In [ ]:
```
#TITANIC SURVIVAL PREDICTION TASK (1)
```

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

In [2]:
```
# Load data
df=pd.read_csv("Titanic-Dataset.csv")
```

In [3]:
```
df
```

Out[3]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **886** | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27.0 | 0 | 0 | 211536 | 13.0 |
| **887** | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 30.0 |
| **888** | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN | 1 | 2 | W./C. 6607 | 23.4 |
| **889** | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.0 |
| **890** | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32.0 | 0 | 0 | 370376 | 7.7 |

891 rows × 12 columns

In [4]:
```
df.head()
```

Out[4]:

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

In [5]:
```
df.shape
```

Out[5]:
```
(891, 12)
```

In [6]:
```
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 [7]:
```
df.describe()
```

Out[7]:

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

In [8]:
```
df.duplicated().sum()
```

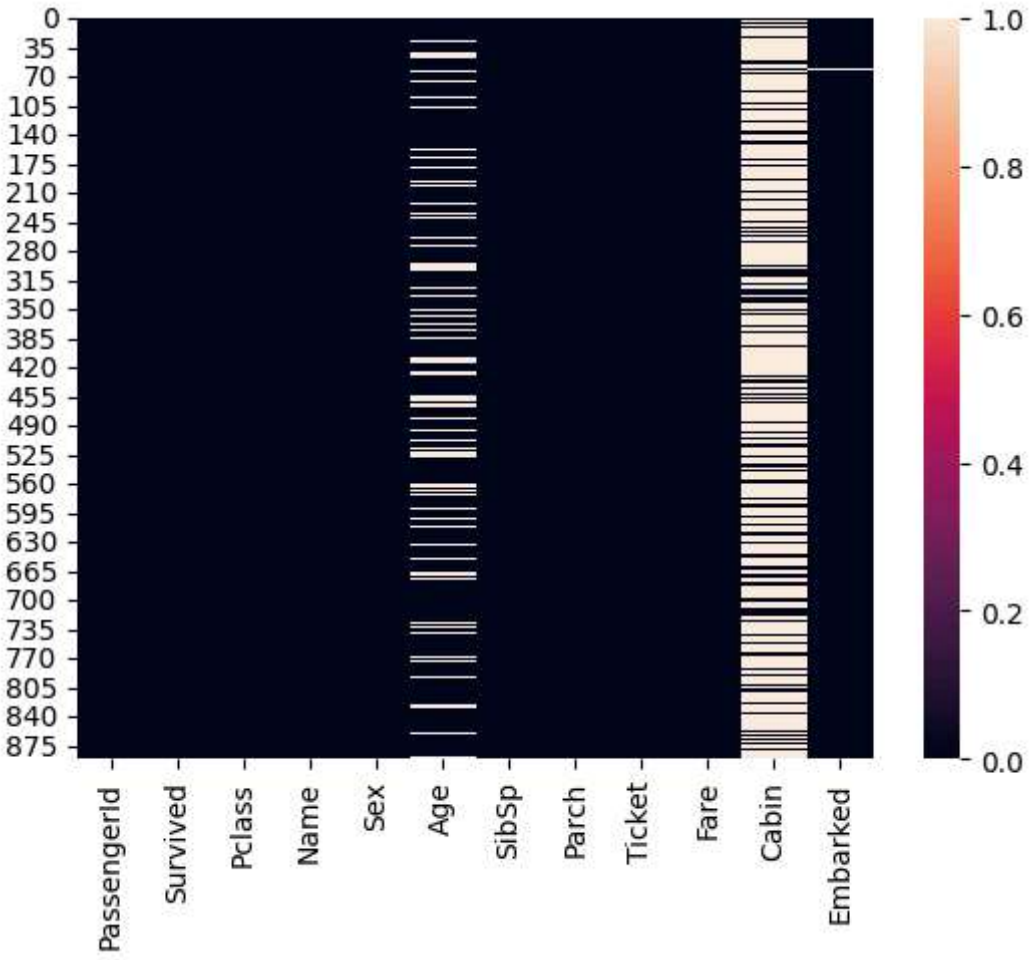
Out[8]:
```
0
```

In [9]:
```
df['Survived'].value_counts()
```

Out[9]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [12]:
```
sns.heatmap(df.isnull())
```

Out[12]:
```
<Axes: >
```




In [13]:
```
df['Sex']
```

Out[13]:
```
0        male
1      female
2      female
3      female
4        male
       ...
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

In [15]:
```
#Data Cleaning
df.drop(['PassengerId','Name','Ticket','Cabin','Embarked','Fare'],axis =1 , inplace
```

In [16]:
```
df.Age=df.Age.fillna(df.Age.mean())
```

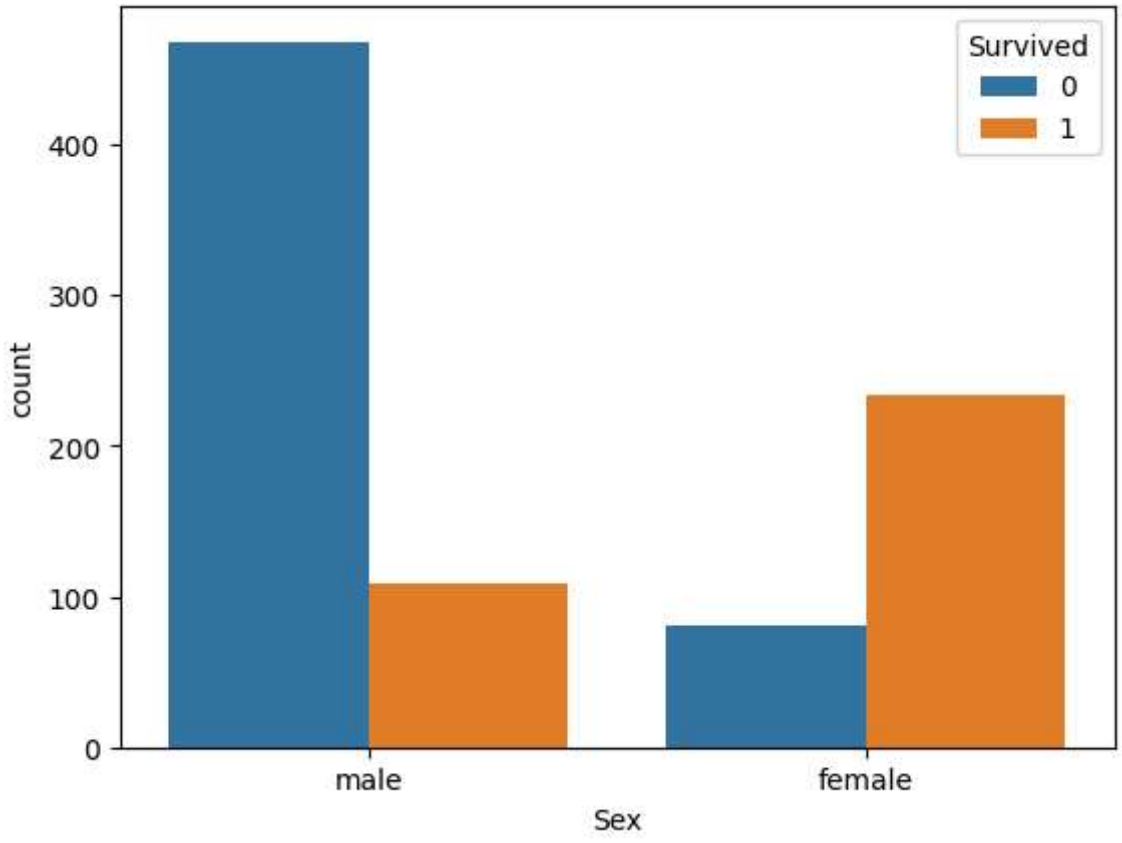
In [17]:
```
df
```

Out[17]:

| | Survived | Pclass | Sex | Age | SibSp | Parch |
|---|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 22.000000 | 1 | 0 |
| **1** | 1 | 1 | female | 38.000000 | 1 | 0 |
| **2** | 1 | 3 | female | 26.000000 | 0 | 0 |
| **3** | 1 | 1 | female | 35.000000 | 1 | 0 |
| **4** | 0 | 3 | male | 35.000000 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **886** | 0 | 2 | male | 27.000000 | 0 | 0 |
| **887** | 1 | 1 | female | 19.000000 | 0 | 0 |
| **888** | 0 | 3 | female | 29.699118 | 1 | 2 |
| **889** | 1 | 1 | male | 26.000000 | 0 | 0 |
| **890** | 0 | 3 | male | 32.000000 | 0 | 0 |

891 rows × 6 columns

In [21]:
```
#Data Aanlysis
sns.countplot(x='Sex', hue='Survived', data=df)
plt.show()
```

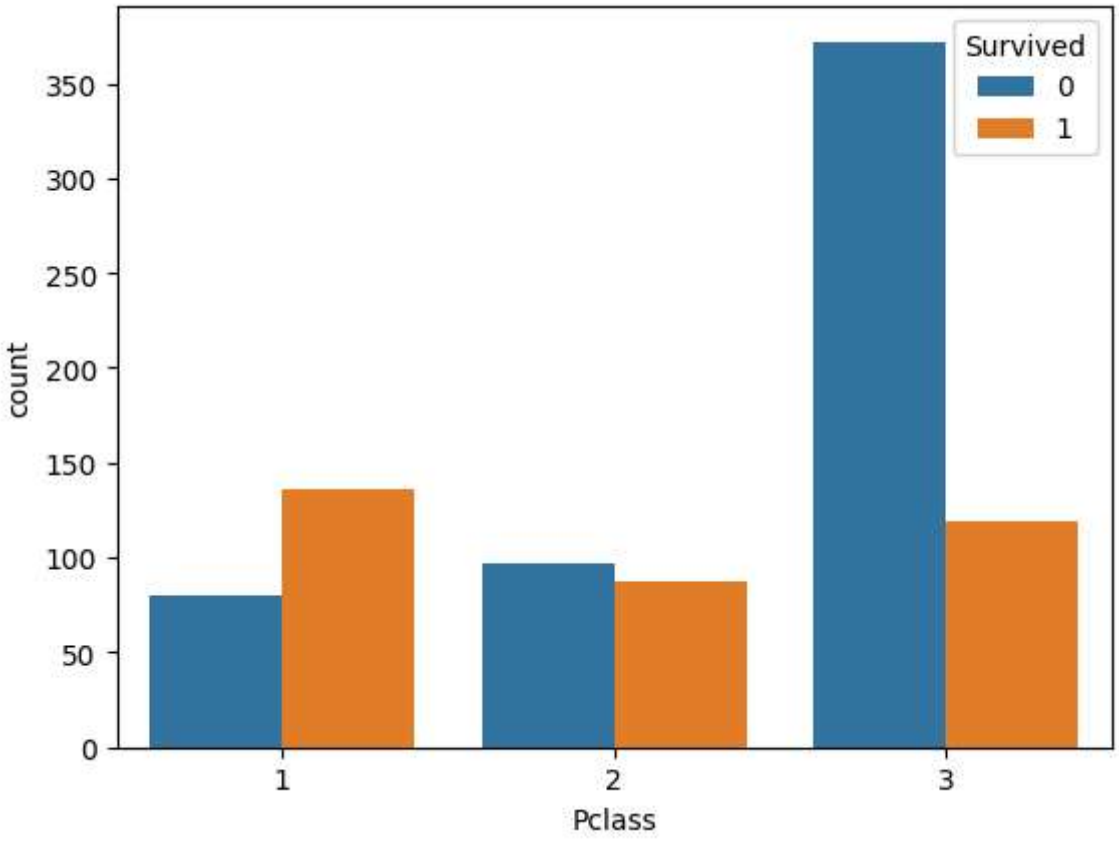


In [19]:
```
df.groupby('Sex')[['Survived']].mean()
```

Out[19]:

| Sex | Survived |
|---|---|
| female | 0.742038 |
| male | 0.188908 |

In [20]:
```
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.show()
```




In [22]:
```
df['Sex'].unique()
```

Out[22]:
```
array(['male', 'female'], dtype=object)
```

In [23]:
```
age=df.Age.groupby(df.Survived).value_counts()
age
```

Out[23]:
```
Survived  Age
0         29.699118    125
          21.000000     19
          28.000000     18
          18.000000     17
          25.000000     17
                        ...
1         43.000000      1
          47.000000      1
          53.000000      1
          55.000000      1
          80.000000      1
Name: count, Length: 144, dtype: int64
```

In [24]:
```
df[df['Survived']==1 ] ['Age'].max()
```

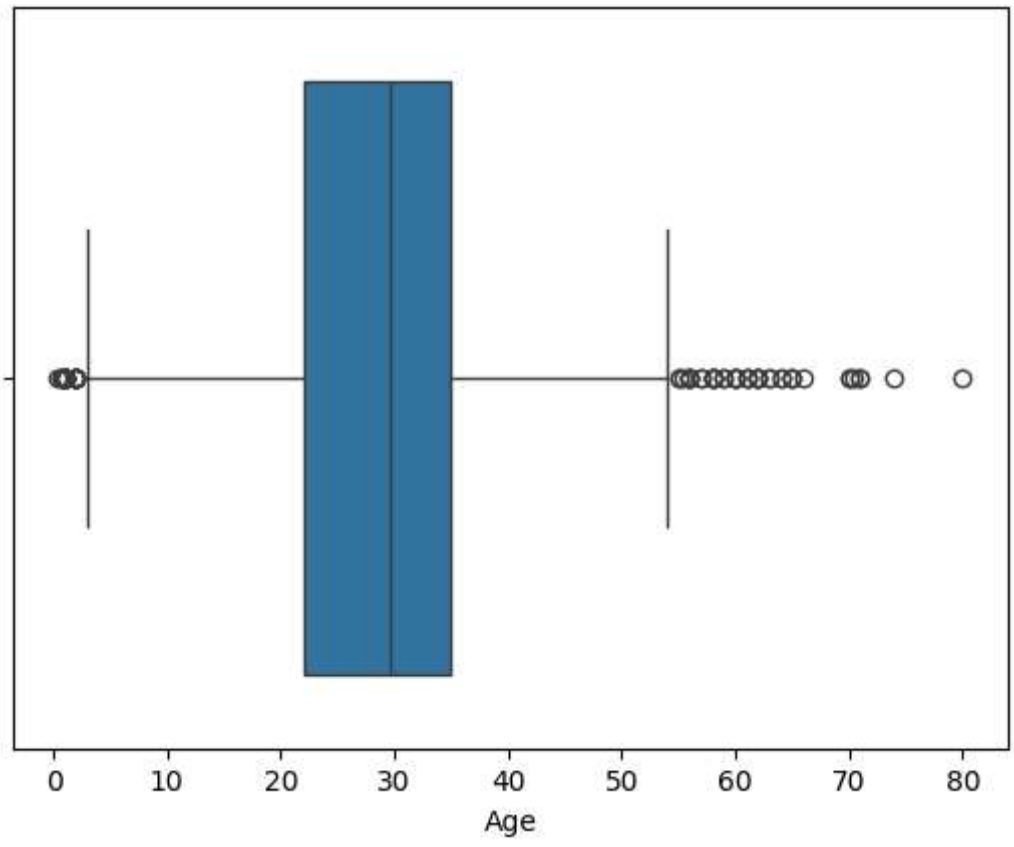
Out[24]:
```
80.0
```

In [25]:
```
df[df['Survived']==0 ] ['Age'].mode()
```

Out[25]:
```
0    29.699118
Name: Age, dtype: float64
```

In [26]:
```
sns.boxplot(x='Age',data=df)
```

Out[26]:
```
<Axes: xlabel='Age'>
```




In [27]:
```
df=df.drop(['Age'],axis=1)
```

In [40]:
```
df
```

Out[40]:

| | Survived | Pclass | Sex | SibSp | Parch |
|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 1 | 0 |
| **1** | 1 | 1 | female | 1 | 0 |
| **2** | 1 | 3 | female | 0 | 0 |
| **3** | 1 | 1 | female | 1 | 0 |
| **4** | 0 | 3 | male | 0 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **886** | 0 | 2 | male | 0 | 0 |
| **887** | 1 | 1 | female | 0 | 0 |
| **888** | 0 | 3 | female | 1 | 2 |
| **889** | 1 | 1 | male | 0 | 0 |
| **890** | 0 | 3 | male | 0 | 0 |

891 rows × 5 columns

In [41]:
```
#Split features & target
x=df.drop('Survived',axis=1)
y=df.Survived
```

In [42]:
```
x.shape,y.shape
```

Out[42]:
```
((891, 4), (891,))
```

In [52]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [56]:
```
#Split train & test data
X = df.drop('Survived', axis=1)
y = df['Survived']
```

In [75]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
df= LogisticRegression(max_iter=1000)
```

In [79]:
```
print(X_train.dtypes)
```

```
Pclass     int64
Sex       object
SibSp      int64
Parch      int64
dtype: object
```

In [80]:
```
X_train = pd.get_dummies(X_train, drop_first=True)
```

```
X_test = pd.get_dummies(X_test, drop_first=True)
```

In [81]:
```
X_test = X_test.reindex(columns=X_train.columns, fill_value=0)
```

In [82]:
```
log = LogisticRegression(random_state=0)
log.fit(X_train, y_train)
```

Out[82]:
```
LogisticRegression(random_state=0)
```

In [83]:
```
pred=print(log.predict(X_test))
```

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

In [84]:
```
print(Y_test)
```

```
     Survived
495         0
648         0
278         0
31          1
255         1
..        ...
780         1
837         0
215         1
833         0
372         0

[179 rows x 1 columns]
```

In [85]:
```
data = {
    'PassengerId': [1, 2, 3, 4, 5],
    'Survived': [1, 0, 0, 1, 1]
}
df = pd.DataFrame(data)

def check_survival_status(survived):
    if survived == 1:
        return 'Survived'
    else:
        return 'Not Survived'
```

In [86]:
```
df['Survival_Status'] = df['Survived'].apply(check_survival_status)
```

In [87]:
```
df
```

Out[87]:

| | PassengerId | Survived | Survival_Status |
|---|---|---|---|
| **0** | 1 | 1 | Survived |
| **1** | 2 | 0 | Not Survived |
| **2** | 3 | 0 | Not Survived |
| **3** | 4 | 1 | Survived |
| **4** | 5 | 1 | Survived |

In [ ]: