

assignment-3

September 19, 2023

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: df=pd.read_csv("Titanic-Dataset.csv")
```

```
[3]: df.head()
```

```
[3]: 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
```

```
[4]: df.describe()
```

```
[4]: 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

```
[5]: 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
```

```
[6]: df.corr()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5216\1134722465.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.
df.corr()
```

```
[6]:      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

```
[7]: df.corr().Fare.sort_values(ascending = False)
```

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5216\2038886918.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.

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

```
[7]: 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
```

0.0.1 Checking for null values

```
[8]: df.isnull().any()
```

```
[8]: 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
```

```
[9]: df.isnull().sum()
```

```
[9]: 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
```

```
[10]: df["Age"].fillna(df["Age"].mean(),inplace=True)
```

```
[11]: df.isnull().any()
```

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

```
[12]: df.tail()
```

```
[12]:
```

	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	27.000000	0	0	211536	13.00	NaN	S
887	female	19.000000	0	0	112053	30.00	B42	S
888	female	29.699118	1	2	W./C. 6607	23.45	NaN	S
889	male	26.000000	0	0	111369	30.00	C148	C
890	male	32.000000	0	0	370376	7.75	NaN	Q

```
[13]: df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

```
[14]: df.isnull().any()
```

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

```
[15]: df
```

```
[15]:   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
0	Braund, Mr. Owen Harris	male	22.000000
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000
2	Heikkinen, Miss. Laina	female	26.000000
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000
4	Allen, Mr. William Henry	male	35.000000
..

886		Montvila, Rev. Juozas	male	27.000000
887		Graham, Miss. Margaret Edith	female	19.000000
888		Johnston, Miss. Catherine Helen "Carrie"	female	29.699118
889		Behr, Mr. Karl Howell	male	26.000000
890		Dooley, Mr. Patrick	male	32.000000

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

[891 rows x 12 columns]

```
[16]: df['Cabin'].fillna('Unknown', inplace=True)
```

```
[17]: df.isnull().any()
```

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

```
[18]: df
```

```
[18]:   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	\
0		Braund, Mr. Owen Harris	male	22.000000	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...		female	38.000000	
2		Heikkinen, Miss. Laina	female	26.000000	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)		female	35.000000	
4		Allen, Mr. William Henry	male	35.000000	
..		
886		Montvila, Rev. Juozas	male	27.000000	
887		Graham, Miss. Margaret Edith	female	19.000000	
888	Johnston, Miss. Catherine Helen "Carrie"		female	29.699118	
889		Behr, Mr. Karl Howell	male	26.000000	
890		Dooley, Mr. Patrick	male	32.000000	

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

[891 rows x 12 columns]

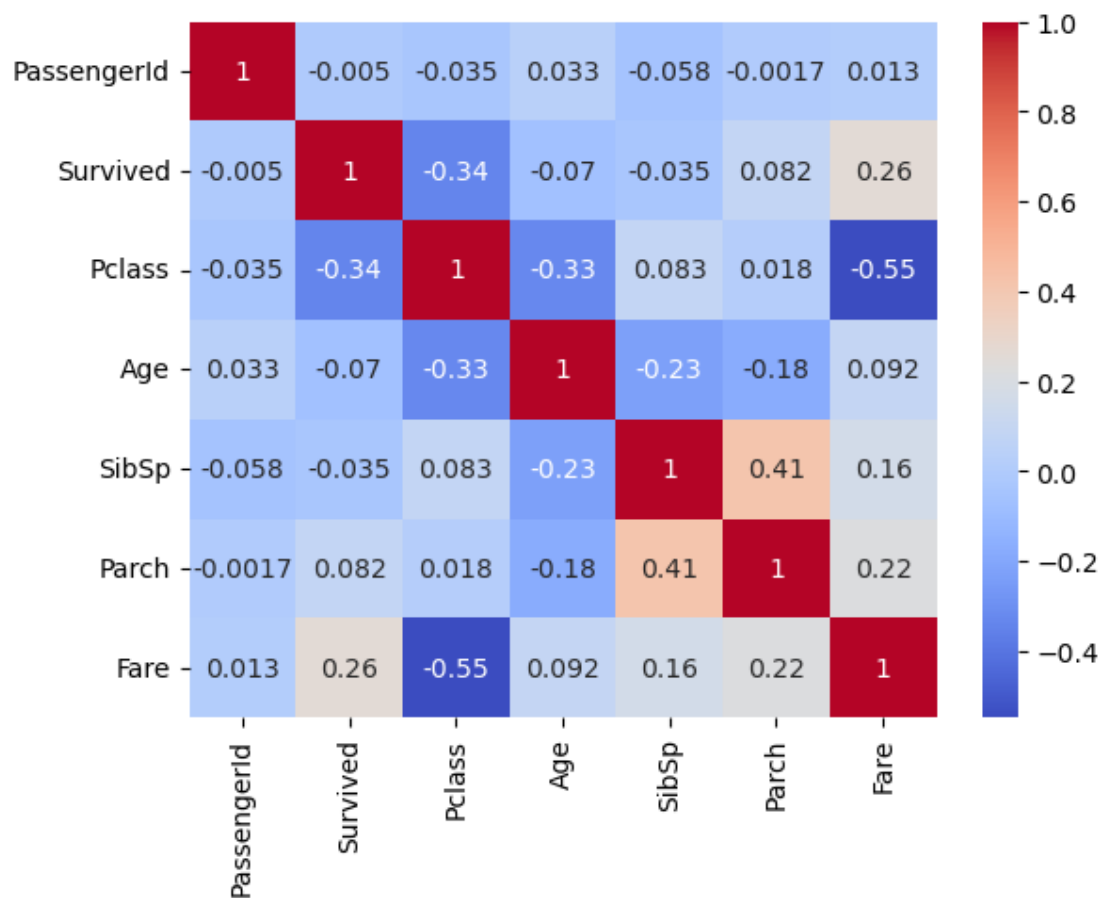
0.0.2 data visualization

```

[22]: correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.show()

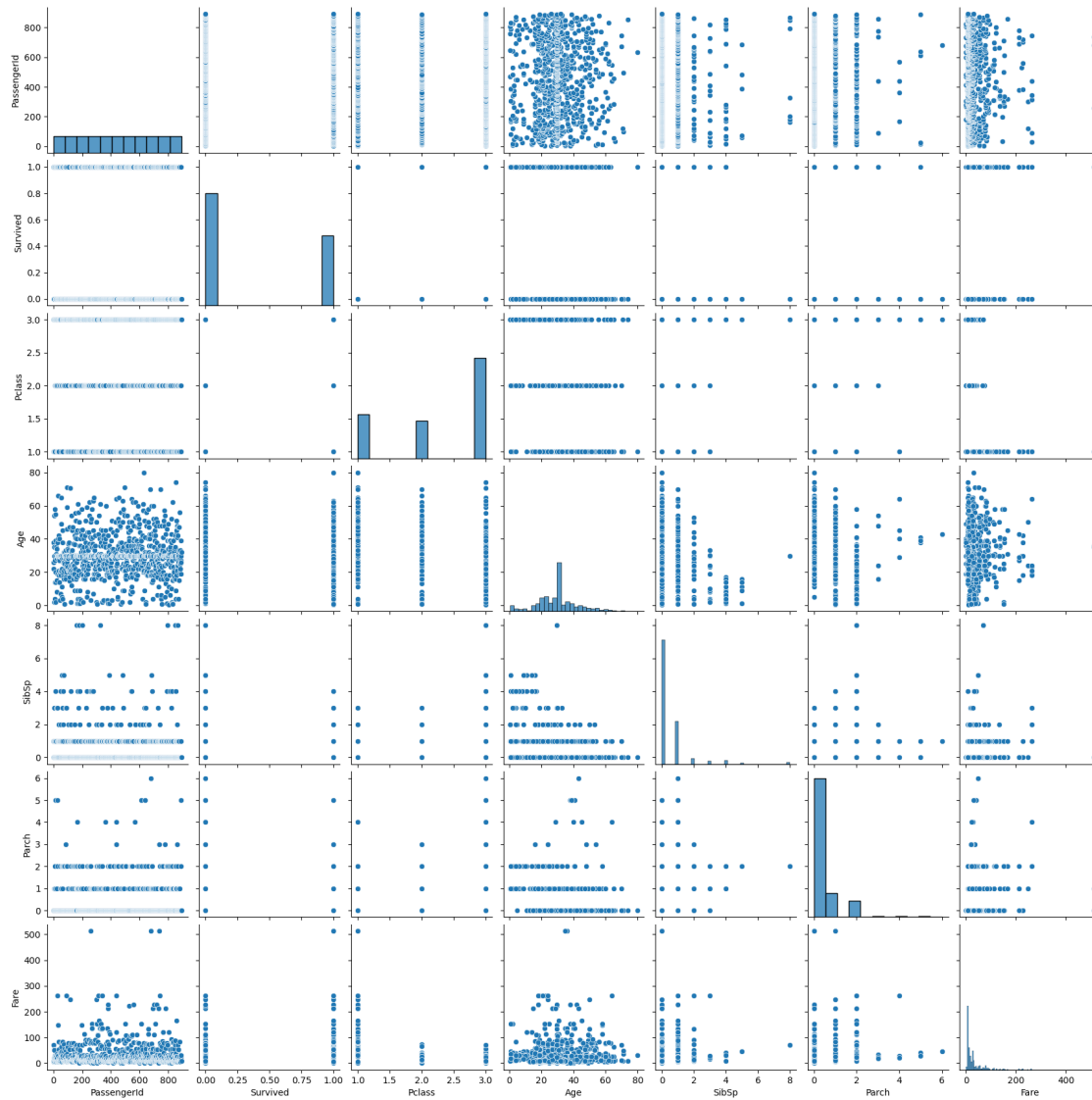
```

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5216\4145345172.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.
correlation_matrix = df.corr()

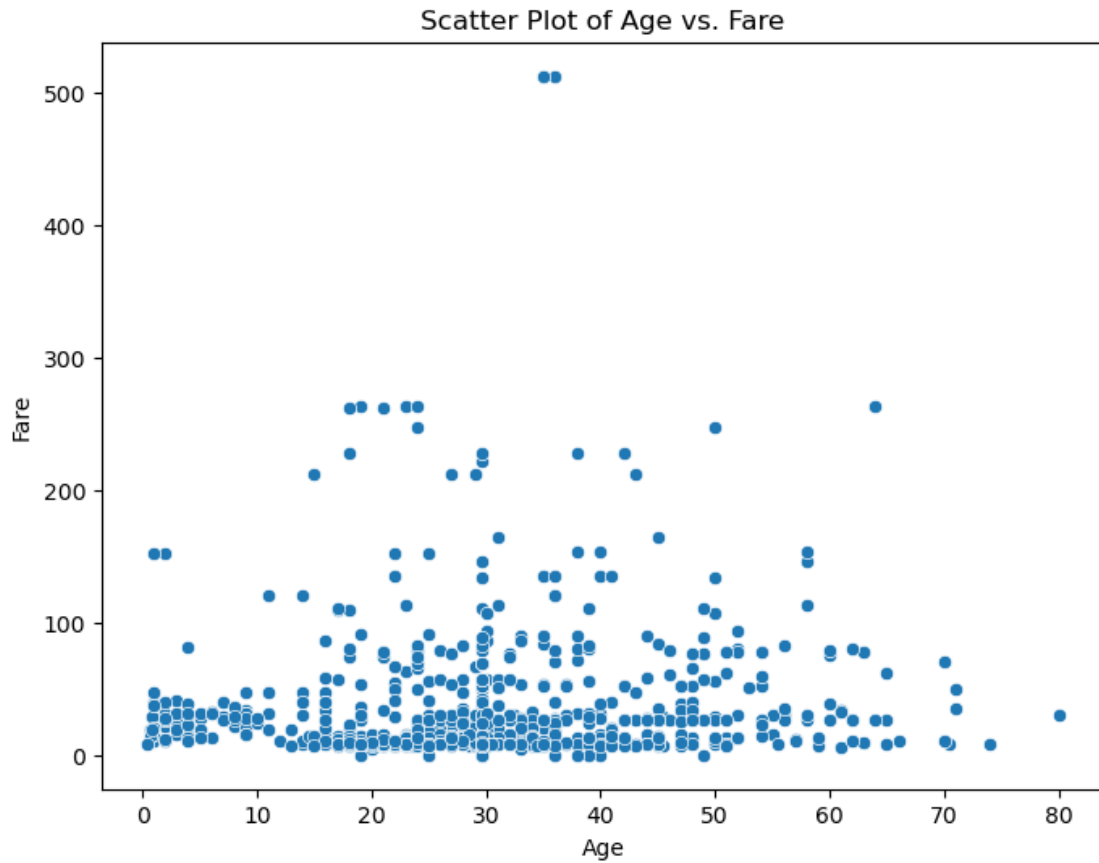


```
[23]: sns.pairplot(df)
```

```
[23]: <seaborn.axisgrid.PairGrid at 0x2acbd14d110>
```

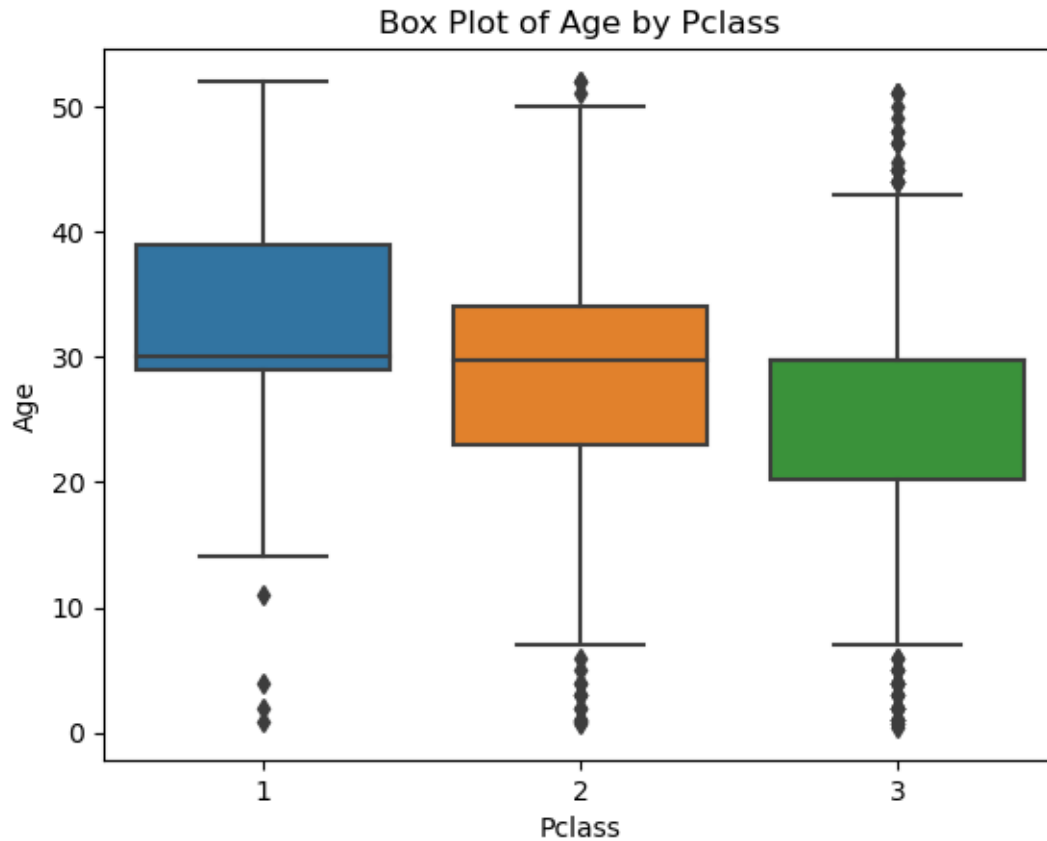



```
[24]: plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Age', y='Fare')
plt.title('Scatter Plot of Age vs. Fare')
plt.show()
```



```
[ ]:
```

```
[51]: sns.boxplot(data=df, x='Pclass', y='Age')  
plt.title('Box Plot of Age by Pclass')  
plt.show()
```

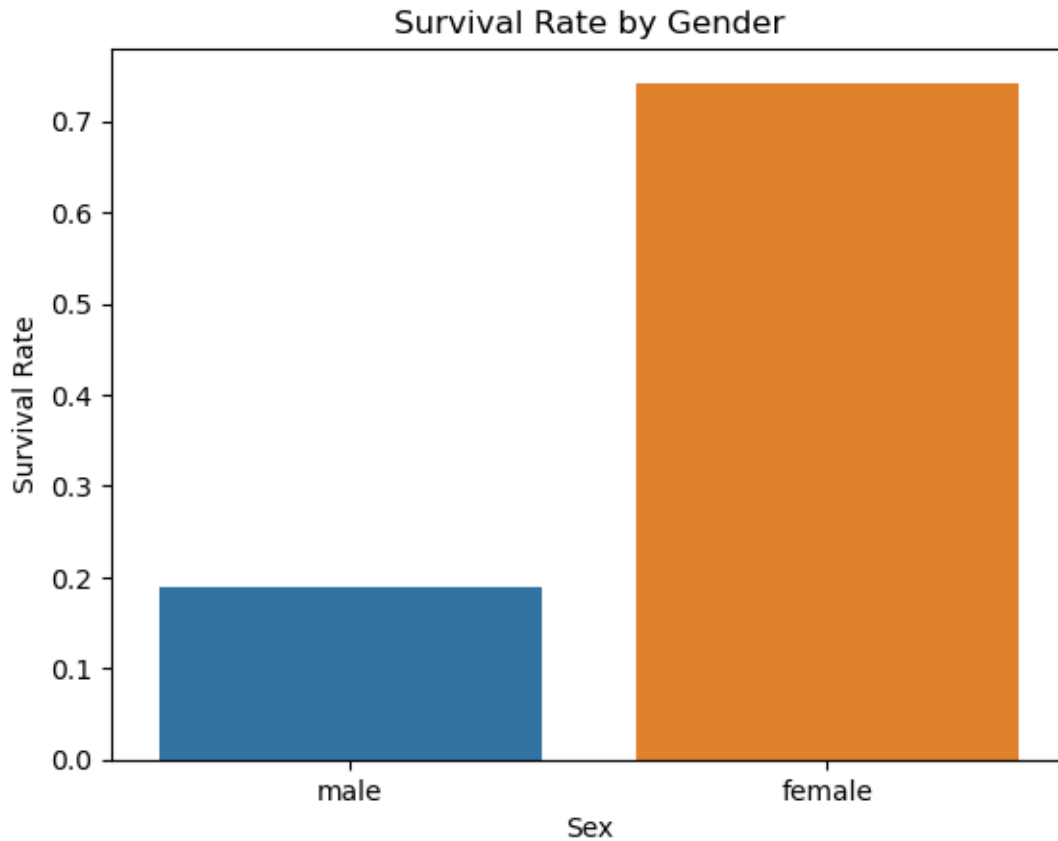


```
[28]: sns.barplot(data=df, x='Sex', y='Survived', ci=None)
plt.title('Survival Rate by Gender')
plt.ylabel('Survival Rate')
plt.show()
```

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5216\637241087.py:1: FutureWarning:

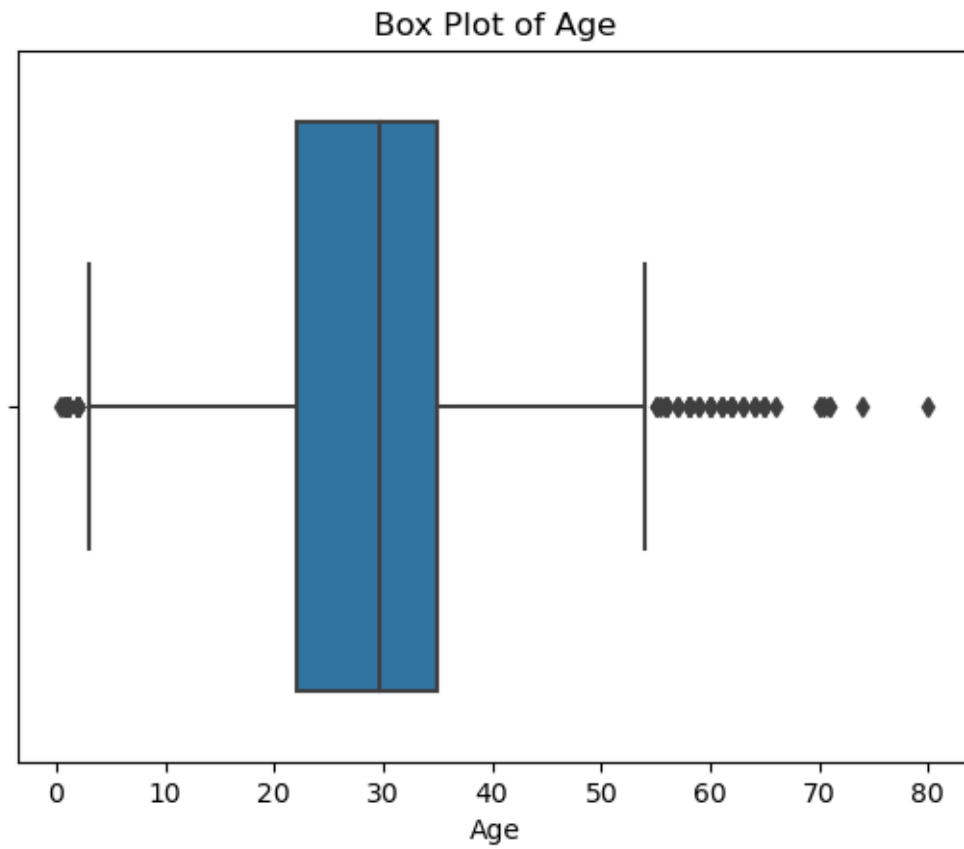
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

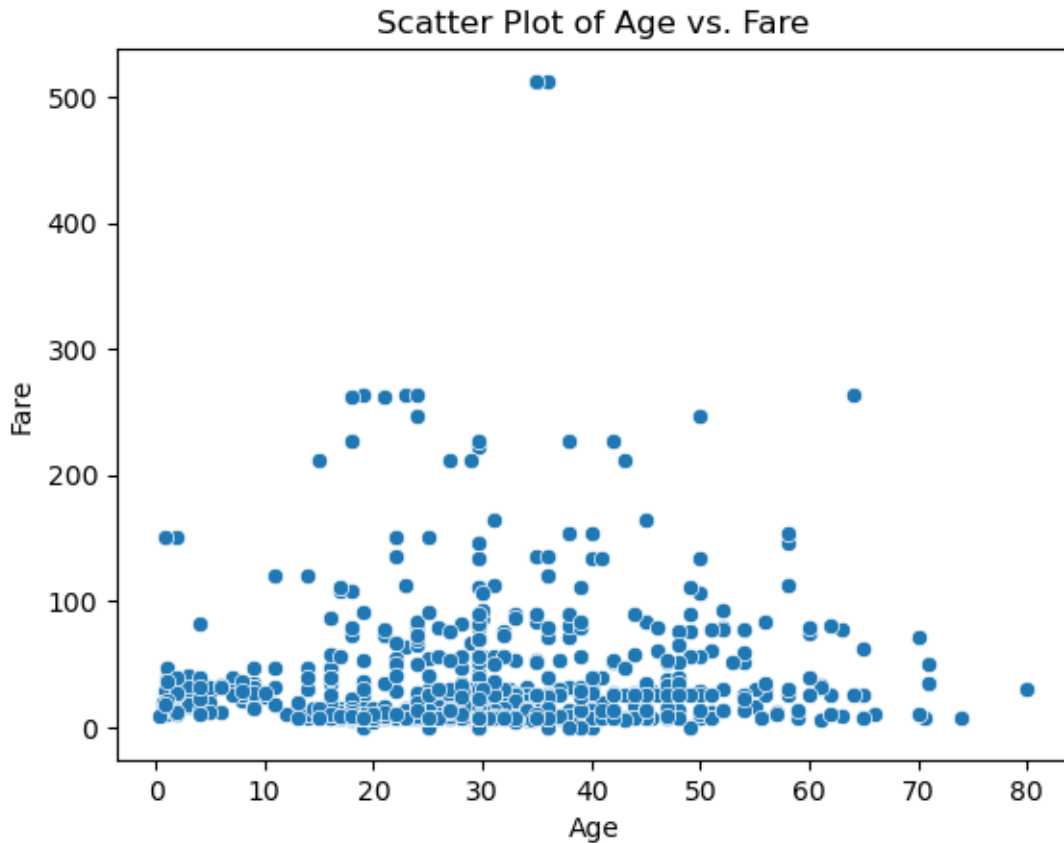
```
sns.barplot(data=df, x='Sex', y='Survived', ci=None)
```



```
[29]: # Box plot for detecting outliers in a variable (e.g., 'Age')
sns.boxplot(x=df['Age'])
plt.title('Box Plot of Age')
plt.show()

# Scatter plot for detecting outliers between 'Fare' and 'Age'
sns.scatterplot(data=df, x='Age', y='Fare')
plt.title('Scatter Plot of Age vs. Fare')
plt.show()
```





```
[34]: q1 = df.Age.quantile(0.25)
      q3 = df.Age.quantile(0.75)
```

```
[35]: IQR = q3 - q1
```

```
[36]: IQR
```

```
[36]: 13.0
```

```
[37]: upper_limit = q3+1.5*IQR
```

```
[38]: upper_limit
```

```
[38]: 54.5
```

```
[39]: lower_limit = q1-1.5*IQR
      lower_limit
```

```
[39]: 2.5
```

```
[40]: df.median()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5216\530051474.py:1: FutureWarning:
The default value of numeric_only in DataFrame.median is deprecated. In a future
version, it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of numeric_only to
silence this warning.
```

```
df.median()
```

```
[40]: PassengerId    446.000000
      Survived      0.000000
      Pclass       3.000000
      Age         29.699118
      SibSp       0.000000
      Parch       0.000000
      Fare       14.454200
      dtype: float64
```

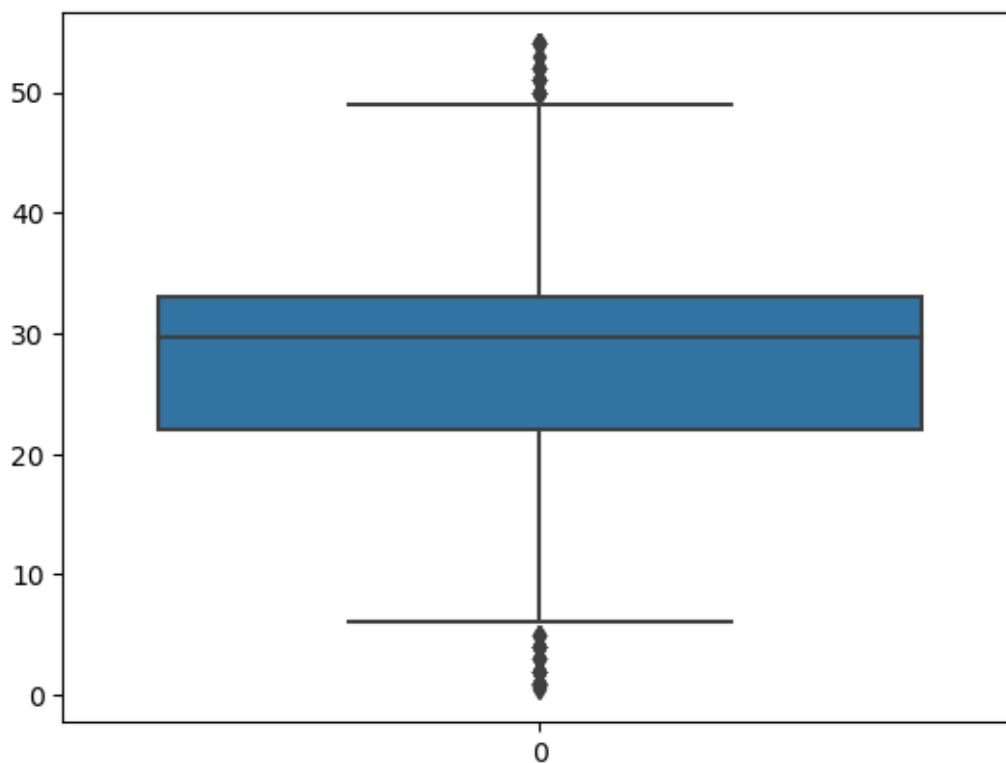
```
[41]: df['Age']=np.where(df['Age']>upper_limit,30,df['Age'])
```

```
[42]: df.Age
```

```
[42]: 0      22.000000
      1      38.000000
      2      26.000000
      3      35.000000
      4      35.000000
      ...
      886    27.000000
      887    19.000000
      888    29.699118
      889    26.000000
      890    32.000000
      Name: Age, Length: 891, dtype: float64
```

```
[43]: sns.boxplot(df.Age)
```

```
[43]: <Axes: >
```



[52]: df

```
[52]:      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  \
0  Braund, Mr. Owen Harris  male  22.000000
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.000000
2  Heikkinen, Miss. Laina  female  26.000000
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.000000
4  Allen, Mr. William Henry  male  35.000000
..          ...         ...         ...
```


886		Montvila, Rev. Juozas	male	27.000000
887		Graham, Miss. Margaret Edith	female	19.000000
888		Johnston, Miss. Catherine Helen "Carrie"	female	29.699118
889		Behr, Mr. Karl Howell	male	26.000000
890		Dooley, Mr. Patrick	male	32.000000

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

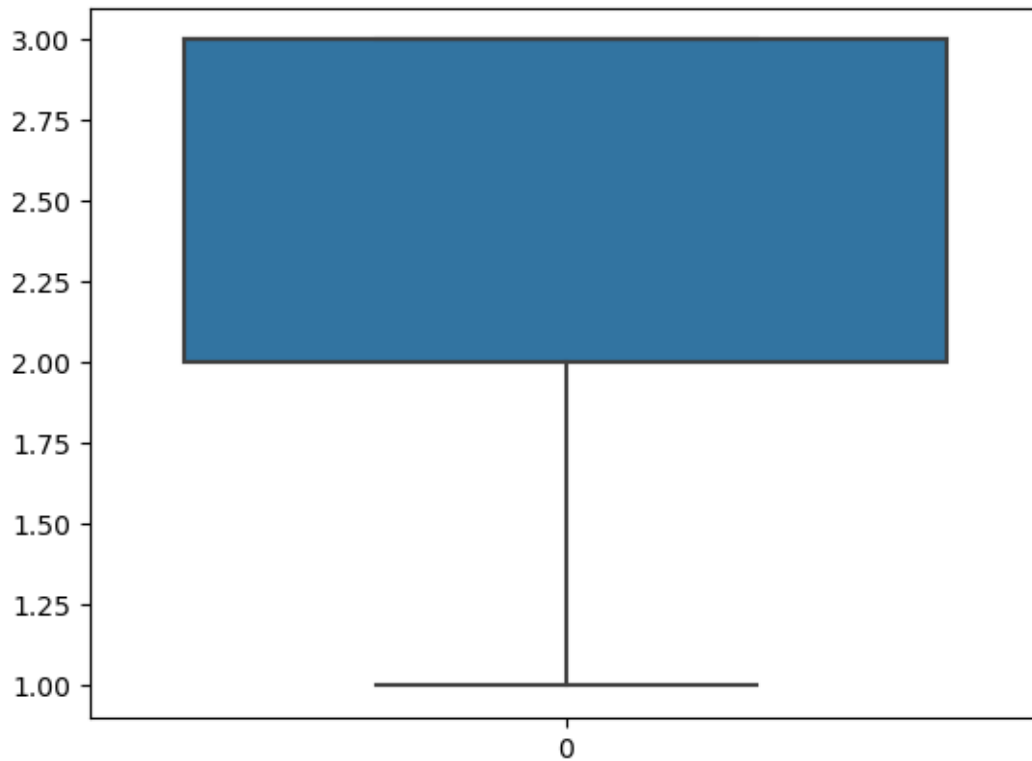
[882 rows x 12 columns]

```
[54]: df.shape
```

```
[54]: (882, 12)
```

```
[55]: sns.boxplot(df.Pclass)
```

```
[55]: <Axes: >
```



0.0.3 Splitting Dependent and independent Variables

```
[122]: df.head()
```

```
[122]: 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  Unknown      S
1      0      PC 17599  71.2833      C85      C
2      0  STON/O2. 3101282   7.9250  Unknown      S
```

3	0	113803	53.1000	C123	S
4	0	373450	8.0500	Unknown	S

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

```
[123]:
```

	PassengerId	Pclass	Name \
0	1	3	Braund, Mr. Owen Harris
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	3	3	Heikkinen, Miss. Laina
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	5	3	Allen, Mr. William Henry

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	male	22.0	1	0	A/5 21171	7.2500	Unknown	S
1	female	38.0	1	0	PC 17599	71.2833	C85	C
2	female	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	S
3	female	35.0	1	0	113803	53.1000	C123	S
4	male	35.0	0	0	373450	8.0500	Unknown	S

```
[124]: X.shape
```

```
[124]: (882, 11)
```

```
[125]: type(X)
```

```
[125]: pandas.core.frame.DataFrame
```

```
[126]: y=df["Survived"]
y.head()
```

```
[126]:
```

0	0
1	1
2	1
3	1
4	0

Name: Survived, dtype: int64

0.0.4 Encoding

```
[127]: X=df.drop(columns=["Name","Cabin","Ticket"],axis=1)
X.head()
```

```
[127]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S

3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S

```
[128]: df.Sex.nunique()
```

```
[128]: 2
```

```
[129]: df.Sex.unique()
```

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

```
[130]: df.Sex.value_counts()
```

```
[130]: male      572
female    310
Name: Sex, dtype: int64
```

```
[131]: df.Embarked.nunique()
```

```
[131]: 3
```

```
[132]: df.Embarked.unique()
```

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

```
[133]: df.Embarked.value_counts()
```

```
[133]: S      639
C      166
Q       77
Name: Embarked, dtype: int64
```

```
[134]: from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

```
[135]: X["Sex"]=le.fit_transform(X["Sex"])
```

```
[136]: X["Embarked"]=le.fit_transform(X["Embarked"])
```

```
[137]: X.head()
```

```
[137]:   PassengerId  Survived  Pclass  Sex  Age  SibSp  Parch    Fare  Embarked
0            1         0        3    1  22.0     1     0    7.2500         2
1            2         1        1    0  38.0     1     0   71.2833         0
2            3         1        3    0  26.0     0     0    7.9250         2
3            4         1        1    0  35.0     1     0   53.1000         2
4            5         0        3    1  35.0     0     0    8.0500         2
```

```
[138]: print(le.classes_)
```

```
['C' 'Q' 'S']
```

```
[139]: mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

```
[139]: {'C': 0, 'Q': 1, 'S': 2}
```

```
[140]: from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

```
[141]: X_Scaled=ms.fit_transform(X)
```

```
[142]: X_Scaled=pd.DataFrame(ms.fit_transform(X),columns=X.columns)
```

```
[143]: X_Scaled.head()
```

```
[143]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	\
0	0.000000	0.0	1.0	1.0	0.418379	0.125	0.0	0.014151	
1	0.001124	1.0	0.0	0.0	0.728577	0.125	0.0	0.139136	
2	0.002247	1.0	1.0	0.0	0.495929	0.000	0.0	0.015469	
3	0.003371	1.0	0.0	0.0	0.670415	0.125	0.0	0.103644	
4	0.004494	0.0	1.0	1.0	0.670415	0.000	0.0	0.015713	

	Embarked
0	1.0
1	0.0
2	1.0
3	1.0
4	1.0

0.0.5 Train test split

```
[144]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X_Scaled,y,test_size =0.
↪2,random_state =0)
```

```
[145]: print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(705, 9) (177, 9) (705,) (177,)
```

0.0.6 Model Building

```
[146]: from sklearn.linear_model import LinearRegression
```

```
[147]: lr=LinearRegression()
```

```
[148]: lr.fit(x_train,y_train)
```

```
[148]: LinearRegression()
```

```
[149]: y_pred=lr.predict(x_test)
```

```
[150]: y_pred
```

```
[150]: array([-2.01277876e-16, -2.32987805e-16, -2.63995225e-16, -7.80756966e-17,
        -3.81787553e-16,  1.00000000e+00,  1.78513786e-16,  1.00000000e+00,
         1.00000000e+00,  1.00000000e+00,  1.00000000e+00, -2.45528956e-16,
         1.00000000e+00, -1.95050382e-16,  1.00000000e+00,  1.00000000e+00,
         1.00000000e+00,  1.00000000e+00,  1.00000000e+00, -1.36671680e-16,
        -1.64586838e-16,  1.00000000e+00, -1.09194904e-16,  3.03626896e-16,
        -1.83146977e-16,  1.00000000e+00, -1.62697549e-16,  1.00000000e+00,
         1.00000000e+00, -1.86001296e-16, -2.25026404e-16,  1.00000000e+00,
         1.00000000e+00,  1.78718188e-16, -2.64128236e-16, -1.06073056e-16,
         1.00000000e+00,  1.00000000e+00,  1.00000000e+00,  1.00000000e+00,
        -2.31190028e-16,  1.79784139e-16, -1.17932488e-16, -1.90246969e-16,
         1.00000000e+00,  1.00000000e+00,  2.41846960e-16,  4.04990686e-17,
         1.00000000e+00,  1.00000000e+00,  1.00000000e+00,  1.00000000e+00,
         1.60291134e-16, -1.33210019e-16,  1.11167223e-16,  2.38420838e-17,
        -1.55535054e-16,  1.00000000e+00,  1.00000000e+00, -2.19118648e-16,
        -1.70632477e-16,  1.00000000e+00,  1.00000000e+00, -1.75232524e-16,
        -1.01922305e-16,  1.00000000e+00,  1.00000000e+00,  1.90763198e-17,
         1.00000000e+00,  1.00000000e+00,  1.00000000e+00,  1.00000000e+00,
        -2.97195570e-16, -8.02544875e-17,  1.00000000e+00,  5.14567479e-16,
         1.00000000e+00,  1.00000000e+00,  1.26759398e-16,  1.00000000e+00,
        -2.01732232e-16,  2.76171217e-17,  1.93816083e-16,  1.29631591e-16,
         1.00000000e+00, -2.04397063e-16,  1.00000000e+00, -3.77939632e-16,
         1.00000000e+00, -2.61451023e-16, -1.99399217e-16,  4.07264087e-17,
        -7.49445019e-17,  1.02342387e-16,  1.00000000e+00,  2.90274268e-16,
        -3.75732427e-16,  1.00000000e+00, -1.49725639e-16, -2.46380474e-16,
         1.00000000e+00,  1.00000000e+00, -2.23110332e-16,  1.00000000e+00,
        -2.53228855e-16,  1.97163555e-16,  1.00000000e+00,  1.00000000e+00,
         1.00000000e+00, -2.58994532e-16, -3.02438232e-17,  1.00000000e+00,
        -2.70922236e-16, -9.00596065e-17, -2.99263391e-16, -2.11534726e-16,
        -3.82130359e-16,  1.00000000e+00, -2.26586279e-16,  1.00000000e+00,
        -5.59176198e-17,  1.00000000e+00,  3.70660000e-16,  1.00000000e+00,
        -1.76478761e-16, -2.22222946e-16,  4.59113427e-16,  1.00000000e+00,
        -1.06375597e-16,  2.30255850e-16, -1.55252341e-16,  1.00000000e+00,
        -3.35051080e-16,  3.22563706e-16, -1.81183245e-16,  4.09050750e-16,
         1.00000000e+00, -1.67769473e-16, -2.49551577e-16,  3.00224716e-17,
        -2.23673540e-16, -1.06934916e-16,  1.66830768e-16, -1.34457829e-16,
         5.14198941e-17, -2.14394074e-16,  1.00000000e+00,  1.50680087e-16,
        -3.14431701e-16, -2.52337686e-16, -3.41603697e-16, -2.82838380e-16,
        -1.78183304e-16,  1.00000000e+00, -1.08997780e-16, -1.09899182e-16,
```

```
-1.89000125e-16, 1.00000000e+00, -3.59929022e-16, -1.65032933e-16,
1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
1.00000000e+00, -1.70975224e-16, 1.00000000e+00, 1.00000000e+00,
-2.97501152e-16, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
1.00000000e+00, -6.46133504e-17, 1.00000000e+00, 3.10594173e-16,
-2.40508647e-16])
```

```
[151]: y_test
```

```
[151]: 152    0
      410    0
      519    0
      102    0
      592    0
      ..
      366    1
      371    0
      267    1
      324    0
      470    0
      Name: Survived, Length: 177, dtype: int64
```

```
[152]: survived=pd.DataFrame({"actual_survived":y_test,"Predicted _survived":y_pred})
```

```
[153]: survived
```

```
[153]:   actual_survived  Predicted _survived
152              0      -2.012779e-16
410              0      -2.329878e-16
519              0      -2.639952e-16
102              0      -7.807570e-17
592              0      -3.817876e-16
..              ...
366              1      1.000000e+00
371              0      -6.461335e-17
267              1      1.000000e+00
324              0      3.105942e-16
470              0      -2.405086e-16

[177 rows x 2 columns]
```

0.0.7 Evaluation of the model

```
[156]: from sklearn import metrics
```

```
[157]: print(metrics.r2_score(y_test,y_pred))
```

1.0

```
[158]: print(metrics.mean_squared_error(y_test,y_pred))
```

6.482544721493499e-32

```
[159]: print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

2.546084193716598e-16

```
[ ]:
```