Practical Machine Learning Project - Human Activity Recognition

Michele Ronco

3/29/2019

Assignment

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).

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Downloding dataset

Load useful packages

```
library(caret)
```

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

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

```
library(rpart)
library(randomForest)
```

```
## 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
```

Training and testing dataset

```r
fileUrl1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(fileUrl1,destfile = "train.csv",method="curl")
training <- read.csv("train.csv")

fileUrl2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(fileUrl2,destfile = "test.csv",method="curl")
testing <- read.csv("test.csv")
```

Data cleaning and subsetting for cross validation

```r
training<-training[,colSums(is.na(training)) == 0]
training <- training[,c(8:93)]
testing <-testing[,colSums(is.na(testing)) == 0]
testing <- testing[,c(8:59)]
inTrain <- createDataPartition(y=training$classe,p=0.6,list=FALSE)
subtrain <- training[inTrain,]
subtest <- training[-inTrain,]
subsubtrain <- subtrain[,!is.na(sapply(subtrain,mean))]
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
subsubtrain $ classe <- subtrain $ classe
dim(subsubtrain)
```

```
## [1] 11776    53
```

Building my Machine Learning Model

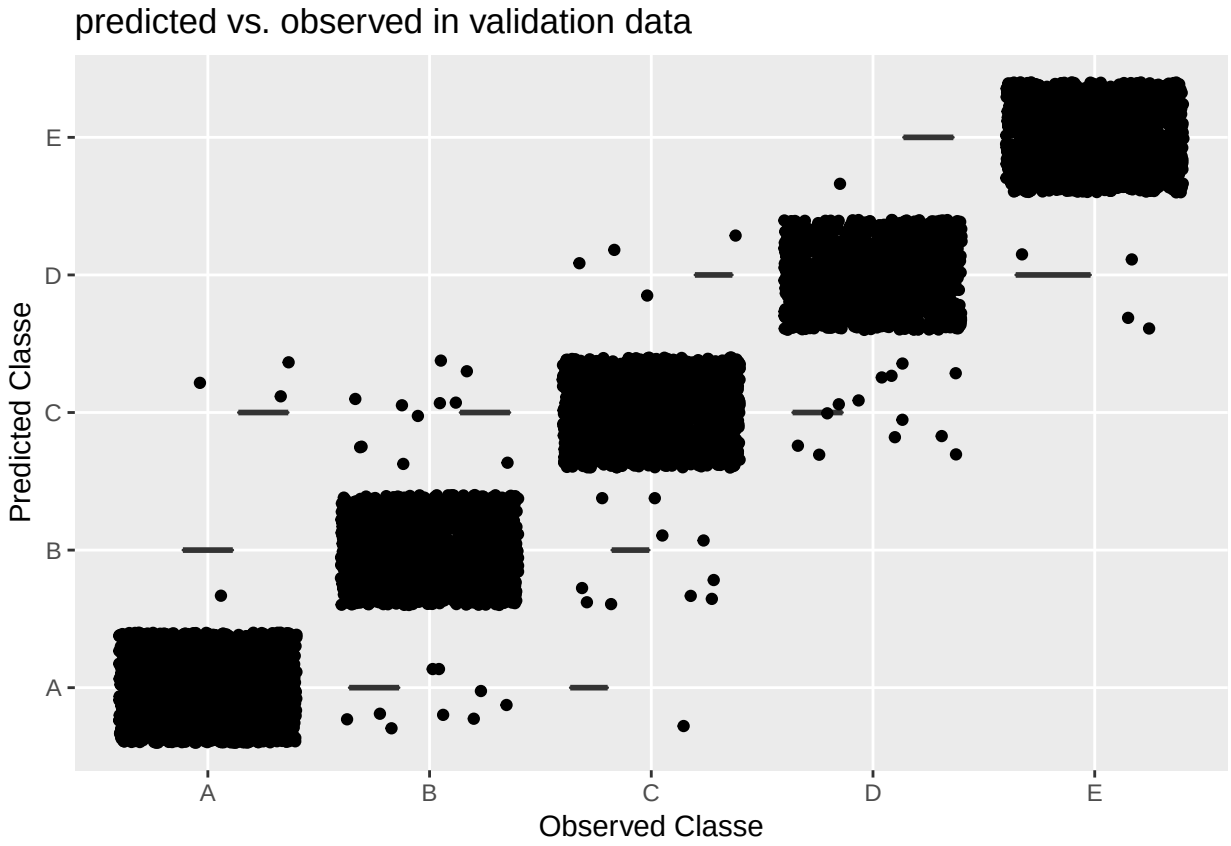
Classification And Regression With Random Forest

Second model:

```
model2 <- randomForest(classe ~ ., data=subsubtrain, method="class")
prediction2 <- predict(model2, subtest, type = "class")
```

Let's plot predictions obtained with model2 versus real values in test subset:

```
qplot(classe, prediction2, data=subtest, geom = c("boxplot", "jitter"), main = "predicted vs. observed
```



The curve is almost a straight line at 45 degrees and thus model2 has a very good performance. We then expect a small out-of-sample error.

Finally, we compute the confusion matrix to get accuracy and other statistic estimators:

```
confusionMatrix(prediction2,subtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2228    9    1    0    0
##          B    1 1498   10    0    0
##          C    3   11 1353   13    0
##          D    0    0    4 1272    4
##          E    0    0    0    1 1438
##
## Overall Statistics
##
##                Accuracy : 0.9927
##                  95% CI : (0.9906, 0.9945)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9908
##
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9868   0.9890   0.9891   0.9972
## Specificity            0.9982   0.9983   0.9958   0.9988   0.9998
## Pos Pred Value         0.9955   0.9927   0.9804   0.9938   0.9993
## Neg Pred Value         0.9993   0.9968   0.9977   0.9979   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2840   0.1909   0.1724   0.1621   0.1833
## Detection Prevalence   0.2852   0.1923   0.1759   0.1631   0.1834
## Balanced Accuracy      0.9982   0.9925   0.9924   0.9939   0.9985
```

Model2 performs much better, with a 99.39 % accuracy and 0.0061 out-of-sample error.

Predictions over 20 test cases with model2

```
prediction <- predict(model2, testing, 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
```

Comments

Only one model (the most accurate one) has been included in the html file in order to lighten the file size in terms of memory and make it visible to reviewers.