# BFS Capstone Project

Mid-Term Submission

***Group Members:***

***Payel Jain***

***Ranip Hore***

***Rishikesh Ojha***

***Sanchari Gautam***

## Context:

CredX is a leading credit card provider that gets thousands of credit card applications every year. But in the past few years, it has experienced an increase in credit loss. The CEO believes that the best strategy to mitigate credit risk is to ‘acquire the right customers’.

## Business Task:

To identify the right customers using predictive models. Using past data of the bank’s applicants, we need to determine the factors affecting credit risk, create strategies to mitigate the acquisition risk and assess the financial benefit.

## Business Strategy:

- We aspire to build several supervised approach of classification algorithms like Random Forest, Logistic Regression, SVM etc., after evaluation of which we will move on with the model with highest ROC curve area or similar metrics.
- The columns as determined by the chosen model would be considered worthy factors affecting our credit risk.
- Finally we will use credit scoring techniques that assess the risk in lending to a particular customer and build a scorecard model, the scores of which will determine the likelihood of a customer defaulting on a credit obligation. Thus we will be able to assess the financial benefit of the model.

**Data Understanding:**

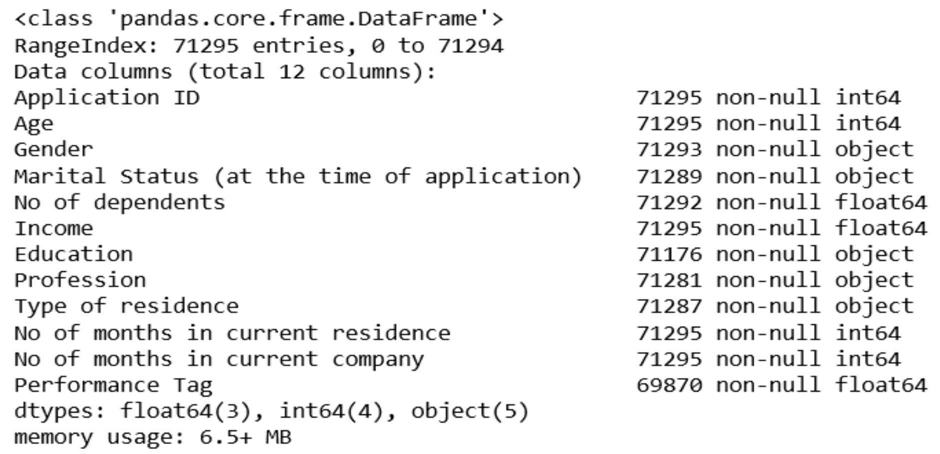
Two datasets are provided, demographic data and credit bureau data.

1.Demographic/application data: This dataset contains the information provided by the applicants at the time of credit card application. It contains customer-level information on age, gender, income, marital status, etc.

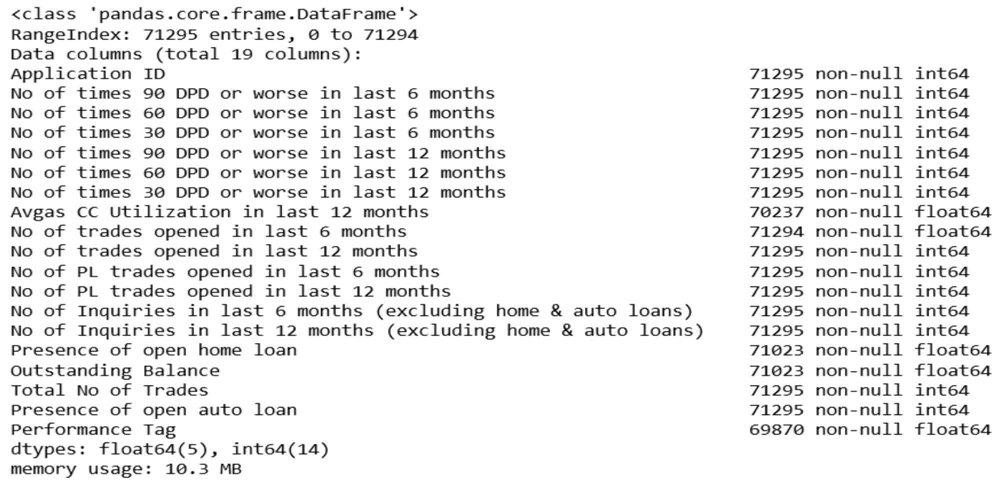
2. Credit bureau data: This information is taken from the credit bureau and contains variables such as 'number of times 30 DPD or worse in last 3/6/12 months', 'outstanding balance', 'number of trades', etc.

Nature of data:

- The demographic data consists of 71295 observations with 12 variables.
- The credit bureau data consists of 71295 observations with 19 variables.
- Application ID is the common key between the two datasets for merging.
- Performance Tag is the target variable which depicts if  customer is default or not. The values are 0(non-default) and 1(default).



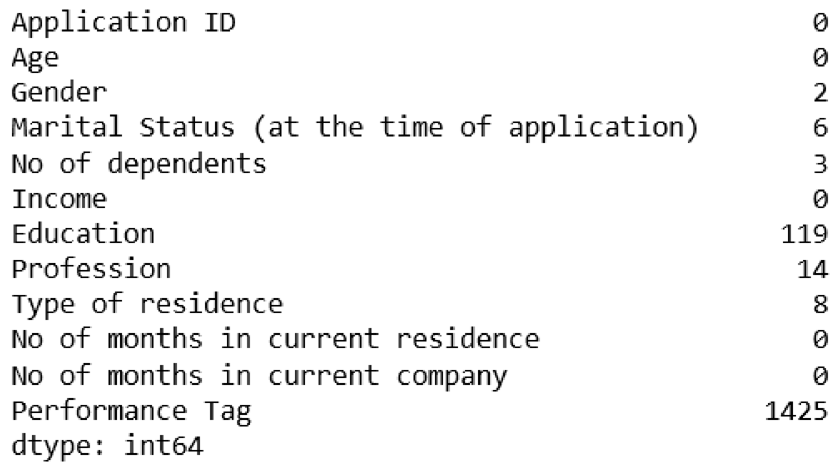
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71295 entries, 0 to 71294
Data columns (total 12 columns):
Application ID                                 71295 non-null int64
Age                                            71295 non-null int64
Gender                                         71293 non-null object
Marital Status (at the time of application)    71289 non-null object
No of dependents                               71292 non-null float64
Income                                         71295 non-null float64
Education                                      71176 non-null object
Profession                                     71281 non-null object
Type of residence                              71287 non-null object
No of months in current residence              71295 non-null int64
No of months in current company                71295 non-null int64
Performance Tag                                69870 non-null float64
dtypes: float64(3), int64(4), object(5)
memory usage: 6.5+ MB
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71295 entries, 0 to 71294
Data columns (total 19 columns):
Application ID                                                    71295 non-null int64
No of times 90 DPD or worse in last 6 months                      71295 non-null int64
No of times 60 DPD or worse in last 6 months                      71295 non-null int64
No of times 30 DPD or worse in last 6 months                      71295 non-null int64
No of times 90 DPD or worse in last 12 months                     71295 non-null int64
No of times 60 DPD or worse in last 12 months                     71295 non-null int64
No of times 30 DPD or worse in last 12 months                     71295 non-null int64
Avgas CC Utilization in last 12 months                            70237 non-null float64
No of trades opened in last 6 months                              71294 non-null float64
No of trades opened in last 12 months                             71295 non-null int64
No of PL trades opened in last 6 months                           71295 non-null int64
No of PL trades opened in last 12 months                          71295 non-null int64
No of Inquiries in last 6 months (excluding home & auto loans)    71295 non-null int64
No of Inquiries in last 12 months (excluding home & auto loans)   71295 non-null int64
Presence of open home loan                                        71023 non-null float64
Outstanding Balance                                               71023 non-null float64
Total No of Trades                                                71295 non-null int64
Presence of open auto loan                                        71295 non-null int64
Performance Tag                                                   69870 non-null float64
dtypes: float64(5), int64(14)
memory usage: 10.3 MB
```

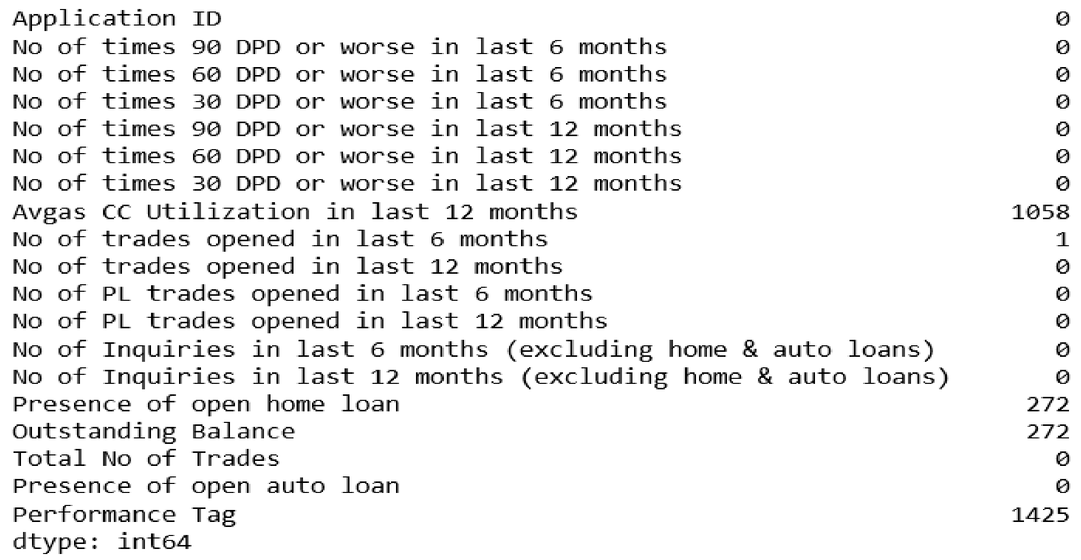
## Data Cleansing of Demographic Data:

- 1425 rows out of 71295 rows in *Performance Tag* column have null values which indicates that the applicants are not given credit card. Hence these records are deleted.
- 6 different records having 3 duplicate *Application IDs* (653287861, 765011468 and 671989187) are found in the dataset. Hence 3 records having separate Application IDs are kept and rest are removed.
- *Age* column has negative values which are erroneous data and are removed from the dataset.
- The NA values in the *Gender* column are replaced by the value having highest frequency, which is 'Male'.
- There are 81 records having negative *Income* values, which are removed from the dataset.
- The NA values in the *Marital Status* column are replaced by 'Married' value, as that has the highest frequency for the rest of the dataset.
- The NA values in *Types of Residence* Column and *Education* column are replaced by 'Others'.
- After cleansing the dataset having 71295 records, we find a dataset which has the final shape of **69752 records with 12 variables.**
- Few column names have space at the end, which are trimmed.


```
Application ID                                   0
Age                                              0
Gender                                           2
Marital Status (at the time of application)      6
No of dependents                                 3
Income                                           0
Education                                      119
Profession                                      14
Type of residence                                8
No of months in current residence                0
No of months in current company                  0
Performance Tag                               1425
dtype: int64
```

## Data Cleansing of Credit Bureau Data:

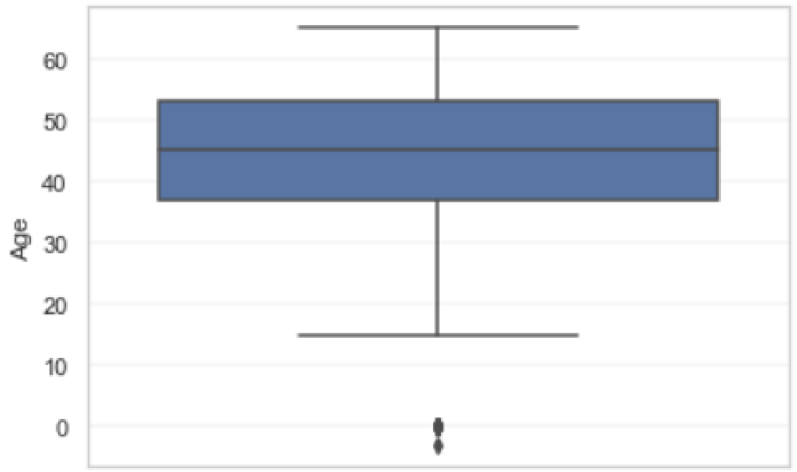
- 1425 rows out of 71295 rows in *Performance Tag* column have null values which indicates that the applicants are not given credit card. Hence these records are deleted.
- 6 different records having 3 duplicate *Application IDs* (653287861, 765011468 and 671989187) are found in the dataset. Hence 3 records having separate Application IDs are kept and rest are removed.
- *Avgas_CC_Utilization_in_last_12_months* Column has 1058 NA values and are removed from the dataset.
- After cleansing the dataset having 71295 records, we find a dataset which has the final shape of **68844 records with 19 variables.**

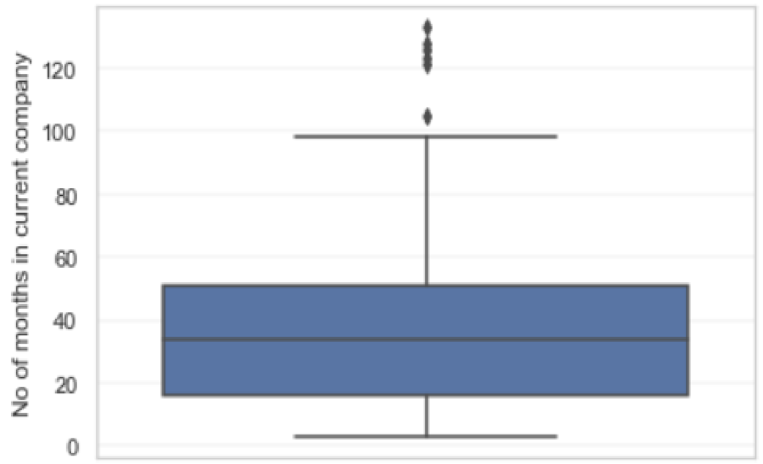


```
Application ID                                                        0
No of times 90 DPD or worse in last 6 months                          0
No of times 60 DPD or worse in last 6 months                          0
No of times 30 DPD or worse in last 6 months                          0
No of times 90 DPD or worse in last 12 months                         0
No of times 60 DPD or worse in last 12 months                         0
No of times 30 DPD or worse in last 12 months                         0
Avgas CC Utilization in last 12 months                             1058
No of trades opened in last 6 months                                  1
No of trades opened in last 12 months                                 0
No of PL trades opened in last 6 months                               0
No of PL trades opened in last 12 months                              0
No of Inquiries in last 6 months (excluding home & auto loans)        0
No of Inquiries in last 12 months (excluding home & auto loans)       0
Presence of open home loan                                          272
Outstanding Balance                                                 272
Total No of Trades                                                    0
Presence of open auto loan                                            0
Performance Tag                                                    1425
dtype: int64
```

## Outlier Treatment Of Demographic Data:

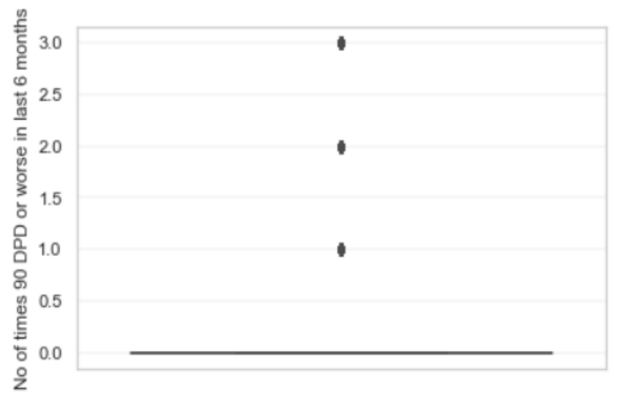
- The Age column has some outlier values which are negative and are already handled in the data cleansing part.
- The number of months in current company column has some outlier values above 75 %tile, which are treated by normalizing the values using standardization before feeding into the prediction model. Other columns where outlier values are needed to be treated to get a better output of the model, are treated by capping the outliers to the nearest non outlier value.
- Hence scaling is performed on all the columns except Application ID and Performance Model to standardize the data.

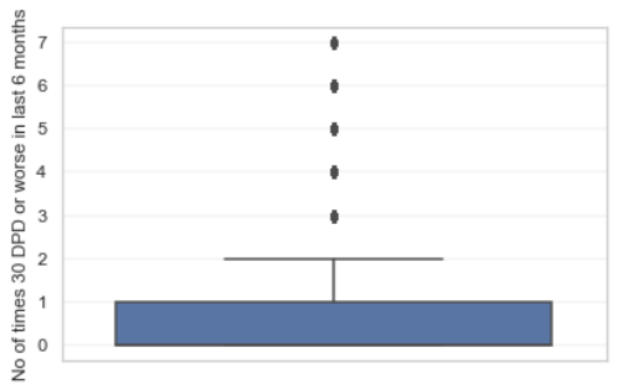


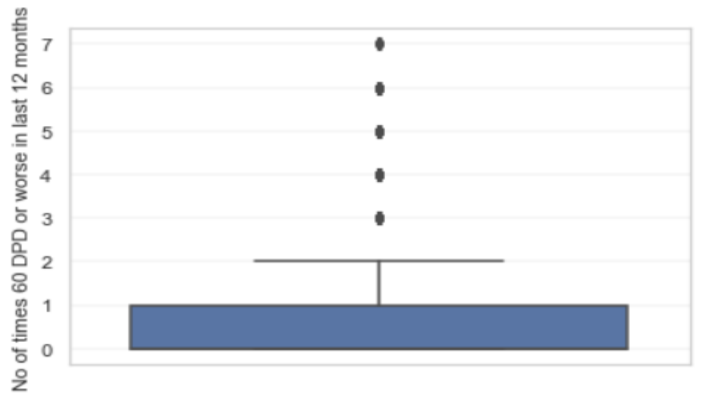


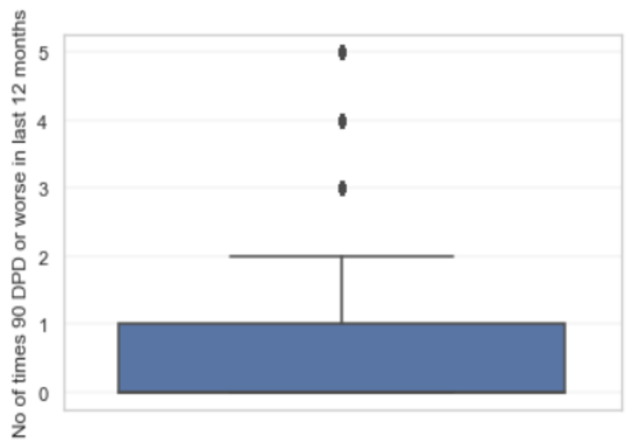
**Outlier Treatment Of Credit Bureau Data:**

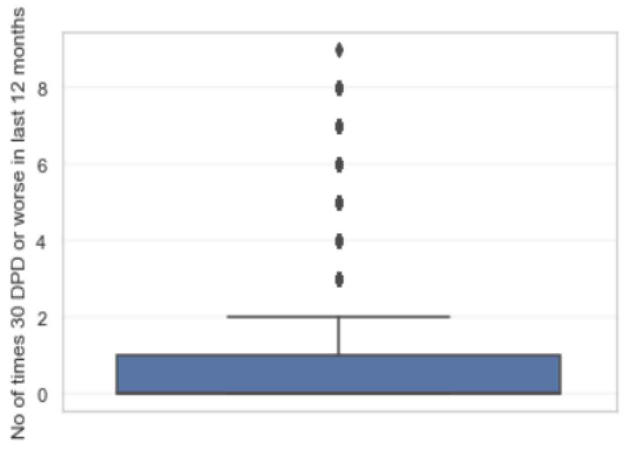
The columns having outlier values are shown below, which are expected to have some outlier and removing or replacing them would manipulate the dataset. Hence normalizing them or capping them to the nearest non outlier value would be the best approach to solve the outlier treatment.

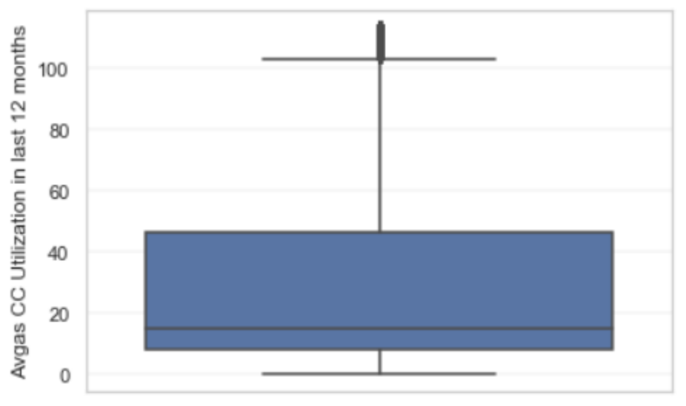






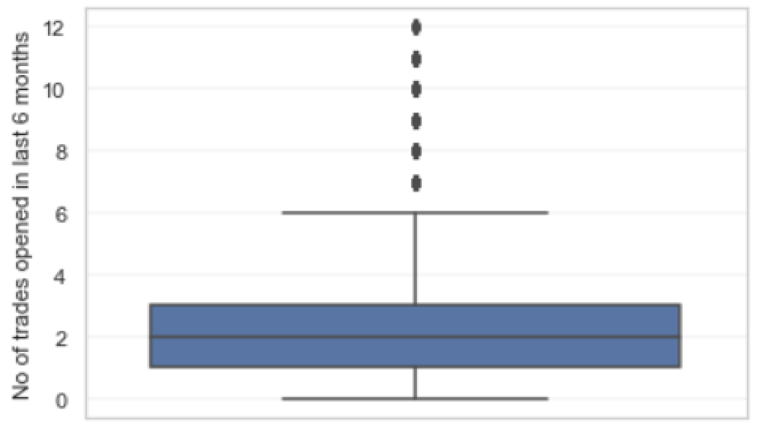


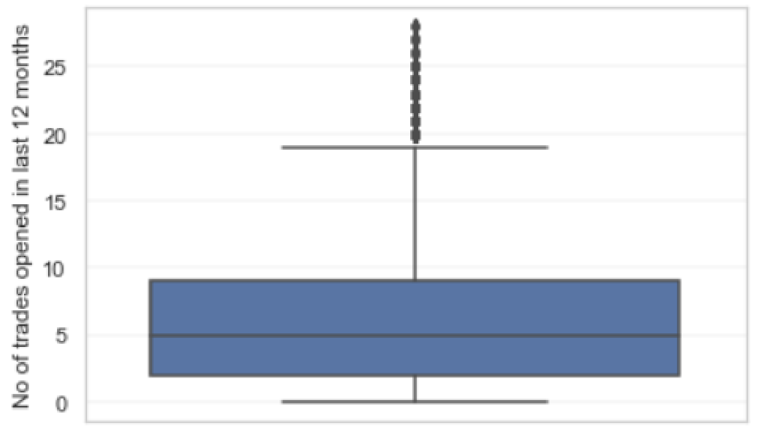


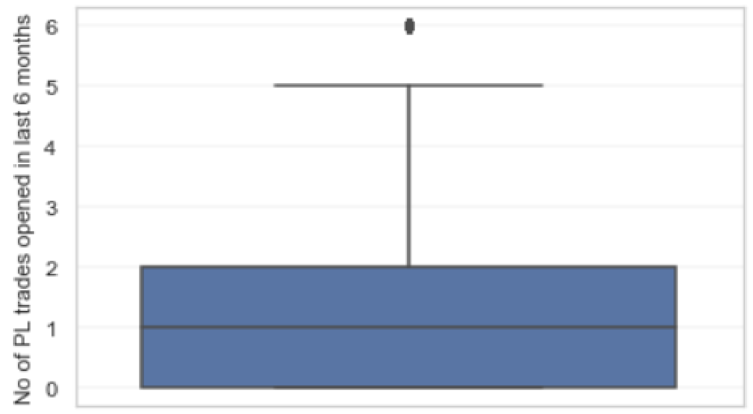


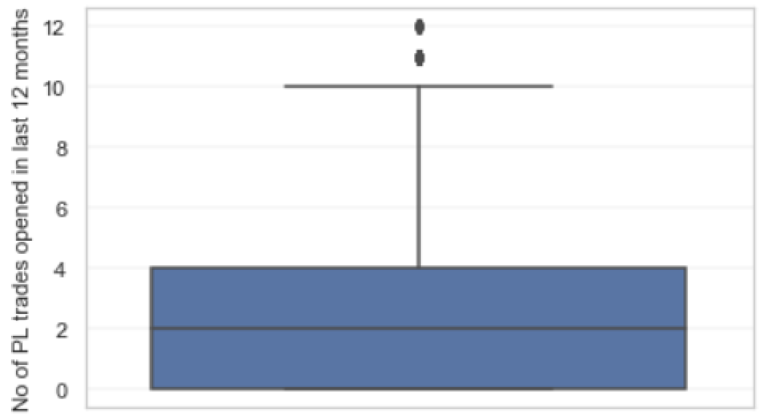
## Outlier Treatment Of Credit Bureau Data: (contd.)

The columns having outlier values are shown below, which are expected to have some outlier and removing or replacing them would manipulate the dataset. Hence normalizing them would be the best approach to solve the outlier treatment.





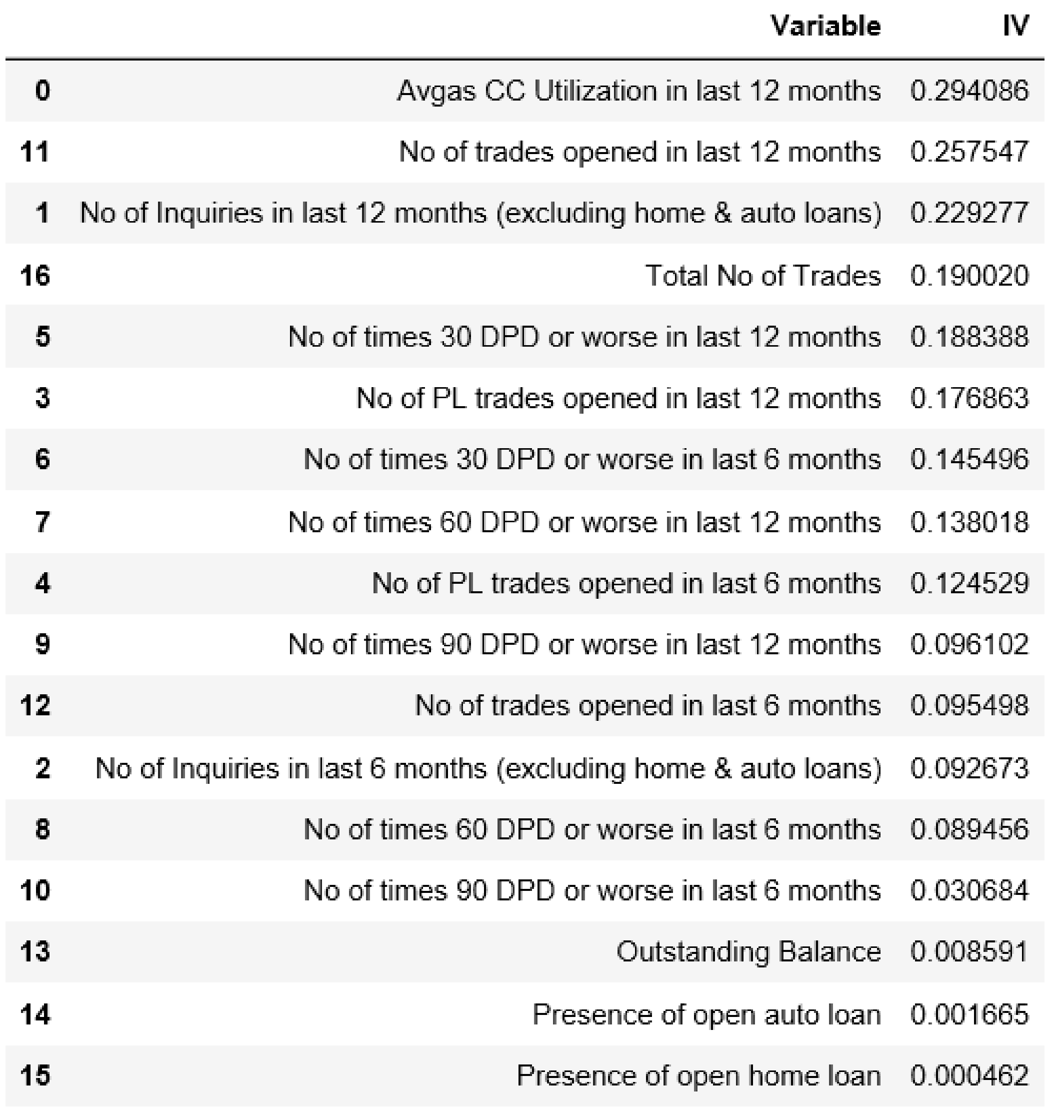




## WOE Analysis and Information Value

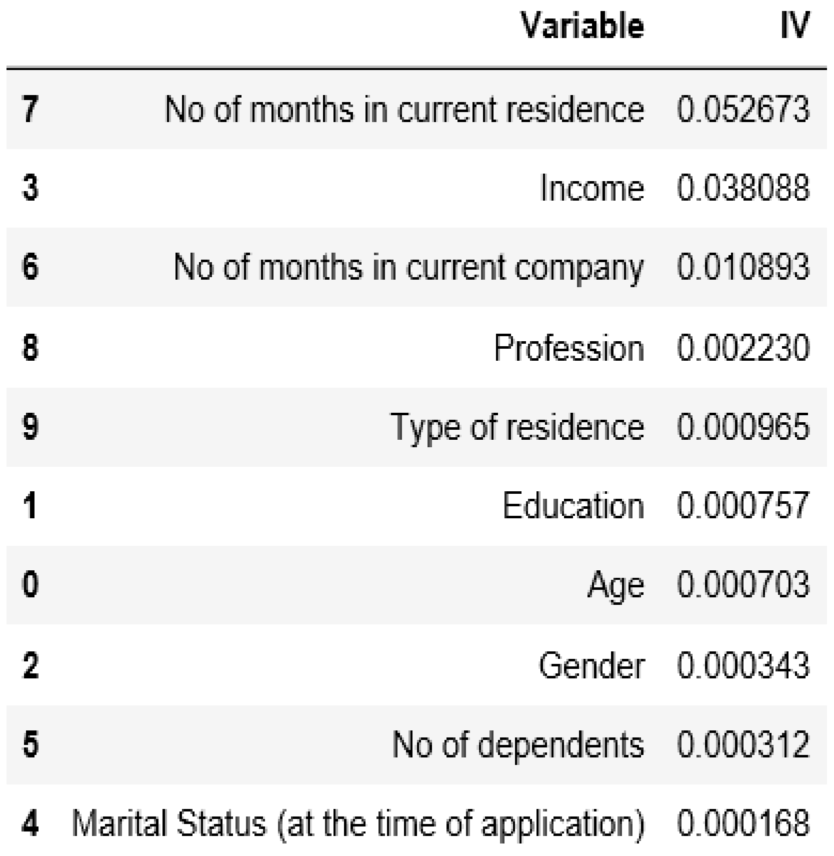
- WOE and IV values are calculated for each of the attributes using information and woe.binning package.
- Attributes for which WOE values that were made by default using the Information package in R and were not monotonically changing across bins, coarser bins were made by decreasing the number of bins until monotonic behavior is observed across bins .
- For 9 variables with Missing values, the variable values were replaced by their corresponding WOE values.
- From the IV values we can conclude that parameters in the demographic data don’t play much significant role in prediction and most of the significant variables are from Credit Bureau data.
- Top 12 Variables with IV value of 0.1 to 0.3 has medium predictive power and are considered significant. There is no variable with strong predictive power.

## Variables with IV Values:



| | Variable | IV |
|---|---|---|
| 0 | Avgas CC Utilization in last 12 months | 0.294086 |
| 11 | No of trades opened in last 12 months | 0.257547 |
| 1 | No of Inquiries in last 12 months (excluding home & auto loans) | 0.229277 |
| 16 | Total No of Trades | 0.190020 |
| 5 | No of times 30 DPD or worse in last 12 months | 0.188388 |
| 3 | No of PL trades opened in last 12 months | 0.176863 |
| 6 | No of times 30 DPD or worse in last 6 months | 0.145496 |
| 7 | No of times 60 DPD or worse in last 12 months | 0.138018 |
| 4 | No of PL trades opened in last 6 months | 0.124529 |
| 9 | No of times 90 DPD or worse in last 12 months | 0.096102 |
| 12 | No of trades opened in last 6 months | 0.095498 |
| 2 | No of Inquiries in last 6 months (excluding home & auto loans) | 0.092673 |
| 8 | No of times 60 DPD or worse in last 6 months | 0.089456 |
| 10 | No of times 90 DPD or worse in last 6 months | 0.030684 |
| 13 | Outstanding Balance | 0.008591 |
| 14 | Presence of open auto loan | 0.001665 |
| 15 | Presence of open home loan | 0.000462 |

***Credit Bureau Data***

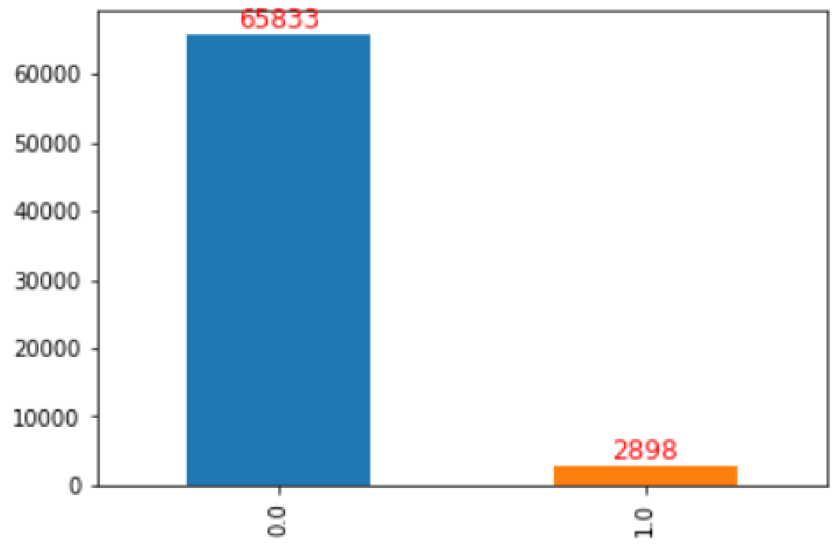


| | Variable | IV |
|---|---|---|
| 7 | No of months in current residence | 0.052673 |
| 3 | Income | 0.038088 |
| 6 | No of months in current company | 0.010893 |
| 8 | Profession | 0.002230 |
| 9 | Type of residence | 0.000965 |
| 1 | Education | 0.000757 |
| 0 | Age | 0.000703 |
| 2 | Gender | 0.000343 |
| 5 | No of dependents | 0.000312 |
| 4 | Marital Status (at the time of application) | 0.000168 |

***Demographic Data***

**<u>Exploratory Data Analysis On Performance Tag</u>**:

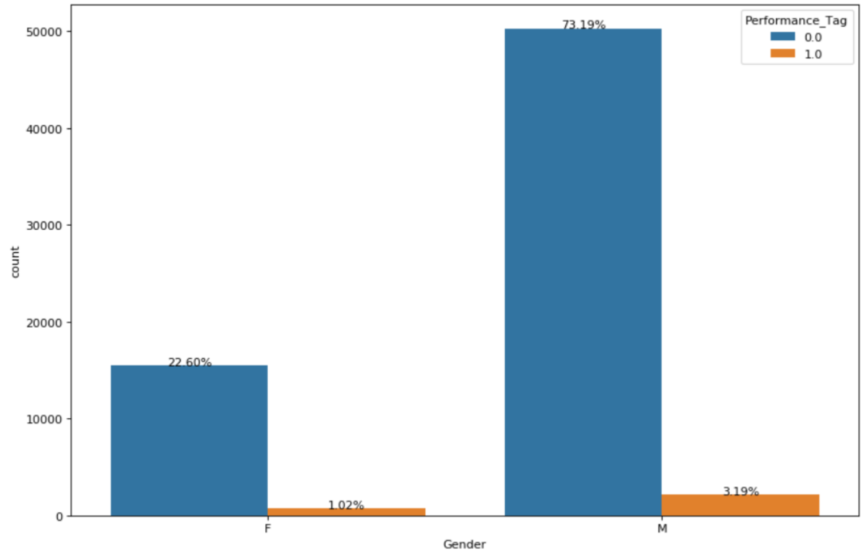
The variations in the distribution of performance tag is shown in the first figure. We can see that almost 3000 records are defaulters in our dataset having performance tag 1.



*Performance Tag*

**<u>Exploratory Data Analysis On Gender</u>**:

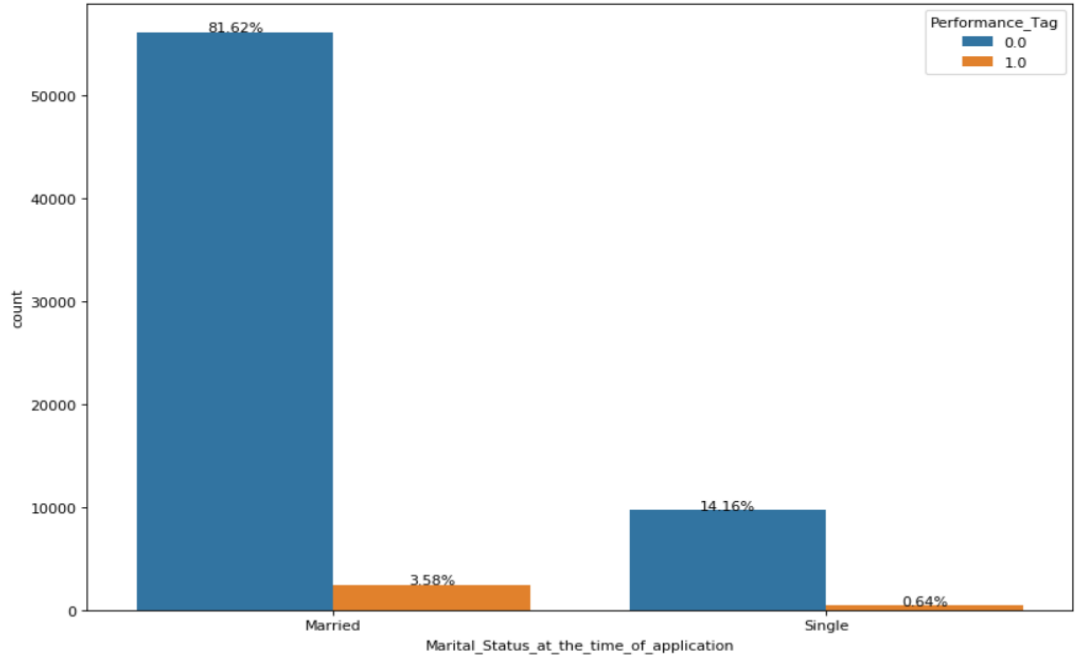
The distribution of performance tag over Gender is shown in the second figure. We can see that approx. 1% of the Female records and 3% of Male records are likely to default in our dataset.



*Gender*

**<u>Exploratory Data Analysis On Marital Status</u>**:

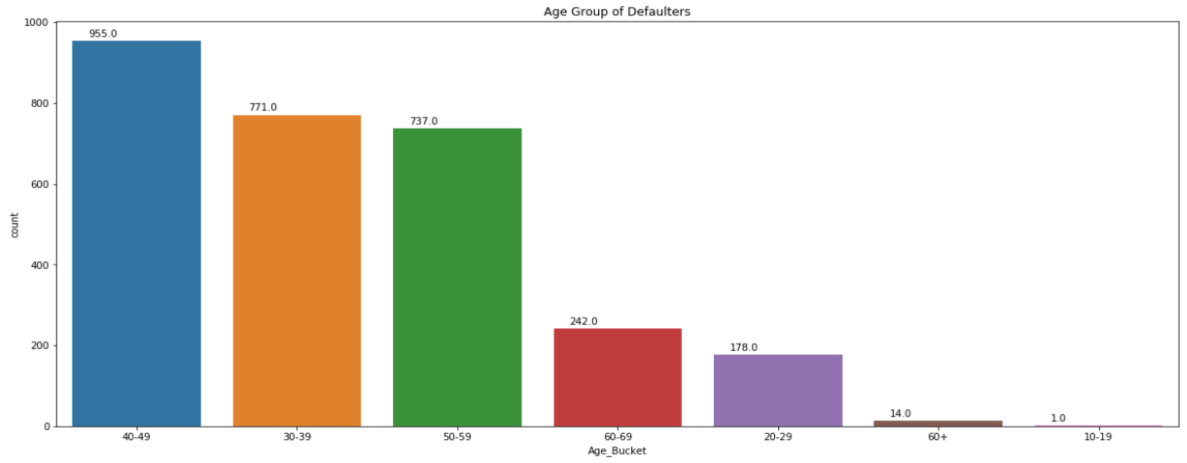
The variations of Marital Status in the distribution of performance tag is shown in the first figure. We can see that married people have a higher tendency of defaulting than single people.



*Marital Status At the time of Application*

**<u>Exploratory Data Analysis On Age Bins</u>**:

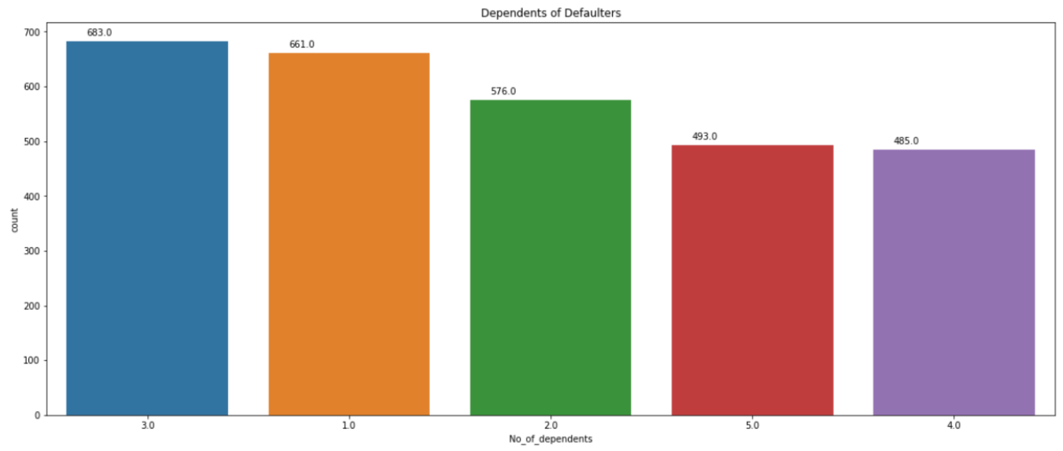
The variations of age in the distribution of performance tag is shown in the figure. We can observe that the age range of 40-49 have a much higher tendency of being a defaulter than others. 955 records in our dataset are defaulters who have age range of 40-49.



*Age Bucket*

**<u>Exploratory Data Analysis On No Of Dependents of Defaulters</u>**:

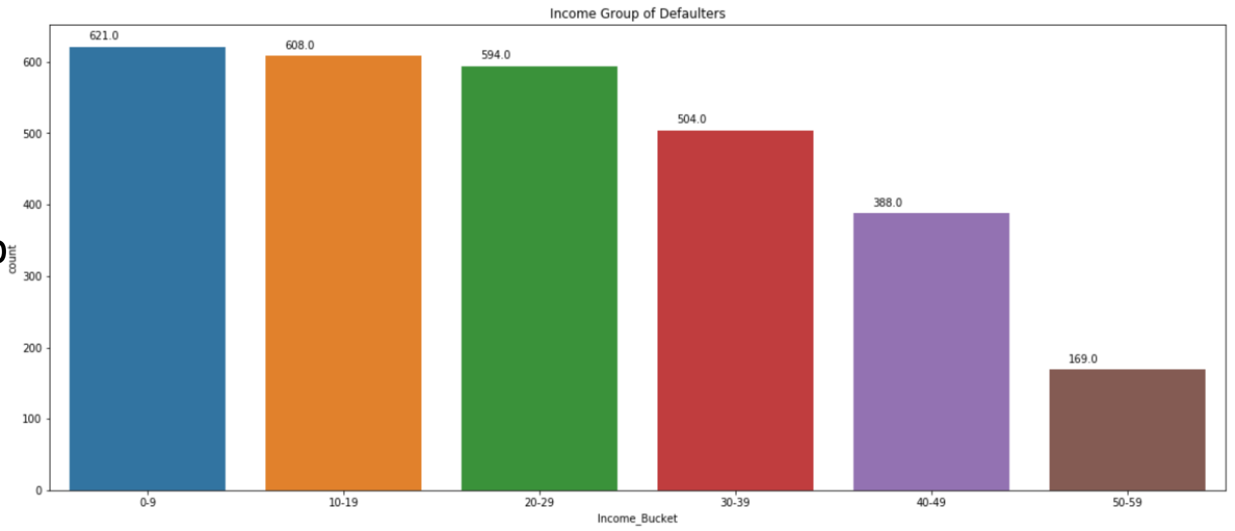
The frequency of No Of Dependents of records having Performance Tag as 1 is shown in the first figure. We can see that defaulters having 3 dependents are of maximum frequency followed by defaulters having 1 dependent.



*No Of Dependents*

**<u>Exploratory Data Analysis On Income</u>**:

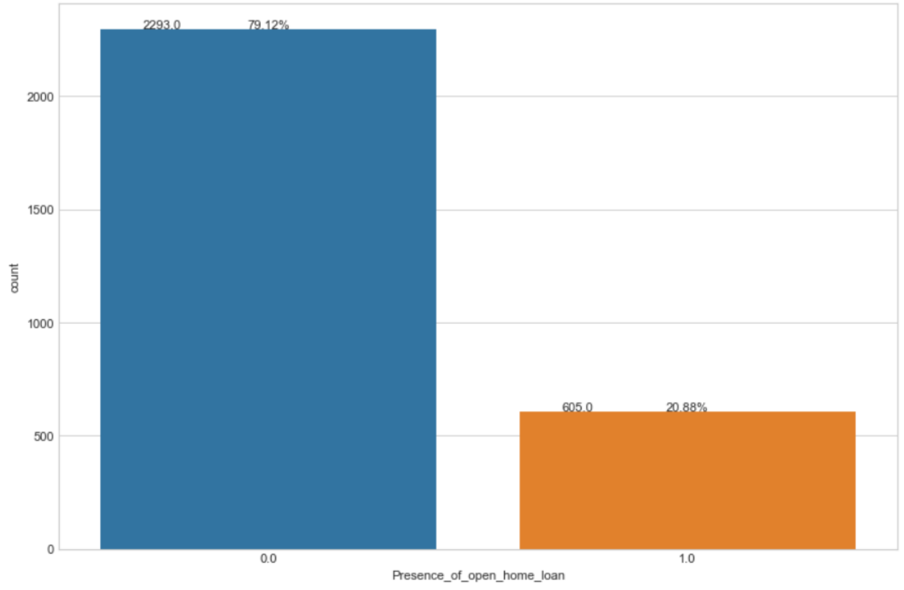
The variations in the distribution of Income group having the highest defaulters is shown in the figure. We can see that group having 0-9 range in income group have the highest tendency to default.



*Income Bucket*

## Exploratory Data Analysis On Open Home Loans :

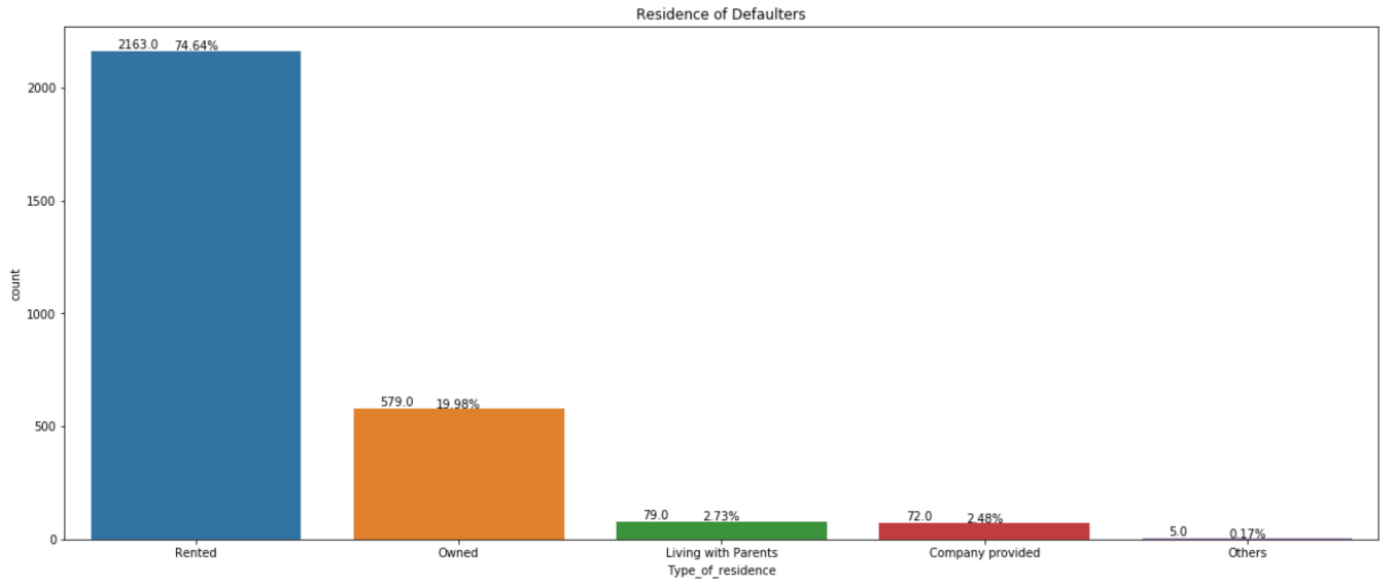
Around 80% of the defaulters does not have `Home Loan` which means the applicants taking home loan has less chances of being a defaulter.



*Open Home Loans*

## Exploratory Data Analysis On Types Of Residence:

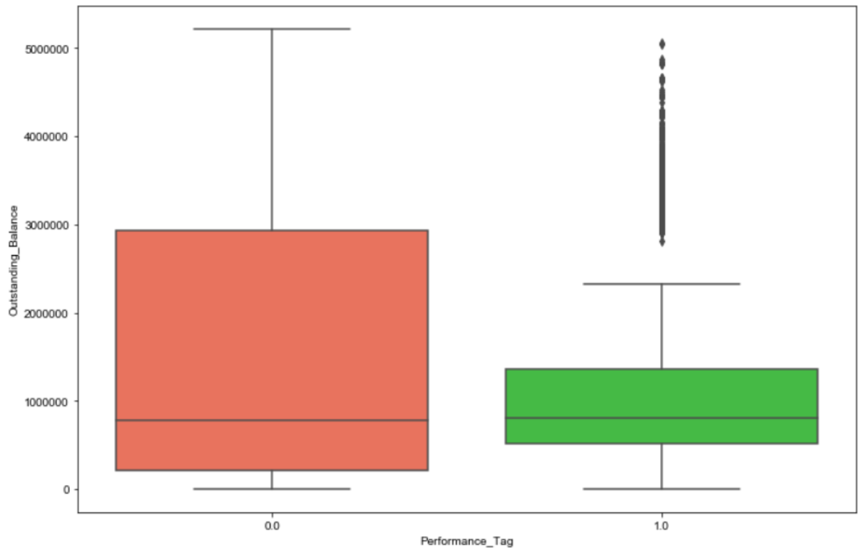
The variation of Type Of Residence is shown in figure. It is evident that around 77% of the defaulters stay in a rented place.



*Types of residence*

## Exploratory Data Analysis On Outstanding Balance:

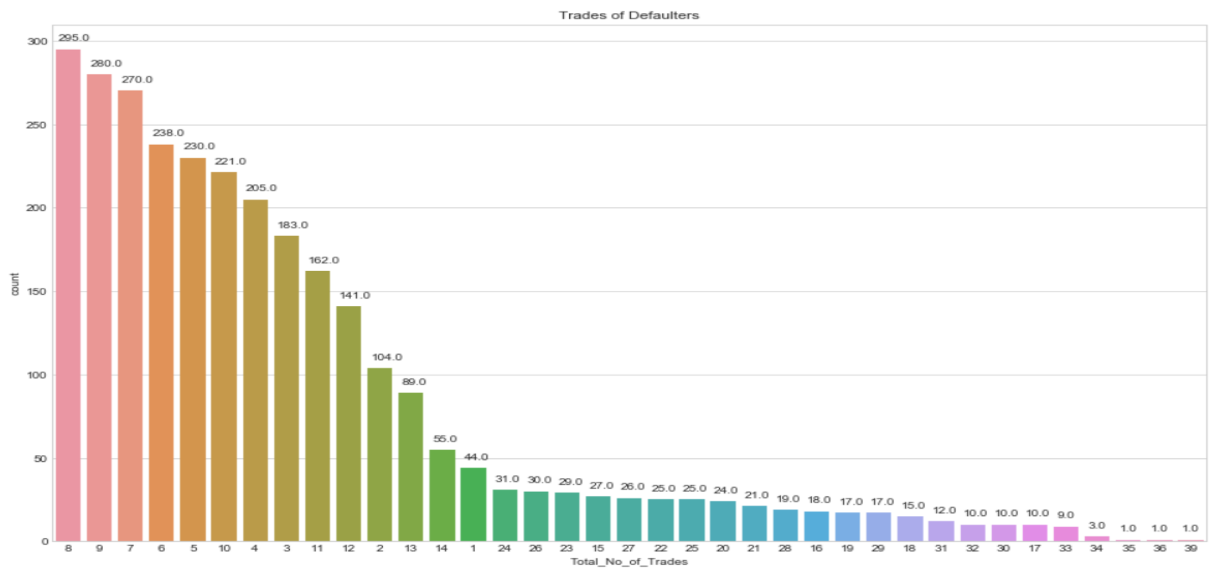
A huge outlier can be seen in the defaulters from the boxplot where the outstanding amount is in a very high range. From the Boxplot it is clear that higher the outstanding balance on the applicant, higher the chances of him being a defaulter.



*Outstanding Balance*

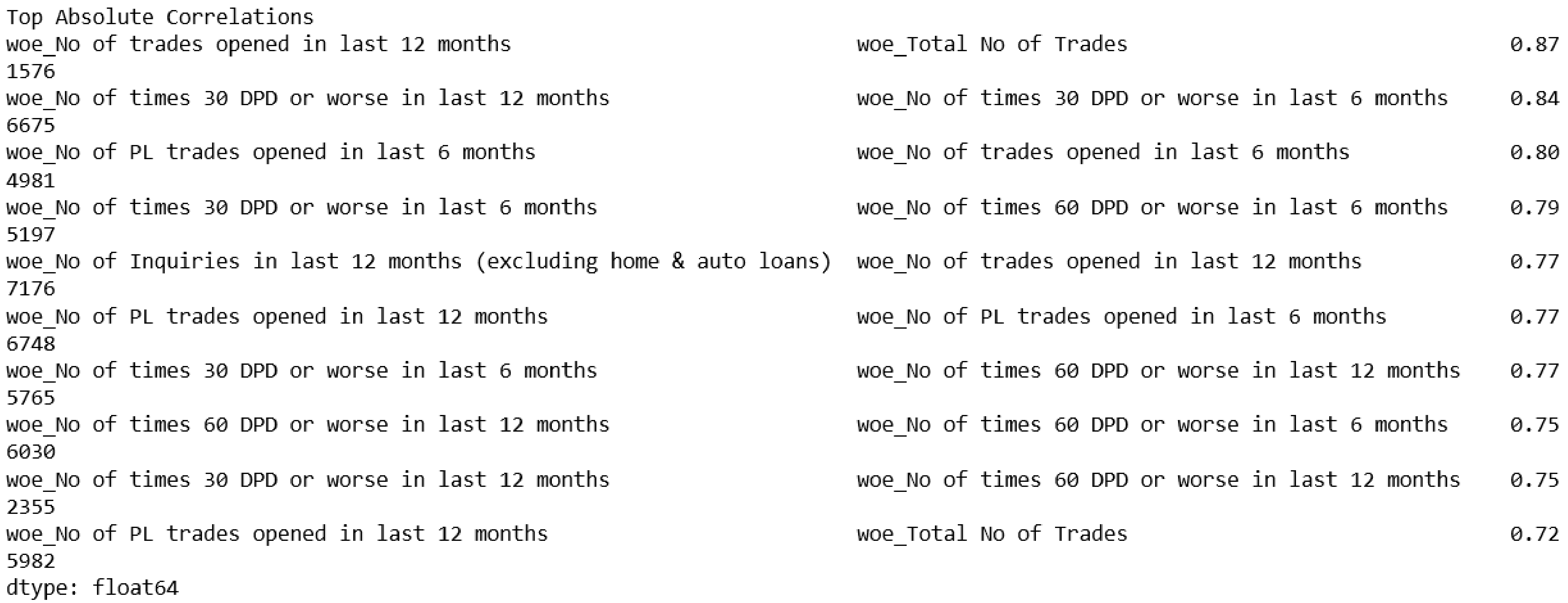
## Exploratory Data Analysis On No Of Trades:

From the plot, it is clear that the No of Trades done between 4-10 has more probability of getting a defaulter. As most of the defaulters have made a transaction of 4-10 trades.



*Total No Of Trades*

# Top Correlations Amongst Variables in Merged Dataset

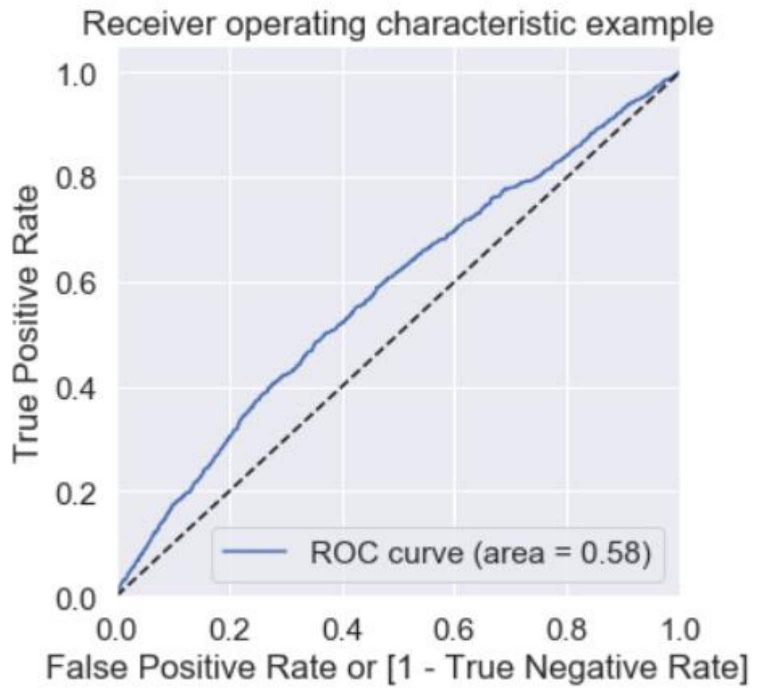


```
Top Absolute Correlations
woe_No of trades opened in last 12 months                            woe_Total No of Trades                               0.87
1576
woe_No of times 30 DPD or worse in last 12 months                    woe_No of times 30 DPD or worse in last 6 months     0.84
6675
woe_No of PL trades opened in last 6 months                          woe_No of trades opened in last 6 months             0.80
4981
woe_No of times 30 DPD or worse in last 6 months                     woe_No of times 60 DPD or worse in last 6 months     0.79
5197
woe_No of Inquiries in last 12 months (excluding home & auto loans)  woe_No of trades opened in last 12 months            0.77
7176
woe_No of PL trades opened in last 12 months                         woe_No of PL trades opened in last 6 months          0.77
6748
woe_No of times 30 DPD or worse in last 6 months                     woe_No of times 60 DPD or worse in last 12 months    0.77
5765
woe_No of times 60 DPD or worse in last 12 months                    woe_No of times 60 DPD or worse in last 6 months     0.75
6030
woe_No of times 30 DPD or worse in last 12 months                    woe_No of times 60 DPD or worse in last 12 months    0.75
2355
woe_No of PL trades opened in last 12 months                         woe_Total No of Trades                               0.72
5982
dtype: float64
```

## Model Building On Demo-Data: Logistic with WOE and Smote Analysis

The columns Age, Education, Gender and etc. as shown in the figure are found out to be important by Logistic Regression, with AUC 0.58 and accuracy 0.55.

```
{'woe_Age': 0.11812060405068744,
 'woe_Education': -1.035448583149849,
 'woe_Gender': -2.1862061074230175,
 'woe_Income': -0.9002814495404263,
 'woe_Marital Status (at the time of application)': -0.04188678769956458,
 'woe_No of dependents': -0.48913906236865784,
 'woe_No of months in current company': -0.9166622907598143,
 'woe_No of months in current residence': -0.8497732421434512,
 'woe_Profession ': -0.8828477219485825,
 'woe_Type of residence': -0.7318554087067599}
```

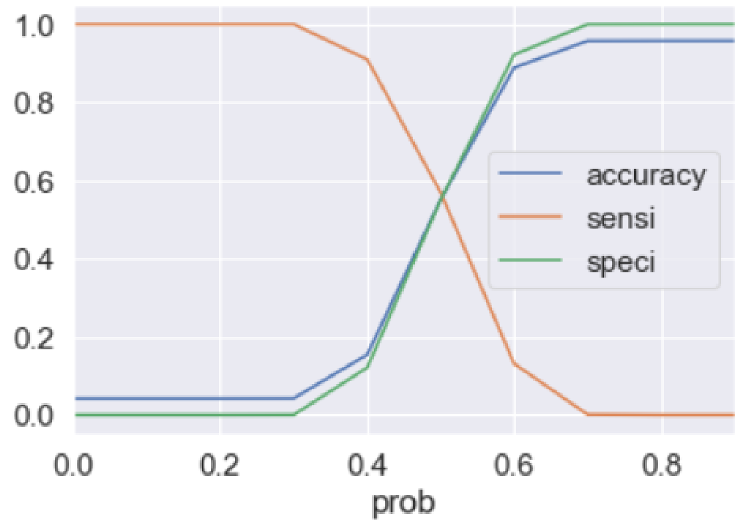


```
Accuracy:0.552
Sensitivity:0.567
Specificity:0.551
AUC:0.58
Ks_2sampResult(statistic=0.13184498117905452, pvalue=2.5632291794947974e-13)
```

# Model Building On Demo-Data: Logistic with WOE Using Balanced Class Weight and Grid Search CV

Logistic Regression with StratifiedKfold cross validation (with 5 folds) and grid search cv is applied on the demographic dataset with precision 0.97 for non defaulters.

```
GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=4, shuffle=True),
             error_score='raise-deprecating',
             estimator=Pipeline(memory=None,
                                steps=[('scaler',
                                        StandardScaler(copy=True,
                                                       with_mean=True,
                                                       with_std=True)),
                                       ('logistic',
                                        LogisticRegression(C=1.0,
                                                           class_weight='balanced',
                                                           dual=False,
                                                           fit_intercept=True,
                                                           intercept_scaling=1,
                                                           l1_ratio=None,
                                                           max_iter=100,
                                                           multi_class='warn',
                                                           n_jobs=None,
                                                           penalty='l2',
                                                           random_state=None,
                                                           solver='warn',
                                                           tol=0.0001,
                                                           verbose=0,
                                                           warm_start=False))],
                                verbose=False),
             iid='warn', n_jobs=-1,
             param_grid={'logistic__C': [0.1, 0.5, 1, 2, 3, 4, 5, 10],
                         'logistic__penalty': ['l1', 'l2']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='roc_auc', verbose=1)
```

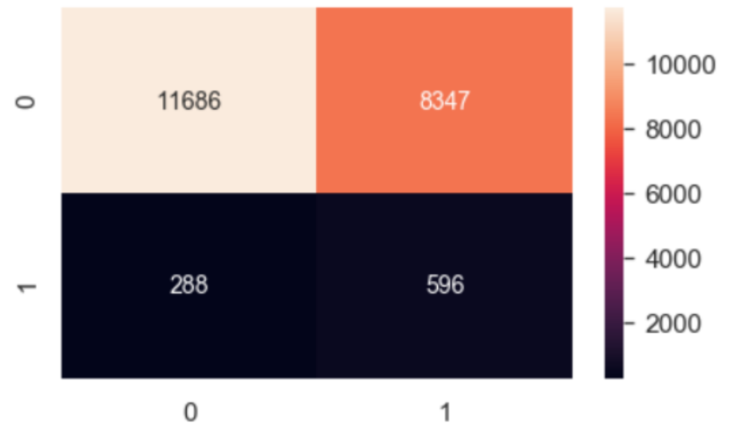


```
              precision    recall  f1-score   support

         0.0       0.97      0.55      0.70     20034
         1.0       0.05      0.57      0.10       884

    accuracy                           0.55     20918
   macro avg       0.51      0.56      0.40     20918
weighted avg       0.93      0.55      0.68     20918
```

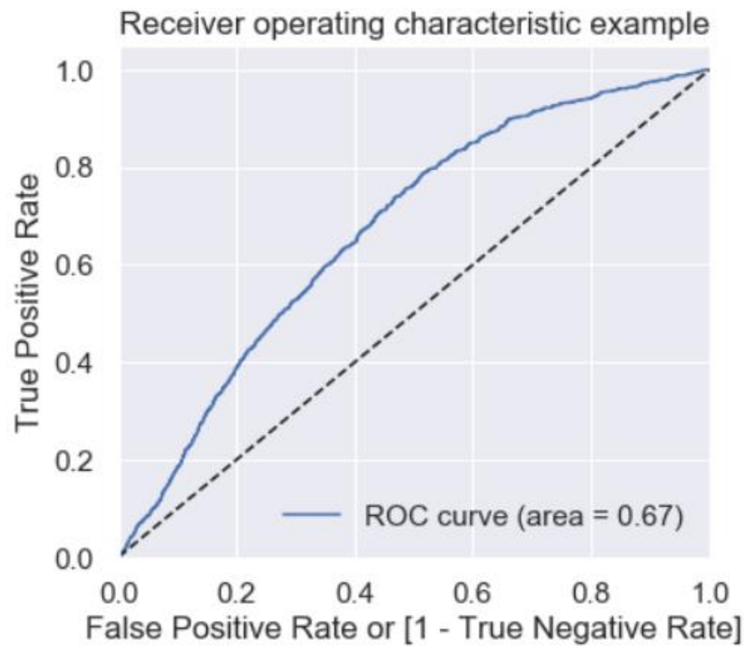
# Final Model Building: Logistic with Balanced Class Weight

Logistic Regression with balanced class weight is applied on the merged dataset to obtain AUC 0.67 and precision of 94% with recall 59%.

```
Pipeline(memory=None,
         steps=[('scaler',
                 StandardScaler(copy=True, with_mean=True, with_std=True)),
                ('logistic',
                 LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                                    fit_intercept=True, intercept_scaling=1,
                                    l1_ratio=None, max_iter=100,
                                    multi_class='warn', n_jobs=None,
                                    penalty='l2', random_state=None,
                                    solver='warn', tol=0.0001, verbose=0,
                                    warm_start=False))],
         verbose=False)
```



```
              precision    recall  f1-score   support

         0.0       0.98      0.58      0.73     20033
         1.0       0.07      0.67      0.12       884

    accuracy                           0.59     20917
   macro avg       0.52      0.63      0.43     20917
weighted avg       0.94      0.59      0.70     20917
```

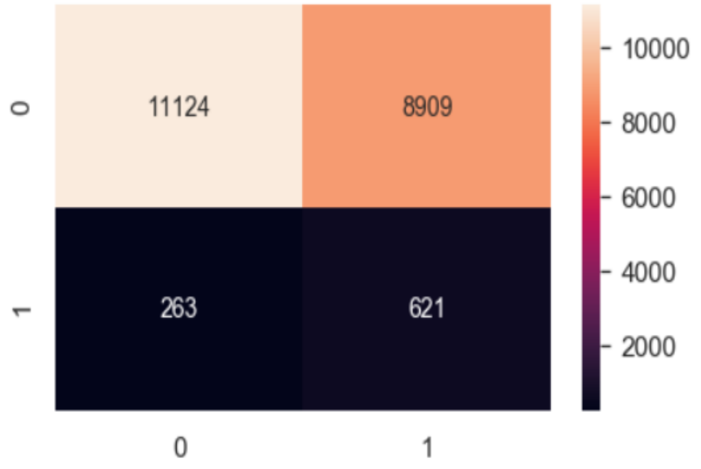


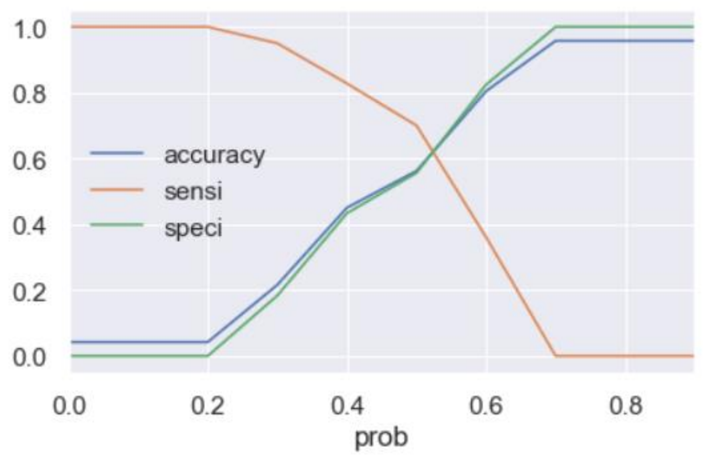
# Hyper Parameter Tuning on Logistic Regression:

After hyper tuning the parameter of the logistic regression using StratifiedKFold of 5 splits and implementing L1 regression for feature selection, we have achieved 70% sensitivity with 56% accuracy with the final dataset representing seven columns to be actually effective.

| | | |
|---|---|---|
| **18** | woe_No_of_times_90_DPD_or_worse_in_last_12_months | -0.038512 |
| **16** | woe_No_of_times_60_DPD_or_worse_in_last_12_months | -0.042447 |
| **14** | woe_No_of_PL_trades_opened_in_last_6_months | -0.042517 |
| **13** | woe_No_of_PL_trades_opened_in_last_12_months | -0.191696 |
| **11** | woe_No_of_Inquiries_in_last_12_months_excludin... | -0.390215 |
| **15** | woe_No_of_times_30_DPD_or_worse_in_last_12_months | -0.391697 |
| **10** | woe_Avgas_CC_Utilization_in_last_12_months | -0.525953 |

```
GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=4, shuffle=True),
             error_score='raise-deprecating',
             estimator=Pipeline(memory=None,
                                steps=[('scaler',
                                        StandardScaler(copy=True,
                                                       with_mean=True,
                                                       with_std=True)),
                                       ('logistic',
                                        LogisticRegression(C=1.0,
                                                           class_weight='balanced',
                                                           dual=False,
                                                           fit_intercept=True,
                                                           intercept_scaling=1,
                                                           l1_ratio=None,
                                                           max_iter=100,
                                                           multi_class='warn',
                                                           n_jobs=None,
                                                           penalty='l2',
                                                           random_state=None,
                                                           solver='warn',
                                                           tol=0.0001,
                                                           verbose=0,
                                                           warm_start=False))],
                                verbose=False),
             iid='warn', n_jobs=-1,
             param_grid={'logistic__C': [0.005, 0.008, 0.01, 0.03, 0.05, 0.1,
                                         0.5, 1],
                         'logistic__penalty': ['l1', 'l2']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='roc_auc', verbose=1)
```

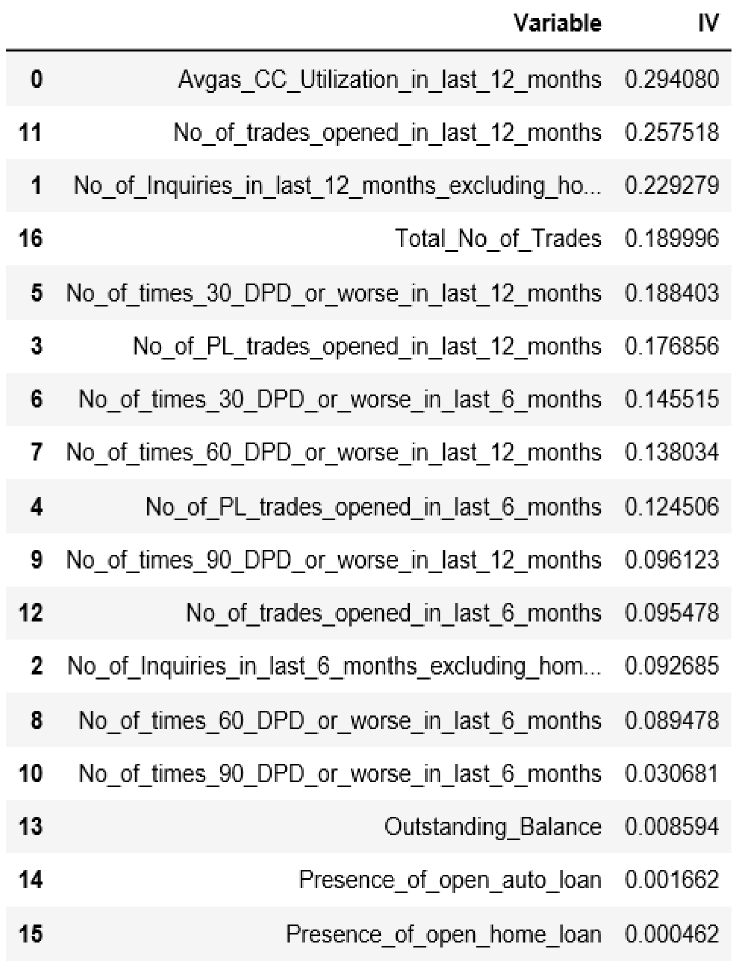




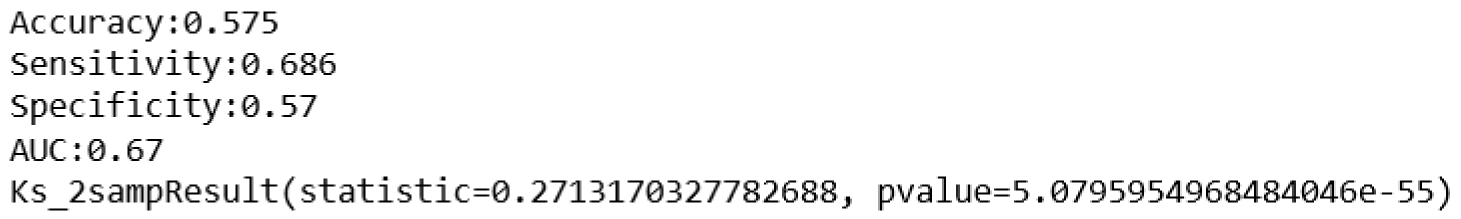
```
              precision    recall  f1-score   support

         0.0       0.98      0.56      0.71     20033
         1.0       0.07      0.70      0.12       884

    accuracy                           0.56     20917
   macro avg       0.52      0.63      0.41     20917
weighted avg       0.94      0.56      0.68     20917
```

# Logistic Regression on IV Selected Values

The logistic regression with stratifiedKFold is applied on the IV value selected seven columns. The model is found to be working accurately on this dataset, which proves our model building is on the right track.



| | Variable | IV |
|---|---|---|
| 0 | Avgas_CC_Utilization_in_last_12_months | 0.294080 |
| 11 | No_of_trades_opened_in_last_12_months | 0.257518 |
| 1 | No_of_Inquiries_in_last_12_months_excluding_ho... | 0.229279 |
| 16 | Total_No_of_Trades | 0.189996 |
| 5 | No_of_times_30_DPD_or_worse_in_last_12_months | 0.188403 |
| 3 | No_of_PL_trades_opened_in_last_12_months | 0.176856 |
| 6 | No_of_times_30_DPD_or_worse_in_last_6_months | 0.145515 |
| 7 | No_of_times_60_DPD_or_worse_in_last_12_months | 0.138034 |
| 4 | No_of_PL_trades_opened_in_last_6_months | 0.124506 |
| 9 | No_of_times_90_DPD_or_worse_in_last_12_months | 0.096123 |
| 12 | No_of_trades_opened_in_last_6_months | 0.095478 |
| 2 | No_of_Inquiries_in_last_6_months_excluding_hom... | 0.092685 |
| 8 | No_of_times_60_DPD_or_worse_in_last_6_months | 0.089478 |
| 10 | No_of_times_90_DPD_or_worse_in_last_6_months | 0.030681 |
| 13 | Outstanding_Balance | 0.008594 |
| 14 | Presence_of_open_auto_loan | 0.001662 |
| 15 | Presence_of_open_home_loan | 0.000462 |

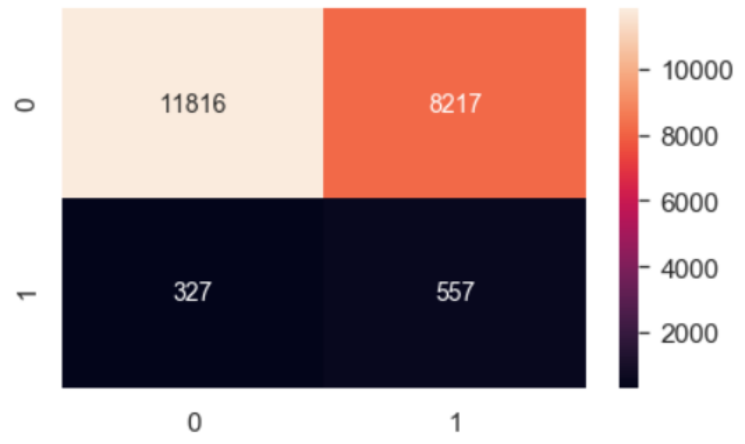


```
Accuracy:0.575
Sensitivity:0.686
Specificity:0.57
AUC:0.67
Ks_2sampResult(statistic=0.2713170327782688, pvalue=5.0795954968484046e-55)
```

# Random Forest Classifier on Woe-Merged Data

The Random Forest Classifier is applied on the dataset which also achieved similar metrics.

```
Pipeline(memory=None,
         steps=[('scaler',
                 StandardScaler(copy=True, with_mean=True, with_std=True)),
                ('random_forest',
                 RandomForestClassifier(bootstrap=True, class_weight='balanced',
                                        criterion='gini', max_depth=2,
                                        max_features='auto',
                                        max_leaf_nodes=None,
                                        min_impurity_decrease=0.0,
                                        min_impurity_split=None,
                                        min_samples_leaf=1, min_samples_split=2,
                                        min_weight_fraction_leaf=0.0,
                                        n_estimators=10, n_jobs=None,
                                        oob_score=False, random_state=None,
                                        verbose=0, warm_start=False))],
         verbose=False)
```



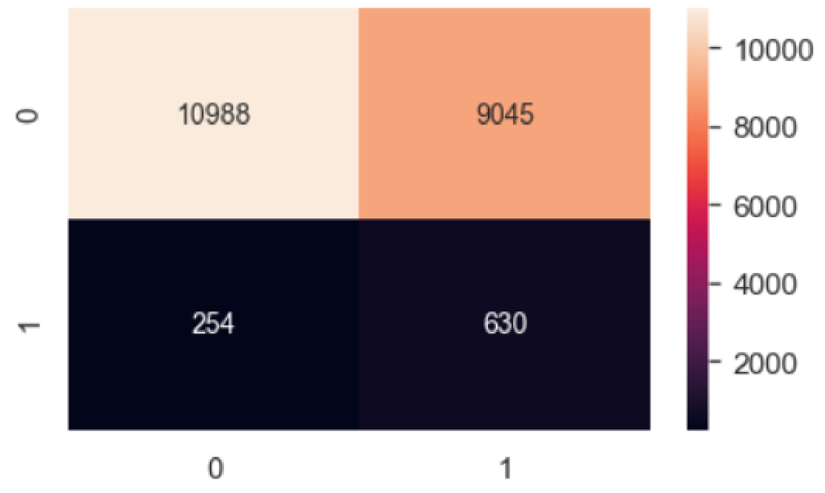
```
Accuracy:0.592
Sensitivity:0.63
Specificity:0.59
AUC:0.66
Ks_2sampResult(statistic=0.25205543206650205, pvalue=1.5220175634977577e-47)
```

# XGBoost Classifier on Woe-Merged Data

The XGBoost Classifier is also applied on the dataset, but logistic regression seems to be the most accurate model built on the dataset.

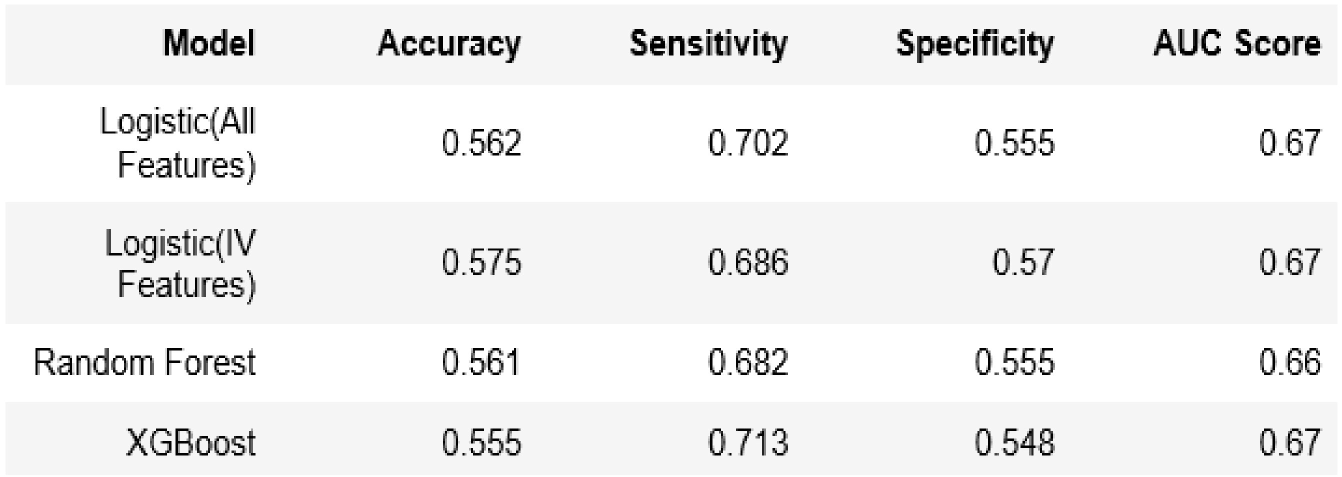
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=2,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=22.680252304706453,
              seed=None, silent=None, subsample=1, verbosity=1)
```

```
Accuracy:0.555
Sensitivity:0.713
Specificity:0.548
AUC:0.67
Ks_2sampResult(statistic=0.2762625491468489, pvalue=4.9818125474692054e-57)
```



## Model Evaluation:

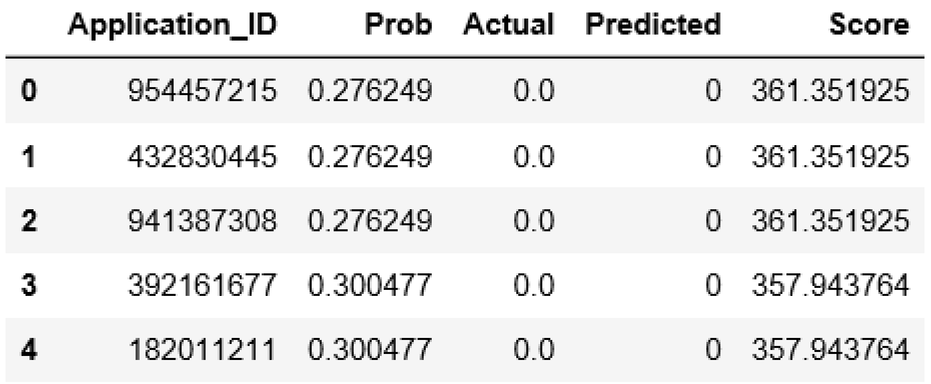
We built different models on the woe merged dataset and compared the performance. As the dataset is highly imbalanced we used class_weight feature of models to handle the imbalance. Considering the business scenario, where we need to identify the defaulters more strictly we need to reduce the false negatives, so we chose Sensitivity as the choice of metric.



| Model | Accuracy | Sensitivity | Specificity | AUC Score |
|---|---|---|---|---|
| Logistic(All Features) | 0.562 | 0.702 | 0.555 | 0.67 |
| Logistic(IV Features) | 0.575 | 0.686 | 0.57 | 0.67 |
| Random Forest | 0.561 | 0.682 | 0.555 | 0.66 |
| XGBoost | 0.555 | 0.713 | 0.548 | 0.67 |

## Application Score Card:

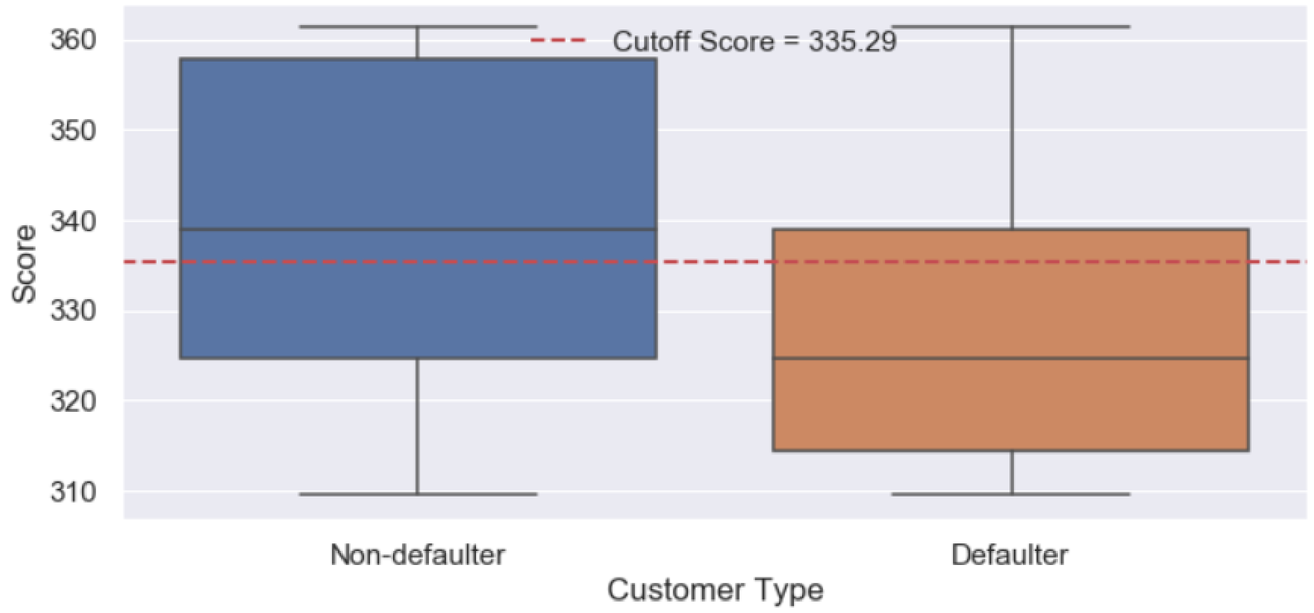
- Final application scorecard was made using the Logistic regression model on the entire dataset which also contained predictions for missing values in “Performance Tag” in 1425 records.
- The logistic regression model was chosen since its evaluation metrics were comparable to other models as well it’s an easily interpretable simple model.
- The scorecard was made using the following steps:
  1. Application score card was made with odds of 10 to 1 being a score of 400. Score increases by 20 points for doubling odds.
  2. Probability of default for all applicants were calculated
  3. Odds for good was calculated. Since the probability computed is for rejection (bad customers), Odd(good) = (1-P(bad))/P(bad)
  4. ln(odd(good)) was calculated 5. Used the following formula for computing application score card: 400 + slope * (ln(odd(good)) - ln(10)) where slope is 20/(ln(20)-ln(10)) Where, slope=20/(log(20)-log(10))
- Summary of application_score_card values:
  - Scores range from 309.47 to 361.35 for applicants with median score being 338.89.
  - Higher scores indicate less risk for defaulting



| | Application_ID | Prob | Actual | Predicted | Score |
|---|---|---|---|---|---|
| **0** | 954457215 | 0.276249 | 0.0 | 0 | 361.351925 |
| **1** | 432830445 | 0.276249 | 0.0 | 0 | 361.351925 |
| **2** | 941387308 | 0.276249 | 0.0 | 0 | 361.351925 |
| **3** | 392161677 | 0.300477 | 0.0 | 0 | 357.943764 |
| **4** | 182011211 | 0.300477 | 0.0 | 0 | 357.943764 |

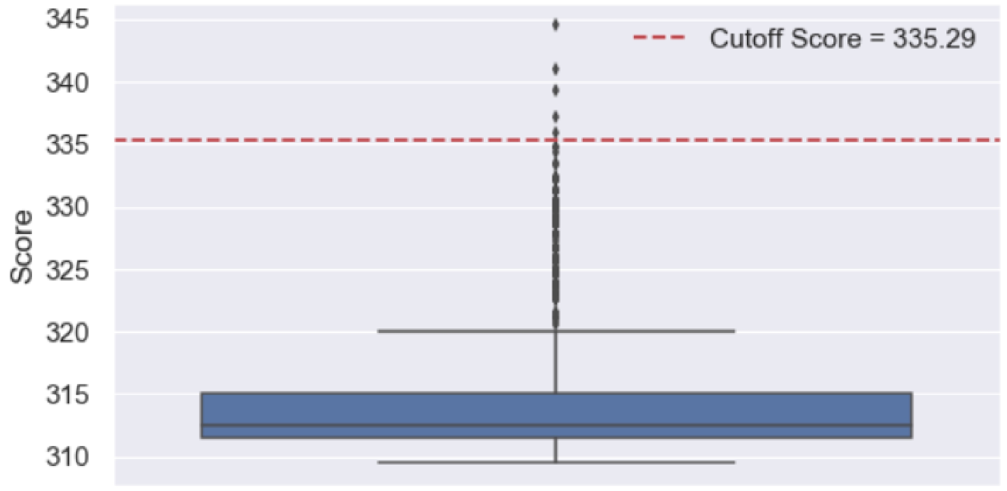
## Cut Off Chosen for Application Scorecard

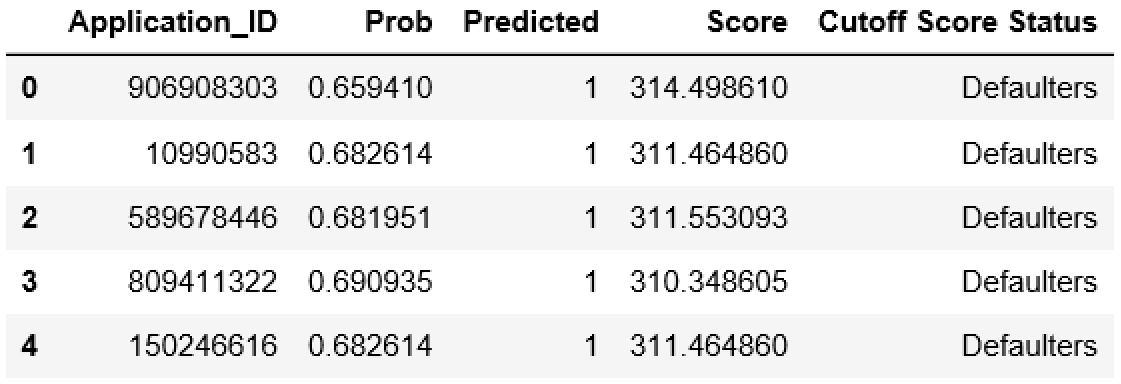
- Cutoff selected for  probability of default for logistic regression model was 0.485
- CUTOFF_SCORE= 400 + (slope * (log((1-0.46)/0.485) - log(10)))
- CUTOFF SCORE is  equal to 335.29
- We could observe that the mean and median score for the approved customers are higher than those of rejected customers. The red dotted line in the boxplot indicates the cutoff score we have chosen.



## Application Scorecard on Rejected Candidates

The scorecard calculation was applied on 1425 rejected candidates where it is found that with the same cut off of 335.29, 1420 candidates are likely to be defaulters and 5 candidates among them are likely to be non defaulters.

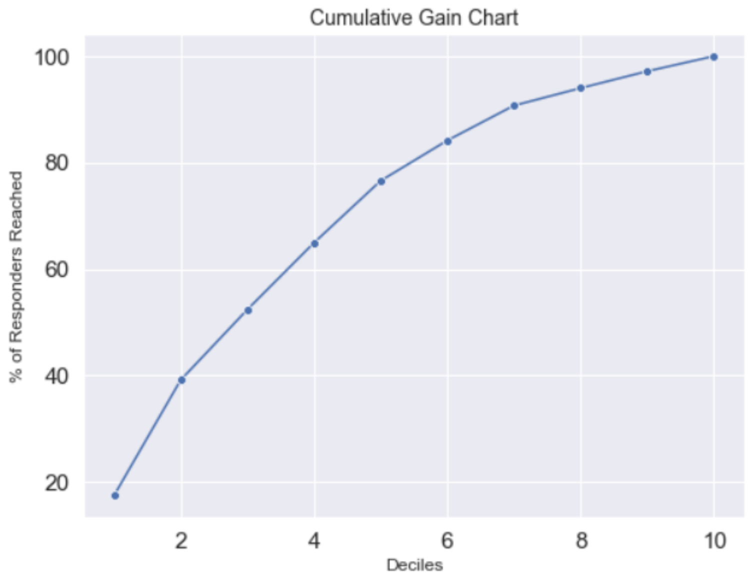


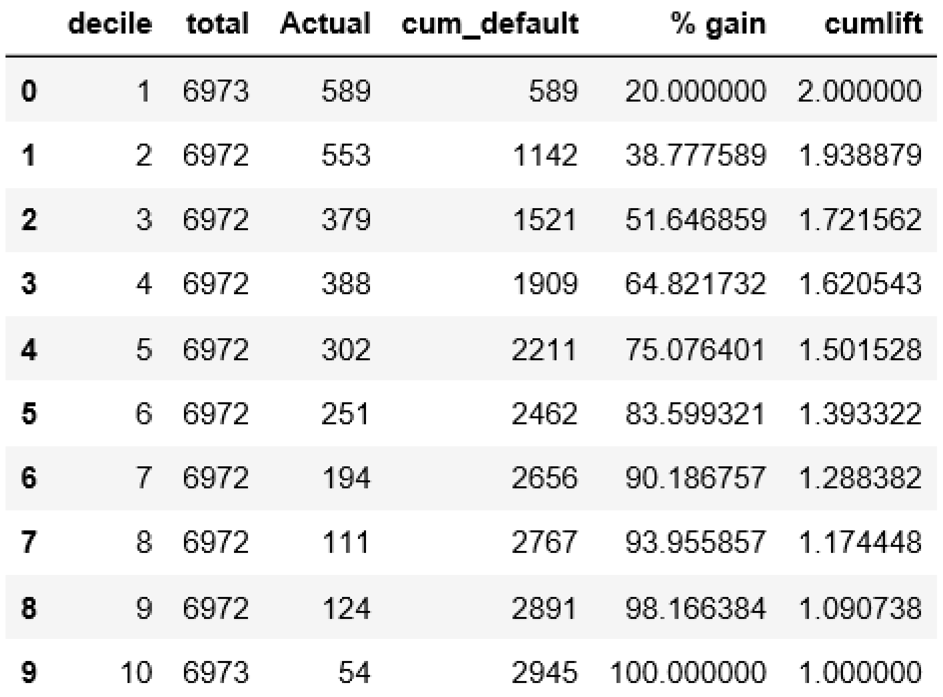


| | Application_ID | Prob | Predicted | Score | Cutoff Score Status |
|---|---|---|---|---|---|
| **0** | 906908303 | 0.659410 | 1 | 314.498610 | Defaulters |
| **1** | 10990583 | 0.682614 | 1 | 311.464860 | Defaulters |
| **2** | 589678446 | 0.681951 | 1 | 311.553093 | Defaulters |
| **3** | 809411322 | 0.690935 | 1 | 310.348605 | Defaulters |
| **4** | 150246616 | 0.682614 | 1 | 311.464860 | Defaulters |

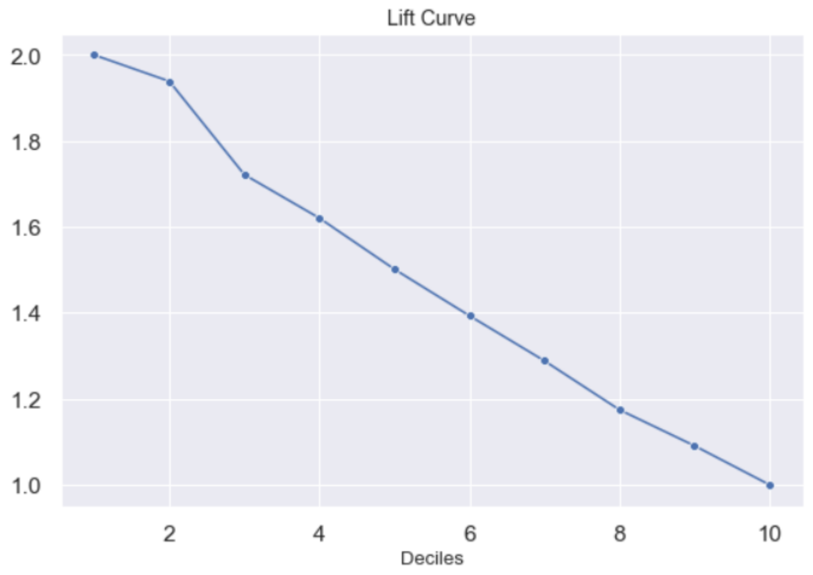
## Risk Analytics Metrics

We have successfully built a logistic regression model with a cut off at 0.5. From the decile table, we could see that we can predict 76% of total defaulters correctly by analyzing only **50%** of the total client base.



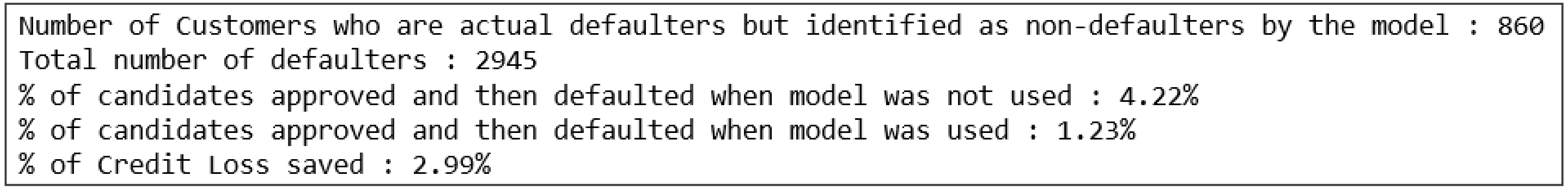


| | decile | total | Actual | cum_default | % gain | cumlift |
|---|---|---|---|---|---|---|
| **0** | 1 | 6973 | 589 | 589 | 20.000000 | 2.000000 |
| **1** | 2 | 6972 | 553 | 1142 | 38.777589 | 1.938879 |
| **2** | 3 | 6972 | 379 | 1521 | 51.646859 | 1.721562 |
| **3** | 4 | 6972 | 388 | 1909 | 64.821732 | 1.620543 |
| **4** | 5 | 6972 | 302 | 2211 | 75.076401 | 1.501528 |
| **5** | 6 | 6972 | 251 | 2462 | 83.599321 | 1.393322 |
| **6** | 7 | 6972 | 194 | 2656 | 90.186757 | 1.288382 |
| **7** | 8 | 6972 | 111 | 2767 | 93.955857 | 1.174448 |
| **8** | 9 | 6972 | 124 | 2891 | 98.166384 | 1.090738 |
| **9** | 10 | 6973 | 54 | 2945 | 100.000000 | 1.000000 |



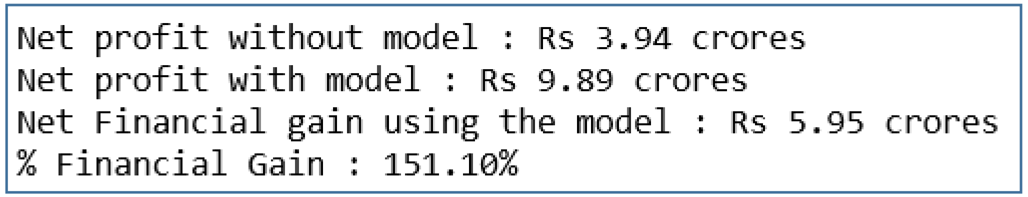
## Financial Benefits of the Model

- We will make some assumptions regarding the average credit loss for each defaulted customers and the profit obtained from each non-defaulters. We will analyze the overall financial benefit of the model and calculate the net financial gain obtained by using the model.
- Let's assume the average credit loss for each defaulted customer is Rs. 100000/- and profit for each non-defaulters be Rs. 5000/-.
- The Confusion Matrix for calculating the Financial gain using our model was made on the dataset without missing Performance tag records, since we need to evaluate how much gain was achieved using our model for applicants who were provided with credit card compared to when no model was used.
- The candidates who have been selected by the bank and have defaulted are responsible for the credit loss to the bank. We will calculate the percentage of credit loss we were able to avert by using the model.
- Revenue Loss occurs when good customers are identified as bad and credit card application is rejected.
- Both the counts of defaulters for credit loss and revenue loss are shown below:



```
Number of Customers who are actual defaulters but identified as non-defaulters by the model : 860
Total number of defaulters : 2945
% of candidates approved and then defaulted when model was not used : 4.22%
% of candidates approved and then defaulted when model was used : 1.23%
% of Credit Loss saved : 2.99%
```

*Credit Loss*



```
Net profit without model : Rs 3.94 crores
Net profit with model : Rs 9.89 crores
Net Financial gain using the model : Rs 5.95 crores
% Financial Gain : 151.10%
```

*Revenue Loss*

## Conclusion

- Logistic regression model is chosen as the final Model with 70% of Sensitivity.
- Optimal score cut-off value of 335.29 is derived to approve and reject the applications.
- By this we found out that credit loss % was decreased when we used this model. Hence it is accurate in rejecting the candidate who may default in future.
- There is Net Financial gain of Rs. 5.95 crore, that is 151.10% after using the model.
- The total number of non defaulters found out is 66777 and that of defaulters is 2945.

*Thank You!*