Detecting Fraudulent Healthcare Providers Using Machine Learning

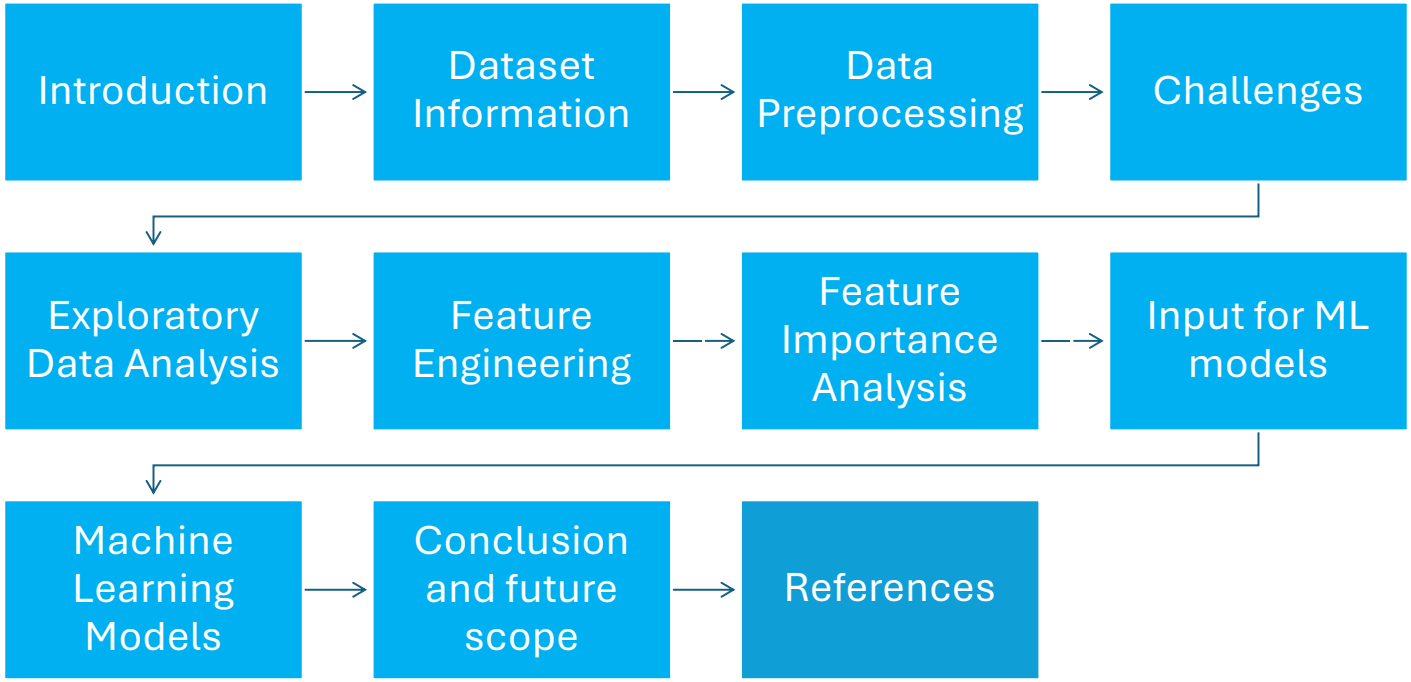
DATA 606 Capstone in Data Science

Professor: Zeynep Kacar

By

Amrita Chandrasekar(NH53017)(Group2)

Outline


Introduction
Dataset Information
Data Preprocessing
Challenges
Exploratory Data Analysis
Feature Engineering
Feature Importance Analysis
Input for ML models
Machine Learning Models
Conclusion and future scope
References

Introduction

Problem Statement

- In the U.S. healthcare system, fraud costs taxpayers billions of dollars every year. A real-world example is the case of a healthcare provider who submits fraudulent Medicare claims for services that were never provided, such as billing for unnecessary medical tests or overcharging for treatments. In 2019, it was reported that healthcare fraud led to losses of over $60 billion in the U.S. alone. This fraudulent behavior not only strains the financial stability of healthcare systems but also puts patient safety at risk, as resources are diverted away from legitimate care.
- Traditional fraud detection methods, which often rely on manual audits and rule-based checks, struggle to keep up with the sheer volume and complexity of claims data, making it difficult to identify subtle patterns of fraud. This project aims to harness **machine learning** to automate the process, providing a more efficient, scalable, and accurate solution for detecting fraudulent healthcare providers, ultimately reducing financial losses and ensuring the quality and integrity of care provided to patients.

Objectives

- Detect fraudulent healthcare providers using machine learning.
- Build an interactive **Streamlit app** for predicting fraudulent providers

Research Questions

- How effectively can machine learning algorithms identify potentially fraudulent medical providers based on claims data?
- Which features in the dataset are most indicative of fraudulent behavior?
- How can we address the class imbalance in the dataset, where fraudulent claims are far fewer than legitimate ones?

Introduction

Background and Context
Healthcare fraud, including falsified billing and inflated claims, severely impacts the financial integrity of healthcare systems and threatens patient care quality. Traditional fraud detection methods are often inadequate due to the vast volume and complexity of claims data. Machine learning offers a transformative solution by analyzing historical claims data to detect subtle, non-obvious fraud patterns. This project utilizes a machine learning models to predict fraudulent healthcare providers, using a dataset that includes comprehensive claims, beneficiary details, and fraud labels. The goal is to improve detection accuracy, reduce financial losses, and enhance the overall efficiency of healthcare fraud prevention.

Significance of the Research Question
Financial Impact
Patient Safety at Risk
System Integrity
Resource Allocation
Ineffective Traditional Methods

Why Is this question important?
Public Trust
Enlightens the need for Proactive Fraud Prevention

What are the key concepts or theories that relate to your project?
Data Imbalance Handling
Feature Engineering
Supervised Classification

Dataset Overview

- Dataset is taken from https://www.kaggle.com/datasets/rohitrox/healthcare-provider- fraud-detection-analysis/data .
- The dataset consists of four main components
- **Inpatient Data**
  Claims filed for patients admitted to hospitals. Includes admission and discharge dates, diagnosis codes.
  It has the shape of (40474, 30)
- **Outpatient Data**
  Claims filed for patients who visit hospitals without admission.
  It has the shape of (517737, 27)
- **Beneficiary Data**
  Contains beneficiary KYC details. Includes health conditions and regional information.
  It has the shape of (138556, 25)
- **Train Data(Fraud Labels)**
  It is labelled as fraud/non-fraud with respect to healthcare providers.
  It has the shape of (5410, 2)

Data Preprocessing

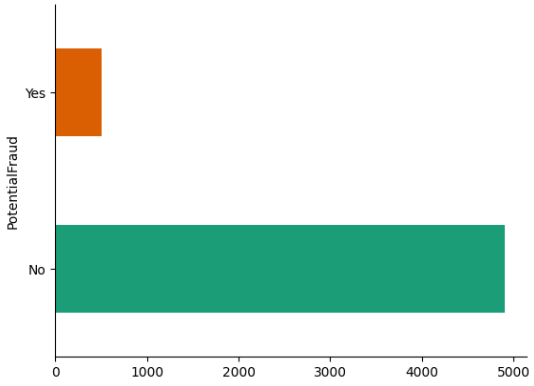
- Dropping columns that have more than 50% null values for inpatient, outpatient and beneficiary datasets individually.
- Merging “inpatient” and “outpatient” as “merged_data” with the help of outer join function. Then merging “merged_data” with beneficiary dataset on BeneID column with outer join function as “final merged” dataset.
- Shape of Final Merged Dataset: (558211, 46)
- Now combining “final merged” dataset with “train dataset” on provider column with the help of inner join as “df_final”
- Shape of df_final dataset: (558211, 47).
- Removing columns from df_final that have more than 50%null values as they are not part of analysis.
- Shape of df_final dataset:(558211, 34)

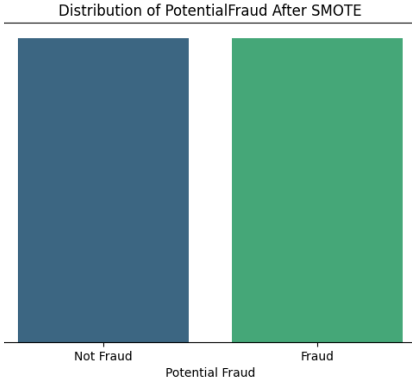
Data preprocessing

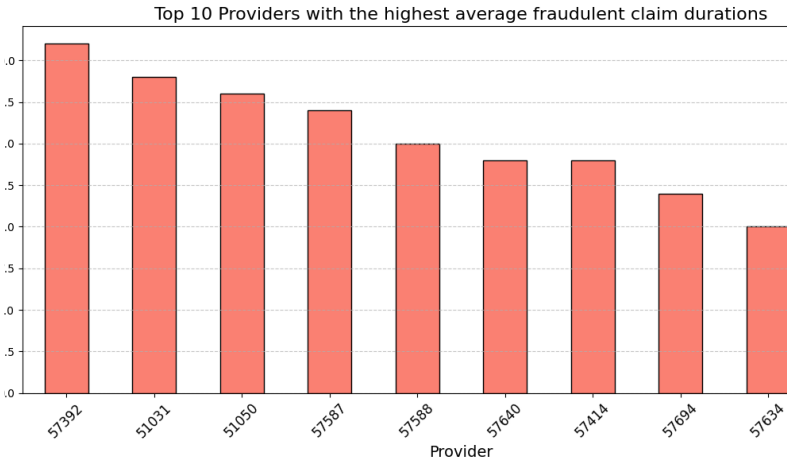
- Checking for duplicate values
- Checking for null values: AttendingPhysician 1508, DeductibleAmtPaid 899, ClmDiagnosisCode_1 10453, ClmDiagnosisCode_2 195606
- To handle null values, “unknown” is used in place of null values for the column “AttendingPhysician” while “median” and “mode” is used for columns “DeductibleAmtPaid”,”ClmDiagnosisCode_1” and “ClmDiagnosisCode_2” respectively.
- Converting date to datetime format
- Performed label encoding to convert categorical features to numerical features for making data ready for visualizations

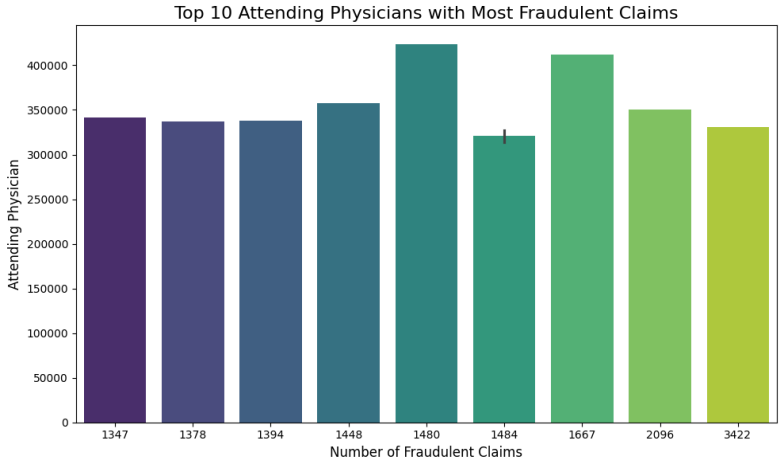
Challenges

- Class Imbalance of the target variable Potentialfraud.
- SMOTE is used to handle class imbalance. SMOTE (Synthetic Minority Over-sampling Technique) is a technique used to address class imbalance by generating synthetic samples for the minority class.
- It works by selecting a sample from the minority class, finding its nearest neighbors, and then creating new samples by interpolating between the original sample and its neighbors. This helps balance the dataset, preventing the model from being biased toward the majority class
- After SMOTE, combined features (X_resampled) and target (y_resampled) into a single data frame as “balanced_df” having shape (690830, 32).

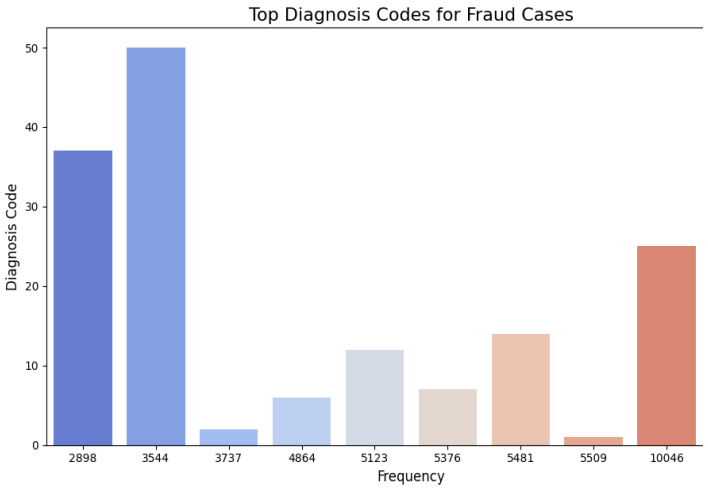


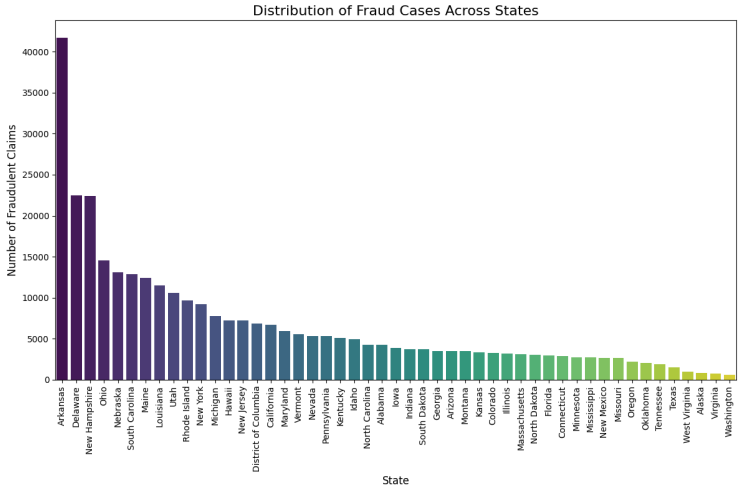




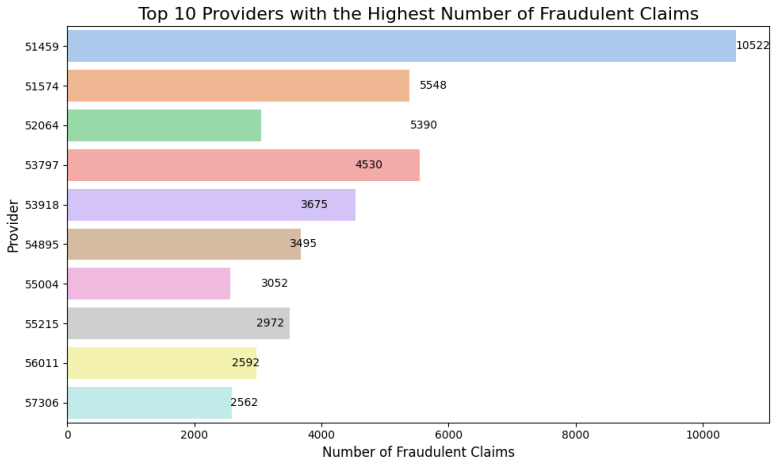


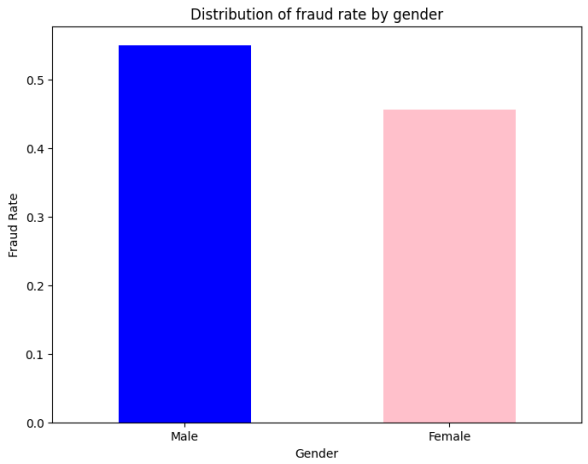
Exploratory Data Analysis





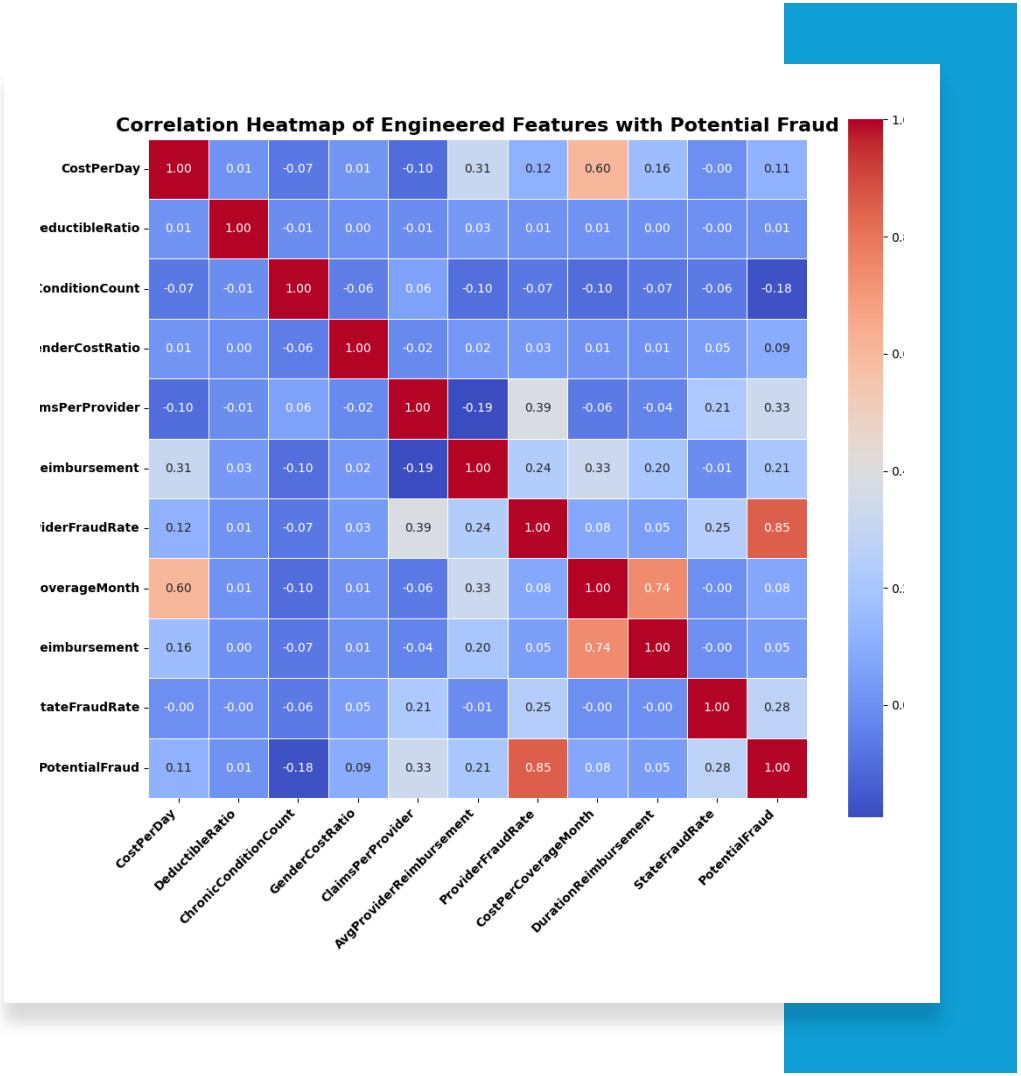
Exploratory Data Analysis

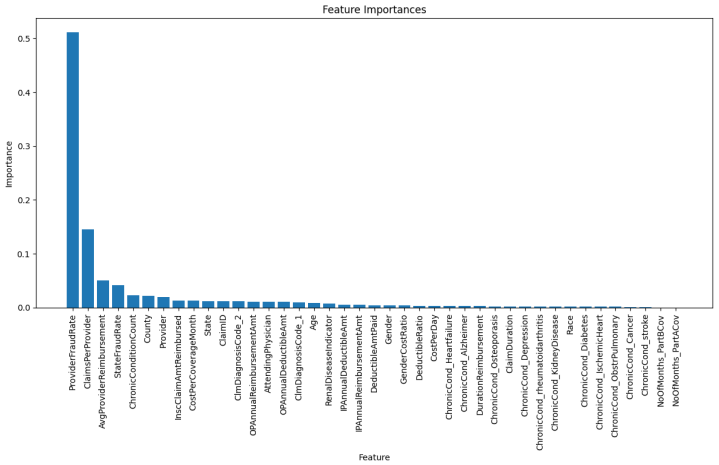


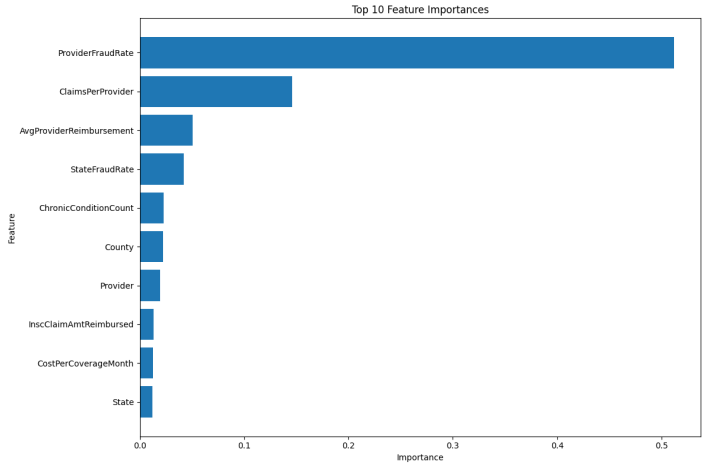


Exploratory Data Analysis

Feature Engineering







Feature Importance Analysis

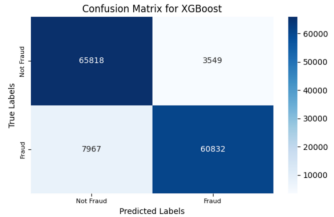
Input for ML Models

- Random Forest is implemented again on these top 10 features to prepare the data for the model implementation.
- The dataset is divided into 4 parts namely before model implementation
- X_train_top.csv
  This file contains the training dataset with only the top features selected based on feature importance. It is used to train machine learning models to detect fraudulent claims.
- X_test_top.csv
  This file contains the testing dataset with only the top features. It is used to evaluate the trained model's performance by testing its ability to predict on unseen data.
- y_train.csv
  This file contains the target variable (PotentialFraud) corresponding to the training dataset. It is used as the true labels during the model training process.
- y_test.csv
  This file contains the target variable (PotentialFraud) corresponding to the testing dataset. It is used as the true labels to evaluate the accuracy, precision, recall, and F1-score of the predictions made by the trained model.

Model Name: XGBoost
Train Accuracy: 0.9833
Test Accuracy: 0.9812
Train Recall: 0.9729
Test Recall: 0.9699
Train F1: 0.9832
Test F1: 0.9809
Train AUC: 0.9978
Test AUC: 0.9973

Classification Report (Test):

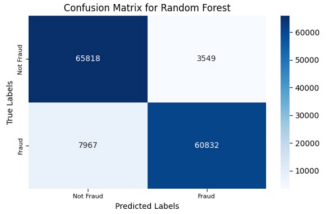
	precision	recall	f1-score	support
0	0.97	0.99	0.98	69367
1	0.99	0.97	0.98	68799
accuracy			0.98	138166
macro avg	0.98	0.98	0.98	138166
weighted avg	0.98	0.98	0.98	138166



Model Name: Random Forest
Train Accuracy: 1.0000
Test Accuracy: 0.9654
Train Recall: 1.0000
Test Recall: 0.9456
Train F1: 1.0000
Test F1: 0.9646
Train AUC: 1.0000
Test AUC: 0.9960

Classification Report (Test):

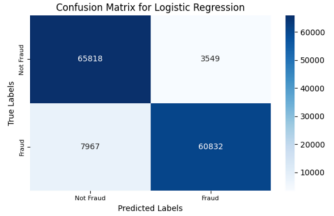
	precision	recall	f1-score	support
0	0.95	0.99	0.97	69367
1	0.98	0.95	0.96	68799
accuracy			0.97	138166
macro avg	0.97	0.97	0.97	138166
weighted avg	0.97	0.97	0.97	138166



Model Name: Logistic Regression
Train Accuracy: 0.9175
Test Accuracy: 0.9167
Train Recall: 0.8857
Test Recall: 0.8842
Train F1: 0.9149
Test F1: 0.9135
Train AUC: 0.9717
Test AUC: 0.9708

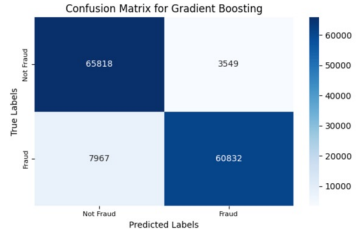
Classification Report (Test):

	precision	recall	f1-score	support
0	0.89	0.95	0.92	69367
1	0.94	0.88	0.91	68799
accuracy			0.92	138166
macro avg	0.92	0.92	0.92	138166
weighted avg	0.92	0.92	0.92	138166

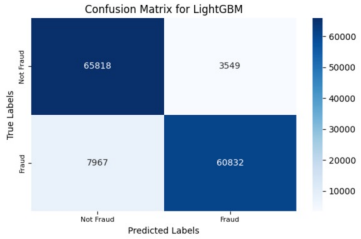


Machine Learning Models

```
Model Name: Gradient Boosting
Train Accuracy: 0.9509
Test Accuracy: 0.9502
Train Recall: 0.9277
Test Recall: 0.9261
Train F1: 0.9498
Test F1: 0.9488
Train AUC: 0.9900
Test AUC: 0.9896

Classification Report (Test):
              precision    recall  f1-score   support

           0       0.93      0.97      0.95     69367
           1       0.97      0.93      0.95     68799

    accuracy                           0.95    138166
   macro avg       0.95      0.95      0.95    138166
weighted avg       0.95      0.95      0.95    138166
```

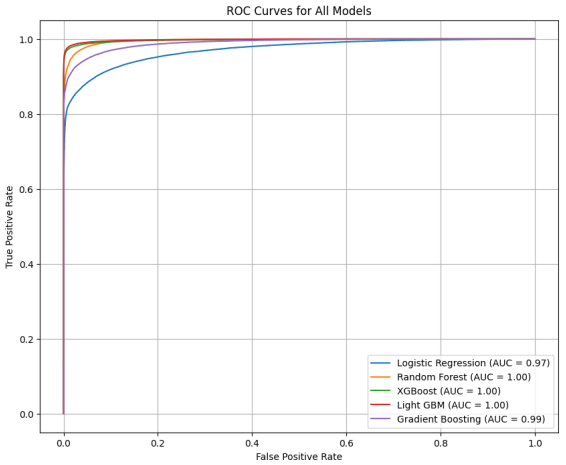



```
Model Name: LightGBM
Train Accuracy: 0.9844
Test Accuracy: 0.9835
Train Recall: 0.9743
Test Recall: 0.9726
Train F1: 0.9842
Test F1: 0.9832
Train AUC: 0.9983
Test AUC: 0.9981

Classification Report (Test):
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     69367
           1       0.99      0.97      0.98     68799

    accuracy                           0.98    138166
   macro avg       0.98      0.98      0.98    138166
weighted avg       0.98      0.98      0.98    138166
```



Machine Learning Models

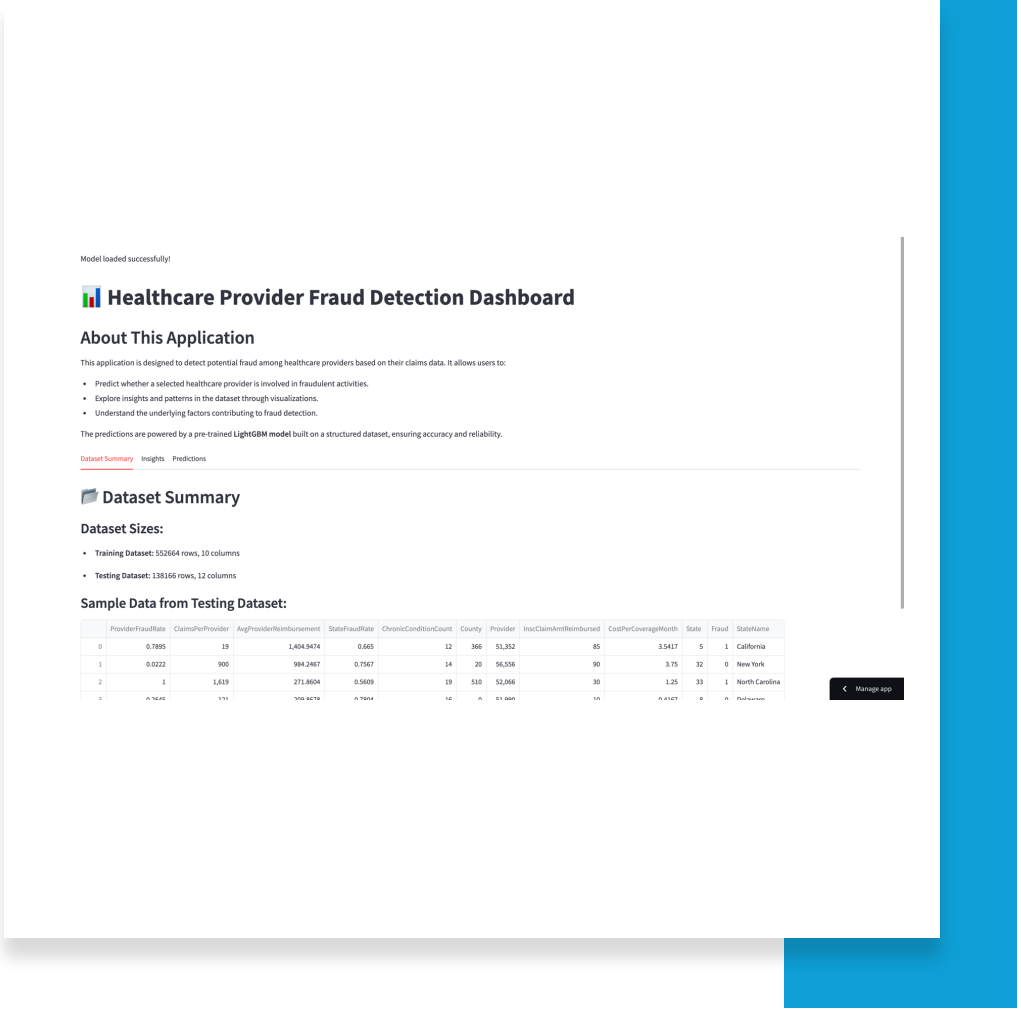
Model Name	Train Recall	Test Recall	Train F1	Test F1	Train AUC	Test AUC
Logistic Regression	0.885737	0.884199	0.914863	0.913531	0.971703	0.970783
Random Forest	0.999986	0.945552	0.999993	0.96457	1	0.996013
XGBoost	0.972937	0.969869	0.983181	0.980911	0.997846	0.997306
Light GBM	0.974318	0.972616	0.984211	0.98322	0.998321	0.998073
Gradient Boosting	0.927669	0.926104	0.949797	0.948803	0.990036	0.98964



Comparison of Model Evaluation Metrics

The LightGBM(gradient boosting model) outperforms other models

Streamlit App Click Here



Results and Insights

Model Performance and Accuracy

LightGBM(Gradient Boosting) showed 98% overall accuracy with a balanced F1-score of 0.98, indicating strong predictive capability.

Key Metrics

- Test Recall: 0.9726 (high sensitivity for detecting fraudulent claims)
- AUC: 0.9981 (excellent at distinguishing fraud from non-fraud)

Addressing Class Imbalance

- SMOTE Improved model performance by generating synthetic samples for the minority class, enhancing recall and F1-scores.
- Ensured the model could effectively detect fraud without bias towards non-fraudulent claims.

Results and Insights

Key Features for Fraud Detection

- **ProviderFraudRate**: Strong predictor with 0.85 correlation to fraud.
- **ClaimsPerProvider**: Strongly correlated (0.33) with fraud, indicating suspicious provider activity.
- **CostPerCoverageMonth**: High correlation (0.74) with fraud, indicating potential fraudulent billing.

Streamlit Dashboard Insights

- Healthcare administrators can interactively assess fraud risks with instant fraud predictions.
- Displays key visualizations such as fraud distribution by state, fraud vs. non-fraud counts, and feature importance.

Financial and Operational Impact

- **Cost Savings**: Early detection of fraud prevents financial losses from fraudulent claims.
- **Operational Efficiency**: Fraud detection reduces manual review time and speeds up claim processing.

Conclusion and Future Scope

- The **LightGBM model** demonstrated excellent performance in all metrics, achieving **98% accuracy** in detecting fraudulent healthcare providers.
- Key features like **ProviderFraudRate** and **ClaimsPerProvider** were identified as critical for fraud detection.
- The **SMOTE** technique effectively addressed class imbalance, boosting the model's ability to identify fraud.
- The integration of a **Streamlit dashboard** allows real-time predictions, empowering healthcare administrators to make faster, informed decisions.
- This project contributes to improving healthcare system integrity by reducing fraud, optimizing resources, and ensuring cost-effective care.

Future Scope

- **Anomaly Detection**: Implement **unsupervised learning models** to detect emerging or unknown fraud patterns that may not be captured by supervised models.
- **Real-time Integration**: Integrate the fraud detection model with live healthcare claim systems for immediate fraud alerts, reducing response times and preventing fraud more effectively.

References

- Bauder, R. A., & Khoshgoftaar, T. M. (2017). Medicare fraud detection using machine learning methods. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 858-865). IEEE. https://doi.org/10.1109/ICMLA.2017.00-48
- Garmdareh, M. S., Neysiani, B. S., Nogorani, M. Z., & Bahramizadegan, M. (2023). A Machine Learning-based Approach for Medical Insurance Anomaly Detection by Predicting Indirect Outpatients' Claim Price. In *2023 9th International Conference on Web Research (ICWR)* (pp. 129-134). IEEE. https://doi.org/10.1109/ICWR57742.2023.10139290

THANK YOU