Credit Card Fraud Detection with Resampling Techniques

Objective: The primary goal of this project is to develop an effective machine learning model for detecting fraudulent credit card transactions. This case study focuses on addressing the significant challenge of **class imbalance** present in the dataset, where fraudulent transactions are extremely rare. Various data resampling techniques are explored to improve model performance, particularly the ability to correctly identify fraud.

1. Data Loading and Exploration

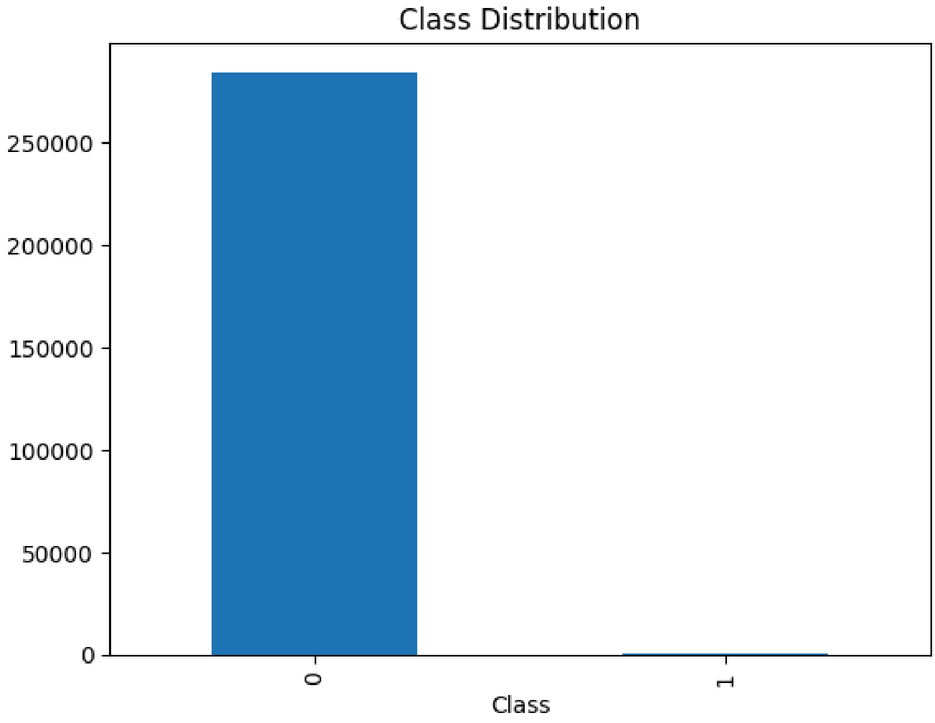
The analysis begins by loading the `creditcard.csv` dataset and performing an initial exploratory data analysis. The dataset contains anonymized features (V1 through V28) from a Principal Component Analysis (PCA), along with `Time` and transaction `Amount`.

1.1. Class Distribution

A critical characteristic of this dataset is its severe imbalance. The vast majority of transactions are legitimate (Class 0), while fraudulent transactions (Class 1) are a tiny minority.

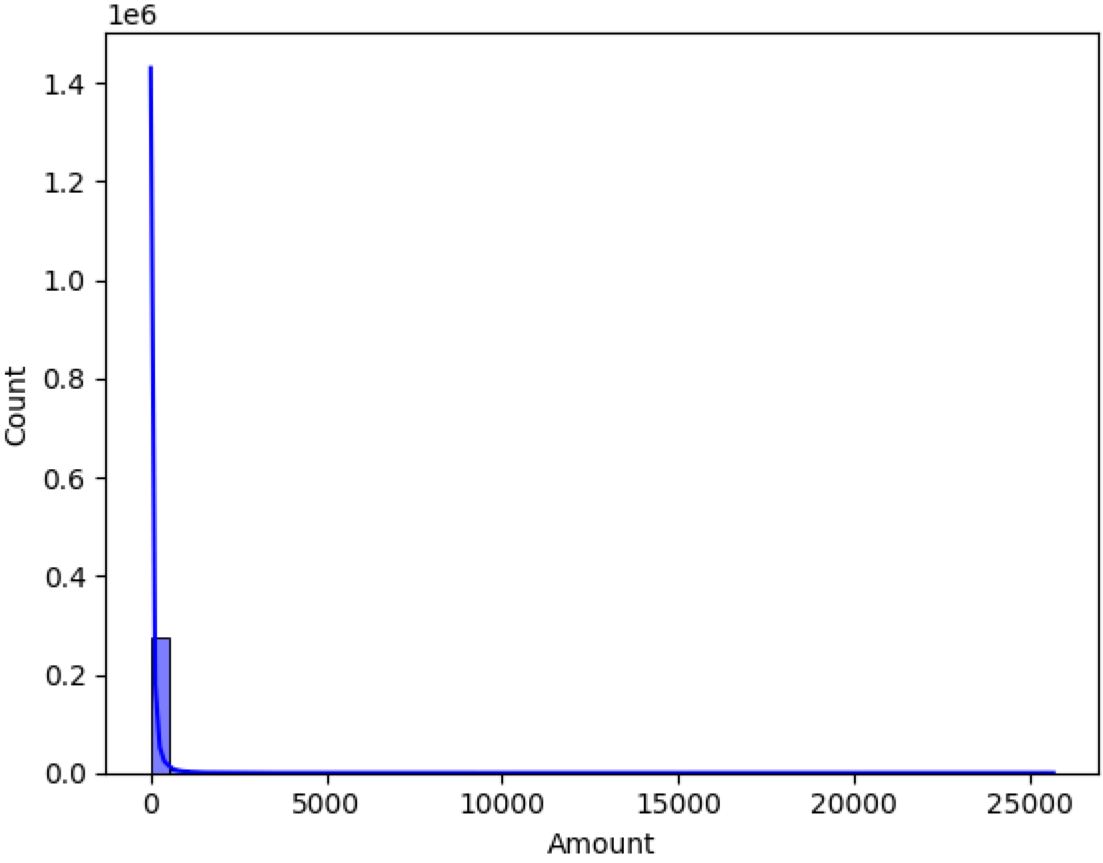
- **Non-Fraudulent (Class 0):** 227,451 transactions
- **Fraudulent (Class 1):** 394 transactions

This extreme imbalance is visualized in the bar chart below, highlighting the challenge for any predictive model.



1.2. Transaction Amount Distribution

The distribution of the transaction `Amount` was also analyzed. The histogram shows that most transactions are for small amounts, with a heavy right skew.



---

2. Data Preprocessing

Before training the models, the data was prepared through scaling and splitting.

- **Feature Scaling:** All features, including `Time` and `Amount`, were scaled using `StandardScaler`. This standardization is crucial for distance-based algorithms like KNN and helps with the convergence of models like Logistic Regression.
- **Data Splitting:** The dataset was divided into an **80% training set** and a **20% testing set**.

---

3. Handling Class Imbalance with Resampling

To address the class imbalance, several resampling techniques were applied to the **training data only**. This ensures the test set remains a realistic, imbalanced representation of real-world data.

- **Random Under-Sampling (RUS):** This method reduces the size of the majority class by randomly removing samples until it balances with the minority class.
    - *Resulting Training Shape:* 394 Fraudulent, 394 Non-Fraudulent samples.
- **Random Over-Sampling (ROS):** This method increases the size of the minority class by randomly duplicating existing samples until it balances with the majority class.
    - *Resulting Training Shape:* 227,451 Fraudulent, 227,451 Non-Fraudulent samples.
- **SMOTE (Synthetic Minority Over-sampling Technique):** Instead of duplicating, SMOTE creates new, synthetic samples for the minority class based on the characteristics of its nearest neighbors.
    - *Resulting Training Shape:* 227,451 Fraudulent, 227,451 Non-Fraudulent samples.
- **Tomek Links:** This is an under-sampling technique that identifies and removes pairs of very close samples from opposite classes (Tomek Links). This helps to clean the boundary between the classes.
    - *Resulting Training Shape:* 394 Fraudulent, 227,434 Non-Fraudulent samples (a minor reduction in the majority class).
- **SMOTE-Tomek:** This hybrid technique first applies SMOTE to generate synthetic data and then uses Tomek Links to clean up any resulting noise.
    - *Resulting Training Shape:* 227,451 Fraudulent, 227,451 Non-Fraudulent samples.

---

4. Model Building and Evaluation

Two models, **Logistic Regression** and **K-Nearest Neighbors (KNN)**, were trained on the original and each of the resampled datasets. Performance was evaluated using metrics better suited for imbalanced classification than simple accuracy: **Precision**, **Recall**, **F1 Score**, and **ROC AUC Score**.

4.1. Logistic Regression Results

- **Original Data:** Achieved high precision (0.86) but very poor recall (0.58), meaning it missed over 40% of fraudulent transactions.
- **Under-Sampled Data (RUS):** Recall skyrocketed to 93%, but precision plummeted to just 4%. This model flags too many legitimate transactions as fraudulent, making it impractical.
- **Over-Sampled Data (ROS & SMOTE):** Both techniques significantly boosted recall to around 92% while maintaining a higher ROC AUC score. However, precision remained low (6%).
- **Tomek Links Data:** Performance was nearly identical to the original dataset, as very few samples were removed.
- **SMOTE-Tomek Data:** Results were the same as with SMOTE, achieving high recall but low precision.

4.2. K-Nearest Neighbors (KNN) Results

- **Original Data:** KNN performed remarkably well, achieving a high precision of 0.94 and a solid recall of 0.78. This resulted in the **highest F1-score (0.85)** of all experiments, indicating a strong balance.

- **Under-Sampled Data (RUS):** Similar to Logistic Regression, recall was high (90%), but precision was very low (6%).
- **Over-Sampled Data (ROS):** Achieved a good recall of 86% with a respectable precision of 71%.
- **SMOTE & SMOTE-Tomek Data:** Both methods yielded identical results, with a good recall of 87% but lower precision (48%).
- **Tomek Links Data:** Performance was again almost identical to the original dataset, achieving a very high F1-score of 0.86.

5. Conclusion and Model Comparison

The experiments demonstrate that resampling techniques have a profound impact on model performance for imbalanced datasets.

- **Under-sampling** is generally not recommended as it leads to an unacceptably high number of false positives (low precision).
- **Over-sampling techniques (ROS, SMOTE)** are effective at increasing recall, ensuring more fraudulent transactions are caught. However, this often comes at the cost of lower precision.
- The **KNN model**, even on the original imbalanced data, provided an excellent balance between precision and recall, suggesting it is robust to the class imbalance in this particular dataset. The Tomek Links method, which slightly cleans the data, also produced a top-performing KNN model.

Comparison Table

The table below summarizes the performance metrics for both models across all resampling techniques. The **F1-Score** is highlighted as it represents the harmonic mean of Precision and Recall.

Dataset	Model	Precision	Recall	F1 Score	ROC AUC
Original	Logistic Regression	0.864	0.582	0.695	0.791
Original	KNN	0.938	0.776	0.849	0.888
Random Under-Sampling	Logistic Regression	0.04	0.929	0.077	0.945
Random Under-Sampling	KNN	0.063	0.898	0.118	0.938
Random Over-Sampling	Logistic Regression	0.063	0.918	0.117	0.947
Random Over-Sampling	KNN	0.706	0.857	0.774	0.928
SMOTE	Logistic Regression	0.059	0.918	0.111	0.947
SMOTE	KNN	0.48	0.867	0.618	0.933

Tomek Links	Logistic Regression	0.848	0.571	0.683	0.786
Tomek Links	KNN	0.939	0.786	0.856	0.893
SMOTE-Tomek	Logistic Regression	0.059	0.918	0.111	0.947
SMOTE-Tomek	KNN	0.48	0.867	0.618	0.933

Based on the results, the **K-Nearest Neighbors model trained on the Tomek Links-cleaned data** provided the best overall performance, achieving the highest F1-Score of **0.856**. It successfully balances the need to identify fraudulent transactions (high recall) while minimizing false alarms (high precision).