a. Predict the no of comments in next H hrs

b. Use regression technique

c. Report the training accuracy and test accuracy

sol= # To predict the comments in next H Hrs , I have built three models :

# One based on Decision Tree and two based on Linear Regression

# Train and Test Accuracy is found for all the three models and

# is observed that decision tree model is better of the three

library(dplyr); library(corrplot);library(car); library(MASS); library(forecast)

# import train data set

Variant\_1 <- read.csv("E:/Acadgild/Class 8/Assignments/Dataset/fbtrain/Features\_Variant\_1.csv", header=FALSE)

Variant\_2 <- read.csv("E:/Acadgild/Class 8/Assignments/Dataset/fbtrain/Features\_Variant\_2.csv", header=FALSE)

Variant\_3 <- read.csv("E:/Acadgild/Class 8/Assignments/Dataset/fbtrain/Features\_Variant\_3.csv", header=FALSE)

Variant\_4 <- read.csv("E:/Acadgild/Class 8/Assignments/Dataset/fbtrain/Features\_Variant\_4.csv", header=FALSE)

Variant\_5 <- read.csv("E:/Acadgild/Class 8/Assignments/Dataset/fbtrain/Features\_Variant\_5.csv", header=FALSE)

fbtrain <- rbind(Variant\_1, Variant\_2, Variant\_3, Variant\_4, Variant\_5)

dim(fbtrain)

# import test data set

setwd("E:/Acadgild/Class 8/Assignments/Dataset/fbtest")

test1 <- read.csv("Test\_Case\_1.csv", header = F); test2 <- read.csv("Test\_Case\_2.csv", header = F)

test3 <- read.csv("Test\_Case\_3.csv", header = F); test4 <- read.csv("Test\_Case\_4.csv", header = F)

test5 <- read.csv("Test\_Case\_5.csv", header = F); test6 <- read.csv("Test\_Case\_6.csv", header = F)

test7 <- read.csv("Test\_Case\_7.csv", header = F); test8 <- read.csv("Test\_Case\_8.csv", header = F)

test9 <- read.csv("Test\_Case\_9.csv", header = F); test10 <- read.csv("Test\_Case\_10.csv", header = F)

fbtest <- rbind(test1, test2, test3, test4, test5, test6, test7, test8, test9, test10)

dim(fbtest)

# Assign variable names to the train and test data set

colnames(fbtrain) <- c("plikes","checkin","talking","category","d5","d6","d7","d8","d9","d10","d11","d12",

"d13","d14","d15","d16","d17","d18","d19","d20","d21","d22","d23","d24","d25","d26",

"d27","d28","d29","cc1","cc2","cc3","cc4","cc5","basetime","postlength","postshre",

"postpromo","Hhrs","sun","mon","tue","wed","thu","fri","sat","basesun","basemon",

"basetue","basewed","basethu","basefri","basesat","target")

colnames(fbtest) <- c("plikes","checkin","talking","category","d5","d6","d7","d8","d9","d10","d11","d12",

"d13","d14","d15","d16","d17","d18","d19","d20","d21","d22","d23","d24","d25","d26",

"d27","d28","d29","cc1","cc2","cc3","cc4","cc5","basetime","postlength","postshre",

"postpromo","Hhrs","sun","mon","tue","wed","thu","fri","sat","basesun","basemon",

"basetue","basewed","basethu","basefri","basesat","target")

dim(fbtrain); dim(fbtest)

View(fbtrain); View(fbtest)

str(fbtrain); str(fbtest)

train <- fbtrain; test <- fbtest

head(train); head(test)

distinct(train) # removing overlapping observations if any

dim(train)

colSums(is.na(train)) # no missing values

#-------------------------------------------------------------------

# Predict the no of comments in next H hrs

#-------------------------------------------------------------------

# using decision tree

library(rpart)

fit <- rpart(target ~ ., data = train)

summary(fit)

# Predict Output

# predictions for test data

prediction3 <- predict(fit, test)

predicted3 <- data.frame(cbind(actuals = test$target, prediction = round(prediction3)))

cor(predicted3)

View(predicted3)

# test accuracy

round(accuracy(predicted3$prediction,predicted3$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set -1.682 76.935 22.45 -Inf Inf

# predictions for train data

prediction3 <- predict(fit, train)

predicted3 <- data.frame(cbind(actuals = train$target, prediction = round(prediction3)))

cor(predicted)

View(predicted)

# train accuracy

round(accuracy(predicted3$prediction,predicted3$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set 0.381 23.629 5.474 -Inf Inf

######################################################################

library(outliers)

train\_out <- rm.outlier(train, fill = TRUE, median = TRUE)

colSums(is.na(train\_out))

TARGET <- lm(target~., data = train\_out)

library(MASS)

#step <- stepAIC(TARGET, direction = "both")

final\_model <- lm(target ~ checkin + talking + d5 + d6 + d7 + d8 + d9 + d10 + d12 +

d13 + d14 + d17 + d18 + d19 + d21 + d22 + d23 + d24 + d25 +

d26 + d28 + d29 + cc1 + cc2 + cc3 + cc4 + basetime + postshre +

Hhrs + tue + wed + thu + fri + basesun + basemon + basetue +

basewed + basethu, data = train\_out[,-38])

summary(final\_model)

# Fine tune the model and represent important features

final\_model <- lm(target ~ checkin + talking + d5 + d6 + d7 + d8 + d10 + d12 +

d13 + d17 + d18 + d19 + d22 + d23 + d25 +

d26 + d28 + d29 + cc2 + cc3 + cc4 + basetime + postshre +

Hhrs, data = train\_out[,-38])

summary(final\_model)

# predictions for test data

prediction <- predict(final\_model, test)

predicted <- data.frame(cbind(actuals = test$target, prediction = prediction))

predicted$prediction <- ifelse(prediction<0, 0, prediction)

cor(predicted)

# test accuracy

round(accuracy(predicted$prediction,predicted$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set 4.201 93.293 23.504 -Inf Inf

# training accuracy

prediction <- predict(final\_model, train)

predicted <- data.frame(cbind(actuals = train$target, prediction = prediction))

predicted$prediction <- ifelse(prediction<0, 0, prediction)

cor(predicted)

round(accuracy(predicted$prediction,predicted$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set -1.08 28.119 6.834 -Inf Inf

par(mfrow=c(2,2))

plot(final\_model)

##################################################################

final\_model1 <- lm(target ~ checkin + talking + d5 + d6 + d7 + d8 + d10 + d12 +

d13 + d17 + d18 + d19 + d22 + d23 + d25 +

d26 + d28 + d29 + cc2 + cc3 + cc4 + basetime + postshre +

Hhrs, data = train)

summary(final\_model1)

# predictions for test data

prediction1 <- predict(final\_model1, test)

predicted1 <- data.frame(cbind(actuals = test$target, prediction = prediction1))

predicted1$prediction <- ifelse(prediction<0, 0, prediction)

cor(predicted1)

# test accuracy

round(accuracy(predicted1$prediction,predicted1$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set 4.417 94.631 23.614 -Inf Inf

# training accuracy

prediction1 <- predict(final\_model1, train)

predicted1 <- data.frame(cbind(actuals = train$target, prediction = prediction1))

predicted1$prediction <- ifelse(prediction<0, 0, prediction)

cor(predicted1)

round(accuracy(predicted1$prediction,predicted1$actuals),3)

# ME RMSE MAE MPE MAPE

# Test set 0 28.085 7.976 NaN Inf

par(mfrow=c(2,2))

plot(final\_model1)