```
ASSIGNMENT NO:1
AIM:

    Introduction to Dataset

Python Libraries for Data Science
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', 'brain_networks',
'brain_networks', 'car_crashes', 'car_crashes', 'diamonds', 'diamonds', 'dots', 'dowjones', 'dowjones', 'exercise',
'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
                              3 male 22.0
                                                              1
                                                                            7.2500
                                                                                                    S
0
                                                                         0
Third
1
                  1
                              1 female 38.0
                                                              1
                                                                         0 71.2833
First
2
                              3 female 26.0
                                                                               7.9250
                                                                                                    S
Third
3
                  1
                              1 female 35.0
                                                              1
                                                                         0 53.1000
```

First										
4	0	3	ma	le 35.0)	0	0	8.050	90	S
Third										
886	0	2	ma	le 27.0	9	0	0	13.000	90	S
Second										
887	1	1	fema	le 19.0	9	0	0	30.000	90	S
First										
888	0	3	fema	le NaM	V	1	2	23.450	90	S
Third	-	_			-	_				
889	1	1	ma	le 26.0	.)	0	0	30.000	90	C
First		_	ilia	2010	,	U	J	30.000	,,,	C
890	0	3	ma	le 32.0		0	0	7.750	00	Q
Third	U	3	IIIa	LC 32.1	,	U	U	7.750	,,	Ų
IIIII U										
who	, adı	ılt male	dock	omharl	k town	aliva	ald	nno		
0 mar		True	NaN		ampton	no	_			
1 womar		False	C		rbourg		Fal			
2 Woman		False	NaN		ampton	yes		rue		
2 womar3 womar		False	C		•	yes				
_					ampton	yes	Fal			
		True	NaN	South	ampton	no		`ue		
		T 1011 0		C a ± la .				• •		
886 mar		True	NaN		ampton	no		rue		
887 woman		False	В		ampton	yes		rue		
888 woman		False	NaN		ampton	no	Fal			
889 mar		True	C		rbourg	yes		·ue		
890 mar	1	True	NaN	Queer	nstown	no	11	·ue		
[001 8016	v 1E	columns	1							
[891 rows	X 12	Coculins	J							
data1=df.h	nead()	\								
data1	icau ()									
datai										
survive	ed po	class	sex	age	sibsp	parc	h	fare	embarked	
class \				3 -	F	P				
0	0	3	male	22.0	1		0 7	7.2500	S	
Third		J			_			500	J	
1	1	1 fe	emale	38.0	1		0 71	1.2833	С	
First	_		Silia CC	30.0	_		0 , 1	112033	Č	
2	1	3 fe	emale	26.0	0		0 7	7.9250	S	
Third	_	5 10	Silia CC	20.0	U		0 ,	. 3230	5	
3	1	1 fe	emale	35.0	1		0 53	3.1000	S	
5 First	1	T 16	-ilia Le	۵. د د	1		0 53). TOOO	3	
	0	2	mala	3E 0	0		0 8	0500	S	
4 Third	0	3	male	35.0	0		U (3.0500	5	
Third										
who	2 du 1 +	male de	ack .	embark †	town - 1	ivo	alone	,		
	auutl	_		_			False			
0 man				Southamp						
1 woman		False	С	Cherbo	burg	yes	False			

<pre>2 woman 3 woman 4 man</pre>	False False True	C So	uthampton uthampton uthampton	yes yes no	True False True					
data2=df.tail() data2										
surviv	ed pclass	sex	age si	bsp pa	arch	fare	embarked			
886 Second	0 2	male	27.0	0	0 1	3.00	S			
887	1 1	female	19.0	0	0 3	0.00	S			
First 888	0 3	female	NaN	1	2 2	3.45	S			
Third 889	1 1	male	26.0	0	0 3	0.00	С			
First 890	0 3	male	32.0	0	0	7.75	Q			
Third	0 3	ma cc	3210		Ū	, , , ,	4			
886 man 887 woman 888 woman 889 man 890 man	Tru Fals Fals Tru Tru	e B e NaN e C	Southampto Southampto Southampto Cherbour Queenstow	n yes n no g yes	s Tru o Fals s Tru	e e e				
<pre>data3=df.info() data3</pre>										
<pre><class 'pandas.core.frame.dataframe'=""> RangeIndex: 891 entries, 0 to 890 Data columns (total 15 columns): # Column Non-Null Count Dtype</class></pre>										
0 surviv 1 pclass 2 sex 3 age 4 sibsp 5 parch 6 fare 7 embark 8 class 9 who 10 adult_ 11 deck 12 embark 13 alive 14 alone	red 891 891 891 714 891 891 891 891 male 891 203 4_town 889	non-nul	l int64 l int64 l object l float l int64 l int64 l float l object l categ l bool l categ l object	t 64 64 t ory t						

```
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
                                                             parch
         survived
                       pclass
                                      age
                                                 sibsp
fare
count 891.000000
                   891.000000 714.000000
                                           891.000000 891.000000
891.000000
                                29.699118
                     2.308642
                                             0.523008
                                                          0.381594
mean
         0.383838
32,204208
                     0.836071
                                14.526497
                                                          0.806057
std
         0.486592
                                             1.102743
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
         1.000000
                     3.000000
                                80.000000
                                             8.000000
                                                          6.000000
512.329200
data6=df["deck"].value counts(normalize=True)
data6
deck
     0.290640
C
В
     0.231527
D
     0.162562
Е
     0.157635
Α
     0.073892
F
     0.064039
G
     0.019704
Name: proportion, dtype: float64
data7=df.drop(["deck"], axis=1)
data7
                                age sibsp parch fare embarked
     survived pclass sex
class \
```

0 Third	0	3	male	22.0	1	0	7.2500	S
1 First	1	1	female	38.0	1	0	71.2833	С
2	1	3	female	26.0	0	0	7.9250	S
Third 3 First	1	1	female	35.0	1	0	53.1000	S
4 Third	0	3	male	35.0	0	0	8.0500	S
886 Second	0	2	male	27.0	0	0	13.0000	S
887 First	1	1	female	19.0	0	0	30.0000	S
888 Third	0	3	female	NaN	1	2	23.4500	S
889	1	1	male	26.0	0	0	30.0000	С
First 890 Third	0	3	male	32.0	0	0	7.7500	Q
who man woman woman woman man man woman man woman woman woman woman woman	F F	male True alse alse True True alse	embark Southa Cher Southa Southa Southa Southa	mpton bourg mpton mpton mpton mpton	alive no yes yes yes no no yes	alone False True False True True True		

[891 rows x 14 columns]

False

True

True

woman

man

man

888

889

890

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

no

yes

False

True

True

	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

Southampton

Cherbourg

Queenstown no

```
886
            0
                     2
                          male
                                27.0
                                           0
                                                     13.0000
                                                                 no
887
            1
                     1
                       female
                                19.0
                                           0
                                                     30.0000
                                                  0
                                                                yes
888
            0
                     3 female
                                 NaN
                                           1
                                                  2
                                                     23.4500
                                                                 no
                     1
                                                     30.0000
889
            1
                          male
                                26.0
                                           0
                                                  0
                                                                yes
            0
                     3
                                                  0
890
                          male 32.0
                                           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</pre>
                                     22.0
1
       38.0
2
       26.0
3
       35.0
4
       35.0
886
       27.0
       19.0
887
888
       NaN
889
       26.0
       32.0
890
Name: age, Length: 891, dtype: float64>
data11=df['age'].mean
data11
<bound method Series.mean of 0</pre>
       38.0
1
2
       26.0
3
       35.0
4
       35.0
       . . .
886
       27.0
       19.0
887
       NaN
888
       26.0
889
890
       32.0
Name: age, Length: 891, dtype: float64>
data12=df.loc[:,"sex"].mode()
data12
     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
                  891 non-null
                                   int64
     pclass
 2
                                   object
     sex
                  891 non-null
3
                  891 non-null
                                   float64
     age
 4
                                   int64
     sibsp
                  891 non-null
 5
                                   int64
     parch
                  891 non-null
 6
                                   float64
     fare
                  891 non-null
 7
                  891 non-null
     embarked
                                   object
 8
                  891 non-null
                                   category
     class
 9
                                   object
     who
                  891 non-null
 10
    adult male
                  891 non-null
                                   bool
 11
     deck
                  203 non-null
                                   category
12
     embark_town 889 non-null
                                   object
13
                  891 non-null
                                   object
     alive
14
                  891 non-null
                                   bool
     alone
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

Saurav Ratilal Raysing B3

Roll no: 13267