CAR EVALUATION

INTRODUCTION: Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

Car acceptability
. PRICE overall price
. . buying price
. . price of the maintenance
. TECH technical characteristics
. . comfort
. . . number of doors
. . . capacity in terms of persons to carry
. . . the size of luggage boot
. . estimated safety of the car

Attribute Information:

Class Values (Car acceptability) : unacc, acc, good, vgood

Attributes:
buying: vhigh, high, med, low.
maint: vhigh, high, med, low.
doors: 2, 3, 4, 5more.
persons: 2, 4, more.
lug_boot: small, med, big.
safety: low, med, high.

Abstract: To model a classifier for evaluating the acceptability of car using its given features

---

IMPORT DATASET

```
car <- read.csv("c:/data/car.csv")
```

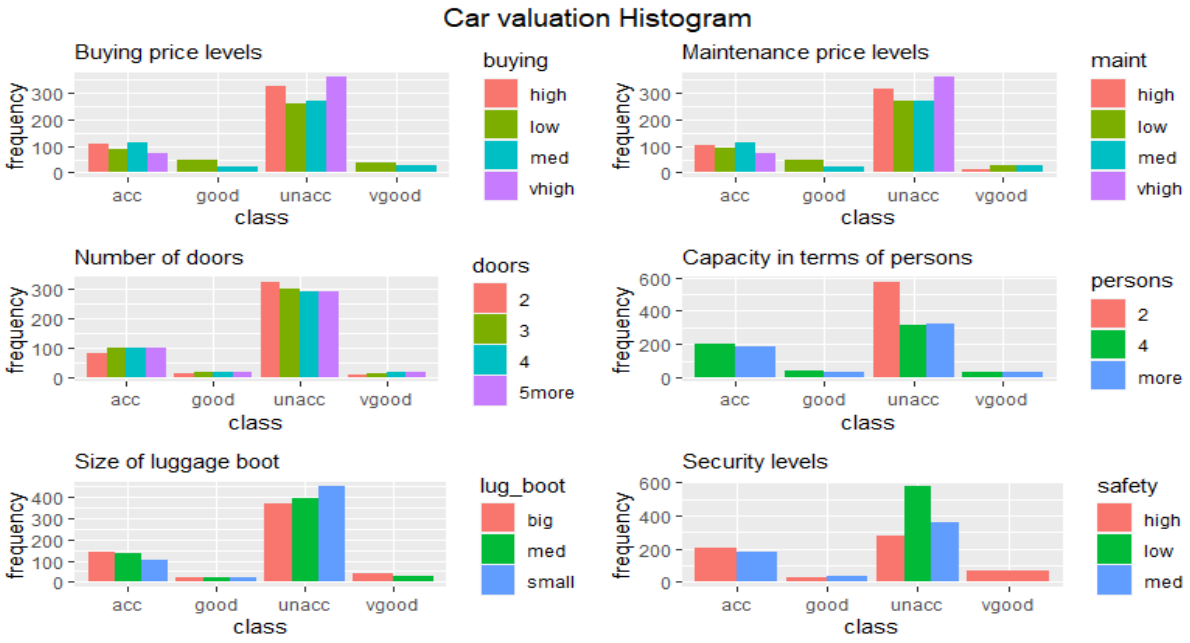
DATA EXPLORATION

```
View(car)
str(car)
summary(car)
head(car)
```

with these functions, we get a summary of the data.

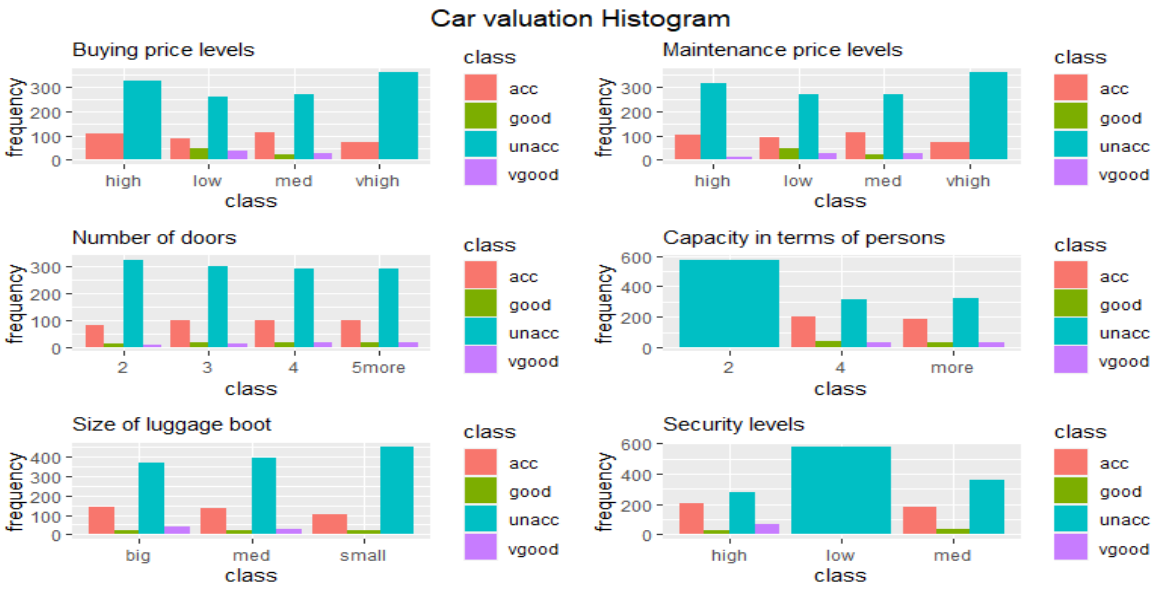
```
> str(car)
'data.frame':   1728 obs. of  7 variables:
 $ buying  : chr  "vhigh" "vhigh" "vhigh" "vhigh" ...
 $ maint   : chr  "vhigh" "vhigh" "vhigh" "vhigh" ...
 $ doors   : chr  "2" "2" "2" "2" ...
 $ persons : chr  "2" "2" "2" "2" ...
 $ lug_boot: chr  "small" "small" "small" "med" ...
 $ safety  : chr  "low" "med" "high" "low" ...
 $ class   : chr  "unacc" "unacc" "unacc" "unacc" ...
> summary(car)
    buying             maint              doors             persons           lug_boot
 Length:1728        Length:1728        Length:1728        Length:1728        Length:1728
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
    safety             class
 Length:1728        Length:1728
 Class :character   Class :character
 Mode  :character   Mode  :character
> head(car)
  buying maint doors persons lug_boot safety class
1  vhigh vhigh     2       2    small    low unacc
2  vhigh vhigh     2       2    small    med unacc
3  vhigh vhigh     2       2    small   high unacc
4  vhigh vhigh     2       2      med    low unacc
5  vhigh vhigh     2       2      med    med unacc
6  vhigh vhigh     2       2      med   high unacc
```

DATA VISUALIZATION



Based on this graph, security is a key benchmark for vehicle acceptance. High safety led to very good acceptability.

Another view of dataset,



This graph also shows that vehicles with less safety are not preferred.

DATA ANALYTIC TECHNIQUES

1. **DECISION TREE**

First step is to split data into separate test and train set

```
sample_split = sample.split(car, SplitRatio = 0.50)
train_split <- subset(car, sample_split == TRUE)
test_split  <- subset(car, sample_split == FALSE)
```

Rpart and rpart.plot packages are used for decision tree.

```
car_decision  <-  rpart(formula = class ~.,
                  data = train_split,
                  method = "class",
                  control = rpart.control(cp = 0),
                  parms = list(split = "information"))
```

Next step is to predict on test data and find accuracy using confusionMatrix. “caret” package is used for doing confusionMatrix

```
car_prediction <- predict(object = car_decision,
                  newdata = test_split,
                  type = "class")

confusionMatrix(data = as.factor(car_prediction),
             reference = as.factor(test_split$class))
```

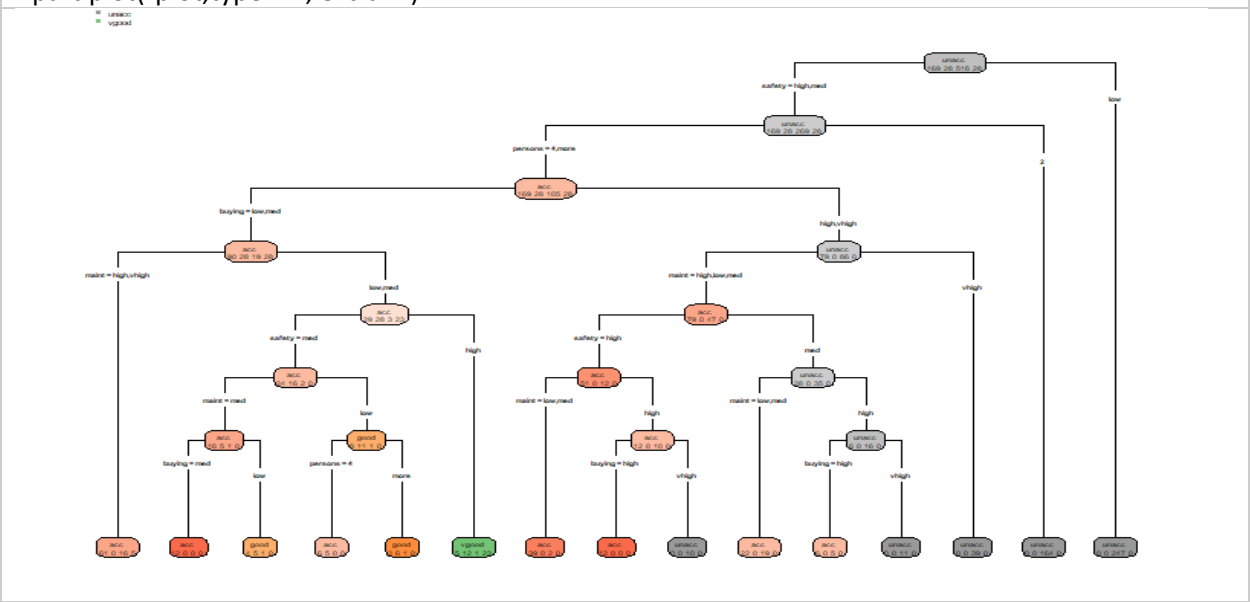
Result of Confusion Matrix :

```
> confusionMatrix(data = as.factor(car_prediction),
+                 reference = as.factor(test_split$class))
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   201    8    22     8
     good    7   28     3     0
     unacc   5    0   669     0
     vgood   2    5     0    29

Overall Statistics

               Accuracy : 0.9392
                 95% CI : (0.9224, 0.9533)
    No Information Rate : 0.7031
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.869

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.9349     0.68293       0.9640      0.78378
Specificity              0.9508     0.98943       0.9829      0.99263
Pos Pred Value           0.8410     0.73684       0.9926      0.80556
Neg Pred Value           0.9813     0.98630       0.9201      0.99159
Prevalence               0.2178     0.04154       0.7031      0.03749
Detection Rate           0.2036     0.02837       0.6778      0.02938
Detection Prevalence     0.2421     0.03850       0.6829      0.03647
Balanced Accuracy        0.9428     0.83618       0.9735      0.88821
```

The accuracy shown above is 0.9392 which means decision tree will able to predict car acceptability with an accuracy of **93.92%**.

Plotting the decision tree.

```
rplot <- rpart(class~ buying + maint + persons +  safety , method = "class",
         data = train_split,
         control = rpart.control(cp = 0),
         parms = list(split="information"))
 rpart.plot(rplot,type= 4 , extra=1)
```



---

2. RANDOM FOREST

Convert dependent variable which are character in class to factor.

```
#character to factor
 value <-  factor(car$class,
           levels = c('acc', 'good', 'unacc','vgood'),
           labels = c(1, 2, 3,4))
```

For applying random forest , 'randomForest' package is used.

```
car_random<- randomForest(value ~ buying + maint + doors + persons + safety + lug_boot,
                  data = car )
print(car_random)
print (importance(car_random,type = 2))
```

```
> print(car_random)

Call:
 randomForest(formula = value ~ buying + maint + doors + persons +      safety + lug_boot, data = car)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 3.3%
Confusion matrix:
    1  2    3  4 class.error
1 371  0   13  0  0.03385417
2  24 42    0  3  0.39130435
3  13  0 1197  0  0.01074380
4   4  0    0 61  0.06153846
> print (importance(car_random,type = 2))
         MeanDecreaseGini
buying           99.07751
maint            81.82396
doors            26.10804
persons         158.64339
safety          200.77024
lug_boot         48.40670
```

In the above table, the error rate is 3.3% which turns to a good accuracy and the MeanDecreaseGini values shows that the least required model is count of doors. So, we can remove doors and rebuild the model.

```
car_randomForest<- randomForest(value~ buying + maint + persons + safety + lug_boot,
                  data = car )
print(car_randomForest)
```

```
> print(car_randomForest)

Call:
 randomForest(formula = value ~ buying + maint + persons + safety +      lug_boot, data = car)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 5.21%
Confusion matrix:
    1  2    3  4 class.error
1 363  3   14  4   0.0546875
2  12 48    0  9   0.3043478
3  38  3 1169  0   0.0338843
4   7  0    0 58   0.1076923
```

Now the error rate is 5.21% .

Predict on test data and find the accuracy

```
test_split$class <-  factor(test_split$class,
            levels = c('acc', 'good', 'unacc','vgood'),
            labels = c(1, 2, 3,4))

prediction_random <- predict(car_randomForest, newdata = test_split)
confusionMatrix(data = as.factor(prediction_random),
          reference = as.factor(test_split$class))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3   4
         1 212   7  18   1
         2   1  27   1   0
         3   4   0 673   0
         4   2   5   0  37

Overall Statistics

               Accuracy : 0.9605
                 95% CI : (0.9464, 0.9718)
    No Information Rate : 0.7004
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9149

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.9680  0.69231   0.9725  0.97368
Specificity            0.9662  0.99789   0.9865  0.99263
Pos Pred Value         0.8908  0.93103   0.9941  0.84091
Neg Pred Value         0.9907  0.98749   0.9389  0.99894
Prevalence             0.2217  0.03947   0.7004  0.03846
Detection Rate         0.2146  0.02733   0.6812  0.03745
Detection Prevalence   0.2409  0.02935   0.6852  0.04453
Balanced Accuracy      0.9671  0.84510   0.9795  0.98316
```

The accuracy shown above is 0.9605 which means Random forest will able to predict car acceptability with an accuracy of **96%**.

---

3. NAIVE BAYES

For Naïve Bayes model , ‘e1071’ package is used. Prediction is done on test data and confusionMatrix () is used for finding accuracy on prediction.

```
classifier_car <- naiveBayes(class ~ ., data = train_split)

# Predicting on test data
prediction <- predict(classifier_car, newdata = test_split)

# Confusion Matrix
confusionMatrix(data = as.factor(prediction),
          reference = as.factor(test_split$class))
```

```
> confusionMatrix(data = as.factor(prediction),
+                 reference = as.factor(test_split$class))
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc   174   26    30    17
     good    5   13     2     0
     unacc  36    0   662     0
     vgood   0    2     0    20

Overall Statistics

               Accuracy : 0.8804
                 95% CI : (0.8586, 0.9)
    No Information Rate : 0.7031
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7323

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.8093     0.31707       0.9539      0.54054
Specificity              0.9054     0.99260       0.8771      0.99789
Pos Pred Value           0.7045     0.65000       0.9484      0.90909
Neg Pred Value           0.9446     0.97104       0.8893      0.98238
Prevalence               0.2178     0.04154       0.7031      0.03749
Detection Rate           0.1763     0.01317       0.6707      0.02026
Detection Prevalence     0.2503     0.02026       0.7072      0.02229
Balanced Accuracy        0.8574     0.65484       0.9155      0.76922
```

The accuracy shown above is 0.8804 which means Naïve bayes will able to predict car acceptability with an accuracy of **88%**.

REPORT

In this project, Decision tree, Random Forest, and Naïve bayes of Classification techniques are used to predict and confusion matrix for finding accuracy. Here decision tree shows **93.92%** of accuracy, random forest shows **96%** of accuracy and finally Naïve Bayes shows **88%** of accuracy. Overall Random Forest is having higher accuracy.
