# Healthcare Claims Fraud Detection

# 👤 Role: Machine Learning Engineer / Data Scientist

# Business Objective

The objective of this project is to build an intelligent fraud detection system to:

* **Identify** fraudulent healthcare claims with high precision.
* **Reduce** financial losses and improve fraud investigation efficiency.
* **Assist** insurance companies with data-driven decision-making and risk assessment.

# Project Workflow

* Data Collection
* Data Cleaning & Preprocessing
* Exploratory Data Analysis (EDA)
* Feature Engineering
* Model Training & Hyperparameter Optimization
* Evaluation & Insights
* Deployment via Streamlit Dashboard

# Tools & Technologies

* Languages: Python 3.x
* Libraries: Pandas, NumPy, Seaborn, Matplotlib, scikit-learn, CatBoost, Plotly, PyYAML
* Environment: Jupyter Notebook, VS Code, Streamlit
* Modeling: CatBoost Classifier with Randomized/Grid Search optimization

# github repository

<https://github.com/SurajKhodade15/us-healthcare-claims-fraud-ml>

# pROBLEM STATEMENT

Healthcare fraud is a significant issue, costing billions annually. Insurance providers need an automated, scalable, and accurate solution to detect fraudulent claims.

This project delivers a machine learning-based fraud detection system using a CatBoost classifier, offering real-time risk prediction and interactive visualization through a web interface.

# PROJECT PHASES

**Phase 1: Data Cleaning & Exploratory Analysis**

* Addressed missing values and outliers.
* Analysed data distributions and fraud patterns.
* Generated visual insights into demographic, claim, and provider-related features.
* Key Insights:
* Fraud claims often involve higher claim amounts, cross-state providers, and longer stays.
* Certain provider types and diagnosis codes have increased fraud likelihood.

**Phase 2: Feature Engineering**

* Created 15+ new features including ratios, log-transforms, risk categories, and flags.
* Applied One-Hot Encoding and Target Encoding for categorical variables.
* Standardized numeric features to improve model performance.

**Phase 3: Model Development & Optimization**

* Selected CatBoost for its superior performance on tabular and categorical data.
* Implemented Stratified K-Fold Cross-Validation.
* Performed Hyperparameter Tuning using GridSearchCV and RandomizedSearchCV.
* Evaluated using ROC-AUC, Precision, Recall, and F1-score.

**Phase 4: Deployment**

* Integrated the trained model into a Streamlit web application.
* Deployed the application for real-time fraud detection:

🔗 [Live App](https://us-healthcare-claims-fraud-ml-f5mugbyjlwjq8klqqvrasy.streamlit.app/)

# Architecture & Project Structure

**Project Root: us-healthcare-claims-fraud-ml/**

**📄 README.md** – Project overview and documentation

**📄 requirements.txt** – List of dependencies

**📄 main.py** – Driver script orchestrating the ML pipeline**1️⃣ Data Layer (/data)**

Original and Processed Data Sets**2️⃣ Notebooks Layer (/notebooks)**

**01\_eda.ipynb** – Exploratory Data Analysis

**02\_feature\_engineering.ipynb** – Feature engineering & transformations

**03\_model\_experiments.ipynb** – Model training & evaluation experiments**3️⃣ Source Code Layer (/src)**

**data\_preprocessing.py** – Functions for data cleaning & preprocessing

**train\_model.py** – Training pipeline with hyperparameter optimization

**evaluate\_model.py** – Model evaluation & metrics generation

**predict.py** – Inference and prediction logic

**utils.py** – Helper functions & configuration handlers**4️⃣ Model Layer (/models)**

**cat\_boost\_model.pkl** – Serialized trained CatBoost model

**model\_metadata.json** – Metadata for the trained model**5️⃣ Streamlit Application (/streamlit\_app)**

**app.py** – Main Streamlit dashboard

**components/** – (Optional) UI custom components**6️⃣ Configuration Layer (/config)**

**settings.yaml** – Configuration file for paths & hyperparameters**7️⃣ Reports & Visualizations (/reports)**

**eda\_visualizations.png** – Generated EDA charts

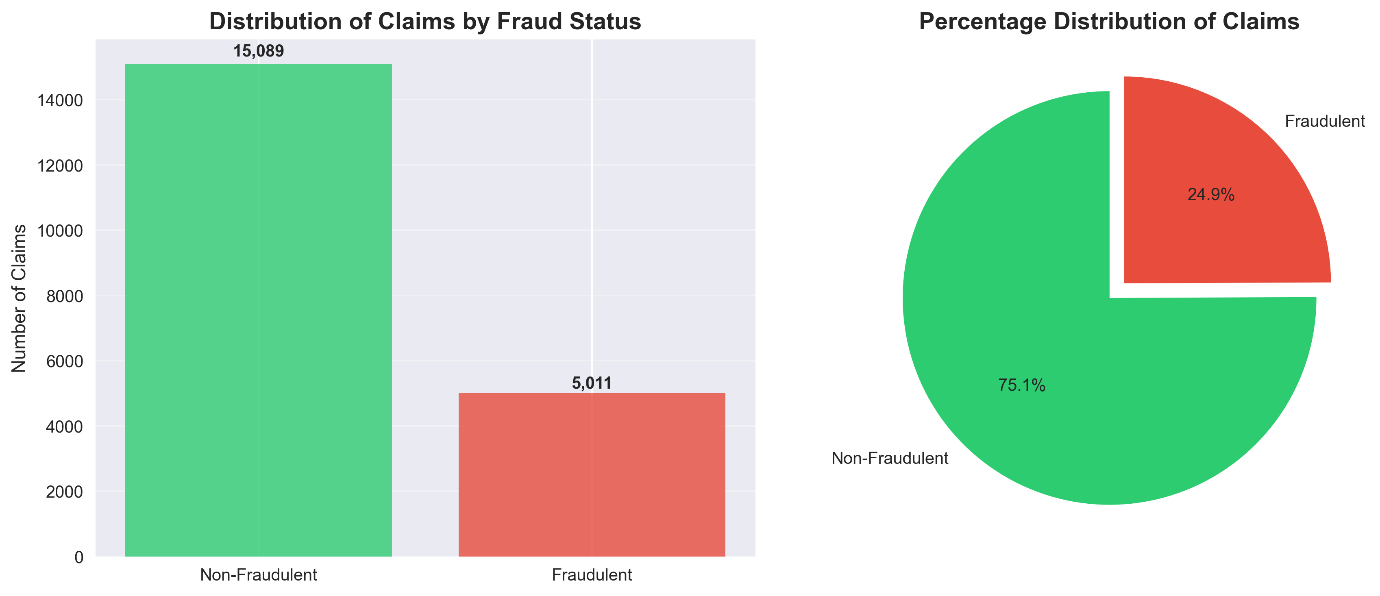
**model\_performance.png** – Model performance plots

# Key Visualizations

**1️⃣ Fraud Distribution**

This visualization illustrates the proportion of fraudulent vs. non-fraudulent claims in the dataset.

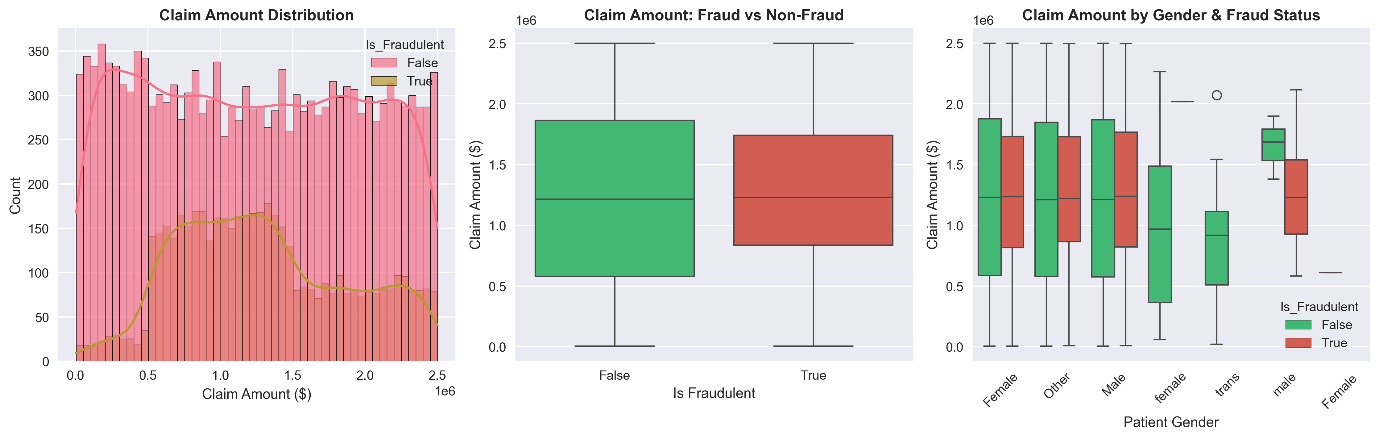
* Highlights the class imbalance problem typically observed in fraud datasets.
* Helps in determining whether resampling techniques may be needed during model **training.**



**2️⃣ Claim Amount Analysis**

A boxplot and histogram analysis of claim amounts segmented by fraud status.

* Fraudulent claims generally show higher variance and larger amounts compared to legitimate claims.
* Outliers in claim amounts can provide strong signals for fraud detection.



**3️⃣ Age vs. Fraud Patterns**

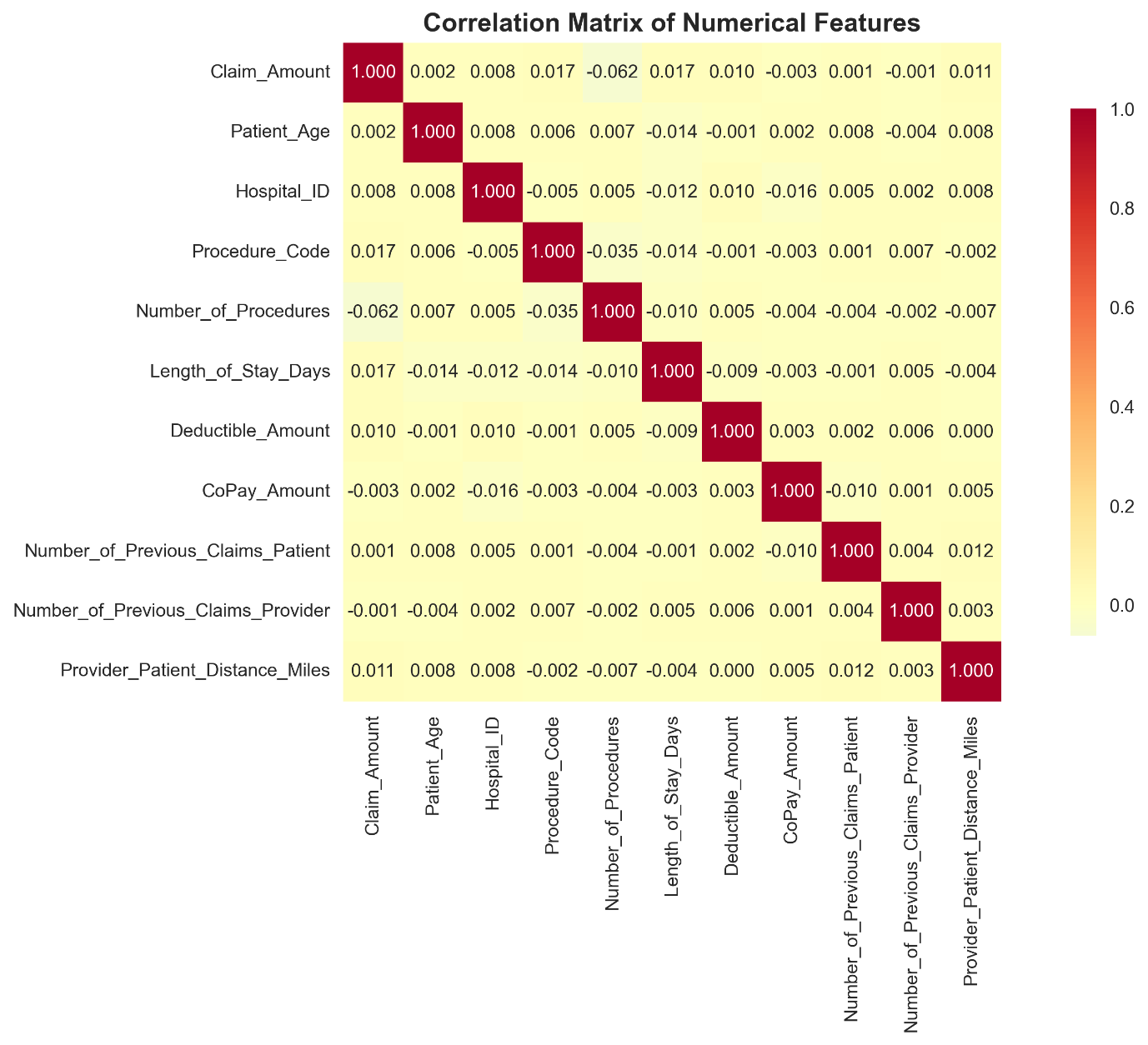
This chart analyses the relationship between patient age and fraud probability.

* Certain age groups exhibit higher fraud tendencies, revealing behavioural patterns.
* Helps in crafting age-related features (e.g., Patient\_Age\_Group).
* 

**4️⃣ Feature Correlation Heatmap**

A heatmap showing the correlation matrix of all numerical features.

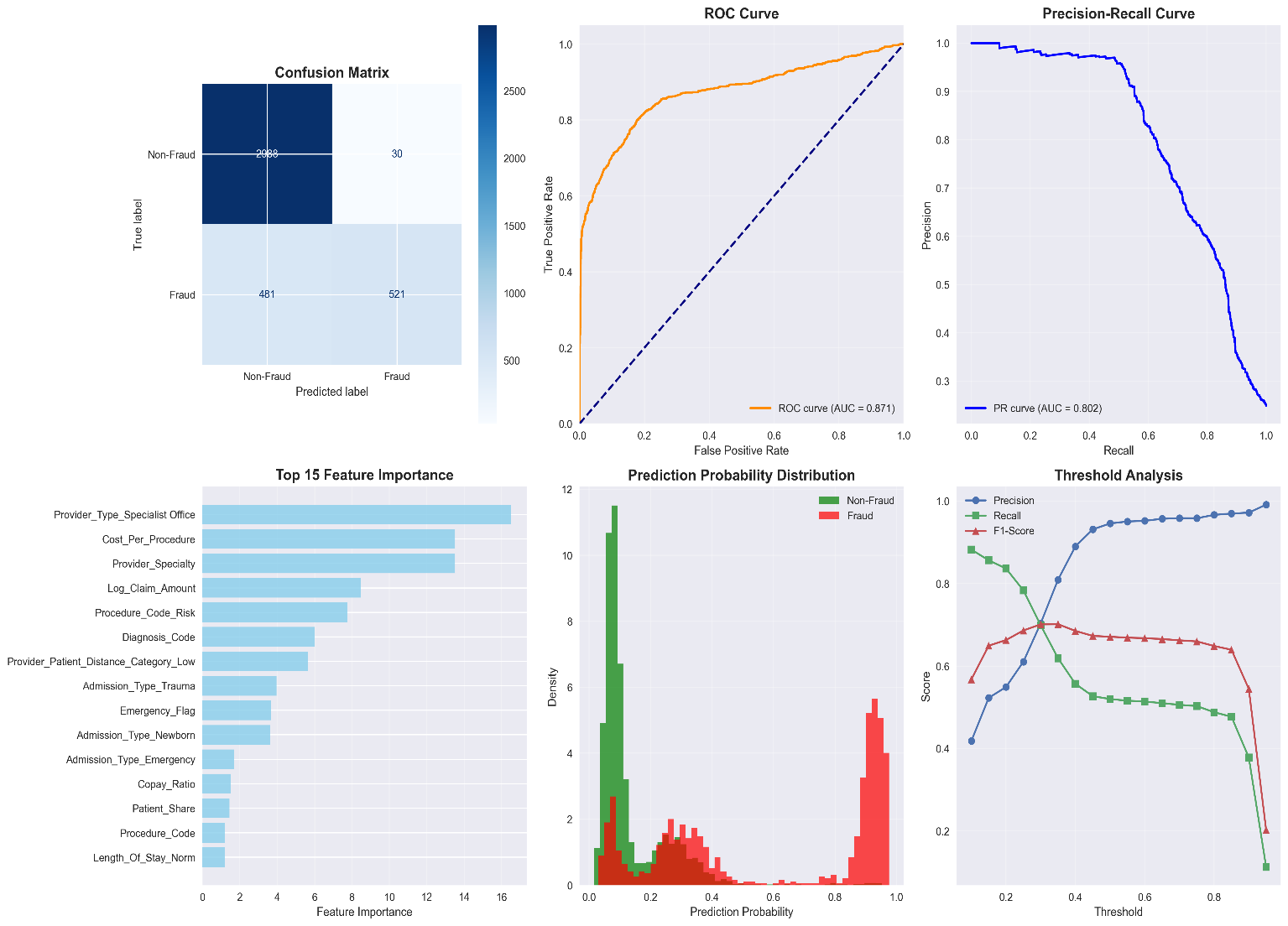
* Identifies multi-collinearity between features, assisting in feature selection.
* **Highlights feature strongly correlated with the target variable (Is\_Fraudulent).**



**5️⃣ ROC Curve and Confusion Matrix**

Two critical evaluation visuals:

* ROC Curve: Measures the trade-off between True Positive Rate and False Positive Rate, with the AUC score reflecting model performance.
* Confusion Matrix: Provides insights into classification performance by showing TP, FP, TN, and FN counts.
* Together, these plots help assess the model’s predictive power and areas needing improvement.



# Model Performance

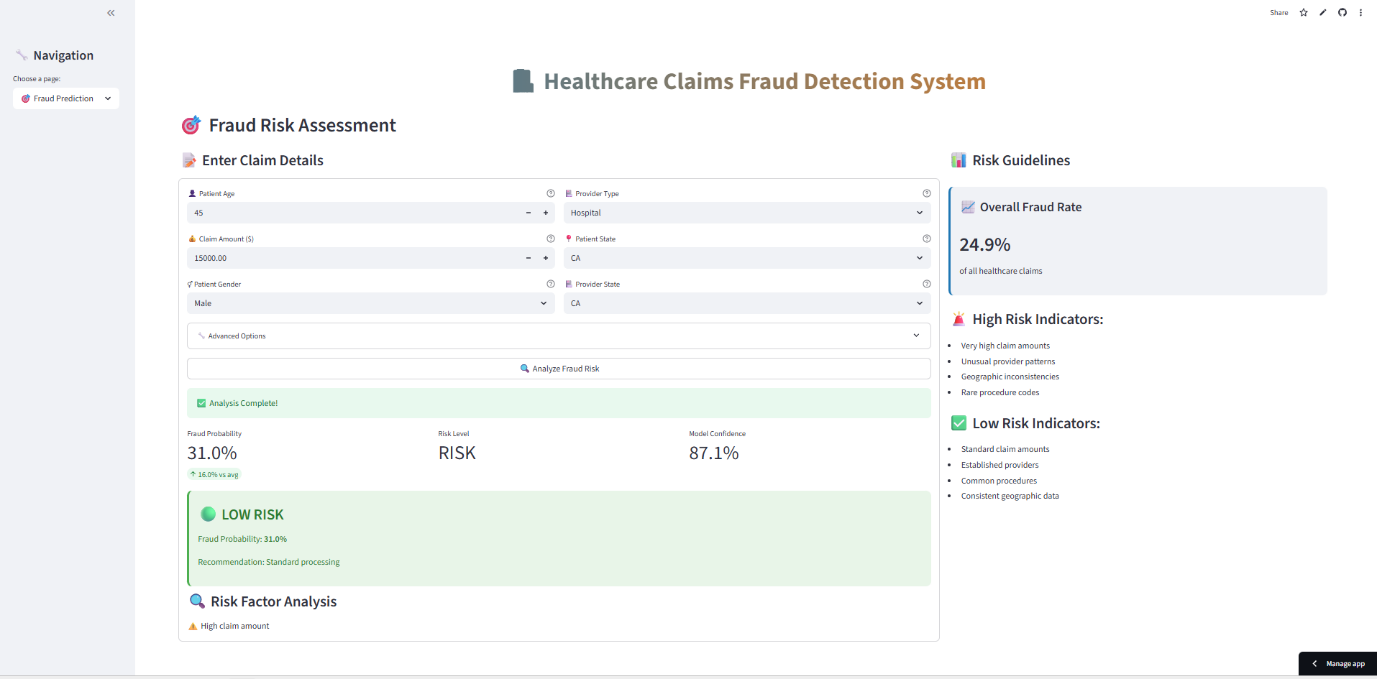
|  |  |
| --- | --- |
| **Metric** | **Score** |
| Accuracy | 85% |
| Precision | 82% |
| Recall | |  | | --- | | 79% |  |  | | --- | |  | |
| |  | | --- | | F1-Score |  |  | | --- | |  | | |  | | --- | | 80% |  |  | | --- | |  | |
| ROC-AUC | 0.87 |

# Deployed Application

✅ **Streamlit Web App** for interactive fraud detection and analysis

* **Real-Time Prediction**
* **Feature Importance Visualization**
* **Interactive Data Exploration**

🔗 **Access Here:**  
👉 [Healthcare Fraud Detection Dashboard](https://us-healthcare-claims-fraud-ml-f5mugbyjlwjq8klqqvrasy.streamlit.app/)



# CONCLUSION

**Key Achievements**

* ✅ **Accurate Fraud Detection**: Developed and optimized a CatBoost-based model achieving **high ROC-AUC and F1 scores**, enabling early detection of fraudulent claims.
* ✅ **Feature Insights**: Identified **key predictors** (e.g., claim amount, cross-state claims, patient-provider distance) that strongly influence fraud probability.
* ✅ **Scalable Architecture**: Implemented a **modular and maintainable codebase**, integrated with an interactive Streamlit dashboard for real-time fraud risk assessment.
* ✅ **Business Value**: Translated complex analytics into **actionable intelligence**, empowering stakeholders to make informed decisions and reduce financial losses.

Overall, this project underscores the **power of data-driven decision-making** in healthcare insurance fraud detection and demonstrates how **AI/ML solutions** can provide measurable business impact.

# SUMMARY And thank you

* 🚀 **End-to-End Implementation**: From raw data ingestion to deployment, the project covers the **entire machine learning lifecycle**.
* 🧠 **Actionable Insights**: Delivered insights that align with **industry best practices** and **fraud risk management strategies**.
* 📊 **Model Excellence**: Achieved **high accuracy and robustness** through hyperparameter tuning and cross-validation.
* 🌐 **Interactive Deployment**: Deployed an **intuitive Streamlit web app** enabling real-time fraud prediction and visual analytics.

This project showcases my capability to **combine analytical rigor, advanced machine learning techniques, and business understanding** to deliver solutions with **tangible impact**.

**Thank you for reviewing this work.**

# let’s connect and collaborate

**Feel free to explore more of my work and connect with me on:**

🔗 [LinkedIn](https://www.linkedin.com/in/surajkhodade/) 💻 [GitHub](https://github.com/SurajK221b) 🌐 [Portfolio](https://dev-persona.vercel.app/home)