**NAIVE BAYES ALGORITHM**

**""" USER DEFINED FUNCTION"""**

def Naive\_Bayes(DV,IDV):

from sklearn.naive\_bayes import BernoulliNB

AU = []

#spliting record of 70 training/ 30 testing

IDV\_train,IDV\_test,DV\_train,DV\_test=train\_test\_split(IDV,DV,test\_size=0.3,random\_state=0)

# Applyin naive bayes algorithm

clf=BernoulliNB()

DV\_pred=clf.fit(IDV\_train,DV\_train).predict(IDV\_test).T

AU.append(accuracy\_score(DV\_test,DV\_pred,normalize=True))

AU.append(confusion\_matrix(DV\_test,DV\_pred))

#print("x =",x)

return AU

**"""ENDS HERE """**

**“””IMPORTING FILES “””**

import pandas as pd

dataset=pd.read\_csv("train.csv")

dataset.columns

**"""** CPOLUMNS DISPLAYED

Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',

'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],

dtype='object')

"""

**""" IMporting all sklearn methods"""**

from sklearn import preprocessing

**#here cross\_validation not working so use alternative i.e model\_selection**

"""from sklearn.cross\_validation import train\_test\_split

File "<ipython-input-7-d05cc6ee7707>", line 1, in <module>

from sklearn.cross\_validation import train\_test\_split

ModuleNotFoundError: No module named 'sklearn.cross\_validation'

"""

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

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

**""" preprocessing """**

#1)

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

""" Converting into numerical"""

dataset1=le.transform(dataset["Sex"])

dataset["Sex"]=dataset1

#2)

le=preprocessing.LabelEncoder()

le.fit(dataset["Embarked"])

""" Converting into numerical"""

dataset1=le.transform(dataset["Embarked"])

dataset["Embarked"]=dataset1

**""" Droping Uneccessary columns """**

Real\_data=dataset.drop(["Cabin","Name","PassengerId","Ticket"],axis=1)

**""" Iterator """**

#l=['Survived','Pclass','Sex','Age','SibSp','Parch','Fare','Embarked']

n=len(Real\_data.columns)

abh=Real\_data.columns

#LISTS to hold accuracy values and Confusion matrix values

acc\_list=[]

**#Iterator**

for x in range(n):

#print("x =",x)

str1=abh[x]

#print(str1)

DV1=Real\_data[str1]

#print(DV)

IDV1=Real\_data.drop(str1,axis=1)

#print(IDV)

acc\_list.append(Naive\_Bayes(DV1,IDV1))

**RESULT:**

**Accuracy and Confusion Matrix**

