**📝 Software Requirements Specification (SRS-3)**

**Project Title:** Predictive Maintenance of Machines and Industrial Equipment  
**Prepared by:** Sparsh Srivastava  
**Date:** 01 November 2025

**1. 📌 Introduction**

**1.1 Purpose**

This project aims to develop a machine learning-based predictive maintenance system that can anticipate potential failures or maintenance needs of industrial machinery using sensor data.

**1.2 Scope**

The system will:

* Analyze sensor data such as Torque, Rotational Speed, Process Temperature, Air Temperature, and Tool Wear.
* Classify machine states into risk categories.
* Minimize unplanned downtime and maintenance costs.
* Enable industries to shift from reactive/scheduled maintenance to predictive strategies.

**1.3 Definitions, Acronyms, and Abbreviations**

* **ML:** Machine Learning
* **EDA:** Exploratory Data Analysis
* **CSV:** Comma-Separated Values
* **SRS:** Software Requirements Specification

**2. 📚 Overall Description**

**2.1 Product Perspective**

This is a standalone predictive analytics system built using Python and Jupyter Notebook. It uses a modular pipeline for preprocessing and classification, and is deployable in industrial environments.

**2.2 Product Functions**

* Data ingestion from CSV files
* Preprocessing of numerical and categorical features
* Classification using Logistic Regression
* Hyperparameter tuning via GridSearchCV
* Exporting trained model as .pkl file

**2.3 User Characteristics**

* Industrial engineers, data scientists, maintenance planners
* Familiarity with basic ML concepts and Python
* Expectation: Accurate, interpretable, and fast predictions

**2.4 Constraints**

* Requires clean sensor data
* Limited to classification-based prediction
* Model performance depends on quality of training data

**3. ✅ Specific Requirements**

**3.1 Functional Requirements**

* **FR1:** The system shall preprocess numerical features using mean imputation and standard scaling.
* **FR2:** The system shall preprocess categorical features using mode imputation and one-hot encoding.
* **FR3:** The system shall classify machine states using Logistic Regression.
* **FR4:** The system shall tune hyperparameters using GridSearchCV.
* **FR5:** The system shall export the best model as logreg\_pipeline.pkl.

**3.2 Non-Functional Requirements**

* **NFR1:** The model shall achieve at least 99% accuracy on validation data.
* **NFR2:** The pipeline shall be modular and interpretable.
* **NFR3:** The system shall be deployable in production environments.

**3.3 External Interface Requirements**

* **User Interface:** Jupyter Notebook
* **Software Interface:** Python (scikit-learn, pandas, joblib)
* **Hardware Interface:** Standard computing environment with Python installed

**4. 🧠 System Features**

| **Feature** | **Description** |
| --- | --- |
| Data Preprocessing | Handles missing values and scales features |
| Classification | Logistic Regression with L1/L2 regularization |
| Hyperparameter Tuning | Uses GridSearchCV with StratifiedKFold |
| Model Export | Saves trained pipeline as .pkl file |
| Interpretability | Enables understanding of feature impact |

**5. 🛠️ Design Approach**

* **Architecture:** Modular ML pipeline using ColumnTransformer and GridSearchCV
* **Model Type:** Logistic Regression
* **Tools Used:** Python, scikit-learn, pandas, joblib
* **Dataset:** Sensor data with operational parameters and tool wear

**6. 📤 Future Enhancements**

* Integration with real-time sensor streams
* Support for multi-class classification
* Deployment via REST API or cloud service
* Visualization dashboards for maintenance teams