ICML 2024

Binning as a Pretext Task: Improving Self-Supervised Learning In Tabular Domains

LG AI Research Data Intelligence Lab.

Kyungeun Lee, Ye Seul Sim, Hye-Seung Cho, Moonjung Eo,
Suhee Yoon, Sanghyu Yoon, Woohyung Lim

Contact: kyungeun.lee@lgresearch.ai

Backgrounds

What is Tabular Data?

The tabular datasets are structured with rows representing individual samples and columns representing heterogeneous features.

a combination of categorical and numerical features

	Feature #1	Feature #2	Feature #3
Sample #1			
Sample #2			
Sample #3			
Sample #4			

Motivation

Challenges in Tabular Learning

Inherent heterogeneity of tabular datasets (categorical, numerical, and textual)
→ Using an additional module [Gorishniy et al., 2021], [Chen et al., 2022]

Difficulty in learning the irregularities (inconsistencies in distributions)
→ Using a special type of activation functions [Gorishniy et al., 2022]

Superior performance of tree-based machine learning algorithms
(e.g. XGBoost, CatBoost)
→ Infuse the proven strengths of tree-based models into deep networks.
[Grinsztajn et al., 2022], [Grishniy et al., 2022]

Most of the previous efforts have focused on supervised setting only.

How about in "unsupervised"?

Motivation

Inductive bias In Unsupervised Setup

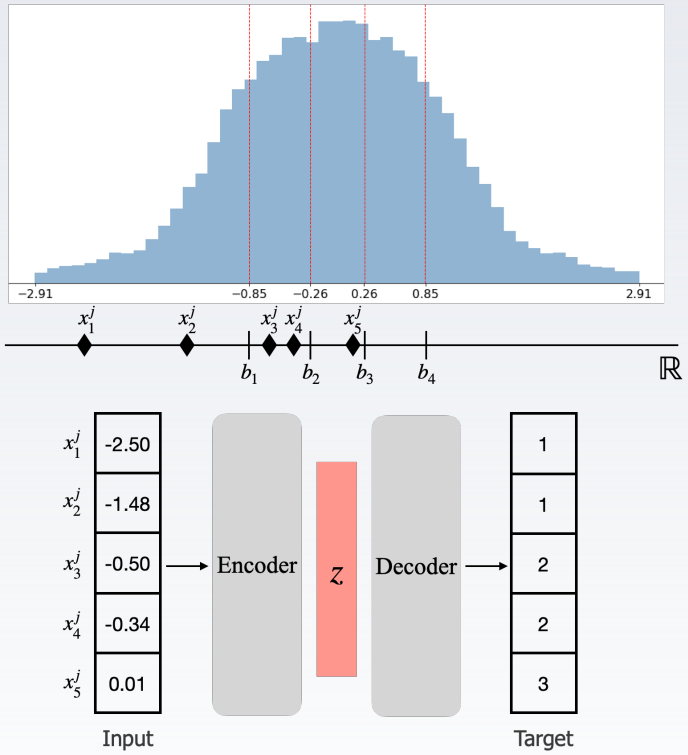
We can learn the representations with the desirable property by introducing a specific objective function, **i.e., defining a specific pretext task.**

- Reconstructing the original values (i.e., auto-encoding): The encoder attempts to impute the masked features by leveraging the correlations present in the input features.

What is the desirable property for tabular representation learning?

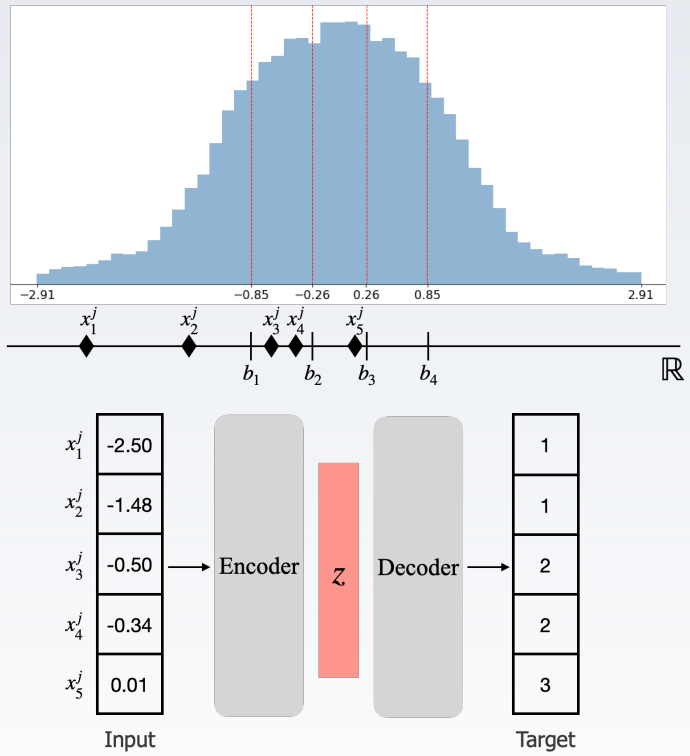
Learning irregularities!
→ Let's define a new pretext task without utilizing the label information.

Binning as a Pretext Task



“Reconstruct bin indices rather than reconstruct the raw values.”

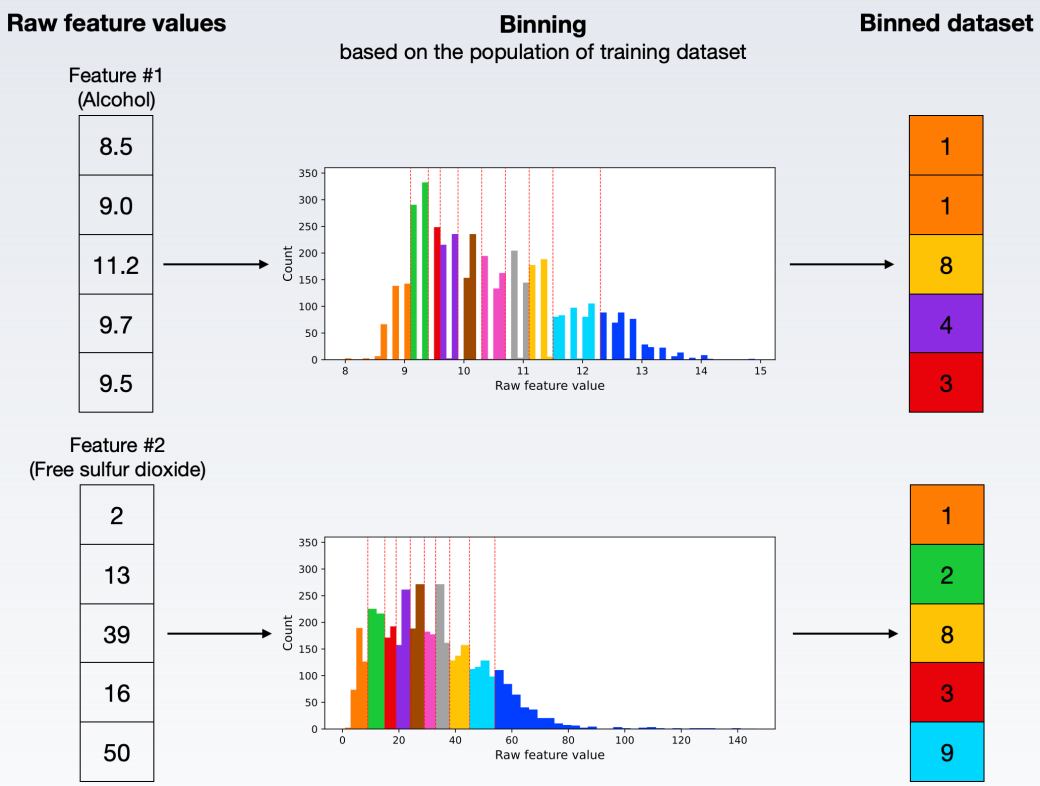
Binning as a Pretext Task



"Reconstruct bin indices rather than reconstruct the raw values."

Once numerical features are discretized into bins based on the quantiles of the training dataset, we optimize the encoder and decoder networks to accurately predict the bin indices given original inputs.

Binning as a Pretext Task



- BinRecon: The bin indices are regarded as the integers. Thus, the target variables are vectors.
- BinXent: The bin indices are regarded as one-hot classes. Thus, the target variable are tensors.

Benefits of Our Method

1. **Employing the inductive bias of capturing irregular functions**
   Deep network learns the function which maps from continuous inputs to discrete targets during SSL.

2. **Mitigating the discrepancy between features**
   For all features, we set the targets as discrete.

3. **Allowing grouping the nearby samples based on the distribution of the training dataset**
   We set the nearby samples to share the targets as bin indices.

4. **Being robust to the minor errors that can yield spurious patterns**
   Deep networks focus on the bin indices, much coarser than the continuous input values.

5. **Standardizing all features into equal sets**
   By making all the features have the same number of bins, all features consist of same elements.

6. **Preventing any uninformative features from dominating during SSL**
   Because all features are standardized into equal sets, all features will contribute equally during training.

7. **Compatibility with any other modifications, e.g. backbone encoder, input transformation**
   Our method is simple as changing the objective function only. So, any modification is allowed.

Comparing SSL Models

Binary classification (Higher is better)

(a) Binary classification (Metric: Accuracy)

Masking	Replacing value	SSL Objective(s)	CH	HI	AD	BM	PH	OS	CS	PO	Average Rank
FALSE	-	ValueRecon	0.810	0.651	0.837	0.899	0.728	0.883	0.709	0.851	7.625
TRUE	Const.	MaskXent	0.807	0.672	0.836	0.899	0.715	0.893	0.708	0.845	7.500
TRUE	Const.	ValueRecon	0.810	0.653	0.839	0.900	0.734	0.884	0.718	0.849	6.000
TRUE	Const.	MaskXent+ValueRecon	0.817	0.669	0.835	0.900	0.724	0.877	0.706	0.837	8.000
TRUE	Random	MaskXent	0.814	0.681	0.843	**0.901**	0.710	0.883	0.706	0.853	6.000
TRUE	Random	ValueRecon	0.811	0.661	0.838	0.898	**0.738**	0.885	0.714	0.842	6.875
TRUE	Random	MaskXent+ValueRecon	0.804	0.647	0.826	0.899	0.715	0.879	0.713	0.861	8.375
FALSE	-	BinXent	0.817	0.683	0.845	**0.901**	0.732	0.886	**0.738**	0.851	3.250
FALSE	-	BinRecon	**0.823**	**0.687**	0.840	0.900	0.737	0.889	0.724	**0.865**	**2.375**
TRUE	Const.	BinRecon	0.820	0.672	0.843	0.899	0.730	**0.896**	0.718	0.858	3.625
TRUE	Random	BinRecon	0.819	0.682	**0.846**	0.898	0.735	0.894	0.718	0.858	3.500

- ✓ Except PH dataset, binning shows the best performance against the other methods.
- ✓ BinRecon without augmentation shows the best performance for three datasets. (Average rank = 2.4)
- ✓ When we simply change the target for reconstruction loss from the raw values(ValueRecon) to bin indices(BinRecon), we gain improvements. (e.g. approximately 3% gain for HI dataset)
  - → It indicates that learning irregular functions is more beneficial than learning smooth functions.

Comparing SSL Models

Multiclass classification (Higher is better)

(b) Multiclass classification (Metric: Accuracy)

Masking	Replacing value	SSL Objective(s)	CO	OT	GE	VO	WQ	AL	HE	MNIST	p-MNIST	Average Rank
FALSE	-	ValueRecon	0.769	0.776	0.527	0.619	0.568	0.931	0.353	0.965	0.928	6.333
TRUE	Const.	MaskXent	0.784	0.777	0.518	0.545	0.547	0.909	0.341	0.793	0.554	9.333
TRUE	Const.	ValueRecon	0.783	0.791	0.557	0.622	0.586	0.931	0.354	0.966	0.925	4.111
TRUE	Const.	MaskXent+ValueRecon	0.750	0.774	0.519	0.610	0.571	0.931	0.360	0.941	0.907	7.444
TRUE	Random	MaskXent	0.763	0.791	0.555	0.549	0.544	0.925	0.336	0.945	0.817	8.000
TRUE	Random	ValueRecon	0.761	0.782	0.538	0.625	0.573	0.930	0.357	0.956	0.934	5.556
TRUE	Random	MaskXent+ValueRecon	0.769	0.779	0.521	0.564	0.519	0.925	0.353	0.945	0.906	8.333
FALSE	-	BinXent	0.742	0.781	0.517	0.600	0.565	0.903	0.354	0.956	0.908	8.333
FALSE	-	BinRecon	0.784	0.783	0.544	0.625	**0.592**	0.935	0.357	0.964	0.950	3.556
TRUE	Const.	BinRecon	0.812	0.792	0.559	0.647	0.581	0.943	0.359	0.974	0.964	2.222
TRUE	Random	BinRecon	**0.814**	**0.794**	**0.580**	**0.655**	0.574	**0.949**	**0.365**	**0.981**	**0.971**	**1.333**

- ✓ BinRecon with masking consistently leads the additional improvements, while BinXent does not work well.
  - → The order information is important for multiclass classification.

Comparing SSL Models

Regression (Lower is better)

(c) Regression (Metric: RMSE)

Masking	Replacing value	SSL Objective(s)	CA	HO	FI	MI	KI	CPU	DIA	EL	Average Rank
FALSE	-	ValueRecon	0.749	4.241	13900.720	0.784	0.163	3.876	1016.641	0.399	8.625
TRUE	Const.	MaskXent	0.709	4.548	13473.750	0.788	0.185	4.475	1259.744	0.396	8.875
TRUE	Const.	ValueRecon	0.693	4.086	13518.683	0.778	0.160	3.728	952.444	0.394	5.000
TRUE	Const.	MaskXent+ValueRecon	0.700	4.157	13915.875	0.775	0.174	5.644	2797.034	0.398	8.750
TRUE	Random	MaskXent	0.677	4.297	13826.641	0.782	0.176	3.951	1358.135	0.388	7.875
TRUE	Random	ValueRecon	0.713	4.127	13668.988	0.777	0.162	3.760	986.306	0.396	6.500
TRUE	Random	MaskXent+ValueRecon	0.701	4.136	14107.645	0.780	0.166	4.506	1917.875	0.397	8.750
FALSE	-	BinXent	0.690	4.116	**13038.762**	0.776	0.170	3.717	1207.923	0.383	4.875
FALSE	-	BinRecon	0.622	3.766	13453.309	**0.767**	**0.158**	3.208	897.645	0.370	2.250
TRUE	Const.	BinRecon	0.634	3.765	13208.133	0.773	**0.158**	**3.156**	957.801	0.371	2.375
TRUE	Random	BinRecon	**0.619**	**3.703**	13075.474	0.773	0.160	3.183	**870.283**	**0.368**	**1.625**

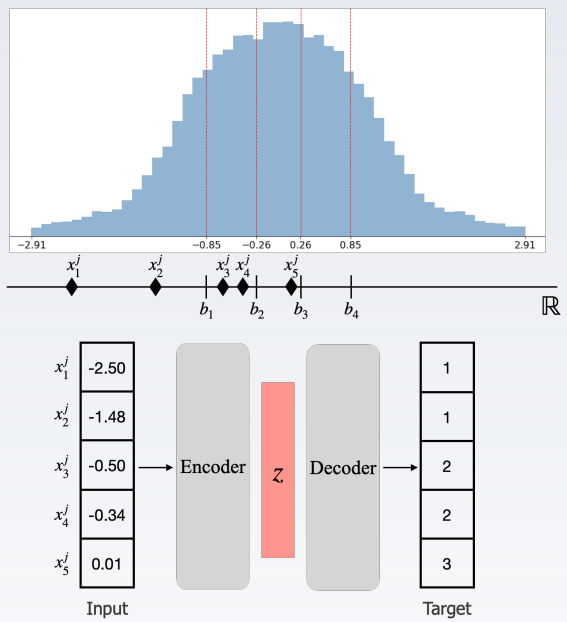
- ✓ BinRecon with masking as the random values shows the best performance. (Average rank = 1.6)
- ✓ Regression tasks exhibit the most significant improvements with the binning task.
  e.g. (Performance gain against the best baselines) HO dataset: 10.27%, DIA dataset: 8.63%, CA dataset: 8.57%

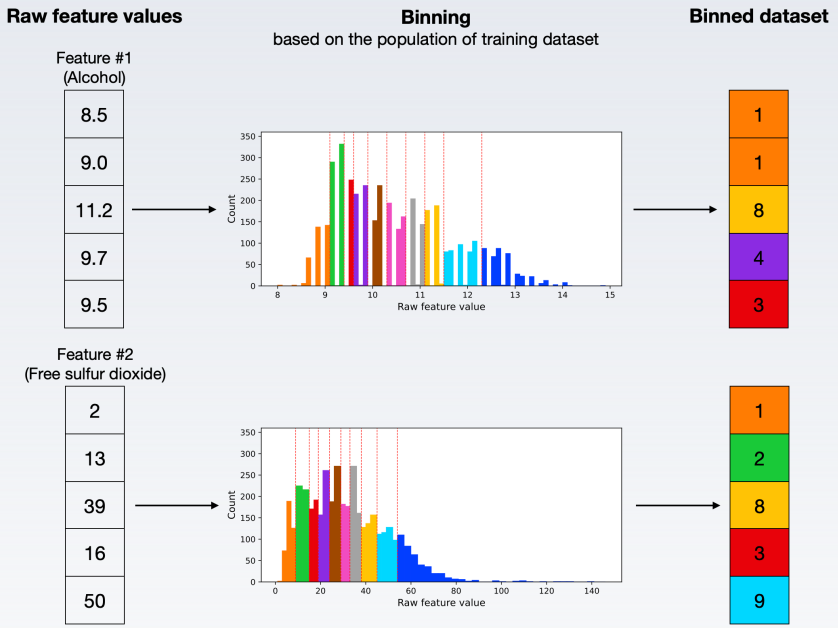
Binning as Pre-training

Training network and method	Binary classification				Multiclass classification						Regression		
	HI ↑	PH ↑	OS ↑	PO ↑	CO ↑	GE ↑	VO ↑	AL ↑	HE ↑	MNIST ↑	CA ↓	HO ↓	FI ↓
Tree-based machine learning algorithms													
XGBoost	0.726	0.721	0.840	0.711	0.969	0.683	0.699	0.924	0.348	0.977	0.434	3.152	10372.778
CatBoost	0.727	0.728	0.833	0.897	0.967	0.692	0.711	0.948	0.386	0.979	0.430	3.093	10636.322
Deep learning methods													
MLP	0.714	0.724	0.896	0.901	0.968	0.659	0.692	0.960	0.378	0.983	0.513	3.146	10086.080
ResNet	0.688	0.728	0.885	0.795	0.729	0.484	0.550	0.220	0.229	0.826	0.706	4.004	10226.508
TabNet (Arik & Pfister, 2021; Gorishniy et al., 2021)	0.719	-	-	-	0.957	0.587	0.568	0.954	0.378	0.968	0.510	-	-
NODE (Popov et al., 2019; Gorishniy et al., 2021)	0.726	-	-	-	0.958	-	-	0.918	0.359	-	0.464	-	-
DCN V2 (Wang et al., 2021; Gorishniy et al., 2021)	0.723	-	-	-	0.965	-	-	0.955	0.385	-	0.484	-	-
SCARF (Bahri et al., 2021)	0.585	0.710	0.878	0.838	0.654	0.325	0.289	0.731	0.050	0.801	1.084	5.595	13632.255
SAINT (Somepalli et al., 2021)	0.713	0.728	0.886	0.877	0.943	0.691	0.713	0.932	0.378	0.981	0.581	6.186	19366.582
FT-Transformer (Gorishniy et al., 2021)	0.729	0.724	0.882	0.890	0.970	0.664	0.705	0.960	**0.391**	0.966	0.487	3.319	10206.127
PLR (MLP-Ensemble) (Gorishniy et al., 2022)	0.734	-	-	-	0.970	0.674	-	-	-	-	0.467	3.050	-
PLR (FT-T-Ensemble) (Gorishniy et al., 2022)	0.734	-	-	-	**0.972**	0.646	-	-	-	-	0.464	3.162	-
T2G-Former (Yan et al., 2023)	0.734	0.746	0.884	0.881	0.968	0.656	0.717	0.964	**0.391**	0.985	**0.455**	3.138	10750.850
SSL(MaskXent)+Fine-tuning	0.725	0.751	0.892	0.897	0.970	0.698	0.717	0.963	0.383	0.985	0.479	3.086	10204.559
SSL(ValueRecon)+Fine-tuning	0.719	0.731	0.894	0.899	0.969	0.690	0.712	0.963	0.381	0.984	0.478	3.119	10333.400
SSL(MaskXent+ValueRecon)+Fine-tuning	0.727	0.737	0.894	0.896	0.968	0.658	0.709	0.959	0.382	0.984	0.475	3.257	10708.780
Ours – SSL(BinRecon)+Fine-tuning	**0.737**	**0.764**	**0.897**	**0.904**	0.971	**0.720**	**0.728**	**0.966**	0.388	**0.986**	0.464	**2.989**	**9757.950**

SSL based on the binning task can be an effective method to learn both the good representations and the initial weights for fine-tuning, surpassing both tree-based and other supervised deep learning methods.

Summary





We suggest a novel pretext task based on binning which can manipulate the unique properties of tabular datasets.
The binning task can effectively address the challenges in tabular SSL, including mitigating the feature heterogeneity and learning the irregularities.

Thank you ☺

Visit our paper at Poster Session 1

Paper

Github