Raed K Alotaibi

raedkma@gmail.com

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

Analysis of Diabetic Population Claims Data

A report

Okay lets now structure the analysis report. It should contain the following sections:

* Executive summary
* Introduction to the project [aims, objectives including demonstrating the applicants analytic and technological capabilities (we will include a skills matrix the would be reflected in the steps of the report). –
* Preprocessing steps
* Descriptive analytics summarizing the data including
  + data set size,
  + column description,
  + column types,
  + numeric and categorical summaries including
  + missing values and duplicate rows] –
* Data quality and plausibility with suggested changes to enhance the data
  + [ e.g. member code is not a unique id with reasons,
  + no time stamps to indicate the trend in outcome or other measures like BMI,
  + claim type with an "I" is a duplicate with an "O" value row] –
* Feature engineering and creation of a feature store prior to advanced analytics –
* Patterns and trends analysis
* Advanced analytics: [
  + hypothesis generation and testing,
  + clustering analysis to identify similar grouping,
  + other machine learning]
* Strategies to address gaps, insights and reflections
* Describe the methodology and tools used:
  + Project setup and version control
  + Data processing & Analytics:
  + Visualization
  + Machine learning
  + AI & LLM: DeepSeek, Agentic flow, Langflow for architecture
  + Front end: streamlit
* Future Directions & Enterprise -level suggestions

Contents

[Analysis of Diabetic Population Claims Data 3](#_Toc189042251)

[1. Executive Summary 3](#_Toc189042252)

[2. Introduction 4](#_Toc189042253)

[2.1 Project Aim 4](#_Toc189042254)

[2.2 Objectives 4](#_Toc189042255)

[2.3 Skills Matrix 4](#_Toc189042256)

[3. Preprocessing Steps 5](#_Toc189042257)

[3.1 Data Ingestion 5](#_Toc189042258)

[3.2 Data Cleaning 6](#_Toc189042259)

[3.3 Data Transformation 7](#_Toc189042260)

[3.3 Data Quality checks 7](#_Toc189042261)

[4. Descriptive Analytics 8](#_Toc189042262)

[4.1 Dataset Summary 8](#_Toc189042263)

[4.2 Numeric Data Summary 8](#_Toc189042264)

[4.3 Categorical Data Summary 8](#_Toc189042265)

[5. Data Quality and Plausibility Assessment 9](#_Toc189042266)

[Suggested Enhancements: 9](#_Toc189042267)

[6. Feature Engineering & Feature Store Creation 10](#_Toc189042268)

[7. Patterns and Trends Analysis 11](#_Toc189042269)

[8. Advanced Analytics 12](#_Toc189042270)

[8.1 Hypothesis Generation & Testing 12](#_Toc189042271)

[8.2 Clustering Analysis 12](#_Toc189042272)

[8.3 Predictive Modeling 12](#_Toc189042273)

[9. Addressing Gaps, Insights, and Reflections 13](#_Toc189042274)

[10. Methodology & Tools Used 14](#_Toc189042275)

[11. AI Capabilities & Agentic Flows 15](#_Toc189042276)

[12. Future Directions & Enterprise-Level Suggestions 16](#_Toc189042277)

# **Analysis of Diabetic Population Claims Data**

## **1. Executive Summary**

This report presents an in-depth analysis of a diabetic population claims dataset to assess data quality, identify key trends, and apply advanced analytical techniques. The report outlines preprocessing steps, descriptive analytics, feature engineering, machine learning applications, and strategic recommendations for improving data usability and deriving actionable insights. The findings will support better decision-making in healthcare claims management.

## **2. Introduction**

### **2.1 Project Aim**

The primary goal of this project is to analyze claims data for a diabetic population, highlighting key trends, data quality issues, and predictive insights. The analysis will demonstrate expertise in data processing, feature engineering, and advanced analytics, ultimately supporting enhanced claims processing and risk assessment.

### **2.2 Objectives**

* **Demonstrate analytical and technological capabilities** through systematic data exploration and predictive modeling.
* **Enhance data quality** by identifying inconsistencies and proposing corrective measures.
* **Uncover significant patterns and trends** in the diabetic population's claims.
* **Develop machine learning models** to generate predictive insights.
* **Recommend strategies** to improve data integrity and analytics adoption in enterprise settings.

### **2.3 Skills Matrix**

|  | Application in Report |
| --- | --- |
| Data Cleaning | Handling missing values, duplicates, and plausibility checks |
| Descriptive Analytics | Summarizing dataset attributes, distribution, and trends |
| Data Quality Assessment | Identifying inconsistencies and proposing enhancements |
| Feature Engineering | Creating structured and meaningful features for modeling |
| Advanced Analytics | Applying clustering, predictive modeling, and hypothesis testing |
| AI & LLM Integration | Exploring retrieval-based AI insights for structured and unstructured data |

## **3. Preprocessing Steps**

### **3.1 Data Ingestion**

The **raw layer** ingests CSV files while enforcing schema validation. This ensures that each column maintains the correct data type, preventing inconsistencies in downstream processing. The data is then converted to **Parquet format** for storage efficiency and type preservation.

#### **Schema Validation and Column Standardization**

To ensure **data integrity**, the ingestion process applies:

* **Explicit data type enforcement** using a predefined dictionary.
* **Column renaming** for consistency across analysis stages.
* **Initial data validation** to verify data types and structure.

##### **Code Implementation:**

# Define the data types for each column in the dataset

dtype\_dict = {

    "MEMBER\_CODE": "int64",    # De-identified member ID, stored as float to match dataset format

    "Age": "int64",             # Age of the member

    "GENDER": "category",       # Gender is a categorical variable

    "POLICY\_NO": "int64",       # Policy number, stored as integer

    "CMS\_Score": "int64",       # Charlson comorbidity index score, stored as integer

    "ICD\_CODE": "category",     # ICD-10 codes are categorical

    "ICD\_desc": "string",       # ICD-10 description as a string

    "City": "string",           # City as a string, handling missing values separately

    "CLAIM\_TYPE": "category",   # Claim type is categorical

    "BMI": "float64"            # BMI as a float

}

# Column renaming lookup table

column\_lookup = {

    "MEMBER\_CODE": "member\_code",

    "Age": "age",

    "GENDER": "gender",

    "POLICY\_NO": "policy\_number",

    "CMS\_Score": "cms\_score",

    "ICD\_CODE": "icd\_code",

    "ICD\_desc": "icd\_description",

    "City": "city",

    "CLAIM\_TYPE": "claim\_type",

    "BMI": "bmi"

}

# Load the dataset with specified data types

raw\_data = pd.read\_csv(raw\_data\_path, dtype=dtype\_dict)

# Rename columns using the lookup table

raw\_data.rename(columns=column\_lookup, inplace=True)

#### **Conversion to Parquet for Type Preservation**

After schema validation, the dataset is **saved in Parquet format** to maintain column types across different processing layers:

# Save the cleaned data to Parquet format for efficient storage and column type preservation

raw\_data.to\_parquet("..\\data\\raw\\diabetic\_claims.parquet", index=False)

### **3.2 Data Cleaning**

To ensure data consistency and integrity, a series of data cleaning steps were performed, including handling missing values, identifying and removing duplicates, and validating the dataset.

#### **1. Identifying Missing Data**

A summary of missing values was generated to assess completeness. The city column was identified as having **4,700 missing values**, while other critical fields remained largely intact.

#### **2. Handling Missing Data**

The missing values in the city column were replaced with "Unknown" to retain records without introducing bias. Other imputation strategies, such as mean or mode imputation, were considered but were not required at this stage.

#### **3. Checking for Duplicates**

Duplicate records were reviewed to prevent redundant information.

* **Duplicate rows found:** **2**, which were removed.
* The claim\_type **column** exhibited duplication where records with an **'I' value** had an identical counterpart with an **'O' value**.
  + **34,233** out of **34,235** records with claim\_type = I had an identical entry with claim\_type = O.
  + These inconsistencies were flagged for further review and correction in subsequent processing stages.

#### **4. Data Validation**

Following the cleaning steps, a final validation was conducted to confirm:

* **No remaining missing values in critical columns.**
* **No duplicate rows** affecting dataset integrity.
* **Flagging of** claim\_type **anomalies** for further processing.

This ensures the dataset is now **structured, consistent, and ready for transformation** in the next stage.

### **3.3 Data Transformation**

Following data cleaning, **data transformation** ensures consistency, usability, and alignment with downstream analytical processes. Key transformations include converting categorical values into structured indicators and standardizing textual information.

#### **1. Converting Claim Type into an Indicator**

The claim\_type column contained duplicate records where **'I' (initial claim) and 'O' (other claim) values** represented the same entry.

* A transformation function was applied to **convert the 'I' value into an indicator column**, ensuring unique records while preserving relevant information.
* Duplicate rows arising from this transformation were removed to prevent redundancy.

#### **2. Normalization Process: Standardizing City Names**

City names were **normalized** to ensure consistency and avoid discrepancies in grouping and analysis.

* Variations in formatting, capitalization, and spelling were addressed.
* The transformation process included:
  + **Trimming whitespace** to remove unnecessary spaces.
  + **Converting names to title case** (e.g., "RIYADH" → "Riyadh").
  + **Correcting hyphenated names** for uniformity (e.g., "'AL-AHASA" → " Al-Ahasa").
* A **lookup table** was created to store and maintain standardized city names for **reusability in downstream processes and documentation.**

These transformations ensure a **clean and structured dataset**, enabling accurate analysis and modeling in later stages.

### **3.4 Data Quality checks**

To ensure the dataset maintains **logical consistency and accuracy**, a series of **data validation checks** were conducted. These checks included **logical value constraints, member data consistency assessments, and ICD code standardization**. Any identified issues were documented for further analysis.

#### **1. Logical Value Checks**

Key numerical and categorical variables were validated against predefined logical ranges:

* age: Ensured all values fell within a plausible range (**0–120 years**).
* bmi: Confirmed all BMI values were within a reasonable range (**10–80**).
* gender: Ensured only expected categorical values (**"M" or "F"**) were present.

#### **2. Identifying Family Units Using** member\_code

A unique observation in the dataset was that some member\_code **values were shared across multiple individuals** with different **ages and genders**. This suggests that **the same** member\_code **may represent a family unit** rather than an error in the dataset. For example records showed **male and female individuals with different ages under the same** member\_code, supporting the hypothesis of **family grouping within policies**. The creation of a unique identifier for an individual is described later in the feature engineering section.

#### **3. ICD Code and Description Lookup**

A **lookup table** was created for icd\_code and icd\_description to:

* Identify **duplicate or conflicting mappings**.
* Ensure consistency in ICD-10 codes across records.
* Detect **missing or mismatched descriptions** requiring correction.

#### **4. Intermediate Quality Report & Documentation**

All **flagged quality issues** were documented and saved for further analysis. This included:

* **Entries with out-of-range values** (e.g., unrealistic BMI or age values).
* **Household-based** member\_code **groupings** for further validation.
* **ICD code standardization inconsistencies**.

A summary of **member\_code groupings** was saved in non\_unique\_member\_codes.csv for **further verification** and future analytical segmentation.

## **4. Feature Engineering & Feature Store**

### **4.1 Overview of Data Transformations & Feature Engineering**

This section outlines the **structured process of transforming raw data into analytical features**, ensuring consistency, standardization, and usability for downstream modeling and analysis. The approach follows a **layered transformation method**, converting raw attributes into **structured feature tables**.

**Note:** Feature engineering is an **iterative process** that evolves based on exploratory analysis. While this report presents feature creation before descriptive analytics for clarity, the actual process involved analyzing data gaps, transforming variables, and iterating based on insights.

### **4.2 Key Data Transformations & Feature Creation**

#### **1. Creating Unique Identifiers**

To ensure **accurate individual tracking** while preserving privacy, a **unique identifier** was created by combining:

* **policy\_number**
* **member\_code**
* **age**
* **gender**

This identifier allows differentiation of **individuals within the same policy** while enabling household-level analysis.

#### **2. Numerical to Categorical Transformations**

To enhance interpretability and improve modeling performance, numerical variables were **categorized** into meaningful groups:

| **Feature** | **Categories** |
| --- | --- |
| **Age Group (age\_group)** | 10-year intervals (e.g., 0-9, 10-19, 20-29, …, 80+) |
| **BMI Category (bmi\_cat)** | - Underweight (<18.5)  - Healthy (18.5–24.9)  - Overweight (25–29.9)  - Obese (≥30) |
| **Obesity Class (obesity\_cat)** | - Class 1 (30–34.9)  - Class 2 (35–39.9)  - Class 3 (≥40) |

These categorizations allow for **comparative risk analysis** across different patient groups.

#### **3. Standardizing City Names**

To ensure **consistency in geographic analysis**, city names were normalized by:

* **Trimming whitespace** to remove unwanted spaces.
* **Converting names to title case** (e.g., "RIYADH" → "Riyadh").
* **Correcting hyphenation inconsistencies** (e.g., "Al Khobar" → "Al-Khobar").
* **Creating a lookup table** to store standardized city names for reusability.

Additionally, a **"Major City"** feature was introduced, categorizing cities based on the **top 5 cities by unique patient count.**

#### **4. Converting** claim\_type **into an Indicator**

A transformation was applied to the claim\_type variable to **convert ‘I’ values into an indicator** while handling duplicate rows:

* **‘I’ (Initial Claim) was transformed into a binary indicator variable.**
* Duplicate entries caused by claim\_type variations were **identified and removed** to ensure data integrity.

### **4.3 Extracting Feature Tables**

To enhance data accessibility, key **feature tables** were created for streamlined analysis:

| **Feature Table** | **Description** |
| --- | --- |
| **Diabetes Type Table** | Aggregates diabetes classification based on ICD-10 codes per unique identifier. |
| **Comorbidity Table** | Stores Charlson Comorbidity Index scores per unique identifier. |
| **Diabetes Feature Table** | Captures diabetes-specific indicators, including treatment intensity. |
| **Family Size Table** | Groups members under the same policy to determine household size. |
| **Unique Identifier Table** | Stores transformed IDs for downstream merging and validation. |

These feature tables enable **segmentation, trend analysis, and machine learning applications**.

### **4.4 Storing Features for Reuse**

* Engineered features were **saved in structured feature tables** to **facilitate efficient retrieval, reuse, and analysis.**
* Outputs were stored in a **feature store** using **Parquet format** for optimized storage and column type consistency.

## **5. Descriptive Analytics**

## **5.1. Introduction**

Descriptive analytics is the first step in understanding the dataset by summarizing its structure, characteristics, and key trends. This section provides an overview of the available dataset, identifies missing values and potential inconsistencies, and explores key population and claims-based statistics.

By analyzing demographic distributions, claim frequencies, and prevalence rates of diabetes-related conditions, we aim to derive insights into healthcare utilization patterns among diabetic patients. The following subsections will systematically explore the dataset’s composition and its implications for further analysis.

## **5.2. Overview of the Dataset and Extracted Feature Tables**

### **5.2.1 Primary Dataset**

The analysis is based on a single **primary dataset (primary\_data.csv)**, which contains de-identified healthcare claims data for diabetic patients. It includes:

* **Demographics:** Age, gender, city, policy information.
* **Claims Data:** Claim type, ICD-10 codes, comorbidities, and complications.
* **Health Indicators:** BMI, Charlson comorbidity index scores.
* **Unique Identifiers:** De-identified member and policy codes.

### **5.2.2 Extracted Feature Tables**

To facilitate analysis, several **feature tables** were derived from the primary dataset:

* **Diabetes Type (diabetes\_type\_feature.csv)** – Diabetes type indicator per patient based on ICD-10 codes.
* **Comorbidity Scores (comorbidity\_feature.csv)** – Comorbidity indicator per patient based on ICD-10 codes.
* **Diabetes Complications (diabetes\_complication\_feature.csv)** – Diabetes-related complications per patient based on ICD-10 codes.
* **Family Size (family\_size\_table.csv)** – Groups members under the same policy to infer household structures.
* **ICD Lookup (icd\_lookup.csv)** – Standardized ICD-10 codes and descriptions.
* **City Lookup (city\_lookup.csv)** – Provides cleaned and standardized city names.

These feature tables were created to streamline analytical workflows and enhance interpretability.

## **5.3. Dataset Characteristics and Data Quality Assessment**

### **3.1 Structure of the Dataset**

* **Number of Observations and Features**
  + Count of rows (observations) and columns (features).
  + Classification of variables as **numerical, categorical, or textual**.

## **4. Population Prevalence Analysis**

This section focuses on the **demographic distribution and prevalence of diabetes-related conditions**.

### **4.1 Methodology**

Prevalence rates are calculated as follows:

Prevalence=Count of Members in a GroupTotal Members×100\text{Prevalence} = \frac{\text{Count of Members in a Group}}{\text{Total Members}} \times 100Prevalence=Total MembersCount of Members in a Group​×100

### **4.2 Population Segmentation**

* **Age Groups** (e.g., 0-9, 10-19, …, 80+).
* **Gender Distribution** (Male/Female).
* **Major Cities** (using city\_lookup.csv).
* **Diabetes Type** (based on diabetes\_type\_feature.csv).
* **Number of Comorbidities** (from comorbidity\_feature.csv).
* **Diabetes Complications** (from diabetes\_complication\_feature.csv).

### **4.3 Summary Statistics**

* Total members per **age, gender, and city category**.
* **Prevalence rates** for diabetes types, comorbidities, and complications.
* Cross-tabulations of **age vs. diabetes type, comorbidity count vs. city**, etc.

## **5. Claims Analysis and Utilization Patterns**

This section examines **healthcare utilization trends** among diabetic patients based on claims data.

### **5.1 Claims Statistics by Policy and Individual**

* **Average Claim Count Per Individual** (mean, median, and distribution).
* **Average Claim Count Per Policy** (distribution of claims at the policy level).
* **Distribution of Claim Counts Across Major Cities** (comparison of claim volumes by location).

### **5.2 Comorbidity and Complication Claims**

* **Comorbidity Distribution Per Policy** – Grouping policies by the number of comorbid conditions per member.
* **Diabetes Complication Claims Per Policy** – Breakdown of claims related to complications.

## **6. Summary and Insights**

* **Key findings** from descriptive statistics.
* Data **quality concerns** (e.g., missing values, claim-type inconsistencies).
* **Potential areas for further analysis**, such as predictive modeling and risk stratification.

## **7. Patterns and Trends Analysis**

* **Distribution of Claim Types across Age Groups.**
* **BMI vs. Claim Frequency:** Identifying whether obesity increases claims likelihood.
* **Charlson Score & ICD-10 Trends:** High-risk conditions and frequent diagnoses.
* **Policy Number Analysis:** Potential clustering of members under specific policies.

## **8. Advanced Analytics**

### **8.1 Hypothesis Generation & Testing**

* Does BMI significantly influence the frequency of claims?
* Are certain ICD-10 codes associated with higher costs?
* Is there a seasonal trend in claims data (if timestamps exist)?

### **8.2 Clustering Analysis**

* Grouping members by claim frequency and comorbidity risk.
* Identifying outlier patterns in claim behavior.

### **8.3 Predictive Modeling**

* **Classification Models** (e.g., Decision Trees, Logistic Regression) to predict high-risk claimants.
* **Regression Models** to estimate claim costs.

## **9. Addressing Gaps, Insights, and Reflections**

* **Data Enhancement Needs:**
  + Improving unique identifiers.
  + Introducing timestamps for trend tracking.
  + Standardizing claim types.
* **Strategic Recommendations:**
  + Implementing real-time anomaly detection for fraudulent claims.
  + Developing AI-driven dashboards for policy risk assessments.
* **Challenges and Considerations:**
  + Data completeness and bias in missing records.
  + The necessity for longitudinal data for better insights.

## **10. Methodology & Tools Used**

* **Data Processing & Analytics:** Python (pandas, NumPy, scikit-learn), SQL
* **Visualization:** Matplotlib, Seaborn, PowerBI
* **Machine Learning:** Clustering (K-Means, DBSCAN), Predictive Modeling (Logistic Regression, Decision Trees)
* **AI & LLM Integration:** Exploring retrieval-based insights using generative AI to enhance structured and unstructured data insights.

## **11. AI Capabilities & Agentic Flows**

* **Retrieval-Augmented Generation (RAG) for Claims Analysis:**
  + Implementing LLMs to extract structured and unstructured insights.
  + Enhancing claims decision-making through AI-assisted summarization.
* **Agentic Flows for Automated Analytics:**
  + Developing AI workflows to classify claims into risk categories.
  + Automating feature engineering using generative AI models.

## **12. Future Directions & Enterprise-Level Suggestions**

* **Enterprise-Level AI Strategy:**
  + Implementing predictive risk assessment in claims processing.
  + Integrating external health data sources for deeper insights.
  + Developing an AI-powered anomaly detection system for fraud detection.
* **Data Governance & Security Considerations:**
  + Ensuring privacy compliance in claims data.
  + Standardizing data pipelines for enhanced efficiency.