Pandas dataframe filter with Multiple conditions

Posted on Jan 21, 2020 ·

Selecting or filtering rows from a dataframe can be sometime tedious if you don't know the exact methods and how to filter rows with multiple conditions

In this post we are going to see the different ways to select rows from a dataframe using multiple conditions

Let's create a dataframe with 5 rows and 4 columns i.e. Name, Age, Salary_in_1000 and FT_Team(Football Team)

```
import pandas as pd
df=pd.DataFrame({'Name':['JOHN','ALLEN','BOB','NIKI','CHARLIE','CHANG'],
                 'Age':[35,42,63,29,47,51],
                 'Salary_in_1000':[100,93,78,120,64,115],
                'FT_Team':['STEELERS','SEAHAWKS','FALCONS','FALCONS','PATRIOTS','STEELERS']})
df
```

Output:

-	Name	Age	Salary_in_1000	FT_Team
0	JOHN	35	100	STEELERS
1	ALLEN	42	93	SEAHAWKS
2	BOB	63	78	FALCONS
3	NIKI	29	120	FALCONS
4	CHARLIE	47	64	PATRIOTS
5	CHANG	51	115	STEELERS

Selecting Dataframe rows on multiple conditions using these 5 functions

In this section we are going to see how to filter the rows of a dataframe with multiple conditions using these five methods

a) loc b) numpy where c) Query d) Boolean Indexing e) eval

What's the Condition or Filter Criteria ?

Get all rows having salary greater or equal to 100K and Age < 60 and Favourite Football Team Name starts with 'S'

Using loc with multiple conditions

loc is used to Access a group of rows and columns by label(s) or a boolean array

As an input to label you can give a single label or it's index or a list of array of labels

Enter all the conditions and with & as a logical operator between them

```
df.loc[(df['Salary_in_1000']>=100) & (df['Age']< 60) &
(df['FT_Team'].str.startswith('S')),['Name','FT_Team']]
```

Output:

	Name	FT_Team
0	JOHN	STEELERS
5	CHANG	STEELERS

Using np.where with multiple conditions

numpy where can be used to filter the array or get the index or elements in the array where conditions are met. You can read more about np.where in this post

Numpy where with multiple conditions and & as logical operators outputs the index of the matching rows

```
import numpy as np
idx = np.where((df['Salary_in_1000']>=100) & (df['Age']< 60) &
(df['FT_Team'].str.startswith('S')))
```

Output:

```
(array([0, 5], dtype=int64),)
```

The output from the np.where, which is a list of row index matching the multiple conditions is fed to dataframe loc function

```
df.loc[idx]
```

Output:

	Name	Age	Salary_in_1000	FT_Team
0	JOHN	35	100	STEELERS
5	CHANG	51	115	STEELERS

Using Query with multiple Conditions

It is used to Query the columns of a DataFrame with a boolean expression

```
df.query('Salary_in_1000 >= 100 & Age < 60 & FT_Team.str.startswith("S").values')
```

Output:

	Name	Age	Salary_in_1000
0	JOHN	35	100
5	CHANG	51	115

pandas boolean indexing multiple conditions

It is a standrad way to select the subset of data using the values in the dataframe and applying conditions on it

We are using the same multiple conditions here also to filter the rows from pur original dataframe with salary >= 100 and Football team starts with alphabet 'S' and Age is less than 60

```
df[(df['Salary_in_1000']>=100) & (df['Age']<60) &
df['FT_Team'].str.startswith('S')][['Name','Age','Salary_in_1000']]
```

Output:

	Name	Age	Salary_in_1000
0	JOHN	35	100
5	CHANG	51	115

Pandas Eval multiple conditions

Evaluate a string describing operations on DataFrame column. It Operates on columns only, not specific rows or elements

```
df[df.eval("Salary_in_1000>=100 & (Age <60) & FT_Team.str.startswith('S').values")]
```

Output:

	Name	Age	Salary_in_1000
0	JOHN	35	100
5	CHANG	51	115

Conclusion:

In this post we have seen that what are the different methods which are available in the Pandas library to filter the rows and get a subset of the dataframe

And how these functions works: loc works with column labels and indexes, whereas eval and query works only with columns and boolean indexing works with values in a column only

Source: https://kanoki.org/2020/01/21/pandas-dataframe-filter-with-multiple-conditions/