**Title:** Predictive Maintenance in Chemical Plants Using Machine Learning

**Name:** Prateek Sachan  
**Roll No:** 220107065

**Introduction**

**In the chemical industry, unplanned equipment failures can lead to hazardous situations, increased downtime, and loss of productivity. Traditional preventive maintenance methods are based on fixed schedules rather than actual equipment conditions, often resulting in inefficient resource usage. Predictive maintenance, powered by artificial intelligence and machine learning (AI/ML), offers a data-driven alternative that forecasts equipment failures before they occur, thereby enhancing safety, reducing costs, and improving operational efficiency.**

**Objectives**

* **Develop a predictive maintenance model to detect potential equipment failures using sensor data.**
* **Compare machine learning models such as Random Forest, Logistic Regression, and XGBoost for failure prediction.**
* **Evaluate model performance using accuracy, precision, recall, and F1-score.**
* **Propose a framework for integrating the predictive model into industrial control systems.**

**Problem Statement**

**To design a robust predictive maintenance system for chemical plant machinery by leveraging real-time sensor data, aiming to reduce downtime and improve operational reliability compared to traditional preventive maintenance approaches.**

**Significance of the Study**

* **Enhances equipment reliability and plant safety.**
* **Reduces maintenance costs by predicting failures before they occur.**
* **Optimizes resource allocation and maintenance scheduling.**
* **Supports data-driven decision-making in plant operations.**

**Data Description and Preprocessing**

**Dataset: AI4I 2020 Predictive Maintenance Dataset (**[**Available at UCI Repository**](https://archive.ics.uci.edu/dataset/601/ai4i%2B2020%2Bpredictive%2Bmaintenance%2Bdataset)**)  
Features: Air temperature, Process temperature, Rotational speed, Torque, Tool wear, Machine failure label.  
Preprocessing Steps:**

* **Handle missing or duplicate records (if any).**
* **Normalize sensor readings for consistent scale.**
* **Encode categorical variables (if present).**
* **Visualize feature distributions and correlations.**

**Methodology Flowchart**

***Data Collection → Data Cleaning → Feature Engineering → Model Training → Model Evaluation → Deployment Strategy***

**Model Selection & Rationale**

* **Random Forest: Good at handling non-linearities and feature importance.**
* **Logistic Regression: Baseline model for classification.**
* **XGBoost: Efficient and accurate for tabular sensor data.**

**Model Training & Validation**

* **Split data into training and testing sets (80:20).**
* **Perform hyperparameter tuning using grid search.**
* **Use stratified K-fold cross-validation.**
* **Metrics: Accuracy, Precision, Recall, F1-score.**

**Evaluation Metrics**

* **Accuracy: Overall correctness of the model.**
* **Precision: Correctly predicted failures among all predicted failures.**
* **Recall: Correctly predicted failures among all actual failures.**
* **F1-Score: Harmonic mean of precision and recall.**

# Challenges and Solutions Explored

Some challenges encountered included preprocessing inconsistencies such as handling missing values and normalizing sensor data. Model selection was another area requiring careful consideration due to imbalanced data. These were addressed using standard preprocessing techniques and basic evaluation strategies.

# Next Steps

Future steps include optimizing the selected models through hyperparameter tuning, exploring additional machine learning algorithms, and implementing deployment solutions to integrate the predictive maintenance system with a real-time monitoring setup. Evaluation will be expanded to assess model robustness and performance in a production-like environment.

# Appendix: Jupyter Notebook

The full code implementation for data preprocessing and model training is provided in the accompanying Jupyter Notebook.  
 Click here to open the notebook file: Predictive\_Maintenance\_Report\_220107065.ipynb