ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Encoding Matters: Impact of Categorical Variable Encoding on Performance and Bias

Daniel Kopp[1], Benjamin Maudet[1,2], Lisheng Sun-Hosoya[2], and Kristin P. Bennett[3]
1- ChaLearn. 2- U. Paris-Saclay. 3- RPI.

Abstract.
Encoding categorical variables impacts model performance and can introduce bias in supervised learning, particularly affecting fairness when some groups are under-represented. We analyze the effects of different encoding methods on synthetic and real datasets to mitigate unintended model reliance on specific variables. We propose CaVaR (Categorical Variable Reliance) to quantify model reliance on variables and an Availability Index to measure CaVaR's sensitivity to partial encoding changes. A high Availability Disparity, measured by the standard deviation of the Availability Index across encodings, highlights potential bias from mixed encodings. The results suggest encoding all categorical variables uniformly, regardless of their ordinal or nominal nature, may reduce bias, with the choice guided by computational and performance considerations.

1 Background and motivations

In machine learning, it is common to encounter various data types that need to be properly encoded before being input into the learning model. Some models, such as linear models and neural networks, work well with quantitative data and require numerical inputs [3]. Other models, like decision trees and grid models, can also handle qualitative data, such as categorical variables [5]. Due to the increasing importance of neural networks in modern AI, it is crucial to correctly encode categorical variables to ensure these models function properly, achieve good generalization, and avoid bias[1] This is the focus of this paper.

Categorical variables are variables that can take one value from a finite set of possible values. Each possible value of a variable is referred to as a level. We consider two types of categorical variables: ordinal variables with a meaningful ordering, *e.g.,* education levels, and nominal variables, which are qualitative with no meaningful ordering, *e.g.,* race. Although nominal variables may be numbered (*e.g.,* user ID's), these numbers do not indicate any particular order. Textbooks and on-line resources[6, 10] usually recommend encoding ordinal and nominal variables differently. Typically, ordinal variables are encoded with a numeric value using their natural order (referred to as "Natural Ordinal" encoding). In contrast, nominal variables are encoded with "One-Hot" encoding, converting k-level categorical variables into a k-dimensional binary vector representation where only one of the levels is "hot" (set to 1) for each observation, while all the others are "cold" (set to 0) [11]. However, these "typical" encoding choices are

[1]To facilitate the reading of this paper, we added a glossary in appendix provided here: `https://github.com/danielkopp4/ESANN2025Appendix`. However, this paper should be readable without the appendix.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

far from neutral. Changing the encoding of a variable can influence the model's reliance on that variable, thereby affecting how well the model generalizes to unseen data. For instance, a model may find it easier to learn a relationship with a variable that has a weaker dependence on the target compared to another, a phenomenon often referred to as shortcut learning [2]. In the case of protected variables (such as gender or race), shortcut learning can be associated with fairness issues by reinforcing or introducing societal biases.

2 Problem setting and methodology

Problem Setting. We study supervised learning problems where data samples $(\mathbf{x}, y) \in \mathcal{X}^d \times \mathcal{Y}$ are drawn from a joint distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$. The goal is to approximate $P(y|\mathbf{x})$ using machine learning (ML).

Training data $\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}$ are drawn *iid* from $P(\mathbf{x}, y)$. We treat the learning process, including encoding, as a black-box model M that outputs an estimator $\hat{P}_M(y|\mathbf{x})$ of $P(y|\mathbf{x})$. In our experiments, encoding is the only variable adjusted; all other factors, including hyperparameters, are held constant. If a variable is disproportionately emphasized in $\hat{P}_M(y|\mathbf{x})$, failing to reflect true statistical dependencies, bias may be introduced. Here, we assume all components of $\mathbf{x}$ are categorical variables and focus on encoding's effect on algorithmic bias.

Categorical variables are drawn from a discrete joint distribution, where a "cell" represents a specific combination of variable levels (*e.g.,* if $\mathbf{x}$ is composed of the two variables "race" and "education level", individuals with "graduate school" education and "Black" race form one cell). Each variable x_i can take k_i levels, and the discrete domain of inputs is represented by a tensor $\mathbf{X}$ of size $K = \prod_{i=1}^{d} k_i$, covering all cells.

Training data follow the natural distribution $P(\mathbf{x})$, while test data are drawn either from the same in-domain distribution or an out-of-domain uniform distribution where all cells in $\mathbf{X}$ have equal probability, including unseen cells.

Methodology. To assess the impact of disparate variable encoding on algorithmic bias, we introduce **CaVaR** (Categorical Variable Reliance), a metric that quantifies the reliance of a predictor $\hat{P}_M(y|\mathbf{x})$ on a subset S of categorical variables. This reliance is measured by evaluating prediction changes when values in S are permuted, neutralizing S's effect during inference. The formula is:

$$\mathrm{CaVaR}_S(M) = \frac{1}{\kappa \times K} \sum_{\pi \in \Pi(S)} \left\| \hat{P}_M(y = 1|\mathbf{X}) - \hat{P}_M(y = 1|\pi(\mathbf{X})) \right\|_1 , \quad (1)$$

where $\Pi(S)$ is the set of κ permutations of S (that can be subsampled for computational reasons), K is the number of cells in $\mathbf{X}$, and the L1-norm difference quantifies the model's reliance on S. This computation does not depend on ground-truth y, making it robust to under-represented cells in real data.

To study encoding effects on M, assume all variables use encoding a, except S, which uses b, represented by $M[a, S \leftarrow b]$. We introduce the notation:

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

$$\mathrm{CaVaR}_S(b; a) = \mathrm{CaVaR}_S(M[a, S \leftarrow b]) \ . \tag{2}$$

$\mathrm{CaVaR}_S(a; a)$ reflects reliance under uniform encoding a, while $\mathrm{CaVaR}_S(b; a)$ shows reliance when S uses b. Averaging the differences across all $b \neq a$ and subsets S yields the **Availability Index**:

$$\mathrm{Avail}(a) = \frac{1}{N_S \times (C-1)} \sum_S \sum_{b \neq a} \left(\mathrm{CaVaR}_S(a; a) - \mathrm{CaVaR}_S(b; a)\right) \ . \tag{3}$$

A high positive $\mathrm{Avail}(a)$ indicates greater reliance when variables are encoded with a compared to b. Variability in $\mathrm{Avail}(a)$ across encodings is summarized by the **Disparity Index**:

$$\mathrm{Disparity} = \mathrm{Std}(\mathrm{Avail}(a)). \tag{4}$$

If Disparity is high, using mixed encodings risks bias by favoring certain variables. Conversely, near-zero Disparity suggests mixed encodings are safe. When Disparity is significant, we recommend uniform encoding for all variables to avoid bias.

3 Experimental conditions

After evaluating various encoding strategies (Appendix C), we selected One-Hot, Ordinal, and Target Continuous encodings for illustration. We also introduce Random Ordinal and Target Ordinal, as specialized forms of Ordinal encoding.

One-Hot encoding is emphasized for its widespread use with nominal variables and its neutrality, as it avoids assumptions about category relationships. In contrast, Ordinal encodings incorporate prior information, which can either aid or hinder model learning. Random Ordinal represents a worst-case scenario, as arbitrary ordering introduces spurious continuity, implying nonexistent interpolation between levels. In our experiments, we average results over multiple random orderings. Target Ordinal assigns integers 0 to $k-1$ (where k is the number of levels) to maximize correlation with the target. Target Continuous encoding maps categories directly to their target variable means, often outperforming Target Ordinal unless tied means cause level merging.

Experiments were carried out on synthetic and real data, described in details in Appendix B. Briefly, the synthetic datasets consist of two-dimensional binary classification problems, with both input variables x_1 and x_2 having 8 levels. They were designed to illustrate various special cases:

Synthetic Simple: x_1 and x_2 are identical and identically dependent on y; any difference in variable availability can only arise from coding.

Synthetic Crossed: This example (similar the the XOR problem) is designed such that neither variable is individually correlated with the target and all values in the marginal distributions are tied. Target Continuous encoding is expected to fail on that example.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Synthetic Doughnut: This is another example of non-linearly separable classes, in which neither variable is correlated with the target. It is designed to show that Target Encoding can be beneficial even when variables are not correlated with the target, if values in the marginal distributions are not tied.

The real datasets, described in details in Appendix B, comprised of Adult Income [1], Bank Marketing [7], Cardiovascular Disease [4], Cook County Sentencing [8], and Mushroom [9]. Their number of features vary between 11 and 22. The number of samples is between 8124 and 266435 samples. The maximum number of levels is between 8 and 1316.

For all datasets, we used 50 examples of each class for training to exacerbate the difficulty of generalizing well out-of-domain. The choice of the same number of examples per class is to simplify the analysis and avoid adding a factor of variability that may confound the results. In future work, we could also vary the number of examples per class. For real data, we used all the remaining samples as test data (so called identically and independently distributed or *iid* test set) and for synthetic data, we drew 1000 samples from the same distribution as the training data. We use the test samples to evaluate the Balanced Accuracy (average of the accuracy on the positive class and the negative class). Additionally, we created a second test set to evaluate both the Availability and Disparity indices, populating the tensor $\mathbf{X}$, providing values for the levels in the ranges defined by the training data distribution, but enforcing uniform sampling.

All experiments reported in the next section were carried out with a 2 layer neural network with 5 hidden units and weight decay regularization of 0.01, using the scikit-learn package. Experiments with various other models are provided as supplemental material in our Github Repo.

4 Results

Visual inspection of the two-dimensional Simple Synthetic data example (Appendix E) demonstrates that encoding variables in different ways can change their availability and therefore result in bias. This suggests that all variable should be encoded in the same way, regardless of whether they are nominal or ordinal, to avoid exacerbating or introducing bias. Because of space limitations, we only discuss in the body of the paper the results of Table 1, which present systematic experimental comparisons of a selection of codes on synthetic and real data, for the two-layer neural network.

We chose a neural network because encoding categorical variables as numerical values is required for such models. We chose two layers and 5 hidden units to have a small model that is capable of making non-linear separations. We also chose to train models on very few examples (50 of each class) so data sparsity will affect variable availability more strongly. We compute the balanced accuracy (average of the accuracy on the positive class and the negative class) and whether encoding affects variable availability with the Availability Index (Equation 3). The last column is encoding Disparity (Equation 4). A positive Disparity is a red flag for mixing codes.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Table 1: Performance indices for various datasets, real (top) and synthetic (bottom). All error bars are on the last significant digit shown (see details in Appendix F). Disparity is significantly positive for all datasets, hinting that the same code should be used for all variables to avoid coding bias. One-Hot encoding is generally best, except in a few cases (see text). The significance of the effect was assessed using error bars, as explained in the appendix (Section F).

	Balanced Accuracy				Availability Index				
	One-Hot	T. Cont.	T. Ord.	R. Ord.	One-Hot	T. Cont.	T. Ord.	R. Ord.	Disparity
Adult Income	0.726	**0.732**	0.718	0.672	**0.036**	-0.080	-0.003	0.022	0.045
Bank Marketing	**0.662**	0.637	0.592	0.554	**0.051**	-0.043	0.029	-0.046	0.043
Cardiovascular Disease	0.621	**0.657**	0.622	0.581	**0.065**	-0.123	0.051	-0.010	0.074
Sentencing	**0.604**	0.578	0.590	0.549	**0.036**	-0.106	0.034	-0.030	0.058
Mushroom	**0.972**	0.849	0.957	0.864	0.019	-0.120	**0.064**	-0.025	0.068
Synthetic Doughnut	**0.977**	0.940	0.875	0.688	**0.069**	0.041	0.003	-0.105	0.066
Synthetic Crossed	**0.965**	0.492	0.768	0.725	**0.126**	-0.080	0.073	0.039	0.076
Synthetic Simple	**1.000**	1.000	1.000	0.862	**0.096**	0.019	0.066	-0.132	0.088
Mean	**0.816**	0.736	0.765	0.687	**0.062**	-0.061	0.040	-0.036	

Let us first analyze Balanced Accuracy results that are obtained by **coding ALL variables in the same way** (first 4 columns), to evaluate whether some codes yield better performance than others overall. At first glance, we notice One-Hot encoding is generally best and Random Ordinal worst. This is true on average for the datasets considered. There are only 2 exceptions: (1) For Cardiovascular disease, Target Encoding is better. This can be explained by the fact that the variables in that dataset are discretized versions of continuous variables, hence they are all ordinal. There is therefore no benefit of One-Hot encoding, known to be tailored to representing nominal variables well, but at the expense of loosing order information for ordinal variables. (2) Another interesting case is that of the Mushroom dataset: there, Target Continuous encoding does not do better than Random Ordinal but Target Ordinal encoding yields similar performance as One-Hot encoding. The good performance of One-Hot encoding is explained by the fact that all variables are nominal. The better performance of Target Ordinal compared to Target Continuous is more subtle. Intuitively, one might think the Target Continuous is a more informative code because Target Ordinal is a discretized code. However, when mean target values are tied, Target Continuous assigns the same value to multiple levels, resulting in a loss of information. Two competing factors determine why one code may be more informative than another, as demonstrated in our synthetic examples designed to highlight this effect. In the Synthetic Crossed example, perfect ties across all levels render Target Continuous non-informative (all levels map to the same value). In contrast, for the Synthetic Doughnut example, Target Continuous outperforms Target Ordinal.

Let us now turn our attention to the Disparity Index. The Disparity is always significant compared to 0, within the error bar, indicating that all variables should be encoded uniformly. This effect is more pronounced in synthetic

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

examples, as real data lacks controlled joint distributions and variable redundancy. Examining Availability reveals a strong correlation with Balanced Accuracy. One-Hot encoding generally maximizes variable availability and is preferable, even with small datasets. Target Ordinal ranks second in Availability but performs worse in Balanced Accuracy, likely due to the issue of ties previously discussed.

5 Conclusion

Our findings indicate that encoding categorical variables differently may introduce bias and we advocate using the same code for all categorical variables to reduce out-of-distribution bias. The optimal code to be chosen depends on the predictive model, the number of levels of variables, the amount of available data, and the specific problem, and can be selected *e.g.,* by cross-validation using the performance metric (e.g. balanced accuracy).

Acknowledgements

Work partially supported by ANR Chair of AI HUMANIA ANR-19-CHIA-0022.

References

[1] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996.

[2] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, nov 2020.

[3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[4] R. K. Halder. Cardiovascular disease dataset, 2020.

[5] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. *An introduction to statistical learning*, volume 112, page 315. Springer, 2013.

[6] M. Kuhn and K. Johnson. *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, 2019.

[7] R. P. Moro, S. and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014.

[8] C. C. S. A. Office. Cook county sentencing data, 2020.

[9] J. Schlimmer. Mushroom. UCI Machine Learning Repository, 1981.

[10] scikit-learn developers. Encoding of categorical variables. `https://inria.github.io/scikit-learn-mooc/python_scripts/03_categorical_pipeline.html`.

[11] A. Zheng and A. Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.", 2018.