INFORMATIONSFABRIK

DATEN VERSTEHEN, ENTSCHEIDUNGEN TREFFEN

Towards automating [supervised] machine learning: Benchmarking tools for hyperparameter tuning

Dr. Thorben Jensen

tjensen@informationsfabrik.de

INFORMATIONSFABRIK

DIGITAL TREASURE HUNTERS

MÜNSTER

Continental

ZURICH

Deutsche Messe

DONNER & REUSCHEL
PRIVATBANK SEIT 1798

SIGNAL IDUNA
Versicherungen und Finanzen

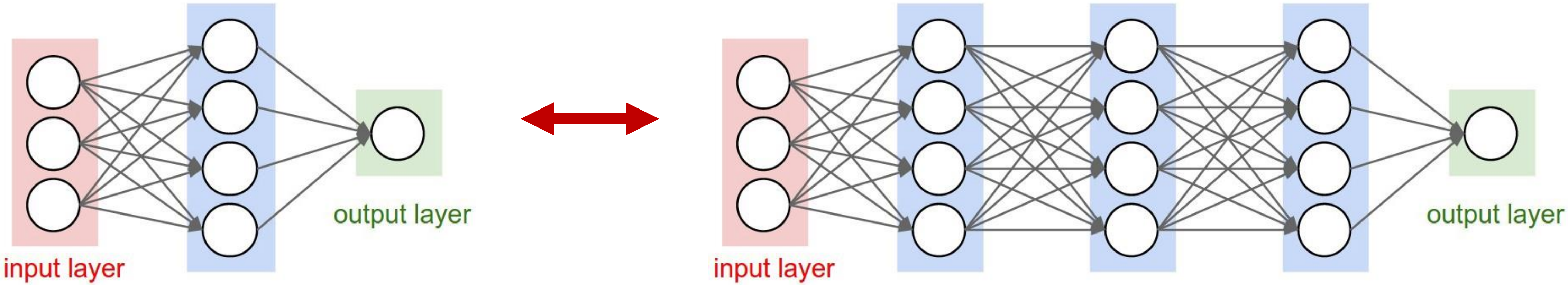
TUI

COMMERZBANK

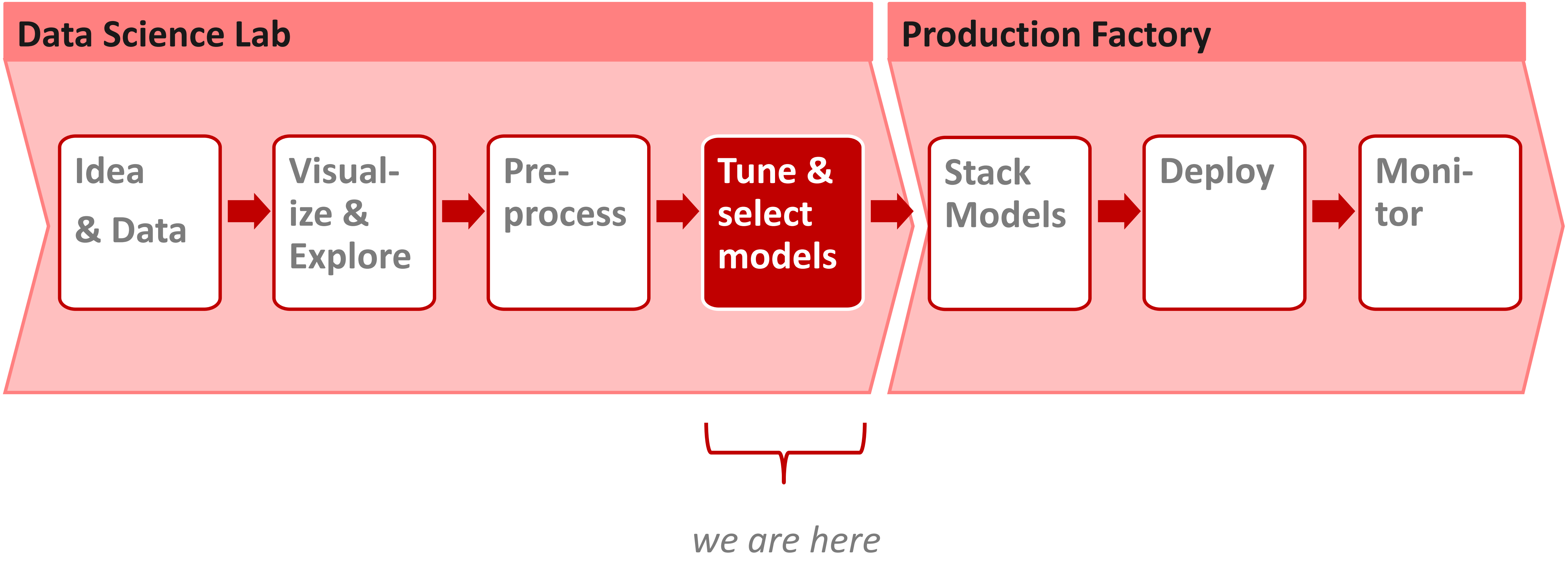
Finding optimal hyperparameters is important!

- Hyperparameters
  are *"parameters whose values [are] set before the learning process begins.*
  By contrast, the values of other parameters are derived via training." (wikipedia.org)

- Hyperparameter example: *depth of neural network*



Hyperparameter tuning in supervised machine learning



A *week* in the life of a data scientist

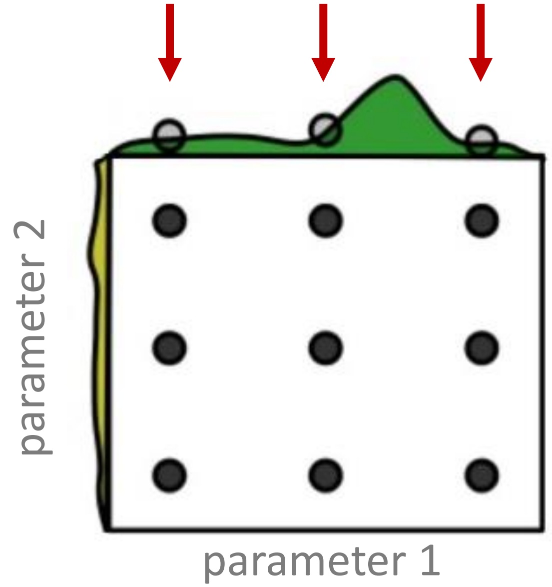
Tuning … **Re-tuning …** **Re-tuning …**

https://phdcomics.com/

Simple automation: Grid and Random Search

Grid search

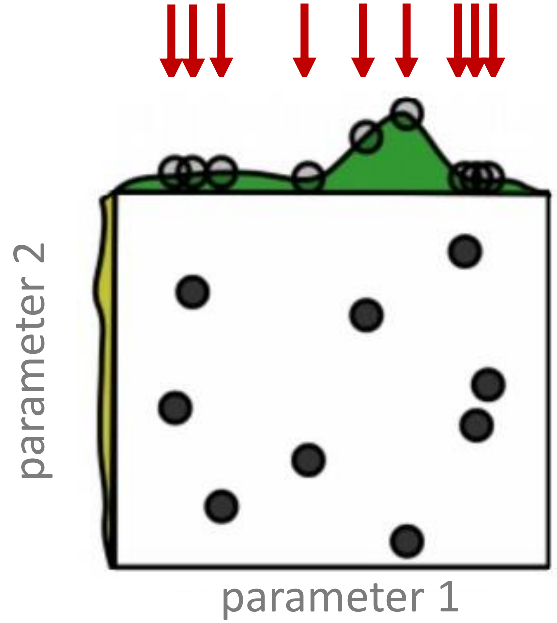
1. Select values for each hyperparameter to test
2. Try ALL combinations



Random search

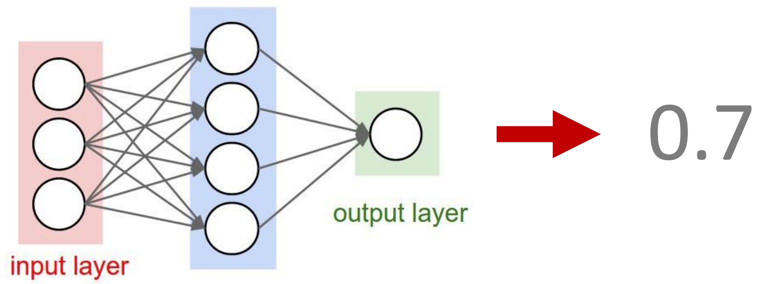
- Varies important hyperparameters more !

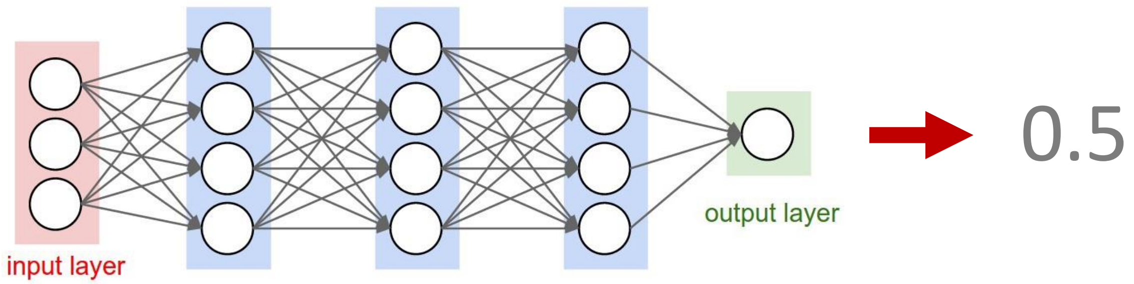
- More efficient at model tuning



http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf

Sequential model-based optimization (SMBO)



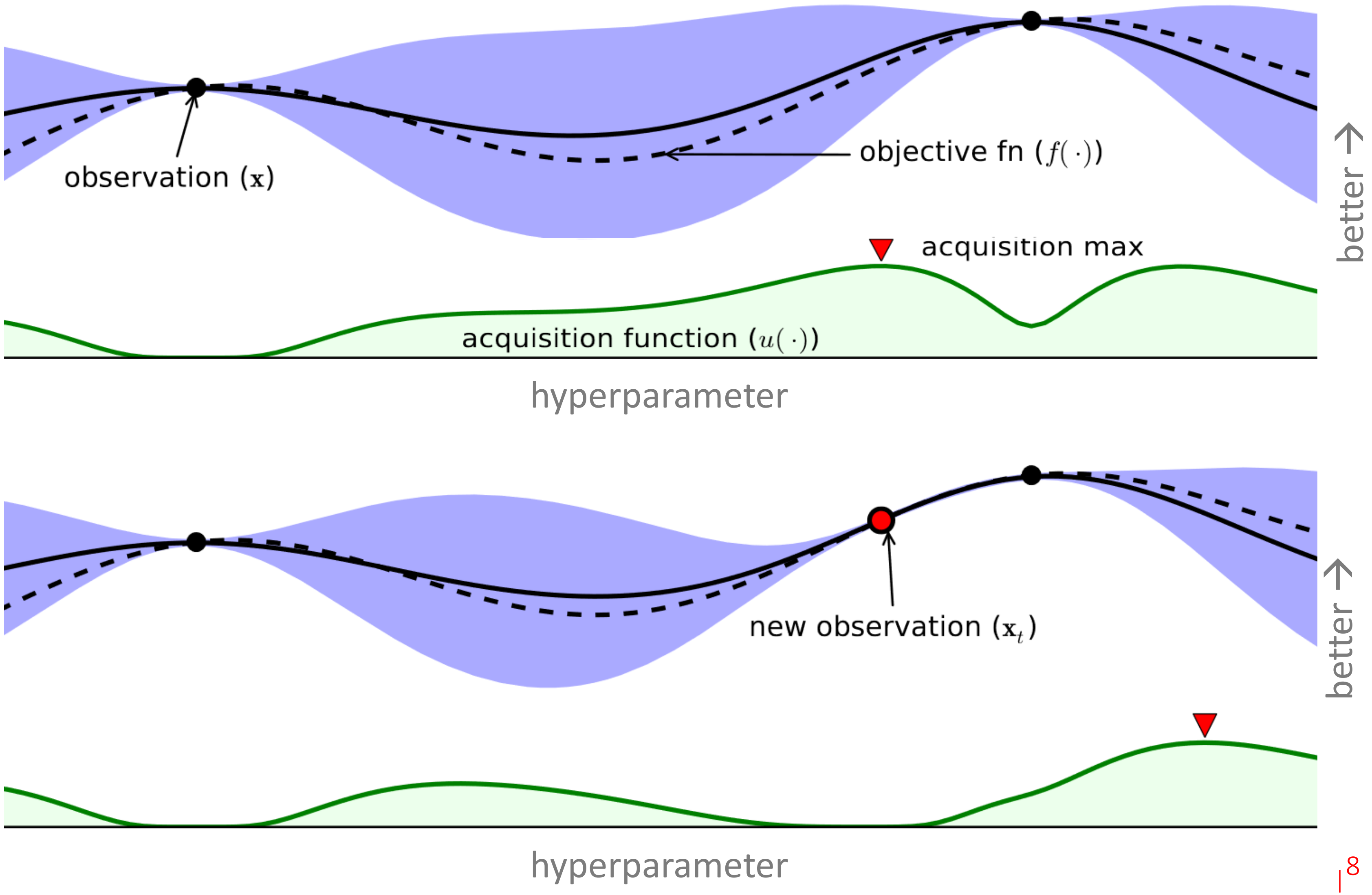


- Use fast *regression model* as proxy for slow ML model:
  1. Evaluate some random sets of hyperparameters
  2. Build a *regression model*: 'hyperparameters -> loss'
  3. Find hyperparameter with lowest loss, according to fast proxy
  4. Evaluate real model for hyperparameter, observe loss
  5. Update *regression model*

- Popular options for proxy model:
  - Gaussian Processes
  - Random Forests
  - Tree-structured Parzen Estimators

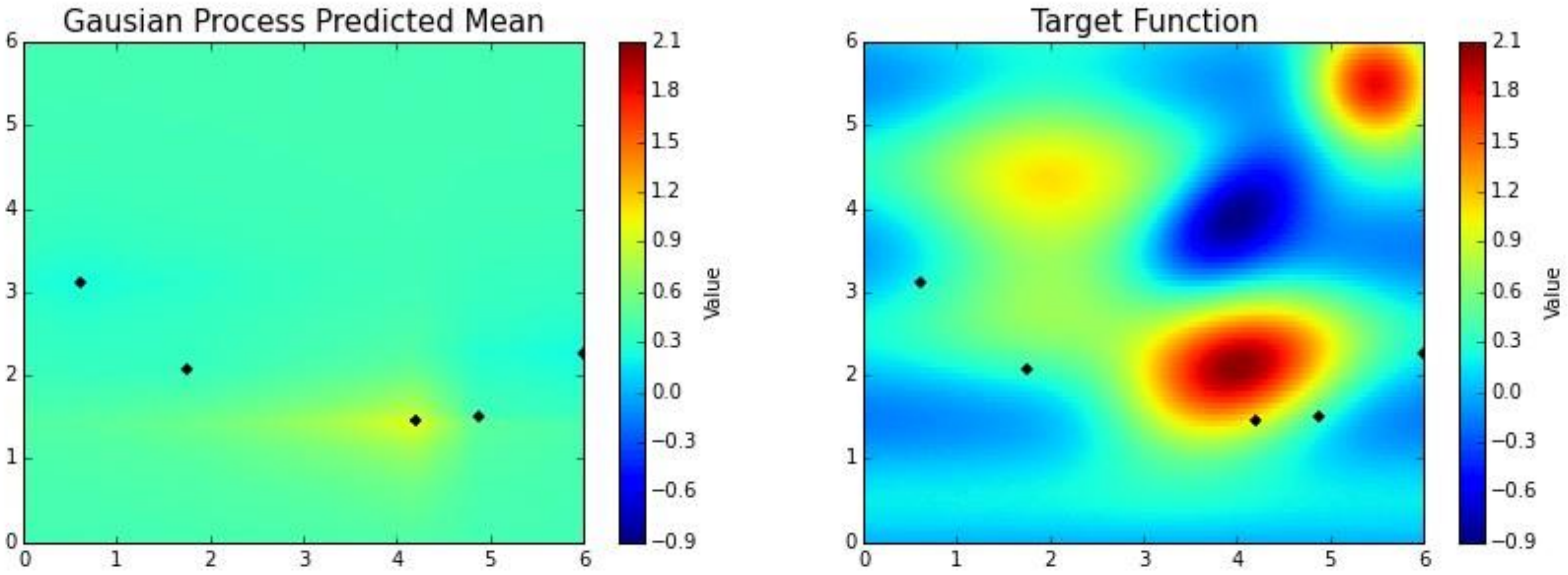
Gaussian Process

1. From previous samples, fit *Gaussian Process*:
   expected value
   uncertainty range
2. Max(**acquisition function**): good **expected value** and **high uncertainty**
3. Use **new observation** to update *Gaussian Process*
4. Repeat from step 2 …



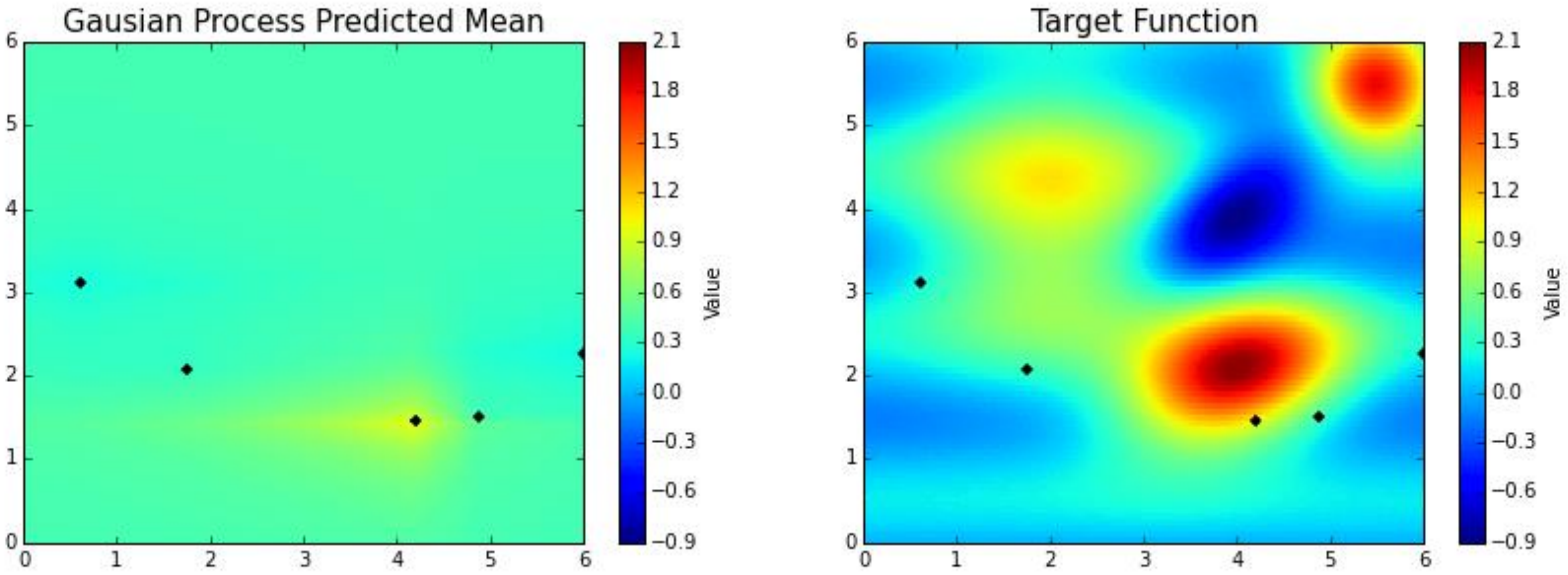
https://arxiv.org/pdf/1012.2599.pdf

Gaussian Process live



https://github.com/fmfn/BayesianOptimization

Gaussian Process live



https://github.com/fmfn/BayesianOptimization

Random Forest

- Differences to Gaussian Process
  - Uses Decision Tree Ensemble, e.g. **10 trees**
  - Also works with **categorically scaled hyperparameters**, e.g. activation function in neural net

http://www.cs.ubc.ca/labs/beta/Projects/SMAC/papers/11-LION5-SMAC.pdf

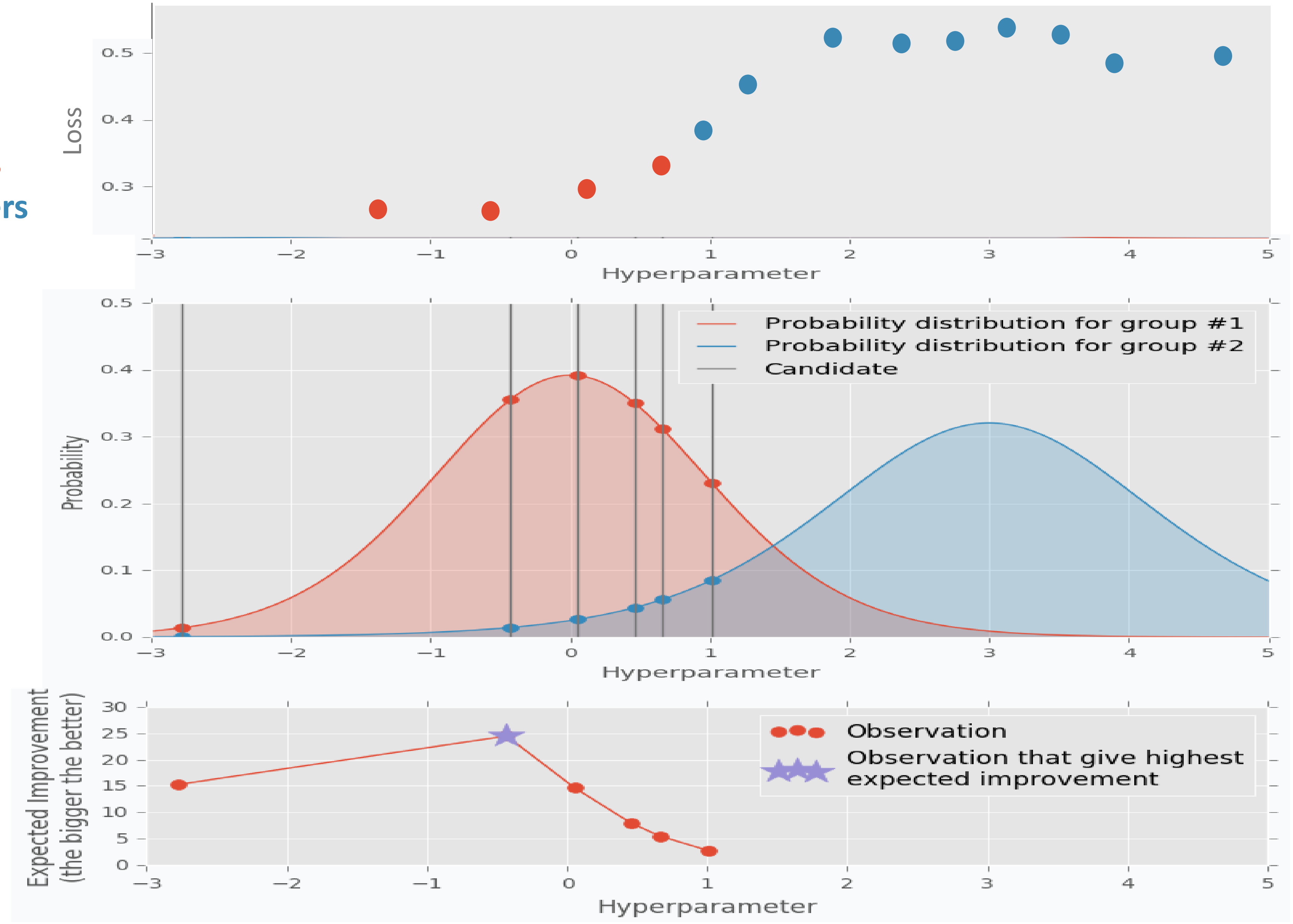
Tree-structured Parzen Estimator

1. Test some hyperparameters

2. Separate into:
   best hyperparameters
   worse hyperparameters

3. For new candidates (vertical lienes), model probability to be in **good** or **bad** group

4. *Expected improvement* for candidate:
   P(good) / **P(bad)**



http://neupy.com/2016/12/17/hyperparameter_optimization_for_neural_networks.html

Algorithms compared

	interacting hyperparameters	also discrete hyperparmeters	easy configuration	Python library
▪ Gaussian Process	✓	✗	✗	✓
▪ Random forest	✓	✓	✓/✗	✓
▪ Tree-structured Parzen Estimator	✗	✓	✓	✓

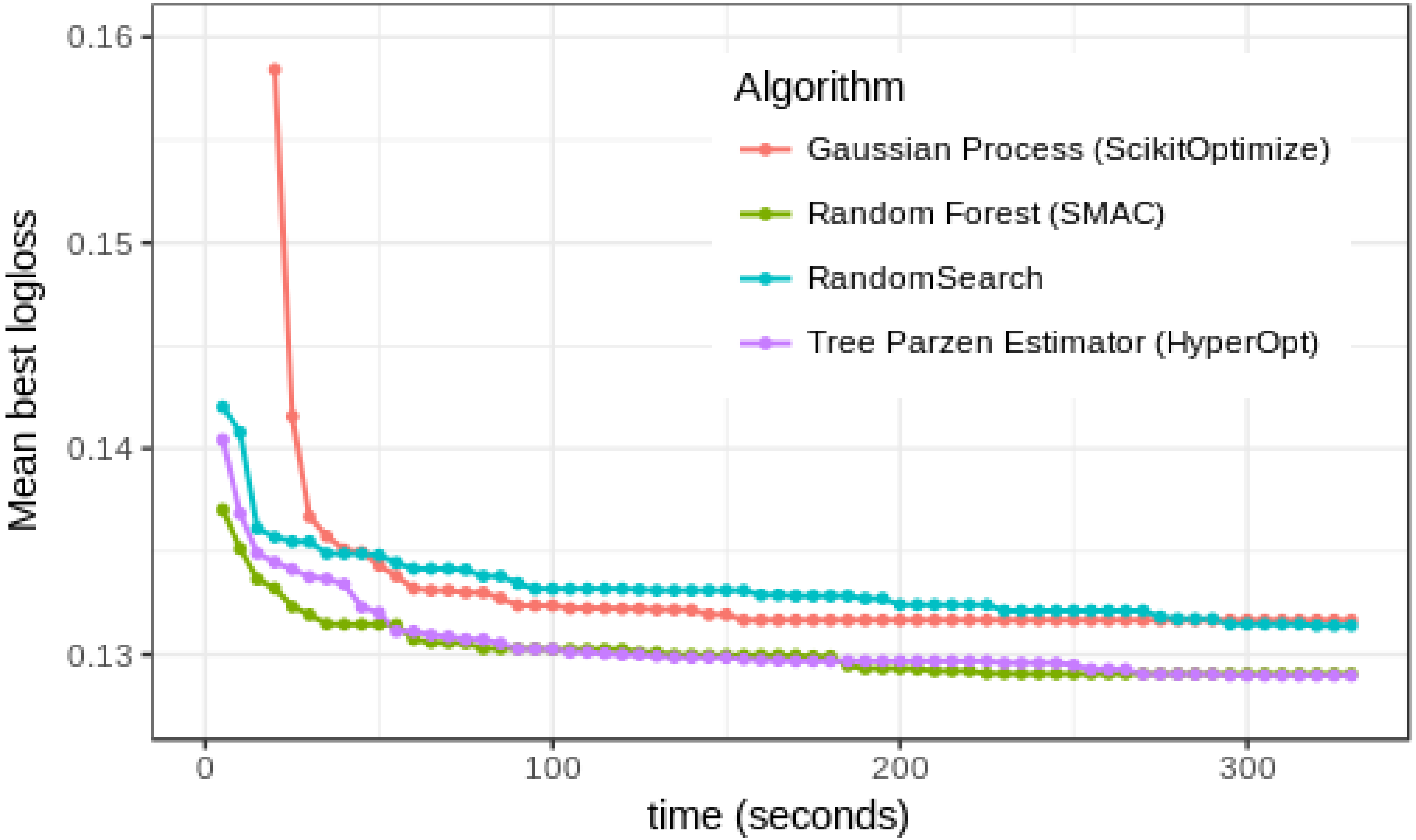
Which one to choose? → *benchmark 'em all!*

- Benchmarked python libraries
  - **Random Search:** own implementation
  - **Gaussian Process**: *bayes_opt, skopt*
  - **Random forest:** smac
  - **Tree-structured Parzen Estimator:** hyperopt
- Comparing to human expert (3 manual steps)

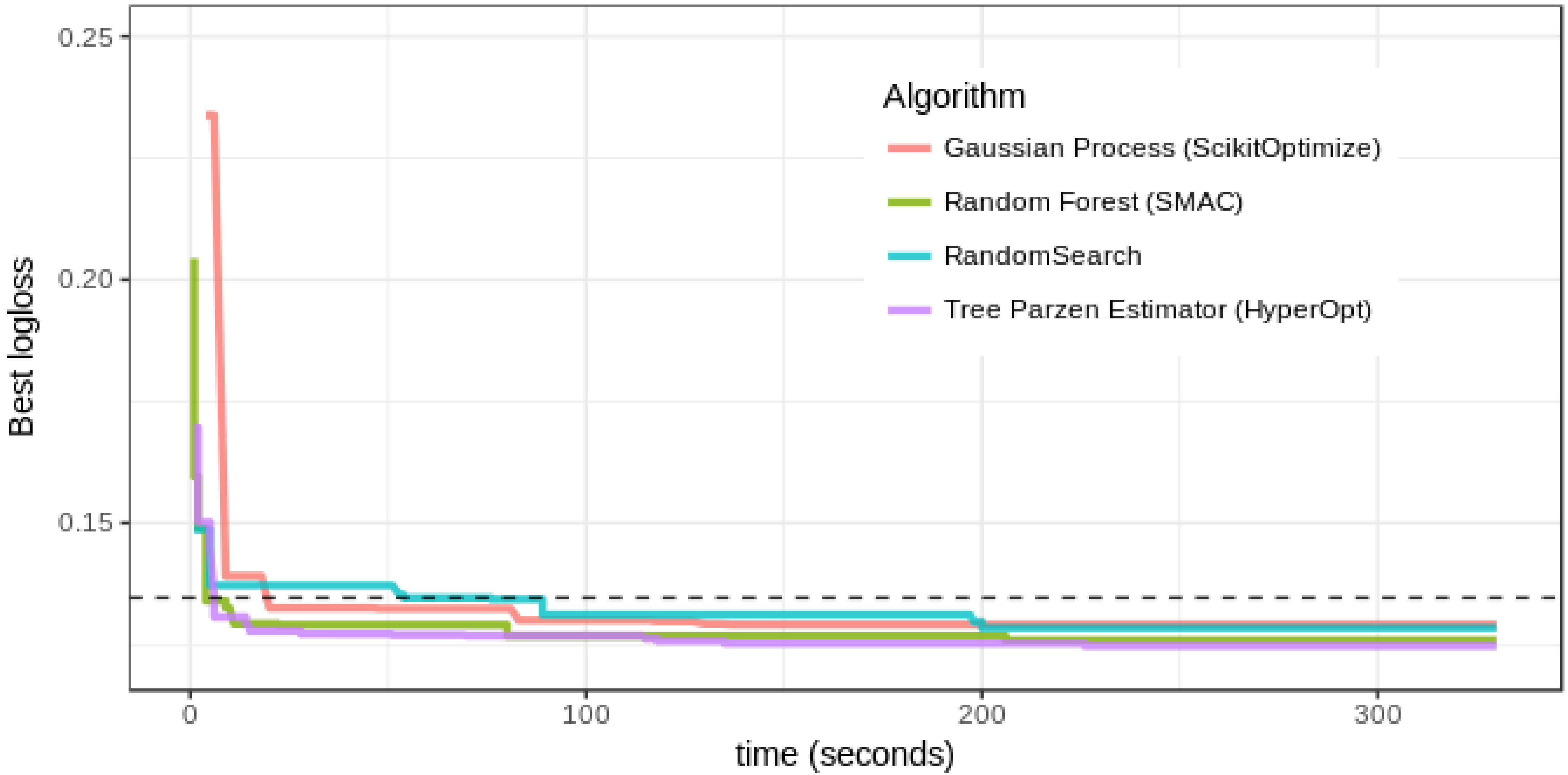
- Task: classification with xgboost library, early stopping
- 16 core CPU machines
- Structured data sets
  - Iris
  - Real-life dataset for decision support in insurance

- Limitations
  - Only 2 datasets
  - Varying both models and implementations

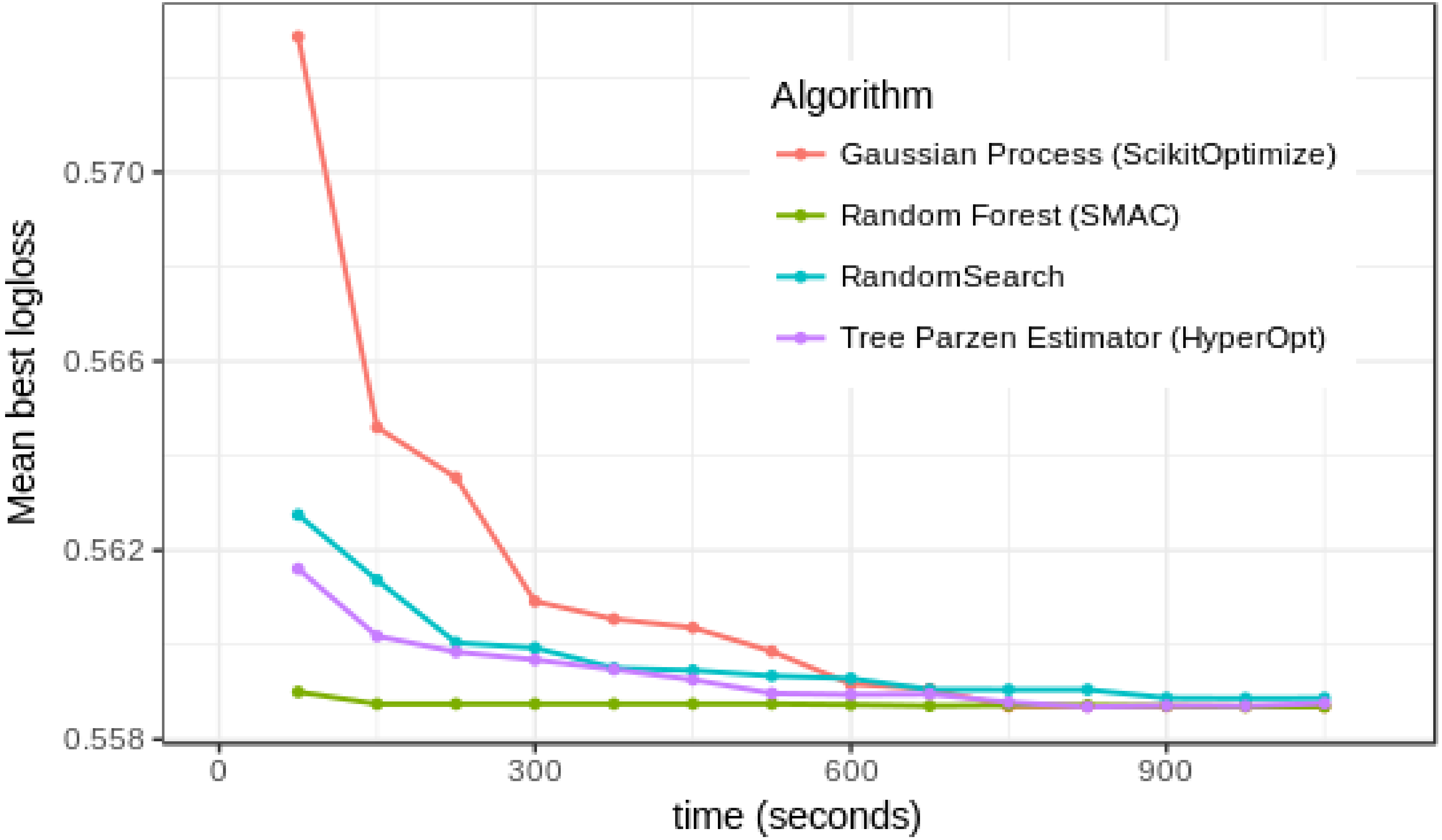
Result I/II: iris dataset



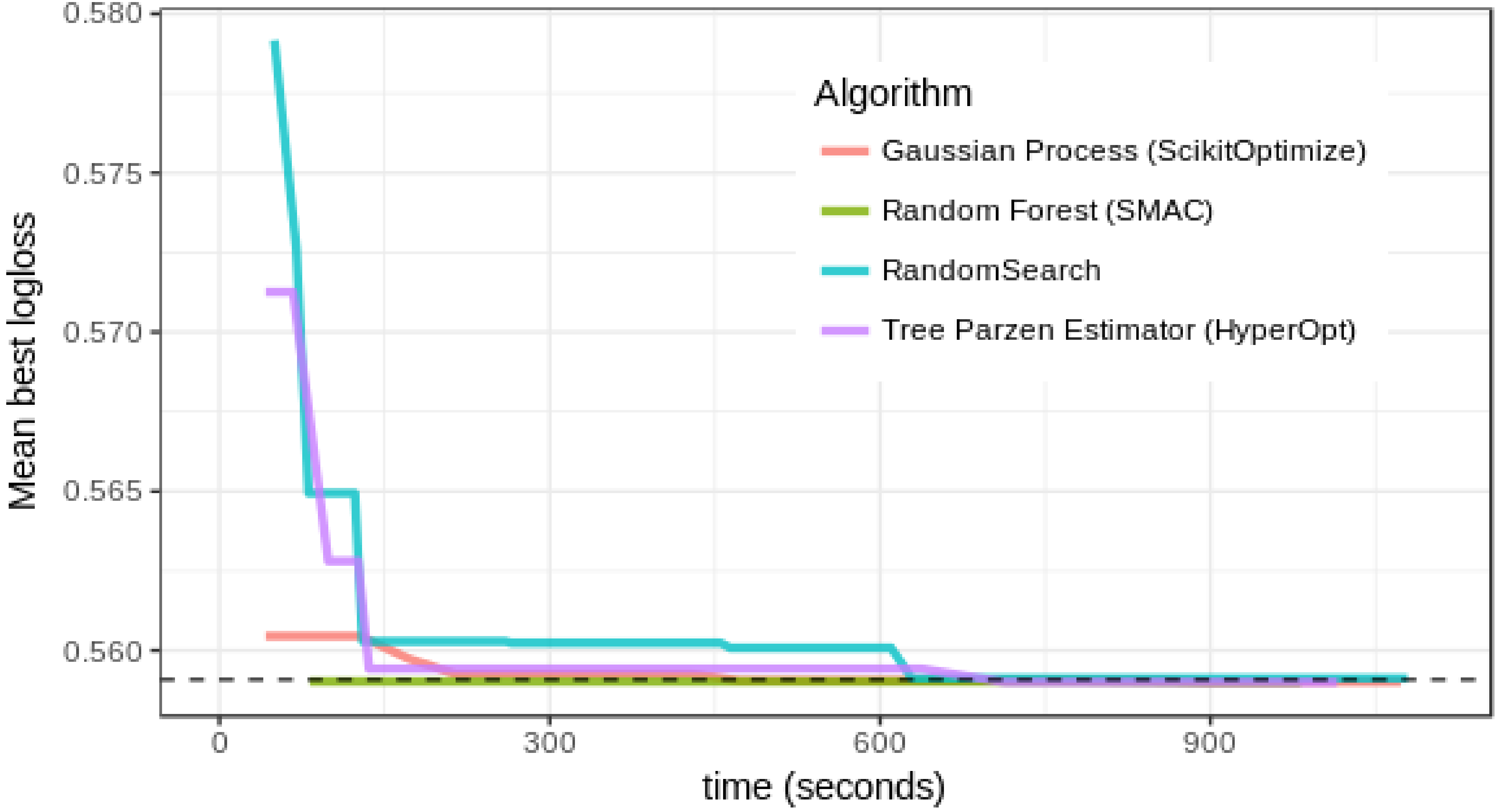
Result I/II: iris dataset – compared to manual tuning



Result II/II: real-life dataset

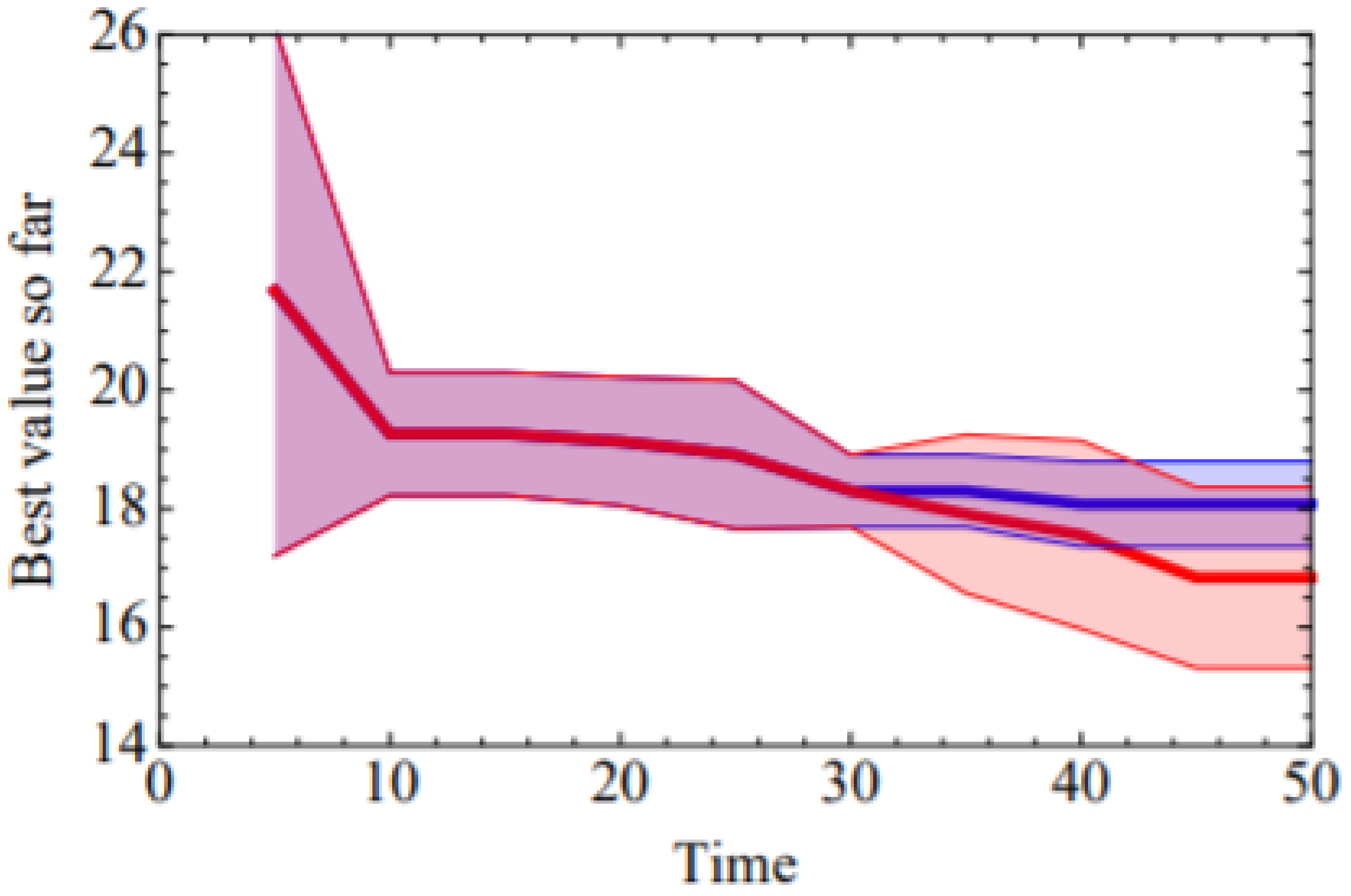


Result II/II: real-life dataset – compared to manual



Literature I/II

Gaussian Process beats Random Search

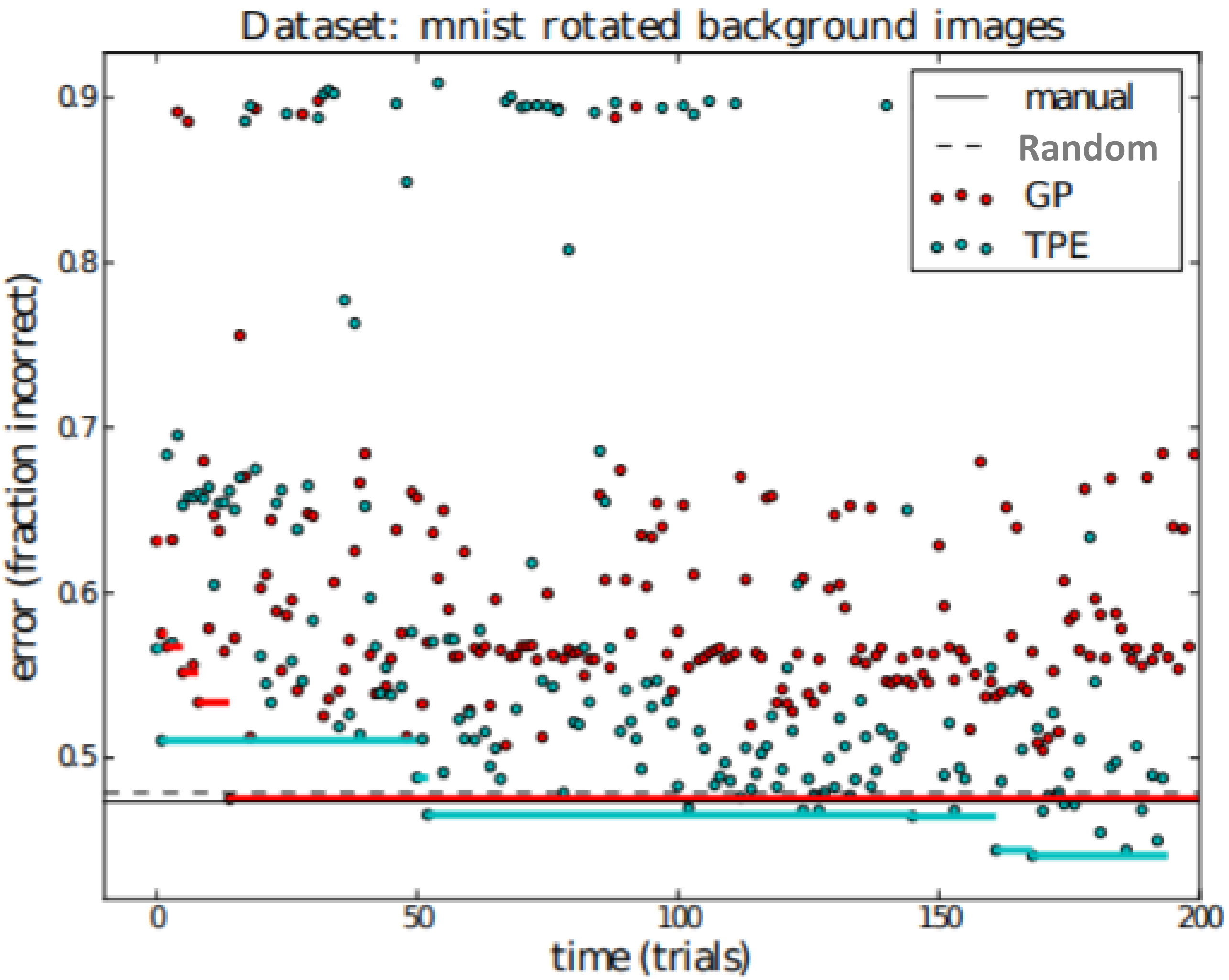


https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

Literature II/II

Tree-structured Parzen Estimator

beats Gaussian Process, Random Search (- - -), and Manual Search (—)

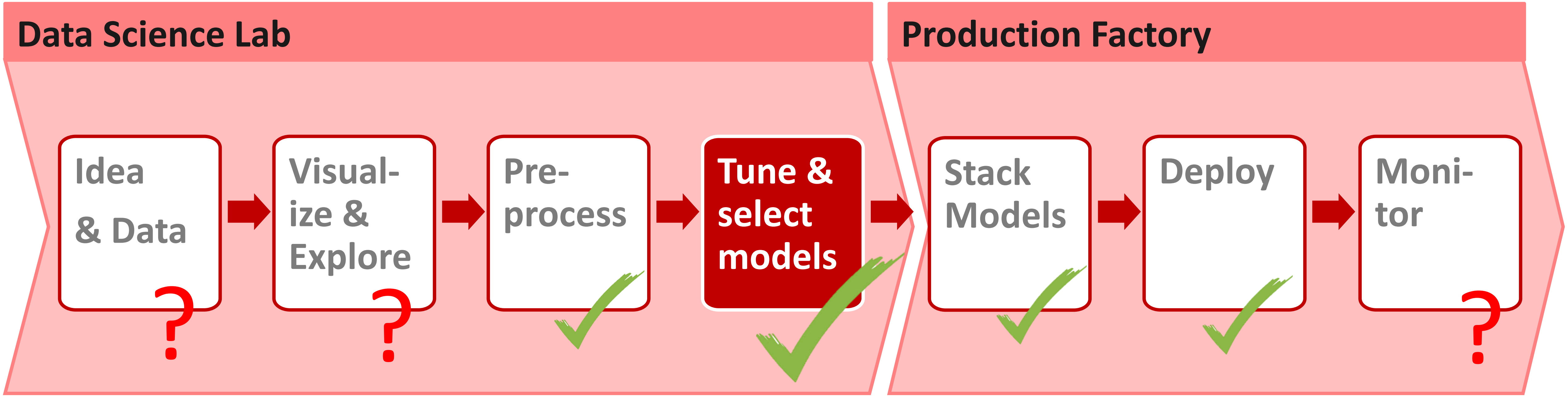


https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

Take home messages

- More free weekends: automated tuning beats manual
- SMBO algorithms do not differ much in performance, but in constraints:
  - Scaling of your hyperparameter
  - Interactions between hyperparameter
  - Algorithms for hyperparameter optimization have hyperparameters…
- Tree-structured Parzen Estimator simple, but surprisingly effective
- Random Search simple, but scalable and reasonably competitive
- Spend your time wisely! → Instead of manual tuning, get more data and features

Towards automating [supervised] machine learning?



TAK

DANK U WEL

谢谢

GRACIAS

KÖSZÖNÖM

CHOKRANE

СПАСИБО

TERIMA KASIH

THANK YOU

VIELEN DANK

GRAZIE

TESEKKÜR EDERIM

DZIĘKUJĘ

MERCI

ขอบคุณครับ

TÄNAN

ARIGATÔ

HVALA

Special thanks to Tobias Rippel, Prasanth Sivabalasingam!