Practical Machine Learning - Prediction Assignment Writeup

Overview

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.

The goal of this writup is to use the data captured by the accelerometers on the devices worn by the 6 participants. The devices will be worn on the forearm, arm, and dumbell. The partifipants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Data

Source location for training data:

- https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

Source location for test data:

- https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```
setwd("./")

if (!file.exists("./data")) {
  dir.create("./data")
}
if (!file.exists("./data/pml-training.csv")) {
  url.training <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
  download.file(url.training, destfile = "./data/pml-training.csv")
}

if (!file.exists("./data/pml-testing.csv")) {
  url.testing <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
  download.file(url.testing, destfile = "./data/pml-testing.csv")
}
```

Reading data and data processing

```
## [1] 19622   160
```

```
## [1]  20 160
```

Breakdown of the collected data: Training data set contains 19622 observations and 160 variables.

Testing data set contains 20 observations and 160 variables.

Clean data to remove variables which are close to zero variance and columns that contain 'N/A' missing values.

Cleaning the data will remove all columns that contains NA and remove features that are not in the testing dataset. Also, since there is no time dependant features, they will also be removed along with features which are not neumeric.

```
train <- train[, colSums(is.na(train)) == 0]
test <- test[, colSums(is.na(test)) == 0]
classe <- train$classe
trainR <- grepl("^X|timestamp|window", names(train))
train <- train[, !trainR]
trainM <- train[, sapply(train, is.numeric)]
trainM$classe <- classe
testR <- grepl("^X|timestamp|window", names(test))
test<- test[, !testR]
testM <- test[, sapply(test, is.numeric)]

dim(trainM); dim(testM);
```

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

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

Data Partitioning

The training data is separated into two data sets into two data sets: 1) 70% for train data 2) 30% for test data, to be used for validation purpose

```
library(caret)
set.seed(62374)
inTrain <- createDataPartition(trainM$classe, p=0.70, list=F)
train_data <- trainM[inTrain, ]
test_data <- trainM[-inTrain, ]
dim(train_data); dim(test_data);
```

```
## [1] 13737    53
```

```
## [1] 5885   53
```

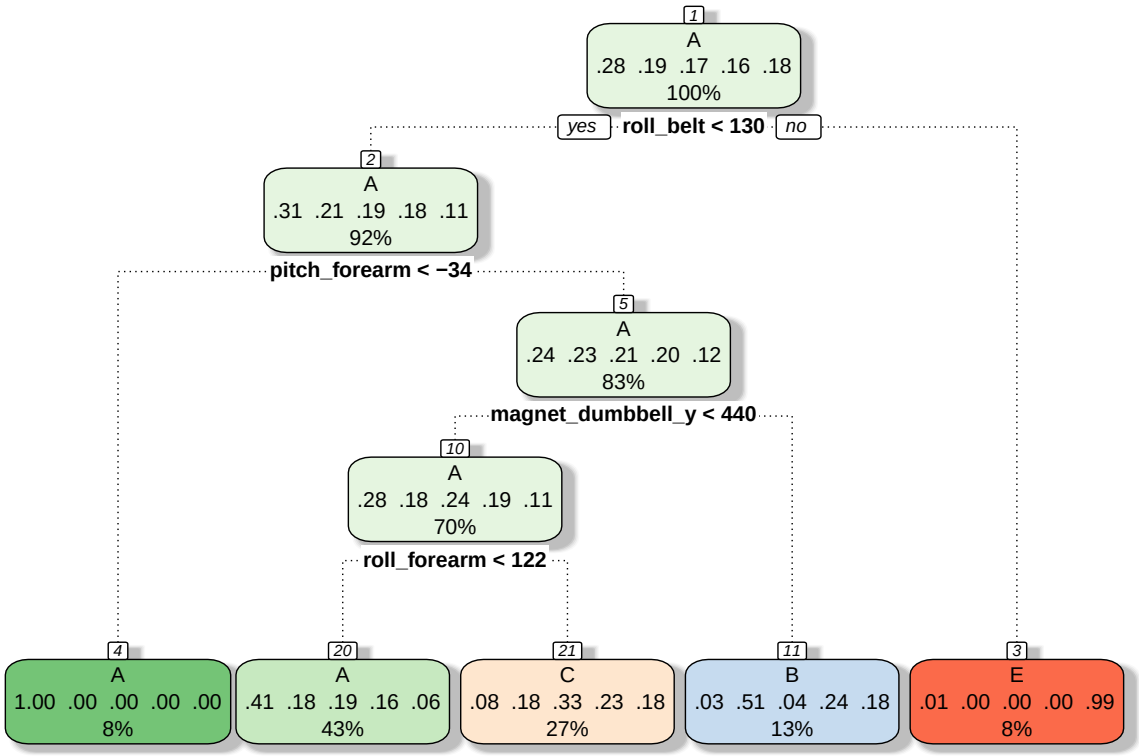
Data Prediction and Modelling

1. Decision Tree

Predicting with the Decision Tree Model

The results of the decision treee testing were not optimal, as expected.

```
treeModel <- train(classe ~ .,method='rpart',data=train_data)
fancyRpartPlot(treeModel$finalModel)
```




```
set.seed(23142)

prediction <- predict(treeModel$finalModel, test_data, type = "class")
confusionMatrix(prediction, test_data$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1518  505  476  469  154
##          B   28  366   29  143  158
##          C  122  268  521  352  298
##          D    0    0    0    0    0
##          E    6    0    0    0  472
##
## Overall Statistics
##
##                Accuracy : 0.4889
##                  95% CI : (0.476, 0.5017)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3311
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9068  0.32133  0.50780   0.0000  0.43623
## Specificity            0.6191  0.92457  0.78596   1.0000  0.99875
## Pos Pred Value         0.4862  0.50552  0.33376      NaN  0.98745
## Neg Pred Value         0.9435  0.85022  0.88321   0.8362  0.88718
## Prevalence             0.2845  0.19354  0.17434   0.1638  0.18386
## Detection Rate         0.2579  0.06219  0.08853   0.0000  0.08020
## Detection Prevalence   0.5305  0.12302  0.26525   0.0000  0.08122
## Balanced Accuracy      0.7630  0.62295  0.64688   0.5000  0.71749
```

The Accuracy of 0.554, rules out using the Decision tree as basis to provide trustworthy guidance.

```
accu=confusionMatrix(prediction,test_data$classe)
accu$overall[1]
```

```
## Accuracy
##  0.48887
```

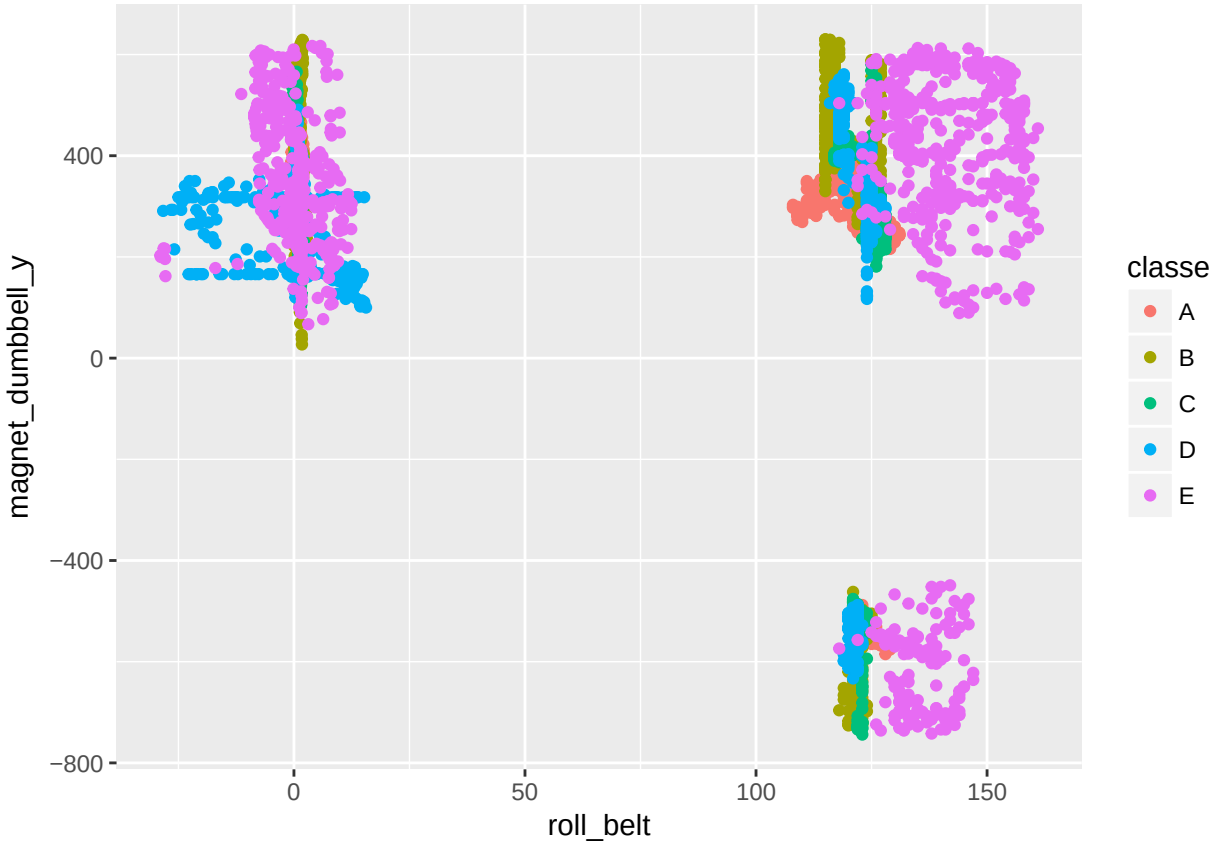
2. Random Forest

Predicting with the Decision Tree Model

The results of the Random Forest testing delivered $< 1\%$

```
set.seed(86752)
rf_fit=randomForest(classe~., data=train_data, method='class')
rf_pred = predict(rf_fit,test_data,type='class')
```

```
qplot(roll_belt, magnet_dumbbell_y, colour=classe, data=test_data)
```




```
accu2=confusionMatrix(rf_pred,test_data$classe)
accu2$overall[1]
```

```
##  Accuracy
## 0.9938828
```

Conclusion

Algorithm which will be used for predictive model here is the Random Forest.It can be seen from the confusion matrix the Random Forest model is 99% accurate.