# 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".

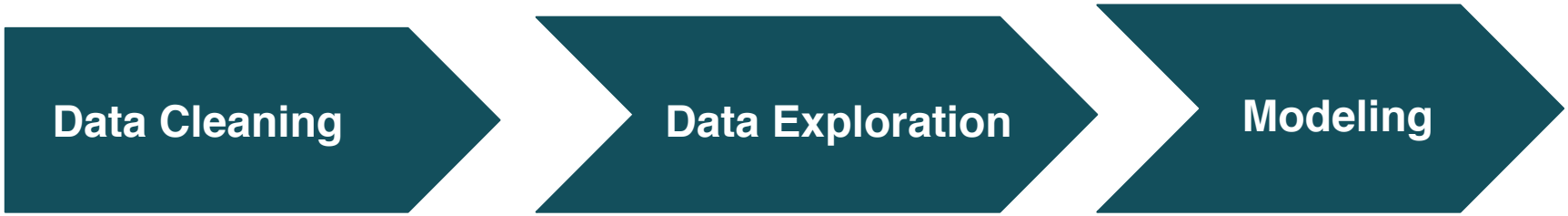
# Problem Statement

**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




**Understanding and Cleaning**

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

**Graphical**

- Examining the data with visualization

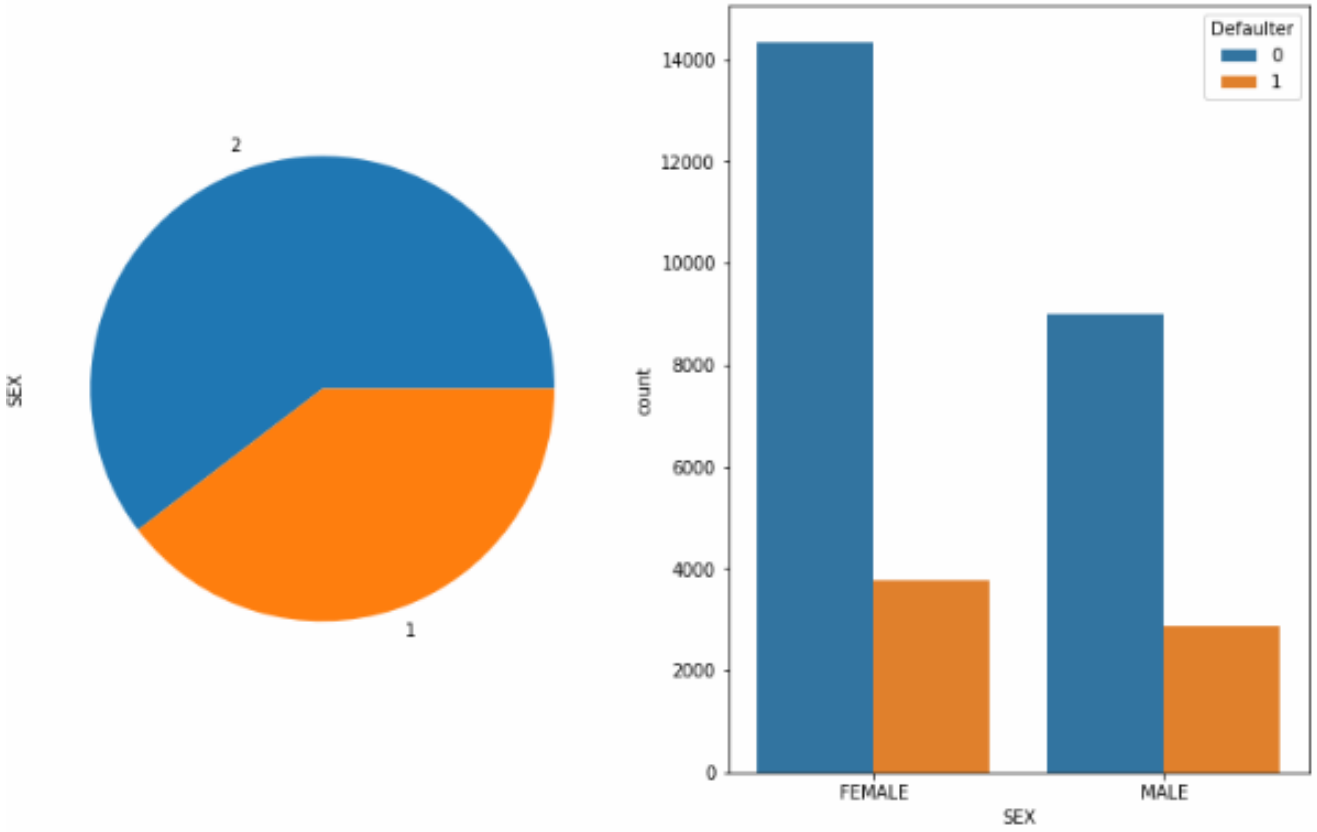
**Machine Learning**

- Logistic
- SVM
- Random Forest
- XGBoost

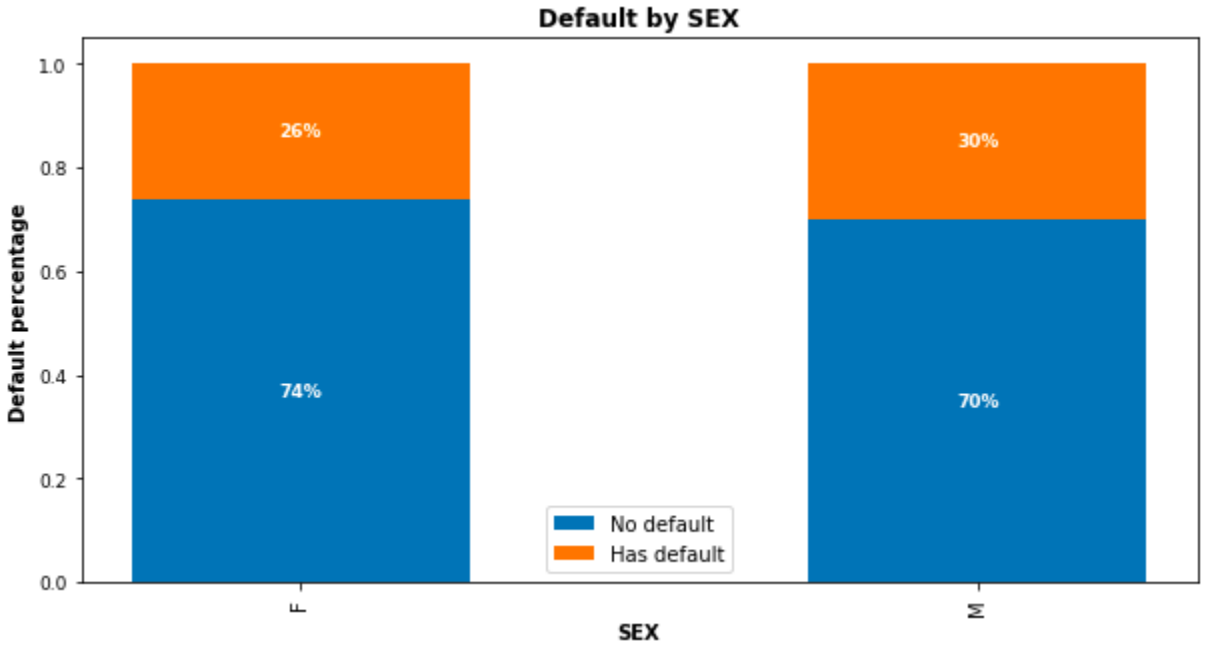
# Basic Exploration

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

# Gender Distribution

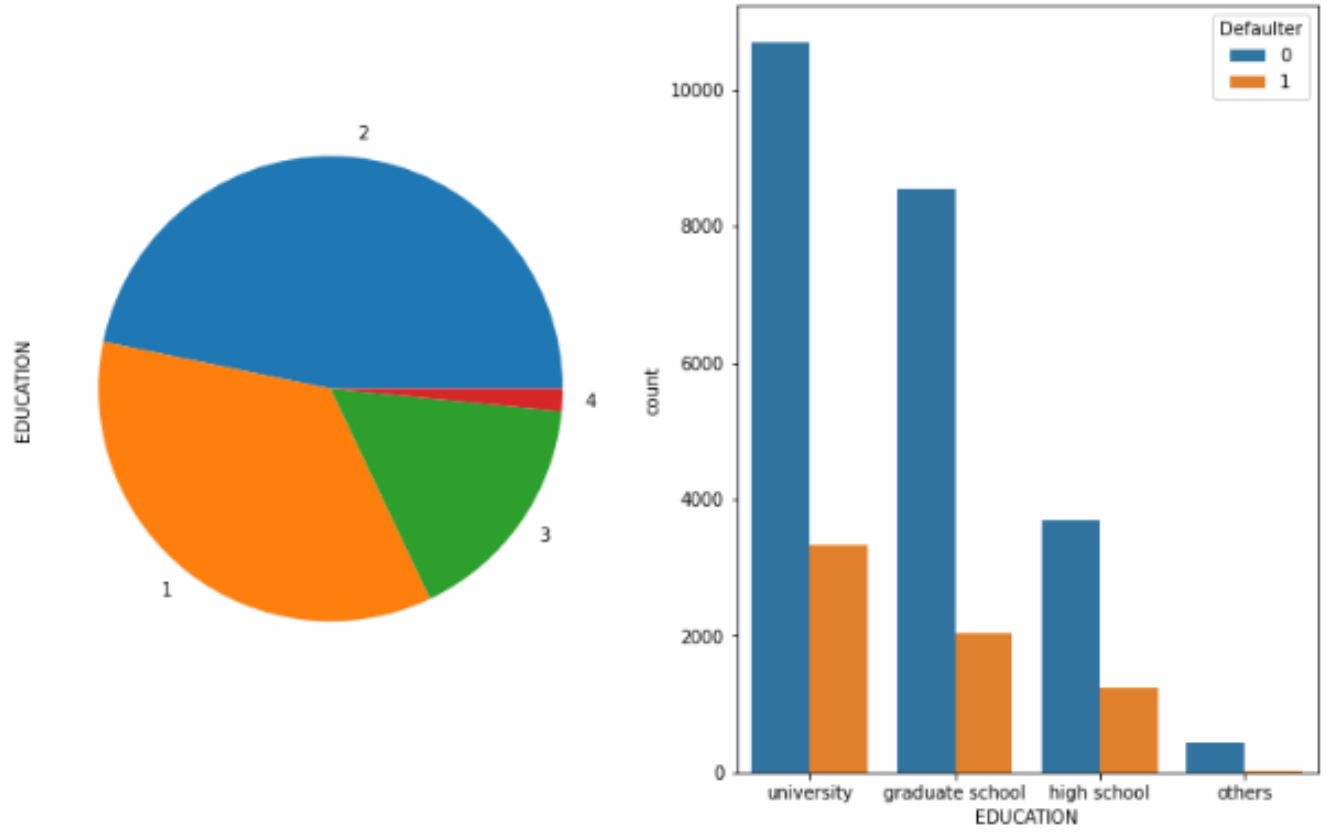


# Gender wise defaulters

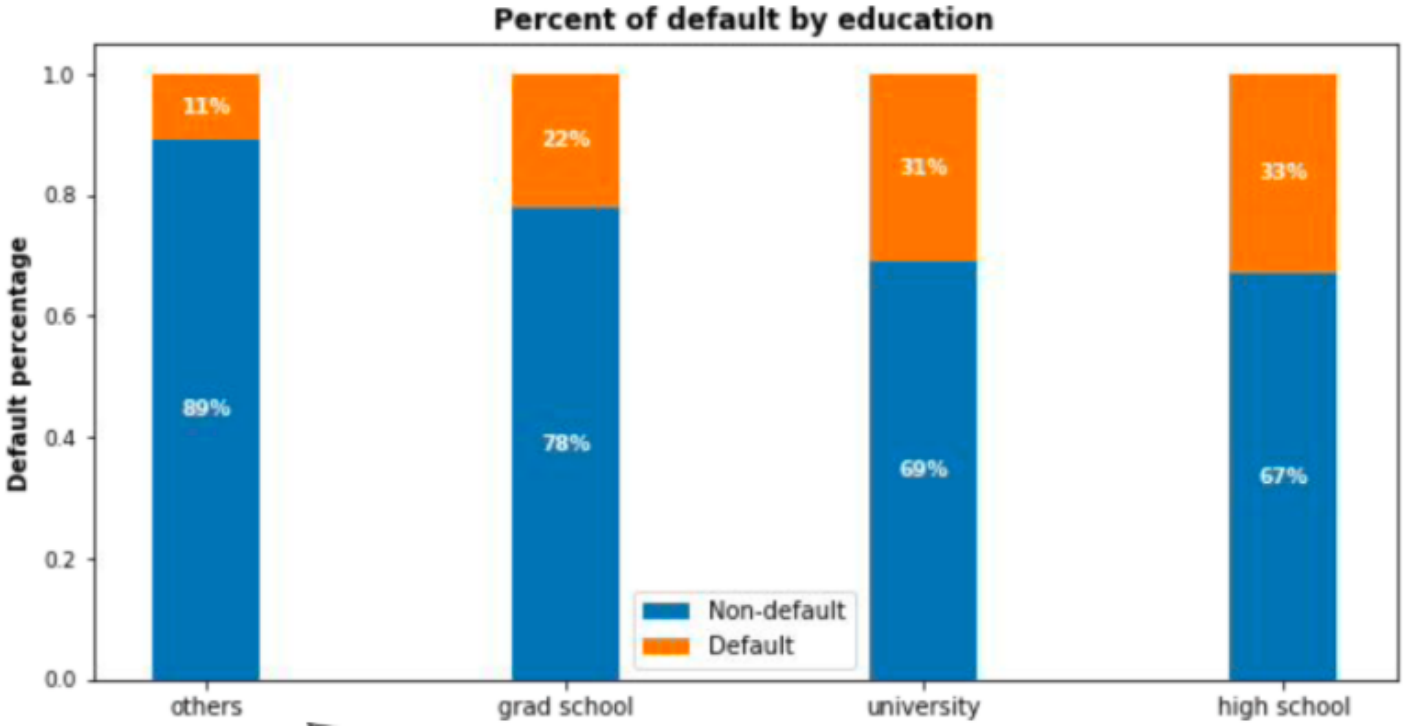



**30%** of Males and **26%** of Females are defaulters

# Education Distribution

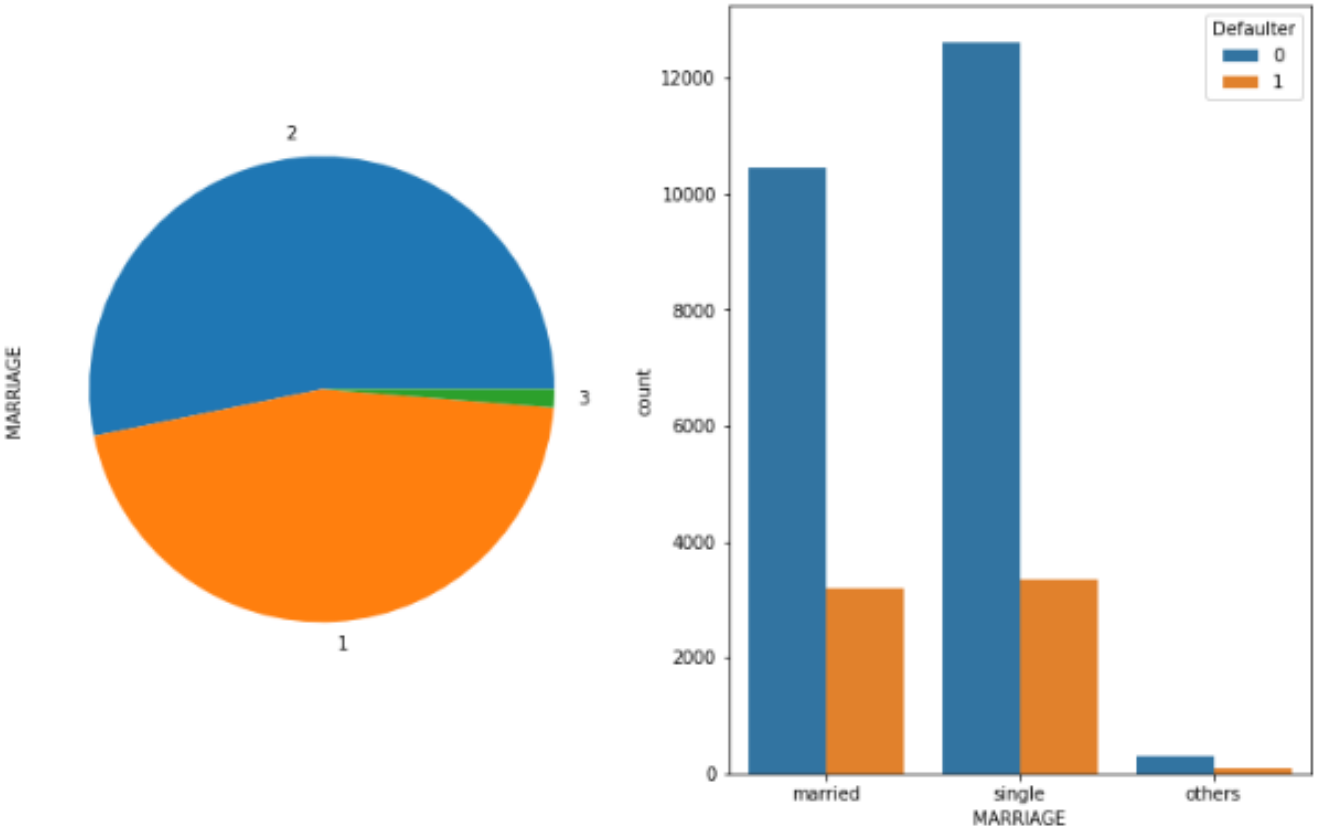


# Education wise defaulters

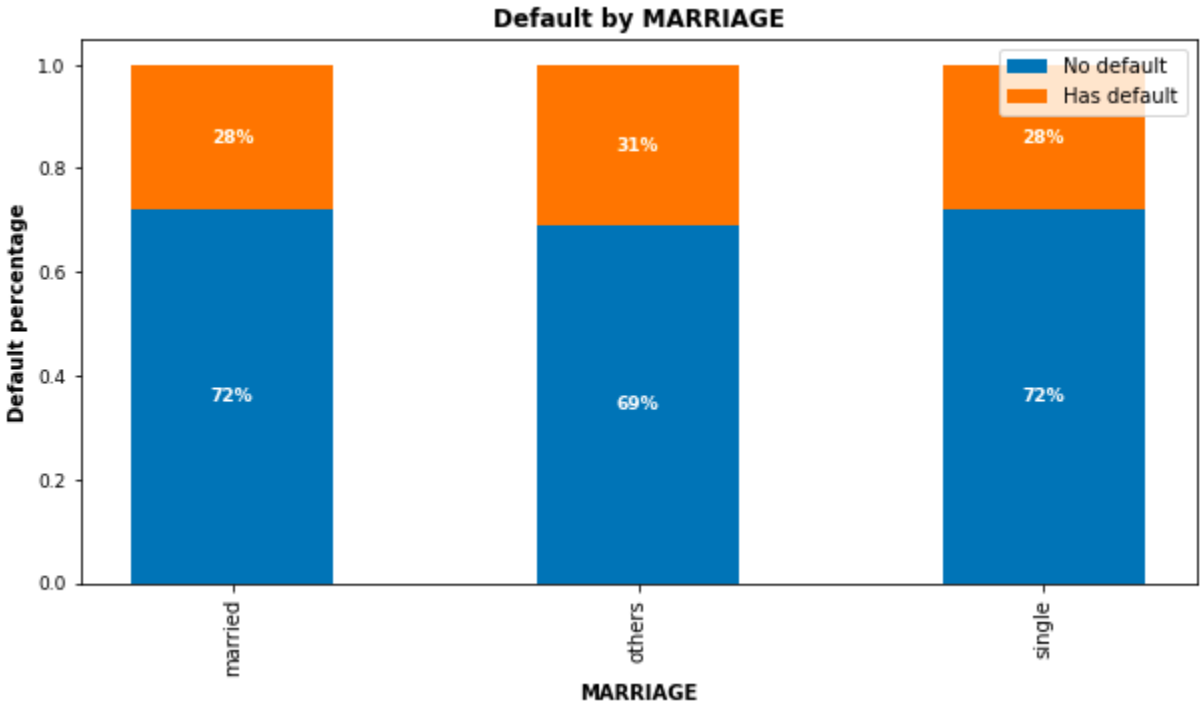



**Higher** Education level, lower Default Risk

# Marital Distributions

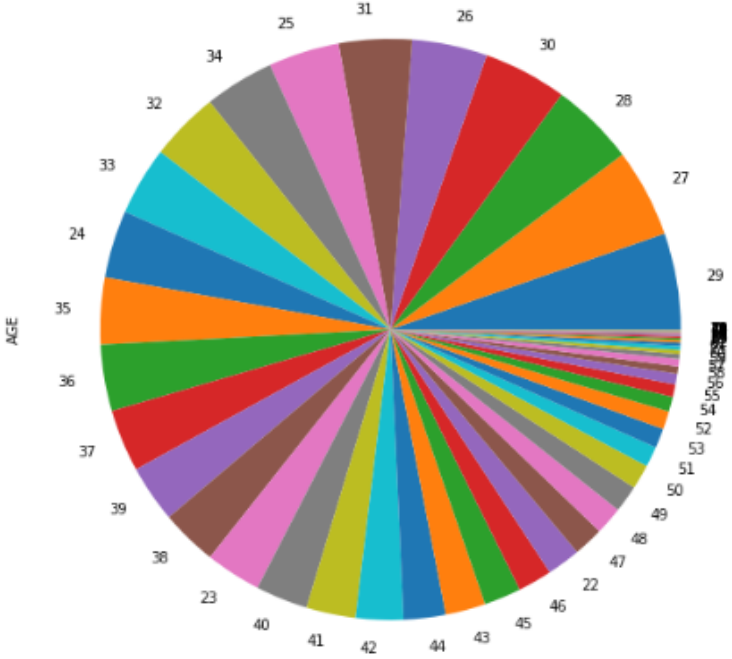


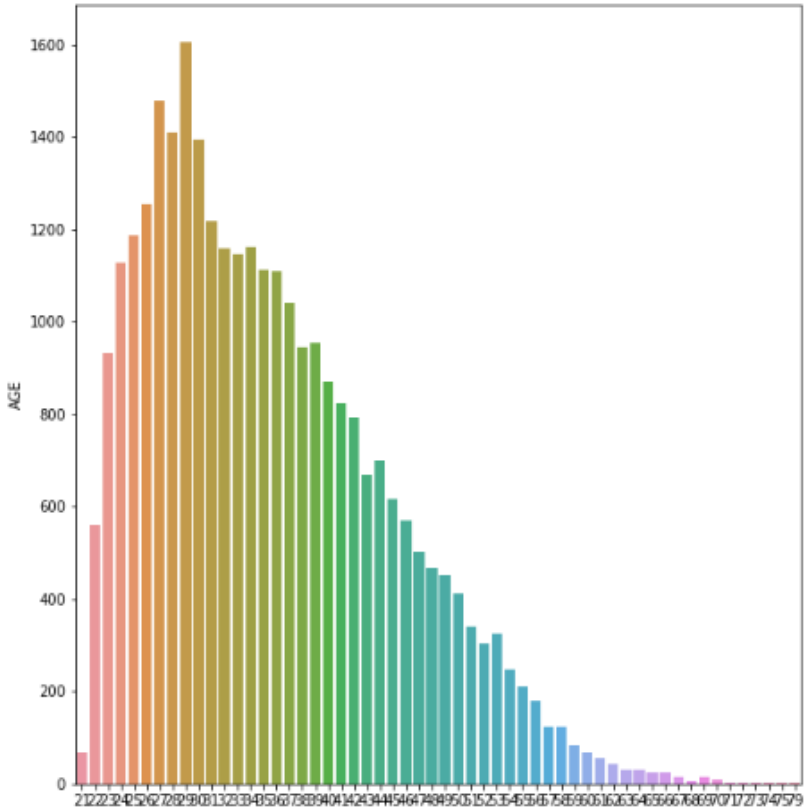
# Marital Status



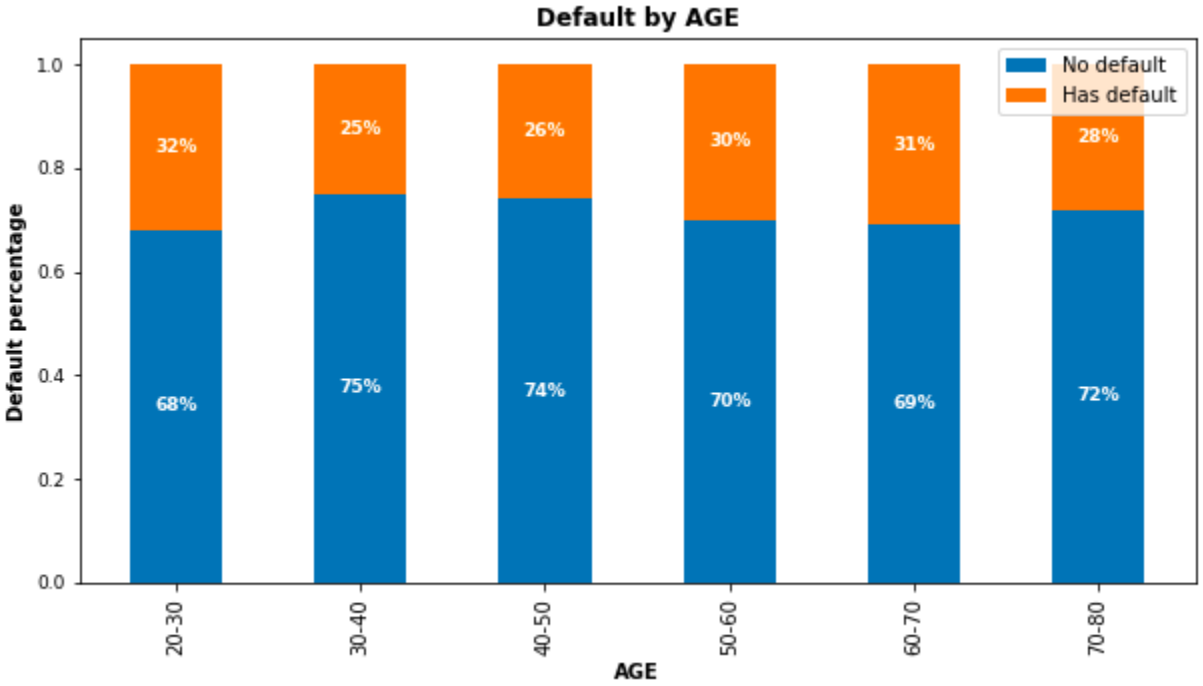

**No** Significant correlation of default risk and marital status

# 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
  - Knn
  - Decision Trees
  - Random Forest
  - SVM
  - XGBoost
  - Naive Bayes

# 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

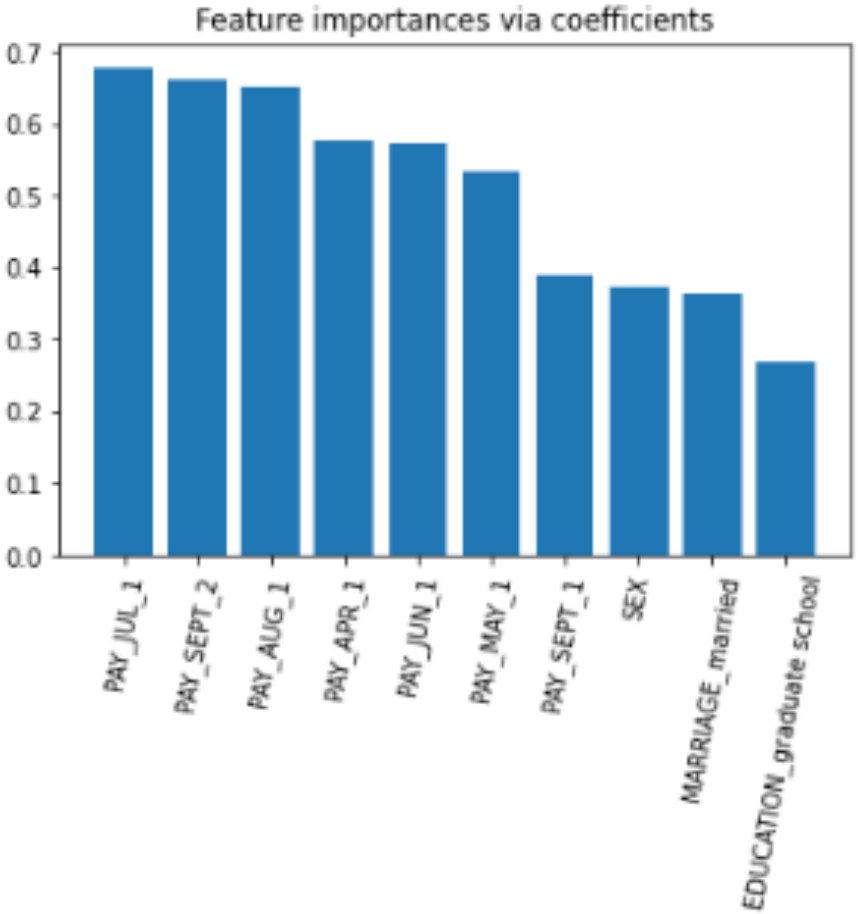
# Logistic Modelling

## Parameters :

- C = 0.01
- Penalty = L2

```
The accuracy on test data is  0.7563711821542053
The precision on test data is  0.6963683527885862
The recall on test data is  0.7913043478260869
The f1 on test data is  0.7408071748878924
The roc_score on train data is  0.7601148881325897
```

# Logistic feature importances



## SVM Modelling

# Parameters

**C = 10**

**Kernel = ‘rbf’**

```
The accuracy on test data is  0.7812074443940081
The precision on test data is  0.7229571984435798
The recall on test data is  0.8182618907809748
The f1 on test data is  0.7676628563558738
The roc_score on train data is  0.7850747251558493
```

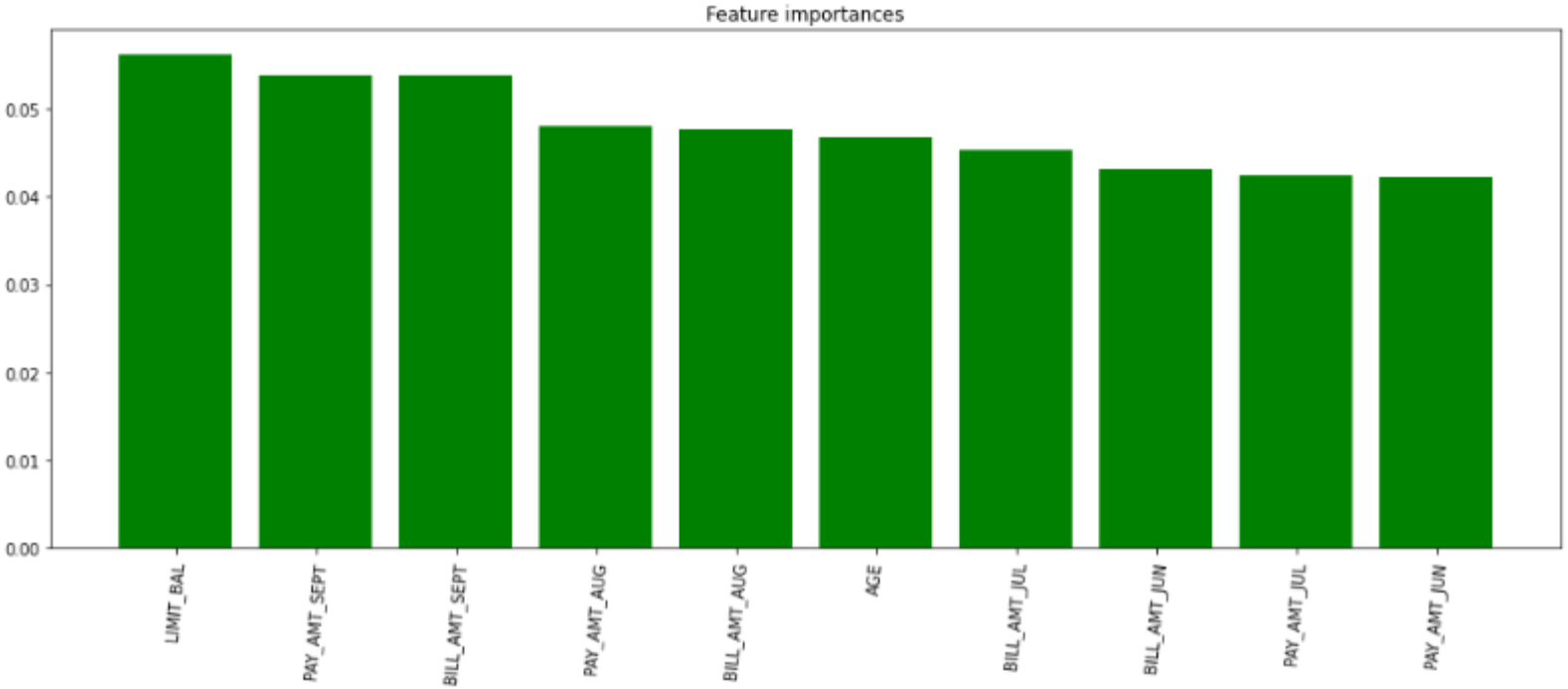
# Random Forest Metrics

## Parameters :

- max_depth=30
- n_estimators=150

```
The accuracy on test data is  0.8357434667012515
The precision on test data is  0.8051880674448768
The recall on test data is  0.8575770133996409
The f1 on test data is  0.8305572279082214
The roc_score on test data is  0.8370016575186912
```

# Random Forest feature importances



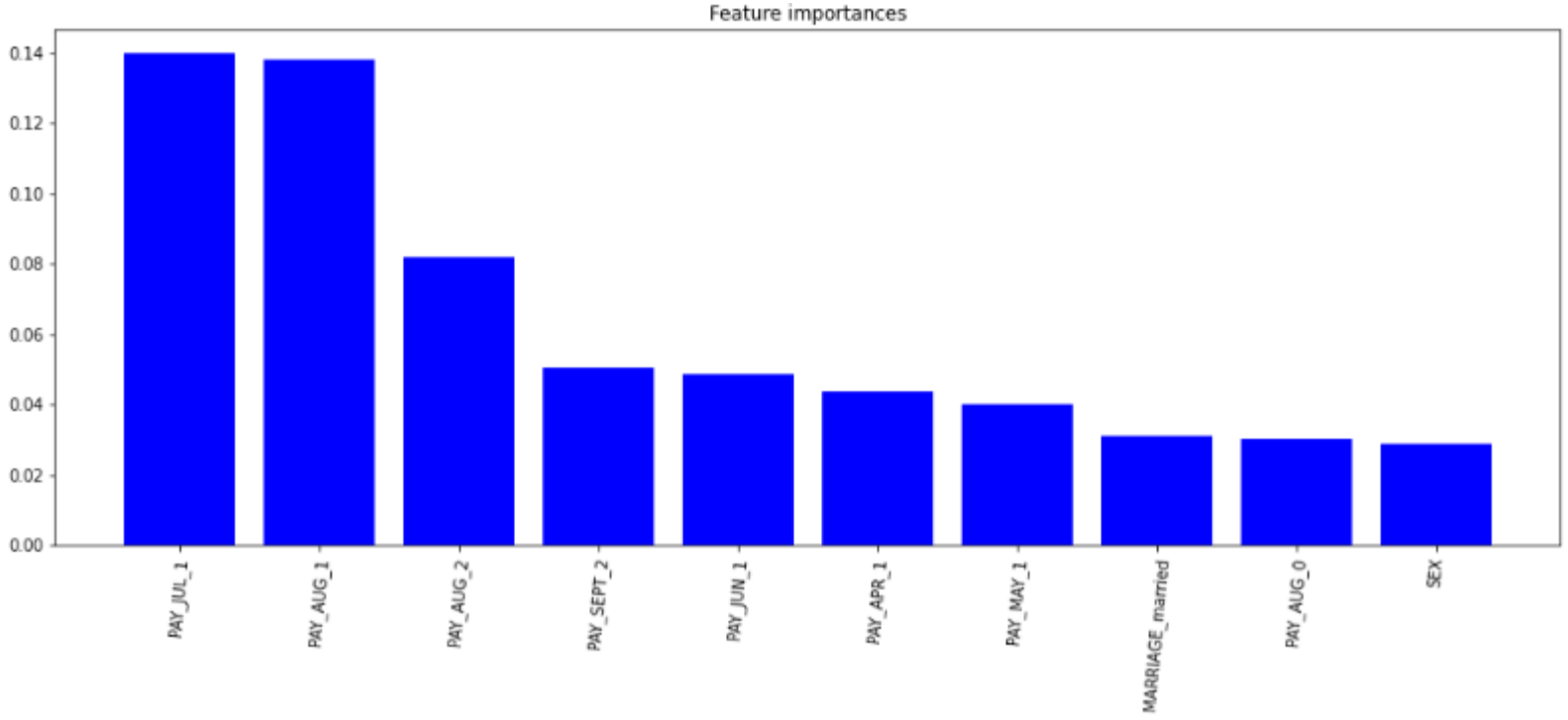
## XGBoost Modelling

### Parameters :

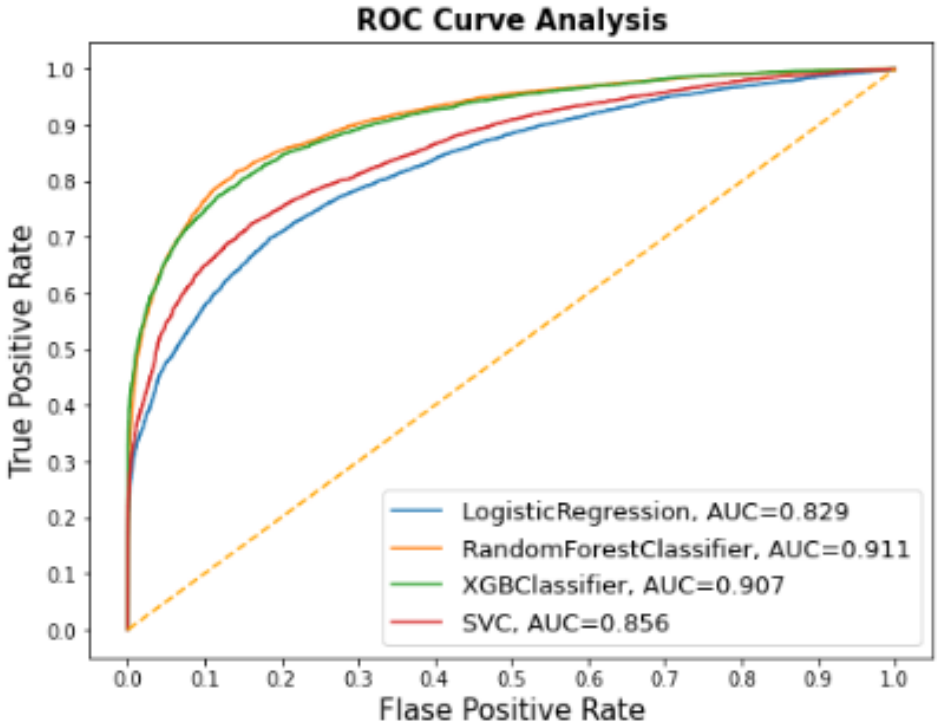
- max_depth= 15
- min_child_weight= 8

```
The accuracy on test data is  0.8271188638869075
The precision on test data is  0.7859922178988327
The recall on test data is  0.856416054267948
The f1 on test data is  0.8196943054240496
The roc_score on train data is  0.8293464333652503
```

# 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.

| Classifier | Train Accuracy | Test Accuracy | Precision Score | Recall Score | F1 Score |
|---|---|---|---|---|---|
| Logistic Regression | 0.754017 | 0.756371 | 0.696368 | 0.791304 | 0.740807 |
| SVC | 0.809851 | 0.781207 | 0.722957 | 0.818262 | 0.767663 |
| Random Forest CLf | 0.998754 | 0.835743 | 0.805188 | 0.857577 | 0.830557 |
| Xgboost Clf | 0.912607 | 0.827119 | 0.785992 | 0.856416 | 0.819694 |

# Thank You