Patient Readmission Prediction using Databricks Lakehouse Architecture

A Capstone Project Report

Submitted as part of
Code basics × Databricks 14-Day Data & AI Challenge

Domain: Healthcare & Life Sciences
Dataset: Diabetic Patient Readmission Dataset
Platform & Tools Used
Databricks | Delta Lake | PySpark | SQL | MLflow | Unity Catalog

Submitted by
Pujitha Pakala

Submission Date
31 January 2026

Mentors / Organizers
Code basics • Databricks • Indian Data Club

Patient Readmission Prediction using Databricks Lakehouse Architecture

Table of Contents

Contents

1.Introduction 4

2.Problem Definition & AI Framing 4

3. Business Context & Healthcare Relevance 5

4: Dataset Overview & Data Understanding 6

5: Databricks Lakehouse Architecture Overview 7

6. Bronze Layer – Raw Data Ingestion 8

7. Silver Layer – Data Cleaning & Transformations 9

8. Gold Layer – Business & ML-Ready Data 9

9. Feature Engineering Strategy 10

10. Analytics & Dashboard Insights 11

11. Machine Learning Approach 13

11.1 Problem Formulation 13

11.2 Choice of Input Data (Gold Layer) 14

11.3 Feature Representation Strategy 14

11.4 Model Selection Strategy 14

11.5 Evaluation Focus: Recall-Oriented Strategy 15

11.6 Training–Inference Consistency 15

12. Model Training, Evaluation & Metrics 15

12.1 Training Setup and Pipeline Design 15

12.2 Logistic Regression Training 16

12.3 Threshold Tuning for Recall-Oriented Prediction 16

12.4 Metrics Computation (Test Data) 16

12.5 Logistic Regression Results 17

12.6 Random Forest Training and Results 17

12.7 Model Comparison and Selection 17

12.8 Notes on Class Imbalance and Practical Evaluation 18

13. MLflow Experiment Tracking 18

13.1 Purpose of MLflow in this Project 18

13.2 Experiment Logging Strategy 18

13.3 Logistic Regression Run Tracking 18

Patient Readmission Prediction using Databricks Lakehouse Architecture

13.4 Random Forest Run Tracking 19

13.5 Model Comparison and Best Model Selection 19

13.6 Notes on MLflow Warnings (Non-blocking) 20

14. End-to-End Database ↔ AI Workflow 20

15.Governance, Security & Data Management 21

16.Orchestration & Workflow Automation 22

17. Business Impact & Practical Use 23

17.1 Business Impact 23

17.2 Practical Use Cases in Healthcare Operations 23

17.3 Alignment with Real-World Data Platforms 23

18. Challenges Faced & Key Learnings 24

18.1 Key Challenges Faced 24

18.2 Key Learnings 24

19. Conclusion & Future Enhancements 25

19.1 Conclusion 25

19.2 Future Enhancements 26

Patient Readmission Prediction using Databricks Lakehouse Architecture

1.Introduction

Healthcare systems across the world are continuously challenged by increasing patient volumes, rising operational costs, and the need to deliver high-quality patient care. One of the major contributors to increased healthcare costs and resource utilization is **unplanned hospital readmissions**, especially within a short period after patient discharge. Patient readmission not only increases financial burden on healthcare providers but also indicates potential gaps in treatment effectiveness, discharge planning, or post-hospitalization care.

With the rapid digitization of healthcare systems, large volumes of patient data are generated every day in the form of electronic health records (EHRs), admission logs, clinical procedures, and medication histories. When analysed effectively, this data can provide valuable insights into patient behavior, disease progression, and risk factors associated with hospital readmissions. However, traditional rule-based approaches often fail to capture complex relationships among multiple variables present in healthcare datasets.

This project focuses on building a **Patient Readmission Prediction system** using the **Databricks Lakehouse Architecture**, leveraging modern data engineering and machine learning capabilities. The goal of the project is to design an end-to-end data pipeline that ingests raw healthcare data, processes it through structured transformation layers, performs analytical exploration, and finally applies machine learning techniques to predict the likelihood of patient readmission.

The **Databricks Lakehouse platform** is used as the core technology for this project as it enables unified data storage, scalable data processing, advanced analytics, and machine learning on a single platform. By combining **Delta Lake**, **Apache Spark**, **SQL analytics**, **MLflow**, and **Unity Catalog**, the Lakehouse architecture allows seamless integration between data engineering, analytics, and AI workflows. This unified approach eliminates data silos and ensures consistency across the entire data lifecycle.

The project uses the **Diabetic Patient Readmission Dataset**, which contains historical patient encounter records along with demographic, clinical, and hospital utilization information. Diabetes is a chronic condition that often requires repeated hospital visits, making it a suitable domain for studying readmission patterns. Predicting readmission risk for diabetic patients can help healthcare providers proactively identify high-risk patients and take preventive actions such as follow-up care, medication adjustments, or patient education programs.

This capstone project is developed as part of the **Code basics × Databricks 14-Day Data & AI Challenge**, with a strong emphasis on architectural design, data quality, explainability, and real-world applicability. Instead of focusing only on model accuracy, the project highlights the importance of building a **robust, scalable, and governed data pipeline** that supports analytics and AI use cases.

2.Problem Definition & AI Framing

Hospital readmissions pose a significant challenge to healthcare systems, both from a clinical and an operational perspective. A readmission occurs when a patient is admitted to the hospital again within a short period after being discharged. In many cases, these readmissions are unplanned and indicate potential gaps in treatment effectiveness, discharge planning, follow-up care, or patient adherence to

medication and lifestyle recommendations. Reducing avoidable readmissions is therefore a key priority for healthcare providers, as it directly impacts patient outcomes, hospital capacity, and overall healthcare costs.

The primary objective of this project is to **predict the likelihood of patient readmission** using historical healthcare data. Specifically, the project focuses on diabetic patients, a group that often requires repeated hospital visits due to the chronic nature of the disease and the presence of associated complications. By analyzing past patient encounters, demographic details, clinical indicators, and hospital utilization patterns, the goal is to identify patients who are at a higher risk of being readmitted after discharge.

Traditional rule-based approaches, such as fixed thresholds on age, diagnosis count, or length of hospital stay, are often insufficient for this problem. Patient readmission is influenced by a combination of factors that interact in complex and non-linear ways. These relationships are difficult to capture using simple rules or static business logic. As a result, an **AI-driven approach** is more suitable, as machine learning models can learn patterns from historical data and generalize those patterns to unseen cases.

In this project, the problem is framed as a **binary classification task**, where the model predicts whether a patient will be readmitted or not. The **input features** include patient demographics, admission details, hospital utilization metrics, and clinical indicators derived from the dataset. The **output** of the model is a prediction label along with a probability score that represents the patient's risk of readmission.

The success of this project is not measured solely by model accuracy. Instead, success is defined by the ability to build an **end-to-end, scalable, and explainable solution** that integrates data engineering, analytics, and machine learning. A successful outcome includes a well-structured data pipeline using the Lakehouse architecture, meaningful feature preparation, appropriate model selection, and actionable outputs that can support healthcare decision-making.

By framing the problem in this manner, the project aligns with real-world healthcare use cases, where predictive insights are used to assist clinicians and administrators rather than replace them. The predicted readmission risk can help hospitals prioritize follow-up care, allocate resources effectively, and design targeted intervention strategies for high-risk patients.

3. Business Context & Healthcare Relevance

Hospital readmissions are a critical concern for healthcare organizations due to their direct impact on patient care quality, operational efficiency, and financial performance. In many healthcare systems, frequent readmissions are considered an indicator of inadequate treatment outcomes or insufficient post-discharge support. As a result, hospitals are increasingly evaluated not only on the number of patients they treat, but also on how effectively they prevent avoidable readmissions.

From a business perspective, unplanned readmissions significantly increase healthcare costs. Each additional hospital stay consumes valuable resources such as hospital beds, medical staff time, diagnostic equipment, and medications. In value-based care models, healthcare providers may also face financial penalties for high readmission rates. Therefore, reducing readmissions is both a clinical necessity and a strategic business objective.

Diabetic patients represent a particularly important group in this context. Diabetes is a chronic condition that often leads to complications such as cardiovascular disease, kidney failure, and infections, which increase the likelihood of repeated hospital visits. Managing diabetic patient readmissions effectively can help hospitals improve long-term patient outcomes while optimizing operational resources. Early identification of high-risk patients allows healthcare providers to design targeted interventions, such as follow-up appointments, medication reviews, lifestyle counselling, and remote monitoring.

The insights generated from a patient readmission prediction system can support multiple stakeholders within a healthcare organization. Clinicians can use risk scores to prioritize patient care and post-discharge planning. Hospital administrators can leverage predictive analytics to improve capacity planning and resource allocation. Care management teams can focus their efforts on patients who are most likely to benefit from additional support.

By implementing this solution on the **Databricks Lakehouse Architecture**, the project demonstrates how modern data platforms can bridge the gap between raw healthcare data and actionable business insights. The unified architecture enables seamless collaboration between data engineers, analysts, and data scientists, ensuring that predictive models are built on reliable, well-governed data. This approach supports scalability, auditability, and compliance, which are essential requirements in the healthcare domain.

Overall, this project highlights the practical value of data-driven decision-making in healthcare. Predicting patient readmission risk is not just a technical exercise, but a business-enabling capability that can lead to improved patient satisfaction, reduced operational costs, and better utilization of healthcare resources.

4: Dataset Overview & Data Understanding

This project uses the **Diabetic Patient Readmission Dataset**, a widely referenced healthcare dataset that contains historical records of patient hospital encounters. The dataset captures demographic information, admission details, hospital utilization metrics, and clinical indicators related to diabetic patients. Due to the chronic nature of diabetes and its associated complications, this dataset is well suited for analyzing hospital readmission patterns.

Each record in the dataset represents a **single patient encounter**, rather than a unique patient. This means that a patient may appear multiple times in the dataset if they were admitted to the hospital more than once. This structure closely reflects real-world hospital operations, where readmissions are tracked as separate encounters over time.

The dataset includes a combination of **categorical and numerical features**, providing a comprehensive view of patient characteristics and hospital interactions. Demographic attributes such as age and gender describe the patient profile, while admission-related features capture how and why patients were hospitalized. Clinical indicators, including the number of diagnoses, laboratory procedures, and medications, offer insight into the complexity of the patient's medical condition.

From a data quality perspective, the dataset contains missing values and encoded categorical fields, which are common challenges in real-world healthcare data. These issues highlight the importance of systematic data cleaning and transformation before performing analytics or machine learning.

Addressing these challenges is a key part of this project and is handled through the structured bronze, silver, and gold layers of the Lakehouse architecture.

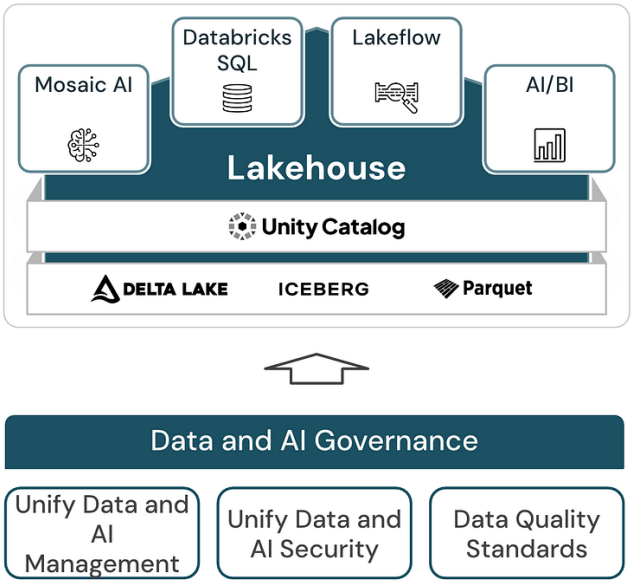
The target variable for this project is **patient readmission status**, which indicates whether a patient was readmitted after discharge. For modelling purposes, this variable is transformed into a binary label representing readmission versus non-readmission. This transformation allows the problem to be framed as a binary classification task suitable for machine learning.

Understanding the dataset at this level is essential before applying any analytical or predictive techniques. A clear understanding of the data ensures that features are selected appropriately, transformations are meaningful, and the resulting models are interpretable and relevant to real-world healthcare scenarios.

5: Databricks Lakehouse Architecture Overview

Modern data-driven applications require a flexible architecture that can support large-scale data ingestion, transformation, analytics, and machine learning within a single unified system. Traditional architectures often separate data lakes and data warehouses, leading to data duplication, governance challenges, and increased operational complexity. To address these limitations, this project adopts the **Databricks Lakehouse Architecture**.

The Lakehouse architecture combines the scalability and low-cost storage of a data lake with the reliability, performance, and governance features of a data warehouse. Databricks provides a unified platform where data engineering, analytics, and machine learning workflows can operate on the same underlying data without unnecessary movement or duplication. This unified approach is particularly beneficial in healthcare scenarios, where data consistency, auditability, and compliance are critical.



The diagram illustrates the Databricks Lakehouse as a unified architecture that supports analytics, machine learning, and AI workloads on a single data foundation. Unity Catalog provides centralized

governance, security, and data quality controls across all layers. In this project, the Lakehouse enables seamless integration of data ingestion, transformation, analytics, and patient readmission prediction without data duplication.

In this project, the Lakehouse architecture is implemented using the **Medallion pattern**, which organizes data into three logical layers: **Bronze**, **Silver**, and **Gold**. Each layer serves a specific purpose and progressively improves data quality and usability. This layered approach ensures that raw data is preserved while also enabling clean, analytics-ready, and machine-learning-ready datasets.

The **Bronze layer** is responsible for ingesting raw data from the source with minimal transformation. It acts as a historical record of incoming data and supports traceability. The **Silver layer** focuses on data cleaning, standardization, and validation, addressing common data quality issues such as missing values and inconsistent formats. The **Gold layer** contains curated, business-ready datasets optimized for reporting, analytics, and machine learning use cases.

By implementing this architecture on Databricks, the project enables seamless integration between structured data processing, analytical querying, and machine learning experimentation. Delta Lake provides ACID transactions and schema enforcement, Unity Catalog supports centralized governance, and MLflow enables experiment tracking and reproducibility. Together, these components form a robust and scalable foundation for building end-to-end AI-driven healthcare solutions.

6. Bronze Layer – Raw Data Ingestion

The Bronze layer represents the first stage of data ingestion in the Databricks Lakehouse Architecture. Its primary purpose is to capture and store raw data exactly as it is received from the source, with minimal or no transformation. This approach ensures that the original data is preserved for traceability, auditing, and future reprocessing if required.

In this project, the Bronze layer is used to ingest the Diabetic Patient Readmission Dataset in its raw form. The dataset is loaded into Databricks using Apache Spark and stored in Delta Lake format. The data is written to a dedicated Bronze storage path using Spark's Delta write operations and validated by reading the stored Delta data back into a DataFrame. This provides reliable storage along with features such as schema enforcement and ACID transactions. Storing the raw data in Delta format allows the system to handle large-scale data efficiently while maintaining consistency.

No major data cleaning, filtering, or business logic is applied at this stage. Fields such as demographic details, admission information, clinical indicators, and readmission status are ingested as-is. This design choice follows best practices in modern data engineering, where raw data is always retained before any transformation logic is introduced.

The Bronze layer acts as a single source of truth for downstream processing. If data quality issues or transformation errors are identified later in the pipeline, the raw data stored in the Bronze layer can be reprocessed without needing to reload data from the original source. This is especially important in healthcare use cases, where data accuracy and auditability are critical.

By clearly separating raw data ingestion from data cleaning and enrichment, the Bronze layer provides a strong foundation for building reliable analytics and machine learning workflows. All subsequent transformations in the Silver and Gold layers depend on the integrity of the data ingested at this stage.

Patient Readmission Prediction using Databricks Lakehouse Architecture

Real-World Scenario:

A hospital receives daily patient encounter data from its electronic health record system. Instead of modifying this data immediately, the hospital stores it in its original form. This ensures that the complete history of patient records is preserved and can be reviewed or reprocessed whenever needed.

7. Silver Layer – Data Cleaning & Transformations

The Silver layer focuses on **improving data quality and consistency** by applying structured cleaning and transformation logic to the raw data ingested in the Bronze layer. In real-world healthcare systems, raw patient data often contains missing values, inconsistent formats, and encoded fields that are not immediately suitable for analytics or machine learning. The Silver layer addresses these challenges by standardizing and validating the data.

In this project, data from the Bronze layer is transformed using **PySpark** to handle common quality issues such as missing or invalid entries. For example, certain clinical fields may contain placeholder values indicating unavailable information, which need to be handled carefully to avoid misleading analysis. Similarly, categorical columns such as admission type or discharge disposition are standardized to ensure consistent interpretation across records. This mirrors real hospital data pipelines, where incoming data from multiple departments must be normalized before use.

Another important transformation performed in the Silver layer is **data type correction and feature standardization**. Numerical fields related to hospital utilization, such as the number of procedures or medications, are cast to appropriate data types to enable accurate aggregations and statistical analysis. At the same time, unnecessary or irrelevant columns that do not contribute to analytical or predictive objectives are filtered out. This step reflects real operational scenarios where healthcare analysts focus only on data elements that provide meaningful insights.

The Silver layer also prepares the dataset for downstream analytics by ensuring that each record represents a clean and reliable patient encounter. By applying consistent transformation logic, the data becomes suitable for exploratory analysis, reporting, and feature engineering. This stage acts as a bridge between raw data storage and business-ready datasets, ensuring that errors or inconsistencies do not propagate further into the pipeline.

Overall, the Silver layer plays a critical role in transforming raw healthcare data into a **trustworthy and analytics-ready form**. Without this layer, insights derived from dashboards or machine learning models could be unreliable or misleading, which is particularly risky in healthcare decision-making contexts.

8. Gold Layer – Business & ML-Ready Data

The Gold layer represents the **final and most refined stage** of the Databricks Lakehouse Architecture. In this layer, data is transformed into **business-ready and machine-learning-ready datasets** that can be directly consumed by analytics dashboards, reporting tools, and predictive models. Unlike the Silver layer, which focuses on data correctness and consistency, the Gold layer focuses on **usability and decision support**.

In this project, the Gold layer is designed to support two primary use cases: **analytical insights** and **patient readmission prediction**. Data from the Silver layer is further refined by applying business logic, selecting relevant attributes, and structuring the data in a way that aligns with real-world healthcare questions. For example, instead of exposing all raw clinical columns, the Gold tables include curated fields such as patient demographics, hospital utilization metrics, and clinical indicators that are meaningful for analysis and modelling.

From an analytics perspective, the Gold layer enables stakeholders to answer practical questions such as identifying patient groups with higher readmission risk, understanding patterns across age groups or admission types, and analysing hospital utilization trends. These curated datasets are optimized for querying using Databricks SQL, allowing dashboards and reports to be built efficiently without requiring additional transformations.

From a machine learning perspective, the Gold layer serves as the **single source of truth for model training and inference**. Features required for prediction are consolidated into a structured format, ensuring that both training and prediction workflows operate on consistent data. This mirrors real-world healthcare analytics systems, where models must be trained on standardized and governed datasets to ensure reliability and reproducibility.

By separating business-ready data into the Gold layer, the project ensures a clear distinction between data preparation and data consumption. This approach simplifies maintenance, improves performance, and allows both technical and non-technical users to interact with the data confidently. In a healthcare setting, this separation is critical, as analytical insights and predictive outputs must be based on trusted and well-defined datasets.

Overall, the Gold layer bridges the gap between raw data processing and actionable intelligence. It enables hospitals and healthcare teams to move from data collection to insight generation and predictive decision-making in a structured and scalable manner.

Additionally, Delta Lake optimizations such as OPTIMIZE are applied at the Gold layer to improve query performance and ensure efficient access for dashboards and machine learning workflows.

9. Feature Engineering Strategy

Feature engineering is a critical step that connects curated data from the Gold layer to the machine learning models used for prediction. In this project, the objective of feature engineering is not to create overly complex features, but to **select and prepare meaningful attributes** that influence patient readmission while maintaining interpretability and reliability.

The features used in this project are derived from the Gold layer, which already contains cleaned and business-ready data. Patient demographic information such as age and gender is included to capture population-level patterns related to readmission risk. Hospital utilization metrics, including the number of inpatient visits, emergency visits, and outpatient visits, are incorporated to reflect the patient's interaction history with the healthcare system. Clinical indicators such as the number of diagnoses, laboratory procedures, and medications provide insight into the complexity and severity of a patient's condition.

Categorical features are prepared in a way that allows machine learning algorithms to interpret them effectively. Rather than relying on raw encoded values, these features are transformed into a standardized format suitable for modeling. Numerical features are validated to ensure consistent scaling and correct data types, enabling accurate comparisons across patient records. This approach reflects real-world healthcare analytics systems, where feature preparation must balance predictive power with explainability.

The target variable for the model represents **patient readmission status**, which is converted into a binary label indicating whether a patient was readmitted after discharge. Framing the problem in this manner allows the use of supervised machine learning techniques and supports straightforward interpretation of model outputs.

An important design consideration in this project is the alignment between training and inference. The same feature preparation logic is applied consistently during both model training and prediction to avoid data leakage or inconsistencies. This ensures that model performance metrics are reliable and that predictions generated in a production-like setting are meaningful.

Overall, the feature engineering strategy focuses on **simplicity, relevance, and consistency**. By selecting features grounded in healthcare context and preparing them systematically, the project ensures that machine learning models operate on high-quality inputs and produce actionable insights for patient readmission management.

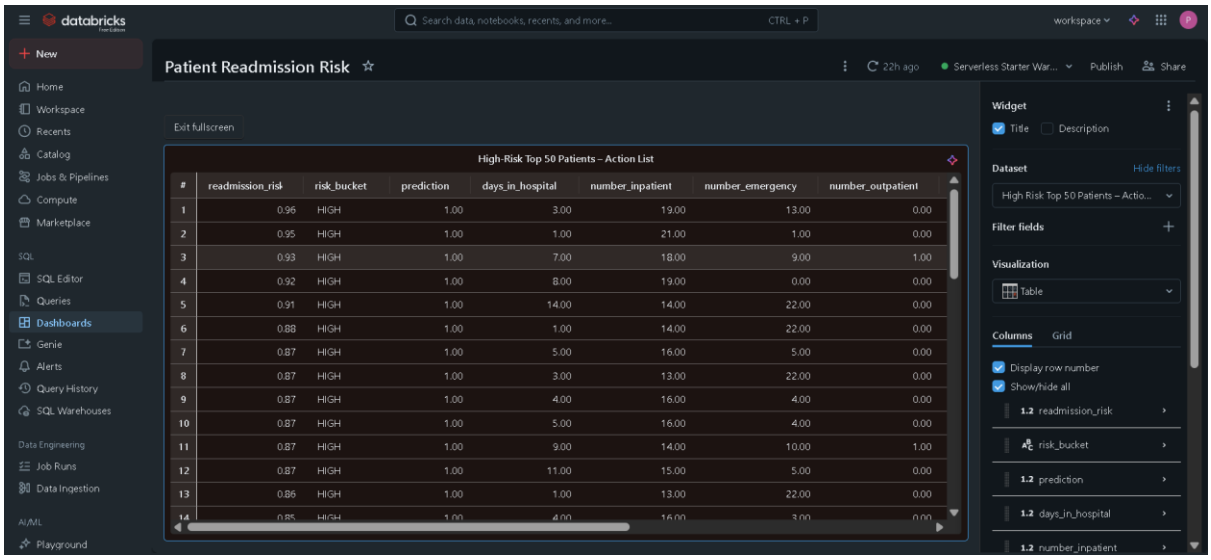
10. Analytics & Dashboard Insights

Analytics plays a crucial role in transforming curated data into **actionable insights** that support decision-making. Before applying machine learning, it is important to understand historical patterns and trends related to patient readmissions. In this project, analytics is performed on the Gold layer datasets using **Databricks SQL**, enabling fast and interactive exploration of patient readmission behavior.

The analytics layer focuses on answering practical healthcare questions that hospital administrators and care teams commonly face. By analyzing readmission rates across different age groups, admission types, and hospital utilization patterns, stakeholders can identify patient segments that are more vulnerable to repeated hospital visits. For example, patients with a higher number of prior inpatient or emergency visits often exhibit a greater likelihood of readmission, indicating the need for closer follow-up care.

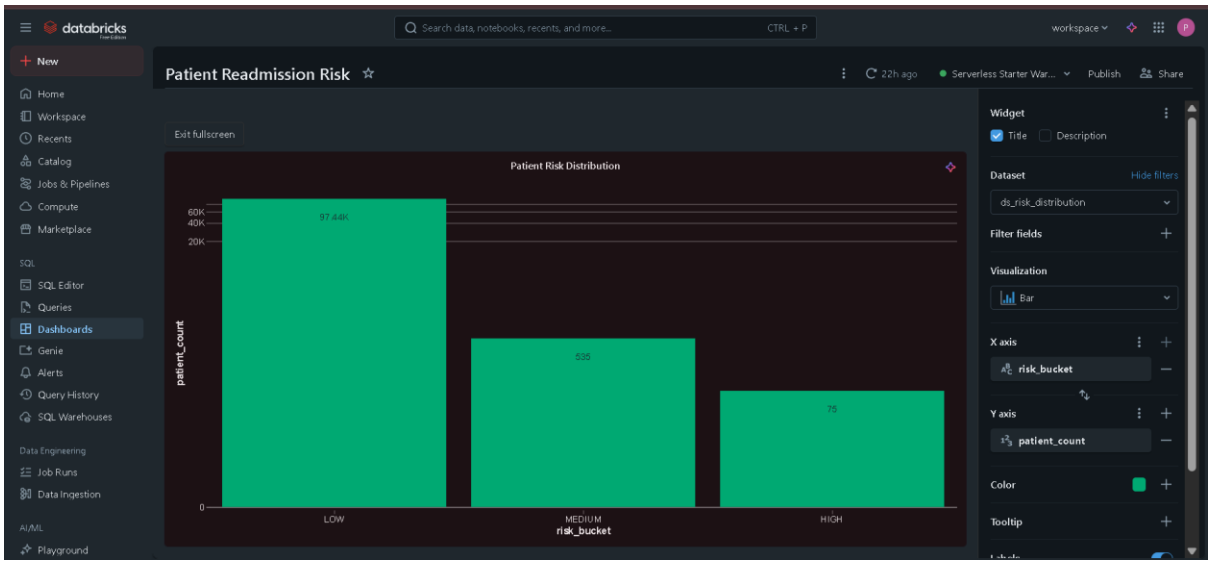
Dashboards built using Databricks SQL provide a consolidated view of these insights in an easily interpretable format. Visualizations such as bar charts and summary tables help compare readmission trends across demographic groups, while aggregated metrics highlight overall hospital performance. These dashboards enable non-technical users, such as healthcare managers and clinicians, to quickly assess risk patterns without requiring knowledge of underlying data processing or machine learning techniques.

Patient Readmission Prediction using Databricks Lakehouse Architecture



Top high-risk patients identified for intervention (partial view)

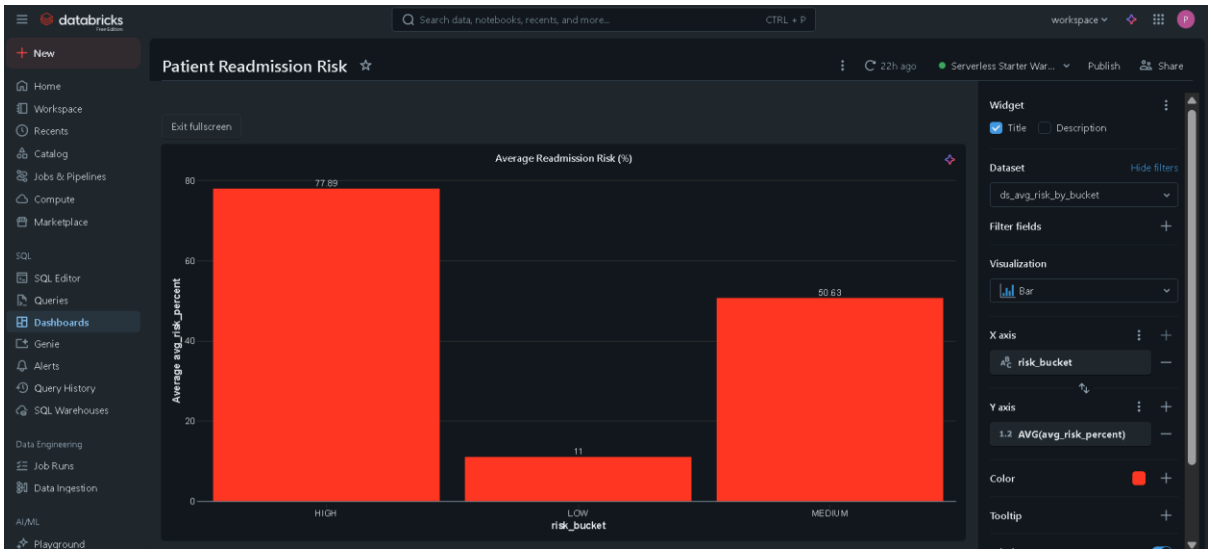
This table provides an actionable view of patients with the highest predicted readmission risk. By including hospitalization duration and visit frequency, care teams can proactively plan follow-ups, medication adjustments, and discharge support to reduce avoidable readmissions. Due to space constraints, only the top records are displayed, while the complete list is available within the Databricks dashboard for operational use.



Distribution of patients across readmission risk categories

The majority of patients fall into the low-risk category, while a smaller proportion are classified as medium and high risk. This imbalance highlights the importance of targeted intervention strategies rather than uniform care plans.

Patient Readmission Prediction using Databricks Lakehouse Architecture



Average readmission risk percentage by risk category

Patients in the high-risk category exhibit significantly higher average readmission probability compared to medium and low-risk groups. This insight enables clinicians to prioritize monitoring and post-discharge care for high-risk patients.

From a real-world perspective, these analytical insights support proactive decision-making. If analytics reveal that certain admission types or patient profiles consistently show higher readmission rates, hospitals can prioritize targeted interventions for those groups. This may include enhanced discharge planning, patient education programs, or follow-up monitoring strategies. In this way, analytics serves as a bridge between raw data and operational actions.

The analytics layer also complements the machine learning component of the project. Insights derived from dashboards help validate feature selection and ensure that predictive models are aligned with observed data trends. By grounding machine learning efforts in descriptive analytics, the project ensures that predictions are interpretable and relevant to real healthcare scenarios.

Overall, the analytics and dashboard layer demonstrates how structured data can be transformed into meaningful insights that support both strategic planning and day-to-day healthcare operations.

11. Machine Learning Approach

11.1 Problem Formulation

In this project, patient readmission prediction is framed as a **binary classification problem**. The objective is to determine whether a discharged patient is likely to be readmitted within 30 days.

Label = 1 → Patient readmitted within 30 days

Label = 0 → Patient not readmitted

From a healthcare perspective, this problem is **risk-sensitive**. Missing a high-risk patient can lead to poor health outcomes and increased hospital costs. Therefore, the modeling approach prioritizes

identifying patients with a higher likelihood of readmission rather than simply maximizing overall accuracy.

11.2 Choice of Input Data (Gold Layer)

All machine learning models are trained using data from the **Gold layer**, which contains cleaned, standardized, and ML-ready patient records.

The Gold dataset includes:

- **Demographic features**: race, gender, age
- **Admission details**: admission type, days in hospital
- **Utilization history**: inpatient, outpatient, and emergency visit counts
- **Clinical indicators**: number of diagnoses, lab procedures, medications

Using Gold-layer data ensures that the models operate on **consistent and validated inputs**, avoiding raw data noise and transformation inconsistencies.

11.3 Feature Representation Strategy

Since Spark ML algorithms require numerical inputs, a structured feature preparation strategy is applied:

Categorical variables (race, gender, age, admission type) are encoded using StringIndexer

Numerical variables are explicitly cast to numeric types

All features are combined into a single vector using VectorAssembler

This transformation logic is implemented using a Spark ML Pipeline, ensuring that:

The same preprocessing steps are applied during training and inference

Feature preparation is reproducible and scalable

This pipeline-based approach reflects real-world ML systems where preprocessing and modeling are tightly coupled.

11.4 Model Selection Strategy

Two classification algorithms were selected for comparison:

Logistic Regression

- Provides strong interpretability
- Commonly used in healthcare risk prediction
- Produces probability scores that support threshold tuning

Random Forest Classifier

- Captures non-linear relationships

- Handles feature interactions automatically
- Serves as a benchmark against a more complex model

The purpose of using multiple models is **not complexity**, but to evaluate which approach better supports the business objective of identifying high-risk patients.

11.5 Evaluation Focus: Recall-Oriented Strategy

Rather than relying solely on accuracy, this project emphasizes **recall for the positive class (readmitted patients)**.

Why recall matters in healthcare:

- A false negative means a high-risk patient is missed
- Early intervention opportunities are lost
- Readmission costs and patient risk increase

To support this goal, **probability threshold tuning** is applied instead of using the default 0.5 cutoff. This allows the model to flag more high-risk patients while maintaining reasonable precision.

11.6 Training–Inference Consistency

A key design principle in this project is maintaining **consistency between training and prediction**:

- The same feature pipeline is used for both phases
- No manual feature manipulation outside the pipeline
- Predictions generated later follow the same transformations

This ensures that model performance observed during evaluation is reliable and transferable to real-world usage.

12. Model Training, Evaluation & Metrics

12.1 Training Setup and Pipeline Design

Model training was performed using the curated Gold dataset and implemented using a **Spark ML Pipeline** to ensure repeatability and consistent preprocessing across training and inference. The dataset was split into training and test subsets, and the same feature preparation steps were applied to both models.

The pipeline consists of three main stages:

1. **Categorical Encoding**: Categorical columns were transformed using StringIndexer with handleInvalid="keep" to safely manage unseen or missing categories during inference.
2. **Feature Assembly**: Indexed categorical features and numeric features were combined using VectorAssembler into a single feature vector column named **features**.

3. **Model Estimation**: The final pipeline stage was the classifier (Logistic Regression or Random Forest).

This design ensures that preprocessing logic is not repeated manually and reduces the risk of mismatched transformations between train and test datasets.

12.2 Logistic Regression Training

A **Logistic Regression** model was trained using the pipeline output column features and the target column label. Logistic Regression was selected as a baseline model because it is fast, scalable in Spark, and produces probability scores that are useful for risk-based decision-making.

After training, the model generated predictions on the test dataset and produced probability values for the positive class (readmission).

12.3 Threshold Tuning for Recall-Oriented Prediction

Because the project goal is to **identify as many high-risk readmission patients as possible**, evaluation prioritized **positive-class recall** (label = 1). Instead of relying on Spark's default decision threshold, a custom probability cutoff was applied:

- **Threshold used:** 0.2

A new prediction field (pred_thr) was created using the positive-class probability:

- If P(readmission) ≥ 0.2 → predict 1
- Else → predict 0

This approach increases sensitivity to high-risk patients and reflects real-world healthcare decision support, where missing a high-risk patient can be more costly than raising additional alerts.

12.4 Metrics Computation (Test Data)

Model performance was evaluated on test data using:

- **AUC (Area Under ROC Curve)**: Measures the model's ability to separate classes.
- **Accuracy**: Overall correctness of predictions.
- **Positive Recall**: Percentage of actual readmissions correctly identified.
- **Positive Precision**: Percentage of predicted readmissions that are correct.

For recall and precision, confusion-matrix components were computed directly from the thresholded predictions:

- True Positives (TP), False Negatives (FN), False Positives (FP)
- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)

This provides explicit control over how the model is evaluated and supports threshold-based healthcare risk screening.

12.5 Logistic Regression Results

The Logistic Regression model produced the following results on test data:

- **AUC:** 0.6341
- **Accuracy:** 0.8872
- **Positive Recall:** 0.1128

Interpretation:

- The model shows moderate class separation ability (AUC ~0.63).
- Accuracy is high largely due to class imbalance (most patients are not readmitted).
- Recall is improved using threshold tuning, but the model still identifies only a portion of readmitted patients, indicating the challenge of predicting readmission using limited structured features.

12.6 Random Forest Training and Results

A **Random Forest Classifier** was trained using the same preprocessing stages (same indexers + assembler) to ensure a fair comparison. The configuration used:

- numTrees = 30
- maxDepth = 6
- seed = 42

Random Forest achieved:

- **AUC:** 0.6353
- **Accuracy:** 0.8866
- **Positive Recall:** 0.0008

Interpretation:

- AUC and accuracy are similar to Logistic Regression.
- However, positive recall is extremely low, meaning the model predicted almost no readmission cases as positive under the chosen threshold.
- This indicates that, with the current setup and threshold strategy, Random Forest does not effectively identify high-risk readmission patients in this dataset.

12.7 Model Comparison and Selection

Both models achieved similar overall accuracy and AUC, but the key business objective was **positive recall** (identifying readmission patients). Based on test results:

- Logistic Regression recall (0.1128) is significantly higher than Random Forest recall (0.0009).

Therefore, **Logistic Regression was selected as the preferred model** for this use case because it better supports recall-oriented identification of high-risk patients.

12.8 Notes on Class Imbalance and Practical Evaluation

Patient readmission datasets are typically imbalanced (far more non-readmitted than readmitted cases). In such scenarios:

- Accuracy can appear high even when the model is weak at detecting the minority (readmitted) class.
- Recall-focused evaluation and threshold tuning provide a more realistic assessment for healthcare risk prediction tasks.

This validates the decision to track recall and precision explicitly and use threshold-based predictions for operational decision support.

13. MLflow Experiment Tracking

13.1 Purpose of MLflow in this Project

MLflow was used to track and manage the machine learning experiments conducted for patient readmission prediction. Since multiple models were trained and evaluated (Logistic Regression and Random Forest), MLflow provided a structured way to log model configurations, evaluation metrics, and trained model artifacts. This ensured reproducibility and made it easy to compare runs in a consistent manner.

13.2 Experiment Logging Strategy

For each model run, MLflow was used to log:

- **Model parameters** (e.g., model type, threshold, hyperparameters)
- **Evaluation metrics** (AUC, accuracy, positive recall, positive precision)
- **Model artifact** (the trained Spark ML pipeline/model)

To improve traceability and enable reproducible deployment, an input_example was also provided while logging the model. This helps MLflow infer the model signature and improves the usability of the stored model artifact.

13.3 Logistic Regression Run Tracking

The Logistic Regression model was logged as a dedicated MLflow run with the run name **"LogisticRegression"**. The following information was tracked:

- **Parameters logged:** model type, regularization settings, elastic net configuration, and probability threshold (0.2)
- **Metrics logged:** AUC, accuracy, positive recall, and positive precision
- **Artifact logged:** complete Spark ML pipeline (indexers + assembler + classifier)

This ensures that the exact preprocessing + model logic can be reproduced for future training, evaluation, or inference.

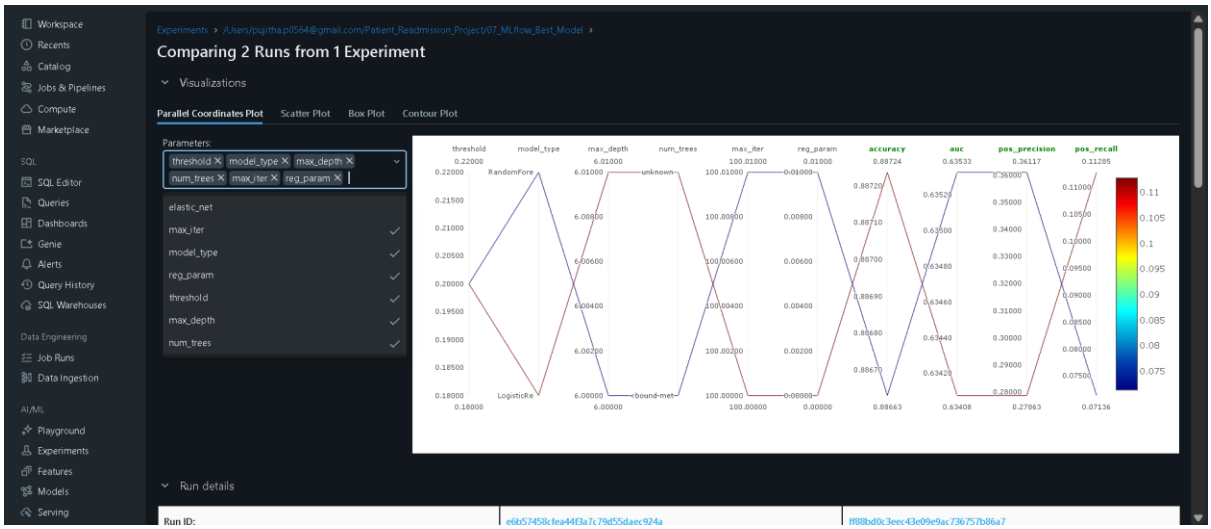
13.4 Random Forest Run Tracking

Similarly, the Random Forest model was tracked under a separate MLflow run named **"RandomForest"**. The following were logged:

- **Parameters logged:** model type, number of trees, maximum depth, and threshold (0.2)
- **Metrics logged:** AUC, accuracy, positive recall, and positive precision
- **Artifact logged:** the trained Random Forest pipeline/model

This enabled fair comparison between the two model runs under the same evaluation strategy.

13.5 Model Comparison and Best Model Selection



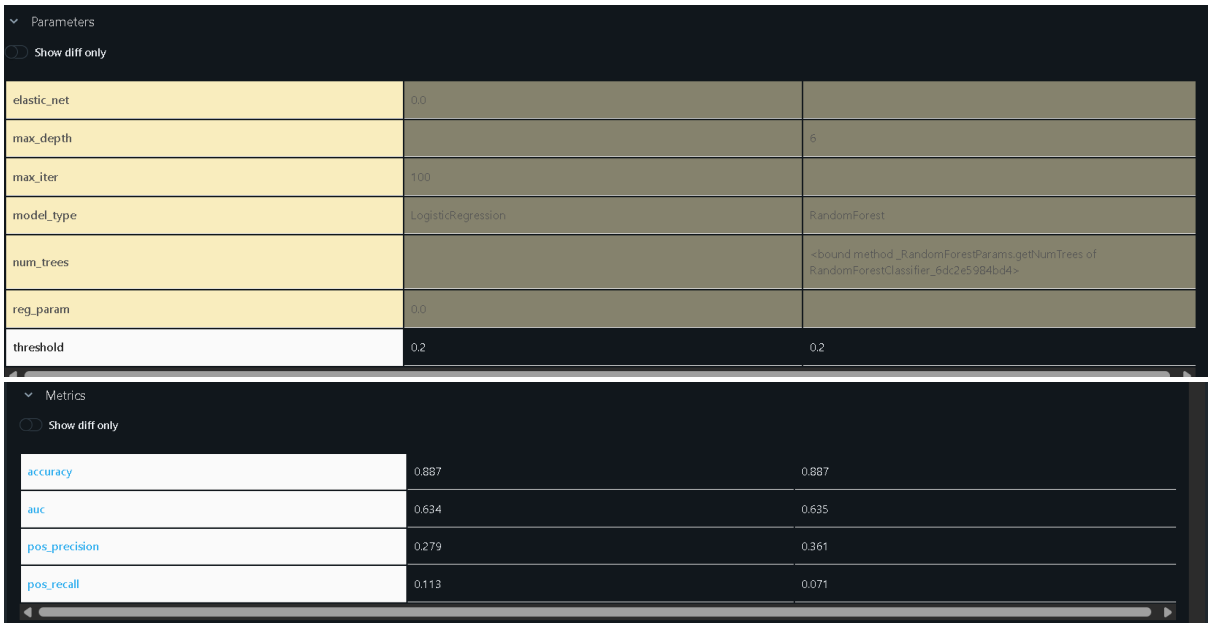
MLflow Model Comparison – Logistic Regression vs Random Forest

This MLflow comparison view shows the performance of Logistic Regression and Random Forest models evaluated on the same test dataset and decision threshold.

The comparison includes key metrics such as accuracy, AUC, positive-class precision, and positive-class recall. While both models achieved similar accuracy and AUC values, Logistic Regression clearly demonstrates higher positive-class recall compared to Random Forest.

Since recall is the primary business metric in this healthcare use case, Logistic Regression was preferred for identifying patients at high risk of readmission.

Patient Readmission Prediction using Databricks Lakehouse Architecture



Tracked Parameters and Evaluation Metrics in MLflow

This table summarizes the parameters and evaluation metrics logged for both models using MLflow. Model-specific parameters such as maximum iterations, regularization strength, tree depth, and number of trees were captured to ensure experiment reproducibility.

Evaluation metrics include accuracy, AUC, positive precision, and positive recall. Although accuracy and AUC are comparable across models, Logistic Regression achieves a higher positive-class recall, making it more suitable for the healthcare objective of minimizing missed readmission cases.

13.6 Notes on MLflow Warnings (Non-blocking)

During model logging, MLflow displayed warnings related to environment dependency inference (e.g., Spark Connect / PySpark version labeling and missing model signature). These warnings do not affect model training or evaluation outcomes. To improve model packaging, pip requirements and an input example were provided while logging the model, ensuring the stored model artifact is usable and reproducible.

14. End-to-End Database ↔ AI Workflow

The project demonstrates a complete end-to-end workflow that connects raw healthcare data ingestion to machine learning–driven insights using the Databricks Lakehouse platform. The workflow begins with raw data ingestion into the Bronze layer, where patient readmission data is stored in Delta Lake format as a reliable and auditable source of truth.

The data then flows through the Silver layer, where structured cleaning, validation, and standardization are applied using PySpark. This ensures that inconsistent values, missing fields, and irrelevant attributes do not propagate further into analytics or machine learning stages. The cleaned and standardized data is promoted to the Gold layer, which contains business- and ML-ready datasets optimized for analytics and prediction.

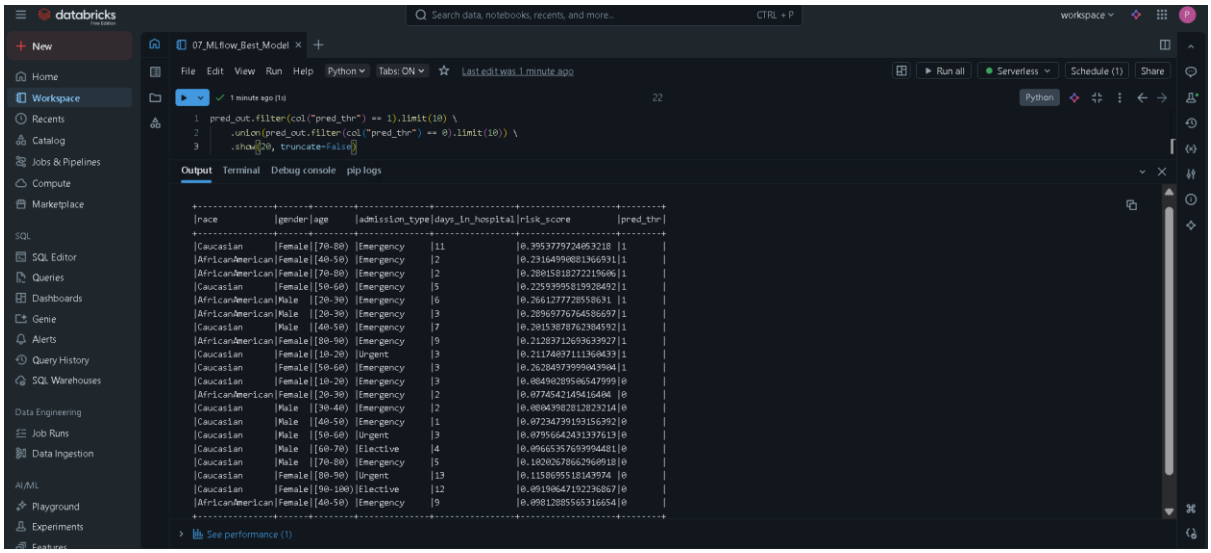
Patient Readmission Prediction using Databricks Lakehouse Architecture

From the Gold layer, two parallel paths are enabled. First, Databricks SQL is used to perform analytics and generate dashboards that reveal historical readmission patterns, risk trends, and operational insights for healthcare stakeholders. Second, the same Gold data is used as input for machine learning pipelines, ensuring consistency between analytics and predictive modeling.

Machine learning models are trained and evaluated using PySpark ML pipelines, with experiments tracked in MLflow. Model parameters, metrics, and artifacts are logged to ensure reproducibility and transparency. Based on business-driven evaluation criteria, the best-performing model is selected and applied to generate patient-level readmission risk scores.

The final predictions are written back to Delta tables, enabling seamless integration between database storage, analytics dashboards, and AI-driven decision support. This closed-loop workflow reflects a production-style architecture where data engineering, analytics, and machine learning operate on a unified platform without data duplication or silos.

Overall, this end-to-end pipeline demonstrates how Databricks Lakehouse enables scalable, governed, and explainable AI workflows that can directly support real-world healthcare decision-making.



This snapshot shows patient-level readmission risk predictions generated by the selected Logistic Regression model. The risk_score represents the probability of readmission, while pred_thr indicates the final risk classification based on a tuned threshold. These predictions are written back to Delta storage, enabling analytics dashboards and operational decision-support workflows.

15.Governance, Security & Data Management

Data governance is a critical requirement in healthcare environments where sensitive patient data must be handled securely and responsibly. In this project, governance is supported through Databricks' centralized data management capabilities.

Data is organized into clearly defined layers (Bronze, Silver, and Gold), ensuring transparency and traceability across the pipeline. Access controls and permissions can be managed using Databricks Unity Catalog, enabling role-based access to datasets and preventing unauthorized data usage.

Patient Readmission Prediction using Databricks Lakehouse Architecture

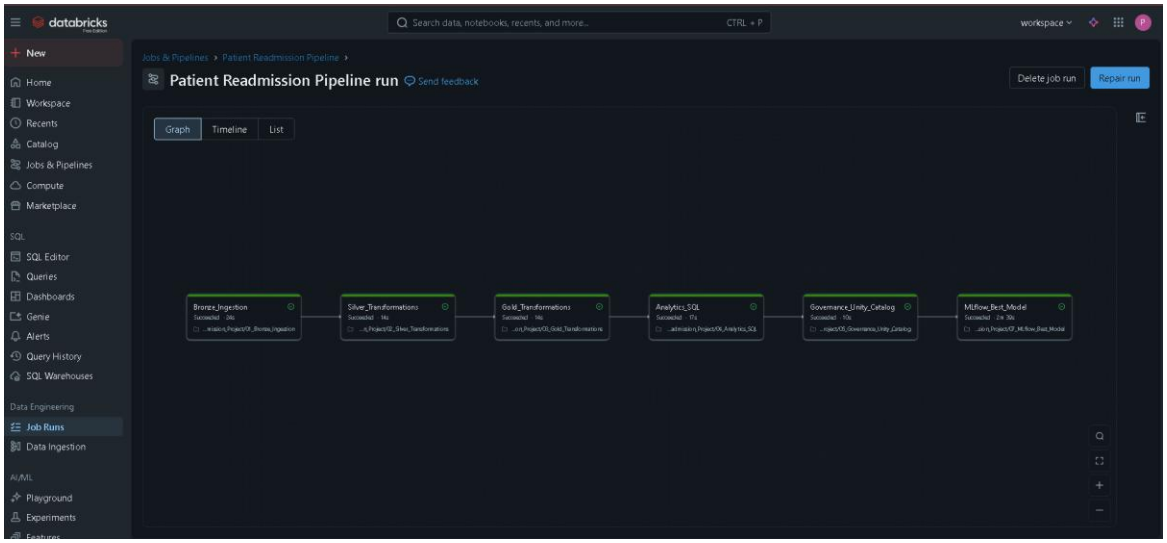
Storing data in Delta Lake further supports governance by providing schema enforcement, versioning, and audit history. These features ensure that data changes are traceable and that analytical and machine learning outputs are built on trusted and well-governed datasets.

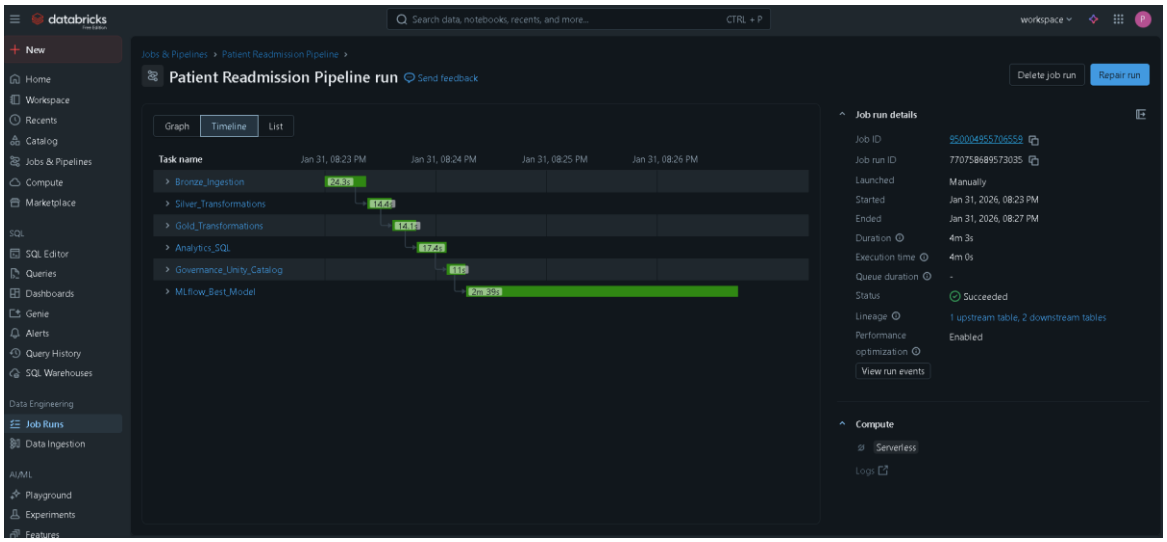
16.Orchestration & Workflow Automation

Orchestration ensures that the data and machine learning workflows run in a controlled and repeatable manner. In this project, the overall pipeline is designed to follow a logical execution order, starting from data ingestion in the Bronze layer, followed by Silver and Gold transformations, analytics, and machine learning workflows.

Databricks Jobs can be used to schedule and automate these notebooks, enabling end-to-end execution without manual intervention. This approach supports production-like execution where data pipelines and model runs can be triggered on a scheduled basis or on demand.

By structuring the project into modular notebooks aligned with the Medallion Architecture, the solution supports scalability, easier maintenance, and reliable execution in real-world data platforms.





Patient Readmission Prediction using Databricks Lakehouse Architecture

17. Business Impact & Practical Use

17.1 Business Impact

Patient readmissions are a major challenge in healthcare, leading to increased operational costs, resource strain, and potential deterioration in patient outcomes. By identifying patients who are at high risk of readmission, hospitals can proactively intervene and reduce avoidable repeat visits.

In this project, the machine learning model generates **patient-level readmission risk scores** rather than simple binary outcomes. This enables healthcare providers to move from reactive care to **proactive risk management**. Patients identified as high risk can receive additional attention at the time of discharge, such as follow-up appointments, medication reconciliation, and patient education.

From a cost perspective, even a small reduction in readmissions can result in significant savings for hospitals. Early identification of high-risk patients allows hospitals to allocate limited resources more effectively, focusing care on patients who need it most rather than applying uniform interventions to all patients.

17.2 Practical Use Cases in Healthcare Operations

The outputs of this project can be integrated into real-world healthcare workflows in several ways:

- **Discharge Planning:** High-risk patients can be flagged for enhanced discharge planning and follow-up care.
- **Care Coordination:** Case managers can prioritize outreach for patients with higher predicted readmission risk.
- **Clinical Decision Support:** Risk scores can support clinicians in identifying patients who may benefit from additional monitoring or post-discharge support.
- **Hospital Performance Monitoring:** Aggregated risk trends can help administrators evaluate the effectiveness of care programs aimed at reducing readmissions.

Because predictions are stored back into Delta tables, these insights can be directly consumed by analytics dashboards, enabling non-technical stakeholders to access AI-driven insights without interacting with machine learning code.

17.3 Alignment with Real-World Data Platforms

This project demonstrates how a unified Lakehouse architecture can support both analytics and AI use cases on the same data foundation. By combining Delta Lake storage, SQL analytics, and machine learning workflows on Databricks, the solution avoids data duplication and ensures consistency across reporting and prediction layers.

Such an architecture is well-suited for healthcare environments where data governance, traceability, and explainability are critical. The approach shown in this project can be extended to other clinical risk prediction scenarios, such as length-of-stay forecasting or chronic disease management.

Patient Readmission Prediction using Databricks Lakehouse Architecture

18. Challenges Faced & Key Learnings

18.1 Key Challenges Faced

1. Handling Real-World Healthcare Data Complexity

The dataset contained numerous categorical variables, encoded values, and placeholder entries that required careful interpretation. Understanding which fields were clinically meaningful and which could introduce noise was a key challenge. Unlike synthetic datasets, healthcare data requires domain-aware decisions to avoid misleading analysis or model behavior.

2. Severe Class Imbalance in Readmission Prediction

Patient readmission is a relatively rare event compared to non-readmission. This imbalance caused traditional metrics such as accuracy to appear high even when the model failed to identify readmitted patients. Recognizing this limitation and redesigning the evaluation strategy was a critical challenge.

3. Aligning Model Evaluation with Business Objectives

Initial model results using default thresholds produced near-zero recall for high-risk patients. This highlighted the gap between technical model performance and real-world healthcare objectives. Adjusting the evaluation focus from accuracy to **positive class recall** required rethinking how model success should be defined.

4. Ensuring End-to-End Consistency Across the Pipeline

Maintaining consistent feature preparation across Silver, Gold, analytics, and machine learning stages was non-trivial. Any mismatch could lead to unreliable predictions. This required careful design of Spark ML Pipelines to ensure that the same transformations were applied during training and inference.

5. Managing Delta Lake Schema Evolution

While persisting predictions back into Delta Lake, schema conflicts occurred due to differences in inferred data types across pipeline stages. Resolving these issues reinforced the importance of explicit schema handling, controlled overwrites, and versioned storage in production-grade data platforms.

6. Balancing Model Complexity and Explainability

Although more complex models like Random Forest were explored, they did not always align with the project's recall-focused objective. Choosing a simpler but more interpretable model required balancing predictive performance with explainability—an essential consideration in healthcare systems.

18.2 Key Learnings

1. Metric Selection Drives Model Value

One of the most important learnings was that **choosing the right evaluation metric is more impactful than choosing the most complex model**. In healthcare, optimizing for recall enables safer and more actionable predictions compared to optimizing for accuracy alone.

2. Threshold Tuning Is Essential for Risk-Based Decisions

Probability thresholds significantly influence model behavior. By tuning the threshold instead of relying on defaults, the model became more sensitive to high-risk patients, making predictions more aligned with clinical decision-support use cases.

3. Unified Lakehouse Architecture Simplifies AI Pipelines

Using Databricks Lakehouse architecture allowed data engineering, analytics, and machine learning workflows to operate on a shared, governed data foundation. This eliminated data silos and ensured consistency across reporting and prediction layers.

4. ML Pipelines Improve Reliability and Reproducibility

Implementing feature transformations and modeling within Spark ML Pipelines ensured reproducible results and prevented data leakage. This approach mirrors real-world ML systems where preprocessing and modeling must be tightly coupled.

5. Experiment Tracking Enhances Decision Confidence

MLflow experiment tracking provided transparency into model behavior and performance trade-offs. Comparing models side-by-side made it easier to justify model selection decisions based on business-aligned metrics rather than intuition.

6. End-to-End Thinking Matters More Than Isolated Models

The project reinforced that machine learning does not exist in isolation. The true value of AI emerges only when predictions are integrated back into data platforms, analytics dashboards, and operational workflows.

19. Conclusion & Future Enhancements

19.1 Conclusion

This capstone project successfully delivered an end-to-end patient readmission prediction solution using the Databricks Lakehouse platform. The workflow covered the complete lifecycle—from raw data ingestion in the Bronze layer, structured cleaning and transformations in the Silver layer, and curated business/ML-ready datasets in the Gold layer.

Using Databricks SQL, the project produced meaningful analytical insights and dashboards that help understand readmission trends across patient groups and hospital usage patterns. Building on these insights, machine learning pipelines were implemented using Spark ML to predict readmission risk. Model evaluation was aligned with real-world healthcare priorities by focusing on positive-class recall and applying threshold tuning to better identify high-risk patients. MLflow experiment tracking ensured reproducibility and enabled transparent comparison of model performance, leading to a justified selection of Logistic Regression as the preferred model for this use case.

Overall, the project demonstrates how data engineering, analytics, governance, orchestration, and machine learning can be combined into a unified, production-style workflow that supports proactive healthcare decision-making.

19.2 Future Enhancements

While the current solution meets the capstone requirements and demonstrates a complete data-to-AI pipeline, the following enhancements can further improve real-world impact:

1.Advanced Class Imbalance Handling

Apply techniques such as class weighting, stratified sampling, or SMOTE-style approaches (where appropriate) to improve recall without increasing false positives significantly.

2.Richer Feature Engineering

Introduce additional clinically meaningful features such as diagnosis grouping, medication categories, comorbidity indicators, and derived utilization trends (e.g., visit frequency over time).

3.Hyperparameter Tuning & Model Expansion

Use systematic hyperparameter tuning and evaluate additional models (e.g., Gradient-Boosted Trees) while maintaining interpretability requirements.

4.Explainability & Model Interpretability

Add feature importance analysis and explainability tools to help clinicians understand why a patient is flagged as high risk, improving trust and adoption.

5.Operationalization & Monitoring

Deploy the best model as a batch scoring job or endpoint and implement monitoring for prediction drift, data quality issues, and model performance over time.

6.Risk Stratification & Intervention Workflows

Convert risk scores into clear risk bands (Low/Medium/High) and connect them to recommended intervention actions such as follow-up scheduling, medication review, and discharge planning support.

Thank you