Predicting manner of doing exercise

A.SI.M.

11 May, 2016

Executive Summary

With devices like Jawbone Up, Nike FuelBand and Fitbit, it is now possible to collect a large amount of data about personal activity relatively inexpensively. People regularly do measure how much of a particular activity they do but they rarely measure how well they do it. In this project, I will 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).

In this assignment, I will predict the manner in which exercise was done. This is the “classe” variable in the training set. Prediction is done via various variables and report below describes model building, cross validation and expected out of sample error. 20 different test cases are used in the prediction model.

Loading libraries

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

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

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

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.2.5
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.1
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.2.5
```

```
## 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
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.2.5
```

Loading Dataset

Dataset to develop model and validate model is downloaded from provided link.

The training data for this project are available here: training dataset (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here: testing dataset (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Downloading the data:

```
train_file <- "pml-training.csv"
test_file <- "pml-testing.csv"
train_file_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
if (!file.exists(train_file)){
    download.file(train_file_url, train_file)
}

test_file_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
if (!file.exists(test_file)){
    download.file(test_file_url, test_file)
}
```

Loading the data into R:

```
train_data <- read.csv(train_file, na.strings = c("#DIV/0!","NA"))
test_data <- read.csv(test_file, na.strings = c("#DIV/0!","NA"))
```

Cleaning Data

I will be removing first five columns namely X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp as they don’t have any significance in building the prediction model.

```
train_data <- subset(train_data, select = -(1:5))

# removing variables with near zero variance
var_nearZeroVar <- nearZeroVar(train_data)
train_data <- train_data[, -var_nearZeroVar]

# removing missing data as denoted by NA
missing_data <- sapply(train_data, function(x) mean(is.na(x))) > 0.9
train_data <- train_data[, missing_data == F]
```

Model Building

I have decided to use `RandomForest` model to see if it returns acceptable performance. I will be using `train` function in `caret` package to train the model and have used ten fold cross validation.

```
# data partitioning
data_partIndex <- createDataPartition(train_data$classe, p = 0.7, list = FALSE)
train_set <- train_data[data_partIndex,]
test_set <- train_data[-data_partIndex,]

model_ctrl <- trainControl(method = "cv", number = 10, verboseIter = FALSE)
# using RandomForest model
rf_model <- train(classe ~ ., method = "rf", data = train_set, trControl = model_ctrl)
```

Not, I’m using `Boosting` algorithm with the ten fold cross validation for the prediction.

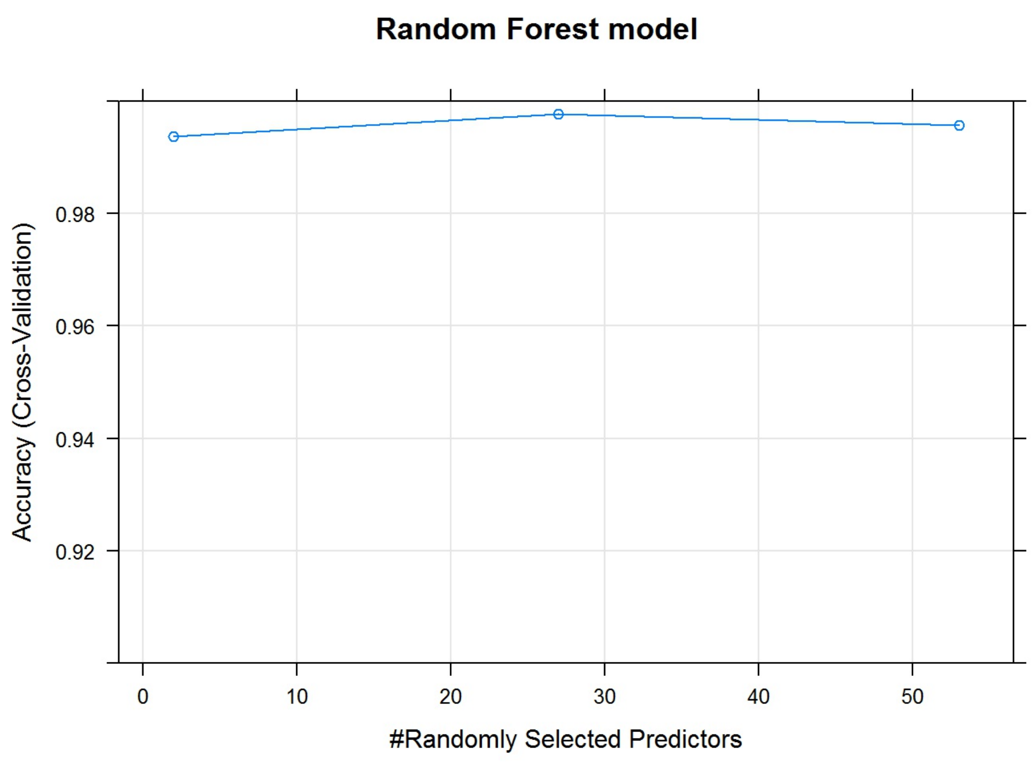
```
boost_model <- train(classe ~ ., method = "gbm", data = train_set, verbose = FALSE, trControl = model_ctrl)
```

```
## Loading required package: plyr
```

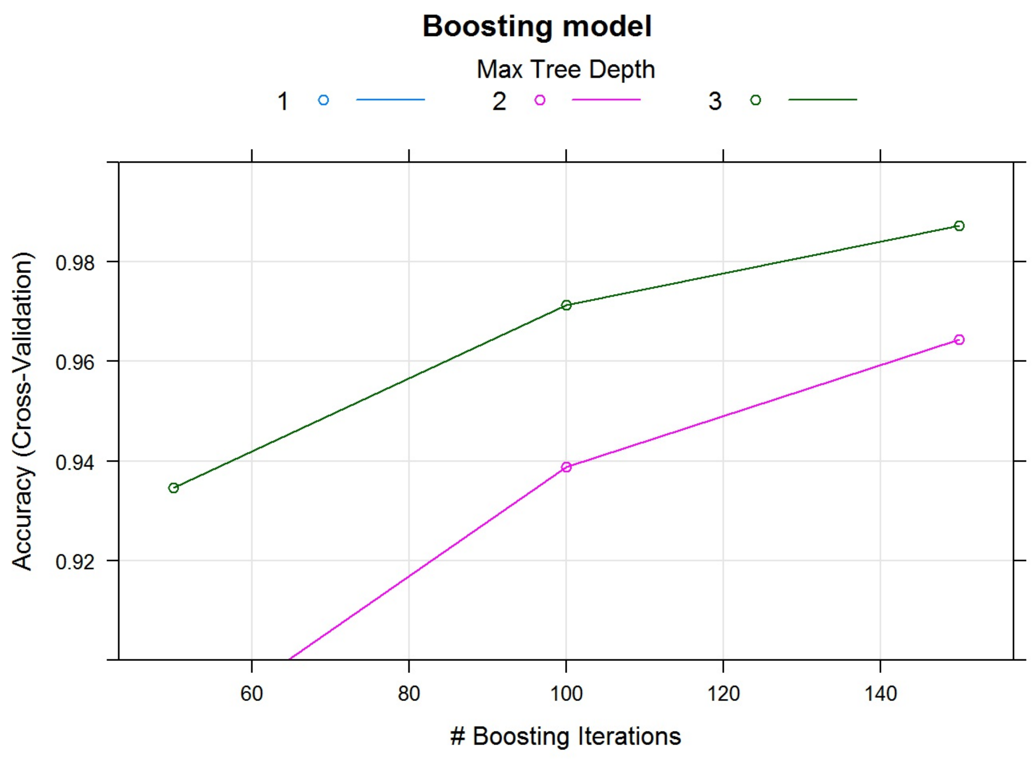
Random Forest vs Boosting Model Evaluation

Predicting the classe in test dataset using the fitted model from training dataset. Confusion matrix will compare predicted vs actual values.

```
plot(rf_model, ylim = c(0.9, 1), main = "Random Forest model")
```



```
plot(boost_model, ylim = c(0.9, 1), main = "Boosting model")
```



```
# predicting classe in test set using RandomForest fitted from training set
rf_model_predicted <- predict(rf_model, newdata = test_set)

# confusion matrix for out-of-sample error estimation from prediction for RF fitted
confusionMatrix(test_set$classe, rf_model_predicted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1137    2    0    0
##          C    0    3 1023    0    0
##          D    0    0    3  960    1
##          E    0    0    0   10 1072
##
## Overall Statistics
##
##                Accuracy : 0.9968
##                  95% CI : (0.995, 0.9981)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9959
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9951   0.9897   0.9991
## Specificity            1.0000   0.9996   0.9994   0.9992   0.9979
## Pos Pred Value         1.0000   0.9982   0.9971   0.9959   0.9908
## Neg Pred Value         1.0000   0.9994   0.9990   0.9980   0.9998
## Prevalence             0.2845   0.1937   0.1747   0.1648   0.1823
## Detection Rate         0.2845   0.1932   0.1738   0.1631   0.1822
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      1.0000   0.9985   0.9973   0.9944   0.9985
```

```
# predicting classe in test set using Boosting fitted from training set
boost_model_predicted <- predict(boost_model, newdata = test_set)

# confusion matrix for out-of-sample error estimation from prediction for Boosting fitted
confusionMatrix(test_set$classe, boost_model_predicted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1669    4    0    1    0
##          B    6 1122   10    1    0
##          C    0   15 1009    1    1
##          D    0    5   13  946    0
##          E    0    4    1   19 1058
##
## Overall Statistics
##
##                Accuracy : 0.9862
##                  95% CI : (0.9829, 0.9891)
##     No Information Rate : 0.2846
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9826
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9757   0.9768   0.9773   0.9991
## Specificity            0.9988   0.9964   0.9965   0.9963   0.9950
## Pos Pred Value         0.9970   0.9851   0.9834   0.9813   0.9778
## Neg Pred Value         0.9986   0.9941   0.9951   0.9955   0.9998
## Prevalence             0.2846   0.1954   0.1755   0.1645   0.1799
## Detection Rate         0.2836   0.1907   0.1715   0.1607   0.1798
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9976   0.9860   0.9866   0.9868   0.9970
```

It is clear from the comparsion that Random Forest model is the best model to fit the dataset.

Out of Sample (OOS) error

```r
# calculating out of sample error for Random Forest model
miss_class = function(values, predicted) {
        sum(predicted != values) / length(values)
}
OOS_error_rate_rf = miss_class(test_set$classe, rf_model_predicted)
OOS_error_rate_rf
```

```
## [1] 0.003228547
```

Estimated out of sample error rate for the random forests model is 0.0032285 as reported by the final model.

Final Prediction

Finally, predicting the classe of testing dataset using the model selected and writing the result to files.

```r
# prediction on test set
test_prediction <- predict(rf_model, newdata = test_data)
test_prediction <- as.character(test_prediction)

# output directory to hold prediction results
dir_output <- "predicted_output"
if (!file.exists(dir_output)){
  dir.create(dir_output)
}

# creating function to write prediction result to files
fn_write_files <- function(x) {
    n <- length(x)
    for (i in 1:n) {
        filename <- paste0(dir_output, "/problem_id_", i, ".txt")
        write.table(x[i], file = filename, quote = FALSE, row.names = FALSE, col.names = FALSE)
    }
}

# creating prediction files
fn_write_files(test_prediction)
```