# Supervised ML – Classification Capstone Project On Credit Card Prediction

**Prepared By**

**Raja Chowdhury**

# INDEX

# Introduction

Credit card default prediction is the process of using historical data to predict whether or not a credit card holder will default on their payments in the future. Default prediction is important for both lenders and borrowers, as it can help lenders to identify high-risk customers and make better lending decisions, while helping borrowers to understand their chances of default and take steps to avoid it.

# Problem Statement

This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the K-S chart to evaluate which customers will default on their credit card payments.

# Data Summary

- **X1** - Amount of credit limit(includes individual as well as family credit)
- **X2** - Gender
- **X3** - Education
- **X4** - Marital Status
- **X5** - Age
- **X6 to X11** - History of past payments from April to September
- **X12 to X17** - Amount of bill statement from April to September
- **X18 to X23** - Amount of previous payment from April to September
- **Y** - Default payment

# Approach Overview

## ***Data Cleaning and Understanding***

- ❑ Find information on documented columns values
- ❑ Clean data to get it ready for Analysis

## ***Data Exploration (EDA)***

- ❑ Examining the data with visualization
- ❑ Plotting graphs

## ***Modeling (Machine Learning)***

- Logistic Regression
- Support Vector Classifier
- Decision Tree Classifier
- Random Forest Classifier
- XGBoost

## ***METRICS***

1. Confusion Matrix
2. Accuracy Score
3. Precision Score
4. Recall Score
5. F1 Score
6. Roc_Auc Score

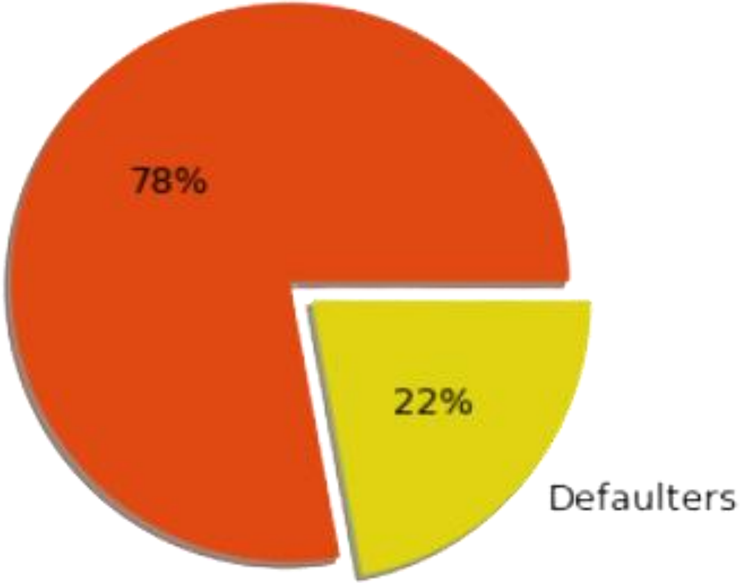
# Basic Exploration

- ✓ Dataset for Taiwan.
- ✓ Shape of data is 30000 rows and 25 columns
- ✓ Six months payment and bill data available.
- ✓ No null data.
- ✓ Nine Categorical variables present.
- ✓ ID column can be drop

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   30000 non-null  int64
 2   SEX                         30000 non-null  int64
 3   EDUCATION                   30000 non-null  int64
 4   MARRIAGE                    30000 non-null  int64
 5   AGE                         30000 non-null  int64
 6   PAY_0                       30000 non-null  int64
 7   PAY_2                       30000 non-null  int64
 8   PAY_3                       30000 non-null  int64
 9   PAY_4                       30000 non-null  int64
 10  PAY_5                       30000 non-null  int64
 11  PAY_6                       30000 non-null  int64
 12  BILL_AMT1                   30000 non-null  int64
 13  BILL_AMT2                   30000 non-null  int64
 14  BILL_AMT3                   30000 non-null  int64
 15  BILL_AMT4                   30000 non-null  int64
 16  BILL_AMT5                   30000 non-null  int64
 17  BILL_AMT6                   30000 non-null  int64
 18  PAY_AMT1                    30000 non-null  int64
 19  PAY_AMT2                    30000 non-null  int64
 20  PAY_AMT3                    30000 non-null  int64
 21  PAY_AMT4                    30000 non-null  int64
 22  PAY_AMT5                    30000 non-null  int64
 23  PAY_AMT6                    30000 non-null  int64
 24  default payment next month  30000 non-null  int64
dtypes: int64(25)
memory usage: 5.7 MB
```

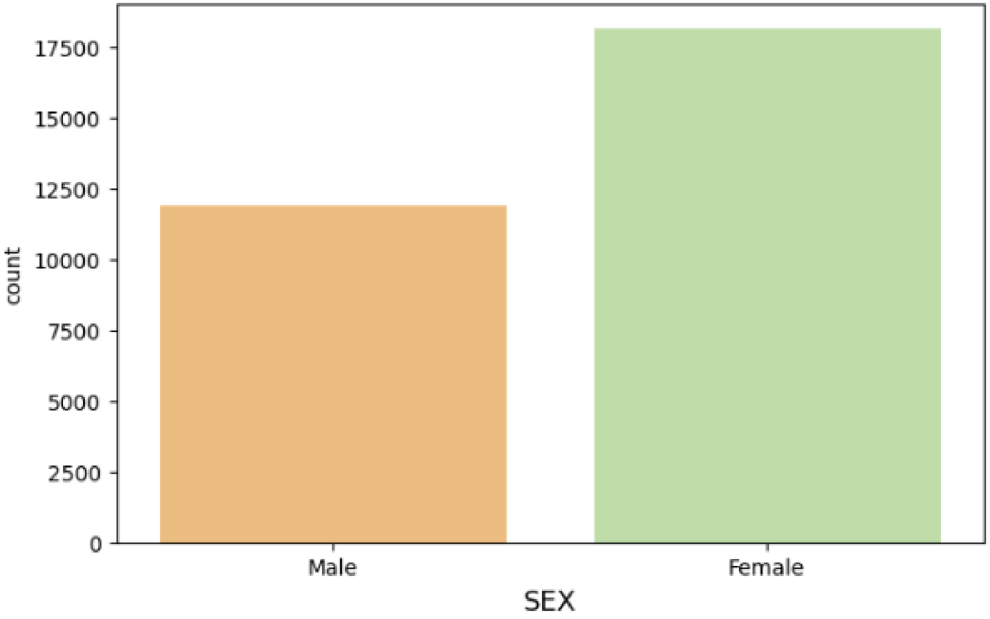
# Ratio of Defaulter to Non-Defaulter

In our dataset the ratio of defaulter to non defaulter is **78:22.** That is **22%** are defaulters while the rest pays on time.



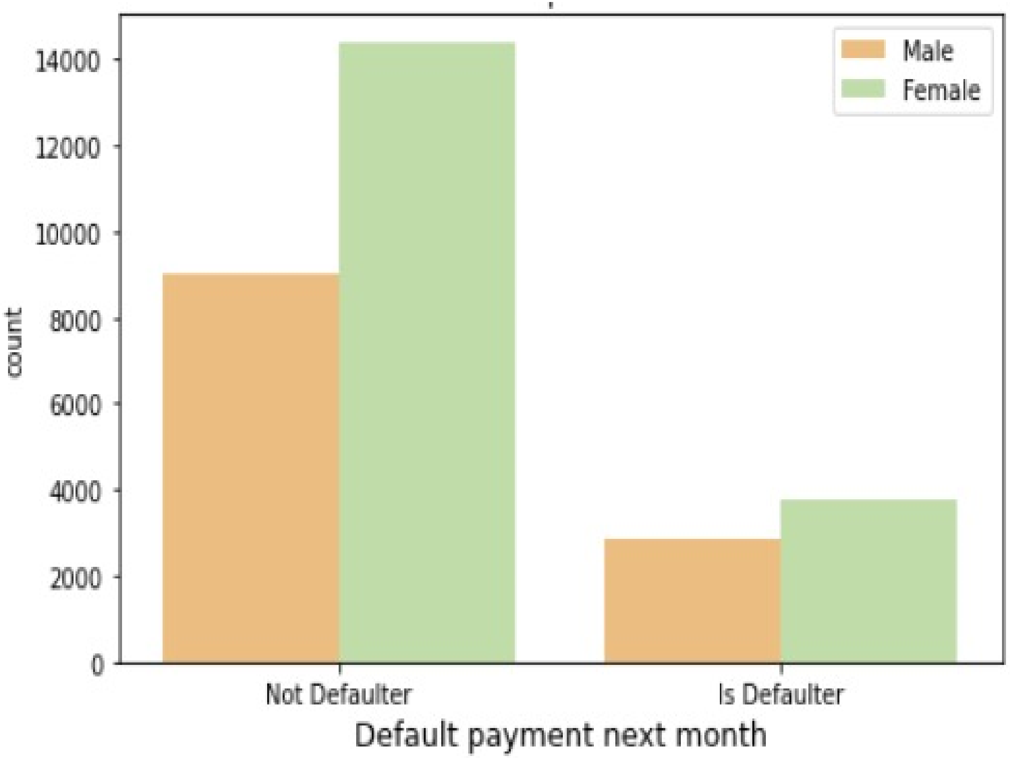
# Gender Distribution

In our dataset,
There are **11,888** Males or **39%** and **18,112** Females or **61%.**



# Defaulter Ratio With Respect To Gender

The Defaulter ratio with respect to gender is **43%** of Males are defaulter while **57%** of Females are defaulter.



# Defaulter Ratio With Respect To Marital Status

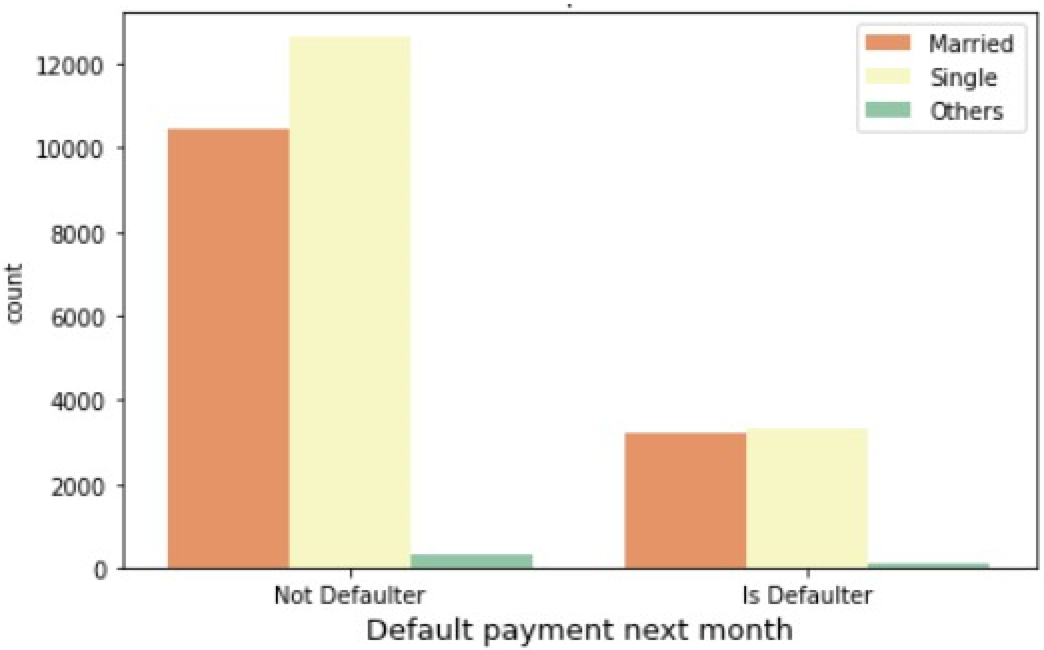
The defaulter ratio according to marital status are as follows:

**Married :** 50%

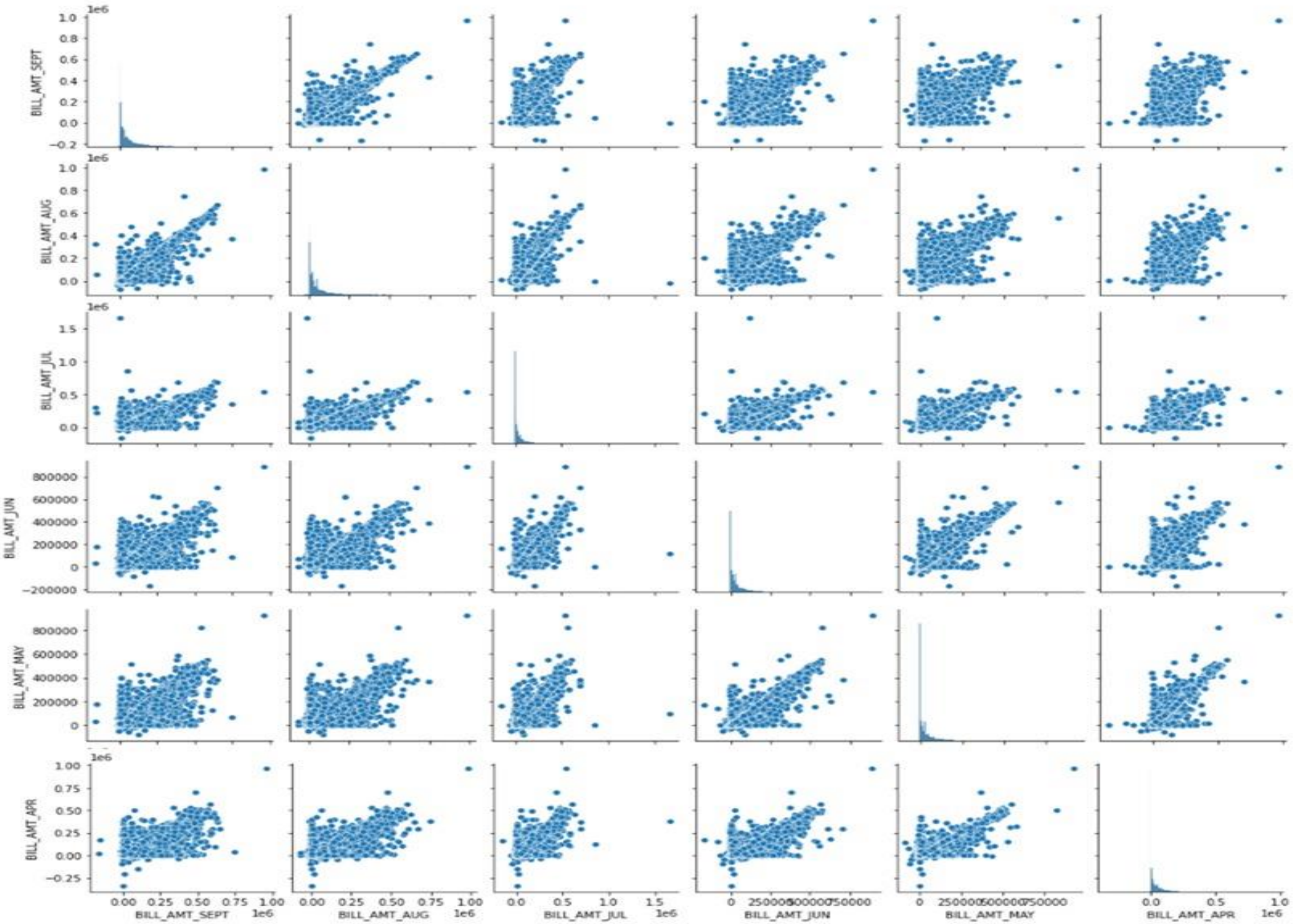
**Unmarried :** 49%

**Single :** 1%

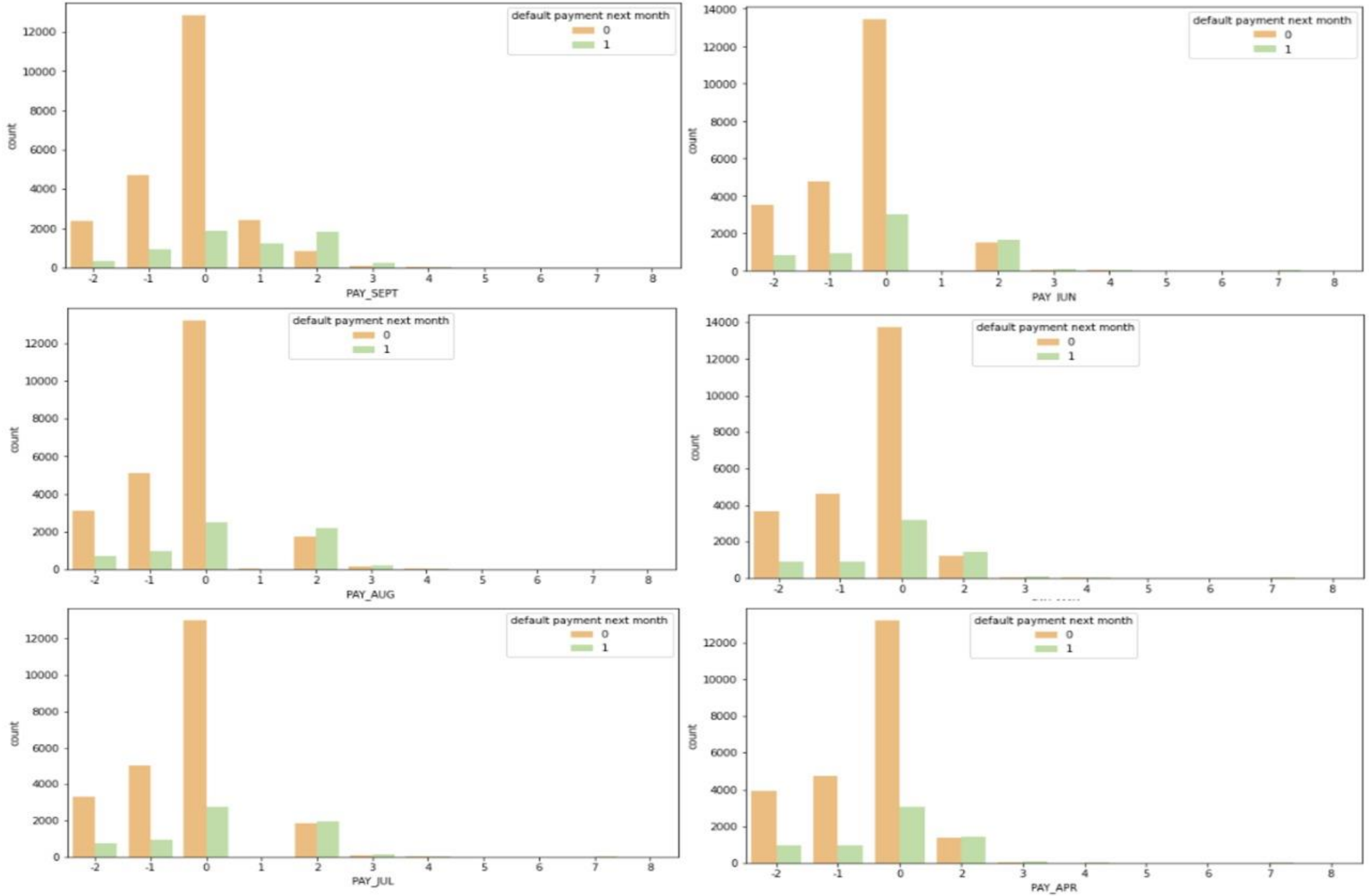
The above calculation says that **Married** people are more likely to fail to pay on time while **Single People** often pays on time.



# History Of Bill Amount Payment



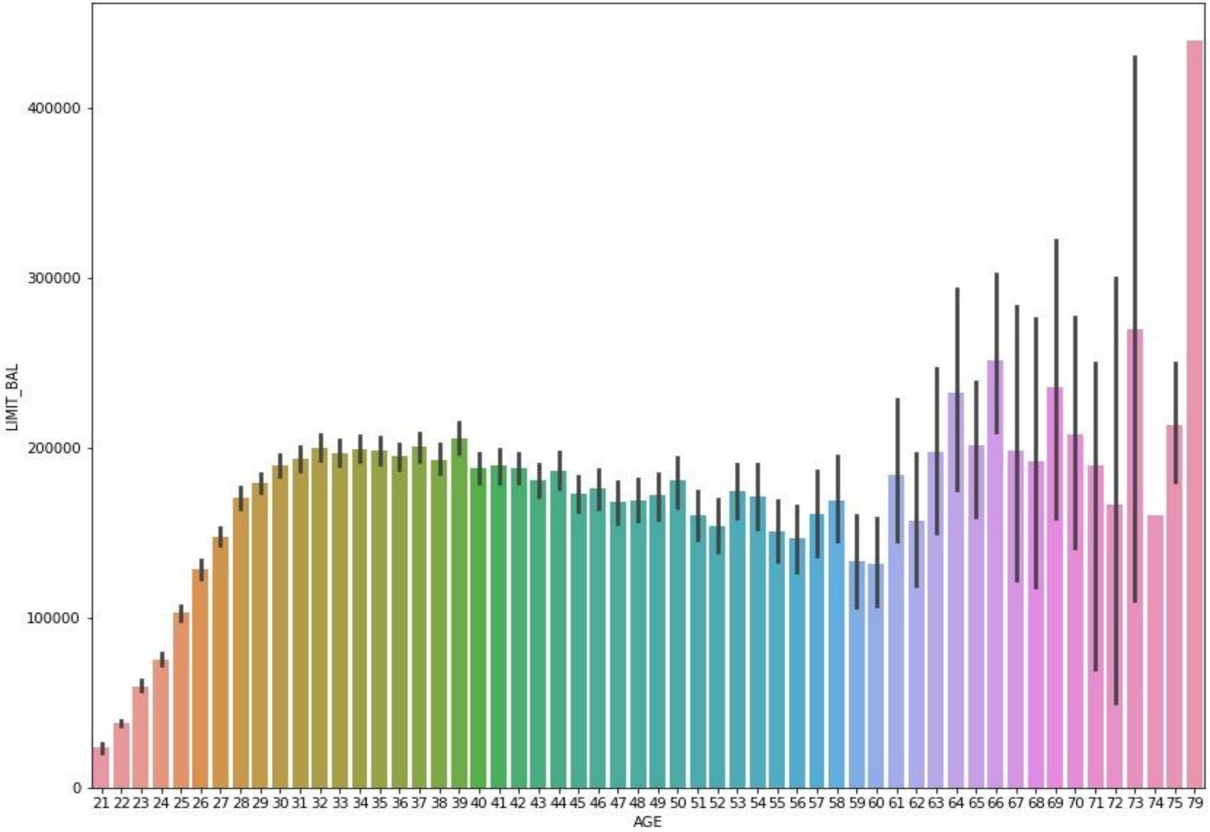
# History Of Pay Amount Payment



# Allotment Of Credit Limit Balance

Trend of limit balance is mixed from age 21 to 39, it is increasing after it has declined a bit but from 62 it is increasing the limit has increased drastically.

*Highest balance given to the age of 79.*



# Modeling Overview

Supervised learning/Binary Classification

**Imbalance data** with **78%** non-defaulters and **22%** defaulters

***Modeling (Machine Learning)***

- Logistic Regression
- Support Vector Classifier
- Decision Tree Classifier
- Random Forest Classifier
- XGBoost

***METRICS***

1. Confusion Matrix
2. Accuracy Score
3. Precision Score
4. Recall Score
5. F1 Score
6. Roc_Auc Score

# Modeling Steps

- **Data Preprocessing**
- ❑ Feature selection and Feature engineering
- ❑ Train test data split(**80%-20%**)
- ❑ SMOTE oversampling(Synthetic Minority Oversampling Technique)
- **Data Fitting and Tuning**
- ❑ Start with default model parameters
- ❑ Hyperparameter tuning
- ❑ Measure AUC- ROC on training data
- **Model Evaluation**
- ❑ Model testing
- ❑ Precision Recall Score
- ❑ Compare with the other models

# Logistic Modeling

**PARAMETERS:** **C** = 0.0, **Max_iter** = 50, **Penalty** = none

**RESULTS:**

- *The accuracy on test data* : **56%**
- *The precision on test data* : **36%**
- *The recall on test data* : **60%**
- *The f1 score on test data* : **56%**
- *The roc_auc on test data* : **57%**

# Support Vector Classification

**PARAMETERS:** **C** = 0.5

**RESULTS:**

- *The precision on test data* : **60%**
- *The recall on test data* : **69%**
- *The accuracy on test data* : **61%**
- *The f1 score on test data* : **64%**
- *The roc_auc on test data* : **51%**

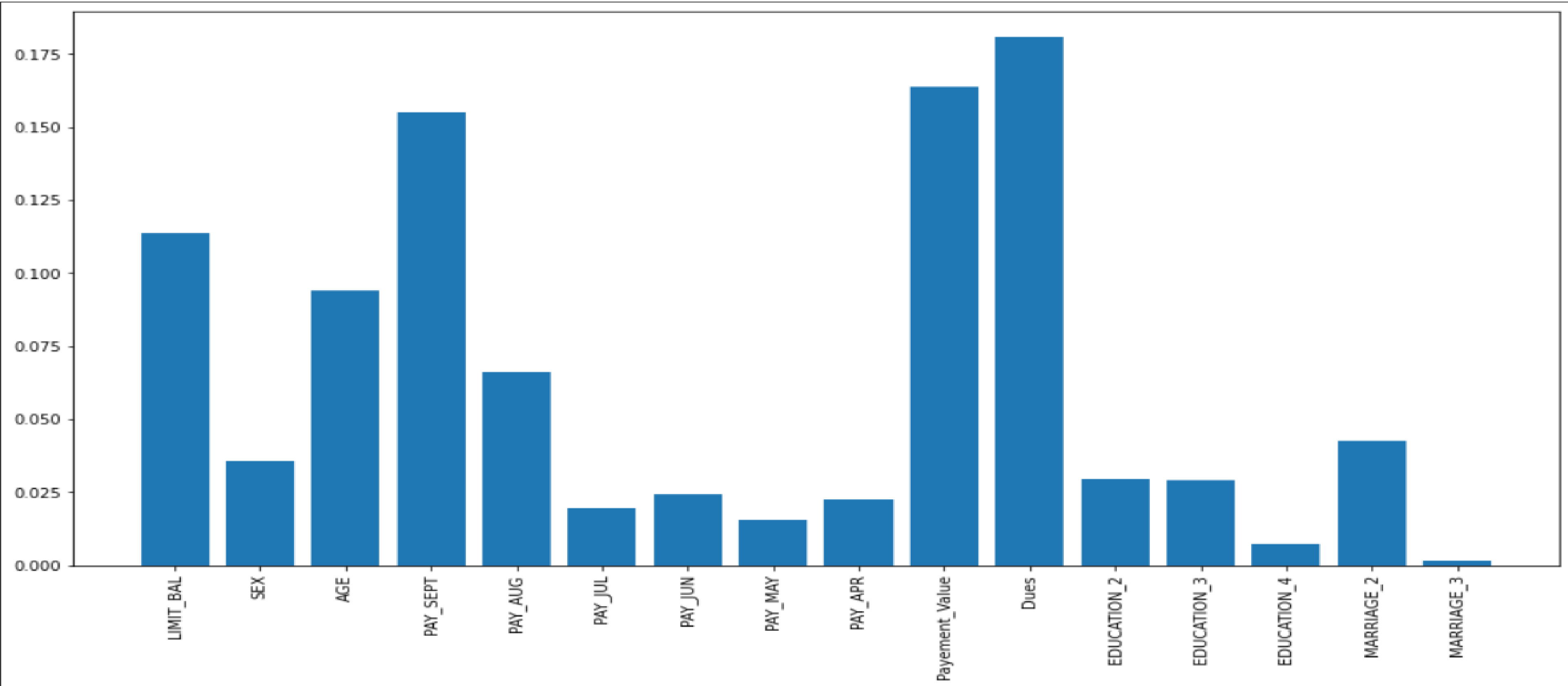
# Decision Tree Classifier

**PARAMETERS:** **min_sample_leaf** = 8, **min_sample_split** = 2

**RESULTS:**

- *The precision on test data* : **60%**
- *The recall on test data* : **69%**
- *The accuracy on test data* : **61%**
- *The f1 score on test data* : **64%**
- *The roc_auc on test data* : **51%**

# Feature Importance Of Decision Tree



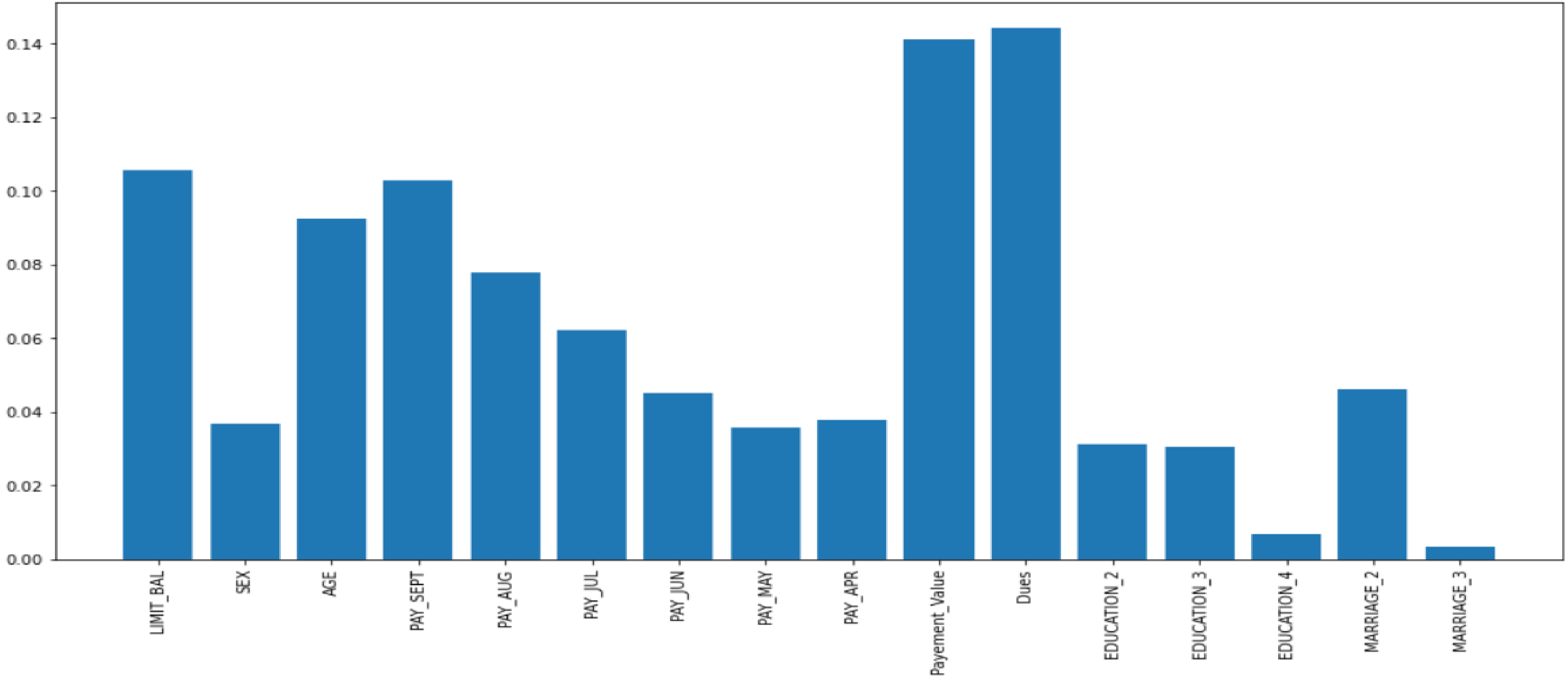
# Random Forest Classifier

**PARAMETERS:** **min_sample_leaf** = 1, **min_sample_split** = 2

**RESULTS:**

- *The precision on test data* : **86%**
- *The recall on test data* : **86%**
- *The accuracy on test data* : **85%**
- *The f1 score on test data* : **85%**
- *The roc_auc on test data* : **85%**

# Feature Importance Of Random Forest



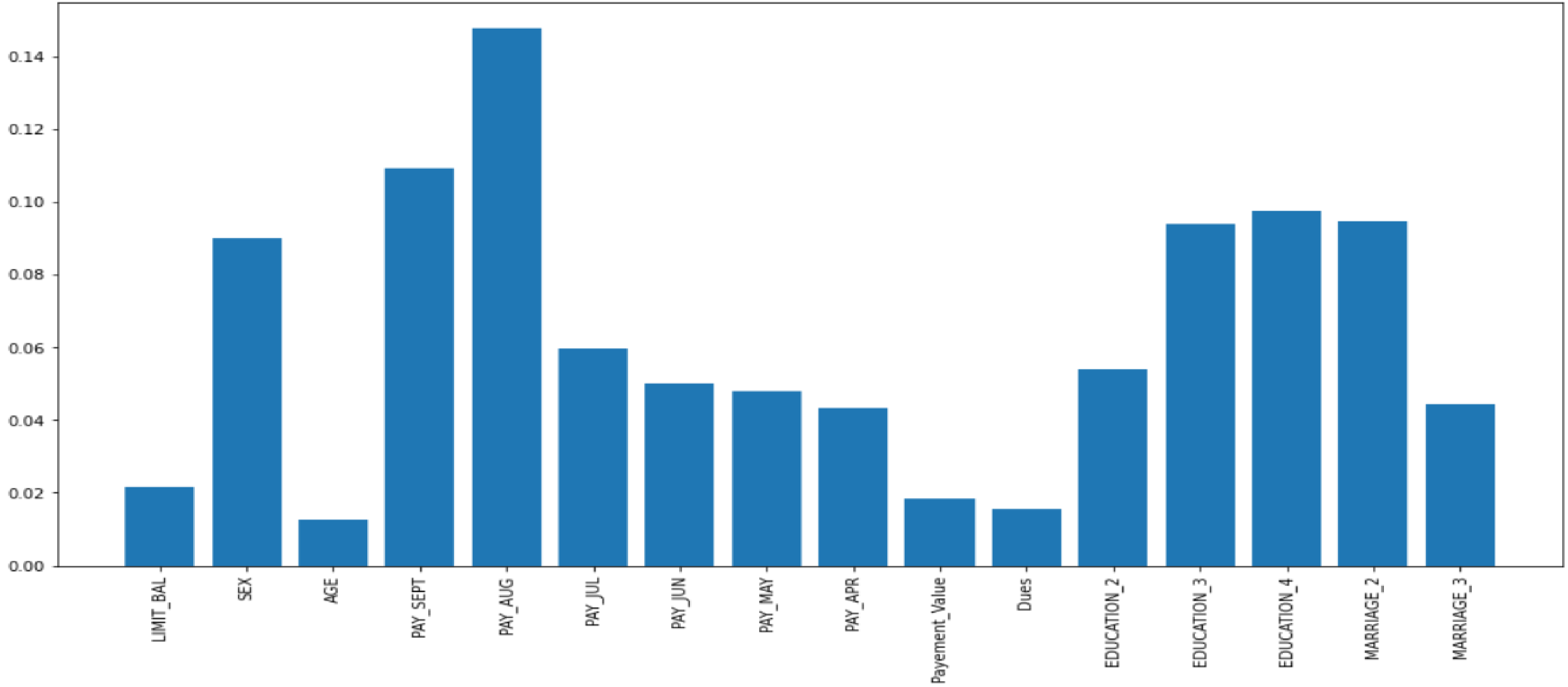
# Xgboost

**PARAMETERS:** **learning_rate = 0.30000000000000004**

**RESULTS:**

- *The precision on test data* : **84%**
- *The recall on test data* : **77%**
- *The accuracy on test data* : **81%**
- *The f1 score on test data* : **80%**
- *The roc_auc on test data* : **81%**

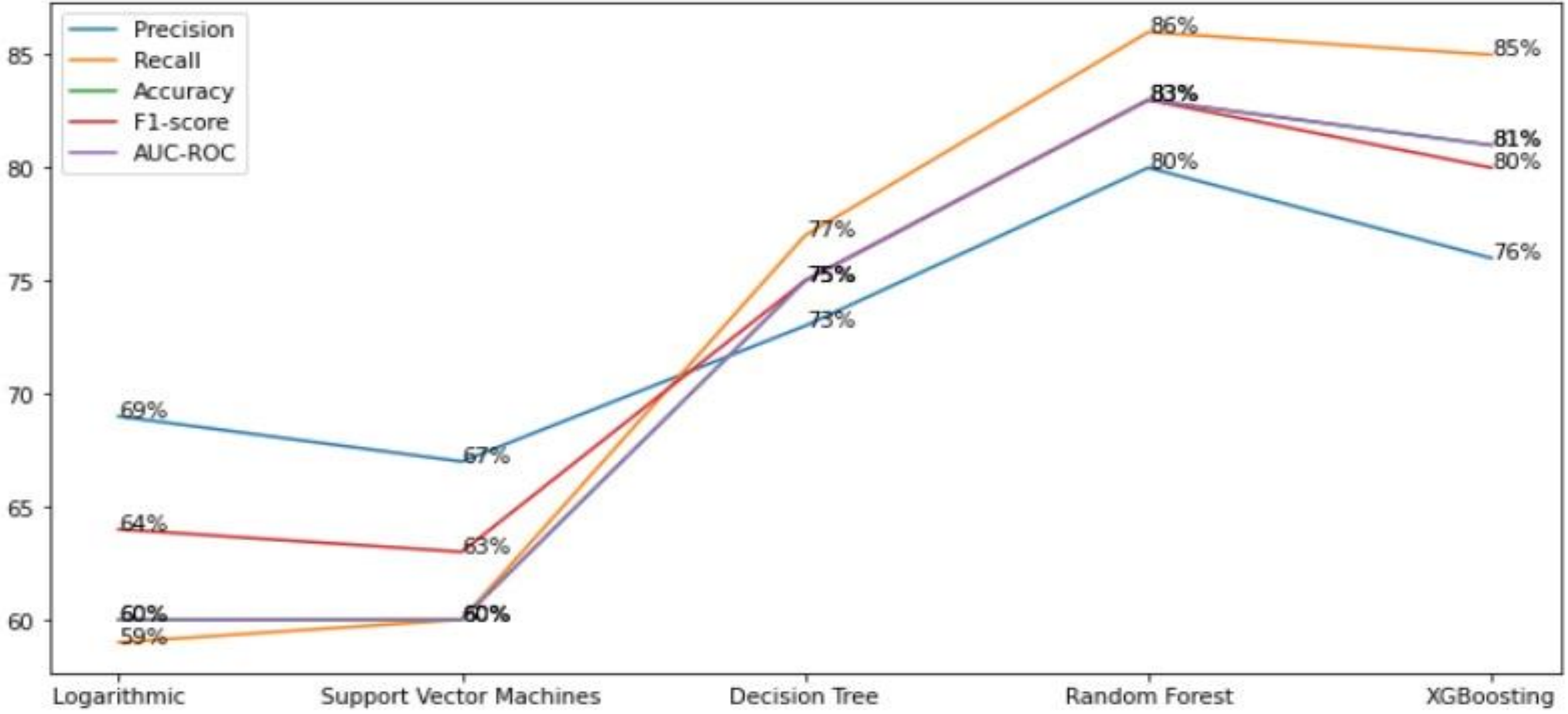
# Feature Importance Of Xgboost



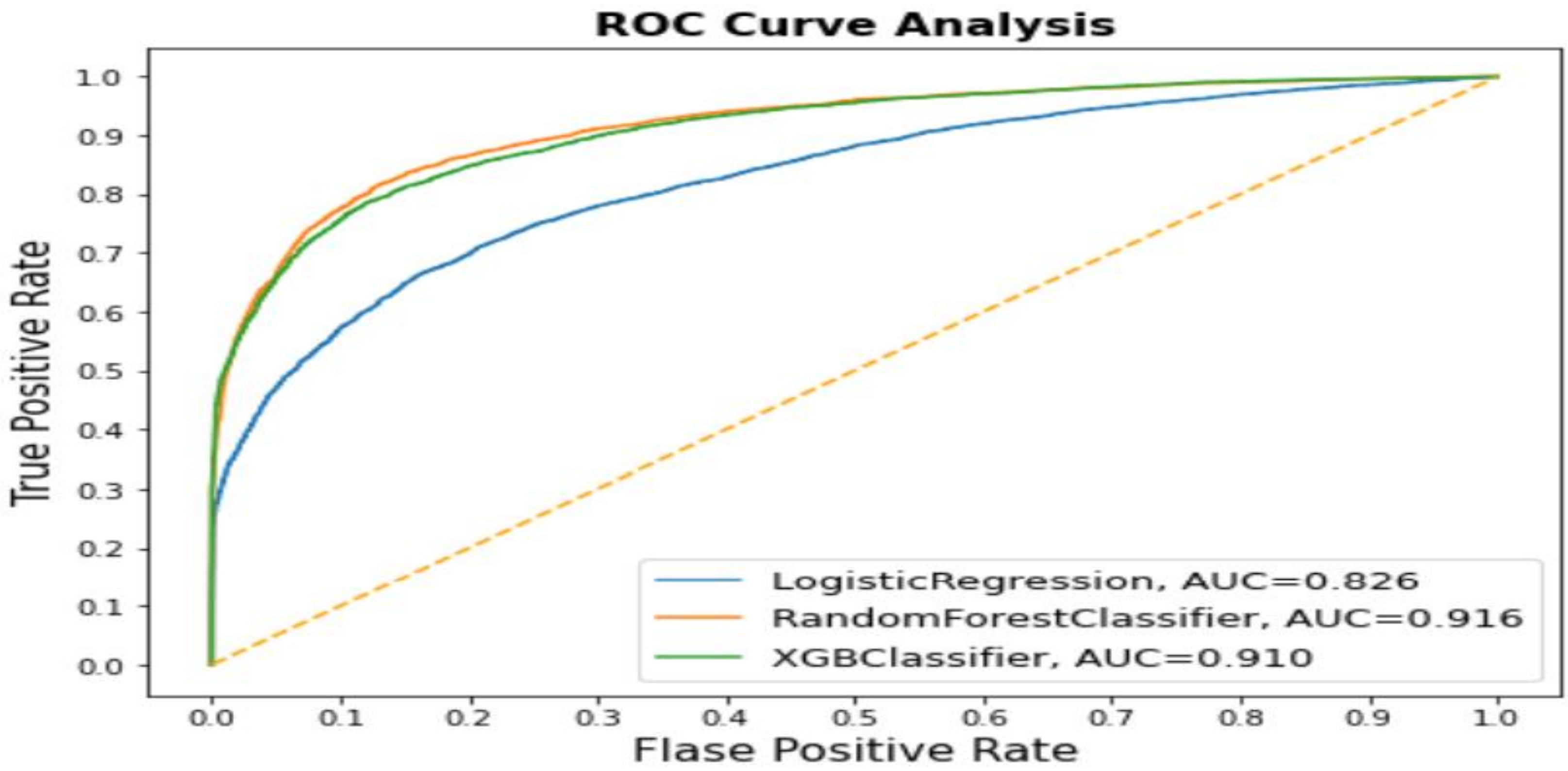
# Challenges

- 9 Categorical variables present.
- Understanding the dataset.
- Cleaning dataset.
- Feature engineering.
- Selecting model.
- Getting a higher accuracy due to data leakage.

# Performance Of Different Metrics



# AUC_ROC Curve Comparision



# Conclusion

- Random forest is the best algorithm for our model. Recall is **86%(***meaning out of* ***100*** *defaulters* ***86*** *will be correctly caught by Random forest*)
- Support Vector Machines has the least recall score of **60%**

| SR.NO | CLASSFIER | ACCURACY | PRECISION | RECALL | F1 SCORE | ROC_AUC |
|---|---|---|---|---|---|---|
| 01 | *LOGISTIC REGRESSION* | 56% | 36% | 60% | 56% | 57% |
| 02 | *SVC* | 61% | 60% | 69% | 64% | 51% |
| 03 | *DECISION TREE* | 61% | 60% | 69% | 64% | 51% |
| **04** | ***RANDOM FOREST*** | **85%** | **86%** | **86%** | **85%** | **85%** |
| 05 | *XGBOOST* | 81% | 84% | 77% | 80% | 81% |

# Thank You