How can I normalize columns in a Pandas DataFrame?

Authored by

stats writer

April 18, 2024

Normalizing columns in a Pandas DataFrame refers to the process of transforming the values in each column to a common scale, usually between 0 and 1. This allows for easier comparison and analysis of the data. In order to normalize columns in a Pandas DataFrame, you can use the "normalize" method and specify the axis (usually 0 for column-wise normalization) and the desired normalization method (such as "min-max" or "z-score"). This will apply the normalization transformation to each column in the DataFrame, ensuring that the data is on a consistent scale for accurate analysis.

Normalize Columns in a Pandas DataFrame

Often you may want to normalize the data values of one or more columns in a pandas DataFrame.

This tutorial explains two ways to do so:

1. Min-Max Normalization

Objective: Converts each data value to a value between 0 and 1.Formula: New value = (value - min) / (max - min)

2. Mean Normalization

Objective: Scales values such that the mean of all values is 0 and std. dev. is 1. Formula: New value = (value - mean) / (standard deviation)

Let's check out an example of how to use each method on a pandas DataFrame.

Example 1: Min-Max Normalization

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'points': ,
'assists': ,
'rebounds': })

#view DataFrame
print(df)
```

```
points assists rebounds
0 25 5 11
1 12 7 8
2 15 7 10
3 14 9 6
4 19 12 6
```


We can use the following code to apply a min-max normalization to each column in the DataFrame:

```
(df-df.min())/(df.max()-df.min())
```

points assists rebounds
0 1.000000 0.000000 1.0
1 0.000000 0.285714 0.4
2 0.230769 0.285714 0.8
3 0.153846 0.571429 0.0
4 0.538462 1.000000 0.0

The max value in each column is now equal to 1 and the min value in each column is now equal to 0, with all other values ranging between 0 and 1.

Example 2: Mean Normalization

Once again suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'points': ,
'assists': ,
'rebounds': })

#view DataFrame
print(df)
```

points assists rebounds
0 25 5 11
1 12 7 8
2 15 7 10
3 14 9 6
4 19 12 6

We can use the following code to apply a mean normalization to each column in the DataFrame:

(df-df.mean())/df.std()

points assists rebounds
0 1.554057 -1.133893 1.227881
1 -0.971286 -0.377964 -0.087706
2 -0.388514 -0.377964 0.789352
3 -0.582772 0.377964 -0.964764
4 0.388514 1.511858 -0.964764

The values in each column are now normalized such that the mean of the values in each column is 0 and the standard deviation of values in each column is 1.

If a particular data point has a normalized value greater than 0, it's an indication that the data point is greater

than the mean of its column. Conversely, a normalized value less than 0 is an indication that the data point is less than the mean of its column.

Pandas: How to Group and Aggregate by Multiple Columns

How to Filter a Pandas DataFrame on Multiple Conditions

How to Count Missing Values in a Pandas DataFrame