МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)
(підпис) Андрій ФОРСЮК (ім'я та прізвище)
«13» грудня 2024р.

«До захисту допущено»
Завідувач кафедри
(підпис) Сергій ГРИБКОВ (ім'я та прізвище)
«13» грудня 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)
освітньо-професійної програми Управління інформацією та аналітика даних
на тему: Дослідження методів машинного навчання для прогнозування кредитоспроможності клієнтів банку

Виконав: здобувач 2 курсу, групи КН-2-4М
Будаков Іван Миколайович (прізвище, ім'я, по батькові повністю) (підпис)

Керівник Грама Михайло Петрович (прізвище , ім'я та по батькові повністю) (підпис)

Консультанти (ім'я та прізвище) (підпис)
(ім'я та прізвище) (підпис)

Рецензент Володимир ДРУЖИНІН (ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач (підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) __Автоматизації і комп'ютерних систем_____
Кафедра __Інформаційних технологій, штучного інтелекту і кібербезпеки___
Освітній ступінь__магістр_____
Спеціальність__122 «Комп'ютерні науки»_____
(код і назва)
Освітньо-професійна програма _Управління інформацією і аналітика даних
(назва)

ЗАТВЕРДЖУЮ
Завідувач
кафедри Інформаційних технологій, штучного інтелекту і кібербезпеки
_____Грибков С.В.
"07" жовтня 2024року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Будакова Івана Миколайовича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів машинного навчання для прогнозування кредитоспроможності клієнтів банку
керівник роботи Грама Михайло Петрович, доктор філософії, старший викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від 7 жовтня 2024 року №884-кс
2. Строк подання здобувачем роботи 4 грудня 2024 року
3. Вихідні дані до роботи Інструменти: Python(бібліотеки Scikit-learn, Pandas, Matplotlib,TensorFlow,PowerBI.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Огляд сучасних методів машинного навчання у фінансовому секторі.Аналізвихідних даних і їхня підготовка для побудови моделі.Розробка моделі прогнозування кредитоспроможності з використанням алгоритмів.Розробка дашборду.
5. Перелік графічного матеріалу:
Кількістрисунків-32,кількістьтаблиць–2.

6.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Грама М.П., доктор філософії, старший викладач	07.10.2024	15.10.2024
2	Грама М.П., доктор філософії, старший викладач	16.10.2024	27.10.2024
3	Грама М.П., доктор філософії, старший викладач	28.10.2024	07.11.2024
4	Грама М.П., доктор філософії, старший викладач	08.10.2024	29.11.2024

7. Дата видачі завдання 7 жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд сучасних методів управління ризиками	07.10 – 09.10.2024	Виконано
2.	Опрацювання теоретичних моделей прогнозування	10.10 – 11.10.2024	Виконано
3.	Розробка першого розділу кваліфікаційної роботи	11.10 – 15.10.2024	Виконано
4.	Розробка другого розділу кваліфікаційної роботи	16.10 – 20.10.2024	Виконано
5.	Пошук та підготовка набору даних	21.10 – 25.10.2024	Виконано
6.	Навчання моделей даних	26.10 – 31.10.2024	Виконано
7.	Розробка третього розділу кваліфікаційної роботи	01.11 – 05.11.2024	Виконано
8.	Створення дашборду в Power bi	06.11 – 10.11.2024	Виконано
9.	Розробка четвертого розділу кваліфікаційної роботи	11.11 – 15.11.2024	Виконано
10.	Підготовка презентації	16.11 – 20.11.2024	Виконано
11.	Оформлення кваліфікаційної роботи	21.11 – 06.12.2024	Виконано

Здобувач _____ (підпис) Будаков І.М. (прізвище та ініціали)

Керівник роботи _____ (підпис) Грама М.П. (прізвище та ініціали)

АНОТАЦІЯ

Будаков Іван Миколайович

Дослідження методів машинного навчання для прогнозування кредитоспроможності клієнтів банку

Кваліфікаційна робота: 83 ст., 32 рис, 2 табл., 2 дод., 35 джерел.

Предмет дослідження: в якості предмету досліження обрано методи які використовуються машинним навчанням для прогнозування у відсотковому співвідношені можливості надання клієнту кредиту опираючись на його історію кредитів, дохід та інші важливі показники.

В кваліфікаційній роботі за допомогою методів машинного навчання які працюють на мові програмування Python було досліджено різні підходи для того щоб зробити прогноз можливості надання кредиту банком клієнту в залежності від його показників.

Серед обраних моделей було обрано найбільш точні які за допогою алгоритмів можуть достовірно визначити ризики які понесе банк в разі надання кредиту та відсіяти найменш привабливих клієнтів.

Результатом роботи є дослідження та розробка методів машинного навчання на мові Python, побудова аналітичного даштборду в Power BI, детальний опис кожного з методів, вказані переваги та недоліки кожного з обраних алгоритмів, опис очищення даних.

Ключові слова: АНАЛІТИКА ДАНИХ, МАШИННЕ НАВЧАННЯ, МЕТОДИ ПРОГНОЗУВАННЯ, ВІЗУАЛІЗАЦІЯ, КРЕДИТОСПРОМОЖНІСТЬ.

ABSTRACT

Budakov Ivan Mykolayovych

Research of Machine Learning Methods for Forecasting the Creditworthiness of Bank Customers

Qualification work: 83 p., 32 fig., 2 tab., 2 appendices, 35 sources.

Subject of research: the subject of research was chosen methods used by machine learning to predict the percentage of the possibility of providing a loan to a client based on his credit history, income and other important indicators.

In the qualification work, using machine learning methods working in the Python programming language, various approaches were investigated in order to make a forecast of the possibility of providing a loan by a bank to a client depending on his indicators.

Among the selected models, the most accurate ones were selected, which, with the help of algorithms, can reliably determine the risks that the bank will incur in the event of providing a loan and weed out the least attractive clients.

The result of the work is the research and development of machine learning methods in Python, the construction of an analytical dashboard in Power BI, a detailed description of each of the methods, the advantages and disadvantages of each of the selected algorithms, and a description of data cleaning.

Keywords: DATA ANALYTICS, MACHINE LEARNING, FORECASTING METHODS, VISUALIZATION, CREDIT ABILITY.

ЗМІСТ

Умовні позначення

1. КК — Кредитний коефіцієнт

   Відношення загальної суми кредиту до середнього доходу позичальника.
2. РК — Рівень кредитного ризику

   Ймовірність невиконання позичальником своїх зобов’язань.
3. СКО — Середній коефіцієнт оцінки

   Показник, що враховує середню оцінку платоспроможності клієнта.
4. ДПС — Дефолтний показник сектора

   Процент дефолтних кредитів у певному секторі економіки.
5. ВК — Власний капітал

   Загальна сума власних коштів позичальника, доступних для покриття боргів.
6. СПЗ — Ставка прибутковості за заставою

   Очікуваний дохід від заставного майна у випадку дефолту.
7. ЗК — Загальна кількість

   Сукупна кількість кредитів, виданих у певному періоді.
8. ПК — Поточний коефіцієнт

   Поточний стан фінансової спроможності клієнта, розрахований на основі останніх даних.

ВСТУП

Актуальності теми. Банк – це потужний інструмент, який потрібен кожній державі для забезбечення фінансової стабільності та економічного розвитку.

Банк є одним з тих підприємст яке стикається з великими ризиками на кожному етапі своєї роботи, так як за ключові етапи роботи відповідають люди, а люди є головною причиною через яких можливий витік конфіденційних даних або навіть розповсюдження банківської таємниці, що несе за собою дуже великі ризики для подільшого стабільного та безпечного функціювання банку.

Завдання дослідження. Тому в цій кваліфікаційні роботі було розглянуто методи машинного навчання та їх алгоритми щоб за допомогою них можно було замінити людину та довірити цю справу технологіям.

Чим більший ризик несе банк тим ближчим він стає до банкрутства, а цього не можно допускати.

Потужним інструментом для банку є можливість ефективно рпацювати з позичальниками та рительно перевіряти клієнтів на можливвість повернення кредиту вчасно ц відведені терміни.

Мета дослідження. Кваліфікаційна робота має на меті ретельно дослідити сектор надання кредиту клієнтам використовуючи при цьому новітні технології та алгоритми.

В першому розділі мова йде про те наскільки важливо та ризикована така річ як кредит та чому саме машинне навчання має кращі шанси забезпечити надійність надання цієї послуги.

В другому розділі детально розглянуто всі найбільш потужні та відомі методи машинного навчання на основі яких буде проведено дослідження, детально описано принцип їх роботи иа ие як вони вплинуть на майбутнє банків.

В третьому розділі представлена візуалізація кожного з методів обгрунтований вибір обраних моделей які найбільш привабливі для банківського сектору надання кредитів.

У четвертому розділі представлено побудову дашборду в Power bi.

Наукова новизна. Наукова новизна кваліфікаційної роботи полягає в застосуванні алгоритмів машинного навчання для прогнозування кредитоспроможності клієнтів банку.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Визначення поняття «Банківський ризик»

Кожна людина рано чи пізно стикається з необхідністю приняття складних рішень, складними вони є через те що кожна дія несе за собою наслідки.

В банківській сфері це наслідки можуть призвести до серйозних наслідків, а це означає що при принятті кожно з рішень потрібно оцінювати ризики та наперед прораховувати кроки для мінімізації ризиків та швидкому їх усуненні в разі виявлення.

В сучасному світі де банки всими силами намагаються привернути увагу кожного клієнту в той же час необхідно думати про те чи потрібен саме цей клієнт банку адже кількість не завжди означає якість, кожен банк прагне привернути увагу клієнтів котрі будуть активно користуватися послугами та свертати увагу на акційні пропозиції які надає банк[1].

Рано чи пізно кожен банк стикається з потужними ризиками які впливають на подальше функціювання банку, якшо в банку працюють досвідчені фахівціто вони наперед продумують кожен крок в разі виникнення ситуацій коли подальше функціювання банку опиняється під загрозою, це може відбутись з різних причин починаючи від не поверненням заборгованості клієнтом до грандіозних змін в країна або глобальних змін в усьому світі[2].

1.2. Огляд сучасних методів управління ризиками

Кожен бізнес рано чи пізно стикається з певними ризиками та наслідками від них.

В банківській сфері ризики поділяються на три важливі категорії:

- Кредитний;
- Ринковий(відсотковий, пайовий, валютний, товарний);
- Операційний(спеціальний та загальний);

Ретельну та постійну перевірку на показники ризиків банк має проходити на

постійній основі починаючи від ретельного перегляду кредитноо портфеля позичальника та зікінчуючи самою банківською установою.

Для цього банки наперед мають висунути певні категорії та встановити межі за які не можна виходити для того щоб не спровокувати ризик та не дестабілізувати ситуацію всередині установи[3].

Кожен ризик потребує ретельного моніторингу своєї діядьності контролю та оцінки який буде включати в себе перевірку кожної операції та транзакції.

Це є ключовим етапом в управлінні ризиками та апобіганні інцедентів спроможних довести банк до банкрутства(рис.1.1).

Рисунок 1.1 – Загальна схема побудови системи прогнозування

1.3. Висновки до розділу 1

Цей розділ має а меті ознайомити з такими поняттями як ризик та всі його можливі наслідки, а також показати яким чинос банки мають дбати про оцінку ризиків та чому важливо проводити ретельно перевірку.

Описано ризики з якими банки зустрічаються під час надання кредиту.

В даному розділі були введені поняття кредитного скорингу та ризику.

РОЗДІЛ 2. ОПИС МОДЕЛЕЙ ДЛЯ ПРОГНОЗУЙВАННЯ ДАНИХ

2.1. Лінійна регресія

Метод регресії є доволі потужним для машинного навчання який дає можливість працювати з змінними які взаємозалежать між собою.

Також моделью даних представленої у вигляді парної лінійної регресії можно проводити дослідження змінних котрі не будуть залежати одна від одної на початковому етапі, проте в ході алгоритму їх можно швидко перенаправити в потрібно нам сторону.

Для більшого ознайомлення з цим методом потрібно розуміти його математичну складову.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_n X_n + \epsilon$$

Рисунок 2.1 – Модель лінійної регресії

В цій формулі у – це залежна змінна тобто той обєкт який ми будемо намагатись передбачити.

$X_1 X_n$ – чинники що будуть впливати на поведінку у.

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_n X_n$$

Рисунок 2.2 – Коефіцієнт методу найменших квадратів

2.2. Логістична регресія

Метод логістичної регресії є доволі популярним в роботі з банками та кретдитоспроможністю.

Для того щоб використовувати даний мето потрібно розуміти як він праює на рівня математики.

$$\text{logit}(P(y=1)) = \ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_n X_n$$

Рисунок 2.3. – Логіт-функція

Ця фунція потрібна для того щоб обчислити шанс знаходження вхідного або заданого користувачем значення.

$$P(y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_n X_n)}}$$

Рисунок 2.4 – Ймовірність події

Ця формула допоможе обчислити ймовірність того що подія відбудеться, в результаті обчисленн отримане значення буде в межах від 0 до 1, чим ближче значення наближатиметься до 1 тим більша ймовірність того що подія відбудеться.

$$P(y=1|X) = \sigma(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_n X_n)$$

Рисунок 2.5 – Функція логістичної регресії

Лінійна регресія незалежних змінних та коефіцієнтів, використовується для того щоб перетворити задане число в діапазон від 0 до 1.

2.3. Дерева рішень

Метод дерева рішень – користується великою популярністю особливо його використовують в банківській.

Загалом цей метод спрощуться до використання великої кількості умов та представлених в ролі дерева від яких надходить велика кількість гілок, тобто умов при виконанні яких процес буде переходити до іншого етапу.

Цей метод доволі швидко в порівнянні з іншими справляється з поставленою задачею навіть при обробці великих даних, також доволі легко працює з виявлення та усуненням пропусків в наборі даних.

Рекомендується використовувати цей метод для знаходження залежності між змінними (рис.2.6).

Рисунок 2.6 - Загальний вигляд дерева рішень

Метод дерева рішень допомагає вирішити проблему з класифікацією та

регресією а також є відмінним інструментом для аналізування та аналітики.

В основному цей метод використовують там де необхідно ухвалювати рішення повязані з статистикою(рис.2.7).

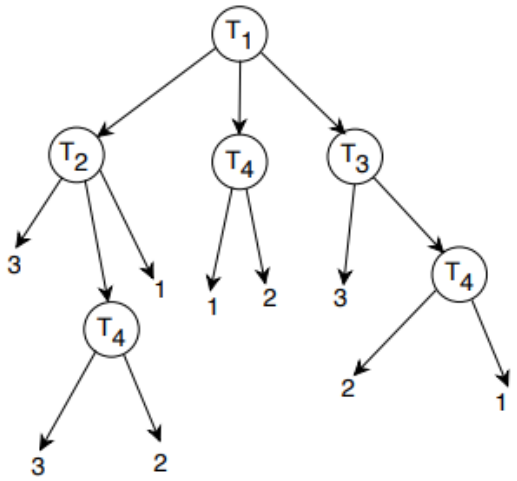


Рисунок 2.7. – Загальний вигляд роботи Дерева рішень

2.4. Випадковий ліс

Метод випадкового лісу має на меті створити набір великої кількості рішень (дерев) даних взятих з зазделегіть обраної підмножини даних.

Для визначення класу до якого належить тестовий об’єкт він обєднує всі можливі комбінації рішень з різних дерев.

Випадковий ліс – це беггінг над вирішальними деревами, під час навчання яких для кожного розбиття ознаки обираються з деякої випадкової підмножини ознак[11].

2.5. Лінійний дискримінантний аналіз

Метод лінійного дискримінативного аналізу добре підходить для можливості точно виявити функції які будуть характеризуватись одночасно на декількох об’єктах класу.

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Рисунок 2.8. – Цільова функція

Формула складається з міжкласової(чисельник) та внутрішньокласової(знаменник) матриці коваріації.

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \text{\му})^T$$

Рисунок 2.9. – Міжкласова дисперсія

Ці формули використовуються для знаходження найкращого вектору в новому просторі класів.

2.6. Нейронні мережі

Метод найренних мереж являє собою набір алгоритмів які можуть розкладати на маленькі шматочки лані для подальшого їх транспортування всередині алгоритму для дослідження поведінки(рис.2.6.).

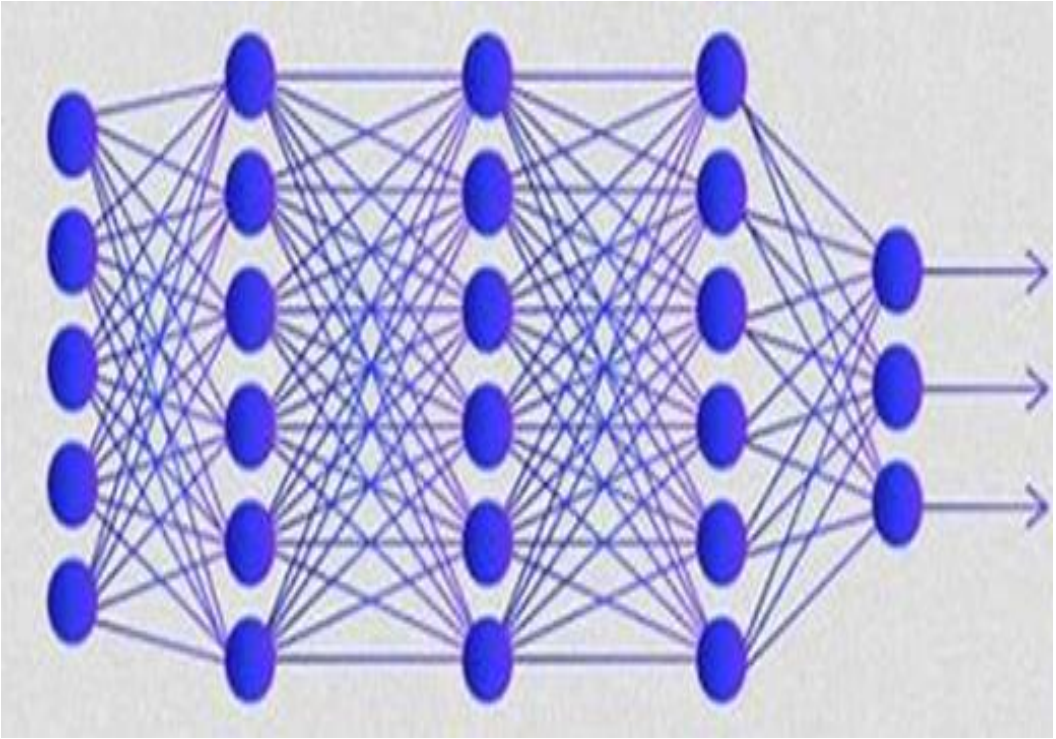


Рисунок 2.10 – Метод нейронних мереж

Під час роботи алгоритму нейроних мереж відбувається зхрещення параметрів які дозволяють відтворювати нові вхідні та вихідні параметри які в свою чергу здатні збільшуватись до заданого алгоритмом розміру.

В сучасному світі нейронні мережі використовують для створення штучних нейронних звязків що здатні замінити фунції мозку людини.

З однієї сторони це важко реалізувати так як для цього потрібно точно знати як працює мозок людини, проте знаючи що інформація зберігається у вигляді великої кількості образів , можно спробувати навчити модель таким чином щоб вона імітувала та виконувала функції близькі до людської поведінки[5].

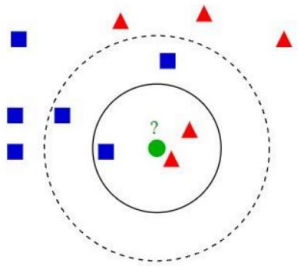
2.7. К-найближчих сусідів

Метод К-найближчих сусідів має на меті на основі точок наданої моделі знайти найбільш подібні данні з іншого набору даних, тим самим допомагає дослідити подібність даних або проаналізувати два набори даних на схожість за допомогою

своїх потужних алгоритмів.

При реалізації KNN першим кроком перетворює точки даних в вектори ознак або їх математичні значення. Потім алгоритм працює, знаходячи відстань між математичними значеннями цих точок.

KNN використовує цю формулу для обчислення відстані між кожною точкою даних і тестовими даними. Потім він знаходить ймовірність того, що ці точки схожі на тестові дані, і класифікує її на основі того, які точки мають найвищі ймовірності.



Риуснок 2.12 – Приклад k-найближчих сусідів

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Рисунок 2.13 – Відстань між точками

Цю формулу використовують для для визначення двох заданих точок(Евклідова відстань).

2.8. Висновки до розділу 2

В даному розділі були роглянуті методи для прогнозування даних, вказані переваги та недоліки кожного методу , описаний загальний підхід до кожного методу та сфери їх застосування .

Для подальшоої роботи було обрано побудову моделі дерева рішень та випадкового лісу, оскільки ці методи мають кілька важливих переваг.

По-перше, дерева рішень є інтуїтивно зрозумілими, візуально інтерпретованими та легко пояснюваними навіть для осіб, які не є фахівцями у сфері машинного навчання. Це важливо для прийняття рішень у банківській сфері, де прозорість алгоритмів має ключове значення.

По-друге, випадковий ліс є більш стійким до переобучення, оскільки базується на ансамблевому методі, що об'єднує результати багатьох дерев рішень. Він добре працює з великим обсягом даних, включаючи випадки, коли дані мають високий рівень шуму або нерівномірний розподіл.

Ці методи є корисними для прогнозування кредитоспроможності клієнтів банку завдяки їхній здатності:

1. Обробляти як числові, так і категоріальні дані, що часто зустрічається у фінансових наборах даних.
2. Розпізнавати важливі фактори, які впливають на кредитоспроможність, автоматично визначаючи найбільш релевантні змінні.
3. Гарно працювати з нерівноваженими наборами даних, що є поширеним у фінансах, коли число клієнтів із низькою кредитоспроможністю значно менше за кількість клієнтів із високою кредитоспроможністю.
4. Забезпечувати високу точність прогнозів завдяки своїй ансамблевій природі (у випадку випадкового лісу), що знижує ймовірність помилок за рахунок середнього по деревам.

Загалом, ці методи є ефективним інструментом для оцінки ризиків і підтримки прийняття рішень у банківській сфері, де важливими є як точність, так і інтерпретація результатів.

РОЗДІЛ 3. РОБОТА З ДАННИМИ

3.1. Опис вхідних даних

В ролі вхідних даних було обрано набір даних що містить дані клієнтів банку розміщені на платформі з відкритими наборами даних Kaggle.

Через свою гибкість та мультизадачність для реалізації було обрано мову програмування Python та його бібліотеки за допомогою яких можно побудувати алгоритм прогнозування та розробити візуаліацію.

3.2. Очистка даних

ETL – це процес збору необроблених даних з окремих джерел, передачі їх у проміжну базу даних для перетворення та завантаження підготовлених даних у єдину цільову систему.

Інструменти ETL використовуються для інтеграції даних для задоволення вимог систем управління реляційними базами даних або традиційних сховищ даних з підтримкою OLAP.

Інструменти та запити OLAP (SQL) вимагають, щоб набори даних були структуровані та стандартизовані за допомогою серії перетворень, які виконуються до того, як дані потраплять до сховища(рис.3.1).

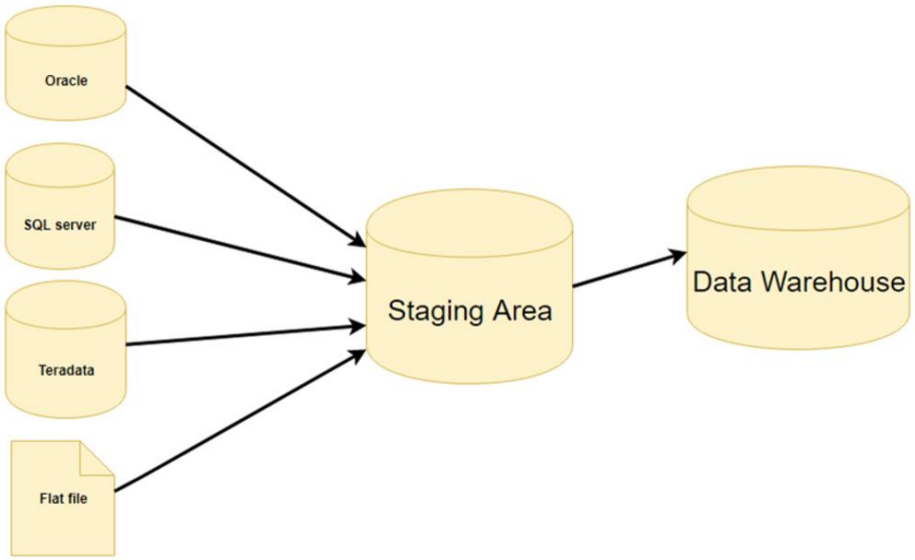


Рисунок 3.1 – ETL процес

Для реалізації ETL процесу було використано Python з бібліотеками (pandas, SQLAlchemy)(рис. 3.2).

```
Перевірка на пропуски:
person_age                      0
person_income                   0
person_emp_length               0
loan_amnt                       0
loan_int_rate                   0
loan_status                     0
loan_percent_income             0
cb_person_default_on_file       0
cb_person_cred_hist_length      0
person_home_ownership_OTHER     0
person_home_ownership_OWN       0
person_home_ownership_RENT      0
loan_intent_EDUCATION           0
loan_intent_HOMEIMPROVEMENT     0
loan_intent_MEDICAL             0
loan_intent_PERSONAL            0
loan_intent_VENTURE             0
loan_grade_B                    0
loan_grade_C                    0
loan_grade_D                    0
loan_grade_E                    0
loan_grade_F                    0
loan_grade_G                    0
dtype: int64
```

Рисунок 3.2 – Результат аналізу пропусків даних

```
Очищені дані:
   person_age  person_income  person_emp_length  loan_amnt  loan_int_rate  \
0          22          59000              123.0      35000          16.02
1          21           9600                5.0       1000          11.14
2          25           9600                1.0       5500          12.87
3          23          65500                4.0      35000          15.23
4          24          54400                8.0      35000          14.27

   loan_status  loan_percent_income  cb_person_default_on_file  \
0            1                 0.59                          1
1            0                 0.10                          0
2            1                 0.57                          0
3            1                 0.53                          0
4            1                 0.55                          1

   cb_person_cred_hist_length  person_home_ownership_OTHER  ...  \
0                           3                        False  ...
1                           2                        False  ...
2                           3                        False  ...
3                           2                        False  ...
4                           4                        False  ...

   loan_intent_HOMEIMPROVEMENT  loan_intent_MEDICAL  loan_intent_PERSONAL  \
0                        False                False                  True
1                        False                False                 False
2                        False                 True                 False
3                        False                 True                 False
4                        False                 True                 False
```

Рисунок 3.3 – Очищені дані

Очищення даних було проведено Google Colab за допомогою Python та його потужник бібліотек.

3.3. Логістична регресія

Для того щоб виконати навчання моделі на основі логістичної регресії набір даних був розкладений на дві частини де 70% були відведені на тестування і 30% для навчання моделі(рис.3.4).

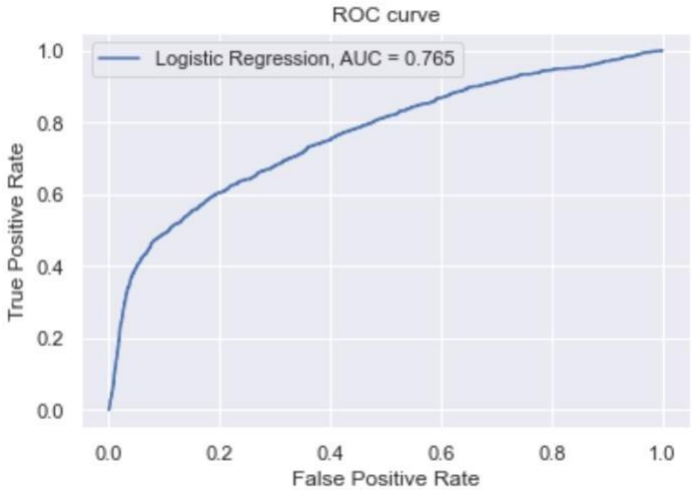


Рисунок 3.5 – Результат виконання алгоритму Логістичної регресії

Після цього була розроблена скорингова карта(рис.3.6).

	Variable	Bin id	Bin	Count	Count (%)	Non-event	Event	Event rate	WoE	IV	JS	Coefficient	Points
0	person_age	0	[-inf, 22.50)	3368	0.148	2502	866	0.257	-0.218	0.007	0.001	-1.244	59.218
1	person_age	1	[22.50, 25.50)	7326	0.321	5738	1588	0.217	0.006	0.000	0.000	-1.244	31.920
2	person_age	2	[25.50, 28.50)	4541	0.199	3573	968	0.213	0.027	0.000	0.000	-1.244	29.321
3	person_age	3	[28.50, 30.50)	2109	0.092	1704	405	0.192	0.158	0.002	0.000	-1.244	13.343
4	person_age	4	[30.50, inf)	5457	0.239	4319	1138	0.209	0.055	0.001	0.000	-1.244	25.925
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3	cb_person_default_on_file_N	3	Missing	0	0.000	0	0	0.000	0.000	0.000	0.000	0.017	32.631
0	cb_person_default_on_file_Y	0	[-inf, 0.50)	18797	0.824	15334	3463	0.184	0.209	0.034	0.004	0.017	32.985
1	cb_person_default_on_file_Y	1	[0.50, inf)	4004	0.176	2502	1502	0.375	-0.769	0.125	0.015	0.017	31.332
2	cb_person_default_on_file_Y	2	Special	0	0.000	0	0	0.000	0.000	0.000	0.000	0.017	32.631
3	cb_person_default_on_file_Y	3	Missing	0	0.000	0	0	0.000	0.000	0.000	0.000	0.017	32.631

83 rows × 13 columns

Рисунок 3.6 – Скорингова карта

Після побудови скорингової карти маємо наступні результати(рис.3.7)

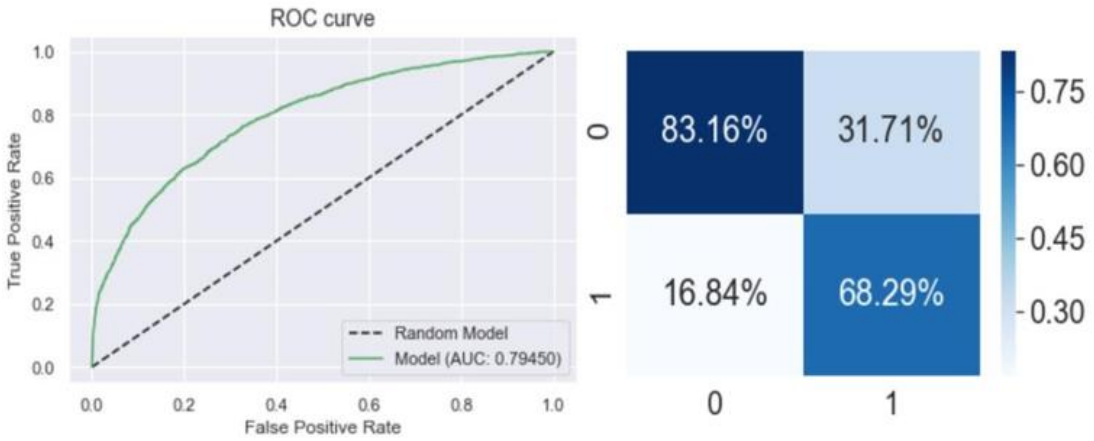


Рисунок. 3.7 - Результат виконання алгоритму Скорингової карти

3.4. Розробка методу дерев рішень

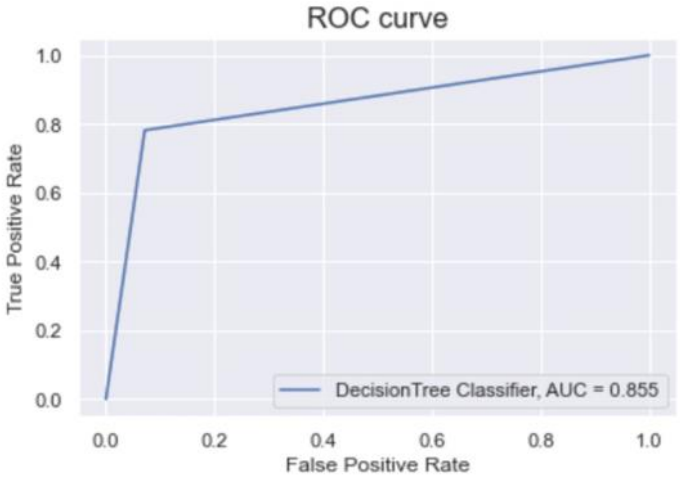


Рисунок 3.8 - Результати роботи моделі дерева рішень

Як ми можемо побачити з наведеного вище рисунку результат роботи дерева рішень дав позитивний результат. Модель побудована на основі цього методу розпізнає потенційно привабливих клієнтів банку з 93,9%(рис.3.9).

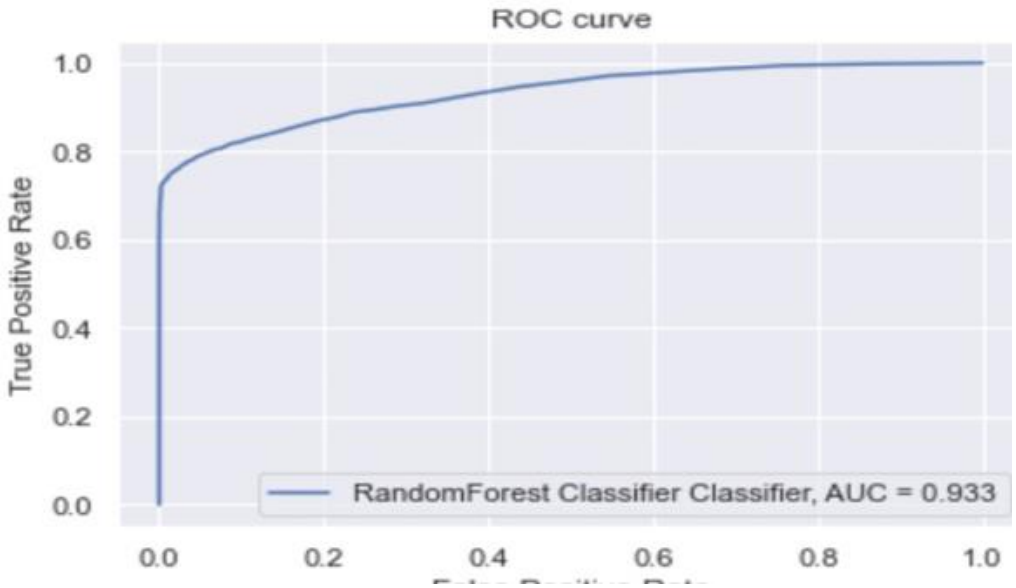


Рисунок 3.9 - Результат виконання алгоритму випадковий ліс

3.5. Побудова моделі випадковий ліс

Побудова моделі випадковий ліс базується на методі дерева рішень, але при цьому алгоритм структурується на основі великої кількості дерева, а не одного(рис.3.10).

RandomForest Classifier

	precision	recall	f1-score	support
0	0.93	0.99	0.96	7631
1	0.96	0.74	0.83	2142
accuracy			0.94	9773
macro avg	0.95	0.86	0.90	9773
weighted avg	0.94	0.94	0.93	9773

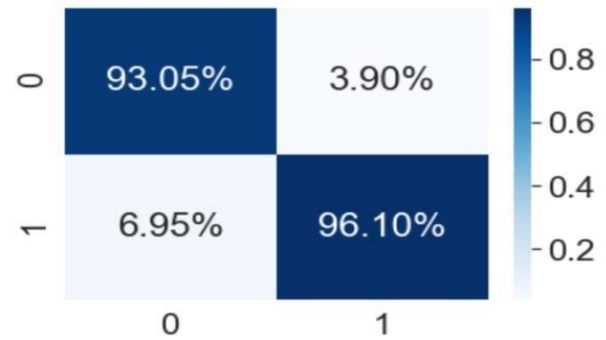


Рисунок 3.10 - Результати роботи моделі випадковий ліс

Якщо проводити порівняння міє методами опираючись на отриманий результат то можно побачити що дерево рішень більш потужно проявив себе на цьому наборі даних.

3.6. Порівняння побудованих моделей та аналіз результатів

Таблиця 3.2. Порівняльна таблиця характеристик якості скорингових моделей

Назва	Точність	AUC	GINI	Час роботи, с
Система показників Логістика Регресія	0.34	0,453	0.78	1.28
Класифікатор KNeighbors	0.5353	0,53	0,8	0.828
Дицизіонні дерева	0.37	0,535	0,7	0.004
Випадковий ліс	0.5515	0,78	0,82	2.85

Опираючись на дані з таблиці, найкраще себе проявив Випадковий ліс, продемонструвавши найбільшу точність обчислення(рис.3.14).

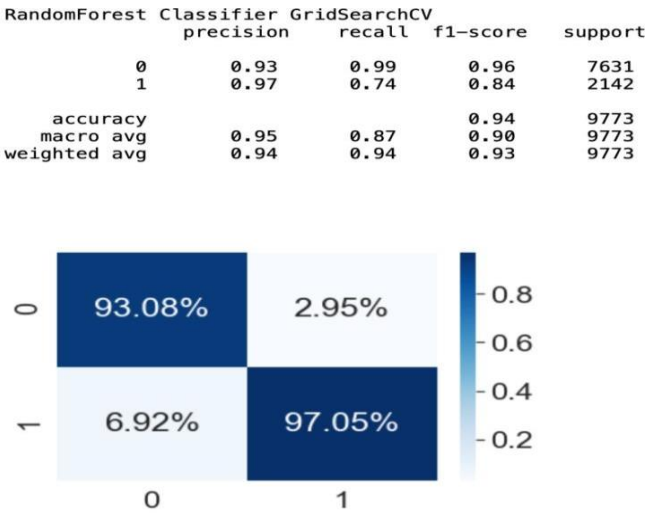


Рисунок 3.14 - Результати роботи моделі Випадковий ліс

3.7. Висновки до розділу 3

В даному розділі було проведено роботу з данними, описано вхідні данні , проведено попередню обробку даних, побудовано логістичну регресію.

Створено модель дерева рішень та випадковий ліс, проведено аналіз та порівняння побудованих моделей.

З наведених моделей найбільш точно відпрацювала побудова моделі випадковий.

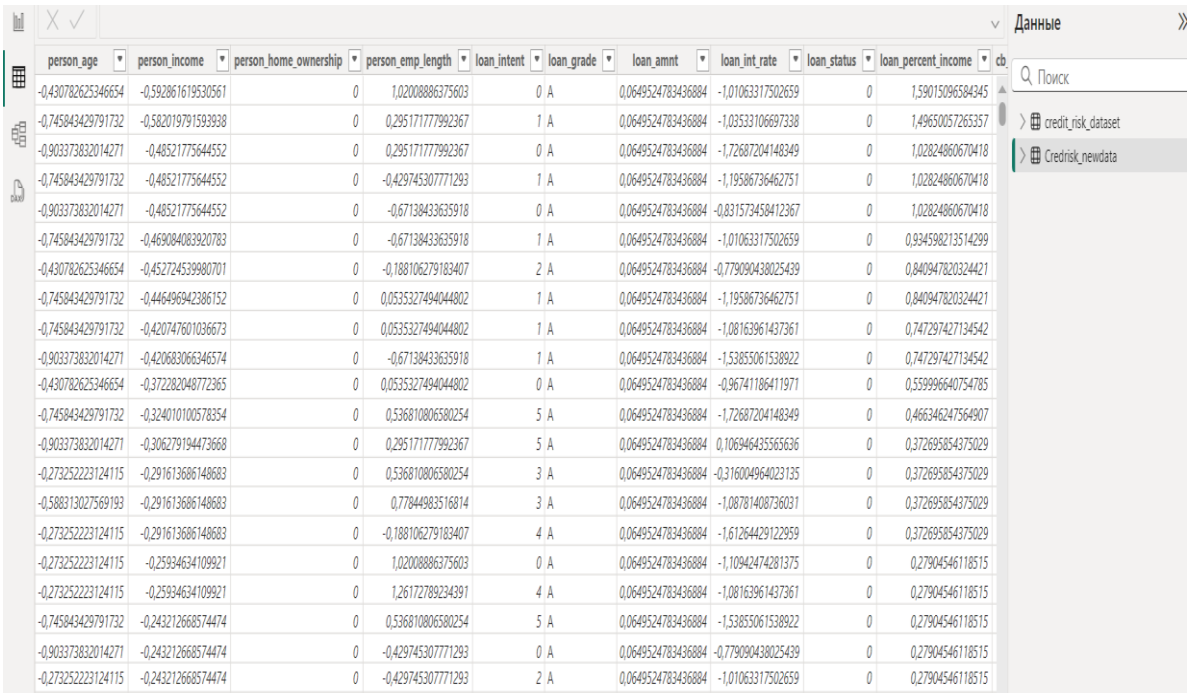
Також в даному розділі детально описаний процес очистки даних, первинної обрабки.

РОЗДІЛ 4. СТВОРЕННЯ ДАШБОРДА В POWER BI

4.1. Робота в середовищі Power bi

Power BI – це потужний інструмент для збору даних їх аналізу та подальшої візуалізації.

Набір даних був завантажений в Power bi для подальшого створення дашборду (рис.4.1).



person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb
-0,430782625346654	-0,592861619530561	0	1,02008886375603	0	A	0,0649524783436884	-1,01063317502659	0	1,59015096584345	
-0,745843429791732	-0,582019791593938	0	0,295171777992367	1	A	0,0649524783436884	-1,03533106697338	0	1,49650057265357	
-0,903373832014271	-0,48521775644552	0	0,295171777992367	0	A	0,0649524783436884	-1,72687204148349	0	1,02824860670418	
-0,745843429791732	-0,48521775644552	0	-0,429745307771293	1	A	0,0649524783436884	-1,19586736462751	0	1,02824860670418	
-0,903373832014271	-0,48521775644552	0	-0,67138433635918	0	A	0,0649524783436884	-0,831573458412367	0	1,02824860670418	
-0,745843429791732	-0,469084083920783	0	-0,67138433635918	1	A	0,0649524783436884	-1,01063317502659	0	0,934598213514299	
-0,430782625346654	-0,452724539980701	0	-0,188106279183407	2	A	0,0649524783436884	-0,779090438025439	0	0,840947820324421	
-0,745843429791732	-0,446496942386152	0	0,0535327494044802	1	A	0,0649524783436884	-1,19586736462751	0	0,840947820324421	
-0,745843429791732	-0,420747601036673	0	0,0535327494044802	1	A	0,0649524783436884	-1,08163961437361	0	0,747297427134542	
-0,903373832014271	-0,420683066346574	0	-0,67138433635918	1	A	0,0649524783436884	-1,53855061538922	0	0,747297427134542	
-0,430782625346654	-0,372282048772365	0	0,0535327494044802	0	A	0,0649524783436884	-0,96741186411971	0	0,559996640754785	
-0,745843429791732	-0,324010100578354	0	0,536810806580254	5	A	0,0649524783436884	-1,72687204148349	0	0,466346247564907	
-0,903373832014271	-0,306279194473668	0	0,295171777992367	5	A	0,0649524783436884	0,106946435565636	0	0,372695854375029	
-0,273252223124115	-0,291613686148683	0	0,536810806580254	3	A	0,0649524783436884	-0,316004964023135	0	0,372695854375029	
-0,588313027569193	-0,291613686148683	0	0,77844983516814	3	A	0,0649524783436884	-1,08781408736031	0	0,372695854375029	
-0,273252223124115	-0,291613686148683	0	-0,188106279183407	4	A	0,0649524783436884	-1,61264429122959	0	0,372695854375029	
-0,273252223124115	-0,25934634109921	0	1,02008886375603	0	A	0,0649524783436884	-1,10942474281375	0	0,27904546118515	
-0,273252223124115	-0,25934634109921	0	1,26172789234391	4	A	0,0649524783436884	-1,08163961437361	0	0,27904546118515	
-0,745843429791732	-0,243212668574474	0	0,536810806580254	5	A	0,0649524783436884	-1,53855061538922	0	0,27904546118515	
-0,903373832014271	-0,243212668574474	0	-0,429745307771293	0	A	0,0649524783436884	-0,779090438025439	0	0,27904546118515	
-0,273252223124115	-0,243212668574474	0	-0,429745307771293	2	A	0,0649524783436884	-1,01063317502659	0	0,27904546118515	

Рисунок 4.1 – Представлення таблиці

Дашборд виконує аналіз кредитного ризику та містить кілька візуалізацій ключових показників (рис.4.2).

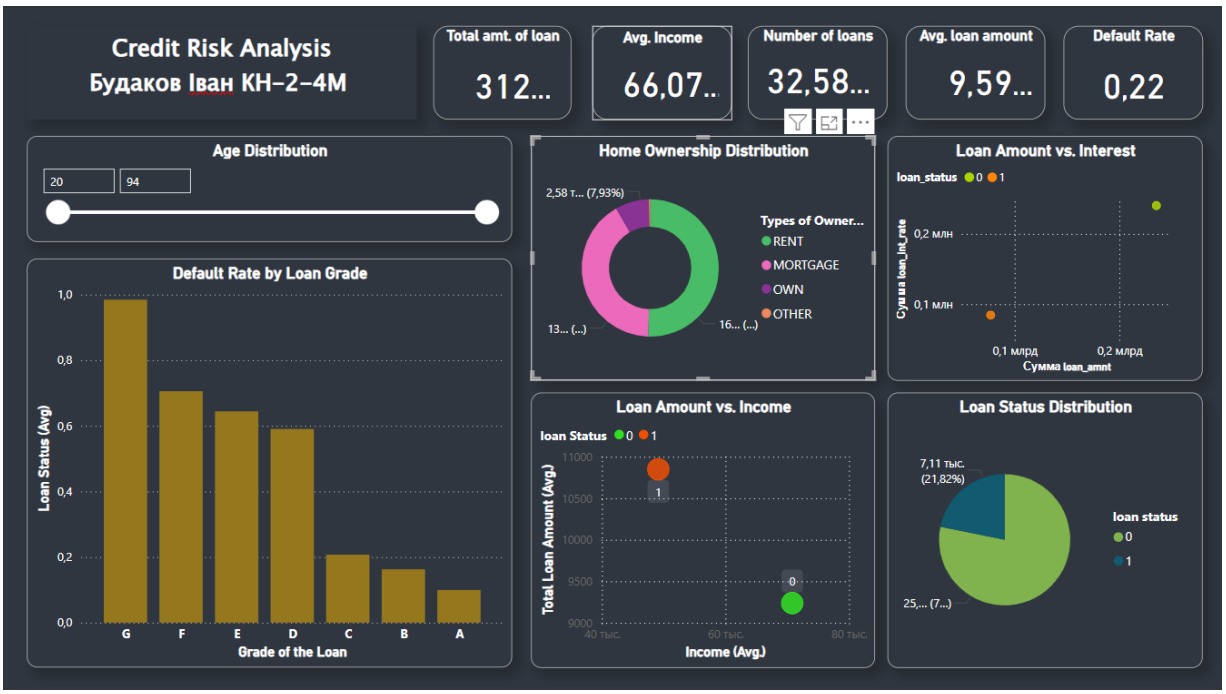


Рисунок 4.2 – Готовий даштборд

На забражені представлені ключові фінансові показники (рис.4.3).

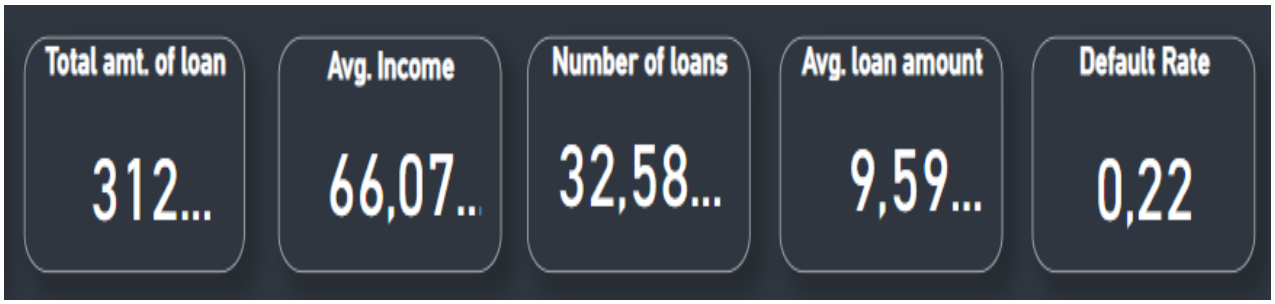


Рисунок 4.3 – Ключові показники KPI

Дає швидке уявлення про обсяг фінансування (рис.4.4).

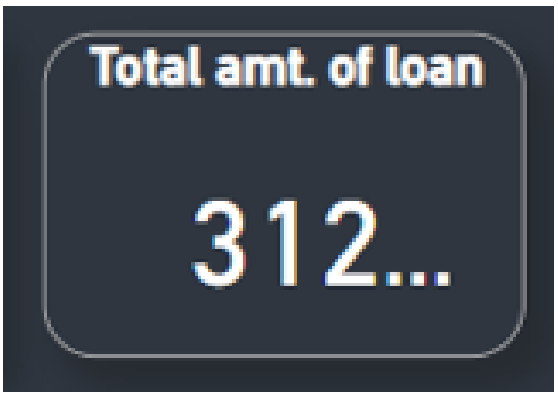


Рисунок 4.4 – Загальна сума виданих кредитів.

Середній дохід позичальників свідчить про те, що клієнти мають відносно стабільні доходи (рис.4.5).

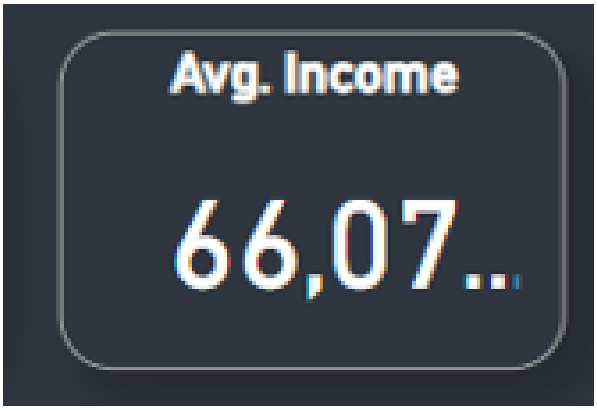


Рисунок 4.5 – Середній дохід позичальників.

Показує середню купівельну спроможність клієнтів (рис.4.6).

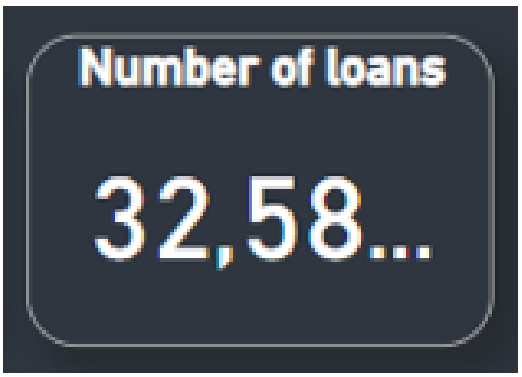


Рисунок 4.6 – Загальна кількість кредитів.

Відображає обсяг роботи кредитного портфеля (рис.4.7).

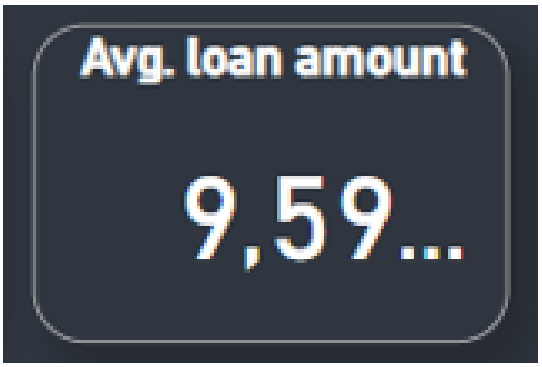


Рисунок 4.7 – Середня сума одного кредиту.

Загальна сума кредитів (412 млн) вказує на значний обсяг роботи кредитного портфеля (рис.4.8).

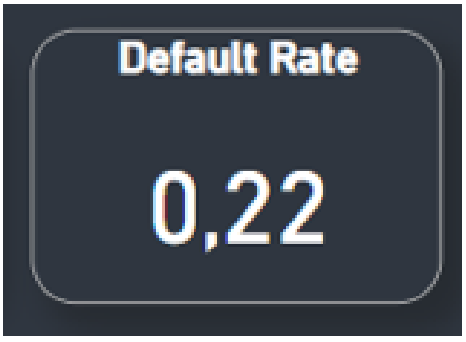


Рисунок 4.8 – Відсоток позичальників, які не змогли повернути кредити.

Ключовий показник для оцінки кредитного ризику.

Середня сума кредиту (9,59 тисяч) є відносно невеликою, що може свідчити про низькоризикову політику видачі кредитів.

Рівень дефолту (22%) є досить високим, що сигналізує про потенційні ризики для фінансової установи (рис.4.9).

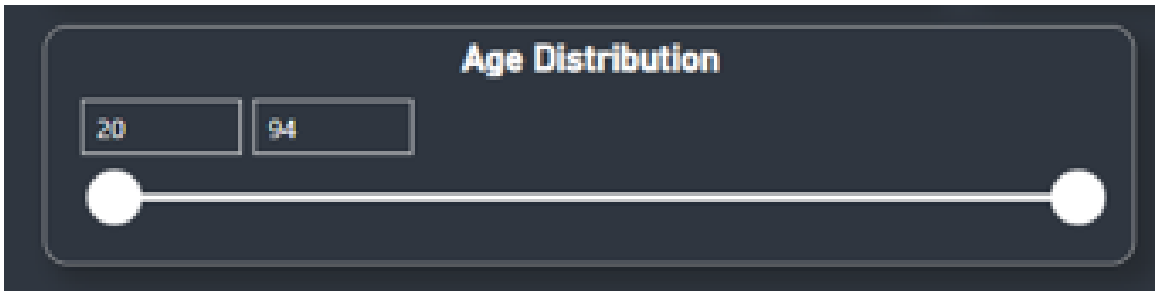


Рисунок 4.9 – Розподіл за віком

Гістограма зі слайдером для вибору вікових діапазонів. Відображає, як розподіляються позичальники за віком.

Позичальники охоплюють широкий віковий діапазон (20–94 роки). Це говорить про різноманітність клієнтської бази, але для більш глибокого аналізу варто перевірити, чи є вікові групи з підвищеним ризиком дефолту (рис.4.10).

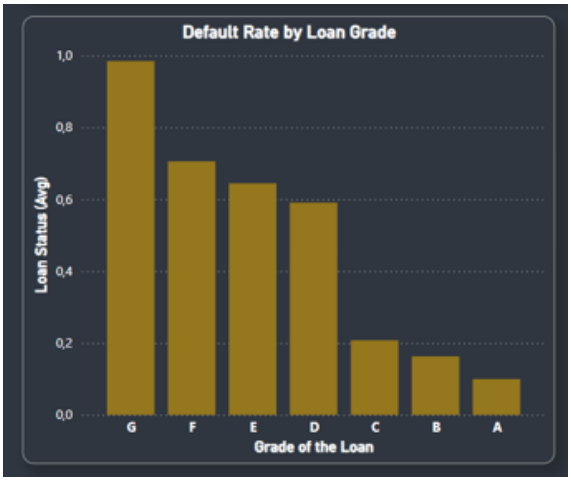


Рисунок 4.10 – Рівень дефолту за рейтингом кредиту

Стовпчаста діаграма, яка показує середній рівень дефолтів для кожного кредитного рейтингу (A, B, C тощо).

Візуалізація допомагає зрозуміти, які рейтинги пов'язані з найвищими ризиками.

Рейтинги кредиту значно впливають на ризик:

Кредити з рейтингом G мають найвищий рівень дефолтів.

Вищі рейтинги (A, B, C) пов’язані з нижчим ризиком.

Це підтверджує, що рейтинг є ключовим фактором у прогнозуванні дефолту (рис.4.11).

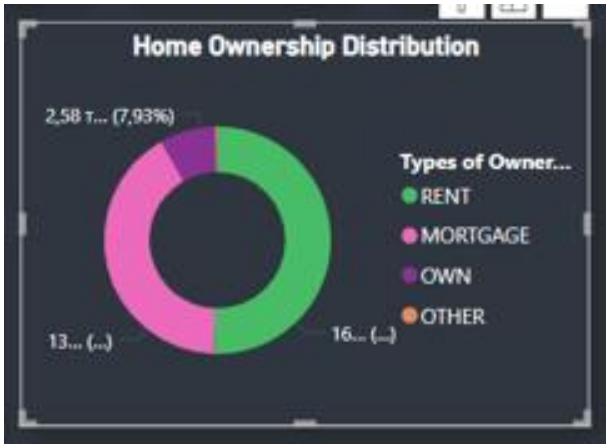


Рисунок 4.11 – Розподіл володіння житлом

Кругова діаграма, яка показує частки позичальників залежно від типу володіння житлом (оренда, іпотека, власність тощо).

Це може вказувати на рівень фінансової стабільності позичальників. (рис.4.12).

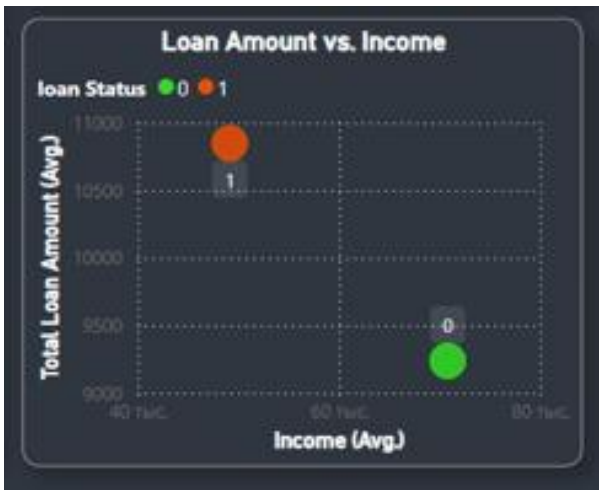


Рисунок 4.12 – Сума кредиту проти процентної ставки

Точкова діаграма, яка порівнює суми кредитів із їхніми процентними ставками. Відображає зв'язок між розміром кредиту та вартістю позики.

Малі кредити мають вищі ставки, що може компенсувати ризики (рис.4.13).

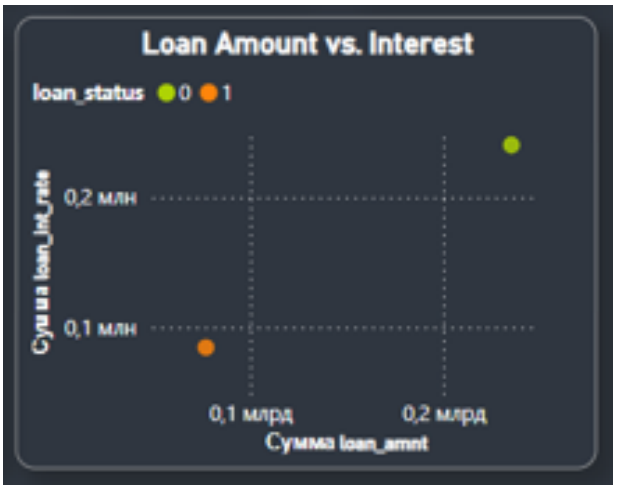


Рисунок 4.13 – Сума кредиту проти доходу

Точкова діаграма, яка відображає середній дохід позичальників і відповідні суми кредитів, розділені за статусом кредиту (0 – не дефолт, 1 – дефолт).

Дозволяє побачити, чи більші кредити видавалися клієнтам із вищими доходами.

Позичальники з більшими доходами частіше отримують вищі суми кредитів.

Дефолти (позначені статусом "1") частіше трапляються серед клієнтів із нижчим доходом і середніми сумами кредитів (рис.4.14).

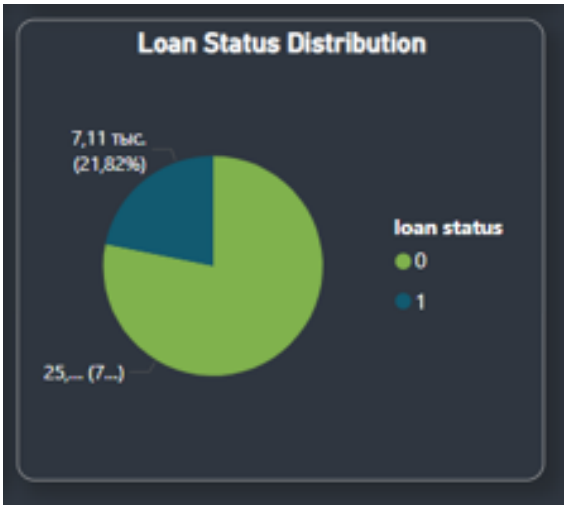


Рисунок 4.14 – Розподіл статусу кредитів

Кругова діаграма, що показує співвідношення кредитів із різними статусами (дефолт/не дефолт). Це дає загальну картину якості кредитного портфеля.

4.2. Висновки до розділу 4

В даному розділі детально описаний дашборд створений в Power bi, детально розписані всі його показники.

Найбільша частка клієнтів перебуває в категоріях оренда та іпотека, що може свідчити про більшу залежність цих груп від фінансової стабільності.

Частка дефолтів становить 21,82%, що є суттєвою загрозою для фінансової стабільності.

Великі кредити (понад 200 тисяч) часто супроводжуються нижчими процентними ставками, що свідчить про довіру до більш платоспроможних позичальників.

Позичальники з власним житлом, ймовірно, є менш ризиковими.

ВИСНОВКИ

В процесі виконання кваліфікаційною роботи було досліджено методи машиного навчання які можуть полегшити та пришвидшити процес надання кредиту клієнтам банку опираючись на ключові показники.

Для реалізації цього завдання було обрано мову програмування Python через його велику кількість бібліотек спрямованих на те щоб полегшити роботу з аналізом даних та візуалізацію всих процесів та побудови відповідних діаграм та графіків.

На основі отриманих результатів можно зробити висновок що за допомогою методів машинного навчання цілком реально визначати та класифікувати клієнтів у відповідності до критерії які банк буде поставляти базуючись на інформації опрацьовану за допомогою машинного навчання.

Було детально розглянуто такі моделі як логістична та лінійна регресія, дерева рішень, випадковий ліс, К-найближчих сусідів.

Також за допомогою Power BI було розроблено дашборд з детальним описом ключових показників.

В майбутньому дослідження та вдосконалення такої системи може стати ще більш точним та призвести до мінімалізації ризиків, використовуючи нові модулі котрі допоможуть покращити роботи алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. НБУ Огляд банківського сектору серпень 2024. [Електронний ресурс]: https://bank.gov.ua/ua/news/all/oglyad-bankivskogo-sektoru-serpen-2024-roku.
2. Бідюк П.І. Система підтримки прийняття рішень для аналізу фінансових даних / П.І. Бідюк, Н.В. Кузнєцова, О.М. Терентьєв // Наук. вісті НТУУ «КПІ». — 2011. — No 1. — с. 48–61.
3. Бідюк П.І. Оцінювання кредитних ризиків методами інтелектуального аналізу даних / В.Я. Данилов, О.Л. Жиров, П.І. Бідюк, 2017 - 46с.
4. Кузнєцова Н.В. Інтегрований підхід до оцінювання кредитних ризиків. 2010. №1-2. с. 187–192.
5. Гаврилюк Г. В. Аналіз вагомості критерів в оцінюванні кредитоспроможності фізичних осіб. Нейро-нечіткі технології моделювання в економіці. 2017. №6. с. 3–23.
6. Камінський А. Б., Сікач В. О. Нейромережеві технології в управлінні портфелем простроченої заборгованості. Моделювання та інформаційні системи в економіці. 2011. №84. с. 5–19.
7. Сушко В.І., Павлюк Т.С. Класифікація моделей оцінки ймовірності банкрутства підприємств. Економіка: теорія та практика. 2014. №1. с.72 - 83.
8. space.py – Офіційний репозиторій OpenAI Gym [Електронний ресурс] – Режим доступу: https://github.com/openai/gym/blob/master/gym/spaces/space.py.
9. core.py – Офіційний репозиторій OpenAI Gym [Електронний ресурс] – Режим доступу: https://github.com/openai/gym/blob/master/gym/core.py .
10. PyTorch [Електронний ресурс] – Режим доступу: https://pytorch.org/.
11. Van den Broeck J., Herbst A.J., Solveig A.C. Data Cleaning: Detecting, Diagnosing,and Editing Data Abnormalities. PLoS Medicine. 2005. №2(10). pp. 966–970.

12. COPOD: Copula-Based Outlier Detection / [Z. Li, Y. Zhao, N. Botta та ін.]. // International Conference on Data Mining. – 2020. - doi.org/10.48550/arXiv.2009.09463.
13. Bao W. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment / W. Bao, N. Lianju, K. Kong Yue. // Expert Systems with Applications. – 2019. – №128. – С. 301–315. - doi.org/10.1016/j.eswa.2019.02.033.
14. Belhaouari S. Unsupervised outlier detection in multidimensional data / S. Belhaouari, A. Rehman // Journal of Big Data / S. Belhaouari, A. Rehman., 2021. - doi.org/10.1186/s40537-021-00469-z.
15. Delany S. k-Nearest Neighbour Classifiers / S. Delany, P. Cunningham // ACM Computing Surveys / S. Delany, P. Cunningham., 2021. – С. 1–25. - doi.org/10.1145/3459665.
16. Gavrylenko S.Y., Chelak V.V., Hornostal O.A., Zozulia V.D. Machine Learning. Laboratory Workshop / S.Y. Gavrylenko, V.V. Chelak, O.A. Hornostal, V.D. Zozulia. - Kharkiv: NTU "KhPI", 2022. - 86 p. https://repository.kpi.kharkov.ua/items/c5ca20d3-d3e5-49e0-99b6-6f8a02b68b57.
17. Halterman R.L. Fundamentals of Python Programming. – Southern Adventist University. 2019.– 658 p. https://folk.ntnu.no/sverrsti/INGG1001-H2019/pythonbook_20191015.pdf.
18. Бобиль В.В. Фінансові ризики банків: теорія та практика управління в умовах кризи: монографія / В.В. Бобиль. Дніпропетровськ, 2016. 298 с.
19. Gavrylenko S., Hornostal O. Application of heterogeneous ensembles in problems of computer system state identification.
20. Харченко В. О. Основи машинного навчання: навч. посіб. /В. О. Харченко. – Суми: Сумський державний університет, 2023. – 264 с.
21. Месюра В. І., Іванчук Я. В., Колесницький О. К. Методичні вказівки до

виконання контрольних робіт з дисципліни «Інтелектуальний аналіз даних» для студентів заочної форми навчання спеціальності 122 – «Комп'ютерні науки» / Уклад. В. І. Месюра, Я. В. Іванчук, О. К. Колесницький. – Вінниця: ВНТУ, 2021. – 42 с.

22. Т.М. Басюк, В.В. .Литвин, Л.М. Захарія, Н.Е. Кунанець. Машинне навчання: Навчальний посібник призначений для студентів, що навчаються за першим (бакалаврським) рівнем вищої освіти за спеціальностями галузі знань 12 „Інформаційні технології". – Львів: Видавництво «Новий Світ - 2000», 2019. - 335 с.
23. Олещенко, Л. М. Машинне навчання. Комп'ютерний практикум [Електронний ресурс]: навчальний посібник для студентів, які навчаються за спеціальністю 121 «Інженерія програмного забезпечення», освітня програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем» / Л. М. Олещенко; КПІ ім. Ігоря Сікорського. – Електронні текстові данні (1 файл: 2,54 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2022. – 92 с.
24. Методи машинного навчання при проєктуванні автоматизованих систем керування [Електронний ресурс] : навч. посіб. для аспірантів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укладач: Т. Г. Баган; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 313 кБайт). Київ: КПІ ім. Ігоря Сікорського, 2021. 28 с.
25. Коваленко В.В. Система ризик-менеджменту в банках: теоретичні та методологічні аспекти: монографія / за ред. В.В. Коваленко. Одеса: ОНЕУ, 2017. 304 с.
26. Гавриленко С.Ю., Челак В.В., Горносталь О.А., Зозуля В.Д. Машинне навчання. Лабораторний практикум, Х.: НТУ «ХПІ», 2022, 86 с.
27. Гавриленко С.Ю. Методичні вказівки до виконання розрахункового завдання з дисципліни «Машинне навчання», Х.: НТУ «ХПІ», 2024, 28 с.

28. Штовба С.Д., Козачко О.М. Machine learning: навч.посіб, Вінниця : ВНТУ, 2020, 81 с.
29. Лубко Д.В., Шаров С.В. Методи та системи штучного інтелекту: навч. посіб, Мелітополь: ФОП Однорог Т.В., 2019, 264 с.
30. Кононова К. Ю. Машинне навчання: методи та моделі: підр., Харків: ХНУ імені В. Н. Каразіна, 2020, 301 с.
31. Eckroth J. Python Artificial Intelligence Projects for Beginners. Birmingham: Packt Publishing, 2018. Dedov, Florian. The Python Bible Volume 6: Neural Networks (Tensorflow, Deep Learning, Keras). N.p., Amazon Digital Services LLC - Kdp, 2020.
32. Artasanchez A., Joshi P. Artificial Intelligence with Python. Second Edition. Birmingham: Packt Publishing, 2020.
33. Da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B., dos Reis Alves, S.F. Artificial Neural Networks A Practical Course: - Springer, 2017.-277.
34. Convolutional Neural Networks In Python: Beginner's Guide To Convolutional Neural Networks In Python. N.p., Frank Millstein, 2020.
35. Graph, Mark. Deep Learning with Python: The Ultimate Guide to Understand Deep Neural Networks with Python Through PyTorch, TensorFlow and Keras. Discover the Ethical Implications of Deep Learning in the New World. USA, Independently Published, 2019. - 235.

ДОДАТКИ

Додаток А. Навчання моделей

```
# Імпортуємо необхідні бібліотеки
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay, roc_curve, roc_auc_score

# Завантаження датасету
from google.colab import files
uploaded = files.upload()

# Читання CSV-файлу
file_name = list(uploaded.keys())[0]
data = pd.read_csv(file_name)

# Закодовуємо категорійні змінні
categorical_columns = ['loan_grade', 'loan_intent']
encoder = LabelEncoder()
```

```
for col in categorical_columns:
    data[col] = encoder.fit_transform(data[col].astype(str))

# Заповнюємо пропущені значення (якщо є)
data = data.fillna(0)

# Визначаємо цільову змінну та ознаки
X = data.drop(['loan_status', 'age_group'], axis=1, errors='ignore')
y = data['loan_status']

# Розділяємо дані на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Навчаємо модель "випадковий ліс"
rf_model = RandomForestClassifier(random_state=42, n_estimators=100)
rf_model.fit(X_train, y_train)

# Оцінюємо модель "випадковий ліс"
y_pred_rf = rf_model.predict(X_test)
y_pred_prob_rf = rf_model.predict_proba(X_test)[:, 1]
roc_auc_rf = roc_auc_score(y_test, y_pred_prob_rf)

print("Звіт класифікації: Випадковий ліс")
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf),
display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
```

```
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()

# Візуалізуємо важливість ознак
feature_importances = rf_model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({
    'Ознака': features,
    'Важливість': feature_importances
}).sort_values(by='Важливість', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Важливість', y='Ознака', data=importance_df, palette='viridis')
plt.title('Важливість ознак: Випадковий ліс', fontsize=16)
plt.xlabel('Важливість', fontsize=12)
plt.ylabel('Ознака', fontsize=12)
plt.show()

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break
print("Звіт класифікації: Випадковий ліс")
```

```
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf), display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()

# Візуалізуємо важливість ознак
feature_importances = rf_model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({
    'Ознака': features,
    'Важливість': feature_importances
}).sort_values(by='Важливість', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Важливість', y='Ознака', data=importance_df, palette='viridis')
plt.title('Важливість ознак: Випадковий ліс', fontsize=16)
plt.xlabel('Важливість', fontsize=12)
plt.ylabel('Ознака', fontsize=12)
plt.show()

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
```

```
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break
print("Звіт класифікації: Випадковий ліс")
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf), display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()

# Візуалізуємо важливість ознак
feature_importances = rf_model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({
    'Ознака': features,
    'Важливість': feature_importances
}).sort_values(by='Важливість', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Важливість', y='Ознака', data=importance_df, palette='viridis')
plt.title('Важливість ознак: Випадковий ліс', fontsize=16)
plt.xlabel('Важливість', fontsize=12)
plt.ylabel('Ознака', fontsize=12)
plt.show()
plt.figure(figsize=(10, 6))
sns.barplot(x='Важливість', y='Ознака', data=importance_df, palette='viridis')
```

```
plt.title('Важливість ознак: Випадковий ліс', fontsize=16)
plt.xlabel('Важливість', fontsize=12)
plt.ylabel('Ознака', fontsize=12)
plt.show()

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break
print("Звіт класифікації: Випадковий ліс")
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf),
display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()
# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
```

```
        break
print("Звіт класифікації: Випадковий ліс")
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf), display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break
print("Звіт класифікації: Випадковий ліс")
print(classification_report(y_test, y_pred_rf))
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, y_pred_rf), display_labels=rf_model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title("Матриця плутанини: Випадковий ліс")
plt.show()


# Візуалізуємо важливість ознак
```

```
feature_importances = rf_model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({
    'Ознака': features,
    'Важливість': feature_importances
}).sort_values(by='Важливість', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Важливість', y='Ознака', data=importance_df, palette='viridis')
plt.title('Важливість ознак: Випадковий ліс', fontsize=16)
plt.xlabel('Важливість', fontsize=12)
plt.ylabel('Ознака', fontsize=12)
plt.show()

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break

# Виведення інформації про погодженого клієнта
if approved_client:
    index, client_data = approved_client
    print(f"Клієнт з індексом {index}, якому погоджено кредит:")
```

```
    print(client_data)

    # Передбачення ймовірності
    client_probabilities_rf = rf_model.predict_proba(client_data.values.reshape(1, -1))
    print(f"Ймовірність одобрення: {client_probabilities_rf[0][1]:.2f}")
# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break

# Виведення інформації про погодженого клієнта
if approved_client:
    index, client_data = approved_client
    print(f"Клієнт з індексом {index}, якому погоджено кредит:")
    print(client_data)

    # Передбачення ймовірності
    client_probabilities_rf = rf_model.predict_proba(client_data.values.reshape(1, -1))
    print(f"Ймовірність одобрення: {client_probabilities_rf[0][1]:.2f}")

# Цикл для пошуку клієнта, якому погоджено кредит
approved_client = None
for index, client_data in X_test.iterrows():
```

```
    client_data_reshaped = pd.DataFrame([client_data], columns=X.columns)
    client_prediction_rf = rf_model.predict(client_data_reshaped)
    if client_prediction_rf[0] == 1:  # Якщо кредит погоджено
        approved_client = (index, client_data)
        break

# Виведення інформації про погодженого клієнта
if approved_client:
    index, client_data = approved_client
    print(f"Клієнт з індексом {index}, якому погоджено кредит:")
    print(client_data)

    # Передбачення ймовірності
    client_probabilities_rf = rf_model.predict_proba(client_data.values.reshape(1, -1))
    print(f"Ймовірність одобрення: {client_probabilities_rf[0][1]:.2f}")



    # Виведення ключових ознак
    print("\nКлючові ознаки, які вплинули на рішення:")
    for feature in importance_df.head(5).itertuples():
        feature_name = feature.Ознака
        feature_importance = feature.Важливість
        client_value = client_data[feature_name]
        print(f"Ознака: {feature_name}, Важливість: {feature_importance:.2f}, Значення клієнта: {client_value}")
else:
    print("Жоден клієнт у тестовій вибірці не отримав схвалення кредиту.")
```

Реалізація ETL процесу за допомогою Python

```
import pandas as pd
from sqlalchemy import create_engine

# Етап Extract
def extract(csv_file):
    """
    Завантажує дані з CSV файлу.
    """
    data = pd.read_csv(csv_file)
    return data

# Етап Transform
def transform(data):
    """
    Виконує обробку даних:
    - Видаляє дублікатні записи.
    - Заповнює пропущені значення.
    - Заповнює пропущені значення в колонці 'age_group' як 'Unknown'.
    """
    # Видалення дублікатів
    data = data.drop_duplicates()

    # Заповнення пропусків
    data = data.fillna('')

    # Заповнення пропусків у колонці 'age_group'
    if 'age_group' in data.columns:
        data['age_group'] = data['age_group'].replace('', 'Unknown')

    return data

# Етап Load
def load(data, db_name, table_name):
    """
    Завантажує дані в базу даних SQLite.
```

```
    """
    # Створення з'єднання з SQLite
    engine = create_engine(f'sqlite:///{db_name}')

    # Завантаження даних у таблицю
    data.to_sql(table_name, engine, if_exists='replace', index=False)
    print(f"Дані завантажені у базу даних: {db_name}, таблиця: {table_name}")

# Основний ETL процес
def etl_process(csv_file, db_name, table_name):
    """
    Запускає ETL процес: Extract -> Transform -> Load.
    """
    print("Запуск ETL процесу...")

    # Етап Extract
    print("Етап Extract: Завантаження даних з CSV...")
    data = extract(csv_file)

    # Етап Transform
    print("Етап Transform: Обробка даних...")
    data = transform(data)

    # Етап Load
    print("Етап Load: Завантаження даних у SQLite...")
    load(data, db_name, table_name)

    print("ETL процес завершено успішно.")

# Основний запуск
if __name__ == "__main__":
    # Шляхи до файлу CSV та бази даних
    csv_file = '/mnt/data/Credrisk_newdata.csv'  # Ваш файл CSV
    db_name = '/mnt/data/Credrisk_data.db'       # База даних SQLite
    table_name = 'credit_risk_data'              # Назва таблиці

    # Запуск ETL процесу
```

```
    etl_process(csv_file, db_name, table_name)
# Основний ETL процес
def etl_process(csv_file, db_name, table_name):
    """
    Запускає ETL процес: Extract -> Transform -> Load.
    """
    print("Запуск ETL процесу...")

    # Етап Extract
    print("Етап Extract: Завантаження даних з CSV...")
    data = extract(csv_file)

    # Етап Transform
    print("Етап Transform: Обробка даних...")
    data = transform(data)

    # Етап Load
    print("Етап Load: Завантаження даних у SQLite...")
    load(data, db_name, table_name)

    print("ETL процес завершено успішно.")

# Основний запуск
if __name__ == "__main__":
    # Шляхи до файлу CSV та бази даних
    csv_file = '/mnt/data/Credrisk_newdata.csv'  # Ваш файл CSV
    db_name = '/mnt/data/Credrisk_data.db'       # База даних SQLite
    table_name = 'credit_risk_data'              # Назва таблиці

    # Запуск ETL процесу
    etl_process(csv_file, db_name, table_name)
```