Predictive Maintenance Machine Learning Project Proposal
Completed January–March 2023
Author: Goitom Abirha

Project Overview

- Build a predictive maintenance model that detects equipment failures 72 hours before they occur.
- Use simulated IoT sensor data stored in Azure SQL.
- Demonstrate end-to-end ML workflow: simulation, SQL integration, feature engineering, model training, evaluation, documentation.

Purpose

- Modern machines generate continuous IoT sensor streams containing early indicators of degradation.
- This project identifies pre-failure patterns to enable proactive maintenance.

Project Focus

- IoT-style data simulation
- Azure SQL cloud integration
- Time-series feature engineering
- Binary classification model development
- Optional Flask API for predictions

Specific Goals

- Simulate high-quality IoT sensor data for 12 months.
- Store and query data using Azure SQL.
- Engineer rolling-window and trend-based features.
- Create 72-hour failure label (y_failure_72h).
- Train machine learning models (LR, RF, Gradient Boosting).
- Evaluate model performance using ROC-AUC, precision, recall, F1.
- Publish professional GitHub repository.

Expected Outcomes

- A complete and polished GitHub repository.

- Trained predictive maintenance model.
- Documentation, diagrams, and evaluation charts.
- Azure SQL integration demonstration.

Project Objective and Scope

- Develop a predictive maintenance system that accurately predicts failures within a 72-hour window using cloud-based IoT data.

Data Description

- Simulated IoT sensor dataset replicating industrial conditions.
- sensor_readings.csv – temperature, vibration, pressure, rpm, current, status_code.
- failures.csv – failure events and types.
- dataset_ready_for_model.csv – engineered features + labels.
- Simulated data chosen for realism, scalability, and control.

Exploratory Data Analysis (EDA)

- Sensor distribution and summary statistics.
- Rolling trend visualization.
- Correlation analysis.
- Pre-failure drift patterns.
- Class imbalance review.

Data Preparation

- Timestamp validation and cleaning.
- Data merging for sensor + failure alignment.
- Rolling window features: 6h, 12h, 24h.
- Trend features: deltas and gradients.
- 72-hour failure labeling.
- Time-based train/test split.

Model Training Approach

- Models: Logistic Regression, Random Forest, Gradient Boosting.
- Handle imbalance using class weights.
- Hyperparameter tuning with GridSearchCV.
- Cross-validation and feature importance analysis.

Model Evaluation

- Metrics: ROC-AUC, Precision, Recall, F1.

- Confusion matrix and error analysis.
- Precision@Top-K to simulate maintenance prioritization.

Optional User Interface

- Build a Flask/FastAPI endpoint for real-time predictions.

Capstone Complexity

- Cloud integration with Azure SQL.
- Synthetic data generation with realistic patterns.
- Advanced feature engineering.
- ML modeling + deployment readiness.

8-Week Timeline (Completed January–March 2023)

Week 1 – Project Setup

- Create GitHub repository and folder structure.
- Define project documentation sections.
- Prepare virtual environment and dependencies.

Week 2 – Data Simulation

- Develop IoT data simulation logic.
- Generate 12 months of sensor data.
- Simulate failures with realistic drift.
- Save sensor_readings.csv and failures.csv.

Week 3 – Azure SQL Integration

- Create Azure SQL database and tables.
- Upload simulated data into SQL.
- Test queries and Python connection.

Week 4 – Feature Engineering

- Load data from Azure SQL.
- Generate rolling windows (6h, 12h, 24h).
- Create trend/delta features.
- Generate y_failure_72h label.

Week 5 – Exploratory Data Analysis

- Visualize distributions and trends.
- Analyze pre-failure behavior.
- Document insights and findings.

Week 6 – Model Training

- Train baseline model (Logistic Regression).
- Train RF and Gradient Boosting.
- Tune hyperparameters and address imbalance.

Week 7 – Model Evaluation

- Evaluate ROC-AUC, Precision, Recall, F1.
- Create confusion matrix and feature importance.
- Save final model.pkl.

Week 8 – Documentation & Publication

- Finalize notebooks and code.
- Add diagrams and EDA charts.
- Complete README.md.
- Upload project plan into /docs folder.
- Polish and publish GitHub repository.

Resources

- Azure SQL Documentation
- Scikit-learn Documentation
- Predictive Maintenance Research Papers
- Kaggle IoT Datasets (for reference)