CourseProject - ML

VK

October 27, 2017

Instructions

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.

Review criterialess What you should submit

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.

Peer Review Portion

Your submission for the Peer Review portion should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-).

Course Project Prediction Quiz Portion

Apply your machine learning algorithm to the 20 test cases available in the test data above and submit your predictions in appropriate format to the Course Project Prediction Quiz for automated grading.

Reproducibility

Due to security concerns with the exchange of R code, your code will not be run during the evaluation by your classmates. Please be sure that if they download the repo, they will be able to view the compiled HTML version of your analysis.

Prediction Assignment Writeupless 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

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

SOLUTION

LOAD AND CLEAN THE DATA

```
library("caret")
```

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

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

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

```
## Warning: package 'ggplot2' was built under R version 3.4.1
```

```
#Download the data
if(!file.exists("pml-training.csv")){download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", destfile = "pml-training.csv")}

if(!file.exists("pml-testing.csv")){download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", destfile = "pml-testing.csv")}

trainData<- read.csv("pml-training.csv", sep=",", header=TRUE, na.strings = c("NA","",'#DIV/0!'))
testData<- read.csv("pml-testing.csv", sep=",", header=TRUE, na.strings = c("NA","",'#DIV/0!'))
dim(trainData)
```

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

```
dim(testData)
```

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

```
trainData <- trainData[,(colSums(is.na(trainData)) == 0)]
dim(trainData) # remove cols with missing val.
```

```
## [1] 19622    60
```

```
testData <- testData[,(colSums(is.na(testData)) == 0)]
dim(testData)
```

```
## [1] 20 60
```

Now, pre-process the data

```
num_Index <- which(lapply(trainData, class) %in% "numeric")

pre_process_Mdl <-preProcess(trainData[,num_Index],method=c('knnImpute', 'cen
ter', 'scale'))
pre_trainData <- predict(pre_process_Mdl, trainData[,num_Index])
pre_trainData$classe <- trainData$classe

pre_testData <-predict(pre_process_Mdl,testData[,num_Index])
```

Make predictions relevant by removing trivial data - to use only where relevant

```
#trivia_data <- nearZeroVar(pre_trainData,saveMetrics=TRUE)
#pre_trainData <- pre_trainData[,trivia_data$trivia_data==FALSE]
#trivia_data <- nearZeroVar(pre_testData,saveMetrics=TRUE)
#pre_testData <- pre_testData[,trivia_data$trivia_data==FALSE]
```

Model validation and Training We will use a 75% observation dataset to train our model.. and use random forest with 5 folds

```
set.seed(12341234)
index_Train<- createDataPartition(pre_trainData$classe, p=3/4, list=FALSE)
training <- pre_trainData[index_Train, ]
validation <- pre_trainData[-index_Train, ]
dim(training); dim(validation)
```

```
## [1] 14718    28
```

```
## [1] 4904   28
```

```
library(randomForest)
```

```
## 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
```

```
modelFit_RandF <- randomForest(classe~., data = training)
modelFit_RandF
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 0.54%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4179    6    0    0    0 0.001433692
## B   12 2828    7    1    0 0.007022472
## C    0    8 2541   16    2 0.010128555
## D    0    0   19 2391    2 0.008706468
## E    0    0    2    4 2700 0.002217295
```

Apply to test data set

```
predict_by_RF <- predict(modelFit_RandF, validation)
cM <- confusionMatrix(validation$classe, predict_by_RF)
cM$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    2    0    0    0
##          B    1  947    1    0    0
##          C    0    8  845    2    0
##          D    0    0    7  795    2
##          E    0    1    1    2  897
```

```
accuracy <- postResample(validation$classe, predict_by_RF)
modAccuracy <- accuracy[[1]]
modAccuracy
```

```
## [1] 0.9944943
```

```
out_of_sample_err <- 1 - modAccuracy
out_of_sample_err
```

```
## [1] 0.00550571
```

The estimated accuracy of the model is 99.5% while the estimated out-of-sample error is 0.5%

Now, applying to the 20 test case provided

```
prediction_final <- predict(modelFit_RandF, pre_testData)
prediction_final
```

```
##  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
```

Our results, as above. thanks.