import pandas as pd

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

from sklearn import preprocessing

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

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score,confusion\_matrix

le = preprocessing.LabelEncoder()

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

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

y = dataset['Survived']

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

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

from sklearn.naive\_bayes import \*

clf = BernoulliNB()

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

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



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



**Function Model**

def titanic\_model (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)

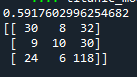
print(accuracy\_score(y\_test,y\_pred,normalize=True))

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

**Pclass**

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

y = dataset['Pclass']



**Sex**

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

y = dataset['Sex']

titanic\_model(X,y)

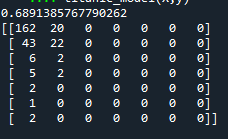


**SibSp**

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

y = dataset['SibSp']

titanic\_model(X,y)

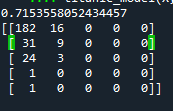
****

**Parch**

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

y = dataset['Parch']

titanic\_model(X,y)

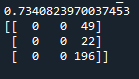
****

**Embarked**

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

y = dataset['Embarked']

titanic\_model(X,y)



**Age**

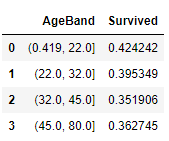
dataset['Age'] = dataset['Age'].astype(int)

**##### Convert float into integers**

dataset['AgeBand'] = pd.qcut(dataset['Age'], 4)

dataset[['AgeBand', 'Survived']].groupby(['AgeBand'], as\_index=False).mean().sort\_values(by='AgeBand', ascending=True)

**##### To create New column “AgeBrand”**

****

dataset = dataset.drop(['AgeBand'],axis=1)

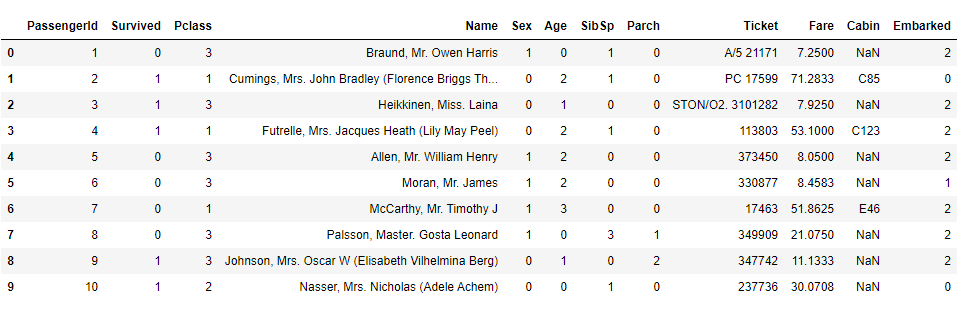
**###### Drop the AgeBand Columns.**

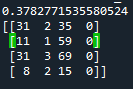
dataset.loc[dataset['Age'] <= 22, 'Age'] = 0

dataset.loc[(dataset['Age'] > 22) & (dataset['Age'] <= 32), 'Age'] = 1

dataset.loc[(dataset['Age'] > 32) & (dataset['Age'] <= 45), 'Age'] = 2

dataset.loc[(dataset['Age'] > 45) & (dataset['Age'] <= 80), 'Age'] = 3





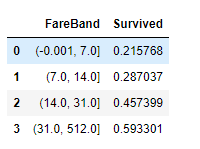
**Fare**

dataset['Fare'] = dataset['Fare'].astype(int)

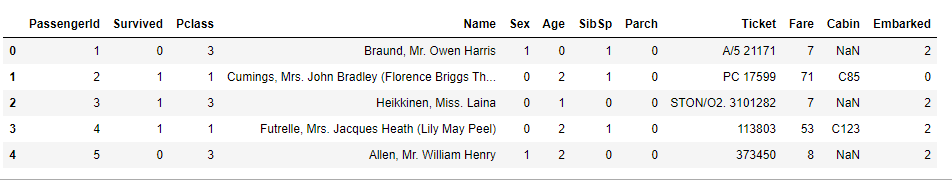
**###### convert float type into integers**

dataset['FareBand'] = pd.qcut(dataset['Fare'], 4)

dataset[['FareBand', 'Survived']].groupby(['FareBand'], as\_index=False).mean().sort\_values(by='FareBand', ascending=True)



dataset = dataset.drop(['FareBand'],axis=1)



dataset.loc[dataset['Fare'] <=7,'Fare'] = 0

dataset.loc[(dataset['Fare'] > 7) & (dataset['Fare'] <=14)] = 1

dataset.loc[(dataset['Fare'] > 14) & (dataset['Fare'] <=31)] = 2

dataset.loc[(dataset['Fare'] > 31) & (dataset['Fare'] <=512)] = 3

def titanic\_model (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)

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

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

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

y = dataset['Fare']

titanic\_model(X,y)

