Software Requirements Specification (SRS)

Project Title: Cancer Prediction Model – End-to-End ML Lifecycle

---

1. Introduction

1.1 Purpose

This document defines the requirements for developing, deploying, and monitoring a **Cancer Prediction Model** that can classify patients into risk categories (e.g., cancerous vs. non-cancerous) or predict tumor progression probability. The system uses historical patient data, medical test results, and imaging-derived metrics to make predictions.

1.2 Scope

The Cancer Prediction Model will:

- Support **data collection** from medical records or public datasets (e.g., Breast Cancer Wisconsin Dataset).
- Perform **data preprocessing** (handle missing medical test values, remove noise).
- Enable **feature engineering** (derive tumor-related metrics, correlations with diagnosis).
- Select and train models (Classification for diagnosis, Regression for progression rate).
- Evaluate models using medical accuracy standards.
- Deploy for real-time or batch predictions in healthcare settings.
- Continuously monitor performance and integrate **doctor/clinician feedback**.

1.3 Definitions

- **TNM Staging** – Tumor, Node, Metastasis classification for cancer.
- **Benign** – Non-cancerous growth.
- **Malignant** – Cancerous growth.
- **Sensitivity/Recall** – The ability to correctly identify patients with cancer.

---

2. Problem Statement and Background

2.1 Problem Statement

Early detection of cancer significantly improves treatment outcomes, but manual diagnosis based solely on lab results and scans can be time-consuming and prone to human error.
Challenges include:

- Missing or incomplete patient records.
- High-dimensional medical data with irrelevant features.
- Need for models that generalize well to unseen patient cases.

2.2 Background

Machine learning has proven effective in cancer detection by identifying subtle patterns in patient data that are difficult for humans to detect. By combining **data preprocessing, feature engineering, and model optimization**, the Cancer Prediction Model aims to improve diagnostic accuracy and assist oncologists in decision-making.

---

3. Overall Description

3.1 Product Perspective

The system will function as a **medical decision support tool**, not a replacement for professional diagnosis. It integrates:

1. **Data Preparation** – Collecting patient data from hospital systems or open datasets.
2. **Data Preprocessing** – Handling missing test results, removing noisy measurements.
3. **Feature Engineering** – Calculating tumor size ratios, genetic markers, and patient risk scores.
4. **Model Selection & Training** – Choosing classification models for detection (Logistic Regression, Random Forest, XGBoost) or regression models for tumor progression prediction.
5. **Evaluation** – Using medical metrics (Accuracy, Recall, ROC-AUC, F1-score).
6. **Deployment & Monitoring** – API or dashboard for hospitals; periodic retraining with new patient data.

---

4. Specific Requirements

4.1 Functional Requirements

ID	Requirement	Description
FR1	Data Collection	Import data from hospital databases or medical datasets.
FR2	Data Preprocessing	Clean, normalize, and impute missing values in patient records.
FR3	Feature Engineering	Create derived features (tumor size ratios, age-risk factors).
FR4	Model Selection	Select classification models for diagnosis.
FR5	Model Training	Train models on labeled medical data.
FR6	Model Evaluation	Evaluate using recall, precision, ROC-AUC.
FR7	Hyperparameter Tuning	Optimize model parameters for maximum accuracy.
FR8	Deployment	Provide API or UI for doctors to upload patient data.
FR9	Monitoring	Track accuracy over time and detect model drift.

ID	Requirement	Description
FR10	Feedback Loop	Incorporate doctor feedback to improve predictions.

4.2 Non-Functional Requirements

- **Accuracy:** Must exceed 90% recall to minimize false negatives.
- **Security:** Comply with HIPAA and healthcare data privacy standards.
- **Performance:** Real-time prediction latency under 2 seconds.
- **Scalability:** Handle 10,000+ patient records.
- **Usability:** Clear, non-technical output for doctors.

---

5. Assumptions, Constraints, and Dependencies

Assumptions

- Historical patient datasets are available and labeled.
- Doctors will validate predictions before using them clinically.

Constraints

- Strict compliance with medical data privacy regulations.
- Limited computing resources in hospital systems.

Dependencies

- Python libraries: Pandas, NumPy, Scikit-learn, XGBoost.
- Cloud infrastructure for secure data storage and model hosting.