Peer-graded Assignment: Prediction Assignment Writeup

Olusanya Oluwole

1/22/2021

Contents

Introduction **1**

Data Source Data Cleaning and Exploration **1**

Exploratory Analysis with Correlation 2

Prediction Models **3**

Decision Tree 3

Generalized Boosted Model 4

Random Forest 4

Linear Discriminant Analysis (LDA) 5

Class Prediction for `testing` Dataset **5**

Introduction

The practical machine learning assignment intend to demonstrated how to integrate knowledge gain during the enter course course.

"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." The goal of this project is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise (classe variable).

Data Source Data Cleaning and Exploration

Our data is source from http://groupware.les.inf.puc-rio.br/har

- Training: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
- Testing: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```
#install.packages("parallel")
#install.packages("doParallel")
library(tidyverse)
library(caret)
library(rattle)
library(parallel)
library(doParallel)
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training <- read_csv(trainUrl)
testing <- read_csv(testUrl)
```

First, let remove near-zero variance predictor that will not contribute to our prediction from our training and testing data using `nearZeroVar` fuction.

```
# training dataset
nzvTraining <- nearZeroVar(training)
training <- training[, -nzvTraining]

# testing dataset
nzvtesting <- nearZeroVar(testing)
testing <- testing[, -nzvtesting]
```

There are still butch of `NAs` that can be removed as follow.

```
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
```

Variables `X1`, `user_name`, `raw_timestamp_part_1`, `raw_timestamp_part_2` and `cvtd_timestamp` are not going to be useful for our prediction. These will be remove as follow.

```
training <- select(training, -c(1:5))
testing <- select(testing, -c(1:5))
dim(training); dim(testing)
```

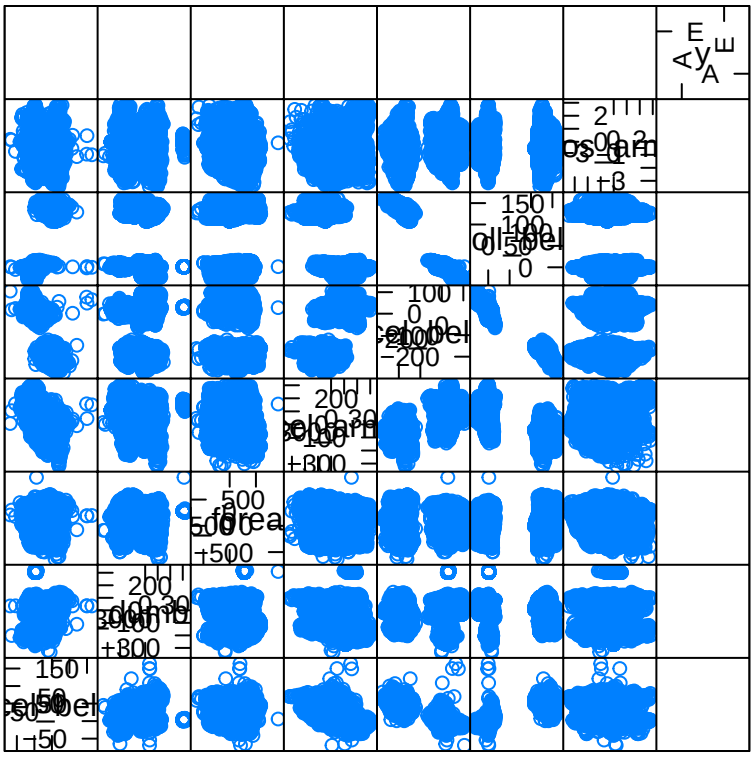
```
## [1] 19622    54
```

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

Exploratory Analysis with Correlation

Let take a look at how some of the variables are related to `classe` variable with `featurePlot`.

```
featurePlot(x=training[, c("accel_belt_y", "accel_dumbbell_z", "accel_forearm_y",
                           "accel_arm_y", "accel_belt_z", "roll_belt", "gyros_arm_y")],
            y = training$classe,
            plot="pairs")
```



Scatter Plot Matrix

The panel show some variables are correlated while others are not.

The dataset `testing` will be set aside for answering the **Quiz Portion** of this project. The `training` dataset will be further divided to `trainingData` (70%) and `testingData` (30%) for the portion of the first portion of this projected.

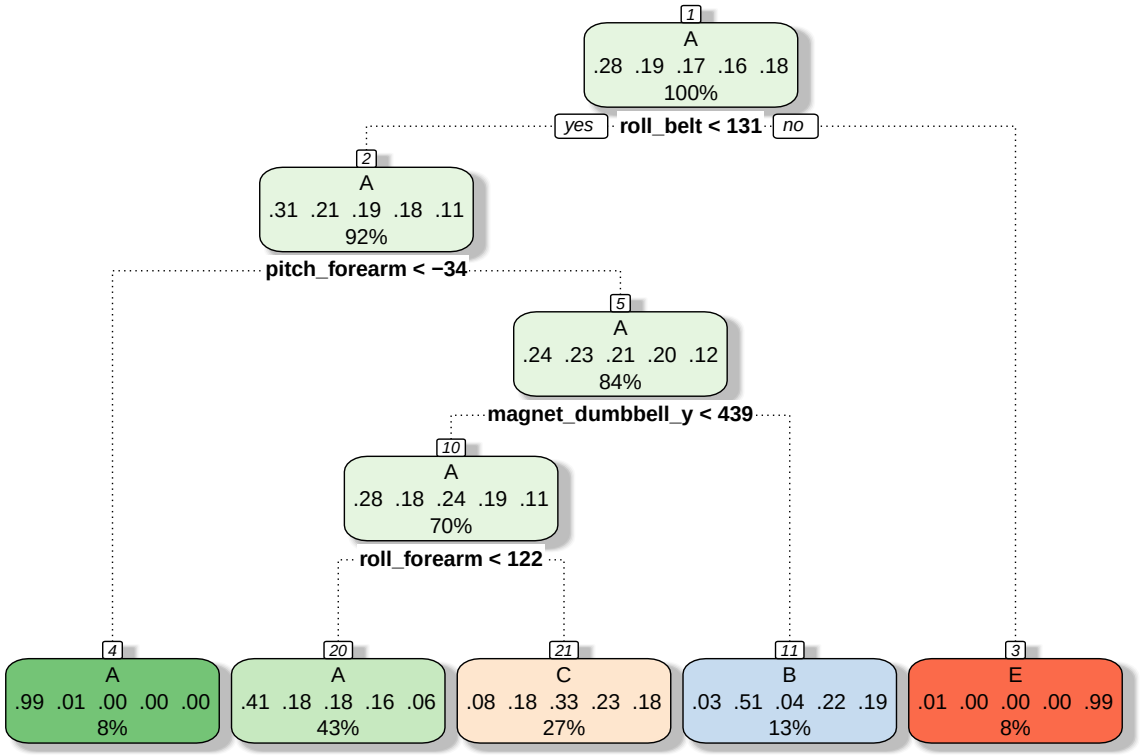
```
inTrain <- createDataPartition(training$classe, p = 0.7, list = FALSE)
trainingData <- training[inTrain,]
testingData <- training[-inTrain,]
```

Prediction Models

```
intervalStart <- Sys.time()
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)
fitControl <- trainControl(method = "cv",
number = 5,
allowParallel = TRUE)
```

Decision Tree

```
set.seed(12345)
treeMod <- train(classe ~ .,method="rpart",data=trainingData)
fancyRpartPlot(treeMod$finalModel)
```




```
treePredict <- predict(treeMod, newdata=testingData)
treeConfusionMatrix <- confusionMatrix(treePredict, factor(testingData$classe))
decisionTreeAccuracy <- treeConfusionMatrix$overall["Accuracy"]
```

Generalized Boosted Model

```
set.seed(12345)
system.time(gbmMod <- train(classe ~ .,method="gbm",data=trainingData, trControl = fitControl))
gbmPredict <- predict(gbmMod, newdata=testingData)
gbmConfusionMatrix <- confusionMatrix(gbmPredict, factor(testingData$classe))
boostingAccuracy <- gbmConfusionMatrix$overall["Accuracy"]
```

Random Forest

```
set.seed(12345)
system.time(rfMod <- train(classe ~ .,method="rf",data=trainingData, trControl = fitControl))
```

```
rfPredict <- predict(rfMod,newdata=testingData)
rfConfusionMatrix <- confusionMatrix(rfPredict, factor(testingData$classe))
randomForestAccuracy <- rfConfusionMatrix$overall["Accuracy"]
```

Linear Discriminant Analysis (LDA)

```
set.seed(12345)
system.time(ldaMod <- train(classe ~ .,method="lda",data=trainingData, trControl = fitControl))
ldaPredict <- predict(ldaMod, newdata=testingData)
ldaConfusionMatrix <- confusionMatrix(ldaPredict, factor(testingData$classe))
ldaAccuracy <- ldaConfusionMatrix$overall["Accuracy"]
```

```
accuracyTable <- tibble(decisionTreeAccuracy, boostingAccuracy,
                        randomForestAccuracy, ldaAccuracy)
accuracyTable
```

```
## # A tibble: 1 x 4
##   decisionTreeAccuracy boostingAccuracy randomForestAccuracy ldaAccuracy
##                  <dbl>            <dbl>                <dbl>       <dbl>
## 1                0.488            0.991                0.998       0.711
```

Of the four prediction method use on the training dataset, random forest is the most accurate with score of approximately 99%.

Class Prediction for `testing` Dataset

Random forest gives most accurate prediction on the test portion of the training dataset. Therefore, it will be use to predict class of testing dataset that will be use to answer the course project quiz portion.

```
rfPredict2 <- predict(rfMod,newdata=testing)
rfPredict2
```

```
##  [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
```