

```
In [1]: import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: data=pd.read_csv("/home/placement/Desktop/reddy/Titanic Dataset.csv")
```

```
In [3]: data.describe()
```

Out[3]:

	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 [4]: data.head(10)
```

```
Out[4]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

```
In [5]: data.tail(100)
```

```
Out[5]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
791	792	0	2	Gaskell, Mr. Alfred	male	16.0	0	0	239865	26.0000	NaN	S
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.5500	NaN	S
793	794	0	1	Hoyt, Mr. William Fisher	male	NaN	0	0	PC 17600	30.6958	NaN	C
794	795	0	3	Dantcheff, Mr. Ristiu	male	25.0	0	0	349203	7.8958	NaN	S
795	796	0	2	Otter, Mr. Richard	male	39.0	0	0	28213	13.0000	NaN	S
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

100 rows × 12 columns

In [6]: data

Out[6]:

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

891 rows × 12 columns

In [7]: 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 [8]: data.isna().sum()

```
Out[8]: 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 [9]: data['Pclass'].unique()
```

```
Out[9]: array([3, 1, 2])
```

```
In [10]: data['Survived'].unique()
```

```
Out[10]: array([0, 1])
```

```
In [11]: data['SibSp'].unique()
```

```
Out[11]: array([1, 0, 3, 4, 2, 5, 8])
```

```
In [12]: data['Age'].unique()
```

```
Out[12]: 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 [13]: data['Fare'].unique()
```

```
Out[13]: array([ 7.25 , 71.2833, 7.925 , 53.1 , 8.05 , 8.4583,  
 51.8625, 21.075 , 11.1333, 30.0708, 16.7 , 26.55 ,  
 31.275 , 7.8542, 16. , 29.125 , 13. , 18. ,  
 7.225 , 26. , 8.0292, 35.5 , 31.3875, 263. ,  
 7.8792, 7.8958, 27.7208, 146.5208, 7.75 , 10.5 ,  
 82.1708, 52. , 7.2292, 11.2417, 9.475 , 21. ,  
 41.5792, 15.5 , 21.6792, 17.8 , 39.6875, 7.8 ,  
 76.7292, 61.9792, 27.75 , 46.9 , 80. , 83.475 ,  
 27.9 , 15.2458, 8.1583, 8.6625, 73.5 , 14.4542,  
 56.4958, 7.65 , 29. , 12.475 , 9. , 9.5 ,  
 7.7875, 47.1 , 15.85 , 34.375 , 61.175 , 20.575 ,  
 34.6542, 63.3583, 23. , 77.2875, 8.6542, 7.775 ,  
 24.15 , 9.825 , 14.4583, 247.5208, 7.1417, 22.3583,  
 6.975 , 7.05 , 14.5 , 15.0458, 26.2833, 9.2167,  
 79.2 , 6.75 , 11.5 , 36.75 , 7.7958, 12.525 ,  
 66.6 , 7.3125, 61.3792, 7.7333, 69.55 , 16.1 ,  
 15.75 , 20.525 , 55. , 25.925 , 33.5 , 30.6958,  
 25.4667, 28.7125, 0. , 15.05 , 39. , 22.025 ,  
 50. , 8.4042, 6.4958, 10.4625, 18.7875, 31. ,  
 113.275 , 27. , 76.2917, 90. , 9.35 , 13.5 ,  
 7.55 , 26.25 , 12.275 , 7.125 , 52.5542, 20.2125,  
 86.5 , 512.3292, 79.65 , 153.4625, 135.6333, 19.5 ,  
 29.7 , 77.9583, 20.25 , 78.85 , 91.0792, 12.875 ,  
 8.85 , 151.55 , 30.5 , 23.25 , 12.35 , 110.8833,  
 108.9 , 24. , 56.9292, 83.1583, 262.375 , 14. ,  
 164.8667, 134.5 , 6.2375, 57.9792, 28.5 , 133.65 ,  
 15.9 , 9.225 , 35. , 75.25 , 69.3 , 55.4417,  
 211.5 , 4.0125, 227.525 , 15.7417, 7.7292, 12. ,  
 120. , 12.65 , 18.75 , 6.8583, 32.5 , 7.875 ,  
 14.4 , 55.9 , 8.1125, 81.8583, 19.2583, 19.9667,  
 89.1042, 38.5 , 7.725 , 13.7917, 9.8375, 7.0458,  
 7.5208, 12.2875, 9.5875, 49.5042, 78.2667, 15.1 ,  
 7.6292, 22.525 , 26.2875, 59.4 , 7.4958, 34.0208,  
 93.5 , 221.7792, 106.425 , 49.5 , 71. , 13.8625,  
 7.8292, 39.6 , 17.4 , 51.4792, 26.3875, 30. ,  
 40.125 , 8.7125, 15. , 33. , 42.4 , 15.55 ,  
 65. , 32.3208, 7.0542, 8.4333, 25.5875, 9.8417,  
 8.1375, 10.1708, 211.3375, 57. , 13.4167, 7.7417,
```

```
9.4833, 7.7375, 8.3625, 23.45 , 25.9292, 8.6833,  
8.5167, 7.8875, 37.0042, 6.45 , 6.95 , 8.3 ,  
6.4375, 39.4 , 14.1083, 13.8583, 50.4958, 5. ,  
9.8458. 10.51671)
```

```
In [14]: data1=data.drop(['PassengerId', 'Name', 'Ticket', 'Cabin', 'SibSp', 'Parch'],axis=1)
```

```
In [15]: data1
```

```
Out[15]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	male	22.0	7.2500	S
1	1	1	female	38.0	71.2833	C
2	1	3	female	26.0	7.9250	S
3	1	1	female	35.0	53.1000	S
4	0	3	male	35.0	8.0500	S
...
886	0	2	male	27.0	13.0000	S
887	1	1	female	19.0	30.0000	S
888	0	3	female	NaN	23.4500	S
889	1	1	male	26.0	30.0000	C
890	0	3	male	32.0	7.7500	Q

891 rows × 6 columns


```
In [16]: data1.isna().sum()
```

```
Out[16]: Survived      0  
Pclass      0  
Sex          0  
Age        177  
Fare        0  
Embarked     2  
dtype: int64
```

```
In [17]: data1.shape
```

```
Out[17]: (891, 6)
```

```
In [18]: data1['Sex']=data1['Sex'].map({'male':1,'female':0})
```

```
In [19]: data1
```

```
Out[19]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	0	38.0	71.2833	C
2	1	3	0	26.0	7.9250	S
3	1	1	0	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...
886	0	2	1	27.0	13.0000	S
887	1	1	0	19.0	30.0000	S
888	0	3	0	NaN	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

```
In [20]: data1['Age'].unique()  
data1.fillna(35, inplace=True)
```

```
In [21]: data1.isna().sum()
```

```
Out[21]: Survived    0  
Pclass      0  
Sex         0  
Age         0  
Fare        0  
Embarked    0  
dtype: int64
```

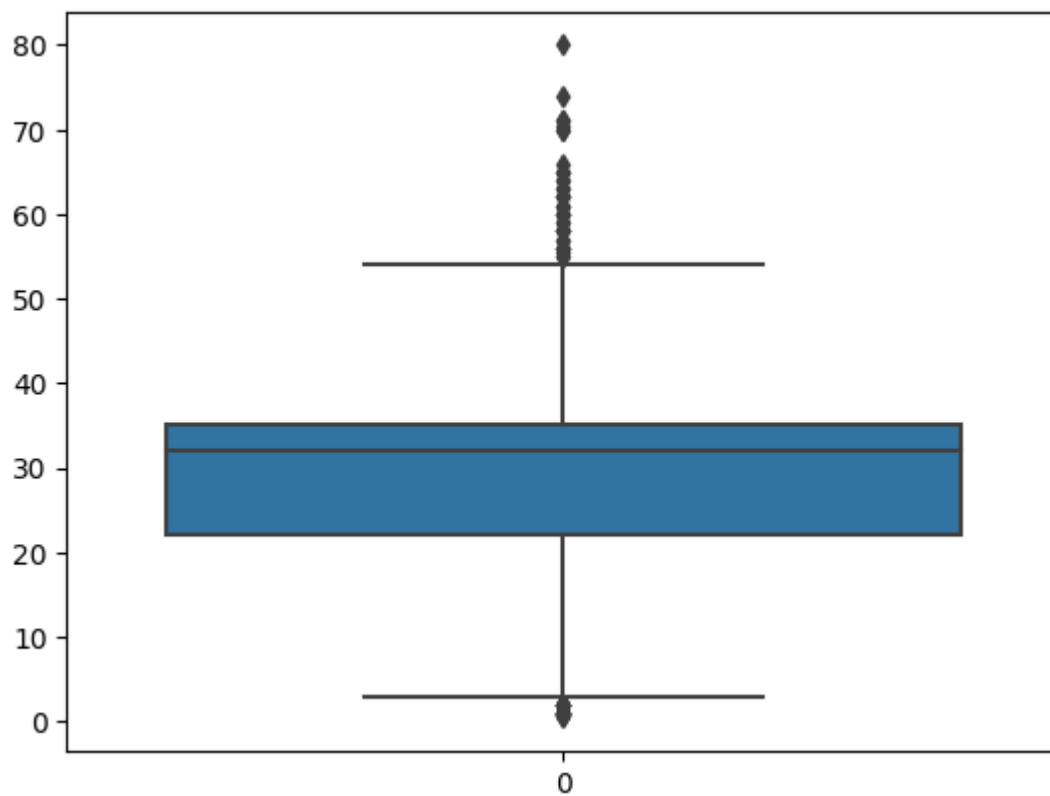
In [22]: data1.describe()

Out[22]:

	Survived	Pclass	Sex	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	0.647587	30.752155	32.204208
std	0.486592	0.836071	0.477990	13.173100	49.693429
min	0.000000	1.000000	0.000000	0.420000	0.000000
25%	0.000000	2.000000	0.000000	22.000000	7.910400
50%	0.000000	3.000000	1.000000	32.000000	14.454200
75%	1.000000	3.000000	1.000000	35.000000	31.000000
max	1.000000	3.000000	1.000000	80.000000	512.329200

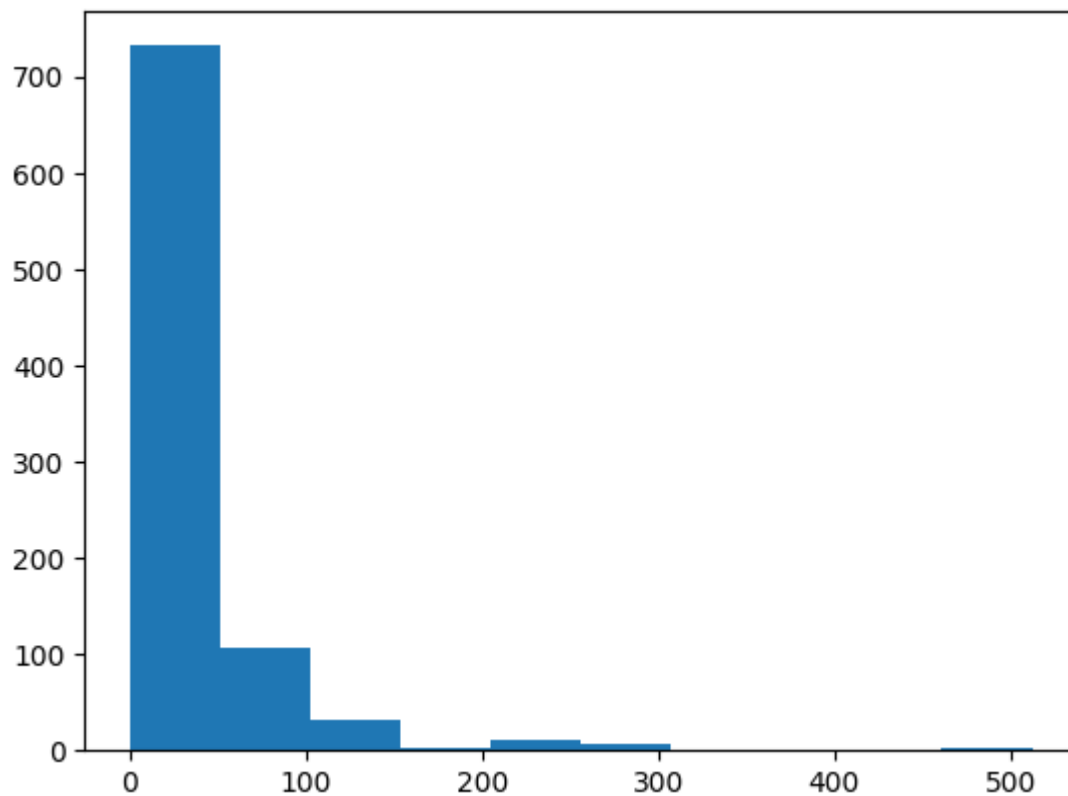
```
In [23]: import seaborn as sns  
import matplotlib.pyplot as mp  
sns.boxplot(data1.Age)
```

Out[23]: <Axes: >



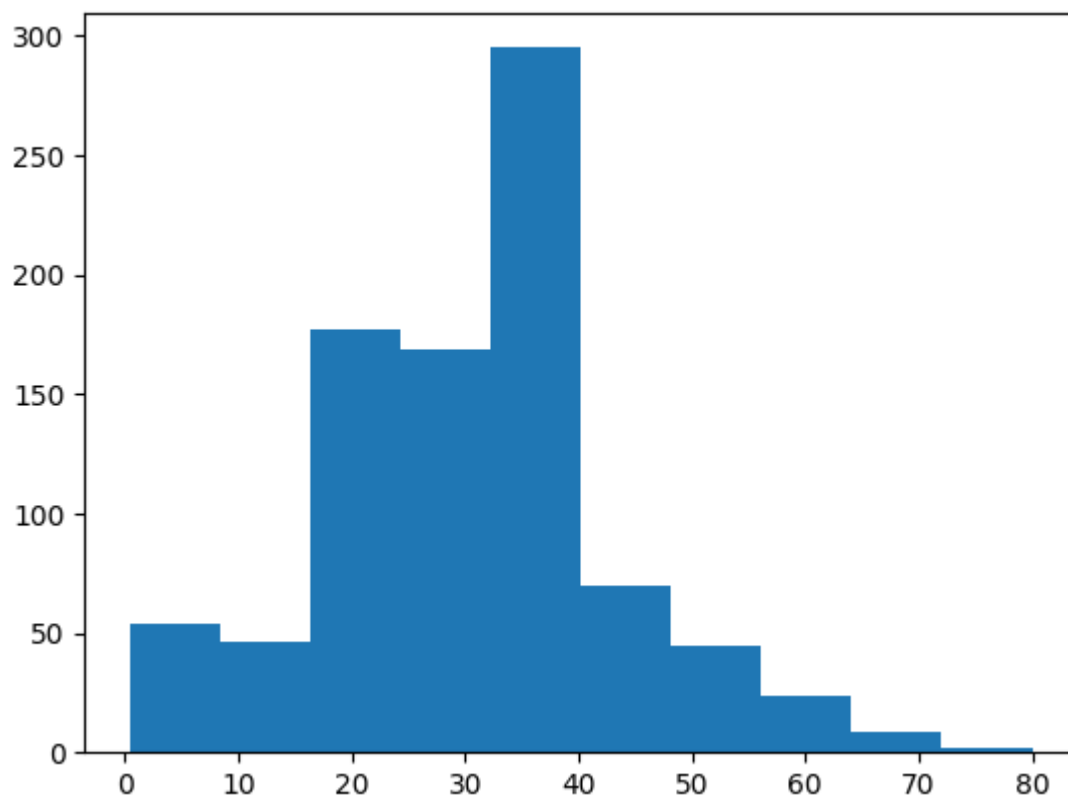
```
In [24]: mp.hist(data1['Fare'])
```

```
Out[24]: (array([732., 106., 31., 2., 11., 6., 0., 0., 0., 3.]),  
          array([ 0., 51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,  
                307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),  
          <BarContainer object of 10 artists>)
```



```
In [25]: mp.hist(data1['Age'])
```

```
Out[25]: (array([ 54.,  46., 177., 169., 295.,  70.,  45.,  24.,   9.,   2.]),  
          array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,  
                64.084, 72.042, 80.   ]),  
          <BarContainer object of 10 artists>)
```



```
In [26]: data1['Age'].unique()
```

```
Out[26]: array([22. , 38. , 26. , 35. , 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 [27]: data1.groupby(['Age']).count()
```

```
Out[27]:
```

	Survived	Pclass	Sex	Fare	Embarked
Age					
0.42	1	1	1	1	1
0.67	1	1	1	1	1
0.75	2	2	2	2	2
0.83	2	2	2	2	2
0.92	1	1	1	1	1
...
70.00	2	2	2	2	2
70.50	1	1	1	1	1
71.00	2	2	2	2	2
74.00	1	1	1	1	1
80.00	1	1	1	1	1

88 rows × 5 columns

```
In [28]: data1['Pclass']=data1['Pclass'].map({1:'F',2:'S',3:'Third'})
```

```
In [29]: data1
```

```
Out[29]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	Third	1	22.0	7.2500	S
1	1	F	0	38.0	71.2833	C
2	1	Third	0	26.0	7.9250	S
3	1	F	0	35.0	53.1000	S
4	0	Third	1	35.0	8.0500	S
...
886	0	S	1	27.0	13.0000	S
887	1	F	0	19.0	30.0000	S
888	0	Third	0	35.0	23.4500	S
889	1	F	1	26.0	30.0000	C
890	0	Third	1	32.0	7.7500	Q

891 rows × 6 columns


```
In [30]: data1=pd.get_dummies(data1,dtype=int)
data1
```

Out[30]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	0	1	22.0	7.2500	0	0	1	0	0	0	1
1	1	0	38.0	71.2833	1	0	0	0	1	0	0
2	1	0	26.0	7.9250	0	0	1	0	0	0	1
3	1	0	35.0	53.1000	1	0	0	0	0	0	1
4	0	1	35.0	8.0500	0	0	1	0	0	0	1
...
886	0	1	27.0	13.0000	0	1	0	0	0	0	1
887	1	0	19.0	30.0000	1	0	0	0	0	0	1
888	0	0	35.0	23.4500	0	0	1	0	0	0	1
889	1	1	26.0	30.0000	1	0	0	0	1	0	0
890	0	1	32.0	7.7500	0	0	1	0	0	1	0

891 rows × 11 columns

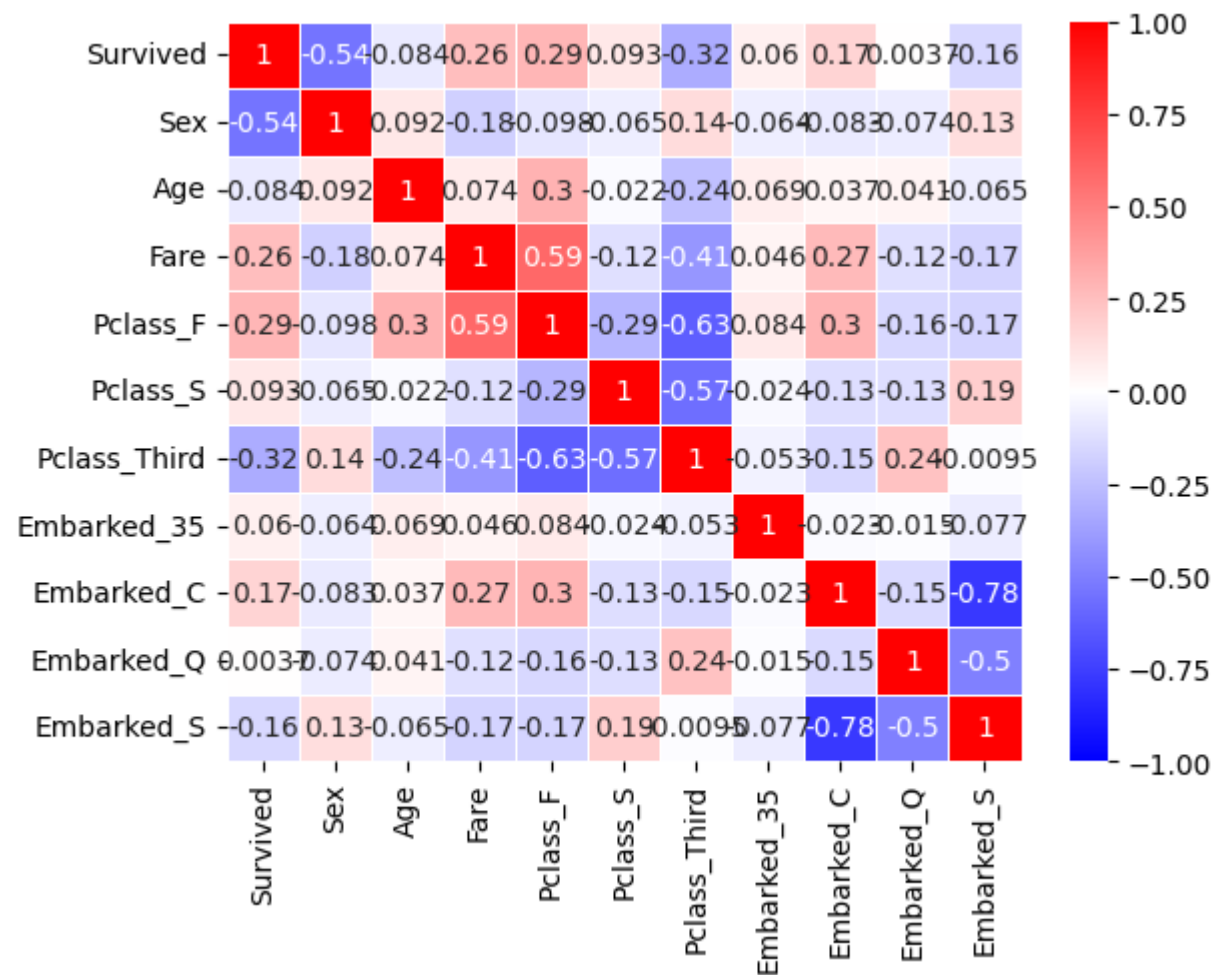
```
In [31]: cor=data1.corr()  
cor
```

Out[31]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	-0.543351	-0.083713	0.257307	0.285904	0.093349	-0.322308	0.060095	0.168240	0.003650	-0.155660
Sex	-0.543351	1.000000	0.091930	-0.182333	-0.098013	-0.064746	0.137143	-0.064296	-0.082853	-0.074115	0.125722
Age	-0.083713	0.091930	1.000000	0.074199	0.302149	-0.022021	-0.242412	0.069343	0.036953	0.040528	-0.065062
Fare	0.257307	-0.182333	0.074199	1.000000	0.591711	-0.118557	-0.413333	0.045646	0.269335	-0.117216	-0.166603
Pclass_F	0.285904	-0.098013	0.302149	0.591711	1.000000	-0.288585	-0.626738	0.083847	0.296423	-0.155342	-0.170379
Pclass_S	0.093349	-0.064746	-0.022021	-0.118557	-0.288585	1.000000	-0.565210	-0.024197	-0.125416	-0.127301	0.192061
Pclass_Third	-0.322308	0.137143	-0.242412	-0.413333	-0.626738	-0.565210	1.000000	-0.052550	-0.153329	0.237449	-0.009511
Embarked_35	0.060095	-0.064296	0.069343	0.045646	0.083847	-0.024197	-0.052550	1.000000	-0.022864	-0.014588	-0.076588
Embarked_C	0.168240	-0.082853	0.036953	0.269335	0.296423	-0.125416	-0.153329	-0.022864	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	-0.074115	0.040528	-0.117216	-0.155342	-0.127301	0.237449	-0.014588	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	0.125722	-0.065062	-0.166603	-0.170379	0.192061	-0.009511	-0.076588	-0.778359	-0.496624	1.000000

```
In [32]: import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr')
```

Out[32]: <Axes: >



```
In [33]: data1.groupby(['Survived']).count()
```

```
Out[33]:
```

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
Survived										
0	549	549	549	549	549	549	549	549	549	549
1	342	342	342	342	342	342	342	342	342	342

```
In [34]: y=data1['Survived']  
x=data1.drop('Survived',axis=1)
```

```
In [35]: x
```

```
Out[35]:
```

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	1	22.0	7.2500	0	0	1	0	0	0	1
1	0	38.0	71.2833	1	0	0	0	1	0	0
2	0	26.0	7.9250	0	0	1	0	0	0	1
3	0	35.0	53.1000	1	0	0	0	0	0	1
4	1	35.0	8.0500	0	0	1	0	0	0	1
...
886	1	27.0	13.0000	0	1	0	0	0	0	1
887	0	19.0	30.0000	1	0	0	0	0	0	1
888	0	35.0	23.4500	0	0	1	0	0	0	1
889	1	26.0	30.0000	1	0	0	0	1	0	0
890	1	32.0	7.7500	0	0	1	0	0	1	0

891 rows × 10 columns

In []:

In [36]:

y

Out[36]:

```
0      0
1      1
2      1
3      1
4      0
..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

In [37]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [38]:

```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[38]:

```
▼ LogisticRegression
LogisticRegression()
```

In [39]:

```
y_pred=classifier.predict(x_test)
```

In [40]: `y_pred`

Out[40]: `array([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, 0, 0, 0, 1, 0, 0, 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, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
0, 0, 0, 1, 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, 0, 0, 1, 1, 0, 0, 0, 1, 1, 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, 1, 1, 0, 1, 0, 0,
0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
1, 0, 0, 0, 0, 0, 1, 1, 0])`

In [41]: `from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)`

Out[41]: `array([[155, 20],
[37, 83]])`

In [42]: `from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)`

Out[42]: `0.8067796610169492`