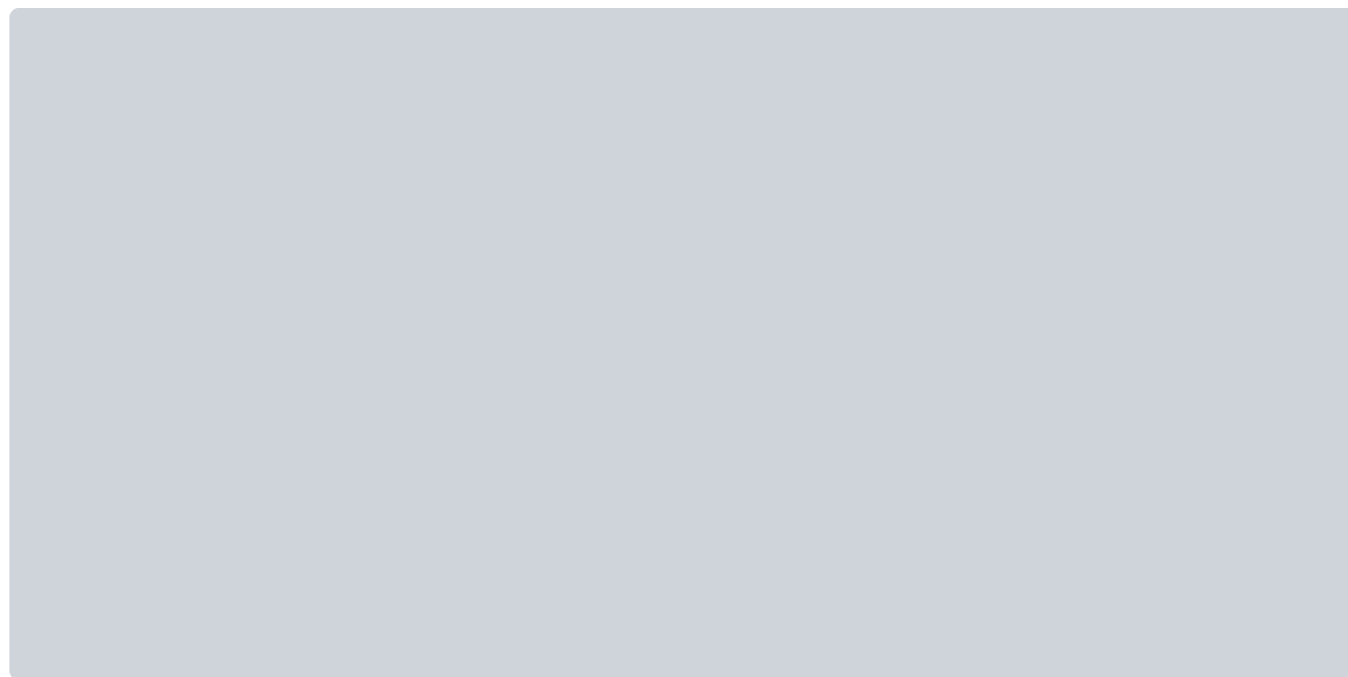


2 Easy Ways to Standardize Data in Python for Machine Learning

By Safa Mulani / October 13, 2020



Hey, readers. In this article, we will be focusing on **2 Important techniques to Standardize Data in Python**. So, let us get started!!

Why do we need to standardize data in Python?

Before diving deep into the concept of standardization[ⓧ], it is very important for us to know the need for it.

This ad will end in 3



So, you see, the datasets which we use to build a model for a particular problem statement is usually built from various sources. Thus, it can be assumed that the data set contains variables/features of different scales.

In order for our machine learning or deep learning model to work well, it is very necessary for the data to have the same scale in terms of the Feature to avoid bias in the outcome.



Thus, **Feature Scaling** is considered an important

Feature Scaling can be broadly classified into the

Normalization

Standardization

This ad will end in 3



Standardization is used on the data values that are normally distributed. Further, by applying standardization, we tend to make the mean of the dataset as 0 and the standard deviation equivalent to 1.

That is, by standardizing the values, we get the following statistics of the data distribution

mean = 0

standard deviation = 1

$$z = \frac{x - \mu}{\sigma}$$

Standardization

Thus, by this the data set becomes self explanatory
turns down to 0 and it happens to have an **unit**

This ad will end in 3



Let us now focus on the various ways of implementing Standardization in the upcoming section.

1. Using preprocessing.scale() function

The `preprocessing.scale(data)` function can be used to standardize the data values to a value having mean equivalent to zero and standard deviation as 1.

Here, we have loaded the **IRIS dataset** into the environment using the below line:

```
from sklearn.datasets import load_iris
```

Further, we have saved the iris dataset to the data object as created below.

```
from sklearn import preprocessing
data = load_iris()

# separate the independent and dependent variables
X_data = data.data
target = data.target

# standardization of dependent variables
standard = preprocessing.scale(X_data)
```



This ad will end in 3



After segregating the dependent and the response/target variable, we have applied `preprocessing.scale()` function on the dependent variables to standardize the data.

Output:

```
[ [-9.00681170e-01  1.01900435e+00 -1.34022653e+00 -1.31544430e+00]
 [ -1.14301691e+00 -1.31979479e-01 -1.34022653e+00 -1.31544430e+00]
 [ -1.38535265e+00  3.28414053e-01 -1.39706395e+00 -1.31544430e+00]
 [ -1.50652052e+00  9.82172869e-02 -1.28338910e+00 -1.31544430e+00]
 [ -1.02184904e+00  1.24920112e+00 -1.34022653e+00 -1.31544430e+00]
 [ -5.37177559e-01  1.93979142e+00 -1.16971425e+00 -1.05217993e+00]
 [ -1.50652052e+00  7.88807586e-01 -1.34022653e+00 -1.18381211e+00]
 [ -1.02184904e+00  7.88807586e-01 -1.28338910e+00 -1.31544430e+00]
 [ -1.74885626e+00 -3.62176246e-01 -1.34022653e+00 -1.31544430e+00]
 [ -1.14301691e+00  9.82172869e-02 -1.28338910e+00 -1.44707648e+00]
 [ -5.37177559e-01  1.47939788e+00 -1.28338910e+00 -1.31544430e+00]
 [ -1.26418478e+00  7.88807586e-01 -1.22655167e+00 -1.31544430e+00]
 [ -1.26418478e+00 -1.31979479e-01 -1.34022653e+00 -1.31544430e+00]
 [ -1.87002413e+00 -1.31979479e-01 -1.50652052e+00 -1.31544430e+00]
 [ -5.25060772e-02  2.16998818e+00 -1.44707648e+00 -1.31544430e+00]
 [ -1.73673948e-01  3.09077525e+00 -1.22655167e+00 -1.31544430e+00]
 [ -5.37177559e-01  1.93979142e+00 -1.34022653e+00 -1.31544430e+00]
 [ -9.00681170e-01  1.01900435e+00 -1.34022653e+00 -1.31544430e+00]
 [ -1.73673948e-01  1.70959465e+00 -1.16971425e+00 -1.31544430e+00]
 [ -9.00681170e-01  1.70959465e+00 -1.16971425e+00 -1.31544430e+00]
```

This ad will end in 3



2. Using StandardScaler() function

Python sklearn library offers us with StandardScaler() function to perform standardization on the dataset.

Here, again we have made use of Iris dataset.

Further, we have created an object of StandardScaler() and then applied fit_transform() function to apply standardization on the dataset.

```
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler

data = load_iris()
scale= StandardScaler()

# separate the independent and dependent variables
X_data = data.data
target = data.target

# standardization of dependent variables
scaled_data = scale.fit_transform(X_data)
print(scaled_data)
```

This ad will end in 3



```
[ [-9.00681170e-01  1.01900435e+00 -1.34022653e+00 -1.31544430e+00]
[ -1.14301691e+00 -1.31979479e-01 -1.34022653e+00 -1.31544430e+00]
[ -1.38535265e+00  3.28414053e-01 -1.39706395e+00 -1.31544430e+00]
[ -1.50652052e+00  9.82172869e-02 -1.28338910e+00 -1.31544430e+00]
[ -1.02184904e+00  1.24920112e+00 -1.34022653e+00 -1.31544430e+00]
[ -5.37177559e-01  1.93979142e+00 -1.16971425e+00 -1.05217993e+00]
[ -1.50652052e+00  7.88807586e-01 -1.34022653e+00 -1.18381211e+00]
[ -1.02184904e+00  7.88807586e-01 -1.28338910e+00 -1.31544430e+00]
[ -1.74885626e+00 -3.62176246e-01 -1.34022653e+00 -1.31544430e+00]
[ -1.14301691e+00  9.82172869e-02 -1.28338910e+00 -1.44707648e+00]
[ -5.37177559e-01  1.47939788e+00 -1.28338910e+00 -1.31544430e+00]
[ -1.26418478e+00  7.88807586e-01 -1.22655167e+00 -1.31544430e+00]
[ -1.26418478e+00 -1.31979479e-01 -1.34022653e+00 -1.44707648e+00]
[ -1.87002413e+00 -1.31979479e-01 -1.51073881e+00 -1.44707648e+00]
[ -5.25060772e-02  2.16998818e+00 -1.45390138e+00 -1.31544430e+00]
[ -1.73673948e-01  3.09077525e+00 -1.28338910e+00 -1.05217993e+00]
[ -5.37177559e-01  1.93979142e+00 -1.39706395e+00 -1.05217993e+00]
[ -9.00681170e-01  1.01900435e+00 -1.34022653e+00 -1.18381211e+00]
[ -1.73673948e-01  1.70959465e+00 -1.16971425e+00 -1.18381211e+00]
[ -9.00681170e-01  1.70959465e+00 -1.28338910e+00 -1.18381211e+00]
```

Standardization-Output

Conclusion

By this, we have come to the end of this topic. Feel free to comment below, in case you come across any question.



This ad will end in 3



Till then, Stay tuned and Happy Learning!! 😊

[← Previous Post](#)

[Next Post →](#)

Search ...



Recent Posts

[How to Write a Styler to a file, buffer or string in LaTeX?](#)

[Appending Dataframes in Pandas with For Loops](#)

[Vanishing Gradient Problem With Solution](#)

[Converting String to Numpy Datetime64 in a Dataframe](#)

[How to Change Datetime Format in Pandas](#)

[Activating a Virtual Environment in Windows 10 Command Prompt](#)

[How to Convert a Datetime to Date](#)

[Determine if Two Lists Have Same Elements, Regardless of Order](#)

[Adding Tuples to Lists in Python](#)

[How to Get Week Numbers in Python](#)

Kubernetes best practices.



Favorite Sites

This ad will end in 3




[Linux Tutorials](#)

[MySQL Tutorials](#)

[CodeForGeek](#)

[Mkyong](#)

Copyright © 2023 AskPython · All Rights Reserved 
[Privacy Policy](#) · [Terms and Conditions](#) · [Contact Us](#)
AskPython is part of JournalDev IT Series

This ad will end in 3

