TECHNICAL REPORT

Healthcare Provider Fraud Detection Using Machine Learning

Team Members: Roba Ahmed · Yahia Walid · Abdelrahman Shady · Ali Yasser

1. Introduction

Healthcare fraud is a significant financial burden on medical systems, costing an estimated **$68+ billion annually**. Traditional fraud detection approaches rely on manual audits and fixed rules, which struggle to detect evolving or subtle fraudulent behavior.

The objective of this project is to build a **machine learning system** that identifies potentially fraudulent healthcare providers using large-scale Medicare claim datasets. The focus is on:

- Creating a clean and accurate provider-level dataset
- Engineering meaningful features
- Handling severe class imbalance
- Training and comparing multiple classifiers
- Prioritizing **Recall**, to ensure that most fraudulent providers are captured
- Conducting detailed evaluation and error analysis to understand model behavior

This report documents the full pipeline: from data exploration and preprocessing, to modeling, evaluation, and insights.

2. Dataset Understanding

We worked with four Medicare datasets at different granularities:

1. **Train_Beneficiarydata.csv** – Patient-level demographics and chronic conditions
2. **Train_Inpatientdata.csv** – Claim-level inpatient hospitalization records
3. **Train_Outpatientdata.csv** – Claim-level outpatient visit records
4. **Train_Labels.csv** – Provider-level fraud labels ("Yes"/"No")

Key Join Keys

- `BeneID`: Links beneficiary → inpatient & outpatient claims
- `Provider`: Links all claims → fraud labels

Merging Structure

A two-step hierarchical merge was necessary:

1. Beneficiary → Claims (via BeneID)
2. Claims → Provider-level aggregates (via Provider)

This mapping ensured we could engineer provider-level features for modeling.

3. Data Cleaning

A thorough quality assessment identified several issues:

3.1 Missing Values

- Chronic condition indicators contained inconsistent strings ("Yes/No") and missing entries.
- Financial claim fields occasionally contained blanks or zeros.
- Date fields had multiple formats and invalid entries.

3.2 Cleaning Actions

- Converted chronic condition fields to binary (1 = condition present).
- Replaced blank numeric entries with 0 only where medically appropriate.
- Standardized all date fields with `pd.to_datetime`.
- Applied **IQR-based filtering** to remove extreme outliers in claim amounts.

These steps ensured consistency during merging and aggregation.

4. Exploratory Data Analysis (EDA)

4.1 Beneficiary-Level Findings

- Fraudulent providers treat **older patients** on average.
- Beneficiaries served by fraudulent providers show higher rates of:
  - Diabetes
  - Chronic Kidney Disease
  - Other chronic conditions

4.2 Claim-Level Findings

From inpatient & outpatient datasets:

- Fraudulent providers submit **more claims per beneficiary**.
- They have **higher average reimbursement amounts**.
- Outpatient visits show **greater variability** in procedure counts.

4.3 Fraud Distribution

- Only **~9%** of providers are fraudulent → **severe class imbalance.**
- This imbalance heavily influenced later modeling decisions.

4.4 Correlation Heatmap

- Strong correlation between total claim counts and total reimbursement.
- Moderate relationships between chronic conditions and claim frequency.
- No multicollinearity severe enough to hinder model performance.

5. Feature Engineering

To enable provider-level prediction, we engineered aggregated statistical features. These included:

Counts

- `total_claims_per_provider`
- `num_unique_beneficiaries`

Sums

- Total reimbursement amounts
- Total deductible amounts

Means

- Average inpatient claim amount
- Average outpatient deductible
- Average length of stay

Ratios

- `inpatient_to_outpatient_ratio`
- Percentage of high-cost claims
- Chronic condition prevalence percentages

Examples of final features

- `avg_inpatient_claim_amount`
- `avg_chronic_condition_count`
- `percent_diabetic_patients`
- `total_outpatient_deductible`
- `avg_length_of_stay`

These transformations converted thousands of raw claim rows into meaningful provider-level indicators.

6. Data Merging Pipeline

Final merging steps:

1. Merge Beneficiary → Inpatient claims (via BeneID)
2. Merge Beneficiary → Outpatient claims (via BeneID)
3. Concatenate inpatient & outpatient claims
4. Group by Provider and compute aggregated features
5. Merge with fraud labels

Final Output

A complete modeling dataset: **provider_level.csv**

Containing:

- Engineered features
- Provider-level aggregates
- Fraud label (0/1)

7. Modeling Approach

Objective:

Build a classifier that predicts whether a provider is fraudulent.

Data Preparation

- Target variable: `PotentialFraud` (0/1)
- Provider ID removed to avoid leakage
- Stratified split: 60% train, 20% validation, 20% test
- Standardization applied only for Logistic Regression

Handling Class Imbalance

Two strategies:

1. **Class Weights**
   - Misclassifying fraud is penalized heavier
2. **SMOTE Oversampling**
   - Generates synthetic fraud samples
   - Applied ONLY to the training set

These techniques ensure the model learns fraud signals despite the imbalance.

8. Models Implemented

Model Families

1. **Logistic Regression**
    - Baseline interpretability
2. **Random Forest**
    - Non-linear, high performance
3. **HistGradientBoostingClassifier**
    - Handles complex relationships

Each was trained under two configurations:

- Class weights only
- SMOTE + default or tuned hyperparameters

9. Experiments and Trials

All candidate models were evaluated on the validation set using the following metrics:

- **Precision** (reduce false investigations)
- **Recall** (catch more fraud cases — the priority)
- **F1-score** (balanced performance)
- **ROC-AUC** (ranking capability)
- **PR-AUC** (critical metric under severe imbalance)

From this comprehensive comparison, the **top two models** consistently performing best across the most important fraud-related metrics were:

1. **Random Forest (Class Weights)**
   – strong Precision and PR-AUC (not selected for tuning)
2. **Random Forest (SMOTE)**
   – Best **Recall**, strong sensitivity to fraud
   – Higher F1 than other SMOTE models (selected for tuning)

These two models were compared, but tuning was applied **only to the SMOTE-based Random Forest**, because it achieved the highest Recall — the project's top priority.

10. Hyper parameter Tuning of Top Models

After validation experiments, the **Random Forest model trained using SMOTE** showed the strongest Recall and the highest sensitivity to fraudulent behavior. Because Recall is the top

priority for fraud detection, **this SMOTE-based model was selected for hyperparameter tuning**.

A lightweight GridSearchCV was performed with a small search space:

Best Parameters Found

- `n_estimators = 200`
- `max_depth = 10`
- `min_samples_split = 5`

Why the SMOTE model was tuned

- It achieved the highest Recall before tuning
- It showed the strongest ability to detect fraud (the primary goal)
- SMOTE improved minority class representation, enabling the model to learn subtle fraud patterns
- Tuning further improved F1 and PR-AUC

Impact of Tuning

The tuned SMOTE-based Random Forest improved:

- Fraud Recall
- F1-score
- Overall balanced performance
- PR-AUC under class imbalance

This tuned SMOTE model became the final selected model.

11. Final Best Model Selection

After hyperparameter tuning, the **Random Forest model trained with SMOTE (RF_SMOTE_TUNED)** demonstrated the strongest and most balanced performance across the fraud-specific metrics.

Why RF_SMOTE_TUNED was selected

✔ Highest Recall (0.67) — **critical for detecting fraud**
✔ Strong F1-score (0.58)
✔ Competitive Precision (0.50)
✔ Strong ROC-AUC (0.922)
✔ Robust PR-AUC under imbalance (0.627)
✔ Best overall tradeoff between catching fraud and reducing false positives

Final Saved Model

`best_model.pkl`
This file contains the tuned Random Forest model trained on SMOTE-resampled data.

12. Evaluation Results

After full retraining, RF_tuned achieved the following **test set metrics**:

Precision (Fraud)	**0.50**
Recall (Fraud)	**0.67**
F1-score	**0.58**
ROC-AUC	**0.922**
PR-AUC	**0.627**

Confusion Matrix

- **True Positives (TP): 68**
- **False Positives (FP): 67**
- **False Negatives (FN): 33**
- **True Negatives (TN): 914**

Interpretation

- The model captures **67% of all fraudulent providers**, aligning with project goals.
- False positives are expected in fraud detection but manageable.
- Fraudulent providers missed (FN) typically have very subtle or low-volume activity.

ROC Curve

- ROC-AUC = **0.922**
- Excellent ranking performance.

PR Curve

- PR-AUC = **0.627**
- Strong for a rare-event problem.

Sample FP and FN Cases

- **FP Examples:** Providers with unusually high claim amounts or unusual billing patterns
- **FN Examples:** Low-volume providers with minimal claims making fraud harder to detect

The tuned model reduced false positives while maintaining high recall—ideal for CMS fraud detection.

ROC Curve

- ROC-AUC = **0.94**
- Indicates excellent separation between fraud and non-fraud providers.

PR Curve

- PR-AUC = **0.738**
- Indicates strong performance for rare-event detection.

13. Error Analysis

False Positives (FP)

Providers incorrectly flagged as fraud typically showed:

- Higher-than-normal average claim amounts
- Aggressive billing patterns
- High inpatient utilization
- Elevated chronic condition proportions

Interpretation:
The tuned SMOTE-Random Forest model is sensitive to unusual billing patterns and high-cost claims — appropriate for prioritizing investigation.

False Negatives (FN)

Fraud cases the model missed had:

- Very low billing volume
- Subtle fraud patterns resembling legitimate providers
- Missing temporal or specialty-specific patterns

Interpretation:
These edge-case frauds require more advanced modeling (e.g., sequence models, time-based features).

Insights & Recommendations

- Introduce specialty-based peer comparisons
- Add temporal features to detect sudden billing spikes
- Use anomaly detection for low-activity providers
- Consider incorporating SHAP for interpretability

14. Conclusion

The final system presents a robust and explainable fraud detection pipeline aligned with CMS objectives and the course project requirements.

✔ Strong Predictive Power

- High recall: captures most fraud cases
- Strong PR-AUC & ROC-AUC
- Reduced false positives after tuning

✔ Business Impact

- Prioritizes high-risk providers
- Supports smarter audit allocation
- Potentially saves millions by reducing undetected fraud

✔ Future Enhancements

- Time-series fraud modeling
- SHAP interpretability
- Deployment as an API
- Integration with provider profiling tools

The tuned SMOTE-based Random Forest model (RF_SMOTE_TUNED) was selected as the final best model due to its excellent precision-recall balance and overall reliability.