Practical Machine Learning Project

Flora Ye

Oct 1, 2017

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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/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 (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/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.

Load the data

```
pm1training <- read.csv("pml-training.csv", na.strings = c("NA", ""))
pm1testing <- read.csv("pml-testing.csv", na.strings = c("NA", ""))
dim(pm1training)
```

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

```
dim(pm1testing)
```

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

Both training data and testing data set has 160 variables.

Data Cleaning

Get rid of the columns that contain ONLY NAs

```
cols <- colSums(is.na(pm1training)) ==0
pm1training <- pm1training[, cols ]
pm1testing <- pm1testing[, cols]
dim(pm1training)
```

```
## [1] 19622    60
```

```
dim(pm1testing)
```

```
## [1] 20 60
```

Now the training data has 19622 rows and 60 variables, and the testig data has 20 rows and 60 variables. Remove column 1 to 7 which is irrelevant to accelerometer measurements. To reduce the dimension of the data, we will find the correlations of the variables, and remove the variables that are correlated. Also remove the classe (which is a factor) column to find the correlations of the variables.

```
data.new <- pm1training[,-c(1:7, ncol(pm1training))]
```

Remove highly correlated variables

```
tmp <- cor(data.new)
tmp[!lower.tri(tmp)] <- 0
inCor <- !apply(tmp,2,function(x) any(abs(x) > 0.75))
data.new <- data.new[,inCor]
data.new$classe <- pm1training$classe
pm1training <- data.new
```

Clean the testing data

```
pm1testing <- pm1testing[,-c(1:7)]
pm1testing <- pm1testing[, inCor]
```

Now the data has 31 variables.

Split data into training and validation data

```
suppressWarnings(library(caret))
```

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

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

```
set.seed(123456)
inTrain = createDataPartition(pm1training$classe, p = 3/4, list = FALSE)
training <- pm1training[inTrain,]
validation <- pm1training[-inTrain,]
```

Fit a model

Here we use 5-fold Cross Validation, and Random Forrest to fit a model on the training data.

```
modFit <- train(classe ~ ., data=training, method="rf", trControl=trainControl(method
="cv", 5))
```

```
## 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
```

```
modFit
```

```
## Random Forest 
## 
## 14718 samples
##    30 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E' 
## 
## No pre-processing
## Resampling: Cross-Validated (5 fold) 
## Summary of sample sizes: 11774, 11776, 11774, 11776, 11772 
## Resampling results across tuning parameters:
## 
##   mtry  Accuracy   Kappa    
##    2    0.9854596  0.9816022
##   16    0.9844406  0.9803132
##   30    0.9773062  0.9712824
## 
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

Check the accuracy of the model on the validation dat

```
pred <- predict(modFit, validation)
confusionMatrix(pred, validation$classe)$overall[1]
```

```
##  Accuracy 
## 0.9916395
```

The accuracy on the validation data is 0.99 which is very high.

Run the model on the testing data

```
predict(modFit, pm1testing)
```

```
##  [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
```