

# README

## Project Overview

This project focuses on loading, processing, and visualizing sensor data, followed by training a machine learning model to classify the data. The workflow includes loading historical sensor data for training and evaluation, and then using the trained model to make predictions on the latest sensor data.

## Steps Explanation

### 1. Library Imports

We import various libraries necessary for data manipulation, numerical operations, data visualization, and machine learning. Libraries include `pandas` for data manipulation, `numpy` for numerical operations, `matplotlib` and `seaborn` for visualization, and `sklearn` for machine learning tasks.

### 2. Logging Setup

Logging is configured to capture and report events during code execution. This helps in debugging and tracking the progress of the code execution.

### 3. Loading Historical Data

The historical sensor data is loaded from a CSV file. Exception handling is implemented to manage potential errors such as the file not being found or the file being empty. If an error occurs, an empty DataFrame is initialized as a fallback.

### 4. Displaying Data

The first few rows of the historical data are displayed if the data is loaded successfully. This helps in verifying the contents of the loaded data.

### 5. Preparing Data for Model Training

The data is prepared for model training by separating the features (X) and the target variable (y). The data is then standardized using `StandardScaler` to ensure that all features contribute equally to the model. The standardized data is split into training and testing sets.

### 6. Plotting Training and Testing Data

The training and testing data points are visualized using scatter plots. This helps in understanding the distribution and separation of different classes in the data.

### 7. Model Training

The `Gradient Boosting Classifier` is initialized and trained on the training data. This model replaces the previously used `Gaussian Process Classifier` for better performance and accuracy.

## 8. Making Predictions

The trained model is used to make predictions on the testing data. The predictions are evaluated using metrics such as accuracy, confusion matrix, and a detailed classification report. The results are logged and printed for review.

## 9. Loading and Predicting Latest Data

The latest sensor data is loaded from a CSV file, standardized, and predictions are made using the trained model. If the latest data file is not found or is empty, appropriate error handling is in place.

## 10. Plotting Latest Data Predictions

The density plot for the latest data predictions is visualized using `seaborn`. This step involves plotting the predicted classes using KDE plots and scatter plots to show the distribution and classification of the latest sensor data.

## Key Changes

- **Comments:** Added throughout the code to explain each step.
- **Seaborn Library:** Used for enhanced data visualization, providing better plots and insights.
- **Model Update:** The `Gaussian Process Classifier` has been replaced with the `Gradient Boosting Classifier` for improved accuracy and performance.