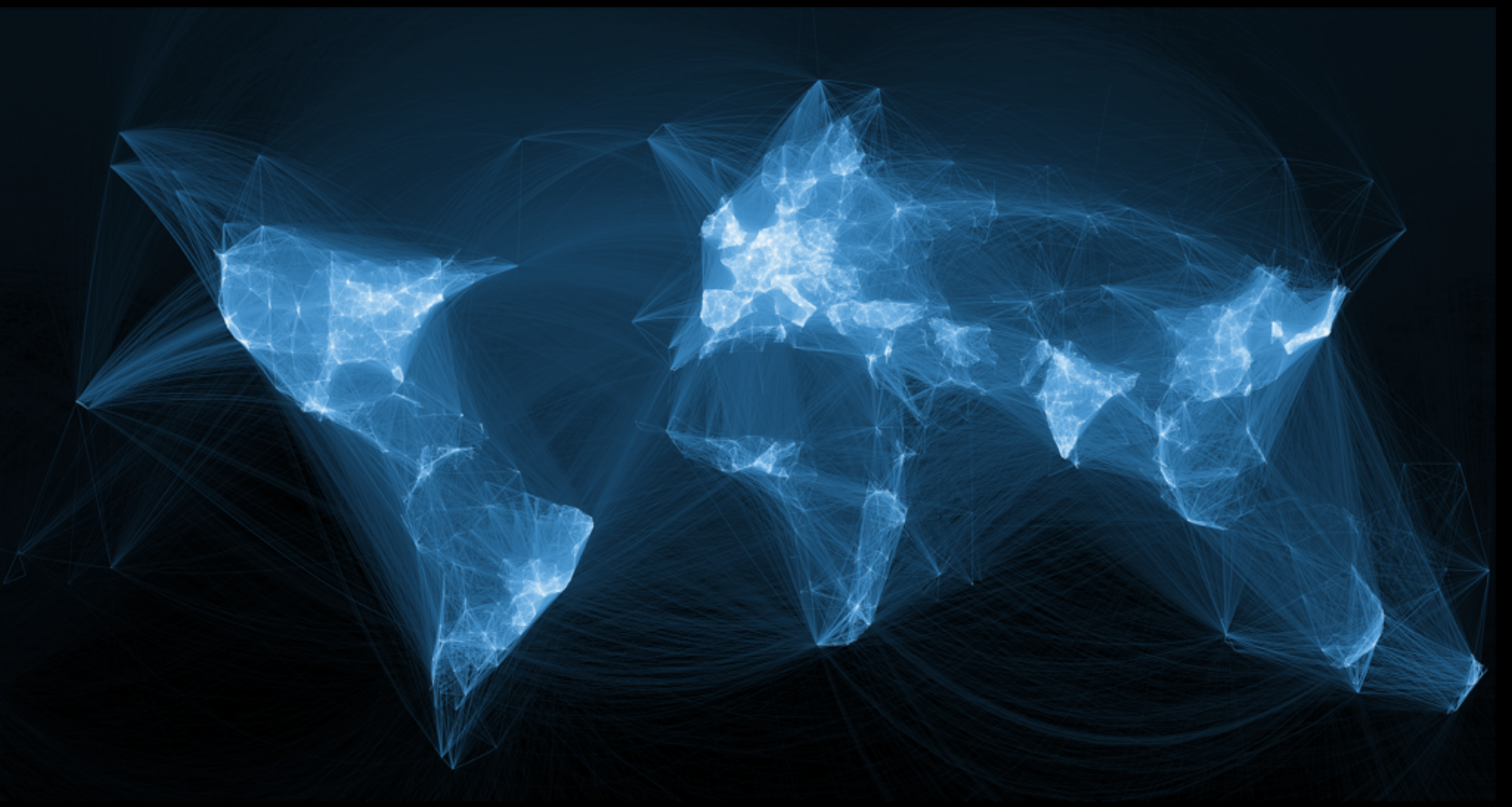
OpenML

NETWORKED MACHINE LEARNING

#OpenML

JOAQUIN VANSCHOREN (TU/E),2015



Research different.

A REALTIME, WORLDWIDE LAB

MUCH OF WHAT MEDICAL RESEARCHERS conclude in their studies is misleading, exaggerated, or flat-out wrong. So why are doctors—to a striking extent—still drawing upon misinformation in their everyday practice? Dr. John Ioannidis has spent his career challenging his peers by exposing their bad science.

LIES, DAMNED LIES, AND MEDICAL SCIENCE

By DAVID H. FREEDMAN

IN 2001, RUMORS were circulating in Greek hospitals that surgery residents, eager to rack up scalpel time, were falsely diagnosing hapless Albanian immigrants with appendicitis. At the University of Ioannina medical school's teaching hospital, a newly minted doctor named Athina Tatsioni was discussing the rumors with colleagues when a professor who had overheard asked her if she'd like to try to prove whether they were true—he seemed to be almost daring her. She accepted the challenge and, with the professor's and other colleagues' help, eventually produced a formal study showing that, for whatever reason, the appendices removed from patients with Albanian names in six Greek hospitals were more than three times as likely to be perfectly healthy as those removed from patients with Greek names. "It was hard to find a journal willing to publish it, but we did," recalls Tatsioni. "I also discovered that I really liked research." Good thing, because the study had actually been a sort of audition. The professor, it turned out, had been putting together a team of exceptionally brash and curious young clinicians and Ph.D.s to join him in tackling an unusual and controversial agenda.

Last spring, I sat in on one of the team's weekly meetings on the medical school's campus, which is planked crazily across a series of sharp hills. The building in which we met, like most at the school, had the look of a barracks and was festooned with political graffiti. But the group convened in a spacious conference room that would have been at home at a Silicon Valley start-up. Sprawled around a large table were Tatsioni and eight other youngish Greek researchers and physicians who, in contrast to the pasty younger staff frequently seen in U.S. hospitals, looked like the casually glamorous cast of a television medical drama. The professor,

Dr. John Ioannidis, photographed in August at Stanford University's Cecil H. Green Library

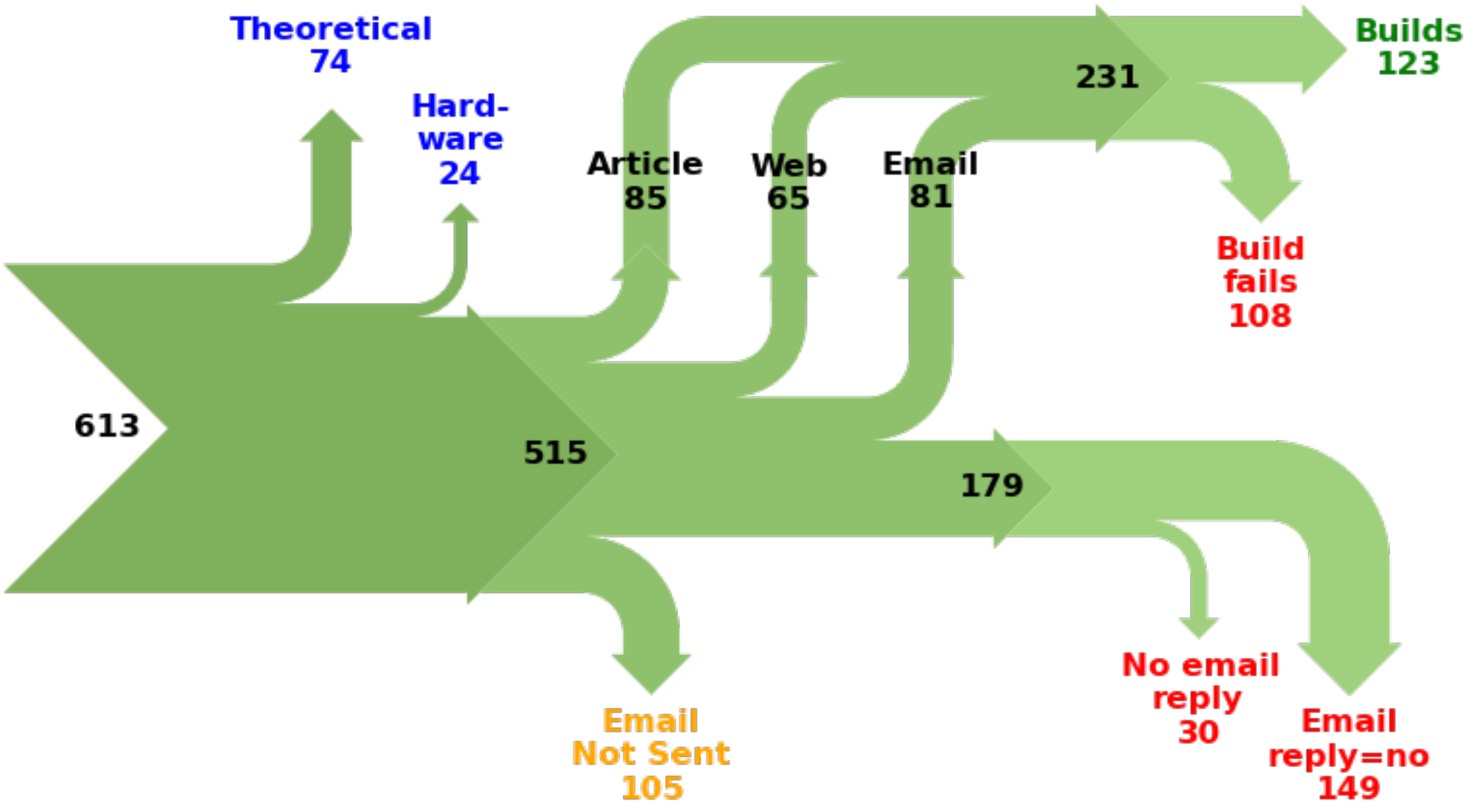
85% research resources are wasted: associations/effects are false, exaggerated, translation into applications is inefficient

<u>Reasons</u>:
Underpowered (small) studies, flexibility in design, lack of collaboration.

<u>Increase credibility</u>:
Large-scale collaboration, replication culture, reproducibility, registration/sharing of data, better statistical methods, better study design, training

Computer science

Reproducibility of published results (ACM literature)



Machine learning (or data science)

Complex code, large-scale data, experiments (impossible to print)

Experiments not shared online: impossible to build on prior work: inhibits deeper analysis

Low reproducibility, generalisability (studies contradict)

What if we could all connect with each other, and with other scientists, to explore and apply machine learning?

Research different.

Polymaths: Solve math problems through massive collaboration (not competition)

Broadcast question, combine many minds to solve it

Solved hard problems in weeks

Many (joint) publications

Research different.

SDSS: Robotic telescope, data publicly online (SkyServer)

Broadcast data, allow many minds to ask the right questions

Thousands of papers, citations

+1 million distinct users
vs. 10.000 astronomers

Next: Synoptic Telescope

Designed serendipity

Broadcasting data (and questions)
fosters spontaneous discoveries

What's hard for one scientist
is easy for another

What's surprising for one scientist
sparks ideas in another

Network effects:
studies become cheaper, faster

Remove friction

Organised body of compatible scientific data (and tools)

Micro-contributions

Easy, organised communication

Measure impact (altmetrics)
Give credit when credit is due

Protect preliminary work

OpenML Search

OpenML beta

Exploring machine learning better, together

1340 data sets

Find or add **data** to analyse

4828 tasks

Download or create scientific **tasks**

1346 flows

Find or add data analysis **flows**

434613 runs

Upload and explore all **results** online.

1 minute intro

Data from
various sources
analysed and
organised online
for easy access

Scientists can **broadcast data,** explaining the challenge that needs to be addressed. OpenML will (for known data formats) **automatically analyze the data**, compute data characteristics, **annotate and index it for easy search**

1 minute intro

Scientific tasks that can be **interpreted by tools**, and **solved collaboratively**

Tasks are **realtime (collaborative) data mining challenges**, allowing anyone to build on previous results. OpenML creates **machine-readable descriptions** so that tools can **automatically download data**, use the correct procedures, and **upload all results online**.

1 minute intro

Tool plugins
for automated
data download,
workflow upload and
experiment logging
and sharing

Flows are implementations of algorithms, workflows, or scripts **solving OpenML tasks**. OpenML keeps track of **flow details and versioning**, **organizes all their results** for easy comparison, even across tools.

1 minute intro

Experiments auto-uploaded, linked to **data, flows** and **authors**, and organised for easy reuse

Runs contain the results that **flows** obtained on specific tasks. Runs are **fully reproducible**, linked to the underlying data, tasks, flows and authors. OpenML **organizes all results online** for **discovery, comparison and reuse**

OpenML.org

Search, add/ delete, share, studies,…

Data

Tasks

Flows

Runs

Java API

R API

Python API

Web API

moa

Open for Innovation KNIME

rapidminer

mlr

caret*, ipred*

SciKit*

…

OpenML

Demo (beta)

covertype

v. 2

Sparse ARFF | Publicly available | Visibility: public | Uploaded 15-08-2014 by aydin demircioglu | Edit

Edit

Author: Jock A. Blackard, Dr. Denis J. Dean, Dr. Charles W. Anderson
Source: LibSVM repository - 2013-11-14
Please cite: For the binarization: R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. Neural Computation, 14(05):1105-1114, 2002.

This is the famous covertype dataset in its binary version, retrieved 2013-11-13 from the libSVM site (called covtype.binary there). Additional to the preprocessing done there (see LibSVM site for details), this dataset was created as follows: -load covertpype dataset, unscaled. -normalize each file columnwise according to the following rules: -If a column only contains one value (constant feature), it will set to zero and thus removed by sparsity. -If a column contains two values (binary

Show more

Properties

0.51	2	55	581012
DefaultAccuracy	NumberOfClasses	NumberOfFeatures	NumberOfInstances

0	54
NumberOfMissingValues	NumberOfNumericFeatures

Show more

Properties

0.47	7	55	110393
DefaultAccuracy	NumberOfClasses	NumberOfFeatures	NumberOfInstances

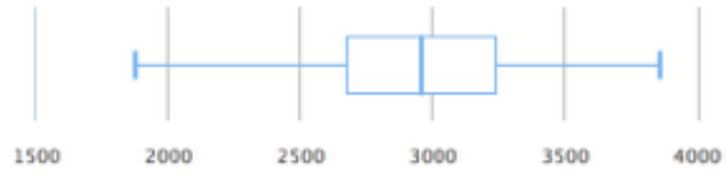
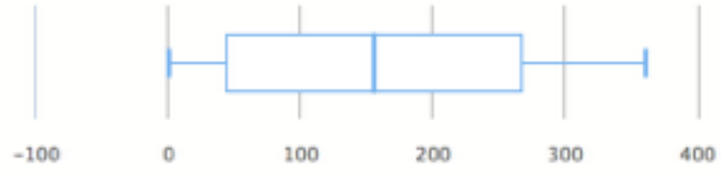
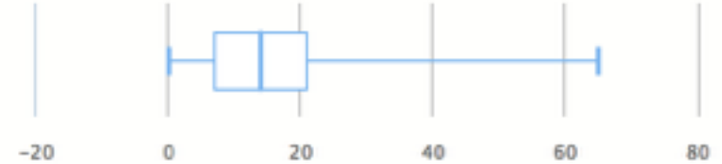
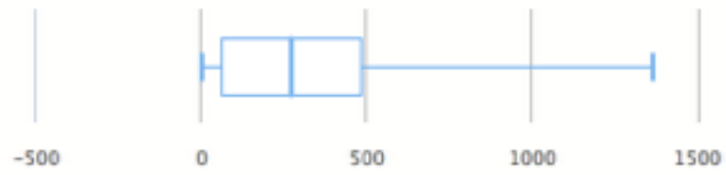
0	14	40
NumberOfMissingValues	NumberOfNumericFeatures	NumberOfSymbolicFeature:

▸ Show more

ClassCount	7
ClassEntropy	1.87
DecisionStumpAUC	0.6
DecisionStumpErrRate	53.12
DecisionStumpKappa	0.06
Dimensionality	0
EquivalentNumberOfAtts	148.97
HoeffdingAdwin.changes	17
HoeffdingAdwin.warnings	0

Open source library to calculate over 75 meta-features:
simple, statistical, information-theoretic, landmarkers, streaming,…

Features

class **(target)**	nominal	values, 0 missing	
elevation	numeric	values, 0 missing	
aspect	numeric	values, 0 missing	
slope	numeric	values, 0 missing	
horizontal_distance_to_hydrology	numeric	values, 0 missing	
Vertical_Distance_To_Hydrology	numeric	values, 0 missing	
soil_type_29	nominal	values, 0 missing	

Visualisations to inspect data

Search API Plugins

Create new task

Choose Task Type

Task types

Supervised Classification

Overview

Supervised Classification In supervised classification, you are given an input dataset in which instances are labeled with a certain class. The goal is to build a model that predicts the class for future unlabeled instances. The model is evaluated using a train-test procedure, e.g. cross-validation.

To make results by different users comparable, you are given the exact train-test folds to be used, and you need to return at least the predictions generated by your model for each of the test instances. OpenML will use these predictions to calculate a range of evaluation measures on the server.

Supervised Classification

Dataset(s)

(*) include all datasets

Target Feature

class

Estimation Procedure

10 times 10-fold Crossvalidation

Evaluation Measures

Submit

Create task: choose task type (inputs, outputs)

Task 59

JSON XML

Supervised Classification

In supervised classification, you are given an input dataset in which instances are labeled with a certain class. The goal is to build a model that predicts the class for future unlabeled instances. The model is evaluated using a train-test procedure, e.g. cross-validation.

To make results by different users comparable, you are given the exact train-test folds to be used, and you need to return at least the predictions generated by your model for each of the test instances. OpenML will use these predictions to calculate a range of evaluation measures on the server.

You can also upload your own evaluation measures, provided that the code for doing so is available from the implementation used. For extremely large datasets, it may be infeasible to upload all predictions. In those cases, you need to compute and provide the evaluations yourself.

Optionally, you can upload the model trained on all the input data. There is no restriction on the file format, but please use a well-known format or PMML.

Given inputs

estimation_procedure	10-fold Crossvalidation	Estimation Procedure (required)
evaluation_measures		String (optional)
source_data	iris (1)	Dataset (required)
target_feature	class	String (required)

<oml:task xmlns:oml="http://openml.org/openml">
  <oml:task_id>59</oml:task_id>
  <oml:task_type>Supervised Classification</oml:task_type>
  <oml:input name="source_data">
    <oml:data_set>
      <oml:data_set_id>61</oml:data_set_id>
      <oml:target_feature>class</oml:target_feature>
    </oml:data_set>
  </oml:input>
  <oml:input name="estimation_procedure">
    <oml:estimation_procedure>
      <oml:type>crossvalidation</oml:type>
      <oml:data_splits_url>
        http://openml.org/api_splits/get/59/Task_59_splits.arff
      </oml:data_splits_url>
      <oml:parameter name="number_repeats">1</oml:parameter>
      <oml:parameter name="number_folds">10</oml:parameter>
      <oml:parameter name="percentage"/>
      <oml:parameter name="stratified_sampling">true</oml:parameter>
    </oml:estimation_procedure>
  </oml:input>
  <oml:input name="evaluation_measures">
    <oml:evaluation_measures>
      <oml:evaluation_measure/>
    </oml:evaluation_measures>
  </oml:input>
  <oml:output name="predictions">
    <oml:predictions>
      <oml:format>ARFF</oml:format>
      <oml:feature name="repeat" type="integer"/>
      <oml:feature name="fold" type="integer"/>
      <oml:feature name="row_id" type="integer"/>
      <oml:feature name="confidence.classname" type="numeric"/>
      <oml:feature name="prediction" type="string"/>
    </oml:predictions>
  </oml:output>
</oml:task>

Machine-interpretable tasks

weka.RandomForest

v. 1

Leo Breiman (2001). Random Forests. Machine Learning. 45(1):5-32.

Attribution

Author(s)	
Contributor(s)	
Uploader	
Licence	
Citation	Show
Dependencies	Weka_3.7.10

Parameters

D	If set, classifier is run in debug mode and may output additional info to the console	
I	Number of trees to build.	10
K	Number of features to consider (<1=int(logM+1)).	0
S	Seed for random number generator. (default 1)	1
depth	The maximum depth of the trees, 0 for unlimited. (default 0)	
num-slots	Number of execution slots. (default 1 - i.e. no parallelism)	1
print	Print the individual trees in the output	

Results (per task type)

Flows, with parameter details

Compare flows
Draw learning curves
Advanced queries
SQL editor

Table
Scatterplot
Line plot

Run 25271

JSON XML

Task

Task	Task 77 (Learning Curve)
Input data	mfeat-morphological (1)

Run Details

Uploader	Jan van Rijn
Start time	2014-08-17 04:43:48
Status	OK

Flow

weka.J48(3)	Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.
365_C	0.25
365_M	2

A run is a successful execution of a flow on a task

Instance-level predictions, model binaries

Results

- description
- predictions
- model_serialized
- model_readable

Scores computed on server

Evaluations

area_under_roc_curve	0.9001	[0.993311,0.958499,0.854493,0.807607,0.83917,0.866759,0.90
build_cpu_time	0.0074	
build_memory	4608297777.0109	
confusion_matrix		[[21685,0,218,3,0,91,0,0,0,3],[0,20042,118,73,861,8,60,790,0,48], [162,139,14725,1993,559,3161,172,1024,2,63],[3,228,2010,10177 [0,1230,612,3895,12886,416,74,2868,1,18],[93,5,5452,3376,526,1 [0,132,126,94,16,18,14774,26,77,6737],[3,457,975,1410,2820,251 [0,1,18,0,0,0,610,0,21109,262],[4,143,376,123,27,139,14658,32,16
f_measure	0.6775	[0.986803,0.903261,0.631568,0.471769,0.596367,0.571877,0.5
kappa	0.6466	
kb_relative_information_score	155016.6079	
mean_absolute_error	0.0708	
mean_prior_absolute_error	0.1800	

```
@relation openml_task_77_predictions

@attribute repeat numeric
@attribute fold numeric
@attribute sample numeric
@attribute row_id numeric
@attribute confidence.1 numeric
@attribute confidence.2 numeric
@attribute confidence.3 numeric
@attribute confidence.4 numeric
@attribute confidence.5 numeric
@attribute confidence.6 numeric
@attribute confidence.7 numeric
@attribute confidence.8 numeric
@attribute confidence.9 numeric
@attribute confidence.10 numeric
@attribute prediction {1,2,3,4,5,6,7,8,9,10}
@attribute correct {1,2,3,4,5,6,7,8,9,10}

@data
0,0,0,9,1,0,0,0,0,0,0,0,0,0,1,1
0,0,0,388,0,1,0,0,0,0,0,0,0,0,2,2
0,0,0,474,0,0,1,0,0,0,0,0,0,0,3,3
0,0,0,750,0,0,0,1,0,0,0,0,0,0,4,4
0,0,0,903,0,0,0,0,0.75,0,0,0.25,0,0,5,5
0,0,0,1023,0,0,1,0,0,0,0,0,0,0,3,6
0,0,0,1252,0,0,0.111111,0,0,0,0.222222,0,0,0.666667,10,7
0,0,0,1483,0,0,0,0,0,0,0,1,0,0,8,8
0,0,0,1682,0,0,0,0,0,0,0,0,1,0,9,9
0,0,0,1870,0,0,0.111111,0,0,0,0.222222,0,0,0.666667,10,10
0,0,0,3,1,0,0,0,0,0,0,0,0,0,1,1
0,0,0,398,0,1,0,0,0,0,0,0,0,0,2,2
0,0,0,468,0,0,0.333333,0,0,0.666667,0,0,0,0,6,3
0,0,0,705,0,0,0.333333,0,0,0.666667,0,0,0,0,6,4
0,0,0,990,0,0,0,0,0.75,0,0,0.25,0,0,5,5
0,0,0,1131,0,0,0,0,0,1,0,0,0,0,6,6
0,0,0,1355,0,0,0.111111,0,0,0,0.222222,0,0,0.666667,10,7
0,0,0,1443,0,0,0,0,0,0,0,1,0,0,8,8
0,0,0,1663,0,0,0,0,0,0,0,0,1,0,9,9
0,0,0,1996,0,0,0.111111,0,0,0,0.222222,0,0,0.666667,10,10
0,0,0,27,1,0,0,0,0,0,0,0,0,0,1,1
0,0,0,380,0,1,0,0,0,0,0,0,0,0,2,2
0,0,0,487,0,0,1,0,0,0,0,0,0,0,3,3
0,0,0,714,0,0,0,1,0,0,0,0,0,0,4,4
0,0,0,912,0,0,0,0,0.75,0,0,0.25,0,0,5,5
0,0,0,1192,0,0,0,1,0,0,0,0,0,0,4,6
0,0,0,1375,0,0,0,0,0,0,1,0,0,0,7,7
0,0,0,1477,0,1,0,0,0,0,0,0,0,0,2,8
0,0,0,1648,0,0,0,0,0,0,0,0,1,0,9,9
0,0,0,1903,0,0,0.111111,0,0,0,0.222222,0,0,0.666667,10,10
0,0,0,164,0,0,1,0,0,0,0,0,0,0,3,1
```

```
J48 pruned tree
------------------

att1 <= 0
|   att5 <= 1.548576: 2 (202.0/15.0)
|   att5 > 1.548576
|   |   att6 <= 8574.682952
|   |   |   att6 <= 5804.513685
|   |   |   |   att5 <= 1.609052
|   |   |   |   |   att2 <= 2
|   |   |   |   |   |   att4 <= 144.472861
|   |   |   |   |   |   |   att6 <= 4528.578862: 8 (3.0/1.0)
|   |   |   |   |   |   |   att6 > 4528.578862: 2 (3.0/1.0)
|   |   |   |   |   |   att4 > 144.472861
|   |   |   |   |   |   |   att6 <= 5120.638273: 8 (7.0)
|   |   |   |   |   |   |   att6 > 5120.638273: 5 (3.0/1.0)
|   |   |   |   |   att2 > 2
|   |   |   |   |   |   att5 <= 1.589619: 5 (3.0)
|   |   |   |   |   |   att5 > 1.589619: 2 (2.0)
|   |   |   |   att5 > 1.609052: 8 (104.0/6.0)
|   |   |   att6 > 5804.513685
|   |   |   |   att2 <= 2
|   |   |   |   |   att4 <= 163.296861
|   |   |   |   |   |   att5 <= 1.773592
|   |   |   |   |   |   |   att6 <= 6493.591741
|   |   |   |   |   |   |   |   att4 <= 155.794861: 5 (22.0/8.0)
|   |   |   |   |   |   |   |   att4 > 155.794861
|   |   |   |   |   |   |   |   |   att5 <= 1.643385: 4 (4.0/1.0)
|   |   |   |   |   |   |   |   |   att5 > 1.643385
|   |   |   |   |   |   |   |   |   |   att5 <= 1.743657: 8 (7.0/1.0)
|   |   |   |   |   |   |   |   |   |   att5 > 1.743657: 4 (2.0/1.0)
|   |   |   |   |   |   |   att6 > 6493.591741
|   |   |   |   |   |   |   |   att4 <= 161.344861: 5 (16.0/2.0)
|   |   |   |   |   |   |   |   att4 > 161.344861
|   |   |   |   |   |   |   |   |   att6 <= 7329.973134: 3 (6.0/2.0)
|   |   |   |   |   |   |   |   |   att6 > 7329.973134: 5 (2.0)
|   |   |   |   |   |   att5 > 1.773592
|   |   |   |   |   |   |   att6 <= 7329.973134: 8 (21.0/5.0)
|   |   |   |   |   |   |   att6 > 7329.973134: 5 (5.0/2.0)
|   |   |   |   |   att4 > 163.296861
|   |   |   |   |   |   att6 <= 7908.003397
|   |   |   |   |   |   |   att5 <= 1.703938
|   |   |   |   |   |   |   |   att5 <= 1.639492: 4 (2.0/1.0)
|   |   |   |   |   |   |   |   att5 > 1.639492: 3 (2.0)
|   |   |   |   |   |   |   att5 > 1.703938: 8 (50.0/6.0)
|   |   |   |   |   |   att6 > 7908.003397
|   |   |   |   |   |   |   att4 <= 172.932861
|   |   |   |   |   |   |   |   att6 <= 8306.160447
|   |   |   |   |   |   |   |   |   att6 <= 8105.99096: 6 (3.0/1.0)
|   |   |   |   |   |   |   |   |   att6 > 8105.99096: 4 (5.0)
|   |   |   |   |   |   |   |   att6 > 8306.160447
```

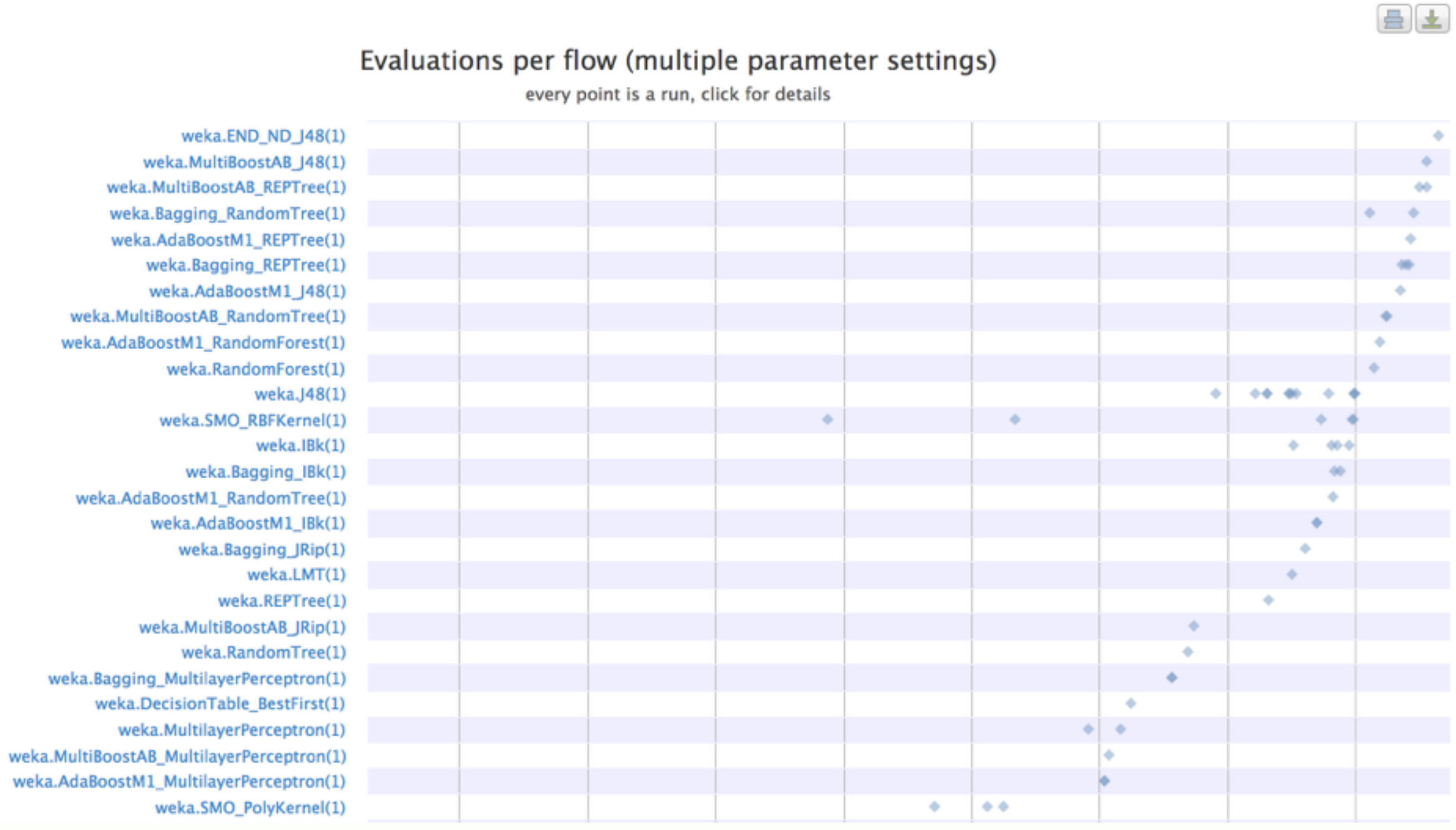
Overview of results per task

Results (per task)

Task 2072 (Supervised Classification) view task details

Performance evaluation

Evaluation measure: predictive accuracy

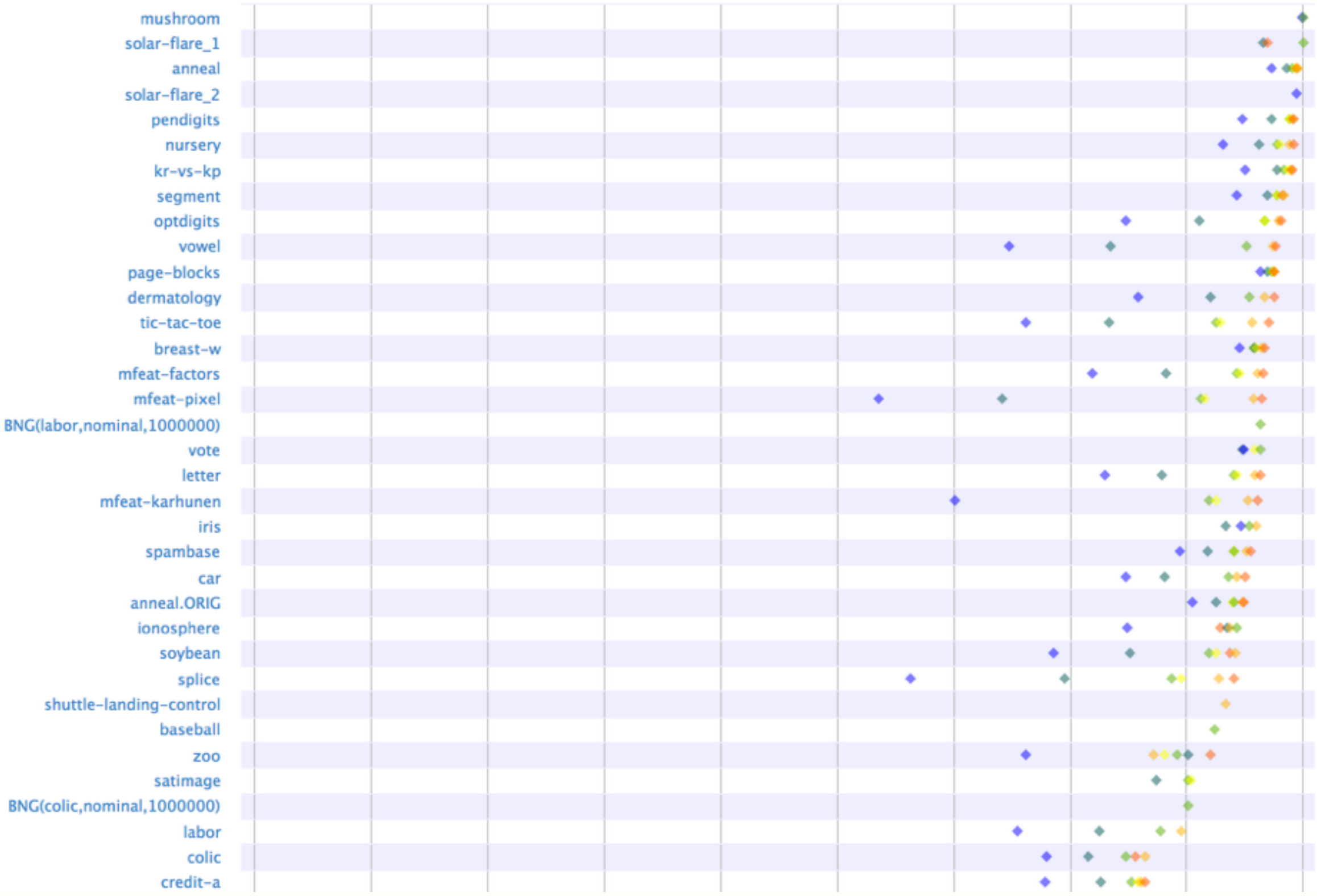


Performance evaluation

Overview of results per flow

Evaluation measure: predictive accuracy

Parameter: I



- All 27448
- Run 25249
- Task 1287
- Flow 381
- **Data 240**
- Measure 171
- User 116
- Task type 4

Show all

Filter results

Number of instances
1000..10000

Number of features
>5

Number of missing values
100..200, 500, >10000

Number of classes
2

Results for *

Found 4 results (0.002 seconds)

Sort: **best match**

mushroom (1)
1. Title: Mushroom Database 2. Sources: (a) Mushroom records drawn from The Audubon Society Field...
306 runs - 8124 instances - 23 features - 2 classes - 2480 missing values

credit-g (1)
Description of the German credit dataset. 1. Title: German Credit data 2. Source Information...
299 runs - 1000 instances - 21 features - 2 classes - 0 missing values

spambase (1)
1. Title: SPAM E-mail Database 2. Sources: (a) Creators: Mark Hopkins, Erik Reeber, George Forman,...
309 runs - 4601 instances - 58 features - 2 classes - 0 missing values

kr-vs-kp (1)
1. Title: Chess End-Game -- King+Rook versus King+Pawn on a7 (usually abbreviated KRKPA7). The pawn...
297 runs - 3196 instances - 37 features - 2 classes - 0 missing values

« 1 »

Search everything

Coming very soon

Projects
- View on specific study (notebook)
- Datasets, flows, runs, results + description
- Easily include (build on) data of others
- Online counterpart of paper
- Linked to paper plus backlink in paper

Circles of trust
- Add scientists in teams (circles)
- Share resources, results within team only
- Make public at any time (e.g. after publication)

Web API

OpenML offers a RESTful Web API for uploading and downloading machine learning resources. Below is a list of common use cases.

Using REST services

REST services can be called using simple HTTP GET or POST actions.

The REST Endpoint URL is

```
http://www.openml.org/api/
```

For instance, to request the `openml.data.description` service, invoke like this (e.g., in your browser):

```
http://www.openml.org/api/?f=openml.data.description&data_id=1
```

From your command-line, you can use curl:

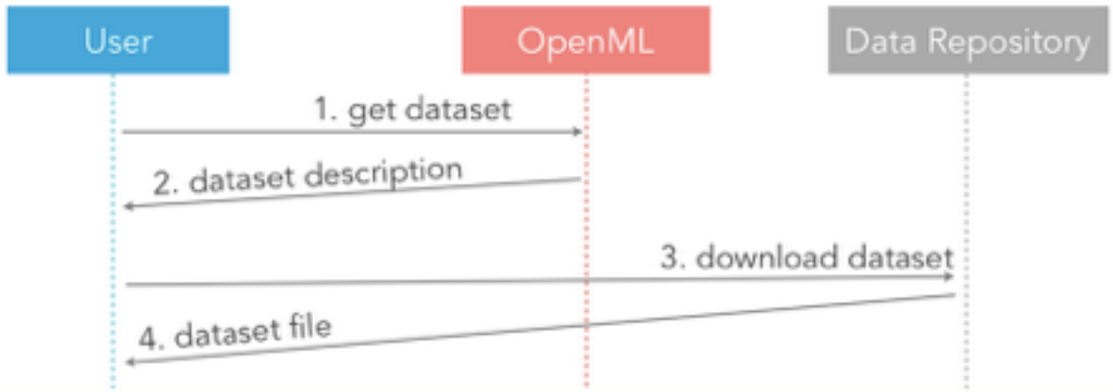
```
curl -XGET 'http://www.openml.org/api/?f=openml.data.description&data_id=1'
```

Responses are always in XML format, also when an error is returned. Error messages will look like this:

```
<oml:error xmlns:oml="http://openml.org/error">
  <oml:code>100</oml:code>
  <oml:message>Please invoke legal function</oml:message>
  <oml:additional_information>Additional information, not always available. </oml:
additional_information>
</oml:error>
```

All services and the corresponding error messages are listed below.

Download a dataset

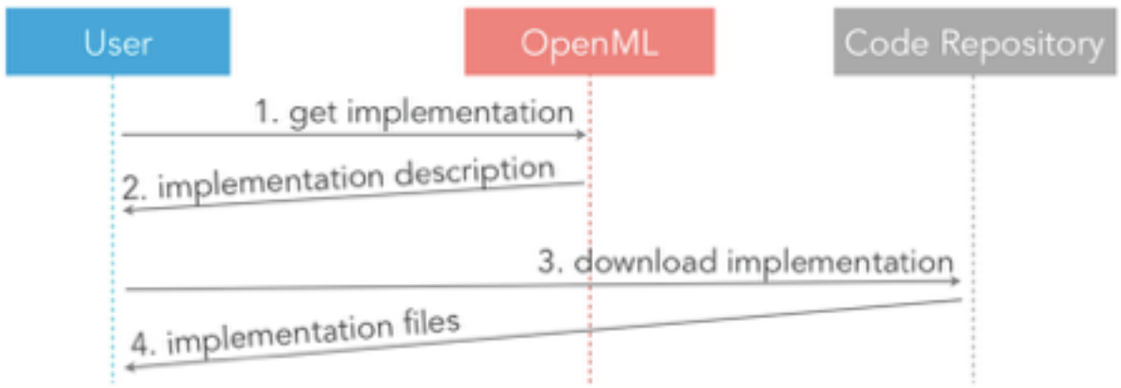


1. User asks for a dataset using the openml.data.description service and a `dataset id`. The `dataset id` is typically part of a task, or returned when searching for datasets.
2. OpenML returns a description of the dataset as an XML file. Try it now
3. The dataset description contains the URL where the dataset can be downloaded. The user calls that URL to download the dataset.
4. The dataset is returned by the server hosting the dataset. This can be OpenML, but also any other data repository. Try it now

Services:

- openml.data.description

Download an implementation



Java API

The Java API allows you connect to OpenML from Java applications.

Download

Stable releases of the Java API are available from Maven central. Or, you can check out the developer version from GitHub. Include the jar file in your projects as usual, or install via Maven. You can also separately download all dependencies and a fat jar with all dependencies included.

Quick Start

Create an `OpenmlConnector` instance with your username and password. This will create a client with all OpenML functionalities.

```
OpenmlConnector client = new OpenmlConnector("username", "password");
```

All functions are described in the Java Docs, and they mirror the functions from the Web API functions described below. For instance, the API function `openml.data.description` has an equivalent Java function `openmlDataDescription(String data_id)`.

Downloading

To download data, flows, tasks, runs, etc. you need the unique **id** of that resource. The id is shown on each item's webpage and in the corresponding url. For instance, let's download Data set 1. The following returns a DataSetDescription object that contains all information about that data set.

```
DataSetDescription data = client.openmlDataDescription(1);
```

R API

The R package openML allows you to connect to the OpenML server from R scrips. Users can download and upload files, run their implementations on specific tasks, get predictions in the correct form, make SQL queries, etc. directly via R commands.

Download

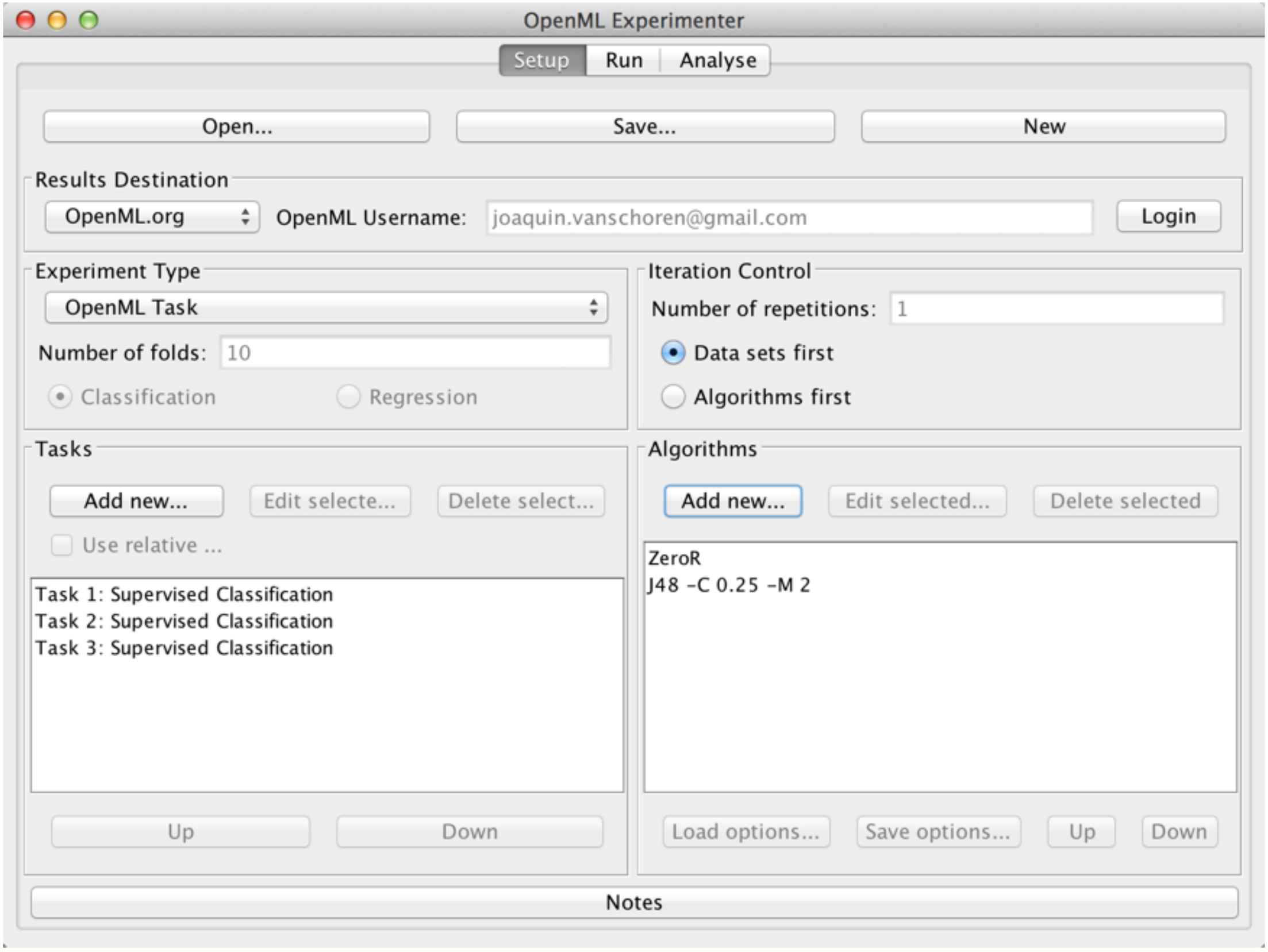
The openML package can be downloaded from GitHub.

Quick Start

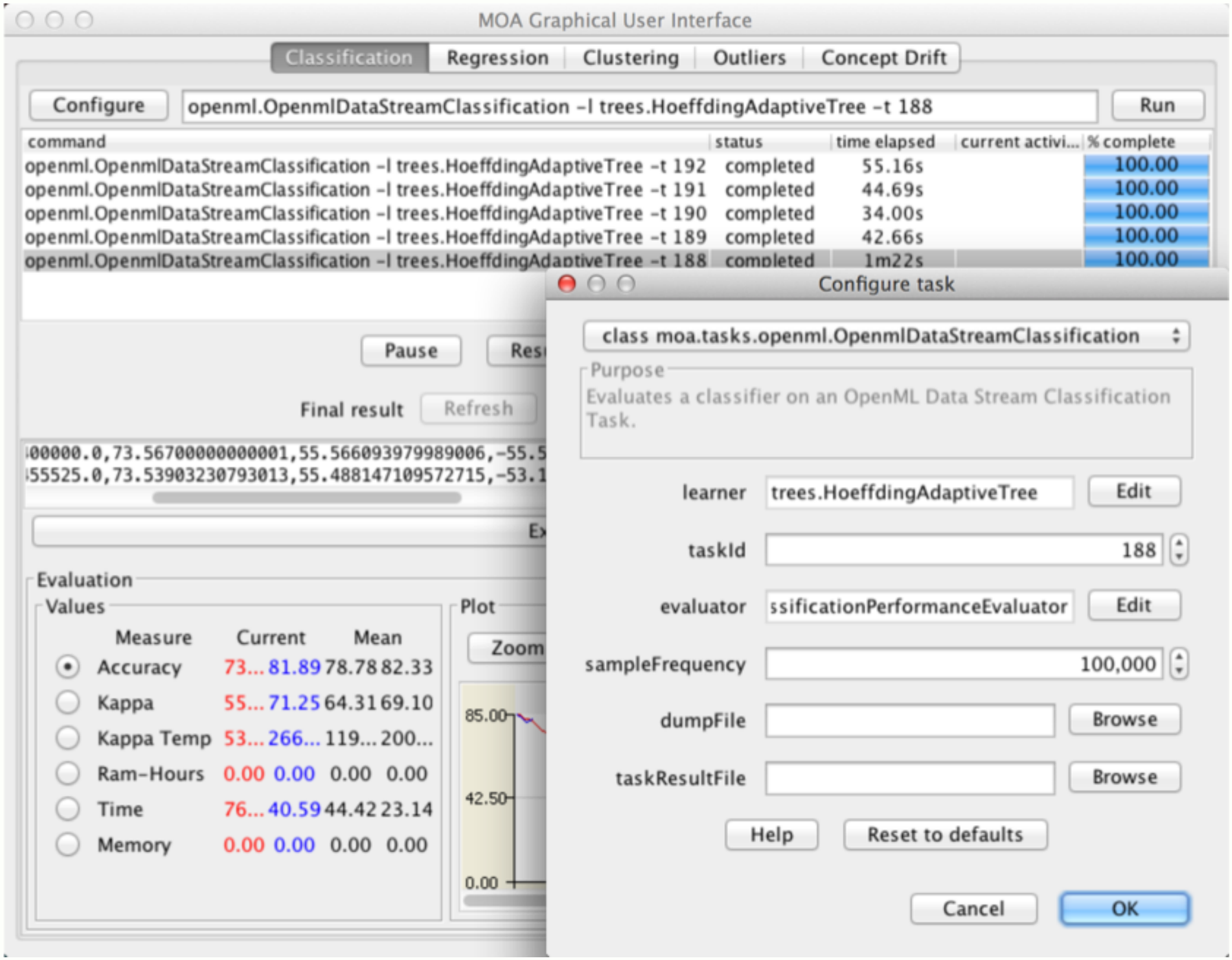
In this tutorial, we will show you the most important functions of this package and give you examples of standard use cases.

```
# Download OpenML task to disk and mem
task <- downloadOpenMLTask(id = 4)

# Let's use a classification tree
learner <- makeLearner("classif.J48")

# Crossvalidate the tree on the task
predictions <- runTask(task, learner)

# Authenticate so we can upload stuff
hash <- authenticateUser(email = "bernd_bischl@gmx.net", passwor
# Upload our predictions
run.ul <- uploadOpenMLRun(task = task, mlr.lrn = learner,
  predictions = predictions, session.hash = hash)
```

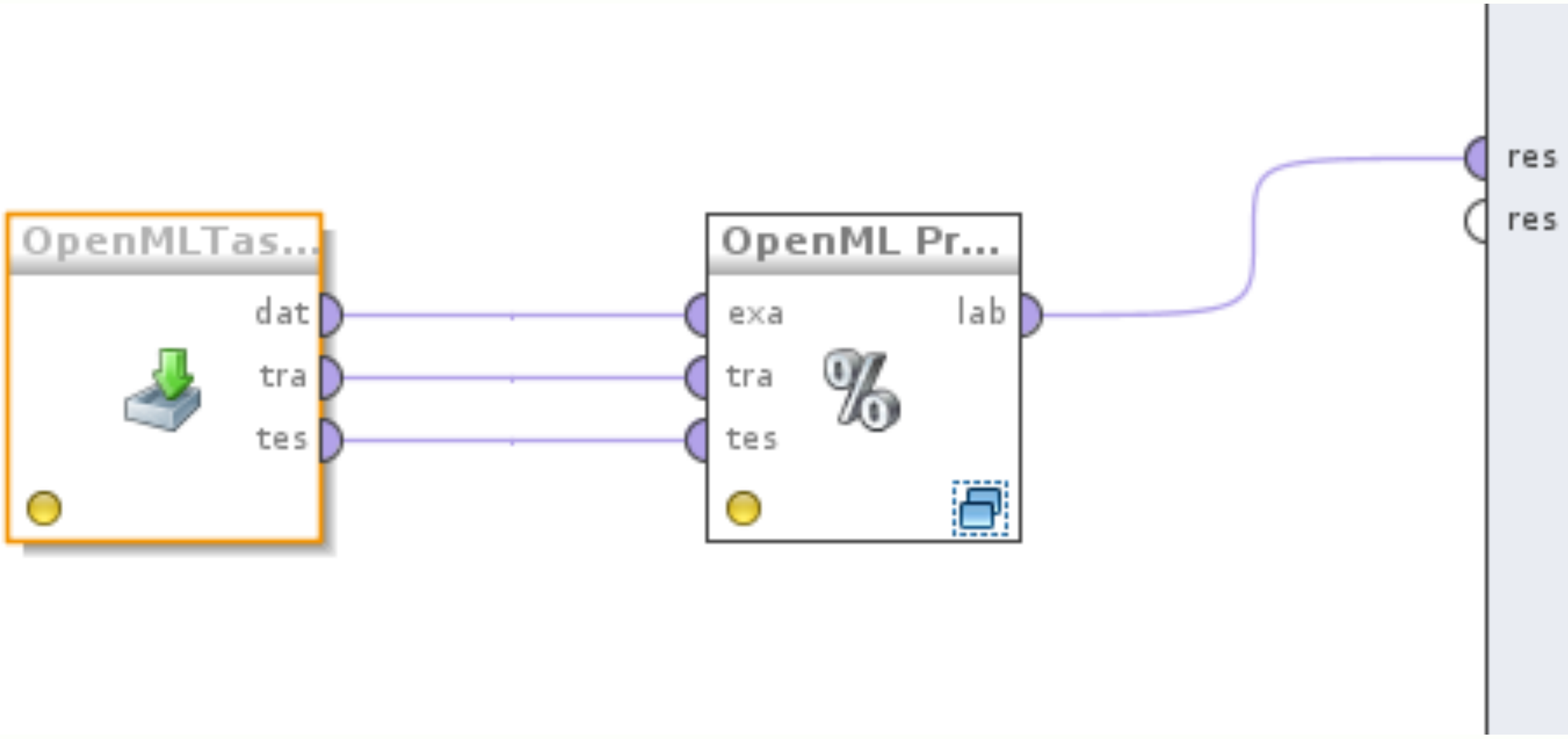
Plugins: WEKA



Plugins: MOA

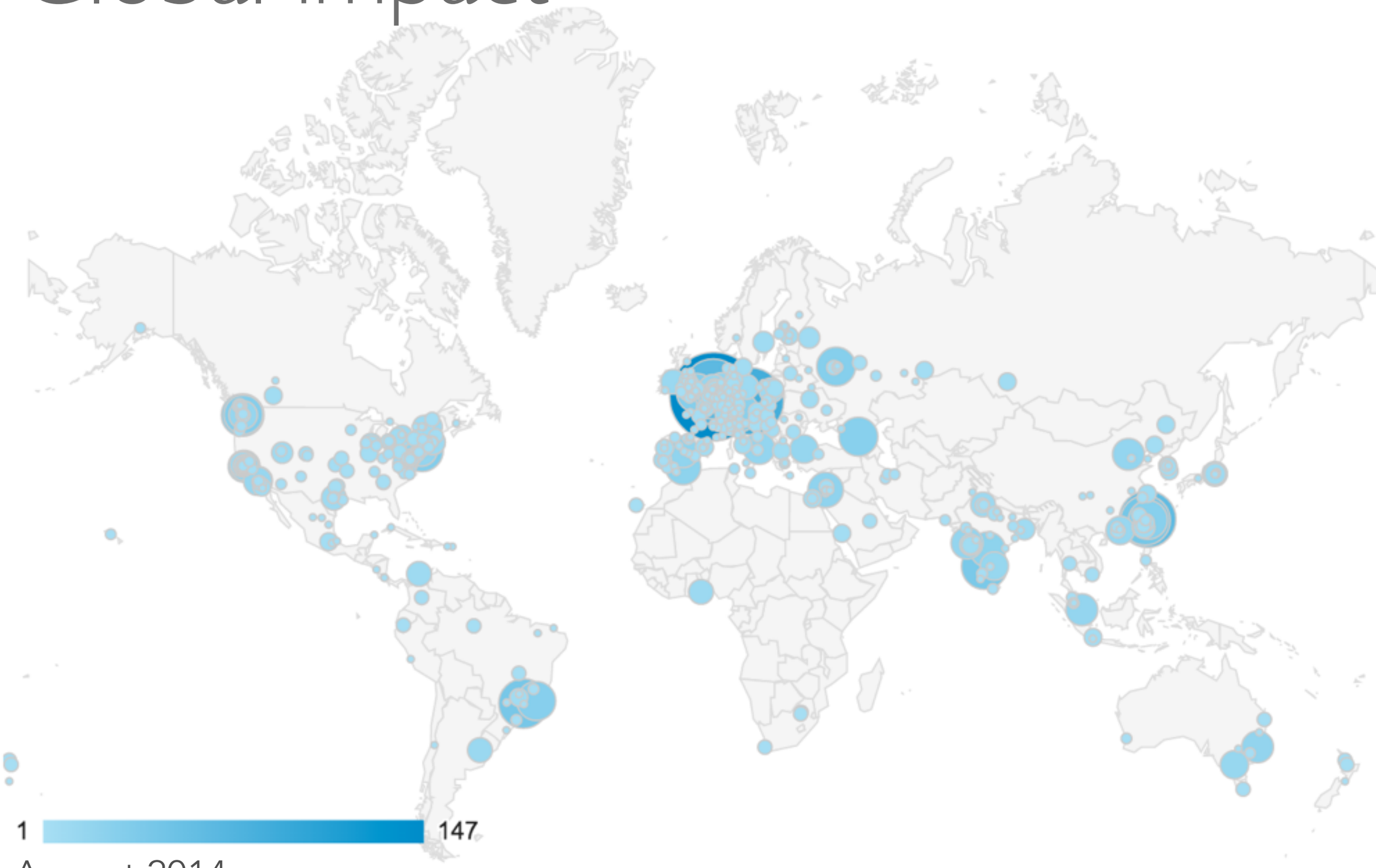

MOA Graphical User Interface
Classification
Regression
Clustering
Outliers
Concept Drift
Configure
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 188
Run
command
status
time elapsed
current activi...
% complete
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 192 completed 55.16s 100.00
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 191 completed 44.69s 100.00
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 190 completed 34.00s 100.00
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 189 completed 42.66s 100.00
openml.OpenmlDataStreamClassification -l trees.HoeffdingAdaptiveTree -t 188 completed 1m22s 100.00
Pause
Final result
Refresh
Evaluation
Values
Measure Current Mean
Accuracy 73... 81.89 78.78 82.33
Kappa 55... 71.25 64.31 69.10
Kappa Temp 53... 266... 119... 200...
Ram-Hours 0.00 0.00 0.00 0.00
Time 76... 40.59 44.42 23.14
Memory 0.00 0.00 0.00 0.00
Plot
Zoom
85.00
42.50
0.00
Configure task
class moa.tasks.openml.OpenmlDataStreamClassification
Purpose
Evaluates a classifier on an OpenML Data Stream Classification Task.
learner
trees.HoeffdingAdaptiveTree
Edit
taskId
188
evaluator
ssificationPerformanceEvaluator
Edit
sampleFrequency
100,000
dumpFile
Browse
taskResultFile
Browse
Help
Reset to defaults
Cancel
OK

Plugins: RapidMiner



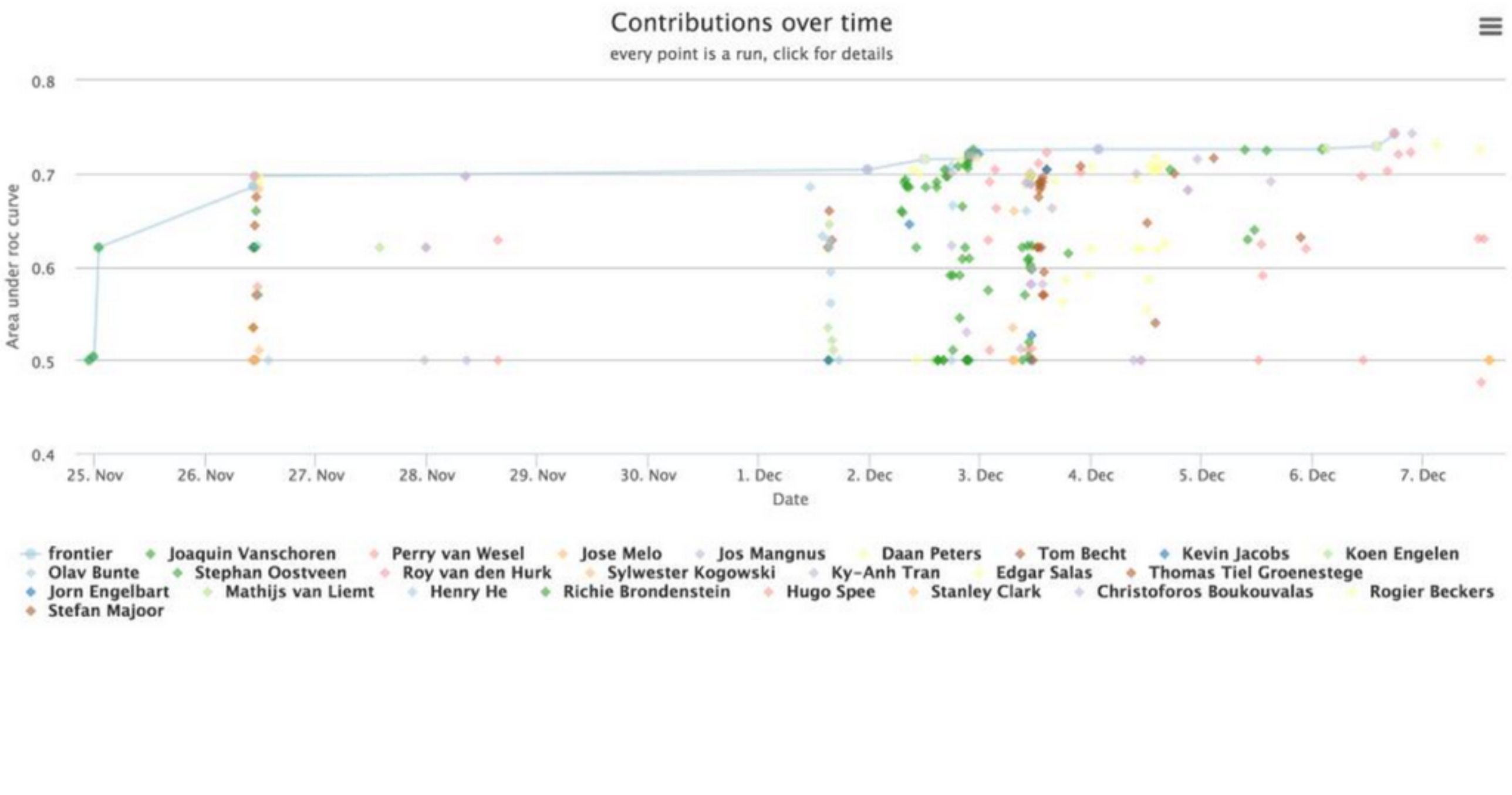
1. OPERATOR TO DOWNLOAD TASK (TASK TYPE SPECIFIC)
2. SUBWORKFLOW THAT SOLVES THE TASK, GENERATES RESULTS
3. OPERATOR FOR UPLOADING RESULTS

Global impact

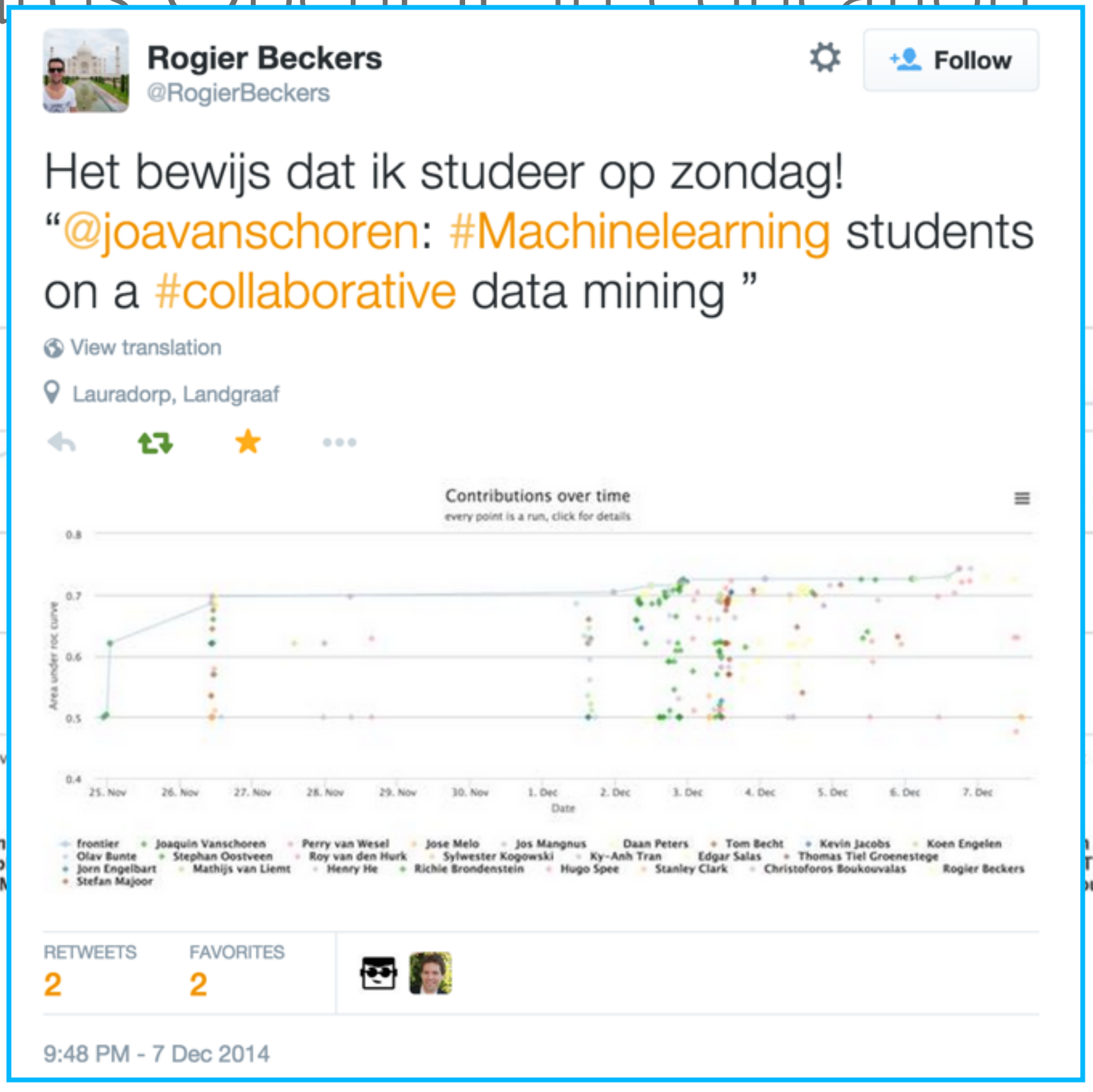


August 2014

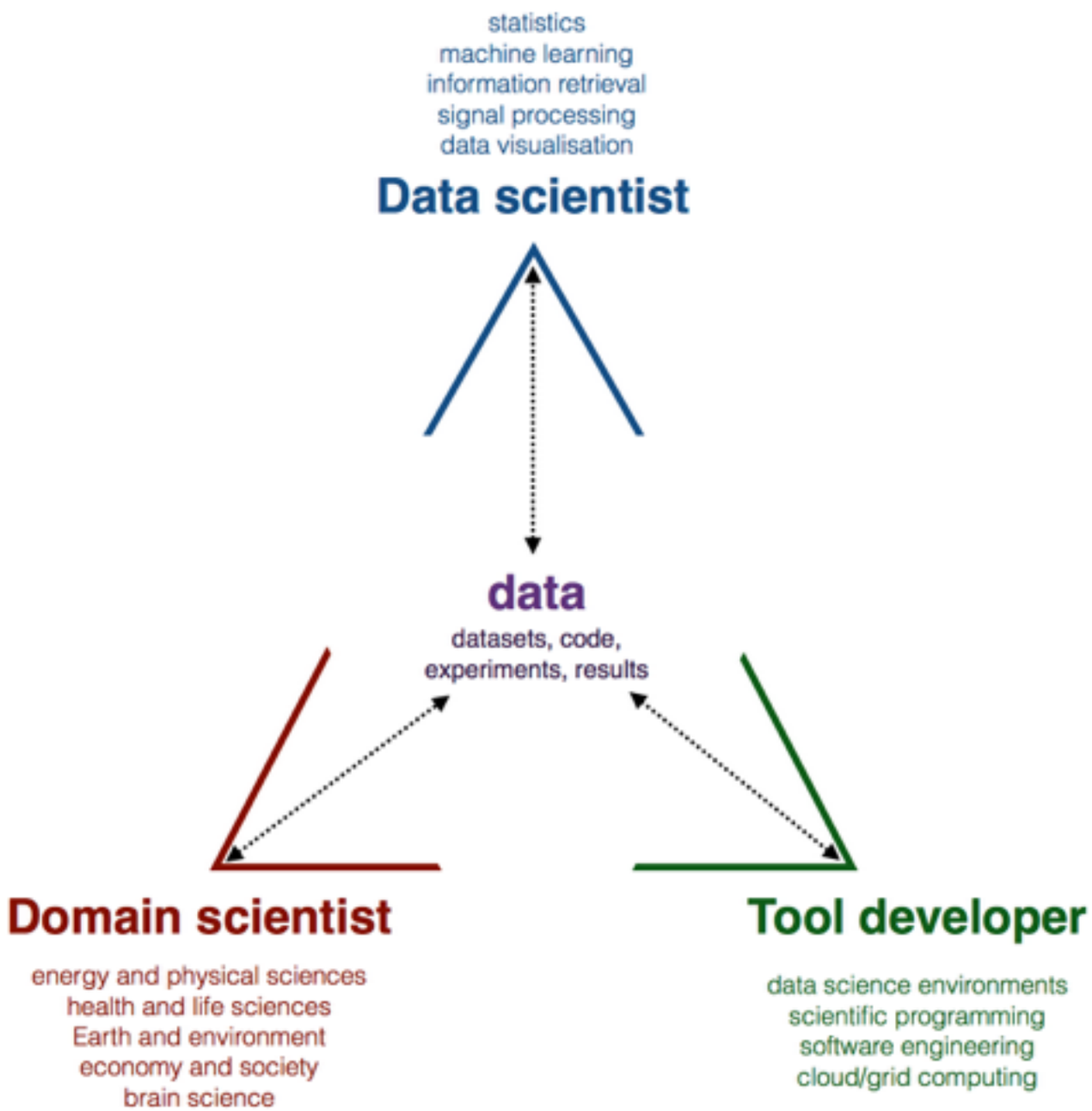
Towards OpenML in education



Towards OpenML in education



Towards a Data Science Collaboratory

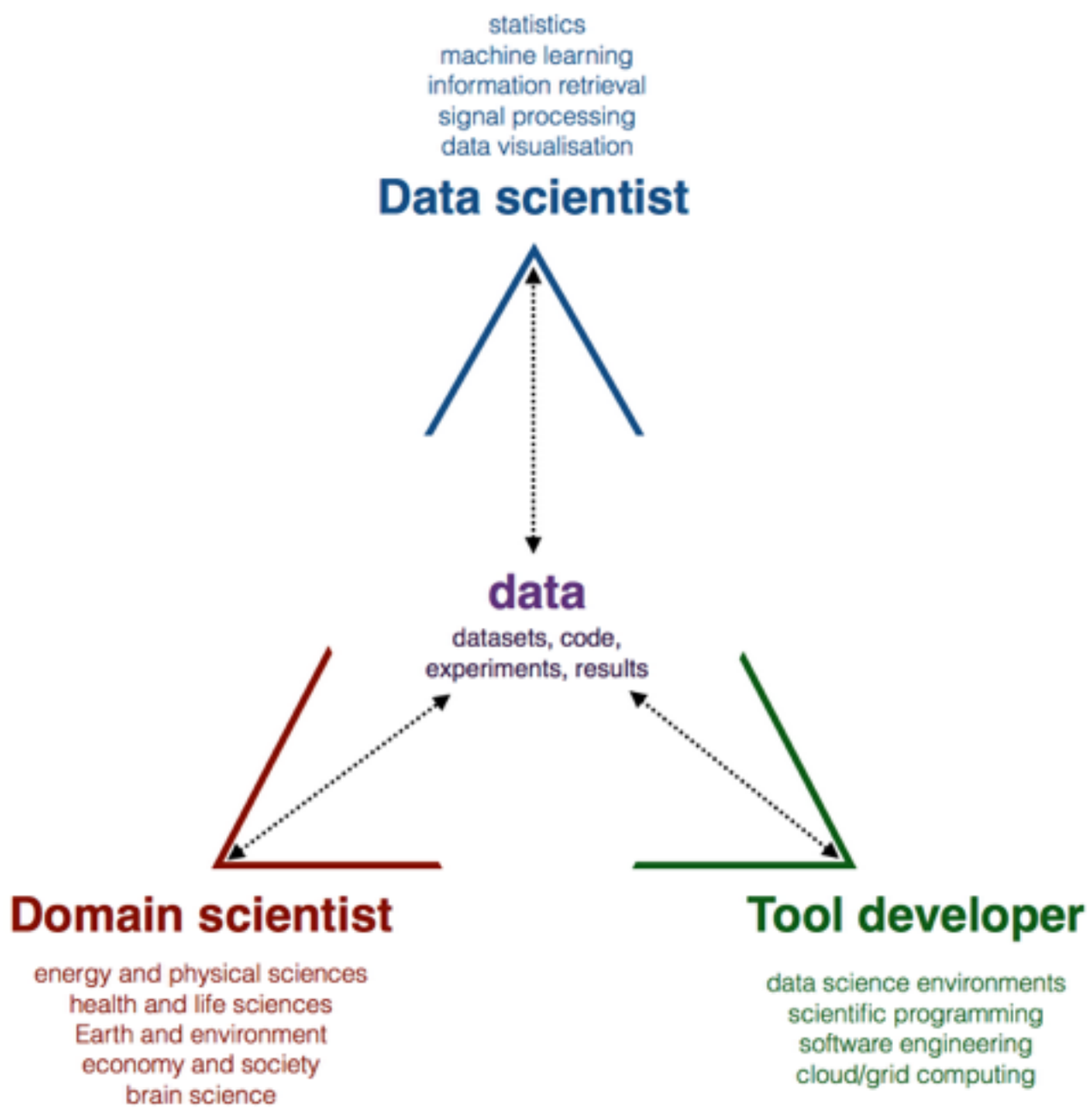


Gap between data scientists and domain scientists: doubts about latest/best techniques, don't speak the language, small (biased) collaborations

Gap between academia and industry: small companies delimitated by lack of access to data/expertise

Fragmented access to tools/ data: don't know how to access scientific databases, latest DS techniques. Toy data/algorithms.

Towards a Data Science Collaboratory

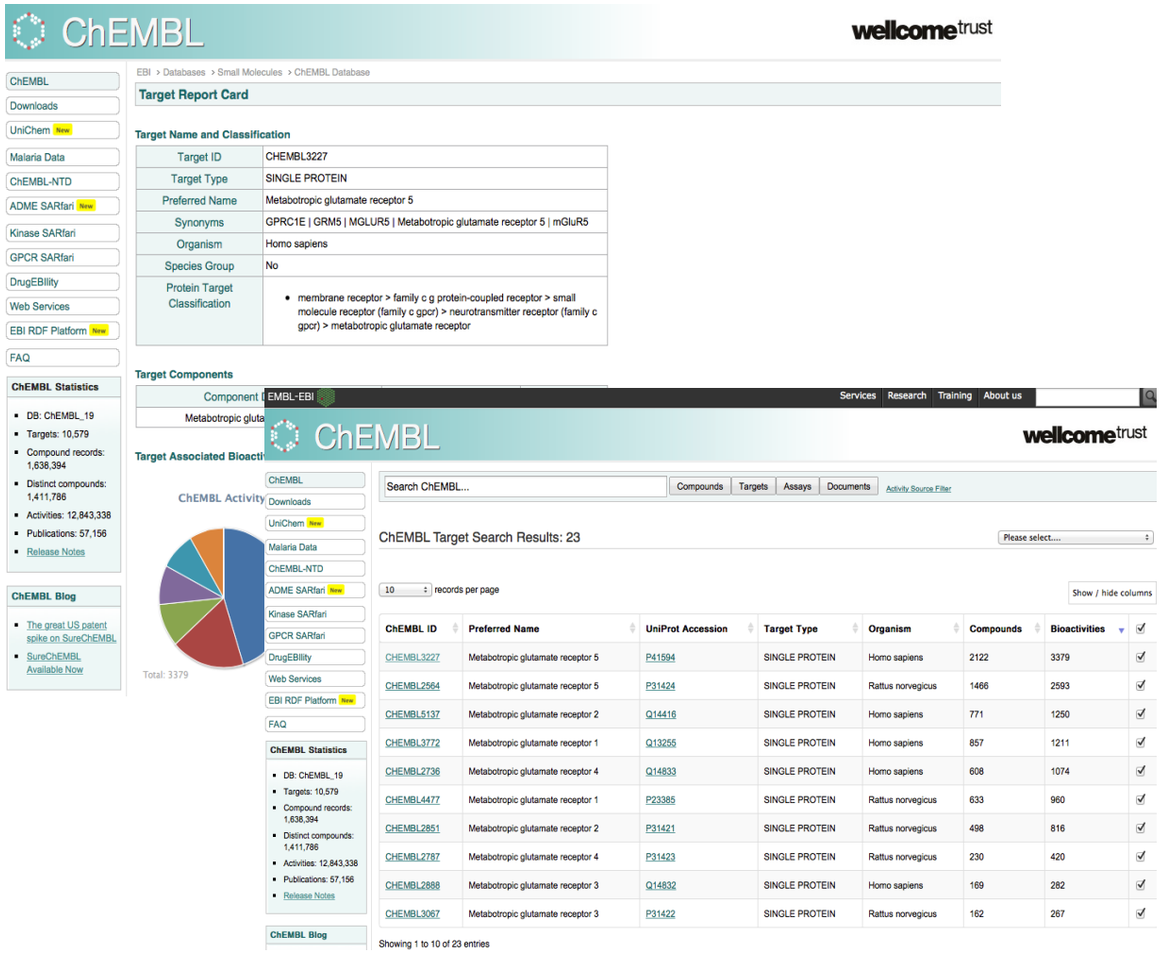


Collaboratory: bring data scientists and domain scientists together online (and their data and tools)

Easy, large-scale collab: Extract actionable datasets, key tools. Scientists shares data and get help, DS can test technique on many current datasets.

Real-time collab: share experiments automatically, discuss online. Automate experimentation.

OpenML in drug discovery



ChEMBL database

SMILES

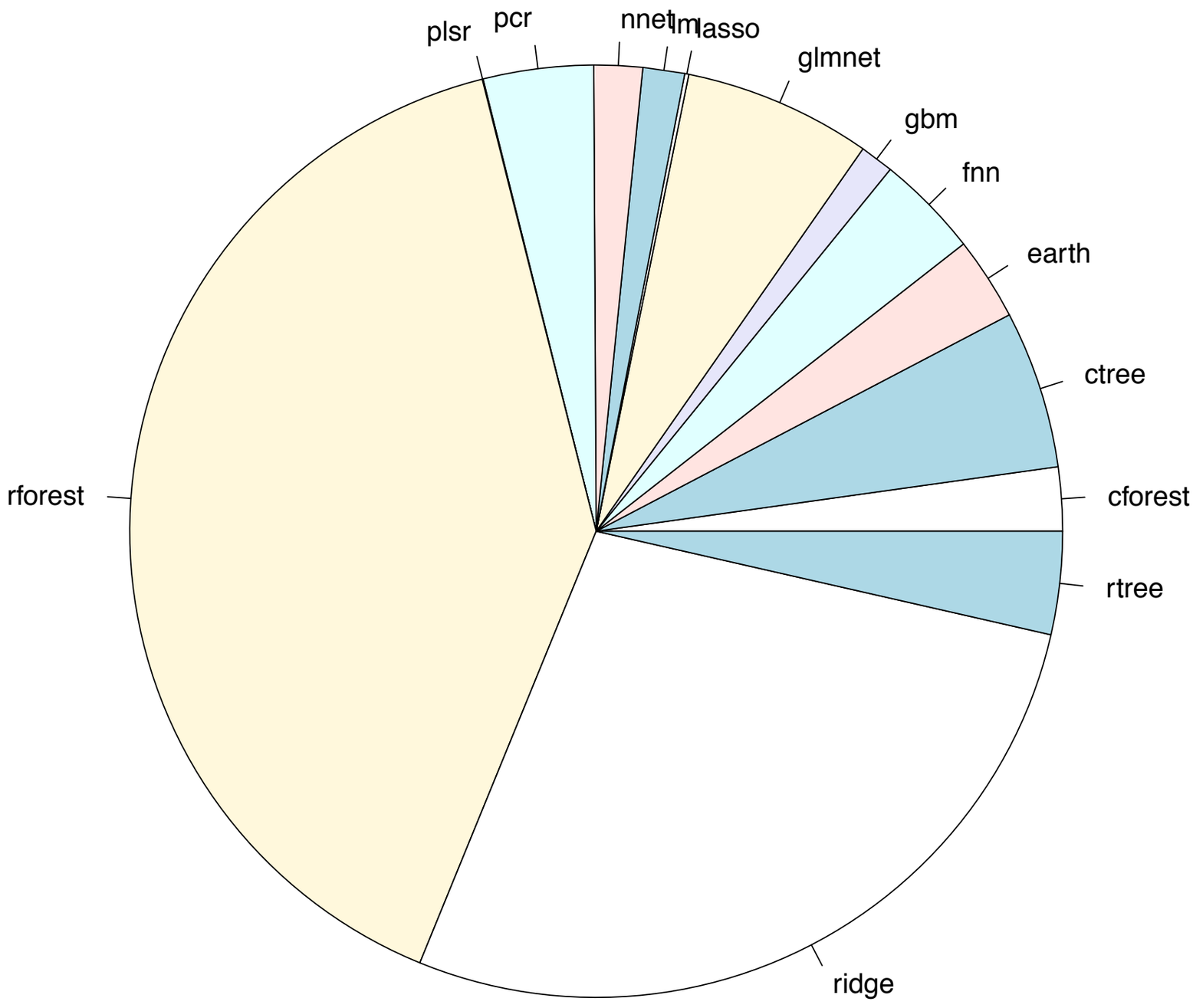
Molecular properties (e.g. MW, LogP)

Fingerprints (e.g. FP2, FP3, FP4, MACSS)

MW	LogP	TPSA	b1	b2	b3	b4	b5	b6	b7	b8	b9	
377.435	3.883	77.85	1	1	0	0	0	0	0	0	0	
341.361	3.411	74.73	1	1	0	1	0	0	0	0	0	...
197.188	-2.089	103.78	1	1	0	1	0	0	0	1	0	
346.813	4.705	50.70	1	0	0	1	0	0	0	0	0	
							⋮					

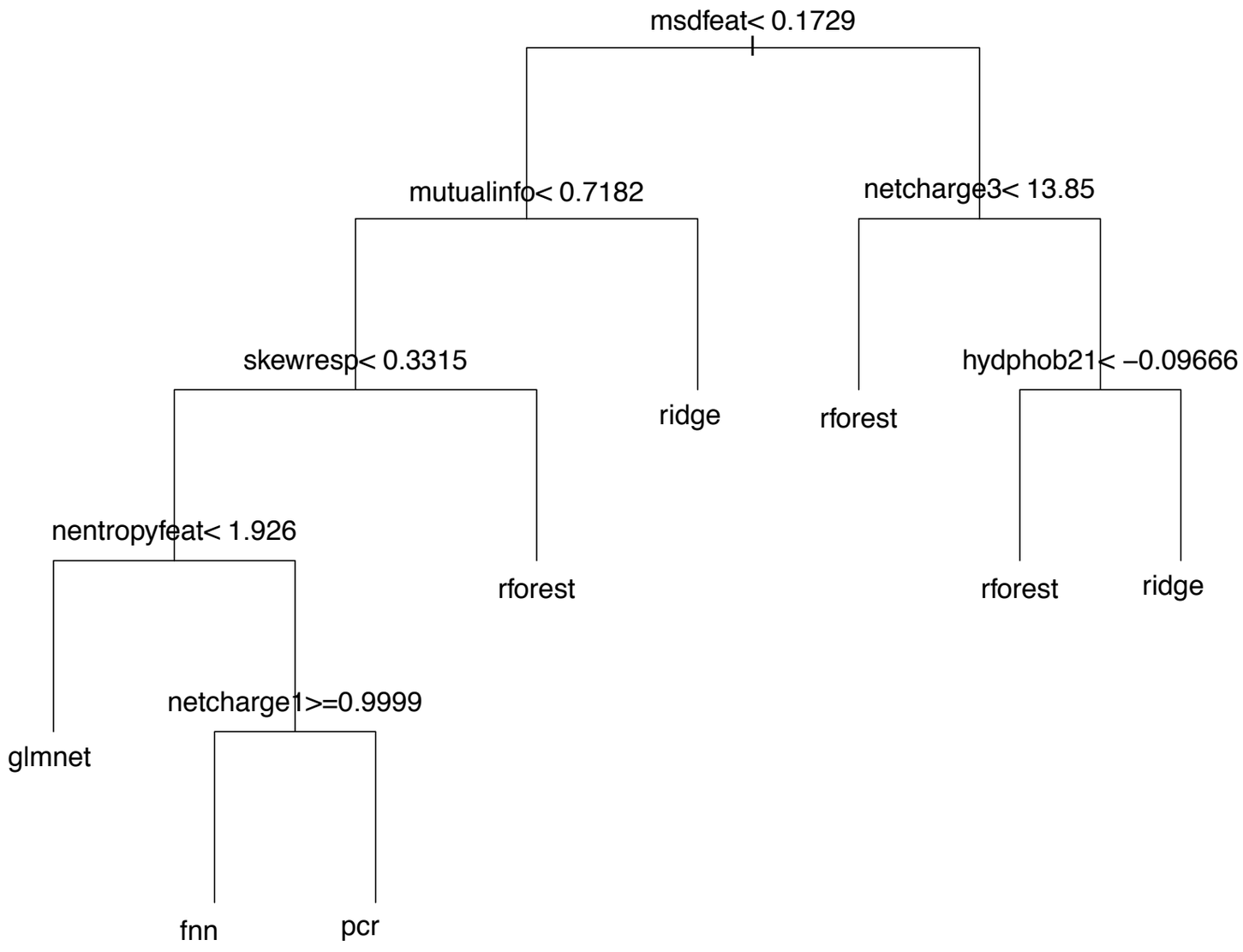
10.000+ regression datasets

OpenML in drug discovery



Best algorithm

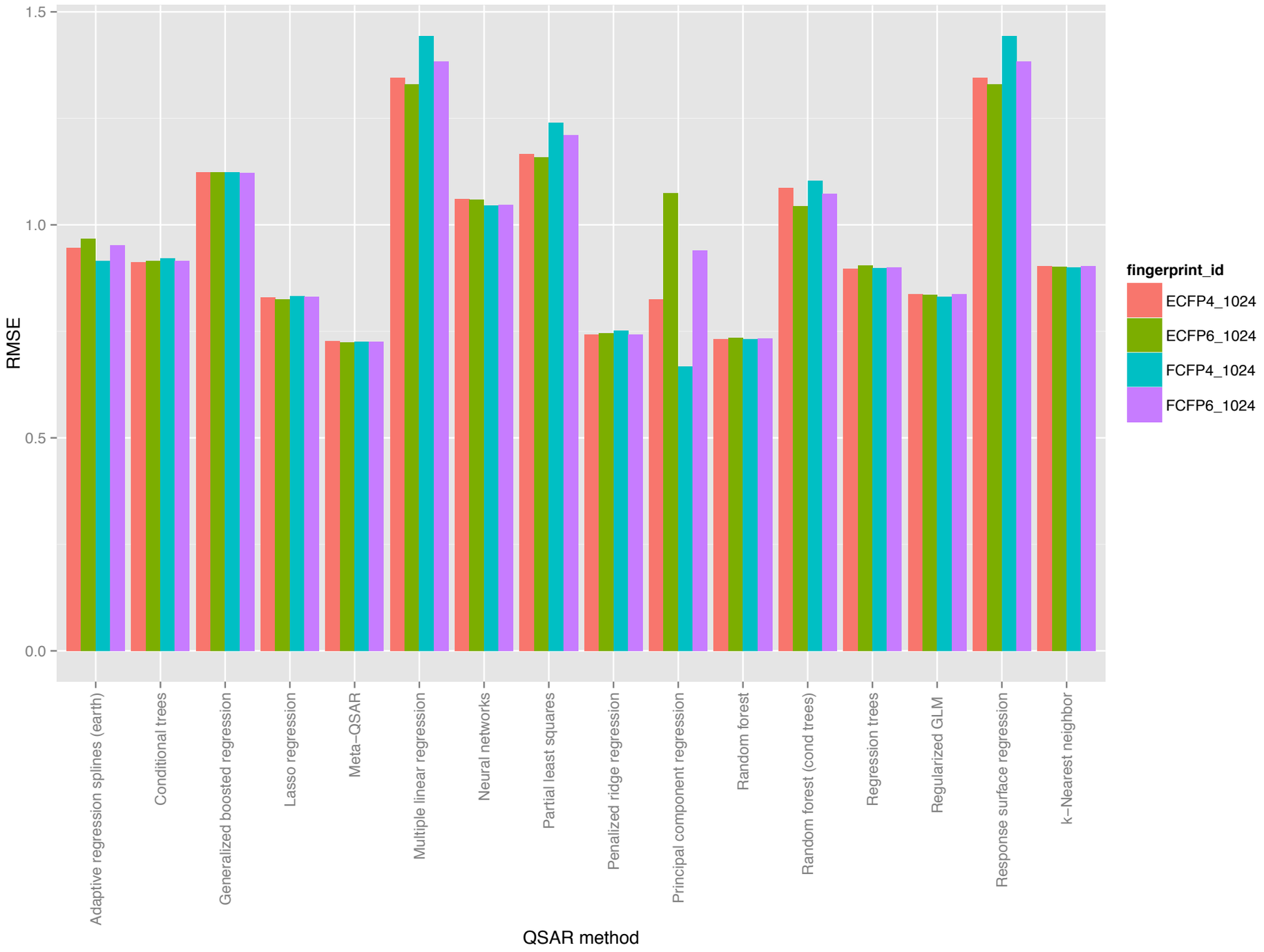
OpenML in drug discovery



ECFP4_1024	FCFP4_1024	**ECFP6_1024**	FCFP6_1024
0.697	0.427	**0.725**	0.627

meta-model

OpenML in drug discovery



usefulness of fingerprints per algorithm

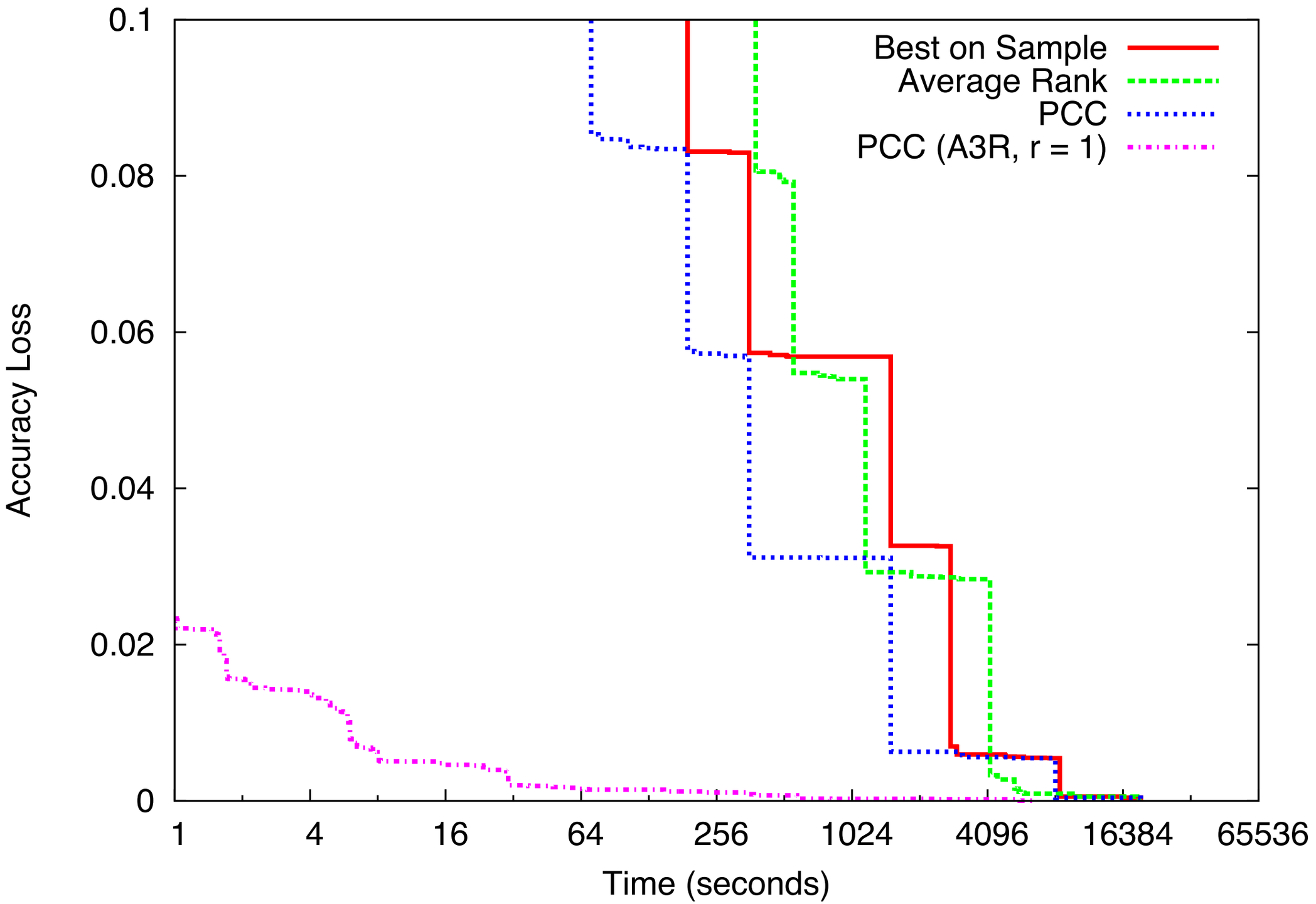
Towards automating machine learning

Meta-data:
- Growing collection of datasets
- Wide array of meta-features (more coming)
- Wide range of integrated machine learning algorithms

Algorithm selection
- Use existing experiments to train algorithm selection techniques
- Upload dataset, ask system to propose ML algorithms
- People can start, track, visualise AS process from website

Techniques (and combinations!):
- Meta-learning
- Optimisation (e.g. sequential model based optimisation)
- Dataset similarity

Fast algorithm selection (WIP)



Towards automated ML support

Allow scientists to get the support they want

AI (novice)
Go to website, upload dataset, give system 1 day to return best possible flow

Man-Computer Symbiosis (intermediate/expert):
Answer specific questions by scientist. Combine machine learning with talents of human mind (hunches, expertise, cut and try).
E.g. find similar datasets, recommend techniques, tune workflows
System explains *why* it makes a recommendation.

Expert level (expert)
Expert asks system *how* it did something, system explains all details, show code, data (e.g. how did it recommend algorithm)

Man-Computer Symbiosis

Scientist:

Here's a new dataset I just created.

I want to do classification.

Are there similar datasets?

Tool:

I analysed the data. Here's a report on the main characteristics.

I created a task, you can now run workflows on it.

Here are similar dataset:
D1 (similarity 83%)
by J. AppleSeed, published in Nature
Best workflow: RandomForests…

D2 (similarity 64%)
…

Man-Computer Symbiosis

Scientist:

Which methods should I use?

Which outlier detection technique should I use

Who uses Technique 1?

Tool:

Your data has a lot of missing values and outliers, I would recommend fixing that.

On this dataset, I recommend the following:

- Technique 1, used in studies …
- Technique 2, very useful for this type of outliers (see study X)

Here are some studies. It was developed by J. Jonhson,…

Man-Computer Symbiosis

Scientist:	Tool:
Which studies have worked on similar data?	Here's a list ranked by similarity and classification performance.
What's the difference between technique 1 and 2	Here's a comparison. Technique 1 seems to be better on datasets with large number of features
Here's my workflow, can you optimize it in 1 day	Running. I will try the fastest, most promising techniques first.

OpenML

THANK YOU

Joaquin Vanschoren
University of Eindhoven

Hendrik Blockeel
University of Leuven

Geoffrey Holmes
University of Waikato

Jan van Rijn
University of Leiden

Luis Torgo
University of Porto

Bernd Bischl
Universty of Dortmund

Simon Fischer
RapidMiner

Patrick Winter
KNIME.com

Bo Gao
University of Leuven

Milan Vukicevic
University of Belgrade

Sandro Radovanović
University of Belgrade

You?
Join now!