Fraudulent Account Prediction

Thomas Johnson

12/15/2025

Motivation

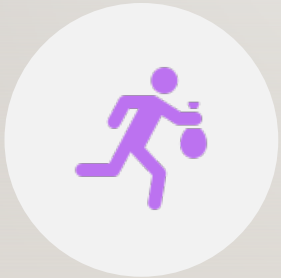
Currently employed at a large bank, building automations

I worked on a project called Security Closures, a process for closing accounts flagged for fraud

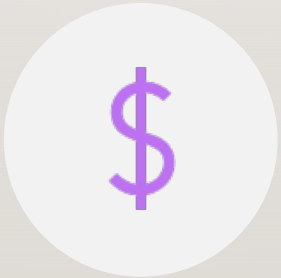
Security Closures is reactive, run only after fraud has been detected

With how fast the fraud at my bank is scaling, ML methods to identify fraud is not only interesting but useful.

BACKGROUND



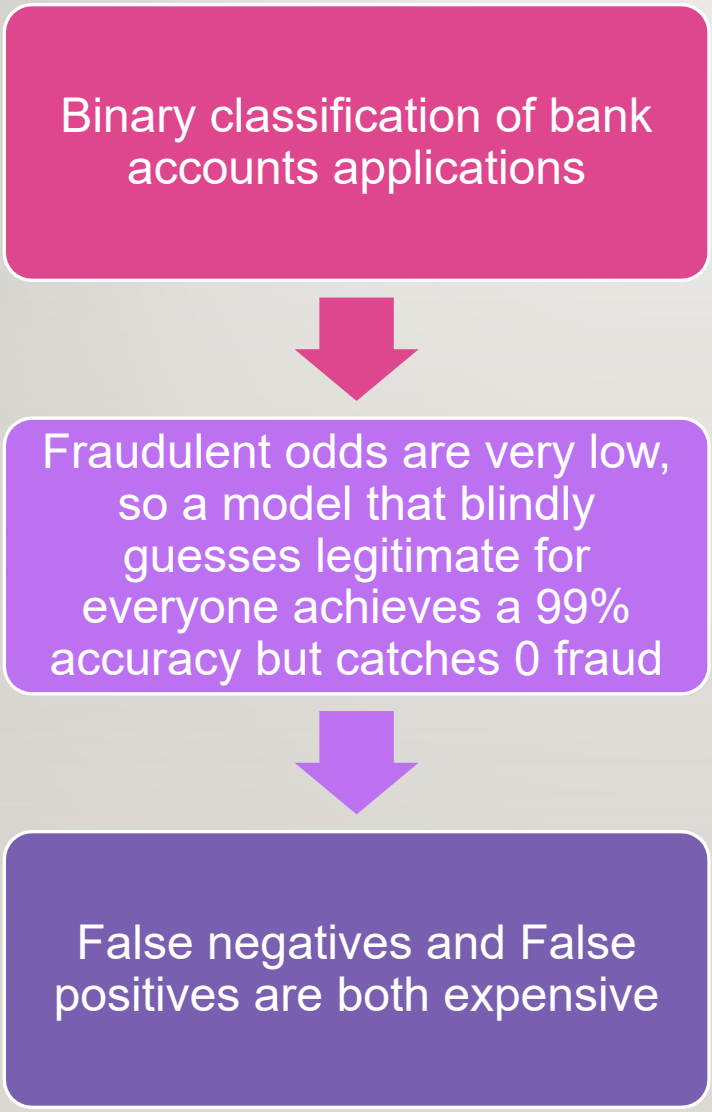
US CONSUMERS LOST $12 BILLIONS IN 2024 ALONE



TRADITIONAL SYSTEMS USE HARD RULES (INCOME < $5K)

WE NEED A SYSTEM THAT LEARNS COMPLEX, EVOLVING PATTERNS AS FRAUD INCREASINGLY BECOMES MORE ADVANCED

CLAIM / TARGET TASK



PROPOSED SOLUTION

Implement a Balanced Random Forest

Tune with Precision-recall curves and confusion matrix

Optimize threshold to balance fraud detection and review costs

DATA SUMMARY

- Use Base Application Fraud (BAF) Dataset
- 1,000,000 rows of applications and transactions
- Includes 32 Features like income, age, employment status, source, etc.
- Includes missing values and extremely skewed columns

IMPLEMENTATION

Start by	Start by pre-processing data to handle skewed numeric data and encode categorical data
Keep	Keep all fraud cases but randomly down sample legitimate cases
Run	Run Random Forest with balanced subsampling and optimize hyperparameters
Report on	Report on data with PR-AUC , Confusion Matrix, SHAP

EXPERIMENTAL RESULTS

Achieved PR AUC score of .15

Able to successfully identify differing amounts of fraud correctly with differing amounts of classifying fraud and false fraudulent flags.

Experimental Analysis

- Used Confusion Matrices to identify fraud rates tradeoff
- Used SHAP to find most important features

	feature	mean_abs_shap
39	housing_status_BA	0.084500
51	device_os_windows	0.058582
9	name_email_similarity	0.040364
21	keep_alive_session	0.039151
23	phone_home_valid	0.037192
11	current_address_months_count	0.034061

CONCLUSION AND FUTURE WORK

SUCCESSFULLY BUILT ML PIPELINE

FRAUD CATCHING CAN BE HEAVILY OPTIMIZED

USING XGBOOST OR A GRAPH NEURAL NETWORK (GNN).