Universiteit Leiden

Automated Machine Learning for Science

Jan N. van Rijn

Leiden University

AWASES Workshop

[1] Based on Baratchi et al., 2024

Scope

Scope

Predictive AI vs Generative AI

Scope

Predictive AI vs Generative AI

Single Purpose AI vs Multi-Purpose AI

Scope

Predictive AI vs Generative AI

Single Purpose AI vs Multi-Purpose AI

Training of AI vs Inference using existing models

Scope

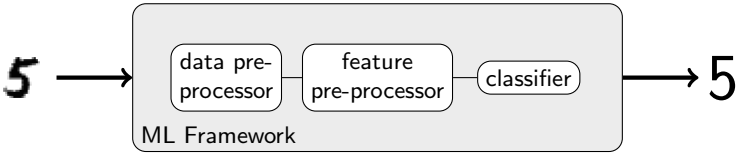
Predictive AI vs Generative AI

Single Purpose AI vs Multi-Purpose AI

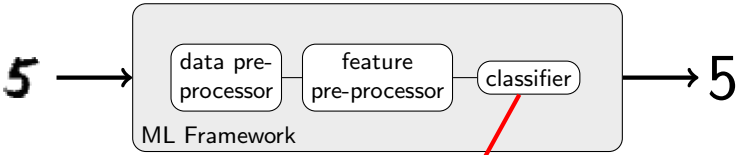
Training of AI vs Inference using existing models

AutoML can be applied to all, but today we focus **mostly** on the training of Single Purpose Predictive AI

The Machine Learning Pipeline

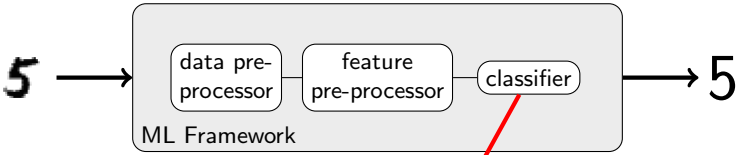


The Machine Learning Pipeline



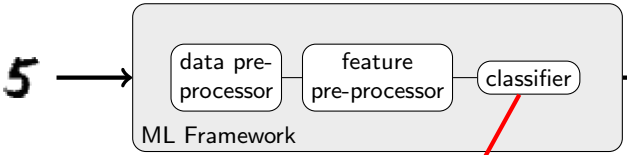
classifier	$\#\lambda$
Adaboost	4
Bernoulli Naive Bayes	2
Decision Tree	4
Extra Trees	5
Gradient Boosting	6
k-NN	3
LDA	4
...	

The Machine Learning Pipeline



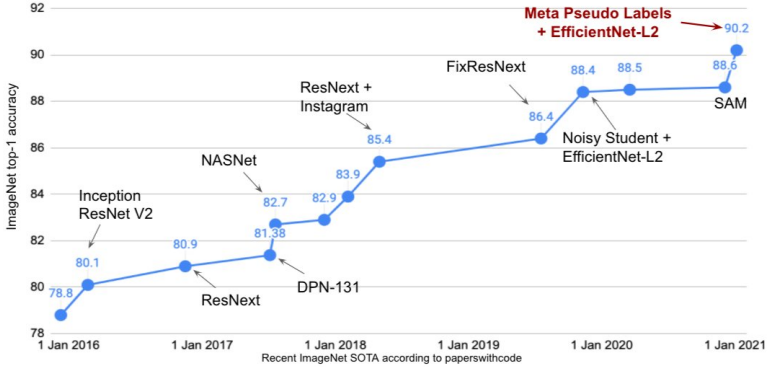
classifier	$\#\lambda$
Adaboost	4
Bernoulli Naive Bayes	2
Decision Tree	4
Extra Trees	5
Gradient Boosting	6
k-NN	3
LDA	4
...	

The Machine Learning Pipeline



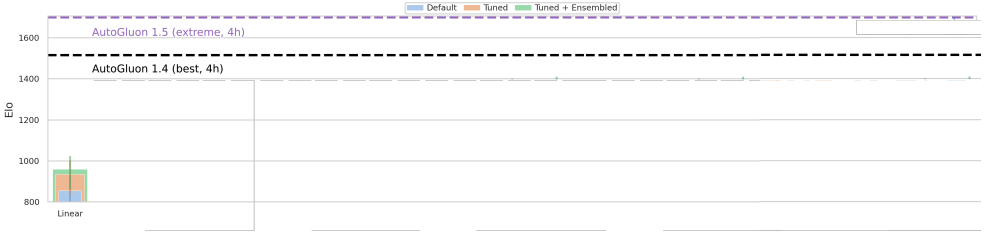
classifier	$\#\lambda$
Adaboost	4
Bernoulli Naive Bayes	2
Decision Tree	4
Extra Trees	5
Gradient Boosting	6
k-NN	3
LDA	4
...	

ImageNet – Progress of the State of the Art



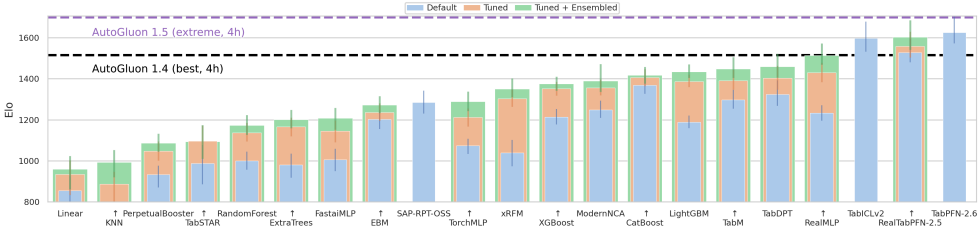
[https://x.com/quocleix/status/1349443438698143744/photo/1]

The right choices matter



[Figure: TabArena Leaderboard on Hugging Face (per May 21).]

The right choices matter

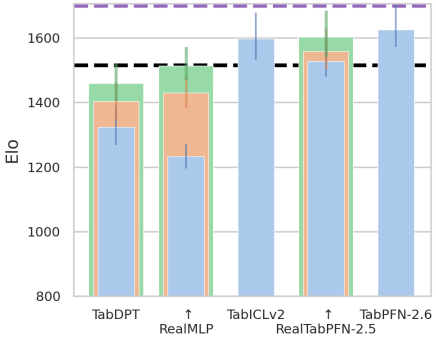


[Figure: TabArena Leaderboard on Hugging Face (per May 21).]

Various AutoML solutions

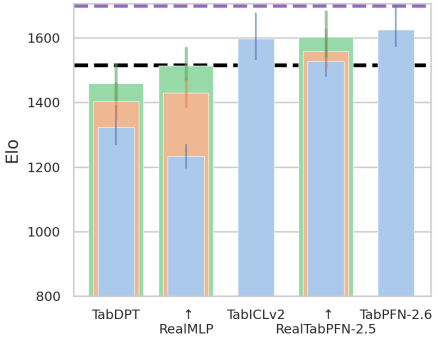
[Figure by: Satyam Kumar (2020), Medium.com blogpost.]

What about TabPFN?



[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

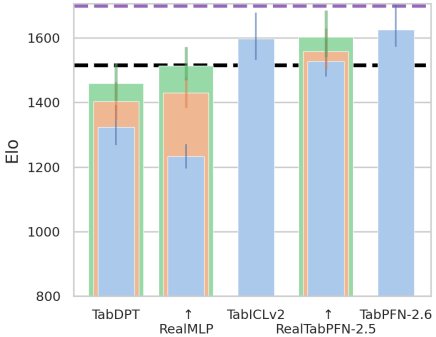
What about TabPFN?



- Foundation model for tabular data

[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

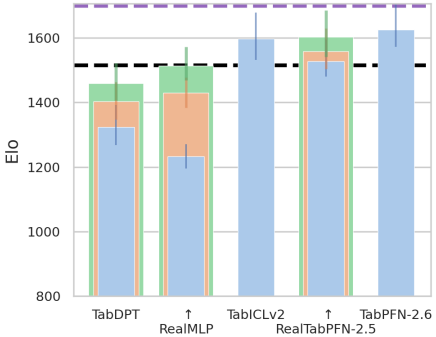
What about TabPFN?



- Foundation model for tabular data
- Based on transformer architecture

[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

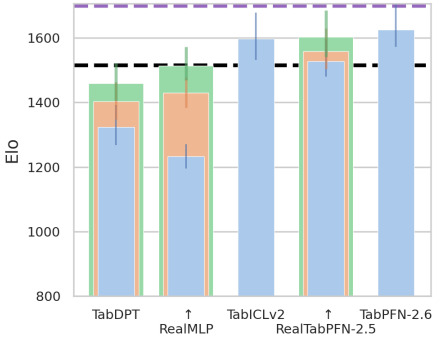
What about TabPFN?



- Foundation model for tabular data
- Based on transformer architecture
- Pre-trained on massive amounts of data, only inference needed at runtime

[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

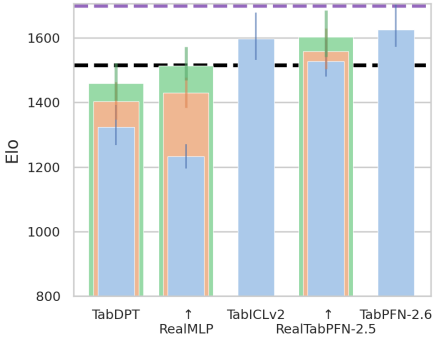
What about TabPFN?



- Foundation model for tabular data
- Based on transformer architecture
- Pre-trained on massive amounts of data, only inference needed at runtime
- In-context learning: Similar idea as LLM-inference

[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

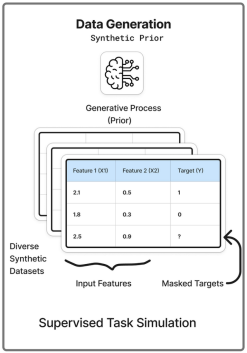
What about TabPFN?



- Foundation model for tabular data
- Based on transformer architecture
- Pre-trained on massive amounts of data, only inference needed at runtime
- In-context learning: Similar idea as LLM-inference
- Initially limited to small(er) datasets, but solved in latest version

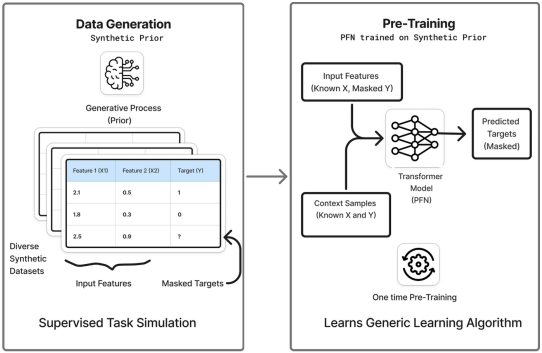
[Grinsztajn, ..., Hutter, TabPFN-3: Technical Report, May 2026]]

TabPFN architecture



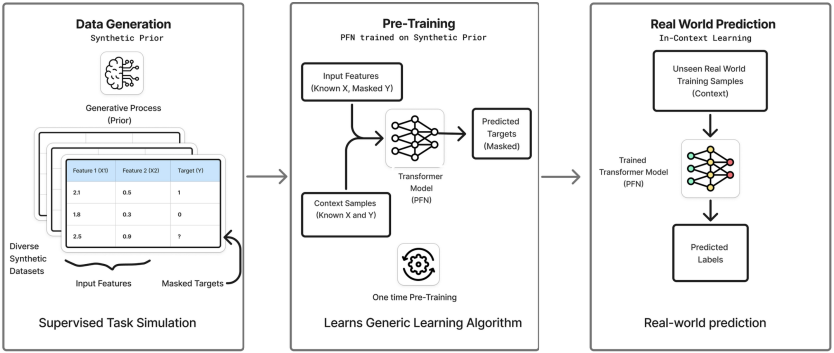
[TowardsDataScience.com]

TabPFN architecture



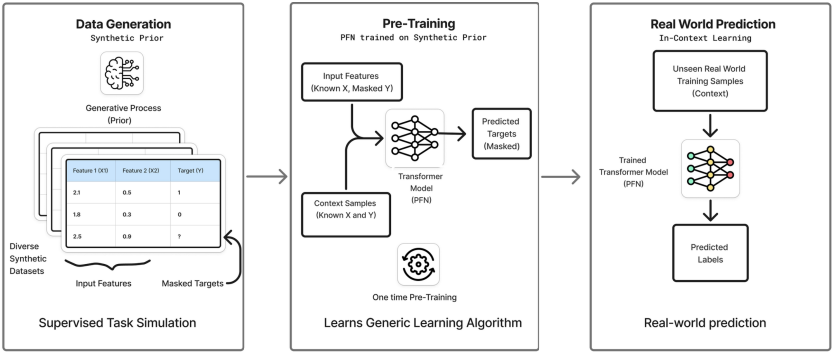
[TowardsDataScience.com]

TabPFN architecture



[TowardsDataScience.com]

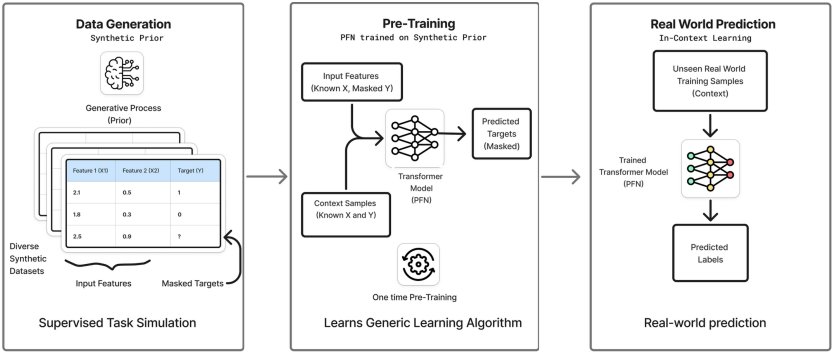
TabPFN architecture



In-context learning: Similar idea as LLM-inference

[TowardsDataScience.com]

TabPFN architecture



In-context learning: Similar idea as LLM-inference

Disadvantages: Robustness, Sustainability, Black-box

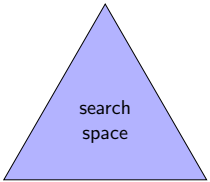
[TowardsDataScience.com]

Automated Machine Learning: Past, Present and Future

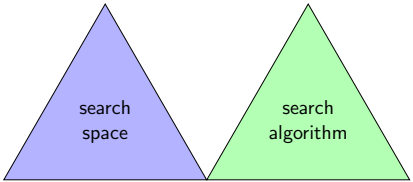
Automated Machine Learning: Past, Present and Future

search space

- hyperparameters to explore
- ranges
- which are important?
- how to sample?



Automated Machine Learning: Past, Present and Future



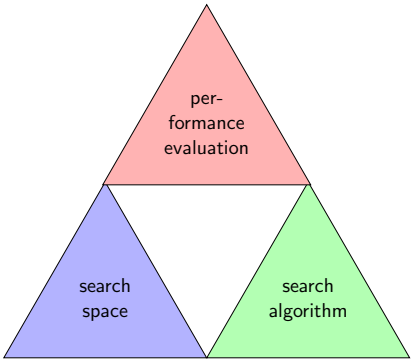
search space

- hyperparameters to explore
- ranges
- which are important?
- how to sample?

search algorithm

- random/grid search
- Bayesian optimisation
- hyperband

Automated Machine Learning: Past, Present and Future



search space

- hyperparameters to explore
- ranges
- which are important?
- how to sample?

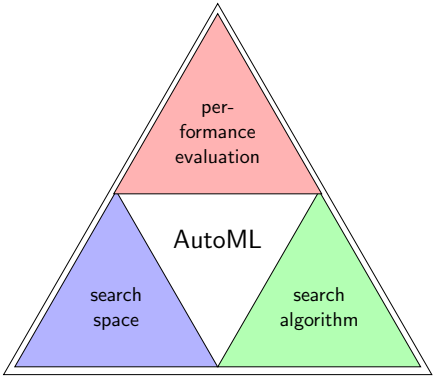
search algorithm

- random/grid search
- Bayesian optimisation
- hyperband

evaluation mechanism

- train/validation/test set
- optimisation criteria
- multi-armed bandits
- learning curves

Automated Machine Learning: Past, Present and Future



search space

- hyperparameters to explore
- ranges
- which are important?
- how to sample?

search algorithm

- random/grid search
- Bayesian optimisation
- hyperband

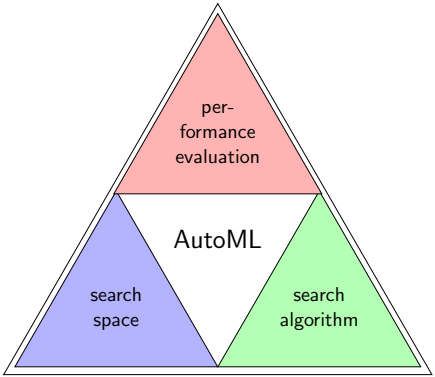
evaluation mechanism

- train/validation/test set
- optimisation criteria
- multi-armed bandits
- learning curves

AutoML

- Auto-sklearn
- Auto-pytorch
- Auto-sklearn (2.0)

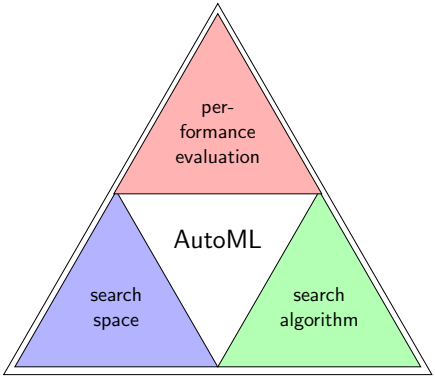
Components of AutoML



Hyperparameter optimisation problem:

$$\lambda^* \in \arg\min_{\lambda \in \Lambda} c(A_\lambda, D_{train}, D_{valid})$$

Components of AutoML



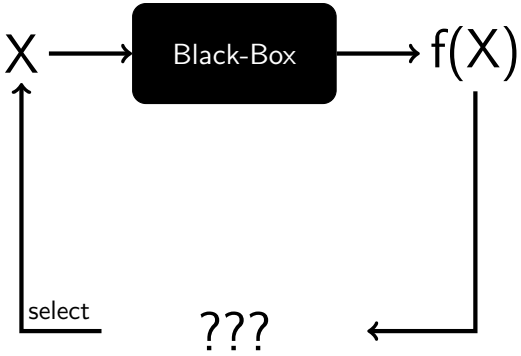
Hyperparameter optimisation problem:

$$\lambda^* \in \underset{\lambda \in \Lambda}{\arg\min}\ c(A_\lambda, D_{train}, D_{valid})$$

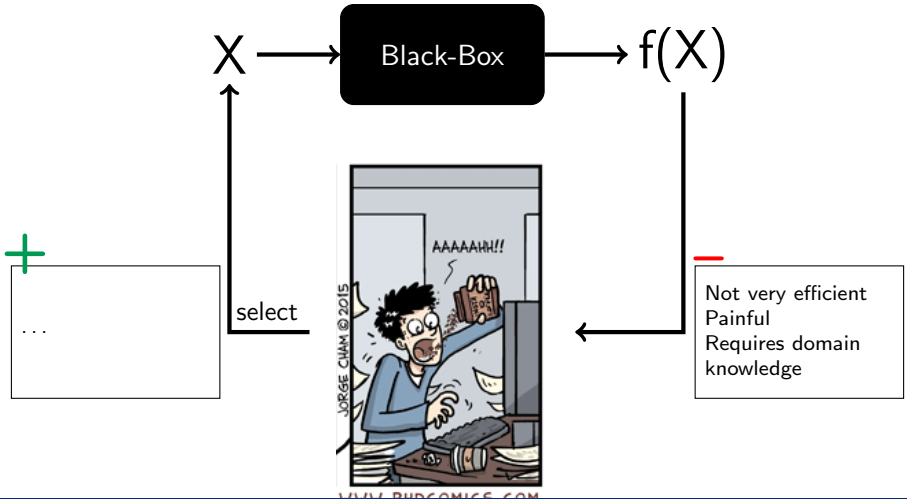
Search Algorithm

Black-Box Optimisation

The Loop



The Human in the Loop

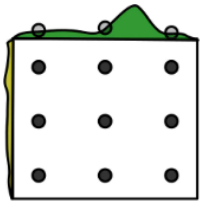


Grid Search

X → Black-Box → f(X)

\+ Very simple approach
Can be used to study the problem

select



− Does not scale to high dimensions
Grid needs to be defined

Random Search

\+

Even more simple
Easily parallelizable
Eventually converges to optimum

\−

Not data efficient
Computationally expensive

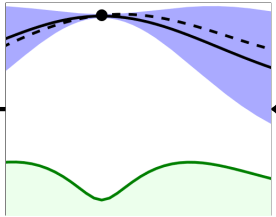
Bayesian Optimisation

X → Black-Box → f(X)

\+

Data efficient
State of the art

select



−

Not easy to paral-
lelise

Universiteit Leiden

Search Space

sklearn.neural_network.MLPClassifier

class sklearn.neural_network.MLPClassifier(*hidden_layer_sizes=(100,)*, *activation='relu'*, ***, *solver='adam'*, *alpha=0.0001*, *batch_size='auto'*, *learning_rate='constant'*, *learning_rate_init=0.001*, *power_t=0.5*, *max_iter=200*, *shuffle=True*, *random_state=None*, *tol=0.0001*, *verbose=False*, *warm_start=False*, *momentum=0.9*, *nesterovs_momentum=True*, *early_stopping=False*, *validation_fraction=0.1*, *beta_1=0.9*, *beta_2=0.999*, *epsilon=1e-08*, *n_iter_no_change=10*, *max_fun=15000*) [source]

Multi-layer Perceptron classifier.

This model optimizes the log-loss function using LBFGS or stochastic gradient descent.

New in version 0.18.

Parameters:

hidden_layer_sizes : *tuple, length = n_layers - 2, default=(100,)*
The ith element represents the number of neurons in the ith hidden layer.

activation : *{'identity', 'logistic', 'tanh', 'relu'}, default='relu'*
Activation function for the hidden layer.

- 'identity', no-op activation, useful to implement linear bottleneck, returns f(x) = x
- 'logistic', the logistic sigmoid function, returns f(x) = 1 / (1 + exp(-x)).
- 'tanh', the hyperbolic tan function, returns f(x) = tanh(x).
- 'relu', the rectified linear unit function, returns f(x) = max(0, x)

solver : *{'lbfgs', 'sgd', 'adam'}, default='adam'*
The solver for weight optimization.

- 'lbfgs' is an optimizer in the family of quasi-Newton methods.
- 'sgd' refers to stochastic gradient descent.
- 'adam' refers to a stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba

Note: The default solver 'adam' works pretty well on relatively large datasets (with thousands of training samples or more) in terms of both training time and validation score. For small datasets, however, 'lbfgs' can

Configuration Space

- Categorical
  - e.g., activation function
- Numerical
  - continuous, e.g., learning rate, learning rate init
  - integer, e.g., hidden layers size, batch size
- Conditional Hyperparameters
  - e.g., beta_1, beta_2 (for controlling Adam)
- Other important aspects:
  - Range
  - Sampling strategy (uniform, log-scale)
  - Some hyperparameters should not be optimised (random seed, ensemble size, . . .)

*Scikit-learn implementation does not cover interesting features that are covered by Tensorflow and pyTorch

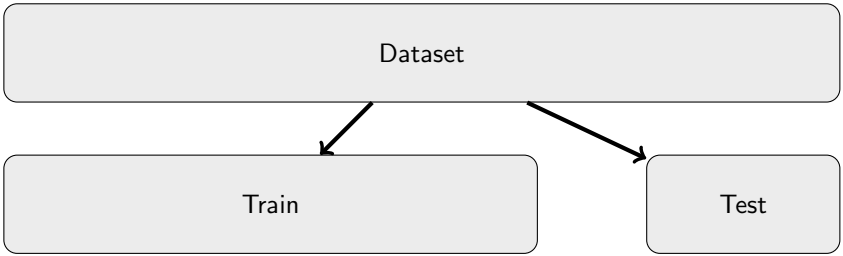
Universiteit Leiden

Evaluation Mechanism

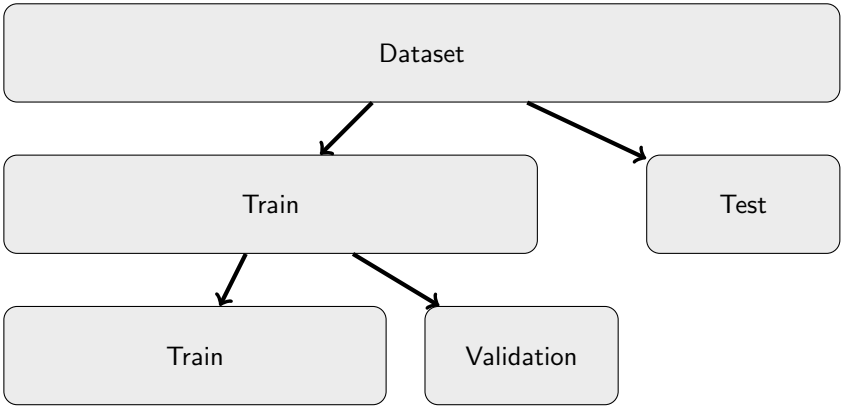
Test set

- (automated) machine learning is all about generalisation
- Tested using a test set, that can only be inspected once!
- Repeatedly inspecting the test set:
  - violation of assumptions
  - leads to overly optimistic performance estimations

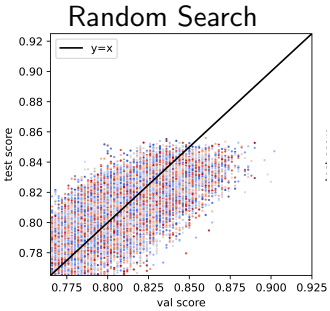
Validation set Vs. Test set



Validation set Vs. Test set



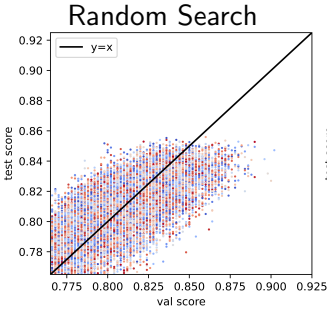
Overfitting



Colour indicates the iteration in the random search process

Figure by Schröder et al. (2024), MSc thesis.

Overfitting



Colour indicates the iteration in the random search process

Figure by Schröder et al. (2024), MSc thesis.

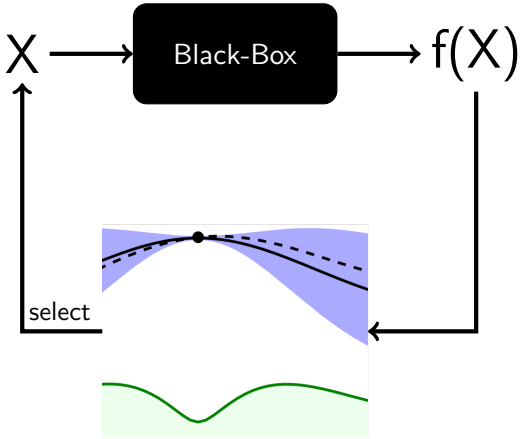
Effects of overfitting can be mitigated by

- using a large validation set (at the cost of train data)
- having multiple classes
- multiple validation folds (e.g., use multiple folds at inner level)
- shuffling validation and test splits

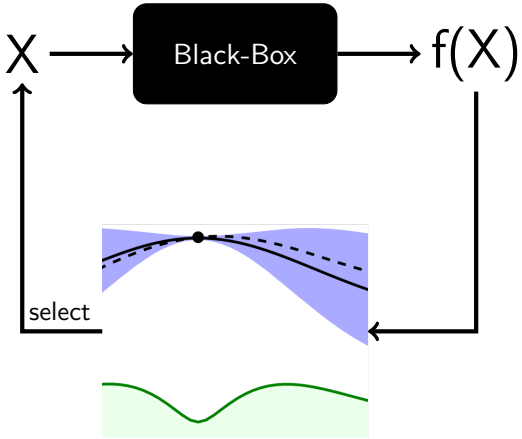
Universiteit Leiden

Explainable AutoML

Rashomon set and AutoML



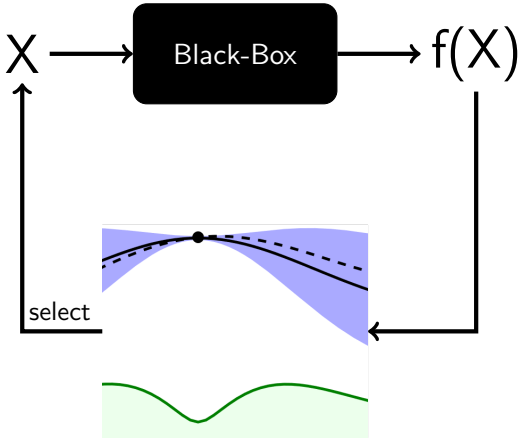
Rashomon set and AutoML



- Rashomon set is already used in practice (e.g., AutoGluon)

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

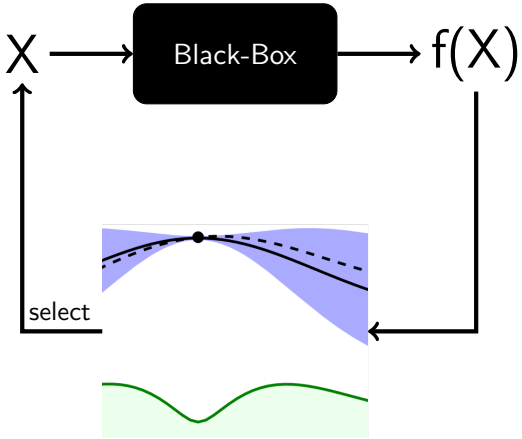
Rashomon set and AutoML



- Rashomon set is already used in practice (e.g., AutoGluon)
- Goal: more insight into the black box using the Rashomon set

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

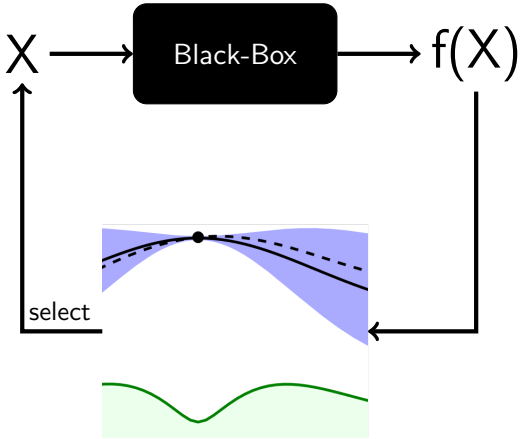
Rashomon set and AutoML



- Rashomon set is already used in practice (e.g., AutoGluon)
- Goal: more insight into the black box using the Rashomon set
- Partial Dependency Plots: AutoML for regression

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

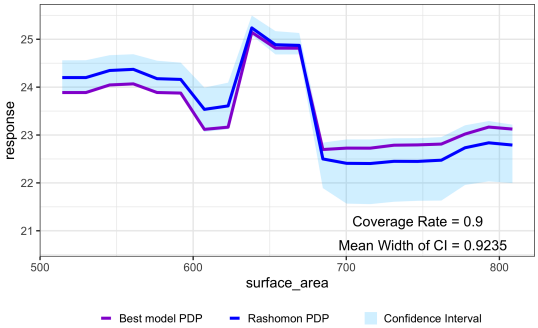
Rashomon set and AutoML



- Rashomon set is already used in practice (e.g., AutoGluon)
- Goal: more insight into the black box using the Rashomon set
- Partial Dependency Plots: AutoML for regression
- Discrepancy in explanation is not necessarily bad

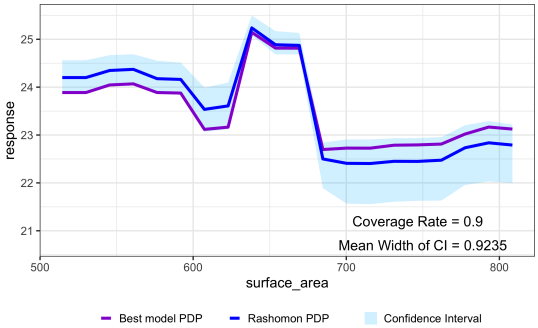
[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

Partial Dependency Plot



[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

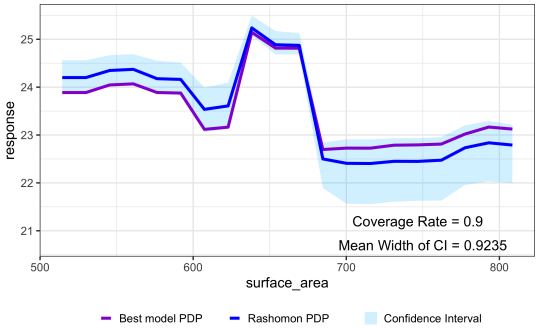
Partial Dependency Plot



- Shows for each input variable the expected output

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

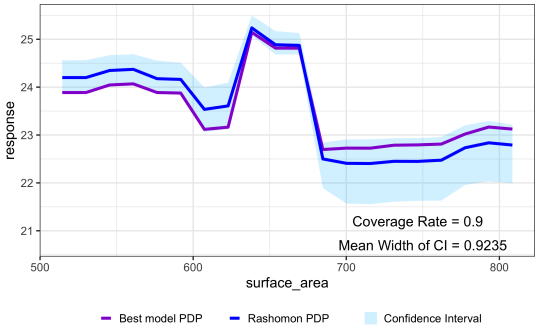
Partial Dependency Plot



- Shows for each input variable the expected output
- Metric: Mean with of Confidence Interval

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

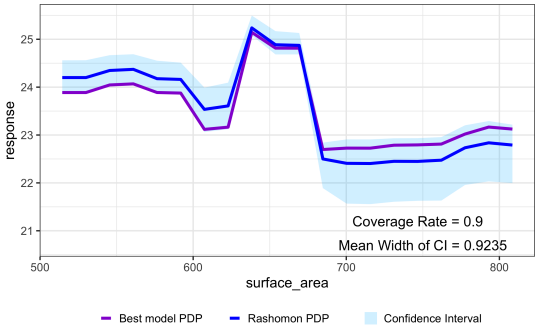
Partial Dependency Plot



- Shows for each input variable the expected output
- Metric: Mean with of Confidence Interval
- Metric: Coverage rate of single best model

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

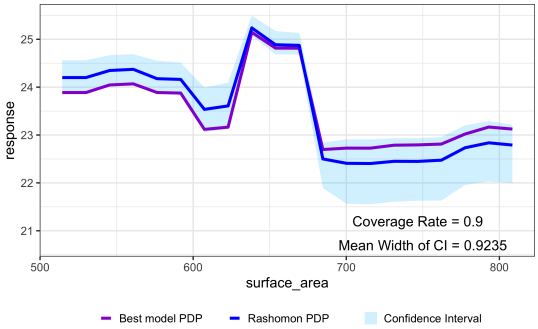
Partial Dependency Plot



- Shows for each input variable the expected output
- Metric: Mean with of Confidence Interval
- Metric: Coverage rate of single best model
- No ground-truth available for real-world datasets

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

Partial Dependency Plot

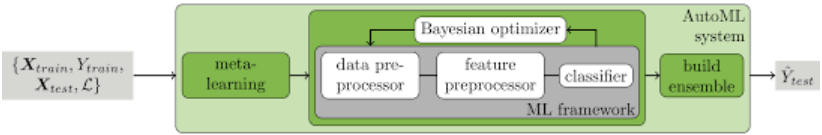


- Shows for each input variable the expected output
- Metric: Mean with of Confidence Interval
- Metric: Coverage rate of single best model
- No ground-truth available for real-world datasets
- https://github.com/mcavs/ - AutoML integration on the way

[Cavus et al., Quantifying Model Uncertainty with AutoML and Rashomon Partial Dependence Profiles: Enabling Trustworthy and Human-centered XAI, 2026]

AutoML

Auto-sklearn



- Defines a distance function between tasks
- Initialises Bayesian optimisation with configurations that worked well on similar tasks
- Additionally: builds ensembles of several well-working solutions
- See also: Auto-WEKA, Auto-sklearn 2.0

Figure and content based on original version. [Feurer et al. (2015), Efficient and Robust Automated Machine Learning.]

A Simple Example

The first step is to prepare your data. Here we use the MNIST dataset as an example

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print(x_train.shape)  # (60000, 28, 28)
print(y_train.shape)  # (60000,)
print(y_train[:3])  # array([7, 2, 1], dtype=uint8)
```

The second step is to run the ImageClassifier. It is recommended have more trials for more complicated datasets. This is just a quick demo of MNIST, so we set max_trials to 1. For the same reason, we set epochs to 10. You can also leave the epochs unspecified for an adaptive number of epochs.

```
# Initialize the image classifier.
clf = ak.ImageClassifier(overwrite=True, max_trials=1)
# Feed the image classifier with training data.
clf.fit(x_train, y_train, epochs=10)


# Predict with the best model.
predicted_y = clf.predict(x_test)
print(predicted_y)


# Evaluate the best model with testing data.
print(clf.evaluate(x_test, y_test))
```

AutoML libraries

[Figure by: Satyam Kumar (2020), Medium.com blogpost.]

AutoML libraries

[Figure by: Satyam Kumar (2020), Medium.com blogpost.]

Some distinguishing features:

- Bayesian optimisation
- Genetic algorithms
- Multi-armed bandits
- Ensembles
- Meta-learning
- Stacking
- Multi-layer stacking
- ...

AutoML libraries

[Figure by: Satyam Kumar (2020), Medium.com blogpost.]

Some distinguishing features:

- Bayesian optimisation
- Genetic algorithms
- Multi-armed bandits
- Ensembles
- Meta-learning
- Stacking
- Multi-layer stacking
- ...

Which AutoML tool to select?

Conclusions

- AutoML is already useful tool for supporting AI in Science

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available, **often better than LLMs**

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available, **often better than LLMs**
- TabFPN state-of-the-art on many (not all) tasks

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available, **often better than LLMs**
- TabFPN state-of-the-art on many (not all) tasks
- Generally: Trade human time for computer time

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available, **often better than LLMs**
- TabFPN state-of-the-art on many (not all) tasks
- Generally: Trade human time for computer time
- Using AutoML frees up time for human to focus on supervisory jobs in the loop

Conclusions

- AutoML is already useful tool for supporting AI in Science
- Various easy-to-use tools available, **often better than LLMs**
- TabFPN state-of-the-art on many (not all) tasks
- Generally: Trade human time for computer time
- Using AutoML frees up time for human to focus on supervisory jobs in the loop
- Explainable AI methods available for traditional AutoML

OpenML

A worldwide machine learning lab

Machine learning research should be easily accessible and reusable. OpenML is an open platform for sharing datasets, algorithms, and experiments - to learn how to learn better, together.

Search OpenML Datasets

Sign Up to start tracking and sharing your own work. OpenML is open and free to use.

AI-ready data

All datasets are uniformy formatted, have rich, consistent metadata, and can be loaded directly into your favourite environments.

ML library integrations

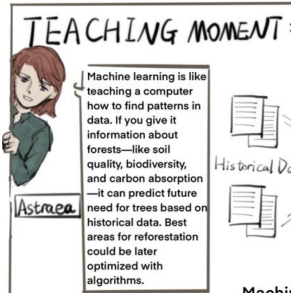
Pipelines and models can be shared directly from your favourite machine learning libraries. No manual steps required.

A treasure trove of ML results

Learn from millions of reproducible machine learning experiments on thousands of datasets to make informed decisions.

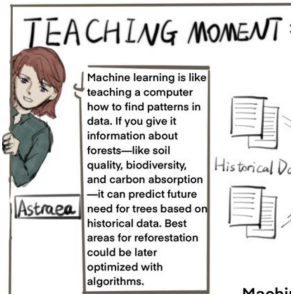
Universiteit Leiden

AutoML Comic

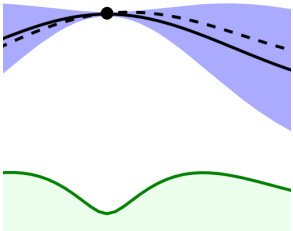


Universiteit Leiden

AutoML Comic

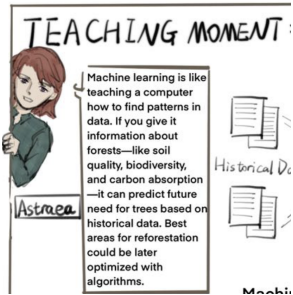


AutoML intro



Universiteit Leiden

AutoML Comic



AutoML intro