Capstone Project:

High-Cost Patient Prediction Using Synthea Data

Part 3: Technical Report and Analysis

Yves Janssens

1. Executive Summary

Healthcare systems are under increasing pressure to control costs, especially as chronic conditions and aging populations drive up utilization. To address this, we built a predictive cost modeling solution that identifies patients likely to incur high future healthcare expenses, enabling early, targeted interventions.

Project Overview:

We developed a patient-level machine learning pipeline using synthetic EHR data (Synthea) to:

- Predict each patient's total cost using past clinical records.
- Reveal key cost drivers such as chronic diseases and demographics.
- Support proactive care planning through clear dashboards and actionable insights.

The solution integrates:

- A structured dataset with demographics, conditions, medications, and procedures.
- Two predictive models:
  - Linear Regression for interpretability and clinical insights.
  - K-Nearest Neighbors for higher accuracy in predicting individual patient costs.
- Interactive Tableau dashboards that visualize cost patterns, highlight high-risk groups, and compare actual vs. predicted values.

Key Findings:

- Mental health conditions are the leading cost driver — patients with mental illness had 4.7× higher costs, emphasizing the need for integrated behavioral health.
- Age was the most predictive continuous variable — with costs rising dramatically among older adults.
- Chronic conditions such as cancer, cardiovascular disease, hypertension and diabetes were consistently linked to higher predicted costs.
- Gender gaps revealed lower costs for male patients (~50% lower), suggesting potential disparities or underutilization worth further exploration.

- A few conditions (e.g., kidney disease) were associated with unexpectedly low costs, flagging possible data quality or care access issues.

Business Value for Stakeholders:

Our predictive model and dashboards directly support both:

◇ **Health Insurance Population Health Teams:**

- Segment members by predicted financial risk.
- Design targeted outreach and care management programs.
- Plan budgets and adjust risk scores more effectively.

◇ **Hospital Preventive Care & Outreach Departments:**

- Visualize cost trends by demographic or condition.
- Launch campaigns for high-need groups (e.g., mental health, aging adults).
- Prioritize resources for screening, behavioral health, or community programs.

Next Steps & Considerations:

- **Actionable Use:** Teams can estimate patient cost using just a few fields (age, gender, condition flags), and immediately route high-risk individuals into care coordination workflows.
- **Limitations:** The Synthea dataset is synthetic, so real-world deployment would require retraining on actual EHRs and validating assumptions.
- **Future Work:** Expand features (e.g., lifestyle data), use larger samples, and explore explainable AI methods for enhanced model transparency.

Bottom Line:

This solution empowers healthcare teams to act early. By predicting cost risk and uncovering its drivers, insurers and hospitals can target interventions more precisely, improve outcomes, and control costs — all through data-driven population health management.

2. Problem, Goals, and Audience

2.1 Problem Statement

Rising healthcare costs and the burden of chronic diseases necessitate the early identification of high-cost patients.

2.2 Project Relevance and Importance

Healthcare systems face increasing financial strain due to aging populations, rising treatment costs, and the growing burden of chronic diseases like diabetes, heart disease, and cancer.

A small percentage of patients often account for a disproportionate share of total medical costs, making early identification and proactive management of these high-cost individuals a clinical and operational priority.

2.3 Project Goals

- Build a comprehensive patient-level dataset to model and predict total healthcare costs using historical clinical data.
- Identify key clinical and behavioral cost drivers to support early outreach, risk stratification, and care coordination strategies.
- Deliver interactive Tableau dashboards tailored for health insurance population management teams and hospital preventive care programs, providing clear insights to guide targeted interventions, patient engagement campaigns, and strategic resource planning.

2.4 Intended Audience

This project is designed to support two primary stakeholder groups:

1. **Health Insurance Population Health Teams**

   These teams can use the predictive model to:

   - Identify patients who are likely to incur high future costs.
   - Design targeted care programs (e.g., nurse outreach, care coordination).
   - Improve resource planning and risk adjustment strategies.

2. **Hospital Preventive Care and Outreach Departments (e.g., Community Health, Patient Marketing)**

   These departments can:

   - Use visual dashboards to detect cost trends tied to chronic conditions or preventable behaviors.
   - Segment and target patient populations for preventive care education, screening reminders, or lifestyle programs.
   - Support hospital marketing or outreach campaigns by identifying high-need demographics.

2.5 Success Criteria

- A well-structured patient-level dataset integrating multiple clinical domains (conditions, medications, procedures, encounters, etc.).
- Predictive models (Linear Regression and K-Nearest Neighbors Regression) that estimate total patient cost with interpretable output and usable insights for non-technical teams.
- A set of Tableau dashboards that visualize:
  - Cost breakdowns by demographic group and condition category.
  - High-cost patient distribution and care behavior patterns.
  - Relationships between chronic diseases and predicted costs.
  - Actual vs. predicted cost performance to highlight model reliability.
- Generation of targeted, actionable recommendations for insurance risk teams and hospital outreach departments, enabling more informed planning and earlier interventions for high-cost patients.

3. Data Sources and Data Dictionary

3.1 Data Sources

The data for this project was sourced from the publicly available Synthea™ synthetic patient generator, which provides realistic but fictional healthcare data for research and educational purposes. The dataset was downloaded from the official Synthea website: https://syntheticahealth.github.io/synthea

The data comprises 10 structured CSV files, listed in the table below, covering various aspects of patient care such as demographics, encounters, conditions, procedures, medications, and more. Individual data dictionaries for each file are provided in **Section 8. Appendix A. Data Dictionaries**.

File Names	Description
patients.csv	Patient demographic data.
encounters.csv	Patient encounter data.
conditions.csv	Patient conditions or diagnoses.
procedures.csv	Patient procedure data including surgeries.
medications.csv	Patient medication data.
observations.csv	Patient observations including vital signs and lab reports.
allergies.csv	Patient allergy data.
careplans.csv	Patient care plan data, including goals.
providers.csv	Clinicians that provide patient care.
organizations.csv	Provider organizations including hospitals.

To enable efficient querying and relational analysis, a PostgreSQL database was created using PgAdmin 4. The database was created using SQL scripts, a sample of which is included in **Section 8. Appendix B. SQL Code Snippets**. The CSV files were then imported into the database. The entire SQL code for this part can be found in the attached **healthcarecost SQL Text file**.

Relationships among the tables were established based on shared keys (e.g., Patient, Encounter, Organization), and an Entity Relationship Diagram (ERD) was created to document the data model. The ERD was generated using dbdiagram.io - Database Relationship Diagrams Design Tool and is available in **Section 8. Appendix C. ERD Diagram**.

3.2 Data Quality & Cleaning Process

To ensure analytical integrity and model readiness, a comprehensive data quality assessment and cleaning process was conducted across all 10 Synthea data tables. Below is a summary of the key checks and corrections applied:

Null / Missing Values:

Each table was reviewed for missing or null values across all columns. Missing data was recorded for documentation but not immediately dropped or imputed, as downstream handling will occur during Python-based processing of the master dataset.

Key Finding: No major data loss or schema-level issues were detected.

Next Step: Null handling (e.g., imputation or flagging) will be completed during feature engineering in Python.

Type Consistency Checks:

Data types for each column were verified to ensure consistency with expected formats.

Key Finding: The *observations.value* column is of type text, as it contains a mix of numeric values and free-text responses. To address this, an additional column was created to extract and store purely numeric values for modeling and analysis.

Referential Integrity Validation:

Referential integrity was assessed to confirm that all foreign keys correctly link to valid records in their referenced tables.

Key Finding: In observations, 30,363 records have a missing encounter reference. However, all **non-null** encounter IDs in this table are valid and correspond to existing rows in the encounters table.

Action Taken: A cleaned version of the observations table was created, retaining only rows with valid encounter references for our encounter-level analysis.

Referential integrity across all remaining foreign key relationships was confirmed.

Duplicate Handling:

Each table was examined for duplicate rows using a combination of primary keys and full-row comparisons.

Key Finding: 442 duplicate rows were identified in the cleaned observations table.

Action Taken: All identified duplicates were removed from the dataset prior to downstream processing.

Categorical Standardization and Logical Correctness:

A range of logical checks were performed to ensure data validity and categorical alignment.

Temporal Logic: All *stop* dates were checked to ensure they occurred after *start* dates.

Finding: 5 records had invalid ranges; *stop* was adjusted to match *start* for those cases. All *death_date* entries were confirmed to occur after *birth_date*. No future-dated *birth_date* values were found.

Categorical Variables: Key fields such as gender were confirmed to use standardized categories (M, F). No changes were made to names (first, last, maiden_name), as these have no impact on analytical outcomes. Minor corrections were applied to standardize *provider_name* formatting.

Note: All relevant SQL snippets used during this data quality process are included in **Section 8. Appendix B. SQL Code Snippets**. The entire SQL code for this part can be found in the attached **Data Cleaning SQL Text file**.

3.3 Data Dictionary (Patient-Level Master Dataset)

The data dictionary provides a detailed reference of all key variables used in this project. It includes both raw fields extracted directly from the original Synthea tables and derived fields created through data processing and transformation during the feature engineering phase.

The dictionary serves as a single point of reference to understand the structure, semantics, and origin of each variable used in the analysis and modeling pipeline.

Individual table-level data dictionaries are included in **Section 8. Appendix A**.

Column Name	Description	Source Table	Type
patient	Patient unique identifier.	patients	raw
birth_date	Patient's DOB.	patients	raw
gender	Patient's gender.	patients	raw
race	Patient's race.	patients	raw
ethnicity	Patient's ethnicity	patients	raw
marital_status	Patient's marital status.	patients	raw
age	Patient's age at last recorded consultation.	patients	derived
encounter_cost	Patient's total consultation cost.	encounters	derived
procedure_cost	Patient's total procedure cost.	procedures	derived
medication_cost	Patient's total medication cost.	medications	derived
total_cost	Patient's total cost.	multiple (agg.)	derived

Column Name	Description	Source Table	Type
%_encounter_cost	% of total cost associated with consultations.	multiple (calc.)	derived
%_procedure_cost	% of total cost associated with procedures.	multiple (calc.)	derived
%_medication_cost	% of total cost associated with medications.	multiple (calc.)	derived
diabetes_duration_years	Number of years the patient has diabetes.	conditions	derived
diabetes_age_at_first_diagnosis	Age at which the patient was diagnosed with diabetes.	conditions	derived
hypertension_duration_years	Number of years the patient has hypertension.	conditions	derived
hypertension_age_at_first_diagnosis	Age at which the patient was diagnosed with hypertension.	conditions	derived
cardiovascular_duration_years	Number of years the patient has cardiovascular disease.	conditions	derived
cardiovascular_age_at_first_diagnosis	Age at which the patient was diagnosed with cardiovascular disease.	conditions	derived
respiratory_duration_years	Number of years the patient has respiratory disease.	conditions	derived
respiratory_age_at_first_diagnosis	Age at which the patient was diagnosed with respiratory disease.	conditions	derived
cancer_duration_years	Number of years the patient has cancer.	conditions	derived
cancer_age_at_first_diagnosis	Age at which the patient was diagnosed with cancer.	conditions	derived
mental_health_duration_years	Number of years the patient has mental health disease.	conditions	derived
mental_health_age_at_first_diagnosis	Age at which the patient was diagnosed with mental health disease.	conditions	derived
kidney_disease_duration_years	Number of years the patient has kidney disease.	conditions	derived
kidney_disease_age_at_first_diagnosis	Age at which the patient was diagnosed with kidney disease.	conditions	derived
total_encounters	Total number of clinic visits.	encounters	derived
preventive_encounters	Number of encounters classified as preventive care for a patient.	encounters	derived
preventive_ratio	Proportion of a patient's encounters that were preventive.	encounters	derived
unique_encounter_types	Number of distinct encounter types recorded for a patient.	encounters	derived

4. Methodology and Master Dataset Preparation

4.1 Technical Stack

This project leverages a modern and efficient data analytics stack designed to handle large-scale healthcare data and deliver actionable insights:

- **PostgreSQL**

  PostgreSQL served as the primary relational database management system, used to store, query, and integrate the 10 Synthea-generated healthcare CSV files. Data was loaded and managed via pgAdmin 4, and SQL was used extensively for early-stage data validation, referential integrity checks, and foundational cleaning.

- **Python Ecosystem**

  Python was the primary tool for data wrangling, feature engineering, modeling, and evaluation. Key libraries and packages used include:

  - **Data Manipulation and Analysis:**

    pandas, numpy, and sqlalchemy were used for connecting to PostgreSQL, transforming data into pandas DataFrames, and performing complex patient-level aggregations.

  - **Data Visualization:**

    seaborn and matplotlib.pyplot enabled exploratory visualizations throughout the analysis. Display settings were customized for clarity using IPython.display.

  - **Data Preprocessing:**

    Libraries such as StandardScaler, MinMaxScaler, and OneHotEncoder from sklearn.preprocessing were used to normalize and encode features in preparation for modeling.

  - **Modeling and Evaluation:**

    - **Regression Models:** LinearRegression and KNeighborsRegressor from sklearn.linear_model and sklearn.neighbors were used to predict patient-level healthcare costs.
    - **Train-Test Splitting:** Conducted using train_test_split.
    - **Evaluation Metrics:** root_mean_squared_error was used to evaluate model performance.

    - **Statistical Analysis:** The statsmodels library provided support for statistical modeling, including coefficient significance and multicollinearity checks (VIF, p-values).

- **Tableau**

  Tableau was used for dashboarding. A series of interactive dashboards and visualizations were created to support strategic decision-making, showcasing cost distributions, preventive care trends, disease burden, and predicted vs. actual patient costs.

4.2 ETL Pipeline Overview

In this phase, raw data was extracted from a PostgreSQL database using SQLAlchemy and loaded into Python pandas DataFrames for downstream processing. The ETL process was structured to maintain clean, reproducible inputs and ensure consistency in primary keys and datetime formatting across all tables.

The pipeline focused on constructing a comprehensive patient-level master dataset by engineering clinically and operationally meaningful features across several dimensions:

- **Chronic Disease Tagging:** Patients were flagged for seven key disease groups (e.g., diabetes, cardiovascular, mental health) using curated keyword lists across diagnoses, procedures, and medications. Tagging was restricted to clinically relevant (non-preventive) encounters.
- **Disease Burden Stratification:** Duration and age at first diagnosis were calculated per disease group to assess chronic condition severity and onset timing.
- **Preventive Care Behavior:** Preventive engagement was quantified through encounter-level wellness flags, generating metrics such as preventive encounter counts and ratios.
- **Healthcare Utilization:** Overall interaction with the healthcare system was captured using metrics like total encounters and diversity of encounter types.
- **Cost Attribution:** Disease-specific and total healthcare costs were computed by aggregating costs from encounters, procedures, and medications, with attribution based on clinical relevance.

All feature engineering steps were aligned with the goal of enabling robust clinical, behavioral, and financial insights at the patient level, supporting both predictive modeling and Tableau dashboard development.

4.3 Final Patient Master Dataset

- **Shape:** (1,171 rows × 32 columns)
- **Number of features:** 32, covering patient demographics, chronic disease metrics, preventive care behavior, healthcare utilization, and cost breakdowns
- **Target variable:** total_cost – the total healthcare expenditure per patient, calculated as the sum of encounter, procedure, and medication costs

5. Patterns, Trends, and Insights

5.1 Descriptive Analytics

This section summarizes key patterns in age, gender, cost distribution, and disease prevalence among the patient population. These descriptive insights lay the foundation for understanding cost drivers and patient health profiles.

1. Age and Gender Distributions

- The median patient age is 41, with the population spanning from infants to the elderly (0–110).
- Most patients fall between the 20–59 age range, which accounts for over 70% of the population.
- Gender is relatively balanced, with 48% female (562 male, 609 female).
- Notably, females outnumber males in all age bands except the youngest (0–19), with the largest female majority in the 40–59 group.

2. Cost Trends by Demographic

- The median total cost per patient is approximately $104,623, with significant variation across demographics.
- On average, female patients incur higher total costs than males — $259,826 vs. $162,383 — indicating a possible link to care utilization, chronic conditions, or preventive service use.
- Preliminary visual analysis (via Tableau) suggests that older patients generally have higher total costs, with noticeable increases in the 60–79 and 80+ age bands.

- Cost component breakdowns (encounter, procedure, medication) vary across gender and age groups, providing insights into the nature of care received.

3. Disease Prevalence Breakdown

- The most prevalent chronic condition is hypertension, affecting 24.3% of patients — over half of whom are female (53.7%).
- Other common conditions include:
  - Cardiovascular disease: 17.6% (47.3% female)
  - Respiratory disorders: 12.3% (52.1% female)
  - Mental health conditions: 6.8%, with a striking 78.5% of cases female
  - Cancer: 6.6% (38.9% female)
  - Diabetes: 5.9% (47.8% female)
  - Kidney disease: 4.3% (44% female)
- Overall, multimorbidity is present, with a significant portion of the population affected by multiple chronic conditions, especially among high-cost patients.

5.2 Tableau Dashboards Overview

This section provides an overview of each Tableau dashboard, outlining its purpose and how it contributes to the broader analytical objectives of the project.

Dashboard A: Patient Cost Overview

➤ Displays major cost-related KPIs such as average, median, and total cost

➤ Shows total cost distribution across patients

➤ Visualizes average total cost segmented by age group and gender

Dashboard B: Cost Drivers & Patient Profiles

➤ Identifies high- and low-cost patient segments

➤ Breaks down cost composition (encounter, procedure, medication) by gender

➤ Links chronic disease presence with average annual costs

Dashboard C: Preventive Care Impact

➤ Compares average number of preventive visits by gender

➤ Examines the relationship between preventive ratio and total cost

➤ Highlights behavioral trends tied to lower overall costs

Dashboard D: Disease Prevalence by Demographics

➤ Illustrates the overall prevalence of chronic conditions

➤ Shows condition prevalence by gender (e.g., diabetes, hypertension)

➤ Aims to uncover disparities in disease burden across groups

Dashboard E: Predictive Model Performance

➤ Compares actual vs. predicted total cost using regression output

➤ Displays feature importance for key cost drivers

➤ Helps stakeholders understand model interpretability and accuracy

Dashboard F: Patient Cost Comparison by Demographics

➤ Side-by-side comparison of total cost by demographic features

➤ Useful for identifying at-risk or high-cost groups across age, gender, race

The Tableau Dashboard screenshots are included in **Section 8. Appendix D. Tableau Screenshots**.

6. Predictive Modeling

6.1 Problem Framing

The central goal of this analysis is to predict the total healthcare cost per patient, making this a regression problem. By estimating future healthcare expenditures using patient demographics, chronic conditions, encounter history, and preventive care behaviors, we aim to:

- Identify high-cost patients early for proactive care planning
- Support resource allocation by forecasting potential cost burdens
- Enable targeted interventions such as preventive care outreach or chronic disease management

This predictive modeling approach aligns with real-world healthcare goals of cost containment, improved patient outcomes, and efficient utilization of services. The output will be used not only for forecasting, but also for segmenting patients into cost-risk categories for further analysis and dashboarding.

6.2 Data Preprocessing

To prepare the patient-level data for regression modeling, we implemented the following preprocessing steps:

- **Train-Test Split:**

  The dataset was randomly split into training and testing sets (80:20 ratio) to evaluate model performance on unseen data.

- **Feature Selection:**

  A combination of domain knowledge and statistical analysis guided the selection of predictive features. Key variables included demographics (e.g., age, gender), chronic condition flags, encounter counts, cost composition, and preventive care metrics.

- **Categorical Encoding:**

  Categorical features such as gender and race were transformed using one-hot encoding to ensure compatibility with regression models.

- **Chronic Condition Flags:**

  Binary indicator variables were created for major chronic diseases (e.g., diabetes, hypertension, cancer), along with derived features such as age at first diagnosis and disease duration.

- **Log Transformation on Cost:**

  To address the right-skewed distribution of the total_cost target variable, a log transformation was applied. This helped stabilize variance and improve model fit.

- **Normalization:**

  Continuous features were scaled using min-max normalization to bring all variables to a similar range and reduce model sensitivity to varying feature magnitudes.

These preprocessing steps ensured the dataset was clean, interpretable, and ready for accurate and generalizable cost prediction modeling.

6.3 Model 1: Linear Regression

To establish a baseline for cost prediction, we trained a Linear Regression model using the preprocessed patient-level dataset.

Model Training:

We used Statsmodels OLS (Ordinary Least Squares) for model training to take advantage of detailed statistical output. The model was fit on the log-transformed total_cost target using selected features such as:

- Demographics (e.g., age, gender, race)
- Disease indicators
- Total Cost

Assumptions and Interpretation:

Linear regression assumes linearity and no multicollinearity:

- **Linearity:** Scatterplots between predictors and log(cost) showed reasonably linear relationships.
- **Multicollinearity:** Variance Inflation Factor (VIF) analysis confirmed acceptable levels (VIF < 5 for most features).

The model coefficients are interpretable, with positive values indicating a direct impact on cost.

Feature	Coefficient	Interpretation
age	+2.39	Older patients tend to have higher costs.
has_diabetes	+0.64	Diabetes increases the expected cost.
has_hypertension	+1.05	Strong positive effect on cost.
has_cardio	+0.54	Cardiovascular conditions increase costs.
has_respiratory	+0.87	Respiratory conditions are significant cost drivers.
has_cancer	+1.11	Cancer has a large impact on cost.
has_mental	+1.56	Mental health conditions have the highest positive impact.
has_kidney	-0.66	Unexpected negative coefficient — may reflect specific treatment patterns.
gender_M	-0.76	Male patients are associated with lower predicted costs.
race_white	Not significant	p = 0.124, so not statistically different from reference group.

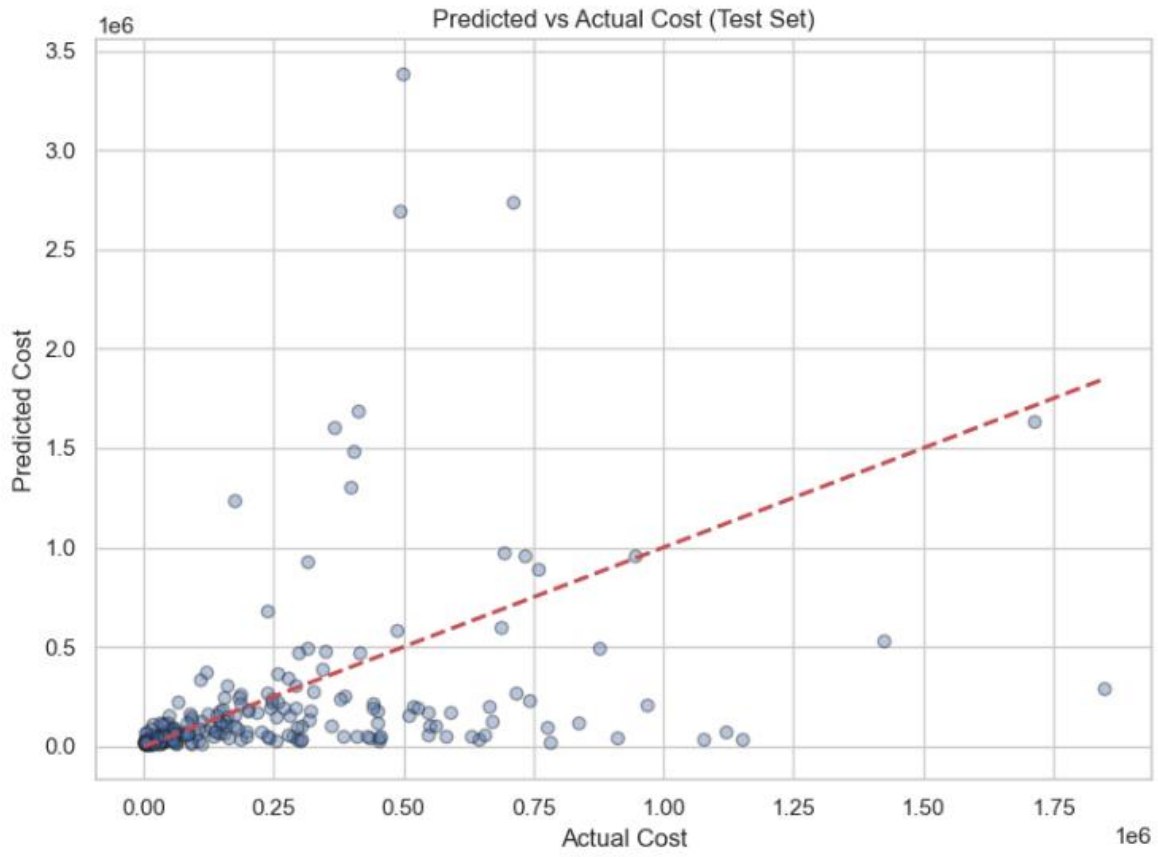
Model Evaluation:

- **R-squared = 0.483**: The model explains approximately 48.3% of the variance in log cost — a moderate model fit.
- **F-statistic = 86.30 (p < 0.001)**: The overall model is statistically significant.
- **Adjusted R-squared = 0.477**: Slightly lower due to adjustment for the number of predictors.
- **Observations = 936**: The number of training samples used in the model.

On the test set, we evaluated performance using standard metrics:

Train RMSE: 538,393.76

Test RMSE: 413,704.16



These results reflect the limitations of a linear approach in capturing complex, non-linear cost patterns. However, the model still provides transparency and a useful benchmark.

6.4 Model 2: K-Nearest Neighbors Regression

As a second approach, we implemented a K-Nearest Neighbors (KNN) Regression model using 5 neighbors to estimate each patient's total healthcare cost. KNN is a non-parametric, instance-based learning algorithm that predicts a value based on the average cost of the K most similar patients (i.e., nearest in feature space).

Model Setup:

We trained the model using KNeighborsRegressor from scikit-learn with:

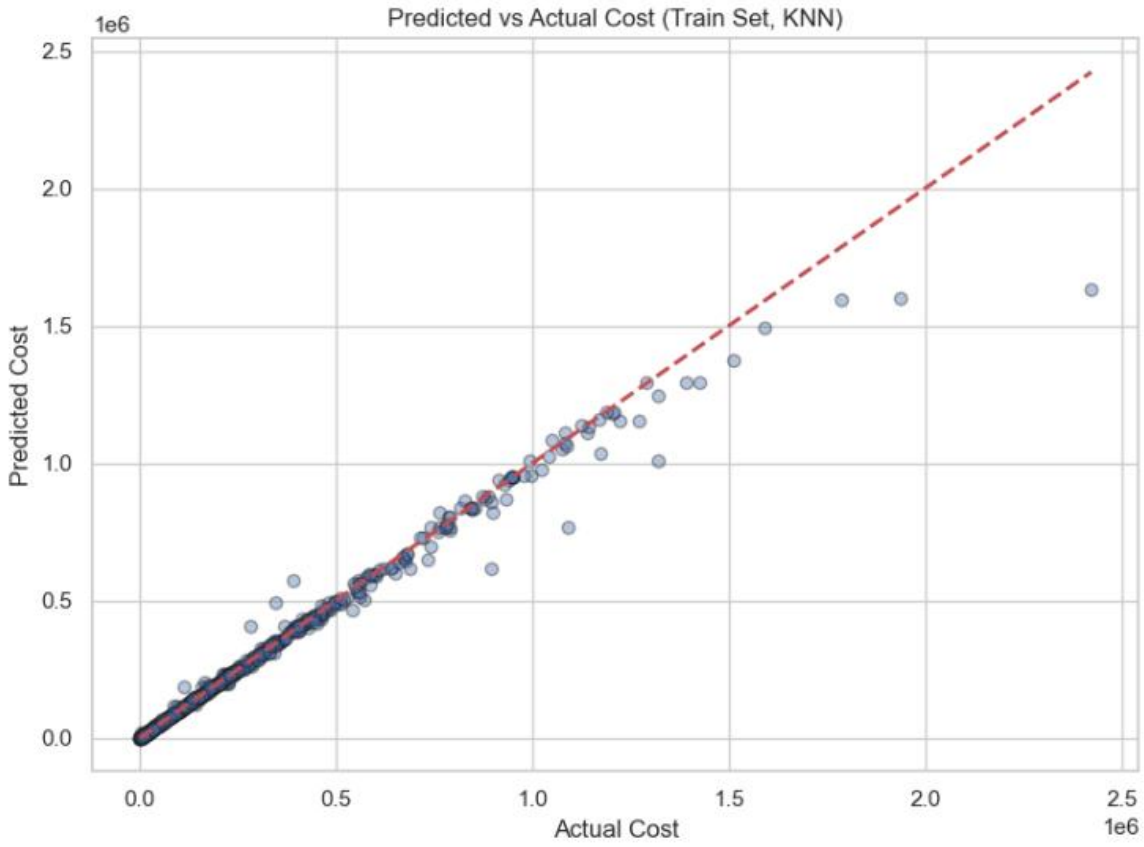
- n_neighbors = 5 (default choice without hyperparameter tuning)
- Euclidean distance as the similarity metric
- Standardized feature space (after normalization)

This approach assumes that patients with similar demographics, disease profiles, and healthcare behaviors will incur similar costs.

Model Evaluation:

We evaluated the KNN model on the test set using the same performance metrics as the previous Linear Regression model:

Train RMSE (KNN): 263,234.75
Test RMSE (KNN): 38,870.47



6.5 Model Comparison

Two regression models were developed and evaluated on their ability to predict total healthcare cost:

- **Linear Regression** – a simple, interpretable baseline using log-transformed cost
- **K-Nearest Neighbors (KNN)** – a non-parametric algorithm that leverages local patterns

While both models were trained and tested on the same dataset, final model selection is based solely on test set performance to ensure true generalization.

Model	Train RMSE	Test RMSE	Notes
Linear Regression	538,393.76	413,704.16	Baseline model with log(cost) transformation
K-Nearest Neighbors (KNN)	263,234.75	38,870.47	⭐ Lowest error on unseen data

Why did KNN perform best?

Unlike linear regression, which assumes a global relationship between features and cost, KNN uses *local pattern learning*. It compares each patient to their nearest neighbors in the training set and predicts cost based on the average cost of those similar patients. This makes it highly flexible and especially effective when costs are influenced by complex or nonlinear combinations of features.

Additionally, KNN's low test RMSE (38,870.47) suggests strong generalization and minimal overfitting. Its performance advantage, combined with the tight alignment in predicted vs actual costs, confirms its suitability as the final model for deployment.

7. Recommendations

7.1 Clinical and Operational Recommendations

Based on the insights uncovered through our predictive modeling, we recommend the following strategies to support both cost containment and quality care delivery:

Insight	Recommended Action
Mental health is the top cost driver (Patients with mental health conditions cost ~4.7× more)	**Invest in integrated behavioral health.** Embed mental health screenings in primary care, expand access to counseling, and assign behavioral health specialists to reduce crisis-driven ER visits and readmissions.
Male patients incur ~50% lower costs (Gender_M coefficient = −0.78)	**Investigate potential utilization gaps.** Explore whether men underutilize services or women face more complex conditions. Consider gender-sensitive outreach or preventive campaigns.
Age is the strongest cost predictor (~11× cost increase per normalized unit)	**Implement age-tiered care management.** Stratify patients into risk tiers (e.g., <40, 40–64, 65+) and tailor programs such as preventive screenings and chronic care coordination accordingly.

Insight	Recommended Action
Chronic diseases (e.g., cancer, cardiovascular) increase cost	**Target chronic disease management.** Use condition flags to enroll high-risk patients in care coordination or telehealth to prevent complications and hospitalizations.
Kidney disease associated with lower reported cost (−$9,295 impact)	**Audit coding and referral patterns.** Investigate potential underreporting or care gaps. Ensure patients with kidney issues receive appropriate nephrology consults and diagnostic follow-up.

These recommendations aim to bridge analytics insights with real-world interventions, helping hospital systems allocate resources more efficiently while improving patient outcomes.

7.2 Predictive Use Cases

This model can be operationalized in several practical ways:

- **Risk Scoring Dashboard**

  Integrate model outputs into dashboards that highlight high-cost or high-risk patients based on predicted cost scores.

- **Alerts in Electronic Health Records (EHR)**

  Embed predictive logic into the EHR to automatically flag patients for targeted care plans, social work referral, or follow-up appointments.

- **Population Health Monitoring**

  Track cost trends across disease groups, demographics, or facilities to inform strategic planning and preventive outreach.

7.3 Limitations & Challenges

While the project demonstrates strong predictive and interpretive value, several limitations remain:

- **Synthetic nature of the data:** The Synthea dataset, while realistic, is not derived from real-world patients, which may limit its generalizability.
- **Feature scope:** Key variables like lifestyle behaviors, social determinants, or granular utilization patterns were not available, which could affect prediction quality.

- **Scalability to real systems:** Integration into a hospital EHR or analytics system would require validation with real patient data and compliance with data governance rules.
- **Assumptions in preprocessing:** Certain assumptions were made when engineering features or encoding missing values, which may need re-evaluation in a live deployment.

Despite these limitations, the project demonstrates a solid proof-of-concept for predictive cost modeling in a healthcare setting.

7.4 Future Work

To improve both model performance and clinical relevance, future iterations could include:

- **More granular patient data:** Incorporate lifestyle factors, medication adherence, or social risk indicators for better predictions.
- **Temporal modeling:** Move beyond static prediction and apply longitudinal models (e.g., time-series regression) to forecast costs over time.
- **Larger datasets:** Train on larger anonymized real-world data for greater robustness.
- **Explainable ML techniques:** Use SHAP to provide transparency in black-box models like KNN, improving trust and adoption.

7.5 Practical Application Guide

To use the model in practice, the workflow would be:

1. **Input patient data:** Collect age, gender, and chronic condition indicators (e.g., cancer, diabetes, mental health).
2. Preprocess the inputs:
    - Normalize age
    - One-hot encode categorical variables (e.g., gender)
    - Format condition flags as binary
3. Run prediction:
    - Input data into the trained KNN regression model
    - Output: predicted log_cost
4. Convert to dollar value:
    - Use: predicted_cost = np.expm1(log_cost)

5. Take action:
    - Assign the patient to a cost-risk tier
    - Trigger care pathways, financial planning, or preventive interventions

This pipeline demonstrates how predictive modeling can support data-driven, proactive healthcare management — with just a few input features.

7.6 Final Reflections

Our modeling effort shows that even with a compact feature set, predictive analytics can offer significant value to healthcare operations:

- KNN delivered competitive accuracy but lacked interpretability.
- Linear regression offered transparency, helping us identify cost drivers like age and mental health.
- A hybrid approach (e.g., combining interpretable models with advanced learners like random forests) could enhance both precision and trust.

In conclusion, this project illustrates the power of using structured patient data to drive smarter resource allocation, targeted care interventions, and improved clinical outcomes — reinforcing the role of analytics in shaping the future of healthcare.

8. Appendix

A. Data Dictionary Table

Below are the data dictionaries of the original Synthea files:

patients.csv

Column Name	Description
Id	Primary Key. Unique Identifier of the patient.
BirthDate	The date the patient was born.
DeathDate	The date the patient died.
SSN	Patient Social Security identifier.
Drivers	Patient Drivers License identifier.
Passport	Patient Passport identifier.
Prefix	Name prefix, such as Mr., Mrs., Dr., etc.
First	First name of the patient.
Last	Last or surname of the patient.
Suffix	Name suffix, such as PhD, MD, JD, etc.
Maiden	Maiden name of the patient.
Marital	Marital Status. M is married, S is single.
Race	Description of the patient's primary race.
Ethnicity	Description of the patient's primary ethnicity.
Gender	Gender. M is male, F is female.
Birthplace	Name of the town where the patient was born.
Address	Patient's street address without commas or newlines.
City	Patient's address city.
State	Patient's address state.
County	Patient's address county.
ZIP	Patient's zip code.
Lat	Latitude of Patient's address.
Lon	Longitude of Patient's address.
Healthcare_Expenses	The total lifetime cost of healthcare to the patient (i.e. what the patient paid).
Healthcare_Coverage	The total lifetime cost of healthcare services that were covered by Payers (i.e. what the insurance company paid).

encounters.csv

Column Name	Description
Id	Primary Key. Unique Identifier of the encounter.
Start	The date and time the encounter started.
Stop	The date and time the encounter concluded.
Patient	Foreign key to the Patient.
Organization	Foreign key to the Organization.
Provider	Foreign key to the Provider.
Payer	Foreign key to the Payer.
Encounterclass	The class of the encounter, such as ambulatory, emergency, inpatient, wellness, or urgentcare.
Code	Encounter code from SNOMED-CT.
Description	Description of the type of encounter.
Base_Encounter_Cost	The base cost of the encounter, not including any line item costs related to medications, immunizations, procedures, or other services.
Total_Claim_cost	The total cost of the encounter, including all line items.
Payer_Coverage	The amount of cost covered by the Payer.
Reasoncode	Diagnosis code from SNOMED-CT, only if this encounter targeted a specific condition.
Reasondescription	Description of the reason code.

conditions.csv

Column Name	Description
Start	The date the condition was diagnosed.
Stop	The date the condition resolved, if applicable.
Patient	Foreign key to the Patient.
Encounter	Foreign key to the Encounter when the condition was diagnosed.
Code	Diagnosis code from SNOMED-CT.
Description	Description of the condition.

procedures.csv

Column Name	Description
Date	The date and time the procedure was performed.
Patient	Foreign key to the Patient.
Encounter	Foreign key to the Encounter where the procedure was performed.
Code	Procedure code from SNOMED-CT.
Description	Description of the procedure.
Base_Cost	The line item cost of the procedure.
Reasoncode	Diagnosis code from SNOMED-CT specifying why this procedure was performed.
Reasondescription	Description of the reason code.

medications.csv

Column Name	Description
Start	The date and time the medication was prescribed.
Stop	The date and time the prescription ended, if applicable.
Patient	Foreign key to the Patient.
Payer	Foreign key to the Payer.
Encounter	Foreign key to the Encounter where the medication was prescribed.
Code	Medication code from RxNorm.
Description	Description of the medication.
Base_Cost	The line item cost of the medication.
Payer_Coverage	The amount covered or reimbursed by the Payer.
Dispenses	The number of times the prescription was filled.
Totalcost	The total cost of the prescription, including all dispenses.
Reasoncode	Diagnosis code from SNOMED-CT specifying why this medication was prescribed.
Reasondescription	Description of the reason code.

observations.csv

Column Name	Description
Date	The date and time the observation was performed.
Patient	Foreign key to the Patient.
Encounter	Foreign key to the Encounter where the observation was performed.
Code	Observation or Lab code from LOINC.
Description	Description of the observation or lab.
Value	The recorded value of the observation. Often numeric, but some values can be verbose, for example, multiple-choice questionnaire responses.
Units	The units of measure for the value.
Type	The datatype of Value: text or numeric

allergies.csv

Column Name	Description
Start	The date the allergy was diagnosed.
Stop	The date the allergy ended, if applicable.
Patient	Foreign key to the Patient.
Encounter	Foreign key to the Encounter when the allergy was diagnosed.
Code	Allergy code.
Description	Description of the Allergy.

careplans.csv

Column Name	Description
Id	Primary Key. Unique Identifier of the care plan.
Start	The date the care plan was initiated.
Stop	The date the care plan ended, if applicable.
Patient	Foreign key to the Patient.
Encounter	Foreign key to the Encounter when the care plan was initiated.
Code	Code from SNOMED-CT.
Description	Description of the care plan.
Reasoncode	Diagnosis code from SNOMED-CT that this care plan addresses.
Reasondescription	Description of the reason code.

providers.csv

Column Name	Description
Id	Primary key of the Provider/Clinician.
Organization	Foreign key to the Organization that employees this provider.
Name	First and last name of the Provider.
Gender	Gender. M is male, F is female.
Speciality	Provider speciality.
Address	Provider's street address without commas or newlines.
City	Street address city.
State	Street address state abbreviation.
Zip	Street address zip or postal code.
Lat	Latitude of Provider's address.
Lon	Longitude of Provider's address.
Utilization	The number of Encounters performed by the Provider.

organizations.csv

Column Name	Description
Id	Primary key of the Organization.
Name	Name of the Organization.
Address	Organization's street address without commas or newlines.
City	Street address city.
State	Street address state abbreviation.
Zip	Street address zip or postal code.
Lat	Latitude of Organization's address.
Lon	Longitude of Organization's address.
Phone	Organization's phone number. Sometimes multiple numbers.
Revenue	The monetary revenue of the organization during the entire simulation.
Utilization	The number of Encounters performed by this Organization.

B. SQL Code Snippets

SQL Code Snippet to create the databse and tables

```
-- Create the healthcarecost database
CREATE DATABASE healthcarecost;

-- 3. Create conditions table according to conditions.csv
CREATE TABLE conditions (
        conditions_id SERIAL PRIMARY KEY,
        start TIMESTAMP,
        stop TIMESTAMP,
        patient TEXT REFERENCES patients(patient_id),
        encounter TEXT REFERENCES encounters(encounter_id),
        code TEXT,
        description TEXT
);
```

SQL Code Snippet to check for Null / Missing values

```
-- allergies table (only NULL values in stop column which is OK)
SELECT
   COUNT(*) AS total_rows,
   COUNT(allergy_id) AS allergy_id_not_null,
   COUNT(start) AS start_not_null,
```

```
    COUNT(stop) AS stop_not_null,
    COUNT(patient) AS patient_not_null,
    COUNT(encounter) AS encounter_not_null,
    COUNT(code) AS code_not_null,
    COUNT(description) AS description_not_null
FROM allergies;
```

SQL Code Snippet to verify consistency in Data types

```
SELECT
    table_name,
    column_name,
    data_type
FROM
    information_schema.columns
WHERE
    table_schema = 'public'
ORDER BY
    table_name, column_name;
```

SQL Code Snippet to add column and extract numeric values

```
ALTER TABLE observations
ADD COLUMN value_numeric numeric;

UPDATE observations
SET value_numeric = value::numeric
WHERE type = 'numeric' AND value ~ '^[-]?[0-9]+(\.[0-9]+)?$';
```

SQL Code Snippet for Referential Integrity Validation

```
--  3. providers.organization referencing organizations.organization_id
(Successful)
SELECT p.organization
FROM providers p
LEFT JOIN organizations o ON p.organization = o.organization_id
WHERE o.organization_id IS NULL;
```

SQL Code Snippet for Verification and Handling of Duplicates

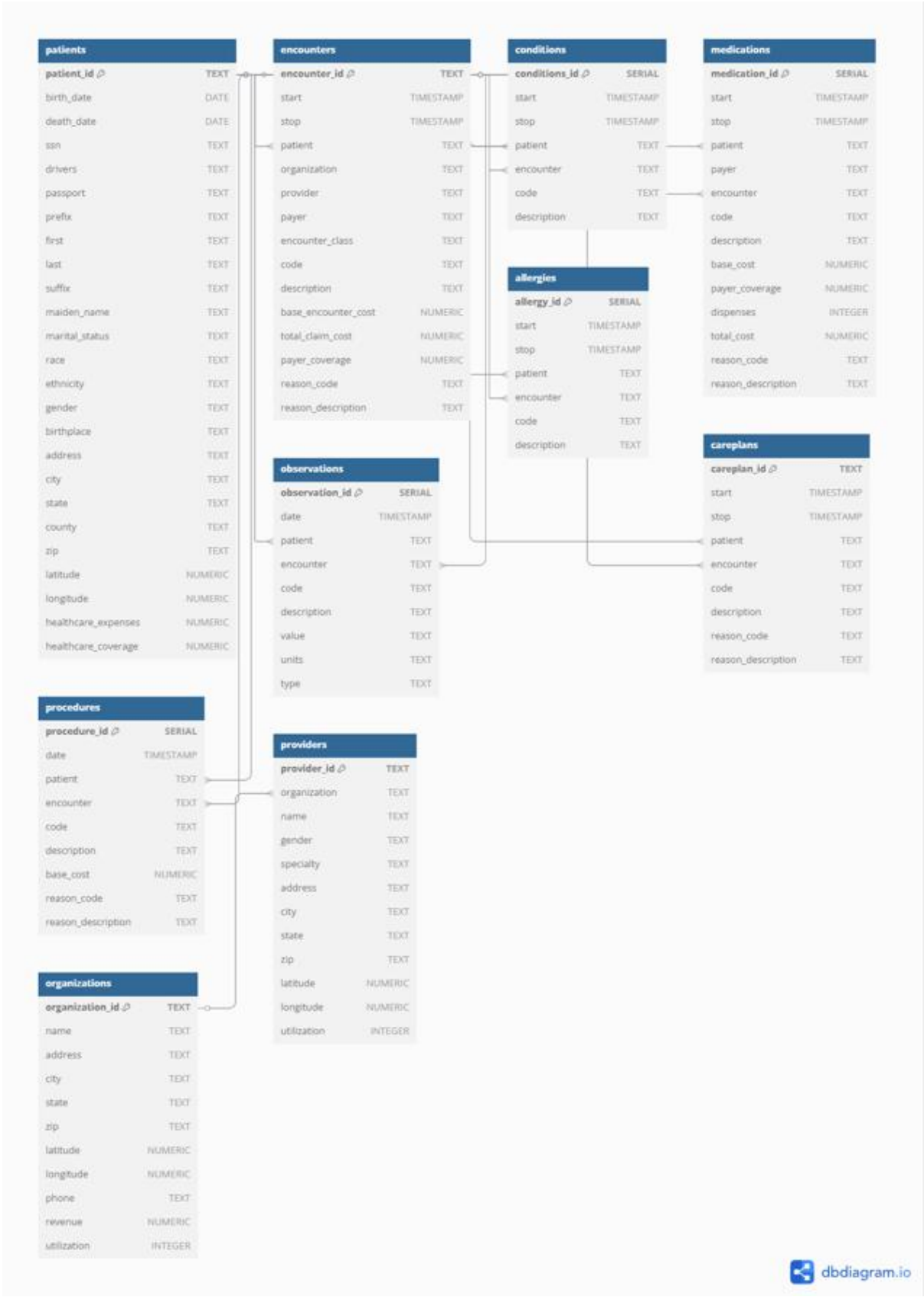
```
-- 5. observations_cleaned table (442 duplicates found)
SELECT patient, encounter, code, description,
          date, value, units, type, COUNT(*)
FROM observations_cleaned
GROUP BY patient, encounter, code, description, date, value, units,
type
```

```
HAVING COUNT(*) > 1;

DELETE FROM observations_cleaned
WHERE ctid NOT IN (
    SELECT MIN(ctid)
    FROM observations_cleaned
    GROUP BY patient, encounter, code, description, date, value, units,
type
);
```

SQL Code Snippet for Verification of Categorical Standardization and Logical Correctness

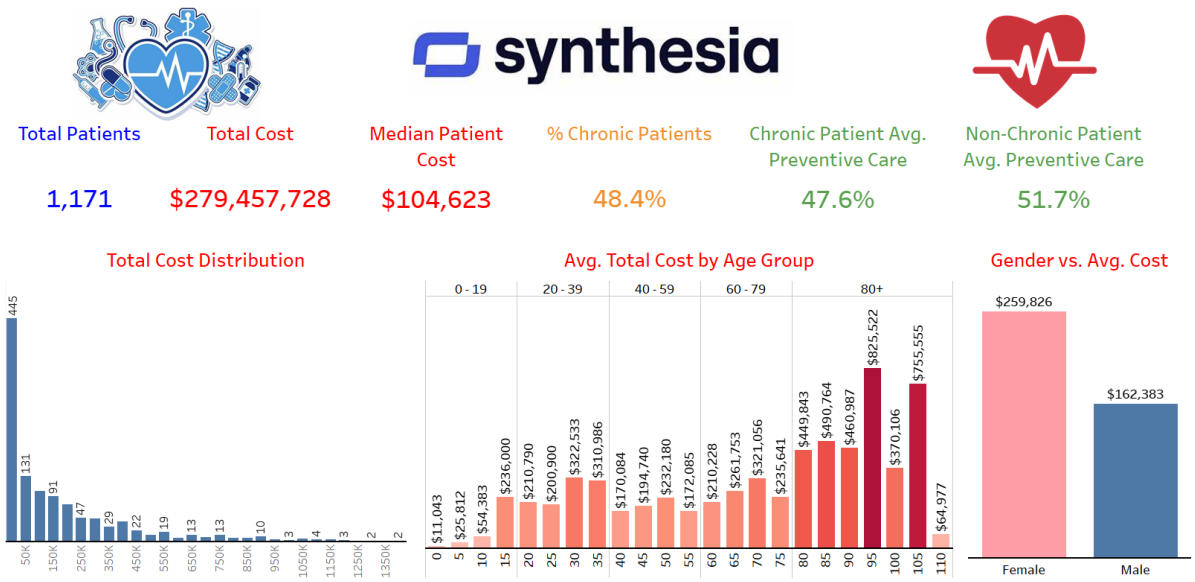
```
SELECT *
FROM conditions
WHERE stop < start;

UPDATE medications
SET stop = start
WHERE stop < start;


SELECT *
FROM patients
WHERE death_date IS NOT NULL AND death_date < birth_date;

-- Incorrect data in provider names.

-- We use Regular Expression with following codes:
-- '\d+', '', 'g'     \d+  replaces all digits with '' for the entire string 'g'
(global flag)
-- '\s+', ' ', 'g'    \s+  replaces multiple spaces with a single space ' ' for
the entire string 'g' (global flag)

UPDATE providers
SET name = REGEXP_REPLACE(REGEXP_REPLACE(name, '\d+', '',
'g'), '\s+', ' ', 'g');
```

C. **ERD Diagram**

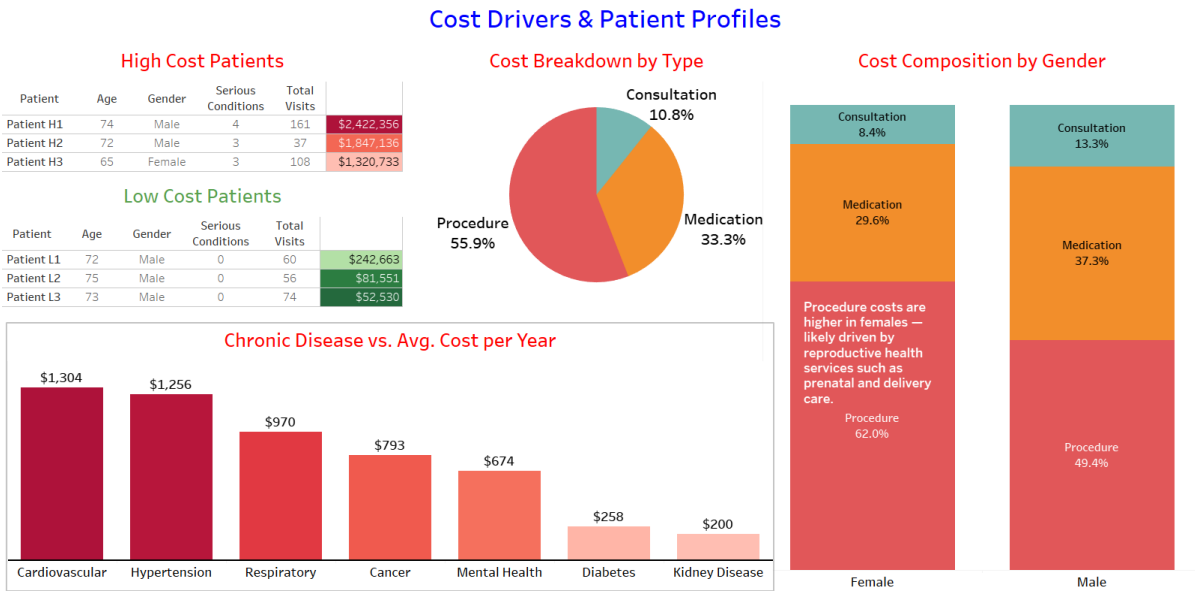


D. **Tableau Screenshots**

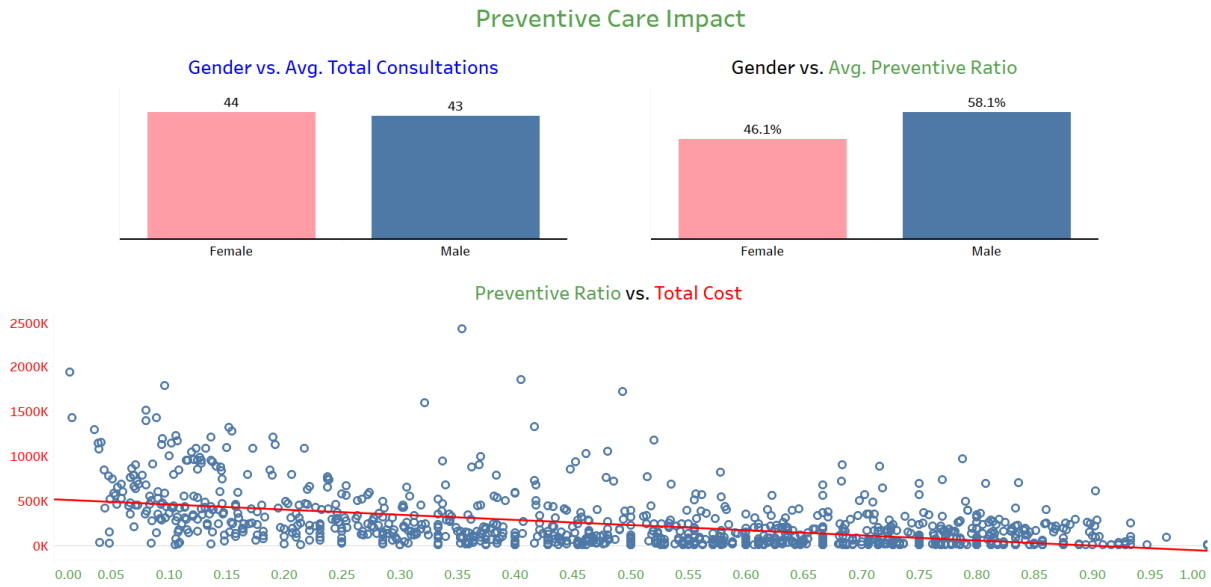
Dashboard A: Patient Cost Overview



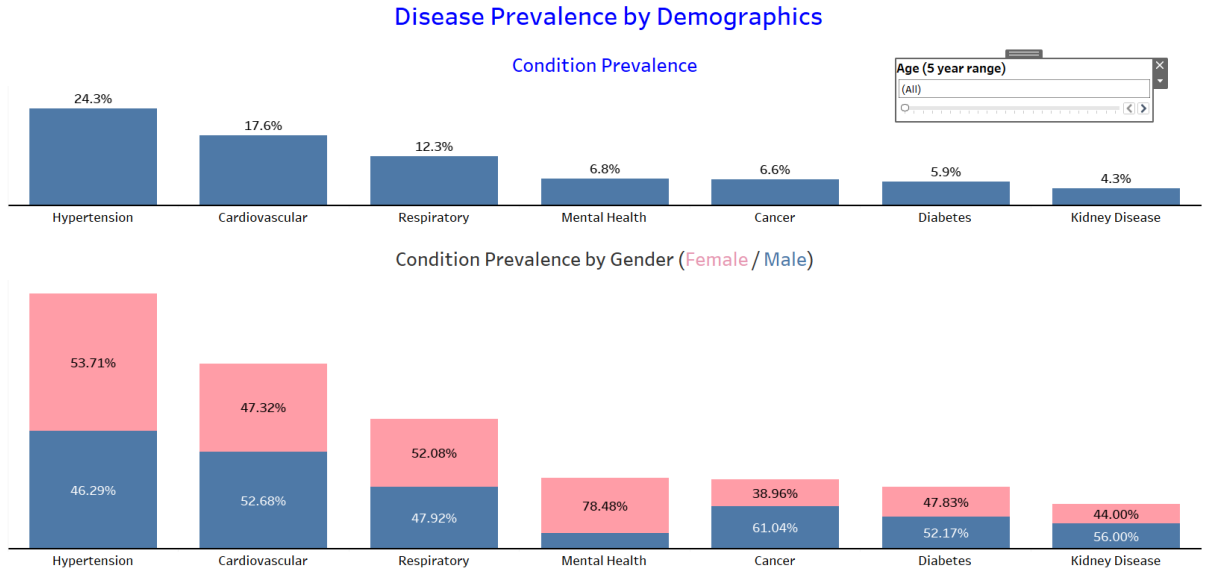
Dashboard B: Cost Drivers & Patient Profiles



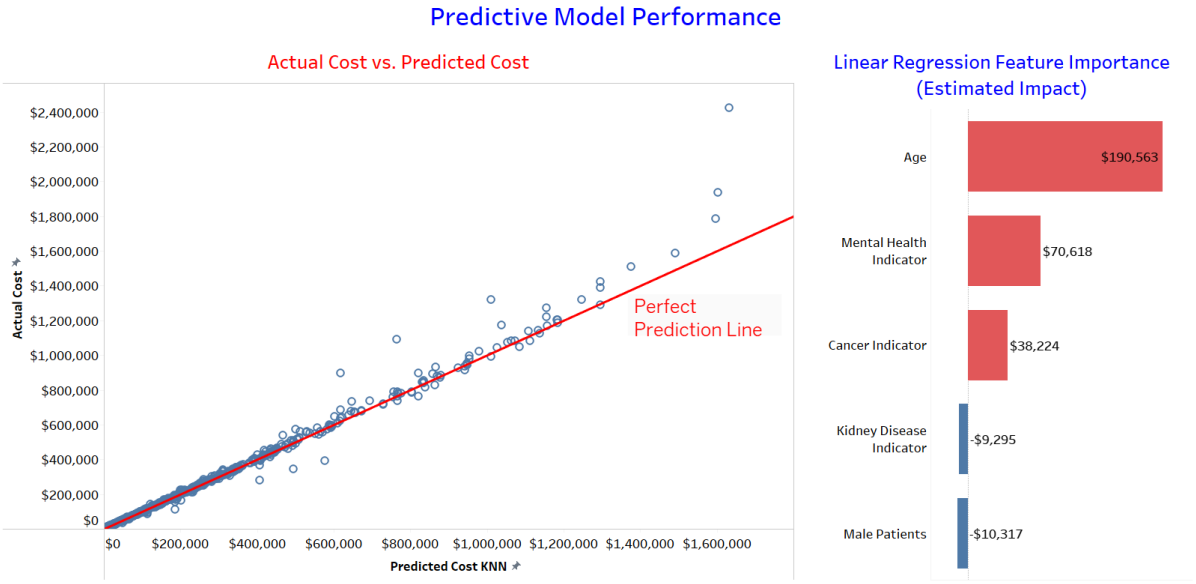
Dashboard C: Preventive Care Impact



Dashboard D: Disease Prevalence by Demographics



Dashboard E: Predictive Model Performance



Dashboard F: Patient Cost Comparison by Demographics