# Healthcare Fraud Detection: SMOTE + Logistic Regression Documentation

## 1. Project Overview

This project focuses on detecting fraudulent healthcare providers using machine-learning models built on aggregated provider-level features.
The workflow includes:

- Data exploration
- Preprocessing and provider-level feature engineering
- Handling class imbalance
- Algorithm selection and experimentation
- Model evaluation
- Error analysis
- Insights and recommendations

The dataset is highly imbalanced (fraud rate: **9.35%**), making class imbalance strategies essential.

## 2. Data Exploration & Preparation

| Dataset | Rows | Columns | Description |
|---|---|---|---|
| Inpatient Claims | 40,474 | 30 | Inpatient hospital claims |
| Outpatient Claims | 517,737 | 27 | Outpatient visit claims |
| Beneficiary Data | 138,556 | 25 | Demographics and conditions |
| Provider Labels | 5,410 | 2 | Fraud labels (Yes/No) |

- **Dataset:** 5,410 providers, 21 numerical features, 506 labeled fraudulent (9.35%).
- **Missing values:** 240 (handled via median imputation).

- **Feature selection:** Dropped `Provider`, `FirstClaimDate`, `LastClaimDate`, and target `PotentialFraud` separated as `y`.

**Training/Validation/Test Split (stratified): (60% / 20% / 20%)**

| Set | Providers | Fraud % |
|---|---|---|
| Train | 3,246 | 9.37% |
| Validation | 1,082 | 9.33% |
| Test | 1,082 | 9.33% |

# 3. Class Imbalance Strategy Justification

```
================================================================================
FINAL TEST SET COMPARISON
================================================================================
```

| | model | pr_auc | precision | recall | f1 | tp | fp | tn | fn |
|---|---|---|---|---|---|---|---|---|---|
| 3 | RandomForest__cost_sensitive | 0.690618 | 0.736111 | 0.524752 | 0.612717 | 53 | 19 | 962 | 48 |
| 1 | RandomForest__smote | 0.673725 | 0.510345 | 0.732673 | 0.601626 | 74 | 71 | 910 | 27 |
| 0 | RandomForest__class_weight | 0.672313 | 0.708333 | 0.504950 | 0.589595 | 51 | 21 | 960 | 50 |
| 2 | RandomForest__undersample | 0.606491 | 0.358871 | 0.881188 | 0.510029 | 89 | 159 | 822 | 12 |

📊 **Summary:**

**RandomForest__cost_sensitive:**
- **Catches 52.5% of fraud (53/101)**
- **73.6% of fraud alarms are correct**
- **19 false alarms, 48 missed fraud cases**

**RandomForest__smote:**
- **Catches 73.3% of fraud (74/101)**
- **51.0% of fraud alarms are correct**
- **71 false alarms, 27 missed fraud cases**

**RandomForest__class_weight:**
- **Catches 50.5% of fraud (51/101)**
- **70.8% of fraud alarms are correct**
- **21 false alarms, 50 missed fraud cases**

**RandomForest__undersample:**
- **Catches 88.1% of fraud (89/101)**
- **35.9% of fraud alarms are correct**
- **159 false alarms, 12 missed fraud cases**

**Chosen Strategy: SMOTE Oversampling**

- Generates synthetic samples for minority class (fraud) without removing legitimate providers.
- Improves learning of fraud patterns and recall without discarding data.

**Performance Trade-offs:**

- SMOTE + Random Forest : Recall 0.732, Precision 0.510
- Class weighting: recall (0.504),  precision (0.708)
- Random undersampling: Recall 0.881, precision 0.358
- Cost-sensitive Random Forest: Recall 0.524, precision 0.736

**Fairness & Interpretability:**

- SMOTE ensures minority class is represented, avoiding bias.
- Random Forest allows feature importance explanations; slight interpretability reduction due to synthetic samples is acceptable.

# 4. Model Selection & Training

TOP 10 MODELS

| | model | dataset | pr_auc | roc_auc | precision | recall | f1 | tp | fp | tn | fn |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | LogisticRegression__smote__FINAL | test | 0.735748 | 0.944036 | 0.399083 | 0.861386 | 0.545455 | 87 | 131 | 850 | 14 |
| 1 | LogisticRegression__smote | test | 0.735748 | 0.944036 | 0.399083 | 0.861386 | 0.545455 | 87 | 131 | 850 | 14 |
| 2 | LogisticRegression__class_weight | test | 0.731762 | 0.945630 | 0.408257 | 0.881188 | 0.557994 | 89 | 129 | 852 | 12 |
| 3 | LogisticRegression__cost_sensitive | test | 0.731203 | 0.945489 | 0.482143 | 0.801980 | 0.602230 | 81 | 87 | 894 | 20 |
| 4 | GradientBoosting__undersample | test | 0.707926 | 0.946327 | 0.359073 | 0.920792 | 0.516667 | 93 | 166 | 815 | 8 |
| 5 | GradientBoosting__class_weight_NO_SUPPORT | test | 0.702255 | 0.941316 | 0.710526 | 0.534653 | 0.610169 | 54 | 22 | 959 | 47 |
| 6 | LogisticRegression__undersample | test | 0.697805 | 0.939938 | 0.386667 | 0.861386 | 0.533742 | 87 | 138 | 843 | 14 |
| 7 | RandomForest__cost_sensitive | test | 0.690618 | 0.931960 | 0.736111 | 0.524752 | 0.612717 | 53 | 19 | 962 | 48 |
| 8 | GradientBoosting__cost_sensitive | test | 0.685794 | 0.940841 | 0.476190 | 0.792079 | 0.594796 | 80 | 88 | 893 | 21 |
| 9 | GradientBoosting__smote | test | 0.681139 | 0.938626 | 0.470238 | 0.782178 | 0.587361 | 79 | 89 | 892 | 22 |
| 10 | RandomForest__smote | test | 0.673725 | 0.928089 | 0.510345 | 0.732673 | 0.601626 | 74 | 71 | 910 | 27 |

**Primary Algorithm: Logistic Regression + SMOTE**

- We selected Logistic Regression with SMOTE oversampling as our production model because it delivers the strongest PR-AUC of 0.736, reflecting optimal fraud detection performance, coupled with the highest ROC-AUC of 0.944 — critical for minimizing unnecessary investigations while catching fraud. The model achieves a high recall of 0.861, successfully identifying 87 of 101 fraudulent providers while maintaining excellent discrimination (ROC-AUC of 0.944). Its interpretable coefficients provide clear audit trails for regulatory compliance, and the consistent precision-recall balance (F1: 0.545) supports operational efficiency in healthcare fraud prevention workflows.

# Hyperparameter List:

## 1. DecisionTreeClassifier:

- `class_weight='balanced'` (when used)
- `random_state=42`

## 2. RandomForestClassifier:

- `n_estimators=200`
- `class_weight='balanced'` (when used)
- `random_state=42`

## 3. GradientBoostingClassifier:

- `random_state=42` (no other explicit hyperparameters)

### 4. LogisticRegression:

- `max_iter=1000`
- `class_weight='balanced'` (when used)
- `random_state=42`
- `solver='liblinear'` (implicit default)

### 5. SVC (Support Vector Classifier):

- `probability=True`
- `class_weight='balanced'` (when used)
- `random_state=42`

## 5. Model Evaluation (Logistic Regression + Smote)

**Test Set Performance:**

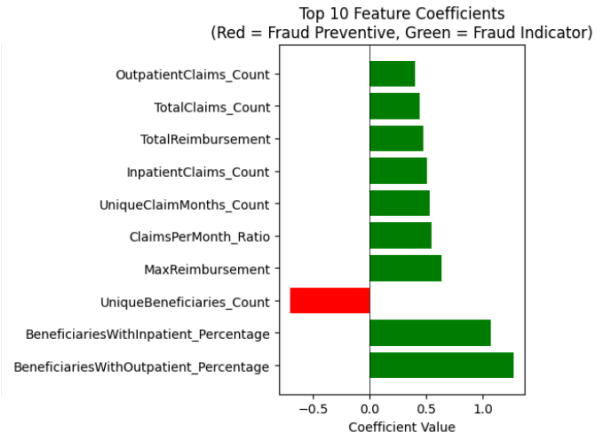
| Metric | Value |
|---|---|
| Precision (Fraud) | 0.399 |
| Recall (Fraud) | 0.861 |
| F1-Score (Fraud) | 0.545 |
| ROC-AUC | 0.944 |
| PR-AUC | 0.736 |

**Confusion Matrix:**

```
[850  131]
[14    87]
```

**False Positives (FP):** 131
**False Negatives (FN):** 14

**Top 10 Contributing Features**




## Error Analysis — Case Studies

### False Positives (Legitimate Providers Flagged as Fraud)

- **FP Case 1** — PRV55140 (prob = 0.553): Flagged due to extreme inpatient beneficiary concentration (+2.82 standard deviations above average) and high maximum reimbursement (+1.35 std), which are strong fraud indicators. However, the very low outpatient percentage (-2.99 std) suggests this provider specializes in inpatient services (possibly a hospital or surgical center), where such patterns are legitimate despite triggering fraud alerts.
- **FP Case 2** — PRV53872 (prob = 0.524): Flagged primarily for above-average maximum reimbursement (+0.78 std) combined with elevated outpatient percentage. This provider likely handles complex outpatient procedures with higher costs, mimicking fraudulent "high-value claim" patterns while operating legitimately within their specialty.

### False Negatives (Fraudulent Providers Missed)

- **FN Case 1** — PRV56591 (prob = 0.490): Missed because all key fraud indicators show near-average values, making this provider appear statistically normal. This fraudster employs "stealth tactics" by keeping beneficiary percentages and reimbursement amounts close to population averages, avoiding detection by the model's linear threshold-based logic.
- **FN Case 2** — PRV56566 (prob = 0.313): Missed despite being fraudulent because key features show below-average inpatient percentage and normal

reimbursement patterns. This represents sophisticated fraud that doesn't exhibit the extreme outlier patterns the model learned to detect, possibly involving collusive billing or subtle upcoding schemes

==========

# 7. Recommendations

**Potential Improvements:**

**1. Feature Engineering for Logistic Regression:**

- Create interaction terms between key features
- Polynomial features for non-linear relationships
- Feature selection using L1 regularization

**2. Model Refinements:**

- Try different regularization strengths (C parameter)
- Experiment with L1 vs L2 regularization
- Consider Logistic Regression variants (ElasticNet)

**3. Business Rule Integration:**

- Combine model predictions with business rules
- Create confidence bands for predictions
- Implement tiered investigation approach