Table Of Contents:

Credit Card Fraud Detection Problem

Credit card fraud is a significant challenge in today's digital economy, with fraudulent transactions causing substantial financial losses each year. The problem arises because detecting fraud is like finding a needle in a haystack, as **fraudulent transactions are far less common than legitimate ones**. This imbalance between fraud and non-fraud cases **presents unique challenges** in training machine learning models, as the model tends to be **biased** towards the majority class (non-fraudulent transactions), **leading to poor detection of actual fraud.**

Preprocessing Techniques for Fraud Detection

To address the imbalance and prepare the dataset for accurate predictions, the preprocessing phase involved several steps:

1. Handling Missing Data: The first step was to check for any null or missing values in the dataset. Ensuring a complete dataset is critical for accurate model training, and any missing data was either filled or discarded accordingly.

2. Class Imbalance Visualization: A **key challenge** in this problem is the **imbalance** between fraud and non-fraud transactions. Upon visualizing the class distribution, it was confirmed that the dataset had a significant imbalance, with **only 498 fraud** cases compared to **over 200,000 non-fraud cases.** This imbalance would result in **bias** and insufficient training for fraud detection, so addressing it was crucial.

3. Feature Scaling: To normalize the data, we used a Standard Scaler. A Min-Max scaler was deemed unsuitable due to the high dimensionality of the dataset, as many features contained outliers. Given that Principal Component Analysis (PCA) had already been applied to reduce dimensionality, the standard scaling helped stabilize the feature ranges for more effective learning.

4. Exporting the Cleaned Dataset: After preprocessing, the cleansed and under-sampled dataset was exported for model training. An alternative dataset was also created using GAN augmentation to balance the class distribution by oversampling the fraud cases.

5. Handling Class imbalance problem:

Resampling the Dataset: To mitigate the class imbalance, we explored two main techniques:

- **Under-sampling:** We reduced the number of non-fraudulent transactions to match the count of fraud transactions (482 examples). This method helps balance the dataset but risks losing valuable information from the legitimate transactions.

- **Over-sampling using GAN:** To avoid information loss, we applied Generative Adversarial Networks (GANs) to synthetically generate new fraud examples, thereby increasing the size of the fraud class to match the non-fraudulent transactions. While this technique provided more balanced data, it led to a large dataset, slowing down the computation due to the sheer number of non-fraud cases (over 200k).