Unsupervised Learning Lab Assignment:

In this assignment, you will work with the TEP dataset to explore PCA (1) and clustering (2) methods for fault detection. Your main task is to develop the code and to perform analysis and build a practical understanding of the concepts used.

1. **Dimensionality Reduction & Fault Detection using PCA (15 points)**

2. **Clustering with K-Means and DBSCAN for Noise / Outlier Removal (5 points)**

Dataset Link: The Tennessee Eastman Process (TEP) fault simulation dataset. Download the CSV dataset from this link: Custom TEP Dataset - Github Repo

Task 1: ----------

Goal: Industrial process data (like TEP) is highly multivariate with dozens of sensors and manipulated variables. To monitor process health and visualize structure (normal vs fault), we first reduce dimensionality using PCA, then we detect faults using Hotelling's T^2 statistic.

Instructions:

1. **Load the dataset. Ensure you have features (sensor + manipulated variables) and a label.**
2. **Pre-process:**
    - Standardize/normalize features (zero mean, unit variance).
3. **Load the four datasets:**
    - tep_faultfree_training.csv
    - tep_faultfree_testing.csv
    - tep_faulty_training.csv
    - tep_faulty_testing.csv
    - tep_mixed_labeled.csv
    - Use the **training fault-free data** to fit the PCA model. Use the **testing fault-free** and **testing faulty** datasets to evaluate fault detection.
    - Standardize the features (zero mean, unit variance) using the scaling parameters from the training fault-free data.

4. **Apply PCA to the feature set:**
    - Compute the principal components; examine explained variance by the first few components.
    - Choose a number of components (e.g., 2 or 3) that capture a suitable amount of variance (say >70%) based on training dataset.
    - Plot a scatter of the dataset in the PCA space (e.g., PC1 vs PC2), with colors indicating normal vs fault or different fault types.

5. **PCA-Based Fault Detection using Hotelling's T^2**
    - Compute PCA and Hotelling's T^2 statistic.
    - Fit PCA on the **fault-free training** data.

- For each sample (in both fault-free and faulty test sets), compute the **Hotelling's T²** statistic:
  - $T^2 = x_{score}^T \Lambda^{-1} x_{score}$ where x_{score} are the PCA scores and Λ are the eigenvalues (variances of each PC).
  - Compute the **95% confidence threshold** for T² using:
    - $T_{lim}^2 = \frac{a(n-1)}{n-a} F_{a,n-a,0.95}$

      where:

      1. a = number of principal components retained
      2. n = number of training samples
      3. $F_{a,n-a,0.95}$ = 95th percentile of the F-distribution

  Classify a point as **faulty if** $T^2 > T_{lim}^2$.

6. **Visualization:**
   - Plot the T² statistic for test samples (both fault-free and faulty) against the threshold line.
   - Create a confusion matrix or simple counts of detected vs actual faults.
7. **Compare and interpret:**
   - How well does PCA perform; does it segregate the faulty and non-faulty data?
   - Are the clusters visible? Are there overlaps? What might that mean in process monitoring context?
   - Report how well the PCA-based T² detection distinguishes faulty and fault-free samples.
   - Discuss any false positives or false negatives.
   - Interpret results: e.g., "certain faults are not detected well because they lie in PCA subspace similar to normal operation."

Deliverables:

1. Scatter plot of PCA embedding (with axis labels, legend). (3 pts.)
2. A short commentary describing what you observe: separation, overlap, clustering structure, implications for fault detection/monitoring. (3 pts.)
3. Plot of T² statistic vs sample index for test datasets (fault-free and faulty), showing the 95% confidence threshold. (3 pts.)
4. Table or summary of detection accuracy (e.g., number of true faults detected). (3pts.)
5. Short discussion interpreting the effectiveness of PCA + T² as a fault detection tool. (3 pts.)

Task 2: ----------

Goal: In monitoring industrial processes, it's useful to identify clusters of "normal" behavior and isolate outlying observations (potential faults or anomalies). Clustering algorithms like K-Means can segment data into groups, and density-based clustering such as DBSCAN can identify noise/outliers. Applying these to the TEP dataset helps students explore unsupervised fault-detection and process monitoring.

Instructions:

1. **Use the provided TEP datasets.**
    - Fit PCA on **fault-free training data** to obtain the transformation matrix.
    - Then apply PCA to the **combined test dataset** (normal + faulty).
2. **Use the PCA-reduced data (X_pca) as input for clustering:**
    - **K-Means**: Start with k=2 (normal/fault), compare clusters with true labels.
        - Apply K-Means with k=2 (normal vs faulty hypothesis).
        - Compare predicted clusters with true labels (label column).
        - Calculate cluster purity or confusion matrix.
        - Visualize results in 2D (e.g., PC1 vs PC2) colored by cluster.
    - **HDBSCAN and DBSCAN**: Detect dense regions (normal clusters) and outliers (faults).
        - Use **DBSCAN** and **HDBSCAN** to identify dense regions (normal states) and outliers (potential faults).
        - Tune eps and min_samples (or use HDBSCAN which adapts automatically).
        - Visualize clusters; highlight outliers (label = -1) in black or red.
        - Evaluate overlap with true labels to see if DBSCAN isolates faults.
    - Visualize results in 2D using PCA.
    - Highlight detected noise points and evaluate correspondence with true fault labels.

Deliverables:

1. 2D scatter plots for K-Means and HDBSCAN clusters (2pts.)
2. Cluster vs label confusion matrix or purity score (2pts.)
3. Short report (~300 words) interpreting whether clusters correspond to normal/faulty operation, and how density-based clustering identifies unusual states. (1 pts.)

Optional task: If you like, you can experiment using the above techniques of clustering for fault detection in the following different unlabeled dataset (is a large file, you may need to down sample before using): TEP dataset