

# Patient Risk Prediction System — Final Submission (v2)

Prepared by: SHAIK MOHAMMAD ELIYAZ

Date: October 31, 2025

## Executive Summary

1-Page Summary: Patient Risk Prediction System

**Problem Identification** Hospitals face high patient readmission rates (up to 20% within 30 days), leading to increased costs, resource strain, and poorer patient outcomes. Key issues include delayed interventions due to manual monitoring of patient records, lab results, and vitals, which often misses early warning signs of deterioration. This results in preventable readmissions, higher mortality, and inefficient healthcare delivery.

**Solution Design** We propose an end-to-end AI/ML system to predict patient readmission risk using historical data, enabling proactive alerts to doctors.

**Data Pipeline:** Collect patient records (demographics, history), lab results, and vitals from EHR systems. Store raw data in AWS S3 for scalability. Preprocess by handling missing values (imputation), normalizing features (e.g., scaling vitals), and feature engineering (e.g., aggregating time-series data). Use AWS Glue for ETL automation.

# Architecture Diagram

Architecture diagram not found.

## Solution Design & Details

**Model:** Train a supervised model (XGBoost for its handling of tabular data and interpretability) on labeled readmission data. Random Forest as a baseline alternative. Evaluate using AUC-ROC and precision-recall metrics.

**Cloud Integration:** Use AWS SageMaker for model training and inference. Store processed data and predictions in AWS RDS (relational database for structured queries). Integrate with AWS Lambda for serverless preprocessing.

**Deployment Strategy:** Deploy the model as a REST API using Flask, containerized with Docker and hosted on AWS Elastic Beanstalk. Build a user-friendly dashboard with Streamlit for doctors to view risk scores, alerts, and patient timelines. Enable real-time predictions via API calls from EHR systems.

**Monitoring:** Implement AWS CloudWatch for alerts on model drift, latency, and errors. Track KPIs like prediction accuracy and alert response times. Use AWS X-Ray for tracing API performance.

### Outcomes

**Reduced Readmissions:** Early alerts could lower rates by 15-25%, saving \$1-2B annually in U.S. healthcare costs.

**Improved Interventions:** Doctors receive automated notifications, enabling timely care (e.g., medication adjustments), reducing mortality by 10-15%.

**Efficiency Gains:** Automates manual monitoring, freeing staff for patient care. Scalable to multiple hospitals with ROI in 6-12 months.

**Ethical Considerations:** Ensure HIPAA compliance, model fairness (bias audits), and explainability (SHAP values for predictions).

## Appendix: Generated Files

Files generated:

- patient\_risk\_all\_in\_one\_v2.pdf (this file)
- patient\_risk\_all\_in\_one.pdf (previous version)
- patient\_risk\_submission.zip (contains sample app, diagram, summary)

Next steps I can perform:

1. Prepare a GitHub-ready repo structure and provide exact git commands & README.
2. Generate a PPTX slide deck from this PDF.
3. Create export-ready images or split the PDF into separate deliverables.

I cannot directly upload to Google Drive or push to GitHub, but I will provide step-by-step commands to do so.