The following Libraries were used for this project

library(rattle)

library(caret)

library(rpart)

library(rpart.plot)

library(corrplot)

library(randomForest)

library(RColorBrewer)

Set seed:

set.seed(56789)

Download the dataset:

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

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

trainFile <- "./data/pml-training.csv"

testFile <- "./data/pml-testing.csv"

if (!file.exists("./data")) {

dir.create("./data")

}

if (!file.exists(trainFile)) {

download.file(trainUrl, destfile = trainFile, method = "curl")

}

if (!file.exists(testFile)) {

download.file(testUrl, destfile = testFile, method = "curl")

}

rm(trainUrl)

rm(testUrl)

Reading the Data:

trainRaw <- read.csv(trainFile)

testRaw <- read.csv(testFile)

dim(trainRaw)

dim(testRaw)

rm(trainFile)

rm(testFile)

Data Cleaning:

1.Non Zero Variance :

NZV <- nearZeroVar(trainRaw, saveMetrics = TRUE)

head(NZV, 20)

training01 <- trainRaw[, !NZV$nzv]

testing01 <- testRaw[, !NZV$nzv]

dim(training01)

dim(testing01)

rm(trainRaw)

rm(testRaw)

rm(NZV)

2. Unnecessary columns:

regex <- grepl("^X|timestamp|user\_name", names(training01))

training <- training01[, !regex]

testing <- testing01[, !regex]

rm(regex)

rm(training01)

rm(testing01)

dim(training)

dim(testing)

3.NA’s:

cond <- (colSums(is.na(training)) == 0)

training <- training[, cond]

testing <- testing[, cond]

rm(cond)

Correlation Matrix:

corrplot(cor(training[, -length(names(training))]), method = "color", tl.cex = 0.5)

Split the dataset:

set.seed(56789) # For reproducibile purpose

inTrain <- createDataPartition(training$classe, p = 0.70, list = FALSE)

validation <- training[-inTrain, ]

training <- training[inTrain, ]

rm(inTrain)

Decision Tree Algorithm:

modelTree <- rpart(classe ~ ., data = training, method = "class")

prp(modelTree)

Estimate performance on validation data set:

predictTree <- predict(modelTree, validation, type = "class")

confusionMatrix(validation$classe, predictTree)

accuracy <- postResample(predictTree, validation$classe)

ose <- 1 - as.numeric(confusionMatrix(validation$classe, predictTree)$overall[1])

rm(predictTree)

rm(modelTree)