MACHINE LEARNING TERM PROJECT

CS 6375.002

CAR EVALUATION

MITAL SURESH MODHA (MSM160530)

SHIVANGI GAUR (SXG163230)

Mital Suresh Modha (msm160530)
Shivangi Gaur (sxg163230)

Car Evaluation

Abstract: Cars are essentially part of our everyday lives. There are different types of cars as produced by different manufacturers; therefore, the buyer has a choice to make. Various machine learning techniques like KNN, SVM, Random Forest, GB (Gradient Boosting), LVQ (Linear Vector Quantization) is implemented to determine which car is acceptable by the buyer. This Car evaluation dataset is taken from UCI Machine Learning Repository derived from simple hierarchical decision model.

1. Problem Description:

Car Evaluation: Data Mining the Car Evaluation dataset contains the data created by Marko Bohanec. The dataset provides various information about the car, like buying price, maintenance price, number of doors, persons capacity, size of luggage boot and estimated car safety**.**

. The class label (class) in the dataset can have four possible values:

- Unacceptable(unacc) - the car is in an unacceptable quality for buyers.
- Acceptable(acc) - the car is in an acceptable quality for buyers.
- Good(good) - the car is in a good quality for buyers.
- Very Good(vgood) – the car is in a very good quality for buyers.

Two data frames are provided for building and validating the model:

- train_values corresponds to the independent variables for the training set
- validation_values is the independent variables that need predictions

The objective of this challenge is to predict the quality of the car. We have examined all the features of the dataset and try to find out the correlation between them.

2. Dataset Description:

Number of attributes: 6
Class Label: 1 (Class)
Number of classes: 4
Number of instances: 1728
Number of training instances: 1384
Number of testing instances: 344

Dataset: Car Evaluation

This Car Evaluation dataset is taken from UCI Machine learning repository derived from simple hierarchical decision model

Total no. of Observations: 1728

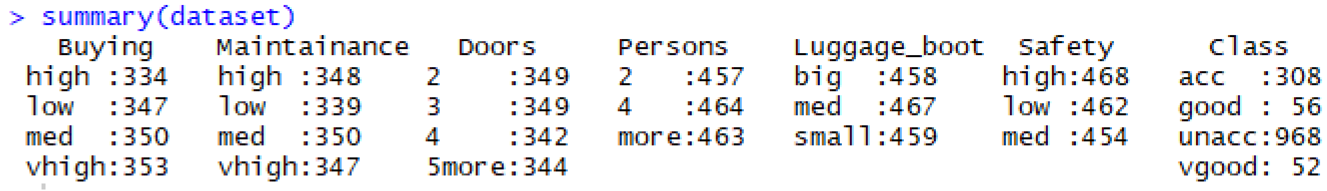
Input Variable:

- Buying price (vhigh, high, med, low)
- Price of the maintenance (vhigh, high, med, low)
- Number of doors (2, 3, 4, 5more)
- Persons capacity in terms of persons to carry (2, 4, more)
- Size of luggage boot (small, med, big)
- Estimated safety of the car (low, med, high)

Output Variable:

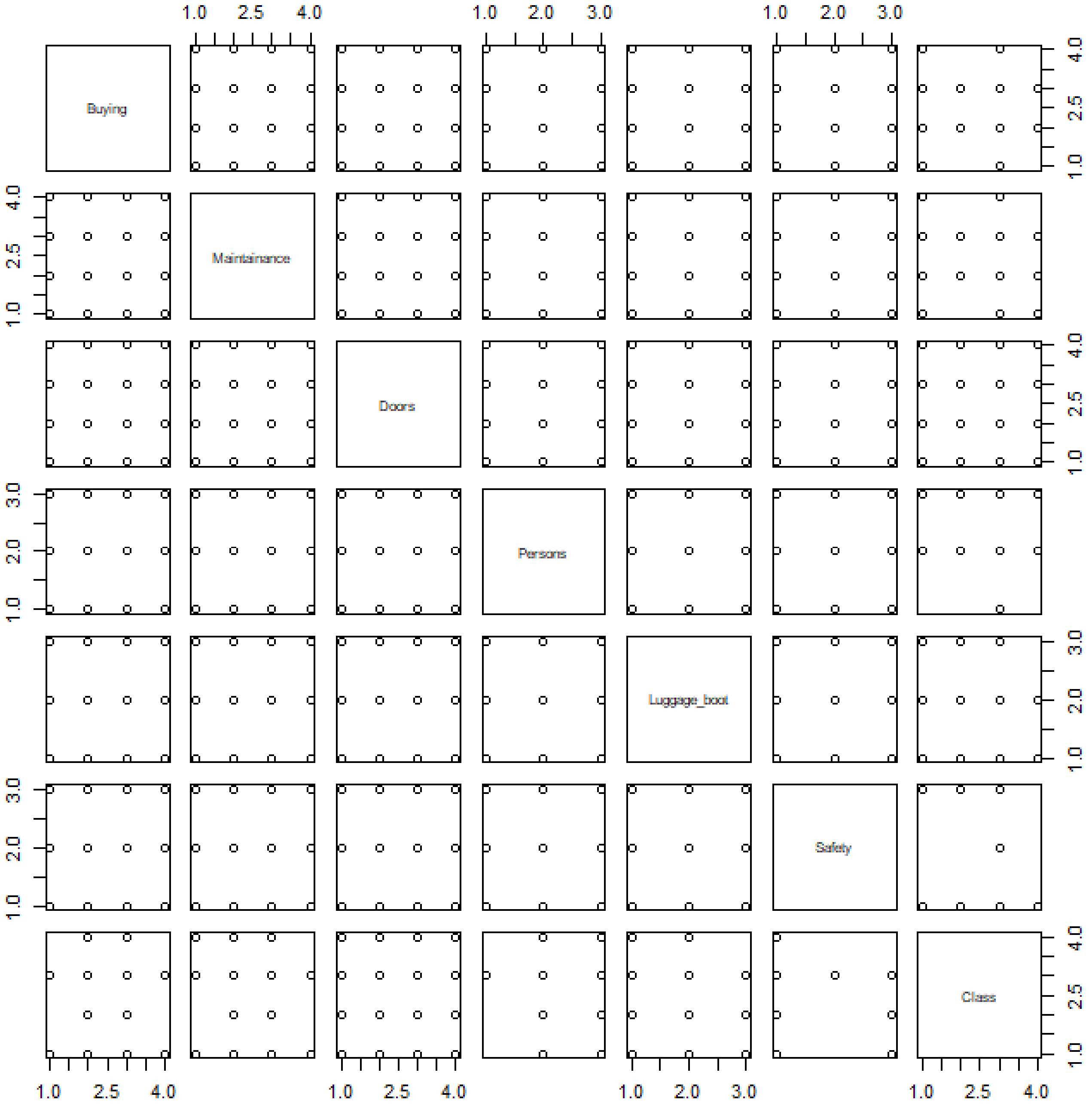
- Car acceptability (unacc, acc, good, vgood)

Summary of Dataset:

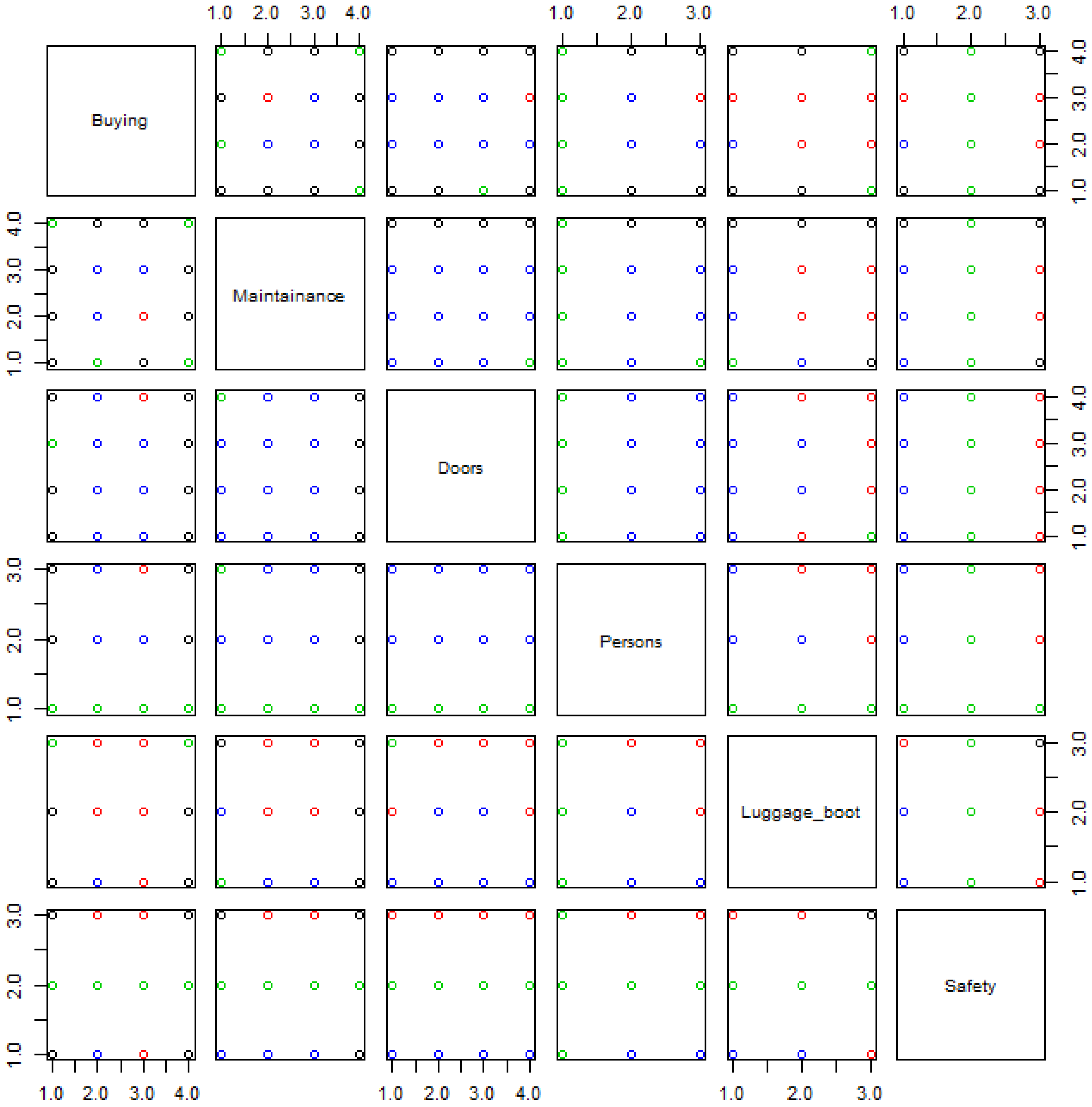


```
> summary(dataset)
   Buying     Maintainance    Doors       Persons     Luggage_boot  Safety      Class    
 high :334   high :348    2    :349   2   :457   big  :458    high:468   acc  :308  
 low  :347   low  :339    3    :349   4   :464   med  :467    low :462   good : 56  
 med  :350   med  :350    4    :342   more:463   small:459    med :454   unacc:968  
 vhigh:353   vhigh:347    5more:344                                      vgood: 52  
```

Dotplot of the complete graph and correlation between the data:



Dotplot of all the attributes classified by the class (color classification):



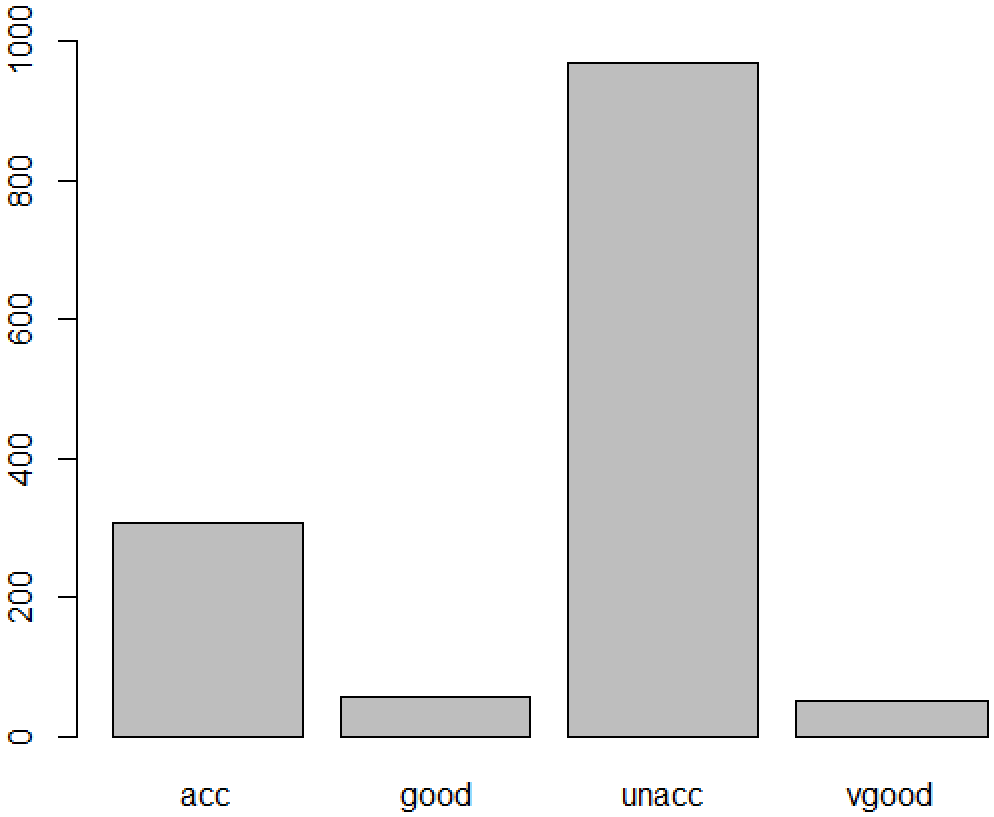
3. Steps performed to understand the dataset in detail:

- We performed the merge operation on train values and the train labels, i.e. to merge the independent values and the dependent class label. By performing this operation, we get the absolute number of the cars in each class label that is:

Unacceptable	Acceptable	Good	Very Good
968	308	56	52

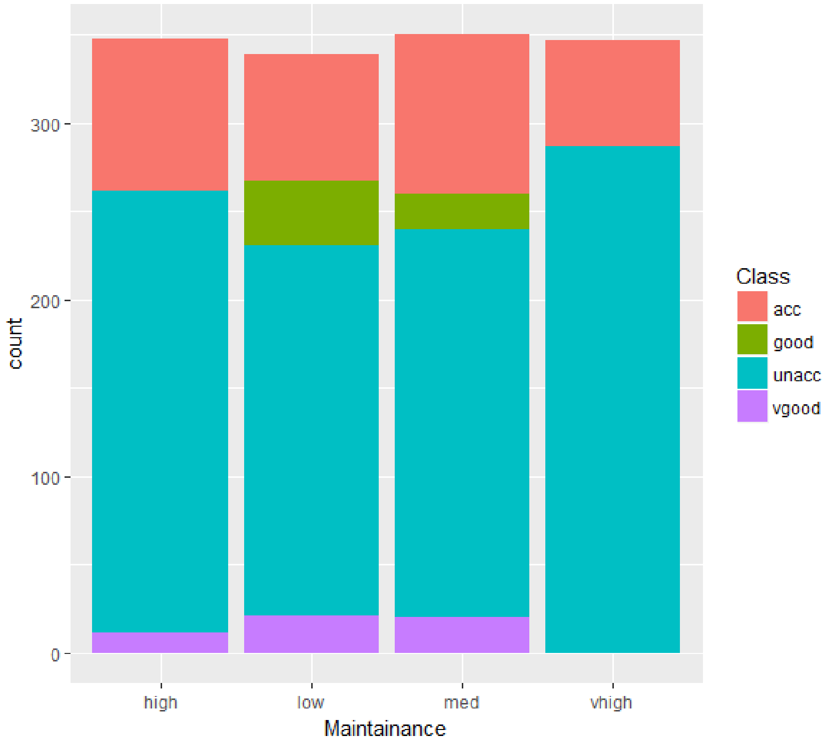
Frequency of cars:

Unacceptable	Acceptable	Good	Very Good
69.94%	22.25%	4.05%	3.76%

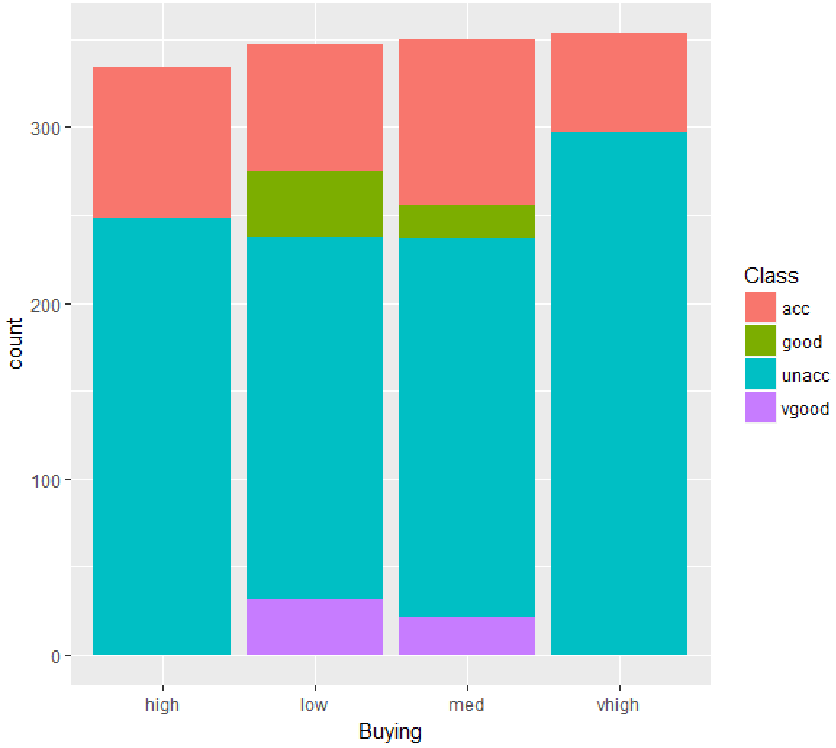


- For exploring and visualizing the data, we used the bar plots to understand the correlation of attributes with the class label.

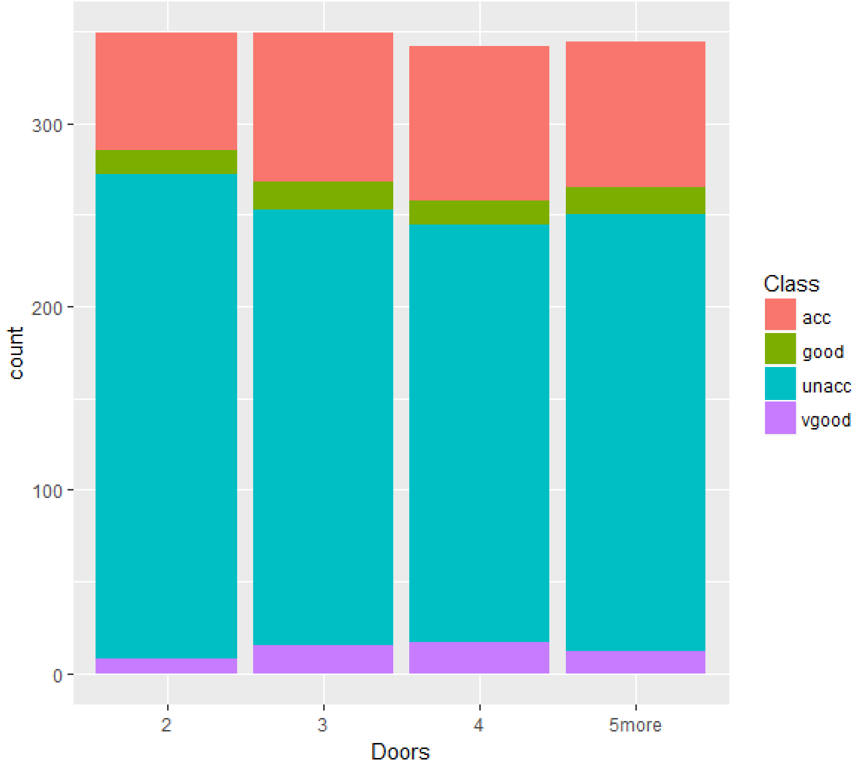
1. Maintenance



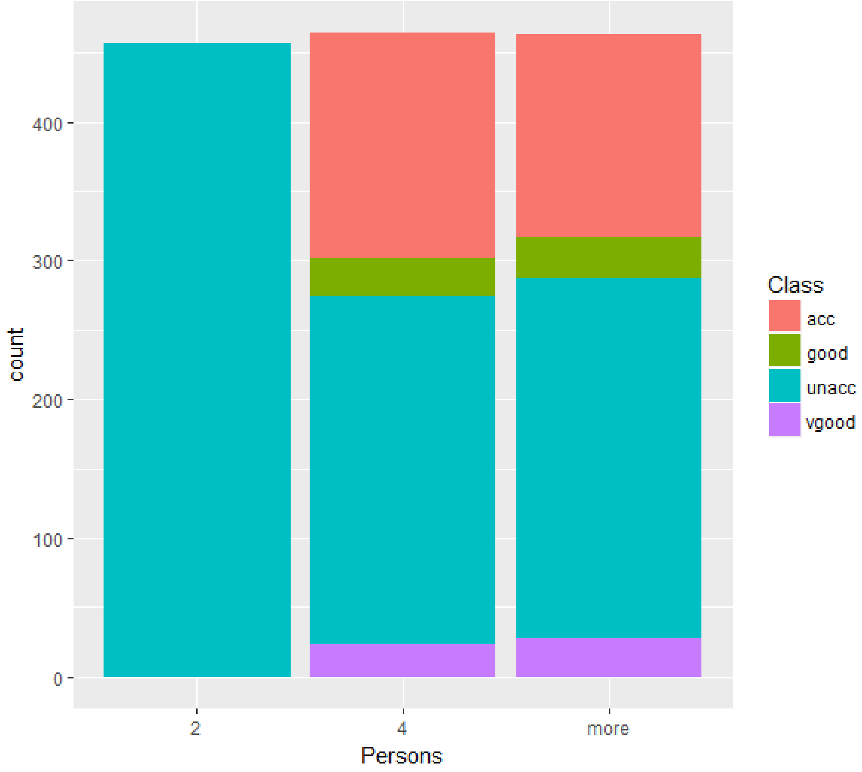
2. Buying Price



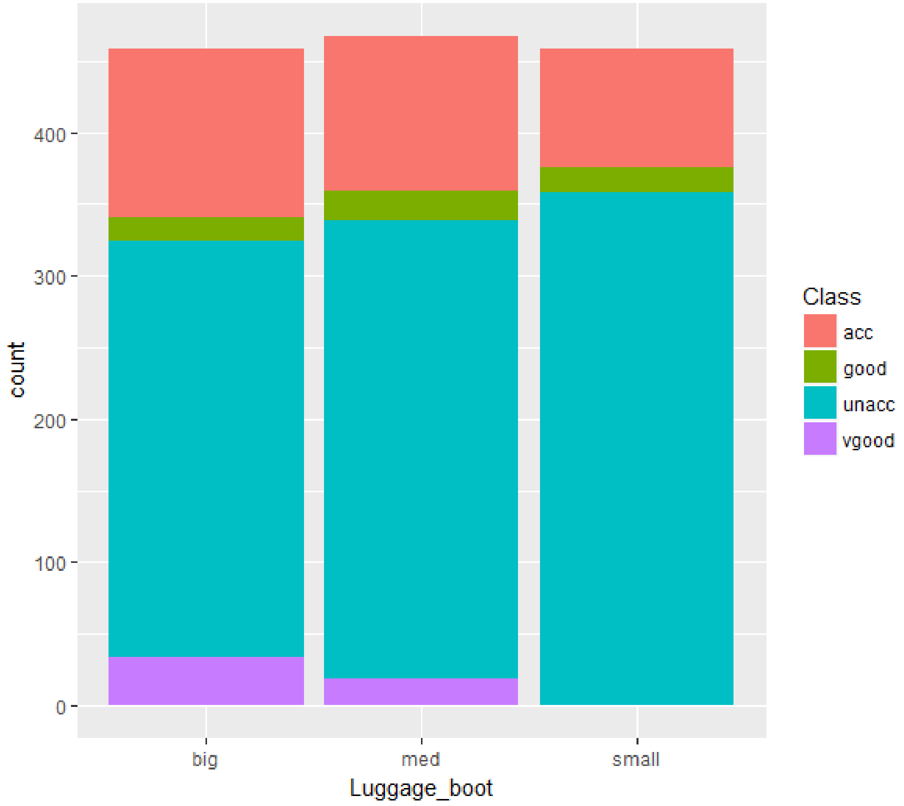
3. Number of Doors



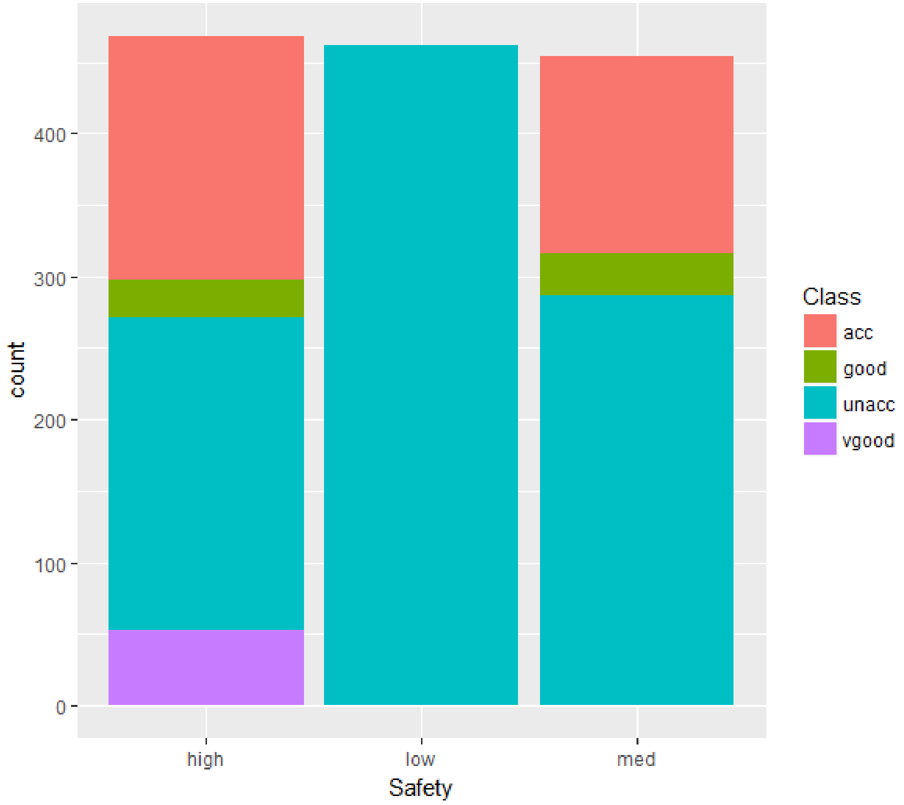
4. Person Capacity



5. Luggage Boot size



6. Car safety



4. Proposed Solutions and Methods with Analysis:

Following classifiers and methods have been used to get the best possible solution for the problem statement:

4.1. K Nearest Neighbor:

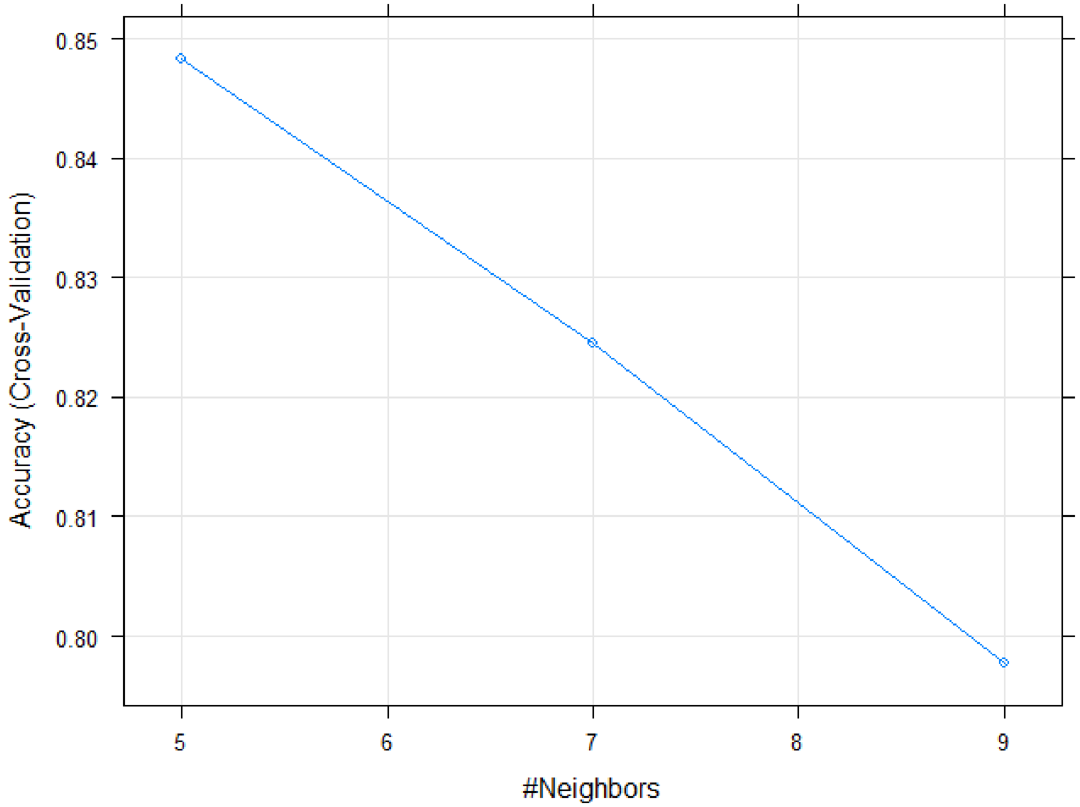
K-NN is a type of instance based learning, also called as lazy learning where the function is only approximated locally and all computation is deferred until classification.

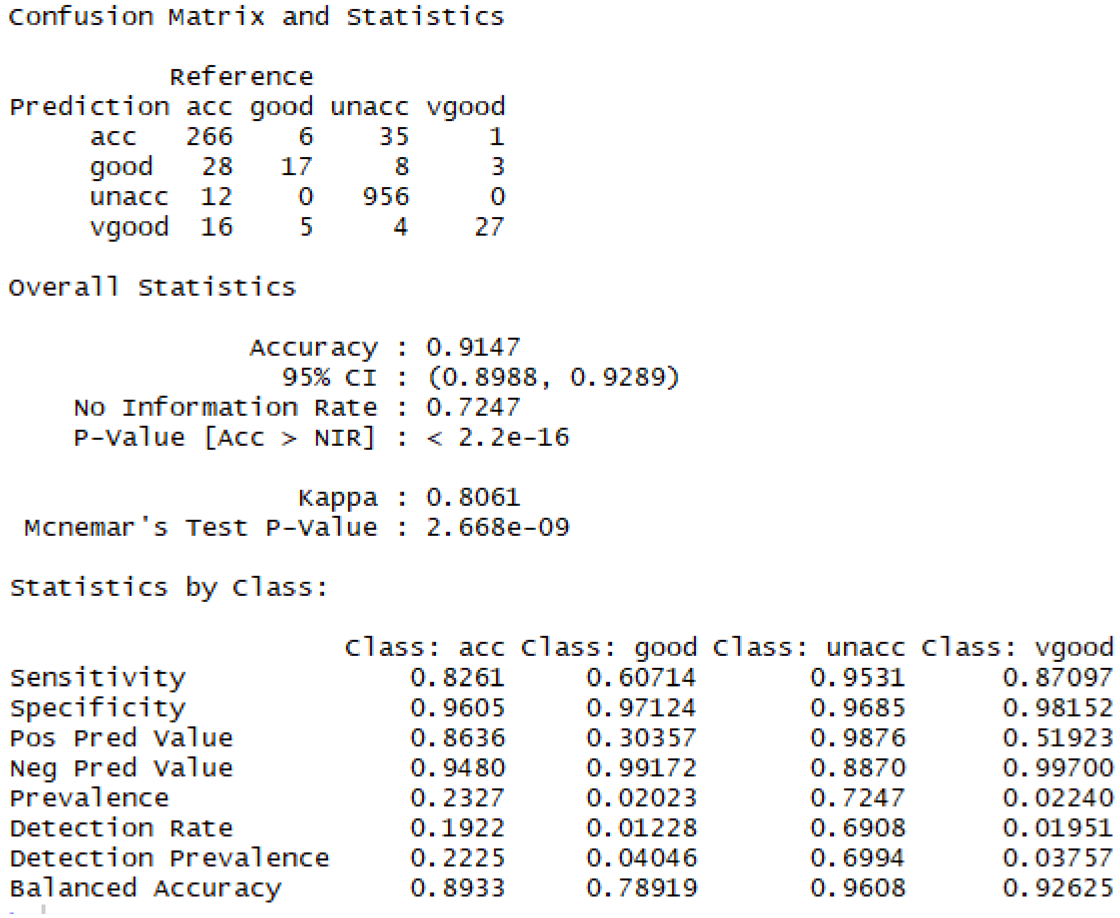
For our experiment-
R package used: Caret, e1071

One of the advantages in knn is that it is among the simplest of all the machine learning algorithms.
To measure the accuracy of our model, we used the error reduction with values of k.

Accuracy Table

Experiment #	Method	Cross-validation fold	Parameter K	Average Accuracy
1	kNN	10	5	0.8482
2	kNN	10	7	0.8244
3	kNN	10	9	0.7977





```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   266    6    35     1
     good   28   17     8     3
     unacc  12    0   956     0
     vgood  16    5     4    27

Overall Statistics

               Accuracy : 0.9147
                 95% CI : (0.8988, 0.9289)
    No Information Rate : 0.7247
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8061
 Mcnemar's Test P-Value : 2.668e-09

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.8261     0.60714       0.9531      0.87097
Specificity              0.9605     0.97124       0.9685      0.98152
Pos Pred Value           0.8636     0.30357       0.9876      0.51923
Neg Pred Value           0.9480     0.99172       0.8870      0.99700
Prevalence               0.2327     0.02023       0.7247      0.02240
Detection Rate           0.1922     0.01228       0.6908      0.01951
Detection Prevalence     0.2225     0.04046       0.6994      0.03757
Balanced Accuracy        0.8933     0.78919       0.9608      0.92625
```

- **The highest accuracy is 91.47 % on training dataset from the above confusion matrix.**
- **The accuracy on test dataset after submitting the predicted class values is 86.34%. Since the accuracy on training dataset is high but on testing dataset is low so it is not a good classifier to classify the cars. Hence it is not used for classifying.**

4.2. Support Vector Machine:

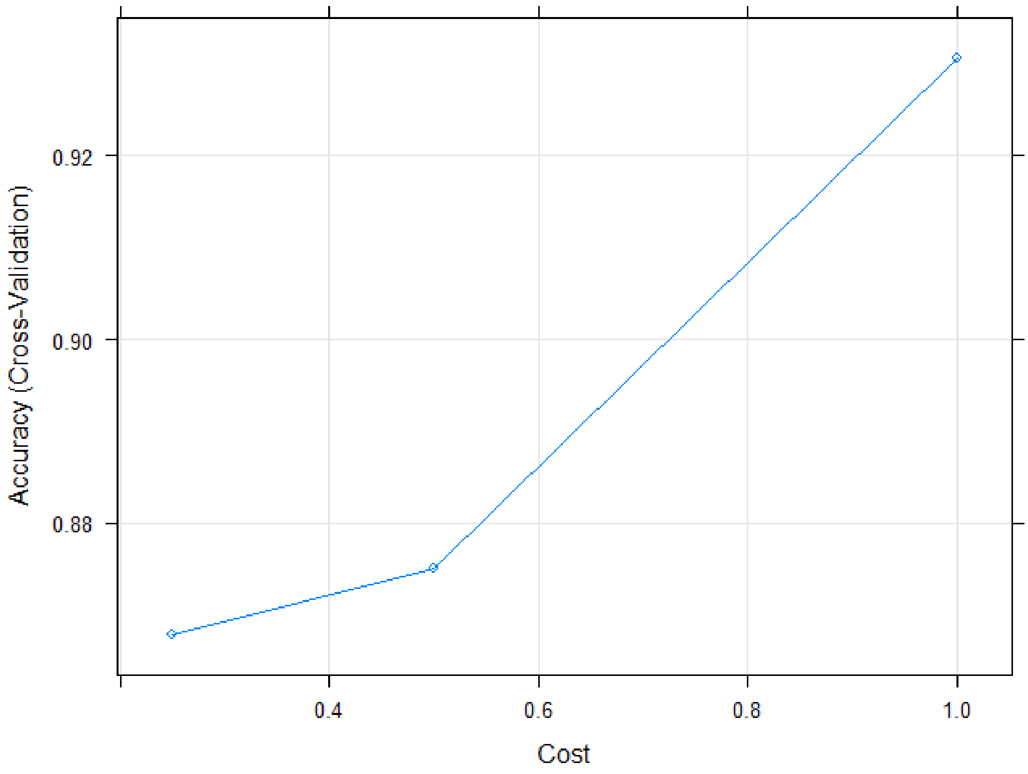
The SVM classifiers find the maximum-margin hyperplane using only those data instances that are closest to the separation boundary or "support vectors" to determine soft margin for classification. Nonlinearly separable data may be projected into a space of higher dimensionality with the use of a kernel function, making hyperplane separation in this new space occur more readily.

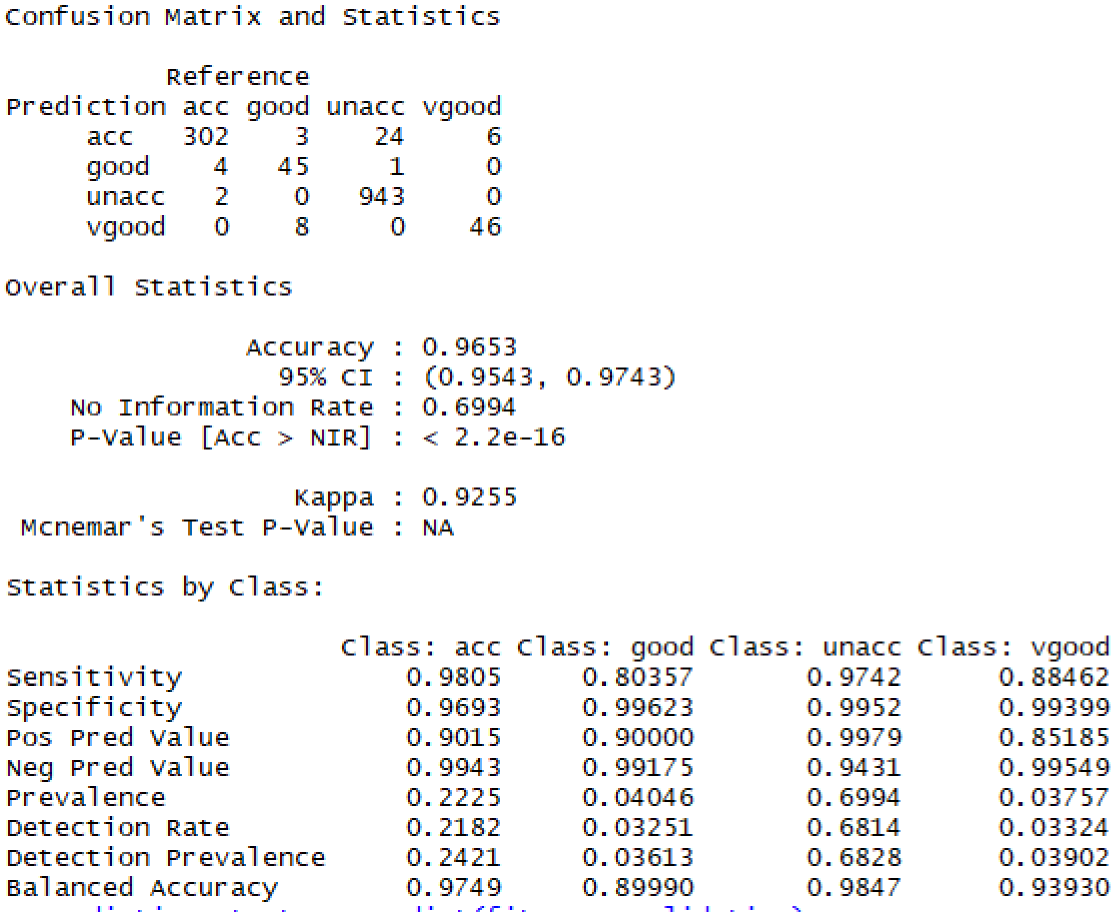
For our experiment-

R package used: Caret, kernlab

Accuracy Table

Experiment #	Method	Cross-validation fold	Parameter Cost	Average Accuracy
1	SVM	10	0.23	0.7021
2	SVM	10	0.50	0.7288
3	SVM	10	1.00	0.8508





```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   302    3    24     6
     good    4   45     1     0
     unacc   2    0   943     0
     vgood   0    8     0    46

Overall Statistics

               Accuracy : 0.9653
                 95% CI : (0.9543, 0.9743)
    No Information Rate : 0.6994
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9255
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.9805     0.80357       0.9742      0.88462
Specificity              0.9693     0.99623       0.9952      0.99399
Pos Pred Value           0.9015     0.90000       0.9979      0.85185
Neg Pred Value           0.9943     0.99175       0.9431      0.99549
Prevalence               0.2225     0.04046       0.6994      0.03757
Detection Rate           0.2182     0.03251       0.6814      0.03324
Detection Prevalence     0.2421     0.03613       0.6828      0.03902
Balanced Accuracy        0.9749     0.89990       0.9847      0.93930
```

- **The highest accuracy is 96.53 % on training dataset from the above confusion matrix.**
- **The accuracy on test dataset after submitting the predicted class values is 95.93%. Since the accuracy on training dataset is high but on testing dataset is low so it is not a good classifier to classify the cars. Hence it is not used for classifying.**

4.3. Random Forest:

Random forest is an ensemble learning method which is used for classification and regression that operates by constructing a multitude of decision trees at training time and giving output as the class i.e. mode of the class(classification).

For our experiments -

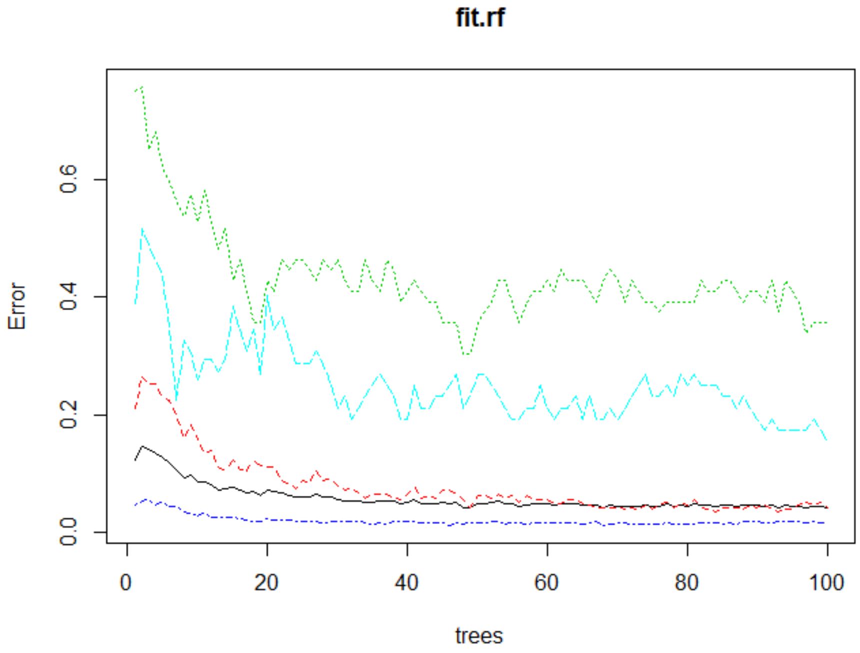
R package used: randomForest

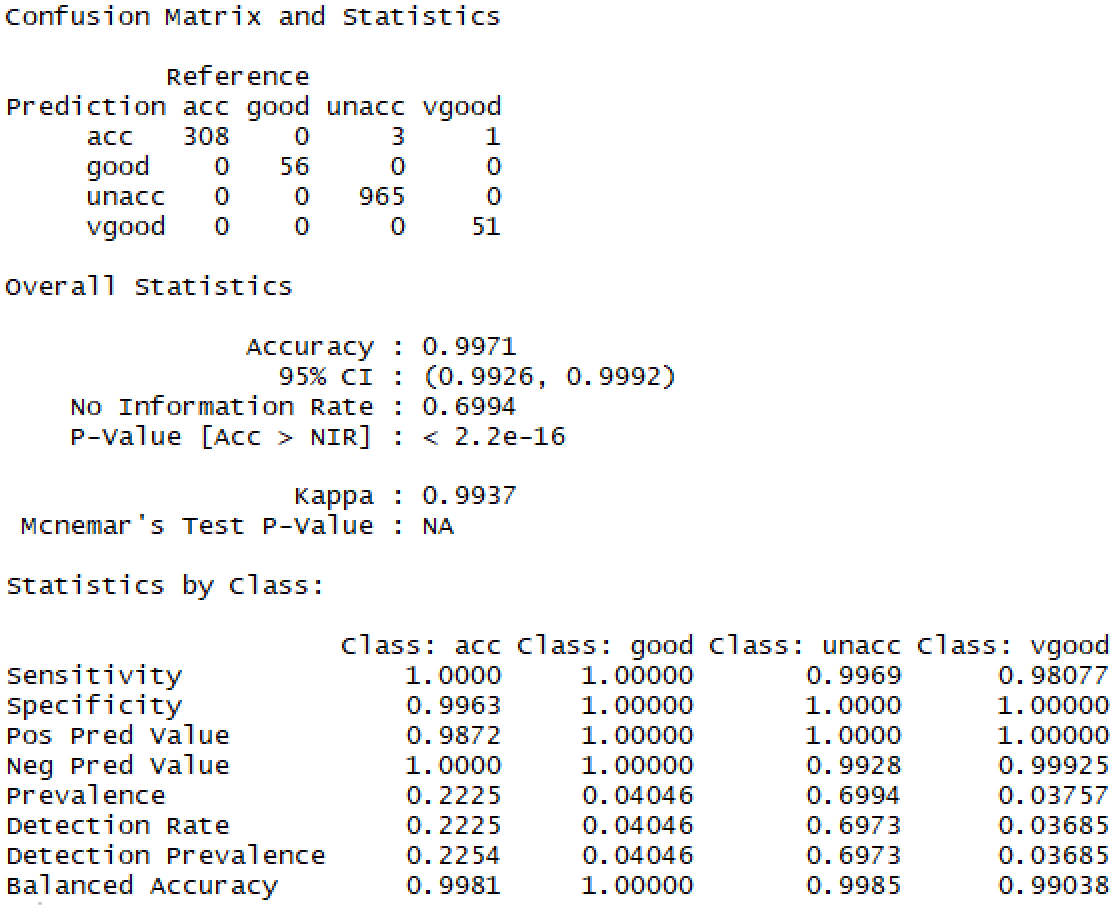
One of the advantages in Random Forest is that it can handle large number of features. Also, it is fast and can help quickly estimate which attributes are important.

The computations for our model is like that based on two parameters**.**

Accuracy Table:

Experiment #	Parameter(ntree)	Parameter(nodesize)	Mean Accuracy
1	100	2	0.7948
2	75	2	0.9957
3	50	2	0.9964
4	25	2	0.9913
5	100	4	0.9899
6	75	4	0.9921
7	50	4	0.9942
8	25	4	0.9877





```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   308    0     3     1
     good    0   56     0     0
     unacc   0    0   965     0
     vgood   0    0     0    51

Overall Statistics

               Accuracy : 0.9971
                 95% CI : (0.9926, 0.9992)
    No Information Rate : 0.6994
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9937
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              1.0000     1.00000       0.9969      0.98077
Specificity              0.9963     1.00000       1.0000      1.00000
Pos Pred Value           0.9872     1.00000       1.0000      1.00000
Neg Pred Value           1.0000     1.00000       0.9928      0.99925
Prevalence               0.2225     0.04046       0.6994      0.03757
Detection Rate           0.2225     0.04046       0.6973      0.03685
Detection Prevalence     0.2254     0.04046       0.6973      0.03685
Balanced Accuracy        0.9981     1.00000       0.9985      0.99038
```

- **The highest accuracy is 99.71 % on training dataset from the above confusion matrix.**
- **The accuracy on test dataset after submitting the predicted class values is 98.55%. Since the accuracy on training dataset is high but on testing dataset is low so it is not a good classifier to classify the cars. Hence it is not used for classifying.**

4.4. Gradient Boosting:

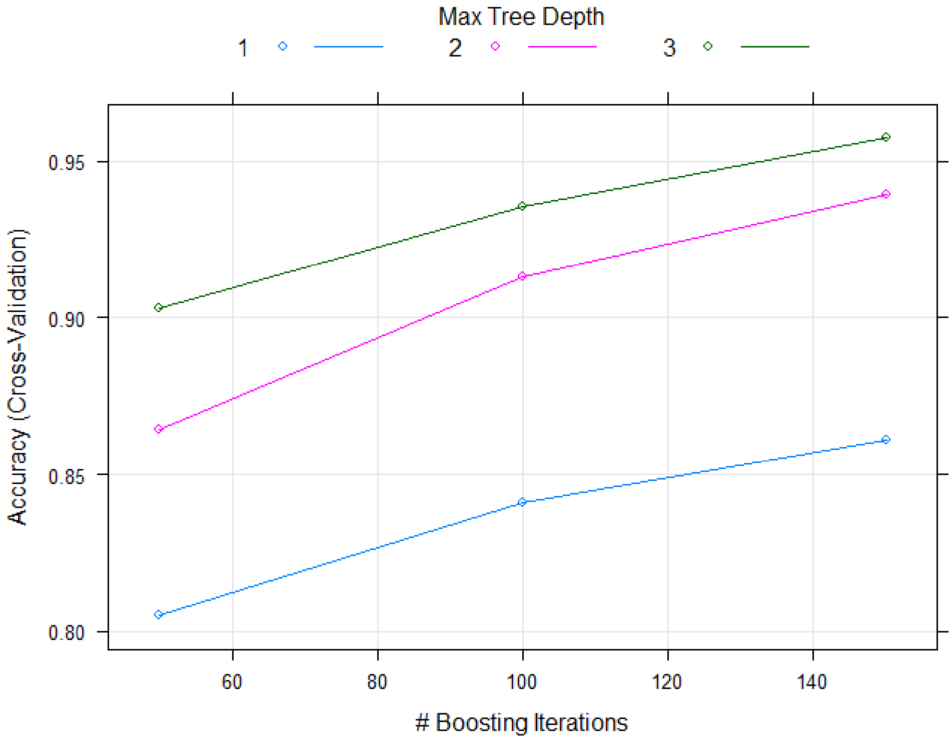
Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

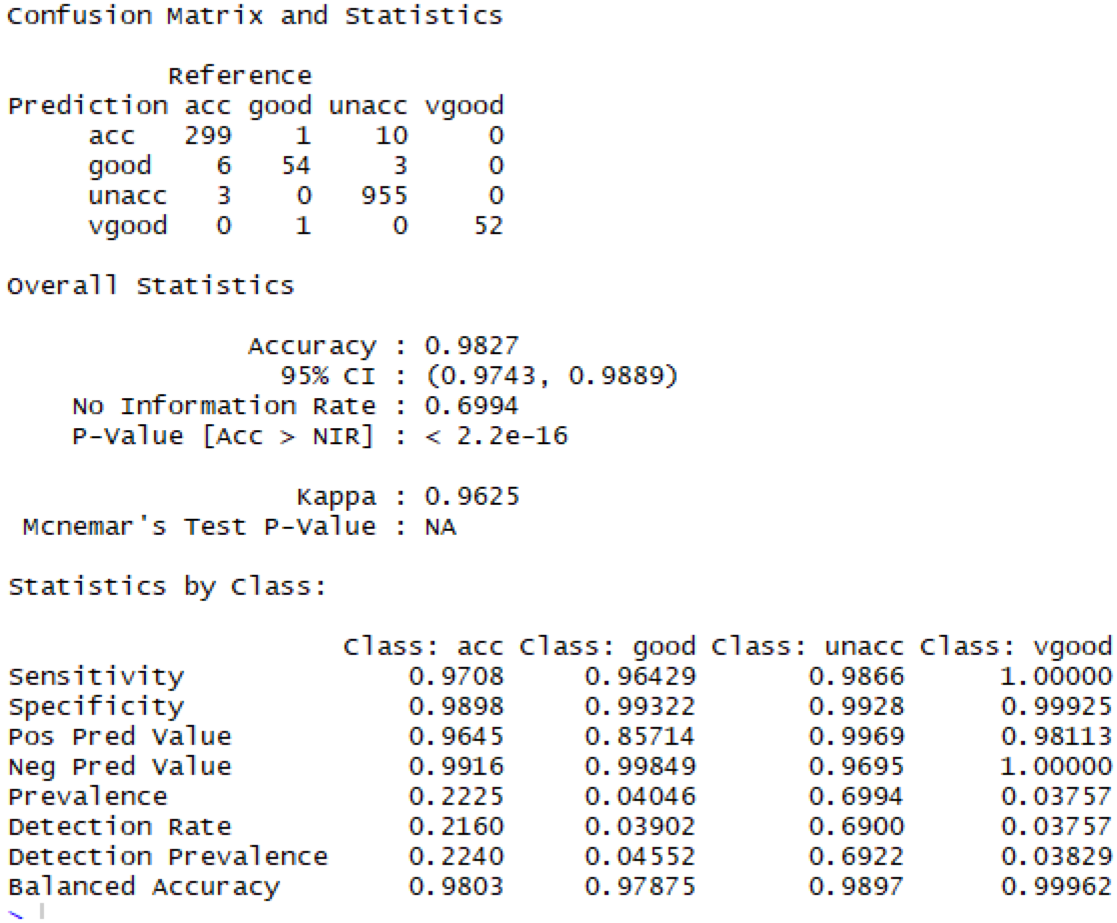
For our experiments -

R package used: Caret, gbm

Accuracy Table

Experiment #	Method	Cross-validation fold	Parameter Ntrees	Parameter interaction depth	Average Accuracy
1	GBM	10	50	1	0.8049
2	GBM	10	100	1	0.8410
3	GBM	10	50	2	0.8641
4	GBM	10	100	2	0.9132
5	GBM	10	50	3	0.9031
6	GBM	10	100	3	0.9356
7	GBM	10	150	3	0.9573





```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   299    1    10     0
     good    6   54     3     0
     unacc   3    0   955     0
     vgood   0    1     0    52

Overall Statistics

               Accuracy : 0.9827
                 95% CI : (0.9743, 0.9889)
    No Information Rate : 0.6994
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9625
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.9708     0.96429       0.9866      1.00000
Specificity              0.9898     0.99322       0.9928      0.99925
Pos Pred Value           0.9645     0.85714       0.9969      0.98113
Neg Pred Value           0.9916     0.99849       0.9695      1.00000
Prevalence               0.2225     0.04046       0.6994      0.03757
Detection Rate           0.2160     0.03902       0.6900      0.03757
Detection Prevalence     0.2240     0.04552       0.6922      0.03829
Balanced Accuracy        0.9803     0.97875       0.9897      0.99962
>
```

- **The highest accuracy is 98.27 % on training dataset from the above confusion matrix.**
- **The accuracy on test dataset after submitting the predicted class values is 96.8%. Since the accuracy on training dataset is high but on testing dataset is low so it is not a good classifier to classify the cars. Hence it is not used for classifying.**

4.5. Learning Vector Quantization:

LVQ is a prototype based supervised classification algorithm. LVQ is the supervised counterpart of vector quantization system.

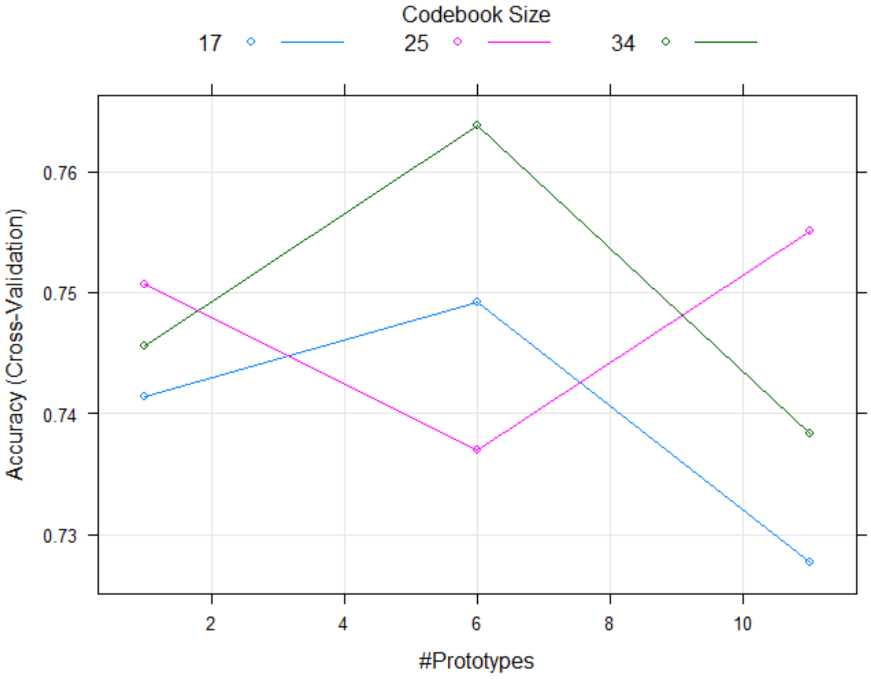
LVQ algorithm is represented by a prototype W=(w(i),...,w(n)) that are defined in the feature space of observed data. It creates prototypes that are easy to interpret in the respective application domain. LVQ systems can applied to multi-class classification problems.

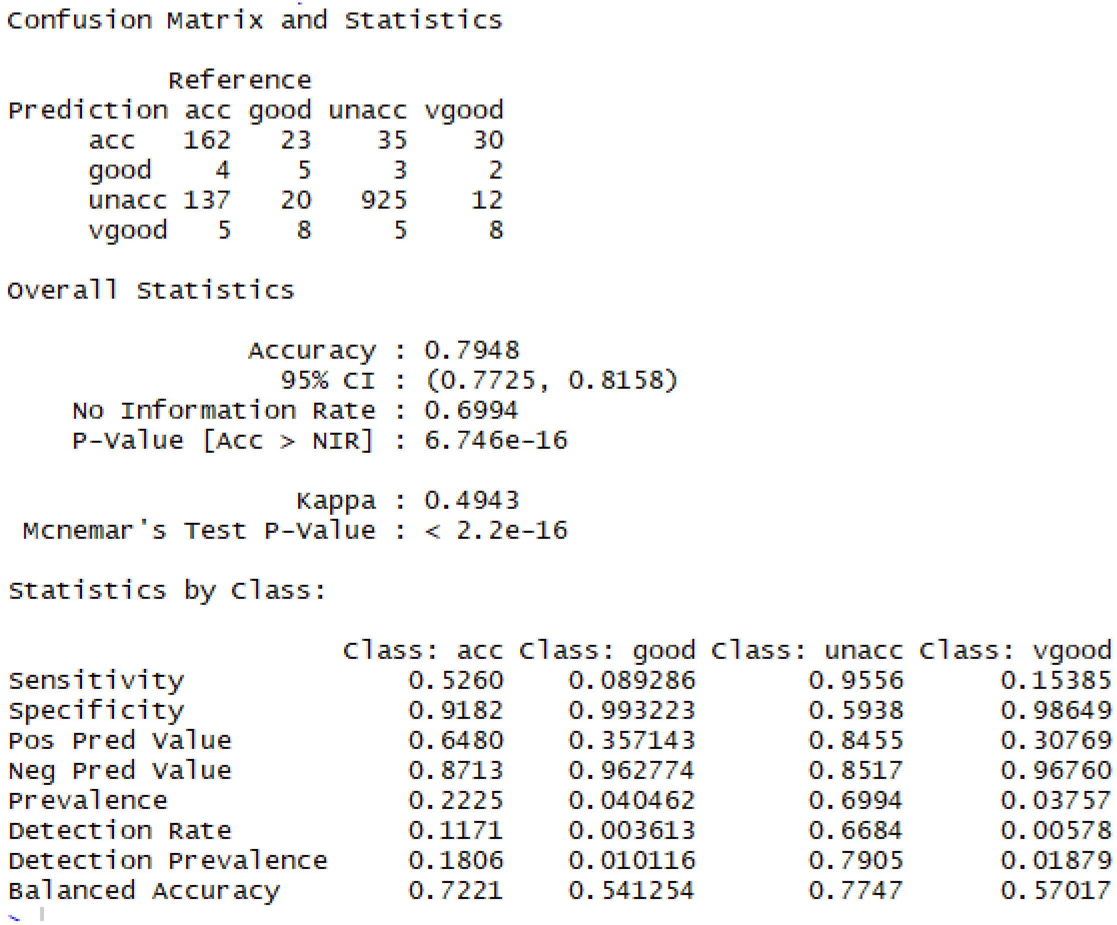
For our experiments -

R package used: Caret, class

Accuracy Table

Experiment #	Method	Cross-validation fold	Parameter Size	Parameter interaction depth	Average Accuracy
1	GBM	10	17	1	0.7413
2	GBM	10	25	1	0.7507
3	GBM	10	34	1	0.7455
4	GBM	10	17	6	0.7491
5	GBM	10	25	6	0.7369
6	GBM	10	34	6	0.7637





```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   162   23    35    30
     good    4    5     3     2
     unacc 137   20   925    12
     vgood   5    8     5     8

Overall Statistics

               Accuracy : 0.7948
                 95% CI : (0.7725, 0.8158)
    No Information Rate : 0.6994
    P-Value [Acc > NIR] : 6.746e-16

                  Kappa : 0.4943
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.5260    0.089286       0.9556      0.15385
Specificity              0.9182    0.993223       0.5938      0.98649
Pos Pred Value           0.6480    0.357143       0.8455      0.30769
Neg Pred Value           0.8713    0.962774       0.8517      0.96760
Prevalence               0.2225    0.040462       0.6994      0.03757
Detection Rate           0.1171    0.003613       0.6684      0.00578
Detection Prevalence     0.1806    0.010116       0.7905      0.01879
Balanced Accuracy        0.7221    0.541254       0.7747      0.57017
```

- **The highest accuracy is 79.48 % on training dataset from the above confusion matrix.**
- **The accuracy on test dataset after submitting the predicted class values is 75.58%. Since the accuracy on training dataset is high but on testing dataset is low so it is not a good classifier to classify the cars. Hence it is not used for classifying.**

5. Conclusion:

Based on the proposed solutions and experimental methodology and analysis, we found out that the best classifier which can be used to classify each instance of car is **Random Forest.**
The highest accuracy is 99.71 % on training dataset and 98.55% on testing dataset.

Hence, **RANDOM FOREST** is the best classifier which can be used to find the acceptability of the cars based on given six attributes.

6. References:

- https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
- https://www.r-bloggers.com/
- http://machinelearningmastery.com/
- https://www.wikipedia.org/
- https://www.datacamp.com/
- http://www.statmethods.net/