# Practical ML Analytical work

Harhita More

20th October 2020

Here I am doing a visualisation of practical ML.I will be using RStudio Markdown for the purpose of my code and Knitr for the analysing stuff.

A detailed introduction of my assignment:

We are working on the humongous database from different renowned categories like Nike band, Fitbit, jawbone. Now we are going to leverage the given data for analytical purpose.Various people do not perform same level of exercise or other fitness routine. Therefore, in this detailed project work, we are making use of the values from the measure of accelerometer of such people. This data is used for anticipation purpose. The prediction is basically based on if the person is following his fitness routine religiously and timely. There are two files. They consist of the following: 1) Test data 2) Training data These data will be used to anticipate the ordering of exercise as well. Our first step is to load the data and the second step is to process the data. After this, the third step is to perform the exploratory analysis and the fouth one is to predict the selection of the most efficient model.The last step is to predict the output we are going to obtain.

In the following lines of code, we are loading the various packages which will be helpful further

```
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
```

Here we have starting making use of the data by assigning the URL to the variables “fityes” and “workyes”. Then we read the files in two different variables “fityesvalue” and “valueworkyes”.

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

fityesvalue <- read.csv(url(fityes))
valueworkyes <- read.csv(url(workyes))
```

Let us now clean the data for our smooth analysis later on. For this, we rae using two variables “vwork” and“vfit” and filtering out the non applicable data in the given values.

```
vwork <- valueworkyes[, colSums(is.na(valueworkyes)) == 0]
vfit<- fityesvalue[, colSums(is.na(fityesvalue)) == 0]
```

Let us assume “vwork” as our training dataset and “vfit” as our testing dataset. vwork has seventy percent data and vfit has thirty percent data. vfit will be used again later for predicting twenty more cases.

```
vwork <- vwork[, -c(1:7)]
vfit <- vfit[, -c(1:7)]
dim(vwork)

## [1] 19622    86
```

Now we are using one more variable “worknow” to create partition of the training dataset. Then we check the dimensions of the modified dataset in the variables declared above.

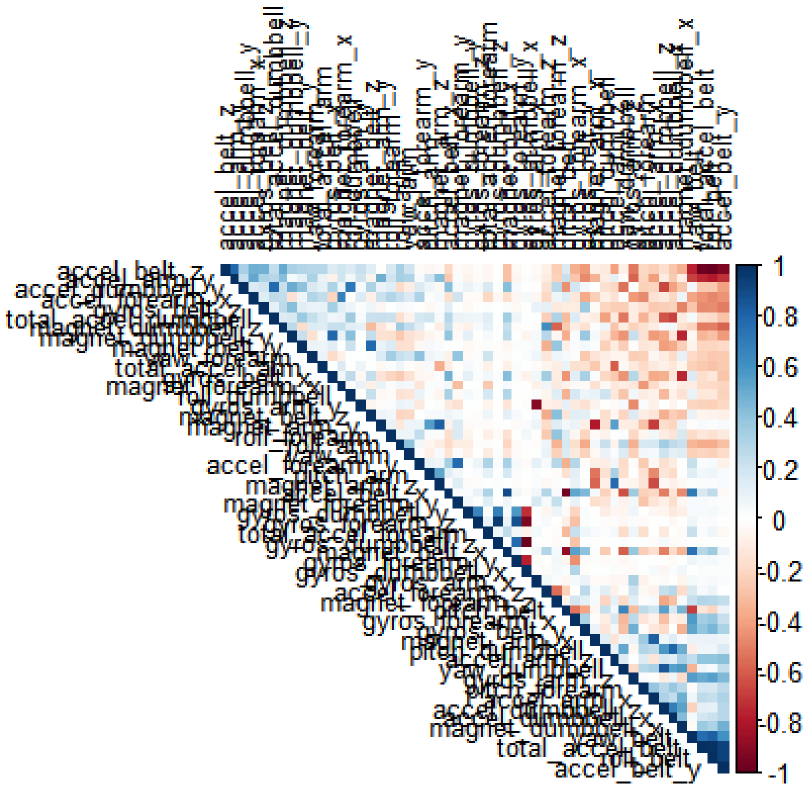
```
set.seed(1234)
worknow <- createDataPartition(valueworkyes$classe, p = 0.7, list = FALSE)
vwork <- vwork[worknow, ]
vfit <- vwork[-worknow, ]
dim(vwork)

## [1] 13737    86

dim(vfit)

## [1] 4123   86
```

In the following chunk, we remove unwanted values from the dataset and check dimensions again to verify.

```
notzerohere <- nearZeroVar(vwork)
vwork <- vwork[, -notzerohere]
vfit <- vfit[, -notzerohere]
dim(vwork)

## [1] 13737    53

dim(vfit)

## [1] 4123   53
```

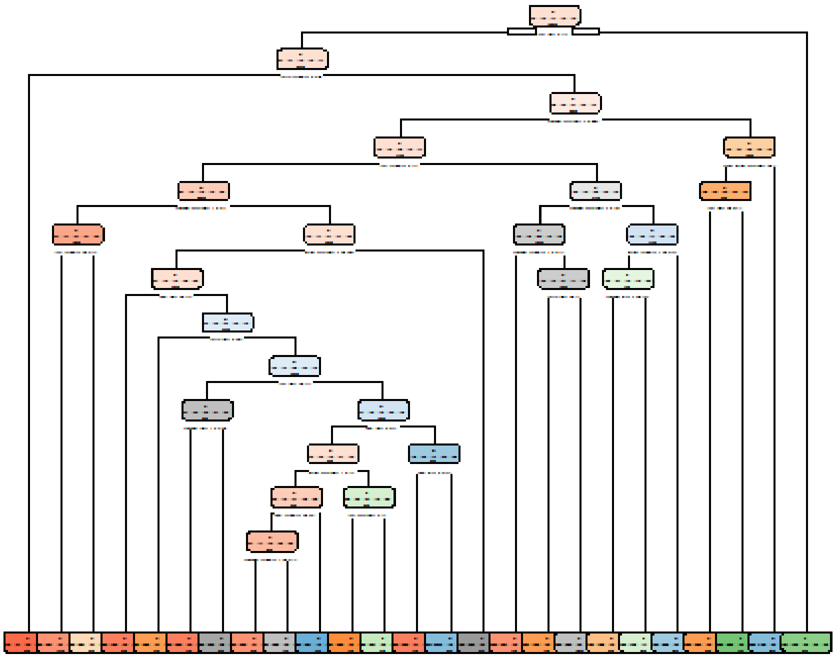
Now we plot the outcome of the cleaned data for the pupose of exploratory analysis

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



The observation is that the correlation prediction are the dark coloured intersects. In the next chunk of code, we are going to predict. Hencefoth, we use trees algorithm and random forests algorithm to build models.

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

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



Since the models are build, now it is time to validate each one of them.

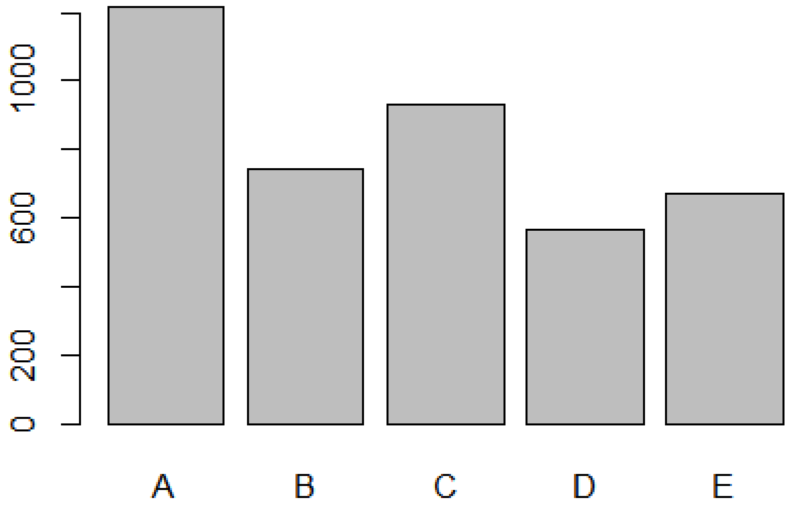
```
makingmodel <- predict(makingalgo, vfit, type = "class")
wow<- confusionMatrix(makingmodel, vfit$classe)
wow
```

```
## 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
```

Now we will plot our validated model.

```
plot(makingmodel)
```



We are not applying the models simultaneously here. The models one after another are applied as follows: 1) General boosted 2)GBM

```
set.seed(10000)
newone <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
newnewone <- train(classe ~ .,data=vwork, method = "gbm", trControl = newone,
verbose = FALSE)
newnewone$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: Hence, we have successfully completed the analysis to arrive at significant results.

Note: I am unable to attach the file ue to some glitch. The file comprises of the output. Thus, attached herewith are the rmd and pdf files. I attached the link to GitHub as well which originally contained the HTML and rmd line. Sorry for the inconvenience caused.

Thank you so much for going through my analysis.