Absolutely! Here’s a **comprehensive, professional yet accessible documentation** of your **Feature Engineering Notebook** for your health insurance fraud detection project.  
 You can use this as a Markdown file (feature\_engineering.md), add to your project docs, or as a detailed summary cell in your notebook.

# **📊 Feature Engineering Notebook: Comprehensive Documentation**

## **Overview**

This notebook transforms raw healthcare claims and beneficiary data into **modeling-ready features** for provider-level health insurance fraud detection.  
 We systematically engineer, clean, and validate features, ensuring the resulting dataset is robust, interpretable, and suitable for both tree-based and linear machine learning models.

## **Steps and Rationale**

### **1. Data Import**

We start by loading all required datasets:

* **Claims data**: Inpatient and Outpatient claim files (multiple files merged if needed).
* **Beneficiary data**: Patient demographics and health conditions.
* **Provider labels**: Which providers are flagged as fraudulent (the target variable).

**Why?** Getting all data in one place is essential for feature creation and later merging.

### **2. Claim Aggregation by Provider**

We aggregate claims at the **provider level**, calculating metrics such as:

* **Total claims count** (inpatient/outpatient/overall)
* **Total and average reimbursed amounts**
* **Total and average deductibles**
* **Maximum claim value**
* **Claims per beneficiary**

**Why?** Fraudulent providers often file more or higher-value claims. Aggregating at the provider level is crucial since fraud detection is done per provider.

### **3. Patient Demographic Features**

We enrich providers with their **patient base characteristics**:

* **Average patient age**
* **Percent male/female**
* **Percent deceased**
* **Chronic condition rates** (e.g., diabetes, heart failure, Alzheimer’s)

**Why?** Some providers may target specific demographics or high-risk groups, which can signal abnormal behavior.

### **4. Claim Pattern Features & High-Risk Billing Indicators**

Additional features engineered include:

* **Unique beneficiaries per provider**
* **Claims per beneficiary**
* **Ratio of inpatient to outpatient claims**
* **Diagnostic/procedure diversity:**
  + *How many unique diagnosis/procedure codes are used?*
* **High-value claim rate:**
  + *Percent of claims above a certain dollar threshold*
* **Percent of claims on weekends**
* **Completeness of diagnostic/procedure codes**

**Why?** Unusual claim patterns, high diversity in codes, or frequent high-value claims are potential red flags for fraud.

### **5. Feature Table Merging**

All features are **combined into a single provider-level table**, merged with the fraud label for supervised modeling.

**Why?** This ensures each row corresponds to a provider and contains all relevant information.

### **6. Feature Cleaning**

We:

* **Drop non-informative columns:**
  + Provider ID (for modeling)
  + Columns with all missing or constant values (e.g., pct\_all\_proc\_filled)
* **Check for and document missing values and outliers.**

**Why?** Non-informative columns add no value; missing values or constants can break or dilute models.

### **7. Correlation Analysis**

We generate and inspect a **correlation matrix** of all engineered features, especially checking correlations with the fraud target.

#### **Key Findings:**

* **Highly correlated pairs** (correlation > 0.95):  
  + total\_claims ↔ outpatient\_claims
  + total\_reimb ↔ inpatient\_claims
  + total\_deductible ↔ inpatient\_claims and total\_reimb
* **Top features correlated with fraud:**
  + total\_reimb (0.58)
  + total\_deductible (0.53)
  + inpatient\_claims (0.53)
  + max\_reimb (0.51)
  + total\_claims (0.37)

**Why?** Highly correlated features can cause instability in some models (like Logistic Regression) but are less problematic for tree-based models (Random Forest, XGBoost).

### **8. Feature Set Preparation**

To support different modeling approaches:

* **Full Feature Set:**
  + Keeps all features (for tree-based models)
* **Reduced Feature Set:**
  + Drops one of each highly correlated pair (for linear models)
* **All-NaN and constant columns removed**

**Why?** Tree models can handle redundant features, but linear models require reduced multicollinearity for stable results and interpretability.

### **9. Dataset Export**

All feature sets and the main table are saved for easy access in future modeling steps.

## **Metrics & Dataset Shapes**

* **Providers in dataset:** 5,410
* **Features after cleaning:**
  + *Full set:* 28
  + *Reduced set (for linear models):* 25
* **No columns had all NaN values; one constant column dropped.**

## **Key Insights & Best Practices**

* **Most predictive features** are related to claim volume and total reimbursement/deductible amounts—consistent with industry fraud patterns.
* **Demographic features** (age, sex, chronic conditions) have weaker direct correlation to fraud but may provide value in interaction with other features.
* **High multicollinearity** detected and addressed via dual feature sets.
* **No major missing data issues.**
* **All transformations and cleanup steps are reproducible and clearly documented for audit or further analysis.**

## **What’s Next?**

* Move to the **modeling phase**:  
  + Split data into train/validation sets.
  + Train baseline models (Logistic Regression, Random Forest, XGBoost).
  + Track and compare performance with AUC, Precision, Recall, F1.
  + Use SHAP to explain model predictions.

## **TL;DR Summary**

This notebook engineered, cleaned, and validated provider-level features for health insurance fraud detection, preparing both full and reduced feature sets suitable for a variety of machine learning models. The resulting datasets are saved and ready for robust, auditable modeling in the next phase.