# Credit Card Default Prediction


Team Members :

Arman Alam

Syed Sharin

Shyam Sundar K

Fathima K

Abdul  Rahman

# 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.
- Before approving the credit card, It is important to predict if a customer will be a defaulter or not, so that we can filter out customers who have high chances of being a defaulter and thus reduce the loss for the company.

# Contents

# Data Summary

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

- X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- X2: Gender (1 = male; 2 = female).
- X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- X4: Marital status (1 = married; 2 = single; 3 = others).
- X5: Age (year).
- X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
- X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
- X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

# Feature Engineering

- Categorical Variables
  - Labelling the Genders as Male and Female
  - Labelling the Education column into Graduate, University, High School and Others
  - Labeling the Marital Status column into Married, Single and Others
  - Binning the Age column with a bin size of 5 years each

- Repayment Status columns
  - Labelling the values  -2,-1 and 0 as 0, so as to consider these labels as payment made on time.

# Outlier treatment

As the available dataset is small, using IQR method for outlier removal was not feasible hence we have used Manual method to select the values, beyond which the records are deleted.

- Limit Balance: After Visualizing the outliers, the values more than 703000 were removed.
- Bills Columns: There are 6 columns ( April - September), each column was Visualized individually using box plot and were grouped into 11 bins each, and different values for upper and lower limits were chosen manually (September: 504218 & -2000, August: 522039,  July: 505046 & -5000, June: 467719 & -5000, May: 494494 & -5000, April: 510935 & -5000) and the records exceeding these values were removed.
- Payments Columns: There are 6 columns ( April - September), each column was Visualized individually using box plot and were grouped into 11 bins each, and different values were chosen manually (September: 238241, August: 220994,  July: 161644, June: 169363, May: 155101, April: 192242)  and the records exceeding these values were removed.

# Profile Formatting

Pandas profiling is an open source Python module with which was used for quick EDA to extract useful information.

The following code snippet was used and html file was obtained.

```python
from pandas_profiling import ProfileReport
report = ProfileReport(df)
report.to_file(output_file='output.html')
```

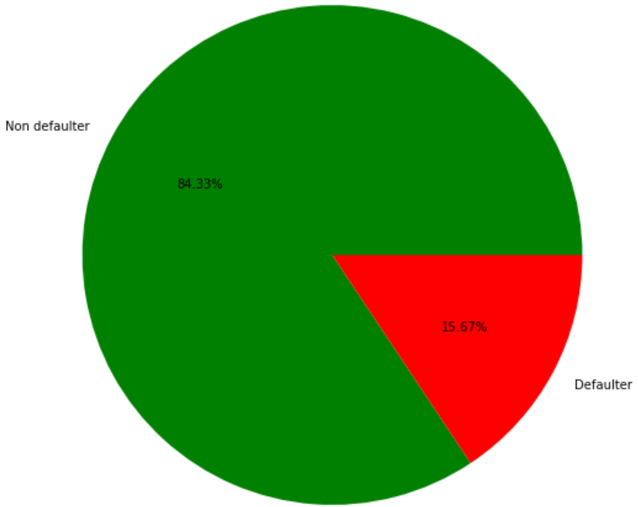
```
Summarize dataset: 39/? [01:44<00:00, 6.23s/it, Completed]
Generate report structure: 100% 1/1 [00:12<00:00, 12.92s/it]
Render HTML: 100% 1/1 [00:14<00:00, 14.27s/it]
Export report to file: 100% 1/1 [00:00<00:00, 6.85it/s]
```

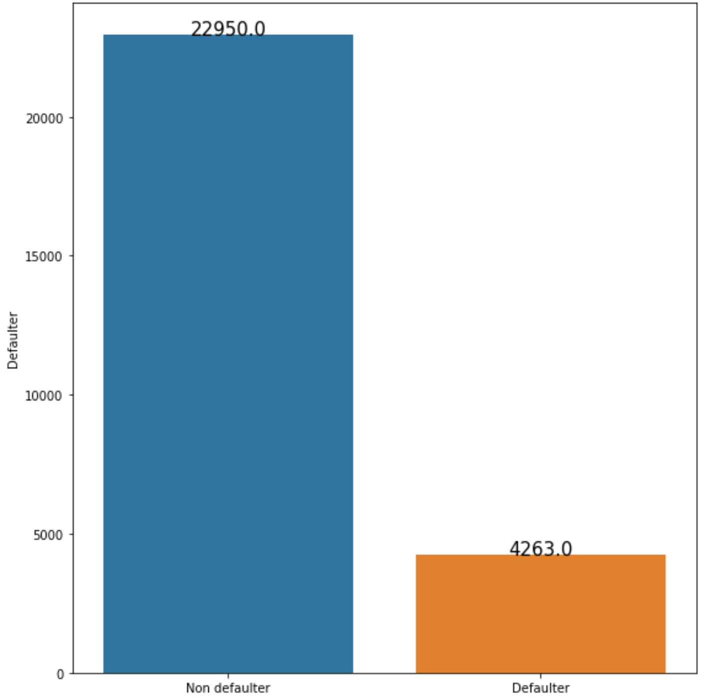
Further in depth EDA was performed as per our need which is show in our next slides.

# Exploratory Data Analysis

# Dependant Variable

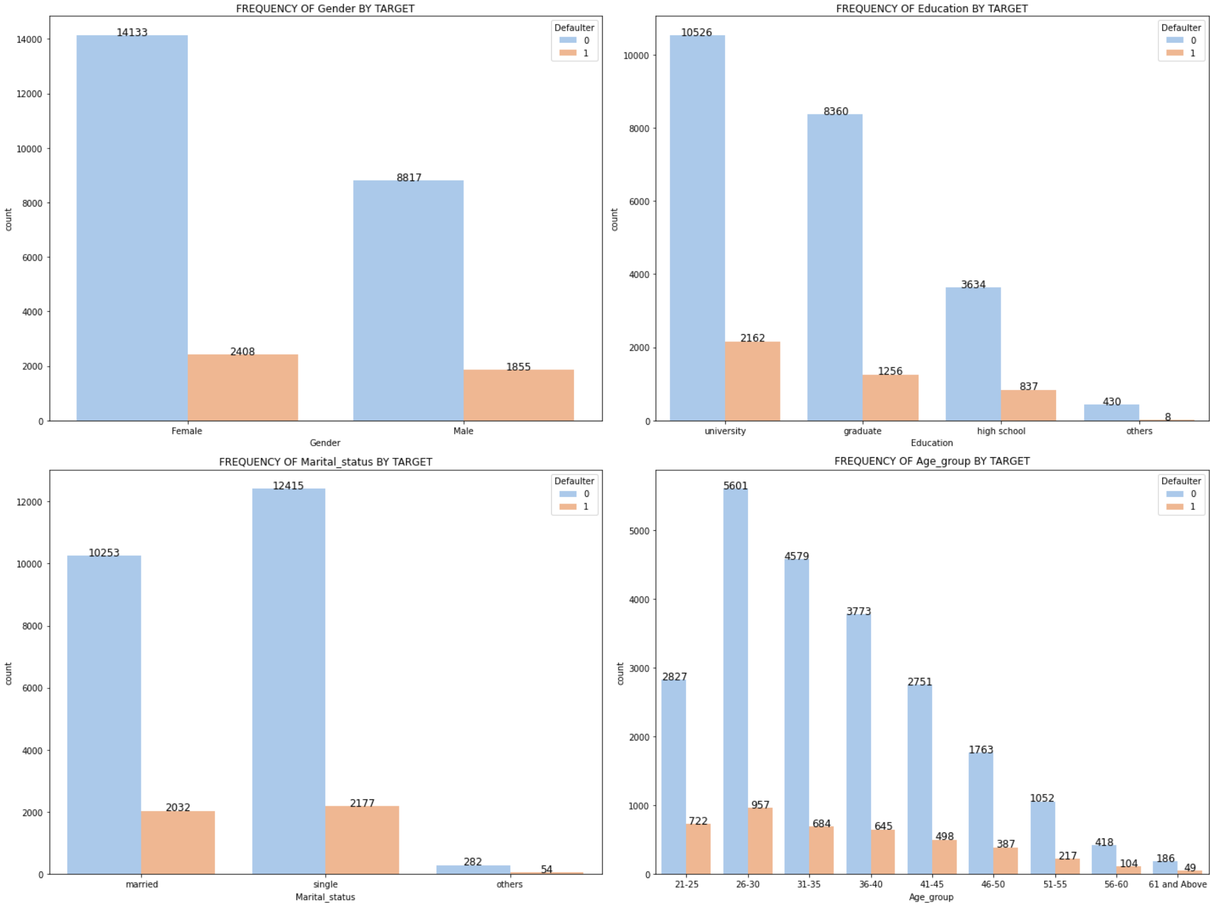
Defaulter vs non defaulter



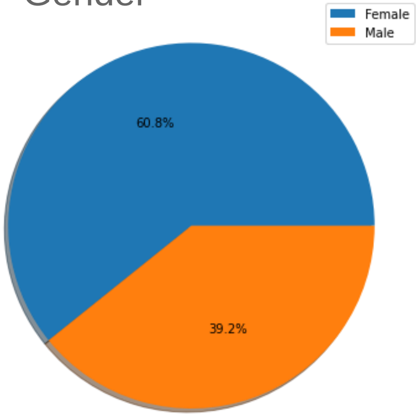

**We observe that 15.67 % of the customers are Defaulters and rest are non defaulters, thus we can say that data is imbalanced**

# Categorical variables

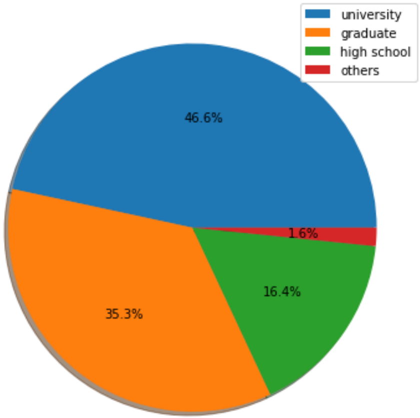



- This is the visualized comparison of Defaulters and non defaulters with respect to categorical features.
- We observe that the ratio of defaulters and non defaulters are following same trend on all the categorical features.

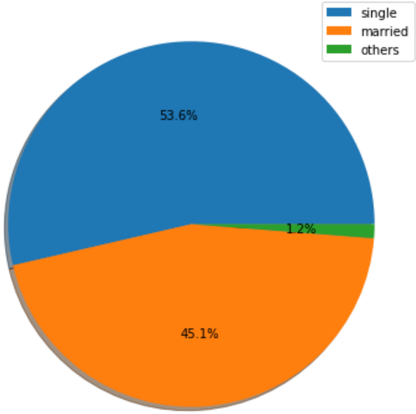
Gender



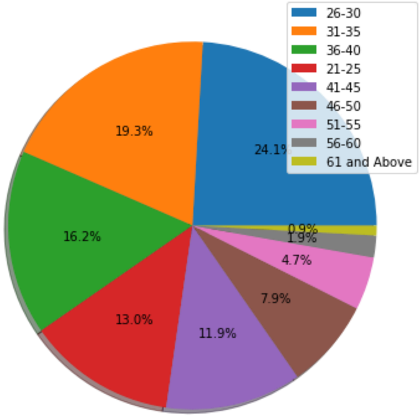

Education




Marital Status

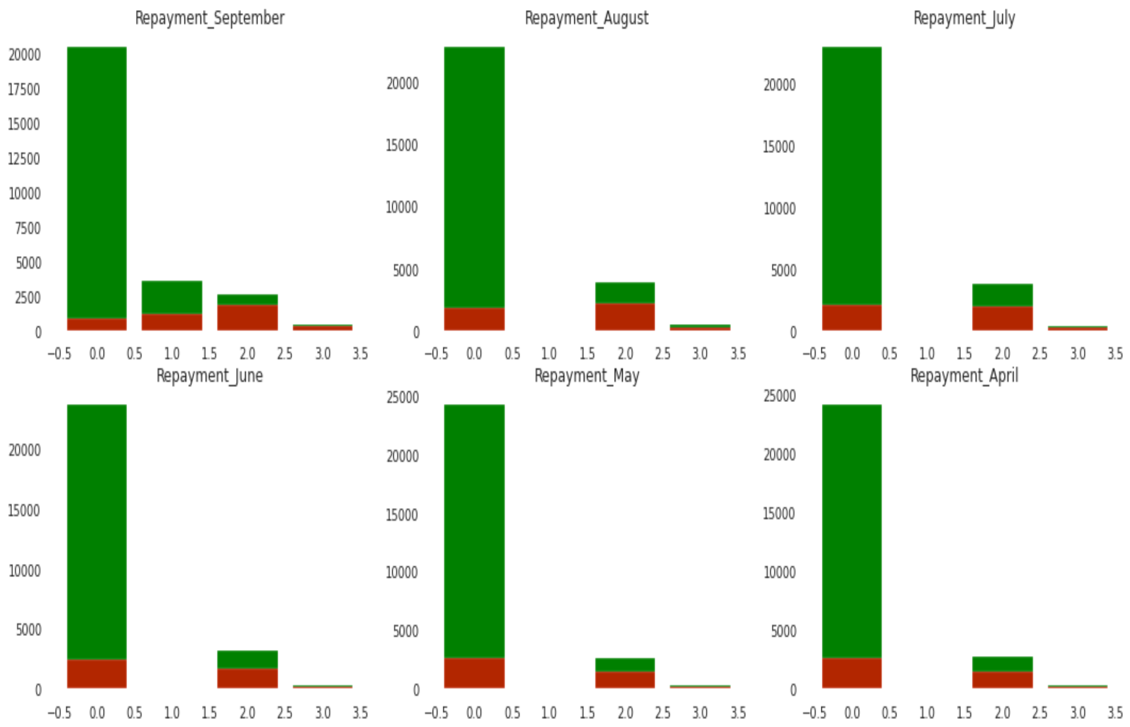



Age




- We can observe that over 60% of the customers are females and rest 39.2% are males.
- In Education column, 46.6% of the customers are from university, 35.3% are graduates, 16.4% are from high school and rest 1.6% belongs to others.
- We can observe 53.6% of the customers are single and 45.1% are married, the remaining 1.2% belongs to others.
- We can observe that as the age increases, the number of customers decreases and vice-versa.

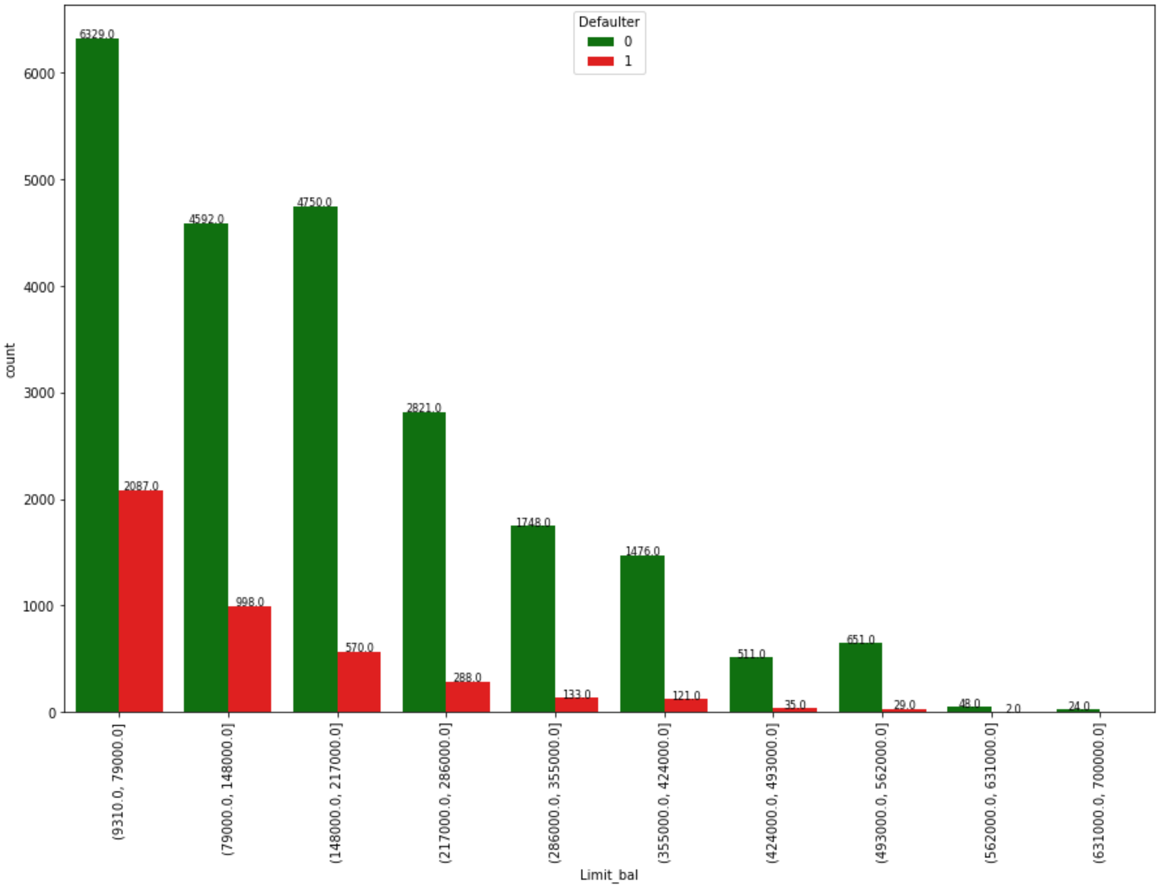
# Numerical Variables - Repayments column




- We can observe that majority of customers made repayment on time.
- We can also observe that in case of delay in repayment, it is usually 2 months of delay.

0: Repayment on time
1: One month delay
2: Two months delay
3: Three months delay

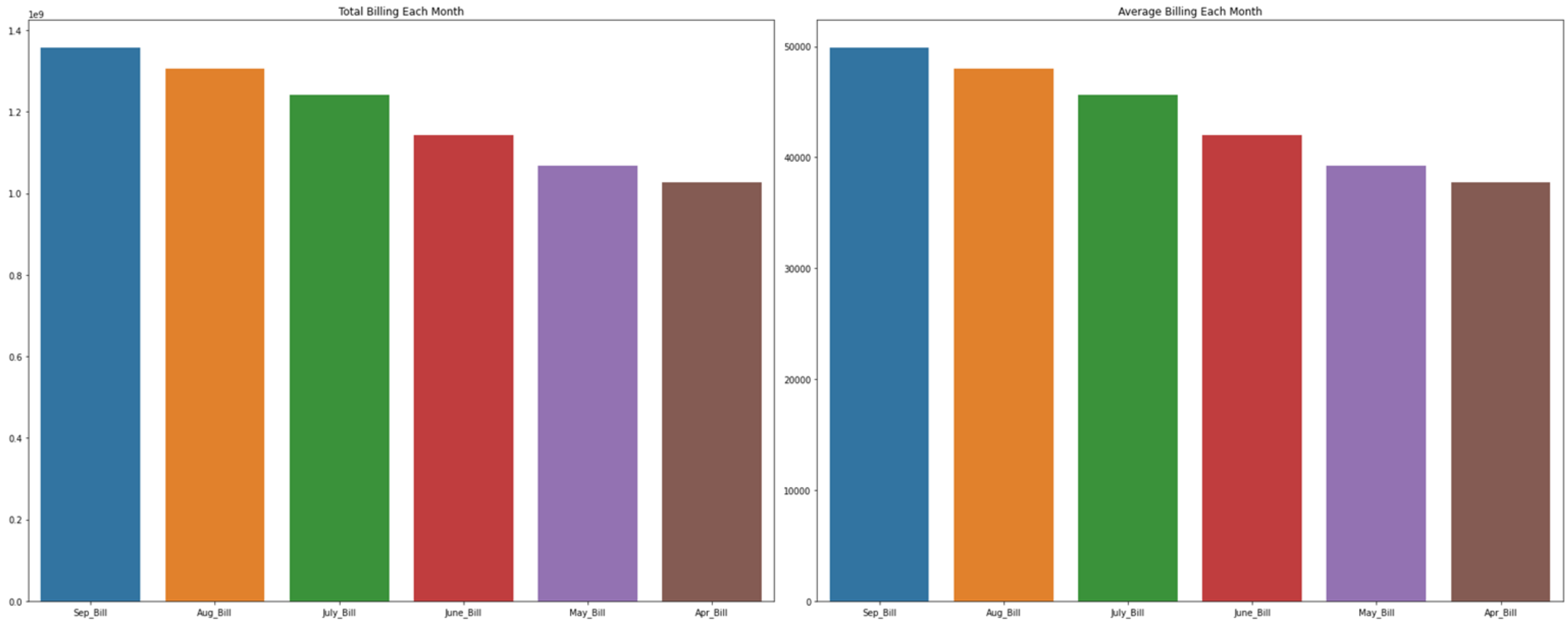
# Numerical Variables - Limit Balance




Most of the customers have a limit balance of less than NT$424,000.

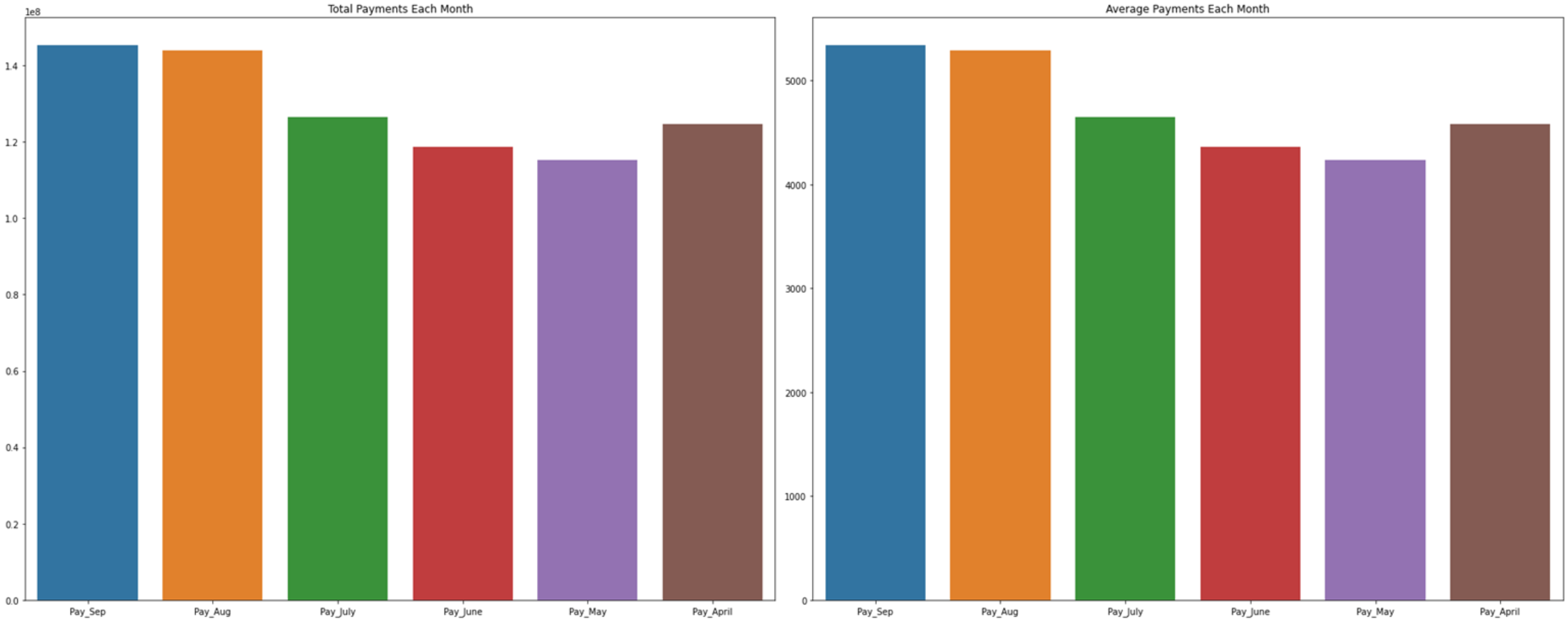
We observed that the limit balance does not have a high impact on the customer being a defaulter.

# Numerical Variables - Monthly Bills



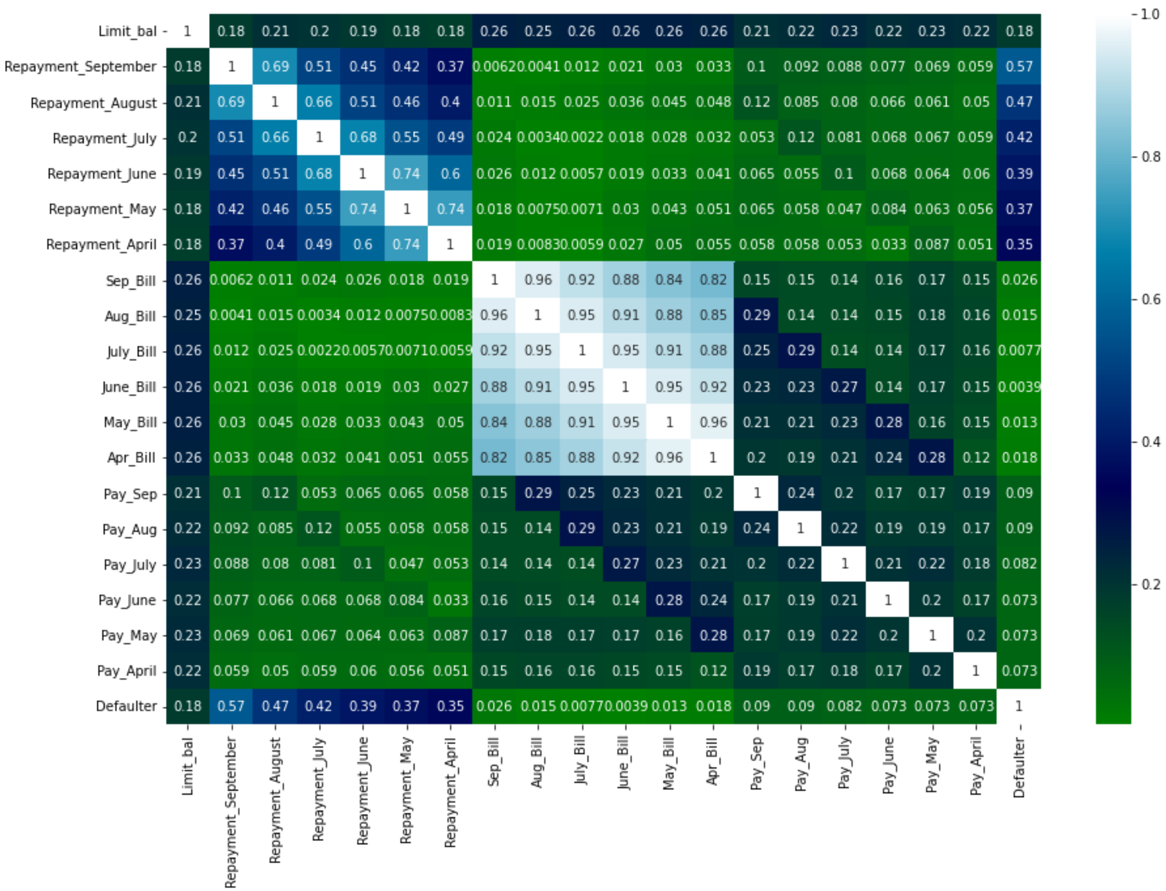

It is observed that the both total biling of the month and the average billing has an increasing trend over each month, (The graph is show in decreasing order of months).

# Numerical Variables - Monthly Payments




Overall payments were slightly higher in the month of April and we can see a dip in the month of May and from there we can see an increase in payments every month. (The graph is show in reverse  order of months).

# Correlation Graph




It is observed that repayment columns and limit balance have higher correlation with the dependant variable.

# Insights from EDA

- 22 % of the customers are Defaulters and rest are non-defaulters, we observed data is imbalanced
- This is the visualized comparison of Defaulters and non-defaulters with respect to categorical features. and we observe that the ratio of defaulters and non-defaulters are following same trend on all the categorical features
- In the given data, we observed that there are more female customers with 60.4% than male customers with 39.6%, however men have slightly higher chance of being a defaulter than women.
- There is higher no of customers from university level, and next highest is from graduate level and 3 comes the high school level and a small portion of customers from others. The customers from high school level have the highest of 25% chance of being a defaulter and other categories have the least chance of 7.1% of being a defaulter.
- The customers are mostly single or married, and we can notice that marries customers have a slightly higher chance of 23.1% being a defaulter compared to singles having 20.9% chance of being a defaulter.
- The probability of customer being Defaulter is directly proportional to delay in repayment and When Customers are pay duly then the count of defaulter is low.
- There is increase trend in the average billing of the month, it was lowest in the month of April and highest in the month of September

# Optimization

Before moving into performance metrics, let's discuss optimization. What metric exactly are we optimizing? In this case, we are optimizing recall.

Ideally, we do not want to allow any defaults to fall through the cracks, so our optimal model will minimize False Negatives (So Recall Score is as high as possible).

# Defining Function

By defining the function, we are creating a template to show the reports.

Two Functions to show the following reports. One with cross validation and hyperparameter tuning, and other without it.

- Training score
- Metrics scores on Train and Test Set
  - Accuracy Score
  - Precision Score
  - Recall Score
  - F1 Score
  - ROC Score
- Classification Report
- Confusion Matrix
- ROC AUC Curve

Three more functions were created for model Explainability.

- Lime
- ELI5
- Shap

# Modelling

The following algorithms were built and hyperparameter tuning and cross validation was performed.

1. **Logistic Regression**
2. **Stochastic Gradient Descent**
3. **Support Vector Classifier**
4. **K Nearest Neighbor**
5. **Decision Tree**
6. **Random Forest**
7. **AdaBoostClassifier**
8. **Gradient Boosting**
9. **Extreme Gradient Boosting**
10. **Stacking**

# Final Results before Hyperparameter Tuning And Cross Validation

| | model | Train accuracy score | Test accuracy score | Train precision score | Test precision score | Train recall score | Test recall score | Train f1 score | Test f1 score | Train roc score | Test roc score |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Random Forest | 0.998693 | 0.929739 | 0.997926 | 0.910338 | 0.999453 | 0.955522 | 0.998689 | 0.932383 | 0.998695 | 0.929374 |
| 1 | Extreme Gradient Boosting | 0.906590 | 0.902832 | 0.890717 | 0.891875 | 0.926159 | 0.919854 | 0.908092 | 0.905648 | 0.906658 | 0.902592 |
| 2 | Gradient Boosting | 0.907707 | 0.902614 | 0.892517 | 0.893141 | 0.926323 | 0.917705 | 0.909106 | 0.905256 | 0.907772 | 0.902401 |
| 3 | Decision Tree | 0.998693 | 0.897712 | 0.998852 | 0.895634 | 0.998524 | 0.903524 | 0.998688 | 0.899561 | 0.998692 | 0.897630 |
| 4 | K Nearest Neighbor | 0.926225 | 0.892157 | 0.924275 | 0.895851 | 0.927962 | 0.890847 | 0.926115 | 0.893342 | 0.926232 | 0.892175 |
| 5 | Support Vector Classifier | 0.902669 | 0.890959 | 0.901582 | 0.895603 | 0.903258 | 0.888483 | 0.902419 | 0.892029 | 0.902671 | 0.890994 |
| 6 | Stochastic Gradient Descent | 0.886547 | 0.883769 | 0.882554 | 0.883472 | 0.890850 | 0.887838 | 0.886683 | 0.885650 | 0.886562 | 0.883712 |
| 7 | Logistic Regression | 0.884504 | 0.882571 | 0.889141 | 0.890564 | 0.877624 | 0.876021 | 0.883345 | 0.883232 | 0.884480 | 0.882663 |

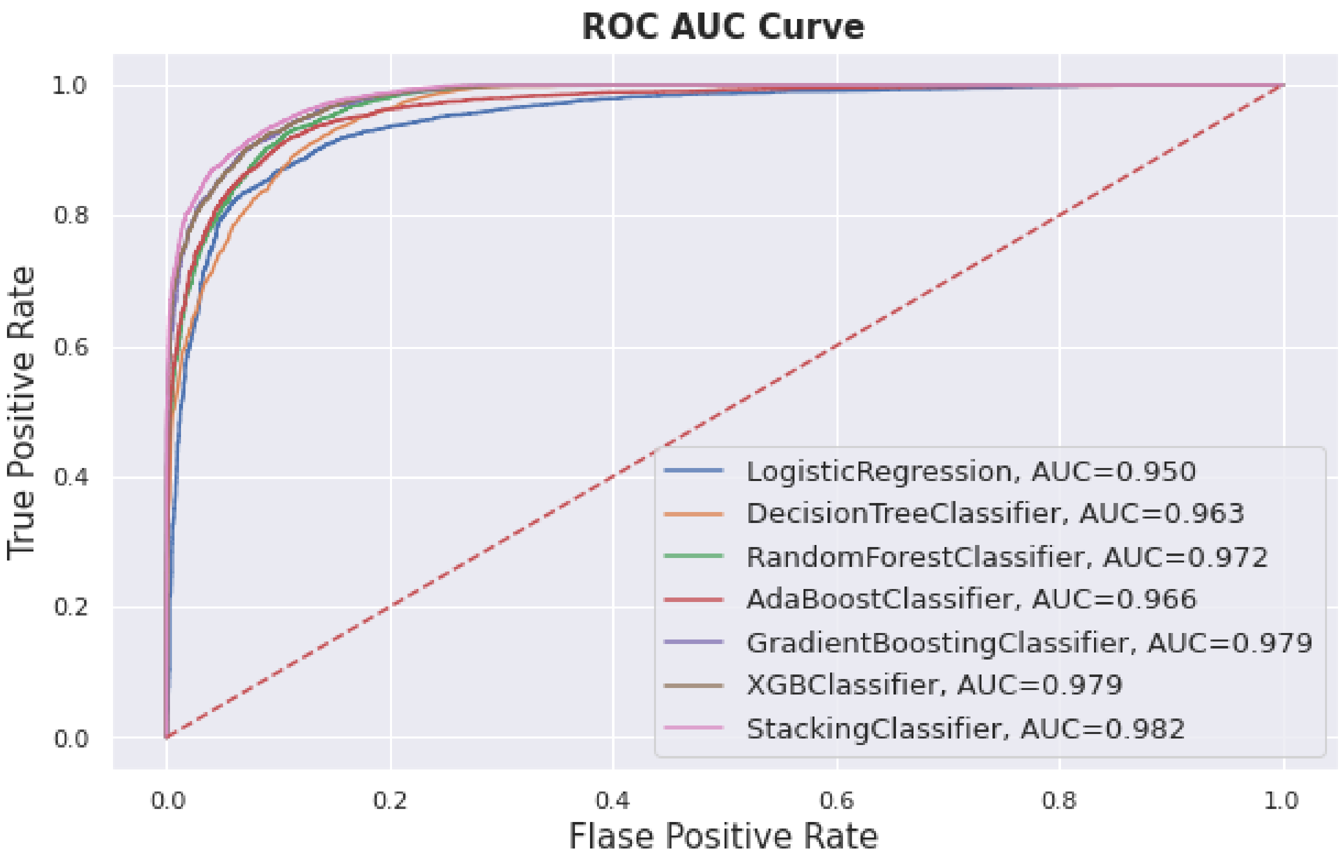
Random Forest Classifier has given the best performance, but we can observe that the model is overfitted. Over all, XGBC and Gradient Boosting also performs the better with train recall of 0.92 and test recall of 0.91

# Final Results after Hyperparameter Tuning And Cross Validation

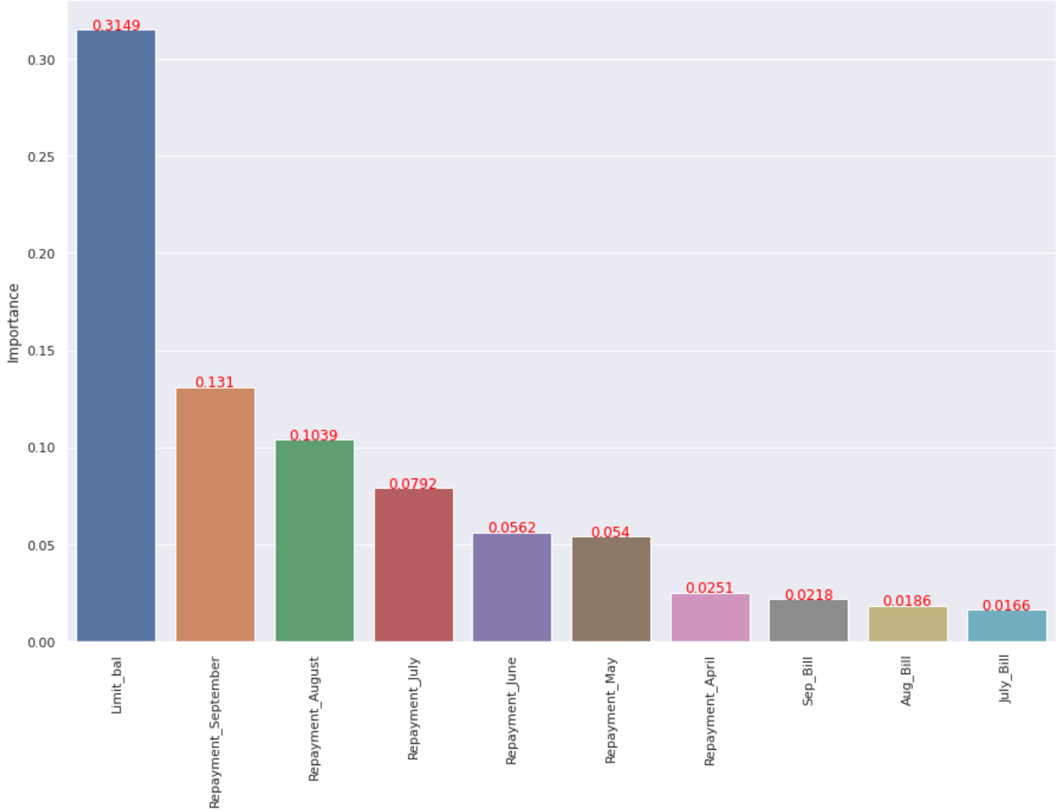
| | model | Train accuracy score | Test accuracy score | Train precision score | Test precision score | Train recall score | Test recall score | Train f1 score | Test f1 score | Train roc score | Test roc score |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Random Forest | 0.911629 | 0.902941 | 0.870769 | 0.867265 | 0.966003 | 0.954663 | 0.915917 | 0.908868 | 0.911817 | 0.902210 |
| 1 | Gradient Boosting | 0.952560 | 0.914379 | 0.938633 | 0.899917 | 0.968080 | 0.935110 | 0.953129 | 0.917176 | 0.952614 | 0.914086 |
| 2 | Extreme Gradient Boosting | 0.928813 | 0.916667 | 0.919619 | 0.911012 | 0.939222 | 0.926085 | 0.929317 | 0.918487 | 0.928849 | 0.916533 |
| 3 | Stacking Classifier | 0.951443 | 0.920261 | 0.947240 | 0.918838 | 0.955783 | 0.924366 | 0.951492 | 0.921594 | 0.951458 | 0.920203 |
| 4 | Decision Tree | 0.894363 | 0.890523 | 0.870407 | 0.871816 | 0.925831 | 0.919209 | 0.897264 | 0.894885 | 0.894472 | 0.890117 |
| 5 | Support Vector Classifier | 0.912064 | 0.892048 | 0.905441 | 0.891765 | 0.919545 | 0.895789 | 0.912439 | 0.893772 | 0.912090 | 0.891995 |
| 6 | Ada Boost Classifier | 0.906236 | 0.900436 | 0.911646 | 0.910628 | 0.898940 | 0.891061 | 0.905248 | 0.900738 | 0.906211 | 0.900568 |
| 7 | K Nearest Neighbour | 0.916830 | 0.888998 | 0.916721 | 0.895711 | 0.916321 | 0.883971 | 0.916521 | 0.889802 | 0.916828 | 0.889069 |
| 8 | Logistic Regression | 0.884858 | 0.882571 | 0.888404 | 0.889034 | 0.879373 | 0.877954 | 0.883865 | 0.883459 | 0.884839 | 0.882636 |
| 9 | Stochastic Gradient Descent | 0.890523 | 0.889978 | 0.908212 | 0.911101 | 0.868004 | 0.867641 | 0.887653 | 0.888840 | 0.890445 | 0.890294 |

After performing hyper parameter tuning and cross validation, there is a improvement in the performance. The Random Forest classifier has the best recall score of 0.95 on test set followed by Gradient Boosting XGBoost model having a recall score of 0.93 and 0.92 on test set.

# ROC AUC Curve

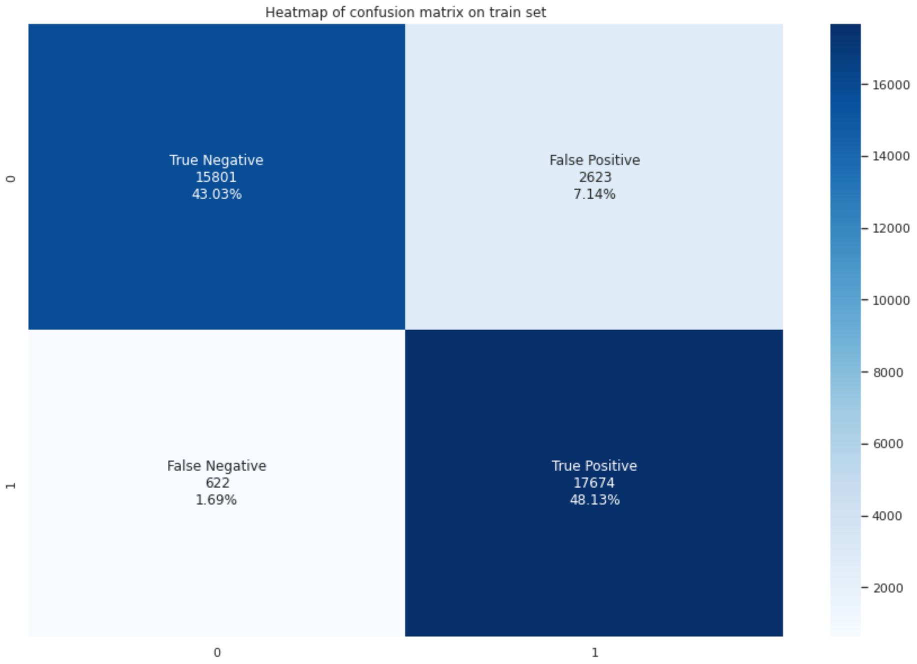


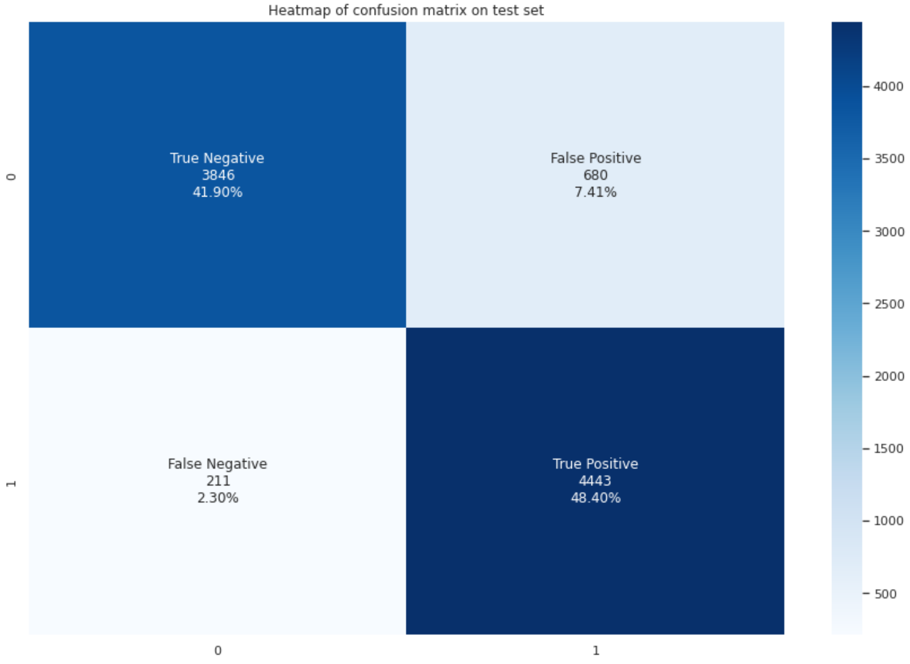
# Model Explainability - Random Forest




Top 10 Important features are shown in the figure along with their relative importance. Limit Balance has the highest significance on the final prediction.

# Confusion Matrix - Random Forest



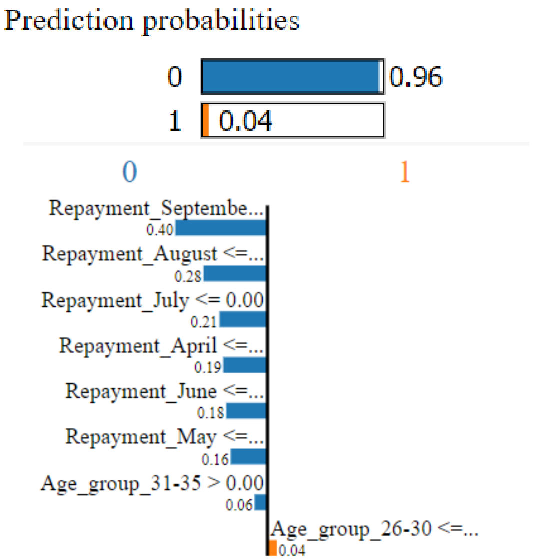

As per the use case, we need the false negative as low as possible. Train set has a false negative of 7.14% and test set has 7.41%.

# ELI5 AND Lime

**y=1** (probability **0.836**) top features

| Contribution? | Feature | Value |
|---|---|---|
| +0.498 | <BIAS> | 1.000 |
| +0.198 | Repayment_September | 2.000 |
| +0.134 | Repayment_August | 2.000 |
| +0.036 | Marital_status_single | 0.000 |
| +0.021 | Age_group_26-30 | 0.000 |
| +0.020 | Age_group_31-35 | 0.000 |
| +0.019 | Pay_Aug | 0.000 |
| +0.017 | Age_group_36-40 | 0.000 |
| +0.012 | Age_group_41-45 | 0.000 |
| +0.012 | Gender_Male | 0.000 |
| +0.011 | Pay_July | 0.000 |
| +0.009 | Pay_June | 0.000 |
| +0.007 | Age_group_46-50 | 0.000 |
| +0.007 | Education_high school | 0.000 |
| +0.006 | Pay_May | 0.000 |
| +0.003 | Pay_April | 0.000 |
| +0.003 | Sep_Bill | 77669.000 |
| +0.003 | June_Bill | 0.000 |
| +0.002 | Age_group_51-55 | 0.000 |
| +0.002 | May_Bill | 0.000 |
| +0.000 | Age_group_56-60 | 0.000 |
| +0.000 | Education_others | 0.000 |
| +0.000 | Marital_status_others | 0.000 |
| -0.003 | Apr_Bill | 0.000 |
| -0.003 | July_Bill | 0.000 |
| -0.004 | Pay_Sep | 0.000 |
| -0.014 | Repayment_April | 0.000 |
| -0.015 | Aug_Bill | 0.000 |
| -0.020 | Repayment_May | 0.000 |
| -0.021 | Limit_bal | 84371.000 |
| -0.027 | Repayment_June | 0.000 |
| -0.036 | Education_university | 1.000 |
| -0.039 | Repayment_July | 0.000 |

ELI5



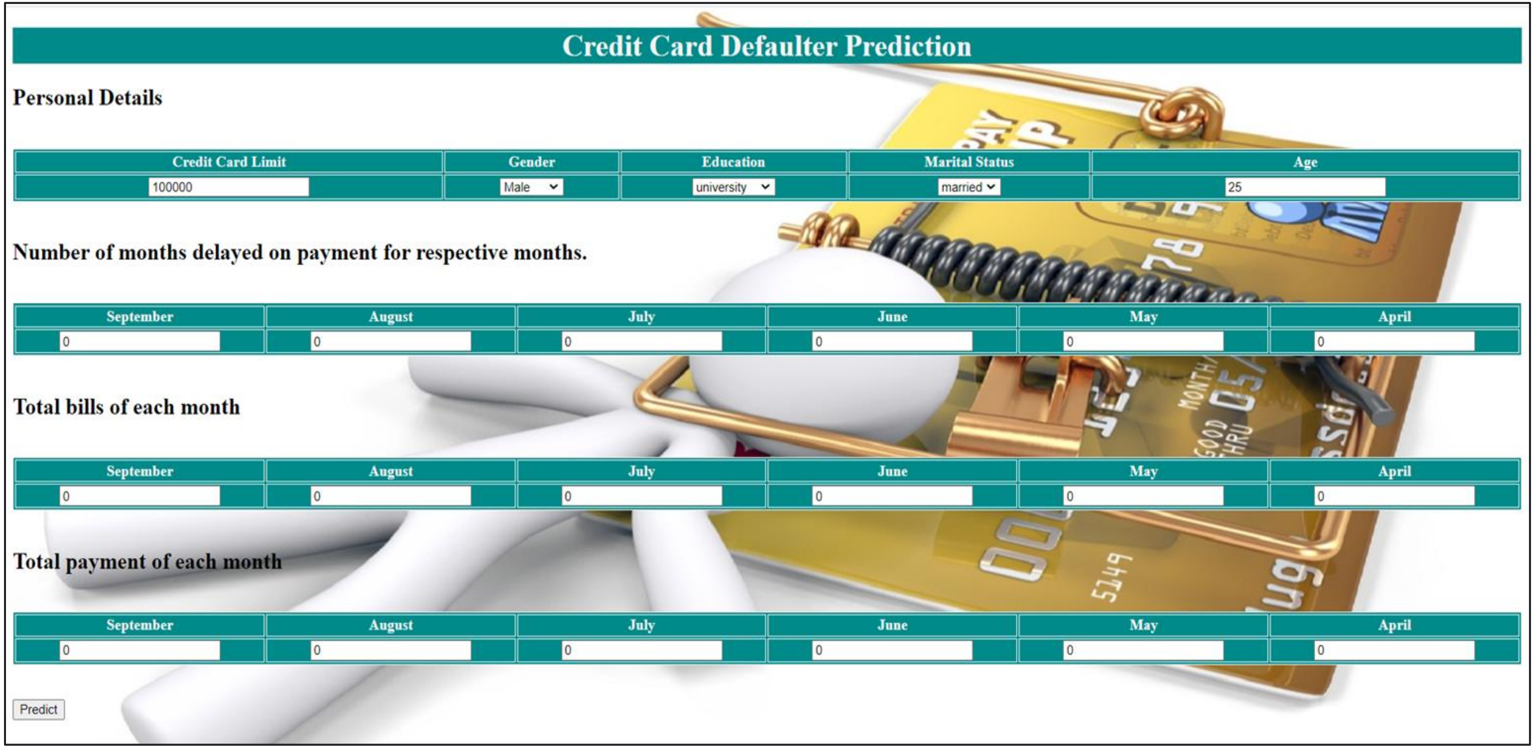

| Feature | Value |
|---|---|
| Repayment_September | 0.00 |
| Repayment_August | 0.00 |
| Repayment_July | 0.00 |
| Repayment_April | 0.00 |
| Repayment_June | 0.00 |
| Repayment_May | 0.00 |
| Age_group_31-35 | 1.00 |
| Age_group_26-30 | 0.00 |

Lime

ELI5 and Lime was coded to check the model explainability and the above tables were obtained for **first row in our dataset**. We can observe that, for our selected point Repayment September has the most significant effect on the final Prediction. The process can be checked on any data point on the dataset for similar tables.

# Model Deployment

The model was deployed for public use (https://creditcarddefaulter-nobi.herokuapp.com/) with the help of website herokuapp.com. Flask framework was used to create the interface. Interface is shown below.



# Challenges

- The data had a imbalanced data set with the dependant variable having two classifications of 77.79% and 22.21% respectively.
- We noticed that, in payments and bills columns there were outliers and outlier treatment was necessary. Using IQR and Percentile method, many values were removed which is not acceptable considering the size of the original data. Hence manually the values were selected for each column and the rows having values beyond that was removed.
- As multiple models were run along with cross validation and hyperparameter tuning, the computational time was quite high. Gradient boost, XGBoost and stacking had the highest computational time.

# Conclusions

- Random Forest Classifier performs the best with a recall of 0.95 on the test set.
- Followed by RFC we observed that Gradient Boosting and XGBoost performed the best with test recall score of 0.93 and 0.92 respectively.
- We could overcome the overfitting of multi models by using hyperparameter tuning and cross validation, which is evident by comparing the scores before and after.
- Model explainability was performed on the Random Forest Classifier using Lime and ELI5, Repayments columns (especially the latest months) has the most significant impact on whether a customer is a Defaulter or Not.

# THANK YOU