14. PANDAS – DataFrame – Filtering

Contents

14. PANDAS – DataFrame - Filtering

1. Filtering

- ✓ Filtering the data in DataFrame is very common requirement.
- ✓ It is very important step in Data Analysis project
- ✓ Based on condition we can filter the data from DataFrame

1.1. Filtering Examples

- ✓ Banking
  - Select all the active customers whose accounts were opened after 1st January 2020
  - Get the details of all the customers who made more than 300 transactions in the last 6 months

- ✓ Organization
  - Fetch information of employees who spent more than 3 years in the organization and received highest rating in the past 2 years

- ✓ Telecom
  - Analyse complaints data and identify customers who filed more than 5 complaints in the last 1 year

- ✓ General
  - Extract details of metro cities where per capita income is greater than 40K dollars

- ✓ Many more...

1.2. We can filter data by using

- ✓ By using relational operators.
  - Single condition
  - Multiple condition

- ✓ By using loc & iloc indexers

1.3. Creating a DataFrame

- ✓ We can create DataFrame by loading csv file.

Program Name: Loading csv file
Input file: demo1.py
sales4.csv

```
import pandas as pd

df = pd.read_csv("sales4.csv")

print(df)
```

Output

```
        Order_Id Customer_Name  Customer_Id               Product_Name  Product_Cost
0           1023         Venki           15           27in FHD Monitor         59000
1           1024     Chaithanya          14                  iPhone 11         69000
2           1025        Shahid           20  Bose SoundSport Headphones         65999
3           1026         Veeru            3       Apple iPad 10.2-inch         63999
4           1027          Venu           23               Google Phone         63999
...          ...           ...          ...                        ...           ...
299995    301018       Karteek            4       Apple iPad 10.2-inch         51999
299996    301019         Veeru            3         Macbook Pro Laptop         51999
299997    301020        Harsha            5         LG Washing Machine         65000
299998    301021     Nireekshan           1                  LG Mobile         60000
299999    301022        Pradhan          17      34in Ultrawide Monitor         55000

[300000 rows x 5 columns]
```

2. By using relational operators

✓ By using relational operators we can apply the filter on dataframe.

Program Name: Filtering DataFrame by using relational operator: Single condition
Input file: demo2.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1['Product_Cost']  > 65000
df2 = df1[con1]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id               Product_Name  Product_Cost
1           1024    Chaithanya           14                  iPhone 11         69000
2           1025        Shahid           20  Bose SoundSport Headphones         65999
11          1034         Tarun           11          Samsung Galaxy S20         69999
13          1036    Chaithanya           14      LG ThinQ Refrigerator          69999
17          1040       Sumanth           22                   iPhone 8          65999
...          ...           ...          ...                        ...           ...
299982    301005         Tarun           11          Samsung Galaxy S20         69000
299983    301006         Shafi           25      27in 4K Gaming Monitor         69000
299985    301008         Venki           15    Apple Airpods Headphones         65999
299988    301011         Venki           15                Google Phone         65999
299990    301013        Balaji           12                  iPhone 11          65999

[112280 rows x 5 columns]
```

Program Name: Filtering DataFrame by using relational operator: Single condition
Input file: demo3.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1['Product_Cost']  > 70000
df2 = df1[con1]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id           Product_Name  Product_Cost
25          1048        Partha            8  Samsung Galaxy S9 Plus         75000
28          1051       Lavanya           16      LG Washing Machine         75000
42          1065     Madhurima            7             20in Monitor         75999
45          1068         Vinay           10  Samsung Galaxy S9 Plus         75999
53          1076         Tarun           11      LG Washing Machine         75999
...          ...           ...          ...                     ...           ...
299955    300978        Siddhu           18  34in Ultrawide Monitor         75000
299963    300986        Shahid           20  34in Ultrawide Monitor         75999
299964    300987        Harsha            5      Macbook Pro Laptop         75000
299965    300988     Madhurima            7           Flatscreen TV         75000
299973    300996         Vinay           10  Samsung Galaxy S9 Plus         75999

[37401 rows x 5 columns]
```

Program

Filtering DataFrame by using relational operator: Multiple conditions

Name demo4.py
Input file sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1['Product_Cost']  > 50000
con2 = df1['Product_Cost'] < 60000

df2 = df1[con1 & con2]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id              Product_Name  Product_Cost
0           1023         Venki           15          27in FHD Monitor         59000
8           1031         Kedar            2              20in Monitor         55999
12          1035         Kedar            2              Google Phone         55999
14          1037     Nireekshan            1  Apple Airpods Headphones         55999
22          1045       Lavanya           16                 iPhone 7s         51999
...          ...           ...          ...                       ...           ...
299993    301016        Siddhu           18                 iPhone 9         59000
299994    301017         Kedar            2           ThinkPad Laptop         55999
299995    301018       Karteek            4      Apple iPad 10.2-inch         51999
299996    301019         Veeru            3        Macbook Pro Laptop         51999
299999    301022        Pradhan           17    34in Ultrawide Monitor         55000

[74794 rows x 5 columns]
```

Program

Filtering DataFrame by using relational operator: Multiple conditions

Name demo5.py
Input file sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1.Product_Name == "iPhone 11"
con2 = df1.Customer_Name == "Nireekshan"

df2 = df1[con1 & con2]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id Product_Name  Product_Cost
821         1844    Nireekshan            1    iPhone 11         63999
1086        2109    Nireekshan            1    iPhone 11         65000
1529        2552    Nireekshan            1    iPhone 11         55000
1539        2562    Nireekshan            1    iPhone 11         61000
1676        2699    Nireekshan            1    iPhone 11         65000
...          ...           ...          ...          ...           ...
296768    297791    Nireekshan            1    iPhone 11         55000
297335    298358    Nireekshan            1    iPhone 11         63999
297717    298740    Nireekshan            1    iPhone 11         50000
297766    298789    Nireekshan            1    iPhone 11         55999
298524    299547    Nireekshan            1    iPhone 11         65999

[581 rows x 5 columns]
```

Program

Filtering DataFrame by using relational operator: Multiple conditions

Name demo6.py
Input file sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1.Product_Name == "iPhone 11"
con2 = df1.Customer_Name == "Shahid"

df2 = df1[con1 & con2]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id Product_Name  Product_Cost
26          1049        Shahid           20    iPhone 11         65999
783         1806        Shahid           20    iPhone 11         69000
1260        2283        Shahid           20    iPhone 11         69000
1834        2857        Shahid           20    iPhone 11         65999
1848        2871        Shahid           20    iPhone 11         69999
...          ...           ...          ...          ...           ...
298667    299690        Shahid           20    iPhone 11         65999
298969    299992        Shahid           20    iPhone 11         69000
299206    300229        Shahid           20    iPhone 11         65000
299691    300714        Shahid           20    iPhone 11         63999
299950    300973        Shahid           20    iPhone 11         75000

[594 rows x 5 columns]
```

3. Filtering by using loc and iloc

✓ We can filter the dataframe by using loc and iloc indexers as well

Program Name: Filtering DataFrame by using loc indexer
Input file: demo7.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

con1 = df1.Product_Name == "iPhone 11"
con2 = df1.Customer_Name == "Shahid"

df2 = df1.loc[con1 & con2]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id Product_Name  Product_Cost
26          1049        Shahid           20    iPhone 11         65999
783         1806        Shahid           20    iPhone 11         69000
1260        2283        Shahid           20    iPhone 11         69000
1834        2857        Shahid           20    iPhone 11         65999
1848        2871        Shahid           20    iPhone 11         69999
...          ...           ...          ...          ...           ...
298667    299690        Shahid           20    iPhone 11         65999
298969    299992        Shahid           20    iPhone 11         69000
299206    300229        Shahid           20    iPhone 11         65000
299691    300714        Shahid           20    iPhone 11         63999
299950    300973        Shahid           20    iPhone 11         75000

[594 rows x 5 columns]
```

Program Name: Filtering DataFrame by using iloc indexer
Input file: demo8.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

df2 = df1.iloc[:5, ]

print(df2)
```

Output

	Order_Id	Customer_Name	Customer_Id	Product_Name	Product_Cost
0	1023	Venki	15	27in FHD Monitor	59000
1	1024	Chaithanya	14	iPhone 11	69000
2	1025	Shahid	20	Bose SoundSport Headphones	65999
3	1026	Veeru	3	Apple iPad 10.2-inch	63999
4	1027	Venu	23	Google Phone	63999

3.1. Rows position and column name

- ✓ We can even select the dataframe by providing rows position and column name

Program Name: Filtering DataFrame by using loc indexer
Input file: demo9.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

rows = df1.index[0:]
cols = ["Product_Name", "Customer_Id"]

df2 = df1.loc[rows, cols]

print(df2)
```

Output

```
                     Product_Name  Customer_Id
0                 27in FHD Monitor           15
1                        iPhone 11           14
2       Bose SoundSport Headphones           20
3             Apple iPad 10.2-inch            3
4                     Google Phone           23
...                            ...          ...
299995        Apple iPad 10.2-inch            4
299996          Macbook Pro Laptop            3
299997          LG Washing Machine            5
299998                   LG Mobile            1
299999      34in Ultrawide Monitor           17

[300000 rows x 2 columns]
```

Program Name: Filtering DataFrame by using loc indexer
Input file: demo10.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

rows = df1.index[0:4]
cols = ["Product_Name", "Customer_Id"]

df2 = df1.loc[rows, cols]

print(df2)
```

Output

```
                 Product_Name  Customer_Id
0             27in FHD Monitor           15
1                    iPhone 11           14
2  Bose SoundSport Headphones           20
3         Apple iPad 10.2-inch            3
```

Program Name: Filtering DataFrame by using loc indexer
Input file: demo11.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

rows = df1.index[5:]
cols = ["Product_Name", "Customer_Id"]

df2 = df1.loc[rows, cols]

print(df2)
```

Output

```
                  Product_Name  Customer_Id
5       Samsung Galaxy S9 Plus            6
6                    iPhone 11           23
7               27in FHD Monitor           25
8                  20in Monitor            2
9             LG Washing Machine           19
...                          ...          ...
299995    Apple iPad 10.2-inch             4
299996      Macbook Pro Laptop             3
299997      LG Washing Machine             5
299998                  LG Mobile          1
299999  34in Ultrawide Monitor            17

[299995 rows x 2 columns]
```

3.2. Selecting multiple values of a column

✓ We can filter dataframe by providing multiple values of a column

Program Name: Filtering DataFrame by using loc indexer
Input file: demo12.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

a = df1.Product_Name == "LG Washing Machine"
b = df1.Customer_Id == 1
c = a | b

df2 = df1.loc[c]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id             Product_Name  Product_Cost
9           1032       Neelima           19        LG Washing Machine         63000
14          1037    Nireekshan            1  Apple Airpods Headphones         55999
24          1047         Shafi           25        LG Washing Machine         63999
28          1051       Lavanya           16        LG Washing Machine         75000
39          1062         Tarun           11        LG Washing Machine         61000
...          ...           ...          ...                       ...           ...
299936    300959       Partha            8        LG Washing Machine         65999
299937    300960    Nireekshan            1          27in FHD Monitor         60000
299970    300993    Nireekshan            1                  iPhone 8         60000
299997    301020       Harsha            5        LG Washing Machine         65000
299998    301021    Nireekshan            1                 LG Mobile         60000

[26278 rows x 5 columns]
```

3.3. isin() method

- ✓ isin() is predefined method in Series class.
- ✓ We should access this method by using Series object.
- ✓ By using this method we can select data from DataFrame

Program Name: Filtering DataFrame by using loc isin() method
Input file: demo13.py
Input file: sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

a = ["Macbook Pro Laptop"]

b = df1.Product_Name.isin(a)

df2 = df1[b]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id        Product_Name  Product_Cost
23          1046     Madhurima            7  Macbook Pro Laptop         50000
49          1072        Balaji           12  Macbook Pro Laptop         61000
56          1079         Tarun           11  Macbook Pro Laptop         75000
60          1083         Tarun           11  Macbook Pro Laptop         65999
85          1108         Vijay            9  Macbook Pro Laptop         59000
...          ...           ...          ...                 ...           ...
299906    300929        Partha            8  Macbook Pro Laptop         51999
299907    300930       Neelima           19  Macbook Pro Laptop         63000
299964    300987        Harsha            5  Macbook Pro Laptop         75000
299974    300997     Chaithanya           14  Macbook Pro Laptop         65999
299996    301019         Veeru            3  Macbook Pro Laptop         51999

[15144 rows x 5 columns]
```

Program Name: Filtering DataFrame by using loc isin() method
Input file: demo14.py
sales4.csv

```
import pandas as pd

df1 = pd.read_csv("sales4.csv")

a = ["34in Ultrawide Monitor", "Macbook Pro Laptop"]

b = df1.Product_Name.isin(a)

df2 = df1[b]

print(df2)
```

Output

```
        Order_Id Customer_Name  Customer_Id           Product_Name  Product_Cost
23          1046     Madhurima            7     Macbook Pro Laptop         50000
49          1072        Balaji           12     Macbook Pro Laptop         61000
56          1079         Tarun           11     Macbook Pro Laptop         75000
60          1083         Tarun           11     Macbook Pro Laptop         65999
71          1094        Shahid           20  34in Ultrawide Monitor         61000
...          ...           ...          ...                    ...           ...
299964    300987        Harsha            5     Macbook Pro Laptop         75000
299969    300992         Tarun           11  34in Ultrawide Monitor         65999
299974    300997    Chaithanya           14     Macbook Pro Laptop         65999
299996    301019         Veeru            3     Macbook Pro Laptop         51999
299999    301022       Pradhan           17  34in Ultrawide Monitor         55000

[30299 rows x 5 columns]
```

3.4. unique() function

- ✓ unique() is predefined function in pandas.
- ✓ We should access this function by using pandas module.
- ✓ This function returns the unique values from the column.

Program Name: Selecting unique column values
Input file: demo15.py
sales4.csv

```
import pandas as pd

df = pd.read_csv("sales4.csv")

a = pd.unique(df.Product_Name)

print(a)
print(len(a))
```

Output

```
['27in FHD Monitor' 'iPhone 11' 'Bose SoundSport Headphones'
 'Apple iPad 10.2-inch' 'Google Phone' 'Samsung Galaxy S9 Plus'
 '20in Monitor' 'LG Washing Machine' 'iPhone 7s' 'Samsung Galaxy S20'
 'LG ThinQ Refrigerator ' 'Apple Airpods Headphones' 'iPhone 8'
 'Macbook Pro Laptop' 'LG Mobile' 'ThinkPad Laptop' 'Flatscreen TV'
 '34in Ultrawide Monitor' 'iPhone 9' '27in 4K Gaming Monitor']
20
```

Program Name: Selecting unique column values
Input file: demo16.py
sales4.csv

```
import pandas as pd

df = pd.read_csv("sales4.csv")

a = pd.unique(df.Customer_Name)

print(a)
print(len(a))
```

Output

```
['Venki' 'Chaithanya' 'Shahid' 'Veeru' 'Venu' 'Daniel' 'Shafi' 'Kedar'
 'Neelima' 'Vijay' 'Tarun' 'Nireekshan' 'Karteek' 'Sumanth' 'Mallikarjun'
 'Vinay' 'Lavanya' 'Madhurima' 'Partha' 'Siddhu' 'Jaya Chandra' 'Balaji'
 'Pradhan' 'Harsha' 'Mohan']
25
```

4. Select Non-Missing Data from DataFrame

notnull() method

- ✓ notnull() is predefined method in Series class.
- ✓ We should access this method by using Series object
- ✓ By using this function we can select the DataFrame which having non NaN values

Program Name: Creating a DataFrame
demo17.py

```
import pandas as pd
import numpy as np

data = [
    ['Shahid', 21, 40000],
    ['Nireekshan', 22, 20000],
    ['Veeru', 45, 90000],
    ['Sumanth', 20, 95000],
    [np.nan, 2, 99000],
    ['Prasad', 1, 41000]
]

c = ['Name', 'Age', 'Salary']

df1 = pd.DataFrame(data, columns = c)

print(df1)
```

Output

```
         Name  Age  Salary
0      Shahid   21   40000
1  Nireekshan   22   20000
2       Veeru   45   90000
3     Sumanth   20   95000
4         NaN    2   99000
5      Prasad    1   41000
```

Program Name: notnull() method
demo18.py

```
import pandas as pd
import numpy as np

data = [
      ['Shahid', 21, 40000],
      ['Nireekshan', 22, 20000],
      ['Veeru', 45, 90000],
      ['Sumanth', 20, 95000],
      [np.nan, 2, 99000],
      ['Prasad', 1, 41000]
]

c = ['Name', 'Age', 'Salary']

df1 = pd.DataFrame(data, columns = c)

d = df1.Name.notnull()

df2 = df1[d]

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

Output

```
         Name  Age  Salary
0      Shahid   21   40000
1  Nireekshan   22   20000
2       Veeru   45   90000
3     Sumanth   20   95000
4         NaN    2   99000
5      Prasad    1   41000

         Name  Age  Salary
0      Shahid   21   40000
1  Nireekshan   22   20000
2       Veeru   45   90000
3     Sumanth   20   95000
5      Prasad    1   41000
```