

18-08-2025
Expt No: 4

Experiment to detect outliers in a given dataset

Description: Understand the procedure to identify these outliers in a given dataset.

Soln: - Aim: To detect and report outliers in a given dataset (assuming) Interquartile Range method.

Algorithm:

1. Sort the data
2. Compute Q_1, Q_2, Q_3 lower, median, upper
3. Compute $IQR = Q_3 - Q_1$

Compute lower bound $= Q_1 - 1.5 \times IQR$ and upper bound $= Q_3 + 1.5 \times IQR$

5. Any point $<$ lower bound or $>$ upper bound is an outlier. (non-outliers are non-outliers)

Program

```
import numpy as np
import matplotlib.pyplot as plt

data = np.array([27, 50, 44, 6, 58, 61, 23, 86, 67, 20,
                 75, 7, 74, 61, 90, 54])

Q1 = np.percentile(data, 25)
Q2 = np.percentile(data, 50)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = data[(data < lower_bound) | (data > upper_bound)]
non_outliers = data[(data >= lower_bound) & (data <= upper_bound)]

print("Data:", data)
print("Q1 =", Q1, "Q2 (median) =", Q2, "Q3 =", Q3)
print("IQR =", IQR)
print("Lower bound (lr) =", lower_bound)
print("Upper bound (ur) =", upper_bound)
print("Outliers detected:", list(outliers))
print("Non-outliers:", list(non_outliers))
```

```
plt.figure()
plt.hist(data, bin=8)
plt.title("Histogram of data")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```

```
plt.figure()
plt.boxplot(data, vert=False)
plt.title("Boxplot (IQR)")
plt.xlabel("Value")
plt.show()
```

Output :

$Q1 = 26.0$ (25th Percentile)

$Q2$ (Median) = 56.0

$Q3 = 69.0$

$IQR = 43.0$

Lower bound (lr) = 38.5

Upper bound (ur) = 133.5

Outliers detected : [] (no outliers)

Non-outliers : all 16 data points (since none fall outside the bounds)

Experiment 1: Feature Scaling
Description: This experiment demonstrates the importance of feature scaling in machine learning models.

To understand the importance of feature scaling, by applying standardization as a normalization technique, we can ensure that all features have a similar range.

Requirements:
• Support necessary libraries (numpy, pandas, sklearn).

Load the dataset: `data = load_data()`

Handle missing values: `data = data.dropna()`

Encode categorical data: `data = data.drop('category', axis=1)`

Feature preprocessing steps:

1. Apply scaler on numeric features.

Result:

Thus, no outliers were found, since all data points lie within the calculated bounds.