Credit Risk Analysis and Prediction using Machine Learning

ID/X Partners Data Scientist Virtual Internship Program

Mahmudin Rizal

Rakamin Academy

id/x partners

Outline

Business Understanding

Credit risk refers to the possible loss due to the borrower's failure to make payments on any type of debt.
Credit risk management, meanwhile, is the practice of mitigating these losses by understanding a bank's capital adequacy and loan loss reserves at any given time – a process that has long been a challenge for financial institutions.

Objective this Project

The Value

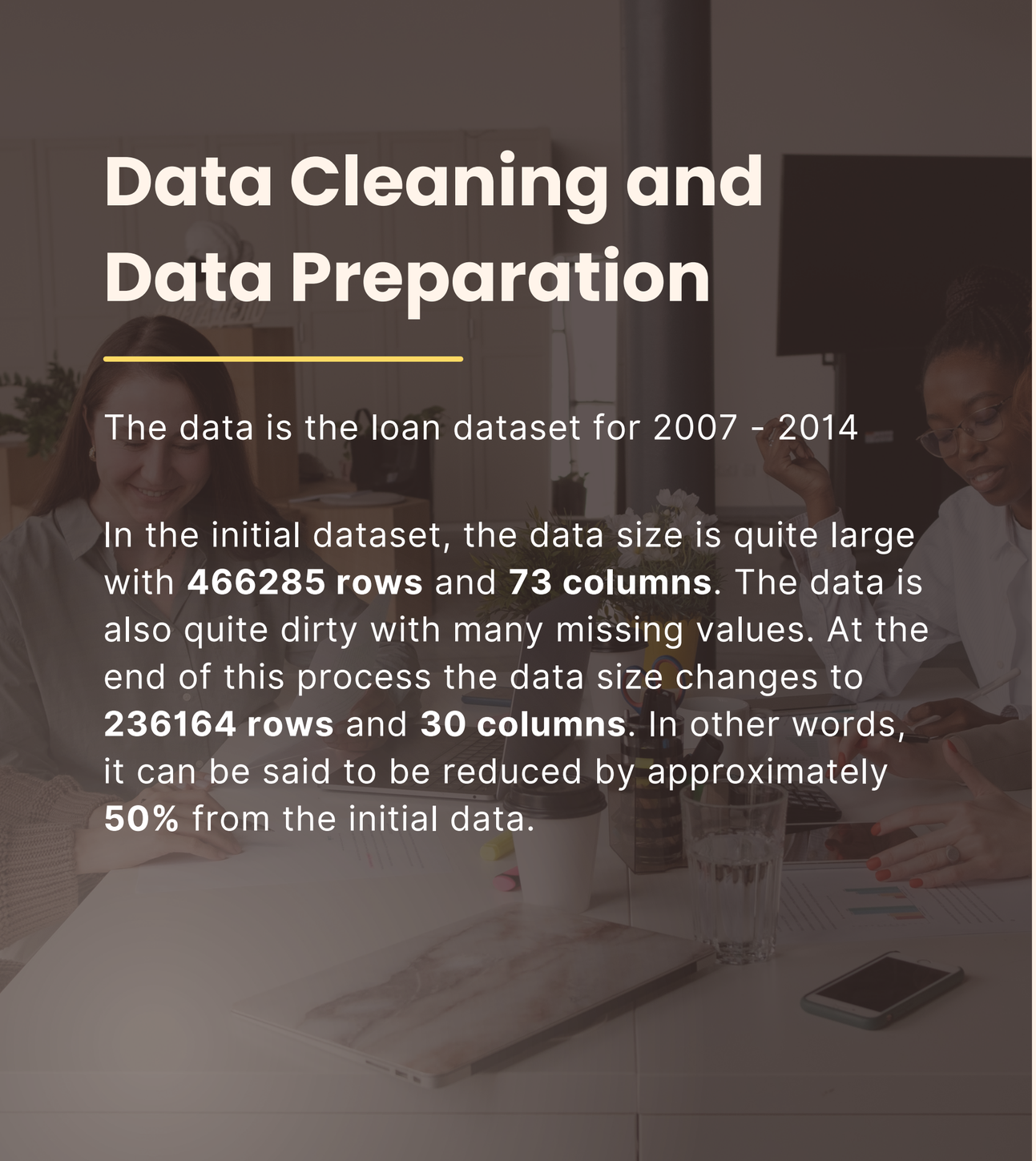
Credit risk refers to the possible loss due to the borrower's failure to make payments on any type of debt.

The Vision

Provide insight for future credit risk needs

The Target

The goal is to build a model that can predict credit risk using a dataset provided by the company consisting of data on loans received and rejected.



Data Cleaning and Data Preparation

The data is the loan dataset for 2007 - 2014

In the initial dataset, the data size is quite large with **466285 rows** and **73 columns**. The data is also quite dirty with many missing values. At the end of this process the data size changes to **236164 rows** and **30 columns**. In other words, it can be said to be reduced by approximately **50%** from the initial data.

01

Handling Missing Values

Remove columns or features that have a number of missing values and zeros more than **15%** of the total data.

02

Remove Meaningless Features

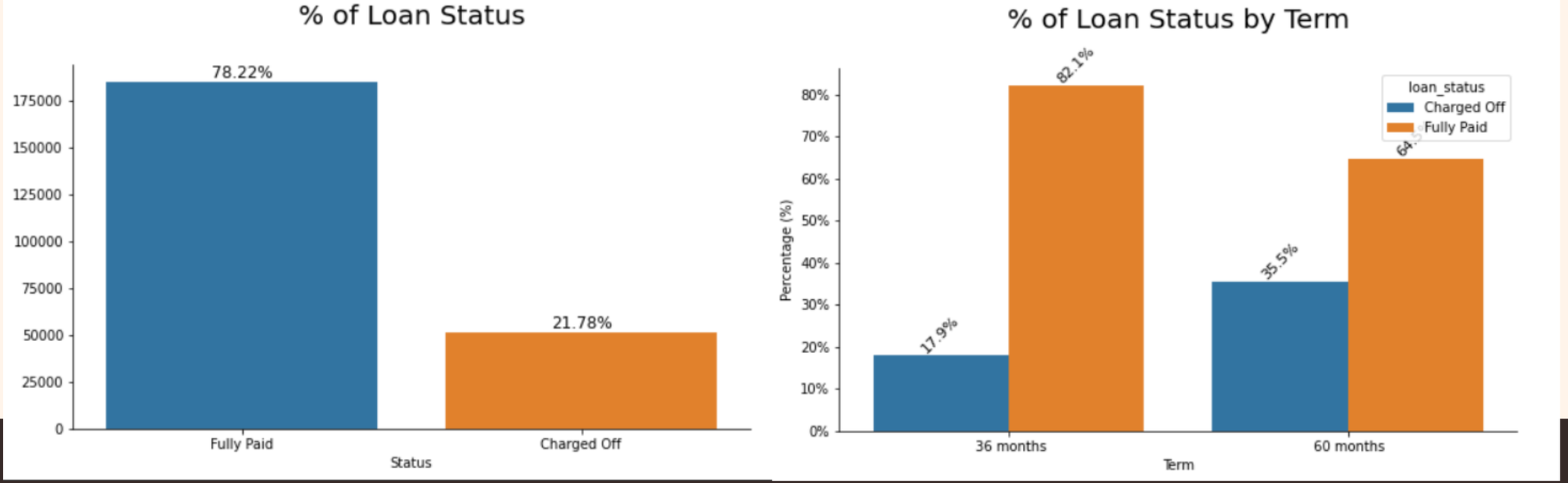
Remove columns or features that have columns that are assumed to have less meaning, for example features that have the same value for all data.

03

Handling Target Features

Due to lack of in-depth knowledge of loan status, it is assumed that Late (31-120 days), Late (16-30 days), and Default features are charged off. The rest other than fully paid is removed. The final form of Target has 2 categories, namely **fully paid and charged off**

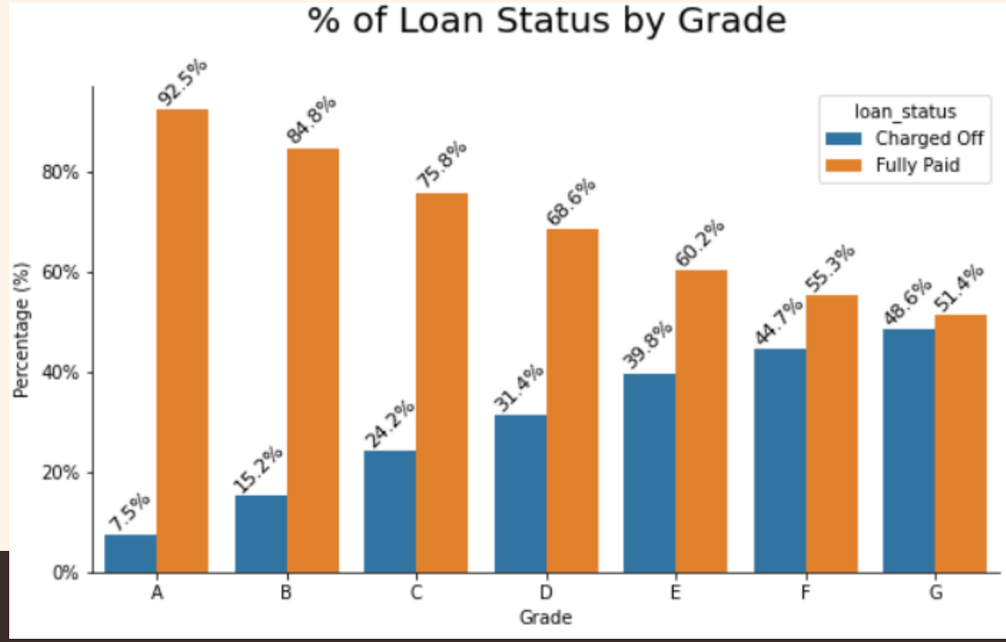
Exploratory Data Analysis

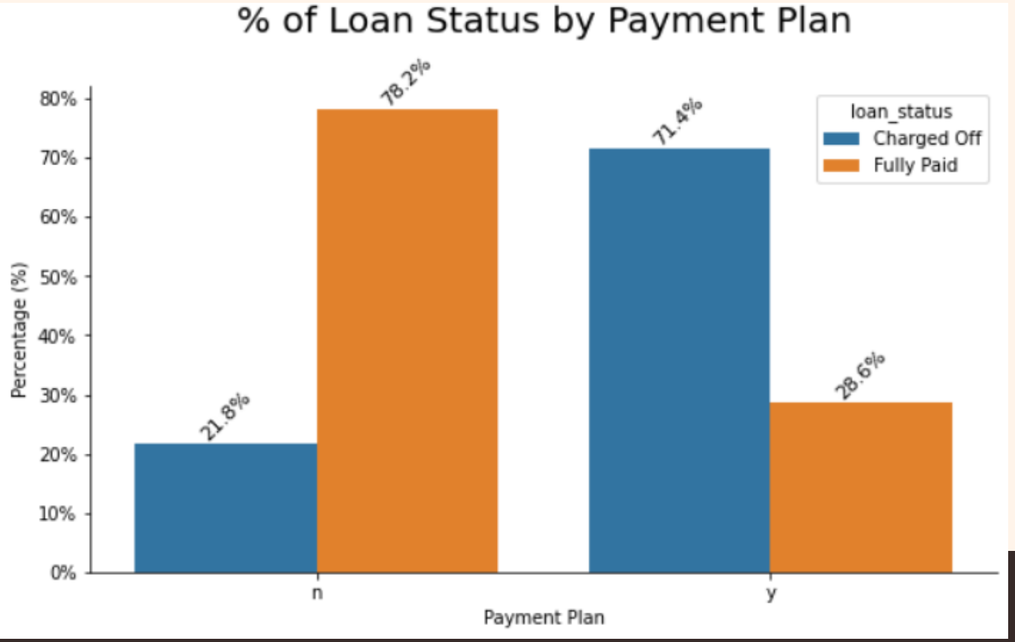


It can be seen that this data target is not balanced as much as 78.22% and 21.78%.

In % loan status based on term, it can be seen that the higher the term, the higher the % charged off level

Exploratory Data Analysis





In % loan status based on grade, it can be seen that the higher the grade A-G, the higher the percentage of charged off that occurs.

In % loan status based on the payment plan, it can be seen that the payment plan "n" has a higher fully paid rate than charged off. This is actually inversely proportional to the "y" payment plan.

Features Engineering

Handling Categorical Features

- Filling Missing Value using Mode
- Label Encoding
- One-Hot Encoding

Handling Numerical Features

- Filling Missing Value using Median

Handling Target Features

- Mapping Fully paid and Charged Off to 1 and 0

Features Engineering

Handling Categorical Features

```
df_model['emp_length'] = df_model['emp_length'].fillna(df_model['emp_length'].mode()[0])
```

```
[ ]  label_encoder = preprocessing.LabelEncoder()

     label_enc_features = ['grade', 'emp_length', 'home_ownership']
     for col in label_enc_features:
         df_model[col]= label_encoder.fit_transform(df_model[col])
```

```
     df_onehot = pd.get_dummies(df_model[['term']], prefix='term', prefix_sep="_")
     df_onehot2 = pd.get_dummies(df_model[['verification_status']], prefix="verification_status", prefix_sep="_")
     df_onehot3 = pd.get_dummies(df_model[['pymnt_plan']], prefix="pymnt_plan", prefix_sep="_")
     df_onehot4 = pd.get_dummies(df_model[['purpose']], prefix="purpose", prefix_sep="_")
     df_onehot5 = pd.get_dummies(df_model[['addr_state']], prefix="addr_state", prefix_sep="_")
     df_onehot6 = pd.get_dummies(df_model[['initial_list_status']], prefix="initial_list_status", prefix_sep="_")

     df_final_onehot = df_onehot.join([df_onehot2, df_onehot3, df_onehot4, df_onehot5, df_onehot6])
```

Features Engineering

Handling Numerical Features

```
median_value = df_model['revol_util'].median()
df_model['revol_util'].fillna(value = median_value, inplace = True)
```

Handling Target Features

```
df_model['loan_status'].value_counts()
```

```
Fully Paid     184739
Charged Off     51425
Name: loan_status, dtype: int64
```

```
df_model['loan_status'] = df_model['loan_status'].map({'Fully Paid': 1, 'Charged Off': 0})
```

```
df_model['loan_status'].value_counts()
```

```
1    184739
0     51425
Name: loan_status, dtype: int64
```

Modelling ML and Evaluation

Train Test Split

The data is split to **80% for training** and **20% for testing**

Standar scalling and oversampling

Data is also scaling with **StandardScaler** and oversampling due to unbalanced target with **SMOTE**

Machine Learning

Using 3 supervised machine learning for classification prediction, namely **Logistic Regression, Random Forest Classifier, and Decission Tree Classifier.**

Evaluation Model

The result is good, the three methods are not much different. **Random Forest Classifier gets the best score**

Result of Machine Learning

	precision	recall	f1-score	support
0	0.99	0.98	0.99	10249
1	1.00	1.00	1.00	36984
accuracy			1.00	47233
macro avg	1.00	0.99	0.99	47233
weighted avg	1.00	1.00	1.00	47233

Accuracy Score: 0.9955539559206487

	precision	recall	f1-score	support
0	1.00	0.99	0.99	10249
1	1.00	1.00	1.00	36984
accuracy			1.00	47233
macro avg	1.00	0.99	1.00	47233
weighted avg	1.00	1.00	1.00	47233

Accuracy Score: 0.9968454258675079

	precision	recall	f1-score	support
0	0.99	0.99	0.99	10249
1	1.00	1.00	1.00	36984
accuracy			1.00	47233
macro avg	0.99	0.99	0.99	47233
weighted avg	1.00	1.00	1.00	47233

Accuracy Score: 0.9960620752440031

Logistic Regression

99,6 %

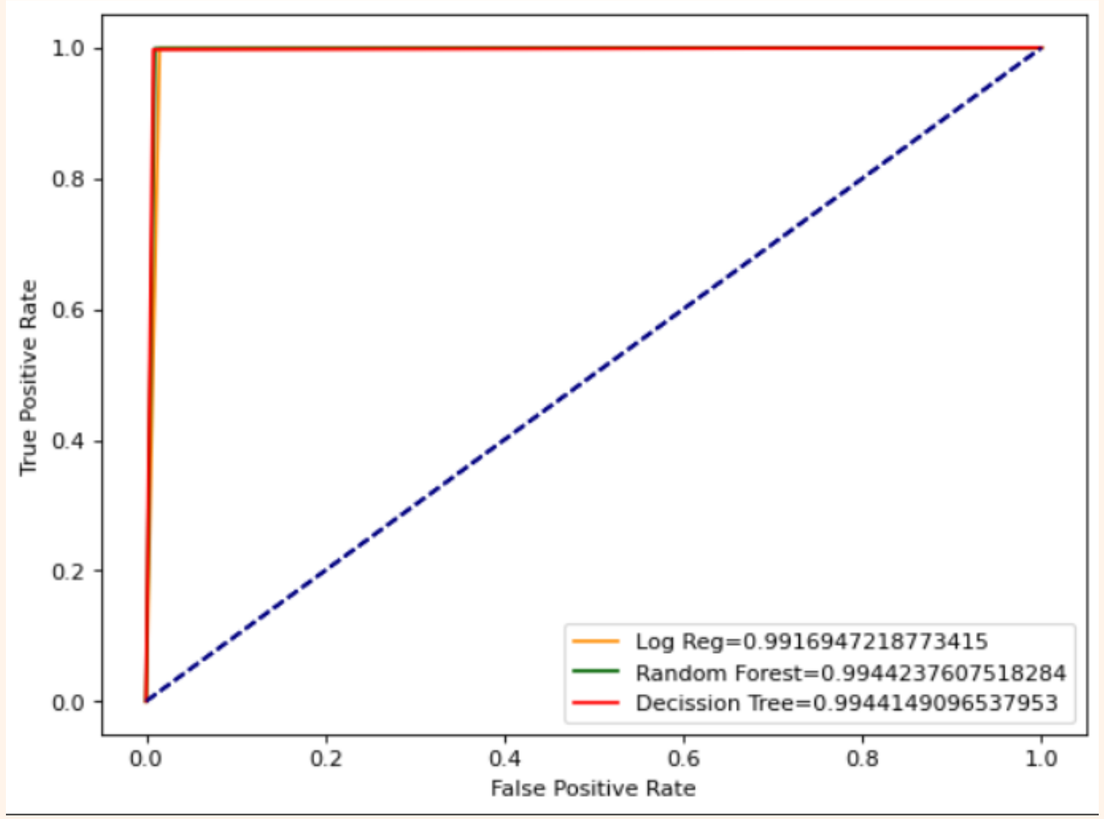
Random Forest Classifier

99,7 %

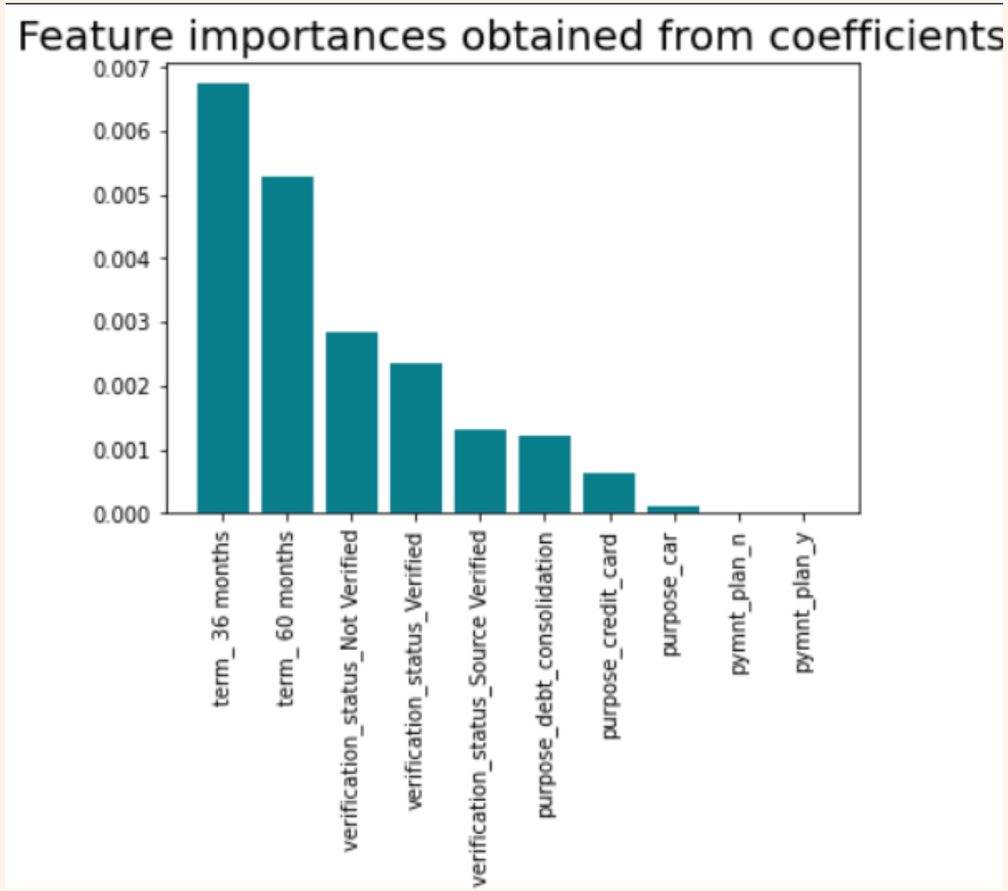
Decission Tree Classifier

99,6 %

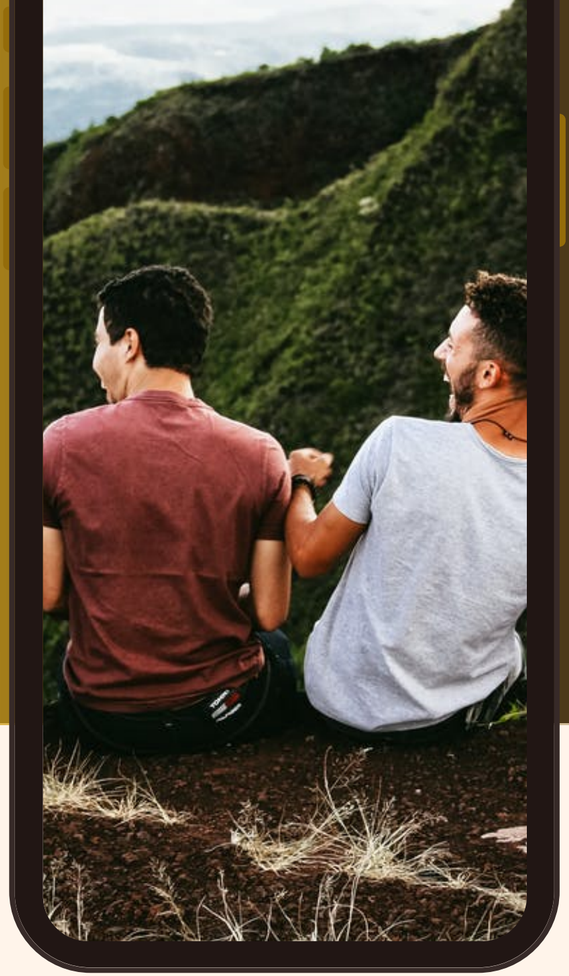
AUC - ROC Curve



TOP 10 Features Importance



Let's Connect!



Instagram

@mahmdnrizal

LinkedIn

https://www.linkedin.com/in/mahmudin-rizal/

Github

https://github.com/mahmudinrizal/Virtual-Internship-Rakamin-Academy/tree/main/IDX-Partners

Medium

https://medium.com/@mahmudinr