**Naïve Bayes : [Classification Technique]**

**#Importing Packages**

import pandas as pd

from sklearn import preprocessing

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

from sklearn.naive\_bayes import \*

**#Loading dataset**

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

dataset.columns

Out[10]:

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

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

dtype='object')

**#Removing unwanted columns**

dataset=dataset.drop(['PassengerId','Name','Ticket','Cabin'],axis=1)

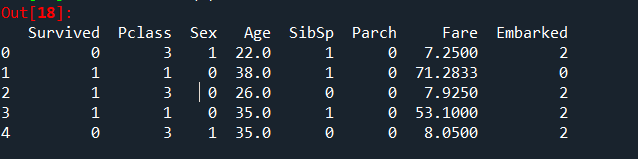
**#Converting text into numerical**

le=preprocessing.LabelEncoder()

dataset['Sex'] = le.fit\_transform(dataset['Sex'])

dataset['Embarked'] = le.fit\_transform(dataset['Embarked'])

dataset.head(5)



**#Implementing Naïve Bayes Theorum**

def Naive\_bayesT(X,y):

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)

clf=BernoulliNB()

y\_pred=clf.fit(X\_train,y\_train).predict(X\_test)

acc\_score=accuracy\_score(y\_test,y\_pred,normalize=True)

conf\_matrix=confusion\_matrix(y\_test,y\_pred)

return acc\_score,conf\_matrix

dependent\_variable=['Pclass','SibSp','Sex', 'Parch','Embarked']

for i in dependent\_variable:

y=dataset[i]

X=dataset.drop(i,axis=1)

acc\_score,conf\_matrix=Naive\_bayesT(X,y)

print("Accuracy Score for",i,"vs Remaining=",acc\_score\*100,"%\n")

print("Confusion Matrix:")

print(conf\_matrix,"\n\n")

**#Result of Accuracy score and confusion matrix**

