```
import pandas as pd
import numpy as np
df=pd.read_csv('titanic.csv')
df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
# Drop the unnecessary columns
df.drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embark
ed'], axis = 'columns', inplace = True)
df.head()
```

```
   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
```

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

dummies = pd.get_dummies(inputs.Sex)
dummies.head()
```

```
   female  male
0       0     1
1       1     0
2       1     0
3       1     0
4       0     1
```

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

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

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

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

```
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
inputs.head(10)
```

```
   Pclass        Age     Fare  female  male  female  male  female
male
0       3  22.000000   7.2500       0     1       0     1       0
1
1       1  38.000000  71.2833       1     0       1     0       1
0
2       3  26.000000   7.9250       1     0       1     0       1
0
3       1  35.000000  53.1000       1     0       1     0       1
0
4       3  35.000000   8.0500       0     1       0     1       0
1
5       3  29.699118   8.4583       0     1       0     1       0
1
6       1  54.000000  51.8625       0     1       0     1       0
1
7       3   2.000000  21.0750       0     1       0     1       0
1
8       3  27.000000  11.1333       1     0       1     0       1
0
9       2  14.000000  30.0708       1     0       1     0       1
0
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs, target,
test_size=0.2, random_state=1)
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

clf = GaussianNB()
clf.fit(X_train, y_train)

GaussianNB(priors=None, var_smoothing=1e-09)

pred = clf.predict(X_test)
pred

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

# Create a Confusion Matrix
matrix = pd.DataFrame(
        confusion_matrix(y_test, pred),
        columns=['Predicted 0', 'Predicted 1'],
        index=['Actual 0', 'Actual 1'])
matrix

          Predicted 0  Predicted 1
Actual 0           90           16
Actual 1           23           50
```