Patient Risk Simulation Pipeline

Chris Adan

June 2025

Agenda

- Business Problem & Objective
- Data Landscape & Technical Approach
- Architecture & Flowchart
- Business Impact & Sample Analytics
- Extension Opportunities

Enhancing Clinical Decision-Making with Simulated FHIR-Based Risk Models

Task

- Hospitals use predictive models to flag patients at risk of clinical deterioration based on EHR data (vitals, labs, encounters)
- These models depend on structured, reliable data yet EHR systems expose data through complex FHIR resources
- This project explores building a transparent, modular risk scoring pipeline from scratch using open FHIR data

Project Outline

- Developed an end-to-end pipeline to ingest raw FHIR data, perform transformations, and generate clean analytics-ready datasets
- Implemented medallion architecture in Snowflake with Bronze → Silver → Gold layers to structure RAW → STAGE → ANALYTICS flow
- Addressed FHIR's nested, hierarchical structure via flexible schema design, modular dbt models, and Python-based ETL scripts
- Established testing, documentation, and metadata practices to ensure data quality, traceability, and extensibility

Design

Medallion architecture: Promotes clean separation of concerns and modular data processing across Bronze (raw), Silver (cleaned), and Gold (analytics) layers

Python: Enabled flexible FHIR data extraction via APIs and transformation using powerful built-in libraries (e.g., requests, pandas, json)

dbt: Provided orchestrated, version-controlled SQL transformations with built-in testing, documentation, and reusability through macros and packages

End-to-End FHIR Data Pipeline: Landscape & Approach

Data Extraction

Queried open FHIR simulation data (10,000+ Patient profiles) from the public HAPI FHIR API using Python scripts

Raw Ingestion

Parsed raw nested JSON into Snowflake using Python ETL scripts and staged it into the **RAW** layer

Staging & Normalization

Applied initial transformations and vocabulary mappings (e.g., LOINC codes) to standardize lab and observation data in the **STAGE** layer

Analytics Layer

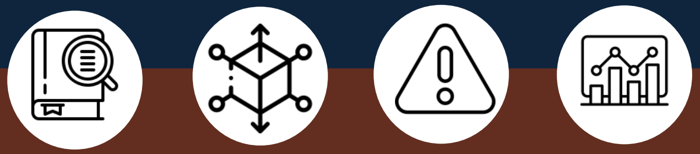
Created clean, analysis-ready tables in the ANALYTICS layer using modular, testable dbt models

Feature Engineering

Generated derived fields and ML-ready features for downstream modeling and KPI reporting

Resource Type	Example Fields
Patient	Name, ID, Age
Encounter	Start/End, Class
Observation	Consumer Name, Result Value
Condition	SNOMED code

Extension Opportunities



- **Support Additional Vocabularies:** Integrate RxNorm, SNOMED, and ICD-10 for enhanced clinical context
- **Ingest New FHIR Resources:** Expand pipeline to handle new resources like Procedure, MedicationRequest Immunization
- **Enable Risk Scoring Experiments:** Engineer new ML features and simulate early warning models
- **Add Visualization Layer:** Connect to Tableau or Looker to explore trends and monitor pipeline outputs

Modular & Scalable Architecture Design

Python ETL -> Snowflake
Extracted and parsed FHIR JSON using Python, loading into structured Snowflake tables with metadata and type validation

1

dbt Transformations
Built modular dbt models per resource layer (e.g., observation, encounter) using DRY principles with macros and packages

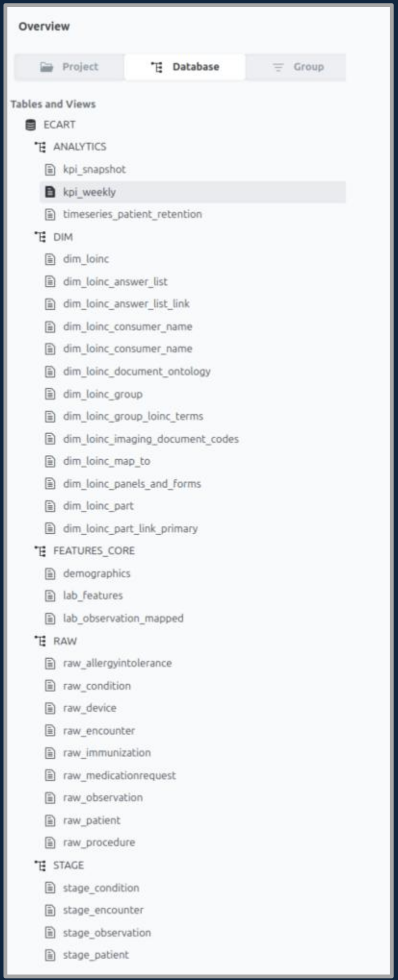
2

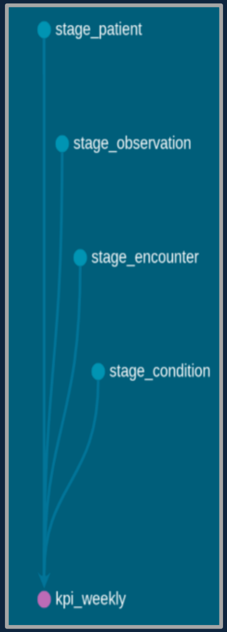
Built-in Testing & Documentation
Implemented column-level tests and descriptions for maintainability and data quality assurance

3

Plan for CI/CD
Explore GitHub Actions for scheduling and plan future integration with incremental loads, visualization, and model pipelines

4





Schema	Purpose
RAW	FHIR payloads
STAGE	Unpacked JSON into tabular
DIM	LOINC seed mapping for semantic labeling
ANALYTICS	KPIs, daily retention, and time series
FEATURES_CORE	Lab features, demographics for ML

Extension Opportunities

- **Strengthen Governance:** Add tags, contracts, more custom testing, and macros for consistent quality
- **Automate Updates:** Implement incremental models with scheduled runs
- **Streamlit Frontend:** Build an interactive app for visualization and on-demand prediction execution

Translating Raw Records into Clinical Metrics

Daily Retention Tracking
Monitor new vs. returning patients using unique IDs to monitor patient activity and outcomes

KPI Snapshots
Generate daily metrics on patient volume, encounters, conditions, and average lab observations

Weekly Resource Trends
Track week-over-week changes across core FHIR resources for operational and clinical insights

ML Feature Outputs
Produce structured features (e.g., vitals, labs, demographics) ready for downstream modeling or alert systems

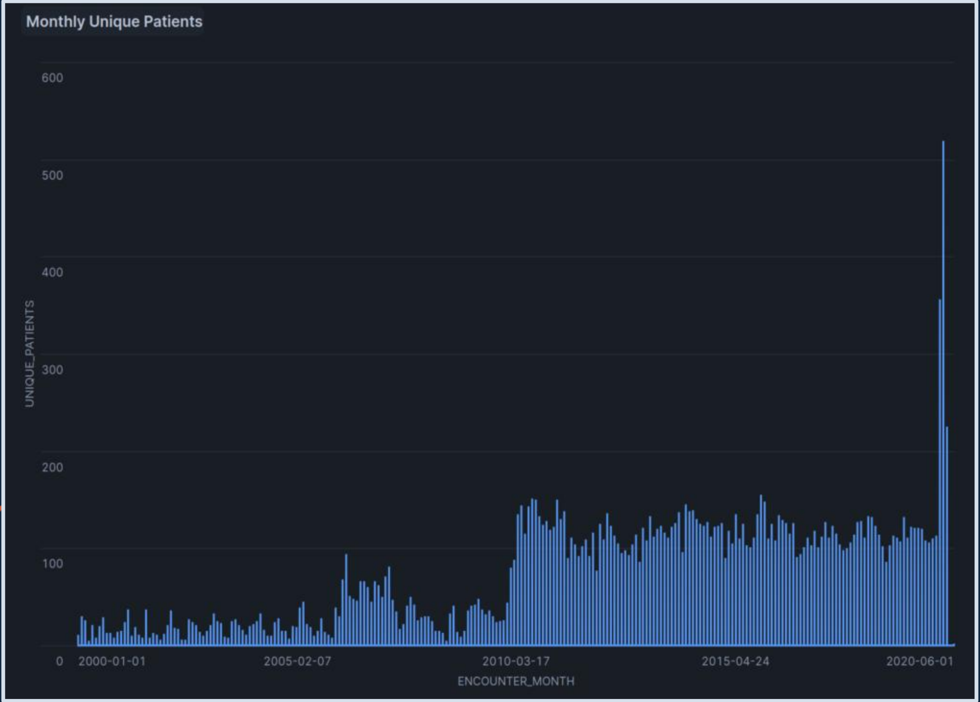
Extension Opportunities

Integrate BI Tools: Connect outputs to platforms like Looker, Tableau, or Metabase for interactive dashboards

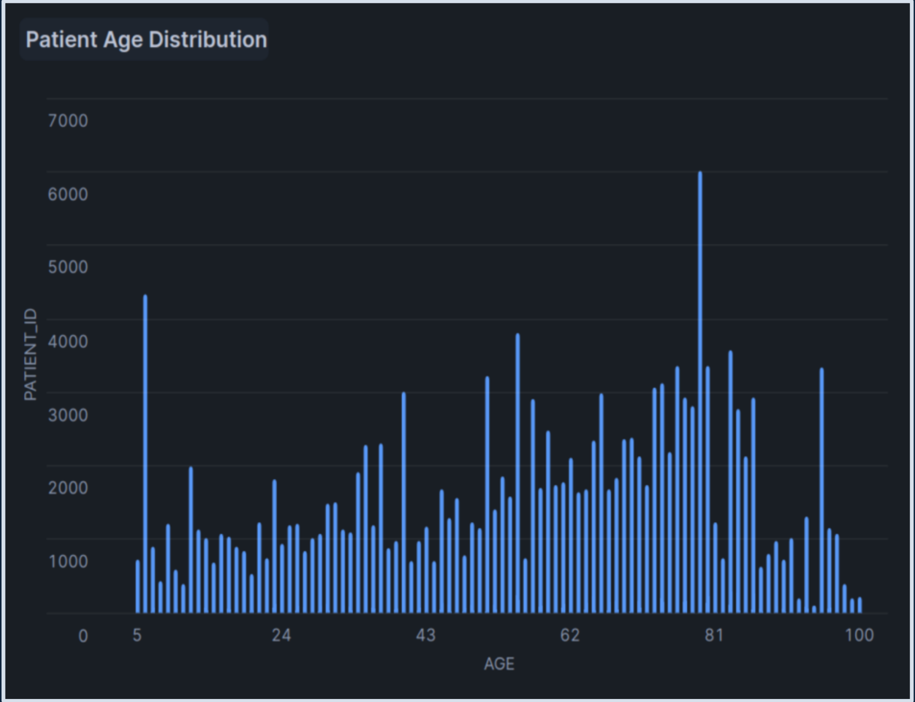
Implement Anomaly Alerts: Set up rule-based or statistical triggers to flag unusual patient trends or data issues

Model Training Integration: Feed engineered features directly into ML pipelines for real-time risk scoring or experimentation

Sample Analytics



Source: fhir.analytics.kpi_weekly



Source: fhir.features_core.demographics

	SNAPSHOT_DATE	NUM_PATIENTS	NUM_ENCOUNTERS	NUM_CONDITIONS	AVG_OBSERVATION_VALUE
1	2025-06-26	9000	21198	1415	69.03973988

Source: fhir.analytics.kpi_snapshot

Takeaways & Ideas for Expansion

Lessons Learned

- Gained hands-on experience with FHIR schema complexity, healthcare vocabularies, and data sparsity challenges
- Reinforced the value of modular, testable pipeline design for maintainability and scalability
- Built foundational knowledge in clinical data modeling and system extensibility using open-source tools and APIs

Opportunities

- Data Sources: Ingest new open health datasets (e.g., CDC APIs, CMS, HealthData.gov) or generate synthetic FHIR-like data on demand (Synthea)
- Pipeline Expansion: Add support for streaming data (Kafka/Kinesis), additional FHIR resources, and new ML features
- Visualization & Scale: Deploy a full-stack app with Streamlit, GitHub Actions, and live dashboards via Tableau Public or Power BI

Thank You!

Materials

[GitHub Repository]

[HAPI – Open FHIR API]

[LOINC Public Dataset]

Socials

[Find me on LinkedIn]

[Check out my GitHub]

[Read on Medium]