|  |
| --- |
| Experiment No.4 |
| Perform Outlier Detection |
| Name : Dhaigude Vishal |
| Date of Performance:29/07/2024 |
| Date of Submission:05/08/2024 |

**AIM:** To perform Outlier Detection

**Objective:-** Develop a program to detect and handle outliers during the data pre-processing phase using the IQR method.

**Theory:**

Why preprocess the data?

In the real world, data is often messy, incomplete, and noisy. Preprocessing the data is crucial to handle these issues and improve the quality of the dataset before further analysis. One key issue in data is the presence of outliers, which are data points that deviate significantly from the majority of the dataset.

• What are Outliers?

Outliers are extreme values in the dataset that can skew results and affect the performance of data mining algorithms. They may arise due to data entry errors, equipment malfunctions, or natural variations in the data.

Outliers need to be identified and either removed or adjusted to improve the quality of the dataset. Outlier detection methods, such as the Interquartile Range (IQR) method, are designed to identify these extreme data points.

**Interquartile Range (IQR) Method:**

The Interquartile Range (IQR) method is a widely used technique for detecting outliers. It is based on the spread of the middle 50% of the data.

1. **Calculate Q1 (First Quartile)** – The 25th percentile of the data.
2. **Calculate Q3 (Third Quartile)** – The 75th percentile of the data.
3. **Calculate IQR** – The difference between Q3 and Q1:

IQR=Q3−Q1

1. **Define Outlier Boundaries**:

o Lower Boundary: Q1−1.5×IQR o Upper Boundary: Q3+1.5×IQR

Any data point falling below the lower boundary or above the upper boundary is considered an outlier.

**Code and output:**

import numpy as np

data = [22, 24, 26, 28, 29, 31, 35, 37, 41, 53, 64]

data = np.array(data) Q1 = np.percentile(data, 25)

Q3 = np.percentile(data, 75)

# Compute the Interquartile Range (IQR)

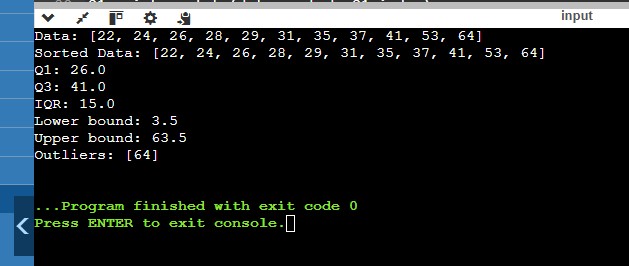
IQR = Q3 - Q1

# Determine outlier bounds lower\_bound = Q1 - 1.5 \* IQR upper\_bound = Q3 + 1.5 \* IQR

# Find outliers

outliers = data[(data < lower\_bound) | (data > upper\_bound)

print(f"Q1 (25th percentile): {Q1}") print(f"Q3 (75th percentile): {Q3}") print(f"IQR: {IQR}") print(f"Lower bound for outliers: {lower\_bound}") print(f"Upper bound for outliers: {upper\_bound}") print(f"Outliers: {outliers}")



**Conclusion**: In conclusion, performing outlier detection in data warehouse management is essential for ensuring data quality and accuracy. Identifying and addressing outliers helps prevent skewed analyses and enhances the reliability of insights derived from the data. By effectively managing outliers, organizations can improve decision-making and maintain the integrity of their data-driven strategies.