Final Assignment for Practical Machine Learning course

Load all relevant libraries and data files

```
## Load all relevant packages


library(caret)
library(rpart)  ## contains the classifier we will be using
library(rpart.plot)
library(RColorBrewer)
library(rattle)   ## contains functions for graphical plotting that we will be using in the application.

## Download data files and save locally. Set relevant working directory in R
## Load Test and Train files provided for the assignment.
## Remove features or columns with sparse data from both files. They won't add to the quality of the model,
## they may skew the results or increase computational complexity unnecessarily.


TrainingFile <-  read.csv("MLQuizTraining.csv", na.strings=c("NA","#DIV/0!",""))
TrainingFile <- TrainingFile[,colSums(is.na(TrainingFile)) == 0]
TestingFile <- read.csv("MLQuizTesting.csv", na.strings=c("NA","#DIV/0!",""))
TestingFile <- TestingFile[,colSums(is.na(TestingFile)) == 0]
```

Split Training file into Training and Validation data sets

```
## Split the training file  provided for the exercise into training and validation sets
## Seed and then randomly assign 70% to the training set and 30% to the validation set
## The 20 samples in the testing file will be used separatey as a standalone final testing set

set.seed(787878)
inTrain <- createDataPartition(y=TrainingFile$classe, p=0.7, list=FALSE)
TrainingSet <- TrainingFile[inTrain, ]
ValidationSet <- TrainingFile[-inTrain, ]
TestingSet <- TestingFile
dim(TrainingSet); dim(ValidationSet); dim(TestingSet)
```

```
[1] 13737    60
[1] 5885   60
[1] 20 60
```

Remove unnecessary columns that are not reflective of features for the model building exercise. This will also prevent overfitting. Remove first 7 columns for all 3 data sets (Training, Validation and Testing)

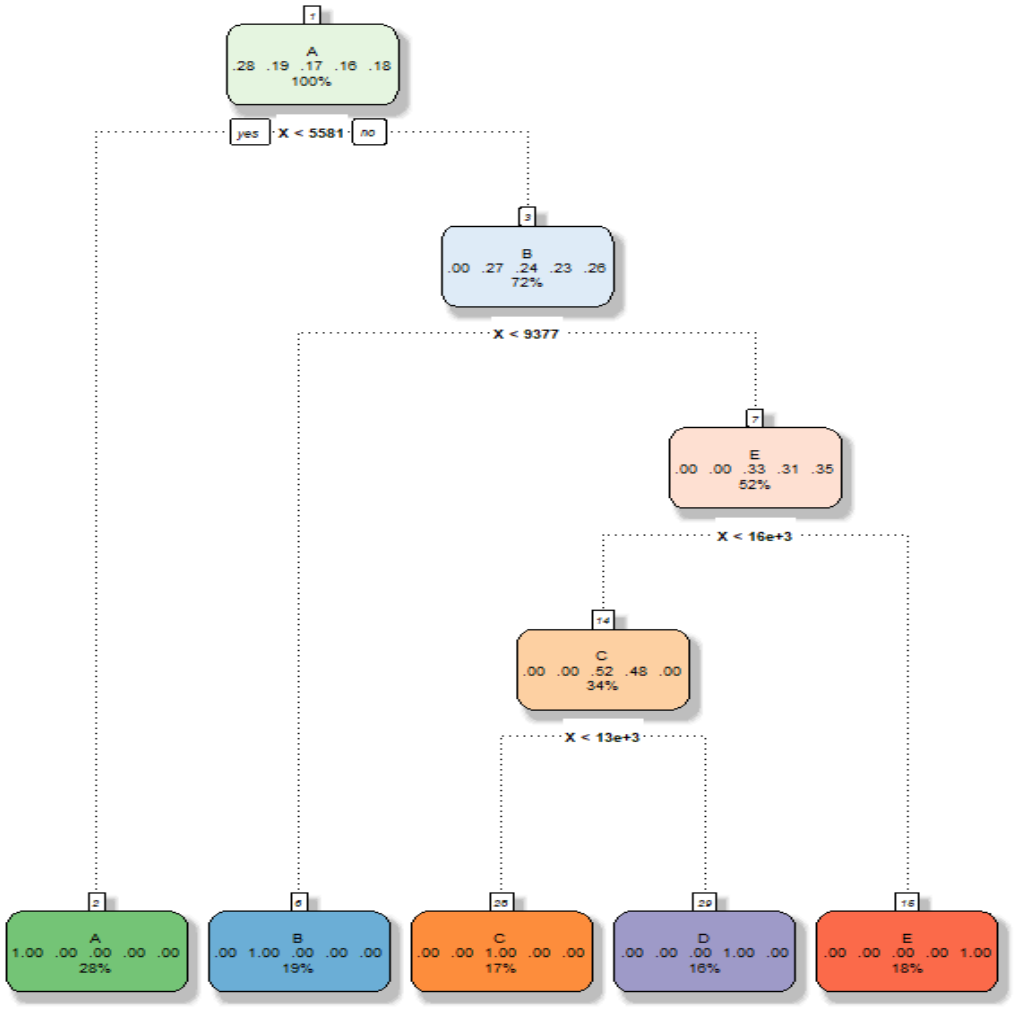
```
> str(TrainingSet)
'data.frame':   13737 obs. of  60 variables:
 $ X                    : int  1 2 6 8 9 11 12 15 17 19 ...
 $ user_name            : Factor w/ 6 levels "adelmo","carlitos",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ raw_timestamp_part_1: int  1323084231 1323084231 1323084232 1323084232 1323084232 1323084232 13230
 $ raw_timestamp_part_2: int  788290 808298 304277 440390 484323 500302 528316 604281 692324 740353 .
 $ cvtd_timestamp       : Factor w/ 20 levels "2/12/2011 13:32",..: 15 15 15 15 15 15 15 15 15 15 ...
 $ new_window           : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ num_window           : int  11 11 12 12 12 12 12 12 12 12 ...
 $ roll_belt            : num  1.41 1.41 1.45 1.42 1.43 1.45 1.43 1.45 1.51 1.57 ...
 $ pitch_belt           : num  8.07 8.07 8.06 8.13 8.16 8.18 8.18 8.2 8.12 8.06 ...
 $ yaw_belt             : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...
 $ total_accel_belt     : int  3 3 3 3 3 3 3 3 3 3 ...
 $ gyros_belt_x         : num  0 0.02 0.02 0.02 0.02 0.03 0.02 0 0 0 ...
 $ gyros_belt_y         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ gyros_belt_z         : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 0 -0.02 -0.02 ...
 $ accel_belt_x         : int  -21 -22 -21 -22 -20 -21 -22 -21 -21 -20 ...
 $ accel_belt_y         : int  4 4 4 4 2 2 2 2 4 5 ...
 $ accel_belt_z         : int  22 22 21 21 24 23 23 22 22 21 ...
 $ magnet_belt_x        : int  -3 -7 0 -2 1 -5 -2 -1 -6 -3 ...
 $ magnet_belt_y        : int  599 608 603 603 602 596 602 597 598 603 ...
 $ magnet_belt_z        : int  -313 -311 -312 -313 -312 -317 -319 -310 -317 -313 ...
```

```
## conduct exploratory analysis of the data.
## There are several columns that don't appeat to be relevant to the model building.
## The first column is the sample index, followed by user name, date and time stamps, etc.
## Remove columns 1-7 as they don't contribute to the model from the Training Set, Validation set and Testing set

TrainingSet   <-TrainingSet[,-c(1:7)]
ValidationSetcolnames <- colnames(TrainingSet)
ValidationSet <- ValidationSet[ValidationSetcolnames]
TestingSet   <-TestingSet[,-c(1:7)]
dim(TrainingSet) ; dim(ValidationSet); dim(TestingSet)
```

```
[1] 13737    53
[1] 5885   53
[1] 20 53
```

Build model and plot outcome

```
## Build model for output classe againt all features in the data set
modFitPass1 <- rpart(classe ~ ., data=TrainingSet, method="class")

## view tree
fancyRpartPlot(modFitPass1)
```



Clean tree structure with A-28%, B-19%, C-17%, D-16%, E18%

Given a strong model fit, with steps being taken to prevent overfitting this model is expected to perform well against the validation data set.

Run Prediction against validation set and analyze outcome via ConfusionMatrix

```
## Predict using validation set created from training file for the assignment
predictionsPass1 <- predict(modFitPass1, ValidationSet, type = "class")

## Confusion Matrix
confusionMatrix(predictionsPass1, ValidationSet$classe)
```

Very high accuracy outcome of the validation set right off the bat with only 2 items out of 5885 cases wrongly classified. No further models need to be test. Performance against any test set expected to be good.

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1674    0    0    0    0
         B    0 1138    0    0    0
         C    0    1 1026    0    0
         D    0    0    0  964    1
         E    0    0    0    0 1081

Overall Statistics

               Accuracy : 0.9997
                 95% CI : (0.9988, 1)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9996
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9991   1.0000   1.0000   0.9991
Specificity            1.0000   1.0000   0.9998   0.9998   1.0000
Pos Pred Value         1.0000   1.0000   0.9990   0.9990   1.0000
Neg Pred Value         1.0000   0.9998   1.0000   1.0000   0.9998
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2845   0.1934   0.1743   0.1638   0.1837
Detection Prevalence   0.2845   0.1934   0.1745   0.1640   0.1837
Balanced Accuracy      1.0000   0.9996   0.9999   0.9999   0.9995
```

Final Prediction

```
> print(predictPass2 <- predict(modFitPass1, TestingSet,type="class"))
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 B  A  B  D  A  C  D  A  A  A  C  B  C  A  E  E  A  A  A  B
Levels: A B C D E
```