## Importing the libraries

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

## Importing thr dataset

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

dataset
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age
SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0
1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                               Heikkinen, Miss. Laina  female  26.0
0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                             Allen, Mr. William Henry    male  35.0
0
..                                                 ...     ...   ...
...
886                              Montvila, Rev. Juozas    male  27.0
0
887                       Graham, Miss. Margaret Edith  female  19.0
0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN
1
889                              Behr, Mr. Karl Howell    male  26.0
0
890                                Dooley, Mr. Patrick    male  32.0
0
```

```
     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
dataset.head(3)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
```

```
dataset.tail()
```

```
     PassengerId  Survived  Pclass
Name  \
886          887         0       2                     Montvila, Rev.
Juozas
887          888         1       1              Graham, Miss. Margaret
Edith
888          889         0       3  Johnston, Miss. Catherine Helen
"Carrie"
889          890         1       1                     Behr, Mr. Karl
Howell
890          891         0       3                       Dooley, Mr.
Patrick
```

```
        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked
886    male  27.0      0      0      211536  13.00   NaN        S
887  female  19.0      0      0      112053  30.00   B42        S
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S
889    male  26.0      0      0      111369  30.00  C148        C
890    male  32.0      0      0      370376   7.75   NaN        Q
```

```
dataset.shape
```

```
(891, 12)
```

```
dataset.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
```

```
dataset.describe()
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000   
mean    446.000000    0.383838    2.308642   29.699118    0.523008   
std     257.353842    0.486592    0.836071   14.526497    1.102743   
min       1.000000    0.000000    1.000000    0.420000    0.000000   
25%     223.500000    0.000000    2.000000   20.125000    0.000000   
50%     446.000000    0.000000    3.000000   28.000000    0.000000   
75%     668.500000    1.000000    3.000000   38.000000    1.000000   
max     891.000000    1.000000    3.000000   80.000000    8.000000   

            Parch        Fare  
count  891.000000  891.000000  
mean     0.381594   32.204208  
std      0.806057   49.693429  
min      0.000000    0.000000  
```

```
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```
corr=dataset.corr(numeric_only=True)
```

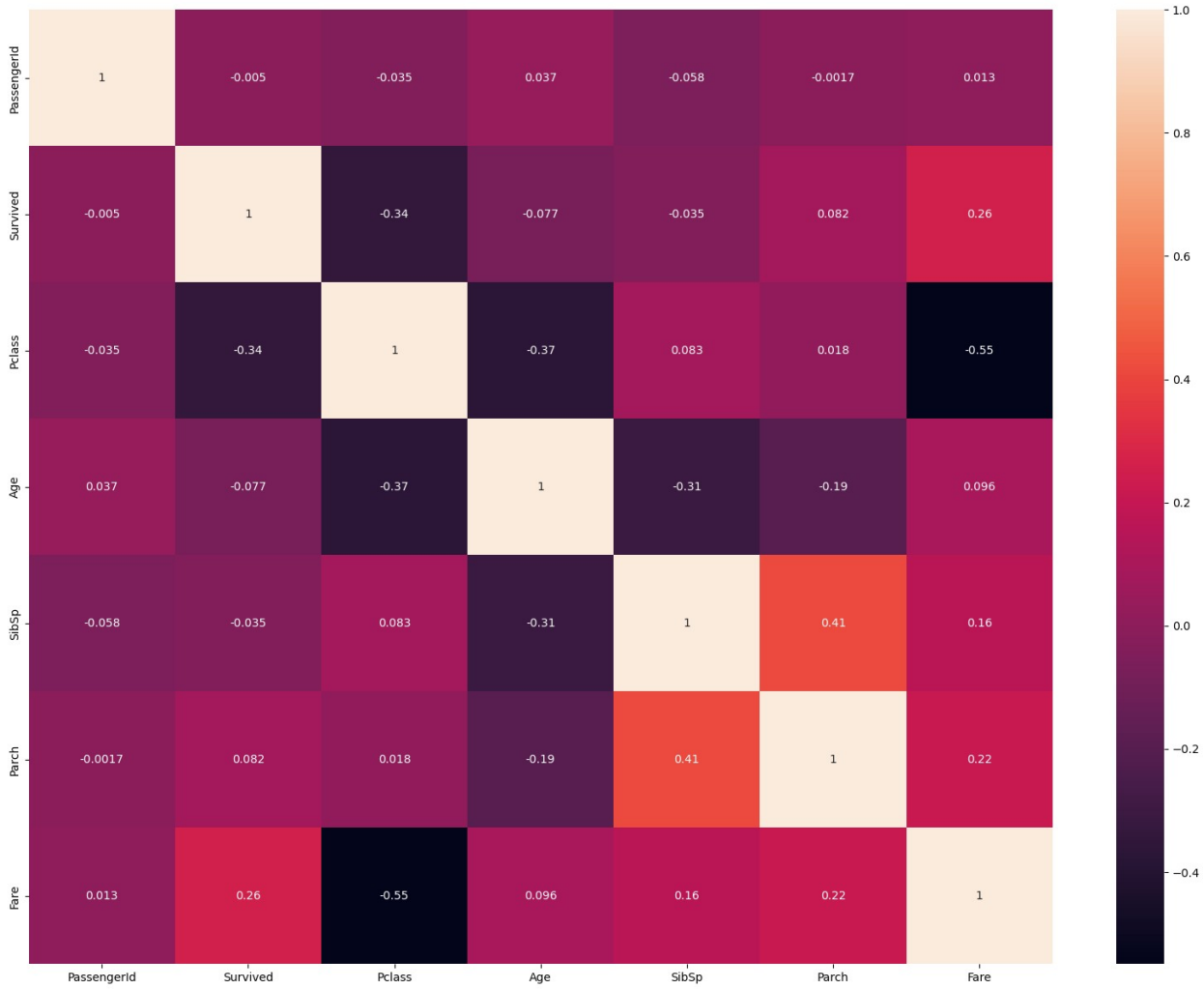
```
corr
```

```
             PassengerId  Survived    Pclass       Age     SibSp
Parch  \
PassengerId     1.000000 -0.005007 -0.035144  0.036847 -0.057527 -
0.001652
Survived       -0.005007  1.000000 -0.338481 -0.077221 -0.035322
0.081629
Pclass         -0.035144 -0.338481  1.000000 -0.369226  0.083081
0.018443
Age             0.036847 -0.077221 -0.369226  1.000000 -0.308247 -
0.189119
SibSp          -0.057527 -0.035322  0.083081 -0.308247  1.000000
0.414838
Parch          -0.001652  0.081629  0.018443 -0.189119  0.414838
1.000000
Fare            0.012658  0.257307 -0.549500  0.096067  0.159651
0.216225

                 Fare
PassengerId  0.012658
Survived     0.257307
Pclass      -0.549500
Age          0.096067
SibSp        0.159651
Parch        0.216225
Fare         1.000000
```

```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

```
<Axes: >
```




## Handling Null Values

```
dataset.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
```

```
dataset.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
```

```
dataset["Age"].fillna(dataset["Age"].mean(),inplace=True)

dataset["Cabin"].fillna(dataset["Cabin"].mode()[0],inplace=True)

dataset.isnull().sum() ## no null values now
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       2
dtype: int64
```

```
dataset.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
```

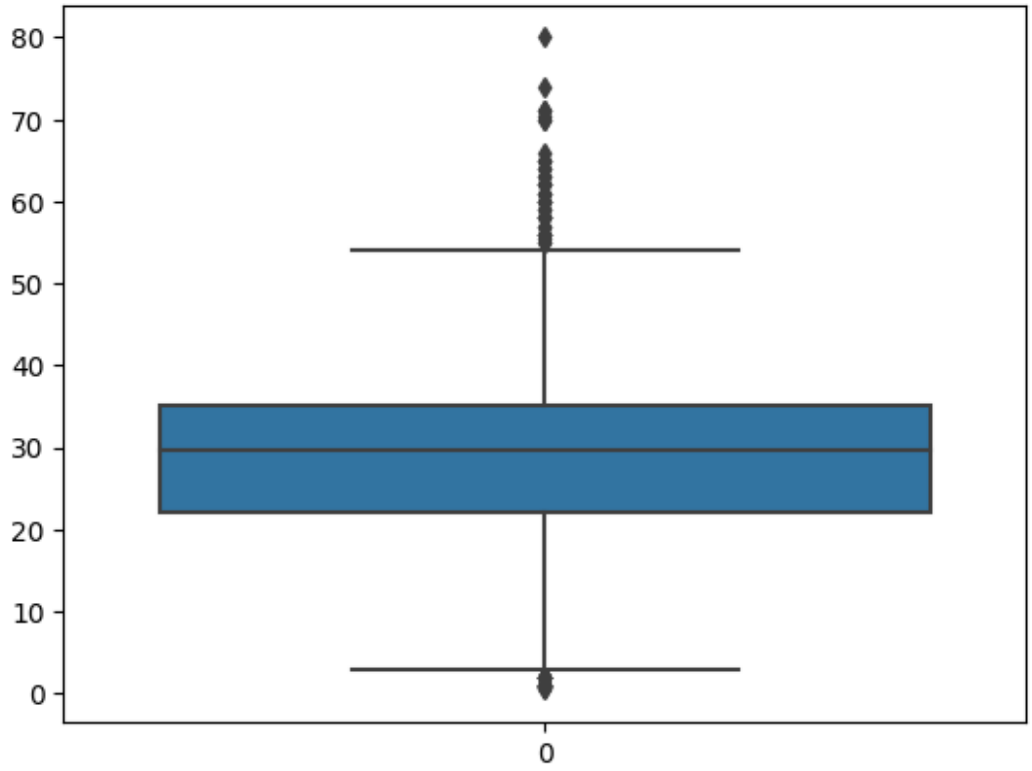
```
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                            Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare    Cabin Embarked
0      0         A/5 21171   7.2500  B96 B98        S
1      0          PC 17599  71.2833      C85        C
2      0  STON/O2. 3101282   7.9250  B96 B98        S
3      0            113803  53.1000     C123        S
4      0            373450   8.0500  B96 B98        S
```

## Outliers

```
sns.boxplot(dataset.Age)
```

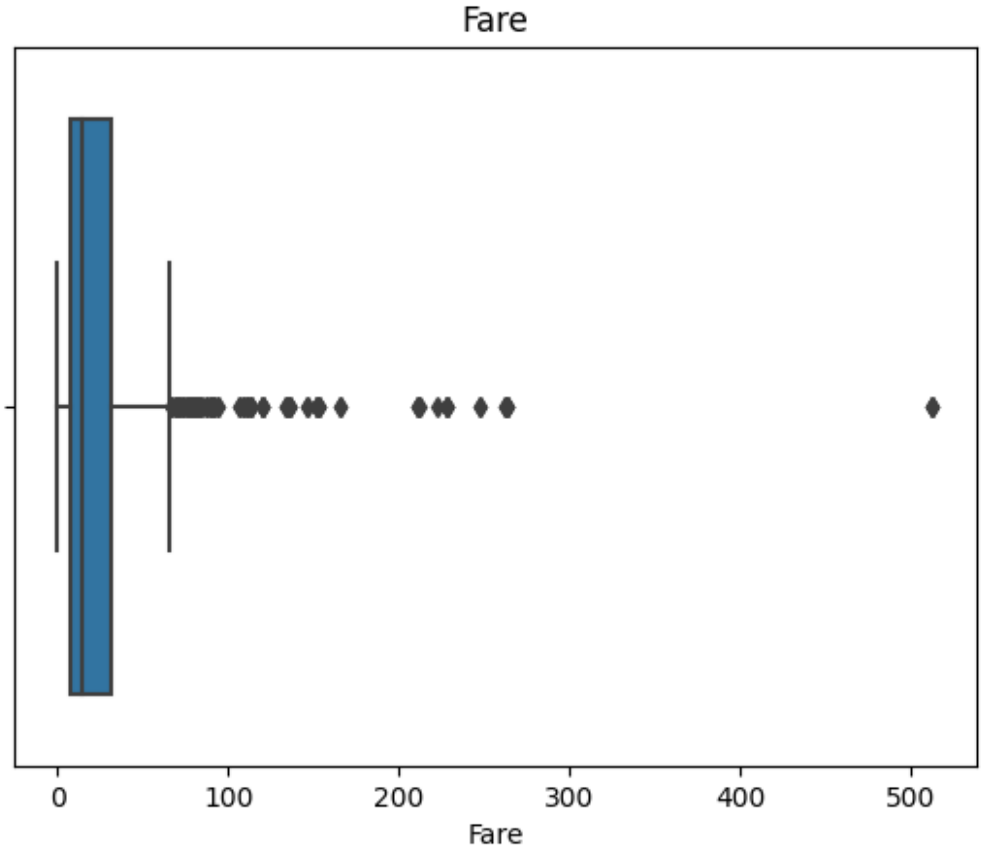
```
<Axes: >
```




```
sns.boxplot(data=dataset, x='Fare')
plt.title("Fare")
plt.show()
```

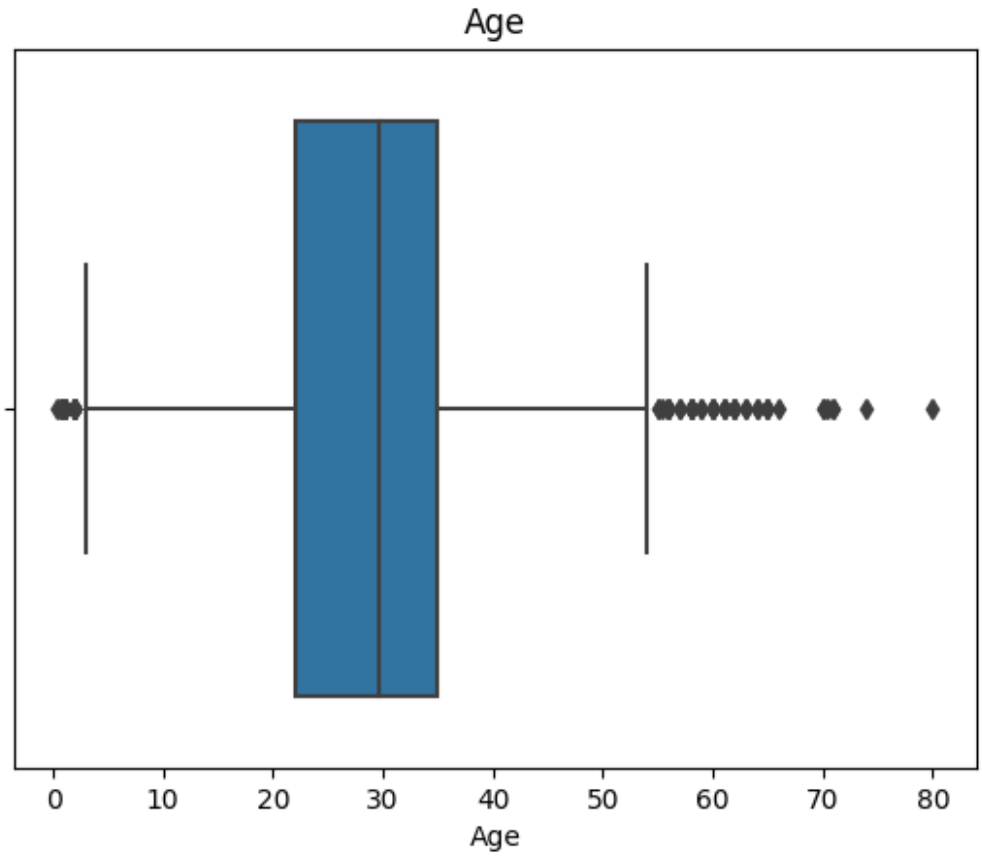
```
C:\Users\sahit\AppData\Local\Programs\Python\Python311\Lib\site-
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype
is deprecated and will be removed in a future version. Use
```

```
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```




```
sns.boxplot(data=dataset, x='Age')
plt.title("Age")
plt.show()
```

```
C:\Users\sahit\AppData\Local\Programs\Python\Python311\Lib\site-
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype
is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```




```
from scipy import stats

z_scores = np.abs(stats.zscore(dataset['Age']))
max_threshold=3
outliers = dataset['Age'][z_scores > max_threshold]
# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
96     71.0
116    70.5
493    71.0
630    80.0
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

```
z_scores = np.abs(stats.zscore(dataset['Fare']))
max_threshold=3
outliers = dataset['Fare'][z_scores > max_threshold]
# Print and visualize the outliers
```

```
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
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
```

## Separate dependent and Independent Variables

```
x=dataset.iloc[:,3:13]
y=dataset.iloc[:,13:14]

x.head()
```

```
   Pclass                                                Name     Sex
Age  \
0       3                             Braund, Mr. Owen Harris    male
22.0
1       1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0
2       3                              Heikkinen, Miss. Laina  female
26.0
3       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
35.0
4       3                            Allen, Mr. William Henry    male
35.0

   SibSp  Parch            Ticket     Fare    Cabin Embarked
0      1      0         A/5 21171   7.2500  B96 B98        S
1      1      0          PC 17599  71.2833      C85        C
2      0      0  STON/O2. 3101282   7.9250  B96 B98        S
```

```
3      1      0            113803  53.1000     C123        S
4      0      0            373450   8.0500  B96 B98        S
```

```
y.head()
```

```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4]
```

```
dataset.shape
```

```
(891, 12)
```

```
x.shape
```

```
(891, 9)
```

```
y.shape
```

```
(891, 0)
```

## Encoding

```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

x["Sex"]=le.fit_transform(x["Sex"])

x["Sex"]
```

```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int32
```

```
x["Sex"].value_counts()
```

```
Sex
1    577
0    314
Name: count, dtype: int64
```

```
x["Sex"].nunique()
```

```
2
```

```
x.head()
```

```
                                                Name  Sex   Age  SibSp
Parch  \
0                            Braund, Mr. Owen Harris    1  22.0      1
0
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0  38.0      1
0
2                             Heikkinen, Miss. Laina    0  26.0      0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  35.0      1
0
4                           Allen, Mr. William Henry    1  35.0      0
0

             Ticket     Fare    Cabin Embarked
0         A/5 21171   7.2500  B96 B98        S
1          PC 17599  71.2833      C85        C
2  STON/O2. 3101282   7.9250  B96 B98        S
3            113803  53.1000     C123        S
4            373450   8.0500  B96 B98        S
```

One hot encoding on Embarked column

```
x.shape
```

```
(891, 9)
```

```
Embarked=pd.get_dummies(x["Embarked"],drop_first=True)
```

```
Embarked
```

```
         Q      S
0    False   True
1    False  False
2    False   True
3    False   True
4    False   True
..     ...    ...
886  False   True
887  False   True
888  False   True
889  False  False
890   True  False

[891 rows x 2 columns]
```

```
#concat
x=pd.concat([x,Embarked],axis=1)
```

```
x.head()
```

```
                                                Name  Sex   Age  SibSp
Parch  \
0                            Braund, Mr. Owen Harris    1  22.0      1
0
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0  38.0      1
0
2                             Heikkinen, Miss. Laina    0  26.0      0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  35.0      1
0
4                           Allen, Mr. William Henry    1  35.0      0
0

             Ticket     Fare    Cabin Embarked      Q      S
0         A/5 21171   7.2500  B96 B98        S  False   True
1          PC 17599  71.2833      C85        C  False  False
2  STON/O2. 3101282   7.9250  B96 B98        S  False   True
3            113803  53.1000     C123        S  False   True
4            373450   8.0500  B96 B98        S  False   True
```

```
#dropping Embarked  column
x.drop(["Embarked"],axis=1,inplace=True)
```

```
x.head(10)
```

```
                                                Name  Sex        Age
SibSp  \
0                            Braund, Mr. Owen Harris    1  22.000000
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0  38.000000
1
2                             Heikkinen, Miss. Laina    0  26.000000
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  35.000000
1
4                           Allen, Mr. William Henry    1  35.000000
0
5                                 Moran, Mr. James    1  29.699118
0
6                           McCarthy, Mr. Timothy J    1  54.000000
0
7                    Palsson, Master. Gosta Leonard    1   2.000000
3
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)    0  27.000000
0
9                Nasser, Mrs. Nicholas (Adele Achem)    0  14.000000
1
```

```
   Parch            Ticket     Fare    Cabin      Q      S
0      0         A/5 21171   7.2500  B96 B98  False   True
1      0          PC 17599  71.2833      C85  False  False
2      0  STON/O2. 3101282   7.9250  B96 B98  False   True
3      0            113803  53.1000     C123  False   True
4      0            373450   8.0500  B96 B98  False   True
5      0            330877   8.4583  B96 B98   True  False
6      0             17463  51.8625      E46  False   True
7      1            349909  21.0750  B96 B98  False   True
8      2            347742  11.1333  B96 B98  False   True
9      0            237736  30.0708  B96 B98  False  False
```

```
x.shape
```

```
(891, 10)
```

## Splitting Data into Train and Test

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,rando
m_state=0)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((623, 10), (268, 10), (623, 0), (268, 0))
```

```
x_train
```

```
                                                 Name  Sex        Age
SibSp  \
857                            Daly, Mr. Peter Denis     1  51.000000
0
52          Harper, Mrs. Henry Sleeper (Myna Haxtun)    0  49.000000
1
386                  Goodwin, Master. Sidney Leonard    1   1.000000
5
124                      White, Mr. Percival Wayland    1  54.000000
0
578                 Caram, Mrs. Joseph (Maria Elias)    0  29.699118
1
..                                                ...  ...        ...
...
835                      Compton, Miss. Sara Rebecca    0  39.000000
1
192  Andersen-Jensen, Miss. Carla Christine Nielsine    0  19.000000
1
629                          O'Connell, Mr. Patrick D    1  29.699118
0
559     de Messemaeker, Mrs. Guillaume Joseph (Emma)    0  36.000000
1
684                 Brown, Mr. Thomas William Solomon    1  60.000000
```

```
1

     Parch    Ticket     Fare    Cabin      Q      S
857      0    113055  26.5500      E17  False   True
52       0  PC 17572  76.7292      D33  False  False
386      2   CA 2144  46.9000  B96 B98  False   True
124      1     35281  77.2875      D26  False   True
578      0      2689  14.4583  B96 B98  False  False
..     ...       ...      ...      ...    ...    ...
835      1  PC 17756  83.1583      E49  False  False
192      0    350046   7.8542  B96 B98  False   True
629      0    334912   7.7333  B96 B98   True  False
559      0    345572  17.4000  B96 B98  False   True
684      1     29750  39.0000  B96 B98  False   True

[623 rows x 10 columns]

x_test

                                                  Name  Sex        Age
SibSp  \
495                            Yousseff, Mr. Gerious    1  29.699118
0
648                               Willey, Mr. Edward    1  29.699118
0
278                               Rice, Master. Eric    1   7.000000
4
31   Spencer, Mrs. William Augustus (Marie Eugenie)    0  29.699118
1
255         Touma, Mrs. Darwis (Hanne Youssef Razi)    0  29.000000
0
..                                                ...  ...        ...
...
263                            Harrison, Mr. William    1  40.000000
0
718                              McEvoy, Mr. Michael    1  29.699118
0
620                              Yasbeck, Mr. Antoni    1  27.000000
1
786                         Sjoblom, Miss. Anna Sofia    0  18.000000
0
64                             Stewart, Mr. Albert A    1  29.699118
0

     Parch         Ticket      Fare    Cabin      Q      S
495      0           2627   14.4583  B96 B98  False  False
648      0  S.O./P.P. 751    7.5500  B96 B98  False   True
278      1         382652   29.1250  B96 B98   True  False
31       0       PC 17569  146.5208      B78  False  False
255      2           2650   15.2458  B96 B98  False  False
```

```
..     ...            ...       ...      ...    ...    ...
263      0         112059    0.0000      B94  False   True
718      0          36568   15.5000  B96 B98   True  False
620      0           2659   14.4542  B96 B98  False  False
786      0        3101265    7.4958  B96 B98  False   True
64       0       PC 17605   27.7208  B96 B98  False  False

[268 rows x 10 columns]
```

```
a=[1,2,3,4,5,6]
b=[1,0,1,5,6,3]

for i in range(5):

a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3,rando
m_state=100)
    print("with random state",a_train)
```

```
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
```

```
y_train
```

```
Empty DataFrame
Columns: []
Index: [857, 52, 386, 124, 578, 549, 118, 12, 157, 127, 653, 235, 785,
241, 351, 862, 851, 753, 532, 485, 695, 475, 17, 476, 533, 416, 345,
242, 344, 170, 187, 800, 457, 652, 451, 78, 889, 198, 492, 813, 526,
870, 21, 885, 799, 250, 243, 701, 35, 81, 159, 744, 524, 109, 337,
443, 92, 364, 434, 465, 731, 876, 211, 811, 165, 238, 188, 471, 553,
456, 366, 592, 738, 155, 391, 886, 724, 453, 66, 841, 408, 462, 268,
161, 363, 406, 866, 881, 618, 100, 722, 678, 229, 334, 558, 669, 807,
520, 816, 220, ...]

[623 rows x 0 columns]
```

```
y_test
```

```
Empty DataFrame
Columns: []
Index: [495, 648, 278, 31, 255, 298, 609, 318, 484, 367, 704, 346,
196, 535, 310, 14, 350, 145, 614, 803, 144, 708, 778, 270, 474, 319,
519, 141, 880, 642, 285, 458, 200, 55, 477, 632, 818, 883, 666, 317,
587, 693, 538, 686, 230, 101, 656, 311, 808, 262, 740, 97, 750, 566,
839, 655, 252, 542, 819, 301, 60, 567, 496, 766, 8, 890, 796, 848,
530, 887, 316, 712, 34, 77, 676, 726, 27, 30, 202, 181, 768, 489, 312,
764, 397, 627, 380, 483, 516, 505, 815, 877, 193, 523, 634, 531, 247,
266, 694, 681, ...]
```

```
[268 rows x 0 columns]
```

```
a=[1,2,3,4,5,6]    # 4 values for training and 2 for testing
b=[1,0,1,5,6,3]

for i in range(5):
    a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3)
    print("without random state",a_train)
```

```
without random state [4, 5, 1, 2]
without random state [1, 6, 3, 2]
without random state [2, 1, 3, 5]
without random state [2, 5, 4, 6]
without random state [4, 2, 5, 3]
```

## Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()

scale = StandardScaler()
x[['Age', 'Fare']] = scale.fit_transform(x[['Age', 'Fare']])

x.head()
```

```
                                                Name  Sex       Age
SibSp  \
0                            Braund, Mr. Owen Harris    1 -0.592481
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0  0.638789
1
2                             Heikkinen, Miss. Laina    0 -0.284663
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  0.407926
1
4                           Allen, Mr. William Henry    1  0.407926
0

   Parch            Ticket      Fare    Cabin      Q      S
0      0         A/5 21171 -0.502445  B96 B98  False   True
1      0          PC 17599  0.786845      C85  False  False
2      0  STON/O2. 3101282 -0.488854  B96 B98  False   True
3      0            113803  0.420730     C123  False   True
4      0            373450 -0.486337  B96 B98  False   True
```