IntelliFraud: Bank Account Fraud Detection using Machine Learning

Nukta Bhatia | Emir Talundzic | Piyush Shivrain
Ashish Puri | Jagannath Banerjee | Peter Kovari
CSE 6242 Data & Visual Analytics
Fall 2023

Georgia Tech

Objectives

Motivation

- Build a tool that detects fraudulent bank accounts opened through pnline applications in a consumer bank.
- Fraudulant account costs sustained by the bank as tracing back individuals is difficult, time consuming & expensive.
- Fraud accounts is class imbalance

Current Practice

- Transactional Monitoring
- Rule Based systems
- Behavioral Analysis
- Anomaly Detection
- Standard Machine Learning (ML) techniques

Goal

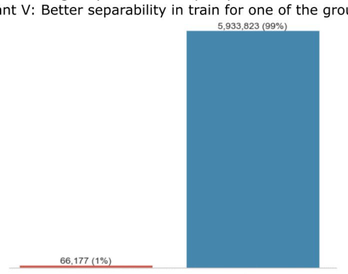
- Create a graph network to detect fraud tarnsaction flow
- Enable bank analysts to visualize trends and patterns in bank applications data
- Enable analysts to input application parameters
- Use ensemble methods like voting and stacking classifiers, comparing with LightGBM, XGBoost & AdaBoost
- Allow analysts to to compare & select ML models
- Provide insights into performance metrics to aid informed model selection

Dataset

- Bank Account Fraud Dataset Suite (NeurIPS2022) [Kaggle]
- 6 synthetic Variants
- Realistic, robust test bed based on a present-day, real-world dataset for fraud detection
- Each dataset has distinct controlled types of bias
- Extremely low prevalence of positive (fraud) class
- Privacy techniques (noise addition), feature encoding added

Variants

- Base: Samped to represent original dataset
- Variant I: Higher groupsize disparity than base
- Variant II: Higher prevalence disparity than base
- Variant III: Better separability for one of the groups
- Variant IV: Higher prevalence disparity in train
- Variant V: Better separability in train for one of the groups



Fraud vs Non-Fraud Count (All Variants)

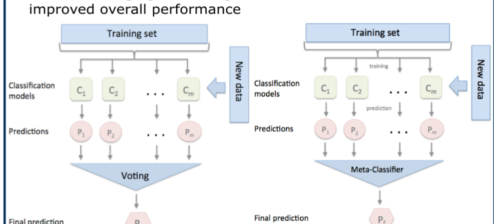
Sampling Methods

Imbalanced Datasets

- Employed 1:1, 1:2, and 1:3 random sampling
- To emphasize learning from the minority class

Models

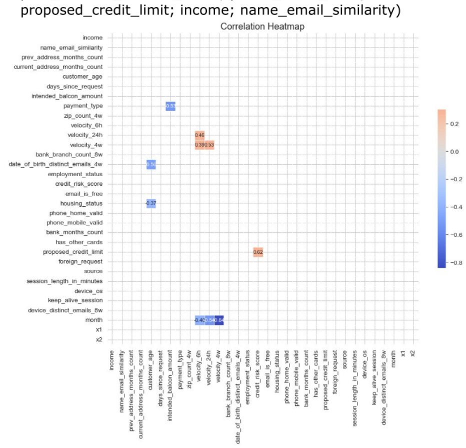
- **LGBM Classifier**
  Employs bossting to efficiently train decision trees, optimizing for speed and accuracy
- **AdaBoost Classifier**
  Combines weak learners to create a robust model, assigning more weight to misclassified instances for improved accuracy
- **XGB Classifier**
  Enhances predictive performance by sequentially refining weak learners and minimizing errors
- **Voting Classifier**
  Integrates predictions from multiple algorithms to make collective decisions, enhancing model accuracy through voting mechanism
- **Stacking Classifier**
  Integrates diverse model predictions by training a meta model, leveraging the strengths of individual models for improved overall performance

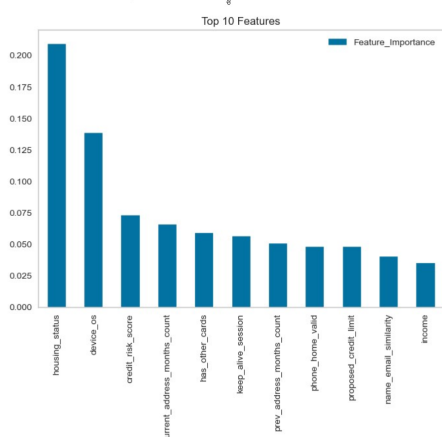


Working of Voting & Stacking Classifiers

Feature Selection

- **Variance Threshold**
  Removes low variance feature [device_fraud_count]
- **Pearson's Correlation Matrix**
  Remove highly correlated features [velocity_4w]
- **Random Forest Classifier**
  Using Feature Importance, picks the most important features (housing_status; device_os; credit_risk_score; has_other_cards; current_address_months_count; keep_alive_session; prev_address_months_count; phone_home_valid; proposed_credit_limit; income; name_email_similarity)

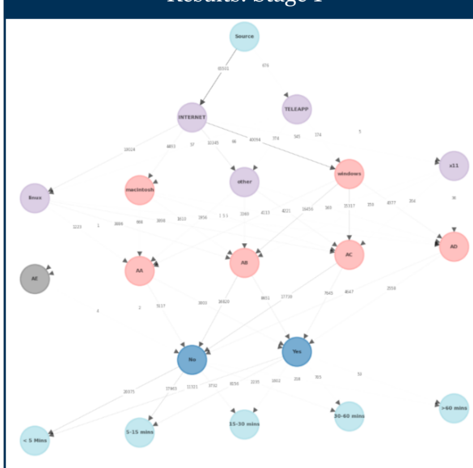




Implementation

- **Dataset Partition:**
  80% Train; Stratified 5-Fold Cross Validation; 20% Test
- **Tournament-style Procedure**
  # Stage 1: Graph network visualization to see fraudulent transaction flow through various subsystems
  # Stage 2: Train variants of the models using sampling strategies
  # Stage 3: Analyze and provide best performing model
- **Performance Metric**
  AUC-ROC & F1 Score due to imbalanced dataset

Results: Stage 1



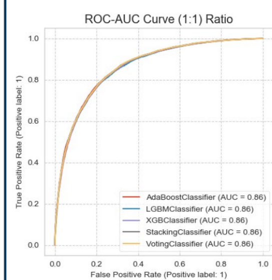
Results: Stage 2

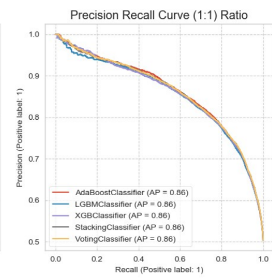
Model Comparison for this Dataset on Test Set

Sample Class (Fraud: Non-Fraud)	Classifier	5 Fold CV Score	Precision	Recall (Fraud)	ROC_AUC_Scr	Accuracy	F1 Score
1:01	XGBClassifier	0.783	0.786	0.785	0.784	0.784	0.785
1:01	AdaBoostClassifier	0.785	0.792	0.776	0.785	0.784	0.784
1:01	LGBMClassifier	0.782	0.782	0.783	0.781	0.781	0.783
1:01	VotingClassifier	0.787	0.789	0.784	0.785	0.785	0.786
1:01	StackingClassifier	0.787	0.789	0.784	0.785	0.785	0.786
1:02	XGBClassifier	0.797	0.722	0.649	0.763	0.801	0.684
1:02	AdaBoostClassifier	0.797	0.744	0.622	0.758	0.804	0.678
1:02	LGBMClassifier	0.794	0.741	0.613	0.753	0.801	0.671
1:02	VotingClassifier	0.799	0.733	0.636	0.761	0.803	0.681
1:02	StackingClassifier	0.799	0.732	0.637	0.761	0.803	0.681
1:03	XGBClassifier	0.82	0.685	0.542	0.73	0.825	0.605
1:03	AdaBoostClassifier	0.819	0.718	0.515	0.724	0.83	0.6
1:03	LGBMClassifier	0.816	0.723	0.496	0.717	0.828	0.588
1:03	VotingClassifier	0.821	0.71	0.527	0.728	0.829	0.605
1:03	StackingClassifier	0.821	0.708	0.532	0.73	0.83	0.608

Results: Stage 3

Best Model Performance for this Dataset on Test Set





User Interface

Conclusion

- Voting and Stacking Classifiers performed slightly better than LightGBM, XGBoost & AdaBoost on this dataset
- The network graph helps analyze fraud transaction flow
- The EDA page provides the bank analysts to visualize trends and partterns in bank applications data
- Bank analysts can compare and select the ML Models based on the performance parameters

Future Work

- Connect the IntelliFraud application to real dataset and test the performance of voting & stacking classifiers against that of LightGBM, XGBoost & AdaBoost
- Extend SHAP explainability features by adding LIME(Local Interpretable Model-Agnostic Explanations) allowing users to compare the features
- Provide a feedback loop to aid model improvement

- Thank you to the CSE 6242 Data Analytics and Visualization Prof Polo, and TA staff