```
ASSIGNMENT NO:1

AIM:
1. Introduction to Dataset
2. Python Libraries for Data Science
3. Description of Dataset
4. Panda Dataframe functions for load the dataset
5. Panda functions for Data Preprocessing
6. Panda functions for Data Formatting and Normalisation
7. Panda Functions for handling categorical variables

import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

df=sns.get_dataset_names()
print(data_set_name)
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri',
'geyser', 'glue', 'healthexp', 'iris', 'mpg', 'penguins', 'planets',
'seaice', 'taxis', 'tips', 'titanic', 'anagrams', 'anagrams',
'anscombe', 'anscombe', 'attention', 'attention', 'brain_networks',
'brain_networks', 'car_crashes', 'car_crashes', 'diamonds',
'diamonds', 'dots', 'dots', 'dowjones', 'dowjones', 'exercise',
'exercise', 'flights', 'flights', 'fmri', 'fmri', 'geyser', 'geyser',
'glue', 'glue', 'healthexp', 'healthexp', 'iris', 'iris', 'mpg',
'mpg', 'penguins', 'penguins', 'planets', 'planets', 'seaice',
'seaice', 'taxis', 'taxis', 'tips', 'tips', 'titanic', 'titanic',
'anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri',
'geyser', 'glue', 'healthexp', 'iris', 'mpg', 'penguins', 'planets',
'seaice', 'taxis', 'tips', 'titanic']
```

```
df=sns.load_dataset('titanic')
df
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0           0       3    male  22.0      1      0   7.2500        S
Third
1           1       1  female  38.0      1      0  71.2833        C
First
2           1       3  female  26.0      0      0   7.9250        S
Third
3           1       1  female  35.0      1      0  53.1000        S
```

```
First
4           0       3    male  35.0      0      0   8.0500         S
Third
..        ...     ...     ...   ...    ...    ...      ...       ...
...
886         0       2    male  27.0      0      0  13.0000         S
Second
887         1       1  female  19.0      0      0  30.0000         S
First
888         0       3  female   NaN      1      2  23.4500         S
Third
889         1       1    male  26.0      0      0  30.0000         C
First
890         0       3    male  32.0      0      0   7.7500         Q
Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]

data1=df.head()
data1

   survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0         0       3    male  22.0      1      0   7.2500        S
Third
1         1       1  female  38.0      1      0  71.2833        C
First
2         1       3  female  26.0      0      0   7.9250        S
Third
3         1       1  female  35.0      1      0  53.1000        S
First
4         0       3    male  35.0      0      0   8.0500        S
Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
```

```
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True

data2=df.tail()
data2

     survived  pclass     sex   age  sibsp  parch   fare embarked
class  \
886         0       2    male  27.0      0      0  13.00        S
Second
887         1       1  female  19.0      0      0  30.00        S
First
888         0       3  female   NaN      1      2  23.45        S
Third
889         1       1    male  26.0      0      0  30.00        C
First
890         0       3    male  32.0      0      0   7.75        Q
Third

       who  adult_male deck  embark_town alive  alone
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

data3=df.info()
data3

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
```

```
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
data4=df['sex'].value_counts(normalize=True)
data4
```

```
sex
male      0.647587
female    0.352413
Name: proportion, dtype: float64
```

```
data5=df.describe()
data5
```

```
         survived      pclass         age       sibsp       parch
fare
count  891.000000  891.000000  714.000000  891.000000  891.000000
891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594
32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057
49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000
0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000
7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000
14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000
31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000
512.329200
```

```
data6=df["deck"].value_counts(normalize=True)
data6
```

```
deck
C    0.290640
B    0.231527
D    0.162562
E    0.157635
A    0.073892
F    0.064039
G    0.019704
Name: proportion, dtype: float64
```

```
data7=df.drop(["deck"], axis=1)
data7
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0           0       3    male  22.0      1      0   7.2500        S
Third
1           1       1  female  38.0      1      0  71.2833        C
First
2           1       3  female  26.0      0      0   7.9250        S
Third
3           1       1  female  35.0      1      0  53.1000        S
First
4           0       3    male  35.0      0      0   8.0500        S
Third
..        ...     ...     ...   ...    ...    ...      ...      ...
...
886         0       2    male  27.0      0      0  13.0000        S
Second
887         1       1  female  19.0      0      0  30.0000        S
First
888         0       3  female   NaN      1      2  23.4500        S
Third
889         1       1    male  26.0      0      0  30.0000        C
First
890         0       3    male  32.0      0      0   7.7500        Q
Third

       who  adult_male  embark_town alive  alone
0      man        True  Southampton    no  False
1    woman       False    Cherbourg   yes  False
2    woman       False  Southampton   yes   True
3    woman       False  Southampton   yes  False
4      man        True  Southampton    no   True
..     ...         ...          ...   ...    ...
886    man        True  Southampton    no   True
887  woman       False  Southampton   yes   True
888  woman       False  Southampton    no  False
889    man        True    Cherbourg   yes   True
890    man        True   Queenstown    no   True

[891 rows x 14 columns]
```

```
data8=df.drop(["embarked","class","who","adult_male","deck","embark_to
wn","alone"],axis=1)
data8
```

```
     survived  pclass     sex   age  sibsp  parch     fare alive
0           0       3    male  22.0      1      0   7.2500    no
1           1       1  female  38.0      1      0  71.2833   yes
2           1       3  female  26.0      0      0   7.9250   yes
3           1       1  female  35.0      1      0  53.1000   yes
4           0       3    male  35.0      0      0   8.0500    no
..        ...     ...     ...   ...    ...    ...      ...   ...
```

```
886         0       2    male  27.0      0      0  13.0000    no
887         1       1  female  19.0      0      0  30.0000   yes
888         0       3  female   NaN      1      2  23.4500    no
889         1       1    male  26.0      0      0  30.0000   yes
890         0       3    male  32.0      0      0   7.7500    no

[891 rows x 8 columns]
```

```
data9=df['sex'].mode()[0]
data9
```

```
'male'
```

```
data10=df['age'].mode
data10
```

```
<bound method Series.mode of 0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: age, Length: 891, dtype: float64>
```

```
data11=df['age'].mean
data11
```

```
<bound method Series.mean of 0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: age, Length: 891, dtype: float64>
```

```
data12=df.loc[:,"sex"].mode()
data12
```

```
0    male
Name: sex, dtype: object
```

```
bool_series = pd.notnull(df["sex"])
bool_series
```

```
0      True
1      True
2      True
3      True
4      True
       ...
886    True
887    True
888    True
889    True
890    True
Name: sex, Length: 891, dtype: bool
```

```
df['age'].fillna(df['age'].mean(), inplace=True)
```

```
data15=df.info()
data15
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          891 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     891 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

Shreyash Tanaji Patil
Batch-B3
Roll no: 13266