**Tree- Image Features: file used- tree\_imagefeaturess1**

#importing datat from csv file

setwd("~/train\_tree")

A<-read.csv('tree\_imagefeaturess1.csv', header=TRUE, sep=",")

#checking the imported datadrame

A[1:5,]

#checking no. of rows in A

nrow(A)

#checking number of columns in A

ncol(A)

#checking summary of stats of dataframe A

summary(A)

#Sampling the data as train and test

#sampling rows randomly

train.rows<- sample(nrow(A), 160)

#assigning the row values for the sampled row-indexes

#train set

train.set <- A[train.rows,]

#test set

test.set<- A[-train.rows,]

#lets check the number of rows and columns in our train set

nrow(train.set)

nrow(test.set)

#Lets develop the Linear Regression Model now

fit <- lm(formula=Class\_label~., data= train.set)

#lets check out the sumamry

summary(fit)

#coefficients and intercepts obtained by the linear model

fit$coefficients

#lets check out the fitted values for our train data

fit$fitted.values

#lets use our model to predict the labels for the test set

predict(fit, test.set)

test.set

#performance on the train set of linear model

#using mean square error

mean\_sq\_er1= mean(fit$residuals^2)

mean\_sq\_er1

#performance on the testset of the linear model

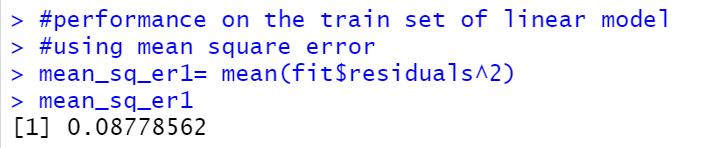
y\_cap= predict(fit, test.set)

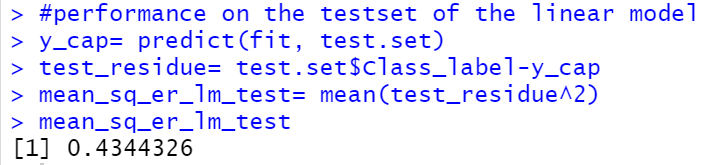
test\_residue= test.set$Class\_label-y\_cap

mean\_sq\_er\_lm\_test= mean(test\_residue^2)

mean\_sq\_er\_lm\_test

**Output:**





**Tree- text Features: file used- tree\_text\_featuresand\_labelss1**

#importing datat from csv file

setwd("~/train\_tree")

A<-read.csv('tree\_text\_featuresand\_labelss1.csv', header=TRUE, sep=",")

#checking no. of rows in A

nrow(A)

#checking number of columns in A

ncol(A)

#checking summary of stats of dataframe A

summary(A)

#Sampling the data as train and test

#sampling rows randomly

train.rows<- sample(nrow(A), 20)

#assigning the row values for the sampled row-indexes

#train set

train.set <- A[train.rows,]

#test set

test.set<- A[-train.rows,]

#lets check the number of rows and columns in our train set

nrow(train.set)

nrow(test.set)

#Lets develop the Linear Regression Model now

fit <- lm(formula=Class\_label~., data= train.set)

#lets check out the sumamry

summary(fit)

#coefficients and intercepts obtained by the linear model

fit$coefficients

#lets check out the fitted values for our train data

fit$fitted.values

#lets use our model to predict the labels for the test set

predict(fit, test.set)

test.set

#performance on the train set of linear model

#using mean square error

mean\_sq\_er1= mean(fit$residuals^2)

mean\_sq\_er1

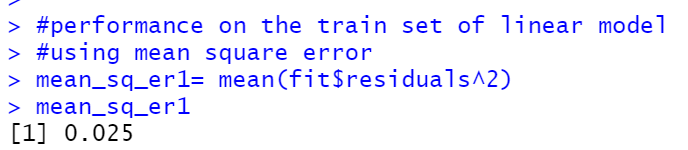
#performance on the testset of the linear model

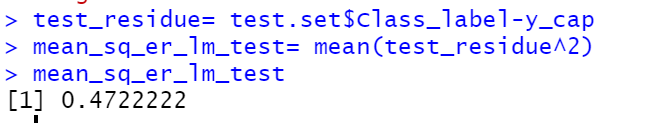
y\_cap= predict(fit, test.set)

test\_residue= test.set$Class\_label-y\_cap

mean\_sq\_er\_lm\_test= mean(test\_residue^2)

mean\_sq\_er\_lm\_test





**Tree- combined Features: file used- tree\_combined\_featuresand\_labelss:**

#importing datat from csv file

setwd("~/train\_tree")

A<-read.csv('tree\_combined\_featuresand\_labelss.csv', header=TRUE, sep=",")

#checking no. of rows in A

nrow(A)

#checking number of columns in A

ncol(A)

#checking summary of stats of dataframe A

summary(A)

#Sampling the data as train and test

#sampling rows randomly

train.rows<- sample(nrow(A), 20)

#assigning the row values for the sampled row-indexes

#train set

train.set <- A[train.rows,]

#test set

test.set<- A[-train.rows,]

#lets check the number of rows and columns in our train set

nrow(train.set)

nrow(test.set)

#Lets develop the Linear Regression Model now

fit <- lm(formula=Label~., data= train.set)

#lets check out the sumamry

summary(fit)

#coefficients and intercepts obtained by the linear model

fit$coefficients

#lets check out the fitted values for our train data

fit$fitted.values

#lets use our model to predict the labels for the test set

predict(fit, test.set)

test.set

#performance on the train set of linear model

#using mean square error

mean\_sq\_er1= mean(fit$residuals^2)

mean\_sq\_er1

#performance on the testset of the linear model

y\_cap= predict(fit, test.set)

test\_residue= test.set$Label-y\_cap

mean\_sq\_er\_lm\_test= mean(test\_residue^2)

mean\_sq\_er\_lm\_test

