**Introduction**

Health conditions can be maintained/improved by exercise such as weight lifting. In order to measure the health conditions through these exercises, various measurements are evaluated/analysed by the data collected by devices such as Jawbone Up, Nike FuelBand, and Fitbit. In this project, data has been analysed to find pattern in the behaviour of set of groups who have been doing a particular set of activities. The aim of this report was to use data from accelerometers placed on the belt, forearm, arm, and dumbell of six participants to predict how well they were doing the exercise in terms of the classification in the data.

**Task**

The goal is to predict the manner in which people did the exercise. This is the "classe" variable in the training data set.

**Steps involved in the analysis**

1. Understanding the task
2. Downloading data
3. Pre-processing
4. Data Exploration and identifying the predictor variables
5. Partitioning the data
6. Choosing the algorithm
7. Build the model
8. Predict the unknown data
9. Measure the performance
10. Conclusion

**Data**

The data is taken from the [Human Activity Recognition](http://groupware.les.inf.puc-rio.br/har) programme at [Groupware](http://groupware.les.inf.puc-rio.br/).

**Objective**

The goal is to predict the manner in which people did the exercise. This is the "classe" variable in the training data set.

**Data Pre-Processing**

Data contains lots of NA values which would create a lot of noise for the model. As a result, these NAs were removed from the data set. The first eight columns that acted as identifiers for the experiment were also removed. First, remove all the columns which has maximum NAs, these columns will not help in our analysis anyway. Also remove the columns which does not make any sense for the predictions.

Remove the columns with all/maximum Blanks

# remove columns which contains maximum NAs

raw\_har[raw\_har==""] <- NA

nonas <- raw\_har[, colSums(is.na(raw\_har)) == 0]

# Beacuse of the above cleaning, number of variables has been reduced to 60

names(nonas)

nonas<-nonas[,-c(1,2,3,4)]

**Data Partition**

Create data partition – training and cross validation set from the train.csv file. This is required to measure the performance of the model.

# Create data partition - test and train

library(caret)

inTrain <- createDataPartition(y=raw\_har$classe,p=0.7,list=FALSE)

train<-nonas[inTrain,]

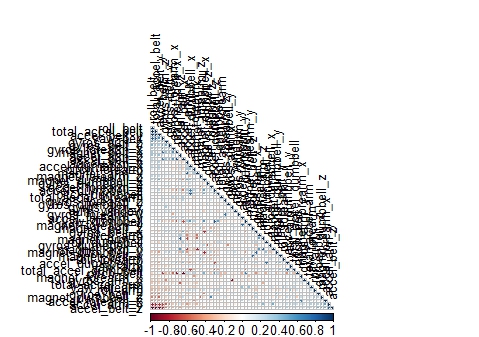
crossval<-nonas[-inTrain,]

dim(train)

dim(crossval)

**Data Exploration**

Plot the correlation between different variables to see how predictor variables are correlated. A correllation plot was produced in order to see how strong the variables relationships are.



**Build the Model**

A random forest model was selected to predict the classification because it has methods for balancing error in class population unbalanced data sets.

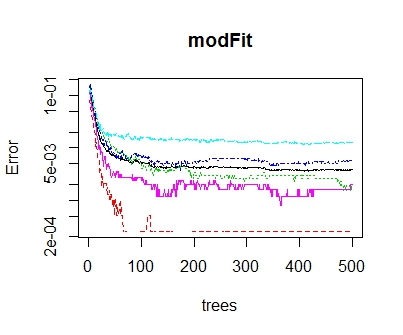
## Random Forest model to predict the classe

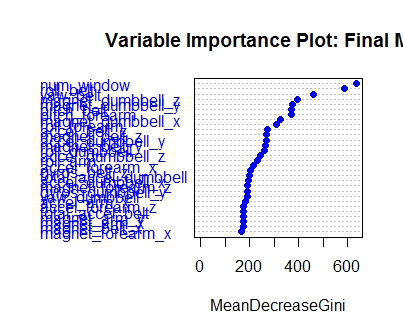
modFit<-randomForest(classe ~., data=train, mtry=3)

summary(modFit)

plot(modFit,log="y")

Plot the error.





The test data gave a prediction accuracy of 99.3%. This is as shown by the confusion matrix.

Cross validation of 30% data in the train.csv file

predCrossValCar <- predict(modCaret, crossval)

confusionMatrix(crossval$classe, predCrossValCar)

|  |
| --- |
| > confusionMatrix(crossval$classe, predCrossValCar)  Confusion Matrix and Statistics  Reference  Prediction A B C D E  A 1674 0 0 0 0  B 1 1136 2 0 0  C 0 6 1020 0 0  D 0 0 6 958 0  E 0 0 0 1 1081  Overall Statistics    Accuracy : 0.9973  95% CI : (0.9956, 0.9984)  No Information Rate : 0.2846  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.9966  Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: E  Sensitivity 0.9994 0.9947 0.9922 0.9990 1.0000  Specificity 1.0000 0.9994 0.9988 0.9988 0.9998  Pos Pred Value 1.0000 0.9974 0.9942 0.9938 0.9991  Neg Pred Value 0.9998 0.9987 0.9984 0.9998 1.0000  Prevalence 0.2846 0.1941 0.1747 0.1630 0.1837  Detection Rate 0.2845 0.1930 0.1733 0.1628 0.1837  Detection Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  Balanced Accuracy 0.9997 0.9971 0.9955 0.9989 0.9999 |
|  |
| |  | | --- | | > | |

**Results**

Finally, the prediction of the 20 testcases given in test.csv file also predicted using the above model. Before we run the predict model, the data has to be cleaned and formatted using the above methods and then finaly predict the class of the 20 test cases. The results are as shown below.

|  |
| --- |
| > predTest <-predict(modFit,test\_nonas)  > predTest  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  > summary(predTest)  A B C D E  7 8 1 1 3 |
|  |
| |  | | --- | | > | |