


Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Магістерська кваліфікаційна робота на тему:
**«Інформаційна технологія аналізу та передбачення рейтингу ICO
проектів»**


Виконав: студент 2 курсу, групи 2ІСТ–22м
спеціальності 126 – «Інформаційні системи
та технології»

 Дмитро ВАСИЛЮК

Керівник: к.т.н., доц. каф. САІТ

 Олексій КОЗАЧКО
« 01 » 12 2023 р.

Опонент: к.т.н., доц. каф. КН

 Володимир ОЗЕРАНСЬКИЙ
« 05 » 12 2023 р.

Допущено до захисту

Завідувач кафедри САІТ


 д.т.н., проф. Віталій МОКІН

« 01 » 12 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

« 04 » 09 2023 р.

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Василюку Дмитру Олеговичу**

1. Тема роботи: “Інформаційна технологія аналізу та передбачення рейтингу ICO проєктів”,

керівник роботи: Олексій КОЗАЧКО, к.т.н., доц. каф. САІТ,
затверджені наказом ВНТУ « 18 » 09 2023 року № 247

2. Строк подання студентом роботи « 30 » 11 2023 року

3. Вихідні дані до роботи: Kaggle Dataset “ICO project”

<https://www.kaggle.com/datasets/dmytrovasyliuk/ico-data>

4. Зміст текстової частини:

- 1) Загальна характеристика та аналіз ICO;
- 2) Вибір інформаційних технологій рейтингування ICO проєктів;
- 3) Розроблення інформаційної технології аналізу та передбачення ICO проєктів;
- 4) Економічна частина;

5. Перелік ілюстративного матеріалу:

- 1) Кількість ICO у відповідних роках;
- 2) Загальний обсяг ICO в відповідних роках;
- 3) Зовнішній вигляд датасету;
- 4) Кореляція даних зібраних в датасеті;
- 5) Матриця помилок для оптимальної моделі для навчальної вибірки;
- 6) Матриця помилок для оптимальної моделі для тестової вибірки;
- 7) Графік важливості ознак для оптимальної моделі для навчальної вибірки;
- 8) Графік важливості ознак для оптимальної моделі для тестової вибірки;

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Наталія БУРЕННІКОВА, д. е. н., проф. каф. ЕПВМ	20.10.23	08.11.23

7. Дата видачі завдання « 04 » 09 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Загальна характеристика та аналіз ІСО	01.09	20.09	Вик
2	Вибір інформаційних технологій рейтингування ІСО проєктів	20.09	01.10	Вик
3	Розроблення інформаційної технології аналізу та передбачення ІСО проєктів	01.10	20.10	Вик
4	Економічна частина	20.10	05.11	Вик
5	Оформлення матеріалів до захисту МКР	05.11	30.11	Вик

Студент

Дмитро ВАСИЛЮК

Керівник роботи

Олексій КОЗАЧКО

АНОТАЦІЯ

УДК 004.09

Васильюк Д. О. Інформаційна технологія аналізу та передбачення рейтингу ІСО проєктів. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо–професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. 103 с.

Укр. мовою. Бібліогр.: 21 назв; рис.: 42; табл.: 11.

В магістерській кваліфікаційній роботі проведено аналіз предметної області, обґрунтовано доцільність та актуальність даного дослідження. Проаналізовано фактори, які впливають на рейтингування ІСО проєктів.

Досліджено існуючі методи передбачення та рейтингування ІСО проєктів та розроблено інформаційну технологію передбачення та рейтингування ІСО проєктів за допомогою методів машинного навчання.

Ілюстративна частина складається з 8 плакатів.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів.

Ключові слова: ІСО, методи машинного навчання, класифікація, рейтинг проєктів.

ABSTRACT

Information technology for analyzing and predicting the ICO rating of projects. Master's qualification work in the specialty 126 - Information Systems and Technologies, Educational and Professional Program - Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2023. 103 c.

In Ukrainian. Bibliography: 21 titles; Figures: 42; tables: 11.

The master's qualification work analyzes the subject area, substantiates the feasibility and relevance of this study. The factors that influence the rating of ICO projects are analyzed.

The existing methods of predicting and rating ICO projects are investigated and an information technology for predicting and rating ICO projects using machine learning methods is developed.

The illustrative part consists of 6 posters.

The economic section considers the feasibility of developing and implementing an information technology for analyzing and predicting the rating of ICO projects.

Keywords: ICO, machine learning methods, classification, project rating.

ЗМІСТ

ВСТУП.....	4
1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ.....	6
1.1 Аналіз предметної області.....	6
1.2 Переваги та обмеження ІСО.....	15
1.3 Дослідження факторів, що впливають на рейтинг ІСО проєктів.....	17
1.4 Огляд існуючих технологій передбачення рейтингу ІСО	22
1.5 Сучасні технології рейтингування.....	27
1.6 Висновки	28
2 ВИБІР ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ РЕЙТИНГУВАННЯ ІСО ПРОЄКТІВ ДЛЯ РОЗВ’ЯЗАННЯ ЗАДАЧІ.....	29
2.1 Огляд початкових даних.....	29
2.2 Метод опорних векторів.....	43
2.3 Метод k-найближчих сусідів (k-NN).....	51
2.4 Модель "дерево рішень".....	54
2.5 Висновки.....	61
3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РЕЙТИНГУ ІСО ПРОЄКТІВ.....	63
3.1 Розроблення інформаційної технології передбачення рейтингу ІСО проєктів.....	63
3.2 Передбачення рейтингу ІСО проєктів методом опорних векторів.....	64
3.3 Передбачення рейтингу ІСО проєктів методом k-nearest neighbor.....	69
3.4 Передбачення рейтингу ІСО проєктів за допомогою моделі "дерево рішень".....	71
3.5 Висновки.....	75
4 ЕКОНОМІЧНА ЧАСТИНА.....	76
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	76

4.2 Розрахунок узагальненого коефіцієнта якості розробки.....	78
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	79
4.3.1 Витрати на оплату праці.....	79
4.3.2 Відрахування на соціальні заходи.....	82
4.3.3 Сировина та матеріали.....	82
4.3.4 Розрахунок витрат на комплектуючі.....	84
4.3.5 Спецустаткування для наукових (експериментальних) робіт.....	85
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт.....	86
4.3.7 Амортизація обладнання, програмних засобів та приміщень.....	87
4.3.8 Паливо та енергія для науково-виробничих цілей.....	88
4.3.9 Службові відрядження.....	89
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	90
4.3.11 Інші витрати.....	90
4.3.12 Накладні (загальновиробничі) витрати.....	90
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	91
4.5 Висновки.....	96
ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	100
Додаток А (обов'язковий). Технічне завдання.....	103
Додаток Б (обов'язковий). Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень.....	106
Додаток В (довідковий). Лістинг коду.....	107
Додаток Г (обов'язковий). Ілюстративна частина	110

ВСТУП

Актуальність теми. ICO (Initial Coin Offering) став ключовим феноменом в світі криптовалют та блокчейн технологій, відкривши нові можливості для фінансування та інновацій. Його важливість та актуальність полягають в кількох аспектах, які впливають на фінансові та технологічні сфери.

Перш за все, ICO відкриває шлях до фінансування новаторських проєктів. Це дає змогу стартапам збирати кошти для реалізації ідей, які можуть змінити сферу технологій чи послуг. Коли традиційні методи фінансування можуть бути обмеженими, ICO дозволяє звертатися до глобального кола інвесторів.

Друге, що робить ICO актуальним, – це його потенціал для децентралізації. Шляхом видання токенів на блокчейні, ICO надає можливість інвесторам стати частиною проєкту, забезпечуючи прозорість та відкритість у фінансових відносинах.

Третя важливість ICO полягає в сприянні інноваціям. Багато ICO стимулюють розробку нових технологій, які можуть вплинути на різні сфери життя – від фінансів до охорони здоров'я та ланцюжка постачання.

Нарешті, актуальність ICO виявляється в контексті постійного розвитку ринку криптовалют. Цей метод фінансування постійно трансформується, з'являються нові підходи та моделі, такі як STO (Security Token Offerings) чи IEO (Initial Exchange Offerings), що вказує на постійну актуальність цього ринку.

Мета і завдання роботи. Метою дослідження є підвищення точності передбачення рейтингу ICO проєктів за рахунок використання методів машинного навчання.

Для досягнення поставленої мети потрібно розв'язати такі задачі:

- аналіз ICO проєктів;
- створення та обробка датасету, огляд основних ознак;
- створення інформаційної технології для аналізу та передбачення рейтингу ICO проєктів.

Об'єктом дослідження: процес розроблення інформаційних технологій аналізу та передбачення рейтингу ІСО проєктів.

Предметом дослідження: предметом дослідження є методи інформаційних технологій аналізу та передбачення рейтингу ІСО проєктів.

Методи дослідження. У роботі використано методи розвідувального аналізу для дослідження даних, і технології машинного навчання для тренування та передбачення рейтингу ІСО проєктів.

Новизна одержаних результатів. Подальшого розвитку набули математичні моделі рейтингування ІСО проєктів, які на відміну від існуючих використовують методи машинного навчання, за рахунок яких підвищено точність передбачення.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи доповідались на ЛІІ Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації з публікацією тез ВНТУ (м. Вінниця, 2023рр.).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на ЛІІ Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації з публікацією тез ВНТУ (м. Вінниця, 2023рр.) [1].

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Аналіз предметної області

Первинна пропозиція монет, або ICO (Initial Coin Offering), є механізмом залучення коштів для проєкту шляхом емісії та продажу спеціальних цифрових токенів за допомогою криптовалют. В рамках ICO команда проєкту створює та випускає унікальні цифрові активи (токени), які потім продаються інвесторам чи зацікавленим особам. Цей процес можна порівняти з традиційним збором коштів від підтримувачів проєкту, але з використанням криптовалют [2].

Учасники ICO отримують придбані токени, які можуть мати різноманітні застосування у майбутньому, наприклад, доступ до продуктів чи послуг, що розробляються командою проєкту. Зібрані в рамках ICO кошти використовуються для фінансування подальшого розвитку та впровадження проєкту.

Важливим аспектом ICO є використання технології блокчейн, що дозволяє проводити транзакції з цифровими токенами через Інтернет без посередників. Блокчейн використовує систему розподіленої бази даних, де інформація про транзакції об'єднується в блоки з унікальними ідентифікаторами. Цей метод забезпечує децентралізацію та прозорість транзакцій у системі ICO. Для того, щоб додати блок до системи, вузол (комп'ютер) використовує складний математичний процес. Потім інші вузли перевіряють правильність цього блоку. Якщо він визнається коректним, він додається до цепочки блоків (блокчейн) і стає частиною запису всіх транзакцій.

Участь у цій мережі може бути вигідною, оскільки вузли, які додають блоки, отримують винагороду у вигляді цифрових токенів. Ця технологія дозволяє створювати криптовалюту, яка стає більш надійною зі зростанням кількості учасників, які підтримують та перевіряють транзакції.

Для створення нових токенів, що представляють вартість в криптовалюті, існують два основних способи. Один з них - це масштабування і створення нової

блокчейн-мережі. Цей підхід передбачає створення власного блокчейну з унікальними правилами і протоколами.

Другий спосіб - це випуск токенів на основі існуючої блокчейн-платформи, такої як Bitcoin чи Ethereum. У цьому випадку використовуються технічні стандарти, такі як ERC-20 для Ethereum, щоб визначити, як токени можуть бути створені, передані та можуть управлятися на блокчейні.

– Масштабування нової мережі блокчейну: Це означає створення з нуля нової блокчейн-мережі, спеціально призначеної для вашої криптовалюти або токенів. Це може бути важким завданням, оскільки потрібно розробити новий блокчейн, забезпечити його безпеку і децентралізацію, а також залучити користувачів та майнерів для забезпечення стійкості мережі. Це може зайняти час і ресурси;

– Випуск токенів на основі існуючої блокчейн-платформи: Цей шлях включає в себе створення токенів, які працюють на блокчейні, що вже існує, наприклад, на основі платформи Bitcoin. Однак це може бути складним через обмежені можливості самої платформи і потребу внесення змін до коду. Також, важливо враховувати, що неможливо змінити поведінку вже існуючих користувачів та майнерів.

Обидва способи мають свої проблеми. Коли ви створюєте нову блокчейн-мережу, це складно і потребує багато ресурсів. Також, важко залучити достатньо людей до цієї нової мережі, щоб вона стала ефективною [3].

Якщо ви випускаєте токени на основі існуючої мережі, наприклад, на основі Bitcoin, то зіткнетесь із складнощами кодування цих токенів і включення їх в існуючий блокчейн. Також, необхідно вирішити питання про те, як зберігати і відслідковувати інформацію про ці нові токени серед інших транзакцій Bitcoin.

У 2015 році Віталік Бутерін презентував криптовалюту Ethereum та платформу для створення Ethereum, яка базується на концепції децентралізованих смарт-контрактів. Смарт-контракти – це протоколи, які можуть виконуватися самою технологією розподіленого реєстру. Це дозволяє повністю самореалізовувати пункти контракту, без перевірки або виконання контракту

будь-якою стороною. Найчастіше для залучення капіталу використовується ERC–20 смарт-контракт, який створює новий токен криптовалюти і дозволяє передавати криптовалюту від однієї особи до іншої. Ця унікальна функція дозволяє розробникам створювати широкий спектр інноваційних додатків на базі блокчейну Ethereum, що робить його найпопулярнішим блокчейном для ICO [4].

Наразі Ethereum займає 90,2%4 ринку платформ для проведення ICO, однак Ethereum є відносно повільним у проведенні транзакцій. На додаток до транзакцій з примітивними цифровими токенами, така інновація, як смарт-контракт, дозволила також забезпечити можливість легко створювати і автономно розподіляти цифрові токени цінності серед користувачів, роблячи токени доступними для торгівлі. Цей процес створення токенів і розподілу їх серед користувачів в обмін на примітивний цифровий токен мережі (криптовалюту) називається процесом первинної пропозиції монет (ICO) і може розглядатися як новий канал розподілу активів. У широкому сенсі слова, будь-яка криптовалюта, окрім біткоіна, може вважатися ICO, оскільки це фінансовий інструмент, який є відправною точкою для кожної "молодої" криптовалюти [5].

У процесі ICO відбувається пропозиція (продаж токенів) оригінальних монет (токенів) для майбутніх власників у вигляді криптовалюти або криптоактивних активів на основі блокчейну. ICO може передбачати випуск трьох різних типів монет:

- токени корисності, які пропонують майбутній доступ до послуг компанії;
- токени безпеки, які діють подібно до цінних паперів і забезпечують частку майбутнього прибутку компанії;
- криптовалюти або платіжні токени, які діють як засіб обміну для інвесторів.

Однак усі три види монет можуть продаватися після первинної пропозиції на нерегульованих платформах і, таким чином, мають характеристики, притаманні акціям/облігаціям, які можна передавати. ICO можна трактувати як форму колективної підтримки інноваційних технологічних проєктів, один з видів краудфандингу (краудселінгу) у вигляді залучення нових учасників. При цьому

інвестори (покупці токенів) не стають власниками акцій компанії, а державне регулювання ринку криптовалют відсутнє. Для інвесторів основною мотивацією для вкладення коштів у новий проєкт є надія на збільшення вартості токенів після запуску за рахунок успішного функціонування валюти. Основною інформаційною формою, що представляє ICO, є White paper, при цьому не існує жодних стандартів чи регламенту щодо підготовки цього документу [6].

Як правило, технічна записка повинна містити наступні розділи:

- Опис бізнес-ідеї або проблеми: У цьому розділі ви пояснюєте, яку проблему ваш проєкт вирішує або яку бізнес-ідею він розвиває. Важливо надати чіткий контекст і розуміння, чому ваш проєкт є цікавим і важливим;

- Запропоноване рішення та опис проєкту/продукту: Тут ви розкриваєте вашу конкретну пропозицію і як ваш продукт чи проєкт цю проблему вирішує. Вкажіть основні функції і переваги вашого рішення;

- Опис механізму реалізації токена: Цей розділ пояснює, як саме ваш токен взаємодіє з вашим продуктом чи сервісом. Розгляньте його економічні аспекти і технічну реалізацію;

- Команда проєкту: Представлення вашої команди, її навиків і досвіду. Інвестори зазвичай дуже уважно ставляться до складу команди, оскільки це важливий фактор успіху проєкту;

- Параметри і терміни випуску токенів, плани на майбутнє: Тут ви розглядаєте деталі ICO, такі як ціна токенів, обсяг випуску, тривалість ICO і плани використання зібраних коштів;

Щодо процедури випуску ICO, ви також правильно описали два основні етапи:

- Pre-ICO: Це попередній продаж токенів за зниженою ціною. Його ціль – залучити ранніх інвесторів і зібрати початковий капітал для подальшого розвитку проєкту;

- ICO: Це безпосередній етап продажу токенів великій кількості інвесторів. Інвестори купують токени з надією на їхнє зростання вартості в майбутньому.

Перший продаж токенів був проведений Mastercoin у липні 2013 року. Ethereum залучив кошти за допомогою продажу токенів у 2014 році, зібравши 3

700 BTC за перші 12 годин, що на той час дорівнювало приблизно \$2,3 млн. Блокчейн використовується як технологічна база для ICO. Протягом двох останніх років загальний обсяг запозичених коштів різко збільшився. У квітні 2017 року відбувся великий стрибок, витрати на ICO за 3 місяці зросли з \$290 000 до \$1,05 млрд, а до листопада 2017 року збільшилися в 4 рази. Однак з кінця минулого року розмір ринку ICO падає, для порівняння зростання ринку в червні 2017 року становило 93%, а в листопаді – лише 23%. Якщо говорити про географію проєктів ICO, то найбільше ICO було проведено в США (\$1,031 млрд), Китаї (\$452 млн), включаючи Гонконг) яка за рівнем регулювання та підтримки блокчейн-індустрії випереджають інші країни. Найуспішніші блокчейн-проєкти реалізуються у сфері розвитку блокчейн-інфраструктури, на другому місці – фінанси, на третьому – соціальні медіа, контент і реклама, а на четвертому – індустрія ігор і віртуальної реальності [7].

Згідно з аналітикою ICObench, 2018 рік був сповнений як позитивних, так і негативних змін у всьому криптовалютному світі (рис 1.1). У 2018 році відбулося надзвичайне ICO відкритої мережі Telegram. Воно порушило всі правила ринку ICO і збрало 1,7 мільярда доларів. Загальна кількість ICO, що завершилися у 2018 році, зросла у 3,5 рази порівняно з попереднім роком. Весь ринок ICO перейшов на нові стандарти обміну досвідом – збільшилася кількість радників та експертних рейтингів на ICO. Проєкти частіше встановлюють жорсткі та м'які ліміти, що свідчить про зростання ролі таргетингу. Загалом, 2018 рік завершився з меншою кількістю залучених коштів порівняно з початком року. Загальна кількість залучених коштів зросла лише на 15% порівняно з 2017 роком [8].

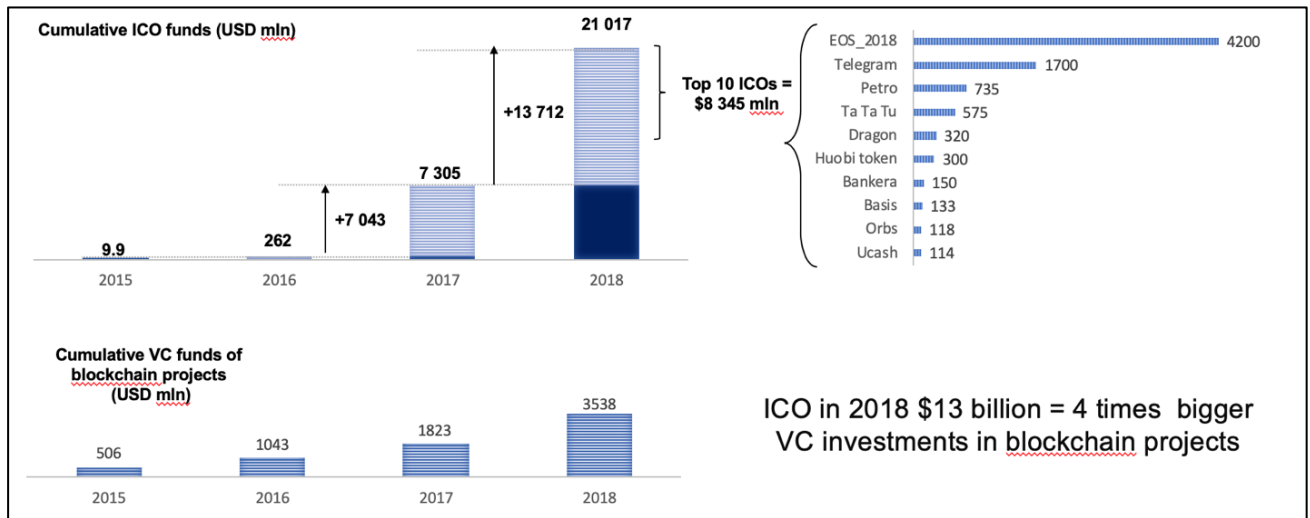


Рисунок 1.1 – Кількість ICO у відповідних роках

Якщо говорити про 2017 рік, то було 718 завершених ICO, і лише 43 з них стали прибутковими через 1 рік. Сукупна ринкова капіталізація ICO 2017 року на 40% менша, ніж загальна кількість коштів, які вони залучили разом. У 2018 році середня кількість коштів, залучених одним ICO, зменшилася порівняно з 2017 роком (рис 1.2). Але все ж таки були проекти, які зібрали вражаючу кількість коштів. Однак ці ICO мають вкрай негативний ROI або поки що не котируються на біржах [9].

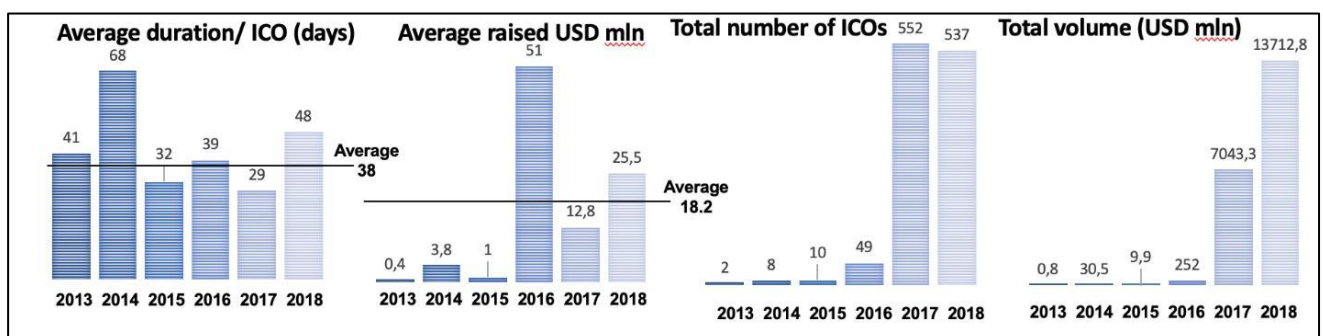


Рисунок 1.2 – Загальний обсяг ICO

Завдяки цій статистиці США залишаються провідною країною для проведення ICO, що підкріплюється чіткими та твердими регуляторними вимогами (наприклад, KYC). У Європі столицею ICO є Швейцарія, але у 2018 році Великобританія вирвалася вперед за обсягом та кількістю угод. В Азії головним

центром ICO є Сінгапур, за яким слідує Гонконг. Кайманові острови та Британські Віргінські острови посідають одне з перших місць за обсягом ICO, оскільки саме вони приймали у себе Unicorn ICO EOS (4,1 млрд доларів США) та Telegram (1,7 млрд доларів США). Інші країни, не відомі своїми великими фінансовими ринками, також з'являються серед країн-лідерів за обсягами ICO (наприклад, Естонія, Литва, Ізраїль). У 2017 році багато запланованих ICO не відбулися або не були публічно задокументовані чи про них не було повідомлено. Зміни в країнах-лідерах ґрунтуються на змінах у регулюванні [10].

Коли ринок криптовалют був дуже популярним, багато людей інвестували гроші в проекти, що випускали свої власні криптовалютні "жетони" (токени) через процедуру, яку називають "ICO" (Initial Coin Offering). Багато з цих інвесторів робили це, бо сподівалися швидко заробити гроші, але не завжди розуміли, в що саме вони інвестували [11].

У першому періоді ця стратегія працювала, і багато жетонів зростали в ціні. Але потім ринок почав падати, і багато людей втратили гроші. Коли це сталося, інвестори почали судитися і намагалися повернути свої гроші, звинувачуючи проекти у порушенні закону або недобросовісному веденні бізнесу [12].

Зокрема, в 2019 році в США було багато судових позовів проти ICO-проектів, які звинувачувалися у порушенні закону про цінні папери. Такі позови стають все поширенішими, і це створює ризики для криптовалютних проектів. Загалом, ринок криптовалют і ICO дуже став дуже ризикований, і інвестори повинні бути обережними та ретельно досліджувати проекти, в які вони вкладають свої гроші.

Після буму ICO в 2017 році і наступних проблем, таких як масові шахрайські схеми та недостатня регуляція, інвестори стали обережнішими. Вони більше не стільки довіряли проектам, що проводили ICO, і вкладали свої кошти в них [13].

Багато країн почали вводити строгі правила і норми для проведення ICO з метою захисту інвесторів. Це регулювання може відрізнитися від країни до країни і вимагати реєстрації та дотримання визначених правил збору коштів.

Окрім ICO, виникли Security Token Offerings (STO), які вважаються більш регульованими і безпечними для інвесторів. STO видає токени, які представляють цінні папери або права на прибуток, і підпадають під законодавство про цінні папери.

Багато проєктів, які проводили ICO в 2020 році, були більш прозорими та відповідальними в порівнянні з тими, що були на ринку під час буму в 2017 році. Вони намагалися відзначитися доброю репутацією і реальними продуктами. Загальна кількість проведених ICO в 2020 році значно скоротилася через вищезгадані тенденції та пандемію COVID-19, яка також вплинула на інвестиційну активність .

Замість традиційних ICO, більша увага зосереджувалася на галузі DeFi (децентралізовані фінанси), де проєкти видають токени для створення фінансових послуг на блокчейні.

Узагальнюючи, ICO в 2020 році стали менш популярними і більш регульованими, а інвестори стали обережнішими при вкладенні коштів у такі проєкти. DeFi став однією з основних тем у галузі криптовалют і блокчейну [14].

У 2021 році ринок ICO продовжував зазнавати змін. Зацікавленість інвесторів у криптовалютах, особливо в Bitcoin і Ethereum, залишалася високою. Це призводило до збільшення інтересу до криптовалютних проєктів і ICO.

Великий витік грошей у NFT-проєкти (невзаємозамінні токени) призвів до зменшення інтересу до традиційних ICO. NFT-проєкти стали важливим гравцем на ринку криптовалют.

Більше країн усіх континентів розробляли або впроваджували строгі правила і регулювання для ICO. Це відбувалося для захисту інвесторів від можливих шахрайських схем і фінансового обману.

Багато ICO почали випускати токени на блокчейні Binance Smart Chain і інших другорядних мережах, де транзакції були дешевшими та швидше підтверджувалися, порівняно з Ethereum.

Загальна кількість проведених ICO продовжувала зменшуватися порівняно з попередніми роками, оскільки інвестори ставали більш обережними.

Деякі проекти перехопили інтерес інвесторів, проводячи STO замість звичайних ICO, оскільки це може забезпечити більший ступінь захисту інвесторів завдяки статусу цінних паперів .

Велика частина уваги спрямовувалася на DeFi проекти, які надають фінансові послуги на блокчейні, без потреби в традиційних фінансових посередниках.

У 2022 році ринок ICO значно спав: загальна сума коштів, залучених через ICO, впала до близько 10 мільярдів доларів. Це пов'язано з низкою факторів, серед яких регуляторні обмеження на ICO в деяких юрисдикціях, зростання альтернативних методів залучення коштів, таких як розміщення токенів безпеки (STO), а також загальний ведмежий ринок криптовалют.

Очікується, що у 2023 році ринок ICO залишатиметься відносно спокійним, оскільки регулятори продовжуватимуть боротися з шахрайством, а інвестори стануть більш обережними щодо шахрайства. Однак все ж таки є кілька перспективних проектів, на які варто звернути увагу, наприклад, такі:

Ось деякі з факторів, які можуть вплинути на ринок ICO у 2023 році:

- Регуляторне середовище: Регулятори по всьому світу все ще намагаються з'ясувати, як регулювати ICO. Ця невизначеність може ускладнити для проектів залучення коштів через ICO;

- Ціна криптовалют: Ціна криптовалют знижується протягом останніх місяців. Це може зробити інвестиції в ICO менш привабливими для інвесторів;

- Поширення альтернативних методів залучення коштів: Альтернативні методи залучення коштів, такі як розміщення токенів безпеки (STO), стають все більш популярними. Це може призвести до зменшення кількості запущених ICO.

Незважаючи на ці виклики, все ж є кілька причин для оптимізму щодо ринку ICO у 2023 році. Технологія блокчейн все ще перебуває на ранніх стадіях розвитку, і є багато перспективних проектів, які все ще перебувають у стадії

розробки. У міру того, як технологія розвиватиметься, а регуляторне середовище стане більш чітким, ми можемо побачити відродження ринку ICO.

1.2 Переваги та обмеження ICO

Найбільшим недоліком інвестицій в ICO є ризик. Ринок нестабільний, і ви ніколи не знаєте справжніх намірів новоствореної компанії. Першим ризиком можна назвати звичайне шахрайство, коли команда проекту переслідує лише одну мету: зібрати гроші інвесторів. Крім того, оскільки на даний момент не існує законів, які б регулювали проведення криптовалютних краудсейлів з позиції інвестора, не можна виключати, що проєкт може не дожити до стадії появи продукту або розчарувати інвестора своєю реалізацією. Згідно зі статистичними дослідженнями Satis Group⁷, провідної консалтингової компанії у сфері ICO, приблизно 81% ICO є шахрайством, ~ 6% – провальними, ~ 5% – "померли", а ~ 8% продовжують торгуватися на біржі. За оцінкою Wall Street Journal, через шахрайські краудфандингові кампанії власники токенів втратили понад мільярд доларів США. Власне, однією з головних причин такої статистики може бути відсутність контролю власників токенів за своїми інвестиціями, відсутність законопроектів та законів, які б регулювали правове поле у сфері ICO [15].

Іншою серйозною загрозою є хакерські атаки. Дослідження, проведене Ernst&Young (2017), показало, що понад 10% усіх коштів, залучених через ICO, було викрадено кіберзлочинцями. Аналітики вивчили 372 ICO, проведені в період з 2015 по 2017 рік. Щомісячна сума збитків від хакерських атак на ICO становить \$1,5 млн. Більше того, зловмисникам часто вдається отримати доступ до персональних даних інвесторів: від їхніх адрес і номерів телефонів до платіжної інформації. Причиною такої вразливості ICO зазвичай є помилки в коді смарт-контракту, які не були виявлені під час аудиту. Одним з найпоширеніших типів атак є "атака 51%". Вона відбувається, коли атакуюча сторона, в ролі якої може виступати відносно невелика кількість майнерів, володіє "контрольним пакетом" хешрейту, тобто обчислювальними потужностями. В результаті атаки майнери

отримують контроль над всією мережею і можуть створювати власні блоки. Таким чином хакери можуть отримати доступ до коштів і перевести їх третім особам.

Перед запуском ICO команда розробників визначає завдання, для виконання яких необхідно зібрати кошти, і вказує в своєму white paper 2 цифри: мінімальну і максимальну, які називаються SoftCap і Hard Cap. Hard Cap визначає кінцеву мету, верхню межу вкладених коштів, найбільш бажаний результат. Це дуже важливий показник, саме тому, що багато криптовалют мають ліміт на загальну кількість одиниць в обігу. Це, в свою чергу, є одним з найважливіших факторів, що впливають на вартість монети, природно, крім попиту і пропозиції. Soft Cap – мінімально необхідна сума інвестицій для того, щоб команда могла продовжити реалізацію проєкту відповідно до планів. Якщо вона не досягнута в зазначений термін, контракт закривається і автоматично повертає всі залучені кошти вкладникам. При досягненні Hard Cap продаж токенів припиняється, але після подолання Soft Cap інвестори контролюють лише придбані токени і не можуть контролювати вкладені кошти або вилучити частину інвестицій [16].

Іншим недоліком є "Gas war". Основним платіжним інструментом в ICO був Ethereum. Для проведення транзакцій в мережі Ethereum необхідно платити комісію майнерам, щоб вони підтверджували операції і заносили їх в нову одиницю. Чим вища комісія (GWEI), тим більш пріоритетною є ваша транзакція для майнерів. Оскільки вартість криптовалюти ETH постійно змінюється, розробники вирішили спростити завдання і ввели додаткову одиницю розрахунку – Gas. Він, у свою чергу, поділяється на дві складові – ліміт (gas limit) і ціну (gasprice). Кожен майнер отримує комісію, яка розраховується в газі і виплачується в ETH. Загальну вартість комісії розраховує відправник, він встановлює ліміт і ціну газу, а потім множить одне значення на інше. Коли учасники ICO використовують продаж токенів Ethereum, щоб конкурувати і отримувати монети нових проєктів, вони встановлюють високі ліміти газу, щоб збільшити швидкість транзакцій і першими придбати токени.

Загалом, основними ризиками є податкові ризики (немає згоди щодо того, чи потрібно сплачувати податки/податок на додану вартість), ризики регулювання та законодавства (не існує єдиної позиції між країнами щодо законодавства та регулювання у сфері ICO, більше того, певне місце проведення ICO навіть за наявності законів, що стосуються юридичної процедури проведення ICO, може зменшити ймовірність успіху. Важливим питанням для інвесторів є те, чи мають документи ICO юридичну основу і чи мають вони якісь права в суді), бізнес-ризиками (та ризиками інвесторів), структурні ризики (напр. обфускація того, як засновники будуть використовувати кошти ICO, незадекларовані зарплати, неопубліковані фінансові документи, власники токенів не можуть голосувати за керівництво емітента токенів тощо), ризики команди (значна кількість некваліфікованої команди (без реального бізнес-досвіду), а якість менеджменту є класичним фактором, що впливає на фінансові показники цінних паперів), ризики токенів (вони можуть бути вкрадені/загублені або гаманець може бути зламаний). Нарешті, велике занепокоєння викликає зв'язок між власниками токенів і холдинговою компанією, і виникає кілька важливих питань, наприклад, що станеться, якщо компанія, яка випустила токени, продається, або чи матимуть власники токенів якісь права при новому керівництві?

1.3 Фактори, що впливають на рейтинг ICO проєктів

Аналіз ICO (Initial Coin Offering) проєктів може включати в себе ряд фундаментальних аспектів, подібних до того, як аналізуються традиційні фінансові активи. Основні ознаки які ми використали:

– роль токена : токени (також відомі як криптотокени або цифрові токени) є цифровими активами, які можуть представляти різні цінності і використовуватися на блокчейн-платформах для різноманітних цілей. Токени можуть бути класифіковані у кілька основних категорій в залежності від їхнього призначення та функціональності. В ICO використовують декілька видів;

Токени корисності (Utility Tokens): ці токени призначені для доступу до певних послуг або функціональності на блокчейн-платформі або в мережі. Вони використовуються для оплати транзакцій або послуг на цій платформі.

Токени управління (Governance tokens) – це вид криптовалютних tokenів, які надають їхнім власникам право голосувати та приймати рішення щодо різних аспектів діяльності в децентралізованих системах чи організаціях. Ці токени дозволяють учасникам голосувати за зміни, які стосуються протоколу, правил, розподілу ресурсів чи інших питань, які впливають на екосистему.

– категорія tokenів : категорія ICO (Initial Coin Offering) вказує на область або сферу, в якій проєкт здійснює збір коштів через ICO. Категорія може визначати сферу діяльності проєкту та його цільову аудиторію;

Є такі типи категорій:

Platform – Використовуються для взаємодії на платформі, яка може бути блокчейн-основаною. Користувачі можуть використовувати токени для оплати за транзакції, що відбуваються на платформі. Це дозволяє забезпечити ефективну та швидку передачу вартості в межах екосистеми. Токени можуть бути використані для запуску та виконання смарт-контрактів на платформі. Смарт-контракти є програмами, які автоматизовано виконують угоди при виконанні певних умов, а використання tokenів для їхнього запуску може бути одним з ключових механізмів стимулювання їхнього використання. Власники tokenів можуть мати можливість голосувати за різні питання, пов'язані з розвитком та управлінням платформою. Це може включати зміни в протоколах, впровадження нових функцій або вирішення інших ключових питань.

Social Network – використовуються у блокчейн-проєктах, спрямованих на розробку або покращення соціальних мереж. Користувачі можуть отримувати токени в обмін на активну участь в соціальній мережі. Ці токени можуть використовуватися для отримання спеціальних привілеїв або доступу до розширених функцій платформи. Токени можуть використовуватися для винагородження користувачів за важливі внески в соціальну мережу, наприклад, за рекомендації друзям, популярність контенту або інші визначені дії.

DeFi – описує проекти, які працюють у сфері фінансів, використовуючи блокчейн і принципи децентралізації.

Gaming – проекти, що об'єднують галузь ігор та технології блокчейн. Токени, пов'язані із галуззю геймінгу в рамках ICO. Токени, які використовуються в межах ігрової екосистеми для здійснення внутрішніх транзакцій, придбання внутрішніх ресурсів, обміну предметами чи отримання спеціальних вигід.

Marketplace – проекти, що створюють децентралізовані ринки або платформи для купівлі та продажу товарів і послуг. Токени, які використовуються для списання товарів або послуг на платформі. Вони можуть бути вартістю для додавання нових продуктів чи послуг до маркетплейсу.

Artificial Intelligence – проекти, що використовують технології штучного інтелекту для розробки нових продуктів або послуг. Токени можуть використовуватися для фінансування розробки нових алгоритмів, моделей машинного навчання або інших інноваційних технологій штучного інтелекту. В деяких випадках, токени можуть використовуватися для обміну даними між розробниками або користувачами для покращення навчання моделей штучного інтелекту.

DEX – проекти, які розробляють децентралізовані криптовалютні обмінники. Токени в цій категорії можуть виконувати різноманітні функції для покращення екосистеми обміну криптовалютами. Токени можуть використовуватися для оплати комісій за торгівлю, виведення коштів, або інших послуг на платформі DEX. Вони можуть також надавати знижки на комісії чи інші привілеї користувачам. Токени можуть представляти участь у ліквідності на DEX, де учасники можуть вкладати свої токени у ліквідні пули та отримувати частку від торгівлі та комісій.

Exchange – проекти, які розробляють криптовалютні обмінники або платформи для торгівлі цифровими активами. Токени в цій категорії можуть виконувати різноманітні функції для поліпшення функціонування платформи та надання додаткових вигід користувачам. Токени можуть використовуватися як

базові або котирувальні валюти для торгівельних пар на біржі. Вони можуть визначати, які активи доступні для торгівлі один з одним.

Data Service – проекти, які пропонують різноманітні послуги, пов'язані з обробкою, збереженням або передачею даних. Токени в цій категорії можуть використовуватися для різноманітних цілей, спрямованих на покращення доступу, безпеки чи обробки інформації. Токени можуть використовуватися для оплати послуг зберігання даних на розподілених або блокчейн-базованих платформах. Користувачі можуть використовувати ці токени для зберігання своїх даних або надання доступу до них. Токени можуть використовуватися для обміну даними між користувачами чи компаніями на децентралізованому ринку даних. Учасники можуть отримувати токени за надання своїх даних або витрачувати їх для отримання доступу до інших даних.

мета : залучення фінансування для розвитку та впровадження проекту на основі блокчейну чи криптовалютної технології. Учасники ICO продають цифрові токени інвесторам або користувачам в обмін на криптовалюту, таку як Bitcoin або Ethereum;

– **ціль збору коштів:** ICO використовують зібрані кошти для розробки та впровадження свого проекту або блокчейн-платформи. Це може включати розробку програмного забезпечення, платформи, інфраструктури, створення споживчого продукту або запуск головної мережі;

– **загальна кількість токенів:** загальна кількість токенів, яка випускається під час Initial Coin Offering (ICO), зазвичай вказується в білій книзі проекту;

– **ціна одного токена:** ціну одного токена, яку інвестори повинні заплатити під час ICO, щоб придбати токени. Ціна ICO може варіюватися від проекту до проекту і може бути визначена різними способами;

– **отримані кошти:** отримані кошти в Initial Coin Offering (ICO) представляють собою суму коштів, яку проект збирає від інвесторів під час ICO;

Ці кошти призначені для фінансування подальшого розвитку проекту та виконання його цілей. Способи та мета використання цих коштів можуть варіюватися в залежності від конкретного проекту і його цілей.

– мінімальний особистий капітал: мінімальний особистий капітал (або мінімальний особистий внесок) в Initial Coin Offering (ICO) – це мінімальна сума коштів, яку інвестор повинен вкласти в проєкт під час ICO, щоб стати учасником та отримати токени. Мінімальний особистий капітал може встановлюватися проєктом і може різнитися від одного ICO до іншого. Сама сума може бути визначена в кількості криптовалюти (наприклад, Bitcoin, Ethereum) чи в фіатних грошах. Зазвичай вона вказується в білій книзі проєкту разом із загальною кількістю tokenів, ціною на токени та іншими важливими параметрами ICO;

Мінімальний особистий капітал може бути встановлений з метою обмеження участі та захисту від спекуляції. Він також може визначати, якого типу інвесторів проєкт планує залучити. Деякі ICO можуть не мати мінімального особистого капіталу, дозволяючи інвесторам вкладати будь-яку суму, а інші можуть вимагати певного мінімуму.

– максимальний особистий капітал: максимальний особистий капітал в Initial Coin Offering (ICO) визначає максимальну суму коштів, яку один інвестор може вкласти в проєкт під час ICO. Ця обмеження може бути встановлене для регулювання участі інвесторів та забезпечення рівних умов для всіх учасників ICO. Максимальний особистий капітал може відрізнятись від одного ICO до іншого і може бути виражений в кількості криптовалюти (наприклад, Bitcoin, Ethereum) або в фіатних грошах. Визначення максимального особистого капіталу може включати додаткові обмеження, такі як обмеження на кількість tokenів, які можуть бути придбані одним інвестором.

– тип токenu : це стандарт , який визначає правила для створення tokenів, що підтримуються на тій чи іншій платформі;

– назви криптовалюти приймає для покупки tokenів ICO: це можуть бути USDT USDC DOT та інші;

– Білий список: Основною інформаційною формою, що представляє ICO, є whitelist, при цьому не існує жодних стандартів чи регламенту щодо підготовки цього документу. У ньому пояснюється яку проблему проєкт вирішує або яку бізнес-ідею він розвиває. Опис механізму як саме ваш token взаємодіє з вашим

продуктом чи сервісом. Представлення команди, її досвіду. Параметри і терміни випуску токенів, плани на майбутнє.

1.4 Огляд існуючих технологій передбачення ICO

Вивчення передбачення ICO включає в себе багато теоретичних та практичних аспектів, і дослідники працюють над розробкою методологій і моделей, які б допомогли краще розуміти цю область. Нижче наведено деякі з теоретичних та практичних аспектів, що вивчаються у дослідженнях передбачення рейтингу ICO [17].

Команда за проектом ICO має велике значення для його успіху. Дослідники аналізують досвід та експертизу членів команди, їх попередні проекти та здатність виконувати обіцяні завдання.

Оцінка того, як проєкт збирається генерувати прибуток та вирішувати конкретну проблему або надавати цінність для користувачів, є важливою частиною аналізу.

Дослідники досліджують ринок, на якому працює проєкт ICO. Вони вивчають конкурентне середовище, потенційних користувачів та відповідність проєкта потребам ринку.

Регуляторні питання і законодавство можуть суттєво вплинути на успішність ICO проєкту. Дослідники вивчають юридичні аспекти та можливі ризики.

Реакція громадськості та інвесторів на проєкт може вплинути на його успішність. Дослідники аналізують публічні відгуки та реакцію спільноти.

Використання методів аналізу даних та статистики для створення моделей передбачення на основі історичних даних про рейтинги ICO проєктів і їхній подальший успіх чи невдачу.

Використання аналізу тексту для оцінки інформації з білетерів, веб-сайтів та соціальних медіа для виявлення ключових слів та фраз, які можуть вказувати на потенційний успіх чи ризики проєкту.

У всіх аналогічних дослідженнях використовуються наукові статті, доступні на різних платформах. Вони включають аналіз безлічі приватних датасетів, які містять інформацію про рейтинги та передбачення ICO проєктів, а також проводять інші дослідження, схожі за тематикою на наше дослідження.

Було вирішено, першим розглянути роботу «The Role of Sentiment Analysis in Forecasting Successful ICO» якого звали Ettore Falde [18].

У роботі він робив аналізатор настроїв, по твітам в Твіттері по ключовим словам які пов'язані з ICO проєктами. Та на основі них робив аналіз.

Першим була Логістична регресія (біноміальне сімейство) була використана в цій моделі, тому що ця модель використовується для задач бінарної класифікації, де відповідь є категоріальною змінною з двома рівнями. Вона моделює ймовірність того, що спостереження належить до вихідної категорії, враховуючи дані.

Отже, результати є досить чіткими і показують: – max f1: поріг = 0,237056; значення = 0,459113

– max точність: поріг = 0.462706; значення = 0.761491;

– max точність: поріг = 0.462706; значення = 0.637676 – max чутливість (пригадування): поріг = 0.005629;

Це можна інтерпретувати як те, що модель є незбалансованою, або є потреба модифікувати модель.

Проте ніяких графіків дослідник не надав та перейшов до інших варіантів оцінювання.

Друга модель показала себе краще порівняно з попередньою моделлю, йому вдалося отримати прийнятні результати (рис 1.3).

– max f1: поріг = 0.314068; значення = 0.605722;

– max точність: поріг = 0.418700; значення = 0.836884;

– max точність: поріг = 0.999187; значення = 1.000000;

– max чутливість (відгук): поріг = 0.006475; значення = 1.000000;

– максимальна специфічність: поріг = 0.999187; значення = 0.999988;

У той час як біноміальні метрики такі: – MSE: 0.1157803 – RMSE: 0.340265 – LogLoss: 0.3501232 – Mean Per-ClassError: 0.2543348 – AUC: 0.8652959 – AUCPR: 0.7275513 – Gini: 0.7305918 – R²: 0.3711521

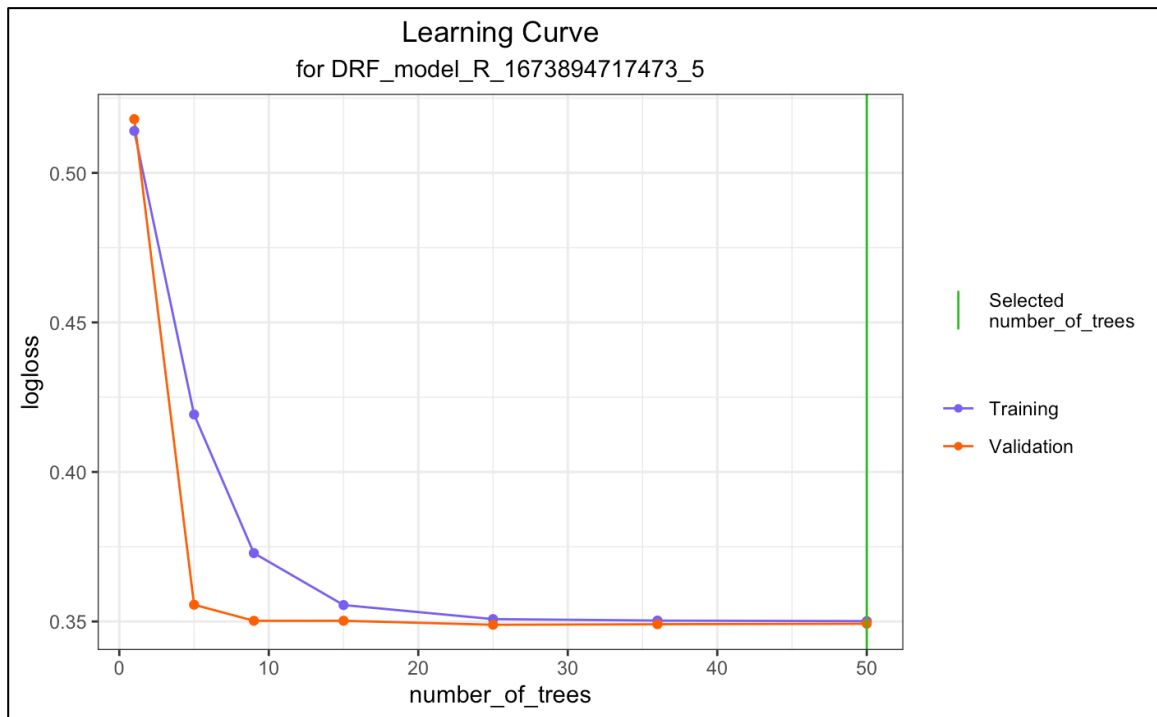


Рисунок 1.3– крива навчання моделі

Наступною роботою було розглянуто ICO «Success and Post-ICO Performance» дослідників Evgeny Lyandres, Berardino Palazzo, Daniel Rabetti [19].

В роботі вони зібрали повний набір даних про первинний випуск монет (ICO) з 19 джерел, включаючи 11 агрегаторів ICO. Проводячи перший систематичний аналіз якості даних про ICO, і використовуємо наш набір даних для вивчення детермінант успіху фінансування ICO, а також успіху фінансування ICO, а також операційних та фінансових показників після ICO. Вони виділили нові для літератури детермінанти успіху ICO та спростували деякі висновки, зроблені в існуючих дослідженнях. Ми також вказали, що участь підприємців у грі є важливою детермінантною операційної діяльності венчурного підприємства після ICO діяльності венчурного підприємства після ICO.

Також вони надали графіки щодо своїх досліджень. Одним з них є дослідження скільки ICO змогли зібрати необхідну суму для інвестицій ICO в часі (рис 1.4).

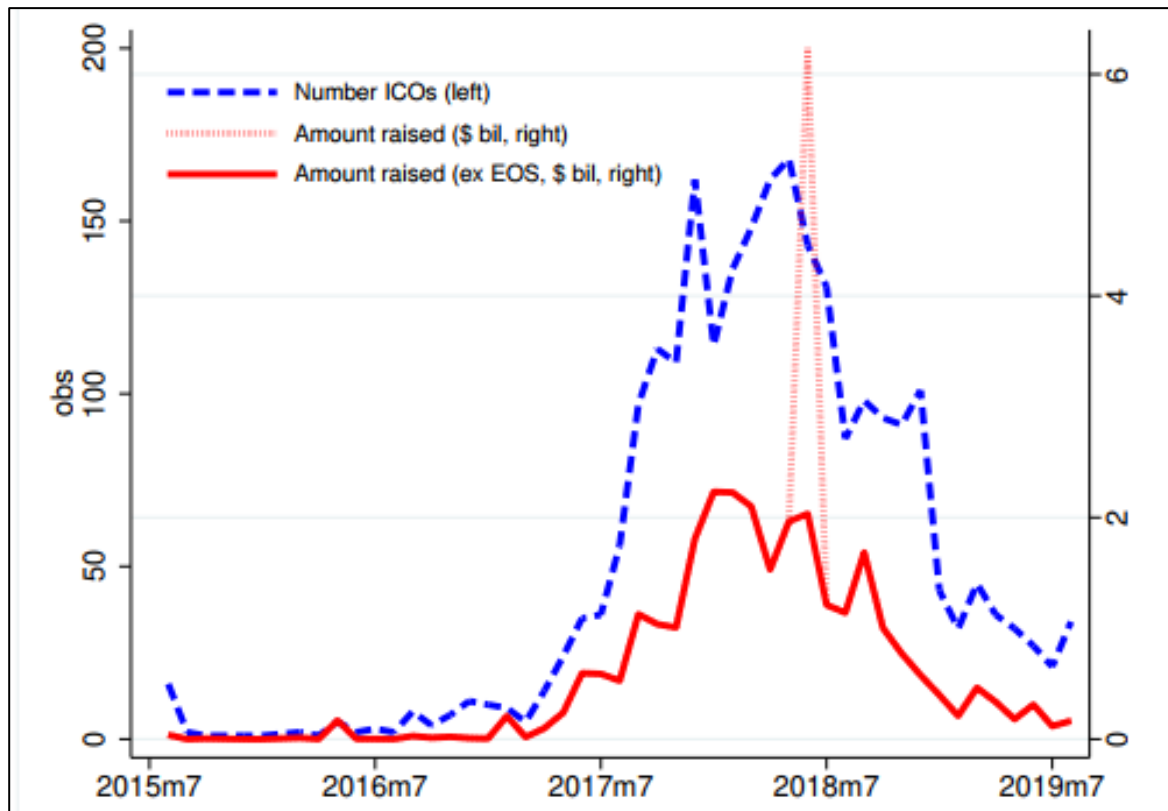


Рисунок 1.4 – Щомісячні значення кількості

На цьому рисунку показано щомісячні значення кількості ICO, які змогли зібрати кошти (пунктирна синя лінія, ліва вісь) та загальну суму, зібрану всіма ICO щомісяця (мільярди доларів, права вісь). Суцільна червона лінія не включає ICO EOS у червні 2018 року, тоді як пунктирна червона лінія включає його. Щомісячні спостереження охоплюють період з серпня 2015 року по серпень 2019 року. Спостереження за серпень 2015 року об'єднують всі ICO до серпня 2015 року.

Також було розглянуто роботу «Success factors of initial coin offerings» [20].

Авторів Maksim Belitski Dmitri Boreiko. На початку роботи вони хотіли давати оцінку ICO проте відмовились від своєї ідеї і вирішили зробити більше аналіз факторів які на це впливають і як вони впливають на оцінювання Рейтингів

ICO. Вони зробили регресійний аналіз кількох основних факторів та підтвердили свої гіпотези які найбільше впливають на оцінку рейтингу ICO.

Вони виявили, що ICO, які отримали фінансування від венчурних фондів або бізнес-ангелів (людей які дають великий капітал на розвиток) до початку кампанії або під час попереднього продажу, залучили більше фінансування в 1,22 рази частіше досягали жорсткого ліміту фінансування (частіше потрапляли до списку крипто валют на [Coinmarketcap.com](https://coinmarketcap.com)). Цікаво, що рейтинг токенів і кілька інвесторів не були пов'язані з фінансуванням з боку венчурних фондів.

Наступна їхня гіпотеза частково підтверджується. На відміну від того, що ми знаємо про важливість білих книг для стимулювання інвесторів, публікація білої книги збільшує кількість інвесторів, в той час як це не впливає на суму залучених коштів та інші показники ефективності ICO.

І остання гіпотеза повністю підтверджується. Засновники ICO, які публікували свої проєкти в репозиторії GitHub, залучали більше коштів мали більше інвесторів, з більшою ймовірністю досягали жорсткого ліміту, були зареєстровані на [Coinmarketcap.com](https://coinmarketcap.com) і мали вищий рейтинг. Іншими факторами, що підвищують рівень успішності ICO, є повторне залучення коштів інвесторів, яке не було пов'язане з підвищенням ефективності ICO, навпаки, воно було негативно пов'язане з досягненням жорсткого ліміту.

Також ICO які проведені в країнах зі сприятливим для ICO законодавством, мали більше шансів досягти більш високих показників, зокрема, досягти ліміту, підвищити рейтинг, а також залучити більше фінансування від більшої кількості інвесторів.

Попередній продаж токенів був позитивно пов'язаний з успішністю та показниками ICO, в той час як частка засновників негативно впливала на загальний обсяг фінансування.

1.5 Сучасні технології рейтингування

Сучасні технології рейтингування широко використовуються у різних галузях і сферах, включаючи, але не обмежуючись інтернет-пошуком, соціальними мережами, електронною комерцією, фінансовим аналізом та іншими галузями.

Ось декілька основних з них:

Алгоритми машинного навчання: Сучасні рейтингові системи використовують алгоритми машинного навчання, такі як випадковий ліс, градієнтний бустінг, нейронні мережі. Ці алгоритми дозволяють автоматично визначати важливість та релевантність об'єктів для рейтингування.

Аналіз тексту та обробка природної мови (NLP) – це сучасні технології, що дозволяють розуміти, інтерпретувати та взаємодіяти з текстовою інформацією. Вони знаходять застосування в багатьох галузях, включаючи інтернет-пошук, соціальні мережі, електронну комерцію, фінансовий аналіз та інші галузі.

Сортування за популярністю: У соціальних мережах і новинах, часто використовується сортування за популярністю, де об'єкти рейтингування впорядковуються за кількістю лайків, коментарів, ретвітів тощо.

Семантичне рейтингування: Системи можуть використовувати семантичні аналізатори для розуміння змісту об'єктів і рейтингування їх на основі смислової близькості до запиту користувача.

Аналітика в реальному часі: Деякі системи рейтингування використовують потокову аналітику для визначення рейтингу в режимі реального часу, забезпечуючи актуальні результати.

Колаборативний і контентний підхід: Рейтингові системи можуть використовувати як колаборативний (заснований на діях користувачів), так і контентний (заснований на аналізі властивостей об'єктів) підходи для рейтингування.

1.6 Висновки

У даному розділі проведено аналіз ICO проєктів, яке є способом залучення коштів для криптовалютних проєктів. Розглянуто ситуацію на ринку ICO, вивчено історію розвитку цього явища від моменту його народження до сучасних часів.

Також був проведений огляд існуючих методів рейтингування проєктів ICO та відзначено інші підходи до оцінки їхньої якості та перспектив.

Методи рейтингування ICO-проєктів можуть варіюватися в залежності від специфіки проєкту та індустрії. Деякі з них можуть базуватися на технічних показниках, фінансових даних та команді розробників. Інші можуть враховувати аспекти соціальної мережевої активності та реакції спільноти на проєкт.

Рейтингові агентства, спеціалізовані сайти та інші джерела інформації можуть надавати рейтинги та оцінки ICO проєктів для допомоги інвесторам та зацікавленим сторонам у прийнятті рішень щодо участі в цих проєктах.

2 ВИБІР ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ РЕЙТИНГУВАННЯ ІСО ПРОЄКТІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Огляд початкових даних

Для вирішення поставленого завдання було створено датасет, фрагмент якого наведено на рисунку 2.1.

name	role_of_category	goal	fundraising_goal	total_tokens	ico_price	received_mone	min_persons	max_person	token_type	available_for	si	accepts_1	accepts_2	accepts_3	accepts_4	accepts_5	whitelist	rating
1 Ozone Metaverse	1 Platform	1430000	280000	2000000000	0.007	1430000	100	500	Flow	21%	USDT	0	0	0	0	0	0	low
2 TypeIT	1 Blockchain S	3900000	2200000	1000000000	0.05	3120000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
3 ECO (Paysenger)	1 Social Networ	4160000	859000	500000000	0.06	4160000	100	2500	BEP-20	26.6%	ETH	USDT	USDC	DOT				1 low
4 Goracle	1 Blockchain Si	3730000	7070000	52200000	0.732	3725000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
5 BIDS SHOP	1 DeFi	17080000	8090000	20320000	0.515	250000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
6 Tectum	1 Blockchain	18700000	8120000	22140000	0.99	4202573	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
7 Ethlas	1 Gaming	2760000	6640000	11320000	0.041	8700000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
8 Archway	1 Blockchain	27000000	6000000	1000000000	0.2	27000000	100	1000	Cosmos N	15%	USDT	USDC						low
9 PolyGame	1 Platform	5230000	100000	500000000	0.075	5225000	100	500	Polygon	6%	USDC	BUSD						low
10 Neon	2 Blockchain Si	45000000	5000000	1000000000	0.1	45000000	100	500	ERC-20	5%	USDT	USDC						high
11 War Legends	1 Gaming	16130000	250000	350000000	0.037	12904000	100	500	Polygon	21%	USDT	0	0	0	0	0	0	low
12 OtterHome	1 DeFi	18100000	8220000	28720000	0.946	14480000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
13 HyperGPT	1 Marketplace	1670000	7990000	26800000	0.089	1665000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
14 Cyber Arena	1 Gaming	2100000	2110000	11800000	0.132	2100000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
15 KAIF	1 Platform	2120000	500000	90000000	0.15	500000	100	500	BEP-20	10%	BUSD	0	0	0	0	0	0	low
16 GPT Guru	1 Artificial Intelli	1450000	6260000	16100000	0.182	1448000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
17 Metaverse Kombai	1 Gaming	18020000	1200000	300000000	0.08	2200000	100	750	BEP-20	11%	USDT	0	0	0	0	0	0	low
18 Cogito Protocol	1 DeFi	11540000	4760000	39600000	0.874	9232000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
19 The Qwan	1 Gaming	18640000	270000	8300000	0.865	500000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
20 Eldarune	1 Gaming	1750000	5720000	27620000	0.288	1750500	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
21 Bware Labs	1 Blockchain PI	1480000	1780000	100000000	0.5	7200000	100	500	ERC-20	25%	USDT	0	0	0	0	0	0	low
22 Orbofi	1 Platform	2600000	880000	27410000	0.905	2600000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
23 Pika Protocol	1 Exchange	10480000	8230000	30120000	0.632	8700000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
24 Wistaverse	1 Platform	930000	2370000	6480000	0.478	930000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
25 CyberConnect	1 Protocol	3550000	3000000	100000000	1.8	30400000	100	500	ERC-20	28%	USDT	0	0	0	0	0	0	high
26 BladeDAO	1 Gaming	2860000	5960000	10000000	0.001	2288000	100	500	Own block	10%	USDT	0	0	0	0	0	0	low
27 Tenet	1 Blockchain	2400000	1560000	1200000000	0.02	2400000	100	500	Own block	10%	USDT	0	0	0	0	0	0	low
28 The Unfettered	1 Gaming	13020000	6250000	29510000	0.112	2500000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low
29 Turbos Finance	1 DEX	3750000	750000	10000000000	0.005	3750000	100	500	Own block	15%	USDT	0	0	0	0	0	0	low
30 WeFi	1 Lending	3210000	8410000	29100000	0.055	1125000	100	500	ERC-20	10%	USDT	0	0	0	0	0	0	low

Рисунок 2.1 – Датасет

В даному датасеті найбільші ролі tokenів Utility Tokens позначені цифрою 1 і кількість таких tokenів 435, та Governance tokens позначені цифрою 2, їх кількість 13 (рис 2.2).

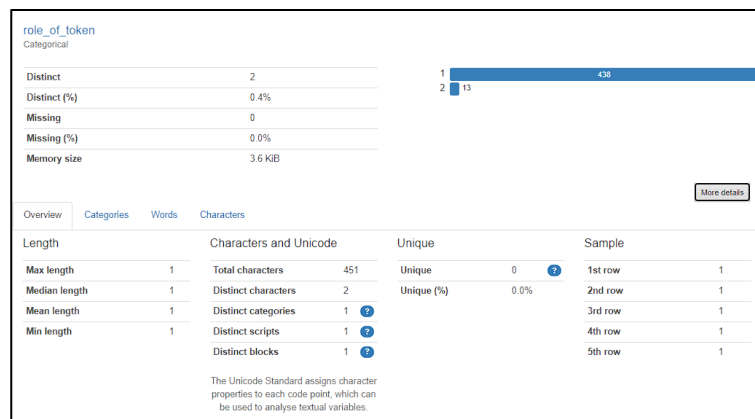


Рисунок 2.2 – Параметр role_of_token

Мета збору коштів: максимальне значення – 720000000, мінімальне значення – 10000, середнє значення –15055263 (рис 2.5).

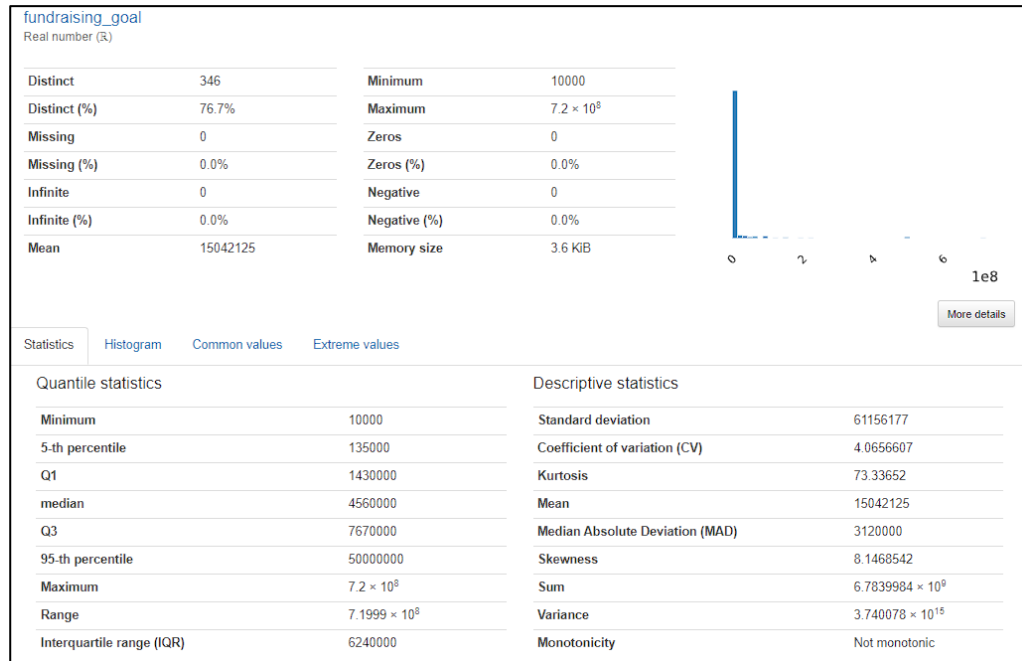


Рисунок 2.5 – Параметр fundraising_goal

Кількість токенів: максимальне значення – 1560000000000, мінімальне значення – 1240000, середнє значення –1066495097 (рис. 2.6).

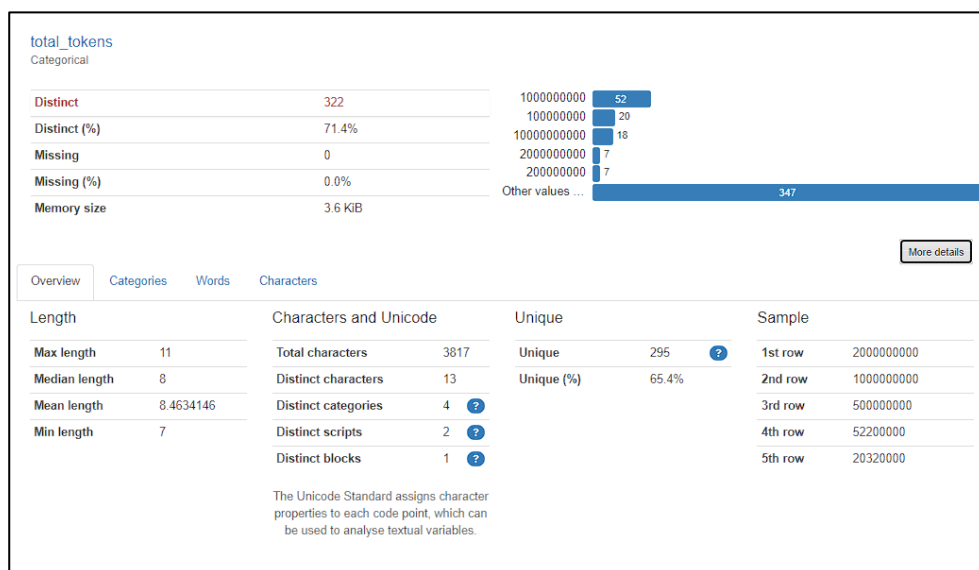


Рисунок 2.6 – Параметр total_tokens

Мінімальна ціна одного токєну: максимальне значення – 15,53, мінімальне значення – 0,001, середнє значення –0,48 (рис. 2.7).

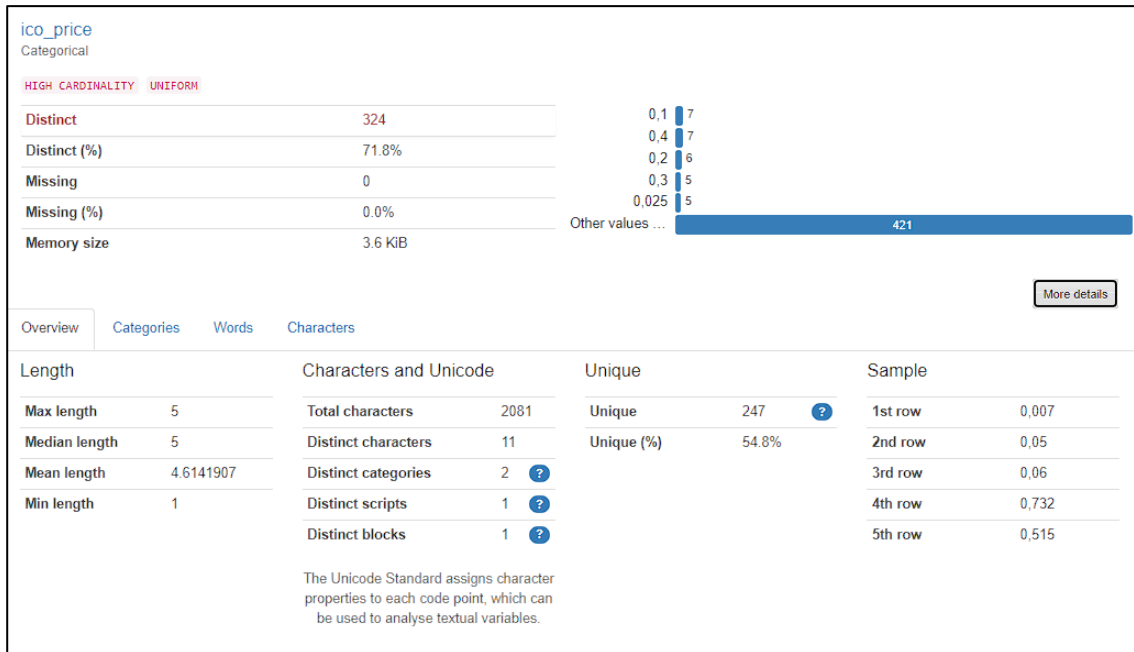


Рисунок 2.7 – Параметр ico_price

Кількість залучених коштів: максимальне значення – 3770000010, мінімальне значення – 0,001, середнє значення – 30000 (рис 2.8).

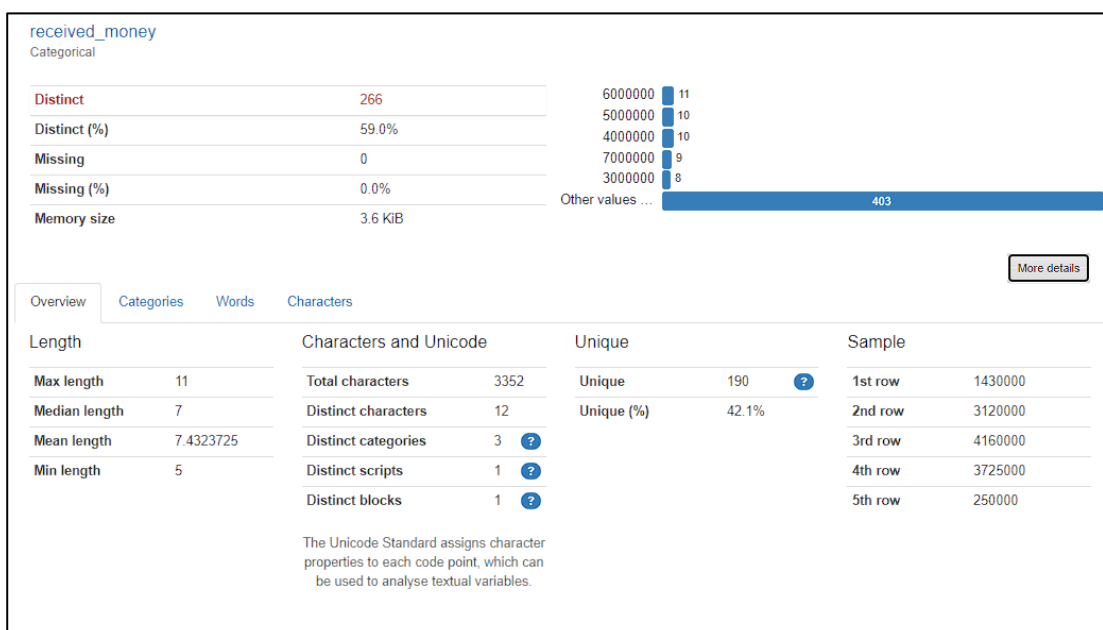


Рисунок 2.8 – Параметр received_money

Мінімальний внесок: максимальне значення – 100000, мінімальне значення –1, середнє значення – 30000 (рис 2.9).

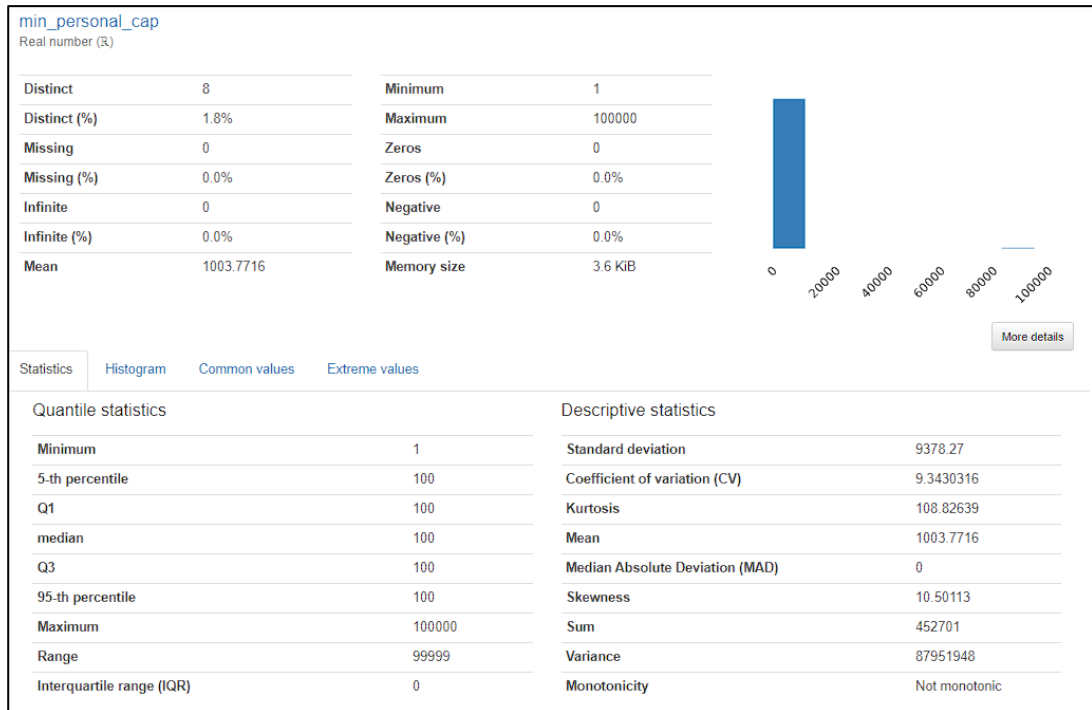


Рисунок 2.9 – Параметр min_personal_cap

Максимальний внесок: максимальне значення – 2000000, мінімальне значення 100, середнє значення – 7534 (рис 2.10).

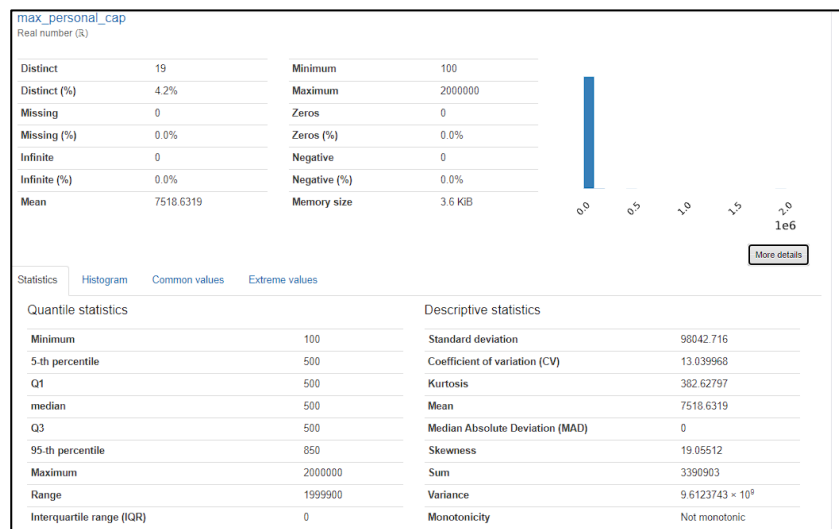


Рисунок 2.10 – Параметр max_personal_cap

Серед типів токенів в нашому датасеті вони відносяться:

ERC-20: найбільш розповсюджений тип, з 369 екземплярами. Ontology – 1, Polygon – 7, ERC777– 1, Solana– 15, Own blockchain– 18, NEAR Blockchain – 1, POLKADOT– 4, Cosmos Network– 1, OEP-4 – 1 (рис 2.11).

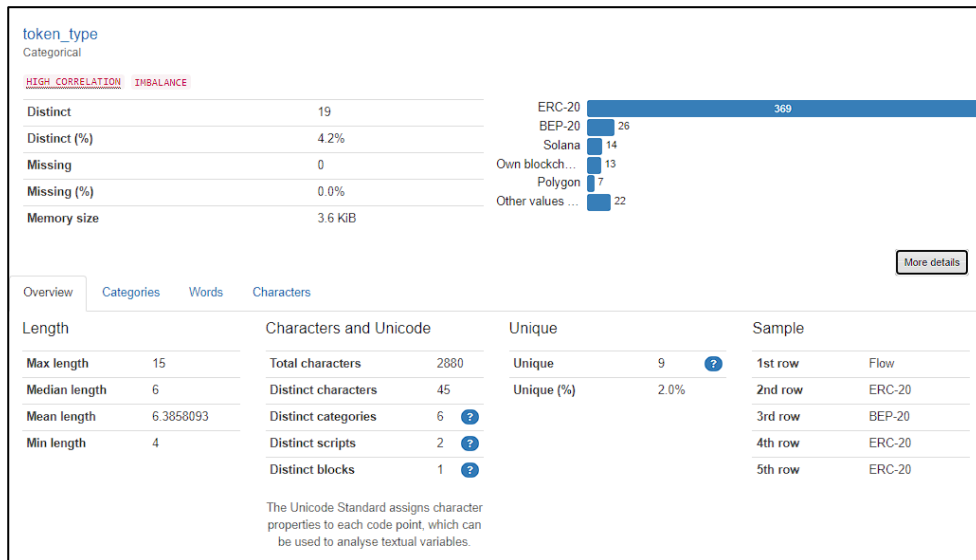


Рисунок 2.11 – Параметр token_type

Відсоток токенів які доступні для продажу: максимальне значення – 40%, мінімальне значення 2%, середнє значення – 11% (рис 2.12).

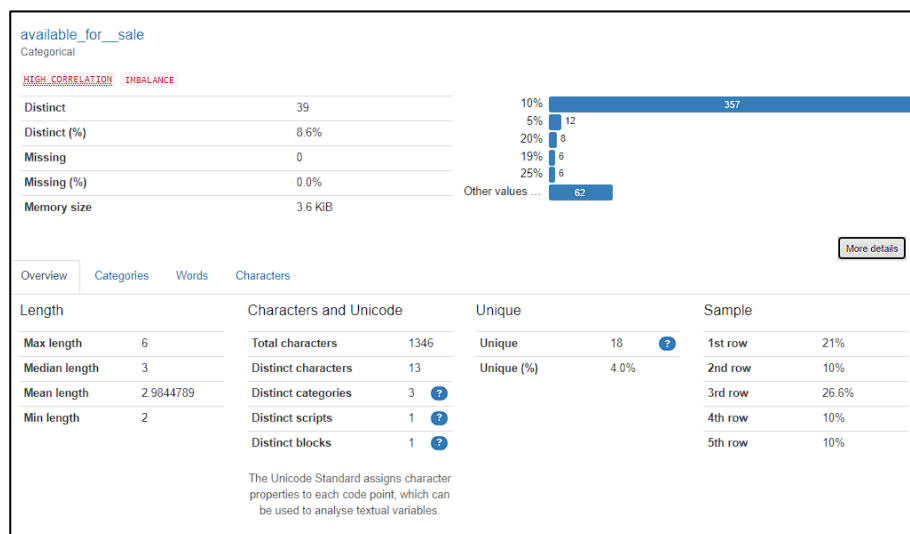


Рисунок 2.12– Параметр available_for__sale

Білий список: має два параметри 1 – даний проєкт має білий список та 0 – проєкт немає білого списку. В даному датасеті проєктів без білого списку 398, та 56 проєктів мають білий список (рис. 2.13).

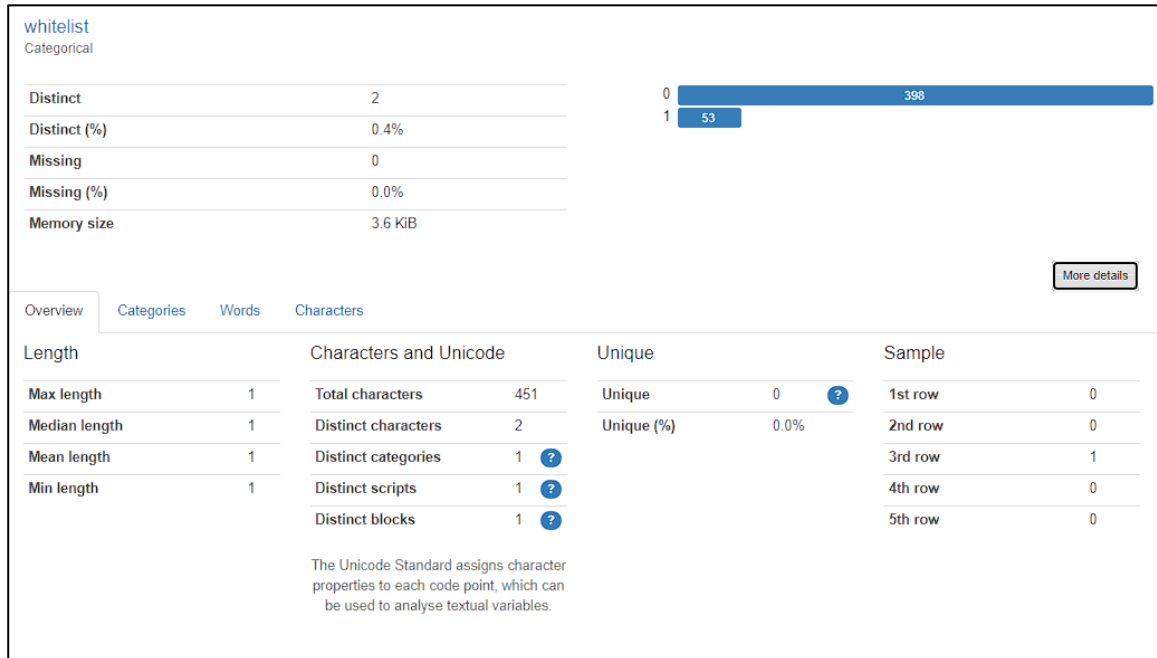


Рисунок 2.13 – Параметр whitelist

Рейтинг: має три параметри низький середній та високий. В даному датасеті проєктів з низьким рейтингом 325, з середнім 96, з високим 30 (рисунок 2.14).

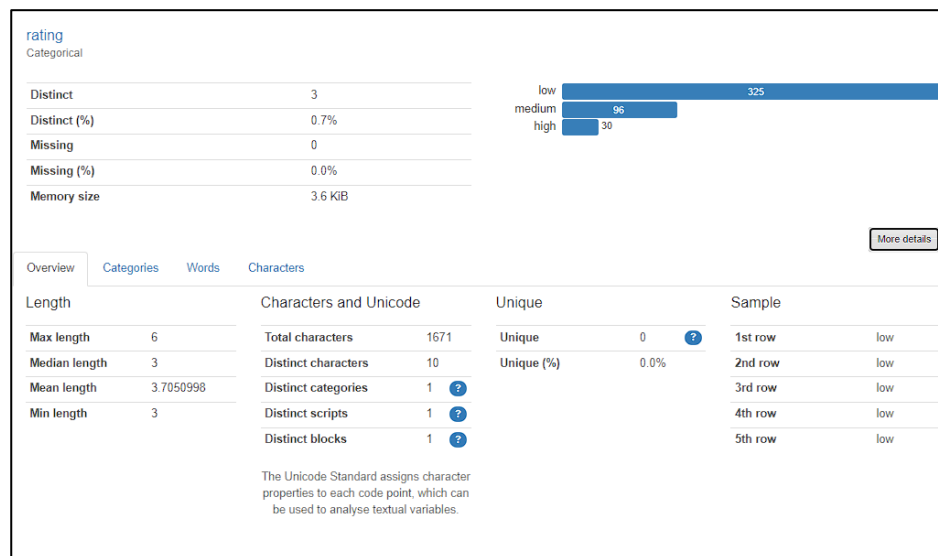


Рисунок 2.14 – Параметр rating

На рисунку 2.15 наведено кореляційну матрицю факторів. З цього рисунку видно, що параметр `goal` мало корелює з іншими параметрами крім параметру `category`.

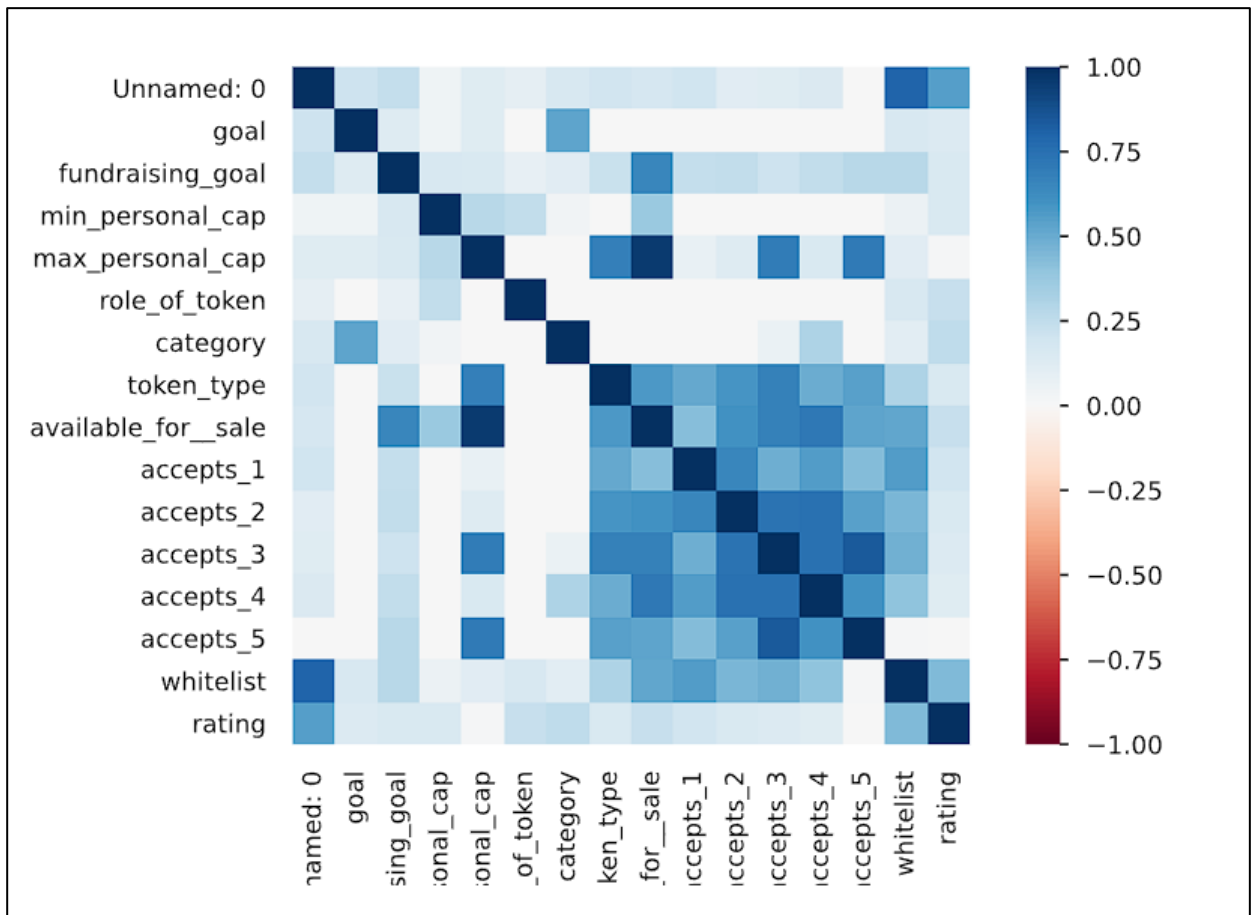


Рисунок 2.15 – Кореляція факторів

Параметр `fundraising_goal` високо корелює з параметрами `available_for_sale`, гірше корелює з параметрами `accept_1`, `accept_2`, `accept_3`, `accept_4`, `accept_5`, з іншими параметрами кореляцію низька. Висока кореляція з параметром `available_for_sale` вказує на те, що чим більша ціль збору коштів, тим більше доступних для продажу токенів.

Параметр `min_personal_cap` має високу кореляцію з параметром `available_for_sale`, з іншими параметрами кореляція низька. Висока кореляція з `available_for_sale` вказуватиме на те, що мінімальна особиста капіталізація

залежить від доступних для продажу токенів. Проєкти можуть встановлювати мінімальну особисту капіталізацію в залежності від розміру збору коштів.

Параметр `max_personal_cap` має високу кореляцію з параметром `avialible_for_sale`, `token_type`, `ассепт_3`, `ассепт_4`, з іншими параметрами кореляція низька. Висока кореляція з параметрами `avialible_for_sale`, `token_type`, `ассепт_3`, `ассепт_4` означає, що максимальна особиста капіталізація залежить від декількох факторів, таких як за яку валюту ми придбати токен та умови прийняття.

Параметр `role_of_token` має низьку кореляцію з всіма параметрами крім `personal_cap` та `raiting`.

Параметр `category` має низьку кореляцію з всіма параметрами крім параметру `goal` та `ассепт_4`.

Параметр `token_type` має високу кореляцію з параметрами `ассепт_1`, `ассепт_2`, `ассепт_3`, `ассепт_4`, `ассепт_5`, `personal_cap`, `avialible_for_sale`, меншу кореляцію з параметром `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Параметр `avialible_for_sale` має високу кореляцію з параметрами `fundraising_goal`, `min_personal_cap`, `max_personal_cap`, `token_type`, `whitelist`, `ассепт_1`, `ассепт_2`, `ассепт_3`, `ассепт_4`, `ассепт_5`, `whitelist`, з іншими параметрами кореляція маленька. Висока кореляція з численними параметрами, включаючи `fundraising_goal`, `min_personal_cap`, `max_personal_cap`, `token_type`, `whitelist`, `ассепт_1`, `ассепт_2`, `ассепт_3`, `ассепт_4`, `ассепт_5`, свідчить про те, що доступність токенів для продажу сильно впливає на інші аспекти проєкту.

Параметр `ассепт_1` має високу кореляцію з параметрами `token_type`, `ассепт_2`, `ассепт_3`, `ассепт_4`, `ассепт_5`, `avialible_for_sale`, `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Параметр `ассепт_2` має високу кореляцію з параметрами `token_type`, `ассепт_1`, `ассепт_3`, `ассепт_4`, `ассепт_5`, `avialible_for_sale`, меншу кореляцію з параметром `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Параметр `ассепт_3` має високу кореляцію з параметрами `token_type`, `ассепт_1`, `ассепт_2`, `ассепт_4`, `ассепт_5`, `max_personal_cap`, `avialible_for_sale`, меншу

кореляцію з параметром `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Параметр `accept_4` має високу кореляцію з параметрами `token_type`, `accept_1`, `accept_2`, `accept_3`, `accept_5`, `avialible_for_sale`, меншу кореляцію з параметром `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Параметр `accept_5` має високу кореляцію з параметрами `token_type`, `accept_1`, `accept_2`, `accept_4`, `accept_5`, `max_personal_cap`, `avialible_for_sale`, з іншими параметрами кореляція або мала або відсутня.

Параметр `whitelist` має високу кореляцію з параметрами `token_type`, `accept_1`, `accept_2`, `accept_3`, `accept_4`, `max_personal_cap`, `avialible_for_sale`, `raiting`, з іншими параметрами кореляція або мала або відсутня.

Параметр `raiting` має високу кореляцію з параметрами `whitelist`, з іншими параметрами кореляція або мала або відсутня.

Визначення рейтингів для проєктів ICO на основі даних у наборі відноситься до завдання класифікації в машинному навчанні. У цьому випадку, модель вчиться класифікувати ICO проєкти за різними параметрами, щоб прогнозувати їхні рейтинги. Це може включати в себе аналіз різноманітних аспектів, таких як купівельні звички, ринкові умови та регуляторне середовище для кожного проєкту.

Задача класифікації дозволяє системі не лише присвоювати рейтинги проєктам відповідно до різних критеріїв, але й ідентифікувати нестандартні випадки або потенційні ризики, які можуть виникнути під час ICO. Враховуючи різноманітні аспекти, модель може надавати глибший аналіз та корисну інформацію для прийняття рішень у галузі ICO. Класифікація може бути корисною при визначенні потенційних ризиків та переваг для кожного ICO проєкту. Наприклад, шляхом аналізу звичок покупця, ринкових умов і регуляторного середовища можна провести класифікацію проєктів за їхньою потенційною прибутковістю та ступенем відповідності потребам ринку.

Така система класифікації дозволяє не тільки ранжувати проєкти відповідно до різних параметрів, але і виділяти нестандартні випадки або потенційні ризики,

які можуть виникнути під час проведення ІСО. Наприклад, виявлення негативних аспектів, таких як обмежена ринкова потреба чи правові ризики, може послужити основою для спеціальної класифікації та подальшого детального аналізу таких проєктів.

Класифікація – це систематичний розподіл об'єктів, явищ, процесів або предметів дослідження за родами, видами, типами на основі істотних ознак для полегшення їх аналізу. Це упорядкована множина об'єктів, які мають спільні класифікаційні ознаки, що вибираються для визначення схожості або відмінності між ними.

При класифікації слід дотримуватися кількох основних правил:

- Використовувати лише одну основу в кожному акті ділення;
- Забезпечити пропорційність ділення, щоб обсяг видових понять відповідав обсягу родового поняття;
- Зробити члени ділення взаємно виключними, уникнувши перехрещення їх обсягів;
- Здійснювати ділення послідовно та логічно.

Допоміжна (штучна) класифікація виникає на підставі зовнішніх ознак та призначена для ефективного організації множини предметів, процесів або явищ з метою швидкого та зручного їх відшукування. Цей вид класифікації визначається конкретною метою, а саме полегшенням пошуку конкретного індивідуального об'єкта серед класифікованих об'єктів. Основу для цієї класифікації становить зовнішня, несуттєва ознака, яка, тим не менше, виявляється корисною під час процесу пошуку.

Природна класифікація формується на основі суттєвих ознак, що визначають внутрішню спільність предметів і явищ. Вона є результатом наукового дослідження і відображає результати вивчення закономірностей об'єктів, що класифікуються. Природна класифікація виступає важливим інструментом у наукових дослідженнях, оскільки вона передбачає і закріплює вивчені закономірності та характеристики класифікованих об'єктів.

Ці два підходи до класифікації можуть сприяти в різних контекстах, де швидкий доступ або вивчення суттєвих особливостей є ключовими аспектами організації інформації.

Класифікація може приймати різні форми в залежності від обраних ознак, їх комбінацій та процедур розподілу понять:

Проста класифікація: Розподіл родового поняття за однією ознакою та лише один раз до розкриття всіх видів. Наприклад, дихотомія, де класифікація здійснюється на основі двох суперечливих понять, кожне з яких відображає "А" і "не А".

Складна класифікація: Використовується для розподілу одного поняття за різними основами та синтезу таких простих ділень в єдине ціле. Ця форма класифікації дозволяє враховувати різноманітні аспекти та характеристики об'єктів під час їх групування.

Класифікація відноситься до стратегії навчання з вчителем, також відомої як контрольоване чи кероване навчання в машинному навчанні. Це штучна генерація знань із досвіду, де система навчається на прикладах і після завершення фази навчання може узагальнювати отримані знання. У контексті класифікації, це означає, що система не тільки вивчає наведені приклади, а й розпізнає певні закономірності в даних для навчання.

Програмні продукти, які використовують класифікацію, застосовуються в різних областях, таких як автоматичне діагностування, виявлення шахрайства з кредитними картками, аналіз ринку цінних паперів, класифікація ланцюжків ДНК, розпізнавання мовлення та тексту, а також автономні системи.

У традиційному машинному навчанні з учителем програміст використовується для навчання системи, розмічаючи дані, подаючи машині конкретні приклади та спостерігаючи за її прогресом. Завдання, які вирішуються через навчання з учителем, включають класифікацію і регресію. У машинному навчанні без учителя використовуються такі типи, як кластеризація, узагальнення та пошук правил.

Алгоритми машинного навчання поділяються на кілька категорій:

Навчання з вчителем (Supervised Learning) представляє собою метод машинного навчання, при якому модель вивчається на основі доступного набору даних, що включає в себе пари вхідних та відповідних вихідних значень. В цьому процесі модель розпізнає зв'язок між вхідними та вихідними даними і може застосовувати цей зв'язок для прогнозування відповідей для нових, раніше невідомих вхідних даних. Використання навчального набору, де кожному вхідному прикладу відповідає правильна відповідь або мітка, дозволяє моделі коригувати свої прогнози та покращувати точність прогнозів.

Навчання без вчителя (Unsupervised Learning) – це стратегія машинного навчання, де алгоритм аналізує неетикетовані або непозначені дані. У цьому випадку система самостійно виявляє закономірності, структури або взаємозв'язки в наборі даних без конкретних вказівок щодо правильних відповідей. Навчання без вчителя охоплює різні методи, такі як кластеризація, зменшення розмірності, виявлення аномалій та інші, що дозволяють системі вирішувати завдання без конкретного навчального супроводу.

Навчання із закріпленням (Reinforcement Learning) - це метод машинного навчання, де алгоритм отримує досвід через взаємодію з навколишнім середовищем. У цьому підході система, яка навчається (агент), взаємодіє з певним оточенням, отримуючи нагороди або штрафи в залежності від своїх дій. Мета агента полягає в максимізації загальної кількості нагород протягом часу, вивчаючи оптимальні стратегії для досягнення цього результату.

Мета класифікаційного процесу полягає в створенні моделі, яка використовує прогнозуючі атрибути як вхідні параметри і надає значення залежного атрибута. Суть класифікації полягає в розподілі множини об'єктів на класи за конкретним критерієм.

Термін "класифікатор" вказує на сутність, яка визначає, до якого з певних класів належить об'єкт на основі вектора ознак.

Для проведення класифікації за допомогою математичних методів необхідно мати формальний опис об'єкта, яким можна оперувати,

використовуючи математичний апарат класифікації. В даному випадку таким описом є база даних, де кожен об'єкт містить інформацію про певні властивості.

Набір вихідних даних розділяється на дві множини: навчальну і тестову.

Навчальна множина використовується для навчання або конструювання моделі, містить вхідні та вихідні значення прикладів. Тестова множина використовується для перевірки працездатності моделі, також містить вхідні та вихідні значення для прикладів.

Процес класифікації складається з двох етапів: конструювання моделі та її використання.

Конструювання моделі:

- Формування опису множини визначених класів, де кожен приклад набору даних призначається конкретному класу;

- Використання навчальної множини для побудови моделі;

Отримана модель представлена класифікаційними правилами, "дерево рішень" або математичною формулою.

Використання моделі:

- Класифікація нових або невідомих значень;

- Оцінка правильності (точності) моделі;

- Порівняння відомих значень з тестового прикладу із результатами використання отриманої моделі;

- Визначення рівня точності, вираженого у відсотках правильно класифікованих прикладів у тестовій множині.

Тестова множина, на якій проводиться тестування побудованої моделі, повинна бути незалежною від навчальної множини. При задовільній точності моделі можливе використання її для класифікації нових прикладів, класифікація яких невідома.

Оцінка точності класифікації, яка визначає рівень помилок, часто виконується через процедуру крос-перевірки. Крос-перевірка, також відома як процедура оцінки класифікації на тестовій множині, інколи називається крос-

перевірочною множиною. У цьому процесі точність класифікації тестової множини порівнюється з точністю класифікації навчальної множини.

Крос-перевірка розглядається успішною, якщо класифікація тестової множини показує подібні результати за точністю, що й класифікація навчальної множини. Процедура включає поділ вибірки на навчальну та тестову множини в певних пропорціях, наприклад, дві третини даних для навчання та одна третина для тестування.

У випадку великих вибірок цей підхід ефективний. Запропоноване розподілення може бути менш ефективним для вибірок з обмеженою кількістю прикладів, тому для таких випадків рекомендується використовувати спеціальні методи, які дозволяють частково перетинати навчальну та тестову вибірки.

Оцінка класифікаційних методів варто проводити, враховуючи такі важливі характеристики: швидкість, робастність, інтерпретованість та надійність.

Швидкість визначає, як швидко створюється та використовується модель, враховуючи часові обмеження.

Робастність вказує на стійкість методу до різних порушень вихідних передумов, зокрема здатність працювати з зашумленими даними та випадками пропущених значень

Інтерпретованість гарантує зручність розуміння моделі аналітиком чи взагалі користувачем.

Властивості класифікаційних правил, такі як розмір "дерево рішень", та компактність класифікаційних правил, також важливі для оцінки.

Надійність методів класифікації передбачає їх здатність працювати ефективно при наявності шумів і викидів в наборі даних.

2.2 Метод опорних векторів

Метод опорних векторів (SVM) – це машинний алгоритм, який навчається на основі прикладів і використовується для класифікації об'єктів. Наприклад, SVM може бути використаний для виявлення аварійного режиму роботи

електромеханічних систем і класифікації його на основі наявних даних і можливих технологічних вимог для режимів роботи. Цей підхід відкриває значні можливості для розробки адаптивних систем автоматичного управління [21].

Лінійний метод опорних векторів складається з такого набору даних:

$$(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n), \quad (2.1)$$

де y_i є або 1, або -1 , і кожен з них вказує клас, до якого належить точка \vec{x}_i

Кожен \vec{x}_i є p -вимірним дійсним вектором. Нам треба знайти «максимально розділову гіперплощину», яка буде відділяти групу точок \vec{x}_i , для яких $y_i = 1$ від групи точок, для яких $y_i = -1$, вона визначається таким чином, щоб відстань між цією гіперплощиною та найближчою точкою \vec{x}_i з кожної з груп була максимальною.

Будь-яку гіперплощину може бути записано як множину точок \vec{x}_i які задовольняють:

$$\vec{w} \cdot \vec{x} - b = 0, \quad (2.2)$$

де \vec{w} є вектором нормалі для цієї гіперплощини. Параметр $\frac{b}{\|\vec{w}\|}$ визначає зсув гіперплощини від початку координат вздовж вектора нормалі \vec{w} .

Якщо тренувальні дані лінійно роздільні, тобто існують два класи даних, які можна ідеально розділити за допомогою гіперплощини, то можна визначити дві паралельні гіперплощини, які розділять ці два класи. $\vec{w} \cdot \vec{x} - b = 1$ (будь-що на або вище цієї межі належить до класу з міткою 1). $\vec{w} \cdot \vec{x} - b = -1$ (будь-що на або нижче цієї межі належить до класу з міткою -1).

Відстань між цими гіперплощинами з геометричної точки зору є $\frac{2}{\|\vec{w}\|}$ тож для максимізації відстані між ними нам треба мінімізувати $\|\vec{w}\|$. Оскільки ми також маємо завадити потраплянню точок даних до розділення, ми додаємо таке обмеження: для кожного i , або $\vec{w} \cdot \vec{x} - b \geq 1$ якщо $y_i = 1$, або або $\vec{w} \cdot \vec{x} - b \leq$

-1 якщо $y_i = -1$. Ці обмеження визначають, що кожна точка даних повинна знаходитися на правильному боці гіперплощини розділення. Можемо об'єднати ці ідеї, щоб отримати задачу оптимізації:

Мінімізувати $\|\vec{w}\|$ за умови $y_i(\vec{w} \cdot \vec{x} - b \geq 1)$ для $i = 1, \dots, n$. \vec{w} та b , які розв'язують дану задачу, визначають наш класифікатор, $\vec{x} \rightarrow \text{sgn}(\vec{w} \cdot \vec{x} - b)$.

Очевидним, але важливим наслідком цього геометричного опису є те, що максимально розділова гіперплощина повністю визначається тими \vec{x} які лежать найближче до неї.

Одна з основних переваг SVM полягає в його здатності ефективно опрацьовувати високо-вимірні дані. SVM виконується особливо добре, коли кількість ознак більша, ніж кількість вибірок, що відомо як “прокляття вимірності”. Це робить SVM ідеальним вибором для завдань, таких як розпізнавання зображень, де вхідні дані часто містять велику кількість пікселів.

Крім того, SVM відмінно справляється зі сценаріями, де навчальні дані обмежені. SVM може давати точні результати навіть зі відносно невеликим набором навчальних даних. Ця характеристика робить SVM особливо цінним у галузях, де збір даних є складним або дорогим.

Якщо навчальний набір містить два класи даних, які можна лінійно розділити, то існує безліч лінійних класифікаторів, які можна використовувати для поділу цих класів. Очевидно, що поверхня, яка проходить через середину смуги, що відділяє два класи, є більш ефективною, ніж поверхня, що знаходиться дуже близько до представників одного чи обох класів.

Різні методи навчання використовують різні підходи до знаходження лінійного роздільника. Наприклад, персептрон дозволяє знайти хоча б один лінійний роздільник, тоді як наївний байєсівський метод шукає оптимальний лінійний роздільник за певним критерієм. Метод опорних векторів, зокрема, визначає поділяючу поверхню, яка максимально віддалена від усіх точок даних.

Зазор між цією поверхнею та найближчою точкою даних називається зазором класифікатора. Важливо враховувати, що в методі опорних векторів

вирішальна функція формується на основі обмеженої підмножини даних, яка визначає положення роздільника. Ці точки називаються опорними векторами, оскільки їх можна розглядати як вектори від початку координат до цих точок у векторному просторі. Зазор і опорні вектори для простої задачі, і інші точки даних не мають впливу на визначення поділяючої поверхні.

Максимізація зазору є перспективною ідеєю, оскільки точки, які знаходяться близько до поділяючої поверхні, створюють значну невизначеність: класифікатор може вибрати будь-яке з двох рішень з ймовірністю 50%. Класифікатор з великим зазором зменшує цю невизначеність, створюючи визначений запас надійності: невеликий вимір помилки чи невелика зміна у вхідних даних не призведе до неправильної класифікації.

Ще однією інтуїтивно зрозумілою обґрунтуванням методу опорних векторів є представлення на рисунку 2.16. Класифікатор SVM конструктивно вимагає широкого зазору навколо поділяючої поверхні. Спроба розмістити широку смугу між класами призводить до того, що діапазон кутів, за яким це можливо, значно скорочується порівняно з гіперплощиною. Це призводить до зменшення обсягу запам'ятовування моделі, і можна очікувати, що здатність моделі до правильного узагальнення на тестових даних збільшиться.

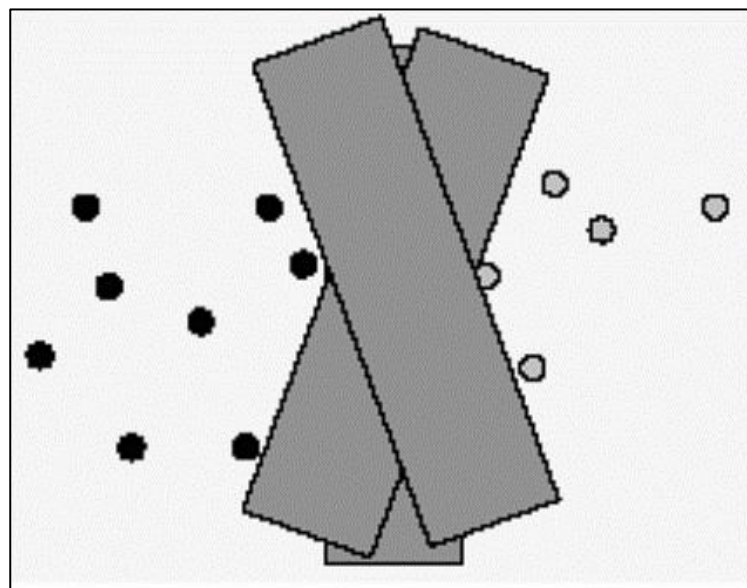


Рисунок 2.16 – Кореляція факторів

Інтуїтивне обґрунтування класифікації із широким зазором полягає в тому, що прагнення до широкого зазору призводить до зменшення обсягу запам'ятовування моделі. Діапазон кутів, за яким можна помістити широку поділяючу смугу між двома класами, є меншим, ніж для поділяючої гіперплощини.

Ще одна важлива особливість це можливість інтерпретації. SVM надає чіткі рішення, що дозволяє користувачам розуміти і тлумачити прогнози моделі. Ця прозорість особливо важлива в галузях, де зрозумілість має важливе значення, таких як охорона здоров'я чи фінанси. У порівнянні з цим, нейронні мережі часто вважаються «чорними скриньками», що ускладнює розуміння мотивів їх прогнозів. Крім того, SVM менш схильний до перенавчання.

Стандартний метод опорних векторів спрямований на класифікацію між двома класами. У випадку класифікації на кілька класів за допомогою методу опорних векторів, традиційно використовується один із методів, які були описані у лекції 7. Наприклад, зазвичай будується $|C|$ класифікаторів, які працюють за принципом "один проти іншому" (іноді це називають "один проти всіх" або OVA). Потім обирається клас, на якому тестовий документ знаходиться найдалі від поділяючої поверхні.

Інша стратегія включає побудову групи класифікаторів, які працюють за принципом "один проти одного", а потім обирається клас, який отримав більшість голосів серед класифікаторів. Навіть якщо ця процедура передбачає створення $|C|(|C|-1)/2$ класифікаторів, час на їхнє навчання на практиці може зменшитися, оскільки навчальний набір для кожного класифікатора є набагато меншим.

Проте, усі ці підходи виглядають не дуже елегантно. Ефективніше було б розробити класифікатор для кількох класів, побудувавши бінарний класифікатор на основі вектора ознак $\Phi(\vec{x}, y)$. Цей вектор складається з пар вхідних ознак і відповідного класу. На етапі тестування класифікатор обирає клас $y = \arg \max \bar{w}^t \Phi(\vec{x}, y)$.

Зазор на етапі навчання визначається різницею між значеннями, що відповідають правильному класу та найближчому неправильному класу. Таким чином, задача квадратичного програмування має наступну умову:

$$\forall i \forall y \neq y_i \bar{w}^t \Phi(\vec{x}, y) - \bar{w}^t \Phi(\vec{x}, y_i) \geq 1 - \varepsilon_i \quad (2.3)$$

Цей загальний метод дозволяє сформулювати задачу багатокласової класифікації для різних лінійних класифікаторів. Це також представляє собою просте узагальнення класифікації, де класи не обов'язково є простим набором незалежних категоріальних міток, а можуть бути структурованими об'єктами з взаємозалежностями. Такі варіанти методу опорних векторів відомі як структурні.

Перенавчання виникає, коли модель стає занадто складною і починає запам'ятовувати навчальні дані замість вивчення загальних закономірностей.

Регуляризуючий параметр SVM допомагає запобігти перенавчанню, контролюючи складність моделі. Ефективність в роботі з високо-вимірними даними, надійність при обмеженому обсязі навчальних даних, можливість інтерпретації та стійкість до перенавчання роблять SVM надійним вибором для багатьох задач машинного навчання.

Формальніше, опорно-векторна машина будує гіперплощину, або набір гіперплощин у просторі високої або нескінченної вимірності, які можна використовувати для класифікації, регресії та інших задач. Інтуїтивно, добре розділення досягається гіперплощиною, яка має найбільшу відстань до найближчих точок тренувальних даних будь-якого з класів (так зване функціональне розділення), оскільки в загальному випадку що більшим є розділення, то нижчою є похибка узагальнення класифікатора.

SVM використовується для завдань класифікації тексту, таких як призначення категорій та виявлення спаму. Він також зазвичай використовується для вирішення проблем розпізнавання зображень, особливо ефективним у розпізнаванні на основі аспектів та класифікації на основі кольорів. SVM також

відіграє важливу роль у багатьох областях розпізнавання рукописних цифр, таких як служби автоматизації поштового зв'язку.

Суть SVM полягає в математичному алгоритмі, який максимізує певну функцію відносно набору даних. Для кращого розуміння принципів роботи SVM важливо розглянути чотири ключові концепції:

Відділяюча гіперплощина (The Separating Hyperplane) – це математичний об'єкт, який визначається таким чином, щоб ідеально відділяти об'єкти двох різних класів на основі їхніх ознак. Уявімо тривимірний простір для кращого уявлення: відділяюча гіперплощина може представляти собою площину, яка ефективно розділяє світлі та темні об'єкти.

Це математичне поняття має широкі застосування в задачах класифікації та машинного навчання. Його використовують для створення оптимального розділу між різними класами даних. Відділяючі гіперплощини дозволяють алгоритмам навчання визначати оптимальний роздільний між класами, що допомагає у розпізнаванні та класифікації об'єктів на підставі їхніх ознак.

Гіперплощина максимальної межі (The Maximum-Margin Hyperplane) – це ключовий концепт у методі опорних векторів (SVM). SVM прагне знайти гіперплощину, яка має максимальний інтервал або межу між собою та найближчими точками кожного класу в навчальній множині (рис 2.17).

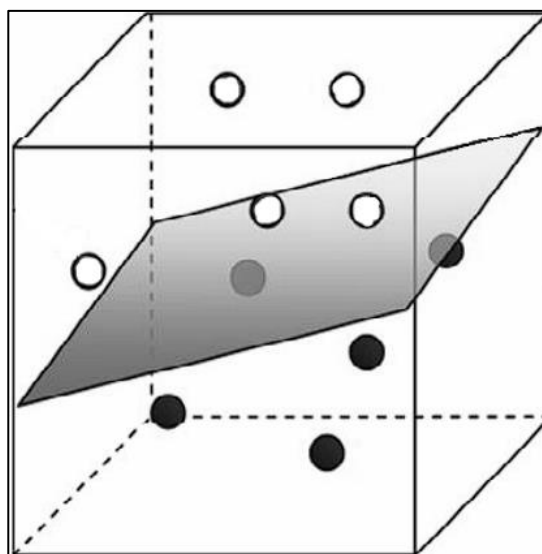


Рисунок 2.17 – приклад відділяючої площини

Максимізація цього інтервалу сприяє створенню оптимального розділу між класами. Гіперплощина максимальної межі дозволяє досягти кращої узагальнюючої здатності моделі, оскільки вона обирає такий розділ, який має найбільшу дистанцію до точок кожного класу. Це допомагає уникнути перенавчання моделі на навчальних даних і підвищує ефективність класифікації на нових, раніше не бачених даних. М'яка межа (The Soft Margin): Іноді даних не можна ідеально розділити лінійною гіперплощиною через шуми або нерівномірний розподіл даних. В таких випадках використовуються м'які межі, де деякі помилки класифікації допускаються.

Функція ядра (The Kernel Function): Функція ядра використовується для перетворення даних в більш високорозмірний простір, де вони можуть бути краще розділені гіперплощинами.

Під час обчислення перетворень для методу опорних векторів (SVM) може виникнути проблема через потребу враховувати багато нових вимірів, кожен з яких може вимагати значних обчислень. Проте існує ефективний підхід, який дозволяє уникнути цих витрат.

Суть магії SVM полягає в тому, що для його функціонування не потрібні самі вектори; достатньо здійснювати точкові добутки між векторами. Замість того, щоб обчислювати нові розмірності, можна використовувати точкові добутки, що дозволяє економити обчислювальні ресурси.

Введення поняття ядер вирішує цю проблему. Ядро є методом, який дозволяє уникнути обчислення нових розмірностей. Коли ядро лінійне, отримуємо лінійний класифікатор. Однак використовуючи нелінійне ядро, можна отримати нелінійний класифікатор, не змінюючи самі дані. Ми просто змінюємо точковий добуток на добуток у відповідному просторі, і SVM легко впорається з цим завданням.

Трюк із ядром фактично не є частиною самого SVM, але є інноваційним підходом, який дозволяє використовувати SVM більш ефективно, зокрема в ситуаціях, коли розглядається нелінійний роздільник.

2.3 Метод k-найближчих сусідів (k-NN)

Метод k-найближчих сусідів (k-NN) є одним з простих та ефективних алгоритмів машинного навчання, використовуваним для класифікації та регресії. Основна ідея полягає в тому, що об'єкти, які подібні за ознаками, розташовані близько один до одного. В методі k-NN, для класифікації нового об'єкта визначається, які класи найближчих сусідів до цього об'єкта (рис. 2.18), і вибирається найчастіший клас серед цих сусідів [22].

Число "k" в методі k-найближчих сусідів вказує на кількість найближчих об'єктів у просторі ознак, які порівнюються з об'єктом, який класифікується. Іншими словами, коли $k = 10$, то кожний об'єкт порівнюється з 10-ма найближчими сусідами.

Під час навчання алгоритм просто запам'ятовує всі вектори ознак і відповідні мітки класів. При роботі з реальними даними, тобто зі спостереженнями, для яких не відомі мітки класів, обчислюється відстань між вектором нового спостереження і попередньо запам'ятованими векторами. Потім вибирається "k" найближчих до нього векторів, і новий об'єкт відноситься до класу, до якого належить більшість з них.

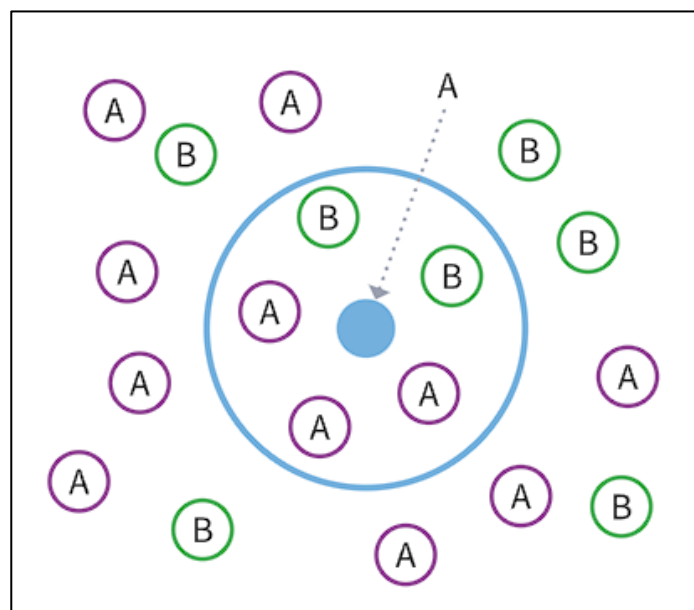


Рисунок 2.18 – Метод k-найближчих сусідів

Основні кроки методу k-NN:

- Визначення відстаней: Для кожного об'єкта в навчальному наборі обчислюються відстані між ним і новим об'єктом, який потрібно класифікувати. Це може виконуватися за допомогою різних метрик відстані, таких як евклідова відстань, Манхеттенська відстань і т. д.;

- Вибір k-найближчих сусідів: Об'єкти, які мають найменші відстані до нового об'єкта, вибираються як "сусіди". Кількість сусідів (k) вибирається користувачем;

- Класифікація: Для класифікації нового об'єкта обчислюється, які класи належать до k-найближчих сусідів.

Нехай є набір даних, що складається з n $X_i (i = 1, \dots, n)$ для кожного з яких задано клас $C_j (j = 1, \dots, n)$. Тоді на його основі може бути сформовано навчальну множину, усі приклади якої являють собою пари X_i, C_j .

Алгоритм KNN можна розділити на дві прості фази: навчання і класифікації. Під час навчання алгоритм просто запам'ятовує вектори ознак спостережень і їхні мітки класів. Також задається параметр алгоритму k , який задає кількість "сусідів", які будуть використовуватися під час класифікації [23].

На фазі класифікації пред'являється новий об'єкт, для якого мітка класу не задана. Для нього визначаються k-найближчих (у сенсі деякої метрики) попередньо класифікованих спостережень. Потім вибирають клас, якому належить більшість із k-найближчих прикладів-сусідів, і до цього ж класу належить об'єкт, що класифікується.

У найпростішому випадку клас нового об'єкта може бути визначений простим вибором класу, що найчастіше зустрічається, серед k прикладів. Однак на практиці це не завжди вдале рішення, наприклад, у випадку, коли частота появи для двох або більше класів виявляється однаковою. Крім цього, розумно припустити, що не всі навчальні мають однаковою значущість для визначення класу. У цьому разі використовують деяку функцію, за допомогою якої визначається клас, звану функцією поєднання (combination function).

У звичайному випадку використовують так зване просте незважене голосування (simple unweighted voting). При цьому передбачається, що всі k прикладів мають однакове право "голосу" незалежно від відстані до класифікованого об'єкта.

Однак, логічно припустити, що чим далі приклад розташований від об'єкта, що класифікується, у просторі ознак, тим нижча його значущість для визначення класу. Тому для поліпшення результатів класифікації вводять зважування прикладів залежно від їхньої віддаленості. У цьому разі використовують зважене голосування (weighted voting).

В основі ідеї зваженого голосування лежить введення "голосу" для класу, залежно від того, наскільки приклади, що належать до нього, віддалені від об'єкта, що класифікується. Такий "голос" подається як сума величин, обернених квадрату відстаней від прикладу j -го класу до об'єкта, що класифікується :

$$Q_j = \sum_{i=1}^{n_j} \frac{1}{D^2(x, a_{ij})}, \quad (2.4)$$

де D – оператор обчислення відстані,

x – вектор ознак об'єкта, що класифікується,

a_{ij} – i прикладу j класу.

Таким чином, "перемагає" той клас, для якого величина Q_j виявиться найбільшою. При цьому також знижується ймовірність того, що класи отримають однакову кількість голосів.

Вибір параметра k є важливим для отримання коректних результатів класифікації. Якщо значення параметра мале, то виникає ефект перенавчання, коли рішення щодо класифікації ухвалюють на основі малої кількості прикладів, і воно має низьку значущість. Це схоже на перенавчання в моделі "дерево рішень", коли в них багато правил, які стосуються невеликої кількості прикладів. Якщо встановити $k = 1$, то алгоритм буде просто присвоювати будь-якому новому спостереженню мітку класу найближчого об'єкта. Крім цього, слід враховувати,

що використання невеликих значень k збільшує вплив шумів на результати класифікації, коли невеликі зміни в даних призводять до великих змін у результатах класифікації. Але при цьому межі класів виявляються більш вираженими (клас під час голосування перемагає з великим рахунком).

Навпаки, якщо значення параметра занадто велике, то в процесі класифікації бере участь багато об'єктів, що належать до різних класів. Така класифікація виявляється занадто грубою і погано відображає локальні особливості набору даних. Таким чином, вибір параметра k є компромісом між точністю та узагальнюючою здатністю моделі.

За великих значень параметра k зменшується зашумленість результатів класифікації, але знижується вираженість меж класів.

У задачах бінарної класифікації буває доцільно вибрати k як непарне число, оскільки це дає змогу уникнути рівності "голосів" під час визначення класу для нового спостереження.

Але є недоліки цього методу. Новий об'єкт призначається класу, який є найчастішим серед цих сусідів. Доводиться зберігати навчальну вибірку цілком. Це призводить до неефективної витрати пам'яті і надмірного ускладнення вирішального правила. При наявності похибок (як у вихідних даних, так і в моделі подібності, це може призводити до зниження точності класифікації поблизу кордону класів. Має сенс відбирати мінімальне підмножина еталонних об'єктів, дійсно необхідних для класифікації. Пошук найближчого сусіда передбачає порівняння класифікуючого об'єкта з усіма об'єктами вибірки за $O(l)$ операцій. Для задач з великими вибірками та високою частотою запитів це може виявитися проблемою. Проблема вирішується за допомогою ефективних алгоритмів пошуку найближчих сусідів, які потребують в середньому $O(\ln l)$ операцій.

2.4 Модель "дерево рішень"

Модель "Дерево рішень" представляє собою потужний інструмент, що використовується в галузі машинного навчання та аналізу даних. Воно дає

можливість створювати моделі, які спроможні класифікувати дані та приймати рішення, враховуючи вхідні параметри [24].

Модель "дерево рішень" є одним з найпопулярніших способів вирішення завдань класифікації і прогнозування в сфері Data Mining, іноді його називають також деревами вирішальних правил, деревами класифікації і регресії.

За допомогою цього методу можна вирішувати два типи завдань:

Якщо цільова змінна має дискретні значення, "дерево рішень" допомагає вирішувати завдання класифікації, тобто визначення, до якого класу належить об'єкт.

Якщо цільова змінна є чисельною і безперервною, "дерево рішень" допомагає встановити залежність цієї змінної від незалежних змінних і вирішувати завдання числового прогнозування.

Моделі "дерево рішень" можуть бути корисним інструментом для прийняття рішень в умовах невизначеності. Вони здатні аналізувати різні вхідні параметри і надавати рекомендації щодо оптимальних дій на підставі заданих критеріїв [25].

В найпростішій формі, "дерево рішень" представляє собою ієрархічну структуру, в якій правила відображаються послідовно. Ця структура базується на відповідях "Так" або "Ні" на послідовні питання.

Подібно до гілок дерева, в даному методі оцінки подій використовуються два типи напрямків: гілка рішень і гілка результатів. Гілка рішень визначає вибір конкретного рішення і представлена вершиною з гілками, що виходять з неї (рис 2.19). З іншого боку, гілка результатів відображає випадкові події, визначаючи можливі наслідки зовнішніх факторів. Кожна гілка представляє конкретний можливий результат, і числове значення (наприклад, P_i) асоційоване з кожною гілкою вказує ймовірність виникнення даної події. Під час побудови дерева проводиться оцінка кожного шляху, зазвичай на основі ймовірностей, і обирається варіант дій з найменшою ймовірністю. Кожен варіант отримує кількісну оцінку, що допомагає в прийнятті рішення.

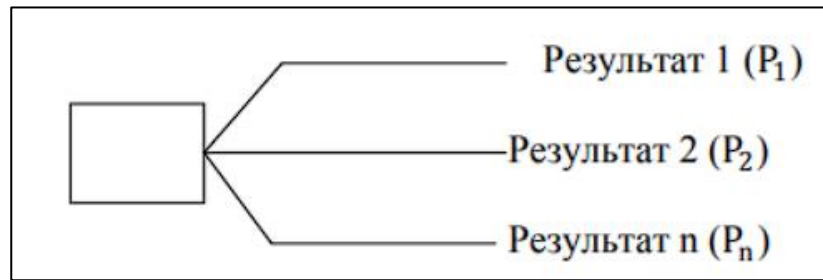


Рисунок 2.19 – Принциповий вигляд моделі "дерево рішень"

У випадках, коли існує послідовність безлічі рішень або станів середовища, що взаємодіють один з одним, застосовується модель "дерево рішень". Цей підхід дозволяє моделювати сценарії, де подальші рішення ґрунтуються на результаті попередніх, і може включати кілька послідовних станів середовища з відомими або заданими ймовірностями.

"Дерево рішень" часто використовується для оцінки ризиків у проектах, де інвестиції здійснюються протягом тривалого періоду. На рисунку 2.20 показана послідовність рішень і станів середовища, а також вказує відповідні ймовірності та виграші для різних комбінацій альтернатив і станів середовища.

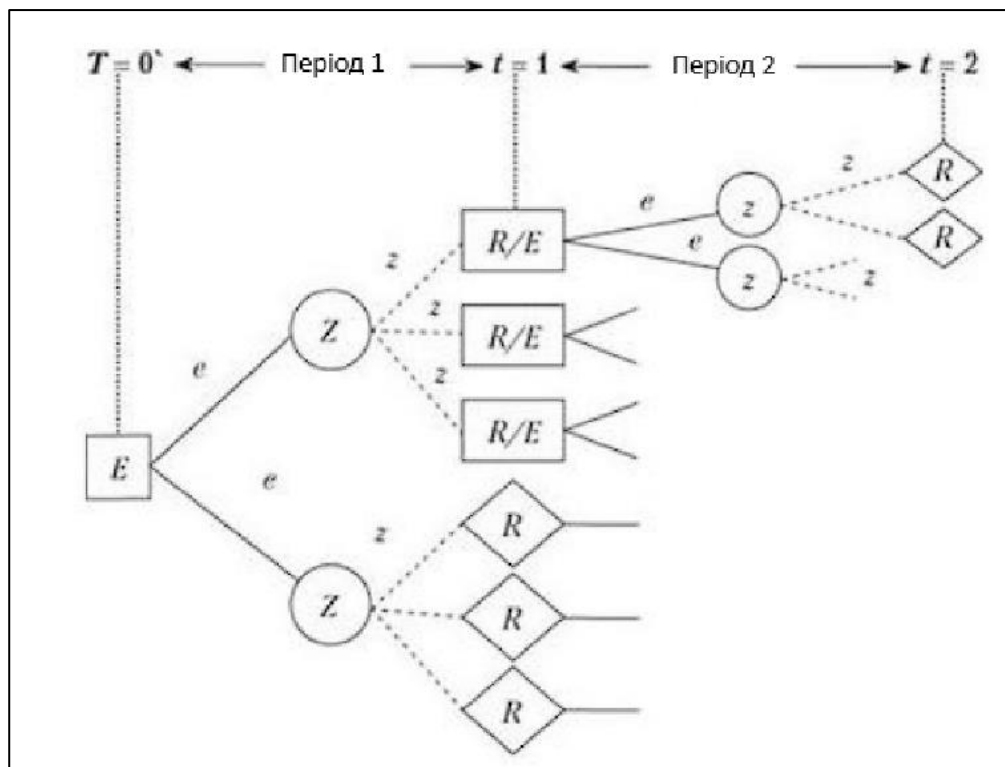


Рисунок 2.20 – Формальна структура моделі "дерево рішень"

Е - це вузол рішення, що вказує на момент прийняття конкретного рішення. Кожна лінія "е" представляє альтернативу рішення. Z - це вузол події, який відображає випадкові події. Кожна лінія "z" описує стан навколишнього середовища, який є наслідком виникнення випадкової події. R - це вузол результату, що показує наслідки, пов'язані з конкретними альтернативними рішеннями і станами середовища. R/E - це вузол, який вказує на наявність певного результату і необхідність прийняття рішення [26].

Графік моделі "дерево рішень" має три основні компоненти, які можуть змінюватися в залежності від складності задачі:

- Поле дій (поле можливих альтернатив): Тут перераховані всі можливі альтернативи дій для вирішення проблеми;

- Поле можливих подій (поле ймовірностей подій): Вказані можливі сценарії для реалізації кожної альтернативи, а також визначені ймовірності виникнення цих сценаріїв;

- Поле можливих наслідків (поле очікуваних результатів): Кількісно оцінюються наслідки для кожної ситуації.

Також графік містить три типи вузлів:

- Перша точка прийняття рішення: Це місце, де приймається остаточне рішення, і зображено у вигляді чотирикутника (квадрата);

- Точка можливостей: Зображена у вигляді кола, вона описує очікувані результати можливих подій;

- Гілки дерева: Представлені лініями, вони ведуть від першої точки прийняття рішення до результатів реалізації кожної альтернативи. Замикаючі вузли зображуються у вигляді трикутників.

Модель "дерево рішень" використовується для аналізу рішень і враховує очікувані виграші, дозволяючи приймати остаточний вибір найкращої альтернативи. Цей метод використовується як інструмент для візуалізації та аналізу сценаріїв розвитку проекту, де розглядаються ймовірності та час.

Вхідні дані, які використовуються для створення моделі "дерево рішень", можуть бути представлені у вигляді таблиці або матриці. Кожен рядок таблиці

відповідає окремому об'єкту, а кожний стовпець вказує на конкретну характеристику цих об'єктів. Наприклад, вхідні параметри можуть включати такі властивості, як вік, стать, дохід, освіту та інші.

Критерії розгалуження визначають, які тести або умови слід використовувати для прийняття рішень при створенні моделі "дерево рішень". Наприклад, критерієм може бути використання ентропії або оцінка чистоти вузлів для вибору найкращого способу розгалуження.

Створення моделі "дерево рішень" включає кілька етапів у своєму алгоритмі. Розглянемо основні кроки процесу створення:

На початку процесу побудови моделі "дерево рішень" вибирається початковий вузол, відомий як кореневий вузол, на підставі критеріїв розгалуження. Вибір кореневого вузла здійснюється з метою максимізувати інформаційну цінність або зменшити ентропію системи.

Процес розгалуження виконується рекурсивно і триває до виконання певної умови зупинки. Ця умова може включати досягнення певної глибини дерева, вичерпання всіх вхідних параметрів або досягнення певної кількості об'єктів у вузлі.

Оцінка моделі "дерево рішень" може проводитися за допомогою різних критеріїв. Нижче наведено декілька з таких критеріїв:

Одним з основних критеріїв оцінки є точність або ефективність моделі "дерево рішень" при проведенні класифікації або прогнозуванні. Іншим важливим аспектом є розмір дерева, оскільки більші дерева можуть бути складнішими для інтерпретації і вимагати більше обчислювальних ресурсів.

Під час оцінки критеріїв можна вибрати найкраще "дерево рішень" з серед кількох побудованих варіантів. Це вибір може ґрунтуватися на тому, яке дерево має найвищу точність або на тому, яке забезпечує оптимальний баланс між точністю та складністю моделі.

При оцінці критеріїв можна обрати найкращу модель "дерево рішень" з серед численних побудованих варіантів. Вибір може залежати від того, яке дерево

володіє найвищою точністю або відповідає оптимальному балансу між точністю та складністю моделі.

Алгоритмічно цей принцип реалізується так: маємо навчальну множину S , яка складається з n прикладів, і для кожного з них визначено мітку класу $C_i (i = 1..k)$, і m атрибутів $A_j (j = 1..m)$ які, як передбачається, визначають приналежність об'єкту до того або іншого класу.

Тож має три випадки:

- Всі приклади множини S мають однакову мітку класу C_i , що означає, що всі навчальні приклади належать лише до одного класу. Очевидно, що навчання в цьому випадку не має сенсу, оскільки усі приклади, які представляються моделі, належать одному класу. "Дерево рішень" в такому випадку буде мати лише один листок, який асоціюється з класом C_i . В практичному застосуванні таке дерево є безглуздом, оскільки будь-який новий об'єкт воно відноситиме тільки до цього одного класу;

- Множина S взагалі не містить прикладів, тобто є порожньою. У цьому випадку для неї також буде створений лист (застосування правила для створення вузла до порожньої множини немає сенсу), а клас для цього листа може бути вибраний з іншої множини (наприклад, клас, який найчастіше зустрічається в батьківській множині);

- Множина S містить навчальні приклади усіх класів C_k . У цьому випадку необхідно розбити множину S на підмножини, які асоціюються з класами. Для цього обирається один із атрибутів A_j множини S , який має два чи більше унікальних значень (a_1, a_2, \dots, a_p) , де p – кількість унікальних значень ознаки. Потім множина S розбивається на p підмножин (S_1, S_2, \dots, S_p) , кожне з яких включає приклади, що мають відповідне значення атрибуту. Далі обирається наступний атрибут, і процедура розбиття повторюється. Ця процедура рекурсивно виконується до тих пір, поки всі приклади в отриманих підмножинах не належать одному класу.

Вищезазначений процес є основою багатьох сучасних алгоритмів побудови дерев рішень. Зрозуміло, що при використанні цієї методики побудова моделі "дерево рішень" відбувається зверху вниз (від кореневого вузла до листя). На сьогодні існує значна кількість алгоритмів навчання дерев рішень: ID3, CART, C4.5, C5.0, NewId, ITrule, CHAID, CN2 і так далі. Проте найбільшого поширення та популярності набули такі:

ID3 (Iterative Dichotomizer 3) – це алгоритм, який працює тільки з дискретною цільовою змінною, тому моделі "дерево рішень", побудовані за його допомогою, призначені для класифікації. Кількість нащадків у вузлі дерева не обмежена. Алгоритм не може обробляти пропущені дані. Основою цього алгоритму є поняття інформаційної ентропії – міри невизначеності інформації. Для визначення наступного атрибута необхідно розрахувати ентропію для всіх невикористаних ознак щодо тестових зразків і обрати той, для якого ентропія є мінімальною. Цей атрибут вважається найбільш відповідним для класифікації.

C4.5 – це вдосконалена версія алгоритму ID3, в яку додана можливість роботи з пропущеними значеннями атрибутів. Цей алгоритм дозволяє "усікати" гілки дерева, якщо воно занадто сильно "розростається", і працювати не лише з атрибутами–категоріями, але й з числовими. Загалом, сам алгоритм виконується за тим же принципом, що і його попередник; відмінність полягає в можливості розбиття області значень незалежної числової змінної на декілька інтервалів, кожен з яких стає атрибутом. Відповідно до цього початкова множина ділиться на підмножини. Якщо дерево стає занадто великим, можливе зворотне угруповання декількох вузлів в один лист. При цьому, оскільки перед побудовою дерева помилка класифікації вже врахована, вона не збільшується.

CART (Classification and Regression Tree) – це алгоритм навчання дерев рішень, який дозволяє використовувати як дискретну, так і неперервну цільову змінну, тобто вирішувати як завдання класифікації, так і регресії. Алгоритм розроблено з метою побудови бінарних дерев рішень, де кожен вузол при розбитті має тільки двох нащадків. Його дія полягає в розділенні безлічі прикладів навпіл на кожному кроці – по одній гілці йдуть ті приклади, в яких правило виконується

(правий нащадок), а по іншій – ті, в яких правило не виконується (лівий нащадок). У процесі "зростання" на кожному вузлі дерева алгоритм проводить перебір усіх атрибутів і вибирає для наступного розбиття той, який максимізує значення показника, обчисленого за математичною формулою і залежного від відношення кількості прикладів в правому і лівому нащадку до загальної кількості прикладів.

Переваги дерев рішень включають:

- Простоту інтерпретації: Ці моделі легко інтерпретувати, оскільки їх структуру можна відобразити у вигляді графіка або діаграми;
- Використання для навчання з наглядом: "Дерево рішень" можуть працювати з наборами даних, які містять як категоріальні, так і числові атрибути;
- Ефективність з великими наборами даних: Ці моделі можуть ефективно використовуватися для обробки великих наборів даних та великої кількості атрибутів.

Недоліки дерев рішень включають:

- Надмірна чутливість до шуму: Ці моделі можуть бути вразливими до шуму або незначних змін у вхідних даних, що може призвести до прийняття неправильних рішень;
- Накопичення помилок: Під час рекурсивного розгалуження "Дерево рішень" можуть накопичуватися помилки, оскільки кожне розгалуження базується на попередніх рішеннях;
- Недостатня універсальність: "Дерево рішень" можуть бути обмеженими у здатності моделювати складні взаємозв'язки між атрибутами.

2.5 Висновки

У даному розділі був проведений ретельний розвідувальний аналіз даних, спрямований на вивчення особливостей датасету та встановлення взаємозв'язків між різними факторами. Огляд датасету дав можливість отримати уявлення про структуру та характеристики даних, що є ключовим для подальшого розуміння та використання їх у контексті рейтингування проєктів ІСО.

Описано алгоритми рейтингування, які використовуються для оцінки та ранжування проєктів ICO. У цьому контексті були використані метод опорних векторів (SVM), метод k-nearest neighbor (k-NN) і модель "дерево рішень". Кожен з цих методів має свої особливості та переваги.

Метод опорних векторів (SVM) відзначається високою точністю в роботі з великими обсягами даних та здатністю ефективно вирішувати завдання класифікації. Метод k-nearest neighbor (k-NN) є простим у використанні та може виявити взаємозв'язки в даних, особливо в разі невеликих вибірок. "Дерево рішень", з іншого боку, дозволяє візуалізувати рішення та враховує важливі фактори в процесі класифікації.

Розглядаючи кожен метод, були визначені їхні переваги і недоліки. Наприклад, SVM може бути витратним з точки зору обчислювальних ресурсів, k-NN чутливий до викидів та шуму, а модель "дерево рішень" можуть бути схильними до перенавчання.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЮ РЕЙТИНГУ ІСО ПРОЄКТІВ

3.1 Підготовка вибірки

Виконуємо конвертацію категоріальних змінних у числовий формат (рис 3.1), щоб їх можна було використовувати в алгоритмах машинного навчання, які приймають на вхід числові дані.

```
categorical_columns = []
numerics = ['int8', 'int16', 'int32', 'int64', 'float16', 'float32', 'float64']
features = data.columns.values.tolist()
for col in features:
    if data[col].dtype in numerics: continue
    categorical_columns.append(col)
categorical_columns
for col in categorical_columns:
    if col in data.columns:
        le = LabelEncoder()
        le.fit(list(data[col].astype(str).values))
        data[col] = le.transform(list(data[col].astype(str).values))
```

Рисунок 3.1 – Конвертація даних

Виконуємо підготовку даних для тренування та тестування моделі машинного навчання (рисунок 3.2). Наступним кроком створюється новий набір даних `train`, який є копією оригінального набору даних `data`, але без змінної `"rating"`. Змінна `"rating"` видаляється за допомогою методу `drop`, де `axis=1` вказує, що ми видаляємо стовпець (оскільки `"rating"` – це стовпець). Визначаємо цільової змінної `"rating"` і розділення даних на тренувальний та тестувальний набори.

`train_test_split` – це функція, яка розбиває дані на тренувальний та тестувальний набори. Параметр `test_size = 0.2` вказує на те, що 20% даних буде використано для тестування, а решта – для тренування. Параметр `random_state = 42` встановлює випадковий стартовий стан для забезпечення відтворюваності результатів.

```
train = data.drop('rating', axis=1)
target = data['rating']
X_train, X_test, y_train, y_test = train_test_split(train, target, test_size=0.2, random_state=42)
```

Рисунок 3.2 – Розбивання даних на вибірки

Побудуємо модель оцінювання рейтингу ICO–проектів у такому вигляді:

$$Y = F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}), \quad (3.1)$$

де x_1 – роль токена;

x_2 – категорія;

x_3 – мета;

x_4 – загальна кількість токенів;

x_5 – ціна ICO;

x_6 – кількість залучених коштів;

x_7 – мінімальний особистий капітал;

x_8 – максимальний особистий капітал;

x_9 – назва криптовалют які можливі для покупки ICO;

x_{10} – білий список;

x_{11} – назва криптовалют які можливі для покупки ICO;

x_{12} – відсоток який доступний для продажу.

Y – множина класів {Низький, Середній, Високий}.

Для розв'язання задачі застосуємо методи машинного навчання:

- метод опорних векторів;
- метод найближчих сусідів;
- модель "дерево рішень".

3.2 Передбачення рейтингу ICO проектів методом опорних векторів

Використаємо модель Support Vector Classifier (SVC) з бібліотеки scikit-learn для навчання на тренувальних даних і оцінює точність (accuracy) моделі на цих даних. Об'єкт svc представляє собою модель, яку ми хочемо навчити.

Спочатку створюємо об'єкта класифікатора методу опорних векторів (SVM). `random_state=42` встановлює фіксоване значення для випадкових чисел, щоб результати були відтворювані.

Використаємо метод `fit` є основним методом для навчання моделі в бібліотеці `scikit-learn`. У цьому випадку, `svm_classifier` є об'єктом класифікатора методу опорних векторів (SVC). Метод `fit` отримує два аргументи:

`X_train`: Це матриця ознак (features), яка містить вхідні дані для навчання моделі. Кожен рядок матриці представляє один зразок, а кожний стовпець представляє окремию ознаку (або атрибут) цього зразка.

`y_train`: Це вектор міток класів, які відповідають зразкам в `X_train`. Кожен елемент вектора `y_train` вказує на клас, до якого належить відповідний зразок в `X_train`.

Використаємо `svm_classifier.predict()`. Здійснюємо прогнози на навчальних даних (`X_train`). Метод `predict` використовує навчену модель для прогнозування класів для кожного зразка в матриці ознак `X_train`. Результатом є вектор `y_train_pred`, який містить прогнозовані класи для кожного зразка навчальної вибірки. Потім з використанням цієї ж функції здійснюємо прогнози на тестових даних (`X_test`). Аналогічно до попереднього пункту, метод `predict` використовує навчену модель для прогнозування класів для кожного зразка в матриці ознак `X_test`. Результатом є вектор `y_test_pred`, який містить прогнозовані класи для кожного зразка тестової вибірки.

Використаємо функцію `accuracy_score()`. Дана функція потрібна для обчислення точності на навчальних даних.

Результат, який зберігається у змінній `accuracy_train`, представляє собою відсоток правильно класифікованих зразків серед усіх навчальних зразків.

На рисунку 3.3 зображено передбачення методом опорних векторів на навчальній та тестовій вибірці.

```

svm_classifier = SVC(random_state=42)

svm_classifier.fit(X_train, y_train)

y_train_pred = svm_classifier.predict(X_train)
y_test_pred = svm_classifier.predict(X_test)

accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)

print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")

```

Рисунок 3.3 – Передбачення рейтингу ІСО проєктів методом опорних векторів на навчальній та тестовій вибірці

На рисунку 3.4 показано точність на навчальних та тестових даних. Та можемо побачити що точність на навчальних даних 75%, та на тестових даних 73%.

```

Точність на навчальних даних: 75.56 %
Точність на тестових даних: 73.63 %

```

Рисунок 3.4 – Результати передбачення методом опорних векторів

Наступним кроком побудуємо матрицю помилок для навчальної та тестової вибірки (рисунок 3.5).

```

cm_train = confusion_matrix(y_train, y_train_pred)
cm_test = confusion_matrix(y_test, y_test_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm_train, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2], yticklabels=[0, 1, 2])
plt.title('Матриця помилок на навчальній вибірці')
plt.xlabel('Предбачені значення')
plt.ylabel('Справжні значення')
plt.show()

plt.figure(figsize=(8, 6))
sns.heatmap(cm_test, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2], yticklabels=[0, 1, 2])
plt.title('Матриця помилок на тестовій вибірці')
plt.xlabel('Предбачені значення')
plt.ylabel('Справжні значення')
plt.show()

```

Рисунок 3.5 – Побудова матриці помилок методом опорних векторів на навчальній та тестовій вибірці

Матриця помилок (confusion matrix) визначає, як модель класифікації правильно або неправильно класифікує приклади в кожен клас. У тепловій карті (heatmap), яка відображає матрицю помилок, можна визначити такі елементи:

Головна діагональ (зліва вгору до права вниз): Кожен елемент у цій діагоналі представляє собою кількість прикладів, які були правильно класифіковані для відповідного класу. Наприклад, елемент у рядку 1, стовпці 1 вказує на кількість прикладів, які були правильно класифіковані як клас 1.

Не головна діагональ: Елементи поза головною діагоналлю представляють собою кількість прикладів, які були помилково класифіковані. Наприклад, елемент у рядку 1, стовпці 2 вказує на кількість прикладів, які насправді належать класу 1, але були помилково класифіковані як клас 2.

Анотації (числові значення в кожному квадраті): Ці значення показують кількість прикладів для кожної комбінації передбачених та справжніх класів.

Використаємо функцію `confusion_matrix` з бібліотеки `scikit-learn` для обчислення матриці помилок на навчальних даних. Матриця помилок визначає, скільки елементів кожного класу було правильно або неправильно класифіковано на навчальних даних. Аналогічно до попереднього пункту, обчислюємо матрицю помилок для тестових даних. `plt.figure(figsize=(8, 6))`: Створюється фігура для графіку, яка має розміри 8 на 6 дюймів. Створюємо теплову карту за допомогою бібліотеки `seaborn`.

На рисунку 3.6 побудовано матрицю помилок на навчальній вибірці.



Рисунок 3.6 – Матриця помилок методом опорних векторів на навчальній вибірці

На даному рисунку (рис3.6)матриці можемо зробити висновок 1 приклад був правильно класифікований до класу 0, 261 приклади були правильно класифікований до класу 2 та 10 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.

На рисунку 3.7 побудовано матрицю помилок на тестовій вибірці.

З даної матриці можемо зробити висновок 0 приклад був правильно класифікований до класу 0, 63 приклади були правильно класифікований до класу 2 та 4 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.

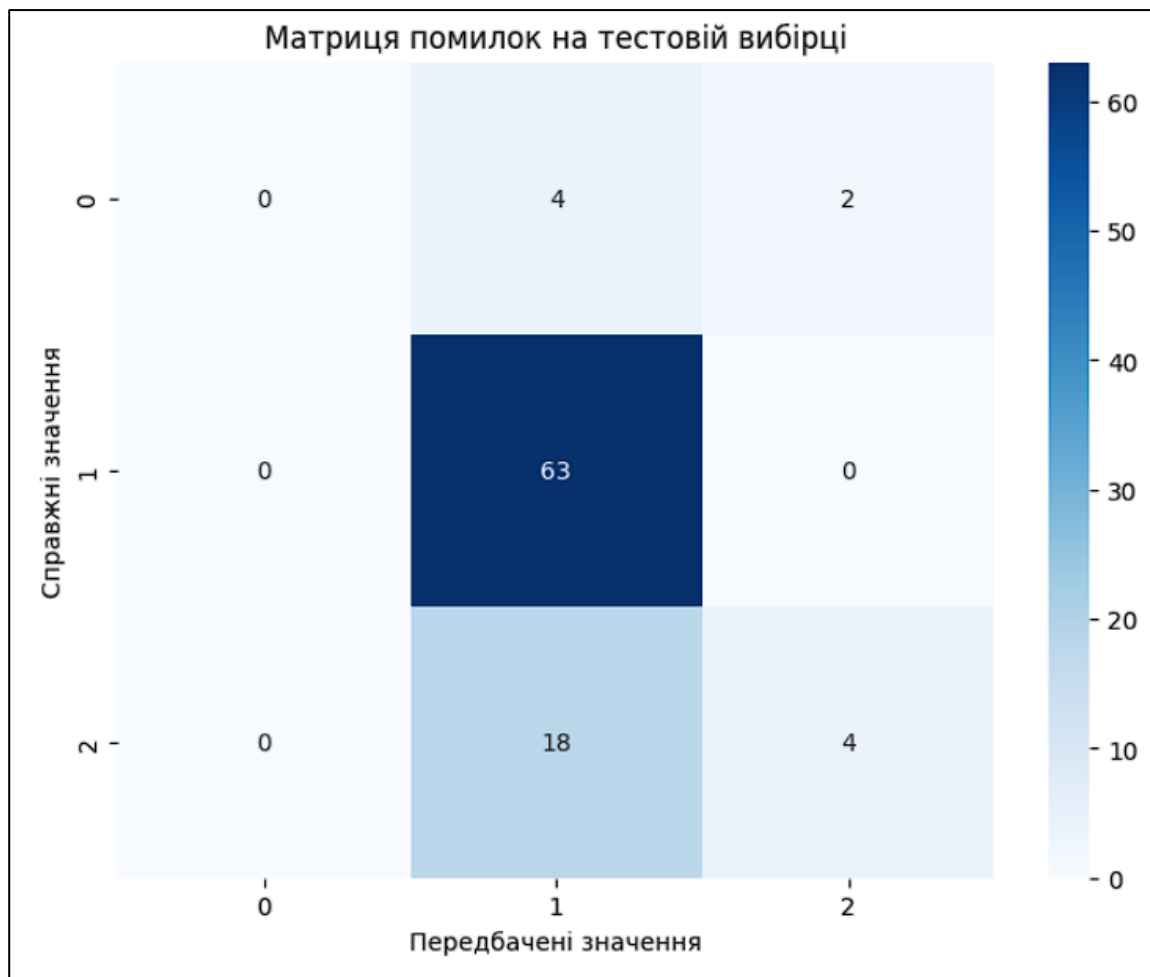


Рисунок 3.7 – Матриця помилок методом опорних векторів на навчальній вибірці

3.3 Передбачення рейтингу ICO проєктів методом k-nearest neighbor

Використаємо бібліотеку scikit-learn для створення та навчання моделі класифікації k-найближчих сусідів (KNN) за допомогою конвеєра (Pipeline). StandardScaler використовується для центрування та нормалізації значень ознак, щоб забезпечити стабільність та ефективність алгоритму KNN. Використовуємо класифікатор k-найближчих сусідів з встановленими параметри n_neighbors=3 вказують, що модель буде використовувати 3 найближчих сусідів для класифікації, а p=3 визначає відстань Мінковського порядку 3 (Euclidean distance).

Наступним кроком за допомогою функції knn_classifier.fit() навчаємо модель. Прогнозуємо класи за допомогою функції knn_pipeline.fit() за допомогою навченої моделі. Використаємо функцію accuracy_score для обчислення точності моделі на навчальних даних.

На рисунку 3.8 зображено передбачення методом k-nearest neighbor на навчальній вибірці. На рисунку 3.9 зображено точність передбачення на навчальній та тестовій вибірці.

```
knn_pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('knn_classifier', KNeighborsClassifier(n_neighbors=3,p=3))
])

knn_pipeline.fit(X_train, y_train)

y_train_pred = knn_pipeline.predict(X_train)
y_test_pred = knn_pipeline.predict(X_test)

accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)

print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")
```

Рисунок 3.8 – Передбачення рейтингу ICO проєктів методом k-nearest neighbor на навчальній та тестовій вибірці

```
Точність на навчальних даних: 80.28 %
Точність на тестових даних: 74.73 %
```

Рисунок 3.9 – Результати передбачення методом k-nearest neighbor на навчальній та тестовій вибірці

Також побудуємо матрицю помилок для методу k-nearest neighbor. На рисунку 3.10 побудовано матрицю помилок на навчальній вибірці. На рисунку 3.11 побудовано матрицю помилок на тестовій вибірці.



Рисунок 3.10 – Матриця помилок методом k-nearest neighbor на навчальній вибірці

На даному рисунку (рис 3.10) матриці можемо зробити висновок 9 прикладів був правильно класифікований до класу 0, 244 прикладів були правильно класифікований до класу 2 та 36 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.

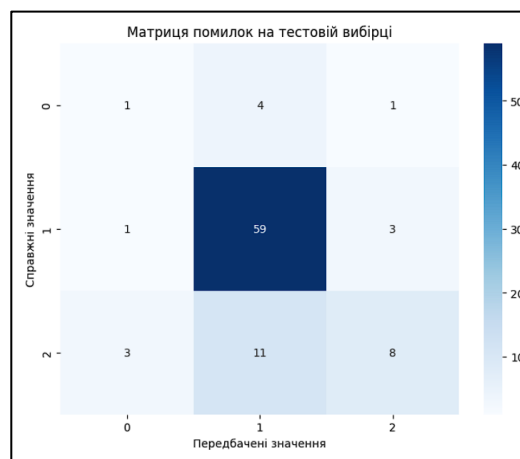


Рисунок 3.11 – Матриця помилок методом k-nearest neighbor на тестовій вибірці

На даному рисунку (рис 3.11) матриці можемо зробити висновок 1 приклад був правильно класифікований до класу 0, 59 прикладів були правильно класифікований до класу 2 та 8 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.

3.4 Передбачення рейтингу ІСО проєктів за допомогою моделі "дерево рішень"

Використаємо бібліотеку scikit-learn для створення та навчання моделі класифікації за допомогою випадкового лісу (Random Forest). Функція `random_forest_classifier.fit` навчає модель на навчальних та тестових даних.

Функція `random_forest_classifier.predict` прогнозує класи для навчальних та тестових даних за допомогою навченої моделі. За допомогою функції `accuracy_score` обчислюємо точність моделі на навчальних та тестових даних.

На рисунку 3.12 зображено передбачення моделлю "дерево рішень" на навчальній вибірці.

```
random_forest_classifier = RandomForestClassifier(random_state=42, n_estimators=1000, max_depth=8)
random_forest_classifier.fit(X_train, y_train)

y_train_pred = random_forest_classifier.predict(X_train)
y_test_pred = random_forest_classifier.predict(X_test)

accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)

print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")
```

Рисунок 3.12 – Передбачення рейтингу ІСО проєктів за допомогою моделі "дерево рішень" на навчальній та тестовій вибірці

На рисунку 3.13 показано точність на навчальних та тестових даних. Та можемо побачити що точність на навчальних даних 86%, та на тестових даних 82%.

Точність на навчальних даних: 86.94 %
Точність на тестових даних: 82.42 %

Рисунок 3.13 – Результати передбачення моделі "дерево рішень" на навчальній та тестовій вибірці

Також побудуємо матрицю помилок для моделі "дерево рішень" на навчальній та тестовій вибірці (рис 3.14, рис 3.15).

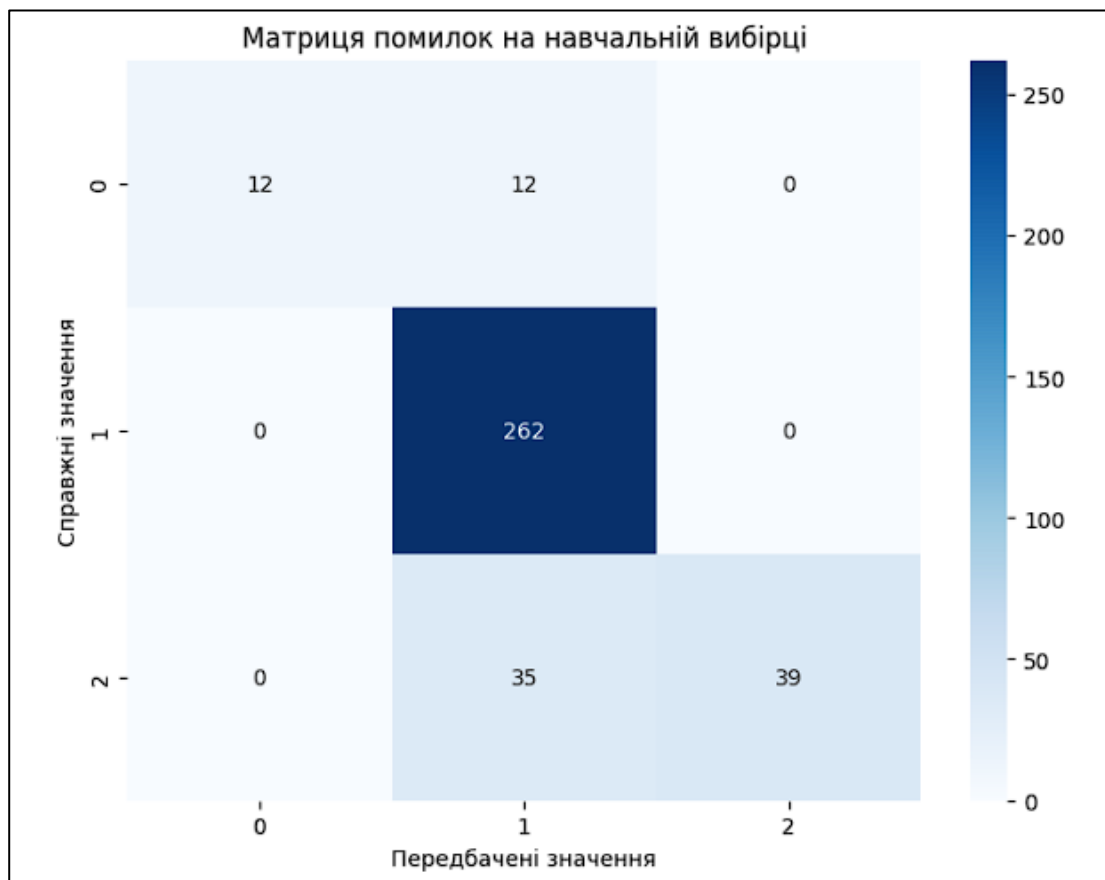


Рисунок 3.14 – Матриця помилок моделі "дерево рішень" на навчальній вибірці

На даному рисунку (рис 3.14) можемо зробити висновок 12 приклади був правильно класифікований до класу 0, 262 прикладів були правильно класифікований до класу 2 та 39 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.



Рисунок 3.15 – Матриця помилок моделі "дерево рішень" на навчальній вибірці

На даному рисунку (рис 3.15) можемо зробити висновок 1 приклад був правильно класифікований до класу 0, 62 прикладів були правильно класифікований до класу 2 та 12 прикладів було правильно класифіковано до класу 3. Інші значення які знаходяться не по головній діагоналі вказують на помилки моделі.

Наступним кроком побудуємо графік важливості ознак для тестової та навчальної вибірки (рис 3.16).

```
plt.figure(figsize=(10, 6))
feature_importance = tree_classifier.feature_importances_
feature_names = X_train.columns
sns.barplot(x=feature_importance, y=feature_names, orient="h")
plt.title('Важливість ознак при побудові дерева рішень (навчальна вибірка)')
plt.show()

plt.figure(figsize=(10, 6))
feature_importance_test = tree_classifier.feature_importances_
sns.barplot(x=feature_importance_test, y=feature_names, orient="h")
plt.title('Важливість ознак при побудові дерева рішень (тестова вибірка)')
plt.show()
```

Рисунок 3.16 – Побудова графіків залежності ознак

На рисунку 3.17 можемо побачити що найбільшою ознакою за важливістю є goal. Це означає, ціль яку хоче зібрати проєкт є найбільш інформативним атрибутом для розбиття даних. Менш важливими є ознаки whitelist received_money fundraising_goal, total_tokens, ico_price. Ще меншими є accepts_2 avialiable_for_sale, token_type, max_personal_cap, category та role_of token.

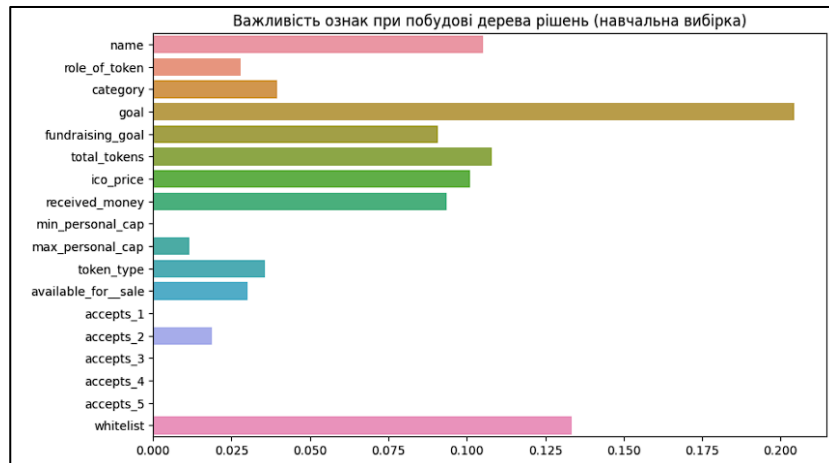


Рисунок 3.17 – графік важливості ознак для моделі "дерево рішень" для навчальної вибірки

На рисунку 3.18 ми можемо спостерігати що важливість ознак на тестовій та навчальній вибірці однакова.

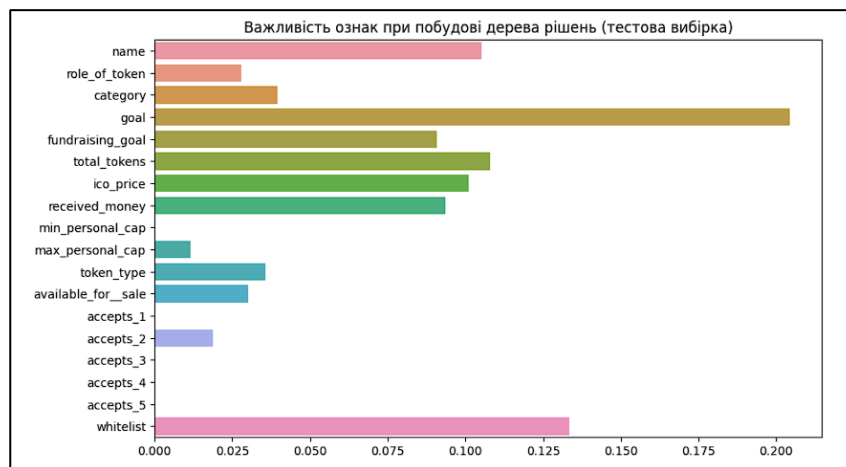


Рисунок 3.18 – графік важливості ознак для моделі "дерево рішень" для тестової вибірки

3.5 Висновки

У даному розділі була розроблена інформаційна технологія для передбачення рейтингу ІСО проєктів, використовуючи методи опорних векторів, метод k-nearest neighbor та моделлю "Дерево рішень".

Дослідження показали, що точність передбачення методом опорних векторів на навчальному дорівнює 75% та тестовому наборі даних дорівнює 73%. Це свідчить про те, що алгоритм добре справляється з класифікацією даних і правильно класифікує 73% з них.

Точність передбачення методом k-nearest neighbor на навчальному та тестовому наборі даних складає 80%. Це свідчить про гарну ефективність алгоритму в класифікації та правильну класифікацію 74% даних.

Модель "дерево рішень" показав найвищу точність – 86% на навчальному, на тестовому наборі даних він показав 82%. Це говорить про те, що алгоритм добре справляється з класифікацією даних і правильно класифікує всі з них.

Додатково, були побудовані матриці помилок для методів опорних векторів та k-nearest neighbor, які вказують на ефективність цих алгоритмів в роботі з реальними даними. Також був побудований графік важливості ознак для моделі "дерево рішень", що дозволяє виділити найбільш значущі фактори в класифікації.

У цьому розділі було продемонстровано високу ефективність розроблених алгоритмів та їх здатність до точного передбачення рейтингу ІСО проєктів.

4 ЕКОНОМІЧНА ЧАСТИНА

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

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями [27]. Результати занесені до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

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

Продовження таблиці 4.1

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
6. Ринкові перспективи (розмір ринку)	3	2	3
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	3	4	3
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	5	4	4
12. Практична здійсненність (розробка документів)	4	5	4
Сума балів	37	40	41
Середньоарифметична сума балів CB_c	39,3		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [28].

Загалом в роботі здійснено створення Інформаційної технології аналізу та передбаченню рейтингу ІСО проєктів.

Відмінні ознаки розробленої технології розширюваність (можна підключити дану технологію до інших), автономність (можливе використання для кількох проєктів).

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» становить 39,3 бала, що, відповідно до [28], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розраховуємо за формулою:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має

виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Швидкість оцінювання	с	0,7	0,3	1,8	0,2
Швидкість генерації графіків оцінювання	с	0,1	0,03	2,33	0,25
Завантаження диску	%	50	20	2,5	0,2
Завантаження процесора	%	50	20	2,5	0,2

Продовження таблиці 4.2

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Використовуваний об'єм оперативної пам'яті	МБ	290	120	2,42	0,1
Масштабованість	екземпляр	1	5	5	0,25

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,8 \cdot 0,2 + 2,33 \cdot 0,25 + 2,5 \cdot 0,2 + 2,42 \cdot 0,1 + 5 \cdot 0,25 + 5 \cdot 0,25 = 3,58.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 3,58 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проектів», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [28]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

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

t_i – число днів роботи конкретного дослідника, днів.;

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

$$Z_o = 17000,00 \cdot 35 / 21 = 28833,33 \text{ (грн).}$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	17000,00	809,52	35	28833,33
Інженер-розробник програмного забезпечення 1-ї категорії	12000,00	714,28	35	25000
Всього				53833,33

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

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

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

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

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де M_M – розмір мінімальної місячної заробітної плати, прийmemo $M_M=6700,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення;

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

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

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1,10 \cdot 1,65 / (21 \cdot 8) = 72,38 \text{ (грн).}$$

$$З_{р1} = 72,38 \cdot 8,30 = 600,79 \text{ (грн).}$$

Занесемо наші розрахунки до таблиці 4.4.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця розробника програмного забезпечення	8,30	2	1,10	72,38	600,79
Інсталяція програмного забезпечення середовища розробки і моделювання	6,20	3	1,35	88,83	550,78
Компіляція програмного коду	7,50	5	1,70	111,87	839,00
Всього					1990,56

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

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.5)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (53833.33 + 1990,56) \cdot 10 / 100\% = 5582,88 \text{ (грн)}.$$

4.3.2 Відрахування на соціальні заходи

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

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{zn}}{100\%}, \quad (4.6)$$

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

$$Z_n = (53833.33 + 1990,56 + 5582,88) \cdot 22 / 100\% = 17083,46 \text{ грн}.$$

4.3.3 Сировина та матеріали

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

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\epsilon j}, \quad (4.7)$$

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

n – кількість видів матеріалів;

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

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

B_j – маса відходів j -го найменування, кг;

C_{vj} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 2,0 \cdot 283,00 \cdot 1,1 - 0 \cdot 0 = 622,60 \text{ (грн).}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір А4 500	283,00	2,0	-	-	622,60
Папір для записів А5 250	154,00	3,0	-	-	508,20
Органайзер офісний	200,00	3,0	-	-	660,00
Набір канцелярський офісний	210,00	3,0	-	-	693,00
Картридж для принтера	980,00	1,0	-	-	1078,00
Диск оптичний CD-RW	23,00	2,0	-	-	50,60
Flesh-пам'ять 32 GB	410,00	1,0	-	-	451,00
Всього					4063,40

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_e = \sum_{j=1}^n H_j \cdot C_j \cdot K_j, \quad (4.8)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

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

$$K_b = 1 \cdot 120,00 \cdot 1,1 = 132,00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Дата кабель OTG USB 3.0 AF to Type-C 0.2m Cablexpert (A-OTG-CMAF3-01)	1	120,00	132,00
Дата кабель USB 2.0 AM to Type-C 2.0m black ColorWay (CW-CBUC008-BK)	1	160,00	176,00
Всього			308,00

4.3.5 Спец. устаткування для наукових (експериментальних) робіт

Балансову вартість спец. устаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.9)$$

де C_i – ціна придбання одиниці спец. устаткування даного виду, марки, грн;

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

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 25000,00 \cdot 1 \cdot 1,1 = 27500 \text{ (грн).}$$

Отримані результати зведемо до таблиці 4.7:

Таблиця 4.7 – Витрати на придбання спец. устаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Мобільний телефон Xiaomi MI 12X 8/128GB Gray	1	25000,00	27500.0
Ноутбук ASUS ROG Strix G15 (G513RC-HN033)	1	48 770	53647.00
Всього			81147.00

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прг}} = \sum_{i=1}^k C_{\text{инрг}} \cdot C_{\text{прг.}i} \cdot K_i, \quad (4.10)$$

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

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

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

k – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 7860,00 \cdot 1 \cdot 1,11 = 8724,60 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.8:

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне програмне забезпечення розробки та моделювання	1	7860,00	8724,60
Всього			8724,60

4.3.7 Амортизація обладнання, програмних засобів та приміщень

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

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

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

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

$$A_{обл} = (24650,00 \cdot 2) / (2 \cdot 12) = 2054,17 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер проведення розробки та моделювання	24650,00	3	1	2054,17

Продовження таблиці 4.9

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Робоче місце інженера-розробника програмного забезпечення	1000,00	5	1	416,67
Пристрої передачі даних	9000,00	4	1	312,92
Пристрій виводу інформації	6740,00	5	1	224,67
Оргтехніка	6750,00	4	1	281,25
Приміщення лабораторії	620000,00	25	1	4133,33
ОС Windows 11	8570,00	2	1	714,17
Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Прикладний пакет Microsoft Office 2019	7825,00	2	1	652,08
Всього				9518,36

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.12)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; прийmemo $C_e = 7,50$ грн;

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

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

$$B_e = 0,25 \cdot 320,0 \cdot 7,50 \cdot 0,95 / 0,97 = 185,71 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер проведення розробки та моделювання	0,25	320,0	185.71
Робоче місце інженера-розробника програмного забезпечення	0,12	300,0	111.34
Пристрої передачі даних	0,01	250,0	18.5
Пристрій виводу інформації	0,42	15,0	54,5
Оргтехніка	0,50	5,0	93,29
Всього			463.34

4.3.9 Службові відрядження

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

$$B_{cb} = (Z_o + Z_p) \cdot \frac{H_{cb}}{100\%}, \quad (4.13)$$

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

$$B_{cb} = (31000 + 1990,56) \cdot 20 / 100\% = 6298.11 \text{ (грн)}.$$

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

Витрати на роботи, які виконують сторонні підприємства, установи і організації відсутні.

4.3.11 Інші витрати

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

$$I_{\text{в}} = (z_o + z_p) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (4.14)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати», приймемо $H_{\text{ів}} = 50\%$.

$$I_{\text{в}} = (31000 + 1990,56) \cdot 50 / 100\% = 16495.28 \text{ (грн)}.$$

4.3.12 Накладні (загальновиробничі) витрати

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

$$B_{\text{нзв}} = (z_o + z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.15)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $H_{\text{нзв}} = 100\%$.

$$B_{\text{нзв}} = (31000 + 1990,56) \cdot 100 / 100\% = 32990,56 \text{ (грн)}.$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + M + K_e + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_e + B_{\text{нзв}}. \quad (4.16)$$

$$B_{\text{заг}} = 31000 + 1990,56 + 5582,88 + 17083,46 + 4063,40 + 308,00 + 81147,00 + 8724,60 + 9518,36 + 463,34 + 16495,28 + 16495,28 + 32990,56 = 315103,94 \text{ (грн)}.$$

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

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,95$.

$$ZB = 315103,94 / 0,95 = 331689,4 \text{ (грн)}.$$

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

Результати дослідження проведені за темою «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» передбачають комерціалізацію протягом 4-х років реалізації на ринку і відповідають випадку - розробка чи

суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

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

ΔN – збільшення кількості споживачів продукту зображено у таблиці 4.11, у періоди часу, що аналізуються, від покращення його певних характеристик;

Таблиця 4.11 – Кількість споживачів

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	2000	7000	15000	8500

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 35000 осіб;

C_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 1020,00 грн;

$\pm \Delta C_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo зростання на 140 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [28]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.18)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 35\%$;

\mathcal{G} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\mathcal{G} = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (140 \cdot 35000,00 + 1160 \cdot 2000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 2102228.84 \text{ (грн)}.$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (140 \cdot 35000,00 + 1160 \cdot 9000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 4431199.56 \text{ (грн)}.$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (140 \cdot 35000,00 + 1160 \cdot 24000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 9458180.42 \text{ (грн)}.$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (140 \cdot 35000,00 + 1160 \cdot 32500) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 10816868.25 \text{ (грн)}.$$

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

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.19)$$

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

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

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,15$;

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

$$ПП = 2102228.84 / (1 + 0,15)^1 + 4431199.56 / (1 + 0,15)^2 + 9458180.42 / (1 + 0,15)^3 + 10816868.25 / (1 + 0,15)^4 = 16578674.52 \text{ (грн).}$$

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

$$PV = k_{инв} \cdot 3B, \quad (4.20)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2,1$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 331689,4 грн.

$$PV = k_{инв} \cdot 3B = 2,1 \cdot 331689,4 = 696547.74 \text{ (грн).}$$

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

$$E_{абс} = ПП - PV \quad (4.21)$$

де PII – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 23940167.39 (грн);

PV – теперішня вартість початкових інвестицій, 552904,11 (грн).

$$E_{abc} = PII - PV = 16578674.52 - 696547.74 = 15882126.78 \text{ (грн)}.$$

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

$$E_g = T_{жс} \sqrt{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.22)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 23940167.39 грн;

PV – теперішня вартість початкових інвестицій, 696547.74 грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_g = T_{жс} \sqrt{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 15882126.78 / 696547.74)^{1/4} = 4,372$$

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

$$\tau_{мін} = d + f, \quad (4.23)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,11$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,3.

$\tau_{\min} = 0,11 + 0,3 = 0,41 < 2,466$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , вища мінімальної внутрішньої дохідності.

Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» доцільно.

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

$$T_{ок} = \frac{1}{E_g}, \quad (4.24)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 4,372 = 0,2 \text{ р.}$$

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

4.5 Висновки

Згідно проведених досліджень, рівень комерційного потенціалу розробки за темою "Інформаційна технологія аналізу та передбачення рейтингу ІСО проєктів" складає 39,3 бали, що свідчить про високу комерційну важливість проведення цих досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, використовуючи узагальнений коефіцієнт якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 3,58 рази. Також термін окупності становить 0,2 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної

розробки і може заохочувати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою "Інформаційна технологія аналізу та передбачення рейтингу ІСО проєктів".

ВИСНОВКИ

В роботі розроблено інформаційну технологію аналізу та передбачення рейтингу ICO проєктів.

Проведено аналіз предметної області, досліджено переваги та обмеження ICO проєктів на ринку криптовалют, досліджено фактори що впливають на рейтинг ICO проєктів. Також проведено огляд існуючих методів рейтингування проєктів ICO та відзначено підходи до оцінки їхньої якості та перспектив, що впливають на рейтинг ICO проєктів.

Проведено розвідувальний аналіз даних для виявлення особливостей факторів датасету та встановлення взаємозв'язків між ними. Описано алгоритми рейтингування, які використовуються для оцінки та ранжування проєктів ICO. Задачу рейтингування формалізовано як задачу класифікацій, яку запропоновано розв'язати такими методами машинного навчання як: метод опорних векторів (SVM), метод k-nearest neighbor (k-NN) і модель "дерево рішень". Кожен з цих методів має свої особливості та переваги. Аналіз запропонованих методів показав, що SVM може бути витратним з точки зору обчислювальних ресурсів, k-NN чутливий до викидів та шуму, а моделі "дерево рішень" можуть бути схильними до перенавчання.

Розроблено інформаційну технологію для передбачення рейтингу ICO проєктів за допомогою таких методів машинного навчання як: методи опорних векторів, метод k-nearest neighbor та модель "дерево рішень".

Дослідження показали, що точність передбачення методом опорних векторів на навчальному дорівнює 75% та тестовому наборі даних дорівнює 73%. Це свідчить про те, що алгоритм добре справляється з класифікацією даних і правильно класифікує 73% з них.

Точність передбачення методом k-nearest neighbor на навчальному та тестовому наборі даних складає 80%. Це свідчить про гарну ефективність алгоритму в класифікації та правильну класифікацію 74% даних.

Модель "дерево рішень" показав найвищу точність – 86% на навчальному, на тестовому наборі даних він показав 82%. Це говорить про те, що алгоритм добре справляється з класифікацією даних і правильно класифікує всі з них.

Додатково, побудовано матриці помилок для методів опорних векторів k-nearest neighbor, які вказують на ефективність цих алгоритмів в роботі з реальними даними. Також побудовано графік важливості ознак для моделі "дерево рішень", що дозволяє виділити найбільш значущі фактори в класифікації.

У розділі економічної оцінки проведено глибокий аналіз доцільності науково-дослідної роботи, За результатами проведених досліджень визначено, що рівень комерційного потенціалу даної розробки за темою «Інформаційна технологія аналізу та передбаченню рейтингу ІСО проєктів» складає 39,3 бали, перевищуючи середній рівень. Це свідчить про високий ступінь її комерційної значущості та перспектив для впровадження.

При оцінці технічних параметрів за допомогою узагальненого коефіцієнта якості розробки встановлено, що науково-технічна розробка перевершує існуючі аналоги у 3,58 рази. Цей аспект підкреслює високу якість та конкурентоспроможність розробки в порівнянні з іншими продуктами на ринку.

Крім того, термін окупності розробки складає 0,2, що менше трьох років. Це говорить про її високу комерційну привабливість і може бути стимулом для потенційного інвестора вкласти кошти у впровадження цієї розробки та введення її на ринок.

Отже, завдання магістерської кваліфікаційної роботи було виконано в повному обсязі.

Отримані під час виконання магістерської роботи результати вказують на можливість передбачення рейтингу ІСО проєктів.

За даною темою дослідження зроблено доповідь на тему «Аналіз методів оцінювання рейтингу ІСО проєктів за допомогою машинного навчання» на ІІІ Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації з публікацією тез ВНТУ (м. Вінниця, 2023рр.).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Василюк Д. О. Козачко О.М. Козловський О.С. Аналіз методів оцінювання рейтингу ICO проектів за допомогою машинного навчання. ЛІІ Наукова-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації: зб. Матеріалів конференції. Вінниця, 2023. URL: <https://press.vntu.edu.ua/index.php/vntu/catalog/view/788/1373/2632-1> (дата звернення: 01.10.2023).
2. What is an Initial Coin Offering (ICO)? URL: <https://corporatefinanceinstitute.com/resources/cryptocurrency/initial-coin-offering-ico/> (дата звернення: 23.09.2023).
3. Fisch, C., Masiak, C., Vismara, S., & Block, J. (2021). *Motives and profiles of ICO investors*. *Journal of Business Research*, 125, 564–576. URL : <https://doi.org/10.1016/j.jbusres.2019.07.036>.
4. Howell, S. T., Niessner, M., & Yermack, D. (2020). *Initial coin offerings: Financing growth with cryptocurrency token sales*. *The Review of Financial Studies*, 33(9), 3925–3974. URL: <https://doi.org/10.1093/rfs/hhz131>.
5. Fisher, G., Kuratko, D. F., Bloodgood, J. M., & Hornsby, J. S. (2017). *Legitimate to whom? The challenge of audience diversity and new venture legitimacy*. *Journal of Business Venturing*, 32(1), 52–71. URL: <https://doi.org/10.1016/j.jbusvent.2016.10.005> (дата звернення: 16.09.2023)..
6. Technology What Is an ICO? – URL: <https://www.coindesk.com/learn/what-is-an-ico/> (дата звернення: 18.09.2023).
7. Hornuf, L., Schilling, T., & Schwiendbacher, A. (2019). *Initial coin offerings, information disclosure, and fraud*, *CESifo*. Working paper no. 7962. URL: <https://link.springer.com/article/10.1007/s11187-021-00471-y>.
8. Олександр Мельник. 19.07.2018. Яким був ринок ICO у I півріччі 2018 року. URL: <https://nachasi.com/crypto/2018/07/19/rynok-ico-2018/> (дата звернення: 24.09.2023).

9. Huang, W., Meoli, M., & Vismara, S. (2020). *The geography of initial coin offerings*. *Small Business Economics*, 55(1), 77–102. URL: <https://link.springer.com/article/10.1007/s11187-019-00135-y>.
10. Супровід ICO проєктів у 2023 році. URL: <https://ybcase.com/ua/company-services/ico/soprovozdienie-ico-proektov-v-2023-godu> (дата звернення: 25.09.2023).
11. Huang, W., Vismara, S., & Wei, X. (2021). *Confidence and capital raising*. *Journal of Corporate Finance*. URL: <https://doi.org/10.1016/j.jcorpfin.2021.101900>.
12. Kastelein, R. (2017). What initial coin offerings are, and why VC firms care. URL: <https://hbr.org/2017/03/what-initial-coin-offerings-are-and-why-vc-firmscare>.
13. Hornuf, L., & Schwienbacher, A. (2018). *Market mechanisms and funding dynamics in equity crowdfunding*. *Journal of Corporate Finance*, 50, 556–574. URL: <https://doi.org/10.1016/j.jcorpfin.2017.08.009>.
14. Kher, R., Terjesen, S., & Liu, C. (2020). *Blockchain, Bitcoin, and ICOs: A review and research agenda*. *Small Business Economics*. URL: <https://doi.org/10.1007/s11187-019-00286-y>.
15. Mulders, M. (2018). 10 Keys for evaluating Initial Coin Offering (ICO) investments. URL: <https://cryptopotato.com/10-keys-evaluating-initial-coin-offeringico-investments>.
16. ForkLog UA 2023. 5 простих кроків для аналізу ICO. URL: <https://www.cryptohackers.club/2018/02/5-prostyh-kroktiv-dlya-analizu-ico.html> (дата звернення: 23.09.2023).
17. Frankenfield J. 2022. Initial Coin Offering (ICO): Coin Launch Defined, with Examples. URL: <https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp>.
18. Sallán Leyes, José María 2023.05.18 The role of sentiment analysis in forecasting successful ICOs. URL: <https://upcommons.upc.edu/handle/2117/388656>.
19. Evgeny Lyandres 2021. ICO Success and Post-ICO Performance. URL: https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=3287583.
20. Maksim Belitski & Dmitri Boreiko 2021 Success factors of initial coin offerings URL: <https://link.springer.com/article/10.1007/s10961-021-09894-x> (дата звернення: 28.09.2023)

21. Шеремет О. І. 2021 Метод опорних векторів. – URL: <https://www.dstu.dp.ua/Portal/Data/74/72/3st13-17.pdf> (дата звернення: 28.09.2023)
22. Метод к найближчих сусідів. URL: http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_03.pdf (дата звернення: 28.09.2023)
23. Метод к найближчого сусіда або системи міркувань на основі аналогічних випадків. URL: <https://studfile.net/preview/7818687/page:2/> (дата звернення: 28.09.2023)
24. Дерево рішень. URL: <https://ua5.org/algorithm/1976-derevo-rishen.html> (дата звернення: 28.09.2023)
25. Методи дерев рішень, класифікації та прогнозування. URL: https://moodle.znu.edu.ua/pluginfile.php?file=/486136/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%209.pdf (дата звернення: 28.09.2023)
26. Методи дерев рішень, класифікації та прогнозування. URL: https://moodle.znu.edu.ua/pluginfile.php?file=/486136/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%209.pdf (дата звернення: 28.09.2023)
27. Про затвердження Методики визначення вартості наукових робіт. URL: <https://zakon.rada.gov.ua/rada/show/v0545666-08#top> (дата звернення: 10.11.2023р.)
28. Середньозважена ціна акцептованої на РДН електроенергії (оптовий ринок). URL : <https://www.oree.com.ua/index.php/newsctr/n/6582> . (дата звернення: 15.11.2023 р)

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

____ д.т.н., проф. Віталій МОКІН

«__» _____ 2023 року

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РЕЙТИНГУ
ІСО ПРОЄКТІВ
08-34.МКР.002.00.000 ТЗ

Керівник: к.т.н., доц.

____ Олексій КОЗАЧКО

«__» _____ 2023 р.

Розробив студент гр. 2ІСТ-22м

____ Дмитро ВАСИЛЮК

«__» _____ 2023 р.

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2023 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2023 р.

2. Джерела розробки:

– Frankenfield J. 2022. Initial Coin Offering (ICO): Coin Launch Defined, with Examples. URL: <https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp>

– Sallán Leyes, José María 2023.05.18 The role of sentiment analysis in forecasting successful ICO. URL: <https://upcommons.upc.edu/handle/2117/388656>

– Evgeny Lyandres 2021. ICO Success and Post-ICO Performance. URL: https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=3287583

– Maksim Belitski & Dmitri Boreiko 2021 Success factors of initial coin offerings. URL: <https://link.springer.com/article/10.1007/s10961-021-09894-x>

3. Метою дослідження є підвищення точності передбачення рейтингу ICO проєктів за рахунок використання методів машинного навчання.

4. Вихідні дані для проведення робіт:

Набір даних, зібраний з сервісу ICO Drops, який містить інформацію про ціну та інші ознаки про ICO.

5. Методи дослідження:

Використання моделей машинного навчання, а саме метод опорних векторів, метод k-nearest neighbor, модель "дерево рішень";

6. Етапи роботи і терміни їх виконання:

- | | |
|---|---------------|
| 1. Загальна характеристика та аналіз ICO | _____ – _____ |
| 2. Вибір інформаційних технологій рейтингування ICO проєктів для розв'язання задачі | _____ – _____ |
| 3. Розроблення інформаційної технології аналізу та передбачення рейтингу ICO проєктів | _____ – _____ |
| 4. Економічна частина | _____ – _____ |
| 5. Оформлення пояснювальної записки | _____ – _____ |
| 7. Очікувані результати та порядок реалізації: | |

Інформаційна технологія аналізу та передбачення рейтингу ICO проєктів, Результатом якої є підвищення точності передбачення рейтингу ICO проєктів за рахунок використання методів машинного навчання.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»).

9. Порядок приймання роботиПублічний захист « » _____ 2023 р.Початок розробки « » _____ 2023 р.Граничні терміни виконання МКР « » _____ 2023 р.

Розробив студент групи 2ІСТ–22м _____ Дмитро ВАСИЛЮК

Додаток Б

Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія аналізу та передбачення рейтингу ІСО проєктів»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра САІТ

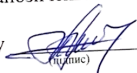
Показники звіту подібності Unicheck

Оригінальність 92.41% Схожість 7.59%

Аналіз звіту подібності (відмітити потрібне)

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

Особа, відповідальна за перевірку



(підпис)

Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



(підпис)

Дмитро ВАСИЛЮК

Керівник роботи



(підпис)

Олексій КОЗАЧКО

Додаток В

Лістинг програми

```

with open("/kaggle/input/ico-data/dataset3.csv", 'rb') as file:
    result = chardet.detect(file.read())
    encoding = result['encoding']
data = pd.read_csv("/kaggle/input/ico-data/dataset3.csv", encoding=encoding, sep=';')
data.info()
data['rating'] = data['rating'].str.strip()
data = data.drop('Unnamed: 0', axis=1, errors='ignore')
pp.ProfileReport(data)
categorical_columns = []
numerics = ['int8', 'int16', 'int32', 'int64', 'float16', 'float32', 'float64']
features = data.columns.values.tolist()
for col in features:
    if data[col].dtype in numerics: continue
    categorical_columns.append(col)
categorical_columns
for col in categorical_columns:
    if col in data.columns:
        le = LabelEncoder()
        le.fit(list(data[col].astype(str).values))
        data[col] = le.transform(list(data[col].astype(str).values))
cv_number = 3
train = data.drop('rating', axis=1)
target = data['rating']
X_train, X_test, y_train, y_test = train_test_split(train, target, test_size=0.2,
random_state=42)
svm_classifier = SVC(random_state=42)
svm_classifier.fit(X_train, y_train)
y_train_pred = svm_classifier.predict(X_train)
y_test_pred = svm_classifier.predict(X_test)
accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)
print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")
cm_train = confusion_matrix(y_train, y_train_pred)
cm_test = confusion_matrix(y_test, y_test_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_train, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на навчальній вибірці')
plt.xlabel('Передбачені значення')

```

```

plt.ylabel('Справжні значення')
plt.show()
plt.figure(figsize=(8, 6))
sns.heatmap(cm_test, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на тестовій вибірці')
plt.xlabel('Передбачені значення')
plt.ylabel('Справжні значення')
plt.show()
random_forest_classifier = RandomForestClassifier(random_state=42,
n_estimators=1000, max_depth=8)
random_forest_classifier.fit(X_train, y_train)
y_train_pred = random_forest_classifier.predict(X_train)
y_test_pred = random_forest_classifier.predict(X_test)
accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)
print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")
cm_train = confusion_matrix(y_train, y_train_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_train, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на навчальній вибірці')
plt.xlabel('Передбачені значення')
plt.ylabel('Справжні значення')
plt.show()
cm_test = confusion_matrix(y_test, y_test_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_test, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на тестовій вибірці')
plt.xlabel('Передбачені значення')
plt.ylabel('Справжні значення')
plt.show()
plt.figure(figsize=(10, 6))
feature_importance = random_forest_classifier.feature_importances_
feature_names = X_train.columns
sns.barplot(x=feature_importance, y=feature_names, orient="h")
plt.title('Важливість ознак при побудові дерева рішень (навчальна вибірка)')
plt.show()
plt.figure(figsize=(10, 6))
feature_importance_test = random_forest_classifier.feature_importances_
sns.barplot(x=feature_importance_test, y=feature_names, orient="h")
plt.title('Важливість ознак при побудові дерева рішень (тестова вибірка)')

```



```

plt.show()
knn_pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('knn_classifier', KNeighborsClassifier(n_neighbors=3,p=3))
])
knn_pipeline.fit(X_train, y_train)
y_train_pred = knn_pipeline.predict(X_train)
y_test_pred = knn_pipeline.predict(X_test)
accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)
print("Точність на навчальних даних:", round(accuracy_train * 100, 2), "%")
print("Точність на тестових даних:", round(accuracy_test * 100, 2), "%")
cm_train = confusion_matrix(y_train, y_train_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_train, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на навчальній вибірці')
plt.xlabel('Передбачені значення')
plt.ylabel('Справжні значення')
plt.show()
cm_test = confusion_matrix(y_test, y_test_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_test, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1, 2],
yticklabels=[0, 1, 2])
plt.title('Матриця помилок на тестовій вибірці')
plt.xlabel('Передбачені значення')
plt.ylabel('Справжні значення')
plt.show()

```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ
РЕЙТИНГУ ІСО ПРОЄКТІВ**

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2023 р.

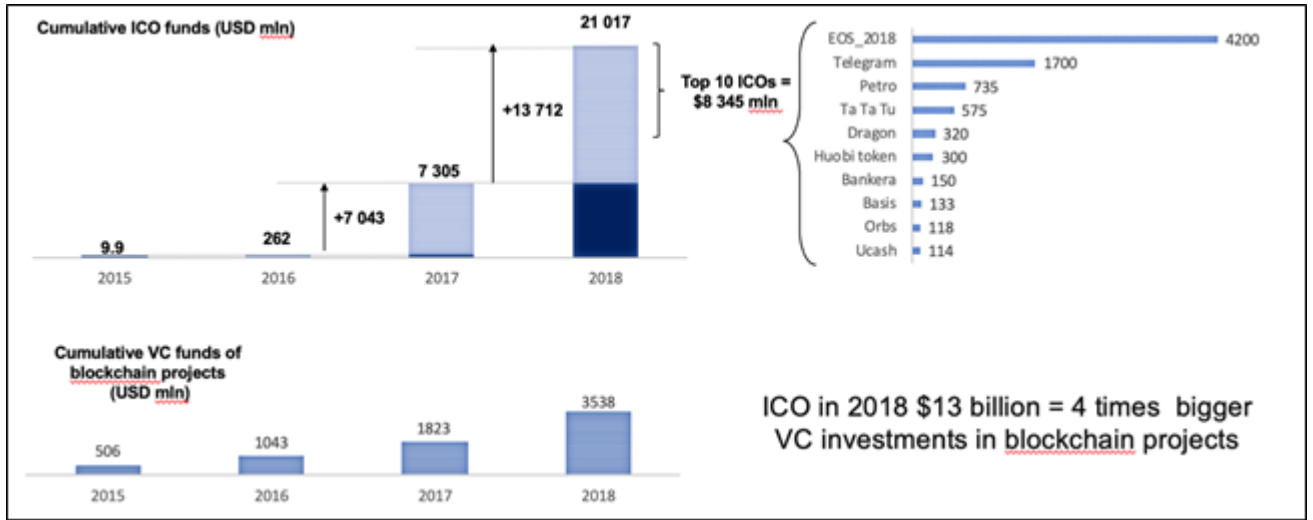


Рисунок Г.1 – Кількість ICO у відповідних роках

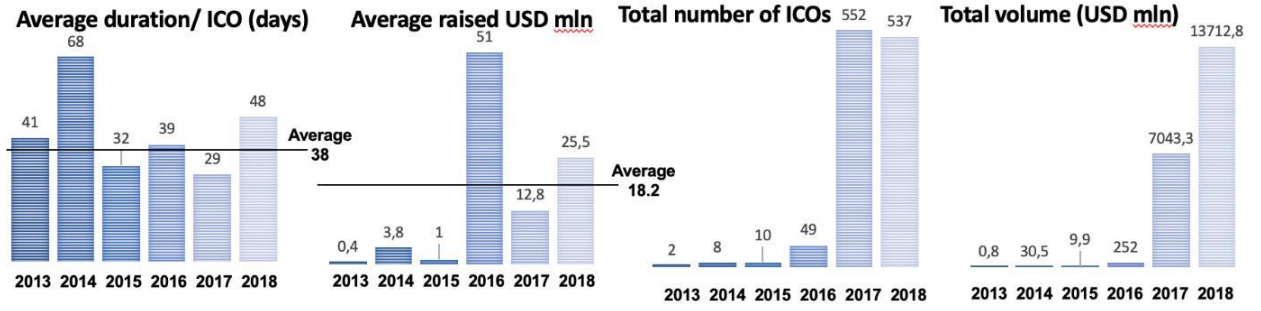


Рисунок Г.2 – Загальний обсяг ІСО в відповідних роках

name	role_of_category	goal	fundraising_goal	total_tokens	token_price	received_money	min_persons	max_persons	token_type	available_for	accepts_1	accepts_2	accepts_3	accepts_4	accepts_5	whitelist	rating
1 Ozone Metaverse	1 Platform	1430000	280000	2000000000	0.007	1430000	100	500	Flow	21%	USDT	0	0	0	0	0	low
2 TypeIT	1 Blockchain S	3900000	2200000	1000000000	0.05	3120000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
3 EGO (Paysenger)	1 Social Networ	4160000	859000	500000000	0.06	4160000	100	2500	BEP-20	26.6%	ETH	USDT	USDC	DOT	0	0	low
4 Goracle	1 Blockchain S	3730000	7070000	522000000	0.732	3725000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
5 BIDS SHOP	1 DeFi	17080000	8090000	20320000	0.515	2500000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
6 Tectum	1 Blockchain	18700000	8120000	22140000	0.99	4202573	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
7 Ethlas	1 Gaming	2760000	6640000	11320000	0.041	8700000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
8 Archway	1 Blockchain	27000000	6000000	1000000000	0.2	27000000	100	1000	Cosmos N	15%	USDT	USDC	0	0	0	0	low
9 PolyGame	1 Platform	5230000	100000	5000000000	0.075	5225000	100	500	Polygon	6%	USDC	BUSD	0	0	0	0	low
10 Neon	2 Blockchain S	45000000	5000000	1000000000	0.1	45000000	100	500	ERC-20	5%	USDT	USDC	0	0	0	0	high
11 War Legends	1 Gaming	16130000	250000	350000000	0.037	12904000	100	500	Polygon	21%	USDT	0	0	0	0	0	low
12 OtterHome	1 DeFi	18100000	8220000	28720000	0.946	14480000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
13 HyperGPT	1 Marketplace	1670000	7990000	26800000	0.089	1665000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
14 Cyber Arena	1 Gaming	2100000	2110000	11800000	0.132	2100000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
15 KAIF	1 Platform	2120000	500000	90000000	0.15	500000	100	500	BEP-20	10%	BUSD	0	0	0	0	0	low
16 GPT Guru	1 Artificial Intelli	1450000	6260000	16100000	0.182	1448000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
17 Metaverse Kombar	1 Gaming	18020000	1200000	300000000	0.08	2200000	100	750	BEP-20	11%	USDT	0	0	0	0	0	low
18 Cogito Protocol	1 DeFi	11540000	4760000	39600000	0.874	9232000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
19 The Qwan	1 Gaming	18640000	270000	8300000	0.865	500000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
20 Eldarune	1 Gaming	1750000	5720000	27620000	0.288	1750500	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
21 Bware Labs	1 Blockchain PI	1480000	1780000	100000000	0.5	7200000	100	500	ERC-20	25%	USDT	0	0	0	0	0	low
22 Orbofi	1 Platform	2600000	880000	27410000	0.905	2600000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
23 Pika Protocol	1 Exchange	10480000	8230000	30120000	0.632	8700000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
24 Wistaverse	1 Platform	930000	2370000	6480000	0.478	930000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
25 CyberConnect	1 Protocol	3550000	3000000	100000000	1.8	30400000	100	500	ERC-20	28%	USDT	0	0	0	0	0	high
26 BladeDAO	1 Gaming	2860000	5960000	10000000	0.001	2288000	100	500	Ovm block	10%	USDT	0	0	0	0	0	low
27 Tenet	1 Blockchain	2400000	1560000	1200000000	0.02	2400000	100	500	Ovm block	10%	USDT	0	0	0	0	0	low
28 The Unfettered	1 Gaming	13020000	5250000	29510000	0.112	2500000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low
29 Turbos Finance	1 DEX	3750000	750000	1000000000	0.005	3750000	100	500	Ovm block	15%	USDT	0	0	0	0	0	low
30 WeFi	1 Lending	3210000	8410000	29100000	0.055	1125000	100	500	ERC-20	10%	USDT	0	0	0	0	0	low

Рисунок Г.3 – Зовнішній вигляд датасету

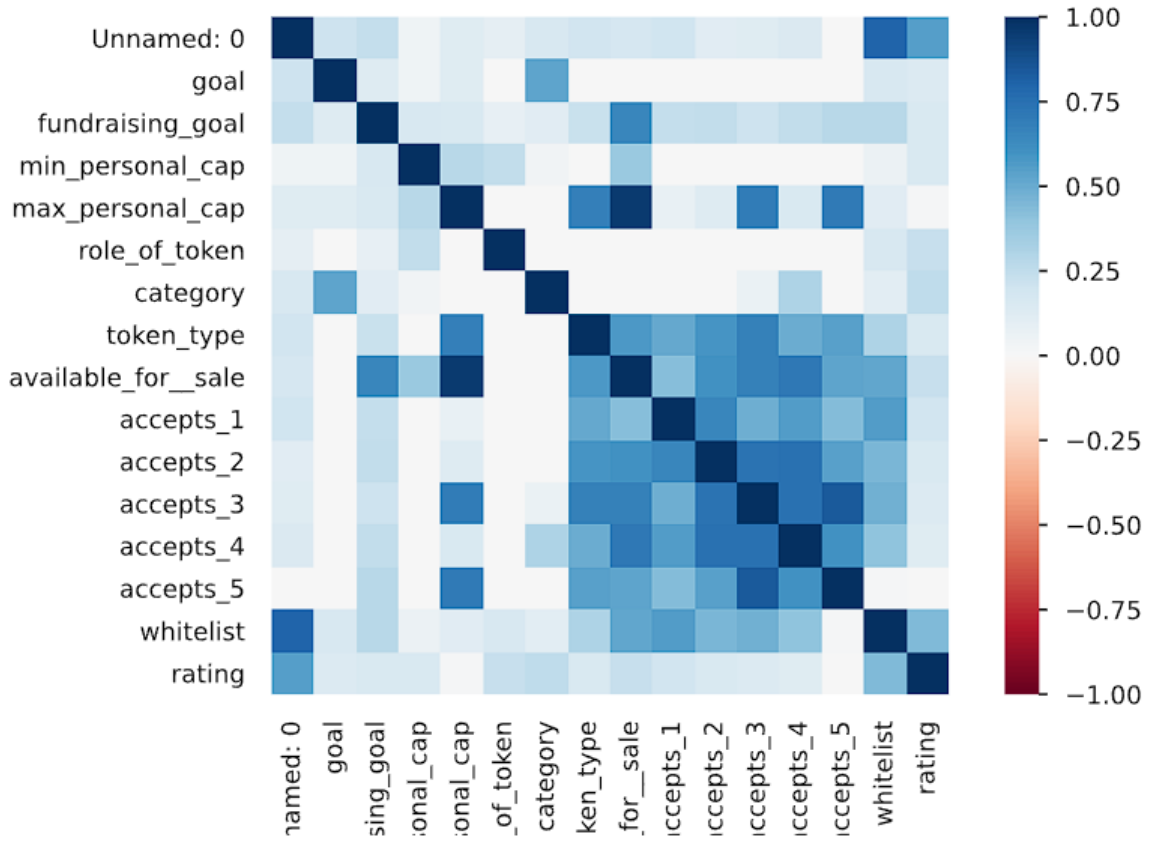
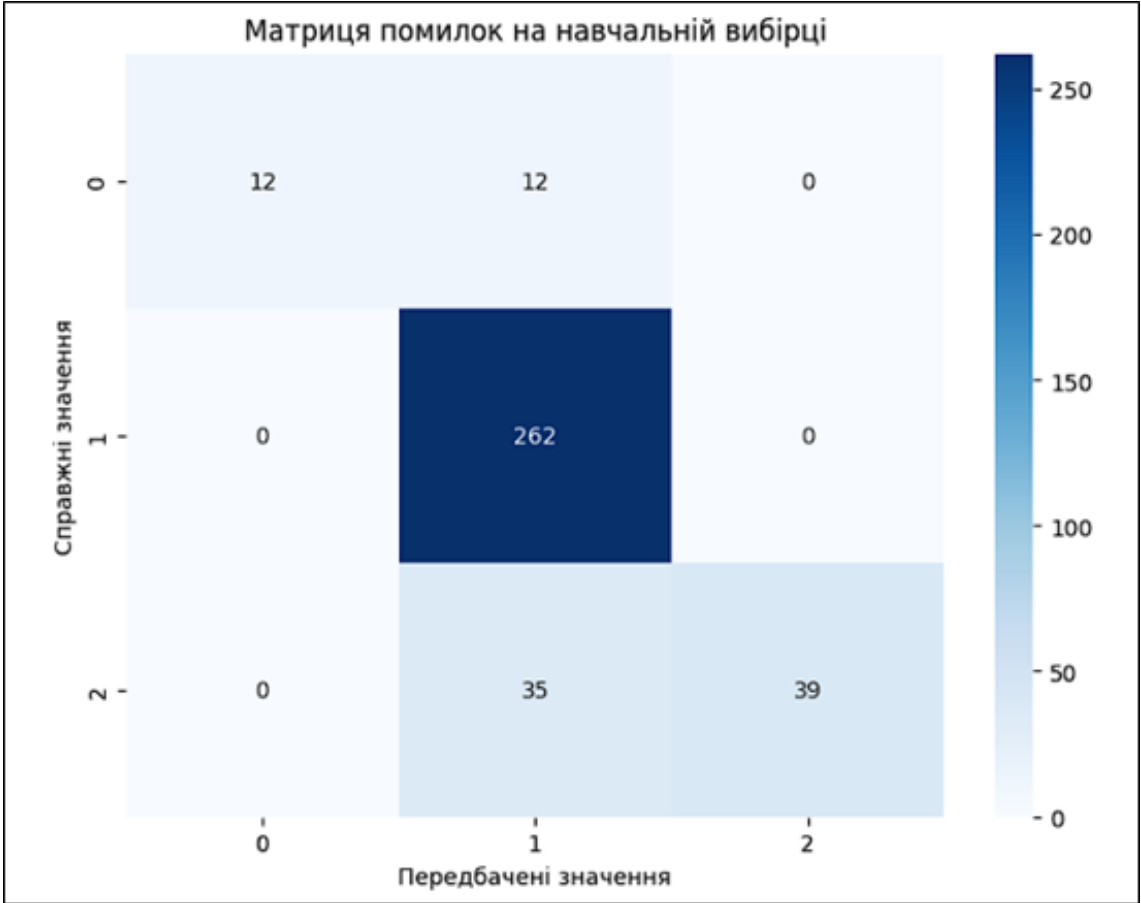
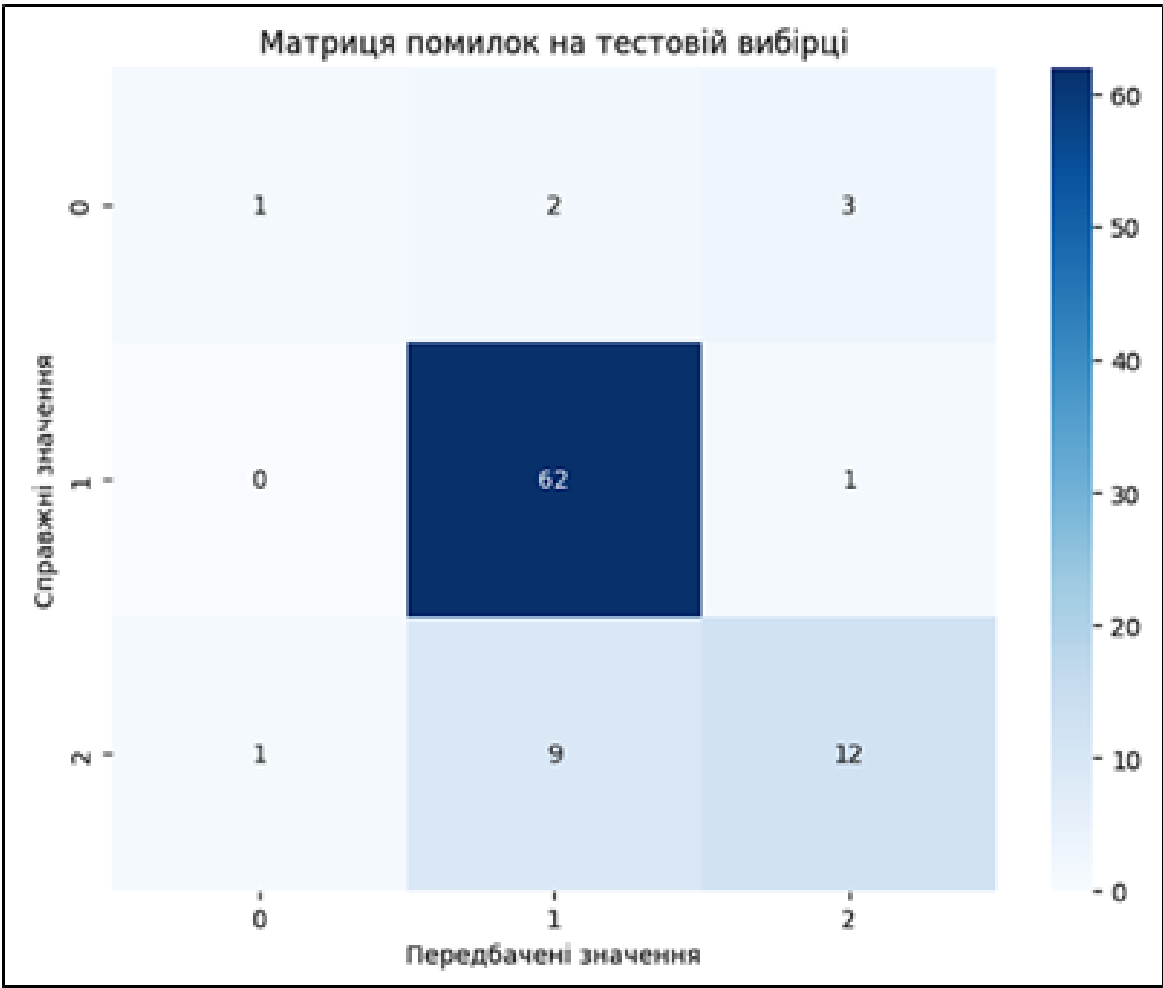


Рисунок Г.4 – Кореляція даних зібраних в датасеті



Додаток Г.5 – Матриця помилок для оптимальної моделі для навчальної вибірки



Додаток Г.6 – Матриця помилок для оптимальної моделі для тестової вибірки

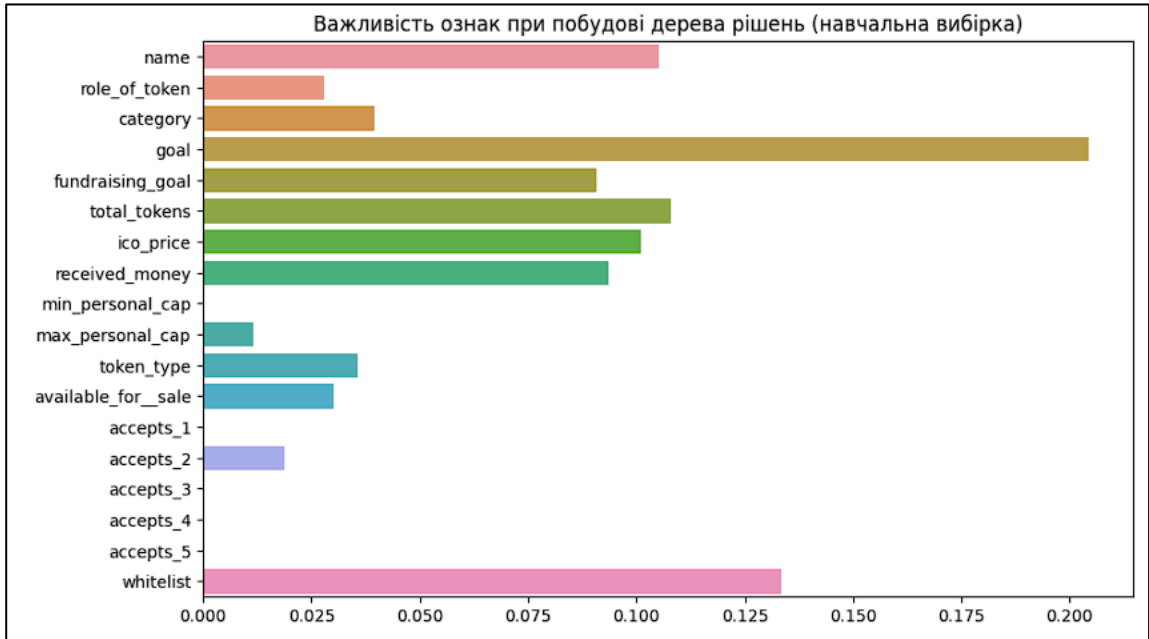


Рисунок Г.7 – Графік важливості ознак для оптимальної моделі для навчальної вибірки

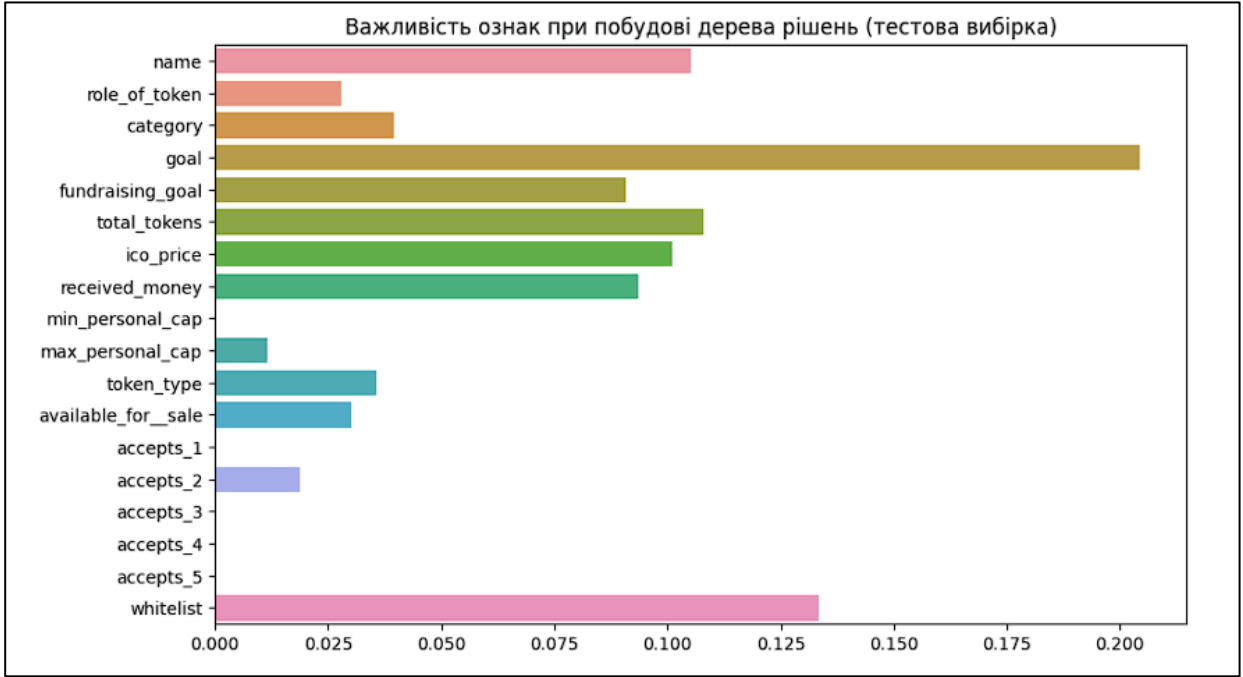


Рисунок Г.8 – Графік важливості ознак для оптимальної моделі для тестової вибірки