Naive Bayes Classifier - Titanic Survival

July 22, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

[2]:
```
amit = pd.read_csv('titanic.csv')
```

[3]:
```
amit.
 ↪drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis=1,inplace=True
```

[4]:
```
amit.head()
```

[4]:

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

[5]:
```
x = amit.drop('Survived',axis=1)
y = amit['Survived']
```

[6]:
```
dummies = pd.get_dummies(x.Sex)
```

[7]:
```
dummies.head()
```

[7]:

	female	male
0	False	True
1	True	False
2	True	False
3	True	False
4	False	True

[8]:
```
x = pd.concat([x,dummies],axis='columns')
```

[9]:
```
x.head()
```

[9]:

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	False	True
1	1	female	38.0	71.2833	True	False
2	3	female	26.0	7.9250	True	False
3	1	female	35.0	53.1000	True	False
4	3	male	35.0	8.0500	False	True

[10]:
```
x.drop('Sex',axis=1,inplace=True)
```

[11]:
```
x.head()
```

[11]:

	Pclass	Age	Fare	female	male
0	3	22.0	7.2500	False	True
1	1	38.0	71.2833	True	False
2	3	26.0	7.9250	True	False
3	1	35.0	53.1000	True	False
4	3	35.0	8.0500	False	True

[12]:
```
x.columns[x.isna().any()]
```

[12]:
```
Index(['Age'], dtype='object')
```

[13]:
```
x.Age = x.Age.fillna(x.Age.mean())
```

[14]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

[15]:
```
from sklearn.naive_bayes import GaussianNB
```

[16]:
```
model = GaussianNB()
```

[17]:
```
model.fit(X_train,y_train)
```

[17]:
```
GaussianNB()
```

[18]:
```
model.score(X_test,y_test)
```

[18]:
```
0.7877094972067039
```

[19]:
```
X_test[:10]
```

[19]:

	Pclass	Age	Fare	female	male
732	2	29.699118	0.0000	False	True
451	3	29.699118	19.9667	False	True
485	3	29.699118	25.4667	True	False
375	1	29.699118	82.1708	True	False
609	1	40.000000	153.4625	True	False

```
564       3  29.699118    8.0500    True  False
286       3  30.000000    9.5000   False   True
33        2  66.000000   10.5000   False   True
443       2  28.000000   13.0000    True  False
695       2  52.000000   13.5000   False   True
```

[20]:
```
y_test[:10]
```

[20]:
```
732    0
451    0
485    0
375    1
609    1
564    0
286    1
33     0
443    1
695    0
Name: Survived, dtype: int64
```

[21]:
```
model.predict(X_test[:10])
# Here 0 means person Survived and 1 means not Survived
```

[21]:
```
array([0, 0, 1, 1, 1, 1, 0, 0, 1, 0])
```

[22]:
```
model.predict_proba(X_test[:10])
```

[22]:
```
array([[9.79241074e-01, 2.07589259e-02],
       [9.91128889e-01, 8.87111118e-03],
       [5.35157328e-02, 9.46484267e-01],
       [1.45534462e-03, 9.98544655e-01],
       [6.28878709e-06, 9.99993711e-01],
       [5.45917111e-02, 9.45408289e-01],
       [9.91122883e-01, 8.87711734e-03],
       [9.53676226e-01, 4.63237736e-02],
       [2.56443527e-02, 9.74355647e-01],
       [9.75079628e-01, 2.49203725e-02]])
```

[ ]: