What is Normalization?

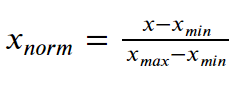
Before diving into normalization, let us first understand the need of it!!

**Feature Scaling** is an essential step in the data analysis and preparation of data for modeling. Wherein, we make the data scale-free for easy analysis.

Normalization is one of the feature scaling techniques. We particularly apply normalization when the data is **skewed**on the either axis i.e. when the data does not follow the **gaussian distribution**.

In **normalization**, we **convert the data features of different scales to a common scale** which further makes it easy for the data to be processed for modeling. Thus, all the data features(variables) tend to have a similar impact on the modeling portion.

According to the below formula, we normalize each feature by subtracting the minimum data value from the data variable and then divide it by the range of the variable as shown–

Normalization

Thus, we transform the values to a range between **[0,1]**. Let us now try to implement the concept of Normalization in Python in the upcoming section.

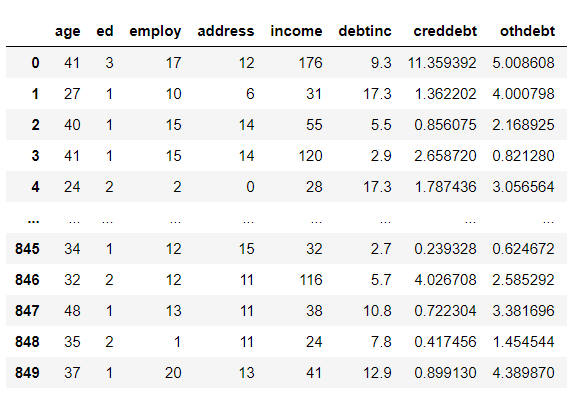
## Steps to Normalize Data in Python

There are various approaches in Python through which we can perform Normalization.

Today, we will be using one of the most popular way– **MinMaxScaler**.

Let us first have a look at the dataset which we would be scaling ahead.

**Dataset:**

**Dataset For Normalization**

Further, we will be using **min and max scaling in sklearn** to perform normalization.

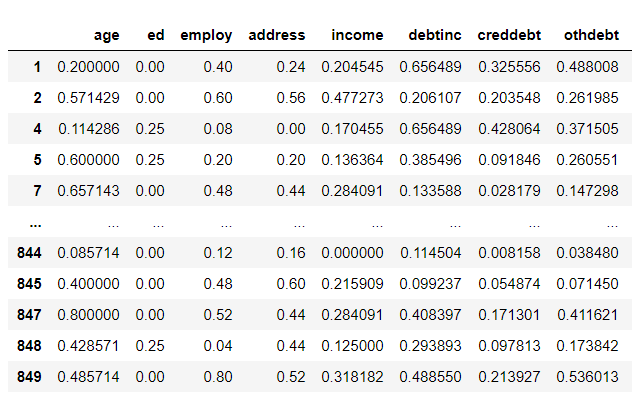
**Example:**

|  |
| --- |
| **import** pandas as pd  **import** os  **from** sklearn.preprocessing **import** MinMaxScaler  #Changing the working directory to the specified path--  os.chdir("D:/Normalize - Loan\_Defaulter")    data **=** pd.read\_csv("bank-loan.csv") # dataset  scaler **=** MinMaxScaler()    loan**=**pd.DataFrame(scaler.fit\_transform(data),              columns**=**data.columns, index**=**data.index)  print(loan) |

Here, we have created an object of MinMaxScaler() [class](https://www.askpython.com/python/oops/python-classes-objects). Further, we have used fit\_transform() method to normalize the data values.

**Output:**

So, as clearly visible, we have transformed and normalized the data values in the range of 0 and 1.

**Dataset After Normalization Using MinMaxScaler**

## Summary

Thus, from the above explanation, the following insights can be drawn–

* Normalization is used when the data values are **skewed**and **do not follow gaussian distribution**.
* The data values get converted between a **range of 0 and 1**.
* Normalization makes the data **scale free**.