In [1]:

```
import pandas as pd
df = pd.read_csv("titanic.csv")
df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [2]:

```
df.drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis='columns',inplace=True)
df.head()
```

Out[2]:

	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

In [3]:

```
inputs = df.drop('Survived',axis='columns')
target = df.Survived
inputs.head()
```

Out[3]:

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

In [4]:

```
# sex Male=1 Female=2
dummies = pd.get_dummies(inputs.Sex)
dummies
```

Out[4]:

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...	...	...
886	0	1
887	1	0
888	1	0
889	0	1
890	0	1

891 rows × 2 columns

In [5]:

```
inputs = pd.concat([inputs,dummies],axis='columns')
inputs.head(3)
```

Out[5]:

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	0	1
1	1	female	38.0	71.2833	1	0
2	3	female	26.0	7.9250	1	0

In [6]:

```
inputs.drop(['Sex','male'],axis='columns',inplace=True)
inputs.head(3)
```

Out[6]:

	Pclass	Age	Fare	female
0	3	22.0	7.2500	0
1	1	38.0	71.2833	1
2	3	26.0	7.9250	1

inputs.columns[inputs.isna().any()]

inputs.shape

In [7]:

```
inputs.Age[:10]
inputs.head(6)
```

Out[7]:

	Pclass	Age	Fare	female
0	3	22.0	7.2500	0
1	1	38.0	71.2833	1
2	3	26.0	7.9250	1
3	1	35.0	53.1000	1
4	3	35.0	8.0500	0
5	3	NaN	8.4583	0

In [8]:

```
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
inputs.head(6)
```

Out[8]:

	Pclass	Age	Fare	female
0	3	22.000000	7.2500	0
1	1	38.000000	71.2833	1
2	3	26.000000	7.9250	1
3	1	35.000000	53.1000	1
4	3	35.000000	8.0500	0
5	3	29.699118	8.4583	0

In [9]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.3)
```

In [10]:

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

In [11]:

```
model.fit(X_train,y_train)
```

Out[11]:

```
▼ GaussianNB
GaussianNB()
```

In [ ]:

In [12]:

```
X_test[0:10]
```

Out[12]:

	Pclass	Age	Fare	female
356	1	22.000000	55.0000	1
590	3	35.000000	7.1250	0
700	1	18.000000	227.5250	1
151	1	22.000000	66.6000	1
782	1	29.000000	30.0000	0
724	1	27.000000	53.1000	0
520	1	30.000000	93.5000	1
447	1	34.000000	26.5500	0
432	2	42.000000	26.0000	1
229	3	29.699118	25.4667	1

In [13]:

```
y_test[0:10]
```

Out[13]:

```
356    1
590    0
700    1
151    1
782    0
724    1
520    1
447    1
432    1
229    0
Name: Survived, dtype: int64
```

In [14]:

```
model.predict(X_test[0:10])
```

Out[14]:

```
array([1, 0, 1, 1, 0, 0, 1, 0, 1, 1], dtype=int64)
```

In [ ]:

cross checking

In [15]:

```
y_predicted = model.predict(X_test)
```

In [16]:

```
print('Accuracy on the training subset: {:.3f}'.format(model.score(X_train, y_train)))
print('Accuracy on the test subset: {:.3f}'.format(model.score(X_test, y_test)))
```

```
Accuracy on the training subset: 0.783
Accuracy on the test subset: 0.761
```

In [18]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_predicted)
```

```
Out[18]: array([[127,  30],
                [ 34,  77]], dtype=int64)
```

```
In [ ]:
```