Deloitte
Auto ML project

Summer - 2018

Contents:

Problem statement

AutoML problem

Criteria of good model:

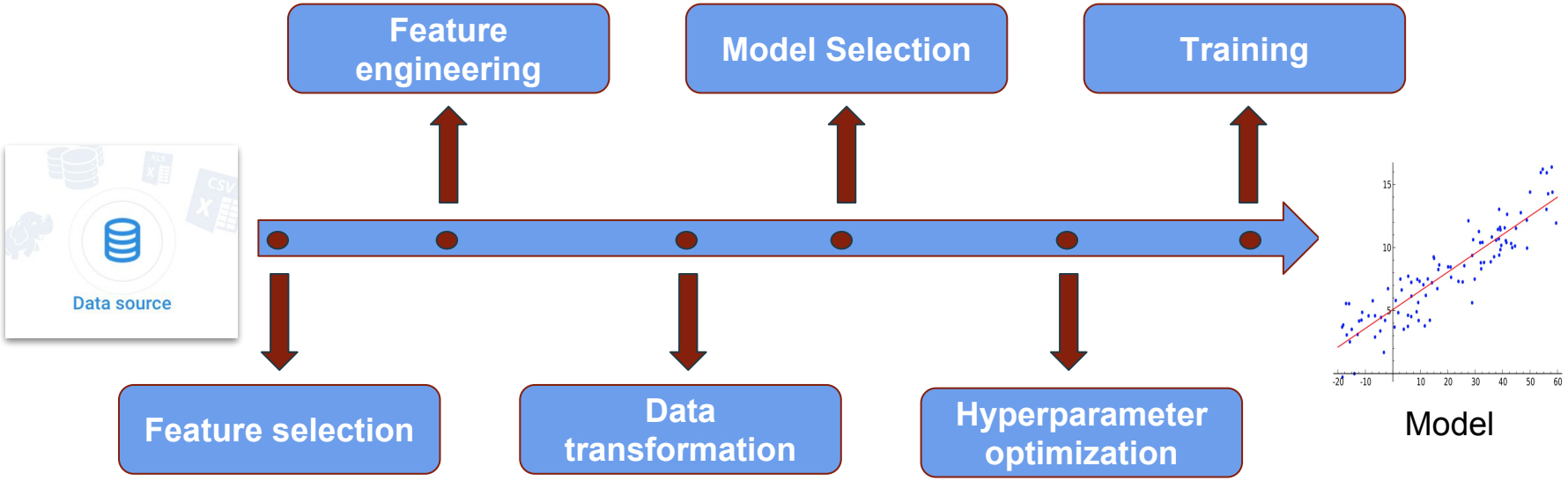
1) High quality
2) Interpretation

Optimization space

- feature selection
- feature engineering
- data transformation
- model selection
- hyperparameter optimization
- parameter optimization (fitting)

AutoML state-of-the-art approaches

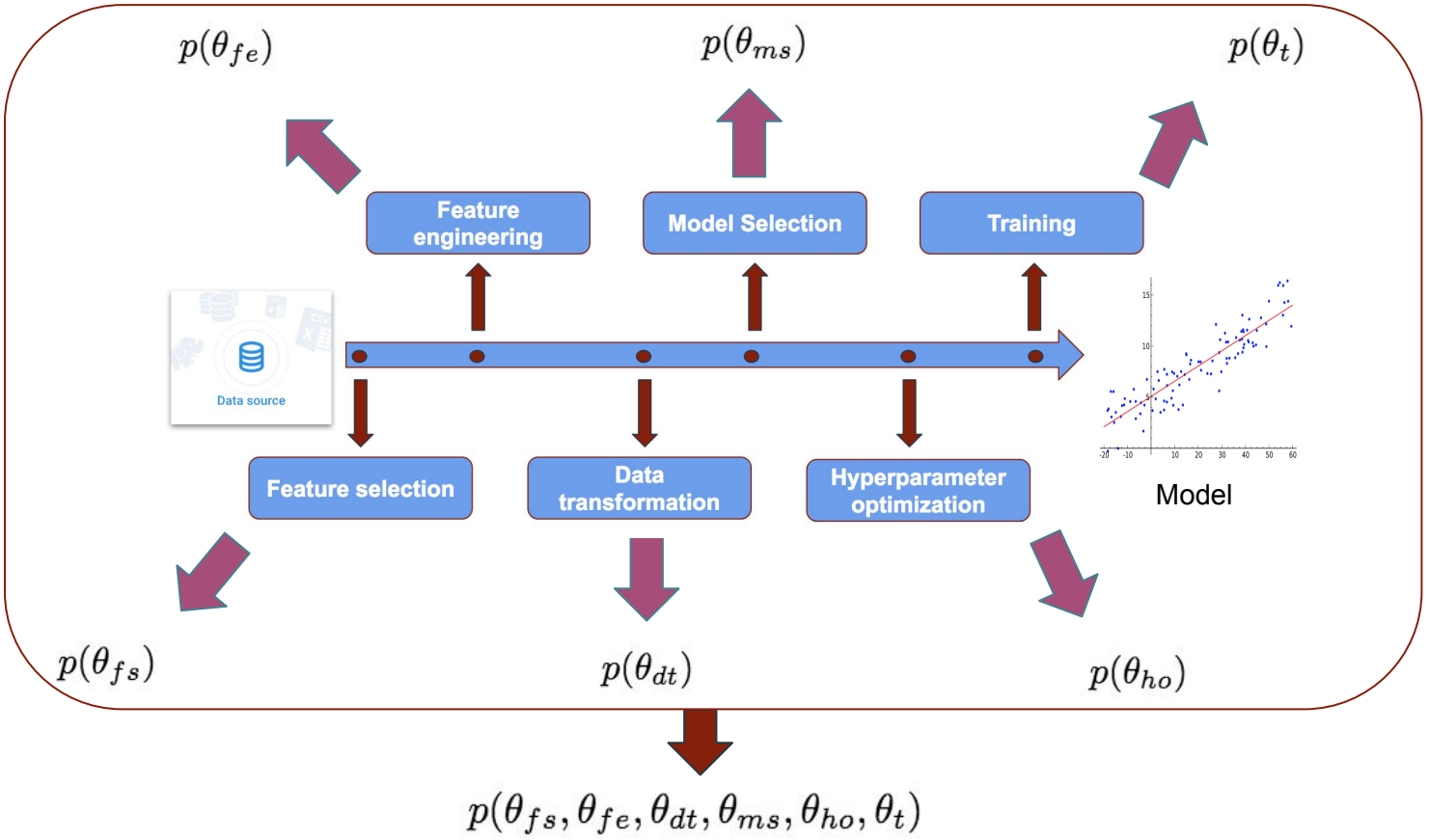
Sequential optimization



practitioner choose own pipeline

could be not best model

Global optimization



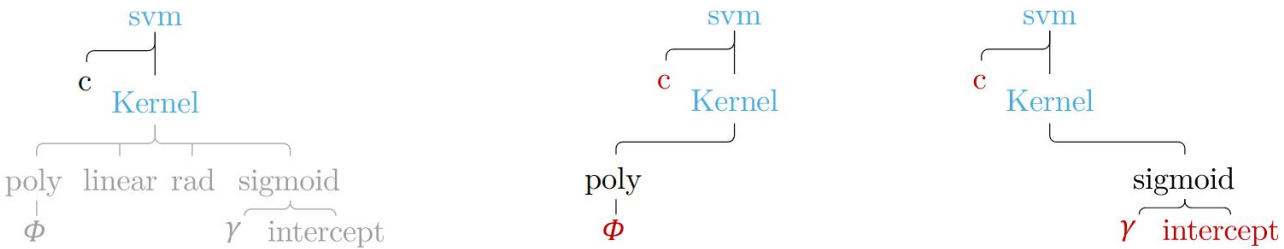
Mega - hyperparameter: combination of parameters of full pipeline

Example (discrete parameters):

$$p(\theta_{fs}, \theta_{fe}, \theta_{dt}, \theta_{ms}, \theta_{ho}, \theta_t)$$

f_1	f_2	f_3	...	f_m
0/1	0/1	0/1	...	0/1

The idea of conditional hyperparameters:



Review of Optimization methods

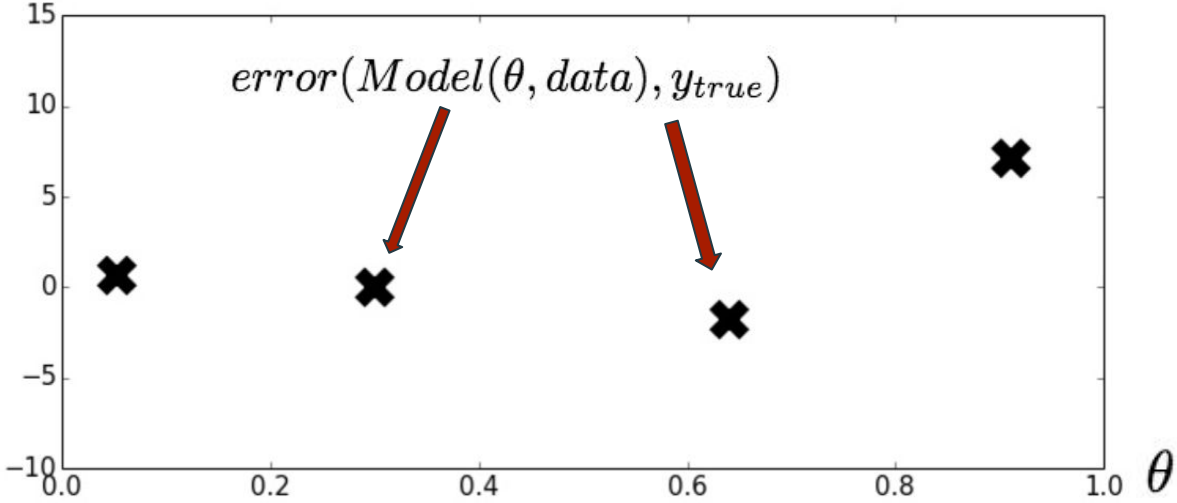
Review of optimization methods

Popular approaches of hyperparameters tuning:

1. Grid search
2. Random search
3. ***SMBO:***
    a. Greedy algorithms
    b. Bayesian optimization:
        i. Gaussian processes
        ii. Random Forest (or ensembles)
        iii. Hybrid (Parzen estimator) -TPE
    c. Hyperband (Bandit-Based Approach, RL)
4. Experience of experts

Approximation of loss function

$f(\theta)$ - loss function, e.g. error on validation set



θ - parameters of the model

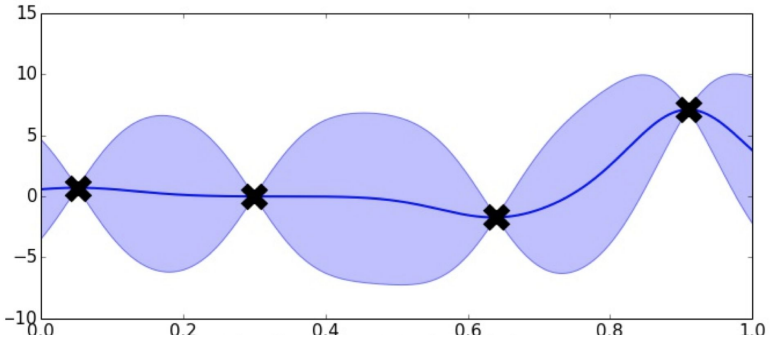
- Where is the minimum of f?
- Where should we evaluate the function next?

SMBO
1. Approximation of loss function

We have to construct model that approximate f and construct **confidence intervals**

>>

How to deal with this problem?

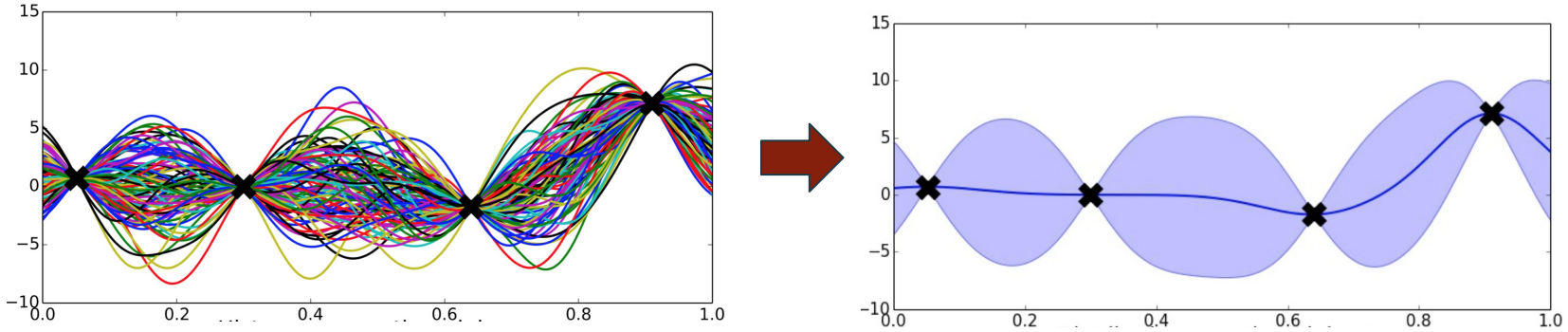


- Gaussian Processes
- Ensemble of models
- Bagging, bootstrap

→

- Choose base models (e.g. Random forest)
- Train and make predictions
- Calculate mean and std
- Construct distribution Normal(mean, std)

Example of construction confidence interval using ensemble approach:



SMBO
2. How to choose next point (next parameters) to evaluate function (loss function) ?

*We need to construct function **G**(parameters) that can somehow measure that we go to minimum*

1. Find $\theta^* = argmin f(\theta)$
2. Calculate $G(parameters|\theta^*)$
3. Find new point $\tilde{\theta}$ based on knowledge of $G(parameters|\theta^*)$

   Example: argmax G, median of distribution based on values of G

Example of the G function: Expected Improvement

Calculate positive improvement function $I_{f_{min}(\theta)} := max\{f_{min} - f(\theta), 0\}$

Problem! We can't calculate f directly

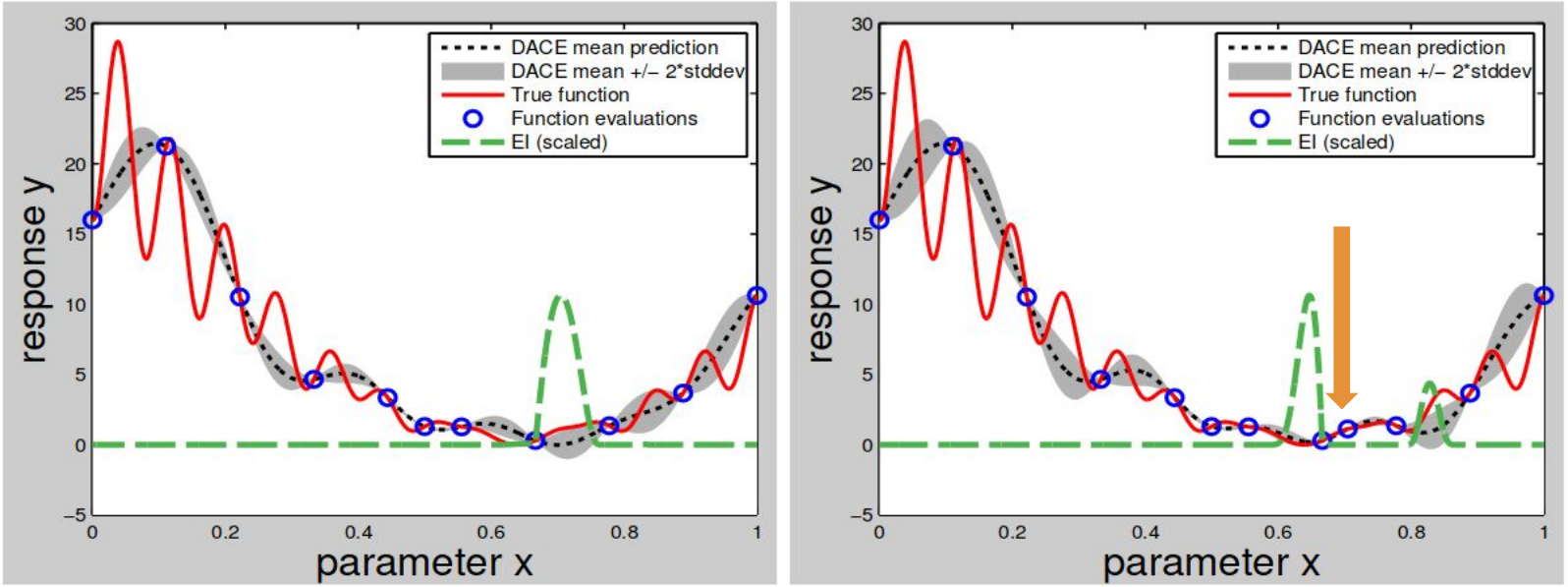
Solution

Calculate mathematical expectation of improvement function

$$\mathbb{E}[I_{f_{min}(\theta)}] := \int_{-\infty}^{f_{min}} max\{f_{min} - f(\theta), 0\} p(f|\theta) df$$

SMBO

3. Evaluate loss for new point sampled according **EI function**



(a) SMBO, step 1

(b) SMBO, step 2

Loss function (for which we want to find min)

SMAC (special name for SMBO applied to AutoML)

Main concepts:

- Random forest of regression trees to model the objective function
- Normal distribution with predictive mean and variance obtained by individual trees to construct predictive distribution

Advantages:

- perform well with discrete and high-dimensional input data
- handles conditional parameters (a graph-structured generative process)

Implemetations (Python):

- SMAC3

TPE

Main concepts:

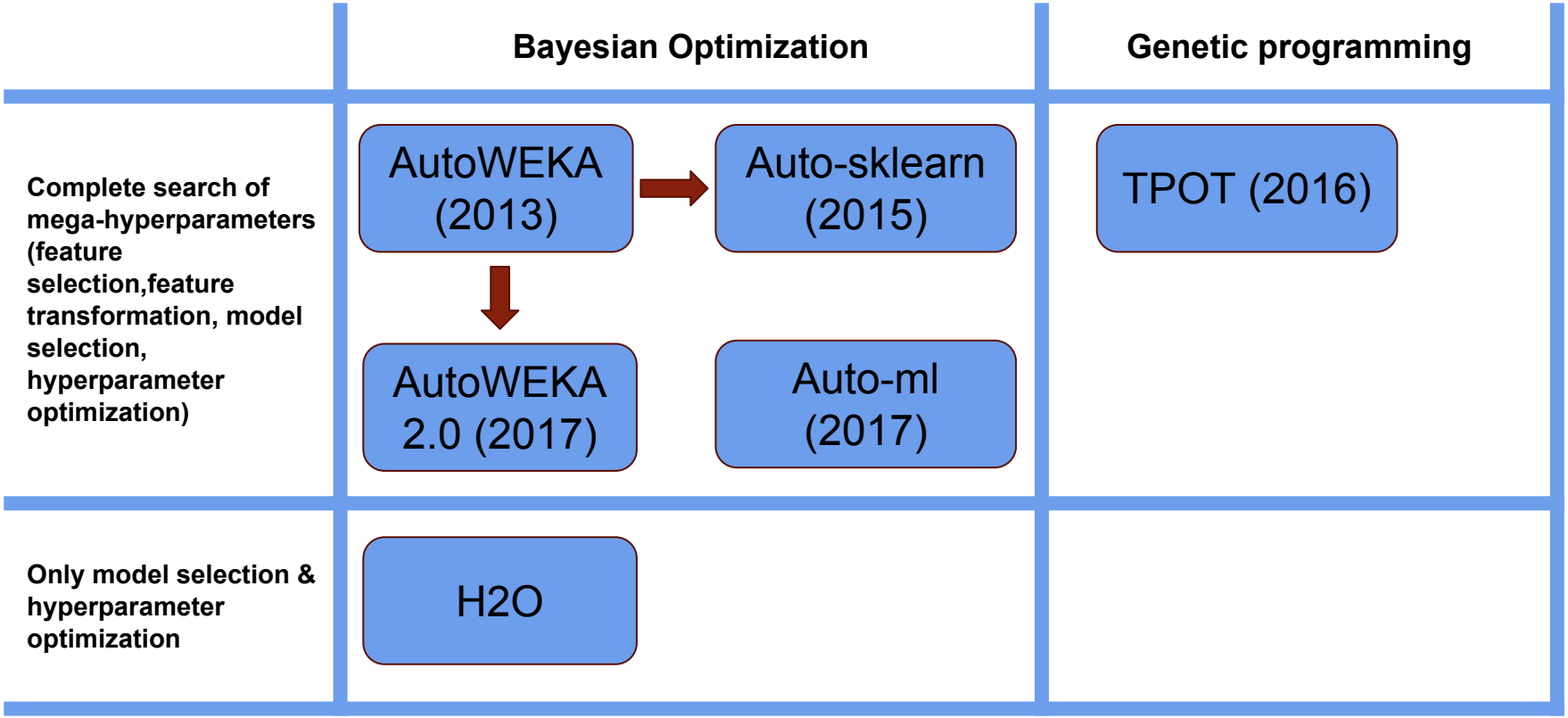
- construct prior ⇨ posterior distribution of parameters
- kind of hybrid algorithm with GP ***for continuous variables and discrete distributions for discrete variables***
- also uses regression trees
- as SMAC uses graph-structured generative process

Implemetations (Python):

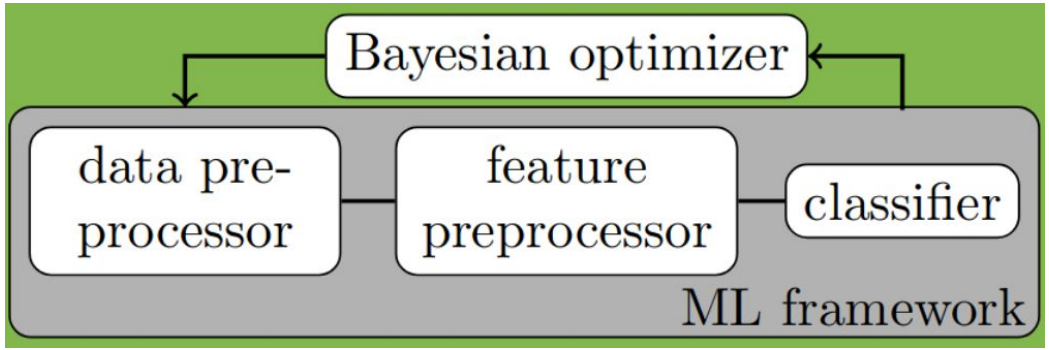
- Hyperopt
- Optunity

Auto - ML algorithms (python packages)

AutoML algorithms (python packages)



AutoWEKA (2.0)

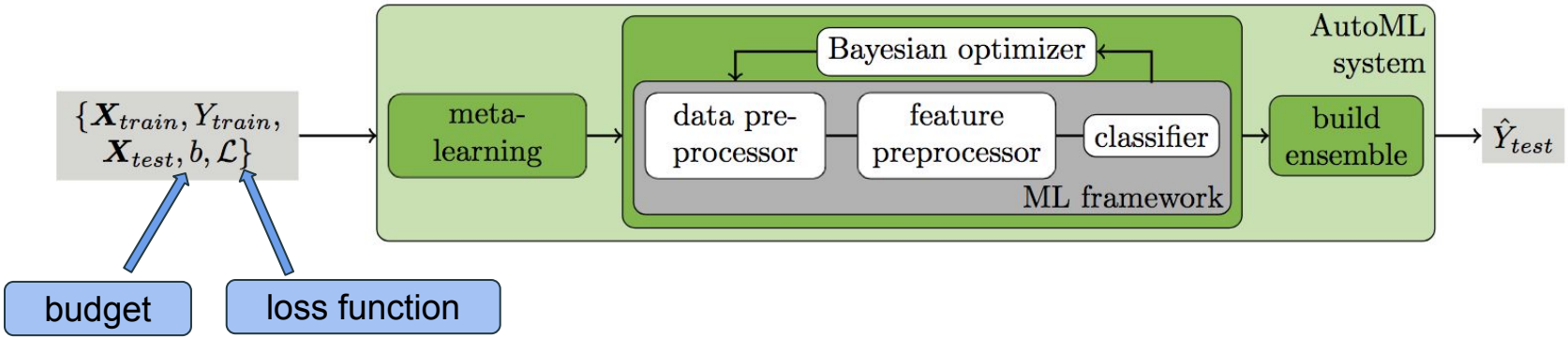


Main concepts:

- Classification/regression algorithms;
- Performs **only** feature selection;
- The idea of conditional hyperparameters;
- 786 hyperparameters;
- TPE / SMAC (authors claim **SMAC** is better)

Auto-sklearn

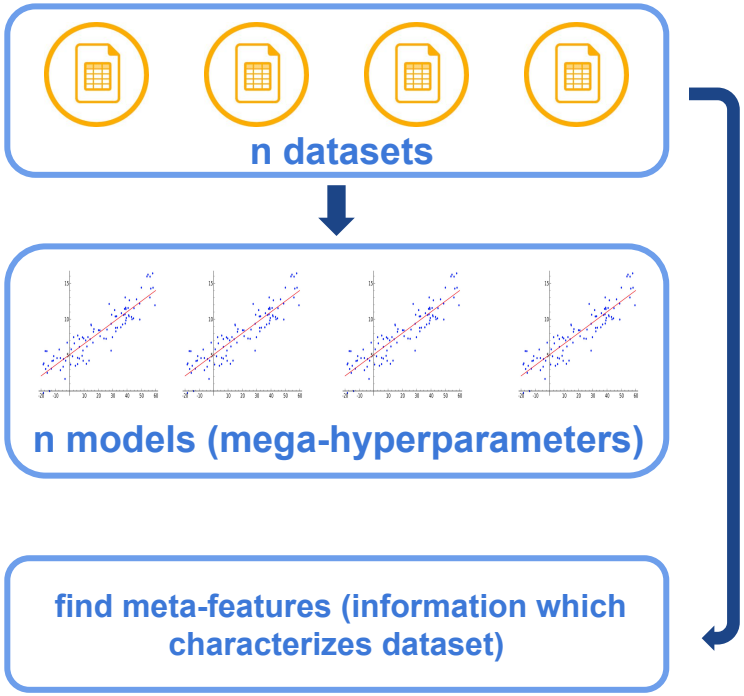
Auto-sklearn approach



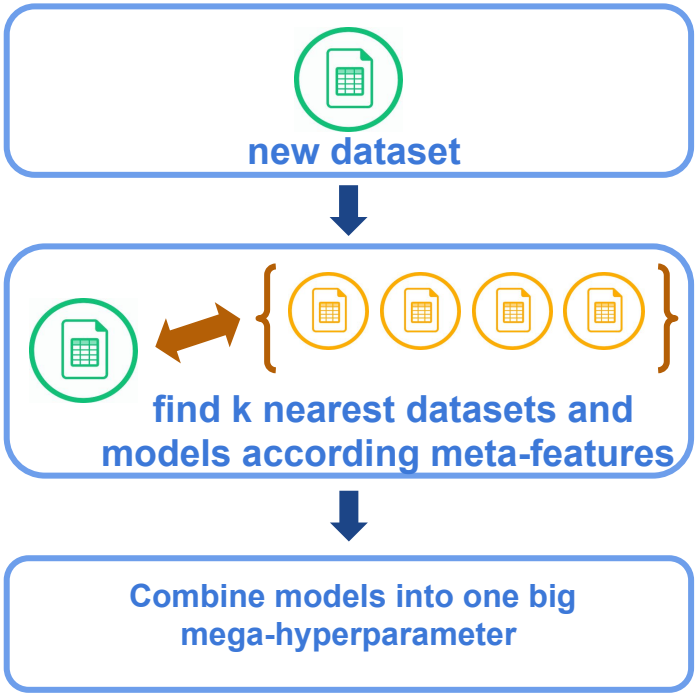
- extension of AutoWEKA;
- meta-learning is used as a warmstart for the Bayesian optimization procedure;
- tree based Bayesian models (SMAC);
- an automated ensemble construction step allows to use all classifiers that were found by Bayesian optimization;

Meta-learning - finding good instantiations of machine learning frameworks

1. Offline stage:

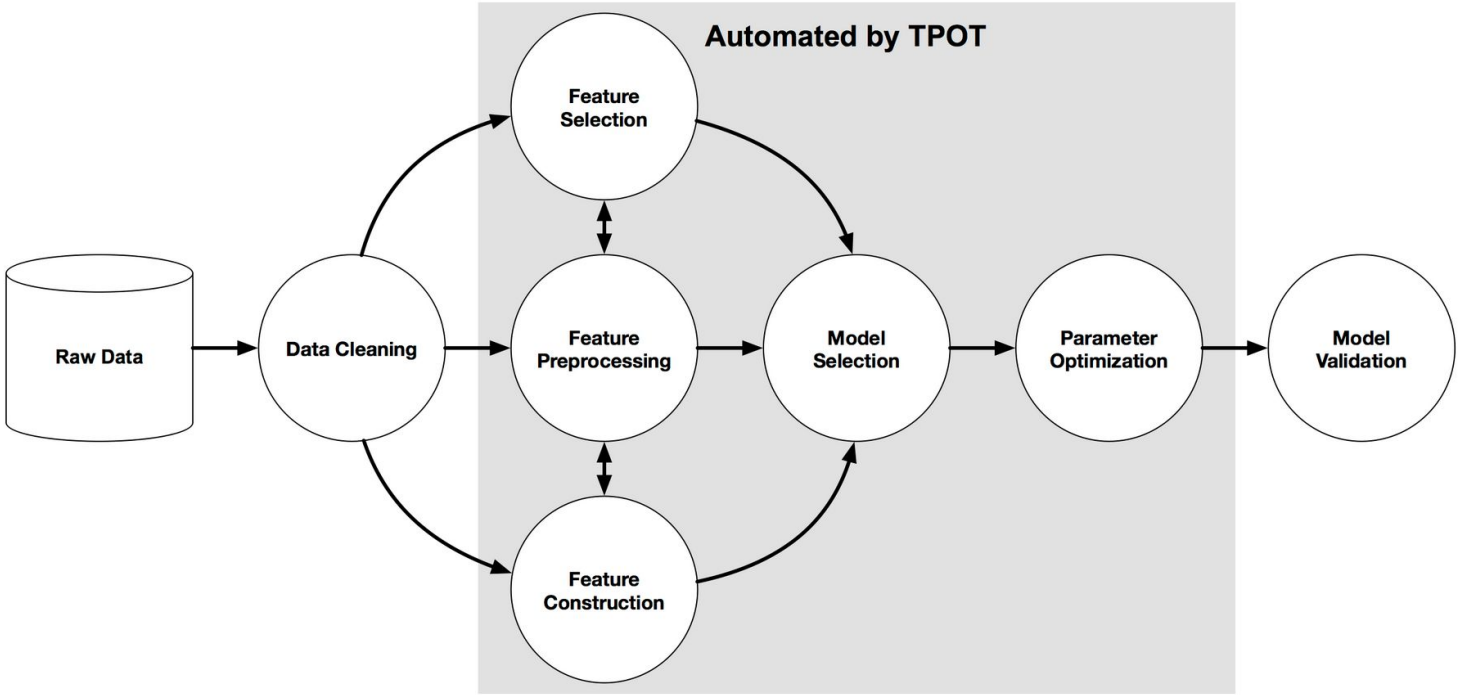


2. Online stage:



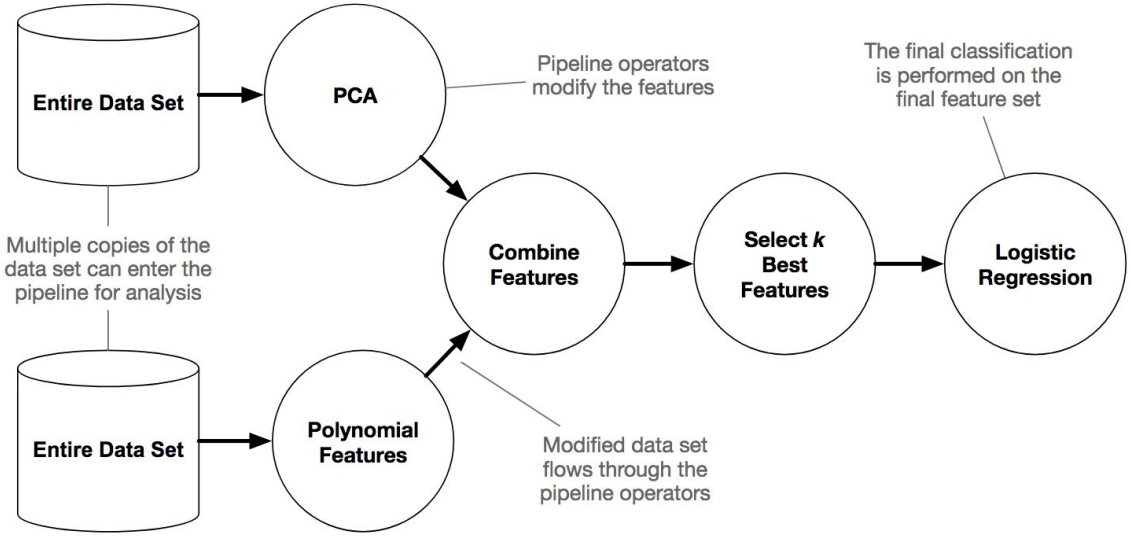
TPOT

Process


Automated by TPOT
Raw Data
Data Cleaning
Feature Selection
Feature Preprocessing
Feature Construction
Model Selection
Parameter Optimization
Model Validation

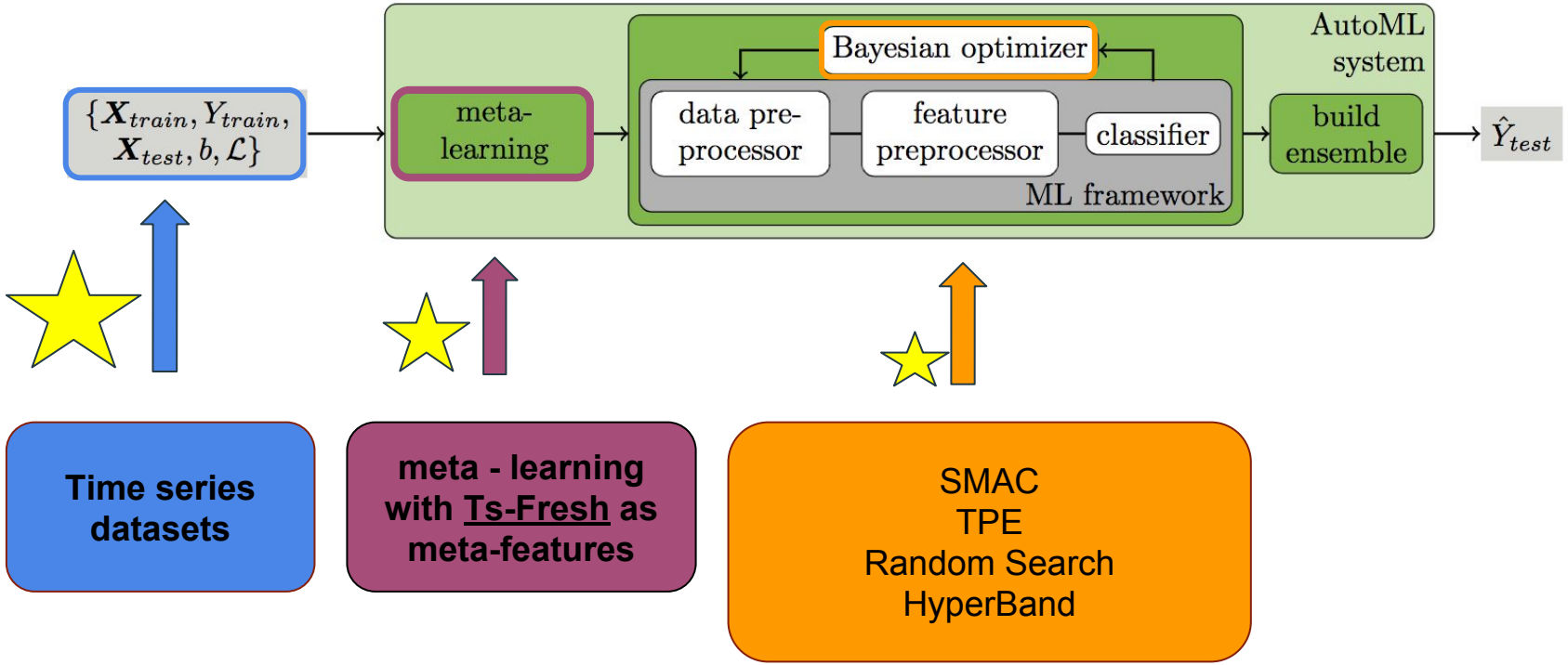
An example tree-based pipeline from TPOT

Genetic Programming for hyperparameter optimization



Time Series Auto ML - proposed approach

Adaptation of auto-sklearn approach for time-series analysis



References:

Future work

Future work

Comparative analysis of Auto-sklearn, TPOT, Auto_ML **and human created expert models** on Time Series data sets

TS-Fresh + Auto - sklearn

Ensemble of TS-Fresh + Auto - sklearn and Sarimax

TS-Fresh + Deep Learning approach

Thank you for your attention!