**5. Problem Statement**

1. Use the below given data set

[DataSet](https://archive.ics.uci.edu/ml/datasets/Weight+Lifting+Exercises+monitored+with+Inertial+Measurement+Units)

|  |  |
| --- | --- |
|  | #1. Use the below given data set |
|  | #DataSet |
|  | cs2m <- read.csv("D:\\BIG DATA\\DATA ANALYTICS WITH R, EXCEL & TABLEAU\\17 ENSEMBLE MODELS\\cs2m.csv") |
|  | View(cs2m) |
|  | #2. Perform the below given activities: |
|  | #a. Create classification model using different decision trees. |
|  | #b. Verify model goodness of fit. |
|  | #c. Apply all the model validation techniques. |
|  | #d. Make conclusions |
|  |  |
|  | #Answers for a),b),c),d) using above dataset same as assignment 17 |
|  |  |
|  |  |
|  | names(cs2m) |
|  | nrow(cs2m) |
|  | ncol(cs2m) |
|  | str(cs2m) |
|  |  |
|  | #decision tree |
|  | select\_rows<- sample(1:nrow(cs2m),round(0.2\*nrow(cs2m)),replace = F) |
|  | cs2mTest<- cs2m[select\_rows,] |
|  | cs2mTest |
|  | cs2mTrain<- cs2m[-(select\_rows),] |
|  | cs2mTrain |
|  |  |
|  | library(tree) |
|  | modelRegTree<- tree(cvtd\_timestamp~classe+total\_accel\_belt+yaw\_dumbbell+roll\_forearm+accel\_forearm\_y,data = cs2mTrain) |
|  | plot(modelRegTree) |
|  |  |
|  | text(modelRegTree,pretty = 0 ,cex=0.75) |
|  |  |
|  | pred<- predict(modelRegTree,newdata= cs2mTest) |
|  | head(pred,3) |
|  |  |
|  | ME<- sum(cs2mTest$cvtd\_timestamp - pred)/nrow(cs2mTest) |
|  | ME |
|  |  |
|  | RSS<- sum(cs2mTest$cvtd\_timestamp-pred)^2 |
|  | RSS |
|  |  |
|  | RMSE<- sqrt(RSS/nrow(cs2mTest)) |
|  | RMSE |
|  |  |
|  | MAPE<- sum(abs(cs2mTest$cvtd\_timestamp-pred)/cs2mTest$BP)\*100 |
|  | MAPE |
|  |  |
|  | #one more |
|  | library(tree) |
|  | modelRegTree1<- tree(classe~cvtd\_timestamp+total\_accel\_belt+yaw\_dumbbell+roll\_forearm+accel\_forearm\_y,data = cs2mTrain) |
|  | plot(modelRegTree1) |
|  |  |
|  | text(modelRegTree1,pretty = 0 ,cex=0.75) |
|  |  |
|  | pred<- predict(modelRegTree1,newdata= cs2mTest) |
|  | head(pred,3) |
|  |  |
|  | ME<- sum(cs2mTest$classe - pred)/nrow(cs2mTest) |
|  | ME |
|  |  |
|  | RSS<- sum(cs2mTest$classe-pred)^2 |
|  | RSS |
|  |  |
|  | RMSE<- sqrt(RSS/nrow(cs2mTest)) |
|  | RMSE |
|  |  |
|  | MAPE<- sum(abs(cs2mTest$classe-pred)/cs2mTest$classe)\*100 |
|  | MAPE |
|  |  |
|  | #classification |
|  | library(caTools) |
|  | library(tree) |
|  | #splitting |
|  | set.seed(1) |
|  | split<- sample.split(cs2m$classe,SplitRatio = 0.70) |
|  | cs2mTrain <- subset(cs2m,split == TRUE) |
|  | cs2mTest<- subset(cs2m, split == FALSE) |
|  |  |
|  | table(cs2m$classe) |
|  |  |
|  | table(cs2mTrain$classe) |
|  |  |
|  | table(cs2mTest$classe) |
|  |  |
|  | prop.table(table(cs2mTest$classe)) |
|  |  |
|  | table(cs2mTest$classe) |
|  |  |
|  | prop.table(table(cs2mTrain$classe)) |
|  |  |
|  | modelClassTree<- tree(classe~cvtd\_timestamp+total\_accel\_belt+yaw\_dumbbell+roll\_forearm+accel\_forearm\_y,data = cs2mTrain) |
|  | plot(modelClassTree) |
|  |  |
|  | text(modelClassTree,pretty = 0 ,cex=0.75) |
|  |  |
|  | pred<- predict(modelClassTree,newdata= cs2mTest) |
|  | head(pred,3) |
|  | cs2m$predict <- predict |
|  | cs2m$predictROUND<- round(predict,digits = 0) |
|  |  |
|  | #confusion matrix |
|  | table(cs2m$classe,predict>= 0.5) |
|  |  |
|  | sum<- sum(table(cs2m$classe,predict>= 0.5)) |
|  |  |
|  | #interpretation, Accuracy and model goodness of our model |
|  | #accuracy of our model |
|  | accuracy<- (1185+679)/(2266) |
|  | accuracy |
|  | #0.8225949 |
|  |  |
|  | #model goodness |
|  | library(verification) |
|  | predictTrain<- predict(model,cs2m,type="response") |
|  | table(cs2m$classe,predictTrain >=0.5) |
|  | head(predictTrain,3) |
|  | auc(cs2m$classe,predictTrain) |
|  |  |
|  | #conclusions |
|  | #\*\*\*\*NOTE\*\*\*\* |
|  | #Area under the curve: 0.9333333 |
|  | #also our accuracy of our model is 0.8225949 |
|  | #also by seeing various measures like ME,RSS,RMSE,MAPE of our tree which is godd |
|  | #by this all things we conclude that our model is good and fit |

3

Data Analytics

**6. Expected Output**

N/A

**7. Approximate Time to Complete Task**

30 mins.

4