Practical Machine Learning - Final Project

Szymon Lipiński

8/7/2019

Introduction

Background

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, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. 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 come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Goal of the Project

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.

Loading All Needed Libraries

```
library(caret)
library(rpart)
library(rattle)
library(randomForest)
```

Getting The Data

```
set.seed(12345)
trainingURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testingURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

originalTraining <- read.csv(url(trainingURL), na.strings=c("NA", "#DIV/0!", ""))
originalTesting <- read.csv(url(testingURL), na.strings=c("NA", "#DIV/0!", ""))
```

Cleaning The Data

The testing csv file contains more columns than the training one, so I need to remove some of them.

```
colnames <- colnames(originalTraining)[!colSums(is.na(originalTraining)) > 0]
colnames <- colnames[8: length(colnames)]
useTraining <- originalTraining[colnames]

trainingColNames <- colnames(useTraining)
testingColNames  <- colnames(originalTesting)

commonColumns <- intersect(trainingColNames, testingColNames)

useFinalTesting <- originalTesting[commonColumns]
useTraining     <- useTraining[append(commonColumns, c("classe"))]

useFinalTesting <- rbind(useTraining[1, commonColumns], useFinalTesting)
useFinalTesting <- useFinalTesting[-1,]
```

Preparing The Training Sets

```
inTrain <- createDataPartition(y=useTraining$classe, p=0.8, list=FALSE)
training <- useTraining[inTrain, ]
testing <- useTraining[-inTrain, ]
```

```
dim(training)
```

```
## [1] 15699    53
```

```
dim(testing)
```

```
## [1] 3923   53
```

Training

For training I'm using random forest method.

```
model <- randomForest(classe ~ ., data=training)
prediction <- predict(model, testing)
```

```
confusionMatrix(testing$classe, prediction)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    0    0    0    0
##          B    2  757    0    0    0
##          C    0    2  681    1    0
##          D    0    0    7  635    1
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.9967
##                  95% CI : (0.9943, 0.9982)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9958
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9974   0.9898   0.9984   0.9986
## Specificity            1.0000   0.9994   0.9991   0.9976   1.0000
## Pos Pred Value         1.0000   0.9974   0.9956   0.9876   1.0000
## Neg Pred Value         0.9993   0.9994   0.9978   0.9997   0.9997
## Prevalence             0.2850   0.1935   0.1754   0.1621   0.1840
## Detection Rate         0.2845   0.1930   0.1736   0.1619   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9991   0.9984   0.9944   0.9980   0.9993
```

The decision tree algorithm gave 100% accuracy on the training set.

Predicting Final Results

```
predict(model, useFinalTesting)
```

```
##  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  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
```