Universiteit Leiden

Automated Machine Learning using OpenML

Jan N. van Rijn

Leiden University

Leiden - Chile International Research Seminar on Computer Science and AI

Motivation

- Galileo Galilei (1564–1642)
- Created the best telescopes
- Discovered the rings of Saturn

Motivation

- Galileo Galilei (1564–1642)
- Created the best telescopes
- Discovered the rings of Saturn
- Sent anagrams of his discoveries, instead of publishing the results

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

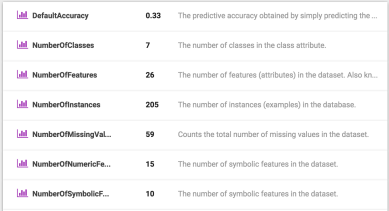
Learn from millions of reproducible machine learning experiments on thousands of datasets to make informed decisions.

Datasets

- Data (ARFF) uploaded or referenced, versioned
- Analysed, characterized, organized on line
- Indexed based on name, meta-features, tags, etc.
- Support for other data formats (on request)

26 features

72 properties



Previous sklearn.datas... Next sklearn.datas... Up API Reference

scikit-learn v0.21.dev0
Other versions

Please cite us if you use the software.

sklearn.datasets.fetch_openml

Examples using sklearn.datasets.fetch_openml

sklearn.datasets.fetch_openml

sklearn.datasets. **fetch_openml** (*name=None, version='active', data_id=None, data_home=None, target_column='default-target', cache=True, return_X_y=False*) [source]

Fetch dataset from openml by name or dataset id.

Datasets are uniquely identified by either an integer ID or by a combination of name and version (i.e. there might be multiple versions of the 'iris' dataset). Please give either name or data_id (not both). In case a name is given, a version can also be provided.

Read more in the User Guide.

Note: EXPERIMENTAL

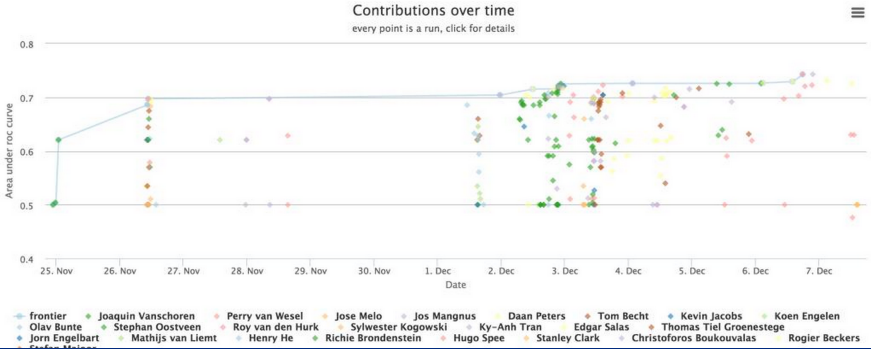
The API is experimental in version 0.20 (particularly the return value structure), and might have small backward-incompatible changes in future releases.

Parameters: **name : *str or None***

String identifier of the dataset. Note that OpenML can have multiple datasets with the same name.

Tasks

- Data alone does not define an experiment
- Tasks contain: data, target attribute, goals, procedures
- Readable by tools, automates experimentation
- Real time 'leaderboard' and overview



Tasks

Various Task Types:

- Supervised Classification
- Supervised Regression
- Learning Curve Classification (time intensive)
- Data Stream Classification (on line learning)
- Survival Analysis
- Clustering (Work in Progress)
- Machine Learning Challenge
- Easily extendable to more ..

Flows (algorithms)

- Run locally, auto-registered by tools
- Integrations + APIs (REST, Java, R, Python, ...)

Flows (algorithms)

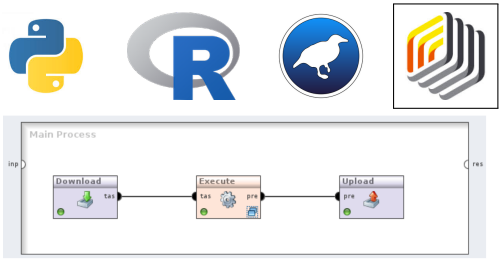
- Run locally, auto-registered by tools
- Integrations + APIs (REST, Java, R, Python, ...)

```
import openml
from sklearn import ensemble

task = openml.tasks.get_task(3954)
clf = ensemble.RandomForestClassifier()
run = openml.runs.run_model_on_task(task, clf)
result = run.publish()
```

Flows (algorithms)

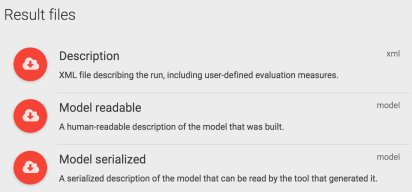
- Run locally, auto-registered by tools
- Integrations + APIs (REST, Java, R, Python, . . .)

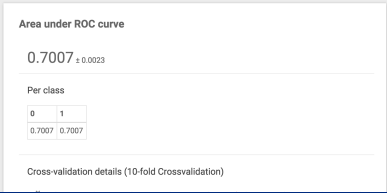


(a) Main Workflow

★ Runs

- Flow uploads predictions
- Predictions are evaluated on OpenML
- Reproducible, linked to data, flows and researcher
- Contains:
  - predictions
  - hyperparameter settings
  - model information
  - evaluation measures





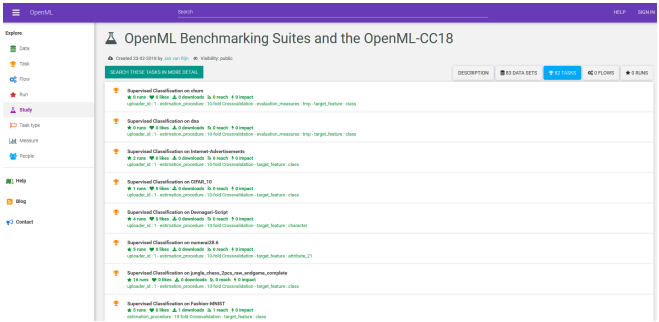
★ Runs

Answer basic questions about performance of algorithms to study ...

- Computation is done client side
- The results (serialized model, predictions) are uploaded to OpenML
- OpenML evaluates the results and calculates default evaluation measures
- Users can calculate custom evaluation measures and share them on OpenML

Studies

- Bundles Data, Flows, Tasks and Runs
- Link attached to publication
- (Work in Progress): attach a notebook



Intermediate Summary

OpenML ...

- offers an eco-system to make Machine Learning experiments re-useable to other scientists
- is integrated in various programming languages and ML toolboxes, including Scikit-learn, Weka, ...
- has an active community; community meetings take place twice a year
- has more than 20,000 unique monthly users (and growing)

Analysis

Having access to many datasets, algorithms and run results allows researchers to ...

- Experiment on a larger set of dataset
- Compare against optimized and state-of-the-art algorithms
- Experiments on large meta-data
- Perform decent meta-learning experiments

Projects

Several projects done with OpenML:

- OpenML Benchmark Suites
- Myth Busting Urban Legends
- Automated Machine Learning
- Hyperparameter Importance
- (Learning good defaults)

Universiteit Leiden

OpenML Benchmark Suites

OpenML Benchmark Suites

Common limitations of experimental evaluations:

- Often done on a small selection of datasets
  - Not sure about generalization to more datasets
  - 'Cherry picking'
- Publication bias
  - Every paper only reports good results
  - More useful to know WHEN (on what type of data) a new algorithm performs well
- Hard to compare conclusions across papers
- Abused datasets (example: liver-disorders)

OpenML Benchmark Suites

OpenML-100: A curated benchmark suite [Bischl et al., 2017]. Inclusion criteria:

- Basic data properties (500–100000 observations, $<$ 5000 features, $>=$ 2 classes)
- The ratio of the minority class and the majority class > 0.05
- Scientific publication introducing the dataset and learning task
- Pure classification tasks only (no data streams, multi-class tasks)
- No artificial data, binarized versions of regression datasets or sub-samples of bigger datasets
- Disadvantage: Very hard to get a good consensus about the inclusion criteria
- Next: The OpenML-CC18 (approx. 85 datasets)

OpenML Benchmark Suites

Ease of use (Scikit-learn, Weka, mlR)

```
import openml
import sklearn
# obtain the benchmark suite
benchmark_suite = openml.study.get_study('OpenML100','tasks')
# build a sklearn classifier
clf = sklearn.pipeline.Pipeline(
        steps=[('imputer',sklearn.preprocessing.Imputer()),
               ('estimator',sklearn.tree.DecisionTreeClassifier())])

for task_id in benchmark_suite.tasks: # iterate over all tasks
  # download the OpenML task
  task = openml.tasks.get_task(task_id)
  # get the data (not used in this example)
  X, y = task.get_X_and_y()
  # set the OpenML Api Key
  openml.config.apikey = 'FILL_IN_OPENML_API_KEY'
  # run classifier on splits (requires API key)
  run = openml.runs.run_model_on_task(task,clf)
  # print accuracy score
  score = run.get_metric_score(sklearn.metrics.accuracy_score)
  print('Data set: %s; Accuracy: %0.2f' % (task.get_dataset().name,score.mean()))
  # publish the experiment on OpenML (optional, requires API key)
  run.publish()
  print('URL for run: %s/run/%d' %(openml.config.server,run.run_id))
```


BUSTING THE
MYTHS

Myth Busting for Data Mining

Universiteit Leiden

- Papers are generally build upon claims that are not well grounded, e.g.,
  - "We performed data transformation X because it is common practise."
  - "We set hyperparameter Y to value Z because the authors recommended these values."
- We can empirically analyze the validity of these claims on the meta-data from OpenML

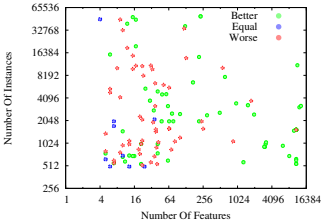
Effect of Feature Selection

Universiteit Leiden

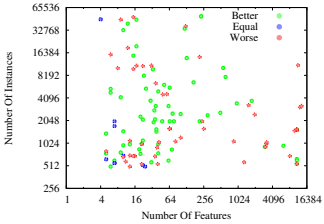
Experimental Setting by Post et al. [2016]:

- 400 binary classification datasets from OpenML
- 12 algorithms from Weka
- Correlation-based Feature Subset Selection
- We added runs that not existed on OpenML
- Recorded Area Under the ROC curve
- Limitation: Hyperparameter Optimization

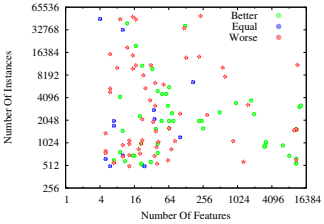
Effect of Feature Selection



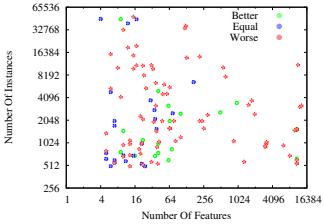
k-NN



Naive Bayes



J48



SMO

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

Linear	Non-linear
ease of tuning	
interpretability	
fit risk	
performance	

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability		
fit risk		
performance		

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk		
performance		

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk	underfit	overfit
performance		

Linear vs. Non-linear

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk	underfit	overfit
performance	-	+

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk	underfit	overfit
performance	-	+
Tree	Decision Stump	Decision Tree

Linear vs. Non-linear

Universiteit Leiden

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk	underfit	overfit
performance	-	+
Tree	Decision Stump	Decision Tree
SVM	Linear Kernel	Gaussian Kernel

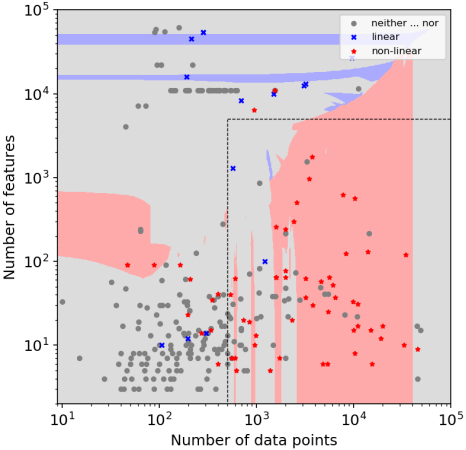
Universiteit Leiden

Linear vs. Non-linear

Reoccuring topic in literature, see [Strang et al., 2018] for several examples

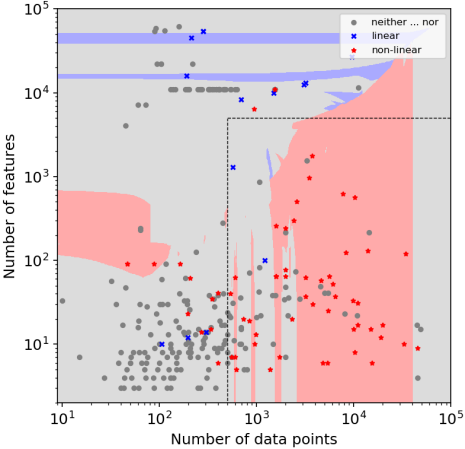
	Linear	Non-linear
ease of tuning	+	-
interpretability	+	-
fit risk	underfit	overfit
performance	-	+
Tree	Decision Stump	Decision Tree
SVM	Linear Kernel	Gaussian Kernel
Neural Network	Perceptron	MLP

Linear vs. Non-Linear

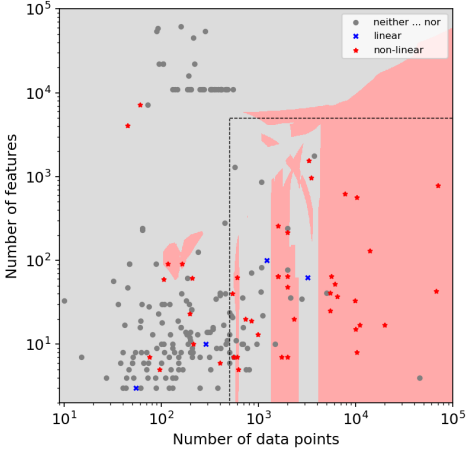


SVM

Linear vs. Non-Linear



SVM



Neural Networks

Effect of Feature Selection

Universiteit Leiden

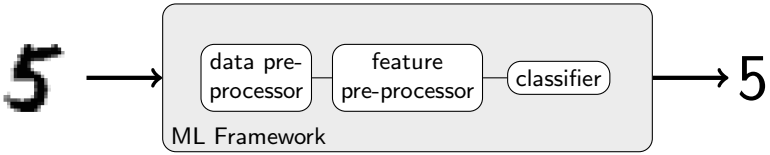
Conclusions:

- Most results as expected:
  - Feature selection is often beneficial for the classifiers for which we expect it to be: k-NN and Naive Bayes
  - Non-linear classifiers exclusively better than linear classifier
- Whether or not to use feature selection can be learned (see paper)
- Low amount of datasets on which feature selection significantly effects performance potentially indicates data bias
- Realization: Limitation of OpenML100

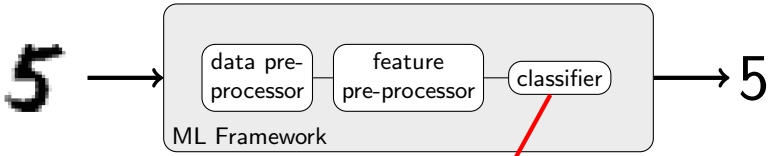
Universiteit Leiden

Automated Machine Learning

The Machine Learning Pipeline

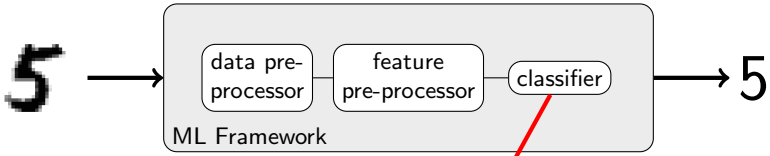


The Machine Learning Pipeline



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 Problem Definition

$$A^*, \lambda_* \in \underset{A \in \mathbf{A}, \lambda \in \Lambda}{\arg\min}\, L(A_{\lambda, D_{train}}, D_{valid})$$

Given a dataset D, find an algorithm A^* and its hyperparameters λ_* that minimizes a given loss function L

Auto-sklearn

```
import autosklearn.classification
import sklearn.model_selection
import sklearn.datasets
import sklearn.metrics

# Load data and split into train and test data
X, y = sklearn.datasets.load_digits(return_X_y=True)
X_train, X_test, y_train, y_test = \
  sklearn.model_selection.train_test_split(X, y)
# Let auto-sklearn run for 1h
autosklearn.classification.AutoSklearnClassifier(
  time_left_for_this_task=3600)
automl.fit(X_train, y_train)
y_hat = automl.predict(X_test)
print("Accuracy score",
      sklearn.metrics.accuracy_score(y_test, y_hat))
```

Feurer et al., Efficient and robust automated machine learning, 2015

Various AutoML systems based on OpenML data

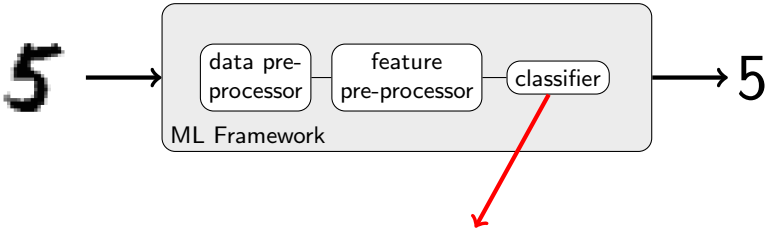
Other meta-learning and Automated Machine Learning papers that use OpenML data:

- Precision-Recall curves (done right) by Flach and Kull [2015]
- Meta-QSAR by Olier et al. [2018]
- Hyperband by Li et al. [2017]
- Auto-sklearn by Feurer et al. [2015]
- BLAST by van Rijn et al. [2018]
- ...

Universiteit Leiden

Hyperparameter Importance

Hyperparameter Optimization



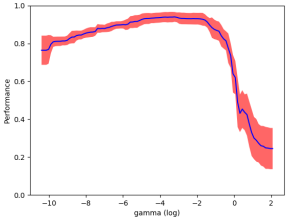
classifier	#λ
Adaboost	4
Bernoulli Naive Bayes	2
Decision Tree	4
Extra Trees	5
Gradient Boosting	6
k-NN	3
LDA	4
...	

Hyperparameter Importance

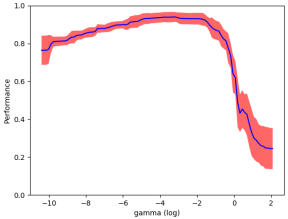
Experimental Setting by van Rijn and Hutter [2017]:

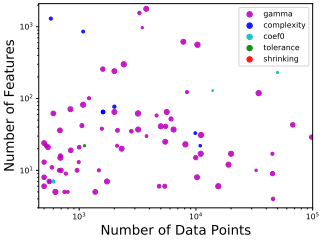
- All datasets from the OpenML-100
- Four classifiers from scikit-learn (Random Forest, Adaboost and SVM with various kernels)
- Include results with at least 200 runs (try to generate if not enough)
- Functional ANOVA
- Limitation: No conditional hyperparameters

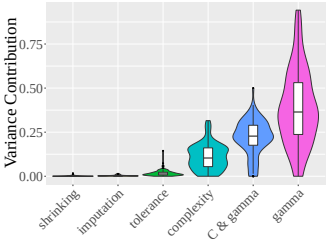
Hyperparameter Importance



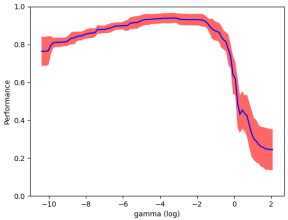
Hyperparameter Importance

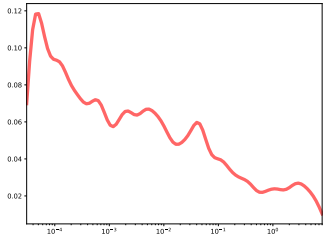


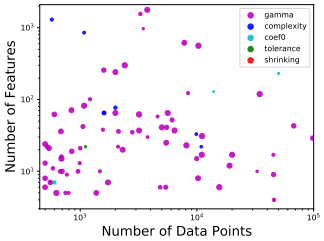


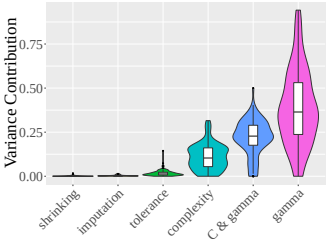


Hyperparameter Importance

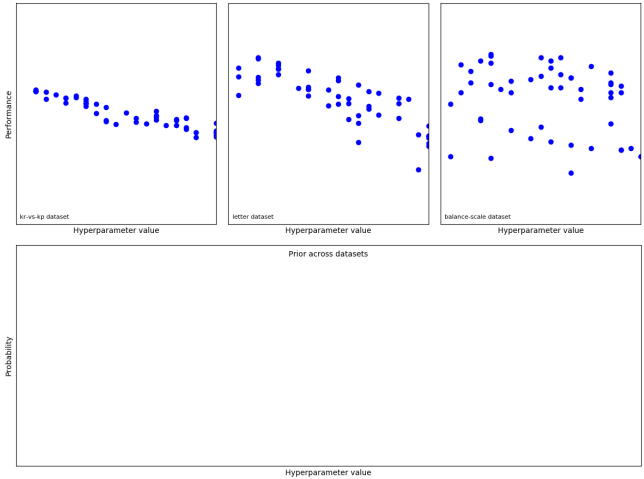




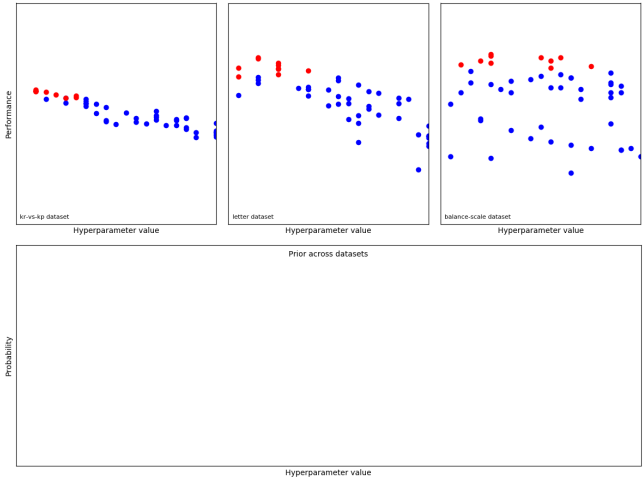




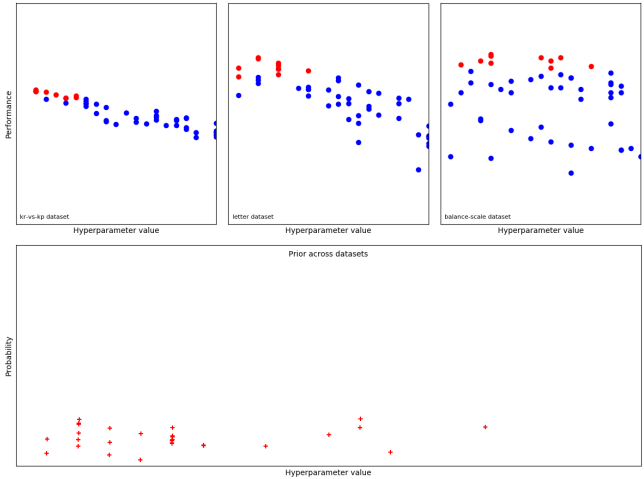
Hyperparameter Importance across Datasets



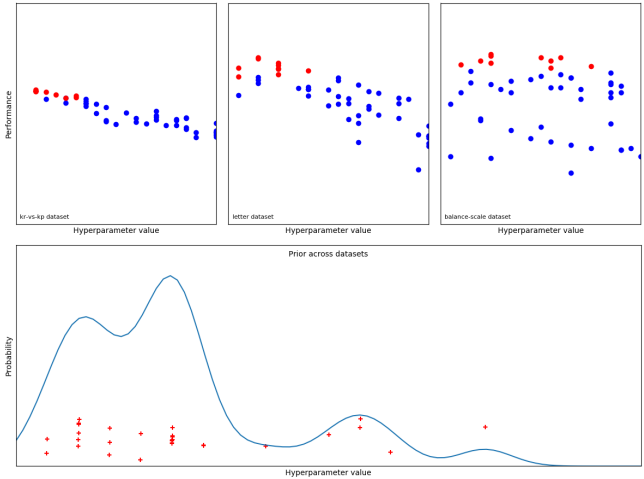
Hyperparameter Importance across Datasets



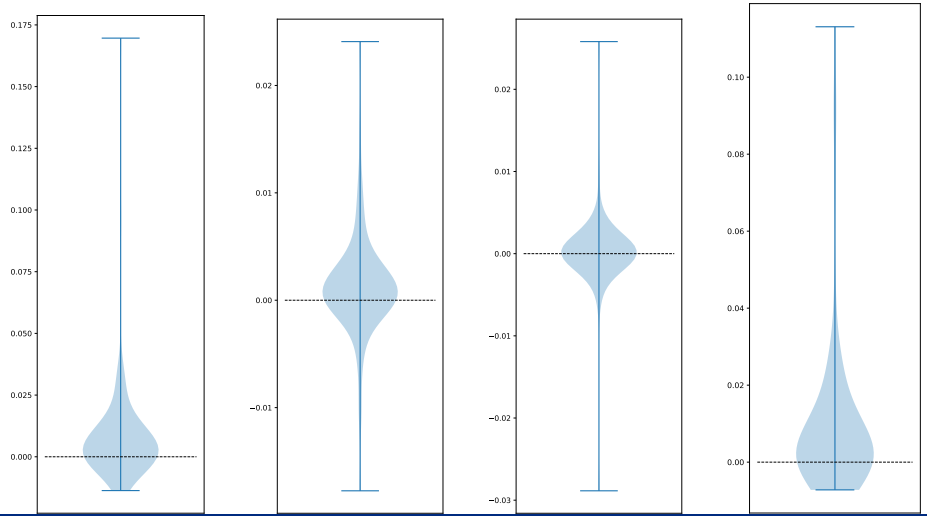
Hyperparameter Importance across Datasets



Hyperparameter Importance across Datasets



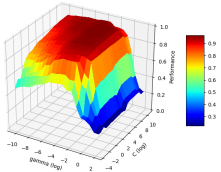
Hyperparameter Importance



Hyperparameter Importance

Initial conclusions from a study by van Rijn and Hutter [2017]:

- Functional ANOVA is a consistent tool for Hyperparameter Importance Analysis
- Obtained expected results (gamma and complexity) and new insights (Random Forest, Adaboost)
- Imputation of Missing Values
- Inferring priors leads to statistically significant better Hyperparameter Optimization results (Random Search and Hyperband)
- Video: `https://www.youtube.com/watch?v=mS4vL7_rSWQ`



Learning Multiple Defaults

Defaults . . .

- are unfortunatelly often used, both in research and industry
- disadvantages:
  - does not work well across datasets
  - several algorithms are extremely sensitive to proper hyperparameter values
  - better alternatives (BO, Hyperband, . . .)

Defaults . . .

- are unfortunatelly often used, both in research and industry
- disadvantages:
  - does not work well across datasets
  - several algorithms are extremely sensitive to proper hyperparameter values
  - better alternatives (BO, Hyperband, . . .)
- advantages
  - easy to use
  - robust
  - easy to implement

Multiple defaults

- Per algorithm a list of n defaults
- Static (not data dependent)
- Design criteria:
  - Should work well together
  - Should work well across multiple datasets

Multiple defaults

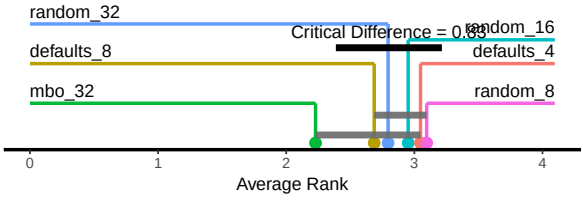
	θ_1	θ_2	...	θ_m
dataset 1	0.1	0.05	...	0.16
dataset 2	0.4	0.25	...	0.04
⋮	⋮	⋮	⋮	⋮
dataset k	0.17	0.11	...	0.31

Multiple defaults

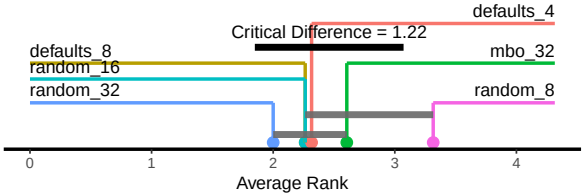
	θ_1	θ_2	...	θ_m
dataset 1	0.1	0.05	...	0.16
dataset 2	0.4	0.25	...	0.04
⋮	⋮	⋮	⋮	⋮
dataset k	0.17	0.11	...	0.31

Select a set of columns such that the sum of the row-wise minimums of these columns is minimized

Multiple defaults



adaboost

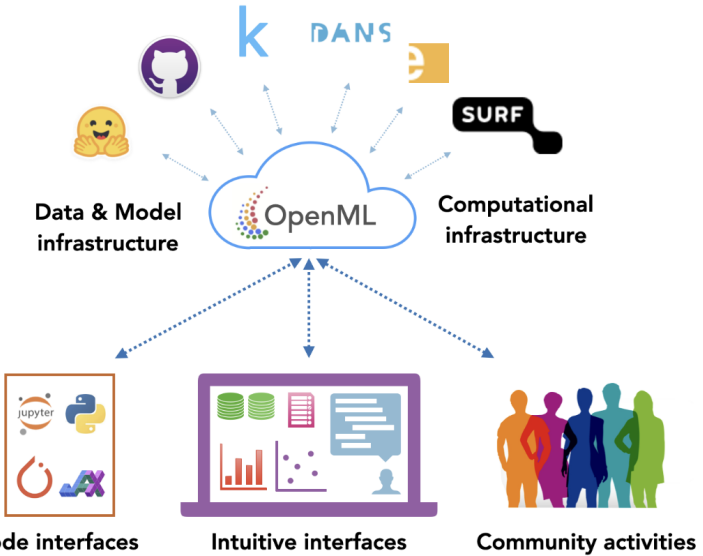


svm

random_32

Universiteit Leiden

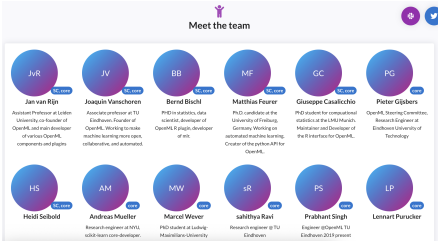
The Future of OpenML



Vacancy:

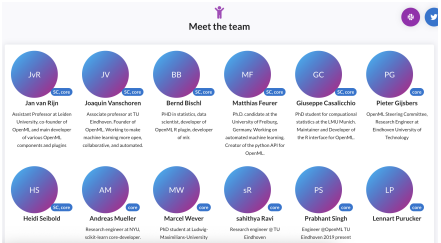
(NL-based, 100% funded)

OpenML Community



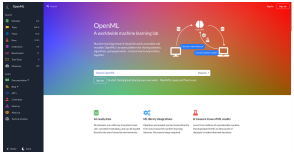
- Researchers (mainly) from Europe and US
- Core team from 9 different Universities
- Twice a year community meeting
- Workshop @ Leiden had approximately 50 attendees
- Next year: OpenML @ Shonnan (Japan)
- Current vacancy: https://tinyurl.com/4hjev8vh (NL-based, 100% funded)

Summarizing

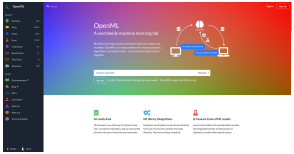


- https://www.openml.org/
- Source for Datasets, Algorithms and earlier experimental results
- All open source and freely available
- Benchmark Suites (OpenML-100, OpenML-CC18)
- Hyperparameter optimization and importance
- Mythbusting for Data Mining
- Meta-learning
- Available for further explanations and collaborations
- Now: break. After: practical session

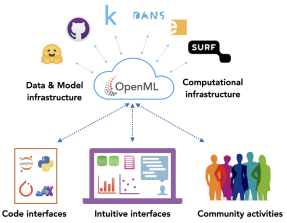
OpenML



OpenML

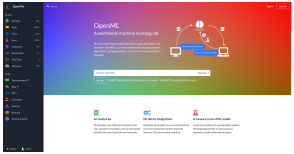


Vacancy

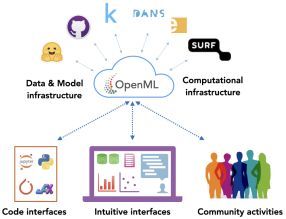


Universiteit Leiden

OpenML



Vacancy



Tutorial

References

B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren. Openml benchmarking suites and the openml100. *arXiv preprint arXiv:1708.03731*, 2017.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In *Advances in neural information processing systems*, pages 2962–2970, 2015.

P. Flach and M. Kull. Precision-recall-gain curves: PR analysis done right. In *Advances in Neural Information Processing Systems*, pages 838–846, 2015.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-Based Configuration Evaluation for Hyperparameter Optimization. In *Proc. of ICLR 2017*, 2017.

I. Olier, N. Sadawi, G. R. Bickerton, J. Vanschoren, C. Grosan, L. Soldatova, and R. D. King. Meta-QSAR: a large-scale application of meta-learning to drug design and discovery. *Machine Learning*, pages 1–27, 2018.

F. Pfisterer, J. N. van Rijn, P. Probst, A. Müller, and B. Bischl. Learning multiple defaults for machine learning algorithms, 2018.

M. J. Post, P. van der Putten, and J. N. van Rijn. Does Feature Selection Improve Classification? A Large Scale Experiment in OpenML. In *Advances in Intelligent Data Analysis XV*, pages 158–170. Springer, 2016.

B. Strang, P. van der Putten, J. N. van Rijn, and F. Hutter. Don't Rule Out Simple Models Prematurely: a Large Scale Benchmark Comparing Linear and Non-linear Classifiers in OpenML. In *Advances in Intelligent Data Analysis XVII*. Springer, 2018.

J. N. van Rijn and F. Hutter. Hyperparameter importance across datasets. *arXiv preprint arXiv:1710.04725*, 2017.

J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren. The online performance estimation framework: heterogeneous ensemble learning for data streams. *Machine Learning*, 107(1):149–167, 2018.