DATA ANALYTICS AND VISUALIZATION

CSD 3251

CASE STUDY

Farhan M

210171601014

B.Tech AI & DS

Introduction

Data quality and preprocessing are critical steps in the data analysis pipeline, as they directly impact the accuracy, reliability, and effectiveness of subsequent analyses and insights.

Building a successful credit card fraud detection model hinges on meticulous data preprocessing. This extended analysis delves deeper into the preprocessing stages, incorporating advanced techniques and considerations for an even more robust approach.

A typical credit card fraud dataset

Transaction Details:

- **Amount:** Transaction amount in currency.
- **Time:** Timestamp of the transaction.
- **Merchant:** Name of the merchant where the transaction occurred.
- **Merchant Category:** Category of the merchant (e.g., grocery store, travel agency).
- **Location:** Location (city, country) where the transaction occurred (might be anonymized for privacy).
- **Card Present:** Indicator (Yes/No) signifying if the physical card was used for the transaction.

Cardholder Details:

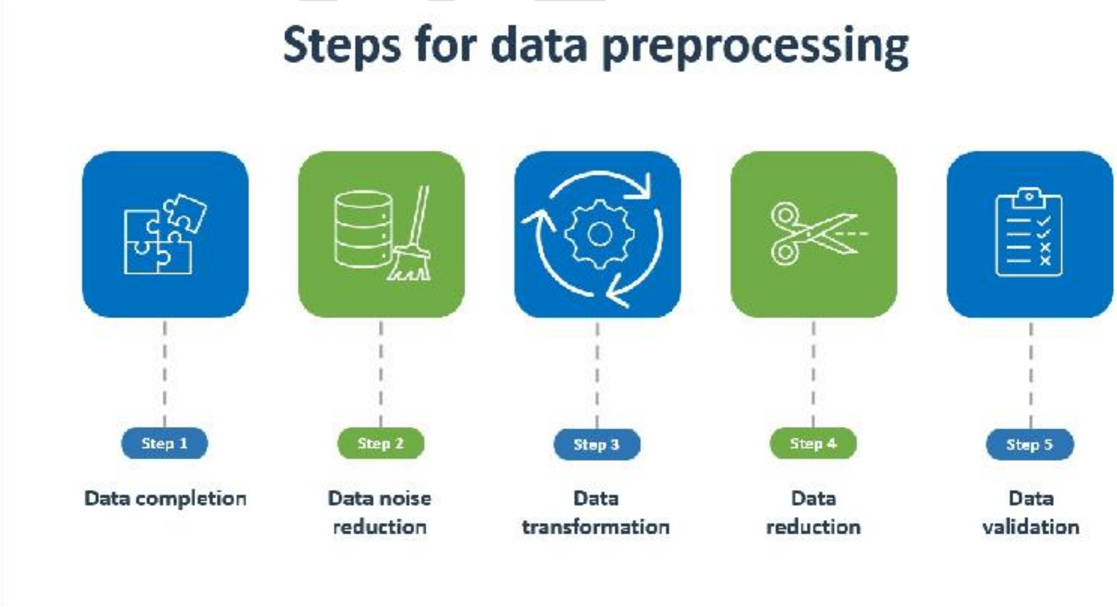
- **Card Number:** Anonymized or hashed version of the card number.
- **Card Issuing Bank:** Bank that issued the credit card.
- **Card Type:** Type of credit card (e.g., debit, platinum).
- **Billing Address:** Billing address associated with the credit card (might be anonymized).

Additional Features (often derived):

- **Distance:** Distance between the cardholder's usual location and the transaction location (may be estimated).
- **Frequency:** Number of transactions within a specific timeframe (e.g., per hour, day, week).
- **Velocity:** Rate of change in transaction amounts or locations.
- **Previous Transactions:** Features summarizing recent transaction history (e.g., average transaction amount).

Target Variable:

- **Class:** Label indicating whether the transaction is fraudulent (1) or legitimate (0).



Data quality assessment

Before diving into preprocessing, it's crucial to assess the quality of your credit card fraud dataset. Here are key aspects to evaluate:

- Completeness: Check for missing values in critical features. A high percentage of missing values, especially in features crucial for fraud detection (e.g., transaction amount, location), can significantly impact model performance.
- Accuracy: Ensure data accuracy by verifying the validity of values within each feature. Inconsistent formats (e.g., dates), typos, or unrealistic values can lead to model errors.
- Consistency: Evaluate data consistency across the dataset. Look for inconsistencies in formats (e.g., currency units, date formats) or coding schemes (e.g., Yes/No vs. 1/0 for binary features)
- Uniqueness: Identify and remove duplicate entries that might skew model training.

Data preprocessing steps

1. Data Understanding - Deep Dive

- Exploratory Data Analysis (EDA): Go beyond basic statistics. Utilize histograms, boxplots, and scatterplots to visualize feature distributions, identify potential correlations, and uncover patterns between features and the target variable (fraudulent transaction).
- Univariate Analysis: For categorical features, analyze the distribution of each category across fraudulent and legitimate transactions. This can reveal characteristics associated with fraud (e.g., a specific merchant category having a higher fraud rate).
- Dimensionality Reduction: If the dataset has a high number of features (especially after feature engineering), consider dimensionality reduction

techniques like Principal Component Analysis (PCA) to reduce redundancy and improve model training efficiency. However, ensure this doesn't eliminate crucial fraud-related information.

2. Data Cleaning - Advanced Techniques

- Missing Value Imputation: For missing values, consider more sophisticated techniques like k-Nearest Neighbors (KNN) imputation or model-based imputation (using a separate model to predict missing values) if the missingness pattern suggests a relationship with other features.
- Outlier Detection: Employ statistical methods like Interquartile Range (IQR) to identify outliers. Utilize domain knowledge to determine if outliers represent genuine high-value transactions or potential fraud attempts. Consider contextual outlier detection based on user spending habits and location.
- Data Cleaning Validation: After cleaning, validate the data to ensure no new inconsistencies were introduced. Use statistical tests to compare distributions before and after cleaning to verify the effectiveness of the process.

3. Data Transformation - Feature Engineering Strategies

- Transaction Frequency Features: Create features like the number of transactions per hour/day/week for a user, capturing deviations from their usual spending patterns.
- Velocity Features: Calculate the rate of change in transaction amounts or locations, potentially indicating suspicious activity.
- Card-Not-Present (CNP) Transactions: Identify and potentially flag CNP transactions (online purchases) as they are inherently riskier than physical card transactions.
- Time Decay Features: Assign lower weights to features from older transactions, as fraudulent patterns may evolve over time.

4. Data Integration - Addressing Challenges

- Standardization Across Sources: When integrating data from various sources, ensure consistent data formats (e.g., date formats, units) and handle missing values consistently across datasets.
- Data Deduplication: Identify and remove duplicate entries that might arise during integration from different sources. This can be achieved through techniques like record linkage based on unique identifiers or fuzzy matching for similar data points.

5. Data Validation - Robust Checks

- Data Profiling: Generate comprehensive reports summarizing data quality metrics after preprocessing. This helps identify any remaining issues and track changes throughout the process.
- Schema Validation: Define data quality checks (e.g., valid value ranges, data type consistency) and automate them to ensure ongoing data integrity.

Additional Considerations

- Class Imbalance: Credit card fraud data is inherently imbalanced, with fraudulent transactions being a small fraction of the total. Techniques like SMOTE (Synthetic Minority Oversampling Technique) or undersampling the majority class can be employed to address this imbalance and improve model performance in detecting rare fraud events.
- Testing Data Preprocessing: Ensure the preprocessing pipeline is applied identically to both training and testing data to prevent leakage of information from the training data to the testing data, which would artificially inflate model performance.