In [7]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:
```
titanic_data = pd.read_csv("tested.csv")
```

In [9]:
```
titanic_data
```

Out[9]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cab |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 892 | 0 | 3 | Kelly, Mr. James | male | 34.5 | 0 | 0 | 330911 | 7.8292 | Na |
| **1** | 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47.0 | 1 | 0 | 363272 | 7.0000 | Na |
| **2** | 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62.0 | 0 | 0 | 240276 | 9.6875 | Na |
| **3** | 895 | 0 | 3 | Wirz, Mr. Albert | male | 27.0 | 0 | 0 | 315154 | 8.6625 | Na |
| **4** | 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22.0 | 1 | 1 | 3101298 | 12.2875 | Na |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **413** | 1305 | 0 | 3 | Spector, Mr. Woolf | male | NaN | 0 | 0 | A.5. 3236 | 8.0500 | Na |
| **414** | 1306 | 1 | 1 | Oliva y Ocana, Dona. Fermina | female | 39.0 | 0 | 0 | PC 17758 | 108.9000 | C1 |
| **415** | 1307 | 0 | 3 | Saether, Mr. Simon Sivertsen | male | 38.5 | 0 | 0 | SOTON/O.Q. 3101262 | 7.2500 | Na |
| **416** | 1308 | 0 | 3 | Ware, Mr. Frederick | male | NaN | 0 | 0 | 359309 | 8.0500 | Na |
| **417** | 1309 | 0 | 3 | Peter, Master. Michael J | male | NaN | 1 | 1 | 2668 | 22.3583 | Na |

418 rows × 12 columns

In [10]:
```
titanic_data.head()
```

Out[10]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Em |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 892 | 0 | 3 | Kelly, Mr. James | male | 34.5 | 0 | 0 | 330911 | 7.8292 | NaN | |
| **1** | 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47.0 | 1 | 0 | 363272 | 7.0000 | NaN | |
| **2** | 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62.0 | 0 | 0 | 240276 | 9.6875 | NaN | |
| **3** | 895 | 0 | 3 | Wirz, Mr. Albert | male | 27.0 | 0 | 0 | 315154 | 8.6625 | NaN | |
| **4** | 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22.0 | 1 | 1 | 3101298 | 12.2875 | NaN | |

In [12]:
```
titanic_data.shape
```

Out[12]:
```
(418, 12)
```

In [13]:
```
titanic_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [14]:
```
titanic_data.isnull().sum()
```

Out[14]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [15]:
```
titanic_data.drop(columns='Cabin',axis=1)
```

Out[15]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Em |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 892 | 0 | 3 | Kelly, Mr. James | male | 34.5 | 0 | 0 | 330911 | 7.8292 | |
| **1** | 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47.0 | 1 | 0 | 363272 | 7.0000 | |
| **2** | 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62.0 | 0 | 0 | 240276 | 9.6875 | |
| **3** | 895 | 0 | 3 | Wirz, Mr. Albert | male | 27.0 | 0 | 0 | 315154 | 8.6625 | |
| **4** | 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22.0 | 1 | 1 | 3101298 | 12.2875 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **413** | 1305 | 0 | 3 | Spector, Mr. Woolf | male | NaN | 0 | 0 | A.5. 3236 | 8.0500 | |
| **414** | 1306 | 1 | 1 | Oliva y Ocana, Dona. Fermina | female | 39.0 | 0 | 0 | PC 17758 | 108.9000 | |
| **415** | 1307 | 0 | 3 | Saether, Mr. Simon Sivertsen | male | 38.5 | 0 | 0 | SOTON/O.Q. 3101262 | 7.2500 | |
| **416** | 1308 | 0 | 3 | Ware, Mr. Frederick | male | NaN | 0 | 0 | 359309 | 8.0500 | |
| **417** | 1309 | 0 | 3 | Peter, Master. Michael J | male | NaN | 1 | 1 | 2668 | 22.3583 | |

418 rows × 11 columns

In [16]:
```
titanic_data['Age'].fillna(titanic_data['Age'].mean(),inplace=True)
```

In [17]:
```
print(titanic_data['Fare'].mode())
```
```
0    7.75
Name: Fare, dtype: float64
```

In [18]:
```
print(titanic_data['Fare'].mode()[0])
```
```
7.75
```

In [23]:
```
titanic_data['Fare'].fillna(titanic_data['Fare'].mode()[0],inplace=True)
```

In [24]:
```
titanic_data.isnull().sum()
```

Out[24]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
dtype: int64
```

In [43]:
```
titanic_data=titanic_data.drop(columns='Cabin',axis=1)
```

In [44]:
```
titanic_data.isnull().sum()
```

Out[44]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [45]:
```
titanic_data.describe()
```

Out[45]:

| | PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Fa |
|---|---|---|---|---|---|---|---|---|
| **count** | 418.000000 | 418.000000 | 418.000000 | 418.000000 | 418.000000 | 418.000000 | 418.000000 | 418.00000 |
| **mean** | 1100.500000 | 0.363636 | 2.265550 | 0.363636 | 30.272590 | 0.447368 | 0.392344 | 35.5604 |
| **std** | 120.810458 | 0.481622 | 0.841838 | 0.481622 | 12.634534 | 0.896760 | 0.981429 | 55.8571 |
| **min** | 892.000000 | 0.000000 | 1.000000 | 0.000000 | 0.170000 | 0.000000 | 0.000000 | 0.0000 |
| **25%** | 996.250000 | 0.000000 | 1.000000 | 0.000000 | 23.000000 | 0.000000 | 0.000000 | 7.8958 |
| **50%** | 1100.500000 | 0.000000 | 3.000000 | 0.000000 | 30.272590 | 0.000000 | 0.000000 | 14.4542 |
| **75%** | 1204.750000 | 1.000000 | 3.000000 | 1.000000 | 35.750000 | 1.000000 | 0.000000 | 31.4718 |
| **max** | 1309.000000 | 1.000000 | 3.000000 | 1.000000 | 76.000000 | 8.000000 | 9.000000 | 512.3292 |

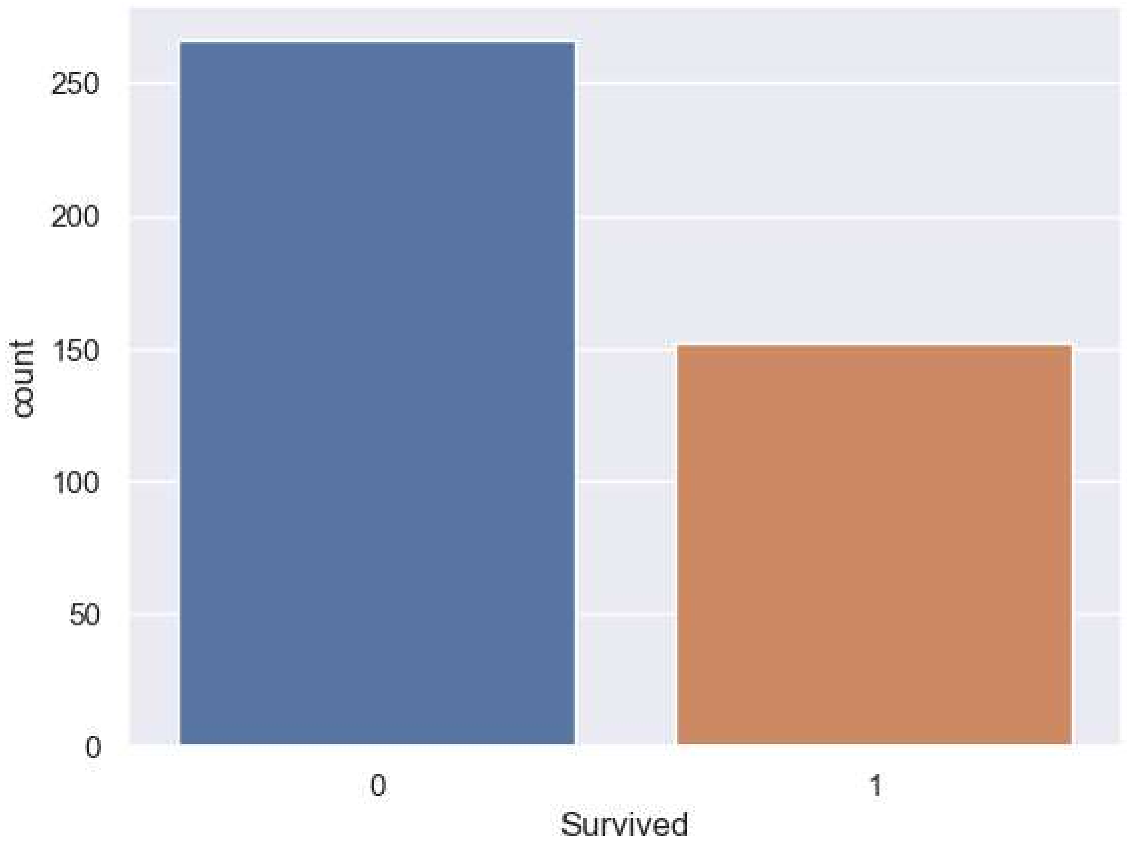
In [46]:
```
titanic_data["Survived"].value_counts()
```

```
Out[46]: 0    266
         1    152
         Name: Survived, dtype: int64
```

```
In [47]: sns.set()
         sns.countplot("Survived",data=titanic_data)
```

```
C:\Users\SAHITHI\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

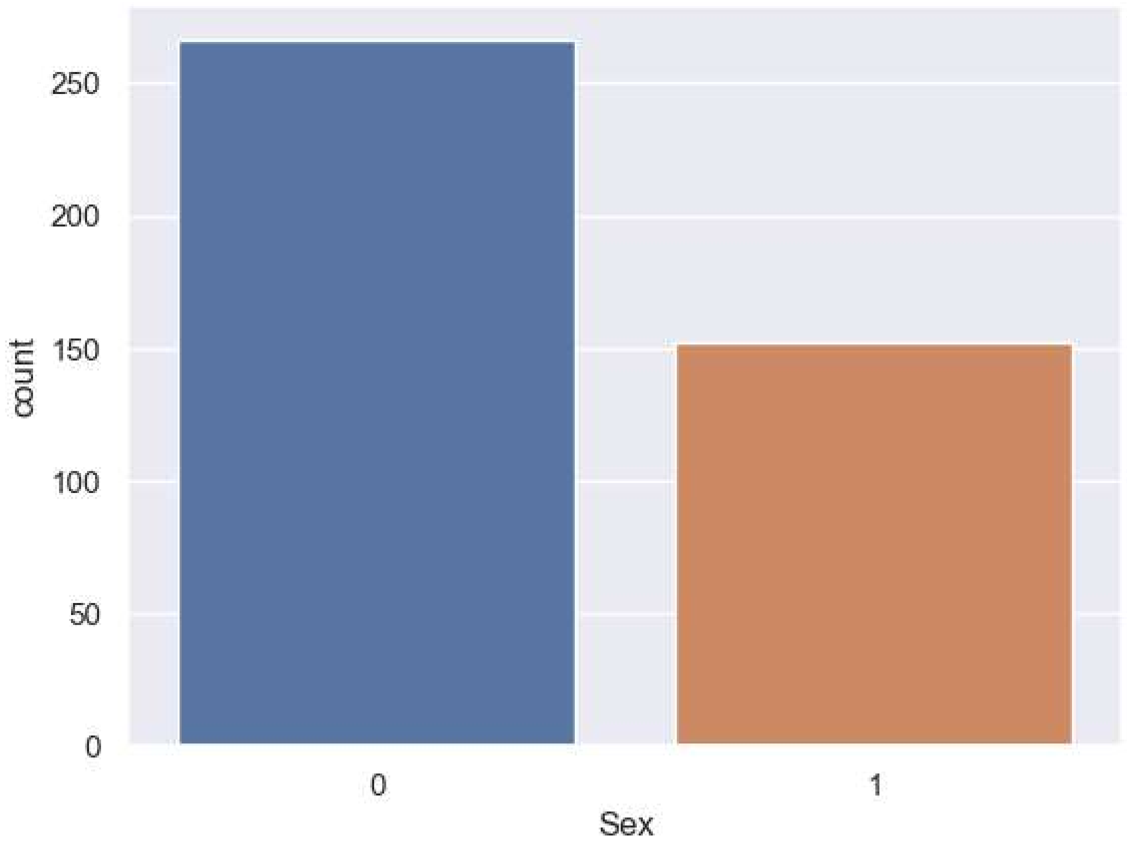
```
Out[47]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



```
In [48]: sns.set()
         sns.countplot("Sex",data=titanic_data)
```

```
C:\Users\SAHITHI\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

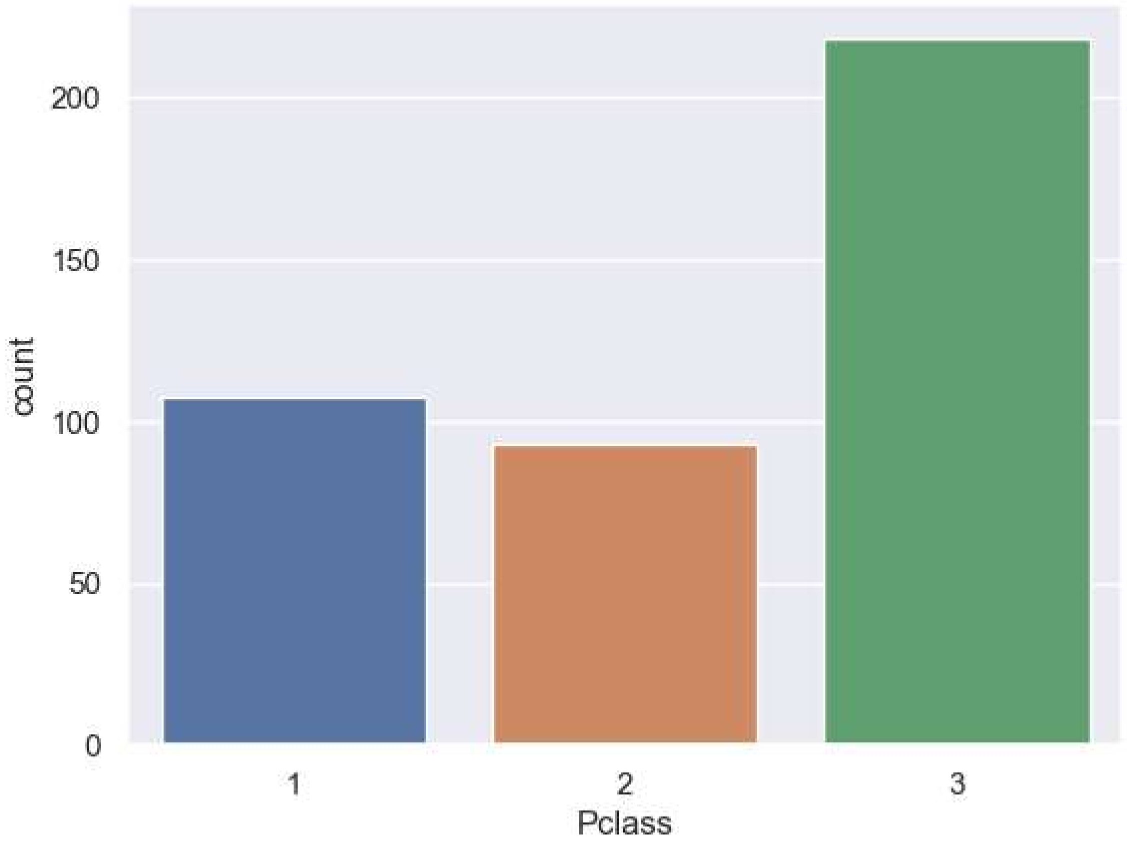
```
Out[48]: <AxesSubplot:xlabel='Sex', ylabel='count'>
```



In [49]:
```
sns.countplot("Pclass",data=titanic_data)
```

```
C:\Users\SAHITHI\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[49]:
```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```



In [50]:
```
titanic_data.replace({"Sex":{"male":0,"female":1}, "Embarked":{"S":0,"C":1,"Q":2}},inp
```

In [51]:
```
titanic_data.head()
```

Out[51]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 892 | 0 | 3 | Kelly, Mr. James | 0 | 34.5 | 0 | 0 | 330911 | 7.8292 | 2 |
| **1** | 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | 1 | 47.0 | 1 | 0 | 363272 | 7.0000 | 0 |
| **2** | 894 | 0 | 2 | Myles, Mr. Thomas Francis | 0 | 62.0 | 0 | 0 | 240276 | 9.6875 | 2 |
| **3** | 895 | 0 | 3 | Wirz, Mr. Albert | 0 | 27.0 | 0 | 0 | 315154 | 8.6625 | 0 |
| **4** | 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | 1 | 22.0 | 1 | 1 | 3101298 | 12.2875 | 0 |

In [52]:
```
X = titanic_data.drop(columns=["Name","Ticket","PassengerId","Survived"],axis=1)
y = titanic_data["Survived"]
```

In [53]:
```
X
```

Out[53]:

| | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
|---|---|---|---|---|---|---|---|
| **0** | 3 | 0 | 34.50000 | 0 | 0 | 7.8292 | 2 |
| **1** | 3 | 1 | 47.00000 | 1 | 0 | 7.0000 | 0 |
| **2** | 2 | 0 | 62.00000 | 0 | 0 | 9.6875 | 2 |
| **3** | 3 | 0 | 27.00000 | 0 | 0 | 8.6625 | 0 |
| **4** | 3 | 1 | 22.00000 | 1 | 1 | 12.2875 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **413** | 3 | 0 | 30.27259 | 0 | 0 | 8.0500 | 0 |
| **414** | 1 | 1 | 39.00000 | 0 | 0 | 108.9000 | 1 |
| **415** | 3 | 0 | 38.50000 | 0 | 0 | 7.2500 | 0 |
| **416** | 3 | 0 | 30.27259 | 0 | 0 | 8.0500 | 0 |
| **417** | 3 | 0 | 30.27259 | 1 | 1 | 22.3583 | 1 |

418 rows × 7 columns

In [54]:
```
y
```

Out[54]:
```
0      0
1      1
2      0
3      0
4      1
      ..
413    0
414    1
415    0
416    0
417    0
Name: Survived, Length: 418, dtype: int64
```

In [55]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [56]:
```
X_train,X_test, y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [57]:
```
print(X.shape,X_train.shape,X_test.shape)
```

```
(418, 7) (334, 7) (84, 7)
```

In [59]:
```
model=LogisticRegression()
```

In [60]:
```
model.fit(X_train,y_train)
```

```
C:\Users\SAHITHI\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: C
onvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[60]:
```
LogisticRegression()
```

In [61]:
```python
X_train_prediction=model.predict(X_train)
```

In [62]:
```python
print(X_train_prediction)
```

```
[1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0
 1 1 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 0 1
 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0
 1 1 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 0 0
 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0 1 0 1 1
 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 0 0
 1 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1
 0 1 1 1 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0
 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0 1 1 1
 1]
```

In [63]:
```python
training_data_accuracy=accuracy_score(y_train,X_train_prediction)
print("accuracy score:",training_data_accuracy)
```

```
accuracy score: 1.0
```

In [64]:
```python
X_test_predict=model.predict(X_test)
```

In [66]:
```python
print(X_test_predict)
```

```
[0 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1
 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0
 0 1 1 0 1 0 0 0 0 0]
```

In [68]:
```python
test_data_accuracy=accuracy_score(y_test,X_test_predict)
print("accuracy score:",test_data_accuracy)
```

```
accuracy score: 1.0
```

In [ ]: