Isolation Forest Tutorial

September 21, 2025

1 Introduction

Isolation Forest detects anomalies by randomly partitioning data points until they become isolated. Unlike density-based approaches, it relies on the observation that anomalies are easier to isolate due to their sparsity and feature-wise extremity. The algorithm builds an ensemble of random trees, yielding scores that quantify how anomalous each observation is.

2 Theory and Formulas

2.1 Random Partitioning

Each isolation tree is grown by recursively selecting a random feature and a random split value within its range. For a sample $\mathbf{x}$, the path length $h(\mathbf{x})$ equals the number of edges traversed from the root to the external node containing $\mathbf{x}$. Anomalies typically have shorter path lengths, while normal points require more splits to isolate.

2.2 Path Length and Anomaly Score

For a subsample size ψ, the average path length of unsuccessful searches in a Binary Search Tree is

$$\mathbb{E}[h(\psi)] = 2H_{\psi-1} - \frac{2(\psi-1)}{\psi}, \qquad H_n = \sum_{k=1}^{n} \frac{1}{k}. \tag{1}$$

The anomaly score for $\mathbf{x}$ is

$$s(\mathbf{x}, \psi) = 2^{-\frac{\overline{h(\mathbf{x})}}{\mathbb{E}[h(\psi)]}}, \tag{2}$$

where $\overline{h(\mathbf{x})}$ is the average path length across all trees. Scores near 1 indicate strong anomalies; scores below 0.5 are likely normal.

2.3 Hyperparameters and Complexity

Key parameters include the number of estimators, subsample size, and contamination ratio that calibrates the decision threshold. Isolation Forest runs in $O(t\psi \log \psi)$ time for t trees with subsample size ψ, offering linear scalability in the number of samples and features. Feature scaling and handling categorical variables are crucial for meaningful splits.

3 Applications and Tips

- **Fraud detection**: surface unusual transactions in finance or e-commerce.
- **Network security**: detect rare traffic signatures in high-dimensional logs.
- **Manufacturing**: flag sensor readings that deviate sharply from normal operating patterns.
- **Best practices**: standardize features, tune subsample size and contamination, inspect score distributions, and combine with domain knowledge to validate alerts.

4 Python Practice

The script `gen_isolation_forest_figures.py` creates synthetic data with injected anomalies, fits scikit-learn's IsolationForest, and visualizes anomaly scores over the feature space along with a histogram highlighting detected outliers.

Listing 1: Excerpt from gen$_{i}solation_{f}orest_{f}igures.py$

```python
from sklearn.ensemble import IsolationForest

model = IsolationForest(
    n_estimators=200,
    max_samples=256,
    contamination=0.08,
    random_state=42,
)
model.fit(points)
score = model.decision_function(points)
labels = model.predict(points)
```

5 Result

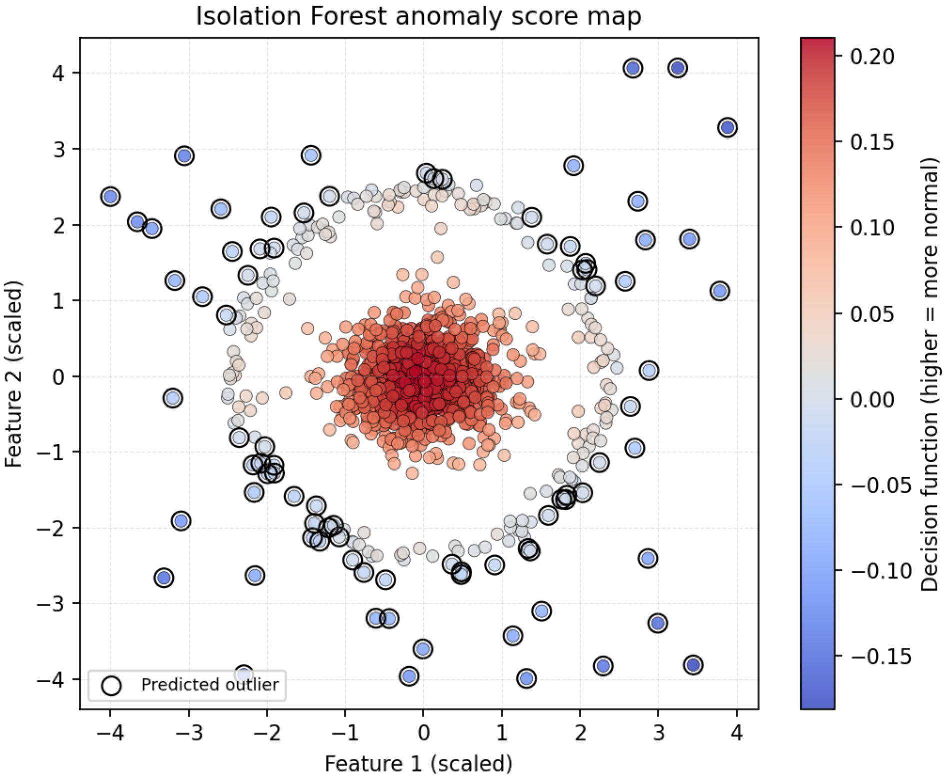

Figure 1: Isolation Forest anomaly scores on synthetic data; darker regions indicate higher anomaly likelihood

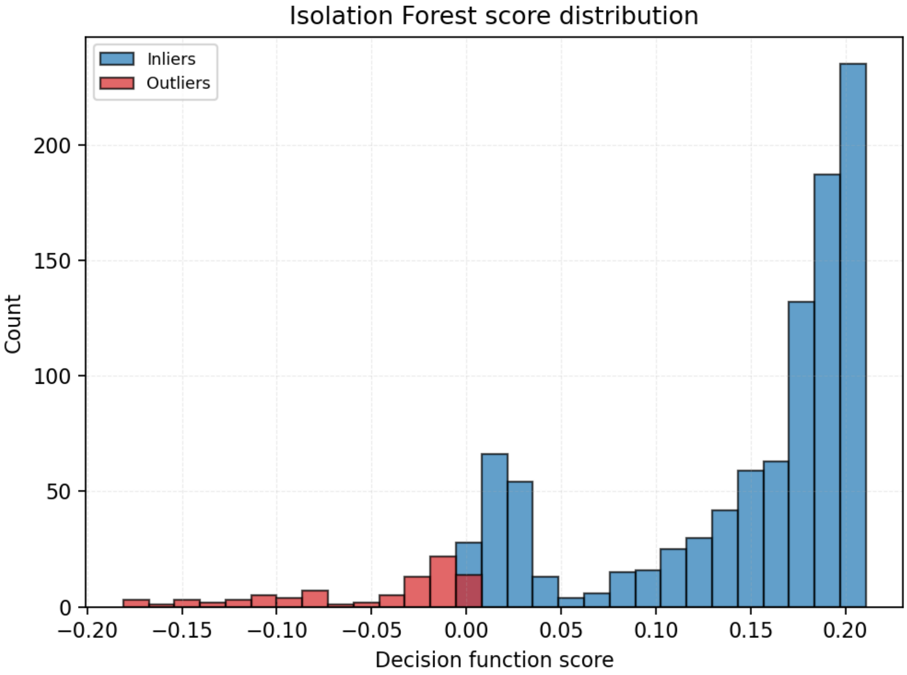

Figure 2: Histogram of anomaly scores separating predicted inliers and outliers

6 Summary

Isolation Forest isolates anomalies via random splits, producing interpretable scores with few assumptions about data distribution. Proper feature scaling and parameter tuning yield robust detectors for high-dimensional datasets. The synthetic example demonstrates how score visualizations validate threshold choices and highlight anomalous clusters.