**Project Deliverable 4: Project Recap and Lessons Learned**

**Financial Fraud Detection: Patterns, Risks, and Predictive Modeling**

**AIT-664-DL3: Information Processing, Representation, and Visualization**  
**George Mason University**  
**Dr. Charles Lynch**  
**April 27, 2025**

**Team Members:**

* Seshasai Venkat Sampath Kiran Sanku (G01481080)
* Pramath Rajprasad Rao (G01483865)
* Charan Chalasani (G01470867)

**1. Restatement of Initial Hypothesis**

**Initial Research Questions:**

Our project focused on addressing the following key research questions:

* Are certain merchant categories more targeted by fraudsters?
* How do transactional features (amount, velocity, error types) influence fraud detection?
* Do demographic factors like age, account age, or card type correlate with fraud risk?
* What role does time-based analysis (seasonality, account tenure) play in detecting fraud trends?
* Which machine learning models best detect fraudulent transactions?
* Can customer behavior and spending patterns predict future fraudulent activities?

**Restatement of Initial Hypothesis:**

Our hypothesis was that **fraudulent financial transactions are not random events** but exhibit **distinct and measurable patterns** based on transactional behavior, customer demographics, merchant types, and time-based factors. Specifically, we posited that features such as unusually high or inconsistent transaction amounts, frequent online transaction activity, account age nearing expiration, high credit utilization, and rapid transaction bursts would serve as key predictors of fraud risk. Furthermore, we hypothesized that fraudsters would selectively target certain merchant categories and exploit vulnerabilities in card features such as non-chip transactions. By deeply analyzing these behavioral and demographic patterns, we aimed to design machine learning models capable of accurately detecting and predicting fraudulent activities, thereby improving the overall financial security ecosystem and minimizing financial losses for institutions and customers alike.

**2. Summary of Data Acquisition**

We sourced our dataset from Kaggle, ["Financial Transactions Dataset: Analytics"](https://www.kaggle.com/datasets/computingvictor/transactions-fraud-datasets/data). The dataset contained structured information including transaction amounts, timestamps, merchant category codes, fraud labels, card details, and customer demographics. Initial review revealed:

* Good coverage across transaction types, encompassing online, chip-based, and swipe transactions across diverse merchant categories.
* Presence of missing values in critical fields like merchant names, cities, income levels, and credit scores, which necessitated careful handling through imputation and cleaning techniques.
* Significant class imbalance with very few fraudulent transactions relative to non-fraudulent ones, creating challenges for model training and necessitating oversampling techniques like SMOTE to ensure model fairness.
* Presence of a broad time span in transaction records, allowing us to analyze temporal fraud patterns over years.
* Rich set of customer demographic variables (age, income, credit score) enabling deeper behavioral analysis of fraud trends.
* Variations in transaction amounts and spending behaviors across merchants and customer profiles, adding complexity and richness to feature engineering efforts.

**3. Data Preparation & Information Modeling**

* **Data Cleaning**:
  + Removed duplicates based on unique transaction identifiers.
  + Handled missing values in fields like income, merchant city, and credit score using mean, median, or mode imputation strategies.
  + Standardized numerical columns like transaction amounts and monetary fields.
  + Converted fields such as "Per Capita Income," "Yearly Income," and "Total Debt" into floating point numbers to ensure numerical consistency.
  + Transformed "Has Chip" column from categorical ('Yes'/'No') into binary (1/0) format for modeling purposes.
  + Converted transaction timestamps into datetime format to enable time-based feature engineering.
* **Feature Engineering**:
  + Created new features such as transaction\_velocity (time difference between consecutive transactions) to capture rapid transaction bursts often indicative of fraud, and merchant\_risk\_score (based on merchant fraud history) to identify high-risk merchant categories. Additionally, calculated spending deviation metrics to highlight unusual customer behavior patterns compared to their historical trends.
  + Extracted features like account tenure and card expiration proximity to understand customer loyalty and vulnerability to fraud based on account/card age.
  + Added a flag Age\_Retirement\_Issue to indicate inconsistencies where a customer's current age exceeded their retirement age, helping spot potential data anomalies or suspicious profiles.
  + Created outlier flags (income\_outlier, debt\_outlier, per\_capita\_outlier) using the IQR method for yearly income, total debt, and per capita income, to capture unusual financial profiles often associated with fraud.
  + Computed card\_age\_years from account opening date to measure the longevity of cards, and flagged card\_expired cards to assess risk from outdated cards.
  + Engineered individual error flags (e.g., error\_insufficient\_balance, error\_bad\_pin, error\_bad\_cvv, etc.) from the transaction error descriptions to precisely model types of transaction failures linked to fraud attempts.
  + Created a negative\_transaction flag to capture transactions where the amount was negative, an unusual pattern that may indicate fraudulent refund or reversal tactics.
  + Normalized use\_chip column and derived transaction\_type categorizing each transaction as chip, swipe, or online, thus enabling more granular modeling of transaction security levels.
  + Calculated debt\_to\_income\_ratio to assess financial strain and risk levels for customers.
  + Developed credit\_utilization as a ratio of transaction amount to credit limit, a key indicator of potential over-leverage and fraud risk.

Each of these engineered features added new dimensions to our models, enhancing predictive power and allowing deeper exploration into fraud-related behaviors.

* **SMOTE**: We addressed class imbalance by splitting the data using train\_test\_split with stratification to preserve fraud ratios. We dropped datetime and high-cardinality columns to avoid noise, applied one-hot encoding to categorical variables, reindexed validation and test sets for consistency, and handled infinite/missing values using SimpleImputer. After cleaning, we applied SMOTE to synthetically oversample fraud cases, significantly improving model sensitivity and recall by preventing bias toward non-fraudulent transactions. This careful preprocessing ensured that our machine learning models received high-quality, consistent input data. It also helped stabilize model training and evaluation across different subsets of the data.
* **Outlier Detection**: Used **Isolation Forest** to detect extreme outlier transactions indicative of potential fraud. Isolation Forest works by randomly partitioning data points and identifying those that require fewer splits to isolate, flagging them as anomalies. Applying this method helped highlight suspicious transactions and customer profiles, improving our ability to detect rare fraudulent patterns without making assumptions about data distribution, which was crucial given the dataset's complexity.

**Exploratory Data Analysis (EDA):**

Feature Importance Plot: Visualized the top 15 features influencing fraud prediction using Random Forest. Interpretation: Features like "use\_chip" and "transaction\_type" were highly significant, showing that transaction method heavily impacts fraud risk. This revealed that non-chip transactions are inherently riskier. It also emphasized the importance of understanding transaction methods while designing fraud detection systems.

Class Distribution Before and After SMOTE: Displayed the imbalance and subsequent balancing of fraud and non-fraud cases. Interpretation: Initially, fraudulent transactions were heavily outnumbered by non-fraudulent ones, leading to poor model learning. Applying SMOTE corrected this imbalance, ensuring better model sensitivity and fairness across both classes.

Top 15 Merchant Categories for Fraudulent Transactions: Identified merchant types like department stores and money transfer services as high-risk. Interpretation: These categories had a disproportionately high number of fraudulent activities. It indicated that fraudsters deliberately target high-volume or easily resellable goods, reinforcing the need for sector-specific fraud monitoring.

Fraudulent Transactions by Account Opening Year: Showed older accounts (especially opened around 2005-2010) faced higher fraud. Interpretation: Accounts with long histories may have relaxed monitoring, making them prime targets. Fraudsters could also perceive older accounts as more "trusted," leading to reduced suspicion during transactions.

Fraudulent Transactions by Credit Card Expiration Year: Indicated peaks around cards expiring in 2020 and 2022. Interpretation: Cards near expiration were more frequently compromised, suggesting that fraudsters may exploit less actively monitored cards nearing their end of life. This finding supported targeted interventions around card renewal periods.

Fraudulent Transactions by Number of Credit Cards: Found fraud risk peaks among users with 3-5 cards. Interpretation: Managing multiple cards may increase exposure to security lapses. It also highlighted that customers with moderate but manageable numbers of cards faced higher risks compared to those with only one or two cards.

Fraudulent vs. Non-Fraudulent Transactions by Age Group: Older groups (40–60, 60+) faced higher fraud rates. Interpretation: Higher disposable income and possibly lower digital literacy in these groups made them vulnerable. It suggested that fraud prevention strategies should include age-specific awareness initiatives.

Fraudulent Transactions Across Transaction Types: Online transactions had the highest fraud incidence compared to chip and swipe. Interpretation: The lack of physical verification in online transactions made them a frequent target. Strengthening online security measures could significantly reduce fraud rates.

Credit Utilization KDE Plot: Fraudulent transactions had much higher credit utilization ratios (near 1.0). Interpretation: Fraudsters often attempted to maximize the available credit before detection. Monitoring credit utilization spikes could thus serve as an early warning system for fraud.

Correlation Matrix: Showed strongest positive fraud correlations with credit utilization, number of credit cards, and income-related factors. Interpretation: Financial behavior patterns had a direct influence on fraud likelihood. These insights reinforced the importance of incorporating behavioral analytics into fraud detection models.

**4. Modeling**

We developed multiple machine learning models to predict fraud.

**Models Built (placeholders for results):**

* **Random Forest Classifier**
  + Test Set Results:
    - Precision (Fraud class 1.0): 0.00
    - Recall (Fraud class 1.0): 0.02
    - F1-Score (Fraud class 1.0): 0.00
    - Accuracy: 0.98
    - ROC-AUC Score: 0.6756

**Interpretation:** On the test set, the Random Forest model achieved high overall accuracy but struggled to detect actual frauds, achieving only 2% recall. Despite performing well on non-fraudulent transactions, it missed a large portion of fraudulent cases, underscoring the difficulty of working with heavily imbalanced data.

* + Validation Set Results (2016):
    - Precision (Fraud class 1.0): 0.06
    - Recall (Fraud class 1.0): 0.81
    - F1-Score (Fraud class 1.0): 0.11
    - Accuracy: 0.98
    - Macro Average F1-Score: 0.55

**Interpretation:** On the validation set, Random Forest performed much better for fraud detection with a recall of 81%, capturing most fraud cases at the cost of higher false positives. Although precision was low at 6%, the high recall rate was critical for detecting rare but important fraud events.

* **XGBoost Classifier (Validation Set)**
  + Precision (Fraud class 1.0): 0.07
  + Recall (Fraud class 1.0): 0.73
  + F1-Score (Fraud class 1.0): 0.13
  + Accuracy: 0.99
  + ROC-AUC Score: 0.9411

**Interpretation:** On the validation set, the XGBoost model achieved excellent overall performance with a ROC-AUC score of 0.94, indicating strong model capability in distinguishing fraud from non-fraud cases. While precision for fraud detection was low at 7%, the high recall of 73% was critical, meaning the model successfully identified a majority of fraudulent transactions, which is essential for minimizing financial risk.

* **XGBoost Classifier (Test Set)**
  + Precision (Fraud class 1.0): 0.00
  + Recall (Fraud class 1.0): 0.03
  + F1-Score (Fraud class 1.0): 0.01
  + Accuracy: 0.99
  + ROC-AUC Score: 0.6793

**Interpretation:** While the XGBoost model achieved a very high overall accuracy of 99% on the test set, its fraud detection capability was weak, with only 3% recall for fraudulent transactions. This suggests that despite good separation for the overall dataset, the model still struggled to capture rare fraud instances, reflecting the challenge posed by severe class imbalance in real-world fraud detection.

* **Logistic Regression (Validation Set)**
  + Precision (Fraud class 1.0): 0.01
  + Recall (Fraud class 1.0): 0.57
  + F1-Score (Fraud class 1.0): 0.01
  + Accuracy: 0.77
  + ROC-AUC Score: 0.7109

**Interpretation:** On the validation set, Logistic Regression achieved a moderate ROC-AUC score of 0.71. Although precision remained very low at 1%, the recall of 57% showed that the model was able to detect more than half of the fraudulent transactions, making it somewhat useful for identifying frauds despite its limited precision.

* **Logistic Regression (Test Set)**
  + Precision (Fraud class 1.0): 0.00
  + Recall (Fraud class 1.0): 0.38
  + F1-Score (Fraud class 1.0): 0.00
  + Accuracy: 0.78
  + ROC-AUC Score: 0.6096

**Interpretation:** On the test set, Logistic Regression achieved an overall accuracy of 78% and a moderate ROC-AUC of 0.61. Although the model struggled with precision, the 38% recall for fraudulent transactions indicated an improvement in capturing fraud cases compared to random guessing, but still highlighted limitations in handling imbalanced data.

* **Isolation Forest (Unsupervised Anomaly Detection)**
  + Isolation Forest (Unsupervised Anomaly Detection)

Precision (Fraud class 1.0): 0.00

Recall (Fraud class 1.0): 0.01

F1-Score (Fraud class 1.0): 0.00

Accuracy: 0.99

ROC-AUC Score: 0.50

Interpretation:

The Isolation Forest model was effective at identifying non-fraudulent transactions but performed poorly at detecting actual fraud cases, achieving a very low recall and F1-score for the fraud class. The ROC-AUC score of 0.50 indicated random performance for fraud detection. While Isolation Forest was useful for highlighting general outliers, it was insufficient for precise fraud prediction due to the extreme class imbalance and subtle patterns of fraud in the dataset.

**5. Visualizations Summary**

**We conducted an extensive Exploratory Data Analysis (EDA) using Python (Matplotlib/Seaborn).**

The Feature Importance Plot showed 'use\_chip' and 'transaction\_type' as key predictors, with non-chip transactions posing higher risks. The Class Distribution Plot emphasized the severe class imbalance and demonstrated how SMOTE balanced the data. Merchant Category analysis revealed department stores and money transfer services were highly targeted. Older accounts (2005–2010) and cards near expiration were more vulnerable, while users with 3-5 credit cards had higher fraud exposure.

Age group analysis showed that users aged 40–60 and 60+ faced more fraud incidents. Fraud was most frequent in online transactions due to lack of physical verification. The Credit Utilization KDE Plot indicated that fraudsters often max out available credit limits. Finally, the Correlation Matrix highlighted strong links between fraud, high credit utilization, and the number of credit cards owned.

Overall, these visualizations validated our hypotheses and helped shape the modeling strategy.

**6. Prediction Models Summary**

Overall, XGBoost demonstrated the strongest performance during validation with a ROC-AUC score of 0.94 and a recall of 73%, although its test performance dropped significantly. Random Forest showed high validation recall (81%) but struggled on the test set, capturing only 2% of fraud cases. Logistic Regression achieved moderate recall rates (57% on validation and 38% on testing) but had consistently low precision. Isolation Forest, being unsupervised, performed poorly in detecting fraud with a ROC-AUC close to random guessing. These results highlight that while ensemble models like Random Forest and XGBoost excelled on balanced validation data, real-world imbalanced test data remained challenging across all models..

**7. Challenges Faced**

* **Class Imbalance**: Fraudulent transactions formed less than 2% of the dataset, leading to initially poor model sensitivity.
* **Outlier Management**: Differentiating genuine high-value transactions from fraud without losing valid data was difficult.
* **Overfitting Risk**: Especially while using complex models like XGBoost, preventing overfitting required careful hyperparameter tuning.
* **Handling Missing Values**: Ensuring the imputation strategies did not introduce bias into the models.
* **Feature Consistency**: Maintaining consistency while converting categorical fields and normalizing financial figures required extensive validation.

**7. Successes**

* Successfully balanced the dataset using SMOTE.
* Random Forest and XGBoost performed very well, achieving high recall (critical for fraud detection).
* Identified key variables like use\_chip, transaction\_type, and credit\_utilization as strong fraud predictors.
* Visualizations effectively communicated fraud trends and insights.
* Data preparation steps such as correct typecasting and normalization improved model robustness.

**8. Lessons Learned**

* **Handling Imbalanced Datasets**: Balancing classes is critical in fraud detection.
* **Feature Engineering Matters**: Derived features improved model performance significantly.
* **Simple Models Sometimes Win**: Random Forest and Logistic Regression provided very competitive results against more complex models.
* **Visualization is Key**: Data visualization was not just for presentation; it guided many modeling decisions.
* **Data Preparation Quality is Crucial**: Thorough and careful preparation, cleaning, and feature engineering directly impact model success.
* **Importance of Categorical Transformation**: Correctly transforming categorical variables early avoids downstream model confusion.

**9. Future Improvements**

* Implement real-time fraud detection pipeline.
* Explore deep learning models (e.g., LSTM networks) for sequential fraud pattern detection.
* Integrate external data sources (device, IP address, geolocation) to enhance model robustness.
* Experiment with additional anomaly detection algorithms beyond Isolation Forest.
* Apply hyperparameter tuning using GridSearchCV or RandomizedSearchCV for model optimization.

**References**

* Financial Transactions Dataset, Kaggle.
* Gazi, M. S., & Ray, R. K. (2023). Exploring Machine Learning Techniques for Fraud Detection.
* Achary, R., & Shelke, C. J. (2023). Fraud Detection in Banking Transactions Using Machine Learning.
* IEEE Xplore papers on Isolation Forest and ADASYN methods.