Analysis of Electronic Nose (E-Nose) Sensor Data using Python in Experimental Physics

Henri Setyo Pambudi

Overview

This Python code is designed for analyzing electronic nose (e-nose) sensor data to characterize scents. The code processes and analyzes data from an e-nose system, which uses multiple sensors to detect and identify different scent characteristics. The dataset contains raw sensor readings over time, which are preprocessed and visualized for further analysis.

Code Explanation

1. Mounting Google Drive and Importing Libraries

- Mounts Google Drive to access data files.
- Imports essential libraries like `pandas`, `matplotlib`, `seaborn`, `plotly`, and `scikit-learn` for data manipulation, visualization, and machine learning.

2. Loading and Inspecting Data

- Reads a CSV file (`DATA_ENOSE.csv`) containing raw e-nose sensor data.
- Displays the dataset for initial inspection.

3. Data Cleaning

- Drops unnecessary columns (`Time (s)` and `Proses`) to focus on sensor data.
- Displays the cleaned dataset.

4. Raw Data Visualization

- Plots the raw sensor data to observe trends over time.

5. Dropping Unnecessary Sensors

- Removes `Sensor 5`, assuming it is irrelevant or redundant for the analysis.
- Plots the data without `Sensor 5` to verify the updated dataset.

6. Baseline Normalization

- Normalizes the baseline for each sensor by subtracting the minimum value within the first 10 readings.
- Ensures all sensors start from a uniform baseline for better comparability.
- Visualizes the normalized data.

7. Labeling Data

- Adds a new column (`label`) to the dataset, initialized to 0, potentially for classification purposes in future steps.

8. Data Normalization

- Uses `StandardScaler` to standardize the sensor readings, ensuring all features have a mean of 0 and a standard deviation of 1.
- Prepares the data for Principal Component Analysis (PCA).

9. Principal Component Analysis (PCA)

- **Eigenvectors and Eigenvalues:** Computes eigenvectors and eigenvalues to determine the principal components and displays the eigenvalues (variance explained by each component).
- **Scree Plot:** Visualizes eigenvalues against component numbers to determine the optimal number of components to retain.
- **2D PCA Visualization:** Projects the data onto 2 principal components and plots it in a 2D scatter plot.
- **3D PCA Visualization:** Projects the data onto 3 principal components and visualizes it in a 3D scatter plot using `plotly`. The total variance explained by the first 3 components is displayed in the plot title.

Purpose of the Code

1. **Data Preprocessing:** Prepares raw e-nose sensor data for analysis by cleaning, normalizing, and visualizing. Removes irrelevant features and normalizes data to improve the reliability of machine learning algorithms.

2. **Principal Component Analysis (PCA):** Reduces the dimensionality of the sensor data while preserving the variance. Identifies the most important features (principal components) contributing to scent characteristics.

3. **Visualization:** Provides insights into the variance explained by principal components and visualizes the data distribution in lower-dimensional spaces (2D and 3D) for easier interpretation.

4. **Scalable Analysis:** Lays the foundation for further classification, clustering, or pattern recognition tasks using e-nose data.

Applications

- Scent profiling and classification in food, beverages, or fragrances.
- Detection of hazardous gases or pollutants.
- Quality control in manufacturing processes.

Conclusion