Practical Machine Learning - Prediction Assignment Writeup

Rajalakshmi Santhakumar

Executive Summary

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. In this project, we will use data recorded from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

The goal of this project is to predict the manner in which the volunteers did the exercise. The outcome is measured against 60 observations and classifed as "A,B,C,D,E" categorize and it is stored in the classe variable in the data set.

Loading the data

First we will load the data and the required packages.

```
library (caret)
library (munsell)
library (e1071)
library (rpart)
library (rpart.plot)
library (rattle)
library (randomForest)

set.seed(12345)
trainingSet <- read.csv("pml-training.csv", na.strings = c("NA", "#DIV/0!", ""))
testingSet  <- read.csv("pml-testing.csv",  na.strings = c("NA", "#DIV/0!", ""))
```

Cleaning the data

The data is cleaned by removing 'NA' and unnecessary variables.

```
NA_Count = sapply(1:dim(trainingSet)[2],function(x)sum(is.na(trainingSet[,x])))
NA_list = which(NA_Count>0)

trainingSet = trainingSet[,-NA_list]
trainingSet = trainingSet[,-c(1:7)]
trainingSet$classe = factor(trainingSet$classe)

NA_Count1 = sapply(1:dim(testingSet)[2],function(x)sum(is.na(testingSet[,x])))
NA_list1 = which(NA_Count1>0)
testingSet = testingSet[,-NA_list]
testingSet = testingSet[,-c(1:7)]
dim(trainingSet)
```

```
## [1] 19622    53
```

```
dim(testingSet)
```

```
## [1] 20 53
```

From the data set creating the traning and testing data partitions. The test set of 20 observations provided will be used to validate the model at the end.

```
inTrain=createDataPartition(y=trainingSet$classe, p=0.6, list=FALSE)
training <-trainingSet[inTrain,]
testing <- trainingSet[-inTrain,]
```

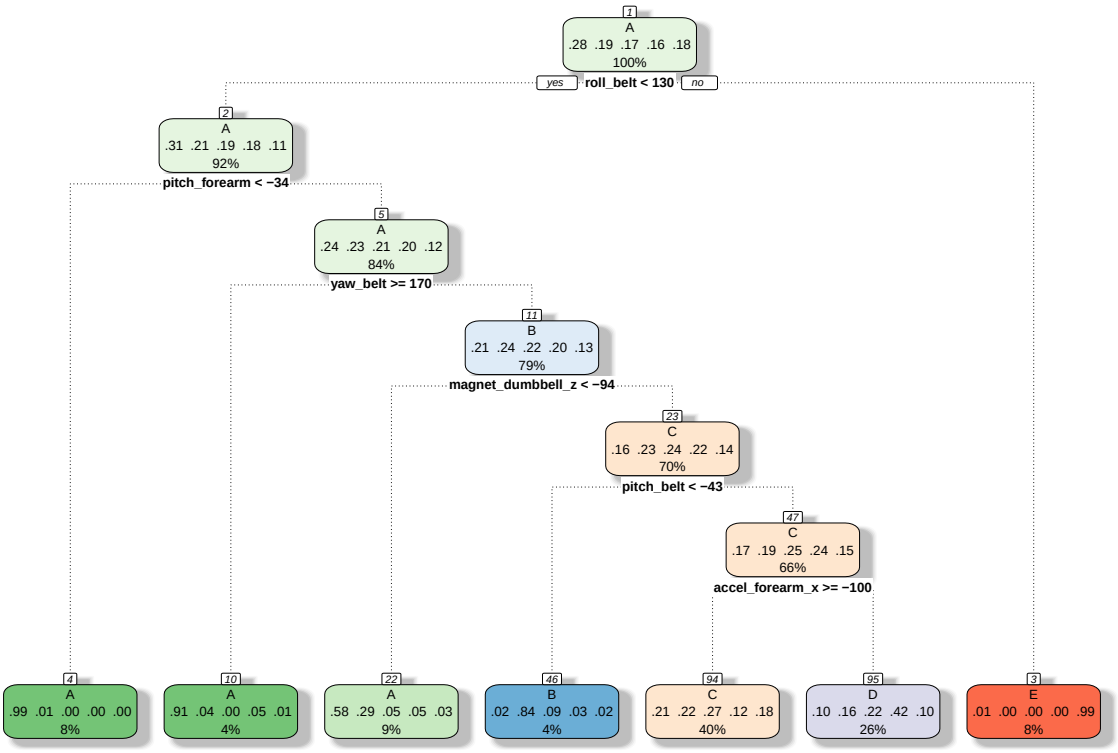
Modeling

We now create our model using the functions provided in caret package in R.

a) Tree Method

Using the Tree method to do the prediction of 'classe'.

```
modfit1 <- train(classe ~ .,method='rpart',data=training)
fancyRpartPlot(modfit1$finalModel, sub="")
```




```
pred=predict(modfit1,newdata=testing)
z=confusionMatrix(pred,testing$classe)
z$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1357  229   38   66   15
##          B    3  259   28    8   10
##          C  693  725  819  389  557
##          D  171  305  483  823  200
##          E    8    0    0    0  660
```

```
z$overall[1]
```

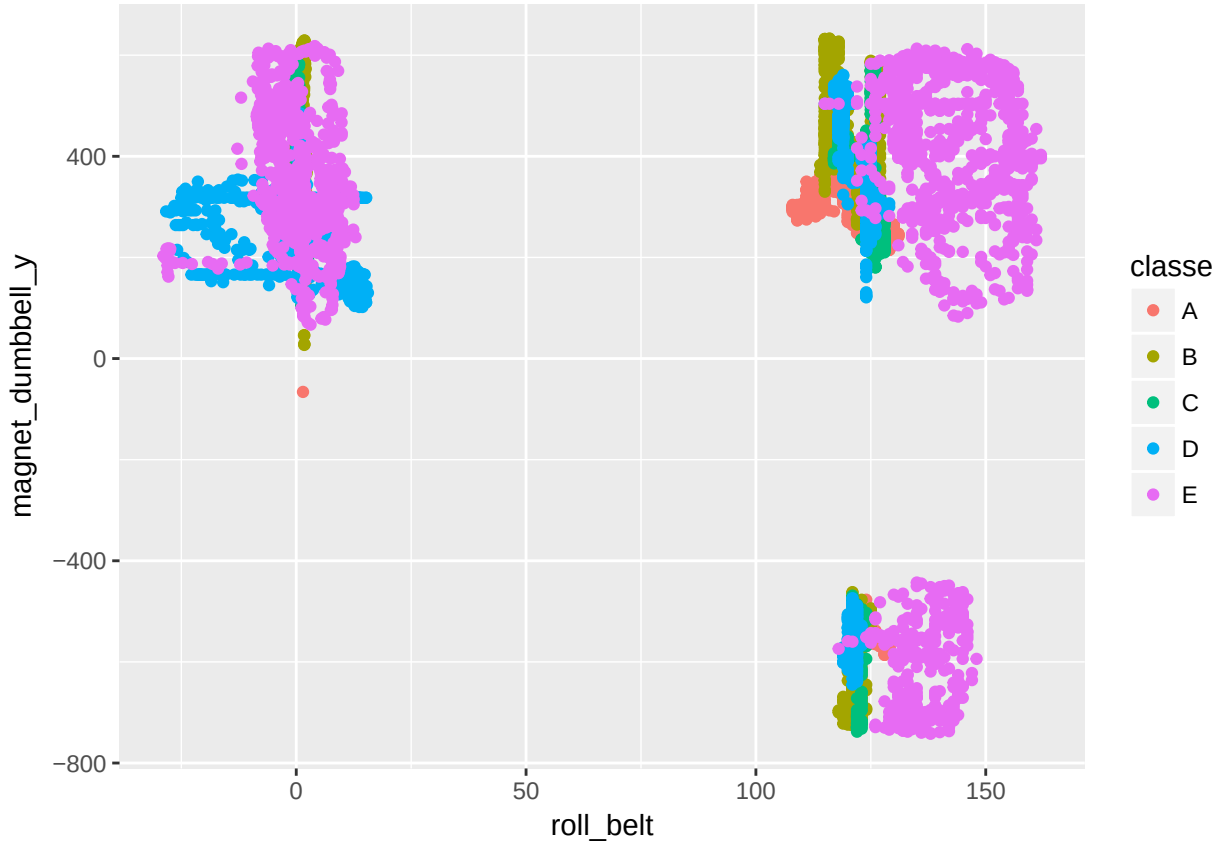
```
##  Accuracy
## 0.4993627
```

From the confusion matrix it is clear the accuracy of "0.49" for this model fit clearly shows "no purity" hence this model fit is rejected.

b) Random Forest Method

Using Random forest method to do the prediction.

```
modfit2=randomForest(classe~., data=training, method='class')
pred2 = predict(modfit2,testing,type='class')
qplot(roll_belt, magnet_dumbbell_y, colour=classe, data=training)
```




```
z2=confusionMatrix(pred2,testing$classe)
z2$table
```

```
##            Reference
## Prediction    A    B    C    D    E
##          A 2231    7    0    0    0
##          B    1 1507    6    0    0
##          C    0    4 1361   16    3
##          D    0    0    1 1268    4
##          E    0    0    0    2 1435
```

```
z2$overall[1]
```

```
## Accuracy
## 0.994392
```

Random forest prediction model provides 99% accurancy hence this model has been choosen to do predict the testing data set.

Conclusion

From the above results the random forest method provides the best fit model and it is been considered for testing the test data set to submit results.

```
pred3 =  predict(modfit2,testingSet,type='class')
nofiles = length(pred3)
for (i in 1:nofiles){
```

```
    filename =  paste0("problem_id",i,".txt")
    write.table(pred3[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
}
pred3
```

```
##  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
```