BSFI CAPSTONE PROJECT FINAL SUBMISSION

Group Members:

1. Suman Sarkar
2. Aniket Anand

Solution Approach

Business Understanding and Data Understanding

Data Cleansing and Preparation

Exploratory Data Analysis (Graphs & plots)

Model Building & Model Evaluation

Assessing Financial benefit of the model

Business understanding

- 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’.

- In this project, our task is to help CredX 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 of your project.

About the Data

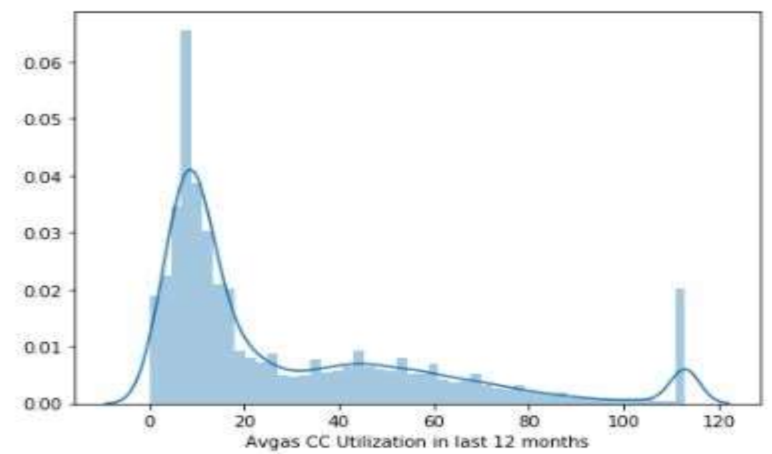
- The Demographics dataset has 71295 observations with 12 variables.
- The Credit bureau dataset has 71295 observations with 19 variables.
- We would be merging the datasets using a common key which is 'Application ID' in our case.
- Both files contain a **performance tag**, which indicates whether the applicant has gone 90 days past due (DPD) or worse in the past 12 months (i.e. defaulted) after getting a credit card. It would serve as a target variable for us whether customer has defaulted or not.

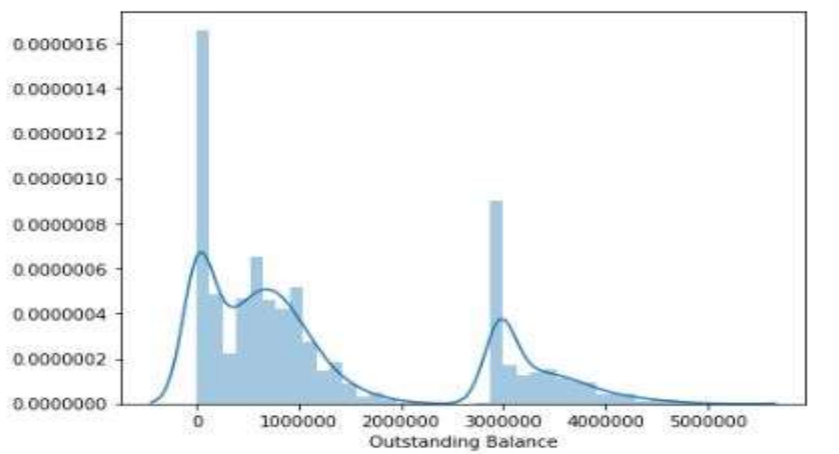
Data Quality Checks

- All Application.ID in demographic file are present in credit_bureau
- Both occurrences of 3 duplicate Application ID records (765011468, 653287861, 671989187) has been excluded from the dataset.
- Both files have a Synonym field “Performance Tag” hence, keeping only one field and removing the redundant column.
- The 1425 rows with no performance tag indicates that the applicant is not given credit card, we are going to append this with validation set
- The data seems to have outliers also by looking at summary. These will be observed and taken care in the next process.
- We notice some issue with 'Age' & 'Income' column in both the cases it is showing -ve which is not possible, For this case we are going to replace the value which are <0 with Q1 value for respective columns.
- We can see there is significant number of missing values in dataframe which we will try to impute them with WOE.

WOE AND IV ANALYSIS

- The weight of evidence tells the predictive power of an independent variable in relation to the dependent variable. Since it evolved from credit scoring world, it is generally described as a measure of the separation of good and bad customers.
- Information value is one of the most useful technique to select important variables in a predictive model. It helps to rank variables on the basis of their importance.
- We can see there is significant number of missing values in dataframe we will try to impute them with WOE.
- Avgas CC Utilization in last 12 months and Outstanding Balance , as this distributions are skewed we are imputing with median

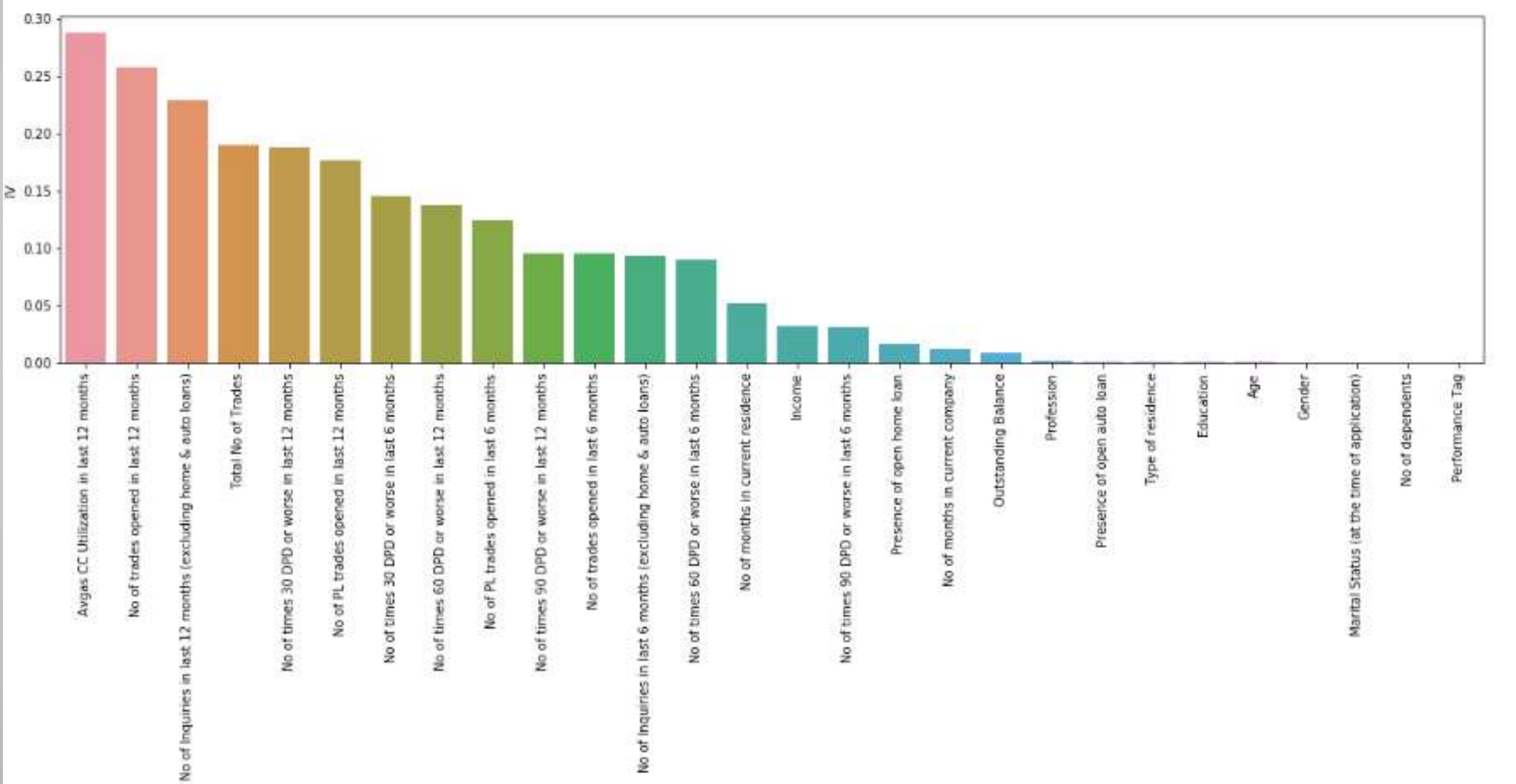




From the IV values we can conclude that parameters in the demographic data don’t play much ,significant role in prediction and most of the significant variables are from Credit Bureau data.
We are aware that Information Value 0.1 to 0.3, then the predictor has a medium strength relationship to the Goods/Bads odds ratio.

	VAR_NAME	IV
1	Avgas CC Utilization in last 12 months	0.287537
19	No of trades opened in last 12 months	0.257429
6	No of Inquiries in last 12 months (excluding h...	0.229218
26	Total No of Trades	0.189907
13	No of times 30 DPD or worse in last 12 months	0.188045
8	No of PL trades opened in last 12 months	0.176644
14	No of times 30 DPD or worse in last 6 months	0.145708
15	No of times 60 DPD or worse in last 12 months	0.137676
9	No of PL trades opened in last 6 months	0.124744
17	No of times 90 DPD or worse in last 12 months	0.095714
20	No of trades opened in last 6 months	0.095337
7	No of Inquiries in last 6 months (excluding ho...	0.092939
16	No of times 60 DPD or worse in last 6 months	0.089574
12	No of months in current residence	0.052060
4	Income	0.032533
18	No of times 90 DPD or worse in last 6 months	0.030711
24	Presence of open home loan	0.016970
11	No of months in current company	0.012735
21	Outstanding Balance	0.008403
25	Profession	0.002228

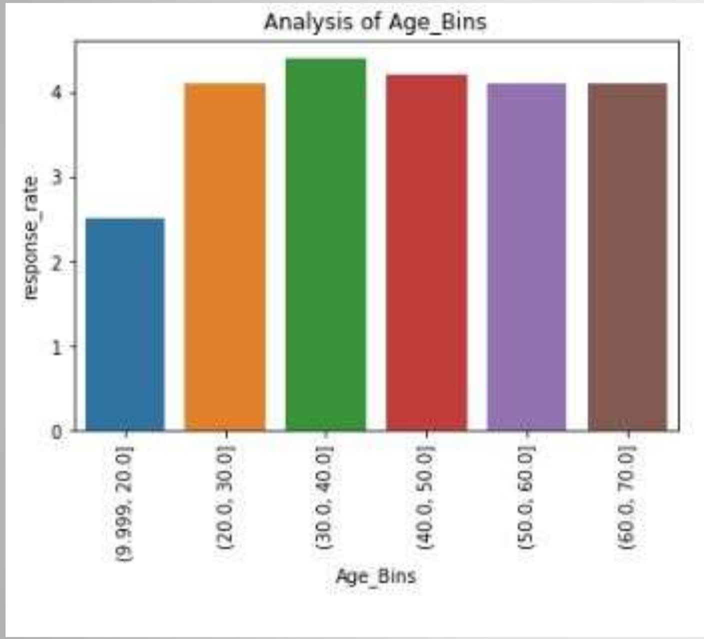
Graph showing the rank and predicting power of the independent variable



Exploratory Data Analysis

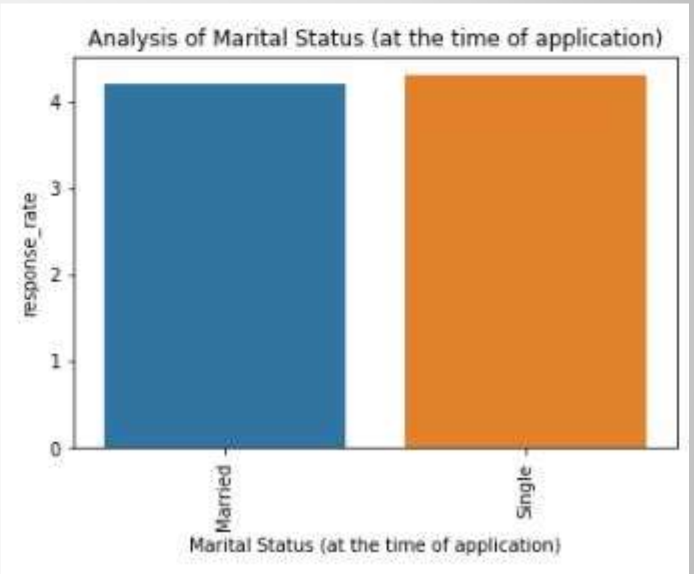
Before Starting modelling lets do some EDA on the actual values

Age



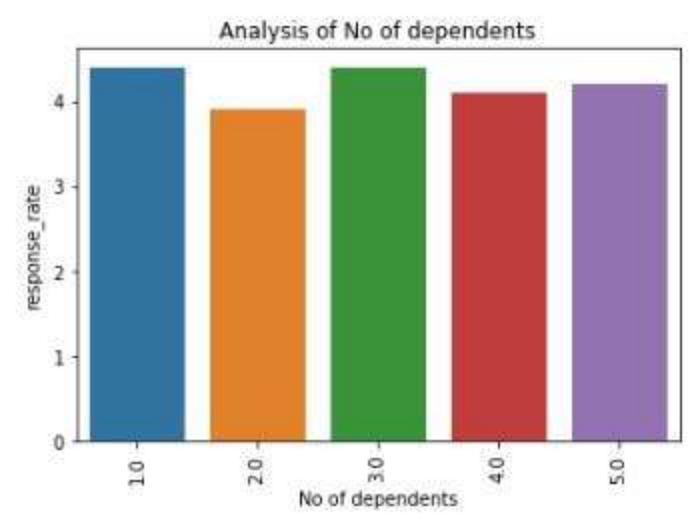
It seems the age group of [30-50] are more likely responsible for credit loss

Marital Status



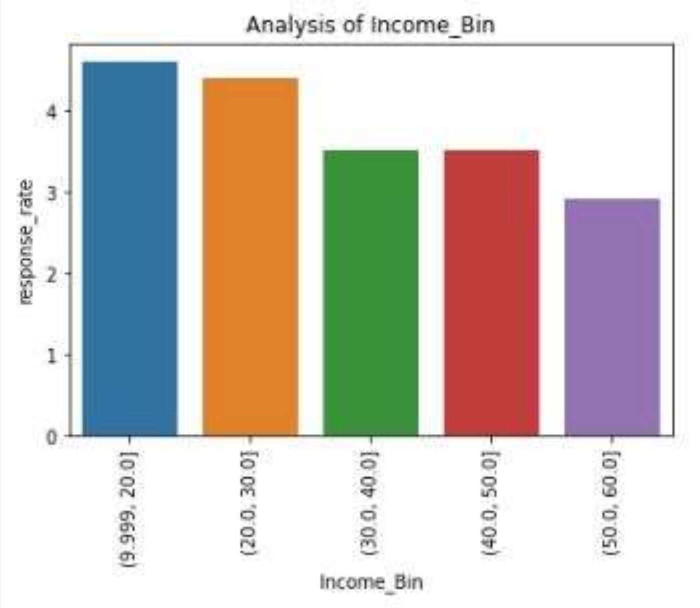
Here we can see that single people are more likely to default

No of dependents



No of dependent 1,3 are more likely to default

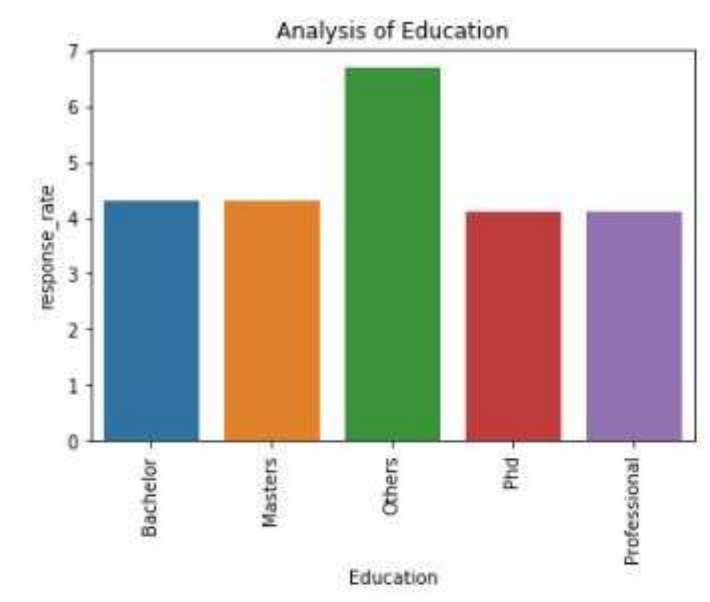
Income



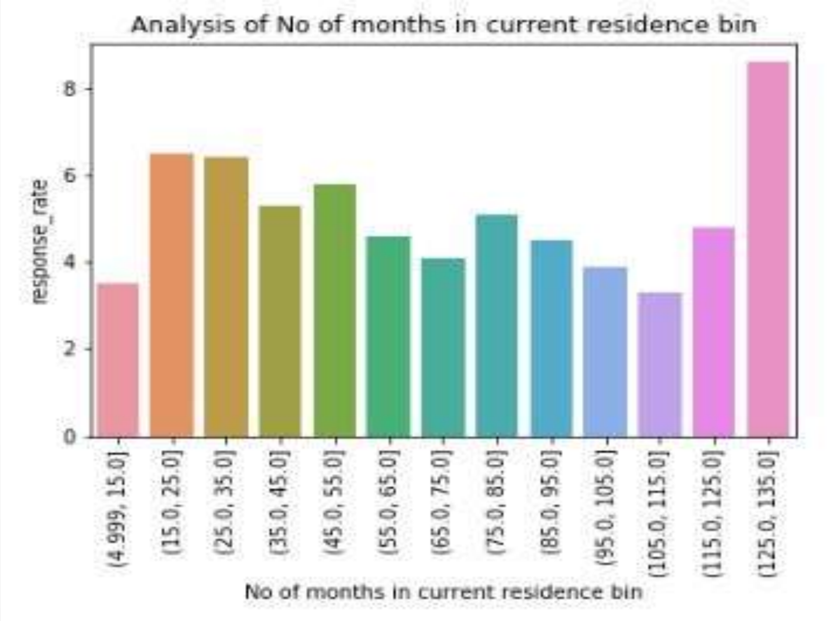
Here we can see that the people with low income are more likely to default on credit

Education

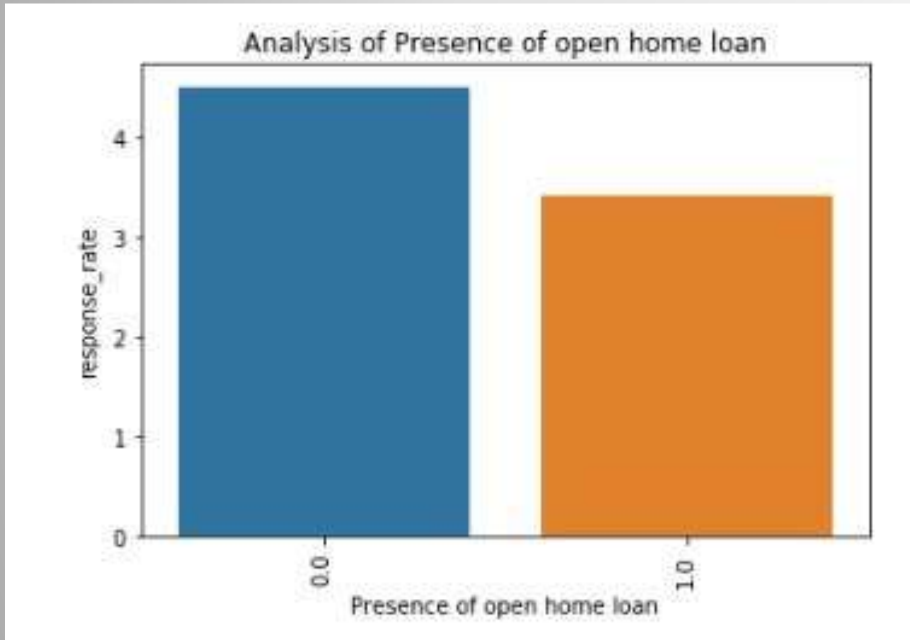
No of Months in Current Residence



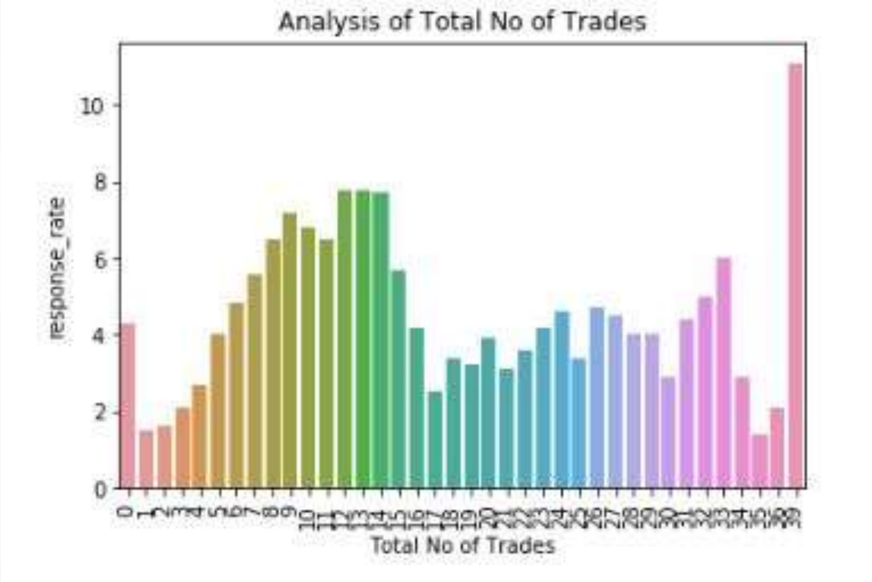
It seems others are uneducated people who are responsible for a credit loss



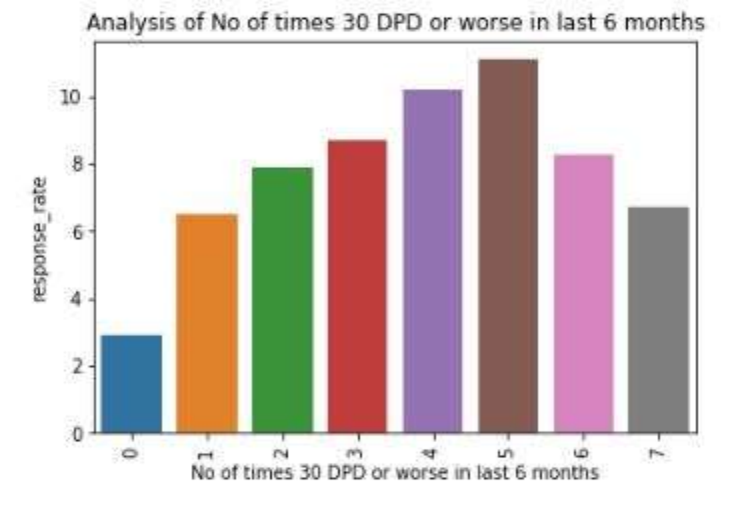
Presence of open home loan

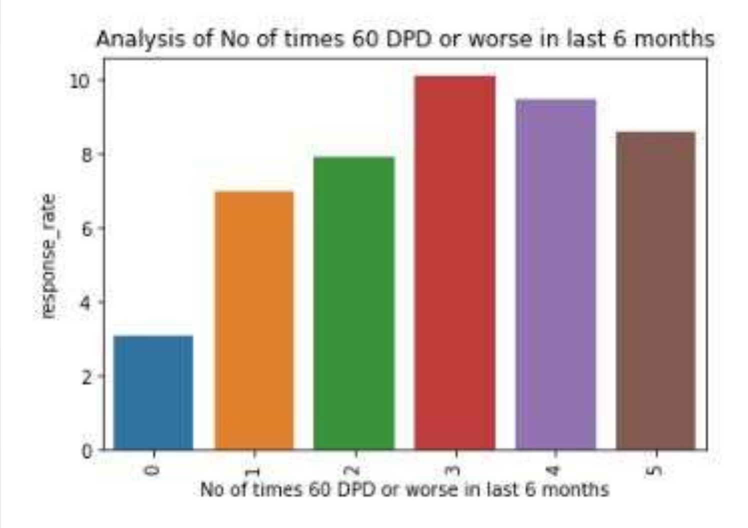


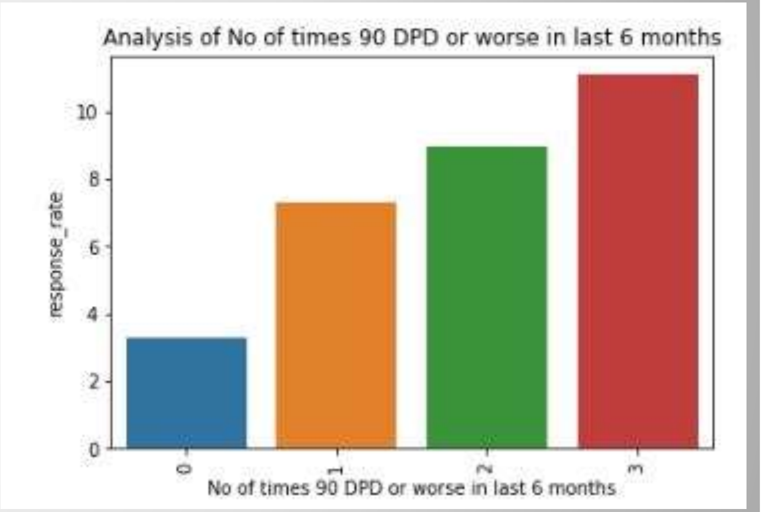
Total No of Trades



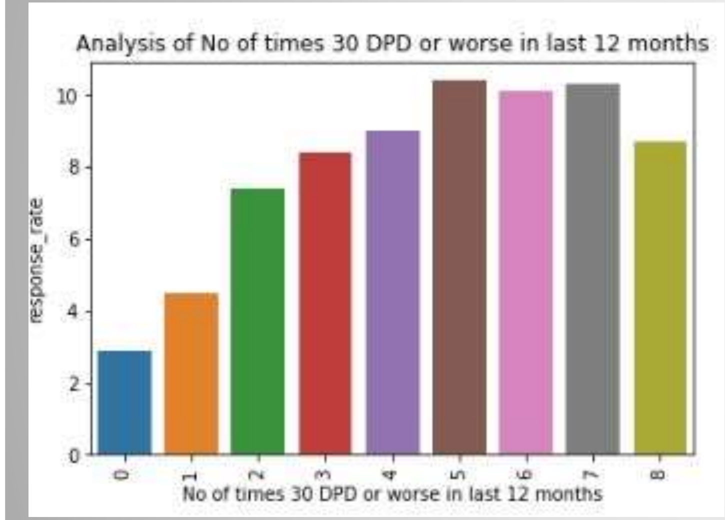
- Number of defaulters are increasing with increase in Number of 30/60/90 DPD or worse in last 6 months variable values. Hence these variables can be important predictors.

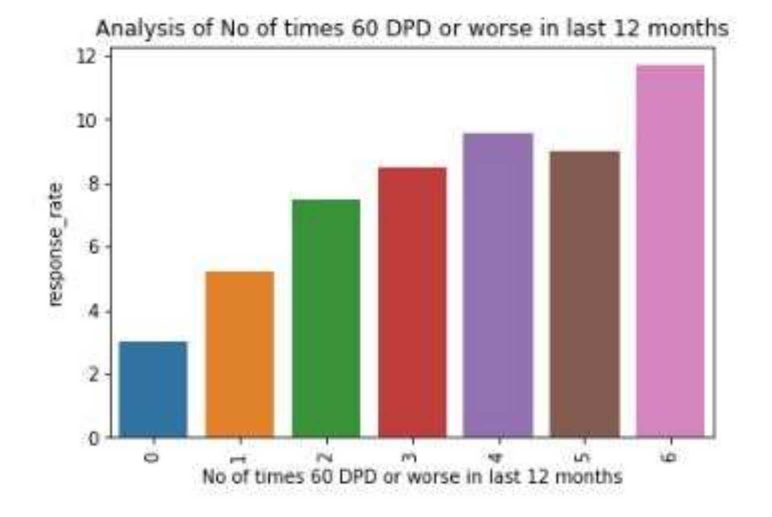


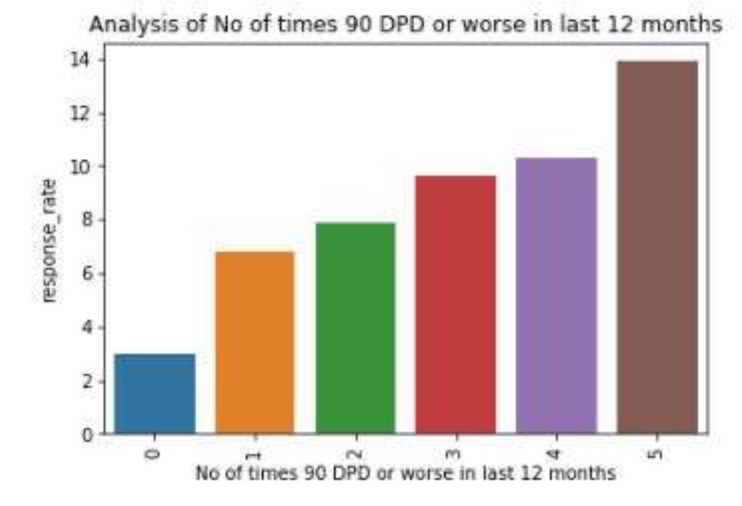




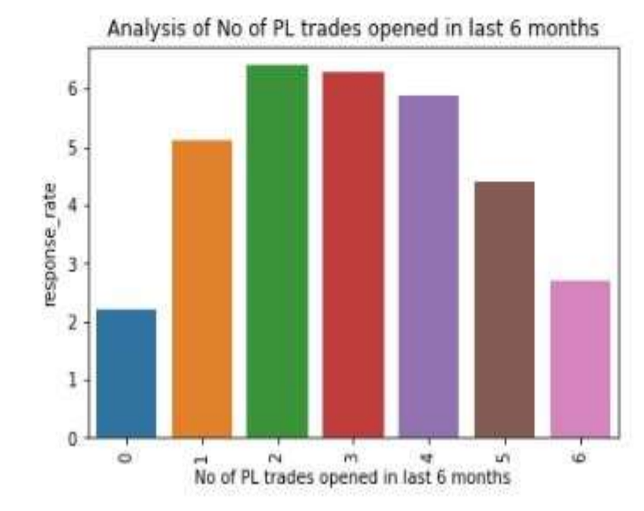
- Number of defaulters are increasing with increase in Number of 30/60/90 DPD or worse in last 12 months variable values. Hence these variables can be important predictors.

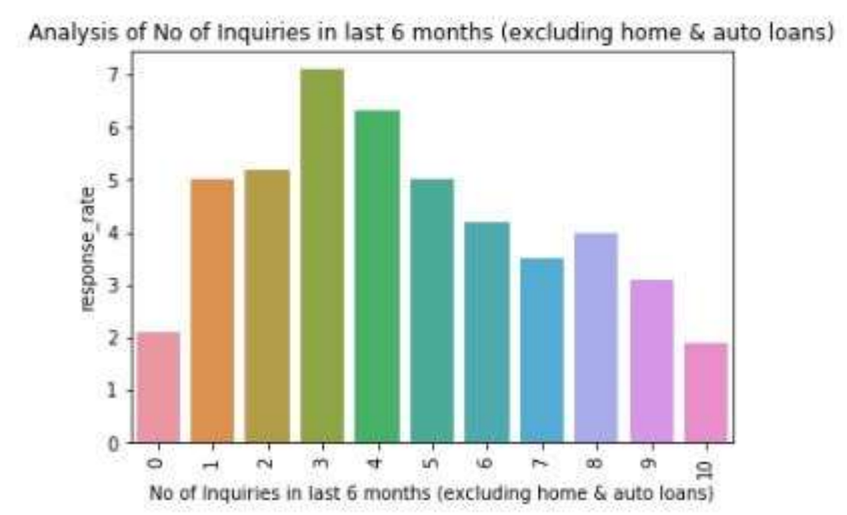


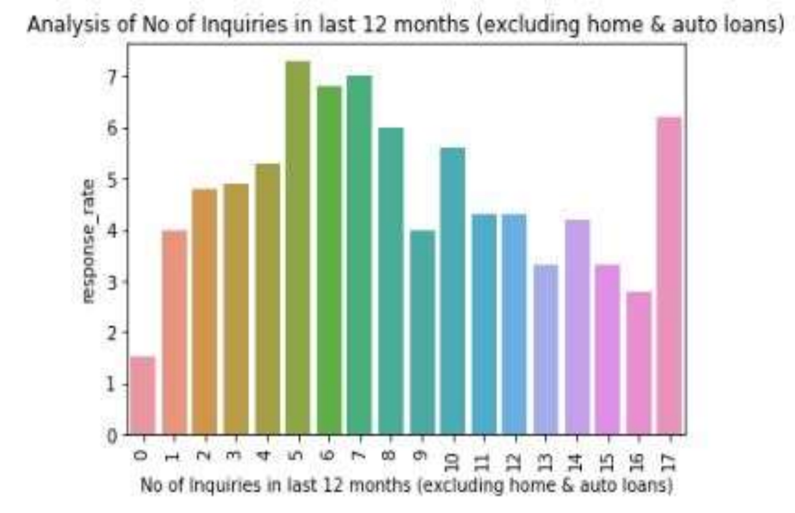




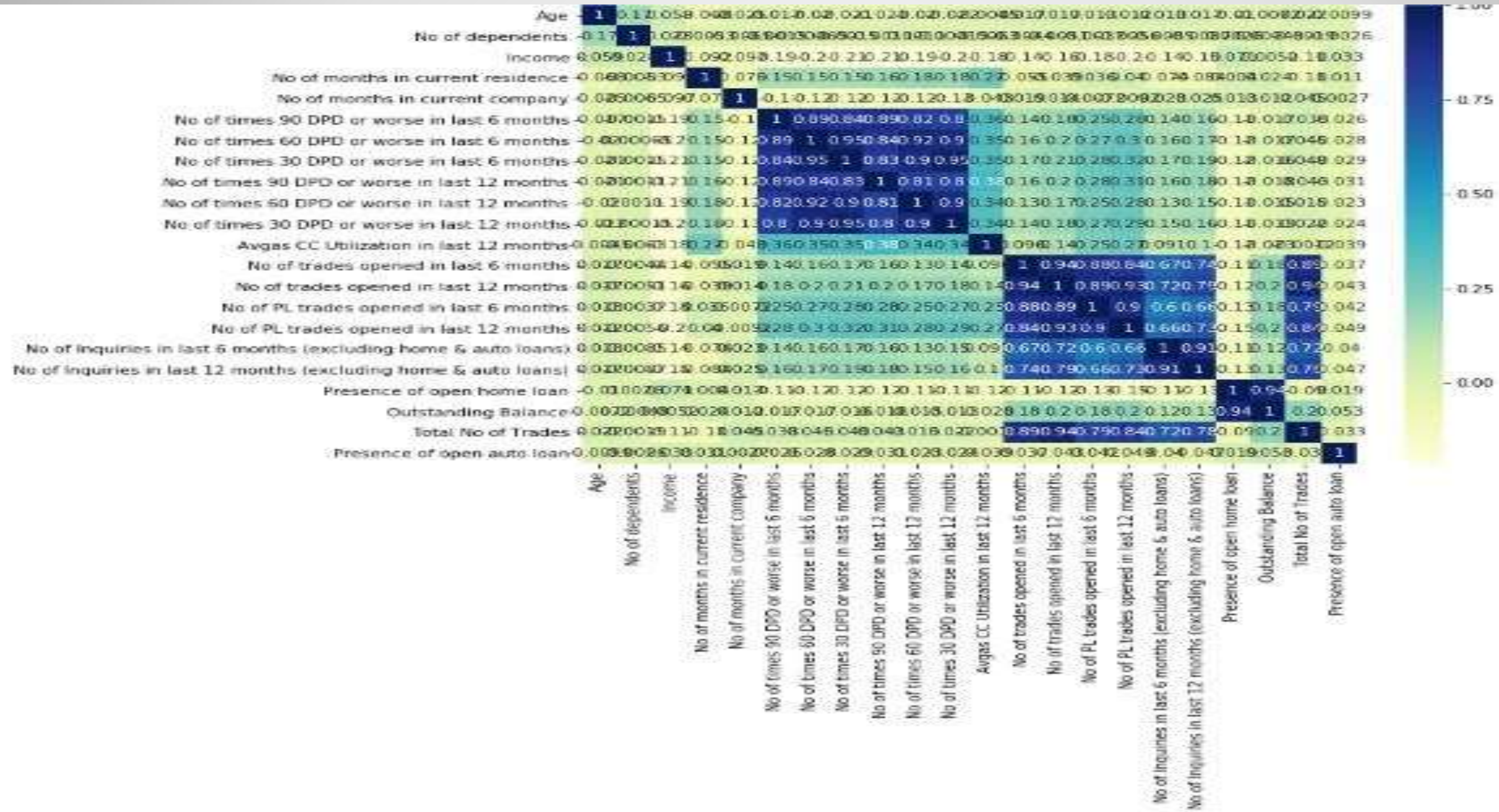
- Number of enquiries and numbers of enquiries fields don’t show any pattern.







Correlation Heatmap to show relationship between the various variables.



MODEL BUILDING AND EVALUATION

- Scaling is performed for all variables except Application ID and performance tag to standardize the data into common scale.
- The Final dataset is split into Train and Test in 70:30 ratio for model building.
- The data is highly imbalanced.We use **SMOTE** (Synthetic Minority Over-sampling Technique).
- We have evaluated the model based on accuracy, Sensitivity and Specificity values but the overall performance of model seemed low.

```
Accuracy Score=0.5432946901388293
Precision Score=0.04576253518190347
Recall Score=0.5116550116550117
AUC Score=0.5281500447520445
TN:10949,FP:9154,FN:419,TP:439
sensitivity:0.5116550116550117,specicity:0.5446450778490772,fpr:0.4553549215092277
F1 Score:0.0840110994163238
log loss:15.774391772499213
              precision    recall  f1-score   support

         0.0       0.96      0.54      0.70     20103
         1.0       0.05      0.51      0.08       858

    accuracy                           0.54     20961
   macro avg       0.50      0.53      0.39     20961
weighted avg       0.93      0.54      0.67     20961
```

Model Building Approach

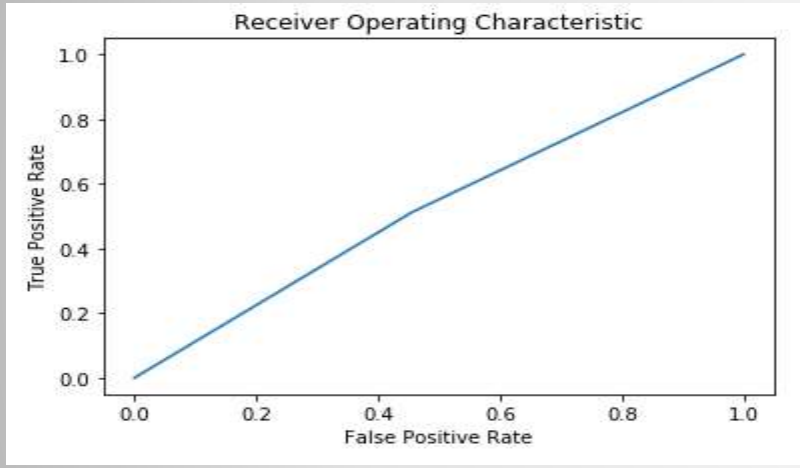
- **Outlier Treatment-** Outlier detection is done using boxplot on continuous variables and quintiles function and the variables with outliers has been corrected by capping the outliers to the nearest non-outlier values.
- **Data Scaling:** Scaling is performed for all variables except Application ID and performance tag to standardize the data into common scale.
- **DATASPLIT:** The final dataset is split into Train and Test in 70:30 ratio for model building.

  •All models are trained on training datasets and regularization was done by tuning of hyper parameters with cross validation on validation datasets.

  •All the models are tested on test datasets that were kept separate from training and validation datasets.
- **Data Sampling:** The given data is highly imbalanced. We have used **SMOTE** which stands for Synthetic Minority Oversampling Technique. This is a statistical technique for increasing the number of cases in your dataset in a balanced way.

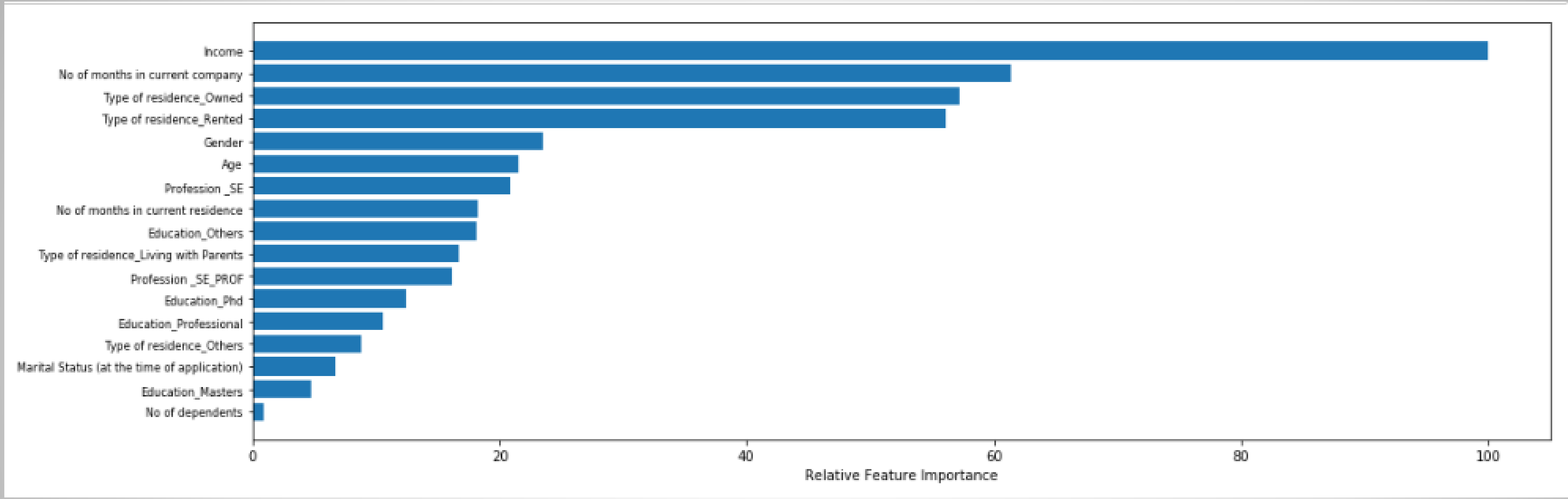
Model-building

- Lets start with Demographic data model (LogisticRegression)



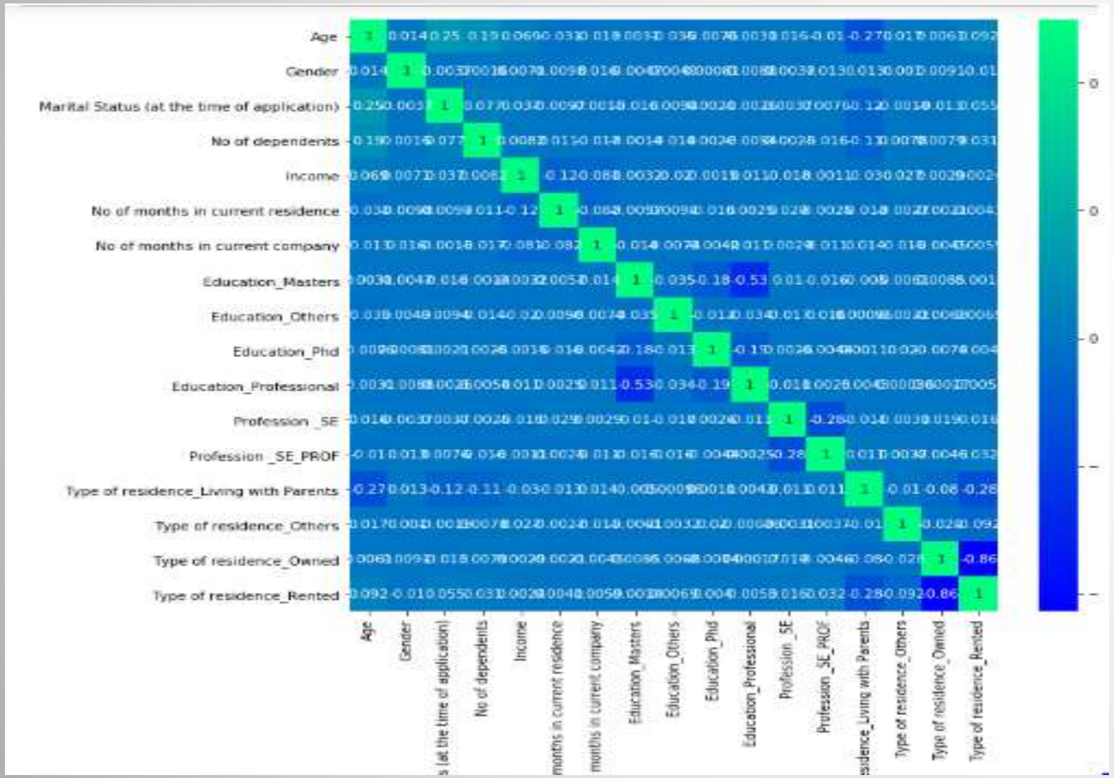

```
Accuracy Score=0.5432946901388293
Precision Score=0.04576253518190347
Recall Score=0.5116550116550117
AUC Score=0.5281500447520445
TN:10949,FP:9154,FN:419,TP:439
sensitivity:0.5116550116550117,specicity:0.5446450778490772,fpr:0.4553549215092277
F1 Score:0.0840110994163238
log loss:15.774391772499213
              precision    recall  f1-score   support

         0.0       0.96      0.54      0.70     20103
         1.0       0.05      0.51      0.08       858

    accuracy                           0.54     20961
   macro avg       0.50      0.53      0.39     20961
weighted avg       0.93      0.54      0.67     20961
```

We can notice that it is more likely a random model



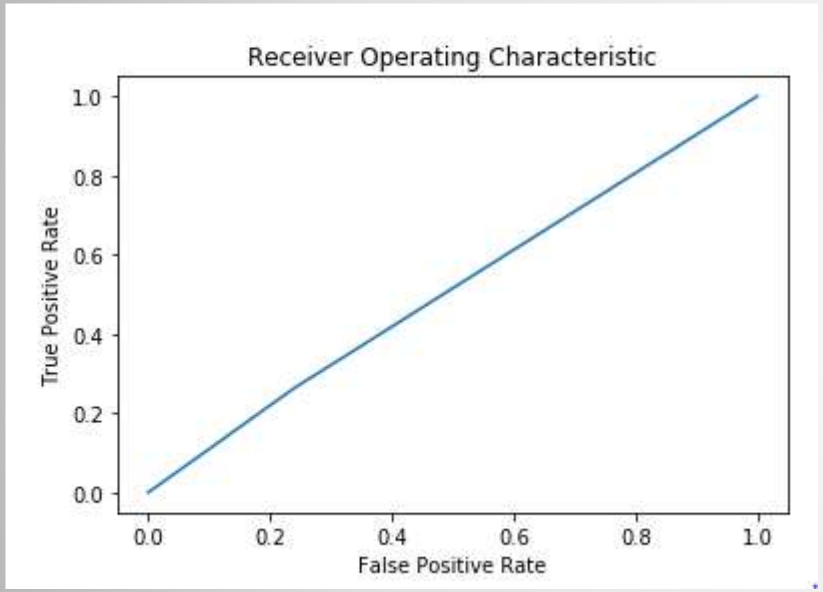
Above graph we can identify the important features

- Before we move on to creating more models lets create a heat map and understand the multi co linearity.



We can see that attributes are negatively correlated in some of the cases, but strong correlation is not present between the features

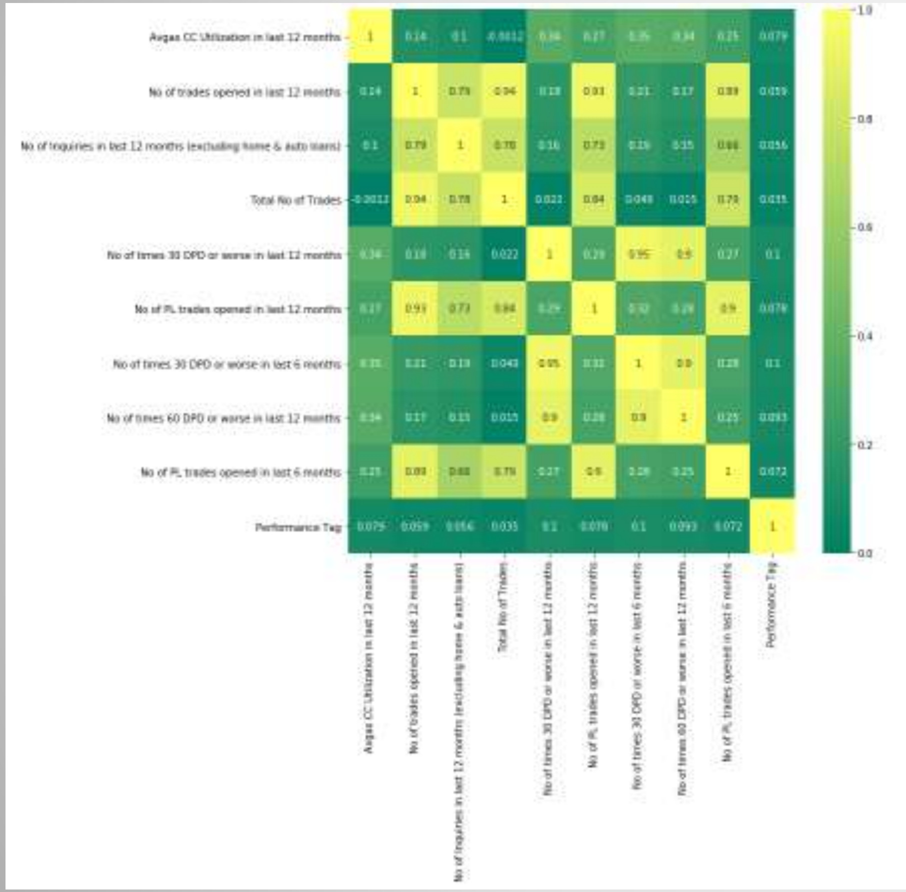
- We went for PCA before passing it to Random Forest Classifier



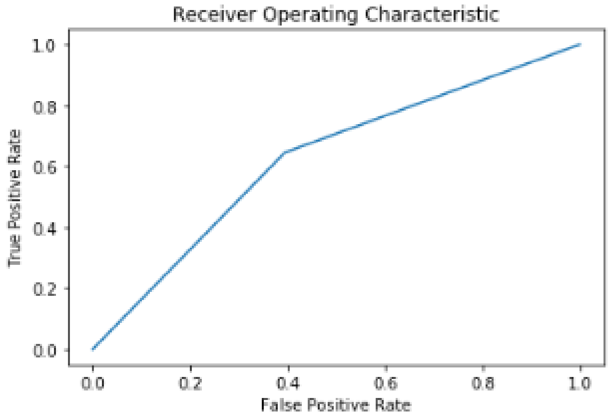

```
Accuracy Score=0.740947473880063
Precision Score=0.044802867383512544
Recall Score=0.26223776223776224
AUC Score=0.5118083304548011
TN:15306,FP:4797,FN:633,TP:225
sensitivity:0.26223776223776224,specificity:0.76137889867184,fpr:0.2386211013281597
F1 Score:0.07653061224489797
log loss:8.947540264931904
              precision    recall  f1-score   support

         0.0       0.96      0.76      0.85     20103
         1.0       0.04      0.26      0.08       858

    accuracy                           0.74     20961
   macro avg       0.50      0.51      0.46     20961
weighted avg       0.92      0.74      0.82     20961
```

We already know that only demographics data is having very low predictive power final result

- Lets Create a model on variable where IV is high

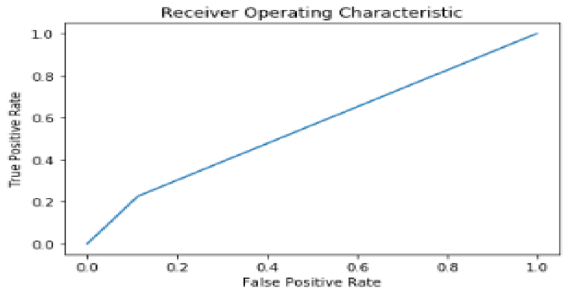



```
Accuracy Score=0.6072041984732824
Precision Score=0.06999176567462652
Recall Score=0.6453362255965293
AUC Score=0.6253929356348751
TN:12132,FP:7906,FN:327,TP:595
sensitivity:0.6453362255965293,specificity:0.6054496456732209,fpr:0.3945503543267791
F1 Score:0.12628674519792
log loss:13.566987961669023
              precision    recall  f1-score   support

         0.0       0.97      0.61      0.75     20038
         1.0       0.07      0.65      0.13       922

    accuracy                           0.61     20960
   macro avg       0.52      0.63      0.44     20960
weighted avg       0.93      0.61      0.72     20960
```



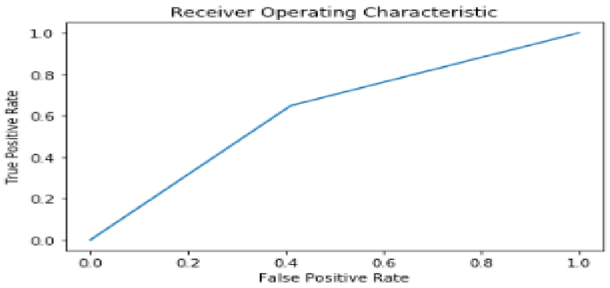
- We also tried Random Forest Clarifier, ANN and XGBoost Classifier
- We can see that the random forest is giving the best score

```
Accuracy Score=0.8579675572519084
Precision Score=0.08451273756570966
Recall Score=0.22668112798264642
AUC Score=0.5568479000528064
TN:17774,FP:2264,FN:713,TP:209
sensitivity:0.22668112798264642,specicity:0.8870146721229664,fpr:0.11298532787703364
F1 Score:0.123122238586l561
log loss:4.905713149629345
              precision    recall  f1-score   support

         0.0       0.96      0.89      0.92     20038
         1.0       0.08      0.23      0.12       922

    accuracy                           0.86     20960
   macro avg       0.52      0.56      0.52     20960
weighted avg       0.92      0.86      0.89     20960
```



XG Boost

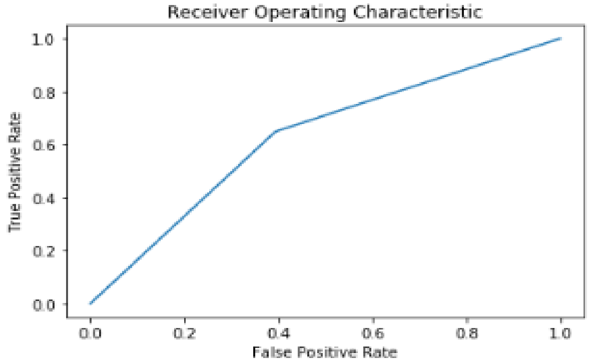
```
Accuracy Score=0.5919847328244274
Precision Score=0.067754362111942
Recall Score=0.648590021691974
AUC Score=0.6189850996772077
TN:11810,FP:8228,FN:324,TP:598
sensitivity:0.648590021691974,specicity:0.589380177662441 3,fpr:0.4106198223375586
F1 Score:0.12269183422240461
log loss:14.092661966456673
              precision    recall  f1-score   support

         0.0       0.97      0.59      0.73     20038
         1.0       0.07      0.65      0.12       922

    accuracy                           0.59     20960
   macro avg       0.52      0.62      0.43     20960
weighted avg       0.93      0.59      0.71     20960
```



Random Forest Classifier

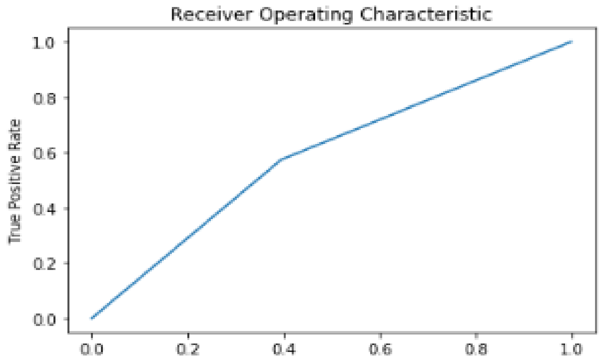
- Now lets create model with full data without the WOE replaced values

```
Accuracy Score=0.607824427480916
Precision Score=0.06754685842272781
Recall Score=0.6489241223103058
AUC Score=0.6274704787474227
TN:12167,FP:7910,FN:310,TP:573
sensitivity:0.6489241223103058,speciticity:0.6060168351845395,fpr:0.3939831648154605
F1 Score:0.12235746316463807
log loss:13.54556616325566
              precision    recall  f1-score   support

         0.0       0.98      0.61      0.75     20077
         1.0       0.07      0.65      0.12       883

    accuracy                           0.61     20960
   macro avg       0.52      0.63      0.43     20960
weighted avg       0.94      0.61      0.72     20960
```



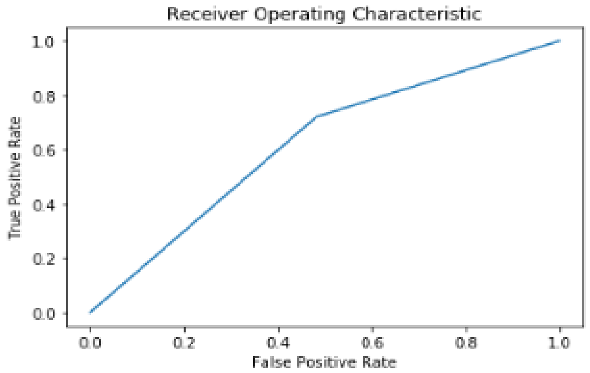
Logistic Regression with incremental PCA

```
Accuracy Score=0.6034351145038168
Precision Score=0.06004974535117849
Recall Score=0.5741789354473387
AUC Score=0.5894503782182651
TN:12141,FP:7936,FN:376,TP:507
sensitivity:0.5741789354473387,speciticity:0.6047218209891916,fpr:0.3952781790108836
F1 Score:0.10872828651082994
log loss:13.697168654566013
              precision    recall  f1-score   support

         0.0       0.97      0.60      0.74     20077
         1.0       0.06      0.57      0.11       883

    accuracy                           0.60     20960
   macro avg       0.52      0.59      0.43     20960
weighted avg       0.93      0.60      0.72     20960
```



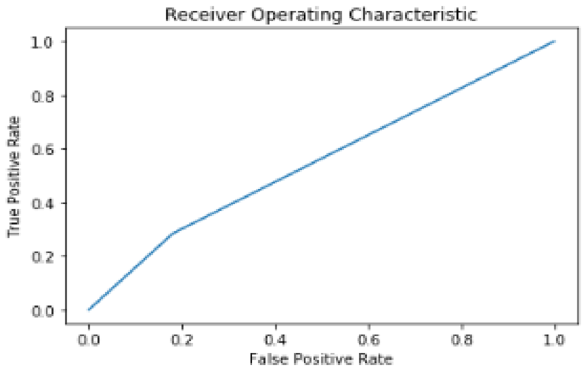
Decision Tree Classifier

```
Accuracy Score=0.5263358778625954
Precision Score=0.06164582727537075
Recall Score=0.7202718006795017
AUC Score=0.6190391229327677
TN:10396,FP:9681,FN:247,TP:636
sensitivity:0.7202718006795017,specificity:0.5178064451860338,fpr:0.48219355481396625
F1 Score:0.11357142857142859
log loss:16.360148518673423
              precision    recall  f1-score   support

         0.0       0.98      0.52      0.68     20077
         1.0       0.06      0.72      0.11       883

    accuracy                           0.53     20960
   macro avg       0.52      0.62      0.40     20960
weighted avg       0.94      0.53      0.65     20960
```



RandomForestClassifier(Bagging technique)

```
Accuracy Score=0.7967557251908397
Precision Score=0.06470739881412735
Recall Score=0.2842582106455266
AUC Score=0.5517769610781051
TN:16449,FP:3628,FN:632,TP:251
sensitivity:0.2842582106455266,specificity:0.8192957115106839,fpr:0.18070428848931613
F1 Score:0.10541789164216715
log loss:7.019946964780356
              precision    recall  f1-score   support

         0.0       0.96      0.82      0.89     20077
         1.0       0.06      0.28      0.11       883

    accuracy                           0.80     20960
   macro avg       0.51      0.55      0.50     20960
weighted avg       0.93      0.80      0.85     20960
```



XG Boost

- Out of these we see Logistic Regression seems to be the best model.

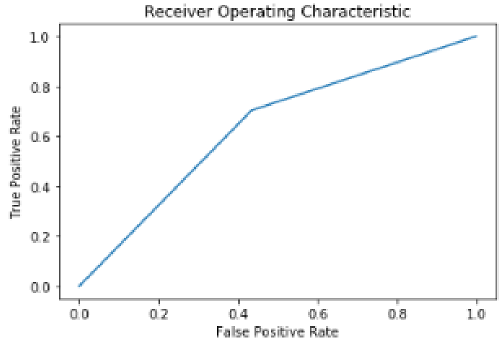
Accuracy Score=0.	0.627099
sensitivity	0.602492
specificity	0.628182

- Till now we have made models on
- Demographics Data
- Full Data Without WOE Replaced
- High IV Values

- We will move on to WOE replaced values. WOE has following advantages
  - It can treat outliers.
  - It can handle missing values as missing values can be binned separately.
  - Since WOE Transformation handles categorical variable so there is no need for dummy variables.
  - WoE transformation helps you to build strict linear relationship with log odds.

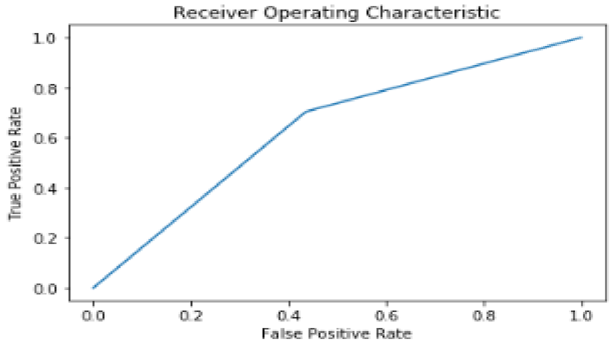
- Now lets create model with full data with the WOE replaced values
- As we are now working with WOE replaced dataset so there is no need for encoding and it also help to deal with the outliers

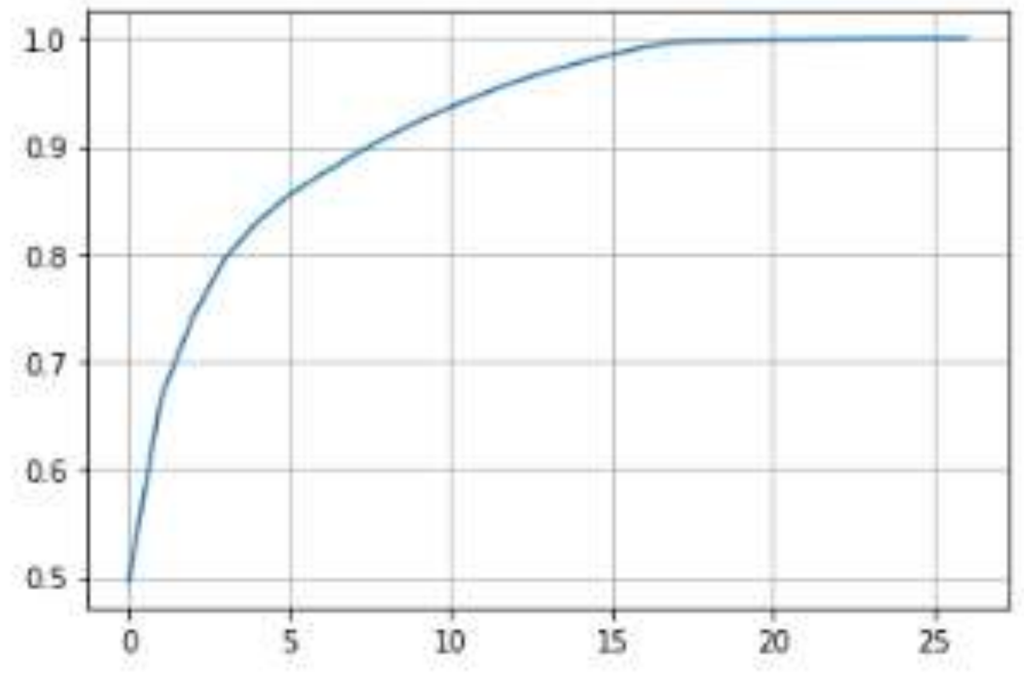
```
Accuracy Score=0.5712786259541984
Precision Score=0.06613869863013698
Recall Score=0.7038724373576309
AUC Score=0.6346769815510394
TN:11356,FP:8726,FN:260,TP:618
sensitivity:0.7038724373576309,specificity:0.565481525744478,fpr:0.4345184742555223
F1 Score:0.12091567207982781
log loss:14.807844559725348
              precision    recall  f1-score   support

         0.0       0.98      0.57      0.72     20082
         1.0       0.07      0.70      0.12       878

    accuracy                           0.57     20960
   macro avg       0.52      0.63      0.42     20960
weighted avg       0.94      0.57      0.69     20960
```



Basic Logistic Regression Model

```
Accuracy Score=0.5695610687022901
Precision Score=0.06588486140724947
Recall Score=0.7038724373576309
AUC Score=0.633780656483815
TN:11320,FP:8762,FN:260,TP:618
sensitivity:0.7038724373576309,specificity:0.563688875609999,fpr:0.43631112439000097
F1 Score:0.12049132384480406
log loss:14.867168258948835
              precision    recall  f1-score   support

         0.0       0.98      0.56      0.72     20082
         1.0       0.07      0.70      0.12       878

    accuracy                           0.57     20960
   macro avg       0.52      0.63      0.42     20960
weighted avg       0.94      0.57      0.69     20960
```

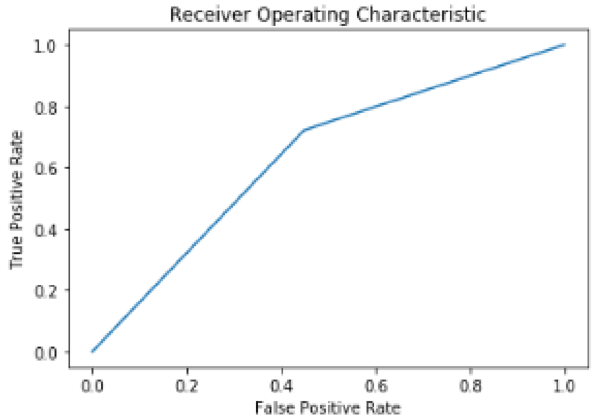




Logistic Regression with Incremental PCA

We can see 5 PC,s are explaining a variance of 85%

```
Accuracy Score=0.5581106870229008
Precision Score=0.06568586821384169
Recall Score=0.7220956719817767
AUC Score=0.6365184066511812
TN:11064,FP:9018,FN:244,TP:634
sensitivity:0.7220956719817767,specicity:0.5509411413205856,fpr:0.4490588586794144
F1 Score:0.12041785375118708
log loss:15.262660197485125
              precision    recall  f1-score   support

         0.0       0.98      0.55      0.70     20082
         1.0       0.07      0.72      0.12       878

    accuracy                           0.56     20960
   macro avg       0.52      0.64      0.41     20960
weighted avg       0.94      0.56      0.68     20960
```



Random Forest Classifier

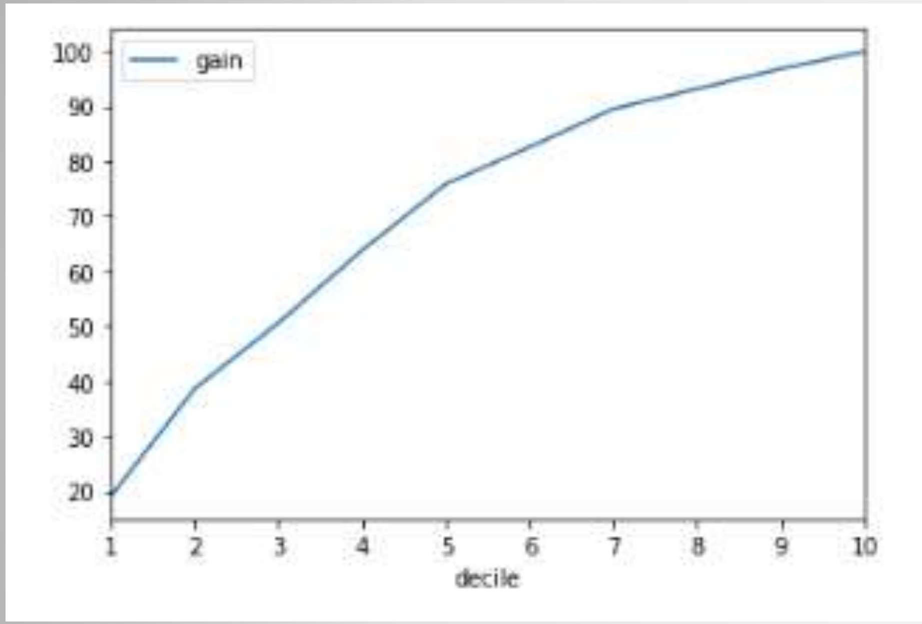
Final Model

- Based on our analysis on various models, we select Logistic Regression with Incremental PCA on the WOE replaced dataset as our Final Model.

Accuracy Score	Precision Score	AUC Score	Sensitivity	Specificity
0.57	0.07	0.63	0.70	0.56

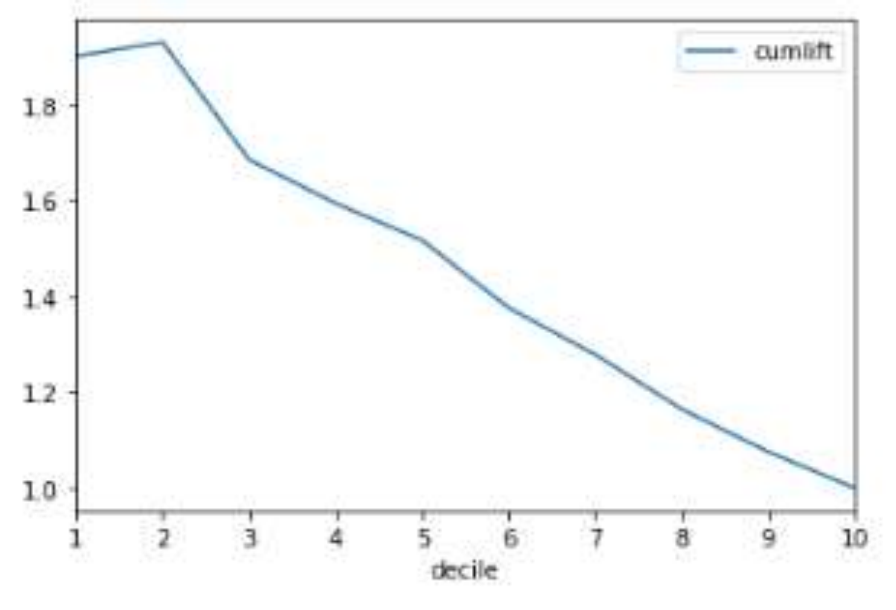
- TN:11320,FP:8762,FN:260,TP:618

Gain Chart



Within first 5 deciles as per the model we are able to predict 75% of defaulters correctly.

Lift Chart



A lift of 2 times is achieved with the model within first 2 deciles compared to random mode

Application ScoreCard

- 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))
- **Summary of application score cardvalues:**
- •Scores range from 294.28 to 369.0 for applicants.
- •Higher scores indicate less risk for defaulting

CUTOFF SCORE FOR ACCEPTING OR REJECTING AN APPLICATION

- Cutoff selected for probability of default for logistic regression model was 0.46
- CUTOFF_SCORE= 400 + (slope * (ln((1-0.50)/0.50) -ln(10)))
- CUTOFF SCORE is equal to **333.57**
- No.ofapplicants above score 338.18 and thus their credit card application will be accepted as per our model is 30871
- No.ofapplicants below score 338.18 and thus their credit card application will not be accepted as per our model is 38993

FINANCIAL BENEFITS OF THE MODEL

- 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.

	Reference	
Prediction	0	1
0	38089	904
1	28828	2043

- **Revenue Loss :** Occurs when good customers are identified as bad and credit card application is rejected.
- No of candidates rejected by the model who didn't default –28828.
- Total No of candidates who didn't default –66927
- % of good candidates rejected by our model –43.07%
- About 31.38% of the non defaulting customers are rejected which resulted in revenue loss.

- **Credit Loss Saved :** The candidates who have been selected by the bank and have defaulted are responsible for the credit loss to the bank.
- % of candidates approved and then defaulted when model was not used = 4.2%
- % of candidates approved and then defaulted when model was used = 904/69799 = 1.3%
- Credit loss saved => 4.2 –1.3 = 2.9%