**Naïve Bayes Algorithm**

Inference : Calculating accuracy score when Pclass is dependent variable

import pandas as pd

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

from sklearn.metrics import confusion\_matrix

from sklearn.naive\_bayes import \*

from sklearn.metrics import accuracy\_score

from sklearn import preprocessing

dataset=pd.read\_csv("C:/Users/PC/Downloads/train.csv")

encoder=preprocessing.LabelEncoder()

dataset["Sex"]=encoder.fit\_transform(dataset["Sex"])

dataset["Sex"]=encoder.fit\_transform(dataset["Sex"])

0 3

1 1

2 3

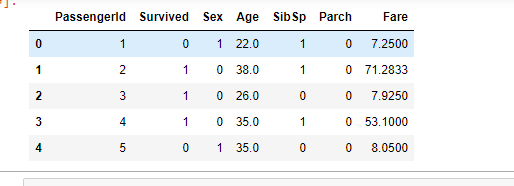
3 1

4 3

Name: Pclass, dtype: int64

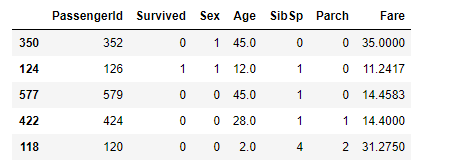
X=dataset.drop(columns=['Pclass','Name','Ticket','Embarked','Cabin'])

X.head()



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

X\_train.head()



clf=BernoulliNB()

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

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

0.5617977528089888

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

array([[ 27, 8, 35],

[ 15, 4, 30],

[ 24, 5, 119]], dtype=int64)

Accuracy score is 56% when Pclass is taken as output variable

**Now Dependent Variable is Gender**

import pandas as pd

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

from sklearn.metrics import confusion\_matrix

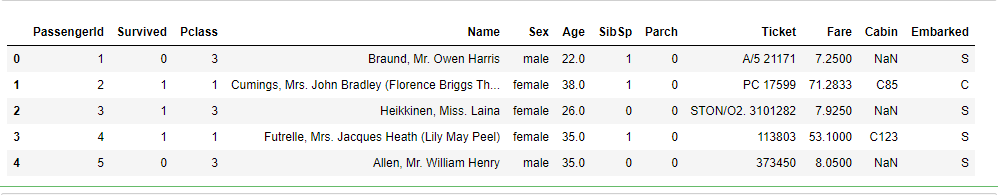
from sklearn.naive\_bayes import \*

from sklearn.metrics import accuracy\_score

from sklearn import preprocessing

dataset1=pd.read\_csv("C:/Users/PC/Downloads/train.csv")

dataset1.head()



l=preprocessing.LabelEncoder()

dataset1['Sex']=l.fit\_transform(dataset1['Sex'])

y1=dataset1['Sex']

y1.head()

0 1

1 0

2 0

3 0

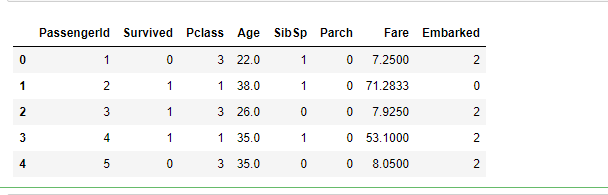
4 1

Name: Sex, dtype: int32

dataset1['Embarked']=l.fit\_transform(dataset1['Embarked'])

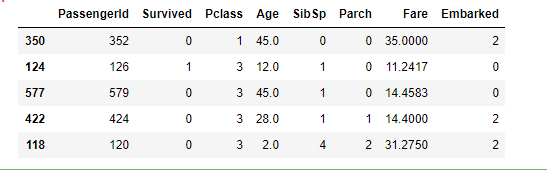
x1=dataset1.drop(columns=['Sex','Name','Cabin','Ticket'])

x1.head()



X1\_train,X1\_test,y1\_train,y1\_test=train\_test\_split(x1,y1,test\_size=0.3,random\_state=0)

X1\_train.head()



cl=BernoulliNB()

y\_pred=cl.fit(X1\_train,y1\_train).predict(X1\_test)

accuracy\_score(y1\_test,y\_pred,normalize=True)

0.7453183520599251

confusion\_matrix(y1\_test,y\_pred)

array([[ 49, 49],

[ 19, 150]], dtype=int64)

In [ ]:

Accuracy Score is 74.5%

**Now Dependent Variable is SIBSP**

l=preprocessing.LabelEncoder()

dataset1['Sex']=l.fit\_transform(dataset1['Sex'])

dataset1['Embarked']=l.fit\_transform(dataset1['Embarked'])

y1=dataset1['SibSp']

y1.head()

0 1

1 1

2 0

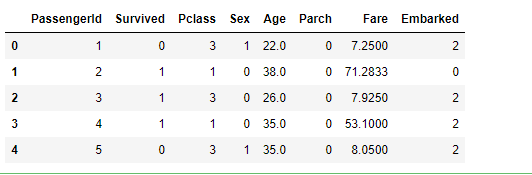
3 1

4 0

Name: SibSp, dtype: int64

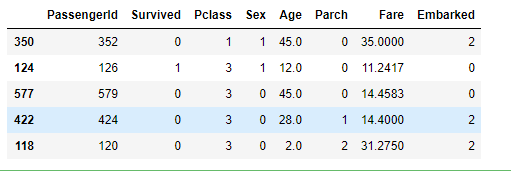
x1=dataset1.drop(columns=['SibSp','Name','Cabin','Ticket'])

x1.head()



X1\_train,X1\_test,y1\_train,y1\_test=train\_test\_split(x1,y1,test\_size=0.3,random\_state=0)

X1\_train.head()



cl=BernoulliNB()

y\_pred=cl.fit(X1\_train,y1\_train).predict(X1\_test)

accuracy\_score(y1\_test,y\_pred,normalize=True)

0.6891385767790262

confusion\_matrix(y1\_test,y\_pred)

array([[162, 20, 0, 0, 0, 0, 0],

[ 43, 22, 0, 0, 0, 0, 0],

[ 6, 2, 0, 0, 0, 0, 0],

[ 5, 2, 0, 0, 0, 0, 0],

[ 2, 0, 0, 0, 0, 0, 0],

[ 1, 0, 0, 0, 0, 0, 0],

[ 2, 0, 0, 0, 0, 0, 0]], dtype=int64)

**Now Embarked is output variable**

import pandas as pd

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

from sklearn.metrics import confusion\_matrix

from sklearn.naive\_bayes import \*

from sklearn.metrics import accuracy\_score

from sklearn import preprocessing

dataset1=pd.read\_csv("C:/Users/PC/Downloads/train.csv")

dataset1.head()

l=preprocessing.LabelEncoder()

dataset1['Sex']=l.fit\_transform(dataset1['Sex'])

dataset1['Embarked']=l.fit\_transform(dataset1['Embarked'])

y1=dataset1['Embarked']

y1.head()

0 2

1 0

2 2

3 2

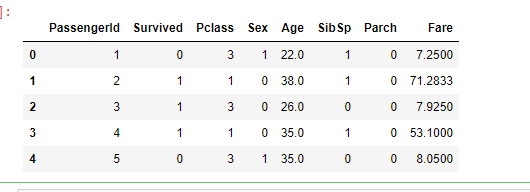
4 2

Name: Embarked, dtype: int32

In [54]:

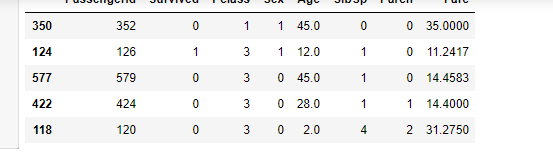
x1=dataset1.drop(columns=['Embarked','Name','Cabin','Ticket'])

x1.head()



X1\_train,X1\_test,y1\_train,y1\_test=train\_test\_split(x1,y1,test\_size=0.3,random\_state=0)

X1\_train.head()



cl=BernoulliNB()

y\_pred=cl.fit(X1\_train,y1\_train).predict(X1\_test)

accuracy\_score(y1\_test,y\_pred,normalize=True)

0.7340823970037453

confusion\_matrix(y1\_test,y\_pred)

array([[ 0, 0, 49],

[ 0, 0, 22],

[ 0, 0, 196]], dtype=int64)