Practical Machine learning

Astha Agarwal

10/23/2020

Details regarding the assignment:

So basically for the analysis of this project, I have collected some databases from NIKE,FITBIT,etc. These datasets I will be using for my final project of Practical Machine Learning.

So basically we are checking whether the exercises are being done with utmost precision or not and also we check whether they are int the correct order.'

Step 1 of this project:Dataset Loading Step 2:Data is being processed Step 3:the next step would be exploration of the dataset. Step 4:Now we need to predict the model we will be selecting to get the output. Step 5:Now, we will be predicting the data on the given dataset for testing.

```
library(caret)

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

## Loading required package: lattice

## Loading required package: ggplot2

library(knitr)

library(data.table)

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

library(rpart.plot)

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

## Loading required package: rpart

library(rpart)

library(gbm)

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

## Loaded gbm 2.1.8
```

```
library(ggplot2)

library(corrplot)

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

## corrplot 0.84 loaded
```

Exploring and cleaning the data.

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

traU  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv"

d_tst <- read.csv(url(tU))

d_tr <- read.csv(url(traU))
```

In this step we would be cleaning the data.

```
train_dat <- d_tr[, colSums(is.na(d_tr)) == 0]

test_d <- d_tst[, colSums(is.na(d_tst)) == 0]
```

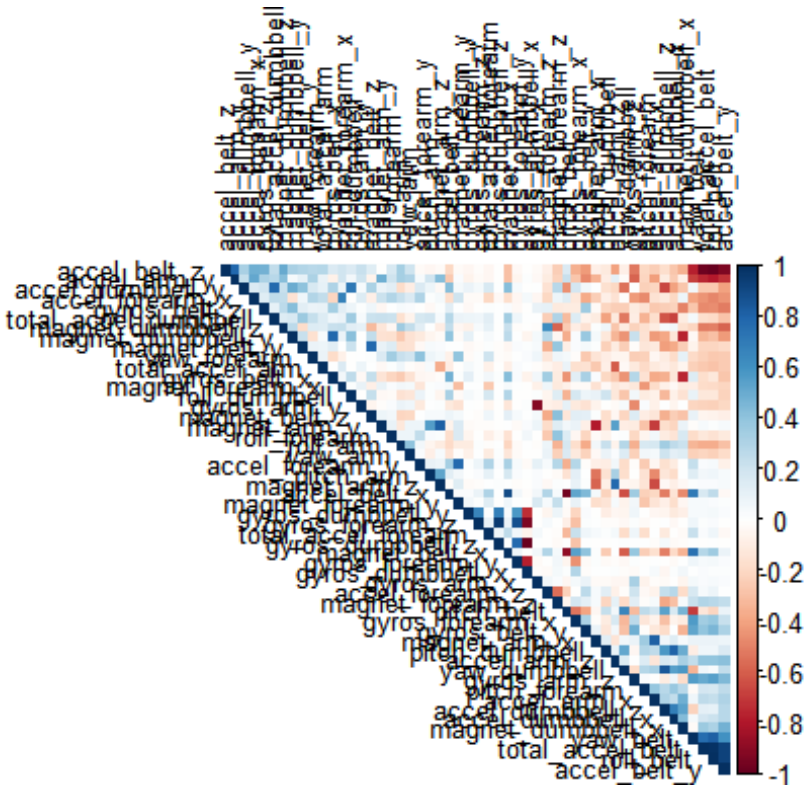
Next step is prediction of the data. So 30% is used for testing our dataset and 70% for training the set we are using in this project.

```
train_dat <- train_dat[, -c(1:7)]

test_d <- test_d[, -c(1:7)]

dim(train_dat)

## [1] 19622    86
```

in this step we are deleting the variables that are non-zero referred to as 'nz' in this code

```
set.seed(1234)

dtrain <- createDataPartition(d_tr$classe, p = 0.7, list = FALSE)


train_dat <- train_dat[dtrain, ]

test_d <- train_dat[-dtrain, ]

dim(train_dat)

## [1] 13737    86
```

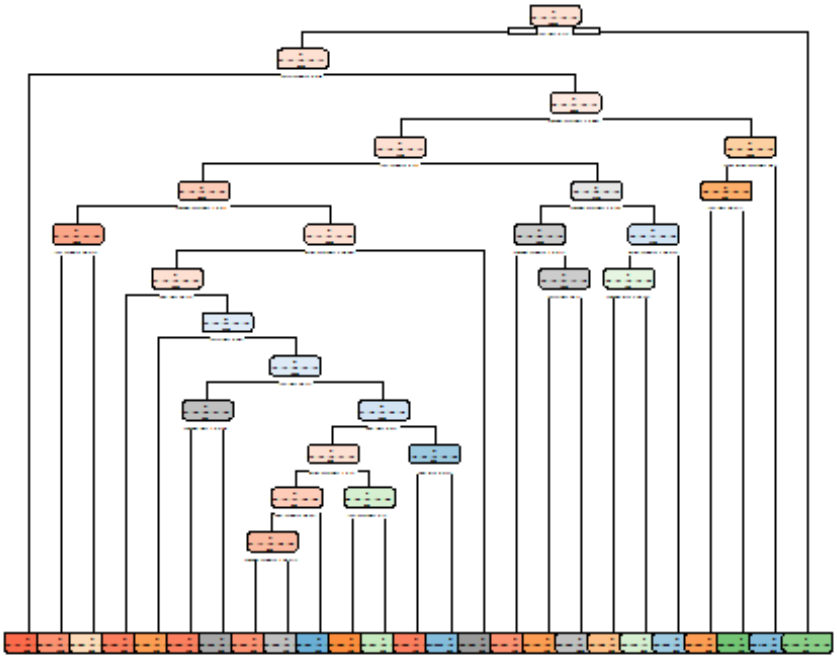
```
dim(test_d)

## [1] 4123   86

nZ <- nearZeroVar(train_dat)

train_dat <- train_dat[, -nZ]

test_d <- test_d[, -nZ]

dim(train_dat)

## [1] 13737    53

dim(test_d)

## [1] 4123   53

p_cor <- cor(train_dat[, -53])

corrplot(p_cor, order = "FPC", method = "color", type = "upper", tl.cex =
0.8, tl.col = rgb(0, 0, 0))
```



the corr. predic. are with the dark colour intersec.This is the observation in this case.

Next step is the building of our model for the dataset we are using. The Algorithms we will be using are trees and random forests for the prediction part.

```r
set.seed(20000)

tr <- rpart(classe ~ ., data=train_dat, method = "class")

rpart.plot(tr)

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

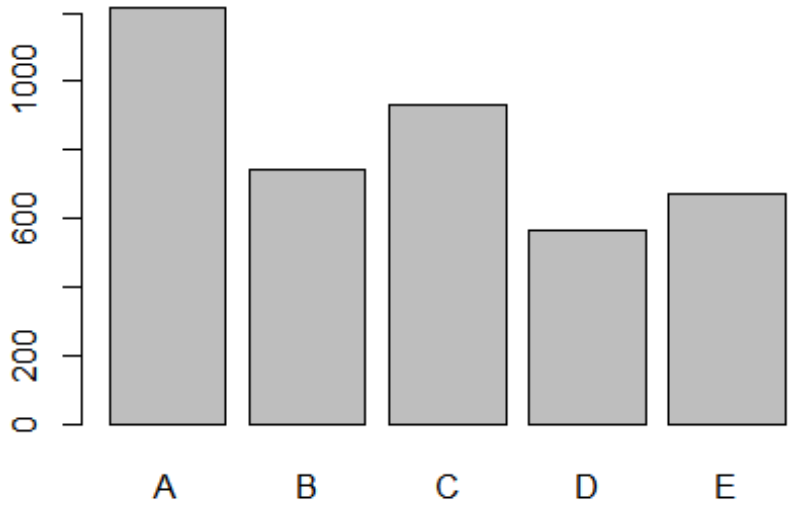


Validation of the model

```r
modp <- predict(tr, test_d, type = "class")

ab <- confusionMatrix(modp, test_d$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
```

```
plot(modp)
```



Lets apply two models in this case: First is General boosted model. Second is gbm model.

```r
set.seed(10000)

cand_gbm <- trainControl(method = "repeatedcv", number = 5, repeats = 1)

val <- train(classe ~ .,data=train_dat, method = "gbm", trControl = cand_gbm,
verbose = FALSE)
val$finalModel

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

Conclusion: Prediction that someone did the exercise order wise. Ananlysing using some techniques like cross validation. This is the end of my project.Hope you find it useful and i would like to thank the Profs. for helping me out!! I am attaching the outputs as well for further use if required.