Medicare Fraud Detection Course Project Report

Prepared By

Team 4:

Peyton Nash, Bradley Stoller,
Kyler Rosen, Mustapha Mbengue

March 9, 2025

TABLE OF CONTENTS

PROBLEM DEFINITION & OBJECTIVE

Medicare fraud, including false claims, overbilling and service misrepresentation, led to $2 billion in taxpayer losses in 2024 and increased policyholder premiums by $25–$50. These fraudulent activities strain healthcare resources, inflate costs, and undermine the integrity of the system.

While preventing fraudulent reimbursements and recouping funds that have already been paid out saves money, there is also a cost to investigating potential fraud. Given the magnitude of the Medicare program, it is untenable to investigate every reimbursement request.

This presents an opportunity to leverage machine learning to create a more manageable shortlist of potentially fraudulent requests that can be investigated by the Department of Health and Human Services. The success of the model should be measured by its ability to balance the benefits of correctly identifying true cases of fraud and the costs of incorrect identifications (recall and precision). Increasing recall lowers the rate of undetected fraud, while increasing precision lowers investigation costs spent on non-fraudulent claims.

Using high dimensional insurance data, we developed supervised, offline anomaly detection and classification models, that enable the Department of Health and Human Services to engage in targeted investigations of potential fraud.

DATA SOURCING

Sourced from Kaggle, our dataset consists of eight CSV files that contain Medicare information related to inpatient claims, outpatient claims, and beneficiary details by provider. There is a labeled and unlabeled file for each file type (inpatient, outpatient, and beneficiary), as well as an additional training file denoting whether each provider was fraudulent.

We initially intended to use both the labeled and unlabeled files as our train and test datasets. However, because we did not have access to the labels corresponding to the test files and the training files were sufficiently large (130k+ rows each), we decided to perform our own train-test split on the labeled data only.

DATA EXPLORATION

Before merging and preprocessing the data, we conducted an exploratory data analysis (EDA) to inform our data preparation approach. Specifically, we examined each of the three datasets (inpatient, outpatient, beneficiary) as well as the merged final product with appended fraud labels.

The following are our main findings:

INPATIENT FINDINGS

- **Missing Data:** When examining the data, we found that many of the columns were missing significant proportions of data. However, we realized that this was due to the dataset's design, and not a reflection of missing information. For example, columns like *ClmProcedure_7* had to be added to accommodate the patients who needed seven procedures, but patients who did not need a seventh procedure would have a missing value in that column and can be handled within the feature engineering step.

- **Skewed Distributions:** We also found that some of the quantitative columns, such as *InsClaimAmtReimbursed*, were heavily skewed. However, this is expected because high reimbursements were often correlated with potential fraud.

OUTPATIENT FINDINGS

- **Missing Data:** Like the inpatient data, many columns contained missing values, but these were expected since not all patients needed every claim/procedure.

- **Skewed Distributions:** As with the inpatient data, some of the quantitative columns were skewed, but also correlated to fraud.

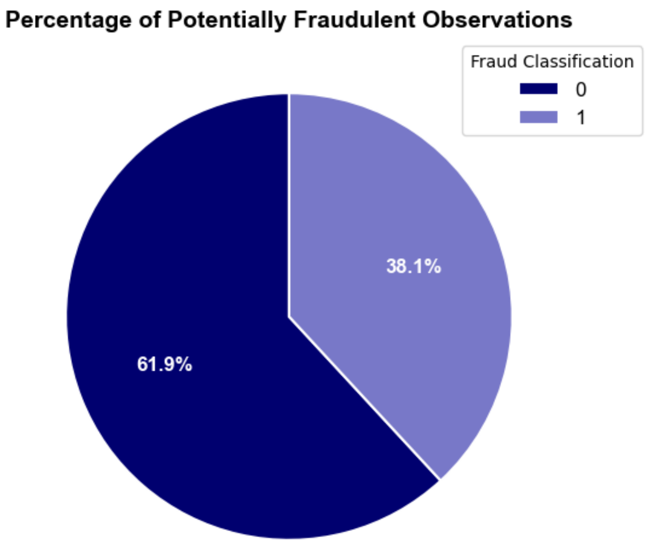
BENEFICIARY FINDINGS

- **Missing Data:** Unlike the inpatient and outpatient datasets, the beneficiary data did not contain many missing values. The exception was the *DOD* column which was entirely missing. While we initially assumed that this was because most patients were alive at the time the data was collected, since the entire column was missing, we realized it would need to be dropped in feature engineering.

MERGED FINDINGS

- **Claim and Diagnosis Code Encoding:** When cross-referencing the dataset against the Medicare website, we noticed that many of the diagnosis and claim codes required encoding to extract their meaning. For example, the data's original patient code for xeroderma of the eyelid was 37333 instead of the correct Medicare code of 373.33.

- **Target Distribution Analysis:** To ensure that we were able to use machine learning to classify the instances of fraud, we examined the distribution of the *PotentialFraud* column and found that 62% of claims were fraudulent. While this means that the dataset does not have a class imbalance, the high fraud rate does not match the real-world fraud rate, raising concerns about any model's generalizability.



DATA PREPARATION

We performed three main steps to prepare the data for modeling: merging the datasets, adjusting the claim/procedure codes and applying feature engineering.

Data Merging

We began by loading the labeled inpatient, outpatient, beneficiary and fraud CSV files, merging them on the shared *Provider* column and saved the data to be further processed for modeling.

Claim & Procedure Encoding

Each claim and procedure was originally labeled numerically in XX.YYY format, where the XX represented the category and YYY the subcategory. Because these numeric labels did not represent continuous features, we transformed them into hierarchical categorical variable by creating two columns, one for the category alone and another combining the category and subcategory.

Feature Engineering

As a last step before training machine learning models, we applied the following feature engineering steps to get the most out of the data:

1. Extracting Datetime Features:

- Computed HospitalStayDays as (DischargeDt - AdmissionDt) and ClaimDuration as (ClaimEndDt - ClaimStartDt), filling missing values appropriately
- Derived DaysBeforeAdmission (AdmissionDt - ClaimStartDt) to capture the time gap between claim initiation and hospital admission
- Extracted *ClaimStartMonth*, *ClaimStartWeekday* and *ClaimStartYear* from *ClaimStartDt* to capture temporal trends
- Calculated *DaysSinceLastClaim* per beneficiary to track claim frequency
- Derived *AgeAtClaim* by subtracting DOB from *ClaimStartYear*

2. Discretizing Age:
    - Binned the *Age* feature into five meaningful categories: Under 50, 50-60, 61-79, 80-97, and 98+
3. Handling Missing Values:
    - Imputed *DeductibleAmtPaid* with 0 for missing values
    - Filled *AttendingPhysician* with the most frequent physician per provider
    - Replaced missing *OtherPhysician* and *OperatingPhysician* values with "None"
    - Set missing *ClmAdmitDiagnosisCode* values to "Not Applicable"
    - Imputed missing *ClmDiagnosisCode* and *ClmProcedureCode* values with "No Diagnosis" and "No Procedure", respectively
4. Transforming Skewed Distributions:
    - Applied log transformation to highly skewed numerical variables:
        - *InscClaimAmtReimbursed*
        - *DeductibleAmtPaid*
        - *IPAnnualDeductibleAmt*
        - *OPAnnualDeductibleAmt*
5. Encoding Categorical Variables:
    - Used label encoding to convert categorical fields into numeric representations for model compatibility
6. Dropping Unnecessary Columns:
    - Removed redundant, excessive, or irrelevant fields, including:

- Identifiers & PII: *BeneID, ClaimID*
- Date Fields (after extracting features): *AdmissionDt, DischargeDt, ClaimStartDt, ClaimEndDt, DOB*
- Redundant Financial Fields: *InscClaimAmtReimbursed, DeductibleAmtPaid, IPAnnualDeductibleAmt, OPAnnualDeductibleAmt*
- Flags for Unknown Values: *Flag_Unknown_Procedures, Flag_Unknown_Diagnoses*
- Descriptive Columns: Any columns containing "Desc" in their name

SHORTLISTED MODELS

To build an effective fraud detection system, we explored various machine learning models using high-dimensional insurance data. We initially tested logistic regression, decision trees, and Naïve Bayes models. Logistic regression, though fast and interpretable, struggled to capture complex fraud patterns due to its linear decision boundary, which led to moderate accuracy and recall. Decision trees improved upon logistic regression by identifying non-linear relationships in the data but were prone to overfitting. Naïve Bayes models, both Gaussian and Bernoulli variants, proved computationally efficient but struggled with recall, making them less effective at identifying fraudulent claims accurately.

Because of the shortcomings of these models, we turned to more advanced and computationally intensive techniques, particularly Random Forest and XGBoost. These ensemble-based methods demonstrated superior accuracy in handling large, complex datasets. Random Forest, by aggregating multiple decision trees, improved prediction reliability and reduced overfitting, while XGBoost further enhanced precision by refining weak models into a highly optimized fraud classification system.

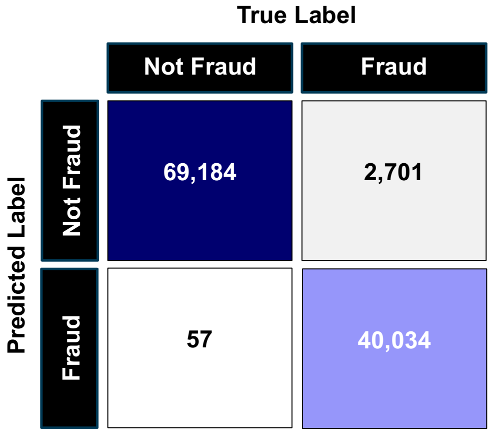
MODELING RESULTS & ANALYSIS

After our initial experimentation, we fine-tuned both Random Forest and XGBoost to enhance their fraud detection performance. For Random Forest, we focused on optimizing the number of estimators, maximum depth, and minimum samples split, ensuring that the model balanced complexity and accuracy. With these refinements, the model achieved an accuracy of 93.17%, a precision of 99.02% and a recall of 82.92% on the test data. While the model was excellent at correctly identifying fraudulent claims, it also minimized the number of false positives, making it a highly reliable tool for Medicare fraud detection.

XGBoost underwent a similar optimization process, with fine-tuning adjustments to the number of estimators, learning rate, and maximum depth. The optimized model delivered exceptional results, with an accuracy of 97.54%, a precision of 99.86%, a recall of 93.68%, an F1 score of 96.67% and a near-perfect ROC AUC of 99.82% on the test set.

We also trained several ensemble models using the predictions of the base models as inputs. First, we used a soft voting model that weights the vote based on the confidence of each model's predictions. This model achieved an accuracy score of 94.76% a precision of 99.73% and a recall of 86.51%. While this model is great at predicting true positives, it suggests far more non-fraudulent cases for investigation than XGBoost.

Figure 2: XGBoost Confusion Matrix



Predicted Label	True Label: Not Fraud	True Label: Fraud
Not Fraud	69,184	2,701
Fraud	57	40,034

Next, we trained multiple stacked models using the logistic regression, Gaussian Naïve Bayes, Random Forest and XGBoost predictions as model features and a logistic regression and SVM classifier as the stacking models. The SVM classifier was tuned on kernel, degree and C value. Both models produce results comparable to XGBoost. The logistic regression model achieves an accuracy of 97.64%, a precision of 99.82%, a recall of 93.98% and an F1 score of 96.81%. Similarly, the SVM classifier achieved an accuracy of 97.65%, a precision of 99.79%, a recall of 94.03% and an F1 score of 96.82%.

Through experimentation, we found that ensemble models, particularly XGBoost, and the stacked models provide the best balance between precision and recall. All three models not only catch fraudulent claims effectively but also minimize unnecessary investigations, allowing Medicare and the Department of Health and Human Services to direct its resources where they are most needed. While the stacked models edge out XGBoost in some metrics, we lack the data and domain knowledge to determine which metrics to prioritize. There is a financial trade-off between true positives and false negatives, but we do not have data on the specifics of this trade-off. The models perform similarly on all metrics, but because XGBoost identifies a greater proportion of the fraudulent cases and is less computationally intensive, we believe XGBoost to be the best model. This strategic use of machine learning ensures that fraud detection efforts remain both efficient and impactful.

Commented [PN1]: Added this - does this make sense?

Table 1: Classification Metrics for Contender Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	81.07%	78.35%	69.63%	73.73%	86.30%
Random Forest	93.17%	99.02%	82.92%	90.25%	98.42%
Gaussian Naïve Bayes	80.11%	81.98%	61.39%	70.21%	80.43%
Bernoulli Naïve Bayes	62.83%	56.48%	11.30%	18.83%	53.17%
Voting Model	94.76%	99.73%	86.51%	92.65%	-
Logistic Regression Stacked	97.64%	99.82%	93.58%	96.81%	97.31%
SVM Stacked	97.65%	99.79%	94.03%	96.82%	-
XGBoost	**97.54%**	**99.86%**	**93.68%**	**96.67%**	**99.82%**

LIMITATIONS AND EXTENSIONS FOR FUTURE WORK

While our champion model, XGBoost, performs very well at the task of identifying fraudulent cases, there is still room for improvement in rooting out Medicare fraud. First, the data is surprisingly balanced given the nature of the problem. Nearly 40% of cases are fraudulent, which does not reflect reality and raises questions about the sampling methods:

- How and from where was the data collected?
- What is the exact definition of the ‘PotentialFraud’ column?
  - Are these a subset of the investigated cases?
  - Have the fraud cases been adjudicated?

Knowing this will give us a better grasp of the issue of partial observability whereby only cases investigated and determined to be fraudulent are marked as fraud. This limits the model’s ability to generalize and accurately detect previously unseen fraud patterns.

Regardless of the class-balance, with additional measures we can better model the financial trade-offs between detecting and investigating fraud. Knowing the cost of each investigation can provide insight into which performance metrics to prioritize in evaluating models and allow the model to balance the benefit of identifying fraud with the costs of investigating it.