# Capstone Project - 3

## Credit Card Default Prediction

By: Akmal Jah Quamri

# Scope of Work

## Problem Description

- Aim of the project is to build a predictive model that will predict the credit default risk

## Tool Used

- Pandas
- NumPy
- Matplotlib
- Seaborn
- Scikit-Learn

## Problem Statements

- Supervised Learning / Binary Classification
- Imbalance target variables
- Build Machine Learning model to predict
- Identifying the model performance
- Finding the models most explained features

# Approach

**Data Exploration**

- Exploring dataset
- Finding trends
- Processing dataset
- Visual Representation of key findings

**Model Building**

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

**Model Evaluation**

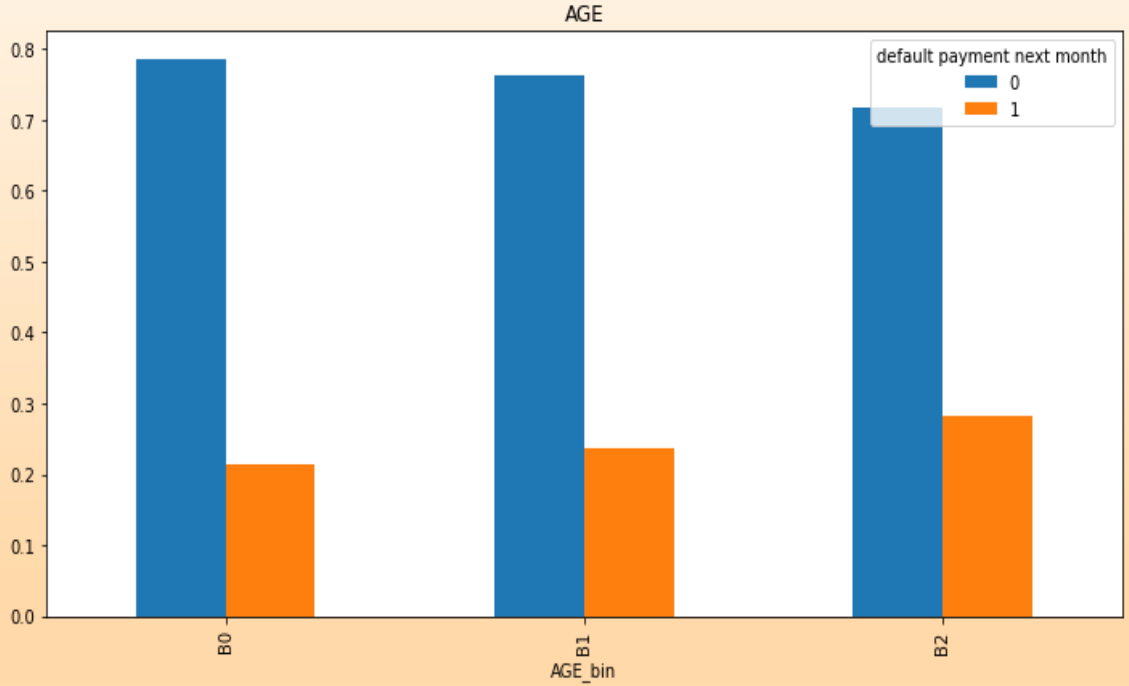
- Comparing Models
- Classification Report
- ROC_AUC Curve
- Confusion Metrics

# Part 1

Exploratory Data Analysis

- Finding the factor that impacting on default credit risk

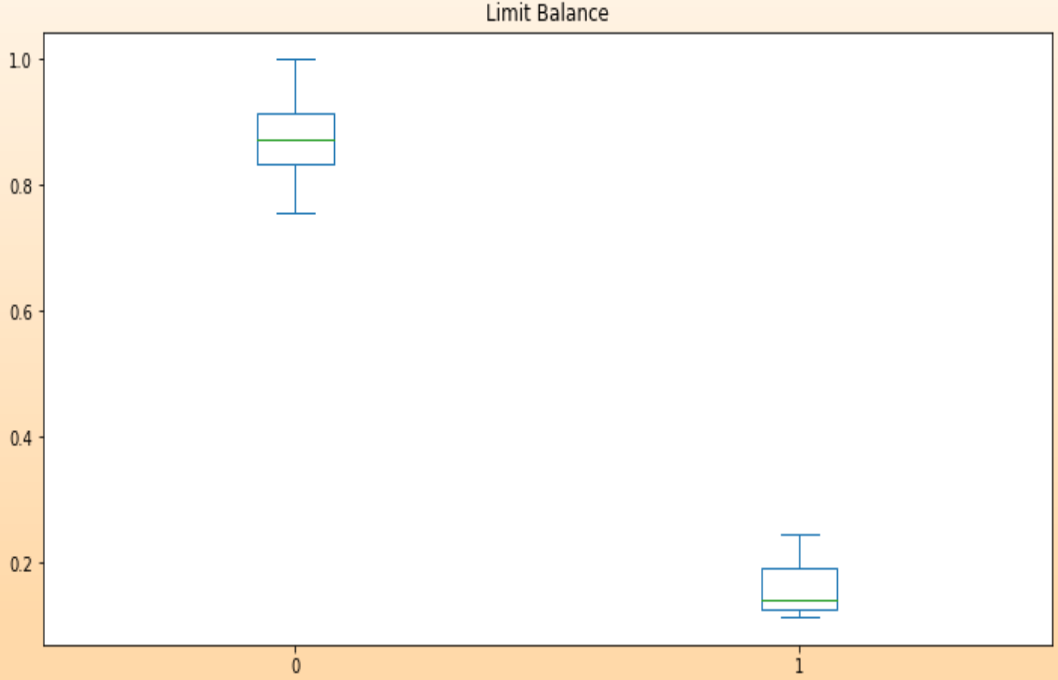
# Age Group– Default Risk




Age bin: 21, 40, 60, 79

- Older people will tend to fall more in default category

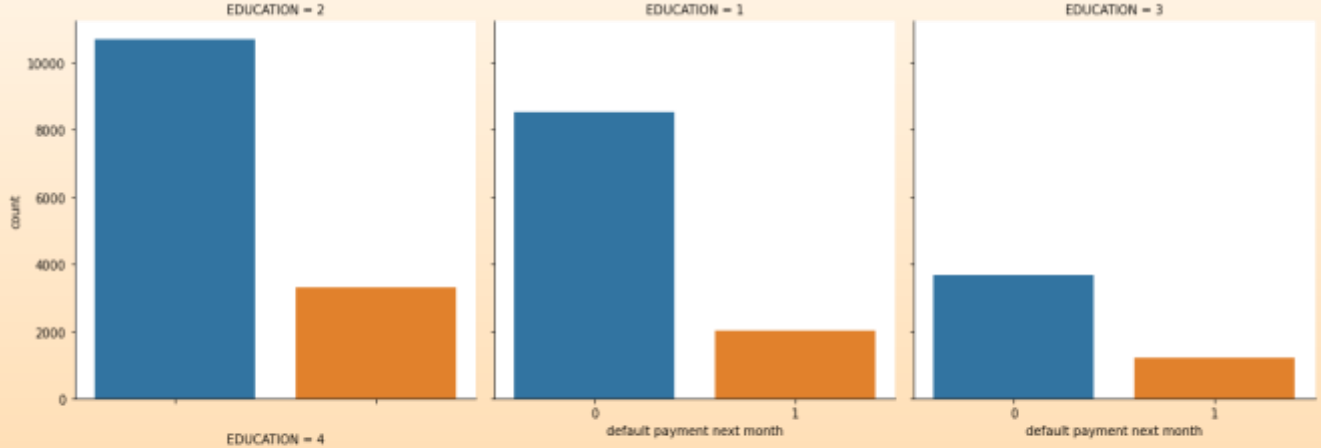
# Limit Balance – Default Risk

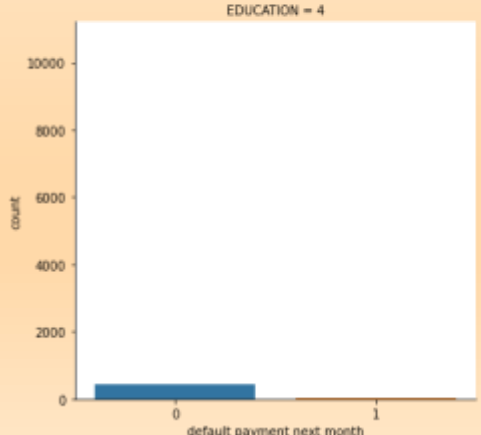



- The lower the limit the higher the chances of being default
- With highest limit no default is recorded

```
Limit balance bin: 10000, 257500, 505000, 752500, 1000000
```

# Education– Default Risk

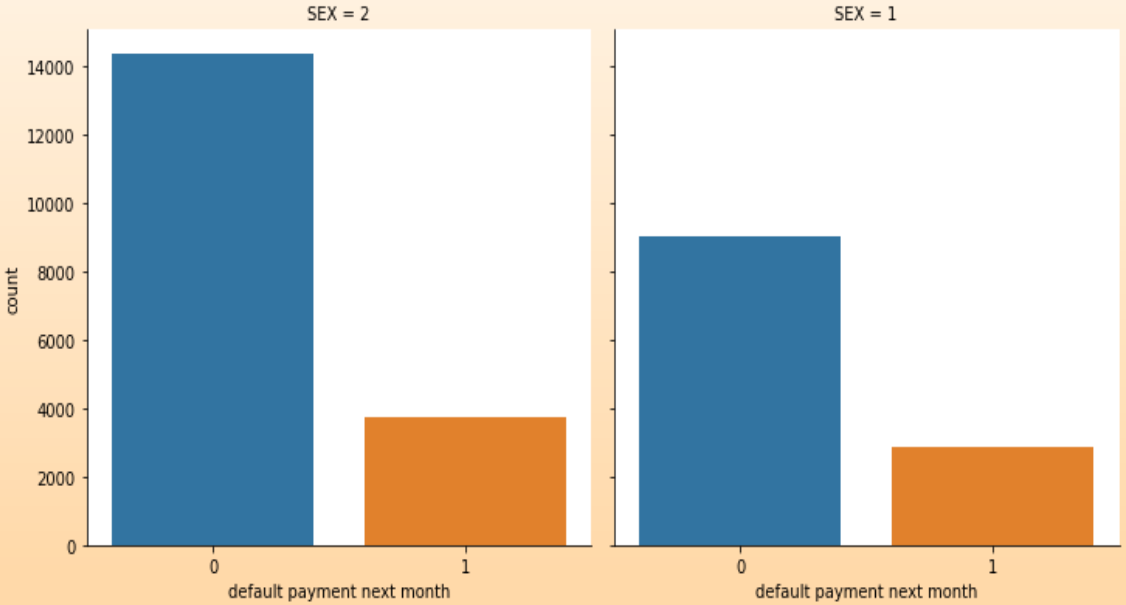



- Education1 – Graduate school
- Education 2 – University
- Education 3 – High school
- Education 4 – others

If we see the ration of being default among all these education then, we can realize that the higher the education the lower the rate of being default.

# Gender – Default Risk




- Sex 1 – Male
- Sex 2 – Female

From above graph we can say that male has higher chances of being default compared to female.

# Part 2

Building Predictive Model

- Finding the best performing model with respect to precision and recall

# Modelling Steps

**Feature Engineering**

- Splitting dataset into dependent and independent variables
- Train Test split
- Standardization Scaling
- Handling Imbalance

**Model Training**

- Building Model
- Finding best hyperparameter using grid search, random search
- Training Model and predicting on test data

**Model Evaluation**

- Checking accuracy
- Classification Report of precision, recall
- Roc Auc Curve
- Confusion Metrics

# Data Pre-processing

| | |
|---|---|
| **Independent and Target variables** | Defined a variable X and stored all independent features and stored target feature in variable y. |
| **Train Test Split** | Split the dataset into two part for training with 80 percent rows and for testing with 20 percent datapoints. |
| **Scaling** | Used scikit learn standard scaler to scale our dataset. Scaling become more important when we have different units of numerical data. |
| **Imbalance** | Our target variable is binary class and there is a huge difference between majority and minority class. To balance the class I used oversampling technique to create synthetic data points for minority class. |

# Modelling with Performance

## Logistic Regression

| | |
|---|---|
| Accuracy | 0.74 |
| Precision | 0.75 |
| Recall | 0.72 |

## Random Forest

| | |
|---|---|
| Accuracy | 0.79 |
| Precision | 0.81 |
| Recall | 0.74 |

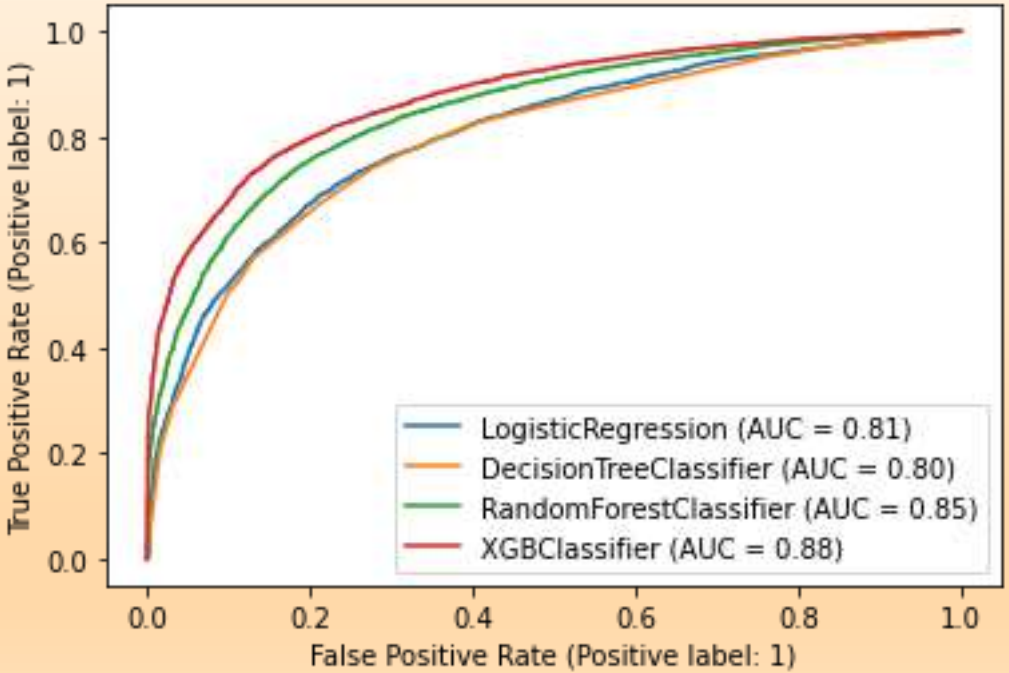
## Decision Tree

| | |
|---|---|
| Accuracy | 0.73 |
| Precision | 0.76 |
| Recall | 0.68 |

## XGBoost

| | |
|---|---|
| Accuracy | 0.80 |
| Precision | 0.83 |
| Recall | 0.77 |

# Model Evaluation – AUC ROC Curve

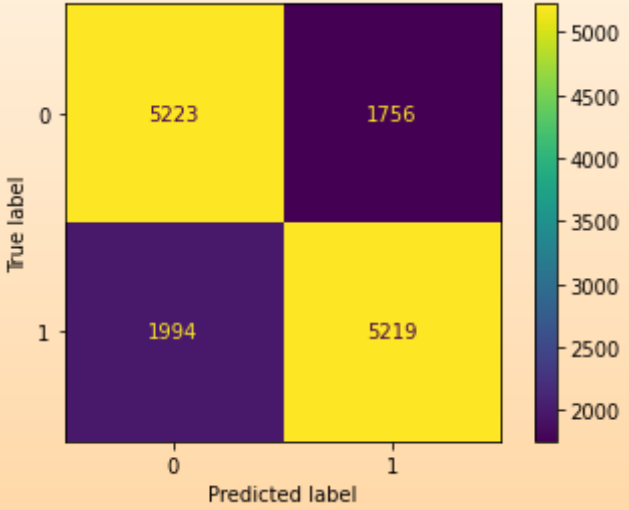



With the help of this roc_auc plot it is clear that XGBoost classifier is scoring best with 0.88 points.

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes

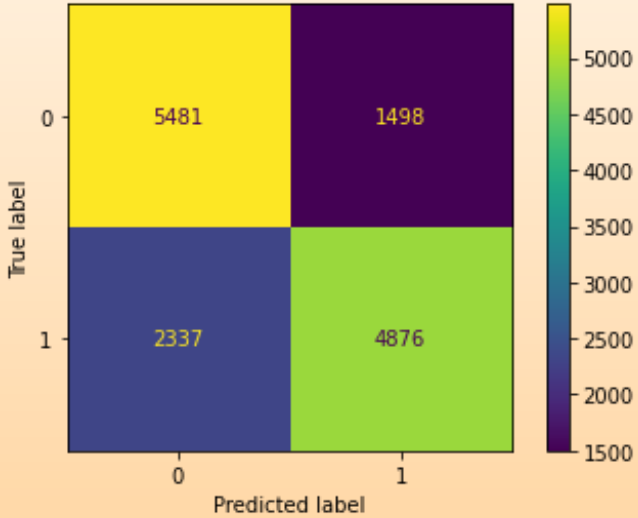
# Model Evaluation – Confusion Metrix

## Logistic Regression




| | |
|---|---|
| True Negative | 5223 |
| False Positive | 1756 |
| False Negative | 1994 |
| True Positive | 5219 |

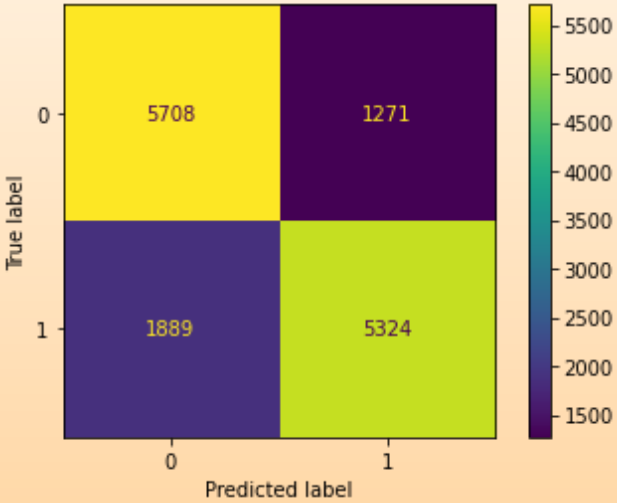
## Decision Tree Classifier




| | |
|---|---|
| True Negative | 5481 |
| False Positive | 1498 |
| False Negative | 2337 |
| True Positive | 4876 |

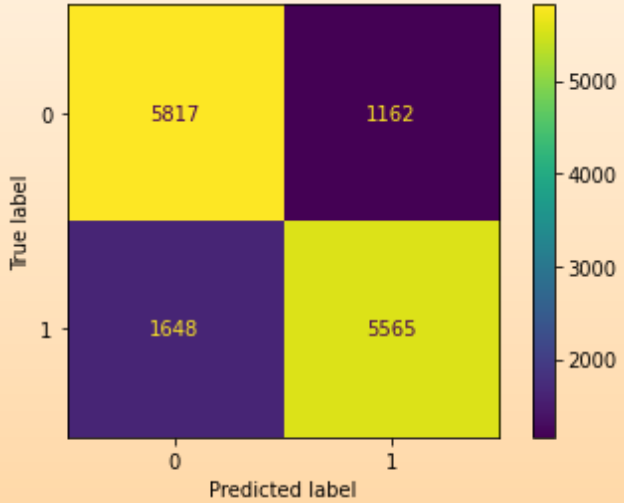
# Model Evaluation – Confusion Metrix (contd.)

## Random Forest Classifier




| | |
|---|---|
| True Negative | 5708 |
| False Positive | 1271 |
| False Negative | 1889 |
| True Positive | 5324 |

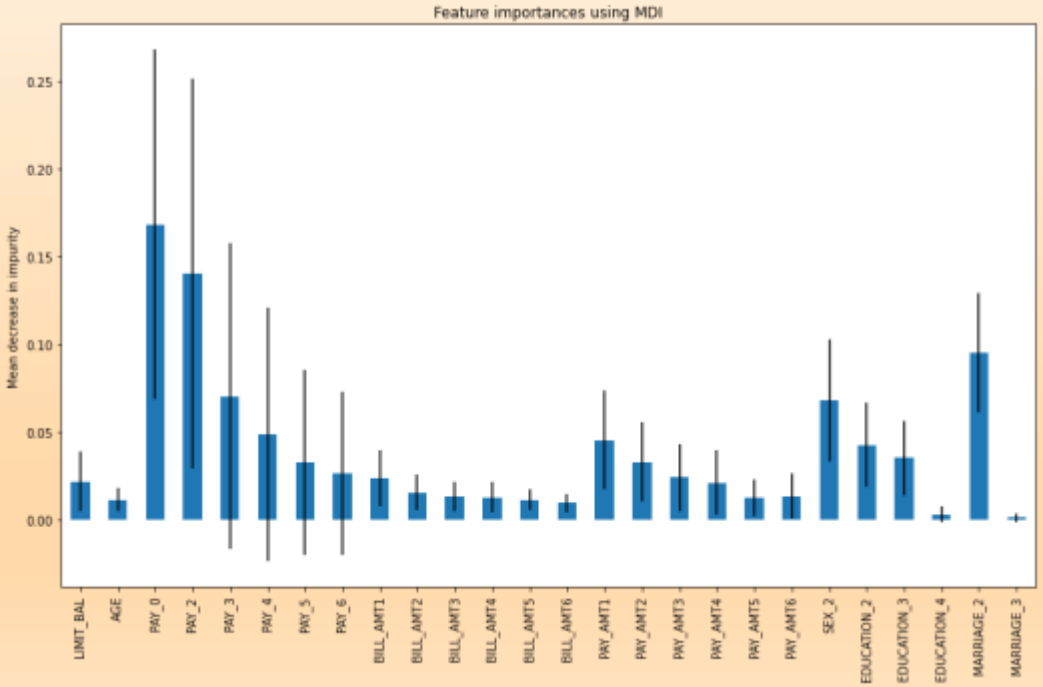
## XGBoost Classifier




| | |
|---|---|
| True Negative | 5817 |
| False Positive | 1162 |
| False Negative | 1648 |
| True Positive | 5565 |

# Feature Importance

## Random Forest Classifier Feature Importance

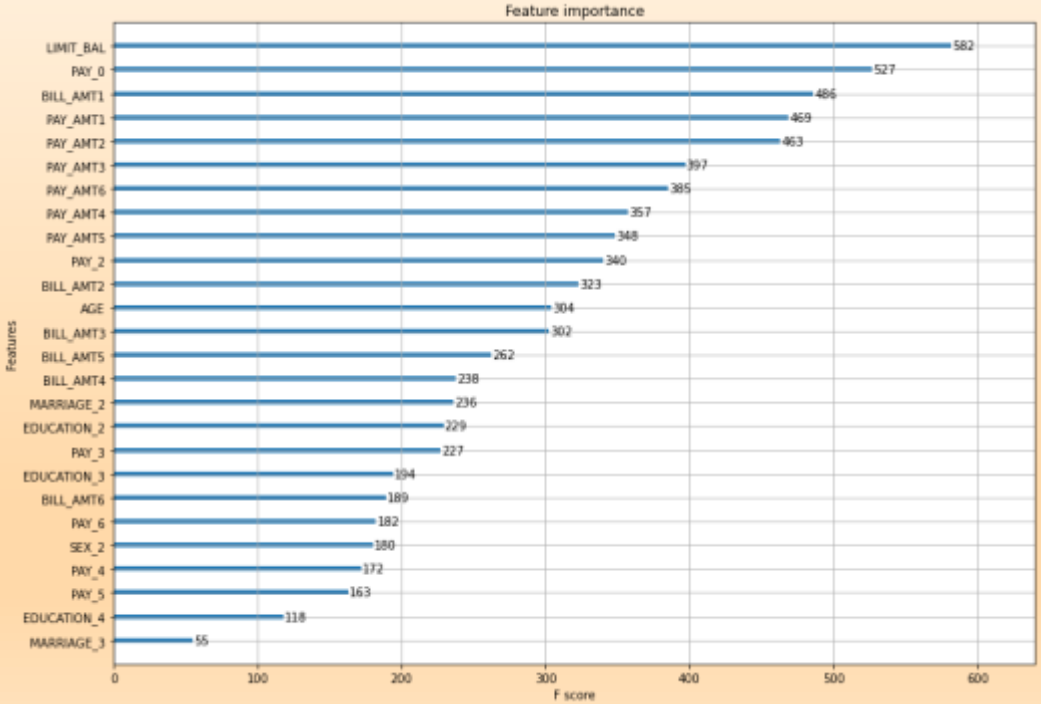



Random Forest feature importance:

- PAY_0
- PAY_2
- PAY_3
- PAY_AMT1
- SEX
- EDUCATION
- MASSIAGE

# Feature Importance

## XGBoost Classifier Feature Importance




XGBoost feature importance:

- LIMIT_BAL
- PAY_0
- BILL_AMT1
- PAY_AMT1
- PAY_AMT2
- PAY_AMT3

# Conclusions:

EDA

- The lower the limit the higher the chances for being default
- With highest limit no default is recorded
- Older people will tend to fall more in default category
- Male has higher chances of being default compared to female.
- Higher educated persons have less rate to be default whereas lower educated will maximum chances to be default.

Modelling

- XGBoost classifier is performing great with roc_auc score 0.88, whereas randomforest is second highest with 0.85 score, and decision tree classifier is with least score of 0.80.
- From the above reports we can see that xgboost is giving good accuracy precision and recall as well. In all way xgboost is performing well. Since our aim highly interested in finding positive class so we will lean towards recall in this case and XGboost is giving high recall values.
- We will go ahead and deploy XGBoost model with handling class imbalanced

# Thank you

Submit to: ALMA BETTER

## References:


- Scikit-learn - https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
- Analytics Vidya - https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/
- Machine Learning Mastery - https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/
- Stack overflow - https://stackoverflow.com/questions/40081888/xgboost-plot-importance-figure-size