Practical Machine Learning Assignment

Okbwoy

Saturday, November 21, 2015

1. Coursera Instructions

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://groupware.les.inf.puc-rio.br/har (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 (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (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.

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.

1. Your submission 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 :-).
2. You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.

2. Building a prediction model

Install all required R libraries

All necessary packages have been pre-installed

```
library(ggplot2)
library(Rcpp)
library(caret)
```

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

```
library(kernlab)
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(gridExtra)
library(ISLR)
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:Hmisc':
##
##     combine
```

Download and load testing and training files

```
## Download training and testing .csv file
## Load .csv files into 2 data frames
fileUrl <- 'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
download.file(fileUrl, destfile = 'pml-training.csv', method = 'auto')
pmltraining <- read.csv("pml-training.csv ")

##fileUrl <- 'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'
download.file(fileUrl, destfile = 'pml-testing.csv', method = 'auto')
pmltesting <- read.csv("pml-testing.csv ")
```

Data cleaning

Prior to any analysis, the first tasks consist in cleaning the data set: 1. First, let's remove the fisrt 7 columns that clearly cannot be used as predictors 2. Second, let's get rid of all near zero covariates 3. Lastly, let's remove predictors containing more than 50% NA

```
## Set Seed for reproductibility
set.seed(12345)

### Remove first seven columns
pmltraining <- subset(pmltraining, select=-c(X, user_name, raw_timestamp_part_1, raw_timestamp_part
_2, cvtd_timestamp, new_window, num_window))

## Remove all near zero covariates
nzv <- nearZeroVar(pmltraining, saveMetrics=TRUE)
pmltraining <- pmltraining[,nzv$nzv==FALSE]

## Use sapply to remove NAs
datatoremove <- sapply(colnames(pmltraining), function(x) if(sum(is.na(pmltraining[, x])) > 0.50*nr
ow(pmltraining))    {return(TRUE)
}else{
  return(FALSE)
}
)

pmltraining <- pmltraining[, !datatoremove]
```

Let's see how many 'good' predictors we are now left with:

```
dim(pmltraining)
```

```
## [1] 19622    53
```

We have 53 predictors to use in our prediction models

Partioning training set

The training data set will be partionned into two data sets, 70% myTraining, 30% myTesting:

```
## Use createDataPartition from the Caret Package to quickly split the data set
## classe being the outcome
inTrain <- createDataPartition(pmltraining$classe, p=0.7, list=FALSE)
myTraining <- pmltraining[inTrain, ]
myTesting <- pmltraining[-inTrain, ]

## Let's verify that the 70/30 split has been applied:
dim(myTraining)
```

```
## [1] 13737    53
```

```
dim(myTesting)
```

```
## [1] 5885   53
```

Using Machine Learning algorithms for prediction

After running a few models offline, such as General Linear Model or Decision Trees, I elected to use the Random Forest model. While GLM or DT provided decent in-sample and out-of-sample errors, Random Forest proved to be the most accurate model:

```
## Let's run the random forest model
modelfit <- randomForest(classe ~. , data=myTraining)

# Let's apply this model to myTesting data set
predictions <- predict(modelfit, myTesting, type = "class")

## Use confusion Matrix to test results:
confusionMatrix(predictions, myTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   11    0    0    0
##          B    1 1124    8    0    0
##          C    0    4 1018   13    0
##          D    0    0    0  951    4
##          E    0    0    0    0 1078
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9868   0.9922   0.9865   0.9963
## Specificity            0.9974   0.9981   0.9965   0.9992   1.0000
## Pos Pred Value         0.9935   0.9921   0.9836   0.9958   1.0000
## Neg Pred Value         0.9998   0.9968   0.9984   0.9974   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1910   0.1730   0.1616   0.1832
## Detection Prevalence   0.2862   0.1925   0.1759   0.1623   0.1832
## Balanced Accuracy      0.9984   0.9925   0.9944   0.9929   0.9982
```

We can see that accuracy for Random Forest model was 99.30%, with a [99.30% - 99.70%] 95% CI. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set, which in our case translates to an expected out-of-sample error estimated at 0.70%.

Since our Test data set contains 20 cases, an accuracy above 99% on our cross-validation data almost gurantess that very few, or none, of the test samples will be missclassified.

Submission

Lastly, let's predict the outcome on the original Testing set, using the previously built and cross-validated model. Based on above expected out-of-sample error, this likely to lead to a 20/20 submission… Fingers crossed…

```
Finalpredictions <- predict(modelfit, newdata = pmltesting, type="class")
```

Write files for submission using Coursera sample code

```r
## Write files as per Coursera instructions
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(Finalpredictions)
```