Project Title: Healthcare Fraud Detection
Document Type: Data Preparation & Class Imbalance Strategy
Prepared For: Machine Learning – Winter 2025
Prepared By: Abdullah Mahmoud, Omar Sherif, Omar Ayman, Ahmed Hassan, Ahmed Amr
Date: 28 November 2025

PHASE 1 — DATA EXPLORATION & FEATURE ENGINEERING

Notebook: 01_data_exploration_and_feature_engineering.ipynb

---

1. Purpose of the Notebook

This notebook prepares the healthcare provider dataset for fraud detection.
It covers data loading, cleaning, exploratory analysis, feature engineering, provider-level aggregation, and exporting processed data for modeling.

---

2. Data Loading

The dataset is imported and examined through:

- Head previews
- Datatype checks
- Identification of missing values
  The goal is to understand structure, quality, and potential issues.

---

3. Data Cleaning

Cleaning operations include:

- Correcting datatypes (numeric, categorical, datetime)

- Standardizing column names
- Handling missing values appropriately
- Removing duplicates
- Ensuring ID consistency across all records

This results in a stable and reliable dataset.

---

4. Exploratory Data Analysis (EDA)

EDA includes:

- Distribution analysis of numerical fields
- Boxplots to highlight outliers
- Correlation heatmaps to study relationships
- Categorical summaries
  Insights show noticeable differences between fraud and legitimate providers.

---

5. Feature Engineering

New features added include:

- Aggregate claim metrics (totals, averages, counts)
- Ratios such as inpatient percentage
- Unique beneficiary counts
- Encoded categorical fields
- Scaled or normalized numeric fields

These features help models capture fraud-related patterns.

---

6. Provider Aggregation

The dataset is grouped by provider to compute meaningful provider-level metrics, including:

- Total claim counts
- Total and average claim amounts
- Percentage of inpatient claims
- Number of unique beneficiaries
- Service variety

Fraud patterns typically emerge more clearly at the provider level, making these features essential.

---

7. Exported Outputs (Phase 1)

The following files are produced:

- EDA visualizations
- provider_features_final.csv

PHASE 2 — CLASS IMBALANCE STRATEGY DOCUMENT

Healthcare Fraud Detection – Class Imbalance Analysis
Date: 28 November 2025
Prepared by: Abdallah

---

1. Dataset Characteristics

Total Providers: 5410
Fraudulent Providers: 506 (9.35%)

Legitimate Providers: 4904 (90.65%)
Imbalance Ratio: approximately 9.7 to 1

The dataset is heavily imbalanced and requires appropriate techniques before training models.

---

2. Selected Strategy: Hybrid Approach

A. SMOTE (Primary Method)

Sampling Strategy: minority class increased to 50% of majority
K-Neighbors: 5
Random State: 42
Total Samples After SMOTE: 7356

B. SMOTEENN (Alternative Method)

Used for reducing noise and improving boundary clarity.
Samples Removed by ENN: 1772
Total Samples After Cleaning: 9128

C. Class Weighting (Baseline Method)

Fraud Weight: 5.35
Legit Weight: 0.55
Useful for models such as Random Forest and XGBoost.

---

3. Rationale Behind Strategy

- SMOTE avoids losing legitimate data and preserves distribution.
- Moderate oversampling prevents overfitting caused by too many synthetic samples.
- Class weighting supports models that naturally handle imbalance.
- SMOTEENN improves generalization by removing noisy boundary points.

---

4. Exported Datasets for Phase 2

The following final datasets were generated:

- data_original_with_weights.csv
- data_smote_resampled.csv
- data_smoteenn_cleaned.csv
- class_weights.json

These datasets support flexible modeling strategies.

---

5. Recommended Evaluation Metrics

Primary Metric: Precision-Recall AUC (PR-AUC).
It is best suited for imbalanced classification and focuses on minority-class detection.

Additional Metrics:

- F1-Score (Target: above 0.70)
- Recall (Target: above 0.75)
- Precision (Target: above 0.65)

Avoid using Accuracy, as it is misleading with a 9.7:1 imbalance — predicting all providers as “legit” yields about 90.6% accuracy.

---

6. Cross-Validation Strategy

- Use Stratified K-Fold with k=5 to preserve class ratios.
- Apply SMOTE only to the training folds to prevent data leakage.
- Keep validation folds in original distribution.
  This ensures realistic performance evaluation.

---

7. Key Statistical Findings

Fraudulent providers show significantly higher activity levels. Differences include:

- Much higher total claim counts
- Much higher overall paid amounts
- Higher average claim costs
- Higher percentage of inpatient claims
- More unique beneficiaries

These consistent differences support the effectiveness of engineered features.

SUMMARY

Phase 1 provides the foundational data preparation steps, including cleaning, exploration, and feature engineering.
Phase 2 resolves the class imbalance problem with SMOTE, SMOTEENN, and class weighting strategies.
Together, they form a complete workflow that prepares the dataset for Phase 3 — model development and evaluation.

Say less — I'll **add the Phase 3 Modeling Notebook** to your documentation exactly like the previous phases, following the same clean, Google-Docs-friendly, no-tables style.

I parsed your notebook, checked the code structure, and here is the full **Phase 3 – Modeling Documentation** you can paste directly under Summary.

PHASE 3 — MODELING & EVALUATION PIPELINE

Notebook: 02_modeling.ipynb

---

1. Purpose of the Notebook

This notebook builds and trains several machine learning models for healthcare fraud detection.
It takes as input the preprocessed datasets generated in Phases 1 and 2 and applies multiple classification algorithms using consistent parameters, **cross-validation**, and **parameter tuning** to ensure robust and optimized performance.

The goal is to identify the most effective model for detecting fraudulent providers.

---

2. Data Preparation for Modeling

Before training, the notebook performs several preparation steps:

- Loads the selected resampled dataset (SMOTE, SMOTEENN, or original + class weights).
- Identifies the correct fraud label column (since different datasets may use different naming).
- Splits the dataset into training and testing partitions.
- Drops provider identifier columns to prevent model leakage.
- Ensures features are purely numerical and model-friendly.

This establishes a clean input space for all models.

Parameter Tuning

The notebook applies **hyperparameter tuning** to improve the performance of selected models such as Random Forest, Logistic Regression, and Gradient Boosting.

Tuning includes searching for:

- Depth of trees
- Number of estimators
- Regularization strength

- Learning rate
- Minimum samples per split

This ensures each model operates under optimized configurations rather than relying on default parameters.

Cross-Validation

All models are evaluated using **Stratified K-Fold cross-validation**, consistent with the imbalance strategy in Phase 2.

Cross-validation ensures:

- Fraud/legit proportions remain stable across folds
- SMOTE or SMOTEENN is applied **only** on the training folds
- Validation folds remain untouched to prevent leakage
- Performance is measured consistently across splits

This produces reliable and generalizable model results.

---

3. Models Implemented

The notebook trains a suite of machine learning models using the cleaned and engineered feature set.

Models include:

A. **Logistic Regression**
B. **Decision Tree Classifier**
C. **Random Forest Classifier**
D. **Support Vector Machine (SVM)**
E. **Gradient Boosting Classifier**

Each model uses the same feature space to ensure fair comparison.

---

4. Training Procedure

For each model:

- The algorithm is initialized (tuned parameters when applicable).
- The model is trained using the training partition.
- Cross-validation scores are computed.
- Predictions and performance metrics are generated on the final test set.
- Outputs are stored for Phase 4 evaluation.

This creates a unified and optimized modeling pipeline.

---

5. Evaluation Logic

The notebook evaluates models using:

- Precision-Recall AUC (primary metric)
- F1-Score
- Recall
- Precision

These metrics are consistent with Phase 2 priorities and support performance comparison.

---

6. Feature Handling

The notebook ensures:

- Provider IDs are dropped
- All remaining features are strictly numerical
- No leakage occurs between train and test

This preserves model integrity.

---

7. Model Exporting

After training, the notebook exports:

- Trained model files
- Test datasets
- Cross-validation results
- Prediction outputs

These artifacts are used in Phase 4.

---

8. Summary of Phase 3

Phase 3 completes model development by:

- Training multiple ML models
- Applying parameter tuning
- Performing Stratified K-Fold cross-validation
- Evaluating models with fraud-appropriate metrics
- Exporting all artifacts for final analysis

PHASE 4 — MODEL EVALUATION & PERFORMANCE ANALYSIS

Notebook: 03_evaluation.ipynb

---

1. Purpose of the Notebook

This notebook performs the final evaluation of all trained machine learning models.
It loads the predictions and test sets generated during Phase 3, calculates performance metrics, visualizes results, and identifies the best-performing model for healthcare fraud detection.

The goal is to provide a complete, metric-based comparison aligned with the imbalanced classification strategy defined in Phase 2.

---

2. Loading Predictions and Test Data

The notebook begins by importing:

- Saved test datasets
- Model predictions
- Evaluation utilities
- Performance metric functions

Multiple models are evaluated, ensuring the comparison is consistent and fair.

---

3. Metrics Calculated

The notebook computes the following metrics for every model:

- Precision-Recall AUC (PR-AUC) — primary metric
- F1-Score
- Recall (True Positive Rate)
- Precision
- ROC-AUC (secondary, optional)

These metrics allow detection of both the ability to catch fraud (recall) and avoid false accusations (precision).

---

4. Confusion Matrix Analysis

Confusion matrices are generated for each model.
They show:

- True Positives (fraud correctly detected)
- False Positives (legit flagged as fraud)
- True Negatives
- False Negatives (fraud missed)

These breakdowns help explain model behavior beyond numeric scores.

---

5. Precision-Recall Curves

The notebook generates Precision-Recall curves for each model.
These curves are more meaningful than ROC curves in imbalanced datasets.
A higher curve and larger area represent better fraud detection.

---

6. ROC Curves (Optional)

ROC curves are also plotted, though secondary for imbalance.
They illustrate how well each model separates fraud vs legit providers across thresholds.

---

7. Model Comparison Summary

The notebook simulates ranking all models using the priority metrics:

- Best model by PR-AUC
- Best model by recall
- Tradeoff between recall and precision

- Stability of performance across seeds

This provides a clear view of which model performs best for detecting healthcare provider fraud.

---

8. Feature Importance (If Supported)

For tree-based models (Random Forest, Gradient Boosting), the notebook extracts and visualizes feature importances.
This shows which features contributed most to fraud detection, such as:

- Total claim count
- Total amount billed
- Mean claim value
- Inpatient claim percentage
- Number of unique beneficiaries

These insights validate the engineered features from Phase 1.

---

9. Best Model Selection

Based on the final metric evaluation, the notebook identifies:

- The overall best model
- The best model under high-recall requirements
- Any model with suspiciously high/low performance potentially caused by leakage or imbalance issues

---

10. Exported Evaluation Artifacts

The notebook exports multiple final outputs:

- Final metrics summary (text and plots)
- All confusion matrices
- Precision-Recall and ROC curves
- Best-model selection information

These serve as the final stage before deployment or report submission.

---

11. Summary of Phase 4

Phase 4 completes the machine learning pipeline by offering a comprehensive evaluation of all trained models.
This phase confirms whether the resampling strategy, engineered features, and chosen models effectively identify fraudulent providers.

It is fully aligned with the priorities established in earlier phases and finalizes the fraud detection system’s performance analysis.