**Naïve Bayes:**

1. **Model 1**

**DV-> ["Survived"]**

**IDV-> ["Pclass","Sex","Age","SibSp","Parch","Fare"]**

**Code:**

#import 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

#load dataset

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

#dropping unimportant variables

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

#ckeck for null values

print(dataset.isna().sum())

#convert text into numerical

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

#assigning DV to y and IDV to x

y=dataset["Survived"]

X=dataset[["Pclass","Sex","Age","SibSp","Parch","Fare"]]

print(y.count())

#training the model

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

#applying naive bayes algorithm

from sklearn.naive\_bayes import BernoulliNB

clf=BernoulliNB()

#prediction

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

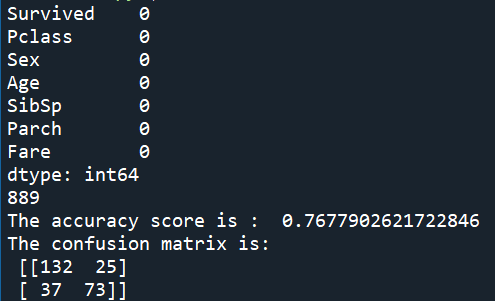
#accuracy score

print("The accuracy score is : ",accuracy\_score(y\_test, y\_pred, normalize=True))

#confusion matrix

print("The confusion matrix is: \n",confusion\_matrix(y\_test, y\_pred))

**Output:**



1. **Model 2**

**DV-> ["Pclass"]**

**IDV-> ["Survived ","Sex","Age","SibSp","Parch","Fare"]**

**Code:**

#import 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

#load dataset

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

#dropping unimportant variables

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

#ckeck for null values

print(dataset.isna().sum())

#convert text into numerical

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

#assigning DV to y and IDV to x

y=dataset["Pclass"]

X=dataset[["Survived","Sex","Age","SibSp","Parch","Fare"]]

print(y.count())

#training the model

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

#applying naive bayes algorithm

from sklearn.naive\_bayes import BernoulliNB

clf=BernoulliNB()

#prediction

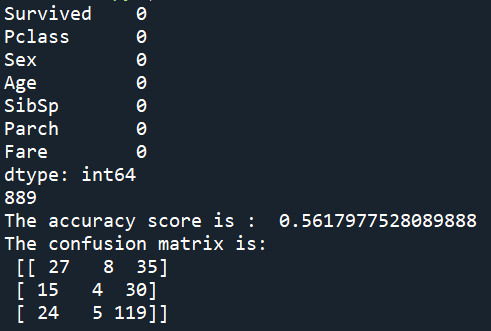
y\_pred=clf.fit(X\_train,y\_train).predict(X\_test)

#accuracy score

print("The accuracy score is : ",accuracy\_score(y\_test, y\_pred, normalize=True))

#confusion matrix

print("The confusion matrix is: \n",confusion\_matrix(y\_test, y\_pred))

**Output:**

1. **Model 3**

**DV-> ["Sex"]**

**IDV-> ["Survived ","Pclass","Age","SibSp","Parch","Fare"]**

**Code:**

#import 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

#load dataset

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

#dropping unimportant variables

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

#ckeck for null values

print(dataset.isna().sum())

#convert text into numerical

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

#assigning DV to y and IDV to x

y=dataset["Sex"]

X=dataset[["Survived","Pclass","Age","SibSp","Parch","Fare"]]

print(y.count())

#training the model

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

#applying naive bayes algorithm

from sklearn.naive\_bayes import BernoulliNB

clf=BernoulliNB()

#prediction

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

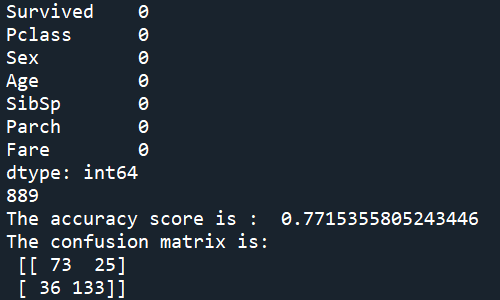
#accuracy score

print("The accuracy score is : ",accuracy\_score(y\_test, y\_pred, normalize=True))

#confusion matrix

print("The confusion matrix is: \n",confusion\_matrix(y\_test, y\_pred))

**Output:**



1. **Model 4**

**DV-> ["SibSp"]**

**IDV-> ["Survived ","Pclass","Sex","Age","Parch","Fare"]**

**Code:**

#import 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

#load dataset

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

#dropping unimportant variables

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

#ckeck for null values

print(dataset.isna().sum())

#convert text into numerical

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

#assigning DV to y and IDV to x

y=dataset["SibSp"]

X=dataset[["Survived","Pclass","Age","Sex","Parch","Fare"]]

print(y.count())

#training the model

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

#applying naive bayes algorithm

from sklearn.naive\_bayes import BernoulliNB

clf=BernoulliNB()

#prediction

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

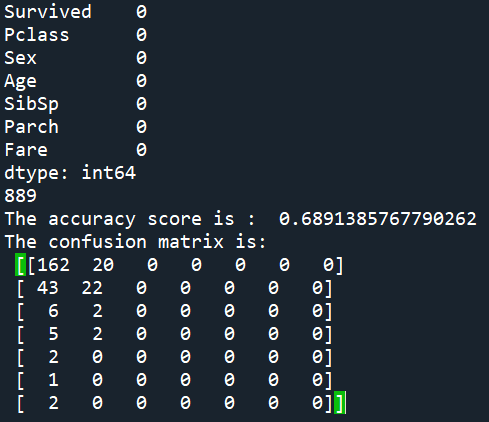
#accuracy score

print("The accuracy score is : ",accuracy\_score(y\_test, y\_pred, normalize=True))

#confusion matrix

print("The confusion matrix is: \n",confusion\_matrix(y\_test, y\_pred))

**Output:**



1. **Model 5**

**DV-> ["Parch"]**

**IDV-> ["Survived ","Pclass","Age","SibSp","Fare","Sex"]**

**Code:**

#import 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

#load dataset

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

#dropping unimportant variables

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

#ckeck for null values

print(dataset.isna().sum())

#convert text into numerical

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

#assigning DV to y and IDV to x

y=dataset["Parch"]

X=dataset[["Survived","Pclass","Age","Sex","Fare","SibSp"]]

print(y.count())

#training the model

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

#applying naive bayes algorithm

from sklearn.naive\_bayes import BernoulliNB

clf=BernoulliNB()

#prediction

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

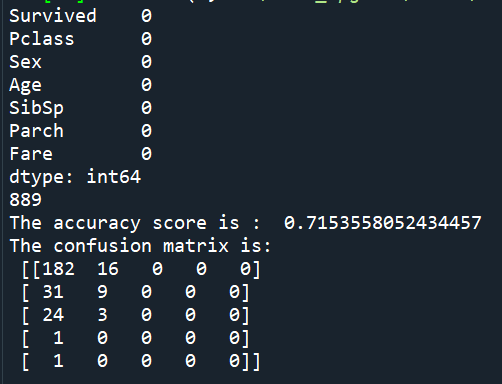
#accuracy score

print("The accuracy score is : ",accuracy\_score(y\_test, y\_pred, normalize=True))

#confusion matrix

print("The confusion matrix is: \n",confusion\_matrix(y\_test, y\_pred))

**Output:**



|  |  |  |
| --- | --- | --- |
| **DV** | **IDV** | **Accuracy Score** |
| ["Survived"] | ["Pclass","Sex","Age","SibSp","Parch","Fare"] | 76.77% |
| ["Pclass"] | ["Survived ","Sex","Age","SibSp","Parch","Fare"] | 56.17% |
| ["Sex"] | ["Survived ","Pclass","Age","SibSp","Parch","Fare"] | 77.15% |
| ["SibSp"] | ["Survived","Pclass","Age","Sex","Parch","Fare"] | 68.91% |
| ["Parch"] | ["Survived","Pclass","Age","Sex","Fare","SibSp"] | 71.53% |