Data Mining HW3
Anomaly Detection

109350008 張詠哲

1. Explain your implementation which get the best performance in detail.

For this assignment, I tried 4 different method to do the anomaly detection, which are one-class SVM, Isolation Forest, VAE, rule-based method. And the overall performance shows in following table. (Because I don't download extra dataset for validation. Thus, I take Kaggle public score as the validated score for experiment) Also I will explain the rule-based method lately.

	Kaggle public score (AUC)
One-class SVM	0.85078
IsoForest	0.81036
VAE	0.67476
Rule-based	0.99634

In the beginning, I used the machine learning or deep learning method, such one-class SVM, isolation forest, and the VAE. All of them can't perform well. So, I check the data distribution. Found that there have many similar data between the training dataset and the testing dataset, also the feature space is small. Therefore, I tried the rule-based method, which calculates the distance between the testing data to the closest training data as the weight. And it gets the highest result, there is my possible explanation.

The test data closely resembles the training data in terms of feature distribution. So, the distance-based method can be effective. Also, maybe the normal data points are densely clustered and anomalies are sparse and distant, a simple distance measure will effectively separate them. And the feature space is quite small causing the distance can be significant distinguish from normal data.

Also, for the one-class SVM if the classes can't be well-represented by a hyperplane or hypersphere in the feature space, performance can suffer. And for the isolation forest. If anomalies are not well-isolated or if normal data points are spread out in a way that does not facilitate easy isolation, performance may degrade. For the VAE, I think maybe is due to the similarity is quiet high between each feature. So, the VAE

can't map the normal and anomalous data cleanly to different regions of this latent space.

2. Explain the rationale for using AUC score instead of F1 score for binary classification in this homework.

1. Imbalanced data distribution
Anomaly detection problems typically involve imbalanced datasets where the number of normal instances far exceeds the number of anomalies (for this assignment is 3:2 for normal:anomaly data). In this scenarios, traditional metrics F1 score can be misleading. Also, the AUC score is robust to class imbalance as it evaluates the performance of the model across all possible threshold values.

2. Threshold
The AUC measures the model's ability to distinguish between positive (anomaly) and negative (normal) classes across all threshold values. It provides an aggregate measure of performance regardless of the specific threshold chosen. While F1 score is calculated at a specific threshold, typically where precision and recall are balanced. In anomaly detection, the optimal threshold may vary, and focusing on a single threshold may not capture the overall performance of the model.

3. Trade-off between precision and recall
AUC score can consider both the sensitivity and false positive rate across different thresholds. This is crucial in anomaly detection where we want to minimize false positives without missing too many true anomaly data. While F1 score balances precision and recall, it does so at a specific threshold. This might not reflect the model's performance at different decision boundaries, which is often necessary in the context of anomaly detection.

3. Discuss the difference between semi-supervised learning and unsupervised learning.

The mainly different from semi-supervised and unsupervised is that the they deal with different amounts of labeled data. Unsupervised learning involves training models on data that has no labeled responses. The goal is to find the hidden patterns or latent structures within the data. Some common model includes clustering (k-means, hierarchical clustering), and dimensionality reduction (PCA, t-SNE). While semi-supervised learning combines a small amount of labeled data with a large

amount of unlabeled data. It leverages the labeled data to guide the learning process and improve the model's performance compared to using only unlabeled data. it often involve self-training, co-training, and graph-based methods.