

BFSI – Capstone Project Roadmap

Sunil Sarthak Rout

- Problem Statement
- Data loading/cleaning/massaging
- EDA
- Model Building
- Model Evaluating
- Model Choosing
- Application Scorecard
- Financial benefit assessment

Index

Background

- 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. The CEO believes that the best strategy to mitigate credit risk is to ‘acquire the right customers’

Strategy

- Identify the right customers for CredX 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 the project

Problem Statement

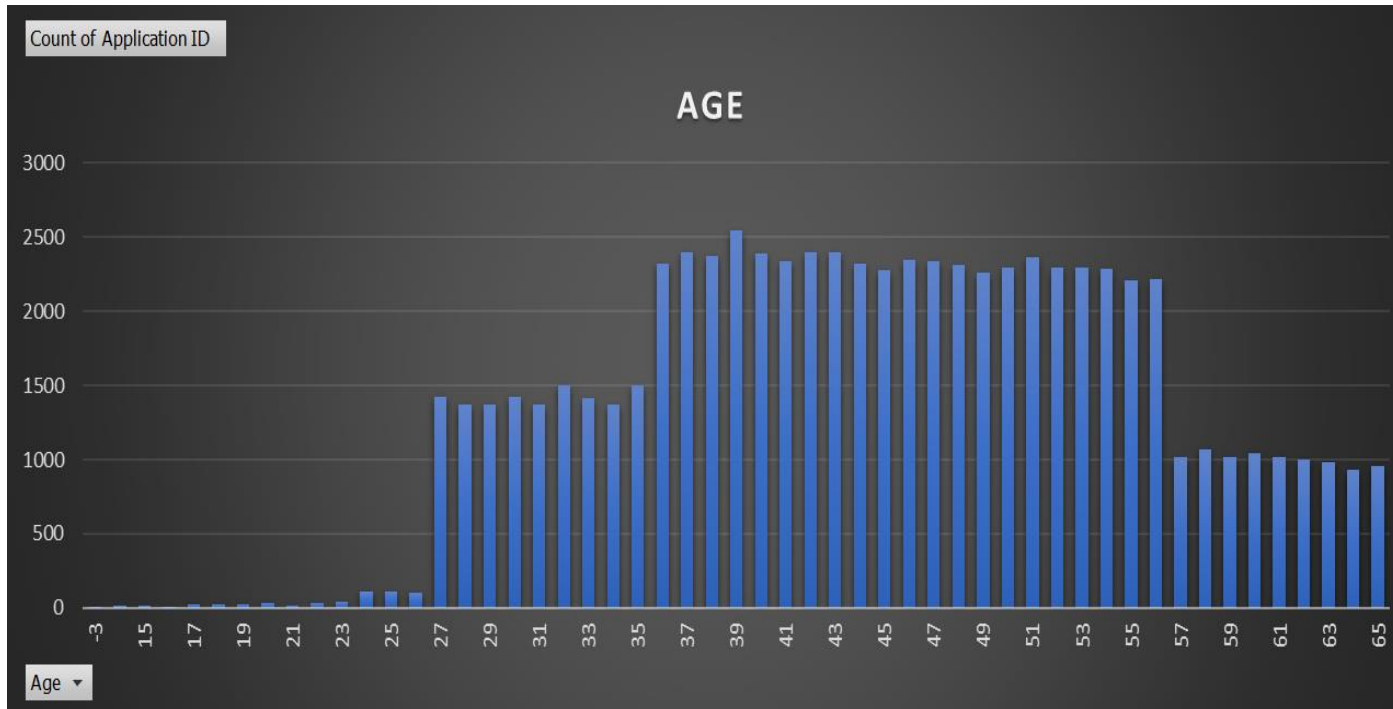
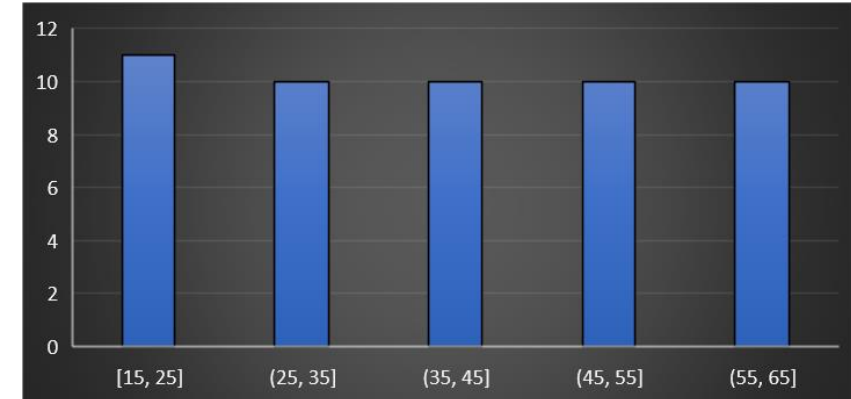
- Data is available in 2 parts - **Demographic/application data & Credit bureau**
- Both Datasets are equally important for the project and will be used to prepare the initial and final models
- Data was loaded in the traditional way to apply EDA (univariate and multivariate)
- Both Datasets were combined to perform

Data Loading/Cleaning/ Massaging

EDA – Application ID & Age

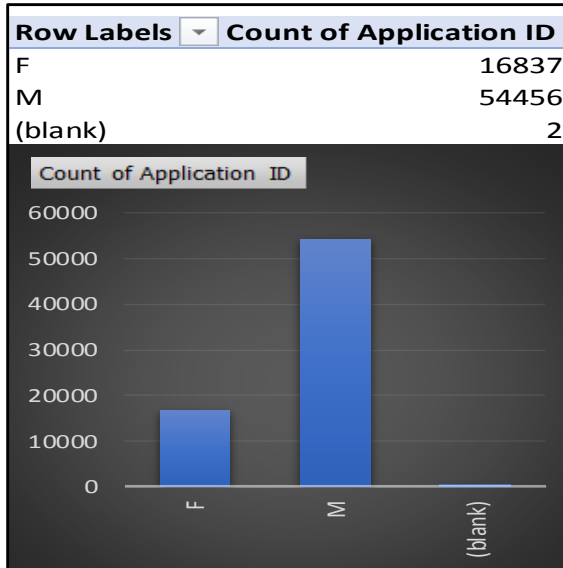
Row Labels	Count of Application
653287861	2
671989187	2
765011468	2

Found 3 Duplicates in Application ID



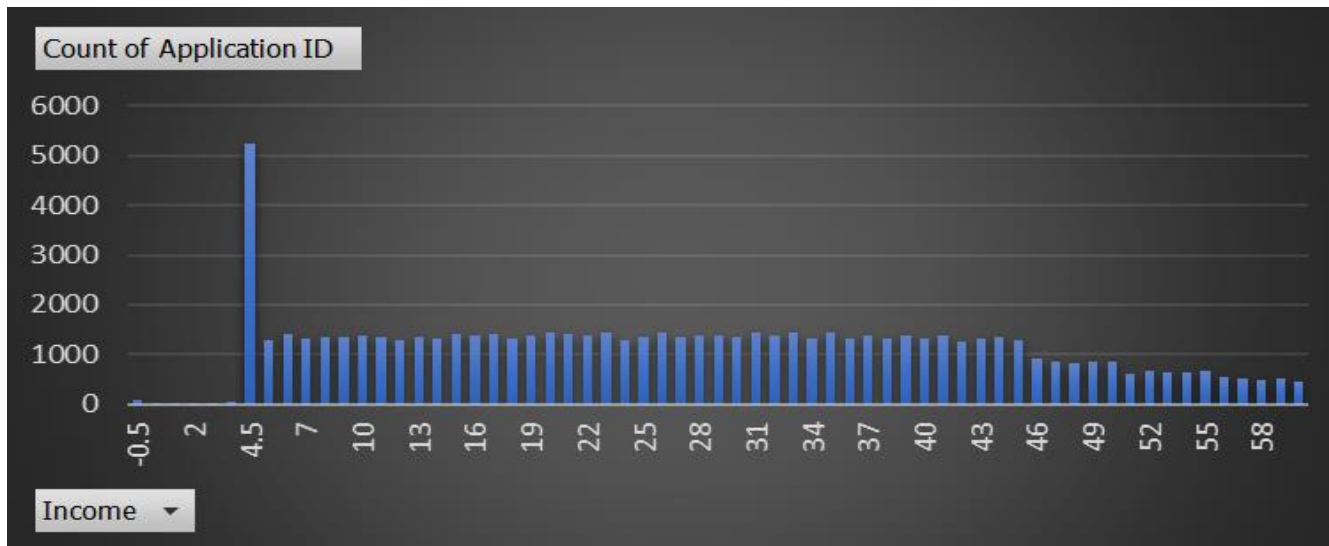
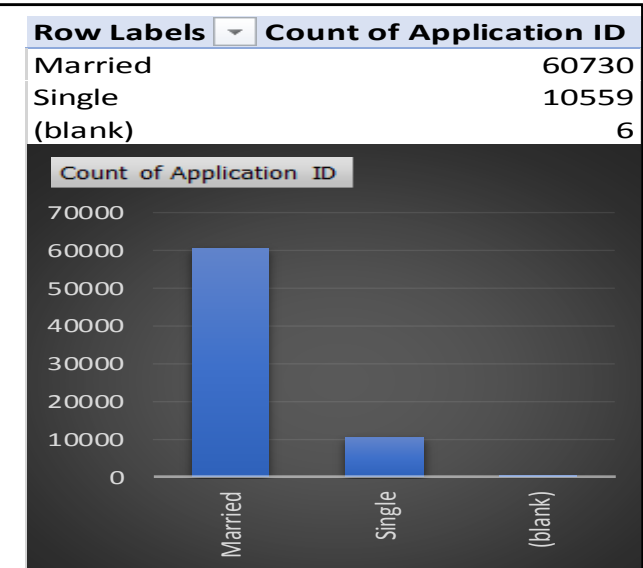
- Outliers in AGE variable as -3 & 0
- Maximum Applicants are in range of 27-56 which is 87% of total number of applicants
- On further binning Group 15-25 appears to maximum applicants. It might be due to young people with modern needs but limited income

EDA – Gender, Marital Status and Income



- 2 <blanks> in the gender variable
- Should be deleted to clean the data
- 76% of applicants are males

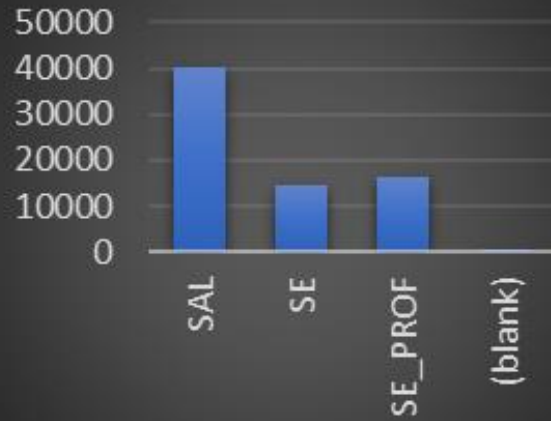
- 6 <blanks> in the Marital Status variable
- Should be deleted to clean the data
- 85% of applicants are married



- Income variable with outlier as -0.5 value
- 4.5 value has the maximum count of application ids, may be due to the minimum wage rule

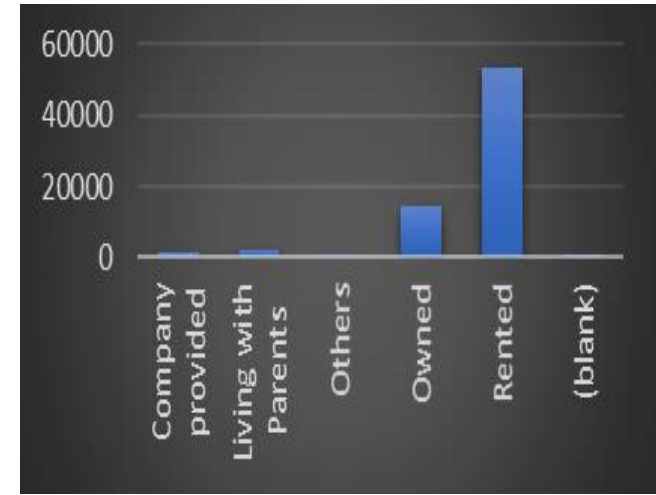
EDA – Profession, Residence & Education

Count of Application ID

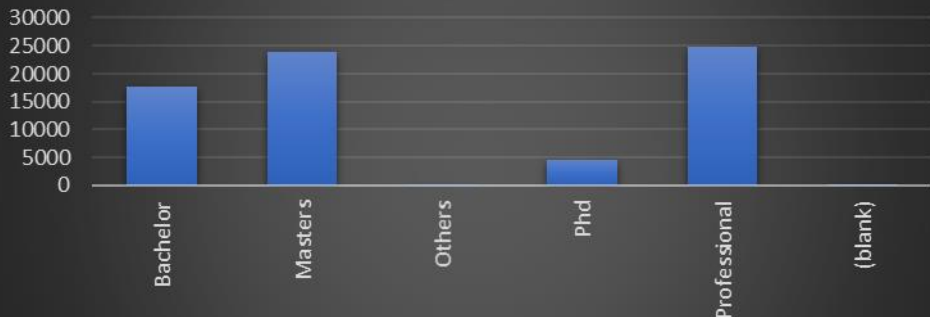


- 14 <blank> outliers
- Should be deleted

- Maximum people living in rented house
- Few outliers as <blank>, should be converted as “Others”

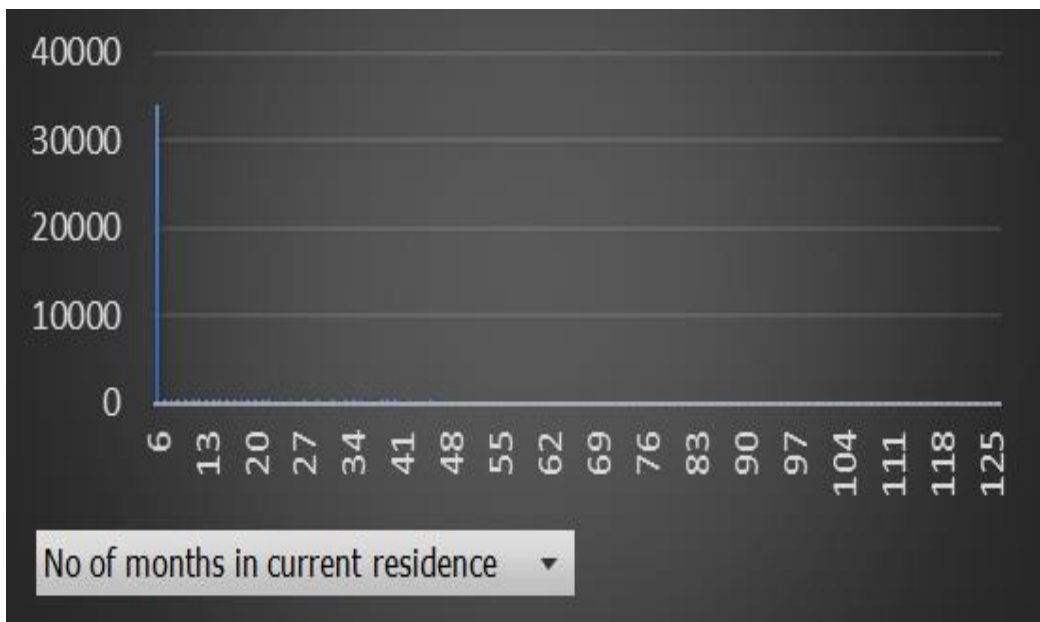


Count of Application ID

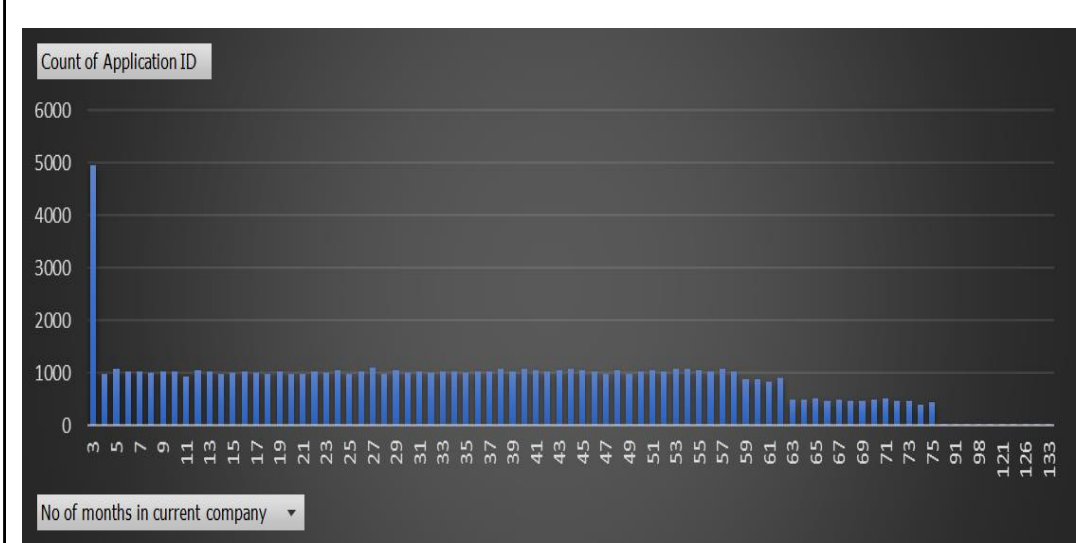


- Education has 119 <Blank> as outliers
- We can put them in “Others” to clean it

EDA – Num of months (residence and company)



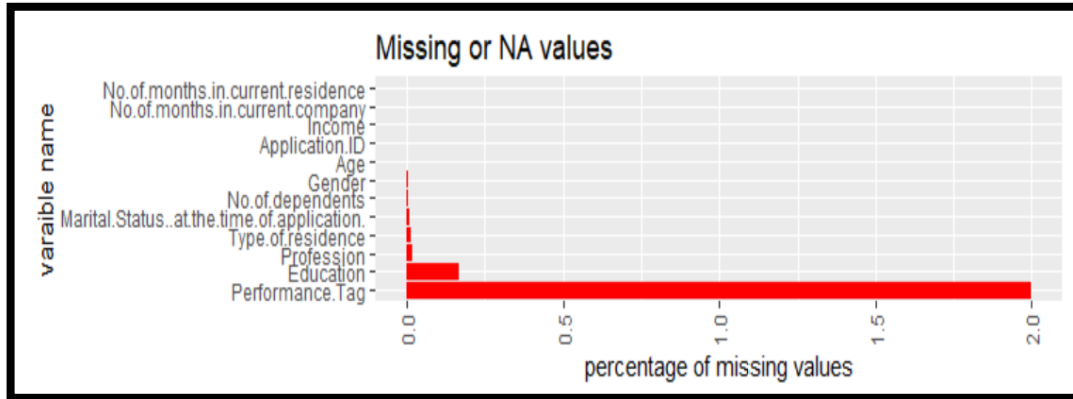
- Approx 50% of people are in current residence for less than 6 months



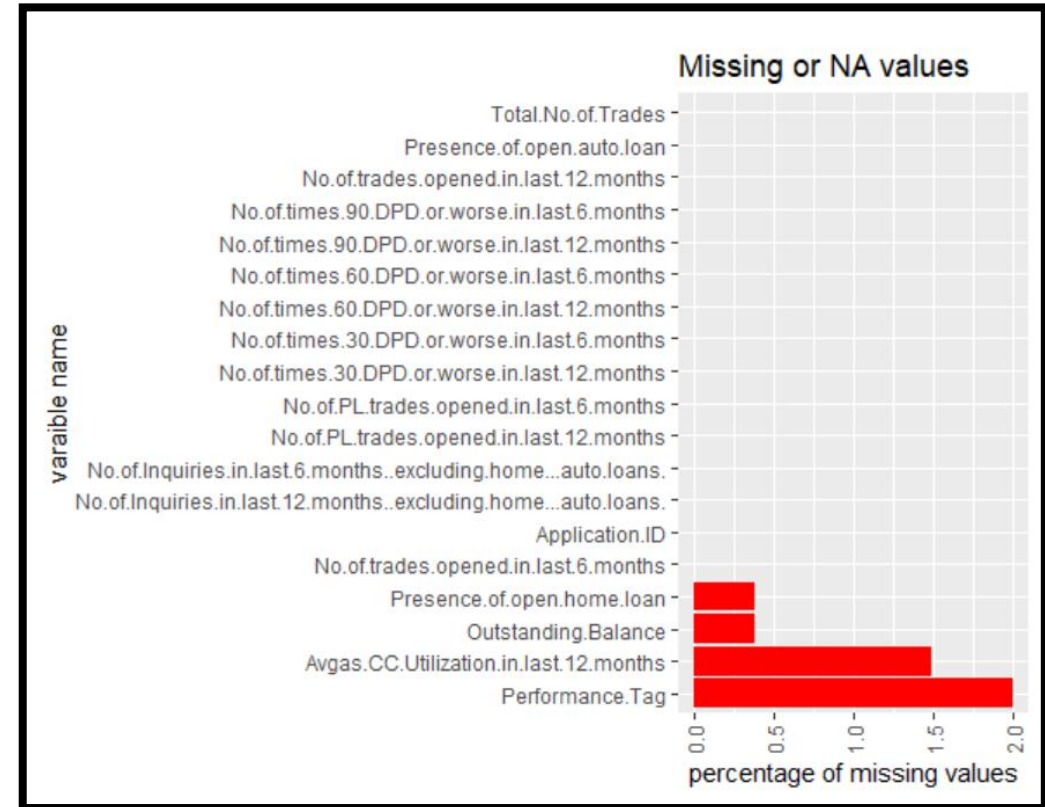
- Maximum applicants are less than 3 months old in their company

EDA – Performance Tag

Application.ID	Age
0	0
Gender	Marital.Status..at.the.time.of.application.
2	6
No.of.dependents	Income
3	0
Education	Profession
119	14
Type.of.residence	No.of.months.in.current.residence
8	0
No.of.months.in.current.company	Performance.Tag
0	1425



- Approximately 2% of values in 'Performance.Tag' column is missing and can be removed
- Similarly, the other variables which too have trivial percentage of missing values/blanks can be removed



- These NA records will be used against the model validation to test the accuracy as these NA's signifies the applicants who got rejected in the first place

EDA – Other Variables

No. of
dependents

- 3 is the majority category present

DPD

- Defaulter's are a mere 4.2% of the entire population after the NA's removal for performance column

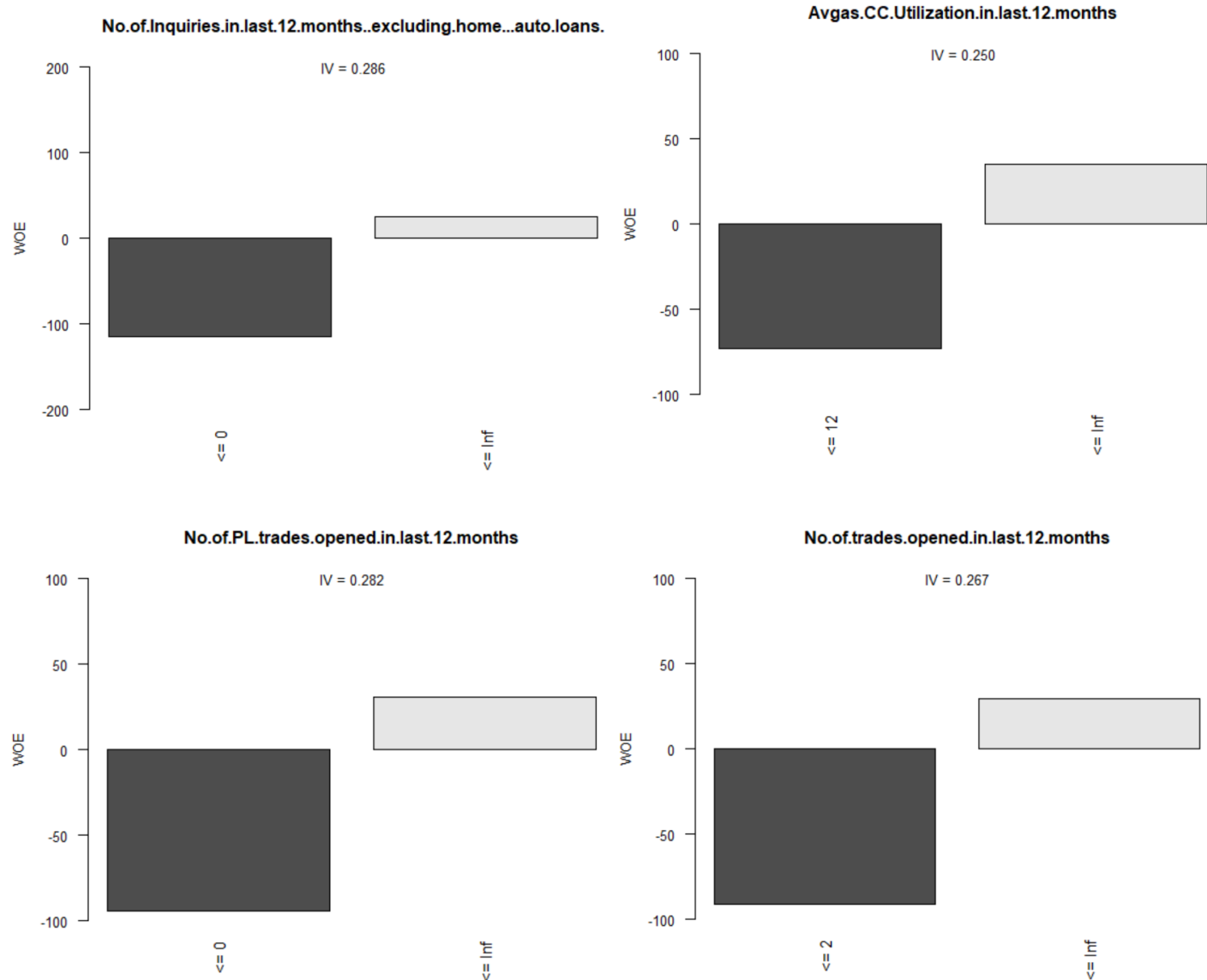
No
Defaulters

- More than 60% of Applicants never defaulted
- This is good and a bad thing for mode building. Lesser the default case lesser the data train upon

WOE - Analysis

Using the “woeBinning” package which optimizes the number of bins for the IV (Information value)

This generates a supervised fine and coarse classing of numeric variables and factors with respect to a dichotomous target variable

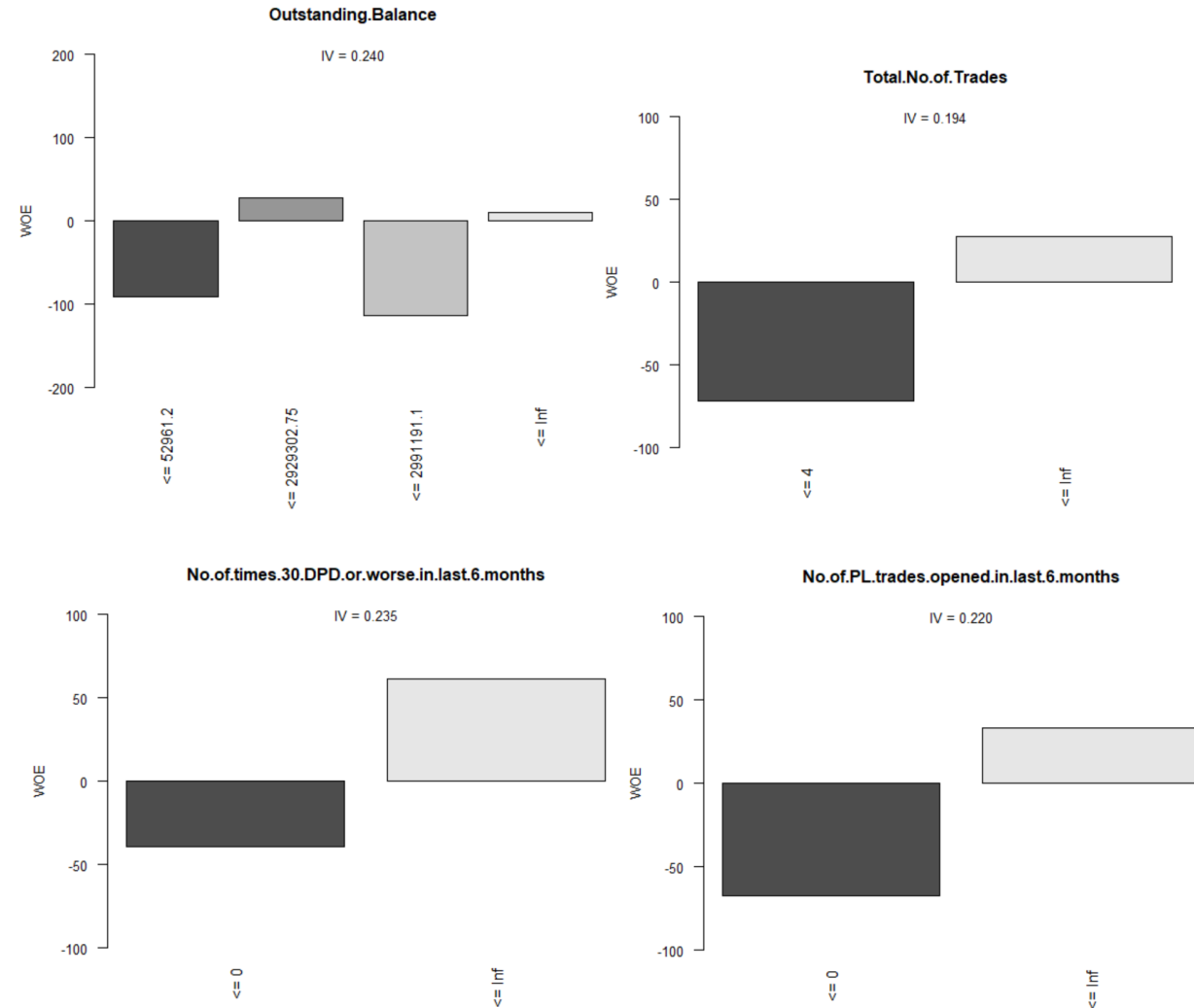


WOE – Analysis

Using the “woeBinning” package which optimizes the number of bins for the IV (Information value)

This generates a supervised fine and coarse classing of numeric variables and factors with respect to a dichotomous target variable

... contd

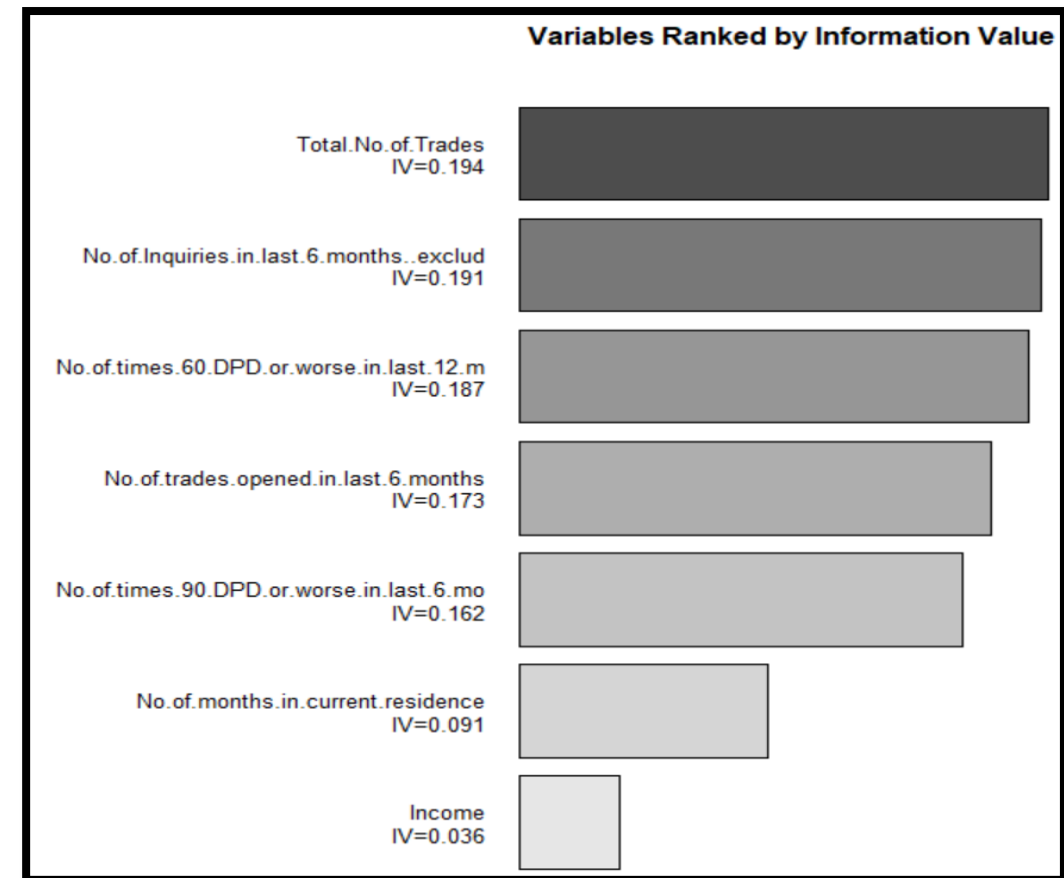
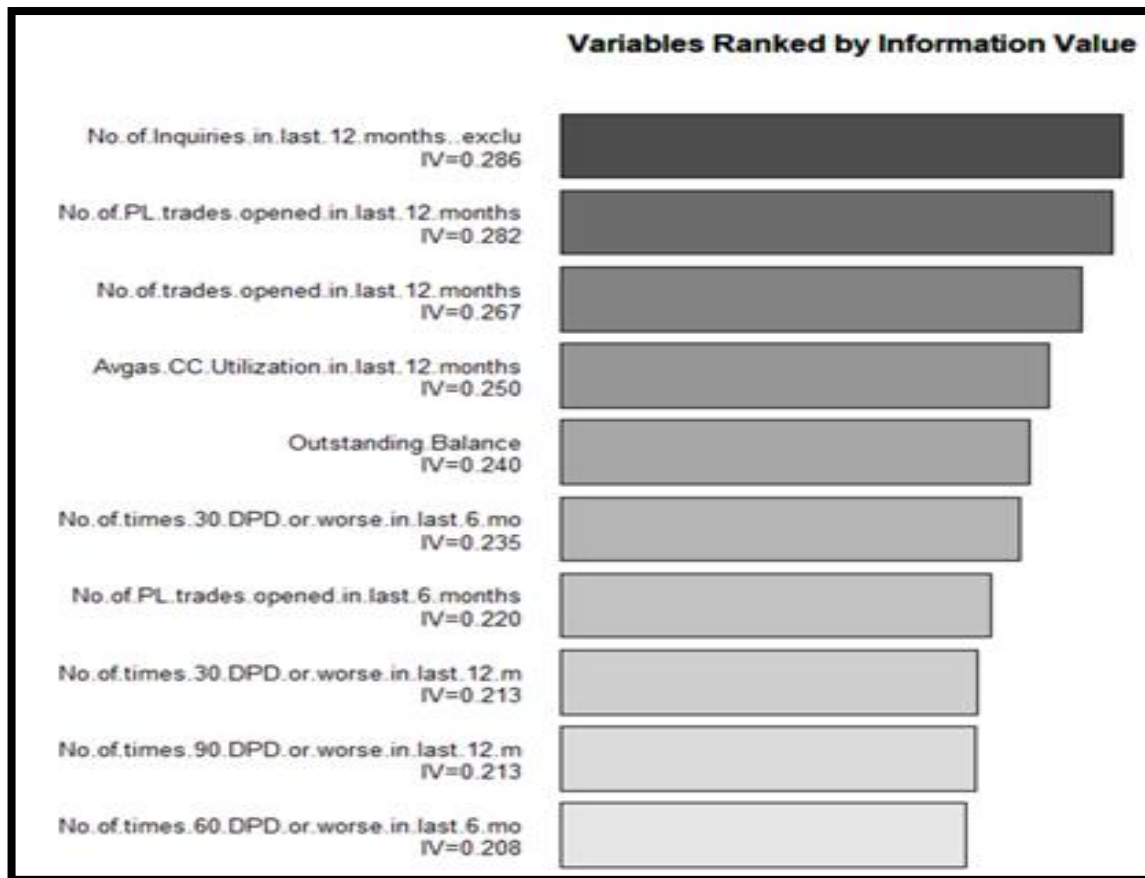


Plotting of Variables as per their IV value order

The threshold for the significant variable is considered here as >0.1 as per the convention of IV statistic in credit scoring


If IV statistic is between 0.1 to 0.3, then the predictor has a medium strength relationship to the Goods/Bads odds ratio

Hence, ALL the variables having value greater than 0.1 as the IV value are significant here



Observations

Replacing the actual values of all the variables by the corresponding WOE values and storing the data in a separate dataset



Our dataset is returned with two extra columns -
`Variable_name.binned`, `woe.Variable_name.binned`



Numeric variables (continuous and ordinal) are binned by merging initial classes with similar frequencies



Factors (categorical variables) are binned by merging factor levels

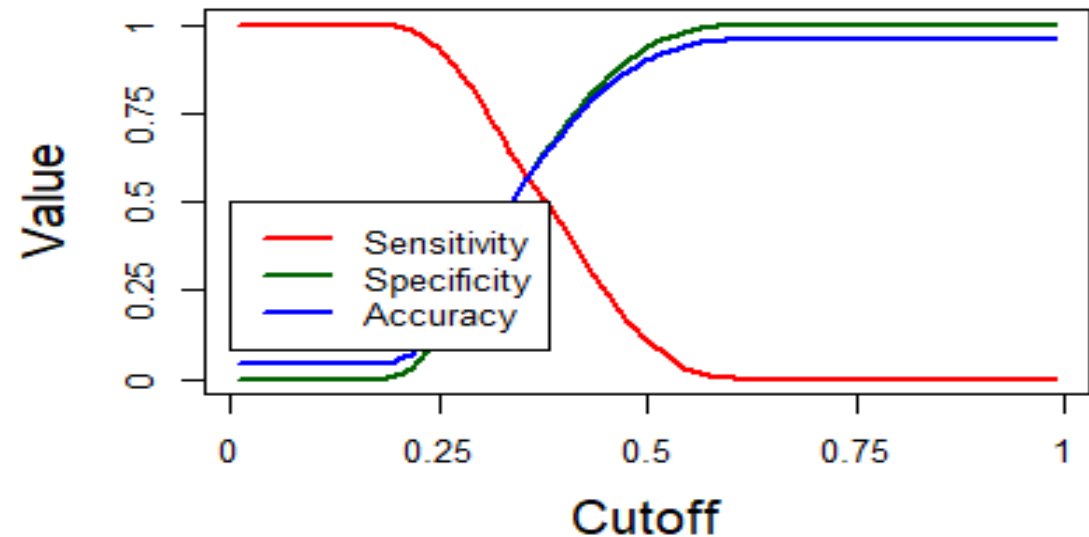
- 6 models were built in whole project
- Algorithms used were Logistic Regression and Random forest
- Each model was built, tested and validated with data provided to finally come-up with most optimal model

Model Building

Model Building – 1. Demographic Data Model

- Log Model with 32% of logistic regression model with all the variables with the dataset having woe replaced values
- Accuracy – 42%, Specificity – 41.2%, Sensitivity – 69.4%
- **Verdict – Not an Optimal Model**

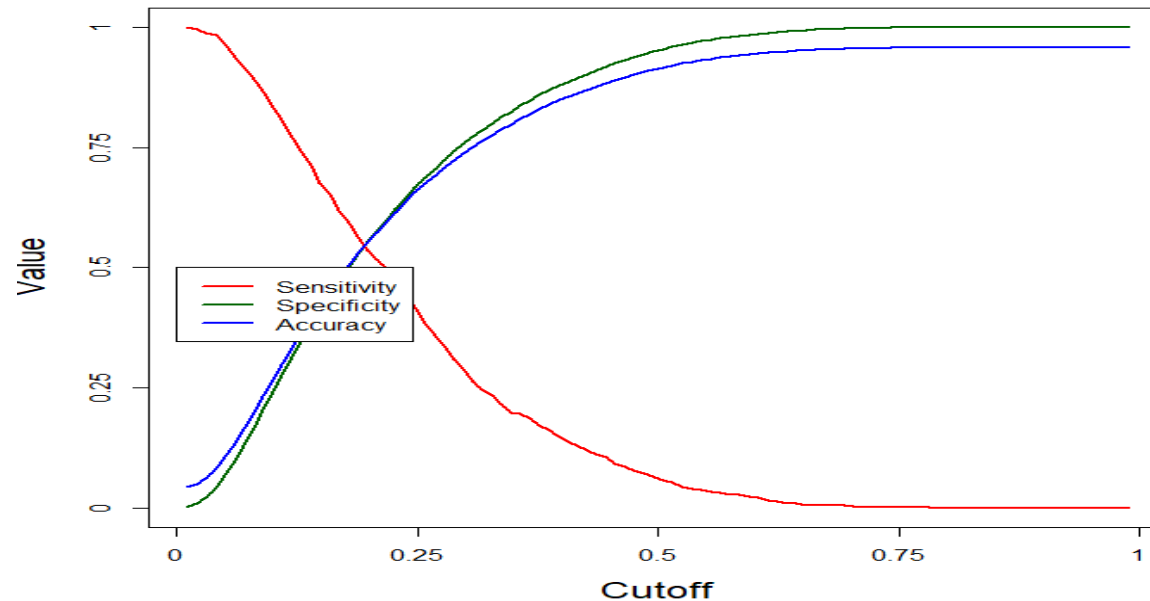
```
#Removing insignificant 'woe.Marital.Status..at.the.time.of.application..binned' variable as per p-value  
train_Demographic_woe_SMOTE_5<-glm(formula = Performance.Tag ~ woe.No.of.months.in.current.residence.binned +  
    woe.Income.binned + woe.No.of.months.in.current.company.binned +  
    woe.Age.binned + woe.No.of.dependents.binned + woe.Profession.binned,  
    family = "binomial", data = train_Demographic_woe_SMOTE)
```



Model Building – 2. Demographic Data Model using Random Forest

- Random Forest Algorithm
- with 18.8% of probability as cutoff
- Accuracy – 52%
- Specificity – 52.2%
- Sensitivity – 57.0%
- **Verdict – Not an Optimal Model**

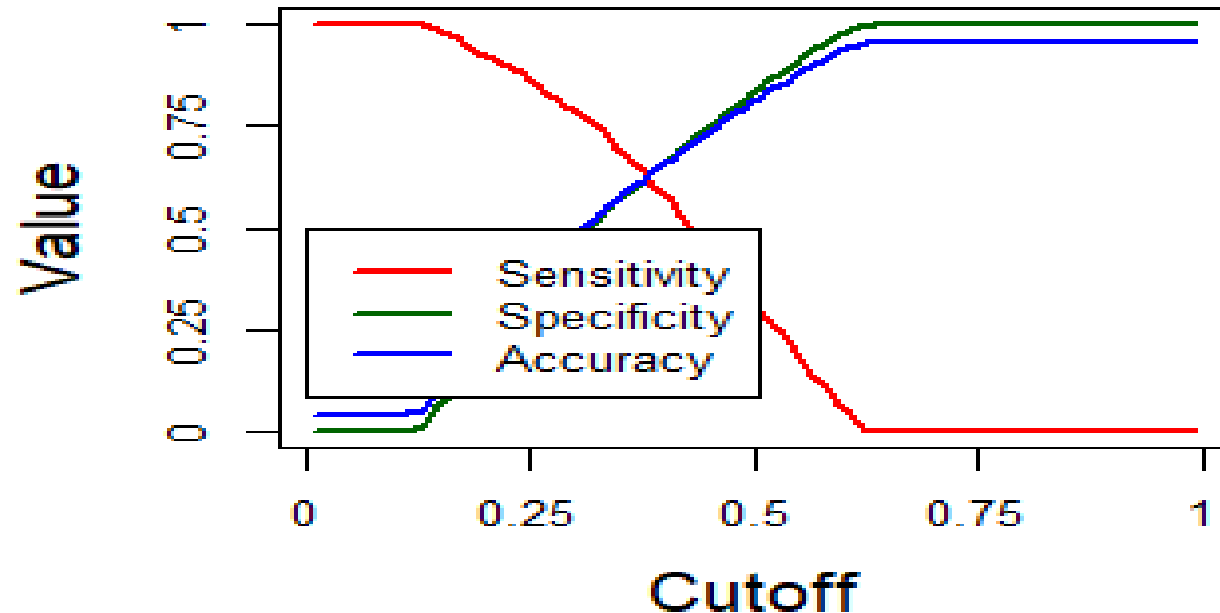
```
##MODEL 2:  
## Random Forest with dummy variables data-----  
train_model_dummy_smote_RF <- randomForest(Performance.Tag ~.,  
                                             data = train_model_dummy_smote,  
                                             proximity = F, do.trace = T, mtry = 5)
```



Model Building – 3. Both Data Sets

- Logistic Regression Algorithm
- with 38% of probability as cutoff
- Accuracy – 62.3%
- Specificity – 62.3%
- Sensitivity – 62.9%
- **Verdict – Not an Optimal Model**

```
train_Merged_woe_SMOTE_9<-glm(formula = Performance.Tag ~ woe.Avgas.CC.Utilization.in.last.12.months.binned +  
  woe.No.of.times.30.DPD.or.worse.in.last.12.months.binned +  
  woe.No.of.Inquiries.in.last.6.months..excluding.home...auto.loans..binned +  
  woe.No.of.trades.opened.in.last.6.months.binned +  
  woe.Income.binned +  
  woe.Age.binned +  
  woe.No.of.dependents.binned + woe.Presence.of.open.auto.loan.binned +  
  woe.Type.of.residence.binned + woe.Marital.Status..at.the.time.of.application..binned,  
  family = "binomial", data = train_Merged_woe_SMOTE)
```

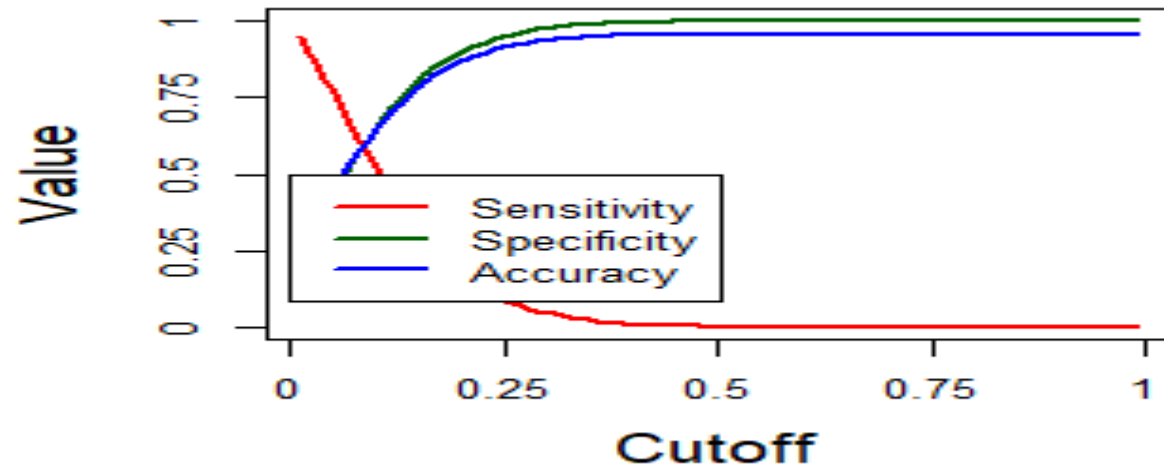


Model Building – 4. Both Data Sets

- Random Forest Algorithm
- with 9% of probability as cutoff
- Accuracy – 60.1%
- Specificity – 56.2%
- Sensitivity – 61.0%
- **Verdict – Not an Optimal Model**

##MODEL 4:

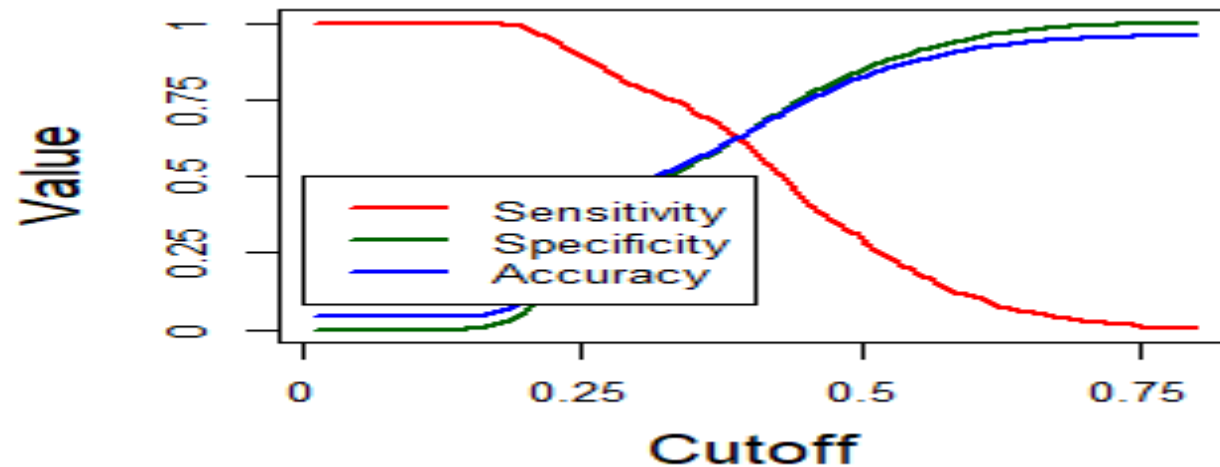
```
## Random Forest with woe variables data-----  
train_Merged_woe_SMOTE_RF <- randomForest(Performance.Tag ~., data = train_Merged_woe_SMOTE,  
                                           proximity = F, do.trace = T, mtry = 5)
```



Model Building – 5. Both Data Sets

- Logistic Regression Algorithm with Dummy variables
- With 39% of probability as cutoff
- Accuracy – 62.9%
- Specificity – 62.4%
- Sensitivity – 62.9%
- **Verdict – Not an Optimal Model**

```
#Removing insignificant 'No.of.trades.opened.in.last.6.months' variable as per high p-value & VIF
train_Merged_dummy_SMOTE_14<-glm(formula = Performance.Tag ~ Marital.Status..at.the.time.of.application. +
Income + No.of.months.in.current.residence + No.of.months.in.current.company +
No.of.times.90.DPD.or.worse.in.last.12.months + No.of.times.30.DPD.or.worse.in.last.12.months +
Avgas.CC.Utilization.in.last.12.months +
No.of.PL.trades.opened.in.last.12.months +
Presence.of.open.auto.loan,
family = "binomial",
data = train_merged_model_dummy_smote)
```



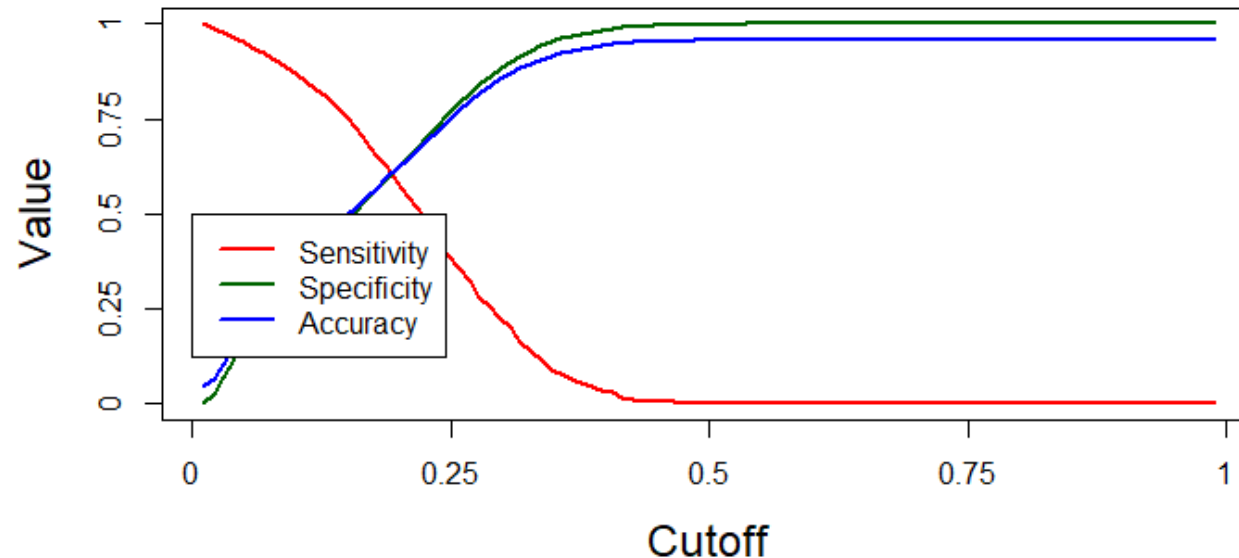
Model Building – 6. Both Data Sets

- Random Forest Algorithm with Dummy variables
- With 20% of probability as cutoff
- Accuracy – 62.1%
- Specificity – 62.3%
- Sensitivity – 58.2%
- **Verdict – Optimal Model**

##MODEL 6:

Random Forest with dummy variables on MERGED data-----

```
train_merged_model_dummy_smote_RF <- randomForest(Performance.Tag ~., data = train_merged_model_dummy_smote,  
proximity = F, do.trace = T, mtry = 5)
```



Observations – Why we chose 6th model as Optimal/best

Specificity is the metric we chose for optimal model

- Actual NO's correctly predicted
- In line with the Accuracy of the model

Sensitivity/TPR could be on the lower end

- Prediction of Actual Yes's is of lesser importance here

Application Scorecard

Comparing the Approved Application data with the Rejected Application data it is observed that Applicants with a Score value above 298 would be less likely to default.

So, below 298 score credit card would not be granted to the applicants.

```
#Define a target:
#Target Score Value (ts): 400
ts<-400
#Inverted Target Odds (to): 10
to<-10

#Define slope:
#points to double the odds (pdo): 20
pdo<-20

Factor = pdo / log(2)
#28.8539

#Offset<- (ts)-(Factor*log(to))
fac<-Factor*log(to)
fac
#66.43856
Offset<-400-66.43856
#333.5614
#Score = Offset + Factor * ln (odds)
print (" The Equation is: Score<-333.5614+(28.8539*validation_Rejected_df$log_odds)")
```

Financial Benefit Assessment

- As per the confusion matrix of Model6 we are able to detect maximum number of good customers



Let's assume on average one customer uses 1000\$ of credit and we gain 10% from each customer



On the basis of our confusion matrix –

A. If we wrongly identify a good customer as bad we lose 100\$

B. If we identify bad customer as good then we lose 1000\$ whole



In case we don't have any model at all then we lose 1000\$ from each person defaulted

Financial Benefit Assessment .. contd

- As evident from the calculation we just profited around 10% of total loss
- This can be further minimized by either introducing more variables or further tuning of model

		Reference	
		N	Y
Predicted	N	19929	800
	Y	25	2

				Loss in \$	Number of Customers wrongly predicted	Total Loss made	Actual customers defaulted	Actual Loss Made	Total Profit
Good customers lost		fp rate	0.1%	\$ (10)	\$ 25	\$ (250)			
Bad customers gained		fn r	99.8%	\$ (100)	\$ 800	\$ (80,000)	900	\$ (90,000)	
Total						\$ (80,250)		\$ (90,000)	\$ 9,750

Appendix