# 🕵️‍♂️ Fraud Detection on Financial Transactions Using Machine Learning 🧠📊

Welcome to a complete exploratory data analysis (EDA) project aimed at uncovering fraud patterns from financial transaction data. This project works on different parameters of fraudulent users. A machine learning model is created to automatically predict fraud based on historical data.

# 📊 Data Analysis

## 🗂️ Dataset Overview

* 📁 **File**: SOI\_2025\_Dataset.csv
* 🎯 **Target column**: fraud\_bool (0 = Legit, 1 = Fraud)
* 📊 **Features analyzed**:
  + credit\_risk\_score, bank\_branch\_count\_8w, device\_distinct\_emails\_8w
  + foreign\_request, month, session\_length\_in\_minutes, payment\_type
  + prev\_address\_months\_count, current\_address\_months\_count
  + total\_relationship\_count, income

## 🔍 Key Insights & Findings

### 📈 Credit Risk Score

* 🚩 Fraudsters had **unexpectedly higher credit scores**, possibly indicating **synthetic identity fraud** or stolen identities.

### 🏦 Bank Branch Usage

* Fewer branches used by fraudsters — suggesting **online-only** or **temporary accounts**.

### 💻 Device–Email Mapping

* Devices linked to multiple emails → likely **account farming** or **bot usage**.

### 🌐 Foreign Requests

* foreign\_request = 1 showed higher fraud — useful for **geolocation profiling**.

### 🗓️ Monthly Trend

* Spike in **Month 7 (July)** — possibly seasonal or reporting-cycle-related fraud.

### ⌛ Session Length

* Fraudulent sessions were **short** — possible **automation** or **scripted behavior**.

### 💻 Device OS

* Fraud was higher on **Windows OS** — potentially due to botnet/tool accessibility.

### 🔌 Keep-Alive Sessions

* keep\_alive\_session = 0 frequent in frauds — indicating **non-persistent** behavior.

### 🏠 Address Stability

* Many fraudsters had **missing/low residence durations** — hard to verify identity.

### 💰 Income Trends

* **Low/missing income values** had slightly more frauds — indicating **fake profiles**.

### 🧾 Total Relationships

* Fewer relationships = newer/temporary/fake accounts used for fraud.

### 💳 Payment Types

* Prepaid/anonymous cards saw higher fraud rates.

## 📊 Visualizations Included

* Boxplots, bar charts and countplots showing fraud patterns across:
  + Month, OS, payment type, device type, foreign requests, etc.
* 📍 Embedded in: [data\_analysis.ipynb](./notebooks/01_data_analysis.ipynb)

# 🧱 Feature Engineering

### 🏠 Address Stability

* -1 replaced with column medians for prev\_address\_months\_count, current\_address\_months\_count

### 💳 Encoded Categorical Features

* Label-encoded: payment\_type, employment\_type, device\_os, etc.
* Saved mappings to mappings.json

### 🚀 Velocity Risk Features

* Transformed: velocity\_6h, velocity\_8h, velocity\_1w with log scaling
* Created velocity\_risk\_score using weighted score map

### ✏️ Data Standardization

* Features normalized or scaled

### 🧪 Train-Test Split

* Used train\_test\_split to prepare model datasets

📓 Code in: [feature\_Engineering.ipynb](./notebooks/02_feature_engineering.ipynb)

## 📁 Project Structure

fraud\_detect/  
├── data/  
│ └── SOI\_2025\_Dataset.csv  
├── notebooks/  
│ ├── 01\_data\_analysis.ipynb  
│ ├── 02\_feature\_engineering.ipynb  
│ ├── logistic\_model.ipynb  
│ ├── lightGBM.ipynb  
│ └── XG\_boost\_part2.ipynb  
├── mappings/  
│ ├── mappings.json  
│ └── weight\_map(2).json  
├── models/ # Saved models  
├── plots/  
├── requirements.txt  
├── README.md  
└── .gitignore

## ⚙️ Technologies Used

* 🐍 Python 3.x, JSON
* 📊 Pandas, NumPy, Scikit-learn, XGBoost, LightGBM
* 🎨 Matplotlib, Seaborn
* 🧠 Jupyter Notebook

# 🤖 Model Building & Evaluation

### 1️⃣ Logistic Regression

* sklearn.linear\_model.LogisticRegression
* Scaled numeric + encoded categorical features
* 📈 ROC Curve, Confusion Matrix, F1-score
* 📓 [logistic\_model.ipynb](./notebooks/logistic_model.ipynb)

### 2️⃣ XGBoost Classifier

* xgboost.XGBClassifier
* Tuned parameters, tree ensembles
* Feature importance plots
* 📓 [XG\_boost\_part2.ipynb](./notebooks/XG_boost_part2.ipynb)

### 3️⃣ LightGBM Classifier

* lightgbm.LGBMClassifier
* Faster gradient boosting
* 📓 [lightGBM.ipynb](./notebooks/lightGBM.ipynb)

## 🔧 Upcoming Improvements

* Cross-validation for robustness
* Better handling of imbalance via:
  + class\_weight='balanced'
  + SMOTE, undersampling
* Model ensembling to boost recall

## 🧠 Conclusion

This project explored fraudulent financial behaviors via EDA, feature engineering, and predictive modeling. It highlights the importance of precision, interpretability, and class balancing in real-world fraud detection. Further improvements can increase model generalization and deployment readiness.

## 👤 Author

Developed by: **Anubhav Goyal**  
🎓 B.Tech (Mathematics and Computing) @ IIT Dharwad  
📬 GitHub: [@Anj2307](https://github.com/Anj2307)  
📨 Email: [MC24BT005@iitdh.ac.in](mailto:MC24BT005@iitdh.ac.in)

📎 *This project is part of a learning journey in AI, Finance, and Machine Learning applied to fraud detection.*