Fraud Detection Using Machine Learning (XGBoost)

27.07.2025

Juniyad Tamboli

Kapurhol, Pune.

Overview

This project implements a machine learning pipeline to detect fraudulent transactions in a financial dataset. Utilizing a robust gradient boosting classifier (XGBoost), the solution preprocesses transactional data, handles class imbalance, and applies hyperparameter tuning for optimal performance. The system achieves high accuracy and AUC scores, demonstrating its effectiveness in distinguishing legitimate versus fraudulent transactions.

Goals

1. Build a predictive model to accurately identify fraudulent transactions in a customer transaction dataset.
2. Handle highly imbalanced class distributions using SMOTE to improve recall on minority fraud class.
3. Explore and tune XGBoost hyperparameters for best ROC-AUC performance.
4. Identify key features contributing to fraud detection for interpretability.
5. Provide an end-to-end reproducible workflow suitable for deployment or further research.

Specifications

Dataset

- File: fraud_dataset.csv

- Samples: (10,000,10)initially
- Features:

- *Categorical (encoded):*
- merchant_category
- customer_location
- device_type
- *Numerical (scaled):*
- previous_transactions
- customer_age
- amount
- Dropped Columns: transaction_id, customer_id, timestamp

Data Processing

- Label Encoding applied to all categorical features
- StandardScaler normalization applied to numerical features
- SMOTE oversampling used to balance classes when fraud rate < 15%

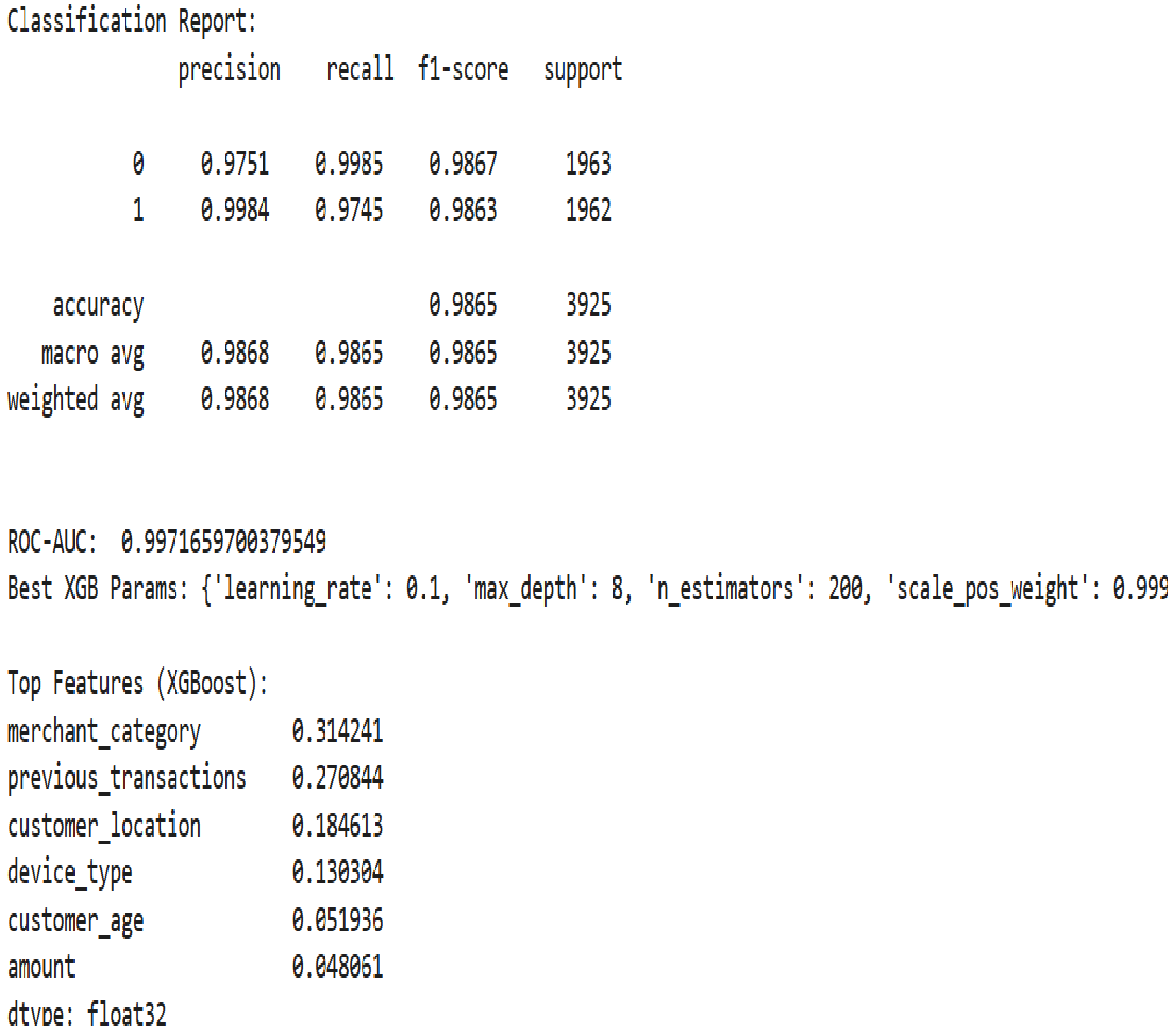
Model & Training

- Algorithm: XGBoost Classifier
- Hyperparameters tuned via GridSearchCV:
- n_estimators: [150,200]- max_depth: [4, 6,8] learning_rate: [0.03, 0.07, 0.1]
- Cross-validation: Stratified 3-fold

Evaluation

- Metrics:
- Confusion Matrix
- Precision, Recall, F1-Score (classification report)

- ROC-AUC Score



```
Classification Report:
              precision    recall  f1-score   support

           0     0.9751    0.9985    0.9867      1963
           1     0.9984    0.9745    0.9863      1962

    accuracy                         0.9865      3925
   macro avg     0.9868    0.9865    0.9865      3925
weighted avg     0.9868    0.9865    0.9865      3925


ROC-AUC:  0.9971659700379549
Best XGB Params: {'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 200, 'scale_pos_weight': 0.999

Top Features (XGBoost):
merchant_category       0.314241
previous_transactions   0.270844
customer_location       0.184613
device_type             0.130304
customer_age            0.051936
amount                  0.048061
dtype: float32
```

Milestones

I. Data Acquisition and Inspection

- Successfully loaded the fraud_dataset.csv comprising around 4,900 transaction records..

II. Data Cleaning and Preprocessing

- Dropped non-informative columns: transaction_id, customer_id, and timestamp.
- Encoded categorical variables: merchant_category, customer_location, and device_type.
- Scaled numerical features: previous_transactions, customer_age, and amount.

III. Class Imbalance Handling

- Determined fraud class ratio was less than 15%; applied SMOTE to balance classes.

IV. Partitioning Data

- Performed stratified 80/20 train-test split to maintain class distribution.

V. Model Training and Optimization

Built an XGBoost classifier and optimized hyperparameters using GridSearchCV with stratified 3-fold cross-validation.

Discovered best parameters:

- learning_rate: 0.1
- max_depth: 8
- n_estimators: 200

VI. Feature Importance Analysis

- Top contributing features identified:

1. Merchant_category
2. Previous_transactions
3. Customer_location
4. Device_type
5. Customer_age
6. Amount

VI. Conclusion

Hence we have successfully implemented the fraud detection using XGBOOST