Machine Learning Project

## Project Requirement

# Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: <http://groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset).

## Data

The training data for this project are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv>

The test data are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>

The data for this project come from this source: <http://groupware.les.inf.puc-rio.br/har>. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

## Goal

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

## 1.0 loading library

## 2.0 Reading downloaded files

training <- read.csv("C:\\Users\\Kin\\Desktop\\R files\\Machine learning\\Project\\pml-training.csv",header = TRUE)  
  
testing <- read.csv("C:\\Users\\Kin\\Desktop\\R files\\Machine learning\\Project\\pml-testing.csv",header = TRUE)

## 3.0 Preposessing

First we clean the training data.

# Making the training data into a data frame.  
training<-as.data.frame(training)  
  
# Remove columns with more than 90% of the entries are "NA".   
training<-training[ , colMeans(is.na(training)) <0.9]  
  
# Remove columns with NearZeroVariance variables  
nzv <- nearZeroVar(training, saveMetrics=TRUE)  
training <- training[,nzv$nzv==FALSE]  
  
# Remove the first 5 columns which are time stamps and unlikely to be predictors   
training <- training[,-c(1:5)]

## 4.0 Creating training and test set for cross validation

We subdivide the training data into training and testing data set.

inTrain <- createDataPartition(training$classe, p=0.7, list=FALSE)  
sub\_Training <- training[inTrain, ]  
sub\_Testing <- training[-inTrain, ]  
dim(sub\_Training); dim(sub\_Testing)

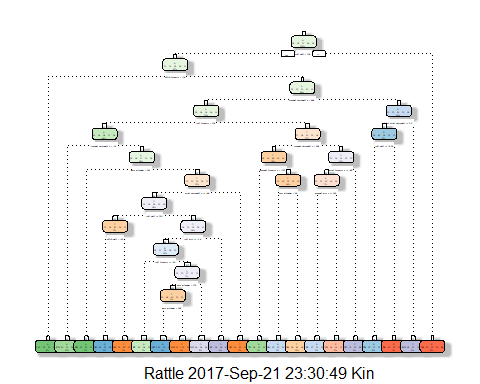
## [1] 13737 54

## [1] 5885 54

## Fitting Decision Tree

set.seed(12345)  
modFit\_rpart <- rpart(classe ~ ., data=sub\_Training, method="class")  
fancyRpartPlot(modFit\_rpart)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



pred\_rpart <- predict(modFit\_rpart, sub\_Testing, type = "class")  
Conf\_mat\_rpart <- confusionMatrix(pred\_rpart, sub\_Testing$classe)  
Conf\_mat\_rpart

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1516 282 15 92 70  
## B 48 619 73 35 99  
## C 21 72 838 139 78  
## D 64 133 75 659 130  
## E 25 33 25 39 705  
##   
## Overall Statistics  
##   
## Accuracy : 0.737   
## 95% CI : (0.7255, 0.7482)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6655   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9056 0.5435 0.8168 0.6836 0.6516  
## Specificity 0.8910 0.9463 0.9362 0.9183 0.9746  
## Pos Pred Value 0.7676 0.7082 0.7300 0.6211 0.8525  
## Neg Pred Value 0.9596 0.8962 0.9603 0.9368 0.9255  
## Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Detection Rate 0.2576 0.1052 0.1424 0.1120 0.1198  
## Detection Prevalence 0.3356 0.1485 0.1951 0.1803 0.1405  
## Balanced Accuracy 0.8983 0.7449 0.8765 0.8010 0.8131

## Fitting Randon Forest

set.seed(12345)  
modFit\_RF <- randomForest(classe ~ ., data=sub\_Training)  
pred\_RF <- predict(modFit\_RF, sub\_Testing, type = "class")  
Conf\_mat\_RF <- confusionMatrix(pred\_RF, sub\_Testing$classe)  
Conf\_mat\_RF

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1674 3 0 0 0  
## B 0 1136 2 0 0  
## C 0 0 1024 8 0  
## D 0 0 0 956 0  
## E 0 0 0 0 1082  
##   
## Overall Statistics  
##   
## Accuracy : 0.9978   
## 95% CI : (0.9962, 0.9988)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9972   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9974 0.9981 0.9917 1.0000  
## Specificity 0.9993 0.9996 0.9984 1.0000 1.0000  
## Pos Pred Value 0.9982 0.9982 0.9922 1.0000 1.0000  
## Neg Pred Value 1.0000 0.9994 0.9996 0.9984 1.0000  
## Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Detection Rate 0.2845 0.1930 0.1740 0.1624 0.1839  
## Detection Prevalence 0.2850 0.1934 0.1754 0.1624 0.1839  
## Balanced Accuracy 0.9996 0.9985 0.9982 0.9959 1.0000

## Fitting Generalised Boosting bt tree

set.seed(12345)  
fitControl <- trainControl(method = "repeatedcv",number = 5,repeats = 1)  
modFit\_gbm <- train(classe ~ ., data=sub\_Training, method = "gbm",trControl = fitControl,verbose = FALSE)

## Loading required package: gbm

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3

## Loading required package: plyr

pred\_gbm <- predict(modFit\_gbm, newdata=sub\_Testing)  
Conf\_mat\_gbm <- confusionMatrix(pred\_gbm, sub\_Testing$classe)  
Conf\_mat\_gbm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1674 10 0 0 0  
## B 0 1118 10 2 4  
## C 0 10 1013 16 0  
## D 0 1 2 946 12  
## E 0 0 1 0 1066  
##   
## Overall Statistics  
##   
## Accuracy : 0.9884   
## 95% CI : (0.9854, 0.991)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9854   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9816 0.9873 0.9813 0.9852  
## Specificity 0.9976 0.9966 0.9946 0.9970 0.9998  
## Pos Pred Value 0.9941 0.9859 0.9750 0.9844 0.9991  
## Neg Pred Value 1.0000 0.9956 0.9973 0.9963 0.9967  
## Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Detection Rate 0.2845 0.1900 0.1721 0.1607 0.1811  
## Detection Prevalence 0.2862 0.1927 0.1766 0.1633 0.1813  
## Balanced Accuracy 0.9988 0.9891 0.9910 0.9891 0.9925

## Predicting Results on the Test Data

Random Forests gave better Accuracy (99.8%) than Decision Tree (73.46%) and GBM(98.8%) in the sub\_Testing dataset.We therefore use it to predict the outcome variable classe for the testing set. The test prediction is then saved as a data file.

pred\_Test <- predict(modFit\_RF, testing, type = "class")  
pred\_Test

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 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

pred\_Test<-as.data.frame(pred\_Test)  
  
pred\_Test<-cbind(1:dim(testing)[1],pred\_Test)  
colnames(pred\_Test)<-c("Test\_record","prediction")  
write.table(pred\_Test,file="C:\\Users\\Kin\\Desktop\\R files\\Machine learning\\Project\\Test\_prediction.txt",row.names = F)