WEEK4 COURSE PROJECT PML

RACHIT KUMAR

16th October' 2020

This is the week 4 final course project of the practical machine learning also in this I will be using rstudio markdown and knitr. proceeding for the analysis

Introduction of the project

we have collected huge databases from the Nike band, Fitbit, jawbone, and we will be utilizing those data for our analysis in this peer grade assignment.

So in this project, with the data from the accelerometer measure. of the individuals of their different-different class of physical activity

with the help of data, we will be predicting whether the individual is doing the exercises properly or not and the two files comprise of the test and training data, and from this, we will also predict the numbering of exercise like the order of them basically

firstly we will load the data and then proceed for the processing of the data and then we will do the exploratory analysis and then prediction for which model to select and then finally for the predicting of the o/p of the testing set

```
library(caret)
```

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

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

```
library(knitr)
```

```
library(data.table)
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
library(rpart)
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(ggplot2)
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

Now we will take the data and do the cleaning and then exploring the data.

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

data_test <- read.csv(url(testUrl))
data_tra <- read.csv(url(traUrl))
```

now proceeding for the cleaning the input of the data

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

now we will prepare the data for pred. in which we will consider seventy percentage of the data for the training set and rest of the thirty percentage of the data for the testing data set and testing_data will be used furthuer again for the prediction of the 20 of the 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_tra$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
```

now we will be removing the variables that are non zero from the data gives

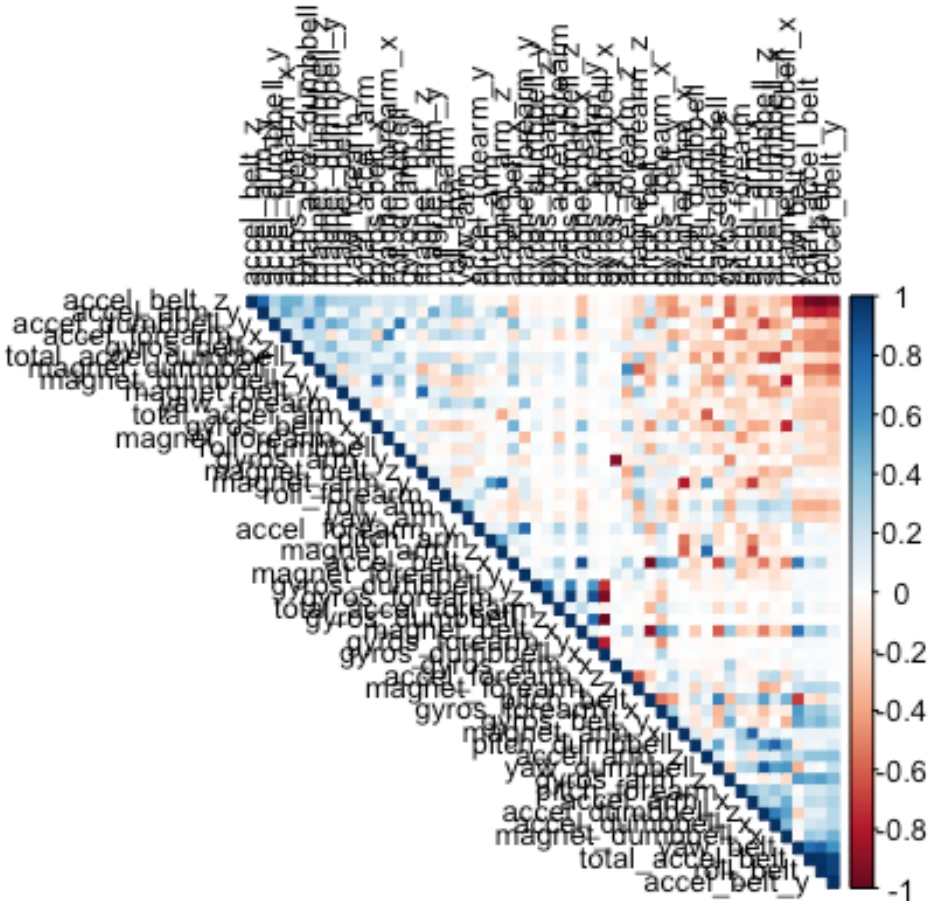
```
noneZero <- nearZeroVar(training_data)
training_data <- training_data[, -noneZero]
testing_data <- testing_data[, -noneZero]
dim(training_data)

## [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))
```

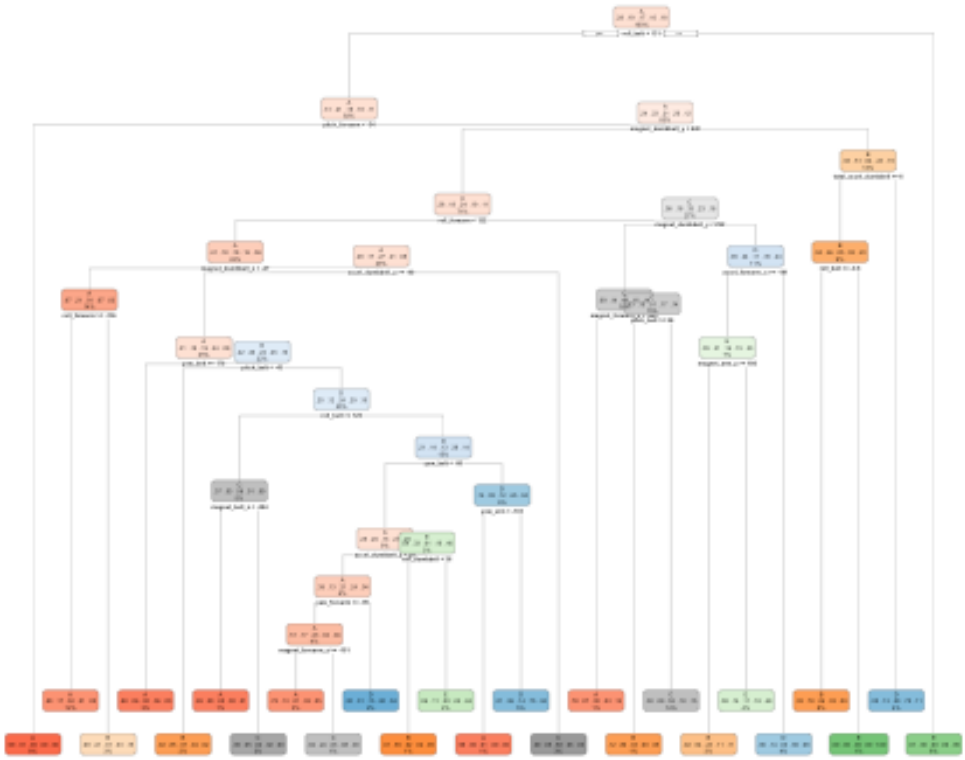


now in this as we can see that the corr. predic. are the ones with the dark colour intersec.

now we will be proceeding for the model building and for this we will use 2 different types of algorithms , trees and random forests for the prediction part

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

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



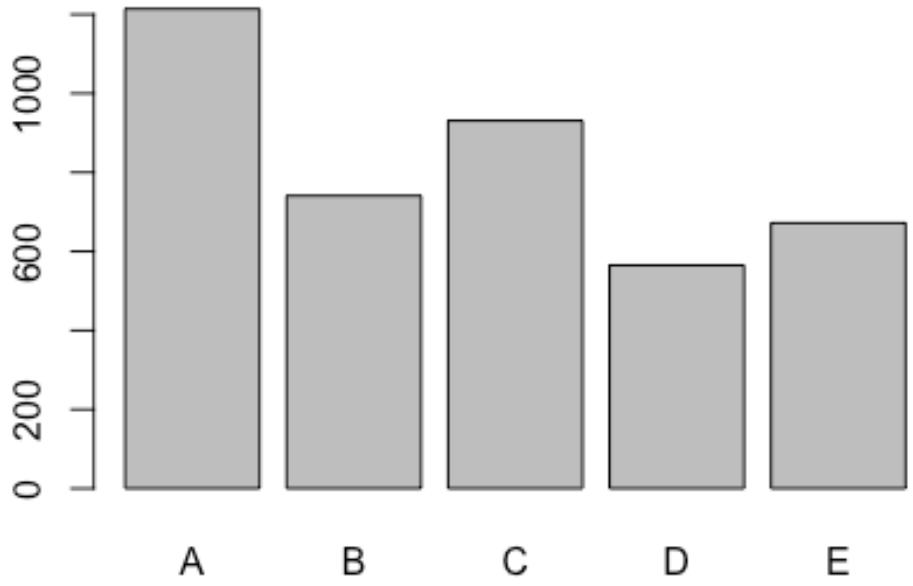
now we will be validate the model

```
modelpre <- predict(tredec, 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
```

```
plot(modelpre)
```



now for the last part we will apply two models one by one the first one will be general boosted model and then the second one will be gbm model for this

```
set.seed(10000)
ctr_gbm <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
valid_gbm <- train(classe ~ .,data=training_data, method = "gbm", trControl =
```

```
ctr_gbm, verbose = FALSE)
valid_gbm$finalModel

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

print(valid_gbm)

## Stochastic Gradient Boosting
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 10990, 10987, 10991, 10990, 10990
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7496535  0.6825583
##   1                  100      0.8198296  0.7720129
##   1                  150      0.8507676  0.8111914
##   2                   50      0.8515680  0.8120275
##   2                  100      0.9028175  0.8770250
##   2                  150      0.9272049  0.9078823
##   3                   50      0.8926251  0.8640870
##   3                  100      0.9388508  0.9226183
##   3                  150      0.9587981  0.9478755
##
```

So, in this project, we tried to predict the order wise the someone did the exercise, and then we created the analysis in which we did some cross-validation and why I chose this specific way towards approaching and then predicted for 20. and i have attached the link to GitHub, which contained the HTML and rmd file. Still, due to some unprecedented reason, 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, and thank you!!!..... it was a great experience learning many things thank you, mentors and university

.....