PMLW4_Prediction

Greg Ricci

3/5/2020

Practical Machine Learning Course Project Week 4

Note : The Background, Submission and Data requirments where copied from the Coursera's course - Practical Machine Learning Course Week4 assignment page, as part of the Specialization in Data Science. The document/write-up was produced utlizing RStudio and the knitr functions compiled to be published in html and/or pdf format.

Background

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.

Submission Requirment

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.

Peer Review Portion

- Your submission for the Peer Review portion should:
    - consist of a link to a Github repo
    - with your R markdown and compiled HTML file describing your analysis.

Course Project Prediction Quiz Portion

- Apply your machine learning algorithm to the 20 test cases available in the test data above and
- submit your predictions in appropriate format to the Course Project Prediction Quiz for automated grading.

Reproducibility

Due to security concerns with the exchange of R code, your code will not be run during the evaluation by your classmates. Please be sure that if they download the repo, they will be able to view the compiled HTML version of your analysis.

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).

Synopsis of Cited Data Set - Weight Lifting Exercises Dataset +This human activity recognition (HAR) research has traditionally focused on discriminating between different activities, i.e. to predict "which" activity was performed at a specific point in time (like with the Daily Living Activities dataset above). The approach they propose for the Weight Lifting Exercises dataset is to investigate "how (well)" an activity was performed by the wearer. The "how (well)" investigation has only received little attention so far, even though it potentially provides useful information for a large variety of applications,such as sports training.

In this work they first define quality of execution and investigate **three aspects***that pertain to qualitative activity recognition: the problem of specifying correct execution, the automatic and robust detection of execution mistakes, and how to provide feedback on the quality of execution to the user. they tried out an on-body sensing approach, but also an "ambient sensing approach".*

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

```
+*Class A corresponds to the specified execution of the exercise, while the other 4 cl
asses correspond to common mistakes.*

+*Participants were supervised by an experienced weight lifter to make sure the execut
ion complied to the manner they were supposed to simulate.*

+*The exercises were performed by six male participants aged between 20-28 years, with
little weight lifting experience.*

+*They made sure that all participants could easily simulate the mistakes in a safe an
d controlled manner by using a relatively light dumbbell (1.25kg).*
```

The Training Data

The training data sets 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). + Velloso, E.; Bulling, A.;

Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

THE MODEL

How was the Model built

The goal of this project is to predict the manner in which they did the exercise. This is the “Classe” variable (the Outcome) in the training set.

The participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in 5 different manners (Class A,B,C,D,E): +Class A: exactly according to the specifications +Class B: throwing the elbows to the front +Class C: lifting the dumbbell only halfway up +Class D: lowering the dumbbell only halfway down +Class E: throwing the hips to the front of the body

+It was also noted the Class A corresponds to the *specified execution* of the exercise, while the other 4 classes correspond to *common mistakes*." + My prediction analytics are geared to maximizing the accuracy and minimizing the out-of-sample error. + All other available variables (after cleaning) will be used for prediction purposes. + Two models will be created and tested using *decision tree and random forest* algorithms. The model with the highest accuracy will be chosen as our final model.

Cross-validation

```
+Cross-validation was performed by sub-sampling the training data set randomly without
replacement into two sub-samples:
1) subTraining data (75% of the original Training data set) and
2) subTesting data (25%).
3) The models will be fitted on and tested on the subTraining data set.
4) The most accurate model will be determined and will be tested on the original Testi
ng data set.
```

Expected out-of-sample error

```
+ The expected out-of-sample error will correspond to the quantity:  1-accuracy in the
cross-validation data.
a) Accuracy is the proportion of correct classified observation over the total sample
in the subTesting data set.
b) Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. orig
inal testing data set).
+Thus, to infer that the expected value of the out-of-sample error will correspond to
the expected number of misclassified observations/total observations in the Test data
set, which is the quantity: 1-accuracy found from the cross-validation data set.
```

Reasons for my choices

```
+ The Outcome variable “classe” is an unordered factor variable
+ With an unordered factor variablr I can choose an error type as 1-accuracy.
+ A large sample size with N= 19622 in the Training data set which enabled Training sa
mple to be divided into subTraining and subTesting to allow cross-validation.
+ Also all missing values will be discarded as well as those features that are deemed
irrelevant.
+ Decision tree and random forest algorithms are good for detecting the features that
are important for classification.
+ Feature selection is inherent and not so necessary during the data preparation step
s. There is no feature selection section in this analysis.
```

Loading five librarys

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

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

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

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.5.3
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## 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
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

Download of the Data sets

```
trainUrl <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

trainFile <- "./data/pml-training.csv"

testFile <- "./data/pml-testing.csv"

if (!file.exists("./data")) {

dir.create("./data")

}

if (!file.exists(trainFile)) {

download.file(trainUrl, destfile=trainFile, method="curl")

}

if (!file.exists(testFile)) {

download.file(testUrl, destfile=testFile, method="curl")

}
```

Read the Data into useable form

Download the data from the data sources and then read the two csv files into two data frames.

```
trainRaw <- read.csv("./data/pml-training.csv")

testRaw <- read.csv("./data/pml-testing.csv")

dim(trainRaw)
```

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

```
dim(testRaw)
```

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

The training data set contains 19,622 observations and 160 variables, while the testing data set contains 20 observations and 160 variables. The “classe” variable in the training set is the Outcome to predict.

Clean the data

First - Clean the data to get rid of observations with missing and meaningless variables.

```
sum(complete.cases(trainRaw))
```

```
## [1] 406
```

Next, remove columns that contain **NA** missing values.

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

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

Next, remove columns that do not contribute to the accelerometer measurements.

```
classe <- trainRaw$classe

trainRemove <- grepl("^X|timestamp|window", names(trainRaw))

trainRaw <- trainRaw[, !trainRemove]

trainCleaned <- trainRaw[, sapply(trainRaw, is.numeric)]

trainCleaned$classe <- classe

testRemove <- grepl("^X|timestamp|window", names(testRaw))

testRaw <- testRaw[, !testRemove]

testCleaned <- testRaw[, sapply(testRaw, is.numeric)]
```

Next, notice that the cleaned training data set contains 19,622 observations and 53 variables, while the testing data set contains 20 observations and 53 variables. The “classe” variable is still in the cleaned training set.

Seperate the data

Next, bifurcate the cleaned training data set into a pure training data set (70%) and a validation data set (30%). Utilized the validation data set to conduct cross validation for next steps.

```
set.seed(22519) # For reproducibile purpose

inTrain <- createDataPartition(trainCleaned$classe, p=0.70, list=F)

trainData <- trainCleaned[inTrain, ]

testData <- trainCleaned[-inTrain, ]
```

Data modeling

I utlized a **Random Forest** algorithm to fit a predictive model for activity recognition because the RF algrorithm automatically selects important variables, correlates covariates and any outliers. Also, selected a **5-fold cross validation** when applying the algorithm.

```
controlRf <- trainControl(method="cv", 5)

modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=controlRf, ntree=2
50)

modelRf
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10991, 10988, 10989, 10991
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9909729  0.9885802
##   27    0.9914091  0.9891325
##   52    0.9849311  0.9809363
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

next, estimated the performance of the model on the validation data set.

```
predictRf <- predict(modelRf, testData)

confusionMatrix(testData$classe, predictRf)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    0    0    0    1
##          B    6 1129    4    0    0
##          C    0    0 1021    5    0
##          D    0    0   15  948    1
##          E    0    0    0    6 1076
## 
## Overall Statistics
## 
##                Accuracy : 0.9935          
##                  95% CI : (0.9911, 0.9954)
##     No Information Rate : 0.2853          
##     P-Value [Acc > NIR] : < 2.2e-16       
## 
##                   Kappa : 0.9918          
## 
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   1.0000   0.9817   0.9885   0.9981
## Specificity            0.9998   0.9979   0.9990   0.9968   0.9988
## Pos Pred Value         0.9994   0.9912   0.9951   0.9834   0.9945
## Neg Pred Value         0.9986   1.0000   0.9961   0.9978   0.9996
## Prevalence             0.2853   0.1918   0.1767   0.1630   0.1832
## Detection Rate         0.2843   0.1918   0.1735   0.1611   0.1828
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9981   0.9989   0.9903   0.9926   0.9984
```

```
accuracy <- postResample(predictRf, testData$classe)

accuracy
```

```
##  Accuracy     Kappa 
## 0.9935429 0.9918320
```

```
oose <- 1 - as.numeric(confusionMatrix(testData$classe, predictRf)$overall[1])

oose
```

```
## [1] 0.006457094
```

Results determined where estimated accuracy of the model is 99.35% and the estimated out-of-sample error is 0.64%.

Predicting outcomes for the Quize Portion of the assingment

Next, apply the model to the original testing data set downloaded from the data source. Instructed to apply the machine learning algorithm to the 20 test cases available in the test data.

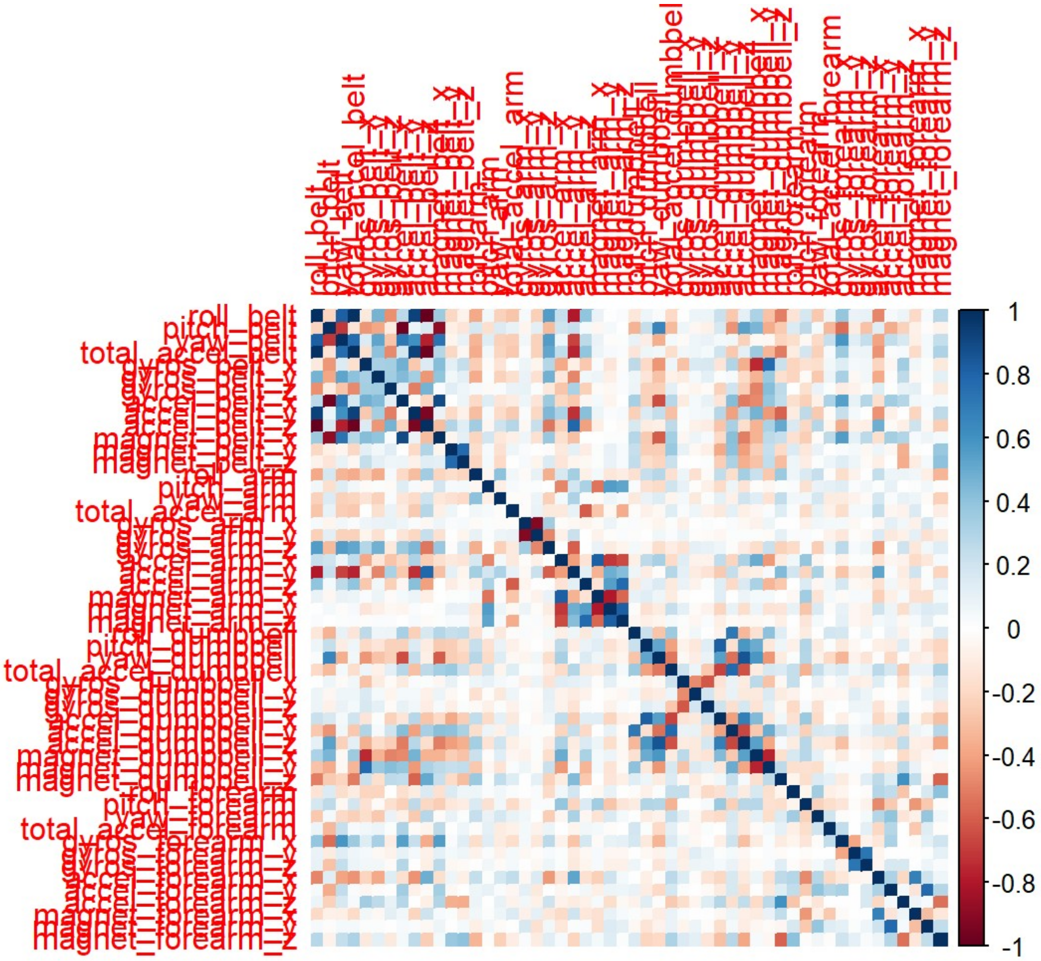
```
result <- predict(modelRf, testCleaned[, -length(names(testCleaned))])

result
```

```
##  [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
```

Diagrams created and utilized for analysis

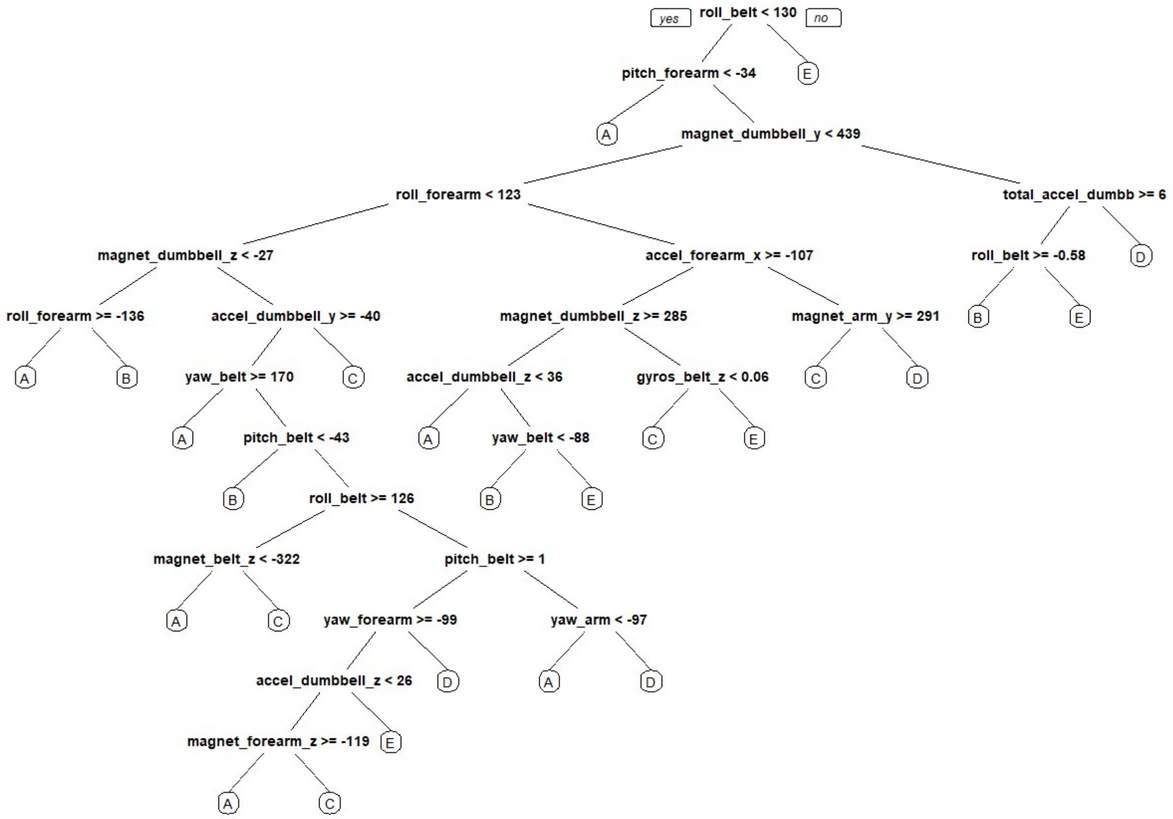
1. Correlation Matrix Visualization

```
corrPlot <- cor(trainData[, -length(names(trainData))])

corrplot(corrPlot, method="color")
```



2. Decision Tree Visualization

```
treeModel <- rpart(classe ~ ., data=trainData, method="class")

prp(treeModel) # fast plot
```



End of Document