Practical Machine Learning Assignment

Pradeep

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, 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 (see the section on the Weight Lifting Exercise Dataset).

Data Location

- Training data is available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
- Testing data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

Goal

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.

Data Manipulation

Initial Setup

Create local files and add required packages.

```
## Data location
trainingFilePath <- './data/pml-training.csv'
testingFilePath <- './data/pml-testing.csv'
trainingDataUrl <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
testingDataUrl <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'

# Loca directories
```

```r
if (!file.exists("data")){
  dir.create("data")
}
if (!file.exists("data/submission")){
  dir.create("data/submission")
}

# Packages
library("caret")
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

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

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library("rpart")
library("rpart.plot")
library(rattle)
```

```
## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```r
# Setting seed to facilitate reproducible results
set.seed(1234)
```

Data Processing

```
# Download data
download.file(trainingDataUrl, trainingFilePath)
download.file(testingDataUrl,testingFilePath )

# Clean data
trainingDF <- read.csv(trainingFilePath, na.strings=c("NA","#DIV/0!", ""))
testingDF <- read.csv(testingFilePath , na.strings=c("NA", "#DIV/0!", ""))
trainingDF <- trainingDF[,colSums(is.na(trainingDF)) == 0]
testingDF <- testingDF[,colSums(is.na(testingDF)) == 0]

# remove the Near Zero Variance columns
NZV <- nearZeroVar(trainingDF)
trainingDF <- trainingDF[,-NZV]
testingDF <- testingDF[,-NZV]

# remove ID variables
trainingDF <- trainingDF[,-(1:5)]
testingDF <- testingDF[,-(1:5)]

# See dimensions of data frames
dim(trainingDF)
```

```
## [1] 19622    54
```

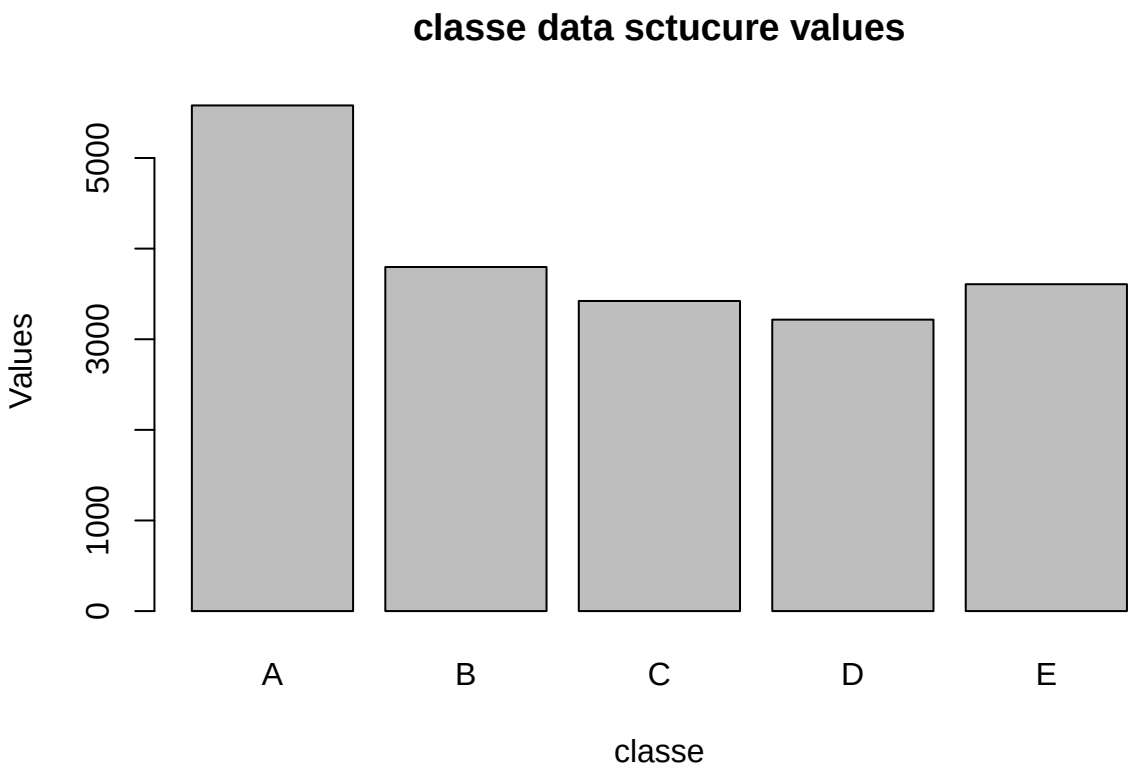
```
dim(testingDF)
```

```
## [1] 20 54
```

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

Data Exploration

```
plot(as.factor(trainingDF$classe), main="classe data sctucure values", xlab="classe", ylab="Values", )
```



Datastructures Creation

Create data structures for training. Use 5 folds Cross-Validation.

```
subSamplesDF <- createDataPartition(y=trainingDF$classe, p=0.6, list=FALSE)
createMultiFolds(subSamplesDF, k = 5, times = 5)
subTrainingDF <- trainingDF[subSamplesDF, ]
subTestingDF <- trainingDF[-subSamplesDF, ]
```

See dimensions of data structures

```
dim(subTrainingDF)
```

```
## [1] 11776    54
```

```
dim(subTestingDF)
```

```
## [1] 7846   54
```

Models

Two models will be created. The model with higher accuracy will be used to answer the quiz question.

Decision Tree

```
# Fitting the model
modFitDT <- rpart(classe ~ ., data=subTrainingDF, method="class")

# Performing prediction
predictDT <- predict(modFitDT, subTestingDF, type = "class")
```

```
confusionMatrix(predictDT, subTestingDF$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2017  299   61   84   94
##          B   94  908   55  108  167
##          C    7   96 1074  168   73
##          D   59   92   71  761   59
##          E   55  123  107  165 1049
##
## Overall Statistics
##
##                Accuracy : 0.7404
##                  95% CI : (0.7305, 0.7501)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6697
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9037   0.5982   0.7851  0.59176   0.7275
## Specificity            0.9042   0.9330   0.9469  0.95716   0.9297
## Pos Pred Value         0.7894   0.6817   0.7574  0.73033   0.6998
## Neg Pred Value         0.9594   0.9064   0.9543  0.92284   0.9381
## Prevalence             0.2845   0.1935   0.1744  0.16391   0.1838
## Detection Rate         0.2571   0.1157   0.1369  0.09699   0.1337
## Detection Prevalence   0.3256   0.1698   0.1807  0.13281   0.1911
## Balanced Accuracy      0.9039   0.7656   0.8660  0.77446   0.8286
```

Random Forest

```
# Fitting the model
modFitRF <- randomForest(classe ~ ., data=subTrainingDF, method="class")

# Perform prediction
predictRF <- predict(modFitRF, subTestingDF, type = "class")
```

```
confusionMatrix(predictRF, subTestingDF$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    2    0    0    0
##          B    0 1516    3    0    0
##          C    0    0 1364    9    0
##          D    0    0    1 1277    4
##          E    0    0    0    0 1438
##
## Overall Statistics
##
##                Accuracy : 0.9976
##                  95% CI : (0.9962, 0.9985)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9969
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9987   0.9971   0.9930   0.9972
## Specificity            0.9996   0.9995   0.9986   0.9992   1.0000
## Pos Pred Value         0.9991   0.9980   0.9934   0.9961   1.0000
## Neg Pred Value         1.0000   0.9997   0.9994   0.9986   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1932   0.1738   0.1628   0.1833
## Detection Prevalence   0.2847   0.1936   0.1750   0.1634   0.1833
## Balanced Accuracy      0.9998   0.9991   0.9978   0.9961   0.9986
```

Results

It is possible to observe that the **Random Forest Model** has the highest accuracy rate **0.9972**. As a result, this model is used to answer the quiz question.

```
# Performing prediction
predictSubmission <- predict(modFitRF, testingDF, type="class")
predictSubmission
```

```
##  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
```