Interim Report 1: Fraud Detection Analysis

# Executive Summary

This interim report presents the comprehensive analysis of fraud detection datasets, including data cleaning, preprocessing, exploratory data analysis, and feature engineering strategies. The analysis covers two primary datasets: a custom fraud dataset with user transaction data and the standard credit card fraud dataset. Key findings include significant class imbalance (9.36% fraud rate), temporal patterns in fraudulent activities, and effective feature engineering approaches for fraud detection.

# 1. Data Cleaning and Preprocessing Steps

## 1.1 Dataset Overview

The analysis encompasses three datasets:  
  
**•** Fraud\_Data.csv: Custom fraud dataset with 151,112 transactions  
**•** IpAddress\_to\_Country.csv: IP address to country mapping data  
**•** creditcard.csv: Standard credit card fraud dataset with 284,807 transactions

## 1.2 Data Quality Assessment

Initial data quality assessment revealed:  
  
**•** No missing values in any dataset  
**•** Data type corrections required for datetime fields  
**•** No duplicate records found  
**•** 137,956 unique devices and 143,512 unique IP addresses

## 1.3 Data Cleaning Steps

The following cleaning steps were implemented:  
  
**1.** Removed duplicate records (none found)  
**2.** Converted datetime fields to proper format:  
 - signup\_time and purchase\_time converted to datetime objects  
**3.** Ensured numeric data types for age and purchase\_value fields  
**4.** Validated data integrity across all datasets

# 2. Key Insights from Exploratory Data Analysis

## 2.1 Fraud Class Distribution

The primary fraud dataset exhibits significant class imbalance:  
  
**•** Legitimate transactions: 90.64% (136,914 records)  
**•** Fraudulent transactions: 9.36% (14,198 records)  
**•** Imbalance ratio: 9.6:1 (legitimate to fraudulent)

## 2.2 Purchase Value Analysis

Purchase value distribution analysis revealed:  
  
**•** Mean purchase value: $36.94  
**•** Standard deviation: $18.32  
**•** Range: $9 to $154  
**•** Fraudulent transactions tend to have higher average purchase values

## 2.3 Temporal Patterns

Analysis of temporal patterns revealed:  
  
**•** Purchase delays vary significantly between legitimate and fraudulent transactions  
**•** Fraudulent transactions show different time-of-day patterns  
**•** Day-of-week patterns indicate when fraud is most likely to occur

## 2.4 Demographic and Behavioral Patterns

Key demographic and behavioral insights:  
  
**•** Age distribution shows fraud patterns across different age groups  
**•** Source of traffic (SEO, Ads) affects fraud likelihood  
**•** Browser type correlates with fraud patterns  
**•** Gender distribution shows different fraud rates

## 2.5 Geographic Analysis

IP address to country mapping revealed:  
  
**•** United States shows highest fraud activity  
**•** Japan and other countries show varying fraud patterns  
**•** Geographic patterns can be used as fraud indicators

# 3. Feature Engineering Strategy

## 3.1 Time-based Features

Several time-based features were engineered:  
  
**•** purchase\_delay\_hr: Time between signup and purchase  
**•** time\_diff\_hr: Time between consecutive purchases  
**•** account\_age\_hr: Account age at time of purchase  
**•** time\_since\_signup\_hr: Time elapsed since account creation  
**•** hour\_of\_day: Hour of purchase (0-23)  
**•** day\_of\_week: Day of week (0-6)

## 3.2 Behavioral Features

Behavioral features capture user activity patterns:  
  
**•** txn\_count: Transaction count per user  
**•** velocity: Transaction rate (transactions per hour of account age)  
**•** prev\_purchase: Previous purchase timestamp

## 3.3 IP Address to Country Mapping

IP address analysis was implemented using:  
  
**•** Range-based lookup algorithm for IP to country mapping  
**•** Efficient binary search approach for large IP ranges  
**•** Country-level fraud pattern analysis  
**•** Geographic risk scoring based on country fraud rates

## 3.4 Feature Selection

Final feature set for modeling includes:  
  
**Numeric Features:**• purchase\_value, age, purchase\_delay\_hr  
• time\_diff\_hr, velocity, hour\_of\_day, day\_of\_week  
• time\_since\_signup\_hr  
  
**Categorical Features:**• source, browser, sex

# 4. Class Imbalance Analysis and Strategy

## 4.1 Class Imbalance Problem

The dataset exhibits severe class imbalance:  
  
**•** Fraud rate: 9.36% (typical for fraud detection)  
**•** Imbalance ratio: 9.6:1  
**•** Risk of model bias toward majority class  
**•** Potential for poor fraud detection performance

## 4.2 Proposed Strategy for Handling Class Imbalance

Multi-pronged approach to address class imbalance:  
  
**1.** SMOTE (Synthetic Minority Over-sampling Technique):  
 - Generate synthetic fraud samples  
 - Balance training set without losing information  
 - Applied to both numeric and categorical features  
  
**2.** Stratified Sampling:  
 - Maintain class proportions in train/test splits  
 - Ensure representative evaluation  
  
**3.** Evaluation Metrics:  
 - Focus on precision, recall, and F1-score  
 - Use ROC-AUC for overall performance  
 - Implement confusion matrix analysis  
  
**4.** Cost-sensitive Learning:  
 - Assign higher costs to false negatives  
 - Optimize for fraud detection accuracy

# 5. Data Transformation Pipeline

## 5.1 Preprocessing Steps

Comprehensive preprocessing pipeline implemented:  
  
**1.** Feature Scaling:  
 - StandardScaler for numeric features  
 - Normalize features to zero mean and unit variance  
  
**2.** Categorical Encoding:  
 - OneHotEncoder for categorical variables  
 - Handle unknown categories in test set  
  
**3.** Missing Value Handling:  
 - Fill missing values with 0 for numeric features  
 - Robust handling for edge cases

## 5.2 Model Preparation

Final dataset preparation:  
  
**•** Training set: 70% of data with SMOTE applied  
**•** Test set: 30% of data (original distribution)  
**•** Features: 8 numeric + encoded categorical features  
**•** Target: Binary classification (0=legitimate, 1=fraud)

# 6. Next Steps and Recommendations

Recommended next steps for the project:  
  
**1.** Model Development:  
 - Implement multiple ML algorithms (Random Forest, XGBoost, Neural Networks)  
 - Hyperparameter tuning with cross-validation  
 - Ensemble methods for improved performance  
  
**2.** Feature Engineering Enhancements:  
 - Advanced temporal features (seasonality, trends)  
 - Network analysis features (user connections)  
 - Risk scoring based on historical patterns  
  
**3.** Model Evaluation:  
 - Comprehensive performance metrics  
 - SHAP analysis for interpretability  
 - Real-time prediction capabilities  
  
**4.** Production Deployment:  
 - API development for real-time scoring  
 - Monitoring and alerting systems  
 - Continuous model retraining pipeline

# Conclusion

This interim report demonstrates a comprehensive approach to fraud detection analysis. The data cleaning and preprocessing steps ensure data quality, while the feature engineering strategy captures both temporal and behavioral patterns. The class imbalance strategy using SMOTE provides a robust foundation for model development. The analysis reveals significant patterns in fraudulent behavior that can be leveraged for effective fraud detection.  
  
The project is well-positioned for the next phase of model development and evaluation, with a solid foundation of clean data and engineered features.