Data splitting for Anomaly detection

When training **anomaly detection** tasks **and** you **do have labeled data** for anomaly detection (e.g., labeled as "anomaly" or "normal"), you can approach this task using both supervised and unsupervised algorithms.
If you want to evaluate your model (supervised or unsupervised) and provide evaluation metrics - you need to work carefully to prevent data leakage.

Here's how you should approach splitting the data and evaluating model performance:

A. Supervised Model (if labels are used in training):
    - Use both normal and anomalies in training.
    - Don’t forget to use stratified train_test_split.
    - Train using X_train, y_train.
    - Evaluate on X_val, y_val / X_test, y_test using metrics like precision, recall, F1.
B. Unsupervised Model (e.g., Isolation Forest, HDBSCAN):
    - Option 1:
        i. Use stratified split like for Supervised model
    - **Option 2 (semi-supervised split):**
        i. Train = only normal samples
        ii. Test = mix of normal + anomalies (labeled)

Note, it is possible to train Isolation Forest and DBSCAN / HDBSCAN **only on normal data** since these **unsupervised** algorithms only learn patterns of the data, and they don’t actually need (like supervised algorithms) to see anomalies during training.