Machine Learning Coursera project

Savita Kohli

April 7, 2018

Practical Machine Learning Project : Prediction Assignment Writeup

Overview:

Human Activity Recognition - HAR - has emerged as a key research area in the last years and is gaining increasing attention by the pervasive computing research community, especially for the development of context-aware systems. There are many potential applications for HAR, like: elderly monitoring, life log systems for monitoring energy expenditure and for supporting weight-loss programs, and digital assistants for weight lifting exercises.

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.

Data being analyzed here is downloaded from https://d396qusza40orc.cloudfront.net/predmachlearn/. Original source of data is http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har.

The goal of this project is to predict the manner in which they did the exercise. Outcome is 'classe' variable in the training set.

Loading libraries

```
library(caret)
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
library(randomForest)
library(knitr)
```

Following will be sequence of activities:

1. Load both training and testing data into R. Further divide the training data into 70:30 ratio for training and test validation. Original testing data will be used to predict 20 cases for the quiz

2. Preprocessing the training data. Remove columns which have more than 95% 'NAs' or missing values

3. Build Multiple models on the training data

4. Use the models to predict on Testing data (30% of population)

5. Run accuracy tests on the predicted values

6. Select the highest accuracy model and predict values for 20 cases in testing data.

1. Load data into R and create partitions of the data

```
training <-
read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv", na.strings=c("NA","#DIV/0!",""))
testing <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv", na.strings=c("NA","#DIV/0!",""))

inTrain <- createDataPartition(training$classe, p=0.7, list=FALSE)
myTraining <- training[inTrain, ]
myTesting <- training[-inTrain, ]
dim(myTraining); dim(myTesting)

## [1] 13737   160

## [1] 5885  160
```

2. Preprocessing data by removing columns with more than 95% NAs

```
nzv <- nearZeroVar(myTraining, saveMetrics = T)
myTraining <- myTraining[, nzv$nzv == FALSE]
NAs <- sapply(myTraining, function(x) sum(is.na(x))/nrow(myTraining)) > .95
myTraining_new <- myTraining[, NAs == FALSE]
## first  column are identification only
myTraining_new <- myTraining_new[, -(1)]
myTesting_New <- myTesting[, names(myTesting) %in% names(myTraining_new)]
dim(myTraining_new) ; dim(myTesting_New)

## [1] 13737    58

## [1] 5885   58
```

3. Build Multiple models on the training data and

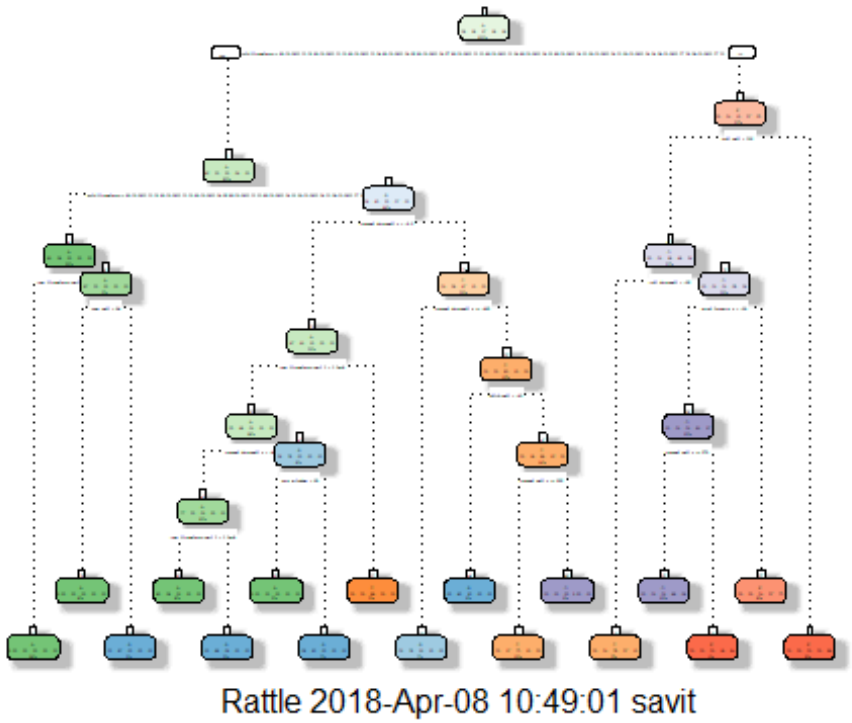
4. Use the models to predict on Testing data (30% of population)

Model rpart - fit classification tree as a model

```
trControl <- trainControl(method="cv", number=3)

model_rpart <- rpart(classe ~ ., data=myTraining_new, method="class")
trainpred_rpart <- predict(model_rpart, myTesting_New, type = "class")
confMat_rpart <- confusionMatrix(trainpred_rpart, myTesting_New$classe)
```

rf (random forest algorithm)

```
model_RF <- randomForest::randomForest(classe ~ ., data = myTraining_new)
trainpred_RF <- predict(model_RF,newdata=myTesting_New)
confMatRF <- confusionMatrix(trainpred_RF, myTesting_New$classe)
fancyRpartPlot(model_rpart)
```



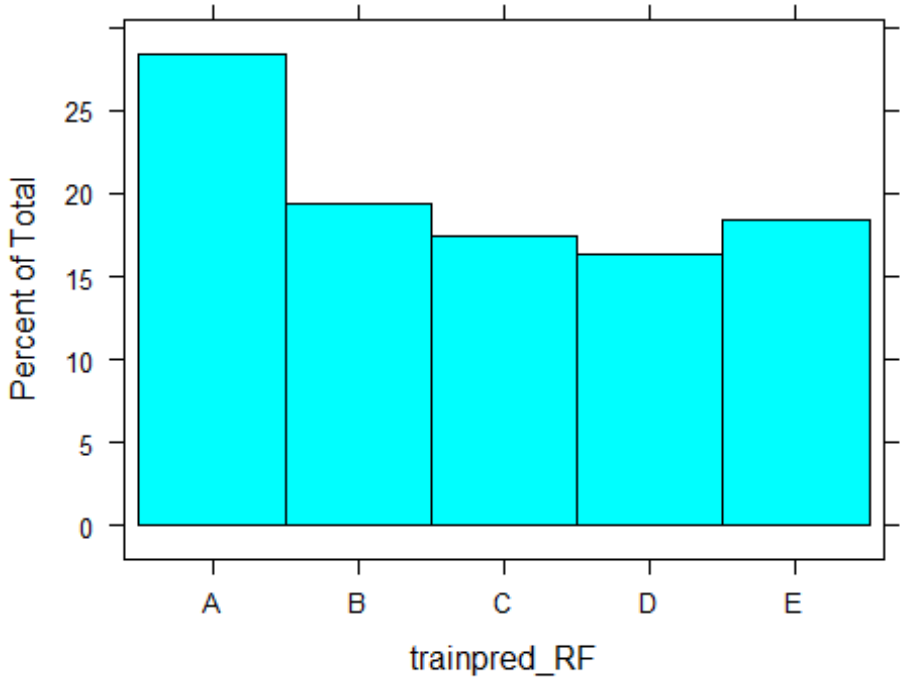

```
confMatRF$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1138    2    0    0
##          C    0    1 1023    3    0
##          D    0    0    1  959    1
##          E    0    0    0    2 1081
```

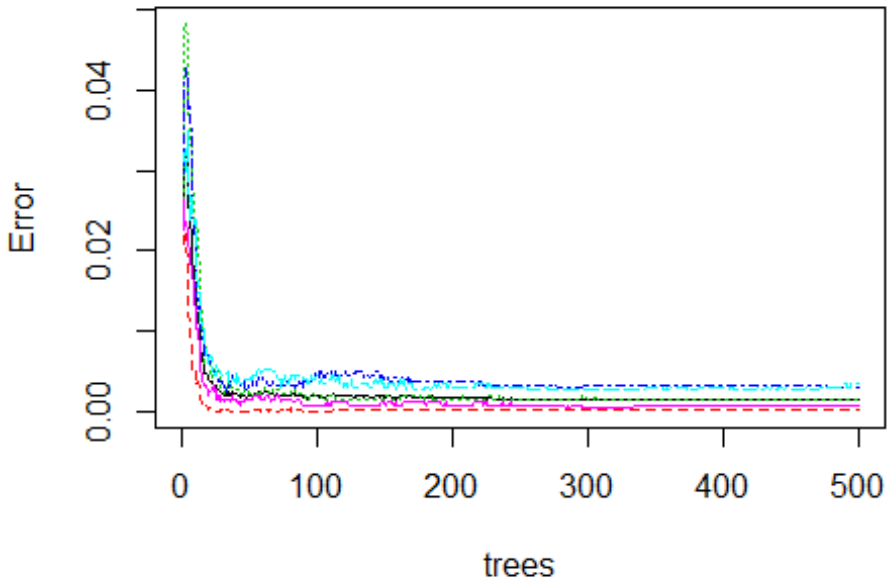
```
model_RF

##
## Call:
##  randomForest(formula = classe ~ ., data = myTraining_new)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.16%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3905    1    0    0    0 0.0002560164
## B    4 2654    0    0    0 0.0015048909
## C    0    5 2389    2    0 0.0029215359
## D    0    0    5 2244    3 0.0035523979
## E    0    0    0    2 2523 0.0007920792
```

Out of Sample Error Rate = .09% as shown in the results of Rain Forest Model

```
histogram(trainpred_RF); plot(model_RF,main="Accuracy of Random forest model
by number of predictors")
```



ccuracy of Random forest model by number of predi



lda (linear discriminant analysis) model

```
model_LDA <- train(classe~., data=myTraining_new, method="lda",
trControl=trControl, verbose=FALSE)
trainpred_LDA <- predict(model_LDA,newdata=myTesting_New)
```

```
confMatLDA <- confusionMatrix(myTesting_New$classe,trainpred_LDA)
confMatLDA$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1525  121   27    0    1
##          B  135  839  163    2    0
##          C    3   99  896   25    3
##          D    0    1  127  788   48
##          E    0    0    6   89  987
```

```
##confMatLDA$overall
```

gbm (boosting model)

```
model_GBM <- train(classe~., data=myTraining_new, method="gbm",
trControl=trControl, verbose=FALSE)
trainpred_GBM <- predict(model_GBM,newdata=myTesting_New)
confMatGBM <- confusionMatrix(myTesting_New$classe,trainpred_GBM)
confMatGBM$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1136    2    1    0
##          C    0    0 1020    6    0
##          D    0    0    7  952    5
##          E    0    0    0    0 1082
```

```
##confMatGBM$overall
```

Display and compare accuracy of 4 models

```
cbind(rpart = confMat_rpart$overall[1],RandomForest = confMatRF$overall[1],
LDA = confMatLDA$overall[1], GBM = confMatGBM$overall[1] )
```

```
##              rpart RandomForest      LDA       GBM
## Accuracy 0.8635514    0.9983008 0.855565 0.9964316
```

Apply gbm model to predict 20 cases in testing data

```
model_GBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 79 predictors of which 45 had non-zero influence.
```

```
testing$classe <- predict(model_GBM, testing)
testing$classe
```

```
##  [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
```