Peer-Graded-PML

Samriddhi

10/20/2020

Introduction

People love numbers. They want a to measure everything and store that data to compare it for future purposes.They like to do so with their health (or bodies) too. Using devices such as Jawbone Up, Nike FuelBand, and Fitbit are now a trend which is followed by everyone around the world. These devices allow us to measure and collect large data-sets containing information about personal activities and health for example, heart rate, pulse, steps walked,etc , to find a pattern and prevent any future health problems.

```
library(data.table)

## Warning: package 'data.table' was built under R version 3.6.3

library(rpart)

## Warning: package 'rpart' was built under R version 3.6.3

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.3

## corrplot 0.84 loaded

library(gbm)

## Warning: package 'gbm' was built under R version 3.6.3

## Loaded gbm 2.1.8

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

library(knitr)

## Warning: package 'knitr' was built under R version 3.6.3

library (caret)

## Warning: package 'caret' was built under R version 3.6.3

## Loading required package: lattice

library (rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.6.3
```

A raw data has no meaning. It needs to be cleaned or else it's useless. So, the data will now be cleaned, followed by exploring of the data.

```
Url_testing <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv"
Url_trading  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv"

data_testing <- read.csv(url(Url_testing))
data_trading <- read.csv(url(Url_trading))
```

This is followed by Cleaning of the data.

```
data_training <- data_trading[, colSums(is.na(data_trading)) == 0]
data_testing_2 <- data_testing[, colSums(is.na(data_testing)) == 0]
```

After the cleaning and exploring of data, it can now be used to make predictions. Some percentage of data is used for training while the rest is used in testing process. 70% of the database will be used in training while the rest 30 percent of the database will be used in testing procedure. The thirty prcent data used in testing will later be used for making future predictions.

```
data_training <- data_training[, -c(1:7)]
data_testing_2 <- data_testing_2[, -c(1:7)]
dim(data_training)

## [1] 19622    86
```

Testing and training data is split into groups . WE are cross-validating the samples.

```
set.seed(1234)
datatraining2 <- createDataPartition(data_trading$classe, p = 0.7, list =
FALSE)
data_training <- data_training[datatraining2, ]
data_testing_2 <- data_training[-datatraining2, ]
dim(data_training)

## [1] 13737    86

dim(data_testing_2)

## [1] 4123   86
```

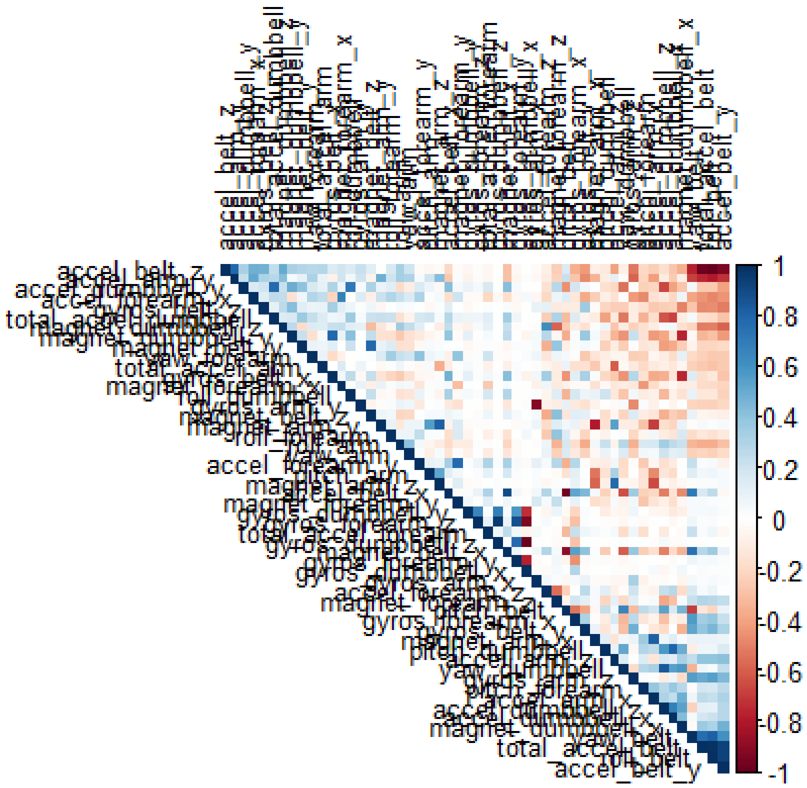
We are now going to diagnose the prediction for a unique/single/different value; the non zero values. Also while setting the dimensions.

```
non_Zero <- nearZeroVar(data_training)
data_training <- data_training[, -non_Zero]
data_testing_2 <- data_testing_2[, -non_Zero]
dim(data_training)

## [1] 13737    53
```

```
dim(data_testing_2)

## [1] 4123   53
```

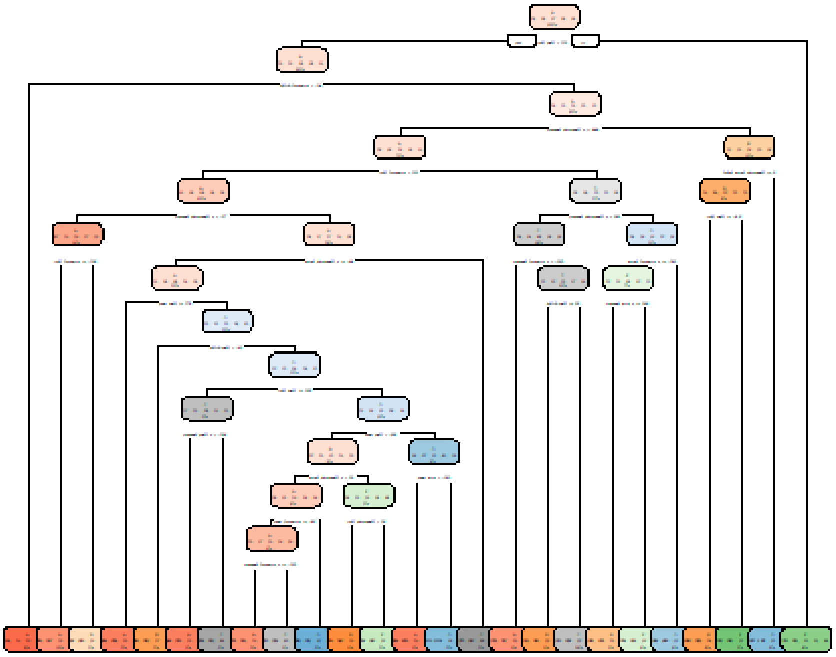
Now we will the the coorelated variances. And then plot the correlation on a chart.

```
plot_cor_sd <- cor(data_training[, -53])
corrplot(plot_cor_sd, order = "FPC", method = "color", type = "upper", tl.cex
= 0.8, tl.col = rgb(0, 0, 0))
```



The dark-coloured intersection marked the correlation and prediction in the chart. Ahead of this, we can see the model building using 2 different algorithms ; namely trees and random forests.

```
set.seed(20000)
tre_dec_sd <- rpart(classe ~ ., data=data_training, method = "class")
rpart.plot(tre_dec_sd)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Validating the database.

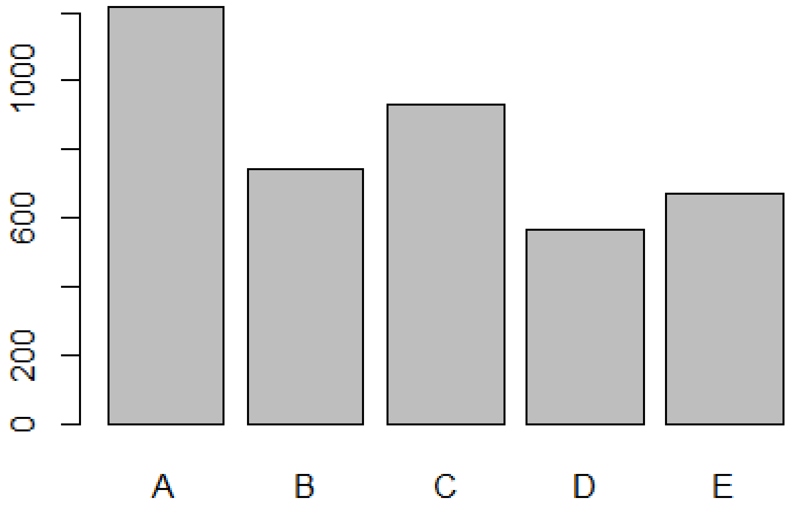
```
model_pre_sd <- predict(tre_dec_sd, data_testing_2, type = "class")
ab <- confusionMatrix(model_pre_sd, data_testing_2$classe)
ab
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1067  105    9   24    9
##          B   40  502   59   63   77
##          C   28   90  611  116   86
##          D   11   49   41  423   41
##          E   19   41   18   46  548
## 
## Overall Statistics
## 
##                Accuracy : 0.7642
##                  95% CI : (0.751, 0.7771)
##     No Information Rate : 0.2826
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.7015
## 
##  Mcnemar's Test P-Value : < 2.2e-16
## 
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9159   0.6379   0.8279   0.6295   0.7201
## Specificity            0.9503   0.9284   0.9055   0.9589   0.9631
## Pos Pred Value         0.8789   0.6775   0.6563   0.7487   0.8155
## Neg Pred Value         0.9663   0.9157   0.9602   0.9300   0.9383
## Prevalence             0.2826   0.1909   0.1790   0.1630   0.1846
## Detection Rate         0.2588   0.1218   0.1482   0.1026   0.1329
## Detection Prevalence   0.2944   0.1797   0.2258   0.1370   0.1630
## Balanced Accuracy      0.9331   0.7831   0.8667   0.7942   0.8416
```

Plotting the Model Chart.

```
plot(model_pre_sd)
```



All the models are being applied one after the other: General Booster Model followed by the GBM Model.

```
set.seed(10000)
ctr_gbm_sd <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
valid_gbm_sd <- train(classe ~ .,data=data_training, method = "gbm",
trControl = ctr_gbm_sd, verbose = FALSE)
valid_gbm_sd$finalModel

## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 52 had non-zero influence.
```

Due to some technical issues, I couldn’t attach the output file. So instead of that I’ve attached the Rmd file and the pdf file. Hope you’ll consider.

It was a very informative project. I learned a lot along this journey. And all the credit goes to the wonderful team who put up such a great course. Thanking the team behind this course and the Uni.