Data and Text Mining

MCDA5580

Master of Science in Computing and Data Analytics

Assignment-2

Classification: Acceptability of cars

Submitted by:

Vivekanand Boopathy	A00425792
Parijat Bandyopadhyay	A00430847
Kothai Kannappan Murugappan	A004727876

Submitted to:

Dr. Pawan Lingras

SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

Table of Contents

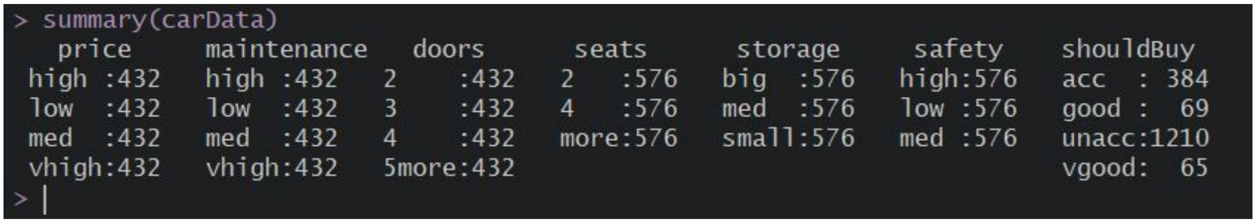
Executive summary

The objective of this report is to describe our study of decision to buy a car and analyze its results with respect to the features of the car. The study includes creating a model to best predict the customers decision to buy a car from 1728 observations of historical data. We decided that this problem is a classification problem and hence we have tried out “rpart”and “random forest” algorithms in order to achieve this. From our study, Random forest (with 10-fold validation) yields the best accuracy results for prediction. The detailed comparison results of the both algorithms have been discussed in detail in the report.

Data Summary

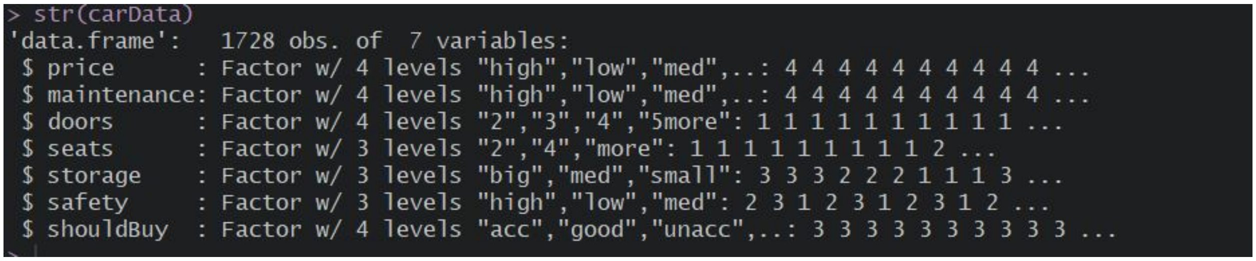
The historic data on the customers decision to buy car has been cleaned and given as CSV file for our purpose. The total number of observations are 1728.The independent variables in our case are price, maintenance, doors, seats, storage, and safety which are categorical in nature. While our dependent variable, “shouldBuy” is also a categorical value.

Following table is the summary of the data:



```
> summary(carData)
   price      maintenance    doors        seats        storage       safety      shouldBuy
 high :432   high :432    2    :432   2    :576   big  :576   high:576   acc  : 384
 low  :432   low  :432    3    :432   4    :576   med  :576   low :576   good :  69
 med  :432   med  :432    4    :432   more:576   small:576   med :576   unacc:1210
 vhigh:432   vhigh:432    5more:432                                      vgood:  65
> 
```

Following table shows the categorical variables in data and the levels for each.



```
> str(carData)
'data.frame':   1728 obs. of  7 variables:
 $ price      : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ maintenance: Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ doors      : Factor w/ 4 levels "2","3","4","5more": 1 1 1 1 1 1 1 1 1 1 ...
 $ seats      : Factor w/ 3 levels "2","4","more": 1 1 1 1 1 1 1 1 1 2 ...
 $ storage    : Factor w/ 3 levels "big","med","small": 3 3 3 2 2 2 1 1 1 3 ...
 $ safety     : Factor w/ 3 levels "high","low","med": 2 3 1 2 3 1 2 3 1 2 ...
 $ shouldBuy  : Factor w/ 4 levels "acc","good","unacc",..: 3 3 3 3 3 3 3 3 3 3 ...
```

Independent Variables:

price (vhigh, high, med, low)

maintenance (vhigh, high, med, low)

doors (2, 3, 4, 5more)

seats (2, 4, more)

storage (big, med, small)

safety (high, low, med)

Dependent Variable:

shouldbuy (unacc, acc, good, vgood)

Data Analysis

Splitting Data

We have split the complete dataset into training and test dataset as train and test in 75:25 ratio respectively.

We also defined and store the independent and dependent variable in training set as x_train and y_train and similarly for test set as x_test and y_test for compatibility with Random Forest syntax in R.

Analysis by Decision Trees algorithm:

We import the rpart library which is used to create the decision tree which uses Classification and regression trees (CART) algorithm.

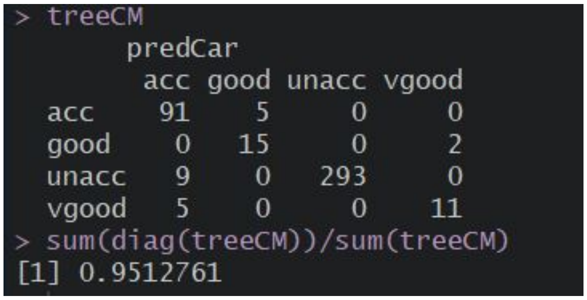
We try out different values of minsplit for creating our decision trees and corresponding accuracies have been tabulated

Algorithm	Minsplit	Accuracy
ID3	1	0.9512761
ID3	10	0.9512761
ID3	50	0.9419954
ID3	100	0.8723898

On trying values of minsplit lesser than 10 there is not any improvement in accuracy, So minsplit = 10 as our optimum model.

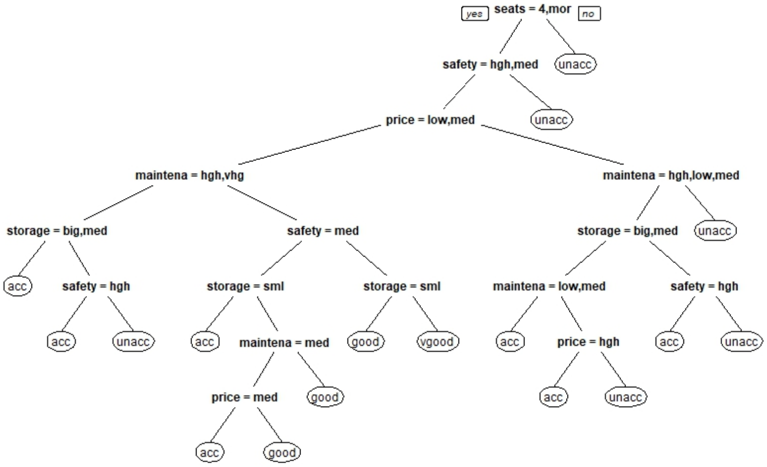
Below is the confusion matrix of the decision tree with minsplit =10

Confusion matrix



```
> treeCM
       predCar
        acc good unacc vgood
  acc    91    5     0     0
  good    0   15     0     2
  unacc   9    0   293     0
  vgood   5    0     0    11
> sum(diag(treeCM))/sum(treeCM)
[1] 0.9512761
```

On visualizing the rules used create our decision tree, we get following:



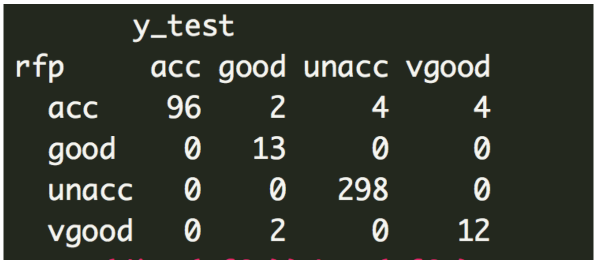
Analysis by Random Forest algorithm (with different “ntree” values):

Unlike the decision tree method, the Random-Forest builds multiple decision trees and uses prediction results from all the trees to get a more accurate and stable prediction.

In Random-Forest algorithm, the “ntree” parameter has been varied and the corresponding accuracy of the model has been observed.

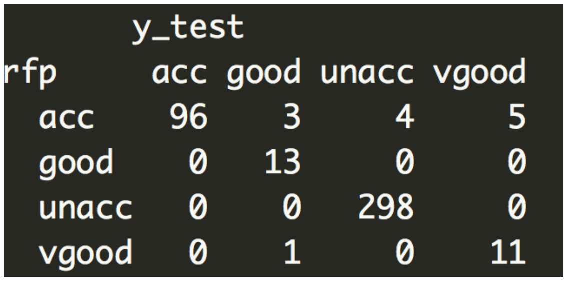
Following tables shows confusion matrix for different “ntree” values and corresponding achieved accuracies:

ntree=500



```
       y_test
rfp      acc good unacc vgood
  acc     96    2     4     4
  good     0   13     0     0
  unacc    0    0   298     0
  vgood    0    2     0    12
```

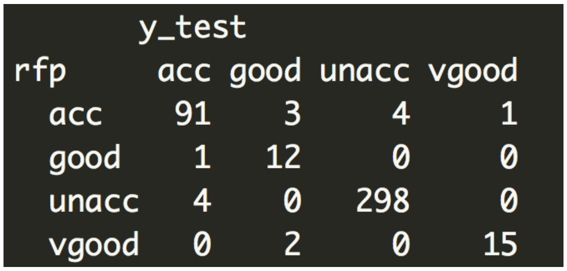
Achieved Accuracy of the model on the test dataset is: 0.9791183 ntree=50



```
       y_test
rfp      acc good unacc vgood
  acc     96    3     4     5
  good     0   13     0     0
  unacc    0    0   298     0
  vgood    0    1     0    11
```

Achieved Accuracy of the model on the test dataset is: 0.9698376

ntree=10

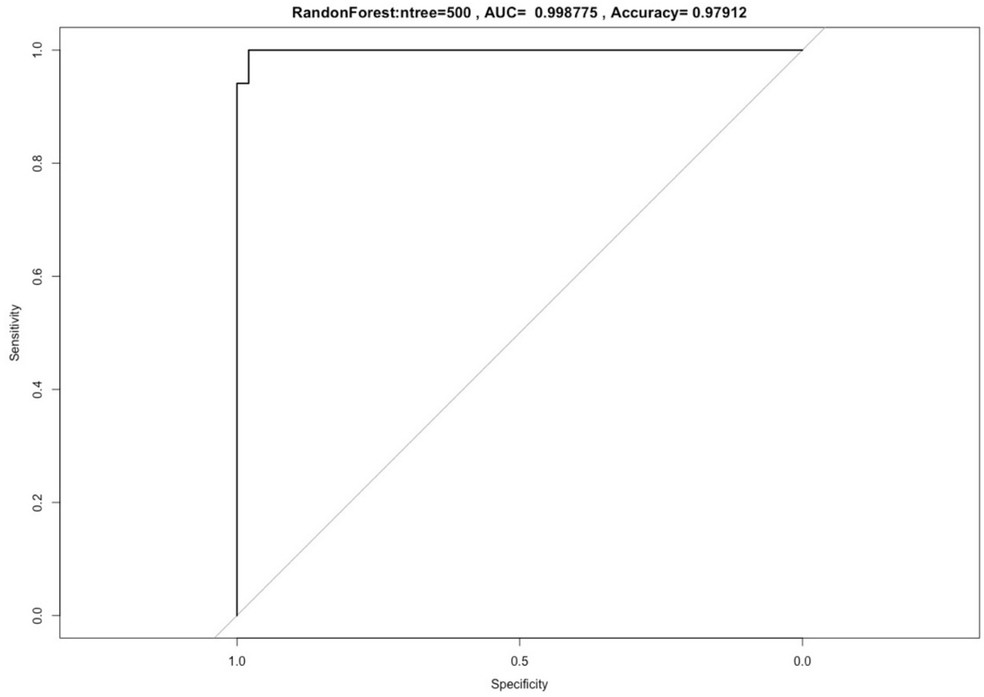


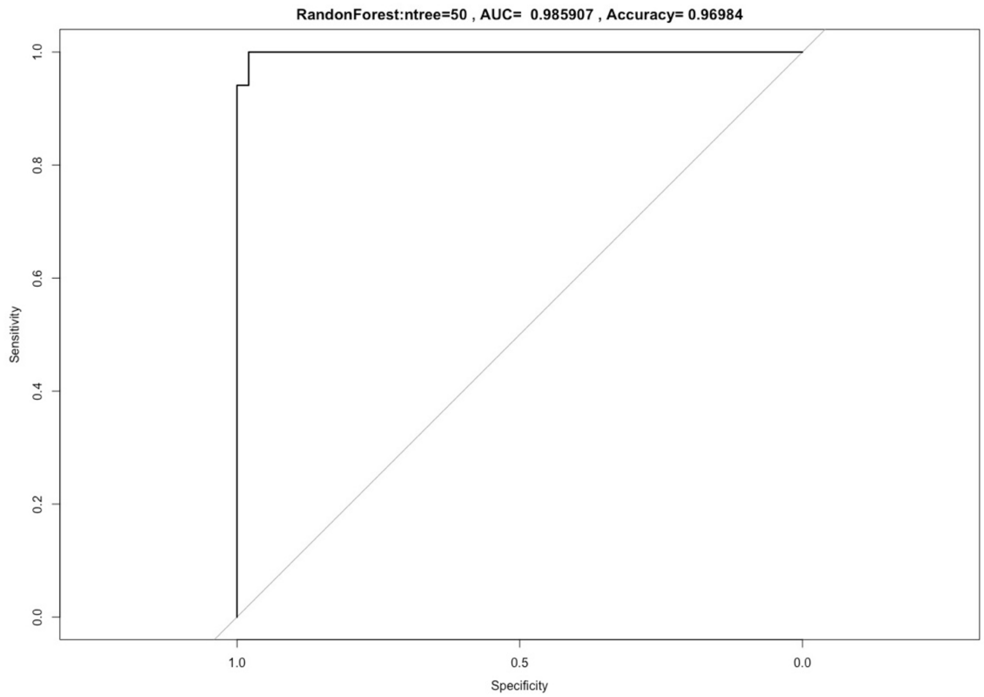
```
       y_test
rfp      acc good unacc vgood
  acc     91    3     4     1
  good     1   12     0     0
  unacc    4    0   298     0
  vgood    0    2     0    15
```

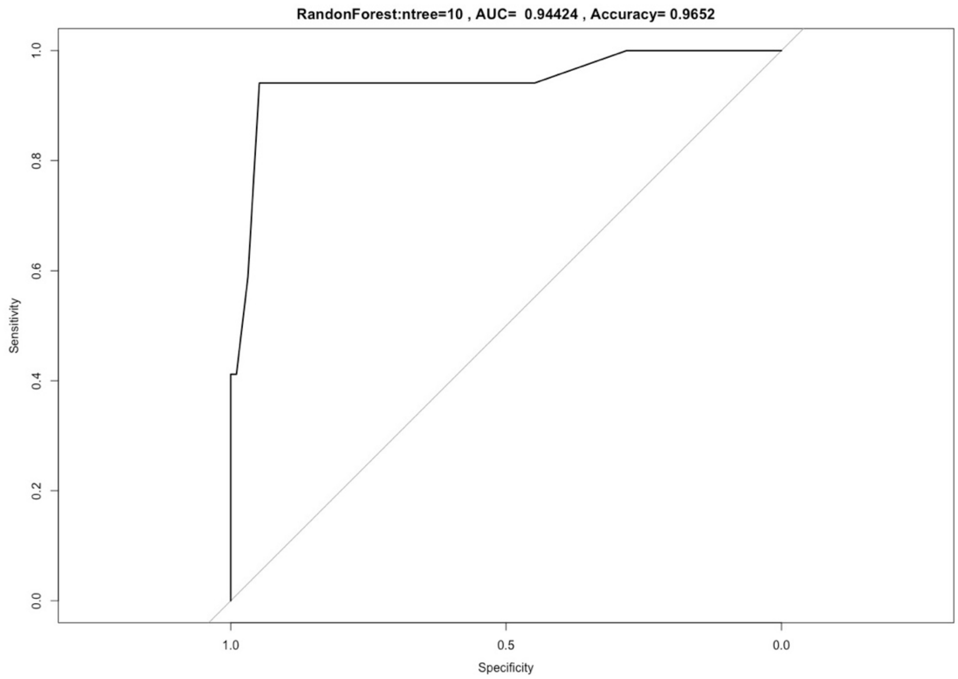
Achieved Accuracy of the model on the test dataset is: 0.9651972

It is evident from the above table that a higher value of “ntree” provides a higher accuracy, provided other parameters kept unchanged.

The following figures show ROC curves and AUC values for different “ntree” values:







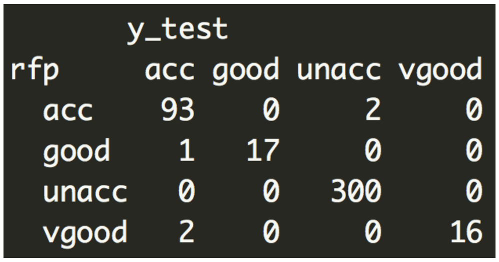
It can be concluded from the above curves that a higher "ntree" value provides better AUC as well as accuracy i.e. better model. So, a model with "ntree" value of 500 produces best model.

Analysis by Random Forest algorithm (with 10-fold cross-validation):

The 10-fold cross validation process splits the training data set into a 90:10 ratio to a virtual training and test data sets, to test the validity of the model and its prediction capability on an unknown test data set.

In this analysis the 10-fold cross validation algorithm is applied with a “repeats” parameter value of 3, to take an average from three predictions.

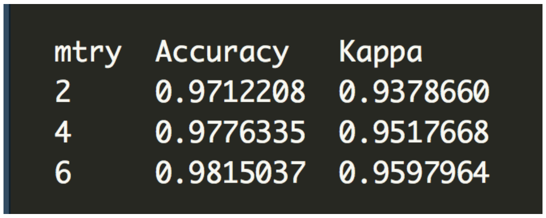
Table below shows confusion matrix for the analysis with 10-fold cross validation and an “ntree” value of 500:



```
       y_test
rfp     acc good unacc vgood
  acc    93    0     2     0
  good    1   17     0     0
  unacc   0    0   300     0
  vgood   2    0     0    16
```

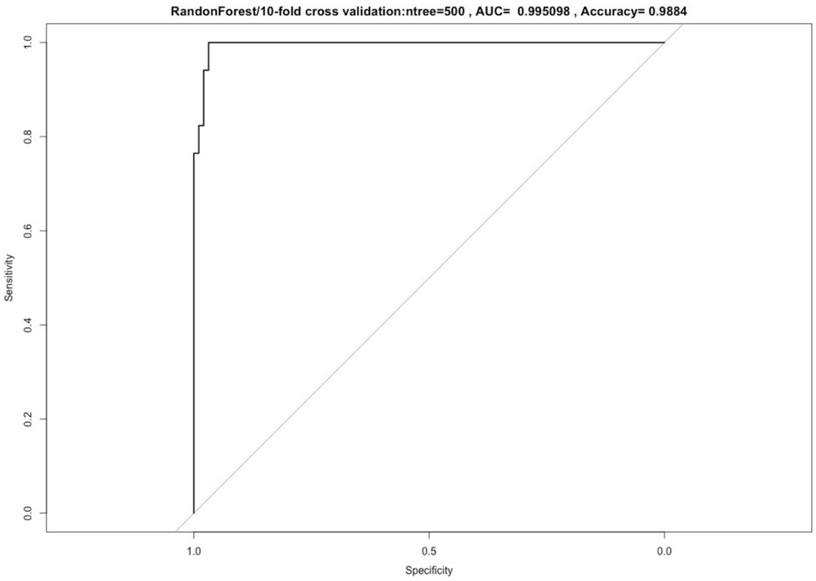
The achieved accuracy of the model is: 0.9883991, which is higher than the accuracy value of the model obtained also with an “ntree” value of 500 without cross validation.

Table below shows the summary of the analysis:

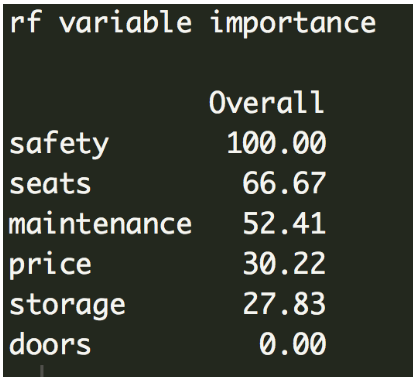


```
  mtry  Accuracy   Kappa
  2     0.9712208  0.9378660
  4     0.9776335  0.9517668
  6     0.9815037  0.9597964
```

Below is the ROC plot for this 10-fold validation analysis:



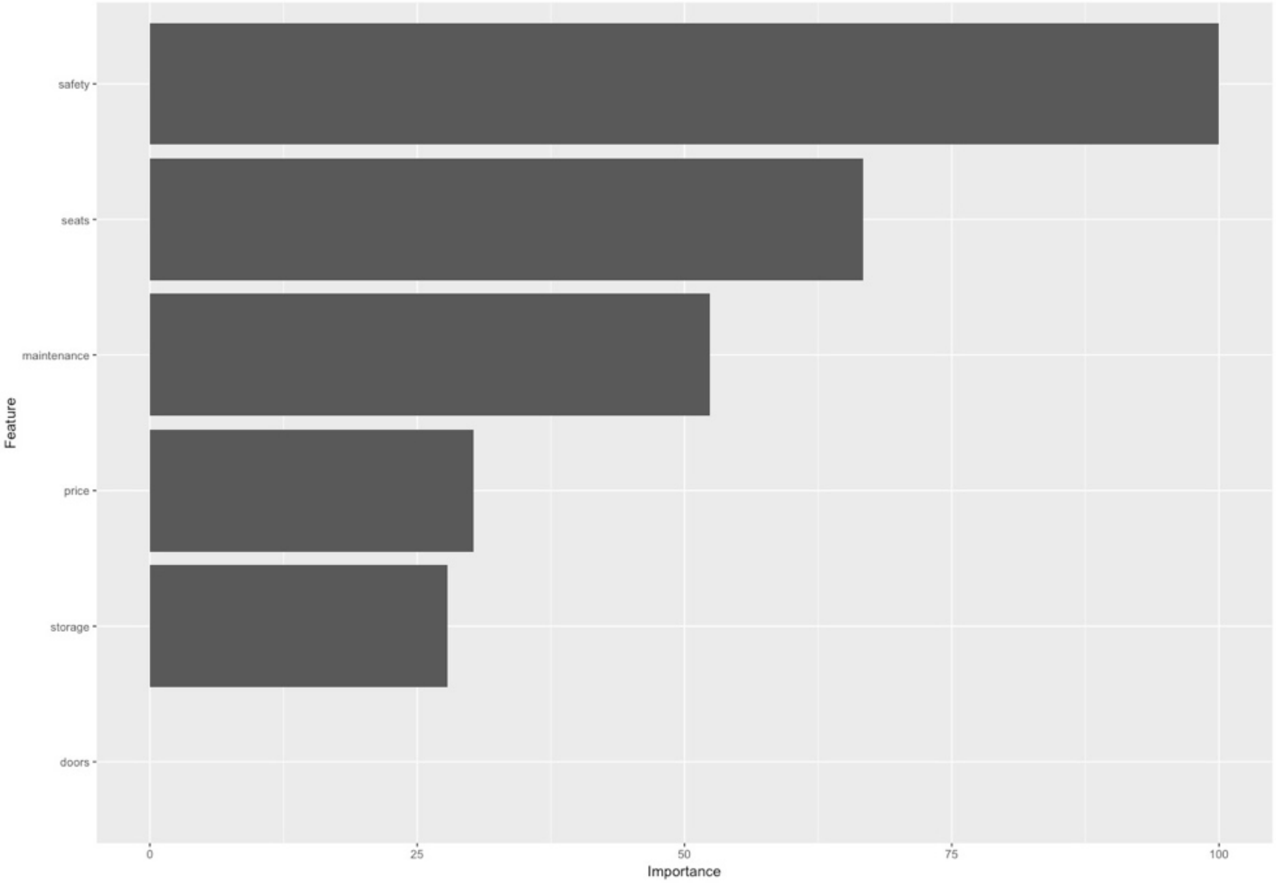
The following table shows the importance of each dependent variable(feature) obtained from 10-fold classification analysis:



```
rf variable importance

            Overall
safety       100.00
seats         66.67
maintenance   52.41
price         30.22
storage       27.83
doors          0.00
```

The following plot shows the same information (i.e. Feature vs Importance) in a bar-graph for better understanding:



Conclusion

The following table summarizes different “ntree values” and variations for random-forest algorithm and achieved accuracies:

Algorithm	ntree value	Accuracy	AUC
Random-Forest	500	0.9791183	0.97912
Random-Forest	50	0.9698376	0.96984
Random-Forest	10	0.9651972	0.94424
Random-Forest with 10-fold cross validation	500	0.9883991	0.995098

It can be concluded that the Random-Forest algorithm (with 10-fold cross validation) with an “ntree” value of 500 is the best classifier model generator.

Note: For R-script kindly refer to the Resource folder.