

Tabular deep learning: missingness, uncertainty, foundation models

Jill-Jênn Vie

Featuring many works from folks at Soda, Inria



March 3, 2025

Tabular learning: most data is tabular

```
!pip install pandas skrub
import pandas as pd
df = pd.read_csv('employees_salaries.csv')
from skrub import TableReport
TableReport(df).open()
```

Table Stats Distributions Associations Filter columns

Click a table cell for more info about its column.

	gender	department	department_name	division	assignment_category	employee_position_title	date_first_hired	year_first_hired	salary
0	F	POL	Department of Police	MSB Information Mgmt and Tec...	Fulltime-Regular	Office Services Coordinator	09/22/1986	1986	69222.18
1	M	POL	Department of Police	ISB Major Crimes Division Fugiti...	Fulltime-Regular	Master Police Officer	09/12/1988	1988	97392.47
2	F	HHS	Department of Health and Hum...	Adult Protective and Case Mana...	Fulltime-Regular	Social Worker IV	11/19/1989	1989	104717.28
3	M	COR	Correction and Rehabilitation	PRRS Facility and Security	Fulltime-Regular	Resident Supervisor II	05/05/2014	2014	52734.57
4	M	HCA	Department of Housing and Co...	Affordable Housing Programs	Fulltime-Regular	Planning Specialist III	03/05/2007	2007	93396.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
9223	F	HHS	Department of Health and Hum...	School Based Health Centers	Fulltime-Regular	Community Health Nurse II	11/03/2015	2015	72094.53
9224	F	FRS	Fire and Rescue Services	Human Resources Division	Fulltime-Regular	Fire/Rescue Division Chief	11/28/1988	1988	169543.85
9225	M	HHS	Department of Health and Hum...	Child and Adolescent Mental He...	Parttime-Regular	Medical Doctor IV - Psychiatrist	04/30/2001	2001	102736.52
9226	M	CCL	County Council	Council Central Staff	Fulltime-Regular	Manager II	09/05/2006	2006	153747.5
9227	M	DLC	Department of Liquor Control	Licensure, Regulation and Educa...	Fulltime-Regular	Alcohol/Tobacco Enforcement S...	01/30/2012	2012	75484.08

9,228 rows × 9 columns.

Scikit-learn is downloaded 3M times per day. Pandas: 10M times per day.

skrub's TableReport

Like `df.info()` or `df.describe()`

Table	Stats	Distributions	⚠ Associations	Filter columns																									
Column	↓ ↑	↓ ↑	Column name	↓ ↑	↓ ↑	dtype	↓ ↑	↓ ↑	Null values	↓ ↑	↓ ↑	Unique values	↓ ↑	↓ ↑	Mean	↓ ↑	↓ ↑	Std	↓ ↑	↓ ↑	Min	↓ ↑	↓ ↑	Median	↓ ↑	↓ ↑	Max	↓ ↑	↓ ↑
0			gender			ObjectDType			17 (0.2%)			2 (< 0.1%)																	
1			department			ObjectDType			0 (0.0%)			37 (0.4%)																	
2			department_name			ObjectDType			0 (0.0%)			37 (0.4%)																	
3			division			ObjectDType			0 (0.0%)			694 (7.5%)																	
4			assignment_category			ObjectDType			0 (0.0%)			2 (< 0.1%)																	
5			employee_position_title			ObjectDType			0 (0.0%)			443 (4.8%)																	
6			date_first_hired			ObjectDType			0 (0.0%)			2264 (24.5%)																	
7			year_first_hired			Int64DType			0 (0.0%)			51 (0.6%)			2.00e+03			9.33			1,965			2,005			2,016		
8			salary			Float64DType			0 (0.0%)			3403 (36.9%)			7.34e+04			2.91e+04			9.20e+03			6.94e+04			3.03e+05		

skrub's TableReport

Histograms and most frequent values

Table Stats Distributions [Associations](#) Filter columns ▼



Select all Deselect all

☐ gender ObjectDType

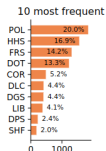
Null values: 17 (0.2%)
Unique values: 2 (<0.1%)



► Most frequent values

☐ department ObjectDType

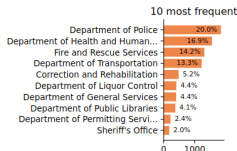
Null values: 0 (0.0%)
Unique values: 37 (0.4%)



► Most frequent values

☐ department_name ObjectDType

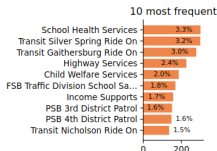
Null values: 0 (0.0%)
Unique values: 37 (0.4%)



► Most frequent values

☐ division ObjectDType

Null values: 0 (0.0%)
Unique values: 694 (7.5%)



► Most frequent values

☐ assignment_category ObjectDType

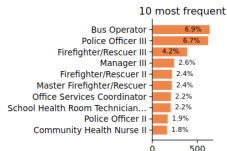
Null values: 0 (0.0%)
Unique values: 2 (<0.1%)



► Most frequent values

☐ employee_position_title ObjectDType

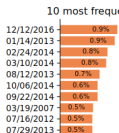
Null values: 0 (0.0%)
Unique values: 443 (4.8%)



► Most frequent values

☐ date_first_hired ObjectDType

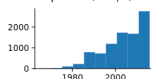
Null values: 0 (0.0%)
Unique values: 2,264 (24.5%)



► Most frequent values

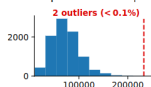
☐ year_first_hired Int64DType

Null values: 0 (0.0%)
Unique values: 51 (0.6%)
Mean ± Std: 2,00e+03 ± 9.33
Median ± IQR: 2,005 ± 14
Min | Max: 1,965 | 2,016



☐ salary Float64DType

Null values: 0 (0.0%)
Unique values: 3,403 (36.9%)
Mean ± Std: 7.34e+04 ± 2.91e+04
Median ± IQR: 6.94e+04 ± 3.94e+05
Min | Max: 9.20e+03 | 3.03e+05



skrub's TableReport

Table Stats Distributions **⚠ Associations**

Column 1	Column 2	Cramér's V
department	department_name	1.00
assignment_category	salary	0.773
division	assignment_category	0.580
assignment_category	employee_position_title	0.486
division	employee_position_title	0.458
department_name	employee_position_title	0.408
department	employee_position_title	0.408
department_name	assignment_category	0.392
department	assignment_category	0.392
gender	department_name	0.375
gender	department	0.375
department_name	division	0.364
department	division	0.364
employee_position_title	salary	0.308
gender	employee_position_title	0.285
gender	division	0.264
gender	assignment_category	0.260
employee_position_title	date_first_hired	0.244
division	salary	0.216
year_first_hired	salary	0.211

The table below shows the strength of association between the most similar columns in the dataframe. [Cramér's V](#) statistic is a number between 0 and 1. When it is close to 1 the columns are strongly associated — they contain similar information. In this case, one of them may be redundant and for some models (such as linear models) it might be beneficial to remove it.

Example: electronic health records (EHR)

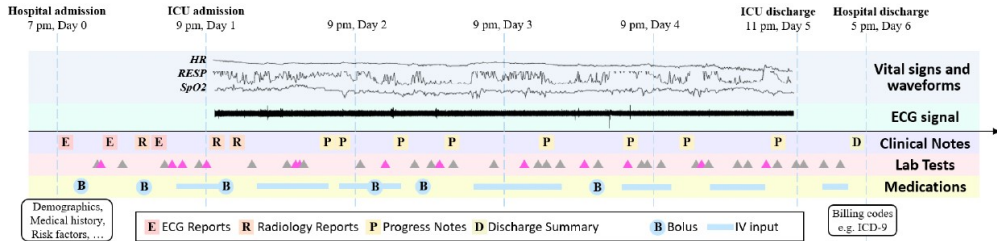


Figure 1: Illustration of various modalities recorded in an EHR during a hospital visit. The horizontal axis represents the time. Multi-modal data including demographics, clinical notes, laboratory test results, medication prescriptions, vital signs and ECG waveform are recorded at various times during the hospital visit.

Kejing Yin et al. (2019). “Learning phenotypes and dynamic patient representations via RNN regularized collective non-negative tensor factorization”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 1246–1253

Challenges of (tabular) data

- ▶ Continuous and categorical variables
- ▶ Missing data, measurements at irregular time intervals
- ▶ Human errors

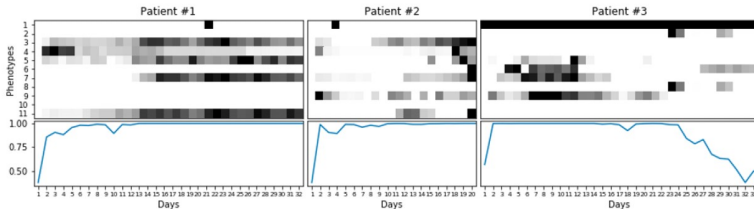


Figure 2: An illustration of the dynamic patient representations of ICU stays of three different patients. As illustrated, Patient #1 is first having “Chronic Heart Disease” for the first several days of the ICU stay. The dynamic representation then shows that the patient further suffers from “Other Disease of the Lung”, “Cardiac Dysrhythmias”, “Acute Kidney Failure”, and “Cardiac Dysrhythmias with Heart Failure” at the same time. Essentially, this describes a clinical scenario in which the patient is first having a problem of chronic heart disease which cannot be properly treated, and he deteriorates and exhibits multiple organ failure. It is also noticed that the predicted mortality score is generally going up and eventually reached 1.0. [57]

Accurate predictions on small data with a tabular foundation model

<https://doi.org/10.1038/s41586-024-08328-6>

Received: 17 May 2024

Accepted: 31 October 2024

Published online: 8 January 2025

Open access

 Check for updates

Noah Hollmann^{1,2,3,7}, Samuel Müller^{1,7}, Lennart Purucker¹, Arjun Krishnakumar¹, Max Körfer¹, Shi Bin Hoo¹, Robin Tibor Schirrmeyer^{4,5} & Frank Hutter^{1,3,6}

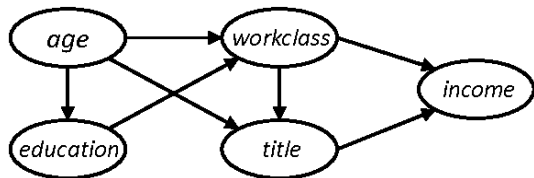
Tabular data, spreadsheets organized in rows and columns, are ubiquitous across scientific fields, from biomedicine to particle physics to economics and climate science^{1,2}. The fundamental prediction task of filling in missing values of a label column based on the rest of the columns is essential for various applications as diverse as biomedical risk models, drug discovery and materials science. Although deep learning has revolutionized learning from raw data and led to numerous high-profile success stories^{3–5}, gradient-boosted decision trees^{6–9} have dominated tabular data for the past 20 years. Here we present the Tabular Prior-data Fitted Network (TabPFN), a tabular foundation model that outperforms all previous methods on datasets with up to 10,000 samples by a wide margin, using substantially less training time. **In 2.8 s, TabPFN outperforms an ensemble of the strongest baselines tuned for 4 h in a classification setting.** As a generative transformer-based foundation model, this model also allows fine-tuning, data generation, density estimation and learning reusable embeddings. TabPFN is a learning algorithm that is itself learned across millions of synthetic datasets, demonstrating the power of this approach for algorithm development. By improving modelling abilities across diverse fields, TabPFN has the potential to accelerate scientific discovery and enhance important decision-making in various domains.

Outline today

The goal of this talk is to understand some characteristics of tabular data and some concepts related to the 2025 Nature article at the previous slide.

- ▶ Causality and missingness
- ▶ Human errors, “dirty categories”
- ▶ Training pipelines
- ▶ Bayesian learning
- ▶ Meta-learning, transfer learning, in-context learning (cf. foundation models, LLMs)
- ▶ TabPFN, a foundation model for tabular data

Structural causal model (SCM)



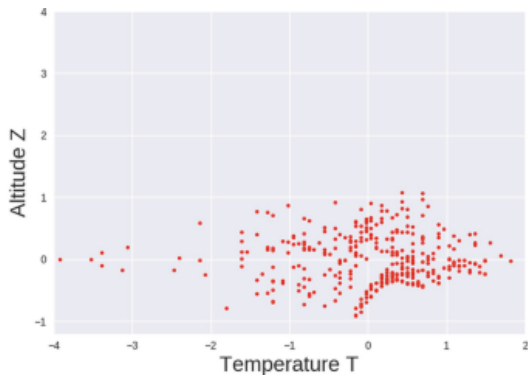
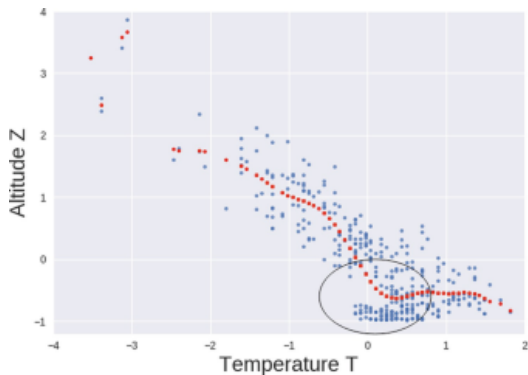
Conditional probability tables e.g. $\Pr(\text{workclass} = w \mid \text{age} = a, \text{education} = e)$

Can be used to generate synthetic datasets, e.g. sometimes ensuring privacy

Jun Zhang et al. (2017). “PrivBayes: Private data release via Bayesian networks”. In: *ACM Transactions on Database Systems (TODS)* 42.4, pp. 1–41

Causality vs. correlation

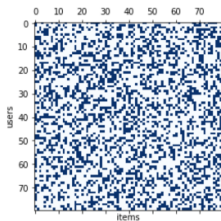
Observed data cannot discriminate between $A \rightarrow B$ or $B \rightarrow A$



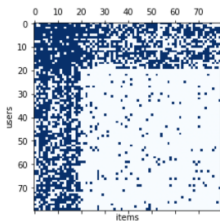
Is it increasing elevation that reduces temperature? Or changing temperature that changes elevation?

Olivier Goudet et al. (2019). "Learning bivariate functional causal models". In: *Cause effect pairs in machine learning*, pp. 101–153. URL: [https://arxiv.org/abs/1906.05232v1](#)

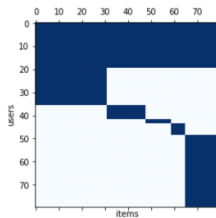
Missing data pattern brings information about dataset



MCAR

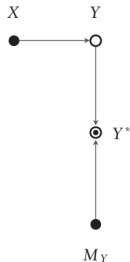


L-MNAR

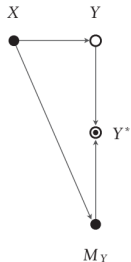


C-MNAR

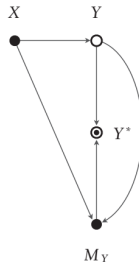
(a) MCAR



(b) MAR



(c) MMAR



Anish Agarwal et al. (2023).
“Causal matrix completion”.
In: *The thirty sixth annual
conference on learning
theory*. PMLR,
pp. 3821–3826

Craig K Enders (2022).
*Applied missing data
analysis*. Guilford
Publications

Missing data theorems

Any impute will do as long as you have enough data (because it does not add information)

More efficient impute will provide better results (smoother curve to learn)

Marine Le Morvan and Gaël Varoquaux (Feb. 2025). “Imputation for prediction: beware of diminishing returns”. In: *ICLR 2025*, in press. URL: <https://hal.science/hal-04662937>

Two schools of human errors

`pip install cleanlab`

“6% of ImageNet test labels are wrong”

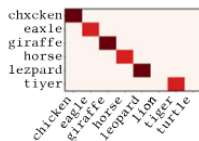
Clean your dataset before you train your algorithm

<https://github.com/cleanlab/cleanlab>

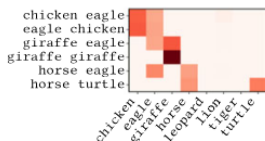
Curtis G. Northcutt, Anish Athalye, and Jonas Mueller (Dec. 2021). “Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks”. In: *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*

`dirty_cat` → `pip install skrub`

Learn with mistakes (“dirty categories”)



(b) Simulated typos



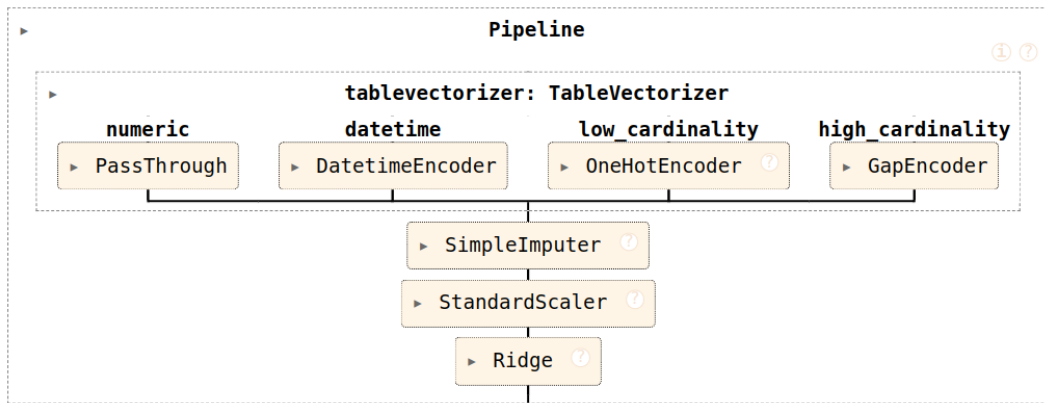
(c) Simulated multi-label categories

<https://skrub-data.org>

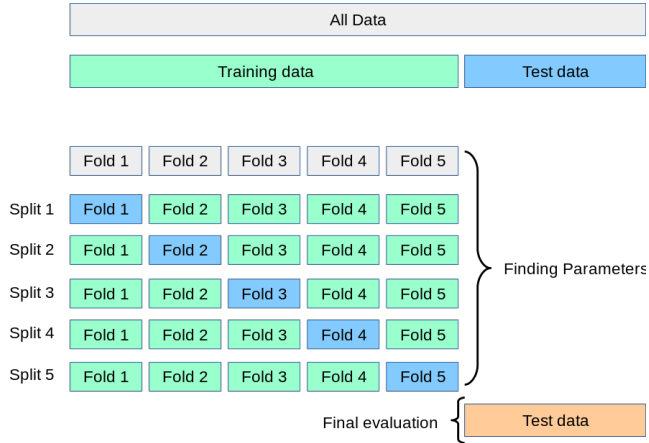
Patricio Cerda and Gaël Varoquaux (2020). “Encoding high-cardinality string categorical variables”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.3, pp. 1164–1176

Scikit-learn's pipelines (some of them directly generated by skrub)

```
model = Pipeline([
    ('onehot', OneHotEncoder(handle_unknown='ignore')),
    ('lr', LogisticRegression(solver='liblinear', C=1e-1, max_iter=300))
])
```



Hyperparameter search using cross validation



Source: https://scikit-learn.org/stable/modules/cross_validation.html

Scikit-learn's cross validation

Cross validation of a model defined by a pipeline

```
cv_results = cross_validate(  
    model, X, y,  
    scoring=['accuracy_score', 'roc_auc_score'], # Use all scores  
    return_train_score=True, n_jobs=-1, # Use all cores (parallel)  
    cv=5, verbose=10  
)
```

Check the docs: https://scikit-learn.org/stable/modules/cross_validation.html#the-cross-validate-function-and-multiple-metric-evaluation

Decision trees

Decision tree trained on all the iris features



Ensemble models

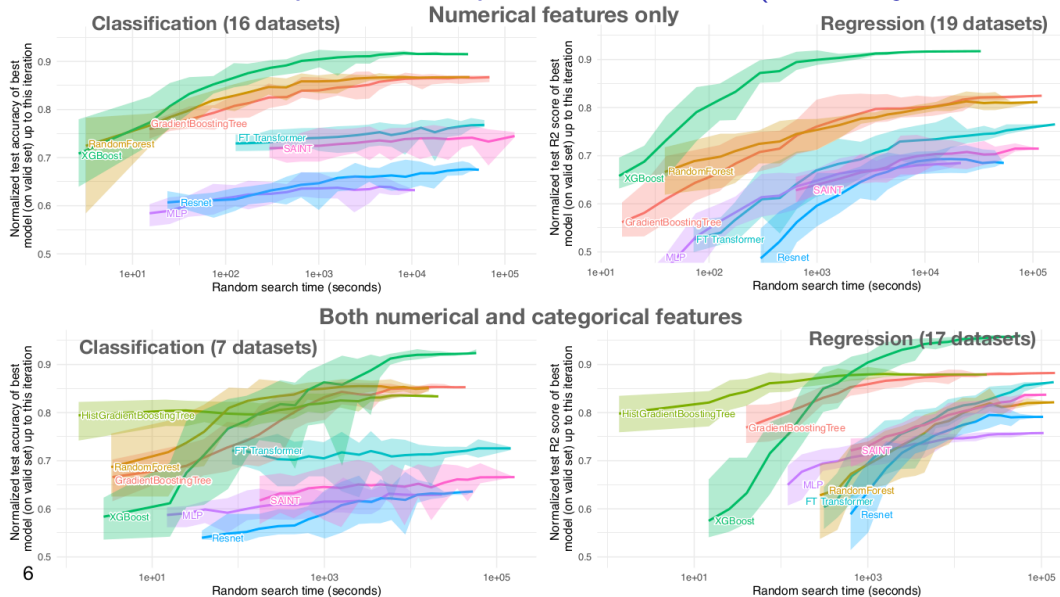
Bagging: reduce variance by averaging; mixture of experts

e.g. random forests

Boosting: reduce bias by increasing the complexity of model and fitting the error

e.g. gradient boosting decision trees (CatBoost, XGBoost)

Tree-based models outperform deep neural networks (Grinsztajn et al. 2022)



Findings of tree-based vs. DNNs paper

Tree-based models remain SOTA on medium-sized data (10k samples)

Trees are:

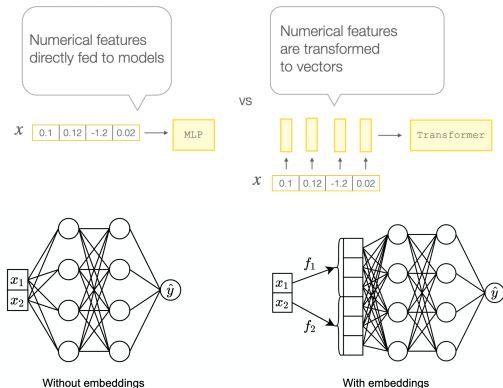
- ▶ robust to uninformative features
- ▶ can learn irregular functions (because nonparametric?)

DNNs are:

- ▶ rotational invariant (but tabular data is not)
- ▶ biased to overly smooth solutions

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux (2022). “Why do tree-based models still outperform deep learning on typical tabular data?” In: *Advances in neural information processing systems*. Vol. 35, pp. 507–520

So-called “numerical embeddings” compensate difference trees/DL



Backbone	Embedding	Average rank (std. dev.)
XGBoost	–	4.6 (2.7)
CatBoost	–	3.6 (2.9)
MLP	–	8.5 (2.6)
Transformer	Linear	5.9 (2.2)
MLP	PLE + Linear + ReLU	5.1 (1.7)
MLP	Periodic + Linear + ReLU	3.0 (2.4)
Transformer	PLE + Linear + ReLU	3.7 (2.2)
Transformer	Periodic + Linear + ReLU	3.9 (2.5)

Yury Gorishniy, Ivan Rubachev, and Artem Babenko (2022). “On embeddings for numerical features in tabular deep learning”. In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 24991–25004

Foundation models

“One pre-trained model to rule them all”

Generalize pretraining to new tasks

Tom Brown et al. (2020). “Language models are few-shot learners”. In: *Advances in neural information processing systems*. Vol. 33, pp. 1877–1901

Long Ouyang et al. (2022). “Training language models to follow instructions with human feedback”. In: *Advances in neural information processing systems* 35, pp. 27730–27744

Example: Large Language Models (LLM)

1. Generative pretraining using a transformer to predict the next word

Transformers / are / a / new / machine / [learning]

Transformers / are / a / new / machine / learning / [architecture]

2. Supervised finetuning: expert demonstration data

Query: put the first letters in uppercase in "optimizing human learning"

Answer: Optimizing Human Learning

3. Reinforcement learning using human feedback: expert comparison data

Query: write a poem

Answer 1: Roses are red

Answer 2: Once upon a time, a prince in a castle

Where Answer 2 is favorably voted by experts

ChatGPT is therefore trained to optimize user preference and generalize to new tasks

In-context learning

Normal ML: `fit(train)`, `predict(test)`

Transfer learning: `pretrain`, `finetune(train)`, `predict(test)` (e.g. word embeddings)

In-context learning: `pretrain (LLM)`, `predict(test, train)`

Bayesian learning

Instead of learning parameters θ

Update a distribution over parameters $p(\theta \mid D)$

For example, $\theta_1, \dots, \theta_S \sim p(\theta \mid D)$

$$\text{Then } \mathbb{E}_{\theta} f_{\theta}(x) = \int_{\theta} f_{\theta}(x) d\theta \simeq \frac{1}{S} \sum_{i=1}^S f_{\theta}(x)$$

“Advantage”: reduce the number of hyperparameters (if we have a good prior)

Can you name any Bayesian models?

Are LLMs Bayesian?

Do you know Bayesian deep learning models?

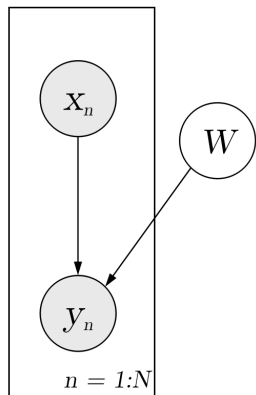
Answer from students

“Naive Bayes”

“No”

“Variational autoencoders”

Bayesian neural networks (BNNs)



$$W_i \sim \mathcal{N}(W_i | 0, I), \quad i = 1 \cdots L.$$
$$y_{mean} = f_{NN}(x, \{W_i\}_{i=1}^L)$$
$$y \sim \mathcal{N}(y | y_{mean}, \sigma^2)$$

Source: zhusuan.readthedocs.io

Snapshot ensembles: keeping last T trained weights simulates a distribution over weights, which can give a distribution over outputs, and regularizes the model (see also “stochastic weight averaging”)

TabPFN is pretrained on 1 million synthetic datasets

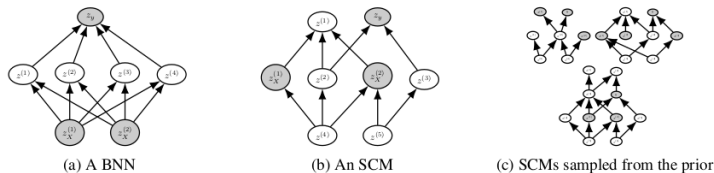


Figure 2: Overview of graphs generating data in our prior. Inputs x are mapped to the output y through unobserved nodes z . Plots based on Müller et al. (2022).

- ▶ Pretrain a foundation model to “train and test” (very meta)
- ▶ Directly estimate $p(y_{test} \mid x_{test}, D_{train})$ using in-context learning
- ▶ It becomes a strong baseline, see also results in the Nature paper
- ▶ Limitations: up to 10000 samples, TabPFN v1 (2023) does not consider missing data, TabPFN v2 (2025) version considers only MCAR data and is not open source.

Noah Hollmann, Samuel Müller, Katharina Eggersperger, et al. (2023). “TabPFN: A transformer that solves small tabular classification problems in a second”. In: *ICLR 2023*; Noah Hollmann, Samuel Müller, Lennart Purucker, et al. (2025). “Accurate predictions on small data with a tabular foundation model”. In: *Nature* 637.8045, pp. 319–326

Questions from students

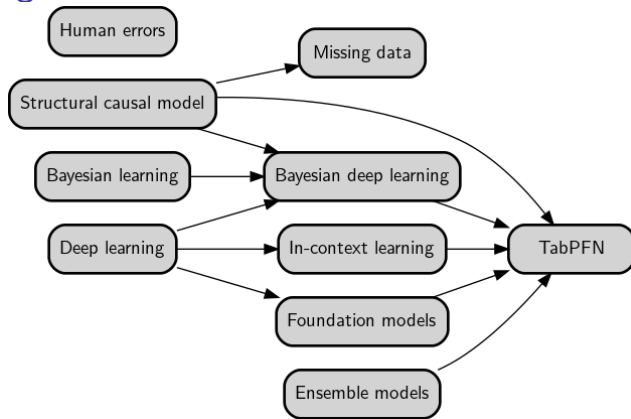
Q: The TabPFN paper says “up to 10000 samples”, how to make it scalable?

A: Use numerical embeddings, see other methods like TabM. This is the approach proposed by Yandex Research's Gorishniy, Rubachev, and Babenko [2022](#), see their [code](#) and [blog post](#).

Q: Can you please give more details about the TabPFN architecture?








A: Sorry, I didn't look at [TabPFN code](#) yet (the 2023 paper has no detail but code, the 2025 has details but not all code). To have an intuition, check the [causal matrix completion paper](#) and slides. Attention mechanism is a bit like “finding in the training set the nearest neighbors of the test data” (in the paper they also make an analogy with Gaussian processes).






Take home message





TabPFN gets inspiration from transformers/LLMs but also from old techniques like structural causal models (SCMs), while tabular data is encoded as tokens; promising directions for interpretability (see [Explaining TabPFN](#) by shapiq)

Please use TableReport from `pip install skrub`

-  Agarwal, Anish et al. (2023). “Causal matrix completion”. In: *The thirty sixth annual conference on learning theory*. PMLR, pp. 3821–3826.
-  Brown, Tom et al. (2020). “Language models are few-shot learners”. In: *Advances in neural information processing systems*. Vol. 33, pp. 1877–1901.
-  Cerda, Patricio and Gaël Varoquaux (2020). “Encoding high-cardinality string categorical variables”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.3, pp. 1164–1176.
-  Enders, Craig K (2022). *Applied missing data analysis*. Guilford Publications.
-  Gorishniy, Yury, Ivan Rubachev, and Artem Babenko (2022). “On embeddings for numerical features in tabular deep learning”. In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 24991–25004.
-  Goudet, Olivier et al. (2019). “Learning bivariate functional causal models”. In: *Cause effect pairs in machine learning*, pp. 101–153. URL: <https://inria.hal.science/hal-02433201/document>.
-  Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux (2022). “Why do tree-based models still outperform deep learning on typical tabular data?” In: *Advances in neural information processing systems*. Vol. 35, pp. 507–520.

-  Hollmann, Noah, Samuel Müller, Katharina Eggensperger, et al. (2023). “TabPFN: A transformer that solves small tabular classification problems in a second”. In: *ICLR 2023*.
-  Hollmann, Noah, Samuel Müller, Lennart Purucker, et al. (2025). “Accurate predictions on small data with a tabular foundation model”. In: *Nature* 637.8045, pp. 319–326.
-  Le Morvan, Marine and Gaël Varoquaux (Feb. 2025). “Imputation for prediction: beware of diminishing returns”. In: *ICLR 2025*, in press. URL: <https://hal.science/hal-04662937>.
-  Northcutt, Curtis G., Anish Athalye, and Jonas Mueller (Dec. 2021). “Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks”. In: *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*.
-  Ouyang, Long et al. (2022). “Training language models to follow instructions with human feedback”. In: *Advances in neural information processing systems* 35, pp. 27730–27744.

-  Yin, Kejing et al. (2019). “Learning phenotypes and dynamic patient representations via RNN regularized collective non-negative tensor factorization”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 1246–1253.
-  Zhang, Jun et al. (2017). “PrivBayes: Private data release via Bayesian networks”. In: *ACM Transactions on Database Systems (TODS)* 42.4, pp. 1–41.