Cite as: Ioannis Tsamardinos, Vincenzo Lagani, Automated Machine Learning and Knowledge Discovery, ECCB 2018 Tutorial

Automated Machine Learning and Knowledge Discovery

IOANNIS TSAMARDINOS

PROFESSOR, CSD, UNIVERSITY OF CRETE

GNOSIS DATA ANALYSIS, CO-FOUNDER

VINCENZO LAGANI

ILIA STATE UNIVERSITY

GNOSIS DATA ANALYSIS, CO-FOUNDER

Outline

- **Part I (45')**
  - Introduction to the problem and the tutorial
  - Estimation of performance (single configuration)
- **Part II (45')**
  - Estimation of performance (multiple configurations)
  - Incorporating User Preferences
- **Part III (45')**
  - Feature Selection and Knowledge Discovery
  - Hyper-parameter search strategies
  - Feature construction, preprocessing, imputation, transformations
- **Part IV (45')**
  - **Post-analysis interpretation and visualizations**
  - **AI-assisted Auto-ML (algorithm selection, pipeline synthesis, meta-learning, feature learning)**
  - **Putting all together – The Just Add Data Bio platform**
  - **Tools for Auto-ML**

Post-analysis interpretation and visualizations

Why interpreting a predictive model

- Understanding **how** the model operates contributes to a better understanding of the problem (knowledge discovery):
  - What can the effect of each predictor be ? Is it always the same? Or does it changes depending on the values of the other predictors?
  - How can I explain why a specific sample is assigned to a class and not to another?
- Alternative approach: **black-box**
  - Suitable is you are interested exclusively in predictive performances

Effect sizes in linear models

- Simple case: linear model
  - $P(Disease|predictors) = 0.21 \cdot Ikzf1 - 0.78 \cdot Myc + 0.45 \cdot H3k4$
  - The fictional example depicts a linear model where the probability of disease is computed on the basis of the expression of a group of genes

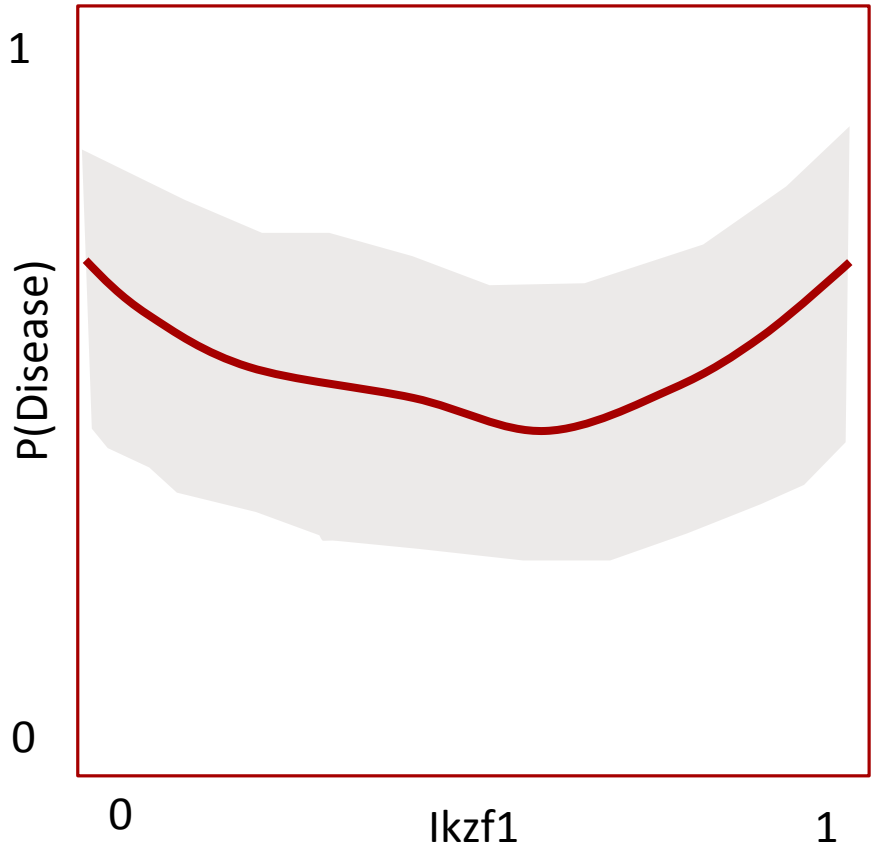
- If the expression data are **all standardized**, then the coefficients of the model correspond to effect sizes
  - Furthermore, the effect sizes are constant, i.e., they do not change depending on the value of the other predictors

Effect sizes in linear models with interactions

- Not so simple case: linear model with interaction
  - $P = 0.21 \cdot Ikzf1 - 0.78 \cdot Myc + 0.45 \cdot H3k4 + 0.18 \cdot Ikzf1 \cdot Myc$
- Adding an interaction term implies that the effect of IKZF1 and Myc is not constant anymore
- IKZF1 and Myc now depends on each other value

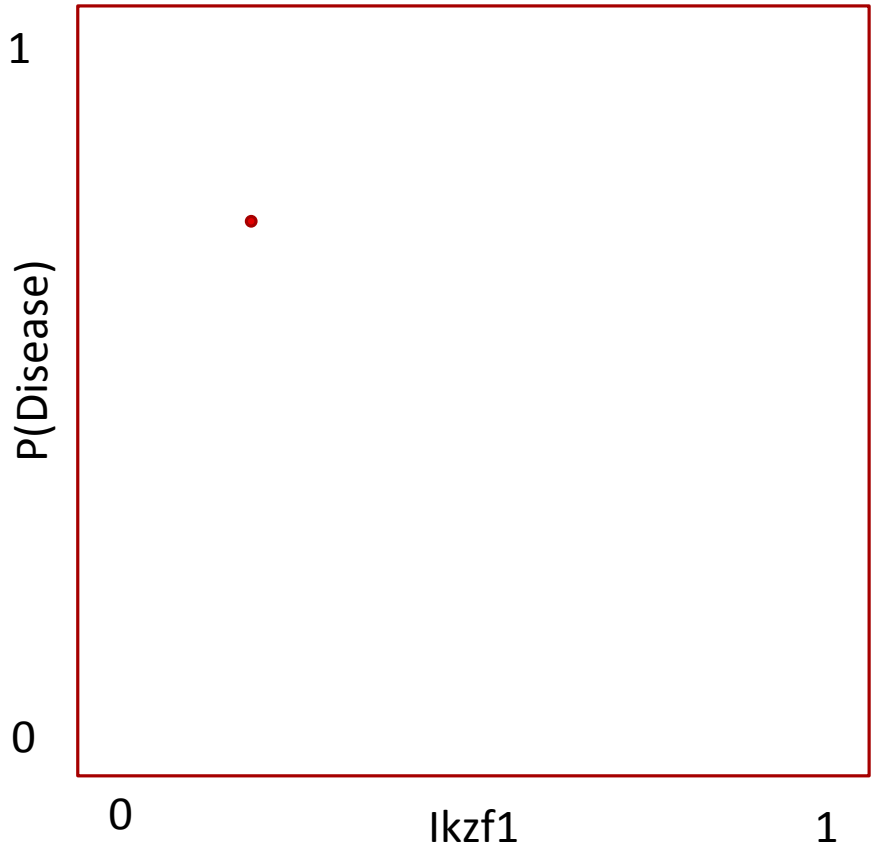
ICE plots: visualizing effect sizes in general models

- Individual Conditional Expectation (ICE, Goldstein at al. 2015) plots allow to visualize the effect of predictors in any type of model:
  - $P(Disease|predictors) = f(Myc, Ikzf1, H3k4)$
- The solid line corresponds to the average effect of Ikzf1 on the probability of disease
  - Confidence interval as shaded area



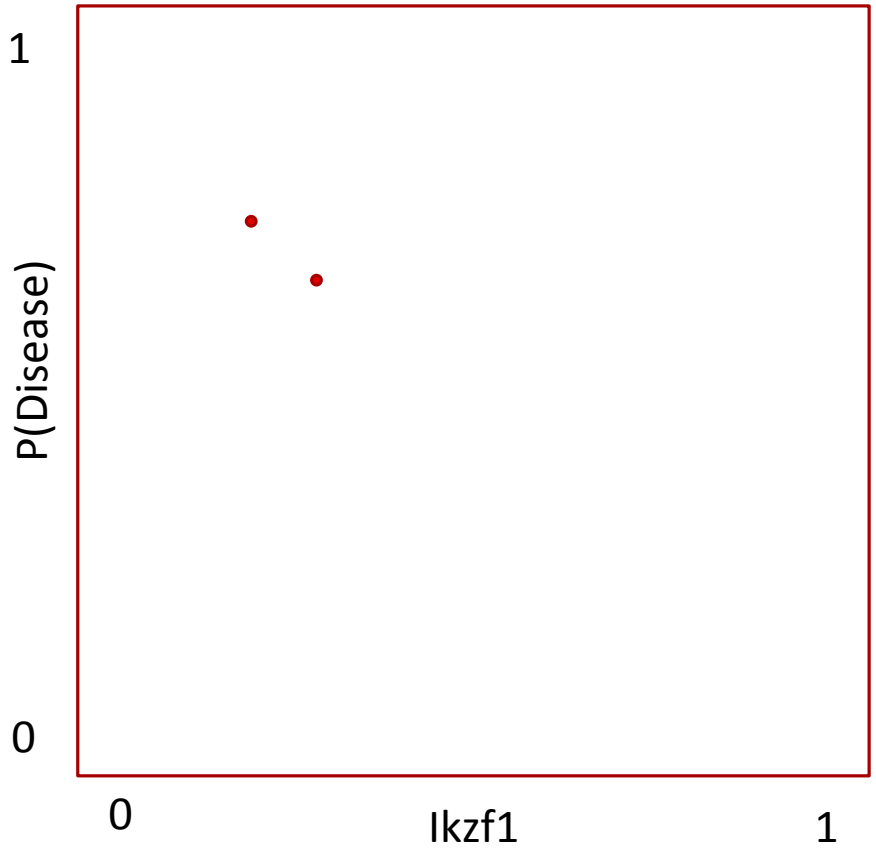
ICE plots: visualizing effect sizes in general models

- Let us assume we have a specific sample, S1, with
  - Ikzf1 = 0.38,
  - Myc = 0.26
  - H3k4 = 0.56
  - f(Ikzf1, Myc, H3k4) = 0.7
- The sample would correspond to the **red** point in the graph



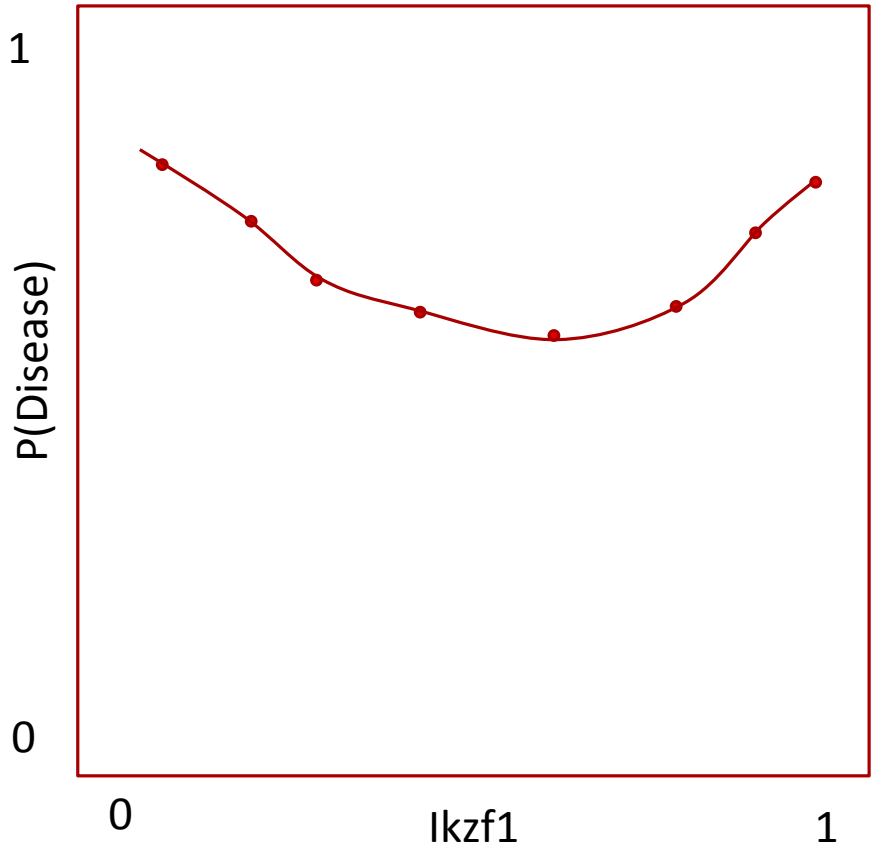
ICE plots: visualizing effect sizes in general models

- We now change the value of Ikzf1, leaving Myc and H3k4 unchanged:
  - Ikzf1 = **0.45**,
  - Myc = 0.26
  - H3k4 = 0.56
  - f(Ikzf1, Myc, H3k4) = **0.6**
- The new fictional sample would correspond to the second red point



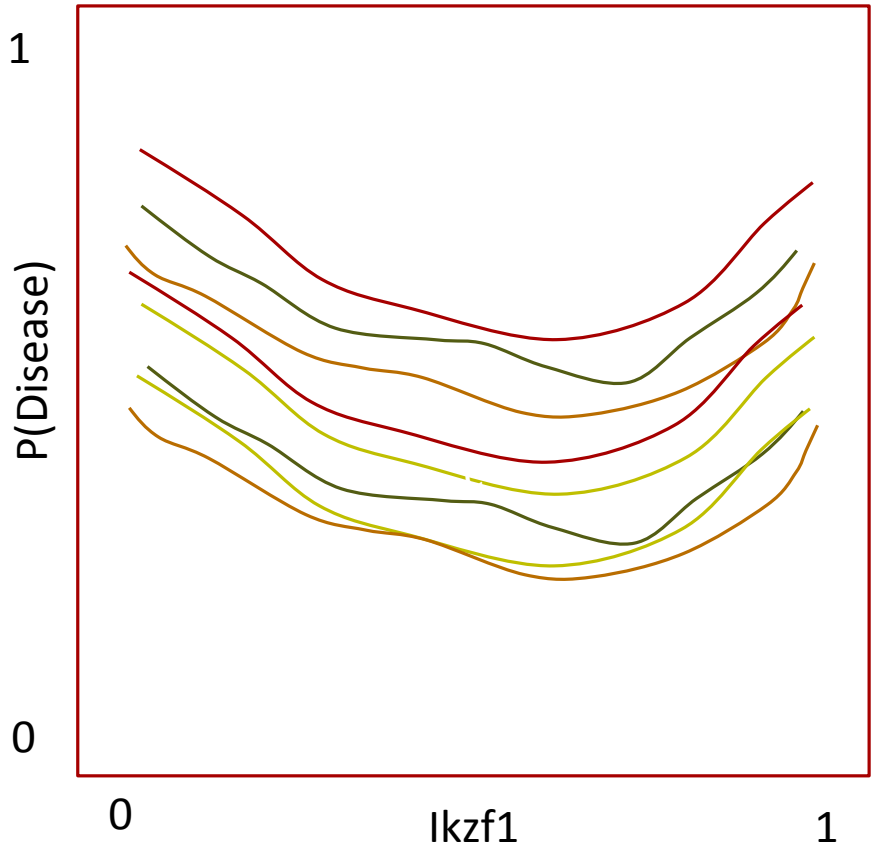
ICE plots: visualizing effect sizes in general models

- Changing the Ikzf1value several times allow to plot a curve representing Ikzf1 **effect on the probability of disease** for sample S1



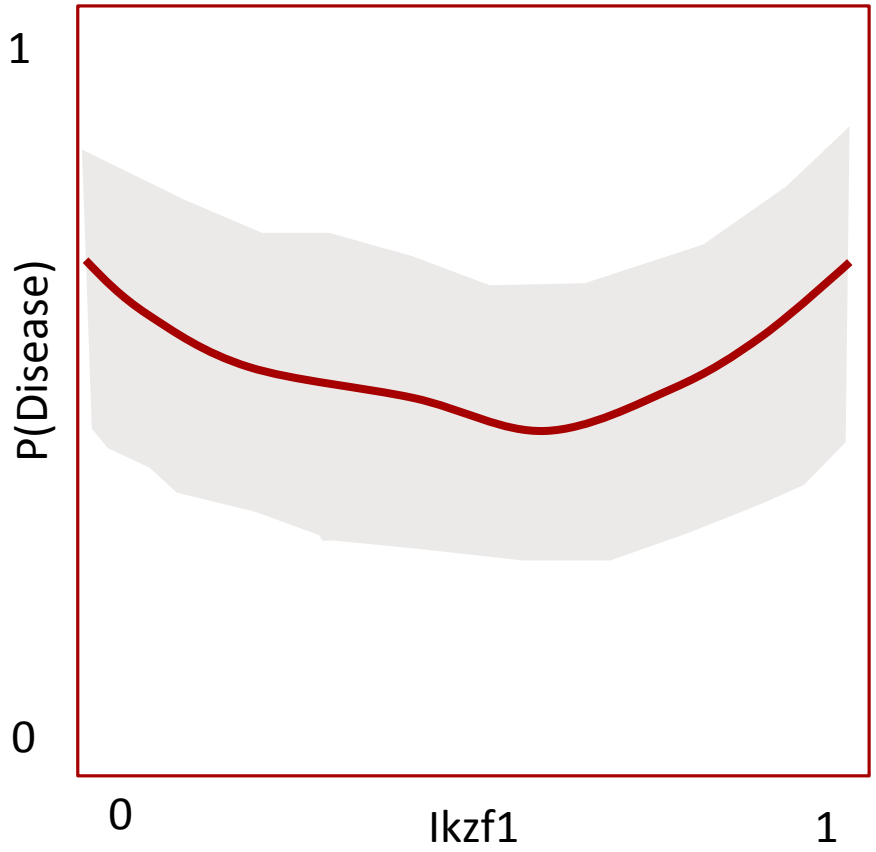
ICE plots: visualizing effect sizes in general models

- Changing the Ikzf1value several times allow to plot a curve representing Ikzf1 effect on the probability of disease for sample S1
- Repeating the same procedure for all other samples produce a distribution of effect-size curves



ICE plots: visualizing effect sizes in general models

- The final ICE plot is produced by:
  - computing an average line out of the sample-specific curves
  - computing confidence intervals
- These plots allow to detect and represent **non-linear dependencies** between predictors and outcome



Single prediction explanation

- Question: which predictor influenced the most the prediction on a specific sample?
  - Sample S1: <Ikzf1, Myc, H3k4> = <0.83, 0.11, 0.31>
  - Trivial answer for linear models: the predictor corresponding to the largest monomial in absolute value
  - $P(Disease|predictors) = 0.21 \cdot 0.83 - 0.78 \cdot 0.11 + 0.45 \cdot 0.31 = \mathbf{0.17} - 0.09 + 0.14 = 0.22$

Ikzf1 has the highest monomial

Leave-One-Covariate-Out (LOCO)

- The LOCO methodology offers a possible solution for non-linear models [Lei at al. 2018]
- Let us assume to have the following dataset, augmented with the predictions ($\hat{Y}$) from our model:

Sample	Ikzf1	Myc	H3k4	Y	$\hat{Y}$
S1	0.20	0.24	0.53	1	0.89
S2	0.69	0.91	0.78	0	0.23
S3	0.43	0.38	0.07	1	0.78

Leave-One-Covariate-Out (LOCO)

- Ikzf1 can be set to zero (or other convenient default value) and the predictions be re-evaluated
- In the example, only the prediction for sample S2 changes considerably

Sample	Ikzf1	Myc	H3k4	Y	$\hat{Y}$	$\hat{Y}_{-Ikzf1}$
S1	**0**	0.24	0.53	1	0.89	**0.82**
S2	**0**	0.91	0.78	0	0.23	**0.65**
S3	**0**	0.38	0.07	1	0.78	**0.73**

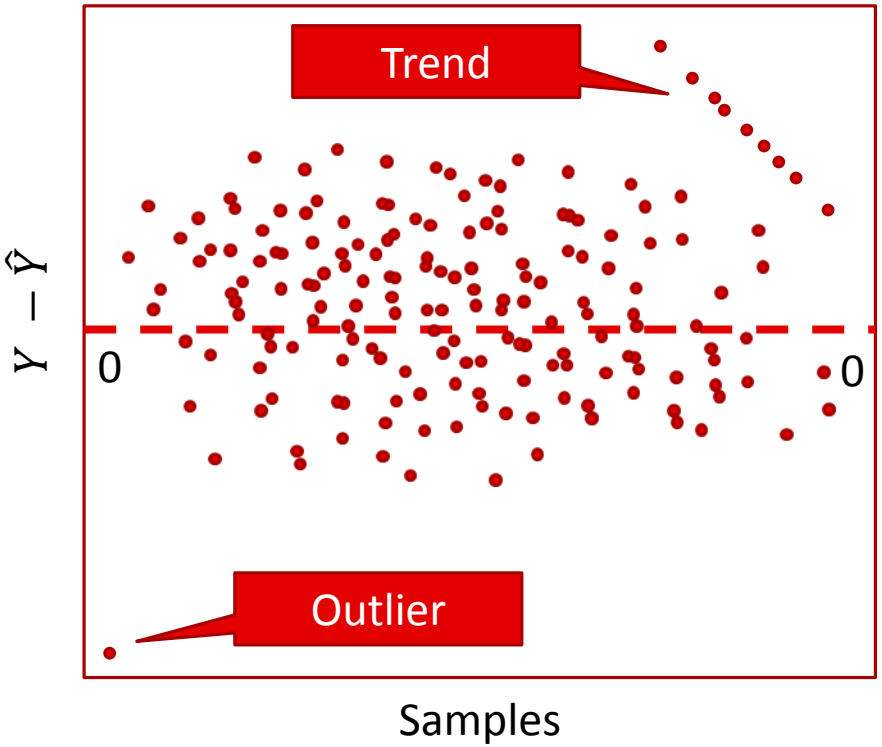
Leave-One-Covariate-Out (LOCO)

- We repeat by leaving out one covariate at the time
- It is evident that the prediction for S1 is particularly sensitive to a change of the Myc predictor, while the S2 prediction is influenced by Ikzf1. The prediction for S3 seems quite stable

Sample	Ikzf1	Myc	H3k4	Y	$\hat{Y}$	$\hat{Y}_{-Ikzf1}$	$\hat{Y}_{-Myc}$	$\hat{Y}_{-H3k4}$
S1	0.20	0.24	0.53	1	0.89	0.82	**0.21**	0.85
S2	0.69	0.91	0.78	0	0.23	**0.65**	0.25	0.22
S3	0.43	0.38	0.07	1	0.78	0.73	0.76	0.77

The old good way: residual inspection

- The difference between the actual and predicted values $Y - \hat{Y}$ should always be assessed
- Linear models require normally distributed residuals
- The presence of any outlier or suspicious trend should be carefully checked



References

- Goldstein, Alex, et al. "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." *Journal of Computational and Graphical Statistics* 24.1 (2015): 44-65.

- Lei, Jing, et al. "Distribution-free predictive inference for regression." *Journal of the American Statistical Association*(2018): 1-18.

AI-assisted Auto-ML

ALGORITHM SELECTION, PIPELINE SYNTHESIS, META-LEVEL LEARNING

ML ⊂ AI

- The terms Machine Learning (ML) and Artificial Intelligence (AI) are progressively more often used as synonym
- AI is actually a wider topic and includes different technologies
- We are interested in AI technologies that can help the data analyst in devising better ML analyses
- Ideally, we would like to have an **AI system smart enough to automatically solve ML tasks**

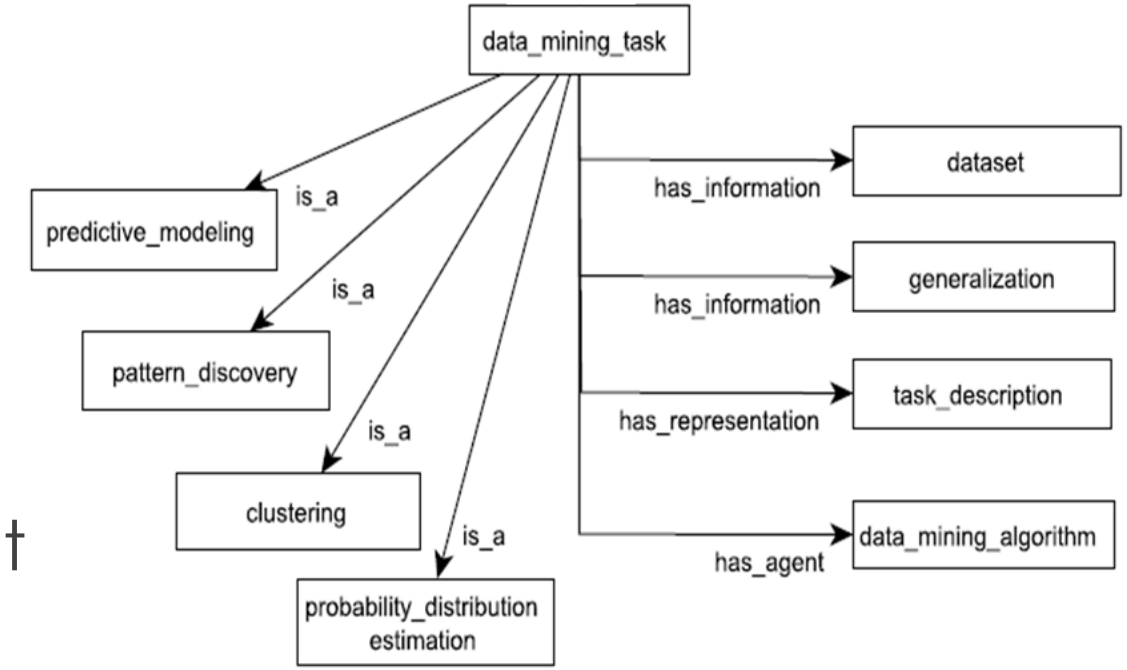
Knowledge-based Artificial Intelligence

- Knowledge-based AI attempts to represent human knowledge in a structured way, namely Knowledge Bases (KB)

- The information contained in a KB is used by inferential engines for automatically inferring new facts.

What is in a Knowledge Base? Ontologies and Rules

- KB are usually composed by **ontologies and rules**
- **Ontologies** represents entities and their relationships
- E.g., "predictive_modeling" is_a "data_mining_task"
- Several formal languages exist for ontologies, e.g., the Web Ontology Language (OWL, https://www.w3.org/OWL/)



Adapted from Panov et al., 2008

What is in a Knowledge Base? Ontologies and Rules

- **Rules** can be added to a KB in order to increase the deductive reasoning capabilities of the ontology

- "IF the data mining task is predictive modelling AND the dataset is high dimensional, THEN use a linear SVM classifier"

- The Semantic Web Rule Language (SWRL) is one of the languages used for encoding rules in KBs (https://www.w3.org/Submission/SWRL/); different languages offer varying degrees of expressiveness and analyzability

How to use a Knowledge Base? Populating and Querying

- Once entities are defined in a ontology, it is possible to specify exact **instances**
  - E.g., for the entity "dataset" and its attribute "sample_size" and "feature_size", we may want to specify instances like <myCyTOFData, 20000, 35> and <myNGSData, 120, 40000>
  - Similarly, we may want to indicate the instances <RandomForest> and <SVM> for the entity "classifier"

How to use a Knowledge Base? Populating and Querying

- A populated KB can be analyzed by an inferential engine for answering queries asked by the user

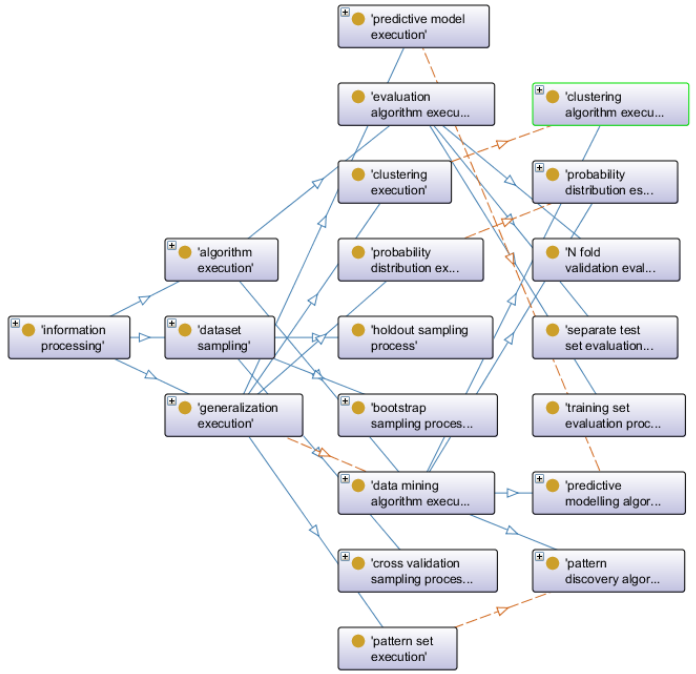
- Example: find all classifiers that are compatible with myCyTOFData dataset and that produce interpretable models
  - SPARQL (https://www.w3.org/TR/rdf-sparql-query/) is one of the most common languages for encoding queries

- Queries are the most useful feature of KBs, allowing to infer non-trivial facts through deductive logic

Existing ontologies for ML and data mining

- **Several ontologies** for ML have been proposed over the years, no formal consensus has been reached yet

- KD Ontology [Žáková et al. 2010]
- KDDONTO Ontology [Diamantini et al. 2009]
- DMWF Ontology [Kietz et al. 2009]
- DMOP Ontology [Hilario et al. 2009]
- OntoDM [Panov et al. 2008]



Part of the OntoDM-core ontology

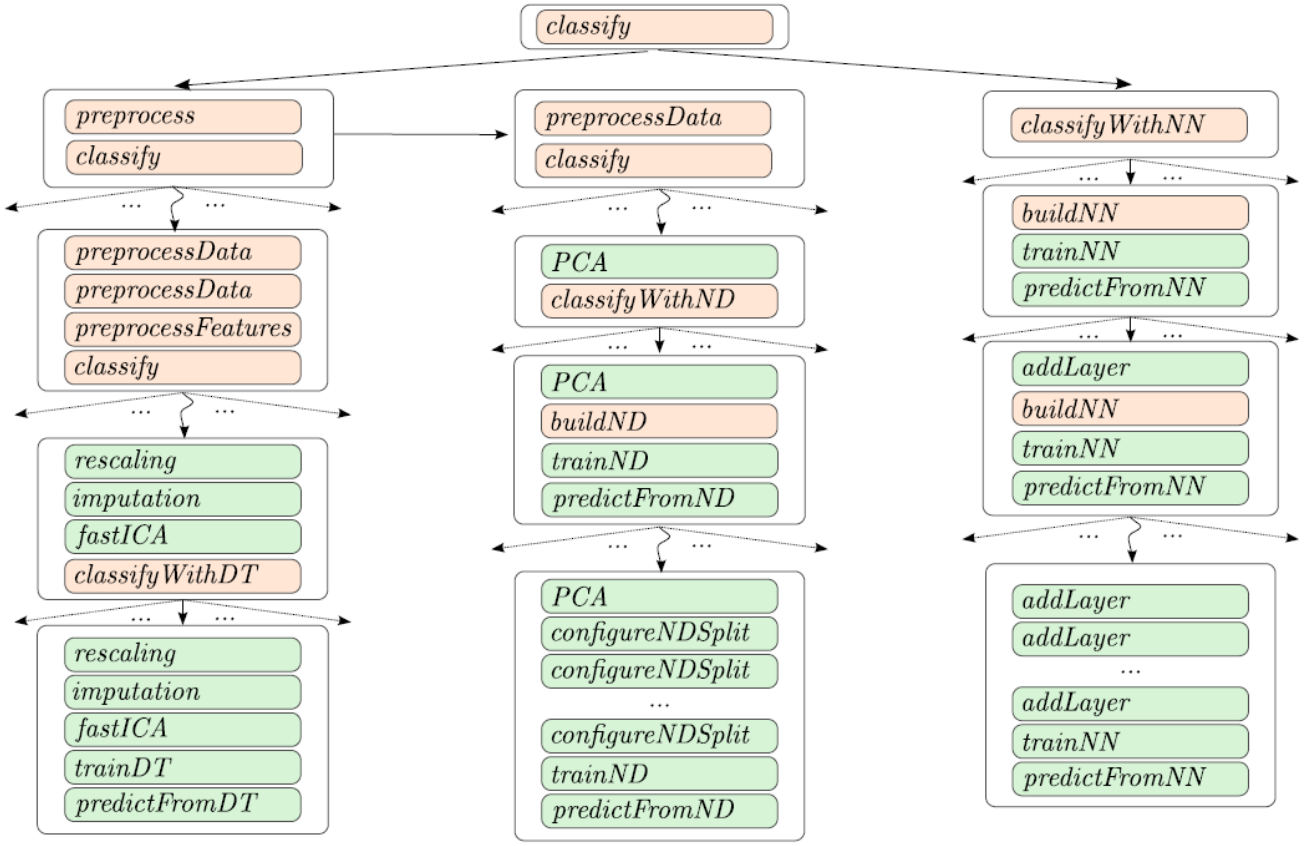
Beyond querying: planning the whole ML workflow

- Final goal of AI-assisted ML: identifying the complete set of steps (a.k.a. **workflow**) needed for analyzing the data at hand

- Special inferential engines are needed, able to take into account precedence constraints
  - e.g., data normalization should be performed before classification

Example of workflow planning

- Left: a pipeline that pre-processes data with rescaling, imputation, and features are fast ICA before using a decision tree for prediction.
- Middle: the data are transformed with PCA before prediction with nested dichotomy
- Right: no pre-processing, neural networks used for prediction
- **Each set** of arrows indicate points where **alternative choices** can take place



Adapted from Mohr et al., 2018

Works on planning for auto-ML

- eProPlan: an ontology-based AI planner for ML workflows, based on the DMWF ontology [Kietz et al. 2010][Kietz et al. 2012]
- The forward chaining planning algorithm based on the KD Ontology [Žáková et al 2011]
- Workflow optimization based on ontology and meta-mining [Hilario et al. 2011]
- ML-Plan: a system using hierarchical task networks for identifying the best ML workflow [Mohr et al., 2018]

ML to improve ML analyses

- Large data analysis centers or data analytics services perform **thousands of ML tasks a day**
- It makes sense to register as many information on each task as possible, for example:

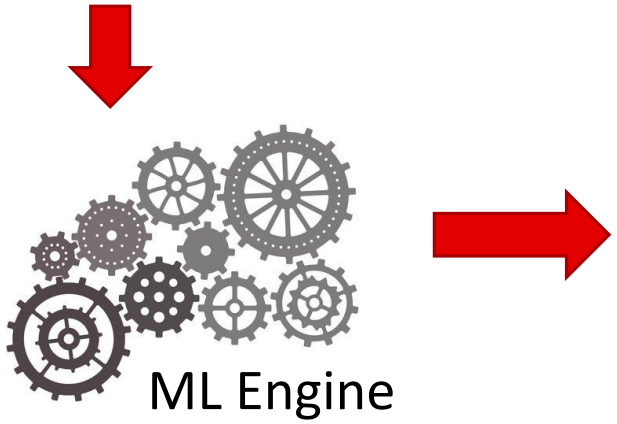
Info about dataset | Info about the analysis | Info about performances

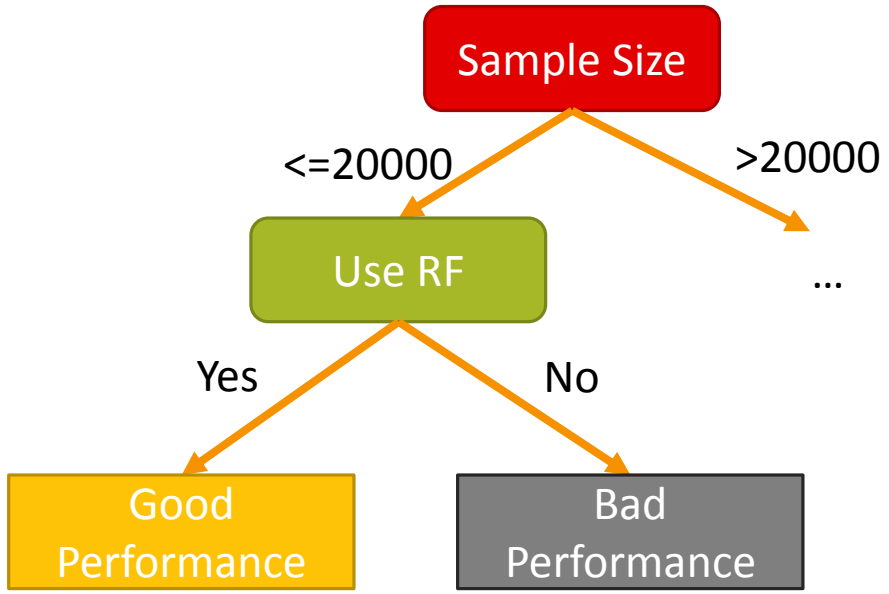
Task	# samples	# features	Pre-processing	Feature selection	Classifier	AUC	Accuracy
1	12879	452	None	No	SVM	0.91	0.89
2	235	12000	Normalize	Yes	RF	0.87	0.67
…	…	…	…	…	…	…	…

ML to improve ML analyses

- **Meta-learning or Meta-Level learning**: applying ML for predicting which method/protocol/workflow will likely lead to the best model

Task	# samples	# features	Pre-processing	Feature selection	Classifier	AUC	Accuracy
1	12879	452	None	No	SVM	0.91	0.89
2	235	12000	Normalize	Yes	RF	0.87	0.67
...	...	...	...	...	...	...	...





Works on meta-level learning for auto-ML

- Meta-learning for clustering algorithms [De Souto et al. 2008][Ferrari et al. 2015]
- Meta-learning based on mining rules [Nascimento et al. 2009]
- Cloud-based meta-learning system for biomedical data [Vukićević et al. 2014]

Availability AI assisted ML tools

- **No off-the-shelf tool offers KB- or planning-based solutions for ML in a user-friendly way**
  - Exception: the IDA plugin for the RapidMiner platform (last updated in 2012) [Kietz et al. 2012]
- Several ML ontologies are available, however their use require significant experience
  - http://www.e-lico.eu/dmwf.html
  - http://www.e-lico.eu/DMOP.html
  - http://www.ontodm.com/doku.php?id=ontodm-core

References

- M. Žáková, P. Kremen, F. Zelezny, N. Lavrac, Automating knowledge discovery workflow composition through ontology-based planning, IEEE Trans. Autom. Sci. Eng. 8 (2) (2010) 253–264.
- C. Diamantini, D. Potena, E. Storti, Kddonto: An ontology for discovery and composition of kdd algorithms, in: Third Generation Data Mining: Towards Service-Oriented Knowledge Discovery, SoKD'09, 2009, pp. 13–24.
- J. Kietz, F. Serban, A. Bernstein, S. Fischer, Towards cooperative planning of data mining workflows, in: Proceedings of the Third Generation Data Mining Workshop at the 2009 European Conference on Machine Learning, ECML 2009, 2009, pp. 1–12
- M. Hilario, A. Kalousis, P. Nguyen, A. Woznica, A data mining ontology for algorithm selection and meta-mining, in: Proceedings of the ECML/PKDD09 Workshop on 3rd Generation Data Mining, SoKD-09, 2009, pp. 76–87.
- P. Panov, S. Dzeroski, L.N. Soldatova, Ontodm: An ontology of data mining, in: Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on, IEEE, 2008, pp. 752–760.

References

- Mohr, Felix, Marcel Wever, and Eyke Hüllermeier. "ML-Plan: Automated machine learning via hierarchical planning." *Machine Learning* 107.8-10 (2018): 1495-1515.
- Kietz, J U; Serban, F; Bernstein, A (2010). eProPlan: a tool to model automatic generation of data mining workflows. In: 3rd Planning to Learn Workshop (WS9) at ECAI'10, Lisbon, Portugal, 16 August 2010 - 20 August 2010, 15-17.
- Kietz, Jörg-Uwe; Serban, Floarea; Bernstein, Abraham; Fischer, Simon (2012). Designing KDD-Workflows via HTN-Planning for Intelligent Discovery Assistance. In: Planning to Learn 2012, Workshop at ECAI 2012, Montpellier, France, 28 August 2012 - 28 August 2012.
- Záková, Monika, et al. "Automating knowledge discovery workflow composition through ontology-based planning." *IEEE Transactions on Automation Science and Engineering* 8.2 (2011): 253-264.
- Hilario, Melanie, et al. "Ontology-based meta-mining of knowledge discovery workflows." *Meta-learning in computational intelligence*. Springer, Berlin, Heidelberg, 2011. 273-315.

References

- De Souto, Marcilio CP, et al. "Ranking and selecting clustering algorithms using a meta-learning approach." *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008.
- Nascimento, André CA, et al. "Mining rules for the automatic selection process of clustering methods applied to cancer gene expression data." *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2009.
- Vukićević, Milan, et al. "Cloud based metalearning system for predictive modeling of biomedical data." *The Scientific World Journal* 2014 (2014).
- Ferrari, Daniel Gomes, and Leandro Nunes De Castro. "Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods." *Information Sciences* 301 (2015): 181-194.

Putting all together

THE JUST ADD DATA BIO PLATFORM

JAD DEMO in class

Tools for Auto-ML

Auto-ML tools landscape

Open source, academic software

AutoWeka
An Automated Data Mining Software Based on Weka

Auto-Sklearn

TPOT

mlrMBO
mlrHyperopt

DataRobot

H2O.ai

SIGOPT

OptiML / bigml

Commercial tools

Auto-ML tools characterization

On-line service vs. stand-alone
- On line service: remote service accessible through web-based interface
- Stand-alone: software / libraries to use locally

Automation level
- Hyper-parameter optimization
- Additional features: feature construction, visualization

User interface
- GUI: graphical user interface
- Software library: needs programming skills

Academic vs. commerical
- Academic: open-source, free-of-charge for research
- Commercial: requiring subscriptions / payments

Level of customization
- Flexible: users can largely customize the tool operation
- Fixed: no customization options

Academic auto-ML software

- Auto-ML tools developed into the **academia** usually share some common characteristics:

1. They are stand-alone, open source software libraries, requiring advanced programming skills
2. They offer hyper-parameter tuning, but lack other functionalities (visualization, results explanation, etc.)
3. Their operation is largely customizable, provided that the user has the necessary programming and theoretical skills

Main academic auto-ML tools

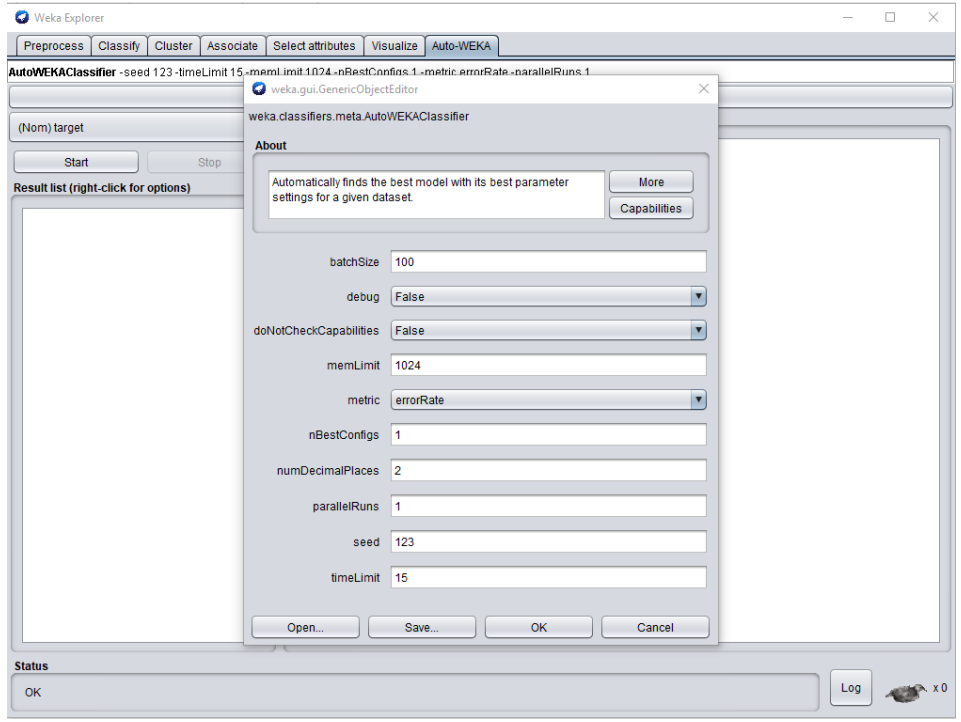
- Python libraries: **auto-sklearn**, **spearmint**, **Hyperopt**, **TPOT**
  - They implement Bayesian Optimization algorithms customized for machine learning. TPOT is an exception, being based on genetic algorithms.

- R libraries: **mlrMBO**, **mlrHyperopt**
  - Similarly to their Python counterparts, these libraries implement Bayesian Optimization approaches specialized for machine learning applications within the R Statistical Software

```
>>> import autosklearn.classification
>>> import sklearn.model_selection
>>> import sklearn.datasets
>>> import sklearn.metrics
>>> 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, random_state=1)
>>> automl = autosklearn.classification.AutoSklearnClassifier()
>>> automl.fit(X_train, y_train)
>>> y_hat = automl.predict(X_test)
>>> print("Accuracy score", sklearn.metrics.accuracy_score(y_test, y_hat))
```

An academic auto-ML tool outlier: AutoWeka

- Hyper-parameter tuning adds-on for the Weka datamining software
- It offers an easy-to-use GUI (no programming skills required)
- Poor level of customization: the user is left with only the choice of how many time and computational resource to assign to the search



Academic auto-ml tools applicability on high-dimensional, biological data

Pros

- Highly customizable systems, can be adapted to the characteristic of different studies (exception: autoweka)
- Part of these tools support parallel computation
- Free, open source

Cons

- focus on hyper-parameter optimization: no support for data preparation or visualization / interpretation of the results
- Default parameters usually not suitable for high dimensional datasets or knowledge discovery (e.g., lack of feature selection)
- Need advanced coding skills

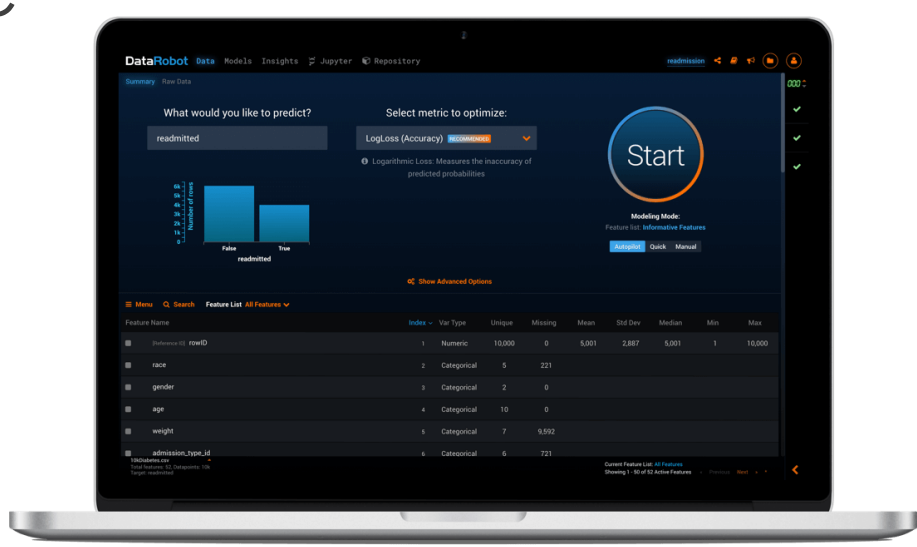
Commercial auto-ml software

- Common traits of commercial auto-ml tools:

1. Optimized for tasks common in industry / retail sectors, with million of samples and relatively few variables (ranging from hundreds to a few thousands)
2. Easy-to-use user interfaces requiring no programming skills
3. Offering several functionalities beyond hyper-parameters tuning, such as feature construction, results inspection and visualization

On-line commercial auto-ML tools

- These services are based on a simple schema:
  - upload data (usually csv format) on external servers
  - indicate preferences (e.g., variable to predict)
  - the service iterates over a number of models searching for the best option
  - A set of results is presented to the users



On-line commercial auto-ML systems

- Most relevant examples:
  - DataRobot
  - bigML
  - IBM Watson Predictive Analytics
  - etc.
- Services largely differentiates on the basis of:
  - type and number of employed algorithms
  - level of customizability for the users
  - presentation of results
  - pricing schema

Other commercial auto-ML systems

- Cloud AutoML from Google
  - Similar to other on-line systems, however to date it only processes Natural Language Text and Images

- H20 Driveless AI
  - AI add-on for the machine learning platform H20
  - Can be installed on local premises
  - Focus on:
    - Automatic Feature Engineering
    - Machine Learning Interpretability

Commercial auto-ml tools applicability on high-dimensional, biological data

Pros

- Friendly user interface
- Additional features ranging from feature constructions to model interpretation and visualization

Cons

- Not suitable for small samples size (< 500 samples)
- Not customized for biology: no interpretation of the results against biological knowledge
- Pricing schema can be an obstacle

Which AutoML Tools are Correct?

Correctness

- What about correct, non-optimistic estimation of performance?
- Which AutoML tools follow correct estimation protocols?
- Work under progress
- Our experience with Auto-Weka follows

Setting up the comparison

- We contrasted AutoWeka and JAD Bio on a chemosensitivity analysis
- Training data from the Cancer Cell Line Encyclopaedia (CCLE)
- Test set from the The Genomics of Drug Sensitivity in Cancer (GDSC)
- Both tools were used with default settings
  - "Quick" configuration for JAD Bio
  - AUC was used as optimization metric for both analysis

The CCLE and GDSC studies

- CCLE [Barretina et al. 2012]
- 24 active compounds
- 1061 cell lines

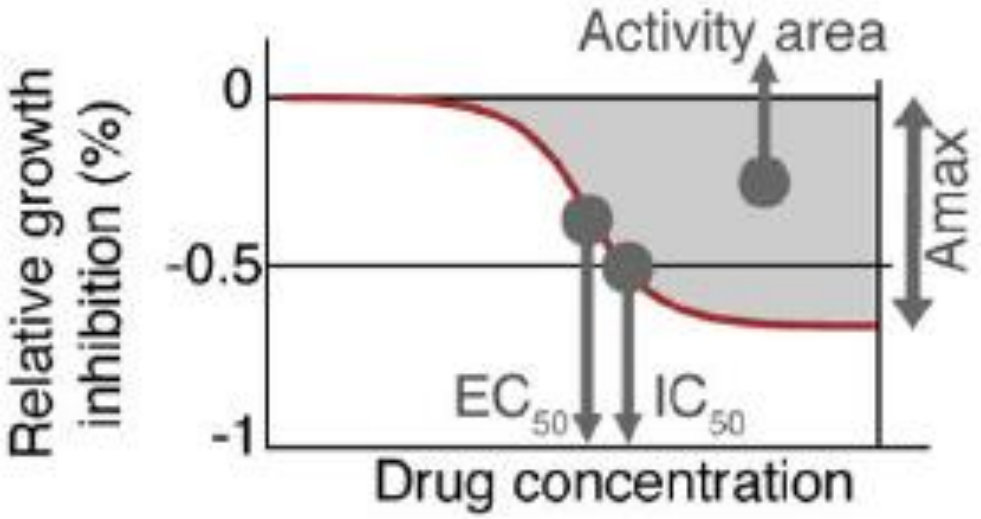
- GDSC [Garnett et al. 2012]
- 140 active compounds
- 1097 cell lines

- 45000+ measurements across
  - Transcriptomics
  - Copy Number Variation
  - Genomic information

We use the data as processed in a subsequent publication by Smirnov et al. 2016

Measuring drug activity

- IC50: drug concentration needed to shrink the tumour by half
- The smaller the IC50, the faster the action of the compound



Results on the GDSC test set

- **JAD Bio** results
  - Estimate on the **training** set: **0.853 AUC with CI [0.77, 0.91]**
  - Estimate on the GDSC **test** set: **0.73 AUC**
- **AutoWeka** results
- Estimate (using cross-validation) on the training set: **0.99 AUC**
- Estimate on the GDSC **test** set: **0.64 AUC**

Unacceptable, misleading estimation

Further testing required to evaluate the extent of this phenomenon

References

- Barretina, J. et al. (2012) The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature, 483, 603–607.
- Garnett, M.J. et al. (2012) Systematic identification of genomic markers of drug sensitivity in cancer cells. Nature, 483, 570–575.
- Smirnov P, Safikhani Z, El-Hachem N, Wang D, She A, Olsen C, Freeman M, Selby H, Gendoo D, Grossman P, Beck A, Aerts H, Lupien M, Haibe-Kains AG, (2016). "PharmacoGx: an R package for analysis of large pharmacogenomic datasets." *Bioinformatics (Oxford, England)*.

End of Part IV