12. PANDAS – Handling missing or NaN values

Contents

12. PANDAS – Handling missing or NaN values

1. NaN Value

- ✓ The full form of NaN is **N**ot **a** **N**umber
- ✓ The purpose of NaN is, to represent the missing values in data.
- ✓ The data type of NaN is float.
- ✓ While loading csv file, if file having missing values then it will be considered as NaN values.
- ✓ During data analysis we need to handle these NaN values.
    - For Example, suppose different users being surveyed may choose not to share their income, some user may choose not to share the address in this way many datasets went missing.

None and **NaN**

- ✓ None : None is a Python object which is holding nothing
- ✓ NaN : NaN is a pandas related object which represents missing data

2. Creating a DataFrame by loading csv file

- ✓ We can create DataFrame by loading csv file
- ✓ The given fruits.csv file having missing values.
- ✓ Kindly observe the missing/NaN values in DataFrame.

Program Name: Loading fruits csv file
Input file: demo1.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")

print(df1)
```

Output

```
     Order Id  Apples  Orange  Banana
0       16723     9.0     NaN     6.0
1       16724    12.0    12.0     6.0
2       16725     9.0     NaN     NaN
3       16726    12.0     NaN    24.0
4       16727    30.0     8.0    10.0
..        ...     ...     ...     ...
145     16868    24.0    12.0     6.0
146     16869     8.0     8.0    12.0
147     16870     5.0    30.0    30.0
148     16871    24.0     NaN    10.0
149     16872     5.0    30.0    10.0

[150 rows x 4 columns]
```

3. isna() and isnull() method – Checking NaN values

- ✓ isna() and isnull() are a predefined methods in DataFrame
- ✓ We can access these methods by using DataFrame object.
- ✓ By using these methods we can check missing values exist in DataFrame or not.
- ✓ If missing values are available then it return as True, otherwise False

Program Name: isna() method
Input file: demo2.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.isna()

print(df1.head())
print()
print(df2.head())
```

Output

```
   Order Id  Apples  Orange  Banana
0     16723     9.0     NaN     6.0
1     16724    12.0    12.0     6.0
2     16725     9.0     NaN     NaN
3     16726    12.0     NaN    24.0
4     16727    30.0     8.0    10.0

   Order Id  Apples  Orange  Banana
0     False   False    True   False
1     False   False   False   False
2     False   False    True    True
3     False   False    True   False
4     False   False   False   False
```

Program Name: isnull() method
Input file: demo3.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.isnull()

print(df1.head())
print()
print(df2.head())
```

Output

```
   Order Id  Apples  Orange  Banana
0     16723     9.0     NaN     6.0
1     16724    12.0    12.0     6.0
2     16725     9.0     NaN     NaN
3     16726    12.0     NaN    24.0
4     16727    30.0     8.0    10.0

   Order Id  Apples  Orange  Banana
0     False   False    True   False
1     False   False   False   False
2     False   False    True    True
3     False   False    True   False
4     False   False   False   False
```

Make a note

- ✓ isnull() and isna() both methods works in same way

4. notnull() method – Checking NaN values

- ✓ notnull() is a predefined method in DataFrame
- ✓ We can access this method by using DataFrame object.
- ✓ By using this method we can check missing values exist in DataFrame or not.
- ✓ If missing values are available then it return as False, otherwise True

Program Name: notnull() method
Input file: demo4.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.notnull()

print(df1.head())
print()
print(df2.head())
```

Output

```
   Order Id  Apples  Orange  Banana
0     16723     9.0     NaN     6.0
1     16724    12.0    12.0     6.0
2     16725     9.0     NaN     NaN
3     16726    12.0     NaN    24.0
4     16727    30.0     8.0    10.0

   Order Id  Apples  Orange  Banana
0      True    True   False    True
1      True    True    True    True
2      True    True   False   False
3      True    True   False    True
4      True    True    True    True
```

5. Counting NaN values in column wise

- ✓ We can count number of missing values in DataFrame
- ✓ By using isna() and sum() methods we can count the number of missing values in each column.

Program Name: Counting the missing values in each column
demo5.py
Input file: fruits1.csv

```
import pandas as pd

df1 = pd.read_csv('fruits1.csv')
s = df1.isna().sum()

print(s)
```

Output

```
Order Id     0
Apples      16
Orange      16
Banana      21
dtype: int64
```

Program Name: Counting the missing values in each column with percentage
demo6.py
Input file: fruits1.csv

```
import pandas as pd

df1 = pd.read_csv('fruits1.csv')
s = df1.isna().sum()
per = (s * 100) / len(df1)

print(per)
```

Output

```
Order Id      0.000000
Apples       10.666667
Orange       10.666667
Banana       14.000000
dtype: float64
```

6. dropna() method – Handling missing values

- ✓ dropna() is a predefined method in DataFrame
- ✓ We can access dropna() method by using DataFrame object.
- ✓ This method drops the rows where at least one value is missing.

Program Name: Dropping rows where NaN values existing
Input file: demo7.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.dropna()

print(df2)
```

Output

```
     Order Id  Apples  Orange  Banana
1       16724    12.0    12.0     6.0
4       16727    30.0     8.0    10.0
5       16728     6.0     6.0    12.0
6       16729    12.0    24.0     9.0
7       16730    24.0     6.0     8.0
..        ...     ...     ...     ...
143     16866    12.0    10.0    10.0
145     16868    24.0    12.0     6.0
146     16869     8.0     8.0    12.0
147     16870     5.0    30.0    30.0
149     16872     5.0    30.0    10.0

[105 rows x 4 columns]
```

Program Name: Dropping rows where NaN values existing and counting
Input file: demo8.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.dropna()
s = df2.isna().sum()

print(s)
```

Output

```
Order Id    0
Apples      0
Orange      0
Banana      0
dtype: int64
```

Program Name: Converting float column type into int data type
Input file: demo9.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv('fruits1.csv')
df2 = df1.dropna()
df3 = df2.astype(int)

print(df2.head())
print()
print(df3.head())
```

Output

```
   Order Id  Apples  Orange  Banana
1     16724    12.0    12.0     6.0
4     16727    30.0     8.0    10.0
5     16728     6.0     6.0    12.0
6     16729    12.0    24.0     9.0
7     16730    24.0     6.0     8.0

   Order Id  Apples  Orange  Banana
1     16724      12      12       6
4     16727      30       8      10
5     16728       6       6      12
6     16729      12      24       9
7     16730      24       6       8
```

7. dropna(inplace = True) method – Handling missing values

- ✓ dropna(inplace = True) is a predefined method in DataFrame
- ✓ We can access this method by using DataFrame object.
- ✓ This method drops the rows and perform changes on existing DataFrame.

Program Name: Dropping NaN values by using inplace parameter
Input file: demo10.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df1.dropna(inplace = True)

print(df1)
```

Output

```
     Order Id  Apples  Orange  Banana
1       16724    12.0    12.0     6.0
4       16727    30.0     8.0    10.0
5       16728     6.0     6.0    12.0
6       16729    12.0    24.0     9.0
7       16730    24.0     6.0     8.0
..        ...     ...     ...     ...
143     16866    12.0    10.0    10.0
145     16868    24.0    12.0     6.0
146     16869     8.0     8.0    12.0
147     16870     5.0    30.0    30.0
149     16872     5.0    30.0    10.0

[105 rows x 4 columns]
```

8. fillna() method – Handling missing values

- ✓ fillna() is a predefined method in DataFrame
- ✓ We can access this method by using DataFrame object.
- ✓ By using this method we can fill missing/NaN values with specific value.
  - fillna(0) -> This method fill NaN with Zero values
  - fillna(number) -> This method fill NaN with number

Program Name: Filling NaN values with zero
Input file: demo11.py
fruits1.csv

```
import pandas as pd

df1 = pd.read_csv("fruits1.csv")
df2 = df1.fillna(0)

print(df1.head())
print()
print(df2.head())
```

Output

```
   Order Id  Apples  Orange  Banana
0     16723     9.0     NaN     6.0
1     16724    12.0    12.0     6.0
2     16725     9.0     NaN     NaN
3     16726    12.0     NaN    24.0
4     16727    30.0     8.0    10.0

   Order Id  Apples  Orange  Banana
0     16723     9.0     0.0     6.0
1     16724    12.0    12.0     6.0
2     16725     9.0     0.0     0.0
3     16726    12.0     0.0    24.0
4     16727    30.0     8.0    10.0
```

Program Name: Filling NaN values with specific value

demo12.py

```
import pandas as pd
import numpy as np

data = [
    ["Rajan", 26, 40000],
    ["Daniel", 16, 20000],
    ["Veeru", 45, 90000],
    ["Venkat", np.nan, 45000],
    ["Sumanth", 20, 95000],
    ["Shafi", np.nan, 97000]
]

df1 = pd.DataFrame(data, columns = ['Name', 'Age', 'Salary'])
df2 = df1.fillna(22)

print(df1)
print()
print(df2)
```

Output

```
      Name   Age  Salary
0    Rajan  26.0   40000
1   Daniel  16.0   20000
2    Veeru  45.0   90000
3   Venkat   NaN   45000
4  Sumanth  20.0   95000
5    Shafi   NaN   97000

      Name   Age  Salary
0    Rajan  26.0   40000
1   Daniel  16.0   20000
2    Veeru  45.0   90000
3   Venkat  22.0   45000
4  Sumanth  20.0   95000
5    Shafi  22.0   97000
```

Program Name: Filling NaN value with mean value

demo13.py

```
import pandas as pd
import numpy as np

data = [
      ["Shahid", 26, 40000],
      ["Daniel", 16, 20000],
      ["Karteek", np.nan, 90000],
      ["Venkat", np.nan, 45000],
      ["Veeru", 24, 95000],
      ["Shafi", np.nan, 97000]
]

df1 = pd.DataFrame(data, columns = ['Name', 'Age', 'Salary'])

print(df1)
m = df1['Age'].mean()
df1['Age'] = df1['Age'].fillna(m)
print()
print(df1)
```

Output

```
      Name   Age  Salary
0   Shahid  26.0   40000
1   Daniel  16.0   20000
2  Karteek   NaN   90000
3   Venkat   NaN   45000
4    Veeru  24.0   95000
5    Shafi   NaN   97000

      Name   Age  Salary
0   Shahid  26.0   40000
1   Daniel  16.0   20000
2  Karteek  22.0   90000
3   Venkat  22.0   45000
4    Veeru  24.0   95000
5    Shafi  22.0   97000
```

Program Name: Creating dataframe and replacing nan values with specific value
demo14.py

```
import pandas as pd
import numpy as np

data = [
      ['Shahid', np.nan, 40000],
      ['Daniel', 16, 20000],
      ['Veeru', 45, 90000],
      ['Sumanth', 20, 95000]
]

df1 = pd.DataFrame(data, columns = ['Name', 'Age', 'Salary'])

print(df1)

df2 = df1.replace(np.nan, 0)

print()
print(df2)
```

Output

```
      Name   Age  Salary
0   Shahid   NaN   40000
1   Daniel  16.0   20000
2    Veeru  45.0   90000
3  Sumanth  20.0   95000

      Name   Age  Salary
0   Shahid   0.0   40000
1   Daniel  16.0   20000
2    Veeru  45.0   90000
3  Sumanth  20.0   95000
```