Coursera Practical Machine Learning

Dianne Dino

August 7, 2019

Overview

In this project, we are tasked to **predict the manner in which our correspondents did the exercise**. The data is gathered from link (http://groupware.les.inf.puc-rio.br/har).

The **training data** for this project are available here: link (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) The **test data** are available here: link (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv) ### Set-up Data

```
#Load Libraries
library(data.table); library(caret);library(ggplot2); library(dplyr)
```

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

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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(rpart);library(rpart.plot);library(RColorBrewer);library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(randomForest);library(party);library(rattle)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:rattle':
##
##     importance
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
#Set-up Data
train <- fread("pml-training.csv")
test <- fread("pml-testing.csv")
train$V1 = NULL; test$V1 = NULL
df = train
testing = test

dim(df);dim(testing)
```

```
## [1] 19622   159
```

```
## [1]  20 159
```

Data Exploration

Below is the count of classifiers in the dataset. (5 Levels)

```
table(df$classe)
```

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

Pre-Processing

```
#Drop Columns with 50% and more Missing Data
pMiss <- function(x){return(sum(is.na(x))/length(x)*100)}
tmp = data.frame(apply(df,2,pMiss))
tmp = cbind(row.names(tmp),tmp)
colnames(tmp) = c('col_name','pcnt_missing_val')
row.names(tmp) = 1:nrow(tmp)
tmp = tmp[tmp$pcnt_missing_val>=50,1]
tmp = tmp %>% as.character()
#Train Set
df = select(df,-c(tmp)) #Dropped 100 columns
#Test Set
testing = select(testing, -(tmp)) #Dropped 100 columns
#Dropping columns you dont need
training  <- df[, -c(1:6)]
testing <- testing[,-c(1:6)]
dim(training); dim(testing)
```

```
## [1] 19622    53
```

```
## [1] 20 53
```

Modeling

Cross-Validation

Cross-Validation is done to ensure the accuracy and fitness of the model.

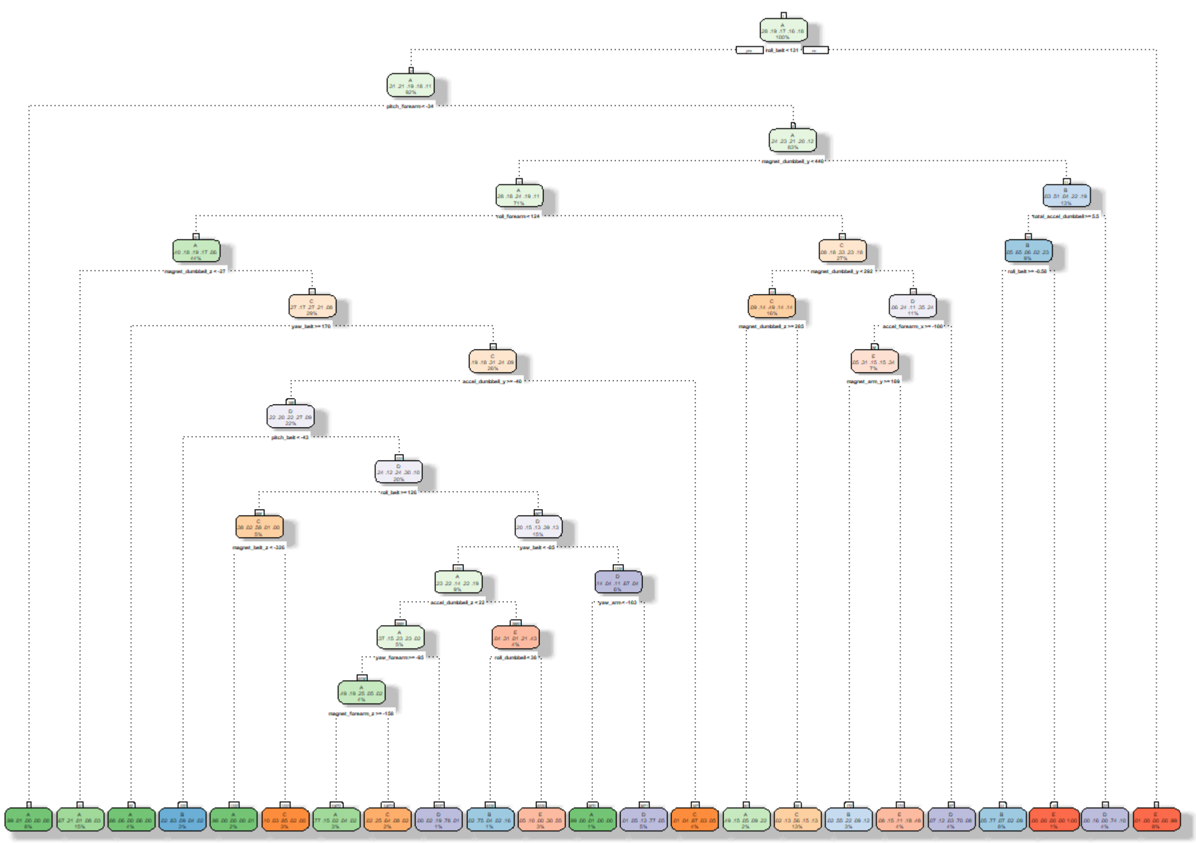
```
inTrain <- createDataPartition(y=training$classe,
                               p=0.75, list = FALSE)
sub_train <- training[inTrain,]; sub_test <- training[-inTrain,]
dim(sub_train); dim(sub_test)
```

```
## [1] 14718    53
```

```
## [1] 4904   53
```

Modeling using Decision Trees

```
#Fit to Model
modFitA1 <- rpart(classe ~ ., data=sub_train, method="class")
#Plot Tree
fancyRpartPlot(modFitA1)
```



Rattle 2019-Aug-07 15:59:00 10012218

```r
#Predict
predictionsA1 <- predict(modFitA1, sub_test, type = "class")
#Metrics
confusionMatrix(predictionsA1, as.factor(sub_test$classe))$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1290  199   20   75   57
##          B   27  518   89   36   60
##          C   39  118  677  125   91
##          D   17   73   49  500   54
##          E   22   41   20   68  639
```

```r
confusionMatrix(predictionsA1, as.factor(sub_test$classe))$overall[1]
```

```
##  Accuracy
## 0.7389886
```

Modeling using RF

```r
trControl <- trainControl(method="cv", number=5) #Set Folds
#Fit to Model
modFitB1 <- train(classe~., data=sub_train, method="rf", trControl=trControl, verbose=TRUE)
#Predict
predictionB1 <- predict(modFitB1, sub_test)
#Metrics
confusionMatrix(predictionB1, as.factor(sub_test$classe))$table
```

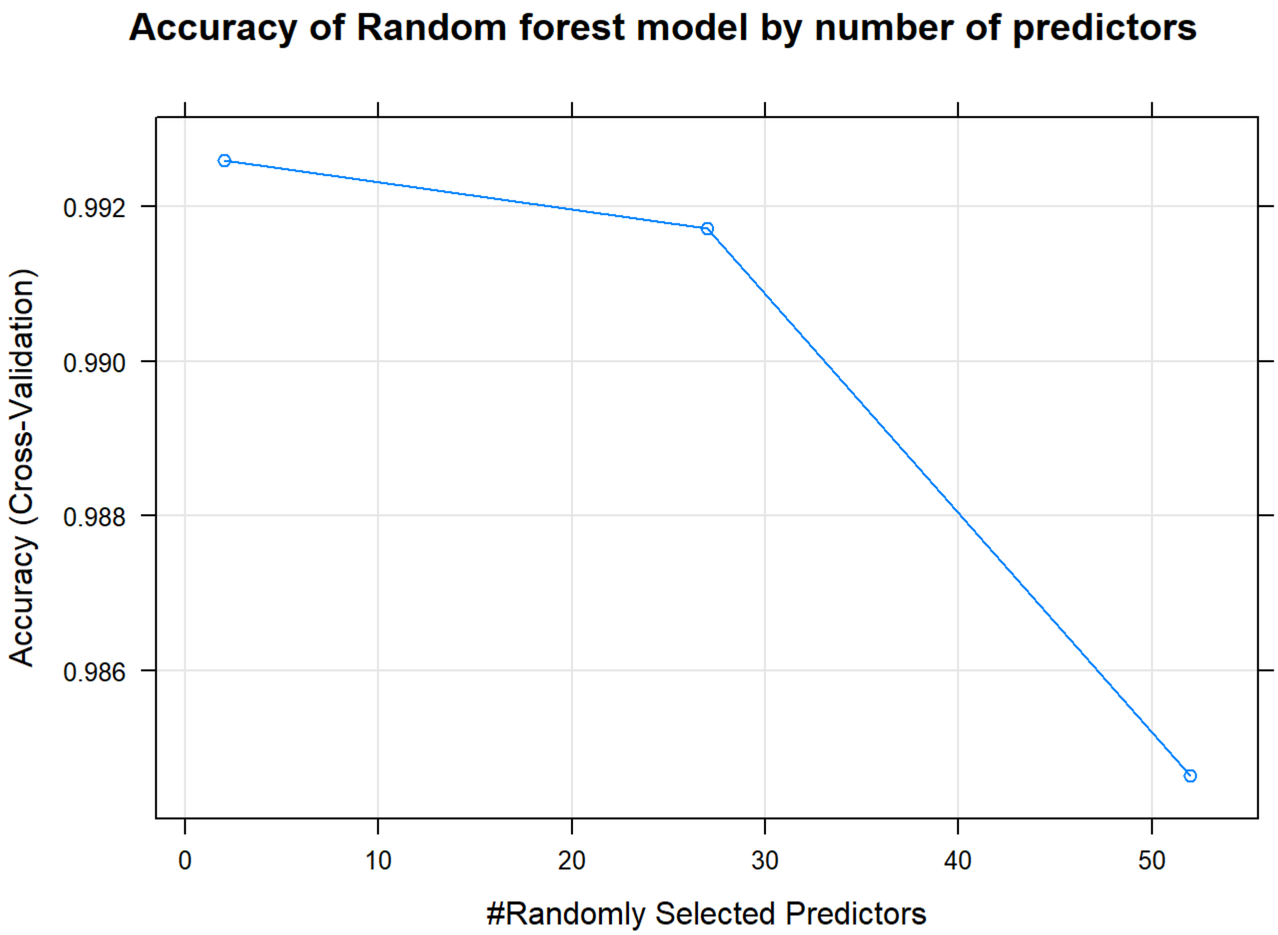
```
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    5    0    0    0
##          B    1  943    8    0    0
##          C    0    1  847   13    1
##          D    0    0    0  790    0
##          E    0    0    0    1  900
```

```r
confusionMatrix(predictionB1, as.factor(sub_test$classe))$overall[1]
```

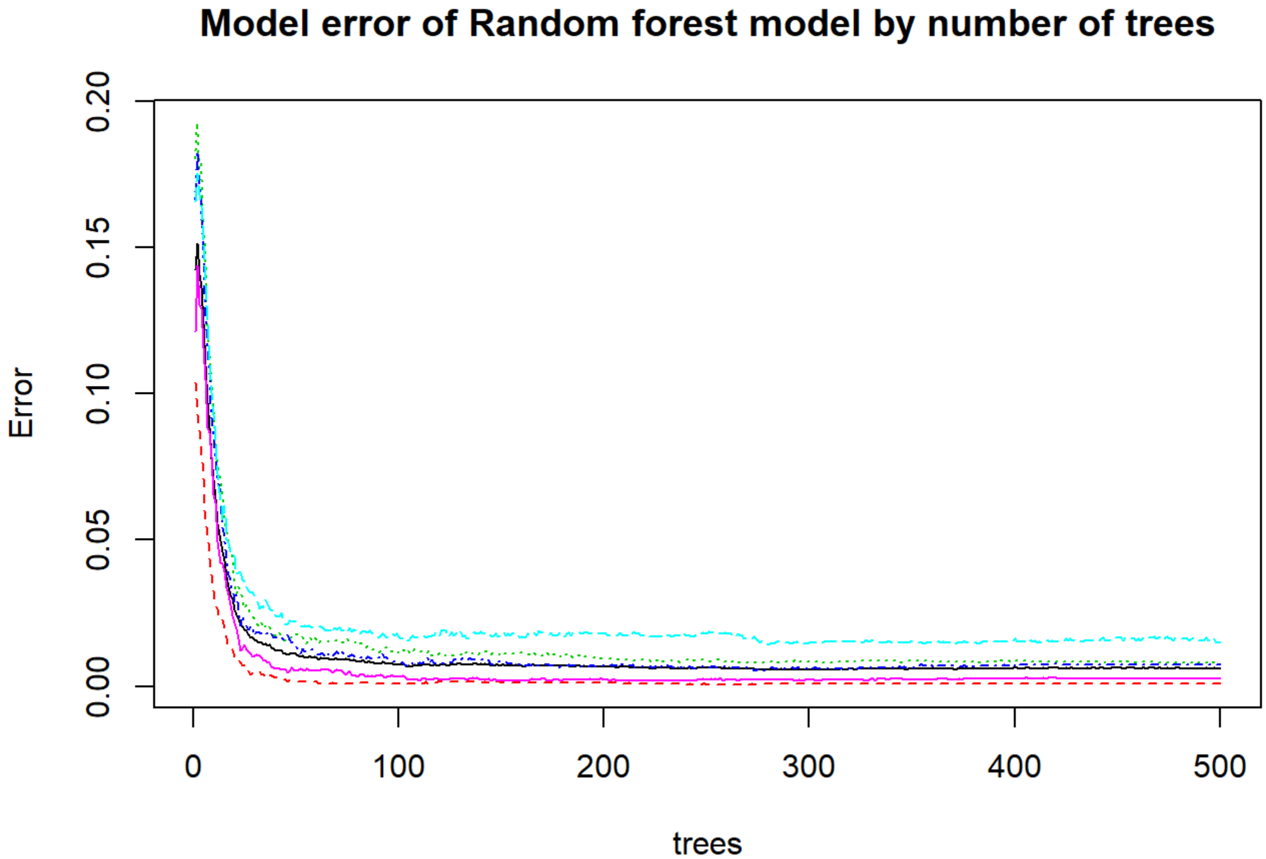
```
##  Accuracy
## 0.9938825
```

Plot RF Metrics

```r
#Plot RF Metrics
plot(modFitB1,main="Accuracy of Random forest model by number of predictors")
```




```
plot(modFitB1$finalModel,main="Model error of Random forest model by number of trees")
```



####

Varible Importance After running the variable importance function, we’ve figured out that roll_belt is the most important variable in this model with the rest with only 75.5 value and below.

```
MostImpVars <- varImp(modFitB1)
MostImpVars
```

```
## rf variable importance
## 
##   only 20 most important variables shown (out of 52)
## 
##                   Overall
## roll_belt          100.00
## yaw_belt            83.47
## magnet_dumbbell_z   71.06
## magnet_dumbbell_y   67.73
## pitch_belt          65.68
## pitch_forearm       57.12
## magnet_dumbbell_x   52.59
## roll_forearm        49.89
## magnet_belt_y       45.38
## accel_dumbbell_y    45.37
## magnet_belt_z       44.87
## roll_dumbbell       44.25
## accel_belt_z        41.11
## accel_dumbbell_z    40.09
## roll_arm            33.79
## accel_forearm_x     33.34
## accel_arm_x         32.00
## gyros_belt_z        31.12
## magnet_arm_y        30.25
## accel_dumbbell_x    29.23
```

What Model to Use?

Random Model will be used in this project because of the stable and high metrics it posses. #### Prediction Results using Test Set

```
FinalTest <- predict(modFitB1,newdata=testing)
FinalTest
```

```
##  [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
```