Machine Learning & Dataset Explanation

Healthcare Fraud Detection Project

---

1. Dataset Overview

Source: Kaggle Medicare Provider Fraud Detection Dataset

Property	Value
Total Claims	558,211
Unique Providers	5,410
Unique Patients	138,556
Fraud Rate	~9.6% (at provider level)
File Size	~40 MB (claims.csv)

Original Kaggle Files

```
Dataset/
├── Train_Beneficiarydata.csv   (Patient demographics)
├── Train_Inpatientdata.csv     (Hospital admissions)
├── Train_Outpatientdata.csv    (Outpatient visits)
└── Train.csv                   (Provider fraud labels)
```

Processed Data (claims.csv columns)

Column	Type	Description
`claim_id`	String	Unique claim identifier
`provider_id`	String	Healthcare provider ID
`patient_id`	String	Patient beneficiary ID
`claim_type`	String	"Inpatient" or "Outpatient"
`amount`	Float	Claim reimbursement amount ($)
`deductible`	Float	Patient deductible amount
`num_diagnoses`	Int	Number of diagnosis codes
`num_procedures`	Int	Number of procedures performed
`length_of_stay`	Int	Hospital stay duration (days)
`diagnosis_code`	String	Primary ICD-9/10 code
`patient_age`	Int	Patient age in years

`patient_gender`	Int	1=Male, 2=Female
`chronic_conditions`	Int	Count of chronic conditions (0-11)
`is_fraud`	Int	0=Legitimate, 1=Fraudulent

---

2. Synthetic Dataset

What is "Synthetic" in this project?

The project uses **two types of synthetic data**:

A. ICD Code Database (Reference Data)

```
Dataset/Synthetic Dataset/
├── ICD9codes.csv    (1.6 MB - Legacy codes)
├── ICD10codes.csv   (14.6 MB - Modern codes)
└── icd9dx2015.csv   (1.4 MB - 2015 codes)
```

Purpose: Provides disease names and descriptions for:

- Converting code "V700" → "General Medical Examination"
- Benchmarking expected costs for each disease

B. Disease Price Benchmarks

```
data/disease_prices.csv (6,016 diagnosis codes)
```

Column	Example
`diagnosis_code`	V700
`disease_name`	General Medical Examination
`standard_price`	500

How prices were generated:

1. Scraped average costs from medical databases
2. Categorized by disease severity
3. Applied regional multipliers

Why it's "Synthetic":

- Real Medicare doesn't publish per-diagnosis prices
- Prices are estimates based on industry research
- Used only for anomaly detection, not billing

---

3. Exploratory Data Analysis (EDA)

Key Findings from EDA

3.1 Fraud Distribution

- **Legitimate Claims**: ~90.4%
- **Fraudulent Claims**: ~9.6%
- **Class Imbalance**: Requires special handling

3.2 Financial Statistics

Metric	Legitimate	Fraudulent
Mean Amount	$7,200	$12,500
Median Amount	$5,000	$8,000
Max Amount	$95,000	$180,000

Key Insight: Fraudulent claims average **74% higher** amounts.

3.3 Claim Type Distribution

- **Outpatient**: 67% of claims
- **Inpatient**: 33% of claims
- **Fraud Rate**: Similar across both types

3.4 Top Fraud Indicators (Correlation with Fraud)

1. `amount_sum` (total provider revenue)
2. `revenue_per_patient` (avg collection per patient)
3. `claims_per_patient` (visit frequency)
4. `inpatient_ratio` (% hospitalization)

3.5 High-Risk Diagnosis Codes

ICD Code	Disease	Fraud Rate
44024	Atherosclerosis w/ Gangrene	65.2%
03842	E. Coli Septicemia	62.6%
V5789	Aftercare NEC	58.4%

---

4. Machine Learning Approach

4.1 The Problem with Individual Claims

Initial Approach (Failed):

- Train on individual claims → **62% accuracy**
- Why? Fraudulent providers have BOTH legitimate and fraudulent claims
- Model gets confused by mixed signals

4.2 Provider-Level Aggregation (Success)

Solution:

- Group 558,211 claims → 5,410 provider summaries
- Calculate aggregate statistics per provider

- Train on provider-level features → **94.82% accuracy**

4.3 Feature Engineering

28 Engineered Features:

Feature	Calculation	Why It Matters
`total_claims`	COUNT per provider	Volume indicator
`unique_patients`	COUNT DISTINCT patients	Patient diversity
`amount_mean`	AVG(amount)	Pricing behavior
`amount_sum`	SUM(amount)	Total revenue
`amount_std`	STDDEV(amount)	Consistency
`amount_max`	MAX(amount)	Outlier detection
`claims_per_patient`	claims / patients	Repeat visits
`revenue_per_patient`	revenue / patients	Per-patient billing
`inpatient_ratio`	inpatient / total	Service mix
`avg_diagnoses_per_claim`	diagnoses / claims	Upcoding indicator

4.4 Model Comparison

Model	Accuracy	ROC-AUC
Gradient Boosting	**94.82%**	**0.9683**
Random Forest	92.4%	0.9421
Logistic Regression	78.6%	0.8234

4.5 Why Gradient Boosting?

1. **Sequential Learning**: Each tree corrects previous errors
2. **Handles Imbalance**: Works well with 9.6% fraud rate
3. **Feature Importance**: Shows which features matter most
4. **Non-Linear Patterns**: Captures complex fraud behaviors

4.6 Model Hyperparameters

```
GradientBoostingClassifier(
    n_estimators=200,       # Number of trees
    max_depth=6,            # Tree depth
    learning_rate=0.1,      # Step size
    min_samples_split=10,   # Split threshold
    min_samples_leaf=5,     # Leaf size
    subsample=0.8,          # Sampling ratio
```

```
    random_state=42        # Reproducibility
)
```

---

5. Two-Layer Detection System

Layer 1: Rule-Based Detection

Catches **obvious fraud** immediately:

Rule	Condition	Action
Overpriced Claim	Amount > 2.5× expected	Flag
Excessive Diagnoses	> 15 diagnoses	Flag
Invalid Age	Age < 0 or > 120	Flag
Zero-Day Inpatient	Inpatient + 0 stay	Flag

Layer 2: ML Model

For claims passing rules:

1. Extract 28 features
2. Scale using StandardScaler
3. Run GradientBoostingClassifier
4. Get fraud probability (0-100%)

---

6. Model Evaluation

Confusion Matrix

```
               Predicted
               Legit  Fraud
Actual Legit    876     12
       Fraud     44    150
```

Metrics

Metric	Value
Accuracy	94.82%
Precision	92.6%
Recall	77.3%
F1-Score	84.3%
ROC-AUC	0.9683

Feature Importance (Top 10)

1. `amount_sum` — 14.2%
2. `revenue_per_patient` — 11.8%
3. `amount_mean` — 9.5%
4. `claims_per_patient` — 8.7%
5. `total_claims` — 7.6%
6. `chronic_conditions_sum` — 6.8%
7. `num_diagnoses_sum` — 6.2%
8. `amount_per_diagnosis_mean` — 5.8%
9. `inpatient_ratio` — 5.4%
10. `length_of_stay_mean` — 4.8%

---

7. Quick Reference (Viva Questions)

Q: Why provider-level aggregation?

A: The dataset marks PROVIDERS as fraudulent, not individual claims. A fraudulent provider files both legitimate and fraudulent claims, so individual claim training causes confusion.

Q: What is Feature Engineering?

A: Creating new calculated columns from raw data. Example: `claims_per_patient = total_claims / unique_patients`

Q: Why Gradient Boosting over Random Forest?

A: GB builds trees sequentially, correcting previous errors. RF builds parallel trees and averages. GB achieves 2.4% higher accuracy on our dataset.

Q: What is ROC-AUC?

A: Area Under the Receiver Operating Characteristic Curve. Measures how well the model distinguishes fraud from legitimate. 0.9683 = excellent discrimination.

Q: What is Standard Scaling?

A: Converting all features to have mean=0 and std=1. Required because features have different ranges (age: 0-100, amount: 0-180000).

Q: How to handle class imbalance?

A: Used `class_weight='balanced'` in Random Forest and provider-level aggregation which balanced the fraud ratio.

---

8. Key Numbers to Remember

What	Value
Total Claims	558,211
Unique Providers	5,410
Model Accuracy	94.82%

ROC-AUC	0.9683
Features Used	28
Disease Prices	6,016
Fraud Rate	9.6%
Top Feature	amount_sum (14.2%)