**1. Project Overview**

**a. Project Title:**  
**Predictive Maintenance using Vibration and Temperature Data**

**b. Project Description:**  
This project aims to predict whether an industrial machine requires maintenance using sensor data such as temperature, vibration, motor current, and operating conditions. The system applies data preprocessing, visual analysis, and machine learning (Random Forest) to classify whether maintenance is required (1) or not (0). The result is an interactive tool that can support predictive maintenance planning in manufacturing.

**c. Timeline:**

* Week 1–2: Data understanding and cleaning
* Week 3: Exploratory Data Analysis
* Week 4: Model building and evaluation
* Week 5: Streamlit app development
* Week 6: Report and presentation creation

**d. Benefits:**

* Reduced machine downtime
* Improved maintenance scheduling
* Cost-effective fault prediction
* Reusable ML model for deployment

**e. Team Members:**  
Tisha Sachwani

Manan Vasant Uttekar

Krunal Mehta

**f. Risks:**

* Limited sensor variety
* Model overfitting on small datasets
* Feature drift in real-time data
* Deployment compatibility across environments

**2. Objectives**

**a. Primary Objective:**  
To predict machine maintenance requirements using sensor-based data and machine learning classification.

**b. Secondary Objectives:**

* Visualize trends and correlations in sensor data
* Build a reusable and accurate predictive model
* Provide a user interface for real-time predictions

**c. Measurable Goals:**

* Achieve ≥ 90% accuracy on test set
* Deliver a functioning Streamlit app
* Complete end-to-end pipeline from data to deployment

**3. Methodology**

**a. Approach:**  
An iterative, data-driven approach following CRISP-DM principles. Used Python-based tools for data analysis, modeling, and app development.

**b. Phases:**

1. Data Collection & Cleaning
2. EDA & Feature Engineering
3. Model Training & Testing
4. Deployment Interface Creation

**c. Deliverables:**

* Cleaned dataset
* Visualizations (.png files)
* Trained .pkl model
* Streamlit app (app.py)
* Final report and presentation

**e. Testing and Quality Assurance:**

* Train-test split validation
* Accuracy, confusion matrix, and classification report used
* Manual app testing with sample inputs

**f. Risk Management:**

* Encoded categorical features consistently
* Ensured column order matches during prediction
* Streamlit UI tested for real-time compatibility

**4. Technologies Used**

**a. Programming Languages:**

* Python 3.8

**b. Development Frameworks:**

* Scikit-learn, Streamlit

**c. Database Management Systems:**

* Not applicable (data in Excel/CSV)

**d. Development Tools:**

* VS Code, Jupyter Notebook, GitHub

**e. Testing Tools:**

* Built-in sklearn metrics (accuracy, confusion matrix)

**f. Cloud Services:**

* (Optional: Streamlit Cloud for deployment — if hosted)

**g. Security:**

* Local .pkl model used; no external API integration

**h. APIs and Web Services:**

* Not applicable

**5. Results**

**a. Key Metrics:**

* Accuracy: ~91%
* Precision & Recall: Balanced for both classes
* Model generalized well on unseen test data

**b. ROI:**  
The app provides a low-cost and quick way to detect faults, reducing unscheduled maintenance and extending equipment life. Suitable for real-time applications in smart factories.

**6. Conclusion**

**a. Recap the Project:**  
The goal of this project was to predict maintenance requirements for industrial machines based on sensor inputs. The outcome is a complete ML pipeline from data to deployment.

**b. Key Takeaways:**

* Sensor data carries meaningful patterns
* Feature order matters in .pkl model prediction
* Streamlit is effective for quick app deployment

**c. Future Plans:**

* Add time-series component with LSTM
* Connect live sensor feed
* Deploy on the cloud for remote use

**d. Successes and Challenges:**

* Successfully completed EDA, modeling, and deployment
* Faced model prediction mismatch due to column ordering
* Solved using model.feature\_names\_in\_ technique

**7. Project Specifics**

**a. Project URL:**  
*(Optional: Streamlit Cloud or localhost URL if deployed)*

**b. GitHub URL:**  
<https://github.com/Tisha-S7/Predictive-Maintenance-Project> *(if available)*

**c. Collab/Notebook URL:**  
*(Optional: if any notebook used)*

**d. Dataset URL:**  
*(Optional: if shared via Drive or Kaggle)*