

# Installation and Deployment Guide: AI-Powered Predictive Maintenance for Industrial Machines

This document provides a step-by-step guide on how to install dependencies, set up the environment, train the model, and run the AI-Powered Predictive Maintenance system.

## 1. System Requirements

Before proceeding, ensure your system meets the following requirements:

### Hardware Requirements:

- CPU: Intel Core i5 or higher (Recommended: i7 or Ryzen 5+)
- RAM: Minimum **8GB** (Recommended: 16GB)
- Storage: At least **2GB free space**

### Software Requirements:

- Operating System: **Windows 10/11, Ubuntu 20.04+, macOS Monterey+**
- Python: **Version 3.10 or 3.11** (Avoid 3.12 due to compatibility issues)
- Required Libraries: Listed below

## 2. Installation of Dependencies

### Step 1: Install Python

1. Download **Python 3.10+** from [Python Official Website](https://www.python.org/downloads/).
2. During installation, **check "Add Python to PATH"** to avoid path-related issues.
3. Verify installation using:
4. `python --version`

### Step 2: Clone or Download the Project

- If using **Git**, run:
- `git clone https://github.com/your-repository/AI-Predictive-Maintenance.git`
- `cd AI-Predictive-Maintenance`
- If downloaded as a **ZIP file**, extract it and navigate into the folder.

### Step 3: Create a Virtual Environment

```
python -m venv venv
```

Activate the virtual environment:

- **Windows:**
- `venv\Scripts\activate`
- **Linux/macOS:**
- `source venv/bin/activate`

#### **Step 4: Install Required Python Libraries**

`pip install -r requirements.txt`

If requirements.txt is missing, install manually:

`pip install numpy pandas scikit-learn streamlit matplotlib seaborn pickle5`

#### **Step 5: Verify Installation**

Run the following command to ensure all dependencies are installed:

```
python -c "import sklearn, streamlit, pandas, numpy; print('All Dependencies Installed Successfully')"
```

### **3. Model Training & Preparation**

#### **Step 1: Generate the Dataset**

If the dataset is missing or needs re-creation, run:

```
python generate_dataset.py
```

This will create synthetic\_machine\_data.csv.

#### **Step 2: Train the Model**

To train the predictive model, execute:

```
python train_model.py
```

This will generate:

- `predictive_model.pkl` → Trained ML model
- `scaler.pkl` → Feature scaler for normalization
- `label_encoder.pkl` → Encodes fault types

#### **Expected Output:**

Model Training Complete  - Accuracy: 0.85

## **4. Running the Predictive Maintenance System**

### **Option 1: Run the Model Manually (CLI-Based Testing)**

To test the model using sample inputs:

```
python check.py
```

You will be prompted to enter machine parameters, and the model will classify the fault.

### **Option 2: Run the Dashboard (GUI-Based Interface)**

For an interactive web-based dashboard:

```
streamlit run dashboard.py
```

This will open the AI-Powered Predictive Maintenance dashboard in your default web browser.

## **5. Understanding the Dashboard**

The dashboard provides:

- ✓ Input fields for **vibration, temperature, pressure, and running hours**
- ✓ **Real-time fault classification**
- ✓ **Fault probability distribution**
- ✓ **Historical predictions table**
- ✓ **Visualizations for fault trends**

## **6. Deployment Options**

### **Local Execution (Recommended for Testing)**

Simply run:

```
streamlit run dashboard.py
```

Access it via `http://localhost:8501/`.

### **Cloud Deployment (Streamlit Share)**

To deploy on Streamlit Share:

1. Push the project to GitHub.
2. Go to [Streamlit Cloud](#).
3. Deploy using `dashboard.py`.

## 7. Troubleshooting Common Issues

Issue	Possible Cause	Solution
ModuleNotFoundError: No module named 'sklearn'	Dependencies missing	Run <code>pip install -r requirements.txt</code>
ValueError: X has 5 features, but StandardScaler is expecting 4	Incorrect input formatting	Ensure input features match trained model
pickle.UnpicklingError	Corrupted model files	Delete .pkl files and retrain the model
Streamlit app not opening	Port conflict	Try <code>streamlit run dashboard.py --server.port 8502</code>

## 8. Final Steps & Best Practices

- ✓ Always **train the model before running the dashboard**.
- ✓ **Do not modify the dataset** after training unless re-training is needed.
- ✓ Keep `predictive_model.pkl`, `scaler.pkl`, and `label_encoder.pkl` safe.
- ✓ For **production deployment**, use a **Docker container or cloud-based solution**.