ASSIGNMENT -3

NAME: Bhukya Sirisha

REG.NO:21BCE9747

ASSIGNMENT-3 NAME:Boya Chaitanya REG NO:21BCE9968

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

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

	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
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
da.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
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
da.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

```python
da.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
```

```python
da.corr()
```

```
<ipython-input-8-c07a06031de5>: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 col
  da.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307

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

```python
da.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
```

```python
da['Age']=da['Age'].fillna(da['Age'].mean())
da
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```python
da['Embarked']=da['Embarked'].fillna(da['Embarked'].mode()[0])
da
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...

886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
da=da.drop("Cabin",axis=1)
```

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

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

```
dtype: int64
```

+ Code + Markdown

```
da.Sex.nunique()
```

```
2
```

```
da.Sex.unique()
```

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

```
da.Sex.value_counts()
```

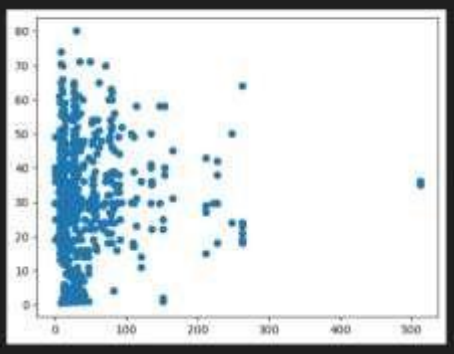
```
male      577
female    314
Name: Sex, dtype: int64
```

DATA VISUALIZATION

```
plt.scatter(da["Fare"],da["Age"])
```

```
<matplotlib.collections.PathCollection at 0x7d6513bf4cd0>
```

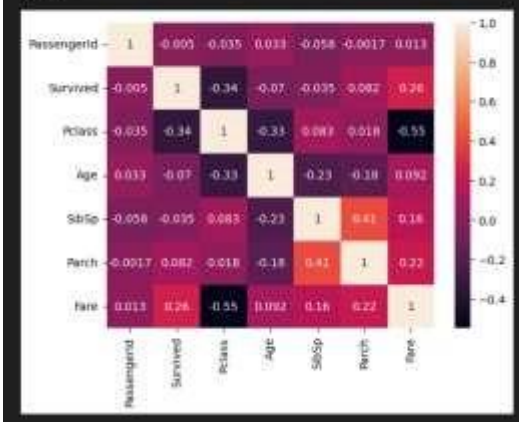
```
<matplotlib.collections.PathCollection at 0x7d6513bfaca0>
```




```
sns.heatmap(da.corr(),annot=True)
```

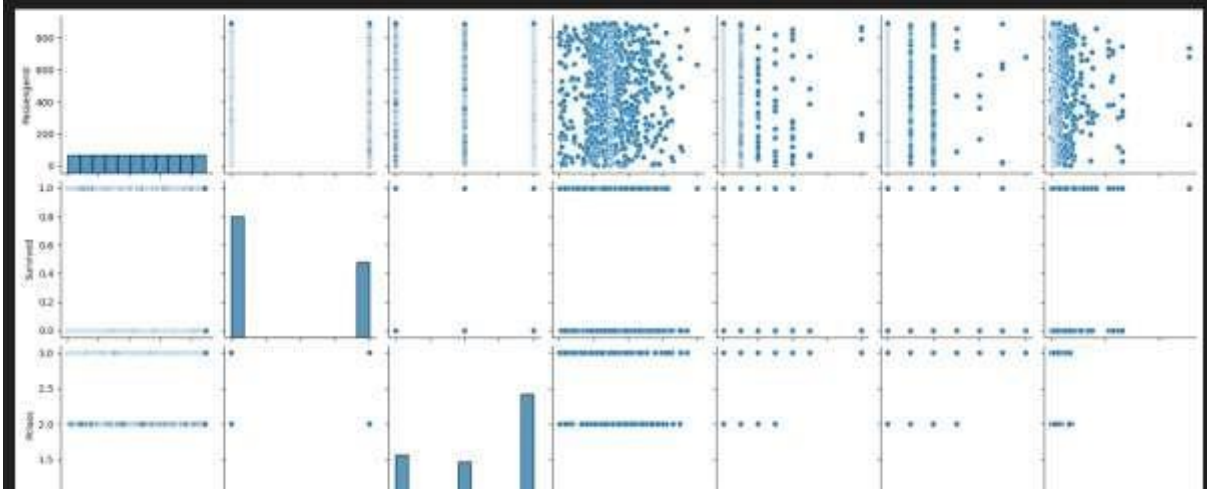
```
<ipython-input-45-48f8510b8221>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid co
  sns.heatmap(da.corr(),annot=True)
```

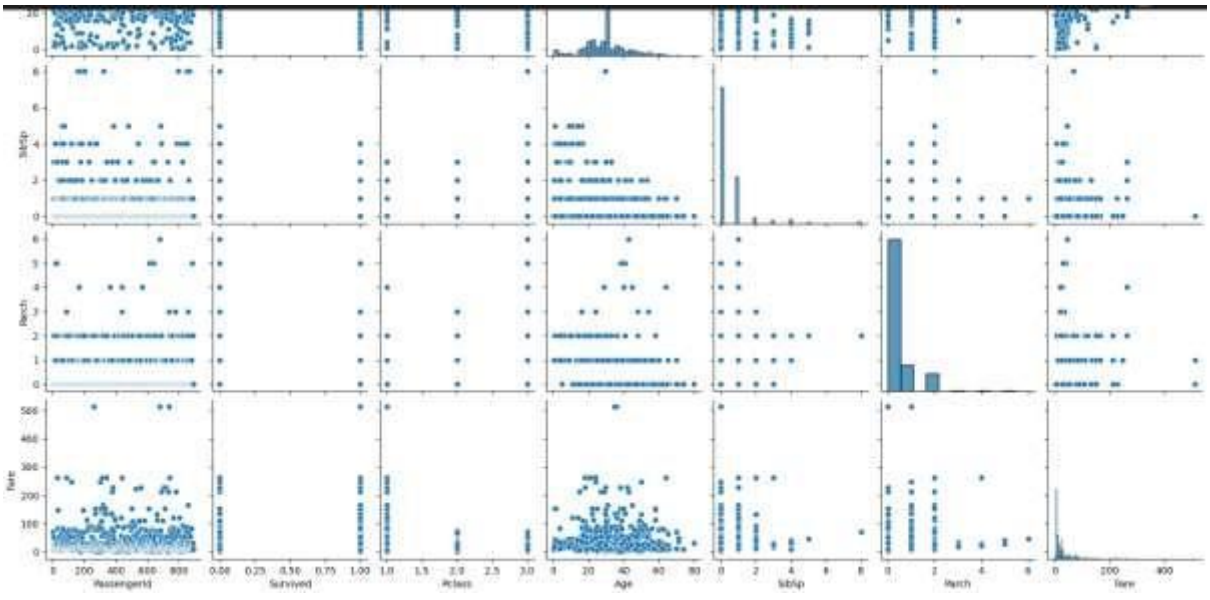
```
<ipython-input-45-4df9516dc02b>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid co
  sns.heatmap(da.corr(),annot=True)
```




```
sns.pairplot(da)
```

```
<seaborn.axisgrid.PairGrid at 0x7d6512f57d90>
```



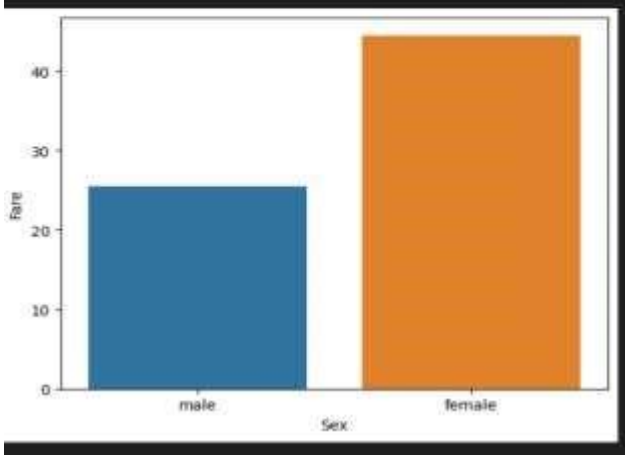


```
sns.barplot(x=da["Sex"],y=da["Fare"],ci=0)
```

```
/tmp/ipython-input-47-fd5f5e41e7ed>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effect.

  sns.barplot(x=da["Sex"],y=da["Fare"],ci=0)
<Axes: xlabel='Sex', ylabel='Fare'>
```




```
da.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

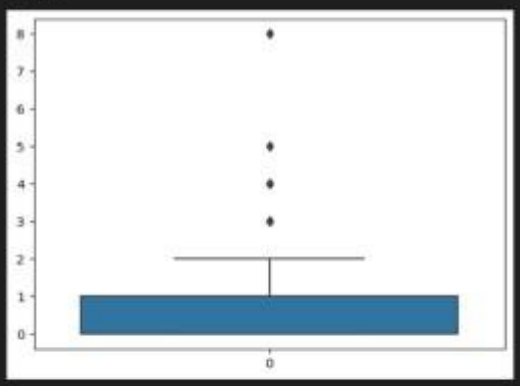
```
sns.boxplot(da["Pclass"])
```

```
<Axes: >
```

+ Code + Markdown

```
sns.boxplot(da["SibSp"])
```

```
<Axes: >
```




```
da.head()
```

```
da.head()
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

```
x=da.drop(columns=["Name"],axis=1)
x.head()
```

PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Sex_female	Sex_male	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Tick
1	0	3	22.0	1	0	7.2500	S	0	1	...	0	0	0	0	0	0	
2	1	1	38.0	1	0	71.2833	C	1	0	...	0	0	0	0	0	0	
3	1	3	26.0	0	0	7.9250	S	1	0	...	0	0	0	0	0	0	
4	1	1	35.0	1	0	53.1000	S	1	0	...	0	0	0	0	0	0	
5	0	3	35.0	0	0	8.0500	S	0	1	...	0	0	0	0	0	0	

rows × 891 columns

```
x.shape
```

Python

Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ticket_W./C. 6609	Ticket_W.E.P. 5734	Ticket_W/C 14208	Ticket_WE/P 5735
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

```
type(x)
```

Python

```
pandas.core.frame.DataFrame
```

```
y=da["Fare"]
y.head()
```

Python

```
0     7.2500
1    71.2833
2     7.9250
3    53.1000
4     8.0500
Name: Fare, dtype: float64
```

```
x.head()
```

Python

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Sex_female	Sex_male	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ticket_V
0	1	0	3	22.0	1	0	7.2500	S	0	1	...	0	0	0	0	0	0	
1	2	1	1	38.0	1	0	71.2833	C	1	0	...	0	0	0	0	0	0	
2	3	1	3	26.0	0	0	7.9250	S	1	0	...	0	0	0	0	0	0	
3	4	1	1	35.0	1	0	53.1000	S	1	0	...	0	0	0	0	0	0	
4	5	0	3	35.0	0	0	8.0500	S	0	1	...	0	0	0	0	0	0	

5 rows × 891 columns

Python

cket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ticket_W./C. 6609	Ticket_W.E.P. 5734	Ticket_W/C 14208	Ticket_WE/P 5735
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

```
print(le.classes_)
```

Python

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

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

Python

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

```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

Python

```
df
```

Python

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Fare	Embarked	Sex_female	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608
0	1	0	3	Braund, Mr. Owen Harris	22.000000	1	0	7.2500	S	0	...	0	0	0	0	0	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.000000	1	0	71.2833	C	1	...	0	0	0	0	0	0
2	3	1	3	Heikkinen, Miss. Laina	26.000000	0	0	7.9250	S	1	...	0	0	0	0	0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.000000	1	0	53.1000	S	1	...	0	0	0	0	0	0
4	5	0	3	Allen, Mr. William Henry	35.000000	0	0	8.0500	S	0	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	27.000000	0	0	13.0000	S	0	...	0	0	0	0	0	0
887	888	1	1	Graham, Miss. Margaret Edith	19.000000	0	0	30.0000	S	1	...	0	0	0	0	0	0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	29.699118	1	2	23.4500	S	1	...	0	0	0	0	1	0
889	890	1	1	Behr, Mr. Karl Howell	26.000000	0	0	30.0000	C	0	...	0	0	0	0	0	0
890	891	0	3	Dooley, Mr. Patrick	32.000000	0	0	7.7500	Q	0	...	0	0	0	0	0	0

891 rows × 692 columns

```
df.drop("Name",axis=1)
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Sex_female	Sex_male	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608
0	1	0	3	22.000000	1	0	7.2500	S	0	1	...	0	0	0	0	0	0
1	2	1	1	38.000000	1	0	71.2833	C	1	0	...	0	0	0	0	0	0
2	3	1	3	26.000000	0	0	7.9250	S	1	0	...	0	0	0	0	0	0
3	4	1	1	35.000000	1	0	53.1000	S	1	0	...	0	0	0	0	0	0
4	5	0	3	35.000000	0	0	8.0500	S	0	1	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	27.000000	0	0	13.0000	S	0	1	...	0	0	0	0	0	0
887	888	1	1	19.000000	0	0	30.0000	S	1	0	...	0	0	0	0	0	0
888	889	0	3	29.699118	1	2	23.4500	S	1	0	...	0	0	0	0	1	0
889	890	1	1	26.000000	0	0	30.0000	C	0	1	...	0	0	0	0	0	0
890	891	0	3	32.000000	0	0	7.7500	Q	0	1	...	0	0	0	0	0	0

891 rows × 697 columns

```
x_Scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_Scaled.head()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Sex_female	Sex_male	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ti
0	0.000000	0.0	1.0	0.271174	0.125	0.0	0.014151	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.001124	1.0	0.0	0.472229	0.125	0.0	0.139136	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	

```
x_Scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_Scaled.head()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Sex_female	Sex_male	...	Ticket_STON/O2. 3101290	Ticket_SW/PP 751	Ticket_W./C. 14258	Ticket_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ti
0	0.000000	0.0	1.0	0.271174	0.125	0.0	0.014151	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.001124	1.0	0.0	0.472229	0.125	0.0	0.139136	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.002247	1.0	1.0	0.321438	0.000	0.0	0.015469	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.003371	1.0	0.0	0.434531	0.125	0.0	0.103644	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.004494	0.0	1.0	0.434531	0.000	0.0	0.015713	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 697 columns

```
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)
```

+ Code + Markdown

```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(712, 697) (179, 697) (712,) (179,)
```

et_W./C. 14263	Ticket_W./C. 6607	Ticket_W./C. 6608	Ticket_W./C. 6609	Ticket_W.E.P. 5734	Ticket_W/C 14208	Ticket_WE/P 5735
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0