Prodigy_Infotech

Task 2

Exploratory Data Analysis (EDA)

Performed Exploratory Data Analysis on Titanic Dataset and obtain the meaningful insights and find out the patterns in the dataset

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

#Ignore warning
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Importing the dataset
Data = pd.read_csv(r"C:\Users\acer\Desktop\Machine Learning PDF\train (1).csv"
Data
```

Out[2]:

	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
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

In [3]:
```
# Data Exploration
```

In [4]:
```
Data.size
```

Out[4]:
```
10692
```

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

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

In [6]:
```
Data.columns
```

Out[6]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [7]:
```
Data.dtypes
```

Out[7]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [8]:
```
# Age Contain Dtype Float as Age Comes never in Decimal it should be convertee
```

In [ ]:
```

```

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

In [10]:
```
Data.isna().sum()
```

Out[10]:
```
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
```

In [11]:
```
Data.isnull().sum()
```

Out[11]:
```
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
```

In [12]:
```
Data['Age'] = Data['Age'].fillna(round(Data['Age'].mean()))
```

In [13]:
```
Data['Cabin'].unique()
```

Out[13]:
```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

In [14]:
```
Data['Cabin'] = Data['Cabin'].fillna(method='bfill')
```

In [15]:
```
Data['Cabin'] = Data['Cabin'].dropna(inplace=True)
```

In [19]:
```
Data.isnull().sum()
```

Out[19]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       2
dtype: int64
```

In [18]:
```
Data.drop(columns=['Cabin'] , axis=1 , inplace=True)
```

In [20]:
```
Data['Embarked'].unique()
```

Out[20]:
```
array(['S', 'C', 'Q', nan], dtype=object)
```

In [21]:
```
Data['Embarked'].fillna(method='bfill',inplace=True)
```

In [22]:
```
Data.isnull().sum()
```

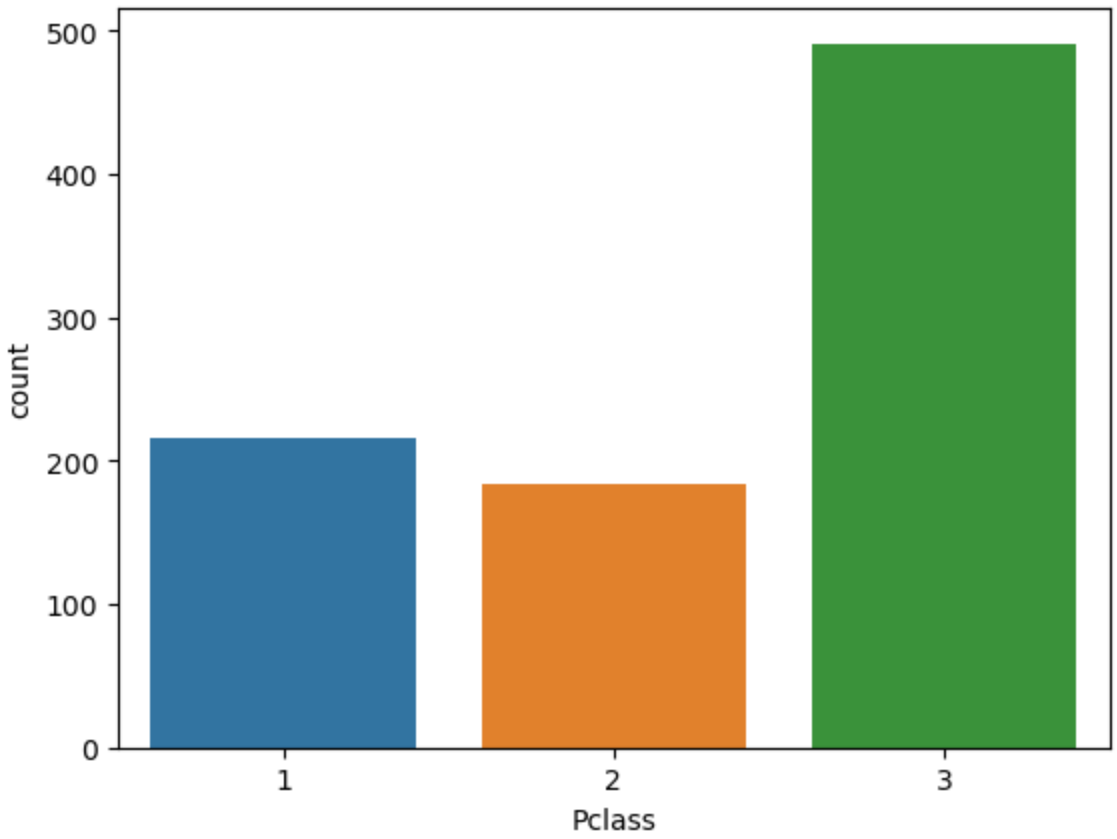
Out[22]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Exploratory Data Analysis

In [29]:
```
print(Data['Pclass'].value_counts())

print()
sns.countplot(x = 'Pclass' , data=Data)
plt.show()
```

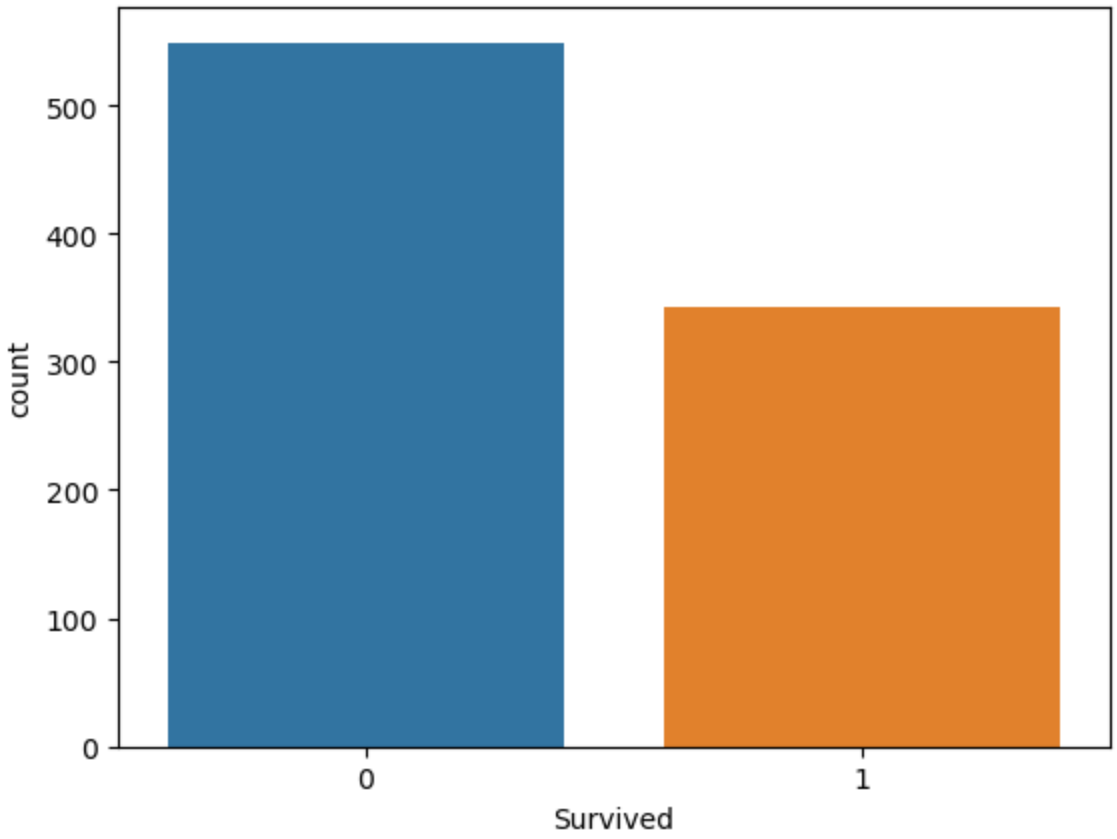
```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```



In [36]:
```
print(Data['Survived'].value_counts())
sns.countplot(x = 'Survived'  , data=Data)
```

```
0    549
1    342
Name: Survived, dtype: int64
```

Out[36]: <AxesSubplot:xlabel='Survived', ylabel='count'>

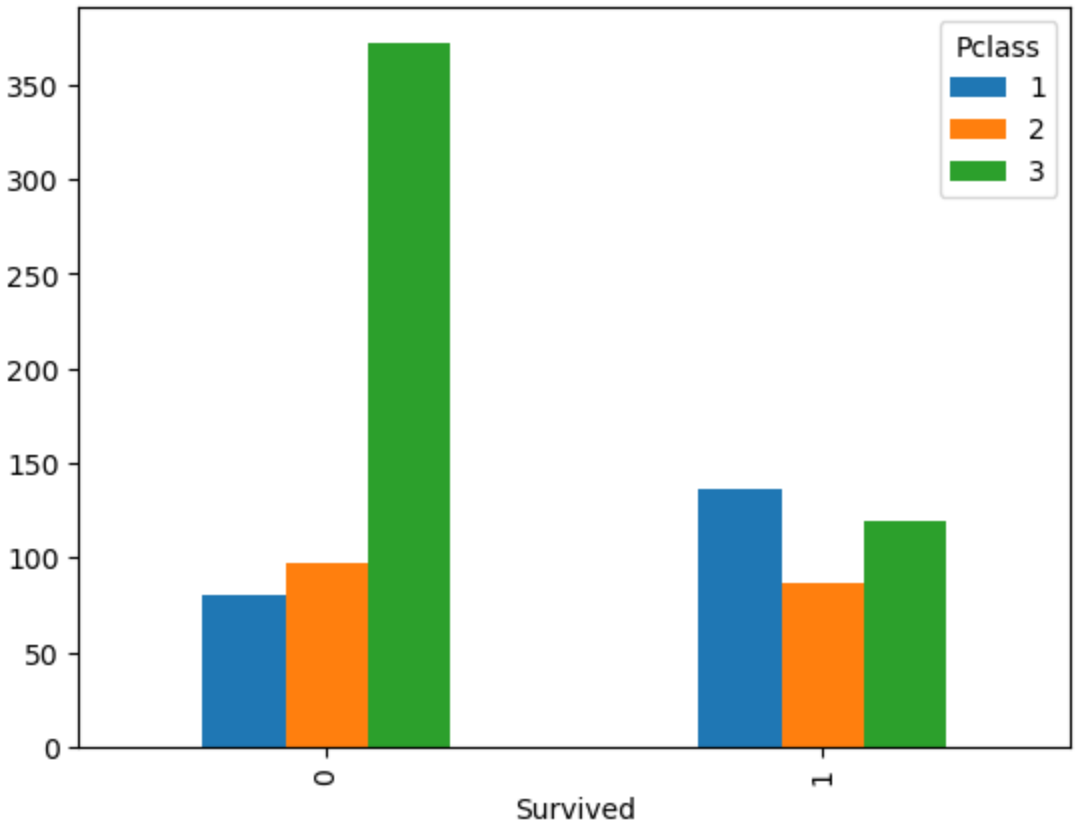


In [47]:
```
class_wise_survived = pd.crosstab(index=Data['Survived'] , columns=Data['Pclas
class_wise_survived
```

Out[47]:

Pclass	1	2	3
Survived			
0	80	97	372
1	136	87	119

In [68]:
```
class_wise_survived.plot(kind = 'bar')
plt.show()
```



In [55]:
```
Data['Sex'].value_counts()
```

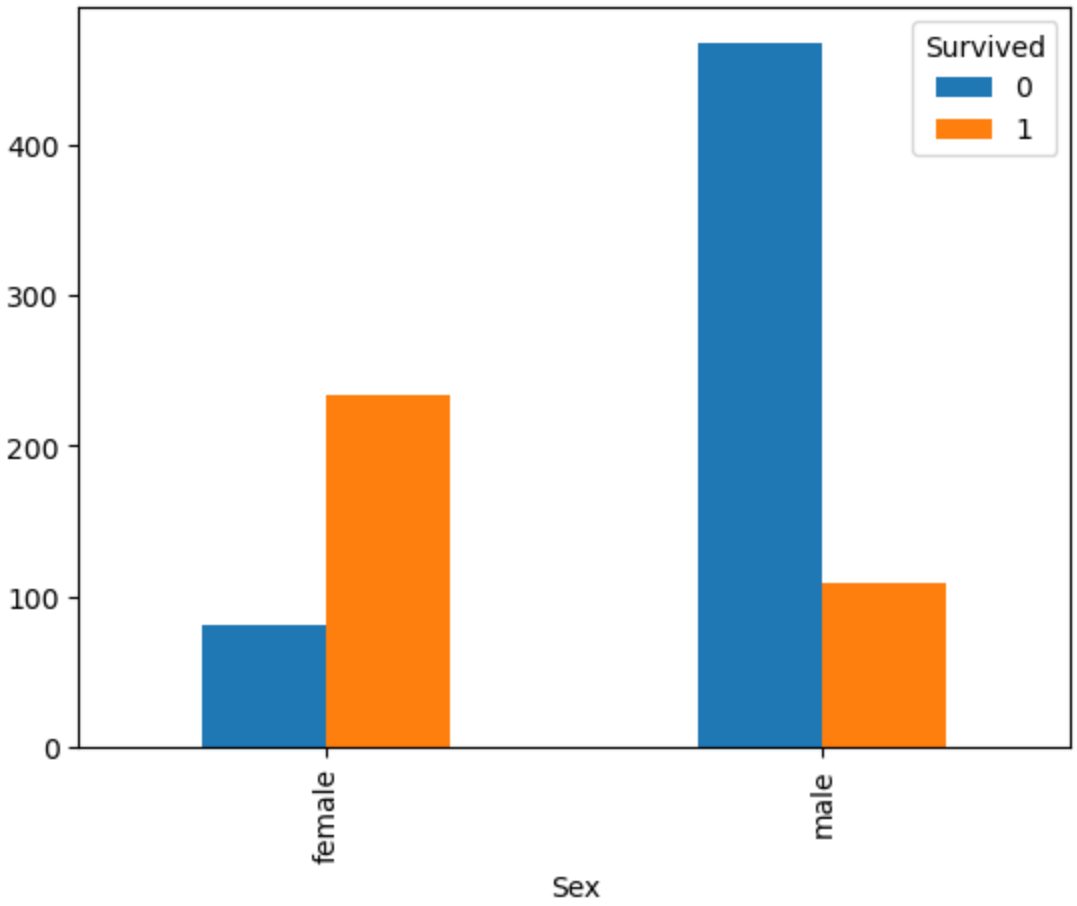
Out[55]:
```
male      577
female    314
Name: Sex, dtype: int64
```

In [65]:
```
pd.crosstab(index=Data['Sex'] ,columns=Data['Survived'])
```

Out[65]:

Survived	0	1
Sex		
female	81	233
male	468	109

In [67]:
```
pd.crosstab(index=Data['Sex'] ,columns=Data['Survived']).plot(kind = 'bar')
plt.show()
```



In [82]:
```
Agg_Data = Data['Fare'].aggregate(['min' , 'max' ,'mean' , 'median' , 'count',
```

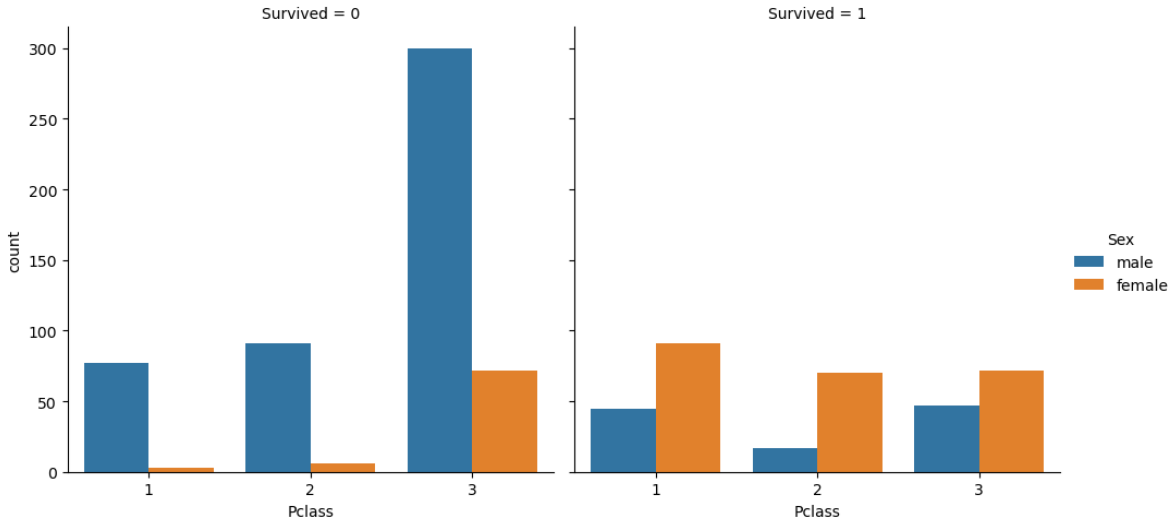
In [83]:
```
pd.DataFrame(Agg_Data)
```

Out[83]:

	Fare
min	0.000000
max	512.329200
mean	32.204208
median	14.454200
count	891.000000
std	49.693429

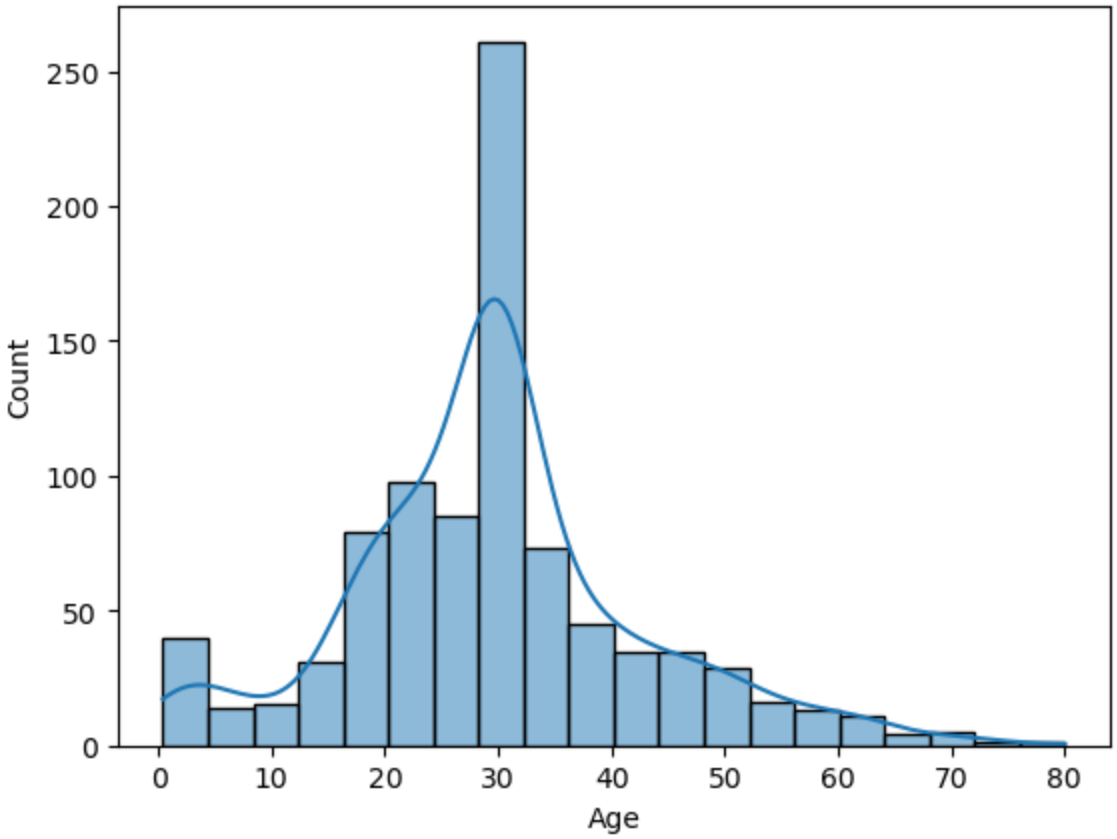
In [87]:
```
sns.catplot(x='Pclass', hue='Sex', col='Survived', data=Data, kind='count')
```

Out[87]:
```
<seaborn.axisgrid.FacetGrid at 0x21f440b9fa0>
```



In [96]:
```
sns.histplot(Data['Age'] , kde=True , bins=20)
```

Out[96]:
```
<AxesSubplot:xlabel='Age', ylabel='Count'>
```



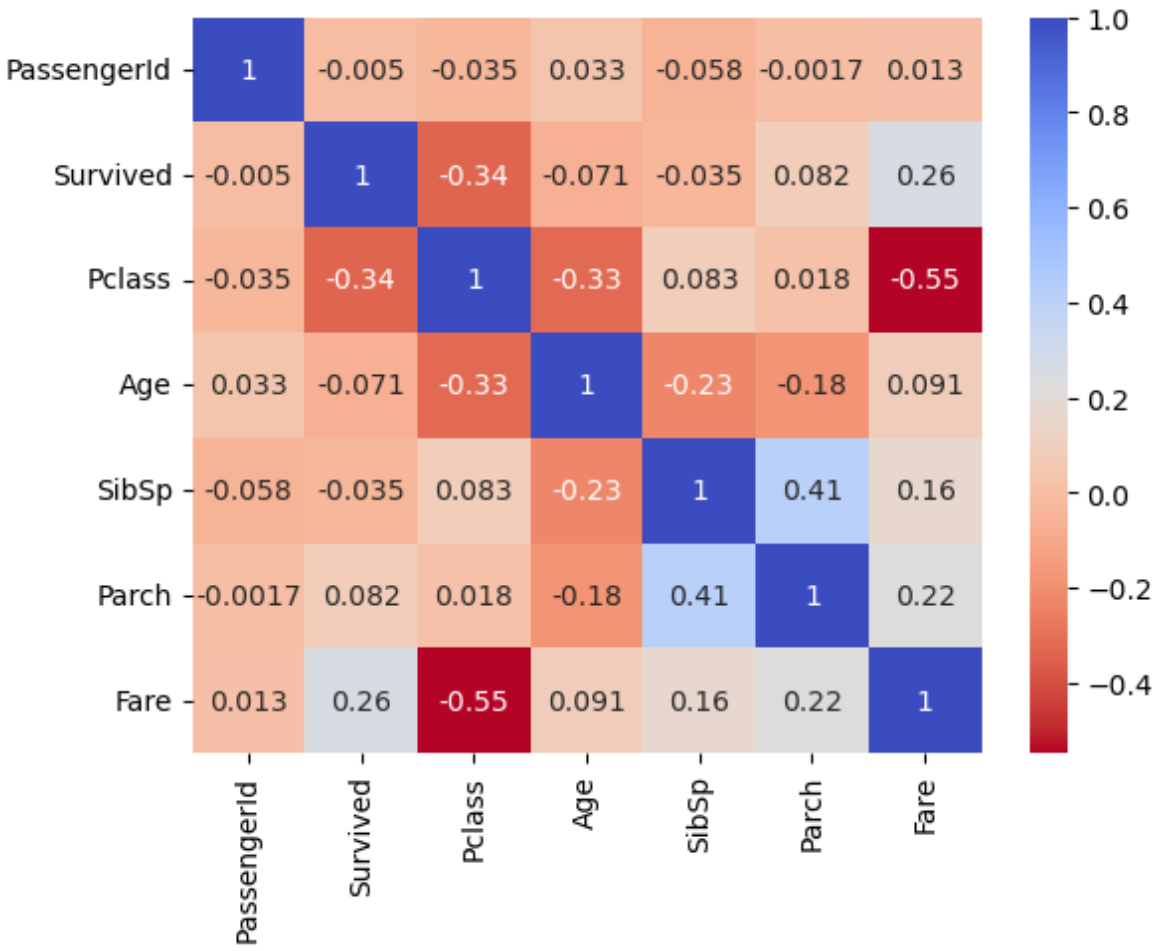
In [97]:
```
Data.corr()
```

Out[97]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.033019	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.070657	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.329727	0.083081	0.018443	-0.549500
Age	0.033019	-0.070657	-0.329727	1.000000	-0.232440	-0.180330	0.090632
SibSp	-0.057527	-0.035322	0.083081	-0.232440	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.180330	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.090632	0.159651	0.216225	1.000000

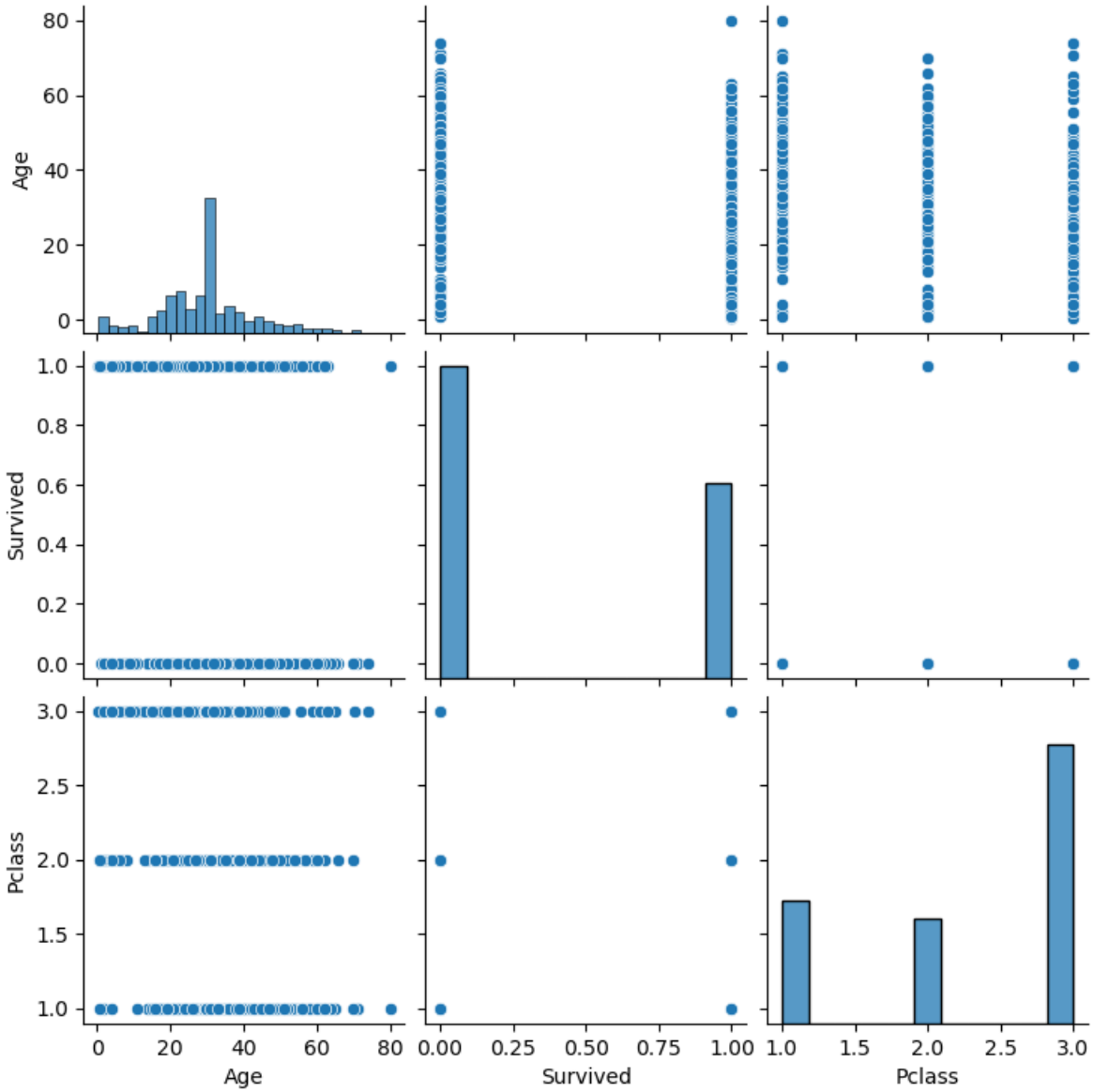
In [106]:
```
sns.heatmap(data=Data.corr() , annot=True , cmap='coolwarm_r')
```

Out[106]: <AxesSubplot:>

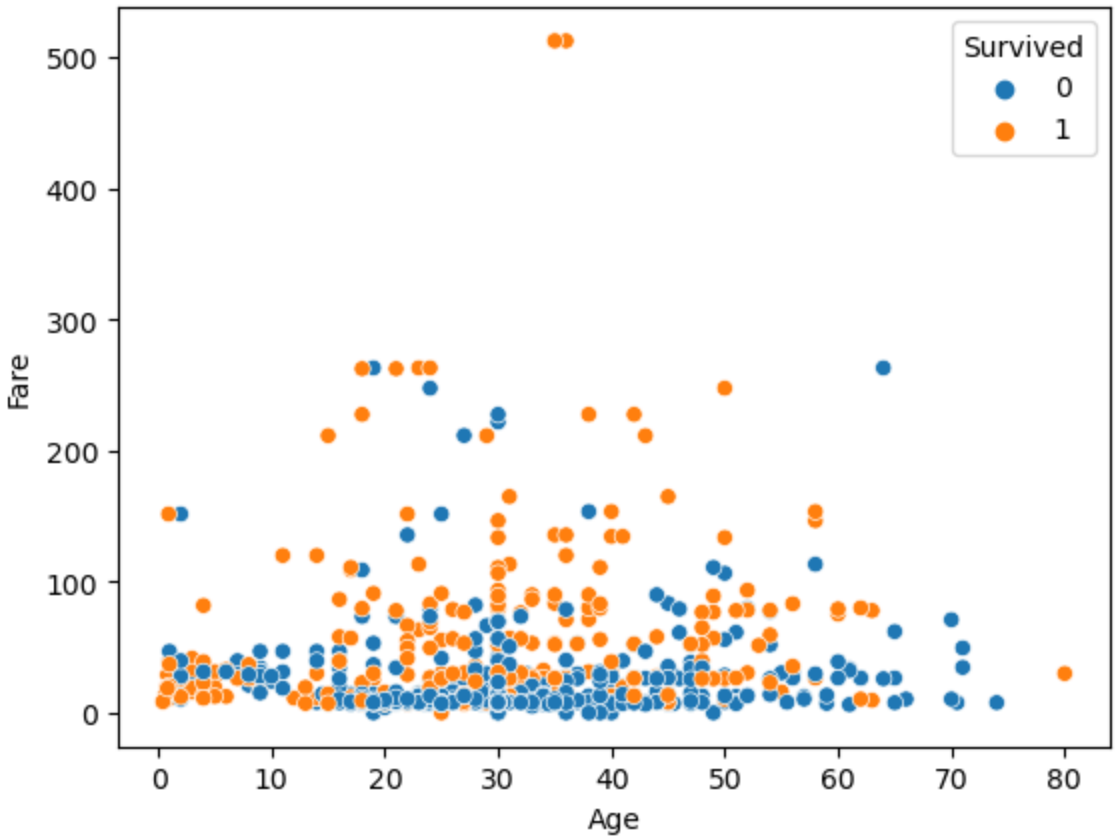


In [111]:
```
sns.pairplot(Data[['Age' , "Survived" , "Pclass"]])
```

Out[111]:
```
<seaborn.axisgrid.PairGrid at 0x21f4a5d45e0>
```



In [114]:
```
sns.scatterplot(x='Age' , y = 'Fare' , hue='Survived' ,  data=Data)
plt.show()
```



In [ ]:
```

```