# Credit Card Fraud Detection Research Report

Hou Zheyan Hu Yali Zhao Xuanzhi

## 1 Research Background and Objectives

### 1.1 Research Background

For credit card companies, accurately identifying fraudulent transactions is crucial to prevent customers from being charged for purchases they did not make. This study is based on a dataset of transactions conducted over two days, containing 284,807 transactions in total, of which only 492 are fraudulent. Fraudulent transactions account for merely 0.172

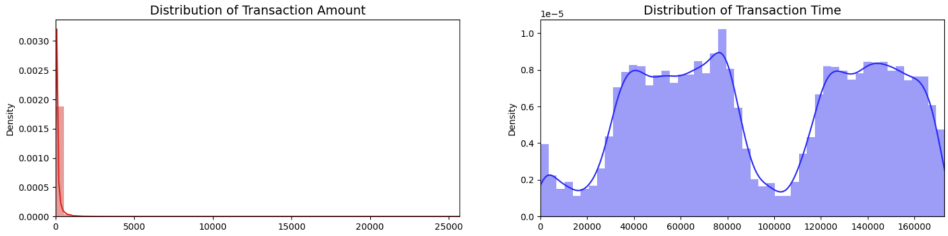


Figure 1: Distribution of Transaction Amount and Transaction Time

### 1.2 Research Objectives

1. Understand the distribution characteristics of the existing data.
2. Identify suitable classifiers and select the one with the highest accuracy.
3. Develop a neural network and compare its accuracy with that of the best-performing classifier.

## 2 Data Overview

### 2.1 Data Summary

- **Transaction Amounts**: The overall transaction amounts are relatively small, with an average value of approximately $88.
- **Data Completeness**: There are no null values in the dataset, eliminating the need for missing value handling.
- **Transaction Class Distribution**: Non-fraudulent transactions dominate, accounting for 99.83

### 2.2 Technical Notes on Features

- **Anonymization of Features**: For privacy protection, all feature names except `Time` and `Amount` have been anonymized and are labeled as V1 to V28.
- **PCA Dimensionality Reduction**: Features V1 to V28 have undergone Principal Component Analysis (PCA), a dimensionality reduction technique.
- **Data Standardization**: Since PCA requires feature standardization prior to application, we assume the dataset's developers have already standardized the V1 to V28 features.

# 3 Preprocessing Phase

## 3.1 Standardization and Distribution Handling

The original `Time` and `Amount` features were standardized, generating new columns labeled `scaled_time` and `scaled_amount`, respectively.

## 3.2 Data Splitting

A balanced dataset of 984 transactions was created by randomly selecting 492 normal transactions from the original 284,315 and combining them with all 492 fraudulent transactions. The test set from the original dataset was kept unchanged, with undersampling applied only to the training set to ensure model evaluation reflects real-world data distribution.

# 4 Random Undersampling

After completing the basic data processing, we first employed the random undersampling method. Undersampling is a technique that achieves a more balanced dataset by removing instances from the majority class (i.e., non-fraudulent transactions in this dataset), thereby preventing model overfitting. However, it should be noted that undersampling may result in the classification model not performing as accurately as desired due to potential information loss. For this specific dataset, there were a total of 492 fraudulent transaction records. From the over 200,000 non-fraudulent transactions, we obtained a subset by first shuffling the data and then selecting the first 492 records, resulting in a balanced subset with a 1:1 ratio between the two classes.

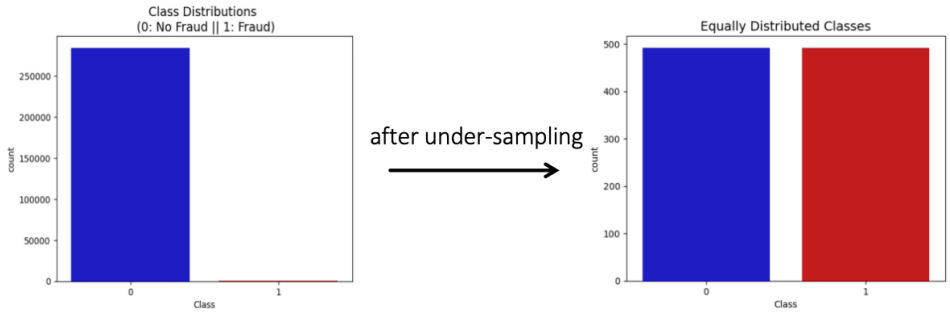


Figure 2: Class Distribution Before and After Undersampling

## 4.1 Correlation Analysis

The correlation matrix can effectively identify which features have a significant impact on fraud detection. However, it is essential to perform correlation analysis using a balanced subset of the data, as the class imbalance in the original dataset may distort the true relationships. Analysis of the correlation matrix from the undersampled dataset reveals distinct patterns in feature relationships with fraudulent transactions. The features V14, V12, and V10 demonstrate significant negative correlations, indicating that lower values of these variables correspond to a higher probability of fraudulent activity. Conversely, the features V4, V11, and V2 exhibit strong positive correlations, suggesting that elevated values in these dimensions are associated with increased fraud risk.

## 4.2 Anomaly Detection

For these highly correlated features with classification, we plotted corresponding boxplots and observed some outliers.

We remove these outliers based on the quartile range method. Firstly, extract the values of features in fraudulent transactions and calculate the quartiles and interquartile range (IQR). IQR is an indicator used in statistics to measure the degree of data dispersion, calculated as follows: IQR=Q3-Q1, where Q3 is the 75th percentile and Q1 is the 25th percentile. Use Tukey's fences method again to determine

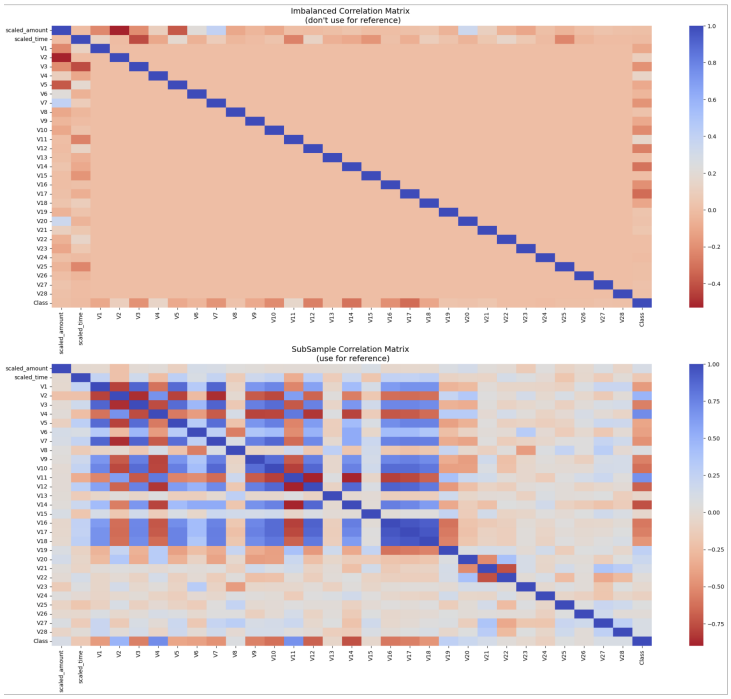


Figure 3: Correlation Heatmaps: Imbalanced Dataset vs Subsampled Dataset

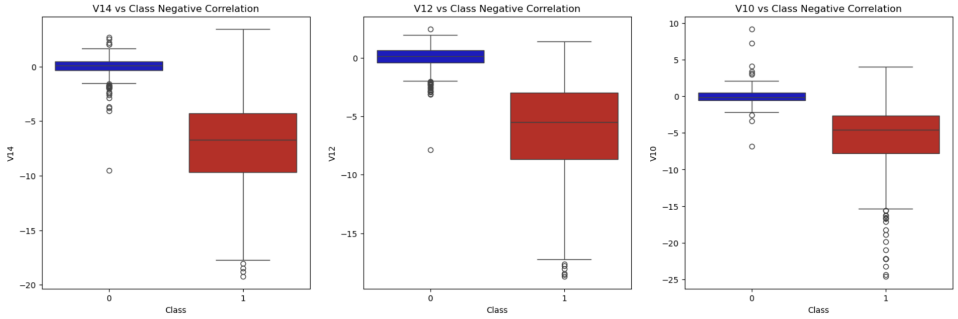


Figure 4: C Feature Negative Correlation with Class

outliers: set the lower limit to Q1-1.5 * IQR and the upper limit to Q3+1.5 * IQR. Afterwards, data exceeding the above range will be identified as outliers and removed. For non transaction fraud data, we did not handle outliers because fraudulent transactions are a minority class, and outliers have a greater impact on the model. In contrast, normal transactions have a large number and contain more natural variations, and we are more concerned about whether the feature distribution of fraudulent transactions is reasonable. By comparing the results of model training, it can also be concluded that the prediction performance is better when only dealing with fraudulent transactions.

## 4.3 Dimensionality Reduction and Clustering

Afterwards, we also attempted to use three dimensionality reduction techniques, t-SNE, PCA, and truncated SVD, to visualize high-dimensional data, observe the distribution patterns of fraudulent and non fraudulent transactions in two-dimensional space, and analyze the dimensionality reduction methods that performed well on this dataset.

It can be seen that for the t-SNE method, its cluster boundaries are relatively clear and can well preserve local features. The overall distribution of PCA data points is sparse, the red clusters are not compact enough, and the two types of boundaries are blurred from a global perspective. The dimensionality reduction visualization results of truncated SVD are similar to PCA, and the extraction of two types of features is not as delicate as t-SNE, but its computational speed is very fast.

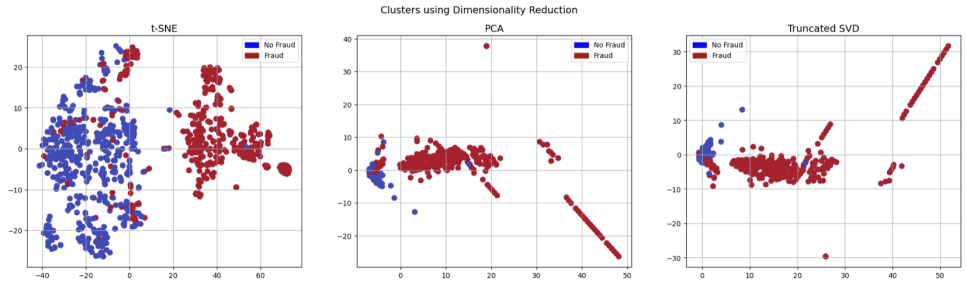


Figure 5: Comparative Visualization of Fraud Detection Patterns Using Dimensionality Reduction Techniques

## 4.4 Classifiers and evaluation indicators

Next, we use four classification models: logistic regression, K-nearest neighbor, support vector machine, and decision tree for training. For the accuracy obtained from 5-fold cross validation, logistic regression performs the best.

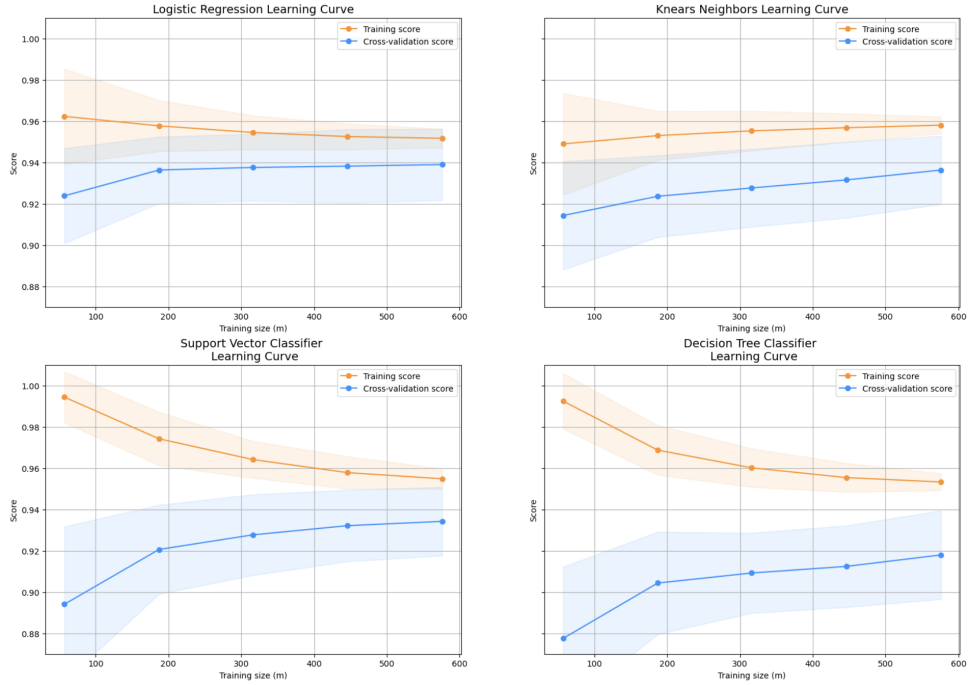


Figure 6: Learning Curve Analysis of Classification Models for Fraud Detection

Through their respective learning curves, it can be seen that the performance of logistic regression on the training set and the validation set has a small gap, indicating that the model has good generalization ability, and the validation score quickly converges and stabilizes with the increase of data volume. The generalization ability, stability, and convergence are all excellent, and the model is reliable. There is always a gap in the performance of K-nearest neighbors on the training and validation sets, and their generalization ability is average. Support vector machine models suffer from severe overfitting in small data volumes, requiring a large amount of data to "calibrate" the overfitting problem. When the data volume is sufficient, the generalization ability improves, but overall it is not as stable and efficient as logistic regression. The decision tree validation score is low, and there is always a significant gap in performance between the training set and the validation set.

## 4.5 Performance Comparison

From the ROC curve ,corresponding AUC values and confusion, it can also be seen that logistic regression performs the best on this dataset.

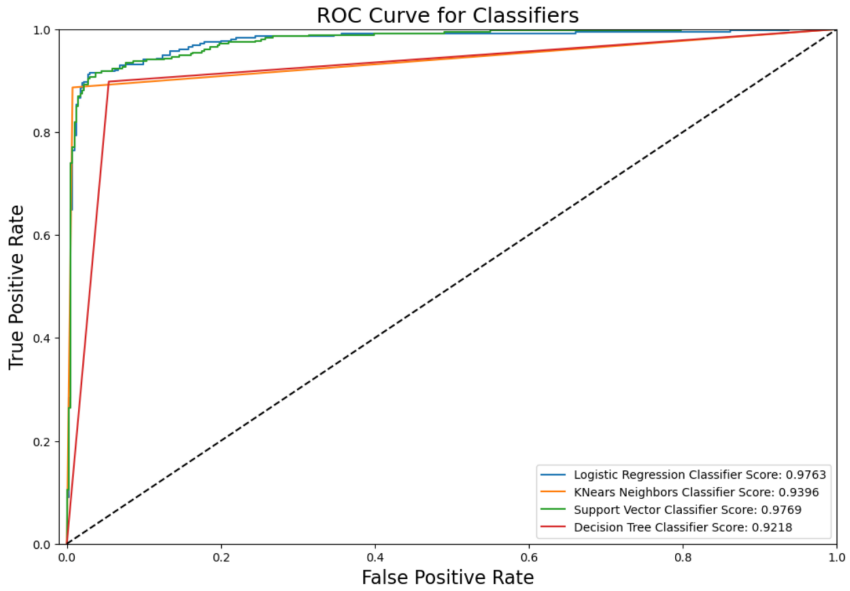


Figure 7: ROC curve for classifiers

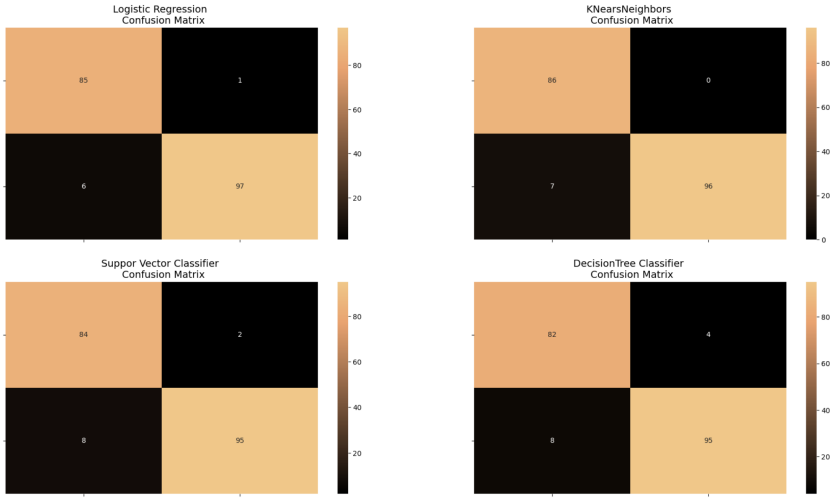


Figure 8: Confusion Matrix for classifiers

# 5 SMOTE Technique (Over-Sampling)

## 5.1 Conceptual Understanding

- **Solving the Class Imbalance**: SMOTE creates synthetic points from the minority class in order to reach an equal balance between the minority and majority class.
- **Location of the synthetic points**: SMOTE picks the distance between the closest neighbors of the minority class, in between these distances it creates synthetic points.
- **Advantage**: More information is retained since we didn't have to delete any rows unlike in random undersampling.
- **Accuracy —— Time Tradeoff**: Although it is likely that SMOTE will be more accurate than random under-sampling, it will take more time to train since no rows are eliminated as previously stated.

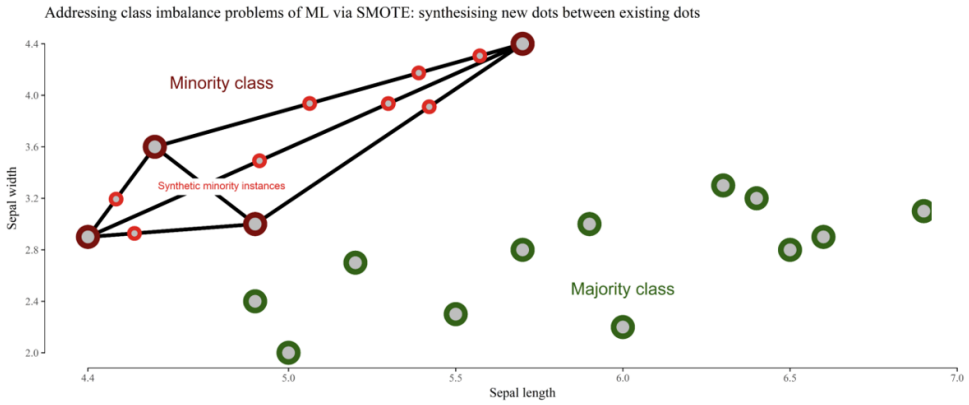


Figure 9: Picture of SMOTE Technique

## 5.2 Implementation cross-validation

### 5.2.1 The Wrong Way

SMOTE should be applied within the training set during cross-validation, not before, to avoid data leakage into the validation set.


Figure 10: the wrong way(left) and the right way(right)

## 5.3 Undersampling vs Oversampling Comparison

Returning to the previously best-performing logistic regression model, when applied to the entire dataset after being trained with random undersampling, its ROC curve appeared excellent. However, when examining the precision-recall curve, we observed that precision remained consistently low regardless of changes in recall. This indicates that among the transactions predicted as fraudulent by the model, very few were actually fraudulent. The high ROC and accuracy scores are largely due to correctly classifying a vast majority of normal transactions, but this does not necessarily mean the model effectively handles imbalanced datasets.

After applying SMOTE oversampling and adjusting the classification threshold uniformly, we plotted the precision-recall curve again. This time, we found that precision significantly improved at lower recall levels. In these cases, we identified optimal classification results and average values across all thresholds. Although the precision improved compared to undersampling, it still remained relatively low overall. Under this tradeoff, in order to reduce the number of falsely flagged legitimate transactions ("false alarms"), we inevitably allow some fraudulent ones to go undetected.

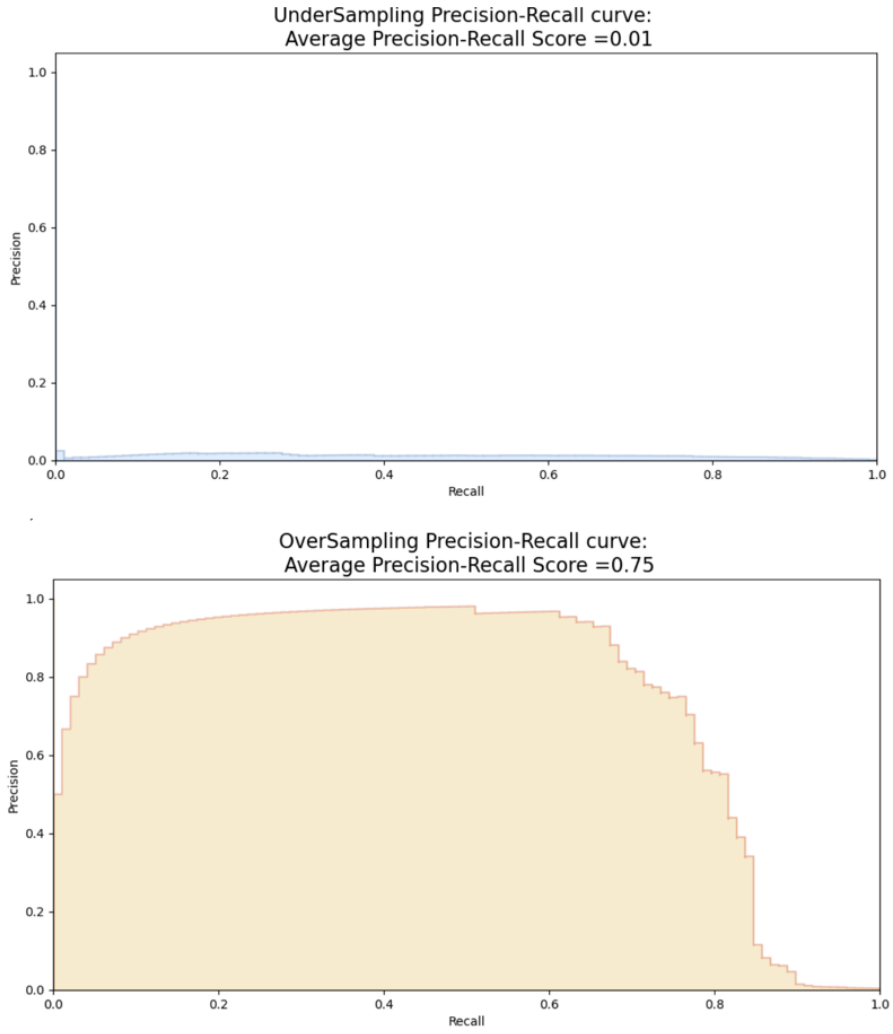


Figure 11: the precision-recall curve of undersampling(up) and oversampling(down)

# 6 Simple Neural Network

In this section we will implement a simple Neural Network (with one hidden layer) in order to see which of the two logistic regressions models we implemented in the (undersample or oversample(SMOTE)) has a better accuracy for detecting fraud and non-fraud transactions.

## 6.1 Structure and Configuration

**Dataset**:In this final phase of testing we will fit this model in both the random undersampled subset and oversampled dataset (SMOTE) in order to predict the final result using the original dataframe testing data.

**Neural Network Structure**:As stated previously, this will be a simple model composed of one input layer (where the number of nodes equals the number of features) plus bias node, one hidden layer with 32 nodes and one output node composed of two possible results 0 or 1 (No fraud or fraud).

**Other characteristics**:The learning rate will be 0.001, the optimizer we will use is the AdamOptimizer, the activation function that is used in this scenario is "Relu" and for the final outputs we will use sparse categorical cross entropy, which gives the probability whether an instance case is no fraud or fraud (The prediction will pick the highest probability between the two.)

### 6.2 Results and Analysis

This is the confusion matrix obtained after training. In my opinion, the results under the undersampling condition are better because the model can more accurately identify fraudulent transactions while maintaining a relatively low false positive rate. Although 1,839 may seem like a large number, it is small compared to the total volume of normal transactions.

Oversampling also has clear advantages — the recall for normal transactions is very high, with only 0.02

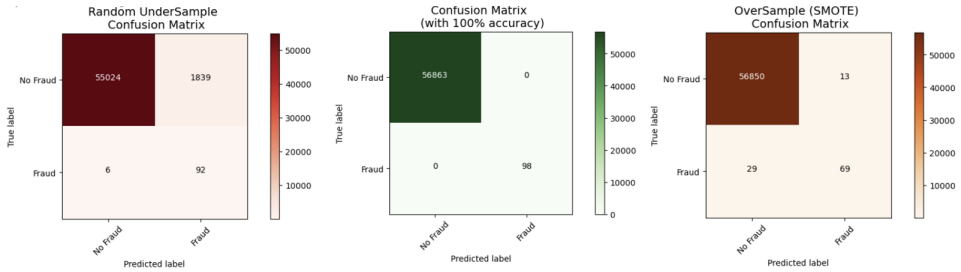


Figure 12: the confusion matrix of undersampling(left) truth(middle) and oversampling(right)

This aspect is crucial. Imagine you are a cardholder and, after making a legitimate purchase, your credit card gets frozen because the bank's algorithm mistakenly identified the transaction as fraudulent. Therefore, we should not only focus on detecting fraud but also emphasize the importance of correctly classifying non-fraudulent transactions.

## 7 Conclusion

Implementing SMOTE on our imbalanced dataset helped address the class imbalance (with far more non-fraudulent than fraudulent transactions). Nevertheless, the neural network trained on the oversampled dataset sometimes predicted fewer correct fraud cases compared to the model trained on the undersampled data.

Through experiments combining both undersampling and oversampling techniques with different classifiers, we clearly observed the importance of balancing precision and recall.