🧠 Fraud Detection and Financial Risk Modeling

End-to-end fraud detection and financial risk modeling system (13M+ transactions) integrating EDA, feature engineering, LightGBM optimization, and explainable AI for business risk auditing.

Python 3.10+

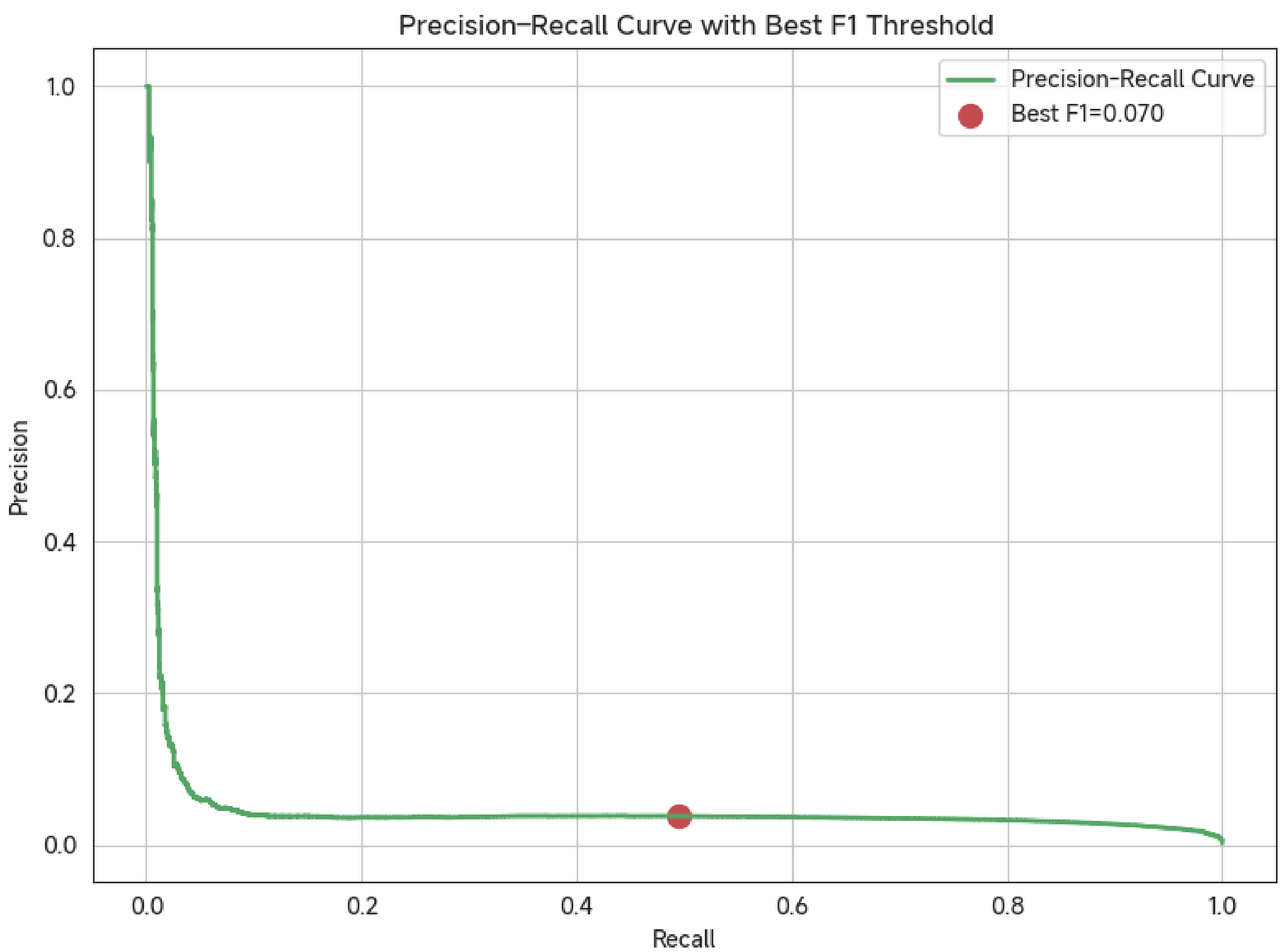
LightGBM 4.x

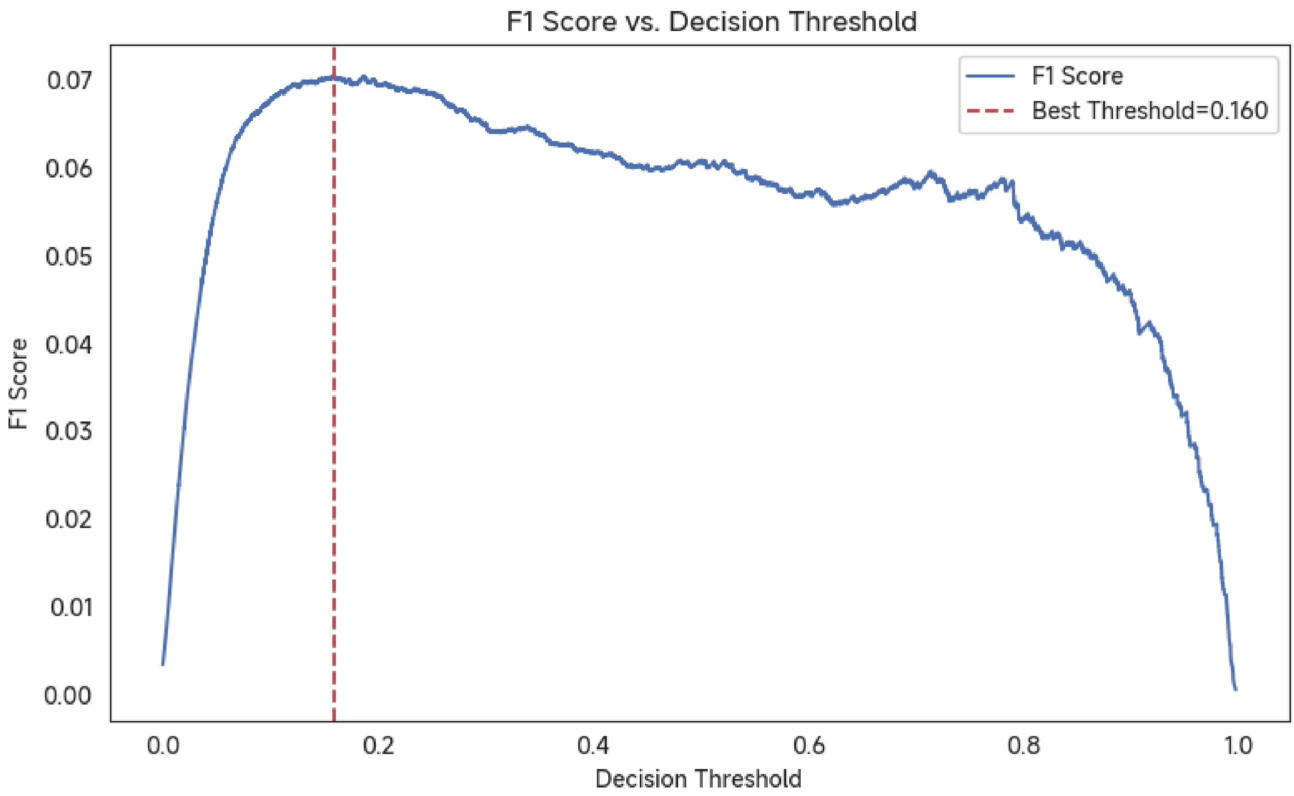
SHAP ExplainableAI

license MIT

📘 Overview

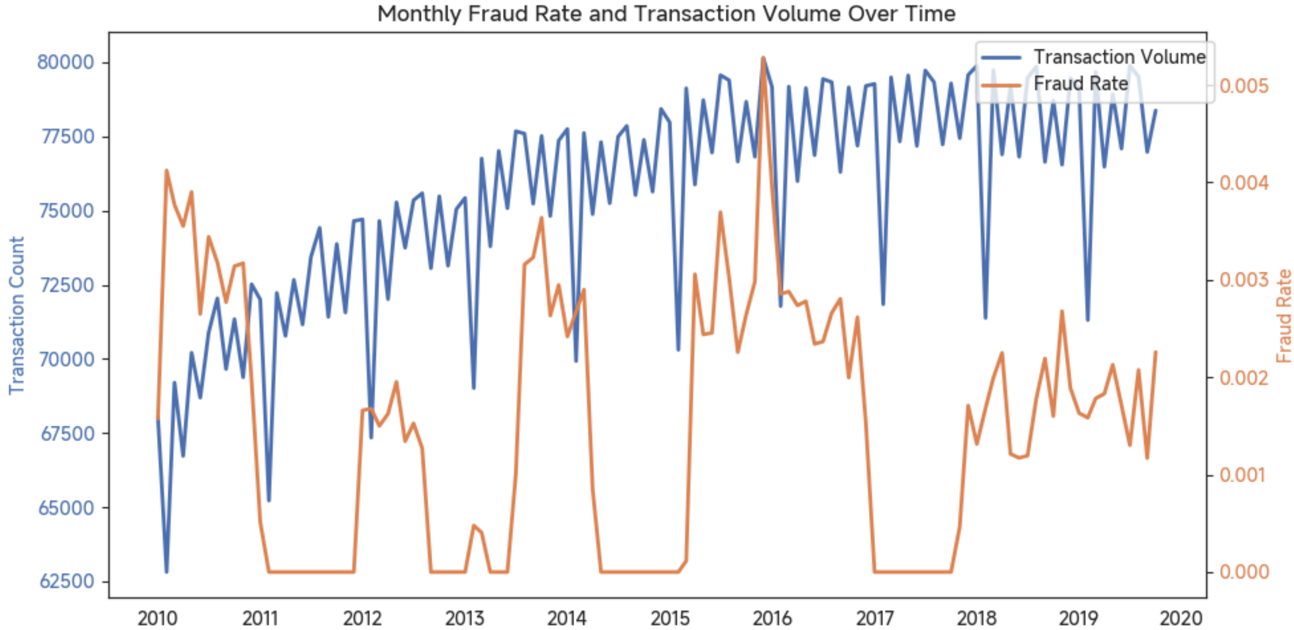
This project builds a **data-driven fraud detection pipeline** capable of identifying suspicious financial transactions from over **13 million records**.
It combines **exploratory data analysis**, **feature engineering**, **imbalanced learning**, and **interpretable machine learning** to deliver actionable business insights.



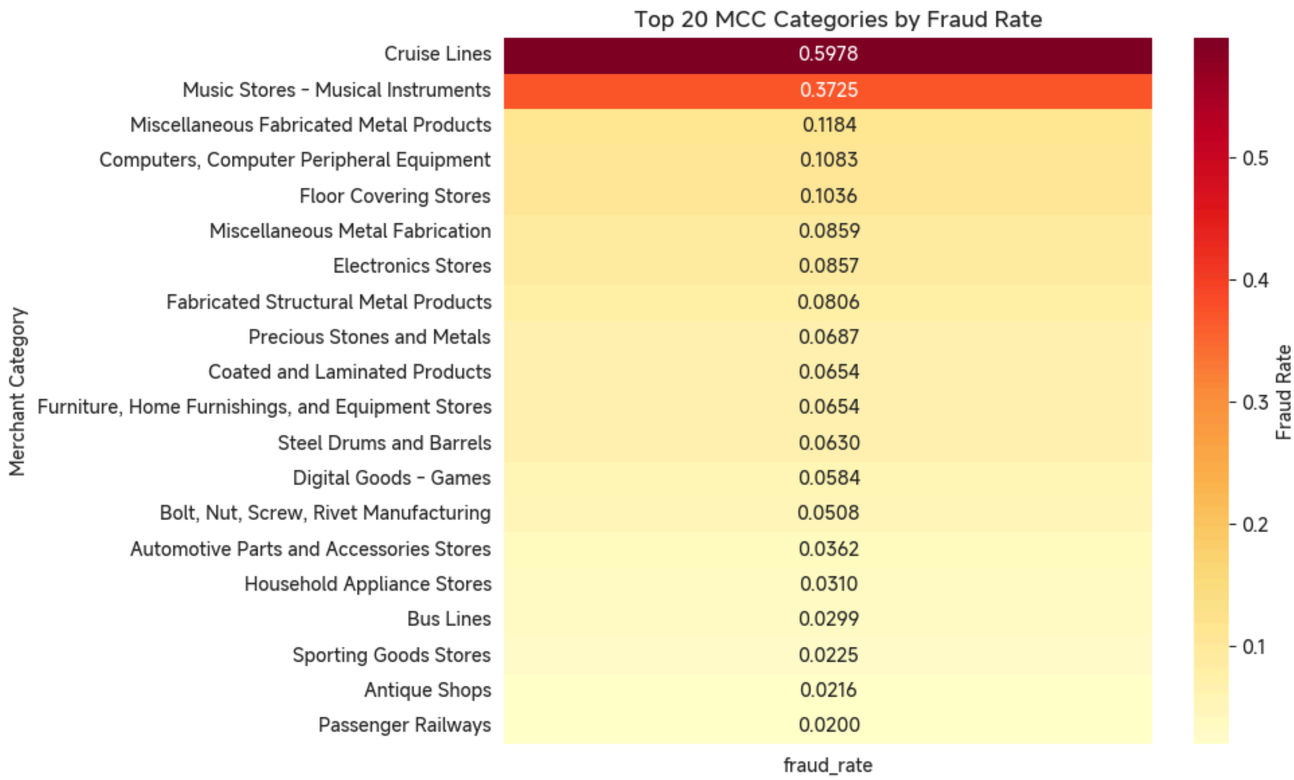


⚙️ Workflow Architecture

```
Data Loading → EDA → Feature Engineering → Class Balancing →
LightGBM Training → Threshold Optimization → SHAP Interpretability →
Risk Report
```



Monthly Fraud Rate and Transaction Volume Over Time
Transaction Volume
Fraud Rate
Transaction Count
80000
77500
75000
72500
70000
67500
65000
62500
0.005
0.004
0.003
0.002
0.001
0.000
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020



Top 20 MCC Categories by Fraud Rate
Merchant Category
Cruise Lines 0.5978
Music Stores - Musical Instruments 0.3725
Miscellaneous Fabricated Metal Products 0.1184
Computers, Computer Peripheral Equipment 0.1083
Floor Covering Stores 0.1036
Miscellaneous Metal Fabrication 0.0859
Electronics Stores 0.0857
Fabricated Structural Metal Products 0.0806
Precious Stones and Metals 0.0687
Coated and Laminated Products 0.0654
Furniture, Home Furnishings, and Equipment Stores 0.0654
Steel Drums and Barrels 0.0630
Digital Goods - Games 0.0584
Bolt, Nut, Screw, Rivet Manufacturing 0.0508
Automotive Parts and Accessories Stores 0.0362
Household Appliance Stores 0.0310
Bus Lines 0.0299
Sporting Goods Stores 0.0225
Antique Shops 0.0216
Passenger Railways 0.0200
fraud_rate
Fraud Rate
0.5
0.4
0.3
0.2
0.1

🧩 Core Steps

Stage	Description
Data Preparation	Cleaned and merged `transactions` , `users` , `cards` , and `MCC` data (13M+ rows).
EDA & Visualization	Distribution, outlier, and geographic/industry-level fraud analysis.
Feature Engineering	Constructed 20+ numerical, categorical, and time-based features (`amount_log` , `is_refund` , `hour` , `mcc_desc` , etc.).
Class Balancing	Undersampling strategy (10× ratio) to stabilize LightGBM training.
Model Training	Tuned LightGBM (GBDT) with early stopping and AUC optimization.
Explainability (SHAP)	Interpreted feature contributions globally and locally (fraud driver visualization).
Threshold Tuning	Optimized precision–recall trade-off with decision curve visualization.
Business Application	Exported Top-200 high-risk transactions for manual verification.

📊 Model Performance

Before & After Threshold Optimization

Metric	Default Threshold (0.5)	Tuned Threshold (0.1598)	Δ Improvement
AUC (Validation)	0.9718	0.9718	—
Precision	0.0374	0.0379	+0.0005
Recall	0.1601	**0.4941**	**+0.3340**
F1 Score	0.0606	**0.0704**	+0.0098
Detection Coverage	0.74%	**2.28%**	↑ Expanded audit scope

Interpretation:

- After tuning the decision threshold, Recall improved from **16.0% → 49.4%**, tripling fraud coverage while maintaining similar precision (~0.04).
- The optimized configuration balances model confidence and operational feasibility — ideal for financial audit pipelines where human verification capacity is limited.

🌍 Explainability

- **Global Importance:** `amount_log` , `hour` , `mcc_desc` , and `client_mean` drive fraud prediction.
- **Local Explanation:** SHAP force plots reveal how transaction timing and amount deviations trigger risk alerts.
- **Interpretability Goal:** bridge **model confidence** and **financial analyst reasoning**.

📦 Exported Deliverables

Output File	Description
`main.ipynb`	Full pipeline notebook with 19 modular cells
`main.html`	Rendered analysis report
`high_risk_transactions_top200.csv`	Top 200 suspicious transactions ranked by fraud probability
`.gitignore`	Excludes large data files (JSON/CSV over 100 MB)

🧰 Tech Stack

- **Python 3.10 + Pandas + NumPy**
- **LightGBM 4.x** (with callbacks for early stopping & AUC logging)
- **Matplotlib / Seaborn** (visual analytics)
- **SHAP 0.43.0** (explainable AI)
- **Scikit-learn 1.4+** (precision-recall & threshold tuning)

💡 Business Impact

This project demonstrates a **scalable, interpretable fraud detection framework** ready for integration into financial risk control systems.

📈 Author

Republic (GitHub: Republic1024) (AI & Data Science)

License

This project is released under the MIT License.