Practical Machine Learning - Course Assignment

Rick M

2/21/2020

Executive Summary

The aim of the project is to predict how well an exercise (in our project specifically is barbell lifts) is done according to a set of variables that have been derived using sensors applied on the body.

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The objective is to correctly predict the variable classe of the Test set. This variable indicates how well the exercise is performed. The valua A indicates that the exercise was well performed while the other letters (from B to E) respectively indicate that common mistakes has been done during the execution of the weightlifting.

First the datasets are loaded and only useful variables are considered. Then two different machine learning algorithms are applied to a subset of the training set and then tested to estimate the accuracy. Finally, the best model found is determined and is applied to the test set to predict the type of performance in doing the weightlifting of 20 instances.

Data Preparation

First, load the packages that are needed to process and read the data for the training data and perform cleaning tasks.

```
library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

## Loading required package: ggplot2

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.2

## 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(rpart)
library(curl)
library(lattice)
library(ggplot2)
library(rattle)

## Warning: package 'rattle' was built under R version 3.6.2

## Rattle: A free graphical interface for data science with R.
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

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

library(e1071)

## Warning: package 'e1071' was built under R version 3.6.2

URL<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train = read.csv(file=URL, na.strings=c("NA","#DIV/0!", ""))
train <- subset(train, select=-c(1:6))
train2<-train[,colSums(is.na(train)) == 0]
classe <- train2$classe
train2 <- train2[,sapply(train2,is.numeric)]
train2$classe <- classe; rm(classe)
dim(train2)

## [1] 19622    54

URL2<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test = read.csv(file=URL2, na.strings=c("NA","#DIV/0!", ""))
test <- subset(test, select=-c(1:6))
test2 <-test[,colSums(is.na(test)) == 0]
test2 <- test2[,sapply(test2,is.numeric)]
dim(test2)

## [1] 20 54
```

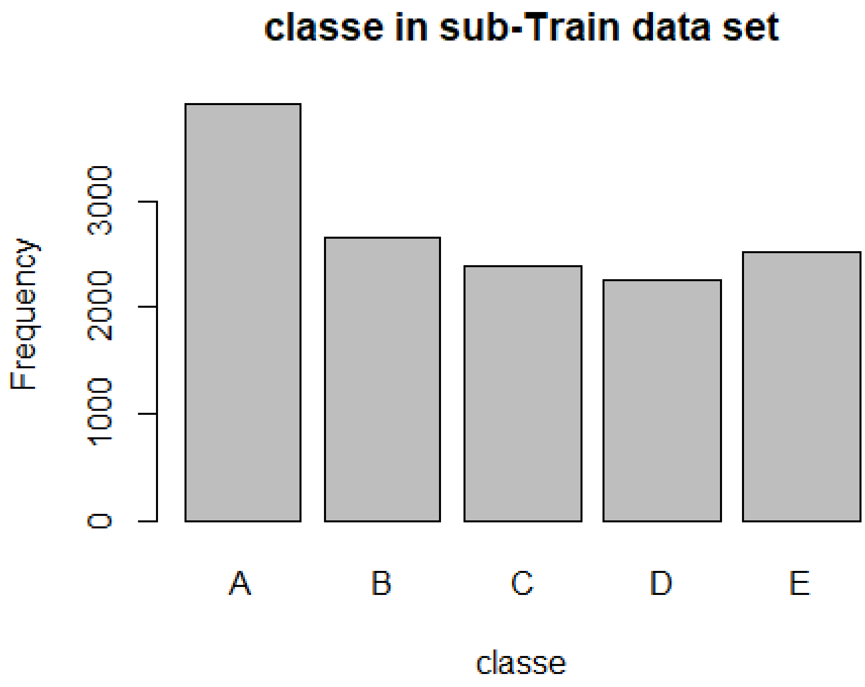
Then Identify set column name differences as it will be important in the decision stage.

```
trainCol <- names(train2)
testCol <- names(test2)
setdiff(trainCol, testCol)

## [1] "classe"

setdiff(testCol, trainCol)

## [1] "problem_id"
```

Cross-Validation

Next use the caret package to divide the trainig set in 70% to sub-training set and 20% to a sub-testing set.

```
set.seed(221)
samples <- createDataPartition(y=train2$classe, p=0.7, list=FALSE)
sTrain <- train2[samples, ]
sTest <- train2[-samples, ]
plot(sTrain$classe,  main="classe in sub-Train data set", xlab="classe",
ylab="Frequency")
```

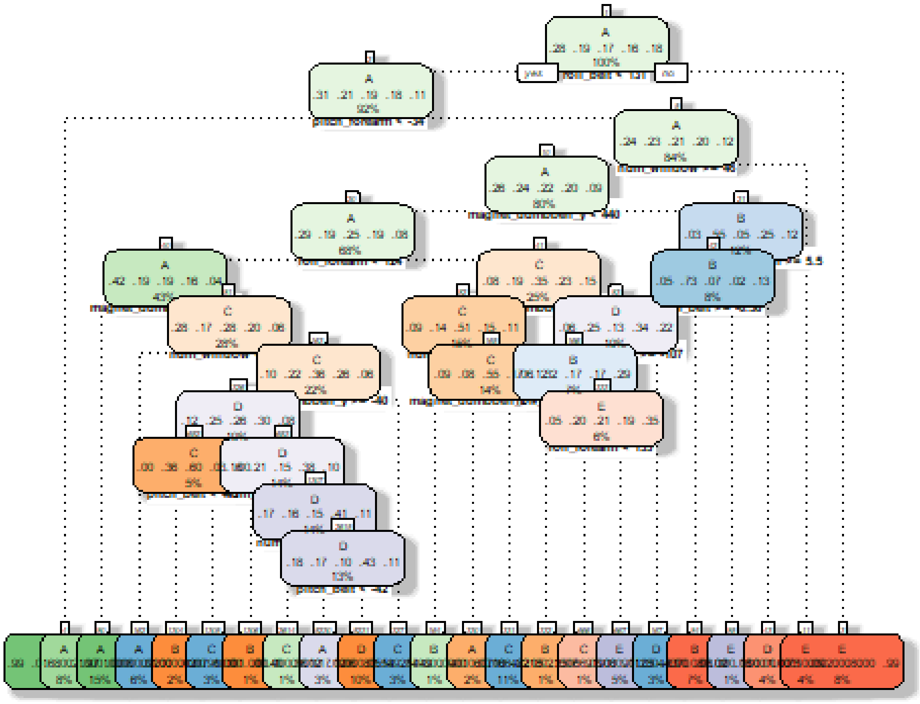


Modeling

Two models are ran and compared, selecting the ‘classe’ variable as the outcome, this variables has 5 levels (sitting-down, standing-up, standing, walking, and sitting) collected on 8 hours of activities of 4 healthy subjects.

First Model: Decision tree model

```
set.seed(221)
modelTree <- rpart(classe ~ ., data=sTrain, method="class")
predicTree <- predict(modelTree, sTest, type = "class")

fancyRpartPlot(modelTree,cex=0.4)

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



Rattle 2020-Feb-21 16:56:06 Rick

Next, test the results on the sub-testing data set for this first model:

```
confusionMatrix(predicTree, sTest$classe)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1508  256   47  102   52
##          B   43  620   30   24   25
##          C   11   68  862  144   66
##          D   98  149   59  595  137
##          E   14   46   28   99  802
##
## Overall Statistics
##
##                Accuracy : 0.7455
##                  95% CI : (0.7341, 0.7565)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6765
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9008   0.5443   0.8402   0.6172   0.7412
## Specificity            0.8915   0.9743   0.9405   0.9100   0.9611
## Pos Pred Value         0.7674   0.8356   0.7489   0.5732   0.8109
## Neg Pred Value         0.9577   0.8991   0.9654   0.9239   0.9428
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2562   0.1054   0.1465   0.1011   0.1363
## Detection Prevalence   0.3339   0.1261   0.1956   0.1764   0.1681
## Balanced Accuracy      0.8962   0.7593   0.8903   0.7636   0.8511
```

Second Model: Random Forest model

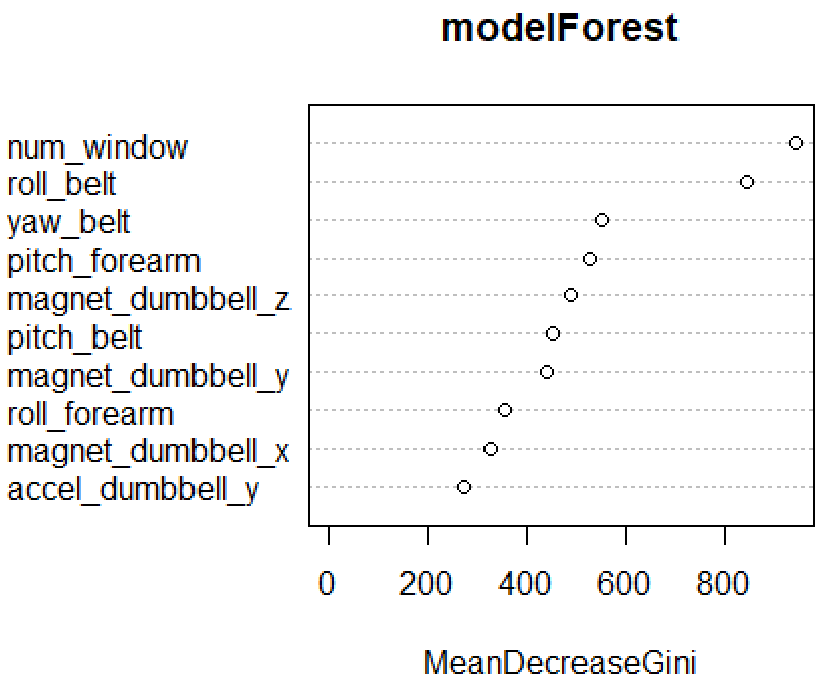
```
set.seed(221)
modelForest <- randomForest(classe ~. , data=sTrain, method="class")
predicForest <- predict(modelForest, sTest, type = "class")
```

Next, test the results on the sub-testing data set for the second model

```
confusionMatrix(predicForest, sTest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1139    2    0    0
##          C    0    0 1024   10    0
##          D    0    0    0  954    1
##          E    0    0    0    0 1081
##
## Overall Statistics
##
##                Accuracy : 0.9978
##                  95% CI : (0.9962, 0.9988)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9972
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   0.9981   0.9896   0.9991
## Specificity            1.0000   0.9996   0.9979   0.9998   1.0000
## Pos Pred Value         1.0000   0.9982   0.9903   0.9990   1.0000
## Neg Pred Value         1.0000   1.0000   0.9996   0.9980   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1935   0.1740   0.1621   0.1837
## Detection Prevalence   0.2845   0.1939   0.1757   0.1623   0.1837
## Balanced Accuracy      1.0000   0.9998   0.9980   0.9947   0.9995

VarImport <- varImp(modelForest)
varImpPlot(modelForest,n.var = 10)
```



Model Selection

Based in the results, the Random Forest model due to it being a better predictor to the classe variable. Tha Accuracy in Random Forest Model was 0.9978 versus 0.7455 in the Decision Tree Model.

Submission (Prediction)

Finally, predict 20 values in the testing data set using the Random Forest Model to predict ‘Class’ for each ‘problem_id’

```r
predictF <-predict(modelForest, type="class", newdata = test2[,-
which(names(test) %in% "problem_id")])
t(data.frame(problem_id = test2$problem_id, prediction = predictF))
```

```
##            1    2    3    4    5    6    7    8    9    10   11   12   13  
## problem_id " 1" " 2" " 3" " 4" " 5" " 6" " 7" " 8" " 9" "10" "11" "12"
"13"
## prediction "B"  "A"  "B"  "A"  "A"  "E"  "D"  "B"  "A"  "A"  "B"  "C"  "B" 
##            14   15   16   17   18   19   20  
## problem_id "14" "15" "16" "17" "18" "19" "20"
## prediction "A"  "E"  "E"  "A"  "B"  "B"  "B"
```