Practical Machine Learning: Prediction Assignment

B.Kumar.Sharma

Dec 19 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. The goal of this project is to form a machine learning model by using 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.

Data

The training data for this project is obtained from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) whereas the test data is avaiable here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data comes from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)

Preliminary Work

The pseudo-random number generator seed was set at 1234. The same seed should be used in order to obtain the same results below.

```
library(caret)
```

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

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

```
set.seed(1234)
```

Loading the data

We first load the datasets into the environment. Before that, it was discovered that the datasets consists of missing values for some predictors, denoted as “Na”.

```
rawtrainSet <- read.csv(file ="pml-training.csv",na.strings = c("NA","#DIV/0!"))
testSet <- read.csv(file="pml-testing.csv",na.strings = c("NA","#DIV/0!"))
```

Data Exploration and Cleaning

We check the number of variables and number of the observations in the datasets.

```
dim(rawtrainSet)
```

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

```
dim(testSet)
```

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

We found that there is some colums with all mising values and they should be removed.

```
rawtrainSet<-rawtrainSet[,colSums(is.na(rawtrainSet)) == 0]
testSet <-testSet[,colSums(is.na(testSet)) == 0]
```

We also determine the number of observations that has missing values

```
sum(!complete.cases(rawtrainSet))
```

```
## [1] 0
```

```
sum(!complete.cases(testSet))
```

```
## [1] 0
```

It seems that every observations has complete values for all variables.

Next, we remove some variables (user and timestamp) that is not related to our model building (predicting whether barbel lift is correct or not based on accelerometer data)

```
cols <- c("user_name", "raw_timestamp_part_1",
                    "raw_timestamp_part_2", "cvtd_timestamp","num_window","new_window")
rawtrainSet <- rawtrainSet[,-which(names(rawtrainSet)%in% cols)]
testSet <- testSet[,-which(names(testSet)%in% cols)]
```

Validation Set

Before we start building our model, we remove a part of the training set as the validation set to test the out-of-sample error.

```
split <- 0.80
trainind <-  createDataPartition(rawtrainSet$classe, p=split, list=FALSE)

trainSet <- rawtrainSet[trainind,]
validSet <- rawtrainSet[-trainind,]
```

Model building

We would use three methods, random forest, multinormial logistic regression and linear discriminant analysis to build models and select the best one out from them.

Cross Validation

To eliminate bias and overfitting when selecting the best models, we would do a data split of 80% for training and the rest for testing. The model with the lowest average error from is the best model.

```
## 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
## # weights:  275 (216 variable)
## initial  value 20217.759056
## iter  10 value 8654.437261
## iter  20 value 6847.686348
## iter  30 value 6564.995601
## iter  40 value 6307.515304
## iter  50 value 6072.477793
## iter  60 value 5911.942737
## iter  70 value 5787.995729
## iter  80 value 5738.566156
## iter  90 value 5714.376593
## iter 100 value 5686.775671
## iter 110 value 5659.081938
## iter 120 value 5651.620073
## iter 130 value 5637.854145
## iter 140 value 5617.825702
## iter 150 value 5588.754050
## iter 160 value 5516.822236
## iter 170 value 5233.227788
## iter 180 value 4582.676002
## iter 190 value 3531.365329
## iter 200 value 2862.718681
## iter 210 value 1302.340784
## iter 220 value 378.907245
## iter 230 value 174.691493
## iter 240 value 97.609541
## iter 250 value 58.295474
## iter 260 value 40.227371
## iter 270 value 23.870406
## iter 280 value 13.840457
## iter 290 value 6.741934
## iter 300 value 1.581966
## iter 310 value 0.574940
## iter 320 value 0.238731
## iter 330 value 0.000798
## final  value 0.000048
## converged
```

```
accuracy1 #for random forest
```

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

```
accuracy2 #for multinormial logistic regression
```

```
## [1] 0.9958559
```

```
accuracy3 #for linear discriminant analysis
```

```
## [1] 0.9620657
```

Final Model

From above, we found that Random Forest model achieve the perfect score. We hence pick Random Forest to be our model. We now use the entire training Set to build the final model and use the validation set to get the expected out-of-sample error.

```
finalmod<- randomForest(classe ~. ,data=trainSet,method="class")
pred <- predict(finalmod,validSet,method="class")

accuracy <- sum(pred==validSet$classe)/length(pred)
1-accuracy
```

```
## [1] 0
```

Our expected out-of-sample error from the model seems to be perfect (0).

Results for the Test Cases

We now use our model to predict the classe for the 20 test cases.

```
finalpred <- predict(mod3,testSet[,-54])
```