Practical Machine Learning Course Project

Jessmae Zafra

December 10, 2018

Executive Summary

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.

This project aims to predict the manner in which the participants did the exercise. That is, topredict the "classe" variable in the training data coming from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. All dataset to be used in this project came from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har.

For this project, the developer employed 2 models: Classification Tree, and Random Forest.

Preprocessing

All the data and packages to be used in this project were downloaded and loaded first in R. Next, train dataset was partitoned to two subsets: subset_train and subset_test with subset_train having 60% of the original train data. Subset_train was used to build the model while the subset_test would be used to measure the model's accuracy on out-of-sample data.

```
library(caret)
library(ggplot2)
library(randomForest)
library(rpart)
library(rattle)

set.seed(102938)
train <- read.csv("pml-training.csv")
test <- read.csv("pml-testing.csv")
train <- train[,-(1:5)]
test <- test[,-(1:5)]
inbuild <-createDataPartition(y=train$classe, p=0.6, list=FALSE)
subset_train <- train[inbuild,]
subset_test <- train[-inbuild,]
dim(subset_train)
```

```
## [1] 11776   155
```

The dataset used in building the model (train) was made up of 155 variables and 11776 observations.

Before model building, dataset must be cleaned. In this case, near-zero covariates were removed, as well as variables with 5% missing values. Only 54 variables were left.

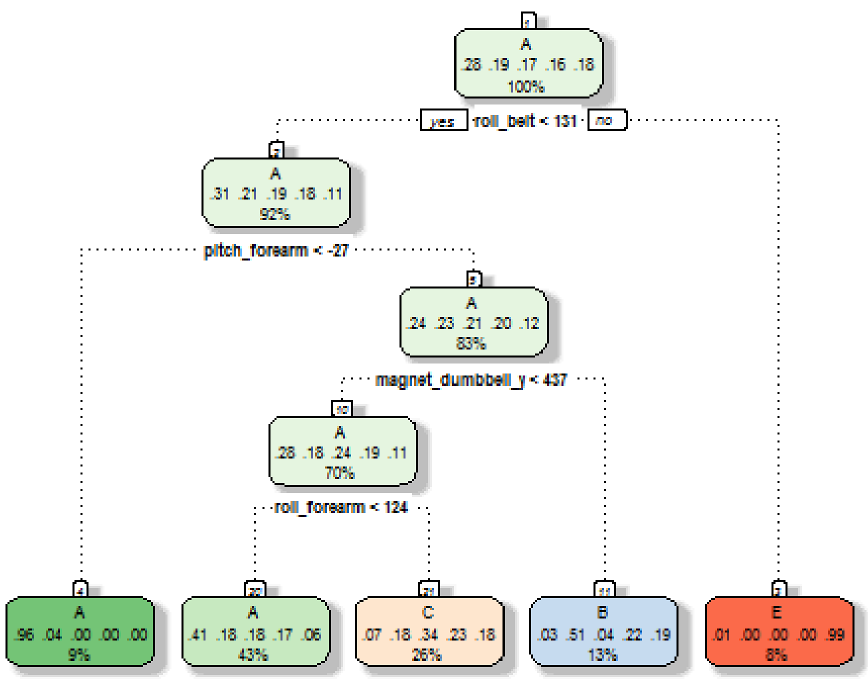
```
nzv <- nearZeroVar(subset_train,saveMetrics=FALSE)
subset_train <- subset_train[,-nzv]
msng <- which(colMeans(is.na(subset_train)) >= 0.05)
subset_train <- subset_train[,-msng]
```

Model Building

Two preliminary models were built in this project, namely: Classification Tree, and Random Forest. The final model would be selected based on its accuracy on the test set (subset_test).

- Classification Tree

```
model_CT <- train(classe ~ ., method="rpart", data=subset_train)
fancyRpartPlot(model_CT$finalModel)
```



Rattle 2018-Dec-12 00:01:19 Jessmae Zafra

- Random Forest Model

```
set.seed(102938)
model_RF <- randomForest(classe~., data=subset_train)
model_RF
```

```
## 
## Call:
##  randomForest(formula = classe ~ ., data = subset_train) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
## 
##         OOB estimate of  error rate: 0.42%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3348    0    0    0    0 0.000000000
## B    9 2268    2    0    0 0.004826678
## C    0   11 2043    0    0 0.005355404
## D    0    0   23 1906    1 0.012435233
## E    0    0    0    3 2162 0.001385681
```

Model Accuracy

To select the best model in predicting classe, accuracy of each model on the test set were computed. However, prior model application, the test set should were also cleaned the same way as the train dataset.

```
subset_test <- subset_test[,-nzv]
subset_test <- subset_test[,-msng]
```

- Classification Tree

```
pred_CT <- predict(model_CT, newdata = subset_test)
confusionMatrix(subset_test$classe, pred_CT)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2013   43  172    0    4
##          B  693  491  334    0    0
##          C  650   49  669    0    0
##          D  571  236  479    0    0
##          E  196  195  402    0  649
## 
## Overall Statistics
##                                          
##                Accuracy : 0.4871         
##                  95% CI : (0.476, 0.4983)
##     No Information Rate : 0.5255         
##     P-Value [Acc > NIR] : 1              
##                                          
##                   Kappa : 0.3292         
##  Mcnemar's Test P-Value : NA             
## 
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4882  0.48422  0.32539       NA  0.99387
## Specificity            0.9412  0.84968  0.87927   0.8361  0.88975
## Pos Pred Value         0.9019  0.32345  0.48904       NA  0.45007
## Neg Pred Value         0.6242  0.91735  0.78589       NA  0.99938
## Prevalence             0.5255  0.12924  0.26204   0.0000  0.08323
## Detection Rate         0.2566  0.06258  0.08527   0.0000  0.08272
## Detection Prevalence   0.2845  0.19347  0.17436   0.1639  0.18379
## Balanced Accuracy      0.7147  0.66695  0.60233       NA  0.94181
```

- Random Forest Model

```
pred_RF <- predict(model_RF, newdata = subset_test, type = "class")
confusionMatrix(subset_test$classe, pred_RF)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    5 1511    2    0    0
##          C    0    7 1361    0    0
##          D    0    0   19 1264    3
##          E    0    0    0    9 1433
##
## Overall Statistics
##
##                Accuracy : 0.9943
##                  95% CI : (0.9923, 0.9958)
##     No Information Rate : 0.2851
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9927
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9978   0.9954   0.9848   0.9929   0.9979
## Specificity            1.0000   0.9989   0.9989   0.9967   0.9986
## Pos Pred Value         1.0000   0.9954   0.9949   0.9829   0.9938
## Neg Pred Value         0.9991   0.9989   0.9968   0.9986   0.9995
## Prevalence             0.2851   0.1935   0.1761   0.1622   0.1830
## Detection Rate         0.2845   0.1926   0.1735   0.1611   0.1826
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9989   0.9971   0.9919   0.9948   0.9983
```

Since the Random Forest obtained higher accuracy (99.43%) compared to the Classification Tree (48.71%), Random Forest wwas considered as the final model to be used in predicting the classe of 20 observations.

Conclusion

In this section, the Random Forest model was used to predict the classe of the observations in the test set. The predicted classe were as follows:

```
pred_test <- predict(model_RF, newdata = test, type = "class")
pred_test

##  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
```