Qualitative Analysis of Human Activity Record

Yashu Seth

May 23, 2016

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. In this project, my 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).

Getting Data

```r
dtrain <- read.csv("C:/work/R/pml-training.csv")
dtest <- read.csv("C:/work/R/pml-testing.csv")
```

Cleaning Data

As we can observe the data has a lot of columns having NA values and blank data. Let us remove these columns first.

```r
# Removing columns with blank values and NA values

r <- which(sapply(dtrain, function(x){sum(x=="")==0}))
dtrain <- dtrain[,r]

r <- which(sapply(dtest, function(x){sum(x=="")==0}))
dtest <- dtest[,r]
```

Now we should also remove the unimportant columns. The first 7 columns of the data frame would not help in a qualitative analysis.

```r
imp_col <- c(8:dim(dtrain)[2])

dtrain <- dtrain[,imp_col]

dtest <- dtest[,imp_col]
```

Now let us convert the classes column as factor variables.

```
dtrain$classe <- as.factor(dtrain$classe)
```

Cross Validation

```
library("caret")
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
intrain <- createDataPartition(dtrain$classe, p=0.60)
dcv <- dtrain[-intrain[[1]],]
dtraining <- dtrain[intrain[[1]],]
```

Model Selection

Before doing any exploratory data analysis let us first apply certain models to see how they perform. Let us apply a random forest model.

```
library("randomForest")
```

```
## Warning: package 'randomForest' was built under R version 3.2.5
```

```
## 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
```

```
rf_fit <- randomForest(classe~., data=dtraining, ntree=100,
                       do.Trace=T)
rf_fit
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = dtraining, ntree = 100,      do.Trace = T)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.74%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3344    2    2    0    0 0.001194743
## B   22 2247   10    0    0 0.014041246
## C    0   12 2039    3    0 0.007302824
## D    0    0   21 1908    1 0.011398964
## E    0    0    5    9 2151 0.006466513
```

As we can see the error rate is less than 1%, we will stick with this model.

Prediction

```r
pred <- predict(rf_fit, dtest)
print(pred)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

Conclusion

Hence, we have used a random forest model, after certain data cleaning to construct our predictor.