

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Факультет інформаційних технологій  
(факультет)

Кафедра системного аналізу та управління  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
кваліфікаційної роботи ступеня бакалавра

Студента \_\_\_\_\_ Ширяєва Артема Олександровича

академічної групи \_\_\_\_\_ 124-20-2

спеціальності \_\_\_\_\_ 124 Системний аналіз

на тему: «Системний аналіз та оптимізація фінансової діяльності за допомогою систем штучного інтелекту на прикладі підприємства ROZETKA»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	Інституційною	
кваліфікаційної роботи	<i>к.т.н., доц. Алексєєв О.М.</i>			
розділів:	3			
Інформаційно-аналітичний	<i>к.т.н., доц. Алексєєв О.М.</i>			
Спеціальний розділ	<i>к.т.н., доц. Алексєєв О.М.</i>			
Рецензент				
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро  
2024

ЗАТВЕРДЖЕНО:

завідувач кафедри  
Системного аналізу та управління  
(повна назва)

\_\_\_\_\_ к.т.н., доц. Желдак Т.А.  
(підпис) (прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня бакалавра**

студенту Ширяєву А. О. академічної групи 124-20-2  
спеціальності: 124 Системний аналіз  
на тему «Системний аналіз та оптимізація фінансової діяльності за допомогою систем штучного інтелекту на прикладі підприємства ROZETKA»

затверджену наказом ректора НТУ «Дніпровська політехніка»  
від 23.05.2024 р. №469-с

Розділ	Зміст	Терміни виконання
1. Теоретичний розділ	Дослідження існуючих джерел щодо обраної теми.	15.02.2024 – 29.02.2024
2. Проєктний розділ	Аналіз та формалізація задачі. Порівняння існуючих програмних реалізацій. Порівняння існуючих мов програмування та IDE для вибору найоптимальнішої.	08.03.2024 – 17.03.2024
3. Програмний розділ	Вибір найоптимальнішої системи штучного інтелекту. Розробка моделі та програмної реалізації завдання.	20.03.2024 – 13.04.2024

Завдання видано \_\_\_\_\_ доц. Алексєєв О.М.  
(підпис) (прізвище, ініціали)

Дата видачі: 06.09.2024 р.

Дата подання до екзаменаційної комісії: \_\_\_\_\_

Прийнято до виконання \_\_\_\_\_  
(підпис студента)

Ширяєв А. О.  
(прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 82 с., 27 рис., 7 табл., 4 додатки, 20 джерел.

*Об'єктом дослідження* в роботі є процес розробки методології та програми, що оптимізує та автоматизує пошук критичних параметрів в фінансовій діяльності підприємства ROZETKA.

*Предметом дослідження* є методологія, що виявляє критичні параметри в фінансовій діяльності підприємства та визначає необхідні покращення для зменшення ризику банкрутства.

*Метою* даної кваліфікаційної роботи є розробка методології, що оптимізує та автоматизує пошук слабких місць в фінансовій діяльності підприємства ROZETKA та написання для неї розрахункової програми.

*Методи дослідження:* методи системного аналізу – для розглядів фінансової діяльності підприємства, методи машинного навчання – для покращення процесів управління фінансами, методи прогнозування – для передбачення фінансових результатів підприємства.

*У теоретичному розділі* наведено огляд створення алгоритмів машинного навчання та систем штучного інтелекту.

*У проєктному розділі* були визначені рішення щодо оптимізації фінансової діяльності підприємства та вибране середовище реалізації.

*У програмному розділі* були обрані системи штучного інтелекту, створено моделі для розв'язку задачі та написаний програмний код.

Практична цінність отриманих результатів полягає в тому, що розроблена методологія оптимізує час на знаходження критичних параметрів та автоматизує розрахунки за допомогою програмної реалізації.

Ключові слова: Системний аналіз, Машинне навчання, Системи штучного інтелекту, Фінансова діяльність, Ймовірність банкрутства, Критичні параметри.

## ABSTRACT

Cover letter: 82 pages, 27 figures, 7 tables, 4 appendices, 20 sources.

*The object of the research* in the paper is the process of developing a methodology and program that optimizes and automates the search for critical parameters in the financial activities of the ROZETKA enterprise.

*The subject of the study* is a methodology that identifies critical parameters in the financial activities of the enterprise and determines necessary improvements to reduce the risk of bankruptcy.

*The aim* of this qualification work is to develop a methodology that optimizes and automates the search for weaknesses in the financial activities of the company ROZETKA and to write a computational program for it.

*Research methods:* methods of system analysis – for examining the financial activities of the enterprise, machine learning methods – for improving financial management processes, forecasting methods – for predicting the financial results of the enterprise.

*The theoretical section* provides an overview of creating machine learning algorithms and artificial intelligence systems.

*In the project section*, decisions were made regarding the optimization of the enterprise's financial activities and the choice of implementation environment.

*In the software section*, artificial intelligence systems were selected, formal models for problem solving were created, and the software code was written.

The practical value of the obtained results lies in the fact that the developed methodology optimizes the time for identifying critical parameters and automates calculations through software implementation.

Keywords: System Analysis, Machine Learning, Artificial Intelligence Systems, Financial Activity, Probability of Bankruptcy, Critical Parameters.

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТЕОРЕТИЧНИХ ВІДОМОСТЕЙ ПРО МАШИННЕ НАВЧАННЯ ТА ШТУЧНИЙ ІНТЕЛЕКТ .....	8
1.1 Теоретичні дані про машинне навчання .....	8
1.2 Принципи роботи машинного навчання.....	11
1.3 Різновиди машинного навчання .....	12
1.4 Типи машинного навчання.....	16
1.5 Машинне навчання та статистика .....	17
1.6 Регресія у машинному навчанні .....	19
1.7 Поширені алгоритми в машинного навчання .....	22
РОЗДІЛ 2 ПРОЄКТНІ РІШЕННЯ ЩОДО ОПТИМІЗАЦІЇ ФІНАНСОВОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ НА ПРИКЛАДІ ТОВ «ROZETKA».....	35
2.1 Побудова Z-моделей визначення ймовірності настання банкрутства ТОВ «ROZETKA» та визначення на основі них проблемних зон управління фінансовою діяльністю та їх оптимізація.....	35
2.2 Мотивація та аналіз задачі .....	38
2.3 Формалізація задачі (виділення вхідних-вихідних параметрів, зав'язків між ними тощо) .....	39
2.4 Вибір технології штучного інтелекту – обґрунтування вибору, огляд можливих варіантів, що розглядались .....	41
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	46
3.1 Обґрунтування вибору середовища реалізації (Python).....	46
3.2 Вибір середовища розробки(IDE) .....	48
3.3 Вибір додаткових бібліотек для розробки.....	50
3.4 Розробка моделі – створення формальної моделі для розв'язку задачі на основі обраної технології .....	52
3.5 Практична реалізація – короткий огляд програмної реалізації моделі в середовищі Python.....	53
3.6 Опис основних функцій програмної реалізації.....	54
ВИСНОВОК.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	72

ДОДАТКИ.....	6
Додаток А. Відомість матеріалів кваліфікаційної роботи .....	74
Додаток Б. Відгук керівника кваліфікаційної роботи .....	75
Додаток В. Рецензія .....	76
Додаток Г. Фінансові дані підприємства ROZETKA .....	77

## ВСТУП

У сучасних умовах економічного розвитку питання фінансової стабільності та здатності підприємств до виживання стають все більш актуальними. Особливо це стосується компаній, що працюють у секторі роздрібної торгівлі, де конкуренція надзвичайно висока, а вимоги споживачів постійно зростають.

Одним із ключових інструментів для оцінки фінансової стійкості підприємства є побудова Z-моделей та визначення ймовірності настання банкрутства. На прикладі ТОВ «ROZETKA» можна провести аналіз та виявити проблемні зони в управлінні фінансовою діяльністю, а також розробити стратегії для їх оптимізації.

У даному дослідженні ми розглянемо процес побудови Z-моделей для визначення ймовірності банкрутства ТОВ «ROZETKA» та проаналізуємо отримані результати. Важливим етапом буде визначення проблемних зон у фінансовому управлінні компанії та розробка пропозицій щодо їх оптимізації. Дослідження виявиться корисним для розуміння та управління фінансовими ризиками в умовах сучасного бізнесу, а також для підвищення ефективності стратегічного планування та управління підприємством.

# РОЗДІЛ 1 АНАЛІЗ ТЕОРЕТИЧНИХ ВІДОМОСТЕЙ ПРО МАШИННЕ НАВЧАННЯ ТА ШТУЧНИЙ ІНТЕЛЕКТ

## 1.1 Теоретичні дані про машинне навчання

Машинне навчання — це розділ штучного інтелекту, який дозволяє комп'ютерним системам навчатися на основі даних і покращувати продуктивність без явного програмування (без написання інструкцій для кожного завдання). Воно працює на основі алгоритмів, які аналізують великі обсяги інформації, шукають у них закономірності та використовують ці знання для прийняття рішень і прогнозів.

Теорія машинного навчання включає завдання передбачення майбутньої поведінки складних систем, навіть за відсутності точних моделей, що описують поведінку подібних систем. Машинне навчання є важливим інструментом для багатьох організацій, який допомагає їм оптимізувати процеси, знаходити нові джерела доходу та вирішувати складні проблеми. Використання даних стало основою для прийняття бізнес-рішень, особливо коли дані надходять з відгуків клієнтів, співробітників, а також з фінансових і виробничих показників.

Алгоритми машинного навчання дозволяють організаціям автоматично аналізувати великі обсяги даних і значно підвищити ефективність і швидкість обробки інформації. Це дозволяє компаніям краще розуміти ринкові тенденції, передбачати попит і вдосконалювати стратегії на основі мінливих умов. Системи машинного навчання здатні вчитися на минулих даних і постійно вдосконалювати прогнози, дозволяючи швидко реагувати на нові виклики та можливості, забезпечуючи конкурентну перевагу.



В даний час машинне навчання застосовується в багатьох сферах завдяки своїй унікальності і високій ефективності при вирішенні різноманітних завдань. Це особливо важливо в таких сферах:

### 1. Виробництві

Машинне навчання також використовується в виробництві для автоматизації контролю якості (QA). Наприклад, General Electric використовує алгоритми машинного навчання для виявлення помилок у виробничому процесі, що дозволяє їм швидко виявляти потенційні проблеми та уникати виробничих збоїв. Це підвищує надійність продукції та зменшує витрати на брак. Крім того, машинне навчання полегшує вдосконалення систем управління якістю, аналізуючи великі обсяги даних про якість продукції та виявляючи закономірності, невидимі для людського ока. Це не тільки дозволяє виробникам знизити витрати на виробництво, але й підвищує задоволеність клієнтів, забезпечуючи високу якість продукції.

### 2. Охороні здоров'я та у медико-біологічних розробках

Машинне навчання відіграє важливу роль у біомедичній інженерії та охороні здоров'я, сприяючи значних покращень у діагностиці, лікуванні та управлінні даними. Наприклад, Google DeepMind використовує методи машинного навчання для аналізу медичних зображень і прогнозування спалахів захворювань. Система компанії автоматично визначає патологію на рентгенівських знімках, що дозволяє швидше та точніше поставити діагноз. Крім того, машинне навчання використовується для прогнозування потенційних побічних ефектів і покращення синтезу молекул під час дослідження нових ліків. Такі підходи можуть прискорити процес відкриття нових ліків і знизити витрати, що важливо в галузі медичних інновацій.

### 3. Фінансових послугах

Машинне навчання також відіграє важливу роль у фінансових послугах, допомагаючи удосконалити прогнозування ринку, ризик-менеджмент та підтримку прийняття рішень. Capital One, наприклад, використовує методи машинного навчання для аналізу даних клієнтів і підтримки в управлінні фінансами. Система може автоматично оцінювати кредитний ризик і приймати рішення щодо видання кредитів на основі великої кількості даних про клієнтів. Такі технології зменшують витрати на ризик-менеджмент та підвищують точність прогнозів у фінансовому секторі. Крім того, використовуючи машинне навчання для автоматизації торгових алгоритмів, фонди та брокерські фірми можуть швидше реагувати на зміни на фінансових ринках і оптимізувати торгові стратегії. Ці інновації підвищують ефективність управління активами та ризиками, забезпечуючи конкурентну перевагу в умовах постійної динаміки на фінансових ринках.

#### 4. Роздрібній торгівлі

Машинне навчання широко використовується в галузі роздрібною торгівлі, забезпечуючи підтримку в управлінні запасами, підрахунку людей і прогнозуванні попиту. Наприклад, Walmart використовує алгоритми машинного навчання для оптимізації запасів і прогнозування попиту на продукцію в різних магазинах. Система компанії аналізує великі обсяги даних про покупки, погоду та інші фактори, щоб точно передбачити, які товари та у якій кількості потрібні в певних місцях і в певний час. Це дозволяє Walmart зменшити витрати на запаси та підвищити рівень задоволеності клієнтів, гарантуючи забезпечення наявності необхідним їм продуктам. Крім того, машинне навчання використовується, щоб персоналізувати рекомендації для клієнтів на основі їхніх покупок та інших даних, допомагаючи збільшити продажі та залучити нових клієнтів. Такі інновації дозволяють роздрібним торговцям стати більш ефективними та конкурентоспроможними в сучасних ринкових умовах.

#### 5. Мультимедіа та розвагах

Машинне навчання використовується в різноманітних мультимедійних і розважальних програмах, від персоналізації контенту до керування великими обсягами даних. Наприклад, Netflix використовує алгоритми машинного навчання, щоб рекомендувати користувачам фільми та серіали. Системи компанії аналізують дані про перегляди, попередні вподобання та інші патерни поведінки, щоб адаптувати рекомендації та підвищити рівень задоволеності аудиторії. Крім того, машинне навчання використовується для автоматизації процесу створення та редагування вмісту, наприклад, в іграх та відео. Наприклад, NVIDIA впроваджує алгоритми машинного навчання для автоматичної генерації реалістичної 3D-зображень та анімації, що значно прискорює розробку та знижує вартість створення цифрового контенту. Ця інновація покращує якість і доступність мультимедійних продуктів, забезпечує нові переживання для користувачів і дозволяє компаніям підвищити ефективність виробництва та розповсюдження контенту [1].

## **1.2 Принципи роботи машинного навчання**

Основна концепція сучасного машинного навчання полягає в навчанні алгоритмів на основі даних без використання чітко прописаної програми. Замість того, щоб вручну давати вказівки алгоритму, ви забезпечуємо його необхідною кількістю даних із прикладами того, чого він має навчитися. Це досягається за допомогою математичного зв'язку, який існує між будь-якою комбінацією всіх вхідних та вихідних даних. Це означає, що всі алгоритми машинного навчання базуються на адаптивній математичній функції, наприклад:

1. Під час навчання алгоритму надається набір даних, що містить пари "вхід/вихід", наприклад: (2,10), (5,19) та (9,31).

2. Алгоритм самостійно шукає математичну залежність між вхідними та вихідними даними. Наприклад, він може встановити, що  $o = 3 \times i + 4$ .

3. Потім алгоритму дають нове значення вхідних даних, наприклад 7, і просять передбачити відповідний вихід. Завдяки знайденій раніше залежності, алгоритм автоматично розраховує вихід – 25.

Це базове пояснення, але воно ілюструє суть машинного навчання, яка полягає в тому, що система навчається самостійно без чіткого плану. Такий підхід робить машинне навчання надзвичайно потужним інструментом для вирішення складних задач, які важко або неможливо запрограмувати вручну.

Однак важливо пам'ятати, що машинне навчання не позбавлене недоліків. Алгоритми можуть стати упередженими або неточними, якщо їх навчати неякісними даними або недостатньою їх кількістю. Тому необхідно ретельно відбирати дані та оцінювати роботу алгоритмів машинного навчання для отримання достовірних результатів [2].

### **1.3 Різновиди машинного навчання**

У машинному навчанні існує широкий спектр різних методів, кожен з яких має унікальні характеристики та використовується для досягнення певних цілей. Ось деякі з найпоширеніших:

#### **1. Контрольоване Навчання**

Цей метод навчання застосовується у випадках, коли є великі обсяги даних, припустимо — тисячі фотографій домашніх тварин з маркерами (мітками, ярликами): це кішка, а це собака. Необхідно створити алгоритм, за допомогою якого машина могла б по фотографії, яку «не бачила» раніше, визначити, хто на ній зображений: кішка або собака. У ролі «вчителя» в даному випадку виступає людина, яка заздалегідь поставила

маркери. Машина сама вибирає ознаки, за якими вона відрізняє кішок від собак. Тому в подальшому знайдений нею алгоритм може бути швидко переналаштований на рішення іншої задачі, наприклад, на розпізнавання курей і качок. Машина знову-таки сама виконає складну і копітку роботу по виділенню ознак, за якими буде розрізняти цих птахів. А неймережа, яку навчили розпізнавати кішок, можна швидко навчити обробляти результати комп'ютерної томографії [3].

## 2. Неконтрольоване Навчання

Хоча маркованих, розмічених даних накопичилося вже досить багато, даних без маркерів (міток) все ж набагато більше. Це зображення без підписів, аудіозаписи без коментарів, тексти без анотацій. Завдання машини при неконтрольованому навчанні — знайти зв'язку між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних. Неконтрольоване навчання використовується, наприклад, в рекомендаційних системах, коли в інтернет-магазині на основі аналізу попередніх покупок покупцеві пропонуються товари, які можуть зацікавити його з більшою ймовірністю, ніж інші. Або коли на після перегляду якогось відеокліпу на порталі YouTube відвідувачеві пропонують десятки посилань на ролики, чимось схожі на переглянутий. Або коли Google у відповідь на один і той же запит ранжує посилання в результатах пошуку для одного користувача інакше, ніж для іншого, оскільки враховує історію пошуків [3].

## 3. Навчання з Підкріпленням

Таке навчання є окремим випадком контрольованого навчання, але вчителем в даному випадку є «середовище». Машина (її в цій ситуації часто називають «агент») не має попередньої інформацією про середовище, але має можливість здійснювати в ній будь-які дії. Середа реагує на ці дії і тим самим надає агенту дані, які дозволяють йому реагувати на них і

вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

Навчання з підкріпленням використовується для вирішення більш складних завдань, ніж навчання з учителем і без вчителя. Воно використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами шляхом набуття досвіду, отримуючи зворотний зв'язок при кожному зіткненні. Навчання з підкріпленням використовується також в логістиці, при складанні графіків і плануванні завдань, при навчанні машини логічним іграм (покер, нарди, го і ін.) [3].

#### 4. Частково контрольоване навчання

Метода поєднує в собі елементи контрольованого та неконтрольованого навчання. Алгоритм навчається на наборі даних, який містить лише частину маркованих даних (чітко визначені вхідні дані), а решта даних не маркована. Цей метод використовується, коли недостатньо маркованих даних, але доступна велика кількість немаркованих даних. Алгоритм спочатку навчається на маркованих даних, а потім використовує отримані знання для класифікації або кластеризації немаркованих даних, тим самим покращуючи точність і ефективність моделі. Частково контрольоване навчання зазвичай використовується в розпізнаванні зображень, обробці природної мови та біоінформатиці, де маркування даних вручну може бути дорогим і трудомістким. Цей метод дозволяє ефективно використовувати наявні ресурси та досягати високих результатів навіть з обмеженою кількістю маркованих даних.

#### 5. Нейронні мережі і глибоке навчання

Для машинного навчання використовують різні технології та алгоритми. Зокрема, можуть застосовуватися дискримінантний аналіз, байєсовські класифікатори та багато інших математичних методів. Але в кінці ХХ століття все більше уваги почали приділяти штучним нейронним

мережам (ANN). Черговий вибух інтересу до них почався в 1986 році, після істотного розвитку т.зв. «Методу зворотного поширення помилки», який з успіхом застосували при навчанні нейронної мережі.

ANN є системою з'єднаних і взаємодіючих між собою штучних нейронів, виконаних на основі порівняно простих процесорів. Кожен процесор ANN періодично отримує сигнали від одних процесорів (або від сенсорів, або від інших джерел сигналів) і періодично посилає сигнали іншим процесорам. Всі разом ці прості процесори, з'єднані в мережу, здатні вирішувати досить складні завдання.

Найчастіше нейрони розташовуються в мережі за рівнями (їх ще називають шарами). Нейрони першого рівня — це, як правило, вхідні. Вони отримують дані ззовні (наприклад, від сенсорів системи розпізнавання осіб) і після їх обробки передають імпульси через синапси нейронів на наступному рівні. Нейрони на другому рівні (його називають прихованим, оскільки він безпосередньо не пов'язаний ні з входом, ні з виходом ANN) обробляють отримані імпульси і передають їх нейронам на вихідному рівні. Оскільки мова йде про імітацію нейронів, то кожен процесор вхідного рівня пов'язаний з декількома процесорами прихованого рівня, кожен з яких, в свою чергу, пов'язаний з декількома процесорами рівня вихідного. Така архітектура найпростішої ANN, яка здатна до навчання і може знаходити прості взаємозв'язку в даних.

Глибоке (глибинне) навчання може бути застосоване лише по відношенню до більш складних ANN, що містить кілька прихованих рівнів. При цьому рівні нейронів можуть чергуватися з шарами, які виконують складні логічні перетворення. Кожен наступний рівень мережі шукає взаємозв'язки в попередньому. Така ANN здатна знаходити не тільки прості взаємозв'язки, а й взаємозв'язки між взаємозв'язками. Саме завдяки переходу на нейромережу з глибинним навчанням компанії Google вдалося різко підвищити якість роботи свого популярного продукту «Перекладач».

Зокрема, якість перекладу між англійською та французькою мовами підвищився відразу на 7 балів, тобто більш ніж на 20%. Попередня система, яка виконувала фразовий статистичний машинний переклад, домоглася подібного поліпшення за весь час свого існування (з 2006 року) [3].

#### **1.4 Типи машинного навчання**

Існує два основних типи машинного навчання: індуктивне та дедуктивне.

Індуктивне навчання (також називається навчанням на основі прецедентів), полягає в тому, що система вивчає правила та закономірності з набору прикладів. Це схоже на те, як навчаються люди: люди формують уявлення про світ на основі досвіду та спостережень.

Алгоритми індуктивного машинного навчання отримують великі обсяги даних і шукають у них повторювані шаблони. Ці шаблони використовуються для прогнозування або класифікації нових випадків. Цей підхід особливо корисний, коли немає чіткої теорії або коли теорію важко застосувати.

Дедуктивне навчання, з іншого боку, ґрунтується на наявних знаннях та правилах. Цей метод використовується, коли доступні дані обмежені або коли важко зібрати достатньо репрезентативний набір даних.

Алгоритми дедуктивного машинного навчання використовують логічні міркування та правила, щоб робити висновки з даних. Цей метод особливо ефективний, коли у є чітка логічна структура та стандартизовані дані [4].

Переваги індуктивного навчання включають здатність ідентифікувати складні шаблони в даних, відсутність потреб у попередніх знаннях предметної області та здатність ефективно обробляти великі обсяги даних. Однак воно може бути чутливим до якості даних, вимагати



великих обсягів даних для навчання, і ускладнювати інтерпретацію результатів. З іншого боку, дедуктивне навчання використовує наявні знання та правила, є ефективним при обмежених даних і його легко інтерпретувати, але має обмеження, пов'язані з наявними знаннями та правилами, не може виявляти нові закономірності в даних і може бути неточним, якщо знання або правила є неповними або неправильними.

Наприклад, передбачення завтрашньої погоди є класичною задачею, яку можна вирішити за допомогою машинного навчання.

Індуктивний підхід до прогнозування погоди може використовувати чисельні фактори, такі як географічні координати, рельєф і рух теплих та холодних вітрів. Модель, навчена на історичних погодних даних, може використовувати ці фактори для створення складних моделей, які враховують різноманітні взаємодії між змінними, які впливають на погоду.

Дедуктивний підхід до прогнозування погоди може використовувати метеорологічні правила та знання про атмосферні процеси. Ці правила та знання можуть бути закодовані в експертну систему, яка може робити висновки про погоду на основі поточних даних.

Індуктивне та дедуктивне навчання — це два потужних методи машинного навчання, які можна використовувати для вирішення широкого кола задач. Вибір методу залежить від конкретного поставленого завдання та доступних даних і ресурсів.

Варто зазначити, що машинне навчання не є панацеєю. Моделі машинного навчання можуть бути чутливими до якості даних, а неповні або неправильні дані можуть давати неточні або оманливі результати.

Тому важливо ретельно відбирати дані, оцінювати продуктивність моделей машинного навчання та використовувати кілька методів для отримання більш надійних результатів.

## **1.5 Машинне навчання та статистика**

Машинне навчання та статистика часто плутають, оскільки вони мають спільні риси та використовуються для аналізу даних і створення прогнозних моделей. Хоча обидві дисципліни використовують математичні та обчислювальні методи для виявлення закономірностей у даних, проте вони підходять до цих завдань з різними акцентами та цілями.

Машинне навчання спрямоване на створення моделей, які можуть самостійно вивчати нові дані та покращувати свої прогнози з часом. Воно використовує алгоритми для оптимізації моделей на основі заданих критеріїв, наприклад, мінімізацію помилок прогнозування. Хоча моделі машинного навчання важко зрозуміти через їхню складність, цей підхід забезпечує високу точність прогнозування, здатність до самонавчання та ефективність при роботі з великим обсягом даних. Недоліки машинного навчання включають складність інтерпретації моделі, чутливість до якості даних і високі обчислювальні витрати [5].

Статистичні моделі зазвичай зосереджуються на дослідженні зв'язків між змінними, розуміння структури даних і перевірку гіпотез. Вони використовують такі методи, як лінійна регресія, щоб знайти пряму лінію, яка найкраще відповідає зв'язку між змінними. Результати статистичних моделей зазвичай легко інтерпретувати через їх чіткі математичні основи та можливість візуалізації даних. Статистичні методи в основному використовуються для аналізу даних, візуалізації та перевірки гіпотез. Їх переваги включають легкість інтерпретації, чіткі математичні основи та меншу обчислювальну вартість. Однак, порівняно з машинним навчанням, передбачення можуть бути менш точними, і оскільки вони не здатні до самонавчання, то можуть бути неефективними під час обробки великих обсягів даних [5].

Хоча між цими двома методами є відмінності, машинне навчання часто використовує статистичні методи для оцінки та перевірки моделей.

Наприклад, регресія та байєсовські мережі є ключовими компонентами машинного навчання. Це свідчить про те, наскільки тісно пов'язані ці дві дисципліни.

Незважаючи на те, що деякі люди вважають, що статистичні методи застаріли, насправді вони все ще є важливим інструментом для аналізу даних. Статистика надає потужні методи обробки та аналізу даних, які доповнюють можливості машинного навчання. Без статистичних основ багато моделей машинного навчання були б неефективними та неточними.

Машинне навчання та статистичні методи широко використовуються компаніями та організаціями для аналізу великих обсягів даних і розробки алгоритмів для точних прогнозів. Обидва методи надають інструменти, необхідні для досягнення поставлених цілей та допомагають зрозуміти дані та приймати обґрунтовані рішення.

Машинне навчання та статистичні методи доповнюють один одного, і вибір залежить від конкретної задачі, доступних даних і вимог щодо точності прогнозування та інтерпретації моделі. Машинне навчання підходить для завдань, де точність передбачення висока, а інтерпретація моделі не є критичною, тоді як статистичні методи підходять для задач, де інтерпретація моделі є важливою.

## **1.6 Регресія у машинному навчанні**

Основною метою машинного навчання є виявлення зв'язків між різними факторами (регресорами, ознаками) і цільовим результатом. Це дозволяє робити точні прогнози та висновки на основі нових даних. Регресія є ключовим методом у цій галузі, оскільки вона допомагає зрозуміти та змодельовати ці зв'язки. Задача регресії полягає у встановленні зв'язку між парами змінних. Оскільки зв'язок між спостереженнями та цільовими результатами часто складний і

багатовимірний, регресія намагається знайти найбільш підходящу модель, яка мінімізує похибку між прогнозованими та фактичними значеннями. Процес включає вибір моделі, налаштування параметрів і оцінку якості для забезпечення надійних прогнозів. Однією з найпростіших моделей є лінійна модель. У машинному навчанні модель — це математична функція, яка збирає спостереження, обробляє їх і поєднує з функцією регресії для прогнозування значення цільової змінної [6].

Лінійна регресія є однією з найпростіших моделей регресії. Вона ґрунтується на припущенні, що існує лінійний зв'язок між незалежними змінними (факторами) і залежною змінною (наслідком). Переваги лінійної регресії включають простоту інтерпретації, наявність математичних методів і широке застосування до різноманітних задач.

Лінійна модель — це інструмент, що використовується для зв'язку між незалежними та залежними змінними. Цей зв'язок показує, як зміни в одній змінній впливають на зміни іншої змінної. Лінійна регресія не тільки дозволяє визначити, які фактори мають найбільший вплив на результат, але також дозволяє створювати прогнозні моделі, які оцінюють майбутнє значення залежної змінної на основі нових даних. Моделі лінійної регресії можна розширити, щоб врахувати складніші взаємозв'язки, що робить їх гнучким інструментом аналізу даних. Загальний вигляд лінійної функції однієї змінної можна записати так:

$$h(x) = k * x + b \quad (1.1)$$

де  $h(x)$  - прогнозоване значення залежної змінної для значення  $x$  незалежної змінної,  $k$  - коефіцієнт нахилу, який описує, наскільки змінюється залежна змінна при зміні незалежної змінної на одиницю, а  $b$  - вільний член, який описує значення залежної змінної, коли незалежна змінна дорівнює нулю. Класичний приклад лінійної залежності - залежність відстані від часу ( $s = v*t$ ).

Діаграма, що ілюструє використання лінійної моделі, а саме, пряму лінію однієї змінної, надає можливість аналізу взаємозв'язку між різними вимірюваними факторами, показана на рисунку 1.1.

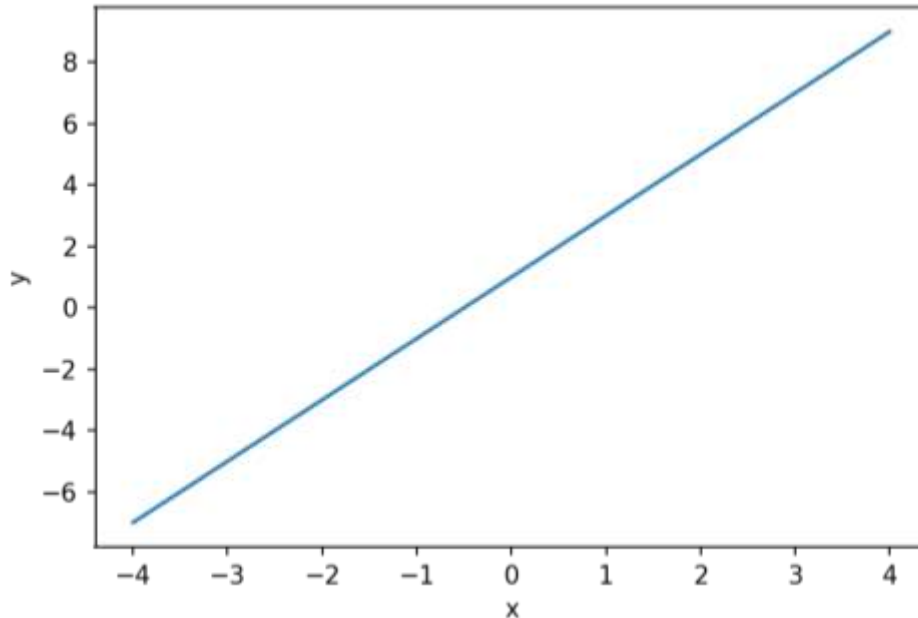


Рисунок 1.1. Графік лінійної функції однієї змінної

У разі, коли є кілька спостережуваних змінних, лінійні функції мають вигляд:

$$h(x_1, x_2, \dots, x_N) = a_0 + a_1 * x_1 + a_2 * x_2 + \dots + a_N * x_N, \quad (1.2)$$

де  $a_0, a_1, \dots, a_N$  - також довільні числа. Графік цієї функції - теж пряма, але вже в N-вимірному просторі, приклад графіку можна побачити на рисунку 1.2.

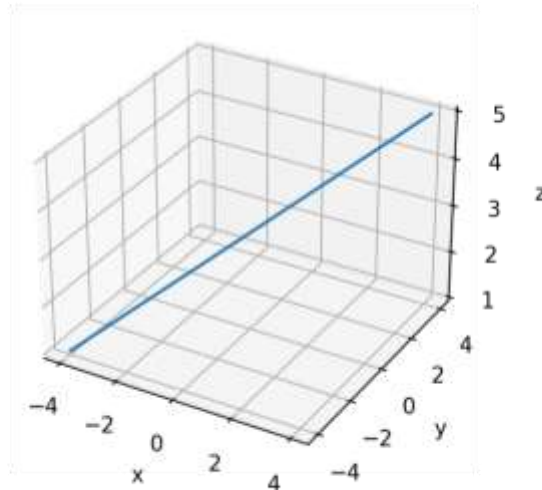


Рисунок 1.2. Графік лінійної функції двох змінних

Підсумовуючи, робимо такі висновки: лінійна регресія — це метод, який дозволяє досліджувати зв'язок між спостережуваною змінною та цільовою змінною за допомогою лінійної залежності. Використання конкретних моделей на основі наявних даних, де  $x$ -змінні є спостережуваними, а  $y$ -змінні — цільовими. Оптимальний підхід полягає у виборі коефіцієнта для кожної змінної, щоб визначити її вплив на модель. Якість змінних оцінюється за допомогою функції втрат, яка вказує, наскільки точно прогнози моделі відповідають цільовому значенню. Процес підбору коефіцієнтів називається навчанням моделі і здійснюється за допомогою спеціальних алгоритмів оптимізації. Параметри, які використовуються в конкретній моделі, називаються вагами. Вони вказують на важливість кожної змінної в прогнозі. Чим більше значення ваги, тим сильніший вплив відповідної змінної на прогноз. Лінійна регресія є інтуїтивно зрозумілою моделлю, оскільки вона дозволяє визначити, які фактори є важливими, а які — ні. Після завершення процесу навчання модель готова до використання. Достатньо ввести набір  $x$ -змінних у отриману функцію (модель) для обчислення прогнозів [7].

## 1.7 Поширені алгоритми в машинного навчання

## 1. Лінійна регресія

Лінійна регресія - це статистичний метод, який досліджує зв'язок між однією або кількома незалежними змінними та залежною змінною, припускаючи, що між ними існує лінійний зв'язок. Лінійну регресію можна виразити у вигляді рівняння, яке представляє пряму лінію, яка найбільш точно описує зв'язок між вхідною змінною  $X$  і вихідною змінною  $Y$ . Основна ідея полягає в мінімізації різниці між фактичними та передбаченими значеннями залежної змінної для створення лінійної моделі, яка найкраще відповідає спостережуваним даним. Фактичні та прогнозовані залежні змінні. Лінійна регресія знаходить оптимальні параметри цієї лінійної залежності, які найкраще описують зміни залежної змінної від незалежних змінних. Таким чином, цей метод дозволяє використовувати просту лінійну модель для прогнозування значення залежної змінної на основі інших змінних. Результат побудови подібного рівняння буде мати вигляд подібний до рисунку 1.3.

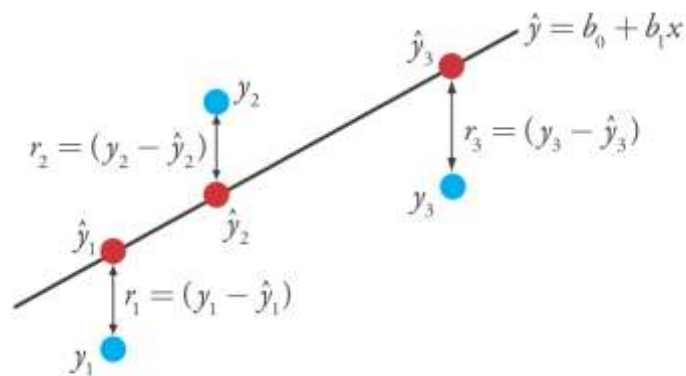


Рисунок 1.3. Приклад Лінійної Регресії

Функція  $Y = B_0 + B_1 * X$  використовується, щоб знайти значення  $Y$  за відомим значенням  $X$ . Основна мета лінійної регресії - знайти значення коефіцієнтів  $B_0$  і  $B_1$ .

Можна оцінити моделі регресії за допомогою різних методів, зокрема лінійної алгебри та методів найменших квадратів.

За допомогою лінійної регресії можна використовувати деякі практичні прийоми для підвищення точності прогнозів. Якщо можливо, важливо усунути спільні (корелюючі) фактори та зменшити шум у даних [7].

Лінійна регресія є швидким і простим методом, який чудово підходить як перший алгоритм для вивчення [8].

## 2. Логістична регресія

Логістична регресія - це статистичний інструмент, спрямований на моделювання біноміального результату з однією або кількома пояснювальними змінними. Логістичну регресію добре використовувати для завдань бінарної класифікації, тобто коли на виході потрібно отримати відповідь, до якого з двох класів належить об'єкт. Логістична регресія схожа на лінійну тим, що в ній теж потрібно знайти значення коефіцієнтів для вхідних змінних, але різниця в тому, що вихідне значення перетвориться за допомогою нелінійної або логістичної функції. Логістична функція перетворює будь-яке значення в число в межах від 0 до 1. Так і передбачається ймовірність належності до одного чи другого класу. Тобто, на відміну від логістичної регресії, тут не проводять передбачення значення числової змінної, виходячи з вибірки вихідних значень, а замість цього значенням функції є ймовірність того, що це початкове значення належить до певного класу. Приклад побудови логістичної функції зображений на рисунку 1.4.



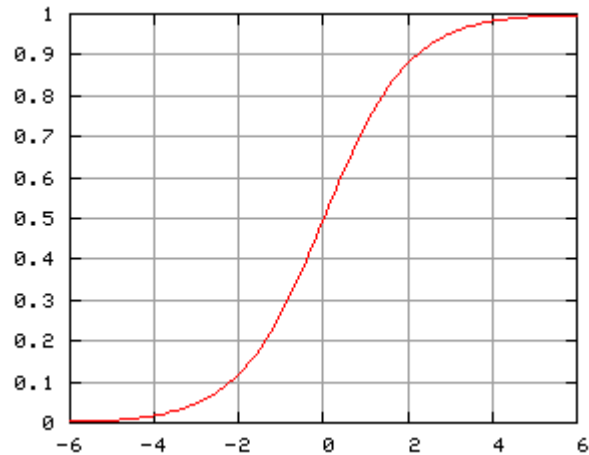


Рисунок 1.4. Приклад логістичної функції

За допомогою статистичних методів логістична регресія дозволяє генерувати результат, який насправді представляє ймовірність того, що задане вхідне значення належить даному класу.

У задачах двочленної логістичної регресії ймовірність того, що вихід належить одному класу, буде  $P$ , тоді як він належить іншому класу  $1-P$  (де  $P$  - число між 0 і 1, оскільки воно виражає ймовірність) [9].

### 3. К-найближчих сусідів (KNN)

Алгоритм К-найближчих сусідів (KNN) - це проста та інтуїтивно зрозуміла техніка машинного навчання для класифікації та пошуку. Основна ідея полягає в тому, щоб класифікувати об'єкти відповідно до категорій їхніх найближчих сусідів у просторі ознак. KNN знаходить  $k$  найближчих точок у навчальному наборі даних і визначає тип нового об'єкта більшістю голосів цих сусідів. Таким чином, цей метод забезпечує простий і ефективний метод прогнозування, особливо коли дані добре розділені в просторі ознак, але він може бути обчислювально витратним для великих наборів даних. Роботу алгоритму KNN можна побачити на рисунку 1.5. [9].

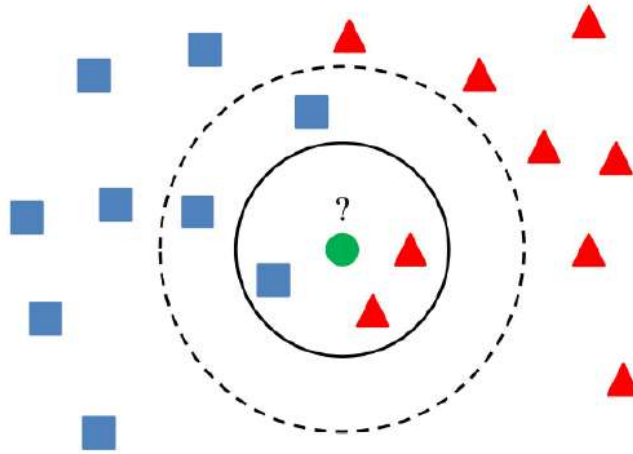


Рисунок 1.5. Приклад K-найближчих сусідів

Незважаючи на те, що алгоритму KNN потрібно багато пам'яті для зберігання всіх минулих даних, прогнози він може генерувати швидко. Крім того, навчальні дані можна регулярно оновлювати для забезпечення точних прогнозів у майбутньому.

У цього методи є своє так зване “Прокляття розмірності”. Цей класифікатор робить припущення, що подібні точки мають подібні мітки. На жаль, у високорозмірних просторах, точки, вибрані з розподілу ймовірностей, майже ніколи не опиняються близько одна до одної [9].

#### 4. Метод Опорних Векторів (SVM)

Метод Опорних Векторів (SVM) - це найпопулярніший для класичної класифікації, тому що він простий та швидкий. Виходячи з того, що об'єкт, який перебуває в N-вимірному просторі, належить до одного з двох класів, метод опорних векторів будує гіперплощину з розмірністю  $(N - 1)$ , щоб всі об'єкти виявилися в одній з двох груп. Ця гіперплощина розділяє точки даних у просторі за їх класом, наприклад, класом 0 або 1. SVM шукає цю найкращу комбінацію поділу, щоб забезпечити найточнішу класифікацію. Таким чином, цей метод забезпечує високу точність прогнозів, оскільки він може добре розрізняти класи та отримувати більш значущі розділові межі навіть у випадках високої складності [9].

Графічний приклад роботи методу Опорних Векторів зображений на рисунку 1.6.

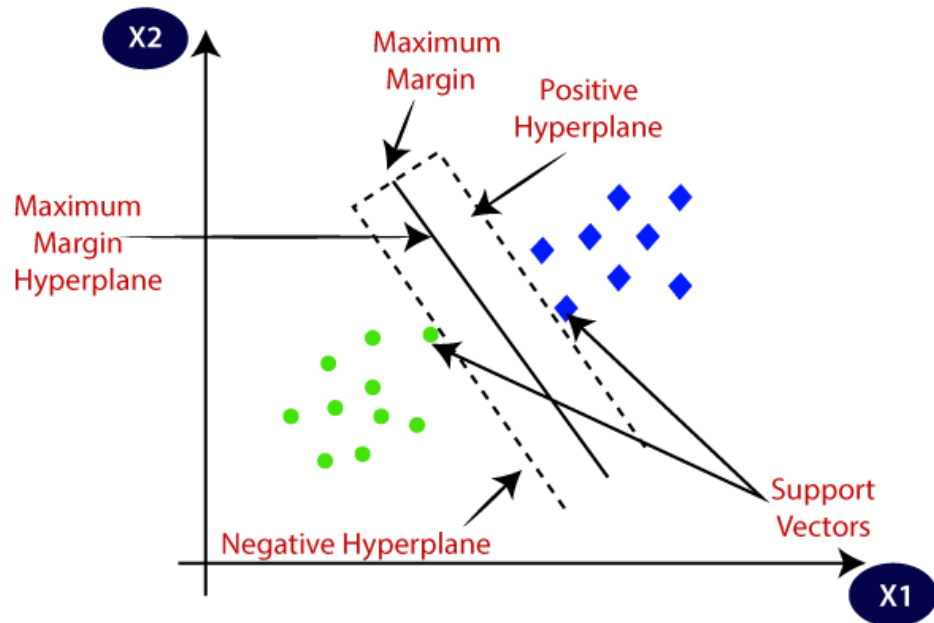


Рисунок 1.6. Приклад Методу Опорних Векторів

Головним недоліком методу є те, що він підходить тільки до розв'язання завдань з двома класами.

SVM є корисними для категоризації текстів та гіпертекстів, оскільки їхнє застосування може значно знижувати потребу в мічених тренувальних зразках як у стандартній індуктивній, так і в трансдуктивній[en] постановках.

Із застосуванням SVM може виконуватися й класифікація зображень. Експериментальні результати показують, що SVM можуть досягати значно вищої точності пошуку, ніж традиційні схеми уточнення запиту, всього лише після трьох-чотирьох раундів зворотного зв'язку про відповідність.

За допомогою ОБМ може здійснюватися розпізнавання рукописних символів [9].

## 5. Лінійний Дискримінантний Аналіз(LDA)

Лінійний дискримінантний аналіз (LDA) - це метод статистичного машинного навчання, який використовується для визначення ліній або

площин, які найкраще відрізняють два або більше типів об'єктів або подій. Метод містить статистичні властивості даних, розраховані для кожного класу. Під час цього передбачається, що дані мають нормальний розподіл, тому перед початком роботи необхідно видалити з даних аномальні значення. Варто зазначити, що, на відміну від цього методу, в інших нормальний розподіл не обов'язковий. Прогнозні класи знаходять шляхом обчислення дискримінантного значення для кожного класу, тобто знаходження лінійної залежності комбінації значень для вибору класу, а потім обирають клас із найбільшим значенням [9].

Загальний вид отриманих результатів у вигляді дисперсії можливо отримати завдяки великій кількості програм, зазвичай вона буде мати вигляд подібний до рисунку 1.7.

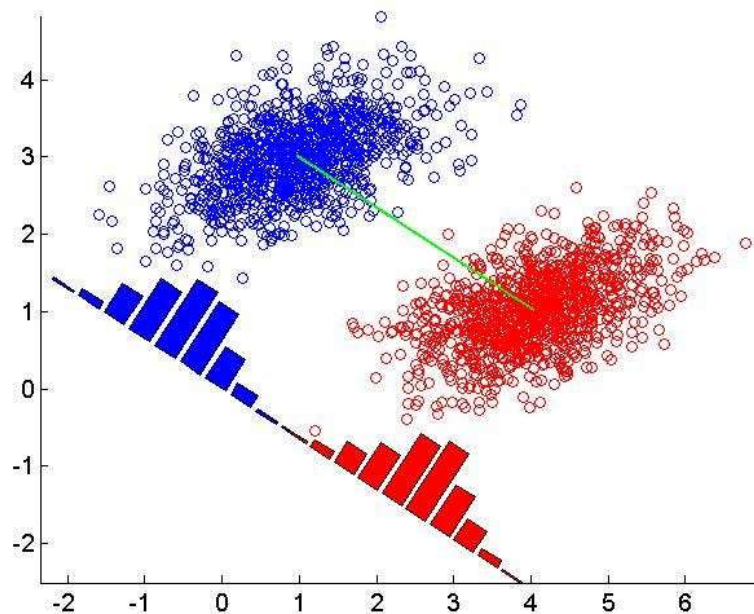


Рисунок 1.7. Приклад роботи Лінійного Дискримінантного Аналізу

Головна перевага методу – це простота реалізації та інтерпретації результатів, недолік – чутливість до розподілу вхідних даних, коли навіть невелике їхнє змінення призводить до значних змін результатів класифікації [9].

## 6. Дерево прийняття рішень

Дерево прийняття рішень - це метод машинного навчання, який використовується для створення моделей класифікації та регресії. Основна ідея цього підходу полягає в тому, щоб розділити дані на менші частини на основі певних умов або правил, що дозволяє приймати рішення на кожному рівні дерева. Дерева рішень знаходять найкращі точки розділення, щоб мінімізувати варіацію або дисперсію в кожній підмножині даних. У результаті цей підхід може чітко виразити та візуалізувати процеси прийняття рішень, роблячи навіть складні завдання інтуїтивно зрозумілими. Приклад найпростішого дерева наведений на рисунку 1.8.

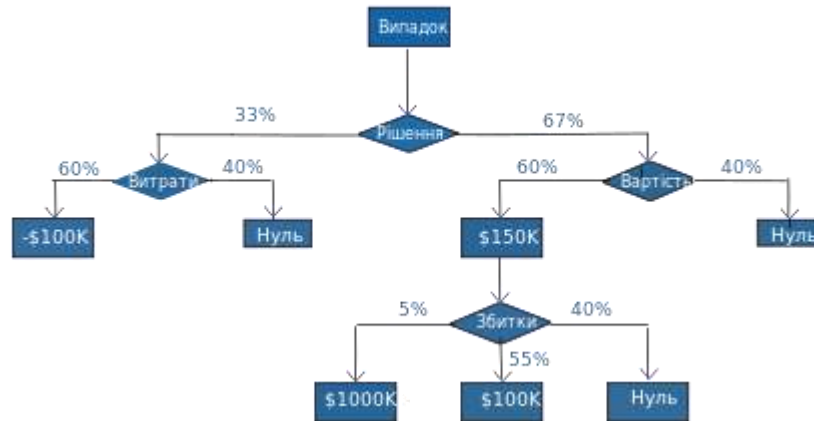


Рисунок 1.8. Приклад Дерева Прийняття Рішень

Термін "листові вузли" відноситься до кінцевої точки дерева, що використовується для прогнозування. Під час процесу прогнозування модель рухається від кореня до кожного термінального вузла та визначає клас, що відповідає цьому вузлу [10].

Дерева характеризуються швидкістю навчання та ефективністю передбачення. Крім того, вони часто дуже точні та не потребують спеціальної підготовки даних.

## 7. Бустинг і AdaBoost

Бустинг - це метод навчання, який поєднує кілька слабких моделей для створення надійної моделі для покращення класифікації або точності

пошуку. Основна ідея полягає в тому, що кожна нова модель враховує, де попередня модель припустилася найбільших помилок, і компенсує ці помилки, тим самим підвищуючи загальну точність [11].

AdaBoost є одним із найпоширеніших варіантів бустингу, де кожна наступна модель навчається на найважливіших даних для правильної класифікації за вагами. Цей процес триває, доки не буде досягнуто певної точності або іншої межі. Таким чином, бустинг можна використовувати для створення складних і потужних моделей, які є високоточними та можуть забезпечити надійні прогнози навіть на дуже складних даних. Процес перевірки першого дерева зображений на рисунку 1.9.

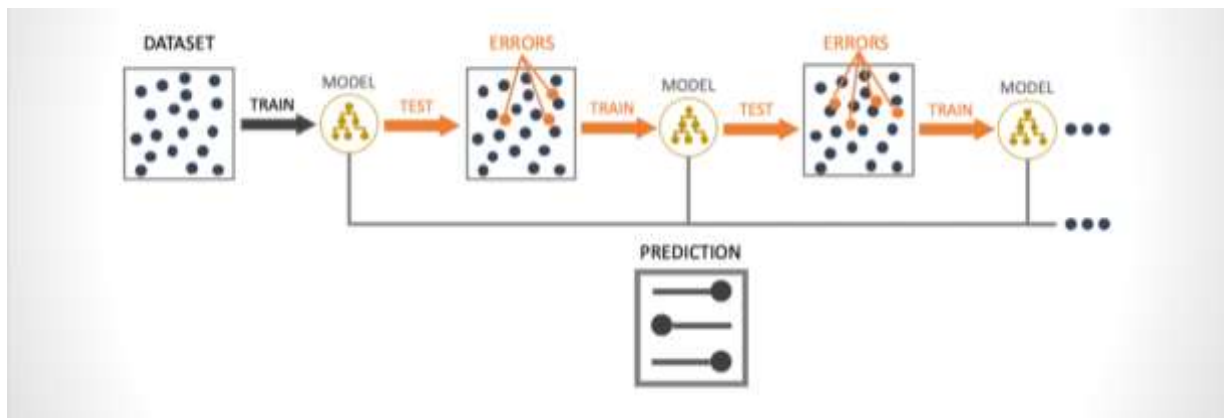


Рисунок 1.9. Приклад Бустингу і AdaBoost

Процес створення проходить серію ітерацій, на кожному кроці адаптуючи ваги для наступної моделі. Після завершення всіх ітерацій моделі використовуються для прогнозування нових даних, а продуктивність кожної моделі оцінюється на основі її точності на навчальних даних.

Оскільки цей метод активно коригує помилки в моделі, тому важливо, щоб вхідні дані були чистими та без викидів.

## 8. Наївний Байєсівський Класифікатор

Наївний Байєс - це простий, але потужний метод машинного навчання, заснований на застосуванні теореми Байєса, припускаючи незалежність ознак. Основна ідея полягає в тому, щоб оцінити ймовірність

того, що об'єкт належить до певної категорії за наявними атрибутами, навіть якщо ці атрибути вважаються незалежними. Наївний Байєсівський Класифікатор знаходить найімовірніший клас на основі атрибутів кожного об'єкта. Таким чином, цей метод забезпечує ефективну та швидку класифікацію та особливо ефективний для великих наборів даних і завдань класифікації тексту [9].

Якщо дані неперервні, то можна використовувати нормальний розподіл для обчислення ймовірностей. Принцип роботи наївного байєсівського класифікатора можна побачити на прикладі, зображеному на рисунку 1.10.

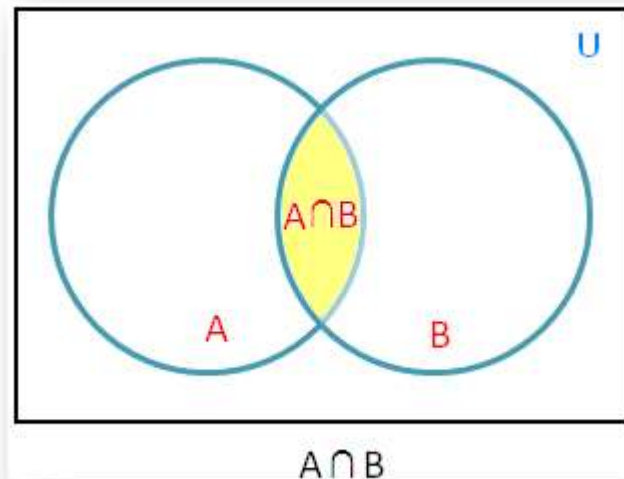


Рисунок 1.10. Приклад Байєсівського Класифікатора

Наївний Баєсовий класифікатор отримав свою назву “наївний” через особливість свого підходу. Він базується на теоремі Баєса і припускає, що всі змінні в наборі даних є незалежними одна від одної. Це “наївне” припущення, оскільки в реальних даних змінні часто корелюють між собою. Таким чином, алгоритм вважає їх “наївними” у відношенні до взаємодії між ними [9].

## 9. Мережі Векторного Квантування (LVQ)

Мережі Векторного Квантування (LVQ) - це техніка машинного навчання для класифікації на основі принципів модельного навчання. Основна ідея полягає в тому, щоб представити кожен клас набором прототипів, типових для цього класу, і використовувати ці прототипи для класифікації нових об'єктів. LVQ шукає найкращу позицію прототипу в просторі ознак, щоб мінімізувати помилку класифікації. Цей алгоритм працює найкраще, коли всі дані знаходяться в одному діапазоні, наприклад, від 0 до 1. Таким чином, цей метод забезпечує зручний та інтуїтивно зрозумілий метод класифікації, який дозволяє вам агрегувати дані та робити точні прогнози на основі невеликої кількості корисних зразків [11].

Роботу алгоритму LVQ можна побачити на рисунку 1.11.

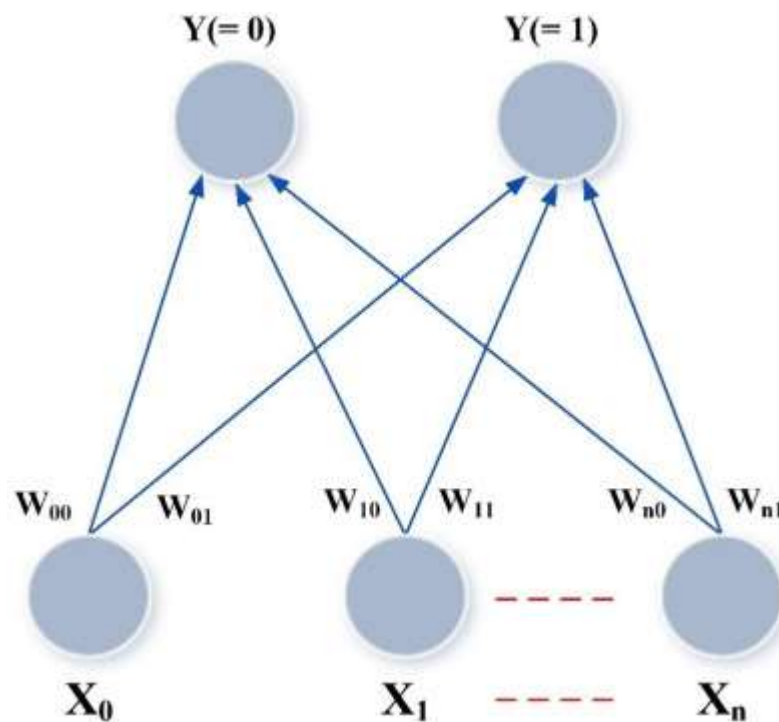


Рисунок 1.11. Приклад Мереж Векторного Квантування

Після завершення процесу навчання можна використовувати метод LVQ для класифікації нового набору даних. Для цього вхідний зразок подається на вхід LVQ, і алгоритм визначає його клас, використовуючи навчені ваги та відстань до центроїдів класів. Наприклад, якщо вхідний



зразок ближчий до центроїда класу “А”, то він буде класифікований як “А”. Якщо він ближчий до центроїда класу “В”, то він буде класифікований як “В”. Таким чином, LVQ може використовуватися для класифікації нових даних на основі навчених ваг та центроїдів класів.

#### 10. Беггінг та Випадковий Ліс

Беггінг (Bagging) та Випадковий Ліс (Random Forest) - це два популярних методи навчання, які використовуються для підвищення точності моделей машинного навчання. Основна ідея беггінгу полягає в тому, щоб побудувати кілька моделей (зазвичай дерев рішень) на різних підмножинах даних і поєднати їхні прогнози для зменшення мінливості та покращення узгодженості результатів.

Випадковий ліс — це розширення беггінгу, яке вводить додаткові міркування у вибір функцій під час побудови кожного дерева та може зменшити кореляції між окремими деревами. Разом ці методи здатні створювати потужні та надійні моделі, ефективно зменшувати зайве навчання та покращувати загальну точність прогнозування [8].

Тренувальні дані розділяються на безліч випадкових підвибірок, для кожної з яких створюється окрема модель, цей процес можна побачити на рисунку 1.12.

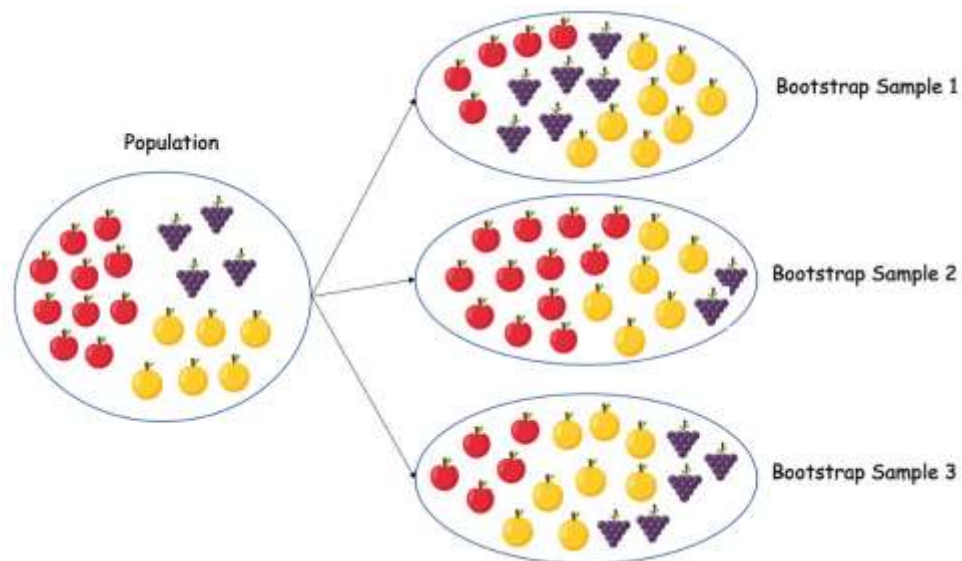


Рисунок 1.12. Приклад Беггінгу та Випадкового Лісу

У алгоритмі випадкового лісу дерево створюється на основі кожної випадкової вибірки навчальних даних і використовується для прийняття рішень. Під час побудови кожного дерева випадкового лісу вибирається підмножина випадкових зразків (з поверненням) з навчального набору даних. Дерево будується на цьому підмножині, і прийняття рішень відбувається шляхом голосування багатьох дерев. Випадковий ліс є потужним алгоритмом для класифікації та регресії, оскільки він зменшує перенавчання та дозволяє враховувати важливі ознаки в даних.

Якщо алгоритми з великими відмінностями між прогнозами, наприклад дерева рішень, добре працюють із набором даних, часто покращити ці результати можна за допомогою методу бегінгу.

## **РОЗДІЛ 2 ПРОЄКТНІ РІШЕННЯ ЩОДО ОПТИМІЗАЦІЇ ФІНАНСОВОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ НА ПРИКЛАДІ ТОВ «ROZETKA»**

### **2.1 Побудова Z-моделей визначення ймовірності настання банкрутства ТОВ «ROZETKA» та визначення на основі них проблемних зон управління фінансовою діяльністю та їх оптимізація**

Сформулюємо постановку задачі наступним чином:

На основі Z-моделей (які отримані засобами регресійного аналізу) визначимо ймовірність банкрутства ТОВ «ROZETKA» у найближчі 1-3 роки. Вхідні дані взяти із фінансових звітів.

Задача визначення ймовірності банкрутства найбільшого підприємства галузі є актуальною, адже достовірна й своєчасна ідентифікація негативних факторів впливу на фінансово-господарську діяльність компанії дає можливість сформувати відповідну фінансову політику та розробити заходи, які направлені на підвищення ефективності використання фінансових ресурсів. А також підвищити платоспроможність, забезпечити конкурентоспроможність у довгостроковому періоді, а головне – попередити ризик банкрутства підприємства. Діагностика банкрутства – це своєчасне виявлення неплатоспроможності, збитковості, фінансової залежності від зовнішніх джерел фінансування, низької ділової активності.

Причини банкрутства підприємств можуть бути найрізноманітнішими. Їх можна розділити на дві групи:

- зовнішні, які практично дуже важко (іноді неможливо) врахувати;
- внутрішні, що безпосередньо залежать від форм, методів та організації роботи на самому підприємстві.

Усі методи діагностики банкрутства поділяються на кількісні та якісні. В дослідженнях найбільш поширеними серед кількісних методів прогнозування банкрутства є зарубіжні та вітчизняні факторні моделі, де факторами виступають окремі показники фінансової діяльності підприємства [12].

Після аналізу розроблених зарубіжних методик діагностики ймовірності банкрутства для розрахунків обрано п'ять моделей.

**1. П'ятифакторна модель Е. Альтмана (1968 рік).**

$$Z = 1,2X_1 + 1,4X_2 + 3,3X_3 + 0,6X_4 + X_5, \quad (2.1)$$

де  $X_1$  – відношення оборотного капіталу до валюти балансу;

$X_2$  – відношення нерозподіленого прибутку до валюти балансу;

$X_3$  – відношення валового прибутку до валюти балансу;

$X_4$  – відношення вартості власного капіталу до вартості всіх зобов'язань;

$X_5$  – відношення доходу від реалізації до валюти балансу.

Якщо значення  $Z < 1,8$  – тоді ймовірність банкрутства дуже висока (понад 80%);  $1,81 < Z < 2,7$  – висока (від 40% до 50%);  $2,71 < Z < 2,99$  – можлива (від 15% до 20%);  $Z > 3$  – дуже низька.

**2. Модель Р. Ліса (1972 рік).**

$$Z = 0,063X_1 + 0,092X_2 + 0,057X_3 + 0,001X_4, \quad (2.2)$$

де  $X_1$  – відношення оборотного капіталу до всіх активів;

$X_2$  – відношення прибутку від реалізації до всіх активів;

$X_3$  – відношення нерозподіленого прибутку до всіх активів;

$X_4$  – відношення власного капіталу до позичкового капіталу.

Якщо значення  $Z < 0,037$  – висока ймовірність банкрутства;  $Z = 0,037$  – граничне значення;  $Z > 0,037$  – низька ймовірність банкрутства.

**3. Модель Р. Таффлера (1977 рік).**

$$Z = 0,53X_1 + 0,13X_2 + 0,18X_3 + 0,16X_4, \quad (2.3)$$

де  $X_1$  – відношення прибутку від реалізації продукції до короткострокових зобов'язань;

$X_2$  – відношення оборотних активів до зобов'язань;

$X_3$  – відношення короткострокових зобов'язань до всіх активів;

$X_4$  – відношення власного капіталу до позичкового капіталу.

Якщо значення  $Z < 0,2$  – ймовірність банкрутства досить висока;  $0,2 < Z < 0,3$  – можливе банкрутство;  $Z > 0,3$  – ймовірність банкрутства низька.

#### 4. Модель Г. Спрінгейта (1978 рік).

$$Z = 1,03X_1 + 3,07X_2 + 0,66X_3 + 0,4X_4 \quad (2.4)$$

де  $X_1$  – частка оборотного капіталу в структурі активів;

$X_2$  – відношення чистого прибутку до виплати податків і відсотків до сукупних активів;

$X_3$  – відношення чистого прибутку до виплати відсотків до поточних зобов'язань;

$X_4$  – відношення доходу від реалізації продукції до всіх активів.

Якщо значення  $Z < 0,862$  – підприємство є потенційним банкрутом;  $Z = 0,862$  – граничне значення ймовірності банкрутства;  $Z > 0,862$  – відсутність ймовірності банкрутства. Після аналізу найбільш поширених зарубіжних методик діагностики ймовірності банкрутства можна зазначити, що основна їх перевага полягає в простоті розрахунків та можливості використання під час проведення зовнішнього діагностичного аналізу. Так п'ятифакторна модель Е. Альтмана відображає різні аспекти господарської діяльності підприємства. Вона надає можливість динамічного прогнозування змін фінансової стійкості та інтерпретації інтегрального показника. В моделі Г. Спрінгейта похибка прогнозу не перевищує 10%; тобто забезпечується достатній рівень надійності прогнозування ймовірності банкрутства.

Головним недоліком цих зарубіжних моделей є те, що вони розроблені за результатами дослідження підприємств США й Західної

Європи. Вагові коефіцієнти не адаптовані до сучасних умов трансформації вітчизняної економіки й не враховують специфіку діяльності українських підприємств, а саме: систему бухгалтерського обліку, податкове законодавство, значний вплив інфляції на формування фінансових показників, галузеву приналежність підприємства тощо. Ці моделі мають кілька суттєвих обмежень. Їх можна використовувати лише як додаткові моделі паралельно з сучасними вітчизняними моделями [13].

### 5. Модель О.О. Терещенка (2004 р.).

$$Z = 1,5X_1 + 0,08X_2 + 10X_3 + 5X_4 + 0,3X_5 + 0,1X_6, \quad (2.5)$$

де  $X_1$  – відношення грошових надходжень до зобов'язань;

$X_2$  – відношення валюти балансу до зобов'язань;

$X_3$  – відношення чистого прибутку до середньорічної суми активів;

$X_4$  – відношення прибутку до доходу від реалізації;

$X_5$  – відношення виробничих запасів до доходу від реалізації;

$X_6$  – відношення доходу від реалізації до основного капіталу.

Якщо  $0 < Z < 1$  – банкрутство не загрожує,  $1 < Z < 2$  – фінансова стійкість порушена,  $Z > 2$  – існує загроза банкрутства.

## 2.2 Мотивація та аналіз задачі

Дана задача може бути вирішена засобами штучного інтелекту. Постановка задачі включає аналіз фінансових даних ТОВ «ROZETKA» для визначення його ймовірності банкрутства протягом наступних 1-3 років на основі Z-моделей, отриманих за допомогою регресійного аналізу. Для виконання цієї задачі потрібно врахувати кілька ключових етапів, а саме:

1. Збір та підготовка даних: На цьому етапі потрібно зібрати фінансові дані ТОВ «ROZETKA» з фінансових звітів за певний період часу. Це можуть бути дані про прибуток, витрати, активи, зобов'язання, а також

будь-які інші фінансові показники, які можуть бути важливими для аналізу банкрутства.

2. Побудова Z-моделей: За допомогою регресійного аналізу необхідно побудувати Z-моделі, які будуть використовуватися для прогнозування ймовірності банкрутства. Z-моделі зазвичай базуються на статистичних залежностях між фінансовими показниками та ризиком банкрутства.

3. Аналіз результатів моделювання: Після побудови Z-моделей необхідно проаналізувати результати, щоб оцінити їх ефективність та адекватність. Це може включати аналіз точності прогнозів, виявлення факторів, які найбільше впливають на ймовірність банкрутства, та інші аспекти.

4. Прогнозування ймовірності банкрутства: На основі побудованих Z-моделей можна прогнозувати ймовірність банкрутства ТОВ «ROZETKA» протягом наступних 1-3 років.

Мотивація застосування такого аналізу полягає у можливості заздалегідь виявити фінансові проблеми та ризики банкрутства компанії, що дозволить вчасно прийняти стратегічні рішення для запобігання негативним наслідкам і збереження фінансової стабільності.

### **2.3 Формалізація задачі (виділення вхідних-вихідних параметрів, зав'язків між ними тощо)**

Формалізація задачі передбачає виділення вхідних та вихідних параметрів, а також визначення залежностей між ними. У визначенні ймовірності банкрутства ТОВ «ROZETKA» можна виділити наступні параметри:

1. Вхідні параметри:

- Фінансові дані звітності: це може включати дані про прибуток, витрати, активи, зобов'язання, оборотний капітал і т.д.
  - Показники ліквідності: наприклад, поточний відношення, швидкого відношення тощо.
  - Показники рентабельності: чистий прибуток до виручки, рентабельність активів, рентабельність власного капіталу та інші.
  - Фінансові показники заборгованості: наприклад, коефіцієнт покриття відсотків, коефіцієнт заборгованості тощо.
  - Інші ключові фінансові та економічні показники, які можуть впливати на фінансову стійкість компанії.
2. Вихідний параметр:
- Ймовірність банкрутства: числове значення, що вказує на ймовірність банкрутства ТОВ «ROZETKA» протягом наступних 1-3 років.
3. Залежності між параметрами:
- Використання регресійного аналізу для побудови Z-моделей: залежності між фінансовими показниками та ризиком банкрутства можуть бути встановлені шляхом аналізу історичних даних.
  - Врахування взаємозв'язків між фінансовими показниками: різні показники можуть бути взаємозалежними, тому їх взаємовплив слід враховувати при побудові моделей.
  - Аналіз результатів та вибір найбільш важливих параметрів: після побудови моделей важливо проаналізувати їх ефективність та визначити, які фінансові показники найбільше впливають на ймовірність банкрутства.
  - Прогнозування ймовірності банкрутства: на основі побудованих моделей можна здійснити прогноз ймовірності банкрутства ТОВ «ROZETKA» на певний період часу.

Формалізація цих параметрів дозволить створити модель для оцінки фінансової стійкості компанії та прогнозування її ризику банкрутства.



## **2.4 Вибір технології штучного інтелекту – обґрунтування вибору, огляд можливих варіантів, що розглядалися**

Вибір технології штучного інтелекту для прогнозування ймовірності банкрутства підприємства є критичним етапом, оскільки від правильності цього вибору залежить точність та ефективність прогнозу. Нижче розглянемо обґрунтування вибору технології та оглянемо можливі варіанти для розв'язку цієї задачі:

### **1. Обґрунтування вибору технології:**

- **Відповідність завданню:** Вибрана технологія повинна мати можливості для обробки великого обсягу даних і проведення складного аналізу для прогнозування банкрутства.

- **Точність:** Технологія має забезпечувати високу точність прогнозування, щоб уникнути помилкових рішень та забезпечити вчасну реакцію на можливі ризики.

- **Інтеграція:** Важливо, щоб обрана технологія могла інтегруватися з існуючими системами та даними підприємства для отримання повного образу ситуації.

- **Масштабованість:** Технологія повинна бути гнучкою та масштабованою для роботи з різноманітними типами даних та великими обсягами інформації.

### **2. Можливі варіанти для розв'язку задачі:**

- **Методи машинного навчання:** Використання алгоритмів класифікації та регресії, таких як логістична регресія, випадковий ліс, градієнтний бустинг тощо.

- **Нейронні мережі:** Використання глибокого навчання для аналізу великих обсягів даних та виявлення складних залежностей між параметрами.

- Аналітика тексту: Використання методів обробки природної мови для аналізу текстової інформації з фінансових звітів та новинних джерел для отримання додаткових ознак банкрутства.

- Системи експертних знань: Розробка баз знань та правил на основі експертного досвіду для автоматизованого прийняття рішень щодо банкрутства.

Обираючи технологію штучного інтелекту для прогнозування банкрутства, важливо враховувати специфіку підприємства, його потреби та особливості даних. Також рекомендується провести тестування різних варіантів технологій на власних даних для вибору оптимальної моделі.

Логістична регресія, випадковий ліс та градієнтний бустинг - це три різні методи машинного навчання, які часто використовуються для розв'язання задач класифікації та регресії. Нижче наведено короткий огляд кожного з цих методів:

1. Логістична регресія:

- Призначення: Логістична регресія використовується для розв'язання задач класифікації, коли потрібно передбачити ймовірність належності об'єкта до певного класу.

- Основна ідея: Модель логістичної регресії використовує логістичну функцію для оцінки ймовірності того, що екземпляр належить до певного класу, в залежності від значень ознак.

- Переваги: Простота та інтерпретованість моделі, добре підходить для бінарної класифікації.

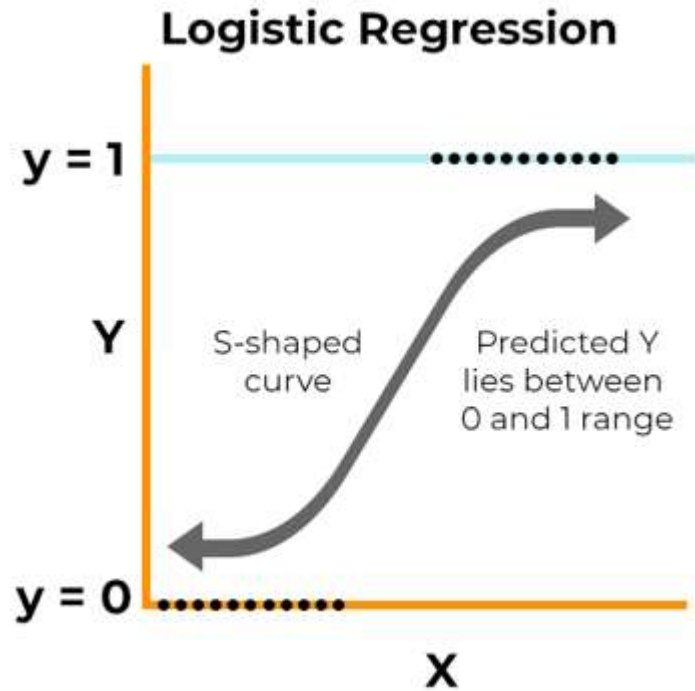


Рисунок 2.1. Модель логістичної регресії

## 2. Випадковий ліс (Random Forest):

- Призначення: Випадковий ліс також використовується для задач класифікації та регресії. Він базується на ідеї ансамблю алгоритмів.
- Основна ідея: Випадковий ліс використовує кілька рішачих дерев для прийняття рішення. Кожне дерево вирішує задачу класифікації, а потім результати об'єднуються для отримання кінцевого результату.
- Переваги: Здатність роботи з великим обсягом даних та високою точністю, стійкість до перенавчання.

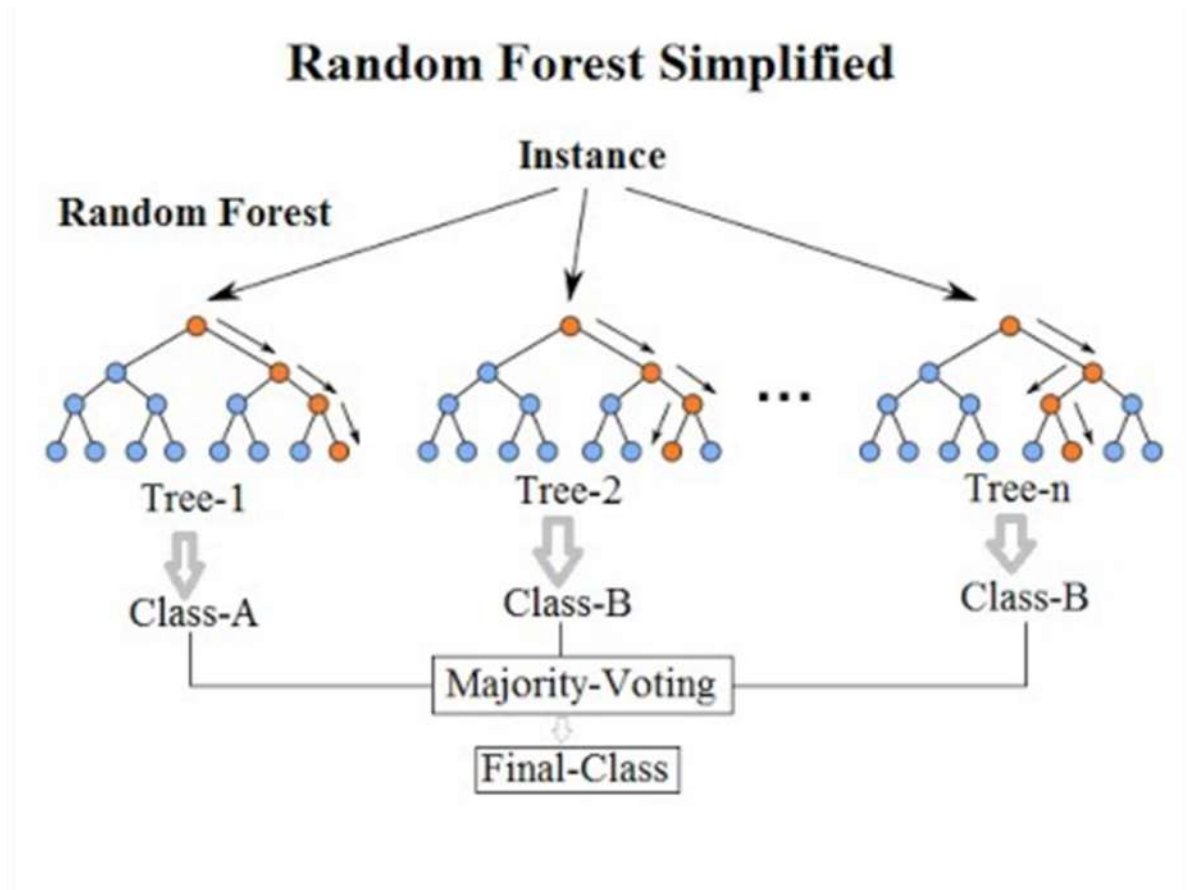


Рисунок 2.2. Модель випадково лісу

### 3. Градієнтний бустинг (Gradient Boosting):

- Призначення: Градієнтний бустинг також використовується для класифікації та регресії. Він також є ансамблевим методом.
- Основна ідея: Градієнтний бустинг використовує послідовне навчання слабких моделей, які покращуються з кожним кроком, адаптуючись до помилок попередніх моделей.
- Переваги: Висока точність, здатність розробляти складні моделі з великою кількістю параметрів.

У кожного з цих методів є свої переваги та обмеження, тому вибір методу залежить від конкретного завдання, обсягу даних та доступних ресурсів. Часто застосовується підход "спробуйте всі три" для порівняння їх ефективності в конкретному контексті задачі.

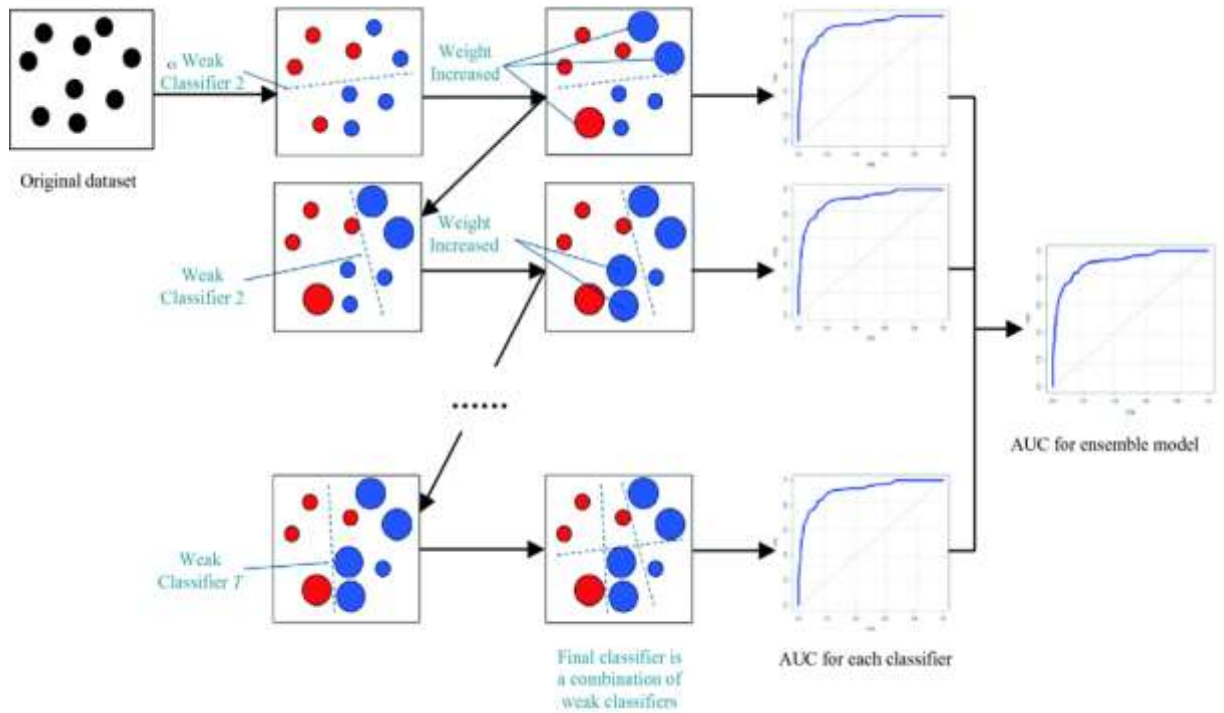


Рисунок 2.3. Модель градієнтного бустингу

Серед трьох моделей ШІ було обрано саме модель логістичної регресії та нейромережі на основі неї. Саме її використовуємо для прогнозування та оцінки ймовірності банкрутства.

## РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Обґрунтування вибору середовища реалізації (Python)

Вибір мови програмування Python для реалізації даної задачі може бути обґрунтований наступними аргументами:

1. Простота вивчення та використання: Python має простий і зрозумілий синтаксис, який дозволяє швидко оволодіти мовою навіть початківцям. Це робить його ідеальним вибором для широкого кола розробників.

2. Широкий вибір бібліотек: Python має велику кількість розширень та бібліотек для обробки даних, статистичного аналізу, машинного навчання та інших суміжних завдань. Наприклад, бібліотеки, такі як Pandas, NumPy, SciPy, Scikit-learn, Matplotlib, дозволяють легко і ефективно працювати з фінансовими даними та будувати моделі прогнозування.

3. Спільнота розробників: Python має велику та активну спільноту розробників, яка постійно розширює функціональність мови, підтримує і вдосконалює бібліотеки та допомагає вирішувати труднощі, які можуть виникнути під час розробки.

4. Переносимість: Python є переносимою мовою програмування, тобто код, написаний на Python, може працювати на різних операційних системах, що робить його універсальним і зручним для розгортання на різних платформах.

5. Інтеграція з іншими технологіями: Python добре інтегрується з іншими мовами програмування та технологіями, що може бути корисним у випадку, якщо для реалізації деяких частин задачі необхідно використовувати інші мови або середовища.

Загалом, Python є потужним та гнучким інструментом, який може ефективно використовуватися для аналізу фінансових даних, побудови

моделей прогнозування та розв'язання багатьох інших завдань, пов'язаних з обробкою і аналізом даних.

STATISTICA - це потужне програмне забезпечення для статистичного аналізу та візуалізації даних, яке широко використовується у наукових дослідженнях, бізнес-аналітиці, фінансовому аналізі та інших галузях. Ось деякі аргументи на користь використання STATISTICA:

1. Потужність і гнучкість: STATISTICA має широкий спектр статистичних методів та інструментів для аналізу даних, включаючи описивну статистику, регресійний аналіз, аналіз варіант, кластерний аналіз, факторний аналіз та інші. Це дозволяє ефективно вирішувати різноманітні завдання аналізу даних.

2. Зручний інтерфейс користувача: STATISTICA має інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко виконувати аналіз даних без необхідності великих навичок програмування. Це робить його доступним для користувачів з різним рівнем підготовки та досвіду.

3. Велика кількість вбудованих функцій і модулів: STATISTICA постачається з багатьма вбудованими функціями та модулями для виконання різних завдань аналізу даних, що включає в себе візуалізацію даних, побудову графіків, підготовку даних для аналізу, статистичний аналіз та багато іншого.

4. Підтримка і документація: STATISTICA має велику базу знань, документацію та підтримку спільноти, що дозволяє користувачам швидко знайти відповіді на свої запитання та вирішити проблеми, які можуть виникнути під час роботи з програмним забезпеченням.

5. Підтримка мов програмування: STATISTICA підтримує вбудовані мови програмування, такі як Python та R, що дозволяє розширити його функціональність та виконувати складні аналітичні завдання, використовуючи додаткові засоби програмування.

Загалом, STATISTICA є потужним інструментом для статистичного аналізу даних, який може бути використаний для вирішення різних завдань у сфері аналітики та досліджень.

### 3.2 Вибір середовища розробки(IDE)

Після вибору мови програмування (в даному випадку Python) наступним кроком є вибір інтегрованого середовища розробки (Integrated Development Environment). Існує багато популярних IDE(Integrated Development Environment).

Перш за все, слід розглянути IDE, яке можна завантажити з офіційного веб-сайту разом із Python, — це Integrated Development and Learning Environment (IDLE). Це просте середовище з обмеженою функціональністю, яке часто використовується для розробки простих інструментів. Але, як на мене, це не найкращий вибір для нашого проєкту. Нижче наведено загальний вигляд вікна середовища IDLE:

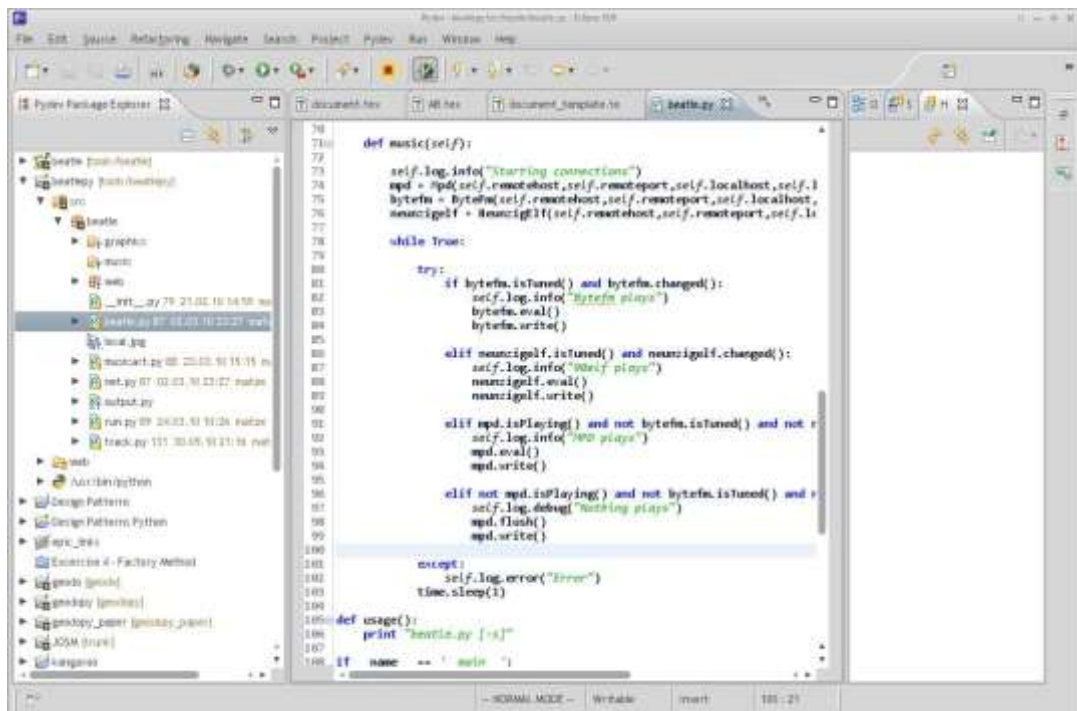


Рисунок 3.1. Зовнішній вигляд IDLE Integrated Development and Learning Environment



Іншим варіантом є PyCharm, розроблений JetBrains, є одним із найпопулярніших інтегрованих середовищ розробки (IDE) для Python.

Найважливіше те, що завдяки своїм інтелектуальним функціям PyCharm значно підвищує продуктивність розробників. Це середовище забезпечує розширене автоматичне завершення коду, що допомагає швидко писати код без помилок і заощадити час на набір та виправлення. Можливість швидко переходити до визначень, оголошень та використання змінних, функцій та класів дозволяє легко орієнтуватися у великому коді. Підтримка різноманітних інструментів для рефакторингу автоматично оновлення коду та змінює його структури без порушення функціональності.

PyCharm також забезпечує вбудовану підтримку різноманітних фреймворків і бібліотек Python, що робить його корисним інструментом для багатьох типів проєктів. Наприклад, підтримка таких веб-фреймворків, як Django, Flask і Pyramid, значно спрощує розробку веб-додатків. Вбудовані інструменти для роботи з науковими бібліотеками, такими як SciPy, NumPy і Pandas, допомагають ефективно розробляти та тестувати наукові та аналітичні програми.

Іншою важливою особливістю є інструменти тестування та налагодження, надані PyCharm. Візуальний налагоджувач дозволяє встановлювати контрольні точки, перевіряти змінні та покроково виконувати код, що значно полегшує процес налагодження. PyCharm підтримує інтеграцію з такими тестовими фреймворками, як pytest і Unittest, що дозволяє легко запускати тести та переглядати результати безпосередньо в IDE.

Інтеграція з такими інструментами CI/CD, як Jenkins і TeamCity, дозволяє легко автоматизувати процеси розробки та розгортання.

JetBrains забезпечує високоякісну підтримку та регулярно оновлює PyCharm, щоб забезпечити доступ до найновіших функцій і виправлень.

Детальна документація, навчальні відео та курси допомагають швидко освоїти PyCharm. Також можна знайти багатьох користувачів PyCharm на форумах, у соціальних мережах і на таких сайтах, як Stack Overflow, де можна знайти відповіді на будь-які запитання та отримати пораду.

Нижче наведено загальний вигляд вікна IDE:

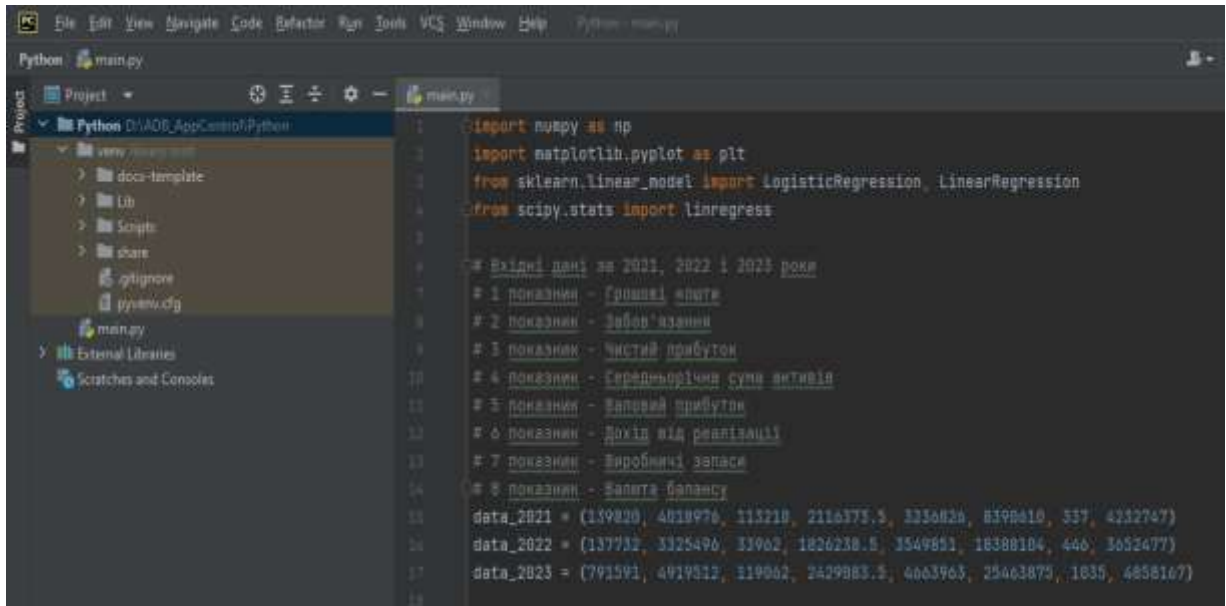


Рисунок 3.2. Зовнішній вигляд IDE PyCharm

Враховуючи всі вищезазначені переваги, я вважаю, що PyCharm буде найефективнішим та найкращим варіантом для розробки нашого проекту. Крім того, PyCharm має дуже зручну функцію швидкого встановлення бібліотек, необхідних для проекту.

### 3.3 Вибір додаткових бібліотек для розробки

У попередньому пункті було обрано PyCharm як середовище розробки, яке має можливість підключення до великої кількості додаткового програмного забезпечення.

Надалі розглянемо бібліотеки, які будемо використовувати для створення нашого проекту:

- NumPy (np):

- Опис: NumPy - це бібліотека для мови програмування Python, яка додає підтримку великих масивів і матриць разом з високорівневими математичними функціями для роботи з цими масивами. Це надає можливість виконувати операції лінійної алгебри, обробки сигналів, та рандомізації даних, що робить його дуже корисним для наукових обчислень та аналізу даних.

- Використання: Часто використовується для обчислень над великими наборами даних, векторизації коду, а також для побудови і обробки матриць.

- Matplotlib (plt):

- Опис: Matplotlib - це бібліотека для створення двовимірних графіків в мові програмування Python. Вона дозволяє будувати різноманітні види графіків, такі як лінійні графіки, гістограми, точкові графіки та багато інших.

- Використання: Використовується для візуалізації даних, щоб зробити їх зрозумілими та легкими для аналізу.

- Scikit-learn (sklearn):

- Опис: Scikit-learn - це бібліотека машинного навчання для мови програмування Python. Вона містить реалізації багатьох алгоритмів навчання з учителем та без учителя, таких як класифікація, регресія, кластеризація, а також інструменти для оцінки та підбору моделей.

- Використання: Використовується для побудови, навчання та оцінки моделей машинного навчання.

- Scipy.stats (linregress):

- Опис: Scipy - це бібліотека для наукових обчислень та технічних обчислень для мови програмування Python. Модуль stats в Scipy містить статистичні функції та інструменти для роботи з даними.

- Використання: Використовується для виконання лінійної регресії, що дозволяє визначити взаємозв'язок між двома змінними,

включаючи розрахунок коефіцієнта кореляції, нахилу, перетину, стандартної помилки та значення  $p$ .

- `linregress`:
  - Опис: `linregress` використовується для побудови простої лінійної регресії та отримання інформації про цю регресію, таку як нахил, перетин, коефіцієнт кореляції та  $p$ -значення.
  - Використання: Часто використовується для аналізу статистичних даних, побудови регресійних моделей та оцінки їхньої ефективності.

### **3.4 Розробка моделі – створення формальної моделі для розв'язку задачі на основі обраної технології**

Для розробки моделі прогнозування банкрутства на основі логістичної регресії і отримання моделі типу Альтмана, нам знадобиться:

1. Збір даних: Зібрати фінансові дані компанії ROZETKA, які включають показники звітності, такі як баланс, звіт про прибутки та збитки, кошторисні дані тощо. Для розрахунку моделі типу Альтмана основними фінансовими показниками є рівень заборгованості, рентабельність активів, оборотність активів, коефіцієнт власного капіталу та ринкова капіталізація.

2. Підготовка даних: Провести попередню обробку даних, таку як видалення відсутніх значень, нормалізація, видалення викидів тощо.

3. Вибір ознак: Визначити набір факторів, що впливають на банкрутство компанії. Для моделі типу Альтмана ці ознаки включають показники фінансового здоров'я, які згадані вище.

4. Побудова моделі логістичної регресії: Використовуючи оброблені дані та вибраний набір ознак, побудуйте модель логістичної регресії для прогнозування ймовірності банкрутства компанії. У моделі типу Альтмана можна використовувати коефіцієнти, які були розроблені

самим Альтманом: Z-бал, що обчислюється як лінійна комбінація фінансових показників.

5. Налаштування та валідація моделі: Налаштуйте параметри моделі та проведіть валідацію для перевірки її точності та ефективності. Для цього можна використовувати різні методи, такі як хресна перевірка або розділення на навчальний та тестовий набори даних.

6. Оцінка результатів: Оцініть результати моделі, провівши аналіз важливості ознак та інші метрики ефективності, такі як точність, чутливість, специфічність тощо.

7. Впровадження та моніторинг: Після успішної побудови та перевірки моделі впровадіть її в робоче середовище. Необхідно також встановити механізми моніторингу та оновлення моделі для забезпечення її актуальності в майбутньому.

### **3.5 Практична реалізація – короткий огляд програмної реалізації моделі в середовищі Python**

Спочатку сформуємо масив вихідних даних та на основі нього розрахуємо коефіцієнти, які характеризують фінансовий стан ТОВ «ROZETKA».

Для українських реалій підходить шестифакторна модель О.О. Терещенка, хоча досі є застарілою, але саме ті коефіцієнти, які у неї входять візьмемо за основу.

$$Z = 1,5X_1 + 0,08X_2 + 10X_3 + 5X_4 + 0,3X_5 + 0,1X_6, \quad (3.1)$$

де  $X_1$  – відношення грошових коштів до зобов'язань;

$X_2$  – відношення валюти балансу до зобов'язань;

$X_3$  – відношення чистого прибутку до середньорічної суми активів;

$X_4$  – відношення прибутку до доходу від реалізації;

$X_5$  – відношення виробничих запасів до доходу від реалізації;

$X_6$  – відношення доходу від реалізації до основного капіталу.

Оцінити ймовірність банкрутства можна за формулою:

$$p = \frac{\exp Z}{1 + \exp Z} \quad (3.2)$$

Таблиця 3.1

### Фінансово-економічні показники (абсолютні) для розрахунку моделі

№	Показник	2021 рік	2022 рік	2023 рік
1	Грошові кошти	139820	137732	791591
2	Зобов'язання	30481+3988495	30896+3294600	61345+4858167
3	Чистий прибуток	113210	33962	119062
4	Середньорічна сума активів	(65128+4167619)/2	(89744+3562733)/2	(85095+4773072)/2
5	Валовий прибуток	3236826	3549851	4663963
6	Дохід від реалізації	8390610	18388104	25463875
7	Виробничі запаси	337	446	1035
8	Валюта балансу	4232747	3652477	4858167

### 3.6 Опис основних функцій програмної реалізації

#### 1. Імпорт бібліотек:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression, LinearRegression
from scipy.stats import linregress
```

- `numpy`: Ця бібліотека використовується для роботи з масивами та числовими даними.
- `matplotlib.pyplot`: Ця бібліотека використовується для візуалізації даних.
- `sklearn.linear_model`: Ця бібліотека використовується для машинного навчання, зокрема для лінійної та логістичної регресії.

- `scipy.stats`: Ця бібліотека використовується для статистичного аналізу даних.

## 2. Вхідні дані:

```
# Вхідні дані за 2021, 2022 і 2023 роки
# 1 показник - Грошові кошти
# 2 показник - Забов'язання
# 3 показник - Чистий прибуток
# 4 показник - Середньорічна сума активів
# 5 показник - Валовий прибуток
# 6 показник - Дохід від реалізації
# 7 показник - Виробничі запаси
# 8 показник - Валюта балансу
data_2021 = (139820, 4018976, 113210, 2116373.5, 3236826, 8390610, 337,
4232747)
data_2022 = (137732, 3325496, 33962, 1826238.5, 3549851, 18388104, 446,
3652477)
data_2023 = (791591, 4919512, 119062, 2429083.5, 4663963, 25463875, 1035,
4858167)
```

Ці рядки коду містять дані про фінансові показники компанії за 2021, 2022 та 2023 роки. Дані представлені у вигляді кортежів, де кожен елемент відповідає певному показнику.

## 3. Розрахунок коефіцієнтів:

```
# Розрахунок коефіцієнтів для кожного року
coef_2021 = [
    data_2021[0] / data_2021[1],
    data_2021[7] / data_2021[1],
    data_2021[2] / data_2021[3],
    data_2021[4] / data_2021[5],
    data_2021[6] / data_2021[5],
    data_2021[5] / data_2021[7]
]

coef_2022 = [
    data_2022[i] / data_2022[j] for i, j in [(0, 1), (7, 1), (2, 3), (4, 5), (6, 5), (5, 7)]
]

coef_2023 = [
    data_2023[i] / data_2023[j] for i, j in [(0, 1), (7, 1), (2, 3), (4, 5), (6, 5), (5, 7)]
]
```

Розраховуються фінансові коефіцієнти для кожного року. Коефіцієнти розраховуються як співвідношення між різними фінансовими показниками.

Назви коефіцієнтів:

- Відношення грошових коштів до зобов'язань
- Відношення валюти балансу до зобов'язань
- Відношення чистого прибутку до середньорічної суми активів
- Відношення прибутку до доходу від реалізації
- Відношення виробничих запасів до доходу від реалізації
- Відношення доходу від реалізації до основного капіталу

4. Виведення розрахованих коефіцієнтів:

```
# Виведення розрахованих коефіцієнтів для кожного року
coef_names = [
    "Відношення грошових коштів до зобов'язань",
    "Відношення валюти балансу до зобов'язань",
    "Відношення чистого прибутку до середньорічної суми активів",
    "Відношення прибутку до доходу від реалізації",
    "Відношення виробничих запасів до доходу від реалізації",
    "Відношення доходу від реалізації до основного капіталу"
]

print("Розраховані коефіцієнти для кожного року:")
print("Коефіцієнти 2021 року:")
for name, coef in zip(coef_names, coef_2021):
    print(f"{name}: {coef}")
print("\nКоефіцієнти 2022 року:")
for name, coef in zip(coef_names, coef_2022):
    print(f"{name}: {coef}")
print("\nКоефіцієнти 2023 року:")
for name, coef in zip(coef_names, coef_2023):
    print(f"{name}: {coef}")
```

Вивід розрахованих коефіцієнтів для кожного року.



Вводиться список `coef_names`, який містить назви відповідних коефіцієнтів. Цикл `for` використовується для друку назви та значення кожного коефіцієнта.

```
Розраховані коефіцієнти для кожного року:
Коефіцієнти 2021 року:
Відношення грошових коштів до зобов'язань: 0.03478995644661725
Відношення валюти балансу до зобов'язань: 1.0531904146727922
Відношення чистого прибутку до середньорічної суми активів: 0.05349244828476637
Відношення прибутку до доходу від реалізації: 0.38576766170755167
Відношення виробничих запасів до доходу від реалізації: 4.016394517204351e-05
Відношення доходу від реалізації до основного капіталу: 1.982308415787667

Коефіцієнти 2022 року:
Відношення грошових коштів до зобов'язань: 0.04141697960244126
Відношення валюти балансу до зобов'язань: 1.0983254828753366
Відношення чистого прибутку до середньорічної суми активів: 0.01859669479095967
Відношення прибутку до доходу від реалізації: 0.1930514967720435
Відношення виробничих запасів до доходу від реалізації: 2.4254811697823767e-05
Відношення доходу від реалізації до основного капіталу: 5.03442020305672

Коефіцієнти 2023 року:
Відношення грошових коштів до зобов'язань: 0.16090843969889695
Відношення валюти балансу до зобов'язань: 0.9875302672297578
Відношення чистого прибутку до середньорічної суми активів: 0.04901519441386021
Відношення прибутку до доходу від реалізації: 0.1831599864513944
Відношення виробничих запасів до доходу від реалізації: 4.0645816868014e-05
Відношення доходу від реалізації до основного капіталу: 5.24145732330733
```

Рисунок 3.3. Скріншот з розрахованими коефіцієнтами для 2021, 2022 та 2023 років

## 5. Підготовка даних для моделювання:

```
# Створення X_train
X_train = np.array([coef_2021, coef_2022, coef_2023])

# Задання років
years = np.array([2021, 2022, 2023])

# Мітки класів: 0 - не банкрутство, 1 - банкрутство
y_train = np.array([0, 0, 1])
```

Підготовка даних для навчання моделей машинного навчання:

- `X_train` - це масив, що містить розраховані коефіцієнти для кожного року.

- `years` - це масив, що містить роки, до яких належать дані в `X_train`.
- `y_train` - це масив міток класів, де 0 означає не банкрутство, а 1 - банкрутство. Дані за 2023 рік марковані як 1, що означає, що компанія зазнала банкрутства.

#### 6. Навчання моделей:

```
# Ініціалізація та навчання моделей
logistic_model = LogisticRegression()
logistic_model.fit(X_train, y_train)

linear_model = LinearRegression()
linear_model.fit(X_train, years)
```

#### Навчання двох моделей машинного навчання:

- `logistic_model`: Це модель логістичної регресії, яка використовується для прогнозування ймовірності банкрутства на основі розрахованих коефіцієнтів.
- `linear_model`: Це модель лінійної регресії, яка використовується для прогнозування року на основі розрахованих коефіцієнтів.

Функція `fit` використовується для навчання моделей на тренувальних даних (`X_train` та `y_train`), де `X_train` - це набір фінансових показників за роки, а `y_train` - мітки класів, де 1 відповідає банкрутству, а 0 - не банкрутству.

Також тут створюється та навчається модель лінійної регресії для прогнозування року для 2024 року на основі фінансових показників.

#### 7. Прогнозування ймовірності банкрутства:

```
# Прогнозування ймовірності банкрутства для 2023 року за допомогою
логістичної регресії
X_test = np.array([
    [data_2023[0] / data_2023[1], data_2023[7] / data_2023[1],
     data_2023[2] / data_2023[3], data_2023[4] / data_2023[5],
     data_2023[6] / data_2023[5], data_2023[5] / data_2023[7]]
]) # Дані за 2023 рік
```

```
probability_of_bankruptcy = logistic_model.predict_proba(X_test)[0][1]
print("\nЙмовірність банкрутства для 2023 року:", probability_of_bankruptcy)
```

Тут використовується навчена модель `logistic_model`, щоб передбачити ймовірність банкрутства для 2023 року на основі даних `X_test`:

Створюється масив `X_test`, що містить розраховані коефіцієнти для 2023 року. Функція `predict_proba` моделі логістичної регресії використовується для прогнозування ймовірності того, що значення мітки класу буде 1 (тобто банкрутство) для даних у `X_test`. Результат зберігається в змінній `probability_of_bankruptcy` та друкується.

```
Ймовірність банкрутства для 2023 року: 0.44944840582447226
```

Рисунок 3.4. Скріншот ймовірності банкрутства для 2023 року

#### 8. Прогнозування року за даними 2023:

```
# Прогнозування року для 2024 року за допомогою лінійної регресії
predicted_year_2024 = linear_model.predict(X_test)[0]
print("Прогнозований рік для 2024 року:", int(predicted_year_2024))
```

Тут використовується навчена модель лінійної регресії для прогнозування року для 2024 року на основі даних `X_test`.

Прогнозується рік на основі даних за 2023 рік. Функція `predict` моделі лінійної регресії використовується для прогнозування значення на основі даних у `X_test`. Оскільки модель навчалася прогнозувати роки, отримане значення відповідає прогнозованому року. Результат зберігається в змінній `predicted_year_2024` та друкується після округлення до цілого числа.

```
Прогнозований рік для 2024 року: 2023
```

Рисунок 3.5. Скріншот прогнозування року за даними 2023

#### 9. Обчислення лінійної регресії для прогнозування ймовірності банкрутства:

```
# Обчислення лінійної регресії для прогнозування ймовірності банкрутства
slope, intercept, _, _, _ = linregress(years,
logistic_model.predict_proba(X_train)[:, 1])
```

Обчислення коефіцієнтів лінійної регресії для зв'язку між роками (years) та прогнозованою ймовірністю банкрутства

(`logistic_model.predict_proba(X_train)[:, 1]`). Функція `linregress` з бібліотеки `scipy.stats` використовується для обчислення цих коефіцієнтів.

Збережені значення:

- `slope`: кутовий коефіцієнт лінії регресії
- `intercept`: точка перетину лінії регресії з віссю Y

10. Прогнозування ймовірності банкрутства для 2024 року на основі лінійної регресії:

```
# Прогнозування ймовірності банкрутства для 2024 року
predicted_probability_2024 = slope * predicted_year_2024 + intercept
print("Прогнозована ймовірність банкрутства для 2024 року:",
predicted_probability_2024)
```

Тут використовується оновлена модель логістичної регресії для прогнозування ймовірності банкрутства для 2024 року на основі оновлених фінансових даних `X_test_modified`.

Використання раніше обчислених коефіцієнтів лінійної регресії (`slope` та `intercept`) для прогнозування ймовірності банкрутства на 2024 рік. `predicted_year_2024` - це прогнозований рік, отриманий раніше. Результат зберігається в змінній `predicted_probability_2024` та друкується.

```
Прогнозована ймовірність банкрутства для 2024 року: 0.49264738412995257
```

Рисунок 3.6. Скріншот ймовірності банкрутства для 2024 року

11. Визначення критичних параметрів:

```
# Критичні параметри, які найбільше впливають на ймовірність
банкрутства
parameters = coef_names
coefficients = logistic_model.coef_[0]
critical_params = sorted(zip(parameters, coefficients), key=lambda x: abs(x[1]),
```

```
reverse=True)

# Вивід критичних параметрів та їх впливу
print("\nКритичні параметри, які найбільше впливають на ймовірність
банкрутства:")
for param, coef in critical_params:
    print(f"{param}: {coef}")
```

Визначаються критичні параметри, які найбільше впливають на ймовірність банкрутства. `coef_names` містить назви параметрів (коефіцієнтів). `logistic_model.coef_[0]` містить ваги моделі логістичної регресії для кожного параметра. `sorted` використовується для сортування пари (назва параметра, вага) за абсолютним значенням ваги в порядку спадання (`reverse=True`). Отриманий список `critical_params` містить пари назва-вага для критичних параметрів. Цикл `for` друкує назву та вагу кожного критичного параметра.

```
Критичні параметри, які найбільше впливають на ймовірність банкрутства:
Відношення доходу від реалізації до основного капіталу: 0.5132473938467106
Відношення грошових коштів до зобов'язань: 0.06664452993344192
Відношення валюти балансу до зобов'язань: -0.05507603103317839
Відношення прибутку до доходу від реалізації: -0.030651448709568818
Відношення чистого прибутку до середньорічної суми активів: 0.01218032336260315
Відношення виробничих запасів до доходу від реалізації: 6.942106062094543e-06
```

Рисунок 3.7. Скріншот критичних параметрів

12. Ітеративний підбір параметрів для зниження ймовірності банкрутства

```
# Ітеративний підбір параметрів для зниження ймовірності банкрутства
# Використовується простий підбір коефіцієнтів для перших трьох
критичних параметрів
target_probability = probability_of_bankruptcy_2023
max_iterations = 1000
tolerance = 1e-6

X_test_modified = X_test.copy()
critical_param_indices = [parameters.index(param[0]) for param in
```

```
critical_params[:3]]

for i in range(max_iterations):
    probability = logistic_model.predict_proba(X_test_modified)[0][1]
    error = target_probability - probability
    if abs(error) < tolerance:
        break
    for index in critical_param_indices:
        # Регулюємо параметр, використовуючи коефіцієнт пропорційності
        X_test_modified[0, index] += error * 0.01 / len(critical_param_indices)
```

Цей фрагмент коду описує процес ітеративного підбору параметрів моделі логістичної регресії з метою зниження ймовірності банкрутства. Він використовує простий метод підбору коефіцієнтів для перших трьох критичних параметрів.

#### 1. Визначення змінних:

- `target_probability`: Ця змінна зберігає цільове значення ймовірності банкрутства, яке ми хочемо досягти. У даному випадку це ймовірність банкрутства для 2023 року, отримана з моделі.
- `max_iterations`: Ця змінна визначає максимальну кількість ітерацій, які буде виконано алгоритмом підбору параметрів.
- `tolerance`: Ця змінна визначає допустиму похибку між цільовою ймовірністю та прогнозованою ймовірністю.
- `X_test_modified`: Ця змінна є копією масиву `X_test`, який використовується для прогнозування ймовірності банкрутства. Її буде модифікувати алгоритм підбору параметрів.
- `critical_param_indices`: Цей список містить індекси перших трьох критичних параметрів, які будуть піддаватися регулюванню.

#### 2. Цикл ітерацій:

- Цикл `for` виконується до `max_iterations` разів або до тих пір, поки похибка між цільовою ймовірністю та прогнозованою ймовірністю не стане меншою за `tolerance`.
- В кожній ітерації:

- Виконується прогнозування ймовірності банкрутства за допомогою модифікованого масиву `X_test_modified` та моделі логістичної регресії.
- Розраховується похибка між цільовою ймовірністю та прогнозованою ймовірністю.
- Якщо похибка менша за `tolerance`, цикл завершується.
- В іншому випадку, для кожного з критичних параметрів:
  - Параметр регулюється шляхом додавання до нього значення, пропорційного похибці та кількості критичних параметрів.

### 13. Зміна параметрів для уникнення банкрутства:

```
# Зміна параметрів для уникнення банкрутства
max_values = {parameters[index]: X_test_modified[0, index] for index in
critical_param_indices}
print("\nДля уникнення банкрутства потрібно змінити:")
for param, value in max_values.items():
    print(f"{param} на {value:.6f}.")

predicted_probability_2024_modified =
logistic_model.predict_proba(X_test_modified)[0][1]
print("\nПрогнозована ймовірність банкрутства для 2024 року з оновленими
даними:", predicted_probability_2024_modified)
```

Демонстрування, як можна змінити критичні параметри, щоб знизити ймовірність банкрутства. `max_values` - це словник, який містить пари назва-значення для змінених критичних параметрів. Друкується список параметрів, які потрібно змінити, та їх цільові значення. Створюється масив `X_test_modified` з оновленими значеннями критичних параметрів. Ймовірність банкрутства для 2024 року прогнозується з використанням оновлених даних. Результат зберігається в змінній `predicted_probability_2024_modified` та друкується.

```

Для уникнення банкрутства потрібно змінити:
Відношення доходу від реалізації до основного капіталу на 5.241457.
Відношення грошових коштів до зобов'язань на 0.160908.
Відношення валюти балансу до зобов'язань на 0.987530.

Прогнозована ймовірність банкрутства для 2024 року з оновленими даними: 0.44944840582447226

```

Рисунок 3.8. Скріншот параметрів, які треба змінити та оновленої ймовірності банкрутства на 2024 рік

#### 14. Візуалізація даних:

```

# Графіки для кожного параметра
fig1, axs1 = plt.subplots(1, 2, figsize=(12, 6))
fig2, axs2 = plt.subplots(1, 2, figsize=(12, 6))
fig3, axs3 = plt.subplots(1, 2, figsize=(12, 6))
axs = axs1.flatten().tolist() + axs2.flatten().tolist() + axs3.flatten().tolist()

for i, (param, coef) in enumerate(critical_params[:6]):
    axs[i].plot(years, X_train[:, i], marker='o')
    axs[i].set_title(param)
    axs[i].set_xlabel('Рік')
    axs[i].set_ylabel('Значення параметра')
    axs[i].grid(True)

# Прогноз ймовірності банкрутства для 2024 року для кожного параметра
X_test_param = np.array([[0.0] * len(parameters)])
X_test_param[0, i] = 1.0
probability_of_bankruptcy =
logistic_model.predict_proba(X_test_param)[0][1]
axs[i].scatter(2024, probability_of_bankruptcy, color='red')

# З'єднання точки для 2024 року лінією
axs[i].plot([years[-1], 2024], [X_train[-1, i], probability_of_bankruptcy],
color='red', linestyle='--')

plt.tight_layout()

# Графік ймовірності банкрутства на проміжку з 2021 по 2024
plt.figure(figsize=(8, 6))

# Графік для існуючих даних
probabilities_existing = logistic_model.predict_proba(X_train)[:, 1]
plt.plot(years, probabilities_existing, marker='o', label='Ймовірність
банкрутства (існуючі дані)', color='blue')

```



```

# Прогнозована ймовірність банкрутства для 2024 року
plt.scatter(2024, predicted_probability_2024, color='red',
            label='Прогноз ймовірності банкрутства (2024)')
plt.plot([years[-1], 2024], [probabilities_existing[-1],
predicted_probability_2024], linestyle='--',
         color='red')

# Прогнозована ймовірність банкрутства для 2024 року з оновленими
даними
plt.scatter(2024, predicted_probability_2024_modified, color='purple',
            label='Прогноз ймовірності банкрутства (зі зміненими даними)')
plt.plot([years[-1], 2024], [probabilities_existing[-1],
predicted_probability_2024_modified],
         linestyle='--', color='purple')

plt.xlabel('Рік')
plt.ylabel('Ймовірність банкрутства')
plt.title('Прогноз ймовірності банкрутства на проміжку з 2021 по 2024')
plt.legend()
plt.grid(True)

# Додати відступ від країв
plt.subplots_adjust(left=0.1, right=0.9, top=0.9, bottom=0.1)

plt.show()

```

Ці рядки коду створюють декілька графіків для візуалізації даних та прогнозів.

Графіки для кожного параметра:

- Кожен графік складається з 2 підграфіків, розміщених на фігурі розміром 12x6.
  - Відображається динаміка одного з критичних параметрів протягом років з 2021 по 2024.
  - Використовуються маркери 'o' для позначення фактичних значень параметра за кожен рік.

○ Додатково відображається прогнозована ймовірність банкрутства на 2024 рік, якби даний параметр мав максимальне значення. Це дозволяє візуально оцінити вплив цього параметра на ймовірність банкрутства.

Графік ймовірності банкрутства на проміжку з 2021 по 2024:

- Цей графік розміщений на фігурі розміром 8x6.
- Відображає динаміку ймовірності банкрутства протягом років з 2021 по 2024.
  - Використовуються маркери 'o' для позначення фактичних значень ймовірності банкрутства за кожен рік.
    - Додатково відображаються:
      - Прогнозована ймовірність банкрутства на 2024 рік (червона точка).
      - Прогнозована ймовірність банкрутства на 2024 рік з урахуванням змінених критичних параметрів (фіолетова точка).
    - Графік має заголовок "Прогноз ймовірності банкрутства на проміжку з 2021 по 2024", вісь x з позначенням років та вісь y з позначенням ймовірності банкрутства.
      - Легенда допомагає розрізнити різні лінії та точки.
      - На графіку використовується сітка для покращення візуального сприйняття.

Функції, що використовуються для побудови графіків:

- `plt.subplots` - створює фігуру та підграфіки.
- `axs.flatten().tolist()` - перетворює список підграфіків у плоский список.
- `plt.plot` - будує лінії на графіку.
- `plt.scatter` - будує точки на графіку.
- `plt.set_title`, `plt.set_xlabel`, `plt.set_ylabel` - задають назви осей та заголовка графіка.
- `plt.grid` - додає сітку на графік.

- `plt.tight_layout` - автоматично підбирає розміри та розташування графіків.
- `plt.legend` - додає легенду до графіка.
- `plt.subplots_adjust` - налаштовує відступи від країв графіка.
- `plt.show` - візуалізує графіки.

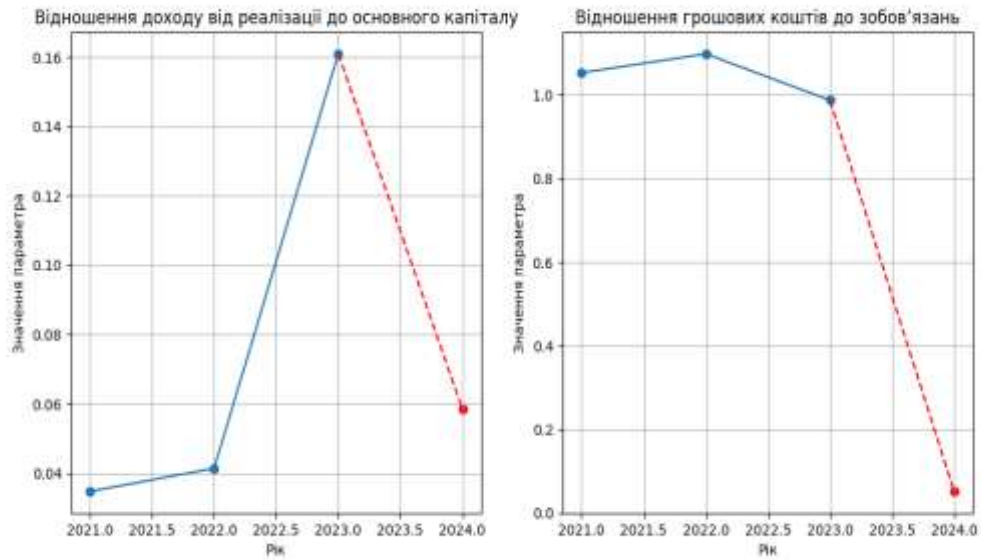


Рисунок 3.9. Тенденції першого та другого критичного параметру

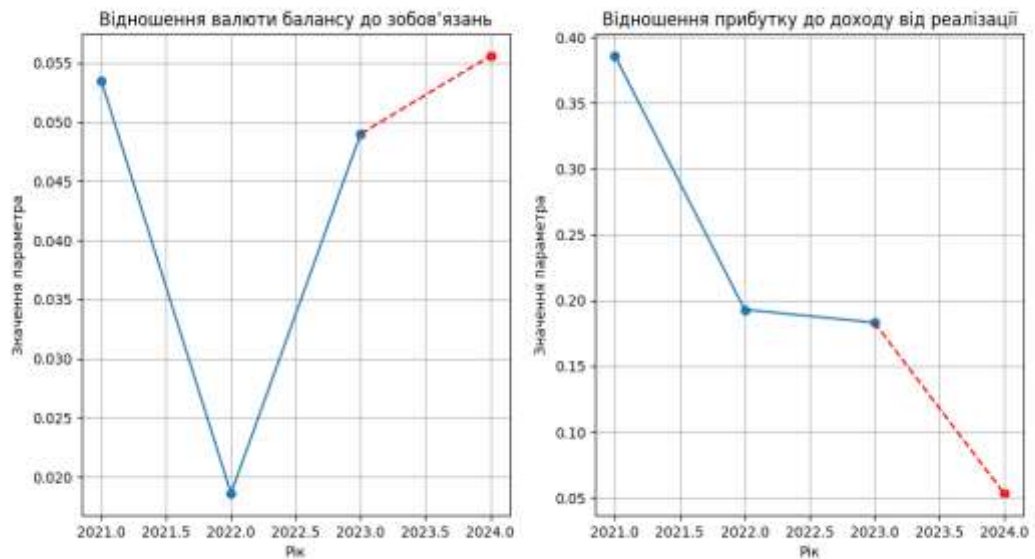


Рисунок 3.10. Тенденції третього та четвертого критичного параметру

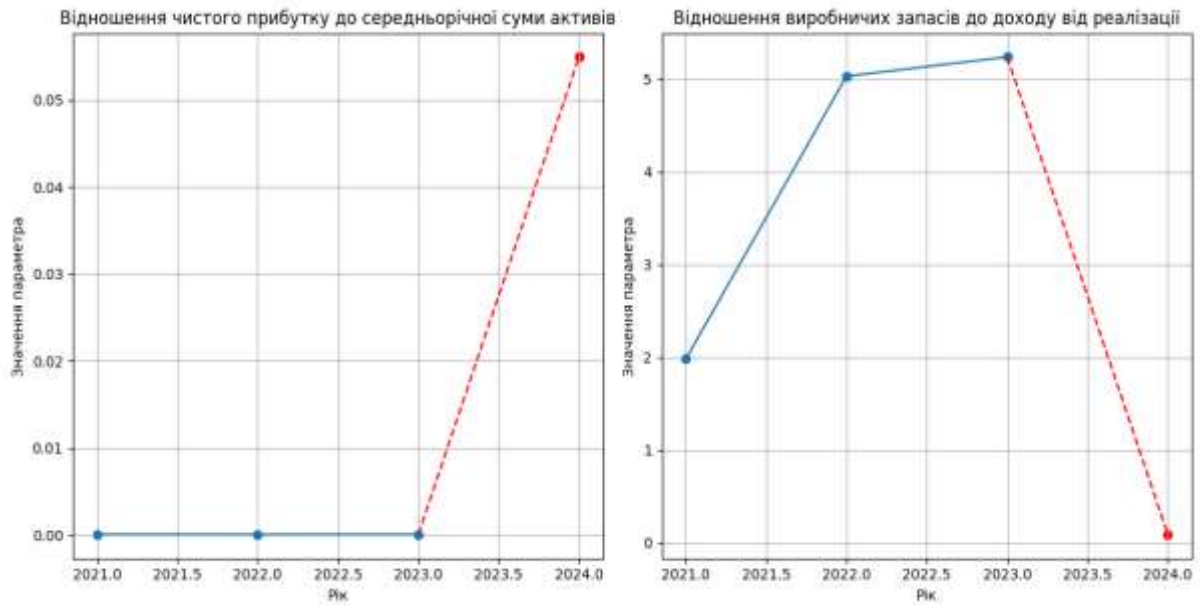


Рисунок 3.11. Тенденція п'ятого та шостого критичного параметру

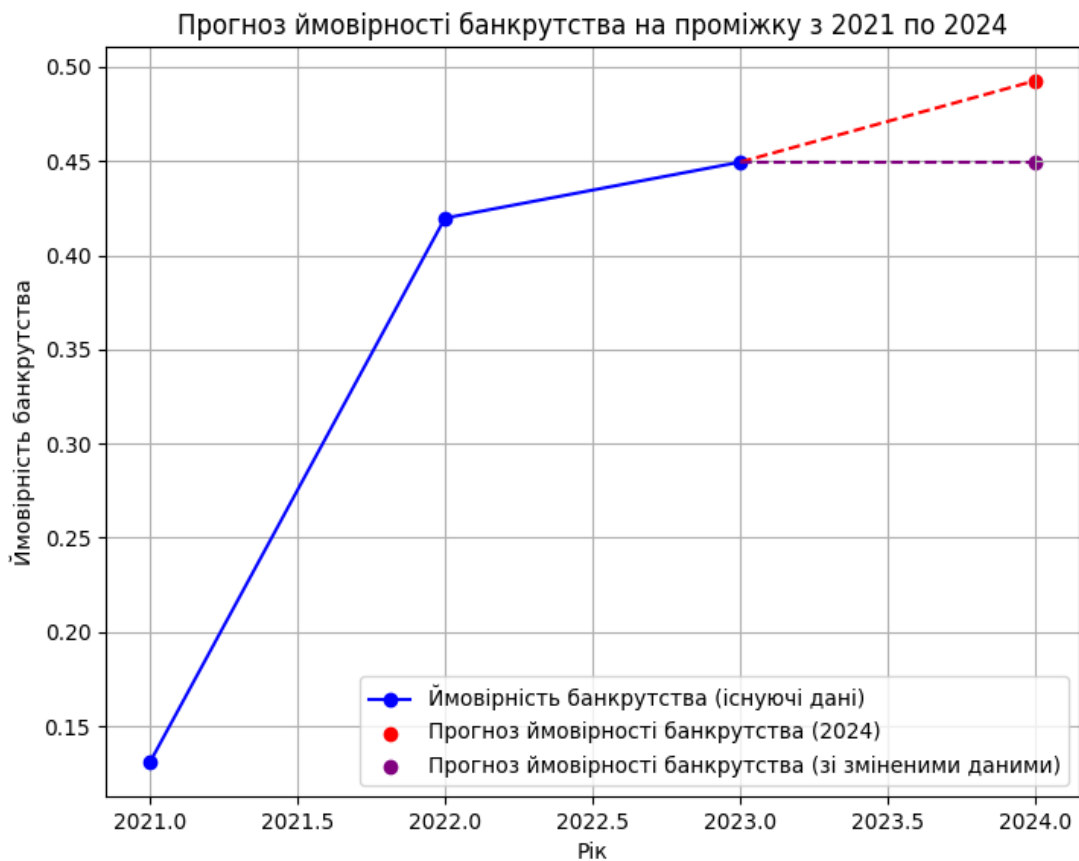


Рисунок 3.12. Ймовірність банкрутства з 2021 по 2024 роки

Підсумовуючи усю інформацію з розділу можемо сказати, що у останньому розділі було описано процес проектування та розробки програми, яка використовує технології машинного навчання.

Була описана логіка програми, описаний її алгоритм, розглянуті основні функції, які представляють собою логіку програми та реалізацію алгоритму у цілому.

## ВИСНОВОК

У даній роботі було проведено глибокий аналіз теоретичних аспектів машинного навчання та створення штучного інтелекту з метою їхнього застосування у сфері фінансового управління.

Розділ 1 присвячений дослідженню основних концепцій машинного навчання, включаючи типи, види та принципи роботи. Важливий акцент було зроблено на регресії у машинному навчанні, яка є одним із ключових інструментів для розв'язання задач прогнозування та оптимізації.

Розділ 2 є практичною частиною роботи, де розглянуто проєктні рішення з оптимізації фінансової діяльності підприємства на прикладі ТОВ «ROZETKA». Була розроблена Z-модель для визначення ймовірності банкрутства та виявлення проблемних зон управління фінансами, що є критичними для ефективного функціонування підприємства. В цьому контексті великий акцент був зроблений на аналізі задачі та виборі технологій для реалізації розроблених моделей.

Розділ 3 присвячений програмній реалізації розроблених моделей. Він включає вибір технології штучного інтелекту, розробку формальної моделі та практичну реалізацію за допомогою мови програмування Python. Важливим етапом було обґрунтування вибору середовища реалізації та вибір додаткових бібліотек для забезпечення ефективності та точності моделі.

Підсумовуючи описане вище створені модель з програмою наразі мають можливість розраховувати наступні параметри: "Відношення грошових коштів до зобов'язань", "Відношення валюти балансу до зобов'язань", "Відношення чистого прибутку до середньорічної суми активів", "Відношення прибутку до доходу від реалізації", "Відношення виробничих запасів до доходу від реалізації", "Відношення доходу від реалізації до основного капіталу" "Ймовірності банкрутства", має змогу

навчатися за допомогою технології машинного навчання та здатна надавати критичні параметри, які найбільше впливають на ймовірність банкрутства.

У результаті цієї роботи я поглибив свої знання щодо машинного навчання, його типів, способів та підходів, був ознайомлений з можливими його реалізаціями та використанням у різних стежах. Також було показано, що машинне навчання може бути успішно використане для вирішення складних завдань управління фінансами підприємств. Отримані результати можуть бути корисні для управлінських рішень та стратегій в контексті фінансової стабільності та ефективності підприємства. Даний дослід може послужити основою для подальших досліджень у цій області та розвитку нових методів оптимізації фінансових процесів.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Christopher Bishop "Pattern Recognition and Machine Learning", 2005, 534с.
2. Michael Negnevitsky David L. Poole and Alan K. Mackworth, "Artificial Intelligence: A Systems Approach" , 2008, 304с.
3. Machine Learning, ML URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
4. Brownlee, J. (2009). A Gentle Introduction to Natural Language Processing URL: <https://machinelearningmastery.com/natural-language-processing/>
5. Heron Oreljen "Deep Learning for Natural Language Processing", 2018,161с
6. David L. Poole and Alan K. Mackworth "Artificial Intelligence: Foundations of Computational Agents", 2011, 612с.
7. He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep Residual Learning for Image Recognition URL: <https://arxiv.org/abs/1512.03385>
8. Heron Oreljen "Deep Learning for Natural Language Processing", 2018,161с.
9. МАШИИННЕ НАВЧАННЯ ПРОСТИМИ СЛОВАМИ URL: [https://learn.ztu.edu.ua/pluginfile.php/200479/mod\\_resource/content/1/%D0%A1%D0%A8%D0%86%D0%9B-2\\_%D0%9C%D0%9D-1\\_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F.pdf](https://learn.ztu.edu.ua/pluginfile.php/200479/mod_resource/content/1/%D0%A1%D0%A8%D0%86%D0%9B-2_%D0%9C%D0%9D-1_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F.pdf)
10. Christopher Bishop "Pattern Recognition and Machine Learning", 2006, 147с.
11. Sebastian Raschka and Vahid Mirjalili "Python Machine Learning", 2017, 442с.



12. Степаненко О.П. Інтелектуальні системи підтримки управління діяльністю організації. Культура Причорномор'я. 2008. № 140. С. 119 – 122
13. Кошелєва В.С. Комп'ютерні технології як засіб формування проєктувальних умінь розробки бізнес-планів у майбутніх інженерів-педагогів економічного профілю. Проблеми інженерно-педагогічної освіти. 2007. № 16. С. 113–126.
14. Weng, L. (2018). Introduction to Reinforcement Learning: An Overview / Lilian Weng URL: <https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html>
15. Chollet, F. (2019). Getting Started with Keras URL: [https://keras.io/guides/sequential\\_model/](https://keras.io/guides/sequential_model/)
16. Francois Chollet "Deep Learning with Python", 2017, 192с.
17. Ситник В.Ф. Системи підтримки прийняття рішень : навч. посібник. Київ : КНЕУ, 2004. 614 с.
18. Єфіменко С. М., Степашко В. С. Прогнозна аналітика як ефективний інструмент підтримки рішень у системах цифрової економіки. Управляючі системи та машини. 2018. № 6.
19. Балацький О.Ф., Теліженко О.М., Соколов М.О. Управління інвестиціями : навч. посібник для студ. вищих навч. закл. Суми : Університетська книга, 2004. 231 с.
20. Фінансова звітність ТОВ «РОЗЕТКА». Електронний ресурс. Режим доступу: [https://clarity-project.info/edr/37193071/finances?current\\_year=2023](https://clarity-project.info/edr/37193071/finances?current_year=2023)
21. Шевченко, Ю. О. (2022). Обробка і аналіз даних з використанням електронних таблиць. Частина II «Аналіз даних та макроси». <https://ir.nmu.org.ua/handle/123456789/162624>

## ДОДАТКИ

## Додаток А. Відомість матеріалів кваліфікаційної роботи

## Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки			
1										
2					Документація					
3										
4					Пояснювальна записка		Формат А4			
5										
6					Демонстраційний матеріал		Презентація на CD-R			
7										
8					Копія роботи		Диск CD-R			
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
Змін.	Аркуш	№ докум.	Підпис	Дата						
Розроб.					<b>Матеріали кваліфікаційної роботи</b>	Літ.	Аркуш	Аркушів	НТУ «ДП», 12; 124-20-2	
К. розд.										
Керівн.										
Н.контр.										
Зав. каф.										

## Додаток Б. Відгук керівника кваліфікаційної роботи

### Відгук на кваліфікаційну роботу бакалавра студента(ки) групи 124 – 20 – 2 спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: \_\_\_\_\_

Обсяг кваліфікаційної роботи \_\_\_\_\_ стор.

Мета кваліфікаційної роботи: \_\_\_\_\_

Актуальність теми \_\_\_\_\_

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки \_\_\_\_\_

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра. Оригінальність наукових рішень полягає в \_\_\_\_\_

Практичне значення результатів кваліфікаційної роботи полягає в \_\_\_\_\_

Висновки підтверджують можливість використання результатів роботи в \_\_\_\_\_

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі (*в разі невідповідності – вказати*)

У роботі відзначено такі недоліки: \_\_\_\_\_

Кваліфікаційна робота в цілому заслуговує оцінки: \_\_\_\_\_

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,  
науковий ступінь, вчене звання, посада \_\_\_\_\_ / ПІБ

## Додаток В. Рецензія

### Рецензія

**на кваліфікаційну роботу бакалавра**  
студента(ки) групи 124 – 20 – 2  
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи:

---



---

Обсяг кваліфікаційної роботи: \_\_\_\_\_  
Висновок про відповідність кваліфікаційної роботи завданню та освітньо-професійній програмі спеціальності \_\_\_\_\_

Загальна характеристика кваліфікаційної роботи, ступінь використання нормативно-методичної літератури та передового досвіду

---



---

Позитивні сторони кваліфікаційної роботи:

---



---

Основні недоліки кваліфікаційної роботи:

---



---



---

Кваліфікаційна робота в цілому заслуговує оцінки: \_\_\_\_\_

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Рецензент,

науковий ступінь, вчене звання, посада \_\_\_\_\_ /

ПІБ

## Додаток Г. Фінансові дані підприємства ROZETKA

Таблиця Г.1

## J0900108 Баланс (Звіт про фінансовий стан)( Актив)

Назва рядка	Код рядка	На початок звітнього періоду, тис. грн	На кінець звітнього періоду, тис. грн
I. Необоротні активи Нематеріальні активи	1000	219.00	27.00
первісна вартість	1001	1 384.00	1 241.00
накопичена амортизація	1002	-1 165.00	-1 214.00
Незавершені капітальні інвестиції	1005	160.00	160.00
Основні засоби	1010	62 180.00	72 045.00
первісна вартість	1011	138 578.00	174 099.00
знос	1012	-76 398.00	-102 054.00
Відстрочені податкові активи	1045	20 179.00	12 863.00
Усього за розділом I	1095	82 738.00	85 095.00
II. Оборотні активи Запаси	1100	2 159 188.00	2 505 961.00
Виробничі запаси	1101	446.00	1 035.00
Товари	1104	2 158 742.00	2 504 926.00
Дебіторська заборгованість за продукцію, товари, роботи, послуги	1125	878 734.00	1 286 662.00
Дебіторська заборгованість за розрахунками: за виданими авансами	1130	279 458.00	53 271.00
з бюджетом	1135	62 903.00	65 723.00
Інша поточна дебіторська заборгованість	1155	8.00	551.00
Гроші та їх еквіваленти	1165	137 732.00	791 591.00
Рахунки в банках	1167	137 732.00	791 591.00
Інші оборотні активи	1190	41 538.00	69 313.00
Усього за розділом II	1195	3 559 561.00	4 773 072.00
Баланс	1300	3 642 299.00	4 858 167.00

**Ю900108 Баланс (Звіт про фінансовий стан)( Пасив)**

Назва рядка	Код рядка	На початок звітнього періоду, тис. грн	На кінець звітнього періоду, тис. грн
I. Власний капітал Зареєстрований (пайовий) капітал	1400	1.00	1.00
Нерозподілений прибуток (непокритий збиток)	1420	332 832.00	344 848.00
Усього за розділом I	1495	332 833.00	344 849.00
Інші довгострокові зобов'язання	1515	27 184.00	61 345.00
Усього за розділом II	1595	27 184.00	61 345.00
III. Поточні зобов'язання і забезпечення Короткострокові кредити банків	1600	74 000.00	
Поточна кредиторська заборгованість за: довгостроковими зобов'язаннями	1610	50 461.00	51 347.00
товари, роботи, послуги	1615	2 617 180.00	4 058 034.00
розрахунками з бюджетом	1620	122 400.00	81 050.00
у тому числі з податку на прибуток	1621	37 401.00	25 772.00
розрахунками зі страхування	1625	5 146.00	5 930.00
розрахунками з оплати праці	1630	44 695.00	70 698.00
за одержаними авансами	1635	127 400.00	184 914.00
Інші поточні зобов'язання	1690	241 000.00	
Усього за розділом III	1695	3 282 282.00	4 451 973.00
Баланс	1900	3 642 299.00	4 858 167.00

**Ю900207 Звіт про фінансові результати (Звіт про сукупний дохід)(  
Фінансові результати)**

Назва рядка	Код рядка	За звітний період, тис. грн	За аналогічний період попереднього року, тис. грн
Чистий дохід від реалізації продукції (товарів, робіт, послуг)	2000	25 463 875.00	18 388 104.00
Собівартість реалізованої продукції (товарів, робіт, послуг)	2050	20 799 912.00	14 838 253.00
Валовий: прибуток	2090	4 663 963.00	3 549 851.00
Інші операційні доходи	2120	31 018.00	38 203.00
Адміністративні витрати	2130	181 245.00	167 524.00
Витрати на збут	2150	4 356 271.00	2 996 497.00
Інші операційні витрати	2180	154 434.00	254 323.00
Фінансовий результат від операційної діяльності: прибуток	2190	3 031.00	169 710.00
Інші фінансові доходи	2220	15 159.00	2 142.00
Інші доходи	2240	30.00	
Фінансові витрати	2250	22 396.00	21 621.00
Інші витрати	2270	-51 254.00	3 346.00
Фінансовий результат до оподаткування: прибуток	2290	47 078.00	146 885.00
Витрати (дохід) з податку на прибуток	2300	-35 062.00	-27 823.00
Чистий фінансовий результат: прибуток	2350	12 016.00	119 062.00

**Ю900108 Баланс (Звіт про фінансовий стан)( Актив)**

Назва рядка	Код рядка	На початок звітнього періоду, тис. грн	На кінець звітнього періоду, тис. грн
I. Необоротні активи Нематеріальні активи	1000	82.00	219.00
первісна вартість	1001	1 047.00	1 384.00
накопичена амортизація	1002	-965.00	-1 165.00
Незавершені капітальні інвестиції	1005	248.00	160.00
Основні засоби	1010	54 531.00	69 186.00
первісна вартість	1011	94 567.00	146 985.00
знос	1012	-40 036.00	-77 799.00
Відстрочені податкові активи	1045	10 267.00	20 179.00
Усього за розділом I	1095	65 128.00	89 744.00
II. Оборотні активи Запаси	1100	2 495 144.00	2 158 662.00
Виробничі запаси	1101	337.00	446.00
Товари	1104	2 494 807.00	2 158 216.00
Дебіторська заборгованість за продукцію, товари, роботи, послуги	1125	1 207 246.00	877 942.00
Дебіторська заборгованість за розрахунками: за виданими авансами	1130	162 098.00	280 669.00
з бюджетом	1135	122 632.00	64 979.00
Інша поточна дебіторська заборгованість	1155	45.00	1 212.00
Гроші та їх еквіваленти	1165	139 820.00	137 732.00
Рахунки в банках	1167	139 820.00	137 732.00
Інші оборотні активи	1190	40 634.00	41 537.00
Усього за розділом II	1195	4 167 619.00	3 562 733.00
Баланс	1300	4 232 747.00	3 652 477.00



**Ю900108 Баланс (Звіт про фінансовий стан)( Пасив)**

Назва рядка	Код рядка	На початок звітнього періоду, тис. грн	На кінець звітнього періоду, тис. грн
I. Власний капітал Зареєстрований (пайовий) капітал	1400	1.00	1.00
Нерозподілений прибуток (непокритий збиток)	1420	213 770.00	326 980.00
Усього за розділом I	1495	213 771.00	326 981.00
Інші довгострокові зобов'язання	1515	30 481.00	30 896.00
Усього за розділом II	1595	30 481.00	30 896.00
III. Поточні зобов'язання і забезпечення Короткострокові кредити банків	1600	74 000.00	
Поточна кредиторська заборгованість за: довгостроковими зобов'язаннями	1610	24 826.00	65 010.00
товари, роботи, послуги	1615	3 624 487.00	2 616 585.00
розрахунками з бюджетом	1620	4 178.00	121 384.00
у тому числі з податку на прибуток	1621	3 055.00	36 386.00
розрахунками зі страхування	1625	1 251.00	5 146.00
розрахунками з оплати праці	1630	9 646.00	44 695.00
за одержаними авансами	1635	154 105.00	126 767.00
Інші поточні зобов'язання	1690	170 002.00	241 013.00
Усього за розділом III	1695	3 988 495.00	3 294 600.00
Баланс	1900	4 232 747.00	3 652 477.00

**Ю900207 Звіт про фінансові результати (Звіт про сукупний дохід)(  
Фінансові результати)**

Назва рядка	Код рядка	За звітний період, тис. грн	За аналогічний період попереднього року, тис. грн
Чистий дохід від реалізації продукції (товарів, робіт, послуг)	2000	18 390 610.00	21 172 647.00
Собівартість реалізованої продукції (товарів, робіт, послуг)	2050	14 840 759.00	17 935 821.00
Валовий: прибуток	2090	3 549 851.00	3 236 826.00
Інші операційні доходи	2120	33 102.00	35 912.00
Адміністративні витрати	2130	168 879.00	90 682.00
Витрати на збут	2150	2 996 510.00	3 122 791.00
Інші операційні витрати	2180	80 139.00	18 499.00
Фінансовий результат від операційної діяльності: прибуток	2190	337 425.00	40 766.00
Інші фінансові доходи	2220	2 142.00	618.00
Інші доходи	2240	1.00	11.00
Фінансові витрати	2250	21 986.00	6 121.00
Інші витрати	2270	177 564.00	1 964.00
Фінансовий результат до оподаткування: прибуток	2290	140 018.00	33 310.00
Витрати (дохід) з податку на прибуток	2300	-26 808.00	652.00
Чистий фінансовий результат: прибуток	2350	113 210.00	33 962.00