```
getwd()

data<-read.csv("/home/sofikul/Desktop/opjindal ml project/diabetic data analysis using logistic regression/Diabatic_data.csv")

View(data)

#data<-Diabatic_data  # first read data

library(caTools)     # load this library for split function

split<- sample.split(data, SplitRatio = 0.8)  # split data into training and testing with ratio 8:2 respectively

split

# execute this command 0.8 will be TRUE value(training data set) abd 0.2 will be FALSE or testing data set

training<- subset(data, split=="TRUE")

testing<- subset(data, split=="FALSE")

model<- glm(type~., training, family = "binomial")  # glm() for logistic regression here, y= "type" and x= (.) means all variable except(y= type(dependent var))

summary(model) # summary of the model

model<- glm(type~. -age, training, family = "binomial") # discard age var to test is model will be  more acqurate or not.

summary(model) # no need to dicard/ delet age from model


# VVI>> Residual deviance:value shuld not increase and AIC: value shuld decrease if these both are happen or true then variable removal is right or ok.

# age is a significant variable hence it can't be remove

model<- glm(type~. -age, training, family = "binomial") # check for skin whether this var is needed or not.

summary(model) # skin need to remove from model

model<- glm(type~. -his, training, family = "binomial") # check for his variable need to delete or not

summary(model) # no need to remove (his ) variable because it causes increase Residual deviance and AIC

res<- predict(model, testing, type = "response") # predict the value for the test dataset and the categorize them according to threshold which is 0.5

res # enter
```

```
TAB<-table(testing$type, res>0.05)  ## you can use confusionmatrix insted of table fun.
TAB

(12+62)/(12+94+2+62)=0.4352941

table(Actualvalue=testing$type, PredictedvalueBymodel=res>0.05) # confusionMatrix for the testing dataset to test acuracy of the model
(12+62)/(12+94+2+62)=0.4352941

#------------------------------------------------------------------------------------------------#
# other way to find threshold by using ROCR curve which is used to calculate the threshold in your model

res<- predict(model, training, type = "response") # new calculate res with respect to training dataset for ROCR curve

library(ROCR) # install ROCR package
ROCRPred= prediction(res, training$type)      # ROCRPREDICTION
ROCRPerf= performance(ROCRPred, "tpr","fpr")  # ROCRPerformance, "tpr"<- true positive rate, "fpr"<- false positive rate

plot(ROCRPerf,colorize=TRUE, print.cutoffs.at=seq(0.1, by=0.1))
plot(ROCRPerf,print.cutoffs.at=seq(0.1, by=0.1))
res<- predict(model, testing, type = "response") # predict the value for the test dataset and the categorize them according to threshold which is 0.5
table(Actualvalue=testing$type, PredictedvalueBymodel=res>0.5) # confusionMatrix for the testing dataset to test for 0.5

table(Actualvalue=testing$type, PredictedvalueBymodel=res>0.3)
(103+30)/(103+20+17+30)  #[1] 0.7823529 acuracy
```