

## Assignment

Q1. What is anomaly detection and what is its purpose?

Ans: Anomaly detection, also known as outlier detection, is a technique used in machine learning and data analysis to identify instances or patterns in data that do not conform to expected behavior or a defined norm. The purpose of anomaly detection is to identify unusual observations, events, or patterns that deviate significantly from the majority of the data. Anomalies are often indicative of errors, irregularities, or events that are not in line with the expected or normal behavior of the system.

## Key Aspects of Anomaly Detection:

Unsupervised Learning:

- Anomaly detection is often performed in an unsupervised learning setting, where the algorithm learns from the majority of the data without explicit labels for normal and anomalous instances.

Noisy Data Identification:

- Anomaly detection helps identify instances of noisy or erroneous data, which can be crucial for maintaining the integrity of datasets.

Fraud Detection:

- In finance and e-commerce, anomaly detection is commonly used to identify fraudulent transactions or activities that deviate from regular spending or usage patterns.

Network Intrusion Detection:

- Anomaly detection is applied to identify unusual patterns in network traffic, helping detect potential security breaches or cyber attacks.

Quality Control:

- In manufacturing and industrial processes, anomaly detection is used for quality control to identify defective products or deviations from standard operating conditions.

Health Monitoring:

- Anomaly detection is employed in healthcare for monitoring patient health data, detecting abnormal physiological readings, or identifying potential health issues.

Predictive Maintenance:

- In equipment and machinery maintenance, anomaly detection can predict potential failures by identifying abnormal behavior in sensor data, allowing for proactive maintenance.

System Monitoring:

- Anomaly detection is used in monitoring the performance of systems, such as servers or infrastructure, to identify unusual behavior that may indicate faults or performance issues.

## Techniques for Anomaly Detection:

#### Statistical Methods:

- Statistical techniques, such as z-score, identify anomalies based on deviations from statistical measures like mean or standard deviation.

#### Machine Learning Algorithms:

- Machine learning models, including clustering methods (e.g., k-means), density estimation methods (e.g., Gaussian Mixture Models), and isolation forest, can be trained to identify anomalies.

#### Deep Learning:

- Deep learning techniques, such as autoencoders and neural networks, are used to learn complex patterns and identify deviations from normal data.

#### Ensemble Methods:

- Combining multiple anomaly detection models into an ensemble can improve robustness and accuracy in detecting anomalies.

#### Time Series Analysis:

- Anomaly detection in time series data involves identifying deviations from expected temporal patterns, often using techniques like seasonality analysis or forecasting.

## Importance of Anomaly Detection:

#### Early Problem Detection:

- Anomaly detection allows for the early identification of issues, errors, or abnormalities before they escalate into larger problems.

#### Cost Reduction:

- Early detection of anomalies can lead to cost savings by preventing fraud, minimizing equipment downtime, or avoiding potential losses.

#### Improved Security:

- In cybersecurity, anomaly detection enhances security measures by identifying unusual activities that may indicate cyber threats or breaches.

#### Enhanced Data Quality:

- Anomaly detection contributes to maintaining high data quality by identifying and handling noisy or erroneous data points.

#### Proactive Maintenance:

- Anomaly detection in predictive maintenance enables organizations to perform maintenance tasks proactively, reducing unplanned downtime and extending equipment lifespan.

In summary, anomaly detection is a valuable technique with diverse applications across various domains. Its primary purpose is to identify deviations from normal behavior or patterns in data,

allowing organizations to detect and address issues early, enhance security, and improve overall data quality and reliability.

Q2. What are the key challenges in anomaly detection?

Ans: Anomaly detection poses several challenges, and addressing these challenges is crucial for the successful implementation of effective anomaly detection systems. Here are key challenges in anomaly detection:

Lack of Labeled Anomaly Data:

- Challenge: Anomaly detection often requires labeled data, specifically instances of anomalies, for model training. However, obtaining a sufficient amount of labeled anomaly data can be challenging as anomalies are typically rare events.
- Mitigation: Techniques such as semi-supervised learning, where only normal data is labeled during training, or unsupervised learning, which doesn't require labeled anomalies, are commonly used.

Class Imbalance:

- Challenge: Anomalies are often rare compared to normal instances, leading to class imbalance. Traditional machine learning models may struggle to learn effectively in such imbalanced settings.
- Mitigation: Techniques like oversampling the minority class, using ensemble methods, or employing specialized algorithms designed to handle imbalanced data can be applied.

Changing Data Patterns:

- Challenge: Anomalies may occur in various forms, and the patterns of normal and anomalous behavior may change over time. Static models may struggle to adapt to evolving data patterns.
- Mitigation: Continuous monitoring and model updating are essential. Techniques like online learning or periodic model retraining can help adapt to changes in data patterns.

Ambiguity in Anomaly Definition:

- Challenge: Defining what constitutes an anomaly can be subjective and domain-specific. What may be considered normal behavior in one context could be an anomaly in another.
- Mitigation: Collaborative efforts involving domain experts and stakeholders are crucial to clearly define and refine what constitutes anomalous behavior in a specific context.

High-Dimensional Data:

- Challenge: In high-dimensional spaces, distinguishing between normal and anomalous instances becomes more challenging. The curse of dimensionality can lead to increased false positives.

- Mitigation: Feature selection, dimensionality reduction techniques, or using specialized algorithms designed for high-dimensional data can help mitigate this challenge.

#### Noise in Data:

- Challenge: Noise or irrelevant features in the data can make it difficult to distinguish true anomalies from random fluctuations.
- Mitigation: Preprocessing steps, such as feature engineering and data cleaning, can help reduce noise. Robust algorithms that are less sensitive to noise may also be employed.

#### Concept Drift:

- Challenge: Changes in the underlying data distribution over time, known as concept drift, can impact the model's performance as it may become less accurate in detecting anomalies.
- Mitigation: Regular monitoring for concept drift, adaptive learning techniques, and periodic model retraining are approaches to address this challenge.

#### Interpretability:

- Challenge: Some anomaly detection models, particularly those based on complex algorithms like deep learning, may lack interpretability, making it challenging to understand why a particular instance is flagged as anomalous.
- Mitigation: Choosing interpretable models, incorporating model-agnostic interpretability techniques, and involving domain experts in the interpretation process can help address this challenge.

#### Scalability:

- Challenge: As data volumes increase, scalability becomes a concern. Some anomaly detection algorithms may struggle to handle large datasets efficiently.
- Mitigation: Choosing scalable algorithms, employing distributed computing frameworks, or parallelizing computations can help handle large-scale data.

#### Trade-Offs between False Positives and False Negatives:

- Challenge: Anomaly detection models often face a trade-off between minimizing false positives and false negatives. A model may be overly conservative, leading to missed anomalies, or overly permissive, resulting in more false positives.
- Mitigation: Adjusting the model's threshold, tuning hyperparameters, and choosing evaluation metrics that balance false positives and false negatives based on the application's requirements are essential.

Addressing these challenges requires a combination of domain knowledge, careful model selection, and ongoing monitoring and adaptation. No single approach fits all scenarios, and tailoring solutions to specific contexts is crucial for effective anomaly detection.

Q3. How does unsupervised anomaly detection differ from supervised anomaly detection?

Ans: Unsupervised anomaly detection and supervised anomaly detection are two distinct approaches to identifying anomalies in data, each with its own characteristics and applications. Here's a comparison of unsupervised and supervised anomaly detection:

## Unsupervised Anomaly Detection:

Data Labeling:

- Unsupervised: In unsupervised anomaly detection, the algorithm is trained on a dataset without explicit labels indicating which instances are normal or anomalous. The algorithm learns the inherent patterns of normal behavior without specific guidance on anomalies.

Anomaly Definition:

- Unsupervised: The algorithm must identify anomalies based on deviations from the learned normal behavior. The definition of anomalies is often subjective and may vary depending on the characteristics of the data.

Applications:

- Unsupervised: Well-suited for scenarios where labeled anomaly data is scarce or unavailable. It is commonly used in situations where anomalies are rare and varied, making it challenging to define and label them explicitly.

Examples:

- Unsupervised: Clustering-based methods (e.g., k-means, DBSCAN), density-based methods (e.g., Isolation Forest), and reconstruction-based methods (e.g., autoencoders) are common in unsupervised anomaly detection.

Training Process:

- Unsupervised: The algorithm learns to identify normal patterns without being provided with explicit information about anomalies during training. It aims to find deviations from the learned normal patterns during testing or deployment.

Scenarios:

- Unsupervised: Suitable for scenarios where the characteristics of normal behavior are well-defined, but anomalies may take diverse and unforeseen forms.

## Supervised Anomaly Detection:

Data Labeling:

- Supervised: In supervised anomaly detection, the algorithm is trained on a dataset with explicit labels indicating which instances are normal and which are anomalous. The training process involves learning the characteristics of both normal and anomalous behavior.

Anomaly Definition:

- Supervised: Anomalies are explicitly labeled during the training phase. The algorithm learns to distinguish between normal and anomalous instances based on the provided labels.

#### Applications:

- Supervised: Useful in scenarios where labeled anomaly data is available or can be generated. It is often employed when anomalies are well-defined and can be identified with precision.

#### Examples:

- Supervised: Support Vector Machines (SVM), Random Forest, and other classification algorithms are commonly used in supervised anomaly detection. Ensemble methods and deep learning approaches may also be applied.

#### Training Process:

- Supervised: The algorithm learns from a labeled dataset, and the model's objective is to correctly classify instances into normal or anomalous categories during training. The model is then evaluated on its ability to generalize to new, unseen data.

#### Scenarios:

- Supervised: Suitable for scenarios where anomalies are well-understood, and labeled data is available for training. It is effective when the characteristics of anomalies are distinct from normal behavior.

## Comparison:

- Data Availability:
  - Unsupervised: Does not require labeled anomaly data, making it applicable in scenarios where obtaining labeled data is challenging.
  - Supervised: Requires labeled anomaly data for training, which may not always be available.
- Flexibility:
  - Unsupervised: More flexible in handling diverse and unforeseen forms of anomalies.
  - Supervised: Requires predefined and labeled anomalies, limiting flexibility in handling novel anomaly types.
- Training Complexity:
  - Unsupervised: Generally simpler in terms of data preparation since labeled anomaly data is not required.
  - Supervised: Requires careful labeling of anomalies, and model training involves a supervised learning process.
- Application:
  - Unsupervised: Well-suited for exploratory analysis and scenarios where anomalies are not well-defined or expected to evolve over time.
  - Supervised: Effective when anomalies are well-understood and labeled data is available, suitable for scenarios with more explicit anomaly definitions.

Both approaches have their merits and drawbacks, and the choice between unsupervised and supervised anomaly detection depends on the specific characteristics of the data and the availability of labeled anomaly information. In practice, a combination of both approaches or semi-supervised methods may also be employed to leverage the advantages of each.

Q4. What are the main categories of anomaly detection algorithms?

Ans: Anomaly detection algorithms can be categorized into several main types based on their underlying principles and methodologies. Here are the main categories of anomaly detection algorithms:

## **1. Statistical Methods:**

- Description: Statistical methods assume that normal data points conform to a specific statistical distribution. Anomalies are identified as instances that deviate significantly from the expected distribution.
- Examples:
  - Z-Score-based methods
  - Gaussian Distribution-based methods
  - Tukey's Fences

## **2. Machine Learning Algorithms:**

- Description: Machine learning-based approaches use algorithms to learn patterns in the data and identify anomalies based on deviations from learned normal behavior. These algorithms can be applied in both supervised and unsupervised settings.
- Examples:
  - Clustering methods (e.g., k-means, DBSCAN)
  - Density-based methods (e.g., Isolation Forest)
  - Support Vector Machines (SVM)
  - Random Forest
  - Neural Networks (especially autoencoders for reconstruction-based methods)

## **3. Proximity-Based Methods:**

- Description: Proximity-based methods identify anomalies based on the proximity or similarity of data points. Instances that are far from their neighbors or have unusual distances are considered anomalies.
- Examples:
  - Nearest Neighbor-based methods

- Local Outlier Factor (LOF)

## **4. Reconstruction-Based Methods:**

- Description: Reconstruction-based methods involve training models to reconstruct normal data. Anomalies are identified by assessing the reconstruction error, i.e., how well the model reconstructs a given instance.
- Examples:
  - Autoencoders
  - Principal Component Analysis (PCA)

## **5. Ensemble Methods:**

- Description: Ensemble methods combine multiple models to improve overall anomaly detection performance. They may involve aggregating results from individual models or training diverse models to capture different aspects of normal behavior.
- Examples:
  - Isolation Forest
  - One-Class SVM

## **6. Time Series Analysis:**

- Description: Time series anomaly detection focuses on identifying deviations from expected temporal patterns. Methods consider the historical behavior of time series data to detect anomalies.
- Examples:
  - Seasonal-Trend decomposition using LOESS (STL)
  - Exponential Smoothing Methods
  - Change-point detection algorithms

## **7. Domain-Specific Methods:**

- Description: Some anomaly detection methods are designed for specific domains or types of data. These methods leverage domain knowledge to tailor anomaly detection to the unique characteristics of the data.
- Examples:
  - Domain-specific rules or heuristics
  - Industry-specific anomaly detection methods (e.g., fraud detection in finance)

## **8. Density-Based Methods:**



- Description: Density-based methods identify anomalies based on the density of data points. Anomalies are instances that have significantly lower or higher density compared to their neighbors.
- Examples:
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - Local Outlier Factor (LOF)

## 9. Graph-Based Methods:

- Description: Graph-based methods model data as graphs and identify anomalies based on structural irregularities or deviations from expected graph properties.
- Examples:
  - Connectivity-based approaches
  - PageRank-based methods

## 10. Spectral Methods:

- Description: Spectral methods involve analyzing the spectral properties of data matrices to identify anomalies. These methods are often applied to high-dimensional data.
- Examples:
  - Spectral Clustering
  - Subspace methods

These categories are not mutually exclusive, and some anomaly detection algorithms may incorporate elements from multiple categories. The choice of an algorithm depends on the specific characteristics of the data, the nature of anomalies, and the available resources for model training and deployment.

Q5. What are the main assumptions made by distance-based anomaly detection methods?

Ans: Distance-based anomaly detection methods rely on the assumption that normal instances in a dataset exhibit similar patterns and tend to be close to each other in the feature space. Anomalies, on the other hand, are expected to deviate significantly from the majority of normal instances and may be isolated or distant. The main assumptions made by distance-based anomaly detection methods include:

### 1. Assumption of Density:

- Description: Normal instances are assumed to form dense clusters in the feature space. Anomalies, being rare or unusual, are expected to have lower density, potentially leading to greater distances from their neighbors.

- Implication: Density-based methods, such as local outlier factor (LOF) and DBSCAN, leverage the assumption that anomalies have lower local densities compared to normal instances.

## **2. Local Homogeneity:**

- Description: The assumption of local homogeneity implies that normal instances are expected to exhibit local similarity or homogeneity within clusters. Anomalies may disrupt this homogeneity, leading to increased distances from their local neighbors.
- Implication: Proximity-based methods, like nearest neighbor-based algorithms, exploit the assumption that anomalies stand out in terms of local patterns and distances.

## **3. Proximity to Neighbors:**

- Description: Normal instances are expected to have close neighbors in the feature space. Anomalies may be isolated or have fewer nearby instances, resulting in larger distances to their neighbors.
- Implication: Algorithms such as k-nearest neighbors (KNN) and LOF assume that anomalies can be identified based on their distances to neighboring instances.

## **4. Assumption of Majority:**

- Description: The assumption that anomalies are in the minority suggests that normal instances dominate the dataset. As a result, normal instances are expected to exhibit more consistent patterns and closer proximity to each other.
- Implication: Outliers or anomalies are identified by their deviations from the majority, often using distance-based metrics.

## **5. Global and Local Distances:**

- Description: Distance-based methods often distinguish between global distances (overall patterns in the entire dataset) and local distances (patterns within local neighborhoods). Anomalies may exhibit deviations in both global and local distances.
- Implication: Techniques like Isolation Forest focus on isolating anomalies by exploiting their tendency to have shorter paths in the feature space.

## **6. Assumption of Stationarity (in Time Series):**

- Description: In the context of time series analysis, distance-based anomaly detection may assume stationarity, where normal behavior remains relatively constant over time. Anomalies are expected to introduce variations in the time series.

- Implication: Time series anomaly detection methods, such as those based on dynamic time warping or Euclidean distance, rely on the assumption that anomalies disrupt the regular temporal patterns.

## 7. Consistency in Density:

- Description: The assumption that normal instances exhibit consistent density implies that anomalies are identified based on their deviations from expected density patterns. Anomalies may have lower or higher density compared to normal instances.
- Implication: Density-based methods consider anomalies as instances that do not conform to the expected density of normal behavior.

## 8. Homogeneous Clusters:



- Description: The assumption that normal instances form homogeneous clusters suggests that anomalies may be isolated or exhibit dissimilar patterns. Anomalies are identified based on their disruption of homogeneous clusters.
- Implication: Clustering-based methods, including k-means, assume that anomalies may be detected based on their distance from cluster centroids.

It's important to note that the effectiveness of distance-based anomaly detection methods depends on the validity of these assumptions in the specific context of the data. Careful consideration of the characteristics of the dataset and the nature of anomalies is essential for choosing an appropriate distance-based method and interpreting the results accurately. Additionally, in some scenarios, combining distance-based methods with other anomaly detection approaches may enhance overall performance.

Q6. How does the LOF algorithm compute anomaly scores?

Ans: The Local Outlier Factor (LOF) algorithm is a proximity-based anomaly detection method that computes anomaly scores for each data point in a dataset. LOF measures the local density deviation of a data point with respect to its neighbors, identifying points that have a significantly lower density than their neighbors. Here's how LOF computes anomaly scores:

### 1. Local Reachability Density (LRD):

- For each data point
- 
- $p$ , LOF calculates its local reachability density (
- 

- $LRD$
- $p$
- 
- $\rho$ , which represents the inverse of the average of the reachability distances from
- $p$
- $p$  to its
- $k$ -nearest neighbors. The reachability distance between points
- $p$  and
- $q$  is the maximum of the distance between them and the core distance of
- $q$ .

The local reachability density is computed as follows:

- 
- $LRD(p) = \frac{1}{|N(p)|} \sum_{o \in N(p)} \frac{core\_distance(o)}{\max(dist(p, o), core\_distance(o))}$
- $LRD$
- $p$
- 
- $=$
- $k$
- $1$
- 
- $\sum$
- $o \in neighbors(p)$
- 
- $core\_distance(o)$
- $\max(dist(p, o), core\_distance(o))$
- 
- $)$
- $-1$
- $dist(p, o)$
- $dist(p, o)$  is the distance between points
- $p$

- $p$  and
- $\diamond$
- $o$ , and
- $\text{core\_distance}(\diamond)$
- $\text{core\_distance}(o)$  is the core distance of point
- $\diamond$
- $o$ .

## 2. Local Outlier Factor (LOF) Calculation:

- For each data point
- $\diamond$
- $p$ , LOF computes its Local Outlier Factor (
- $\diamond\diamond\diamond\diamond$
- $LOF$
- $p$
- 
- ), which measures the local density deviation of
- $\diamond$
- $p$  compared to its neighbors. It is the ratio of the average local reachability density of the neighbors of
- $\diamond$
- $p$  to the local reachability density of
- $\diamond$
- $p$ .

The Local Outlier Factor is calculated as follows:

- 
- $\diamond\diamond\diamond\diamond = \sum_{\diamond \in \text{neighbors}(\diamond)} \diamond\diamond\diamond\diamond \text{num\_neighbors}(\diamond) \times \diamond\diamond\diamond\diamond$
- $LOF$
- $p$
- 
- =
- $\text{num\_neighbors}(p) \times LRD$
- $p$
- 
- $\Sigma$

- $o \in \text{neighbors}(p)$
- 
- $LRD$
- $o$
- 
- 
- 
- $\text{num\_neighbors}(\diamond)$
- $\text{num\_neighbors}(p)$  is the number of neighbors of point
- $\diamond$
- $p$ .

### 3. Anomaly Score:

- The anomaly score of each data point
- $\diamond$
- $p$  is then computed based on its Local Outlier Factor (
- $\diamond\diamond\diamond\diamond$
- $LOF$
- $p$
- 
- ). The anomaly score reflects how much
- $\diamond$
- $p$  deviates from its neighbors in terms of local density.

The anomaly score is calculated as:



- 
- Anomaly Score  $\diamond = \diamond\diamond\diamond\diamond$
- Anomaly Score
- $p$
- 
- $=LOF$
- $p$
- 

### Interpreting Anomaly Scores:

- An anomaly score significantly greater than 1 indicates that the point is less dense than its neighbors, suggesting that it is an outlier or anomaly.

- An anomaly score close to 1 indicates that the point has a similar local density to its neighbors, suggesting that it is likely a normal instance.

## Key Considerations:

- The choice of the parameter
- 
- $k$  (the number of neighbors) influences the sensitivity of the LOF algorithm. A larger
- 
- $k$  considers a broader local neighborhood.
- Anomaly scores are relative, and the threshold for identifying anomalies is typically determined empirically.

The LOF algorithm is effective in identifying local deviations in density, making it suitable for scenarios where anomalies exhibit different local density patterns than normal instances. It is a versatile algorithm applicable to various types of data and can be particularly useful when dealing with datasets with varying densities or complex structures.

Q7. What are the key parameters of the Isolation Forest algorithm?

Ans: The Isolation Forest algorithm is an ensemble-based anomaly detection method that isolates anomalies by leveraging the fact that anomalies are typically rare and have attributes that make them easy to separate from normal instances. The key parameters of the Isolation Forest algorithm include:

### 1. Number of Trees (

$n\_estimators$ ):

- Description: This parameter determines the number of isolation trees in the ensemble. Each tree is trained independently, and the final anomaly score is often the average or sum of the scores from all trees.
- Default: 100

### 2. Subsample Size (

◆◆◆\_◆◆◆◆◆◆◆◆

*max\_samples*):

- Description: The maximum number of samples used to build each isolation tree. A smaller subsample size can lead to faster training times but may reduce the diversity of the trees in the ensemble.
- Default: "auto" (uses a default value based on the number of instances in the dataset)

### 3. Contamination:

- Description: This parameter sets the expected proportion of anomalies in the dataset. It is used to determine the threshold for classifying instances as anomalies. For example, if
- contamination=0.1
- contamination=0.1, the algorithm will classify the top 10% of instances with the highest anomaly scores as anomalies.
- Default: "auto" (estimated as the proportion of outliers in the dataset)

### 4. Maximum Depth of Trees (

◆◆◆\_◆◆◆◆◆h

*max\_depth*):

- Description: The maximum depth of each isolation tree. Deeper trees can capture more complex relationships but may also lead to overfitting.
- Default: None (unbounded)

### 5. Bootstrap:

- Description: Specifies whether to use bootstrapping when sampling instances for building each tree. Bootstrapping involves sampling with replacement.
- Default: True

### 6. Random Seed (



*random state*):

- ## 7. Behaviour (

*behaviour*):

- ## 8. Warm Start:

- ## 9. Evaluation Metric (

*anomaly evaluation*):

- **Description:** Determines the metric used to evaluate the anomalies. The default is "deprecated," and alternatives include "error," "average\_path\_length," and "improved."

- Default: "deprecated"

These parameters allow users to customize the behavior of the Isolation Forest algorithm based on the characteristics of the data and the specific requirements of the anomaly detection task.

It's important to experiment with different parameter settings and evaluate the algorithm's performance to find the configuration that works best for a given dataset.

Q8. If a data point has only 2 neighbours of the same class within a radius of 0.5, what is its anomaly score using KNN with  $K=10$ ?

Ans: In the k-nearest neighbors (KNN) algorithm for anomaly detection, the anomaly score of a data point is often computed based on the distances to its k-nearest neighbors. If a data point has only 2 neighbors of the same class within a radius of 0.5 and  $K$  (the number of neighbors) is set to 10, the anomaly score can be calculated as follows:

Given:

- Number of neighbors ( $K$ ) = 10
- Data point has 2 neighbors of the same class within a radius of 0.5

## Anomaly Score Calculation:

Calculation of Anomaly Score:

- In the KNN algorithm for anomaly detection, the anomaly score is often based on the distances to the k-nearest neighbors. A common approach is to consider the distance to the  $K$ -th nearest neighbor.

Scenario:

- In this case, the data point has only 2 neighbors of the same class within a radius of 0.5. This implies that, within the specified radius, there are fewer than  $K$  neighbors.

Anomaly Score Calculation:

- Since there are fewer than  $K$  neighbors within the specified radius, the anomaly score is high, indicating that the data point is likely an outlier.

- $K$  neighbors, the anomaly score for this data point may be influenced by the distance to the farthest neighbor within the specified radius.
- It's important to note that the specific formula for anomaly score calculation may depend on the implementation or the definition used in the KNN algorithm for anomaly detection. Commonly, the anomaly score can be inversely proportional to the distance to the
- $\diamond$
- $K$ -th nearest neighbor.

Example Formula:

- Anomaly Score =
- $\frac{1}{\text{Distance to the } \diamond\text{-th nearest neighbor}}$ 
  - Distance to the  $K$ -th nearest neighbor
  - 1
  -
- In this case, the anomaly score might be higher if the distance to the
- $\diamond$
- $K$ -th nearest neighbor is smaller, indicating that the data point is farther away from its neighbors.

Please note that the exact formula for anomaly score calculation can vary, and it's important to refer to the specific documentation or implementation details of the KNN algorithm being used for anomaly detection.

Q9. Using the Isolation Forest algorithm with 100 trees and a dataset of 3000 data points, what is the anomaly score for a data point that has an average path length of 5.0 compared to the average path length of the trees?

Ans: In the Isolation Forest algorithm, the anomaly score for a data point is often based on the average path length in the isolation trees. The average path length is the average number of edges traversed to reach the data point in all the trees of the ensemble. A shorter average path length is typically associated with anomalies. The anomaly score can be inversely proportional to the average path length.

Given:

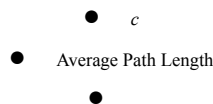
- Number of trees (
- $\diamond \diamond$
- $n_{estimators} = 100$

- Dataset size = 3000 data points
- Average path length of the data point = 5.0

## Anomaly Score Calculation:

Calculation of Anomaly Score:

- The anomaly score in Isolation Forest is often inversely proportional to the average path length. A common formula may be
- $\text{Anomaly Score} = 2 - \text{Average Path Length}$
- $\text{Anomaly Score} = 2$
- –



- , where
- $c$
- $c$  is a normalization factor.

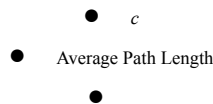
Normalization Factor (

$c$ ):

- The normalization factor
- $c$
- $c$  is often computed based on the average path length distribution in the isolation trees. It helps normalize the anomaly scores across different datasets and tree ensembles.

Example Calculation:

- Using the formula
- $\text{Anomaly Score} = 2 - \text{Average Path Length}$
- $\text{Anomaly Score} = 2$
- –



- , let's assume that
- $c = 0.5$
- $c = 0.5$  (this is an arbitrary value for illustration purposes).
- $\text{Anomaly Score} = 2 - 5.0 \cdot 0.5$
- $\text{Anomaly Score} = 2$
- –

- 0.5
- 5.0
- 

- Anomaly Score=2–10.0
- Anomaly Score=2
- –10.0
- The final computed anomaly score will depend on the exact formula and normalization factor used in the specific implementation of the Isolation Forest algorithm.

Please note that the normalization factor



$c$  is typically determined during the training phase and may depend on the characteristics of the dataset and the specific implementation details. The anomaly score provides a measure of how unusual or isolated the data point is within the ensemble of isolation trees. Lower anomaly scores suggest a higher likelihood of the data point being an anomaly.