COURSERA – FINAL PROJECT

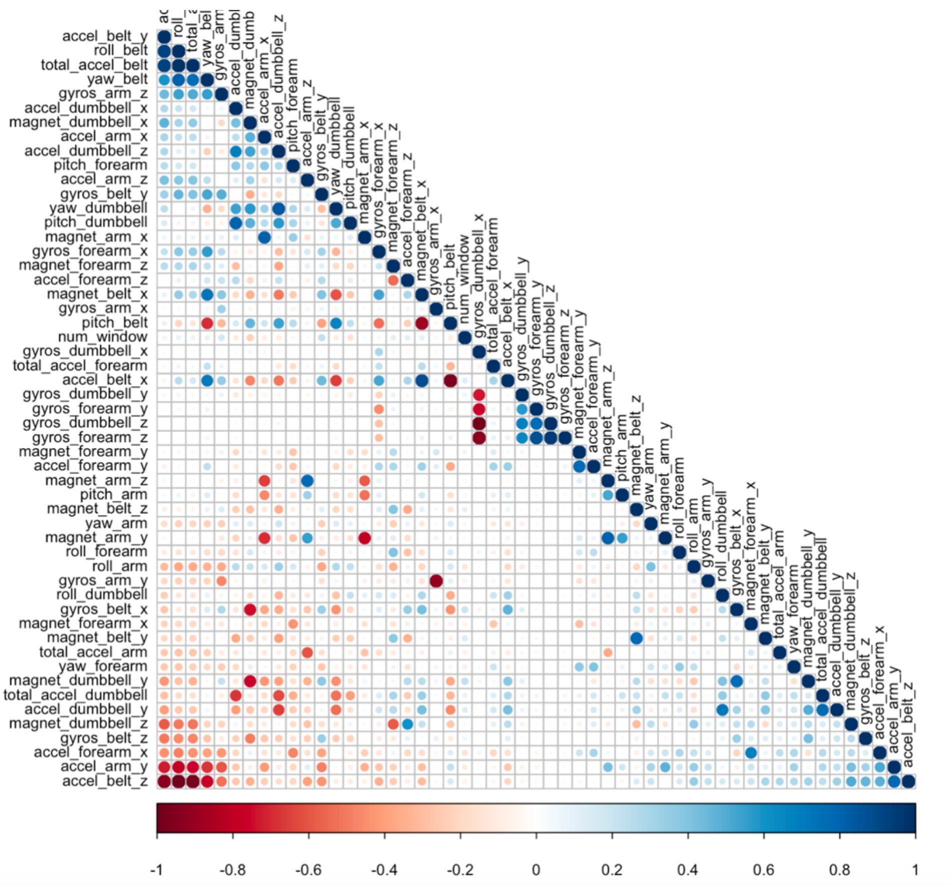
ENVIRONMENT PREPARATION - Downloading necessary libraries for the programming

```
# Set the working directory
setwd("/Users/alexandralocchi/Documents/EDHEC NICE/Cours/FE")

library(lattice)
library(ggplot2)
library(caret)
library(rpart)
library(rpart.plot)
library(corrplot)
library(rattle)
library(randomForest)
library(RColorBrewer)
set.seed(1012)
```

DATA LOADING & CLEANSING

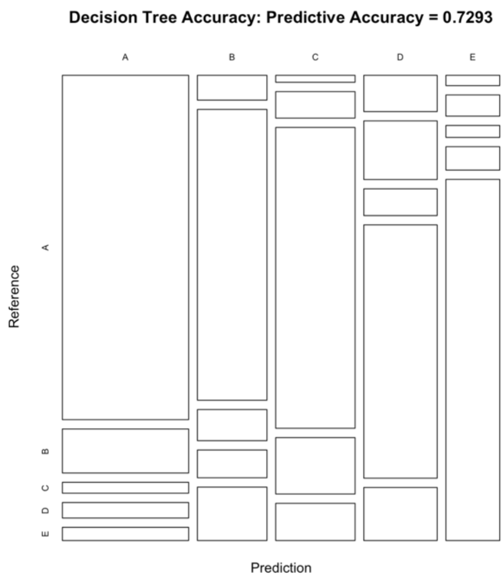
```
# Set URL for the download
UrlTrain <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
UrlTest  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

# Download adequate datasets
training <- read.csv(url(UrlTrain))
testing  <- read.csv(url(UrlTest))

# Create a partition with the training dataset (70% of the data) for the modeling process and
a TestSet with 30% of remaining data for validations
inTrain  <- createDataPartition(training$classe, p=0.7, list=FALSE)
TrainSet <- training[inTrain, ]
TestSet  <- training[-inTrain, ]
dim(TrainSet)
## [1] 13737   160
dim(TestSet)
## [1] 5885  160

#The two datasets (TrainSet & TestSet) have 160 variables and  multiple NA numbers and
near zero variance variables. I will remove both.
nzv_var <- nearZeroVar(TrainSet)
TrainSet <- TrainSet[, -nzv_var]
TestSet <- TestSet[, -nzv_var]
dim(TrainSet)
## [1] 13737    104
dim(TestSet)
## [1] 5885    104
```

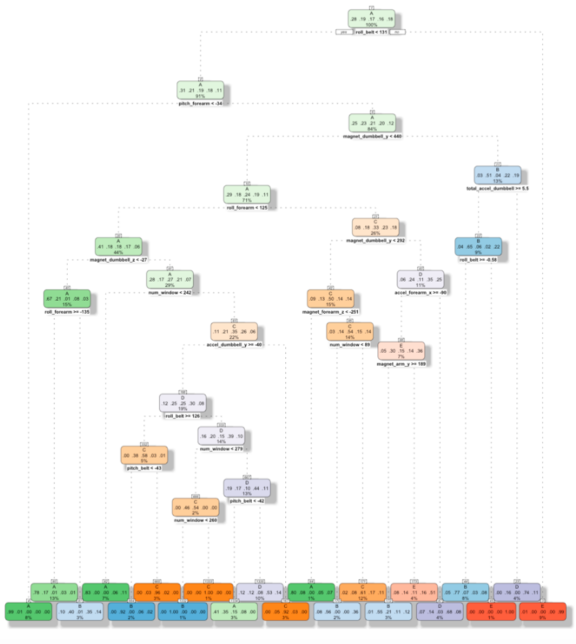
```
na_var <- sapply(TrainSet, function(x) mean(is.na(x))) > 0.95
TrainSet <- TrainSet[ , na_var == FALSE]
TestSet <- TestSet [ , na_var == FALSE]
dim(TrainSet)
## [1] 13737    59
dim(TestSet)
## [1] 5885   59

# I will also remove columns 1 to 5 because they are identification variations
TrainSet <- TrainSet[ , -(1:5)]
TestSet  <- TestSet [ , -(1:5)]
dim(TrainSet)
## [1] 13737    54
dim(TestSet)
## [1] 5885   54

# ➔ With the cleaning of the data, only 54 variables remain.

# CORRELATION ANALYSIS
corr_matrix <- cor(TrainSet[ , -54])
corrplot(corr_matrix, order = "FPC", method = "circle", type = "lower",
        tl.cex = 0.8, tl.col = rgb(0, 0, 0))
```




```
# Variables showing high level of correlation appear in darker colors in the previous plot;
they are dark blue for a positive correlation or dark red for a negative correlation?
```

```r
# PREDICTION MODEL BUILDING

  # DECISION TREE MODEL
set.seed(1012)
fit_dec_tree <- rpart(classe ~ ., data = TrainSet, method="class")
fancyRpartPlot(fit_dec_tree)
# Prediction of the decision tree model on TestSet
predict_dec_tree <- predict(fit_dec_tree, newdata = TestSet, type="class")
conf_matrix_dec_tree <- confusionMatrix(predict_dec_tree, TestSet$classe)
conf_matrix_dec_tree
# Ploting the matrix results of the decision tree model
plot(conf_matrix_dec_tree$table, col = conf_matrix_dec_tree$byClass,
    main = paste("Decision Tree Accuracy: Predictive Accuracy =",
            round(conf_matrix_dec_tree$overall['Accuracy'], 4)))
```

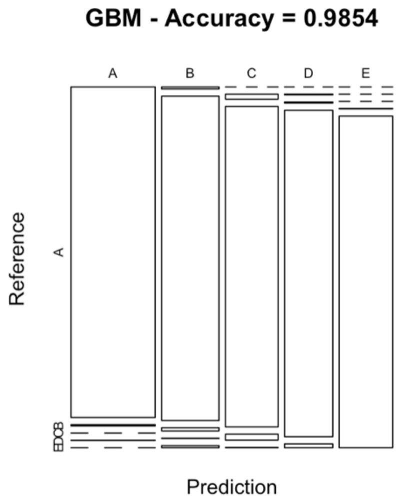




```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1486  190   47   67   57
         B   59  691   74   66  127
         C   19   72  816  153  101
         D   91  147   67  635  133
         E   19   39   22   43  664

Overall Statistics

               Accuracy : 0.7293
                 95% CI : (0.7178, 0.7406)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6567

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

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.8877   0.6067   0.7953   0.6587   0.6137
Specificity            0.9143   0.9313   0.9290   0.9110   0.9744
Pos Pred Value         0.8045   0.6794   0.7028   0.5918   0.8437
Neg Pred Value         0.9534   0.9080   0.9555   0.9316   0.9180
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2525   0.1174   0.1387   0.1079   0.1128
Detection Prevalence   0.3138   0.1728   0.1973   0.1823   0.1337
Balanced Accuracy      0.9010   0.7690   0.8622   0.7849   0.7940
```

```
# The predictive accuracy of the decision tree model is relatively low at 72.93%.

# GENERALIZED BOOSTED MODEL
set.seed(1012)
ctrl_GBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
fit_GBM  <- train(classe ~ ., data = TrainSet, method = "gbm",
                  trControl = ctrl_GBM, verbose = FALSE)
fit_GBM$finalModel
# Prediction of the decision tree model on TestSet
predict_GBM <- predict(fit_GBM, newdata=TestSet)
conf_matrix_GBM <- confusionMatrix(predict_GBM, TestSet$classe)
conf_matrix_GBM
# Ploting the matrix results of the GBM model
plot(conf_matrix_GBM$table, col = conf_matrix_GBM$byClass,
     main = paste("GBM - Accuracy =", round(conf_matrix_GBM$overall['Accuracy'], 4)))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1667    6    0    1    0
         B    7 1115   11    1    7
         C    0   16 1012   19    1
         D    0    2    3  942   11
         E    0    0    0    1 1063

Overall Statistics

               Accuracy : 0.9854
                 95% CI : (0.982, 0.9883)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9815

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9958   0.9789   0.9864   0.9772   0.9824
Specificity            0.9983   0.9945   0.9926   0.9967   0.9998
Pos Pred Value         0.9958   0.9772   0.9656   0.9833   0.9991
Neg Pred Value         0.9983   0.9949   0.9971   0.9955   0.9961
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2833   0.1895   0.1720   0.1601   0.1806
Detection Prevalence   0.2845   0.1939   0.1781   0.1628   0.1808
Balanced Accuracy      0.9971   0.9867   0.9895   0.9870   0.9911
```

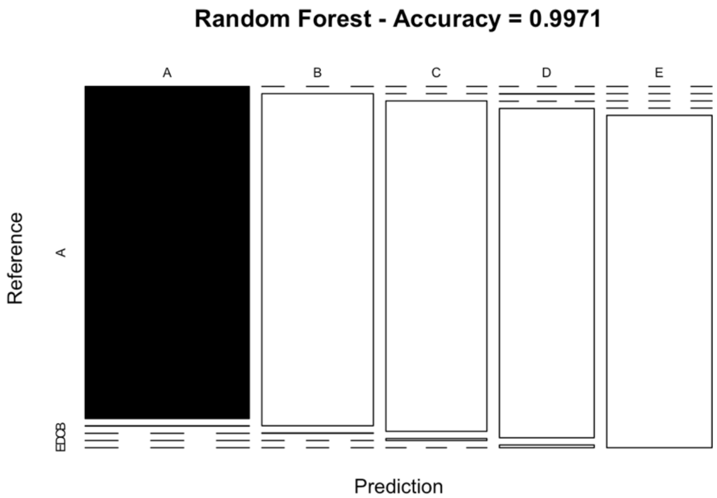


The predictive accuracy of the GBM model is relatively high at 98.54%.

 # RANDOM FOREST MODEL

```
set.seed(1012)
ctrl_RF <- trainControl(method="cv", number=3, verboseIter=FALSE)
fit_RF <- train(classe ~ ., data=TrainSet, method="rf",
                trControl=ctrl_RF)
fit_RF$finalModel
# Prediction of the decision tree model on TestSet
predict_RF <- predict(fit_RF, newdata = TestSet)
conf_matrix_RF <- confusionMatrix(predict_RF, TestSet$classe)
conf_matrix_RF
# Ploting the matrix results of the random forest model
plot(conf_matrix_RF$table, col = conf_matrix_RF$byClass,
```

main = paste("Random Forest: Predictive Accuracy =",
round(conf_matrix_dec_tree$overall['Accuracy'], 4)))




```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1674    1    0    0    0
         B    0 1137    1    0    0
         C    0    0 1025    6    0
         D    0    1    0  958    8
         E    0    0    0    0 1074

Overall Statistics

               Accuracy : 0.9971
                 95% CI : (0.9954, 0.9983)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9963

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9982   0.9990   0.9938   0.9926
Specificity            0.9998   0.9998   0.9988   0.9982   1.0000
Pos Pred Value         0.9994   0.9991   0.9942   0.9907   1.0000
Neg Pred Value         1.0000   0.9996   0.9998   0.9988   0.9983
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2845   0.1932   0.1742   0.1628   0.1825
Detection Prevalence   0.2846   0.1934   0.1752   0.1643   0.1825
Balanced Accuracy      0.9999   0.9990   0.9989   0.9960   0.9963
```

The predictive accuracy of the RF model is the higher amongst all methodologies, at 99.71%.

 # CCL - APPLY THE SELECTED MODEL TO THE DATA TEST

predict_Test <- predict(fit_RF, newdata=testing)
predict_Test
[1] B A B A A E D B A A B C B A E E A B B B
Levels: A B C D E

➔ To conclude, the predictive accuracy of the models are : 72.93% for the Decision Tree model ; 98.54% for the Generalized Boosted model ; and 99.71% for the Random Forest model.

We can select the RF model to predict the 20 quiz results.