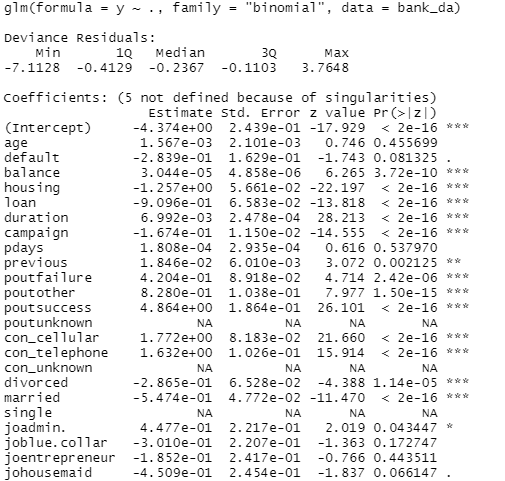
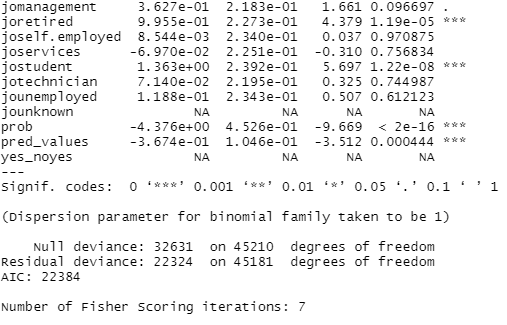
Output variable -> y

y -> Whether the client has subscribed a term deposit or not

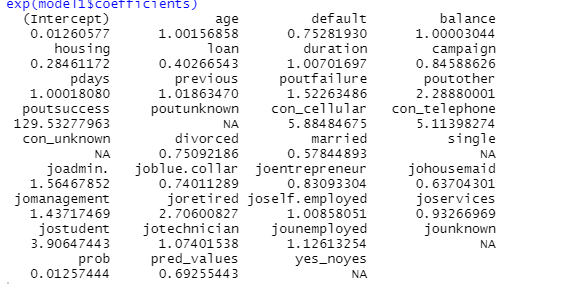
Binomial ("yes" or "no")

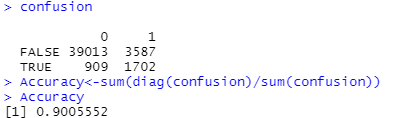
BUSSINESS PROBLEM: To predict on whether the client has subscribed a term deposit or not



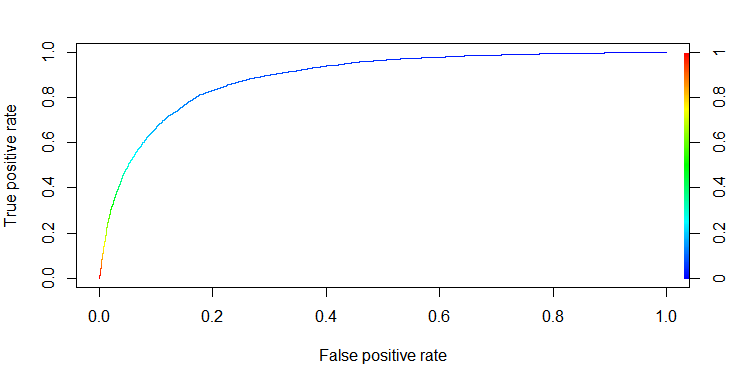


|  |
| --- |
| MODEL1-INFERENCE  Residual deviance is less than null deviance, less the residual deviance better the model.  Most of the input variables are significant  Coefficients are in log |





Roc curve for getting optimal cut-off value



|  |  |  |
| --- | --- | --- |
| MODEL | TRAIN ACCURACY | TEST ACCURACY |
| MODEL1 | 0.90 | 0.89 |

PYTHON CODE

import pandas as pd

import numpy as np

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

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

from sklearn.preprocessing import StandardScaler

bank\_data=pd.read\_csv("C:/Users/USER/Desktop/logistic\_reg/bank\_data.csv")

bank\_data.info()

x=bank\_data.iloc[:,0:31]

y=bank\_data.iloc[:,31]

model=LogisticRegression()

model=model.fit(x,y)

model.score(x,y)#######accuracy=0.89

np.exp(model.coef\_)

###train test check

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)

model1=LogisticRegression()

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

pred=model1.predict(x\_test)

prob=model1.predict\_proba(x\_test)

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

print(metrics.classification\_report(y\_test, pred))###accuracy=0.89

###train data

pred1=model1.predict(x\_train)

print(metrics.confusion\_matrix(y\_train, pred1))

print(metrics.classification\_report(y\_train, pred1))##accuracy=0.891

######checking model accuracy with 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\_score=0.88 hence good model