DM hw3

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

   I tried several methods, including clustering, VAE and another rule-based method. For clustering, I tried to cluster the 5 groups with different classes using k-means and DBSCAN. However, the data has quite a high dimension, so it is hard to decide the number of class (since the data point in single class might not be close enough in high dimemion space). For VAE, I tried different latent code dimensions, but it turned out to perform poor (ROC-AUC score about 0.5).

   Due to these results, I looked closely at the data attributes. The number of features is quite small (about 15 features, each having about 10 features at most). Hence, I think it's a good idea to fix a multivariate distribution on the data. Observing that the dataset is quite large with respect to the feature space, I tried an easier approach, that is, using the training dataset as experience. Given a new testing data, I will find the nearest feature vector in the training dataset (the metric is L2 vector norm). This approach achieves 0.99634 AUC-ROC. In the testing dataset, there are 537 feature sets that already exist in training dataset. Since there should be exactly 500 data point belonging to the class in training dataset, there are some datapoints that should not be included. For all the data existing in the original dataset, I count the number of exisitence in the training dataset and take the negative value as prediction (the smaller value it is, less likely the prediction will be). This result did not help increasing the public score, but I think it might help in private dataset.

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

   If we want to use F1 score as the metric, we can only output the exact classification label as prediction. However, in this lab, we might have predictions based on distance (clustering and SVM) and reconstruction error (VAE). If we use auc score instead of f1 score, we don't need to modify these values to labels (don't need to define a threshold of the values), making it

easier to evaluate how model is correct on these predicted values.

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

   Semi-supervised learning and unsupervised learning are distinct paradigms in machine learning, primarily differing in their use of labeled data. Unsupervised learning works with unlabeled data to uncover hidden patterns and intrinsic structures within the data, with common techniques including clustering and dimensionality reduction (VAE). It is primarily used for exploratory data analysis and preprocessing tasks. In contrast, semi-supervised learning leverages a combination of a small amount of labeled data and a large amount of unlabeled data to enhance learning accuracy (SVM). This approach is particularly useful when labeled data is scarce or expensive to obtain but there are a lot of unlabeled data. Techniques in semi-supervised learning, such as self-training and co-training, use the labeled data to guide the model and improve its performance. While unsupervised learning aims to identify underlying patterns without prior knowledge of outcomes, semi-supervised learning aims to improve predictive models by effectively utilizing the available labeled and unlabeled data.