BIG DATA ANALYTICS

Assignment: Case Studies

Team Members:
Muneezé Malik
Madiha Shahab

Instructor:
Sir Shayan Shah

Submission Date:
21st November 2025

Case Study: Patient Experience & Outcomes

(Using MIMIC-IV Big Data)

Title: Enhancing Patient Journeys & Reducing ICU Readmissions via Big Data Analytics

Phase 1: Problem Identification

Selected Use Case:

Patient Experience & Outcomes (Healthcare)

Business Problem:

Healthcare organizations aim to enhance patient care quality and minimize unnecessary hospital visits. One major challenge is unplanned hospital readmissions, which increase costs and negatively affect patient experience.

Project Objective (Scope):

I will build a machine learning model that predicts whether a patient is likely to be readmitted within 30 days after discharge.
This prediction will help hospitals identify high-risk patients early and provide timely follow-up care.

Dataset Selected:

MIMIC-IV (v3.1)—A large, publicly available real-world hospital dataset containing admissions, lab tests, diagnoses, ICU stays, and outcomes.
Source: PhysioNet (https://physionet.org/content/mimiciv/3.1/)

Why This Dataset Fits the Use Case:

- Contains detailed patient journeys across emergency, inpatient, and ICU departments.
- Supports outcome-based analysis, such as 30-day readmission.
- Includes multi-structured data (labs, notes, vitals), matching the case study requirement.
- Real-world big data scale: over **546,000 hospital admissions**.

Boundaries:

- Focus on adult inpatient admissions.
- Use only data available up to the moment of discharge (to avoid data leakage).
- Only unplanned readmissions within 30 days will be considered.

Phase 2: Data Sourcing

Dataset Selected:

MIMIC-IV (Medical Information Mart for Intensive Care), Version 3.1

Source (Credible Public Repository):

PhysioNet — MIT Laboratory for Computational Physiology
Dataset Link: https://physionet.org/content/mimiciv/3.1/
(PhysioNet is a globally recognized academic repository for healthcare datasets.)

Dataset Metadata

Scale of the Dataset:

- **364,627** unique patients
- **546,028** hospital admissions
- **94,458** ICU stays
- Over **50 GB** of structured + semi-structured data

Modules & Structure:

MIMIC-IV is divided into two main modules:

1. hosp Module (Hospital-Wide EHR Data)

Contains:

- Demographics
- Admissions & discharges
- Transfers between departments
- Laboratory results
- Microbiology tests

- Medications & prescriptions
- Diagnoses & procedures
- Provider orders
- Billing & ICD codes

2. icu Module (ICU High-Granularity Data)

Contains:

- Vitals signs
- Inputs/outputs
- Treatments & interventions
- Clinical events
- Procedural events
- Charted nurse observations
- Medications administered in the ICU

Number of Tables:

~40+ tables across both modules, organized to support patient journey tracking and outcome analysis.

Why This Dataset Fits the Selected Use Case (Patient Experience & Outcomes):

- Contains **complete patient journeys** from admission to discharge.
- Supports **30-day readmission prediction**, which aligns with your chosen use case.
- Includes multi-structured data (labs, vitals, notes) needed for **realistic big data analytics**.
- Provides a **360° view of each patient**, matching the problem requirement.
- Large scale and complexity meet the criteria for a **Big Data study**.

File Formats:

- Comma-separated values (CSV)
- Deidentified, HIPAA-compliant
- Organized for analysis with SQL/Python

Phase 3: Pipeline Design

To process **multimillion-row clinical datasets** like MIMIC-IV, a **distributed Data Lakehouse architecture** is used. The pipeline is designed for scalable ETL, patient journey graph analysis, feature engineering, and visualization for predictive modeling.

A. Data Ingestion & Storage

- **Ingestion:**
    - Batch processing via **Airflow / PySpark** to move CSVs from PhysioNet into the data lake.
    - Optional future real-time streaming using **Kafka** for near real-time hospital events.
- **Storage Layers:**
    - **Raw zone:** immutable CSVs (gzipped) stored in **S3 / HDFS** for auditability.
    - **Bronze:** cleaned Parquet tables with standardized types, null handling, and ingestion metadata.
    - **Silver:** joined and normalized tables (patients, admissions, lab events, transfers) with derived fields like **length of stay**, **comorbidity scores**, and summary vitals.
    - **Gold / Features:** curated feature tables ready for ML models, including graph-derived features from patient journeys.
- **File Format & Partitioning:** Parquet for efficiency, partitioned by anchor_year_group and hashed subject_id to optimize Spark processing.
- **Catalog & Metadata:** AWS Glue / Databricks Unity Catalog to store schemas, track lineage, and support governance.

B. Processing & Patient Journey Graph Analysis

- **Processing Engine: Apache Spark (PySpark)** to handle large tables like chartevents (>300 million rows) and efficiently process joins and aggregations.
- **Patient Journey Graph Modeling: GraphFrames / GraphX** is used to model transfers between care units.
    - **Nodes:** Hospital units (ER, MICU, CCU, Ward, OR).
    - **Edges:** Patient transfers with attributes in-time, out-time, and duration.

- **Graph Analysis Goals:**
  - **PageRank / Centrality:** Identify units with high congestion and bottlenecks.
  - **Path Analysis:** Detect long transfer paths associated with delays and readmissions.
  - **Dwell Time Analysis:** Compute maximum and average unit stay to identify inefficiencies.
  - **Community Detection:** Cluster similar patient flows to discover patterns affecting outcomes.
- **ETL & Feature Extraction:** Aggregate graph metrics (total transfers, max dwell, path length) per patient stay to create predictive features.

C. Feature Engineering

- Demographics: Age, gender, anchor_age_group.
- Clinical History: Charlson / Elixhauser comorbidity scores, prior admissions.
- Admission Snapshot: Length of stay, admission type, discharge disposition.
- Labs & Vitals: Mean, min, max, trend, and last observed values per stay.
- Journey Features: Number of transfers, longest dwell time, path cluster ID, unit PageRank.
- Optional NLP: Embeddings from clinical notes (e.g., ClinicalBERT) for text-based features.

D. Visualization & Reporting

- **Tableau / Superset:**
  - Patient journey dashboards, Sankey maps of transfers.
  - Length-of-stay and readmission risk heatmaps.
- **Python (Matplotlib / Seaborn):**
  - Model performance plots: ROC curves, PR curves, and calibration plots.
  - SHAP summaries for feature importance and interpretability.
- **Graph Visualizations:** Interactive Sankey or network diagrams to explore patient paths.

E. Scalability & Big Data Considerations

- Spark enables distributed computation on large datasets.
- Delta Lake ensures ACID compliance and time-travel for iterative ML workflows.
- Partitioning by anchor_year_group and hashed subject_id avoids skew and ensures even workload distribution.
- Modular design allows future integration of streaming data, NLP, and real-time risk scoring.

Phase 4: Machine Learning Methodology

This phase defines the approach for predicting patient outcomes (e.g., 30-day readmissions, prolonged stay) using MIMIC-IV. It covers preprocessing, feature engineering, scaling, algorithm selection, and dataset characteristics.

A. Pre-Processing Strategy

1. **Handling Missing Values**
    - **Numerical features:**
        - Use **median imputation** for vitals and lab results to reduce skew impact.
        - Optional advanced: **KNN imputation** for critical lab values where patterns exist across similar patients.
    - **Categorical features:**
        - Missing categories are encoded as Unknown to preserve data integrity.
    - **Rationale:** Missing data is common in clinical datasets due to skipped tests or partial documentation. Median/KNN ensures minimal bias while retaining most records.
2. **Handling Categorical Variables**
    - **One-Hot Encoding:** For nominal categories like admission type, unit, or discharge disposition.
    - **Label Encoding:** For ordinal features like severity scores or age groups.
    - **Rationale:** Preserves meaningful order for ordinal data, while one-hot prevents incorrect ordinal assumptions for nominal features.
3. **Outlier Handling & Validation**
    - Clip physiological measurements (heart rate, blood pressure, lab values) to clinically plausible ranges.

- Remove impossible timestamps or duplicate events.

B. Feature Engineering

- **Demographic features:** Age, gender, anchor_age_group.
- **Clinical history:** Charlson/Elixhauser comorbidity scores, prior admissions count.
- **Admission features:**
  - Length of stay (LOS)
  - Admission type (elective, emergency)
  - Discharge disposition
- **Laboratory & vitals aggregates:** Mean, min, max, last value, and trend (slope) for each lab or vital over the admission period.
- **Graph-derived journey features:**
  - Total number of transfers
  - Max/average dwell time per unit
  - Path length in patient journey
  - Unit centrality scores (PageRank)
- **Optional text-based features:** Embeddings from clinical notes (ClinicalBERT) to capture physician observations.

C. Scaling/Normalization

- **Numerical features:**
  - **Standardization (Z-score):** $z = \frac{x - \mu}{\sigma}$
    $z=(x-\mu)/\sigma$ for algorithms sensitive to feature scale (logistic regression, XGBoost, neural networks).
  - **Normalization (Min-Max 0–1):** optional for tree-based models if embedding features are added.
- **Rationale:** Standardization ensures faster convergence and consistent coefficient interpretation for linear models; tree-based models (XGBoost, Random Forest) are scale-insensitive.

D. Algorithm Recommendation

Algorithm	Reason for Selection
XGBoost / LightGBM	Handles high-dimensional, structured data; robust to missing values; supports feature importance; high accuracy. Ideal for predicting readmission (binary classification).
Logistic Regression	Baseline for interpretability; allows clinical insights into feature contributions.
LSTM / GRU	Optional for sequential vitals/labs time-series modeling per admission. Captures temporal patterns for more precise prediction.

Justification:

- The dataset contains **structured numerical, categorical, and temporal data**, making XGBoost ideal.
- Graph-derived features enrich structured representation.
- Trade-offs: Logistic Regression is interpretable but less accurate; XGBoost balances accuracy and some interpretability (via SHAP). LSTM improves temporal modeling at the cost of complexity.

E. Dataset Analysis

- **Size & Scale:**
  - 364,627 unique patients, 546,028 hospital admissions, 94,458 ICU stays.
  - Multiple tables with millions of rows (chartevents, labevents, transfers).
- **Type of Data:**
  - Structured: demographics, labs, vitals, ICD codes, transfers.
  - Multi-structured: optional text (physician notes) for embeddings.
- **Balance:**
  - Readmission or adverse outcome labels are **imbalanced** (fewer positive cases).
  - Can apply **class weighting**, SMOTE, or balanced sampling.
- **High-dimensionality:**
  - Hundreds of lab/vital features + graph-derived metrics.
  - Feature selection/regularization recommended.

- **Time-series component:**
    - Vitals and labs recorded over admission/ICU stay; useful for temporal models or trend-based features.

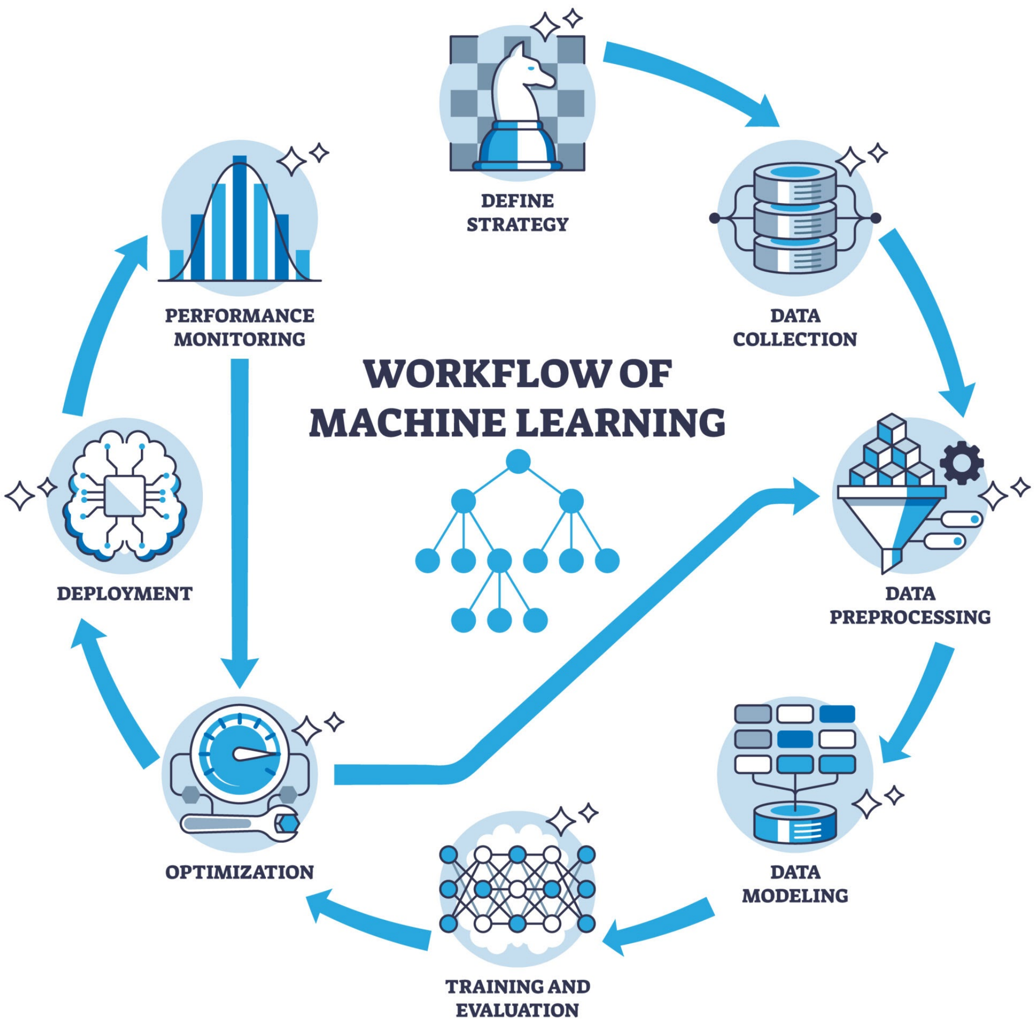
Phase 5: Implementation Plan

This phase outlines a **Python-based strategy** to implement the predictive modeling for patient outcomes using MIMIC-IV. It includes libraries, high-level pseudo-code, and evaluation metrics.

A. Library Selection

Purpose	Python Library / Tool
Data processing & ETL	`pandas`, `numpy`, `PySpark` (for large tables)
Feature engineering & graph analysis	`networkx`, `graphframes`, `scikit-learn` (for aggregation)
Machine learning	`XGBoost`, `LightGBM`, `scikit-learn` (Logistic Regression, metrics)
Time-series / sequential modeling	`tensorflow.keras`, `pytorch` (optional LSTM/GRU)
Visualization	`matplotlib`, `seaborn`, `plotly`, `tableau` (dashboards)
Data validation & quality	`great_expectations`
Explainable AI	`shap`

B. Pseudo-Code / High-Level Logic



Code:

```
# 1. Load raw data
patients = spark.read.parquet("silver/patients")
admissions = spark.read.parquet("silver/admissions")
transfers = spark.read.parquet("silver/transfers")
labevents = spark.read.parquet("silver/labevents")
chartevents = spark.read.parquet("silver/chartevents")

# 2. Merge tables & create labels
data = merge_tables(patients, admissions, labevents, chartevents, transfers)
data["readmit_30"] = compute_readmit_label(data)

# 3. Preprocessing
data = handle_missing_values(data, strategy="median/KNN")
data = encode_categorical(data, strategy="one-hot/label")
data = scale_features(data, method="standardization")

# 4. Feature Engineering
data = create_demographics_features(data)
data = create_lab_vitals_aggregates(data)
data = create_graph_features(transfers)
data = optional_nlp_features(data)

# 5. Train/Test Split
train_data, test_data = split_by_subject_id(data, test_size=0.2)

# 6. Initialize and Train Model
model = XGBoostClassifier(params)
model.fit(train_data.features, train_data.labels)

# 7. Evaluate Model
preds = model.predict(test_data.features)
roc_auc = compute_roc_auc(test_data.labels, preds)
recall = compute_recall(test_data.labels, preds)

# 8. Hyperparameter Tuning (optional)
best_model = hyperparameter_tuning(model, train_data)

# 9. Save Model & Features
save_model(best_model, "gold/models/readmit_xgb.pkl")
save_feature_table(data.features, "gold/features/")
```

C. Evaluation Metrics

Metric	Purpose / Justification
Recall (Sensitivity)	Prioritize identifying patients at risk of readmission; missing a true positive is costly.
ROC-AUC	Measures overall discriminatory power of model across thresholds.
Precision	Optional: balance false positives for operational resource allocation.
F1-Score	Provides harmonic mean of Precision and Recall for imbalanced data.
Calibration plots	Ensure predicted probabilities align with true risk.

Business Justification:

- High **Recall** ensures that at-risk patients are flagged for intervention.
- ROC-AUC ensures model can distinguish high-risk vs low-risk patients across thresholds.
- Precision helps reduce unnecessary interventions but is secondary to Recall in healthcare outcomes.