Practical Machine Learning-Prediction Assignment Writeup

Anish Raj

27/10/2019

Introduction

This project is being carried out in completion of the "Practical Machine Learning" Coursera course.

A dataset of measurement data has been provided by the course. The dataset is comprised of measurements of acceleration made by individuals who are carrying out one of five classes of physical activity. According to this project's instructions, the measurements are made using devices worn on the belt, forearm, arm, and a dumbbell.

Additional information the dataset is available here: http://groupware.les.inf.puc-rio.br/har

My task is to create a model that can predict the which class of activity is being done.

Loading relevant libraries

```
library(knitr)
```

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

```
library(caret)
```

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

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

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

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

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(rattle)
```

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

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart)
```

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

```
library(ggplot2)
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:rattle':
##
##     importance
```

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

```
set.seed(1234)
```

Data Loading & Cleaning

```
train <- read.csv("./pml-training.csv")
test <- read.csv("./pml-testing.csv")
dim(train);dim(test)
```

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

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

Cleaning data

Data cleaning using the following criterias : - Remove columns with (>95%) NAs - Remove columns with Near Zero variance - Remove columns with only information and no contribution

```
# remove columns that are mostly NA
NAFlag <- sapply(train, function(x) mean(is.na(x))) > 0.95
train <- train[, NAFlag==FALSE]
test <- test[, NAFlag==FALSE]
dim(train);dim(test)
```

```
## [1] 19622    93
```

```
## [1] 20 93
```

```
# remove columns with Nearly Zero Variance
NZV <- nearZeroVar(train)
train <- train[, -NZV]
test <- test[, -NZV]
dim(train);dim(test)
```

```
## [1] 19622    59
```

```
## [1] 20 59
```

```
# remove information only columns (columns 1 to 5)
train <- train[, -(1:5)]
test <- test[, -(1:5)]
dim(train);dim(test)
```

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

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

Partitioning the training data for cross validation (using 60% for training and 40% for validation)

```
set.seed(1234)
inTrain <- createDataPartition(train$classe, p=0.6, list=FALSE)
trainT <- train[inTrain,]
trainV <- train[-inTrain,]
dim(trainT);dim(trainV);dim(test)
```

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

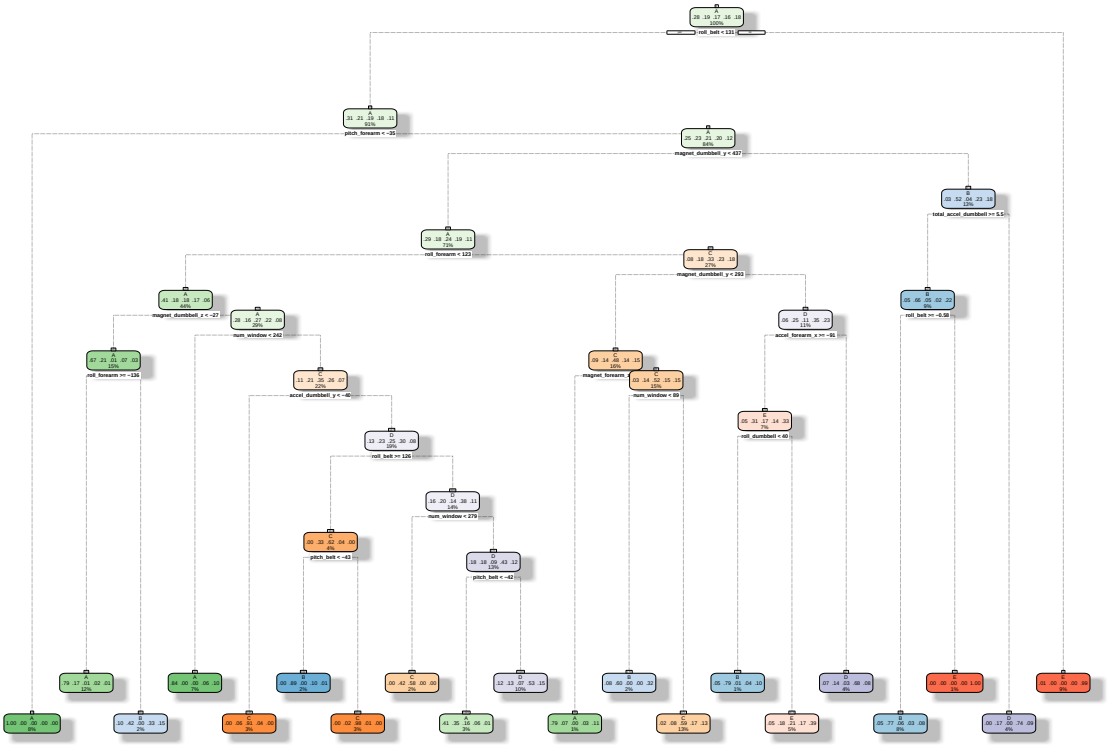
```
## [1] 7846   54
```

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

Prediction model buidling using Decision Tree, Random Forest & Generalized Boosted Model(GBM)

```
## Model using Decision Tree
set.seed(1234)
modFitDT  <- rpart(classe ~ ., data=trainT, method="class")
fancyRpartPlot(modFitDT)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2019−Oct−27 18:48:59 samsung

```
## prediction using validation set on Decison Tree model
prediction <- predict(modFitDT, newdata = trainV,type = "class")
confusionMatrix(prediction, trainV$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1973  239   57   83   75
##          B   93  844   53   92  148
##          C   20  174 1086  206  111
##          D  120  166   93  841  172
##          E   26   95   79   64  936
##
## Overall Statistics
##
##                Accuracy : 0.7239
##                  95% CI : (0.7139, 0.7338)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6501
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8840   0.5560   0.7939   0.6540   0.6491
## Specificity            0.9191   0.9390   0.9211   0.9160   0.9588
## Pos Pred Value         0.8129   0.6862   0.6800   0.6042   0.7800
## Neg Pred Value         0.9522   0.8981   0.9549   0.9311   0.9239
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2515   0.1076   0.1384   0.1072   0.1193
## Detection Prevalence   0.3093   0.1568   0.2035   0.1774   0.1529
## Balanced Accuracy      0.9015   0.7475   0.8575   0.7850   0.8039
```

```
## Model using Random Forest
set.seed(1234)
modFitRF <- randomForest(classe ~ ., data = trainT)
modFitRF
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = trainT)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.47%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3347    1    0    0    0 0.0002986858
## B   11 2265    3    0    0 0.0061430452
## C    0   16 2036    2    0 0.0087633885
## D    0    0   16 1913    1 0.0088082902
## E    0    0    0    5 2160 0.0023094688
```

```
## prediction using validation set on Random Forest model
prediction <- predict(modFitRF, newdata = trainV)
confusionMatrix(prediction, trainV$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    4    0    0    0
##          B    0 1512    5    0    0
##          C    0    2 1360   11    0
##          D    0    0    3 1274    2
##          E    0    0    0    1 1440
##
## Overall Statistics
##
##                Accuracy : 0.9964
##                  95% CI : (0.9948, 0.9976)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                  Kappa : 0.9955
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9960   0.9942   0.9907   0.9986
## Specificity            0.9993   0.9992   0.9980   0.9992   0.9998
## Pos Pred Value         0.9982   0.9967   0.9905   0.9961   0.9993
## Neg Pred Value         1.0000   0.9991   0.9988   0.9982   0.9997
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1927   0.1733   0.1624   0.1835
## Detection Prevalence   0.2850   0.1933   0.1750   0.1630   0.1837
## Balanced Accuracy      0.9996   0.9976   0.9961   0.9950   0.9992
```

```
## Model using Generalized Boosted Model(GBM)
set.seed(1234)
controlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
modFitGBM <- train(classe ~ .,method="gbm",data = trainT,verbose=FALSE,trControl = controlGBM)
modFitGBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 53 had non-zero influence.
```

```
## prediction using validation set on Generalized Boosted Model(GBM)
prediction <- predict(modFitGBM, newdata = trainV)
confusionMatrix(prediction, trainV$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2231   26    0    0    0
##          B    1 1480   19    3    3
##          C    0   11 1343   21    2
##          D    0    0    6 1261   15
##          E    0    1    0    1 1422
##
## Overall Statistics
##
##                Accuracy : 0.9861
##                  95% CI : (0.9833, 0.9886)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9824
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9996   0.9750   0.9817   0.9806   0.9861
## Specificity            0.9954   0.9959   0.9948   0.9968   0.9997
## Pos Pred Value         0.9885   0.9827   0.9753   0.9836   0.9986
## Neg Pred Value         0.9998   0.9940   0.9961   0.9962   0.9969
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1886   0.1712   0.1607   0.1812
## Detection Prevalence   0.2877   0.1919   0.1755   0.1634   0.1815
## Balanced Accuracy      0.9975   0.9854   0.9882   0.9887   0.9929
```

Model Selection & Prediction on test data provided

The accuracy of the 3 models are as follows : Decision Tree : 0.7239, Random Forest : 0.9964 , Generalized Boosted Model(GBM) : 0.9871

Random forest is chosen as the final model for prediction based on highest accuracy and for Random Forest model OOB estimate of error rate (using the train subset in training set) is 0.44% and Out of sample error (using the validaton subset in the training set) is 0.36%

Prediction on the final test data having 20 samples is as follows

```
predictiontest <- predict(modFitRF, newdata=test)
predictiontest
```

```
##  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
```