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

df = pd.read_csv('/content/Titanic-Dataset.csv')

```

exploring the dataset

```
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

Next steps: Generate code with df View recommended plots New interactive sheet

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

```
                0
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.nunique()
```

```
                0
PassengerId   891
   Survived     2
     Pclass     3
       Name   891
        Sex     2
        Age    88
      SibSp     7
      Parch     7
     Ticket   681
       Fare   248
   Embarked     3

dtype: int64
```

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df['Embarked'].value_counts()
```

```
          count
Embarked
   S        644
   C        168
   Q         77

dtype: int64
```

handling the missing data

```
#filling the age colum with the median value why median bacause the data may contain outlier values
df['Age'].fillna(df['Age'].median(),inplace=True)

#filling embarked with mode (mode can be used for categotical columns )
#(why[0]- mode return a frequent values to get the first frequent value we use this [0] )

df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)

#drop cabin (too many null values)

```

```
df.drop('Cabin',axis = 1,inplace=True)

# or we can fill this code with the values like unknown

df['cabin'].fillna("unknown")


```

```
<ipython-input-10-be7b1f61c0a4>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting 

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col


  df['Age'].fillna(df['Age'].median(),inplace=True)
<ipython-input-10-be7b1f61c0a4>:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting 

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col


  df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)
```

Encoding the categorical variables

```
#label encoding sex column assigning o for male and 1 for female
df['Sex'] = df['Sex'].map({'male':0,'female':1})

#using one hot encoding because it is a location

df = pd.get_dummies(df,columns=['Embarked'],drop_first=True)

```

normalizing

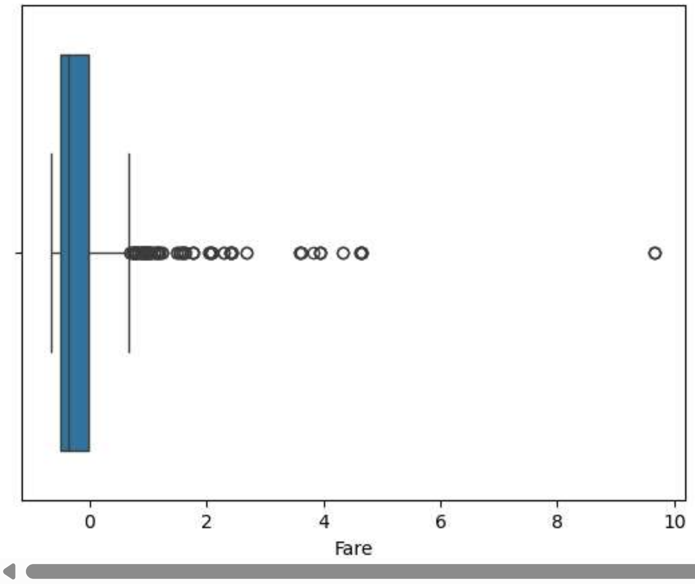
```
#normalizing the age and fare column between 0 to 1 to improve the model performance
#using standardscaler we can use other preprocessors like minmaxscaler but it is more suitable in timeseries data,robustscaler when

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[['Age', 'Fare']] =scaler.fit_transform(df[['Age', 'Fare']])

```

detect,remove outlier values

```
sns.boxplot(x=df['Fare'])
plt.show()
```



```
# there are several methods to remove outlier some of the methods like IQR z-score and isolation forest can be used for this data
#i am using the z-score method where - a value is more than 3 standard deviations from the mean, it’s an outlier.

from scipy import stats
z = np.abs(stats.zscore(df['Fare']))
```

```
df = df[(z<3)]

```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked_S
0	1	0	3	Braund, Mr. Owen Harris	NaN	-0.565736	1	0	A/5 21171	-0.502445	False	True
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	NaN	0.663861	1	0	PC 17599	0.786845	False	False
				Heikkinen, Miss					STON/O2.			

Next steps: Generate code with df View recommended plots New interactive sheet

covert the cleaned data to a csv file

```
#convert the data into csv usin the to_csv function in pandas

df.to_csv("titanic_cleaned_data_by_sivashankar.csv", index=False)
```