дуже велику кількість бібліотек і інструментів, які зроблять розробку програми простішим і комфортнішим.

Функція F_2 . Обидва інструменти задовольняють вимогам до програми.

Можливо використати варіанти А чи Б.

Функція F_3 . Перевагу надаємо більшій кількості додатків та розширень Visual Studio Code, які можуть значно прискорити та спростити розробку програмного застосунку.

Таким чином, будемо розглядати такий варіанти реалізації ПП:

$$F_1a - F_2a - F_3a,$$

$$F_1a - F_2b - F_3a.$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих раніше, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

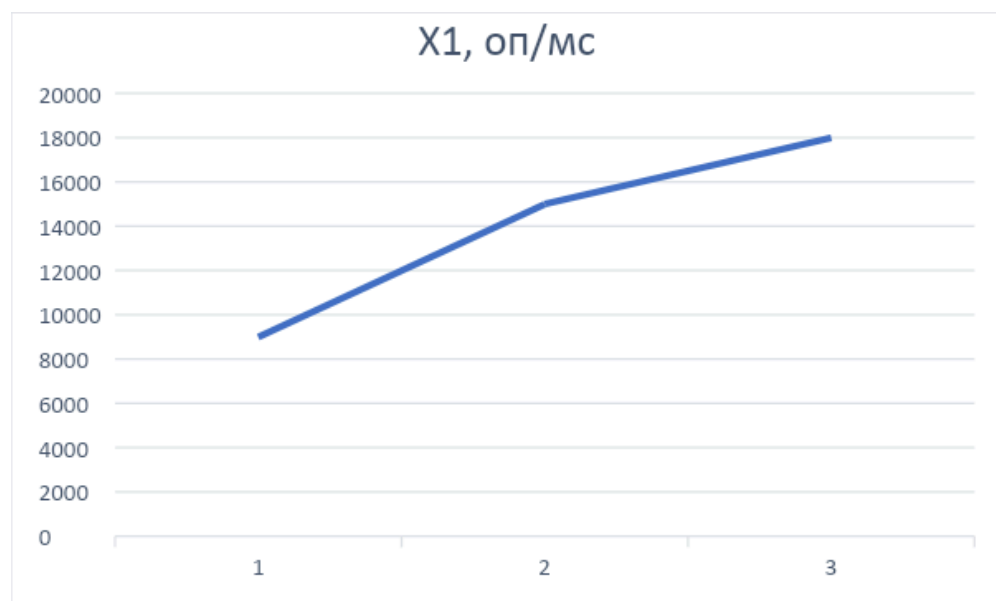
- $X1$ – швидкодія мови програмування;
- $X2$ – об'єм пам'яті для обчислень та збереження даних;
- $X3$ – час роботи моделі;
- $X4$ – потенційний об'єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у табл. 4.2.

Таблиця 4.2 – Основні параметри програмного продукту

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови	X1	оп/мс	9000	15000	18000
Об'єм пам'яті	X2	Мб	600	400	200
Час роботи моделі	X3	с	7	4	2
Потенційний об'єм програмного коду	X4	кількість рядків коду	1800	1400	1000

За даними табл. 4.3 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

**Рисунок 4.2** – X1, швидкодія мови програмування

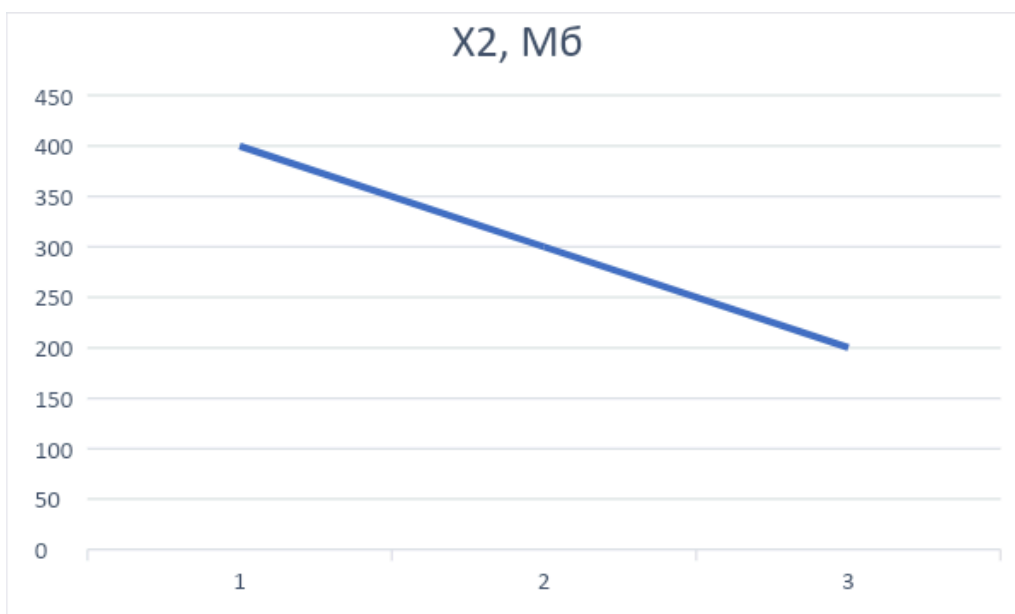


Рисунок 4.3 – X2, об'єм пам'яті

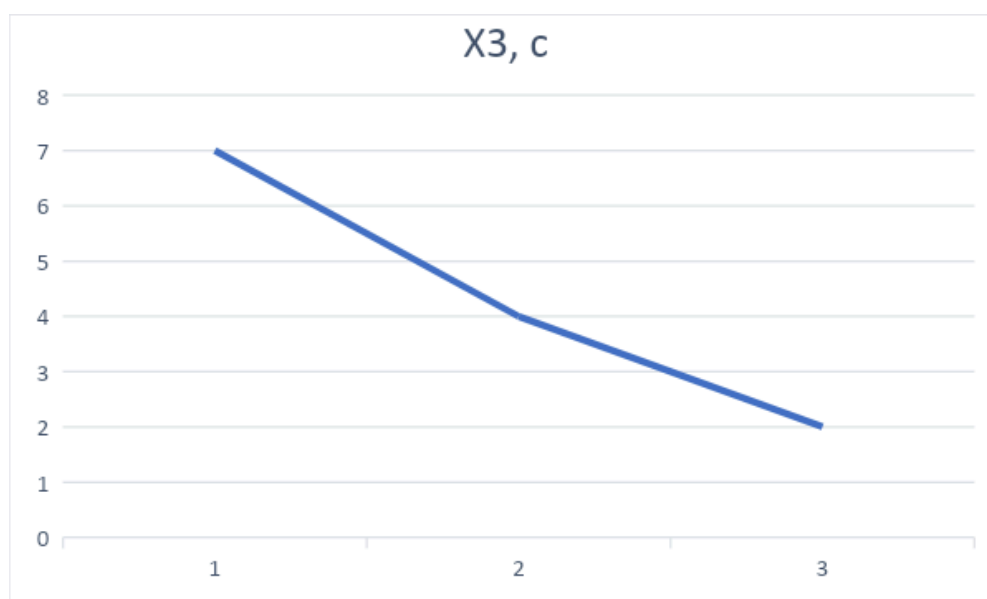


Рисунок 4.4 – X3, час роботи моделі

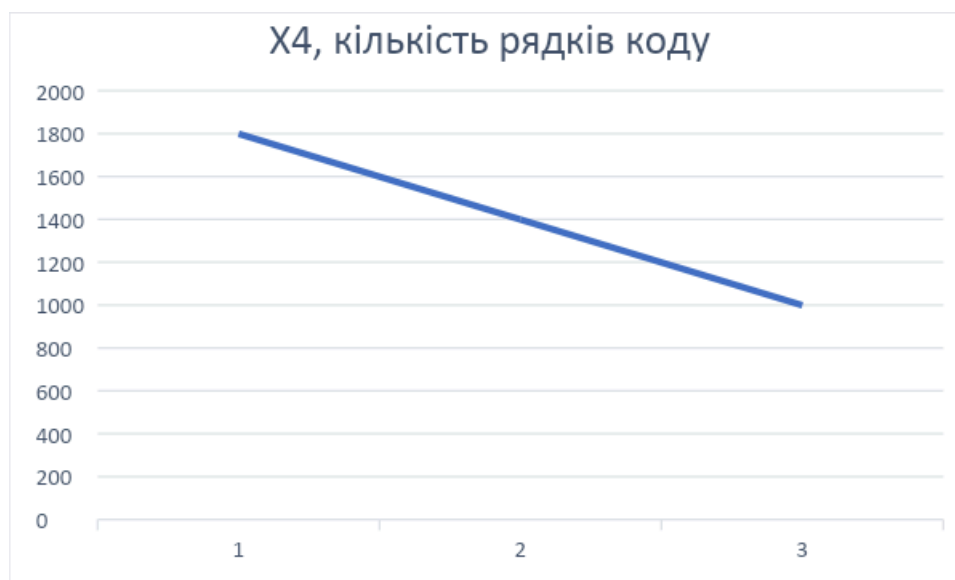


Рисунок 4.5 – X4, потенційний об'єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;

- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у табл. 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
$X1$	Швидкодія мови програмування	Оп/мс	1	2	2	1	2	1	2	11	-6,5	42,25
$X2$	Об'єм пам'яті	МВ	3	4	3	3	4	3	4	24	6,5	42,25
$X3$	Час роботи моделі	с	2	1	1	2	1	2	1	10	-7,5	56,25
$X4$	Потенційний об'єм програмного коду	Кількість рядків коду	4	3	4	4	3	4	3	25	7,5	56,25
	Разом		10	10	10	10	10	10	10	70	0	197

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

Сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.1)$$

де N – число експертів;

n – кількість параметрів.

Середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5. \quad (4.2)$$

Відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (4.3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

Загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 197. \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 197}{7^2(4^3-4)} = 0,804 > W_k = 0,67. \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у табл. 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметр <i>и</i>	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	<	<	<	<	<	0,5
X1 і X3	<	>	>	<	>	<	>	>	1,5
X1 і X4	<	<	<	<	<	<	<	<	0,5
X2 і X3	>	>	>	>	>	>	>	>	1,5
X2 і X4	<	>	<	<	>	<	>	<	0,5
X3 і X4	<	<	<	<	<	<	<	<	0,5

Числове значення, що визначає ступінь переваги *i*-го параметра над *j*-тим, a_{ij} визначається по формулі:

$$a_{ij} = \{1.5 \text{ при } X_i > X_j, 1.0 \text{ при } X_i = X_j, 0.5 \text{ при } X_i < X_j\}. \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \|a_{ij}\|$.

Для кожного параметра зробимо розрахунок вагомості $K_{\text{вi}}$ за наступними формулами:

$$K_{\text{вi}} = \frac{b_i}{\sum_{i=1}^n b_i}, \quad (4.7)$$

$$b_i = \sum_{i=1}^N a_{ij}. \quad (4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K'_{\text{вi}} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (4.9)$$

$$b'_i = \sum_{i=1}^N a_{ij} b'_j. \quad (4.10)$$

Як видно з табл. 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер.	
	$X1$	$X2$	$X3$	$X4$	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
X1	1	0,5	1,5	0,5	3,5	0,2188	12,25	0,2076	44,875	0,2078
X2	1,5	1	1,5	0,5	4,5	0,2813	16,25	0,2754	59,125	0,2737
X3	0,5	0,5	1	0,5	2,5	0,1563	9,25	0,1568	34,125	0,1580
X4	1,5	1,5	1,5	1	5,5	0,3438	21,25	0,3602	77,875	0,3605
Всього:					16	1	59	1	216	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів $X2$ (об'єм пам'яті), $X3$ (час зчитування тексту) та $X4$ (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра $X1$ (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (табл. 4.6):

$$K_K(j) = \sum_{i=1}^n K_{Bi,j} B_{i,j}, \quad (4.11)$$

де n – кількість параметрів;

K_{Bi} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

<i>Основні функції</i>	<i>Варіант реалізації функції</i>	<i>Пара-метри</i>	<i>Абсолютне значення параметра</i>	<i>Бальна оцінка параметра</i>	<i>Коефіцієнт вагомості параметра</i>	<i>Коефіцієнт рівня якості</i>
F1	A	X1	16000	7	0,21	1,47
F2	A	X2	530	3	0,27	0,81
	Б	X3	3	8	0,16	1,28
F3	A	X4	1100	9	0,36	3,24

За даними з табл. 4.6 за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \dots + K_{\text{ТУ}}[F_{zk}], \quad (4.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1,47 + 0,81 + 3,24 = 5,52,$$

$$K_{K2} = 1,47 + 1,28 + 3,24 = 5,99.$$

Як видно з розрахунків, кращим є 2 варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки програмного продукту

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання.

1. Розробка проекту програмного продукту.

2. Розробка програмної оболонки.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М'} \quad (4.13)$$

де T_P – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 60$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{\Pi} = 1.6$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.7$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 60 \cdot 1.6 \cdot 0.7 = 67,2 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_p = 40$ людино-днів, $K_{II} = 0.9$, $K_{СК} = 1$, $K_{СТ} = 0.75$:

$$T_2 = 40 \cdot 0.9 \cdot 0.75 = 27 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (67,2 + 27 + 4.8 + 27) \cdot 8 = 1008 \text{ людино-годин.}$$

$$T_{II} = (67,2 + 27 + 6.91 + 27) \cdot 8 = 1024,88 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці бере участь один програміст-аналітик з окладом 26800 грн. Визначимо середню зарплату за годину за формулою:

$$СЧ = \frac{M}{T_m \cdot t} \text{ грн,} \quad (4.14)$$

де M – місячний оклад працівників;

T_m – кількість робочих днів тиждень;

t – кількість робочих годин в день.

$$СЧ = \frac{26800}{21 \cdot 8} = 159,52 \text{ грн.} \quad (4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$СЗП = C_{\text{ч}} \cdot T_i \cdot КД, \quad (4.16)$$

де C_q – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

K_d – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$\text{I. } C_{3П} = 159,52 \cdot 1008 \cdot 1.2 = 192955,39 \text{ грн.}$$

$$\text{II. } C_{3П} = 159,52 \cdot 1024,88 \cdot 1.2 = 196186,63 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I. } C_{ВІД} = C_{3П} \cdot 0.22 = 192955,39 \cdot 0.22 = 42450,18 \text{ грн.}$$

$$\text{II. } C_{ВІД} = C_{3П} \cdot 0.22 = 196186,63 \cdot 0.22 = 43161,05 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години (C_M).

Так як одна ЕОМ обслуговує одного програміста з окладом 26800 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{Г} = 12 \cdot M \cdot K_3 = 12 \cdot 26800 \cdot 0,2 = 64320 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{Г} \cdot (1 + K_3) = 64320 \cdot (1 + 0.2) = 77184 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0.22 = 77184 \cdot 0,22 = 16980,48 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 35500 грн.

$$C_A = K_{TM} \cdot K_A \cdot C_{PP} = 1.2 \cdot 0.25 \cdot 35500 = 10650 \text{ грн},$$

де K_{TM} – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

C_{PP} – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot C_{PP} \cdot K_P = 1.2 \cdot 35500 \cdot 0.07 = 2982 \text{ грн},$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{EF} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.88 = \\ &= 1640,32 \text{ години}, \end{aligned}$$

де D_K – календарна кількість днів у році;

D_B, D_C – відповідно кількість вихідних та святкових днів;

D_P – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{EL} = T_{EF} \cdot N_C \cdot K_3 \cdot C_{EH} = 1640,32 \cdot 0,35 \cdot 0,88 \cdot 9,34 = 4719 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

C_{EH} – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 35500 \cdot 0.67 = 23785 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H, \quad (4.17)$$

$$C_{\text{ЕКС}} = 77184 + 16980,48 + 10650 + 2982 + 4719 + 23785 = 136300,48 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 136300,48 / 1640,32 = 83,09 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{\text{М-Г}} \cdot T, \quad (4.18)$$

1. $C_M = 83,09 \cdot 1008 = 83754,72 \text{ грн,}$
2. $C_M = 83,09 \cdot 1024,88 = 85157,28 \text{ грн.}$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{\text{ЗП}} \cdot 0,67, \quad (4.19)$$

1. $C_H = 192955,39 \cdot 0,67 = 129280,11 \text{ грн,}$
2. $C_H = 196186,63 \cdot 0,67 = 131445,04 \text{ грн.}$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{пп}} = C_{\text{зп}} + C_{\text{вїд}} + C_{\text{м}} + C_{\text{н}}, \quad (4.20)$$

1. $C_{\text{пп}} = 192955,39 + 42450,18 + 82474,56 + 129280,1 = 447160,23$ грн,
2. $C_{\text{пп}} = 196186,63 + 43161,05 + 83855,68 + 131445,04 = 454648,4$ грн.

4.7 Вибір кращого варіанту програмного продукту техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}j} = K_{\text{К}j} / C_{\text{Ф}j}, \quad (4.21)$$

$$K_{\text{ТЕР}1} = 5,52 / 447160,23 = 12,3446 \cdot 10^{-6},$$

$$K_{\text{ТЕР}2} = 5,99 / 454648,4 = 13,175 \cdot 10^{-6}.$$

Як бачимо, найбільш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{ТЕР}2} = 13,175 \cdot 10^{-6}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є другий варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{ТЕР}} = 13,175 \cdot 10^{-6}$.

Цей варіант реалізації програмного продукту має такі параметри:

- вибір мови програмування – Python;
- інструменти для розробки – Застосування готових бібліотек;
- середовище розробки – Visual Studio Code.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку реалізацію програми та доступний функціонал для роботи.

Висновки до розділу 4

У даному розділі було проведено повний функціонально-вартісний аналіз програмного продукту, що включав всебічний розгляд та оцінку його основних функцій. Цей аналіз дозволив ідентифікувати всі ключові аспекти програмного продукту, зокрема, його функціональні можливості, ефективність виконання поставлених завдань та економічну доцільність розробки.

Зокрема, було здійснено детальну оцінку кожної з основних функцій програмного продукту, з метою визначення їхнього внеску у загальну ефективність та вартість розробки. Цей процес включав вивчення технічних характеристик, оцінку продуктивності, аналіз витрат на розробку та підтримку, а також потенційних вигод від впровадження кожної функції.

У результаті виконання функціонально-вартісного аналізу програмного комплексу, що розроблюється, було ідентифіковано ключові функції програмного продукту та проведено їх детальну оцінку. Цей процес дозволив знайти параметри, які найкраще характеризують програмний продукт з точки зору його функціональності, вартості та ефективності. Виявлені параметри стали основою для подальшого вдосконалення продукту та оптимізації процесів його розробки.

На основі проведеного аналізу було обрано найбільш оптимальний варіант реалізації програмного продукту. Цей вибір ґрунтувався на ретельному порівнянні різних варіантів реалізації, з урахуванням результатів функціонально-вартісного аналізу, що забезпечило максимально ефективне використання ресурсів та досягнення поставлених цілей розробки. Також були розглянуті різні сценарії розвитку продукту, що дозволило визначити найбільш перспективні напрямки для подальшої роботи.

Таким чином, виконаний функціонально-вартісний аналіз не лише забезпечив комплексне розуміння основних функцій програмного продукту, але й надав можливість обрати оптимальний шлях його реалізації, враховуючи всі важливі аспекти та характеристики. Це сприяє підвищенню якості кінцевого продукту, зниженню витрат на його розробку та збільшенню загальної ефективності проекту.

ВИСНОВКИ

Під час цієї роботи було досліджено процес лістингу криптоактивів на криптобіржі та вплив цієї події на ціну, також було вивчено можливість прогнозування факту лістингу на криптобіржі Binance на основі різнотипних показників. Прогнозування лістингу криптоактивів є надзвичайно актуальним завданням через велику кількість нових криптоактивів, що з'являються кожного дня. Було визначено які фактори суттєво впливають на прийняття рішення про лістинг криптоактиву, таких як ринкові фактори (зміна ціни, об'єм торгів активу), соціальні фактори та інші.

Застосування логістичної регресії у поєднанні з методами sentiment аналізу дало змогу створити ефективні моделі, здатні прогнозувати лістинг нових криптовалют на біржі Binance. Тестування моделей на історичних даних показало їх здатність передбачати майбутні лістинги з прийнятною точністю. Отримані результати прогнозування можна вважати сприятливими, зважаючи на оцінку точності моделей.

Для розробки була використана мова програмування Python, через її простоту, можливість швидко писати код. Ця мова є найбільш використовуваною для задач аналізу даних та побудови моделей.

Регулярна валідація, оновлення моделі та аналіз помилок дозволяють адаптувати моделі до динамічних умов криптовалютного ринку та підвищують їх практичну цінність.

Результати цієї роботи показують значимість використання моделей для прогнозування різноманітних подій на криптовалютних ринках, оскільки це може надати перевагу над іншими учасниками ринку.

Перспективи щодо подальших досліджень. В подальшому у рамках обраної теми дослідження пропонується:

- 1) розширити список джерел даних на основі яких можна розробити додаткові інформаційні ознаки для побудови математичних моделей;
- 2) для вирішення прогнозування факту лістингу застосувати додаткові методи, а саме нейронні мережі, дерева рішення, метод групового урахування аргументів та інші методи штучного інтелекту;
- 3) для підвищення прогнозуючих якостей моделей поліпшити промпт запиту для LLM-моделей щодо виявлення додаткових ознак;
- 4) провести порівняльний аналіз застосування LLM-моделей, а саме, окрім використаного в роботі gpt-o4, таким моделей, як gpt-o3, deepseek-v3, llama-3 та інші.

ПЕРЕЛІК ПОСИЛАНЬ

1. Binance Square What is a cryptocurrency listing - <https://www.binance.com/en/square/post/15330973467146> (дата звернення: 28.05.2025).
2. Hedera What is a cryptocurrency exchange - <https://hedera.com/learning/wallets-exchanges/what-is-a-cryptocurrency-exchange> (дата звернення: 28.05.2025).
3. Natalie Luu Demystifying Crypto Exchange Listings: Strategies for a Strong Listing - <https://medium.com/@natalieluu/demystifying-crypto-exchange-listings-strategies-for-a-strong-listing-90c80063fd78> (дата звернення: 28.05.2025).
4. Daniel Nakamoto Secondary Listings Impact on Coin Prices - <https://medium.com/squilla/how-secondary-listings-affect-coin-prices-4e4bd1b7e104> (дата звернення: 28.05.2025).
5. Market Maker Definition: What It Means and How They Make Money - <https://www.investopedia.com/terms/m/marketmaker.asp> (дата звернення: 28.05.2025).
6. Wikipedia Аналіз тональності тексту - https://uk.wikipedia.org/wiki/Аналіз_тональності_тексту (дата звернення: 28.05.2025).
7. Хмарний сервіс програмного забезпечення SAS. - <https://welcome.oda.sas.com> (дата звернення: 28.05.2025).
8. Рейтинг криптобірж від компанії CoinMarketCap. - <https://coinmarketcap.com/rankings/exchanges> (дата звернення: 28.05.2025).
9. Платформа обміну криптоактивів COINBASE. - www.coinbase.com (дата звернення: 28.05.2025).

10. Confusion matrix. –
https://en.wikipedia.org/wiki/Sensitivity_and_specificity (дата звернення: 28.05.2025).
11. 5300 New Tokens Launched Daily in 2024 so Far -
<https://www.coingecko.com/research/publications/how-many-cryptocurrencies-are-there> (дата звернення: 28.05.2025).
12. Що таке лістинг криптовалют? Binance -
<https://www.binance.com/uk-UA/blog/all/що-такє-лістинг-криптовалюти-4184932257952308784> (дата звернення: 28.05.2025).
13. Mexc What is cryptocurrency listing on an exchange? -
<https://blog.mexc.com/what-is-cryptocurrency-listing-guide> (дата звернення: 28.05.2025).
14. Investing Binance co-founder clarifies token listing criteria -
<https://www.investing.com/news/cryptocurrency-news/binance-cofounder-clarifies-token-listing-criteria-93CH-3859905> (дата звернення: 28.05.2025).
15. Платформа обміну криптоактивів Окх. - <https://www.okx.com/>
(дата звернення: 28.05.2025).
16. Chainlink What is Layer 2 -
<https://chain.link/education-hub/what-is-layer-2> (дата звернення: 28.05.2025).
17. CoinDesk What are DeFi Tokens -
<https://www.coindesk.com/learn/what-are-defi-tokens> (дата звернення: 28.05.2025).
18. MoonPay What are utility tokens and how do they work -
<https://www.moonpay.com/learn/cryptocurrency/what-are-utility-tokens> (дата звернення: 28.05.2025).
19. Investopedia Stablecoins: Definition, How They Work, and Types -
<https://www.investopedia.com/terms/s/stablecoin.asp> (дата звернення: 28.05.2025).

20. CoinMarketCap What is a Whitepaper - <https://coinmarketcap.com/academy/glossary/whitepaper> (дата звернення: 28.05.2025).
21. Binance How to Get Your Coin Listed on Binance - <https://www.binance.com/en/support/faq/detail/053e4bdc48364343b863d1833618d8ba> (дата звернення: 28.05.2025).
22. WhiteBit Що таке токеноміка проекту - <https://blog.whitebit.com/uk/tokenomics-to-buy-or-not-to-buy> (дата звернення: 28.05.2025).
23. Coinbase What is MiCa - <https://www.coinbase.com/ru/learn/your-crypto/what-is-mica-markets-in-crypto-as-sets-regulation> (дата звернення: 28.05.2025).
24. CoinMarketCap - <https://coinmarketcap.com/> (дата звернення: 28.05.2025).
25. Itwiki One-hot Encoding - <https://itwiki.dev/data-science/ml-reference/ml-glossary/one-hot-encoding> (дата звернення: 28.05.2025).
26. DEBUT INFOTECH Top 10 Best Crypto Exchanges for USD in 2025 - <https://www.debutinfotech.com/blog/best-crypto-exchanges-for-usd> (дата звернення: 28.05.2025).
27. Dexscreener Графік ціни криптоактиву TST - <https://dexscreener.com/bsc/0x16969fa79651bae11736f2f6576a86fe2726b42b> (дата звернення: 28.05.2025).
28. Бідюк П.І. Прикладна статистика (навчальний посібник) / П.І. Бідюк, О.М. Терентьєв, Т.І. Просянкіна-Жарова. – Вінниця : ПП “ТД Едельвейс і К”, 2013. – 288 с. – ISBN 978-966-2462-21-0
29. Бідюк, П. І., Романенко В.Д, Тимошук О.Л. Аналіз часових рядів : навчальний посібник. – Київ : НТУУ «КПІ», 2011 – 230 с.

30. IBM What is logistic regression - <https://www.ibm.com/think/topics/logistic-regression> (дата звернення: 28.05.2025)
31. Cramer 2002, pp. 3–5. Cramer, J. S. (2002). The origins of logistic regression (PDF) (Technical report). Vol. 119. Tinbergen Institute. pp. 167–178. doi:10.2139/ssrn.360300. (дата звернення: 28.05.2025)
32. Wikipedia Матриця невизначеності - https://uk.wikipedia.org/wiki/Матриця_невизначеності (дата звернення: 28.05.2025)
33. ScienceDirect Misclassification Rate - <https://www.sciencedirect.com/topics/computer-science/misclassification-rate> (дата звернення: 28.05.2025)
34. Wikipedia Precision and recall - https://en.wikipedia.org/wiki/Precision_and_recall (дата звернення: 28.05.2025)
35. Deepchecks What is F-score - <https://www.deepchecks.com/glossary/f-score> (дата звернення: 28.05.2025)
36. Wikipedia ROC-крива - <https://uk.wikipedia.org/wiki/ROC-крива> (дата звернення: 28.05.2025)

ДОДАТОК А ЛІСТИНГ КОДУ

Лістинг коду програми для побудови математичної моделі логістичної регресії.

```
import pandas as pd

import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,
precision_score, recall_score, f1_score,
accuracy_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

# Завантаження даних
df = pd.read_csv("crypto_listings.csv")

# Попередній аналіз
print("Перші 5 рядків:")
print(df.head())
print("\nПропуски по колонках:")
print(df.isnull().sum())

# Заповнення пропущених значень медіаною
columns_to_fill = ['week_price_change',
'month_price_change', 'week_volume_change',
'month_volume_change', 'rank',
'N_of_listings',
'month_delta_listings',
'mean_volume']

for col in columns_to_fill:
    median = df[col].median()
    df[col] = df[col].fillna(median)

# Приведення до цілих чисел тих колонок, де це логічно
df['rank'] = df['rank'].round().astype(int)
df['N_of_listings'] = df['N_of_listings'].astype(int)
df['month_delta_listings'] =
df['month_delta_listings'].astype(int)
```

```

# Формування X та y
X = df[columns_to_fill]
y = df['class']

# Масштабування
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Поділ на train/test
X_train, X_test, y_train, y_test =
train_test_split(X_scaled, y, test_size=0.3,
random_state=42)

# Побудова логістичної регресії
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Прогноз
y_pred = model.predict(X_test)

# Матриця плутанини
tn, fp, fn, tp = confusion_matrix(y_test,
y_pred).ravel()

print("\nМатриця плутанини:")
print(f"True Positives (TP): {tp}")
print(f"True Negatives (TN): {tn}")
print(f"False Positives (FP): {fp}")
print(f"False Negatives (FN): {fn}")

# Метрики
recall = recall_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
misc = 1 - accuracy

print("\nМетрики класифікації:")
print(f"Recall: {recall:.4f}")
print(f"Precision: {precision:.4f}")
print(f"F1 Score: {f1:.4f}")
print(f"Accuracy: {accuracy:.4f}")
print(f"MISC (1-Accuracy): {misc:.4f}")

# Важливість ознак (коефіцієнти логістичної регресії)

```

```
feature_importance = pd.DataFrame({
    'Feature': columns_to_fill,
    'Coefficient': model.coef_[0]
}).sort_values(by='Coefficient', key=abs,
ascending=False)

print("\nВажливість ознак:")
print(feature_importance)

# Візуалізація ваг ознак
plt.figure(figsize=(10,6))
sns.barplot(x='Coefficient', y='Feature',
data=feature_importance)
plt.title("Важливість ознак у моделі логістичної
перспекції")
plt.tight_layout()
plt.show()
```

ДОДАТОК Б ОПИС ПРИНАЛЕЖНОСТІ ПРОЕКТІВ ДО КАТЕГОРІЇ

Усі криптоактиви були поділені на 5 базових категорій.

1 - Layer 1 Protocol Tokens

2 - Layer 2 / Scaling / Infrastructure Tokens

3 - DeFi Tokens

4 - Utility & Ecosystem Tokens.

5 - Stablecoins & Asset-backed Tokens

Таблиця 5.1 – Категорії криптоактивів

Тікер криптоактиву	Категорія, до якої належить криптоактив	Тікер криптоактиву	Категорія, до якої належить криптоактив
WOO	4	AMPL	5
RACA	4	DFI	3
FLUX	4	ROUTE	1
FXS	4	OSMO	3
HIGH	4	CEEK	4
THG	4	AURY	4
SPELL	3	MOON	4
JOE	3	PHB	1
ACH	4	XYM	1
IMX	2	MAGIC	4
API3	4	ZEE	2
ANC	3	RBN	3
PRQ	4	TIME	4
WOO	4	RPL	3
METIS	2	KOK	4
REVV	4	ORAI	1
SAMO	4	H2O	4
ABBC	4	PNK	4
ASTR	2	GNS	3

Кінець табл. 5.1

Тікер криптоактиву	Категорія, до якої належить криптоактив	Тікер криптоактиву	Категорія, до якої належить криптоактив
KDA	1	KLV	4
OOE	3	SYN	2
BSW	3	LQTY	3
XDC	1	UFO	4
PIT	4	SWEAT	4
CEL	4	UFT	4
VLX	1	GLM	4
GST	4	ELON	4
STEEM	4	LOOM	2
NEXO	4	ALEPH	2
LDO	3	KUNCI	4
DEP	4	RDNT	3
IQ	4	HAI	4
FLOKI	4	PLU	4
BOO	3	BONE	3
SDAO	3	WBTC	3
DOME	4	FLOKI	4
OPUL	3	CRTS	4
ZIG	1	ARG	4
STG	3	PENDLE	3
STC	4	STETH	3
XTM	4	TEL	4
GMX	3	CCD	1

Для визначення приналежності активу до категорії були використані описи активів на інтернет ресурсі CoinMarketCap та chat-gpt. Нижче наведений код для визначення приналежності токена до категорії

```

import json
from openai import OpenAI
API_KEY = ''
openai = OpenAI(api_key=API_KEY)
system_message = """Your task is to determine what
category this crypto token belongs to based on it's
description.
There are 5 different categories.
1 - Layer 1 Protocol Tokens
2 - Layer 2 / Scaling / Infrastructure Tokens
3 - DeFi Tokens
4 - Utility & Ecosystem Tokens
5 - Stablecoins & Asset-backed Tokens
You have to respond with just number."""
categories_definition = {
    1: 'Layer 1 Protocol Tokens',
    2: 'Layer 2 / Scaling / Infrastructure Tokens',
    3: 'DeFi Tokens',
    4: 'Utility & Ecosystem Tokens',
    5: 'Stablecoins & Asset-backed Tokens'
}
def get_category(symbol: str, description: str) ->
int:
    result = openai.chat.completions.create(messages=[
        {
            "role": "system",
            "content": system_message
        },
        {
            "role": "user",
            "content": f'Symbol:
{symbol}\nDescription: {description}'
        }
    ],
        model='gpt-4o-mini')
    return int(result.choices[0].message.content)
descriptions = json.load(open('descriptions.json'))
categories = {}

```

```
for symbol, description in descriptions.items():
    category = get_category(symbol, description)
    categories[symbol] = category
with open('categories.csv', 'w') as outfile:
    for symbol, category in categories.items():
        outfile.write(f"{symbol},{category}\n")
```

ДОДАТОК В ЛІСТИНГ КОДУ ПРОМПТУ ДЛЯ СЕНТИМЕНТ АНАЛІЗУ

Для синтемент аналізу був використаний chat-gpt та опис проектів з CoinMarketCap.

Промпт для chat-gpt виглядає наступним чином:

```
You are a crypto market analyst.
Analyze the sentiment of the following crypto project
description.
```

```
Assign a sentiment score from 0 to 1, where:
```

```
0 = very negative sentiment (skeptical, pessimistic,
uncertain)
```

```
0.5 = neutral or balanced (fact-based, no strong
emotion)
```

```
1 = very positive sentiment (visionary, confident,
optimistic)
```

```
Consider tone, confidence, visionary language,
ecosystem strength, and the presence of innovation or
hype.
```

```
Do not validate the claims – focus only on how
positively or negatively the project is presented in
the description.
```

```
Respond with decimal value only.
```

Нижче наведений повний код використаний для отримання синтементу

```
import json
from decimal import Decimal
from openai import OpenAI
API_KEY = ''
openai = OpenAI(api_key=API_KEY)
system_message = """You are a crypto market analyst.
Analyze the sentiment of the following crypto project
description.
Assign a sentiment score from 0 to 1, where:
```


0 = very negative sentiment (skeptical, pessimistic, uncertain)

0.5 = neutral or balanced (fact-based, no strong emotion)

1 = very positive sentiment (visionary, confident, optimistic)

Consider tone, confidence, visionary language, ecosystem strength, and the presence of innovation or hype.

Do not validate the claims – focus only on how positively or negatively the project is presented in the description.

Respond with decimal value only."

def get_sentiment(symbol: str, description: str) -> Decimal:

```

    result = openai.chat.completions.create(messages=[
        {
            "role": "system",
            "content": system_message
        },
        {
            "role": "user",
            "content": f'Symbol:
{symbol}\nDescription: {description}'
        }
    ],
        model='gpt-4o-mini')
    return Decimal(result.choices[0].message.content)
descriptions = json.load(open('descriptions.json'))
sentiments = {}
for symbol, description in descriptions.items():
    sentiment = get_sentiment(symbol, description)
    sentiments[symbol] = str(sentiment)
with open('sentiments.csv', 'w') as outfile:
    outfile.write(f'symbol,sentiment\n')
    for symbol, sentiment in sentiments.items():
        outfile.write(f"{symbol},{sentiment}\n")

```

ДОДАТОК Г ЛІСТИНГ ПРОГРАМИ ВІЗУАЛІЗАЦІЇ РЕЗУЛЬТАТІВ

Для реалізації візуалізації було обрано мову програмування Python та бібліотеку matplotlib, яка надає зручний та гнучкий візуальний інтерфейс для побудови різноманітних графіків і діаграм. За допомогою функцій matplotlib можна налаштовувати вигляд осей, кольорові палітри, підписи та легенди, що дозволяє наочно представити динаміку змін і порівняти різні набори даних. Наведений листинг демонструє підключення необхідних модулів, обробку результатів моделювання та створення готових до аналізу графіків.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix,
ConfusionMatrixDisplay
df = pd.read_csv('results.csv')
y_true = df['true_label']
y_pred1 = df['pred_model1']
y_pred2 = df['pred_model2']
cm1 = confusion_matrix(y_true, y_pred1, labels=[0, 1])
cm2 = confusion_matrix(y_true, y_pred2, labels=[0, 1])
fig, axes = plt.subplots(1, 2, figsize=(10, 5))
disp1 = ConfusionMatrixDisplay(confusion_matrix=cm1,
display_labels=[0, 1])
disp1.plot(ax=axes[0])
axes[0].set_title('Model 1')
disp2 = ConfusionMatrixDisplay(confusion_matrix=cm2,
display_labels=[0, 1])
disp2.plot(ax=axes[1])
axes[1].set_title('Model 2')
plt.tight_layout()
plt.show()
```

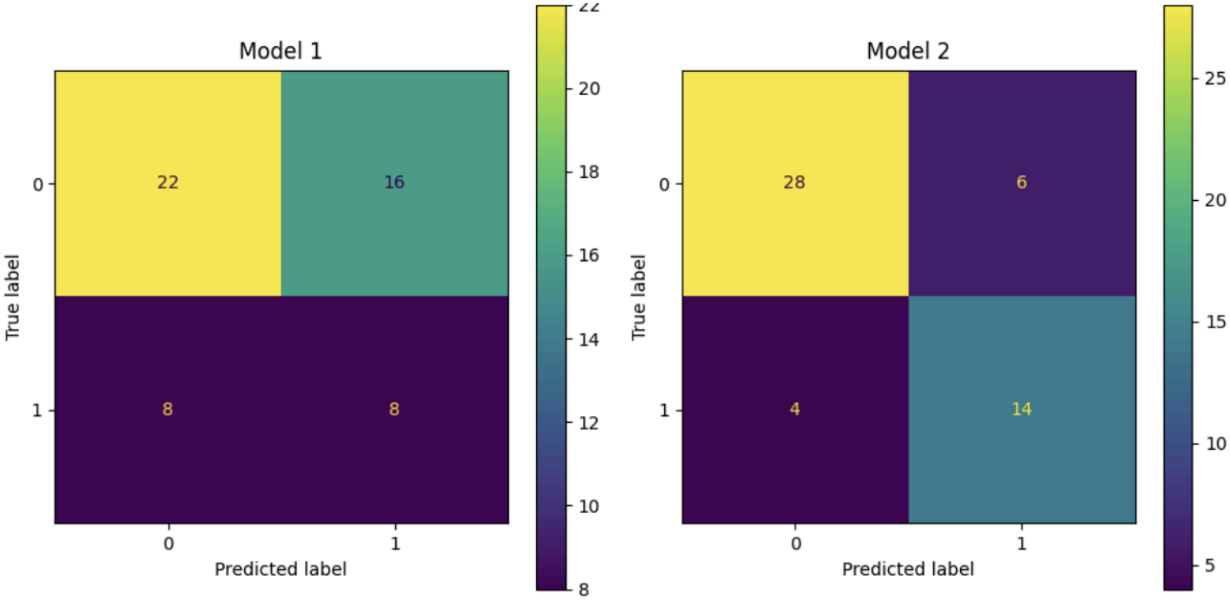


Рисунок 5.1 – Приклад роботи програми візуалізації

ДОДАТОК Д АКТ ВПРОВАДЖЕННЯ

SMART ARBITRAGE
TECHNOLOGIES LIMITED
Company number 12550949
Registered office address
124 City Road, London, City Road,
London, England, EC1V 2NX

**Акт впровадження результатів
бакалаврської роботи Залізняка І. С.**

Останніми роками ринок криптовалют демонструє стрімке зростання як за обсягом капіталізації, так і за кількістю нових цифрових активів. Щодня з'являються десятки нових криптоактивів, які змагаються за увагу інвесторів, користувачів та бірж. Така динаміка створює нові можливості для криптофондів, трейдерів та аналітиків, оскільки раннє виявлення перспективних токенів до їх лістингу на великих біржах, зокрема Binance, може забезпечити значну дохідність при порівняно низьких ризиках.

Було розроблено систему для прогнозування факту лістингу криптовалют на криптобіржі Binance, автором якої є Залізняк І.С. Ця система використовує логістичну регресію, що дозволяє аналізувати велику кількість різноманітних ознак криптопроектів — від технічних параметрів до ончейн-активності та медіа-покриття. Розроблена модель прогнозування побудована на принципах динамічної імовірнісної оцінки, яка враховує постійно оновлювану інформацію з криптовалютних бірж. Модель побудована на принципах імовірнісного класифікування та дозволяє з високою точністю передбачати ймовірність лістингу того чи іншого токена. Вона враховує постійно оновлювані дані з відкритих джерел, що забезпечує її адаптивність до змін ринкового середовища.

Практичне використання створеної моделі прогнозування лістингу криптовалют дозволяє оперативно виявляти перспективні проекти ще до їх офіційного розміщення на біржі Binance. Це відкриває додаткові можливості для формування ранніх інвестиційних позицій з підвищеним потенціалом прибутку. Інтеграція моделі у внутрішню аналітичну систему сприяє автоматизації процесу відбору проектів, підвищує швидкість прийняття рішень та знижує суб'єктивний вплив людського фактору. Отримані результати використовуються як частина комплексного аналізу для побудови стратегій входу в позицію до моменту лістингу, що дозволяє ефективніше розподіляти капітал і зменшувати ризики, пов'язані з хибними очікуваннями ринку.

Цей акт не є документом для фінансових розрахунків

Директор

дата видачі 23 травня 2024 р.



Олексій ГОЛОВІН

ДОДАТОК Е МАТЕРІАЛ ТЕЗ

Національна академія наук України
Інститут телекомунікацій і глобального інформаційного простору
Науковий центр аерокосмічних досліджень Землі Інституту
геологічних наук
Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Національний аерокосмічний університет ім. М.Є. Жуковського
“ХАІ”
Київський національний університет будівництва і архітектури
Національний університет “Києво- Могиллянська академія”
Вінницький національний технічний університет
Національне космічне агентство України
Національний центр “Мала академія наук України”
University of London (Royal Holloway), England
Kielce University of Technology (Politechnika Świętokrzyska), Poland
Opole University of Technology, Opole, Poland
Universita di Roma-2 “Tor Vergata”, Italy

Колективна монографія за матеріалами
23 Міжнародної науково-практичної конференції

**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ЗМІЩЕННЯ ТА
ВІДНОВЛЕННЯ**

(12 – 13 листопада 2024р.)

Київ – 2024

ТОВ «Видавництво «Юстон»

УДК 004:502.17-047,64](082.064)
Д58

*Рекомендовано до видання Вченою Радою Інституту телекомунікацій
і глобального інформаційного простору НАН України,
протокол №10 от 30.10.2024р.*

Д58 Математичне моделювання та інформаційно-комунікаційні технології для зміцнення та відновлення // Колективна монографія за матеріалами XXIII Міжнародної науково-практичної конференції (Київ, 12-13 листопада 2024 р.) / За заг. ред. С.О. Довгого. – К.: ТОВ «Видавництво «Юстон», 2024. – 162 с.

ISBN 978-617-8335-33-5.

До колективної монографії включені сучасні розробки та досягнення вчених за матеріалами XXIII Міжнародної науково-практичної конференції «Математичне моделювання та інформаційно-комунікаційні технології для перемоги та відновлення». Колективна монографія присвячена сучасним проблемним питанням та шляхам їх вирішення, що запропоновані провідними вченими України та представниками вченої спільноти закордонних країн.

Для фахівців науково-дослідних і промислових організацій, керівного складу районних, обласних, центральних органів виконавчої влади, що забезпечують сталий розвиток відповідних одиниць адміністративного устрою держави, викладачів технічних університетів, аспірантів, студентів.

Відповідальність за наведені факти несуть автори. Науково-організаційний комітет та редакційна колегія конференції не завжди поділяє погляди авторів робіт.

УДК 004:502.17-047,64](082.064)

ISBN 978-617-8335-33-5
DOI: 10.37321

© Інститут телекомунікацій
і глобального інформаційного
простору
НАН України

XXIII Міжнародна науково-практична конференція «Математичне моделювання та інформаційно-комунікаційні технології для перемоги та відновлення»

21	Ентропія інформації при виконанні функцій і задач управління об'єктом критичної інфраструктури	61
	<i>Кряжич О.О., Іванов І.А.</i>	
22	Створення підсистеми консолідованої обробки інформації при вогневому ураженні	64
	<i>Нагорний Є.І., Дзюба В.А. Приступа В.В.</i>	
23	Технологій передачі даних: швидкість та стабільність роботи мережі систем зв'язку	65
	<i>Триснюк В.М., Єгоров В.О.</i>	
24	Використання гетерогенних телекомунікаційних мереж спеціального призначення	67
	<i>Довгий С., Козлов О., Копійка О., Яременко А.</i>	
25	Огляд методів оцінювання сценаріїв відновлення в поствоєнних умовах	70
	<i>Оберемко О.О., Кіченко К.В.</i>	
26	Застосування безпілотних засобів в інтересах національної безпеки та оборони країни	72
	<i>Тимчук В.Ю., Мосійчук Д.І.</i>	
27	Інформаційні технології застосування систем машинного аналізу при підготовці маркетингового контенту	74
	<i>Трофимчук О.М., Фадеїчев С.В.</i>	
28	Ідеї Smart City для повоєнного відновлення країни	76
	<i>Данишина С.Ю.</i>	
29	Аналіз оновленого функціоналу ArcGIS Online	78
	<i>Руденко Н.Ю., Данишина С.Ю.</i>	
30	Прогнозування факту лістингу нових криптовалюта на біржі Binance із використанням моделі логістичної регресії	80
	<i>Залізняка І.С., Терентьєв О.М.</i>	
31	Побудова структури інформаційної системи для аналізу та дослідження впливу майнінгу	83
	<i>Дуда В.О., Терентьєв О.М.</i>	

**Прогнозування факту лістингу нових криптовалют на біржі
Binance із використанням моделі логістичної регресії**

Залізняка І. С.¹, Терент'єв О.М.²

*¹ Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»,*

*² Інститут телекомунікацій і глобального інформаційного
простору Національної академії наук України.*

illia09092003@gmail.com, o.terentiev@gmail.com

Вступ

Криптовалюти набули великої популярності за останні роки. З моменту створення першої у світі криптовалюти - біткойн пройшло вже більше 15 років. За цей час сфера криптовалют значно зросла та

розвилася. З'явилася велика кількість додатків для інвестування та торгівлі криптовалютою. Наприклад, додаток Binance є найбільшою у світі крипто біржою за обсягом торгів [1]. Також з'явилася велика кількість криптовалют, тільки за останні півтора роки, на біржі Binance з'явилося щонайменше 100 нових крипто токенів [2]. Більшість з цих токенів впали в ціні після створення та виходу на біржі. Інвестора, які зацікавилися новими токенами в більшості випадків втратили кошти.

Саме тому аналіз можливості лістингу нових криптоактивів на біржі Binance та наступна зміна курсу криптовалют, може допомогти інвесторам приймати більш продумані та зважені рішення. Лістинг – це процес додавання нової монети у список активів для торгівлі.

Опис експерименту та формалізація задачі.

Було накопичено 84 кейсів, фактів лістингу нових криптовалют на біржі Binance, що відбулися у період з 2021-11-16 по 2023-08-17. З 84 кейсів у 32 випадках відбувся факт лістингу відповідного криптоактиву на біржі Binase, а у 52 – не відбувся [2].

Сформований вхідний набір даних містить 8 регресорів та 1 цільову змінну. Список відповідних змінних:

1) `week_price_change` – зміна ціни криптоактиву протягом одного тижня у відсотках.

2) `month_price_change` - зміна ціни криптоактиву протягом одного місяця у відсотках.

3) `week_volume_change` – зміна обсягу обігу криптоактива у відсотках, за останній тиждень.

4) `month_volume_change` – зміна обсягу обігу криптоактива у відсотках, за останній місяць.

5) `rank` – рейтинг криптоактива за капіталізацією, до моменту лістингу на біржі Binance.

6) `N_of_listings` – загальна кількість лістингів на інших біржах, за 2 дні до лістингу на біржі Binaance.

7) `month_delta_listings` – приріст за кількістю бірж, на яких відбувся факт лістингу криптоактиву, доданих за останній місяць.

8) `mean_volume` – середньодобовий обсяг обігу криптоактиву, за останній місяць.

9) `class` – Цільова змінна, що приймає наступні значення: 1 – відбувся лістинг криптоактиву на біржі Binance; 0 – лістинг на біржі Binance не відбувся.

Результати

На основі побудованих ознак-регресорів було побудовано модель логістичної регресії, для прогнозування цільової змінної (`class`). В табл. 1 наведені значення оцінок параметрів побудованої моделі.

Побудована модель логістичної регресії на навчальному наборі даних видає значення помилкової класифікації у 24%, тобто у 76 випадках зі 100 модель вірно повідомляє користувачу, чи відбудеться факт лістингу крипто-активу протягом наступних двох діб.

Таблиця 1. Значення оцінок параметрів моделі логістичної регресії.

Назва параметру моделі	Оцінка параметра моделі	p-значення статистики хі-квадрат
Intercept	-0,8199	0,5739
week_price_change	-0,0232	0,6189
month_price_change	-0,0428	0,0489
week_volume_change	0,00741	0,2196
month_volume_change	0,000396	0,8325
rank	-0,00081	0,4934
N_of_listings	0,0106	0,9837
mean_volume	9,35E-08	0,1455
month_delta_listings	0,5976	0,6664

Список літератури

1. Рейтинг криптобірж від компанії CoinMarketCap. – <https://coinmarketcap.com/rankings/exchanges> (дата звернення: 29.10.2024).
2. Сайт крипто-біржи Binance. – <https://www.binance.com> (дата звернення: 29.10.2024).