Fraud Detection Project

Data Cleaning and Aggregation

1. Introduction

The goal of the data cleaning and aggregation process was to transform raw Medicare Beneficiary, Inpatient, and Outpatient claims files into a structured, reliable dataset suitable for provider-level fraud detection modeling.
This involved correcting inconsistencies, extracting meaningful features, and aggregating claim-level patterns into provider-level indicators.

The steps below summarize the full cleaning and transformation workflow.

2. Beneficiary Data Cleaning

The beneficiary dataset contains demographic and health status information for Medicare patients. Cleaning steps included:

2.1 Date Standardization

- Converted **Date of Birth (DOB)** and **Date of Death (DOD)** to standardized datetime format.
- Ensured chronological consistency (e.g., DOD always after DOB).

2.2 Derived Features

- **Age** calculated from DOB.
- **IsDeceased** flag created (1 if DOD present, 0 otherwise).
- Preserved all **chronic condition indicators**, including:

    - Diabetes
    - Heart Failure
    - Chronic Kidney Disease
    - Depression
    - IHD and other comorbidities

2.3 Monetary Fields

- Kept annual reimbursement entries:
    - *IPAnnualReimbursementAmt*
    - *OPAnnualReimbursementAmt*

These beneficiary-level attributes were later merged with claim files to provide patient context for each provider.

3. Inpatient Claims Cleaning

The inpatient dataset contains hospitalization events. The following procedures were applied:

3.1 Date Validation & Processing

Converted date columns to datetime:

- **ClaimStartDt**, **ClaimEndDt**
- **AdmissionDt**, **DischargeDt**

Derived:

    - **ClaimDuration** = ClaimEndDt − ClaimStartDt

3.2 Physician Information

Created binary physician indicators:

- **HasAttendingPhysician**
- **HasOperatingPhysician**
- **HasOtherPhysician**

These features represent provider involvement and potential claim irregularities.

3.3 Diagnosis & Procedure Codes

- Counted **NumDiagnosisCodes** (across all diagnosis columns)
- Counted **NumProcedureCodes**
- Added flag **HasProcedure** for procedural involvement

3.4 Monetary Fields

Preserved without modification:

- *InscClaimAmtReimbursed*
- *DeductibleAmtPaid*

4. Outpatient Claims Cleaning

Outpatient records were processed using a consistent approach:

4.1 Date Handling

- Converted **ClaimStartDt** and **ClaimEndDt** to datetime
- Calculated **ClaimDuration**

4.2 Physician Flags

Same indicators as inpatient:

- **HasAttendingPhysician**
- **HasOperatingPhysician**
- **HasOtherPhysician**

4.3 Diagnosis Codes

- Counted **NumDiagnosisCodes**
- No length-of-stay metric (outpatient visits do not involve admissions)

4.4 Financial Values

Retained monetary fields unchanged for accuracy in later aggregation.

5. Dataset Merging

A multi-step merging process was applied to combine the cleaned datasets:

1. **Inpatient claims merged with Beneficiary data** on **BeneID**
2. **Outpatient claims merged with Beneficiary data** on **BeneID**
3. Combined all claims (IP + OP) into a unified dataset
4. Joined the unified dataset with **Provider Fraud Labels** on **Provider**

This produced a complete claim-level table linking patient demographics, claim financials, clinical information, and provider fraud status.

6. Provider-Level Aggregation

Fraud labels are assigned at the provider level; therefore, claim-level data was aggregated per provider.

The following metrics were computed for every provider:

6.1 Claim Activity Metrics

- Total claim count (IP + OP)
- Number of unique beneficiaries
- Claims per beneficiary ratio

6.2 Financial Metrics

- Total reimbursement amount
- Average reimbursement amount per claim

6.3 Clinical Behavior Metrics

- Average **LengthOfStay** (inpatient only)
- Average **NumDiagnosisCodes**
- Average **NumProcedureCodes**
- Average patient age

6.4 Patient Health Indicators

- Chronic condition prevalence rates (% of patients with each condition)

These aggregated metrics help capture patterns that differentiate fraudulent providers from normal ones.

7. Final Outputs

Two final modeling files were generated:

- **Train_final.csv** — provider-level features + fraud labels
- **Test_final.csv** — provider-level features without labels

Each row represents one provider and contains all aggregated financial, clinical, behavioral, and patient-based attributes.

8. Imbalance Handling

Fraud rate is ~9–10%.
Training without imbalance correction causes models to label everything as “non-fraud”.

Our Strategy:

- **sample_weight** for minority class
- **scale_pos_weight** for gradient boosting and XGBoost
- **class_weight='balanced'** for Random Forest & Logistic Regression

This ensures:

- Fraud cases are not ignored
- Models learn minority patterns without oversampling artifacts
- Gradient boosting models don't collapse to trivial predictions

9. Baseline Models

We trained the standard set of classifiers:

- Logistic Regression
- Random Forest
- Gradient Boosting (GB)
- XGBoost

Evaluated using:

- **ROC AUC**
- **Average Precision (AP)**
- **Precision/Recall**
- **Confusion matrix**

Baseline Findings

Overall, the initial results confirmed a pattern we expected from structured tabular data:

- **Gradient Boosting and XGBoost** performed almost identically, even though XGBoost is generally more expressive and can handle complex relationships better.

The reason is simple, our provider-level dataset is not large enough for XGBoost to show its usual advantage. Both models landed around high recall (~0.92) and moderate precision (~0.53) before tuning.

- **Logistic Regression** performed the weakest, especially in precision. It hovered around 0.40 precision, although its recall remained high (~0.90). This tells us the decision boundary is non-linear and LR cannot separate the classes cleanly.
- **Random Forest** was competitive, roughly on the same level as GB/XGB before tuning.RF shows similar recall but slightly more volatility in probability outputs.

From these baselines, Gradient Boosting and Random Forest came out as the most promising candidates, with GB having a natural edge due to smoother probability distributions and lower overfitting risk compared to its bigger brother, XGBoost.

10. Hyperparameter Tuning Strategy

Since this is a fraud-detection problem, **false negatives matter far more than false positives**. Missing a fraudulent provider is significantly worse than flagging an innocent one.
Because of that, **recall was the primary objective**, and the entire tuning process was designed around maximizing it without letting the model collapse into noise.

Threshold-based recall is what matters, so we used **Average Precision (AP)** as the tuning metric.
AP focuses on ranking quality instead of the 0.5 threshold, which we don't use anyway.

We kept the search space intentionally "safe", meaning shallow trees, low learning rates, and limited complexity. This avoids overfitting and keeps probability outputs smooth and predictable.

11. Calibration

We calibrated every model using **isotonic regression**. This helps correct the raw probability output and makes threshold selection stable. GB and XGBoost benefit from this the most, giving better recall control.

12. Threshold Optimization

After calibration, we didn’t use the default 0.5 threshold.

Instead, we scanned 2,000–5,000 thresholds and picked the one that:

- **maintains a high recall, and**
- **maximizes precision**

13. Final Model Comparison (Post-Tuning)

After tuning, calibration, and threshold optimization:

Gradient Boosting

- Recall ≈ **0.92**
- Precision ≈ **0.53**
- Very stable, low overfitting, smooth probabilities

XGBoost

- Recall ≈ **0.92**
- Precision ≈ **0.54**
- Slightly better but more prone to overfitting than GB due to data size

Random Forest

- Recall ≈ **0.90**
- Precision ≈ **0.58**
- Most reliable as it has the highest precision and still a recall above 0.9

Logistic Regression

- Recall ≈ **0.9**
- Precision ≈ **0.3**
- Struggles with non-linear boundaries

14. Final Model Choice

Random forest was selected as the final model.
It consistently delivered the best **recall–precision balance**, had the most stable calibration behavior, and showed the lowest overfitting tendency of all tree-based models.

15-Error Analysis

A critical component of evaluating our fraud detection model is understanding the types of mistakes it makes. In classification problems, two major error types matter most: **False Positives (FP)** and **False Negatives (FN)**. Each carries different business and operational implications for medical insurance fraud.

1. False Positives (FP)

Definition:
Cases where the model incorrectly flags a legitimate claim as fraudulent.

Implications:

- **Operational Burden:** Increases the workload for manual reviewers who must validate claims that are actually clean.
- **Provider/Customer Friction:** Legitimate healthcare providers and patients may experience delays in claim approval or reimbursement.
- **Cost Impact:** Although not a direct financial loss, excessive FPs raise internal investigation costs and can damage client relationships.
- **Indicator of Low Precision:** A high FP rate means the model is triggering too many "false alarms."

Data Observations:

We observed 3 false positives case studies where they all share these patterns:

- High claim volumes and many beneficiaries, but consistent with large practices
- High reimbursements, including some outliers, but medically plausible

- Chronic conditions and physician involvement are high but proportional to patient population
- The model may have interpreted large numbers or high reimbursements as suspicious

These false positives suggest the model may overreact to providers with large patient panels or high claim activity, even when the activity is legitimate. More context-aware features could help reduce such misclassifications.

2. False Negatives (FN)

Definition:
Cases where the model fails to detect a fraudulent claim.

Implications:

- **Direct Financial Loss:** Fraudulent claims are approved and paid, causing immediate monetary losses to the insurer.
- **Fraud Pattern Blind Spots:** Repeated undetected fraud strengthens criminals' confidence in exploiting the system.
- **Risk to Business Integrity:** High FN rates can reduce the overall effectiveness of the fraud prevention process.
- **Indicator of Low Recall:** When recall is low, the model is missing too many true fraud cases.

Data Observations:

We observed 3 false negatives case studies where they all share these patterns:

- Chronic disease levels are high but look medically consistent
- Claim Durations aren’t suspicious since Fraudulent inpatient claims often show repetitive/unusual long stays which isn’t shown in our FNs
- High physician involvement appears consistent with legitimate multi-doctor practices

The false negatives suggest the model may have difficulty identifying busy, high-chronic-burden providers, whose activity appears plausible, indicating that more behavior-based features could help improve detection.

3. Model Performance Balance

Our model currently demonstrates **high recall** but **lower precision**, meaning:

- It is **effective at catching most fraudulent claims** (low FN)
- But it **flags some legitimate claims incorrectly** (higher FP)

This trade-off is common in fraud detection systems. Because the cost of missing fraud (FN) is often higher than investigating a false alert (FP), many insurers prefer **recall-oriented models**.

4. Business Recommendation

- Maintain a **high-recall strategy** to minimize financial losses from undetected fraud.
- Mitigate false positives through:
  - Secondary rule-based verification
  - Human review for high-risk scores
  - Model calibration or threshold tuning
- Investigate false negatives thoroughly to understand fraud patterns the model is missing.