MONEY LAUNDERING DETECTION OF SUSPICIOUS TRANSACTION USING MACHINE LEARNING ALGORITHM

NUR ADRIANA BATRISYIA BINTI MOHD SUBRI

UNIVERSITI TEKNOLOGI MALAYSIA

CHAPTER 4

INITIAL RESULTS

4.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) involves statistical analysis and visualization techniques to identify the patterns, trends, and understand the relationship between features and target variables.

4.1.1 Identify Min, Max, and Mean for Laundering and Normal Transactions

Based on Figure 4.1, the maximum amount of money involves in laundering transactions (12,618,498.40) is significantly higher than normal transactions (999,962.19). The mean for laundering transactions is also higher than normal transactions. Furthermore, both transactions have extremely small minimum amount of money where laundering transactions (15.82) is slightly higher than normal transactions (3.73). Therefore, this chart highlights that laundering transactions often involves extreme values which may be a key indicator to identify suspicious transactions in money laundering activities.

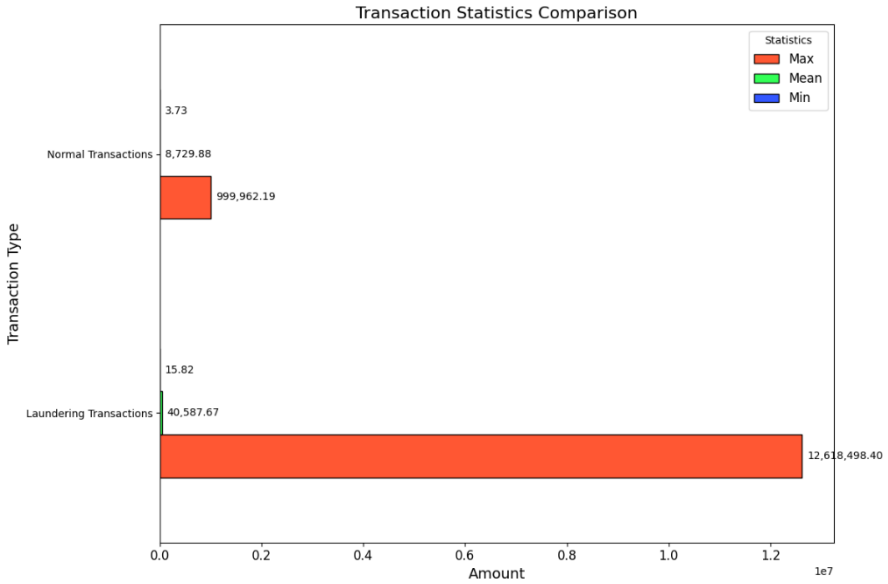

Figure 4.1: Comparison of Transaction Statistics between Transaction Type

4.1.2 Identify Most Frequent Typologies for Laundering Transactions

Bar chart in Figure 4.2 illustrates that Over Invoicing, Fan Out, and Single Large are the least frequent typologies of laundering transactions. On the other hand, Structuring is the most frequent typology for laundering transactions followed by Cash Withdrawal, Deposit Send, and Smurfing. These four typologies are the dominant typologies as they occur significantly more common than others. It is important to identify the characteristics of dominant typologies for a more targeted investigations to improve the efficiency of money laundering detection systems.

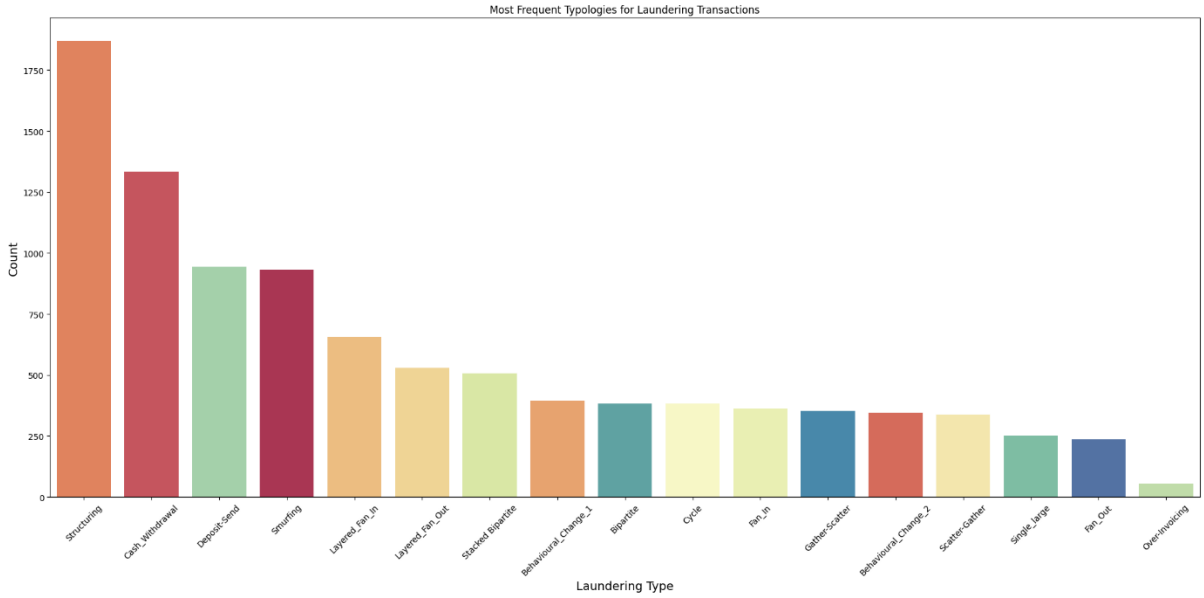

Figure 4.2: Most Frequent Typologies for Laundering Transactions

4.1.3 Identify Most Frequent Payment Types for Laundering Transactions

Pie chart in Figure 4.3 illustrates that the most common money laundering payment method is Cross-border (26.6%) followed by Cash Deposit (14.2%) and Cash Withdrawal (13.5%) emphasizing their significant role in money laundering activities. Meanwhile, ACH (11.7%), Credit Card (11.5%), Debit Card (11.4%) and Cheque (11.0%) have relatively similar proportions. This pie chart highlights the various types of payment method used in money laundering with cross-border transactions as the most favourite method among launderers.

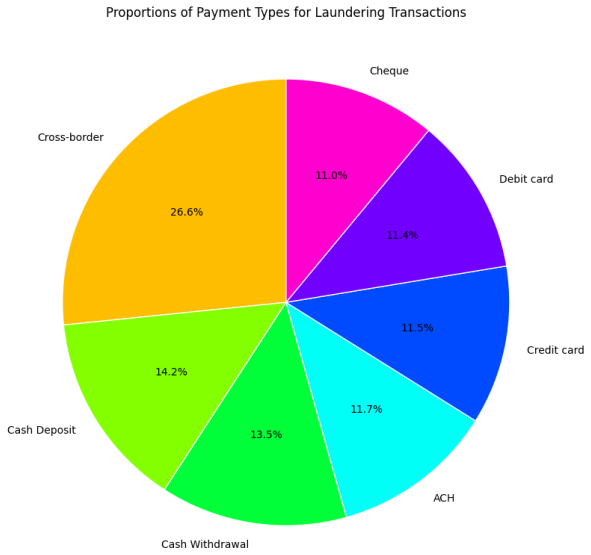

Figure 4.3: Most Frequent Payment Types for Laundering Transactions

4.1.4 Identify the High-Risk Bank Locations

These charts in Figure 4.4 and Figure 4.5 shows the distribution of laundering transactions by sender and receiver bank locations. Both charts depict that UK overwhelmingly leads in the laundering transactions as sender and receiver location. This insight highlights UK as the most high-risk bank locations as it seems to be a central hub for both sending and receiving illicit money from laundering transactions.

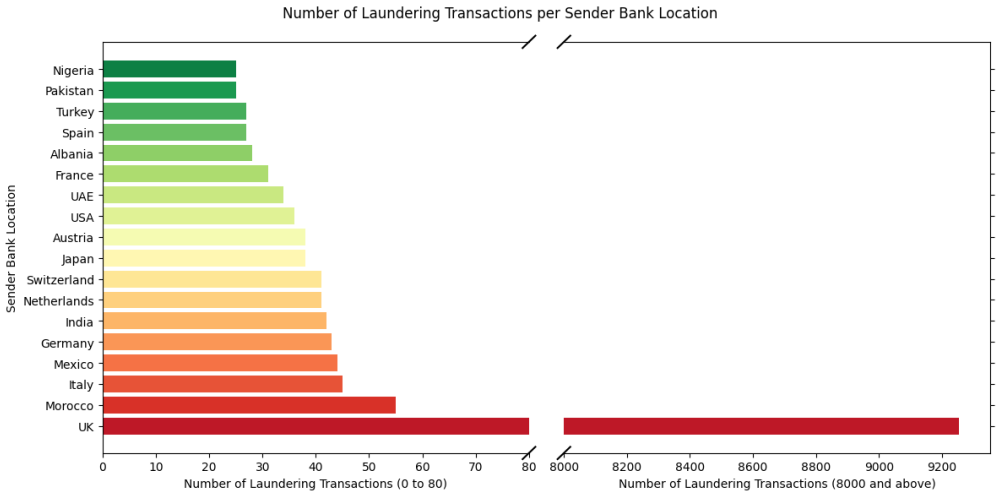

Figure 4.4: Number of Laundering Transactions per Sender Bank Location

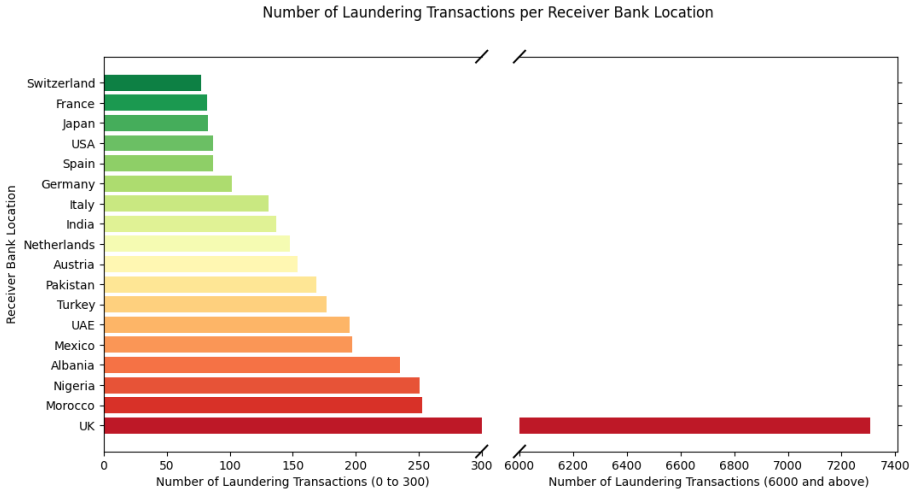

Figure 4.5: Number of Laundering Transactions per Receiver Bank Location

4.1.5 Identify Monthly Transaction Frequency and Average Laundering Amount by Transaction Type

Based on charts in Figure 4.6 and Figure 4.7, it indicates that the frequency of laundering transactions is less common than normal transactions, where laundering transactions occur between 694 to 1024 times per month while normal transactions occur hundreds of thousands of times every month. However, the average laundering amount exhibits sharp fluctuations with significantly higher values compared to average normal transactions that remains low and stable. This sharp contrast in frequency and amount emphasize that laundering transactions occurrence are rare but usually involve larger amounts of money.

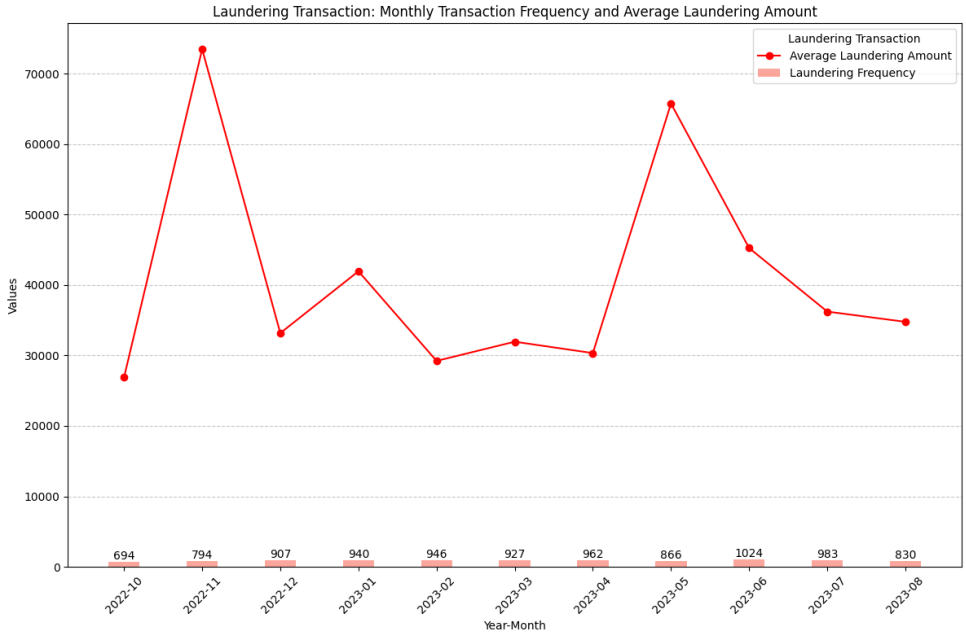

Figure 4.6: Monthly Laundering Transactions Frequency and Average Laundering Amount

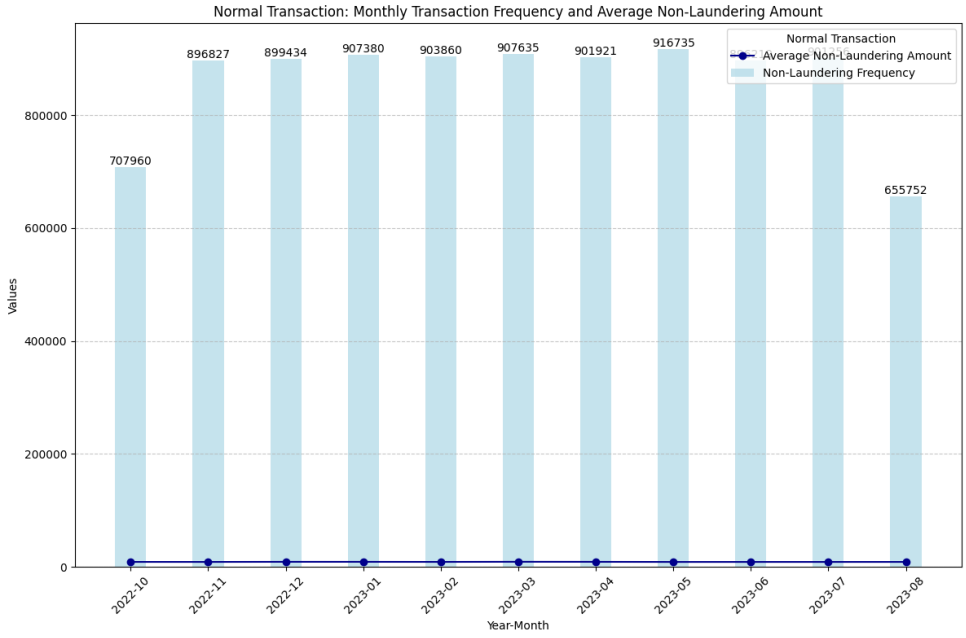

Figure 4.7: Monthly Normal Transactions Frequency and Average Normal Amount

4.2 Feature Engineering

Feature engineering is the process of transforming features in raw data to improve the accuracy and efficiency of machine learning models. It is because the success of a machine learning models depends on the quality of features that are used to train the models. For this project, three techniques of feature engineering including Log Transformation, Label Encoding, and Standard Scaling are used to modify the selected dataset features to make it more usable for machine learning model.

4.2.1 Log Transformation

Log Transformation is applied to feature ‘Amount’ as the data distribution is highly skewed to the right with skewness value of 102.16 as per Figure 4.8. It indicates that most ‘Amount’ values are small but there are some very large outliers exist in the dataset. After applying the log transformation, the skewness has reduced significantly as per Figure 4.9 with value of -1.01 which is near to 0. It is now much more balanced than the original distribution making the data more suitable for modelling.

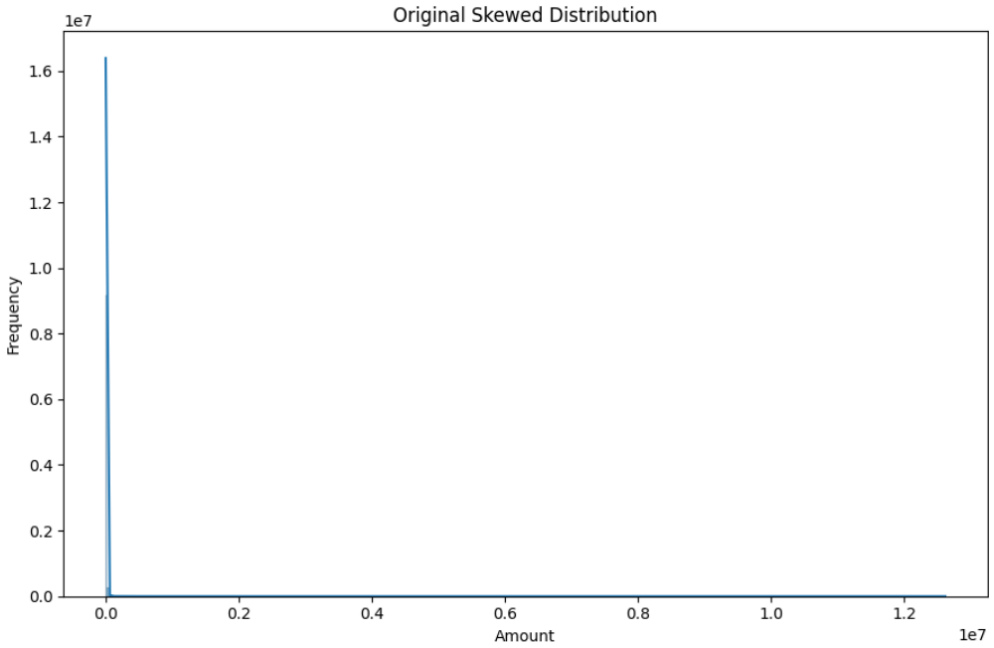

Figure 4.8: Original Skewed Distribution of ‘Amount’ Feature

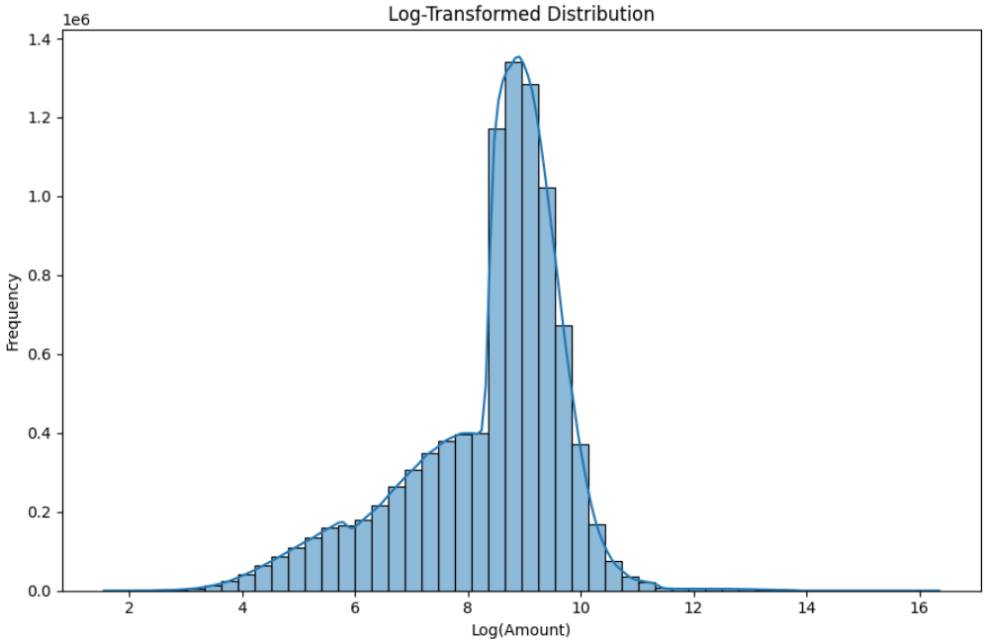

Figure 4.9: Log-Transformed Distribution of 'Amount' Feature

Table 4.1 below shows the comparison between the 'Amount' value before and after log transformation.

Table 4.1: Transactions Statistics Before and After Log Transformation

	Statistics	**Before Log Transformation**	**After Log Transformation**
Laundering Transactions	**Max**	12,618,498.40	16.35
	Min	40,587.67	8.34
	Mean	15.82	2.82
Normal Transactions	**Max**	999,962.19	13.82
	Mean	8,729.88	8.35
	Min	3.73	1.55

4.2.2 Label Encoding

Label Encoding is applied to ten categorical features in this dataset which are 'Sender_account', 'Receiver_account', 'Payment_currency', 'Received_currency', 'Sender_bank_location', 'Receiver_bank_location', 'Payment_type', 'Hour', 'Year-Month', and 'Day'. It is important to perform label encoding because machine learning models usually require numerical input and cannot work with categorical data directly. Figure 4.10 shows the code used to transform categorical features into numerical labels.

```
categorical_cols = ['Sender_account', 'Receiver_account', 'Payment_currency', 'Received_currency',
                    'Sender_bank_location', 'Receiver_bank_location', 'Payment_type',
                    'Hour', 'Year-Month', 'Day']

for col in categorical_cols:
    encoder = preprocessing.LabelEncoder()
    df[col] = encoder.fit_transform(df[col])
```

Figure 4.10: Label Encoding to Transform Categorical Features

4.2.3 Standard Scaling

Standard Scaling is applied to ‘Amount’ features to transform the data into standard normal distribution with mean 0 and standard deviation 1. It is important to perform standard scaling so that the data are on similar scale and works well with machine learning algorithms that are sensitive to feature scaling. Figure 4.11 shows the code used to do standard scaling.

```
numerical_cols = ['Amount']

scaler = preprocessing.StandardScaler()
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])
```

Figure 4.11: Standard Scaling for Numerical Feature

4.3 Split Train-Test Dataset

Before applying machine learning algorithm to the dataset, it is crucial to divide the dataset into two subsets which are training set and test set. Training set is for the machine learning algorithm learns the patterns and relationship in the dataset, while testing set is to evaluate how well the machine learning algorithm predict the outcomes based on unseen data. For this project, 70% of the data is used for training and 30% is used for testing. Table 4.2 shows the size of training set and test set use for this project.

Table 4.2: Size of Training Set and Testing Set

Training Set		**Testing Set**	
x-train:	(6653396, 11)	x-test:	(2851456, 11)
y-train:	(6653396, 1)	y-test:	(2851456, 1)

4.4 Handling Class Imbalance using Random Under Sampling

Figure 4.12 illustrates a serious class imbalance in the dataset in which the proportion of suspicious transactions is only 0.1% compared to normal transactions. There are only 9,873 laundering transactions compared to 9,494,979 normal transactions. Therefore, this project used Random Under Sampling (RUS) techniques to address the class imbalance. Training on a balanced dataset is important to avoid ‘overfitting’ (training datasets produced best results while test datasets have poor performance) or ‘underfitting’ (both training and test datasets has poor results).

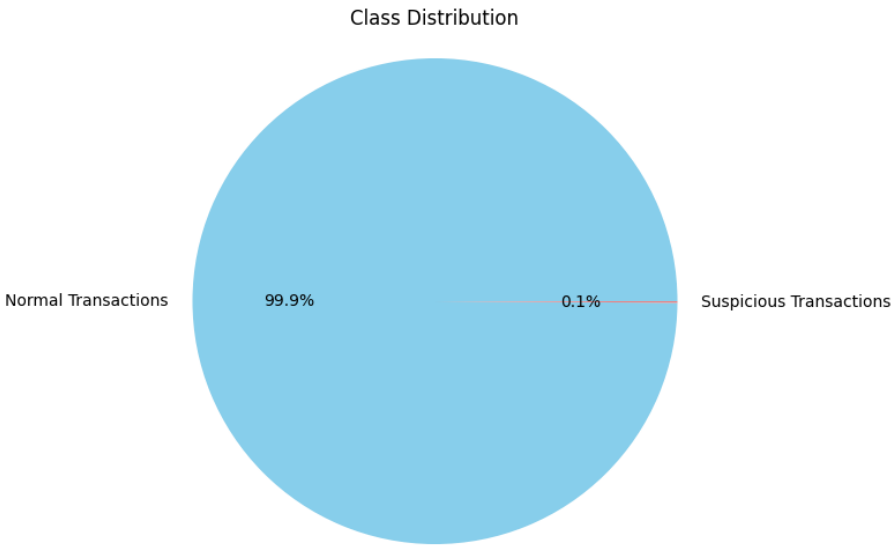

Figure 4.12: Original Class Distribution on Whole Dataset

It is important to note that RUS is only applied to training dataset to keep the testing set with original class distribution to simulate real-world scenarios. RUS technique creates a balanced class of training dataset by reducing the normal transactions samples to match the size of laundering transactions class. Due to reduction in normal transactions class, the computational load also decreases as the dataset size decreases hence optimize the training within a shorter time. Figure 4.13 depicts the new class distribution on training dataset after performing RUS and Table 4.3 shows the total samples of normal and laundering transactions before and after performing RUS.

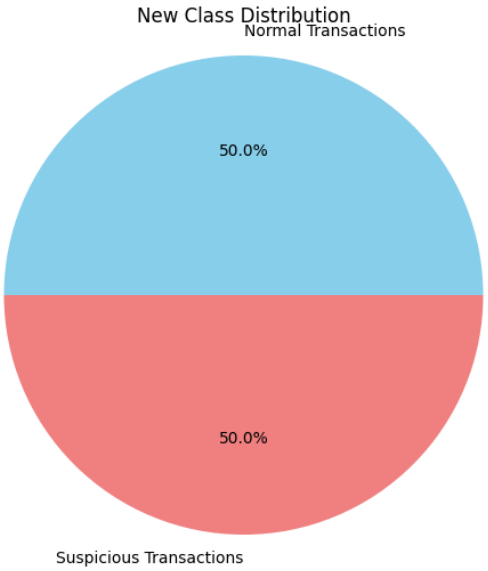

Figure 4.13: New Class Distribution for Training Set after Performing RUS

Table 4.3: Total Samples of Normal and Laundering Transactions on Training Set Before and After RUS

	Training Set Before RUS	**Training Set After RUS**
Normal Transactions	6,646,428	6,968
Laundering Transactions	6,968	6,968