# Genetic Programming in Auto-ML

Celio H.N. Larcher Jr.

Laboratório Nacional de Computação Científica

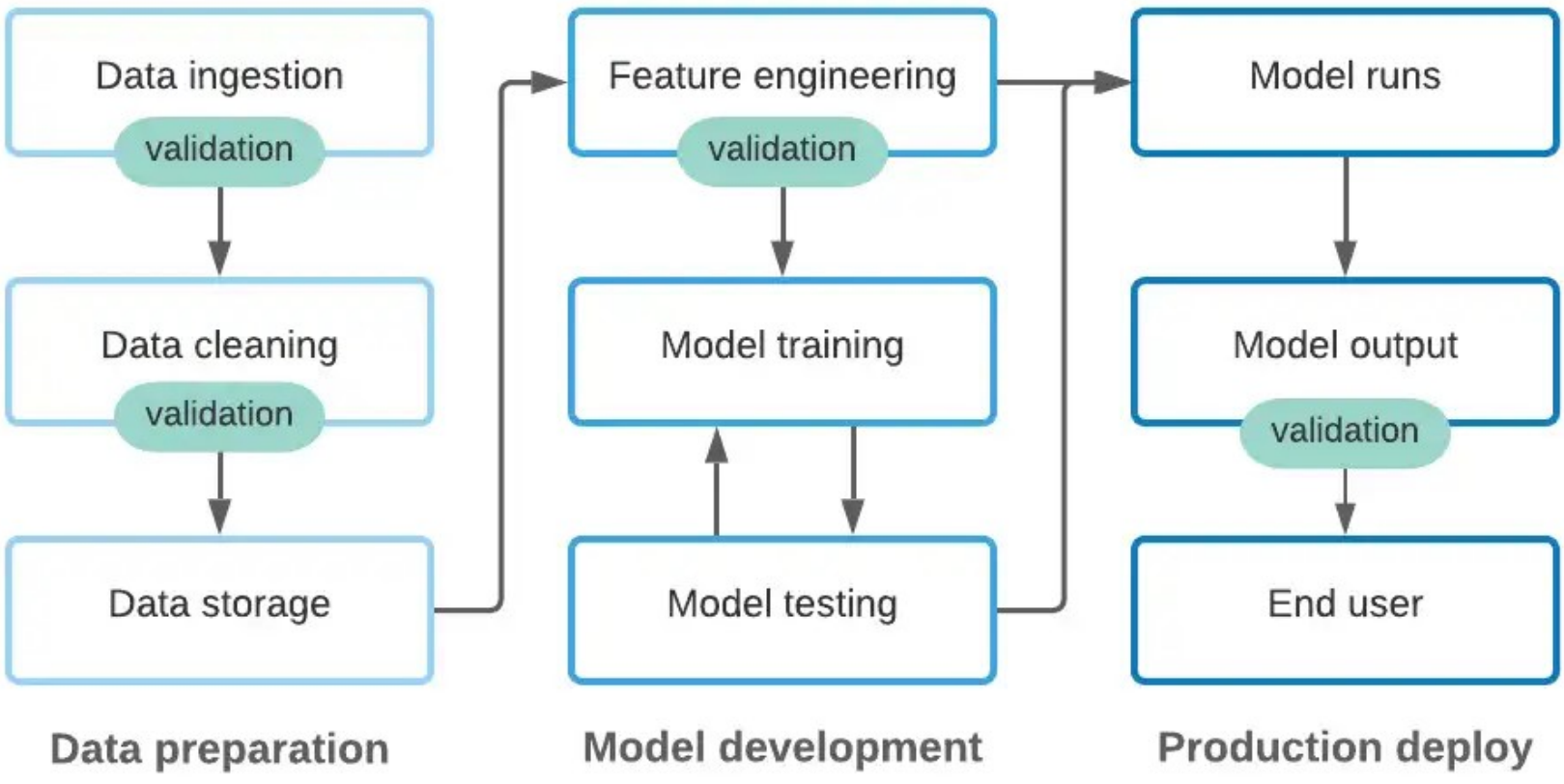
December - 2021

# Topics

# Topics

# Machine Learning

- Machine learning (ML) methods are trends today, with several fields trying to use them
- Building machine learning solutions is not a trivial task and, often, requires some expertise from the ML practitioner
  - Knowledge of the domain
  - Knowledge of the ML techniques
- Besides, building a ML solution involves several steps like data cleaning, data transformation, model building, etc

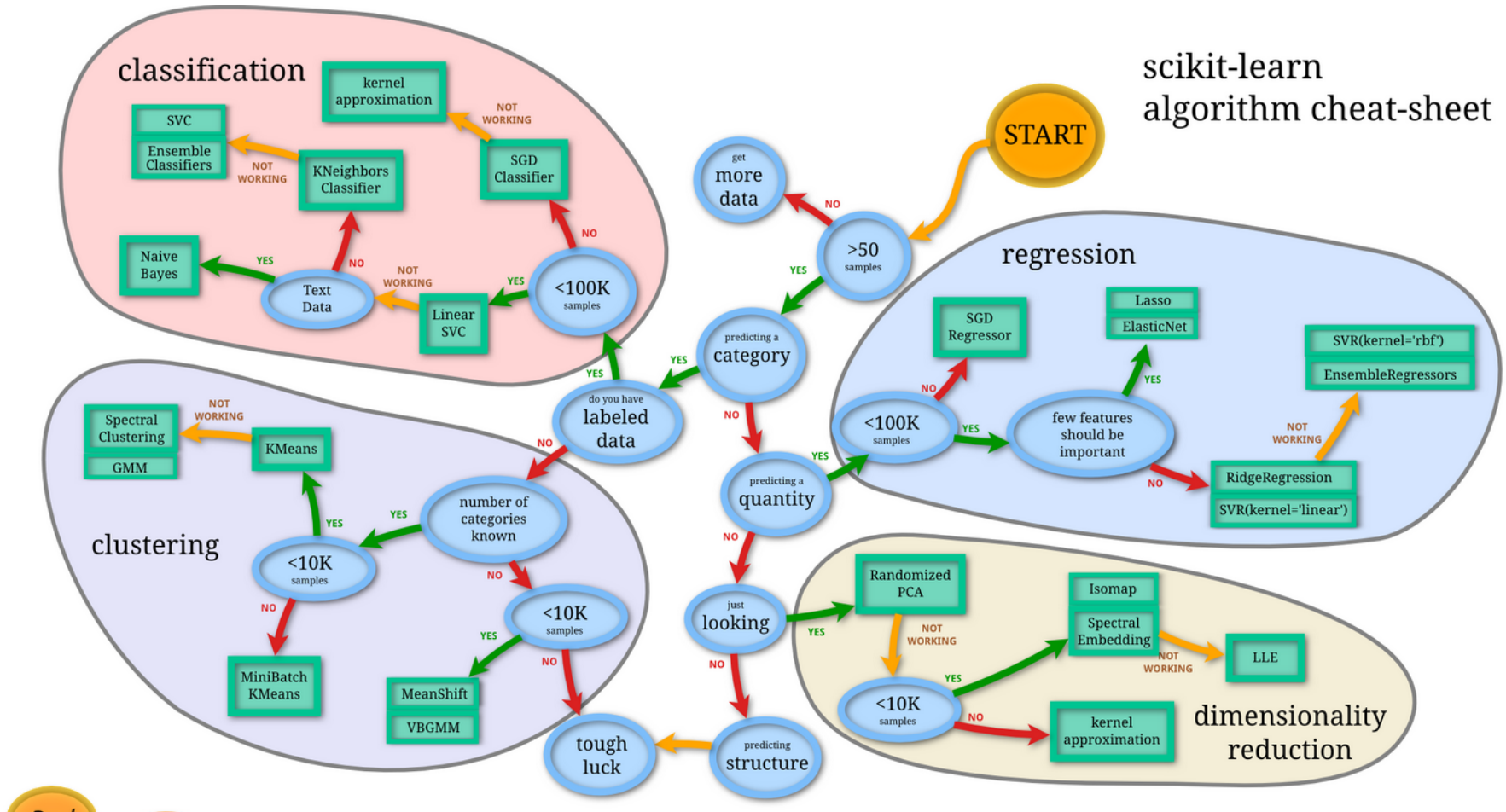

Data ingestion
validation
Data cleaning
validation
Data storage
Feature engineering
validation
Model training
Model testing
Model runs
Model output
validation
End user
Data preparation
Model development
Production deploy

# NFL Theorem

- In addition, there is the No-Free-Lunch Theorem which states:

*"All models perform equally when compared across all possible datasets."*

- which means that there is no best model that can be used in all datasets.


scikit-learn
algorithm cheat-sheet
START
get more data
>50 samples
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
YES
regression
SGD Regressor
NO
Lasso
ElasticNet
<100K samples
YES
few features should be important
YES
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
NO
RidgeRegression
SVR(kernel='linear')
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
dimensionality reduction
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
<10K samples
YES
NO
kernel approximation
Back
scikit learn

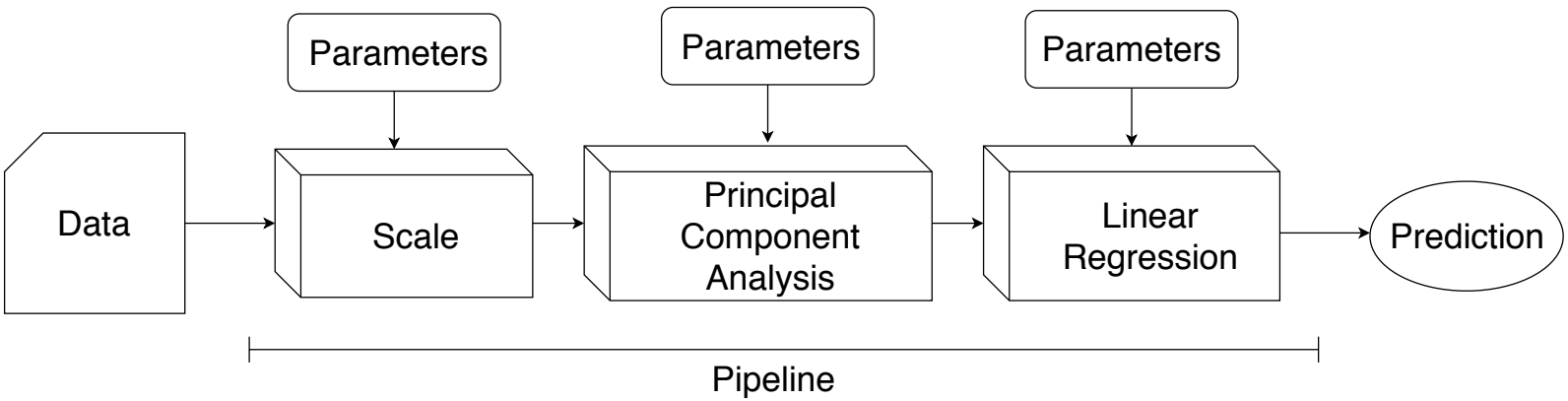
# Automated Machine Learning

- Automated Machine Learning (Auto-ML) aims at automating the ML pipeline
- Auto-ML can be seen as a search problem for a good configuration to meet some (*maybe unknown*) objective
- This work focus on **hyper-parameter search** and **algorithm selection**, as with most of the Auto-ML initiatives

# Topics

# Pipelines

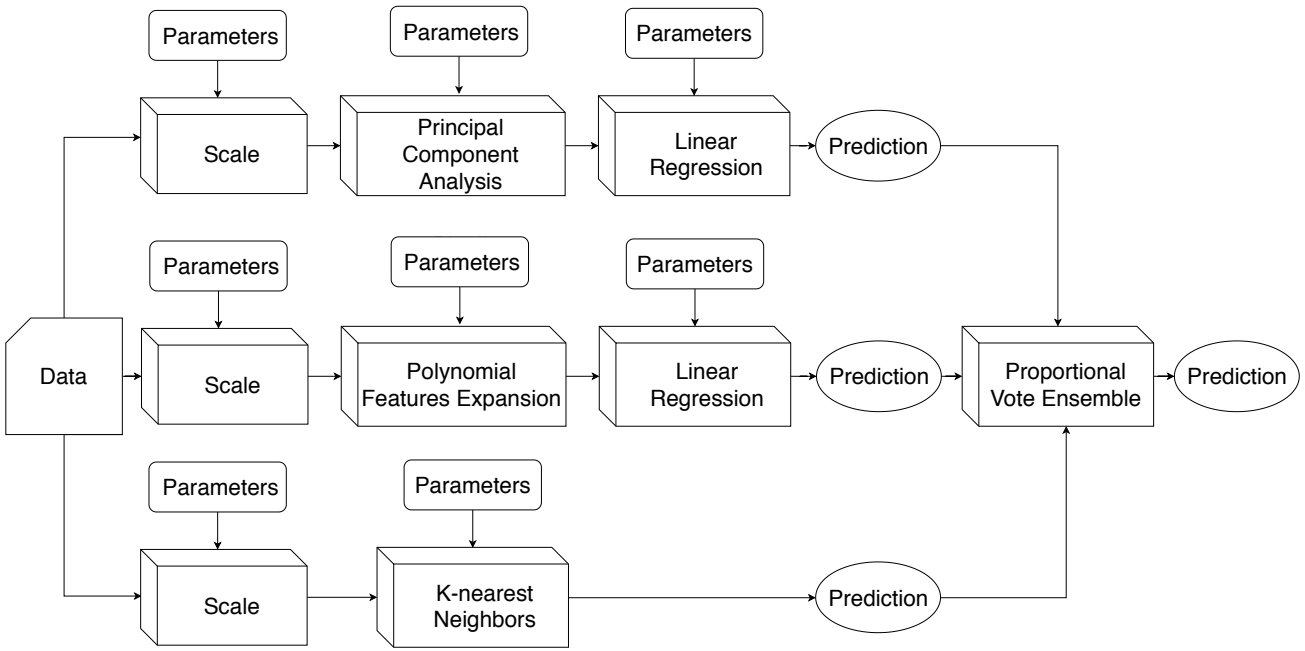
- Pipelines can be defined as a sequence of operators applied to a dataset in order to obtain a desired response




- Distinct pre-processing steps can result in distinc pipelines, even if the same machine learning technique is used

# Ensembles

- Ensembles are the combination of ML models in order to improve the performance




- The basic idea is that the weakness of each model will be compensated by the strength of the others

# Topics

# Topics

# GP in Machine Learning

- Genetic Programming applied to Machine learning is not something recent
- Several papers seeking to evolve **decision trees** to a specific problem
- The usual idea is to define a training procedure that has a better performance than the default heuristics used on the machine learning techniques

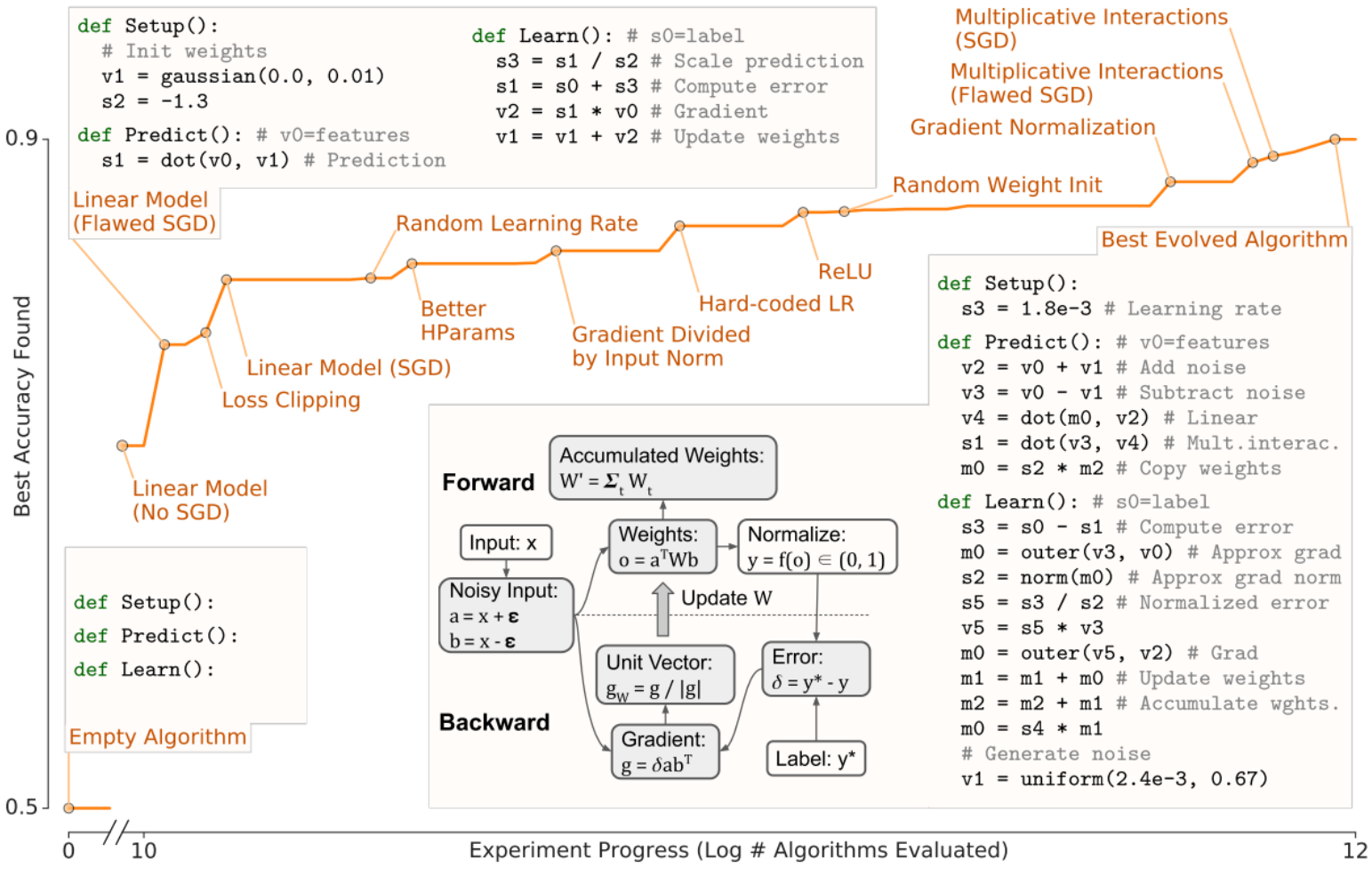
# Topics

# AutoML-Zero

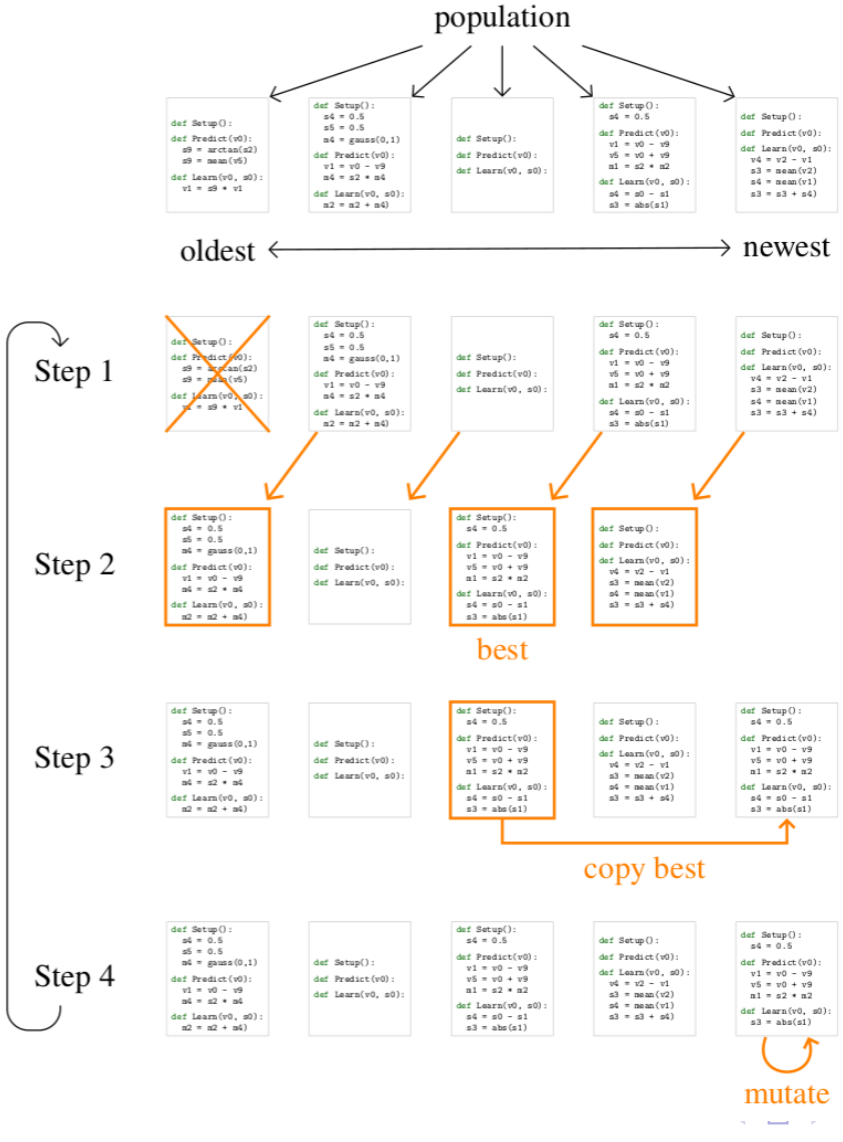
- AutoML-Zero tries to evolve machine learning algorithms from scratch
- It uses matrix vector operations as base components, creating training and predict procedures on the fly
- Google research authors

`https://github.com/google-research/google-research/tree/master/automl_zero`

# AutoML-Zero Evolution



# AutoML-Zero Population



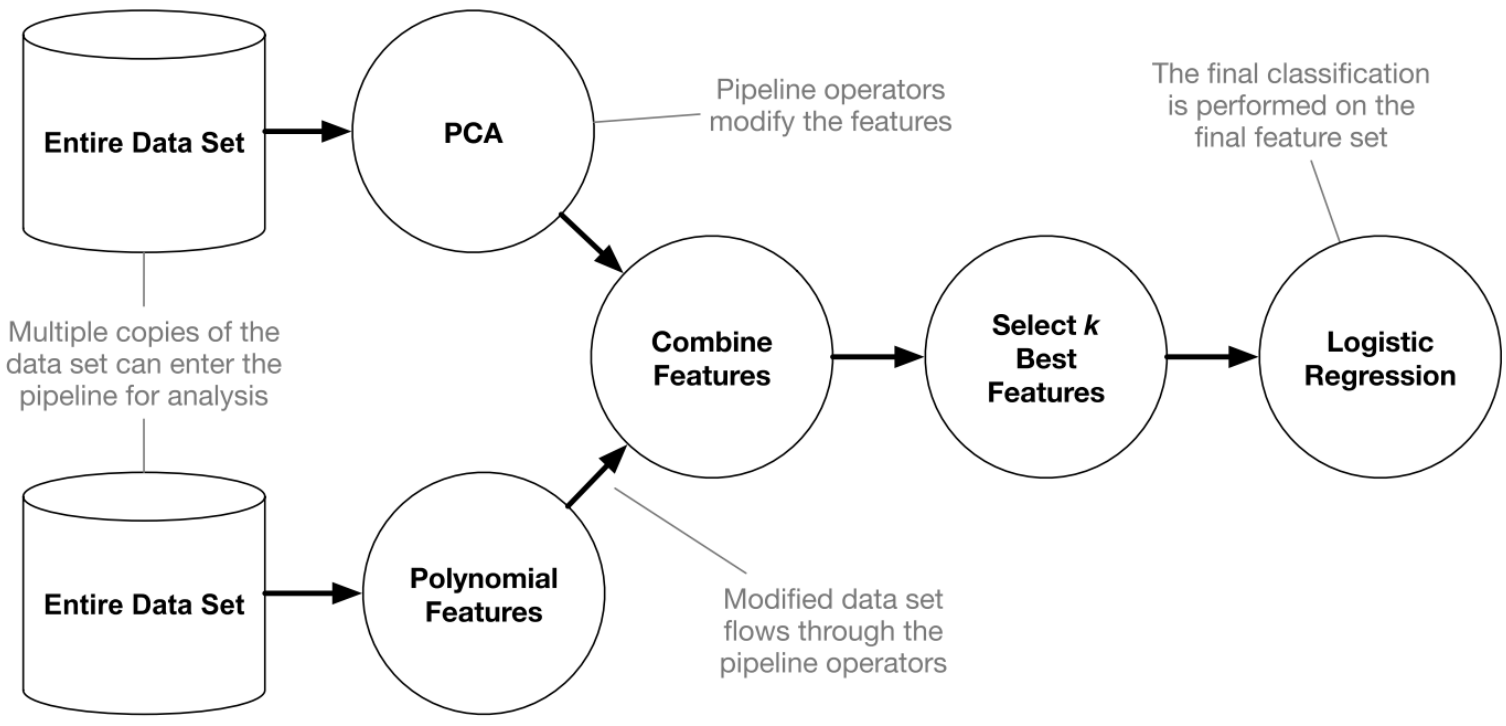
# Topics

# TPOT

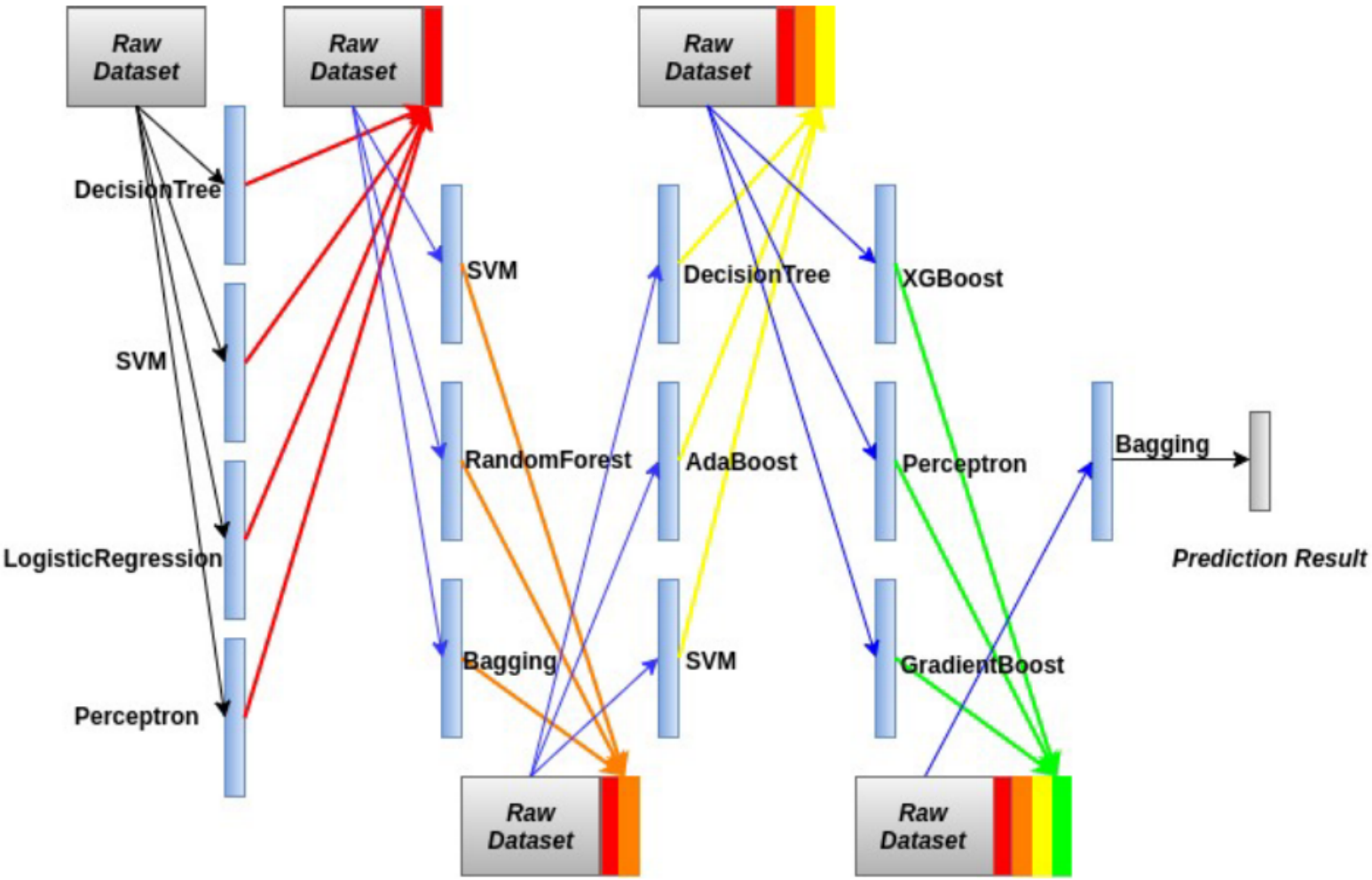
- Perhaps, the most famous Genetic Programming Auto-ML tool
- TPOT employs a Strongly Typed Genetic Programming implementation
- It applies a stacker like concept to build its models
- Also, it uses the DEAP library to implement the Genetic Programming procedure

`https://github.com/EpistasisLab/tpot`

# Base TPOT Pipeline



# Stacker example



# Some Comments on TPOT

- TPOT has a high computational cost
- Its models can have high complexity
- To mitigate the complexity problem, TPOT uses a multi-objective function where both accuracy and model complexity are pursuit
- Evolutionary operators are applied almost as default on GP

# Topics

# Coevolutionary Algorithms (a quick parenthesis)

- Aims at mimetizing the natural coevolution phenomena
- Uses two or more populations of solutions which interact among themselves in the evolutionary procedure
- Every solution is evaluated according to its interaction with other solutions
- Can be used, among other possibilities, to split a problem in subproblems that can be solved in parallel and, in some cases, requiring less computational resources than the full problem

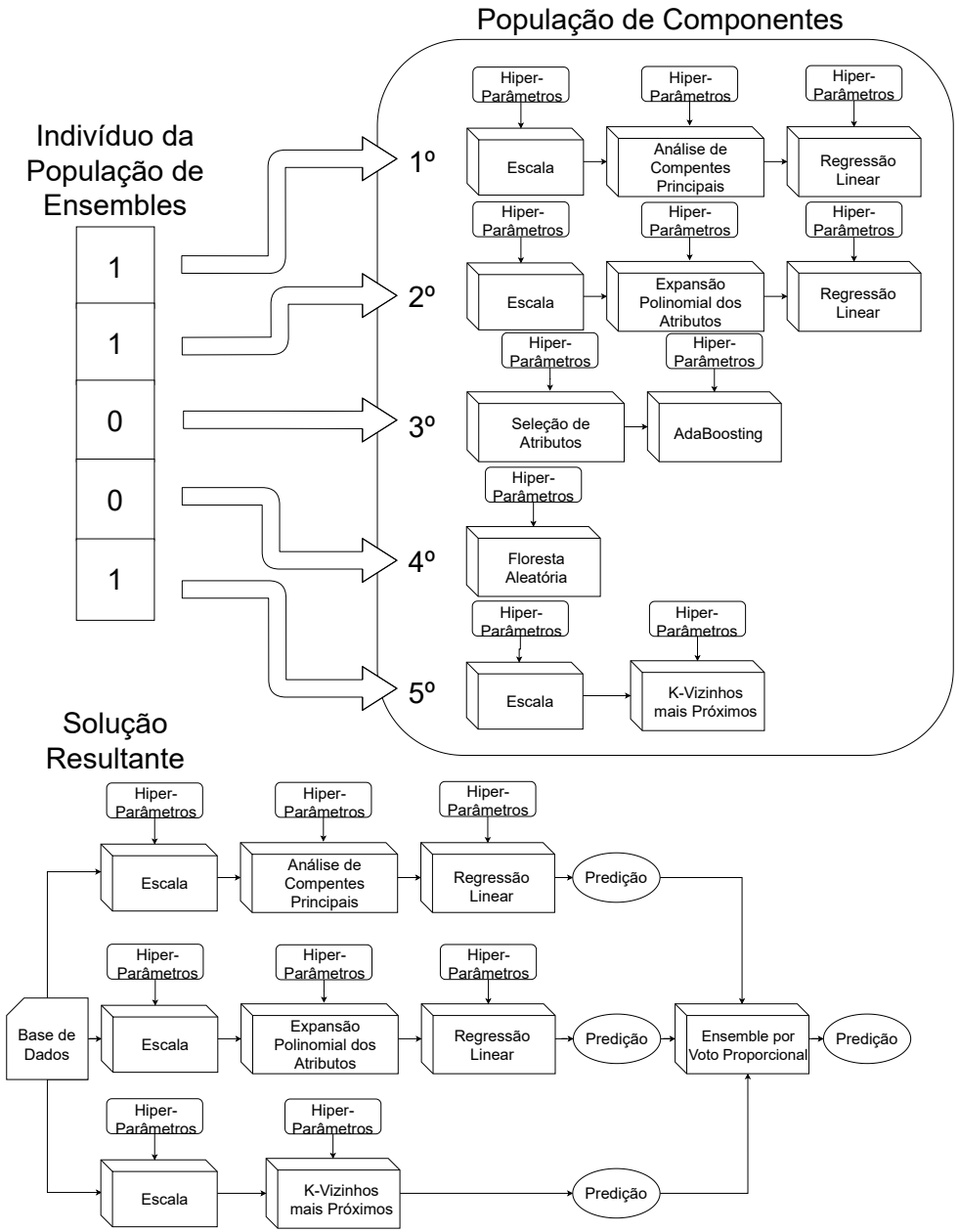
**Algorithm 1** 2-populations coevolutionary framework

1: $P_0^1 \leftarrow$ Initialize();
2: $P_0^2 \leftarrow$ Initialize();
3: $G \leftarrow 0$;
4: **while** stop criteria **do**
5: $\quad P_{G+1}^1 \leftarrow \text{NextGeneration}(P_G^1, P_G^2)$;
6: $\quad P_{G+1}^2 \leftarrow \text{NextGeneration}(P_{G+1}^1, P_G^2)$;
7: $\quad G \leftarrow G + 1$;
8: **end while**
9: **return** best pair $(p_G^1, p_G^2)$, $p_G^1 \in P_G^1$, $p_G^2 \in P_G^2$;

# Auto-CVE

- The Auto-CVE objective is building ensemble models composed by diverse pipelines
- To this end, it uses the populational coevolutionary framework to search for good configurations
- **IDEA**: Coevolve two distinct populations
  - **Components population** composed by pipelines: candidate models to be used by the ensembles
  - **Ensembles population** composed by hard voting ensembles with proportional vote: binary vectors that inform which components are part of each ensemble

`https://github.com/celiolarcher/AUTOCVE`


População de Componentes
Indivíduo da População de Ensembles
1
1
0
0
1
1º
2º
3º
4º
5º
Hiper-Parâmetros
Escala
Análise de Compentes Principais
Regressão Linear
Expansão Polinomial dos Atributos
Seleção de Atributos
AdaBoosting
Floresta Aleatória
K-Vizinhos mais Próximos
Solução Resultante
Base de Dados
Predição
Ensemble por Voto Proporcional

# Coevolutionary Procedure

- The process is performed through the default coevolutionary structure with one step per population:
  - The population of components is evolved using a Context Free Genetic Programming implementation
  - The population of ensembles is evolved through a Genetic Algorithm approach
- The process runs until the stop criterion is reached
- At the end, the best ensemble (composed by the chosen components) is returned

# Evolution of the components

- A Context Free Genetic Programming implementation
- The evaluation is made through a niching scheme, the deterministic crowding
- Traditional mutation and crossover operators:
  - Mutation: delete and reconstruction of a random subtree
  - Crossover: swap between two subtrees with the same non-terminal
- The fitness of each component is given by the mean performance of the ensembles that the component is part of (or of the component itself, if it is not part of any ensemble)

# Grammar applied

- A sample of the used grammar:

```
<pipeline> ::= <add_features> "->" <classifier>
| <add_features> "->" <selector> "->" <classifier>
| <scaler> "->" <preprocessing> "->" <classifier>
| <scaler> "->" <preprocessing> "->" <selector> "->" <classifier>
| <classifier>

<classifier> ::= <NaiveBayes>
| <Tree>
| <scaler> "->" <KNN>
| <scaler> "->" <FunctionalClassifier>
```

- Every non-terminal has a list of techniques
- Also, a list of hyper-parameters for each technique is included in the grammar

# Topics

# Notebook Time!

https://github.com/celiolarcher/knowledge_repository/tree/main/genetic_programming_automl

# Topics

# Conclusions

- Auto-ML is a hot research topic
- Genetic programming can and indeed is heavily applied in the machine learning field
- Several python libraries and papers to follow

# Research Perspectives

- AutoMLZero
  - Regression tasks
  - Adding more operations to the algorithms
  - Implementing crossover
- TPOT
  - Reducing computational time
  - Meta-learning to initialize the population
  - Customizing grammar to specific tasks
- AutoCVE
  - Regression tasks
  - Meta-learning to initialize the population
  - More operators and personalized grammar

# Topics

# References I


Larcher, C., Barbosa, H. (2019).
Auto-CVE: a coevolutionary approach to evolve ensembles in automated machine learning. *GECCO.*


Larcher, C., Barbosa, H. (2021).
Evaluating Models with Dynamic Sampling Holdout. *Evostar.*


Real, E., Liang, C., So, D. R., Le, Q. V. (2020).
AutoML-Zero: Evolving Machine Learning Algorithms From Scratch. *ArXiv.*


Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., Moore, J. H. (2016).
Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. *Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics.*

# Thank you for your attention!

**Contact:** clarcher@lncc.br
**Github:** `https://github.com/celiolarcher`