Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №8
з навчальної дисципліни «Data Science Technology»

Тема:

МАКЕТ CRM СИСТЕМИ SCORING – АНАЛІЗУ (міні проекти в банківській сфері аналізу даних)

Виконав:
Студент 3 курсу кафедри ОТ ФІОТ,
Навчальної групи ІМ-13
Тавлуй Д. О.

Перевірив:
Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2023

I. Мета:
Дослідити виявити та узагальнити особливості реалізації проектного практикуму в галузі кредитного scoring-у
II. Завдання:
Відділ мікрокредитування банківської установи замовив розробку Backend компоненту SRM банківської системи.
Компонента Backend повинна забезпечити:
1. Побудову скорінгової оцінку;
2. Кластерізацію заявників за бінарної оцінкою.
Вихідні дані для опрацювання задачі представлені у формі 2-х файлів:
1. sample_data.xlsx – файл реальних даних про позичальників та проміжних параметрів банківських індикаторів;
2. data_description.xlsx – файл пояснень структури скорингової таблиці та тлумачення індикаторів,
Розробити програмний скрипт, що реалізує функціонал за обраним рівням складності:
II рівень складності 9 балів

8	Розробити програмний скрипт, що реалізує: 1. Скоринговий аналіз позичальників за даними Data_description.xlsx, Sample_data.xlsx відповідно до самостійно обраної моделі. 2. Передбачити чи буде кредит повернено у форматі бінарної оцінки (0 або 1); 3. Виявлення шахрайства та фальсифікації даних.

III. Результати виконання лабораторної роботи.

Результати архітектурного проектування та їх опис

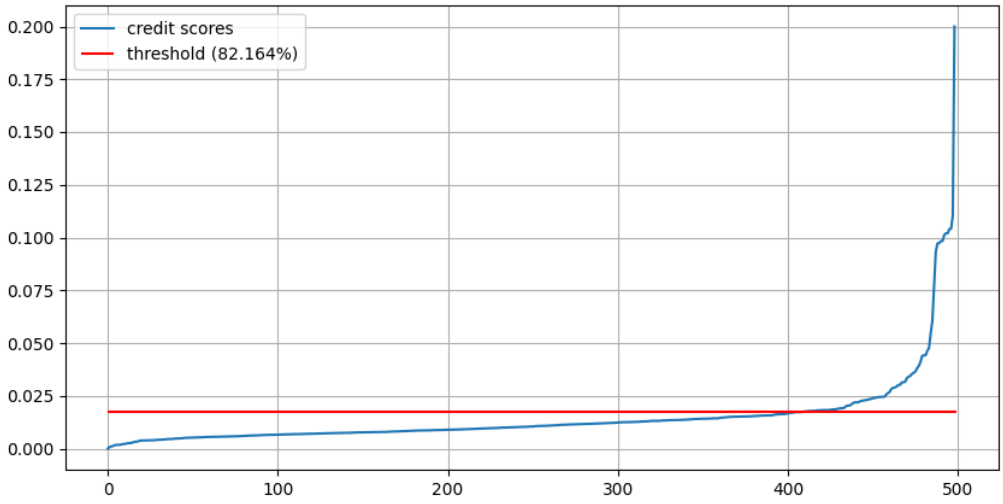
Я обрав монолітну архітектуру проектування. У файлі main.py відбувається оголошення та виклик усіх функцій.

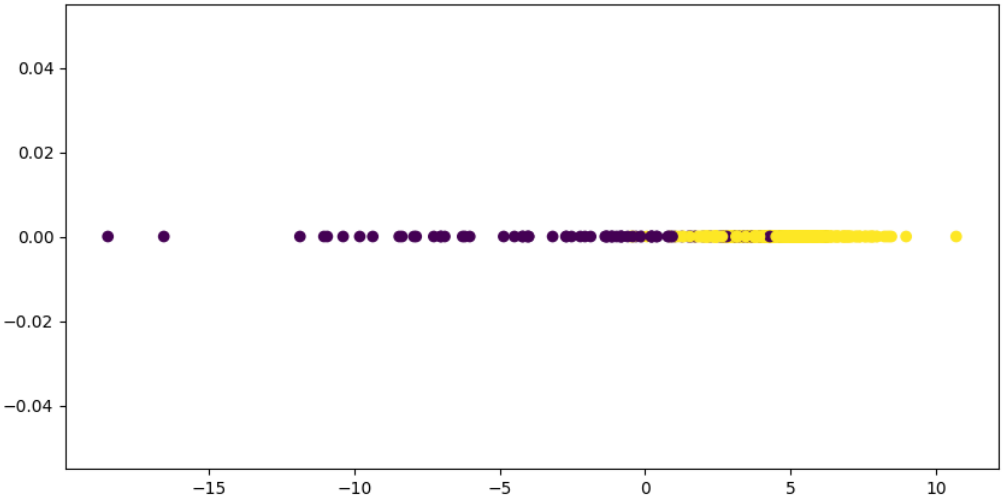
Опис структури проекту програми

Вся структура проекту описана у одному файлі main.py.

Результати роботи програми відповідно до завдання

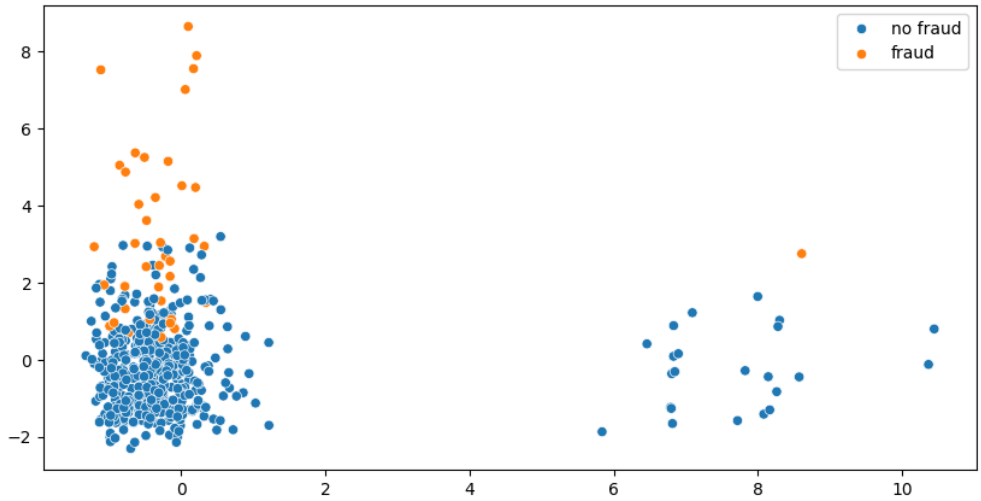
За даними з таблиць Data_description.xlsx та Sample_data.xlsx було проведено скоринговий аналіз. Так як у моєму варіанті можна було обрати будь-яку модель, я обрав модель дискримінантного аналізу, а саме лінійний дискримінантний аналіз. Також я обрав порогове значення скору 0.175.


credit scores
threshold (82.164%)
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
100
200
300
400
500

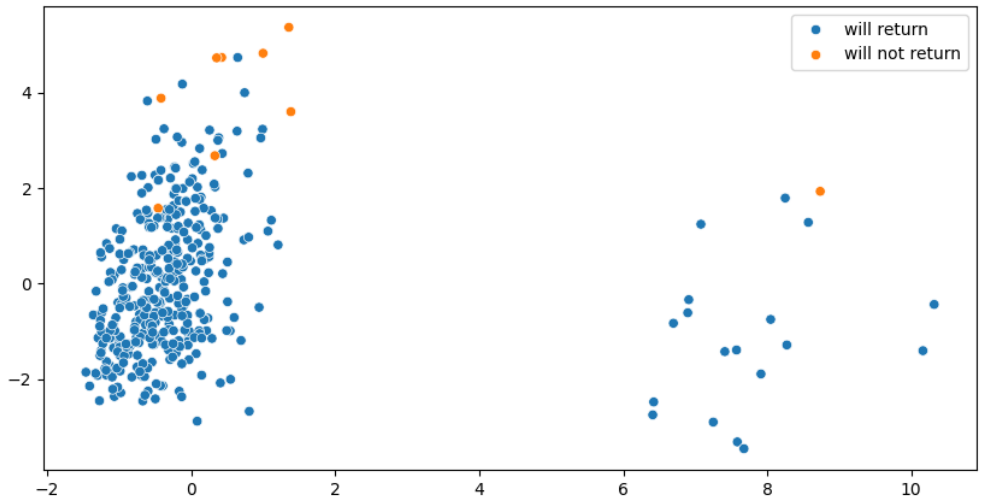

0.04
0.02
0.00
−0.02
−0.04
−15
−10
−5
0
5
10

```
scor_d_line=  [ True  True  True  True  True  True  True  True  True  True False  True
  True  True  True  True  True  True  True  True  True  True  True
  True  True  True False  True  True  True  True  True  True  True False
  True  True False  True  True False  True  True  True  True  True  True
  True False False  True  True  True  True  True  True  True  True  True
  True False  True  True  True False  True  True  True  True False  True
 False  True False  True False False  True  True  True  True  True  True
 False  True False  True  True  True  True  True False  True  True  True
  True  True False  True False  True  True  True False  True  True  True
  True False False  True  True  True  True  True  True  True  True  True
 False  True False  True  True False  True  True  True  True  True False
  True  True  True False  True  True  True  True False  True  True False
  True  True  True  True False  True  True  True  True  True  True False
 False  True False  True  True False  True  True  True False  True  True
  True False  True False  True  True  True  True  True False  True  True
  True  True False  True  True  True  True  True  True  True  True False
  True  True  True  True False  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True False  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True False  True  True  True  True  True  True  True
  True False  True  True  True  True False  True  True  True False  True
  True  True  True False  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True False  True False  True
  True  True False  True  True  True False  True False  True  True  True
  True False  True  True False  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True False  True  True  True False
  True False  True  True  True  True  True  True  True  True  True  True
  True  True  True False  True False False  True  True False  True False
  True  True  True  True False  True False  True  True  True  True  True
  True  True  True  True  True  True  True  True False  True  True  True
 True False False False  True  True  True False  True  True  True  True
 True  True  True  True  True  True False False  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True False
 True  True  True  True  True False  True  True False  True  True  True
 True  True  True  True  True  True  True False  True  True  True  True
 True  True  True False  True  True  True False  True  True False  True
False  True  True  True  True False False  True  True  True False  True
 True  True  True False  True False  True  True  True  True  True  True
 True  True  True  True  True  True  True]
```

На жаль, уся бінарна скорингова таблиця не помістилася на одному скріншоті, але все рівно все видно. True – вище порогового значення, а False – нижче. Також вона знаходиться у папці лабораторної у вигляді текстового файлу, де 1 це True, а 0 False. Вона містить 500 рядків, тому додавати її у звіт у вигляді скріншоту або тексту не є доцільним. Але вона повністю повторює таблицю зі скріншоту вище.



Виявлення шахрайства та фальсифікації даних



Та передбачення чи буде повернуто кредит

Програмний код, що забезпечує отримання результату

main.py

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

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

from pyod.models.knn import KNN
import seaborn as sns


def data_preparation(filename):
```

```
    descriptions = pd.read_excel(filename)
    client_bank_descriptions = descriptions[
        (descriptions.Place_of_definition == 'Вказує позичальник') |
        (descriptions.Place_of_definition == 'параметри, повязані з виданим
продуктом')
        ]
    client_bank_fields = client_bank_descriptions["Field_in_data"]

    data = pd.read_excel("sample_data.xlsx")

    data = data.loc[:,
list(set(client_bank_fields).intersection(data.columns))]
    data = data.dropna(axis=1)
    data.head()
    return data


def clip_data(data):
    return np.where(np.sign(data) >= 0, 1, -1) * np.clip(np.abs(data), 1e-
9, None)


def min_max(data):
    return (data - data.min()) / (data.max() - data.min())


def voronin(data: pd.DataFrame, weights: np.array, direction: np.array,
delta=0.1) -> np.array:
    data = data.copy()
    data.loc[direction] = 1 / clip_data(data[direction].values)
    data = data.values
    criteria_sum = np.sum(data, axis=1, keepdims=True)
    normalized_criteria_values = data / clip_data(criteria_sum)
    integro = np.dot(weights, 1/(1 - normalized_criteria_values))
    return integro


def linear_discriminant_analysis(minimax_data):
    viz_data = minimax_data.T
    model = LinearDiscriminantAnalysis()
    model.fit(viz_data, data["give"])
    predicted = model.decision_function(viz_data)
    plt.figure(figsize=(10, 5))
    plt.scatter(predicted, np.zeros_like(predicted), c=data["give"])
    plt.show()
    return predicted


def fraud_detection(minimax_data):
    outliers_fraction = 0.1
    X = minimax_data.T
    clf = KNN(contamination=outliers_fraction)
    clf.fit(X)
    y_pred = clf.predict(X)
    norm_data = StandardScaler().fit_transform(minimax_data.T)
    compressed = PCA(n_components=2).fit_transform(norm_data)
    plt.figure(figsize=(10, 5))
    sns.scatterplot(x=compressed[:, 0], y=compressed[:, 1],
hue=np.where(y_pred, "fraud", "no fraud"))
    plt.show()
    return
```

```
def fraud_no_fraud(minimax_data):
    outliers_fraction = 0.1
    X = minimax_data.T
    clf = KNN(contamination=outliers_fraction)
    clf.fit(X)
    y_pred = clf.predict(X)
    credit_given = minimax_data.T[data["give"] &
~y_pred].reset_index(drop=True)

    outliers_fraction = 0.026
    X = credit_given
    clf = KNN(contamination=outliers_fraction)
    clf.fit(X)
    y_pred = clf.predict(X)

    norm_data = StandardScaler().fit_transform(credit_given)
    compressed = PCA(n_components=2).fit_transform(norm_data)
    plt.figure(figsize=(10, 5))
    sns.scatterplot(x=compressed[:, 0], y=compressed[:, 1],
hue=np.where(y_pred, "will not return", "will return"))
    plt.show()
    return


if __name__ == '__main__':
    data = data_preparation("data_description.xlsx")
    print(data)
    minimax_info =
pd.read_excel("d_segment_data_description_cleaning_minimax.xlsx")
    minimax_info = minimax_info[["Field_in_data", "Minimax"]]
    minimax_info = minimax_info.dropna()
    minimax_info.head()
    print(minimax_info.head())

    minimax_data = data.T
    col_intersection =
list(set(data.columns).intersection(minimax_info["Field_in_data"]))
    minimax_data = minimax_data.loc[col_intersection, :]
    minimax_data.head()

    criteria_count = len(minimax_data)
    criteria_values = minimax_data.astype(float)
    criteria_values = criteria_values.reset_index(drop=True)
    direction =
(minimax_info.set_index("Field_in_data").loc[minimax_data.index,
:]["Minimax"] == "max").values

    integro = min_max(voronin(criteria_values, np.ones(criteria_count) /
criteria_count, direction))

    plt.figure(figsize=(10, 5))
    plt.plot(np.sort(integro.clip(0, 0.2)), label="credit scores")
    plt.hlines(0.0175, 0, len(integro), color="r",
               label=f"threshold ({(integro <= 0.0175).sum() /
len(integro):.3%})", )
    plt.legend()
    plt.grid()
    plt.show()
```

```
    scor_d_line = data["give"] = integro <= 0.0175
    np.savetxt('Integro_Scor.txt', scor_d_line)
    print('scor_d_line= ', scor_d_line)

    linear_discriminant_analysis(minimax_data)

    fraud_detection(minimax_data)

    fraud_no_fraud(minimax_data)
```

Висновки

Під час виконання лабораторної роботи я скористався пороговим значенням для інтегрованої оцінки та проводити скоринговий аналіз даних за допомогою дискримінантної моделі для того щоб виявляти шахрайство. Також провів аналіз чи поверне людина кредит на основі заданих даних.