xuax878sv

September 29, 2023

```
[ ]: # Q1. What is anomaly detection and what is its purpose?
```

0.1 Anomaly detection is a technique to find unusual or unexpected things in data. Its purpose is to identify outliers or irregular patterns, which can be useful in fields like cybersecurity, fraud detection, quality control, and more.

0.2 Anomaly detection, also known as outlier detection. Anomaly detection involves identifying data points or patterns that do not conform to the expected behavior of the system or dataset. These anomalies can take various forms, such as outliers, spikes, dips, or unexpected trends.

```
[ ]: # Q2. What are the key challenges in anomaly detection?
```

The key challenges in anomaly detection include:

Data Imbalance: Anomalies are often rare compared to normal data, leading to imbalanced datasets that can affect model training and evaluation.

Labeling Anomalies: Obtaining labeled data for training and evaluation can be difficult, as anomalies may be rare and may not always be clearly defined.

Feature Selection: Choosing relevant features or attributes for anomaly detection is crucial, and selecting the wrong ones can hinder detection accuracy.

```
[ ]: # Q3. How does unsupervised anomaly detection differ from supervised anomaly detection?
```

Unsupervised anomaly detection and supervised anomaly detection are two distinct approaches used to identify anomalies in data. Here's how they differ:

Unsupervised Anomaly Detection: It identifies anomalies without labeled data, suitable for exploring unknown or rare anomalies.

Supervised Anomaly Detection: It uses labeled data to train models to classify anomalies, suitable for well-defined anomaly detection tasks with prior knowledge.

1 Key Differences:

Data Labeling: The primary difference is the presence or absence of labeled data. Unsupervised methods work without labeled anomalies, while supervised methods require them.

Use of Prior Knowledge: Unsupervised methods are exploratory and do not rely on prior knowledge of anomalies. Supervised methods, on the other hand, leverage prior knowledge through labeled data.

Algorithmic Approach: Unsupervised methods focus on identifying deviations from the norm within the data itself, often using statistical or distance-based techniques. Supervised methods build models based on labeled data and apply classification techniques.

Applicability: Unsupervised methods are more suitable when anomalies are rare or not well-defined. Supervised methods are applicable when you have a clear understanding of what constitutes an anomaly and can provide labeled examples.

[ ]:

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

The main categories of anomaly detection algorithms include:

2 Statistical Methods:

Z-Score: Detects anomalies by measuring how many standard deviations a data point is from the mean.

Percentile-Based: Identifies anomalies by comparing data points to percentiles or quantiles of the data distribution.

Histogram-Based: Uses histograms to model data distribution and flag data points in low-density regions.

3 Machine Learning-Based:

Clustering Algorithms: Detect anomalies by considering data points that are distant from cluster centers as outliers (e.g., K-Means, DBSCAN).

Classification Algorithms: Treats anomaly detection as a binary classification problem, training models to classify data points as normal or anomalous (e.g., Decision Trees, Support Vector Machines).

4 Distance-Based Methods:

Mahalanobis Distance: Measures the distance of data points from the centroid, accounting for correlations between features.

K-Nearest Neighbors (KNN): Identifies anomalies based on the distance to their nearest neighbors.

5 Local Outlier Factor (LOF): Evaluates the local density of data points to identify outliers.

6 Density-Based Methods:

Kernel Density Estimation (KDE): Models the data distribution as a continuous probability density function and flags data points in low-density regions.

7 Proximity-Based Methods:

8 DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Groups data points into clusters and considers data points in low-density clusters as anomalies.

[ ]:

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

9 Distance-based anomaly detection methods assume that:

10 Data can be represented in Euclidean space.

11 Normal data points are in high-density regions.

12 Anomalies are far from their nearest neighbors.

13 Data points are generated from a common distribution.

14 Anomalies can be detected using a threshold.

15 Scalability challenges in high-dimensional data may exist.

16 Data distribution is relatively uniform with rare anomalies.

[ ]:

```python
# Q6. How does the LOF algorithm compute anomaly scores?
```

The LOF (Local Outlier Factor) algorithm computes anomaly scores for data points by comparing their local density to the local densities of their neighboring data points.

17 The LOF (Local Outlier Factor) algorithm computes anomaly scores as follows:

It calculates the local density of a data point by comparing its distance to its ‘k’ nearest neighbors.

It computes the LRD (Local Reachability Density) for the data point, which measures its local density relative to its neighbors.

The LOF score is determined by comparing the LRD of the data point to the LRD of its neighbors.

A high LOF score (>1) indicates that the data point is in a less dense region compared to its neighbors, suggesting it may be an anomaly.

[ ]:
```
# Q7. What are the key parameters of the Isolation Forest algorithm?
```

The key parameters of the Isolation Forest algorithm are:

17.1 n_estimators: Number of isolation trees in the forest.

17.2 max_samples: Maximum number of data points used for each tree.

17.3 max_features: Maximum number of features considered for splitting.

17.4 contamination: Expected proportion of anomalies in the data.

17.5 random_state: Seed for random number generation.

17.6 bootstrap: Whether to use bootstrap sampling.

17.7 verbose: Controls algorithm verbosity.

17.8 n_jobs: Number of CPU cores for parallelism.

17.9 behaviour: Behavior for datasets with no anomalies.

17.10 outlier_label: Label for detected anomalies.

17.11 warm_start: Allows adding more trees to an existing model.

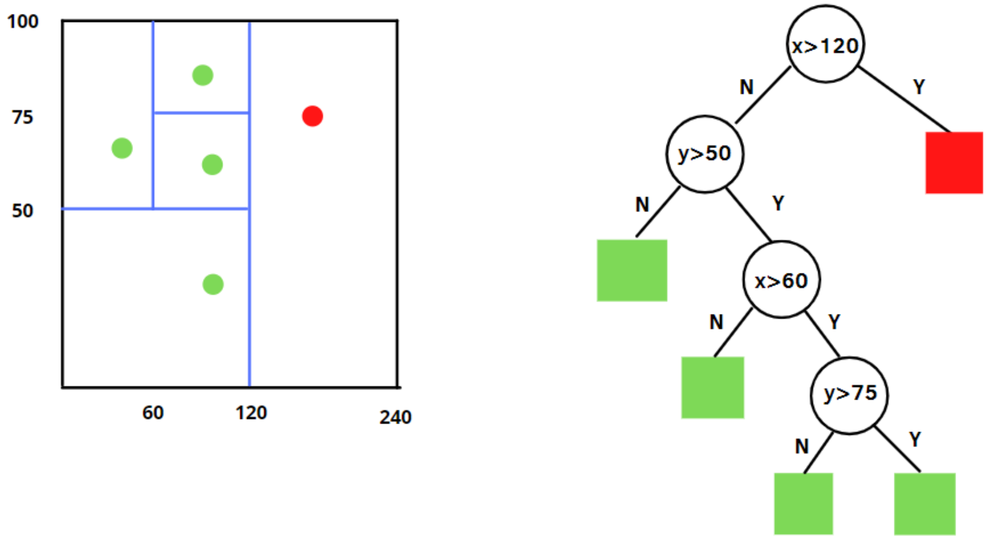
17.12 sample_mode: Controls sample drawing method.

17.13 extension_level: Determines path length calculation method.

18 The Isolation Forest algorithm is an unsupervised machine learning technique used for anomaly detection. It's particularly effective at identifying anomalies (outliers) in large datasets.

[3]:
```
from PIL import Image
import requests
from io import BytesIO
from IPython.display import display  # For Jupyter Notebook

# URL Of The Image
image_url = "https://miro.medium.com/v2/resize:fit:1400/
 ↪1*d-4xINDQHv0G82o2GUApJQ.png"

# Download The Image From The URL
```

```
response = requests.get(image_url)
image_bytes = BytesIO(response.content)

# Open The Image Using Pillow
image = Image.open(image_bytes)

# Display The Image
display(image)
```



Isolation Forests(IF), similar to Random Forests, are build based on decision trees. And since there are no pre-defined labels here, it is an unsupervised model.

In an Isolation Forest, randomly sub-sampled data is processed in a tree structure based on randomly selected features.

The samples that travel deeper into the tree are less likely to be anomalies as they required more cuts to isolate them.

Similarly, the samples which end up in shorter branches indicate anomalies as it was easier for the tree to separate them from other observations.

```
[ ]: # 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?
```

K (number of nearest neighbors) = 10

Number of neighbors with the same class = 2

Majority class = The class of the 2 neighbors

Anomaly score = K - Number of neighbors with the majority class = 10 - 2 = 8

So, the anomaly score for the data point is 8

Anomaly Score = K (10) - Number of neighbors with the same class (2) = 10 - 2 = 8

[ ]:
```
# 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.
```

18.1 In the Isolation Forest algorithm:

18.2 Anomaly Score = 2^(-average_path_length / c(n))

Where:

average_path_length is the average path length of a data point across all the trees in the forest. c(n) depends on the number of data points in the dataset.

18.3 The specific formula for calculating "c(n)" in the Isolation Forest algorithm is:

18.4 c(n) = 2 * (log(n - 1) +) - (2 * (n - 1) / n)

In this formula:

"n" represents the number of data points in the dataset. "log" is the natural logarithm. " " represents the Euler-Mascheroni constant, which is approximately 0.5772156649.

First, calculate c(3000):

18.5 c(3000) = 2 * (log(3000 - 1) + 0.5772156649) - (2 * (3000 - 1) / 3000)

c(3000)   11.83692

Now, calculate the anomaly score:

18.6 Anomaly Score   2^(-5.0 / 11.83692)

Anomaly Score   0.2371