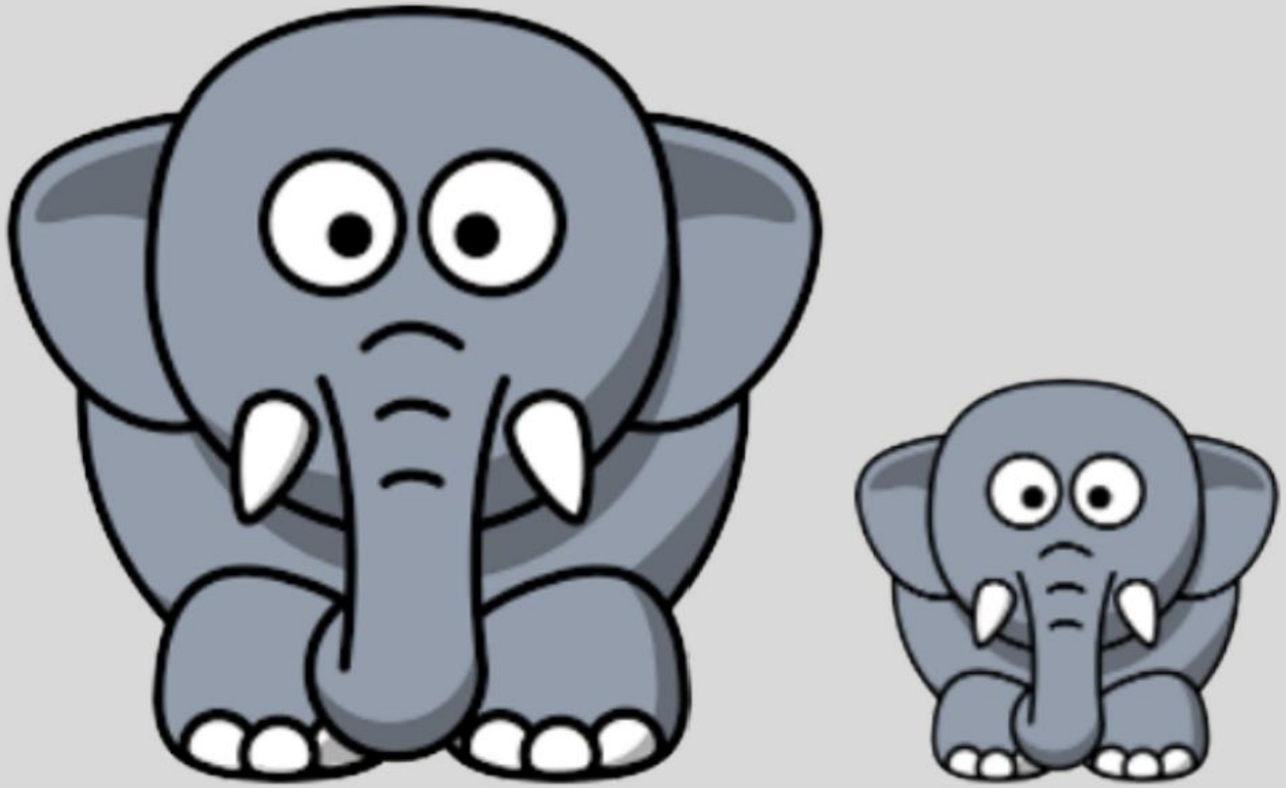
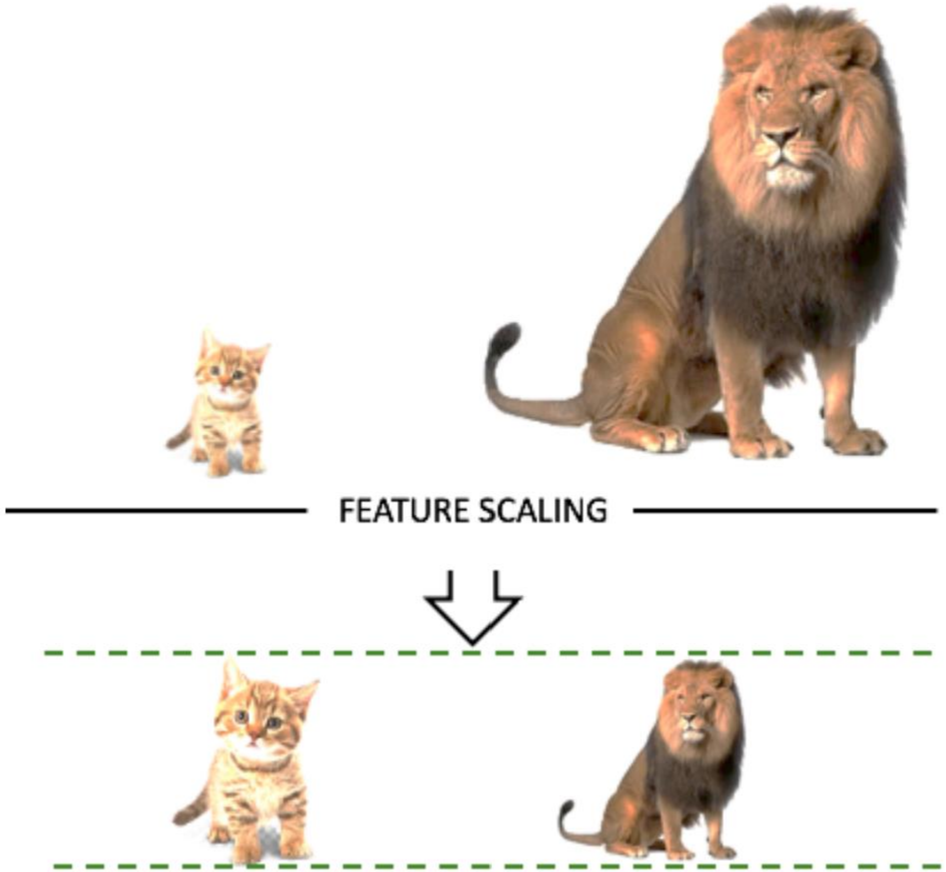
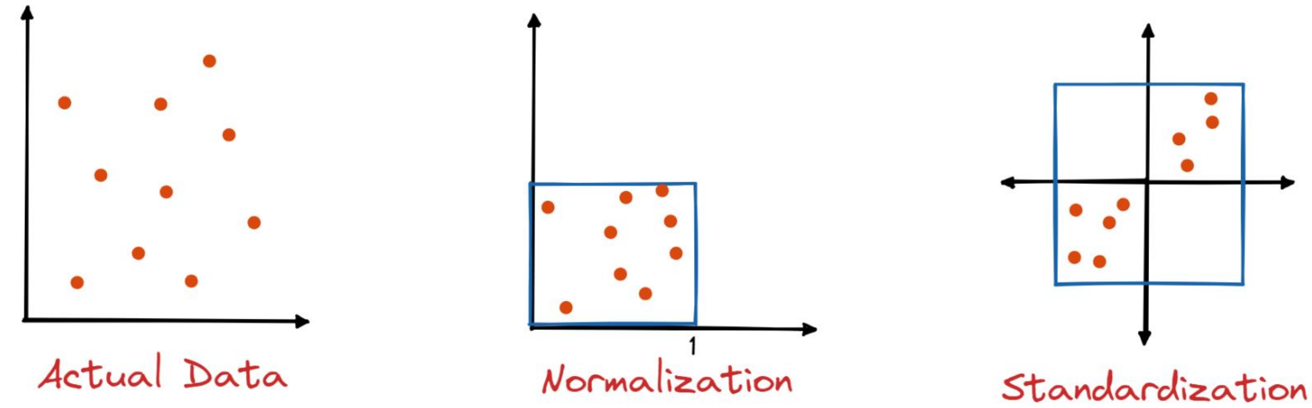
Standardization & Normalization in ML with Python Example

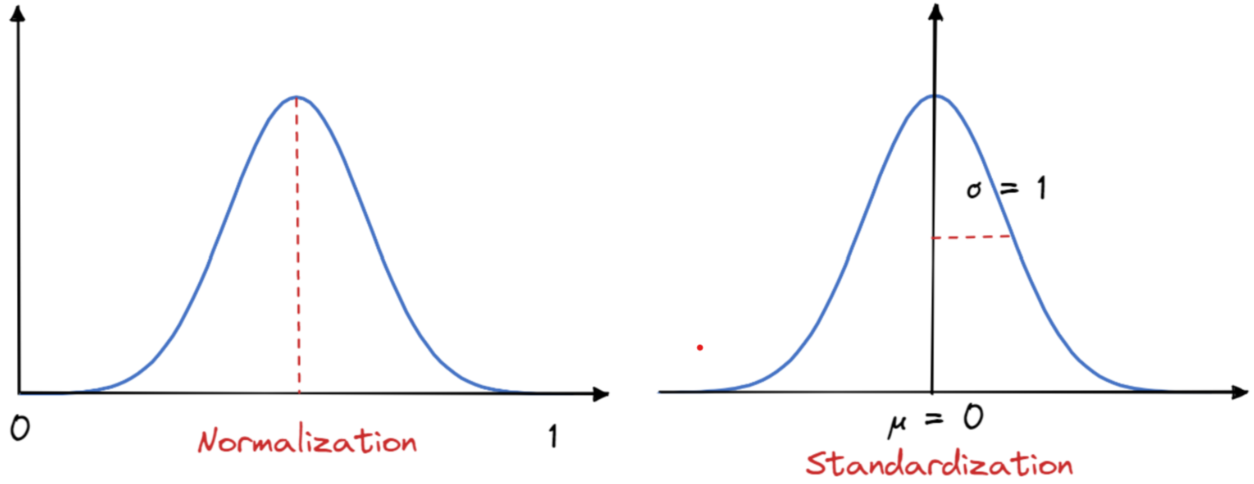


Every machine learning journey starts with data preprocessing. **Feature scaling** is one of the most important steps in preprocessing. In this carousel, we will discover 2 different feature scaling techniques.

- **Normalization**
- **Standardization**







Normalization

In this approach, features are scaled down to values between **[0,1]**.

The formula is used as follows:

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

When **x = min(x)**, function returns **0** and minimum value turns into value **0**.

When **x = max(x)**, function returns **1** and maximum values turns into value **1**.

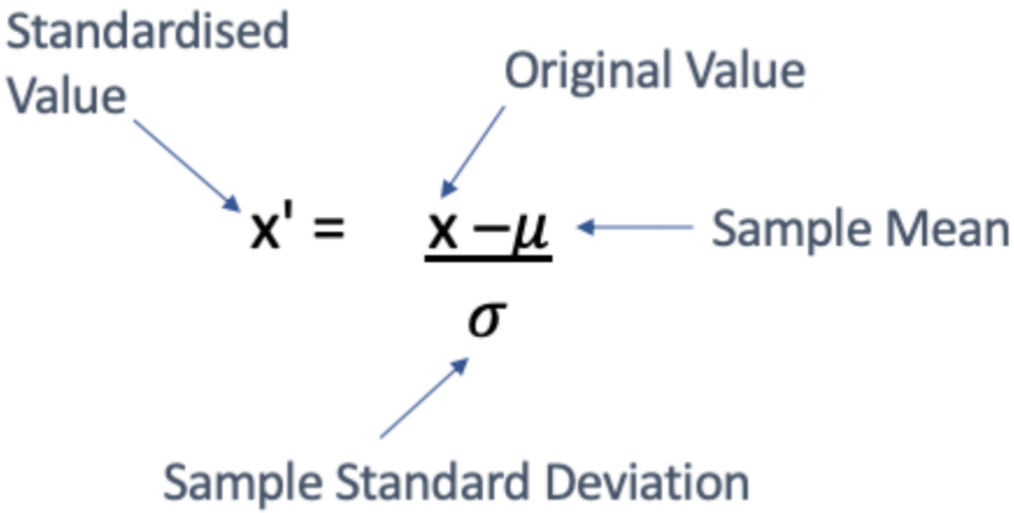
Other values are converted into values between **0** and **1** accordingly.

Standardization

Here, all values will be scaled down in such a way that they will have the characteristic of a standard normal distribution with **mean(μ) = 0**, **standard deviation(σ) = 1** (Almost).

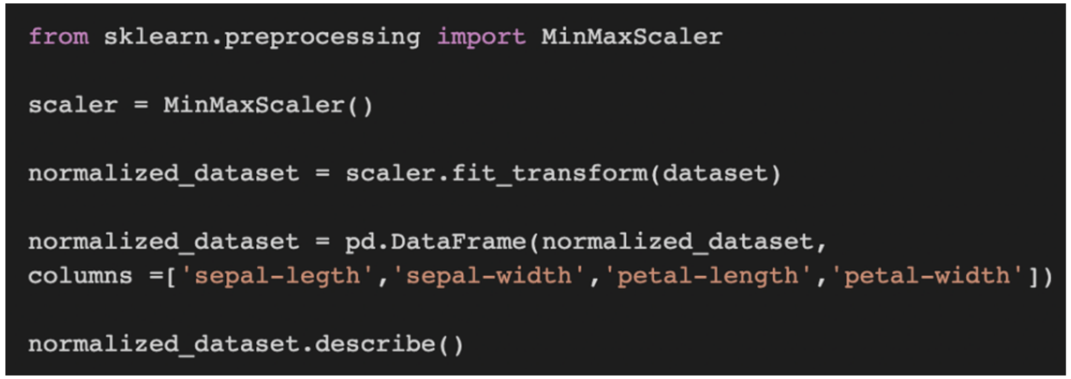
In this technique, there are no min or max boundaries; it is all about distribution.

The formula is used as follows:



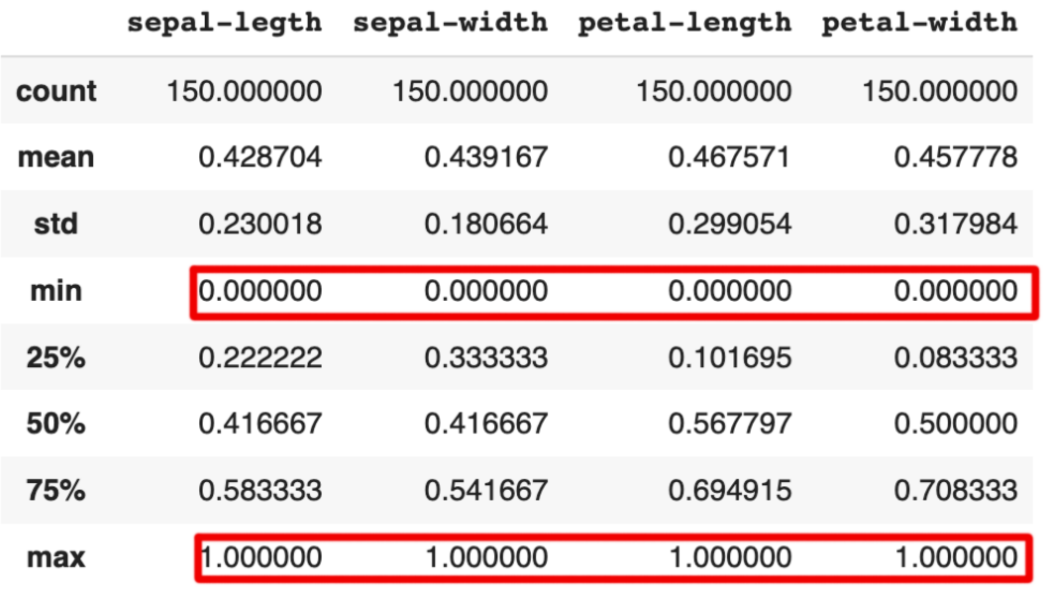
$$x' = \frac{x - \mu}{\sigma}$$

Normalization Code:



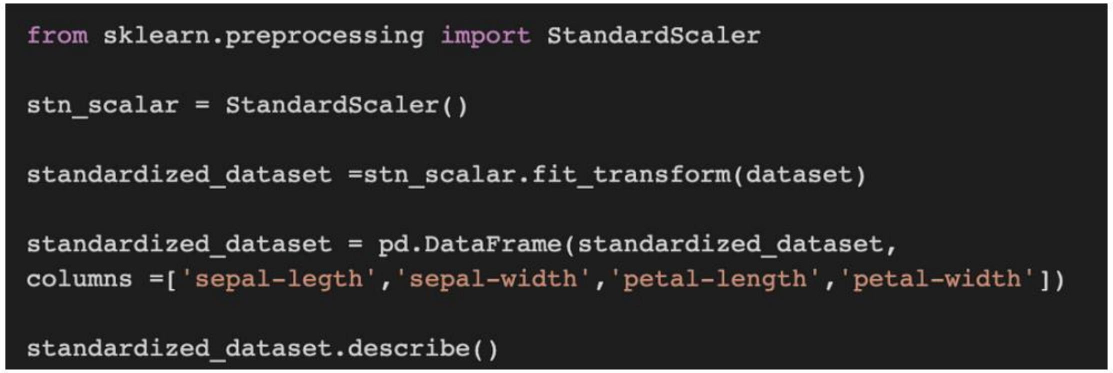
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

normalized_dataset = scaler.fit_transform(dataset)

normalized_dataset = pd.DataFrame(normalized_dataset,
columns =['sepal-legth','sepal-width','petal-length','petal-width'])

normalized_dataset.describe()
```

After the normalization process, our minimum and maximum values changed.



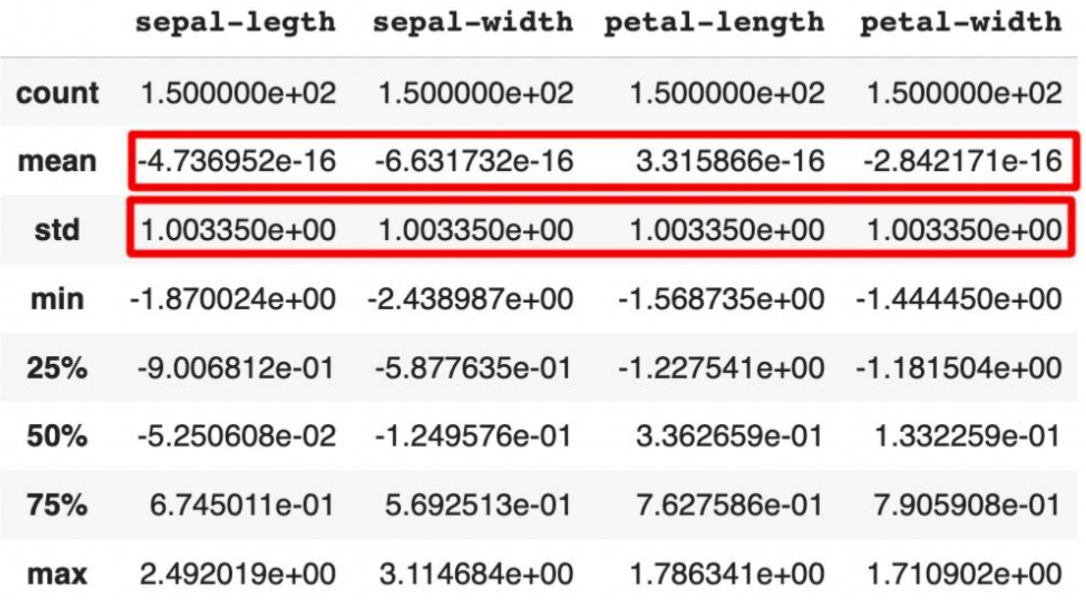
	sepal-legth	sepal-width	petal-length	petal-width
count	150.000000	150.000000	150.000000	150.000000
mean	0.428704	0.439167	0.467571	0.457778
std	0.230018	0.180664	0.299054	0.317984
min	0.000000	0.000000	0.000000	0.000000
25%	0.222222	0.333333	0.101695	0.083333
50%	0.416667	0.416667	0.567797	0.500000
75%	0.583333	0.541667	0.694915	0.708333
max	1.000000	1.000000	1.000000	1.000000

Standardization Code:



```
from sklearn.preprocessing import StandardScaler

stn_scalar = StandardScaler()

standardized_dataset =stn_scalar.fit_transform(dataset)

standardized_dataset = pd.DataFrame(standardized_dataset,
columns =['sepal-legth','sepal-width','petal-length','petal-width'])

standardized_dataset.describe()
```

After the standardization process, our mean and standard deviation changed.



	sepal-legth	sepal-width	petal-length	petal-width
count	1.500000e+02	1.500000e+02	1.500000e+02	1.500000e+02
mean	-4.736952e-16	-6.631732e-16	3.315866e-16	-2.842171e-16
std	1.003350e+00	1.003350e+00	1.003350e+00	1.003350e+00
min	-1.870024e+00	-2.438987e+00	-1.568735e+00	-1.444450e+00
25%	-9.006812e-01	-5.877635e-01	-1.227541e+00	-1.181504e+00
50%	-5.250608e-02	-1.249576e-01	3.362659e-01	1.332259e-01
75%	6.745011e-01	5.692513e-01	7.627586e-01	7.905908e-01
max	2.492019e+00	3.114684e+00	1.786341e+00	1.710902e+00