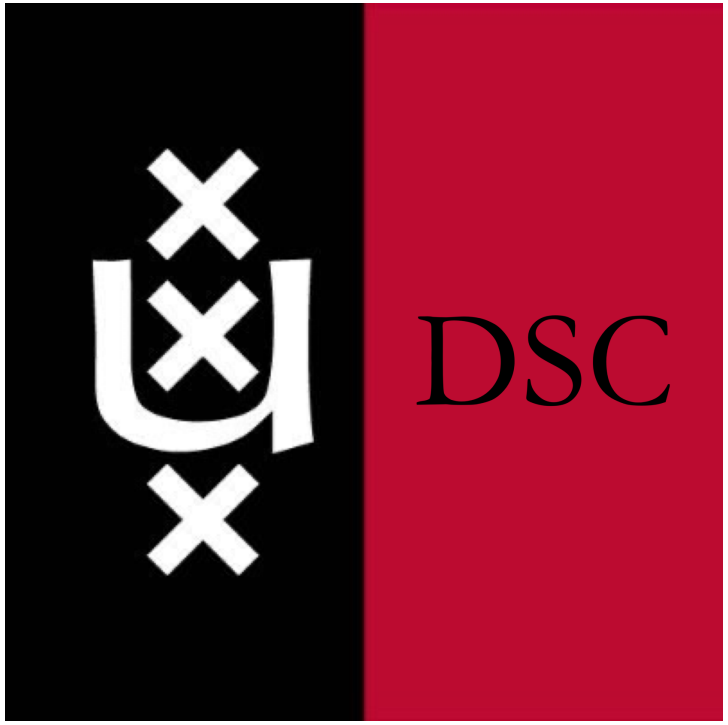


Conventional Machine Learning

20 May 2022 - Marc Galland, support data scientist, SILS

UvA Data Science Center "Data and Coffee"

Definition of Machine Learning

Machine Learning provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Machine Learning @ Vrije Universiteit van Amsterdam: https://mlvu.github.io/

Credit: Peter Bloem

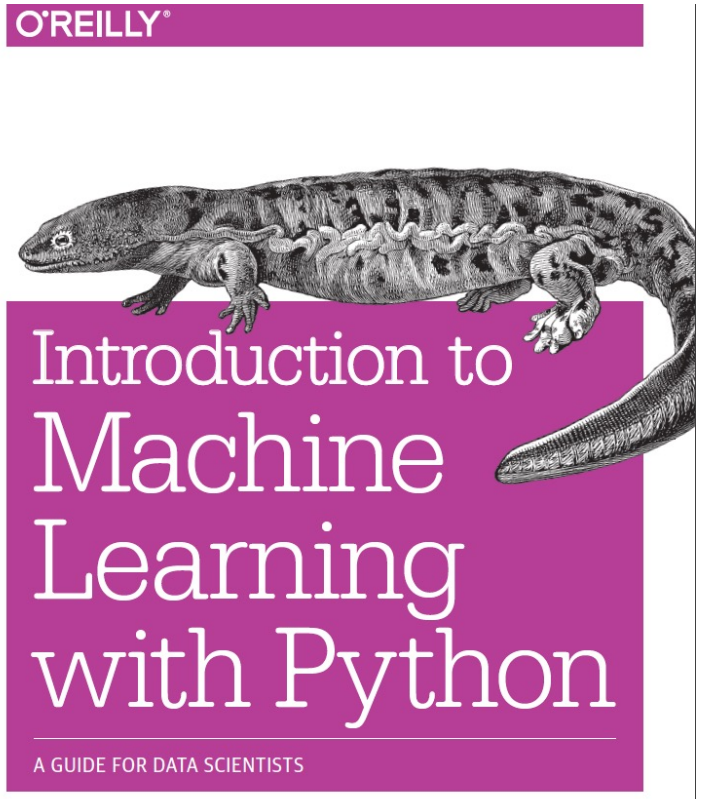
Introducing Machine Learning

"Extracting knowledge from data"

"[at the] intersection of statistics, artificial intelligence and computer science"

Main distinction:

- Supervised learning: inputs with desired outputs
  - *Is a tumor benign given input measurements?*
  - *Should a bank give a mortage given input data?*
- Unsupervised learning: only inputs



Machine Learning @ Vrije Universiteit van Amsterdam: https://mlvu.github.io/

Credit: Peter Bloem

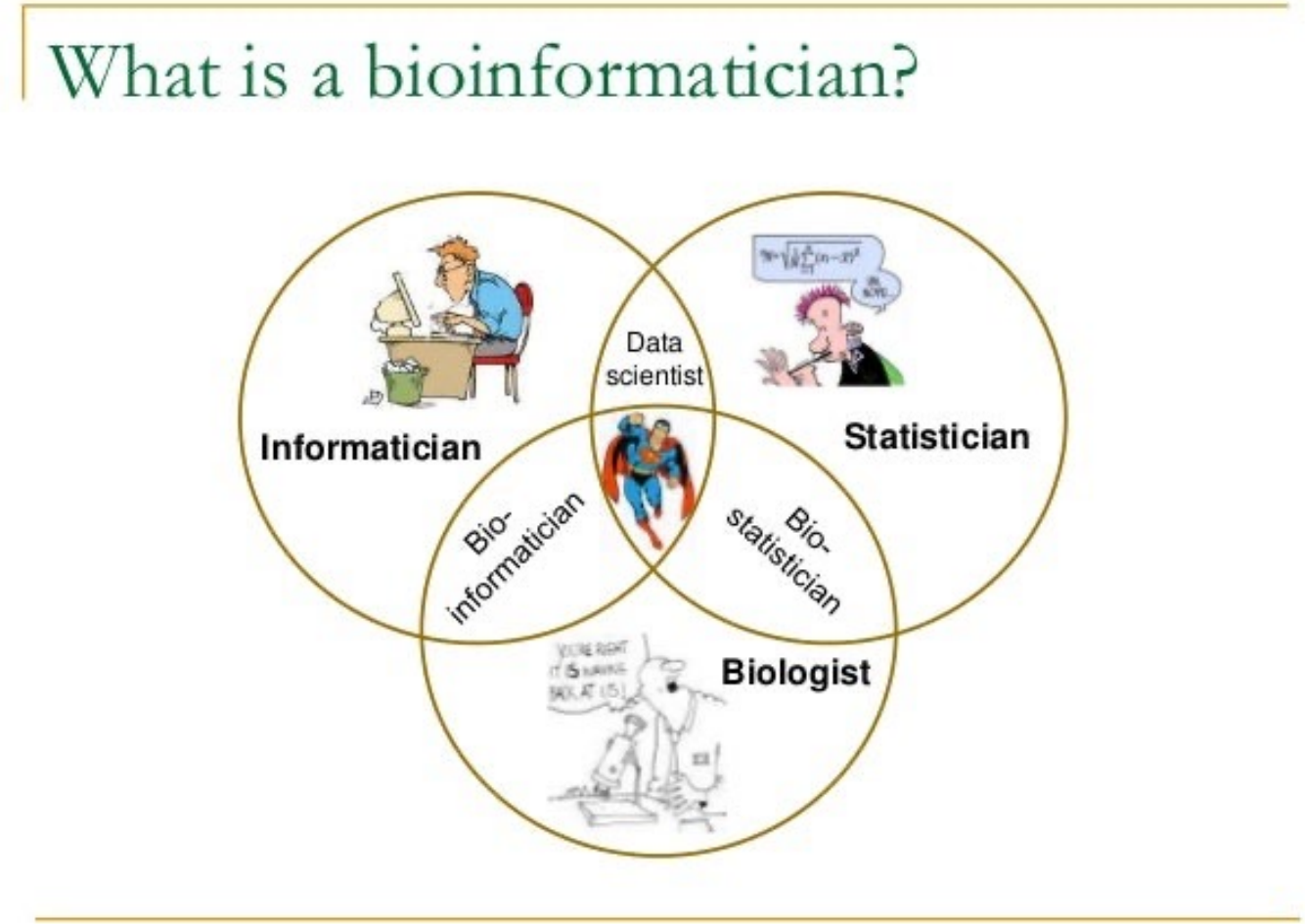
Automated Machine Learning

Machine Learning for real people ☺

Applications of Machine Learning in Biology

Who Am I?

- Wet-lab Plant Biologist since 2009 (PhD)
- Full-time Bioinformatician since 2014
- Aspirant Data Scientist since 2017



Main issues for biologists related to data science

Main issues for biologists related to data science

- **Scientific programming**
  - Learning Python often from scratch
  - Biologists often some command of R… but ML is not as developed compared to Python.

Main issues for biologists related to data science

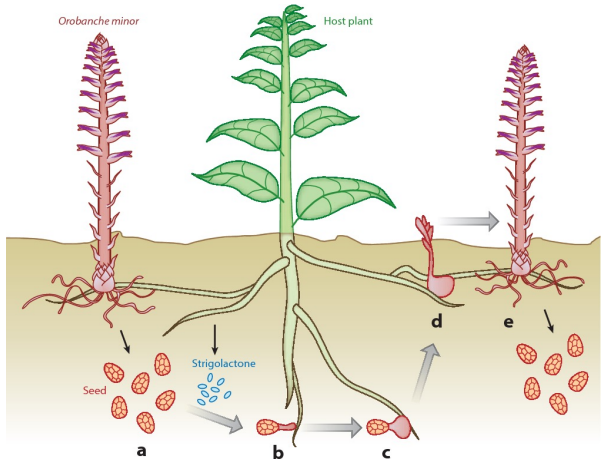
- **Scientific programming**
  - Learning Python often from scratch
  - Biologists often some command of R… but ML is not as developed compared to Python.
- **Machine Learning is a new field to them:**
  - Linear algebra,
  - Statistics,
  - Other type of mathematics.

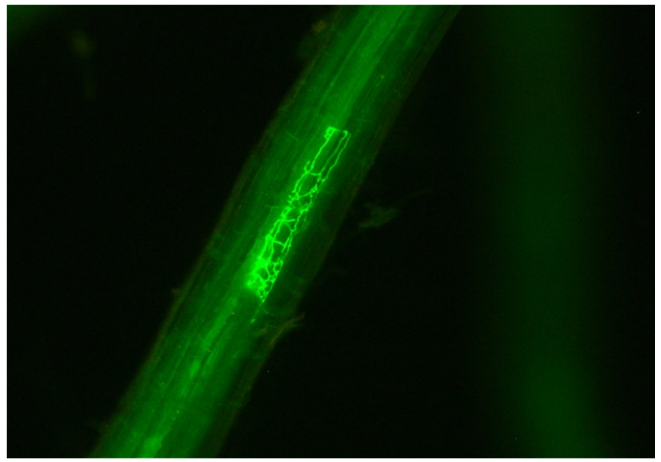
Main issues for biologists related to data science

- **Scientific programming**
  - Learning Python often from scratch
  - Biologists often some command of R… but ML is not as developed compared to Python.
- **Machine Learning is a new field to them:**
  - Linear algebra,
  - Statistics,
  - Other type of mathematics.
- **Interpretation of Machine Learning results:**
  - feature importances, Gini impurity, negative mean squared error : wtf?
  - metric scores: true positives, false negatives, etc.
  - recall, precision, etc.

“Give me a gene to work with”
My work related to ML

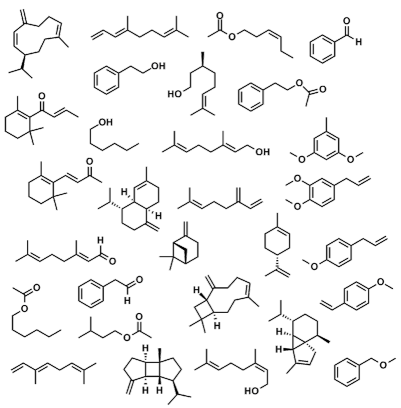
- Close collaboration with plant biologists for research projects
- Identification of relevant features (genes, metabolites) related to a phenotype (e.g. resistance)

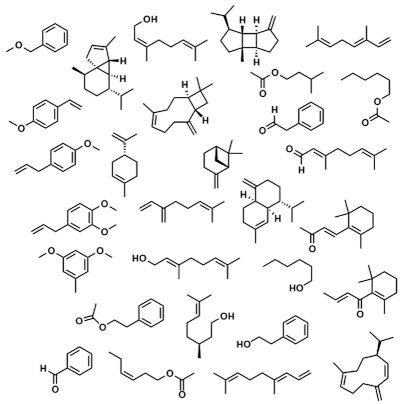


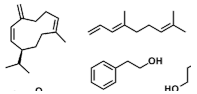


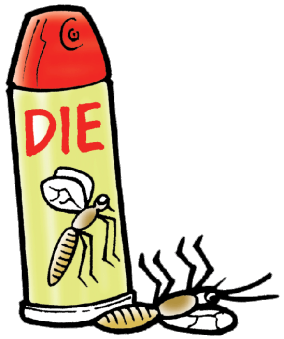
Natural insecticides from wild tomato relatives

breeding



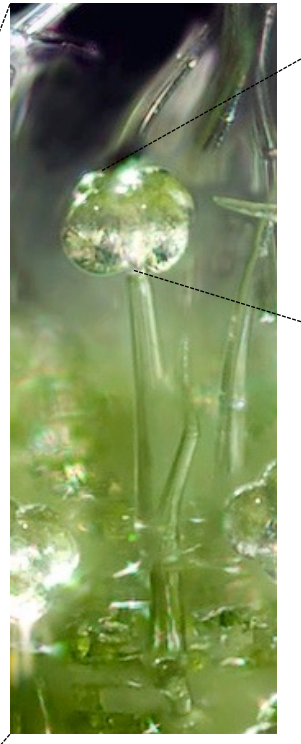


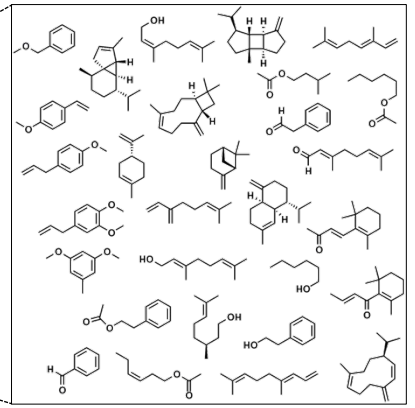


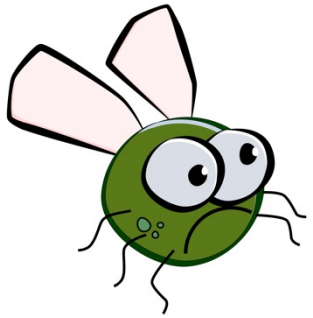


Glandular trichomes from wild tomato are small biochemical factories

Picture: Jan van Arkel







7-epizingiberene

Relating metabolites to whitefly resistance ML (Random Forest) to the rescue

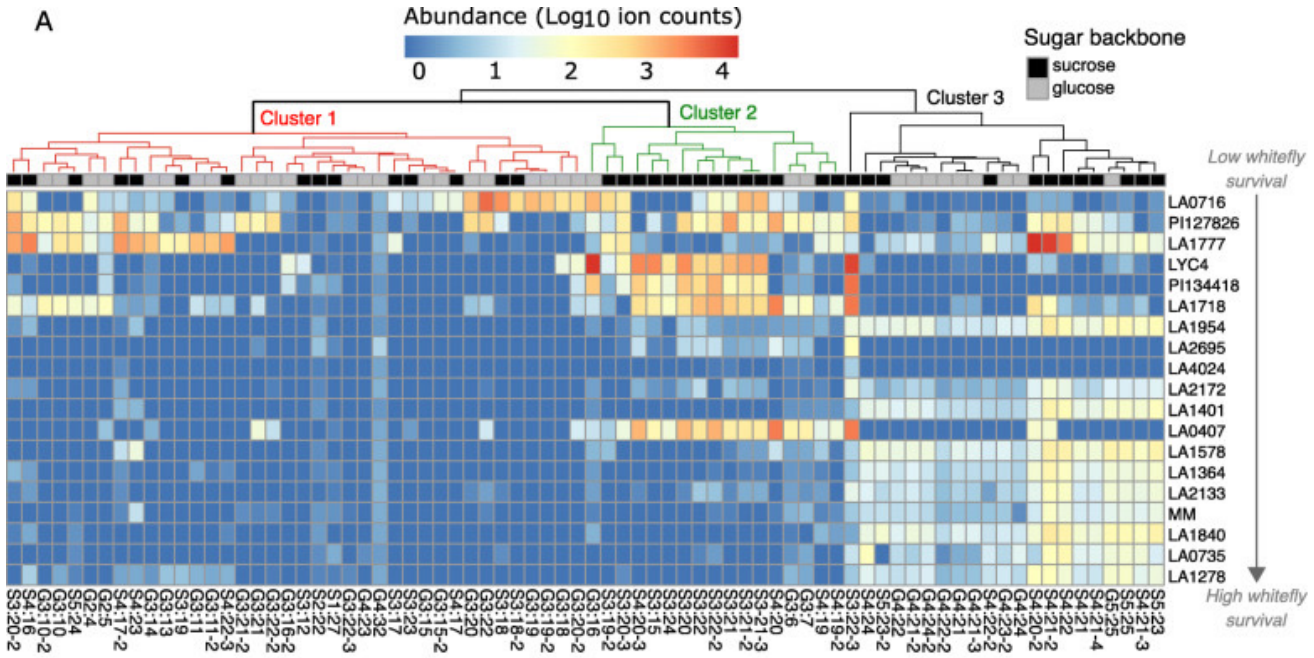


Table 2

Significant metabolites according to the random forest models

Metabolite	m/z	Experimental KI	Theoretical KI	Insect	Classification	*P*-value
S3:15	617 $[M + Na]^+$	–	–	whitefly	resistance	$p < 0.001$
S3:21	715 $[M + Na]^+$	–	–	whitefly	resistance	$p < 0.001$
α-humulene	204.2	1457	1454	whitefly	susceptibility	$p < 0.001$
α-phellandrene[a]	136.1	1000	1002	thrips	susceptibility	$p < 0.001$
α-terpinene[a]	136.1	1013	1017	thrips	susceptibility	$p < 0.001$
β-phellandrene/D-limonene	136.1	1027	1029	thrips	susceptibility	$p < 0.001$

n = 19 tomato samples, p = 76 metabolites (acylsugars)

Kortbeek et al. (2021). *BMC plant biology*, *21*(1), 315.

Can you automate ML implementation?

An introduction to AutoML

Automating ML

To-do list:

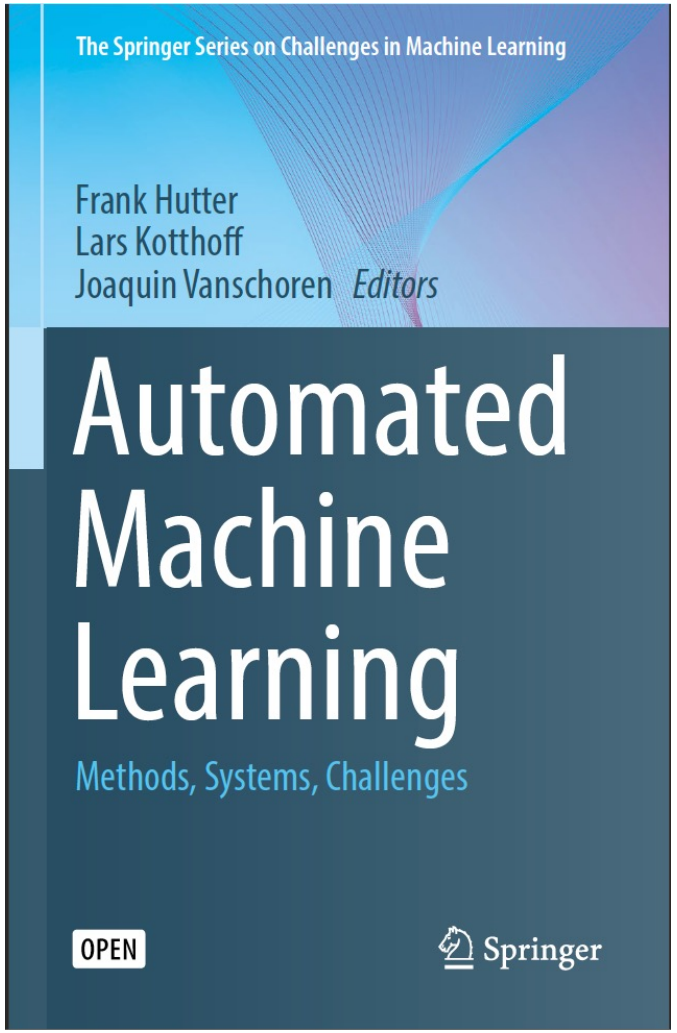
"automating data cleanup and missing data imputation"

"automating feature selection and transformation"

"automating model discovery"

"automating hyperparameter optimization"

"optimally like to automate all of it."



Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_8

Democratization of ML

Thereby, AutoML makes state-of-the-art machine learning approaches **accessible** to domain scientists who are interested in **applying machine learning**

Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_8

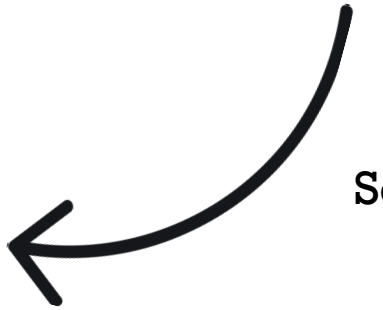
Democratization of ML

Thereby, AutoML makes state-of-the-art machine learning approaches **accessible** to domain scientists who are interested in **applying machine learning** but do not have the resources to learn about the technologies behind it in detail.

Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_8

Democratization of ML

Thereby, AutoML makes state-of-the-art machine learning approaches **accessible** to domain scientists who are interested in **applying machine learning** but do not have the resources to learn about the technologies behind it in detail.



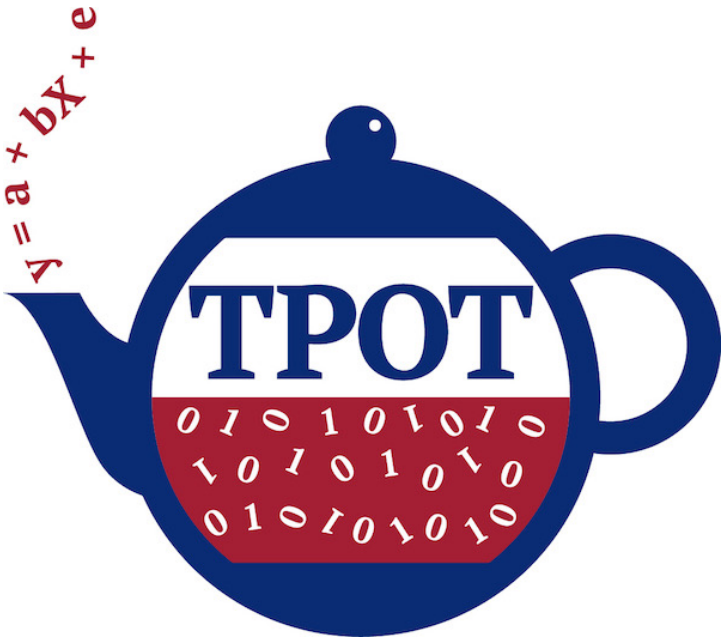
Sounds like me!

Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_8

Implementations in Python

AutoML frameworks

https://mljar.com/automated-machine-learning/

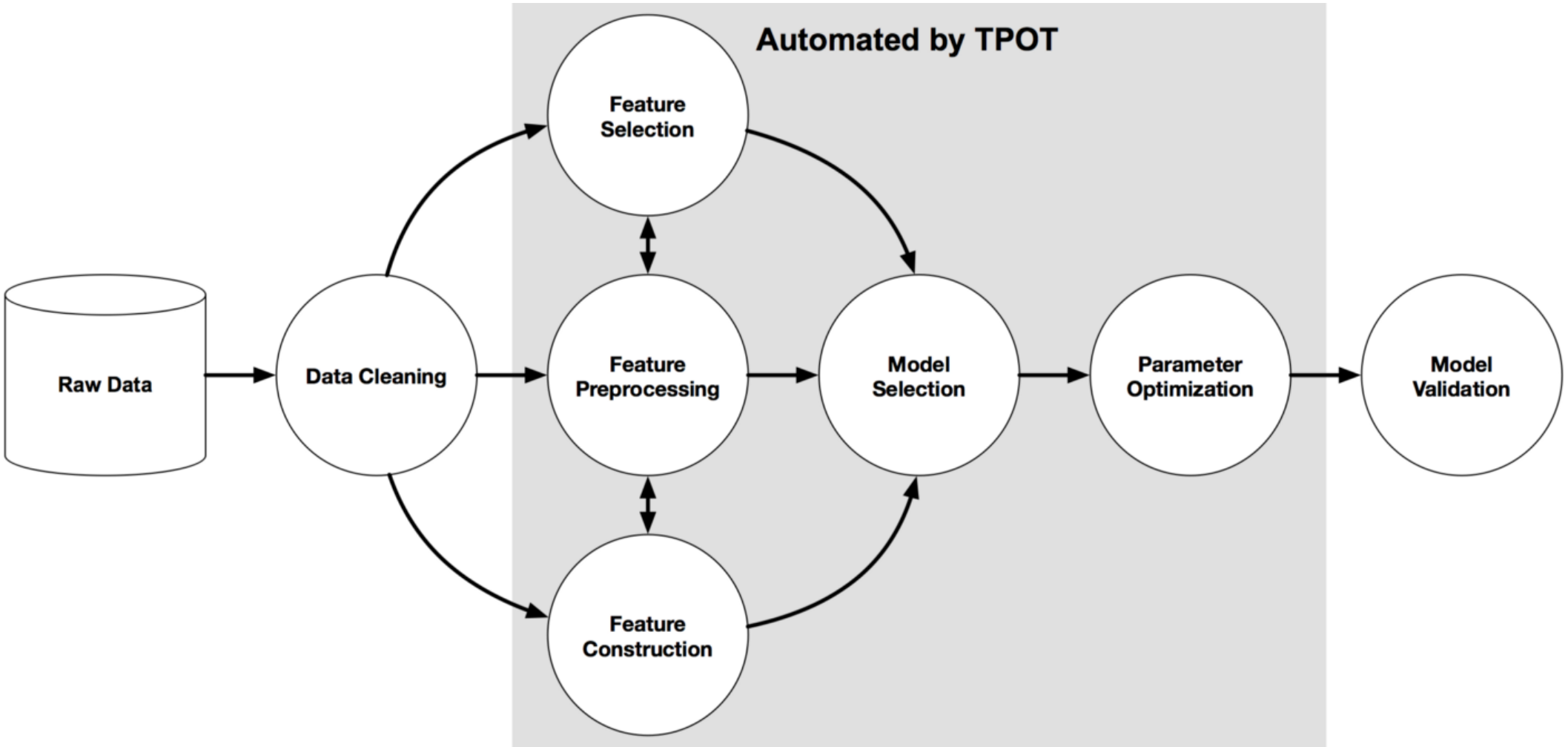


TPOT

OPTIMIZE ML PIPELINES USING GENETIC PROGRAMMING

https://towardsdatascience.com/tpot-automated-machine-learning-in-python-4c063b3e5de9

TPOT workflow



Search space

Examples of models assessed for a classification problem

- Naive Bayes (sklearn.naive_bayes):
  - GaussianNB
  - BernoulliNB
  - MultinomialNB
- Tree (skearn.tree)
  - DecisionTreeClassifier
  - ExtraTreeClassifier
  - RandomForestClassifier
  - GradientBoostingClassifier.
- etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Search space

Examples of models assessed for a classification problem

- Naive Bayes (sklearn.naive_bayes):
  - GaussianNB
  - BernoulliNB
  - MultinomialNB
- Tree (skearn.tree)
  - DecisionTreeClassifier
  - ExtraTreeClassifier
  - RandomForestClassifier
  - GradientBoostingClassifier.
- etc.

In total, for a classification problem TPOT uses:
13 classification models
14 pre-processors
5 feature selection methods

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy
3. Selects top 20 pipelines in the population

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy
3. Selects top 20 pipelines in the population
4. Each of the 20 pipelines generates 5 offspring = 100 pipelines

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy
3. Selects top 20 pipelines in the population
4. Each of the 20 pipelines generates 5 offspring = 100 pipelines
5. 5% of that offspring cross over with another offspring

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy
3. Selects top 20 pipelines in the population
4. Each of the 20 pipelines generates 5 offspring = 100 pipelines
5. 5% of that offspring cross over with another offspring
6. 90% of the remaining encounter a random point mutation

Generation 1

etc.

https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

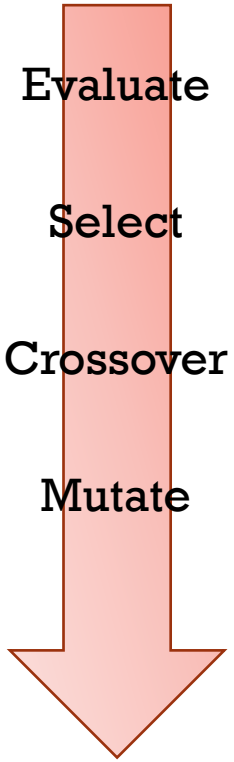
Genetic programming algorithm

Generation 0

1. Generation of 100 pipelines
2. Evaluation of their balanced cross-validation accuracy
3. Selects top 20 pipelines in the population
4. Each of the 20 pipelines generates 5 offspring = 100 pipelines
5. 5% of that offspring cross over with another offspring
6. 90% of the remaining encounter a random point mutation

Generation 1

etc.



https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier.py

Assets

- Follows scikit-learn API:
  - .fit()
  - .score()

Assets

- Follows scikit-learn API:
  - .fit()
  - .score()
- Supports classification and regression analyses

Assets

- Follows scikit-learn API:
  - .fit()
  - .score()
- Supports classification and regression analyses
- Can be executed from the command-line to gain access to more CPUs.

Assets

- Follows scikit-learn API:
  - .fit()
  - .score()
- Supports classification and regression analyses
- Can be executed from the command-line to gain access to more CPUs.
- Hyperparameters can be customised

```
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

digits = load_digits()
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
                                                    train_size=0.75, test_size=0.25)

tpot_config = {
    'sklearn.naive_bayes.GaussianNB': {
    },

    'sklearn.naive_bayes.BernoulliNB': {
        'alpha': [1e-3, 1e-2, 1e-1, 1., 10., 100.],
        'fit_prior': [True, False]
    },

    'sklearn.naive_bayes.MultinomialNB': {
        'alpha': [1e-3, 1e-2, 1e-1, 1., 10., 100.],
        'fit_prior': [True, False]
    }
}

tpot = TPOTClassifier(generations=5, population_size=20, verbosity=2,
                      config_dict=tpot_config)
tpot.fit(X_train, y_train)
print(tpot.score(X_test, y_test))
tpot.export('tpot_digits_pipeline.py')
```

Questions for today's hands-on

- Best model in a given time: 2min, 5min, 10min
- Influence of the number of generations on final accuracy / RSME
- Influence of the mutation and cross over rate on the final accuracy / RSME

https://github.com/UvA-DSC/2022-05-20_conventional-machine-learning