Unsupervised Anomaly Detection with One-Class SVM

Overview

This project demonstrates **One-Class Support Vector Machine (One-Class SVM)**, an unsupervised learning algorithm designed for anomaly detection. One-Class SVM identifies data points that deviate significantly from the norm, making it valuable for applications like fraud detection and network security. (AskPython)

Key Features

1. **Environment**:
   - The algorithm operates on a dataset with features `feature1` and `feature2`, which are selected for anomaly detection.
2. **One-Class SVM**:
   - Utilizes the One-Class SVM model from scikit-learn's `svm` module.
   - The model is initialized with the Radial Basis Function (RBF) kernel, a common choice for non-linear data.
   - The `nu` parameter is set to 0.05, indicating that approximately 5% of the data is expected to be anomalies.
   - The `gamma` parameter is set to 'scale', which is a heuristic for the RBF kernel.
3. **Anomaly Detection**:
   - The model is trained on the dataset to learn the distribution of normal data points.
   - After training, the model predicts anomalies, labeling them as -1.
   - Anomalies are extracted for further analysis.
4. **Visualization**:
   - A scatter plot is generated to visualize the distribution of normal data points and anomalies.
   - Normal points are displayed in blue, while anomalies are highlighted in red.

How It Works

1. **Data Preparation**:
   - Load the dataset and select relevant features (`feature1` and `feature2`) for anomaly detection.
2. **Model Training**:
   - Initialize the One-Class SVM model with specified parameters.
   - Fit the model to the selected features of the dataset.
3. **Anomaly Prediction**:
   - Use the trained model to predict anomalies in the dataset.
   - Label anomalies as -1 and normal points as 1.
4. **Visualization**:
   - Plot the data points, distinguishing between normal points and anomalies.

Code Walkthrough

1. **Data Preparation**:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
data = pd.read_csv('your_dataset.csv')

# Select features for anomaly detection
X = data[['feature1', 'feature2']]  # Replace with relevant feature columns
```

2. **Model Training**:

```
from sklearn.svm import OneClassSVM

# Initialize and fit One-Class SVM
one_class_svm = OneClassSVM(kernel='rbf', nu=0.05, gamma='scale')
one_class_svm.fit(X)
```

3. **Anomaly Prediction**:

```
# Predict anomalies
data['anomaly'] = one_class_svm.predict(X)

# Anomalies are labeled as -1
anomalies = data[data['anomaly'] == -1]
print("Number of anomalies detected:", len(anomalies))
```

4. **Visualization**:

```
# Plot anomalies and regular points
plt.scatter(X['feature1'], X['feature2'], label='Normal', c='blue', s=20)
plt.scatter(anomalies['feature1'], anomalies['feature2'], label='Anomaly', c='red',
plt.title('One-Class SVM Anomaly Detection')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```

Advantages

- **Unsupervised Learning**: Does not require labeled data, making it suitable for scenarios where anomalies are rare or unknown.
- **Non-Linear Boundaries**: The RBF kernel allows the model to capture complex, non-linear relationships in the data.
- **Flexibility**: The `nu` parameter provides control over the proportion of anomalies, allowing for customization based on the dataset.

Future Work

- **Parameter Tuning**: Experiment with different values of `nu` and `gamma` to optimize performance for specific datasets.
- **Kernel Selection**: Explore other kernel functions (e.g., polynomial, sigmoid) to assess their impact on anomaly detection.

- **Scalability**: Investigate the model's performance on larger datasets and consider dimensionality reduction techniques if necessary.

References

- Anomaly Detection Example with One-Class SVM in Python - DataTechNotes
- SVM One-Class Classifier For Anomaly Detection - Analytics Vidhya
- Understanding One-Class SVM for Anomaly Detection - AskPython