Titanic Dataset –

It is one of the most popular datasets used for understanding machine learning basics. It contains information of all the passengers aboard the RMS Titanic, which unfortunately was shipwrecked. This dataset can be used to predict whether a given passenger survived or not.

Features: The titanic dataset has roughly the following types of features:

1.Categorical/Nominal: Variables that can be divided into multiple categories but having no order or priority.

```
Eg. Embarked (C = Cherbourg; Q = Queenstown; S = Southampton)
```

2.Binary: A subtype of categorical features, where the variable has only two categories.

```
Eg: Sex (Male/Female)
```

3.Ordinal: They are similar to categorical features but they have an order(i.e can be sorted).

```
Eg. Pclass (1, 2, 3)
```

4.Continuous: They can take up any value between the minimum and maximum values in a column.

```
Eg. Age, Fare
```

5.Count: They represent the count of a variable.

```
Eg. SibSp, Parch
```

6.Useless: They don’t contribute to the final outcome of an ML model. Here, PassengerId, Name, Cabin and Ticket might fall into this category.

Importing all required libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Read data from dataset

In [2]:

```
df=pd.read_csv("train.csv")
```

Display Data

In [3]:

```
df
```

Out[3]:

	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

Display columns name

In [4]:

```
print(df.columns.values)
```

```
['PassengerId' 'Survived' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch'
 'Ticket' 'Fare' 'Cabin' 'Embarked']
```

Display rows and columns

In [5]:

```
print(df.shape)
```

```
(891, 12)
```

Display top 5 data

In [6]:

```
df.head()
```

Out[6]:

	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

Display bottom 5 data

In [7]:

```
df.tail()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	Na
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	Na
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C1
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	Na

Display information

In [8]:

```
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
```

Summary Statistics

In the train data, there're 891 passengers, and the average survival rate is 38%. Age ranges from 0.42 to 80 and the average is approx 30 year old. At least 50% of passengers don't have siblings / spouses , and at least 75% of passengers don't have parents / children .

In [9]:

```
df.describe()
```

Out[9]:

	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

to check null value

In [10]:

```
df.isnull()
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	I
0	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	True	
...	...	...	...	...	...	...	...	...	...	...	...	
886	False	False	False	False	False	False	False	False	False	False	True	
887	False	False	False	False	False	False	False	False	False	False	False	
888	False	False	False	False	False	True	False	False	False	False	True	
889	False	False	False	False	False	False	False	False	False	False	False	
890	False	False	False	False	False	False	False	False	False	False	True	

891 rows × 12 columns

Display sum of null value

In the train data, there’re 177 in Age, 687 in Cabin, and 2 in Embarked have null values.

In [11]:

```
df.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
```

Covariance

Covariance indicates the direction of the linear relationship between variables.

In [12]:

```
df.cov()
```

Out[12]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Far
PassengerId	66231.000000	-0.626966	-7.561798	138.696504	-16.325843	-0.342697	161.88336
Survived	-0.626966	0.236772	-0.137703	-0.551296	-0.018954	0.032017	6.22178
Pclass	-7.561798	-0.137703	0.699015	-4.496004	0.076599	0.012429	-22.83019
Age	138.696504	-0.551296	-4.496004	211.019125	-4.163334	-2.344191	73.84903
SibSp	-16.325843	-0.018954	0.076599	-4.163334	1.216043	0.368739	8.74873
Parch	-0.342697	0.032017	0.012429	-2.344191	0.368739	0.649728	8.66105
Fare	161.883369	6.221787	-22.830196	73.849030	8.748734	8.661052	2469.43684

Correlated

1.The SibSp and Parch are strong positive correlations. 2.The Pclass and Fare are strong negative corelations.

In [13]:

```
df.corr()
```

Out[13]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

Maximum and Minimum Age

In the train data, there'r Minimum age of passenger is 0.42, And Maximum age is 80.

In [14]:

```
d=df['Age'].min(),df['Age'].max()
d1=pd.DataFrame(d)
d2=d1.rename(columns={0:'Age'})
d2
```

Out[14]:

	Age
0	0.42
1	80.00

Sorting Fare

In [15]:

```
df.sort_values("Fare", ascending = False, inplace = True)
df.head()
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	

Number of people who survived

0 - Not Survived, 1 - Survived

In In the train data, there'r 342 passengers survived, and 549 passengers not survived.

In [16]:

```
df['Survived'].value_counts()
```

Out[16]:

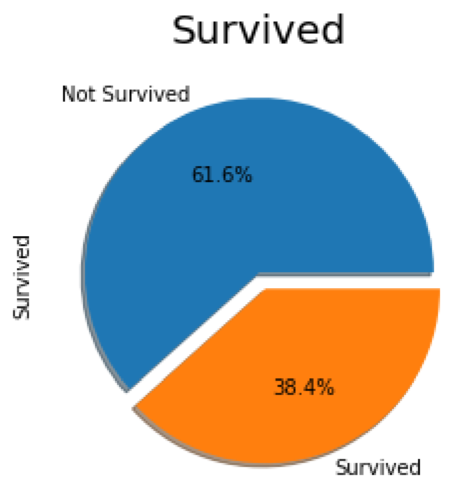
```
0    549
1    342
Name: Survived, dtype: int64
```

Survived in %

In In the train data, 61% of passengers not survived, and 38% of passengers survived. Most of the passengers died in the titanic.

In [17]:

```
plt.title('Survived',fontsize=20)
df['Survived'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True,labels=['Not Survived'
plt.show()
```



Class wise passenger count

In [18]:

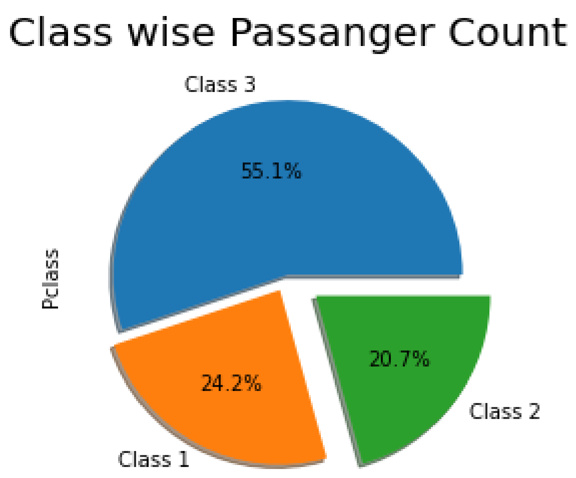
```
c=df['Pclass'].value_counts()
c
```

Out[18]:

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

In [19]:

```
plt.title('Class wise Passanger Count',fontsize=20)
df['Pclass'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True, labels=['Class 3', 'Cla
plt.show()
```



Passanger count (by gender)

In [20]:

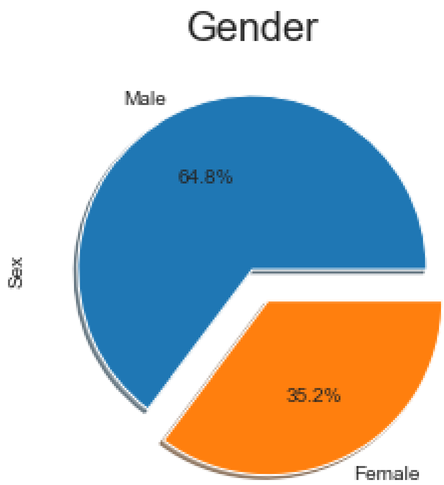
```
d=df['Sex'].value_counts()
d
```

Out[20]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [34]:

```
plt.title('Gender',fontsize=20)
df['Sex'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True, labels=['Male', 'Female'],
plt.show()
```



Pivot Table

In Pclass, here we can see a lot more people survived from the First class than the Second or the Third class, even though the total number of passengers in the First class was less than the Third class.

In [22]:

```
pivot_1=pd.pivot_table(df, index = 'Survived', columns = 'Pclass',values = 'Ticket' ,aggfun
pivot_1
```

Out[22]:

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

Sex: Most of the women survived, and the majority of the male died.

In [23]:

```
pivot_2=pd.pivot_table(df, index = 'Survived', columns = 'Sex', values = 'Ticket' ,aggfunc
pivot_2
```

Out[23]:

Sex	female	male
Survived		
0	81	468
1	233	109

Embarked - Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Embarked: if someone was from “Southampton” had a higher chance of surviving.

In [24]:

```
pivot_3=pd.pivot_table(df, index = 'Survived', columns = 'Embarked',values = 'Ticket' ,aggf
pivot_3
```

Out[24]:

Embarked	C	Q	S
Survived			
0	75	47	427
1	93	30	217

Group by Parent Child

In Train Data, ther'r 678 passangers don't have Parent/Child relation. And there'r 1 Passanger who have maximum Parent/Child relation.

In [25]:

```
df.groupby(["Parch", "Survived"])[["Survived"]].count()
```

Out[25]:

		Survived
Parch	**Survived**	
0	**0**	445
	1	233
1	**0**	53
	1	65
2	**0**	40
	1	40
3	**0**	2
	1	3
4	**0**	4
5	**0**	4
	1	1
6	**0**	1

Group by sibling

In [26]:

```
df.groupby(["SibSp", "Survived"])[["Survived"]].count()
```

Out[26]:

		Survived
SibSp	**Survived**	
0	**0**	398
	1	210
1	**0**	97
	1	112
2	**0**	15
	1	13
3	**0**	12
	1	4
4	**0**	15
	1	3
5	**0**	5
8	**0**	7

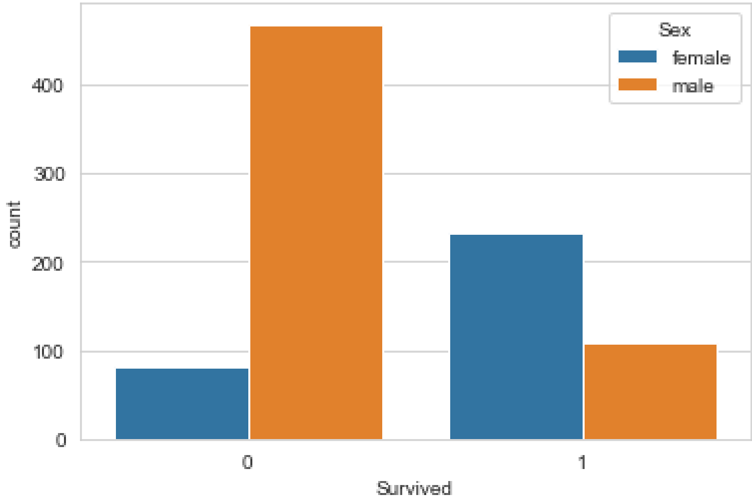
Data Visualization

Countplot of Survived Male and Female

1. Maximum number of Male are not Survived
2. Maximum number of female are Survived

In [27]:

```
sns.set_style("whitegrid")
sns.countplot(x='Survived',data=df,hue='Sex',palette='tab10')
plt.show()
```

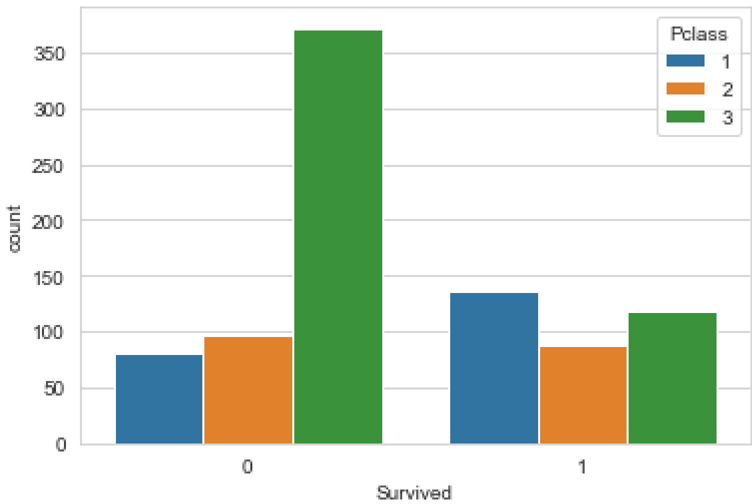


Class wise Survived

1. Most of the Class-3 Passengers not survived

In [28]:

```
sns.set_style("whitegrid")
sns.countplot(x='Survived',data=df,hue='Pclass',palette='tab10')
plt.show()
```

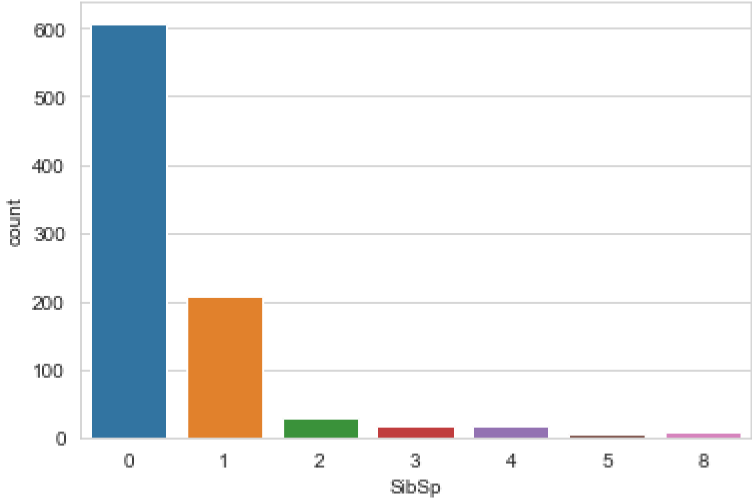


Sibling

Most of the passengers don't have siblings

In [29]:

```
sns.set_style("whitegrid")
sns.countplot(x='SibSp',data=df,palette='tab10')
plt.show()
```

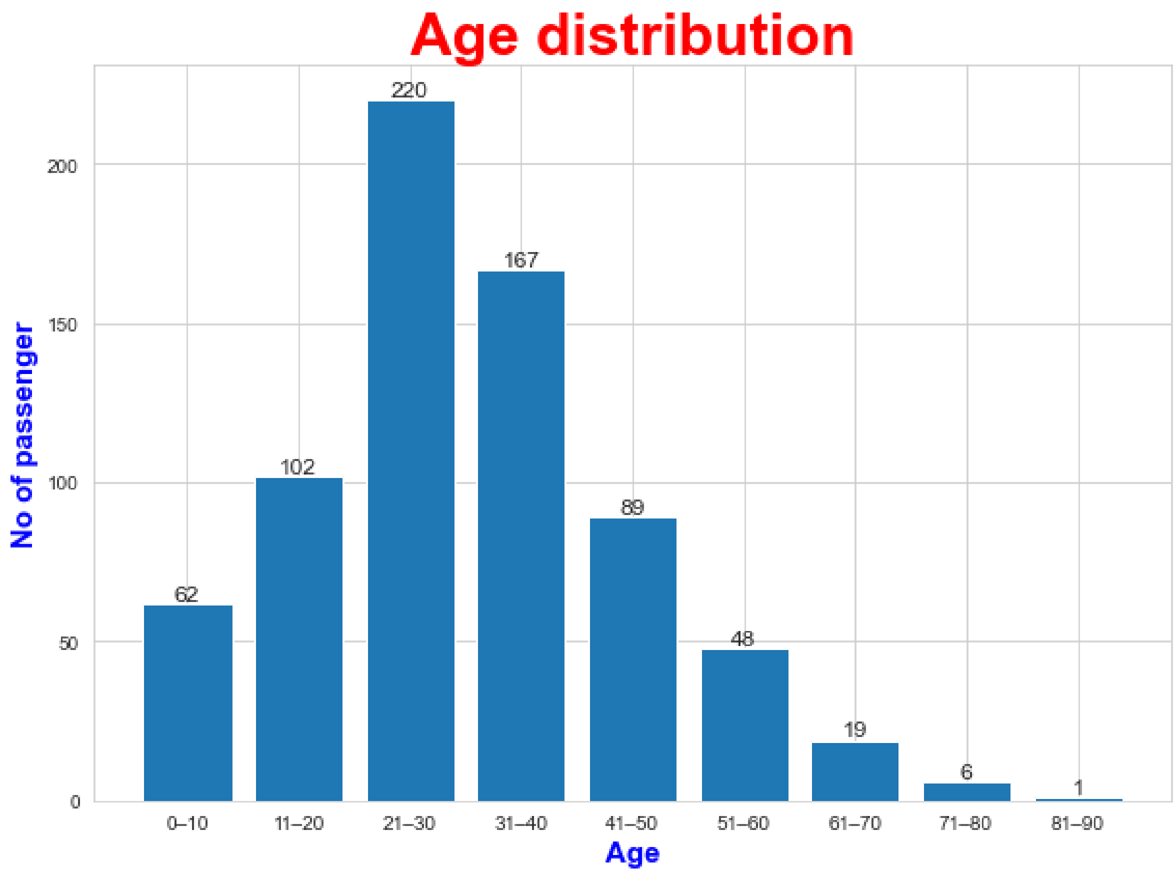


histogram of Age Distribution

In [30]:

```
ages = df[df['Age'].notnull()]['Age'].values
ages_hist = np.histogram(ages, bins=[0,10,20,30,40,50,60,70,80,90])

ages_hist_labels = ['0-10', '11-20', '21-30', '31-40', '41-50', '51-60', '61-70', '71-80',
plt.figure(figsize=(10,7))
plt.title('Age distribution',fontweight='bold', fontsize=30,color='red')
plt.bar(ages_hist_labels, ages_hist[0])
plt.xlabel('Age',fontweight='bold', fontsize=15,color='blue')
plt.ylabel('No of passenger',fontweight='bold', fontsize=15,color='blue')
for i, bin in zip(ages_hist[0], range(9)):
    plt.text(bin, i+3, str(int(i)), fontsize=12,
             horizontalalignment='center', verticalalignment='center')
plt.show()
```



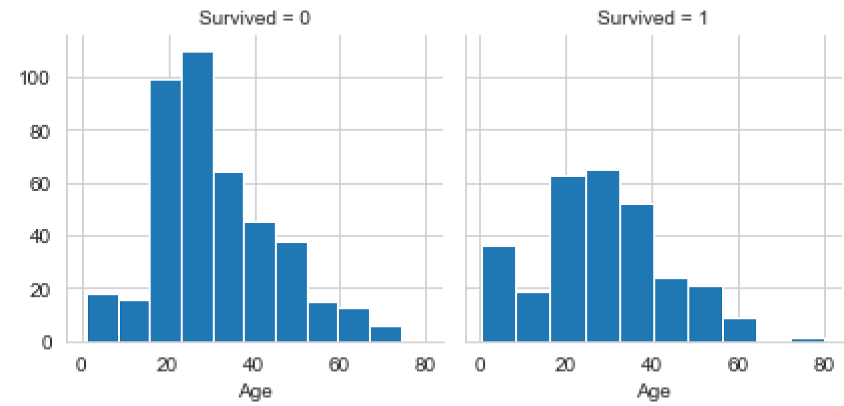
Histogram

Large number of 20-30 year olds did not survive.

In [35]:

```
a1 = sns.FacetGrid(df, col='Survived')
a1.map(plt.hist, 'Age')
plt.show()
```

<Figure size 576x504 with 0 Axes>

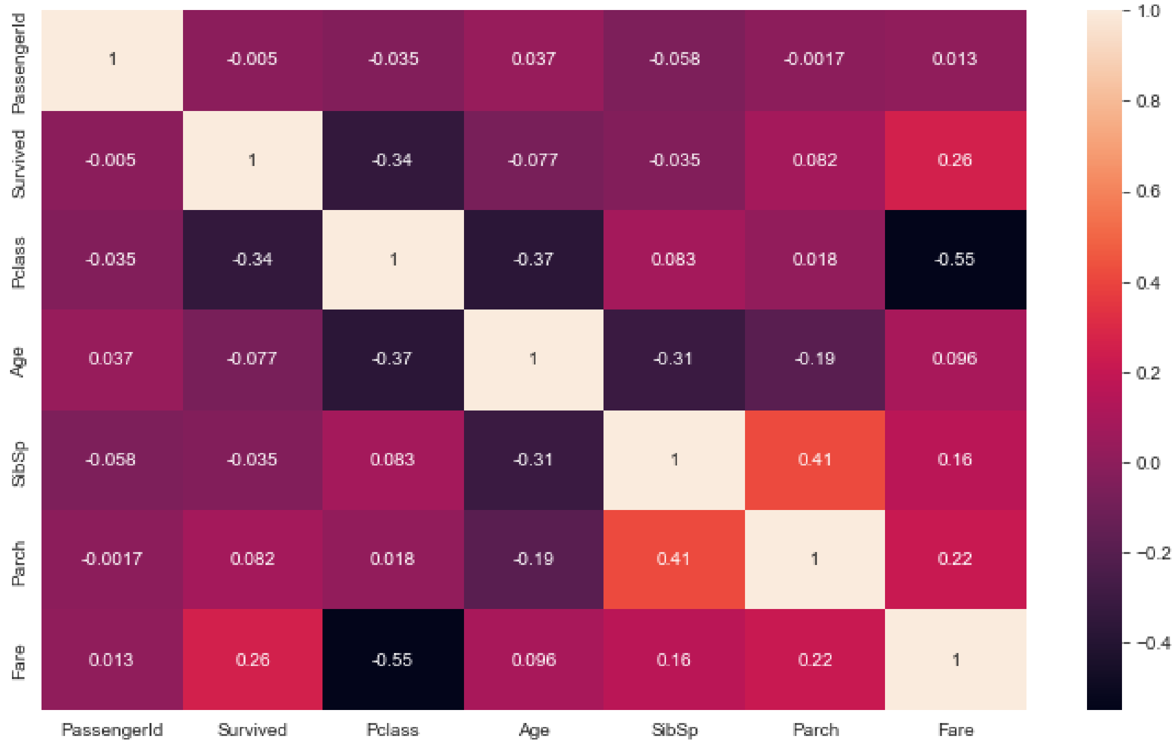


Corrleation Heatmap

1.The strongest positive (Orange) and strongest negative correlations (Black). 2.The SibSp and Parch are strong positive correlations. 3.The Pclass and Fare are strong negative corelations.

In [32]:

```
plt.figure(figsize=(12,7))
sns.heatmap(df.corr(), annot=True)
plt.show()
```



Conclusion:

In In the train data, 61% of passengers not survived, and only 38% of passengers survived. Most of the passengers died in the titanic.(i.e. Maximum number of Male are not Survived.)

In [ ]:

```

```