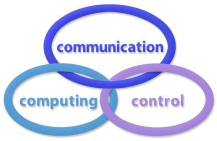
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 18, Issue: 3, Month: June, Year: 2023
Article Number: 5433, https://doi.org/10.15837/ijccc.2023.3.5433



CCC Publications

Categorical Feature Encoding Techniques for Improved Classifier Performance when Dealing with Imbalanced Data of Fraudulent Transactions

D. Breskuvienė, G. Dzemyda

Dalia Breskuvienė*
Data Science and Digital Technologies Institute
Vilnius University, Lithuania
Akademijos g. 4, Vilnius 08412, Lithuania
dalia.breskuviene@mif.vu.lt
*Corresponding author: dalia.breskuviene@mif.vu.lt

Gintautas Dzemyda
Data Science and Digital Technologies Institute
Vilnius University, Lithuania
Akademijos g. 4, Vilnius 08412, Lithuania
gintautas.dzemyda@mif.vu.lt

Abstract

Fraudulent transaction data tend to have several categorical features with high cardinality. It makes data preprocessing complicated if categories in such features do not have an order or meaningful mapping to numerical values. Even though many encoding techniques exist, their impact on highly imbalanced massive data sets is not thoroughly evaluated.

Two transaction datasets with an imbalance lower than 1% of frauds have been used in our study. Six encoding methods were employed, which belong to either target-agnostic or target-based groups. The experimental procedure has involved the use of several machine-learning techniques, such as ensemble learning, along with both linear and non-linear learning approaches.

Our study emphasizes the significance of carefully selecting an appropriate encoding approach for imbalanced datasets and machine learning algorithms. Using target-based encoding techniques can enhance model performance significantly. Among the various encoding methods assessed, the James-Stein and Weight of Evidence (WOE) encoders were the most effective, whereas the CatBoost encoder may not be optimal for imbalanced datasets. Moreover, it is crucial to bear in mind the curse of dimensionality when employing encoding techniques like hashing and One-Hot encoding.

Keywords: imbalanced data, classifier, feature encoding, high-cardinality, fraud detection.

1 Introduction

Financial fraud is a significant issue for businesses and individuals alike, with losses amounting to billions of euros each year. The negative effect of fraud is felt across all sectors of every country and

can significantly lower the overall quality of life. Fraud has clear economic impacts, including reduced financial stability for private enterprises, decreased quality of public services, diminished disposable income for individuals, and lessened resources for philanthropic organizations. The leading organizations fighting fraud are financial institutions such as banks. They are using multiple approaches to detect and prevent financial crime. Machine learning techniques are often employed in fraud detection since they have shown promising results in detecting such transactions [1], [22], [33]. In this case, however, we face a significant challenge due to the imbalanced nature of the data. Most transactions are legitimate, while fraudulent transactions are rare, resulting in imbalanced datasets that negatively impact classifier performance. In the real world, imbalanced data problems are found not only in transactional data but also in cyber-security [9], churn prediction [8], and even protein classification [34], etc. Another challenge when using machine learning in financial crime detection is categorical features, such as City, Merchant Category Code (MCC), or Credit Card Brand. The majority of machine learning algorithms are built for numerical features. When feature categories have an order, converting them to numerical values is straightforward. The challenges come when working with non-ordered categorical features, especially if they have high cardinality. High-cardinality refers to a case where a dataset contains a large number of distinct values or categories in a particular feature or column. One potential solution to this problem is to use categorical feature encoding techniques to transform categorical data into numerical representations that machine learning classifiers can process more effectively. However, the effectiveness of different encoding techniques when dealing with imbalanced data has not been thoroughly evaluated.

In this paper, we compare and evaluate several popular categorical feature encoding techniques for improving the performance of classifiers when dealing with imbalanced data of fraudulent transactions. We explore the impact of these techniques on a range of classifiers, including Decision Trees, Random Forests, and Gradient Boosting. We investigate encoding implications for machine learning performance using two encoding technique groups - *target agnostic*, which does not rely on any target information, and *target based*, which transfers statistical information of the target to the variable. Both groups have their strengths and weaknesses. Target-agnostic techniques ignore the relationship between the values of categorical features and the target value, while target-based methods can suffer from prediction shift. *Prediction shift* is a phenomenon that occurs when the underlying distribution of a dataset changes over time which leads to a shift in the relationships between the input features and the target value. In other words, the patterns and correlations in the training data may no longer hold for new data, which can result in inaccurate predictions and reduced model performance [31]. Prediction shifts can occur for various reasons, including changes in the population being studied, changes in the data collection process, or changes in external factors that affect the relationship between the input features and the target. Prediction shift is also known as concept drift, which can happen in many fields. It can be quantified through the Kullback-Leibler divergence to determine the change in posterior probability distributions for different moments of incoming data streams [24].

The rest of the paper is organized as follows. Section 2 provides an overview of related work in feature encoding techniques, specifically in imbalanced data. Section 3 describes the encoding techniques and classifiers used in this study. Section 4 contains information on the datasets used in the experiment, and Section 5 explains our experimental setup. In Section 5, we also present and analyze the results of our experiments. Finally, in Section 6, we discuss our findings and provide recommendations for practitioners in selecting appropriate categorical feature encoding techniques when dealing with imbalanced data of fraudulent transactions.

2 Related work

In this section, firstly, we review valuable and impactful papers in the scope of high-cardinality categorical features encoding with different sizes of datasets and a variety of applications. In the second part, we review the high-cardinality feature encoding impact when dealing with highly imbalanced datasets, where the minority class contains less than 1% samples from the whole dataset.

2.1 High-Cardinality Categorical Features Encoding

Comprehensive research on high-cardinality feature encoding for classification and regression problems using balanced datasets is presented in the paper [23]. The authors compare seven encoding techniques using five machine-learning algorithms on 24 datasets. Datasets used in the research are binary or multi-class and relatively balanced compared to fraudulent transaction datasets. Chosen datasets differ in size; the smallest is less than a thousand entries, and the biggest is more than a million. The datasets consist of 1 to 20 categorical features, each with over 10 levels (distinct values of a particular feature). The highest number of levels for a feature varies from 14 to 30114. The article suggests that target-based encoders outperform target-agnostic encoding techniques.

Uyar et al. [32] compared automatically calculated techniques against expert judgment. Feature encoding techniques were investigated in IVF (in-vitro fertilization) implantation prediction. The suggested frequency-based encoding technique outperforms expert judgment.

A special case was presented in [29], where the Bayesian encoding technique was developed for WeWork's lead scoring engine. The company faces a high-cardinality feature problem as they have categorical features with more than 300k categories. The authors state that the AUC metric improved from 0.87 to 0.97. However, when researchers compared performance on the publicly available dataset, the developed solution was not so impressive.

Due to high-cardinality, these types of features are sometimes excluded from the modeling scope. However, [11], [21] showed that the model's performance increases statistically significantly when they are included.

2.2 Features Encoding for Imbalanced Data

[7] investigates the impact of feature encoding techniques on highly imbalanced fraudulent transaction dataset. The data used for the research is from a major French bank, and Data Protection Law does not allow sharing it. In this case, replicating the experiment is not possible. However, the results and conclusions inspire more profound research. Another study on real fraudulent dataset [27] proposes a way to encode categorical features by applying Word2Vec embedding, which is usually used for sentence encoding. The outcome of the research was a 50% reduction in memory usage and slightly improved performance.

J. M. Johnson and T. M. Khoshgoftaar published several papers regarding high-cardinality categorical features encoding on Medicare Fraud Prediction [16], [17]. The dataset used in the research is highly imbalanced as in 56 million rows, only 0.06% are fraudulent. With [16], researchers showed that semantic embedding performs significantly better than the traditional one-hot encoder, and the SG embedding performs best overall. One-hot encoding is a technique used to transform categorical data into numerical data, and it defines categorical data as binary vectors. In this method, each category is represented as a binary vector with a length equal to the total number of categories. The vector contains 1 in the position corresponding to the category and 0 elsewhere. SG (Skip-gram) embedding is a neural network trained to predict the surrounding words given a target word. The experiments in [17] showed that One-hot encoding is unsuitable for high-cardinality features when using ensemble learners.

3 Methodology

This paper aims to find which categorical feature encoder impacts the classification model performance the most. Consider the multidimensional dataset as an array $X = \{X_i = (x_{i1}, \ldots, x_{im}),\ i = 1, \ldots, n\}$ of m-dimensional data points (in general, samples) $X_i \in \mathbb{R}^m$. Data point $X_i = (x_{i1}, \ldots, x_{im})$ is the result of the observation of some object or phenomenon dependent on m features $x_1, ..., x_m$. Some of the features are numerical, while others are categorical. In addition, each data point belongs to some class y_i, where the value of y_i is the class label of the sample X_i. In our case, features describe particular characteristics of customers' financial behavior, where we have two classes labeled by 0 or 1 - Regular/Legitimate and Fraudulent transactions, i.e., the target variable y gets values $y_i \in \{0; 1\}, i = 1, ..., n$.

3.1 Selection of machine learning algorithms

To evaluate the impact of encoding techniques on the classification algorithms, we select different models in terms of framework, used loss function, regularization, complexity, and speed. We compare ensemble learning models with non-linear and linear models. Ensemble learning can be visually explained as a judgment of the crowd when the decision is taken by voting. A real-life example of a crowd decision can be a famous TV show named "Who Wants to be a Millionaire". The idea of the show was to answer fifteen questions in a row correctly and win one million dollars. The participant had a chance to ask for help for an intelligent friend or audience. The intelligent friend was right almost 65% of the time. Unexpectedly, the audience of random people was correctly answering 91% of the time [30]. Ensemble learning can improve the accuracy of a model compared to a single model by reducing the risk of overfitting and underfitting when combining multiple models. It is also less sensitive to outliers, and noise [28]. Ensemble learning has a subgroup called gradient-boosting, with examples like XGBoost, LightGBM, and CatBoost. Ensemble learning is usually built on Decision trees.

3.1.1 Decision Tree

The abbreviation CART is used for "Classification and Regression Trees" which was introduced by L. Breiman [4]. CART is a Decision Tree algorithm that recursively partitions data into smaller subsets, represented by nodes, with the final subsets being represented by leaf nodes. For each partition, the best splitting feature is selected. This algorithm typically employs *Entropy* or *Gini index* to identify the best feature to split data. Generally, entropy quantifies the degree of uncertainty in the Decision Tree algorithm. In the context of classification, a partition with low entropy is considered relatively pure, where the majority of the points have the same label. In contrast, a partition with high entropy indicates that the class labels are mixed, and there is no clear majority class.

$$Entropy(D) = -\sum_{i=1}^{c} p(i) \log_2 p(i),$$

where $Entropy(D)$ is the entropy of some dataset D, where c is the number of classes and $p(i)$ is the probability of the sample from D to belong to class i. If dataset D is fully pure, i.e., it only has the same class label, then the entropy is equal to zero. $Information\ gain(IG)$ is used to determine whether a given split leads to a decrease in overall entropy.

$$IG = Entropy(D) - \sum_{j=1}^{k} \frac{n_j}{n} Entropy(D_j),$$

where k is a number of unique values in the splitting feature, n_j is the number of samples in subset D_j, and n is the total number of samples D. $Entropy(D_j)$ is the entropy of subset D_j, which is calculated in the same way as the entropy of the D. A greater reduction in entropy indicates higher information gain, leading to better split points.

The Gini index is a measure used for evaluating the purity of a split point as well, and it is defined as follows:

$$Gini(D) = 1 - \sum_{i=1}^{c} p^2(i),$$

When a partition is pure, the Gini index is 0 because there is only one class, where the probability is 1, and all other classes have a probability of 0. A split may be better if it has a lower weighted Gini index value, where the weighted Gini index is defined as follows:

$$wGini = \sum_{j=1}^{k} \frac{n_j}{n} Gini(D_j),$$

Trees can grow very large when working with large data sets, and that leads to overfitting. To mitigate this, we can specify a minimum number of samples in the leaf or decide on the maximum

depth of the tree. Another technique to mitigate overfitting is called *pruning*. This procedure prevents the Decision Tree from growing to its full depth. Pruning involves removing nodes or branches from the Decision Tree that does not significantly improve the model's performance. The result is a smaller and simpler Decision Tree that is less likely to overfit and more likely to generalize well to new, unseen data. The most significant advantages of a Decision Tree are its simplicity and good performance.

3.1.2 Random Forest

L. Breiman in 2001 [5] introduced a Random Forest algorithm. It is one of the most used ensemble learning algorithms, primarily for its simplicity and prediction power. [15] showed that Random Forest can beat other classifiers from 17 families under different kinds of problems by using 121 databases from UCI. On the other hand, this research does not provide insights and experiments on highly imbalanced data sets.

A Random Forest is a machine-learning algorithm that combines multiple Decision Trees to make a final prediction. The number of Decision Trees in a Random Forest is a hyperparameter that can be set before training the model. Each Decision Tree in the Random Forest is trained on a random subset of the training data and a random subset of the features. This process is repeated multiple times to create a diverse set of Decision Trees. The final result is obtained during prediction by aggregating the predictions of all the individual trees in the forest. The aggregated prediction is either by taking the majority of individual predictions (for classification) or the mean of the predicted values (for regression).

3.1.3 XGBoost - eXtreme Gradient Boosting

XGBoost stands for eXtreme Gradient Boosting [13]. The XGBoost classifier algorithm starts by initializing the model with a single Decision Tree called the base learner. On the other hand, XGBoost also supports other types of base learners, such as linear models. The base learner is typically a shallow Decision Tree with few nodes, which serves as a weak learner. The model then calculates the gradient of the loss function with respect to the predictions made by the base learner. This gradient represents the direction in which the model needs to update the predictions to reduce the loss. The XGBoost classifier constructs a new Decision Tree to correct the errors of the base learner. The construction of this tree is done greedily by iteratively adding nodes that minimize the loss function. The tree is built by selecting the best-split point at each node based on the gradient of the loss function. Once the new tree is constructed, the XGBoost classifier updates its predictions by adding the new tree's predictions to the previous trees' predictions. This process is repeated for a fixed number of iterations or until the model converges to acceptable performance. XGBoost includes several regularization techniques to prevent overfitting, such as $L1$ and $L2$ regularization and tree pruning.

The XGBoost predicted value is as given below [13]:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(X_i), f_k \in F,$$

where K is the number of Decision Trees, $f_k(X_i)$ is the function of input in the k-th Decision Tree, and F is the set of all possible Classification And Regression Trees (CART).

The loss function of the XGBoost consists of training error and regularization:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2,$$

where l is the loss function, T is the number of the leaf nodes, w is the score of the leaf nodes, γ is the leaf penalty coefficient, and λ controls the scale of w.

As the model is trained in an additive way, we can rewrite the loss function as

$$\pounds^{(k)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{k-1} + f_k(X_i)) + \Omega(f_k).$$

Using second-order approximation (an estimate of the second derivative of the loss function with respect to each parameter), we can optimize the loss function:

$$\tilde{\pounds}^{(k)} = \sum_{i=1}^{n} [l(y_i, \hat{y_i}^{k-1}) + g_i f_k(X_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_k),$$

where $g_i = \partial_{\hat{y}^{(k-1)}} l(y_i, \hat{y_i}^{(k-1)})$ and $h_i = \partial^2_{\hat{y}^{(k-1)}} l(y_i, \hat{y_i}^{(k-1)})$. The solutions for the optimal values of w and loss function are [13]

$$w_j = \frac{G_j}{H_j + \lambda},$$

$$L = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \lambda T,$$

where $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$, and I_j is the instance set of leaf j.

This greedy optimization makes XBGoost a fast algorithm but does not necessarily lead to the optimal solution.

3.1.4 LightGBM - Light Gradient Boosting Machine

Gradient Boosting Decision Trees (GBDT) is a machine learning algorithm combining Decision Trees and gradient boosting to create an ensemble model. Gradient boosting iteratively improves a model's predictions by adding new models to the ensemble, each focusing on previously misclassified examples. GBDT face challenges when dealing with large data samples, and they can require a large amount of memory, especially when the number of features or trees is high. For each feature, GBDT requires scanning through all data instances to calculate the information gain (IG) for every potential split point.

Reducing the number of data instances or features seems like a simple solution to address this issue. However, it is not a trivial task. No weight is assigned to the data instance in the GBDT, and the gradient of the loss function is used to update the model in each iteration instead. Data instances with more significant gradients have a more considerable impact on constructing the Decision Tree. It means that they also have a more significant influence on the computation of information gain, even though no exact weight is assigned to each data instance. This conclusion is one of the prominent uniqueness of the LightGBM [19].

Thus, when undersampling the data instances, we should better keep those instances with large gradients to maintain the accuracy of information gain estimation and only randomly drop those instances with slight gradients. The paper [19] proves that the mentioned strategy can increase information gain estimation accuracy better than uniformly random sampling. This approach is called Gradient Based One Side Sampling (GOSS).

Additionally, Light GBM implemented Exclusive Feature Bundling (EFB) algorithm. The authors design an efficient algorithm to solve the optimal bundling problem by reducing it to a graph coloring problem and solving it using a greedy algorithm with a constant approximation ratio which means that the solution it produces is always within a constant factor of the optimal solution.

LightGBM can encode categorical features inside the algorithm. However, in this paper, we are not using this option and feed already encoded data to achieve the goal of the research.

3.1.5 CatBoost - Category Boosting

CatBoost is another boosting algorithm released in 2017 [14] after XGBoost and LightGBM, [25]. CatBoost can automatically handle categorical features by combining one-hot and integer encoding if

needed. It also uses target encoding to deal with high-cardinality categorical features. However, the novelty of this method is that it addresses and suggests solutions for solving the prediction-shifting problem. The solution is called ordered boosting.

CatBoost addresses prediction shifts by creating new datasets at each boosting step, which are independent of the previous datasets, to obtain unshifted residuals. This is accomplished by applying the current model to new training examples. No instances may be used for training the previous models to ensure unbiased residuals for all training examples. In this case, CatBoost maintains a set of models that differ in the examples used for training. When calculating the residual for a particular example, CatBoost uses a model that was trained without that example. The random permutation of the training examples is used to achieve this.

3.1.6 Logistic Regression

Logistic regression is one of the most widespread classical machine learning models, and it is used in many applications and domains. The reason for its popularity, first of all, is its simplicity and explainability. Besides that, logistic regression does not require much computational power. On the other hand, it performs better when the data is linearly separable. It is a machine learning algorithm based on a statistical model with the binary dependent variable. Logistic regression describes data and the relationship between one dependent variable and independent variables.

The logistic regression model can be written as follows:

$$log(\frac{p}{1-p}) = \sum_{j=0}^{m} \beta_j x_j,$$

$$p = \frac{e^{\sum_{j=0}^{m} \beta_j x_j}}{1 + e^{\sum_{j=0}^{m} \beta_j x_j}},$$

where p is the probability that the event will happen. x_j are the individual variables, $j = 1, .., m$ and $x_0 = 1$. Logistic regression aims to estimate β_j, where $j = 0, ..., m$.

3.2 Selection of encoding techniques

Most machine learning algorithms are built for numerical data. Hence researchers and developers must decide how to encode categorical variables. Various encoding techniques exist for this purpose. They can be grouped based on their relation to the target. Namely, target-based and target-agnostic. Another way to group encoding techniques is based on their impact on dataset dimensionality. Encoders like One-Hot or Hashing encoders are the ones that increase the dimensionality of the data set. Our paper analyzes four target-based and two target-agnostic techniques presented in Fig.1.

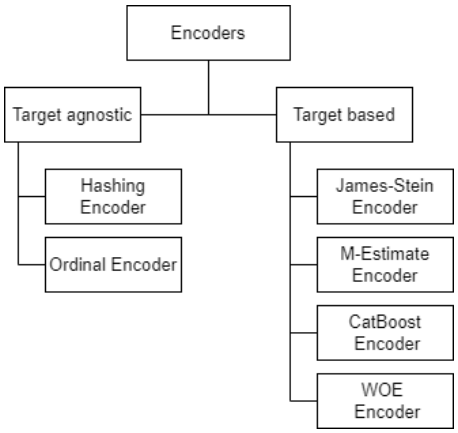


Figure 1: Encoders.

3.2.1 m-estimate Encoder

m-estimate encoder is a target-based encoder. It has one hyperparameter — m, representing the power of regularization where a higher value of m results in stronger shrinking. Recommended values for the m are in the range of 1 to 100. The formula to compute estimated values for a category is [20]:

$$S_i = \frac{n_{iY} + p_{prior}m}{n_i + m},$$

where S_i is the encoded value for category i, n_i is the number of times the category i appears in the dataset, n_{iY} is the number of times the binary target has value 1 ($Y = 1$) when the category is i, p_{prior} is a prior probability of $Y = 1$ without considering categories.

3.2.2 James-Stein Encoder

James-Stein encoder is a target-based encoder as well. Initially, the James-Stein estimator was not meant to be used for binary classification and was defined only for normal distributions. In our case, we want to apply it for binary classification, so firstly, we convert the mean target value to the log-odds ratio.

The James-Stein encoder is a method for shrinking mean estimates only when the variances of those means are assumed to be equal. However, this assumption is often only valid when the sample sizes of each group are equal. In most real-world scenarios, sample sizes and variances of the means are not equal, which makes it difficult to determine the appropriate course of action. For the execution of the James-Stain encoder, we use the *Scikit-learn* library *Category Encoders*, which has implemented a binary version of the James-Stein encoder proposed in the paper [35]

3.2.3 CatBoost Encoder

CatBoost encoder uses the same formula as the m-estimate encoder. However, it was noticed [25] that the usage of the whole sample to compute S_i leads to a target shift. Permutations of the training set were suggested in [25]. For the execution, we use the *Scikit-learn* library *Category Encoders*, where the implementation is time-aware (it does not use random permutation).

3.2.4 Weight of Evidence Encoder

The Weight of Evidence (WOE) is a statistical measure that quantifies the strength of the relationship between a categorical variable and a binary target variable. The WOE for a particular category is calculated by taking the natural logarithm of the ratio of the percentage of observations in that category that belongs to the class $Y = 0$ to the percentage of observations in that category that belongs to the class $Y = 1$.

$$WOE_i = ln\frac{p_{i(Y=0)}}{p_{i(Y=1)}},$$

where $p_{i(Y=0)}$ is percentage of $Y = 0$ when the category is i; $p_{i(Y=1)}$ is percentage of $Y = 1$ when the category is i.

3.2.5 Label/Ordinal Encoder

Label/Ordinal encoder is a target-agnostic encoder, and it does not use the statistical information of the target variable. Label encoder depends on the ordering of the encoded data because it assigns integer numbers from 0 to $k - 1$ despite the meaning of the data, where k is the number of different values of a particular categorical feature. The advantage of this method is its simplicity. However, it gives an unwanted order and weight for the categories.

3.2.6 Hashing Encoder

Feature hashing is a technique for converting categorical features into numerical features for machine learning models. It works by mapping each category of a categorical feature to an integer within a pre-determined range. The output range is usually smaller than the input range, meaning multiple categories may be mapped to the same integer. Such conditions are called collisions. However, collisions are often rare in practice and do not significantly affect performance.

Feature hashing is similar to one-hot encoding but with a few key differences. One of the main advantages of feature hashing is that it allows for control over the output dimensions. Additionally, feature hashing can be faster and more memory-efficient than one-hot encoding, especially when dealing with large datasets.

This encoder applies the hashing trick to a categorical feature and then encodes the resulting integers as numerical features. The basic idea behind the hashing trick is to use a hash function to map the input data to a fixed-size output space. The hash function takes the input data (e.g., a word or categorical feature) as input and produces a hash value that is an integer between 0 and a predefined maximum value.

4 Data used for experiment

The availability of necessary data sets with high volume, velocity, and variety is needed to accelerate scientific research. Research in the area of fraudulent transactions is limited by data availability. Many law regulations regarding private data usage exist in the real world, such as GDPR (General Data Protection Regulation), CCPA (California Consumer Privacy Act), "Act On Payment Services And Electronic Money", etc. Synthetic data is a promising technology that helps to solve privacy, fairness, data augmentation, and many other issues.

The definition of synthetic data proposed in [18] is "Synthetic data is data that has been generated using a purpose-built mathematical model or algorithm, intending to solve a (set of) data science task(s)."

Synthetic Data plays a vital role in research and developments where data availability is limited by laws and its nature to be rare. A great application of synthetic data set is presented in [10]. This synthetic data set is used to generate realistic cyber data for machine learning classifiers for network intrusion detection systems [10]. The paper [10] concludes that their chosen generative methods - CTGAN and TVAE - generate synthetic cyber data reasonably well. Nevertheless, ML models trained with only synthetic data resulted in low classification recall. Further, the authors suggest having at least 15% of actual data when training the model.

Financial fraud is one of those areas where access to data is very limited. That was the main reason to work with synthetic data generated by Erik Altman [2]. This data set aims to allow researchers and developers to work on the data that represents buying habits of U.S. citizens. The dataset is like a virtual world with customers, merchants, and fraudsters. The data was created so that features kept their main statistics like mean and standard deviation that would be the same as in the actual population. But it's not just averages and standard deviations that are needed. Erik Altman [2] selects characteristic values for individuals by stochastic sampling, generally from a Gaussian distribution. The advantage as compared to other synthetic data sets [3] is that the individuals' activities are related. For instance, if an individual is in travel mode, he/she will have different spending behavior. Similarly, the same logic applies if the purchase happens on weekdays or weekends, and much more evidence that this data set reflects the actual population can be found in the [2].

Another essential thing to be mentioned is that this data set's virtual credit card world includes actual banking events like creating the chip in the card. Chips were introduced on a large scale in the U.S. in 2014 before that magnetic stripe technology was used. After that, it became harder to perform "card-present" fraud (a transaction in which the fraudster physically presented the stolen credit card to the merchant).

This data set will be called Dataset1 in the rest of the paper. The study conducted in the paper [6] involves analyzing this dataset and recommending strategies for achieving balance through clustering.

By creating smaller and more homogeneous clusters, undersampling methods can be employed without sacrificing crucial information needed for machine learning algorithms.

The second data source (Dataset2) used for the experiment is also synthetic. The dataset was generated using the Sparkov Data Generation tool. This data set is smaller than the previous one, with 1.3 million transactions, of which 0,57% are fraudulent. The generated data set has five categorical features from eleven in total.

5 Experimental results

Data preparation is one of the essential parts of successful data science research. In the scope of this research, we are targeting massive datasets with less than a 1% imbalanced ratio. Additionally, we are interested in the datasets containing categorical features with high cardinality.

Our goal is to find the best-fitting encoder for highly imbalanced massive data, so we are not hyper-tuning selected machine learning models or changing thresholds. We calculate results using cross-validation with a stratified split of five-fold. We have performed the cross-validation four times with different seeds. We believe that by using the Grid Search algorithm, we could achieve better results in general. For the encoding algorithms, we use default parameters as well. Our focus is on univariate encoding, where features are always encoded separately.

The experiment aims to analyze and show which encoding methods are best suited for imbalanced data. Even though LightGBM and CatBoost have their own feature encoding methods inside the algorithms, we are not comparing these encoding with others as they use optimization strategies. The results would not be comparable.

Both datasets have categorical features with different cardinality. The cardinality of each categorical feature is presented in Table1.

Training set size	5 969 329
Categorical feature	cardinality
Card Brand	4
Card Type	3
Has Chip	2
Use Chip	3
Merchant City	11 391
MCC	109
Error1_cat	8
Error2_cat	5
Gender	2
City	1 074
State	51

(a) Dataset1

Training set size	907 672
feature	cardinality
Category of MCC	14
Gender	2
City	894
State	51
Job	494

(b) Dataset2

Table 1: Feature cardinality.

Below, we present an example of differences in encoding technique performance by plotting histograms of encoded values of the categorical feature "State" from Dataset1 (Fig.2 - Fig.6). The x-axis represents encoded values, and the y-axis shows the number of cases of the appearance of the particular encoded value. The maximum histogram of the Label encoder is much lower than that of the CarBoost encoder. This means that the CatBoost encoder shrinks categorical values, resulting in a much higher number of cases for a particular encoded value.

Upon analyzing the target-based techniques, it is evident that the encoded values display notable variability in terms of their size and shape, as represented in the histograms. More specifically, the James-Stein encoder demonstrates a compact range of values, whereas the WOE encoding method yields predominantly negative values. Additionally, the CatBoost encoder results in a distribution of encoded values that are asymmetrically skewed.

Target-agnostic techniques are not comparable in our case, as the Ordinal encoder does not expand dataset dimensionality while Hashing encoder does. The histogram of the encoded variable "State" with an Ordinal encoder represents the frequency of the values encoded with no logical order.

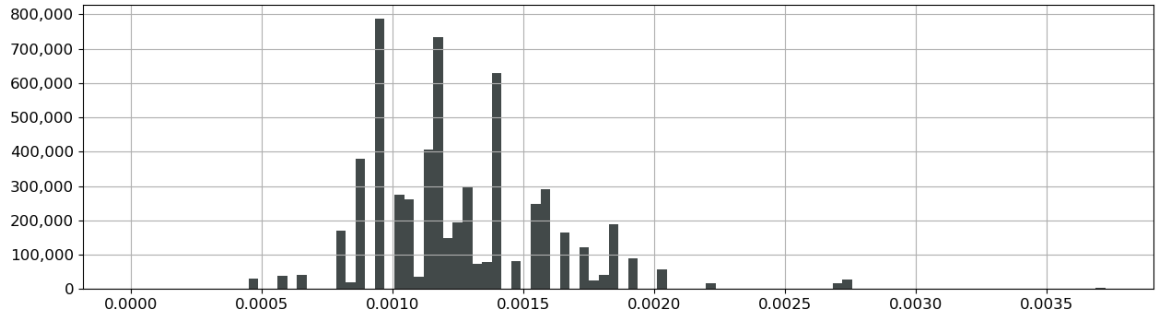


Figure 2: State encoded using m-estimate encoder.

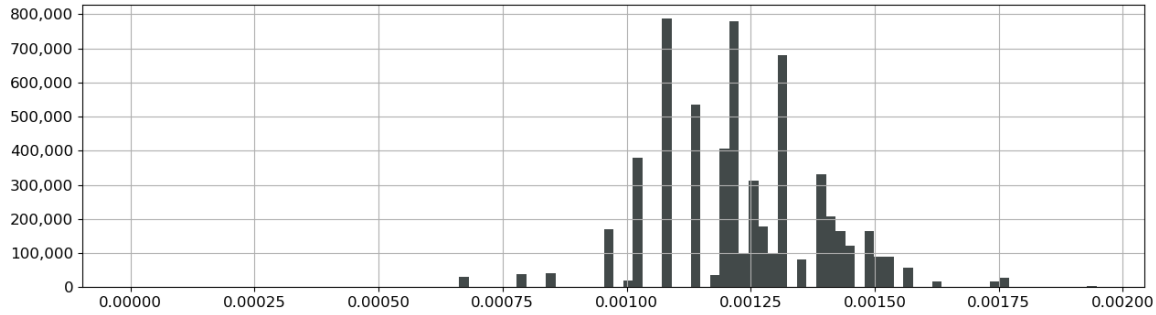


Figure 3: State encoded using James-Stein encoder.

Visual comparison of the feature "State" encoded with different encoders can be challenging, owing to their differing scales. In an attempt to mitigate this challenge, we opt to standardize the encoded values by scaling them within the range of zero and one. Following, we plotted the density function, as depicted in Figure 7. This visualization allowed us to easily discern that the James-Stein, M-estimate, and WOE encoders demonstrate similar shapes and density amplitude. However, their primary variation lies in their position along the x-axis. Conversely, values encoded with the CatBoost encoder are characterized by a significant level of skewness, as previously noted. Moreover, we observed that the encoded values with CatBoost appear to be compressed, as evident from the density function plot.

We can draw several conclusions based on the presented results in Fig.8 - Fig.9. Firstly, we can observe that target-based encoding methods outperform target-agnostic ones on Dataset1 and Dataset2. This indicates that incorporating the target in the encoding process can result in better performance of the machine learning model. Target-based encoding methods allow the model to capture the relationship between the input features and the target variable more effectively, thus improving the model's predictive power. Secondly, the results indicate that logistic regression without hyper-tuning is unsuitable for highly imbalanced datasets. The performance of the logistic regression model was very poor for both datasets, indicating that this algorithm is not robust enough to handle imbalanced data. Therefore, other machine learning algorithms that can handle imbalanced data, such as gradient boosting machines, should be used in such cases. Finally, we can see from the box plot on the bars that LightGBM is highly sensitive to encoding techniques. This suggests that choosing the appropriate encoding method is crucial for achieving optimal performance when using LightGBM. Therefore, it is essential to experiment with different encoding techniques to identify the best one for a given dataset and machine learning algorithm.

The results presented in Fig.9 are presented by the encoders used in the machine learning models. The figure shows that the James-Stein and WOE encoders are consistently chosen as the best-performing ones across the different machine-learning models. Furthermore, it is worth noting that the

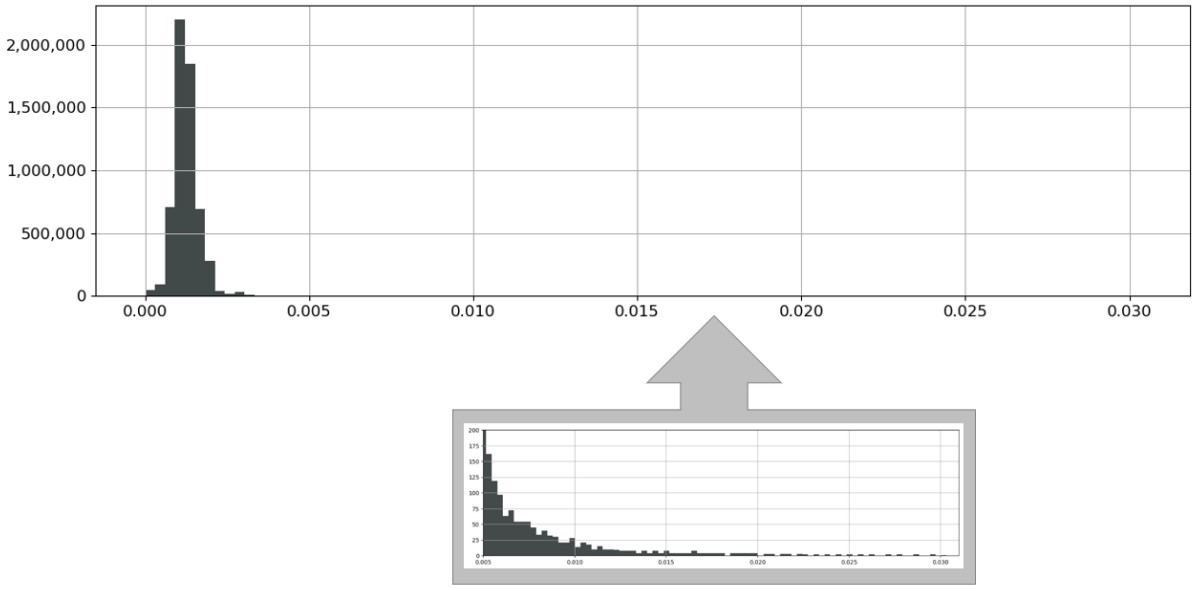


Figure 4: State encoded using CatBoost encoder.

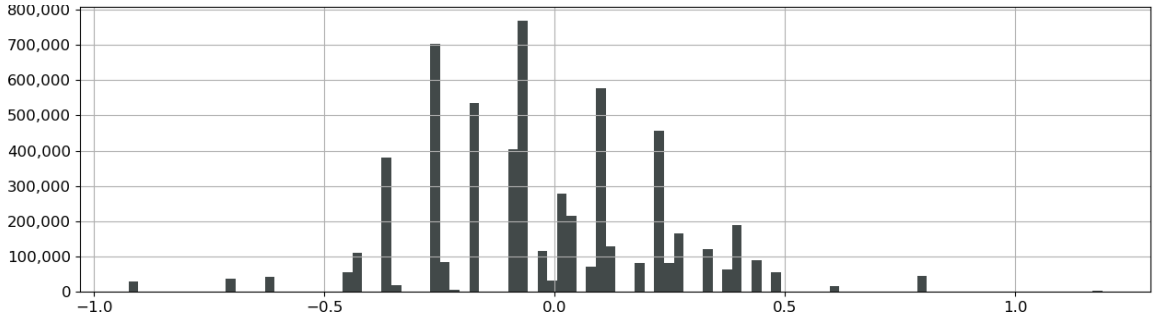


Figure 5: State encoded using WOE encoder.

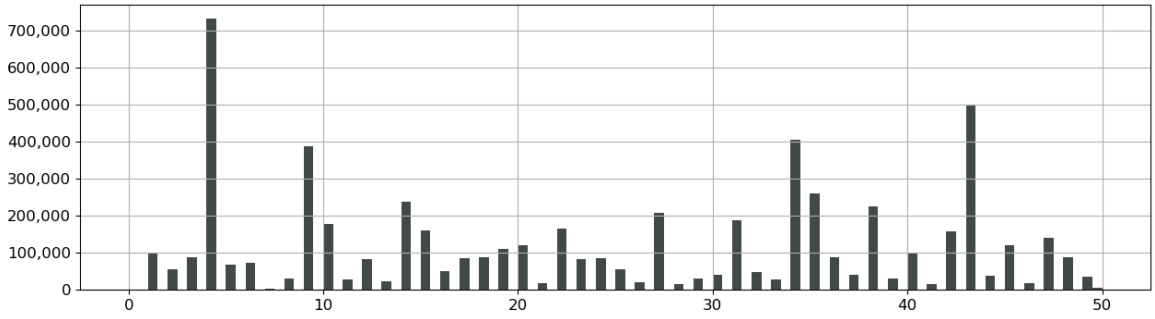


Figure 6: State encoded using Label encoder.

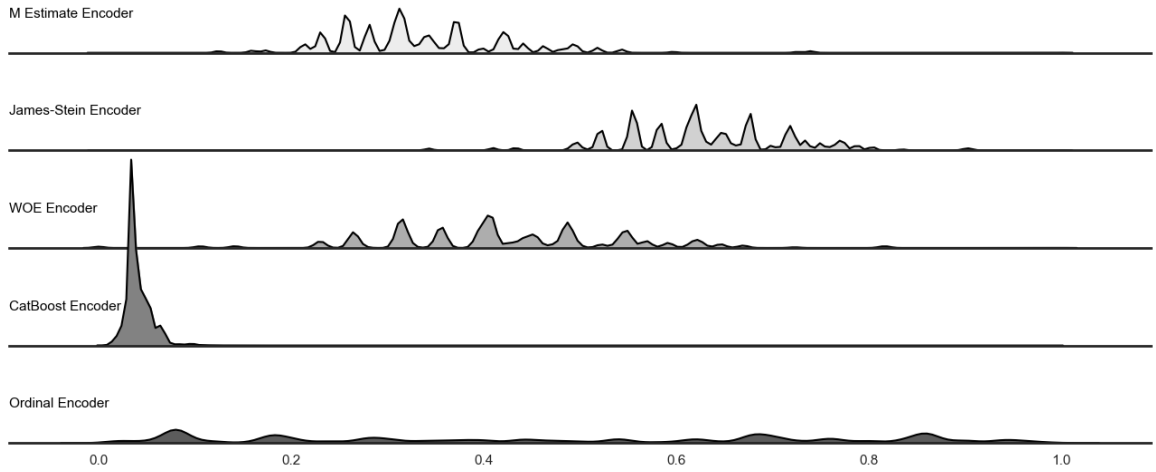


Figure 7: Density functions of the encoded feature "State" using different encoders.

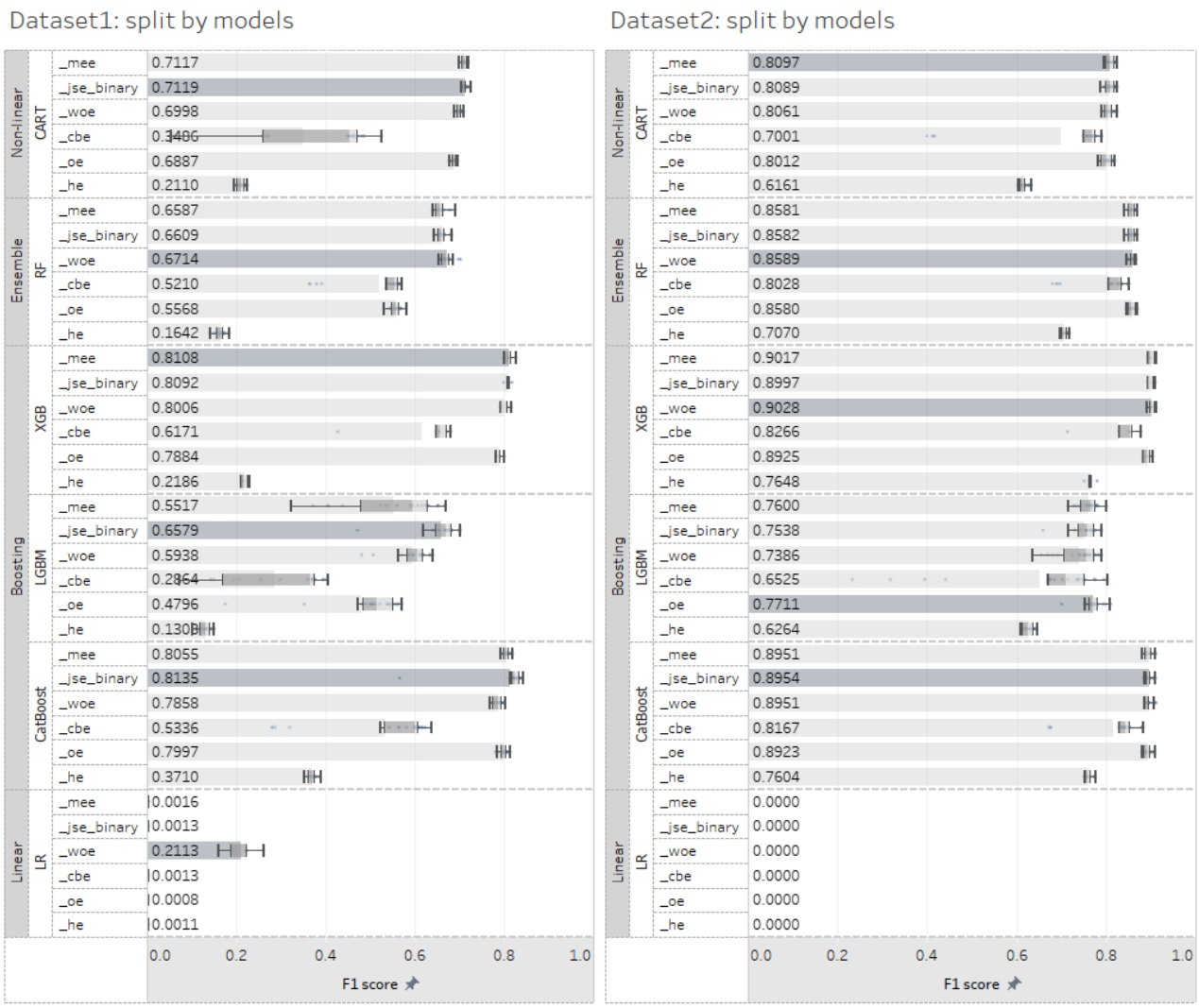


Figure 8: Experimental results grouped by models.

CatBoost encoder performs poorly among the target-based encoding techniques. This suggests that the CatBoost encoder may not be suitable for imbalanced datasets, as it fails to effectively capture the relationship between the input features and the target. The performance of the hashing encoder is consistent with the findings reported in related research [17]. It is important to note that we did not include the One-Hot encoder in our evaluation. However, it is well known that One-Hot encoding can lead to growing dimensionality of the dataset, known as the curse of dimensionality. Similarly, hashing encoding can also increase the dimensionality of the dataset. Therefore, both encoding techniques may not be optimal choices for datasets with a large number of categorical features and instances.

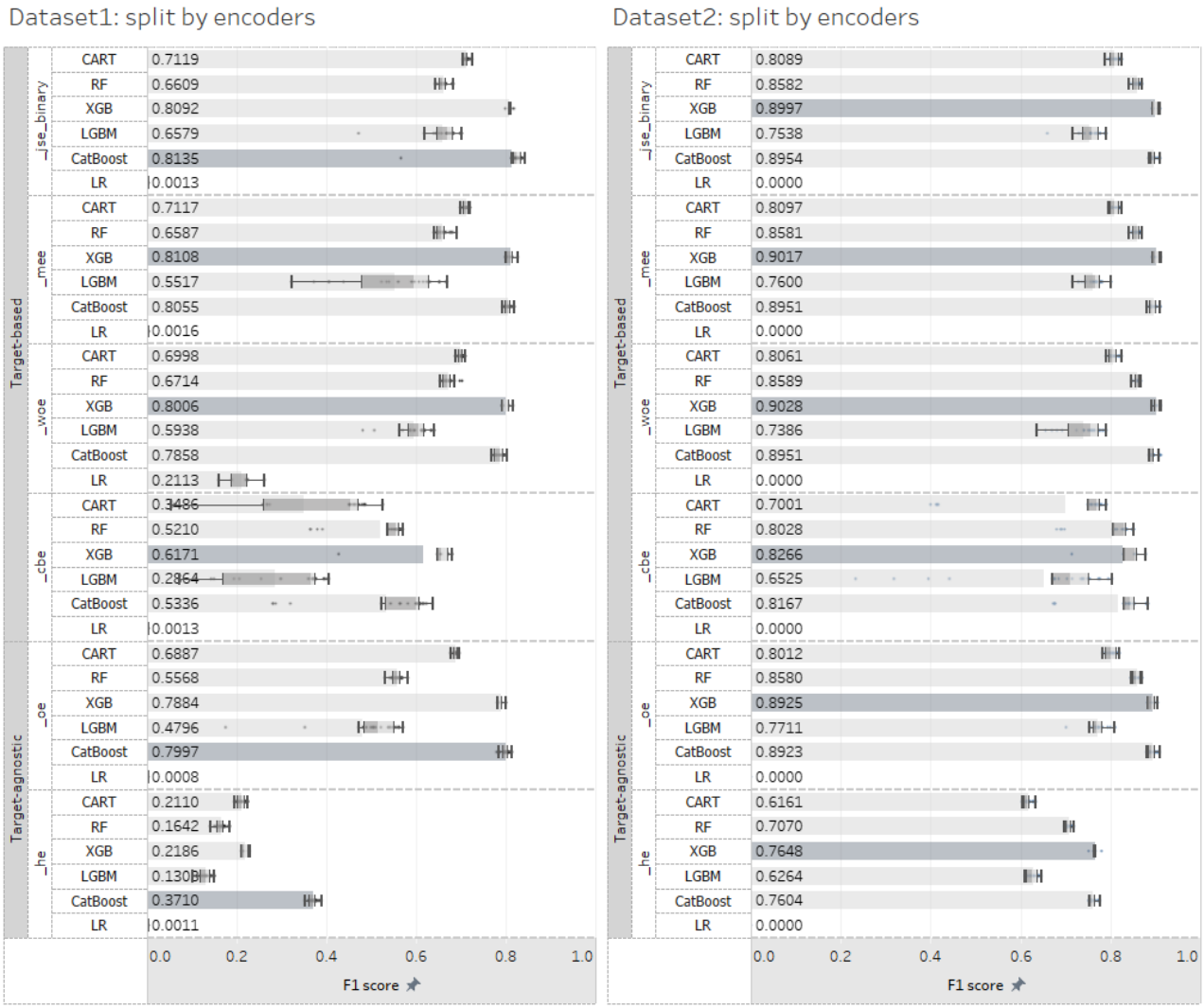


Figure 9: Experimental results grouped by encoders.

6 Discussion and Conclusions

Our research aims to determine the most appropriate encoding technique for handling highly imbalanced datasets. We conducted an experiment to test six encoding methods, both from the target-agnostic and target-based groups. To conduct the experiment, we utilized various machine-learning methods, including ensemble learning, as well as linear and non-linear learning. Specifically, we focused on transaction datasets, where the target variable indicates whether the transaction is regular or fraudulent. These datasets are complex, with several high-cardinality features.

Several papers have been written regarding this subject matter, showcasing experiments conducted on publicly accessible balanced datasets. Nevertheless, minimal investigation has been conducted until now when dealing with highly imbalanced datasets.

The findings presented in Fig.8 - Fig.9 highlight the importance of selecting the appropriate encoding method when working with imbalanced datasets and machine learning algorithms, as well as the benefits of using target-based encoding methods to improve model performance.

The results presented in Fig.9 suggest that the choice of encoding technique can significantly impact the performance of the machine learning models. The James-Stein and WOE encoders appear to be the most effective among the encoding techniques evaluated. Additionally, the CatBoost encoder may not be suitable for imbalanced datasets. Finally, it is vital to consider the potential curse of dimensionality of the dataset when using encoding techniques such as hashing and One-Hot encoding.

The discovered properties will lead to the development of more efficient new classification methods and the improvement of existing ones, e.g., [6] for highly unbalanced financial data.

Funding

The research was funded by Vilnius University.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Alarfaj, F. K.; Malik, I.; Khan, H. U.; Almusallam, N.; Ramzan, M.; Ahmed, M. (2022). Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms, *IEEE Access*, 10, 39700–39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[2] Altman, E. (2021). Synthesizing credit card transactions, *2nd ACM International Conference on AI in Finance (ICAIF'21)*, [Online]. Available: https://doi.org/10.1145/3490354.3494378

[3] Alonso Lopez-Rojas, E.; Axelsson, S. (2014). BankSim: A Bank Payment Simulation for Fraud Detection Research, *The 26th European Modeling and Simulation Symposium*, [Online]. Available: https://www.researchgate.net/publication/265736405

[4] Breiman, L. (1984). *Classification and Regression Trees (1st ed.).* Routledge. https://doi.org/10.1201/9781315139470

[5] Breiman, L. (2001). Random Forests, *Machine Learning* 45, 5–32, 2001, doi: 10.1023/A:1010933404324

[6] Breskuvienė, D.; Dzemyda, G. (2023). Imbalanced Data Classification Approach Based on Clustered Training Set, In: *Dzemyda, G., Bernatavičienė, J., Kacprzyk, J. (eds) Data Science in Applications. Studies in Computational Intelligence*, Springer, Cham. 1084, 43–62, 2023. doi.org/10.1007/978-3-031-24453-7_3

[7] Bourdonnaye, F.; Daniel, F. (2021). Evaluating categorical encoding methods on a real credit card fraud detection database, [Online]. Available: http://www.lusisai.com 2021.

[8] Bugajev, A.; Kriauzienė, R.; Vasilecas, O.; Chadyšas, V. (2022). The Impact of Churn Labelling Rules on Churn Prediction in Telecommunications. *Informatica*, 33(2), 247–277, 2022. doi:10.15388/22-INFOR484

[9] Bulavas, V.; Marcinkevičius, V.; Rumiński, J. (2021). Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets, *Informatica*, 32(3), 441–475, 2021, doi: 10.15388/21-INFOR457

[10] Chalé, M.; Bastian, N. D. (2022). Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems, *Expert Systems with Applications*, 207, 117936, 2022, doi: 10.1016/j.eswa.2022.117936.

[11] Carneiro, E. M.; Forster, C. H. Q.; Mialaret, L. F. S.; Dias, L. A. V.; Cunha, A. M. (2022). High-Cardinality Categorical Attributes and Credit Card Fraud Detection, *Mathematics*, 10(20), 2022, doi: 10.3390/math10203808.

[12] Chen, C.; Liaw, A.; Breiman, L. (2004). Using random forest to learn imbalanced data, *University of California, Berkeley* (110), 1–12, 2004.

[13] Chen, T.; Guestrin, C. (2016). XGBoost: A scalable tree boosting system, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794, 2016. doi:10.1145/2939672.2939785

[14] Dorogush, A. V.; Ershov, V.; Gulin, A. (2018). CatBoost: gradient boosting with categorical features support, [Online]. Available: http://arxiv.org/abs/1810.11363

[15] Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D.; Fernández-Delgado, A. (2014). Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?, [Online]. Available: http://www.mathworks.es/products/neural-network.

[16] Johnson, J. M.; Khoshgoftaar, T. M. (2020). Hcpcs2Vec: Healthcare Procedure Embeddings for Medicare Fraud Prediction, *2020 IEEE 6th International Conference on Collaboration and Internet Computing*, 145–152, 2020. doi: 10.1109/CIC50333.2020.00026.

[17] Johnson, J. M.; Khoshgoftaar, T. M. (2021). Encoding Techniques for High-Cardinality Features and Ensemble Learners, *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science*, 355–361, 2021. doi: 10.1109/IRI51335.2021.00055.

[18] Jordon, J. et al. (2022) Synthetic Data – what, why and how?, [Online]. Available: http://arxiv.org/abs/2205.03257

[19] Ke, G. et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree, [Online]. Available: https://github.com/Microsoft/LightGBM.

[20] Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems, *ACM SIGKDD Explorations Newsletter*, 3(1), 2001. doi: 10.1145/507533.507538.

[21] Moeyersoms, J.; Martens, D. (2015). Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector, *Decision Support Systems*, 72, 72–81, 2015. doi: 10.1016/j.dss.2015.02.007.

[22] Najadat, H.; Altiti, O.; Aqouleh, A. A.; Younes, M. (2020). Credit Card Fraud Detection Based on Machine and Deep Learning, *11th International Conference on Information and Communication Systems*, 204–208, 2020. doi: 10.1109/ICICS49469.2020.239524.

[23] Pargent, F.; Pfisterer, F.; Thomas, J.; Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features, *Computational Statistics*, 37(5), 2671–2692, 2022. doi: 10.1007/s00180-022-01207-6.

[24] Peng, Y.; Qiu, Q; Zhang, D.; Yang, T.; Zhang H.(2023). Ensemble Learning for Interpretable Concept Drift and Its Application to Drug Recommendation, *International Journal of Computers Communications & Control*, 18(1), 5011, 2023. doi.org/10.15837/ijccc.2023.1.5011

[25] Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; Gulin, A. (2017). CatBoost: unbiased boosting with categorical features, [Online]. Available: http://arxiv.org/abs/1706.09516

[26] Reilly, D.; Taylor, M.; Fergus, P.; Chalmers, C.; Thompson, S. (2022). The Categorical Data Conundrum: Heuristics for Classification Problems - A Case Study on Domestic Fire Injuries, *IEEE Access*, 10, 70113–70125, 2022, doi: 10.1109/ACCESS.2022.3187287.

[27] Russac, Y.; Caelen, O.; He-Guelton, L. (2018). Embeddings of Categorical Variables for Sequential Data in Fraud Context, *Advances in Intelligent Systems and Computing* doi: 10.1007/978-3-319-74690-6_53

[28] Sagi,O.; Rokach, L. (2018). Ensemble learning: A survey, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), 2018. doi: 10.1002/widm.1249.

[29] Slakey, A.; Salas, D.; Schamroth, Y. (2019). Encoding Categorical Variables with Conjugate Bayesian Models for WeWork Lead Scoring Engine, [Online]. Available: http://arxiv.org/abs/1904.13001

[30] Surowiecki, J. (2004). *The wisdom of crowds*, Anchor, 2004.

[31] Turhan, B. (2012). On the dataset shift problem in software engineering prediction models, *Empirical Software Engineering*, 17(1–2), 62–74, 2012. doi.org/10.1007/s10664-011-9182-8

[32] Uyar, A.; Bener, A.; Ciray, H. N.; Bahceci, M. (2009). A frequency based encoding technique for transformation of categorical variables in mixed IVF dataset, *31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine*, 6214–6217, 2009. doi: 10.1109/IEMBS.2009.5334548.

[33] Wang, H.; Wang, W.; Liu, Y.; Alidaee, B. (2022). Integrating Machine Learning Algorithms With Quantum Annealing Solvers for Online Fraud Detection, *IEEE Access*, 10,75908–75917, 2022. doi: 10.1109/ACCESS.2022.3190897.

[34] Zhao, X.-M.; Li, X.; Chen, L.; Aihara, K. (2007). Protein classification with imbalanced data, *Proteins*, 70(4), 1125–1132, 2007.

[35] Zhou, X. (2015). Shrinkage Estimation of Log-odds Ratios for Comparing Mobility Tables, *Sociol Methodology*, 45(1), 320–356, 2015. doi: 10.1177/0081175015570097.

Cite this paper as:

Breskuvienė, D.; Dzemyda, G. (2023). Categorical Feature Encoding Techniques for Improved Classifier Performance when Dealing with Imbalanced Data of Fraudulent Transactions, *International Journal of Computers Communications & Control*, 18(3), 5433, 2023.
https://doi.org/10.15837/ijccc.2023.3.5433