Coursera Machine learning final project

Giovanni Madejski

January 8, 2018

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 dumbbell 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 (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

The goal of this project is to create a model that can predict the manner in which exercises were completed.

Loading packages and data

We begin by loading the required libraries and downloading the datasets.

```
library(caret)
library(randomForest)

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

training <- read.csv(url(trainurl), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(testurl), na.strings=c("NA","#DIV/0!",""))
```

Next we clean the data by removing columns with over 50% missing values and removing unnecessary columns.

```
training <- training[,colSums(is.na(training))<nrow(training)*0.5]

unnecessary <- c('X','user_name','raw_timestamp_part_1','raw_timestamp_part_2','cvtd_timestamp',
'new_window')

training <- training[,!names(training) %in% unnecessary]
```

Building and Running the Model

We will use a random forest with 5 fold cross validation to model the data.

```
rf_model <- train(classe~., data = training, method = "rf", trControl=trainControl(method="cv",n
umber=5))
save(rf_model,file="rf_model.Rda")

print(rf_model)
```

```
## Random Forest
##
## 19622 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 15698, 15698, 15698, 15696, 15698
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9962287  0.9952295
##   27    0.9986240  0.9982596
##   53    0.9961269  0.9951006
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

This model appears to be very accurate. Lets look at the confusion matrix.

```
print(rf_model$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: 0.14%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5578    1    0    0    1 0.0003584229
## B    5 3790    2    0    0 0.0018435607
## C    0    5 3416    1    0 0.0017533606
## D    0    0    8 3206    2 0.0031094527
## E    0    0    0    2 3605 0.0005544774
```

The estimate of out of sample error rate is only .14% meaning this model should be very accurate.

predictions on the test data

We will now use our random forest model to make predictions on the test set.

```
predict(rf_model, 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
```

The predictions made by the model were 100% accurate on the test set. This further implies the validity of our model.