EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

machine learning
new perspectives
for science

AutoML
Streamlining Machine Learning

Katharina Eggensperger

katharina.eggensperger@uni-tuebingen.de
AutoML for Science

May 14th, 2024

AutoML.org

Why we need (Auto-)ML

ML is Everywhere. And it is successful!




Why? → Flexible and powerful algorithmic components

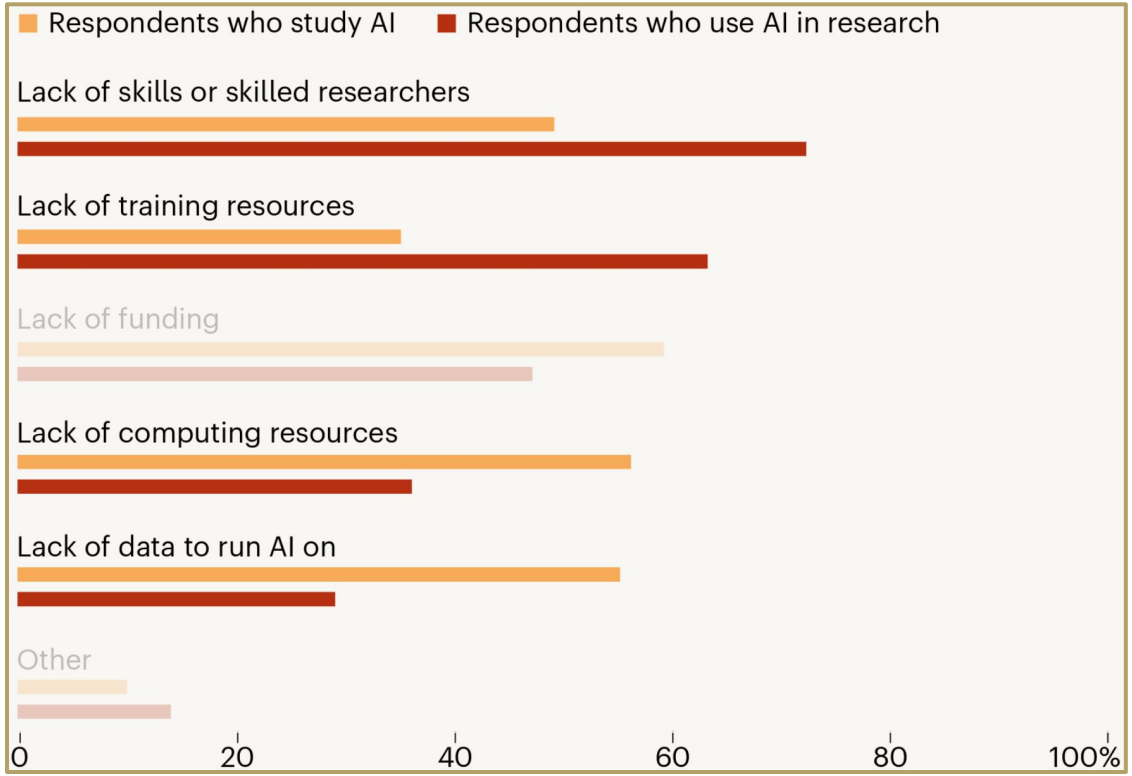
This also means

→ Developing intelligent systems can be complex

→ ML is not yet easy-to-use for everyone

* Images created by OpenAI's DALL·E

Barriers to machine learning



Source: Van Noorden et al. “AI and Science: What 1,600 researchers think” Nature 2023

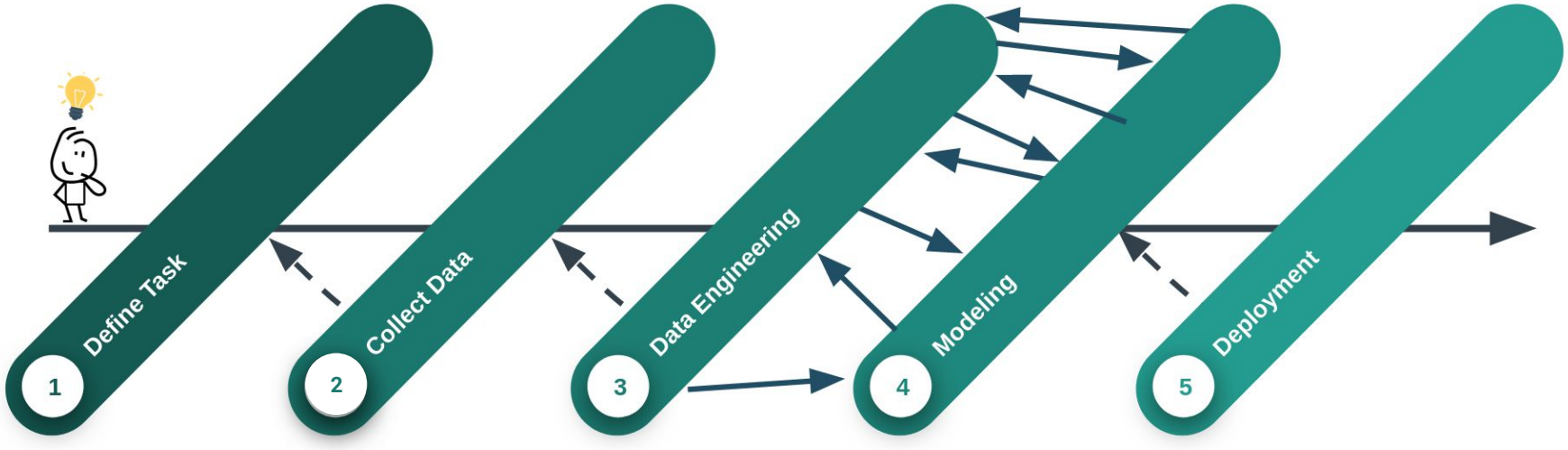
… also true for industry

Challenges for Applying ML

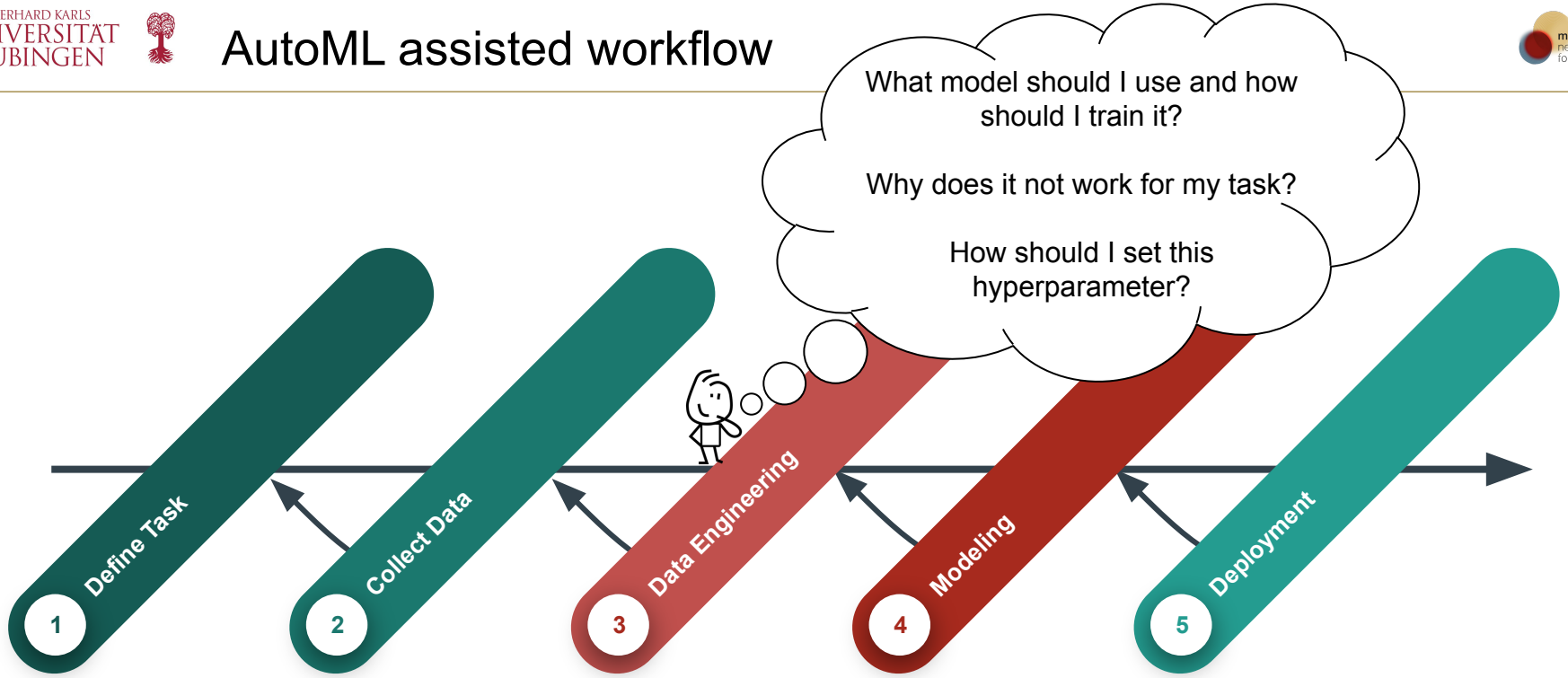
- Required Expertise
- Required Data
- Required Resources

→ Targets of AutoML!

Machine learning workflow



AutoML assisted workflow



Advantages of AutoML

🚀 **Efficient** research and development

→ AutoML can yield state-of-the-art results

🧮 **Systematic** research and development

→ no (human) bias or non-systematic evaluation

🌍 **Broader use** of ML methods

→ less required ML expert knowledge

Storyline for Today

I. AutoML Research Areas

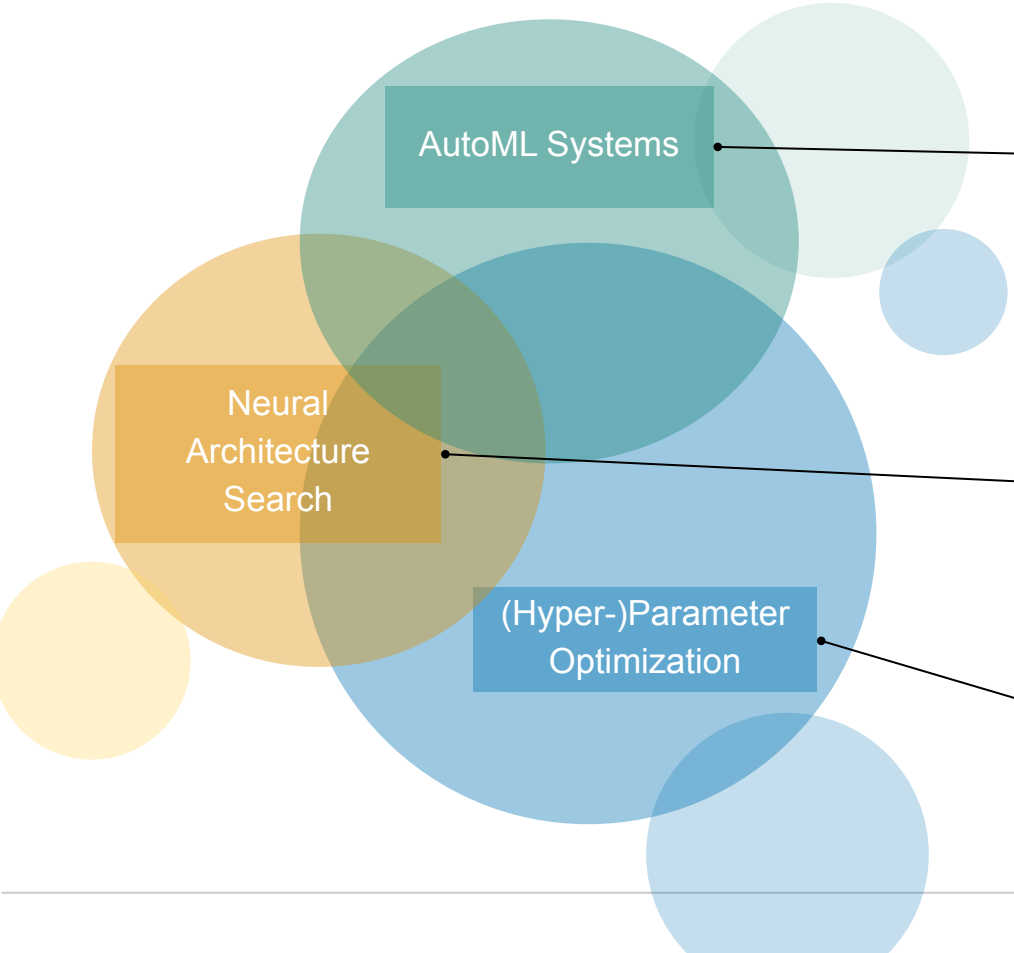
II. AutoML for (tabular) Data

Auto-sklearn: AutoML in Python

TabPFN: In-Context Learning

Modular AutoML systems

AutoML research areas



Find the best performing **algorithm** and **configuration** given a searchspace and costfunction:

$$(\mathcal{A}^*, \boldsymbol{\lambda}^*) \in \underset{\mathcal{A} \in \mathbf{A}, \boldsymbol{\lambda} \in \boldsymbol{\Lambda}}{\arg\min}\, c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$$

Find the best performing **neural architecture** given a searchspace and costfunction:

$$\boldsymbol{\lambda}^* \in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}}{\arg\min}\, c(N_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$$

Find the best performing **configuration** given a searchspace and costfunction:

$$\boldsymbol{\lambda}^* \in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}}{\arg\min}\, c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$$

AutoML Research: Hyperparameter Optimization

(Hyper-)Parameter Optimization

Methods:

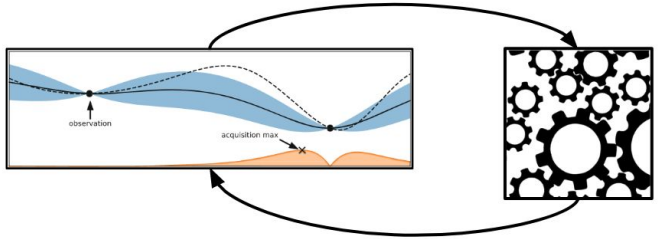
- Bayesian Optimization
- Evolutionary Algorithms

Current challenges:

- learn across tasks
- scale to expensive models
- include expert knowledge

Success Stories:

- Tuning Alpha Go
- W&B, Optuna and more



AutoML Research: Neural Architecture Search

Neural Architecture Search

Methods:

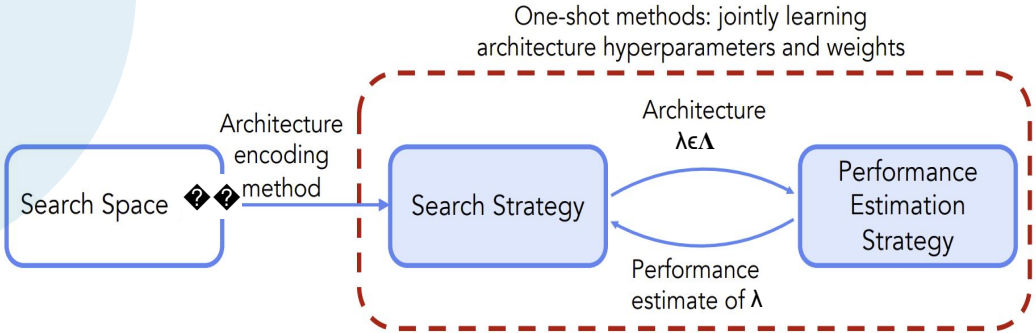
- Black-Box NAS
- One-Shot NAS

Current challenges:

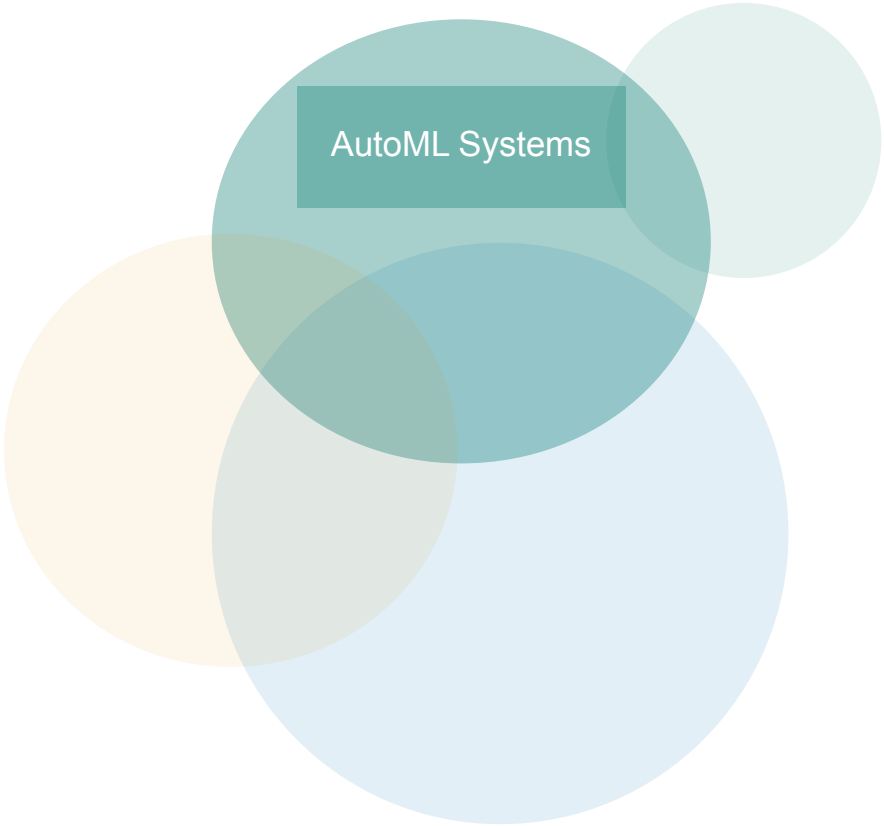
- speed up via zero-cost proxies
- search space design

Success Story: Hardware-aware NAS for embedded devices

Note: NAS is a decade-old problem, but mainstream since 2017; probably the most popular AutoML Problem



AutoML Research: AutoML Systems



Methods:

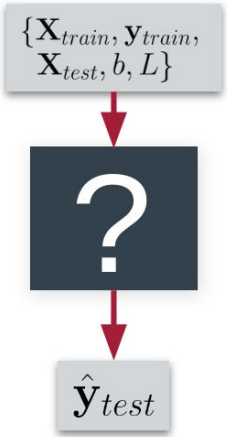
- HPO and NAS
- Ensembling

Current challenges:

- learn from experience
- incorporate human expertise
- multiple data modalities

Success Stories:

- hundreds of (academic) applications
- many open-source tools



General Challenges for AutoML

But automating the ML workflow it is not that easy, because

🔁 Each dataset potentially requires **different optimal ML-designs**

⌛ Training of a single ML model can be **quite expensive**

❓Mathematical **relation** between design and performance is (often) **unknown**

🎛️🎚️ Optimization in **highly complex spaces**

AutoML for (tabular) Data

>> AutoML systems and how they work.

Tabular data is omnipresent

- Simple and easy-accessible
- Available in many domains
- Challenging for ML
- Diverse ML Landscape

```
1,14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065
1,13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050
1,13.16,2.36,2.67,18.6,101,2.8,3.24,.3,2.81,5.68,1.03,3.17,1185
1,14.37,1.95,2.5,16.8,113,3.85,3.49,.24,2.18,7.8,.86,3.45,1480
1,13.24,2.59,2.87,21,118,2.8,2.69,.39,1.82,4.32,1.04,2.93,735
1,14.2,1.76,2.45,15.2,112,3.27,3.39,.34,1.97,6.75,1.05,2.85,1450
1,14.39,1.87,2.45,14.6,96,2.5,2.52,.3,1.98,5.25,1.02,3.58,1290
1,14.06,2.15,2.61,17.6,121,2.6,2.51,.31,1.25,5.05,1.06,3.58,1295
1,14.83,1.64,2.17,14,97,2.8,2.98,.29,1.98,5.2,1.08,2.85,1045
1,13.86,1.35,2.27,16,98,2.98,3.15,.22,1.85,7.22,1.01,3.55,1045
1,14.1,2.16,2.3,18,105,2.95,3.32,.22,2.38,5.75,1.25,3.17,1510
1,14.12,1.48,2.32,16.8,95,2.2,2.43,.26,1.57,5,1.17,2.82,1280
1,13.75,1.73,2.41,16,89,2.6,2.76,.29,1.81,5.6,1.15,2.9,1320
1,14.75,1.73,2.39,11.4,91,3.1,3.69,.43,2.81,5.4,1.25,2.73,1150
1,14.38,1.87,2.38,12,102,3.3,3.64,.29,2.96,7.5,1.2,3,1547
1,13.63,1.81,2.7,17.2,112,2.85,2.91,.3,1.46,7.3,1.28,2.88,1310
1,14.3,1.92,2.72,20,120,2.8,3.14,.33,1.97,6.2,1.07,2.65,1280
1,13.83,1.57,2.62,20,115,2.95,3.4,.4,1.72,6.6,1.13,2.57,1130
1,14.19,1.59,2.48,16.5,108,3.3,3.93,.32,1.86,8.7,1.23,2.82,1680
1,13.64,3.1,2.56,15.2,116,2.7,3.03,.17,1.66,5.1,.96,3.36,845
```

→ **High Demand for AutoML**

- Efficient HPO methods
- AutoML Systems for non-experts

ML pipeline design

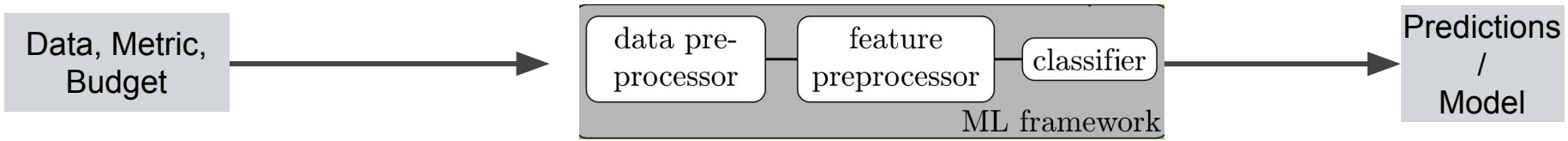
ML pipeline design

Data, Metric, Budget

```
import automl
>>> cls = automl.Classifier()
>>> cls.fit(X_train, y_train)
>>> y_test = cls.predict(X_test)
```

Predictions / Model

Design Space: Traditional ML with scikit learn



Design Space: Traditional ML with scikit learn

Data, Metric, Budget →

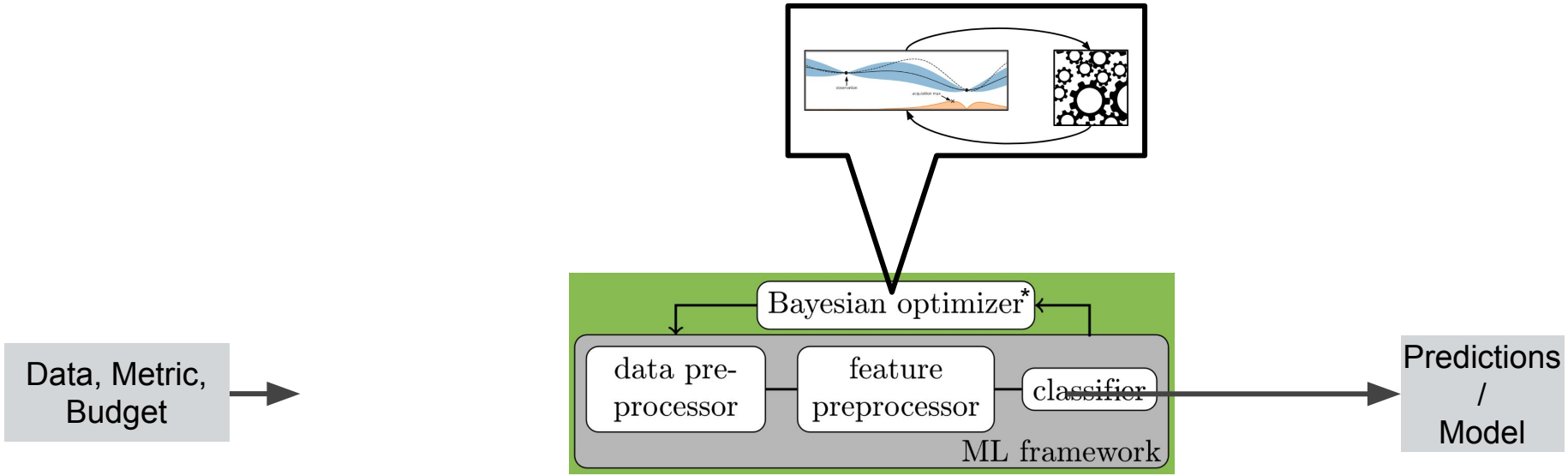
preprocessor	#λ
extreml. rand. trees prepr.	5
fast ICA	4
feature agglomeration	4
kernel PCA	5
rand. kitchen sinks	2
linear SVM prepr.	3
no preprocessing	-
nystroem sampler	5
PCA	2
polynomial	3
random trees embed.	4
select percentile	2
select rates	3
one-hot encoding	2
imputation	1
balancing	1
rescaling	1

→

classifier	#λ
AdaBoost (AB)	4
Bernoulli naïve Bayes	2
decision tree (DT)	4
extreml. rand. trees	5
Gaussian naïve Bayes	-
gradient boosting (GB)	6
kNN	3
LDA	4
linear SVM	4
kernel SVM	7
multinomial naïve Bayes	2
passive aggressive	3
QDA	2
random forest (RF)	5
Linear Class. (SGD)	10

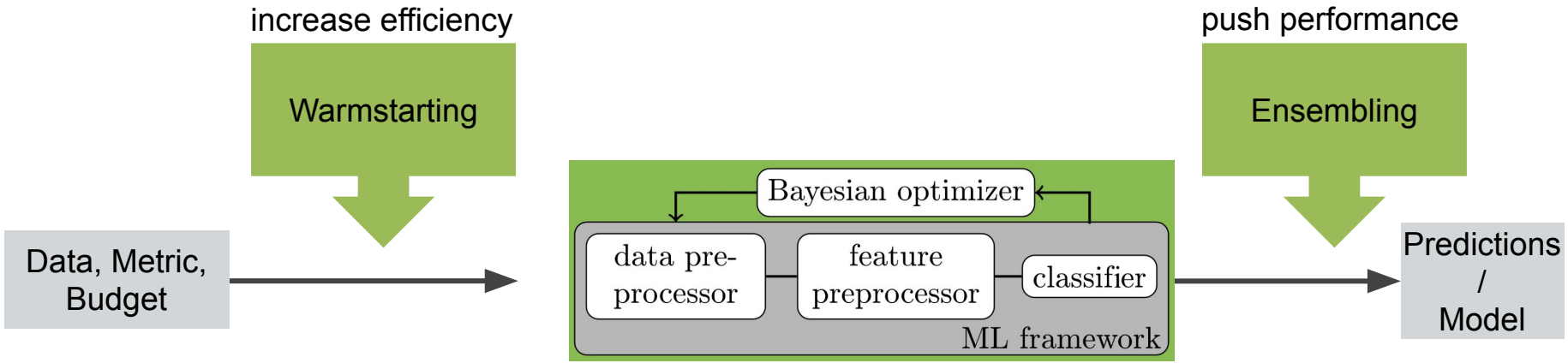
→ Predictions / Model

Simple AutoML system

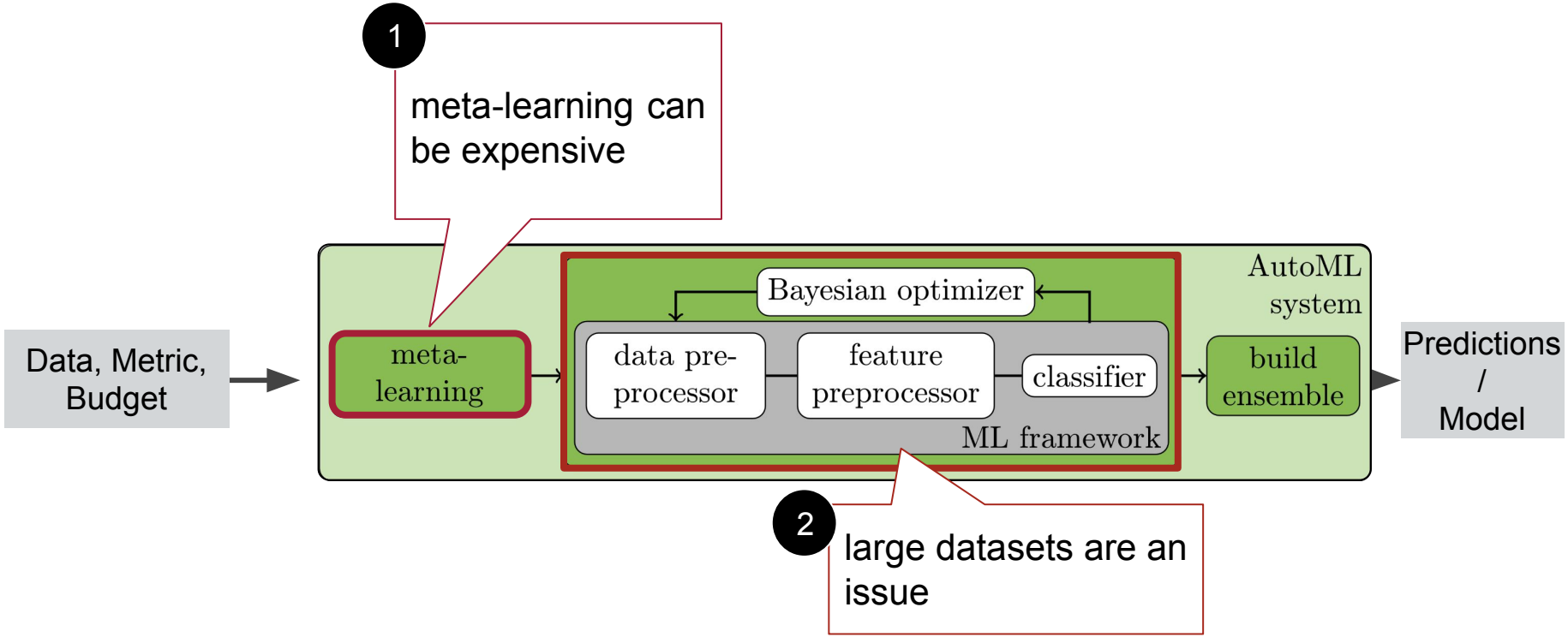


* for a detailed Introduction, seet [Roman Garnett: “Bayesian Optimization”. Cambridge University Press 2023]

Auto-Sklearn 1.0

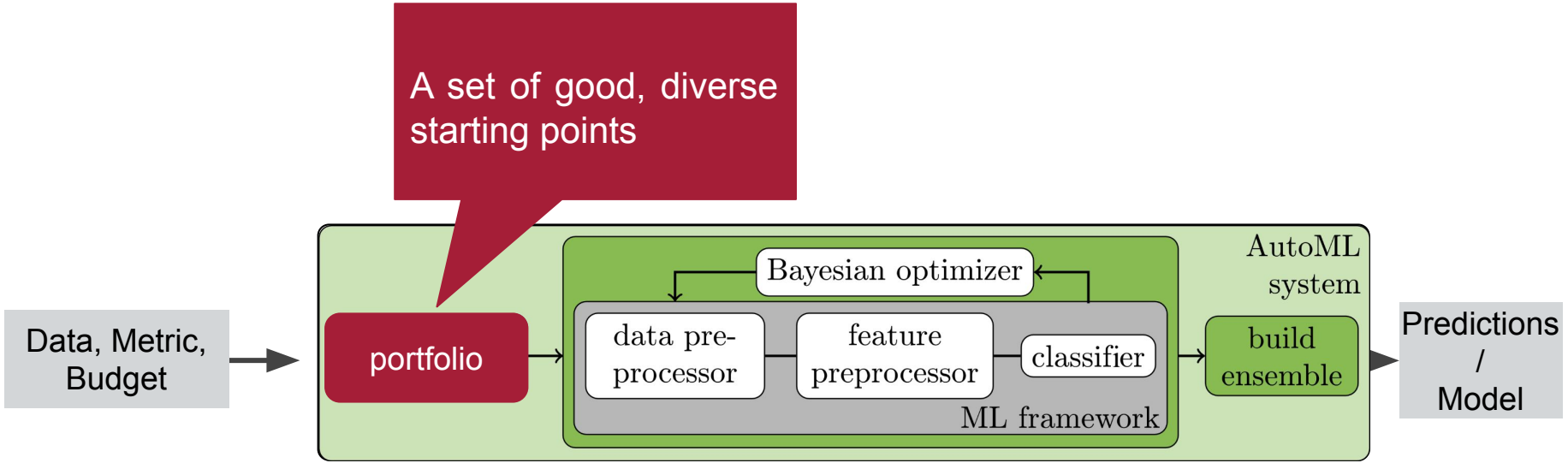


From Auto-sklearn 1.0 to Auto-sklearn 2.0

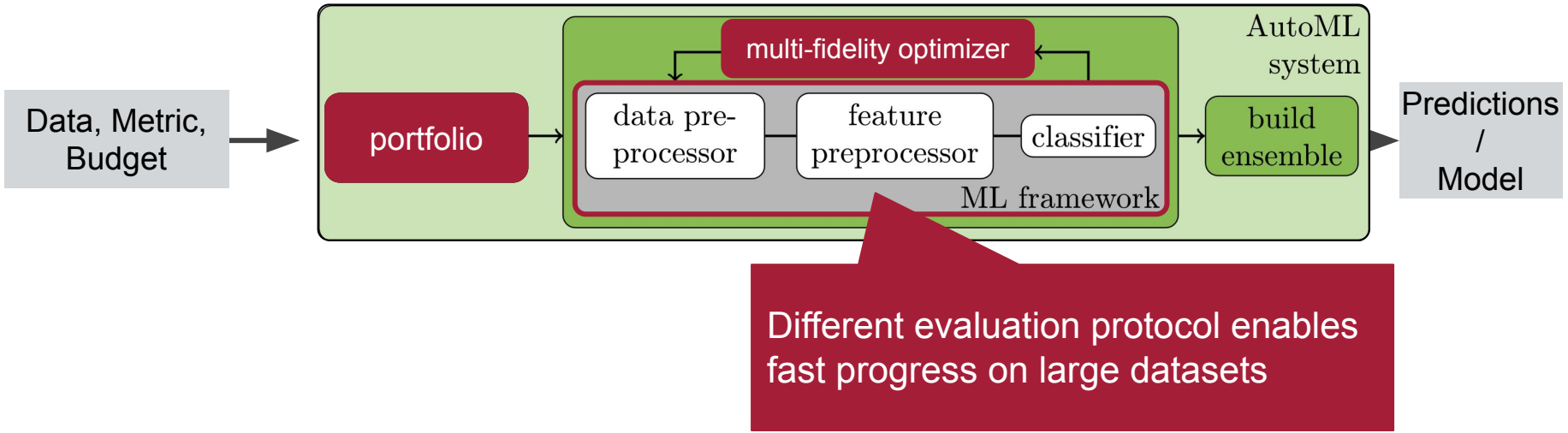


M. Feurer, A. Klein, **K. Eggensperger**, T. Springenberg, M. Blum, F. Hutter: Efficient and Robust Automated Machine Learning. In: NeurIPS 2015
M. Feurer, **K. Eggensperger**, S. Falkner, M. Lindauer, F. Hutter: "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. In: JMLR 2022

Portfolios instead of meta-learning

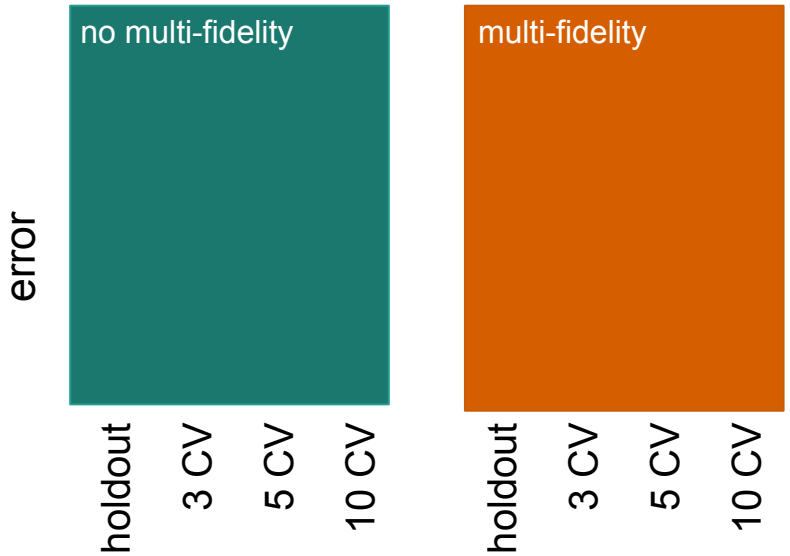


Multi-fidelity optimization for large datasets



Impact of hyper-hyperparameters

performance on a dataset?

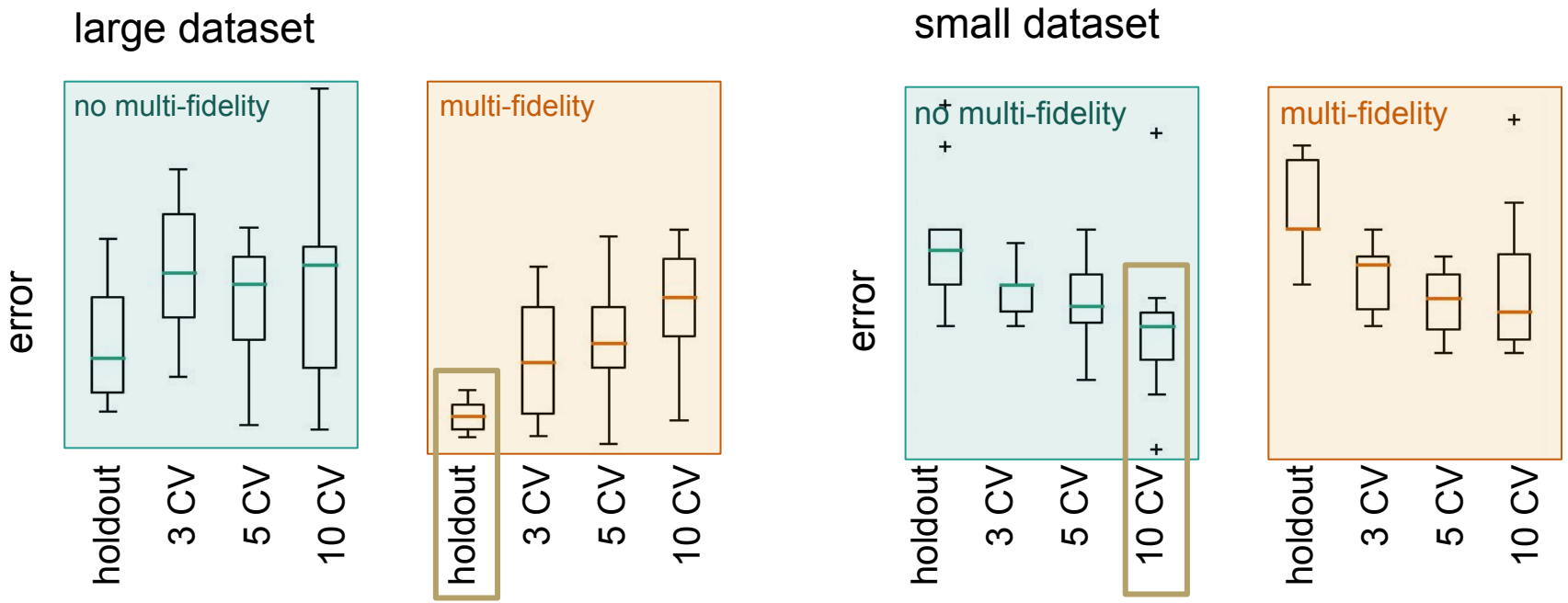


Evaluation via cross-validation or holdout?

and

Model Selection w/ or w/o multi-fidelity?

Impact of hyper-hyperparameters



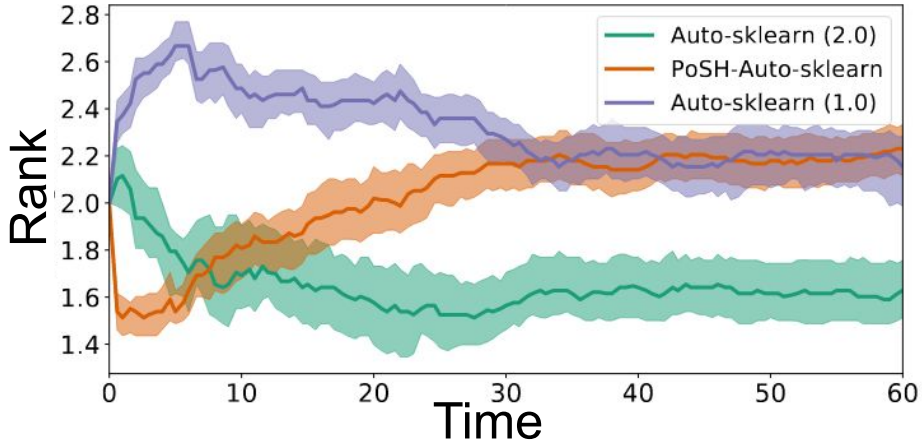
Auto-AutoML in Auto-Sklearn 2.0

AutoML systems are not hands-off!

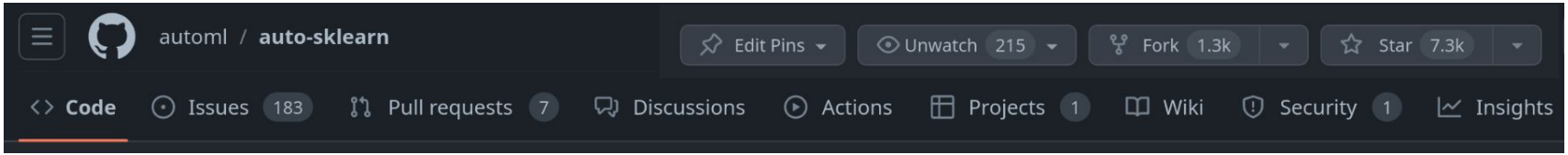
→*Evaluation and model selection protocol* impact performance and best choice depends on dataset

Image created by OpenAI's DALL·E.

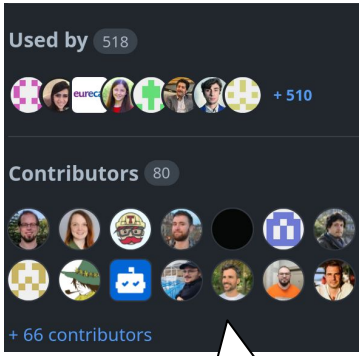
Our solution: Use ML to learn a selector



Open-source AutoML




```
import autosklearn.classification
>>> cls = autosklearn.classification.AutoSklearnClassifier()
>>> cls.fit(X_train, y_train)
>>> predictions = cls.predict(X_test)
```



large team effort!

- used for many **applications** (>**2.5K** citations **and >15K** downloads/month)
- **won** 2 AutoML **competitions**

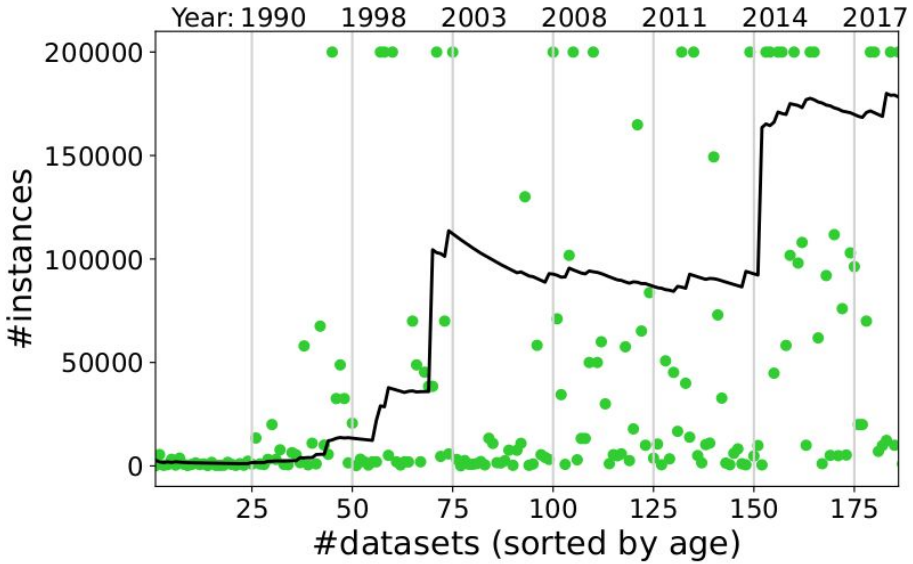
/automl/auto-sklearn
/automl/auto-sklearn-talks

M. Feurer, A. Klein, **K. Eggensperger**, T. Springenberg, M. Blum, F. Hutter: Efficient and Robust Automated Machine Learning. In: NeurIPS 2015
M. Feurer, **K. Eggensperger**, S. Falkner, M. Lindauer, F. Hutter: "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. In: JMLR 2022

AutoML for small (tabular) Data

>> What about deep learning?

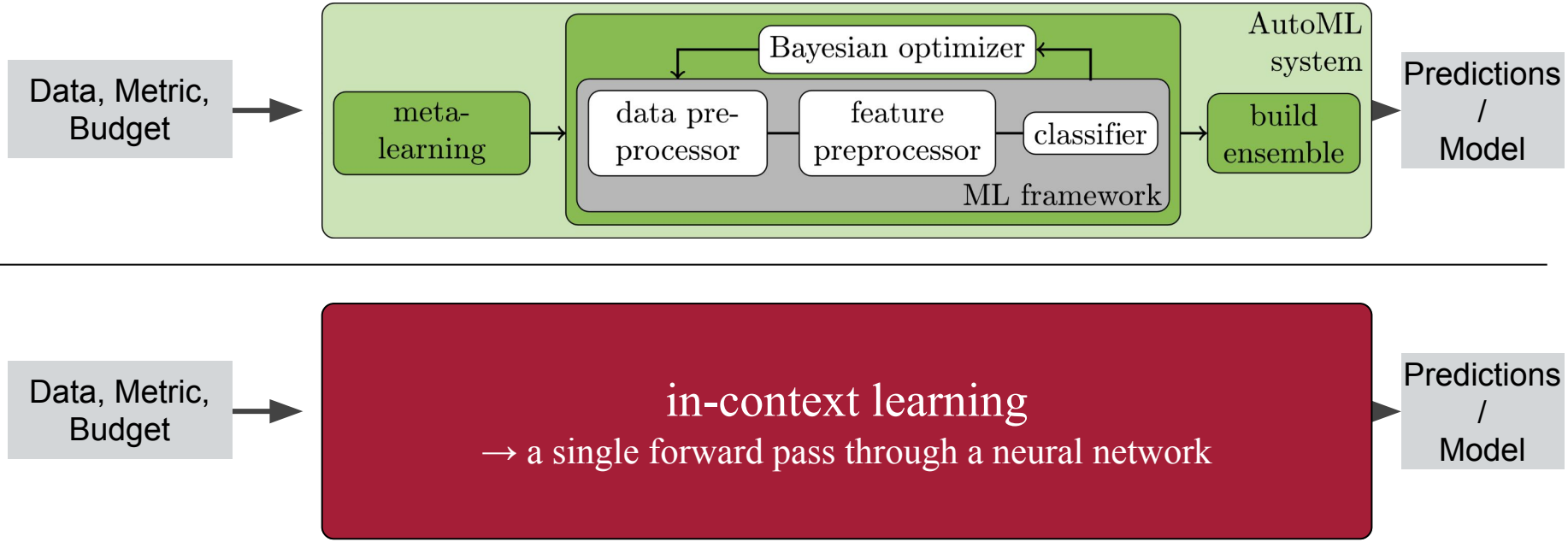
The role of DL for tabular data?



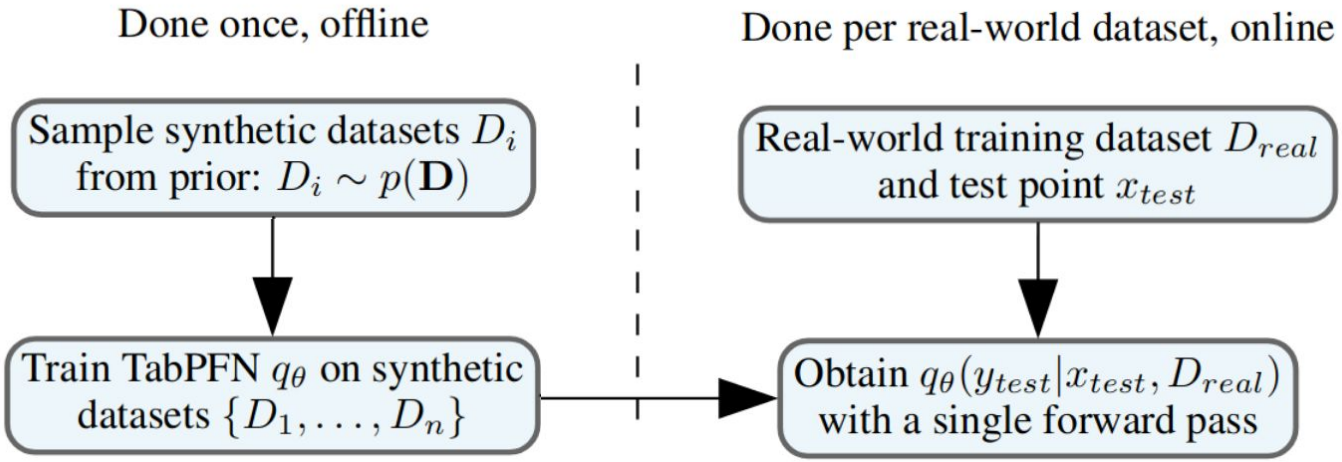
What about these?
Can we get faster predictions?
Without overfitting?

* based on public tabular datasets used to benchmark tabular models

AutoML in less than one second?



Prior-data fitted networks



Prior-data fitted networks using GPs as their prior

E.g. Gaussian Process

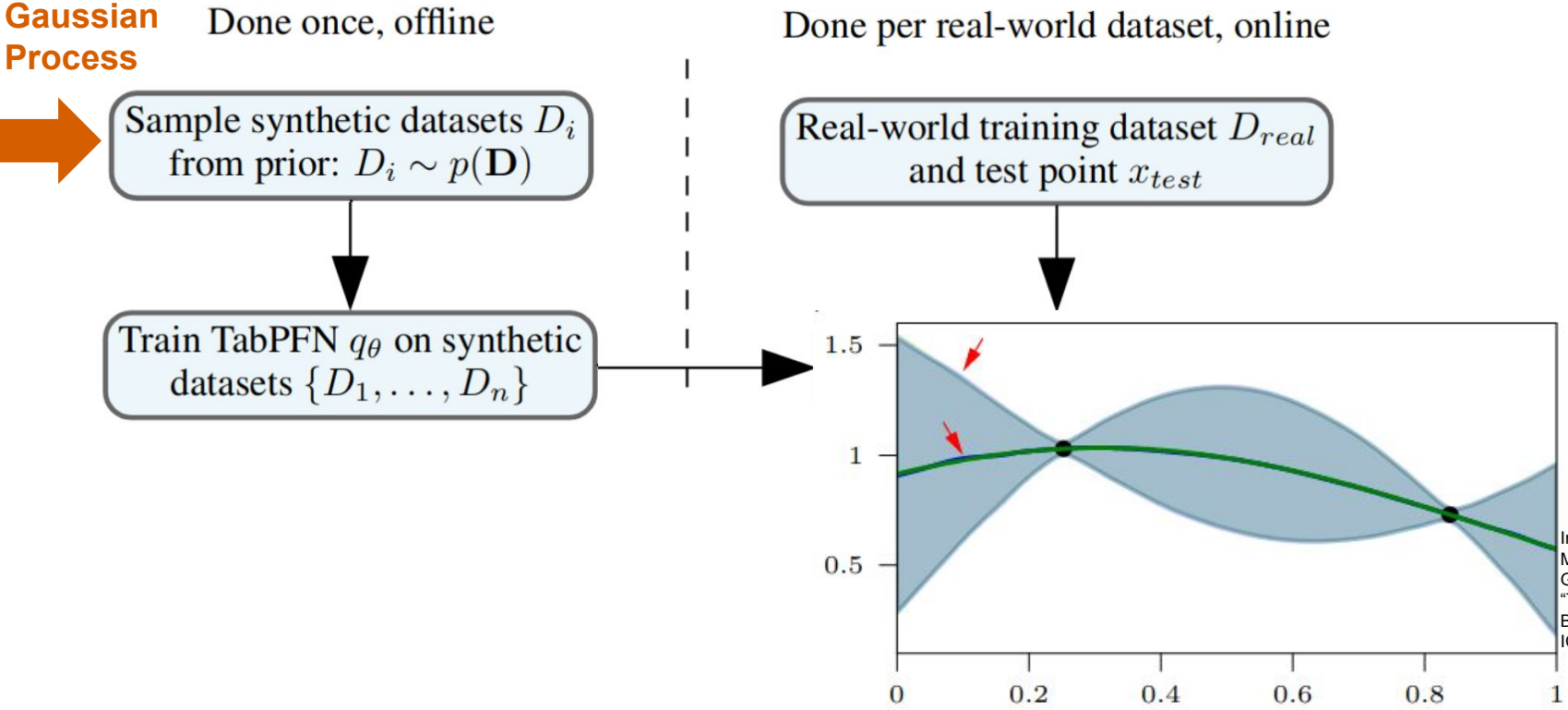
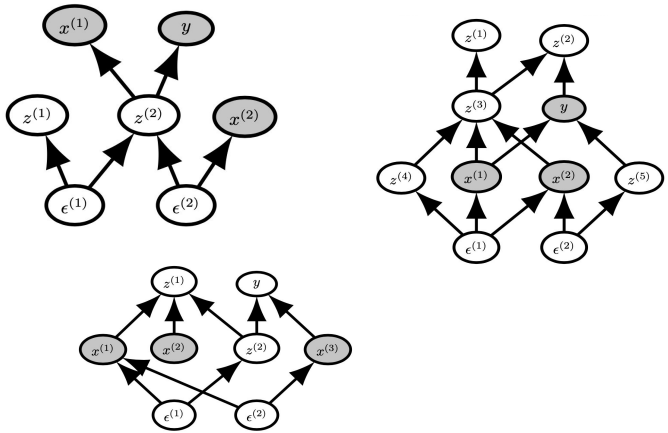
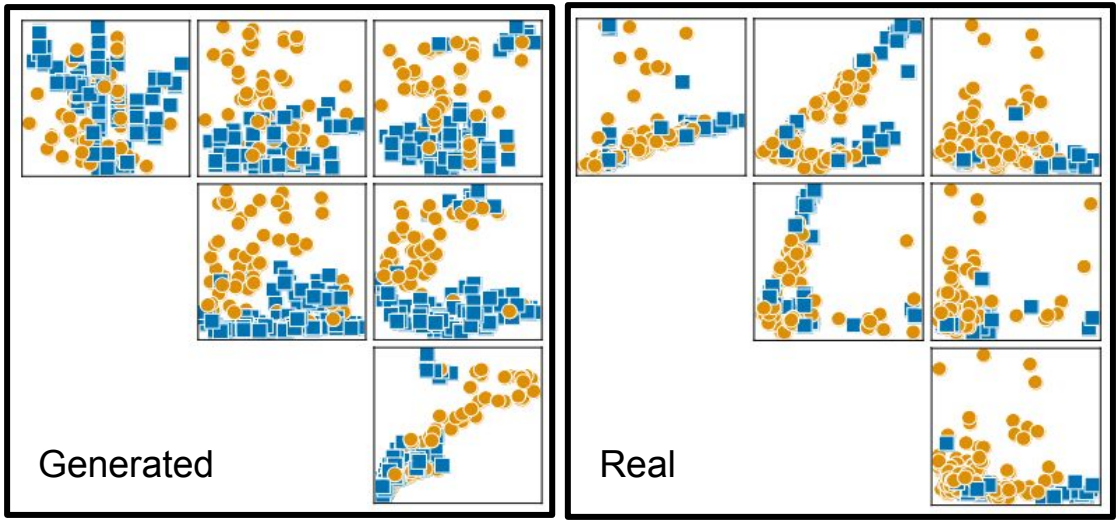


Image: N. Hollmann, S. Müller, S. Pineda, J. Grabocka and F. Hutter: "Transformers can do Bayesian Inference". In ICLR'22

TabPFN: Synthetic training data

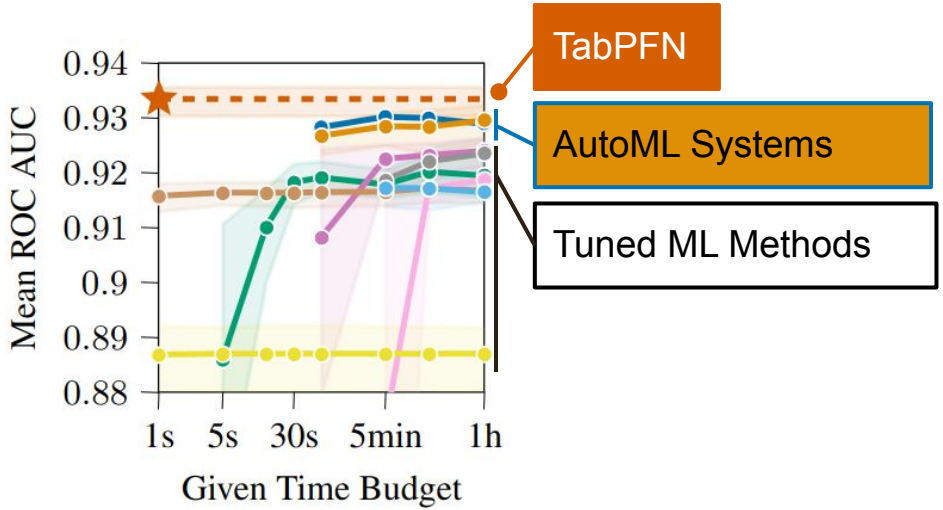
Sample synthetic datasets D_i from prior: $D_i \sim p(\mathbf{D})$

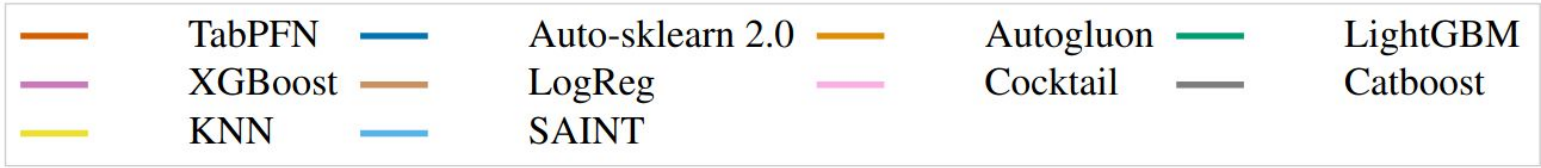




TabPFN: Results

18 small datasets (<1000 samples), continuous features, no missing values





TabPFN: Summary

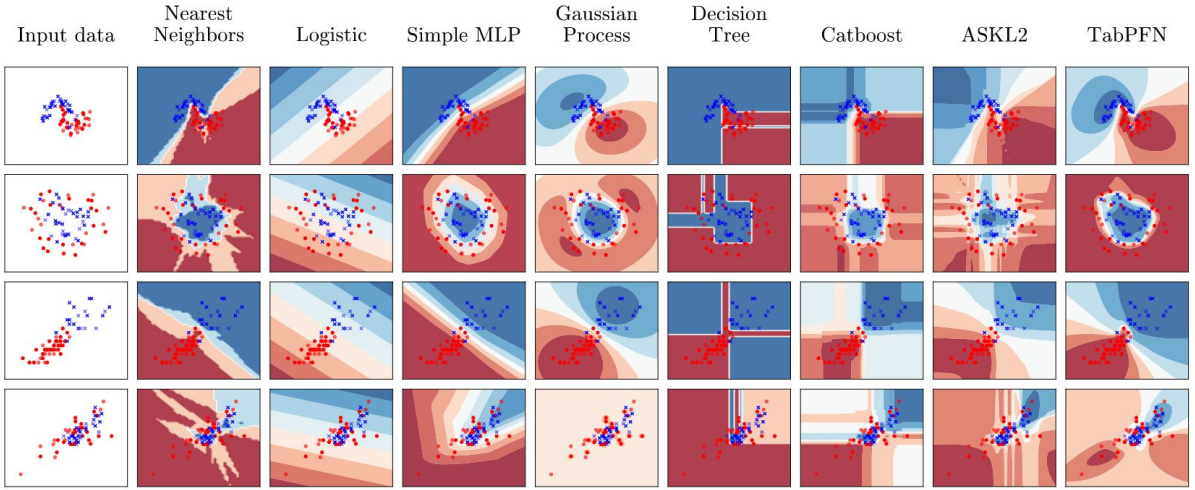
TL;DR TabPFN, a trained transformer, that instantly yields predictions for tabular datasets.

Limitations and Remarks

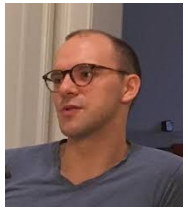
- Up to 1000 samples
- Up to 100 features
- Up to 10 classes

→ works best on **continuous** datasets **without missing** values

/automl/TabPFN







Frank Hutter
Professor, Uni Freiburg

Noah Hollmann
RE, Uni Freiburg

Samuel Müller
PhD student, Uni Freiburg

What's next?

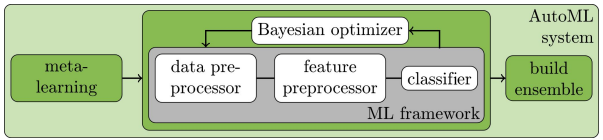
>> Future directions of research.

Modular and flexible AutoML systems

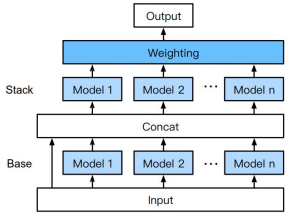
Many high-performance AutoML systems exist. However,

- **Research on AutoML systems is hard**
  AutoML systems are complex, using different search spaces and optimization methods
  →We compare implementations and not methods

Auto-Sklearn
→search the best pipeline and use ensembling



AutoGluon
→stack default pipelines

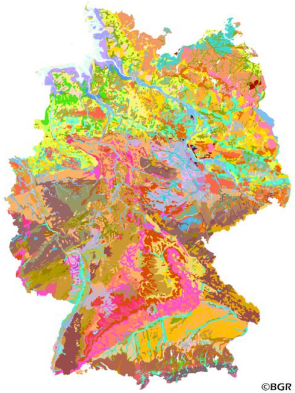


M. Feurer, A. Klein, **K. Eggensperger**, T. Springenberg, M. Blum, F. Hutter: Efficient and Robust Automated Machine Learning. In: NeurIPS 2015
N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, A. Smola: AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. In arXiv 2020

Modular and flexible AutoML systems

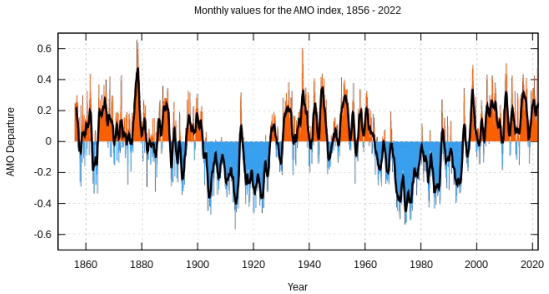
Many high-performance AutoML systems exist. However,

- **Research on AutoML systems is hard**
- **Real-world applications are different**
  Evaluation protocols, data types, modelling approaches, metrics
  → No single solution



Datasource: BUEK1000 V2.1, (C) BGR, Hannover, 2013.

Geodata does not allow a standard train/valid/test or cross-validation split



Datasource: Rosentod, Marsupilami, Amo timeseries 1856-present, als gemeinfrei gekennzeichnet, Details auf Wikimedia Commons

Time series require a customized evaluation protocol and models

Modular and flexible AutoML systems

Many high-performance AutoML systems exist. However,

- **Research on AutoML systems is hard**
- **Real-world applications are different**

AutoML systems should be **easy to adapt** and **extend to any ML workflow**

→ Enables thorough comparisons (what makes a method work well)
→ Enables broad applicability (impact on real-world tasks)

ongoing team effort

/automl/amltk (WIP)

Conclusion

>> Just 2 more minutes.

Take Home Message

- AutoML can streamline ML and make research more
  - 🦾 **robust**
  - 🧮 **systematic** (and reproducible)
  - 🚀 **efficient**
  - 🎯 **and improve results**

- **Auto-Sklearn, an AutoML system** for tabular data and searches for the best performing ML pipeline

- **TabPFN, a pre-trained transformer** that instantly yields predictions on small tabular ML tasks

- We need **modular AutoML systems** to facilitate AutoML research and ML development

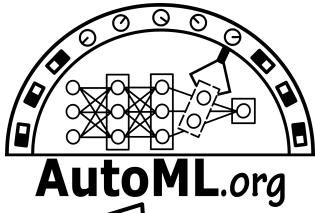
/automl/auto-sklearn

/automl/TabPFN

/automl/amltk

Take Home Message

- AutoML can streamline ML and make research more
  - 🦾 **robust**
  - 🧮 **systematic** (and reproducible)
  - 🚀 **efficient**
  - 🎯 **and improve results**
- **Auto-Sklearn, an AutoML system** for tabular data and searches for the best performing ML pipeline
- **TabPFN, a pre-trained transformer** that instantly yields predictions on small tabular ML tasks
- We need **modular AutoML systems** to facilitate AutoML research and ML development



Want to learn more? Attend
- the **AutoML conference**
- the **AutoML Fall School**

Thanks!

Katharina Eggensperger
katharina.eggensperger@uni-tuebingen.de