Synthetic data generation using Variational Autoencoder

Raj Popat
CS23MTECH14009

Vaibhav Falgun Shah
AI22MTECH02007

Sreyash Mohanty
CS23MTECH14015

Yash Shukla
CS23MTECH14018

Somya Kumar
SM23MTECH11010

Completed as a assignment of the coursework on Fraud Analytics Utilizing Predictive and Social Network Techniques (CS6890).

1. Problem Statement

The goal of this assignment is to create a Variational Autoencoder-based model that generates synthetic credit card transaction data. The model will be trained using an existing credit card transaction dataset, and the synthetic data produced will closely mimic the statistical features and distribution of the original data. The model's performance will be evaluated by comparing the distributions of each column in the real and synthetic datasets and calculating relevant metrics.

The card transaction.v1.csv dataset contains a record of credit card transactions, including details such as transaction amounts, merchant information, and fraud status.

The objective of this assignment is to develop a synthetic data generation model using a Variational Autoencoder (VAE) to augment the dataset for improving the performance of fraud detection algorithms. The VAE will learn the underlying distribution of legitimate transactions from the original dataset and generate synthetic transactions that closely resemble real transactions. This augmented dataset will help in training fraud detection models more effectively by providing a larger and more diverse set of data points.

Our approach would be preprocessing the card transaction.v1.csv dataset to prepare it for training the VAE. and designing and training a VAE architecture suitable for generating synthetic credit card transactions. Generating synthetic transactions using the trained VAE. Evaluating the quality of the synthetic transactions in terms of their similarity to real transactions. Augmenting the original dataset with the synthetic transactions to create a larger and more balanced dataset for training fraud detection models. Assessing the impact of the augmented dataset on the performance of fraud detection algorithms. By addressing these tasks, this project aims to enhance fraud detection capabilities by leveraging the power of generative models like VAEs to create realistic synthetic data for training purposes

2. Description of the dataset

The card 'transaction.v1.csv' dataset provides a detailed record of credit card transactions, offering insights into various aspects of each transaction. Each entry includes a unique user identifier and a corresponding card identifier, allowing for the tracking of individual transaction histories. The dataset includes the specific date and time (year, month, day, and time) of each transaction, enabling temporal analysis of transaction patterns. The transaction amount is recorded for each entry, providing information about the financial value of the transaction.

Additionally, the dataset includes a field indicating whether a chip was used in the transaction, which can be important for analyzing transaction security measures. The dataset also includes details about the merchant, such as the merchant name, merchant city, and merchant state, providing information about the location of the transaction. The merchant's zip code is also included, allowing for more granular geographic analysis.

Moreover, the dataset includes the Merchant Category Code (MCC) for each transaction, which categorizes the type of merchant involved in the transaction. This information can be valuable for understanding spending patterns across different merchant categories. The dataset also includes a field indicating if there were any errors during the transaction, providing insights into transaction reliability and potential issues.

Lastly, the dataset includes a field indicating if the transaction is fraudulent or not. This field is crucial for fraud detection and prevention efforts, as it allows for the identification of potentially fraudulent transactions. Overall, the card transaction.v1.csv dataset provides a comprehensive and detailed view of credit card transactions, making it suitable for various analytical purposes, including synthetic data generation using a Variational Autoencoder (VAE).

3. Algorithm Used

3.1. Variational Autoencoder

VAEs extend autoencoders by incorporating a probabilistic approach to the latent space. The encoder learns the parameters of a probability distribution representing the latent space.

The reparameterization trick allows gradients to backpropagate through the sampling step, enabling training with gradient-based optimization. The decoder then reconstructs the input data from the samples drawn from the learned latent distribution.

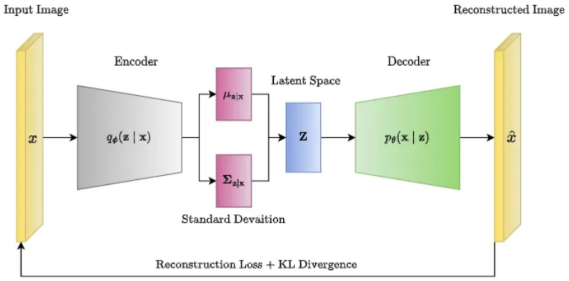

Input Image
Reconstructed Image
Encoder
Latent Space
Decoder
x
$q_\phi(\mathbf{z} \mid \mathbf{x})$
$\mu_{\mathbf{z}|\mathbf{x}}$
$\Sigma_{\mathbf{z}|\mathbf{x}}$
$\mathbf{z}$
$p_\theta(\mathbf{x} \mid \mathbf{z})$
$\hat{x}$
Standard Devaition
Reconstruction Loss + KL Divergence

Figure 1. Process of Variational Autoencoder Approach

In this Architecture, Encoder layers to output the mean and standard deviation of the latent space distribution. Decoder layers to map the sampled latent representation back to the original input size.

Encoder :

$$\mu, \sigma = f(X) \tag{1}$$

Reparameterization Trick :

$$Z = \mu + \sigma \cdot \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0,1) \tag{2}$$

Decoder :

$$\hat{X} = g(Z) \tag{3}$$

Reconstruction Loss: MSE between the input and reconstructed data.

$$\mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^{N} \left(X_i - \hat{X}_i \right)^2 \tag{4}$$

KL Divergence: Divergence between the learned latent distribution and a standard normal distribution.

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{j=1}^{D} \left(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2 \right) \tag{5}$$

Total Loss:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{KL}} \tag{6}$$

Training Procedure: Train the VAE on non-fraudulent data to learn a probabilistic latent representation of normal patterns.

Algorithm 1 Variational Autoencoder (VAE) Algorithm

```
**Training Phase:**
Initialize encoder weights $W_e$ and biases $b_c$.
Initialize decoder weights $W_d$ and biases $b_d$.
**for** each training step **do**
    $\mu, \sigma = f(X; W_c, b_e) = \text{Encoder}(X)$    ▷ $\mu$ and $\sigma$ represent the mean and standard deviation of the latent space.
    $Z = \mu + \sigma \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(0,1)$    ▷ Sample latent representation using the reparameterization trick.
    $\hat{\boldsymbol{X}} = g(Z; W_d, b_d) = \text{Decoder}(Z)$    ▷ $\hat{X}$ is the reconstructed input.
    Compute the reconstruction loss and KL divergence:
    $\mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{X}_i)^2$
    $\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{j=1}^{D} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2)$
    Combine the losses and backpropagate:
    $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{KL}}$    ▷ Update weights $W_e, b_e, W_d, b_d$.
**end for**

**Reconstruction Phase:**
**for** each data sample $X$ **do**
    $\mu, \sigma = f(X; W_c, b_c)$    ▷ Encode to get distribution parameters.
    $Z = \mu + \sigma \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(0,1)$    ▷ Sample from the latent space.
    $\hat{X} = g(Z; W_d, b_d)$    ▷ Decode to reconstruct the input.
    Compute the reconstruction error:
    $\epsilon = \|X - \hat{X}\|^2$    ▷ Calculate the anomaly score using the combined reconstruction error and KL divergence.
    Set a threshold to classify $X$ as an anomaly if the score exceeds the threshold.
**end for**
```

4. Results

This research seeks to enhance credit card transaction datasets while maintaining data privacy and secrecy by utilizing Variational Autoencoder-based synthetic data generation. The project's goal is to evaluate the efficacy of VAEs in generating synthetic data that closely matches the statistical features of real data, allowing for a variety of downstream applications such as model training and analysis.

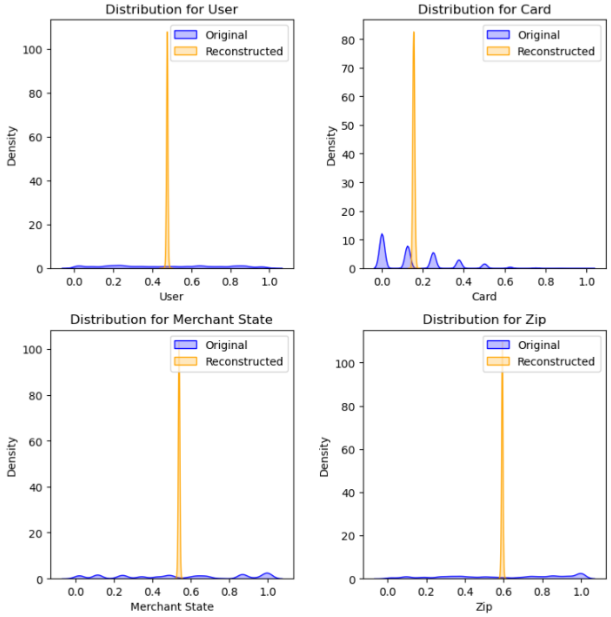


Figure 2. All nodes are arranged in the descending order of their scores and bad nodes are highlighted.

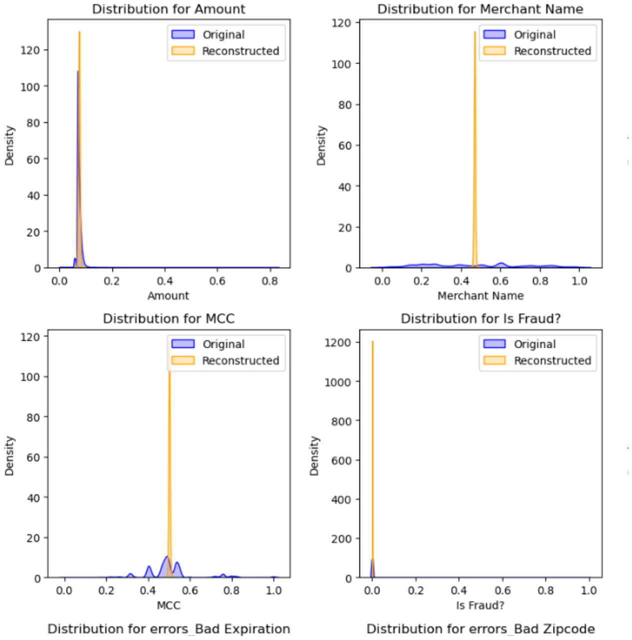


Figure 3. Reconstruction plots

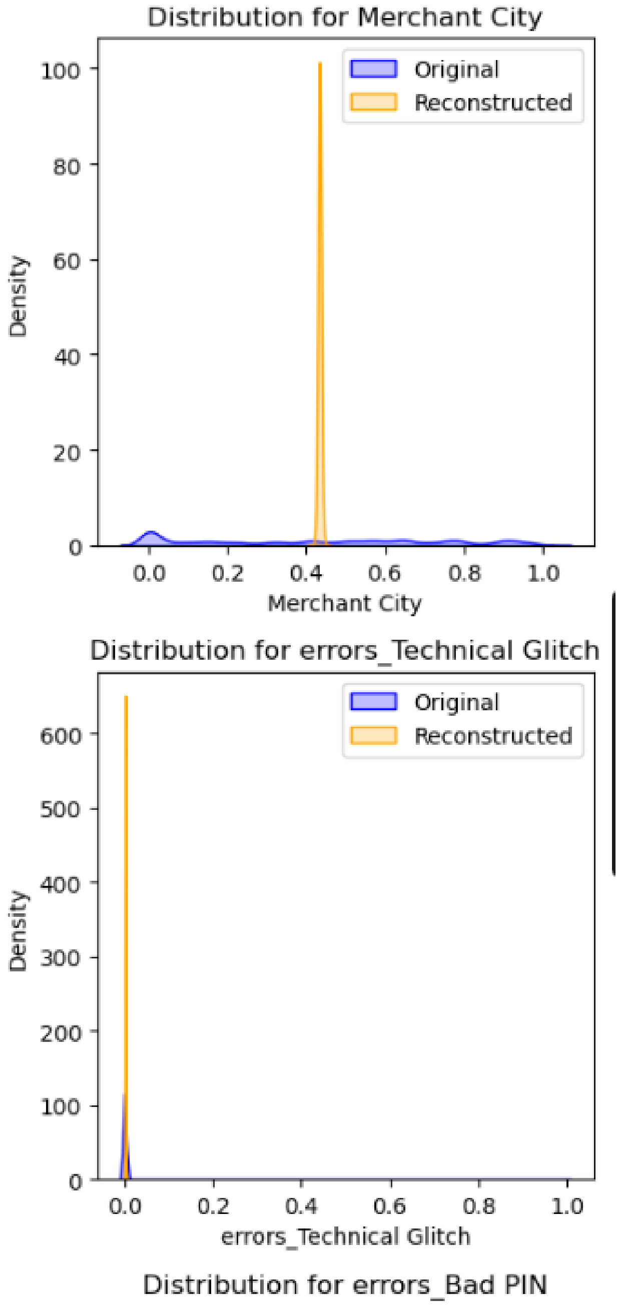


Figure 4. Reconstruction Image

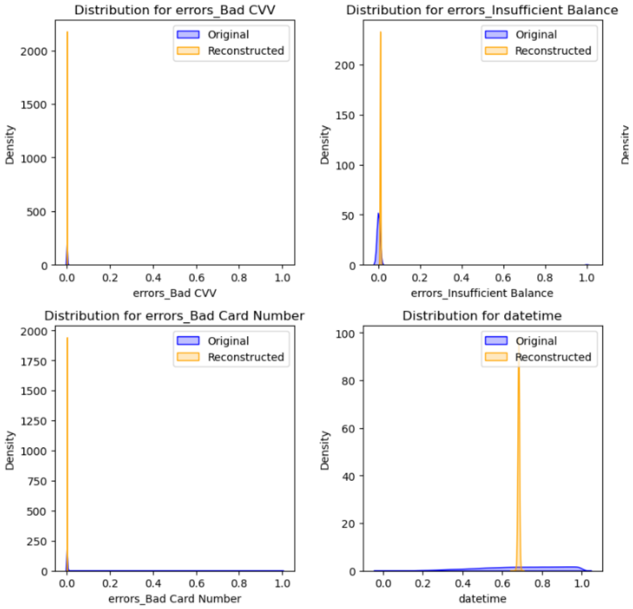
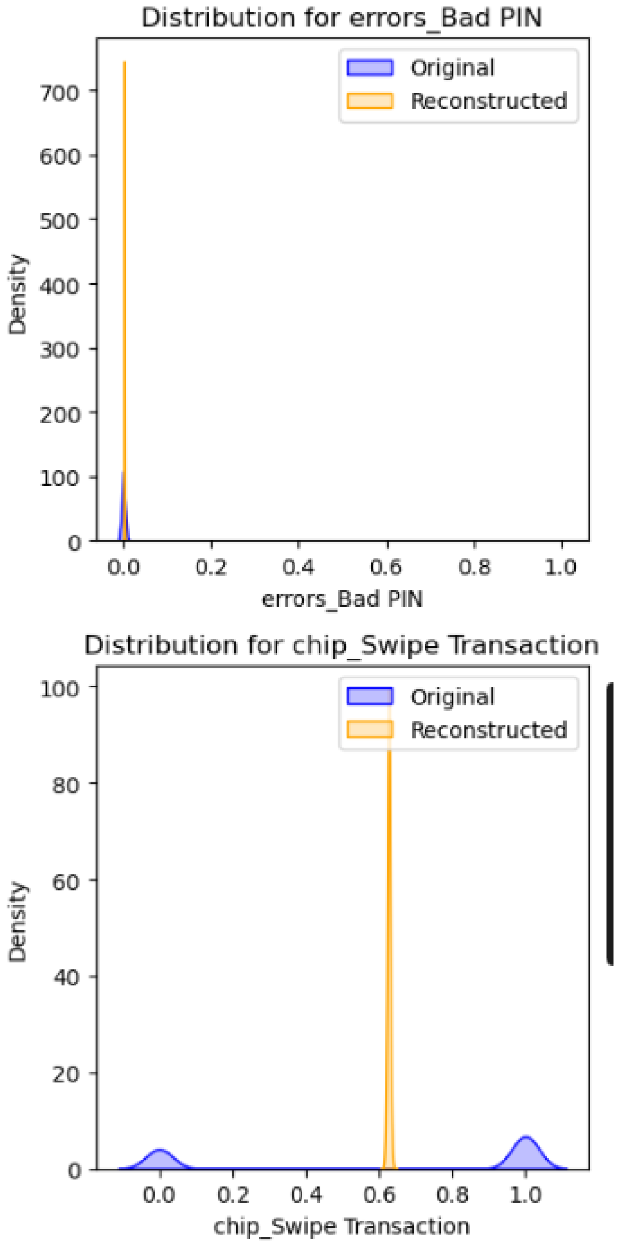


Figure 5. Reconstruction Image



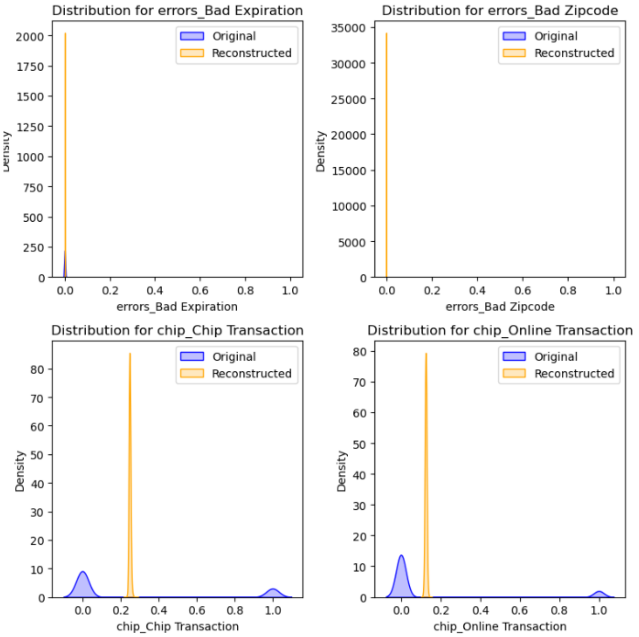
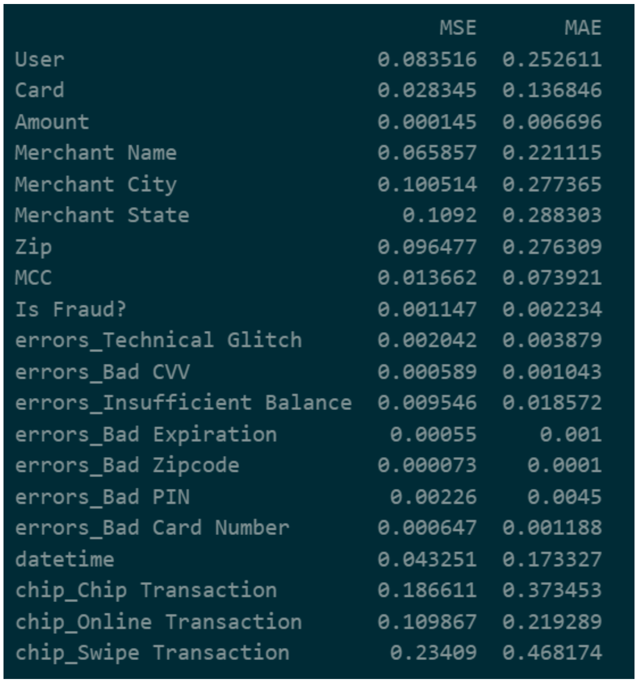


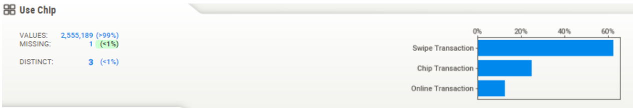
Figure 6. Reconstruction Image

Figure 7. Reconstruction Image



```
                                  MSE       MAE
User                         0.083516  0.252611
Card                         0.028345  0.136846
Amount                       0.000145  0.006696
Merchant Name                0.065857  0.221115
Merchant City                0.100514  0.277365
Merchant State                 0.1092  0.288303
Zip                          0.096477  0.276309
MCC                          0.013662  0.073921
Is Fraud?                    0.001147  0.002234
errors_Technical Glitch      0.002042  0.003879
errors_Bad CVV               0.000589  0.001043
errors_Insufficient Balance  0.009546  0.018572
errors_Bad Expiration         0.00055     0.001
errors_Bad Zipcode           0.000073    0.0001
errors_Bad PIN                0.00226    0.0045
errors_Bad Card Number       0.000647  0.001188
datetime                     0.043251  0.173327
chip_Chip Transaction        0.186611  0.373453
chip_Online Transaction      0.109867  0.219289
chip_Swipe Transaction        0.23409  0.468174
```

Figure 8. Reconstruction Image

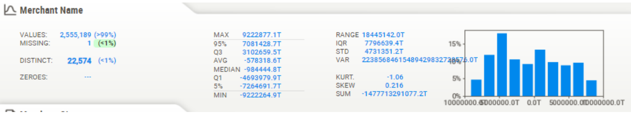


Use Chip

VALUES: 2,555,189 (>99%)
MISSING: 1 (<1%)

DISTINCT: 3 (<1%)

Figure 9. Card Feature Analysis



Merchant Name

VALUES: 2,555,189 (>99%)
MISSING: 1 (<1%)

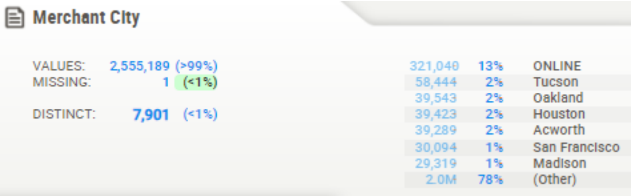
DISTINCT: 22,574 (<1%)

ZEROES: ---

MAX	9222877.1T
95%	7081428.7T
Q3	3102659.5T
AVG	-578318.6T
MEDIAN	-984444.8T
Q1	-4693979.9T
5%	-7264691.7T
MIN	-9222264.9T

RANGE	18445142.0T
IQR	7796639.4T
STD	4731351.2T
VAR	22385684615489429832728576.0T
KURT.	-1.06
SKEW	0.216
SUM	-1477713291077.2T

Figure 10. Merchant Name Feature Analysis



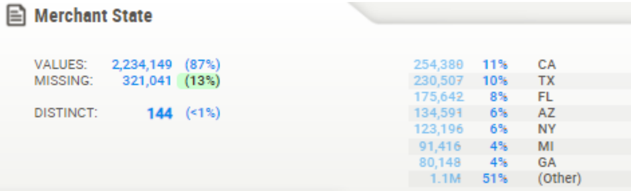
Merchant City

VALUES: 2,555,189 (>99%)
MISSING: 1 (<1%)

DISTINCT: 7,901 (<1%)

321,040	13%	ONLINE
58,444	2%	Tucson
39,543	2%	Oakland
39,423	2%	Houston
39,289	2%	Acworth
30,094	1%	San Francisco
29,319	1%	Madison
2.0M	78%	(Other)

Figure 11. Merchant City Feature Analysis



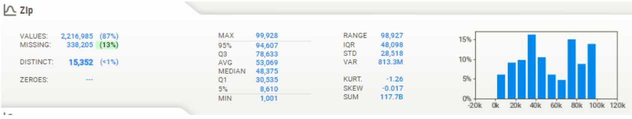
Merchant State

VALUES: 2,234,149 (87%)
MISSING: 321,041 (13%)

DISTINCT: 144 (<1%)

254,380	11%	CA
230,507	10%	TX
175,642	8%	FL
134,591	6%	AZ
123,196	6%	NY
91,416	4%	MI
80,148	4%	GA
1.1M	51%	(Other)

Figure 12. Merchant State Feature Analysis



Zip

VALUES: 2,216,985 (87%)
MISSING: 338,205 (13%)

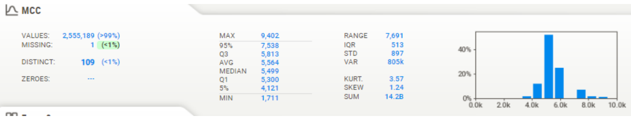
DISTINCT: 15,352 (<1%)

ZEROES: ---

MAX	99,928
95%	94,607
Q3	78,633
AVG	53,069
MEDIAN	48,375
Q1	30,535
5%	8,610
MIN	1,001

RANGE	98,927
IQR	48,098
STD	28,518
VAR	813.3M
KURT.	-1.26
SKEW	-0.017
SUM	117.7B

Figure 13. Day Feature Analysis



MCC

VALUES: 2,555,189 (>99%)
MISSING: 1 (<1%)

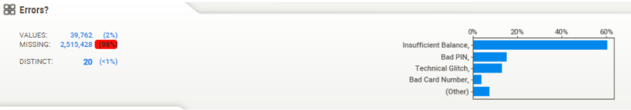
DISTINCT: 109 (<1%)

ZEROES: ---

MAX	9,402
95%	7,538
Q3	5,813
AVG	5,564
MEDIAN	5,499
Q1	5,300
5%	4,121
MIN	1,711

RANGE	7,691
IQR	513
STD	897
VAR	805k
KURT.	3.57
SKEW	1.24
SUM	14.2B

Figure 14. MCC Feature Analysis

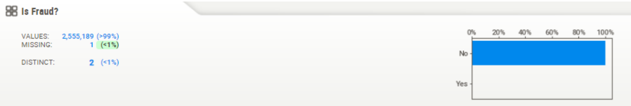


Errors?

VALUES: 39,762 (2%)
MISSING: 2,515,428 (98%)

DISTINCT: 20 (<1%)

Figure 15. Errors? Feature Analysis



Is Fraud?

VALUES: 2,555,189 (>99%)
MISSING: 1 (<1%)

DISTINCT: 2 (<1%)

Figure 16. Is Fraud? Feature Analysis

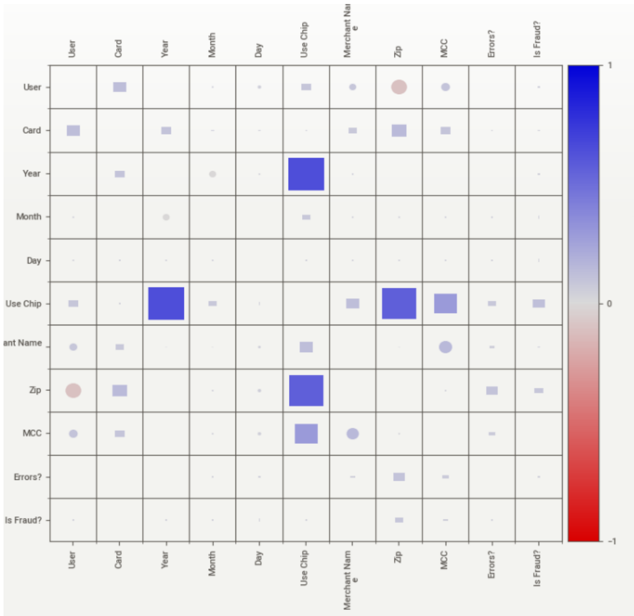


Figure 17. Association Feature Analysis

References :

1] Hamilton, W.L., Ying, R. and Leskovec, J., 2017. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584.