Outlier Detection with Python

Outlier Detection in Machine Learning using Hypothesis Testing.

AMAN KHARWAL / NOVEMBER 12, 2020 / MACHINE LEARNING

In this article, I will walk you through the task of outlier detection in machine learning. An outlier is a terminology commonly used by analysts and data scientists because it requires special attention, otherwise, it can lead to totally wrong estimates.

Simply put, outlier detection is an observation that appears far away from and diverges from an overall pattern in a sample.

Also, Read – Machine Learning Full Course for free.

What is Outlier?

An outlier is an observation that is numerically distant from the rest of the data or, in a nutshell, is the value that is out of range. Let's take an example to check what happens to a dataset with a dataset without outliers.

	Data without Outliers	Data with Outliers
Data	1, 2, 3, 3, 4, 5, 4	1, 2, 3, 3, 4, 5, **400**
Mean	3.142	**59.714**
Median	3	3
Standard Deviation	1.345185	**150.057**

As you can see, the dataset with outliers has a significantly different mean and standard deviation. In the first scenario, we will say that the average is 3.14. But with the outlier, the average climbs to 59.71. This would completely change the estimate.

Let's take a concrete example of an outlier. In a company of 50 employees, 45 people with a monthly salary of Rs. 6000, 5 seniors with a monthly salary of Rs. 100000 each. If you calculate the average monthly salary of the employees of the company is 14,500 rupees, which will give you a bad conclusion.

But if you take the median salary, it is Rs.6000 which is more sensitive than the average. For this reason, the median is an appropriate measure for the mean. Here you can see the effect of an outlier.

Now let's have a quick look at the main causes of outliers before getting started with the task of outlier detection:

1. Data Entry Errors: Human errors such as errors caused during data collection, recording, or entry can cause outliers in data.
2. Measurement Errors: It is the most common source of outliers. This is caused when the measurement instrument used turns out to be faulty.
3. Natural Outliers: When an outlier is not artificial (due to error), it is a natural outlier. Most real-world data belong to this category.

Outlier Detection in Machine Learning using Hypothesis Testing

Now, I will use the Python programming language for the task of outlier detection in machine learning.

An outlier can be of two types: Univariate and Multivariate. Above, we have discussed the example of a univariate outlier. These outliers can be found when we look at the distribution of a single variable. Multivariate outliers are outliers in an n-dimensional space.

An outlier can be of two types: univariate and multivariate. Above we have discussed the example of a univariate outlier. These outliers can be found when we look at the distribution of a single variable. Multivariate outliers are outliers in an n-dimensional space.

Hypothesis testing is a common technique for detecting outliers in machine learning. Hypothesis testing is a method of testing a claim or hypothesis about a parameter in a population, using data measured in a sample. In this method, we test a hypothesis by determining the probability that a sample statistic could have been selected, if the hypothesis regarding the population parameter was true.

The purpose of the hypothesis test is to determine the probability that a population parameter, such as the mean, is likely to be true. There are four steps in the hypothesis test:

1. State the assumptions.
2. Define the criteria for a decision.
3. Calculate the test statistic.
4. Make a decision.

Now let's see how to use the Python programming language to implement the hypothesis testing for the task of Outlier Detection in Machine Learning:

```
import numpy as np
import scipy.stats as stats
x = np.array([12,13,14,19,21,23])
y = np.array([12,13,14,19,21,23,45])
def grubbs_test(x):
    n = len(x)
    mean_x = np.mean(x)
```

```
    sd_x = np.std(x)
    numerator = max(abs(x-mean_x))
    g_calculated = numerator/sd_x
    print("Grubbs Calculated Value:",g_calculated)
    t_value = stats.t.ppf(1 - 0.05 / (2 * n), n - 2)
    g_critical = ((n - 1) * np.sqrt(np.square(t_value))) / (np.sqrt(n) * np.sqrt(n - 2 +
    print("Grubbs Critical Value:",g_critical)
    if g_critical > g_calculated:
        print("From grubbs_test we observe that calculated value is lesser than critical
    else:
        print("From grubbs_test we observe that calculated value is greater than critical
grubbs_test(x)
grubbs_test(y)
```

outlier.py hosted with ❤ by GitHub view raw

```
Grubbs Calculated Value: 1.4274928542926593
Grubbs Critical Value: 1.887145117792422
From grubbs_test we observe that calculated value is lesser than
critical value, Accept null hypothesis and conclude that there is
no outliers

Grubbs Calculated Value: 2.2765147221587774
Grubbs Critical Value: 2.019968507680656
From grubbs_test we observe that calculated value is greater than
critical value, Reject null hypothesis and conclude that there is
an outliers
```

One of the major problems with machine learning is an outlier. If you will neglect the outliers in the data, then it will result in the poor performance of your machine learning model.

I hope you liked this article on the task of outlier detection in Machine Learning using hypothesis testing and the Python programming language.

Aman Kharwal

Coder with the ♥ of a Writer || Data Scientist | Solopreneur | Founder

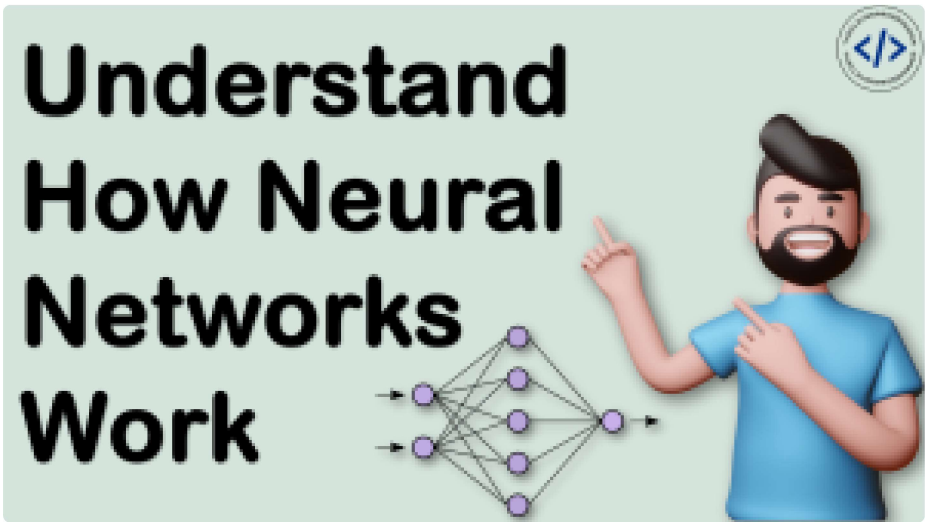
ARTICLES: 1202

PREVIOUS NEXT

Recommended For You

Best Approaches for Time Series Analysis

January 8, 2022



How Neural Network Works

January 6, 2022

Visualize a Machine Learning Algorithm using Python

December 29, 2021

Activation Functions in Neural Networks

December 23, 2021

Leave a Reply

Enter your comment here...

FACEBOOK INSTAGRAM MEDIUM LINKEDIN