SAFE DRIVER PREDICTION

Ayyappa Bhuma

Contents

Problem Statement and Background:

The challenge is to build a model that predicts the probability that a driver will initiate an auto insurance claim.

Port Seguro is one of Brazil's largest auto and homeowner insurance companies. They feel good drivers deserve to pay less money as insurance compared to bad drivers. So, based on the data collected by them from their clients we have to classify their clients as good or bad drivers.

Inaccuracies in car insurance company's claim predictions raise the cost of insurance for good drivers and reduce the price for bad ones.

Understanding the Data:

We are provided with two datasets, first one is train dataset comprising of 595212 observations of 59 variables and the second one is test dataset comprising of 892816 observations of 58 variables. The train dataset is for training our classification model using which we need to predict the test dataset.

Both the data sets are similar, except there is no target variable in the test dataset. Both the datasets are not clean and are to be cleaned before building the model. In order to make the cleaning easier let us join both the datasets, do pre-processing on the combined dataset and then divide back the combined dataset before building the model.

There are 58 independent variables in the dataset and 1 target variable. The variable names include the class of the variable in their suffix though the given dataset has variables with their class inappropriate to their actual class. So, we need to convert the class of the variables to their appropriate classes as indicated in their names.

Now let us have a look at the dataset and structure of the dataset

```
> train1[1:30, 1:10]
   id target ps_ind_01 ps_ind_02_cat ps_ind_03 ps_ind_04_cat ps_ind_05_cat ps_ind_06_bin ps_ind_07_bin ps_ind_08_bin
1   7      0         2              2         5              1              0              0              1              0
2   9      0         1              1         7              0              0              0              0              1
3  13      0         5              4         9              1              0              0              0              1
4  16      0         0              1         2              0              0              1              0              0
5  17      0         0              2         0              1              0              1              0              0
6  19      0         5              1         4              0              0              0              0              0
7  20      0         2              1         3              1              0              0              1              0
8  22      0         5              1         4              0              0              1              0              0
9  26      0         5              1         3              1              0              0              0              1
10 28      1         1              1         2              0              0              0              1              0
11 34      0         5              2         2              0              0              0              0              1
12 35      0         2              1         3              1              0              0              1              0
13 36      0         2              1         1              1              0              0              0              1
14 43      0         1              1         3              1              0              0              1              0
15 46      0         5              1        11              0              0              0              0              0
16 48      0         5              1         3              1              0              0              1              0
17 50      0         1              2         1              0              0              0              0              1
18 58      0         5              1         6              0              1              1              0              0
19 61      0         5              1         8              0              0              1              0              0
20 64      1         0              1         2              1              0              1              0              0
```

```
> str(data)
'data.frame':	595212 obs. of  59 variables:
 $ id            : int  7 9 13 16 17 19 20 22 26 28 ...
 $ target        : int  0 0 0 0 0 0 0 0 0 1 ...
 $ ps_ind_01     : int  2 1 5 0 0 5 2 5 5 1 ...
 $ ps_ind_02_cat : int  2 1 4 1 2 1 1 1 1 1 ...
 $ ps_ind_03     : int  5 7 9 2 0 4 3 4 3 2 ...
 $ ps_ind_04_cat : int  1 0 1 0 1 0 1 0 1 0 ...
 $ ps_ind_05_cat : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_06_bin : int  0 0 0 1 1 0 0 1 0 0 ...
 $ ps_ind_07_bin : int  1 0 0 0 0 0 1 0 0 1 ...
 $ ps_ind_08_bin : int  0 1 1 0 0 0 0 0 1 0 ...
 $ ps_ind_09_bin : int  0 0 0 0 0 1 0 0 0 0 ...
 $ ps_ind_10_bin : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_11_bin : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_12_bin : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_13_bin : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_14     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_15     : int  11 3 12 8 9 6 8 13 6 4 ...
 $ ps_ind_16_bin : int  0 0 1 1 1 1 1 1 1 0 ...
 $ ps_ind_17_bin : int  1 0 0 0 0 0 0 0 0 0 ...
 $ ps_ind_18_bin : int  0 1 0 0 0 0 0 0 0 1 ...
 $ ps_reg_01     : num  0.7 0.8 0 0.9 0.7 0.9 0.6 0.7 0.9 0.9 ...
 $ ps_reg_02     : num  0.2 0.4 0 0.2 0.6 1.8 0.1 0.4 0.7 1.4 ...
 $ ps_reg_03     : num  0.718 0.766 NA 0.581 0.841 ...
 $ ps_car_01_cat : int  10 11 7 7 11 10 6 11 10 11 ...
 $ ps_car_02_cat : int  1 1 1 1 1 0 1 1 1 0 ...
 $ ps_car_03_cat : int  NA NA NA 0 NA NA NA 0 NA 0 ...
 $ ps_car_04_cat : int  0 0 0 0 0 0 0 0 0 1 ...
 $ ps_car_05_cat : int  1 NA NA 1 NA 0 1 0 1 0 ...
 $ ps_car_06_cat : int  4 11 14 11 14 14 11 11 14 14 ...
 $ ps_car_07_cat : int  1 1 1 1 1 1 1 1 1 1 ...
 $ ps_car_08_cat : int  0 1 1 1 1 1 1 1 1 1 ...
 $ ps_car_09_cat : int  0 2 2 3 2 0 0 2 0 2 ...
 $ ps_car_10_cat : int  1 1 1 1 1 1 1 1 1 1 ...
 $ ps_car_11_cat : int  12 19 60 104 82 104 99 30 68 104 ...
 $ ps_car_11     : int  2 3 1 1 3 2 2 3 3 2 ...
 $ ps_car_12     : num  0.4 0.316 0.316 0.374 0.316 ...
 $ ps_car_13     : num  0.884 0.619 0.642 0.543 0.566 ...
 $ ps_car_14     : num  0.371 0.389 0.347 0.295 0.365 ...
 $ ps_car_15     : num  3.61 2.45 3.32 2 2 ...
 $ ps_calc_01    : num  0.6 0.3 0.5 0.6 0.4 0.7 0.2 0.1 0.9 0.7 ...
 $ ps_calc_02    : num  0.5 0.1 0.7 0.9 0.6 0.8 0.6 0.5 0.8 0.8 ...
 $ ps_calc_03    : num  0.2 0.3 0.1 0.1 0 0.4 0.5 0.1 0.6 0.8 ...
 $ ps_calc_04    : int  3 2 2 2 2 3 2 1 3 2 ...
 $ ps_calc_05    : int  1 1 2 4 2 1 2 2 1 2 ...
 $ ps_calc_06    : int  10 9 9 7 6 8 8 7 7 8 ...
 $ ps_calc_07    : int  1 5 1 1 3 2 1 1 3 2 ...
 $ ps_calc_08    : int  10 8 8 8 10 11 8 6 9 9 ...
 $ ps_calc_09    : int  1 1 2 4 2 3 3 1 4 1 ...
 $ ps_calc_10    : int  5 7 7 2 12 8 10 13 11 11 ...
 $ ps_calc_11    : int  9 3 4 2 3 4 3 7 4 3 ...
 $ ps_calc_12    : int  1 1 2 2 1 2 0 1 2 5 ...
 $ ps_calc_13    : int  5 1 7 4 1 0 0 3 1 0 ...
 $ ps_calc_14    : int  8 9 7 9 3 9 10 6 5 6 ...
 $ ps_calc_15_bin: int  0 0 0 0 0 0 0 1 0 0 ...
 $ ps_calc_16_bin: int  1 1 1 0 0 1 1 0 1 1 ...
 $ ps_calc_17_bin: int  1 1 1 0 0 0 0 1 0 0 ...
 $ ps_calc_18_bin: int  0 0 0 0 1 1 0 0 0 0 ...
 $ ps_calc_19_bin: int  0 1 1 0 1 1 1 1 0 1 ...
 $ ps_calc_20_bin: int  1 0 0 0 0 1 0 0 1 0 ...
```

From the structure of the combined data it is evident that the class of variables are inappropriate and the dataset has many missing values. Hence we need to clean the data set.

Data Pre-Processing:

The first step of pre-processing the data is to convert the data types of variables to their appropriate class. Followed by the first step, next we have to do missing value analysis.

Converting the data types of variables:

As there are around 60 variables we could not write code to change the data type of each variable separately. So, I have created an excel file of variable names as one column and their class as other column. Using this excel file we could convert the class of variables by just using a series of if-else conditions in a for-loop.

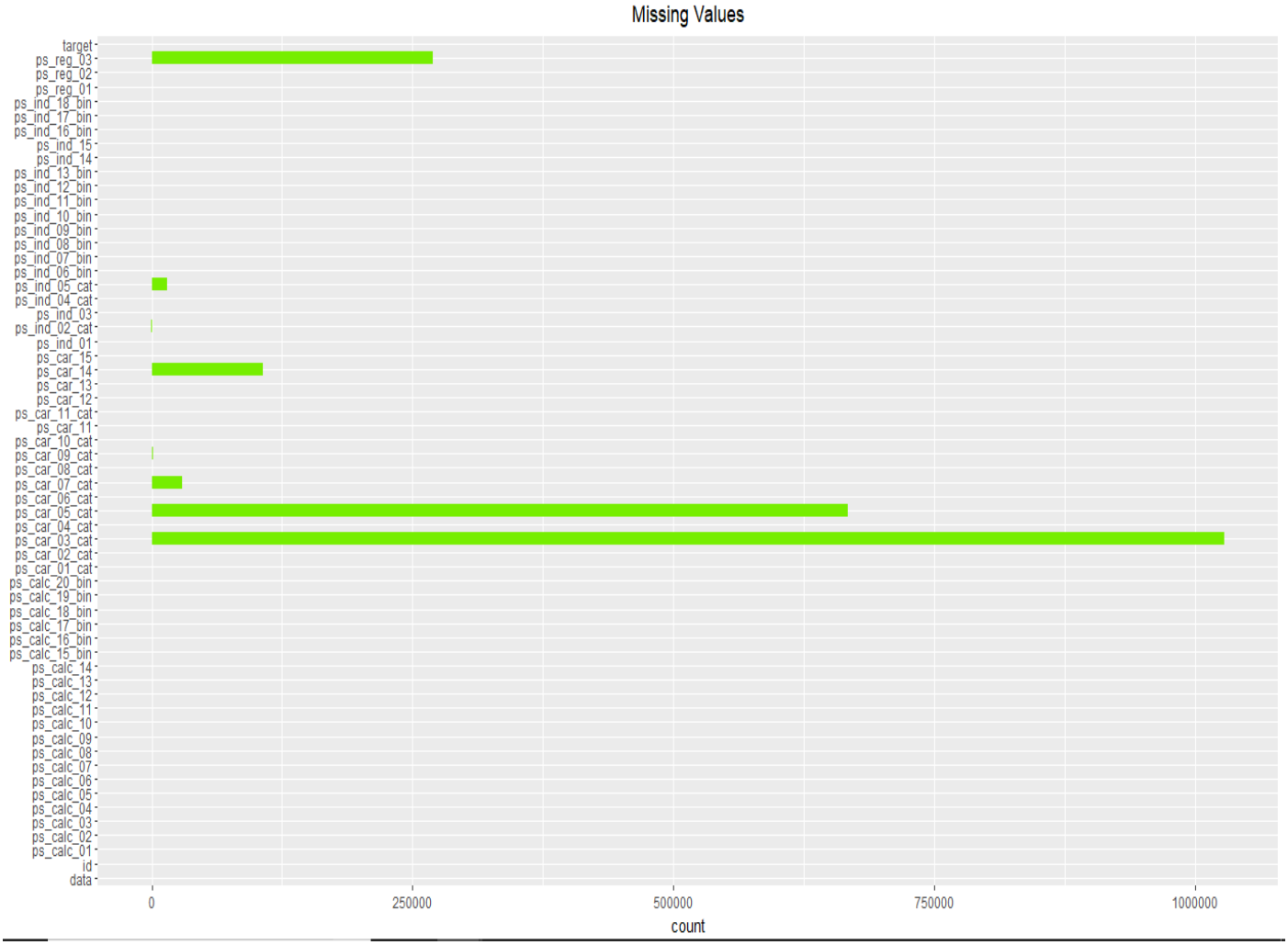
The structure of data after performing this looks like:

```
> str(data)
'data.frame':	1488028 obs. of  56 variables:
 $ id            : num  7 9 13 16 17 19 20 22 26 28 ...
 $ data          : chr  "train" "train" "train" "train" ...
 $ target        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ ps_ind_01     : num  2 1 5 0 0 5 2 5 5 1 ...
 $ ps_ind_02_cat : Factor w/ 4 levels "1","2","3","4": 2 1 4 1 2 1 1 1 1 1 ...
 $ ps_ind_03     : Ord.factor w/ 12 levels "0"<"1"<"2"<"3"<..: 6 8 10 3 1 5 4 5 4 3 ...
 $ ps_ind_04_cat : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 1 ...
 $ ps_ind_05_cat : Factor w/ 7 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_06_bin : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 2 1 1 ...
 $ ps_ind_07_bin : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 2 ...
 $ ps_ind_08_bin : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 1 2 1 ...
 $ ps_ind_09_bin : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
 $ ps_ind_10_bin : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_11_bin : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_12_bin : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_13_bin : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_14     : Ord.factor w/ 5 levels "0"<"1"<"2"<"3"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_15     : Ord.factor w/ 14 levels "0"<"1"<"2"<"3"<..: 12 4 13 9 10 7 9 14 7 5 ...
 $ ps_ind_16_bin : Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 2 2 1 ...
 $ ps_ind_17_bin : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
 $ ps_ind_18_bin : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 2 ...
 $ ps_reg_01     : num  0.7 0.8 0 0.9 0.7 0.9 0.6 0.7 0.9 0.9 ...
 $ ps_reg_02     : num  0.2 0.4 0 0.2 0.6 1.8 0.1 0.4 0.7 1.4 ...
 $ ps_car_01_cat : Factor w/ 12 levels "0","1","10","11",..: 3 4 10 10 4 3 9 4 3 4 ...
 $ ps_car_02_cat : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 2 2 1 ...
 $ ps_car_04_cat : Factor w/ 10 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ ps_car_06_cat : Factor w/ 18 levels "0","1","2","3",..: 5 12 15 12 15 15 12 12 15 15 ...
 $ ps_car_07_cat : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ ps_car_08_cat : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ ps_car_09_cat : Factor w/ 5 levels "0","1","2","3",..: 1 3 3 4 3 1 1 3 1 3 ...
 $ ps_car_10_cat : Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ ps_car_11_cat : Factor w/ 104 levels "1","2","3","4",..: 12 19 60 104 82 104 99 30 68 104 ...
 $ ps_car_11     : Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 3 4 2 2 4 3 3 4 4 3 ...
 $ ps_car_12     : num  0.4 0.316 0.316 0.374 0.316 ...
 $ ps_car_13     : num  0.884 0.619 0.642 0.543 0.566 ...
 $ ps_car_15     : num  3.61 2.45 3.32 2 2 ...
 $ ps_calc_01    : num  0.6 0.3 0.5 0.6 0.4 0.7 0.2 0.1 0.9 0.7 ...
 $ ps_calc_02    : num  0.5 0.1 0.7 0.9 0.6 0.8 0.6 0.5 0.8 0.8 ...
 $ ps_calc_03    : num  0.2 0.3 0.1 0.1 0 0.4 0.5 0.1 0.6 0.8 ...
 $ ps_calc_04    : Ord.factor w/ 6 levels "0"<"1"<"2"<"3"<..: 4 3 3 3 3 4 3 2 4 3 ...
 $ ps_calc_05    : Ord.factor w/ 7 levels "0"<"1"<"2"<"3"<..: 2 2 3 5 3 2 3 3 2 3 ...
 $ ps_calc_06    : Ord.factor w/ 11 levels "0"<"1"<"2"<"3"<..: 11 10 10 8 7 9 9 8 8 9 ...
 $ ps_calc_07    : Ord.factor w/ 10 levels "0"<"1"<"2"<"3"<..: 2 6 2 2 4 3 2 2 4 3 ...
 $ ps_calc_08    : Ord.factor w/ 12 levels "1"<"2"<"3"<"4"<..: 10 8 8 8 10 11 8 6 9 9 ...
 $ ps_calc_09    : Ord.factor w/ 8 levels "0"<"1"<"2"<"3"<..: 2 2 3 5 3 4 4 2 5 2 ...
 $ ps_calc_10    : Ord.factor w/ 26 levels "0"<"1"<"2"<"3"<..: 6 8 8 3 13 9 11 14 12 12 ...
 $ ps_calc_11    : Ord.factor w/ 21 levels "0"<"1"<"2"<"3"<..: 10 4 5 3 4 5 4 8 5 4 ...
 $ ps_calc_12    : Ord.factor w/ 12 levels "0"<"1"<"2"<"3"<..: 2 2 3 3 2 3 1 2 3 6 ...
 $ ps_calc_13    : Ord.factor w/ 16 levels "0"<"1"<"2"<"3"<..: 6 2 8 5 2 1 1 4 2 1 ...
 $ ps_calc_14    : Ord.factor w/ 25 levels "0"<"1"<"2"<"3"<..: 9 10 8 10 4 10 11 7 6 7 ...
 $ ps_calc_15_bin: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
 $ ps_calc_16_bin: Factor w/ 2 levels "0","1": 2 2 2 1 1 2 2 1 2 2 ...
 $ ps_calc_17_bin: Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 2 1 1 ...
 $ ps_calc_18_bin: Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
 $ ps_calc_19_bin: Factor w/ 2 levels "0","1": 1 2 2 1 2 2 2 2 1 2 ...
 $ ps_calc_20_bin: Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 1 2 1 ...
```

Now the structure of the dataset looks nice and the class of variables are appropriate to their own properties. Next step of data pre-processing is to identify if there are any missing values in the data and deal with them.

Missing Value Analysis:

Let us check if there are any missing values in the data. We get to see that there are many missing values in the data. So, let us built a data frame with count of missing values of each variable. Upon plotting the count on a bar plot it looks like,



From the plot we can observe that most of the missing values are concentrated in 3 to 4 variables and most of the other variables do not have missing values. So, let us remove the variables with >5% of values as missing and impute the missing values in other variables.

While imputing the missing values I have used mode for categorical variables and mean for numerical variables.

Now the data looks clean and free of missing values and ready for building a classification model.

We have to divide the combined dataset back into train and test datasets and build the model using the train dataset. I have again divided the train dataset into train and test in order to build the model and check the performance of the model.

Model Building:

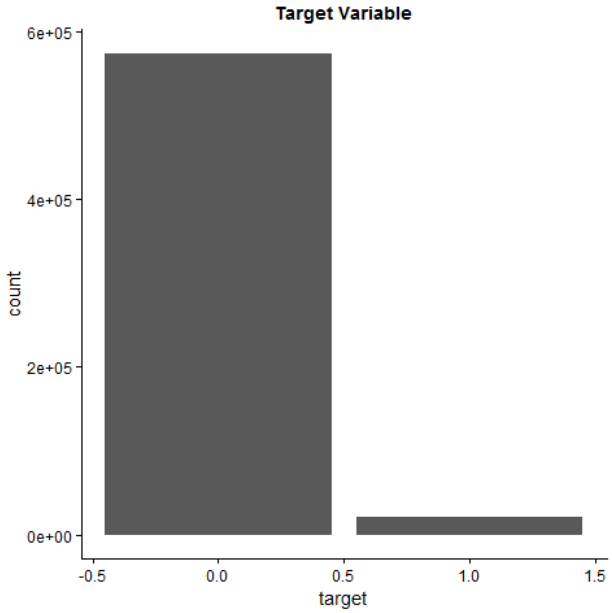
As most of the variables are factors and random forest algorithm works well with factors, I have chosen random forest as our model.

Random forest takes only a maximum of 53 independent variables so, we have to remove 1 variable. Here, the variable ps_car_11_cat has more number of distinct factor levels compared to other variables. So, I have removed this variable from the datasets.

Before building the model let us have a look at the distribution of target variable in train dataset.

```
> table (train$target)/nrow (train)

          0           1
0.96355248 0.03644752
```

This is very uneven and not suitable for building the model.



In order to build the model, we first need to balance the dataset. I have used the “ROSE” package to balance the dataset.

Now, let us observe the distribution of target variable in the balanced dataset.

```
> table (df$target)/nrow (df)

        0         1
0.4995444 0.5004556
```

As the target variable is now evenly distributed we can build our random forest model on the new dataset. We have to divide the train dataset again into train and test datasets. I have taken a sample ratio of 70-30.

The random forest model is giving the following result.

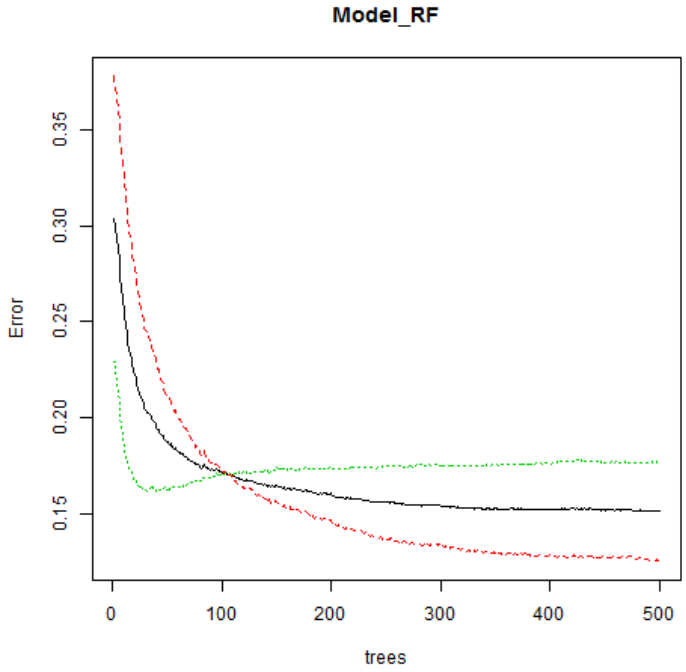
```
##     0     1   #Summary of Predicted target variable
## 14214 12786

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11808  2406
##          1  1574 11212
##
##                Accuracy : 0.8526
##                  95% CI : (0.8483, 0.8568)
##     No Information Rate : 0.5044
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7053
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8824
##             Specificity : 0.8233
##          Pos Pred Value : 0.8307
##          Neg Pred Value : 0.8769
##              Prevalence : 0.4956
##          Detection Rate : 0.4373
##    Detection Prevalence : 0.5264
##       Balanced Accuracy : 0.8529
##
##        'Positive' Class : 0
##
```

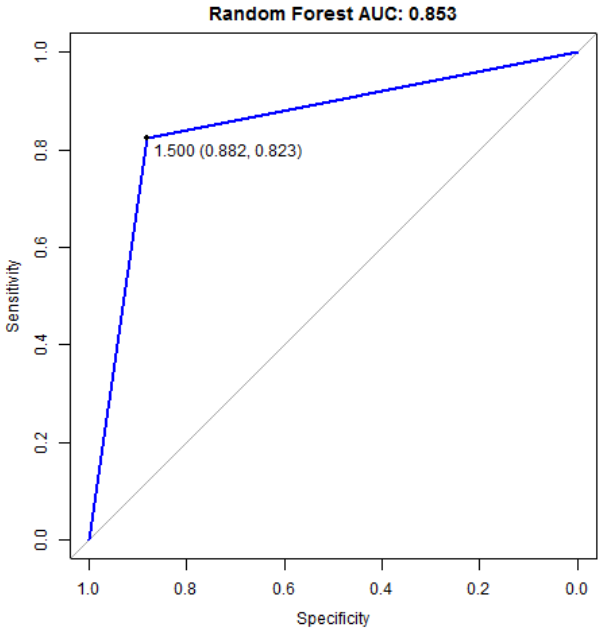
We got an accuracy of 85.3% with the above model. This is pretty good compared to other models and we can freeze the model and use it to predict the test dataset.

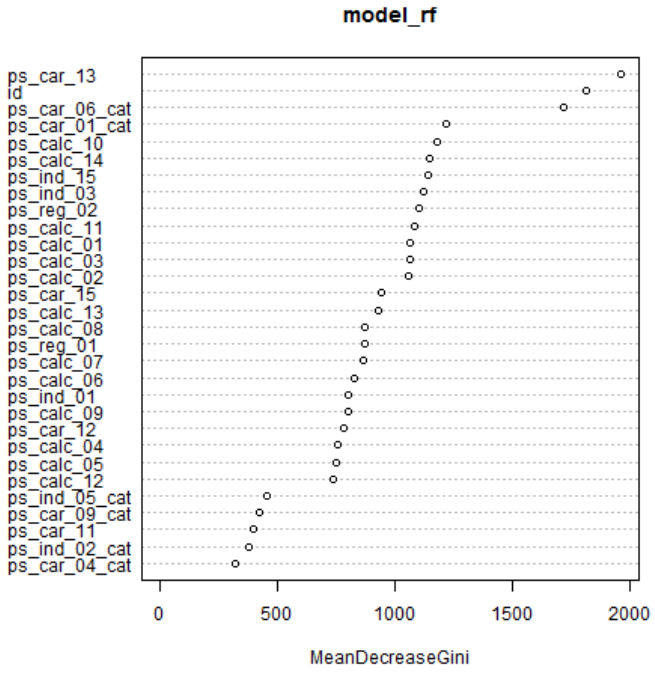
We could even improve the model by including only the important variables but when I tried it didn't change considerably, so I didn't include it here.

Now let us see the plots of the model and variable importance plot based on gini index (Shown in last page).

Once the model is built we have to use it to predict the test dataset. I have predicted the probabilities of the target variable to be 0 and to be 1. Final result is stored in Final_Submission file.


Model_RF
Error
0.15
0.20
0.25
0.30
0.35
0
100
200
300
400
500
trees


Random Forest AUC: 0.853
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.500 (0.882, 0.823)
1.0
0.8
0.6
0.4
0.2
0.0
Specificity


model_rf
ps_car_13
id
ps_car_06_cat
ps_car_01_cat
ps_calc_10
ps_calc_14
ps_ind_15
ps_ind_03
ps_reg_02
ps_calc_11
ps_calc_01
ps_calc_03
ps_calc_02
ps_car_15
ps_calc_13
ps_calc_08
ps_reg_01
ps_calc_07
ps_calc_06
ps_ind_01
ps_calc_09
ps_car_12
ps_calc_04
ps_calc_05
ps_calc_12
ps_ind_05_cat
ps_car_09_cat
ps_car_11
ps_ind_02_cat
ps_car_04_cat
0
500
1000
1500
2000
MeanDecreaseGini