RHays Prediction Assignment

RHays

2/9/2021

Introduction

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 (see the section on the Weight Lifting Exercise Dataset).

Data

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

Project Goal

The goal of my project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. I may use any of the other variables to predict with. I will create a report:

- describing how I built my model
- how I used cross validation
- what I think the expected out of sample error is
- and why you made the choices you did

I will also use my prediction model to predict 20 different test cases.

Obtaining the Data

```
rm(list=ls())
library(lattice); library(ggplot2); library(caret); library(randomForest); library(rpart); library(rpar
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## 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
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.3
```

```
trainingdata <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testdata <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

download.file(trainingdata, destfile = "pml-training.csv")
download.file(testdata, destfile = "pml-testing.csv")

set.seed(100)

trainingdata <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testingdata <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))

trainingdata$classe <- as.factor(trainingdata$classe)


trainingdata[, 7:159] <- lapply(trainingdata[,7:159], as.numeric)
testingdata[, 7:159] <- lapply(testingdata[,7:159], as.numeric)
```

Data QC and Integrity

```
#names(testingdata)
#dim(trainingdata)
#dim(testingdata)
#summary(trainingdata)
#summary(testingdata)
#str(trainingdata)
#str(testingdata)
```

Remove cloumns with all NA data

Exclude non-relevant columns

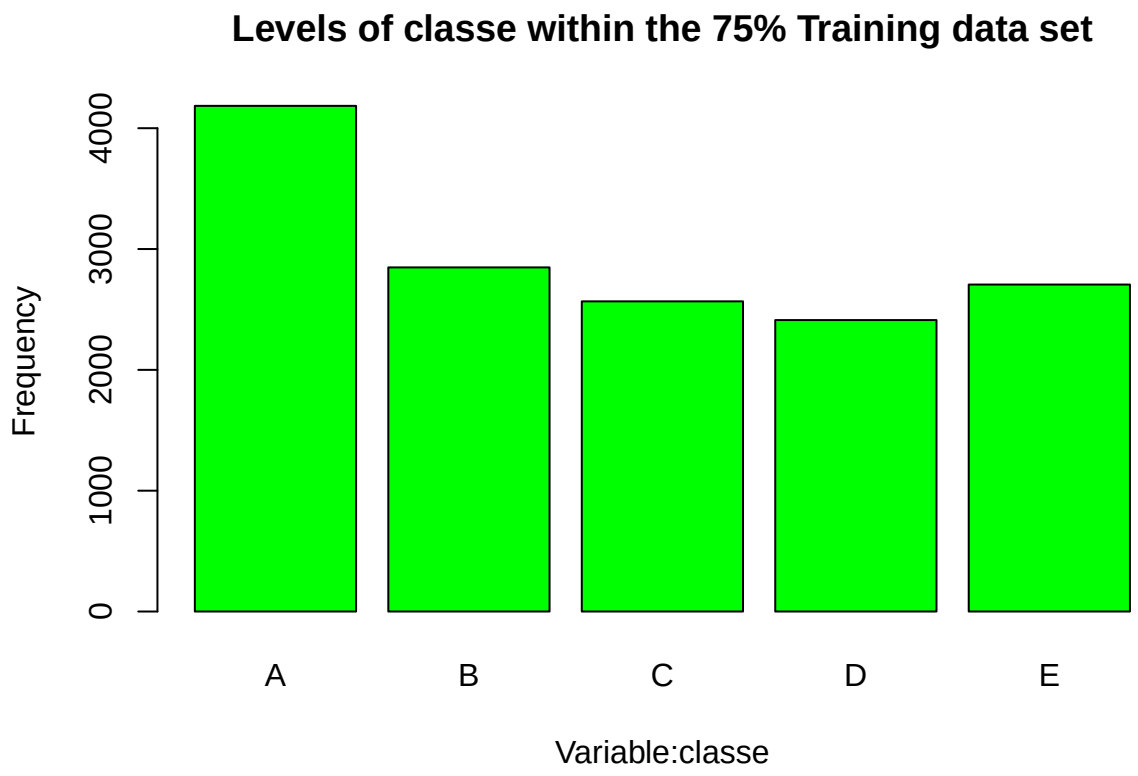
```
trainingdata <-trainingdata[,colSums(is.na(trainingdata)) == 0]
testingdata <-testingdata[,colSums(is.na(testingdata)) == 0]

trainingdata <-trainingdata[,-c(1:7)]
testingdata <-testingdata[,-c(1:7)]
```

Paritioing the data

```
trainset <- createDataPartition(y=trainingdata$classe, p=0.75, list=FALSE)
Train75 <- trainingdata[trainset, ]
TrainTest25 <- trainingdata[-trainset, ]
```

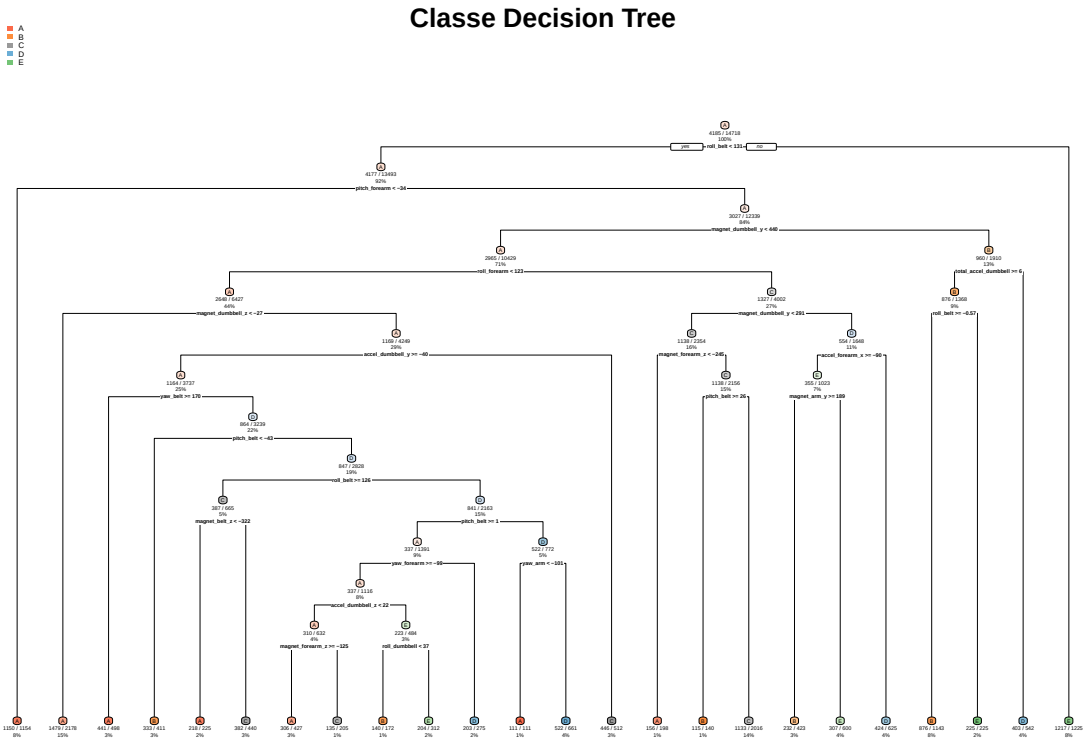
Plot Variable Levels

```
plot(Train75$classe, col="green", main="Levels of classe within the 75% Training data set", xlab="Varial
```



Model #1: Decision Tree

```
Model_1_Decision_Tree <- rpart(classe ~ ., data=Train75, method="class")
prediction1 <- predict(Model_1_Decision_Tree, TrainTest25, type = "class")

# Plot the Decision Tree
rpart.plot(Model_1_Decision_Tree, main="Classe Decision Tree", extra=102, under=TRUE, faclen=0)
```



Test results on our subTesting data set:

```
confusionMatrix(prediction1, TrainTest25$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1276  196   14   80   30
##          B   31  548   72   32   63
##          C   35   86  682  118  111
##          D   18   68   65  516   51
##          E   35   51   22   58  646
##
## Overall Statistics
##
##                Accuracy : 0.748
##                  95% CI : (0.7356, 0.7601)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.6797
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9147   0.5774   0.7977   0.6418   0.7170
## Specificity            0.9088   0.9499   0.9136   0.9507   0.9585
## Pos Pred Value         0.7995   0.7346   0.6609   0.7187   0.7956
## Neg Pred Value         0.9640   0.9036   0.9553   0.9312   0.9377
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2602   0.1117   0.1391   0.1052   0.1317
## Detection Prevalence   0.3254   0.1521   0.2104   0.1464   0.1656
## Balanced Accuracy      0.9118   0.7637   0.8556   0.7963   0.8378
```

Model #2: Random Forrest

```
Model_2_R_Forrest <- randomForest(classe ~. , data=Train75, method="class")

# Predicting:
prediction2 <- predict(Model_2_R_Forrest, TrainTest25, type = "class")

# Test results on TrainTest25 data set:
confusionMatrix(prediction2, TrainTest25$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    0    0    0    0
##          B    2  947    3    0    0
##          C    0    2  852    8    2
##          D    0    0    0  796    1
##          E    0    0    0    0  898
##
## Overall Statistics
##
##                Accuracy : 0.9963
##                  95% CI : (0.9942, 0.9978)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9954
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9979   0.9965   0.9900   0.9967
```

```
## Specificity            1.0000   0.9987   0.9970   0.9998   1.0000
## Pos Pred Value         1.0000   0.9947   0.9861   0.9987   1.0000
## Neg Pred Value         0.9994   0.9995   0.9993   0.9981   0.9993
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1931   0.1737   0.1623   0.1831
## Detection Prevalence   0.2841   0.1941   0.1762   0.1625   0.1831
## Balanced Accuracy      0.9993   0.9983   0.9968   0.9949   0.9983
```

Selecting the best model

Based on a comparison of the accuracy levels between the two models,the Random Forests model is the better model for determining how well the exercises were performed. The accuracy of the Random Forest model was 0.9967 (95% CI: (0.9947, 0.9981). While the Decision Tree model had an accuracy of 0.748 (0.7356, 0.7601). The expected out-of-sample error is estimated at 0.005, or 0.5%.

Prediction on original Testing Data Set

```
# predict outcome levels on the original Testing data set using Random Forest algorithm
Prediction <- predict(Model_2_R_Forrest, testingdata, type="class")
Prediction
```

```
##  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
```