Capstone Project

Credit Card Default Prediction

Content

- Introduction
- Problem Statement ● Data Summary
- Approach Overview
- Exploratory Data Analysis
- Modelling Overview
- Feature Importances
- Challenges ● Conclusion

Introduction

In today's world credit cards have become a lifeline to a lot of people so banks provide us with credit cards. Now we know the most common issue there is in providing these kind of deals are people not being able to pay the bills. These people are what we call "defaulters".

Problem Statement

Predicting whether a customer will default on his/her credit card.

Data Summary

- X1 - Amount of credit(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

Understanding and Cleaning

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

Data Exploration

- **Graphical**

Examining the data with visualization

Modeling

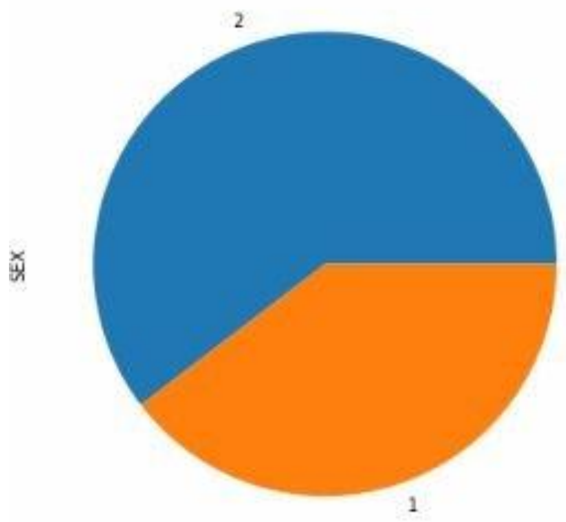
Machine Learning

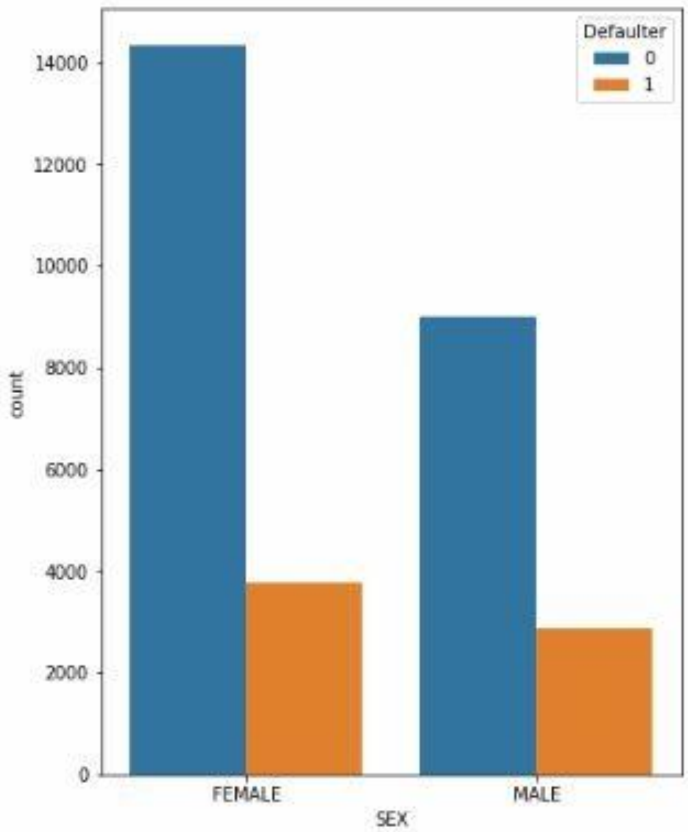
- Logistic
- Random Forest
- XGBoost

Basic Exploration

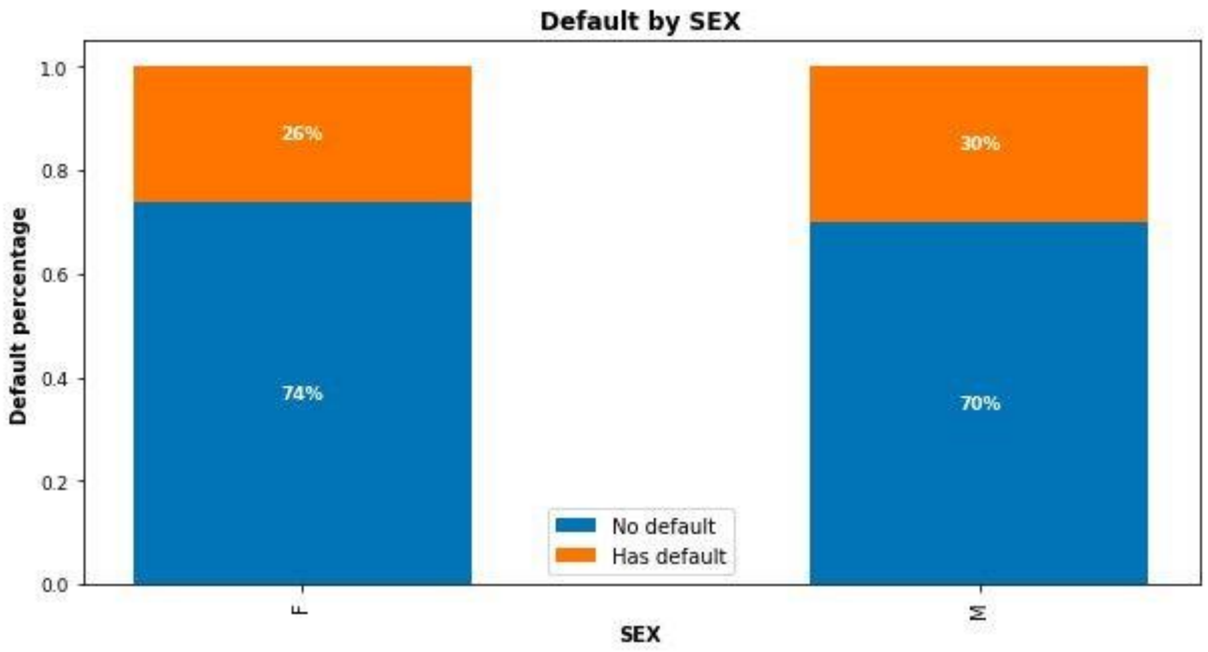
- Dataset for Taiwan.
- Data for 30000 customers.
- 6 Months payment and bill data available.
- No null data.
- 9 Categorical variables present.

Gender Distribution



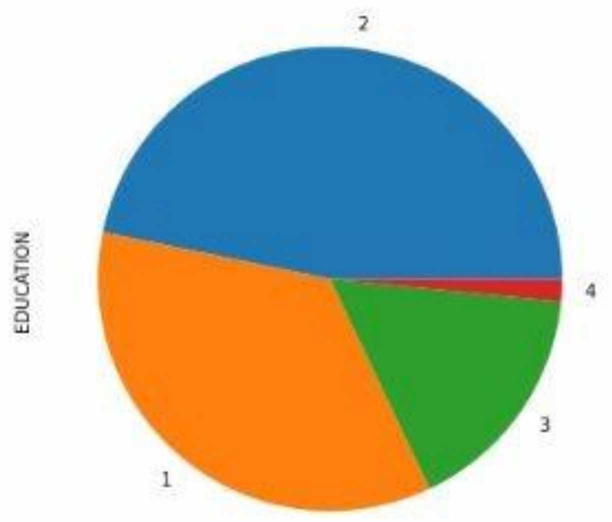


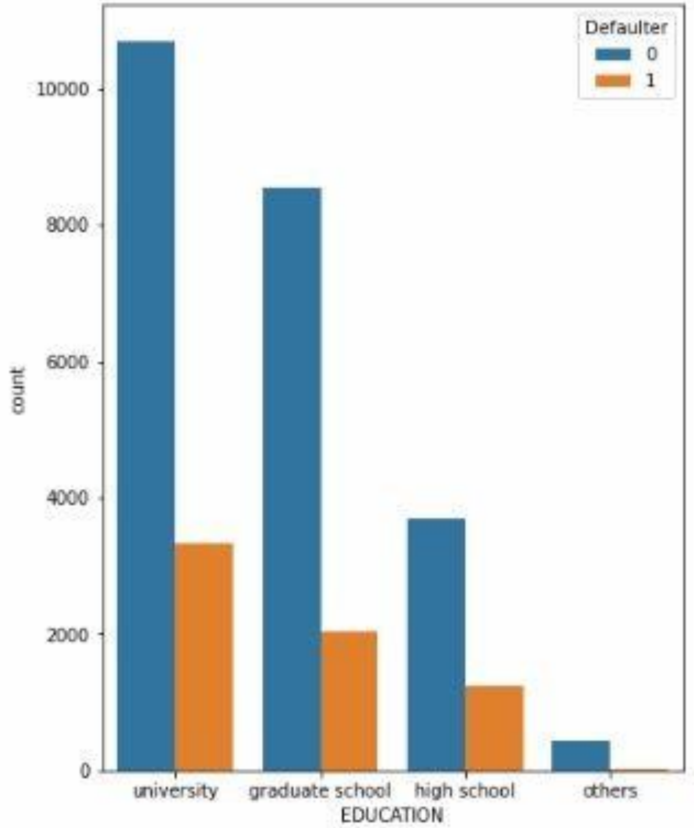
Gender wise defaulters



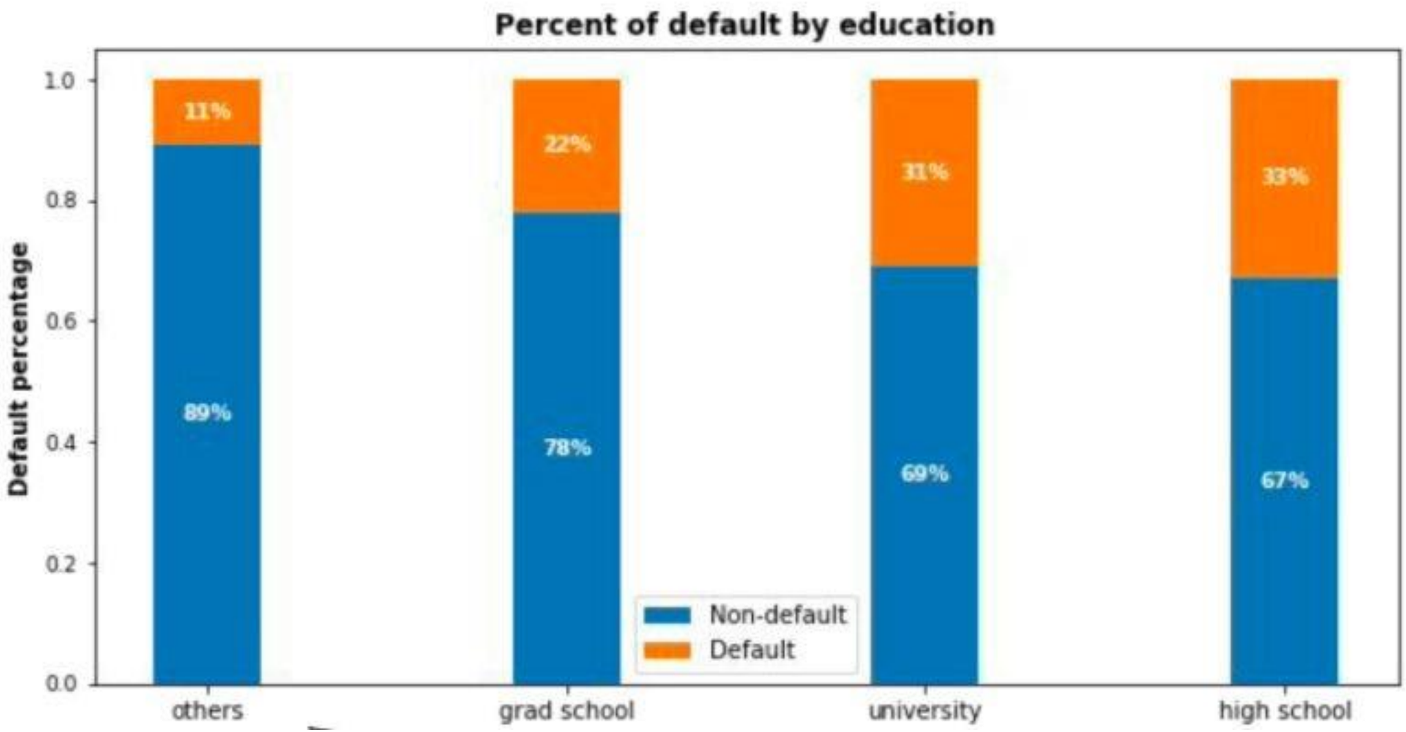
30% of Males and **26%** of Females are defaulters

Education Distribution



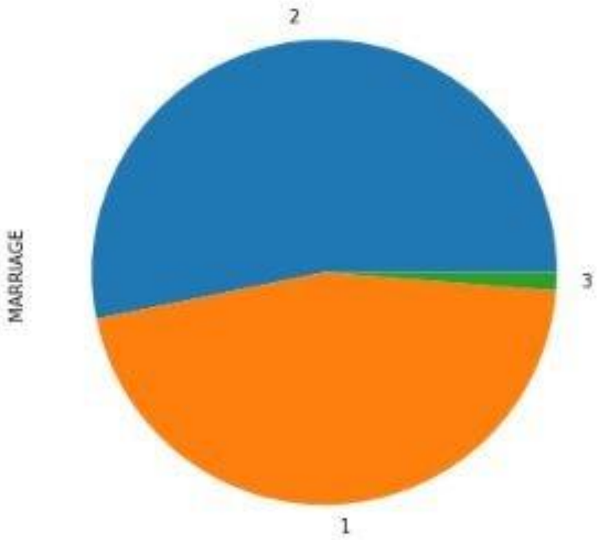


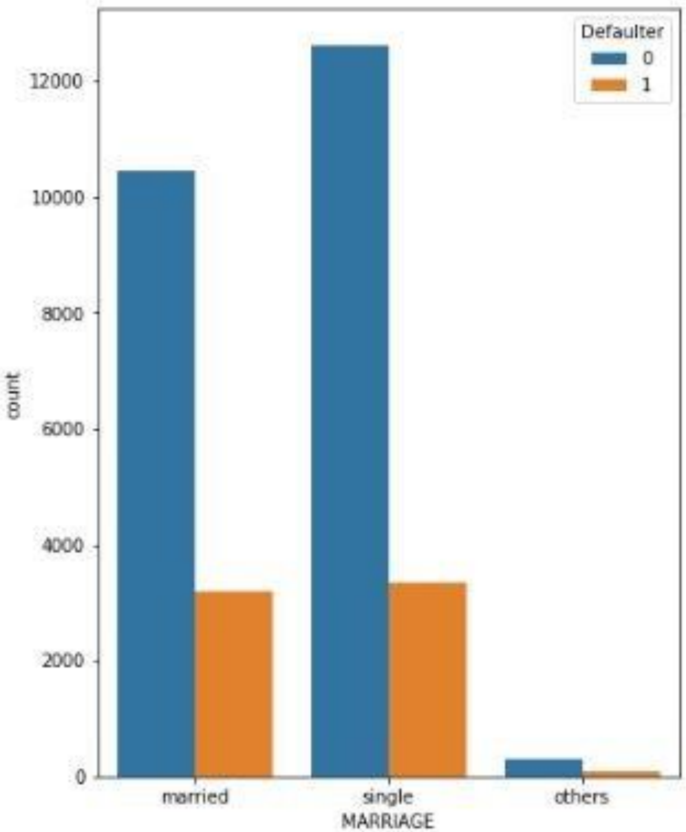
Education wise defaulters



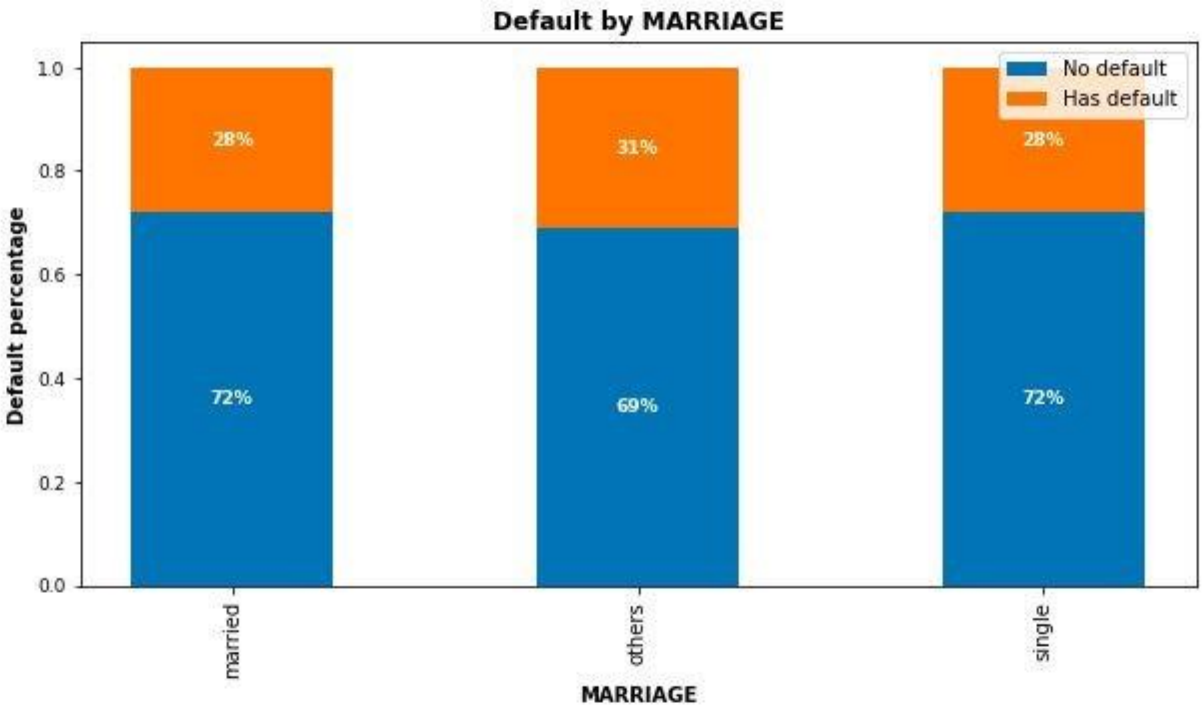
Higher Education level, lower Default Risk

Marital Distributions



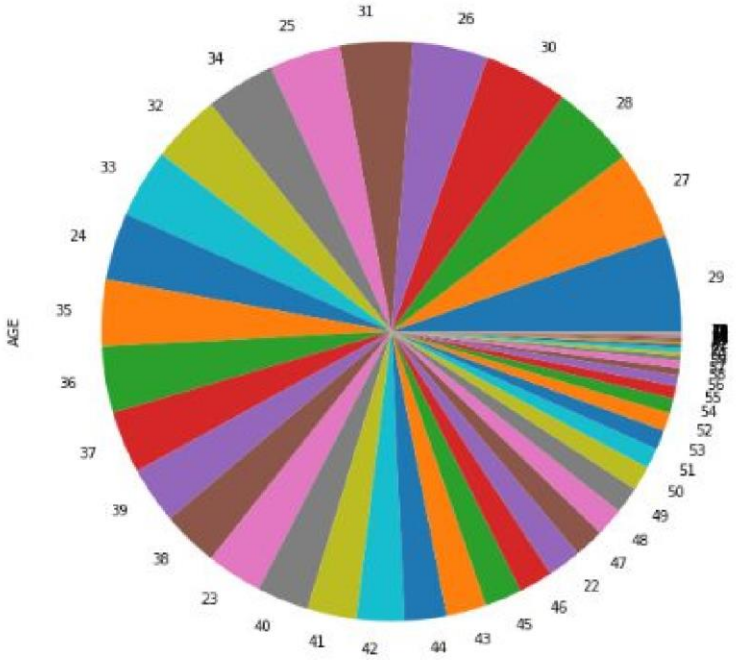


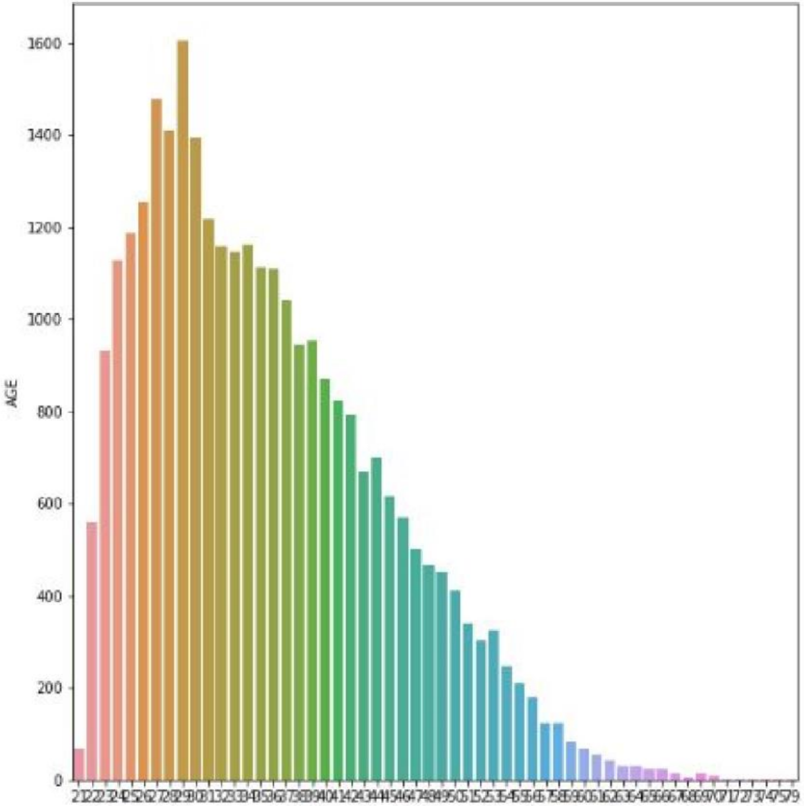
Marital Status



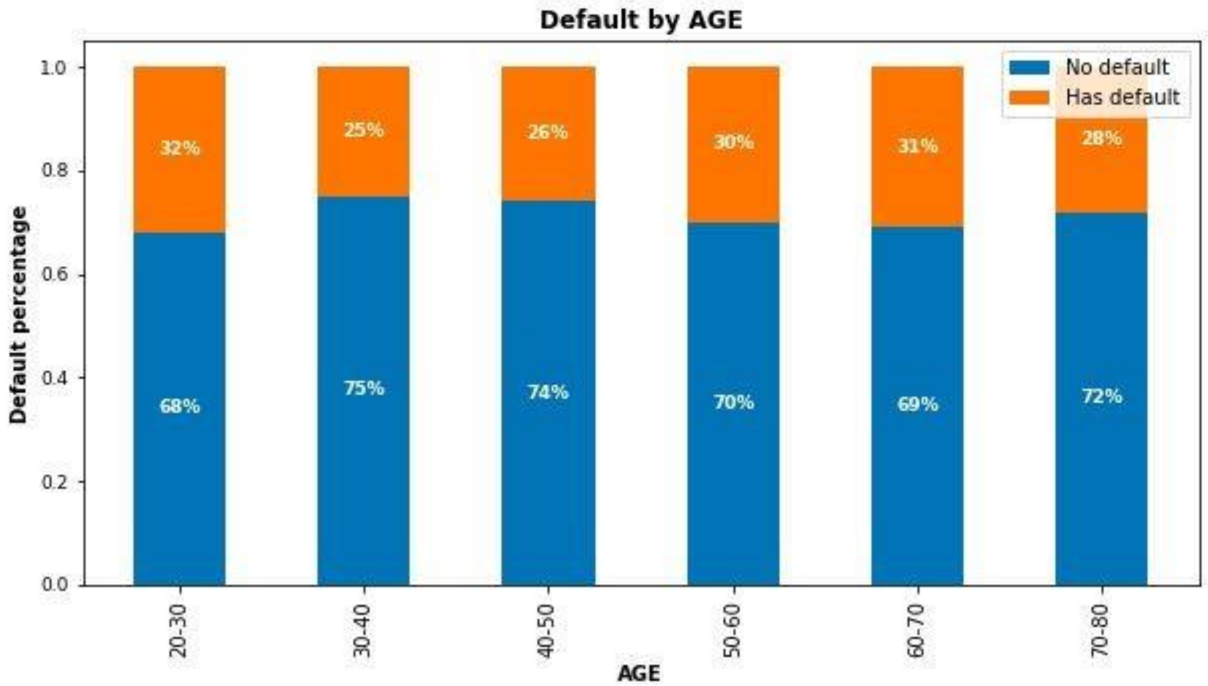
No Significant correlation of default risk and marital status

Age Distribution





Age wise defaulters



30 t0 50:

Lowest Risk

<30 and >50:

Risk Increases

Modeling Overview

- Supervised learning/Binary Classification
- Imbalance data with 78% non-defaulters and 22% defaulters **Models Used:**

- Logistic Regression
- Decision Trees
- Random Forest
- XGBoost

Modeling Steps

Data Preprocessing

- Feature selection
- Feature engineering
- Train test data split(80%-20%)
- SMOTE oversampling

Data Fitting and Tuning

- Start with default model parameters
- Hyperparameter tuning
- Measure RUC-AOC on training data

Model Evaluation

- Model testing
- Precision_Recall Score
- Compare with the other models

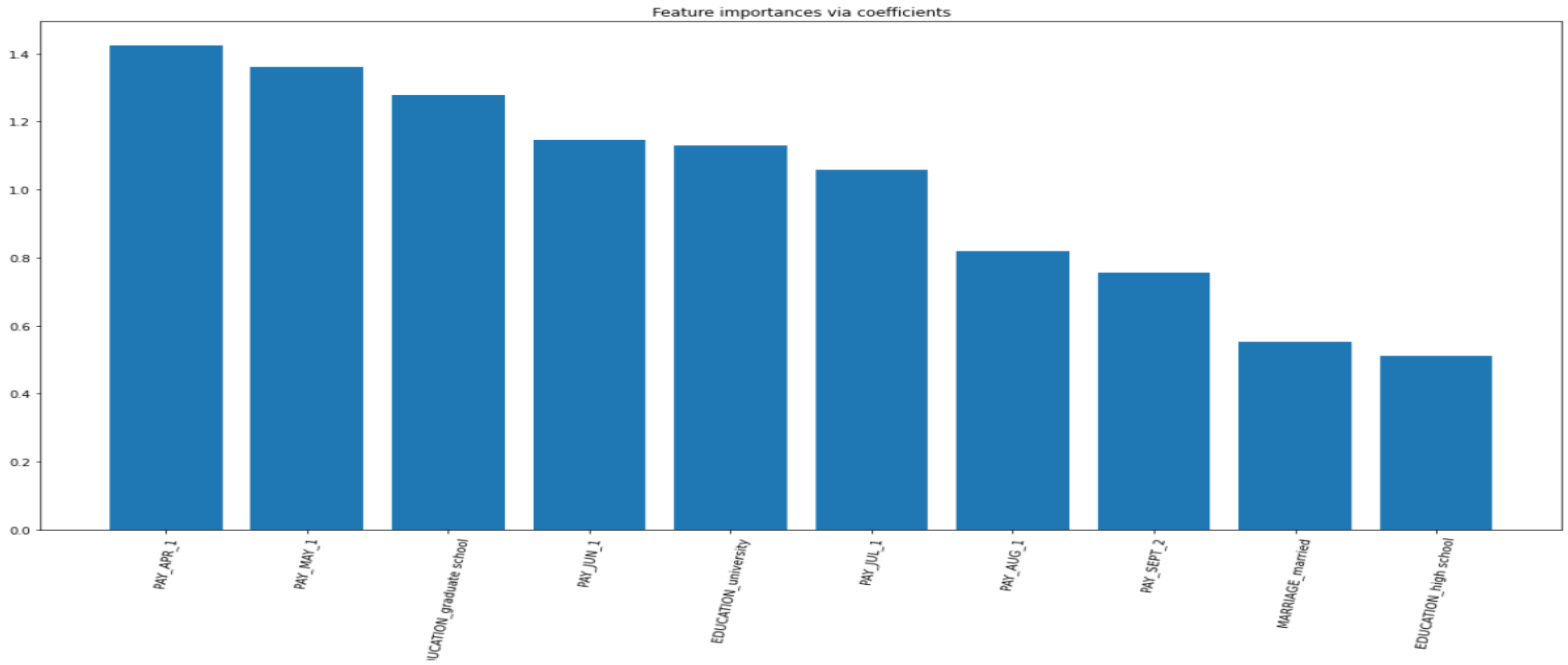
Logistic Modelling

Parameters :

```
The accuracy on test data is  0.7494325919201089
The precision on test data is  0.6797665369649806
The recall on test data is  0.7897830018083183
The f1 on test data is  0.7306566290255124
The roc_score on test data is  0.7543678810976708
```

- C = 100
- Penalty = L2

Logistic Feature Importances



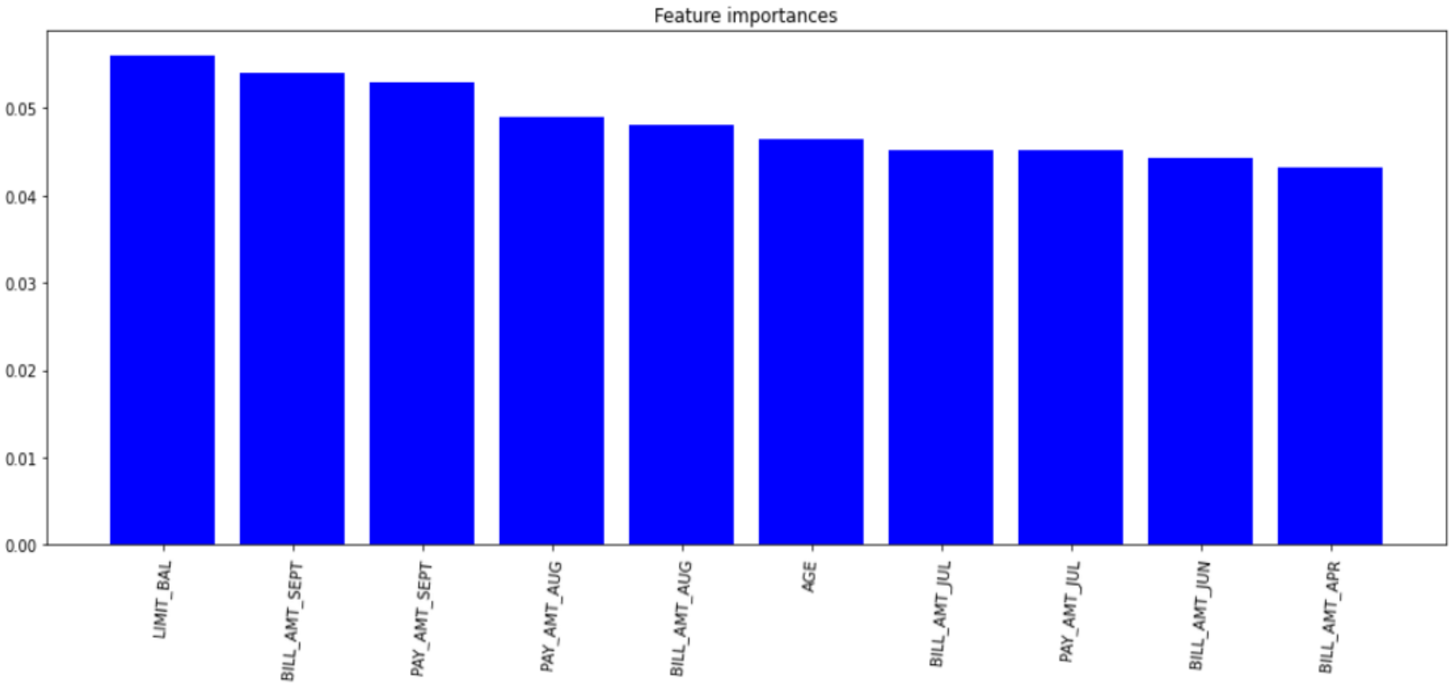
Random Forest Metrics

Parameters :

```
The accuracy on test data is  0.8336683742947928
The precision on test data is  0.7990920881971466
The recall on test data is  0.8584366727044727
The f1 on test data is  0.8277020218983006
The roc_score on test data is  0.8352712233003198
```

- max_depth=30
- n_estimators=150

Random Forest feature importances



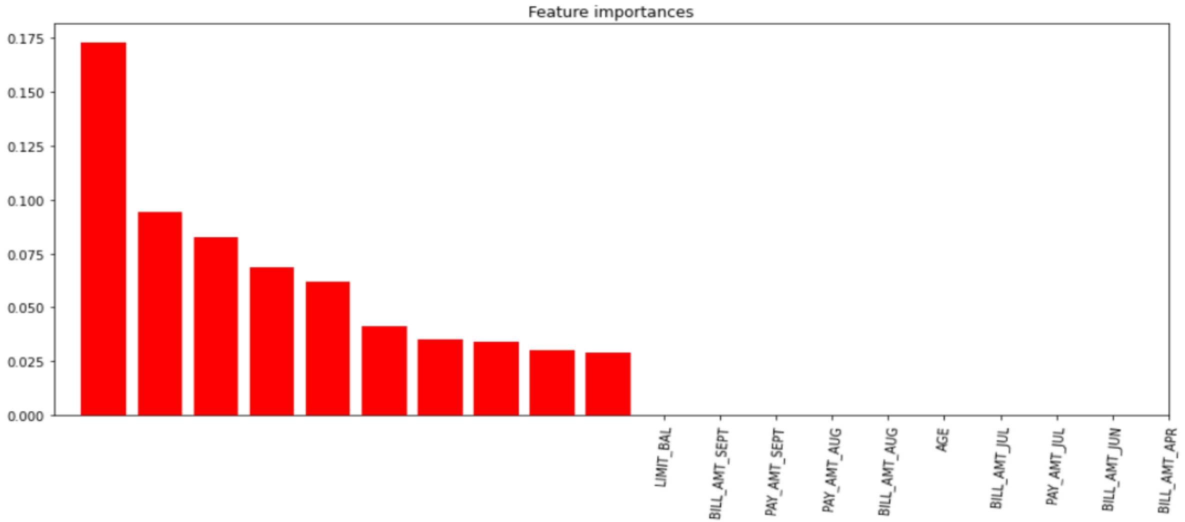
XGBoost Modelling

```
The accuracy on test data is  0.8253680046689579
The precision on test data is  0.7861219195849546
The recall on test data is  0.8530612244897959
The f1 on test data is  0.8182247721903477
The roc_score on train data is  0.8273843880986739
```

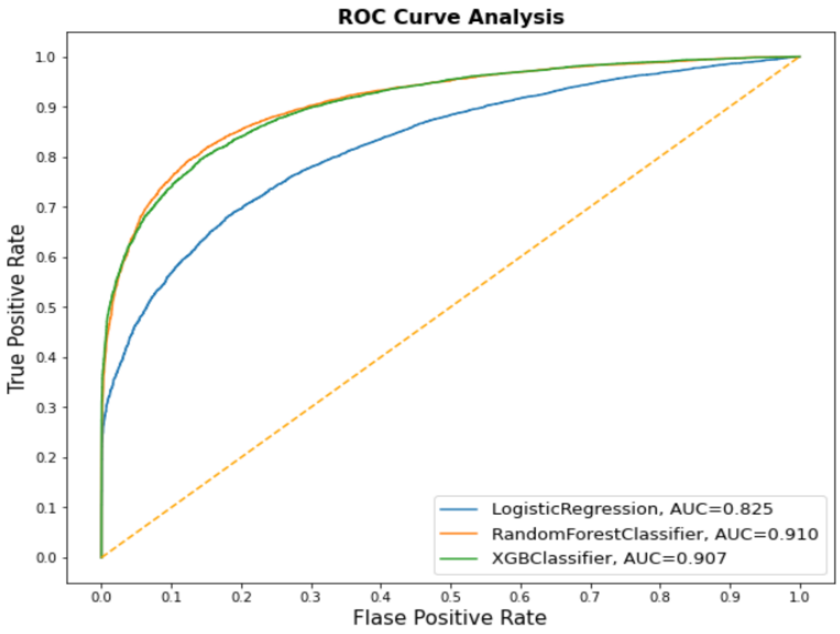
Parameters :

- max_depth= 9
- min_child_weight= 5

X Gradient Boosting feature importances



AUC-ROC curve comparision



Challenges

- Understanding the columns.
- Feature engineering.
- Getting a higher accuracy on the models.

Conclusion

- XGBoost provided us the best results giving us a recall of 85 percent(meaning out of 100 defaulters 85 will be correctly caught by XGBoost)
- Random Forest also had good score as well but leads to overfit the data.
- Logistic regression being the least accurate with a recall of 79.

t[ ]:

	Classifier	Train Accuracy	Test Accuracy	Precision Score	Recall Score	F1 Score
0	Logistic Regression	0.751461	0.749433	0.679767	0.789783	0.730657
1	Random Forest CLf	0.998850	0.832825	0.800389	0.855895	0.827212
2	Xgboost Clf	0.914077	0.825368	0.786122	0.853061	0.818225