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 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.**

```
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
```

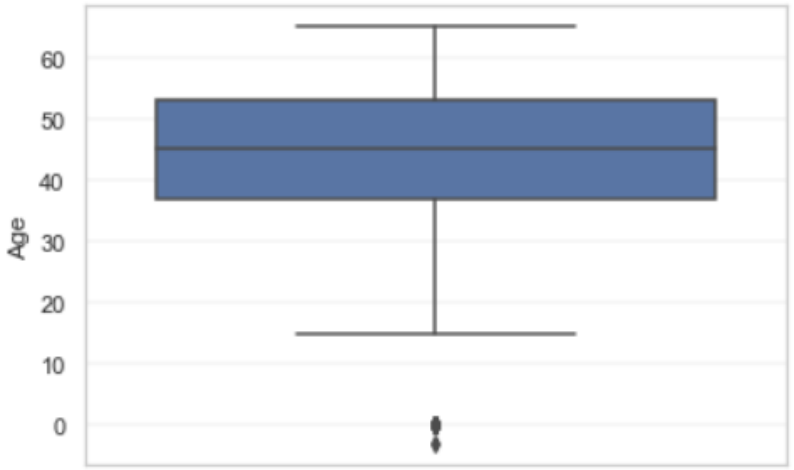
Data Cleansing of Credit Bureau Data:

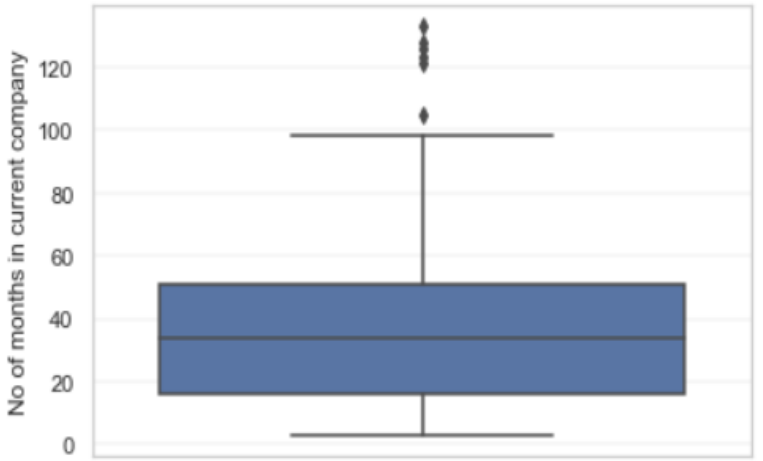
- 1425 rows out of 71295 rows in *Performance Tag* column have null values which indicates that the applicants are not given credit card. Hence these records are deleted.
- 6 different records having 3 duplicate *Application IDs* (653287861, 765011468 and 671989187) are found in the dataset. Hence 3 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:

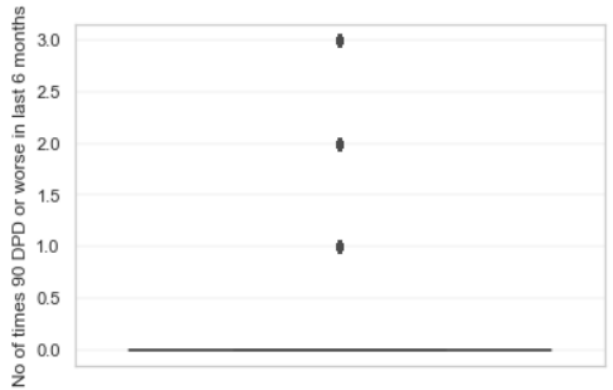
- The Age column has some outlier values which are negative and are already handled in the data cleansing part.
- The number of months in current company column has some outlier values above 75 %tile, which are treated by normalizing the values using standardization before feeding into the prediction 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.

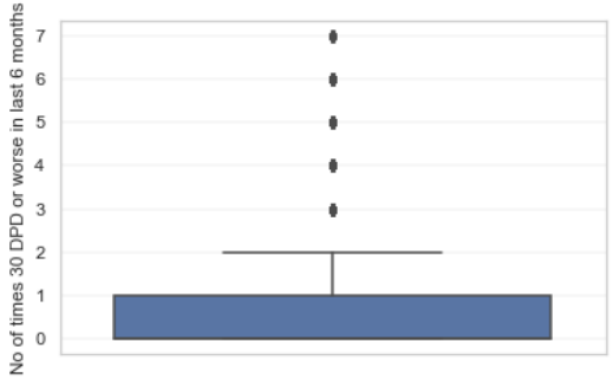


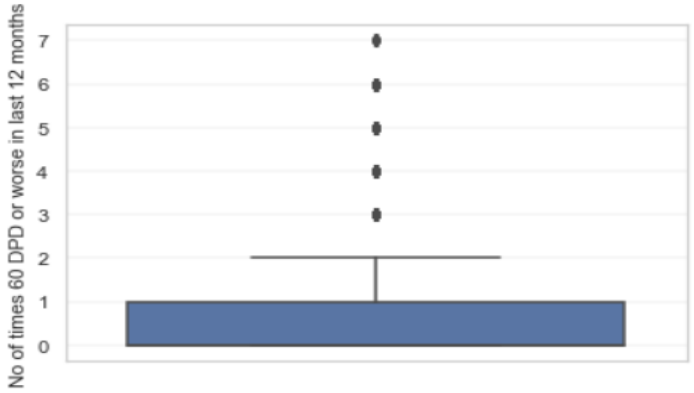


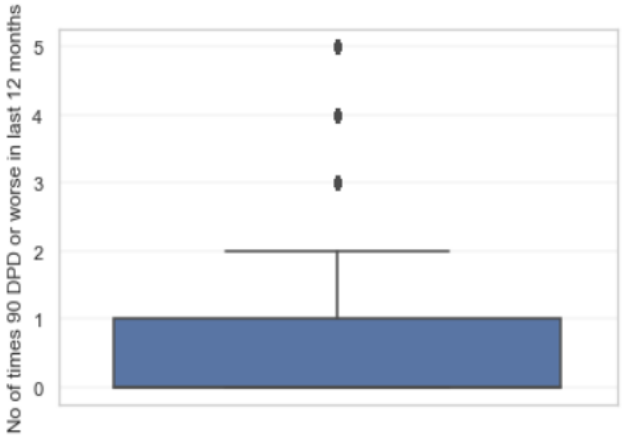
Outlier Treatment Of Credit Bureau Data:

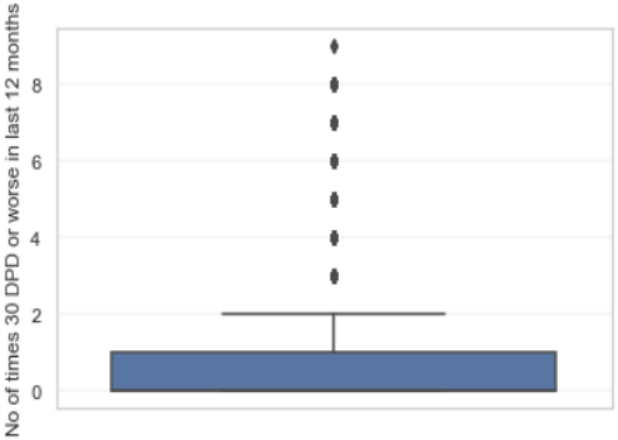
The columns having outlier values are shown below, which are expected to have some outlier and removing or replacing them would manipulate the dataset. Hence normalizing them or capping them to the nearest non outlier value would be the best approach to solve the outlier treatment.

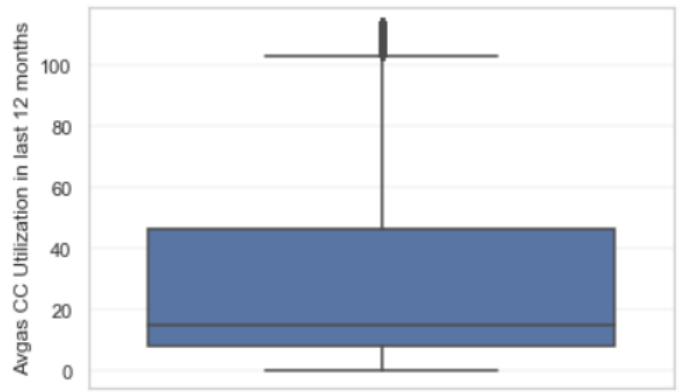






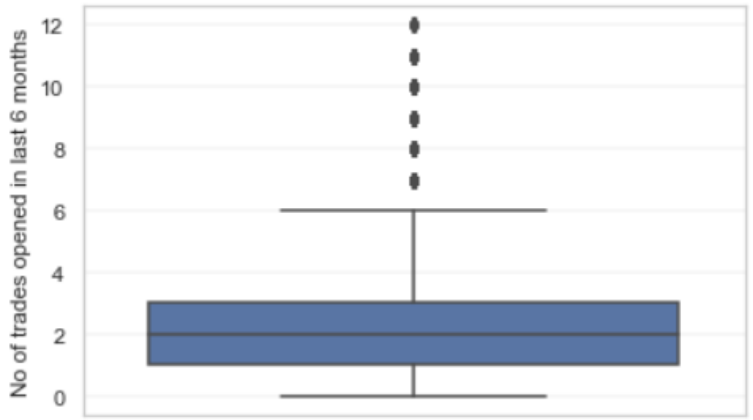


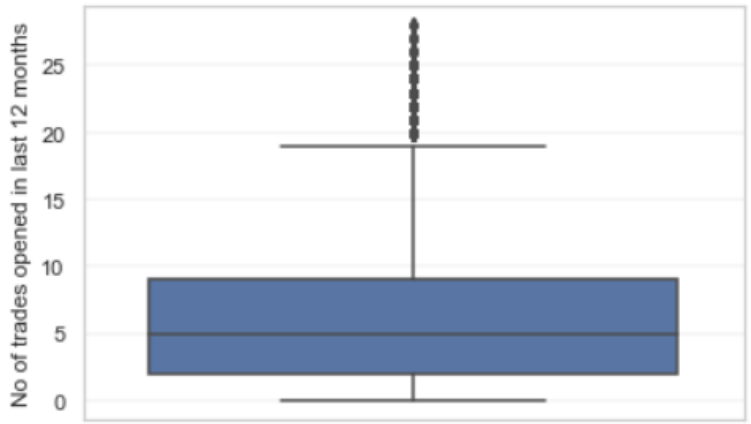


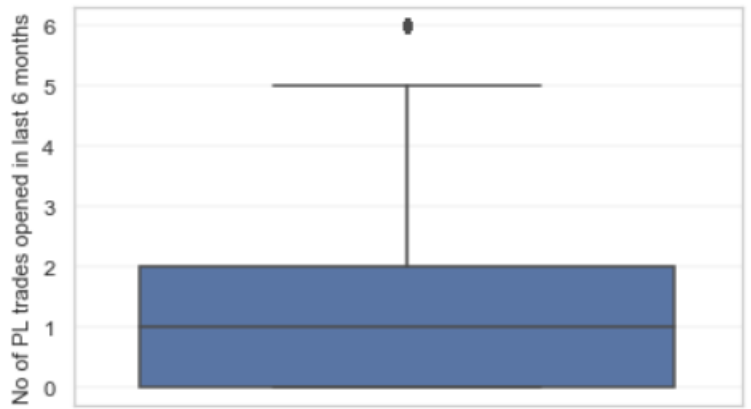


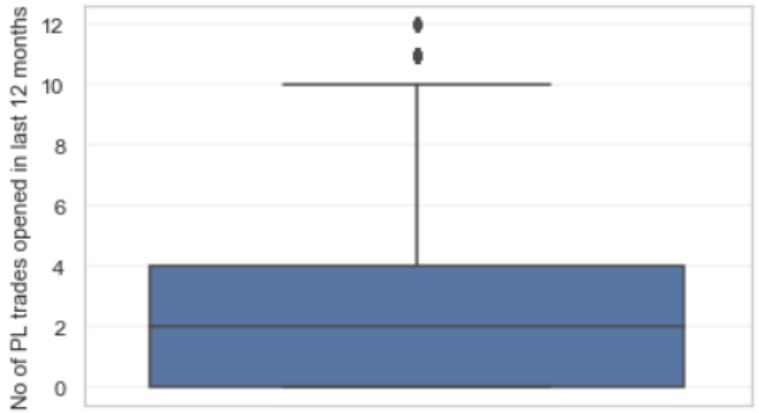
Outlier Treatment Of Credit Bureau Data: (contd.)

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 in demographic and credit burueau data using user defined functions.
- Monotonic binning ensures linear relationship is established between independent and dependent variables. We have used spearman correlation to perform monotonic binning.
- For 9 variables with Missing values, the variable values were imputed 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.
- 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

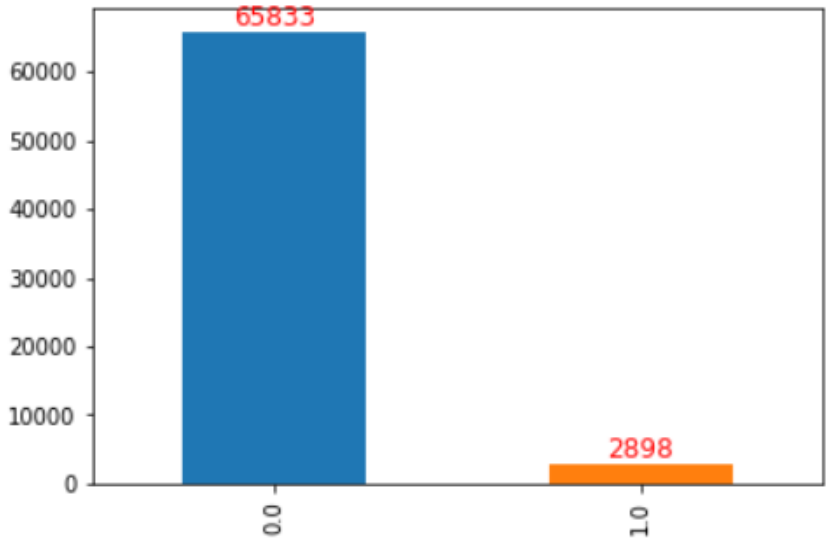
Credit Bureau Data

	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

Exploratory Data Analysis On Performance Tag:

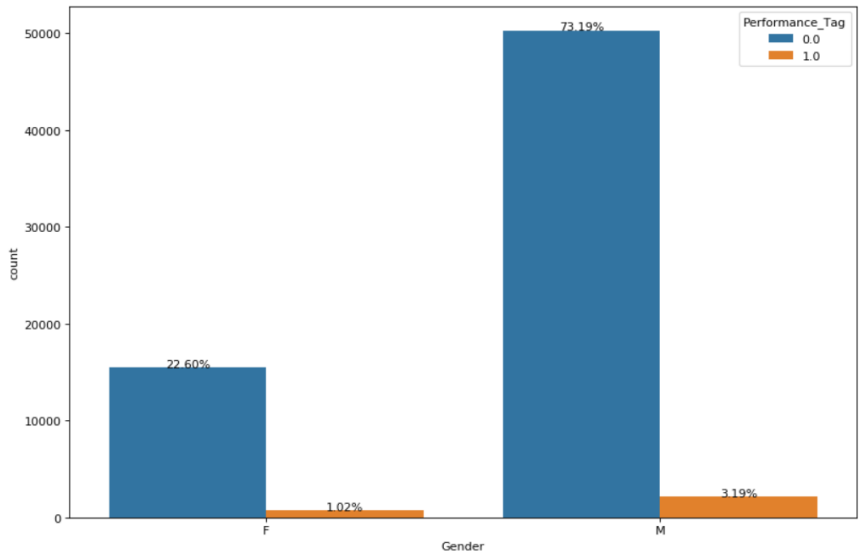
The variations in the distribution of performance tag is shown in the first figure. We can see that almost 3000 records are defaulters in our dataset having performance tag 1.



Performance Tag

Exploratory Data Analysis On Gender:

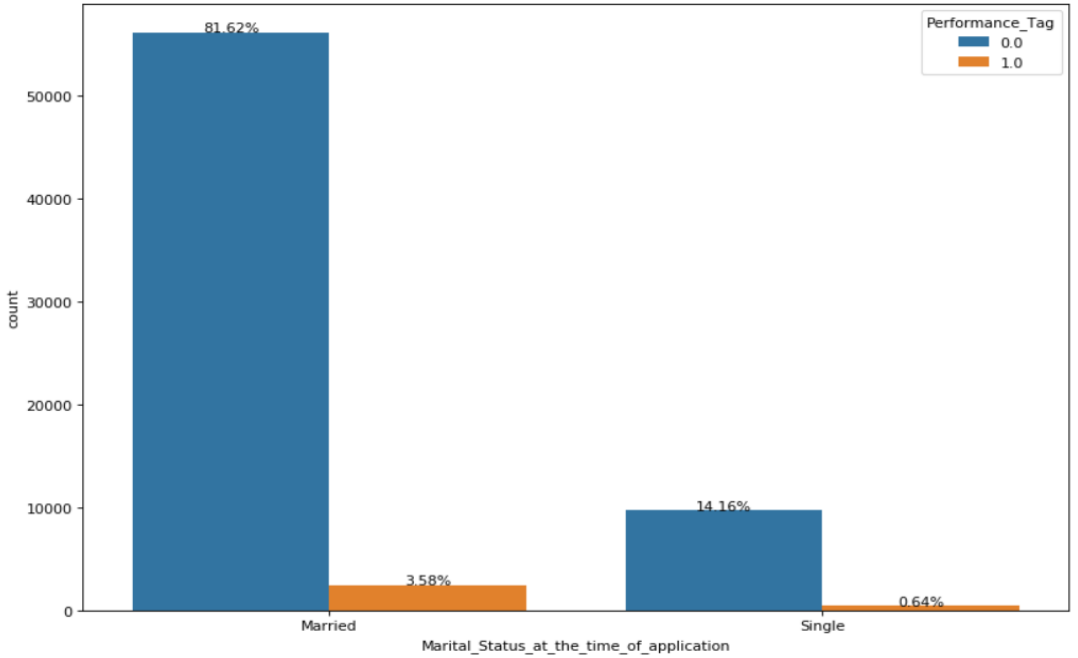
The distribution of performance tag over Gender is shown in the second figure. We can see that approx. 1% of the Female records and 3% of Male records are likely to default in our dataset.



Gender

Exploratory Data Analysis On Marital Status:

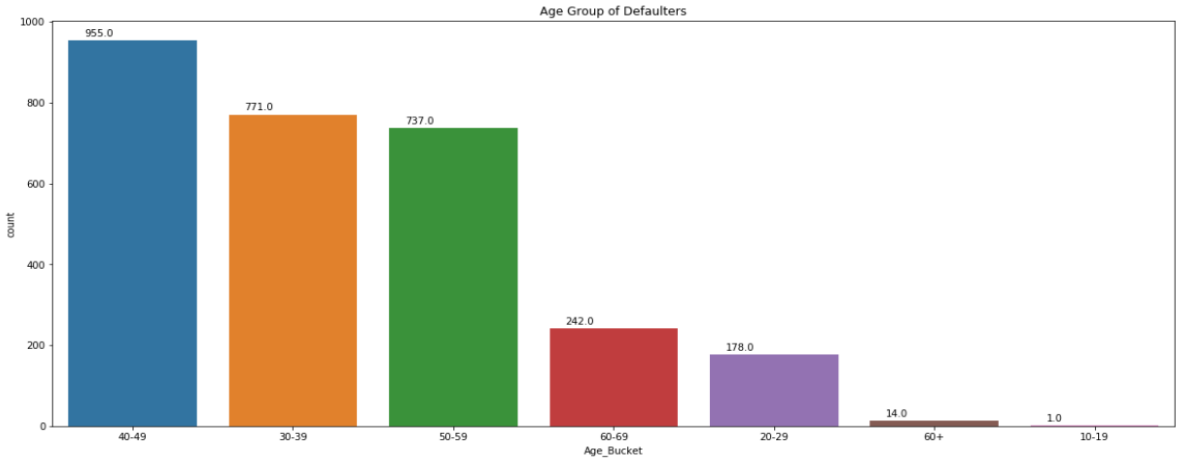
The variations of Marital Status in the distribution of performance tag is shown in the first figure. We can see that married people have a higher tendency of defaulting than single people.



Marital Status At the time of Application

Exploratory Data Analysis On Age Bins:

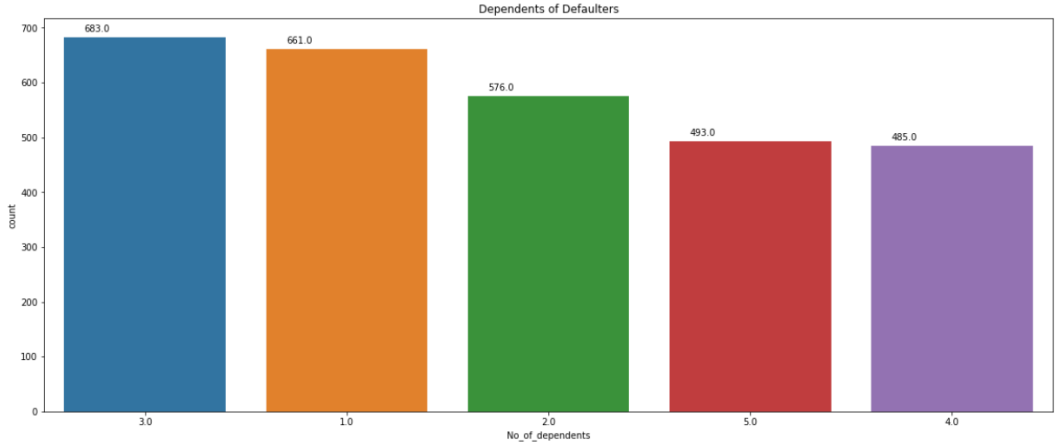
The variations of age in the distribution of performance tag is shown in the figure. We can observe that the age range of 40-49 have a much higher tendency of being a defaulter than others. 955 records in our dataset are defaulters who have age range of 40-49.



Age Bucket

Exploratory Data Analysis On No Of Dependents of Defaulters:

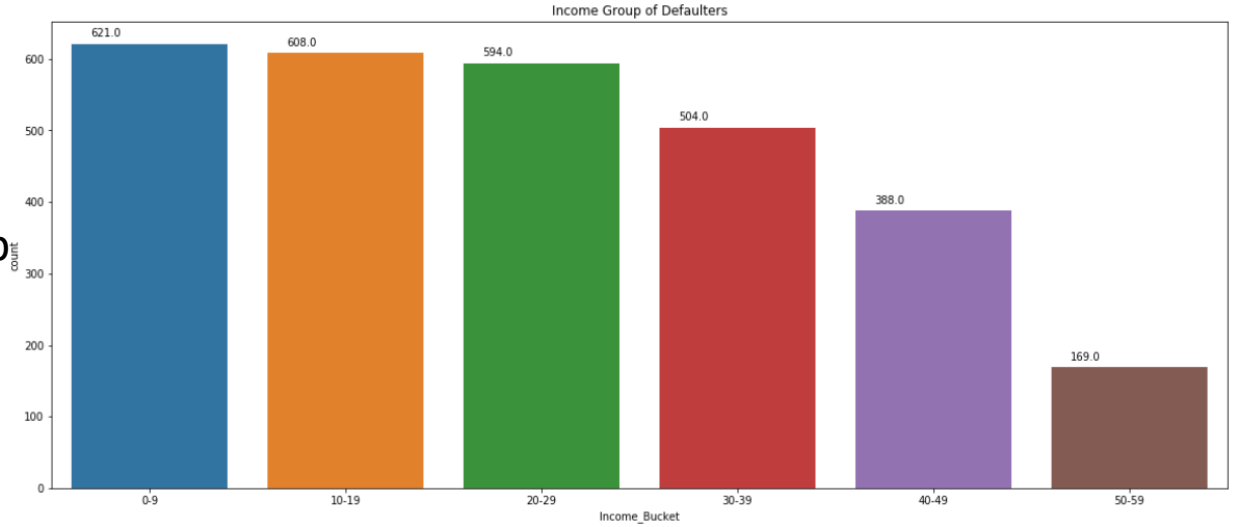
The frequency of No Of Dependents of records having Performance Tag as 1 is shown in the first figure. We can see that defaulters having 3 dependents are of maximum frequency followed by defaulters having 1 dependent.



No Of Dependents

Exploratory Data Analysis On Income:

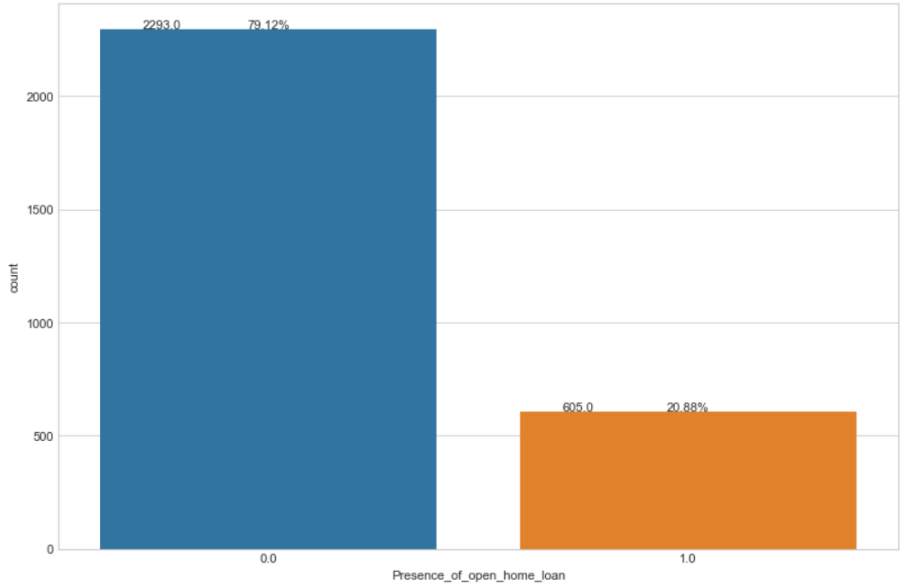
The variations in the distribution of Income group having the highest defaulters is shown in the figure. We can see that group having 0-9 range in income group have the highest tendency to default.



Income Bucket

Exploratory Data Analysis On Open Home Loans :

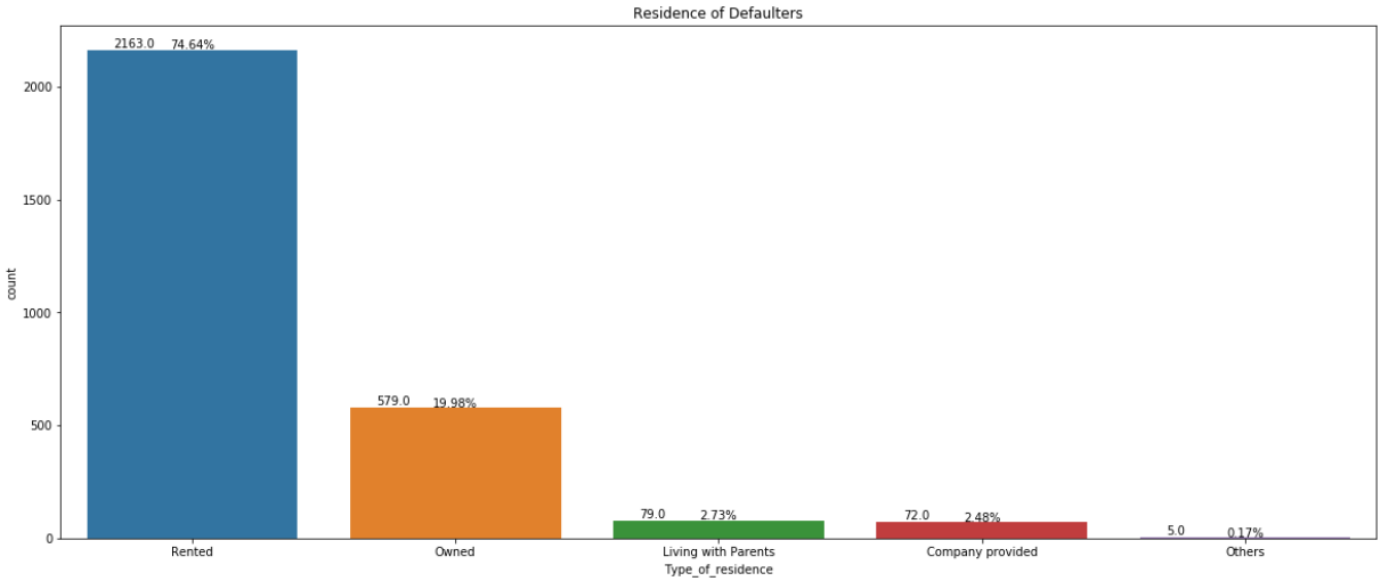
Around 80% of the defaulters does not have `Home Loan` which means the applicants taking home loan has less chances of being a defaulter.



Open Home Loans

Exploratory Data Analysis On Types Of Residence:

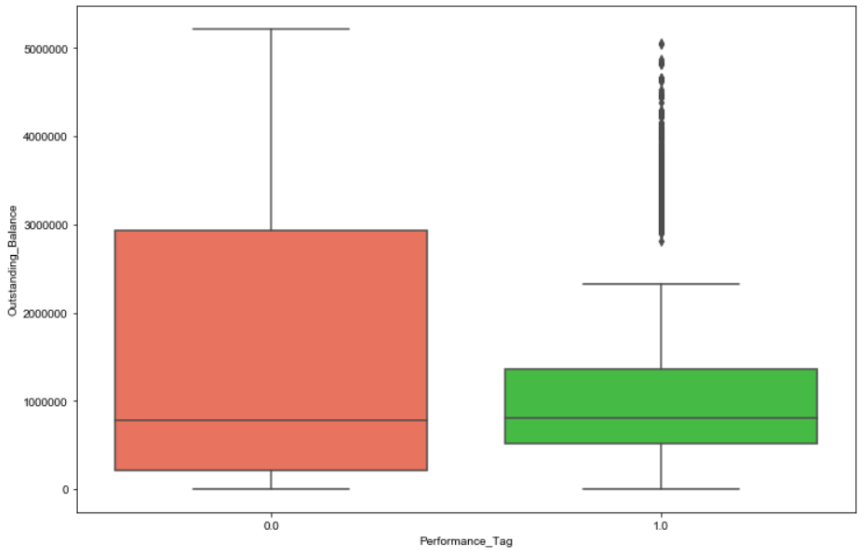
The variation of Type Of Residence is shown in figure. It is evident that around 77% of the defaulters stay in a rented place.



Types of residence

Exploratory Data Analysis On Outstanding Balance:

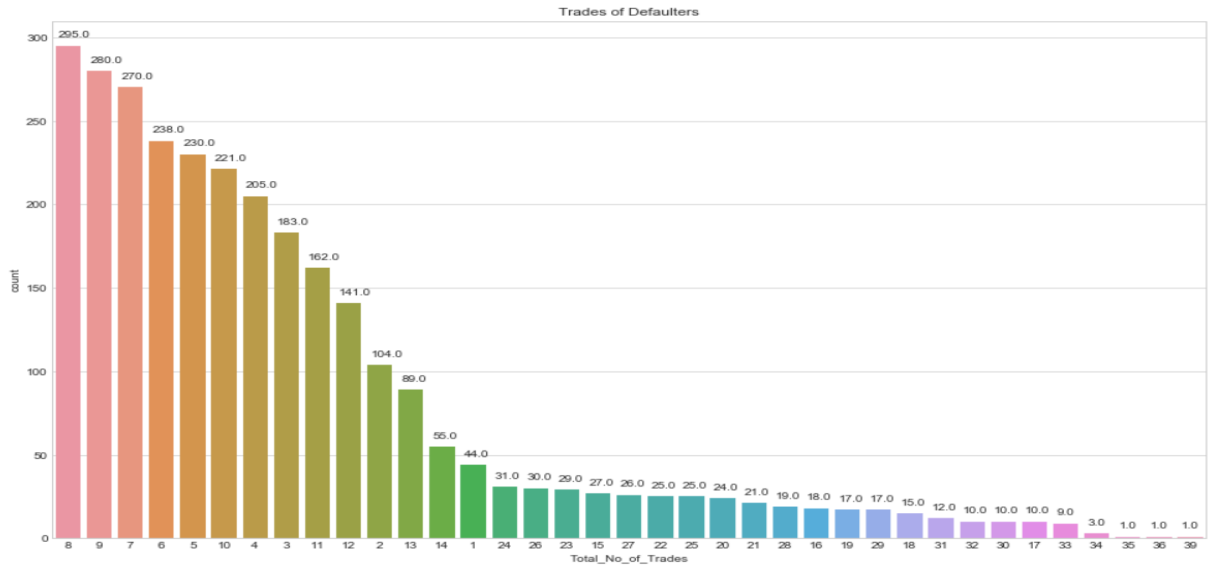
A huge outlier can be seen in the defaulters from the boxplot where the outstanding amount is in a very high range. From the Boxplot it is clear that higher the outstanding balance on the applicant, higher the chances of him being a defaulter.



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

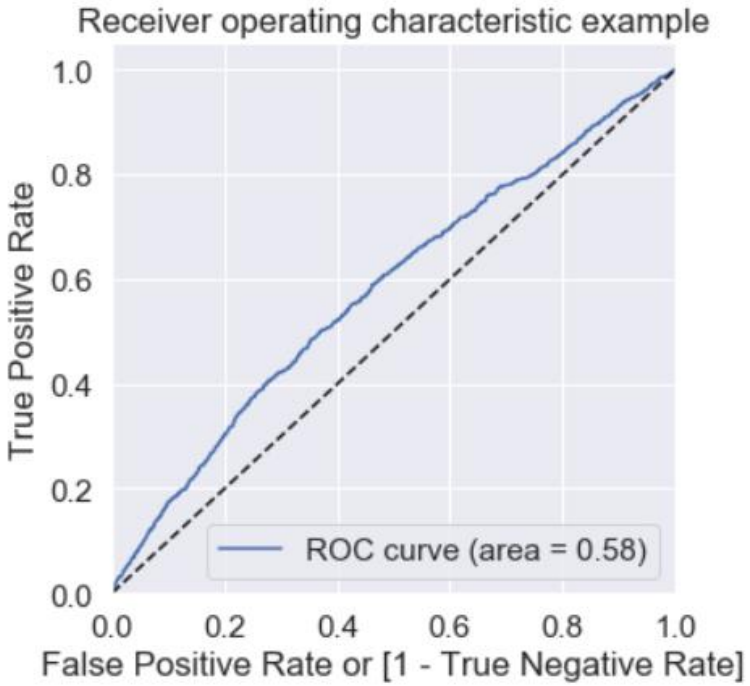
Top Correlations Amongst Variables in Merged Dataset

```
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}
```

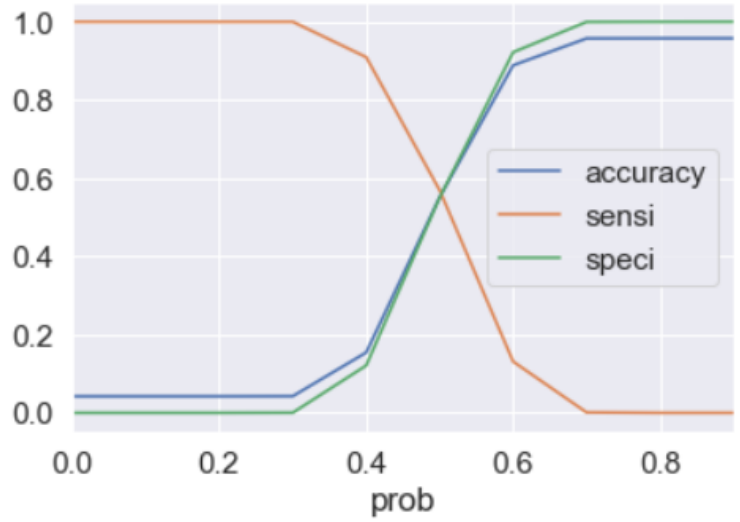



```
Accuracy:0.552
Sensitivity:0.567
Specificity:0.551
AUC:0.58
Ks_2sampResult(statistic=0.13184498117905452, pvalue=2.5632291794947974e-13)
```

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)
```

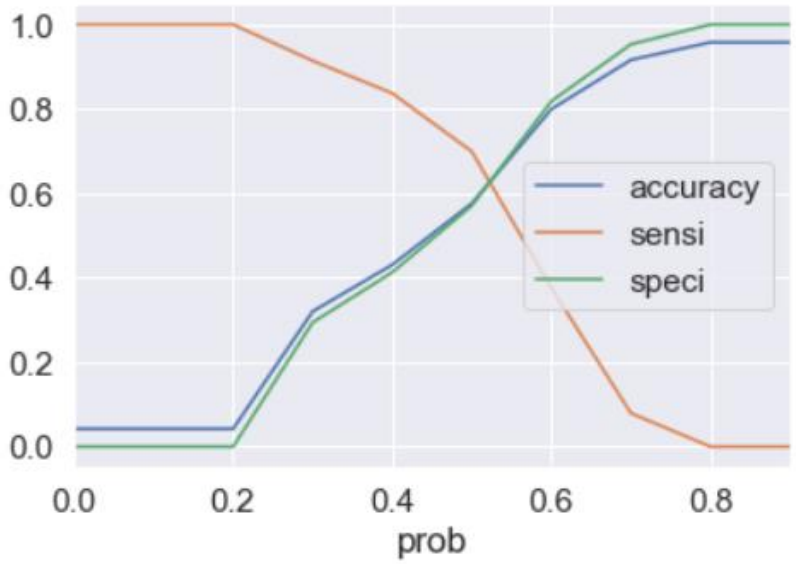


	precision	recall	f1-score	support
0.0	0.97	0.55	0.70	20034
1.0	0.05	0.57	0.10	884
accuracy			0.55	20918
macro avg	0.51	0.56	0.40	20918
weighted avg	0.93	0.55	0.68	20918

Final Model Building: Logistic with Balanced Class Weight and Grid Search CV

Logistic Regression with StratifiedKfold cross validation (with 5 folds) and grid search cv is applied on the merged dataset with AUC 0.68.

```
GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=4, shuffle=True),
             error_score='raise-deprecating',
             estimator=Pipeline(memory=None,
                                steps=[('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)
```




```
Accuracy:0.576
Sensitivity:0.699
Specificity:0.571
AUC:0.68
Ks_2sampResult(statistic=0.27905720267441075, pvalue=3.518824951155083e-58)
```

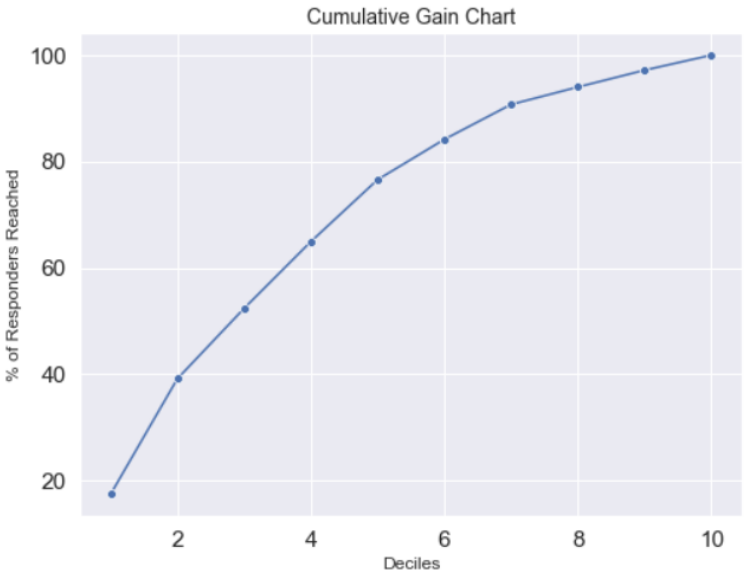
Application Score Card:

- Application Scorecards are tools that allow organizations to predict the probability that an applicant will behave in a way, helping businesses to make effective automated decisions.
- This method can be summarized as follows:
    - The application score for each applicant calculated using the logistic regression model, ranges from 302.47 to 367.46. Score increases by 20 points for doubling odds for good customers. Application score for odds of 10 to 1 is 400. We are yet to finalize the final Cut-off score. Higher the scores indicate lesser risk for defaulting.
    - Method used for computation of application scorecard:
    - a) Computed the probabilities of default for the entire population of applicants using the model. b) Computed the odds for the good. Since the probability computed is for rejection (bad customers), Odd(good) = (1-P(bad))/P(bad) c) Used the following formula for computing application score Application score = 400 + slope * (ln(odd(good)) - ln(10)) , where slope is 20/(ln(20)-ln(10))

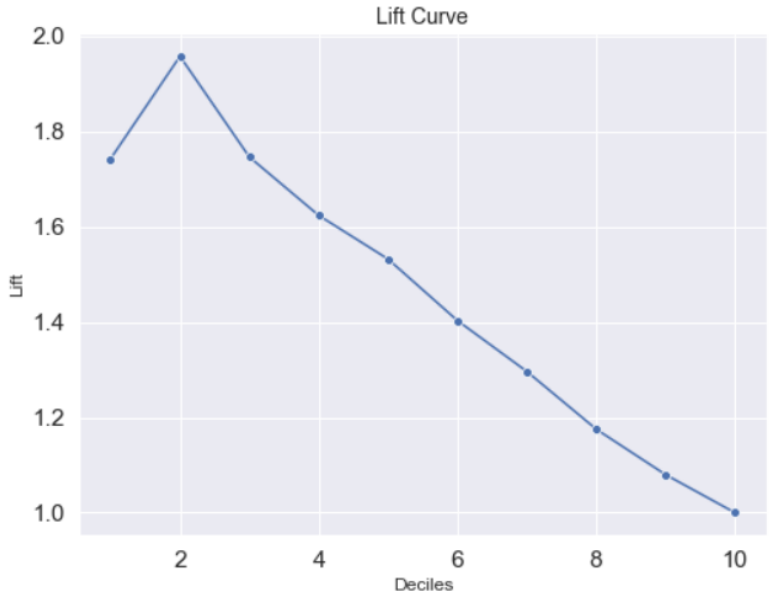
	p	y	Actual Prediction	Final Prediction	Score
0	0.422353	0.0	0.0	0	342.596174
1	0.421953	0.0	0.0	0	342.643569
2	0.547999	0.0	1.0	1	328.004442
3	0.284001	0.0	0.0	0	360.242664
4	0.694050	0.0	1.0	1	309.926602

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	y	cumresp	gain	cumlift
9	1	2092	154	154	17.420814	1.742081
8	2	2092	192	346	39.140271	1.957014
7	3	2065	117	463	52.375566	1.745852
6	4	2118	111	574	64.932127	1.623303
5	5	2089	103	677	76.583710	1.531674
4	6	2093	67	744	84.162896	1.402715
3	7	2092	58	802	90.723982	1.296057
2	8	2059	29	831	94.004525	1.175057
1	9	2062	28	859	97.171946	1.079688
0	10	2155	25	884	100.000000	1.000000



Road-Map for final submission

- We have plans to build more classification models like random forest, xgboost, svm on the final merged data to finalize the model with better discriminative power. We also propose to use class weights to balance the two imbalanced classes and see if it improves the discriminative power of the models.
- We propose to evaluate the model using different techniques like Confusion matrix, K-fold cross validation techniques, KS-Statistics, and based on that we would decide on the best model for our case.
- We will re-build the application scorecard on the final model to find the cut-off score and predict the potential financial benefits for the company.
- Predict the likelihood of default for the rejected candidates using the model.

Thank You!