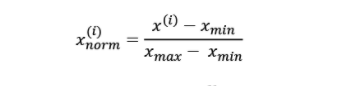
In machine learning, MinMaxscaler and StandardScaler are two scaling algorithms for continuous variables. The MinMaxscaler is a type of scaler that scales the minimum and maximum values to be 0 and 1 respectively. While the StandardScaler scales all values between min and max so that they fall within a range from min to max. In this you will learn about concepts and differences between MinMaxScaler & StandardScaler with the help of Python code examples. Note that these are classes provided by sklearn.preprocessing module and used for feature scaling purposes.

What is Normalization?

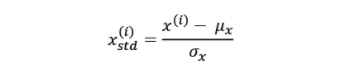
**Normalization** refers to the rescaling of the features to a range of [0, 1], which is a special case of **min-max scaling**. To normalize the data, the min-max scaling can be applied to one or more feature columns. Here is the formula for normalizing data based on min-max scaling. **Normalization is useful when the data is needed in the bounded intervals.**



**Normalizing data based on min-max scaling concepts**

What is Standardization?

**The standardization** technique is used to center the feature columns at mean 0 with a standard deviation of 1 so that the feature columns have the same parameters as a standard normal distribution. Unlike **Normalization**, standardization **maintains useful information about outliers** and makes the algorithm less sensitive to them in contrast to **min-max scalin**g, which scales the data to a limited range of values. Here is the formula for standardization.

 **Standardization formula**

* **Normalization** is about **transforming** the feature values to fall within the **bounded intervals (min and max)**
* **Standardization** is about **transforming** the feature values to fall around **mean as 0 with standard deviation as 1**

SET B

import pandas as pd

from sklearn.preprocessing import StandardScaler

scale = StandardScaler()

df=pd.read\_csv("Data1.csv")

X = df[['Age', 'Income']]

scaler=per.MinMaxScaler(feature\_range=(0,1))

rescaleData=scaler.fit\_transform(X)

print(rescaleData)

[[0.73913043 0.68571429]

[0. 0. ]

[0.13043478 0.17142857]

[0.47826087 0.37142857]

[0.56521739 nan]

[0.34782609 0.28571429]

[ nan 0.11428571]

[0.91304348 0.88571429]

[1. 1. ]

[0.43478261 0.54285714]]

rescaleData=pd.DataFrame(rescaleData,index=df.index)

print(rescaleData)

0 1

0 0.739130 0.685714

1 0.000000 0.000000

2 0.130435 0.171429

3 0.478261 0.371429

4 0.565217 NaN

5 0.347826 0.285714

6 NaN 0.114286

7 0.913043 0.885714

8 1.000000 1.000000

9 0.434783 0.542857

Example 2

import pandas as pd

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

df=pd.read\_csv("SOCR-HeightWeight.csv")

df

| **Index** | **Height(Inches)** | **Weight(Pounds)** |
| --- | --- | --- |
| **0** | 1 | 65.78331 | 112.9925 |
| **1** | 2 | 71.51521 | 136.4873 |
| **2** | 3 | 69.39874 | 153.0269 |
| **3** | 4 | 68.21660 | 142.3354 |
| **4** | 5 | 67.78781 | 144.2971 |
| **...** | ... | ... | ... |
| **24995** | 24996 | 69.50215 | 118.0312 |
| **24996** | 24997 | 64.54826 | 120.1932 |
| **24997** | 24998 | 64.69855 | 118.2655 |
| **24998** | 24999 | 67.52918 | 132.2682 |
| **24999** | 25000 | 68.87761 | 124.8742 |

25000 rows × 3 columns

scaler**=**per.MinMaxScaler(feature\_range**=**(0,1))

rescaleData**=**scaler.fit\_transform(df)

rescaleData**=**pd.DataFrame(rescaleData,index**=**df.index,columns**=**df.columns)

print(rescaleData)

Index Height(Inches) Weight(Pounds)

0 0.00000 0.370095 0.376472

1 0.00004 0.755447 0.629351

2 0.00008 0.613158 0.807370

3 0.00012 0.533683 0.692295

4 0.00016 0.504856 0.713410

... ... ... ...

24995 0.99984 0.620110 0.430705

24996 0.99988 0.287063 0.453975

24997 0.99992 0.297167 0.433226

24998 0.99996 0.487468 0.583940

24999 1.00000 0.578123 0.504357

[25000 rows x 3 columns]