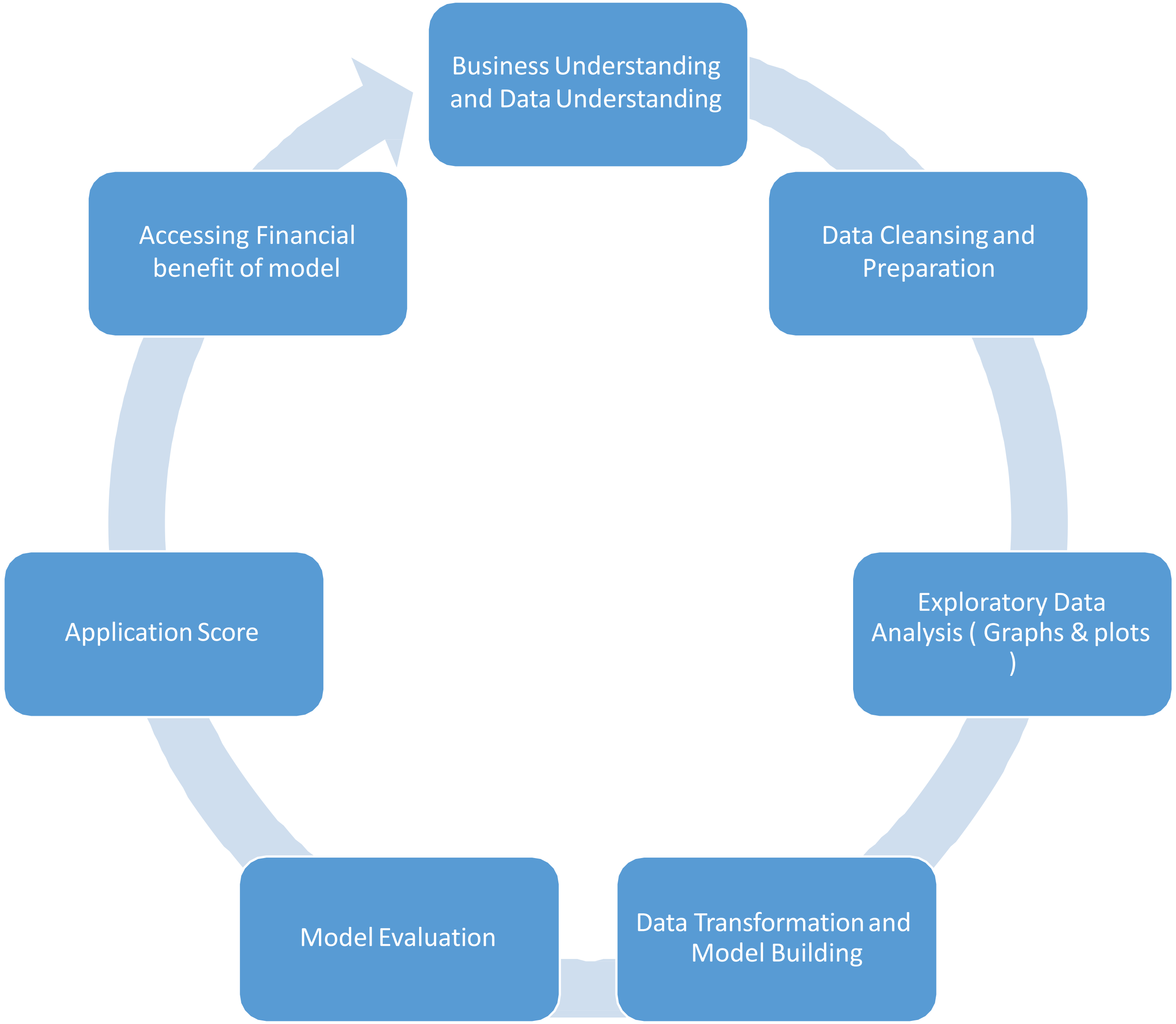
CredX - Credit Default Analysis

Business Understanding

Objective:
Identifying the right customers using predictive models and determining the factors affecting credit risk. Also *c*reating strategies to mitigate them.

Problem Statement:
Credx is a leading credit card provider that gets thousands of credit card applicants every year. But in the past few years, it has experienced an increase in credit loss due to increase in defaults. . The CEO believes that the best strategy to mitigate credit risk is to acquire the right customers.

Solution Approach:
This is a binary supervised classification problem. We have built models such as Logistic regression, Random forest and AdaBoost classifier to identify the customers who are at a risk of defaulting if offered a credit card.



Business Understanding and Data Understanding

Data Cleansing and Preparation

Exploratory Data Analysis (Graphs & plots)

Data Transformation and Model Building

Model Evaluation

Application Score

Accessing Financial benefit of model

WOE and IV

- 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.

- It can treat outliers. Suppose you have a continuous variable such as annual

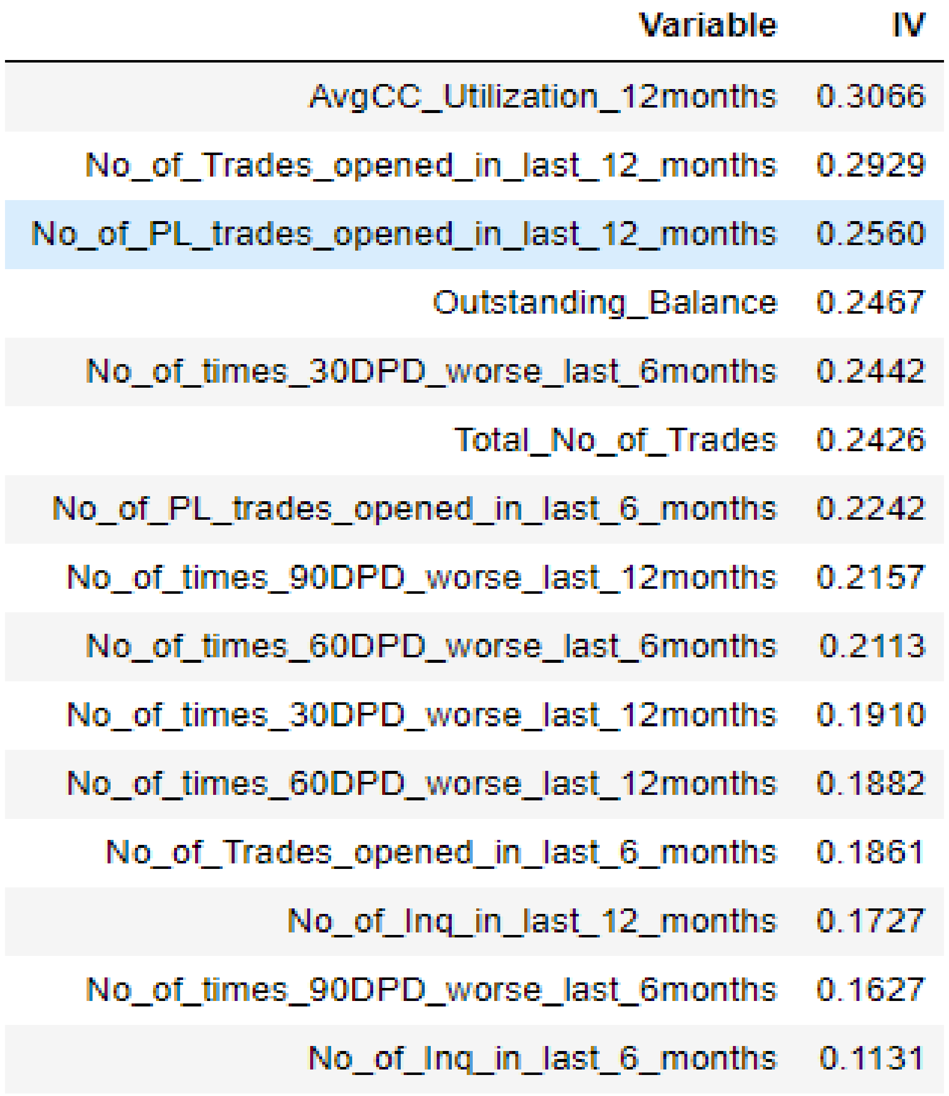
salary and extreme values are more than 500 million dollars. These values would

be grouped to a class of (let's say 250-500 million dollars). Later, instead of using

the raw values, we would be using WOE scores of each classes.

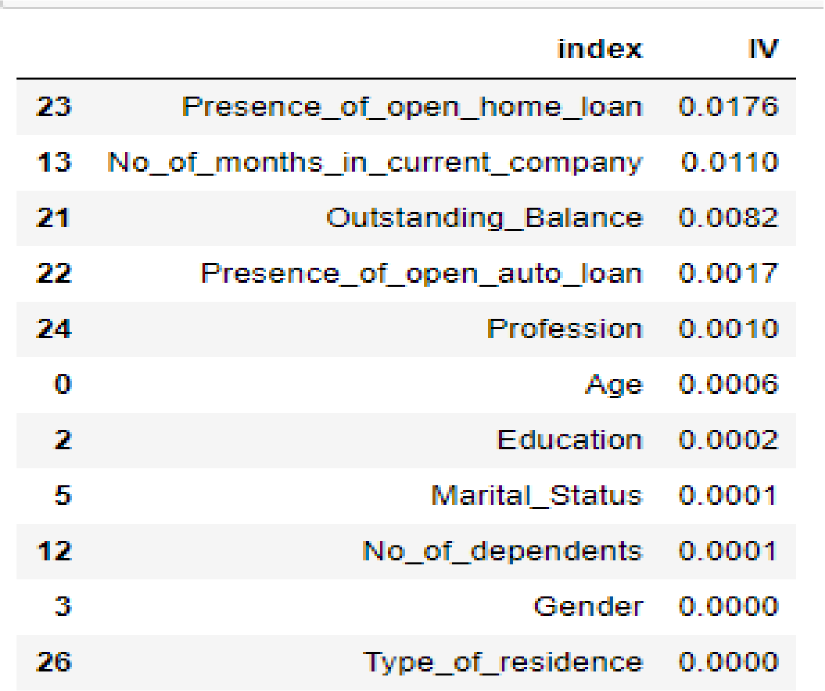
- It can handle missing values as missing values can be binned separately.
- Since WOE Transformation handles categorical variable so there is no need for dummy variables.
- Woe transformation helps you to build strict linear relationship with log odds.
- Information value is one of the most useful technique to select important variables in a predictive model. It helps to rank variables on the basis of their importance.

Variables with good predictive power

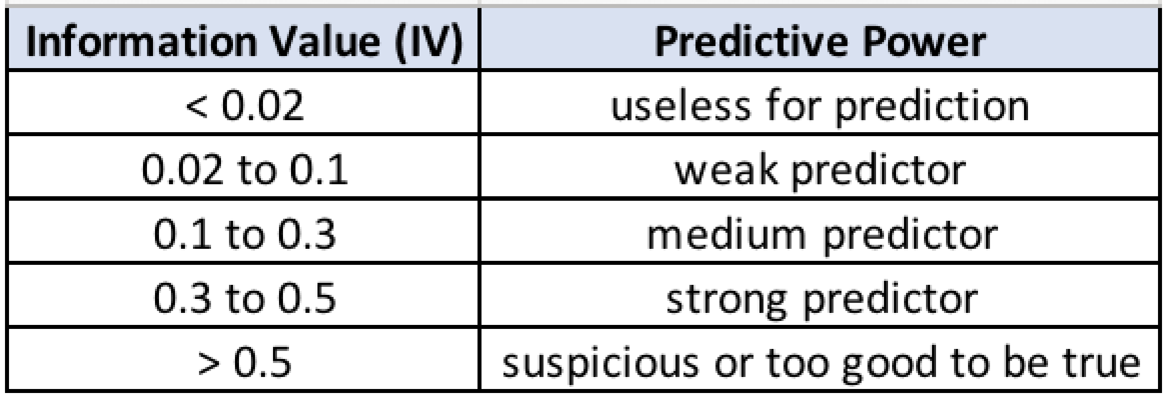


Variable	IV
AvgCC_Utilization_12months	0.3066
No_of_Trades_opened_in_last_12_months	0.2929
No_of_PL_trades_opened_in_last_12_months	0.2560
Outstanding_Balance	0.2467
No_of_times_30DPD_worse_last_6months	0.2442
Total_No_of_Trades	0.2426
No_of_PL_trades_opened_in_last_6_months	0.2242
No_of_times_90DPD_worse_last_12months	0.2157
No_of_times_60DPD_worse_last_6months	0.2113
No_of_times_30DPD_worse_last_12months	0.1910
No_of_times_60DPD_worse_last_12months	0.1882
No_of_Trades_opened_in_last_6_months	0.1861
No_of_Inq_in_last_12_months	0.1727
No_of_times_90DPD_worse_last_6months	0.1627
No_of_Inq_in_last_6_months	0.1131

Variables with useless predictive power



	index	IV
23	Presence_of_open_home_loan	0.0176
13	No_of_months_in_current_company	0.0110
21	Outstanding_Balance	0.0082
22	Presence_of_open_auto_loan	0.0017
24	Profession	0.0010
0	Age	0.0006
2	Education	0.0002
5	Marital_Status	0.0001
12	No_of_dependents	0.0001
3	Gender	0.0000
26	Type_of_residence	0.0000

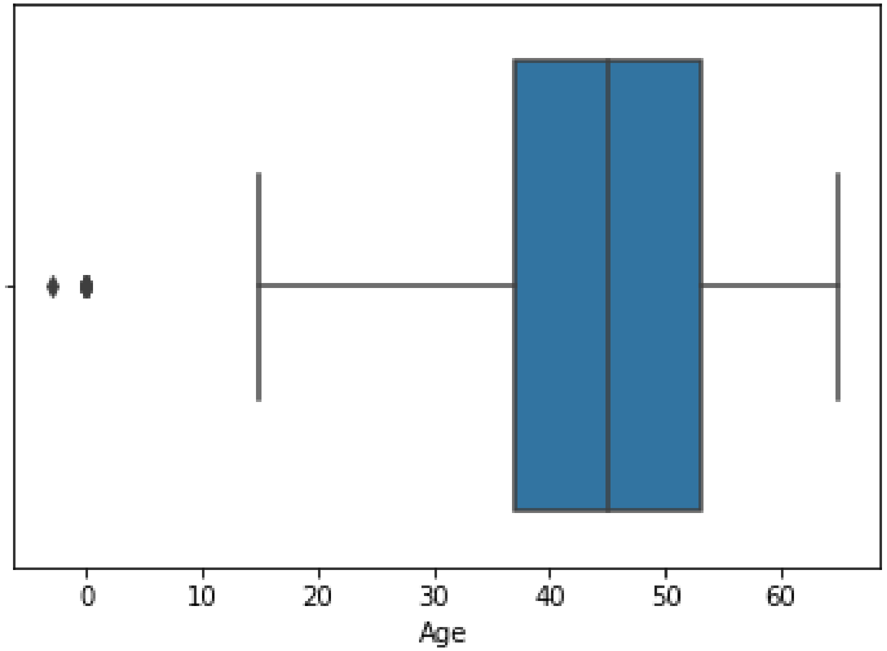


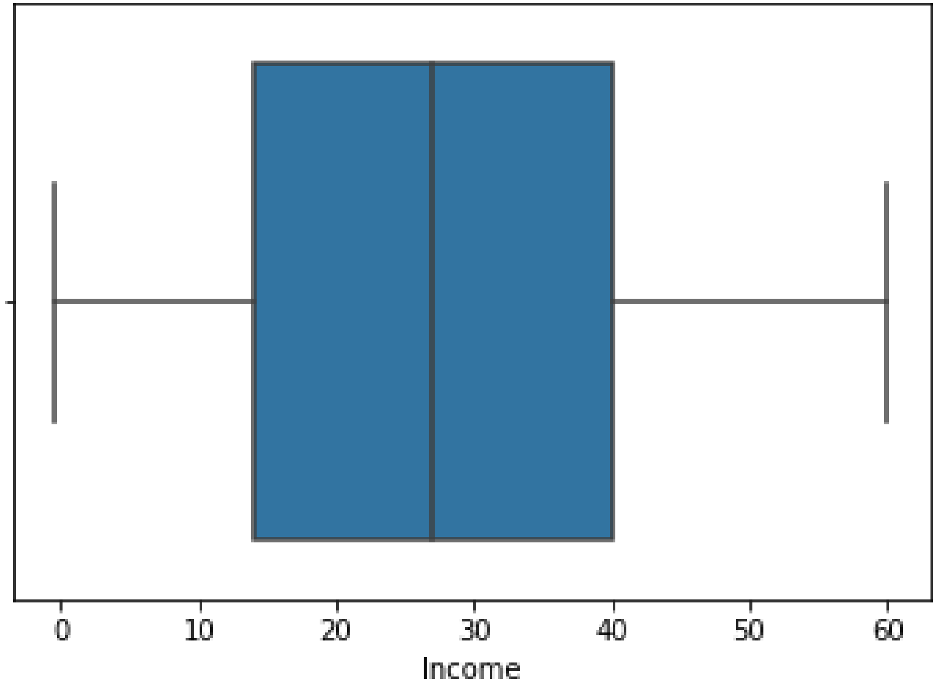
Information Value (IV)	Predictive Power
< 0.02	useless for prediction
0.02 to 0.1	weak predictor
0.1 to 0.3	medium predictor
0.3 to 0.5	strong predictor
> 0.5	suspicious or too good to be true

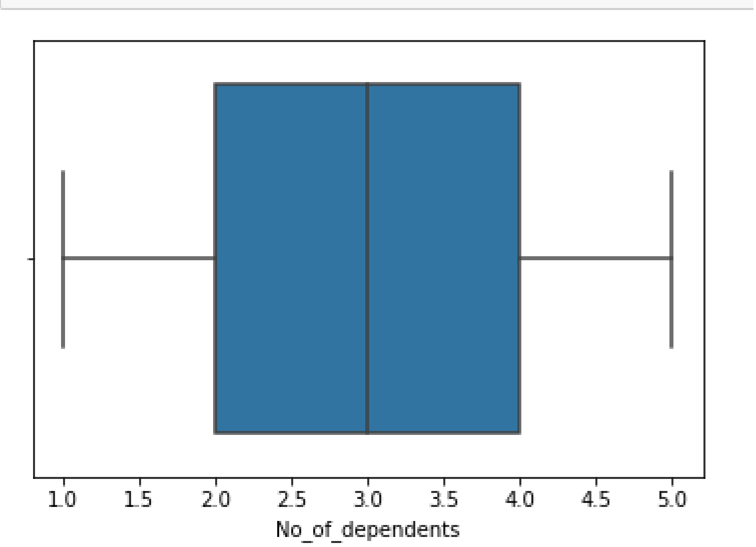
Below are top predictor variables obtained from IV analysis:

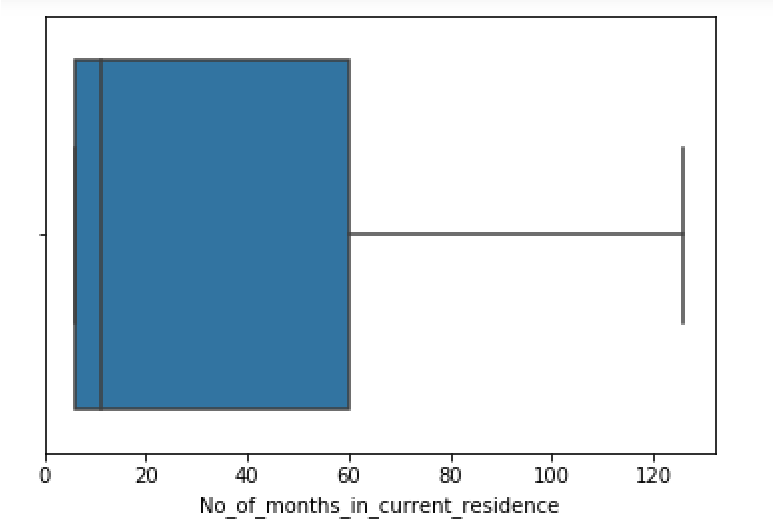
- AvgCC_Utilization_12months
- No_of_Trades_opened_in_last_12_months
- No_of_PL_trades_opened_in_last_12_months
- Outstanding Balance
- No_of_times_30DPD_worse_last_6months
- Total_No_of_Trades
- No_of_PL_trades_opened_in_last_6_months
- No_of_times_90DPD_worse_last_12months
- No_of_times_60DPD_worse_last_6months
- No_of_times_30DPD_worse_last_12months
- No_of_times_60DPD_worse_last_12months

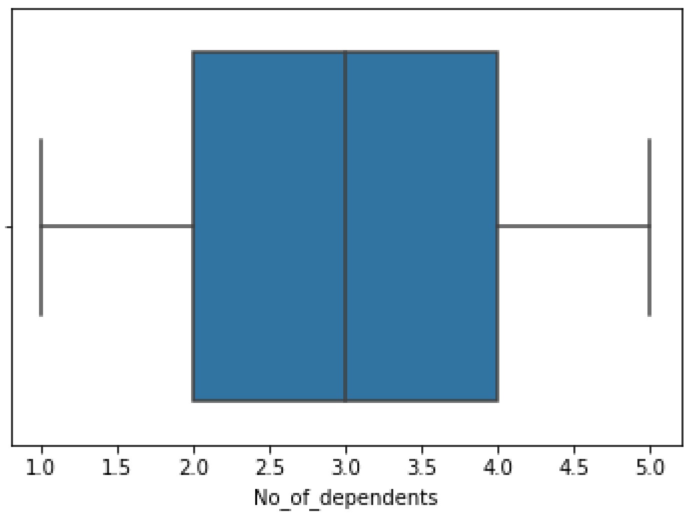
UNIVARIATE ANALYSIS

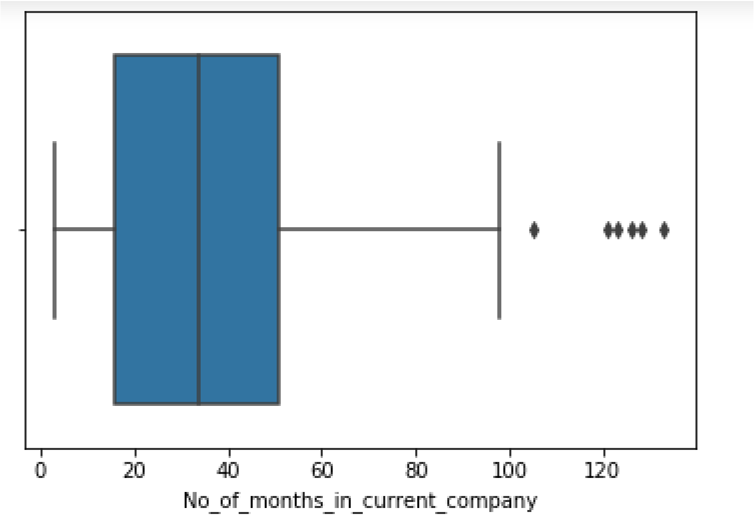


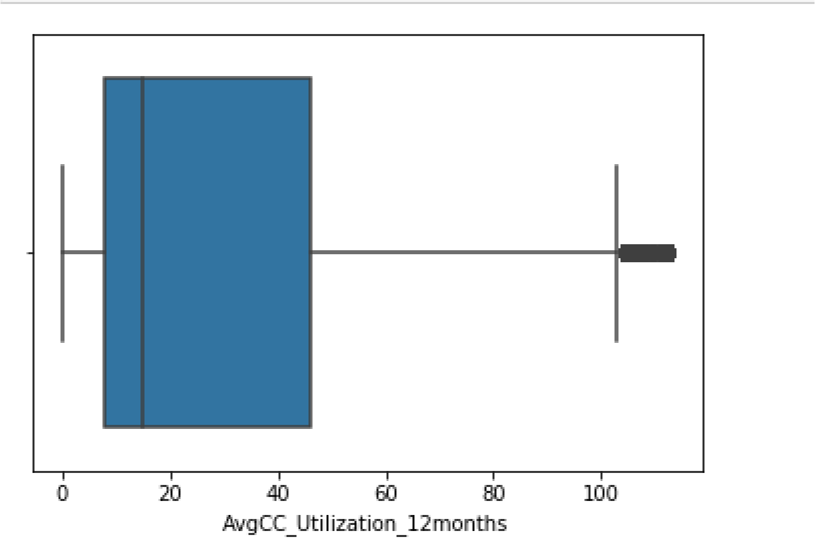


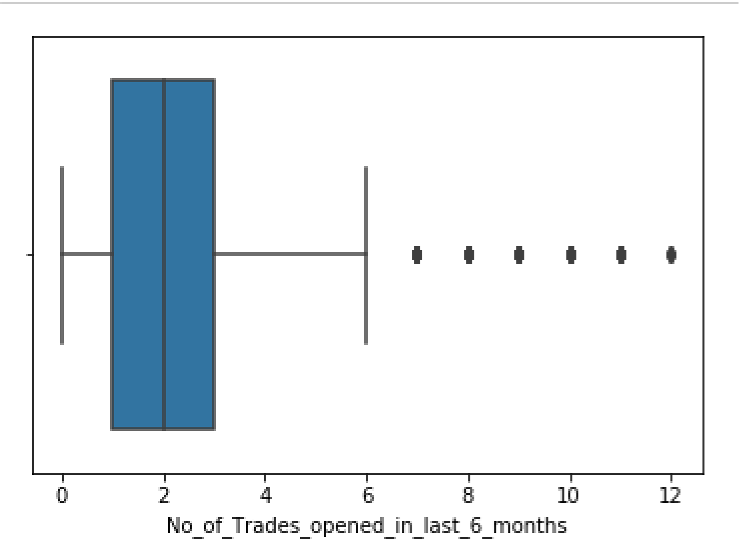

1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
No_of_dependents

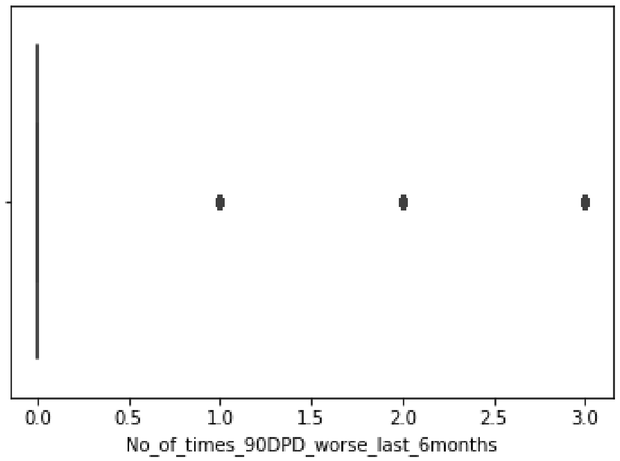

0
20
40
60
80
100
120
No_of_months_in_current_residence

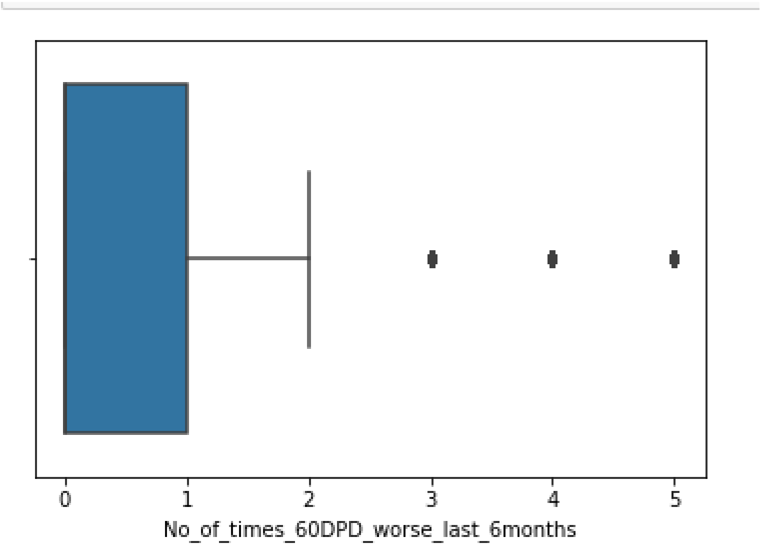

1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
No_of_dependents

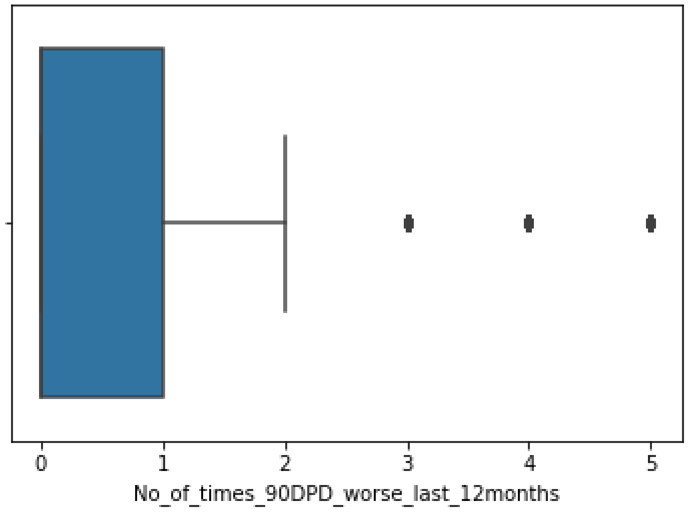

0
20
40
60
80
100
120
No_of_months_in_current_company

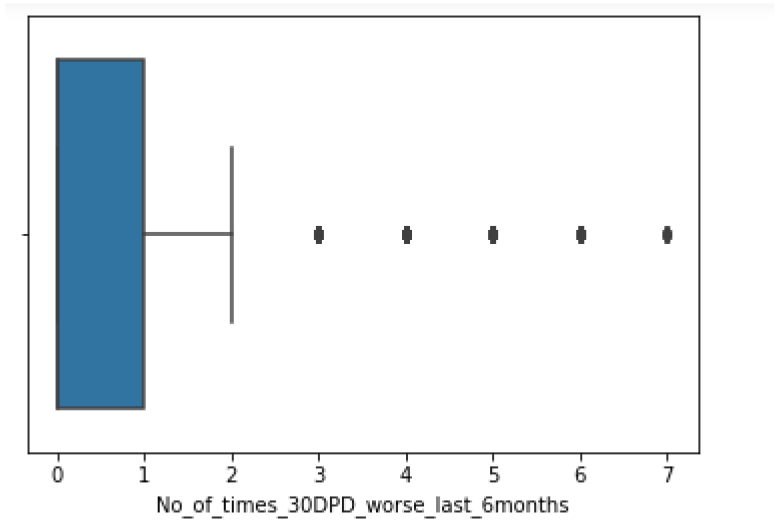

0
20
40
60
80
100
AvgCC_Utilization_12months

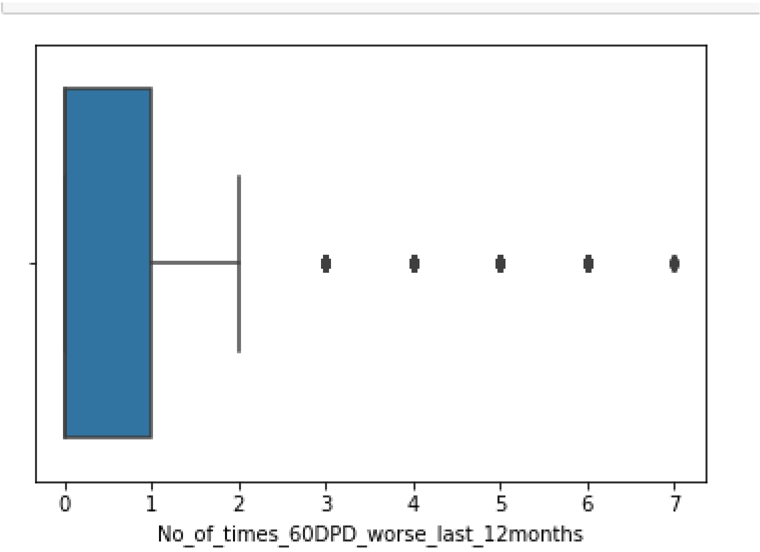

0
2
4
6
8
10
12
No_of_Trades_opened_in_last_6_months

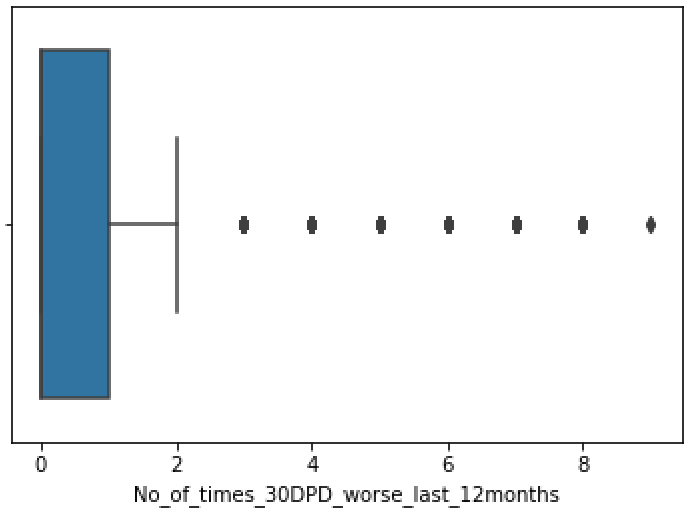

0.0
0.5
1.0
1.5
2.0
2.5
3.0
No_of_times_90DPD_worse_last_6months

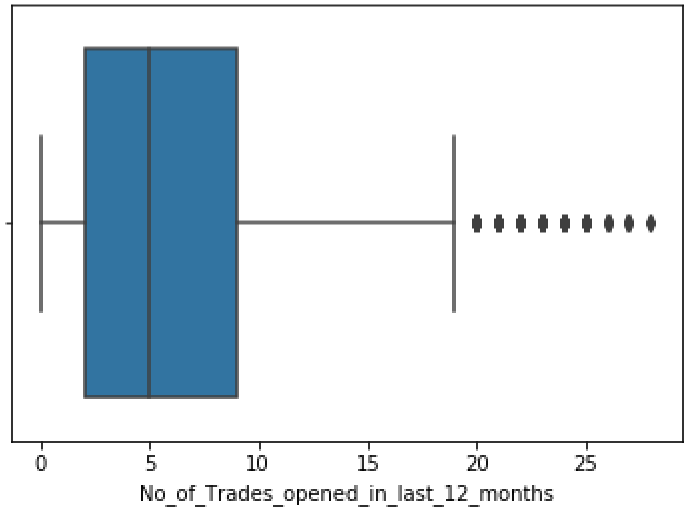

0
1
2
3
4
5
No_of_times_60DPD_worse_last_6months

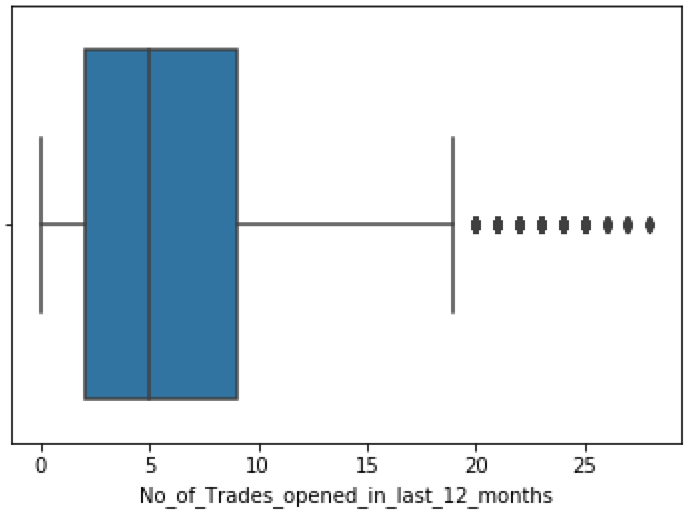

0
1
2
3
4
5
No_of_times_90DPD_worse_last_12months

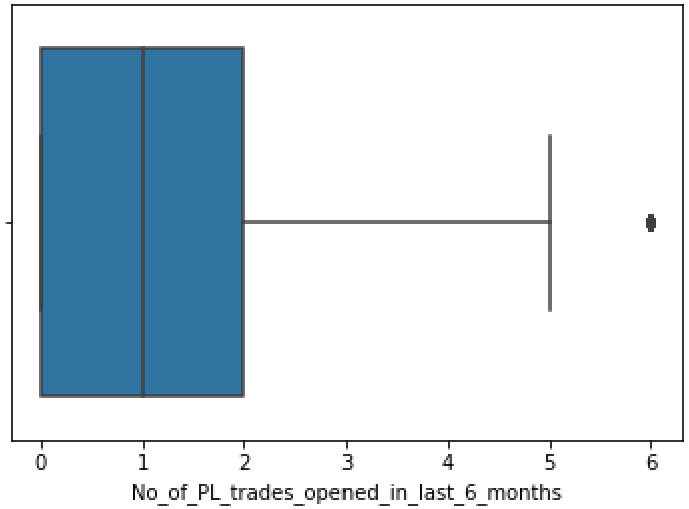

0
1
2
3
4
5
6
7
No_of_times_30DPD_worse_last_6months

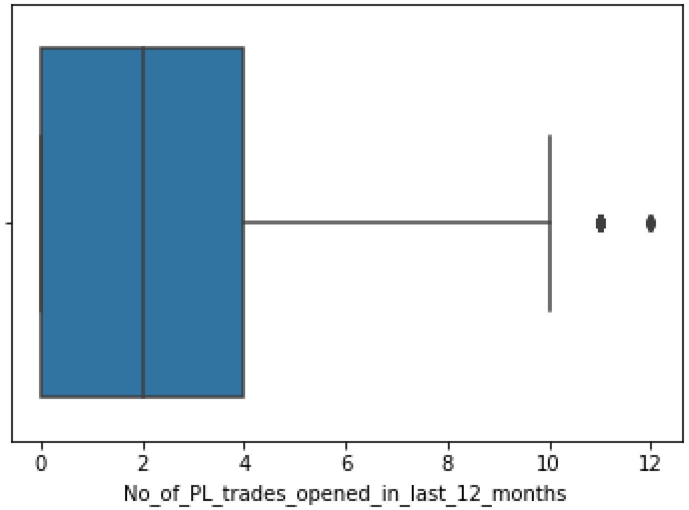

0
1
2
3
4
5
6
7
No_of_times_60DPD_worse_last_12months

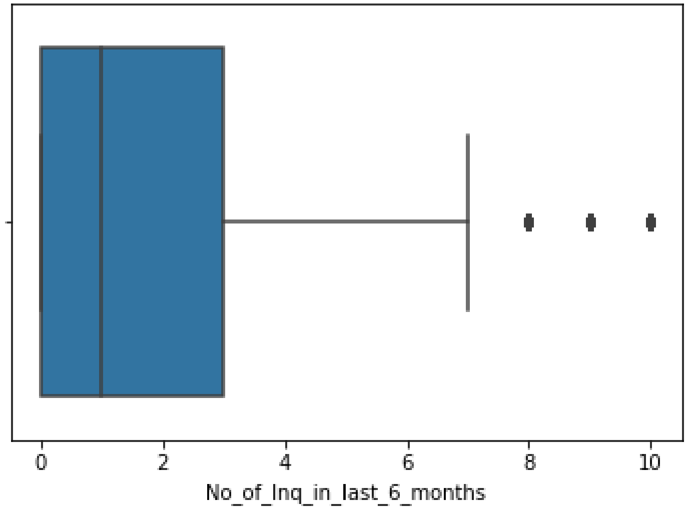

0
2
4
6
8
No_of_times_30DPD_worse_last_12months

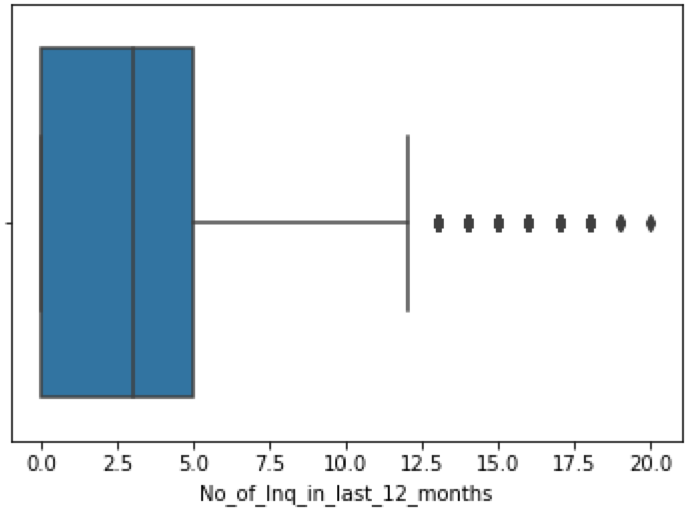

0
5
10
15
20
25
No_of_Trades_opened_in_last_12_months


0
5
10
15
20
25
No_of_Trades_opened_in_last_12_months

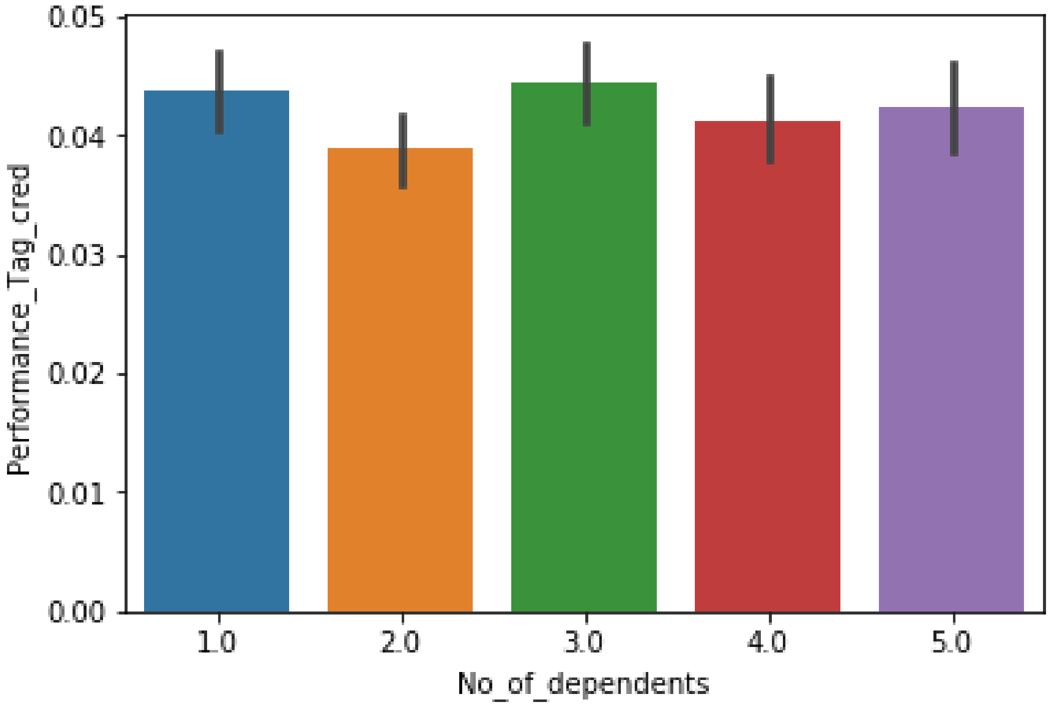

0
1
2
3
4
5
6
No_of_PL_trades_opened_in_last_6_months

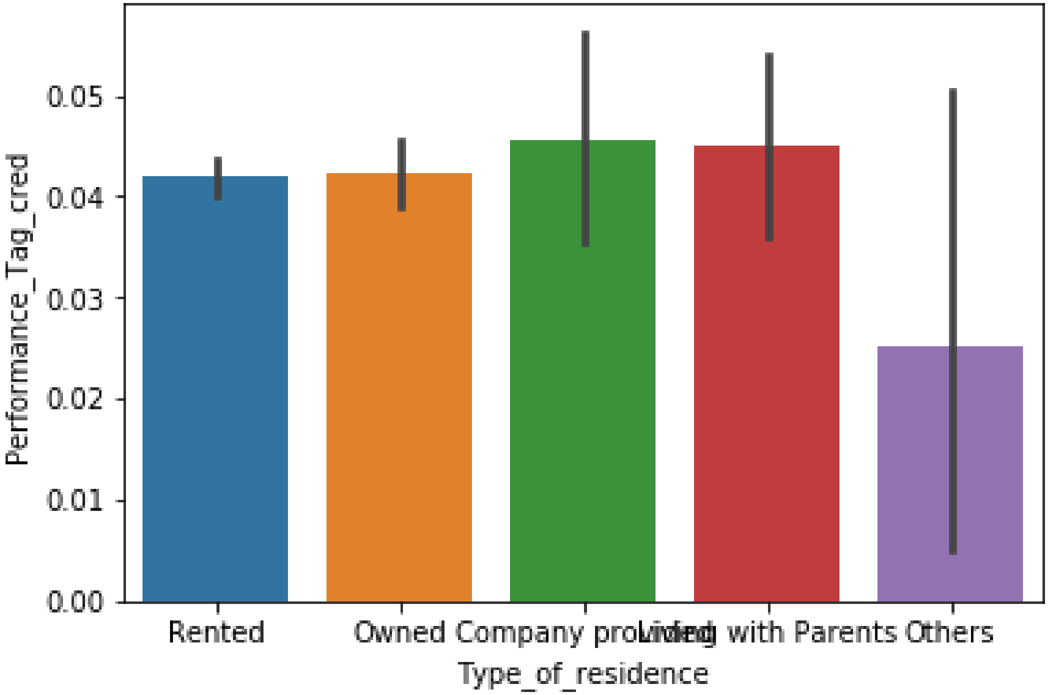

0
2
4
6
8
10
12
No_of_PL_trades_opened_in_last_12_months

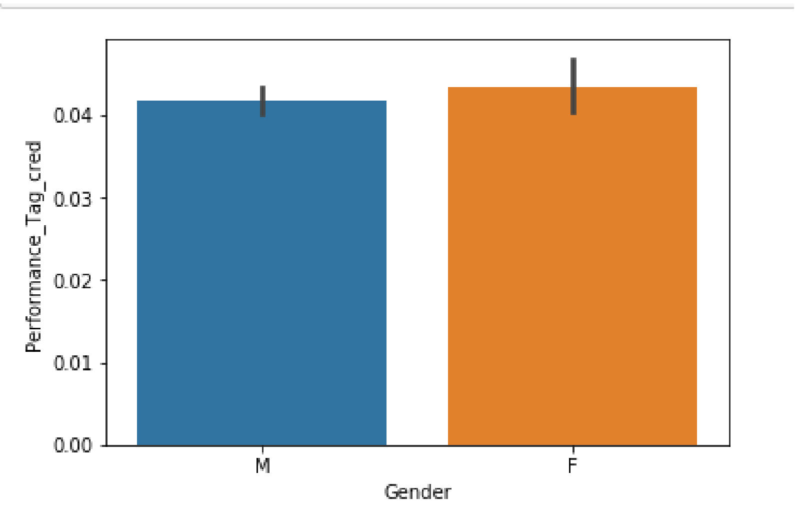

0
2
4
6
8
10
No_of_Inq_in_last_6_months

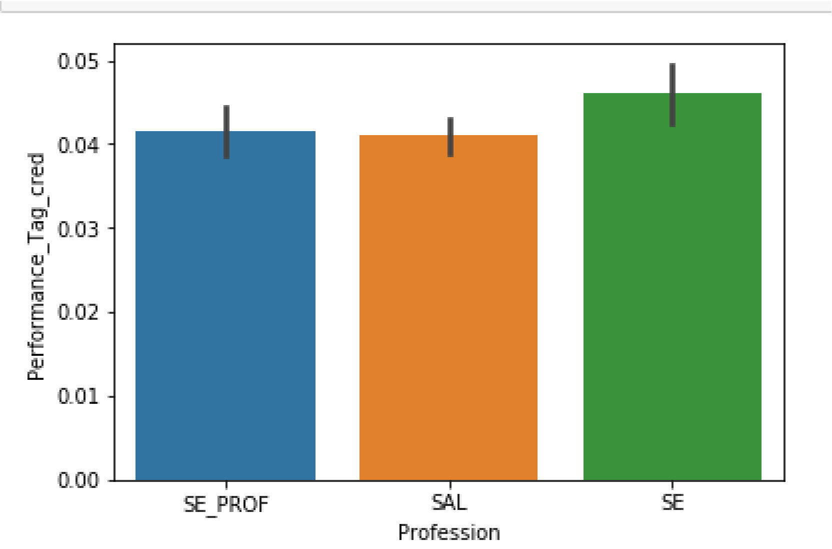

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
No_of_Inq_in_last_12_months

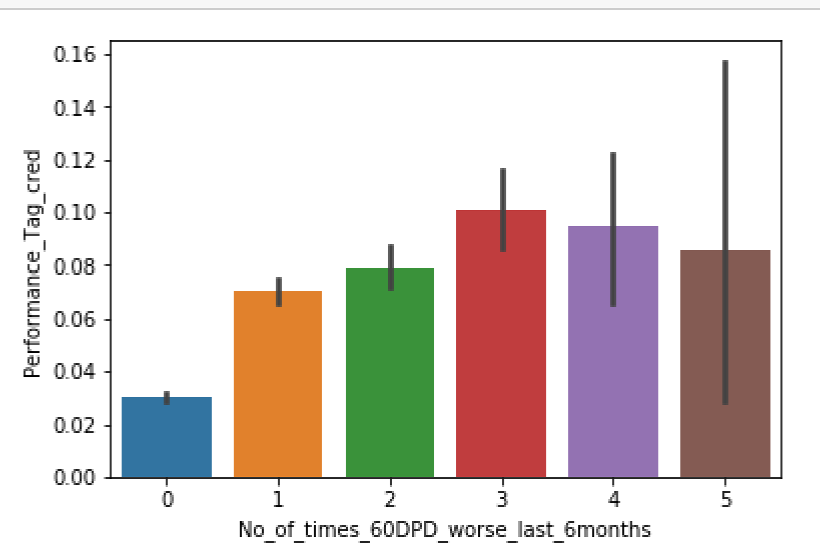
BIVARIATE ANALYSIS

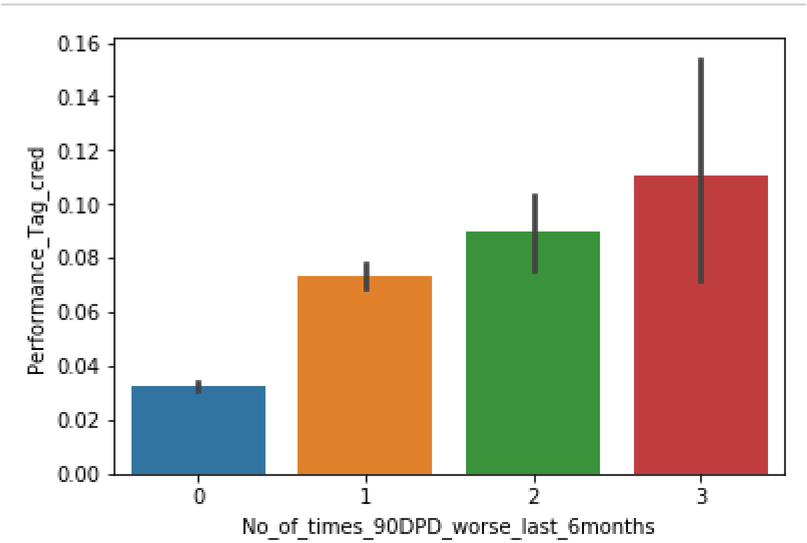


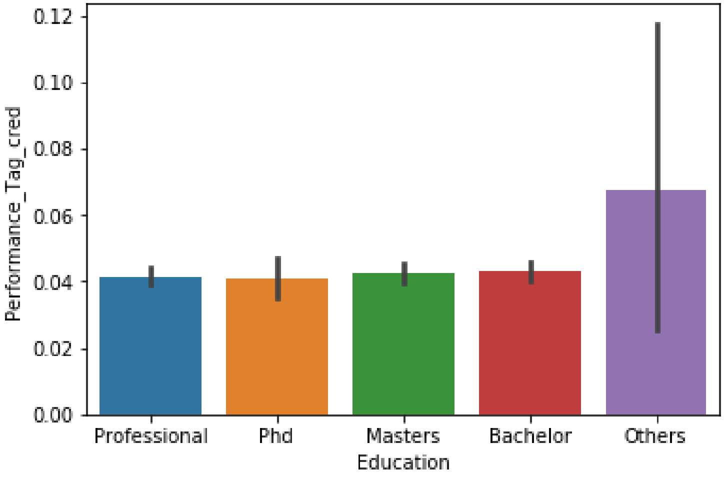


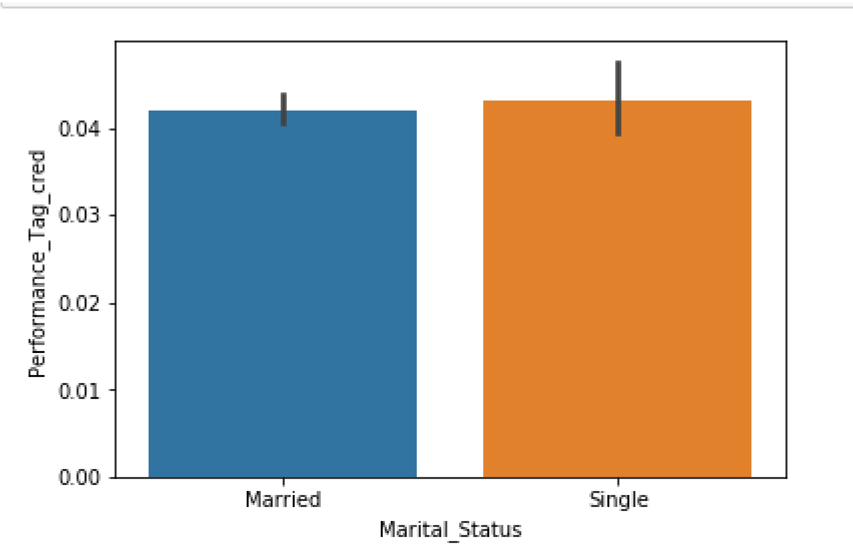

Performance_Tag_cred
0.00
0.01
0.02
0.03
0.04
M
F
Gender

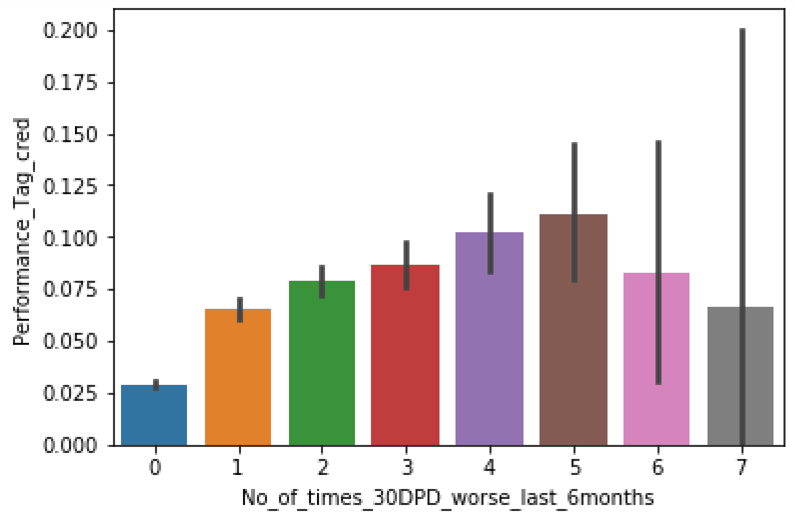

Performance_Tag_cred
0.00
0.01
0.02
0.03
0.04
0.05
SE_PROF
SAL
SE
Profession

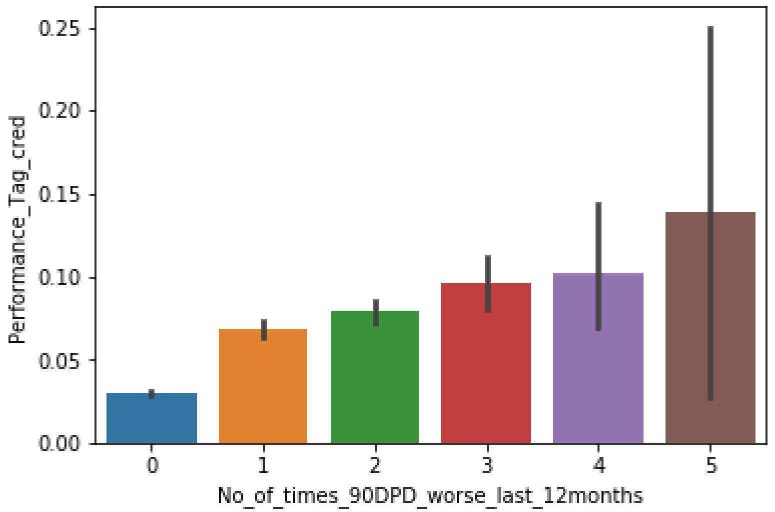

Performance_Tag_cred
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0
1
2
3
4
5
No_of_times_60DPD_worse_last_6months

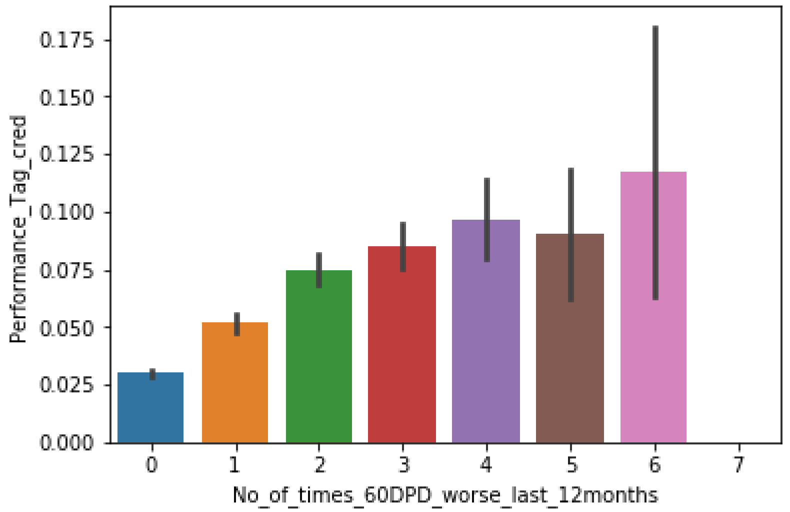

Performance_Tag_cred
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0
1
2
3
No_of_times_90DPD_worse_last_6months

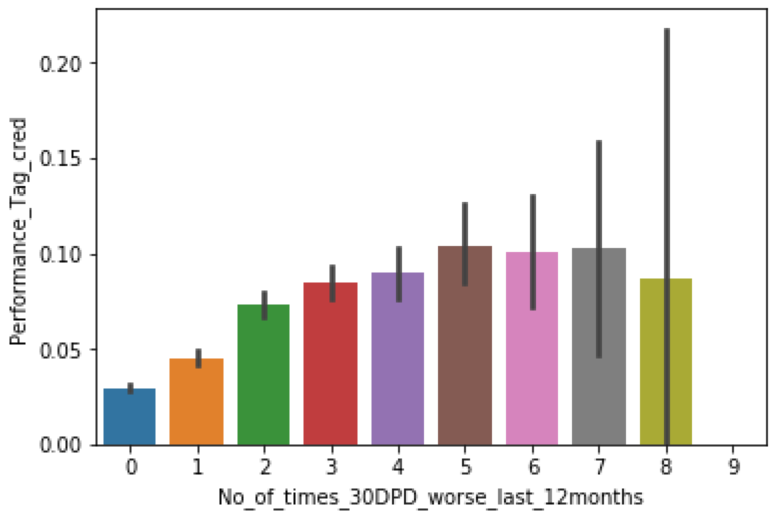

Performance_Tag_cred
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Professional
Phd
Masters
Bachelor
Others
Education


Performance_Tag_cred
0.00
0.01
0.02
0.03
0.04
Married
Single
Marital_Status

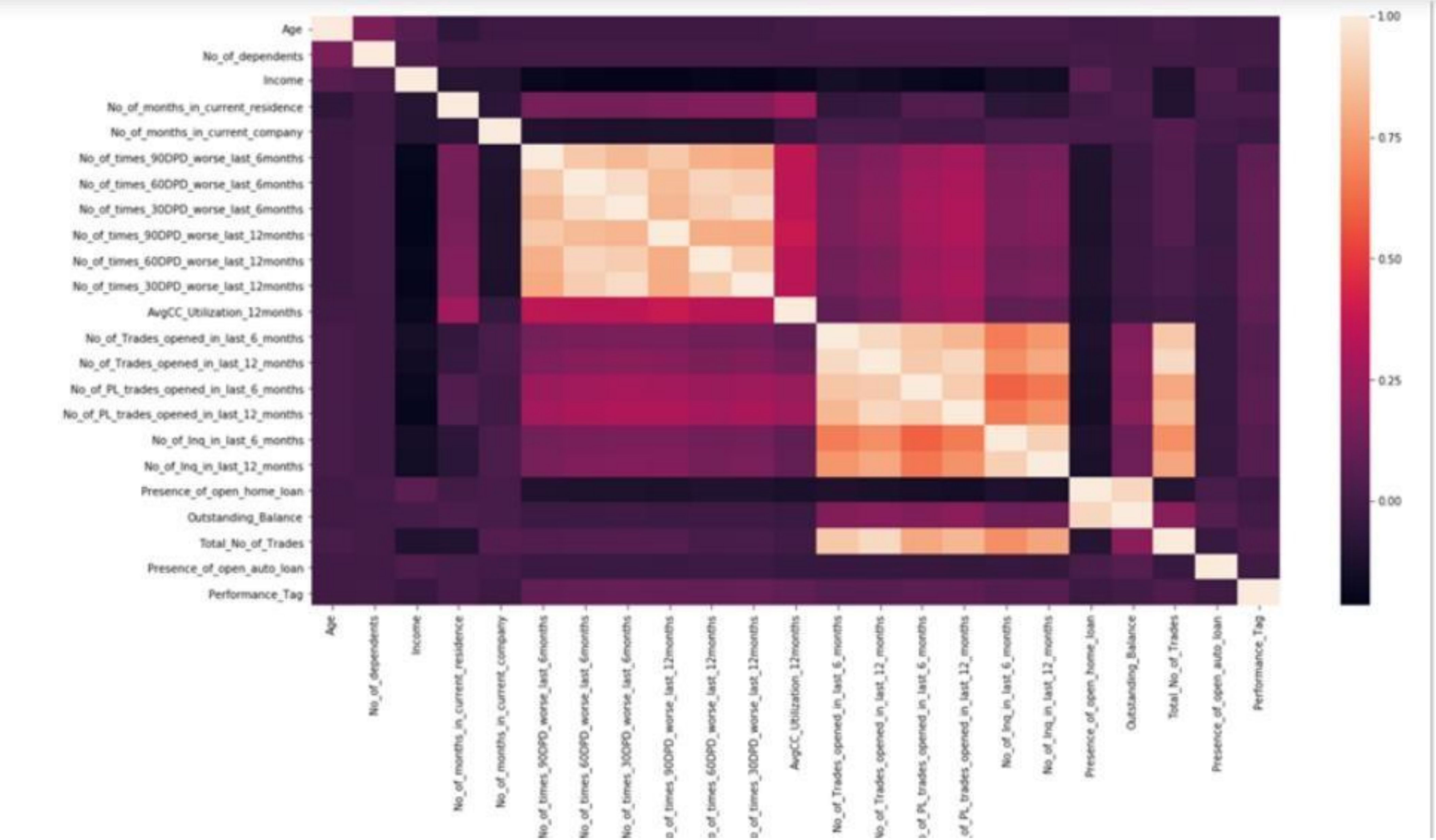

Performance_Tag_cred
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
1
2
3
4
5
6
7
No_of_times_30DPD_worse_last_6months


Performance_Tag_cred
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
No_of_times_90DPD_worse_last_12months


Performance_Tag_cred
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
1
2
3
4
5
6
7
No_of_times_60DPD_worse_last_12months


Performance_Tag_cred
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8
9
No_of_times_30DPD_worse_last_12months

Correlation Map



Observations from EDA

- People who uses credit limit up to 60k are likely to default
- People who does up to 5-6 times inquiries for credit card within 12 months are likely to default
- People who does up to 2-3 times inquiries for credit card within 6 months are likely to default
- Professionals and Master degree holder are more likely to default
- Salaried people are more likely to default
- People between age of 35 to 55 are more likely to default
- Males and People living in rented home are more likely to default
- People who had late payment for 90 days are less likely to default.

Observations from EDA

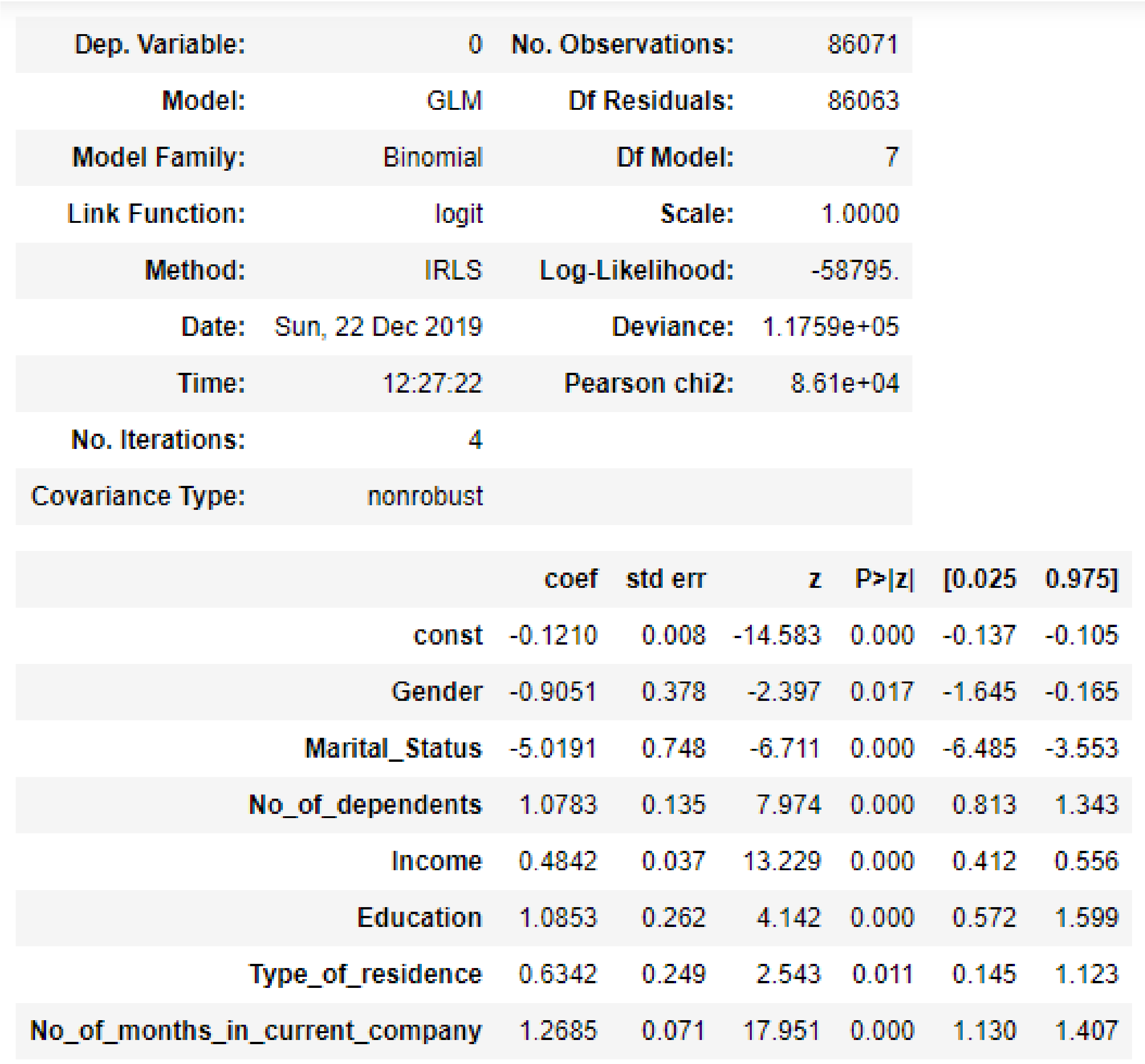
- *Median Age of applicants is 45 years and there are few outliers*
- *Median income of applicants is 27 and there are no outliers*
- Median No_of_months_in_current_residence is 10 and there are no outliers
- Median No_of_months_in_current_company is 34 and there are few outliers
- Median No_of_dependents is 3 and there are no outliers
- Median No_of_times_90DPD_worse_last_6months, No_of_times_60DPD_worse_last_6months, No_of_times_30DPD_worse_last_6months, No_of_times_90DPD_worse_last_12months is zero and there are few outliers
- Median number of avg credit card usage is 15 and there are few outliers

MODELLING

- **Data Sets chosen for model**
  - Demographics WoE transformed data set
  - Combined (Demographics and Credit Bureau) WoE transformed data set

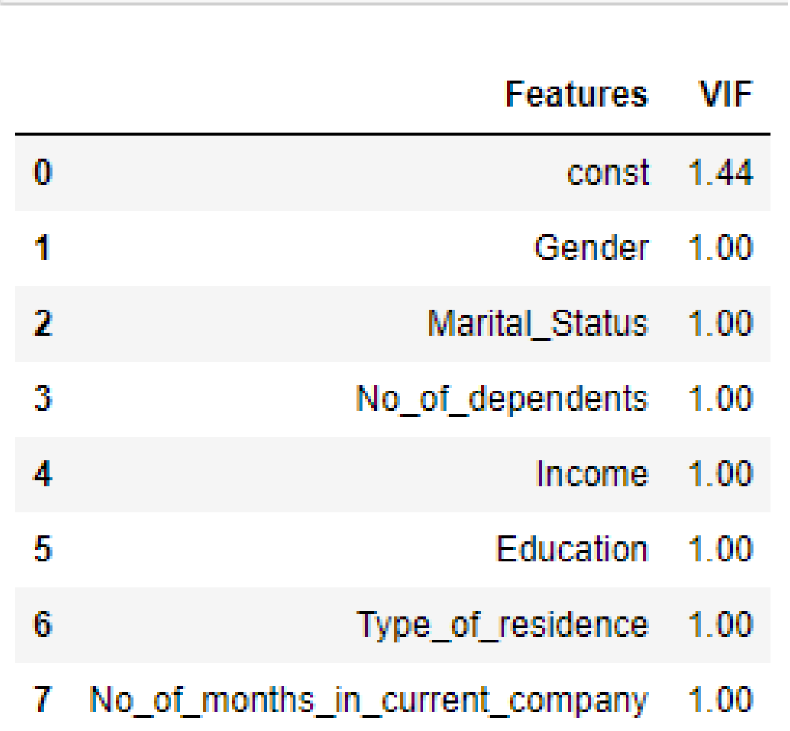
- **Models for each data set**
  - Logistic regression (with both above data sets)
  - Logistic regression with Grid Search (with Combined _woe dataset)
  - Random forest(with Combined _woe dataset)
  - AdaBoost Classifier(with Combined _woe dataset)
  - Hybrid Model(with Combined _woe dataset)

Logistic Regression



Dep. Variable:	0	No. Observations:	86071
Model:	GLM	Df Residuals:	86063
Model Family:	Binomial	Df Model:	7
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-58795.
Date:	Sun, 22 Dec 2019	Deviance:	1.1759e+05
Time:	12:27:22	Pearson chi2:	8.61e+04
No. Iterations:	4		
Covariance Type:	nonrobust		

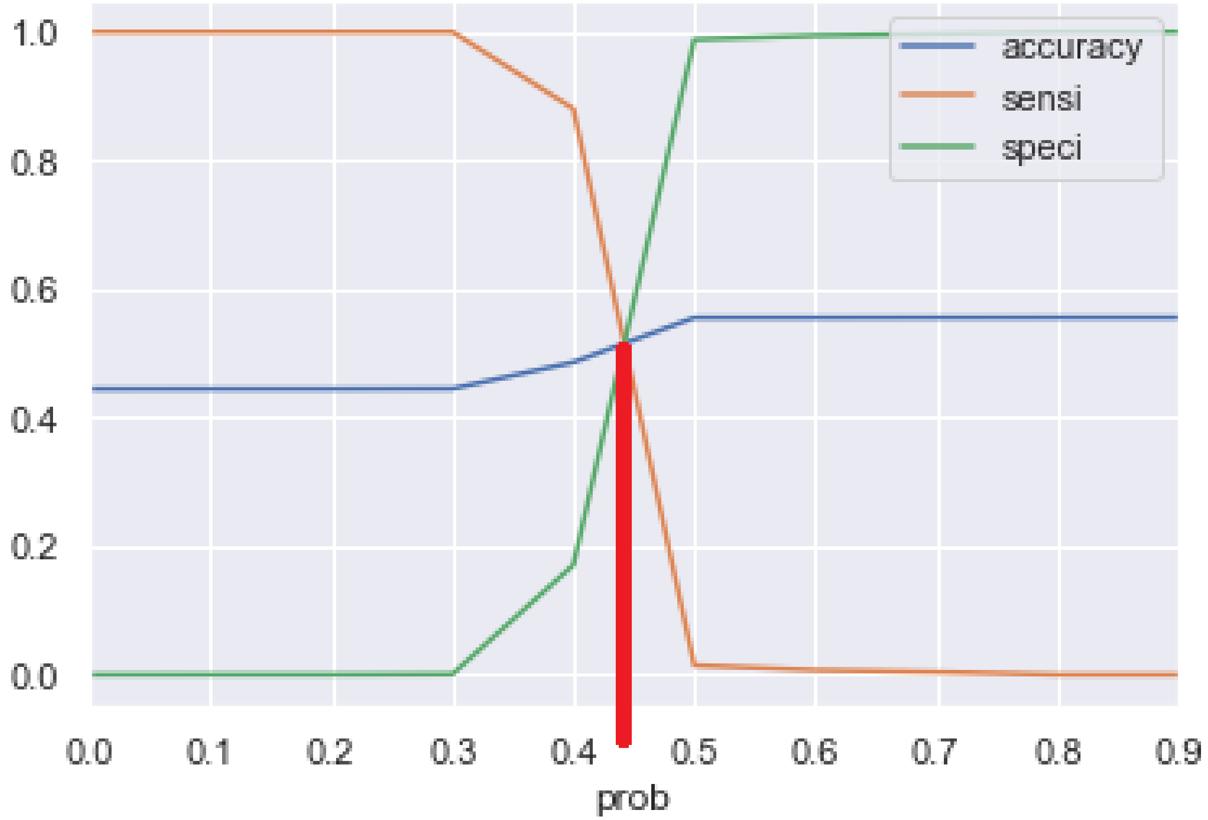
	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.1210	0.008	-14.583	0.000	-0.137	-0.105
Gender	-0.9051	0.378	-2.397	0.017	-1.645	-0.165
Marital_Status	-5.0191	0.748	-6.711	0.000	-6.485	-3.553
No_of_dependents	1.0783	0.135	7.974	0.000	0.813	1.343
Income	0.4842	0.037	13.229	0.000	0.412	0.556
Education	1.0853	0.262	4.142	0.000	0.572	1.599
Type_of_residence	0.6342	0.249	2.543	0.011	0.145	1.123
No_of_months_in_current_company	1.2685	0.071	17.951	0.000	1.130	1.407



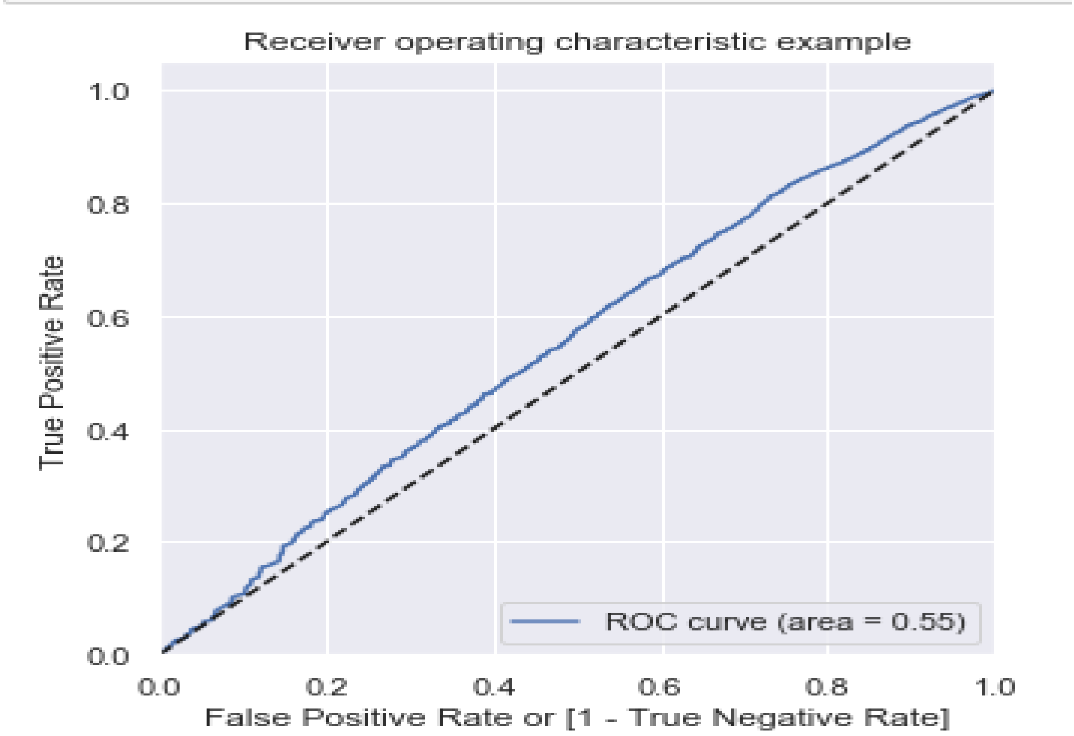
	Features	VIF
0	const	1.44
1	Gender	1.00
2	Marital_Status	1.00
3	No_of_dependents	1.00
4	Income	1.00
5	Education	1.00
6	Type_of_residence	1.00
7	No_of_months_in_current_company	1.00

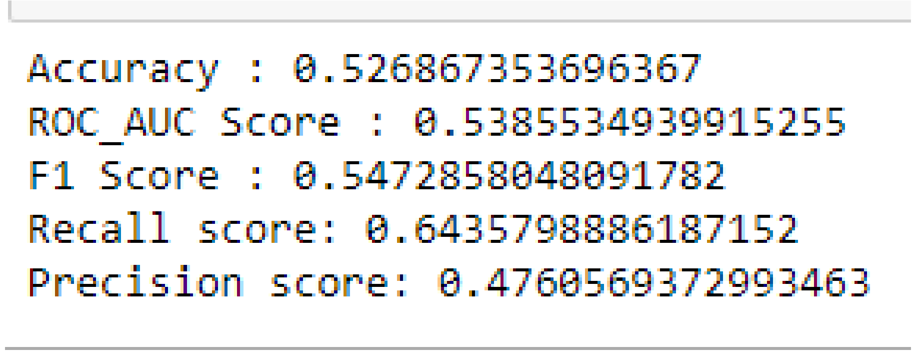
All the above features have less p-values(<0.05) and less VIF(<5 which means less correlation among variables. Hence we can consider above model as final.

Logistic Regression model on demographic dataset imputed with WOE



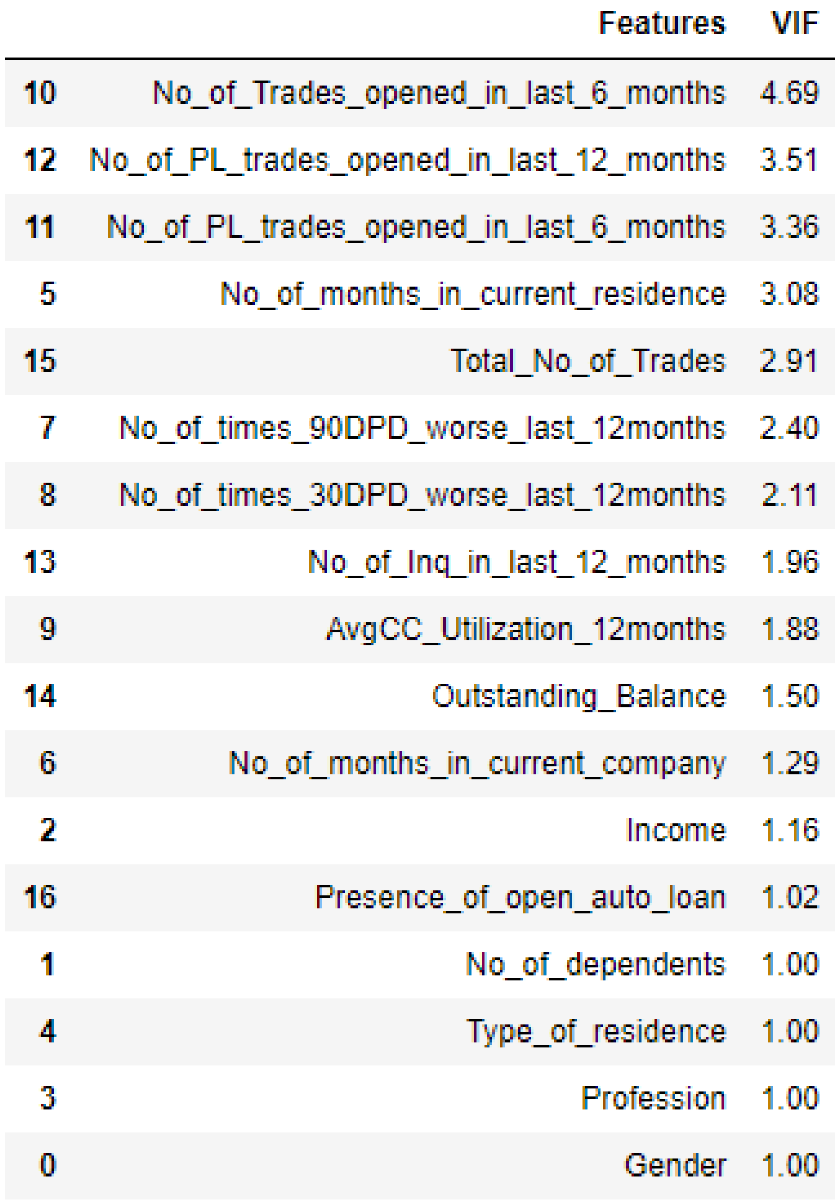
From the curve above, 0.44 is the optimum point to take it as a cutoff probability.



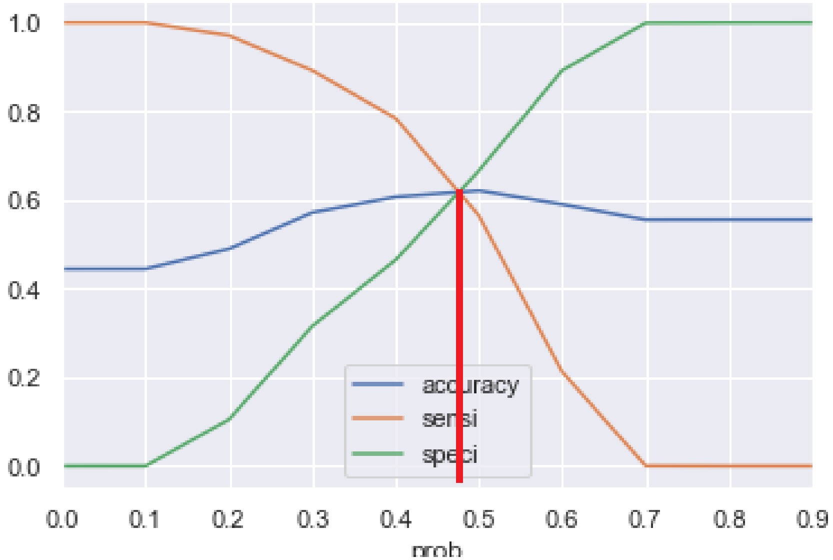


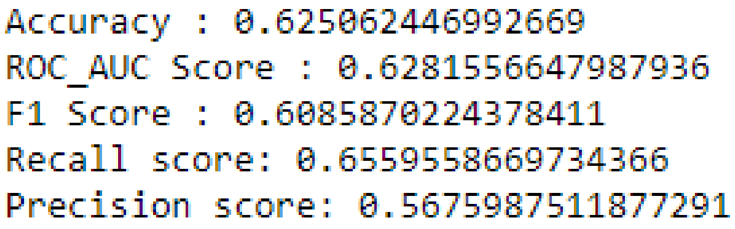
```
Accuracy : 0.526867353696367
ROC_AUC Score : 0.5385534939915255
F1 Score : 0.5472858048091782
Recall score: 0.6435798886187152
Precision score: 0.4760569372993463
```

Logistic Regression model on combined dataset imputed with WOE

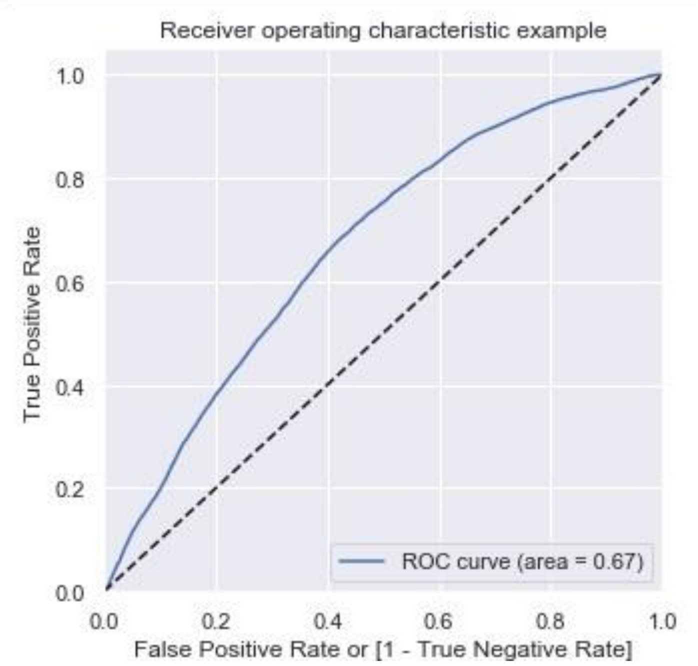


	Features	VIF
10	No_of_Trades_opened_in_last_6_months	4.69
12	No_of_PL_trades_opened_in_last_12_months	3.51
11	No_of_PL_trades_opened_in_last_6_months	3.36
5	No_of_months_in_current_residence	3.08
15	Total_No_of_Trades	2.91
7	No_of_times_90DPD_worse_last_12months	2.40
8	No_of_times_30DPD_worse_last_12months	2.11
13	No_of_Inq_in_last_12_months	1.96
9	AvgCC_Utilization_12months	1.88
14	Outstanding_Balance	1.50
6	No_of_months_in_current_company	1.29
2	Income	1.16
16	Presence_of_open_auto_loan	1.02
1	No_of_dependents	1.00
4	Type_of_residence	1.00
3	Profession	1.00
0	Gender	1.00



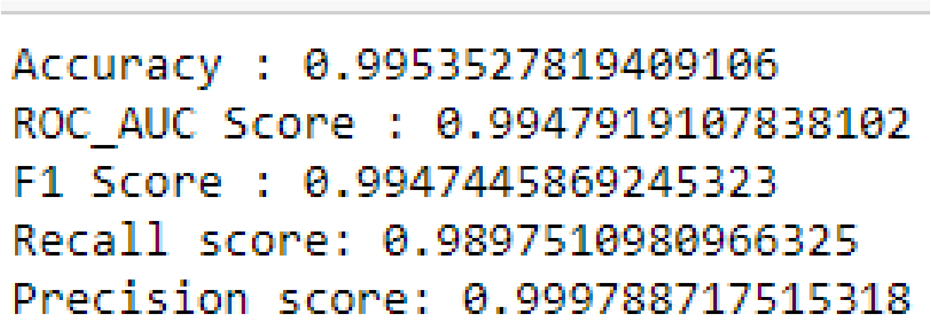


```
Accuracy : 0.625062446992669
ROC_AUC Score : 0.6281556647987936
F1 Score : 0.6085870224378411
Recall score: 0.6559558669734366
Precision score: 0.5675987511877291
```



Random Forest

- Check the report of our default model –



```
Accuracy : 0.9953527819409106
ROC_AUC Score : 0.9947919107838102
F1 Score : 0.9947445869245323
Recall score: 0.9897510980966325
Precision score: 0.999788717515318
```

- Tuning max_depth --

```
We get auc score of 0.9846392126055441 using {'max_depth': 17}
```

- Tuning n_estimators --

```
We get auc score of 0.9182784798437361 using
{'n_estimators': 1300}
```

- Tuning max_features --

```
We get auc score of 0.9174020377589268 using
{'max_features': 20}
```

- Tuning min_samples_leaf --

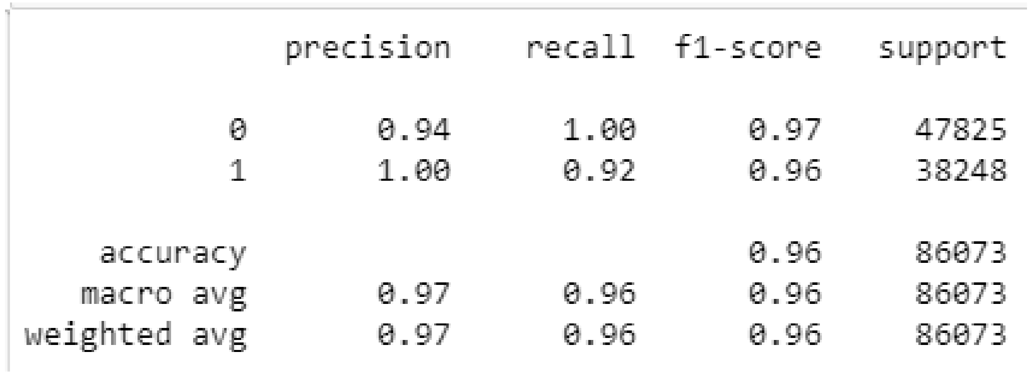
```
We get auc score of 0.9772804696718481 using
 {'min_samples_leaf': 100}
```

- Tuning min_samples_split --

```
We get auc score of 0.9830826117029351
 using {'min_samples_split': 200}
```

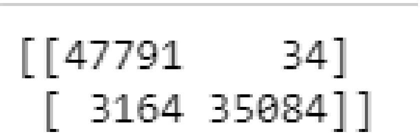
Random forest model with optimal hyper parameters

- Report of our optimal model on training data

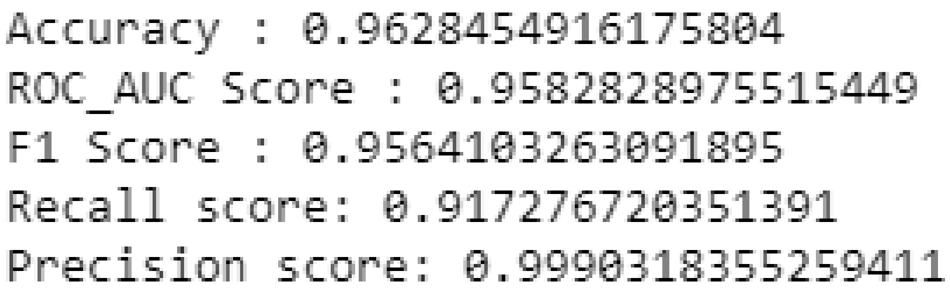


```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     47825
           1       1.00      0.92      0.96     38248

    accuracy                           0.96     86073
   macro avg       0.97      0.96      0.96     86073
weighted avg       0.97      0.96      0.96     86073
```

- confusion matrix –

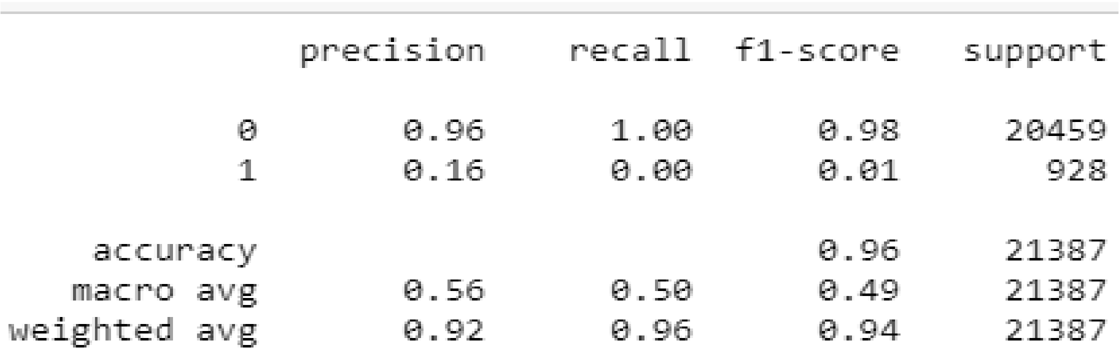


```
[[47791    34]
 [ 3164 35084]]
```



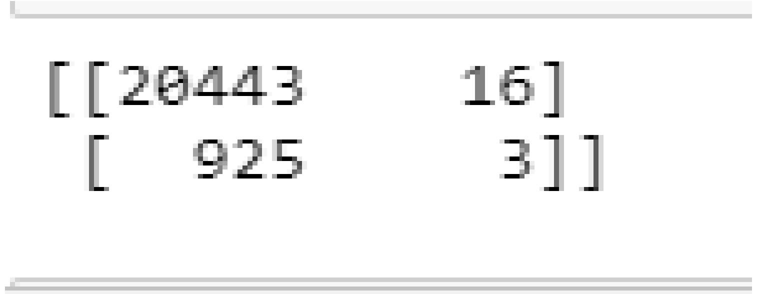
```
Accuracy : 0.9628454916175804
ROC_AUC Score : 0.9582828975515449
F1 Score : 0.9564103263091895
Recall score: 0.9172767203513391
Precision score: 0.9990318355259411
```

Report of optimal model on validation data

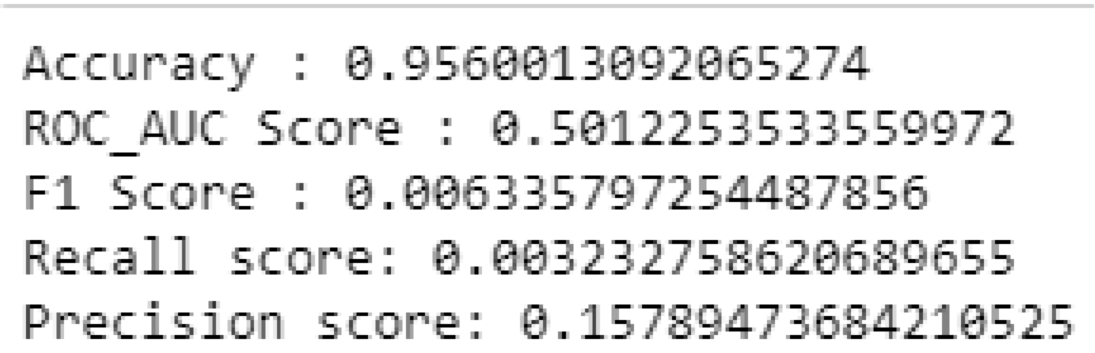


```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     20459
           1       0.16      0.00      0.01       928

    accuracy                           0.96     21387
   macro avg       0.56      0.50      0.49     21387
weighted avg       0.92      0.96      0.94     21387
```

confusion matrix

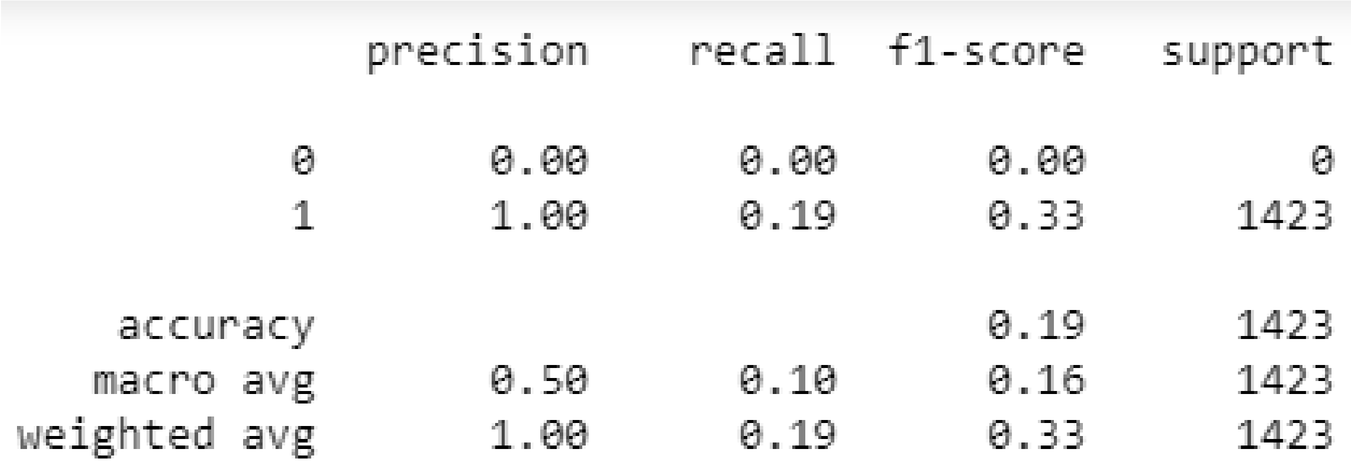


```
[[20443    16]
 [  925     3]]
```



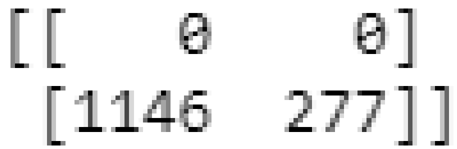
```
Accuracy : 0.9560013092065274
ROC_AUC Score : 0.5012253533559972
F1 Score : 0.006335797254487856
Recall score: 0.003232758620689655
Precision score: 0.15789473684210525
```

Report of optimal model on rejected data

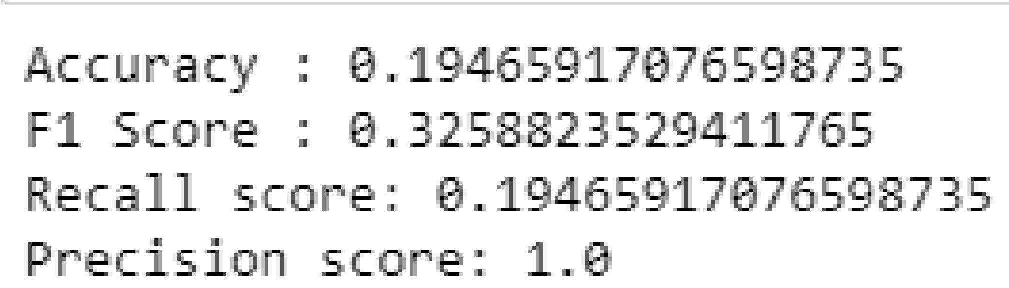


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.19      0.33      1423

    accuracy                           0.19      1423
   macro avg       0.50      0.10      0.16      1423
weighted avg       1.00      0.19      0.33      1423
```

confusion matrix



```
[[   0    0]
 [1146  277]]
```

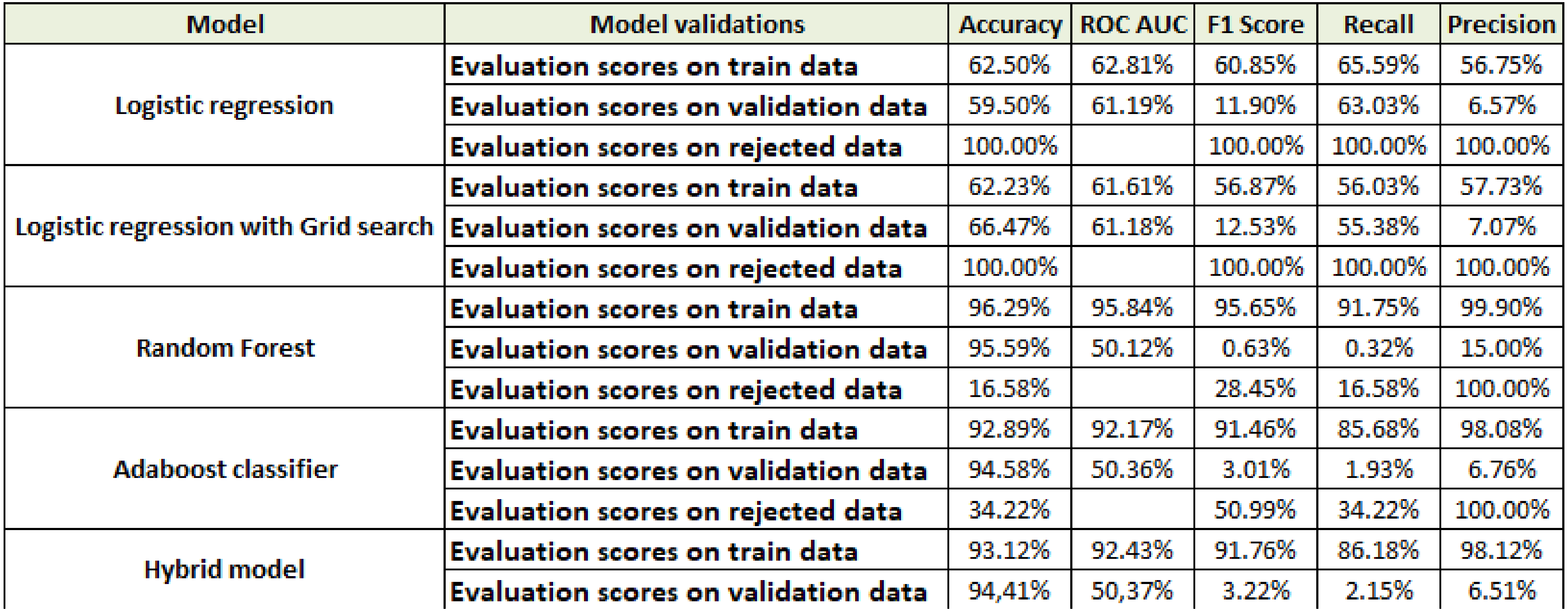


```
Accuracy : 0.19465917076598735
F1 Score : 0.3258823529411765
Recall score: 0.19465917076598735
Precision score: 1.0
```

Modelling Techniques

- The objective of the model is to optimize **Sensitivity / Recall** .
- Confusion matrix is prepared for each and every model.
- Sensitivity, specificity, accuracy curve is seen for Logistic Regression models.
- AUC-ROC curve for the Logistic Regression models using cut-off values is checked for every model.
- Within each model, evaluation using GridSerach (based on recall values) should be done to get models with optimized hyper parameters.
- For evaluation among models, the dataset for rejected applications (with performance tag missing), which were assumed as potentially defaulters should be considered for evaluations.

From the below scores Logistic regression is selected as the best model because recall is not dropped as if it dropped in other models for test data. Also rejected data is predicted with 100% accuracy where it is not in Random forest, Adaboost and hybrid models



Model	Model validations	Accuracy	ROC AUC	F1 Score	Recall	Precision
Logistic regression	**Evaluation scores on train data**	62.50%	62.81%	60.85%	65.59%	56.75%
	Evaluation scores on validation data	59.50%	61.19%	11.90%	63.03%	6.57%
	Evaluation scores on rejected data	100.00%		100.00%	100.00%	100.00%
Logistic regression with Grid search	**Evaluation scores on train data**	62.23%	61.61%	56.87%	56.03%	57.73%
	Evaluation scores on validation data	66.47%	61.18%	12.53%	55.38%	7.07%
	Evaluation scores on rejected data	100.00%		100.00%	100.00%	100.00%
Random Forest	**Evaluation scores on train data**	96.29%	95.84%	95.65%	91.75%	99.90%
	Evaluation scores on validation data	95.59%	50.12%	0.63%	0.32%	15.00%
	Evaluation scores on rejected data	16.58%		28.45%	16.58%	100.00%
Adaboost classifier	**Evaluation scores on train data**	92.89%	92.17%	91.46%	85.68%	98.08%
	Evaluation scores on validation data	94.58%	50.36%	3.01%	1.93%	6.76%
	Evaluation scores on rejected data	34.22%		50.99%	34.22%	100.00%
Hybrid model	**Evaluation scores on train data**	93.12%	92.43%	91.76%	86.18%	98.12%
	Evaluation scores on validation data	94,41%	50,37%	3.22%	2.15%	6.51%

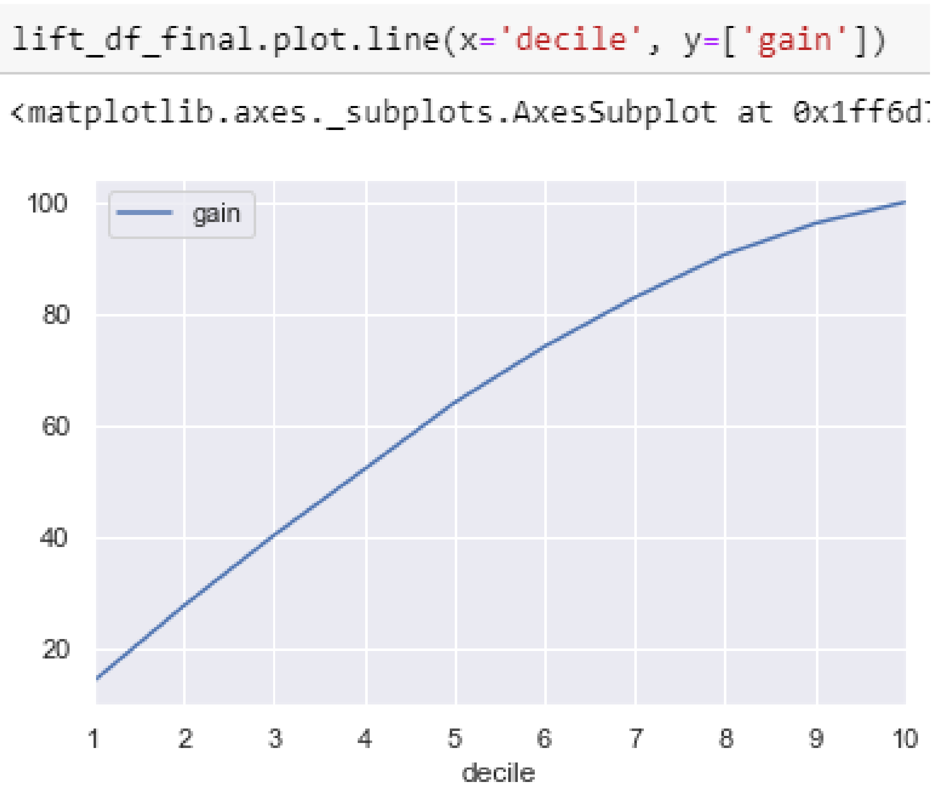
Reasons for Model Selection

- **Final Model chosen :** Logistic Regression

  - The model gave best possible recall values on Test data
  - Prediction on rejected applications is with 100% accuracy.
  - It is a model which adds almost all available information across the data.
  - The model is very stable.
  - The model is expected to have comparatively long life over others.
  - It requires no outlier treatment.
  - The model is expected **not** to over fit on any data.

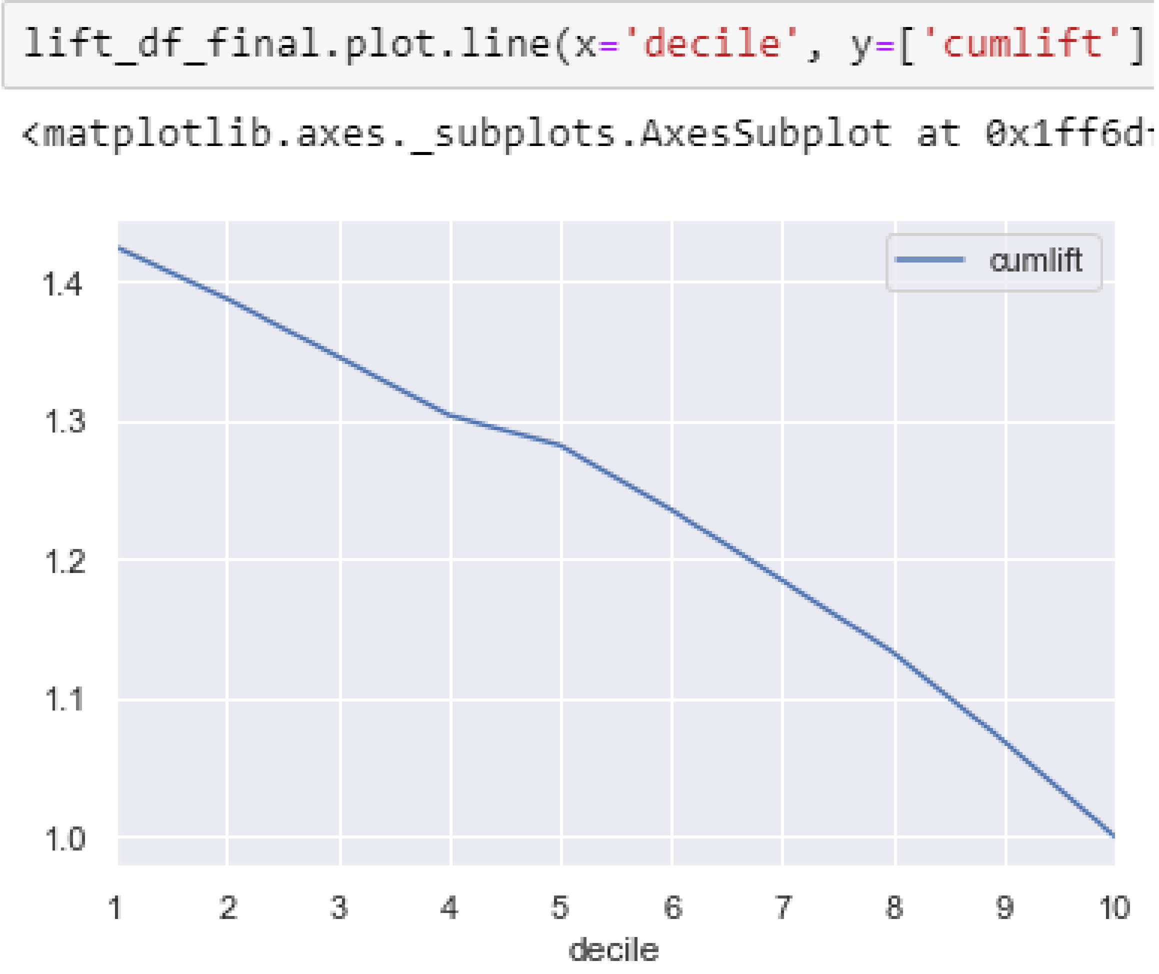
Lift & gain chart

- **We took threshhold value as 0.47 obtained from logistic regression**



```
lift_df_final.plot.line(x='decile', y=['gain'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ff6d
```



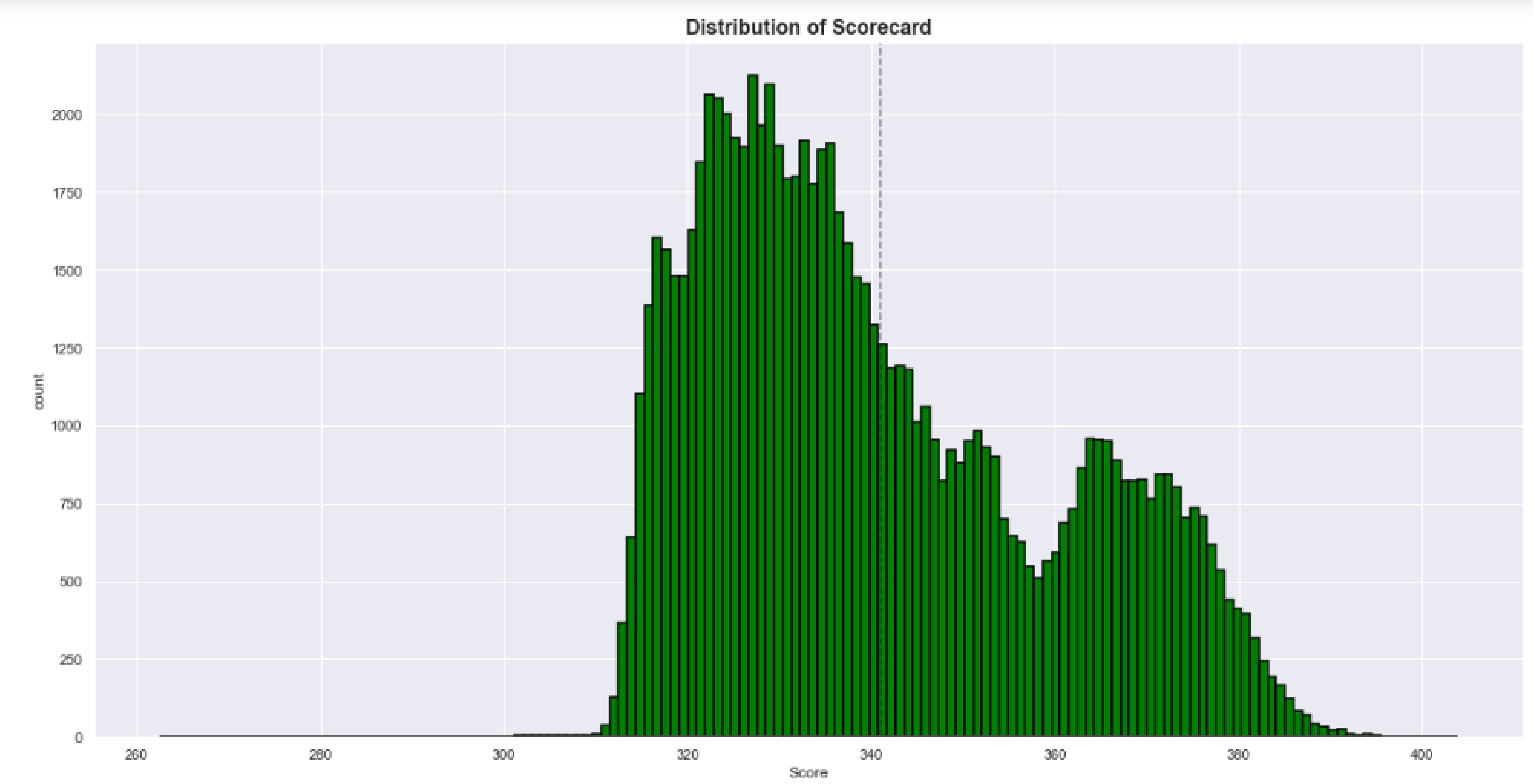
```
lift_df_final.plot.line(x='decile', y=['cumlift']
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ff6d
```

83%(approx) gain is provided at 7th decile.

Original Defaulters rate calculation – 4.22

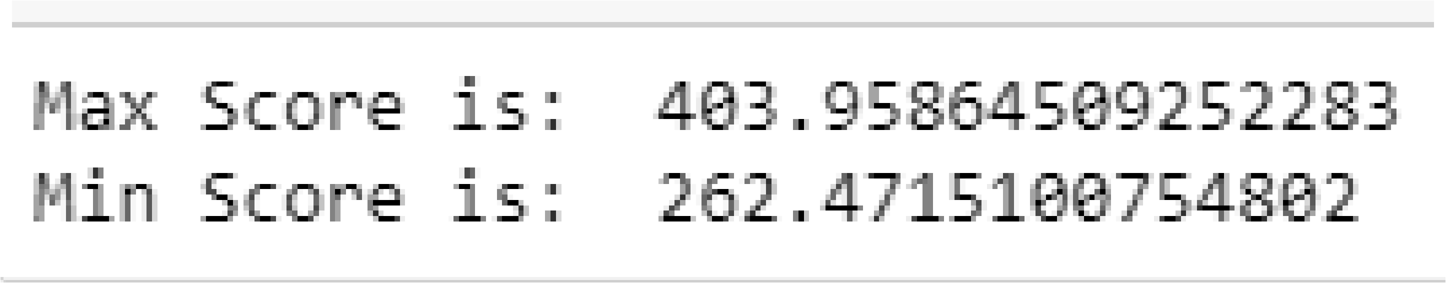
Scorecard calculation



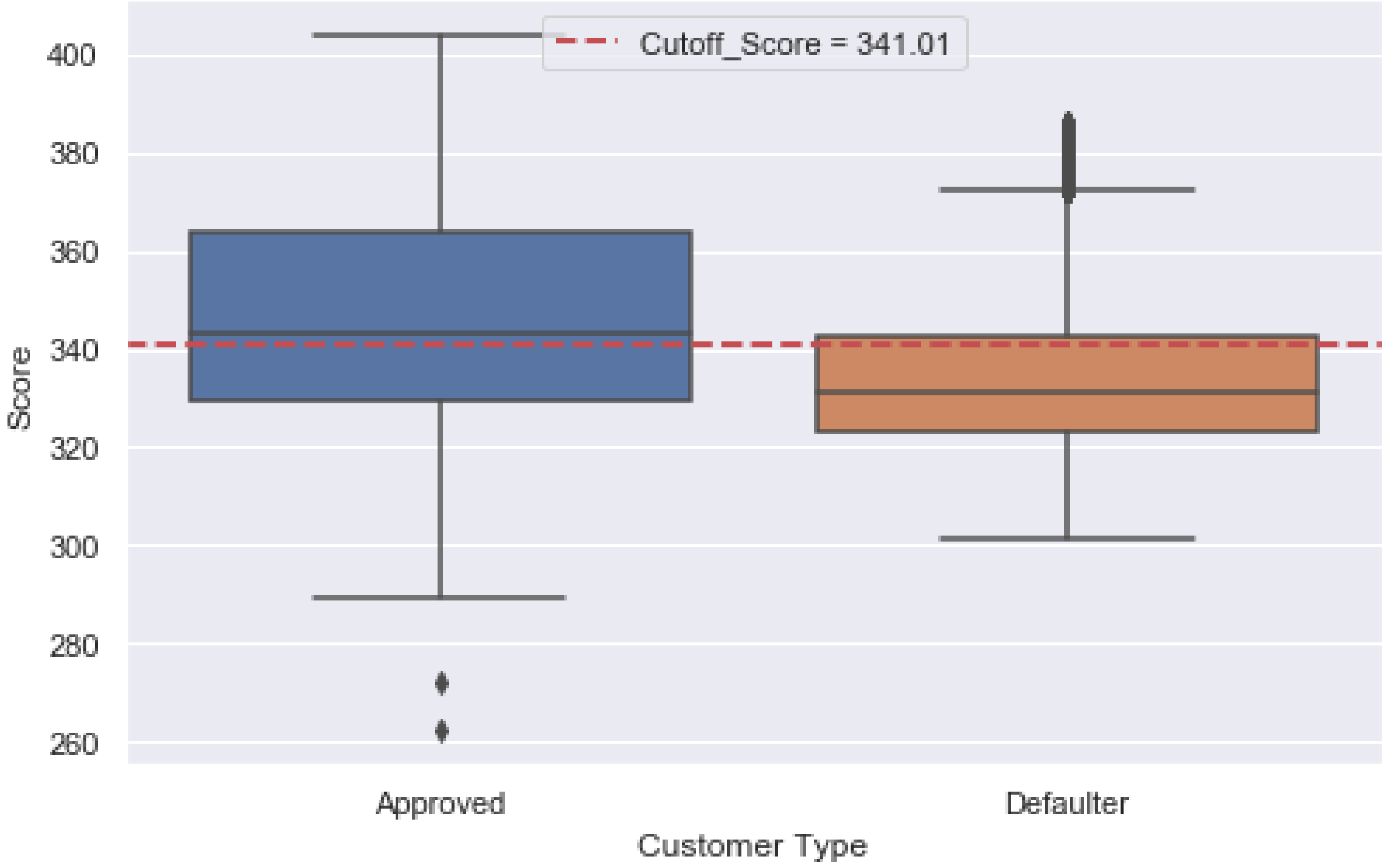
Application Scorecard

- Final application scorecard was made using the Logistic regression model on the entire dataset which also contained predictions for missing values in “Performance Tag”. 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))**



```
Max Score is:  403.95864509252283
Min Score is:  262.4715100754802
```



Mean and median score of the approved customers is higher than rejected customers. Cutoff score is indicated by red dotted line and it is 341.01

Mean and Median score of the Approved applicants are **346.01 and 342.88** respectively. Mean and Median score of the rejected applicants are **334.75 and 331.14** respectively.

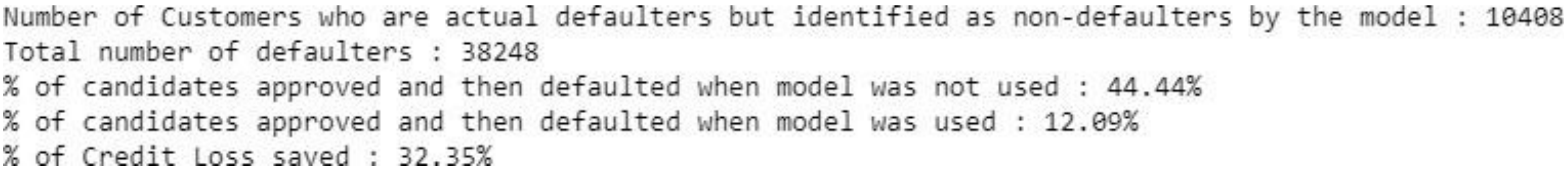
Benefits

- Our objective is to minimize “Net Credit Loss’’ from Profit & Loss perspective.
- With our model we achieved fantastic discriminatory power over pre-identifying risky costumers.
- 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.
- It reduced the cost spent on Underwriters who rejects the applications by viewing and identifying it manually.
- Profit using model will be total profit due to each true positive and each true negative minus loss from each false positive and each false negative prediction
- It Reduced time for processing of application requests
- Procedure is automated
- Prevents manual errors made by employees, hence preventing business loss.

Bias can be easily removed which comes in due to gender, race or religion.

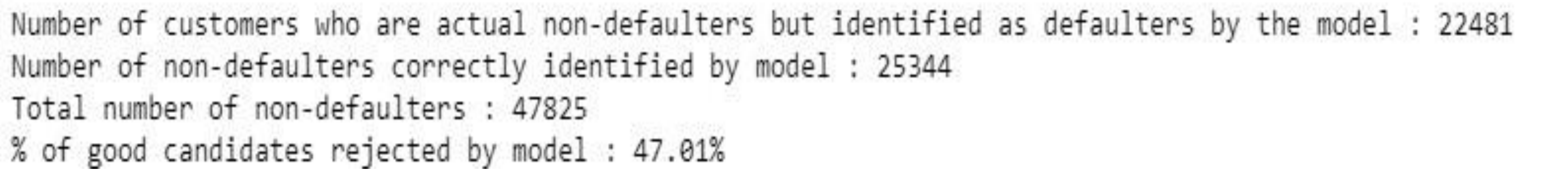
- Scorecard and cut-off provides clear instructions as how to go with application.
- Decision making has become fast.

Calculate credit loss saved when model is used

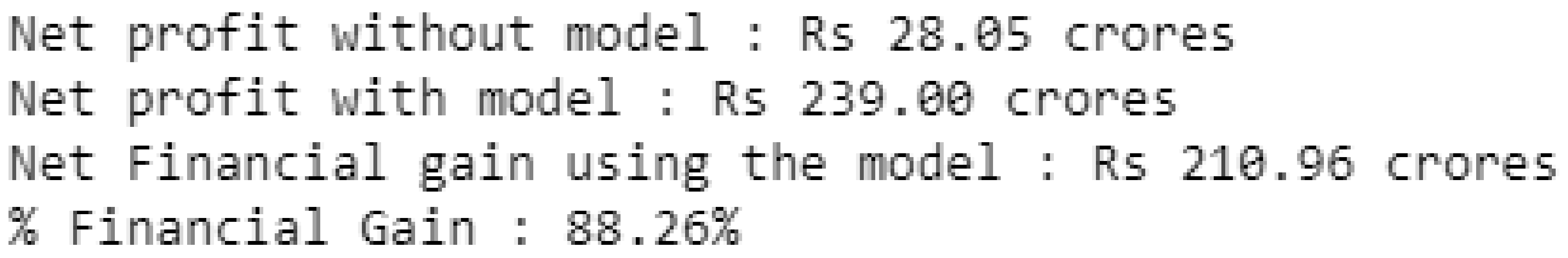


```
Number of Customers who are actual defaulters but identified as non-defaulters by the model : 10408
Total number of defaulters : 38248
% of candidates approved and then defaulted when model was not used : 44.44%
% of candidates approved and then defaulted when model was used : 12.09%
% of Credit Loss saved : 32.35%
```

Assume that average credit loss for defaulted customer is Rs 100000/- and profit for each non-defaulters be Rs 30000/-



```
Number of customers who are actual non-defaulters but identified as defaulters by the model : 22481
Number of non-defaulters correctly identified by model : 25344
Total number of non-defaulters : 47825
% of good candidates rejected by model : 47.01%
```



```
Net profit without model : Rs 28.05 crores
Net profit with model : Rs 239.00 crores
Net Financial gain using the model : Rs 210.96 crores
% Financial Gain : 88.26%
```

Conclusion

We chose Logistic Regression (LR) as against Random forest (RF) as Sensitivity Score was too low for Random Forest (0.44%) as against Logistic Regression (63%).

•Though the accuracy was High for RF, our key Focus here is to have a model with Good True Positive Rate to catch hold of Defaults as much as possible trading off with Non Defaulted Applicant also being marked as Default.

- **Logistic Regression gave us a Sensitivity of 72.8%**
- **Optimal Application Score cut-off identified was 341**
- **Credit Loss saved is 32.35% with New predictive Model**
- **Net Financial Gain of 88.26% with New predictive Model**