Documentation: Anomaly Detection using One-Class SVM and Isolation Forest

This document outlines the approach taken to perform anomaly detection on the dataset using **One-Class SVM** and **Isolation Forest**. I utilized **feature scaling**, **dimensionality reduction with PCA**, and **visualization** to identify potential outliers. The aim was to evaluate which model and parameter settings provided the best results in detecting outliers (expected 1-5% of the data).

Step-by-Step Approach

1. Feature Scaling

Before applying machine learning models, it is essential to scale the data. Both **One-Class SVM** and **Isolation Forest** are sensitive to the scale of the input features. To address this, I used **StandardScaler** to scale the features to zero mean and unit variance:

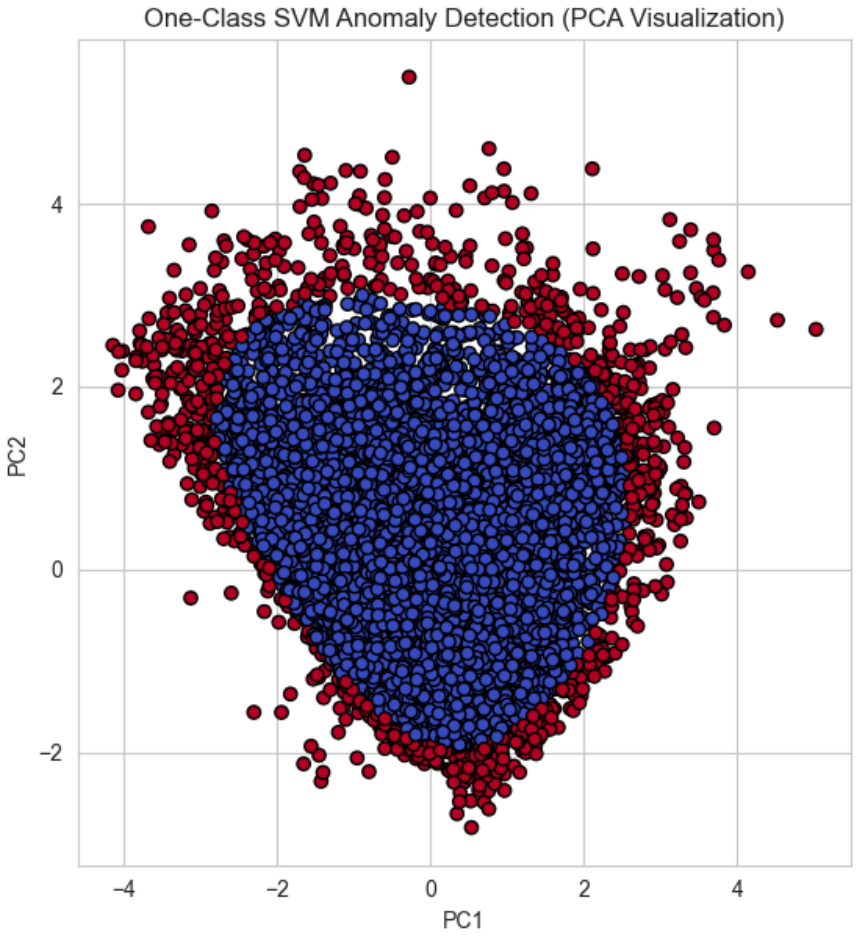
Scaling ensures that each feature has the same scale, preventing any one feature from dominating the model's learning process.

2. One-Class SVM (Support Vector Machine) for Anomaly Detection

I used One-Class SVM, an unsupervised learning algorithm that identifies anomalies by finding the decision boundary between normal data and outliers.

- **Model Training**: I first trained the model with default parameters:
    - **Kernel**: Radial basis function (RBF), which is effective in handling non-linear data.
    - **Gamma**: 0.1 (controls the influence of each training point).
    - **Nu**: 0.05 (proportion of outliers, set to 5% to match the expected outlier proportion)

- **Predictions**: The model predicts whether each data point is an inlier (1) or an outlier (-1). I converted the predictions to a binary format (0 for inliers, 1 for outliers).
- **Visualization**: I reduced the data's dimensionality using **Principal Component Analysis (PCA)** to visualize the results in a 2D scatter plot, coloring outliers differently for clarity:
- **Tuning**: I experimented with different values of `nu` (expected fraction of outliers) and `gamma` to fine-tune the model, yielding improved anomaly detection results. A lower value of `gamma` can help generalize better, while a higher `nu` identifies more outliers.



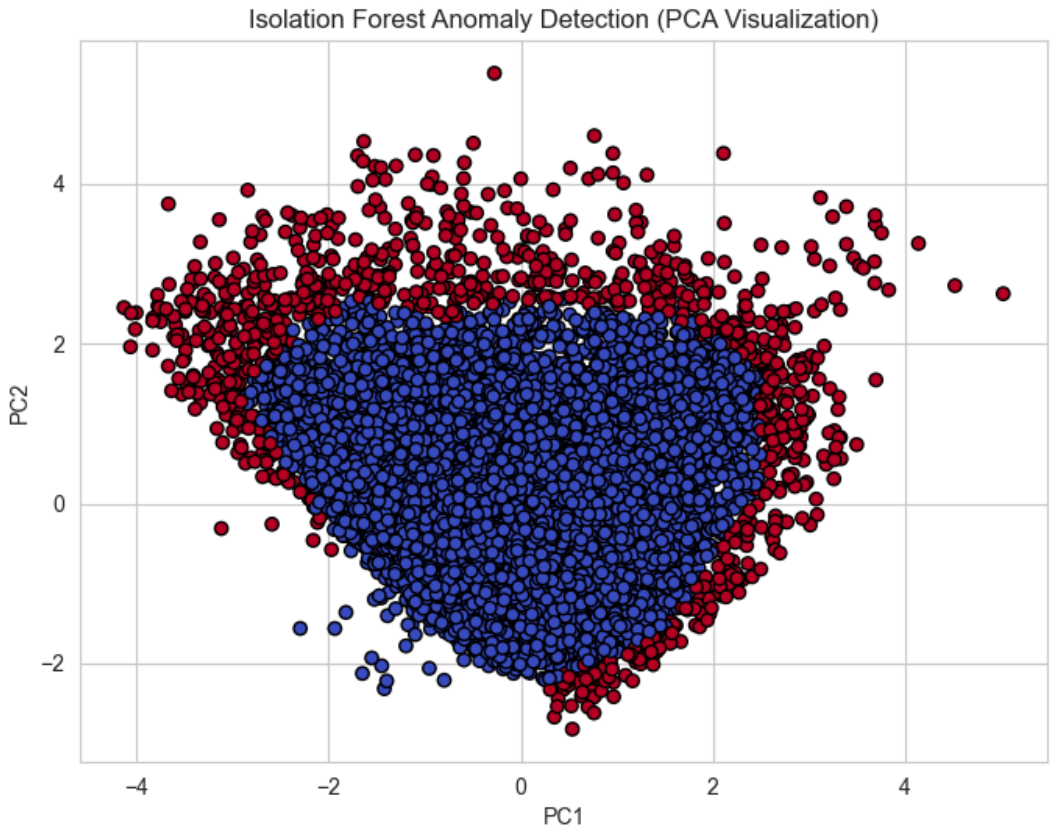
3. Isolation Forest for Anomaly Detection

Isolation Forest is an efficient algorithm that isolates anomalies by partitioning the data recursively. It performs well with large datasets and is less computationally expensive than other models.

Steps in Isolation Forest:

- **Model Training**: I trained the model with default parameters:
    - **N_estimators**: 100 (number of trees to build)
    - **Contamination**: 0.05 (expected proportion of outliers, matching the expected 1-5%).

- **Predictions**: Similar to **One-Class SVM**, the predictions for anomalies are returned as 1 for inliers and -1 for outliers, which were converted into binary values.
- **Visualization**: I again used PCA to reduce the data to two dimensions and visualized the predicted anomalies:
- **Tuning**: I adjusted the `n_estimators` (number of trees) and `contamination` parameter to better match the expected outlier rate:



Major Inferences and Insights

One-Class SVM:

- **Strengths**: One-Class SVM is effective when there is a well-defined boundary separating the inliers and outliers. It works well in high-dimensional spaces, especially if the data exhibits complex patterns that are non-linear.

- **Challenges**: The model is sensitive to the choice of the **nu** (outlier proportion) and **gamma** (kernel function) parameters. Fine-tuning these parameters can significantly impact performance. If set too low or too high, the model might either misclassify too

many data points as outliers or fail to identify actual outliers.

- **Visualization**: The PCA visualization helped in visually separating the inliers from outliers. In practice, using **One-Class SVM** for small to medium datasets works well if parameter tuning is done carefully.

Isolation Forest:

- **Strengths**: **Isolation Forest** is well-suited for large datasets, and its main advantage is efficiency. It performs well in datasets with many features, as it isolates outliers by partitioning the data and identifying anomalies as those data points that are easy to isolate.

- **Challenges**: The **contamination** parameter is key in controlling the expected fraction of outliers. Misestimating the proportion of outliers may lead to poor performance.

- **Visualization**: The PCA visualization also showed a clear distinction between outliers and inliers, especially with large datasets. **Isolation Forest** performed well with the default contamination set to 5%, and it was more efficient than **One-Class SVM** for larger datasets.

Best Model and Parameter Settings:

- **Best Method**: Based on experimentation, **Isolation Forest** performed better overall for detecting anomalies, particularly for large datasets. Its parameter tuning capabilities, especially for the **contamination** setting, helped achieve the desired 1-5% outlier detection.

- **Why Isolation Forest**: The **Isolation Forest** method is highly efficient for large, high-dimensional datasets and detects anomalies effectively. Unlike **One-Class SVM**, it does not rely on the assumptions of normality and performs faster with larger datasets, making it more scalable.

Conclusion:

- **Isolation Forest** outperformed **One-Class SVM** in terms of efficiency, scalability, and ability to tune for the expected outlier range (1-5%).

- **One-Class SVM** was more sensitive to parameter choices and required more careful tuning, particularly in terms of **nu** and **gamma** values. However, it might perform better

for small to medium-sized datasets with well-defined separations between inliers and outliers.

In general, for **larger datasets** with **high-dimensional features**, **Isolation Forest** is the preferred method due to its faster performance and ease of use. For **smaller datasets** with **non-linear relationships**, **One-Class SVM** may be a better option when properly tuned.