

```
In [122]: import pandas as pd
```

```
In [123]: data=pd.read_csv("/home/placement/Desktop/python/Titanic Dataset.csv")
```

```
In [124]: data.describe()
```

Out[124]:

	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 [125]: 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 [126]: list(data)

```
Out[126]: ['PassengerId',
'Survived',
'Pclass',
'Name',
'Sex',
'Age',
'SibSp',
'Parch',
'Ticket',
'Fare',
'Cabin',
'Embarked']
```

In [127]: `data.head(10)`

Out[127]:

	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 [128]: data.isna().sum()
```

```
Out[128]: 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 [129]: data.Pclass.unique()
```

```
Out[129]: array([3, 1, 2])
```

```
In [130]: data.SibSp.unique()
```

```
Out[130]: array([1, 0, 3, 4, 2, 5, 8])
```

```
In [131]: data.Survived.unique()
```

```
Out[131]: array([0, 1])
```

```
In [132]: data.Age.unique()
```

```
Out[132]: 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 [133]: data1=data.drop(['PassengerId', 'Name', 'Ticket', 'Fare', 'Cabin', 'Embarked'],axis=1)
data1
```

```
Out[133]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	male	22.0	1	0
1	1	1	female	38.0	1	0
2	1	3	female	26.0	0	0
3	1	1	female	35.0	1	0
4	0	3	male	35.0	0	0
...
886	0	2	male	27.0	0	0
887	1	1	female	19.0	0	0
888	0	3	female	NaN	1	2
889	1	1	male	26.0	0	0
890	0	3	male	32.0	0	0

891 rows × 6 columns

```
In [134]: data1['Sex']=data['Sex'].map({'male':1,'female':0})
data1
```

Out[134]:

	Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	1	22.0	1	0
1	1	1	0	38.0	1	0
2	1	3	0	26.0	0	0
3	1	1	0	35.0	1	0
4	0	3	1	35.0	0	0
...
886	0	2	1	27.0	0	0
887	1	1	0	19.0	0	0
888	0	3	0	NaN	1	2
889	1	1	1	26.0	0	0
890	0	3	1	32.0	0	0

891 rows × 6 columns

```
In [135]: data2=data1.fillna(data.median())
```

/tmp/ipykernel_5123/1290514040.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
data2=data1.fillna(data.median())
```

```
In [136]: data2
```

```
Out[136]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	1	22.0	1	0
1	1	1	0	38.0	1	0
2	1	3	0	26.0	0	0
3	1	1	0	35.0	1	0
4	0	3	1	35.0	0	0
...
886	0	2	1	27.0	0	0
887	1	1	0	19.0	0	0
888	0	3	0	28.0	1	2
889	1	1	1	26.0	0	0
890	0	3	1	32.0	0	0

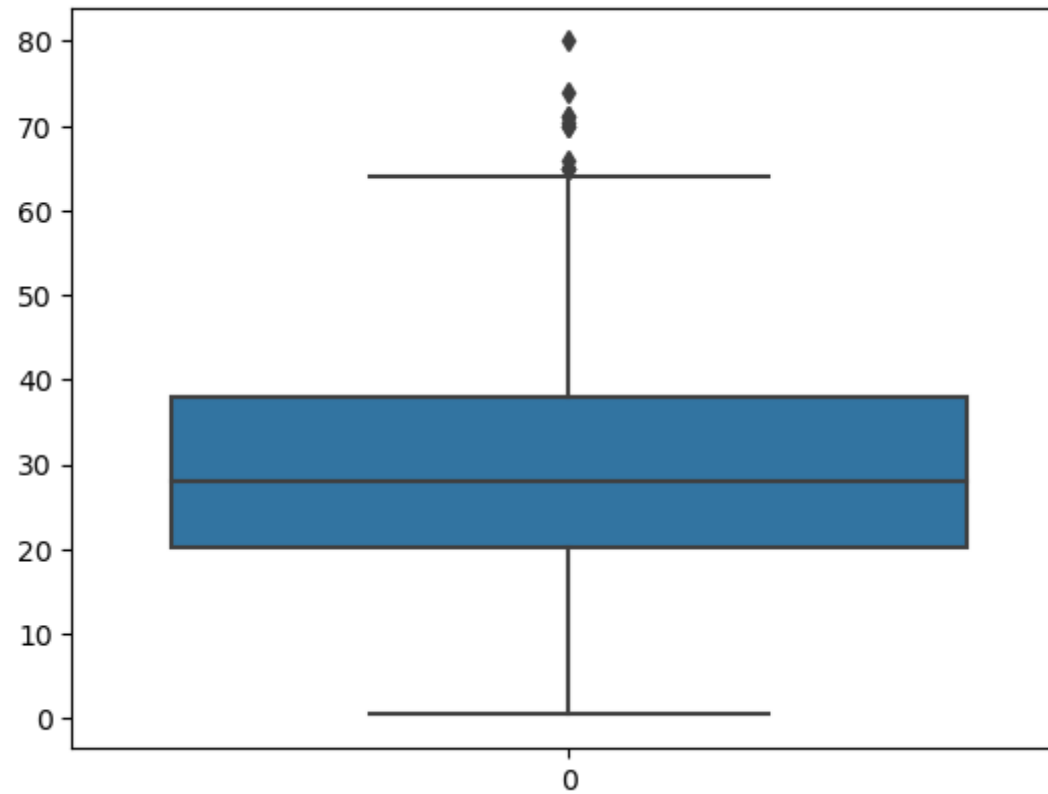
891 rows × 6 columns

```
In [137]: data2.isna().sum()
```

```
Out[137]: Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
dtype: int64
```

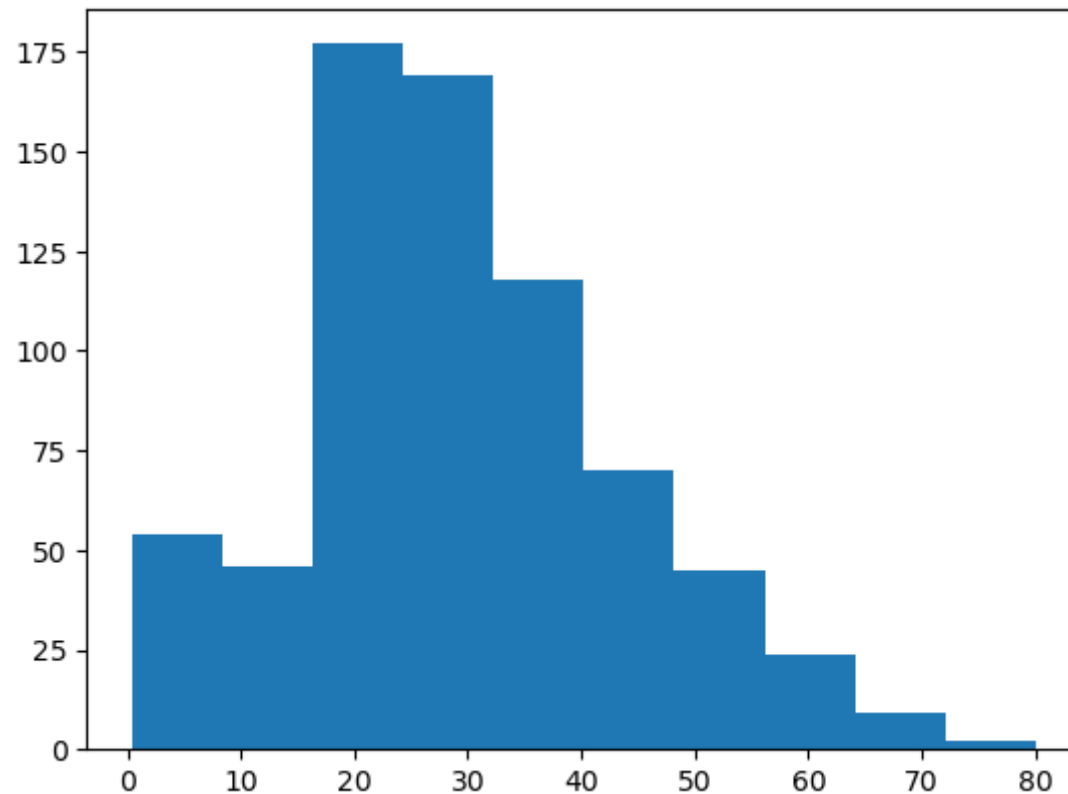
```
In [138]: import seaborn as sns  
import matplotlib.pyplot as plt  
sns.boxplot(data.Age)
```

Out[138]: <Axes: >



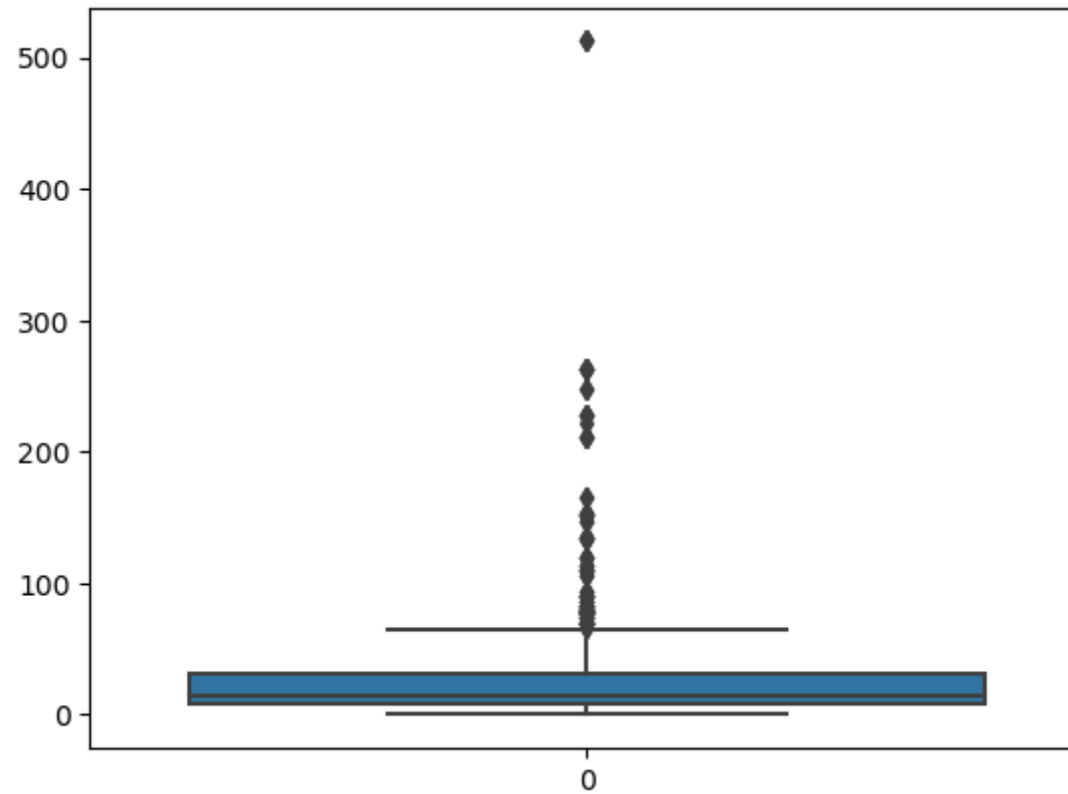

```
In [139]: plt.hist(data['Age'])
```

```
Out[139]: (array([ 54.,  46., 177., 169., 118.,  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>)
```



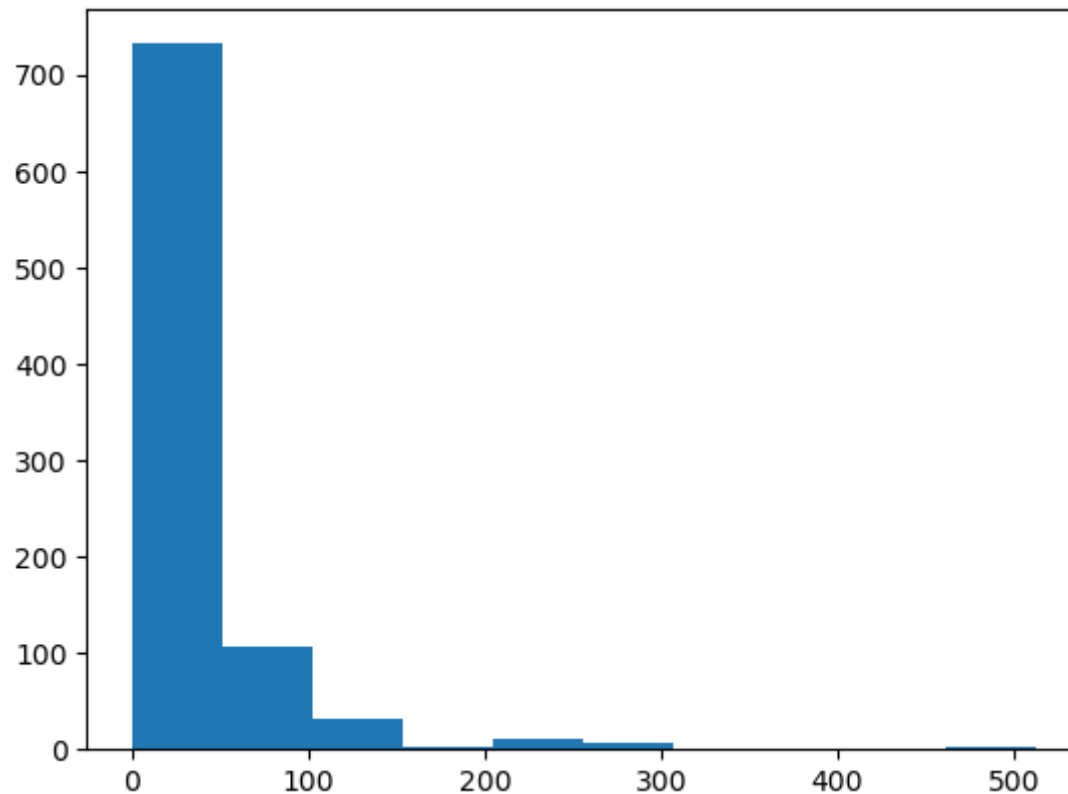
```
In [140]: import seaborn as sns  
import matplotlib.pyplot as plt  
sns.boxplot(data.Fare)
```

Out[140]: <Axes: >



```
In [141]: plt.hist(data['Fare'])
```

```
Out[141]: (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 [142]: data2.isna().sum()
```

```
Out[142]: Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
dtype: int64
```

```
In [143]: data2.describe()
```

```
Out[143]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	0.647587	29.361582	0.523008	0.381594
std	0.486592	0.836071	0.477990	13.019697	1.102743	0.806057
min	0.000000	1.000000	0.000000	0.420000	0.000000	0.000000
25%	0.000000	2.000000	0.000000	22.000000	0.000000	0.000000
50%	0.000000	3.000000	1.000000	28.000000	0.000000	0.000000
75%	1.000000	3.000000	1.000000	35.000000	1.000000	0.000000
max	1.000000	3.000000	1.000000	80.000000	8.000000	6.000000

```
In [144]: data2["Age"].unique()
```

```
Out[144]: array([22. , 38. , 26. , 35. , 28. , 54. , 2. , 27. , 14. ,
        4. , 58. , 20. , 39. , 55. , 31. , 34. , 15. , 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 [145]: data.groupby(["Age"]).count()
```

```
Out[145]:
```

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Age											
0.42	1	1	1	1	1	1	1	1	1	0	1
0.67	1	1	1	1	1	1	1	1	1	0	1
0.75	2	2	2	2	2	2	2	2	2	0	2
0.83	2	2	2	2	2	2	2	2	2	0	2
0.92	1	1	1	1	1	1	1	1	1	1	1
...
70.00	2	2	2	2	2	2	2	2	2	1	2
70.50	1	1	1	1	1	1	1	1	1	0	1
71.00	2	2	2	2	2	2	2	2	2	1	2
74.00	1	1	1	1	1	1	1	1	1	0	1
80.00	1	1	1	1	1	1	1	1	1	1	1

88 rows × 11 columns

```
In [146]: data['Pclass']=data['Pclass'].map({1:'F',2:'S',3:'Third'})
data
```

Out[146]:

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

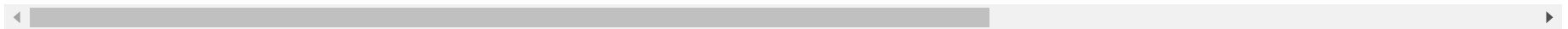
891 rows × 12 columns

```
In [147]: data1=pd.get_dummies(data)
data1
```

Out[147]:

	PassengerId	Survived	Age	SibSp	Parch	Fare	Pclass_F	Pclass_S	Pclass_Third	Name_Abbing, Mr. Anthony	...	Cabin_F G73	Cabin_F2	Cabin_F33
0	1	0	22.0	1	0	7.2500	0	0	1	0 ...		0	0	0
1	2	1	38.0	1	0	71.2833	1	0	0	0 ...		0	0	0
2	3	1	26.0	0	0	7.9250	0	0	1	0 ...		0	0	0
3	4	1	35.0	1	0	53.1000	1	0	0	0 ...		0	0	0
4	5	0	35.0	0	0	8.0500	0	0	1	0 ...		0	0	0
...
886	887	0	27.0	0	0	13.0000	0	1	0	0 ...		0	0	0
887	888	1	19.0	0	0	30.0000	1	0	0	0 ...		0	0	0
888	889	0	NaN	1	2	23.4500	0	0	1	0 ...		0	0	0
889	890	1	26.0	0	0	30.0000	1	0	0	0 ...		0	0	0
890	891	0	32.0	0	0	7.7500	0	0	1	0 ...		0	0	0

891 rows × 1733 columns



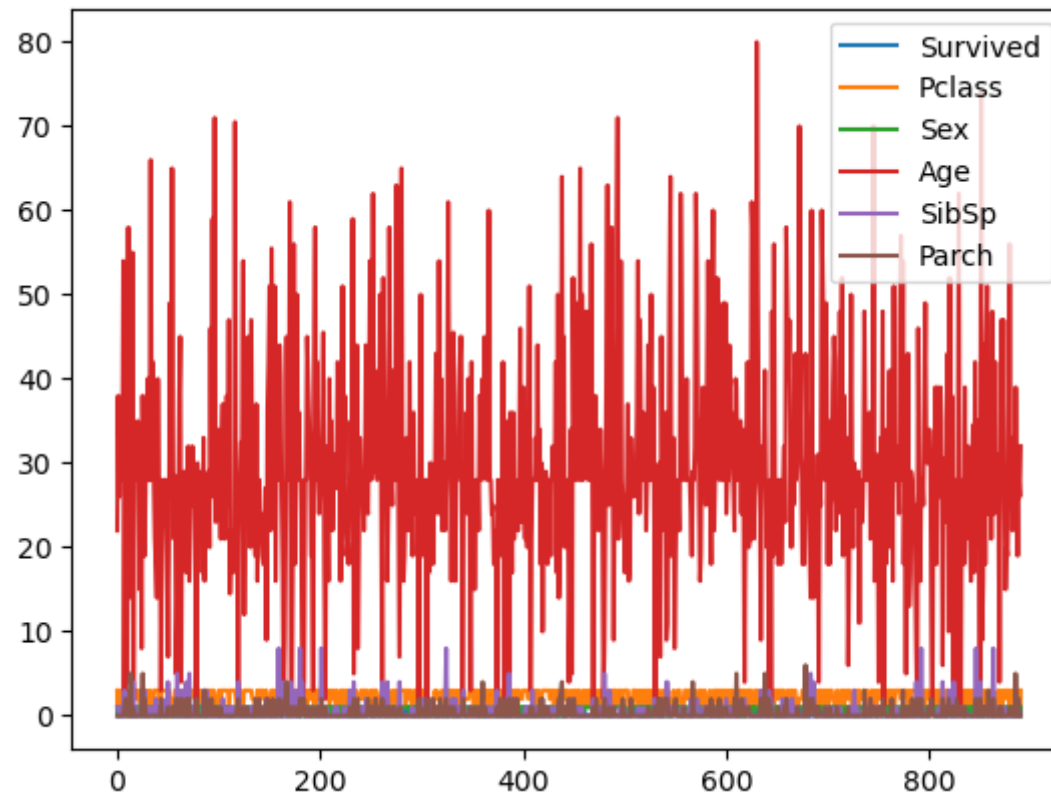
```
In [148]: cor=data2.corr()  
cor
```

Out[148]:

	Survived	Pclass	Sex	Age	SibSp	Parch
Survived	1.000000	-0.338481	-0.543351	-0.064910	-0.035322	0.081629
Pclass	-0.338481	1.000000	0.131900	-0.339898	0.083081	0.018443
Sex	-0.543351	0.131900	1.000000	0.081163	-0.114631	-0.245489
Age	-0.064910	-0.339898	0.081163	1.000000	-0.233296	-0.172482
SibSp	-0.035322	0.083081	-0.114631	-0.233296	1.000000	0.414838
Parch	0.081629	0.018443	-0.245489	-0.172482	0.414838	1.000000


```
In [149]: data2.plot()
```

```
Out[149]: <Axes: >
```



```
In [150]: import seaborn as sns
sns.heatmap(cor, vmax=1, vmin=-1, annot=True, linewidths=5, cmap='bwr')
```

Out[150]: <Axes: >



```
In [151]: data2.groupby(["Survived"]).count()
```

```
Out[151]:
```

	Pclass	Sex	Age	SibSp	Parch
Survived					
0	549	549	549	549	549
1	342	342	342	342	342

```
In [152]: y=data1['Survived']  
x=data2.drop('Survived',axis=1)
```

```
In [153]: y
```

```
Out[153]: 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 [154]:

x

Out[154]:

	Pclass	Sex	Age	SibSp	Parch
0	3	1	22.0	1	0
1	1	0	38.0	1	0
2	3	0	26.0	0	0
3	1	0	35.0	1	0
4	3	1	35.0	0	0
...
886	2	1	27.0	0	0
887	1	0	19.0	0	0
888	3	0	28.0	1	2
889	1	1	26.0	0	0
890	3	1	32.0	0	0

891 rows × 5 columns

```
In [155]: 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 [156]: from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[156]:

▼ LogisticRegression

LogisticRegression()

```
In [157]: y_pred=classifier.predict(x_test)
```

```
In [158]: y_pred
```

```
Out[158]: 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, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 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, 0, 0, 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, 0, 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, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
                1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0,
                0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
                0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
                0, 0, 0, 0, 1, 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 [159]: from sklearn.metrics import confusion_matrix
          confusion_matrix(y_test, y_pred)
```

```
Out[159]: array([[158, 17],
                [ 33, 87]])
```

```
In [160]: from sklearn.metrics import accuracy_score
          accuracy_score(y_test, y_pred)
```

```
Out[160]: 0.8305084745762712
```

In [161]:

y

Out[161]:

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 []: