```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('titanic.csv')
```

```
df.head()
```

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	Braund, Mr. Owen Harris	3	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	Heikkinen, Miss. Laina	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	1	0	113803	53.1000	C123	
4	5	Allen, Mr. William Henry	3	male	35.0	0	0	373450	8.0500	NaN	

```
df.shape
```

```
(891, 12)
```

```
df.describe()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Survived
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	2.308642	29.699118	0.523008	0.381594	32.204208	0.383838
std	257.353842	0.836071	14.526497	1.102743	0.806057	49.693429	0.486592
min	1.000000	1.000000	0.420000	0.000000	0.000000	0.000000	0.000000
25%	223.500000	2.000000	20.125000	0.000000	0.000000	7.910400	0.000000
50%	446.000000	3.000000	28.000000	0.000000	0.000000	14.454200	0.000000
75%	668.500000	3.000000	38.000000	1.000000	0.000000	31.000000	1.000000
max	891.000000	3.000000	80.000000	8.000000	6.000000	512.329200	1.000000

```
list(df.columns)
```

```
['PassengerId',
 'Name',
 'Pclass',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked',
 'Survived']
```

```
df=df.drop(['SibSp','Parch','Ticket', 'Cabin','Fare','Embarked', 'PassengerId','Name'], axis=
```

```
label_encoder = LabelEncoder()
df['Sex'] = label_encoder.fit_transform(df['Sex'])
```

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

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

```
df.Age = df.Age.fillna(df.Age.mean())
df.columns[df.isna().any()]
```

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

```
X=df.iloc[:, [1,2,3]]
Y=df.iloc[:, 0]
```

```
Y.head()
```

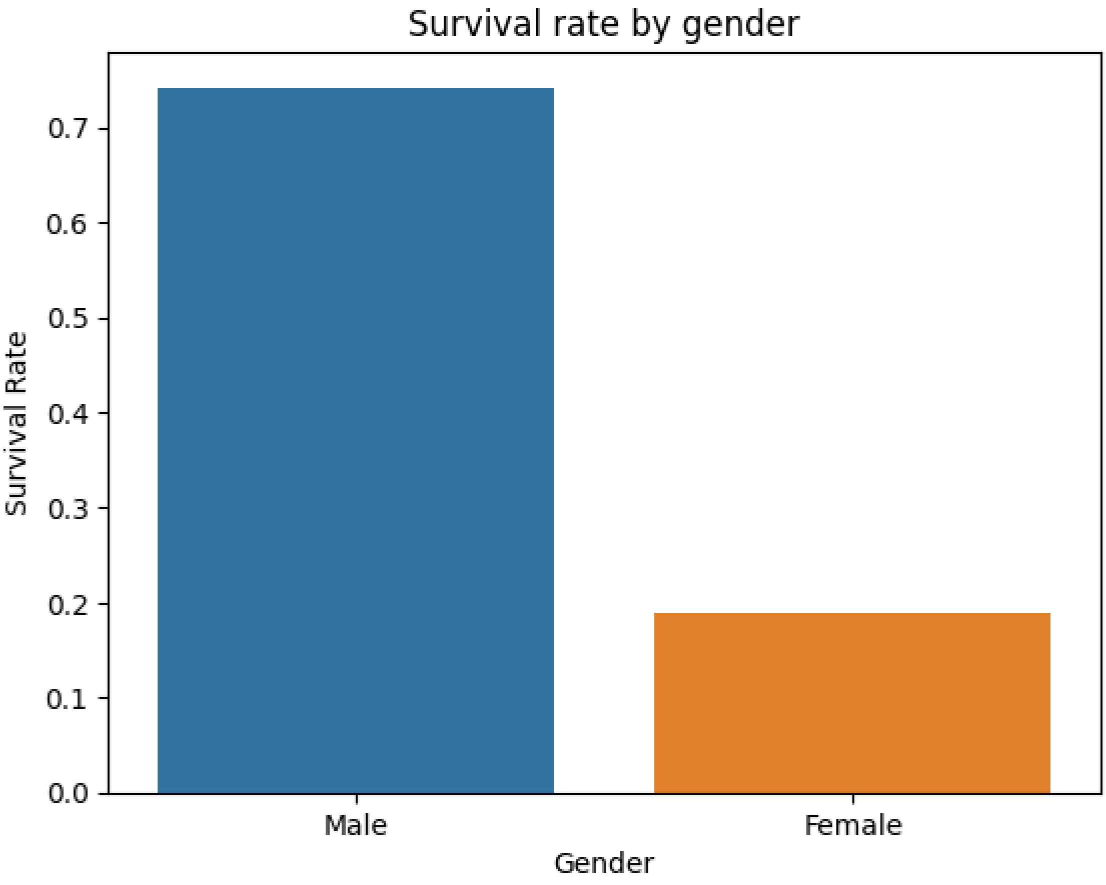
```
0    3
1    1
2    3
3    1
4    3
Name: Pclass, dtype: int64
```

```
sns.barplot(x='Sex', y='Survived', data=df, ci=None)
plt.xlabel('Gender')
plt.ylabel('Survival Rate')
plt.title('Survival rate by gender')
plt.xticks([0, 1], ['Male', 'Female'])
plt.show()
```

```
<ipython-input-15-540d2800e8d6>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='Sex', y='Survived', data=df, ci=None)
```



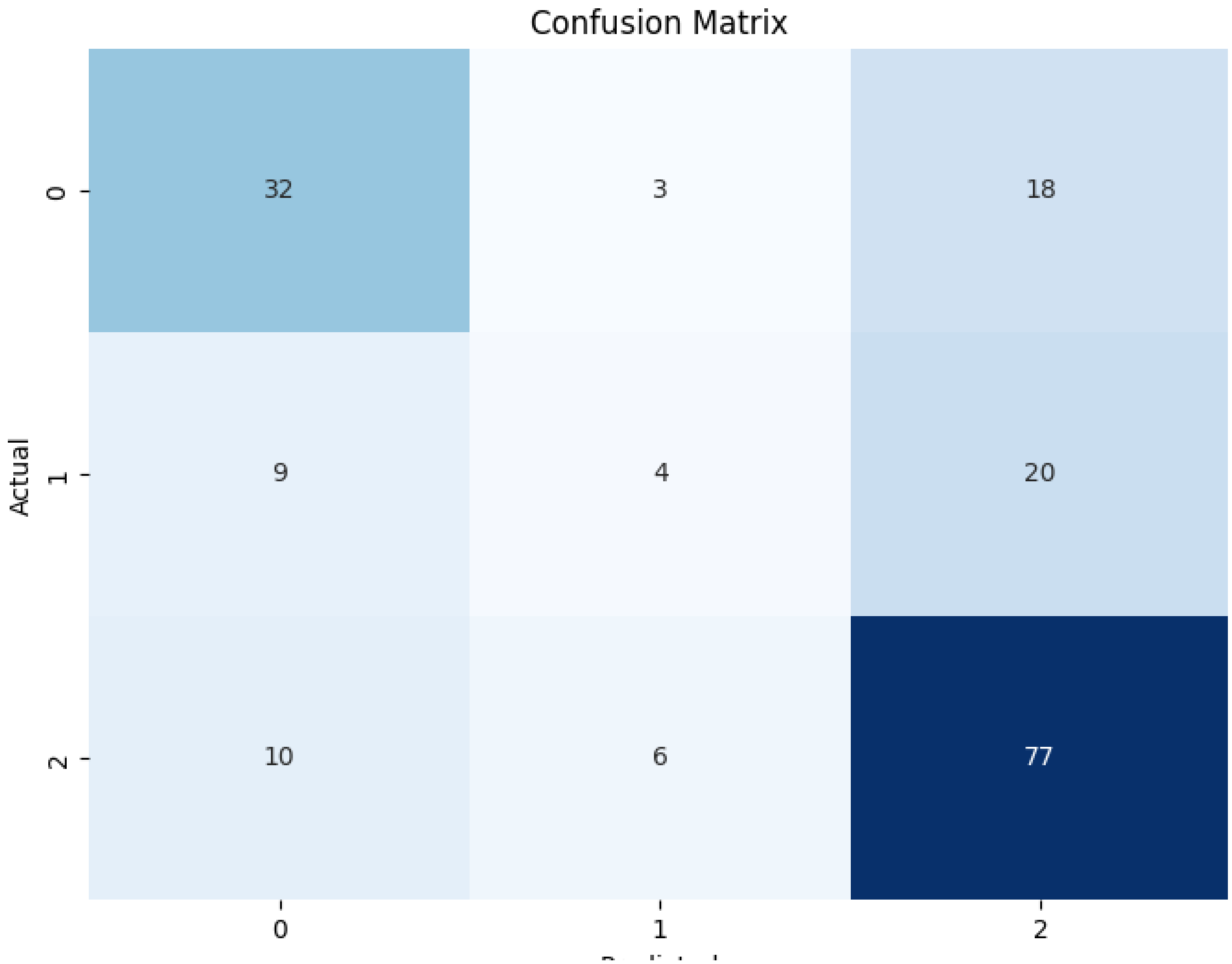
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=42)
```

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

```
▾ GaussianNB
GaussianNB()
```

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

```
from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.6312849162011173
```