Capstone Final Report

Metaverse Fraud Prediction

By Joshua Beasley

Contents:

Introduction:

Problem:

With the rapid growth of the Metaverse, financial transactions in virtual environments have increased significantly. However, this also introduces new avenues for fraudulent transactions. Unlike traditional banking systems, transactions in the Metaverse lack stringent regulatory frameworks, making them prone to fraud. This project aims to develop a predictive model that identifies and flags potential fraudulent transactions within the Metaverse.

Project:

The solution will focus on building a machine learning model that uses pattern recognition and anomaly detection techniques to identify irregular transaction behaviors that deviate from the norm. The success of the project will be measured by the model's accuracy in identifying fraudulent transactions, the reduction in false positives, and its adaptability to new, unknown types of fraud that may evolve as the Metaverse grows.

Clients:

The primary stakeholders are financial regulators within the Metaverse, virtual asset service providers, and end-users who engage in transactions within the Metaverse. Secondary stakeholders include researchers and developers working on digital security solutions.

Methodology:

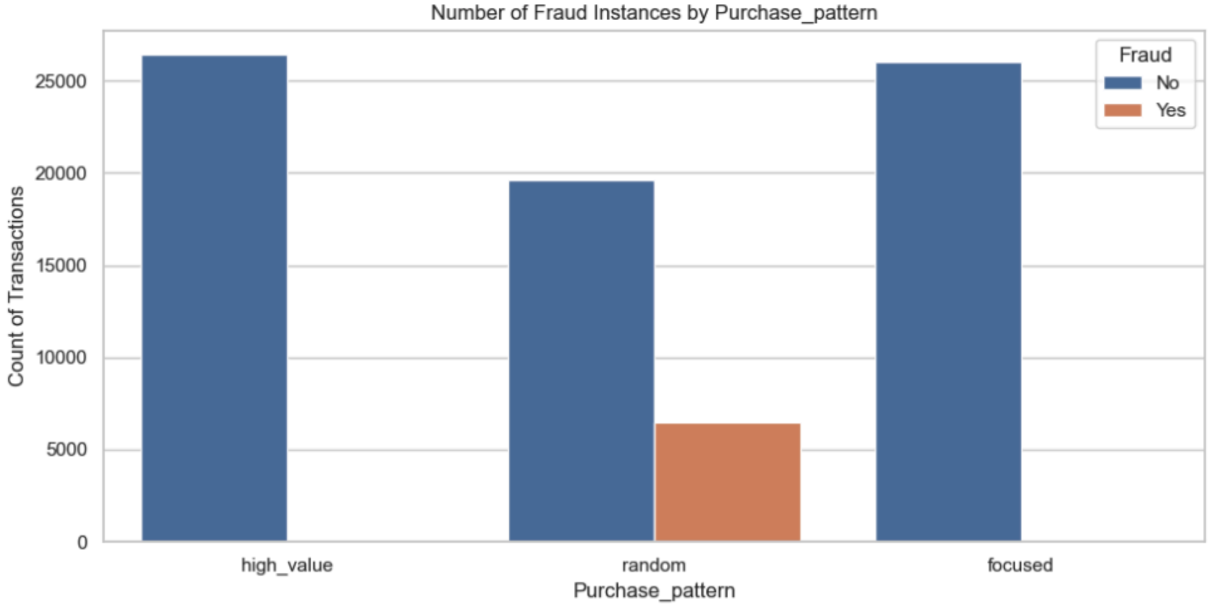
Data:

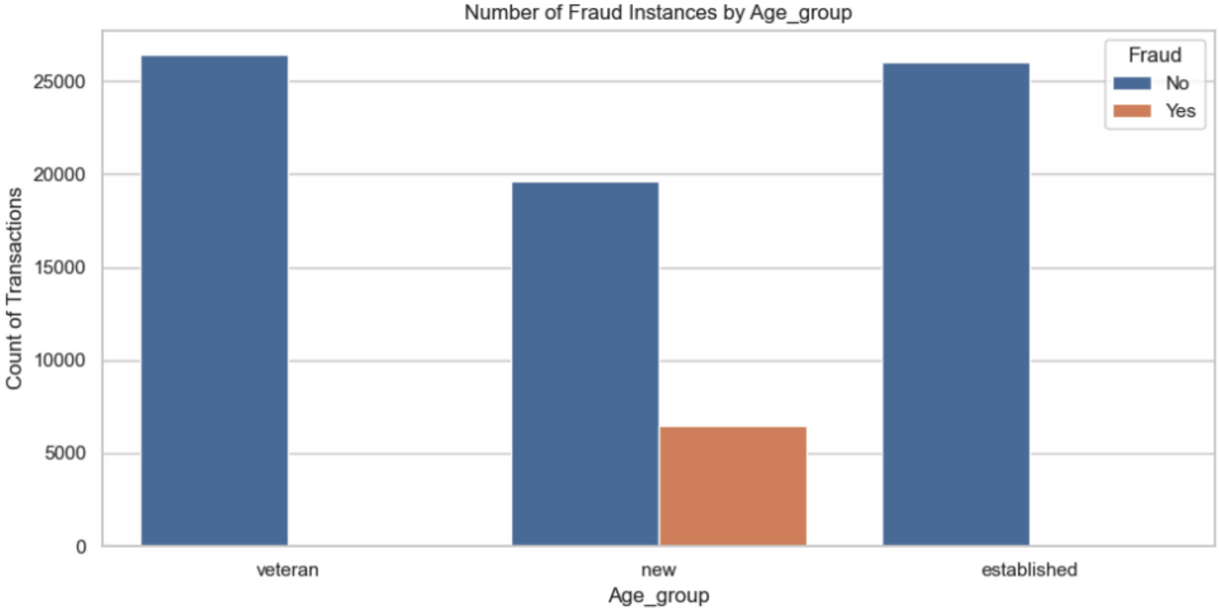
Data will be sourced from Metaverse platforms that record transaction details such as user IDs, transaction amounts, timestamps, and asset types. This data will be accessed through a Kaggle Dataset called "Metaverse Financial Transactions Dataset": https://www.kaggle.com/datasets/faizaniftikharjanjua/metaverse-financial-transactions-dataset

Data Preprocessing:

I conducted extensive data cleaning of the dataset before undertaking further analysis. I leveraged visualizations to identify potential outliers and feature correlations, generated statistical summaries, and uncovered the nature of distributions for each variable. I also engineered some critical features, checked for missing values (there were none, due to it being a clean financial transactions dataset). I then split the data into training and testing subsets, and both scaled the numeric features and encoded the categorical features for model preparation.

Insights from Exploratory Data Analysis:

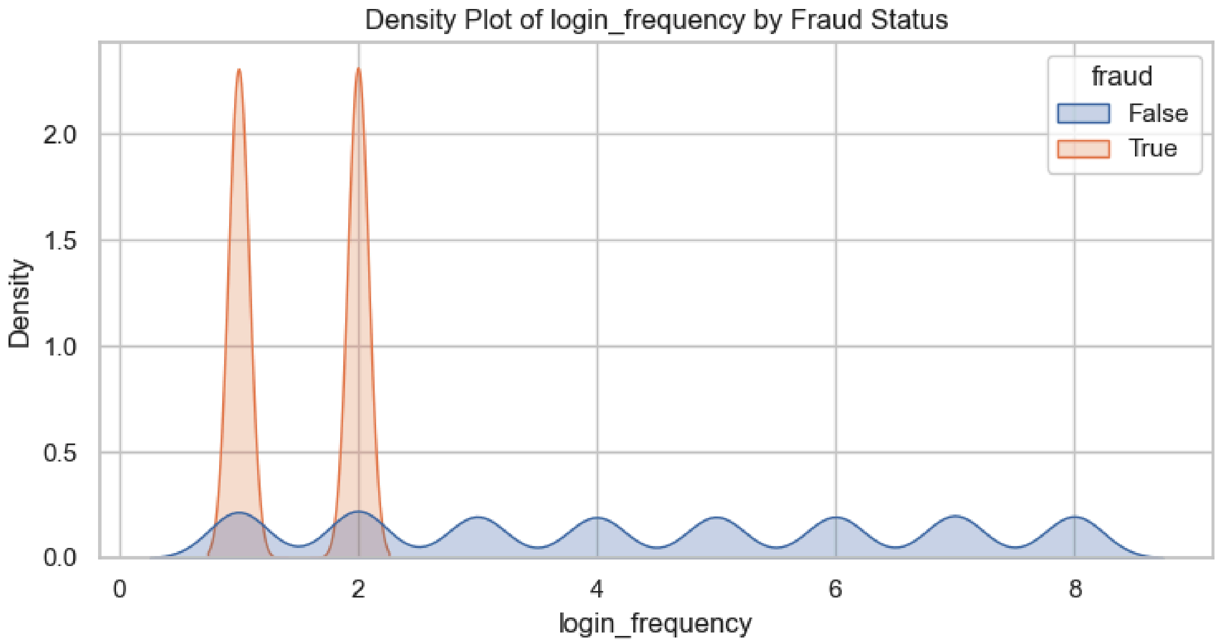


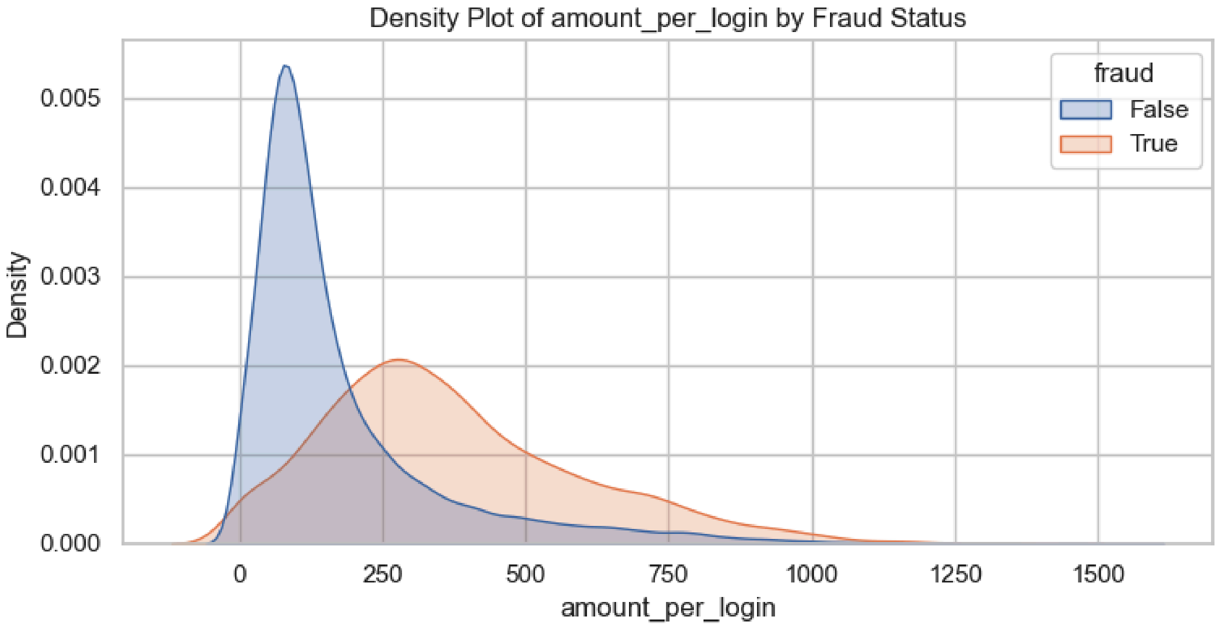


After visualizing fraud instances by the various categories using bar plots, I found that there are no meaningful differences in fraud instances within any of the categorical variables, but with two big exceptions: Purchase Pattern and Age Group.

- **Of the three purchase patterns, the only one with fraud is the random purchase pattern.** This indicates that the random purchase pattern may be more indicative of fraud than other types such as "high_value" or "focused". This makes sense for a few reasons:

    - High value transactions involve significantly larger monetary amounts compared to the typical transactions, and thus could be more scrutinized for fraud as they represent larger financial risks.
    - The "focused" category could refer to transactions that consistently involve specific types of goods or services, or transactions that occur in a regular, predictable manner. This might indicate a customer with specific shopping habits or preferences, and predictability is less likely to be associated with fraud.
    - "Random" transactions might show no clear pattern in terms of timing, amount, or the goods/services involved. This could be seen in sporadic purchases that are unpredictable, which might be flagged for further review since unpredictable transaction patterns can sometimes indicate fraudulent activity.
- **Of the three age groups, the only one with fraud is "new".** This indicates that fraudsters are more likely to be new to the Metaverse based on their activity history, and unlikely to be established or veterans in the Metaverse.





These two density plots yield some interesting insights:

- **Login Frequency by Fraud Status**: The non-fraud distribution shows a pattern of smaller peaks at specific lower frequencies, likely indicating common login behaviors among users. The fraud distribution shows two clear peaks on the left, compared to the wider and more uniform distribution of non-fraud.

- **Amount Per Login by Fraud Status**: The non-fraud distribution shows a peak at a very low amount per login, suggesting regular users have many logins relative to the amount transacted (frequent but small transactions). The fraud distribution is much flatter with a lower peak, suggesting either larger amounts transacted per login or fewer logins for the amount transacted, which could be indicative of attempts to maximize the transaction value in fewer logins.
- TAKEAWAY: Both density plots suggest fraudsters login less frequently (only 1-2 times) for higher average amounts per login compared to legitimate metaverse transactors.

Results and Analysis:

Model Description:

I compared the performance of four different models to determine which one provided superior results, looking at simpler models first and creating more complex models as we progressed. I also performed cross-validation to evaluate how well a model is likely to perform on unseen data by using different subsets of the training data for both training and validation.

Here are all the models tested with their descriptions and parameters:

- *Logistic Regression Model:* estimates the probabilities of a binary outcome (typically representing two classes) by applying a logistic function to a linear combination of the input features.
    - Default Parameters used
- *Random Forest Model:* ensemble learning method that builds multiple decision trees during training and outputs the class that is the mode of the mean prediction (regression) of the individual trees.
    - Parameters: n_estimators=100, random_state=42
- *Gradient Boosting Model:* a powerful ensemble technique that builds models sequentially, each new model correcting errors made by the previous ones.
    - Parameters: n_estimators=100, random_state=42
- *Support Vector Machines (SVM):* classification model that finds the hyperplane which best separates different classes in the feature space, maximizing the margin between the closest points of the classes (support vectors).
    - Parameters: kernel='linear', probability=True, random_state=42

Performance Metrics:

- **Accuracy**: Measures the overall correctness of the model but can be misleading if the data is imbalanced.
- **Precision and Recall**: Especially important in fraud detection to minimize false positives (precision) and false negatives (recall). We might prioritize recall to capture as many fraudulent transactions as possible, even if it means enduring more false positives.
- **F1 Score**: Harmonic mean of precision and recall. It's a balanced metric that is useful when you need a trade-off between precision and recall.
- **ROC-AUC**: Measures the ability of your model to discriminate between classes. A higher AUC value indicates a better performing model.
- **Confusion Matrix**: Provides a summary of prediction results on a classification problem. Great for visualizing the performance of an algorithm.

These metrics will provide a comprehensive view of the model's performance, considering both the accuracy and the robustness in handling class imbalance, which is typical in fraud detection scenarios.

Model Performance:

After evaluating various models, including Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machines, **the model that best balances precision and recall while minimizing false negatives is the Random Forest classifier with class weighting set to 'balanced' (10 feature set).** Minimizing false negatives is essential in fraud prediction to ensure no fraudulent transactions slip through, and this model did the best job of this.

After tuning the Random Forest model hyperparameters with GridSearchCV, the model produced an extremely low recall, indicating that the model failed to identify the majority of fraudulent transactions. To address this, I applied a few techniques: I implemented SMOTE (Synthetic Minority Over-sampling Technique) to oversample the minority class, applied cost-sensitive learning (which adjusts the model to give more importance to certain classes during training), and adjusted the classification threshold.

Recall performance dramatically improved from these tweaks, so I then reduced the original 48 feature set to 10 features and applied the same tuning techniques, which yielded virtually identical results to the full set model while being more interpretable. This indicates the reduced amount had a minimal impact on model predictive accuracy.

- **Precision**: 25%

- **Recall**: 100% (1.0)
- **F1-Score**: 39%
- **Confusion Matrix**:
  - True Negatives: 10,621
  - False Positives: 3,848
  - False Negatives: 2
  - True Positives: 1249

Conclusion and Recommendations:

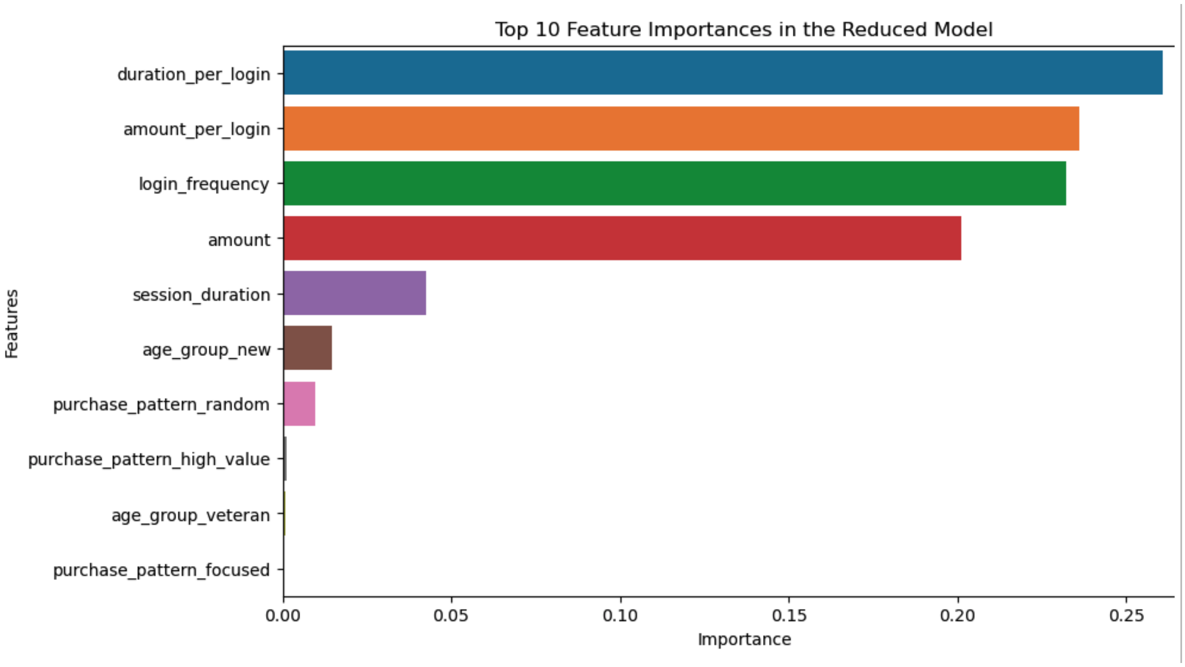
Model Findings:

This model was specifically effective due to its ability to adapt to the class imbalance inherent in the dataset, significantly improving the recall to ensure almost no fraudulent transaction is missed, which is crucial for fraud prevention systems.

However, the obvious limitation to this model is its relatively low precision, which indicates that while the model is highly effective in identifying fraudulent transactions (high recall), it also incorrectly classifies a significant number of legitimate transactions as fraudulent (high false positives). This can lead to increased operational costs and potential customer dissatisfaction due to potentially unnecessary fraud investigations.

Our best parameters that raised our precision while maintaining a high recall were:

- max_depth: 10
- min_samples_leaf: 4
- min_samples_split: 10
- n_estimators: 300



The feature importance analysis revealed that **the five most important features for the model, by a wide margin, are: duration_per_login, amount_per_login, login_frequency, amount, and session_duration.**

This yields a few insights:

- **Duration Per Login**: this is of prime importance since there tends to be a slightly higher duration per login for fraudulent transactions, indicating fraudsters take their time for enacting their criminal activities.
- **Login Frequency and Session Duration**: It's unsurprising that both are among the top five most important features, since we discovered during EDA that both display moderate negative correlations with fraud. This implies that lower login frequencies as well as lower total session durations are associated with fraud. So fraudsters login less frequently, and thus their cumulative session durations are lower than average, but they have longer session times per each individual login (hence the importance of duration per login).
- **Amount Per Login**: We discovered during our EDA section that Amount Per Login shows a moderate positive correlation with fraud, and we now see it as among the top two most important features. This indicates that larger but less frequent transactions are associated with fraudulent behavior.

Recommendations:

Based on the modeling results and overall analysis, here are three concrete recommendations:

1. **Implement Real-Time Monitoring**: Integrate the fraud prediction model into your transaction processing system to enable real-time fraud detection, allowing for immediate action on suspicious transactions, thereby reducing potential losses from delayed fraud identification.
2. **Continuous Model Updates**: Regularly update the fraud prediction model with new transaction data to adapt to evolving fraud patterns and techniques, ensuring that the model remains effective over time and reduces the incidence of both false positives and missed fraud cases.
3. **Enhanced Customer Verification Measures**: Based on insights from the model, especially from features heavily influencing fraud predictions, implement stronger verification processes for transactions identified as high-risk, such as two-factor authentication or manual reviews, to further safeguard against fraud.

Ideas for Further Research:

For future work on this project or similar projects, the following recommendations are made to enhance model performance and operational effectiveness:

1. **Feature Engineering**: Explore additional features or interactions that may improve the model's ability to discriminate between classes, which would ideally increase our currently low precision to allow for a greater balance between this and recall.
2. **Alternative Algorithms**: Investigate other algorithms known for handling imbalanced data well, such as XGBoost or LightGBM, which might offer improvements in precision.
3. **Incremental Learning**: Consider models that support incremental learning, allowing the system to evolve as new data becomes available.
4. **Deployment Strategy**: Develop a deployment strategy that includes real-time analysis and the ability to retrain the model periodically with new transaction data.
5. **Cost-Benefit Analysis**: Perform a detailed cost-benefit analysis to better understand the implications of false positives versus false negatives, potentially adjusting the class weights or threshold accordingly.

By following these recommendations, the fraud detection system can be further refined and better integrated into operational frameworks to reduce both the incidence and impact of financial fraud.

References:

Here is the link to the Kaggle data source used for this project: https://www.kaggle.com/datasets/faizaniftikharjanjua/metaverse-financial-transactions-dataset