NATURALLY PARAMETERISED VAE ON HETEROGENEOUS DATA

Andreas L. Hansen, Yucheng Fu, Phillip C. Højbjerg & Aron D. Jacobsen
DTU COMPUTE Department of Applied Mathematics and Computer Science
Technical University of Denmark
{s194235, s194241, s184984, s194262}@student.dtu.dk

Supervisor: Jes Frellsen
DTU Compute Department of Applied Mathematics and Computer Science
Richard Petersens Plads Building 321, Room 221, 2800 Kgs. Lyngby
jefr@dtu.dk

ABSTRACT

Variational autoencoders (VAEs) are hard to apply to mixed type heterogeneous data. While the current state-of-the-art methods use hierarchical VAEs, we suggest a VAE based on natural parameterisations of the output distributions, called NP-VAE. The NP-VAE is tested on four different datasets and evaluated on root-mean-squared (RMSE) on fully observed and imputed data, where the results show that the RMSE is magnitudes higher than the reported values in existing literature, mainly dominated by the RMSE for numerical attributes. This could indicate that simply using naturally parameterised output distributions is not enough to handle heterogeneous data, however further investigations on the usage of the natural parameterisation is recommended.

1 INTRODUCTION

Variational autoencoders (VAEs) are often used for learning the underlying latent features of complex, high-dimensional data. VAEs particularly excel when each data point follows the same underlying distribution, such as image data, which is modelled as a categorical variable for each pixel. However, VAEs have a hard time handling heterogeneous data, i.e. data with a mixture of numerical, ordinal, categorical, etc. features, where the features must be modelled with different types of distributions Ma et al. (2020); Nazabal et al. (2018).

Existing literature solves this problem by using hierarchical VAEs, where one VAE creates a homogeneous representation of the data, and then independent VAEs - typically called *heads* - are trained for each variable to model the likelihood function of that variable.

In this paper, we attempt another approach without the heads and implement a non-hierarchical VAE for mixed type heterogeneous data where the output distributions - specifically the Gaussian and categorical distributions - are parameterised by their natural parameters. The natural parameterisation VAE (NP-VAE) and a standard VAE without naturally parameterised output distributions are tested on four different datasets, and the results are compared to Ma et al. and Nazabal et al.. The main motivation behind the NP-VAE is that the two distributions can be expressed in the same general form and by their natural parameter, which could make the optimization easier and exclude the need for hierarchical VAE's with heads on heterogeneous data.

2 RELATED WORK

Part of what makes generative models, such as VAEs, so desirable is the ability to do conditional data generation, i.e. given a data observation containing missing features, the model will attempt to reconstruct the entire observation, thus also filling in the blanks. The HI-VAE proposed by Nazabal et al., utilizes an *input drop-out recognition distribution* for their encoder - trained on missing data -

in which the DNNs predicting the μ and σ^2 of the encoder are multilayer perceptrons. This allows missing data to be replaced by zeros, ensuring encoder-output and -derivatives to be independent of missing features.

In Nazabal et al. (2018) the main idea is to use multiple different likelihood models, so-called "heads", for the output distributions of each variable, e.g. a Gaussian likelihood model for real-valued data and a multinomial logit model for categorical data. An intermediate homogeneous representation of the data $\mathbf{Y} = [\mathbf{y}_{n1}, \ldots, \mathbf{y}_{nD}]$ is used, which is generated by a single NN. For each attribute there is an independent NN-head, with the inputs $\mathbf{y}_{nd}$ and $\mathbf{s}_n$, where $\mathbf{s}_n$ is a one-hot encoded vector representing which mean of a Gaussian mixture that generates the latent space $\mathbf{z}_n$. These are combined to form a single heterogeneous-incomplete VAE (HI-VAE).

The method by Ma et al. involves, firstly, training D independent VAEs for each (marginal) variable in the dataset and secondly, training a VAE on top of the latent representations $\mathbf{z}$ from the encoders of the first step, what they call a dependency network. Both use a VampPrior as described by Tomczak.

3 METHODS

3.1 NATURAL PARAMETERISATION

The numerical and categorical can be modelled with the Gaussian and categorical distributions, both of which are part of the exponential family of probability distributions, As noted by Jordan (2009), probability distributions in the natural family can be expressed using a natural parameterisation:

$$p(x|\boldsymbol{\eta}) = h(x) \exp\left(\boldsymbol{\eta}^T T(x) - A(\boldsymbol{\eta})\right),$$

where $\boldsymbol{\eta}$ is a vector of natural parameters, $h(x)$ and $T(x)$ are given functions for a distribution, and the function $A(\boldsymbol{\eta})$ is known as the cumulant function. For numerical variables, the natural parameterisation is a Gaussian distribution, wherein $\boldsymbol{\eta} = \begin{bmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{bmatrix}$, $h(x) = \frac{1}{\sqrt{2\pi}}$, $T(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$, and $A(\boldsymbol{\eta}) = -\frac{\eta_1}{4\eta_2} - \frac{1}{2}\log(-2\eta_2)$. Similarly, the categorical variables are parameterised through the categorical distribution which likewise has a natural parameterisation with

$$\boldsymbol{\eta} = \begin{bmatrix} \log p_1 \\ \vdots \\ \log p_k \end{bmatrix}, h(x) = 1, T(x) = \begin{bmatrix} [x = 1] \\ \vdots \\ [x = 0] \end{bmatrix}, A(\boldsymbol{\eta}) = 0.$$

3.2 VAE MODEL SPECIFICATIONS

3.2.1 MODEL ARCHITECTURE

The *encoder* used here is a simple 3-layer feed-forward neural network with LeakyReLU activations in all layers except the output. The input dimensions D depend on the number of numerical and one-hot encoded categorical attributes - the reason for one-hot encoding is to handle missing data. The hidden dimension is set to $M = 256$ and the output layers has a dimension of $2 \cdot L$ (for μ and $\log \sigma^2$) where the L latent variables corresponds to the total number of attributes in the dataset.

The *decoder* receives the sampled z of dimensions L in the latent space and follows a similar structure as the encoder but outputs to dimension $2 \cdot D_{\text{numerical}} + D_{\text{categorical}}$ to resemble the parameters of their respective distributions.

The NP-VAE and the standard VAE only differ by how their outputs are handled, i.e. what we expect them to be. For the standard VAE, the output for the numerical attributes in the output layer are expected to be their respective μ and $\log \sigma^2$ (the logarithm in order to allow both negative and positive values), to calculate the log-likelihood and sample from the normal distribution from Torch by converting the $\log \sigma^2$ to σ. For the categorical attributes, a $\mathrm{SoftMax}$ operation is applied on the respective values in the output layer to represent probabilities, and then to calculate the log-likelihood and to sample one sample from a multinomial distribution. By comparison and for simplicity, the outputs of the NP-VAE model are expected to be of the η parameters for their respective exponential form, and are then converted back to their corresponding standard parameters to proceed with

the same methods as described in the standard VAE above. The conversion is given in Appendix A.3, where η_2 for the Gaussian distribution is restricted to be negative ($\eta_2 \in \mathbb{R}^-$) to ensure positive variance, and as such the $-\mathrm{SoftPlus}$ operation is applied to all respective η_2 in the output layer (similar to the recommendation for σ^2 by Tomczak (2022)). Furthermore, note that the conversion from NP to standard for the categorical distribution is simply the $\mathrm{Softmax}$-operation, thus there is no difference between the two models for the categorical attributes.

For the *prior* distribution of the latent space, we use a VampPrior, as described by Tomczak (2022), which conditions the posterior of a mixture of Gaussians using pseudo-inputs. An advantage of using a learnable prior like VampPrior is that both the prior $p(\mathbf{z})$ and the posterior $p(\mathbf{z}|\mathbf{x})$ change during training and can match each other. On the other hand, a fixed prior, like the standard normal distribution, is easy to implement, but there might be discontinuities in the latent space, where the prior assigns high probability, but the posterior assigns low probability. (Tomczak (2022)) As a result, a sampled latent variable from the low-density areas of the latent space might not be properly decoded by the decoder. (Li et al. (2021)).

3.2.2 Training & Optimization

The NP-VAE and standard VAE were trained with an AdaMax optimiser with a learning rate of $3 \cdot 10^{-4}$. The loss function has been modified to incorporate a β (also called λ) parameter similar to Higgins et al. (2017) and Asperti & Trentin (2020) only by a decreasing scheduling (decay) of $\frac{\beta}{\text{epoch}_i}$ where epoch_i is the i'th epoch during training, such that the loss function becomes

$$\mathit{Loss} = E_{q_\theta(z|x)}\left[\log p_\varphi(x \mid z)\right] - \frac{\beta}{epoch_i} \cdot KL\left(q_\theta(z \mid x) \| p(z)\right),$$

where θ and φ are the weights of the encoder and decoder respectively.

While the encoder architecture of the proposed NP-VAE resembles that of HI-VAE, it has not been trained on missing data. Evaluation of the imputed data will be expanded on in section 4.

Detailed explanations on these choices are included in Appendix A.2.

3.3 Baseline model specifications

The Baseline follows that of Nazabal et al., predicting the mean of each continuous variable, as well as the most common class for each categorical variables, based on the training data.

3.4 Data

We train and test on four different UCI datasets: Avocado sales, Bank marketing, Boston Housing, and Energy Efficiency. All the datasets feature only numerical and categorical features. Observations with NaN values were dropped, and for the Avocado dataset the date attribute was dropped (due to the special formatting of dates which fit none of our variable types, however a solution might be to convert to UTC timestamps). Further information about the different datasets is shown in Appendix A.5. Binary (true / false) variables were also considered as categorical variables, even though they could also be modelled as Bernoulli distributed variables. Additionally, the datasets did not contain any ordinal variables since this would extend the model to contain more probability distributions.

In addition, all numerical attributes are standardised to $\mu = 0$ and $\sigma^2 = 1$.

4 Evaluation

All models are evaluated using regular RMSE, as well as the imputation error. The imputation error is given as the RMSE of imputed variables, i.e. the RMSE between the reconstructions of imputed variables and said variables before imputation. The implementation of the imputation error follows that of Ma et al., in which $p = 0.5$ of the variables per observation are randomly imputed with the value 0 during testing. The mean-squared-error (MSE) of the imputed reconstructions are calculated per variable, after which the root of the summed MSEs is taken, and divided by the number of variables. The RMSE calculation is seen in Eq. 1, exactly as it was written in Ma et al. (2020).

$$\text{RMSE}_{imp} = \frac{1}{D}\sqrt{\sum_{1\leq d\leq D}\sum_{1\leq n_d\leq N_d}\frac{SE\left(x_{n_d,d}-\hat{x}_{n_d,d}\right)}{N_d}}, \tag{1}$$

where D is the number of variables, N_d is the number of imputed observations for a specific variable d.

For comparisons sake, the regular RMSE follows the same outline, however, N_d is replaced by the total number of observations, N - making the inner-most sum identical to regular MSE.

The RMSE and the imputation error for the baseline will be covered identically, as the limit of the imputation error will be equal to the regular RMSE as the number of observations approaches infinity.

5 RESULTS

Table 1 present the scenario where the model generates a full observation, it shows that overall, the different types of models achieve low errors for categorical variables but high errors for numerical variables. The NLL is lower for canonical paramerisation for all datasets except Bank. The RMSE is lower for Boston with the natural parameterisation, but higher for all other datasets without. The RMSE for the categorical variables is in all cases lower for the naturally parameterised model.

Table 1: Full data generation results. Best result per dataset underlined

	Avocado				***Bank***			
Measure	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$
$\text{VAE}_{\text{REG.}}$	-0.61	$2.41 \cdot 10^4$	$2.41 \cdot 10^4$	0.31	688.84	72.15	72.05	0.11
$\text{VAE}_{\text{EXP.FAM.}}$	-0.2097	$2.64 \cdot 10^5$	$2.64 \cdot 10^5$	0.108	-0.089	172.54	172.50	0.039
BASELINE	-	$3.58 \cdot 10^5$	$3.58 \cdot 10^5$	0.52	-	223.01	222.90	0.11
	Energy				***Boston Housing***			
$\text{VAE}_{\text{REG.}}$	-0.66	1.44	1.09	0.36	-0.36	23.62	23.13	0.48
$\text{VAE}_{\text{EXP.FAM.}}$	-0.45	2.94	2.84	0.10	-0.24	11.80	11.64	0.16
BASELINE	-	10.87	10.39	0.49	-	11.83	11.56	0.28

The results in table 2 are from the imputation case, it is evident that both models are worse than the baseline imputer in all scenarios except the categorical variables for the avocado and energy dataset.

Table 2: Imputation results. Best result per dataset underlined

	Avocado				***Bank***			
Measure	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$
$\text{VAE}_{\text{REG.}}$	-	$2.648 \cdot 10^6$	$2.648 \cdot 10^6$	0.4904	-	1404.142	1404.0185	0.124
$\text{VAE}_{\text{EXP.FAM.}}$	-	$2.46 \cdot 10^6$	$2.46 \cdot 10^6$	0.495	-	1409.926	1409.799	0.126
BASELINE	-	$3.58 \cdot 10^5$	$3.58 \cdot 10^5$	0.52	-	223.01	222.90	0.11
	Energy				***Boston Housing***			
$\text{VAE}_{\text{REG.}}$	-	60.205	59.718	0.4867	-	32.186	31.570	0.615
$\text{VAE}_{\text{EXP.FAM.}}$	-	49.921	49.432	0.4889	-	27.963	27.432	0.531
BASELINE	-	10.87	10.39	0.49	-	11.83	11.56	0.28

6 DISCUSSION

Comparing the results of Table 1, no model is consistently better than others. It is seen that the VAEs typically perform better than the baseline wrt. general RMSE, however, the opposite is true

regarding the imputation error - wherein the baseline consistently performs better than the VAEs. This implies that the VAEs have not learned to do conditional data generation - predicting based solely on observed variables - but instead has learned to reconstruct variables based on the variables themselves. The results for the general RMSE are orders of magnitude larger than the state-of-the-art results presented in Nazabal et al. (2018) and Ma et al. (2020), as seen in table 4. When generating a full observation, we observe that the naturally parameterised model has a far lower error for the categorical variables, whereas the relation switches for the error of the numerical variables. This could be because the categorical variables in the output layer can span more of the dimensions in the output layers (giving them more weight) and are modelling something more simplistic of nature (one-hot encodings), compared to the numerical values that contain only two dimensions in the output layer and a wider span of values. The highest RMSE for a categorical variable is two, where for numerical it can take any value, suggesting a weighting coefficient for the RMSE could be of use - as was mentioned by Ma et al..

A peculiar observation on table 1 and 2 is that the RMSE on the avocado dataset for all models are several magnitudes larger than for the other datasets. Figure 1 reveals that the attributes are long tail distributed, which means some attributes have high means and therefore skews the RMSE calculation.

According to Ma et al. (2020), naively applying a standard VAE on heterogeneous data will likely perform poorly due to the different contributions of the likelihood functions while training, which could be what we observe in this paper. Furthermore, simply introducing natural parameterisation on the output distributions did not improve the performance of the models. One reason for this could be an incomplete implementation of exponential family in the NP-VAE, because the optimisation of the NLL is performed based on the conversion of the natural parameters to the standard, instead of using the likelihood function of the exponential family directly.

As mentioned in the Conditional Data Generation-section, the NP-VAE was not trained on missing data. This means that the proposed model has not learned to understand missing data, and instead could misinterpret the imputed variables as real observations of value 0. Especially considering the standardisation of the numerical attributes (mean to 0) wherein the model might confuse imputed variables with the average value of that attribute - however most attributes might not be be exactly zero. Another explanation could be decay of the regulariser (KL divergence) in the loss function, since the model then looses control on the shape of the latent space, which makes it harder to sample meaningful data. The fact that the VAEs in this paper are much better at reconstructing entire observations, rather than conditional data generation - when Ma et al. found no difference between the two - further enforces this thought and implies the importance of introducing VAEs to missing data during training.

Future work would have to include a deeper intuition about the applications of the exponential family and how they relate with assumptions about the data, and further to calculate the log likelihood directly on the natural parameterization.

7 Conclusion

From our experiments it seems like simply changing the underlying probability distributions to natural parameterisation is not enough to make a VAE handle heterogenous data - however, further investigation on the parameterisation on the output distribution is proposed. Furthermore, it is seen that training on missing data cannot be excluded if aiming for a model able to do conditional data generation.

REFERENCES

Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. CoRR, abs/2002.07514, 2020. URL https://arxiv.org/abs/2002.07514.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Michael Jordan. The exponential family: Basics, 2009. URL https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter8.pdf.

Ruizhe Li, Xutan Peng, and Chenghua Lin. On the latent holes of vaes for text generation, 2021. URL https://arxiv.org/abs/2110.03318.

Chao Ma, Sebastian Tschiatschek, José Miguel Hernández-Lobato, Richard Turner, and Cheng Zhang. Vaem: a deep generative model for heterogeneous mixed type data, 2020. URL https://arxiv.org/abs/2006.11941.

Alfredo Nazabal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes, 2018. URL https://arxiv.org/abs/1807.03653.

Jakub M. Tomczak. Deep Generative Modeling. Springer, 2022.

A APPENDIX

A.1 CODE

Code available at GitHub

A.2 PILOT TRIALS

1. β/λ-VAE'ish: Initially, different β values in the KL term: the range $[0.1, 1, 3, 10]$ were tried, $\beta = 1$ or less gave the best, where for higher values the loss didn't decrease (could be because of posterior collapse). However, initially starting with a low β resulted in exploding log variance in the latent space during training (especially for normalized data [0,1]), so in the end, we settled with a decay strategy (1/epoch), starting at 1. There is a big emphasis on this, since this really did the trick of training the models - even though it not desired, was the model won't learn a proper latent space (maybe ignore it) making sampling inconsistent.
2. Whether to standardise (mean 0 var 1) or min-max normalize (range [0, 1]) the dataset. During an initial trial we found that standardising gives the best results, so we continued with it (also because standardizatation handles outliers better). Batch norm similarNazabal et al. (2018) was also tried out, however it didn't seem to improve the results, but should be experimented further (and with a larger batch size of 32).
3. The prior: VampPrior gave slightly better results than the standard Gaussian prior.

To conclude: the chosen shared parameters for all models were: standardise, VampPrior, $\beta = 1/\text{epoch\_i}$

A.3 CONVERSION FROM NATURAL PARAMETERISATION TO CANONICAL PARAMETERISATION

Gaussian

to natural

$$\eta_1 = -2\mu\eta_2, \quad \eta_2 = -1\frac{1}{2\sigma^2}$$

to canonical

$$\mu = -\frac{\eta_1}{2\eta_2}, \quad \sigma = \sqrt{-\frac{1}{2\eta_2}} \tag{2}$$

Categorical to natural, this is never done, the network just interprets the variables as a natural parameterisation

$$\eta_1, \ldots, \eta_k = \text{softmax}\left(\begin{bmatrix}\log p_1 \\ \vdots \\ \log p_k\end{bmatrix}\right), \qquad \text{where} \sum_{i=1}^{l} \eta_i = 1$$

to canonical

$$p_1, \ldots, p_k = \text{softmax}\left(\begin{bmatrix}e^{\eta_1} \\ \vdots \\ e^{\eta_k}\end{bmatrix}\right), \qquad \text{where} \sum_{i=1}^{k} e^{\eta_i} = 1$$

A.4 REFERENCE RESULTS

External results copied from the respective papers for comparison of results.

A.5 DATA DISTRIBUTIONS

Table 3: Reference Full data generation results. Compare to the NLL in tab. 1

	Avocado				***Bank***			
Measure	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$
MA ET AL.	-0.16				-1.15			
NAZABAL ET AL.	0.04				-0.72			
	Energy				***Boston Housing***			
MA ET AL.	-1.28				-2.16			
NAZABAL ET AL.	0.16				2.11			

Table 4: Reference Imputation results. Compare to the RMSE in tab. 2

	Avocado				***Bank***			
Measure	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$	NLL	$RMSE$	$RMSE_{Num}$	$RMSE_{Cat}$
MA ET AL.	-0.15	0.15			-1.21	0.11		
NAZABAL ET AL.	0.04	0.15			-0.83	0.11		
	Energy				***Boston Housing***			
MA ET AL.	-1.30	0.16			-2.18	0.046		
NAZABAL ET AL.	0.13	0.18			1.58	0.054		

Table 5: Dataset summary

Dataset	**Observations**	**Attributes**	**Numerical attributes**	**Categorical attributes**
Avocado	18249	12	9	3
Bank	4521	17	7	10
Boston	394	14	13	1
Energy	768	10	7	3

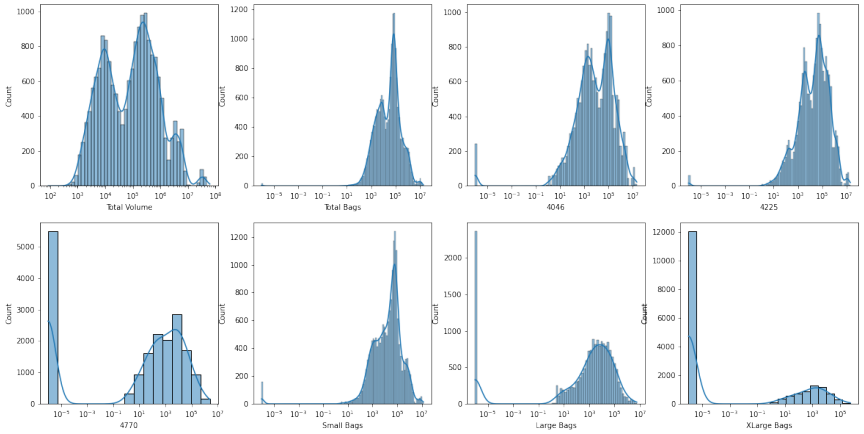

Figure 1: Distribution of variables in the Avocado dataset. Note the log scaling

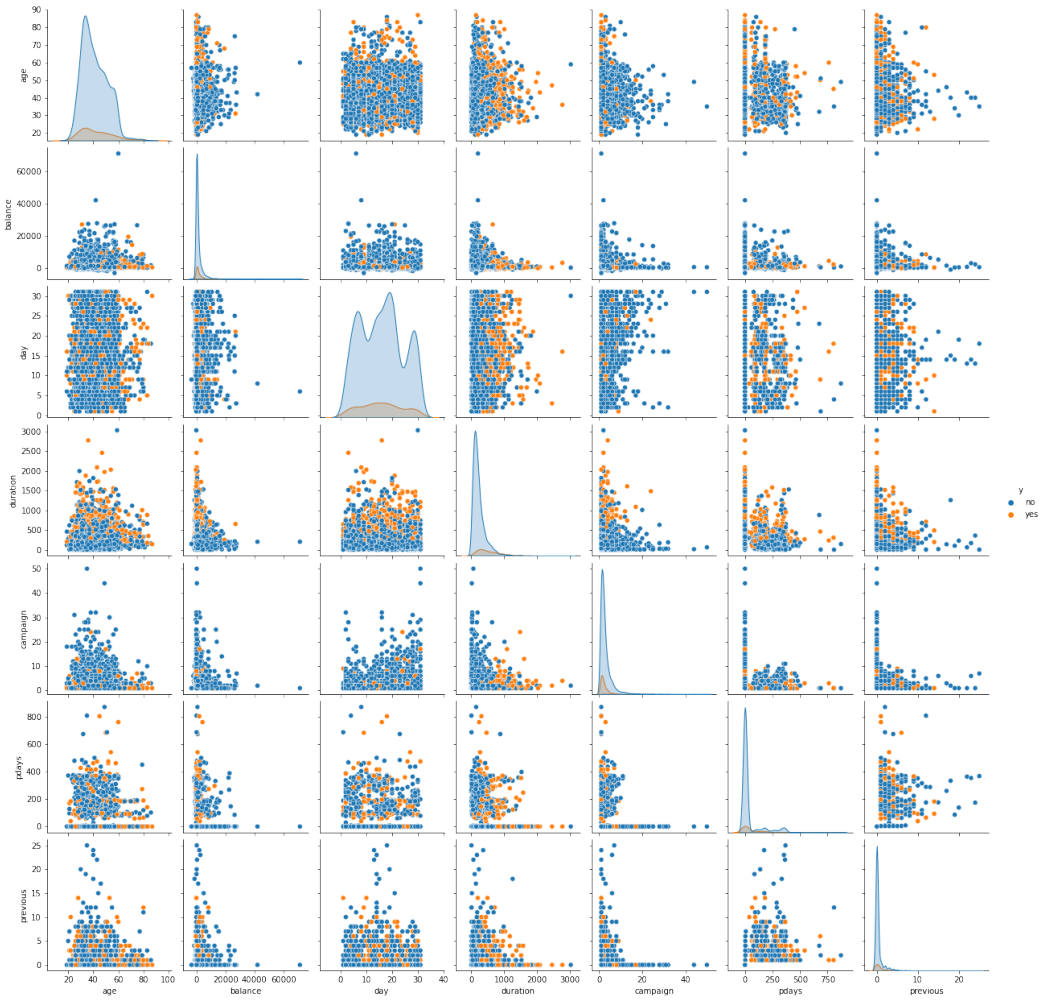

Figure 2: Distribution of variables in the bank dataset

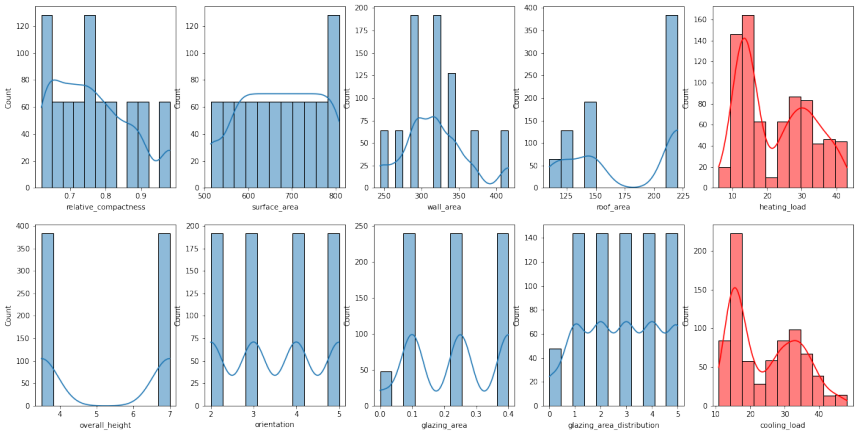

Figure 3: Distribution of variables in the energy dataset

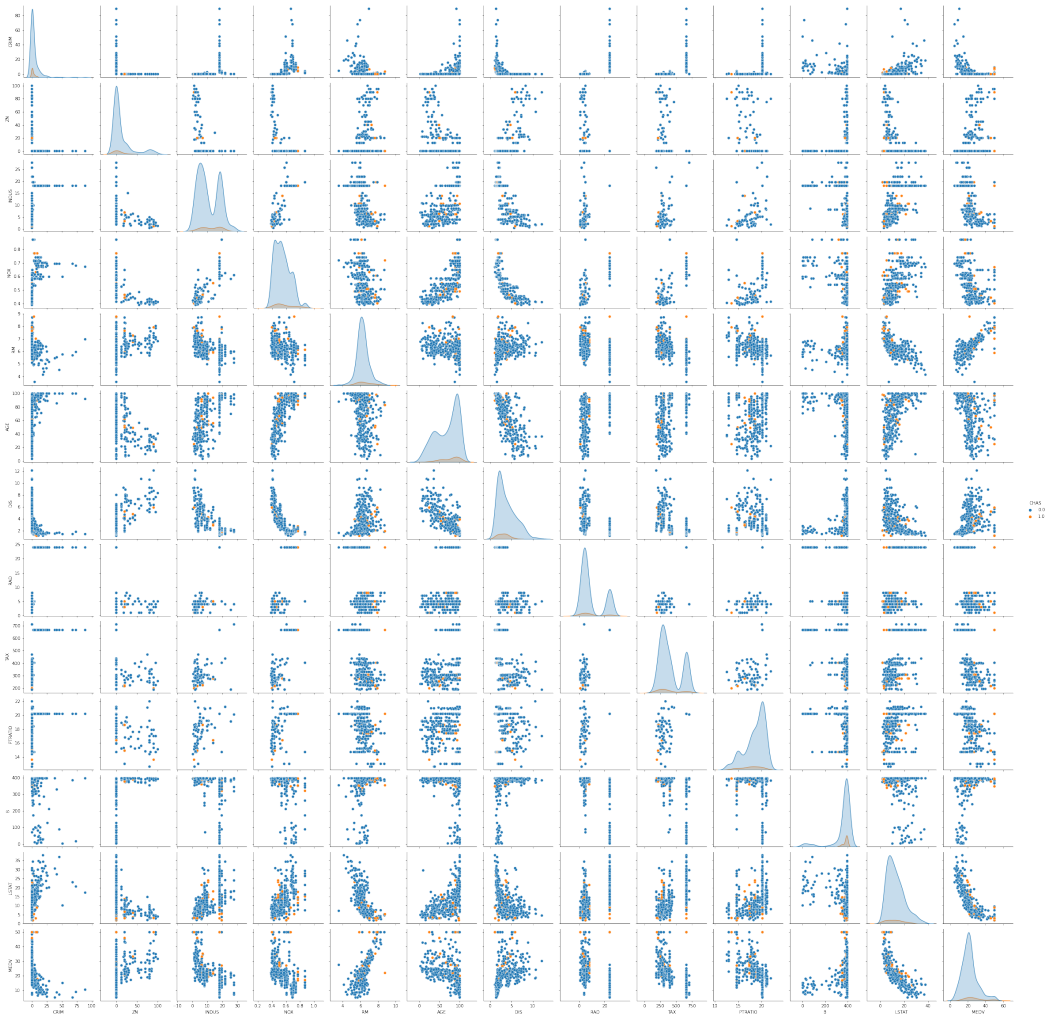

Figure 4: Distribution of variables in the boston dataset