Peer-Graded Assignment: Prediction Assignment Writeup

Nikolai Klassen

30 Oktober 2017

Abstract

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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

1. Data

Training data and testing data used for this project can be downloaded here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

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

Here i am loading the packages needed for my analsisis and while loading, i am dismissing the data which is NA, empty or #DIV0! by one consistent N/A.

```
library(caret); library(rpart);
```

```
## Warning: package 'caret' was built under R version 3.4.2
```

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

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

```
library(ggplot2);library(randomForest);library(rattle)
```

```
## Warning: package 'randomForest' was built under R version 3.4.2
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## Warning: package 'rattle' was built under R version 3.4.2
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Geben Sie 'rattle()' ein, um Ihre Daten mischen.
```

```
##
## Attaching package: 'rattle'

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

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

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

training <- read.csv(url(trainurl), na.strings = c("NA", "", "#DIV0!"))

testing <- read.csv(url(testnurl), na.strings = c("NA", "", "#DIV0!"))
```

After importing the data, we check both sets for consistency and see, that the variable classes is not included in the testing data.

```
samecolnames <- colnames(training) == colnames(testing)
colnames(training)[samecolnames == F]
```

```
## [1] "classe"
```

```
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
training <- training[,c(8:60)]
testing <- testing[c(8:60)]
```

For the sake, that our only our training set contains the classe data, we will generate another training and test set.

```
inTrain <-createDataPartition(training$classe , p = 0.7, list = FALSE)
training2 <- training[inTrain,]
testing2 <- training[-inTrain,]
```

Prediction Models

Here in the following I will analyse the data with the models we have learned in class:

a) Random Forest

```
set.seed(81888)
controlRF <- trainControl(method = "cv", number = 3, verboseIter = F)

modelRF <- train(classe~., data = training2, method = "rf", trControl = controlRF)

modelRF$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.71%
## Confusion matrix:
```

```
##      A    B    C    D    E class.error
## A 3900    2    2    0    2 0.001536098
## B   16 2634    8    0    0 0.009029345
## C    0   16 2374    6    0 0.009181970
## D    0    0   31 2218    3 0.015097691
## E    0    1    4    7 2513 0.004752475
```

```
predictRF <- predict(modelRF, newdata = testing2)
confMat <- confusionMatrix(predictRF, testing2$classe)

confMat
```

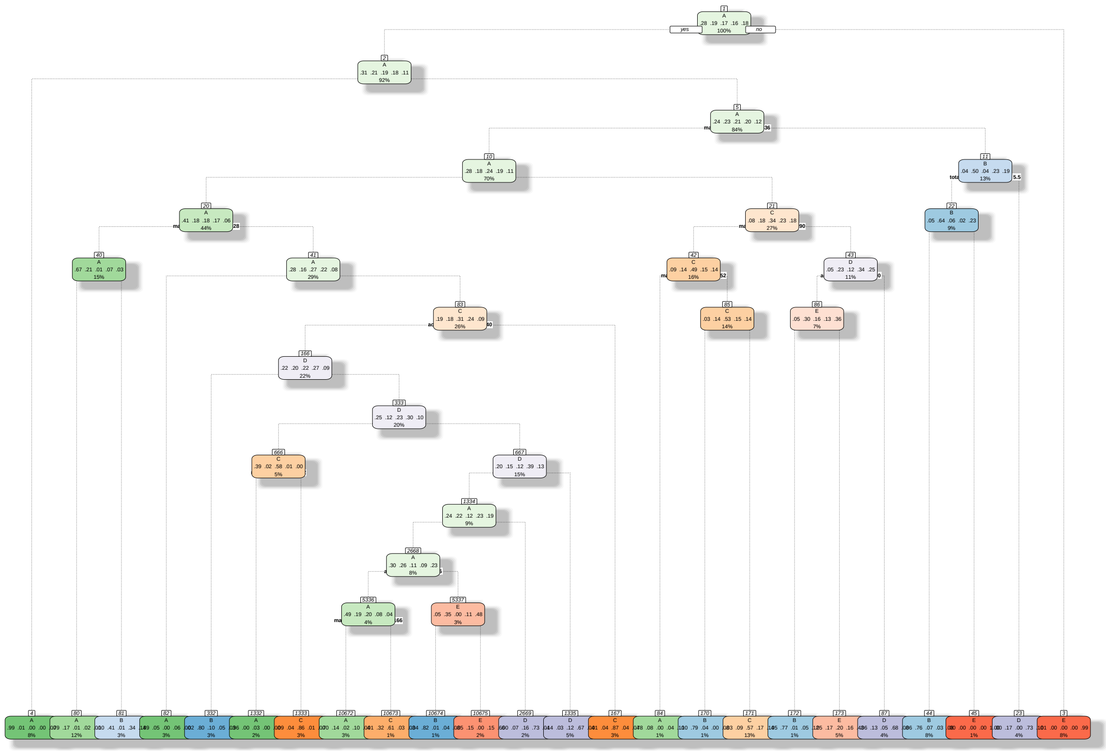
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1667   11    0    0    0
##          B    7 1125    5    1    1
##          C    0    3 1011   14    4
##          D    0    0    9  949    2
##          E    0    0    1    0 1075
##
## Overall Statistics
##
##                Accuracy : 0.9901
##                  95% CI : (0.9873, 0.9925)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9875
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9958   0.9877   0.9854   0.9844   0.9935
## Specificity            0.9974   0.9971   0.9957   0.9978   0.9998
## Pos Pred Value         0.9934   0.9877   0.9797   0.9885   0.9991
## Neg Pred Value         0.9983   0.9971   0.9969   0.9970   0.9985
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2833   0.1912   0.1718   0.1613   0.1827
## Detection Prevalence   0.2851   0.1935   0.1754   0.1631   0.1828
## Balanced Accuracy      0.9966   0.9924   0.9905   0.9911   0.9967
```

b) Decision Trees

```
set.seed(8188)
modelTree <- rpart(classe~., data = training2, method ="class")

fancyRpartPlot(modelTree, cex = 0.2)
```

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



Rattle 2017−Nov−03 13:57:32 Nikolai

Data Validation with our Model:

```r
predict(modelRF, testing)
```

```
##  [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
```