

```
Name:K.Nikhil kumar
```

```
Reg no:21BCE7801
```

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

```
df=pd.read_csv("Titanic-Dataset.csv")
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599 STON/O2. 3101282
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel) Allen, Mr.	female	26.0	0	0	113803 373450
3	4	1	1		female	35.0	1	0	
4	5	0	3		male	35.0	0	0	

```
df.head()
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599 STON/O2. 3101282	71.2833	C85	C
2	3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel) Allen, Mr. William Henry	female	26.0	0	0	113803 373450	7.9250	NaN	S
3	4	1		female	35.0	1	0		53.1000	C123	S
4	5	0		male	35.0	0	0		8.0500	NaN	S

```
df.describe()
```

	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

```
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
```

```
df.corr()
```

```
<ipython-input-8-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	grid icon	more icon
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		

```
df.corr().Survived.sort_values(ascending=False)
```

```
<ipython-input-9-fe51b8bb09d5>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
df.corr().Survived.sort_values(ascending=False)
```

```
Survived      1.000000
Fare         0.257307
Parch        0.081629
PassengerId -0.005007
SibSp        -0.035322
Age          -0.077221
Pclass       -0.338481
Name: Survived, dtype: float64
```

```
df.corr().PassengerId.sort_values(ascending=False)
```

```
<ipython-input-10-cdcf9d104af7>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
df.corr().PassengerId.sort_values(ascending=False)
```

```
PassengerId  1.000000
Age          0.036847
Fare         0.012658
Parch        -0.001652
Survived     -0.005007
Pclass       -0.035144
SibSp        -0.057527
Name: PassengerId, dtype: float64
```

```
df.corr().Pclass.sort_values(ascending=False)
```

```
<ipython-input-11-a3a032996e11>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
df.corr().Pclass.sort_values(ascending=False)
```

```
Pclass       1.000000
SibSp        0.083081
Parch        0.018443
PassengerId -0.035144
Survived     -0.338481
Age          -0.369226
Fare         -0.549500
Name: Pclass, dtype: float64
```

```
df.corr().Age.sort_values(ascending=False)
```

```
<ipython-input-12-675a3c7ad3f2>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
 df.corr().Age.sort_values(ascending=False)
Age      1.000000
Fare     0.096067
PassengerId  0.036847
Survived   -0.077221
Parch     -0.189119
SibSp     -0.308247
Pclass    -0.369226
Name: Age, dtype: float64
```



```
df.corr().SibSp.sort_values(ascending=False)
```

```
<ipython-input-13-51af5db812ae>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
 df.corr().SibSp.sort_values(ascending=False)
SibSp      1.000000
Parch     0.414838
Fare      0.159651
Pclass    0.083081
Survived   -0.035322
PassengerId -0.057527
Age       -0.308247
Name: SibSp, dtype: float64
```



```
df.corr().Parch.sort_values(ascending=False)
```

```
<ipython-input-14-dcd2878cae59>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
 df.corr().Parch.sort_values(ascending=False)
Parch      1.000000
SibSp     0.414838
Fare      0.216225
Survived   0.081629
Pclass    0.018443
PassengerId -0.001652
Age       -0.189119
Name: Parch, dtype: float64
```



```
df.corr().Fare.sort_values(ascending=False)
```

```
<ipython-input-15-f51f352aac84>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
 df.corr().Fare.sort_values(ascending=False)
Fare      1.000000
Survived  0.257307
Parch     0.216225
SibSp     0.159651
Age       0.096067
PassengerId  0.012658
Pclass    -0.549500
Name: Fare, dtype: float64
```



```
df.isnull().any()
```

```
PassengerId  False
Survived    False
Pclass      False
Name        False
Sex         False
Age         True
SibSp      False
Parch      False
Ticket     False
Fare        False
Cabin      True
Embarked   True
dtype: bool
```

```
df.isnull().sum()
```

```
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
```

```
df.Age.nunique()
```

```
88
```

```
df.PassengerId.nunique()
```

```
891
```

```
df.Survived.nunique()
```

```
2
```

```
df.Pclass.nunique()
```

```
3
```

```
df.Name.nunique()
```

```
891
```

```
df.SibSp.nunique()
```

```
7
```

```
df.Parch.nunique()
```

```
7
```

```
df.Fare.nunique()
```

```
248
```

```
df.Cabin.nunique()
```

```
147
```

```
df.Embarked.nunique()
```

```
3
```

```
df.Sex.nunique()
```

```
2
```

```
df.Ticket.nunique()
```

```
681
```

```
df.Age.unique()
```

```
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. ])
```

```
df.PassengerId.unique()
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13,
       14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
       40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,
       53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
       66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78,
       79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
       92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104,
       105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
       118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
       131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
       144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
       157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
       196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208,
       209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
       222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
       235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247,
       248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
       261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273,
       274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286,
       287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299,
       300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312,
       313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325,
       326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338,
       339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351,
       352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364,
       365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377,
       378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
       391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403,
       404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416,
       417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429,
       430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442,
       443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455,
       456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468,
       469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481,
       482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494,
       495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507,
       508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520,
       521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533,
       534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546,
       547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559,
       560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572,
       573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585,
       586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598,
       599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611,
       612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624,
       625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637,
       638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650,
       651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663,
       664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676,
       677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689,
       690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702,
       703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715,
       716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728,
       729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741,
       742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754,
```

```
df.Survived.unique()
```

```
array([0, 1])
```

```
df.Pclass.unique()
```

```
array([3, 1, 2])
```

```
df.Name.unique()
```

```
array(['Braund, Mr. Owen Harris',
       'Cumings, Mrs. John Bradley (Florence Briggs Thayer)',
       'Heikkinen, Miss. Laina',
       'Futrelle, Mrs. Jacques Heath (Lily May Peel)',
       'Allen, Mr. William Henry', 'Moran, Mr. James',
       'McCarthy, Mr. Timothy J', 'Palsson, Master. Gosta Leonard',
       'Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)',
       'Nasser, Mrs. Nicholas (Adele Achem)',
       'Sandstrom, Miss. Marguerite Rut', 'Bonnell, Miss. Elizabeth',
       'Saunderscock, Mr. William Henry', 'Andersson, Mr. Anders Johan',
       'Vestrom, Miss. Hulda Amanda Adolfina',
       'Hewlett, Mrs. (Mary D Kingcome)', 'Rice, Master. Eugene',
       'Williams, Mr. Charles Eugene',
       'Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)'],
```

'Masselmani, Mrs. Fatima', 'Fynney, Mr. Joseph J',
 'Beesley, Mr. Lawrence', 'McGowan, Miss. Anna "Annie"',
 'Sloper, Mr. William Thompson', 'Palsson, Miss. Torborg Danira',
 'Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)',
 'Emir, Mr. Farred Chehab', 'Fortune, Mr. Charles Alexander',
 'O'Dwyer, Miss. Ellen "Nellie"', 'Todoroff, Mr. Lailo',
 'Uruchurtu, Don. Manuel E',
 'Spencer, Mrs. William Augustus (Marie Eugenie)',
 'Glynn, Miss. Mary Agatha', 'Wheadon, Mr. Edward H',
 'Meyer, Mr. Edgar Joseph', 'Holverson, Mr. Alexander Oskar',
 'Mamee, Mr. Hanna', 'Cann, Mr. Ernest Charles',
 'Vander Planke, Miss. Augusta Maria',
 'Nicola-Yarred, Miss. Jamila',
 'Ahlin, Mrs. Johan (Johanna Persdotter Larsson)',
 'Turpin, Mrs. William John Robert (Dorothy Ann Wonnacott)',
 'Kraeff, Mr. Theodor', 'Laroche, Miss. Simonne Marie Anne Andree',
 'Devaney, Miss. Margaret Delia', 'Rogers, Mr. William John',
 'Lennon, Mr. Denis', "O'Driscoll, Miss. Bridget",
 'Samaan, Mr. Youssef',
 'Arnold-Franchi, Mrs. Josef (Josefine Franchi)',
 'Panula, Master. Juha Niilo', 'Nosworthy, Mr. Richard Cater',
 'Harper, Mrs. Henry Sleeper (Myra Haxton)',
 'Faunthorpe, Mrs. Lizzie (Elizabeth Anne Wilkinson)',
 'Ostby, Mr. Engelhart Cornelius', 'Woolner, Mr. Hugh',
 'Rugg, Miss. Emily', 'Novel, Mr. Mansouer',
 'West, Miss. Constance Mirium',
 'Goodwin, Master. William Frederick', 'Sirayanian, Mr. Orsen',
 'Icard, Miss. Amelie', 'Harris, Mr. Henry Birkhardt',
 'Skoog, Master. Harald', 'Stewart, Mr. Albert A',
 'Moubarek, Master. Garios', 'Nye, Mrs. (Elizabeth Ramell)',
 'Crease, Mr. Ernest James', 'Andersson, Miss. Erna Alexandra',
 'Kink, Mr. Vincenz', 'Jenkin, Mr. Stephen Curnow',
 'Goodwin, Miss. Lillian Amy', 'Hood, Mr. Ambrose Jr',
 'Chronopoulos, Mr. Apostolos', 'Bing, Mr. Lee',
 'Moen, Mr. Sigurd Hansen', 'Staneff, Mr. Ivan',
 'Moutal, Mr. Rahamin Haim', 'Caldwell, Master. Alden Gates',
 'Dowdell, Miss. Elizabeth', 'Waelens, Mr. Achille',
 'Sheerlinck, Mr. Jan Baptist', 'McDermott, Miss. Brigidet Delia',
 'Carrau, Mr. Francisco M', 'Ilett, Miss. Bertha',
 'Backstrom, Mrs. Karl Alfred (Maria Mathilda Gustafsson)',
 'Ford, Mr. William Neal', 'Slocovski, Mr. Selman Francis',
 'Fortune, Miss. Mabel Helen', 'Celotti, Mr. Francesco',
 'Christmann, Mr. Emil', 'Andreasson, Mr. Paul Edvin',
 'Chaffee, Mr. Herbert Fuller', 'Dean, Mr. Bertram Frank'.

```
df.SibSp.unique()
```

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

```
df.Parch.unique()
```

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

```
df.Fare.unique()
```

```
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, 10.5167])
```

```
df.Cabin.unique()
```

```
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)
```

```
df.Embarked.unique()
```

```
array(['S', 'C', 'Q', nan], dtype=object)
```

```
df.Sex.unique()
```

```
array(['male', 'female'], dtype=object)
```

```
df.Ticket.unique()
```

```
'349201', '349218', '16988', '3/6566', 'STON/O 2. 3101288',  
'250648', '113773', '335097', '29103', '392096', '345780',  
'349204', '350042', '29108', '363294', 'SOTON/02 3101272', '2663',  
'347074', '112379', '364850', '8471', '345781', '350047',  
'S.O./P.P. 3', '2674', '29105', '347078', '383121', '36865',  
'2687', '113501', 'W.C. 6607', 'SOTON/O.Q. 3101312', '374887',  
'3101265', '12468', 'PC 17600', '349203', '28213', '17465',  
'349244', '2685', '2625', '347089', '347063', '112050', '347087',  
'248723', '3474', '28206', '364499', '112058', 'STON/02. 3101290',  
'S.C./PARIS 2079', 'C 7075', '315098', '19972', '368323', '367228',  
'2671', '347468', '2223', 'PC 17756', '315097', '392092', '11774',  
'SOTON/02 3101287', '2683', '315090', 'C.A. 5547', '349213',  
'347060', 'PC 17592', '392091', '113055', '2629', '350026',  
'28134', '17466', '233866', '236852', 'SC/PARIS 2149', 'PC 17590',  
'345777', '349248', '695', '345765', '2667', '349212', '349217',  
'349257', '7552', 'C.A./SOTON 34068', 'SOTON/OQ 392076', '211536',  
'112053', '111369', '370376'], dtype=object)
```

```
df.Age.value_counts()
```

```
24.00      30  
22.00      27  
18.00      26  
19.00      25  
28.00      25  
..  
36.50      1  
55.50      1  
0.92      1  
23.50      1  
74.00      1  
Name: Age, Length: 88, dtype: int64
```

```
df.PassengerId.value_counts()
```

```
1      1  
599    1  
588    1  
589    1  
590    1  
..  
301    1  
302    1  
303    1  
304    1  
891    1  
Name: PassengerId, Length: 891, dtype: int64
```

```
df.Survived.value_counts()
```

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

```
df.Fare.value_counts()
```

```
8.0500     43  
13.0000    42  
7.8958     38  
7.7500     34  
26.0000    31  
..  
35.0000     1  
28.5000     1  
6.2375     1  
14.0000     1  
10.5167     1  
Name: Fare, Length: 248, dtype: int64
```

```
df.Parch.value_counts()
```

```
0      678  
1      118  
2      80  
5      5  
3      5  
4      4  
6      1  
Name: Parch, dtype: int64
```

```
df.Pclass.value_counts()
```

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

```
df.Embarked.value_counts()
```

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

```
df.Ticket.value_counts()
```

```
347082      7
CA. 2343      7
1601         7
3101295      6
CA 2144       6
..
9234         1
19988        1
2693         1
PC 17612      1
370376       1
Name: Ticket, Length: 681, dtype: int64
```

```
df.Name.value_counts()
```

```
Braund, Mr. Owen Harris      1
Boulos, Mr. Hanna           1
Frolicher-Stehli, Mr. Maxmillian 1
Gilinski, Mr. Eliezer        1
Murdlin, Mr. Joseph          1
..
Kelly, Miss. Anna Katherine "Annie Kate" 1
McCoy, Mr. Bernard          1
Johnson, Mr. William Cahoone Jr 1
Keane, Miss. Nora A          1
Dooley, Mr. Patrick          1
Name: Name, Length: 891, dtype: int64
```

```
df.Sex.value_counts()
```

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

```
df.Cabin.value_counts()
```

```
B96 B98      4
G6          4
C23 C25 C27  4
C22 C26      3
F33          3
..
E34          1
C7          1
C54          1
E36          1
C148         1
Name: Cabin, Length: 147, dtype: int64
```

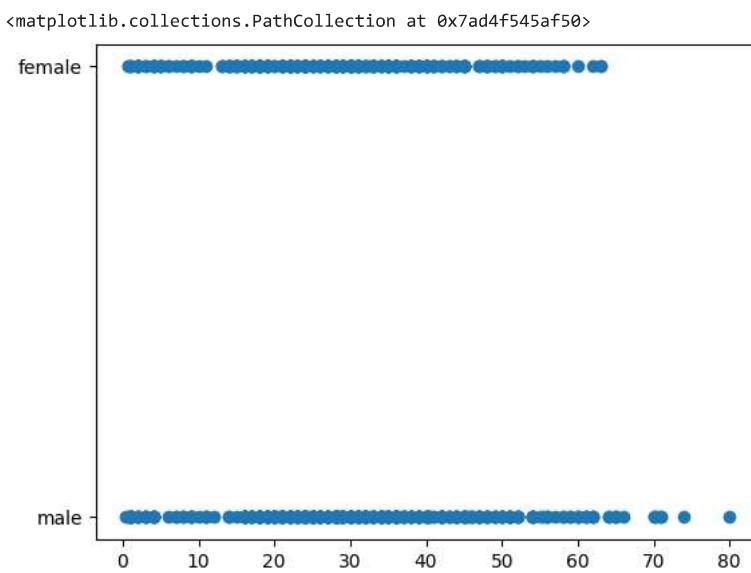
```
df.SibSp.value_counts()
```

```
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

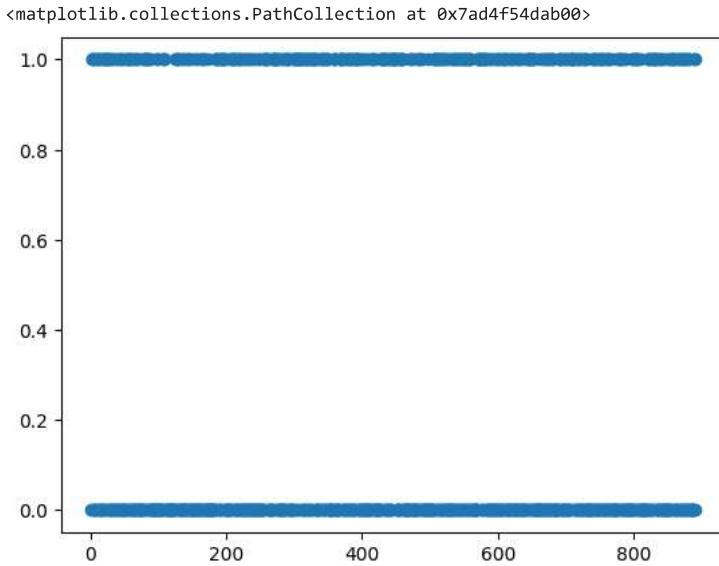
4.Data Visualization.

```
File "<ipython-input-54-65a0ddac9c41>", line 1
  4.Data Visualization.
  ^
SyntaxError: invalid decimal literal
```

```
plt.scatter(df["Age"],df["Sex"])
```



```
plt.scatter(df["PassengerId"],df["Survived"])
```

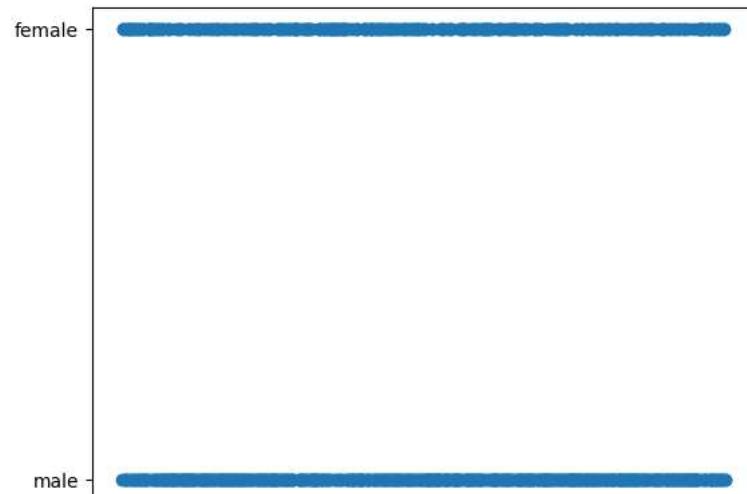


```
plt.scatter(df["Fare"],df["Age"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f545b340>
```

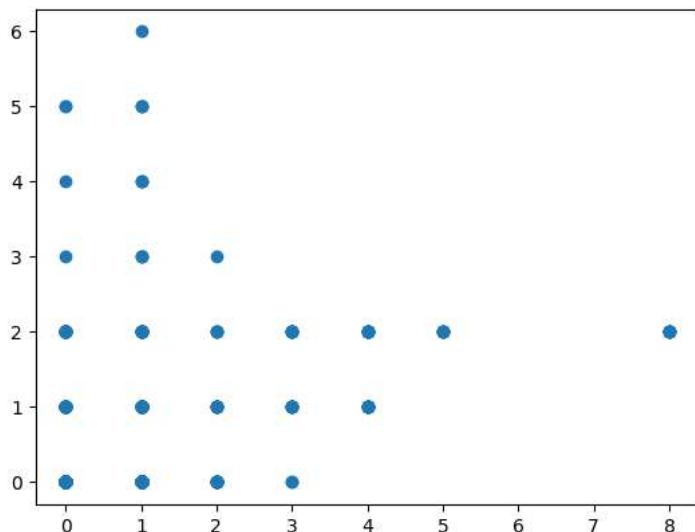
```
plt.scatter(df["Name"],df["Sex"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f53da140>
```



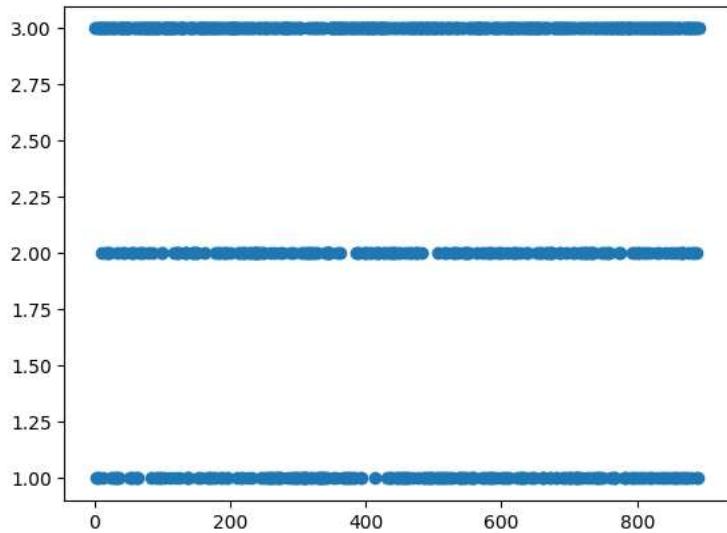
```
plt.scatter(df["SibSp"],df["Parch"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f484ebf0>
```



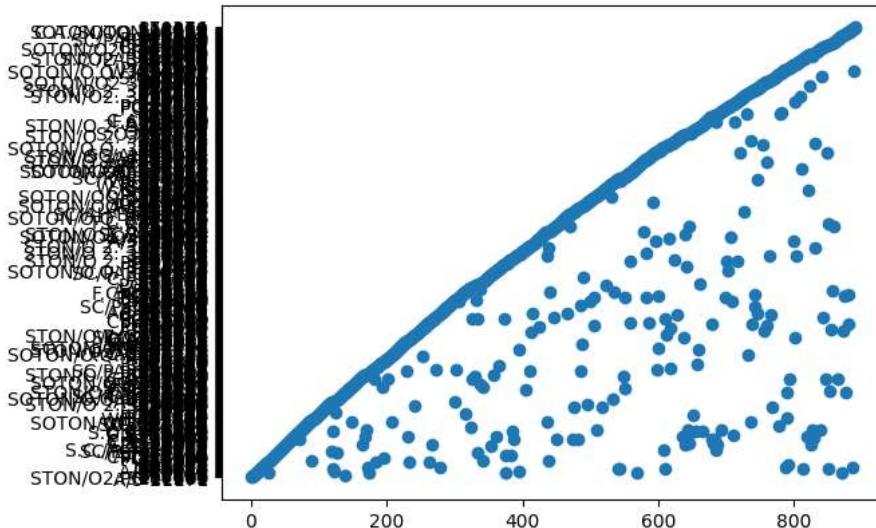
```
plt.scatter(df["PassengerId"],df["Pclass"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f45ce890>
```



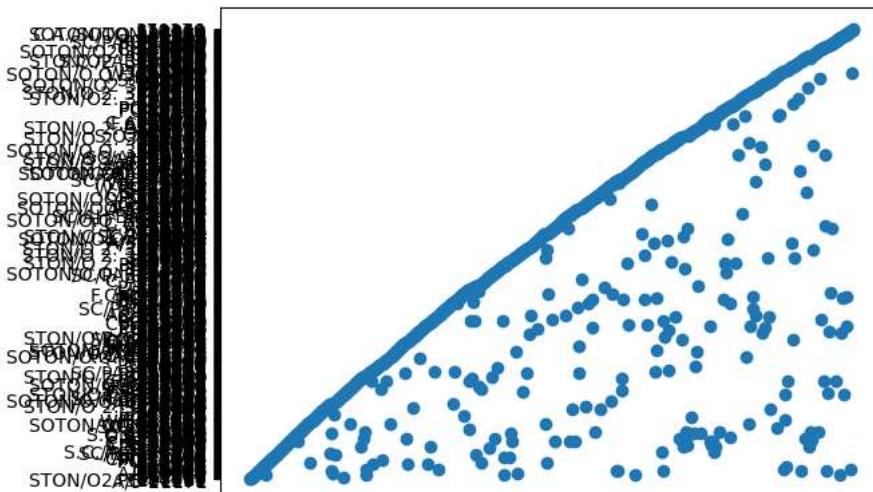
```
plt.scatter(df["PassengerId"],df["Ticket"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f42f2830>
```



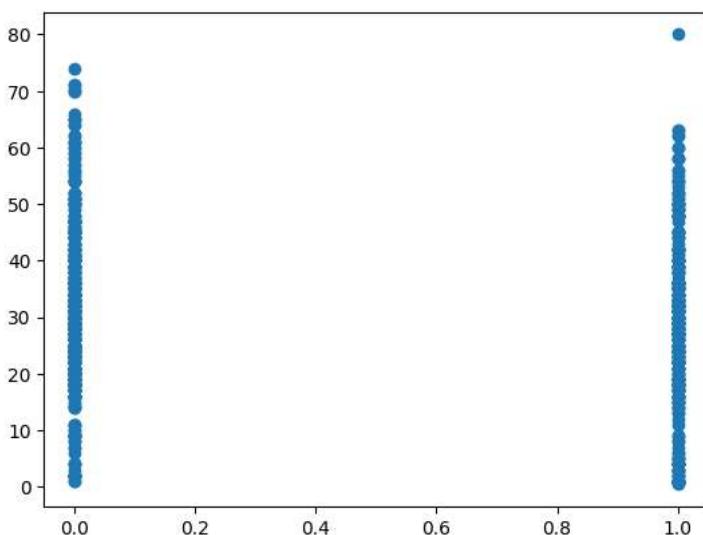
```
plt.scatter(df["Name"],df["Ticket"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f3b22320>
```



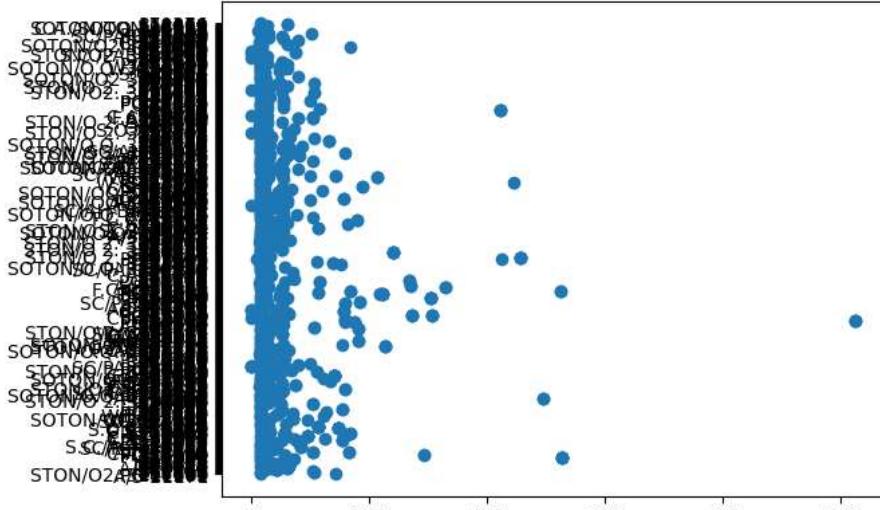
```
plt.scatter(df["Survived"],df["Age"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f3b236a0>
```



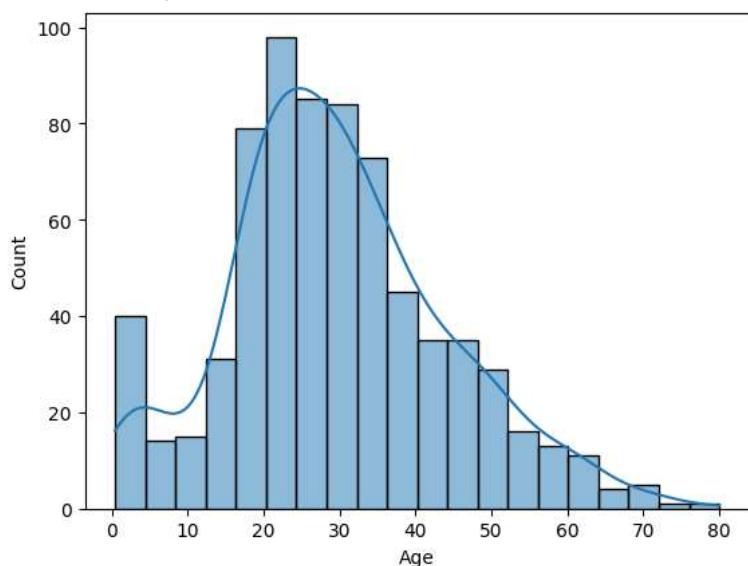
```
plt.scatter(df["Fare"],df["Ticket"])
```

```
<matplotlib.collections.PathCollection at 0x7ad4f0961cc0>
```



```
sns.histplot(data=df,x="Age",bins=20,kde=True)
```

```
<Axes: xlabel='Age', ylabel='Count'>
```

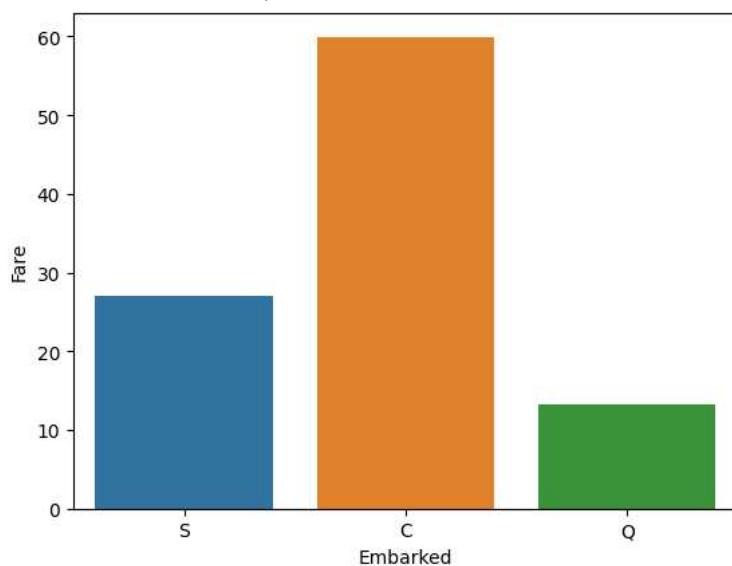


```
sns.barplot(x=df["Embarked"],y=df["Fare"],ci=None)
```

```
<ipython-input-66-f67c208bf54a>:1: FutureWarning:
```

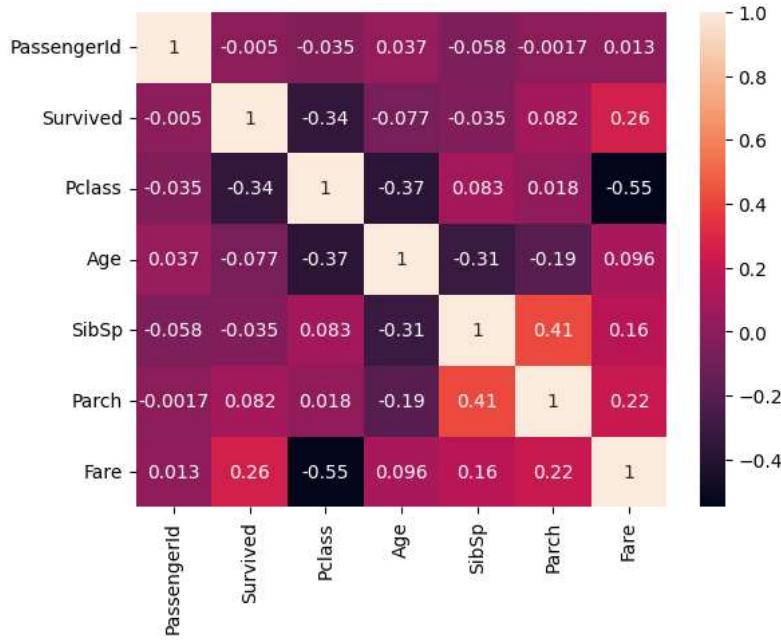
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x=df["Embarked"],y=df["Fare"],ci=None)
<Axes: xlabel='Embarked', ylabel='Fare'>
```

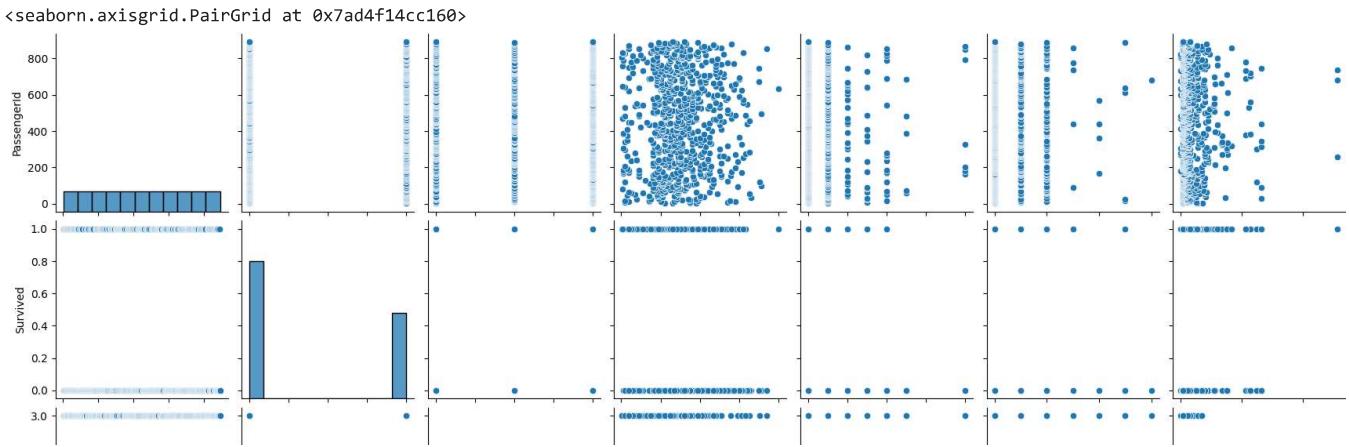


```
sns.heatmap(df.corr(), annot=True)
```

```
<ipython-input-67-8df7bcac526d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  sns.heatmap(df.corr(), annot=True)
  <Axes: >
```



```
sns.pairplot(df)
```

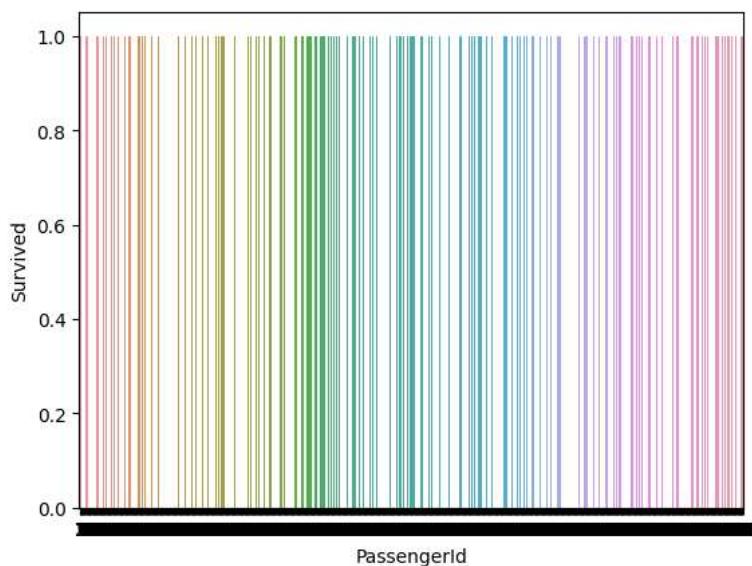


```
sns.barplot(x=df["PassengerId"],y=df["Survived"],ci=0)
```

<ipython-input-69-0a8bf95a1f9c>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

```
sns.barplot(x=df["PassengerId"],y=df["Survived"],ci=0)
<Axes: xlabel='PassengerId', ylabel='Survived'>
```

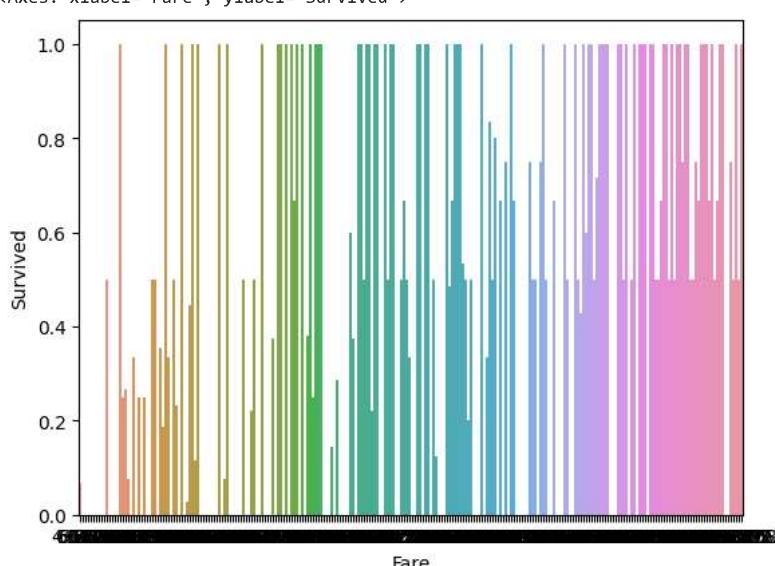


```
sns.barplot(x=df["Fare"],y=df["Survived"],ci=0)
```

<ipython-input-70-c3bcb7432623>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

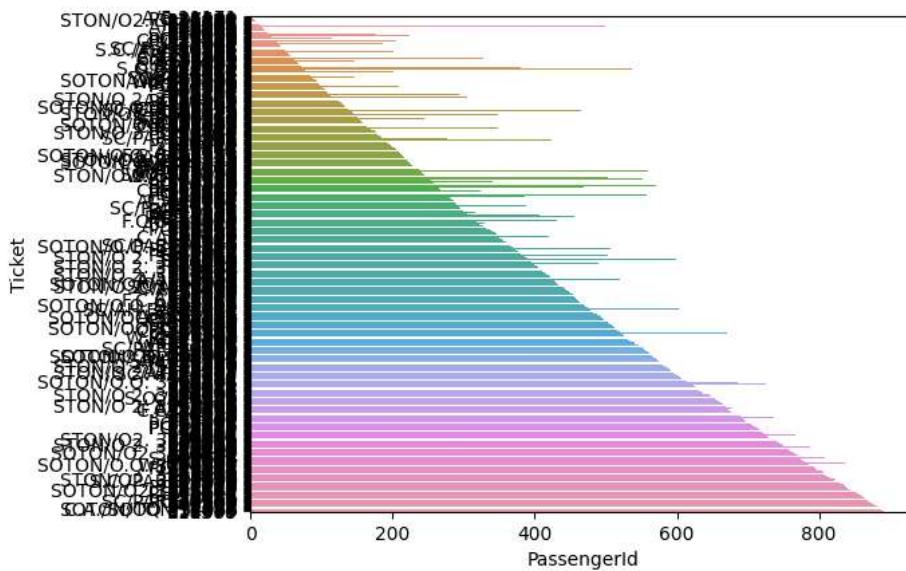
```
sns.barplot(x=df["Fare"],y=df["Survived"],ci=0)
<Axes: xlabel='Fare', ylabel='Survived'>
```



```
sns.barplot(x=df["PassengerId"],y=df["Ticket"],ci=0)

<ipython-input-71-909eba67659c>:1: FutureWarning:
The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

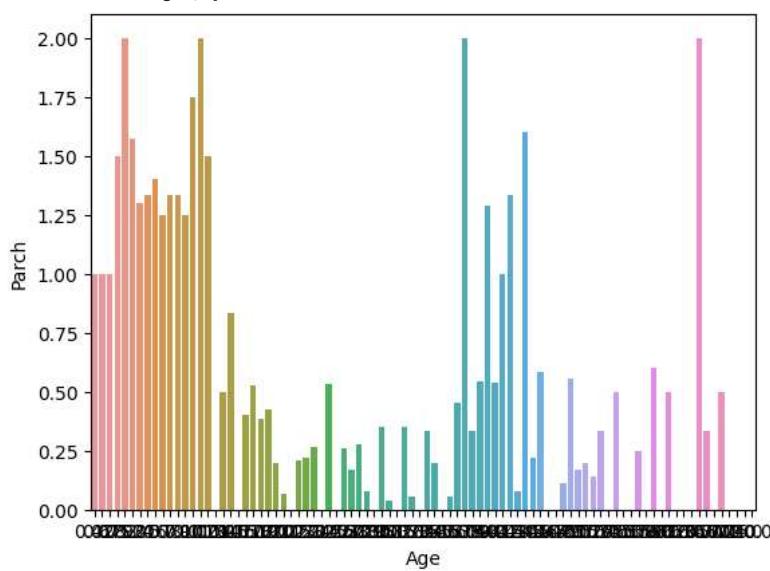
sns.barplot(x=df["PassengerId"],y=df["Ticket"],ci=0)
<Axes: xlabel='PassengerId', ylabel='Ticket'>
```



```
sns.barplot(x=df["Age"],y=df["Parch"],ci=0)

<ipython-input-72-b64b0055c75c>:1: FutureWarning:
The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

sns.barplot(x=df["Age"],y=df["Parch"],ci=0)
<Axes: xlabel='Age', ylabel='Parch'>
```

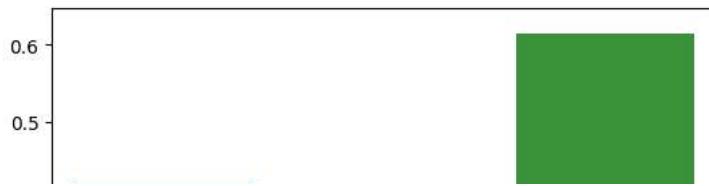


```
sns.barplot(x=df["Pclass"],y=df["SibSp"],ci=0)
```

```
<ipython-input-73-0b4c5f55dce>:1: FutureWarning:
```

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

```
sns.barplot(x=df["Pclass"],y=df["SibSp"],ci=0)
<Axes: xlabel='Pclass', ylabel='SibSp'>
```

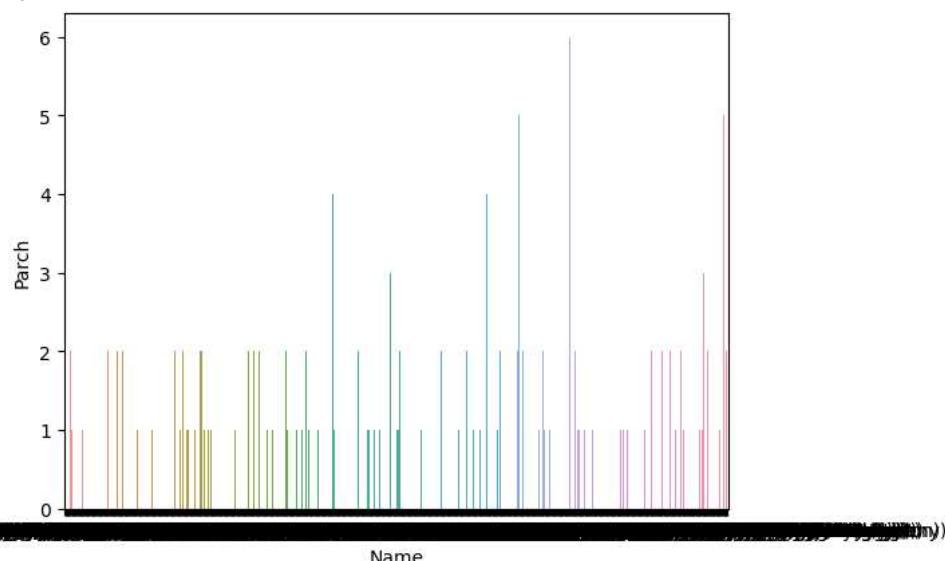


```
sns.barplot(x=df["Name"],y=df["Parch"],ci=0)
```

```
<ipython-input-74-f3ae2ce3cfca>:1: FutureWarning:
```

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

```
sns.barplot(x=df["Name"],y=df["Parch"],ci=0)
<Axes: xlabel='Name', ylabel='Parch'>
```

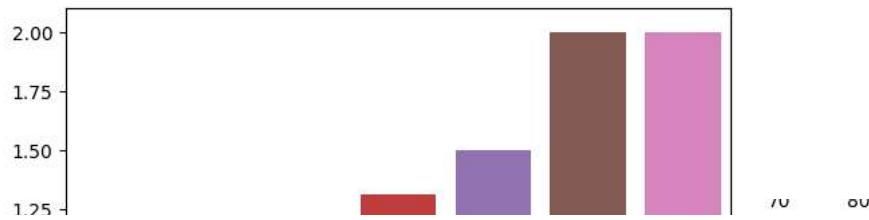


```
sns.barplot(x=df["Age"],y=df["Ticket"],ci=0)
```



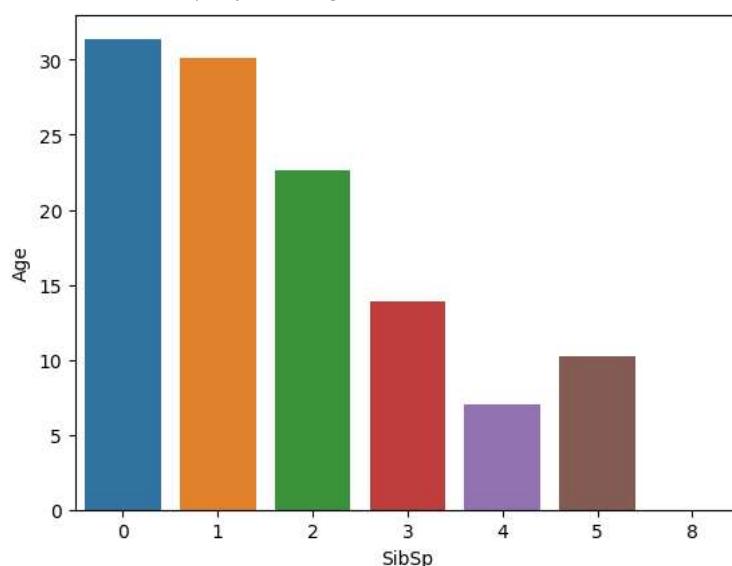
```
<ipython-input-76-9fce518e134>:1: FutureWarning:  
The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.
```

```
sns.barplot(x=df["SibSp"],y=df["Parch"],ci=0)  
<Axes: xlabel='SibSp', ylabel='Parch'>
```



```
sns.barplot(x=df["SibSp"],y=df["Age"],ci=0)
```

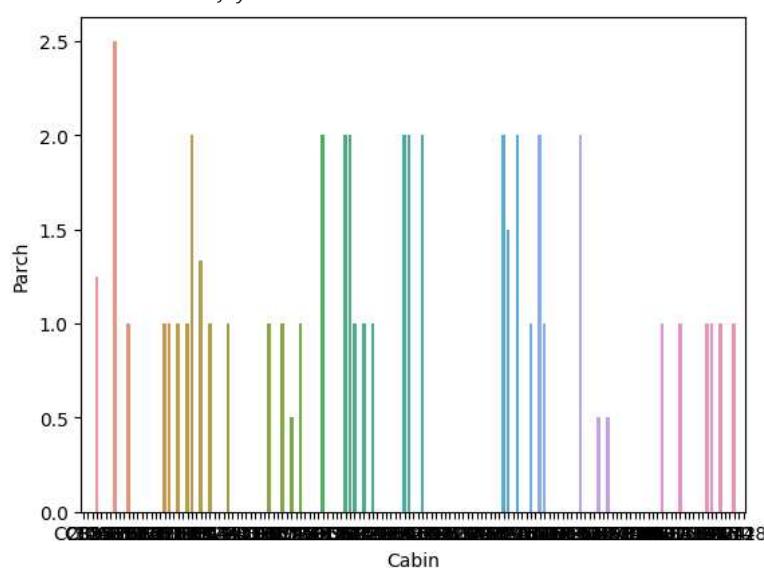
```
<ipython-input-77-316fd99dfc95>:1: FutureWarning:  
The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.  
  
sns.barplot(x=df["SibSp"],y=df["Age"],ci=0)  
/usr/local/lib/python3.10/dist-packages/seaborn/algorithms.py:98: RuntimeWarning: Mean of empty slice  
    boot_dist.append(f(*sample, **func_kwargs))  
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1559: RuntimeWarning: All-NaN slice encountered  
    r, k = function_base._ureduce(a,  
<Axes: xlabel='SibSp', ylabel='Age'>
```



```
sns.barplot(x=df["Cabin"],y=df["Parch"],ci=0)
```

```
<ipython-input-78-35df4013214f>:1: FutureWarning:  
The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.
```

```
sns.barplot(x=df["Cabin"],y=df["Parch"],ci=0)  
<Axes: xlabel='Cabin', ylabel='Parch'>
```



```
df.head()
```

	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	

```
from scipy import stats
z_scores = np.abs(stats.zscore(df["PassengerId"]))
outliers=df["PassengerId"][z_scores>3]
outliers
```

```
Series([], Name: PassengerId, dtype: int64)
```

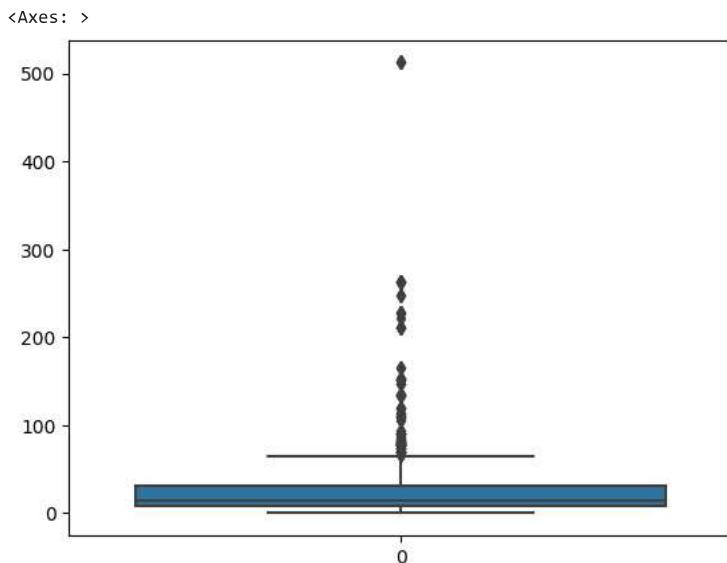
```
z_scores = np.abs(stats.zscore(df["Fare"]))
outliers=df["Fare"][z_scores>3]
outliers
```

```
27    263.0000
88    263.0000
118   247.5208
258   512.3292
299   247.5208
311   262.3750
341   263.0000
377   211.5000
380   227.5250
438   263.0000
527   221.7792
557   227.5250
679   512.3292
689   211.3375
700   227.5250
716   227.5250
730   211.3375
737   512.3292
742   262.3750
779   211.3375
Name: Fare, dtype: float64
```

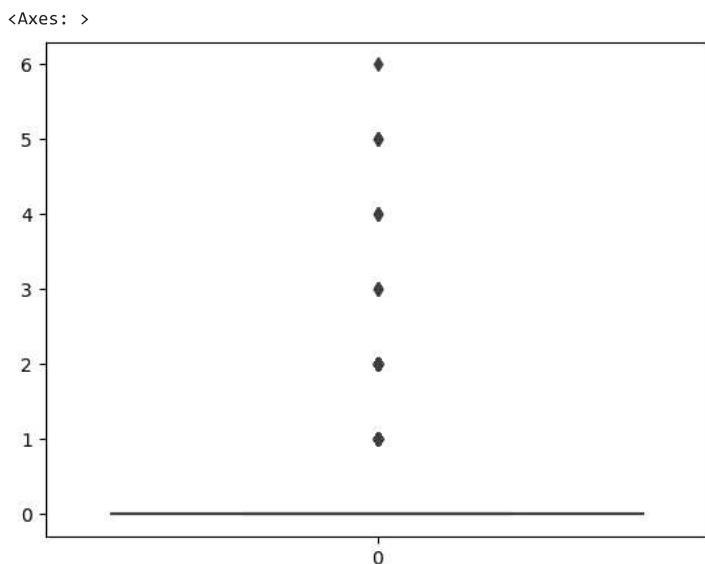
```
Q1=df["Fare"].quantile (0.25)
Q3=df["Fare"].quantile (0.75)
IQR = Q3-Q1
lower_bound = Q1-1.5* IQR
upper_bound = Q3 + 1.5 * IQR
df_cleaned = df[(df["Fare"]> lower_bound) & (df["Fare"] < upper_bound)]
print (f"Original DataFrame size: {df.shape}")
print (f"Cleaned DataFrame size: {df_cleaned.shape}")
df_cleaned
```

```
Original DataFrame size: (891, 12)
```

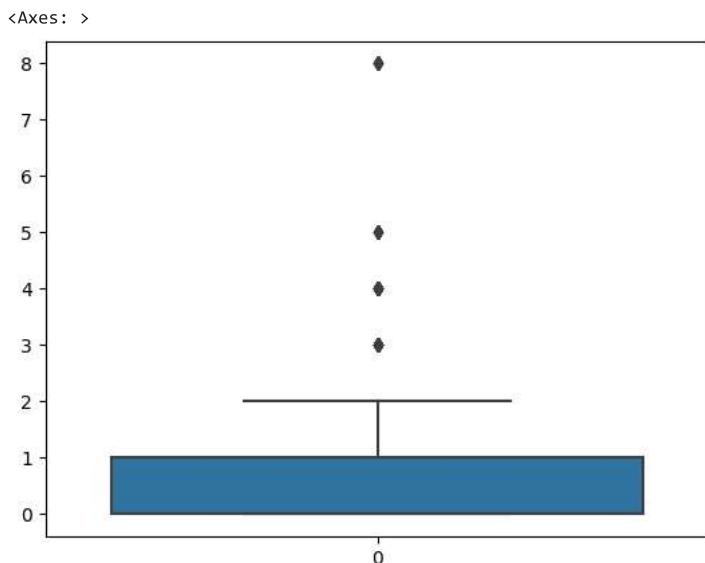
```
sns.boxplot(df["Fare"])
```



```
sns.boxplot(df["Parch"])
```



```
sns.boxplot(df["SibSp"])
```



```
df.head()
```

	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	

```
X=df.drop(columns=["PassengerId"],axis=1)
```

X

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

```
X=df.drop(columns=["Survived"],axis=1)
```

X

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

```
X=df.drop(columns=["Pclass"],axis=1)
```

X

PassengerId	Survived		Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	grid
0	1	0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	grid
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C	grid
2	3	1	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	grid
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	grid
4	5	0	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	grid
...	grid
886	887	0	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	grid

X=df.drop(columns=["Fare"],axis=1)

X

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

X=df.drop(columns=["Age"],axis=1)

X

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

X=df.drop(columns=["Sex"],axis=1)

X

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	NaN	S	!!
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	38.0	1	0	PC 17599	71.2833	C85	C	
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C123	S	
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	NaN	S	
...

X=df.drop(columns=["SibSp"],axis=1)

X

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

X=df.drop(columns=["Parch"],axis=1)

X

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

X=df.drop(columns=["Ticket"],axis=1)

X

PassengerId	Survived	Pclass		Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	female	38.0	1	0	71.2833	C85	C	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S	

X=df.drop(columns=["Cabin"],axis=1)

X

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

X=df.drop(columns=["Embarked"],axis=1)

X

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

X.shape

(891, 11)

type(X)

pandas.core.frame.DataFrame

y=df['PassengerId']
y.head()

```

0    1
1    2
2    3
3    4
4    5
Name: PassengerId, dtype: int64

```

```
y=df['Survived']
y.head()

0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

```
y=df['Pclass']
y.head()

0    3
1    1
2    3
3    1
4    3
Name: Pclass, dtype: int64
```

```
y=df['Fare']
y.head()

0    7.2500
1    71.2833
2    7.9250
3    53.1000
4    8.0500
Name: Fare, dtype: float64
```

```
y=df['Age']
y.head()

0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

```
y=df['Sex']
y.head()

0      male
1    female
2    female
3    female
4      male
Name: Sex, dtype: object
```

```
y=df['SibSp']
y.head()

0    1
1    1
2    0
3    1
4    0
Name: SibSp, dtype: int64
```

```
y=df['Cabin']
y.head()

0      NaN
1    C85
2      NaN
3    C123
4      NaN
Name: Cabin, dtype: object
```

```
y=df['Ticket']
y.head()

0          A/5 21171
1          PC 17599
```

```
2      STON/O2. 3101282
3                  113803
4                  373450
Name: Ticket, dtype: object
```

```
y=df['Name']
y.head()
```

```
0          Braund, Mr. Owen Harris
1  Cumings, Mrs. John Bradley (Florence Briggs Th...
2                  Heikkinen, Miss. Laina
3        Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                Allen, Mr. William Henry
Name: Name, dtype: object
```

```
X.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2.	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	3101282	53.1000	C123

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
X.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2.	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	3101282	53.1000	C123

```
X[["Survived"]]=le.fit_transform(X[["Survived"]])
X.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2.	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	3101282	53.1000	C123

```
print(le.classes_)
```

```
[0 1]
```

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

```
{0: 0, 1: 1}
```

```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
# Assuming 'X' is your original DataFrame
# Separate numeric and non-numeric columns
numeric_cols = X.select_dtypes(include=['number'])
```

```

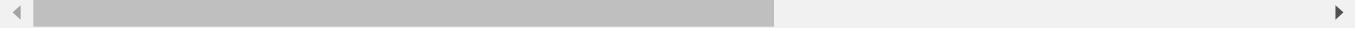
non_numeric_cols = X.select_dtypes(exclude=['number'])
# Preprocess non-numeric columns if needed
# For example, you can use one-hot encoding for categorical columns
non_numeric_cols_encoded = pd.get_dummies(non_numeric_cols)
# Concatenate numeric and processed non-numeric columns
X_processed = pd.concat([numeric_cols, non_numeric_cols_encoded], axis=1)
# Apply StandardScaler to the entire DataFrame
ms = StandardScaler()
X_scaled = pd.DataFrame(ms.fit_transform(X_processed), columns=X_processed.columns)

```

X_scaled

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Name_Abbing, Mr. Anthony	Name_Abbott, Mr. Rossmore Edward	Name_Abbott, Mrs. Stanton (Rosa Hunt)	...	Cabin_
0	-1.730108	-0.789272	0.827377	-0.530377	0.432793	-0.473674	-0.502445	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
1	-1.726220	1.266990	-1.566107	0.571831	0.432793	-0.473674	0.786845	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
2	-1.722332	1.266990	0.827377	-0.254825	-0.474545	-0.473674	-0.488854	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
3	-1.718444	1.266990	-1.566107	0.365167	0.432793	-0.473674	0.420730	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
4	-1.714556	-0.789272	0.827377	0.365167	-0.474545	-0.473674	-0.486337	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
...
886	1.714556	-0.789272	-0.369365	-0.185937	-0.474545	-0.473674	-0.386671	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
887	1.718444	1.266990	-1.566107	-0.737041	-0.474545	-0.473674	-0.044381	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
888	1.722332	-0.789272	0.827377	NaN	0.432793	2.008933	-0.176263	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
889	1.726220	1.266990	-1.566107	-0.254825	-0.474545	-0.473674	-0.044381	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474
890	1.730108	-0.789272	0.827377	0.158503	-0.474545	-0.473674	-0.492378	-0.03352	-0.03352	-0.03352	-0.03352	... -0.0474

891 rows × 1728 columns



```

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)

```

```
print(X_train.shape,X_test.shape,y_train.shape,y_test.shape)
```

```
(712, 11) (179, 11) (712,) (179,)
```