Name: Vuddandi Rishmitha
Reg No: 21BCE9684
Mobile No: 8143953608
Campus: VIT-AP

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

data=pd.read_csv("Titanic-Dataset.csv")

data.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   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
```

```
data.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

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

data.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
```

Handling Null Values

```
data.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
```

```
data.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
```

```
mean=data["Age"].mean()
```

Filling the null values in Age column with Mean

```
data["Age"]=data["Age"].fillna(mean)

data["Age"].tail()
```

```
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, dtype: float64
```

```
data["Age"].isnull().sum()
```

```
0
```

```
Em_mode=data["Embarked"].mode()
```

```
data["Embarked"]=data["Embarked"].fillna(Em_mode[0])
```

```
data["Embarked"].isnull().sum()
```

```
0
```

Filling the null values in Cabin with mode

```
Cabin_mode=data["Cabin"].mode()
```

```
data["Cabin"]
```

```
0       NaN
1       C85
2       NaN
3      C123
4       NaN
       ...
886     NaN
887     B42
888     NaN
889    C148
890     NaN
Name: Cabin, Length: 891, dtype: object
```

```
Cabin_mode
```

```
0        B96 B98
1    C23 C25 C27
2             G6
Name: Cabin, dtype: object
```

```
data["Cabin"]=data["Cabin"].fillna(Cabin_mode[2])
```

```
data["Cabin"].isnull().sum()
```

```
0
```

```
data["Cabin"]
```

```
0         G6
1        C85
2         G6
3       C123
4         G6
        ...
886       G6
887      B42
888       G6
889     C148
890       G6
Name: Cabin, Length: 891, dtype: object
```

```
data.isnull().sum()
```

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

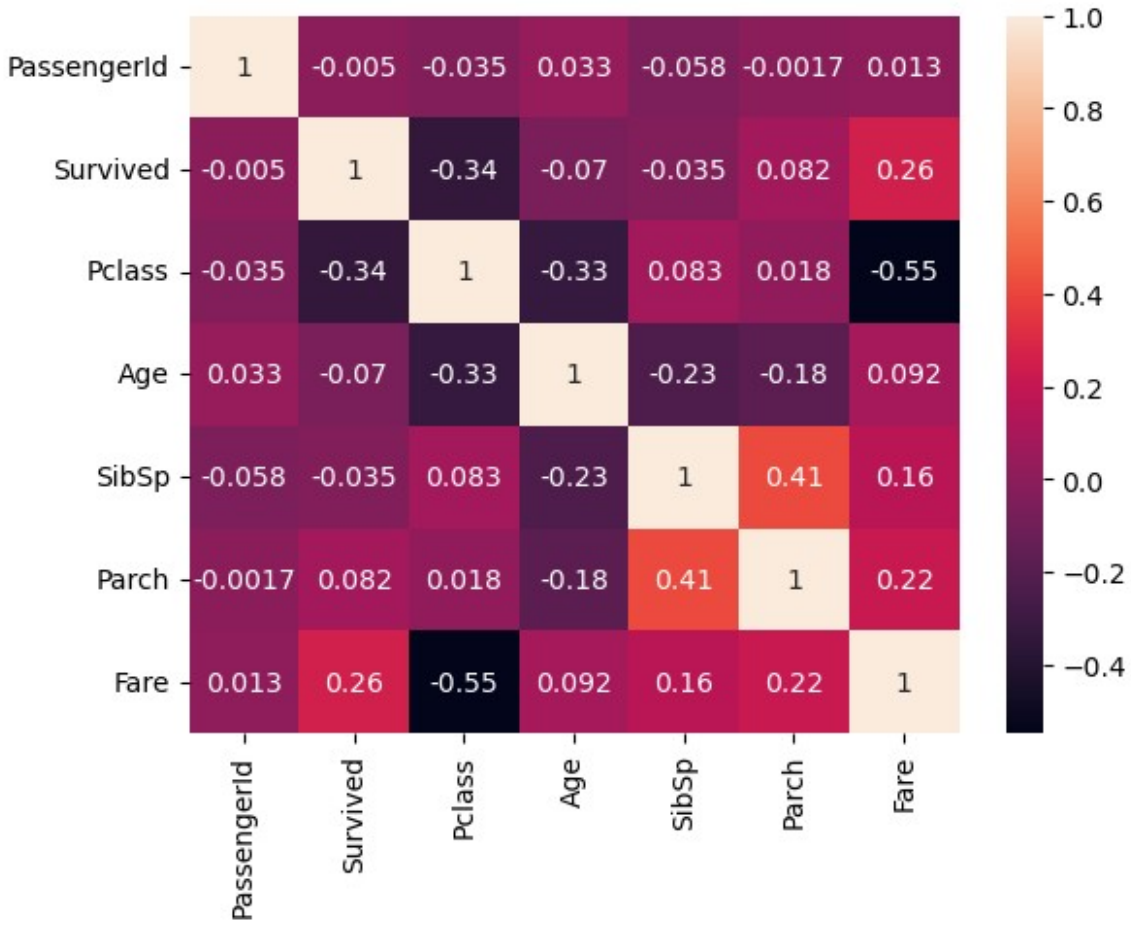
Data Visualisation

```
cor=data.corr()
```

```
C:\Users\vudda\AppData\Local\Temp\ipykernel_6808\1426905697.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  cor=data.corr()
```

```
sns.heatmap(cor,annot=True)
```

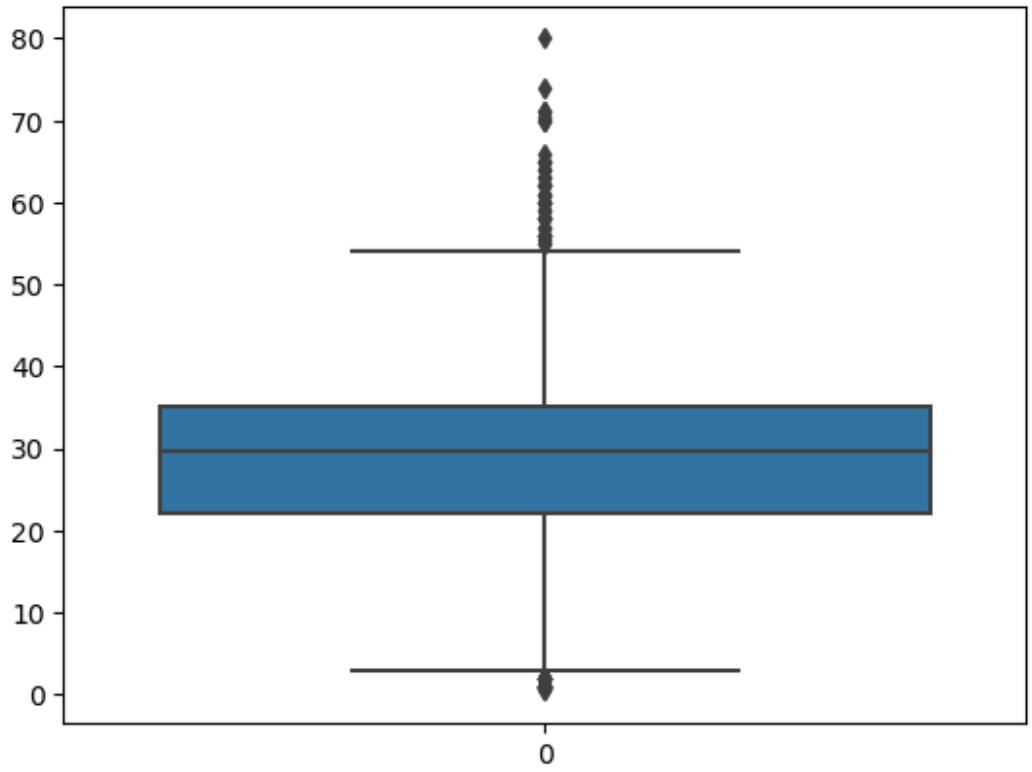
```
<Axes: >
```



Handling the Outliers

```
sns.boxplot(data["Age"])
```

```
<Axes: >
```



Outliers

```
Age_q1 = data.Age.quantile(0.25)
Age_q3 = data.Age.quantile(0.75)
print(Age_q1)
print(Age_q3)
```

```
22.0
35.0
```

```
IQR_Age=Age_q3-Age_q1
IQR_Age
```

```
13.0
```

```
upperlimit_Age=Age_q3+1.5*IQR_Age
upperlimit_Age
```

```
54.5
```

```
lower_limit_Age = Age_q1-1.5*IQR_Age
lower_limit_Age
```

```
2.5
```

```
median_Age=data["Age"].median()
median_Age
```

```
29.69911764705882
```

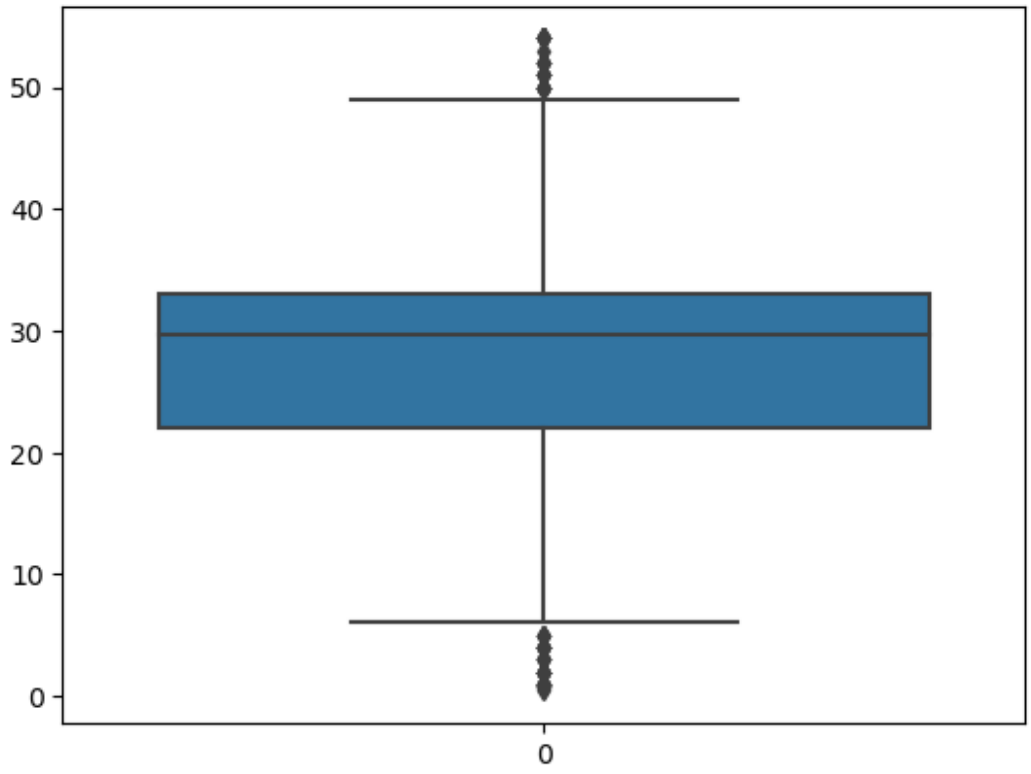
```
data["Age"]=np.where(data["Age"]>upperlimit_Age,median_Age,data["Age"]
)
```

```
(data["Age"]>54.5).sum()
```

```
0
```

```
sns.boxplot(data["Age"])
```

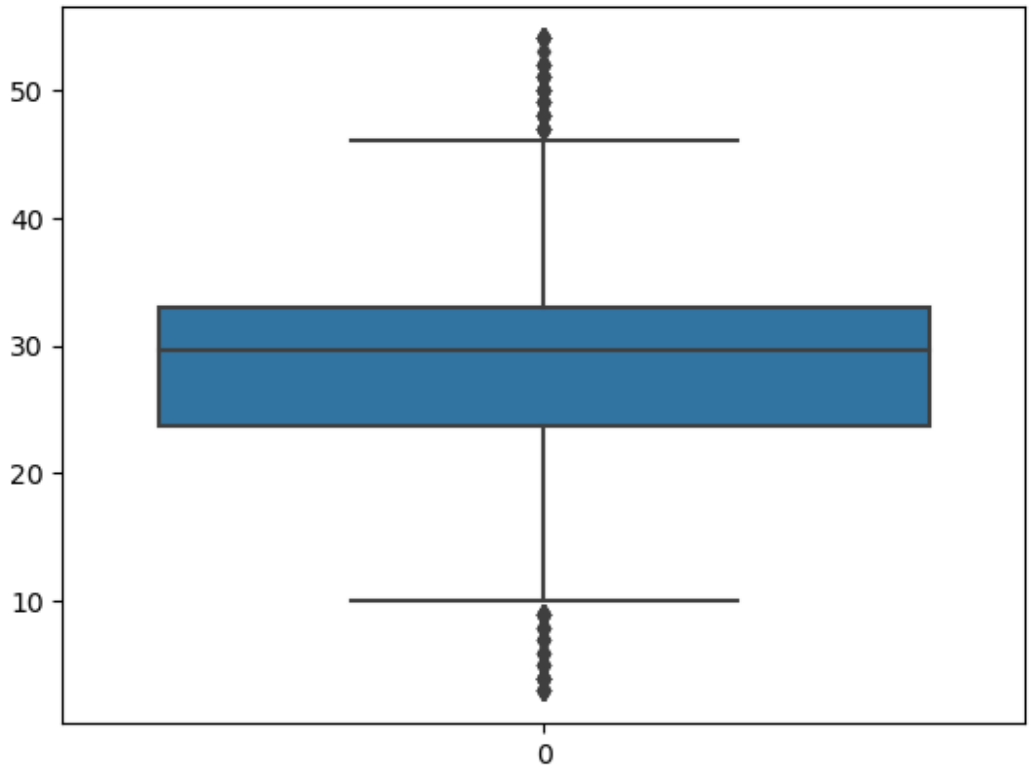
```
<Axes: >
```




```
data["Age"]=np.where(data["Age"]<lower_limit_Age,median_Age,data["Age"
])
```

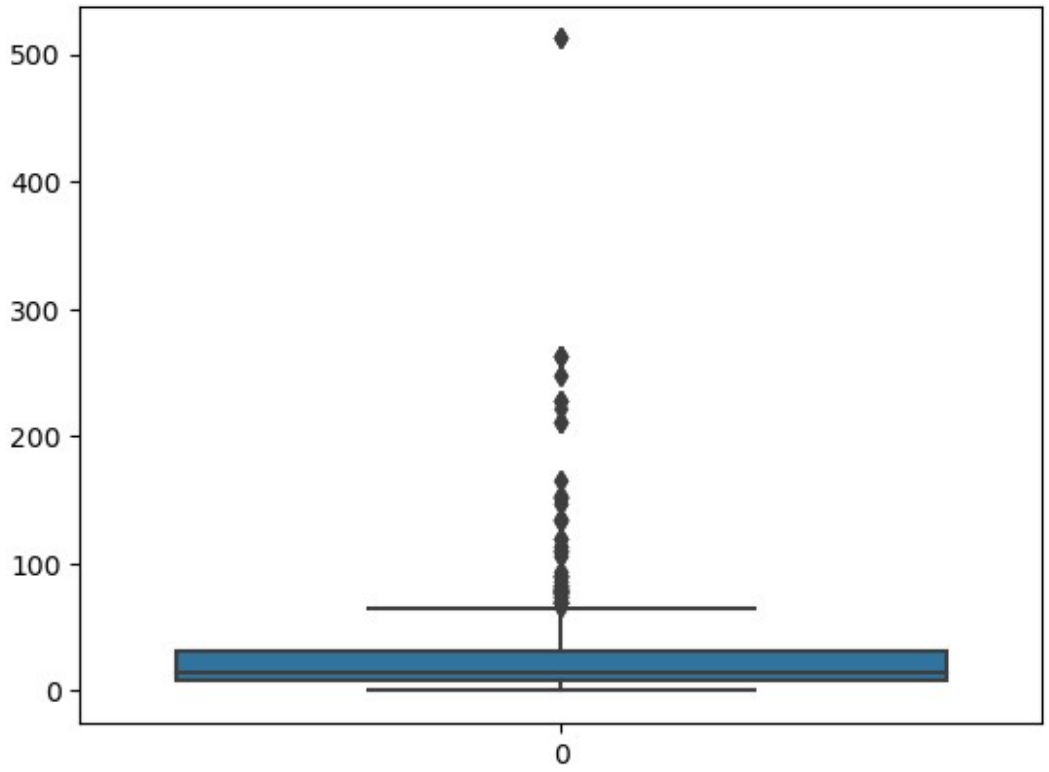
```
sns.boxplot(data["Age"])
```

```
<Axes: >
```




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

```
<Axes: >
```




```
Fare_q1 = data.Fare.quantile(0.25)
Fare_q3 = data.Fare.quantile(0.75)
print(Fare_q1)
print(Fare_q3)
```

```
7.9104
31.0
```

```
IQR_Fare=Fare_q3-Fare_q1
IQR_Fare
```

```
23.0896
```

```
upperlimit_Fare=Fare_q3+1.5*IQR_Fare
upperlimit_Fare
```

```
65.6344
```

```
lower_limit_Fare = Fare_q1-1.5*IQR_Fare
lower_limit_Fare
```

```
-26.724
```

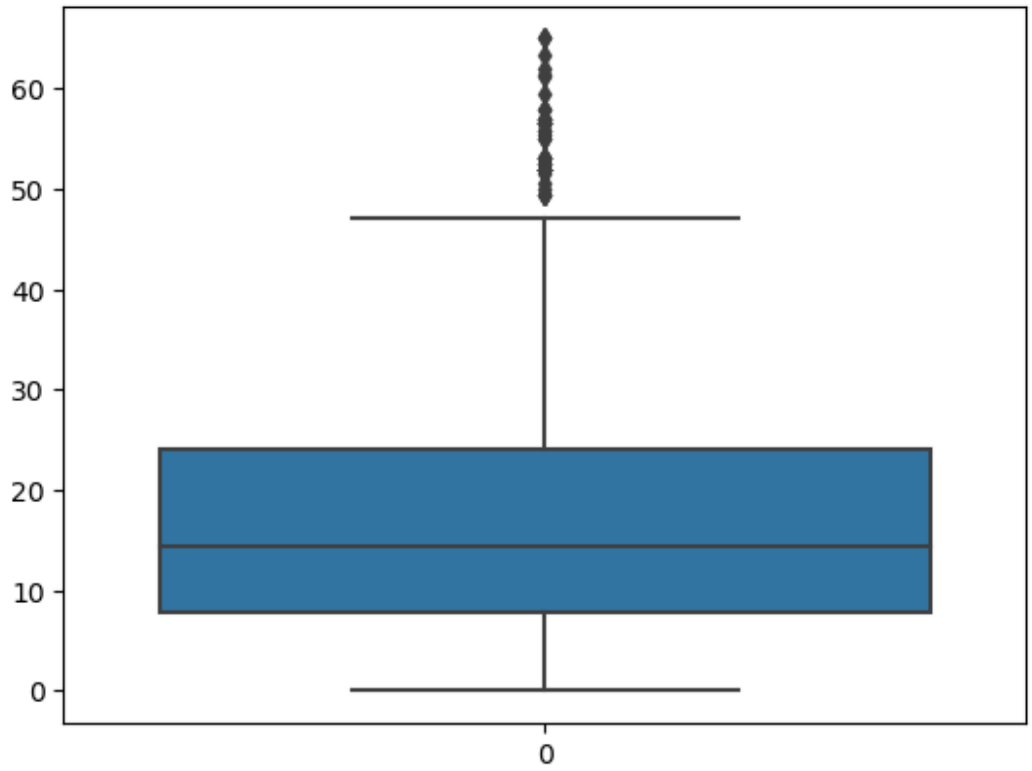
```
median_Fare=data["Fare"].median()
median_Fare
```

```
14.4542
```

```
data['Fare'] = np.where(
 (data['Fare'] > upperlimit_Fare),
 median_Fare,
 data['Fare']
)

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

```
<Axes: >
```




```
(data["Fare"]>65).sum()
```

```
0
```

Dropping the Variables

```
data.drop(['Name'],axis=1,inplace=True)
data
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[47], line 1
```

```
----> 1 data.drop(['Name'],axis=1,inplace=True)
      2 data

File ~\anaconda3\Lib\site-packages\pandas\util\_decorators.py:331, in
deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*arg
s, **kwargs)
    325 if len(args) > num_allow_args:
    326     warnings.warn(
    327
msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:5399, in
DataFrame.drop(self, labels, axis, index, columns, level, inplace,
errors)
   5251 @deprecate_nonkeyword_arguments(version=None,
allowed_args=["self", "labels"])
   5252 def drop(  # type: ignore[override]
   5253     self,
   (...)
   5260     errors: IgnoreRaise = "raise",
   5261 ) -> DataFrame | None:
   5262     """
   5263     Drop specified labels from rows or columns.
   5264
   (...)
   5397             weight  1.0     0.8
   5398     """
-> 5399     return super().drop(
   5400         labels=labels,
   5401         axis=axis,
   5402         index=index,
   5403         columns=columns,
   5404         level=level,
   5405         inplace=inplace,
   5406         errors=errors,
   5407     )

File ~\anaconda3\Lib\site-packages\pandas\util\_decorators.py:331, in
deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*arg
s, **kwargs)
    325 if len(args) > num_allow_args:
    326     warnings.warn(
    327
msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
```

```
    330     )
--> 331 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4505, in
NDFrame.drop(self, labels, axis, index, columns, level, inplace,
errors)
   4503 for axis, labels in axes.items():
   4504     if labels is not None:
-> 4505         obj = obj._drop_axis(labels, axis, level=level,
errors=errors)
   4507 if inplace:
   4508     self._update_inplace(obj)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4546, in
NDFrame._drop_axis(self, labels, axis, level, errors, only_slice)
   4544         new_axis = axis.drop(labels, level=level,
errors=errors)
   4545     else:
-> 4546         new_axis = axis.drop(labels, errors=errors)
   4547     indexer = axis.get_indexer(new_axis)
   4549 # Case for non-unique axis
   4550 else:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6934,
in Index.drop(self, labels, errors)
   6932 if mask.any():
   6933     if errors != "ignore":
-> 6934         raise KeyError(f"{list(labels[mask])} not found in
axis")
   6935     indexer = indexer[~mask]
   6936 return self.delete(indexer)

KeyError: "['Name'] not found in axis"
```

data

```
     PassengerId  Survived  Pclass     Sex        Age  SibSp  Parch  \
0              1         0       3    male  22.000000      1      0
1              2         1       1  female  38.000000      1      0
2              3         1       3  female  26.000000      0      0
3              4         1       1  female  35.000000      1      0
4              5         0       3    male  35.000000      0      0
..           ...       ...     ...     ...        ...    ...    ...
886          887         0       2    male  27.000000      0      0
887          888         1       1  female  19.000000      0      0
888          889         0       3  female  29.699118      1      2
889          890         1       1    male  26.000000      0      0
890          891         0       3    male  32.000000      0      0

               Ticket     Fare Cabin Embarked
```

```
0           A/5 21171   7.2500    G6        S
1            PC 17599  14.4542   C85        C
2    STON/O2. 3101282   7.9250    G6        S
3              113803  53.1000  C123        S
4              373450   8.0500    G6        S
..                ...      ...   ...      ...
886            211536  13.0000    G6        S
887            112053  30.0000   B42        S
888        W./C. 6607  23.4500    G6        S
889            111369  30.0000  C148        C
890            370376   7.7500    G6        Q

[891 rows x 11 columns]
```

```
data.drop(['Ticket'],axis=1,inplace=True)
```

```
data
```

```
     PassengerId  Survived  Pclass     Sex        Age  SibSp  Parch
Fare  \
0              1         0       3    male  22.000000      1      0
7.2500
1              2         1       1  female  38.000000      1      0
14.4542
2              3         1       3  female  26.000000      0      0
7.9250
3              4         1       1  female  35.000000      1      0
53.1000
4              5         0       3    male  35.000000      0      0
8.0500
..           ...       ...     ...     ...        ...    ...    ...
...
886          887         0       2    male  27.000000      0      0
13.0000
887          888         1       1  female  19.000000      0      0
30.0000
888          889         0       3  female  29.699118      1      2
23.4500
889          890         1       1    male  26.000000      0      0
30.0000
890          891         0       3    male  32.000000      0      0
7.7500

    Cabin Embarked
0      G6        S
1     C85        C
2      G6        S
3    C123        S
4      G6        S
..    ...      ...
```

```
886    G6        S
887   B42        S
888    G6        S
889  C148        C
890    G6        Q

[891 rows x 10 columns]
```

```
data.drop(["PassengerId"],axis=1,inplace=True)
```

```
data
```

```
     Survived  Pclass     Sex        Age  SibSp  Parch     Fare Cabin
Embarked
0           0       3    male  22.000000      1      0   7.2500    G6
S
1           1       1  female  38.000000      1      0  14.4542   C85
C
2           1       3  female  26.000000      0      0   7.9250    G6
S
3           1       1  female  35.000000      1      0  53.1000  C123
S
4           0       3    male  35.000000      0      0   8.0500    G6
S
..        ...     ...     ...        ...    ...    ...      ...   ...
...
886         0       2    male  27.000000      0      0  13.0000    G6
S
887         1       1  female  19.000000      0      0  30.0000   B42
S
888         0       3  female  29.699118      1      2  23.4500    G6
S
889         1       1    male  26.000000      0      0  30.0000  C148
C
890         0       3    male  32.000000      0      0   7.7500    G6
Q

[891 rows x 9 columns]
```

```
data.drop(["Cabin"],axis=1,inplace=True)
```

```
data
```

```
     Survived  Pclass     Sex        Age  SibSp  Parch     Fare
Embarked
0           0       3    male  22.000000      1      0   7.2500
S
1           1       1  female  38.000000      1      0  14.4542
C
2           1       3  female  26.000000      0      0   7.9250
S
```

```
3           1       1  female  35.000000      1      0  53.1000
S
4           0       3    male  35.000000      0      0   8.0500
S
..        ...     ...     ...         ...    ...    ...      ...       .
..
886         0       2    male  27.000000      0      0  13.0000
S
887         1       1  female  19.000000      0      0  30.0000
S
888         0       3  female  29.699118      1      2  23.4500
S
889         1       1    male  26.000000      0      0  30.0000
C
890         0       3    male  32.000000      0      0   7.7500
Q

[891 rows x 8 columns]
```

Splitting the data

```
y=data["Survived"]

y.head()

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

data

     Survived  Pclass     Sex        Age  SibSp  Parch     Fare
Embarked
0           0       3    male  22.000000      1      0   7.2500
S
1           1       1  female  38.000000      1      0  14.4542
C
2           1       3  female  26.000000      0      0   7.9250
S
3           1       1  female  35.000000      1      0  53.1000
S
4           0       3    male  35.000000      0      0   8.0500
S
..        ...     ...     ...         ...    ...    ...      ...       .
..
886         0       2    male  27.000000      0      0  13.0000
```

```
S
887         1       1  female  19.000000      0      0  30.0000
S
888         0       3  female  29.699118      1      2  23.4500
S
889         1       1    male  26.000000      0      0  30.0000
C
890         0       3    male  32.000000      0      0   7.7500
Q

[891 rows x 8 columns]
```

Encoding

```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

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

data["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
```

```
data.head()
```

```
   Survived  Pclass  Sex   Age  SibSp  Parch     Fare Embarked
0         0       3    1  22.0      1      0   7.2500        S
1         1       1    0  38.0      1      0  14.4542        C
2         1       3    0  26.0      0      0   7.9250        S
3         1       1    0  35.0      1      0  53.1000        S
4         0       3    1  35.0      0      0   8.0500        S
```

```
data["Embarked"]=le.fit_transform(data["Embarked"])

data.head()
```

```
   Survived  Pclass  Sex   Age  SibSp  Parch     Fare  Embarked
0         0       3    1  22.0      1      0   7.2500         2
```

```
1          1       1    0  38.0      1      0  14.4542          0
2          1       3    0  26.0      0      0   7.9250          2
3          1       1    0  35.0      1      0  53.1000          2
4          0       3    1  35.0      0      0   8.0500          2
```

```
data["Pclass"].nunique()
```

```
3
```

```
data["Pclass"].unique()
```

```
array([3, 1, 2], dtype=int64)
```

```
data["Sex"].unique()
```

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

```
data["Embarked"].unique()
```

```
array([2, 0, 1])
```

Splitting the train and test data

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data,y,test_size=0.3,ra
ndom_state=0)
```

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

```
((623, 8), (268, 8), (623,), (268,))
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
x_train=sc.fit_transform(x_train)
```

```
x_train
```

```
array([[ 1.25474307, -1.5325562 ,  0.72592065, ..., -0.47299765,
         0.67925137,  0.56710989],
       [ 1.25474307, -1.5325562 , -1.37756104, ..., -0.47299765,
        -0.26059483, -2.03075381],
       [-0.79697591,  0.84844757,  0.72592065, ...,  1.93253327,
         2.26045064,  0.56710989],
       ...,
       [-0.79697591,  0.84844757,  0.72592065, ..., -0.47299765,
```

```
        -0.78281017, -0.73182196],
       [ 1.25474307,  0.84844757, -1.37756104, ..., -0.47299765,
        -0.03170555,  0.56710989],
       [-0.79697591, -0.34205431,  0.72592065, ...,  0.72976781,
         1.64661898,  0.56710989]])

x_test=sc.fit_transform(x_test)

x_test

array([[-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.15813988, -1.76531134],
       [-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.72165412,  0.63014911],
       [-0.77151675,  0.77963055,  0.76537495, ...,  0.87064484,
         1.03823178, -0.56758111],
       ...,
       [-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.15847431, -1.76531134],
       [ 1.29614814,  0.77963055, -1.30654916, ..., -0.47809977,
        -0.72607524,  0.63014911],
       [-0.77151675, -1.64991582,  0.76537495, ..., -0.47809977,
         0.92369033, -1.76531134]])
```