Exploratory Data Analysis on Titanic

Introduction

This dataset is taken from this link https://www.kaggle.com/c/titanic/data (https://www.kaggle.com/c/titanic/data)

The data has been split into two groups:

1) data_train (train.csv): The training set should be used to build your machine learning models. For the training set, we provide the outcome (also known as the “ground truth”) for each passenger. Your model will be based on “features” like passengers’ gender and class. You can also use feature engineering to create new features.

2) data_test (test.csv): The test set should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes. For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Titanic.

But here i will just use the Exploratory Data Analysis, Cleaning and Visualization. Model will be use soon, IA

Importing Libraries

In [79]:
```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)


from sklearn.metrics import confusion_matrix # For confusion-matrix
from sklearn.metrics import precision_score # for Precision-Score
```

Importing Dataset

In [80]:
```
data_train = pd.read_csv(r"C:\Users\zahid\Downloads\Titanic Dataset EDA\train
data_test = pd.read_csv(r"C:\Users\zahid\Downloads\Titanic Dataset EDA\test.c
```

Preliminary Data Exploration

The attributes have the following meaning:

- **PassengerId**: a unique identifier for each passenger
- **Survived**: that's the target, 0 means the passenger did not survive, while 1 means he/she survived.
- **Pclass**: passenger class.
- **Name**, **Sex**, **Age**: self-explanatory
- **SibSp**: how many siblings & spouses of the passenger aboard the Titanic.
- **Parch**: how many children & parents of the passenger aboard the Titanic.
- **Ticket**: ticket id
- **Fare**: price paid (in pounds)
- **Cabin**: passenger's cabin number

- **Embarked**: where the passenger embarked the Titanic

For Training Dataset

In [81]:
```
data_train.head()
```

Out[81]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
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 [82]:
```
data_train.tail()
```

Out[82]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	Nal
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B4
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	Nal
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C14
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	Nal

In [83]:
```
data_train.describe
```

Out[83]:
```
<bound method NDFrame.describe of      PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         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
..                                                ...     ...   ...    ...
886                            Montvila, Rev. Juozas    male  27.0      0
887                     Graham, Miss. Margaret Edith  female  19.0      0
888         Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                            Behr, Mr. Karl Howell    male  26.0      0
890                              Dooley, Mr. Patrick    male  32.0      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
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]>
```

In [84]:
```
data_train.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 [85]:
```
data_train.columns
```

Out[85]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [86]:
```
data_train.index
```

Out[86]:
```
RangeIndex(start=0, stop=891, step=1)
```

In [87]:
```
data_train.shape
```

Out[87]:
```
(891, 12)
```

In [88]:
```
data_train.nunique()
```

Out[88]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

In [89]:
```
data_train["Age"].unique()
```

Out[89]:
```
array([22.  , 38.  , 26.  , 35.  ,   nan, 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
        8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
       49.  , 29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  ,
       16.  , 25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  ,
       71.  , 37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 ,
       51.  , 55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  ,
       45.5 , 20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  ,
       60.  , 10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  ,
       70.  , 24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [90]:
```
data_train['Age'].value_counts()
```

Out[90]:
```
24.00    30
22.00    27
18.00    26
19.00    25
28.00    25
         ..
36.50     1
55.50     1
0.92      1
23.50     1
74.00     1
Name: Age, Length: 88, dtype: int64
```

Missing Values and Duplicates in Train

In [91]:
```
# Checking missing values
data_train.isna().sum()
```

Out[91]:
```
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 [92]:
```
train_data_clean = data_train.drop(columns="Cabin", axis=1)
train_data_clean
```

Out[92]:

	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 × 11 columns

In [93]:
```
train_data_clean.describe()
```

Out[93]:

	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 [132]:
```
print('Size of Train Data dataset is  :',train_data_clean.size)
```

```
Size of Train Data dataset is  : 10692
```

In [95]:
```
train_data_clean.columns
```

Out[95]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Embarked'],
      dtype='object')
```

In [96]:
```
mean = train_data_clean["Age"].mean()
print(" Mean of the Age in training data: ", mean)
```

```
 Mean of the Age in training data:  29.69911764705882
```

In [97]:
```
train_data_clean.groupby(["Survived", "Sex"]).mean()
```

Out[97]:

		PassengerId	Pclass	Age	SibSp	Parch	Fare
Survived	**Sex**						
0	**female**	434.851852	2.851852	25.046875	1.209877	1.037037	23.024385
	male	449.121795	2.476496	31.618056	0.440171	0.207265	21.960993
1	**female**	429.699571	1.918455	28.847716	0.515021	0.515021	51.938573
	male	475.724771	2.018349	27.276022	0.385321	0.357798	40.821484

In [ ]:
```

```

For Testing Dataset

In [98]:
```
data_test.head()
```

Out[98]:

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

In [99]:
```
data_test.head()
```

Out[99]:

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

In [100]:
```
data_test.describe
```

Out[100]:
```
<bound method NDFrame.describe of      PassengerId  Pclass
Name  \
0            892       3                                Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                              Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..           ...     ...                                           ...
413         1305       3                            Spector, Mr. Woolf
414         1306       1                  Oliva y Ocana, Dona. Fermina
415         1307       3                  Saether, Mr. Simon Sivertsen
416         1308       3                           Ware, Mr. Frederick
417         1309       3                      Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embarked
0      male  34.5      0      0              330911    7.8292   NaN        Q
1    female  47.0      1      0              363272    7.0000   NaN        S
2      male  62.0      0      0              240276    9.6875   NaN        Q
3      male  27.0      0      0              315154    8.6625   NaN        S
4    female  22.0      1      1             3101298   12.2875   NaN        S
..      ...   ...    ...    ...                 ...       ...   ...      ...
413    male   NaN      0      0           A.5. 3236    8.0500   NaN        S
414  female  39.0      0      0            PC 17758  108.9000  C105        C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416    male   NaN      0      0              359309    8.0500   NaN        S
417    male   NaN      1      1                2668   22.3583   NaN        C

[418 rows x 11 columns]>
```

In [101]:
```
data_test.columns
```

Out[101]:
```
Index(['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [102]:
```
data_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [103]:
```
data_test.shape
```

Out[103]:
```
(418, 11)
```

In [104]:
```
data_test.nunique()
```

Out[104]:
```
PassengerId    418
Pclass           3
Name           418
Sex              2
Age             79
SibSp            7
Parch            8
Ticket         363
Fare           169
Cabin           76
Embarked         3
dtype: int64
```

In [105]:
```
data_test['Age'].value_counts()
```

Out[105]:
```
21.0    17
24.0    17
22.0    16
30.0    15
18.0    13
        ..
76.0     1
28.5     1
22.5     1
62.0     1
38.5     1
Name: Age, Length: 79, dtype: int64
```

Missing Values and Duplicates in Test

In [106]:
```
# Checking missing values
data_test.isna().sum()
```

Out[106]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [107]:
```
test_data_clean = data_test.drop(columns="Cabin", axis=1)
test_data_clean
```

Out[107]:

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

418 rows × 10 columns

In [108]:
```
test_data_clean.describe()
```

Out[108]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [131]:
```
print('Size of Test dataset is  :',test_data_clean.size)
```

```
Size of Test dataset is  : 4598
```

In [110]:
```
test_data_clean.columns
```

Out[110]:
```
Index(['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Embarked'],
      dtype='object')
```

In [111]:
```
mean = test_data_clean["Age"].mean()
print(" Mean of the Age in testing data: ", mean)
```

```
 Mean of the Age in testing data:  30.272590361445783
```

In [112]:
```
test_data_clean.groupby(["Sex", "Pclass"]).mean()
```

Out[112]:

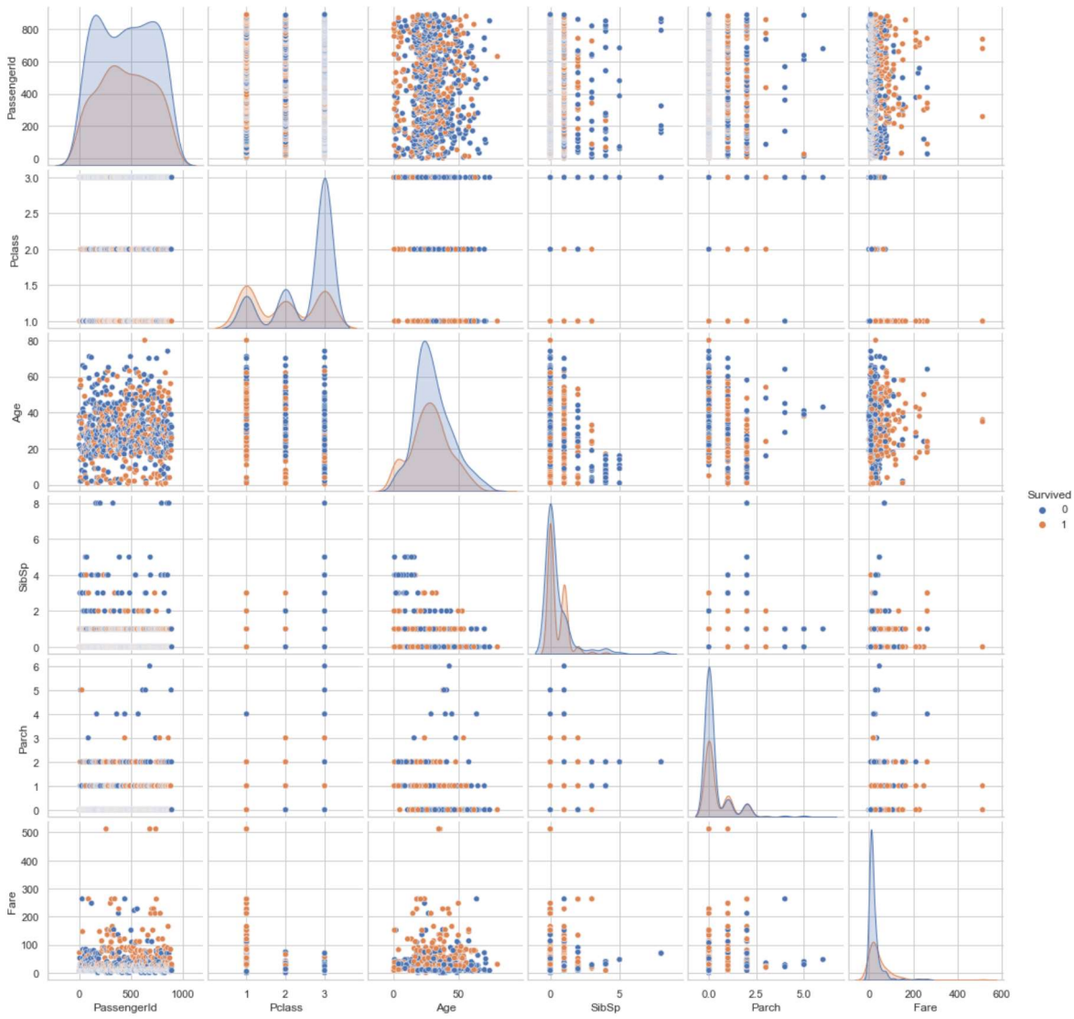
Sex	Pclass	PassengerId	Age	SibSp	Parch	Fare
female	1	1104.080000	41.333333	0.560000	0.500000	115.591168
	2	1111.200000	24.376552	0.533333	0.766667	26.438750
	3	1085.722222	23.073400	0.583333	0.597222	13.735129
male	1	1093.087719	40.520000	0.403509	0.280702	75.586551
	2	1121.142857	30.940678	0.301587	0.142857	20.184654
	3	1098.349315	24.525104	0.404110	0.328767	11.826350

In [ ]:
```
```

Visualization for both (Testing & Training)

In [113]:
```
# Data distribution of each pair of numerical variables
sns.pairplot(train_data_clean, hue = 'Survived')
```

Out[113]:
```
<seaborn.axisgrid.PairGrid at 0x2ad54a28f70>
```



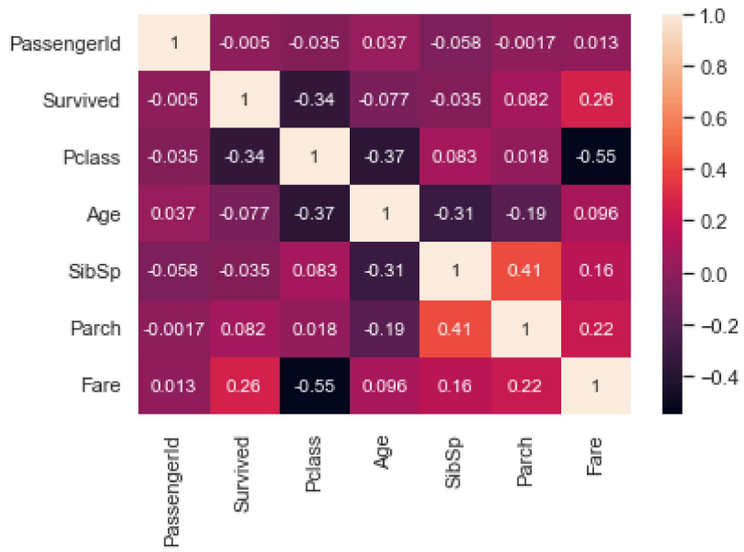
In [114]:
```
# Combining the test and train dataset into all_data
train_data_clean['set'] = 'train'
test_data_clean['set'] = 'test'

all_data = pd.concat([train_data_clean, test_data_clean], axis=0, ignore_inde
```

In [115]:
```
corr_train_clean = train_data_clean.corr()
corr_test_clean = test_data_clean.corr()
```

In [116]:
```
# Heatmap for training data
sns.heatmap(corr_train_clean, annot= True)
```

Out[116]: <AxesSubplot:>

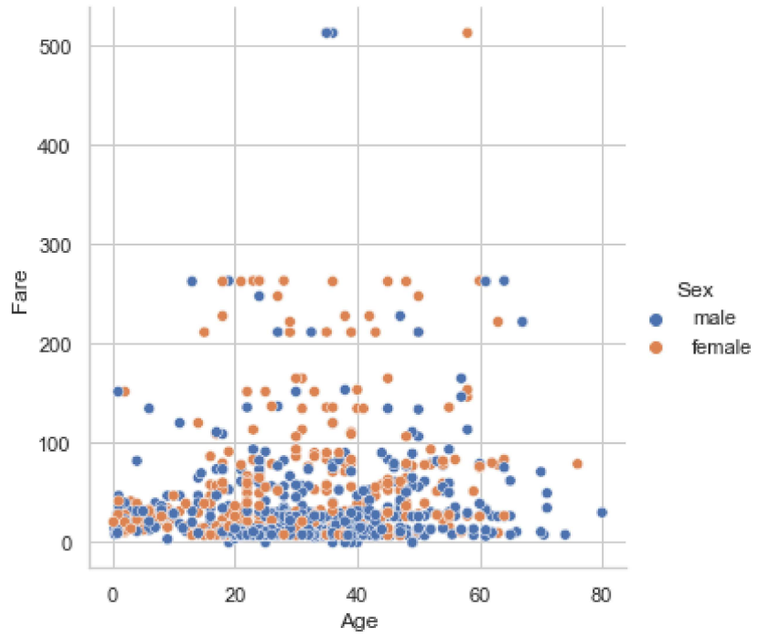


In [117]:
```
# Heatmap for training data
sns.heatmap(corr_test_clean, annot= True)
```

Out[117]: <AxesSubplot:>

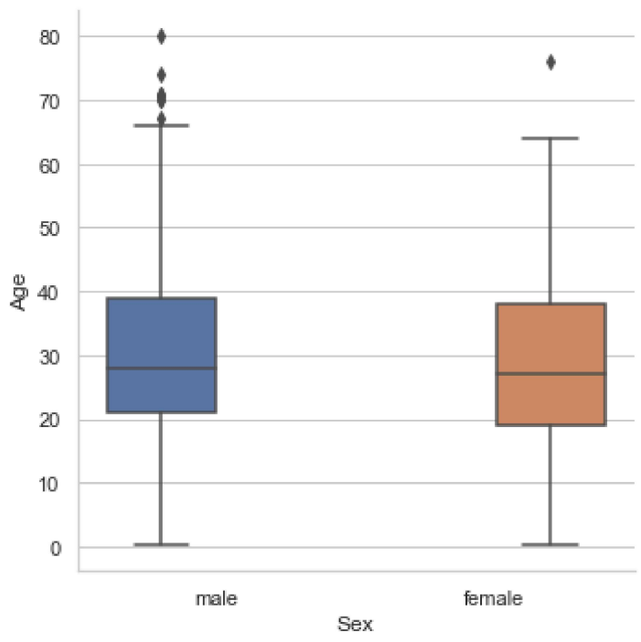
In [118]:
```
sns.relplot(x = "Age", y= "Fare", hue="Sex", data=all_data)
```

Out[118]: <seaborn.axisgrid.FacetGrid at 0x2ad56e09610>



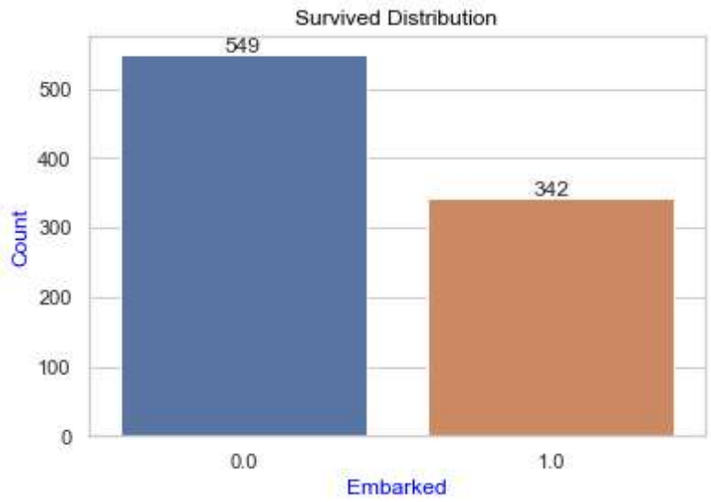
In [119]:
```
sns.catplot(x = "Sex", y= "Age",hue="Sex", data=all_data, kind="box")
```

Out[119]: <seaborn.axisgrid.FacetGrid at 0x2ad56ef7040>

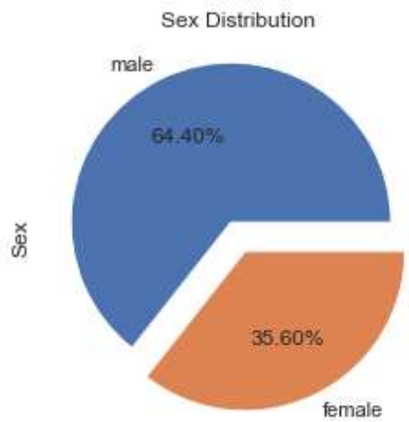


In [120]:
```
ax = sns.set(style="whitegrid")
ax = sns.countplot(data=all_data,x='Survived');
ax.bar_label(ax.containers[0])

plt.title('Survived Distribution',color='black',loc='center');
plt.xlabel('Embarked',color='Blue',loc='center')
plt.ylabel('Count',color='Blue',loc='center');
```

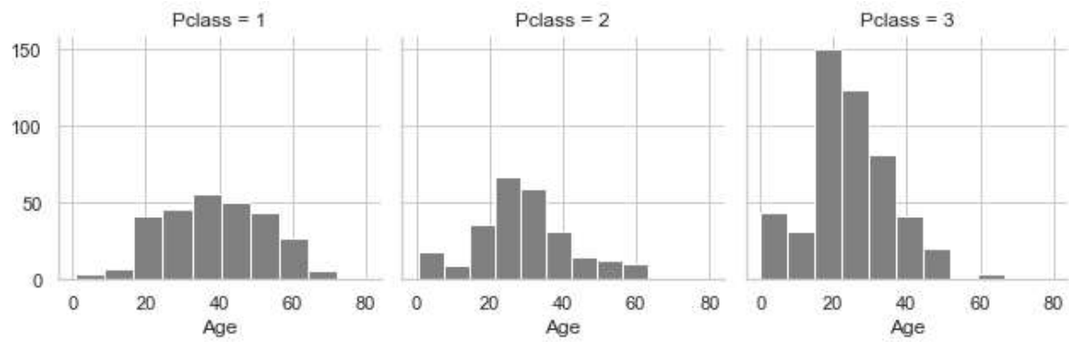


In [121]:
```
all_data['Sex'].value_counts().plot.pie(explode=[0.1, 0.1],
                                        autopct='%1.2f%%'
                                          )
plt.title('Sex Distribution');
```

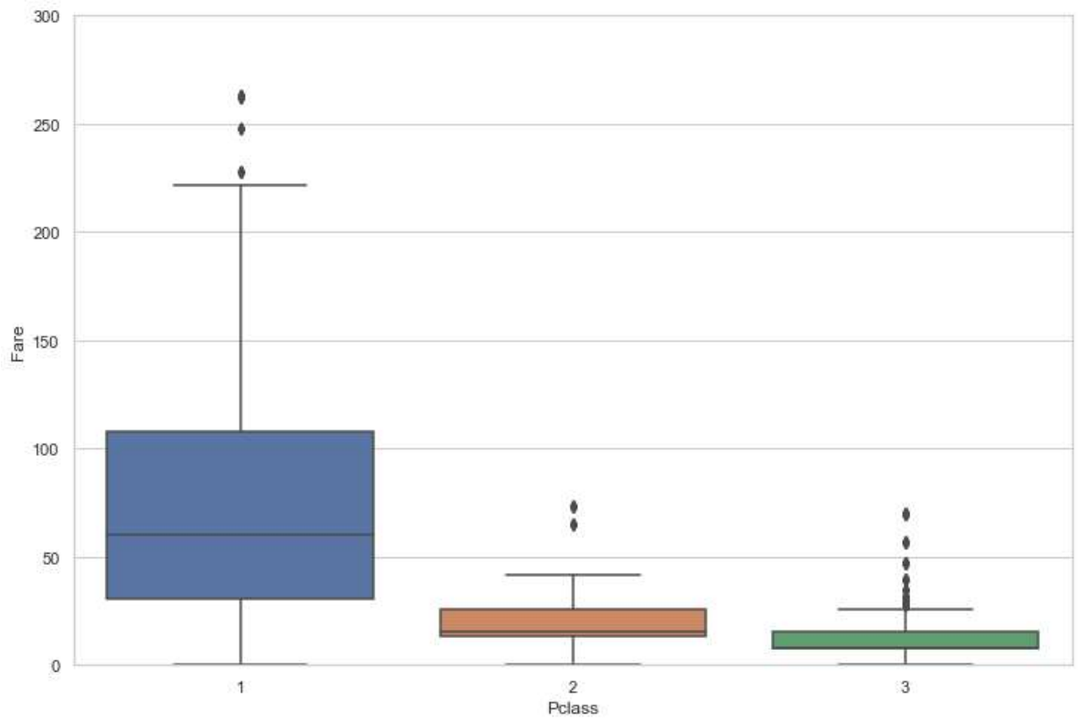


In [122]:
```
graph = sns.FacetGrid(all_data, col="Pclass")
graph.map(plt.hist, "Age", color="Gray")
```

Out[122]: <seaborn.axisgrid.FacetGrid at 0x2ad5700d700>

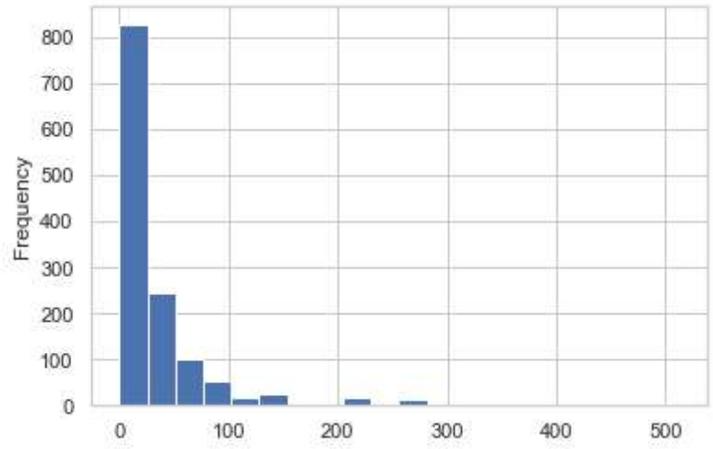


In [123]:
```
plt.figure(figsize=(12,8))
sns.boxplot(data=all_data, x='Pclass', y = 'Fare')
plt.ylim([0, 3_00])
plt.show()
```

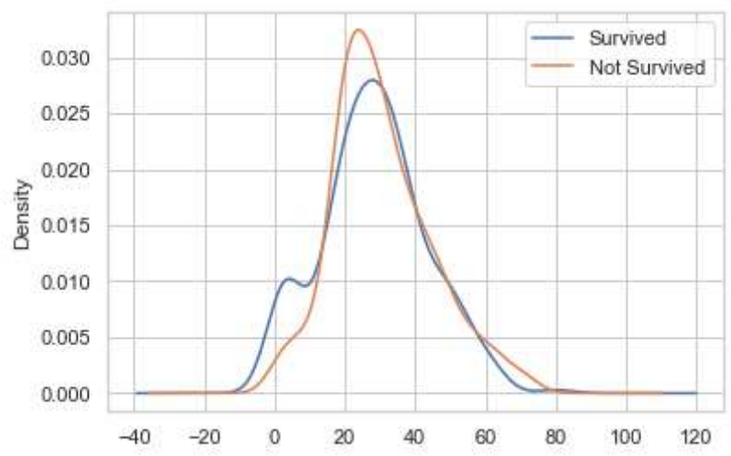


In [124]:
```
all_data['Fare'].plot(kind='hist', bins=20)
```

Out[124]:
```
<AxesSubplot:ylabel='Frequency'>
```

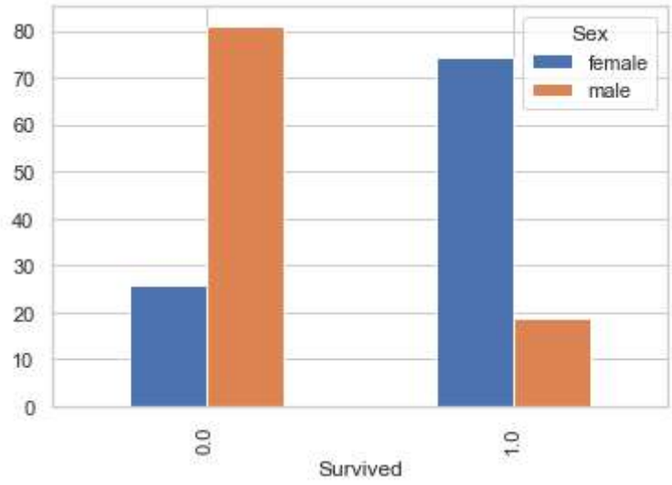


In [125]:
```
all_data[all_data['Survived'] == 1]['Age'].plot(kind='kde', label='Survived')
all_data[all_data['Survived'] == 0]['Age'].plot(kind='kde', label='Not Survive
plt.legend()
plt.show()
```



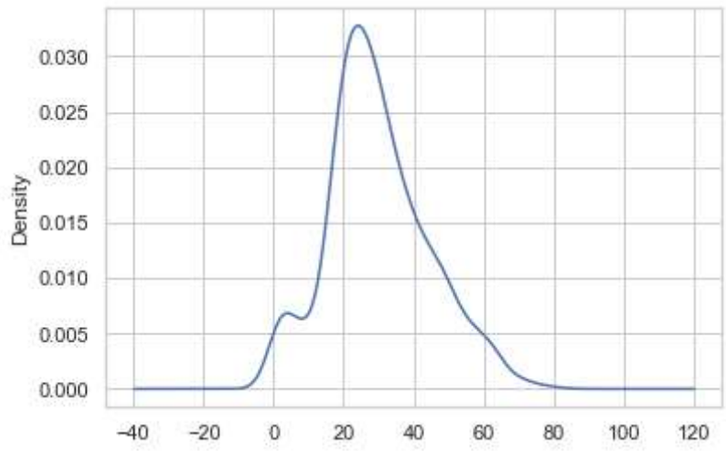
In [126]:
```
a = pd.crosstab(all_data['Survived'], all_data['Sex'], normalize = 'columns')
a.plot(kind='bar')
```

Out[126]: <AxesSubplot:xlabel='Survived'>



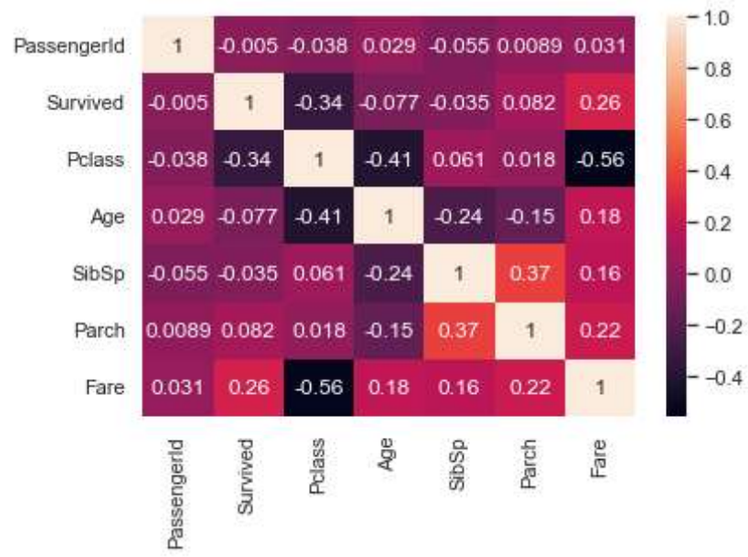
In [127]:
```
all_data['Age'].plot(kind='kde')
```

Out[127]: <AxesSubplot:ylabel='Density'>



In [129]:
```
# Select only numerical variables
X_num = all_data.select_dtypes(include=np.number) # Numerical data
sns.heatmap(X_num.corr(), annot=True)
```

Out[129]: <AxesSubplot:>



Follow for More:

Github: https://www.github.com/ZaidArman (https://www.github.com/ZaidArman)

Linkedin: https://www.linkedin.com/in/zaid-ullah07 (https://www.linkedin.com/in/zaid-ullah07)

Twitter: https://www.twitter.com/ZaidArman (https://www.twitter.com/ZaidArman)_

Instagram: https://www.instagram.com/zaid__arman7 (https://www.instagram.com/zaid__arman7)

Facebook: https://www.facebook.com/profile.php?id=100011010551170 (https://www.facebook.com/profile.php?id=100011010551170)

Made by 💔 Zaid Ullah 💔