Coursera_Practical Machine Learning_CourseProject

Benjamin Smith

February 25, 2018

Introduction

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, we will use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise. 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 were generously provided by: http://groupware.les.inf.puc-rio.br/har .

Data Preparation

```r
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
library(rpart)
library(rpart.plot)
library(randomForest)
```

```
## 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
```

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

Download Data

```
trainUrl <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainFile <- "./data/pml-training.csv"
testFile  <- "./data/pml-testing.csv"
if (!file.exists("./data")) {
  dir.create("./data")
}
if (!file.exists(trainFile)) {
  download.file(trainUrl, destfile=trainFile, method="auto")
}
if (!file.exists(testFile)) {
  download.file(testUrl, destfile=testFile, method="auto")
}
```

Read Data

After downloading the data from the data source, read the two csv files into separate data frames.

```
trainRaw <- read.csv("./data/pml-training.csv")
testRaw <- read.csv("./data/pml-testing.csv")
dim(trainRaw)
```

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

```
dim(testRaw)
```

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

The training data set contains 19622 observations and 160 variables, while the testing data set contains 20 observations and 160 variables. The "classe" variable in the training set is the outcome for which the models will predict.

Data Cleansing

Remove observations with missing values and remove meaningless variables.

```
sum(complete.cases(trainRaw))
```

```
## [1] 406
```

Remove columns that contain NA missing values.

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

Remove columns that do not contribute materially to the accelerometer measurements.

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

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

The cleansed training set contains 19622 observations and 53 variables, while the testing data set contains 20 observations and 53 variables. The "classe" variable remains in the cleaned training set.

Slice the data

Split the cleansed training set into a new training data set (70%) and a new validation data set (30%) to apply cross validation in future steps.

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

Data Modeling

Fit a predictive model for activity recognition using **Random Forest** algorithm. Random Forest (RF) automatically selects important variables and is effective at dealing with correlated covariates & outliers, generally. This analysis employes a **5-fold cross validation** when applying the algorithm.

```r
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=controlRf, ntree=100)
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    0.9904626  0.9879352
##   27    0.9906810  0.9882110
##   52    0.9849312  0.9809359
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

Then, we estimate the performance of the model on the validation data set.

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##           A 1672    1    0    0    1
##           B    7 1129    3    0    0
##           C    0    2 1019    5    0
##           D    0    0   15  948    1
##           E    0    0    1    6 1075
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9904, 0.9949)
##     No Information Rate : 0.2853
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9958   0.9973   0.9817   0.9885   0.9981
## Specificity            0.9995   0.9979   0.9986   0.9968   0.9985
## Pos Pred Value         0.9988   0.9912   0.9932   0.9834   0.9935
## Neg Pred Value         0.9983   0.9994   0.9961   0.9978   0.9996
## Prevalence             0.2853   0.1924   0.1764   0.1630   0.1830
## Detection Rate         0.2841   0.1918   0.1732   0.1611   0.1827
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9977   0.9976   0.9901   0.9926   0.9983
```

```r
accuracy <- postResample(predictRf, testData$classe)
accuracy
```

```
##  Accuracy     Kappa
## 0.9928632 0.9909721
```

```r
oose <- 1 - as.numeric(confusionMatrix(testData$classe, predictRf)$overall[1])
oose
```

```
## [1] 0.007136788
```

The estimated accuracy of the model is 99.29, 99.1%, and the estimated out-of-sample error is 0.71%.

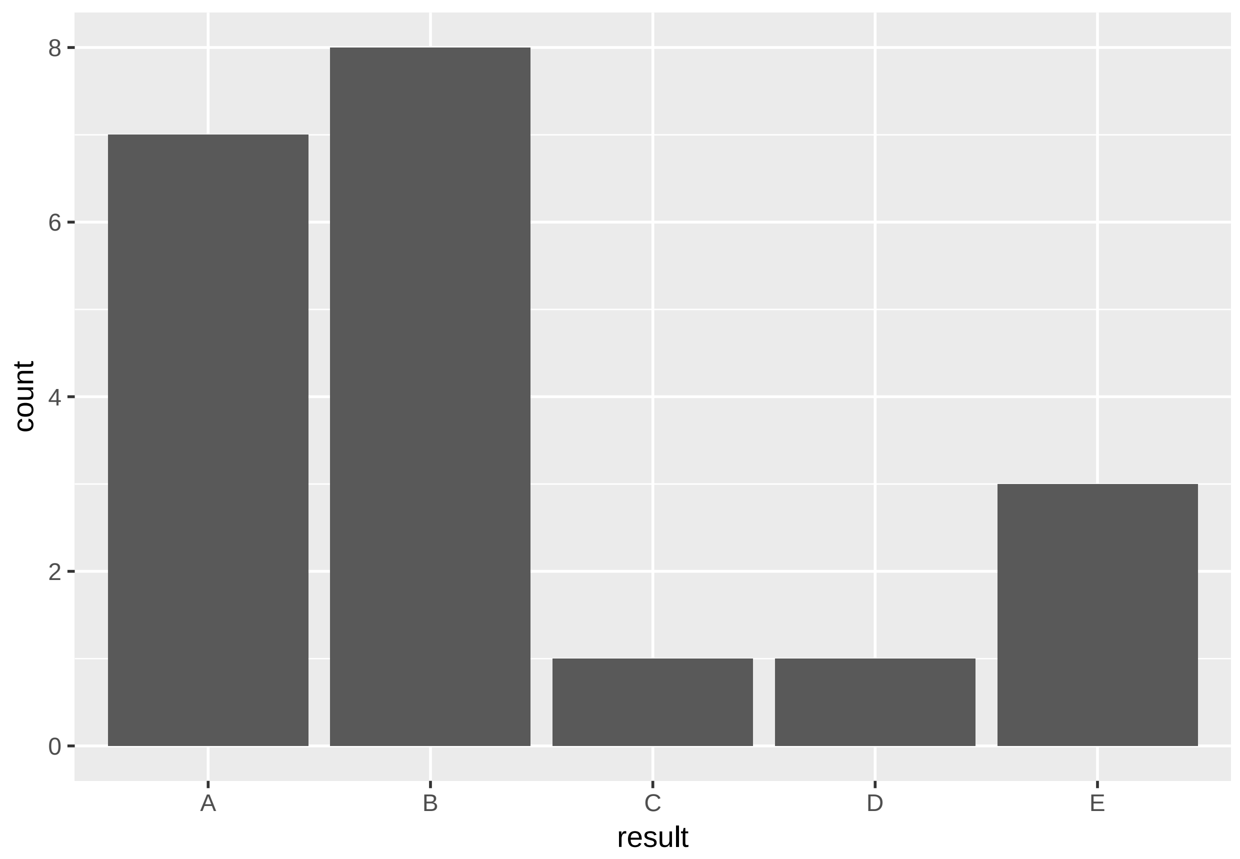
Predicting for Test Data Set

Once a goo dmodel is determined, apply the model to the original test data set downloaded from the data source and remove the `problem_id` column.

```r
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
```

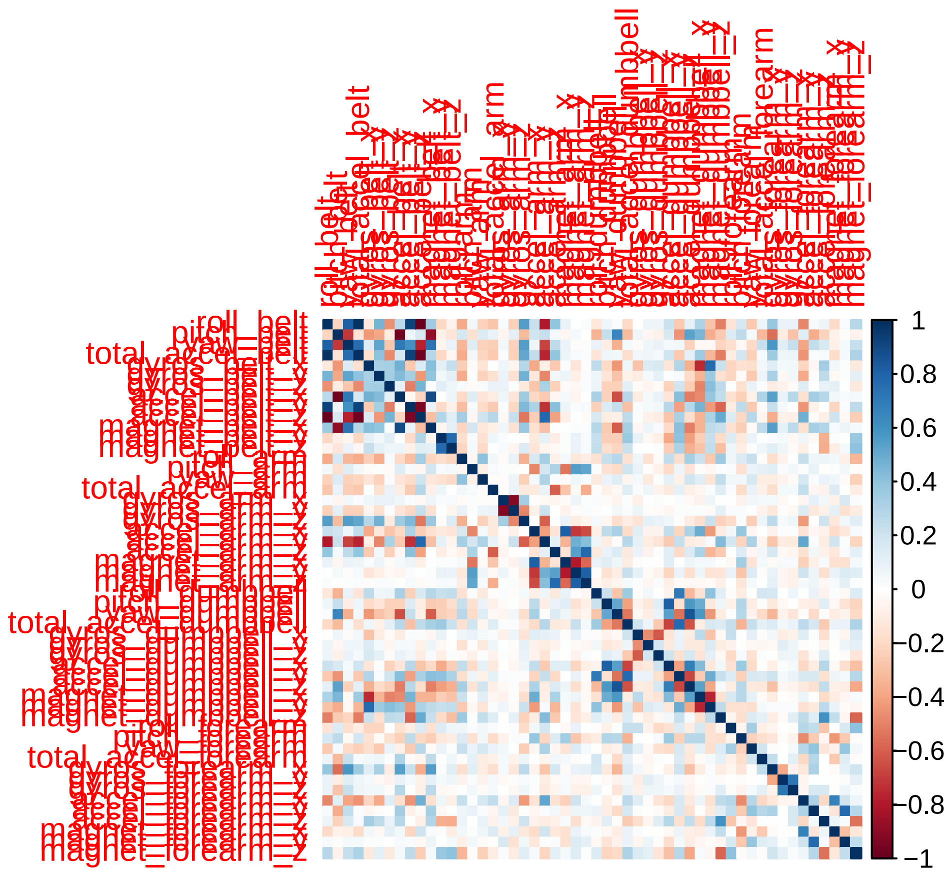
```r
qplot(result)
```



Appendix: Figures

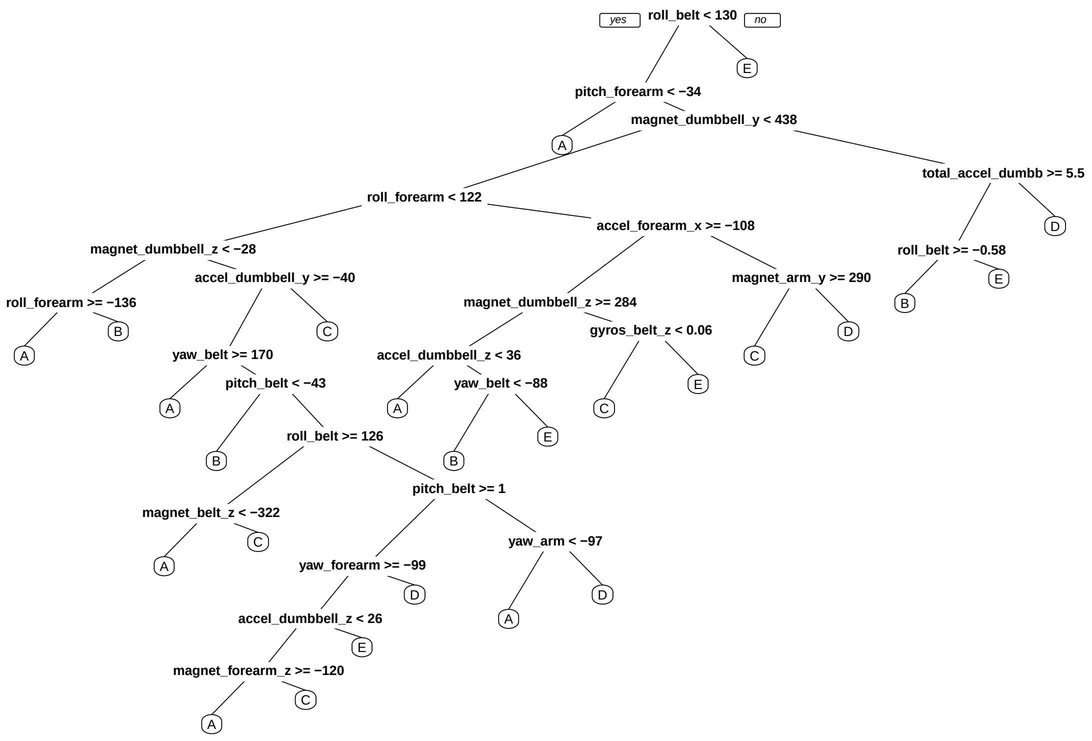
1. Correlation Matrix Visualization

```
corrPlot <- cor(trainData[, -length(names(trainData))])
corrplot(corrPlot, method="color")
```



2. Decision Tree Visualization

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



yes
roll_belt < 130
no
E
pitch_forearm < −34
A
magnet_dumbbell_y < 438
total_accel_dumbb >= 5.5
D
roll_belt >= −0.58
B
E
roll_forearm < 122
magnet_dumbbell_z < −28
roll_forearm >= −136
A
B
accel_dumbbell_y >= −40
C
yaw_belt >= 170
A
pitch_belt < −43
B
roll_belt >= 126
magnet_belt_z < −322
A
C
pitch_belt >= 1
yaw_forearm >= −99
D
accel_dumbbell_z < 26
E
magnet_forearm_z >= −120
A
C
yaw_arm < −97
A
D
accel_forearm_x >= −108
magnet_dumbbell_z >= 284
accel_dumbbell_z < 36
A
yaw_belt < −88
B
E
gyros_belt_z < 0.06
C
E
magnet_arm_y >= 290
C
D