AutoML – The rise of the machines

Nuno César de Sá[1],

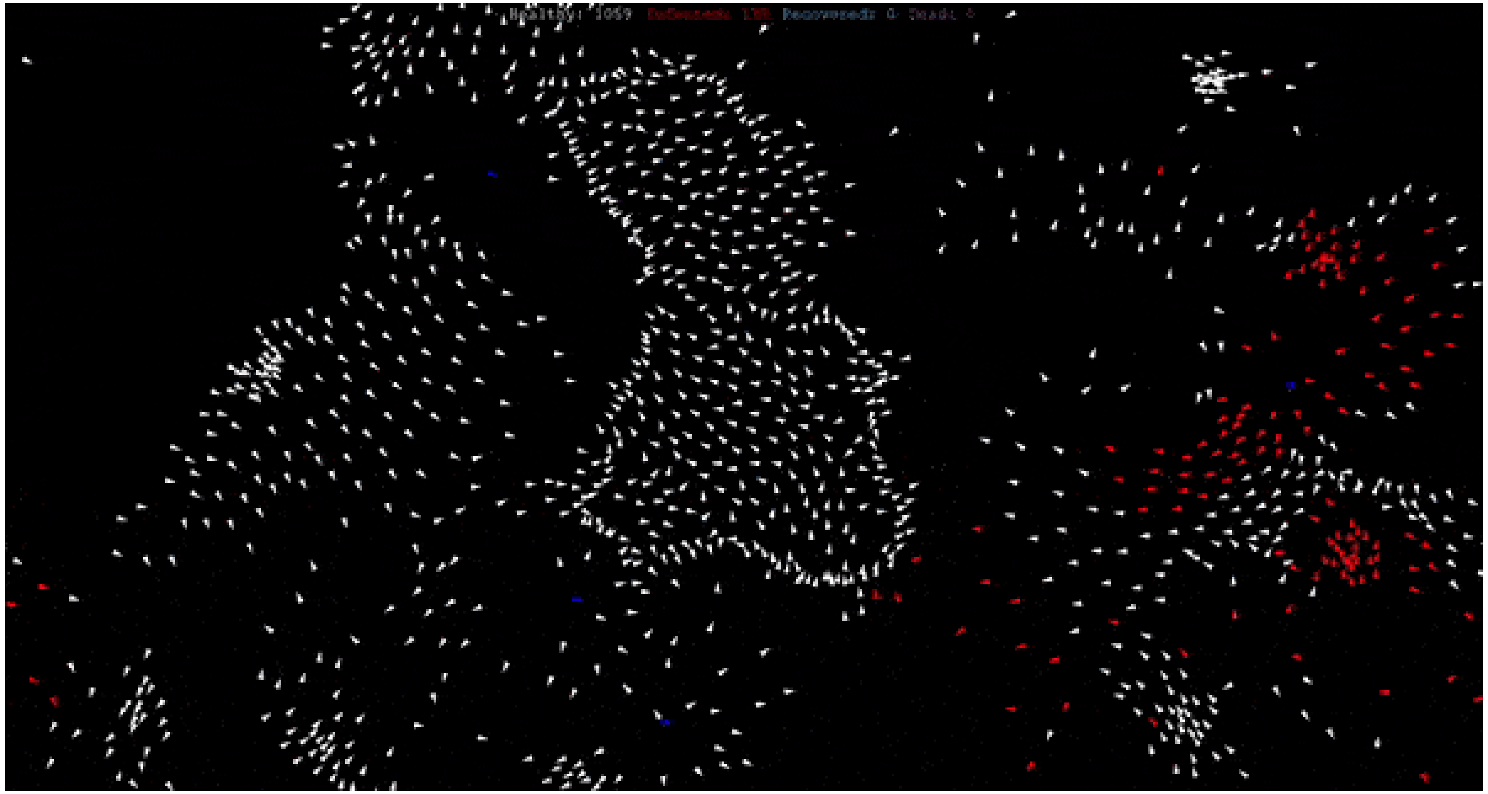
[1] Institute of Environmental Sciences (CML)

AutoML – The rise of the machines

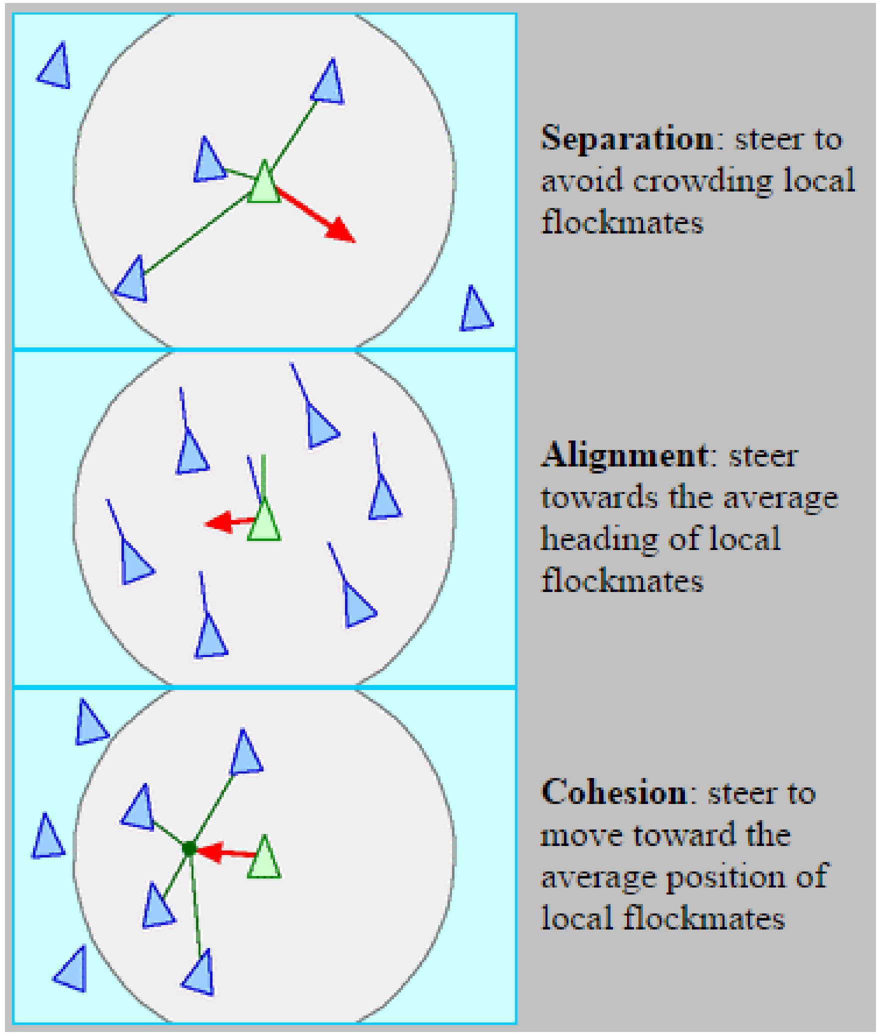
- Machine learning
  - What is it, and how does it work?
- AutoML
- “hands-on” example on RS data
  - Optional, naturally

CCTV footage of an AutoML system dealing with a GIS expert

Is this machine learning?



Flocking algorithm (Reynolds, 1986) – more info @ [link]()

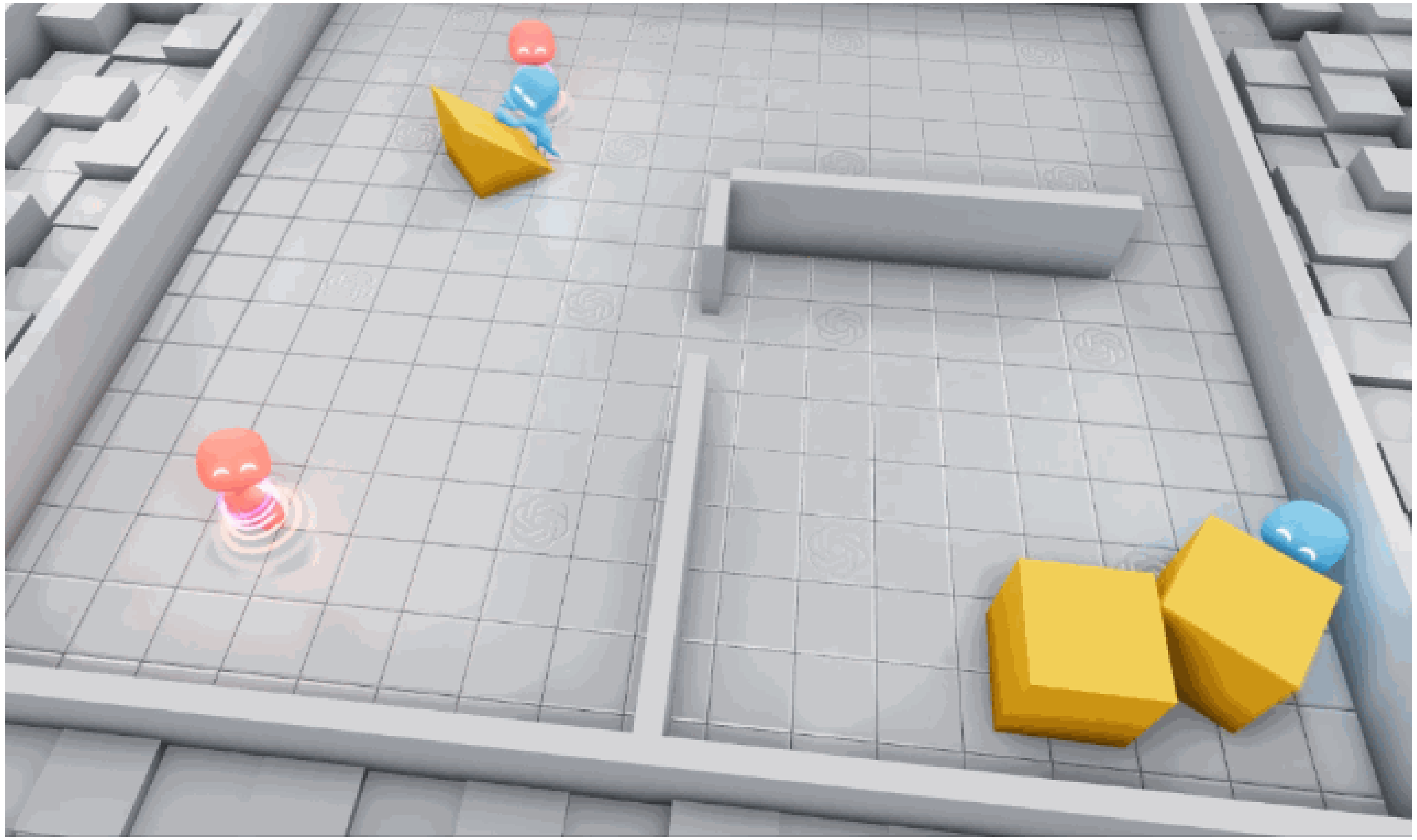


Separation: steer to avoid crowding local flockmates

Alignment: steer towards the average heading of local flockmates

Cohesion: steer to move toward the average position of local flockmates

Is this machine learning?

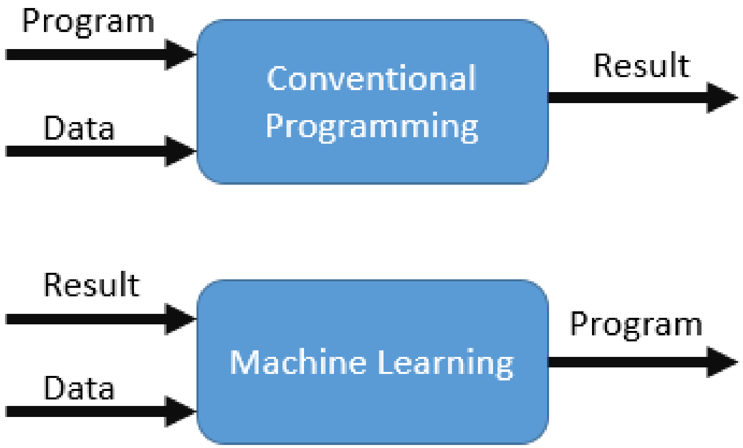


Emergent Tool Use From Multi-Agent Autocurricula, Bowen baker et al; 2 minutes papers

- Hide & Seek game:
  - Reds are seekers
  - Blues are hiding
- Agents:
  - Move blocks
  - Have a field of vision
  - Measure distance to everything else
- Survival strategies are learned

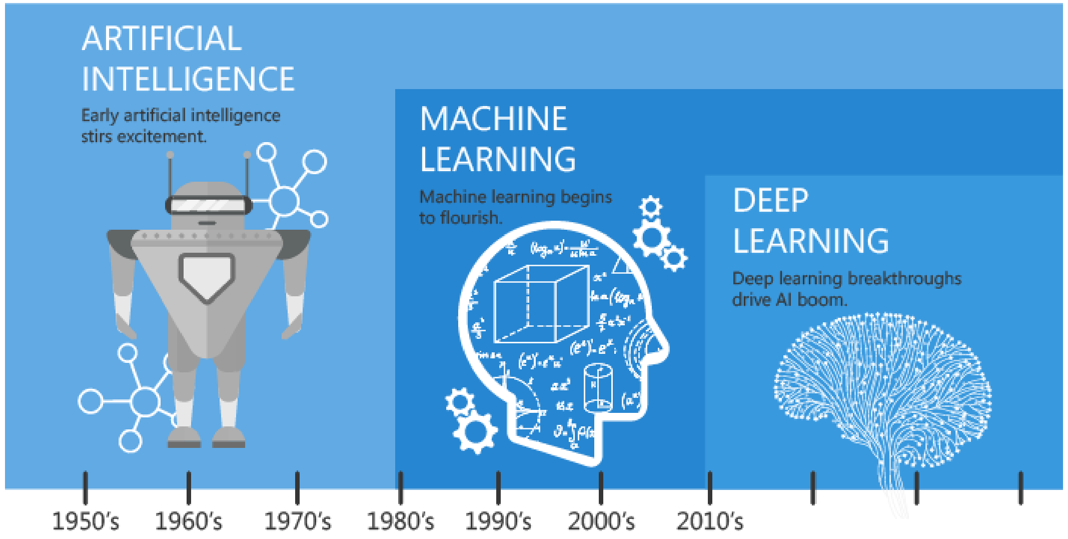
Machine learning – (mis)conceptions

- What is machine learning?
  - Arthur Samuel: “the field of study that gives computers the ability to learn without being explicitely programmed”



- It’s not “new”
  - First concepts of AI – *Boomer* technology
  - “Black-Box” -> mostly advanced statistics + linear models

- Risks:
  - Bad data -> bad model
  - Unequal access to computation -> inquality (Timnit Gebru)
  - “black-box” (or maybe not… Increasingly debatable!)


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950’s
1960’s
1970’s
1980’s
1990’s
2000’s
2010’s

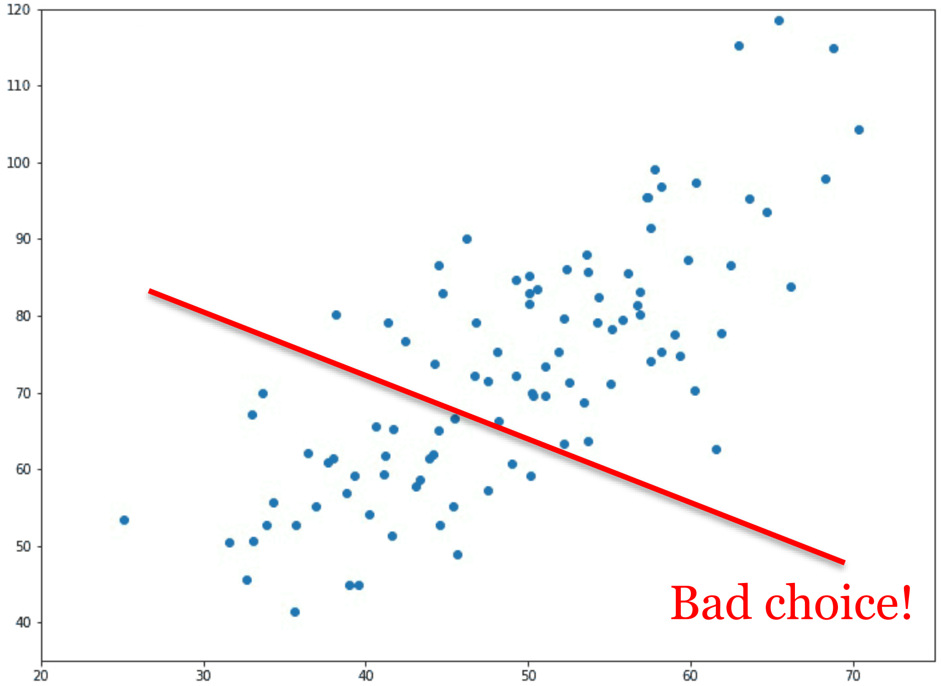
Machine learning – how does it work?

- ***How do machines actually, learn?***

- Consider you are asked to find the best linear model that fits a set of data
  - And you skipped the linear algebra class

- But you know that: Y = mX + C
  - AKA, representation

- Parameters:
  - m - slope
  - C - bias



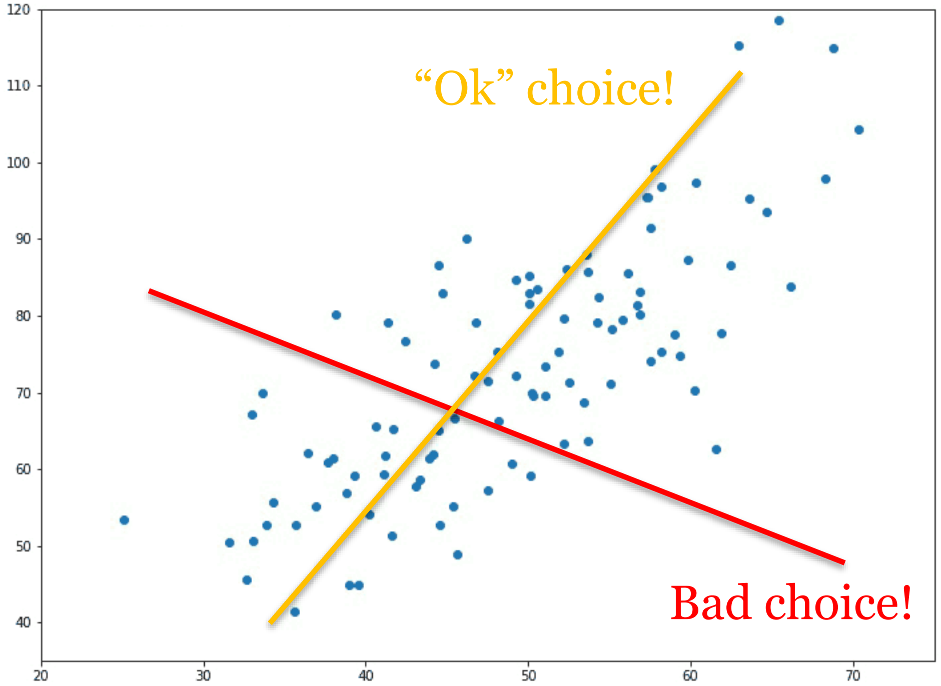
Machine learning – how does it work?

- ***How do machines actually, learn?***

- Consider you are asked to find the best linear model that fits a set of data
  - And you skipped the linear algebra class

- But you know that: Y = mX + C
  - AKA, representation

- Parameters:
  - m - slope
  - C - bias



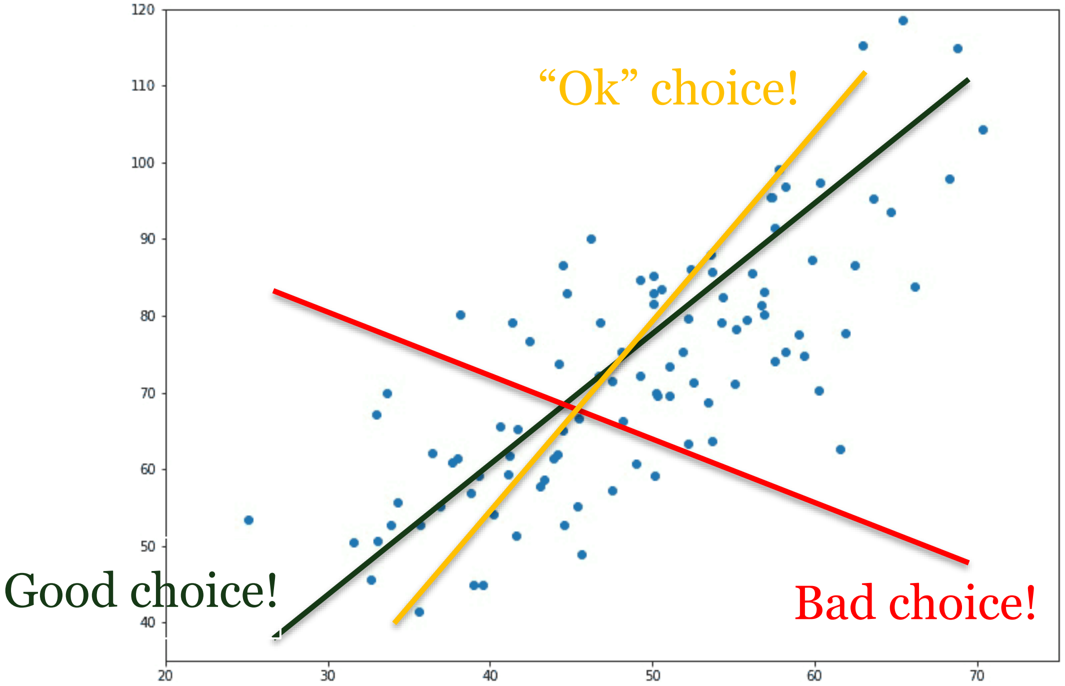
Machine learning – how does it work?

- ***How do machines actually, learn?***

- Consider you are asked to find the best linear model that fits a set of data
  - And you skipped the linear algebra class

- But you know that: Y = mX + C
  - AKA, representation

- Parameters:
  - m - slope
  - C - bias

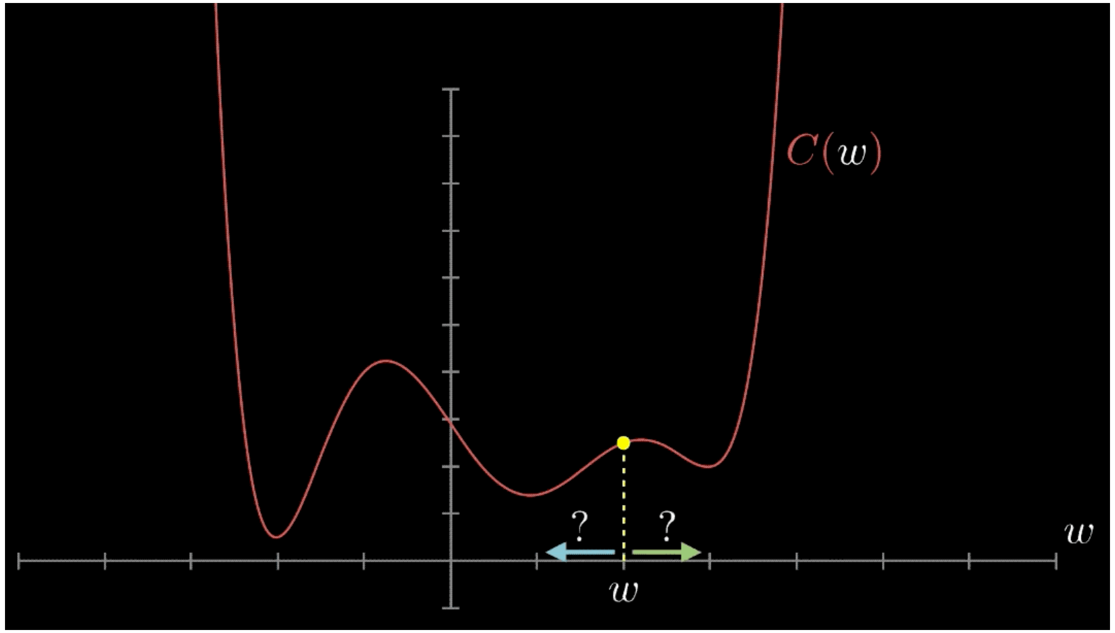


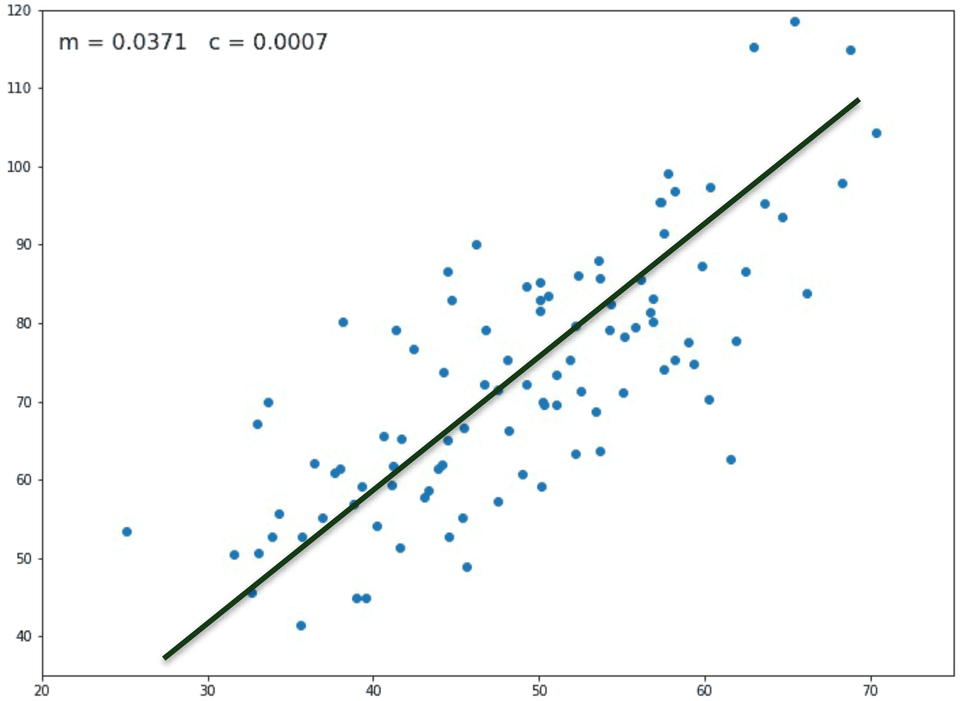
Evaluation:

$$|| Y - Y_{red} || > || Y - Y_{yellow} || > || Y - Y_{green} ||$$

Machine learning – how does it work?

3blue1brown: "But what is a neural network?" - Youtube playlist



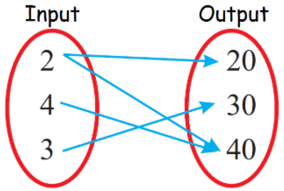


- If we would just try different random sets of parameters we might never find the best solution
- So, in general: start random but on each iteration, focus on the best results (optimization)

Machine learning – how does it work?

- At it’s core:
  - Mapping functions: F(x) = y
  - The computer learns functions F(x) that transforms input X to output Y

F(input) = Output



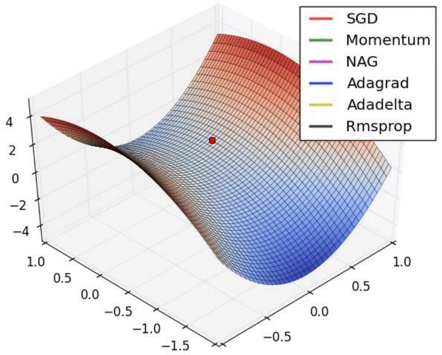
1. Algorithm/Representation
   - How to represent the relationship between the data

$Y = mX + C$

2. Evaluation
   - How to find if the model actually is working

$$|| Y - Y_{red} || > || Y - Y_{yellow} || > || Y - Y_{green} ||$$

3. Optimization
   - How to search the possible solutions to find the best model



Any quick questions?

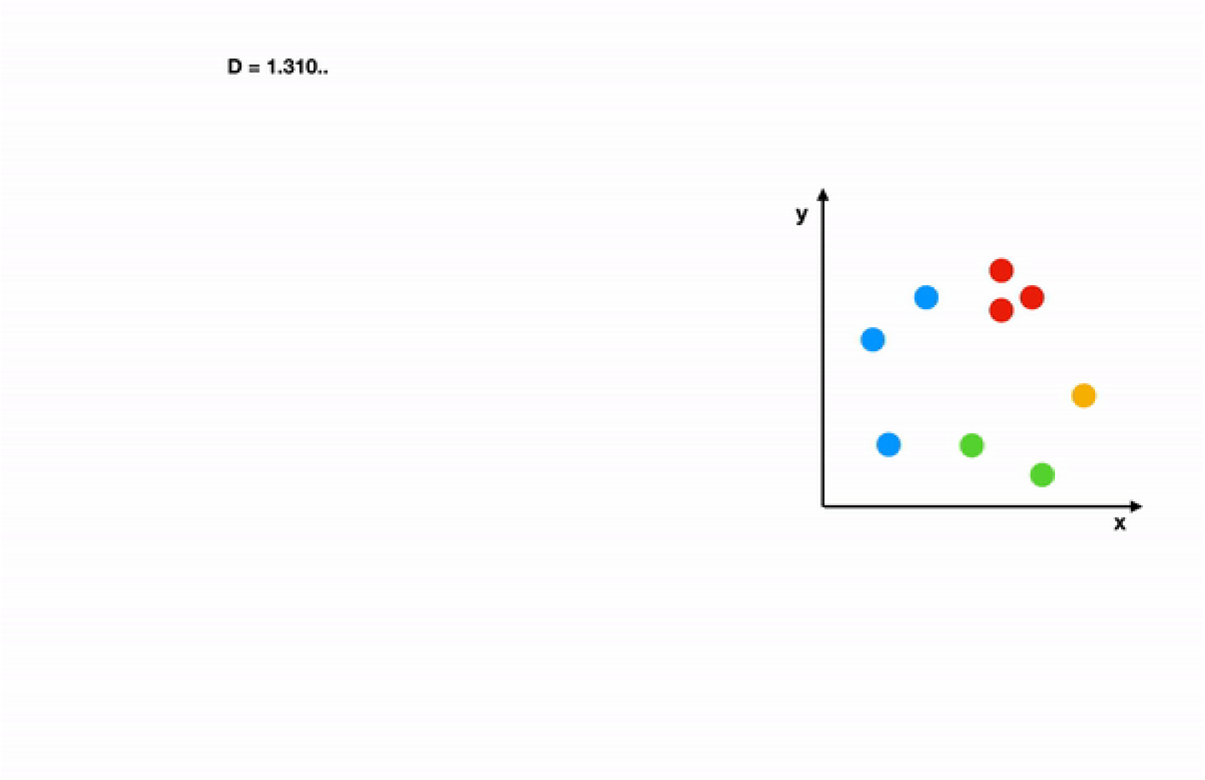
Universiteit Leiden
The Netherlands

Machine learning - Algorithms

- Random forest:
  - Decision trees (exactly what it means!)
  - Select rules that maximize similarity (e.g. Gini coefficient)
  - Fits Ordinary least squares regressions on the data subset

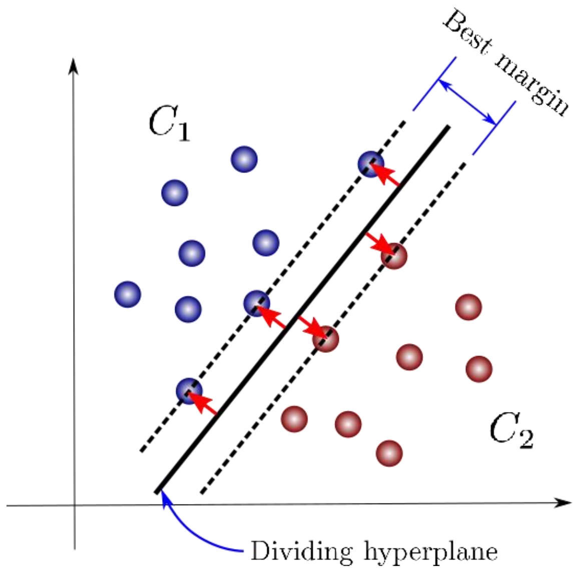
- Prediction:
  - Weighted sum of the weak models

- Typical parameters:
  - Maximum number of trees
  - Tree depth
  - Mininum number of points per “leaf” node, etc



Machine learning - Algorithms

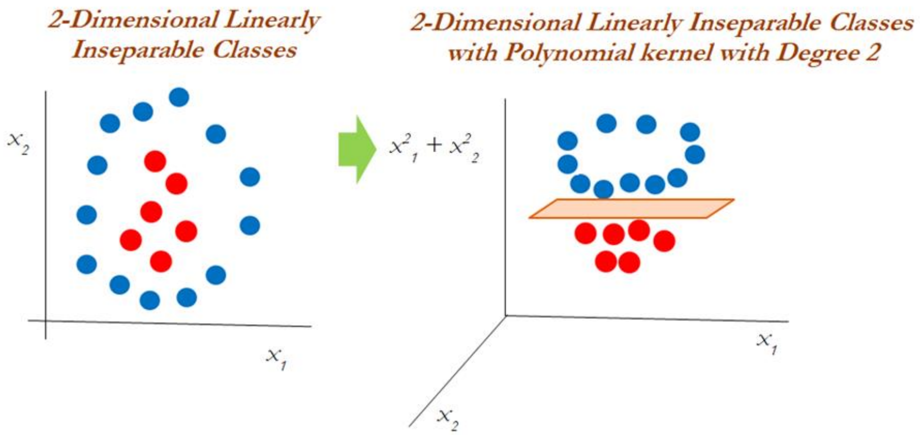
- Support vector machines:
  - Separates classes by finding the best plane that maximizes the separabiity between classes
  - The plane is defined in the vector format (hence, support vector!)



- For non linear behaviours - > kernel tricks
  - Extra feature spaces are created where linear hyperplanes can be created

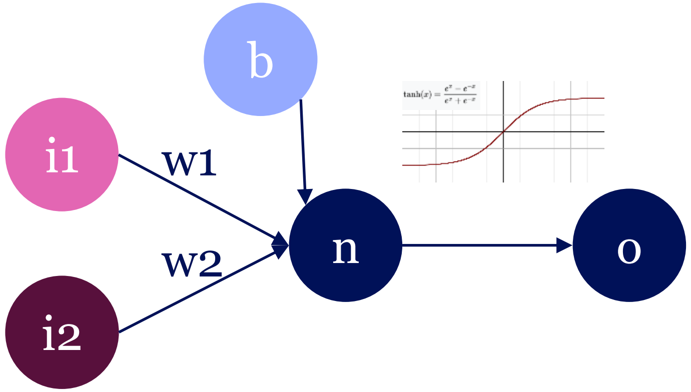
- Typical parameters:
  - C (regularization parameter); specific kernel parameters

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2})$$


2-Dimensional Linearly Inseparable Classes
2-Dimensional Linearly Inseparable Classes with Polynomial kernel with Degree 2
x_2
x_1
$x^2_1 + x^2_2$
x_1
x_2

Machine learning - Algorithms

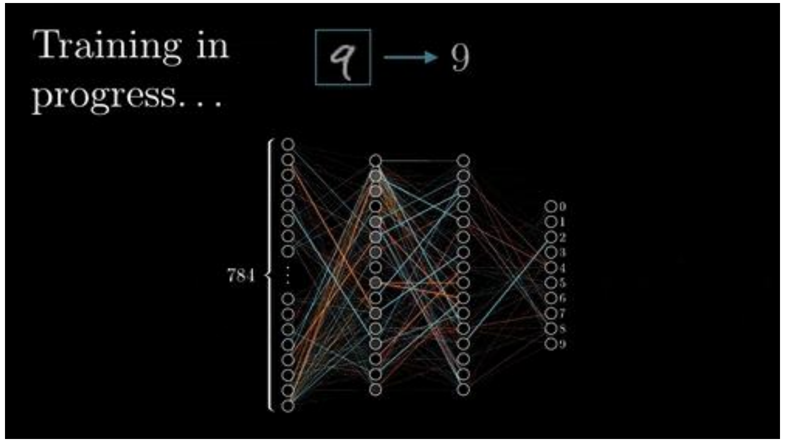
- Artificial neural networks:
  - Initially developed to mimic human brain
  - Each neuron is a “linear model”
  - Activation function transforms linear models to non-linear functions



- Minimizes the error
  - On each iteration, small changes to the various w and b available

w1*i1 + w2*i2 + b = n

- Typical parameters:
  - How many neurons & layers?
  - What type of layers?
  - What activation functions?

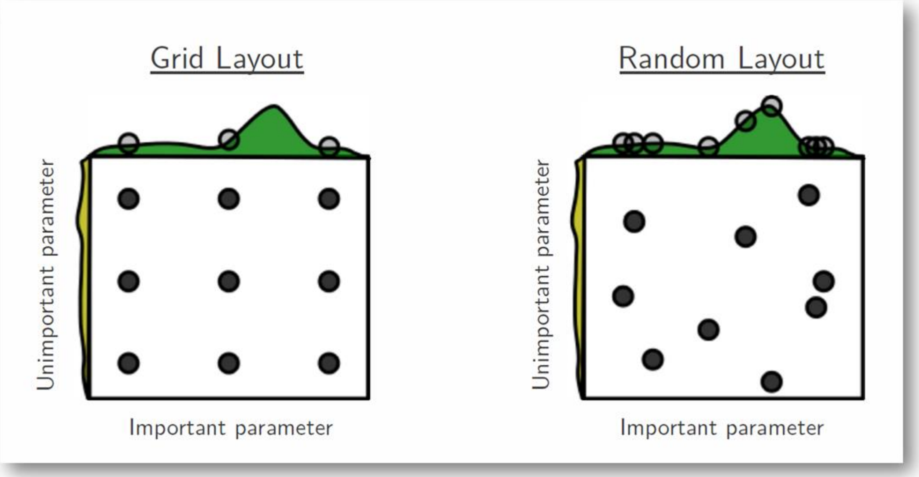


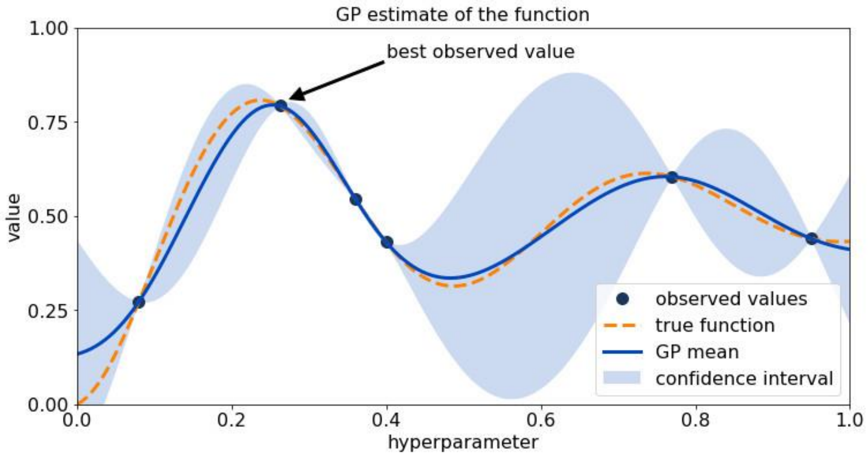
Machine learning – hyperparameters

- How many neurons and layers should a NN have?

- How to search efficiently for the model parameters?
    - We can "try" randomly and hope we find a good combination
    - We can be systematic
    - We can implement yet another optimization procedure..

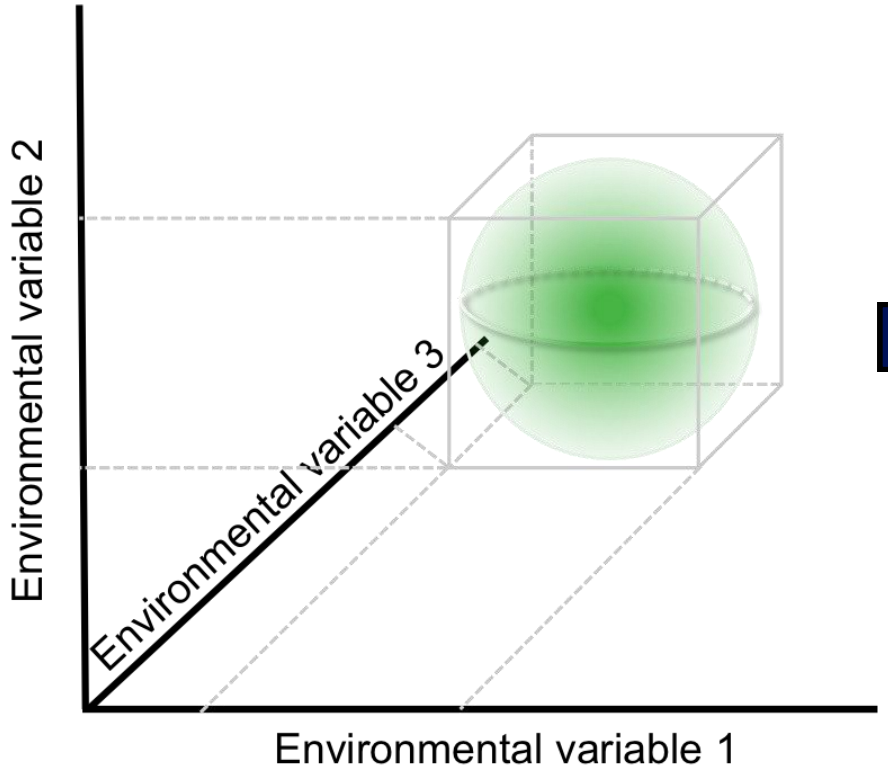
- More advanced: Bayesian optimization





Machine learning – in species distributions

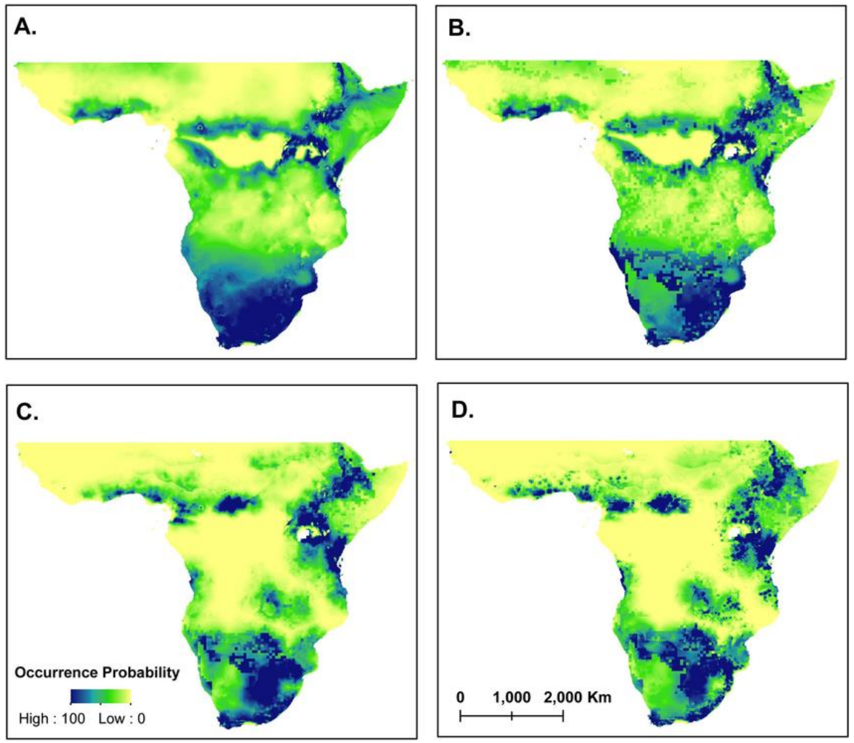
Ecological niche - EN



$\boldsymbol{f}(E1, E2, E3)$

ML

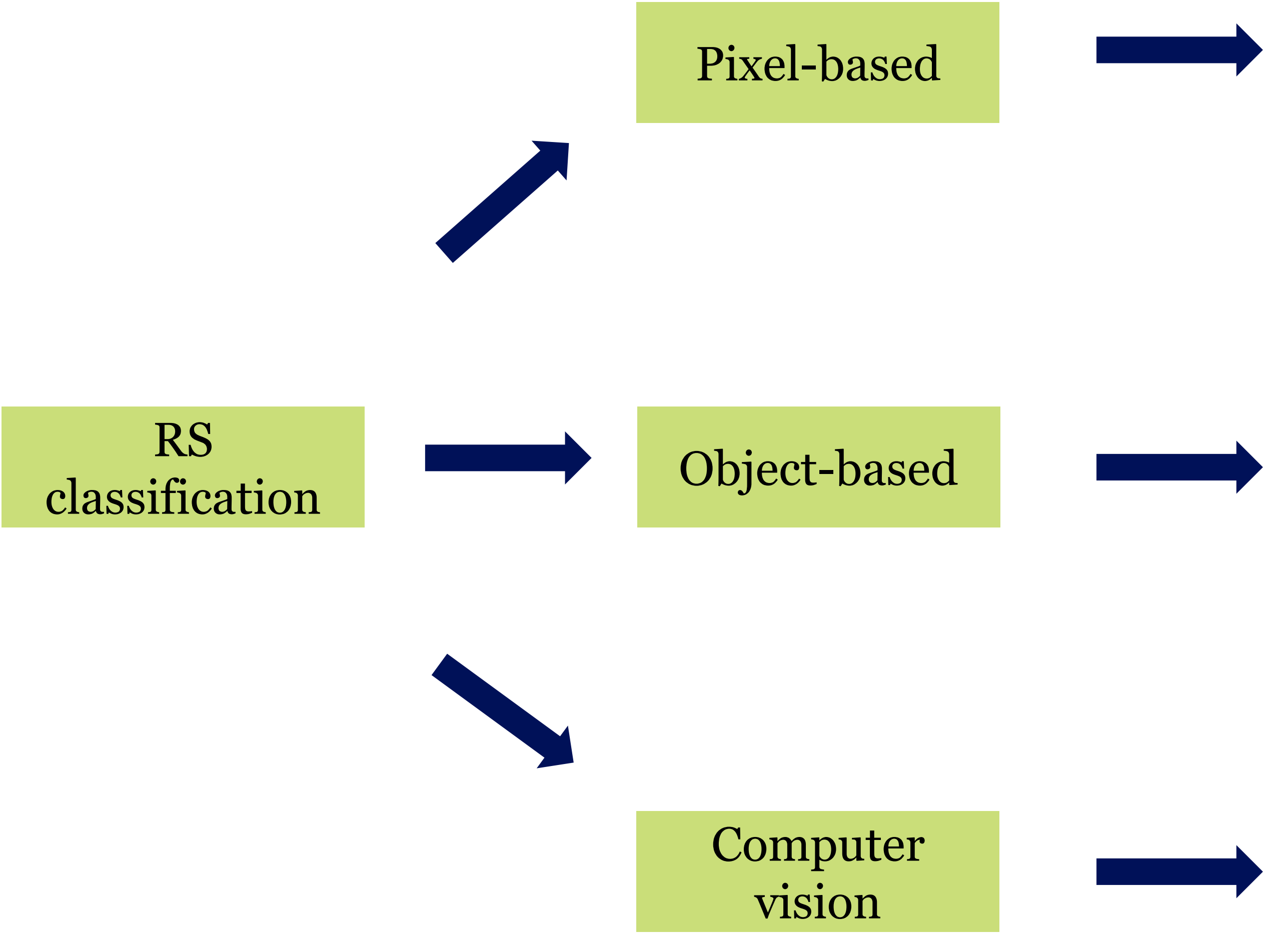
Geographic space - EN projected by f(A,B,M)



From: https://doi.org/10.4404/hystrix-27.1-11678

If the EN was a simple n-dimentional surface… It would be an easy problem….

Machine learning – In remote sensing

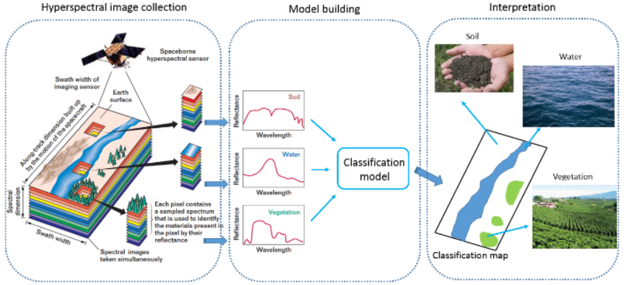


RS classification

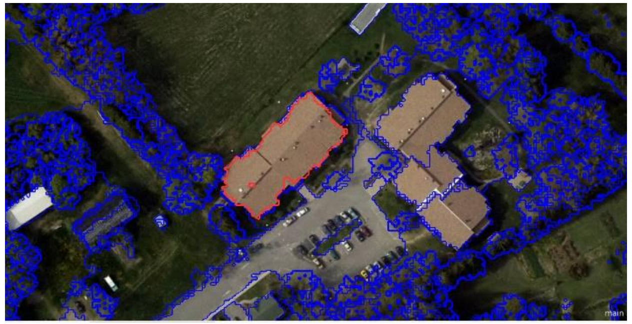
Pixel-based

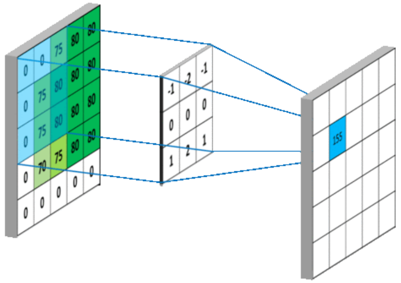
Object-based

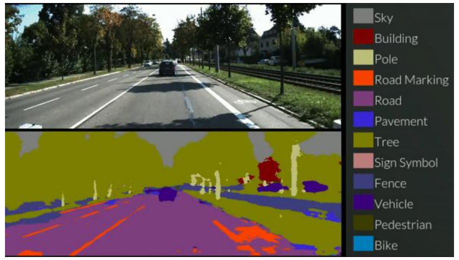
Computer vision



Hyperspectral image collection
Spaceborne hyperspectral sensor
Swath width of imaging sensor
Earth surface
Along-track dimension built up by the motion of the spacecraft
Spectral dimension
Swath width
Each pixel contains a sampled spectrum that is used to identify the materials present in the pixel by their reflectance
Spectral images taken simultaneously
Model building
Reflectance
Wavelength
Soil
Water
Vegetation
Classification model
Interpretation
Soil
Water
Vegetation
Classification map



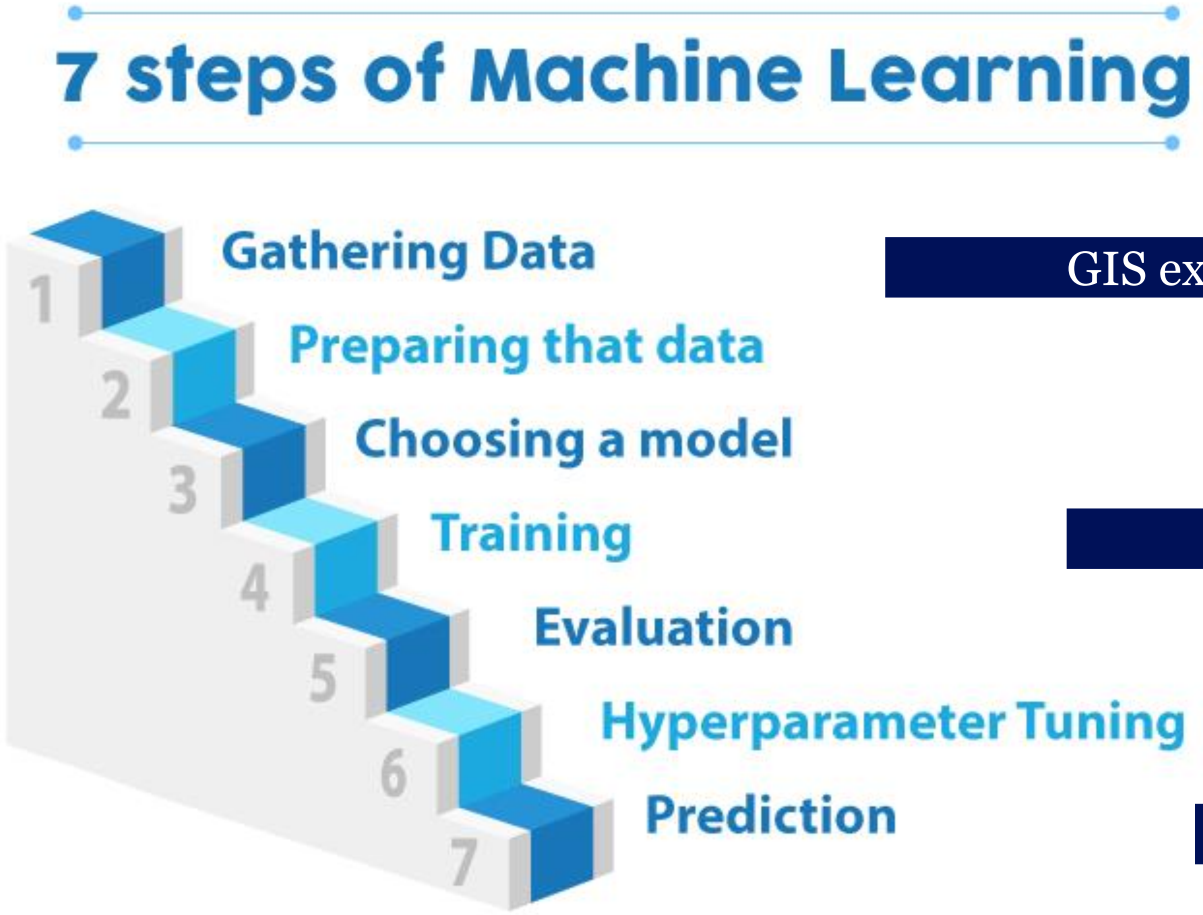




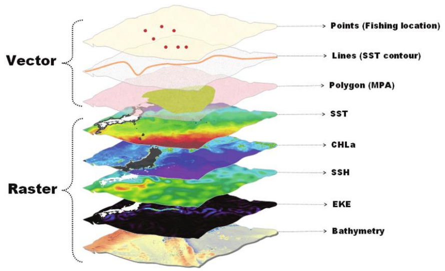
Any questions?

Universiteit Leiden
The Netherlands

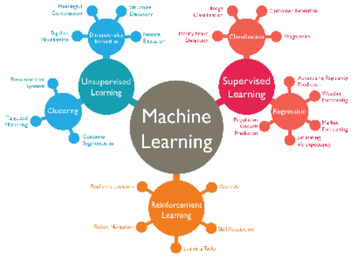
Machine learning – where to start?



GIS expert



ML expert



Client

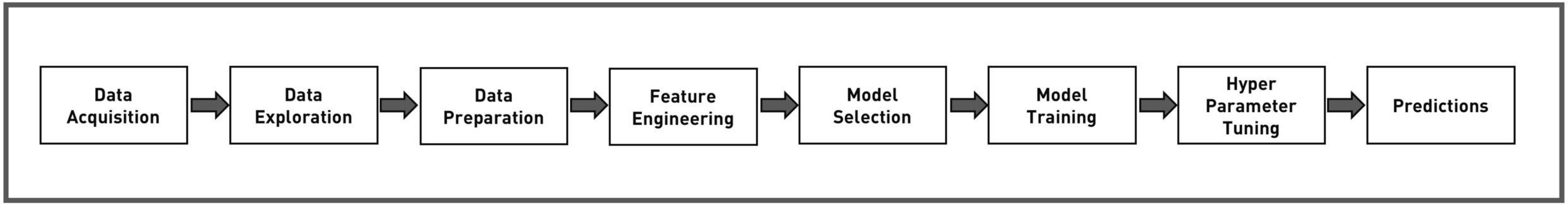
CLIENTS FROM HELL
A collection of anonymously-contributed client horror stories from designers

The problem is:

- What is the best algorithm for that specific task?
- What are the best parameters?
- What pre-processing is required?
- How can “expertise” help?

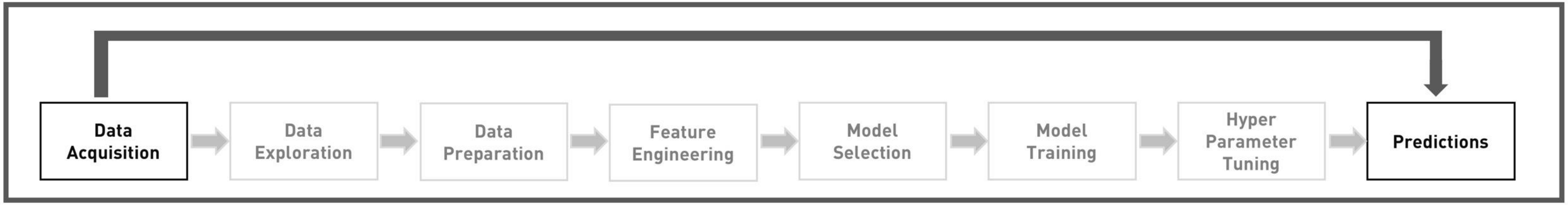
All these need to be addressed.. If i want to scale up a business

AutoML



Data Acquisition → Data Exploration → Data Preparation → Feature Engineering → Model Selection → Model Training → Hyper Parameter Tuning → Predictions

Traditional Machine Learning Workflow



Data Acquisition → Data Exploration → Data Preparation → Feature Engineering → Model Selection → Model Training → Hyper Parameter Tuning → Predictions

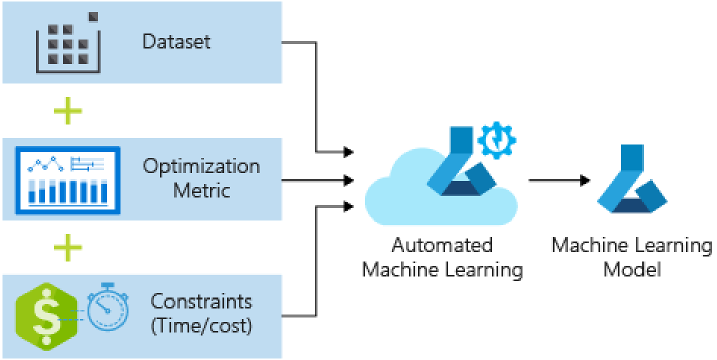
AutoML Workflow

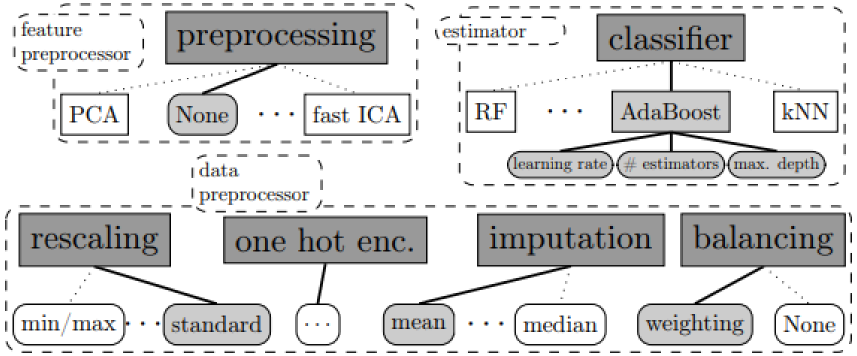
AutoML

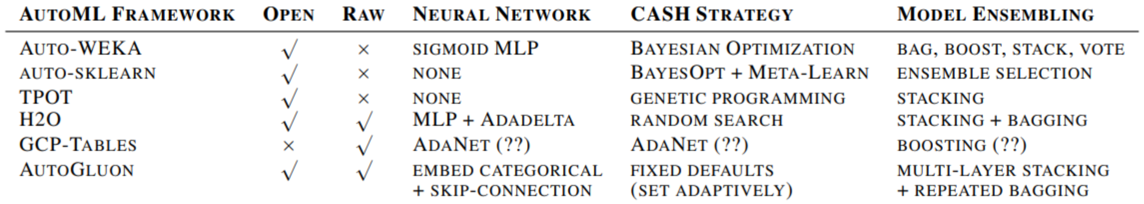
- Automation
  - Lower the entry level for machine learning
  - Optimal for deploying as a service
  - Minimum need to have an expert look at the clients data

- CASH – Combined Algorithm Selection and Hyperparameter optimization

- Many options in the "market":
  - H2o (R also); Auto-Weka; TPOP; AutoGluon
  - Today: AutoSklearn







AutoML Framework	Open	Raw	Neural Network	CASH Strategy	Model Ensembling
Auto-WEKA	√	×	sigmoid MLP	Bayesian Optimization	bag, boost, stack, vote
auto-sklearn	√	×	none	BayesOpt + Meta-Learn	ensemble selection
TPOT	√	×	none	genetic programming	stacking
H2O	√	√	MLP + AdaDelta	random search	stacking + bagging
GCP-Tables	×	√	AdaNet (??)	AdaNet (??)	boosting (??)
AutoGluon	√	√	embed categorical + skip-connection	fixed defaults (set adaptively)	multi-layer stacking + repeated bagging

AutoGluon-Tabular: Robbust and Accurate AutoML for Structured data, 2020

AutoSklearn

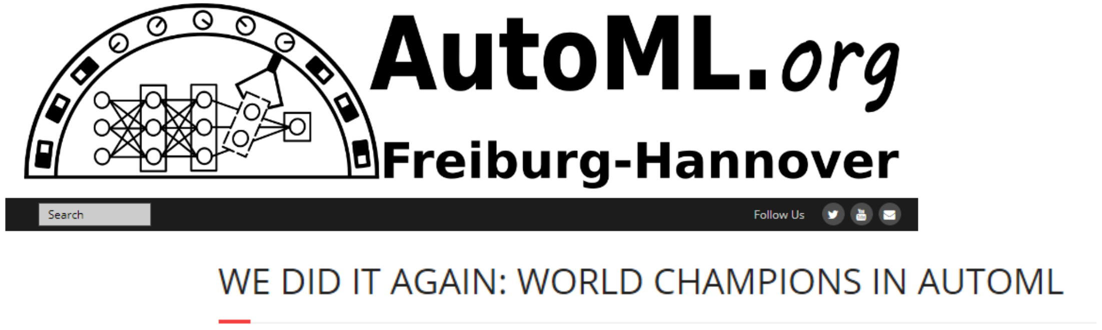
- Uses the algorithms available in scikit-learn
  - The “standard” python library for Machine learning
  - Many different algorithms available

- CASH:
  - Bayesian optimization
  - Meta-learning

- Meta-learning:
  - Pre-tested pipelines first (emulates expertise)
  - Based on pre-trained pipelines on open data

scikit-learn

Machine Learning in Python

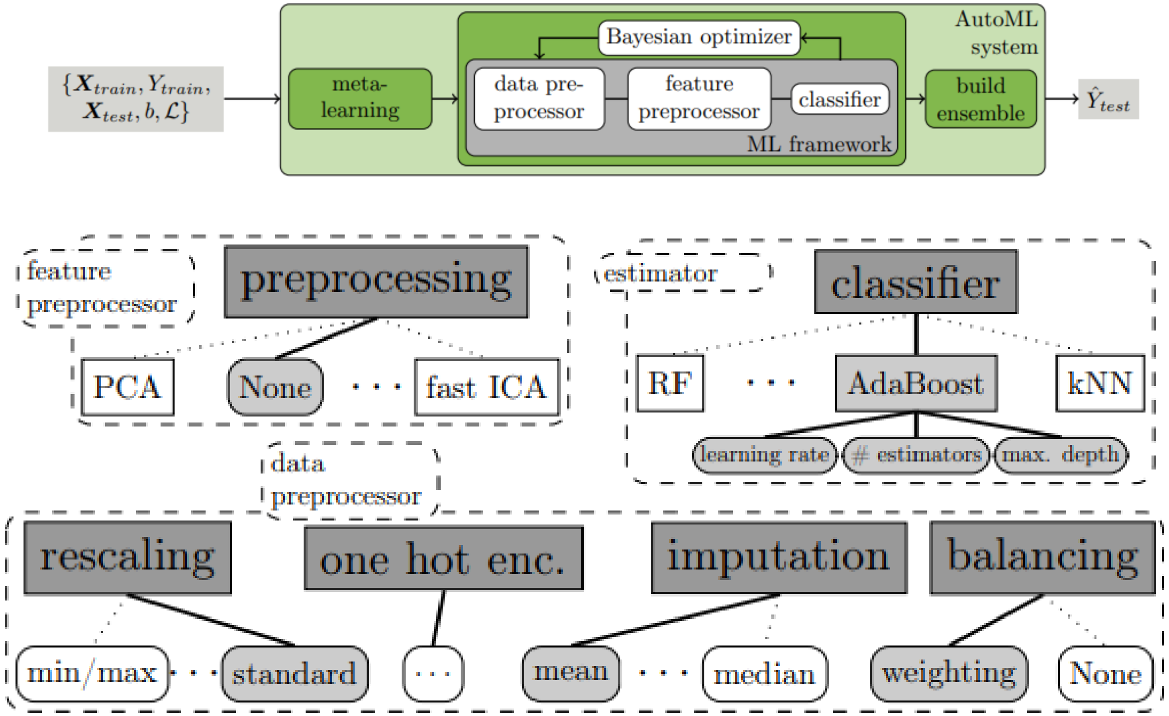


WE DID IT AGAIN: WORLD CHAMPIONS IN AUTOML

https://www.automl.org/

AutoSklearn

- Pre-processing of data and features
- Algorithm selection



$\{X_{train}, Y_{train}, X_{test}, b, \mathcal{L}\}$

AutoML system

meta-learning

Bayesian optimizer

data pre-processor

feature preprocessor

classifier

ML framework

build ensemble

$\hat{Y}_{test}$

feature preprocessor

preprocessing

PCA

None

· · ·

fast ICA

estimator

classifier

RF

· · ·

AdaBoost

kNN

learning rate

estimators

max. depth

data preprocessor

rescaling

one hot enc.

imputation

balancing

min/max

· · ·

standard

· · ·

mean

· · ·

median

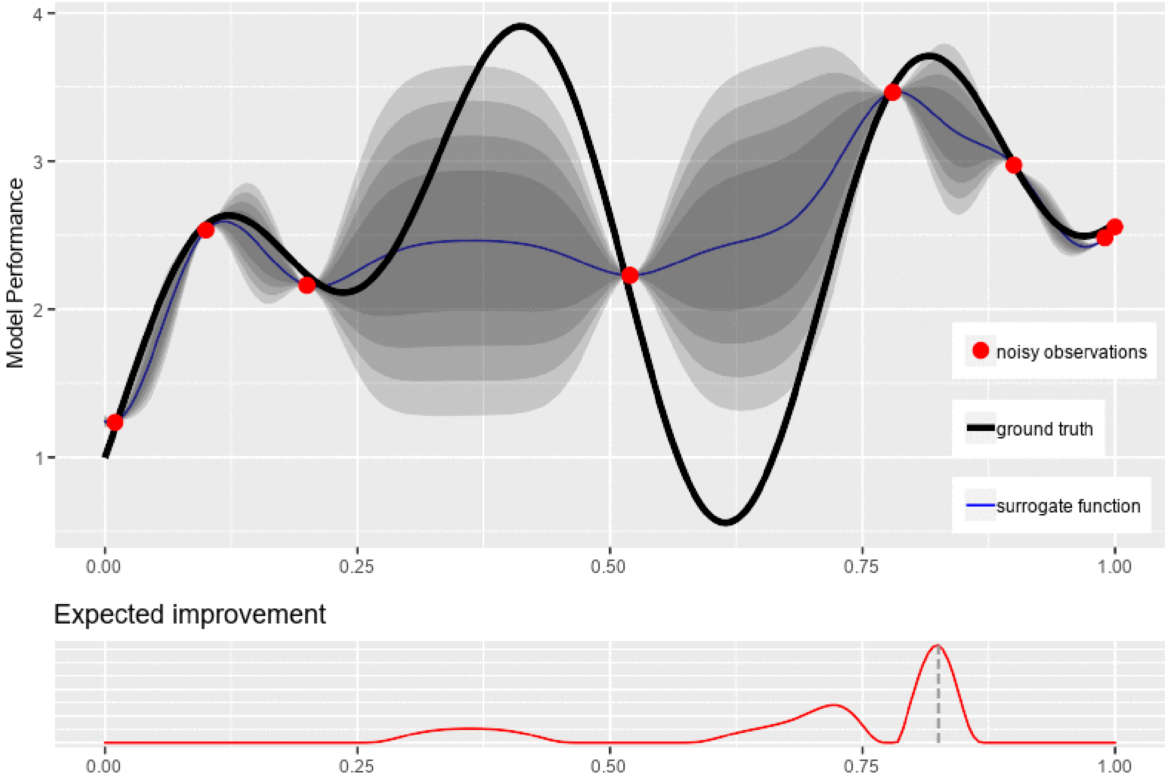
weighting

None

[Efficient and Robust Automated ML](), Feurer et al, 2015

Bayesian optimization

- Uses a probabilistic approach to find the "options" that promises most improvement

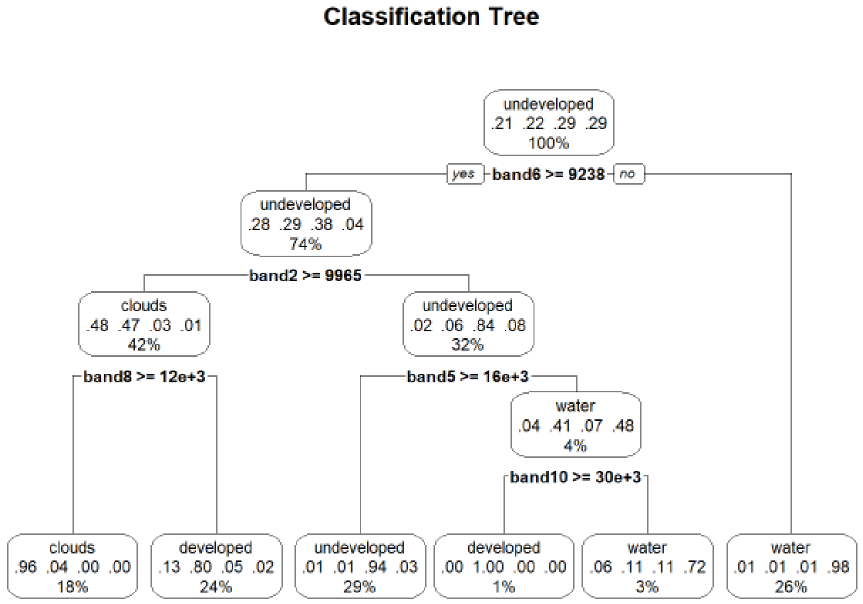


Classification example:

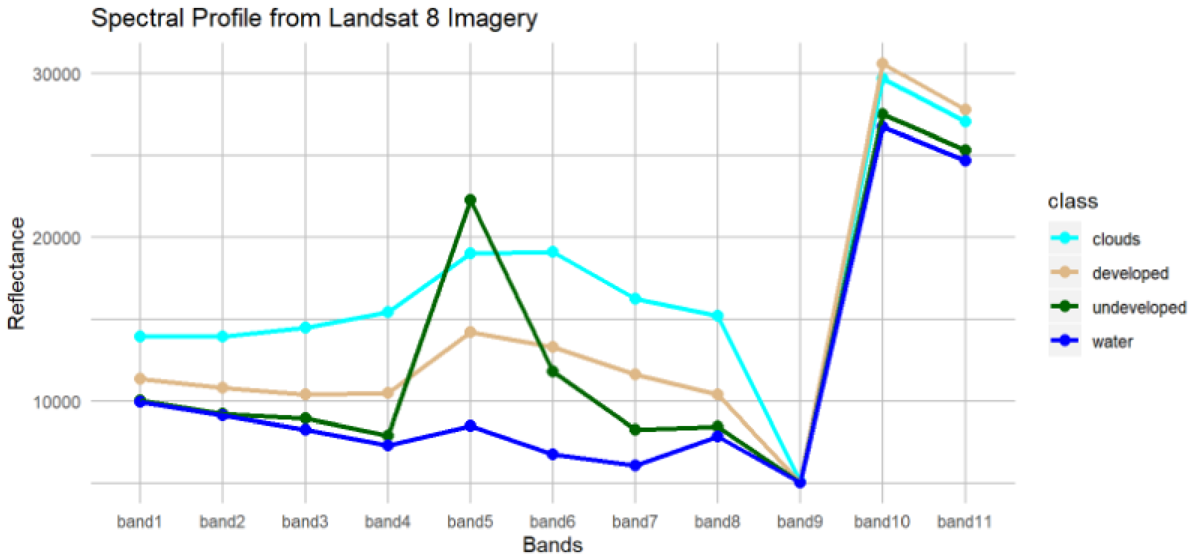
- Applying to RS data:
  - Spectral classification
  - [Adapted from Sydney Goldstein tutorial on R]

- Landsat 8 image, Calgary
  - Clouds
  - Developed
  - Underdeveloped
  - Water

- Accuracy:
  - Overall: 0.9186
  - Kappa: 0.89



Classification Tree
undeveloped .21 .22 .29 .29 100%
yes
band6 >= 9238
no
undeveloped .28 .29 .38 .04 74%
band2 >= 9965
clouds .48 .47 .03 .01 42%
band8 >= 12e+3
undeveloped .02 .06 .84 .08 32%
band5 >= 16e+3
water .04 .41 .07 .48 4%
band10 >= 30e+3
clouds .96 .04 .00 .00 18%
developed .13 .80 .05 .02 24%
undeveloped .01 .01 .94 .03 29%
developed .00 1.00 .00 .00 1%
water .06 .11 .11 .72 3%
water .01 .01 .01 .98 26%



AutoSklearn on RS data

NOT WORKING

- Only bands 1,2,3,4,5,6,7
  - I was lazy to deal with the resolution of the rest..

- AutoSklearn configurations:
  - Use meta-learning
  - Max time 60s
  - 5-fold crossvalidation

- Github: nunocesarsa/Examples/AutoML/AutoSklearn

- Accuracy: ~ 0.96

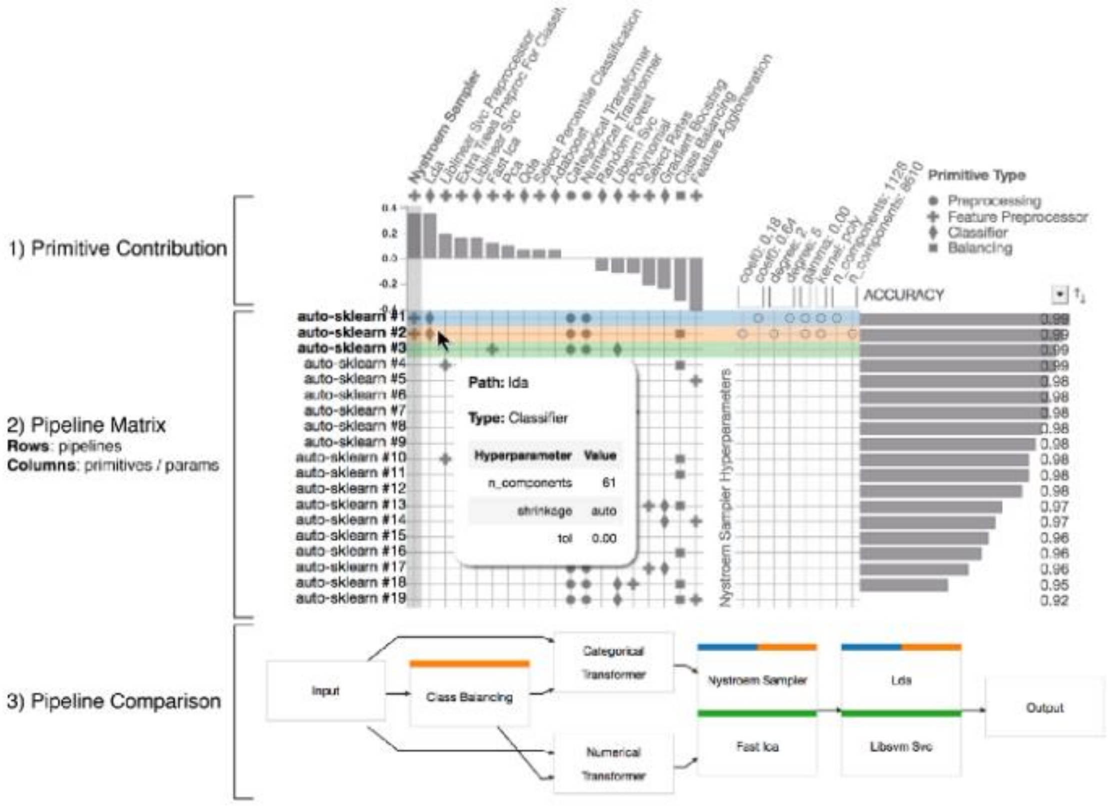


Fig. 1 PipelineProfiler applied to the Digits dataset.

https://towardsdatascience.com/exploring-auto-sklearn-models-with-pipelineprofiler-5b2c54136044

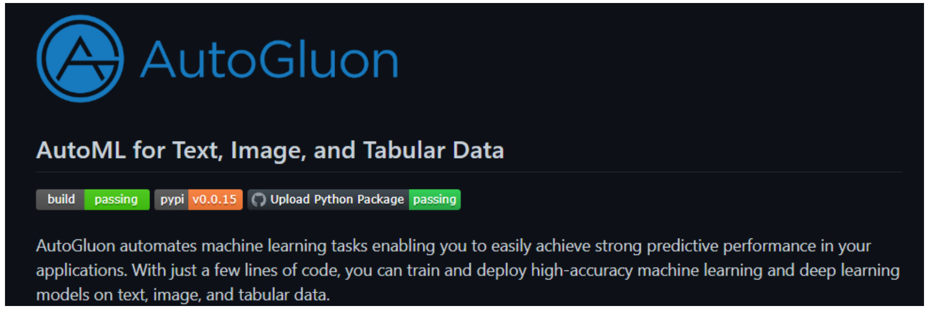
AutoGluon

- Developed by AWS to be deployed as service
  - Work for very different tasks (images, tables, text, NAS)

- Different: not a CASH approach
  - No hyperparameter tunning (possible, but not recommended)

- “Multi-layer stacking”
  - On each layer, the predictions of the models on the previous layers are added as features to the next layer

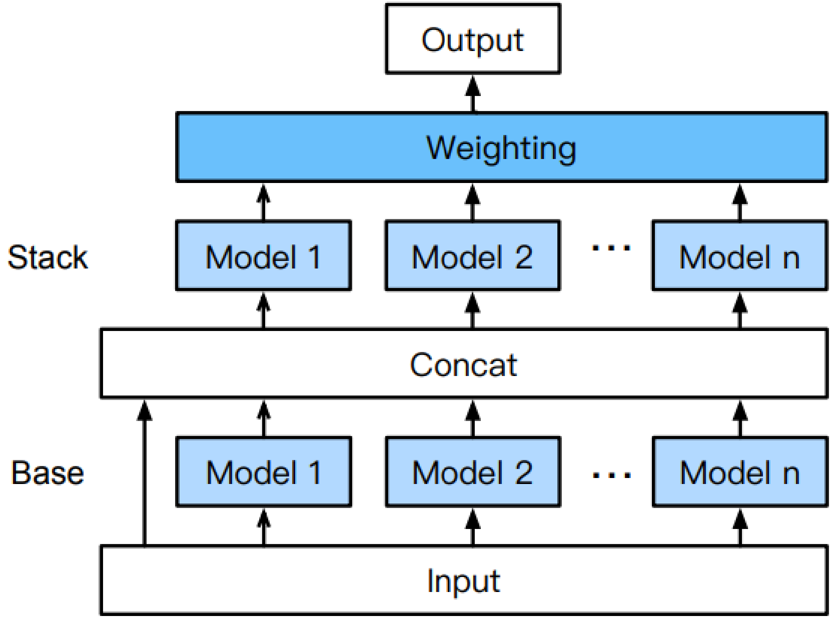
- Weak learners combined > strong learners



AutoGluon

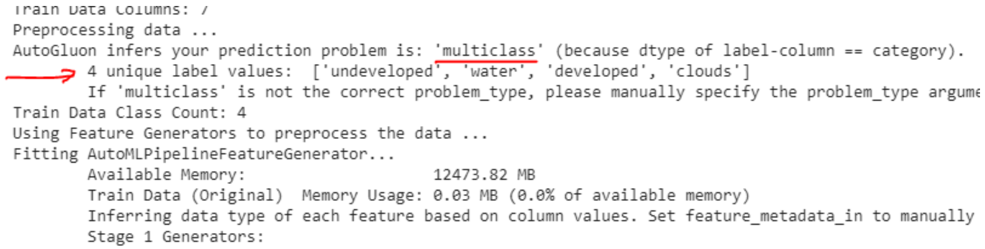
AutoML for Text, Image, and Tabular Data

AutoGluon automates machine learning tasks enabling you to easily achieve strong predictive performance in your applications. With just a few lines of code, you can train and deploy high-accuracy machine learning and deep learning models on text, image, and tabular data.



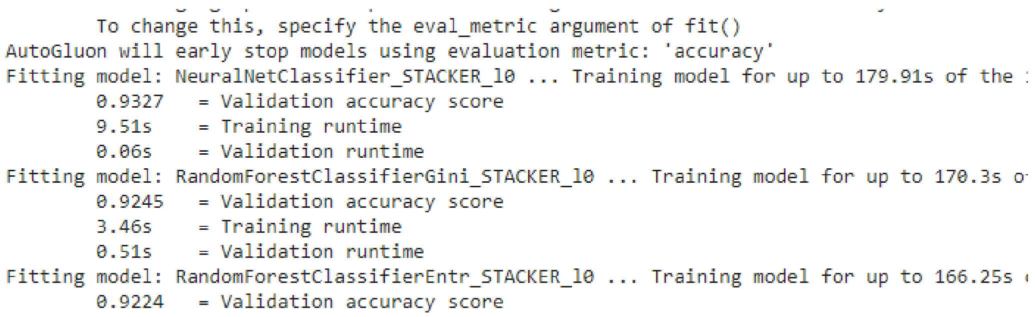
AutoGluon

- There are a number of “pre-sets” that are changeable:



```
Train Data Columns: 7
Preprocessing data ...
AutoGluon infers your prediction problem is: 'multiclass' (because dtype of label-column == category).
        4 unique label values:  ['undeveloped', 'water', 'developed', 'clouds']
        If 'multiclass' is not the correct problem_type, please manually specify the problem_type argume
Train Data Class Count: 4
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
        Available Memory:                    12473.82 MB
        Train Data (Original)  Memory Usage: 0.03 MB (0.0% of available memory)
        Inferring data type of each feature based on column values. Set feature_metadata_in to manually
        Stage 1 Generators:
```

- In our case, we opted for a very simple preset:
  - ‘Hyperameter tuning: light -> meaning very small tunning



```
        To change this, specify the eval_metric argument of fit()
AutoGluon will early stop models using evaluation metric: 'accuracy'
Fitting model: NeuralNetClassifier_STACKER_l0 ... Training model for up to 179.91s of the
        0.9327   = Validation accuracy score
        9.51s    = Training runtime
        0.06s    = Validation runtime
Fitting model: RandomForestClassifierGini_STACKER_l0 ... Training model for up to 170.3s o
        0.9245   = Validation accuracy score
        3.46s    = Training runtime
        0.51s    = Validation runtime
Fitting model: RandomForestClassifierEntr_STACKER_l0 ... Training model for up to 166.25s
        0.9224   = Validation accuracy score
```

- Sequential model training

- Accuracy:
  - Overall: 0.95
  - Kappa: 0.93

Github: https://github.com/nunocesarsa/Examples

In summary:

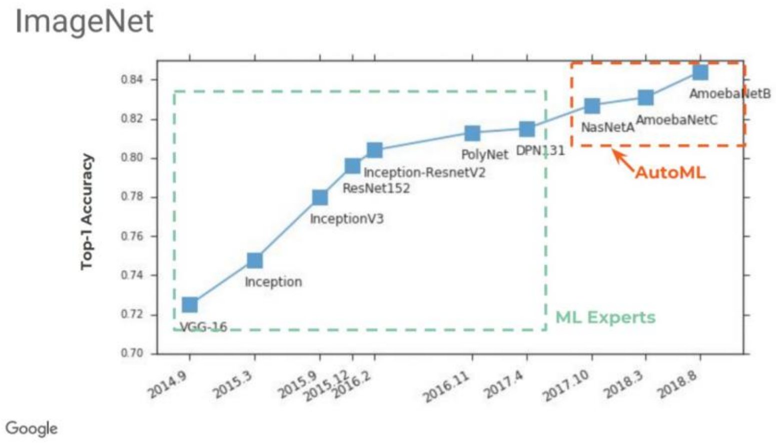
Automated machine learning:

1. Improves accuracy & reduces the “human-in-the-loop”
   - Critical criteria for any cloud service

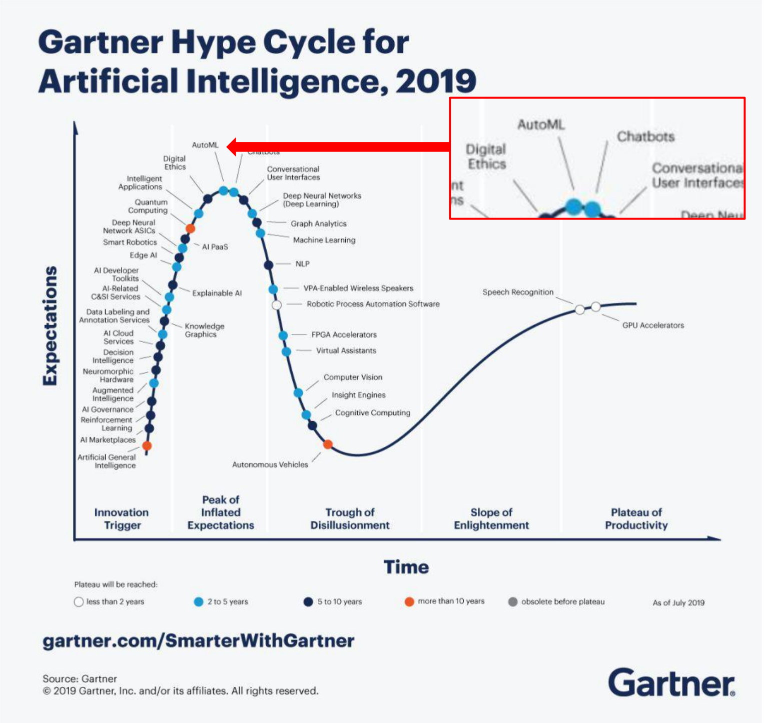
2. Can provide more insights than just single model approaches
   - Not show.. Autosklearn pipeliner is very cool

3. Can/should be fine-tuned for your tasks
   - (e.g., meta-learning from RS classifications)
   - E.g., in the RS field would be benefit from a Vegetation indice feature generator?

4. Deployable!
   - And with pre-set time.. And quicker than any “hands-on” approach



Gartner Hype Cycle for Artificial Intelligence, 2019



gartner.com/SmarterWithGartner

Source: Gartner
© 2019 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner.

Any questions?

Universiteit Leiden
The Netherlands

We can go for the colab after/during