# Bank Fraud Detection Analysis

ML project for detecting fraudulent bank transactions using risk-based scoring and behavioral analysis.

**Author** Rajeev Bhat rajeevmbhat@gmail.com

#### Executive summary

This project develops a machine learning-based fraud detection system for banking transactions using a risk-scoring approach combined with logistic regression. Analyzing 2,512 transactions from a Kaggle dataset, we created a multi-factor risk assessment system that identifies fraudulent patterns with 100% precision and 58.33% recall on test data. The system flags 1.27% of transactions as high-risk fraud cases, aligning with industry standards, and provides interpretable results for fraud prevention teams.

#### Rationale

Financial fraud costs the banking industry billions of dollars annually and erodes customer trust. Traditional rule-based fraud detection systems often generate high false positive rates, causing customer friction and operational inefficiency. Machine learning approaches can identify complex patterns in transaction data to improve fraud detection accuracy while reducing false alarms. This project addresses the critical need for automated, intelligent fraud detection that balances security with customer experience.

#### Research Question

How can we effectively identify fraudulent bank transactions using machine learning while minimizing false positives? Specifically: - What transaction patterns and customer behaviors are most indicative of fraud? - Can we build a risk-scoring system that reliably flags suspicious transactions? - What machine learning approach provides the best balance of precision and recall for fraud detection?

#### Data Sources

**Source**: [Kaggle Bank Transaction Dataset](https://www.kaggle.com/datasets/valakhorasani/bank-transaction-dataset-for-fraud-detection/code) - **Size**: ~105 KB CSV format - **Quality**: 10.0 usability score on Kaggle - **Records**: 2,512 transactions with 16 features

### Features Include:

* **Transaction data**: amounts, types, duration, timestamps
* **Customer info**: age, occupation, account balance
* **Technical details**: device IDs, IP addresses, locations
* **Behavioral data**: login attempts, transaction timing

#### Methodology

**1. Data Quality Assessment & Preprocessing** - Missing value detection and handling - Duplicate removal and data type validation - Exploratory data analysis with statistical summaries and visualizations

**2. Feature Engineering** - Created temporal features (weekends, time patterns) - Developed behavioral indicators (transaction patterns, age groups) - Built binary risk flags for suspicious activities - Applied outlier detection using IQR and Z-score methods

**3. Risk Scoring System** Multi-factor fraud risk assessment based on: - High transaction amounts (top 5%) - Multiple login attempts (>1) - Unusual timing patterns (late night/early morning) - Rapid successive transactions (<30 minutes) - High amount-to-balance ratios (>95th percentile)

**4. Machine Learning Model Development** - Binary target creation (is\_fraud) based on risk scores - Logistic regression with balanced class weights to handle imbalanced data - Train/test split with stratification to maintain fraud distribution - Feature scaling with StandardScaler for numerical variables - Model evaluation using precision, recall, F1-score, and ROC-AUC metrics

#### Results

**Key Patterns Discovered:** - **Fraud prevalence**: 1.27% of transactions flagged as high-risk, aligning with industry standards - **Temporal patterns**: Strong concentration in specific hours (52% of fraudulent transactions at 4pm) - **Network analysis**: 609 devices shared across accounts, 551 IPs from multiple locations indicating potential fraud rings - **Amount patterns**: High-value transactions strongly correlated with risk scores (r=0.552)

**Model Performance - Logistic Regression:** - **Precision**: 100% - Every flagged transaction was actually fraudulent (zero false positives) - **Recall**: 58.33% - Successfully identified 7 out of 12 actual fraud cases - **F1-Score**: 73.68% - Balanced measure showing good overall performance - **ROC-AUC**: 79.17% - Strong ability to distinguish between fraud and legitimate transactions

**Features Used** (11 core features): - TransactionAmount, CustomerAge, LoginAttempts, AccountBalance - TransactionType, Channel, CustomerOccupation  
- IsWeekend, IsHighAmount, MultipleLogins, amount\_to\_balance\_ratio

**Why Precision Matters Most:** In fraud detection, precision is critical because false positives (flagging legitimate transactions as fraud) cause customer dissatisfaction, block valid purchases, and require costly manual review. Our 100% precision means zero false alarms, maintaining customer trust while effectively identifying fraudulent activity.

#### Next steps

**Short-term improvements (1-3 months):** 1. **Advanced Model Development**: Implement Random Forest and XGBoost models for comparison with logistic regression baseline 2. **Feature Enhancement**: Add more sophisticated temporal features (rolling averages, seasonal patterns) and network analysis features 3. **Cross-validation**: Implement k-fold cross-validation for more robust model evaluation 4. **Hyperparameter Tuning**: Use grid search or Bayesian optimization to optimize model parameters

**Medium-term extensions (3-6 months):** 1. **Regression Model**: Develop models to predict potential financial loss amounts for fraudulent transactions 2. **Time Series Analysis**: Build forecasting models to predict fraud frequency and seasonal patterns 3. **Real-time Scoring**: Implement streaming data pipeline for real-time fraud detection 4. **Ensemble Methods**: Combine multiple models for improved performance and robustness

**Long-term goals (6+ months):** 1. **Interactive Dashboard**: Create Streamlit app for real-time monitoring and fraud investigation 2. **Deep Learning**: Explore neural networks with embeddings for high-cardinality features 3. **Explainable AI**: Implement SHAP values and LIME for model interpretability 4. **Production Deployment**: Build scalable API for integration with banking systems

#### Outline of project

* [Main Analysis Notebook](fraud_detection.ipynb) - Complete EDA, feature engineering, and model development (21 cells)
* [Data Quality Assessment Module](data_quality_assessment.py) - Utilities for data cleaning and validation
* [Trained Model](fraud_detection_model.pkl) - Serialized logistic regression model for deployment

## Getting Started

1. **Install dependencies:**

* pip install -r requirements.txt

1. **Run analysis:**

* jupyter notebook fraud\_detection.ipynb

1. **Use data quality tools:**

* from data\_quality\_assessment import DataQualityAssessment  
  dqa = DataQualityAssessment(df)  
  clean\_df = dqa.assess\_and\_clean()

#### Contact and Further Information

**Author**: Rajeev Bhat  
**Email**: rajeevmbhat@gmail.com  
**Project**: Berkeley Haas Capstone - Bank Fraud Detection

For questions about the methodology, results, or potential collaborations, please reach out via email.

## Technical Details

### Dependencies

pandas>=1.3.0  
numpy>=1.21.0  
matplotlib>=3.4.0  
seaborn>=0.11.0  
scikit-learn>=1.0.0  
scipy>=1.7.0

### Project Files

├── data/  
│ └── bank\_transactions\_data\_2.csv # Transaction dataset  
├── fraud\_detection.ipynb # Main analysis notebook (21 cells)  
├── data\_quality\_assessment.py # Data cleaning utilities  
├── fraud\_detection\_model.pkl # Trained logistic regression model  
├── requirements.txt # Python dependencies  
└── README.md

Note: README.md final version courtesy of Claude.