

Importing wanted packages

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as mp
import warnings
warnings.filterwarnings('ignore')
```

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

Out[2]:

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

Out[3]:

	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 [4]: `data.tail(5)`

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

data info

In [5]: `info=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 [6]: `nul=data.isna().sum()`
`nul`

```
Out[6]: 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 [7]: data[data['Cabin'].isnull()]
```

```
Out[7]:
```

	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
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	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
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
...
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

687 rows × 12 columns

unique values

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

```
Out[8]: array([3, 1, 2])
```

```
In [9]: data['Survived'].unique()
```

```
Out[9]: array([0, 1])
```

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

```
Out[10]: array([1, 0, 3, 4, 2, 5, 8])
```

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

```
Out[11]: array([0, 1, 2, 5, 3, 4, 6])
```

```
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.  ])
```

dropping unwanted columns

```
In [13]: drp=data.drop(['PassengerId', 'Name', 'Cabin', 'Parch', 'SibSp', 'Ticket'],axis=1)
drp
```

Out[13]:

	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 [14]: list(drp)
```

```
Out[14]: ['Survived', 'Pclass', 'Sex', 'Age', 'Fare', 'Embarked']
```

```
In [15]: drp['Sex']=drp['Sex'].map({'male':1, 'female':2})  
drp
```

Out[15]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	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 [16]: drp.fillna(drp.mode)
```

```
Out[16]:
```

	Survived	Pclass	Sex		Age	Fare	Embarked
0	0	3	1		22.0	7.2500	S
1	1	1	2		38.0	71.2833	C
2	1	3	2		26.0	7.9250	S
3	1	1	2		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	2		19.0	30.0000	S
888	0	3	2	<bound method DataFrame.mode of Survived ...	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 [17]: drp['Age'].fillna(29,inplace=True)
drp
```

Out[17]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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 [18]: drp.fillna(29,inplace=True)
drp
```

Out[18]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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

mean,median & mode

```
In [19]: me=drp.fillna(drp.mean)
me
```

Out[19]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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]: md=drp.fillna(drp.median)
md
```

Out[20]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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 [21]: mo=drp.fillna(drp.mode)
mo
```

Out[21]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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 [22]: drp['Age']
```

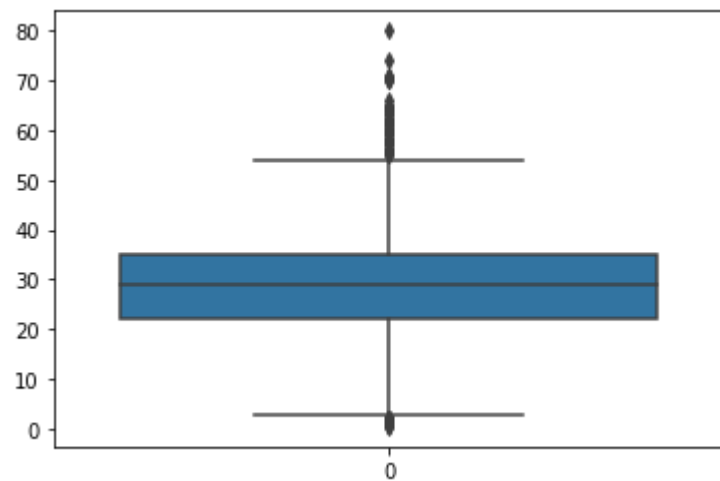
```
Out[22]: 0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
```

```
...
886    27.0
887    19.0
888    29.0
889    26.0
890    32.0
```

Name: Age, Length: 891, dtype: float64

```
In [23]: sb.boxplot(drp['Age'])
```

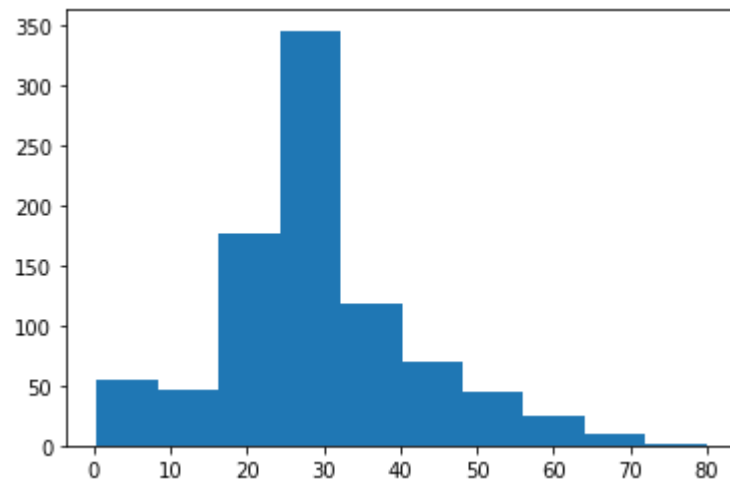
```
Out[23]: <Axes: >
```



histograph

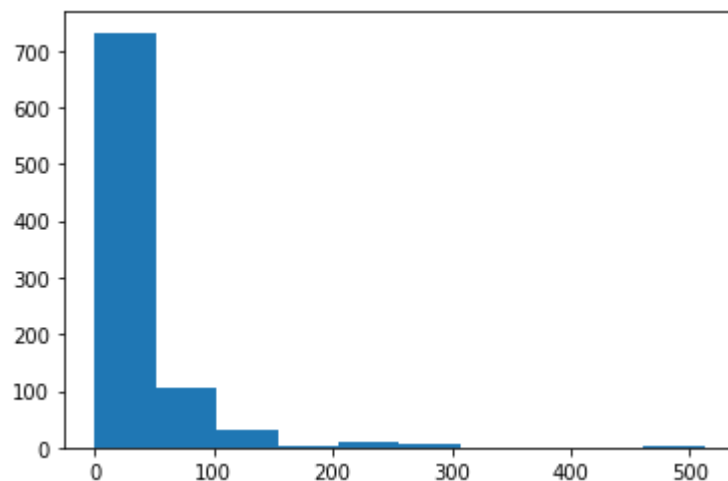
```
In [24]: mp.hist(drp['Age'])
```

```
Out[24]: (array([ 54.,  46., 177., 346., 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 [25]: mp.hist(drp['Fare'])
```

```
Out[25]: (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>)
```



finding null values

```
In [26]: md.isna().sum()
```

```
Out[26]: Survived    0  
Pclass      0  
Sex         0  
Age         0  
Fare        0  
Embarked    0  
dtype: int64
```


In [27]: `drp.describe()`

Out[27]:

	Survived	Pclass	Sex	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	1.352413	29.560236	32.204208
std	0.486592	0.836071	0.477990	13.005010	49.693429
min	0.000000	1.000000	1.000000	0.420000	0.000000
25%	0.000000	2.000000	1.000000	22.000000	7.910400
50%	0.000000	3.000000	1.000000	29.000000	14.454200
75%	1.000000	3.000000	2.000000	35.000000	31.000000
max	1.000000	3.000000	2.000000	80.000000	512.329200

unique values

In [28]: `un=drp['Age'].unique()`
`un`

Out[28]: `array([22. , 38. , 26. , 35. , 29. , 54. , 2. , 27. , 14. ,
4. , 58. , 20. , 39. , 55. , 31. , 34. , 15. , 28. ,
8. , 19. , 40. , 66. , 42. , 21. , 18. , 3. , 7. ,
49. , 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.])`

grouping data

```
In [29]: drp.groupby(['Age']).count()
```

Out[29]:

	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 [30]: `drp['Age']`

```
Out[30]: 0      22.0
          1      38.0
          2      26.0
          3      35.0
          4      35.0
          ...
          886    27.0
          887    19.0
          888    29.0
          889    26.0
          890    32.0
          Name: Age, Length: 891, dtype: float64
```

In [31]: `drp`

```
Out[31]:
```

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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 [32]: dum1=pd.get_dummies(drp,dtype=int)
dum1
```

Out[32]:

	Survived	Pclass	Sex	Age	Fare	Embarked_29	Embarked_C	Embarked_Q	Embarked_S
0	0	3	1	22.0	7.2500	0	0	0	1
1	1	1	2	38.0	71.2833	0	1	0	0
2	1	3	2	26.0	7.9250	0	0	0	1
3	1	1	2	35.0	53.1000	0	0	0	1
4	0	3	1	35.0	8.0500	0	0	0	1
...
886	0	2	1	27.0	13.0000	0	0	0	1
887	1	1	2	19.0	30.0000	0	0	0	1
888	0	3	2	29.0	23.4500	0	0	0	1
889	1	1	1	26.0	30.0000	0	1	0	0
890	0	3	1	32.0	7.7500	0	0	1	0

891 rows × 9 columns

correlation table

```
In [33]: cor=dum1.corr()  
cor
```

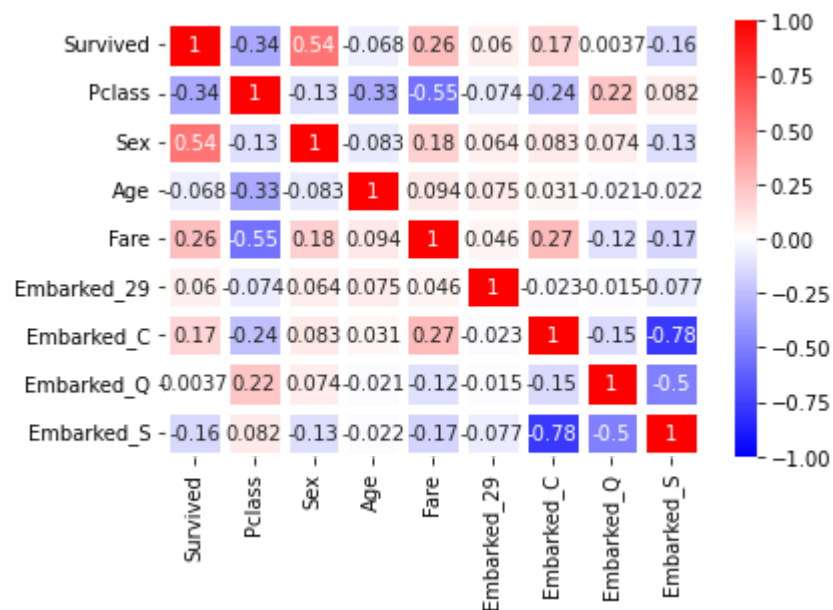
Out[33]:

	Survived	Pclass	Sex	Age	Fare	Embarked_29	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	-0.338481	0.543351	-0.067814	0.257307	0.060095	0.168240	0.003650	-0.155660
Pclass	-0.338481	1.000000	-0.131900	-0.334974	-0.549500	-0.074282	-0.243292	0.221009	0.081720
Sex	0.543351	-0.131900	1.000000	-0.082949	0.182333	0.064296	0.082853	0.074115	-0.125722
Age	-0.067814	-0.334974	-0.082949	1.000000	0.093706	0.074589	0.031303	-0.021093	-0.021999
Fare	0.257307	-0.549500	0.182333	0.093706	1.000000	0.045646	0.269335	-0.117216	-0.166603
Embarked_29	0.060095	-0.074282	0.064296	0.074589	0.045646	1.000000	-0.022864	-0.014588	-0.076588
Embarked_C	0.168240	-0.243292	0.082853	0.031303	0.269335	-0.022864	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.221009	0.074115	-0.021093	-0.117216	-0.014588	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	0.081720	-0.125722	-0.021999	-0.166603	-0.076588	-0.778359	-0.496624	1.000000

correlation graph

```
In [34]: sb.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=5,cmap='bwr')
```

```
Out[34]: <Axes: >
```



```
In [35]: data.groupby('Survived').count()
```

```
Out[35]:
```

	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 [36]: drp.fillna(35,inplace=True)
drp
```

Out[36]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	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	2	19.0	30.0000	S
888	0	3	2	29.0	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 [37]: y=dum1['Survived']
x=dum1.drop('Survived',axis=1)
```

In [38]: x

Out[38]:

	Pclass	Sex	Age	Fare	Embarked_29	Embarked_C	Embarked_Q	Embarked_S
0	3	1	22.0	7.2500	0	0	0	1
1	1	2	38.0	71.2833	0	1	0	0
2	3	2	26.0	7.9250	0	0	0	1
3	1	2	35.0	53.1000	0	0	0	1
4	3	1	35.0	8.0500	0	0	0	1
...
886	2	1	27.0	13.0000	0	0	0	1
887	1	2	19.0	30.0000	0	0	0	1
888	3	2	29.0	23.4500	0	0	0	1
889	1	1	26.0	30.0000	0	1	0	0
890	3	1	32.0	7.7500	0	0	1	0

891 rows × 8 columns

In [39]: y

Out[39]:

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 [40]: 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 [41]: x_train.head(5)
```

Out[41]:

	Pclass	Sex	Age	Fare	Embarked_29	Embarked_C	Embarked_Q	Embarked_S
6	1	1	54.0	51.8625	0	0	0	1
718	3	1	29.0	15.5000	0	0	1	0
685	2	1	25.0	41.5792	0	1	0	0
73	3	1	26.0	14.4542	0	1	0	0
882	3	2	22.0	10.5167	0	0	0	1

```
In [42]: x_test.head(5)
```

Out[42]:

	Pclass	Sex	Age	Fare	Embarked_29	Embarked_C	Embarked_Q	Embarked_S
709	3	1	29.0	15.2458	0	1	0	0
439	2	1	31.0	10.5000	0	0	0	1
840	3	1	20.0	7.9250	0	0	0	1
720	2	2	6.0	33.0000	0	0	0	1
39	3	2	14.0	11.2417	0	1	0	0

```
In [43]: y_train.head(5)
```

```
Out[43]: 6      0
718      0
685      0
73       0
882      0
Name: Survived, dtype: int64
```

In [44]: `y_test.head(5)`

```
Out[44]: 709    1
         439    0
         840    0
         720    1
          39    1
         Name: Survived, dtype: int64
```

Logistic regression

In [45]: `from sklearn.linear_model import LogisticRegression`
`classifier=LogisticRegression()`
`classifier.fit(x_train,y_train)`

```
Out[45]: ▼ LogisticRegression
         LogisticRegression()
```

In [46]: `y_pred=classifier.predict(x_test)`
`y_pred`

```
Out[46]: array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
                1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
                1, 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, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
                1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 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, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
                0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
                0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 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, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                1, 0, 0, 0, 0, 0, 1, 1, 0])
```

```
In [47]: from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,y_pred)
```

```
Out[47]: array([[149,  26],  
               [ 31,  89]])
```

```
In [48]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,y_pred)
```

```
Out[48]: 0.8067796610169492
```

```
In [49]: from sklearn.metrics import r2_score  
r2_score(y_test,y_pred)
```

```
Out[49]: 0.1992857142857143
```

```
In [ ]:
```