Problem- prediction of the number of comments in the upcoming 24 hours on

those blogs, The train data was generated from different base times that may

temporally overlap. Therefore, if you simply split the train into disjoint partitions,

the underlying time intervals may overlap. Therefore, the you should use the

provided, temporally disjoint train and test splits to ensure that the evaluation is

fair.

**a.Read the dataset and identify the right features**

sol= library(dplyr); library(corrplot);library(car); library(MASS); library(ggplot2)

library(reshape2); library(forecast)

# Q1- read the dataset and identify the right features

# import train data set

Variant\_1 <- read.csv("E:/Acadgild/Class 7/Assignments/Dataset/fbtrain/Features\_Variant\_1.csv", header=FALSE)

Variant\_2 <- read.csv("E:/Acadgild/Class 7/Assignments/Dataset/fbtrain/Features\_Variant\_2.csv", header=FALSE)

Variant\_3 <- read.csv("E:/Acadgild/Class 7/Assignments/Dataset/fbtrain/Features\_Variant\_3.csv", header=FALSE)

Variant\_4 <- read.csv("E:/Acadgild/Class 7/Assignments/Dataset/fbtrain/Features\_Variant\_4.csv", header=FALSE)

Variant\_5 <- read.csv("E:/Acadgild/Class 7/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 7/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)

# making the data tidy by constructing single collumn for post publish day

train$pubday<- ifelse(train$sun ==1, 1, ifelse(train$mon ==1, 2, ifelse(train$tue ==1, 3,

ifelse(train$wed ==1, 4, ifelse(train$thu ==1, 5, ifelse(train$fri ==1, 6,

ifelse(train$sat ==1, 7, NA)))))))

# making the data tidy by constructing single collumn for base day

train$baseday<- ifelse(train$basesun ==1, 1, ifelse(train$basemon ==1, 2, ifelse(train$basetue ==1, 3,

ifelse(train$basewed ==1, 4, ifelse(train$basethu ==1, 5,

ifelse(train$basefri ==1, 6, ifelse(train$basesat ==1, 7, NA)))))))

# now the data set is ready

**b. Clean dataset, impute missing values and perform exploratory data analysis.**

Sol= distinct(train) # removing overlapping observations if any

dim(train)

sapply(train, function(x) sum(is.na(x))) # no missing values

correlation <- cor(train[,c("target", "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","pubday","baseday")])

corr <- as.data.frame(reshape::melt(correlation))

corr <- corr%>%filter(X1 == "target" & value != 1 & value > 0.32 & value > -0.32)

corr # good corelations with target variable

corrplot.mixed(cor(train[,c(30:34)]))

# Total comments are strongly correlated to correlated with cc4(comments in first 24 hrs of publish time) and

# cc3(comments in last 48 to last 24 hours relative to base date/time)

df <- train

melt\_df <- melt(df)

# Distribution of all the Variables - Histogram

ggplot(melt\_df, aes(x=value, fill = variable))+

geom\_histogram(bins=10, color = "Blue")+

facet\_wrap(~variable, scales = 'free\_x')

df <- log(train[1:39])

par(mfrow=c(1,1))

**c. Visualize the dataset and make inferences from that**

sol= barplot(table(train$target, train$pubday), col = heat.colors(7),

xlab = "Weekday", ylab = "Number of comments",

main = "Number of comments Vs. Weekday")

# post published on Wednesday has maximum comments

# number of comments vs Post Likes

scatterplot(train$plikes, train$target , col = "Blue",

xlab = "Page Likes", ylab = "Number of comments",

main = "Number of comments Vs. Pagelikes",

xlim = c(0,10000000), ylim = c(0,400))

abline(lm(plikes~target, data = train), col = "red")

# as the page likes increases the comments are not increasing

# Number of comments Vs Post length

scatterplot(train$postlength, train$target , col = "Red",

xlab = "Post Length", ylab = "Number of comments",

main = "Number of comments Vs. Psot Length",

ylim = c(0,400), xlim = c(0,5000))

abline(lm(postlength~target, data = train), col= "blue")

# as the page lenth is increasing the number of comments decreases

hist(train$target, breaks = 1000, xlim = c(0,10) )

# data is very positively skewed. Very less comments after base time

**d. Perform any 3 hypothesis tests using columns of your choice, make conclusions**

sol= # Ho: Mean difference bet comments across the publish day is not significant

day <- aov(target~pubday, data = train)

summary(day)

# Comments are dependent on day of publish

# Ho: Mean difference in comments across the target and cc4 is not significant

cc4 <- t.test(train$target, train$cc4, paired = FALSE, alternative = "two.sided", mu=0)

cc4

# Difference between the number of comments after H hrs and

# comments in first 24 hrs of publish is significant

# Ho: Difference between Mean comments within cc2 and cc4 is not significant

cc2 <- t.test(x=train$cc2, y=train$cc4, paired = FALSE, alternative = "two.sided", mu=0)

cc2

# Difference between the number of comments in last 24 hrs of base time and

# comments in first 24 hrs of publish is significant

**e. Create a linear regression model to predict the number of comments in the next 24 hours**

**(relative to basetime)**

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

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

final\_model <- lm(target ~ checkin + talking + d5 + d6 + d7 + d8 + d9 + d10 + d11 +

d12 + d13 + d16 + d17 + d19 + d20 + d21 + d22 + d23 + d24 +

cc1 + cc2 + cc3 + cc4 + basetime + postshre + Hhrs + wed +

thu + fri + basemon + basewed, data = train)

summary(final\_model)

**f. Fine tune the model and represent important features**

sol= final\_model <- lm(target ~ talking + d5 + d7 + d8 + d10 + d11 +

d12 + d13 + d16 + d17 + d19 + d20 + d22 + d23 +

cc1 + cc2 + cc3 + cc4 + basetime + postshre + Hhrs, data = train)

summary(final\_model)

prediction <- predict(final\_model, test)

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

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

cor(predicted)

View(predicted)

**g. Interpret the summary of the linear model**

sol= # Residual error is distributed between -346.83 to 1271.33

# P-value of the model is less than alpha (0.05), hence we can accept the model

# 32.46% variability is represented by the model

**h. Report the test accuracy vs. the training accuracy**

sol= # test accuracy

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

prediction <- predict(final\_model, test)

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

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

min\_max\_accuracy <- mean(apply(predicted, 1, min) / apply(predicted, 1, max))

min\_max\_accuracy

# training accuracy

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

prediction <- predict(final\_model, train)

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

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

min\_max\_accuracy <- mean(apply(predicted, 1, min) / apply(predicted, 1, max))

min\_max\_accuracy

**i.Interpret the final model coefficients**

sol= summary(final\_model)

coef(final\_model) # coefficients of the model

#comments in H Hrs has slope with Independent variables as below:

# talking d5 d7 d8 d10 d11

# -1.858115e-05 -4.759496e-01 8.609203e-01 1.675394e-01 -1.239555e-01 -2.236221e-03

# d12 d13 d16 d17 d19 d20 d22

# 1.612318e-01 1.276223e-01 1.114969e-02 1.085186e-01 -1.165972e-01 4.201675e-01 -8.837498e-01

# d23 cc1 cc2 cc3 cc4 basetime postshre

# -2.159461e-01 4.338324e-02 2.196493e-01 -2.272725e-02 -6.728051e-02 -1.933110e-01 2.921963e-03

# Hhrs

# 3.880629e-01

**j. Plot the model result and compare it with assumptions of the model**

sol= par(mfrow=c(2,2))

plot(final\_model)

# Model does not pass the test of normality

# the data is heteroscadatic

# Observations 3528,30608,16432 may have the leverage or potential for influencing the model