

IoT Domain Analyst
CHALLENGING TASK – 2

Name:- PANDUGA VENKATA JAYA SRIKANTH REDDY

Reg No:- 21MIS1095

Question 3

Q3	<p>Perform the following preprocessing data</p> <ul style="list-style-type: none">o Remove wrong entries datao Remove null values <p>List out the survived female passengers whose age is between 30 to 40</p> <p>List out the passengers whose cabin details are C series.</p>
----	--

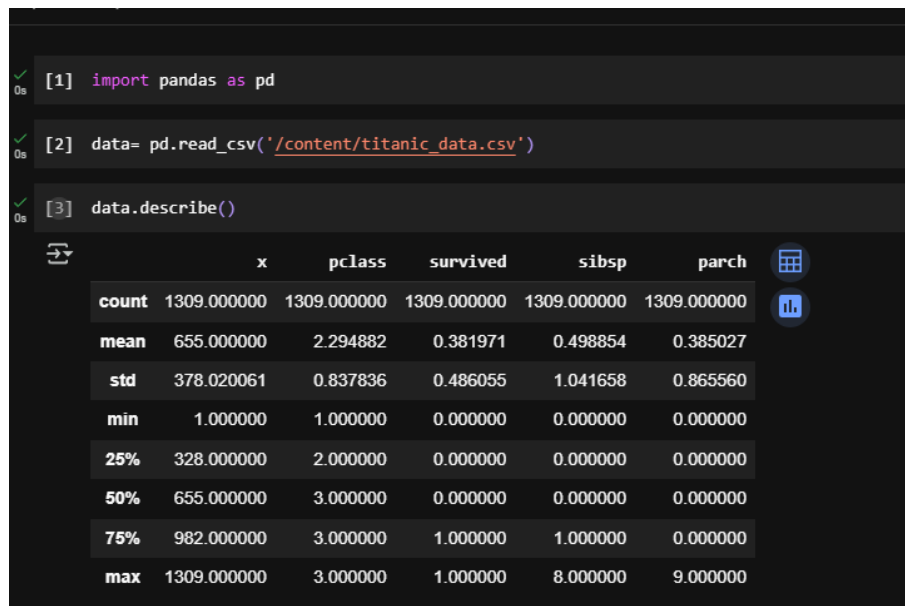
AIM:-

To perform the preprocessing on given titanic dataset and to clean the dataset by removing the wrong entries and null values from the dataset. Also to display the survived female passengers whose age is between 30 to 40 and the passengers whose cabin details are C series.

PROCEDURE:-

- 1) Import the data using pandas
- 2) Describe the dataset, to know its structure and values
- 3) Check if the dataset contains any null values
- 4) Check if there is any wrong entry in the data for numerical columns where the data has AGE, FARE. The values should be > 0.
- 5) The values provided in the dataset are in string format. So, convert them into INTEGER or FLOAT Values.
- 6) Check for the NULL Values in the dataset. All the values should be as na=FALSE, then there are no null values in dataset. If Null values found, remove them.
- 7) Now find the survived female passengers whose age is between 30 to 40
- 8) Now find the passengers whose cabin details are C series.

Initial Data Loading Output



The screenshot shows a Jupyter Notebook with three code cells. The first cell imports pandas as pd. The second cell reads a CSV file from '/content/titanic_data.csv' into a variable named 'data'. The third cell calls 'data.describe()' to display a summary of the data. The output is a table with 6 columns: x, pclass, survived, sibsp, and parch. The rows represent statistical measures: count, mean, std, min, 25%, 50%, 75%, and max.

	x	pclass	survived	sibsp	parch
count	1309.000000	1309.000000	1309.000000	1309.000000	1309.000000
mean	655.000000	2.294882	0.381971	0.498854	0.385027
std	378.020061	0.837836	0.486055	1.041658	0.865560
min	1.000000	1.000000	0.000000	0.000000	0.000000
25%	328.000000	2.000000	0.000000	0.000000	0.000000
50%	655.000000	3.000000	0.000000	0.000000	0.000000
75%	982.000000	3.000000	1.000000	1.000000	0.000000
max	1309.000000	3.000000	1.000000	8.000000	9.000000

TASK – 1: REMOVE WRONG ENTRIES DATA

CODE:-

```
data = data[data['age'] >= 0]
data = data[data['fare'] >= 0]
data = data[data['pclass'].isin([1, 2, 3])]

data = data[data['name'].notnull()]

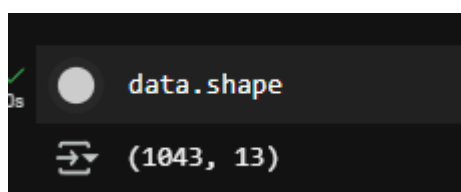
data = data[data['ticket'].notnull()]
data = data[data['cabin'].notnull()]
data = data[data['sibsp'] >= 0]
data = data[data['parch'] >= 0]

data = data[data['embarked'].isin(['C', 'Q', 'S'])]

print(f"Data after removing wrong entries: {data.shape[0]} rows")
```

Output:-

Before Removing the wrong entries



The screenshot shows a Jupyter Notebook cell with the code 'data.shape'. The output is '(1043, 13)', indicating 1043 rows and 13 columns.

```
data.shape
(1043, 13)
```

After Removing:

```
data = data[data['age'] >= 0]
data = data[data['fare'] >= 0]

data = data[data['pclass'].isin([1, 2, 3])]

data = data[data['name'].notnull()]

data = data[data['ticket'].notnull()]
data = data[data['cabin'].notnull()]

data = data[data['sibsp'] >= 0]
data = data[data['parch'] >= 0]

data = data[data['embarked'].isin(['C', 'Q', 'S'])]

print(f"Data after removing wrong entries: {data.shape[0]} rows")
```

Data after removing wrong entries: 1043 rows

TASK – 2: REMOVE NULL VALUES

Code: To check if there are null values

```
data.isnull()
```

data.isnull()

	x	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	home.dest
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...
1301	False	False	False	False	False	False	False	False	False	False	False	False	False
1304	False	False	False	False	False	False	False	False	False	False	False	False	False
1306	False	False	False	False	False	False	False	False	False	False	False	False	False
1307	False	False	False	False	False	False	False	False	False	False	False	False	False
1308	False	False	False	False	False	False	False	False	False	False	False	False	False

1043 rows x 13 columns

Code: To remove null Values

```
data = data.dropna()
```

Output:-

```
↩ Data after removing wrong entries: 1043 rows

0s
data = data.dropna()

print(f"Data after removing rows with null values: {data.shape[0]} rows")

↩ Data after removing rows with null values: 1043 rows
```

TASK - 3: LIST OUT THE SURVIVED FEMALE PASSENGERS WHOSE AGE IS BETWEEN 30 TO 40

Code:

```
survived_female_30_40 = data[(data['sex'] == 'female') &
                              (data['survived'] == 1) &
                              (data['age'] >= 30) &
                              (data['age'] <= 40)]

print(survived_female_30_40[['name', 'age', 'pclass', 'fare',
                              'embarked', 'home.dest']])
```

Output:

```
0s
survived_female_30_40 = data[(data['sex'] == 'female') &
                              (data['survived'] == 1) &
                              (data['age'] >= 30) &
                              (data['age'] <= 40)]

print(survived_female_30_40[['name', 'age', 'pclass', 'fare', 'embarked', 'home.dest']])

↩
```

		name	age	pclass	fare	\
18		Bazzani, Miss. Albina	32.0	1	76.2917	
28		Bissette, Miss. Amelia	35.0	1	135.6333	
32		Bonnell, Miss. Caroline	30.0	1	164.8667	
57		Carter, Mrs. William Ernest (Lucile Polk)	36.0	1	120.0000	
65		Chambers, Mrs. Norman Campbell (Bertha Griggs)	33.0	1	53.1000	
...		
765		Dean, Mrs. Bertram (Eva Georgetta Light)	33.0	3	20.5750	
778		Dowdell, Miss. Elizabeth	30.0	3	12.4750	
823		Goldsmith, Mrs. Frank John (Emily Alice Brown)	31.0	3	20.5250	
1094		Osman, Mrs. Mara	31.0	3	8.6833	
1286		Whabee, Mrs. George Joseph (Shawneene Abi-Saab)	38.0	3	7.2292	
	embarked	home.dest				
18	C	?				
28	S	?				
32	S	Youngstown, OH				
57	S	Bryn Mawr, PA				
65	S	New York, NY / Ithaca, NY				
...				
765	S	Devon, England Wichita, KS				
778	S	Union Hill, NJ				
823	S	Strood, Kent, England Detroit, MI				
1094	S	?				
1286	C	?				

[72 rows x 6 columns]

TASK – 4: LIST OUT THE PASSENGERS WHOSE CABIN DETAILS ARE C SERIRS.

Code:

```
c_series_passengers = data[data['cabin'].str.startswith('C', na=False)]  
print(c_series_passengers[['name', 'pclass', 'sex', 'age', 'cabin']])
```

Output:

```
c_series_passengers = data[data['cabin'].str.startswith('C', na=False)]  
print(c_series_passengers[['name', 'pclass', 'sex', 'age', 'cabin']])
```

	name	pclass	sex	age	\
1	Allison, Master. Hudson Trevor	1	male	0.9167	
2	Allison, Miss. Helen Loraine	1	female	2.0000	
3	Allison, Mr. Hudson Joshua Creighton	1	male	30.0000	
4	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	1	female	25.0000	
8	Appleton, Mrs. Edward Dale (Charlotte Lamson)	1	female	53.0000	
..	
309	Wick, Miss. Mary Natalie	1	female	31.0000	
312	Widener, Mr. George Dunton	1	male	50.0000	
313	Widener, Mr. Harry Elkins	1	male	27.0000	
314	Widener, Mrs. George Dunton (Eleanor Elkins)	1	female	50.0000	
322	Young, Miss. Marie Grice	1	female	36.0000	

	cabin
1	C22 C26
2	C22 C26
3	C22 C26
4	C22 C26
8	C101
..	...
309	C7
312	C80
313	C82
314	C80
322	C32

[86 rows x 5 columns]