

Swamy-21bcb7126-assg-3

September 20, 2023

ASSIGNMENT 3 A SWAMY 21BCB7126

1 DATA PREPROCESSING

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

IMPORTING THE DATASET

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

```
[3]: df
```

```
[3]:      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 **Error! Bookmark not defined.**

1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 2

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 2

2 Heikkinen, Miss. Laina female 26.0 0

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]

```

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

```

[4]: 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

```
[5]: df.shape
```

```
[5]: (891, 12)
```

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

```
[6]:
```

	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

```
[7]: 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

```

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

```

C:\Users\lenovo\AppData\Local\Temp\ipykernel_11992\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()

```

```

[8]:      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

```

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

```

C:\Users\lenovo\AppData\Local\Temp\ipykernel_11992\60082530.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)

```

```
[9]: 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
```

CHECKING FOR NULL VALUES

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

```
[10]: 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
```

```
[11]: df = df.drop(['Cabin'], axis=1)
      df
```

```
[11]: 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 Embarked
0         0  A/5 21171  7.2500        S
1         0   PC 17599 71.2833        C
2         0 STON/O2. 3101282 7.9250        S
3         0  113803 53.1000        S
4         0  373450   8.0500        S
..      ...      ...      ...
886        0  211536 13.0000        S
887        0 112053 30.0000 S 888 2 W./C. 6607 23.4500 S
889        0           111369 30.0000        C
890        0           370376  7.7500        Q

```

[891 rows x 11 columns]

We dropped cabin because it has highest number of null values.

```
[12]: df['Age'].fillna(df['Age'].mean(), inplace=True)
```

```
[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
      Embarked      False
      dtype: bool

```

Finally, we can observe there are no null values in any attribute

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

```
[15]: 3
```

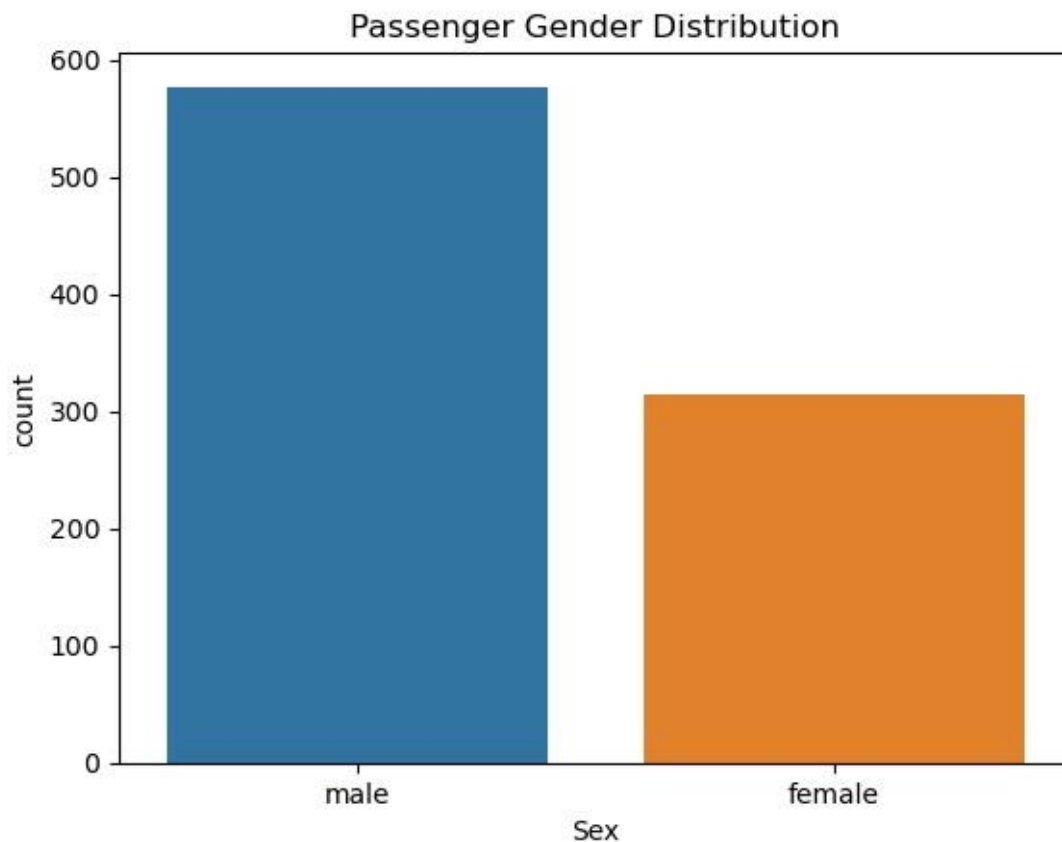
```
[16]: df.Embarked.unique()
```

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

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

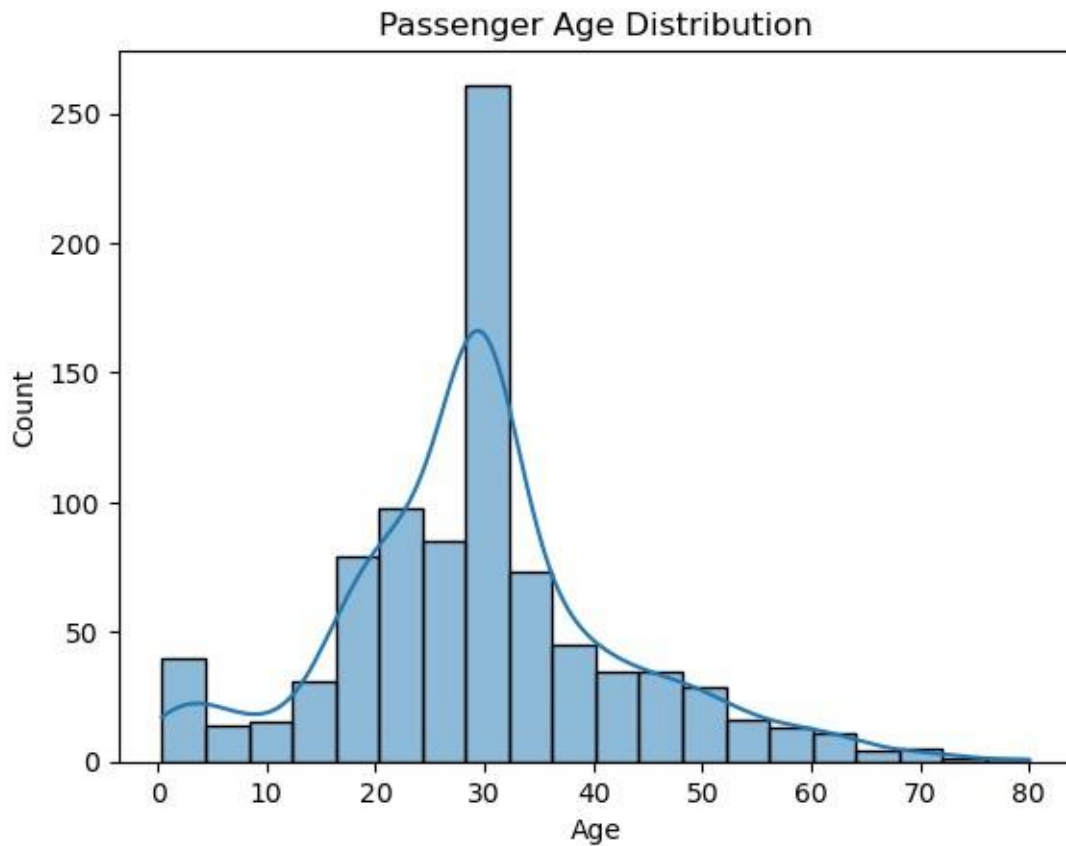
```
[17]: S
      646 C
      168
      Q    77
      Name: Embarked, dtype: int64
```

```
[18]: sns.countplot(data=df, x='Sex')
      plt.title('Passenger Gender Distribution')
      plt.show()
```



INFERENCE: We can observe that there are more number of male passengers than female passengers

```
[19]: sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title('Passenger Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



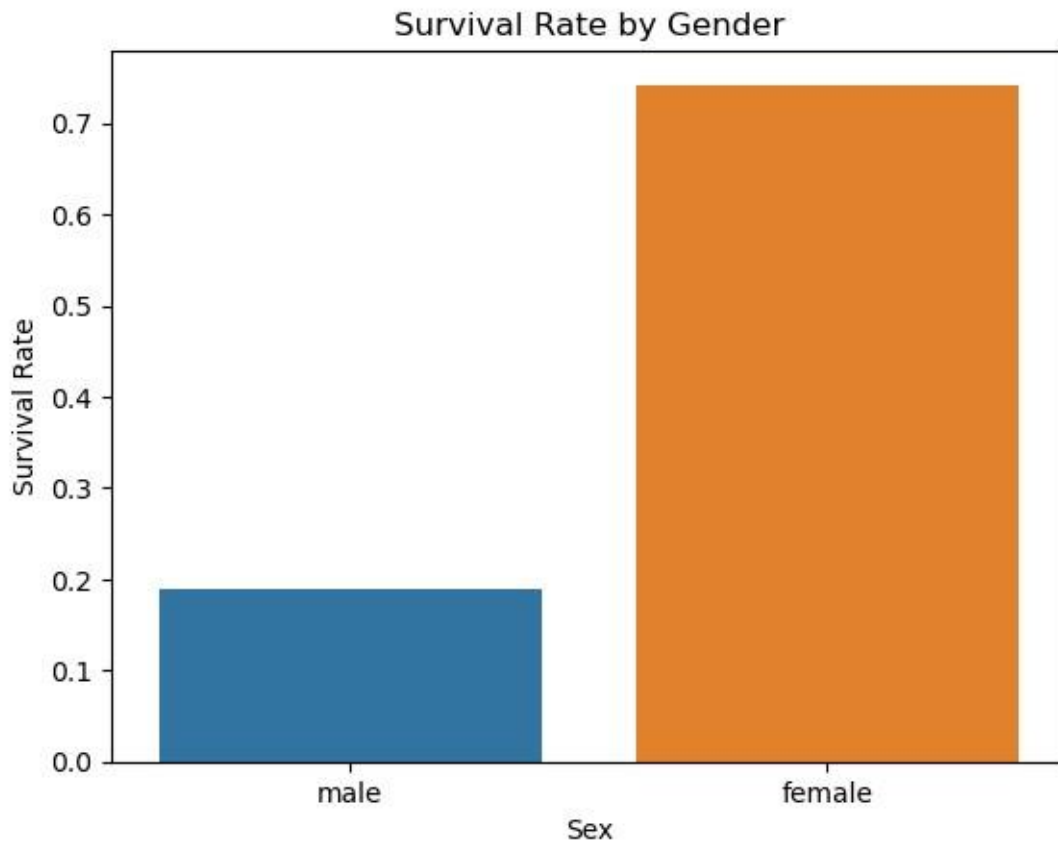
INFERENCE: The histogram of the 'Age' distribution provides insights into the age of Titanic passengers, showing that the majority were 30 to 40 aged adults, but there were also significant numbers of younger and older passengers.

```
[20]: 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_11992\3687825708.py:1:
FutureWarning:

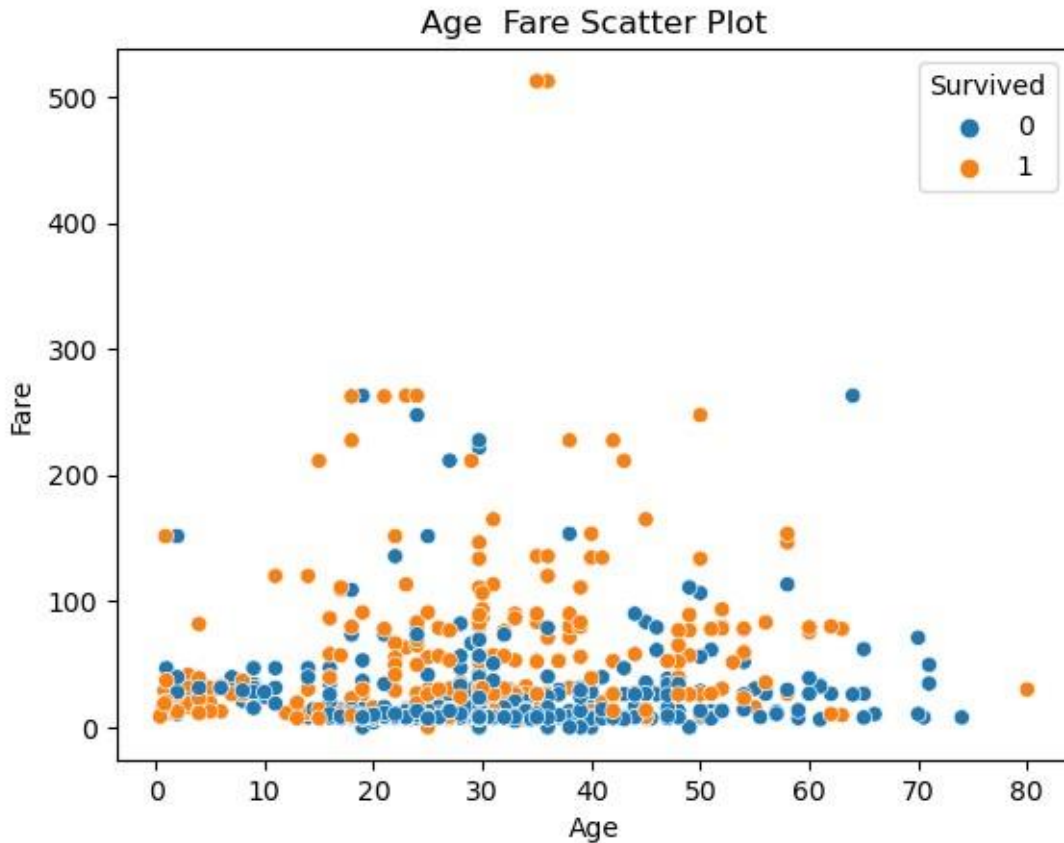
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.


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



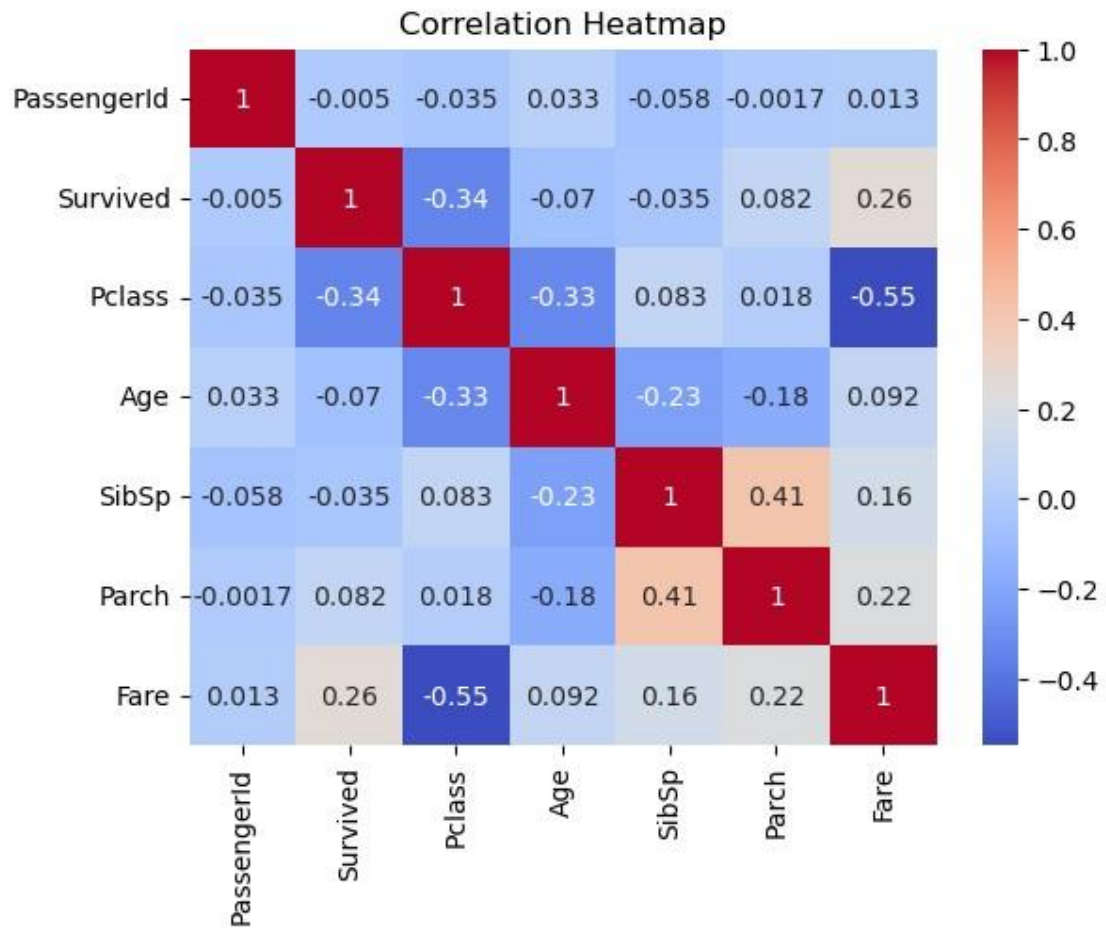
INFERENCE: we can observe that female passengers have high survival rate than male passengers

```
[21]: sns.scatterplot(data=df, x='Age', y='Fare', hue='Survived')
plt.title('Age Fare Scatter Plot')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```



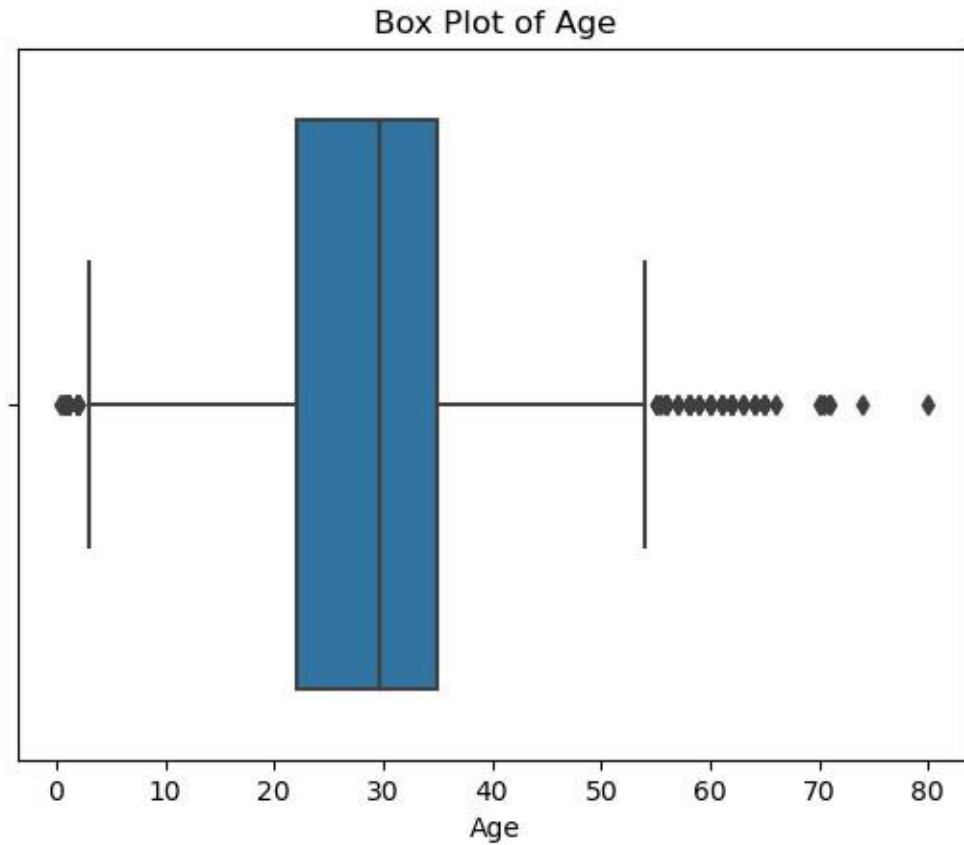
```
[22]: correlation_matrix = df.corr()  
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')  
plt.title('Correlation Heatmap')  
plt.show()
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_11992\2298098936.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()
```



OUTLIER DETECTION

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



```
[24]: df.shape
```

```
[24]: (891, 11)
```

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

```
22.0
```

```
35.0
```

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

```
[26]: 13.0
```

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

```
[27]: 54.5
```

```
[28]: lower_limit=q3-1.5*IQR  
lower_limit
```

```
[28]: 15.5
```

```
[29]: from scipy import stats  
  
z_scores = np.abs(stats.zscore(df['Age']))  
outliers = (z_scores > 3)  
z_scores
```

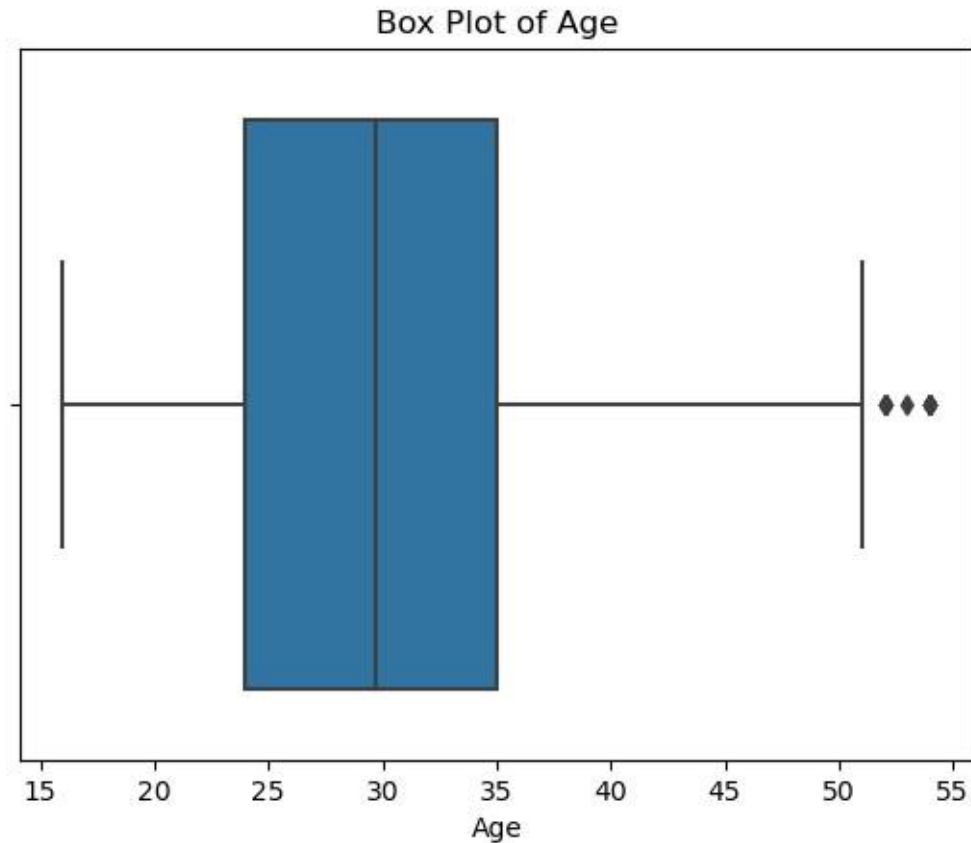
```
[29]: 0      0.592481  
1      0.638789  
2      0.284663  
3      0.407926  
4      0.407926  
...  
886    0.207709  
887    0.823344  
888    0.000000  
889    0.284663  
890    0.177063  
Name: Age, Length: 891, dtype: float64
```

```
[30]: df_no_outliers = df[(df['Age'] >= lower_limit) & (df['Age'] <=  
upper_limit)] print("Original dataset shape:", df.shape)  
print("Dataset shape after removing outliers:",  
df_no_outliers.shape)
```

```
Original dataset shape: (891, 11)
```

```
Dataset shape after removing outliers: (766, 11)
```

```
[31]: sns.boxplot(data=df_no_outliers, x='Age')  
plt.title('Box Plot of Age')  
plt.show()
```



SPLITTING DEPENDENT AND INDEPENDENT VARIABLES

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

```
[32]: 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  26.00
3  Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35.0  1
4  Allen, Mr. William Henry      male  35.0  0
```

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

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

```
[33]: 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	26.00	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1
4	Allen, Mr. William Henry	male	35.0	0

	Parch	Ticket	Embarked
0	0	A/5 21171	S
1	0	PC 17599	C
2	0	STON/O2. 3101282	S
3	0	113803	S
4	0	373450	S

```
[34]: X.shape
```

```
[34]: (891, 10)
```

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

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

```
[36]: y=df["Fare"]
      y.head()
```

```
[36]: 0    7.2500
      1   71.2833
      2    7.9250
```

```
3    53.1000
4     8.0500
Name: Fare, dtype: float64
```

ENCODING

[37]:

X.head()

[37]: 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  26.00
3  Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35.0    1
4  Allen, Mr. William Henry      male  35.0    0
```

```

Parch      Ticket Embarked
0      0   A/5 21171      S
1      0   PC 17599      C
2      0 STON/O2. 3101282      S
3      0   113803      S
4      0      373450      S
```

[38]: `from sklearn.preprocessing import LabelEncoder`

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

[38]: 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 Parch \
0  Braund, Mr. Owen Harris male  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	Embarked
0	A/5 21171	S
1	PC 17599	C
2	STON/O2. 3101282	S
3	113803	S
4	373450	S

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

```
[39]: 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	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	Embarked
0	A/5 21171	2
1	PC 17599	0
2	STON/O2. 3101282	2
3	113803	2
4	373450	2

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

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

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

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

FEATURE SCALING

```
[49]: from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
from sklearn.preprocessing import
MinMaxScaler numerical_features =
['Age', 'Fare'] data =
df[numerical_features] ms =
MinMaxScaler()
scaled_data = ms.fit_transform(data) df_scaled =
pd.DataFrame(scaled_data, columns=numerical_features)
print(df_scaled.head())
```

	Age	Fare
0	0.271174	0.014151
1	0.472229	0.139136
2	0.321438	0.015469
3	0.434531	0.103644
4	0.434531	0.015713

SPLITTING DATA INTO TRAIN AND TEST

```
[50]: 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)
```

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

```
(712, 10) (179, 10) (712,) (179,)
```

```
[52]: X = df.drop('Survived', axis=1)
y = df['Survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

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

```
(712, 10) (179, 10) (712,) (179,)
```

```
[55]:
```

```
[54]: df= df.drop(['PassengerId', 'Name', 'Ticket'], axis=1)
```

```
df
```

```
[55]:
```

	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	71.2833	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]

[]: