Vehicle Loan Default Predictions

Abhilash SR
Krishnamurthy S
Krishnaraj Palanychamy
Prabhakaran S
Pravin Kumar S
Vishwanath Kannan

GUIDED BY
Animesh Tiwari

GREAT LAKES
INSTITUTE OF MANAGEMENT, CHENNAI

greatlearning

Contents

1. Introduction

People avail vehicle loan from banks to buy their dream cars. Car loans have taken off in India witnessing an increase in growth of 18-20% which is a huge increase in 2019.Bank and vehicle finance companies are making this dream come true by providing the vehicle loan facility.Financing a vehicle involves a lot of technicalities like the kind of vehicle to be financed, the route on which the vehicle will be plying, the operating expenses of the customer, etc. Increase in demand for the loans has also resulted in increased chances of loss to banks. Indian Banks have lost 200 Crore Rupees each year due to defaulters.

1.2. Problem Statement

The objective of the capstone project is to predict whether the customers will be Payment default in the first EMI on Vehicle Loan on due date with respect to mainly the Disbursed amount, Loan to Value percentage of the asset. The dataset is taken from Loan Default Prediction Dataset from Kaggle.

1.3. Data Sets

The dataset provided by Kaggle is under file 'train.csv'. The dataset comprises of 233,146 rows and 41 columns.

1.4. Data Dictionary

UniqueID	Identifier for customers
loan_default	Payment default in the first EMI on due date
disbursed_amount	Amount of Loan disbursed
asset_cost	Cost of the Asset
ltv	Loan to Value of the asset
branch_id	Branch where the loan was disbursed
Supplier_id	Vehicle Dealer where the loan was disbursed
manufacturer_id	Vehicle manufacturer (Hero, Honda, TVS etc.)
Current_pincode	Current pincode of the customer
Date.of.Birth	Date of birth of the customer
Employment.Type	Employment Type of the customer (Salaried/Self Employed)
DisbursalDate	Date of disbursement
State_ID	State of disbursement
Employee_code_ID	Employee of the organization who logged the disbursement
MobileNo_Avl_Flag	if Mobile no. was shared by the customer then flagged as 1
Aadhar_flag	if aadhar was shared by the customer then flagged as 1
PAN_flag	if pan was shared by the customer then flagged as 1
VoterID_flag	if voter was shared by the customer then flagged as 1
Driving_flag	if DL was shared by the customer then flagged as 1
Passport_flag	if passport was shared by the customer then flagged as 1
PERFORM_CNS.SCORE	Bureau Score
PERFORM_CNS.SCORE.DESCRIPTION	Bureau score description
PRI.NO.OF.ACCTS	count of total loans taken by the customer at the time of disbursement
PRI.ACTIVE.ACCTS	count of active loans taken by the customer at the time of disbursement
PRI.OVERDUE.ACCTS	count of default accounts at the time of disbursement

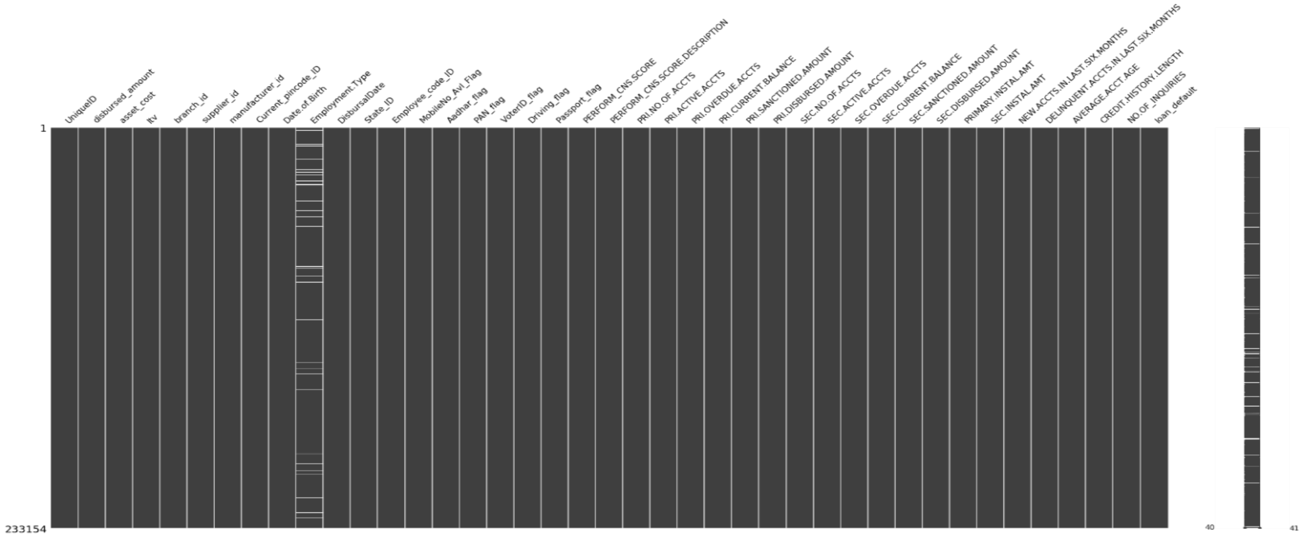
PRI.CURRENT.BALANCE	Principal outstanding amount of the active loans at the time of disbursement
PRI.SANCTIONED.AMOUNT	Amount that was sanctioned for all the loans at the time of disbursement
PRI.DISBURSED.AMOUNT	Amount that was disbursed for all the loans at the time of disbursement
SEC.NO.OF.ACCTS	count of total loans taken by the customer at the time of disbursement
SEC.ACTIVE.ACCTS	count of active loans taken by the customer at the time of disbursement
SEC.OVERDUE.ACCTS	count of default accounts at the time of disbursement
SEC.CURRENT.BALANCE	total Principal outstanding amount of the active loans at the time of disbursement
SEC.SANCTIONED.AMOUNT	total amount that was sanctioned for all the loans at the time of disbursement
SEC.DISBURSED.AMOUNT	total amount that was disbursed for all the loans at the time of disbursement
PRIMARY.INSTAL.AMT	EMI Amount of the primary loan
SEC.INSTAL.AMT	EMI Amount of the secondary loan
NEW.ACCTS.IN.LAST.SIX.MONTHS	New loans taken by the customer in last 6 months before the disbursment
DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS	Loans defaulted in the last 6 months
AVERAGE.ACCT.AGE	Average loan tenure
CREDIT.HISTORY.LENGTH	Time since first loan
NO.OF_INQUIRIES	Enquries done by the customer for loans
loan_default	Defaulters to be predicted.

2. Data Preprocessing

14 Numerical columns,2 Date type columns and 25 Categorical columns

UniqueID	int64
disbursed_amount	int64
asset_cost	int64
ltv	float64
branch_id	int64
supplier_id	int64
manufacturer_id	int64
Current_pincode_ID	int64
Date.of.Birth	object
Employment.Type	object
DisbursalDate	object
State_ID	int64
Employee_code_ID	int64
MobileNo_Avl_Flag	int64
Aadhar_flag	int64
PAN_flag	int64
VoterID_flag	int64
Driving_flag	int64
Passport_flag	int64
PERFORM_CNS.SCORE	int64
PERFORM_CNS.SCORE.DESCRIPTION	object
PRI.NO.OF.ACCTS	int64
PRI.ACTIVE.ACCTS	int64
PRI.OVERDUE.ACCTS	int64
PRI.CURRENT.BALANCE	int64
PRI.SANCTIONED.AMOUNT	int64
PRI.DISBURSED.AMOUNT	int64
SEC.NO.OF.ACCTS	int64
SEC.ACTIVE.ACCTS	int64
SEC.OVERDUE.ACCTS	int64
SEC.CURRENT.BALANCE	int64
SEC.SANCTIONED.AMOUNT	int64
SEC.DISBURSED.AMOUNT	int64
PRIMARY.INSTAL.AMT	int64
SEC.INSTAL.AMT	int64
NEW.ACCTS.IN.LAST.SIX.MONTHS	int64
DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS	int64
AVERAGE.ACCT.AGE	object
CREDIT.HISTORY.LENGTH	object
NO.OF_INQUIRIES	int64
loan_default	int64

2.1 Missing Values Imputation



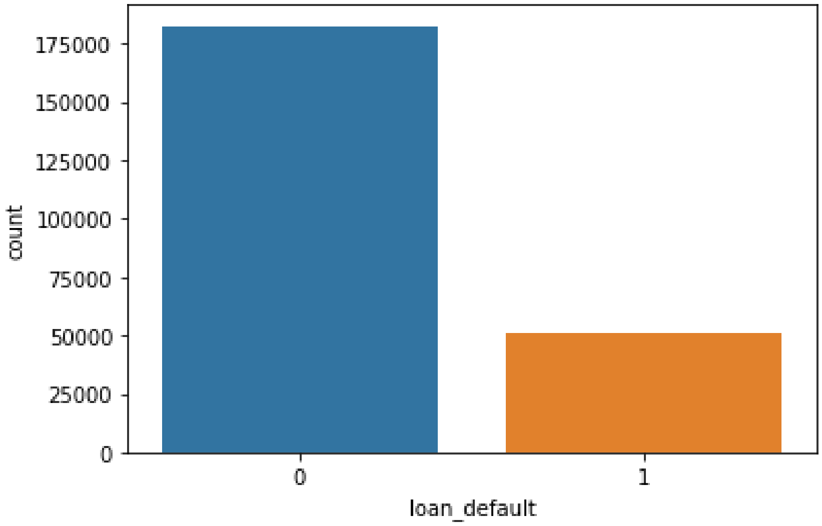
From the Missingno matrix, The Employment Type feature is having 3.29% missing values from the dataset.

2.2 Features

2.2.1 Loan Default

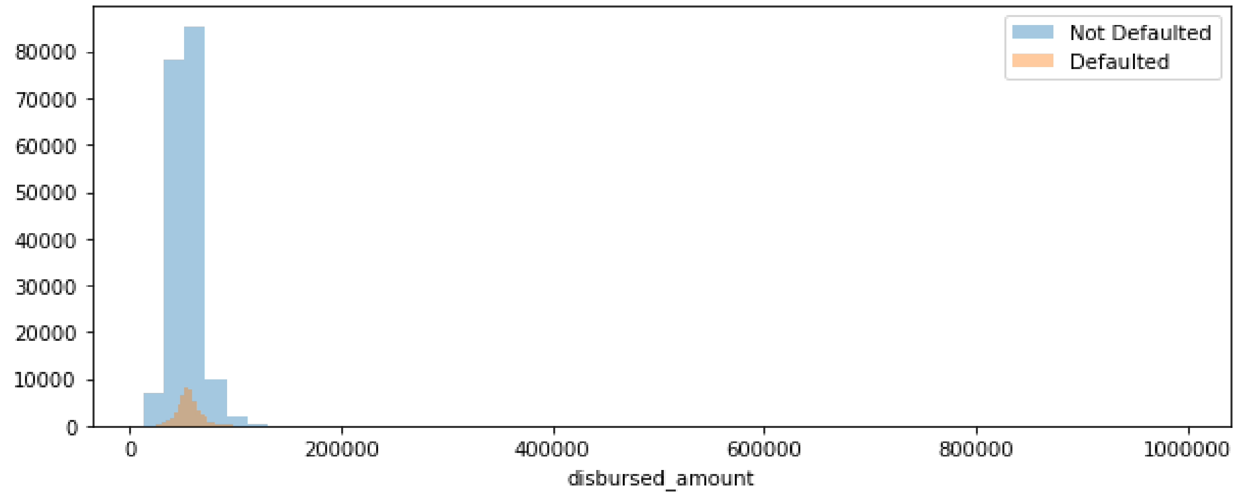
Loan Default column is to predict whether the customer has defaulted during the first EMI on Vehicle Loan on due date. 78.2 of the customers are not-defaulted and 21.7% of defalted.

```
0    182543
1     50611
Name: loan_default, dtype: int64
```



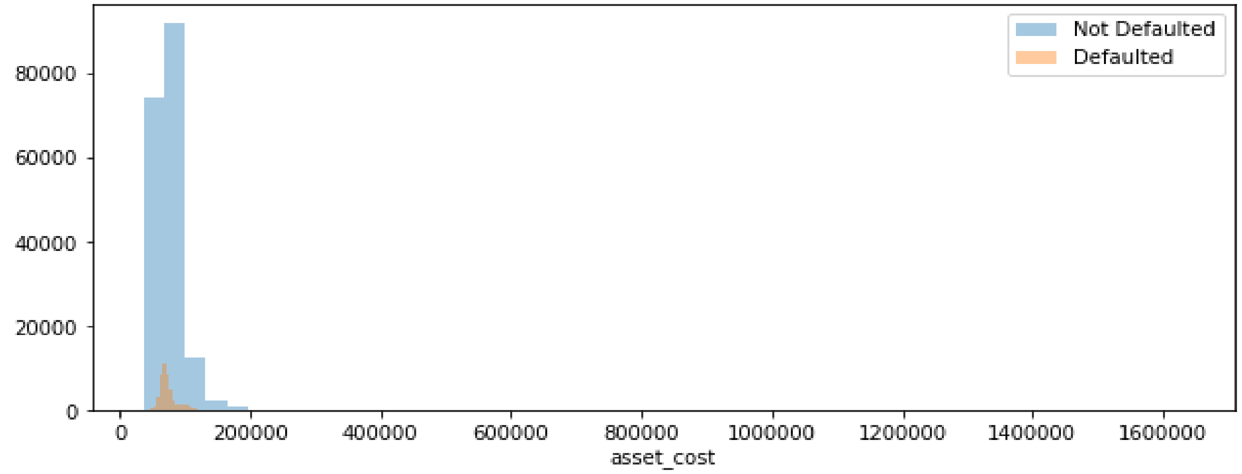
2.2.2 Disbursed Amount

It is the amount of Loan disbursed to the customer. It is a continuous numerical column. Customers getting loan amounts below 200,000 the highest and low greater than 500,000.



2.2.3 Asset Cost

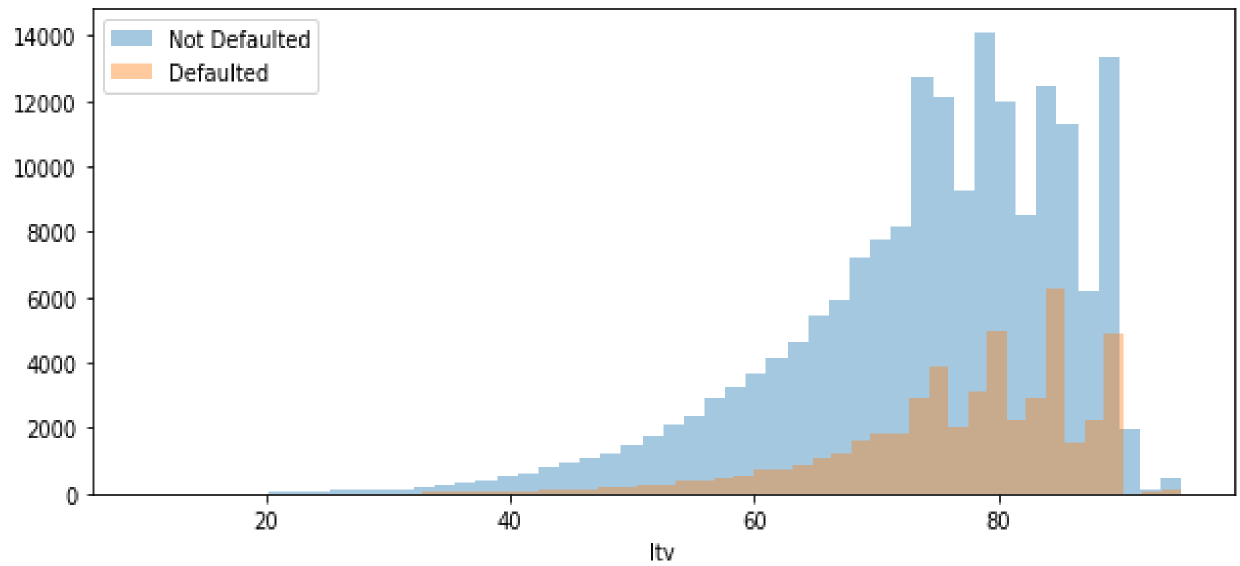
Cost of the vehicle. It is a numerical continuous column to be featured.



2.2.4 Loan to Value

Loan to Value of the asset/vehicle.

$$ltv = \frac{Loan\ Value}{Asset\ Cost}$$

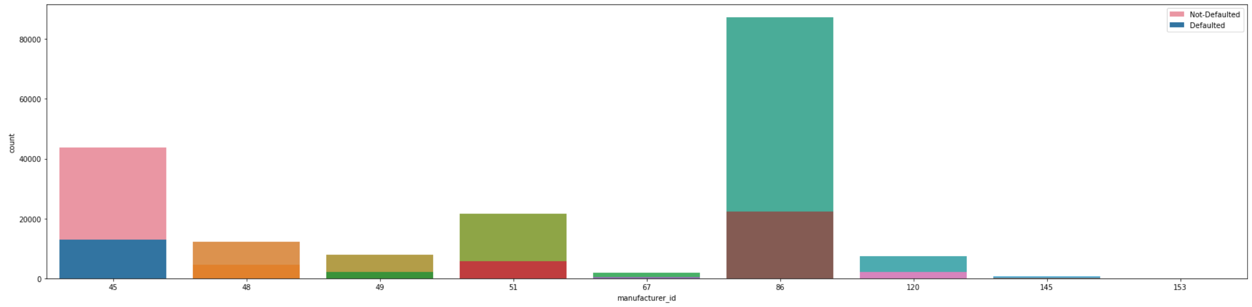


2.2.5 Supplier ID

Vehicle Dealer where the loan was disbursed. There are 2953 Suppliers in the current report.

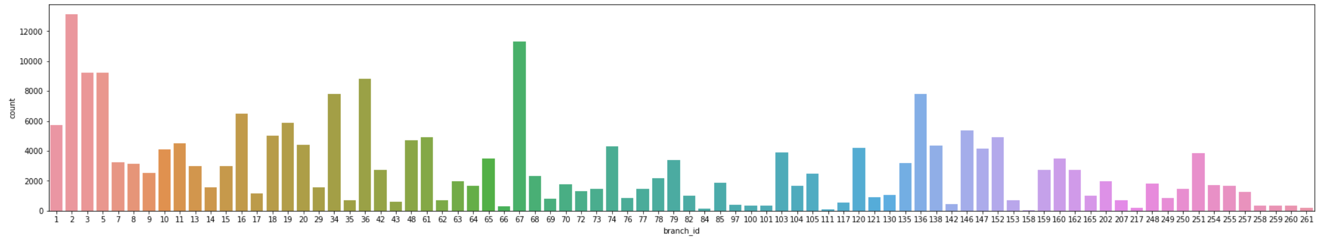
2.2.6 Manufacture ID

Vehicle Manufacturer i-e. TVS, Honda, Hero etc. There are 11 brands given in the label encoded form



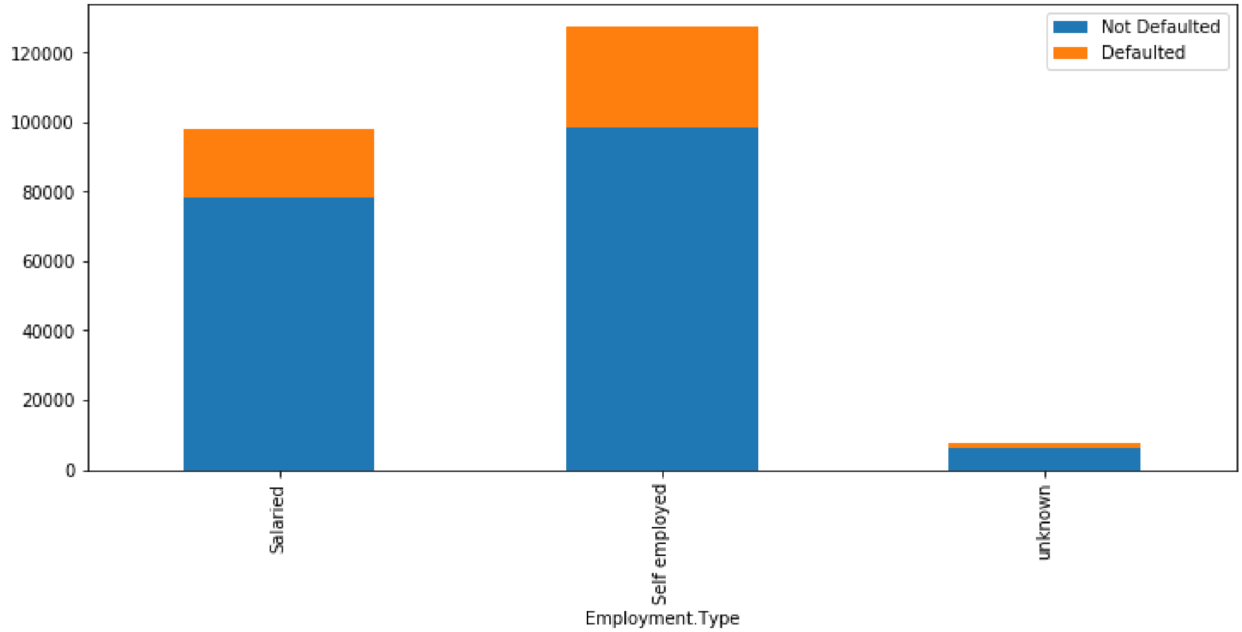
2.2.7 Branch ID

Branch where the loan was disbursed. 82 branches are given in the current report.



2.2.8 Employment Type

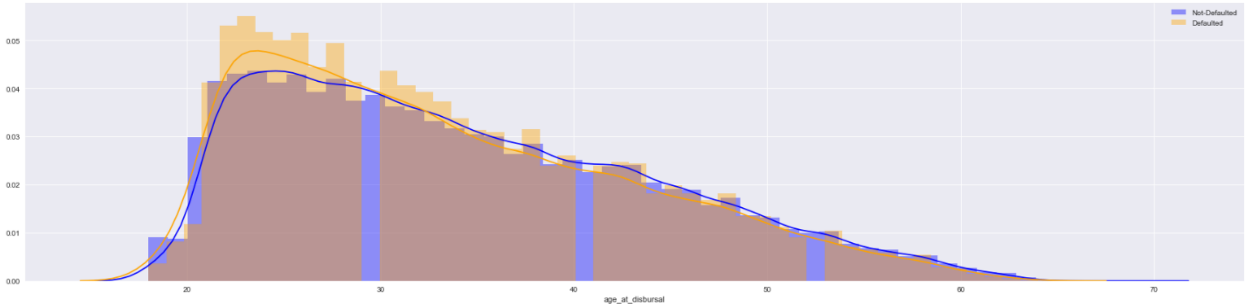
Employment Type of the customer (Salaried/Self Employed) Since There are null values in the Employment Type column. Decided to replace NaN values to ‘unknown’.



2.2.10 Age of Disbursal

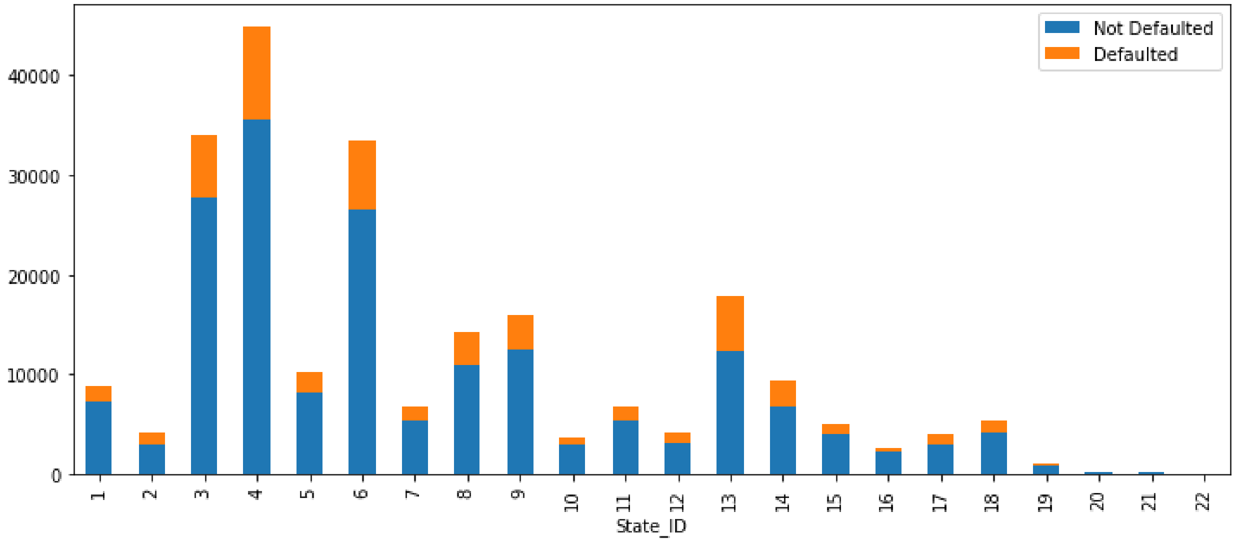
Implemented to combine Date of Birth and Disbursed Date Column to get the age of the customer when he has taken the loan.

```
def age(dur):
    yr = int(dur.split('-')[2])
    if yr >=0 and yr<=19:
        return yr+2000
    else:
         return yr+1900
df['Date.of.Birth'] = df['Date.of.Birth'].apply(age)
df['DisbursalDate'] = df['DisbursalDate'].apply(age)


df['age_at_disbursal']=df['DisbursalDate']-df['Date.of.Birth']
```



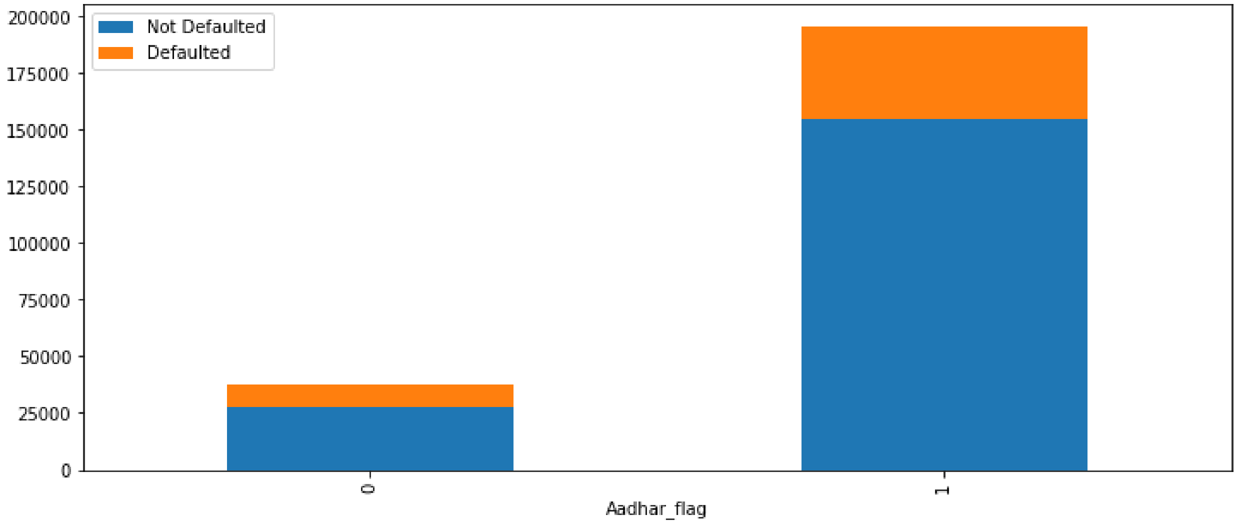
2.2.11 State ID

State at which the loan had been distributed.

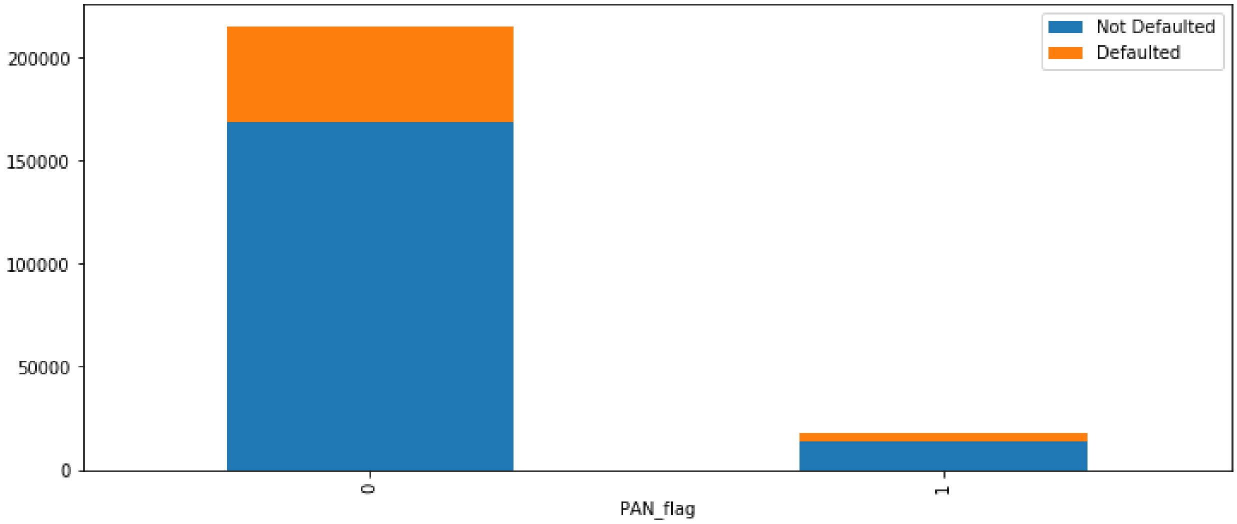


2.2.12 Aadhar Flag

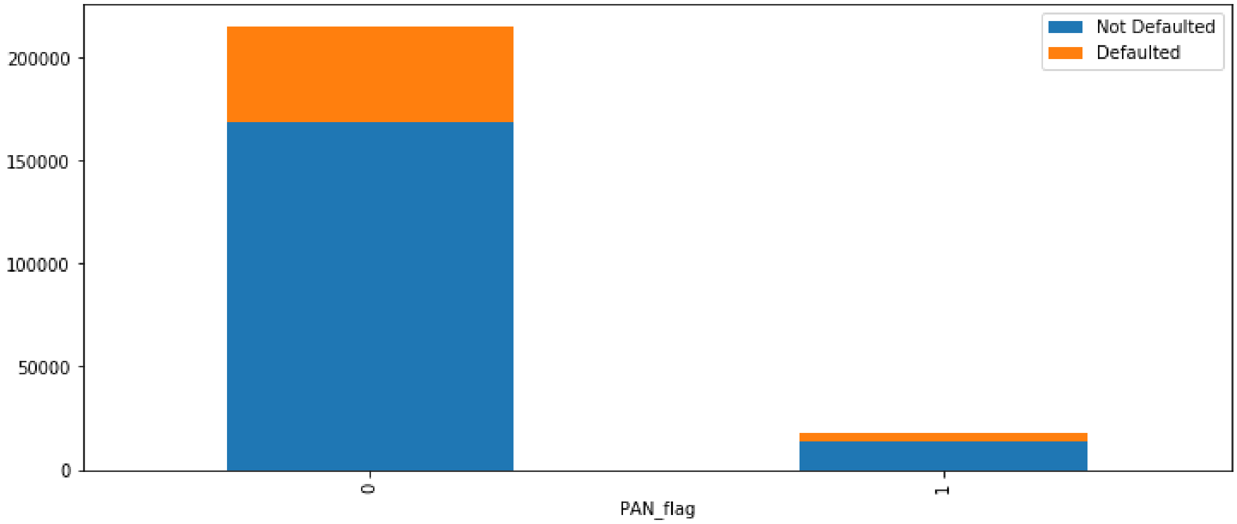
People who have been flagged for the aadar



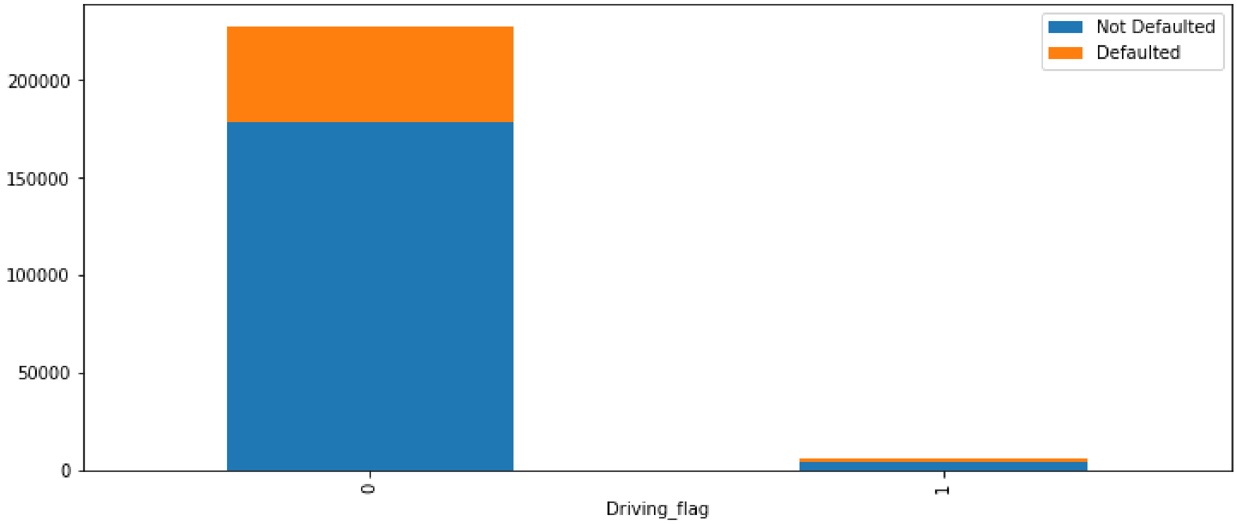
2.2.13 PAN Flag



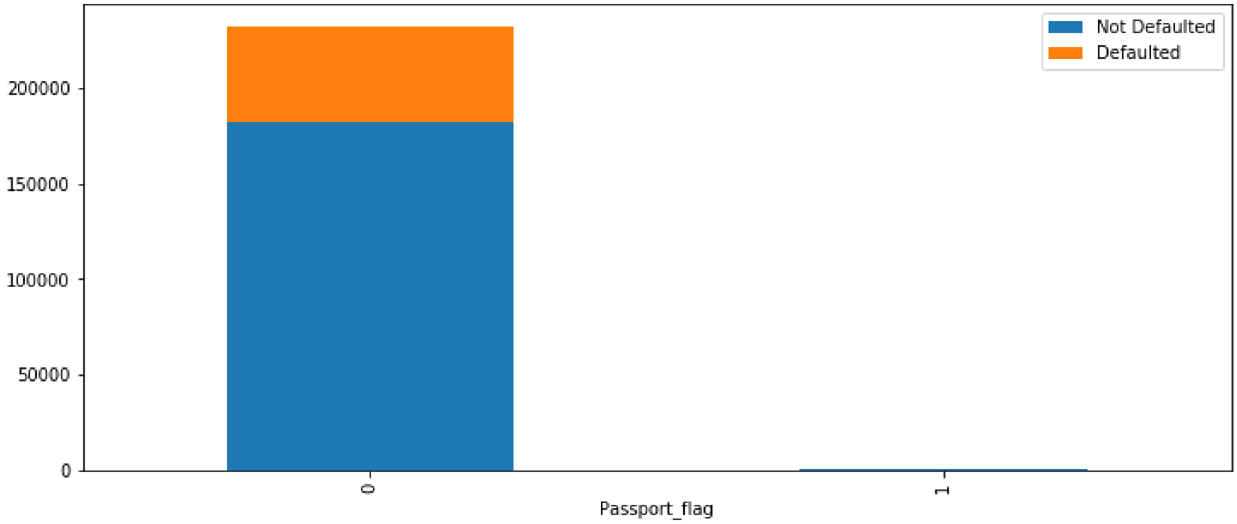
2.2.14 Voter ID Flag



2.2.15 Driving License Flag

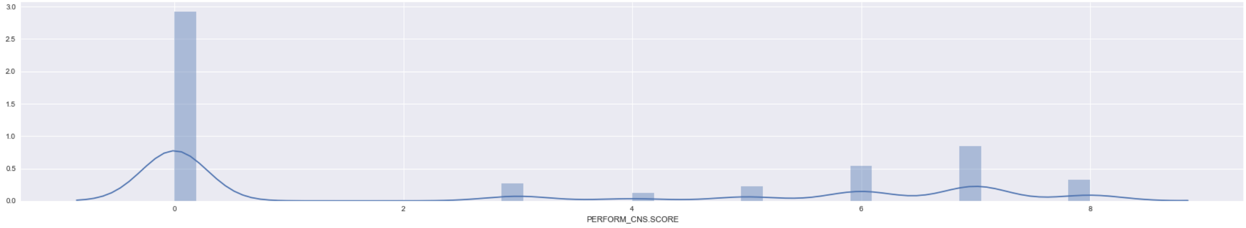


2.2.16 Passport Flag

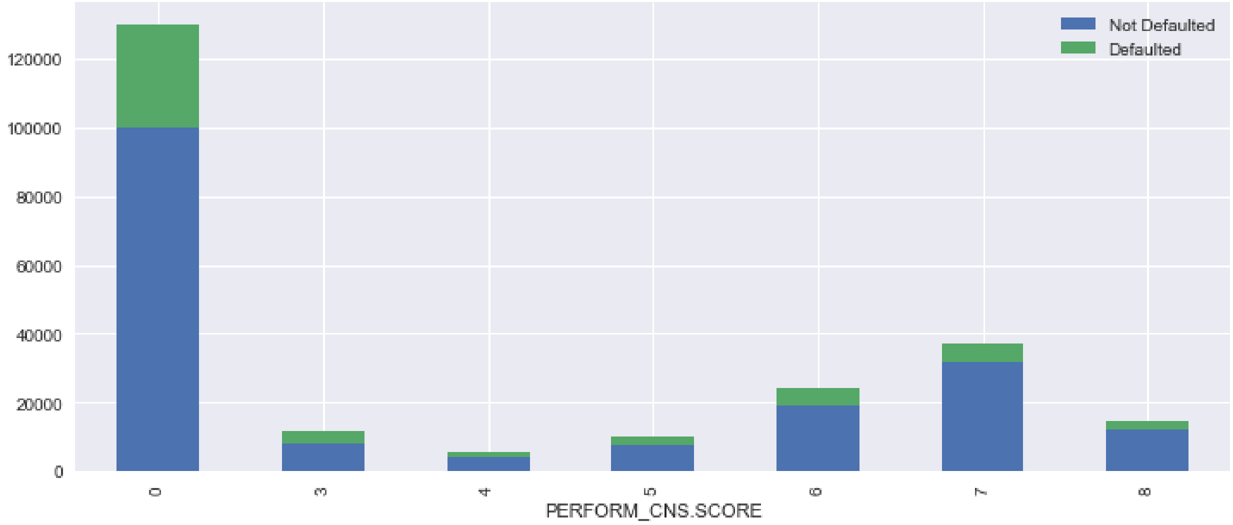


2.2.17 CNS Score

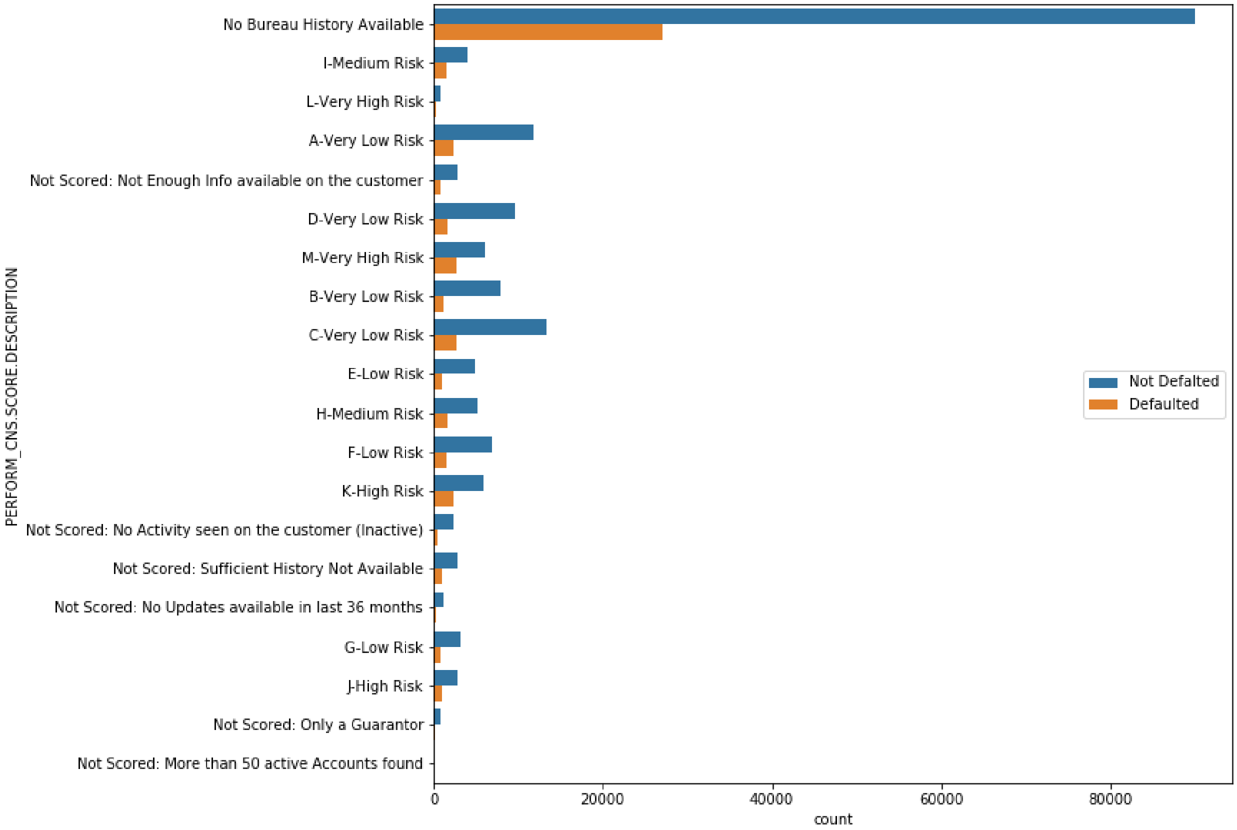
2.2.17.1 PERFORM CNS Score



Converted the CNS Scores in steps of 100 since the score defines his rating



2.2.17.2 PERFORM CNS Score Description



2.2.18 Primary Attributes

2.2.18.1 Primary Number of Accounts:

Count of total loans taken by the customer at the time of disbursement.

2.2.18.2 Primary Active Accounts

Count of active loans taken by the customer at the time of disbursement.

2.2.18.3 Primary Overdue Accounts

Count of default accounts at the time of disbursement

2.2.18.4 Primary Current Balance

total Principal outstanding amount of the active loans at the time of disbursement

2.2.18.5 Primary Sanctioned Amount

total amount that was sanctioned for all the loans at the time of disbursement

2.2.18.6 Primary Disbursed Amount

total amount that was disbursed for all the loans at the time of disbursement

2.2.18.7 Primary Installment Amount

EMI Amount of the primary loan

2.2.19 Secondary Attributes

2.2.19.1 Secondary Number of Accounts:

Count of total loans taken by the customer at the time of disbursement.

2.2.19.2 Secondary Active Accounts

Count of active loans taken by the customer at the time of disbursement.

2.2.19.3 Secondary Overdue Accounts

Count of default accounts at the time of disbursement

2.2.19.4 Secondary Current Balance

total Principal outstanding amount of the active loans at the time of disbursement

2.2.19.5 Secondary Sanctioned Amount

total amount that was sanctioned for all the loans at the time of disbursement

2.2.19.6 Secondary Disbursed Amount

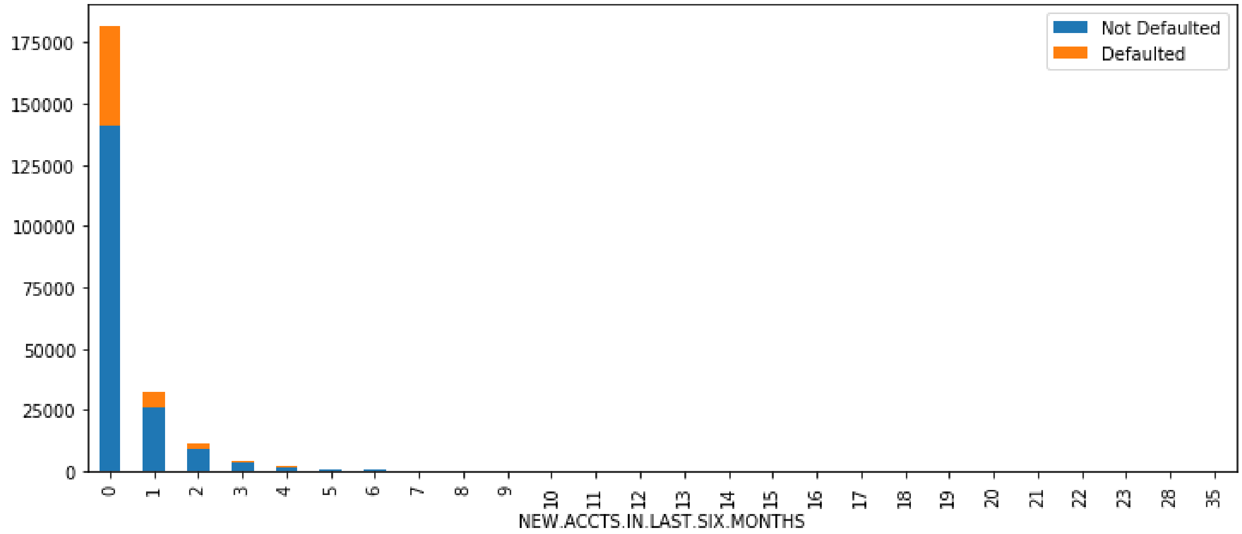
total amount that was disbursed for all the loans at the time of disbursement

2.2.19.7 Secondary Installment Amount

EMI Amount of the primary loan

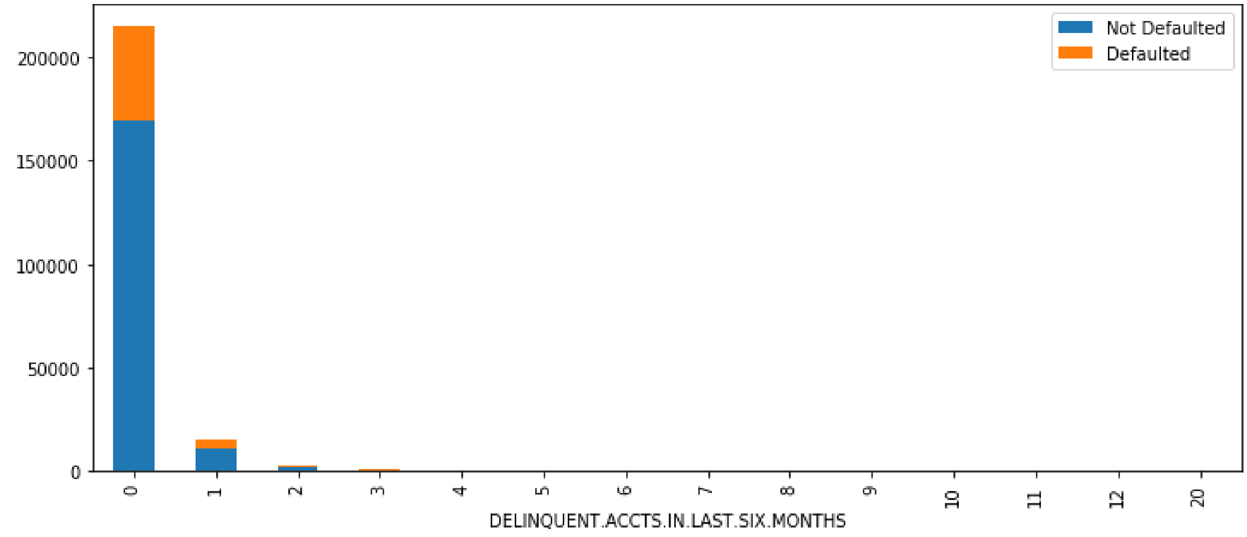
2.2.20 New Accounts in Last Six Months

New loans taken by the customer in last 6 months before the disbursement



2.2.21 Delinquent Accounts in Last Six Months

Loans defaulted in the last 6 months



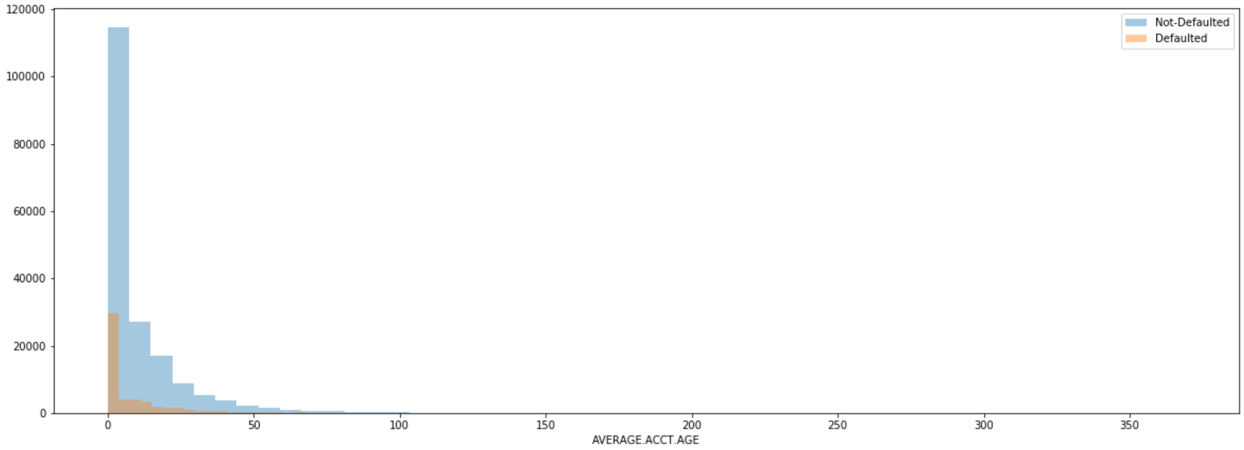
2.2.22 Average Account Age

Average loan tenure

df['AVERAGE.ACCT.AGE']

```
0          0yrs 0mon
1         1yrs 11mon
2          0yrs 0mon
3          0yrs 8mon
4          0yrs 0mon
             ...
233149     1yrs 9mon
233150     0yrs 6mon
233151     0yrs 0mon
233152     0yrs 0mon
233153     0yrs 0mon
Name: AVERAGE.ACCT.AGE, Length: 233154, dtype: object

df['AVERAGE.ACCT.AGE']=df['AVERAGE.ACCT.AGE'].apply(lambda x:(re.sub('[a-z
]','',x)).split())
df['AVERAGE.ACCT.AGE']=df['AVERAGE.ACCT.AGE'].apply(lambda x:int(x[0])*12+
int(x[1]))


df['AVERAGE.ACCT.AGE']
0          0
1         23
2          0
3          8
4          0
          ..
233149    21
233150     6
233151     0
233152     0
```

```
233153     0
Name: AVERAGE.ACCT.AGE, Length: 233154, dtype: int64
```



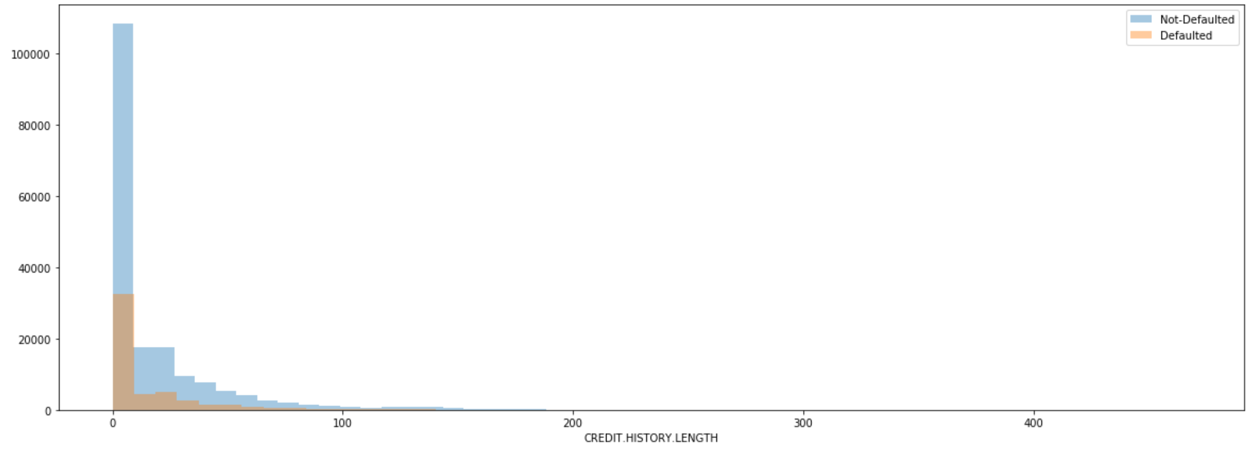
2.2.23 Credit History Length

Duration of the loan

df['CREDIT.HISTORY.LENGTH']

```
0           0yrs 0mon
1          1yrs 11mon
2           0yrs 0mon
3           1yrs 3mon
4           0yrs 0mon
              ...
233149      3yrs 3mon
233150      0yrs 6mon
233151      0yrs 0mon
233152      0yrs 0mon
233153      0yrs 0mon
Name: CREDIT.HISTORY.LENGTH, Length: 233154, dtype: object


Changing years and month format to months


df['CREDIT.HISTORY.LENGTH']=df['CREDIT.HISTORY.LENGTH'].apply(lambda x:
                                                                   (re.sub
('[a-z]','',x)).split())
df['CREDIT.HISTORY.LENGTH']=df['CREDIT.HISTORY.LENGTH'].apply(lambda x:
                                                                   int(x[0
])*12+int(x[1]))

df['CREDIT.HISTORY.LENGTH']

0          0
1         23
2          0
3         15
4          0
          ..
233149    39
233150     6
233151     0
```

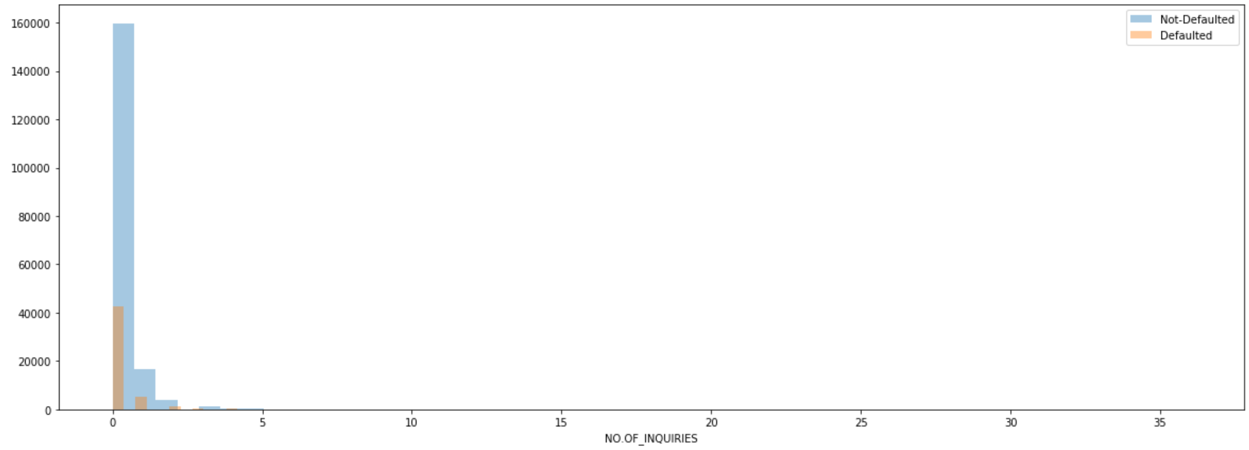
```
233152      0
233153      0
Name: CREDIT.HISTORY.LENGTH, Length: 233154, dtype: int64
```



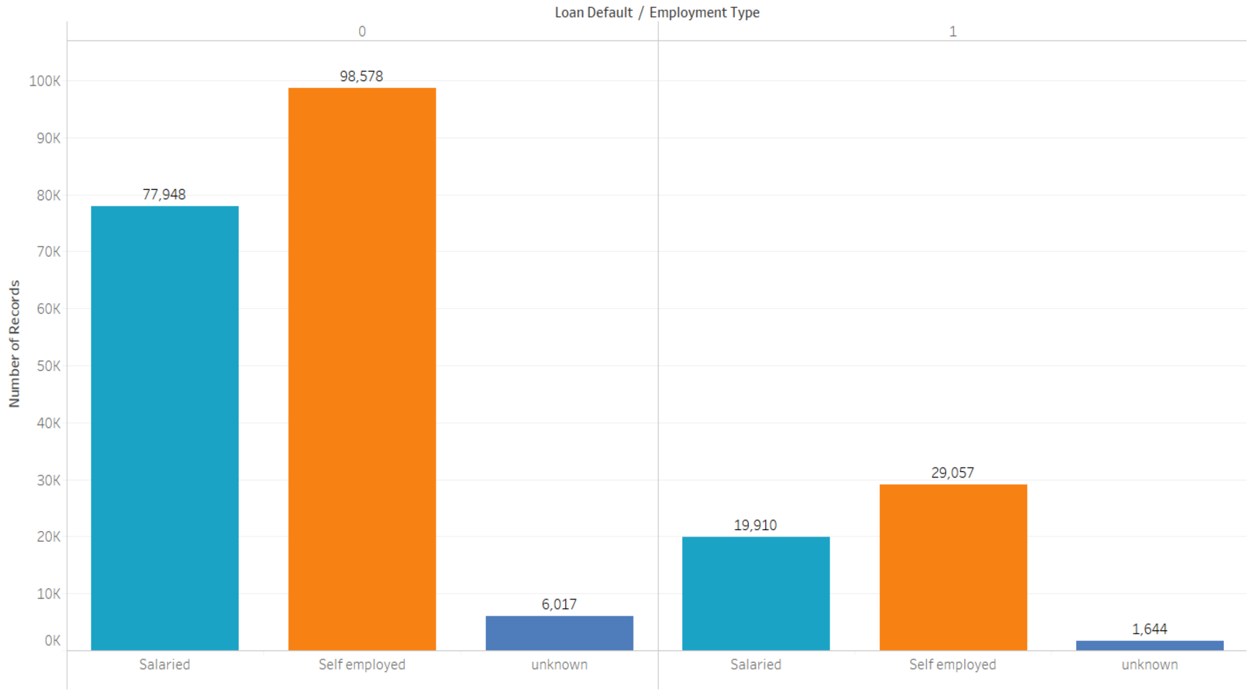
2.2.24 Number of Inquires

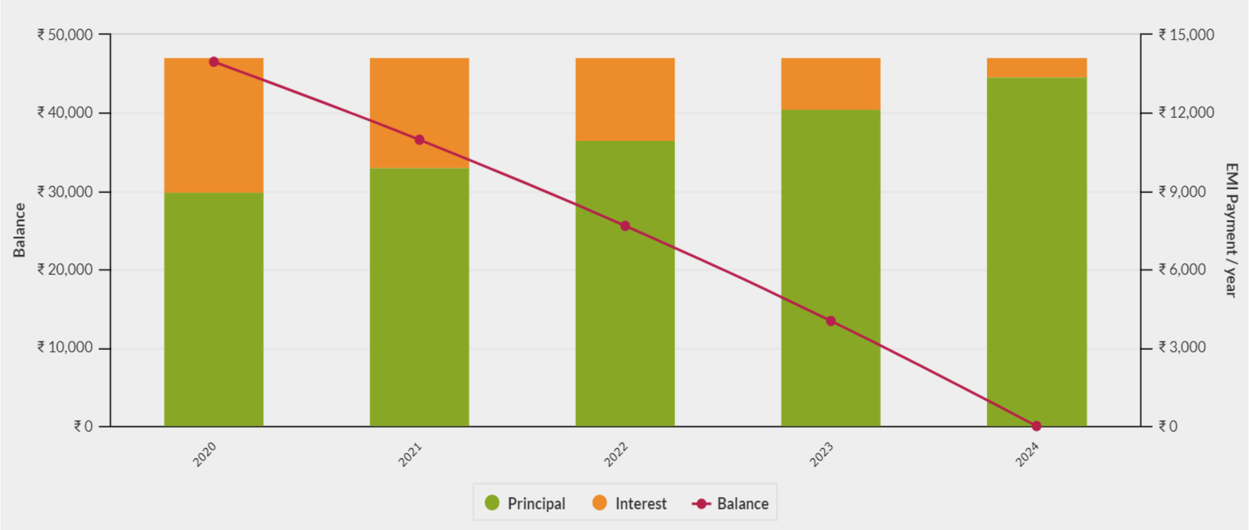
Enquiries

done by the customer for loans

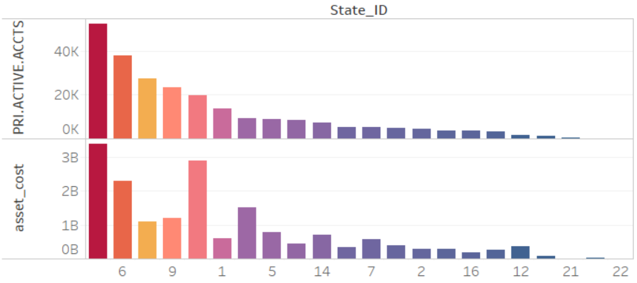


3. Exploratory Data Analysis

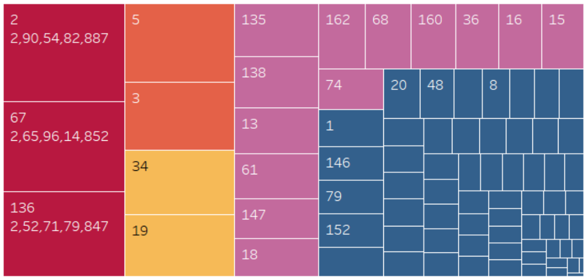




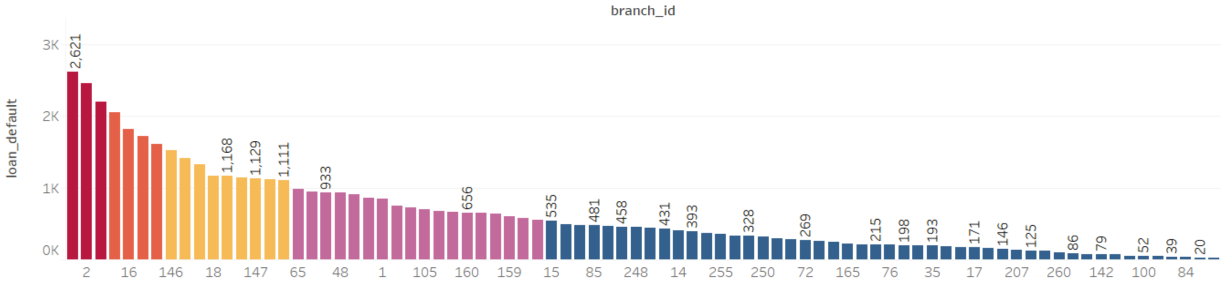
State vs asset value and active accts of borrowers

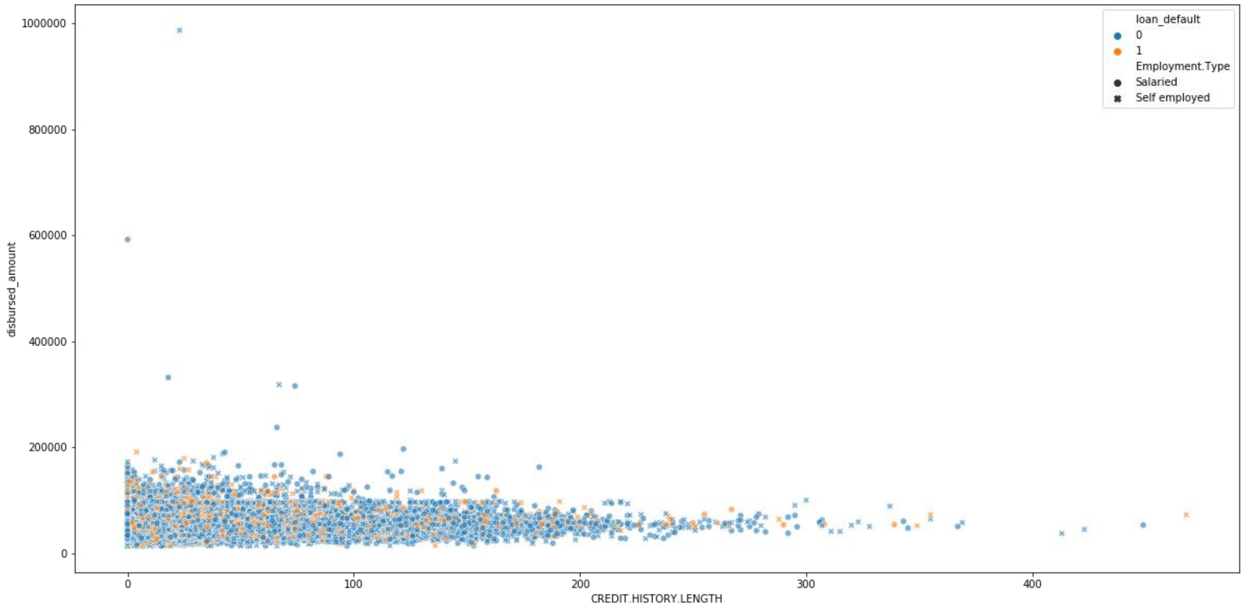


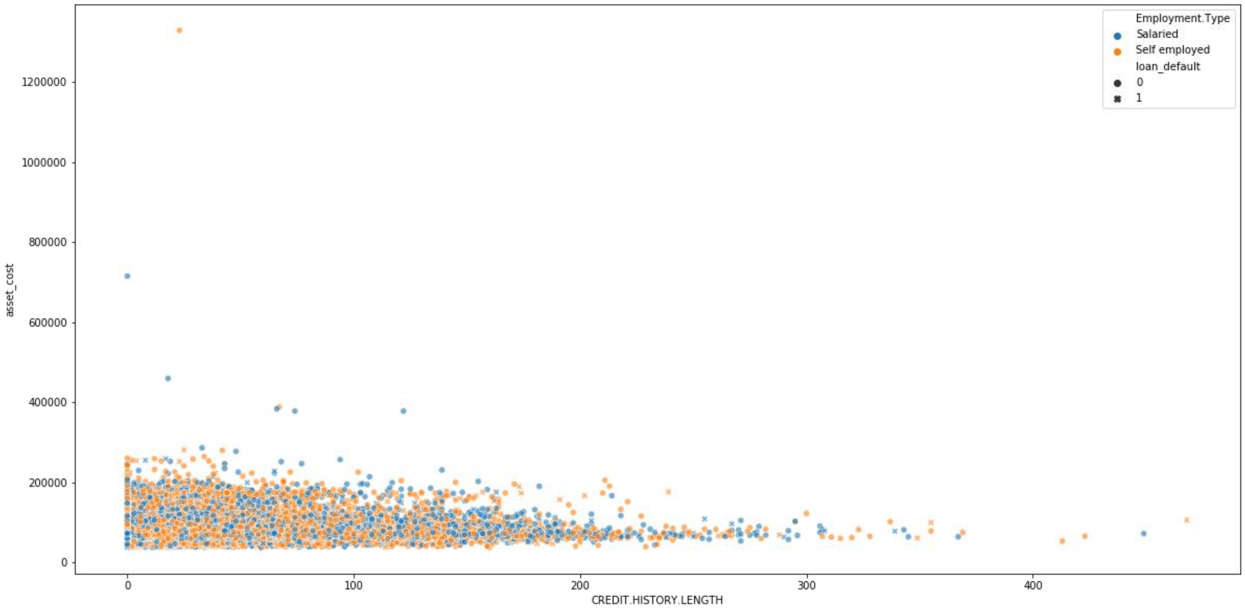
Tree map-Branch vs primary current balance

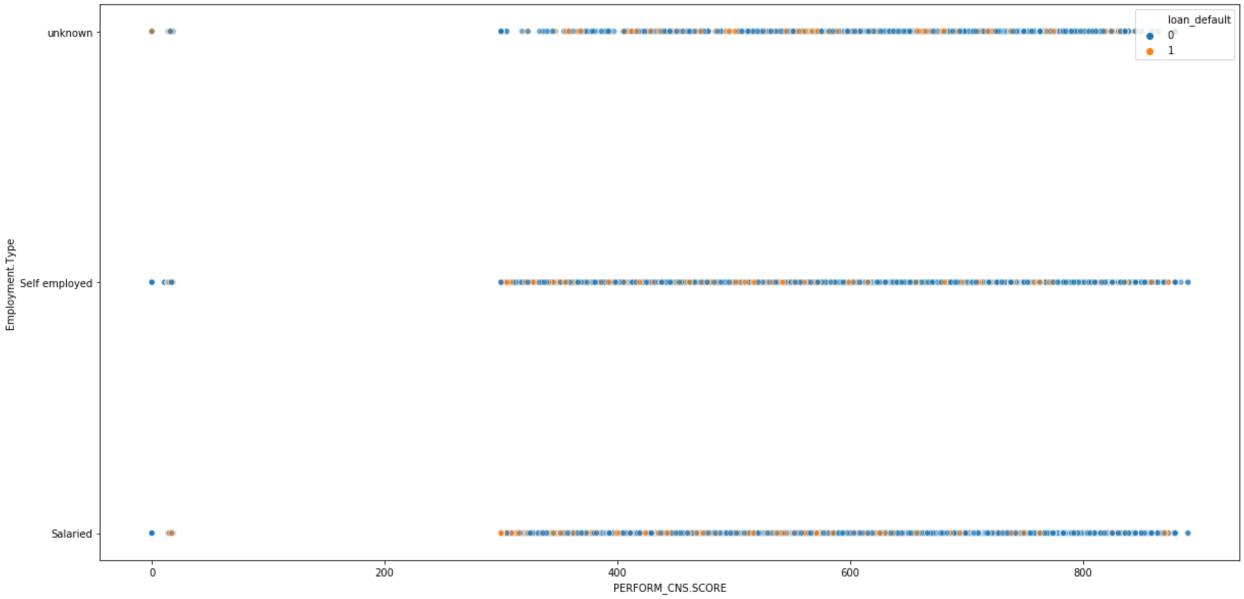


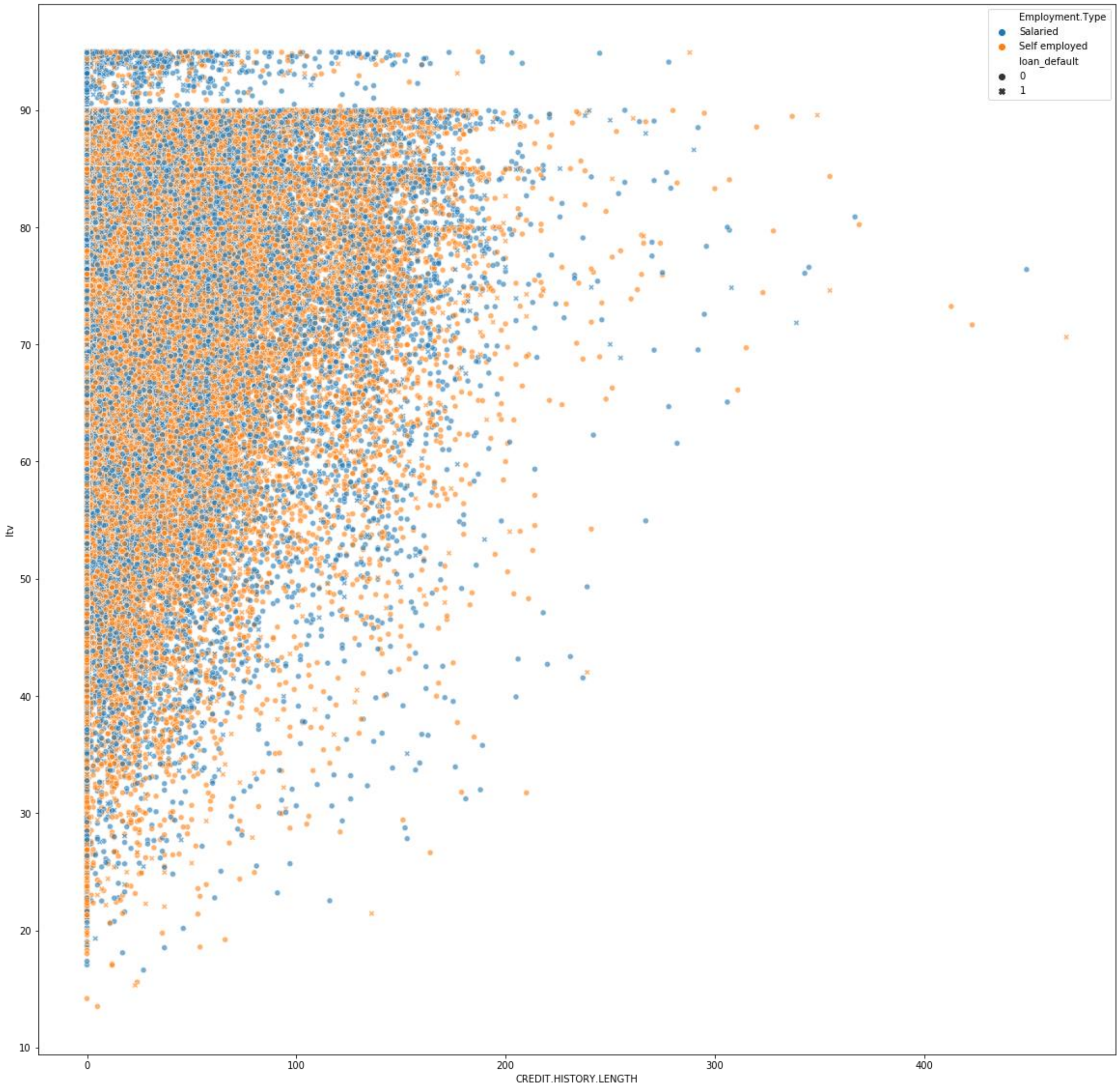
Branch id vs loan defaults

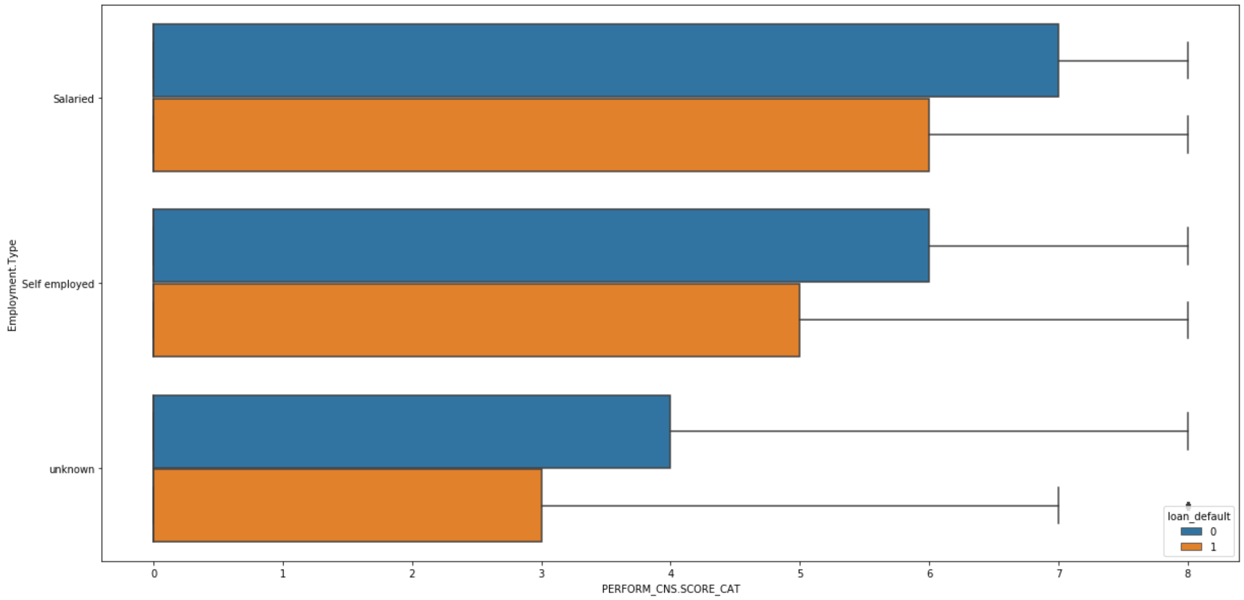


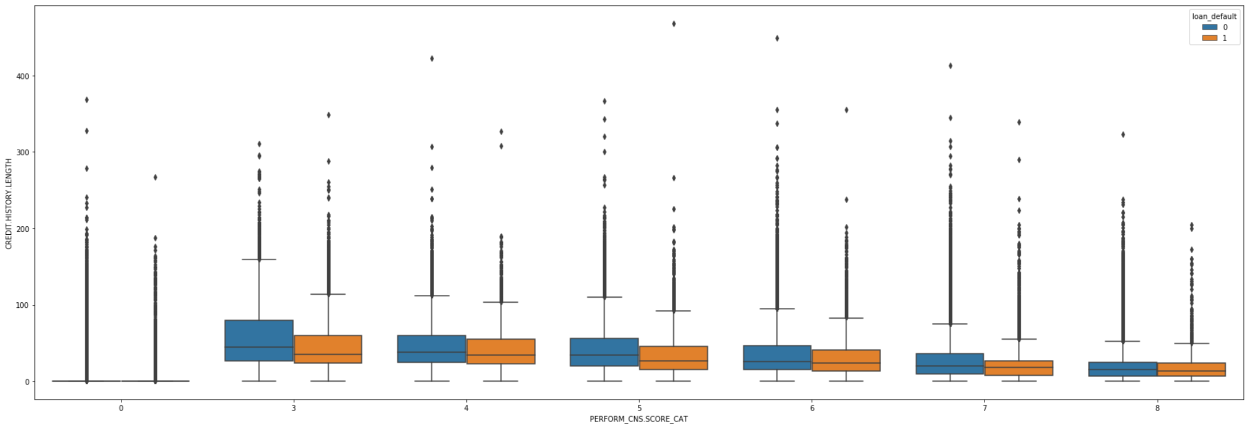


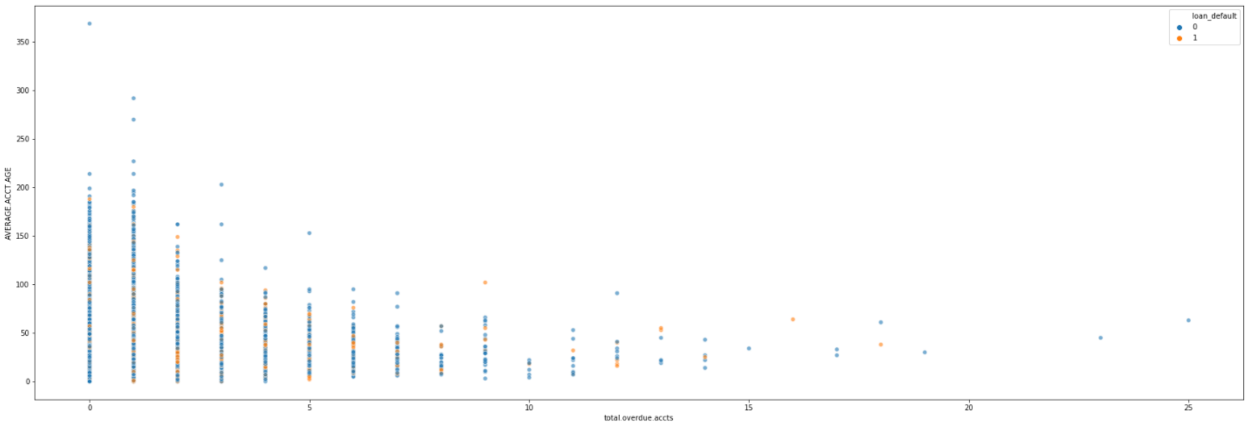

Employment.Type
Salaried
Self employed
loan_default
0
1
1200000
1000000
800000
600000
400000
200000
0
asset_cost
0
100
200
300
400
CREDIT.HISTORY.LENGTH

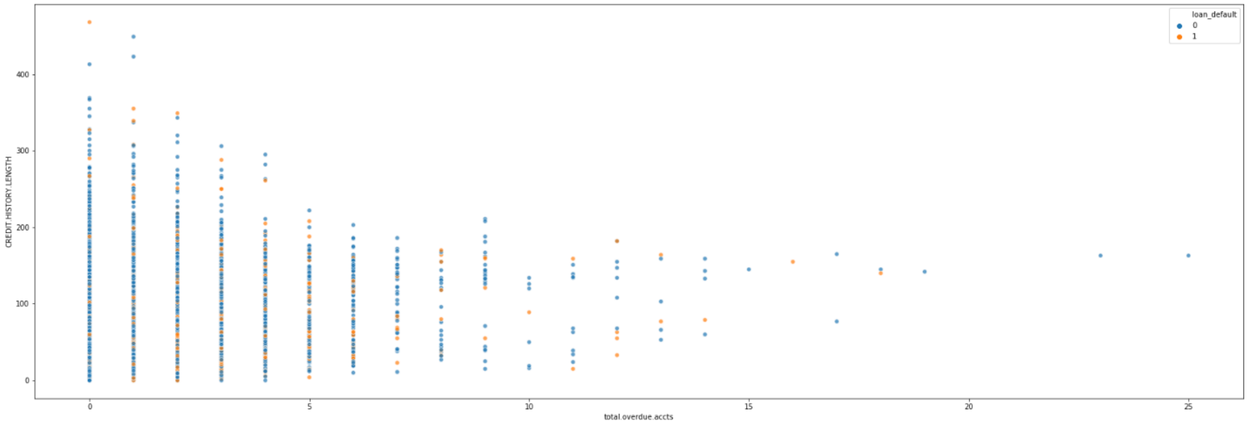

loan_default
0
1
unknown
Self employed
Salaried
Employment.Type
0
200
400
600
800
PERFORM_CNS.SCORE


Employment.Type
Salaried
Self employed
loan_default
0
1
ltv
90
80
70
60
50
40
30
20
10
0
100
200
300
400
CREDIT.HISTORY.LENGTH


Salaried
Self employed
unknown
Employment.Type
0
1
2
3
4
5
6
7
8
PERFORM_CNS.SCORE_CAT
loan_default
0
1

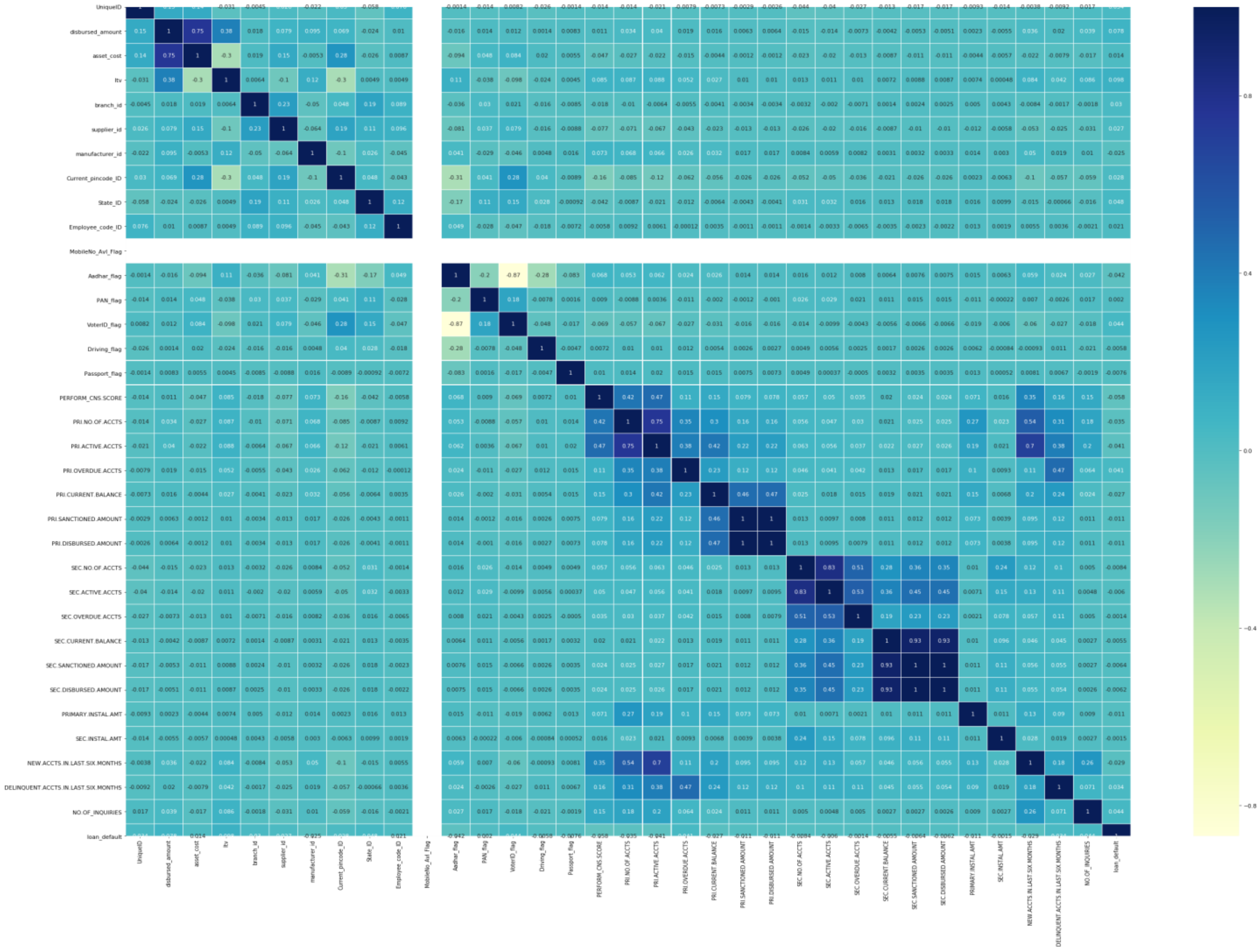

loan_default
0
1
400
300
200
100
0
CREDIT.HISTORY.LENGTH
0
3
4
5
6
7
8
PERFORM_CNS.SCORE_CAT


loan_default
0
1
350
300
250
200
150
100
50
0
AVERAGE.ACCT.AGE
0
5
10
15
20
25
total.overdue.accts



3. Feature Engineering

3.1 Correlation Plot

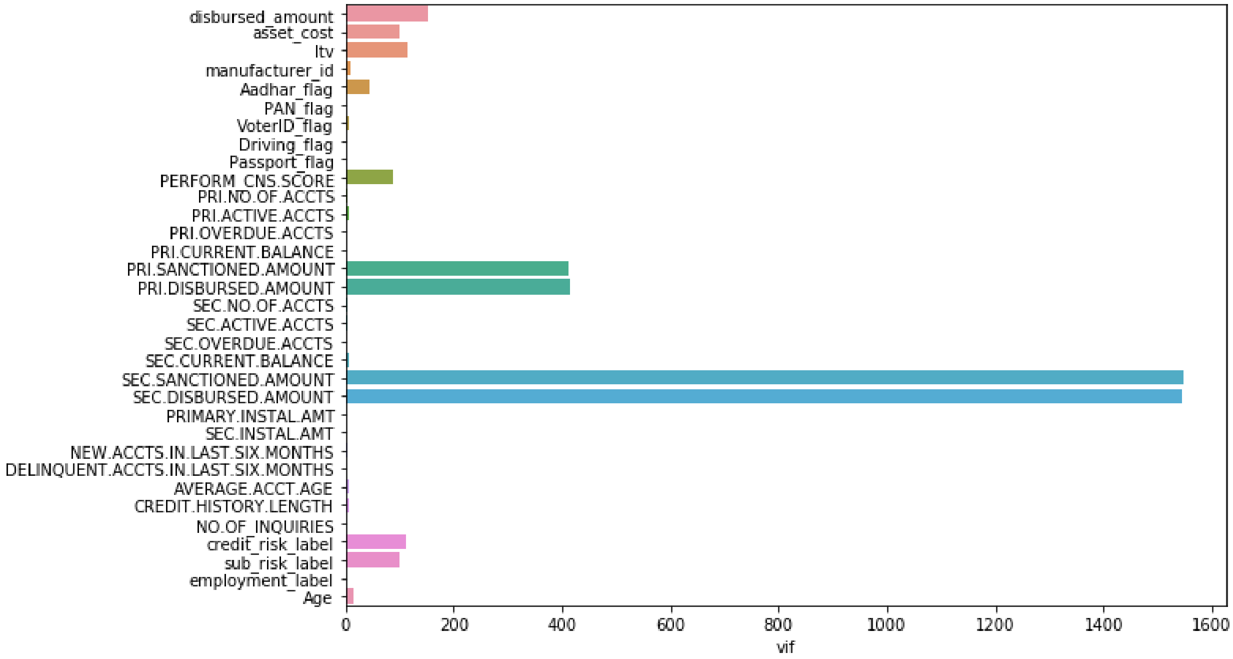


Since there is no Correlation in Mobile Avl Flag and Unique in the dataset. Removing Mobile Avl Flag and Unique ID from the dataset.

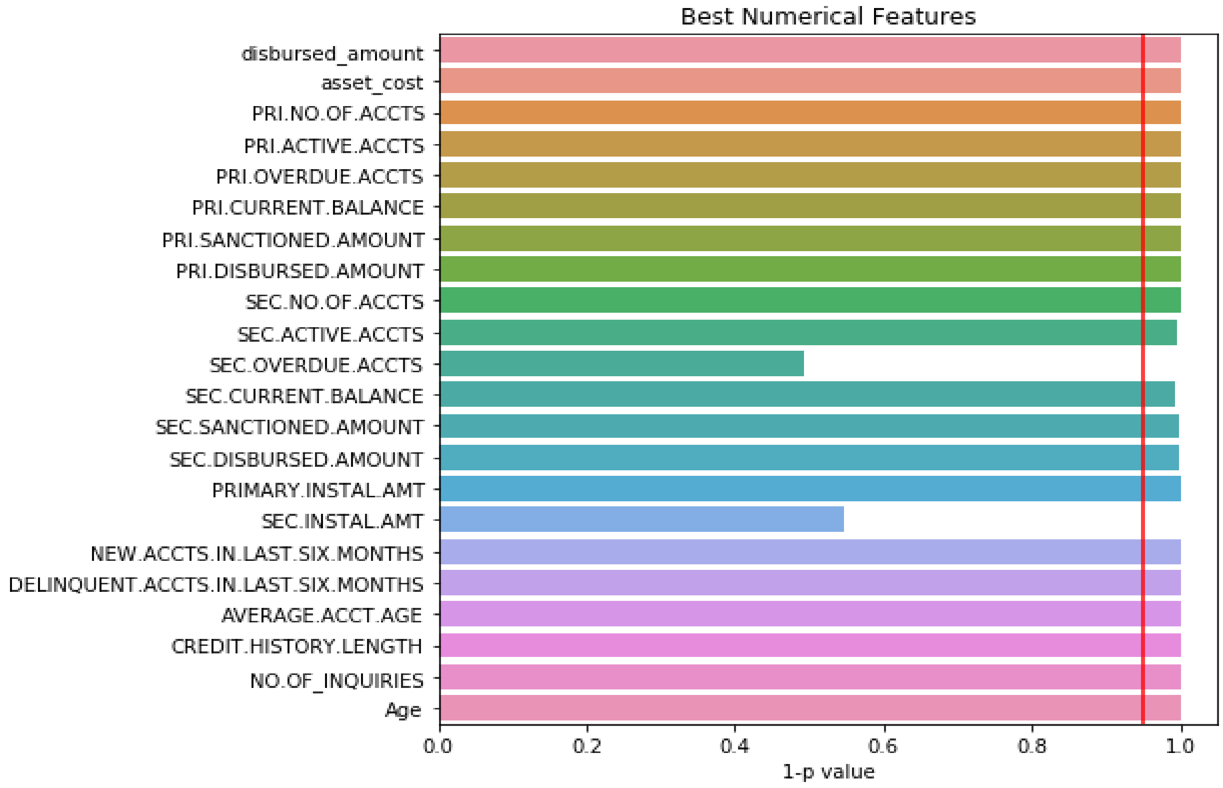
3.2 Multicollinearity Check

Checked multicollinearity with the help of variance influence factor(vif).

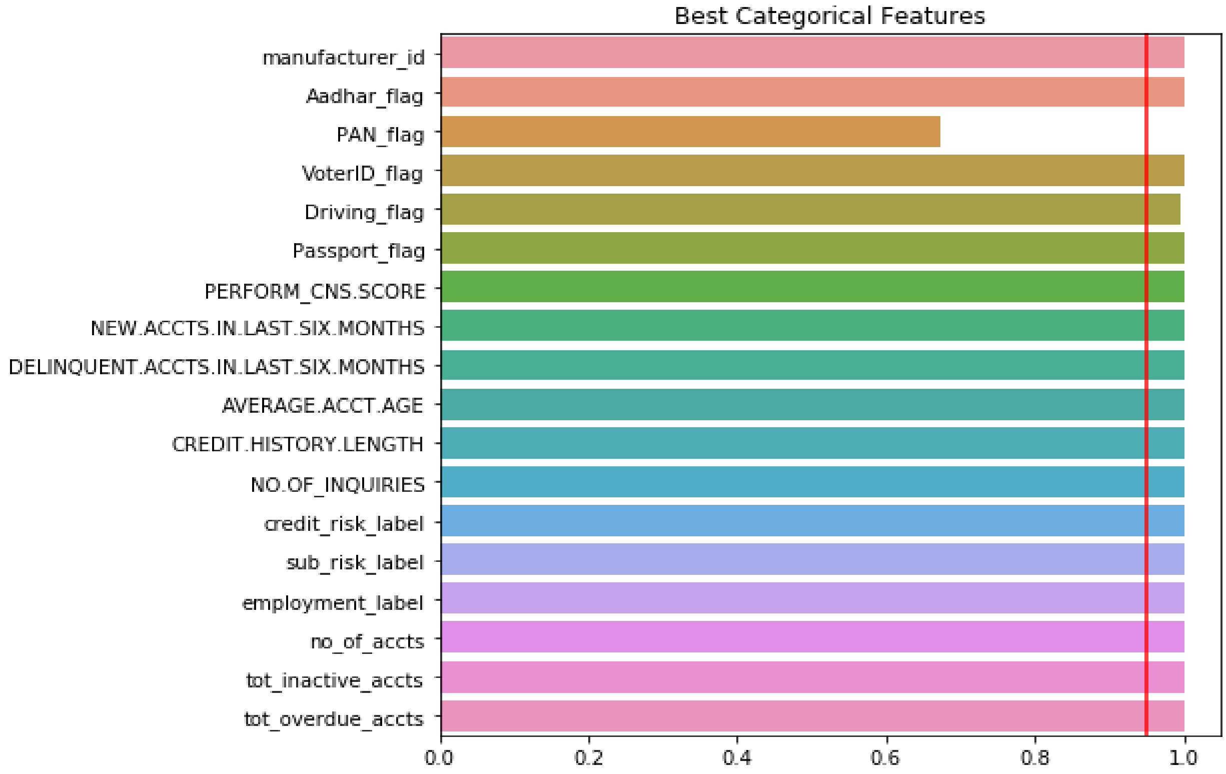
```
disbursed_amount 8.74561717585509e-309
asset_cost 5.716223071536896e-12
PRI.SANCTIONED.AMOUNT 4.798158421546997e-08
SEC.NO.OF.ACCTS 5.1490255376949666e-05
PRI.NO.OF.ACCTS 9.576575137572993e-66
PRI.DISBURSED.AMOUNT 7.176942237800462e-08
PRI.ACTIVE.ACCTS 3.448627479875517e-89
PRI.OVERDUE.ACCTS 9.138488408377107e-87
SEC.CURRENT.BALANCE 0.0075643427363124875
SEC.SANCTIONED.AMOUNT 0.002153062273491789
SEC.OVERDUE.ACCTS 0.5081054926877384
SEC.DISBURSED.AMOUNT 0.0025523226185338705
PRIMARY.INSTAL.AMT 2.958254960232989e-07
SEC.INSTAL.AMT 0.4546434321302706
NEW.ACCTS.IN.LAST.SIX.MONTHS 9.30229371021266e-46
DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS 3.2892517686894386e-62
AVERAGE.ACCT.AGE 5.261091482095756e-33
CREDIT.HISTORY.LENGTH 4.6500173864982836e-92
NO.OF_INQUIRIES 7.912566786376203e-99
```



3.3 Statistical Test for Numerical Columns

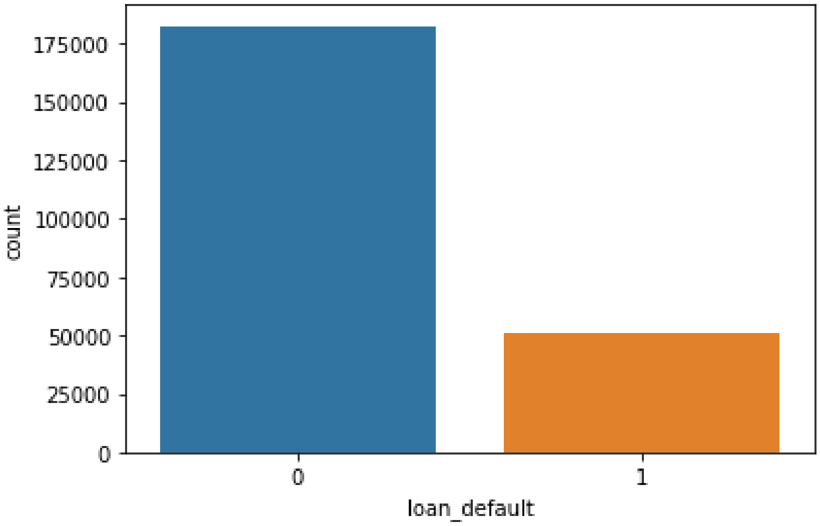


3.4 Statistical Test for Categorical Columns



3.5 Data Imbalance

Target Variable



Not-Defaulters 78.2%

Defaulters 21.3%

In the 233,546 rows in the dataset, which is highly imbalance. The model cannot will make a wrong prediction because of the data.

3.5.1 Under-sampling

```
not_default = df[df.loan_default==0]

default = df[df.loan_default==1]


not_default_downsampled = resample(not_default,

                      replace = True, # sample without replacement

                      n_samples = len(default), # match minority n

                      random_state = 0)

downsampled = pd.concat([not_default_downsampled, default])

downsampled['loan_default'].value_counts()
```

```
1    50611
0    50611
Name: loan_default, dtype: int64
```

From the 233,456 rows in the dataset, we are losing over 50% of the data, so decided not to implement Downsampling

3.5.2 Synthetic Minority Over-sampling Technique (SMOTE)

```
from sklearn.utils import resample

from imblearn.over_sampling import SMOTE

print("X shape",X.shape)

print('y shape',y.shape)
```

```
X shape (233154, 39)
y shape (233154,)
```

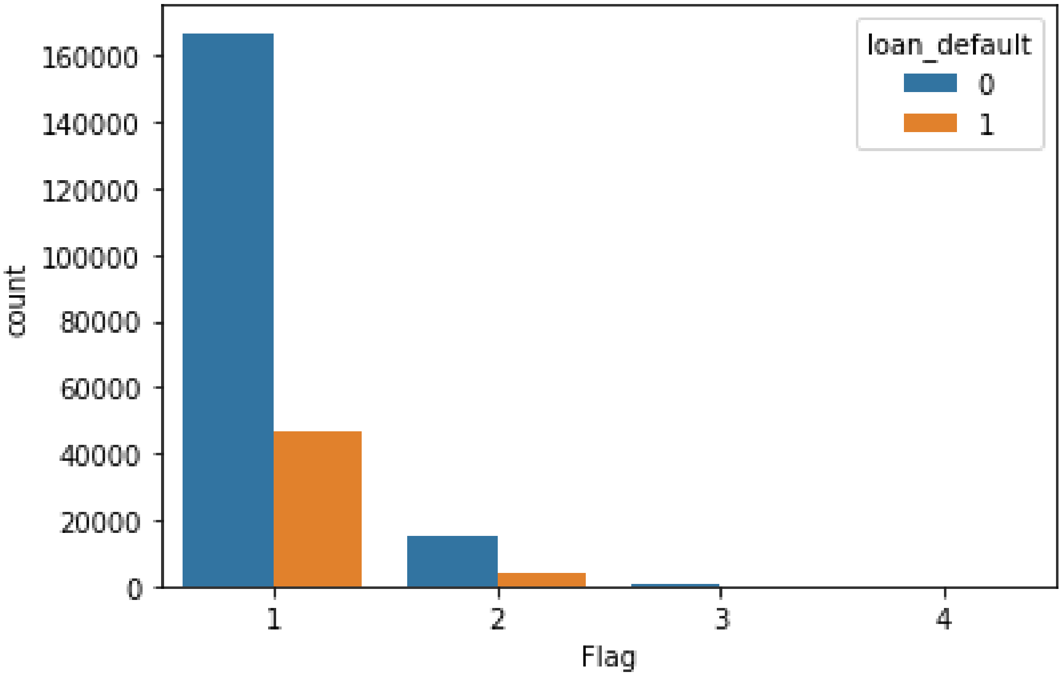
```
sm = SMOTE(random_state=0)

X_smote,y_smote = sm.fit_sample(X,y)



print("X shape",X_smote.shape)

print('y shape',y_smote.shape)
```

```
X shape (365086, 39)
y shape (365086,)
```

3.6 New Columns to be Added

3.6.1 Collapsing Flags to Total Flags

```
df['Flag']=df['Aadhar_flag'].astype('object')+df['PAN_flag'].astype('object')+df['VoterID_flag'].astype('object')+df['Driving_flag'].astype('object')+df['Passport_flag'].astype('object')
```

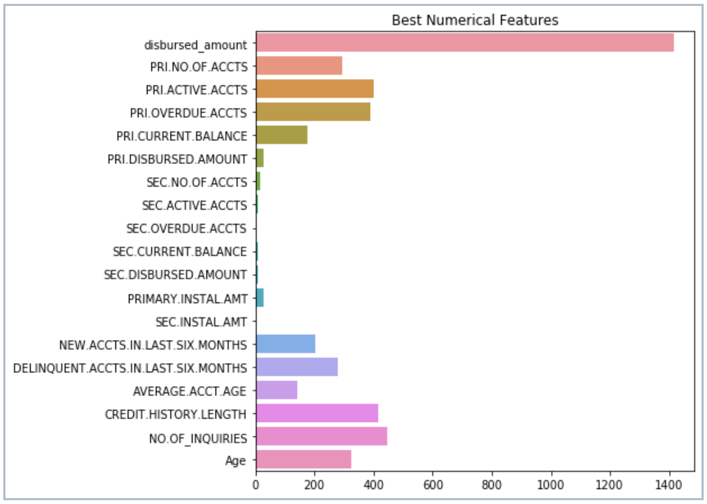


3.6.2 Total Attributes:

```
df.loc[:,'total.no.of.accts']=df['PRI.NO.OF.ACCTS']+df['SEC.NO.OF.ACCTS']

df.loc[:,'pri.inactive.accts']=df['PRI.NO.OF.ACCTS']-df['PRI.ACTIVE.ACCTS']

df.loc[:,'sec.inactive.accts']=df['SEC.NO.OF.ACCTS']-df['SEC.ACTIVE.ACCTS']

df.loc[:,'total.inactive.accts']=df['pri.inactive.accts']-df['sec.inactive.accts']

df.loc[:,'total.overdue.accts']=df['PRI.OVERDUE.ACCTS']+df['SEC.OVERDUE.ACCTS']

df.loc[:,'total.current.balance']=df['PRI.CURRENT.BALANCE']+df['SEC.CURRENT.BALANCE']

df.loc[:,'total.disbursed.amount']=df['PRI.DISBURSED.AMOUNT']+df['SEC.CURRENT.BALANCE']

df.loc[:,'total.sanctioned.amount']=df['PRI.SANCTIONED.AMOUNT']+df['SEC.SANCTIONED.AMOUNT']

df.loc[:,'total.installment']=df['PRIMARY.INSTAL.AMT']+df['SEC.SANCTIONED.AMOUNT']

# df.loc[:,'bal.to.disburse']=np.round((1+df['total.disbursed.amount'])/(1+df['total.current.balance']),2) # balance to disbursed anount ratio

df.loc[:,'pri.tenure']=(df['PRI.DISBURSED.AMOUNT']/(df['PRIMARY.INSTAL.AMT']+1)).astype(int)

df.loc[:,'sec.tenure']=(df['SEC.DISBURSED.AMOUNT']/(df['SEC.INSTAL.AMT']+1)).astype(int)

df.loc[:,'disburse.to.sanctioned']=np.round((1+df['total.disbursed.amount'])/(1+df['total.sanctioned.amount']),2)
```

4.Model Building

4.1 Feature Engineering



1. Feature Importance from 'SelectKBest' showed lower statistical importance for Secondary Account information. The Secondary Account information showed lower significance compared to Primary Account. So, instead of removing the column, these two columns were merged into one column.
2. The columns related to Balance, Sanction Amount and Disbursed Amount showed outliers, as well as many zero values (since 70% customers have no credit history). Standard scaler and Min-Max scaler have shown high effect of outlier as well zero values. To make those columns robust to outliers, we used Robust Scaler on those columns.
3. After using Robust Scaler, we still have many columns with higher number of zero value observations for customers with no credit history. So, to counter this, we made a new feature that counts missing features. This acts as a penalty factor for people with no credit history, which will be low for people with a credit history.
4. The dataset had imbalance in the target column. So, we made iterations by down-sampling and oversampling, both which resulted in loss of Data. We finally decided to use SMOTE library to handle this imbalance.

4.1 Base Model

Model	**Train**	**Test**	**Precision**	**Recall**	**F1**
Logistic	0.783	0. 7827	0.0	0.0	0.0
Decision Tree (Entropy)	1	0. 672	0.26	0.278	0.268
Decision Tree (Gini)	1	0.67	0.263	0.285	0.274
Random Forest	1	0.781	0.4458	0.0351	0.0651
Naïve Bayes	0.774	0.773	0.912	0.004	0.009
Bagging Classifier	0.977	0.765	0.336	0.083	0.133
Adaboost	0.7829	0.7822	0.456	0.012	0.0275
XGBoost	0.7833	0.7828	0.5238	0.0028	0.0057

4.2 Model Iterations

We made a base model with Logistic Regression. After looking at its performances and the required feature engineering, other classification models were used to find the best model and best parameters.

```
lr=LogisticRegression()
nb= GaussianNB()
dt=DecisionTreeClassifier(random_state=0, criterion='entropy')
ranforest=RandomForestClassifier(random_state=0,n_estimators=43,criterion='gini')
bag=BaggingClassifier(n_estimators=26, random_state=0)
adab=AdaBoostClassifier(n_estimators=10)
scores=[]
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
for name,model in models:
    kf = KFold(n_splits=3, shuffle=True, random_state=0)
    cv_score=cross_val_score(model,xs,y,cv=kf,scoring='f1_weighted')
    scores.append(cv_score)
    print(name)
    print(np.mean(scores), " ",np.var(scores,ddof=1))
fig=plt.figure()
ax=fig.add_subplot(111)
plt.boxplot(scores)
plt.show()
```

We weren’t satisfied with these models, so we tried XGBoost. Below are the performances of all the models used.

Model	**Train**	**Test**	**Precision**	**Recall**	**F1**
Logistic	0.621	0.623	0.304	0.57	0.397
Decision Tree (Entropy)	1	0.979	0.94	0.95	0.9522
Decision Tree (Gini)	1	0.979	0.945	0.958	0.952
Random Forest	1	0.953	0.953	0.827	0.885
Naïve Bayes	0.577	0.583	0.263	0.512	0.348
Bagging Classifier	0.999	0.981	0.964	0.949	0.9566
Adaboost	0.885	0.859	0.636	0.821	0.717
XGBoost	0.962	0.955	0.927	0.862	0.893

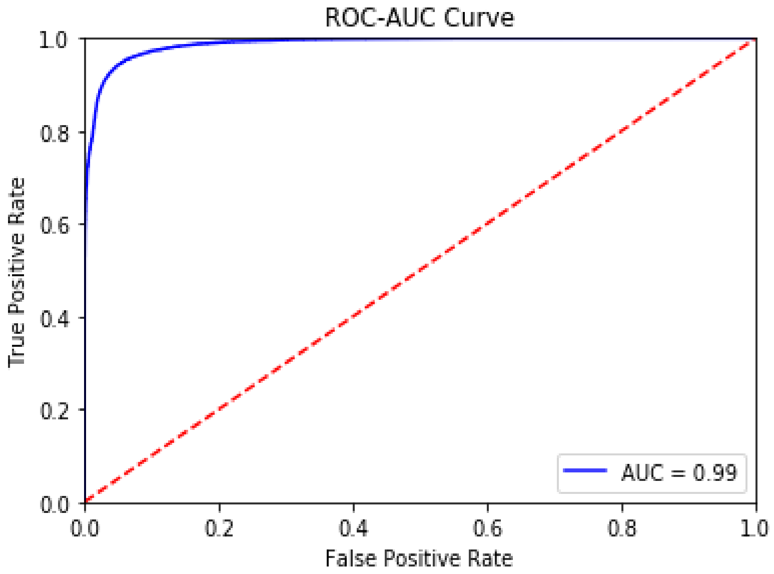
Also, we did Hyperparameter tuning for XGBoost to improve our ROC-AUC values. We got the following best parameters for XGBoost.

```
XGBClassifier( learning_rate =0.01, n_estimators=5000,  max_depth=9, min_child_weight=1,
 gamma=0.4, subsample=0.8, colsample_bytree=0.8, reg_alpha=0.005, objective= 'binary:logistic',
 nthread=4, scale_pos_weight=1, seed=27)
```

Following is the confusion matrix and ROC-AUC Curve

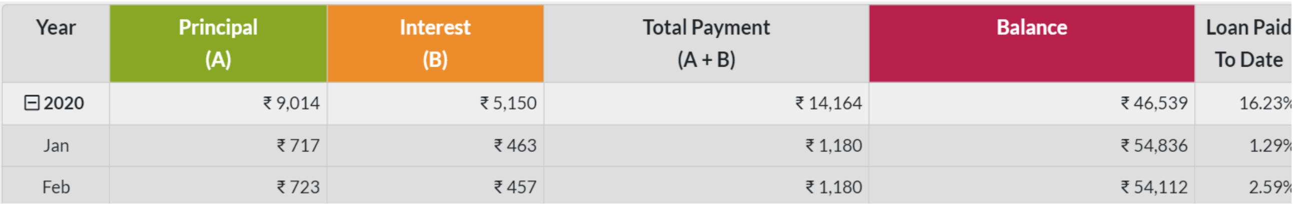
Confusion Matrix

	Actual Defaulters	Predicted Non-Defaulters
Predicted Defaulters	13096	1027
Predicted Non-Defaulters	2095	53729



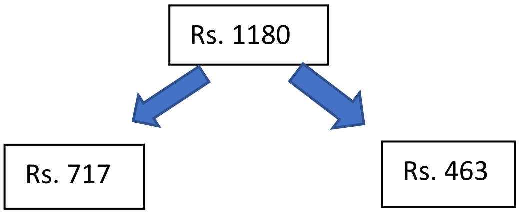
The final model used is XGBoost on the dataset.

5. Profit Generated by Loans



Year	Principal (A)	Interest (B)	Total Payment (A + B)	Balance	Loan Paid To Date
⊟ **2020**	₹ 9,014	₹ 5,150	₹ 14,164	₹ 46,539	16.23%
Jan	₹ 717	₹ 463	₹ 1,180	₹ 54,836	1.29%
Feb	₹ 723	₹ 457	₹ 1,180	₹ 54,112	2.59%

One part of the EMI is The Principal amount and the other part is the Interest amount which is the Bank's income.



Here Rs. 717 is the Principal part and Rs. 463 is the Interest part. This 463 Rs is the income of the bank.

6. Return on Investment

	Actual Defaulters	Predicted Non-Defaulters
Predicted Defaulters	13096	1027
Predicted Non-Defaulters	2095	53729

Potential Customers that are excluded by our model.

Non-Potential Customers that are given loan by our model.

Bank's income from one borrower: Rs 463

Without model -> 2,53,52,028(Profit)

- 70,33,433(Loss)

Net Profit = 1,83,18,595

With our model -> 2,48,76,527(Profit)

- 9,69,985 (Loss)

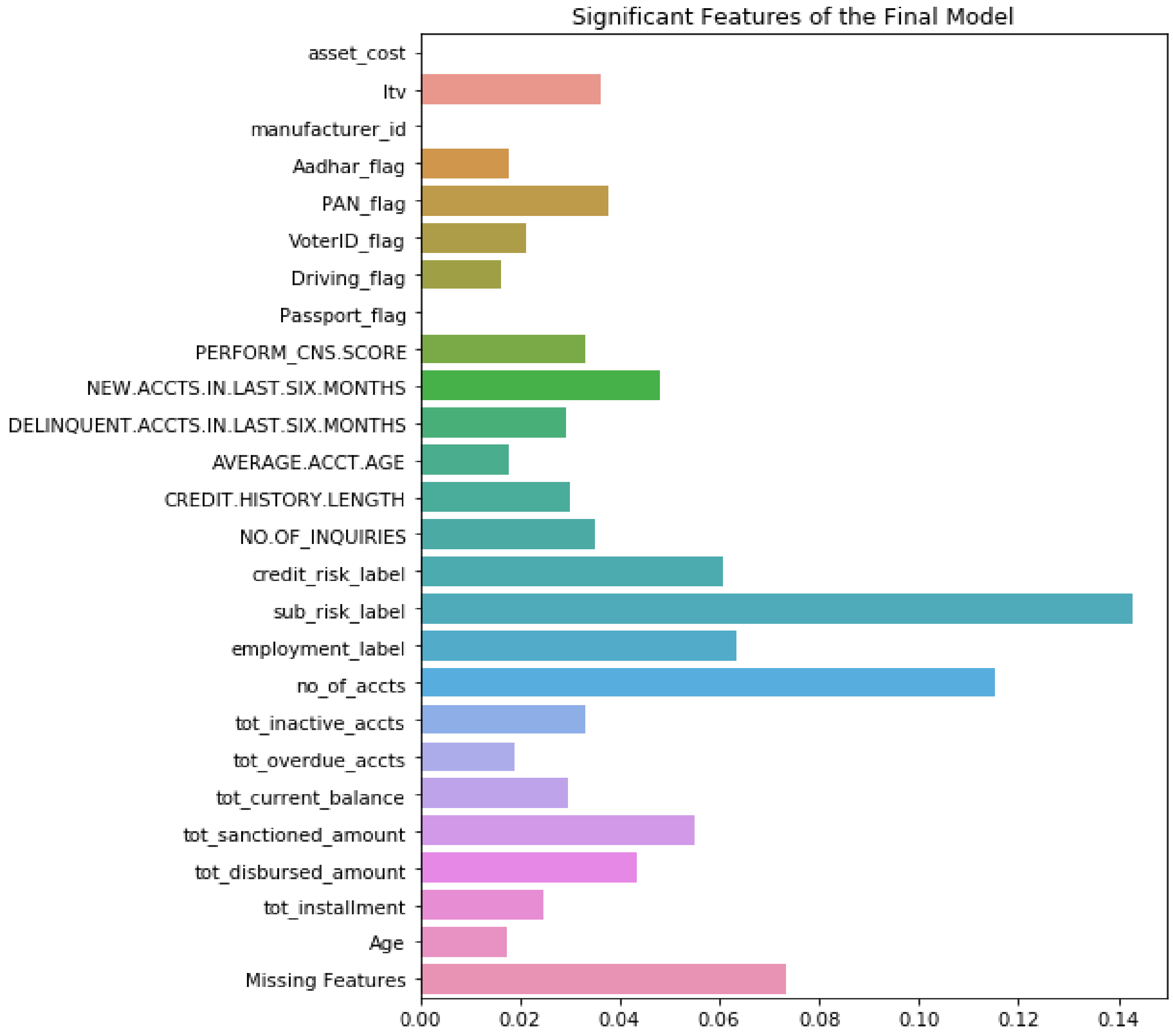
- 4,75,501 (Loss)

Net Profit = 2,34,31,041

50,00,000 more profit.

% profit increased = 0.167

7. Business Suggestions



Businesses are not known to take well to technical jargons and explanations while presenting the projects so instead of presenting a black box to feed data and take instructions we can suggest few features from the data that is provided to help them improve the quality of the way they conduct their business.

From the final model we can get the significance of all the features so we can concentrate on few of those features which contribute significantly to the target variable.

8. Project outcome:

1. We are able to reduce the Type-2 error, i.e., false classification of default. Due to this, we project an increase of 0.16% profit, i.e., Rs.2.34 Crores.
2. We've filtered branches which face higher default rates compared to other branches.
3. The age bracket of customers who default the most has also been observed in the project.
4. We use XGBoost model for Prediction of Loan EMI Default, and improved our model scores over the previous works.

9. Business outcome:

- Age group of 25-30 seems to be most vulnerable to defaulting their EMIs so we can rectify this trend by providing them with low interest loans over a longer period of time. Reducing the LTV ratio for these loans also reduces the risk behind these loans.
- We can observe that there are few branches performing significantly worse so providing branch specific solutions would prove provident.
- Few potential columns needed:
    - Interest Rate
    - Loan period
    - New Vehicle
    - Age of Vehicle

10. References and Bibliography

- **Dataset Source:** https://www.kaggle.com/mamtadhaker/lt-vehicle-loan-default-prediction

- **Research Paper on this dataset:**

1. Sumit Agarwal & Brent W. Ambrose & Souphala Chomsisengphet. "Determinants of automobile loan default and prepayment," Economic Perspectives, Federal Reserve Bank of Chicago, issue qiii, pages 17-28, 2008.
2. Alex Addae-Korankye, Causes and Control of Loan Default/Delinquency in Microfinance Institutions in Ghana Vol. 4, No. 12; December 2014
3. Agrawal, Mohit & Agrawal, Anand & Raizada, Dr. Abhishek. PREDICTING DEFAULTS IN COMMERCIAL VEHICLE LOANS USING LOGISTIC REGRESSION: CASE OF AN INDIAN NBFC. IJRCM. 5. 22-28, 2014.

- **Websites Referred:**

1. https://blog.bankbazaar.com/car-loan-default-what-when-and-how/
2. https://www.ltfs.com/companies/lnt-finance/two-wheeler-loans-detail.html

Github Repository for our project

https://github.com/xavierigneous/Vehicle-Loan-Default-Prediction