# PML

Varnika

10/20/2020

Introduction- This is final course project of the practical machine learning (PML). We will be using rstudio markdown and knitr. proceeding for the analysis.

Since we have collected the databases from nike, fitbit, and jawbone we will be utilizing these data for the analysis of the assignment

In ths project we used data from accelerometer measure of the individuals of unique physicality With the exsisting data collected, we will be able to se the individuals who are doing exersises or not. There are 2 files a)test data b)training data from these files we will see the number of idividuals doing exercise or not.

At first we will load the data, then proceed for the processing the data and then we will do the exploratory analysis later we predict that for which model to select and then finally for the predicting of the output of the testing set

```
library(caret)

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

## Loading required package: lattice

## Loading required package: 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(data.table)
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
```

the data is been taken for cleaning and exploring the data

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

data_testing <- read.csv(url(testUrl))
data_training <- read.csv(url(traUrl))
```

cleaning of the input data

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

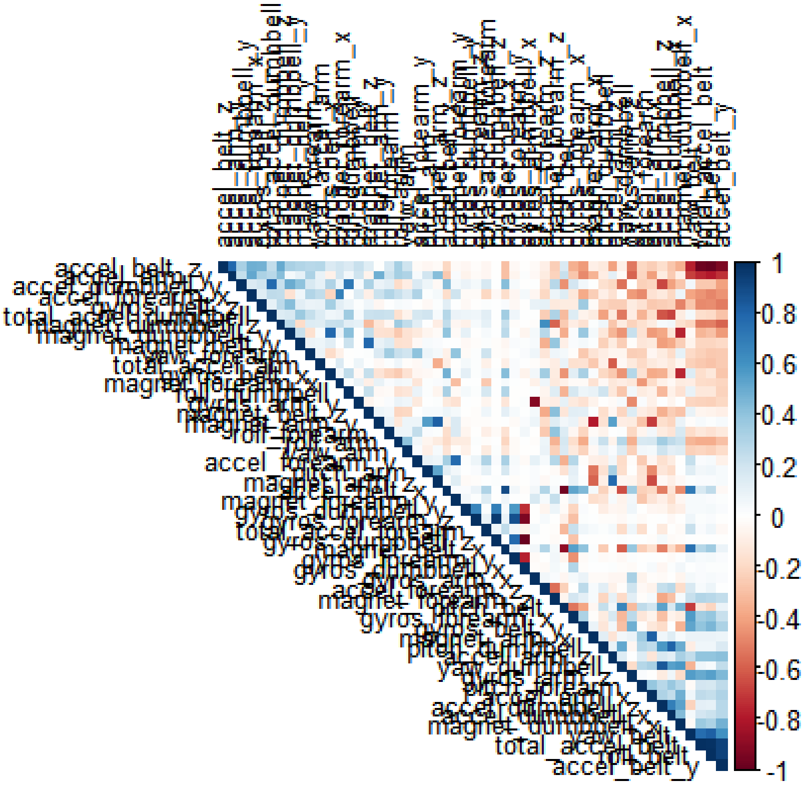
now we will prepare the data for pred. We will consider around 70% of the data for the training set and the 30% of the data for the testing data set. This testing_data is later used in twenty different cases

```
training_data <- training_data[, -c(1:7)]
testing_data <- testing_data[, -c(1:7)]
dim(training_data)

## [1] 19622    86

set.seed(1234)
datatraining <- createDataPartition(data_training$classe, p = 0.7, list = FALSE)
training_data <- training_data[datatraining, ]
testing_data <- training_data[-datatraining, ]
dim(training_data)

## [1] 13737    86

dim(testing_data)

## [1] 4123   86
```

Variables that are nonzero are removed from the data gives

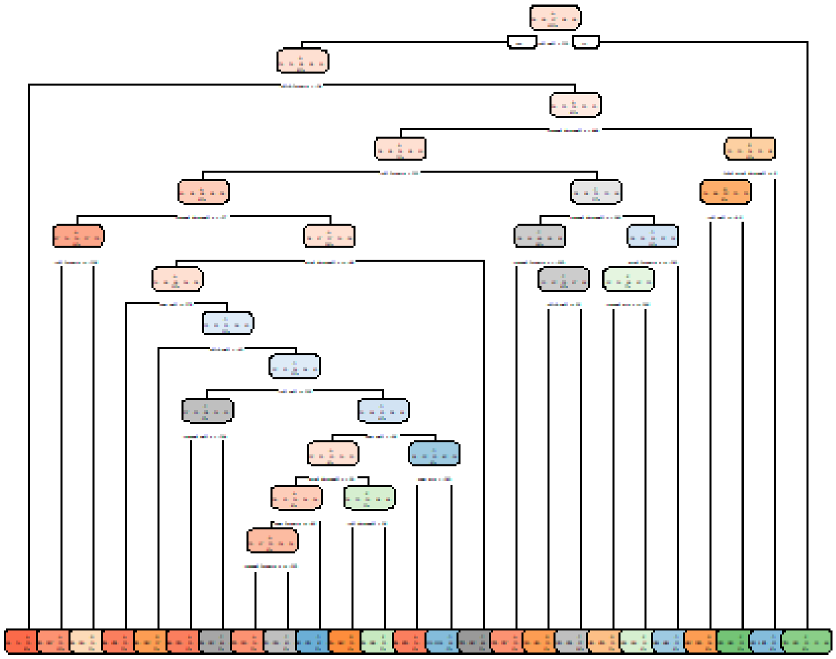
```
none_Zero <- nearZeroVar(training_data)
training_data <- training_data[, -none_Zero]
testing_data <- testing_data[, -none_Zero]
dim(training_data)
```

```
## [1] 13737    53

dim(testing_data)

## [1] 4123   53

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



As you can see corr. predic. are the ones with the dark colour intersec. For proceeding for the model building we will use 2 different types of algorithms , trees and random forests for the prediction part

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

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

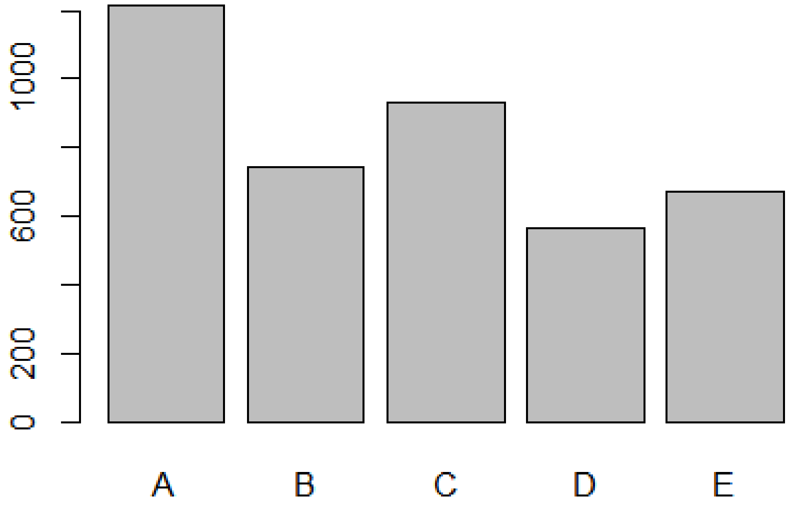


Validating the model

```
modelpre <- predict(tre_dec, testing_data, type = "class")
ab <- confusionMatrix(modelpre, testing_data$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
```

```r
plot(modelpre)
```



By applying two Models one by one a)boosted model b)gbm model

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

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

At last we have predicted the number of individuals who does exercise or not and then later we did a cross validation and this why I chose this specific way towards approaching and then predicted for 20. I have attached the link to GitHub, which contained the HTML and rmd file. Still, due to some unprecedented reason, as I could not attach the file, which consisted of the output, so I have attached the pdf file and the rmd file. Please consider the request.