

In [16]: 1 `import pandas as pd`

In [17]: 1 `data=pd.read_csv("/home/palcement/Downloads/Titanic Dataset.csv")`

In [18]: 1 `data`

Out[18]:

	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 [19]: 1 data.describe()

Out[19]:

	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 [20]: 1 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 [21]: 1 data.isna().sum()
```

```
Out[21]: 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 [22]: 1 data.Pclass.unique()
```

```
Out[22]: array([3, 1, 2])
```

```
In [23]: 1 data.Survived.unique()
```

```
Out[23]: array([0, 1])
```

```
In [24]: 1 data.Age.unique()
```

```
Out[24]: 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 [25]: 1 data.SibSp.unique()
```

```
Out[25]: array([1, 0, 3, 4, 2, 5, 8])
```

```
In [26]: 1 data.Cabin.unique()
```

```
Out[26]: 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 [27]: 1 data.head(3)
```

```
Out[27]:
```

	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

```
In [29]: 1 data1=data.drop(['PassengerId', 'Name', 'Cabin', 'Ticket', 'SibSp', 'Parch'],axis=1)
```

In [30]: 1 data1

Out[30]:

	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 [33]: 1 data1['Sex']=data1['Sex'].map({'male':0,'female':1})  
2

In [34]: 1 data1

Out[34]:

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

891 rows × 6 columns

In [35]: 1 data2=data1.fillna(data1.median())

/tmp/ipykernel\_9513/3414091449.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(data1.median())

```
In [36]: 1 data2
```

```
Out[36]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	0	22.0	7.2500	S
1	1	1	1	38.0	71.2833	C
2	1	3	1	26.0	7.9250	S
3	1	1	1	35.0	53.1000	S
4	0	3	0	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	0	27.0	13.0000	S
887	1	1	1	19.0	30.0000	S
888	0	3	1	28.0	23.4500	S
889	1	1	0	26.0	30.0000	C
890	0	3	0	32.0	7.7500	Q

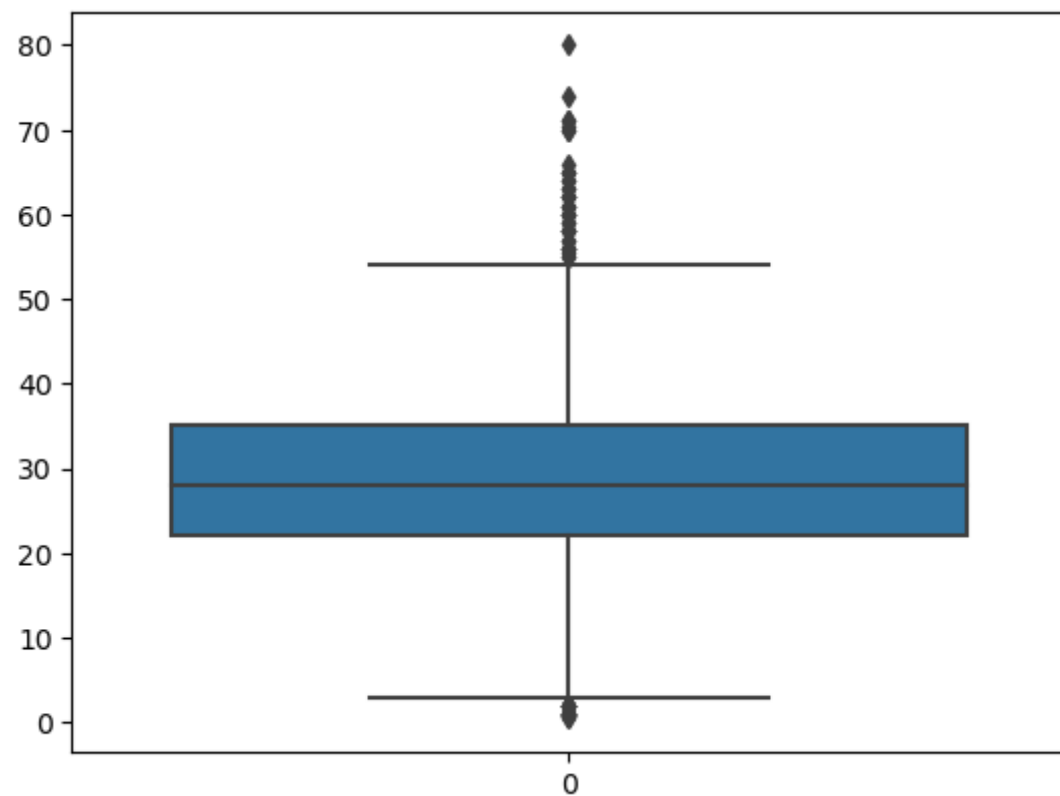
891 rows × 6 columns

```
In [37]: 1 data2.isna().sum()
```

```
Out[37]: Survived    0
Pclass    0
Sex        0
Age        0
Fare       0
Embarked   2
dtype: int64
```

```
In [39]: 1 import seaborn as sns  
2 import matplotlib.pyplot as plt  
3 sns.boxplot(data2.Age)
```

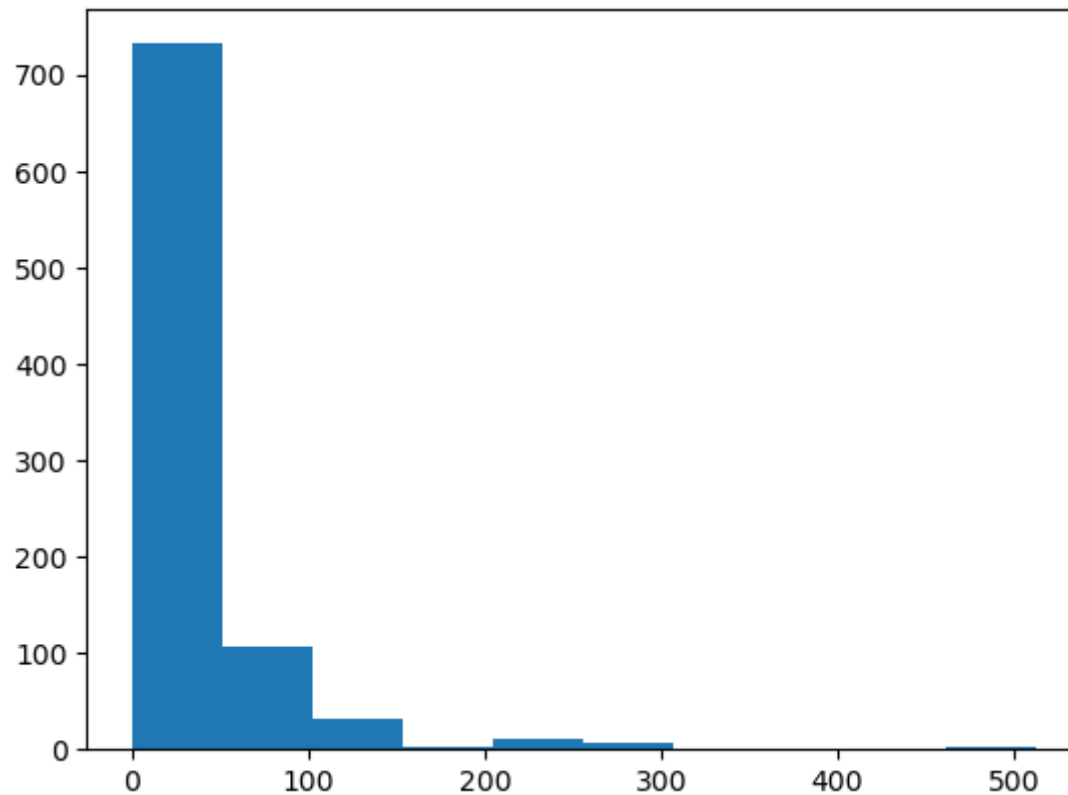
Out[39]: <Axes: >





```
In [42]: 1 plt.hist(data2['Fare'])
```

```
Out[42]: (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 [43]: 1 data2.isna().sum()
```

```
Out[43]: Survived      0
          Pclass       0
          Sex          0
          Age          0
          Fare         0
          Embarked     2
          dtype: int64
```

```
In [44]: 1 data2.describe()
```

```
Out[44]:
```

	Survived	Pclass	Sex	Age	Fare
<b>count</b>	891.000000	891.000000	891.000000	891.000000	891.000000
<b>mean</b>	0.383838	2.308642	0.352413	29.361582	32.204208
<b>std</b>	0.486592	0.836071	0.477990	13.019697	49.693429
<b>min</b>	0.000000	1.000000	0.000000	0.420000	0.000000
<b>25%</b>	0.000000	2.000000	0.000000	22.000000	7.910400
<b>50%</b>	0.000000	3.000000	0.000000	28.000000	14.454200
<b>75%</b>	1.000000	3.000000	1.000000	35.000000	31.000000
<b>max</b>	1.000000	3.000000	1.000000	80.000000	512.329200

```
In [45]: 1 data2['Age'].unique()
```

```
Out[45]: 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 [50]: 1 data2.groupby(['Age']).count()
```

```
Out[50]:
```

	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 [52]: 1 data2

Out[52]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	0	22.0	7.2500	S
1	1	1	1	38.0	71.2833	C
2	1	3	1	26.0	7.9250	S
3	1	1	1	35.0	53.1000	S
4	0	3	0	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	0	27.0	13.0000	S
887	1	1	1	19.0	30.0000	S
888	0	3	1	28.0	23.4500	S
889	1	1	0	26.0	30.0000	C
890	0	3	0	32.0	7.7500	Q

891 rows × 6 columns

```
In [53]: 1 data2['Pclass']=data2['Pclass'].map({1:'F',2:'S',3:'Third'})
          2 data2
```

```
Out[53]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	Third	0	22.0	7.2500	S
1	1	F	1	38.0	71.2833	C
2	1	Third	1	26.0	7.9250	S
3	1	F	1	35.0	53.1000	S
4	0	Third	0	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	S	0	27.0	13.0000	S
887	1	F	1	19.0	30.0000	S
888	0	Third	1	28.0	23.4500	S
889	1	F	0	26.0	30.0000	C
890	0	Third	0	32.0	7.7500	Q

891 rows × 6 columns

```
In [54]: 1 data2=pd.get_dummies(data2)
```

In [55]: 1 data2

Out[55]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_C	Embarked_Q	Embarked_S
0	0	0	22.0	7.2500	0	0	1	0	0	1
1	1	1	38.0	71.2833	1	0	0	1	0	0
2	1	1	26.0	7.9250	0	0	1	0	0	1
3	1	1	35.0	53.1000	1	0	0	0	0	1
4	0	0	35.0	8.0500	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...
886	0	0	27.0	13.0000	0	1	0	0	0	1
887	1	1	19.0	30.0000	1	0	0	0	0	1
888	0	1	28.0	23.4500	0	0	1	0	0	1
889	1	0	26.0	30.0000	1	0	0	1	0	0
890	0	0	32.0	7.7500	0	0	1	0	1	0

891 rows × 10 columns

In [57]: 1 data2.isna().sum()

Out[57]:

Survived	0
Sex	0
Age	0
Fare	0
Pclass_F	0
Pclass_S	0
Pclass_Third	0
Embarked_C	0
Embarked_Q	0
Embarked_S	0
dtype:	int64

In [59]:

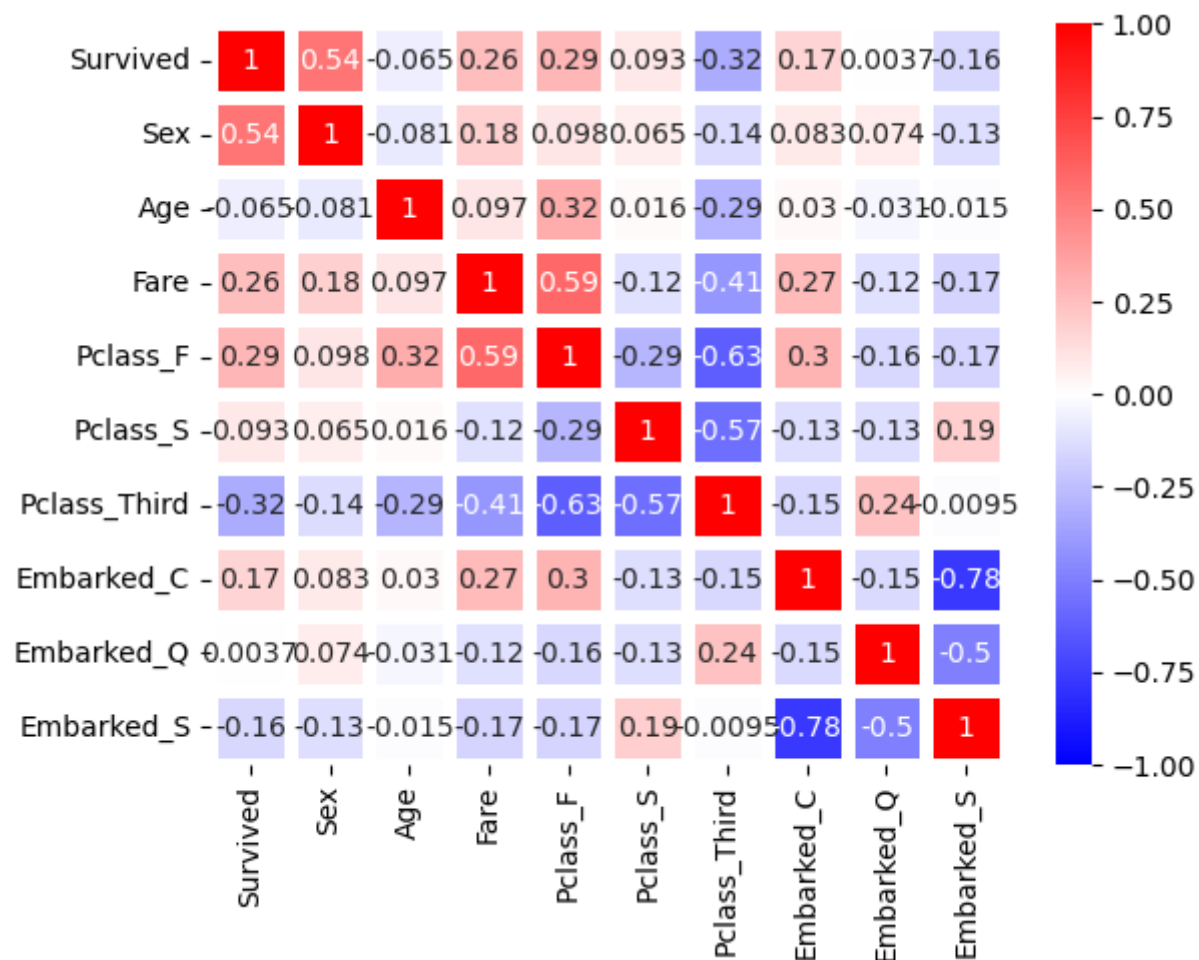
```
1 cor_mat=data2.corr()
2 cor_mat
```

Out[59]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	0.543351	-0.064910	0.257307	0.285904	0.093349	-0.322308	0.168240	0.003650	-0.155660
Sex	0.543351	1.000000	-0.081163	0.182333	0.098013	0.064746	-0.137143	0.082853	0.074115	-0.125722
Age	-0.064910	-0.081163	1.000000	0.096688	0.323896	0.015831	-0.291955	0.030248	-0.031415	-0.014665
Fare	0.257307	0.182333	0.096688	1.000000	0.591711	-0.118557	-0.413333	0.269335	-0.117216	-0.166603
Pclass_F	0.285904	0.098013	0.323896	0.591711	1.000000	-0.288585	-0.626738	0.296423	-0.155342	-0.170379
Pclass_S	0.093349	0.064746	0.015831	-0.118557	-0.288585	1.000000	-0.565210	-0.125416	-0.127301	0.192061
Pclass_Third	-0.322308	-0.137143	-0.291955	-0.413333	-0.626738	-0.565210	1.000000	-0.153329	0.237449	-0.009511
Embarked_C	0.168240	0.082853	0.030248	0.269335	0.296423	-0.125416	-0.153329	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.074115	-0.031415	-0.117216	-0.155342	-0.127301	0.237449	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	-0.125722	-0.014665	-0.166603	-0.170379	0.192061	-0.009511	-0.778359	-0.496624	1.000000

```
In [61]: 1 import seaborn as sns
          2 sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidths=5,cmap='bwr')
```

Out[61]: <Axes: >





In [63]: 1 data.groupby('Survived').count()

Out[63]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Survived											
0	549	549	549	549	424	549	549	549	549	68	549
1	342	342	342	342	290	342	342	342	342	136	340

In [67]: 1 y=data2['Survived'] *#in the dataset named as fiat500, we simply only take the price as separate and st*  
 2 X=data2.drop('Survived',axis=1)

In [68]: 1 from sklearn.model\_selection import train\_test\_split  
 2 X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.33,random\_state=42)

In [72]: 1 from sklearn.linear\_model import LogisticRegression  
 2 classifier=LogisticRegression() *#creating of Logistic Regression*  
 3 classifier.fit(X\_train,y\_train) *#training and fitting LogisticRegression object using training data*

Out[72]: LogisticRegression()  
 In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.  
 On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [73]: 1 y\_pred=classifier.predict(X\_test)

```
In [74]: 1 y_pred
```

```
Out[74]: 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, 1, 0, 0, 0, 0, 1, 1, 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, 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, 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 [75]: 1 from sklearn.metrics import confusion_matrix
          2 confusion_matrix(y_test,y_pred)
```

```
Out[75]: array([[154,  21],
                [ 37,  83]])
```

```
In [77]: 1 from sklearn.metrics import accuracy_score
          2 accuracy_score(y_test,y_pred)
```

```
Out[77]: 0.8033898305084746
```

```
In [ ]: 1
```