Task :1 Titanic Survival Prediction

Author : Anushka Khedekar

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
titanic_data = pd.read_csv('Titanic-Dataset.csv')
titanic_data
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

891 rows × 12 columns

In [4]:
```
titanic_data.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 [5]:
```
titanic_data.describe()
```

Out[5]:

	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 [6]:
```
titanic_data.isnull().sum()
```

Out[6]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

To check Age and Cabin has Null values or blank

In [8]:
```
#  we will fill blank with median value
titanic_data['Age'].fillna(titanic_data['Age'].median(), inplace=True)
```

In [9]:
```
# Count the Embarked
titanic_data['Embarked'].value_counts()
```

Out[9]:
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

In [10]:
```
# replace blanks with mode value
titanic_data['Embarked'].fillna('S', inplace=True)
```

In [11]:
```
# Last check null value and Dataset
print(titanic_data.isnull().sum())
print(titanic_data.head())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
dtype: int64
   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
```

Exploratory Data Analysis

Survival variable describe as

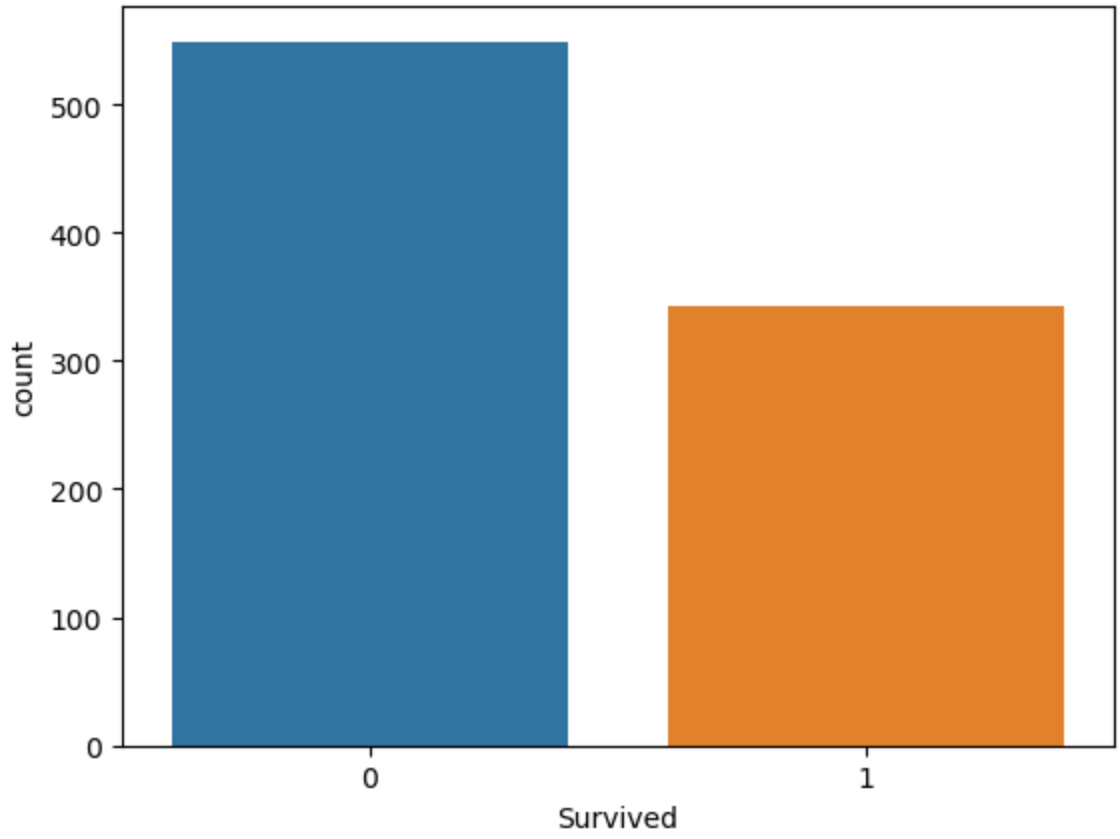
0 = No, 1=Yes

In [12]:
```
titanic_data['Survived'].value_counts()
```

Out[12]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [13]:
```
sns.countplot(data=titanic_data,x='Survived')
```

Out[13]:
```
<Axes: xlabel='Survived', ylabel='count'>
```

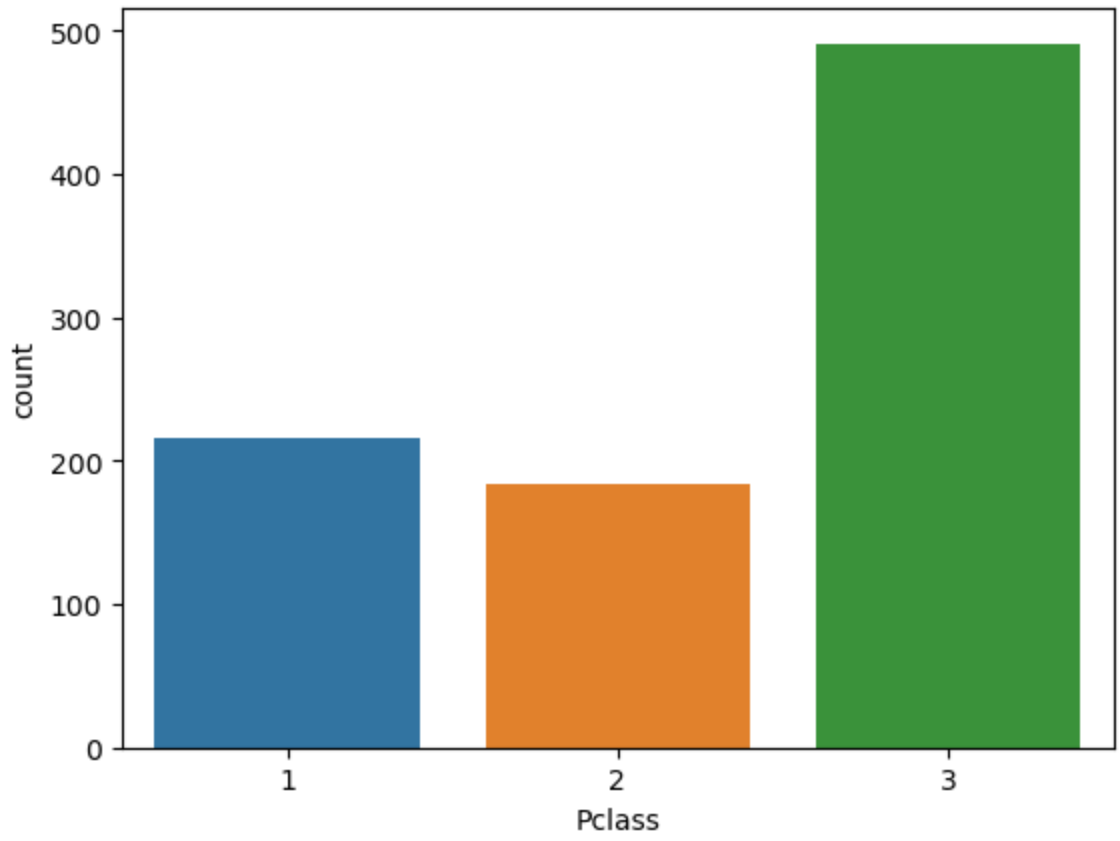


Pclass data describe as:

1st = Upper 2nd = Middle 3rd = Lower

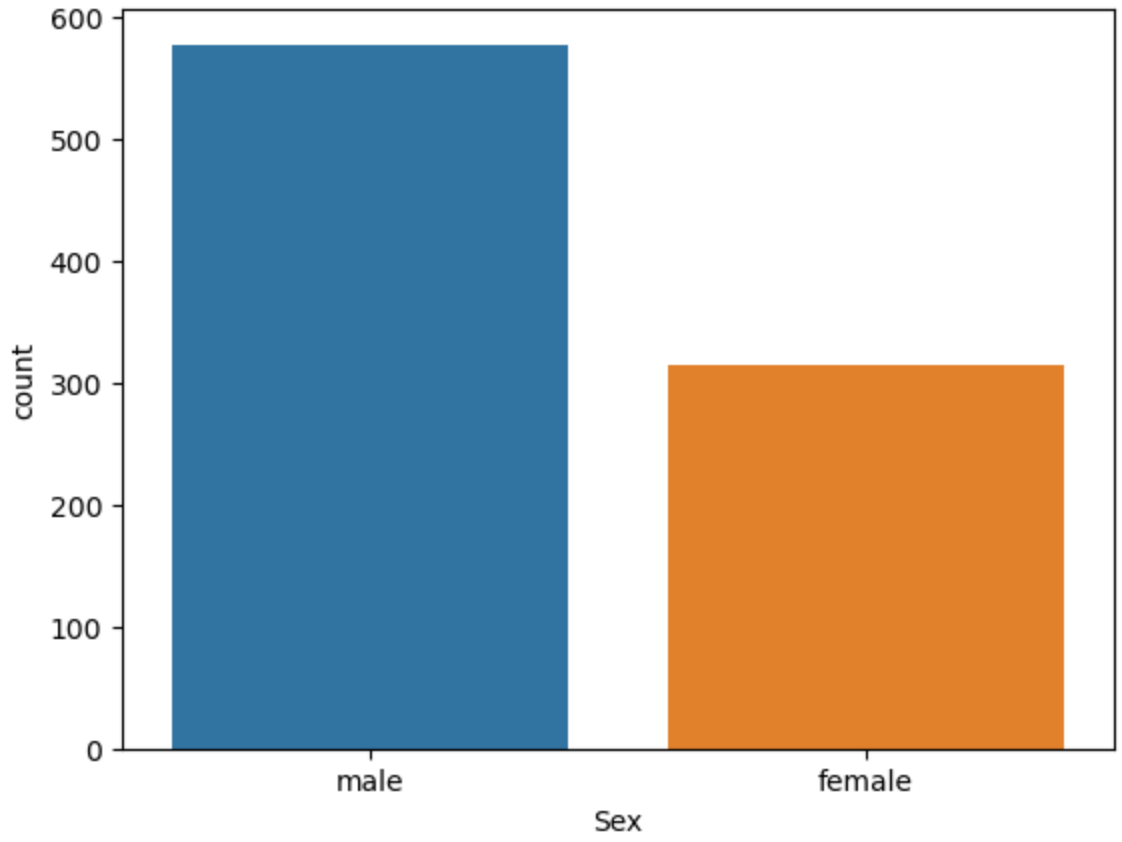
In [14]:
```
sns.countplot(data=titanic_data,x='Pclass')
```

Out[14]:
```
<Axes: xlabel='Pclass', ylabel='count'>
```

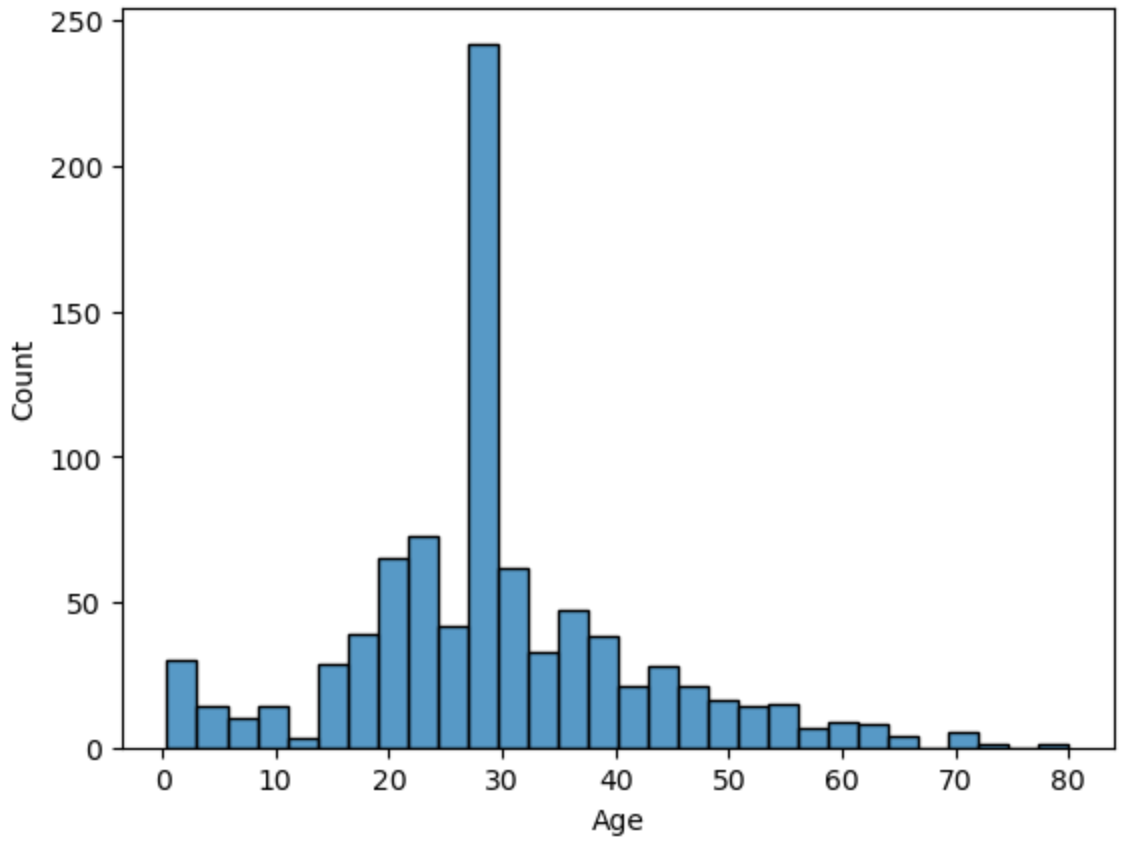


In [15]:
```
sns.countplot(data=titanic_data,x='Sex')
```

Out[15]:
```
<Axes: xlabel='Sex', ylabel='count'>
```

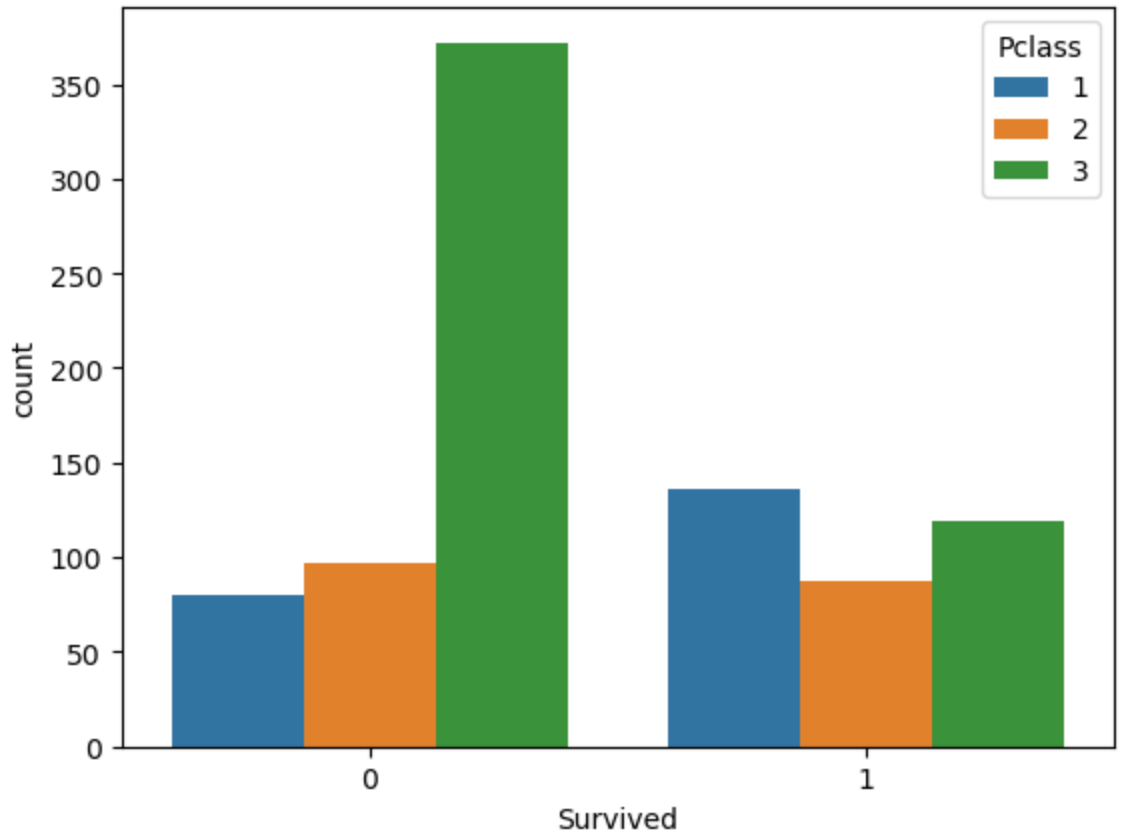


In [16]:
```
sns.histplot(data=titanic_data, x='Age')
plt.show()
```



Survival count WRT pclass

In [17]:
```
sns.countplot(x=titanic_data['Survived'], hue=titanic_data['Pclass'])
plt.show()
```

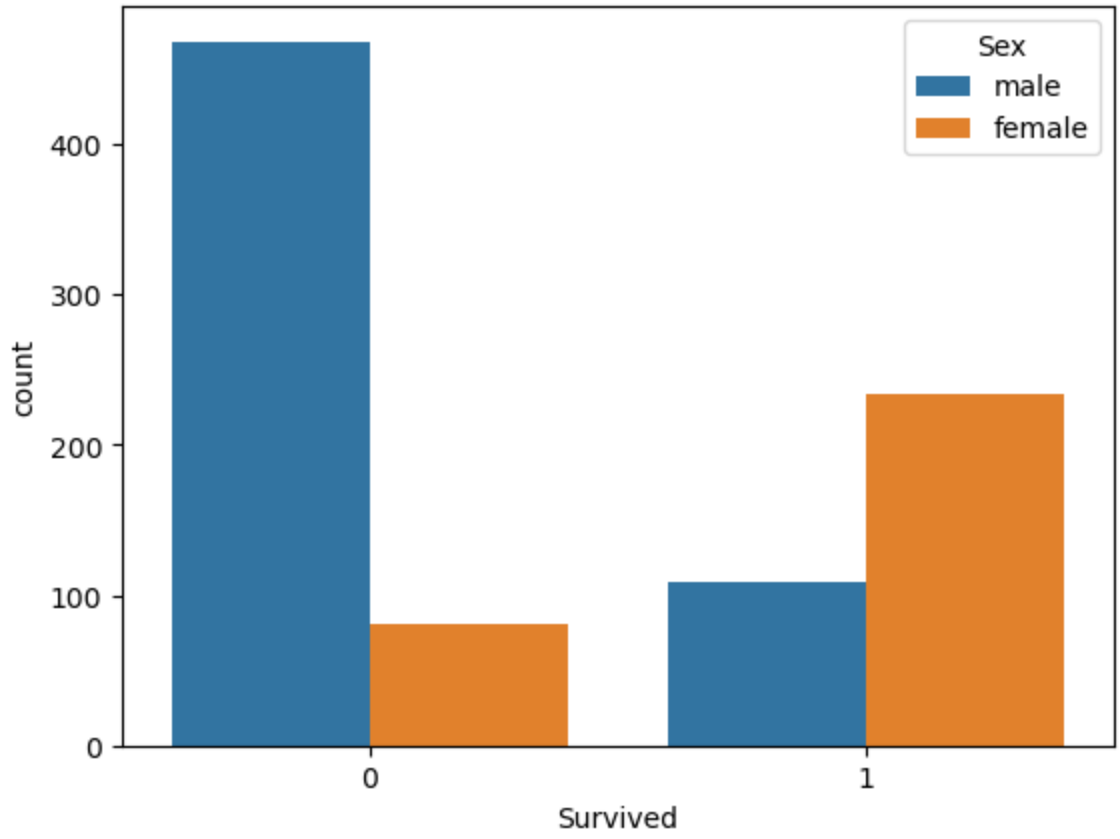


Survival count WRT gender

In [18]:
```
titanic_data['Sex'].head()
```

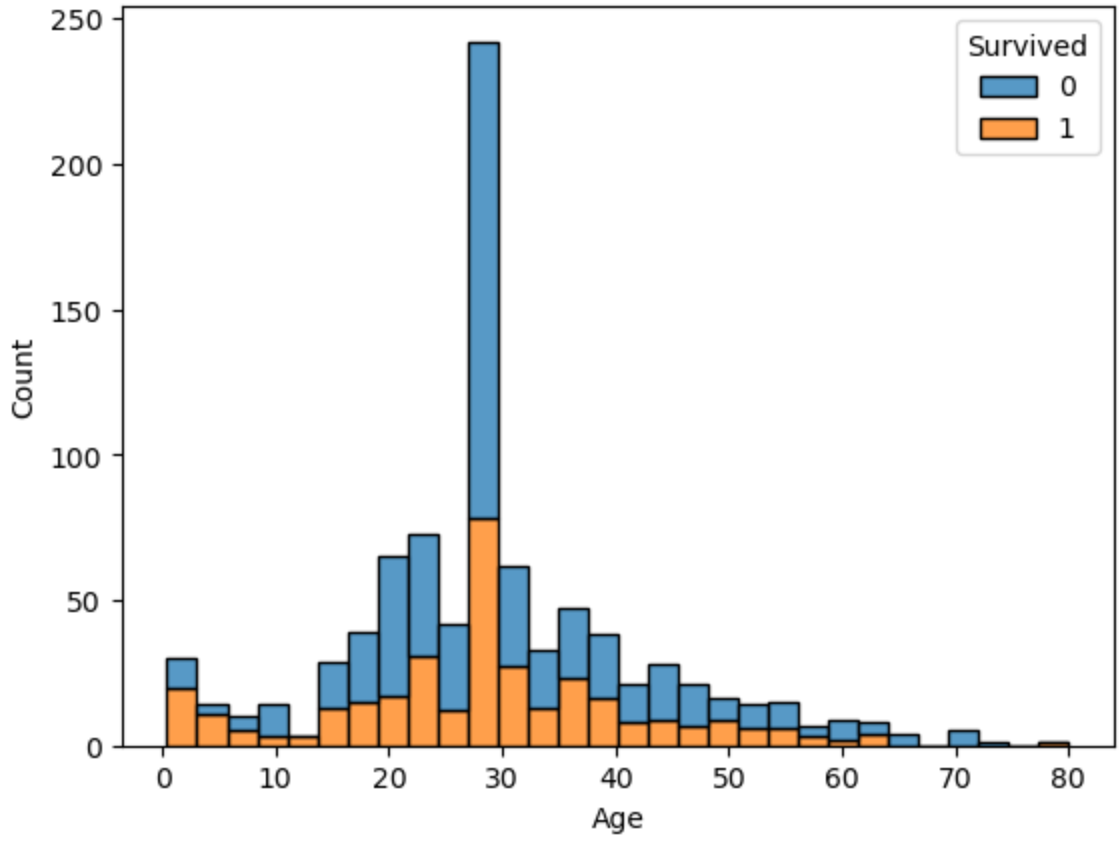
Out[18]:
```
0      male
1    female
2    female
3    female
4      male
Name: Sex, dtype: object
```

In [20]:
```
sns.countplot(x=titanic_data['Survived'], hue=titanic_data['Sex'])
plt.show()
```



Survival count WRT age

In [19]:
```
sns.histplot(x=titanic_data['Age'], hue=titanic_data['Survived'], multiple='stack')
plt.show()
```

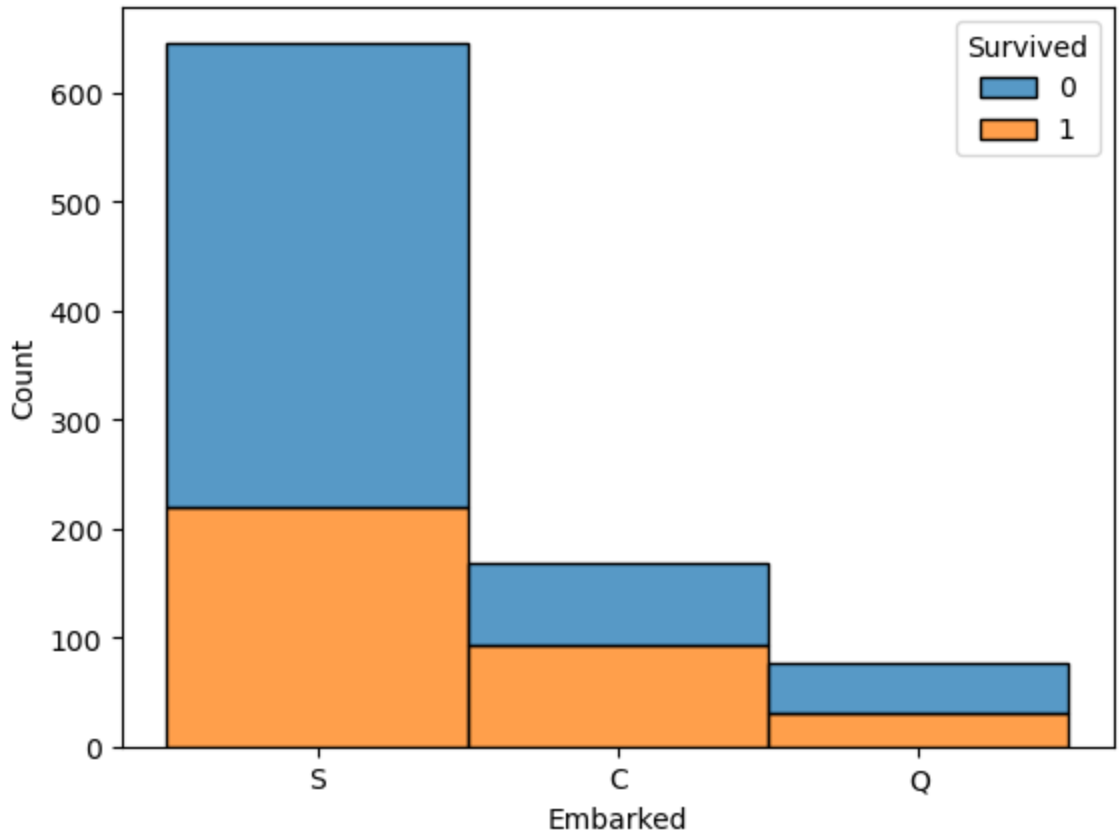


Survival count WRT Embarked

tion

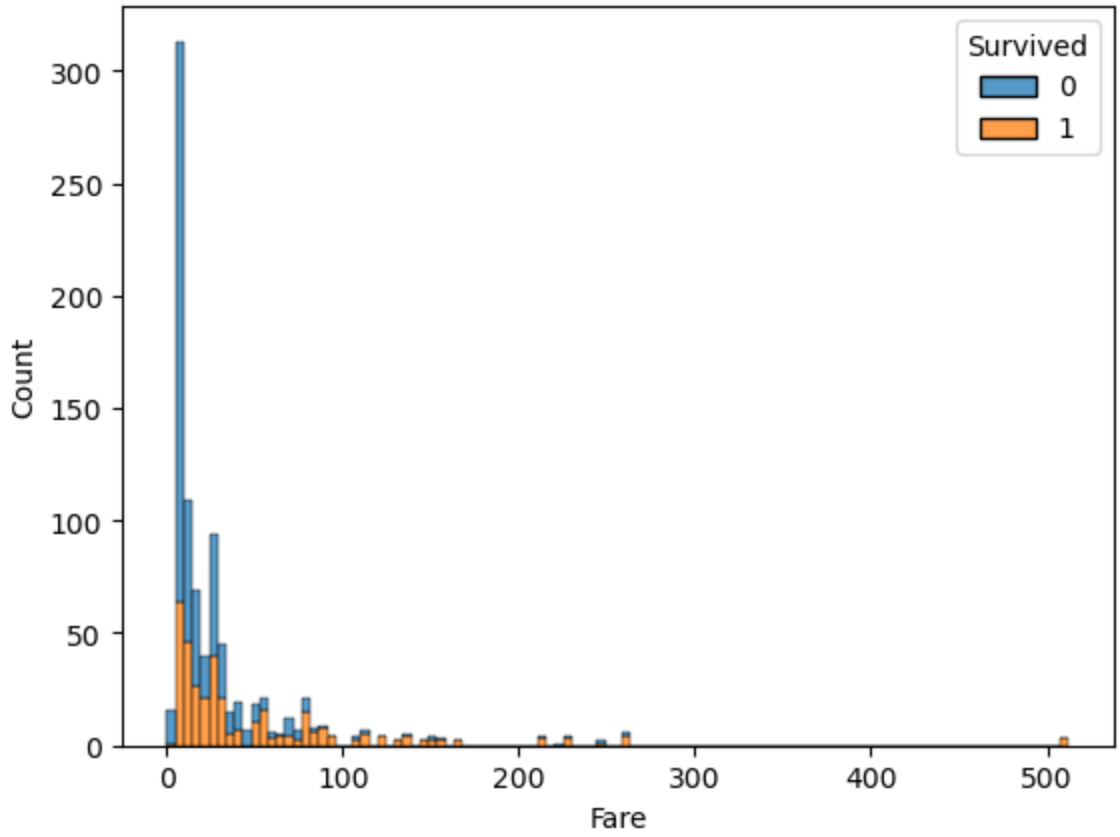
C = Cherbourg, Q = Queenstown, S = Southampton

In [20]:
```
sns.histplot(x=titanic_data['Embarked'], hue=titanic_data['Survived'], multiple='stack')
plt.show()
```



Survival count WRT fare

In [21]:
```
sns.histplot(x=titanic_data['Fare'], hue=titanic_data['Survived'], multiple='stack')
plt.show()
```



Featuring Engineering

before modeling the data, transform gender(Sex) into numeric Male - 1 Female - 0

Use LabelEncoder from sklearn library

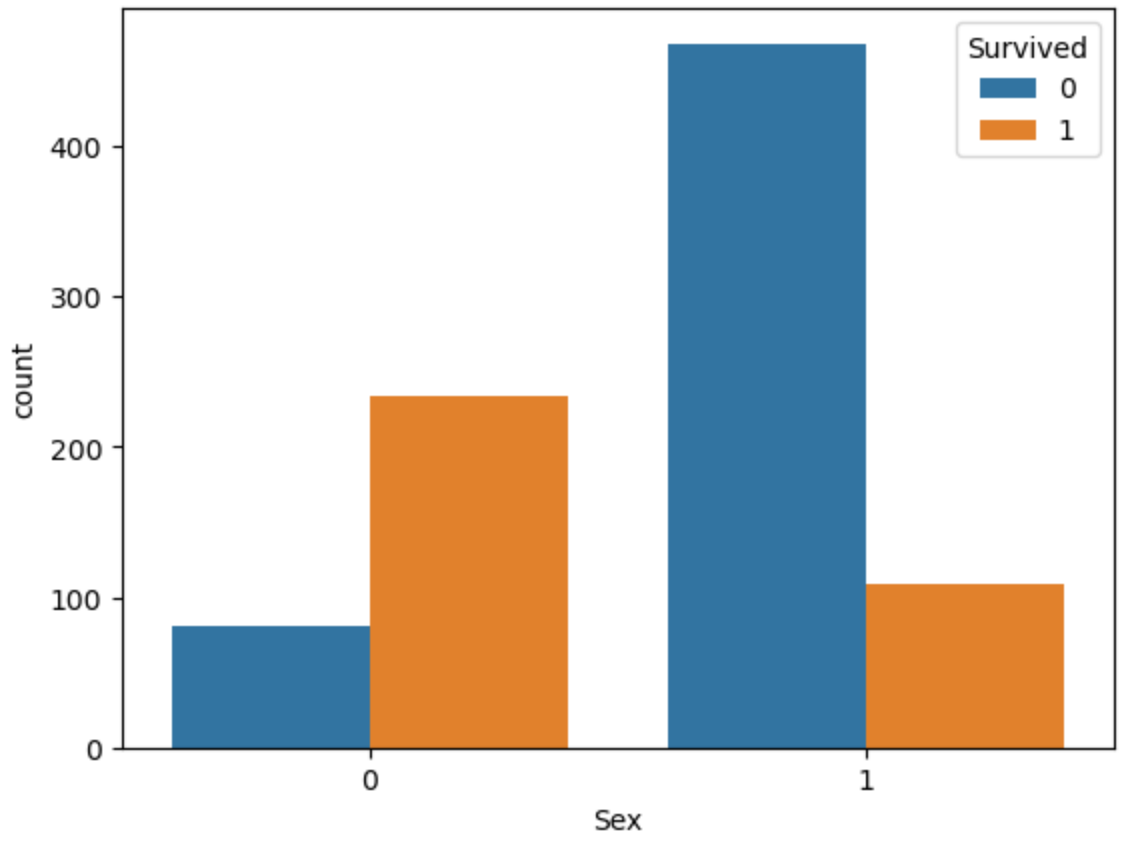
In [22]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
titanic_data['Sex'] = labelencoder.fit_transform(titanic_data['Sex'])

titanic_data.head()
```

Out[22]:

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

In [23]:
```
sns.countplot(x=titanic_data['Sex'],hue=titanic_data['Survived'])
plt.show()
```



In [24]:
```
titanic_data.head()
```

Out[24]:

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

Modeling

In [25]:
```
X=titanic_data[['Sex', 'Pclass']]
Y=titanic_data['Survived']
```

Split data into test and train by using Sklearn library

In [26]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2, random_state=0)
```

Create training Model

In [27]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,precision_score,confusion_matrix
log = LogisticRegression(random_state = 0)
log.fit(X_train, Y_train)
```

Out[27]:
```
▼        LogisticRegression
LogisticRegression(random_state=0)
```

create Prediction model

In [28]:
```
pred = log.predict(X_test)
pred
```

Out[28]:
```
array([0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0], dtype=int64)
```

In [29]:
```
print("Accuracy_score :", accuracy_score(Y_test, pred))
print("Matrix :",confusion_matrix(Y_test,pred))
```

```
Accuracy_score : 0.7877094972067039
Matrix : [[92 18]
 [20 49]]
```

In [30]:
```
Y_test
```

Out[30]:
```
495    0
648    0
278    0
31     1
255    1
      ..
780    1
837    0
215    1
833    0
372    0
Name: Survived, Length: 179, dtype: int64
```

In [31]:
```
submission=X.iloc[:,:].values
y_final=log.predict(submission)
y_final.shape
```

Out[31]:
```
(891,)
```

In [32]:
```
final = pd.DataFrame()
final["Sex"]= X['Sex']
final["survived"]=y_final
```

In [33]:
```
final.to_csv("submission.csv",index=False)
```

Trainig is completed, now check

predict([[Pclass, Sex]]) => survived or not survived

In [34]:
```
import warnings
warnings.filterwarnings("ignore")

result = log.predict([[5,0]])
if(result == 0):
    print("So sorry, Not Survived")
else:
    print("Survived")
```

```
So sorry, Not Survived
```