Machine Learning Project

Andrew Rosa

January 29, 2016

Overview

From Assignment: 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).

The Goal of this assignment is to use this data to create a predictor that will predict which excersize a subject did.

Load Packages

```
library(dplyr)
```

```
##
## 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(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.3
```

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

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

Set Seed and Load Data

```
set.seed(54321)

TrainingData <- read.csv("/Users/andrewrosa/Desktop/pml-training.csv", na.strings=c("NA","#DIV/0!",""))
TestData <- read.csv("/Users/andrewrosa/Desktop/pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Divide and Clean Data Into Training & Test Sets

```
Train <- createDataPartition(y=TrainingData$classe, p=0.6, list=FALSE)
Training <- TrainingData[Train, ]
Testing <- TrainingData[-Train, ]


TrainClean <- Training[, apply(Training, 2, function(x) !any(is.na(x)))]
TestClean <- Testing[, apply(Testing, 2, function(x) !any(is.na(x)))]
ValSet <- TestData[, apply(TestData, 2, function(x) !any(is.na(x)))]
```

Create a Predictive Model

```
fitCon <- trainControl(method="cv", number=5, allowParallel=T, verbose=T)
rffit <- train(classe~.,data=TrainClean, method="rf", trControl=fitCon, verbose=F)
```

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

## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=41
## - Fold1: mtry=41
## + Fold1: mtry=81
## - Fold1: mtry=81
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=41
## - Fold2: mtry=41
## + Fold2: mtry=81
## - Fold2: mtry=81
## + Fold3: mtry= 2
```

```
## - Fold3: mtry= 2
## + Fold3: mtry=41
## - Fold3: mtry=41
## + Fold3: mtry=81
## - Fold3: mtry=81
## + Fold4: mtry= 2
## - Fold4: mtry= 2
## + Fold4: mtry=41
## - Fold4: mtry=41
## + Fold4: mtry=81
## - Fold4: mtry=81
## + Fold5: mtry= 2
## - Fold5: mtry= 2
## + Fold5: mtry=41
## - Fold5: mtry=41
## + Fold5: mtry=81
## - Fold5: mtry=81
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 41 on full training set
```

Run Tests

```
predrf <- predict(rffit, newdata=TestClean)
confusionMatrix(predrf, TestClean$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    0 1518    0    0    0
##          C    0    0 1368    0    0
##          D    0    0    0 1286    0
##          E    0    0    0    0 1442
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9995, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
```

```
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```
predValSet <- predict(rffit, newdata=ValSet)
```

Save Files

```
getwd()
```

```
## [1] "/Users/andrewrosa/Desktop"
```

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(predValSet)
```