Revisiting Nearest Neighbor for Tabular Data: A Deep Tabular Baseline Two Decades Later

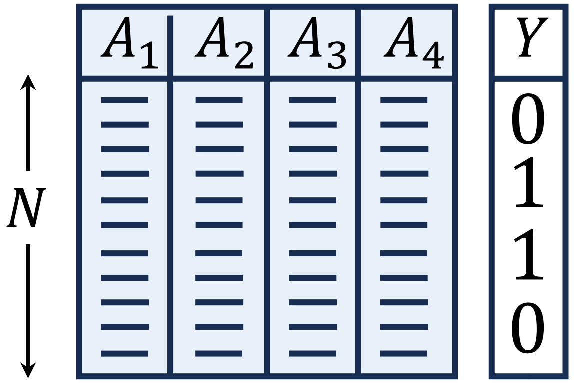
Han-Jia Ye, Huai-Hong Yin, De-Chuan Zhan
Nanjing University

Wei-Lun (Harry) Chao
The Ohio State University

Paper link: https://openreview.net/forum?id=JytL2MrlLT
Code : https://github.com/LAMDA-Tabular/TALENT

Learning with Tabular Data

- Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$,
  - N instances (rows) and d columns (attributes)
  - $\boldsymbol{x}_i \in \mathbb{R}^d$, with *categorical* and *numerical* features/attributes
  - $y_i \in \{0,1\}$ for binary classification, $y_i \in \{1, \dots, C\}$ for C-way classification, and $y_i \in \mathbb{R}$ for regression



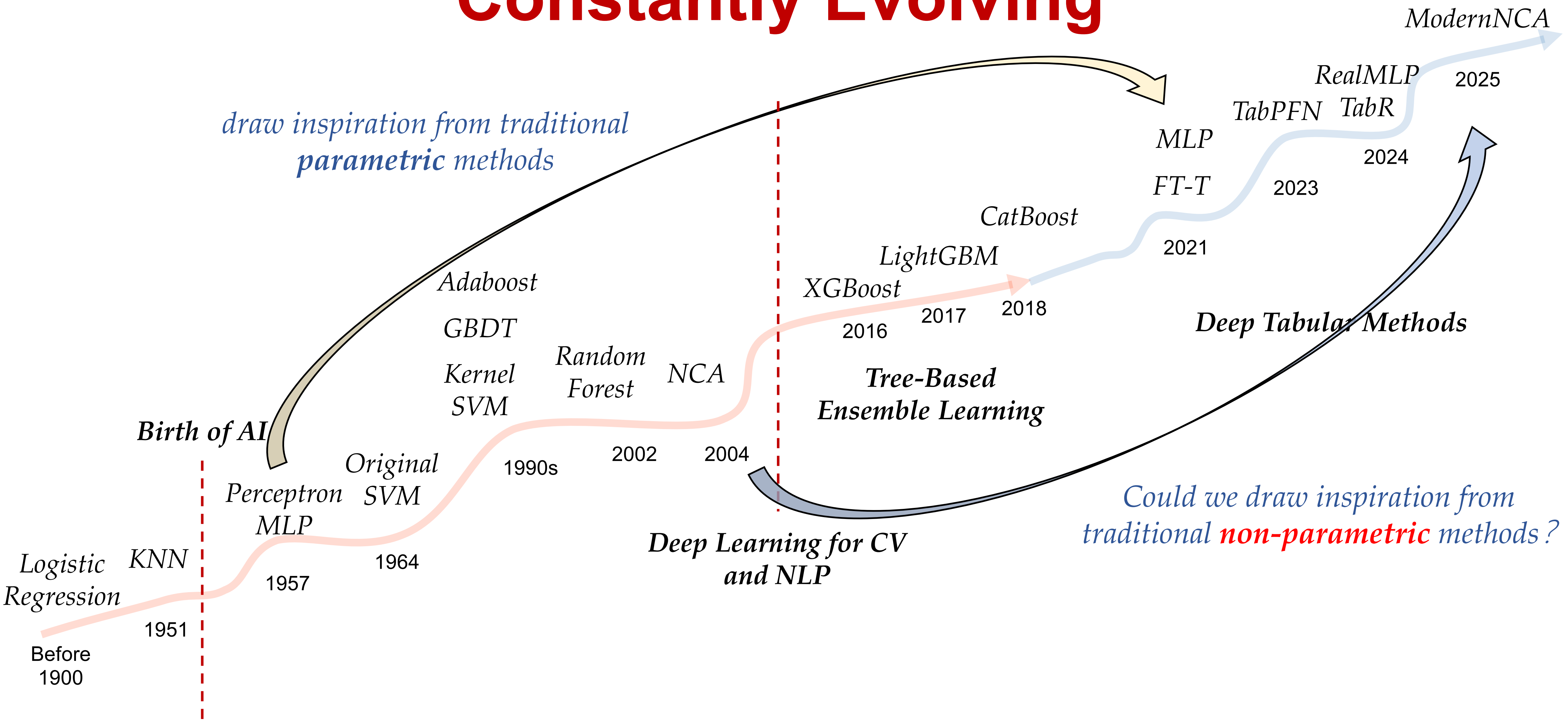
Binary Classification

- The goal is to learn a mapping f. Given an unseen instance $\boldsymbol{x}^*$,

$$\hat{y}^* = f(\boldsymbol{x}^*, \mathcal{D})$$

- Other related tabular tasks
  - Clustering, imputation, generation, anomaly detection, etc.

Methods for Tabular Datasets are Constantly Evolving



Neighborhood Component Analysis (NCA)

- Neighborhood Component Analysis (NCA) for classification [Goldberger et al., NIPS'04]
- The probability that $\boldsymbol{x}_j$ locates in the neighborhood of $\boldsymbol{x}_i$

$$\Pr\left(\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i;\mathcal{D}) \mid \boldsymbol{x}_i, \mathcal{D}, \boldsymbol{L}\right) = \frac{\exp\left(-\mathrm{dist}^2\left(\boldsymbol{L}^\top \boldsymbol{x}_i, \boldsymbol{L}^\top \boldsymbol{x}_j\right)\right)}{\sum_{(\boldsymbol{x}_l, y_l)\in\mathcal{D}, \boldsymbol{x}_l \neq \boldsymbol{x}_i} \exp\left(-\mathrm{dist}^2\left(\boldsymbol{L}^\top \boldsymbol{x}_i, \boldsymbol{L}^\top \boldsymbol{x}_l\right)\right)}$$

- The probability that an instance $\boldsymbol{x}_i$ is classified as the class y_i is

$$\Pr(\hat{y}_i = y_i \mid \boldsymbol{x}_i, \mathcal{D}, \boldsymbol{L}) = \sum_{(\boldsymbol{x}_j, y_j)\in\mathcal{D} \wedge y_j = y_i} \Pr\left(\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i;\mathcal{D}) \mid \boldsymbol{x}_i, \mathcal{D}, \boldsymbol{L}\right)$$

- NCA maximizes the sum of $\Pr(\hat{y}_i = y_i \mid \boldsymbol{x}_i, \mathcal{D}, \boldsymbol{L})$ over all instance in $\mathcal{D}$ and use KNN over the space projected by $\boldsymbol{L}$ in the test phase.

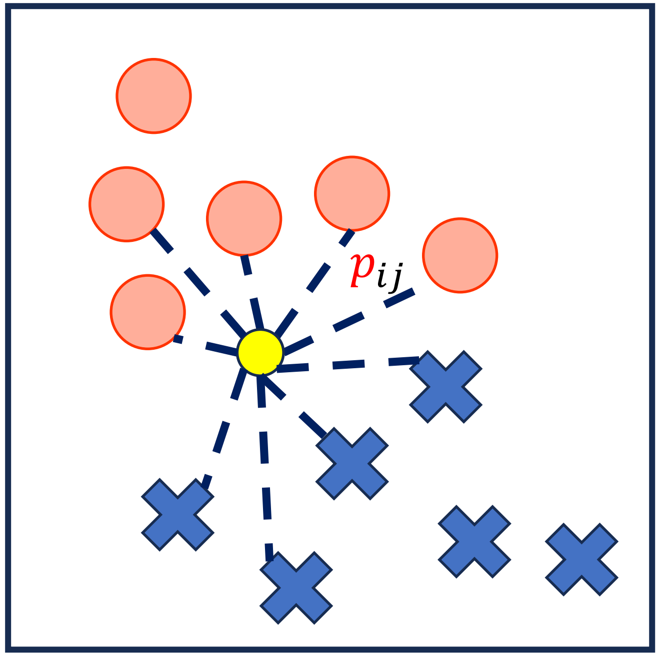
Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. NIPS, 2004.

Our First Attempt

- The learning objective: soft-NN

$$\hat{y}_i = \sum_{(\boldsymbol{x}_j, y_j) \in \mathcal{D}} p_{ij}\, y_j = \sum_{(\boldsymbol{x}_j, y_j) \in \mathcal{D}} \frac{\exp\left(-\mathrm{dist}^2\left(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\right)\right)}{\sum_{(\boldsymbol{x}_l, y_l) \in \mathcal{D}, \boldsymbol{x}_l \neq \boldsymbol{x}_i} \exp\left(-\mathrm{dist}^2\left(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_l)\right)\right)} y_j$$

- $\phi(\boldsymbol{x}_i) = \boldsymbol{L}^\top \boldsymbol{x}_i$. Minimize the sum of negative **log probability**.



- Prediction Strategy
  - Use Soft-NN to make prediction
  - Do not restrict L to project into low-dimensional space
  - Use SGD instead of L-BFGS

Denote this improved *linear* version of NCA as **L-NCA**

Our Second Attempt

- Architectures
  - Define ϕ as MLP, and a single block is implemented by [Gorishniy et al., NIPS'21]

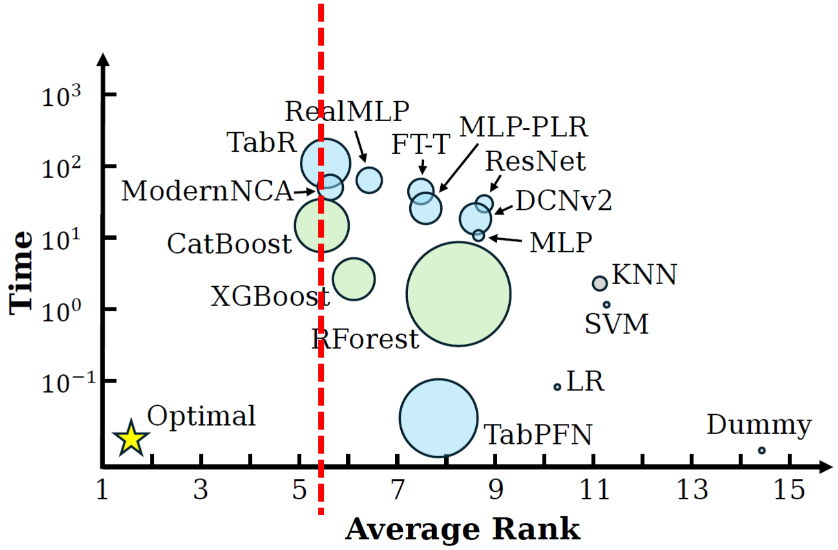
$$g(\boldsymbol{x}_i) = \text{Linear}(\text{Dropout}((\text{ReLU}(\text{Linear}(\text{BatchNorm}(\boldsymbol{x}_i))))))$$

  - One-hot encoding for categorical features
  - PLR (lite) encoding for numerical features
- Stochastic Neighborhood Sampling (SNS)
  - Sample ***a subset of training set*** in a mini-batch to act as the neighbor candidates, while use ***the whole training set*** during the inference.
- Distance Function: Euclidean distance

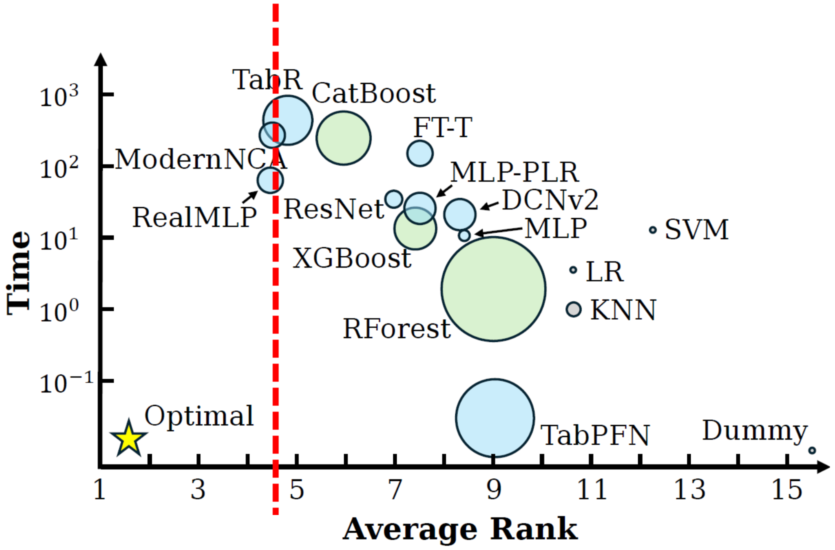
Denote this improved *nonlinear* version of NCA as **ModernNCA (MNCA)**

Experiments

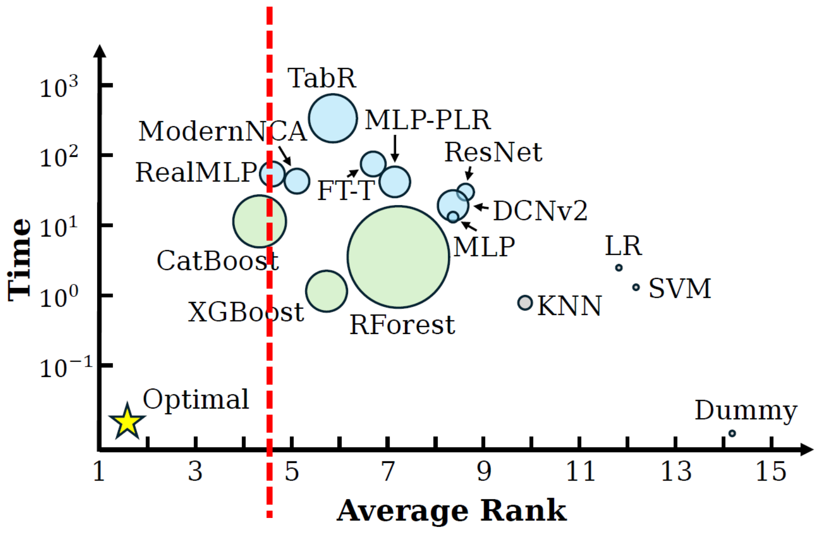
- 300 datasets in total
  - 120 binary classification, 80 multi-class classification, 100 regression
- Average rank vs. Training time vs. Model Size



Binary Classification



Multi-Class Classification

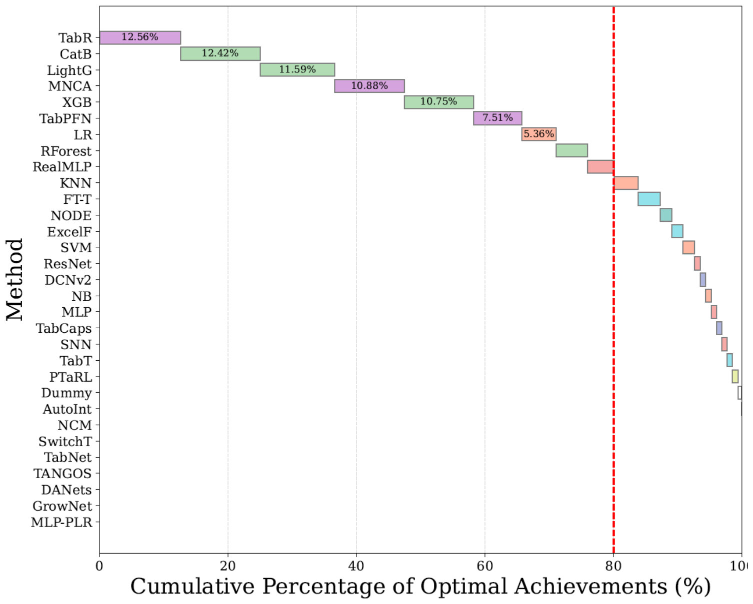


Regression

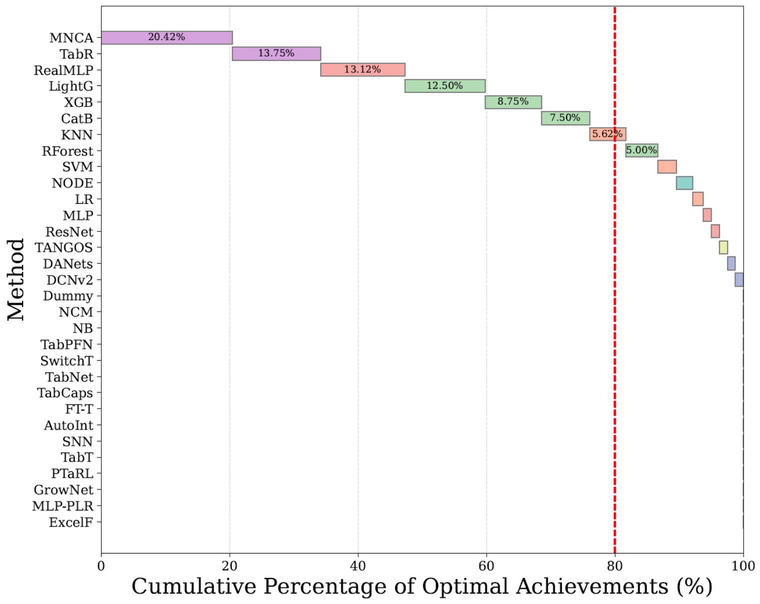
Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, De-Chuan Zhan. A Closer Look at Deep Learning Methods on Tabular Datasets. CoRR 2024.

Experiments

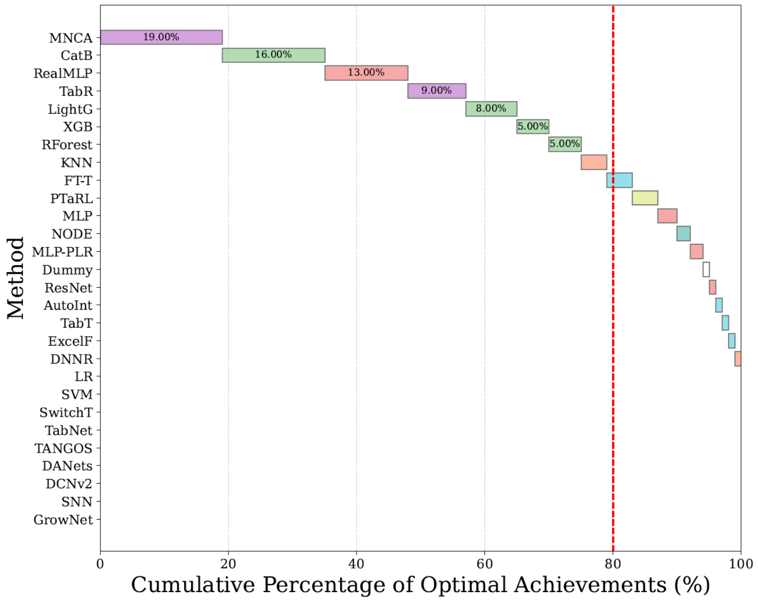
- 300 datasets in total
  - 120 binary classification, 80 multi-class classification, 100 regression
- PAMA (Probability of Achieving the Best Accuracy)



Binary Classification



Multi-Class Classification



Regression

Ablations

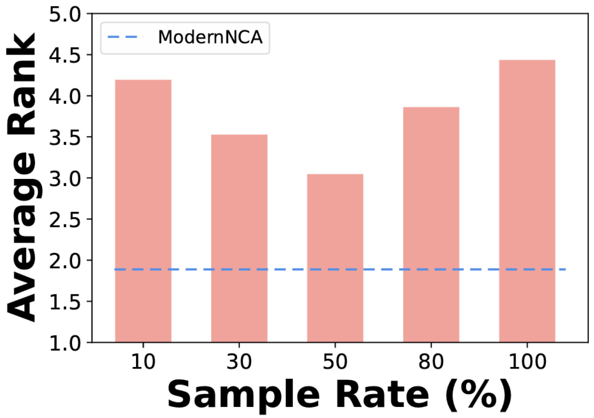
- Tiny-Benchmark with 45 datasets
- From NCA to L-NCA on 27 classification datasets

	High dimension	SGD optimizer	Log loss	Soft-NN prediction	Average rank
NCAv0					4.400
NCAv1	✓				3.708
NCAv2	✓	✓			3.296
NCAv3	✓	✓	✓		3.192
NCAv4	✓	✓	✓	✓	2.962
MLP	✓	✓	✓		3.000

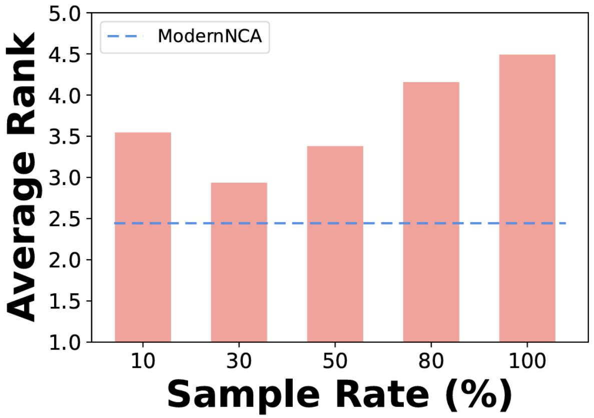
Ablations

- Tiny-Benchmark with 45 datasets
- Average rank changes from L-NCA to M-NCA

	MLP	Linear	w/ LayerNorm	ResNet
Classification	2.333	2.778	2.537	2.352
Regression	2.333	2.433	2.528	2.806



(a) classification



(b) regression

Come to out poster!

Poster Session 3
Fri 25 Apr 10 a.m. CST — 12:30 p.m. CST

- We revisit and enhance one of the most representative neighborhood-based methods, NCA, by incorporating modern deep learning techniques.
- **ModernNCA** establishes itself as **a very strong baseline for deep tabular prediction**, frequently outperforming both tree-based and deep learning models across a wide range of classification and regression tasks.

For more discussions, please contact yehj@nju.edu.cn

Tabular Toolbox

LAMDA
Learning And Mining from DatA

TALENT

Arxiv 2407.04057 | 知 中文解读 | Stars 435 | last commit today

PYTORCH 2.0.1 | PYTHON 3.10 | docs passing | license MIT

TALENT: A Tabular Analytics and Learning Toolbox

https://github.com/LAMDA-Tabular/TALENT

30+ deep learning methods (with methods on NeurIPS'24/ICLR'25), unifying interfaces, customizability.

Si-Yang Liu and Hao-Run Cai and Qi-Le Zhou and Han-Jia Ye. TALENT: A Tabular Analytics and Learning Toolbox. CoRR 2024