Course 8 Practical Machine Learning Project

Suresh B K

2 January 2019

Table of Contents

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 (see the section on the Weight Lifting Exercise Dataset).

Objective of the project

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.

Data for the project

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://groupware.les.inf.puc-rio.br/har.

Execution

Loading the possible essential libraries

```
  ############## Load the required libraries ##############
  suppressMessages(library(dplyr))

## Warning: package 'dplyr' was built under R version 3.4.4

  suppressMessages(library(lubridate))
  suppressMessages(library(ggplot2))

## Warning: package 'ggplot2' was built under R version 3.4.4

  suppressMessages(library(Hmisc))

## Warning: package 'Hmisc' was built under R version 3.4.4

## Warning: package 'Formula' was built under R version 3.4.4

  suppressMessages(library(caret))

## Warning: package 'caret' was built under R version 3.4.4

  suppressMessages(library(randomForest))

## Warning: package 'randomForest' was built under R version 3.4.4

  suppressMessages(library(LiblineaR))

## Warning: package 'LiblineaR' was built under R version 3.4.4
```

```
  suppressMessages(library(logicFS))

## Warning: package 'LogicReg' was built under R version 3.4.4

## Warning: package 'mcbiopi' was built under R version 3.4.4

  suppressMessages(library(gbm))

## Warning: package 'gbm' was built under R version 3.4.4

  suppressMessages(library(grid))
  suppressMessages(library(gridExtra))
  suppressMessages(library(ggpubr))
  suppressMessages(library(rattle))

## Warning: package 'rattle' was built under R version 3.4.4

  suppressMessages(library(rpart))
  suppressMessages(library(e1071))

## Warning: package 'e1071' was built under R version 3.4.4

  suppressMessages(library(caTools))

## Warning: package 'caTools' was built under R version 3.4.4

  library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.4.4
```

Downloading and cleaning data

Download and explore the data

The data is downloaded as per the below code

```
  fileurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-trainin
g.csv"
  filename <- "./course8week4_project/pml_train.csv"
  filedir <- "./course8week4_project"
  if (!file.exists(filedir)){
        dir.create(filedir)
  }
  if (!('pml_train.csv'%in% list.files(path=filedir))){
    download.file(fileurl,file.path(filename))
  }

  if(!('pml_train' %in% ls())){
    pml_train <- read.csv(filename,na.strings = c("NA","#Div/0!",""," "))
  }

  fileurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing
```

```
.csv"
  filename <- "./course8week4_project/pml_test.csv"
  if (!('pml_test.csv'%in% list.files(path=filedir))){
    download.file(fileurl,file.path(filename))
  }

  if(!('pml_test' %in% ls())){
    pml_test <- read.csv(filename,na.strings = c("NA","#Div/0!",""," "))
  }
  dim(pml_train)

## [1] 19622   160

  dim(pml_test)

## [1]  20 160
```

After the downloading the data, the training data was explored in excel by opening the csv file. Upon viewing the file the below was found 1. There are many columns that do not have data. this needs to be cleaned 2. The first 7 columns are demographics and can be excluded from model building and prediction 3. Training dataset has 19622. This needs to be split into 2 datasets one for building the model and another for cross validation and determing the accuracy and out-of-sample cross validation error

The below code does the first 2 actions in the above 3 points

Clean Data and create sub training and sub test samples

```
  #exclude first 7 columns
  pml_train <- pml_train[,c(8:ncol(pml_train))]
  pml_test <- pml_test[,c(8:ncol(pml_test))]

  #exclude near zero variance columns
  nzcols <- nearZeroVar(pml_train)
  pml_train <- pml_train[,-nzcols]

  nzcols <- nearZeroVar(pml_test)
  pml_test <- pml_test[,-nzcols]

  # remove columns that have only NA values in them
  pml_train<- pml_train[,colSums(is.na(pml_train))==0]
  pml_test<- pml_test[,colSums(is.na(pml_test))==0]

  table(pml_train$classe)

## 
##    A    B    C    D    E 
## 5580 3797 3422 3216 3607

  # create sub samples of training data set for model building and cross vali
dation
```

```
  intrain <- createDataPartition(y=pml_train$classe,p=0.7,list=FALSE)
  pml_subtrain <-  pml_train[intrain,]
  pml_subtest <- pml_train[-intrain,]

  table(pml_subtrain$classe)

## 
##    A    B    C    D    E 
## 3906 2658 2396 2252 2525

  table(pml_subtest$classe)

## 
##    A    B    C    D    E 
## 1674 1139 1026  964 1082
```

From the above it is clear that the data is not skewed towards a particular value of classe though the number of observations for A is the highest

Build and Select Model

Build Model

The variable to be predicted is "classe". From the data it is clear that this is a categorical variable and hence, a classification type of model needs to be built.

We can look at multiple models some that are known for speed and some known for accuracy. The below model types will be used and the best among them will be chosen

Speed based model types: - Decision tree (rpart) and Boosted Logistic Regression (LogitBoost)

Model Types known for accuracy - RandomForest (rf) - Gradient Boosting Tree (gbm) - Kernel Support Vector Machine (svm)

Below code builds the model and displays the accuracy of the prediction of each model vis-a-vis the sub training data sample (sample on which the model was built) and sub testing data sample

```
  set.seed(3523)
  # High Speed relatively low accuracy models
  mod_rpart <- rpart(classe~.,data=pml_subtrain,method="class")
  pml_subtrain_data <- pml_subtrain[,-ncol(pml_subtrain)]
  pml_subtrain_label <- pml_subtrain[,ncol(pml_subtrain)]
  mod_boost_logreg <- LogitBoost(pml_subtrain_data,pml_subtrain_label)
  #mod_boost_logreg <- train(classe~.,data=pml_subtrain,method="LogitBoost")

  # Low speed high accuracy models
  mod_rf <- train(classe~.,data = pml_subtrain,method="rf")
  mod_gbm <- train(classe~.,data = pml_subtrain,method="gbm",verbose=FALSE)
  mod_kernelsvm <- train(classe~.,data = pml_subtrain,method="svmLinear3")
```

```r
  # predict the training data set values using each model
  pred_rpart_train <- predict(mod_rpart,pml_subtrain,type="class")
  pred_logreg_train <- predict(mod_boost_logreg,pml_subtrain)
  pred_rf_train <- predict(mod_rf,pml_subtrain)
  pred_gbm_train <- predict(mod_gbm,pml_subtrain)
  pred_kernelsvm_train <- predict(mod_kernelsvm,pml_subtrain)

  # predict the validation data set values using each model
  pred_rpart_test <- predict(mod_rpart,pml_subtest,type="class")
  pred_logreg_test <- predict(mod_boost_logreg,pml_subtest)
  pred_rf_test <- predict(mod_rf,pml_subtest)
  pred_gbm_test <- predict(mod_gbm,pml_subtest)
  pred_kernelsvm_test <- predict(mod_kernelsvm,pml_subtest)

  # Compute the accuracy of the models on training data set
  accuracy_rpart_train <- confusionMatrix(pml_subtrain$classe,pred_rpart_train)$overall[1]
  accuracy_logreg_train <- confusionMatrix(pml_subtrain$classe,pred_logreg_train)$overall[1]
  accuracy_rf_train <- confusionMatrix(pml_subtrain$classe,pred_rf_train)$overall[1]
  accuracy_gbm_train <- confusionMatrix(pml_subtrain$classe,pred_gbm_train)$overall[1]
  accuracy_svm_train <- confusionMatrix(pml_subtrain$classe,pred_kernelsvm_train)$overall[1]

  # Compute the accuracy of the models on validation data set
  accuracy_rpart_test <- confusionMatrix(pml_subtest$classe,pred_rpart_test)$overall[1]
  accuracy_logreg_test <- confusionMatrix(pml_subtest$classe,pred_logreg_test)$overall[1]
  accuracy_rf_test <- confusionMatrix(pml_subtest$classe,pred_rf_test)$overall[1]
  accuracy_gbm_test <- confusionMatrix(pml_subtest$classe,pred_gbm_test)$overall[1]
  accuracy_svm_test <- confusionMatrix(pml_subtest$classe,pred_kernelsvm_test)$overall[1]
  #Create and display accuracy table
  accuracylist <- c(accuracy_rpart_train,accuracy_logreg_train,accuracy_rf_train,accuracy_gbm_train,accuracy_svm_train,accuracy_rpart_test,accuracy_logreg_test,accuracy_rf_test,accuracy_gbm_test,accuracy_svm_test)
  accuracydf <- data.frame(matrix(data=accuracylist,nrow=2,ncol=5,byrow=T))
  colnames(accuracydf) <- c("rpart","boosted.logreg","rf","gbm","svm")
  rownames(accuracydf) <- c("subtrain","subtest")

  accuracydf
```

```
##              rpart boosted.logreg        rf      gbm       svm
## subtrain 0.7390260      0.9320589 1.0000000 0.973939 0.7155128
## subtest  0.7325404      0.9136077 0.9913339 0.957859 0.7131691
```

Select Model

Based on the accuracy values it can be concluded that random forest is the best model. The accuracies are high for both sub training and sub test data sets indicating that there is no overfitting of the model

Based on this random forest model is selected. Details of random forest accuracy is given below

```
  confusionMatrix(pml_subtest$classe,pred_rf_test)

## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    9 1126    4    0    0
##          C    0    7 1015    4    0
##          D    0    1   17  944    2
##          E    0    1    0    6 1075
## 
## Overall Statistics
## 
##                Accuracy : 0.9913
##                  95% CI : (0.9886, 0.9935)
##     No Information Rate : 0.286
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.989
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9947   0.9921   0.9797   0.9895   0.9981
## Specificity            1.0000   0.9973   0.9977   0.9959   0.9985
## Pos Pred Value         1.0000   0.9886   0.9893   0.9793   0.9935
## Neg Pred Value         0.9979   0.9981   0.9957   0.9980   0.9996
## Prevalence             0.2860   0.1929   0.1760   0.1621   0.1830
## Detection Rate         0.2845   0.1913   0.1725   0.1604   0.1827
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9973   0.9947   0.9887   0.9927   0.9983
```

The accuracy of random forest is 99.13% and the expected out-of-sample error is 0.87% or 0.0087 (1-accuracy) and the 95% confidence interval is 0.9886 - 0.9935

Test Set Prediction

Let’s now predict the test set values using the selected model

```
  pred_test <- predict(mod_rf,pml_test)
  pred_test

##  [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
```