Machine Learning Course Project

Sheetal

25 December 2018

Introduction

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.

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.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.3.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.3.3
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.3.3
```

```
library(rpart.plot)
```

Import in datasets

```
test <- read.csv("C:/Users/Sheetal/datasciencecoursera/machinelearning project/pml-testing.csv")
train <- read.csv("C:/Users/Sheetal/datasciencecoursera/machinelearning project/pml-training.csv")

trainRaw <- train[, colSums(is.na(train)) == 0]
testRaw <- test[, colSums(is.na(test)) == 0]

classe <- trainRaw$classe
trainRemove <- grepl("^X|timestamp|window", names(trainRaw))
trainRaw <- trainRaw[, !trainRemove]
trainCleaned <- trainRaw[, sapply(trainRaw, is.numeric)]
trainCleaned$classe <- classe
testRemove <- grepl("^X|timestamp|window", names(testRaw))
testRaw <- testRaw[, !testRemove]
testCleaned <- testRaw[, sapply(testRaw, is.numeric)]
```

The aim is to predict classe which is the manner in which they did their exercise (sitting-down, standing-up, standing, walking, and sitting)

Exploratory Data Analysis

```
table(trainCleaned$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```
prop.table(table(trainCleaned$classe))
```

```
##
##         A         B         C         D         E
## 0.2843747 0.1935073 0.1743961 0.1638977 0.1838243
```

This shows that class A has the most observations and the largerst proportion 28.43%

Split Datasets

```
set.seed(22519) # For reproducibile purpose
inTrain <- createDataPartition(trainCleaned$classe, p=0.70, list=F)
trainData <- trainCleaned[inTrain, ]
testData <- trainCleaned[-inTrain, ]
```

We then build the random forrest model,as it will chose the variables of most importance

```
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=controlRf, ntree=25)
modelRf
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10991, 10988, 10989, 10991
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9875512  0.9842516
##   27    0.9893709  0.9865538
##   52    0.9819467  0.9771590
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The accuracy of the model in the train dataset is 98.6 %. Now we can predict the accuract on the test dataset to validate.

```
predictRfmod<- predict(modelRf, testData)
confusionMatrix(testData$classe, predictRfmod)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    2    0    0    1
##          B    9 1125    5    0    0
##          C    0    2 1020    4    0
##          D    0    0   17  946    1
##          E    0    0    1    5 1076
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.9894, 0.9941)
##     No Information Rate : 0.2855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9899
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9946   0.9965   0.9779   0.9906   0.9981
## Specificity            0.9993   0.9971   0.9988   0.9963   0.9988
## Pos Pred Value         0.9982   0.9877   0.9942   0.9813   0.9945
```

```
## Neg Pred Value         0.9979   0.9992   0.9953   0.9982   0.9996
## Prevalence             0.2855   0.1918   0.1772   0.1623   0.1832
## Detection Rate         0.2839   0.1912   0.1733   0.1607   0.1828
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9970   0.9968   0.9884   0.9935   0.9984
```

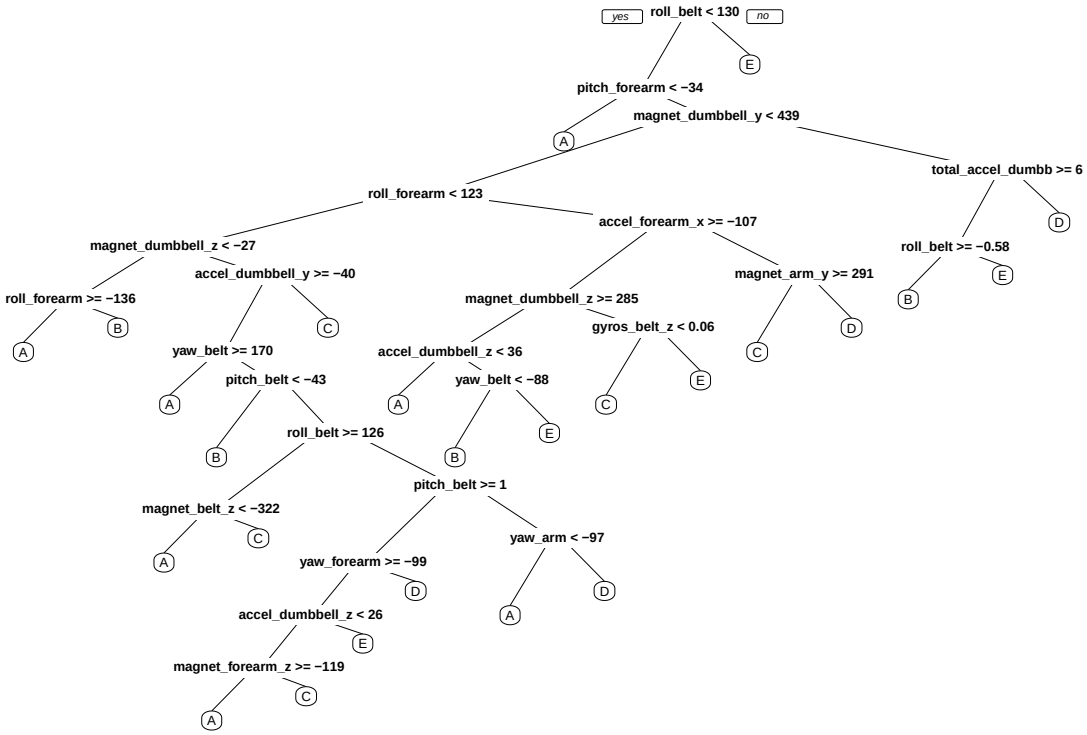
The accuracy of the modl is 99.2%

The below section shows the results for the test data sample

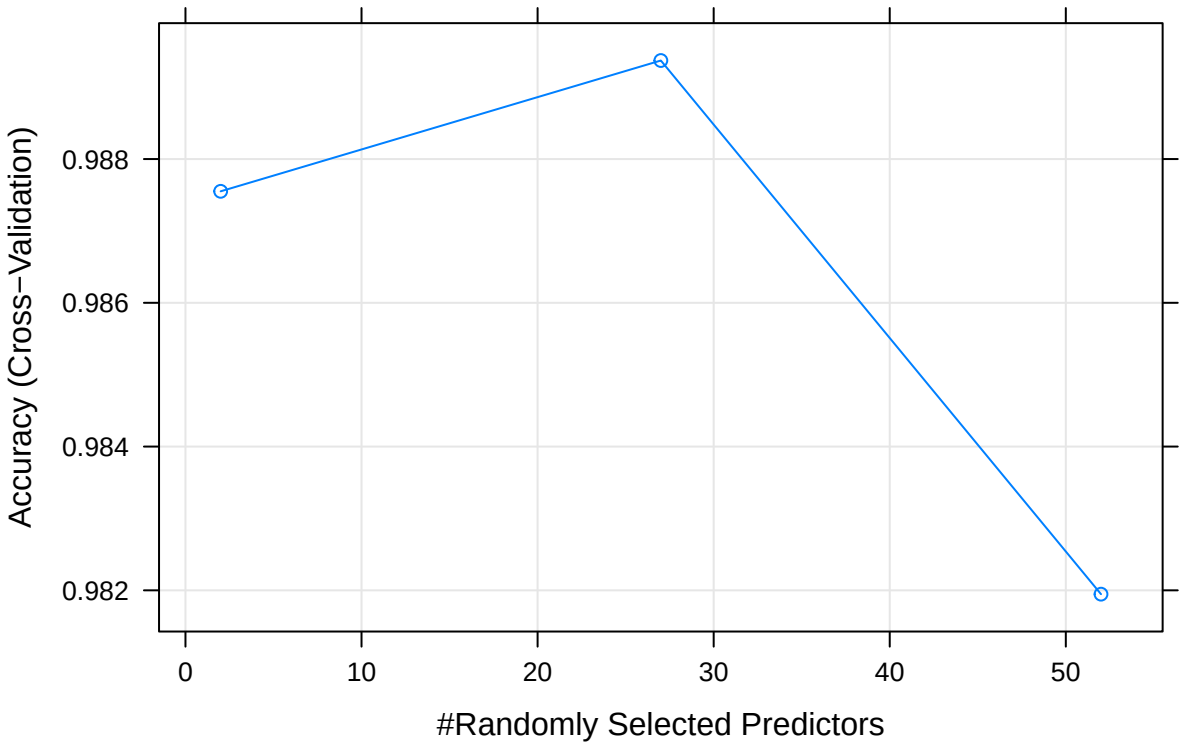
```
result <- predict(modelRf, testCleaned[, -length(names(testCleaned))])
result
```

```
##  [1] 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
```

```
treeModel <- rpart(classe ~ ., data=trainData, method="class")
prp(treeModel) # fast plot
```




```
plot(modelRf,main="Accuracy of Random forest model by number of predictors")
```



Conclusion

The random forrest model has given a highly accurate result with 27 predictors. The model has probably worked well due to the large amount of variables that we had to explore.