



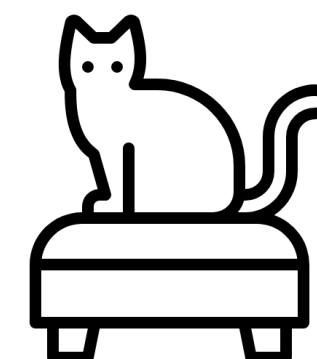
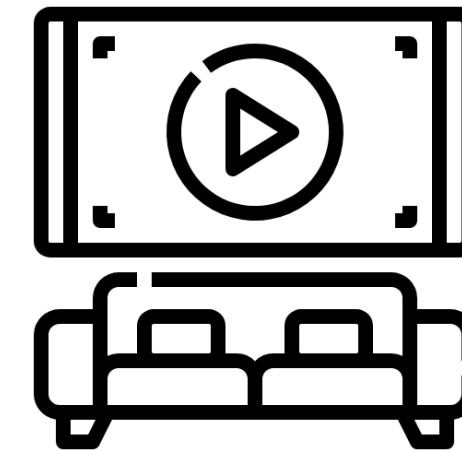
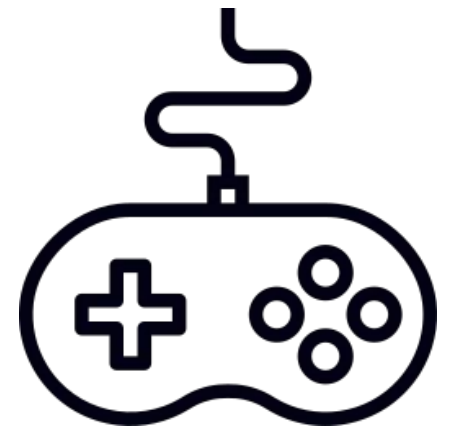
THE BLACK BOX OF THE BLACK BOXES

UMA INTRODUÇÃO A AUTOML COM PYTHON

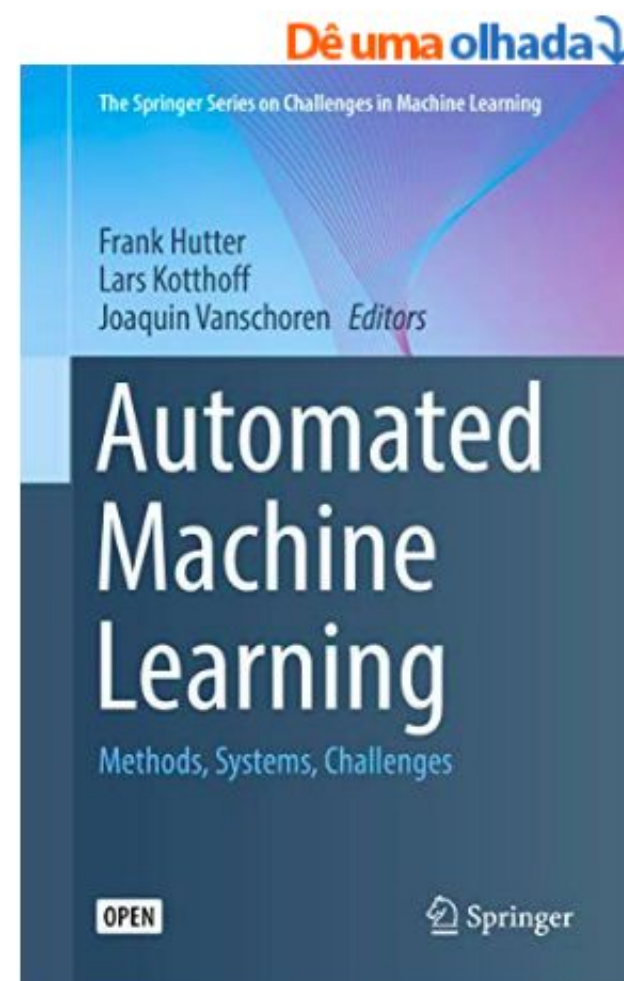
JOSE ANDRADE

```
[~] $ whoami
```

```
{  
  "Name": "José Andrade",  
  "City": Natal,  
  "Job": "Data Scientist",  
  "Company": "Data Science Brigade",  
  "Github": http://github.com/andradejunior  
  "Linkedin": http://linkedin.com/in/andrade-junior,  
  "Twitter": @4ndradejr,  
  "Telegram": @andradejunior  
}
```



// Dica de leitura



Automated Machine Learning: Methods, Systems, Challenges (The Springer Series on Challenges in Machine Learning) (English Edition) eBook Kindle

por Frank Hutter (Autor, Editor), Lars Kotthoff (Autor, Editor), Joaquin Vanschoren (Autor, Editor) | Formato: eBook Kindle

★★★★★ 5 classificações

> Ver todos os 2 formatos e edições

Kindle
R\$0,00

Capa dura
R\$498,99 ✓prime

Leia com nossos apps gratuitos

2 Novo a partir de R\$498,99

This open access book presents the first comprehensive overview of general methods in Automated Machine Learning (AutoML), collects descriptions of existing systems based on these methods, and discusses the first series of international challenges of AutoML systems. The recent success of commercial ML applications and the rapid growth of the field has created a high demand for off-the-shelf ML methods that can be used easily and without expert knowledge. However, many of the recent machine learning successes crucially rely on human experts, who manually select appropriate ML architectures (deep learning architectures or more traditional ML workflows) and their hyperparameters. To overcome this problem, the field of AutoML targets a progressive automation of machine learning, based on principles from optimization and machine learning itself. This book serves as a point of entry into this quickly-developing field for researchers and advanced students alike, as well as providing a reference for practitioners aiming to use AutoML in their work.

<https://www.amazon.com.br/gp/product/B07S3MLGFW>

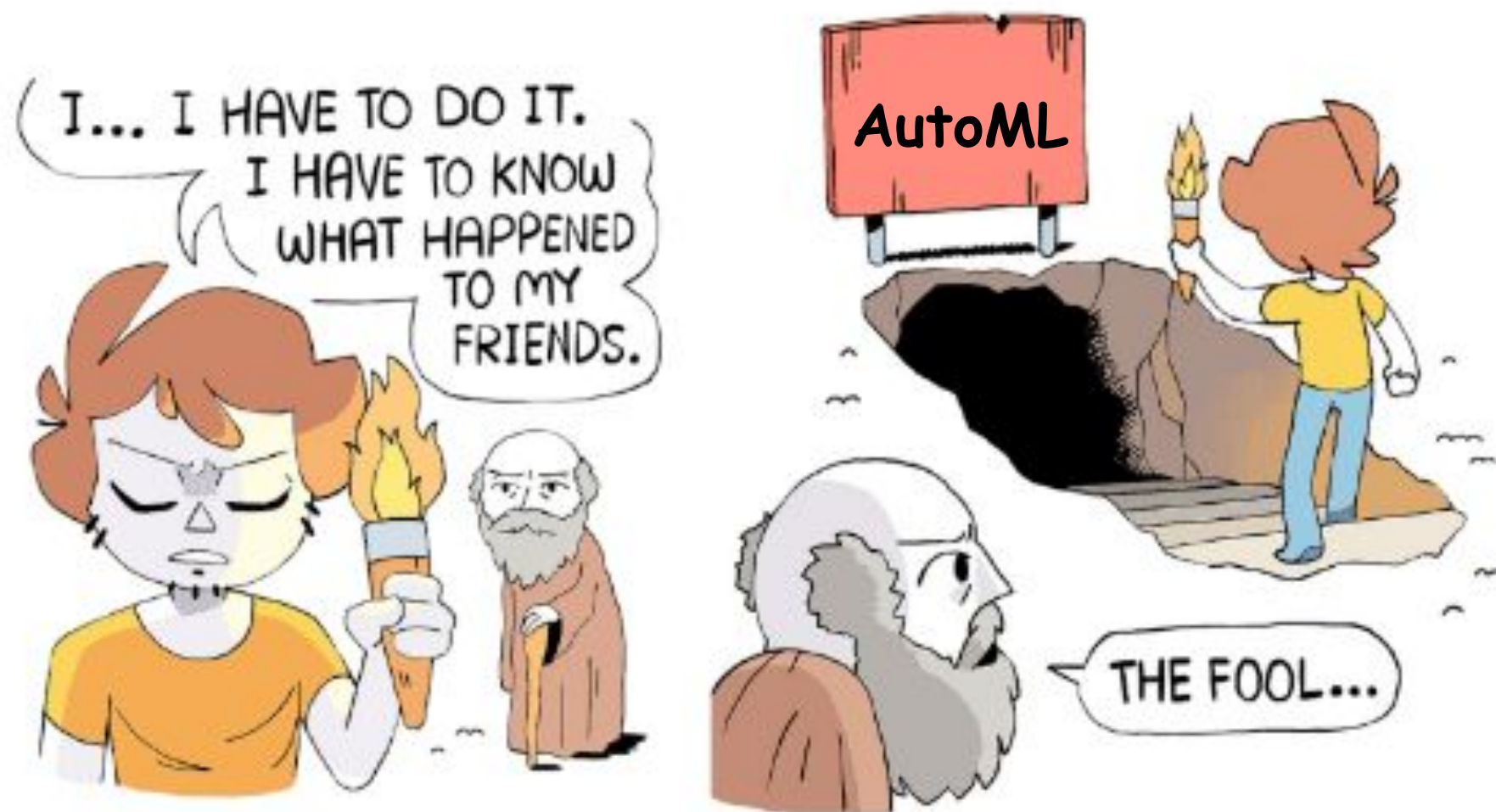
<https://link.springer.com/book/10.1007%2F978-3-030-05318-5>

1 **Motivação**

2 **Métodos de AutoML**

3 **Sistemas de AutoML**

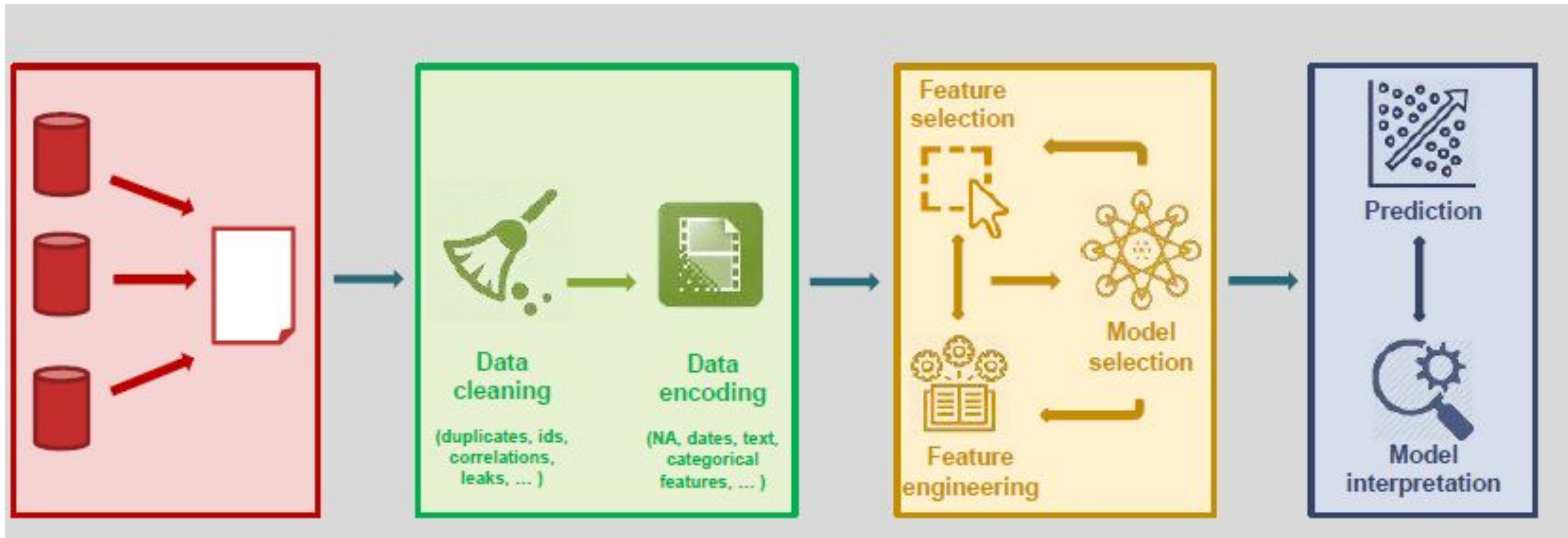
4 **Conclusões**



1

Motivação

// Motivação



// Motivação

Problema

- Alta demanda na área de IA e baixa oferta de mão de obra especializada.

// Motivação

Problema

- Alta demanda na área de IA e baixa oferta de mão de obra especializada.

Expectativa: 58 milhões de empregos em IA até 2022, de acordo com o [Fórum Econômico Mundial](https://www.forbes.com/sites/amitchowdhry/2018/09/18/artificial-intelligence-to-create-58-million-new-jobs-by-2022-says-report/#c07b70c4d4ba).

<https://www.forbes.com/sites/amitchowdhry/2018/09/18/artificial-intelligence-to-create-58-million-new-jobs-by-2022-says-report/#c07b70c4d4ba>

// Motivação

Problema

- Alta demanda na área de IA e baixa oferta de mão de obra especializada.

Expectativa: 58 milhões de empregos em IA até 2022, de acordo com o [Fórum Econômico Mundial](https://www.forbes.com/sites/amitchowdhry/2018/09/18/artificial-intelligence-to-create-58-million-new-jobs-by-2022-says-report/#c07b70c4d4ba).

<https://www.forbes.com/sites/amitchowdhry/2018/09/18/artificial-intelligence-to-create-58-million-new-jobs-by-2022-says-report/#c07b70c4d4ba>

Estima-se que em 2017 havia 300 mil engenheiros de IA no mundo, enquanto milhões são necessário de acordo com o [Instituto de Pesquisa de Tencent](https://www.theverge.com/2017/12/5/16737224/global-ai-talent-shortfall-tencent-report).

<https://www.theverge.com/2017/12/5/16737224/global-ai-talent-shortfall-tencent-report>

// Motivação

Problema

- Alta demanda na área de IA e baixa oferta de mão de obra especializada.

Solução

AutoML

- Engenharia de características
- Seleção de algoritmos
- Otimização de hiperparâmetros



DEFINIÇÃO

Processo de automatizar o passo a passo da aplicação de aprendizado de máquina a problemas do mundo real.

2

Métodos

// Métodos de AutoML



Hyperparameter Optimization

- CASH: Combined Algorithm Selection and Hyperparameter Optimization

// Métodos de AutoML



Hyperparameter Optimization

- CASH: Combined Algorithm Selection and Hyperparameter Optimization



Meta-Learning

- Aprender a aprender

// Métodos de AutoML



Hyperparameter Optimization

- CASH: Combined Algorithm Selection and Hyperparameter Optimization



Meta-Learning

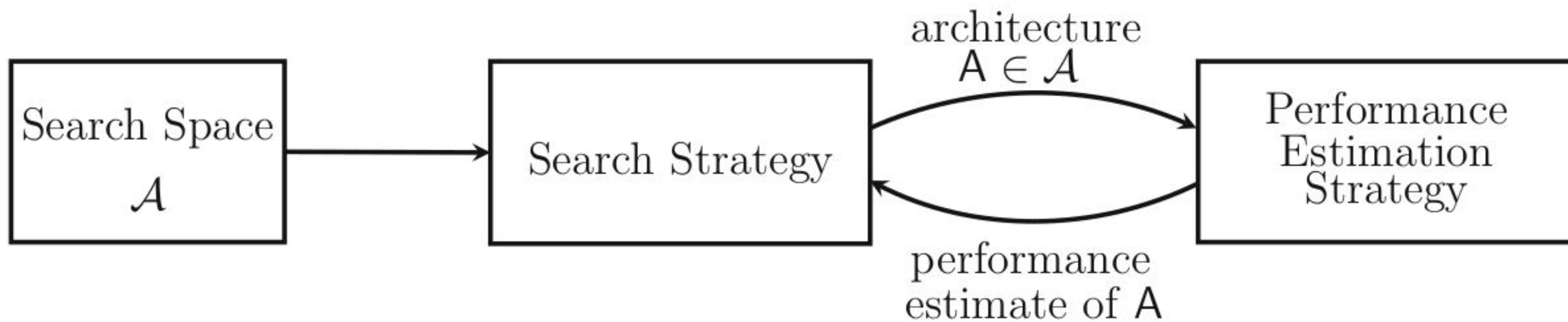
- Aprender a aprender



Neural Architecture Search

- Search space
- Search strategy
- Performance estimation strategy

// Neural Architecture Search



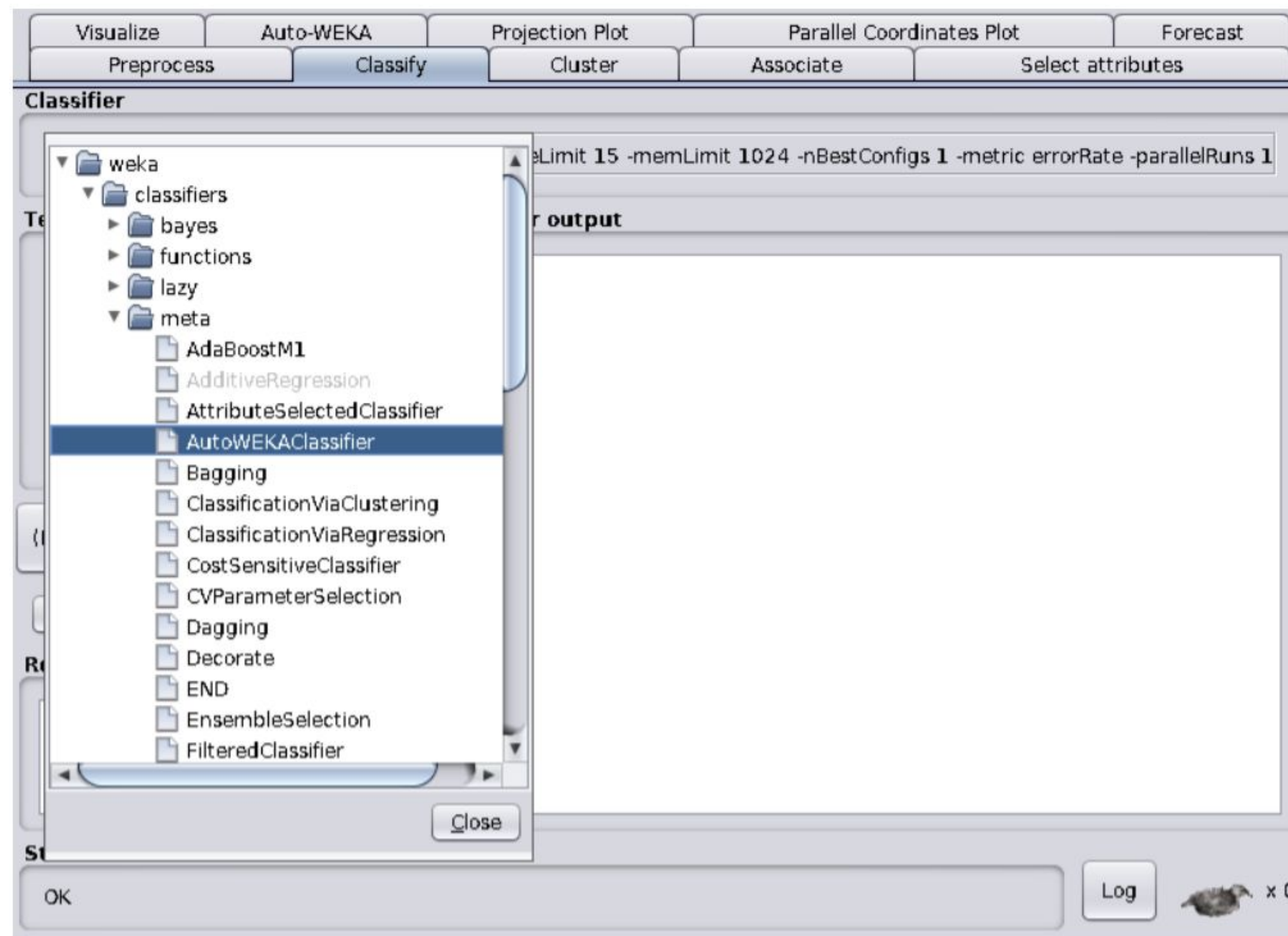
3

Sistemas

// Auto-WEKA



// Auto-WEKA



// Auto-WEKA

The screenshot shows the Auto-WEKA GUI with the 'Auto-WEKA' tab selected. The 'Auto-WEKA output' pane displays the results of a classification task. The 'Result list' on the left shows a single result: '16:14:02 - Auto-WEKA: Iris'.

Auto-WEKA output

Auto-WEKA result:
best classifier: weka.classifiers.lazy.LWL
arguments: [-U, 0, -A, weka.core.neighboursearch.LinearNNSearch, -W, weka.classifiers.functions.Logistic, --, -R, 6
attribute search: weka.attributeSelection.GreedyStepwise
attribute search arguments: [-C, -B, -R]
attribute evaluation: weka.attributeSelection.CfsSubsetEval
attribute evaluation arguments: []
metric: errorRate
estimated errorRate: 0.013333333333333334
training time on evaluation dataset: 0.0 seconds

You can use the chosen classifier in your own code as follows:

```
AttributeSelection as = new AttributeSelection();
ASearch asSearch = ASearch.forName("weka.attributeSelection.GreedyStepwise", new String[]{"-C", "-B", "-R"});
as.setSearch(asSearch);
ASEvaluation asEval = ASEvaluation.forName("weka.attributeSelection.CfsSubsetEval", new String[]{});
as.setEvaluator(asEval);
as.SelectAttributes(instances);
instances = as.reduceDimensionality(instances);
Classifier classifier = AbstractClassifier.forName("weka.classifiers.lazy.LWL", new String[]{"-U", "0", "-A", "weka.classifier.buildClassifier(instances);
```

Correctly Classified Instances 148 98.6667 %
Incorrectly Classified Instances 2 1.3333 %
Kappa statistic 0.98
Mean absolute error 0.0199
Root mean squared error 0.0899
Relative absolute error 4.4858 %
Root relative squared error 19.0622 %
Total Number of Instances 150

=== Confusion Matrix ===

```
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 49 1 | b = Iris-versicolor
0 1 49 | c = Iris-virginica
```

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-setosa
	0.980	0.010	0.980	0.980	0.980	0.970	0.999	0.998	Iris-versicolor
	0.980	0.010	0.980	0.980	0.980	0.970	0.999	0.998	Iris-virginica
Weighted Avg.	0.987	0.007	0.987	0.987	0.987	0.980	0.999	0.998	

For better performance, try giving Auto-WEKA more time.

Status: OK

// Hyperopt-Sklearn



```
from hpsklearn import HyperoptEstimator

estim = HyperoptEstimator()
estim.fit(X_train, y_train)

prediction = estim.predict(X_test)

score = estim.score(X_test, y_test)

model = estim.best_model()
```

// Hyperopt-Sklearn

- Domínio de busca
- Função objetivo
- Algoritmo de otimização

// Hyperopt-Sklearn

- Domínio de busca
- Função objetivo
- Algoritmo de otimização



```
space = hp.choice('my_conditional',  
[  
    ('case 1', 1 + hp.lognormal('c1', 0, 1)),  
    ('case 2', hp.uniform('c2', -10, 10))  
    ('case 3', hp.choice('c3', ['a', 'b', 'c']))  
)
```




- Domínio de busca
- Função objetivo → accuracy, F1-score, etc
- Algoritmo de otimização

// Hyperopt-Sklearn

- Domínio de busca
- Função objetivo
- Algoritmo de otimização
 - Random Search
 - Tree of Parzen Estimators (TPE)
 - Annealing
 - Tree
 - Gaussian Tree Process



```
from hpsklearn import HyperoptEstimator
from hyperopt import tpe

estim = HyperoptEstimator(algo=tpe.suggest,
                          max_evals=150,
                          trial_timeout=60)
```

// Hyperopt-Sklearn

Classificadores:

- SVC
- LinearSVC
- KNeighborsClassifier
- RandomForestClassifier
- ExtraTreesClassifier
- SGDClassifier
- MultinomialNB
- BernoulliRBM
- ColumnKMeans



```
from hpsklearn import HyperoptEstimator, svc  
  
estim = HyperoptEstimator(classifier=svc( 'mySVC' ))
```

// Hyperopt-Sklearn



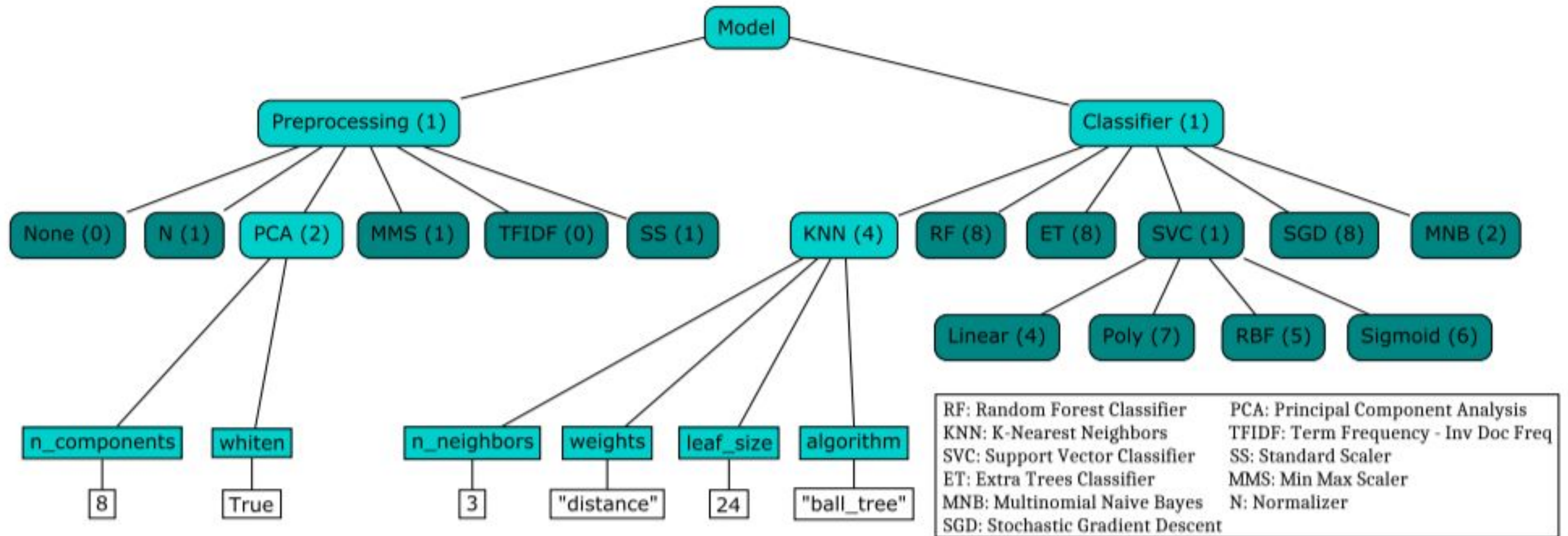
```
from hpsklearn import HyperoptEstimator, any_sparse_classifier, tfidf
from hyperopt import tpe

estim = HyperoptEstimator(classifier=any_sparse_classifier('clf'),
                          preprocessing=[tfidf('tfidf')],
                          algo=tpe.suggest, trial_timeout=300)

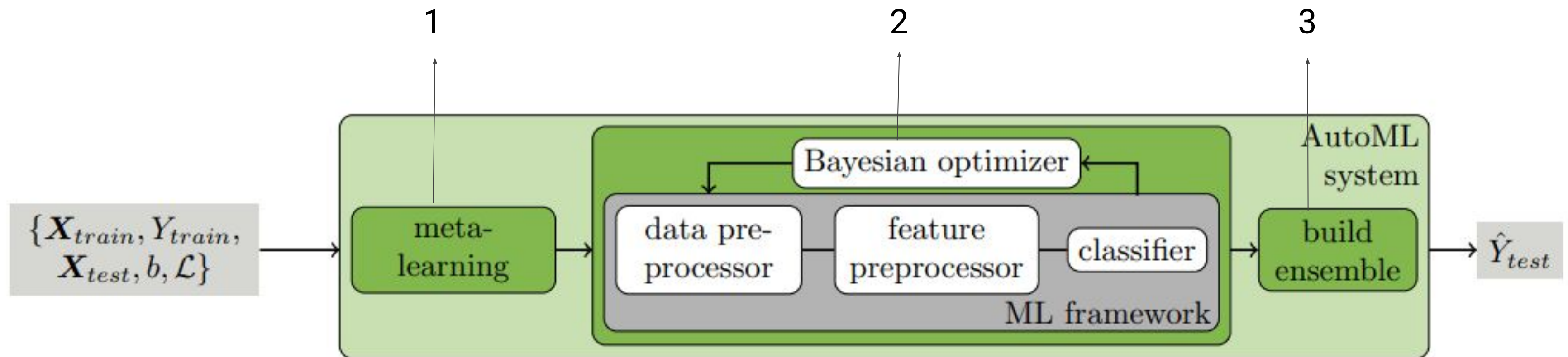
estim.fit(x_train, y_train)

print(estim.score(x_test, y_test))
print(estim.best_model())
```

// Hyperopt-Sklearn



// Auto-Sklearn



// Auto-Sklearn

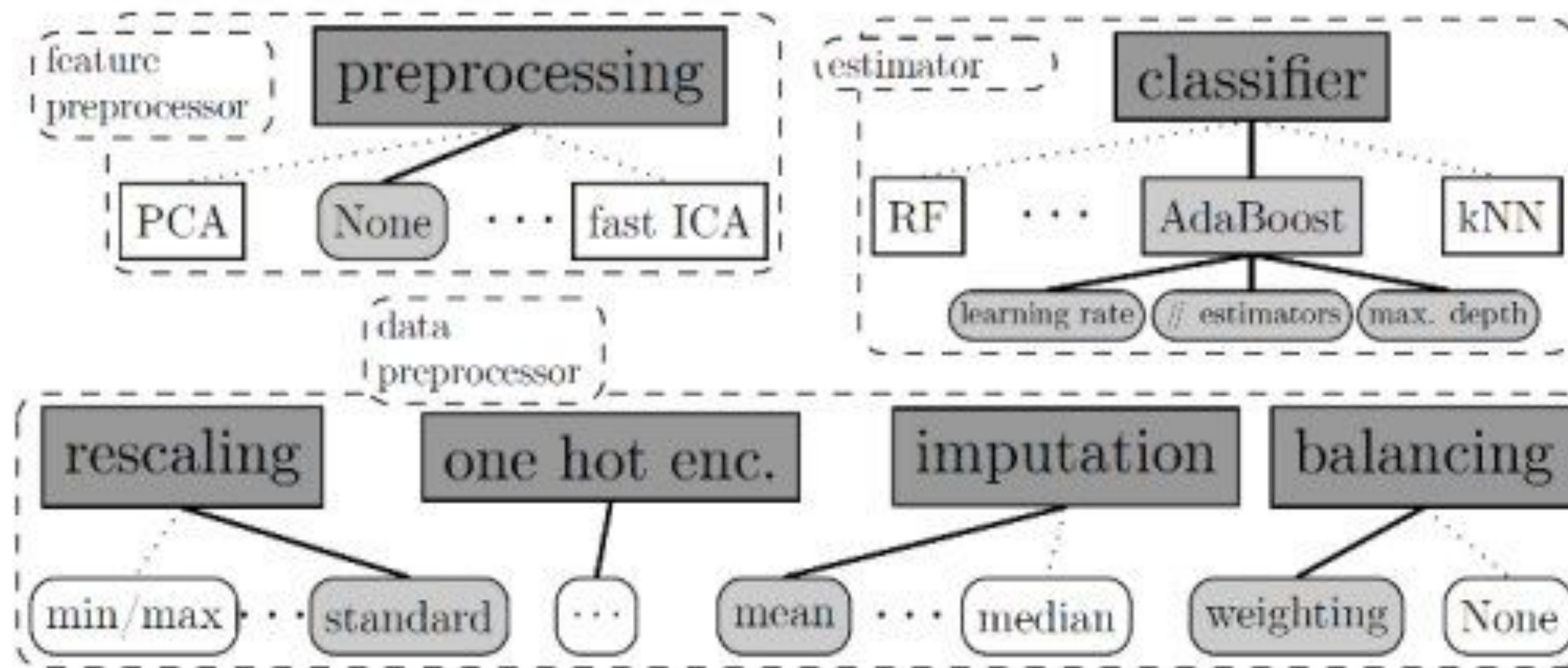
name	#λ	cat (cond)	cont (cond)
AdaBoost (AB)	4	1 (-)	3 (-)
Bernoulli naïve Bayes	2	1 (-)	1 (-)
decision tree (DT)	4	1 (-)	3 (-)
extreml. rand. trees	5	2 (-)	3 (-)
Gaussian naïve Bayes	-	-	-
gradient boosting (GB)	6	-	6 (-)
kNN	3	2 (-)	1 (-)
LDA	4	1 (-)	3 (1)
linear SVM	4	2 (-)	2 (-)
kernel SVM	7	2 (-)	5 (2)
multinomial naïve Bayes	2	1 (-)	1 (-)
passive aggressive	3	1 (-)	2 (-)
QDA	2	-	2 (-)
random forest (RF)	5	2 (-)	3 (-)
Linear Class. (SGD)	10	4 (-)	6 (3)

(a) classification algorithms

name	#λ	cat (cond)	cont (cond)
extreml. rand. trees prepr.	5	2 (-)	3 (-)
fast ICA	4	3 (-)	1 (1)
feature agglomeration	4	3 ()	1 (-)
kernel PCA	5	1 (-)	4 (3)
rand. kitchen sinks	2	-	2 (-)
linear SVM prepr.	3	1 (-)	2 (-)
no preprocessing	-	-	-
nystroem sampler	5	1 (-)	4 (3)
PCA	2	1 (-)	1 (-)
polynomial	3	2 (-)	1 (-)
random trees embed.	4	-	4 (-)
select percentile	2	1 (-)	1 (-)
select rates	3	2 (-)	1 (-)
one-hot encoding	2	1 (-)	1 (1)
imputation	1	1 (-)	-
balancing	1	1 (-)	-
rescaling	1	1 (-)	-

(b) preprocessing methods

// Auto-Sklearn



// Auto-Sklearn



```
import autosklearn.classification

cls = autosklearn.classification.AutoSklearnClassifier( )
cls.fit(X_train, y_train)

predictions = cls.predict(X_test)
```

// Auto-Keras

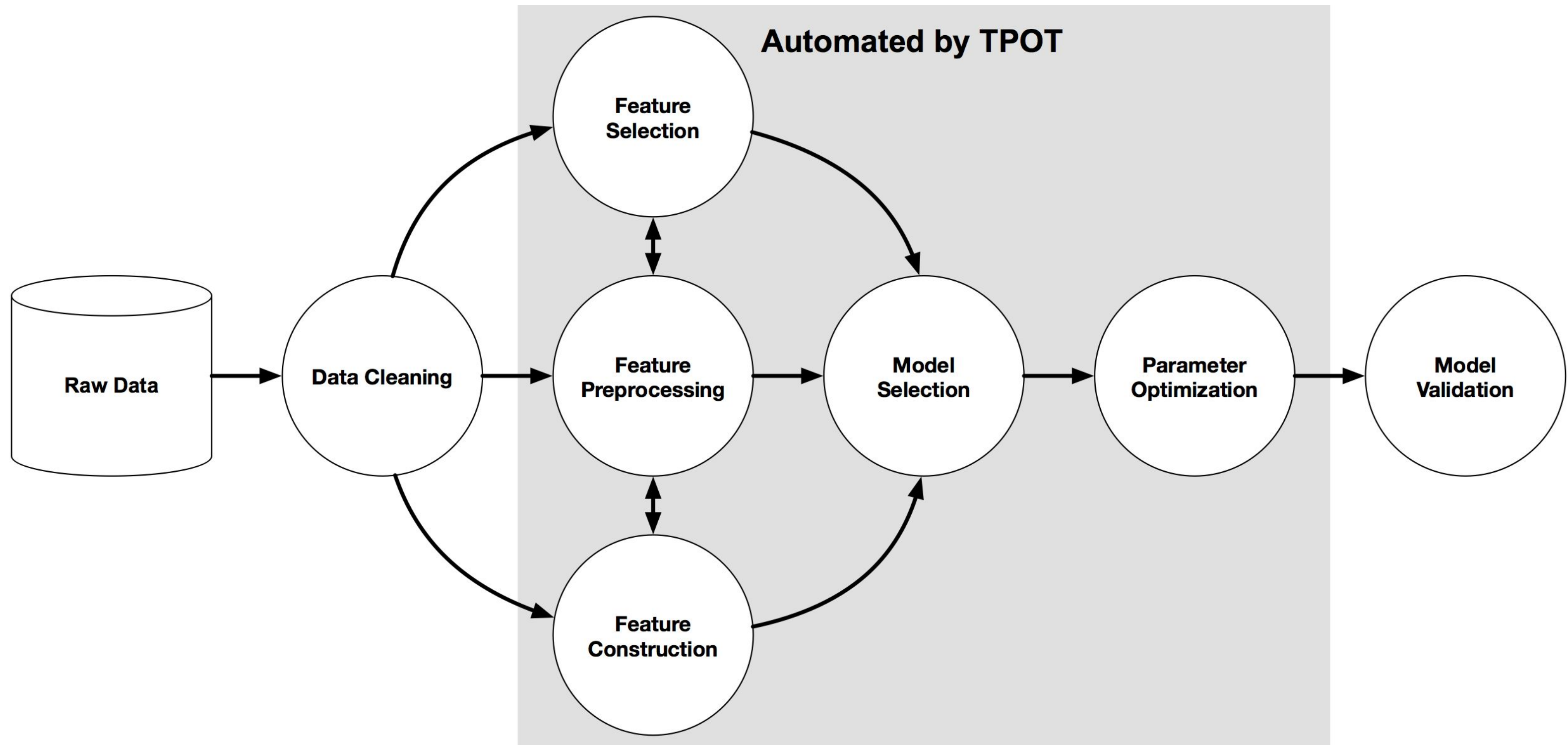


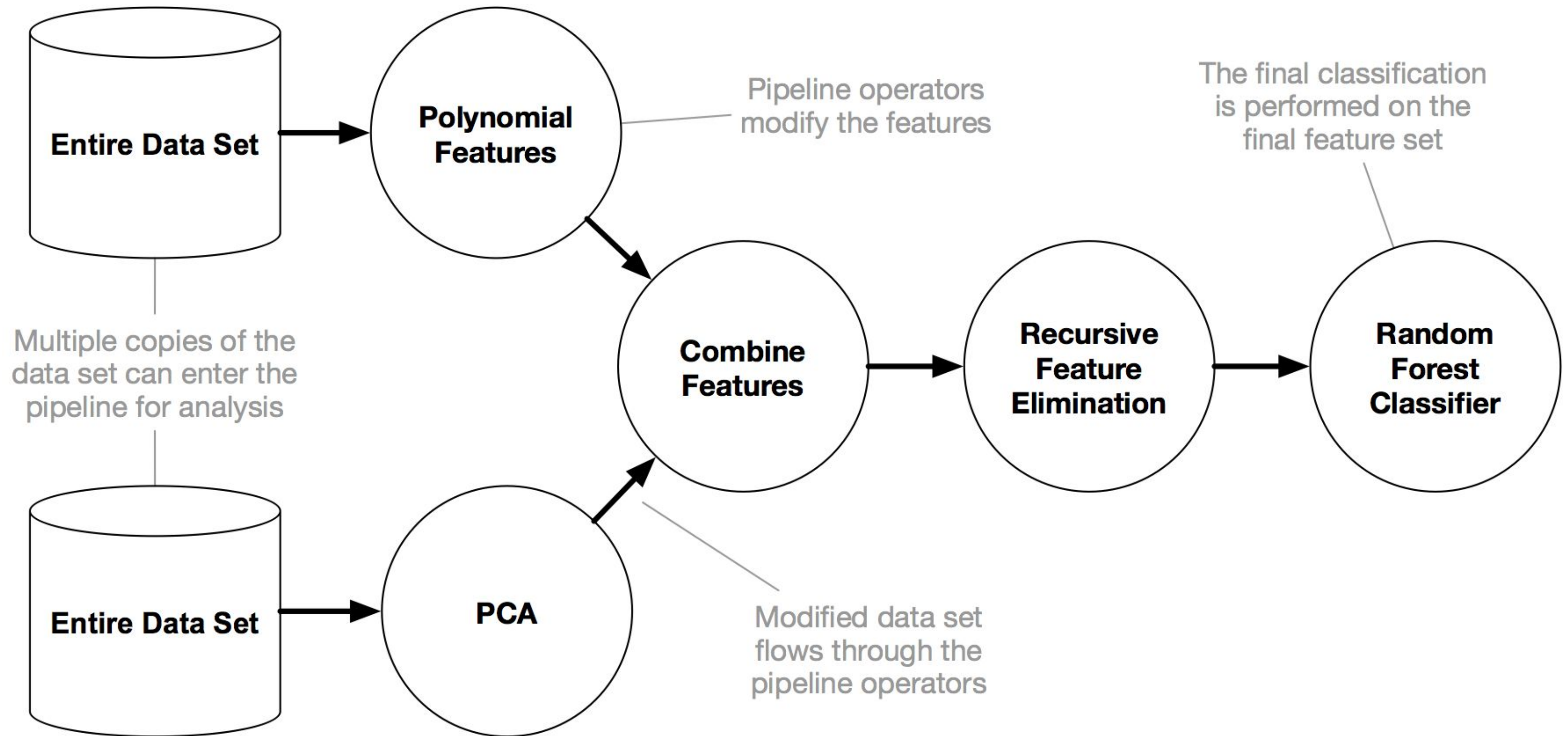
```
import autokeras as ak

clf = ak.TextClassifier(max_trials=3)
clf.fit(x_train, y_train, epochs=2)

predicted_y = clf.predict(x_test)

print(clf.evaluate(x_test, y_test))
```







- H2O é uma biblioteca de ML escrita em Java que possui bibliotecas para uso em R e Python
- Busca trazer ML de forma escalável e facilitar o processo de deploy
- H2O AutoML é um módulo que inclui treinamento e tuning de diversos modelos dado um tempo limite especificado pelo usuário
- Busca aleatória seguida de criação de stacked ensembles

// H2O AutoML



Erin LeDell
@ledell

I had fun playing with @h2oai #AutoML on the #KaggleDaysSF hackathon today. One line of code, 8th place!

Ran H2O AutoML for 100 mins: it trained & 5-fold CV 43 models & 2 stacked ensembles. Wish I had joined the comp earlier & run longer! 🕒

Code here: gist.github.com/ledell/4d4cd24...

Traduzir Tweet

#	△pub	Team Name	Score ?	Entries	Last
1	▲30	Erkut & Mark	0.61691	12	2h
2	▲1	Google AutoML	0.61598	8	3h
3	▼2	Sweet Deal	0.61576	20	2h
4	▲11	Arno Candel @ H2O.ai	0.61549	17	2h
5	▼1	ALDAPOP	0.61504	11	2h
6	▲12	9hr Overfitness	0.61437	17	2h
7	▼5	Shlandryn	0.61413	38	2h
8	▲2	Erin (H2O AutoML 100 mins)	0.61312	5	2h
9	▼2	[ods.ai] bestfitting	0.61298	27	2h

<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>



Data
Preparation



Model
Training



Hyperparameter
Tuning



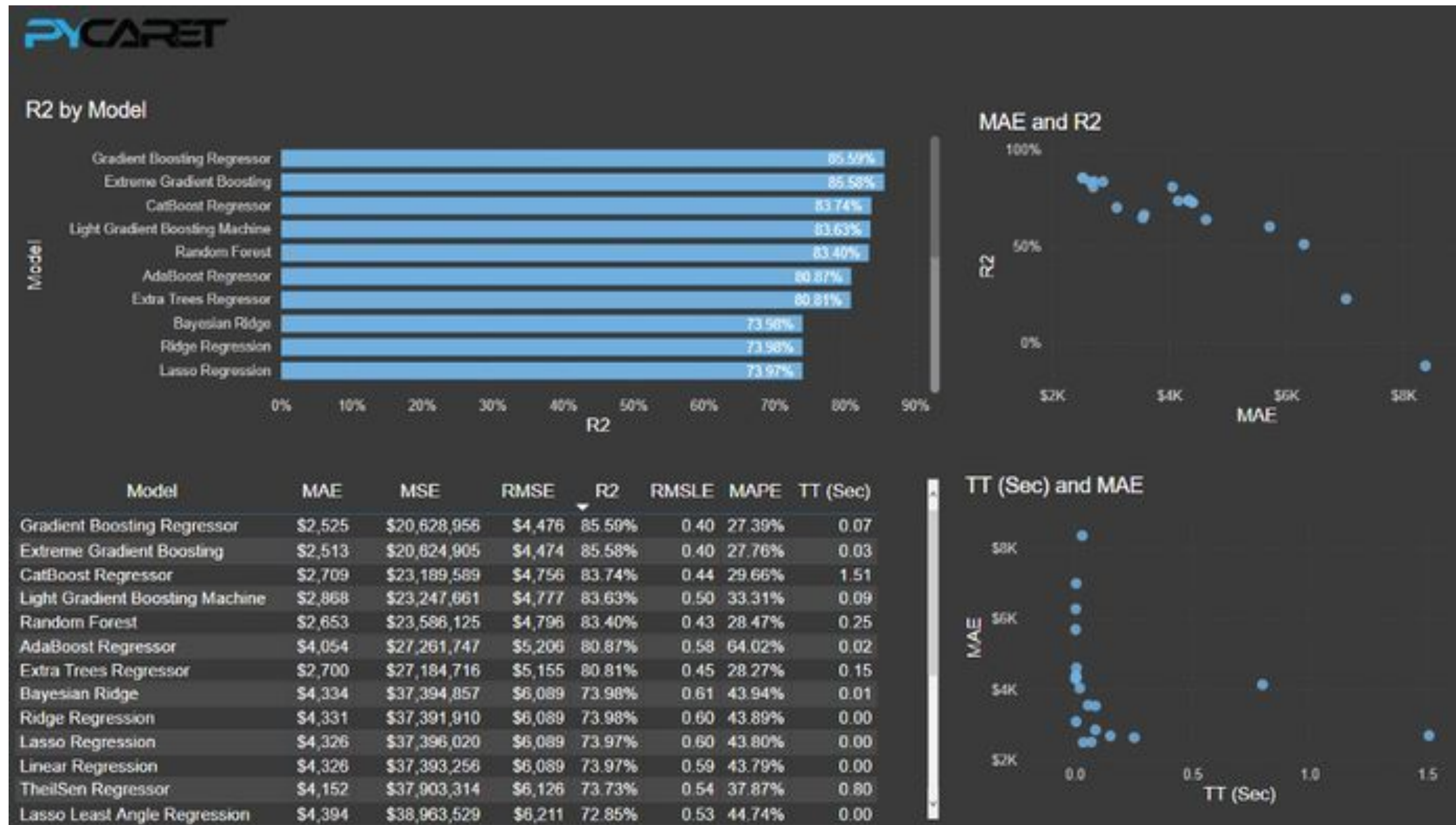
Analysis &
Interpretability



Model
Selection



Experiment
Logging



// Show me the code



// Conclusões

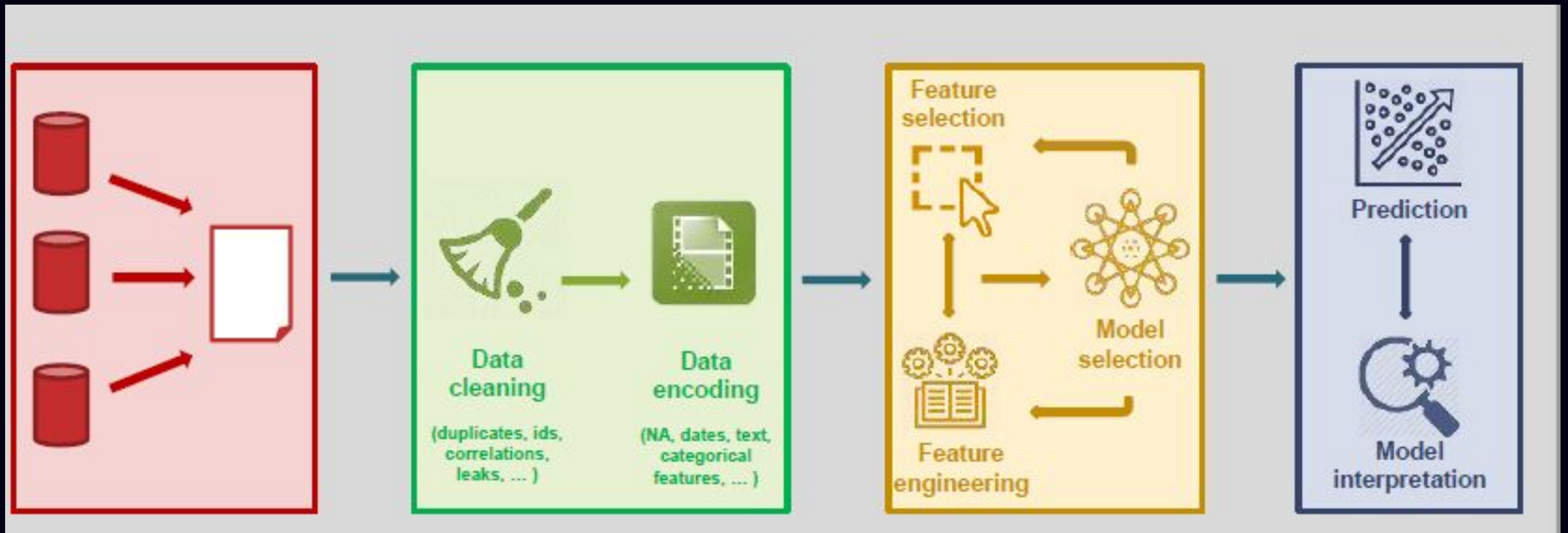


// Conclusões



Limitações

- Não alcança o estado da arte em diversas tarefas, principalmente em relação a processamento de texto, imagens e voz.
 - Problemas de transparência.
 - Ainda não há implementações de AutoML para Sistemas de recomendação e Reinforcement learning.
 - Não possui conhecimento de negócio
-

// Conclusões



// Conclusões

**Mario Filho** • 1st
Lead Data Scientist | Machine Learning Expert | Kaggle Grandmaster
3d • 

Mesmo com AutoML vale a pena entender como funcionam os algoritmos?

Sim.




Estou estudando o **PyCaret** que é uma lib fantástica, super promissora, mas até dia 28/10 (4 dias atrás) ela fazia pré-processamento antes do split (que vaza info da validação).

Não tenha medo da automação, mas sempre verifique os detalhes ;)

E parabéns aos devs que corrigiram rapidamente!

#machinelearning #datascience #deeplearning #ai

See translation

   296 • 10 Comments



// Referências

- Automated Machine Learning: Methods, Systems and Challenges - Frank Hutter, Lars Kotthoff e Joaquin Vanschoren
 - Palestra “machine learning made easy(ish)” - Leonardo Bezerra na Python Brasil [14]: https://www.youtube.com/watch?v=nuRDxYF_35A
 - <https://medium.com/data-hackers/automl-uma-nova-abordagem-de-machine-learning-87a40d866dc1>
 - <https://medium.com/data-hackers/automated-machine-learning-automl-70c1eab669ad>
-



GAME OVER

OBRIGADO!