GE Digital

Isolation Forest Outlier Detection

Description

Isolation Forest Outlier Detection uses a Random Forest of decision trees to detect data anomalies. The Isolation Forest 'isolates' observations by randomly selecting a feature, and then randomly selecting a split value between the maximum and the minimum values of the selected feature. Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node. This path length, averaged over a forest of many random trees, is a measure of abnormality. Random partitioning produces noticeable shorter paths for anomalies. When a forest of random trees collectively produce shorter path lengths for particular samples, they are highly likely to be anomalies. Isolation Forest is one of the fastest anomaly detectors and one of the few that can easily scale up to big data.

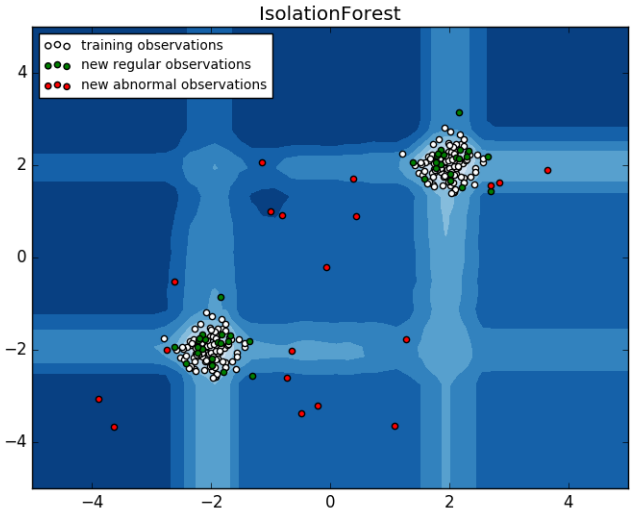


Figure 1: Isolation Forest

The Isolation Forest Outlier Detection use case is as follows:

- It can be used as a supervised learning algorithm or an unsupervised learning algorithm for detecting data anomalies. If training data is provided with "Target" labels, Isolation Forest Outlier Detection will be used as a supervised learning algorithm. If training data is missing "Target" labels, the analytic will be used as an unsupervised learning algorithm.
- If "Target" labels are provided in the training data, the labels should be "1" and "-1", where"1" stands for normal data and "-1" stands for anomalous data.
- The attributes/features should all be numeric values instead of categorical values. There should be no missing values in the attributes/features. In case of missing values in attributes/features, use an imputation method to impute missing values first. If your attributes/features have categorical values, consider using categorical to numerical analytics in our catalog to transfer dataset first.

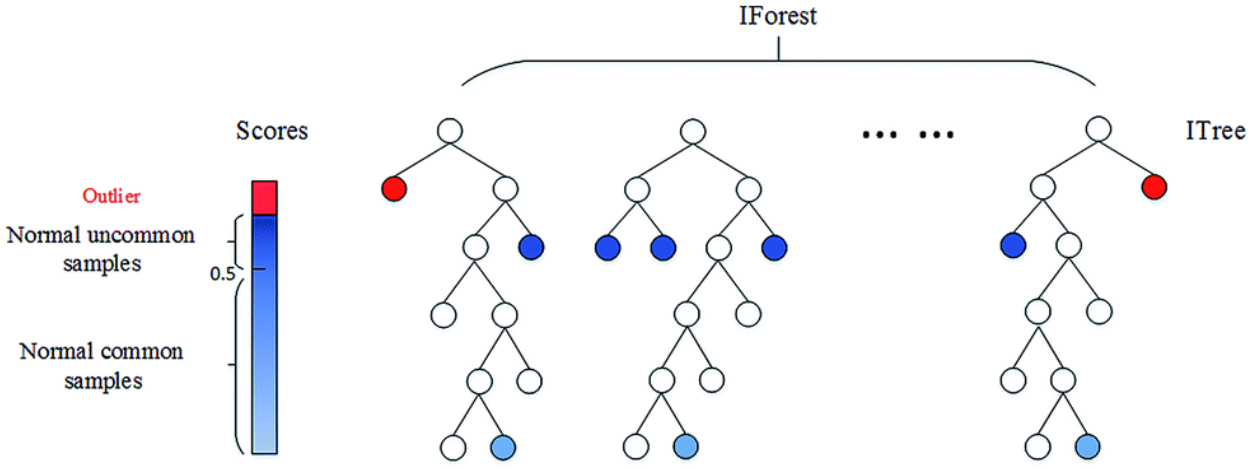
- Isolation Forest Outlier Detection is robust and can be applied to anomaly detection in high dimensions. However, time series anomaly detection is not recommended to be used here as time series data will be treated as non-time series data.

Algorithm details

Isolation Forest explicitly isolates anomalies rather than profiling normal instances as most existing model-based approaches do. It takes advantage of two anomalies' quantitative properties:

- The minority consists of fewer instances
- They have attribute-values, which are very different from those of normal instances

In other words, anomalies are 'few and different'. In Isolation Forest, a tree structure can be constructed effectively to isolate every single instance. Owing of their susceptibility to isolation, anomalies are isolated closer to the root of the tree; whereas normal points are isolated at the deeper end of the tree. This isolation characteristic of tree forms the basis of the method to detect anomalies, and this tree is called Isolation Tree or iTree. Isolation Forest builds an ensemble of iTrees (**Schema 1**) for a given dataset; anomalies are those instances, which have short average path lengths on the iTrees.



Schema 1: How Isolation Forest (iForest) Outlier Detection Works

Input

```
{
  "data": {
     "data-train": {
        "Feature1": [2.149014246, 2.194306561, 1.929753988, 2.473763845],
        "Feature2": [1.95852071, 2.456908957, 1.929758913, 2.230230419, 2.162768013],
        "Target": [1, 1, 1, -1]
```

```
    },
    "data-predict": {
   "Feature1": [2.107336208, 2.324915373, 1.58669919],
   "Feature2": [2.168235358, 2.316140616, 1.718652488]
    }
  },
  "params": {
      "contamination": 0.1
  }
}
```

Data

The "data" section contains training data and prediction data. Prediction data is optional, but training data is required in the input. For the training data, "Target" label is optional.

Params

Parameters for the analytic are located in the 'params' section of the JSON file. "contamination" means the fraction of data that is estimated to be an outlier. For example, "0.1" means that about 10% of the data is an outlier. The "Params" section is optional.

Different scenarios of input

- **Scenario 1**: Training data and contamination level are provided, and training data does not have "Target" labels. (see SampleInput1.json)
- **Scenario 2:** Training data, prediction data, and contamination level are provided. Training data does not have "Target" labels. (see SampleInput2.json)
- **Scenario 3:** Training data and prediction data are provided. Training data has "Target" labels. (see SampleInput3.json)

Output

The output of the Isolation Forest Outlier Detection analytic gives the predicted target for the training data.

```
{
  "data": {
     "data-predict": {
        "Feature2": [2.168235358, 2.3161406160000002, 1.718652488, 2.1541357849999998],
        "Target": [1, 1, -1, 1],
        "Feature1": [2.107336208, 2.3249153730000001, 1.58669919, 2.1545105800000002]
```

```
    }
  }
}
```

If the prediction data is provided, the output will be for prediction data. Otherwise, it will be for input Scenario 1, and the output will be for training data.

Usage samples

Refer to input and output descriptions.