
UNIT 6 – OUTLIER DETECTION

- 6.0 Introduction
- 6.1 Expected Learning Outcomes
- 6.2 Standard deviation method
- 6.3 Box-plots
- 6.4 Z-score
- 6.5 Percentiles
- 6.6 DBSCAN
- 6.7 Local Outlier factor
- 6.8 Summary
- 6.9 Answers
- 6.10 References/Further Readings

6.0 INTRODUCTION

Outliers are data points that significantly differ from the majority of observations in a dataset. In predictive data analysis, they may arise due to measurement errors, data entry mistakes, or genuine but rare events. Understanding outliers is important because they can heavily influence statistical measures such as mean and standard deviation, and may distort patterns learned by predictive models, leading to inaccurate results. At the same time, not all outliers are errors—some may carry valuable insights, such as fraud detection, rare diseases, or unusual customer behaviour.

For example, in a credit card transaction dataset, most purchases may range between ₹100 and ₹5,000, but a sudden transaction of ₹2,00,000 could appear as an outlier. This unusual value might indicate fraudulent activity rather than normal spending behavior. Identifying such outliers helps financial institutions take timely action, improving both security and decision-making in predictive systems.

In Unit 5, *Similarity Measures*, we explored how data points can be compared based on distance, proximity, and resemblance using various mathematical metrics. These measures form the backbone of many predictive data analysis techniques, enabling tasks such as clustering, classification, and recommendation systems. However, an important challenge arises when certain observations do not conform to the general structure of the data. Such atypical observations—known as outliers—can significantly distort similarity computations, leading to misleading patterns and inaccurate predictive models.

Now, this Unit 6, *Outlier Detection*, is directly based upon the concepts of similarity by focusing on identifying and handling these anomalous data points. Techniques such as the standard deviation method, Z-score, and percentiles rely on statistical dispersion to detect

deviations from the norm, while box-plots provide intuitive visual mechanisms for spotting extreme values. Moving beyond purely statistical approaches, methods like DBSCAN and Local Outlier Factor (LOF) leverage similarity and density concepts introduced in this unit to detect outliers in complex, high-dimensional datasets. By understanding how outliers influence similarity relationships and model behaviour, this unit equips learners with essential tools to improve data quality, enhance model robustness, and ensure more reliable predictive analytics outcomes.

6.1 EXPECTED LEARNING OUTCOMES

- Understand the concept and significance of outliers in predictive data analysis.
- Apply the standard deviation method to identify deviations from the mean.
- Use Z-score to quantify how far a data point lies from the average.
- Interpret percentiles to detect extreme values within a dataset.
- Utilize box-plots for visual identification of outliers.
- Implement DBSCAN to detect noise and outliers based on data density.
- Analyse anomalies using the Local Outlier Factor (LOF) method.
- Evaluate the impact of outliers on similarity measures and predictive models.

6.2 STANDARD DEVIATION METHOD

The standard deviation method is one of the most widely used statistical techniques for identifying how much individual data points deviate from the mean (average) of a dataset. In the context of outlier detection, it helps us determine whether a data point lies unusually far from the central tendency of the data.

Standard deviation measures the spread or dispersion of data around the mean.

- A small standard deviation indicates that values are close to the mean.
- A large standard deviation indicates that values are widely spread out.

In outlier detection, observations that lie far beyond the typical spread (usually more than 2 or 3 standard deviations from the mean) are considered potential outliers. To understand this statement, we need to understand a few fundamentals like Mean, Variance etc. Their respective formulas and related context we explore few numerical given below:

Example: Consider the dataset representing daily sales (in ₹): 100, 110, 105, 98, 102, 500, does this data hint you for outlier, Check it.

Solution: We know the Formula for Mean (μ) is given by $\mu = (\sum xi)/N$

Consider the dataset representing daily sales (in ₹): 100,110,105,98,102,500; the mean (μ)

$$\mu = \{100 + 110 + 105 + 98 + 102 + 500\} / \{6\} = 169.17$$

The mean (μ) is ₹169.17, which is unusually high in comparison to most of the values in the given data set, and this is due to the extreme value (₹500), hinting at a possible outlier.

Next, we need to explore Variance and Its Relation to Standard Deviation, actually Variance measures the average squared deviation from the mean, and it is represented by the formula $\sigma^2 = \sum (x_i - \mu)^2 / N$. Whereas Standard deviation (σ) is simply the square root of variance (σ^2)

Now, let's explore Variance and Its Relation to Standard Deviation for Outlier detection.

Step 1: Compute deviations from mean ($\mu = 169.17$)

Value (xi)	$x_i - \mu$	$(x_i - \mu)^2$
100	-69.17	4784.49
110	-59.17	3501.09
105	-64.17	4127.79
98	-71.17	5065.15
102	-67.17	4511.79
500	330.83	109448.69

Step 2: Compute Variance

$$\sigma^2 = (4784.49 + 3501.09 + 4127.79 + 5065.15 + 4511.79 + 109448.69) / 6 = 21906.50$$

Step 3: Compute Standard Deviation $\sigma \approx 148.02$

Now, let's, analyse the results for Identifying Outliers Using Standard Deviation, the common rule for outlier detection is as follows:

- Values beyond $\mu \pm 2\sigma \rightarrow$ Possible outliers
- Values beyond $\mu \pm 3\sigma \rightarrow$ Strong outliers

Based on this said rule let's Calculate Range for Lower & Upper Bounds respectively:

- Lower bound ($\mu \pm 2\sigma$): $169.17 - (2 \times 148.02) = -126.87$
- Upper bound ($\mu \pm 3\sigma$): $169.17 + (2 \times 148.02) = 465.21$

The Interpretation of the Results consolidates that the Most data points (100, 110, 105, 98, 102) fall within the acceptable range. However, the value 500 exceeds the upper bound (465.21) hence identified as an outlier. The presence of ₹500 significantly inflated both the mean and standard deviation, indicating that extreme values can distort statistical measures. Detecting and handling such outliers is essential for building accurate predictive models.

So, we learned that the standard deviation method provides a systematic way to measure deviations from the mean and identify unusual observations. By understanding its relationship with variance and applying threshold rules (such as $\pm 2\sigma$ or $\pm 3\sigma$), analysts can effectively detect outliers and improve the reliability of predictive data analysis.

6.3 Z-SCORE

We learned about the role of standard deviation in outlier detection, in the section 6.2 above. Here in this section 6.3, we are going to discuss the next important characteristic for outlier detection i.e. Z-score.

The Z-score (or standard score) is a statistical measure that indicates how far a data point lies from the mean in terms of standard deviations. It standardizes data, allowing us to compare observations across different scales. In outlier detection, the Z-score is especially useful because it provides a clear numerical measure of extremeness—helping identify values that deviate significantly from the norm.

Z-score, represents a standardized value that indicates how many standard deviations a particular data point is away from the mean of a dataset. The Z-score is essentially a normalized form of deviation that expresses distance from the mean in units of standard deviation. This relationship makes it a powerful tool in predictive data analysis, especially for detecting outliers, comparing datasets, and standardizing variables for further modeling.

Before starting this section, we need to understand that the Z-score and standard deviation are intrinsically connected, as the Z-score is fundamentally derived using the standard deviation to measure how far a data point lies from the mean, and the formula for computation of Z-Score value is:

$$Z = (x - \mu) / \sigma$$

Where:

- x = individual data point
- μ = mean of the dataset
- σ = standard deviation

Basic Understanding of Z-Score

- A Z-score = 0 $\rightarrow$ the data point is exactly at the mean.
- A positive Z-score $\rightarrow$ the value is above the mean.
- A negative Z-score $\rightarrow$ the value is below the mean.
- Larger absolute values ($|Z|$) indicate greater deviation.

The Conceptual Relationship between Standard deviation (σ) and Z-score (Z) are as follows:

- The Standard deviation (σ) measures the overall spread of a dataset—how much the data varies around the mean, and
- The Z-score (Z) uses this spread to determine the relative position of an individual data point.

In simple terms, Standard deviation (σ) describes the dataset, while Z-score describes a single observation within that dataset.

The key insight to the formula ($Z = (x - \mu) / \sigma$) is that without standard deviation, Z-score cannot be computed, and without Z-score, interpreting standard deviation at the individual data point level becomes difficult. That means that they work together such that Standard deviation (σ) provides the scale and Z-score provides the position on that scale

The relationship between Z-Score & σ , becomes particularly important in identifying outliers:

- Standard deviation defines the expected spread.
- Z-score identifies whether a value lies within or outside this spread.

The Typical rule for identifying outliers' states following:

- $|Z| \geq 2 \rightarrow$ Possible outlier
- $|Z| \geq 3 \rightarrow$ Strong outlier

In practice, the following thresholds are commonly used:

- $|Z| < 2 \rightarrow$ Normal data point
- $2 \leq |Z| < 3 \rightarrow$ Potential outlier
- $|Z| \geq 3 \rightarrow$ Strong outlier

This rule is based on the empirical distribution of data (assuming approximate normality), where most values lie within ± 3 standard deviations.

Thus, we can say that Z-score operationalizes standard deviation for practical decision-making.

Example: Consider the dataset (daily transaction amounts in ₹): 50, 55, 52, 48, 51, 300

Analyse the given data and determine the outliers using Z-Score

Solution:

Step 1: Calculate Mean (μ) = $(50+55+52+48+51+300) / 6 = 556/6 = 92.67$

Step 2: Calculate Standard Deviation (σ) ≈ 92.10

Step 3: Compute Z-Scores for Each Value

Value (x)	$Z = (x - \mu)/\sigma$	Interpretation
50	$(50 - 92.67)/92.10 = -0.46$	Close to mean
55	-0.41	Normal
52	-0.44	Normal
48	-0.48	Normal
51	-0.45	Normal
300	$(300 - 92.67)/92.10 = 2.25$	Potential outlier

Interpretation of Results tabulated above identifies that Most values have Z-scores close to 0, indicating they are near the mean. The value 300 has a Z-score of 2.25, which lies beyond ± 2 hence, flagged as a potential outlier. It is to be noted that if a stricter threshold (± 3) is used, then the value 300 may not be considered an extreme outlier, but still requires attention.

☛ Check Your Progress 1

Q1. What are outliers? Explain their significance in predictive data analysis.

.....

.....

Q2. Explain the Standard Deviation method for detecting outliers.

.....

.....

Q3. Define Z-score and explain its role in outlier detection.

.....

.....

Q4. Differentiate between Standard Deviation method and Z-score method.

.....

.....

Q5. Explain the relationship between Standard Deviation and Z-score in outlier detection.

.....

.....

6.4 PERCENTILES

Quantiles are positional measures that divide a dataset into equal parts and provide a deeper understanding of the distribution of data. Unlike measures such as mean, quantiles focus on the relative position of observations, making them highly useful in identifying the spread, concentration, and presence of extreme values (outliers). *The most commonly used quantiles are quartiles, deciles, and percentiles. In this section we are going to discuss these Quantiles with their relevance to Outlier detection.*

Outlier detection is an important aspect of data analysis as extreme values can significantly distort results and lead to incorrect interpretations. In this context, positional measures such as quantiles, percentiles, deciles, and median play a crucial role because they are based on the relative position of data rather than its magnitude. This makes them robust and less sensitive to extreme values, unlike the arithmetic mean.

We had already discussed the concept of Deciles, Quantiles and median in unit-1, here we are going to extend our discussion on these parameters for the detection of outliers. Before starting the mathematical aspects of the Percentiles, Deciles, Quantiles and median; we need to understand their relevance in the context of outlier detection.

Quantiles provide the overall framework for understanding the distribution of data by dividing it into equal parts. They help in identifying how observations are spread across the dataset and enable the setting of threshold boundaries beyond which values may be considered unusual. Since quantiles focus on order rather than value size, they are particularly effective in detecting outliers in skewed distributions.

The **Median**, which is the second quartile (Q_2), acts as a stable central reference point in the dataset. It divides the data into two equal halves and is not influenced by extreme observations. This makes it highly useful for identifying the presence of outliers indirectly. For instance, a large difference between the mean and median often indicates the presence of extreme values or skewness in the data. Thus, the median provides a reliable baseline against which unusually high or low values can be compared.

Quartiles, which are specific types of quantiles, are the most widely used measures for direct outlier detection. The first quartile (Q_1) and the third quartile (Q_3) define the spread of the middle 50% of the data, known as the **interquartile range (IQR)**. The IQR is calculated as $Q_3 - Q_1$, and it forms the basis for identifying outliers using standard statistical rules. Any value lying below $Q_1 - 1.5 \times \text{IQR}$ or above $Q_3 + 1.5 \times \text{IQR}$ is considered an outlier. This method is highly reliable because it is not affected by extreme values and works well even when the data is skewed.

Deciles further divide the dataset into ten equal parts, providing a more detailed segmentation than quartiles. They are useful in identifying extreme segments such as the lowest 10% or highest 10% of observations. Values lying beyond the first decile (D_1) or the ninth decile (D_9) can be treated as potential outliers. Deciles are particularly useful in applications such as performance evaluation, income distribution analysis, and risk assessment, where identifying extreme groups is important.

Percentiles offer the most precise level of data segmentation by dividing the dataset into one hundred equal parts. They are widely used for fine-grained outlier detection, especially in large datasets. Common practice involves identifying values below the 5th percentile (P_5) or above the 95th or 99th percentile (P_{95} or P_{99}) as outliers. Percentiles are extensively applied in fields such as education, finance, and machine learning, where detecting extreme observations is critical for decision-making and model accuracy.

So, we learned that quantiles, median, deciles, and percentiles together provide a comprehensive and robust framework for outlier detection. The median offers a stable central reference, quartiles enable structured boundary-based detection through the IQR method, while deciles and percentiles allow increasingly finer identification of extreme observations. Because these measures are position-based and not influenced by extreme values, they are highly reliable tools for analysing data distributions and identifying outliers effectively.

Now, we elaborate the mathematical aspects of quantiles, median, deciles, and percentiles with their respective formulas and supporting numerical.

Quartiles divide the data into four equal parts. There are three quartiles: Q_1 , Q_2 (median), and Q_3 . Q_1 represents the value below which 25% of observations lie, Q_2 divides the data into two equal halves, and Q_3 represents the value below which 75% of observations lie.

We already learned in Unit-1 that Median is the Second Quartile (Q2) and it relates to the Central Reference Point i.e. The median divides the dataset into two equal halves. The Outlier detection on the basis of Median follows the rules given below:

- If mean $\gg$ median $\rightarrow$ presence of high-end outliers
- If mean $\ll$ median $\rightarrow$ presence of low-end outliers
- If mean $\approx$ median $\rightarrow$ No significant outliers

The Example given below, clarifies these rules

Case 1: Mean $\gg$ Median (High-end Outliers)

Data: 2, 3, 4, 5, 6, 7, 50

Step 1: Calculate Mean: $Mean = \frac{2+3+4+5+6+7+5}{7} = \frac{77}{7} = 11$

Step 2: Calculate Median: Sorted data: 2, 3, 4, 5, 6, 7, 50 therefore Median = 5

Now, the Results are: Mean = 11 and Median = 5 i.e. Mean $\gg$ Median

So, it can be interpreted that the mean is much higher than the median because of the extreme value 50. This indicates the presence of a high-end outlier.

Case 2: Mean $\ll$ Median (Low-end Outliers)

Data: 1, 20, 22, 24, 25, 26, 27

Step 1: Calculate Mean: $Mean = \frac{1+20+22+24+25+26+27}{7} = \frac{145}{7} \approx 20.71$

Step 2: Calculate Median: Sorted data: 1, 20, 22, 24, 25, 26, 27 therefore Median = 24

Now, the Results are: Mean $\approx$ 20.71 and Median = 24 i.e. Mean $\ll$ Median

So, it can be interpreted that the mean is pulled down due to the very small value 1. This indicates the presence of a **low-end outlier**.

Case 3: Mean $\approx$ Median (No Significant Outliers)

Data: 10, 12, 14, 15, 16, 18, 20

Step 1: Calculate Mean: $Mean = \frac{105}{7} = 15$

Step 2: Calculate Median: Median = 15

Now the Results are: Mean = 15 and Median = 15 i.e. Mean $\approx$ Median

So, it can be interpreted that there is no significant difference between mean and median.

This indicates no extreme outliers and a symmetrical distribution.

This method is simple but effective for **quick detection of skewness and outliers**.

Now, let's extend our discussion to **Interquartile Range (IQR) Method for Outlier Detection**, we know that for grouped data, the formula for quartiles is:

$$Q_j = L + \frac{(pcf)}{f} \times i \quad \text{for } j = 1, 2, 3$$

where

L = lower limit of quartile class

N = total frequency

pcf = preceding cumulative frequency

f = frequency of quartile class

i = class interval

To perform Outlier detection using Quartiles, we need to understand that the Q1, Q2 & Q3 are the most widely used for outlier detection, and the same is supported by the **Interquartile Range (IQR) Method**, where $IQR = Q3 - Q1$ and the Outliers are identified by using the rules:

$$\text{Lower Bound} = Q1 - 1.5 \times IQR \quad \text{Upper Bound} = Q3 + 1.5 \times IQR$$

The quartile-based outlier rule says that any observation below "Lower Bound = $Q1 - 1.5 \times IQR$ "

or above " $Upper Bound = Q3 + 1.5 \times IQR$ " will be treated as an outlier.

Interquartile Range (IQR) Method is considered as the Most standard and reliable method because it Works well even for skewed data, and it is Used in box plots and data cleaning

Example: Determine the Inter Quartile Range (IQR) and comment on the existence of Outliers in the grouped data given below:

Class Interval	Frequency
0–10	5
10–20	9
20–30	14
30–40	8
40–50	4

Solution:

Total frequency, $N = (5+9+14+8+4) = 40$

Prepare the Cumulative frequency table:

Class Interval	Frequency	Cumulative Frequency
0–10	5	5
10–20	9	14
20–30	14	28
30–40	8	36
40–50	4	40

Calculation of Q1

$$Q_1 = L + \frac{(pcf)}{f} \times i \quad Q_1 = 10 + \frac{(5)}{9} \times 10 \quad Q_1 = 10 + \frac{5}{9} \times 10 = 10 + 5.56 = 15.56$$

So,

$$Q_1 = 15.56$$

Calculation of Q_3

$$Q_3 = L + \frac{(pcf)}{f} \times iQ_3 = 30 + \frac{(28)}{8} \times 10Q_3 = 30 + \frac{2}{8} \times 10 = 30 + 2.5 = 32.5$$

So,

$$Q_3 = 32.50$$

Calculation of IQR

$$IQR = Q_3 - Q_1 = 32.50 - 15.56 = 16.94$$

So,

$$IQR = 16.94$$

Lower Bound for Outlier Detection

$$Lower\ Bound = Q_1 - 1.5(IQR) = 15.56 - 1.5(16.94) = 15.56 - 25.41 = -9.85$$

So,

$$Lower\ Bound = Q_1 - 1.5(IQR) = -9.85$$

Upper Bound for Outlier Detection

$$Upper\ Bound = Q_3 + 1.5(IQR) = 32.50 + 1.5(16.94) = 32.50 + 25.41 = 57.91$$

So,

$$Upper\ Bound = Q_3 + 1.5(IQR) = 57.91$$

The Analysis of the Obtained results can be summarized as follows:

The quartile-based outlier rule says that any observation below Lower bound i.e. -9.85 or above Upper Bound i.e. 57.91 will be treated as an outlier. In this dataset, the class intervals range from 0 to 50, which means all observations fall within the acceptable range. Therefore, no outliers are detected using the quartile method.

This shows that the data is fairly balanced and does not contain any extreme unusually low or high values. Quartiles are useful for outlier detection because they are based on positional values and are not heavily affected by extreme observations. That is why the IQR method is considered a reliable method for detecting outliers, especially in skewed data.

Now, we are in the position to extend our discussion for Deciles and Percentiles. We begin with Deciles and finally we conclude with Percentiles.

Deciles are the Quantiles that divide the data into ten equal parts, denoted as $D_1, D_2, \dots, D_9$. Each decile represents 10% of the data distribution, providing a more detailed segmentation than quartiles.

For grouped data, the formula is:

$$D_k = L + \frac{(pcf)}{f} \times i \text{ for } k = 1, 2, \dots, 9$$

where,

- $D_k \rightarrow$ The k^{th} decile (e.g., D_1, D_5, D_9)
- $k \rightarrow$ Decile number (1 to 9)
- $N \rightarrow$ Total number of observations (sum of frequencies)
- $\frac{kN}{10} \rightarrow$ Position of the k -th decile in the dataset
- $L \rightarrow$ Lower limit of the class interval containing the k -th decile
- $pcf \rightarrow$ Preceding cumulative frequency (cumulative frequency before the decile class)
- $f \rightarrow$ Frequency of the decile class
- $i \rightarrow$ Class interval width

Before using the deciles for outlier detection let's understand how to calculate them by using the formula given above.

Example: Determine the D_5 and comment on the existence of Outliers in the grouped data given below:

Class Interval	Frequency
0–10	5
10–20	9
20–30	14
30–40	8
40–50	4

Solution:

$$N = (5+9+14+8+4) = 40$$

Position of the k^{th} decile in the dataset $= \frac{kN}{10}$ therefore,

$$D_5 = \frac{5N}{10} = \frac{5 \times 40}{10} = 20^{\text{th}} \text{ observation}$$

From cumulative frequency:

Class	c.f
10–20	14
20–30	28

So D_5 lies in class 20–30

Using the formula: $D_k = L + \frac{(pcf)}{f} \times i$ for $k = 1, 2, \dots, 9$

$$\text{We get, } D_5 = 20 + \frac{(14)}{14} \times 10 = 20 + \frac{6}{14} \times 10 = 20 + 4.29 = 24.29$$

For Conceptual Understanding of Decile calculation, the formula

$$D_k = L + \frac{(pcf)}{f} \times i \text{ for } k = 1, 2, \dots, 9 \text{ works in two steps:}$$

Step 1: Locate the Position: $\frac{kN}{10}$; This tells us the position of the decile in the ordered dataset.

Step 2: Interpolation within the Class: Since data is grouped, we assume values are evenly distributed within the class. The formula: $\frac{(pcf)}{f} \times i$ calculates how far inside the class the decile lies.

The formula finds the exact value of the decile within a class interval, It adjusts the lower limit (L) based on how deep the required observation lies in that class, Essentially, it is a **linear interpolation method**

Let's apply this understanding to perform Calculation of Deciles (D1)

$$\frac{1 \times 40}{10} = 4^{th} \text{ observation}$$

The 4th observation lies in class 0–10.

$$D_1 = 0 + \frac{(0)}{5} \times 10 = 8 \quad D_1 = 8$$

Similarly, we can calculate other deciles from D_1 to D_9 the values obtained are:

$D_1 = 8$, $D_2 = 13.33$, $D_3 = 17.78$, $D_4 = 21.43$, $D_5 = 24.29$, $D_6 = 27.14$, $D_7 = 30$; $D_8 = 35$, $D_9 = 40$

It is to be noted that the deciles divide the data into ten equal parts and help us understand how the observations are spread across the distribution. Here, the first decile is $D_1 = 8$, which means nearly 10% of the observations are below 8, and the ninth decile is $D_9 = 40$, which means nearly 90% of the observations are below 40.

For outlier detection using deciles, following are the rules :

- values far below **D1** may be treated as unusually low,
- values far above **D9** may be treated as unusually high.
- Further, the Values below D_1 or above D_9 may indicate extreme observations, and it is Useful in ranking (top 10%, bottom 10%).
- The K^{th} Decile (D_k) gives the value below which $k \times 10\%$ data lies the Formula uses the Position $\rightarrow \frac{kN}{10}$ and Adjustment i.e. interpolation within class, and it converts grouped data into a precise positional value

It can be interpreted from the result $D_5 = 24.29$ that 50% of observations lie below 24.29 and 50% lie above 24.29.

In the given dataset, the observations are spread smoothly from 0 to 50 without any large jump or isolated extreme class. The decile values also increase gradually, which suggests that the distribution is fairly regular. Therefore, based on deciles, there is **no strong evidence of extreme outliers** in the data.

- About 10% of values lie below **8**
- About 50% of values lie below **24.29**
- About 90% of values lie below **40**

So, the data appears reasonably distributed, and **no serious outlier problem is indicated by the deciles.**

Finally, it's time to conclude this section with the understanding of Percentiles, they are the Quantiles that divide the data into one hundred equal parts and are denoted as $P_1, P_2, \dots, P_{99}$. They provide the most detailed insight into the distribution of data.

The formula for grouped data is:

$$P_k = L + \frac{(pcf)}{f} \times i \text{ for } k = 1, 2, \dots, 99$$

Again, **the Conceptual Understanding of Percentile calculation**, the formula

$$D_k = L + \frac{(pcf)}{f} \times i \text{ for } k = 1, 2, \dots, 99 \text{ works in two steps:}$$

Step 1: Locate the Position: $\frac{kN}{100}$; This tells us the position of the percentile in the ordered dataset.

Step 2: Interpolation within the Class: Since data is grouped, we assume values are evenly distributed within the class. The formula: $\frac{(pcf)}{f} \times i$ calculates how far inside the class the percentile lies.

Example: Determine the P_{80} and comment on the existence of Outliers in the grouped data given below:

Class Interval	Frequency
0–10	5
10–20	9
20–30	14
30–40	8
40–50	4

Solution:

Step 1: Locate the Position (P_k): $\frac{kN}{100} \Rightarrow P_{80} = \frac{80N}{100} = \frac{80 \times 40}{100} = 32^{nd} \text{ observation}$

Step 2: Interpolation within the Class:

From cumulative frequency(c.f.):

Class	c.f
20–30	28
30–40	36

So P_{80} lies in class 30–40

$$P_{80} = 30 + \frac{(28)}{8} \times 10 = 30 + \frac{4}{8} \times 10 = 30 + 5 = 35$$

For outlier detection using percentiles, following are the rules:

- Values below P_5 implies low-end outliers
- Values above P_{95} or P_{99} implies high-end outliers

Percentiles are widely used in Exam rankings, Salary distribution, Machine learning preprocessing etc.

In this section we learned that the Quantiles, deciles, and percentiles provide a structured way to understand data distribution. Quartiles are most commonly used for outlier detection through the IQR method, while deciles and percentiles offer finer segmentation for identifying extreme observations. Since these measures are based on position rather than magnitude, they are robust and reliable even in skewed datasets, making them highly valuable tools in statistical analysis and decision-making.

These Quartiles are used to develop boxplots, we are going to discuss them in the next section of this unit.

6.5 BOX-PLOTS

The Box plots are widely used for outlier detection, skewness analysis, and spread visualization. Actually, Box Plot (or Box-and-Whisker Plot) is a graphical representation of data distribution that summarizes a dataset using five key statistics given below:

- Minimum
- First Quartile (Q1)
- Median (Q2)
- Third Quartile (Q3)
- Maximum

To construct a box plot, we first compute:

1. Minimum value
2. Q1 (25th percentile)
3. Median (Q2)
4. Q3 (75th percentile)
5. Maximum value

Then calculate:

$$IQR = Q3 - Q1$$

Outlier Detection Rule

$$Lower\ Bound = Q1 - 1.5 \times IQR \quad Upper\ Bound = Q3 + 1.5 \times IQR$$

- Values outside these bounds are referred as Outliers
- Values within bounds are referred as Normal data

We already learned the calculation of these statistical parameters in the section 6.4 , given above.

Now let's discuss the Steps to Create a Box Plot :

1. Arrange data in ascending order
2. Find median (Q2)
3. Find Q1 and Q3
4. Compute IQR
5. Identify outliers using bounds
6. Draw:
 - A box from Q1 to Q3
 - A line inside box at median
 - Whiskers to min & max (excluding outliers)
 - Plot outliers as separate points

Example: Draw Box Plot for the Data: 5, 7, 8, 9, 10, 12, 13, 15, 18, 22, 50

Solution:

Step 1: Calculate Median (Q2) : Total observations = 11

$$Q2 = 6^{th} \text{ value} = 12$$

Step 2: Q1 (Median of lower half) : Lower half: 5, 7, 8, 9, 10

$$Q1 = 8$$

Step 3: Q3 (Median of upper half) : Upper half: 13, 15, 18, 22, 50

$$Q3 = 18$$

Step 4: IQR: $= Q3 - Q1 = 18 - 8 = 10$

Step 5: Outlier Bounds

$$\text{Lower Bound} = 8 - 1.5(10) = 8 - 15 = -7 \quad \text{Upper Bound} = 18 + 1.5(10) = 18 + 15 = 33$$

Step 6: Identify Outliers

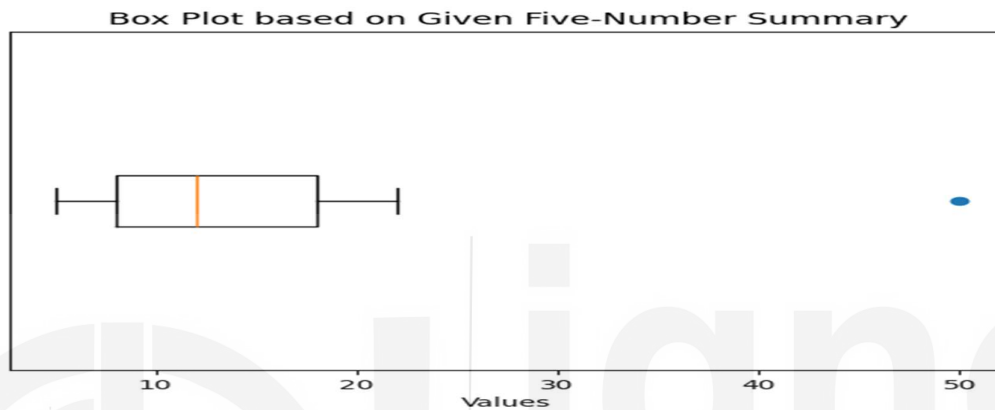
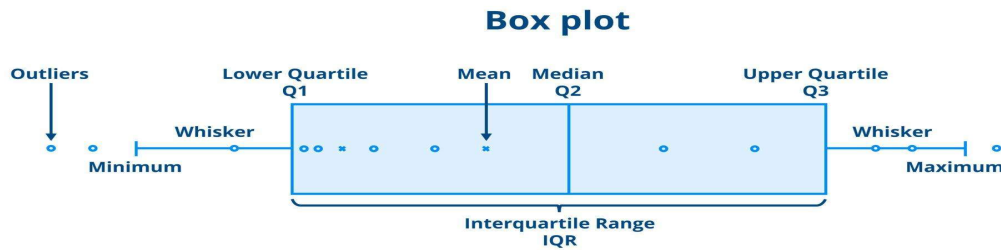
- Data values > 33 i.e. 50 is an outlier
- No lower outlier (since $\min = 5 > -7$)

Final Five-Number Summary

- Minimum = 5
- Q1 = 8
- Median = 12
- Q3 = 18
- Maximum (without outlier) = 22
- Outlier = 50

Interpretation of the Box Plot

The box (8 to 18) shows middle 50% of data
The median (12) divides data into two halves
The whiskers (5 to 22) show normal range
The point 50 lies far beyond the upper bound
→ outlier



It can read from the Box plot that Left whisker is Minimum = 5, the Box starts at Q1 = 8 followed by a Middle line (Q2) i.e. Median = 12 and the Box ends at Q3 = 18. The Right whisker is Maximum (without outlier) = 22 and Point on far right is the Outlier = 50.

It can be Interpreted from the Box plot that the central 50% of data lies between 8 and 18 ; and the Median (12) is closer to Q1 slightly right skewness (Data is right-skewed because of high outlier 50) and the value 50 is clearly separated, confirming it as a high-end outlier

We learned the preparation of Box plots, which is an important tool for Outlier Detection. It clearly highlights extreme values visually and uses IQR, a robust measure, not affected by outliers. It Helps in Data cleaning, Fraud detection, Statistical modelling, Comparing multiple datasets

Also, Box plots are also considered as a simple yet powerful tool to summarize data distribution, detect outliers using IQR, understand skewness and spread. In practice, box plots with the IQR method are among the most reliable techniques for outlier detection in statistics and data science.

*We learned to draw Box Plot for the Discrete data now let's understand **how to create the Box plot for the grouped data***

We learned that a Box-and-Whisker Plot visually displays the distribution, spread, and skewness of a dataset through five summary statistics:

- Minimum
- First Quartile (Q1)

- Median (Q2)
- Third Quartile (Q3)
- Maximum

Example: Construct the Box Plot for the Grouped data given below:

Class Interval	Frequency	Midpoint
48–53	4	50.5
54–59	6	56.5
60–65	6	62.5
66–71	6	68.5
72–77	6	74.5
78–83	5	80.5
84–89	5	86.5
90–95	2	92.5

Solution:

Now, to create the Box plot for the grouped data, we need to follow following steps

Step 1: Reconstruct the Raw Data from Grouped Data

Since a Box Plot needs raw data or an approximation of it, reconstruct a pseudo dataset using the midpoints of each class interval and the corresponding frequency. Here's how:

Class Interval	Frequency	Midpoint
48–53	4	50.5
54–59	6	56.5
60–65	6	62.5
66–71	6	68.5
72–77	6	74.5
78–83	5	80.5
84–89	5	86.5
90–95	2	92.5

Total observations (N) = 40

Construct Pseudo Raw Data: i.e. expand the data based on the frequency:

$50.5 \times 4 \rightarrow$ positions 1–4
 $56.5 \times 6 \rightarrow$ positions 5–10
 $62.5 \times 6 \rightarrow$ positions 11–16
 $68.5 \times 6 \rightarrow$ positions 17–22
 $74.5 \times 6 \rightarrow$ positions 23–28
 $80.5 \times 5 \rightarrow$ positions 29–33
 $86.5 \times 5 \rightarrow$ positions 34–38

$$92.5 \times 2 \rightarrow \text{positions } 39-40$$

Step 2: Arrange the Data in Ascending Order

Arrange the values from smallest to largest. Since the data is already binned using midpoints, and each midpoint is repeated according to its frequency, it's already structured to represent an ordered dataset.

Step 3: Find the Five-Number Summary i.e. Min., Max, Q1, Q2, and Q3

Using the pseudo raw data given above:

- Minimum: Smallest value = 50.5
- Maximum: Largest value = 92.5
- Median (Q2): Middle value (20th and 21st observation average) ≈ 68.5
- First Quartile (Q1): 25th percentile (10th observation) ≈ 58.0
- Third Quartile (Q3): 75th percentile (30th observation) ≈ 80.5

So, the five-number summary is:

- Min = 50.5
- Q1 = 58.0
- Median (Q2) = 68.5
- Q3 = 80.5
- Max = 92.5

Let's understand how to determine the Quartiles Q1, Q2 and Q3; especially in the context of a dataset like the one we've been working with (40 student marks). You can use the same method for any dataset.

So, What Are Quartiles? we have 3 quartiles in general Q1, Q2, and Q3; their respective meaning is as follows:

- Q1 (First Quartile): The value that separates the lowest 25% of the data.
- Q2 (Median): The middle value; separates the lower 50% from the upper 50%.
- Q3 (Third Quartile): The value that separates the lowest 75% from the top 25%.

Now, we understand How to Determine Q1, Q2 and Q3 (Manually)

Assume you have $N = 40$ data points, arranged in ascending order.

Step A: Find the Positions

- Q1 position = $(N+1)/4 = (41)/4 = 10.25^{\text{th}}$ item
- Q2 position = $(N+1)/2 = (41)/2 = 20.5^{\text{th}}$ item
- Q3 position = $3(N+1)/4 = 3(41)/4 = 30.75^{\text{th}}$ item

These are positions, not values yet.

Step B: Estimate the Quartile Values

To get the values at 10.25th and 30.75th positions, you interpolate between the actual values at the 10th, 11th (for Q1), 20th, 21st (for Q2), and 30th, 31st (for Q3).

Suppose the ordered pseudo data looks like:

(Refer to Pseudo Raw Data given above)

$$50.5 \times 4 \rightarrow \text{positions } 1-4$$

$56.5 \times 6 \rightarrow$ positions 5–10

$62.5 \times 6 \rightarrow$ positions 11–16

$68.5 \times 6 \rightarrow$ positions 17–22

$74.5 \times 6 \rightarrow$ positions 23–28

$80.5 \times 5 \rightarrow$ positions 29–33

Finding Quartile-1(Q1), from the pseudo raw data given we can see that there are

4 values of 50.5

6 values of 56.5 $\rightarrow$ positions 5 to 10

6 values of 62.5 $\rightarrow$ positions 11 to 16

So, the 10th value = 56.5, and 11th value = 62.5

Thus, $Q1 = 56.5 + 0.25 \times (62.5 - 56.5) = 56.5 + 1.5 = 58.0$

(Q2) Median position $= (N+1)/2 = 41/2 = 20.5$ th value

We want the value at the 20.5th position in the ordered pseudo data.

Now from the pseudo raw data given we can see that both

20th value = 68.5 and 21st value = 68.5

Thus, $Q2 = (68.5 + 68.5)/2 = 68.5$

Since the 20.5th value lies between 68.5 and 68.5, the median is 68.5

Similarly, 30th value = 80.5, 31st value = 86.5

$Q3 = 80.5 + 0.75 \times (86.5 - 80.5) = 80.5 + 4.5 = 85.0$

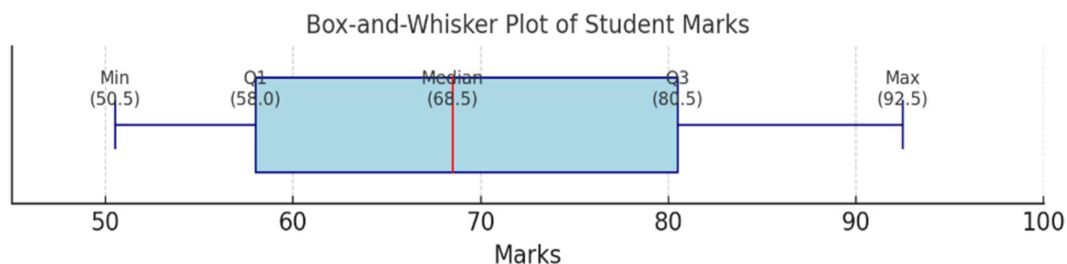
So, $Q1 \approx 58.0$, $Q2 \approx 68.5$ and $Q3 \approx 85.0$ in this case.

Step 4: Draw the Box-and-Whisker Plot

- Draw a horizontal number line that covers the range of your data (e.g., from 45 to 100).
- Mark the five-number summary values on the line.
- Draw a rectangular box from Q1 to Q3 .
- Draw a vertical line inside the box at the median .
- Draw whiskers (horizontal lines) from:
 - Min to Q1
 - Q3 to Max

Step 5: Label the Plot

- Label the five-number summary on the axis.
- Title your plot appropriately: “Box-and-Whisker Plot of Student Marks”



Here is the Box-and-Whisker Plot based on the student marks data. It visually displays the five-number summary:

Minimum: 50.5 ; Q1 (First Quartile): 58.0 ; Median (Q2): 68.5 ;

Q3 (Third Quartile): 80.5 ; Maximum: 92.5

This plot helps you understand the spread, centre, and symmetry of the data, as well as spot potential outliers (though none appear here).

In short, the box plot helps to:

- Visualize the spread and central tendency.
- Detects symmetry or skewness.
- Identify potential outliers (values far from the whiskers).
- Quickly compare multiple datasets if needed.

We learned that the Box plots provide a deeper look into data distribution and help identify variability and outliers.

✎ Check Your Progress 2

Q1. What are Percentiles and how are they used in outlier detection?

.....

Q2. Explain the Interquartile Range (IQR) method for detecting outliers.

.....

Q3. What is the relationship between Quartiles and Box Plots?

.....

Q4. Differentiate between Percentile-based and IQR-based outlier detection methods.

.....

Q5. Why are Box Plots considered effective for outlier detection and data visualization?

.....

Q6. What is a Box Plot? Explain its components.

.....

.....

Q7. Explain how outliers are detected using Box Plots.

.....

.....

Q8. Describe the steps to construct a Box Plot.

.....

.....

Q9. Interpret a Box Plot in terms of skewness and spread.

.....

.....

Q10. Why are Box Plots considered a reliable method for outlier detection?

6.6 DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. It is a density-based clustering algorithm used to identify groups of closely packed data points and, at the same time, detect points that do not belong to any dense region. These isolated points are treated as outliers or noise. DBSCAN is especially useful when the data contains irregularly shaped clusters and when outlier detection is an important objective.

Unlike partition-based methods such as K-means, DBSCAN does not require the number of clusters to be specified in advance. Instead, it uses two parameters, namely Eps and MinPts, to determine whether a region is dense enough to form a cluster. Because of this density-based logic, DBSCAN is widely used in anomaly detection, fraud analysis, geographic data analysis, image processing, and spatial mining.

The main idea of DBSCAN is simple: if a point lies in a sufficiently dense neighbourhood, then it belongs to a cluster. If a point lies alone or in a sparse region, then it is treated as noise or an outlier.

DBSCAN classifies points into three categories:

- **Core Point:** A point is called a **core point** if at least **MinPts** points lie within its neighbourhood of radius **Eps**.

Where,

- a. **Condition for Core Point is:** A point p is a core point if:
 $|N_{\epsilon}(p)| \geq MinPts$

where $|N_\epsilon(p)|$ means the number of points in the Eps-neighbourhood of p , usually including the point itself.

- b. **Eps:** Eps, written as ϵ , is the maximum distance within which points are considered neighbours.
- c. **Eps-Neighbourhood:** For a point p , the Eps-neighbourhood is defined as:

$$N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$$

where:

- D = dataset
- $\text{dist}(p, q)$ = distance between points p and q
This means all points within distance ϵ from point p are its neighbours.

- d. **MinPts:** MinPts is the minimum number of points required in the Eps-neighbourhood for a point to be considered a core point.

- **Border Point:** A point is called a **border point** if it lies within the Eps-neighbourhood of a core point, but it does not itself have enough neighbouring points to become a core point.
- **Noise Point / Outlier:** A point is called **noise** if it is neither a core point nor a border point. Such points do not belong to any cluster and are treated as outliers.

Important Note:

Practical Meaning of Parameters in Outlier Detection

- **If Eps is too small:** Many points may be marked as outliers because neighbourhoods become too small.
- **If Eps is too large:** Different clusters may merge, and true outliers may get absorbed into clusters.
- **If MinPts is too high:** Small but valid clusters may be treated as noise.
- **If MinPts is too low:** Noise points may incorrectly become clusters.

So, parameter tuning is essential.

The figure below describes Core point, Border Point and Noise points.

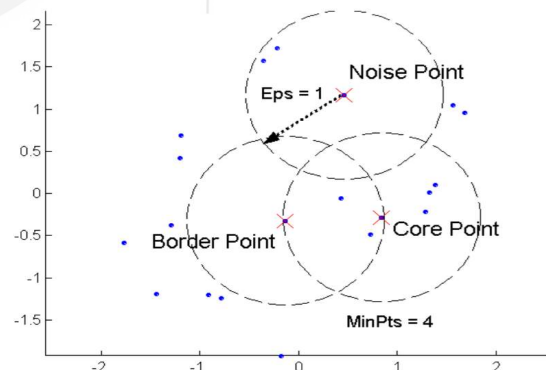


Figure – 6.1: Core point, Border Point and Noise points.

The most common distance used in DBSCAN is Euclidean distance:

$$dist(p, q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

For higher-dimensional data: $dist(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$

Key Concepts Used in DBSCAN are:

- **Directly Density-Reachable:** A point q is directly density-reachable from point p if: $q \in N_\epsilon(p)$, and p is a core point
- **Density-Reachable:** A point is density-reachable from another point if there exists a chain of directly density-reachable points connecting them.
- **Density-Connected:** Two points are density-connected if there exists some point from which both are density-reachable.

These ideas allow DBSCAN to form clusters of arbitrary shape.

DBSCAN Algorithm: The working of DBSCAN can be described in the following steps:

Step 1: Select an unvisited point from the dataset.

Step 2: Find all points in its Eps-neighbourhood.

Step 3: If the number of neighbours is at least MinPts, mark it as a core point and start a new cluster.

Step 4: Add all density-reachable points to this cluster. If any of these points are also core points, expand the cluster further.

Step 5: If a point does not satisfy the core-point condition and is not reachable from any core point, mark it as noise.

Step 6: Repeat the process until all points are visited.

Example: Apply DBSCAN algorithm on the following two-dimensional dataset:

$A(1,1), B(1,2), C(2,1), D(2,2), E(8,8), F(8,9), G(9,8), H(25,25)$ to identify the possible clusters, using Euclidean distance. Given $\epsilon = 1.5, MinPts = 3$

Solution:

Step 1: Check neighbourhood of A (1,1):

Therefore, Distances from A:

$$\begin{aligned} dist(A, B) &= \sqrt{(1-1)^2 + (2-1)^2} = 1 \\ dist(A, C) &= \sqrt{(2-1)^2 + (1-1)^2} = 1 \\ dist(A, D) &= \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2} \approx 1.41 \end{aligned}$$

All are within $\epsilon = 1.5$

So, $N_\epsilon(A) = \{A, B, C, D\}$ i.e. Number of points = 4, Since $|N_\epsilon(A)| = 4 \geq 3$ thus, A is a **core point**.

Step 2: Check neighbourhood of B (1,2)

Therefore, Distances from B:

$$dist(B, A) = 1$$

$$\text{dist}(B, C) = \sqrt{2} \approx 1.41$$

$$\text{dist}(B, D) = 1$$

So, $N_\epsilon(B) = \{A, B, C, D\}$ Again, 4 points. So, B is also a **core point**.

Similarly, C and D will also have enough neighbours within distance 1.5. Thus, points **A, B, C, D** form **Cluster 1**.

Step 3: Check neighbourhood of E(8,8)

Therefore, Distances from E:

$$\text{dist}(E, F) = 1 \quad \text{dist}(E, G) = 1$$

$$\text{dist}(E, H) = \sqrt{(25 - 8)^2 + (25 - 8)^2} = \sqrt{578} \approx 24.04$$

So, $N_\epsilon(E) = \{E, F, G\}$, Number of points = 3, Since $|N_\epsilon(E)| = 3 \geq 3$, E is a **core point**.

Now check F and G: $\text{dist}(F, G) = \sqrt{(9 - 8)^2 + (8 - 9)^2} = \sqrt{2} \approx 1.41$

So each of E, F, and G lies close to the others. Hence **E, F, G** form **Cluster 2**.

Step 4: Check neighbourhood of H (25,25)

Distances of H from all other points are much larger than 1.5. So, $N_\epsilon(H) = \{H\}$, Number of points = 1, Since $|N_\epsilon(H)| = 1 < 3$, H is **not** a core point. Also, H is not within the Eps-neighbourhood of any core point. Therefore, **H is a noise point**, that is, an **outlier**.

Final Result of Clusters

- Cluster 1: $\{D\}$
- Cluster 2: $\{G\}$
- Outlier / Noise H (25,25)

It can be interpreted from the Point of View of Outlier Detection that In this example, DBSCAN successfully identifies two dense groups of points and also detects one point, $H(25,25)$, as an outlier. This happens because H is far away from all dense regions and does not have enough neighboring points within radius $\epsilon = 1.5$.

This interpretation is important because of following reasons:

- points A, B, C, and D are in a dense local region, so they form one cluster
- points E, F, and G form another dense local region
- point H is isolated, so DBSCAN marks it as noise

Thus, from an outlier detection perspective, DBSCAN is very effective because it uses **local density** rather than global averages or fixed cluster assignments.

Summary Table

Point	Eps-Neighbourhood Size	Type	Cluster
A	4	Core	Cluster 1
B	4	Core	Cluster 1
C	4	Core	Cluster 1
D	4	Core	Cluster 1
E	3	Core	Cluster 2
F	3	Core/Border depending on counting	Cluster 2
G	3	Core/Border depending on counting	Cluster 2
H	1	Noise / Outlier	None

Advantages of DBSCAN

- No need to specify number of clusters in advance
- Detects outliers naturally
- Can find clusters of arbitrary shape
- Works well with noisy data
- Useful in spatial and anomaly-detection problems

Limitations of DBSCAN

- Choice of ϵ and MinPts is very important
- Struggles when cluster densities vary a lot
- Performance can reduce in high-dimensional data
- Sensitive to scale of variables, so normalization may be needed

We learned from the above example that *DBSCAN is particularly powerful for outlier detection* because it does not force every point into a cluster. Any point lying in a sparse region automatically becomes noise. This is a major advantage over methods like K-means, where every point is assigned to some cluster, even if it is abnormal.

From the outlier detection point of view, DBSCAN:

- identifies isolated points naturally
- works well when outliers are far from dense groups
- does not assume normal distribution
- can detect clusters of arbitrary shape
- is robust to noise

This section concludes with the understanding that DBSCAN is a powerful clustering and outlier detection algorithm based on the concept of density. It forms clusters where data points are closely packed and labels isolated points as noise. The method is especially valuable in real-world datasets where cluster shapes are irregular and outlier detection is important. In the numerical example, DBSCAN correctly formed two clusters and identified one isolated point as an outlier. This clearly demonstrates how the algorithm supports anomaly detection in a natural and intuitive way.

The final topic in this unit is the local Outlier factor; we are going to discuss it in next section 6.7 given below.

6.7 LOCAL OUTLIER FACTOR

The **Local Outlier Factor (LOF)** is a powerful density-based technique used in anomaly detection to identify observations that deviate significantly from their local neighbourhood. Unlike traditional global outlier detection methods, which evaluate a data point against the entire dataset, LOF focuses on the *local structure* of the data. This makes it particularly useful in real-world datasets where data is not uniformly distributed, and different regions exhibit varying densities.

The fundamental idea behind LOF is intuitive yet mathematically grounded, where a data point is considered an outlier if its **local density is substantially lower than that of its surrounding neighbours**. In other words, LOF does not simply label points far from the centre as anomalies; instead, it detects points that are isolated *relative to their immediate context*. For instance, a point in a sparse region may not be an outlier if its neighbours are equally sparse, whereas a point in a dense cluster that is relatively isolated from nearby points would be flagged as anomalous.

To understand how LOF works, it is essential to explore its key concepts and definitions. The method begins by defining the **k-nearest neighbours (k-NN)** of each data point. For a given point, these are the k closest points based on a chosen distance metric, typically Euclidean distance. This neighbourhood forms the basis for evaluating local density.

Next, LOF introduces the concept of **reachability distance**, which adjusts the raw distance between two points to account for the density of the neighbour. Specifically, the reachability distance between a point *A* and its neighbour *B* is defined as the maximum of two quantities: the actual distance between *A* and *B*, and the distance from *B* to its nearest neighbour. This adjustment ensures that points in dense regions do not appear artificially close to points in sparser regions, thereby stabilizing density comparisons.

Building on this, the **local reachability density (LRD)** of a point is computed. This represents how densely the point is surrounded by its neighbours and is defined as the inverse of the average reachability distance from the point to its k-nearest neighbours. A higher LRD indicates that the point lies in a dense region, while a lower LRD suggests a sparse neighbourhood.

Finally, the **Local Outlier Factor (LOF)** score is calculated by comparing the local reachability density of a point with the densities of its neighbours. Specifically, it is the average ratio of the LRDs of the point's neighbours to the LRD of the point itself. This ratio quantifies how isolated the point is relative to its surroundings. A LOF value close to 1 indicates that the point has a density similar to its neighbours and is therefore considered normal. A value significantly greater than 1 indicates that the point has a much lower density than its neighbours and is likely an outlier. Conversely, values less than 1 suggest that the point lies in a denser region than its neighbours.

So we learned that LOF provides a nuanced and context-aware approach to anomaly detection by evaluating the relative density of data points within their local neighbourhoods. This makes it highly effective in identifying subtle anomalies in complex datasets, especially where clusters of varying densities exist.

A brief introduction of the Key Concepts with their Formulas and Definitions are as follows:

(a) k-distance: For a point p , the k -distance is the distance to its **k-th nearest neighbor**.

For any data point p , the k -distance is defined as the distance between p and its nearest neighbor. This value determines the radius of the neighbourhood around the point. It is important to note that if multiple points are at the same distance, more than k neighbours may be included. The choice of k plays a crucial role: a small k captures very local structures, while a larger k provides a more global perspective. The k -distance of a point p is defined as the distance between p and its nearest neighbour in the dataset. Formally, it can be expressed as:

$$k\text{-distance}(p) = \text{dist}(p, o_k)$$

where:

- $\text{dist}(p, o_k)$ is the distance between point p and its k th nearest neighbour o_k ,
- o_k represents the k th nearest neighbour of p .

If we assume a standard distance metric such as Euclidean distance, then:

$$k\text{-distance}(p) = \sqrt{\sum_{i=1}^d (p_i - o_{k,i})^2}$$

where:

- d is the number of dimensions,
- p_i and $o_{k,i}$ are the coordinates of p and its k th neighbor in the i th dimension.

This formula identifies the radius that defines the local neighbourhood around point p , which is then used in subsequent LOF calculations.

(b) k-distance Neighbourhood

The k -distance neighbourhoods of a point p , denoted as $N_k(p)$, includes all points whose distance from p is less than or equal to the k -distance. It is formally expressed as:

$$N_k(p) = \{\text{dist}(p, q) \leq k\text{-distance}(p)\}$$

This includes all points within the k -distance.

This neighbourhood defines the local region around the point p that will be used for density estimation. It may contain more than k points due to ties in distance.

(c) Reachability Distance: The *reachability distance* is a modified distance measure that improves the robustness of density estimation. It is defined as:

$$\text{reach-dist}_k(p, q) = \max(k\text{-distance}(q), \text{dist}(p, q))$$

This definition ensures that distances smaller than the k -distance of the neighbor q are adjusted upward. The key purpose is to prevent extremely small distances from disproportionately increasing the density estimate, especially in very dense clusters. In effect, it smooths the influence of neighbours and stabilizes the computation. It prevents very small distances from dominating density calculations.

(d) Local Reachability Density (LRD)

The *local reachability density* of a point p , denoted as $LRD(p)$, measures how densely the point is surrounded by its neighbours. It is computed as the inverse of the average reachability distance:

$$LRD(p) = \frac{|N_k(p)|}{\sum_{q \in N_k(p)} reach - dist_k(p, q)}$$

- $|N_k(p)|$ = how many neighbours
- $\sum_{q \in N_k(p)} reach - dist_k(p, q)$ = total distance to neighbours (adjusted via reach-dist)

$|N_k(p)|$ means Total number of k -nearest neighbours of point p . It is NOT the value of q ; It is NOT the sum of neighbours. It is simply the count of neighbours

So, LRD is basically: “Number of neighbours per unit distance” i.e. **LRD** Represents how densely point p is surrounded.

A higher LRD value indicates that the point lies in a dense region (small average reachability distance), whereas a lower LRD suggests that the point is in a sparse region. Thus, LRD serves as a local density estimator.

(e) Local Outlier Factor (LOF)

The *Local Outlier Factor* of a point p , denoted as $LOF(p)$, quantifies how much the density of p deviates from the densities of its neighbours. It is defined as:

$$LOF(p) = \frac{\sum_{q \in N_k(p)} \frac{LRD(q)}{|N_k(p)|}}{LRD(p)}$$

This formula computes the average ratio of the local reachability densities of the neighbours to that of the point p . If p has a density similar to its neighbours, the ratio will be close to 1. If p has a significantly lower density, the LOF value will be greater than 1, indicating a potential outlier.

these concepts i.e. k -distance, k -distance Neighbourhood, Reachability distance, Local Reachability Density (LRD) and Local Outlier Factor (LOF) work together in a structured manner: the k -distance defines the neighbourhood, the reachability distance stabilizes pairwise distances, the LRD estimates local density, and the LOF score compares densities to identify anomalies. This layered formulation allows LOF to effectively detect local outliers even in datasets with varying density distributions.

Interpretation of LOF Values: After the calculation of Local Outlier Factor (LOF), the LOF values can be interpreted as follows:

LOF Value	Interpretation
≈ 1	Normal point
> 1	Possible outlier
$\gg 1$	Strong outlier

The Local Outlier Factor (LOF) algorithm follows a systematic sequence of steps to identify anomalous data points based on their local density. Understanding each step clearly helps you to grasp how the method works in practice.

Step 1: Choose parameter k (number of neighbours)

The first step is to select the value of k , which determines how many nearest neighbours will be considered for each data point. This parameter controls the “locality” of the analysis. A smaller k focuses on very close neighbours and captures fine-grained patterns, while a larger k provides a broader view of the data structure. Choosing an appropriate k is important because it directly affects the sensitivity of outlier detection.

Step 2: Compute k -distance for each point

For every data point p , calculate the distance to its k th nearest neighbour. This value, called the k -distance, defines the radius of the neighbourhood around p . It helps establish how far we need to look to include the nearest k points.

Step 3: Find k -nearest neighbours

Using the k -distance, identify the set of neighbours $N_k(p)$ for each point. These are all points whose distance from p is less than or equal to the k -distance. This step forms the local neighbourhood that will be used to estimate density.

Step 4: Compute reachability distance

Next, compute the *reachability distance* between each point p and its neighbours q . This adjusted distance takes into account both the actual distance and the density around the neighbour. It ensures that very small distances do not overly influence the density calculation, making the method more stable and reliable.

Step 5: Compute Local Reachability Density (LRD)

For each point p , calculate its *local reachability density (LRD)*. This is essentially the inverse of the average reachability distance between p and its neighbours. If the average distance is small, the density is high; if the average distance is large, the density is low. Thus, LRD quantifies how densely the point is surrounded by other points.

Step 6: Compute LOF score

Now, compute the *Local Outlier Factor (LOF)* for each point by comparing its LRD with the LRDs of its neighbours. This step measures how different the density of a point is from the densities of nearby points. A value close to 1 indicates that the point has a similar density to its neighbours, while a value significantly greater than 1 indicates a potential outlier.

Step 7: Identify outliers

Finally, analyse the LOF scores to identify anomalies. Points with high LOF values (typically greater than 1) are considered outliers because they are located in regions of lower density compared to their neighbours. The higher the LOF score, the stronger the indication that the point is an outlier.

Finally, the Steps of LOF Algorithm are:

1. Choose parameter **k (number of neighbours)**
2. Compute **k -distance** for each point
3. Find **k -nearest neighbours**
4. Compute **reachability distance**

5. Compute **LRD for each point**
6. Compute **LOF score**
7. Identify points with **high LOF as outliers**

The LOF algorithm works by gradually building from neighbourhood identification to density estimation and finally to density comparison. This step-by-step approach allows it to effectively detect local anomalies even in complex datasets with varying densities, making it a powerful and widely used technique in data analysis.

Example: Determine the Outliers by applying Local Outlier Factor (LOF) algorithm on the 1-Dimensional Dataset: 1, 2, 2, 3, 3, 4, 10 Given $k = 2$

Solution: Let us apply the Local Outlier Factor (LOF) algorithm step-by-step on the given 1D dataset:

Dataset: {1,2,2,3,3,4,10}, with $k=2$

Step 1: Sort the Data

It is observed that the given data is already sorted: 1,2,2,3,3,4,10

Step 2: Compute k-distance and k-nearest neighbours

For $k=2$, we find the 2 nearest neighbours for each point:

Point	2-Nearest Neighbours	k-distance
1	2, 2	1
2(1)	1, 2	1
2(2)	2, 3	1
3(1)	2, 3	1
3(2)	3, 4	1
4	3, 3	1
10	4, 3	6

Let's understand how the 2-Nearest Neighbours are identified

Given Dataset (1D): {1,2,2,3,3,4,10}, and $k=2$

Since the data is 1D, the distance between two points is simply: $\text{dist}(a, b) = |a - b|$

Now, to Compute k-distance and k-nearest neighbours for each point follow the steps:

1. Compute distance to all other points
2. Sort distances in ascending order
3. Pick the 2 nearest neighbours ($k = 2$)
4. The distance to the 2nd nearest neighbor = k-distance

1. For point = 1 in the Given Dataset (1D): {1,2,2,3,3,4,10}, and $k=2$

Distances: $|1-2|=1, |1-2|=1, |1-3|=2, |1-3|=2, |1-4|=3, |1-10|=9$

Sorted distances are: 1, 1, 2, 2, 3, 9

Among the Sorted Distances analyse the first two entries i.e. $|1-2|=1, |1-2|=1$; the 2 nearest neighbours are 2 & 2, and the k-distance is 1. So, 2 nearest neighbours are (2,2) and the k-distance = 1

Similarly, for other points, as shown below:

2. For point = 2 (first occurrence)

Distances: $|2-1|=1$, $|2-2|=0$, $|2-3|=1$, $|2-3|=1$, $|2-4|=2$, $|2-10|=8$

Sorted: 0, 1, 1, 1, 2, 8 ; with 2 nearest neighbours $\rightarrow 2, 1$ (including duplicate 2) and k-distance=1

Next,

3. For point = 2 (second occurrence)

Distances: $|2-2|=0$, $|2-3|=1$, $|2-3|=1$, $|2-1|=1$, $|2-4|=2$, $|2-10|=8$

Sorted: 0, 1, 1, 1, 2, 8; with 2 nearest neighbours $\rightarrow 2, 3$ and k-distance = 1

4. For point = 3 (first occurrence)

Distances: $|3-2|=1$, $|3-2|=1$, $|3-3|=0$, $|3-4|=1$, $|3-1|=2$, $|3-10|=7$

Sorted: 0, 1, 1, 1, 2, 7 with 2 nearest neighbours $\rightarrow 3, 2$ and k-distance = 1

5. For point = 3 (second occurrence)

Distances: $|3-3|=0$, $|3-4|=1$, $|3-2|=1$, $|3-2|=1$, $|3-1|=2$, $|3-10|=7$

Sorted: 0, 1, 1, 1, 2, 7 with 2 nearest neighbours $\rightarrow 3, 4$ and k-distance = 1

6. For point = 4

Distances: $|4-3|=1$, $|4-3|=1$, $|4-2|=2$, $|4-2|=2$, $|4-1|=3$, $|4-10|=6$

Sorted: 1, 1, 2, 2, 3, 6 with 2 nearest neighbours $\rightarrow 3, 3$ and k-distance = 1

7. For point = 10

Distances: $|10-4|=6$, $|10-3|=7$, $|10-3|=7$, $|10-2|=8$, $|10-2|=8$, $|10-1|=9$

Sorted: 6, 7, 7, 8, 8, 9 with 2 nearest neighbours $\rightarrow 4, 3$ and k-distance = 7 (distance to 2nd nearest neighbor)

Final Summary Table

<u>Point</u>	<u>2-Nearest Neighbours</u>	<u>k-distance</u>
1	2, 2	1
2	2, 1	1
2	2, 3	1
3	3, 2	1
3	3, 4	1
4	3, 3	1
10	4, 3	7

Important Insights: From the above table it can be interpreted that all cluster points have small k-distance (≈ 1) but the Point 10 has a large k-distance (7) and this is the early indication of outlier.

Now, we are done with **Step 2: Compute k-distance and k-nearest neighbours; of the Local Outlier Factor (LOF) algorithm**. The next step is to Determine the Reachability Distance.

Step 3: Compute Reachability Distance:

$$reach - dist_k(p, q) = \max(k - distance(q), dist(p, q));$$

where $dist(p, q) = |p - q|$

Now compute for key points:

- For point 1 (neighbours: 2, 2): reach-dist = $\max(1, 1) = 1$ for both neighbours $\rightarrow$ avg = 1
- For point 2 / 3 / 4 (cluster points): distances are small $\rightarrow$ reach-dist mostly = 1
- For point 10 (neighbours: 4, 3):

- $\text{reach-dist}(10,4) = \max(1,6) = 6$
- $\text{reach-dist}(10,3) = \max(1,7) = 7$

Now, the next step of the Local Outlier Factor (LOF) algorithm is to determine the Local Reachability Density (LRD)

Step 4: Compute Local Reachability Density (LRD)

$$LRD(p) = \frac{|N_k(p)|}{\sum_{q \in N_k(p)} \text{reach-dist}_k(p, q)}$$

- $|N_k(p)|$ = how many neighbours
- $\sum_{q \in N_k(p)} \text{reach-dist}_k(p, q)$ = total distance to neighbours (adjusted via reach-dist)

$|N_k(p)|$ means Total number of k-nearest neighbours of point p. It is NOT the value of q; It is NOT the sum of neighbours. It is simply the count of neighbours

So, LRD is basically: “Number of neighbours per unit distance” i.e. **LRD** Represents how densely point p is surrounded.

A higher LRD value indicates that the point lies in a dense region (small average reachability distance), whereas a lower LRD suggests that the point is in a sparse region. Thus, LRD serves as a local density estimator.

Refer to Summary table derived above under Step-2, as per the data available for point $p = 1$

For $p=1$, $k=2$ Identified Nearest neighbours of 1 are 2 and 2. So: $|N_k(1)| = \{2,2\}$. Thus, $|N_k(1)| = 2$ Because there are 2 neighbors. Now, q represents each individual neighbour and It is used to calculate the denominator part of LRD i.e. $\sum_{q \in N_k(p)} \text{reach-dist}_k(p, q)$. Take each neighbour q , Compute reach-distance and add them.

For $p=1$ we got $|N_k(1)| = 2$ and $\text{reach-dist}(1,2) + \text{reach-dist}(1,2) = 1 + 1 = 2$;
Thus, $LRD(1) = 2 / (1+1) = 2/2 = 1$

Similarly calculate LRD For cluster points (1,2,2,3,3,4): $LRD \approx 2 / (1+1) = 1$

For point $p = 10$ (**Outlier candidate**) the Neighbours are 4, 3.

- For $q = 4$: $d(10,4) = 6$ and $k\text{-distance}(4) = 1$. Also, $\text{reach-dist} = \max(1,6) = 6$
- For $q = 3$: $d(10,3) = 7$ and $k\text{-distance}(3) = 1$. Also, $\text{reach-dist} = \max(1,7) = 7$

Therefore, $LRD(10) = 2 / (6+7) = 2 / 13 \approx 0.154$

Now the final step is to determine Local Outlier Factor (LOF)

Step 5: Compute Local Outlier Factor (LOF)

$$LOF(p) = \frac{\sum_{q \in N_k(p)} \frac{LRD(q)}{|N_k(p)|}}{|N_k(p)|}$$

LOF for point $p = 1$; Neighbors $\rightarrow 2, 2$; Apply LOF formula: $LOF(1) = \{(1/1) + (1/1)\} / 2 = 1$

So, $LOF(1) = 1$ i.e. it is a Normal point

Similarly, LOF for point $p = 4$; Neighbors $\rightarrow 3, 3$; $LOF(4) = \{(1/1) + (1/1)\} / 2 = 1$

So, $LOF(4) = 1$ is also a Normal point

Now, LOF for point $p = 10$; Neighbors $\rightarrow 4$ and 3 ; $LRD(4) = 1$; $LRD(3) = 1$; and $LRD(10) = 0.154$

Therefore, $LOF(10) = \{(1/0.154) + (1/0.154)\} / 2 = 6.49$

- For cluster points: 1,2,2,3,3,4 $LOF=1$
- For point 10: Neighbours have $LRD(4) = 1$; $LRD(3) = 1$; and $LRD(10) = 0.154$
Thus, $LOF(10) = (1/0.154 + 1/0.154) / 2 = (6.49 + 6.49) / 2 \approx 6.49$

After the calculation of Local Outlier Factor (LOF), the LOF values can be interpreted as follows:

LOF Value	Interpretation
≈ 1	Normal point
> 1	Possible outlier
$\gg 1$	Strong outlier

Thus, it can be interpreted from the results that Points 1,2,2,3,3,4 have $LOF = 1$ and they are the Normal points in the dataset. However, LOF for Point 10 is 6.49 hence identified as a Strong Outlier

Final Conclusion from the above results can be expressed as: The LOF algorithm clearly identifies 10 as an outlier because its local density is much lower compared to its neighbours. All other points form a dense cluster and are considered normal.

Finally, we understood that LOF is a density-based outlier detection method that compares the local density of a point with that of its neighbours. If a point has significantly lower density than its surroundings, it is identified as an outlier. In the given example, point 10 has a very high LOF value, confirming it as a strong outlier. Thus, LOF is highly effective for detecting anomalies in datasets with varying density patterns.

We learned the working of Local Outlier Factor (LOF) algorithm for Outlier detection. Now, we are going to conclude this section with a brief on the Advantages and Limitations of Local Outlier Factor (LOF) Algorithm

Advantages of Local Outlier Factor (LOF)

1. Captures local anomalies

LOF focuses on the *local neighbourhood* of each data point rather than the entire dataset. This means it can identify points that are unusual in their immediate surroundings, even if they may not appear extreme globally. This makes it very effective for detecting subtle outliers.

2. Works well in multi-density datasets

In real-world data, different regions may have different densities. Traditional methods may fail in such cases, but LOF adapts by comparing a point only with its nearby neighbors. This allows it to correctly identify outliers in both dense and sparse regions.

3. More flexible than distance-based methods

Unlike simple distance-based techniques that rely only on how far a point is from others, LOF

considers *relative density*. This added flexibility helps in identifying complex patterns where distance alone is not sufficient.

4. **Useful in real-world applications**

LOF is widely used in domains like fraud detection, intrusion detection, and medical diagnosis because it can detect unusual behavior patterns, rare events, or abnormal observations effectively.

Limitations of Local Outlier Factor (LOF)

1. **Sensitive to parameter k**

The choice of the number of neighbors (k) significantly affects the results. A very small or very large k can lead to incorrect detection of outliers, so selecting an appropriate value is important and sometimes challenging.

2. **Computationally expensive for large datasets**

LOF requires computing distances between points and their neighbors, which becomes time-consuming as the dataset size increases. This makes it less efficient for very large datasets without optimization.

3. **Performance depends on distance metric**

The results of LOF depend heavily on how distance is measured (e.g., Euclidean distance). If the chosen distance metric does not properly reflect similarity in the data, the results may be misleading.

4. **Struggles in high-dimensional data**

In very high-dimensional spaces, distances between points tend to become similar (a problem known as the *curse of dimensionality*). This reduces the effectiveness of LOF, as it becomes harder to distinguish between normal points and outliers.

Check Your Progress 3

Q1. What is DBSCAN? Explain its role in outlier detection.

.....

.....

Q2. Define the key parameters of DBSCAN.

.....

.....

Q3. Differentiate between core, border, and noise points in DBSCAN.

.....

.....

Q4. What are the advantages of DBSCAN in outlier detection?

.....

.....

Q5. What is the Local Outlier Factor (LOF)?

.....

.....

Q6. Explain how LOF detects outliers.

.....

.....

Q7. Differentiate between DBSCAN and LOF.

.....

.....

Q8. Why are density-based methods preferred for complex datasets?

.....

.....

Q9. What are the limitations of DBSCAN?

.....

.....

Q10. Explain the importance of LOF in real-world applications.

6.8 SUMMARY

Outlier Detection is an essential aspect of data analysis that focuses on identifying unusual or extreme observations that significantly differ from the rest of the dataset. These outliers may arise due to errors, variability, or rare events, and their identification is crucial for improving data quality and model accuracy. One of the simplest approaches is the Standard Deviation method, where data points lying far from the mean (typically beyond ± 2 or ± 3 standard deviations) are considered outliers. Similarly, the Z-score method standardizes values and identifies outliers based on how many standard deviations a data point is away from the mean. Percentiles provide another intuitive approach by flagging values that fall in extreme ranges (e.g., below the 5th percentile or above the 95th percentile). These statistical techniques are widely used due to their simplicity and effectiveness in normally distributed data.

Graphical techniques such as Box-plots also play an important role in outlier detection by visually representing data distribution through quartiles. Values lying beyond the whiskers (usually 1.5 times the interquartile range) are considered outliers. Moving beyond basic methods, advanced techniques such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identify outliers as points that do not belong to any dense cluster, making it effective for detecting anomalies in complex datasets. Similarly, the Local Outlier Factor (LOF) method evaluates the local density of a data point compared to its neighbors, identifying points that have significantly lower density as outliers.

Overall, the methods covered in this Unit, provides a comprehensive toolkit for detecting anomalies in both simple and complex datasets. While statistical methods are effective for structured and normally distributed data, advanced techniques like DBSCAN and LOF are more suitable for high-dimensional and non-linear data. By combining these approaches, analysts can ensure accurate detection of outliers, leading to more reliable data analysis and better decision-making.

6.9 ANSWERS

✦ Check Your Progress 1

Q1. What are outliers? Explain their significance in predictive data analysis.

Solution: Refer to Section 6.0

Q2. Explain the Standard Deviation method for detecting outliers.

Solution: Refer to Section 6.2

Q3. Define Z-score and explain its role in outlier detection.

Solution: Refer to Section 6.3

Q4. Differentiate between Standard Deviation method and Z-score method.

Solution: Refer to Section 6.2 & 6.3

Q5. Explain the relationship between Standard Deviation and Z-score in outlier detection.

Solution: Refer to Section 6.2 & 6.3

✦ Check Your Progress 2

Q1. What are Percentiles and how are they used in outlier detection?

Solution: Refer to Section 6.4

Q2. Explain the Interquartile Range (IQR) method for detecting outliers.

Solution: Refer to Section 6.4

Q3. What is the relationship between Quartiles and Box Plots?

Solution: Refer to Section 6.4

Q4. Differentiate between Percentile-based and IQR-based outlier detection methods.

Solution: Refer to Section 6.4

Q5. Why are Box Plots considered effective for outlier detection and data visualization?

Solution: Refer to Section 6.5

Q6. What is a Box Plot? Explain its components.

Solution: Refer to Section 6.5

Q7. Explain how outliers are detected using Box Plots.

Solution: Refer to Section 6.5

Q8. Describe the steps to construct a Box Plot.

Solution: Refer to Section 6.5

Q9. Interpret a Box Plot in terms of skewness and spread.

Solution: Refer to Section 6.5

Q10. Why are Box Plots considered a reliable method for outlier detection?

Solution: Refer to Section 6.5

☛ Check Your Progress 3

Q1. What is DBSCAN? Explain its role in outlier detection.

Solution: Refer to Section 6.6

Q2. Define the key parameters of DBSCAN.

Solution: Refer to Section 6.6

Q3. Differentiate between core, border, and noise points in DBSCAN.

Solution: Refer to Section 6.6

Q4. What are the advantages of DBSCAN in outlier detection?

Solution: Refer to Section 6.6

Q5. What is the Local Outlier Factor (LOF)?

Solution: Refer to Section 6.7

Q6. Explain how LOF detects outliers.

Solution: Refer to Section 6.7

Q7. Differentiate between DBSCAN and LOF.

Solution: Refer to Section 6.6 & 6.7

Q8. Why are density-based methods preferred for complex datasets?

Solution: Refer to Section 6.6

Q9. What are the limitations of DBSCAN?

Solution: Refer to Section 6.6

Q10. Explain the importance of LOF in real-world applications.

Solution: Refer to Section 6.7

6.10 REFERENCES/FURTHER READINGS

1. *Data Mining: Concepts and Techniques* by Jiawei Han, Micheline Kamber, and Jian Pei, published by Morgan Kaufmann (3rd Edition, 2011, ISBN: 978-0123814791)
2. *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, published by Springer (2nd Edition, 2021, ISBN: 978-1071614174).
3. *Practical Statistics for Data Scientists* by Peter Bruce, Andrew Bruce, and Peter Gedeck, published by O'Reilly Media (2nd Edition, 2020, ISBN: 978-1492072942).
4. *Business Analytics: Data Analysis and Decision Making* by S. Christian Albright and Wayne L. Winston, published by Cengage Learning (7th Edition, 2020, ISBN: 978-0357131787).
5. *Introduction to Operations Research* by Frederick S. Hillier and Gerald J. Lieberman, published by McGraw-Hill Education (10th Edition, 2014, ISBN: 978-0073523453).
6. *Decision Analysis for Management Judgment* by Paul Goodwin and George Wright, published by Wiley (5th Edition, 2014, ISBN: 978-1119974017).
7. *Handbook of Statistical Analysis and Data Mining Applications* by Robert Nisbet, John Elder, and Gary Miner, published by Elsevier (2nd Edition, 2017, ISBN: 978-0124166455).
8. *Data Mining and Analysis: Fundamental Concepts and Algorithms* by Mohammed J. Zaki and Wagner Meira Jr., published by Cambridge University Press (1st Edition, 2014, ISBN: 978-0521766333).

9. *Core Concepts in Data Analysis: Summarization, Correlation and Visualization* by Boris Mirkin, published by Springer (1st Edition, 2011, ISBN: 978-0857292872).
10. *Applied Predictive Modeling*, Max Kuhn and Kjell Johnson, 1st Edition, Springer, 2013.
11. *An Introduction to Statistical Learning with Applications in R*, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, 1st Edition, Springer, 2013.
12. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Trevor Hastie, Robert Tibshirani and Jerome Friedman, 2nd Edition, Springer, 2009.
13. *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, published by Springer (2nd Edition, 2021, ISBN: 978-1071614174).
14. *Time Series Analysis: Forecasting and Control* by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung, published by Wiley (5th Edition, 2015, ISBN: 978-1118675021)



ignou
THE PEOPLE'S
UNIVERSITY