I have a dataset containing family information of married couples, which have around 10 variables & 600+ observations.

Independent variables are ~ gender, age, years married, children, religion etc.

I have one response variable which is number of extra marital affairs.

Now, I want to know what all factor influence the chances of extra marital affair.

Since extra marital affair is a binary variable (either a person will have or not),

so we can fit logistic regression model here to predict the probability of extra marital affair.

install.packages('AER')

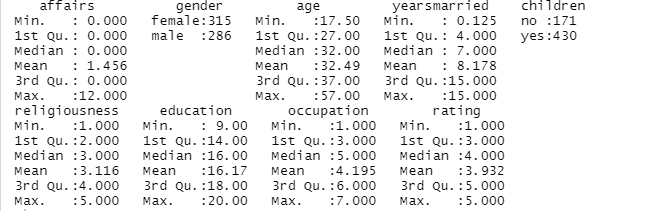
data(Affairs,package="AER")

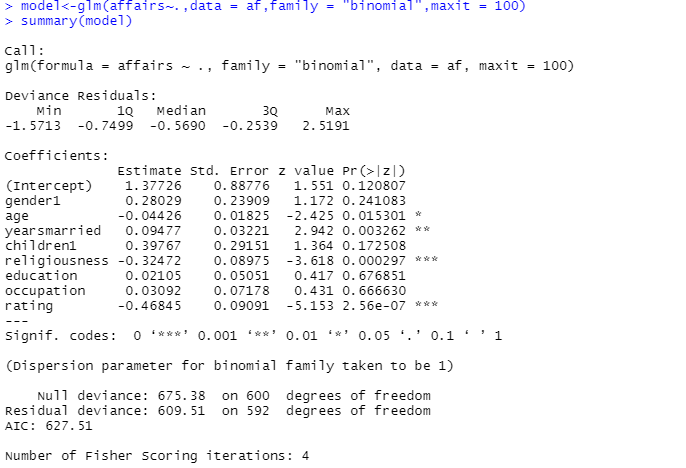
**BUSSINESS PROBLEM:**

**TO CHECK WHAT ALL FACTORS INFLUENCE THE CHANCES OF EXTRA MARITAL**

**AFFAIRS.**

**Summary(affairs)**

****

****

**RESIDUAL DEVIANCE IS LESS THAN NULL DEVIANCE AND DIFFENCE IS LESS**

**IT TELLS THE GOODNESS OF THE MODEL**

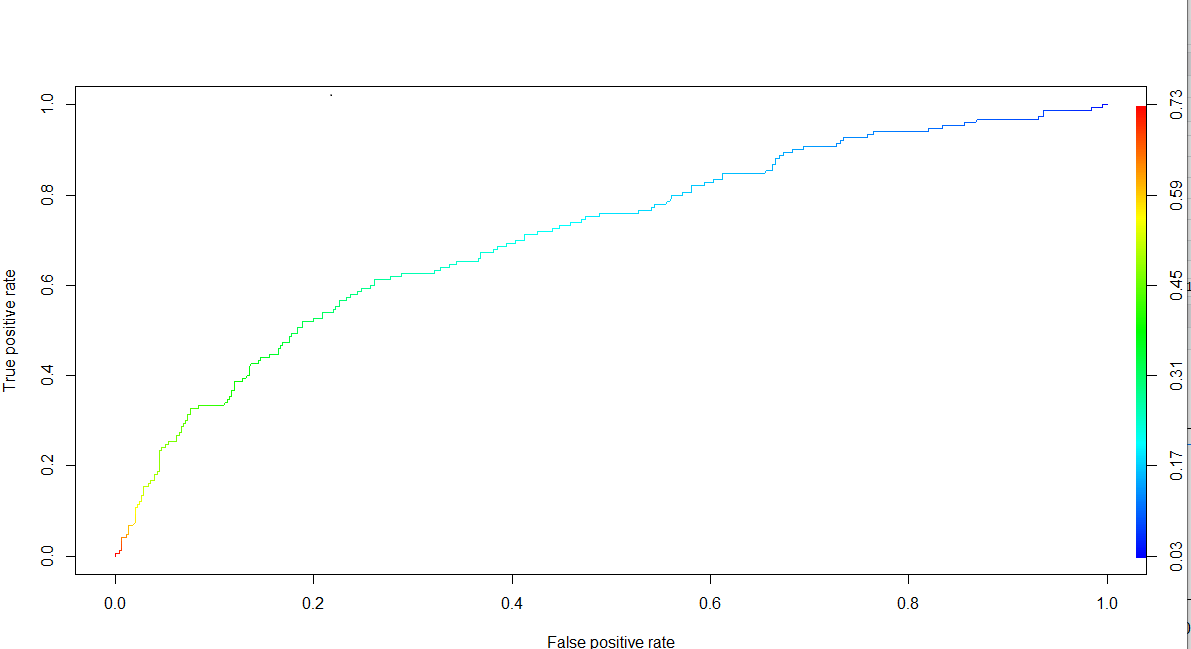
> Accuracy<-sum(diag(confusion)/sum(confusion))

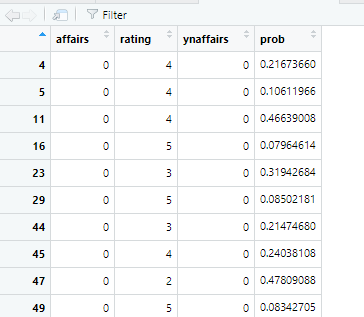
> Accuracy

[1] 0.765391

**-ROC CURVE**

**AREA COVERED GIVES THE ABILITY OF BINARY CLASSIFIER**

**PROBABILITY TABLE**

****

**LIBRARIES USED**

**library(readr)**

**library('AER')**

**library(plyr)**

**library(ROCR)**

**library(dplyr)**

**PYTHON CODE**

**import pandas as pd**

**import numpy as np**

**affairs=pd.read\_csv("C:/Users/USER/Desktop/logistic\_reg/Affairs.csv")**

**affairs1=affairs.iloc[:,1:]**

**affairs1['naffairs'].values[affairs1['naffairs'] > 1] = 1**

**affairs1.info()**

**affairs1.columns**

**#########train -test**

**x=affairs1.iloc[:,1:]**

**y=affairs1.iloc[:,0]**

**from sklearn.model\_selection import train\_test\_split**

**x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size = 0.3)**

**#############**

**from sklearn.linear\_model import LogisticRegression**

**model=LogisticRegression()**

**model=model.fit(x,y)**

**###accuracy of training model**

**model.score(x,y)#######accuracy=0.76**

**y.mean()#####only 24% are having affairs**

**###coeff**

**model.coef\_**

**np.transpose(model.coef\_)**

**#######model validation**

**model1=LogisticRegression()**

**model1=model.fit(x\_train,y\_train)**

**pred=model1.predict(x\_test) ####acc=0.71**

**####probability values**

**prob1=model1.predict\_proba(x\_test)**

**from sklearn import metrics**

**print(metrics.confusion\_matrix(y\_test, pred))**

**print(metrics.classification\_report(y\_test, pred))**

**###predict on train data**

**pred1=model1.predict(x\_train)**

**print(metrics.classification\_report(y\_train,pred1))**

**print(metrics.confusion\_matrix(y\_train, pred1))###acc=0.79**

**######cross-validation**

**from sklearn.model\_selection import cross\_val\_score**

**score=cross\_val\_score(LogisticRegression(),x,y,scoring='accuracy',cv=10)**

**print(score)**

**print(score.mean())##avg acc=0.75**