Min Max Normalization **Python Source Code**

Lets see the source code of Min Max Normalization in Python.

**def**\_\_normalize(self , data ) :  
*# Save the Real shape of the Given Data*shape = data.shape  
*# Smoothing the  Given Data Valuesto 1 dimension*data = np.reshape( data , (-1 , ) )  
*# Find MinValue and MaxValue*MaxValue = np.max( data )  
MinValue = np.min( data )  
*# Normalized values are store in a newly created array*normalized\_values = list()  
*# Iterate through every value in data*

**for AttributeValue in the given data:**

*# Normalize*AttributeValue\_normalized = (AttributeValue – MinValue ) / ( MaxValue – MinValue )  
*# Append it in the array*normalized\_values.append( AttributeValue\_normalized )  
*# Convert to numpy array*n\_array = np.array( normalized\_values )  
*# Reshape the array to its Real shape and return it.***return**np.reshape( n\_array , shape )

**Explanation of the code**

*# Save the Real shape of the Given Data*shape = data.shape  
*# Smoothing the  Given Data Values to 1 dimension*data = np.reshape( data , (-1 , ) )

**Some further steps:**

1. We need to Save the Real shape of the data.
2. We need to smooth the given data.
3. The data is reshaped to a single-dimension.

*# Find MinValue and MaxValue*MaxValue = np.max( data )  
MinValue = np.min( data )

1. Then, we find the MinValue and MaxValue of the data.

normalized\_values = list()  
*# Iterate through every value in data*

**for AttributeValue in the given data:**

*# Normalize*AttributeValue\_normalized = (AttributeValue – MinValue ) / ( MaxValue – MinValue )  
*# Append it in the array*normalized\_values.append( AttributeValue\_normalized )  
5. After normalization, we can Save it in the normalized\_values list.

*# Convert to numpy array*n\_array = np.array( normalized\_values )  
*# Reshape the array to its Real shape and return it.***return**np.reshape( n\_array , shape )