Predictive Analysis for Manufacturing Operations

Objective

To assess your ability to:

1. Build a simple predictive analysis model for manufacturing data.
2. Create RESTful API endpoints to upload data, train a model, and make predictions.

Scenario

You are tasked with creating a RESTful API to predict **machine downtime** or **production defects** using a manufacturing dataset. The API should include endpoints for basic operations and return predictions for decision-making.

Assignment Details

1. **Dataset**:
    - Use any small manufacturing-related dataset (e.g., Machine Downtime or Product Defects). Publicly available options include Kaggle or UCI ML Repository. If unavailable, generate synthetic data with key columns like `Machine_ID`, `Temperature`, `Run_Time`, `Downtime_Flag`.
2. **Model**:
    - Use a simple supervised ML model (e.g., Logistic Regression, Decision Tree).
    - Train the model on provided or uploaded data to predict machine downtime or product defects.
3. **Endpoints**:
    - **Upload Endpoint** (`POST /upload`): Accept a CSV file containing manufacturing data (e.g., `Machine_ID`, `Temperature`, `Run_Time`).
    - **Train Endpoint** (`POST /train`): Train the model on the uploaded dataset and return performance metrics (e.g., accuracy, F1-score).
    - **Predict Endpoint** (`POST /predict`): Accept JSON input (e.g., `{"Temperature": 80, "Run_Time": 120}`) and return the prediction (e.g., `Downtime: Yes/No`).
4. **Output Format**:
    - **Predictions**: Return results in JSON format.

    - Example Response: `{ "Downtime": "Yes", "Confidence": 0.85 }`.
5. **Technical Requirements**:
    - Use **Python** with **Flask** or **FastAPI** for the API.
    - Use **scikit-learn** for the model.
    - Test endpoints locally using **Postman** or **cURL**.
6. **Documentation**:
    - Include a **README** with:
        - Instructions to set up and run the API.
        - Example API requests and responses.

Evaluation Criteria

1. **Functionality**: Are the endpoints working as expected?
2. **Relevance**: Does the model provide meaningful predictions for manufacturing use cases?
3. **Code Quality**: Is the code clear and well-structured?
4. **Delivery**: Can the project be set up and tested in minimal time?

Expected Deliverables

1. A GitHub repository containing:
    - Codebase.
    - A sample dataset.
    - README file with setup and usage instructions.
2. A working API that can be tested locally.

Deadline

Submit within **3 days** from assignment receipt.

Good luck, and we look forward to seeing your implementation!