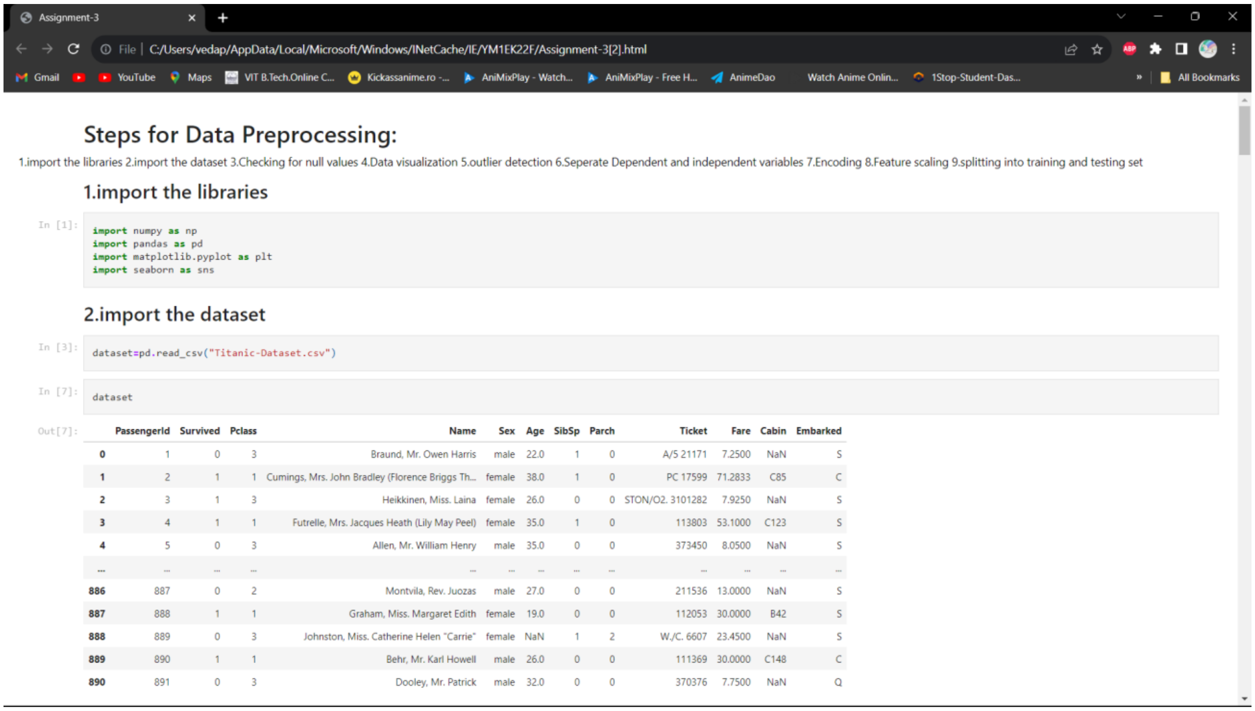
ASSINGNMENT-3



Steps for Data Preprocessing:

1.import the libraries 2.import the dataset 3.Checking for null values 4.Data visualization 5.outlier detection 6.Seperate Dependent and independent variables 7.Encoding 8.Feature scaling 9.splitting into training and testing set

1.import the libraries

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

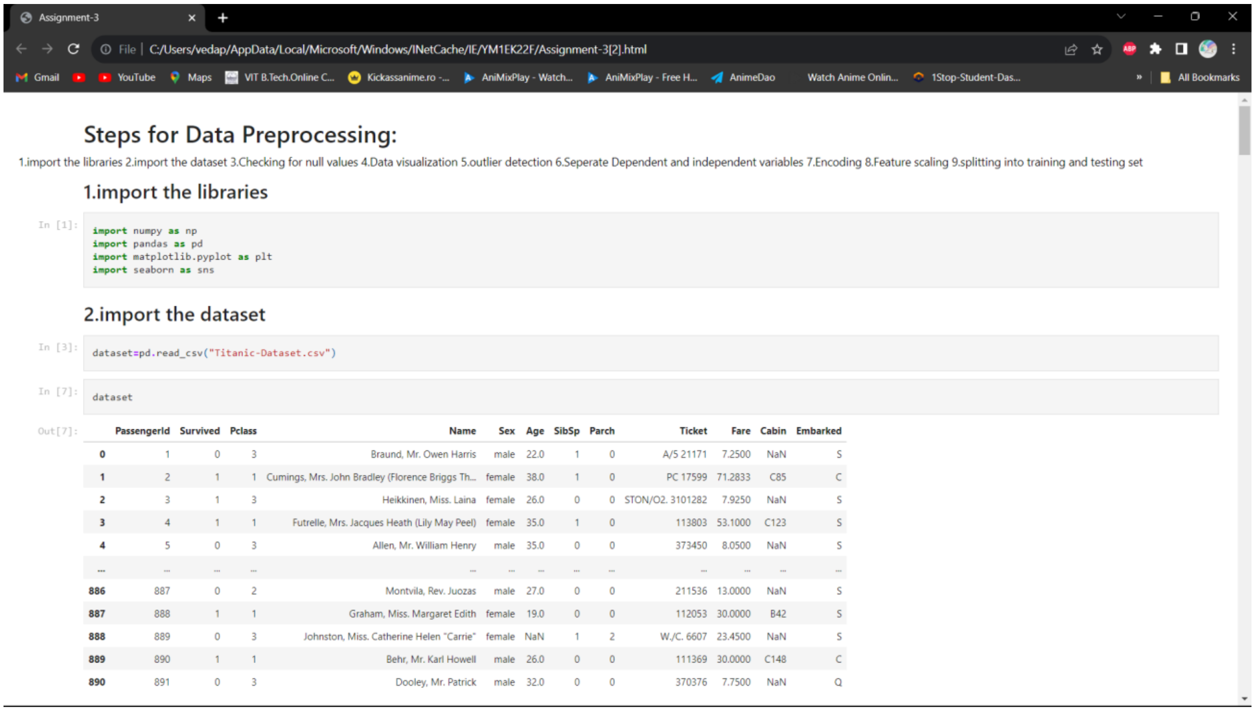
2.import the dataset

```
In [3]:
dataset=pd.read_csv("Titanic-Dataset.csv")
```

```
In [7]:
dataset
```

Out[7]:

	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


Steps for Data Preprocessing:

1.import the libraries 2.import the dataset 3.Checking for null values 4.Data visualization 5.outlier detection 6.Seperate Dependent and independent variables 7.Encoding 8.Feature scaling 9.splitting into training and testing set

1.import the libraries

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

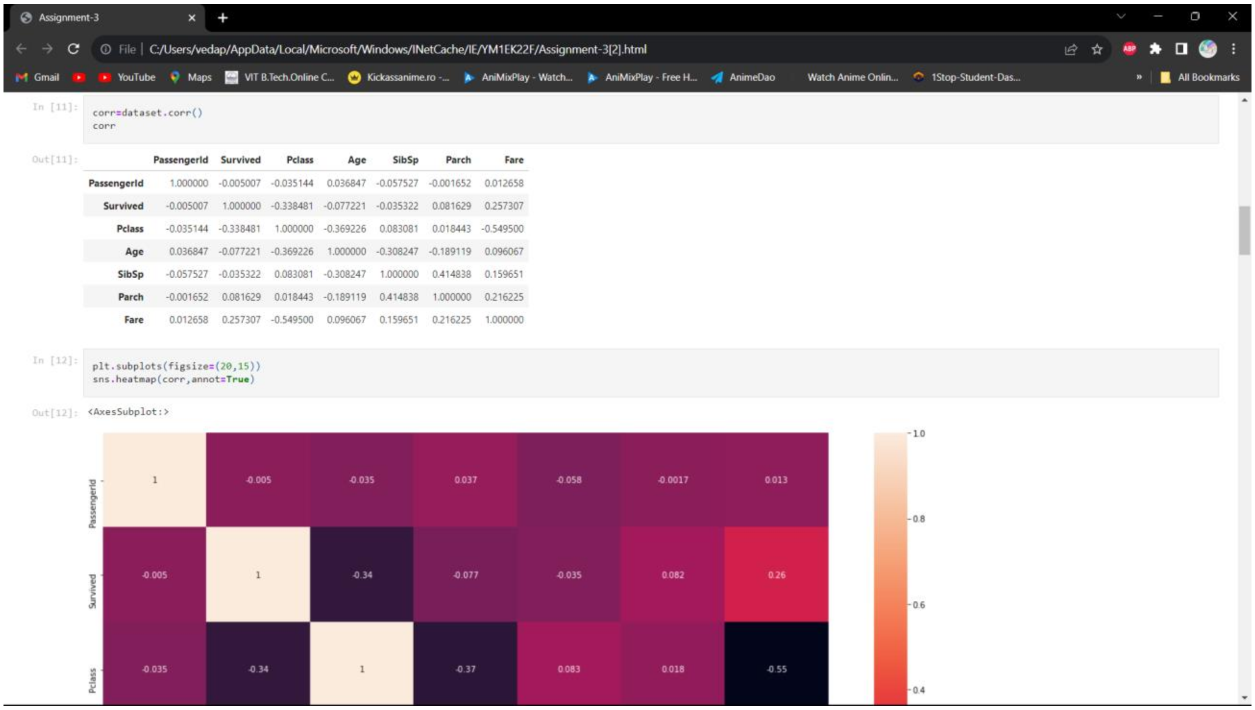
2.import the dataset

In [3]:
dataset=pd.read_csv("Titanic-Dataset.csv")

In [7]:
dataset

Out[7]:

	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

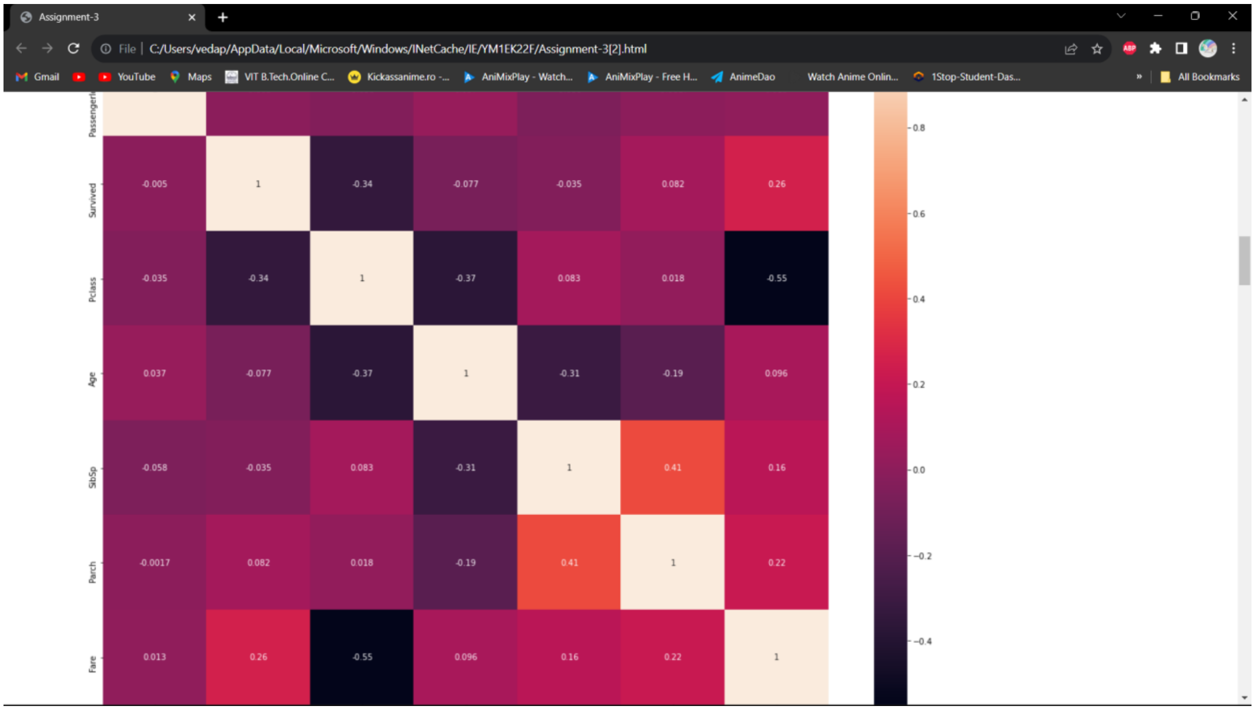


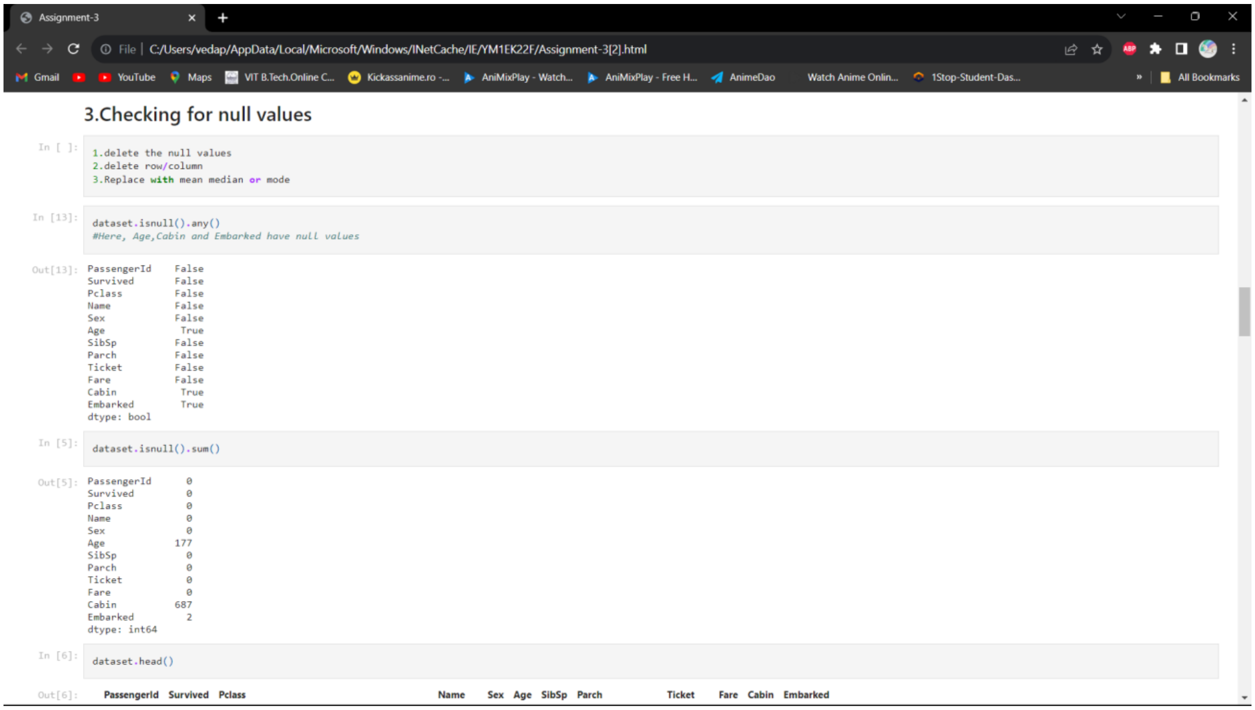
```
In [11]: corr=dataset.corr()
         corr
```

Out[11]:

	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

```
In [12]: plt.subplots(figsize=(20,15))
         sns.heatmap(corr,annot=True)

Out[12]: <AxesSubplot:>
```





3.Checking for null values

In [ ]:

```
1.delete the null values
2.delete row/column
3.Replace with mean median or mode
```

In [13]:

```
dataset.isnull().any()
#Here, Age,Cabin and Embarked have null values
```

Out[13]:

```
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
```

In [5]:

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

Out[5]:

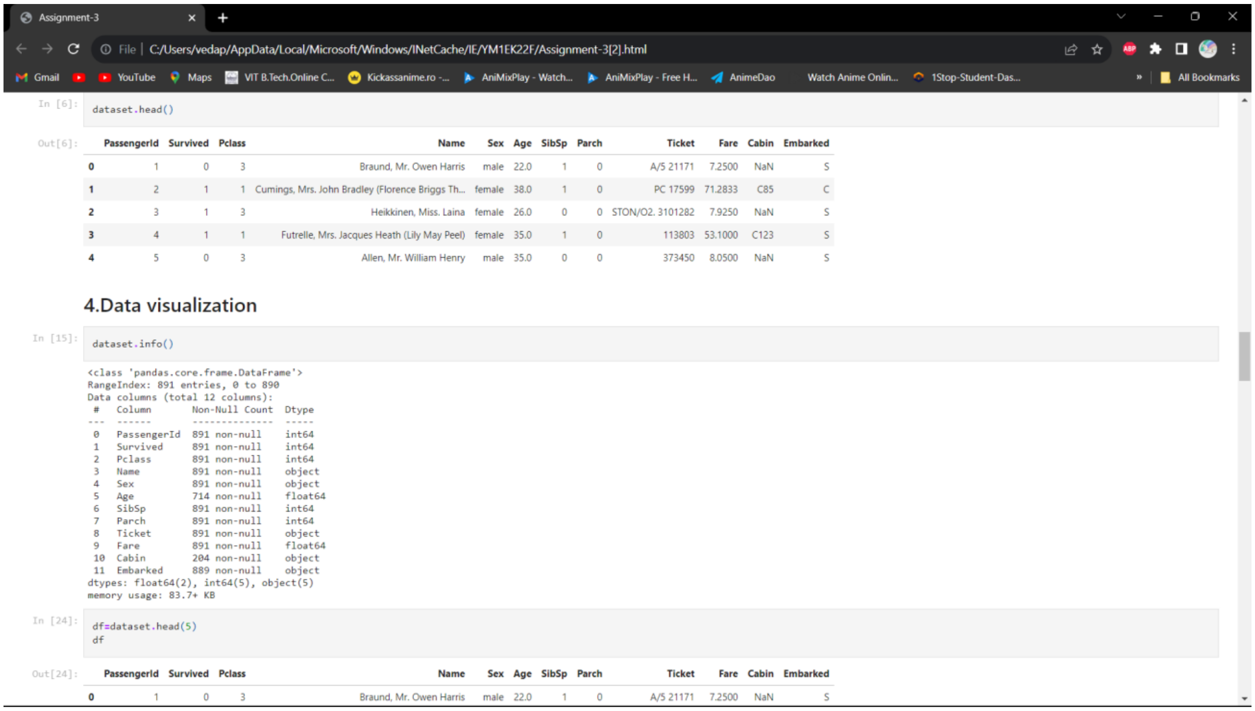
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [6]:

```
dataset.head()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked



In [6]:

```
dataset.head()
```

Out[6]:

	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

4.Data visualization

In [15]:

```
dataset.info()
```

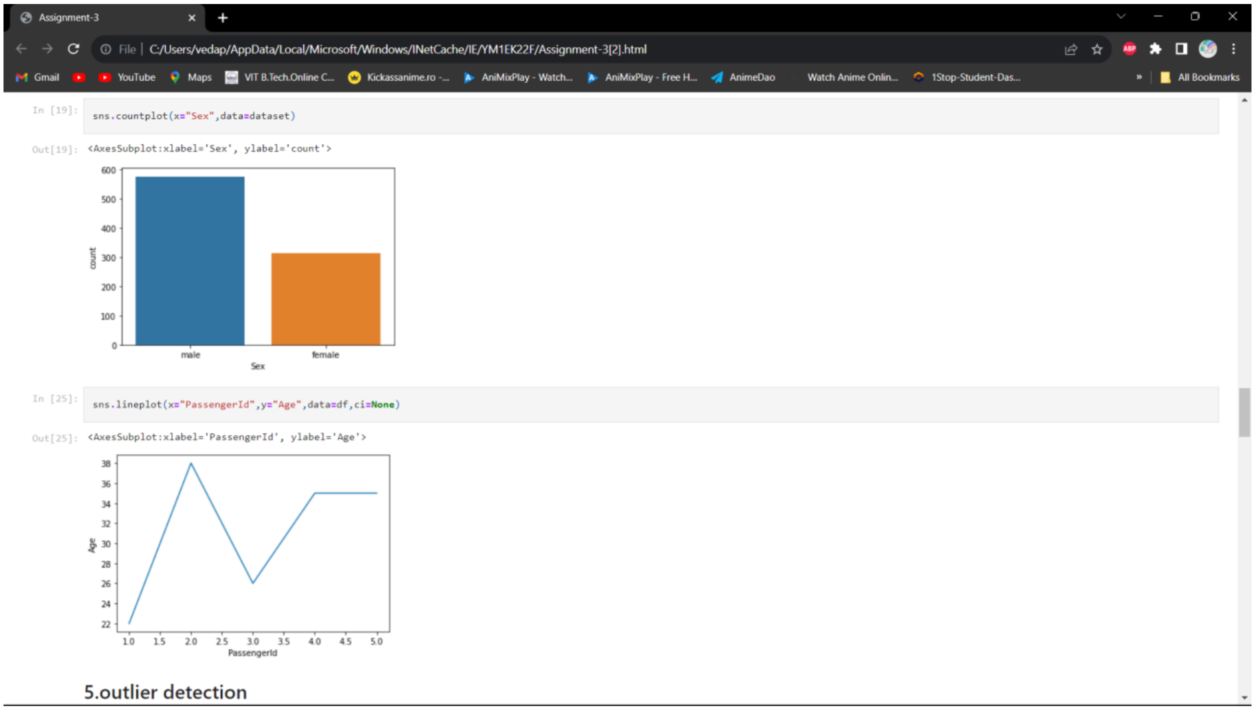
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    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
```

In [24]:

```
df=dataset.head(5)
df
```

Out[24]:

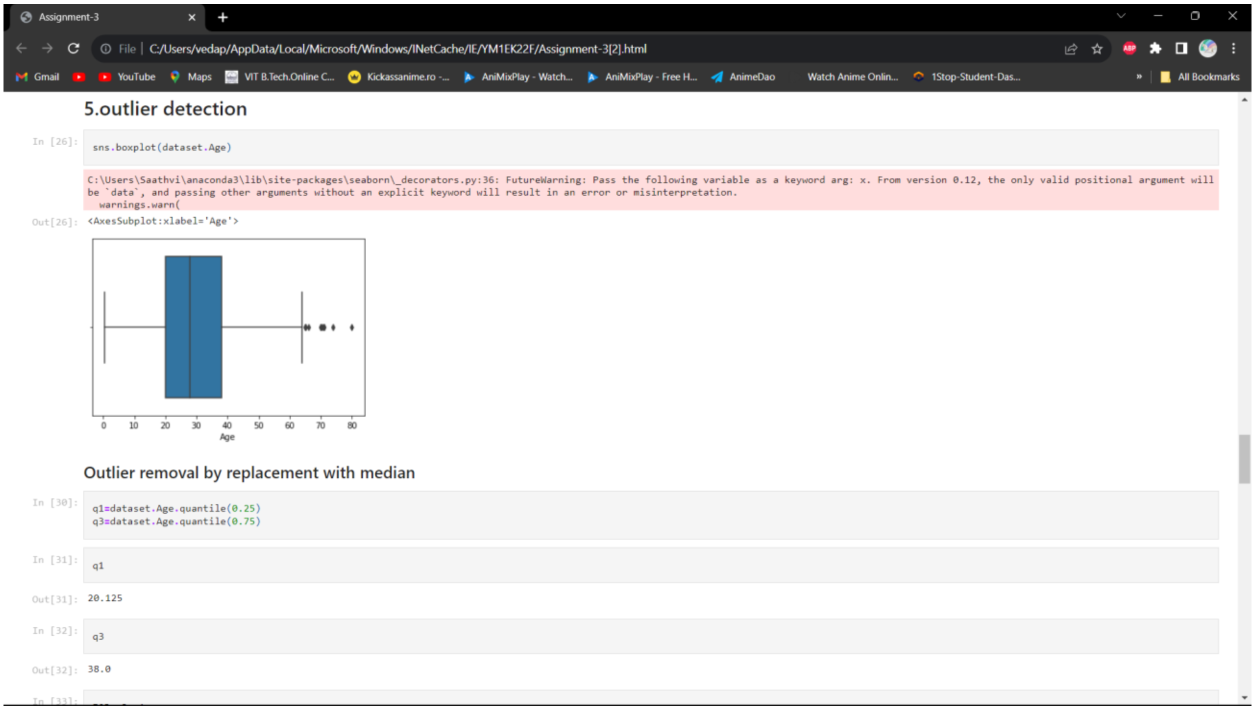
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S



```
In [19]: sns.countplot(x="Sex",data=dataset)

Out[19]: <AxesSubplot:xlabel='Sex', ylabel='count'>
```

```
In [25]: sns.lineplot(x="PassengerId",y="Age",data=df,ci=None)

Out[25]: <AxesSubplot:xlabel='PassengerId', ylabel='Age'>
```

5.outlier detection

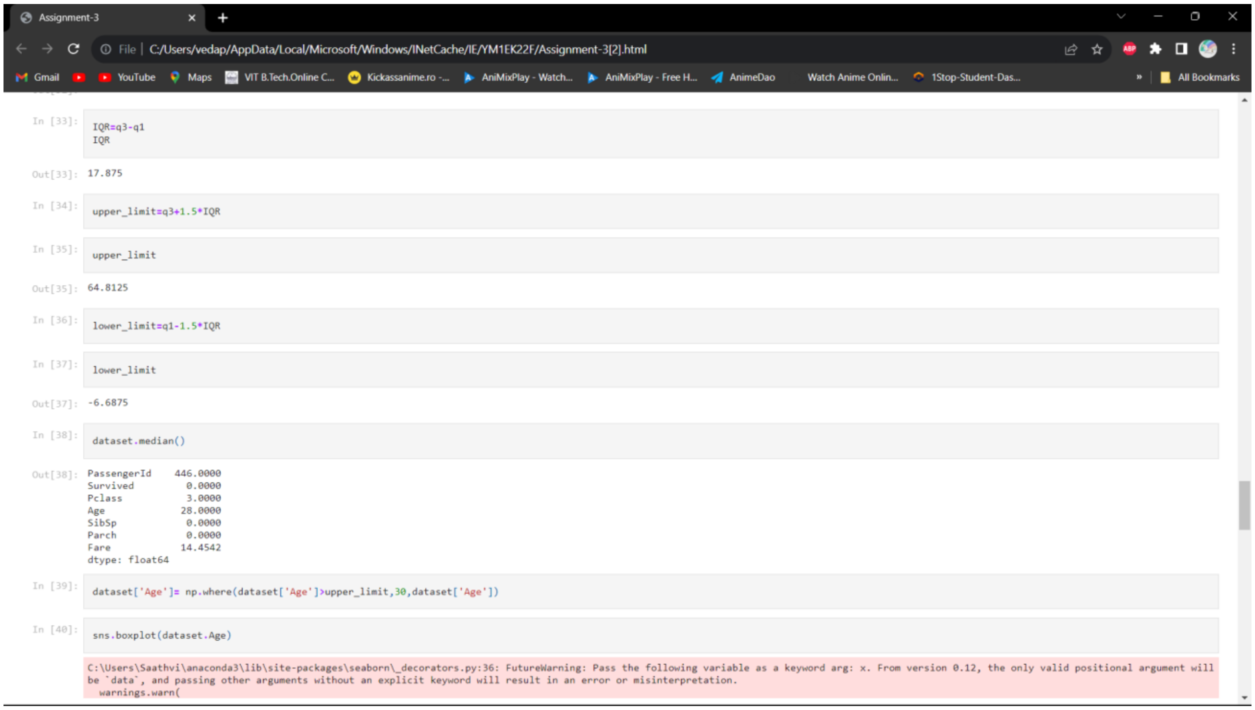


5.outlier detection

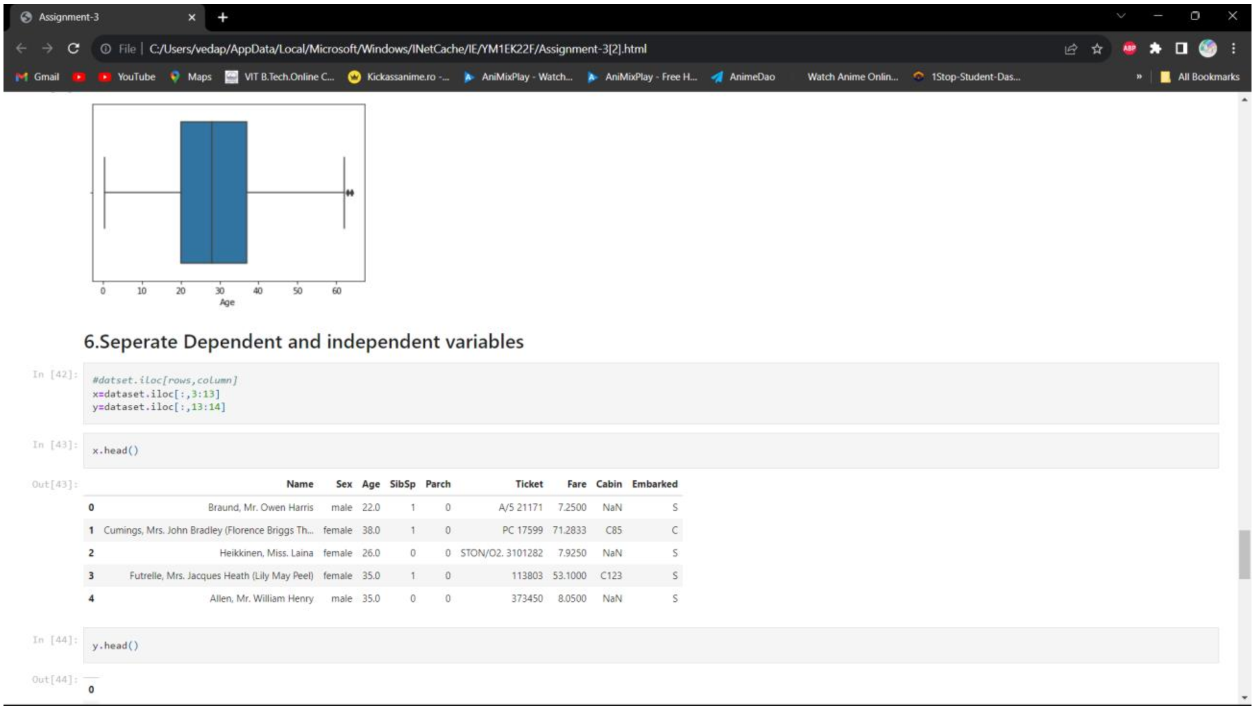
```
In [26]: sns.boxplot(dataset.Age)

C:\Users\Saathvi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[26]: <AxesSubplot:xlabel='Age'>
```

Outlier removal by replacement with median

```
In [30]: q1=dataset.Age.quantile(0.25)
         q3=dataset.Age.quantile(0.75)

In [31]: q1

Out[31]: 20.125

In [32]: q3

Out[32]: 38.0

In [33]:
```



```
In [33]: IQR=q3-q1
         IQR

Out[33]: 17.875

In [34]: upper_limit=q3+1.5*IQR

In [35]: upper_limit

Out[35]: 64.8125

In [36]: lower_limit=q1-1.5*IQR

In [37]: lower_limit

Out[37]: -6.6875

In [38]: dataset.median()

Out[38]: PassengerId    446.0000
         Survived         0.0000
         Pclass           3.0000
         Age             28.0000
         SibSp            0.0000
         Parch            0.0000
         Fare            14.4542
         dtype: float64

In [39]: dataset['Age']= np.where(dataset['Age']>upper_limit,30,dataset['Age'])

In [40]: sns.boxplot(dataset.Age)

         C:\Users\Saathvi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will
         be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
           warnings.warn(
```

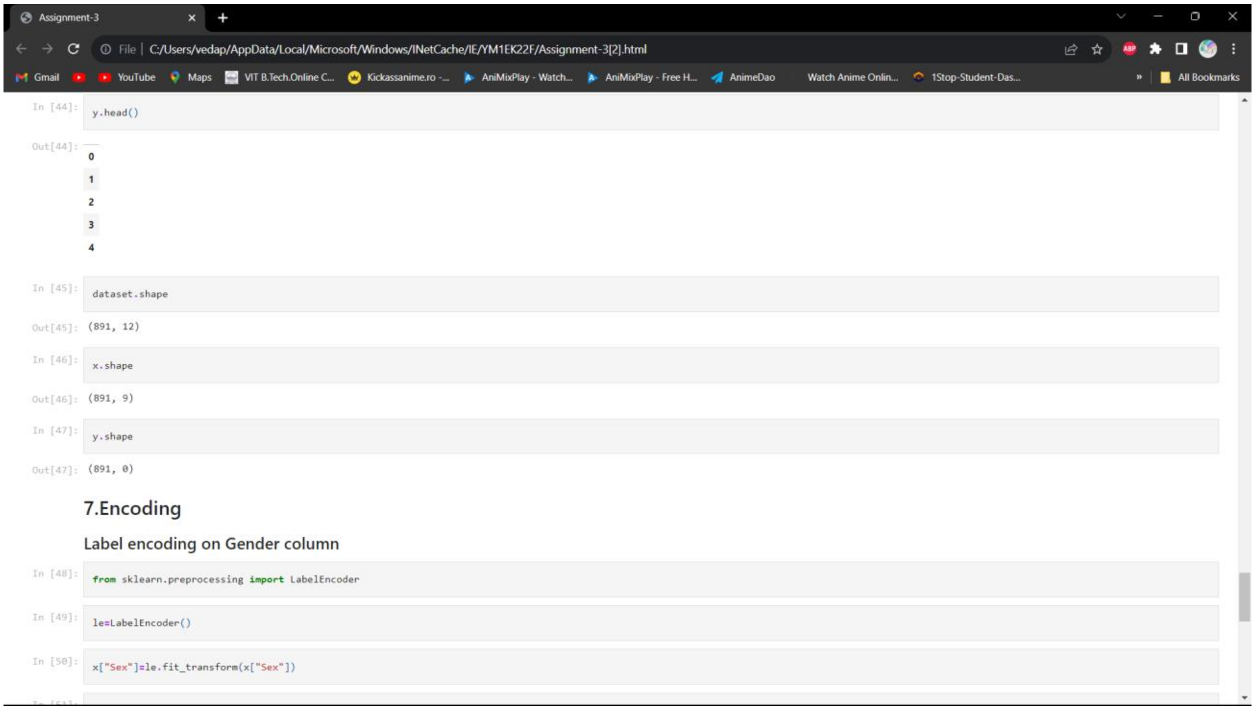


6.Seperate Dependent and independent variables

```
In [42]: #datset.iloc[rows,column]
         x=dataset.iloc[:,3:13]
         y=dataset.iloc[:,13:14]

In [43]: x.head()

Out[43]:
```

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
In [44]: y.head()

Out[44]:
         0
```

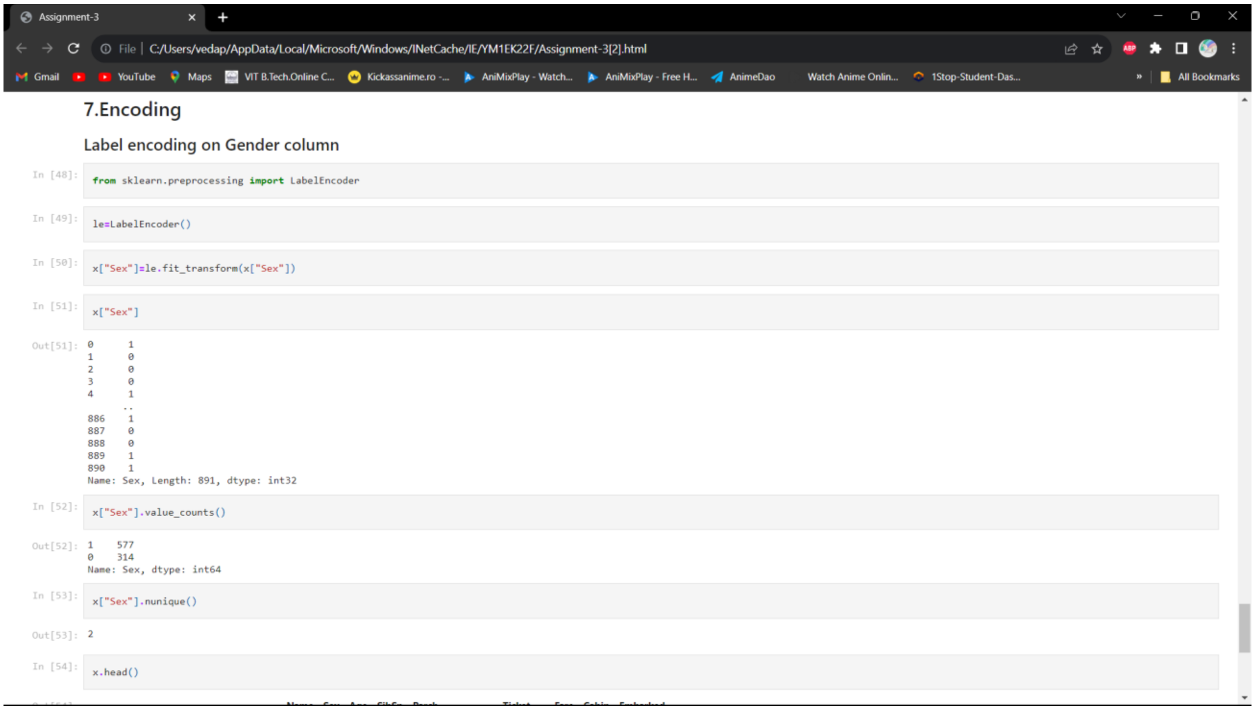


```
In [44]: y.head()

Out[44]:
0
1
2
3
4

In [45]: dataset.shape

Out[45]: (891, 12)

In [46]: x.shape

Out[46]: (891, 9)

In [47]: y.shape

Out[47]: (891, 0)
```

7.Encoding

Label encoding on Gender column

```
In [48]: from sklearn.preprocessing import LabelEncoder

In [49]: le=LabelEncoder()

In [50]: x["Sex"]=le.fit_transform(x["Sex"])
```

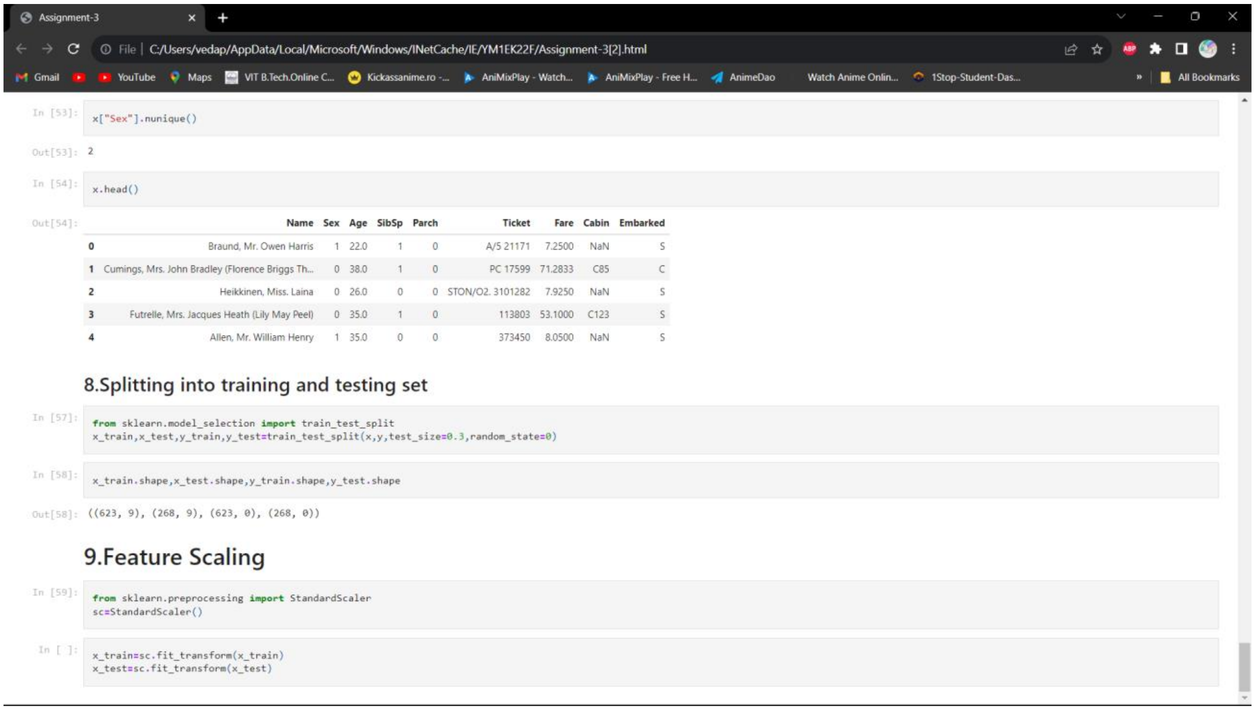


7.Encoding

Label encoding on Gender column

```
In [48]: from sklearn.preprocessing import LabelEncoder

In [49]: le=LabelEncoder()

In [50]: x["Sex"]=le.fit_transform(x["Sex"])
```

```
In [51]: x["Sex"]

Out[51]: 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

In [52]: x["Sex"].value_counts()

Out[52]: 1    577
         0    314
         Name: Sex, dtype: int64

In [53]: x["Sex"].nunique()

Out[53]: 2

In [54]: x.head()
```



```
In [53]: x["Sex"].nunique()
```

```
Out[53]: 2
```

```
In [54]: x.head()
```

Out[54]:

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S
4	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	NaN	S

8.Splitting into training and testing set

```
In [57]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
In [58]: x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
Out[58]: ((623, 9), (268, 9), (623, 0), (268, 0))
```

9.Feature Scaling

```
In [59]: from sklearn.preprocessing import StandardScaler
         sc=StandardScaler()
```

```
In [ ]: x_train=sc.fit_transform(x_train)
        x_test=sc.fit_transform(x_test)
```