Loan Prediction

Light Commercial Vehicle

Alok Ojha
Samyak Sheth

Objective

This project uses various statistical and machine learning concepts in order to identify the bad and the good customers from the existing data for light commercial vehicle loans. A Logistic Regression Model is made to predict whether a customer is a good customer or a bd customer.
So as to decide for which customer giving loans will be safe.

Sub objectives - Flow of the project

- Performing Exploratory Data Analysis on numerical and categorical columns
- Understanding the trend in the data
- Understanding the trend in customer behaviour
- Variable Selection
- Building a logistic regression model
- Testing and Training reports for the model
- Ks and PSI Stats for the final model

Information Value and Weight of Evidence

A Information Value (IV) and Weight of Evidence (WoE) are the two most used concepts for variable selection and variable transformation respectively. Information Value helps quantify the predictive power of a variable in separating the Good Customers from the Bad Customers.

IV

Information Value gives a measure of how variable X is good in distinguishing between a binary response (e.g. good and bad) in some target variable Y. Low Information Value of a variable X means that it may not classify the target variable on a sufficient level and should be removed as an explanatory variable.

Since Information value gives us the importance of a variable it is a very important tool that can be used for variable reduction i.e. we can remove those variable form our analysis which have a very low IV score. For example- variables where IV < 0.02 (2 percent) can be removed.

WoE

WoE is calculated by taking the natural logarithm (log to base e) of the ratio of %Good by %Bad. The weight of evidence tells the predictive power of an independent variable in relation to the dependent variable.

A positive WoE means that %Good Customers> %Bad Customers, and vice-versa when the WoE is negative.

Weight of Evidence (WOE) helps to transform a continuous independent variable into a set of groups or bins

Bad rate analysis

Terms used in the further analysis

1. GoodProp - Stands for Good Proportion. It is the ratio of the number of good customer in a category(bin) to the total number of good customers.(Expressed as percentage)
2. BadProp - Stands for Bad Proportion. It is the ratio of the number of bad customer in a category(bin) to the total number of bad customers.(Expressed as percentage)
3. BadRate - It is the percentage of number of bad customers out of the total customers in that particular bin.
4. PopProp - Refers to the percentage of the total number of customers in that bin to the total number of customers in the dataset.

A bad rate analysis for each variable is performed. This process involves binning the data into bins and then getting a general idea of how the trend of the data is according to the target variable. The above mentioned terms are then calculated for each bin of each variable. Finally, an IV value is found that helps us select only a set of variables from all the groups. The selection for all the variables is done on the basis of the IV values among all the variables in the group. The one with the highest value is selected.

Variables selected

- Location(IV:22.55%)from geographical variable group
- Borrower_age (IV: 15.50%) from the age variables
- Agr_value (IV: 23.44%) from the amount variables
- No_vehicles_financed_kmbl(5.11%) from financed vehicles variable group
- Max_hgh_crd_amt_all_cv_lns (IV:11.61) from maximum credit amount variable group
- Total_no_closed_loan(IV:22.47%) from total loans variable group
- Total_no_secured_loan(IV:15.60%) from types of loans variable group
- No_of_closed_cv_loans(IV:20.57%) from total cv loans variable group
- Pk_delay_across all loans(IV:23.30%) from delay for all loans group
- Pk_delay_across_all_cv_loans(IV:14.30%) from delay for all cv loans group
- Free_vehicles_fleet (IV:10.02%)from fleet vehicle variable group
- Tm_frt_cv_loan(IV:17.65%) from past cv loans details group
- Cibil_score(IV:10.52%)

Out of the total 48 independent variables, these 13 variables are selected on the basis of bad rate and IV analysis. Further ahead in the process, only 6 most important variables out of these will be selected using the random forest variable importance metric.

Model Data

Subsetting the data for the model

In [92]:
```
cols=["Location","Borrower_age_bin","Agr_value_bin","No_vehicles_financed_kmbl_bin","Max_hgh_amt_all__cv_lns_bin",
      "Total_no_closed_loan_bin", "Total_no_secured_loan_bin","No_of_closed_cv_loans_bin",
      "Pk__delay_across_all_loans_bin","Pk_delay_across_all_cv_loans_bin","Free_vehicles_fleet_bin",
      "Tm_lst_cv_loan_bin","Cibil_score_bin","Flag"]
model_data=data[cols]
```

In [93]:
```
model_data.head()
```

Out[93]:

	Location	Borrower_age_bin	Agr_value_bin	No_vehicles_financed_kmbl_bin	Max_hgh_amt_all__cv_lns_bin	Total_no_closed_loan_bin	Total_no_secured_loan_b
0	Group2	>41	> 500000 & <=1000000	<=0	>600000	>8	
1	Group4	>41	<=500000	<=0	<=100000	>1&<=8	>1&<
2	Group4	>41	<=500000	>0&<=2	>600000	<=1	<
3	Group2	>41	> 500000 & <=1000000	>0&<=2	>100000&<=600000	>1&<=8	>1&<
4	Group3	<=41	> 500000 & <=1000000	>2	>600000	>8	

Encoding the Variables

In [95]:
```
1 codes
```

Out[95]:
```
{'Location': {'Group1': 3, 'Group2': 2, 'Group3': 1, 'Group4': 0},
 'Borrower_age_bin': {'<=41': 1, '>41': 0},
 'Agr_value_bin': {'<=500000': 0, '> 500000 & <=1000000': 1, '>1000000': 2},
 'No_vehicles_financed_kmbl_bin': {'<=0': 2, '>0&<=2': 1, '>2': 0},
 'Max_hgh_amt_all__cv_lns_bin': {' <=100000': 2,
  ' >100000&<=600000': 1,
  '>600000': 0},
 'Total_no_closed_loan_bin': {'<=1': 2, '>1&<=8': 1, '>8': 0},
 'Total_no_secured_loan_bin': {'<=1': 2, '>1&<=5': 1, '>5': 0},
 'No_of_closed_cv_loans_bin': {' <=1': 2, ' >1&<=5': 1, '>5': 0},
 'Pk__delay_across_all_loans_bin': {' <=102': 0, '>102': 1},
 'Pk_delay_across_all_cv_loans_bin': {' <=101': 0, '>101': 1},
 'Free_vehicles_fleet_bin': {'<=1': 1, '>1': 0},
 'Tm_lst_cv_loan_bin': {' <=365': 2, ' >365&<=730': 1, '>730': 0},
 'Cibil_score_bin': {'<=300': 2, '>300&<=750': 1, '>750': 0}}
```

In [96]:
```
1 model_data = model_data.replace(codes)
2 model_data.head()
```

Out[96]:

	Location	Borrower_age_bin	Agr_value_bin	No_vehicles_financed_kmbl_bin	Max_hgh_amt_all__cv_lns_bin	Total_no_closed_loan_bin	Total_no_secured_loan_b
0	2	0	1	2	0	0	
1	0	0	0	2	2	1	
2	0	0	0	1	0	2	
3	2	0	1	1	1	1	
4	1	1	1	0	0	0	

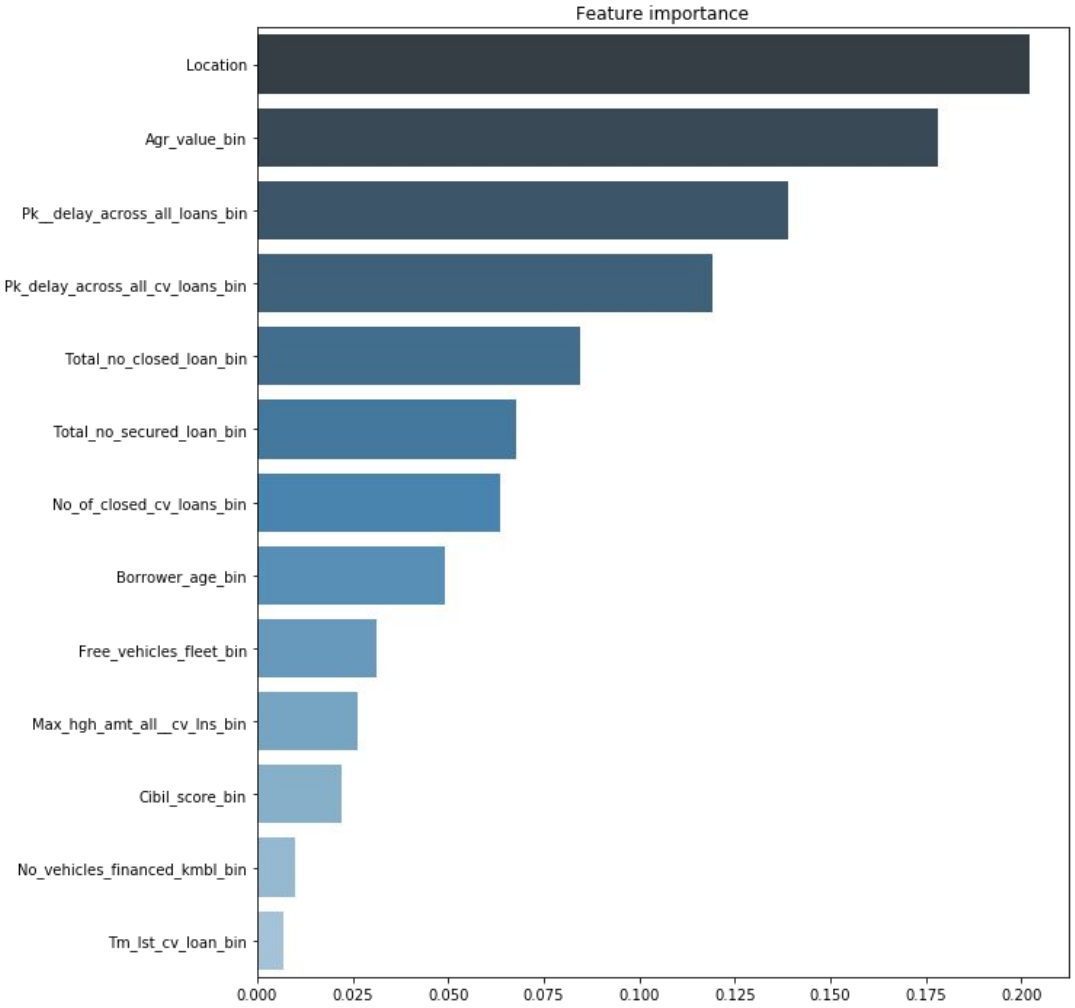
Model Making

Step 1 - Splitting the data into train and test sets

```
train_x = model_data.drop(["Flag"],axis=1)
train_y = model_data.Flag

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(train_x,train_y,test_size=0.2,random_state=132)
```

Step 2 - Passing the data to a Random Forest model to select the top 6 important features ¶

```
np.random.seed(123)
from sklearn.ensemble import RandomForestClassifier

# by default 100 tree will be form
rf = RandomForestClassifier(n_estimators=15,
                            criterion="gini",
                            max_depth=4,
                            min_samples_split=100,
                            min_samples_leaf=50,
                            max_features="sqrt")# n_estimators means number tree we want

rf.fit(X_train, y_train)
```



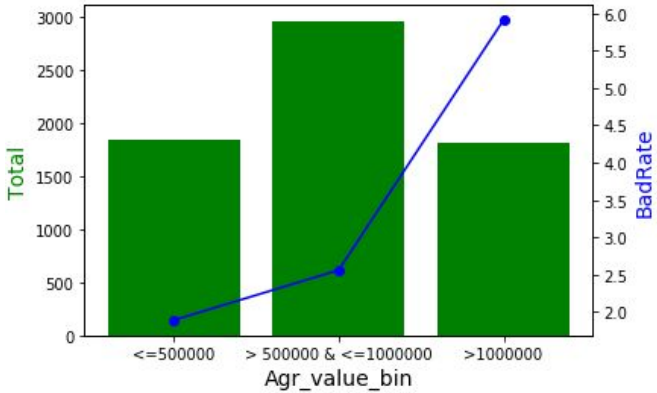
Using the importance chart, we select the top 6 variables based on its importance for all the further processes.

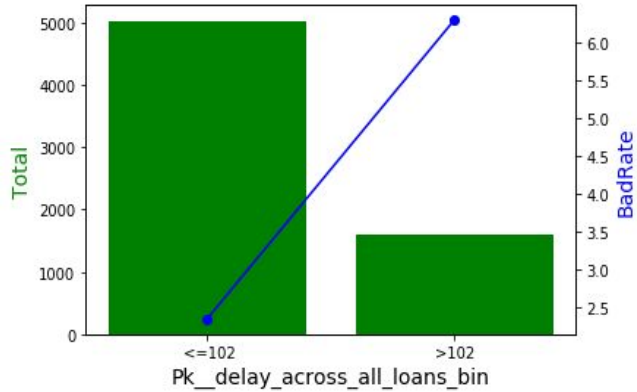
Which are -

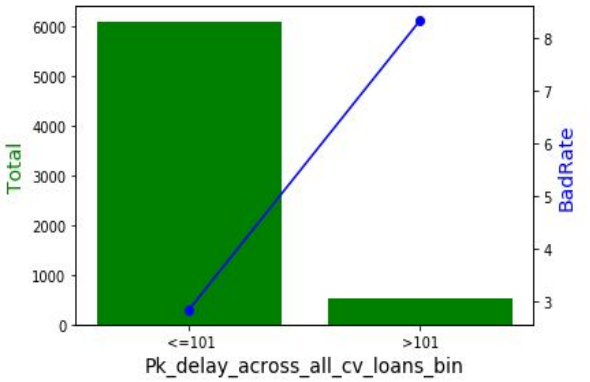
1. Location
2. Agr_value_bin
3. Pk__delay_across_all_loans_bin
4. Pk_delay_across_all_cv_loans_bin
5. Total_no_closed_loan_bin
6. Total_no_secured_loan_bin

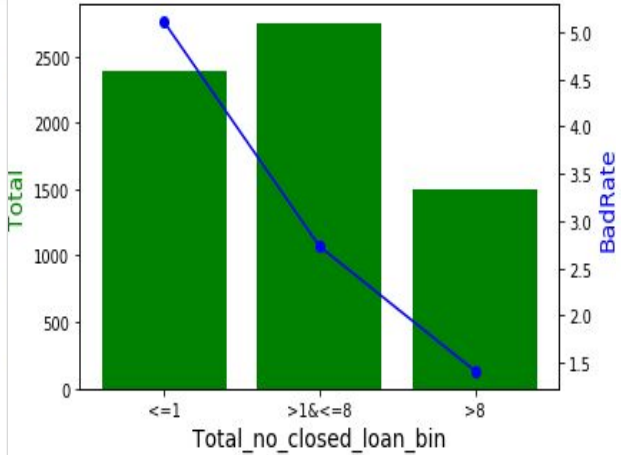
We subset the data by selecting only these 6 independent variables for the further steps.

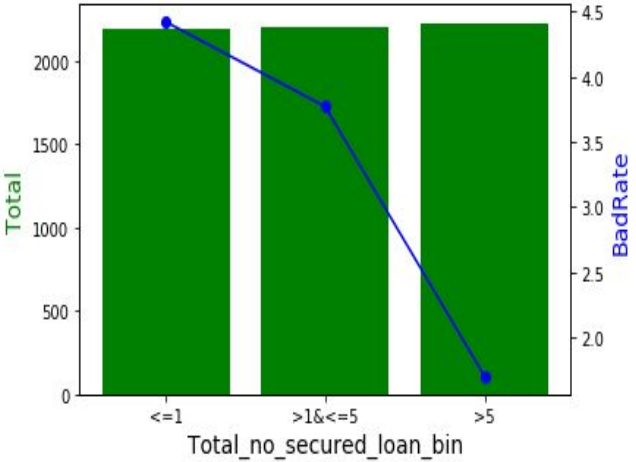
Plot for the selected variables











Step 3 - Building the Logistic Regression Model

```
import statsmodels.formula.api as smf
import statsmodels.api as sm
model=smf.logit(statement,data=model_data).fit()
print(model.summary2())
```

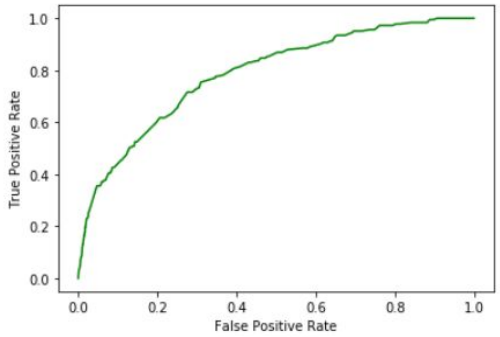
```
Optimization terminated successfully.
         Current function value: 0.125430
         Iterations 9
                                 Results: Logit
=================================================================================================
Model:                      Logit                    Pseudo R-squared:         0.133
Dependent Variable:         Flag                     AIC:                      1686.9531
Date:                       2021-08-11 15:40         BIC:                      1768.5436
No. Observations:           6629                     Log-Likelihood:           -831.48
Df Model:                   11                       LL-Null:                  -958.78
Df Residuals:               6617                     LLR p-value:              3.0193e-48
Converged:                  1.0000                   Scale:                    1.0000
No. Iterations:             9.0000
-------------------------------------------------------------------------------------------------
                                        Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------------------------------------
Intercept                              -6.5137   0.3455 -18.8544 0.0000 -7.1908 -5.8366
C(Location)[T.1]                        0.4916   0.2594   1.8949 0.0581 -0.0169  1.0001
C(Location)[T.2]                        0.5532   0.2327   2.3771 0.0175  0.0971  1.0093
C(Location)[T.3]                        1.1407   0.2099   5.4352 0.0000  0.7294  1.5521
C(Agr_value_bin)[T.1]                   0.1451   0.2108   0.6884 0.4912 -0.2680  0.5583
C(Agr_value_bin)[T.2]                   1.0348   0.2026   5.1085 0.0000  0.6378  1.4318
C(Pk__delay_across_all_loans_bin)[T.1]  1.5313   0.1889   8.1047 0.0000  1.1610  1.9017
C(Pk_delay_across_all_cv_loans_bin)[T.1] 0.5483  0.2238   2.4502 0.0143  0.1097  0.9869
C(Total_no_closed_loan_bin)[T.1]        0.5134   0.3015   1.7028 0.0886 -0.0775  1.1044
C(Total_no_closed_loan_bin)[T.2]        1.3887   0.3624   3.8322 0.0001  0.6785  2.0990
C(Total_no_secured_loan_bin)[T.1]       0.7744   0.2592   2.9875 0.0028  0.2664  1.2825
C(Total_no_secured_loan_bin)[T.2]       0.8699   0.3285   2.6483 0.0081  0.2261  1.5137
=================================================================================================
```

Training Report

In [111]:
```
from sklearn.metrics  import classification_report
print(classification_report(y_train , train.Predicted))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.98	5120
1	0.50	0.01	0.01	183
accuracy			0.97	5303
macro avg	0.73	0.50	0.50	5303
weighted avg	0.95	0.97	0.95	5303

In [112]:
```
from sklearn import metrics
y_pred_proba = model.predict(X_train)
fpr, tpr, _ = metrics.roc_curve(y_train,y_pred_proba)
#create ROC curve`
plt.plot(fpr,tpr,color="green")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

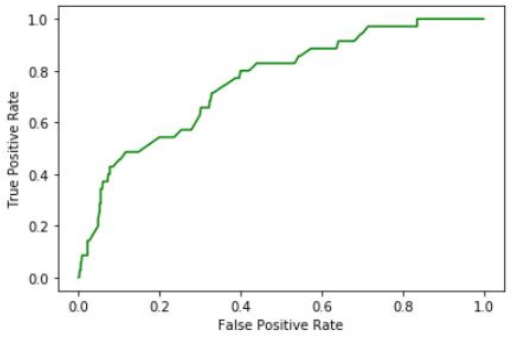


Testing Report

In [115]:
```
from sklearn.metrics  import classification_report
print(classification_report(y_test, test.Predicted))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.99	1291
1	0.00	0.00	0.00	35
accuracy			0.97	1326
macro avg	0.49	0.50	0.49	1326
weighted avg	0.95	0.97	0.96	1326

In [116]:
```
from sklearn import metrics
y_pred_proba = model.predict(X_test)
fpr, tpr, _ = metrics.roc_curve(y_test,y_pred_proba)
#create ROC curve
plt.plot(fpr,tpr,color="green")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



Kolmogorov Smirnov Stat(KS Stat)

- The Kolmogorov Smirnov stat also known as ks stat is the measure of the degree of separation between the positive and the negative distributions. So if the model is able to seperate the distribution into two seperate groups in which one group contains all positives and other has all negatives, the value for ks is 100. On the other hand when the separation is done randomly, the value for ks is 0.

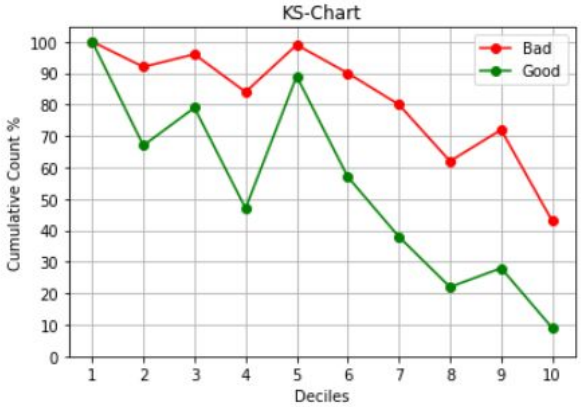
- Hence the value for ks ranges from 0-100.

- It calculates the vertical distance between the cumulative distribution functions of the two classifications, which in our case is Good and Bad.

- The maximum vertical distance between the functions is selected to be the threshold for making the decisions about classifying as good or bad.

Train data

In [121]:
```
1 ksStat(PD_train)
```

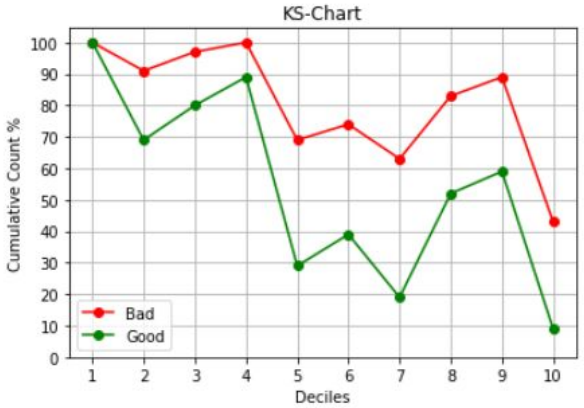


Out[121]:

decile	Bad_Count	Good_Count	max_prob	min_prob	Total_Cust	Bad_Rate	Population%	Cum_Bad%	Cum_Good%	KS_Stat	max_ks
10	78	445	0.500036	0.065709	523	15.00%	9.86%	43.00%	9.00%	34.00%	
8	35	689	0.048837	0.035802	724	5.00%	13.65%	62.00%	22.00%	40.00%	
9	18	314	0.065393	0.049204	332	5.00%	6.26%	72.00%	28.00%	44.00%	*****
7	15	522	0.035036	0.025097	537	3.00%	10.13%	80.00%	38.00%	42.00%	
4	8	449	0.013994	0.011057	457	2.00%	8.62%	84.00%	47.00%	37.00%	
6	10	488	0.024257	0.019443	498	2.00%	9.39%	90.00%	57.00%	33.00%	
2	4	529	0.006926	0.004619	533	1.00%	10.05%	92.00%	67.00%	25.00%	
3	7	589	0.010691	0.007205	596	1.00%	11.24%	96.00%	79.00%	17.00%	
5	7	543	0.019077	0.014589	550	1.00%	10.37%	99.00%	89.00%	10.00%	
1	1	552	0.004290	0.001481	553	0.00%	10.43%	100.00%	100.00%	0.00%	

Test Data

In [122]:
```
1 ksStat(PD_test)
```



Out[122]:

decile	Bad_Count	Good_Count	max_prob	min_prob	Total_Cust	Bad_Rate	Population%	Cum_Bad%	Cum_Good%	KS_Stat	max_ks
10	15	110	0.500036	0.065393	125	12.00%	9.43%	43.00%	9.00%	34.00%	
7	7	140	0.038412	0.026128	147	5.00%	11.09%	63.00%	19.00%	44.00%	*****
5	2	129	0.019077	0.014905	131	2.00%	9.88%	69.00%	29.00%	40.00%	
6	2	128	0.025634	0.019443	130	2.00%	9.80%	74.00%	39.00%	35.00%	
8	3	169	0.048837	0.038799	172	2.00%	12.97%	83.00%	52.00%	31.00%	
9	2	81	0.064945	0.049508	83	2.00%	6.26%	89.00%	59.00%	30.00%	
2	1	129	0.006926	0.004663	130	1.00%	9.80%	91.00%	69.00%	22.00%	
3	2	146	0.010691	0.007205	148	1.00%	11.16%	97.00%	80.00%	17.00%	
4	1	122	0.014696	0.011057	123	1.00%	9.28%	100.00%	89.00%	11.00%	
1	0	137	0.004619	0.001481	137	0.00%	10.33%	100.00%	100.00%	0.00%	

Population stability index (PSI)

- The population stability index (PSI) is a widely used statistic that measures how much a variable has shifted over time.
- A high PSI may alert the business to a change in the characteristics of a population. This shift may require investigation and possibly a model update.
- PSI is commonly used among banks to measure the shift between model development data and current data.
- In practice, the following “rule of thumb” is used:
    - PSI < 0.1: No change. You can continue using the existing model.
    - PSI >=0.1: but less than 0.2 — Slight change is required.
    - PSI >=0.2: Significant change is required. Ideally, we should not use this model anymore. It should be recalibrated/redeveloped.

Step 5 - Population Stability Index (PSI)

```
PSI value = 0.002392
```

Out[123]:

decile	DevPop	CurrPop	DevPop%(Actual)	CurrPop%(Expected)	Actual-Expected	ln(Actual/Expected)	Index
10	523	125	9.86%	9.43%	0.44%	4.52%	0.000197
9	332	83	6.26%	6.26%	0.00%	0.02%	0.000000
8	724	172	13.65%	12.97%	0.68%	5.12%	0.000349
7	537	147	10.13%	11.09%	-0.96%	-9.05%	0.000869
6	498	130	9.39%	9.80%	-0.41%	-4.30%	0.000178
5	550	131	10.37%	9.88%	0.49%	4.86%	0.000239
4	457	123	8.62%	9.28%	-0.66%	-7.36%	0.000485
3	596	148	11.24%	11.16%	0.08%	0.69%	0.000005
2	533	130	10.05%	9.80%	0.25%	2.49%	0.000061
1	553	137	10.43%	10.33%	0.10%	0.93%	0.000009

Conclusion

From the above slides, we can summarize and conclude the following things :

- Out of the total 48 variables, we first managed to select only 13 variables based on the IV values.
- These 13 variables were further drilled down to select only 6 variables for model making process.
- A logistic regression model is built on the data select data now.
- We got a ks value of around 44% which means that this will be acting as the threshold for any decision making done for the classification.
- For the created model , the psi value came out to be 0.002392. This value lies much below than the threshold value (0.1).
- Hence no changes to the model are to be made.

Thank You!