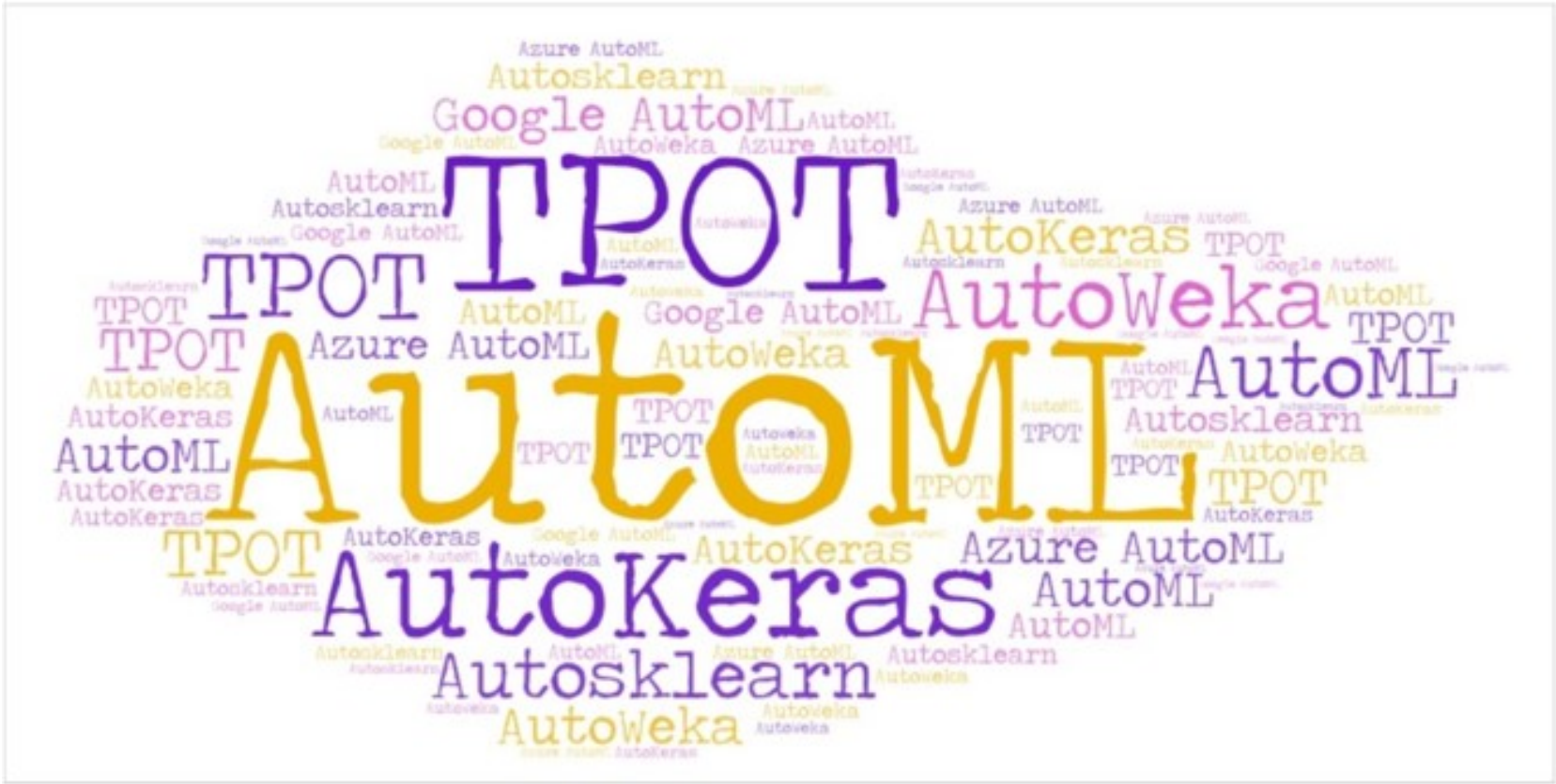
MetaLearning - AutoML

João Gama
jgama@fep.up.pt

FEP & University of Porto

Outline

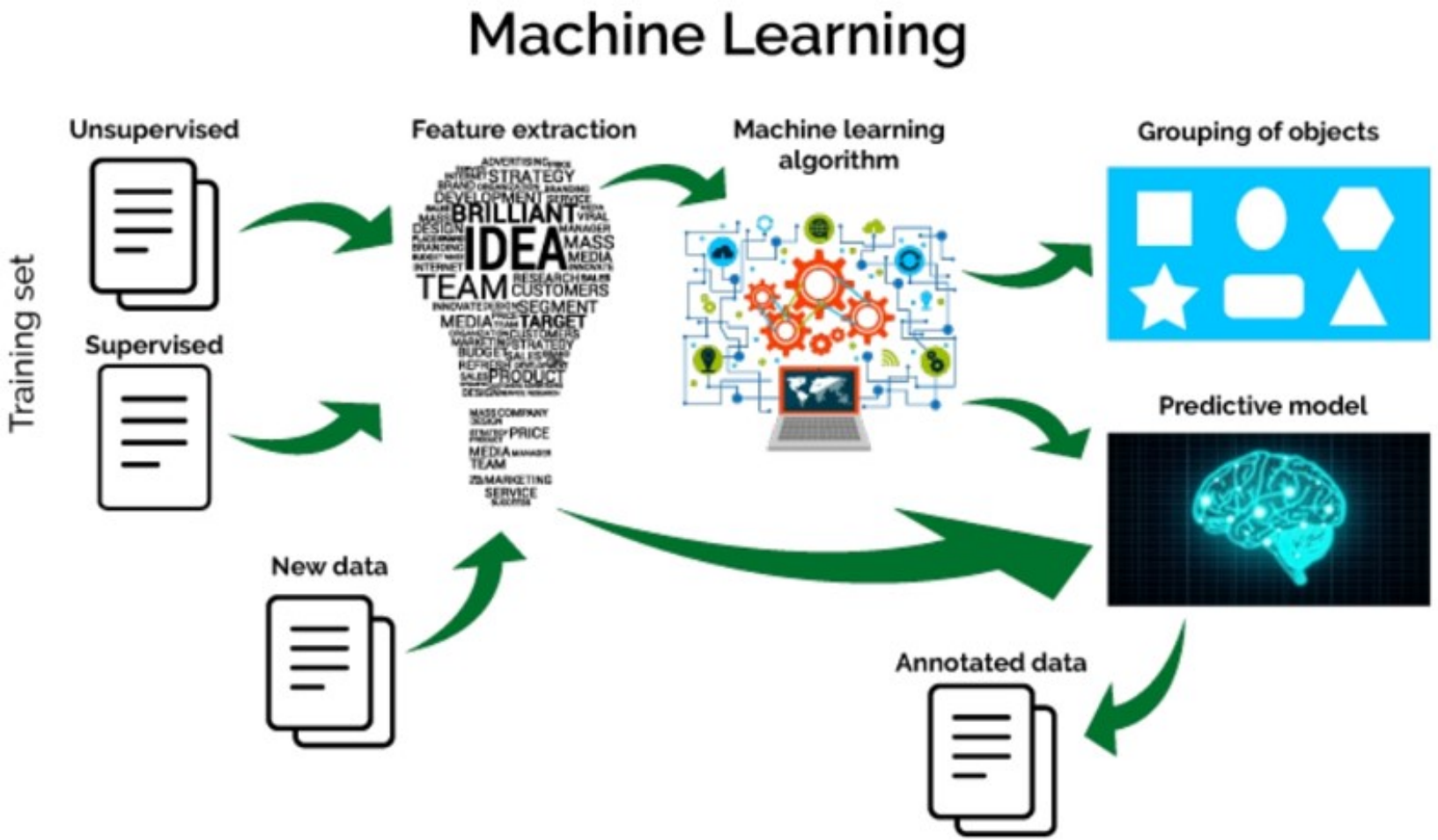
Automated machine learning (AutoML) is the process of automating the end-to-end process of applying machine learning to real-world problems.

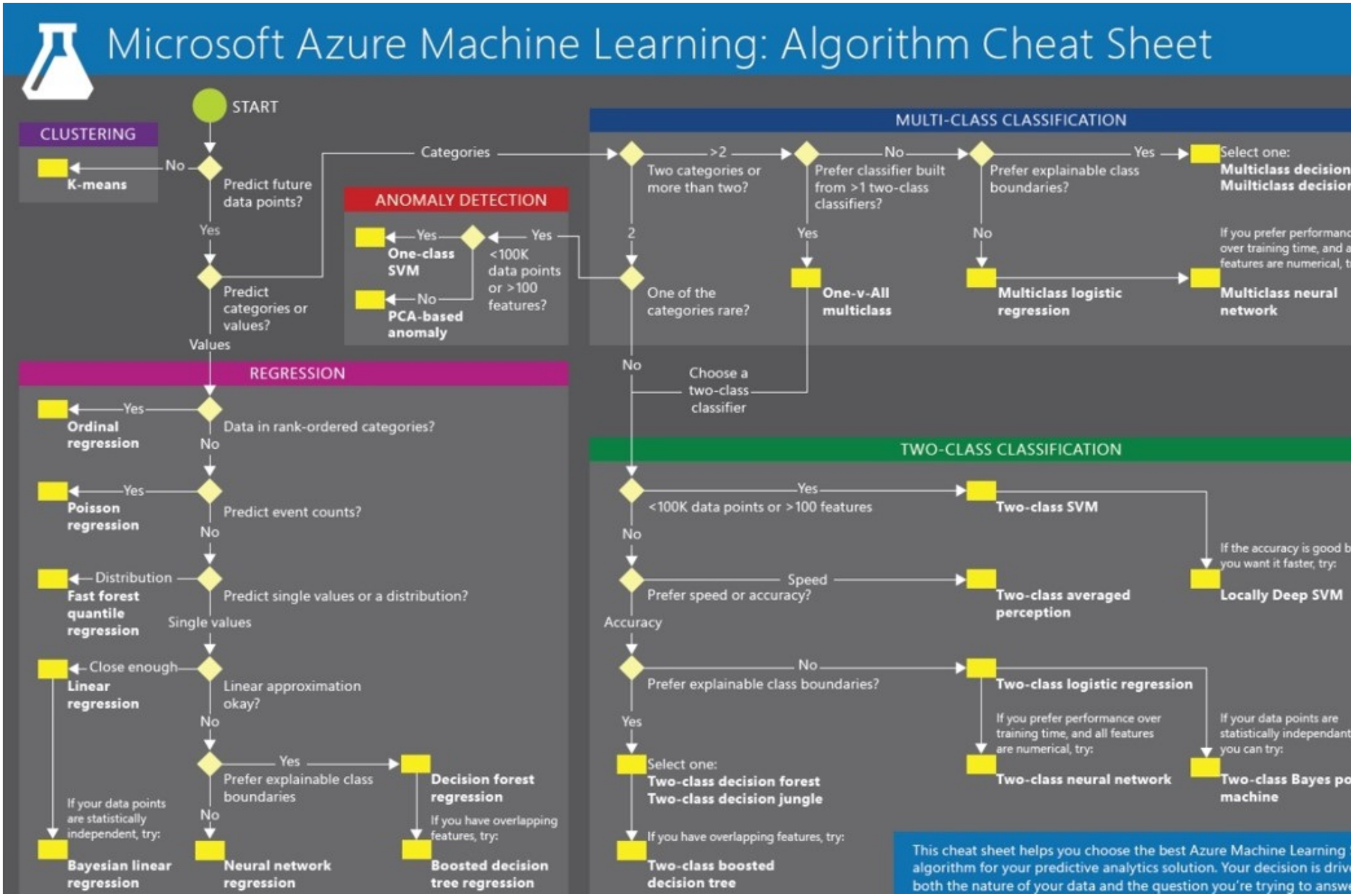


The Perfect Model Does Not Exist

"All models are wrong,
but some are useful."
–GEORGE BOX, 1919-2013

Machine Learning High-Level Overview




Microsoft Azure Machine Learning: Algorithm Cheat Sheet
START
CLUSTERING
No
K-means
Predict future data points?
Yes
Predict categories or values?
Categories
Values
ANOMALY DETECTION
Yes
One-class SVM
No
PCA-based anomaly
Yes
<100K data points or >100 features?
REGRESSION
Yes
Ordinal regression
Data in rank-ordered categories?
No
Yes
Poisson regression
Predict event counts?
No
Distribution
Fast forest quantile regression
Predict single values or a distribution?
Single values
Close enough
Linear regression
Linear approximation okay?
No
Yes
Prefer explainable class boundaries
No
If your data points are statistically independent, try:
Bayesian linear regression
Neural network regression
Decision forest regression
If you have overlapping features, try:
Boosted decision tree regression
MULTI-CLASS CLASSIFICATION
>2
Two categories or more than two?
2
One of the categories rare?
No
Choose a two-class classifier
No
Prefer classifier built from >1 two-class classifiers?
Yes
One-v-All multiclass
Yes
Prefer explainable class boundaries?
No
Multiclass logistic regression
Yes
Select one:
Multiclass decision
Multiclass decision
If you prefer performanc
over training time, and a
features are numerical, t
Multiclass neural network
TWO-CLASS CLASSIFICATION
Yes
<100K data points or >100 features
Two-class SVM
No
Speed
Prefer speed or accuracy?
Two-class averaged perception
If the accuracy is good b
you want it faster, try:
Locally Deep SVM
Accuracy
No
Prefer explainable class boundaries?
Two-class logistic regression
If you prefer performance over training time, and all features are numerical, try:
Two-class neural network
If your data points are statistically independant
you can try:
Two-class Bayes po
machine
Yes
Select one:
Two-class decision forest
Two-class decision jungle
If you have overlapping features, try:
Two-class boosted decision tree
This cheat sheet helps you choose the best Azure Machine Learning
algorithm for your predictive analytics solution. Your decision is driv
both the nature of your data and the question you're trying to answ

What is Automated Machine Learning (AutoML)?

"Machine learning is very successful, but its successes crucially rely on human machine learning experts, who select appropriate ML architectures (deep learning architectures or more traditional ML workflows) and their hyper-parameters.
As the complexity of these tasks is often beyond non-experts, the rapid growth of machine learning applications has created a demand for off-the-shelf machine learning methods that can be used easily and without expert knowledge.
We call the resulting research area that targets progressive automation of machine learning AutoML."

https://sites.google.com/site/automl2016/

Why Automated Machine Learning (AutoML)?

- The demand for machine learning experts has outpaced the supply. To address this gap, there have been big strides in the development of user-friendly machine learning software that can be used by non-experts and experts, alike.
- AutoML software can be used for automating a large part of the machine learning workflow, which includes automatic training and tuning of many models within a user-specified time-limit.

What is NOT Automated Machine Learning (AutoML)?

- AutoML is not automated data science;
- AutoML will not replace Data Scientist;
  - All the methods of automated machine learning are developed to support data scientists, not to replace them.
  - AutoML is to free data scientists from the burden of repetitive and time-consuming tasks (e.g., machine learning pipeline design and hyper-parameter optimization) so they can better spend their time on tasks that are much more difficult to automate.

Tasks

- Automated Feature Engineering
  - Feature selection
  - Feature extraction
  - Detection and handling of skewed data and/or missing values
- Model Selection
  - Meta learning and transfer learning
- Hyper-parameter optimization

Outline

Model selection considerations

Interpretability

Ability of users to understand the model, the parameters of the model and their effect on the outcome

- In regression, coefficients enable us to interpret the influence of an independent variable on the dependent variable.
  The standard error of estimates of the coefficients enable us to determine how confident are we on these estimates
- In decision trees a complex decision is a sequence of simple decision

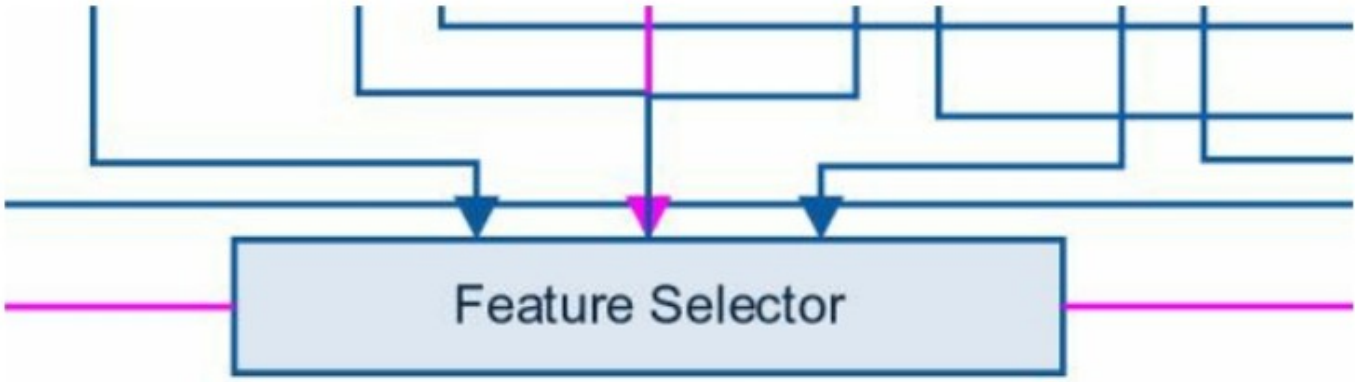
Model selection considerations

Parsimonious models

A parsimonious model is a model that accomplishes a desired level of explanation or prediction with as few predictor variables as possible.

- Models that use internal feature selection: decision and regression trees, decision rules
- In regression, using Exhaustive search, Forward search, Backward search or Stepwise regression in model selection

Feature Selection



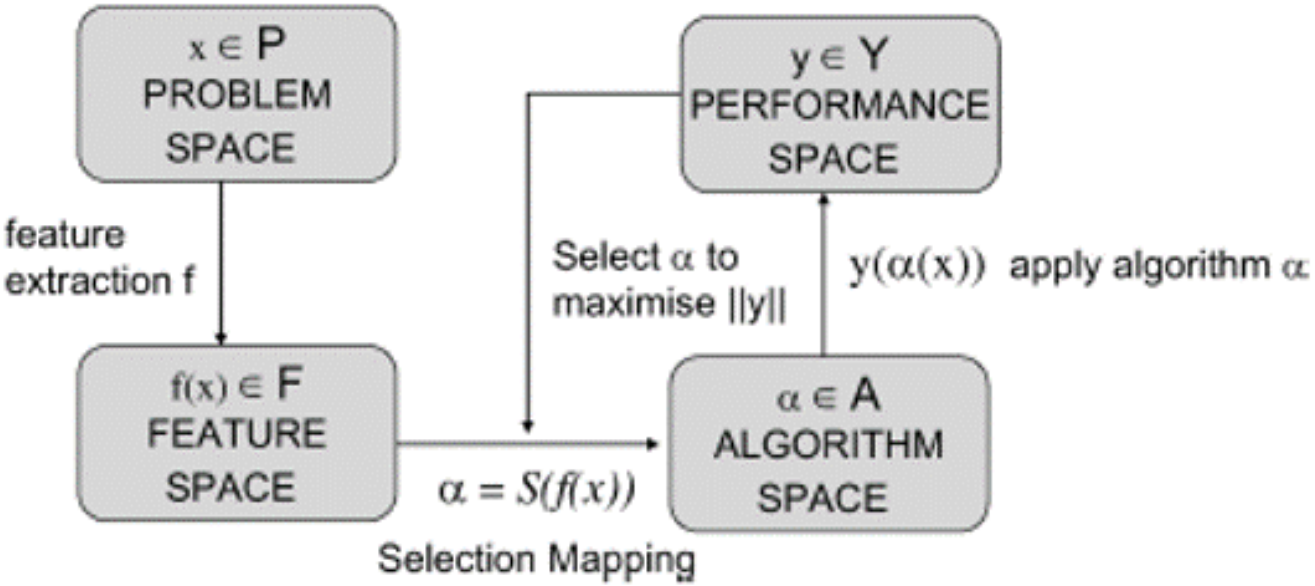
- Greedy Forward Selection
  - Selecting best features iteratively
  - Selecting features based on coefficients of model
- Greedy backward elimination
- Use GBM for normal features
- Random Forest for feature rank

Outline

The Algorithm Selection Problem

- Typically many different algorithms exist in a particular domain (classification, regression, optimization etc.).
- Given a decision problem, We want methods that can help us to select the one with the best performance.
- This problem was first formulated by Rice [1976]:
- For a given problem instance $x \in P$, with features $f(x) \in F$,
  - find the selection mapping $S(f(x))$ into algorithm space A,
  - such that the selected algorithm $\alpha \in A$
  - maximizes the performance mapping $y(\alpha(x)) \in Y$.

The Algorithm Selection Problem



- In subsequent work the selection mapping S(f(x)) was generated using ML methods.
- The process is often referred to as meta-learning.
- The process can be applied to the problem of selecting classification algorithms.

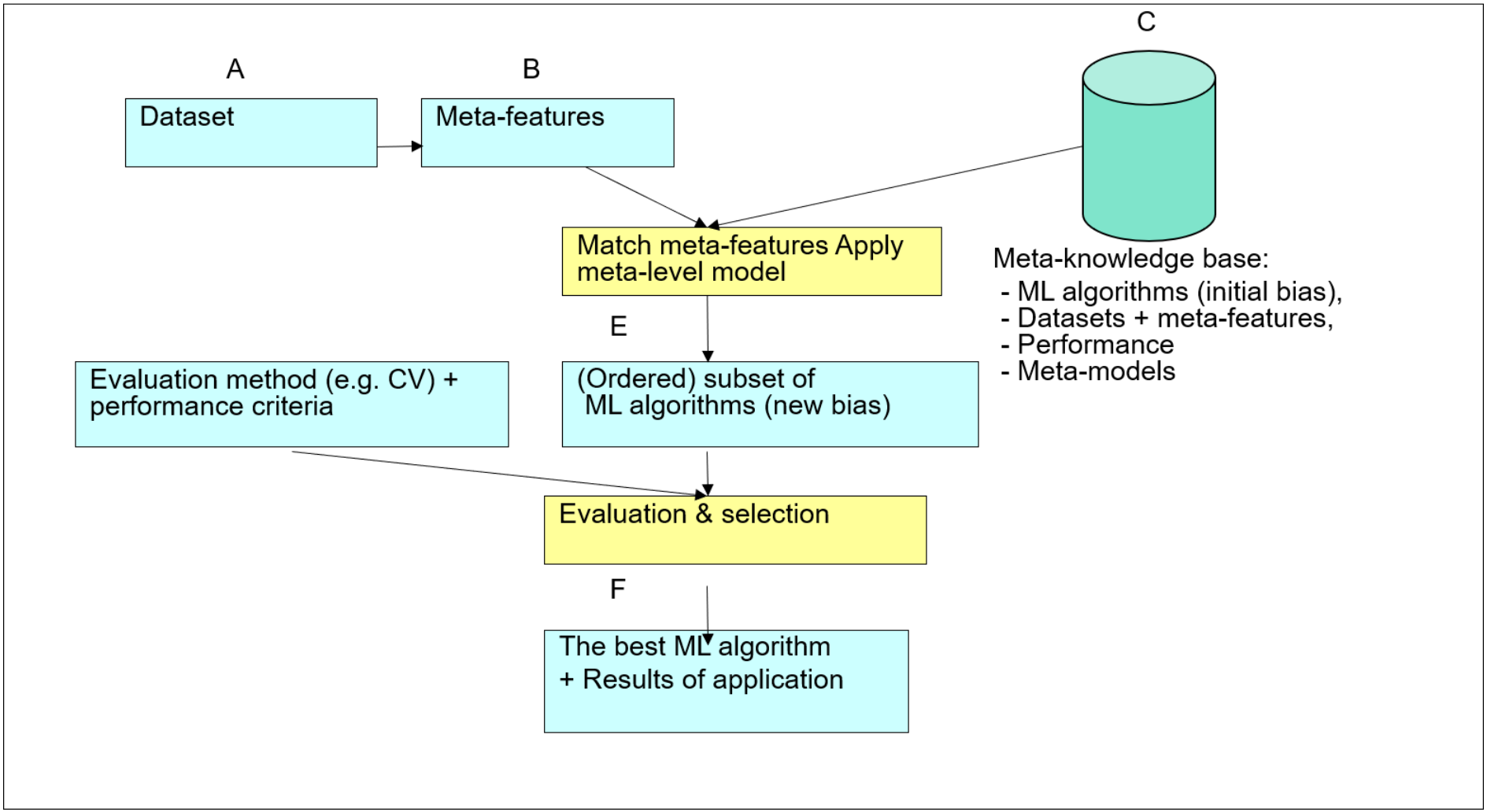
Meta-Learning for Algorithm Selection

- A large set of techniques is available in Machine Learning (ML).
  - + It increases a possibility that a good solution can be found.
  - – It is much harder to find the right ML algorithm, as many alternatives exist.
- The problem of selecting a suitable (the best) algorithm can be seen as a problem of search.
- We cannot test all ML algorithms for computational reasons (there are thousands of variants of ML algorithm + parameter settings)
- Why meta-learning?
  - It helps to build on previous experience and
  - identify the right algorithm more effectively.
- In this part we focus on classification algorithms.

Identifying a Subset of Algorithms with Meta-models

- Phase 1:
  - Consider the given new dataset,
    construct characteristics / meta-features.
  - Exploit meta-level model to
    identify a suitable subset of algorithms.
- In some work the result is a ranked subset of classification algorithms permitting reduced search.
- Phase 2: Search through the reduced space of algorithms.
  - Evaluate each option using
    - a chosen evaluation method (typically a cross-validation) and
    - a given performance criteria (e.g. accuracy).
  - Identify the best alternative (or an algorithm that is comparable).

Selecting ML Algorithms on Meta-features



Meta-Features: Data Characteristics

- The selection method relies on dataset characteristics or meta-features to provide some information that can differentiate performance of a set of given learning algorithms.
- These typically include :
  - statistical and information-theoretic measures,
  - model based characterization,
  - landmarking

Statistical and Information-Theoretic Measures

- These measures typically include :
  - number of classes,
  - number of features,
  - ratio of examples to features,
  - degree of correlation between features and target concept,
  - average class entropy
  - etc.
- + Positive and tangible results (e.g., ESPRIT Statlog and METAL).
- – There is a limit on how much information these measures can capture,
  - as these measures are uni- or bi-lateral measures only
  - capture relationships between two attributes only or
  - one attribute and the class.

Landmarkers

- Examine performance of a set of simple and fast learning algorithms (land-markers).
- The accuracy of these land-markers is used to characterize the dataset.

- A meta-level system / model helps to map characteristics into classification algorithms.
- The meta-level system / model can be in the form of:
  - meta-level rules,
  - k-NN (on the meta-level),
  - neural network,
  - other type of classification model on the meta-level.

Employing a Rule Model on the Meta-Level

- Early approaches (Rendell & Cho, 1990) used rules, such as:
  - If the given dataset characteristics are $C_1, C_2, \ldots, C_n$ then use algorithm A_1 in preference to algorithm A_2.

```
IF (# training instance < 737) AND
      (# prototype per class > 5.5) AND
      (# relevants > 8.5) AND
      (# irrelevants < 5.5)
THEN IB1 will be better than CN2
```

Employing 1-NN on the Meta-Level

One simple approach uses 1-NN :

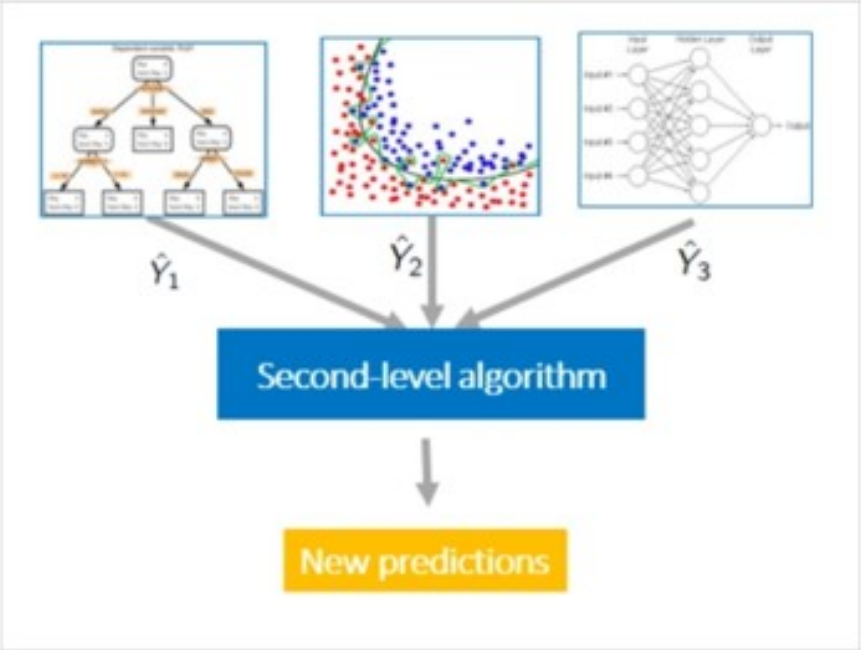
- Compare meta-level characteristics of the new problem with meta-level characteristics of past problems,
- Identify the most similar dataset
- Retrieve either :
  - The classification algorithm that performed best on that dataset,
  - Ranking of classification algorithms, ordered by performance.

Employing k-NN on the Meta-Level

A more complex approach employs k-NN:

- Uses k-NN method to identify the most similar datasets.
- For each of these datasets,
  - retrieves the ranking of the candidate classification algorithms,
  - based on past performance criteria (accuracy, learning time).
- Aggregate the rankings obtained to generate the final recommended ranking of algorithms.
- Evaluate the top N elements in the ranking (Phase II) select the one with the best performance.

Model selection considerations



Ensemble models

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

Outline

Parameters vs Hyper Parameters

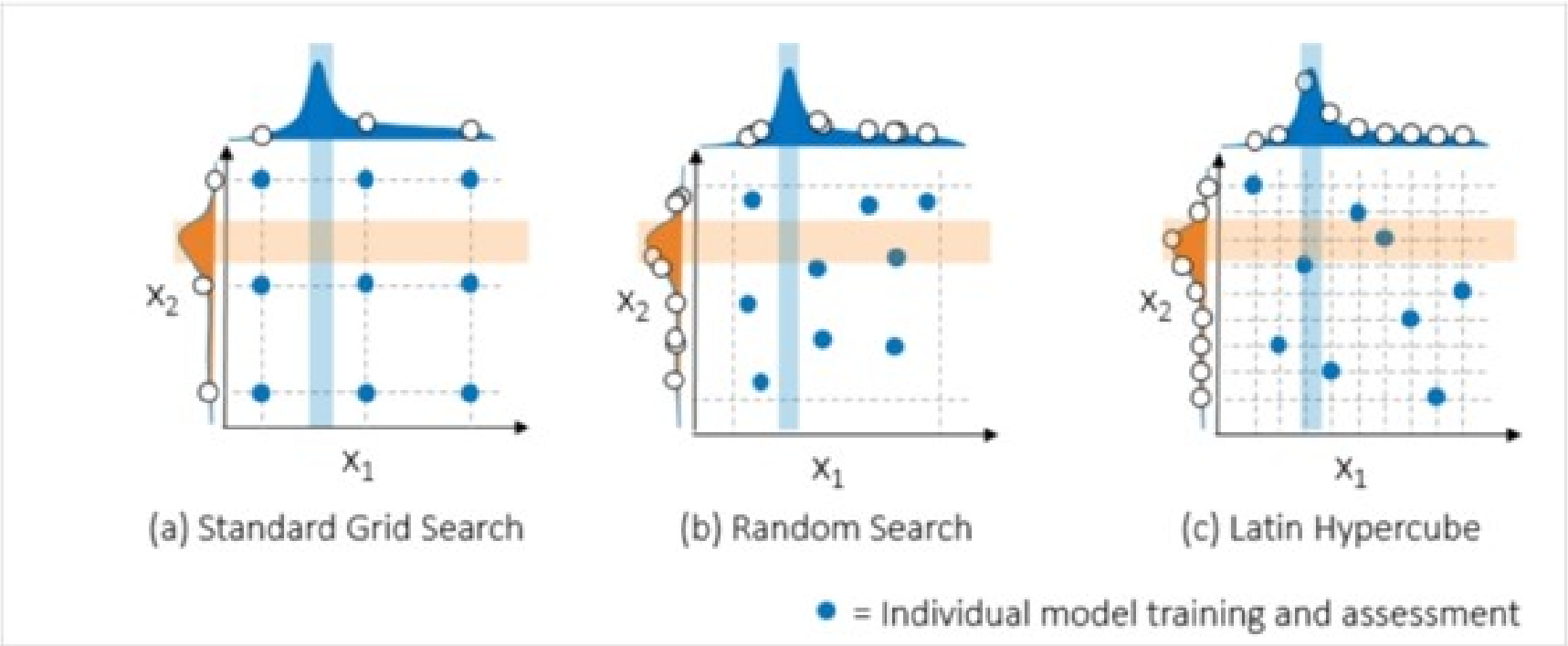
- Parameters: Values that can be estimated from data
  - Examples:
    - Regression Coefficients
    - Weights in a Neural Network
- HyperParameters: Values external to the model and cannot be learnt from the data
  - Examples:
    - k in k-Means
    - Learning rate in Neural Network
    - Regularization parameters

Hyperparameters

Model	Parameters to optimize	Good range of values
Linear Regression	• fit_intercept • normalize	• True / False • True / False
Ridge	• alpha • Fit_intercept • Normalize	• 0.01, 0.1, 1.0, 10, 100 • True/False • True/False
k-neighbors	• N_neighbors • p	• 2, 4, 8, 16 .... • 2, 3
SVM	• C • Gamma • class_weight	• 0.001, 0.01.....10...100...1000 • 'Auto', RS* • 'Balanced' , None
Logistic Regression	• Penalty • C	• L1 or l2 • 0.001, 0.01.....10...100
Naive Bayes (all variations)	NONE	NONE
Lasso	• Alpha • Normalize	• 0.1, 1.0, 10 • True/False
Random Forest	• N_estimators • Max_depth • Min_samples_split • Min_samples_leaf • Max features	• 120, 300, 500, 800, 1200 • 5, 8, 15, 25, 30, None • 1, 2, 5, 10, 15, 100 • 1, 2, 5, 10 • Log2, sqrt, None
Xgboost	• Eta • Gamma • Max_depth • Min_child_weight • Subsample • Colsample_bytree • Lambda • alpha	• 0.01,0.015, 0.025, 0.05, 0.1 • 0.05-0.1,0.3,0.5,0.7,0.9,1.0 • 3, 5, 7, 9, 12, 15, 17, 25 • 1, 3, 5, 7 • 0.6, 0.7, 0.8, 0.9, 1.0 • 0.6, 0.7, 0.8, 0.9, 1.0 • 0.01-0.1, 1.0 , RS* • 0, 0.1, 0.5, 1.0 RS*

Hyperparameter optimization

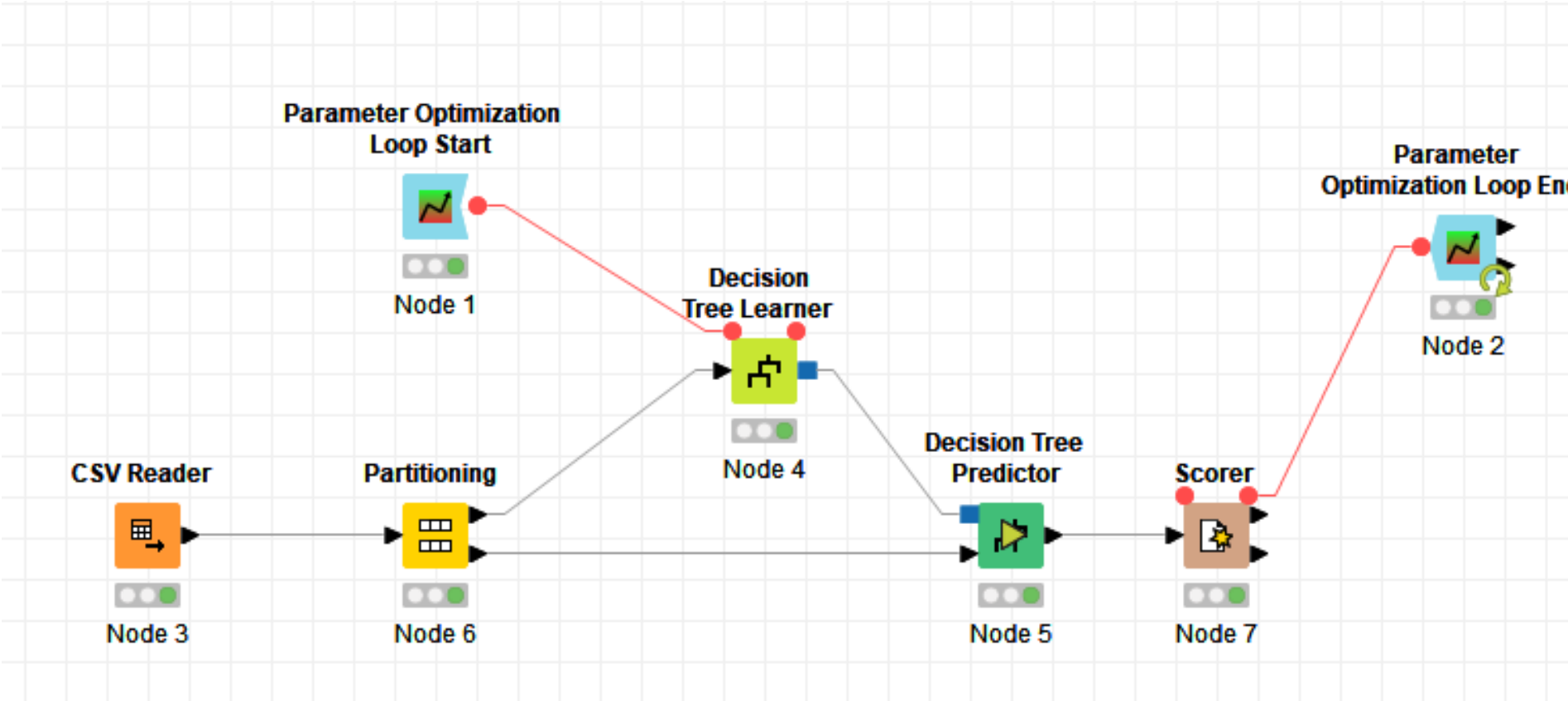
finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data



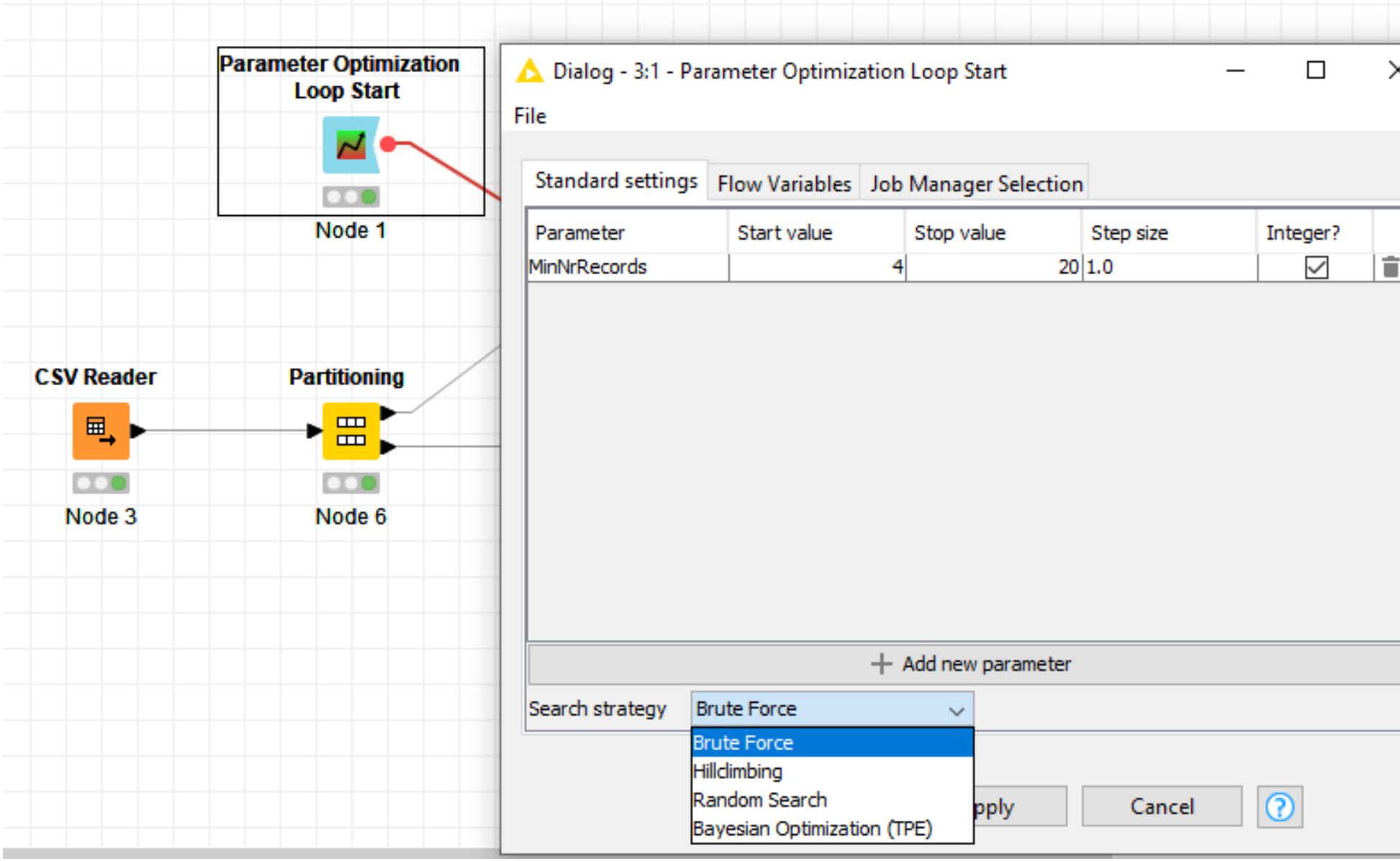
Hyperparameter optimization

- Grid search
  an exhaustive searching through a manually specified subset of the hyper-parameter space of a learning algorithm.
- Random search
  replaces the exhaustive enumeration of all combinations by selecting them randomly.
- Bayesian optimization
  builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set.
- Gradient-based optimization
  For specific learning algorithms, it is possible to compute the gradient with respect to hyper-parameters and then optimize the hyper-parameters using gradient descent.
- Evolutionary optimization
  uses evolutionary algorithms to search the space of hyper-parameters for a given algorithm.

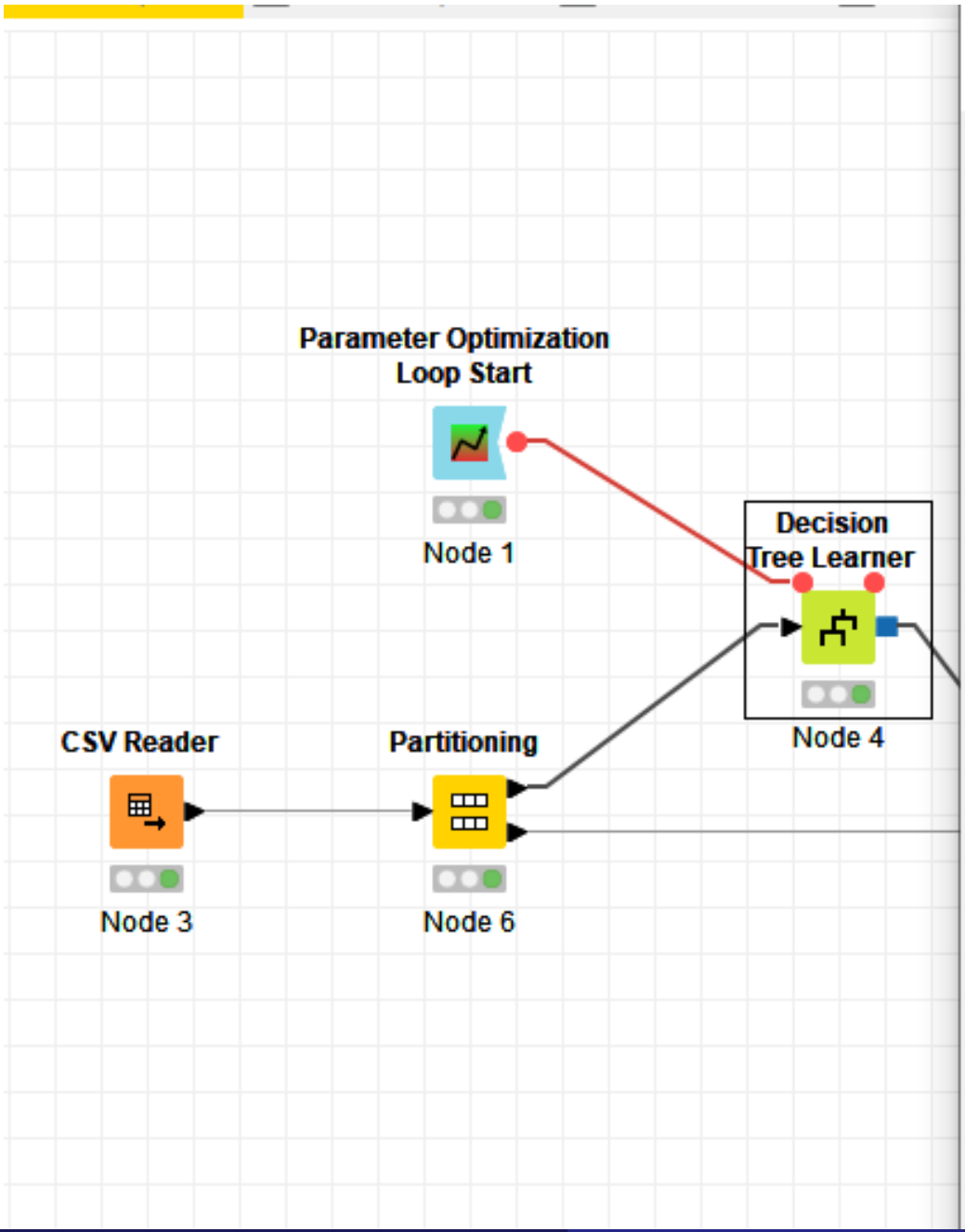
KNIME - Hyper-parameter tunning

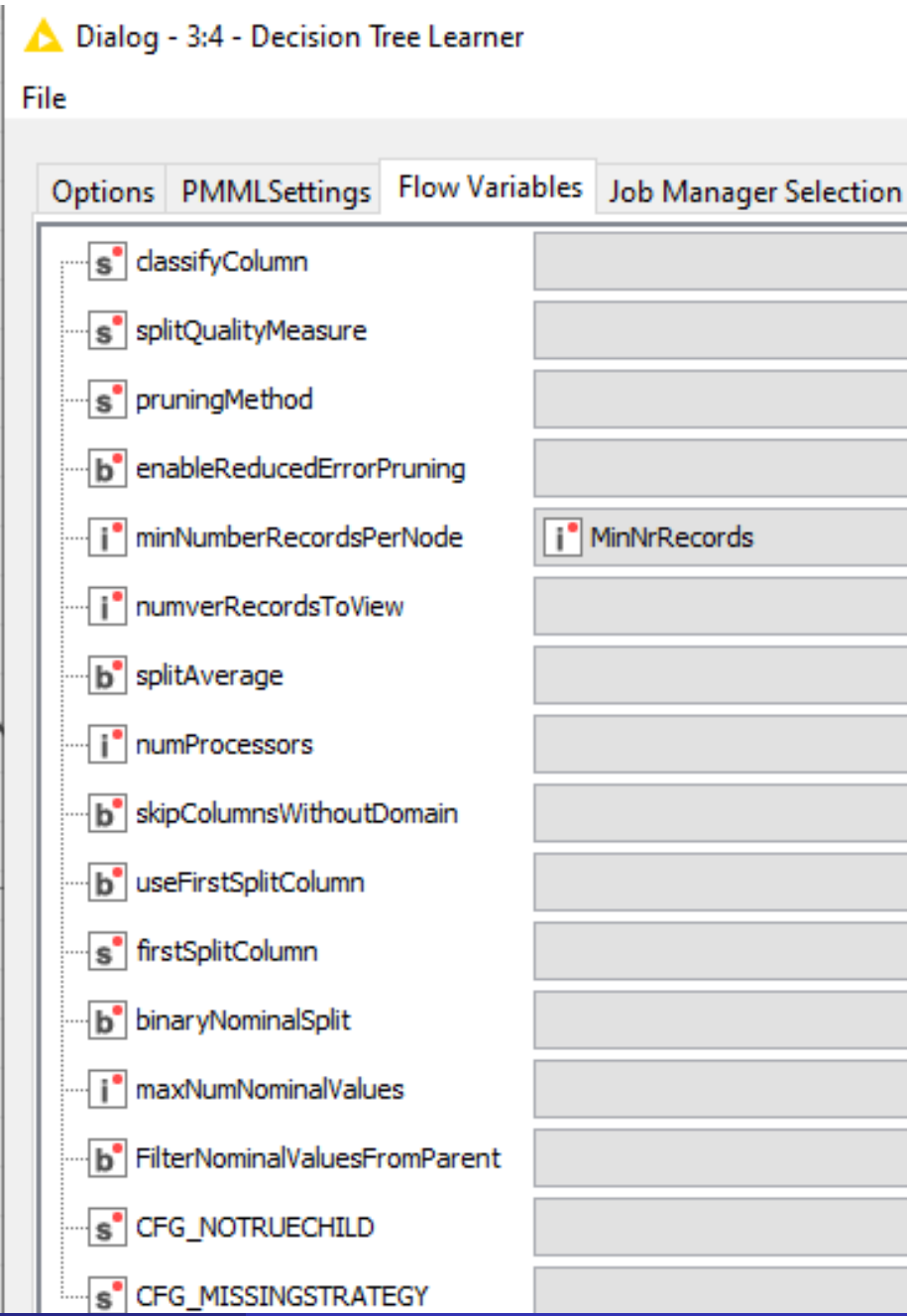


KNIME - Hyper-parameter tunning

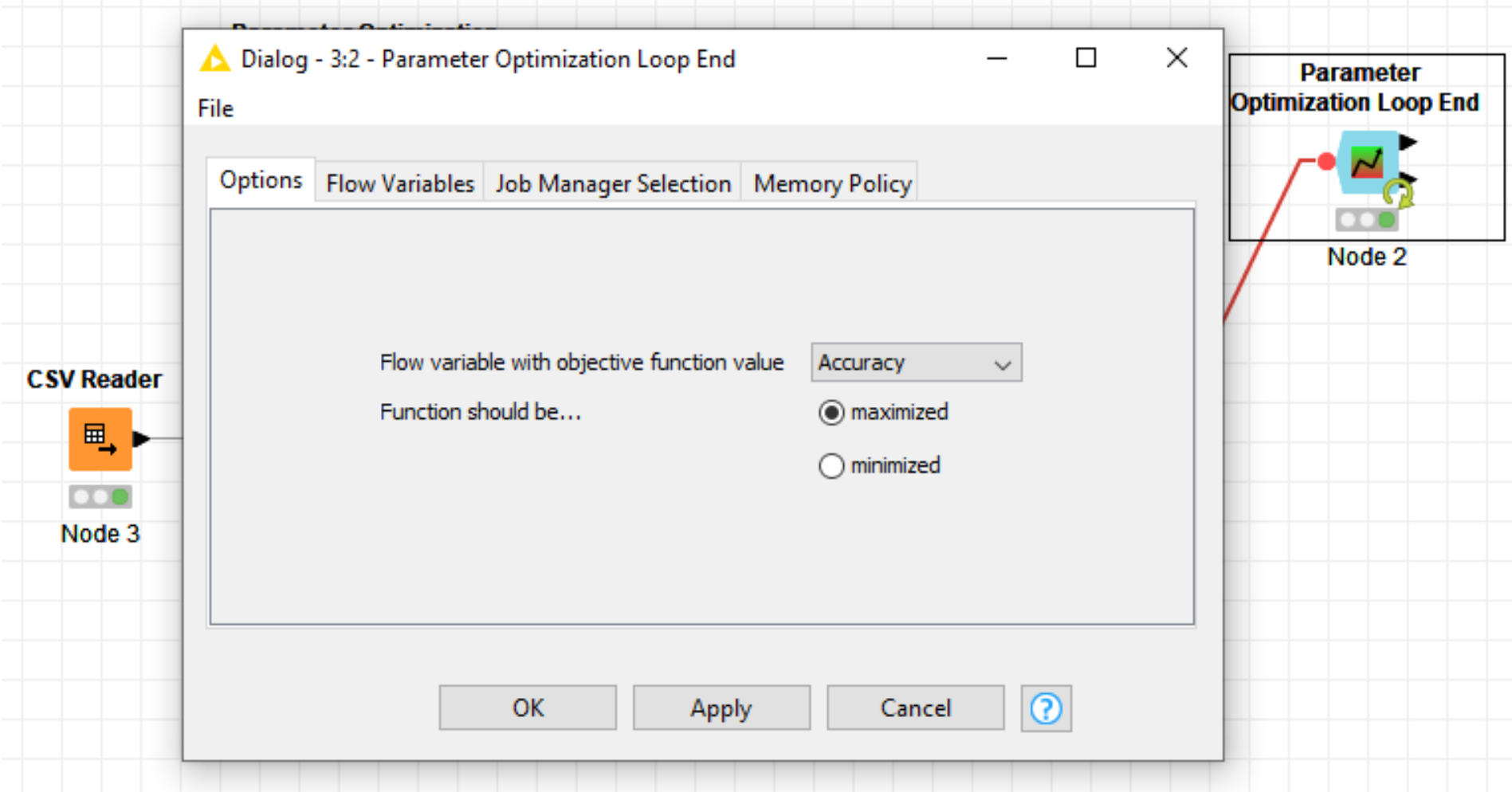


KNIME - Hyper-parameter tunning





KNIME - Hyper-parameter tunning



KNIME - Hyper-parameter tunning

All parameters - 3:2 - Parameter Optimizati...

File Edit Hilite Navigation View

Table "default" - Rows: 17 | Spec - Columns: 2 | Prop

Row ID	MinNrR...	Objecti...
Row0	4	0.963
Row1	5	0.967
Row2	6	0.967
Row3	7	0.967
Row4	8	0.97
Row5	9	0.967
Row6	10	0.963
Row7	11	0.959
Row8	12	0.955
Row9	13	0.957
Row10	14	0.957
Row11	15	0.957
Row12	16	0.953
Row13	17	0.945
Row14	18	0.945
Row15	19	0.945
Row16	20	0.947

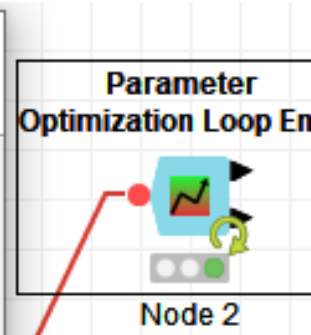
Best parameters - 3:2 - Parameter ...

File Edit Hilite Navigation View

Properties | Flow Variables

Table "default" - Rows: 1 | Spec - Columns: 2

Row ID	MinNrR...	Objecti...
Best parameters	8	0.97



Outline

Hyper-parameter optimization

- Auto Weka (Open Source)
  `http://www.cs.ubc.ca/labs/beta/Projects/autoweka/`
- H2o.ai AutoML (Open Source)
  `https://www.h2o.ai/`
- TPOT (Open Source)
  `https://github.com/rhiever/tpot`
- Auto Sklearn (Open Source)
  `https://github.com/automl/auto-sklearn`
  `http://automl.github.io/auto-sklearn/stable/`
- machineJS (Open Source)
  `https://github.com/ClimbsRocks/machineJS`
- AutoKeras
  `https://autokeras.com/`

Hyper-parameter optimization and Model Selection

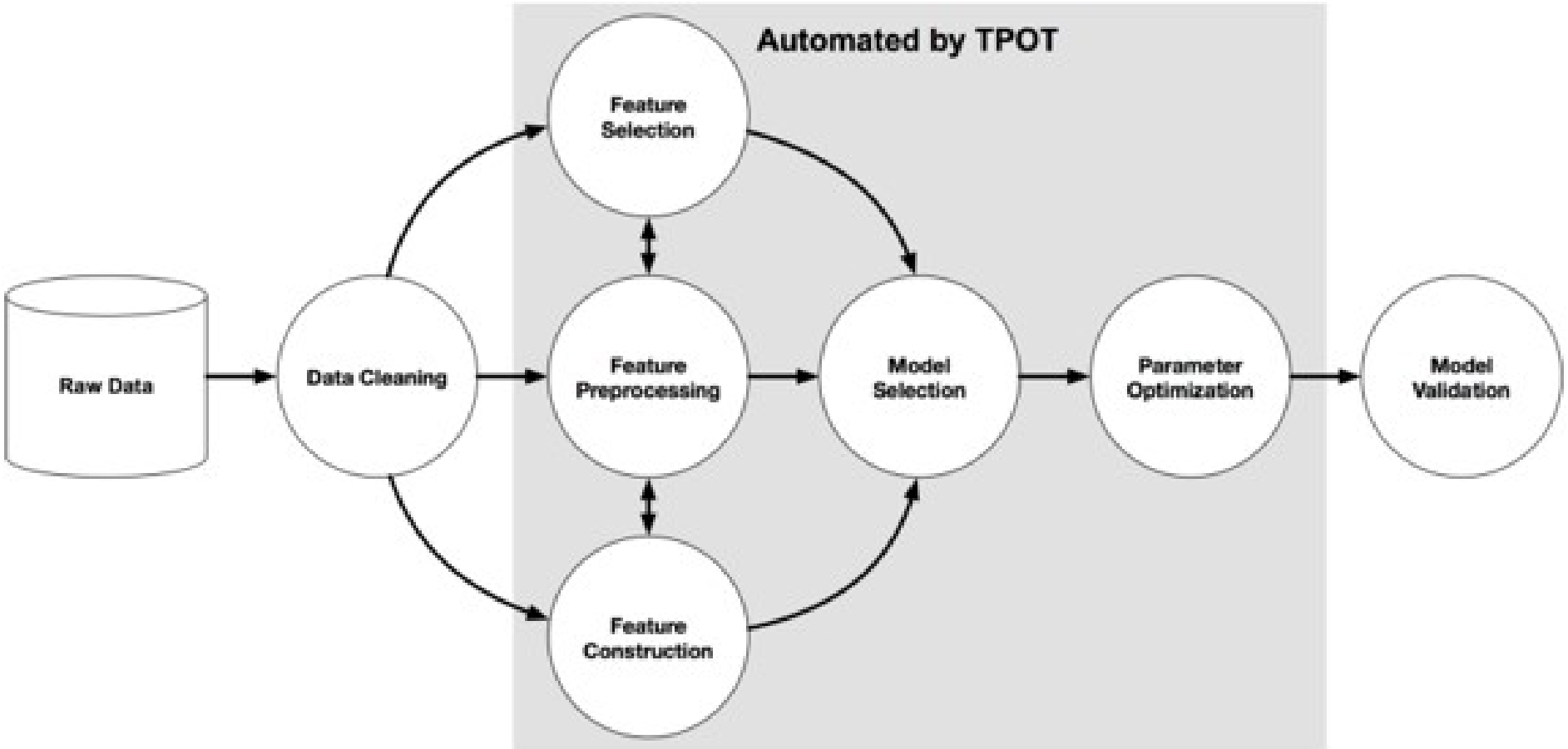
- AutoWEKA is an approach for the simultaneous selection of a machine learning algorithm and its hyper-parameters; combined with the WEKA package.
- Auto-sklearn is an extension of AutoWEKA using the Python library scikit- learn which is a drop-in replacement for regular scikit-learn classifiers and regressors. It improves over AutoWEKA by using meta-learning to increase search efficiency and post-hoc ensemble building to combine the models generated during the hyperparameter optimization process.
- TPOT is a data-science assistant which optimizes machine learning pipelines using genetic programming
- H2O AutoML provides automated model selection and ensembling for the H2O machine learning and data analytics platform.
- mlr is a R package that contains several hyper-parameter optimization techniques for machine learning problems.

Deep Neural Network Architecture search

- Google CLOUD AUTOML
  is an could-based machine learning service which so far provides the automated generation of computer vision pipelines.
- Auto Keras
  is an open-source python package for neural architecture search.
- Automl
  Deep Learning with Metaheuristic

Automated by TPOT

TPOT will automate the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for your data

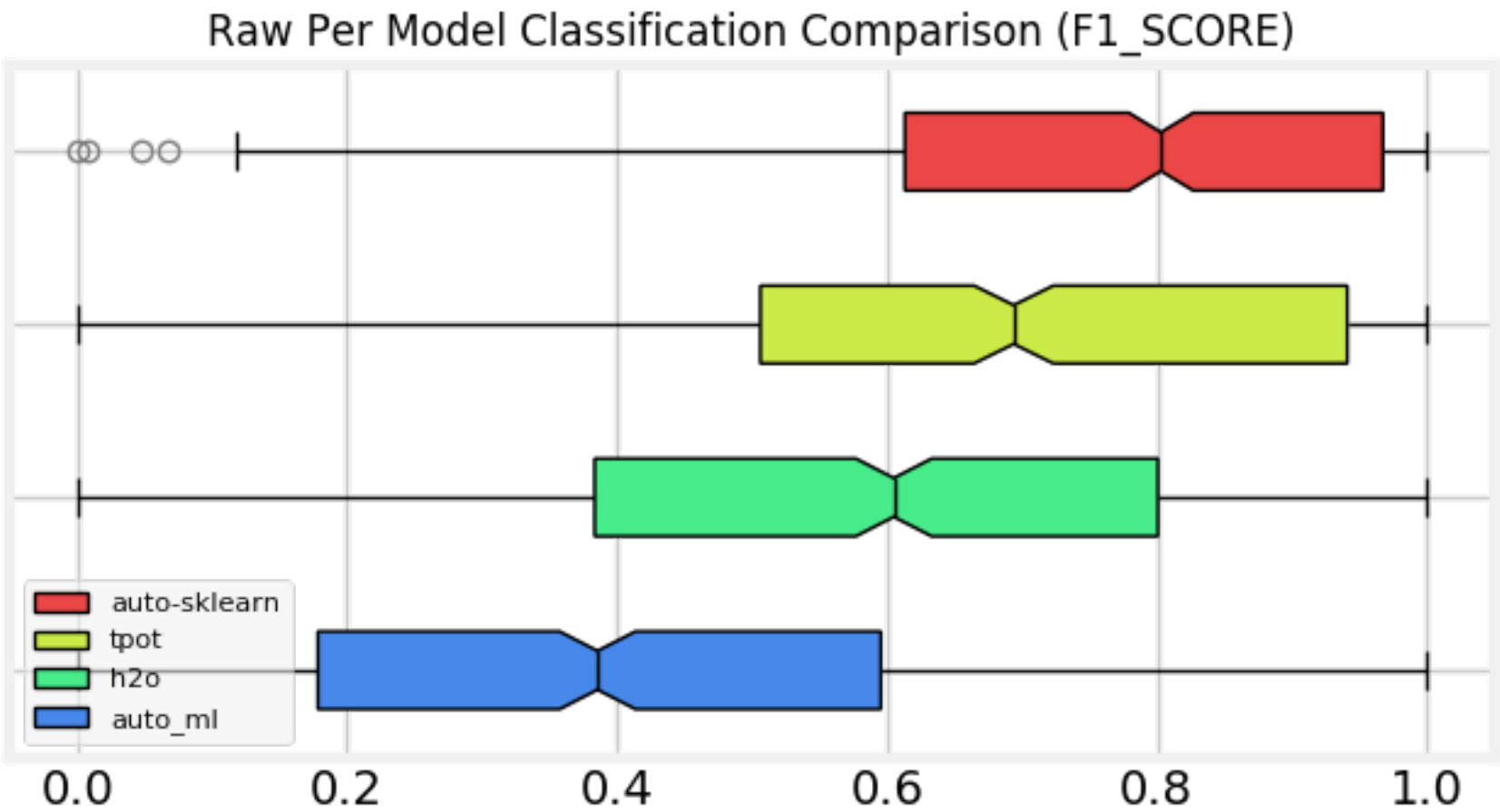


Which AutoML Tool?

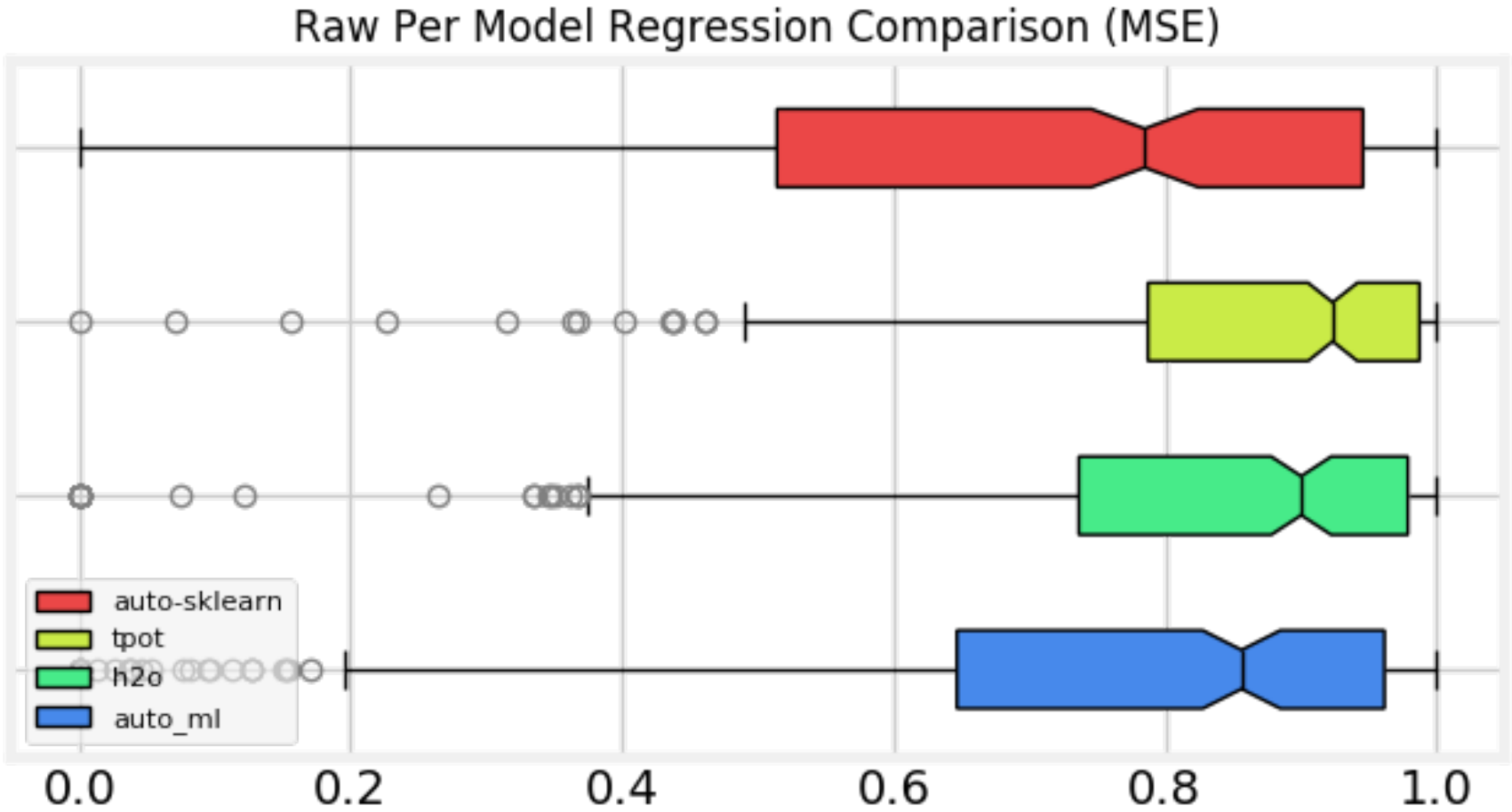
A. Balaji, A. Allen Choosing the best AutoML Framework, 2020
https://medium.com/georgian-impact-blog/choosing-the-best-automl-framework-4f2a90cb1826

- a selection of 87 open datasets, 30 regression and 57 classification, from OpenML
- Comparison between four frameworks: `automl`, `auto-sklearn`, `TPOT`, and `H2O`
- Auto-sklearn performs the best on the classification datasets and TPOT performs the best on regression datasets.

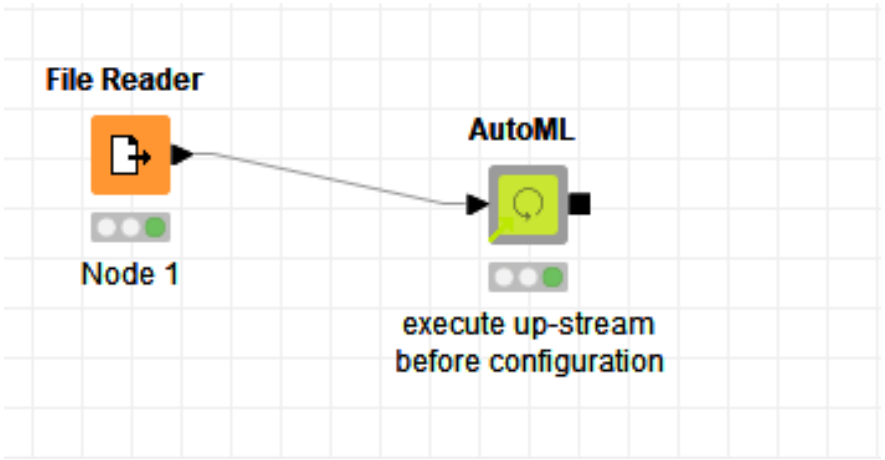
Classification Problems



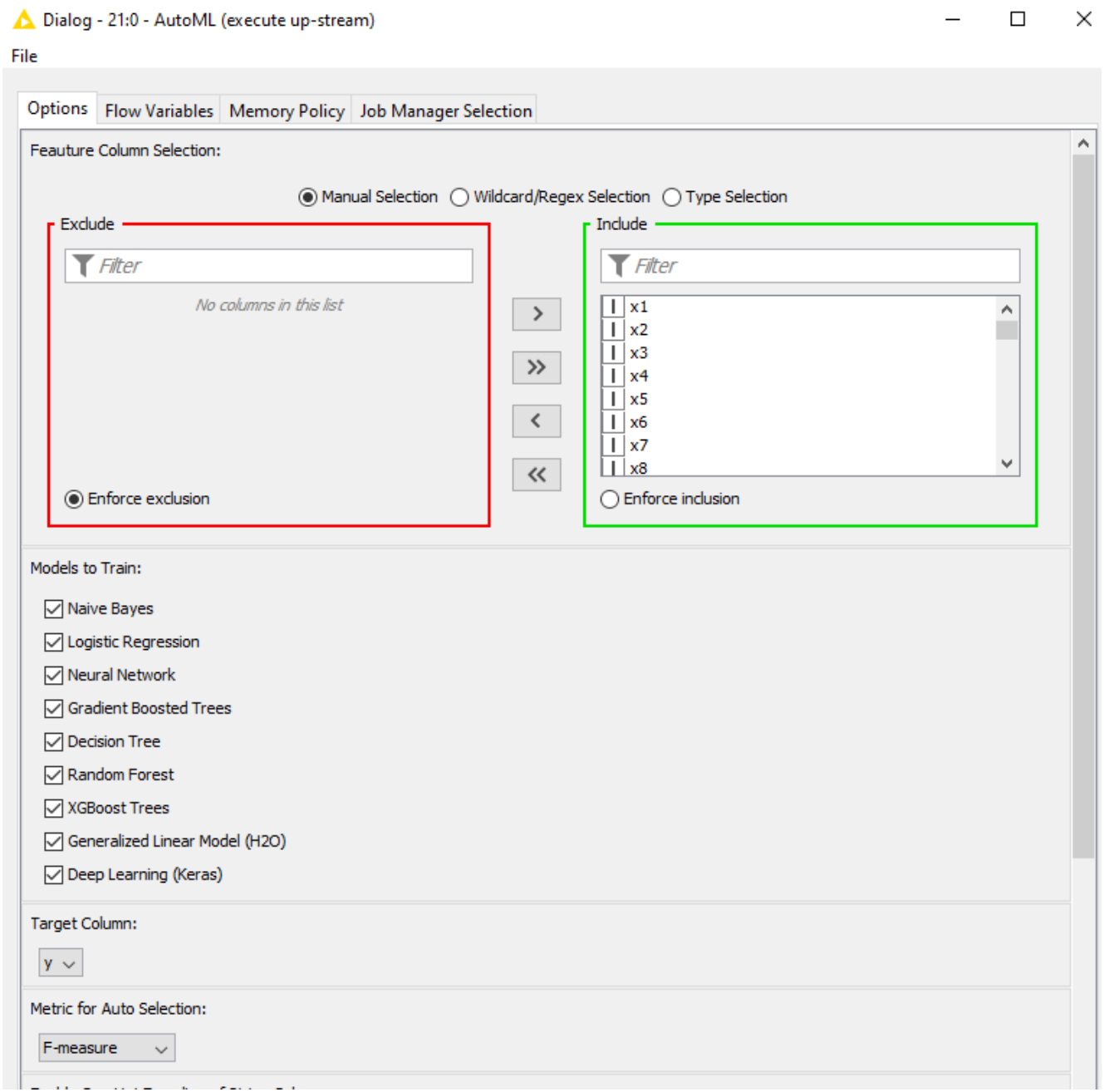
Regression Problems



KNIME - AUTOML



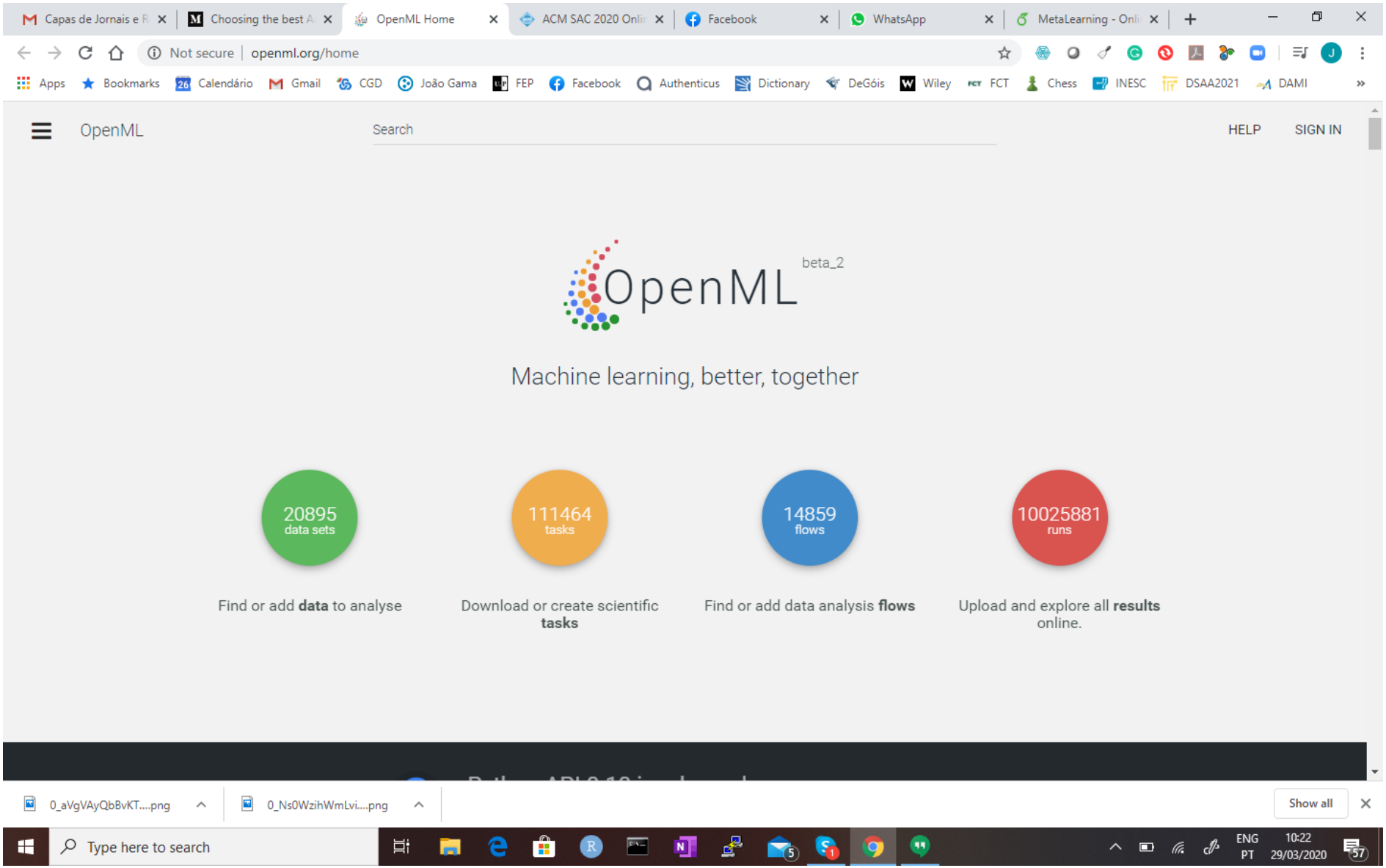
KNIME - AUTOML



KNIME - AUTOML

Outline

OpenML


Capas de Jornais e R
Choosing the best A
OpenML Home
ACM SAC 2020 Onli
Facebook
WhatsApp
MetaLearning - Onli
Not secure
openml.org/home
Apps
Bookmarks
Calendário
Gmail
CGD
João Gama
FEP
Facebook
Authenticus
Dictionary
DeGóis
Wiley
FCT
Chess
INESC
DSAA2021
DAMI
OpenML
Search
HELP
SIGN IN
OpenML
beta_2
Machine learning, better, together
20895
data sets
111464
tasks
14859
flows
10025881
runs
Find or add data to analyse
Download or create scientific tasks
Find or add data analysis flows
Upload and explore all results online.
0_aVgVAyQbBvKT....png
0_Ns0WzihWmLvi....png
Show all
Type here to search
ENG
PT
10:22
29/03/2020

OpenML:Democratizing Machine Learning

OpenML manifesto:

As machine learning is enhancing our ability to understand nature and build a better future, it is crucial that we make it transparent and easily accessible to everyone in research, education and industry.

The Open Machine Learning project is an inclusive movement to build an open, organized, online ecosystem for machine learning.

We build open source tools to discover (and share) open data from any domain, easily draw them into your favourite machine learning environments, quickly build models alongside (and together with) thousands of other data scientists, analyse your results against the state of the art, and even get automatic advice on how to build better models.

Stand on the shoulders of giants and make the world a better place.

Outline

Links

- OpenML: https://www.openml.org/
- https://github.com/rhiever/tpot
- https://cran.r-project.org/web/packages/automl/index.html
- https://www.ml4aad.org/automl/
- https://en.wikipedia.org/wiki/Automated_machine_learning

References

- P.Brazdil, "Meta-Learning", slides
- P.Brazdil, C.Giraud-Carrier, C.Soares, R.Vilalta: Metalearning: Applications to Data Mining, Springer, 2009
- K.Smith-Miles: Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection, ACM Computing Surveys, 2008
- H Jin, Q Song, and X Hu. Auto-keras: An efficient neural architecture search system; ACM SIGKDD, 2019
- Slides AUTOML 101, by Sri Krishnamurthy, Quantun University