1. You need to predict the type of program a student is in based on other attributes --> mdata.csv

prog: is a categorical variable indicating what type of program a student is in: �General� (1), �Academic� (2), or �Vocational� (3)

Ses:�is a categorical variable indicating someone�s socioeconomic class: �Low� (1), �Middle� (2), and �High� (3)

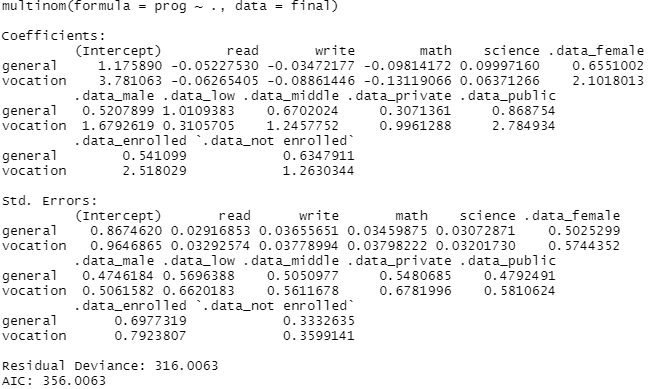
read,write,math,science: is their scores on different tests

honors: Whether they have enrolled or not

BUSSINESS PROBLEM: Need to predict the type of program student is bases

model<-multinom(prog~.,data = final)

summary(model)

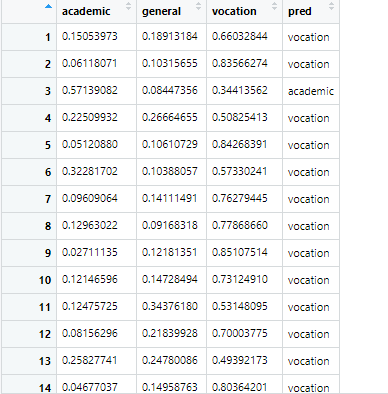


\*\*\*INTERPERTATION:

ACADEMIC IS TAKEN AS BASLINE

MATH-VALUE OF GENERAL(-0.09)INTERPRITATION- WITH INCREASE IN 1 UNIT OF MATH VALUE THE NEGATIVE IMPACT OF -0.098 WILL INCREASE IN GENERAL OVER ACADMIC.

PREDICTED PROBABILITY WITH PRED NAMES



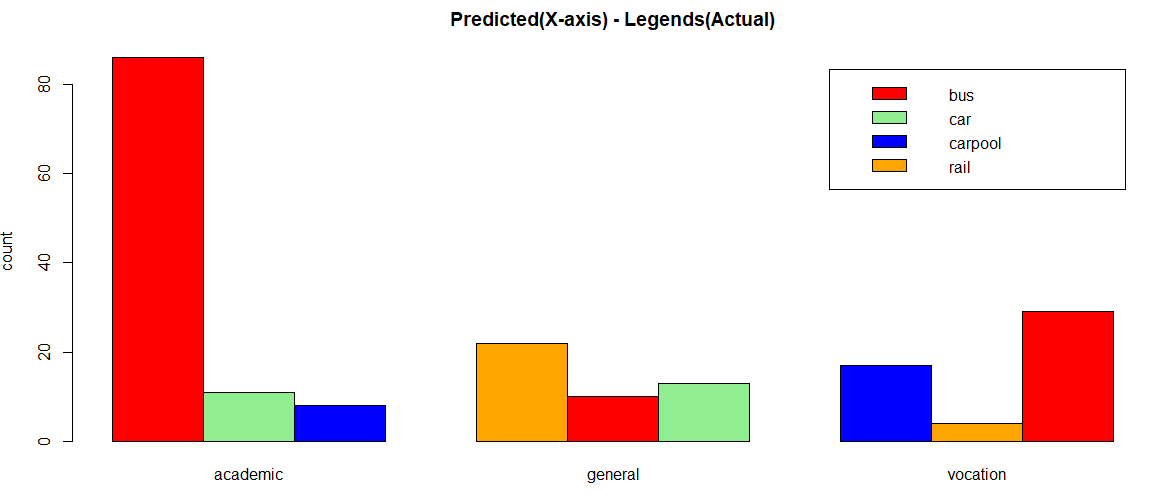
table(pred\_name,prgrm1$prog)

pred\_name academic general vocation

academic 86 22 17

general 11 10 4

vocation 8 13 29



Predicted and actual graph

mean(pred\_name==prgrm1$prog)

####accuracy=0.62

Libraries used

library(readr)

library(plyr)

library(nnet)

library(fastDummies)

python code

import pandas as pd

import seaborn as sns

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

###get file

prgm=pd.read\_csv("C:/Users/USER/Desktop/multinomial\_reg/mdata.csv")

###remove unwanted columns

drp\_prgm=prgm.drop(['Unnamed: 0','id'],axis='columns')

###rename columns

drp\_prgm.columns="prog","ses","schtyp","gender","read","write","math","science","honors"

####swap columns

c = drp\_prgm.columns

drp\_prgm[[c[0], c[3]]] = drp\_prgm[[c[3], c[0]]]

###create dummy variables

dum1=pd.get\_dummies(drp\_prgm.honors)

dum2=pd.get\_dummies(drp\_prgm.gender)

dum3=pd.get\_dummies(drp\_prgm.ses)

dum4=pd.get\_dummies(drp\_prgm.schtyp)

###merge dummies and main data frame

merged=pd.concat([drp\_prgm,dum1,dum2,dum3,dum4],axis='columns')

merged.columns

final=merged.drop(['ses','schtyp','gender','honors','middle'],axis='columns')

##split data

train,test = train\_test\_split(final,test\_size = 0.2)

###building model

model = LogisticRegression(multi\_class="multinomial",solver="newton-cg").fit(train.iloc[:,1:],train.iloc[:,0])

train\_predict = model.predict(train.iloc[:,1:]) # Train predictions

test\_predict = model.predict(test.iloc[:,1:])

accuracy\_score(train.iloc[:,0],train\_predict) # 69.8%

# Test accuracy

accuracy\_score(test.iloc[:,0],test\_predict) # 67.032%