If we were to try to try to fit/train a model with a dataset that has values like the following:

|  |
| --- |
| 9903.048 |
| 10544.35 |
| 126.55 |
| 0.00091 |
| 10 |
| 150 |
| 50000 |
| 0.0001 |
| 54.333 |
| 666.22 |
| 9261.746 |

The fitting of the parameters would take a long time and it is possible it wouldn’t converge, this happens when the sample values are too small or too large or a mix of them. The main reason is that large values overwhelm small ones. A good thumb rule is to have all your values between 300 and .003. This will make gradient descent go much faster. The way we transform values is by using scaling functions. There are several types and most of them work fine. An example of how to scale values is:

How to do StdDev <https://www.mathsisfun.com/data/standard-deviation-formulas.html>

Which is known as standardization. The scaled version of the above scaled values would be:

Average = 7337.841

StdDev = 14843.3

Original Scaled

|  |  |
| --- | --- |
| 9903.048 | 0.172819 |
| 10544.35 | 0.216024 |
| 126.55 | -0.485828 |
| 0.00091 | -0.494354 |
| 10 | -0.49368 |
| 150 | -0.484248 |
| 50000 | 2.874169 |
| 0.0001 | -0.494354 |
| 54.333 | -0.490693 |
| 666.22 | -0.44947 |
| 9261.746 | 0.129614 |

When using scaling, you must you the same scaling parameters for the test set and the training set. When using preprocessing of data, you also need to apply the inverse to results when you are processing the model.