FINAL PROJECT PRESENTATION

TAI PHAM
DA17-2023

Credit Card Approval Prediction

DATA : https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

Mô tả dữ liệu :

application_record.csv

Feature name	Explanation	Remarks
ID	Client number	
CODE_GENDER	Gender	
FLAG_OWN_CAR	Is there a car	
FLAG_OWN_REALTY	Is there a property	
CNT_CHILDREN	Number of children	
AMT_INCOME_TOTAL	Annual income	
NAME_INCOME_TYPE	Income category	

credit_record.csv

Feature name	Explanation	Remarks
ID	Client number	
MONTHS_BALANCE	Record month	The month of the extra month, and so on
		0: 1-29 days past due 1

Load data and check data

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

```
[128] application_record =pd.read_csv("/content/drive/MyDrive/archive/application_record.csv")
      credit_record = pd.read_csv("/content/drive/MyDrive/archive/credit_record.csv")
```

```
credit_record.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 3 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   ID              1048575 non-null  int64
 1   MONTHS_BALANCE  1048575 non-null  int64
 2   STATUS          1048575 non-null  object
dtypes: int64(2), object(1)
memory usage: 24.0+ MB
```

```
[190] len(pd.unique(credit_record["ID"]))
```

```
45985
```

```
new["approval"].value_counts()
```

```
0    45318
1      667
Name: approval, dtype: int64
```

```
new
```

ID	approval
5001711	0
5001712	0
5001713	0
5001714	0
5001715	0
...	...
5150482	0
5150483	0
5150484	0
5150485	0
5150487	0

45985 rows × 1 columns

EDA

-Kiểm tra dữ liệu bị thiếu

```
new_data.isna().sum()
```

```
ID                        0
CODE_GENDER               0
FLAG_OWN_CAR              0
FLAG_OWN_REALTY           0
CNT_CHILDREN              0
AMT_INCOME_TOTAL          0
NAME_INCOME_TYPE          0
NAME_EDUCATION_TYPE       0
NAME_FAMILY_STATUS        0
NAME_HOUSING_TYPE         0
DAYS_BIRTH                0
DAYS_EMPLOYED             0
FLAG_MOBIL                0
FLAG_WORK_PHONE           0
FLAG_PHONE                0
FLAG_EMAIL                0
OCCUPATION_TYPE       11323
CNT_FAM_MEMBERS           0
approval                  0
dtype: int64
```

```
[143] new_data=pd.merge(application_record,new,how="inner",on="ID")
```

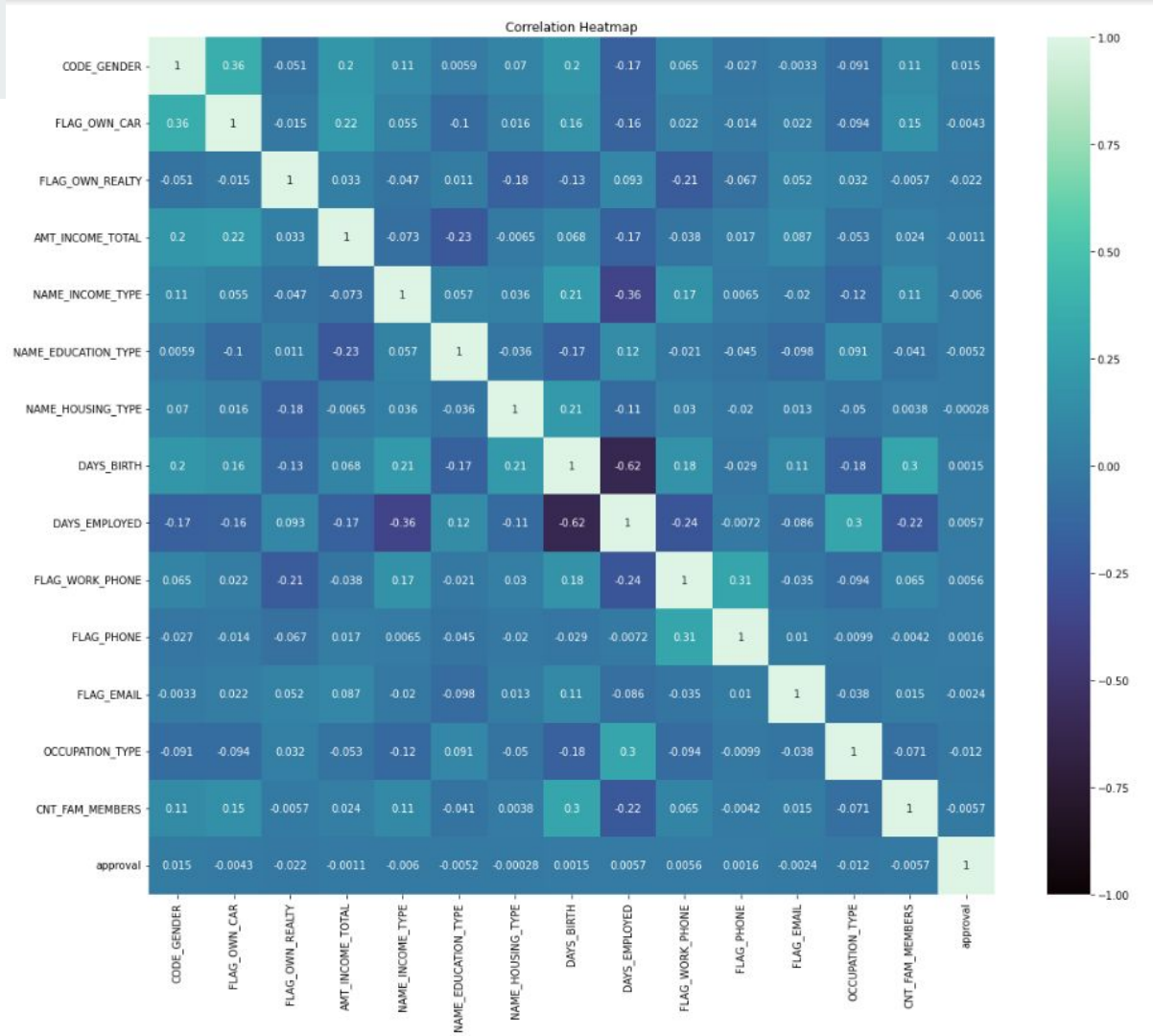
```
new_data
```

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
0	5008804	M	Y	Y	0	427500.0
1	5008805	M	Y	Y	0	427500.0
2	5008806	M	Y	Y	0	112500.0
3	5008808	F	N	Y	0	270000.0
4	5008809	F	N	Y	0	270000.0
...	...	...	...	...	...	...
36452	5149828	M	Y	Y	0	315000.0
36453	5149834	F	N	Y	0	157500.0
36454	5149838	F	N	Y	0	157500.0
36455	5150049	F	N	Y	0	283500.0
36456	5150337	M	N	Y	0	112500.0

36457 rows × 19 columns

Pre Processing Data

```
[26] from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
```


Correlation Heatmap
CODE_GENDER 1 0.36 -0.051 0.2 0.11 0.0059 0.07 0.2 -0.17 0.065 -0.027 -0.0033 -0.091 0.11 0.015
FLAG_OWN_CAR 0.36 1 -0.015 0.22 0.055 -0.1 0.016 0.16 -0.16 0.022 -0.014 0.022 -0.094 0.15 -0.0043
FLAG_OWN_REALTY -0.051 -0.015 1 0.033 -0.047 0.011 -0.18 -0.13 0.093 -0.21 -0.067 0.052 0.032 -0.0057 -0.022
AMT_INCOME_TOTAL 0.2 0.22 0.033 1 -0.073 -0.23 -0.0065 0.068 -0.17 -0.038 0.017 0.087 -0.053 0.024 -0.0011
NAME_INCOME_TYPE 0.11 0.055 -0.047 -0.073 1 0.057 0.036 0.21 -0.36 0.17 0.0065 -0.02 -0.12 0.11 -0.006
NAME_EDUCATION_TYPE 0.0059 -0.1 0.011 -0.23 0.057 1 -0.036 -0.17 0.12 -0.021 -0.045 -0.098 0.091 -0.041 -0.0052
NAME_HOUSING_TYPE 0.07 0.016 -0.18 -0.0065 0.036 -0.036 1 0.21 -0.11 0.03 -0.02 0.013 -0.05 0.0038 -0.00028
DAYS_BIRTH 0.2 0.16 -0.13 0.068 0.21 -0.17 0.21 1 -0.62 0.18 -0.029 0.11 -0.18 0.3 0.0015
DAYS_EMPLOYED -0.17 -0.16 0.093 -0.17 -0.36 0.12 -0.11 -0.62 1 -0.24 -0.0072 -0.086 0.3 -0.22 0.0057
FLAG_WORK_PHONE 0.065 0.022 -0.21 -0.038 0.17 -0.021 0.03 0.18 -0.24 1 0.31 -0.035 -0.094 0.065 0.0056
FLAG_PHONE -0.027 -0.014 -0.067 0.017 0.0065 -0.045 -0.02 -0.029 -0.0072 0.31 1 0.01 -0.0099 -0.0042 0.0016
FLAG_EMAIL -0.0033 0.022 0.052 0.087 -0.02 -0.098 0.013 0.11 -0.086 -0.035 0.01 1 -0.038 0.015 -0.0024
OCCUPATION_TYPE -0.091 -0.094 0.032 -0.053 -0.12 0.091 -0.05 -0.18 0.3 -0.094 -0.0099 -0.038 1 -0.071 -0.012
CNT_FAM_MEMBERS 0.11 0.15 -0.0057 0.024 0.11 -0.041 0.0038 0.3 -0.22 0.065 -0.0042 0.015 -0.071 1 -0.0057
approval 0.015 -0.0043 -0.022 -0.0011 -0.006 -0.0052 -0.00028 0.0015 0.0057 0.0056 0.0016 -0.0024 -0.012 -0.0057 1
CODE_GENDER
FLAG_OWN_CAR
FLAG_OWN_REALTY
AMT_INCOME_TOTAL
NAME_INCOME_TYPE
NAME_EDUCATION_TYPE
NAME_HOUSING_TYPE
DAYS_BIRTH
DAYS_EMPLOYED
FLAG_WORK_PHONE
FLAG_PHONE
FLAG_EMAIL
OCCUPATION_TYPE
CNT_FAM_MEMBERS
approval
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

-Drop columns, Scale dữ liệu

	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_HOUSING_TYPE	DAYS_BIRTH	DAYS_EMPLOYED	FLAG_WORK_PHONE	FLAG_PHONE	
0	1.0	1.0	1.0	0.258721	1.00	0.25	0.8	0.744324	0.029324	1.0	0.0	
1	1.0	1.0	1.0	0.258721	1.00	0.25	0.8	0.744324	0.029324	1.0	0.0	
2	1.0	1.0	1.0	0.055233	1.00	1.00	0.2	0.208232	0.038270	0.0	0.0	
3	0.0	0.0	1.0	0.156977	0.00	1.00	0.2	0.342071	0.033237	0.0	1.0	
4	0.0	0.0	1.0	0.156977	0.00	1.00	0.2	0.342071	0.033237	0.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	
36452	1.0	1.0	1.0	0.186047	1.00	1.00	0.2	0.441828	0.034894	0.0	0.0	
36453	0.0	0.0	1.0	0.084302	0.00	0.25	0.2	0.722697	0.037768	0.0	1.0	
36454	0.0	0.0	1.0	0.084302	0.25	0.25	0.2	0.722697	0.037768	0.0	1.0	
36455	0.0	0.0	1.0	0.165698	1.00	1.00	0.2	0.407292	0.039527	0.0	0.0	
36456	1.0	0.0	1.0	0.055233	1.00	1.00	0.8	0.903810	0.038115	0.0	0.0	

36457 rows × 14 columns

Building Model

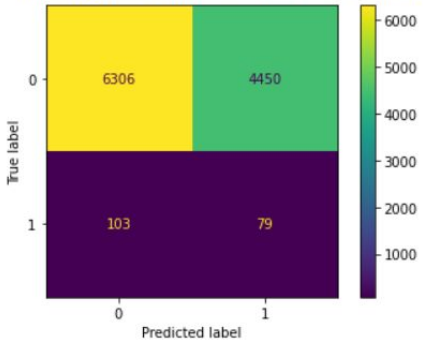
- Logistic Regression
- SVM
- Decision Tree
- RandomForest

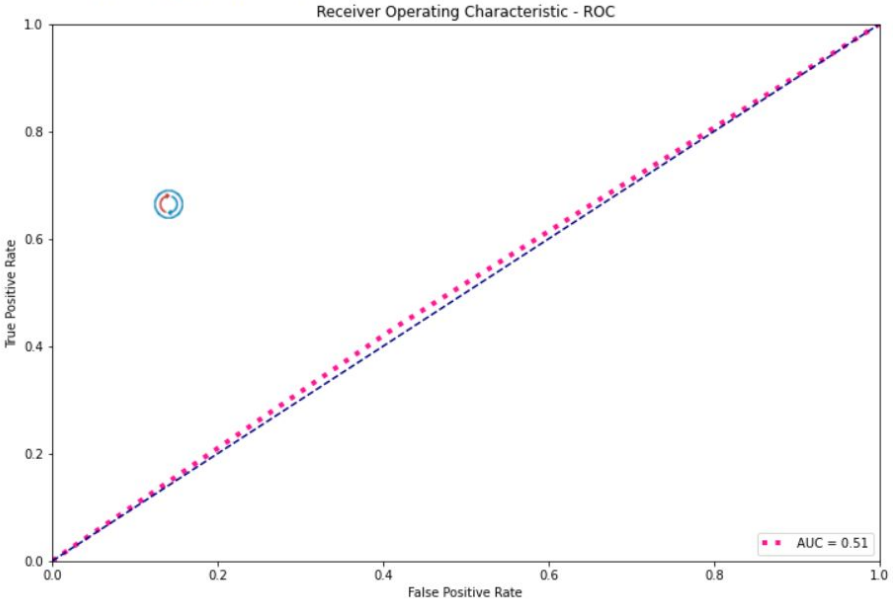
**Upsampling bằng SMOTE

- Logistic Regression

	precision	recall	f1-score	support
0	0.98	0.59	0.73	10756
1	0.02	0.43	0.03	182
accuracy			0.58	10938
macro avg	0.50	0.51	0.38	10938
weighted avg	0.97	0.58	0.72	10938

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/depreca
  warnings.warn(msg, category=FutureWarning)
```

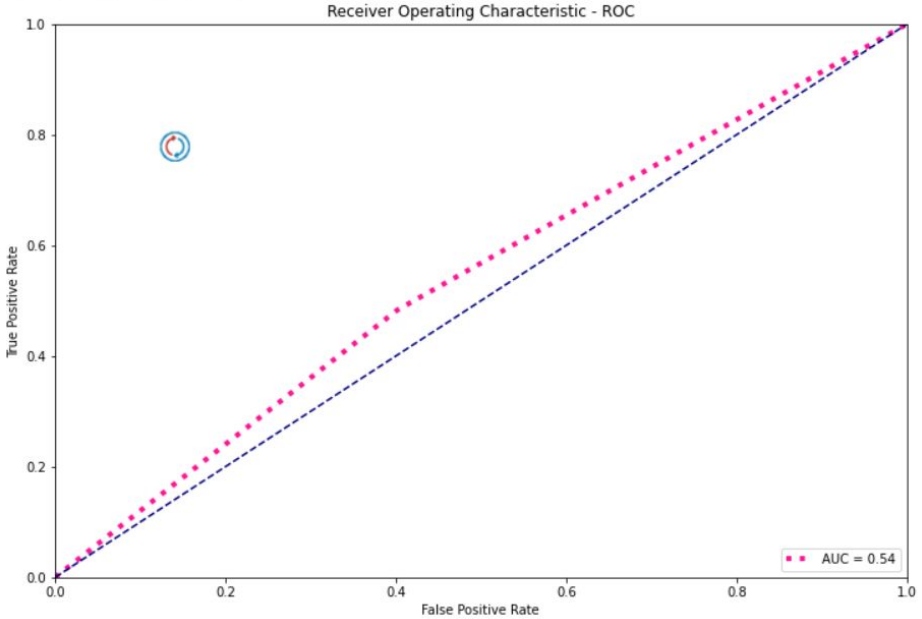




- SVM

```
model = svm.SVC(C = 0.8,
                kernel='linear')
model.fit(X_train_resample, Y_train_resample)
y_predict_1 = model.predict(X_test)

print('Accuracy Score is {:.5}'.format(accuracy_score(Y_test, y_predict_1)))
print(pd.DataFrame(confusion_matrix(Y_test,y_predict_1)))
```

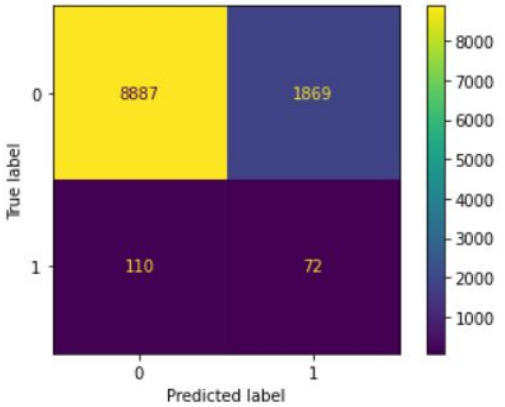
```
Accuracy Score is 0.596
      0     1
0  6431  4325
1    94    88
```

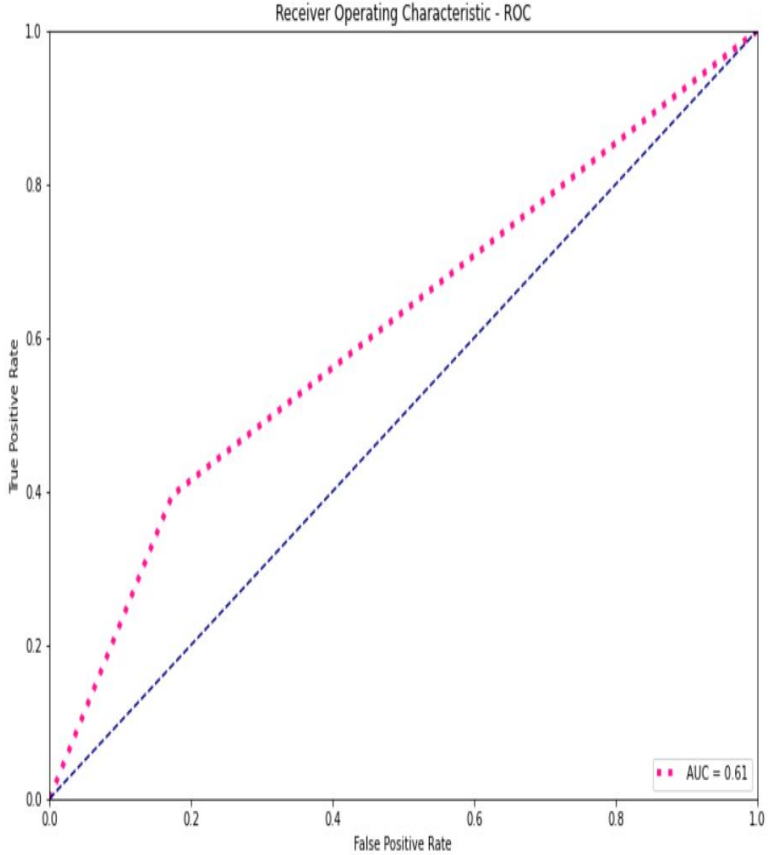


- Decision Tree

```
model = DecisionTreeClassifier(max_depth=12,
                               min_samples_split=8,
                               random_state=1024)
model.fit(X_train_resample, Y_train_resample)
y_predict = model.predict(X_test)
print(classification_report( Y_test, y_predict))
plot_confusion_matrix(model, X_test, Y_test)

print('Accuracy Score is {:.5}'.format(accuracy_score(Y_test, y_predict)))
print(pd.DataFrame(confusion_matrix(Y_test,y_predict)))
```

```
              precision    recall  f1-score   support

           0       0.99      0.83      0.90     10756
           1       0.04      0.40      0.07       182

    accuracy                           0.82     10938
   macro avg       0.51      0.61      0.48     10938
weighted avg       0.97      0.82      0.89     10938

Accuracy Score is 0.81907
      0     1
0  8887  1869
1   110    72
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87:
  warnings.warn(msg, category=FutureWarning)
```

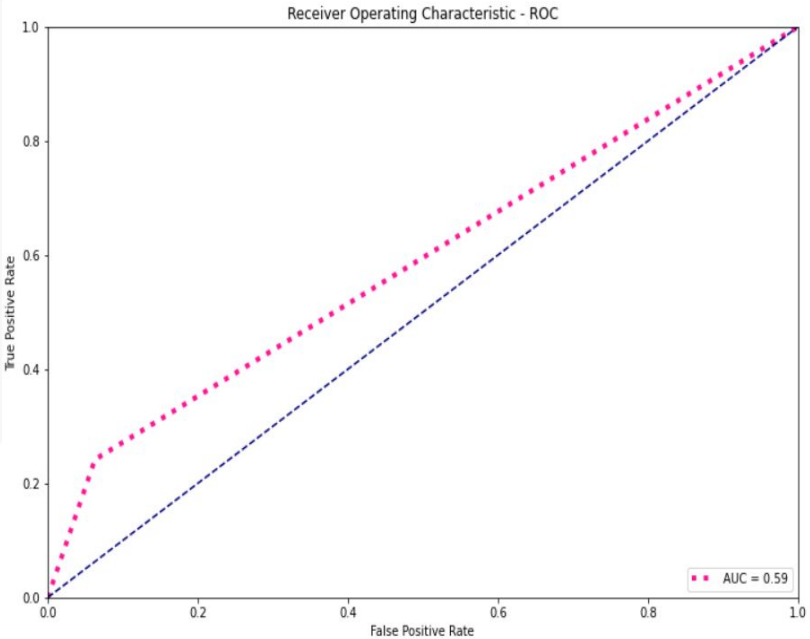




- Random Forest

```
model = RandomForestClassifier(n_estimators=250,
                               max_depth=12,
                               min_samples_leaf=16
                               )
model.fit(X_train_resample, Y_train_resample)
y_predict_2 = model.predict(X_test)

print('Accuracy Score is {:.5}'.format(accuracy_score(Y_test, y_predict_2)))
print(pd.DataFrame(confusion_matrix(Y_test,y_predict_2)))
```

```
Accuracy Score is 0.92558
       0    1
0  10080  676
1    138   44
```



THANK YOU FOR YOUR ATTENTION!!