Outliers Removal using Z-Score

Outliers, those mysterious data points that deviate significantly from the norm, can wreak havoc on analyses and models. Fortunately, the Z-score method offers a powerful tool to detect these outliers. In this article, we'll explore the Z-score technique, understand its significance, and provide a step-by-step code example in Python.

Understanding Z-Score: The Z-score measures how many standard deviations a data point is away from the mean. In other words, it quantifies the relative distance of a data point from the average. A high Z-score indicates that the data point is far from the mean, suggesting the possibility of an outlier.

The Formula: The formula to calculate the Z-score for a data point x is:

- $Z = \sigma x - \mu$

Where:

- μ is the mean of the dataset.
- σ is the standard deviation of the dataset.

Detecting Outliers with Z-Score:

1. Calculate the mean (μ) and standard deviation (σ) of the dataset.
2. For each data point (x) in the dataset, compute its Z-score.
3. Set a Z-score threshold (commonly 2 or 3) beyond which data points are considered outliers.
4. Identify data points with Z-scores greater than the threshold as potential outliers.

Python3

```
import numpy as np

# Generate example data
data = np.array([23, 25, 22, 27, 21, 24, 26, 100, 23, 28, 22, 29])

# Calculate mean and standard deviation
mean = np.mean(data)
std_dev = np.std(data)

# Set Z-score threshold
z_threshold = 2

# Calculate Z-scores
z_scores = [(x - mean) / std_dev for x in data]

# Identify outliers
outliers = [data[i] for i, z in enumerate(z_scores) if abs(z) > z_threshold]

print("Original Data:", data)
print("Outliers detected using Z-score:", outliers)
```

Interpreting Results: In the example above, data point 100 has a high Z-score, indicating that it's significantly far from the mean. This suggests that it might be an outlier. The Z-score threshold of 2 helps determine which data points to flag as outliers.

If you are facing any issue on this page. Please let us know.