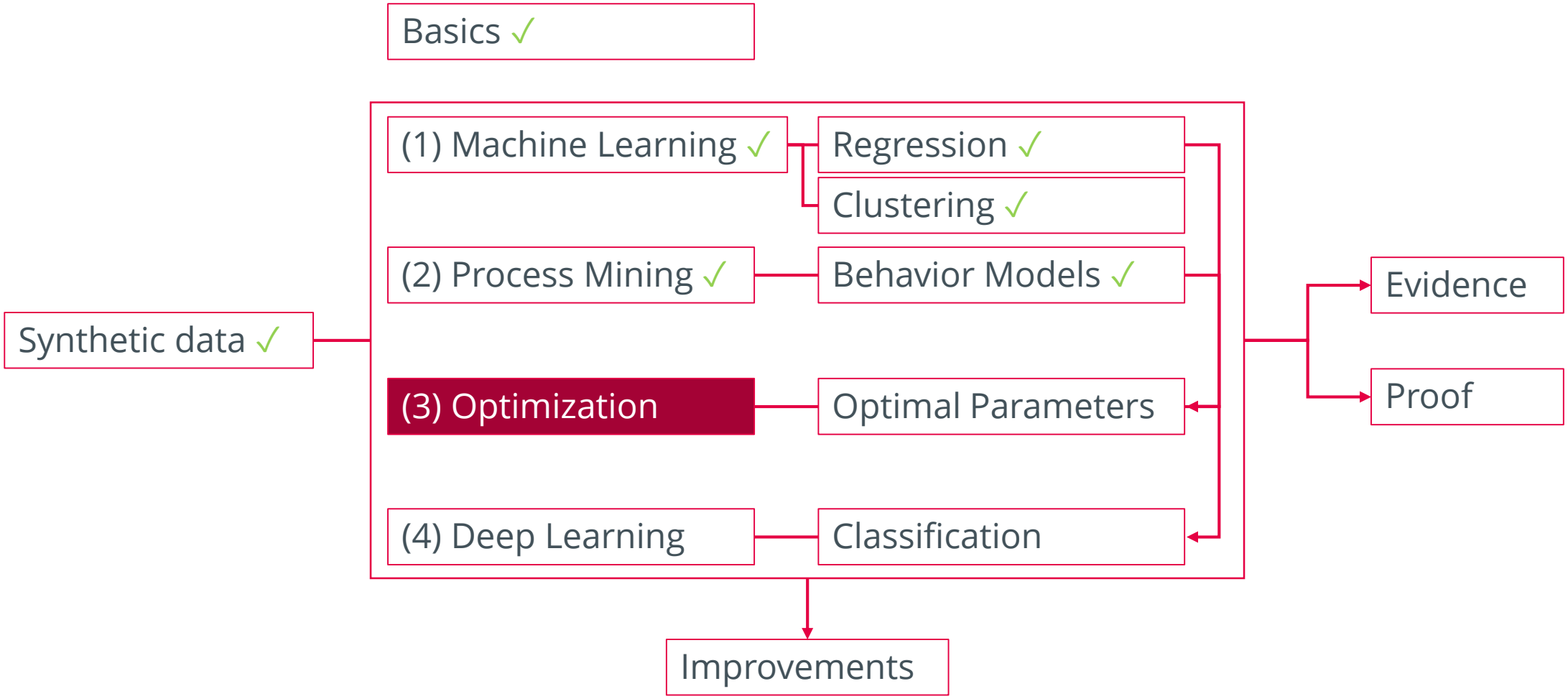
TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Welcome

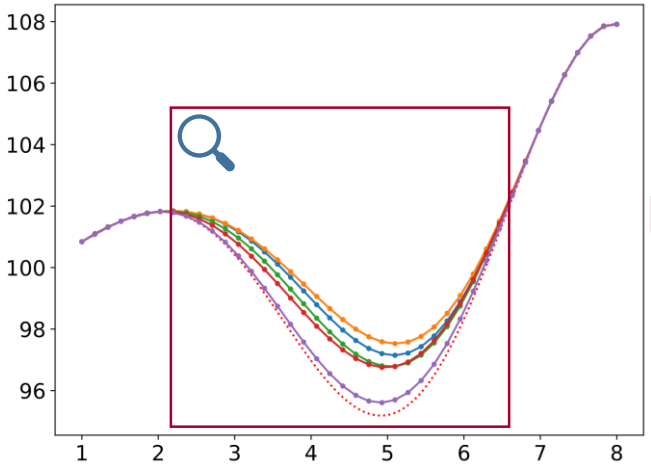
to Advanced Topics in Algorithms

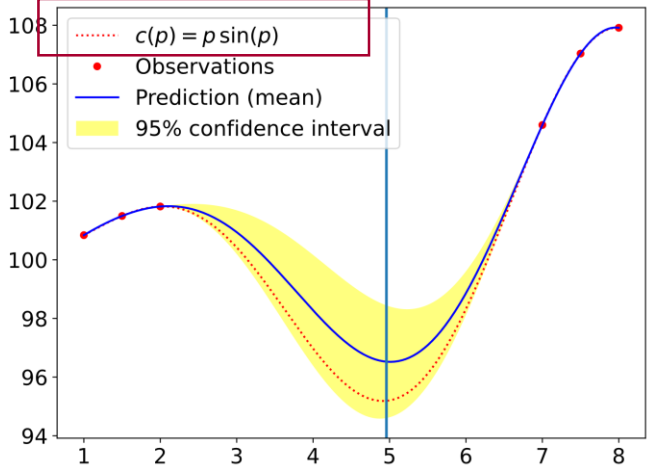
Summary: Advanced Topics in Algorithms

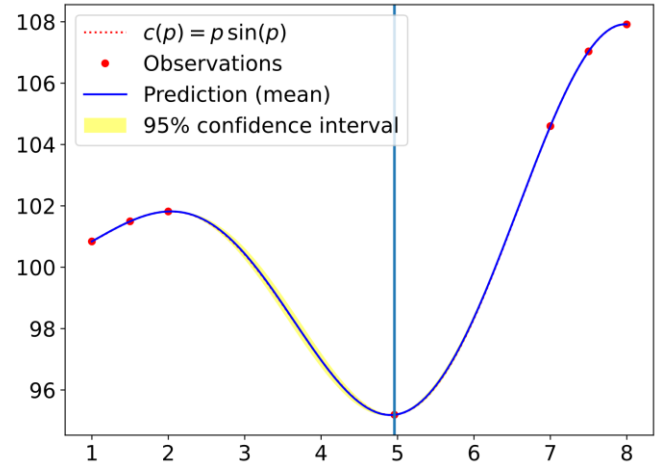


Revision: Bayesian Optimization

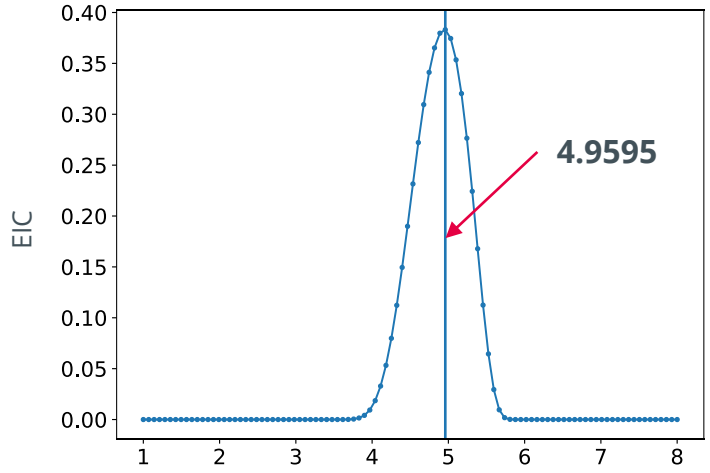
Sample possible values







- Use Gaussian Process Regression
- Calculate values using the Expected Improvement Criterion (EIC)
- Choose parameter configuration with the highest EIC



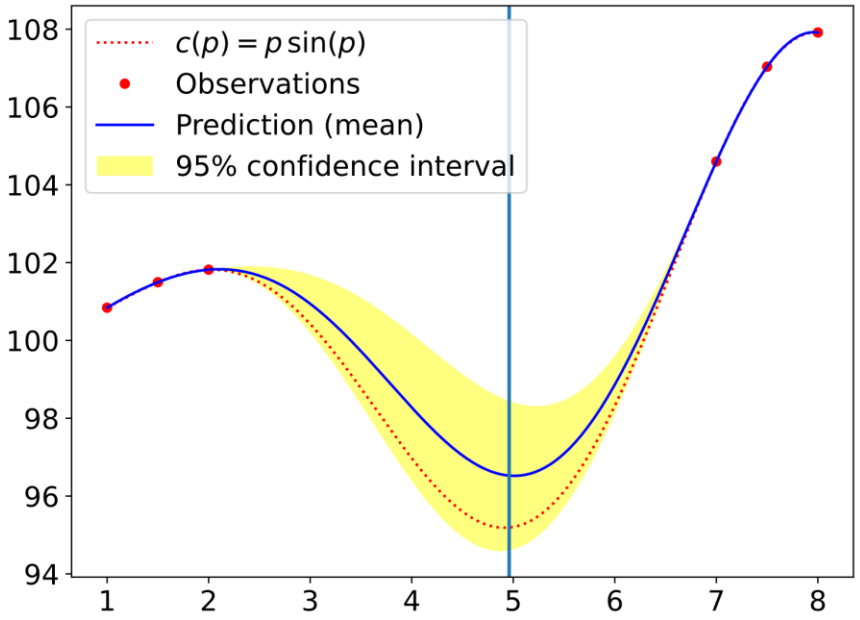
Add observation:
p = **4.9595**
c(p) = 95.1911

Function value

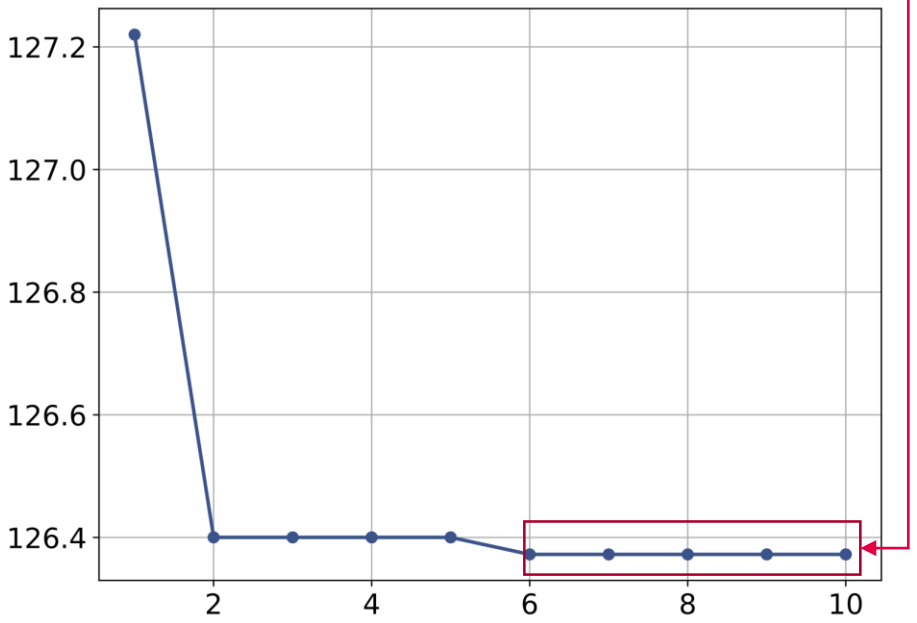
Function

Exercise 1: Bayesian Optimization (scikit-optimize)

Find Optimal Parameters for Regression with Polynomial Features (scikit-optimize)
Implement “Exercises 06: Bayesian Optimization: Exercise 1: Find Optimal Parameters for Regression with Polynomial Features” with scikit-optimize (https://scikit-optimize.github.io/stable/index.html). Use “skopt.plots: Plotting functions” (https://scikit-optimize.github.io/stable/modules/classes.html) to visualize results of the algorithm. Compare the “scikit-optimize solution” with your implementation.

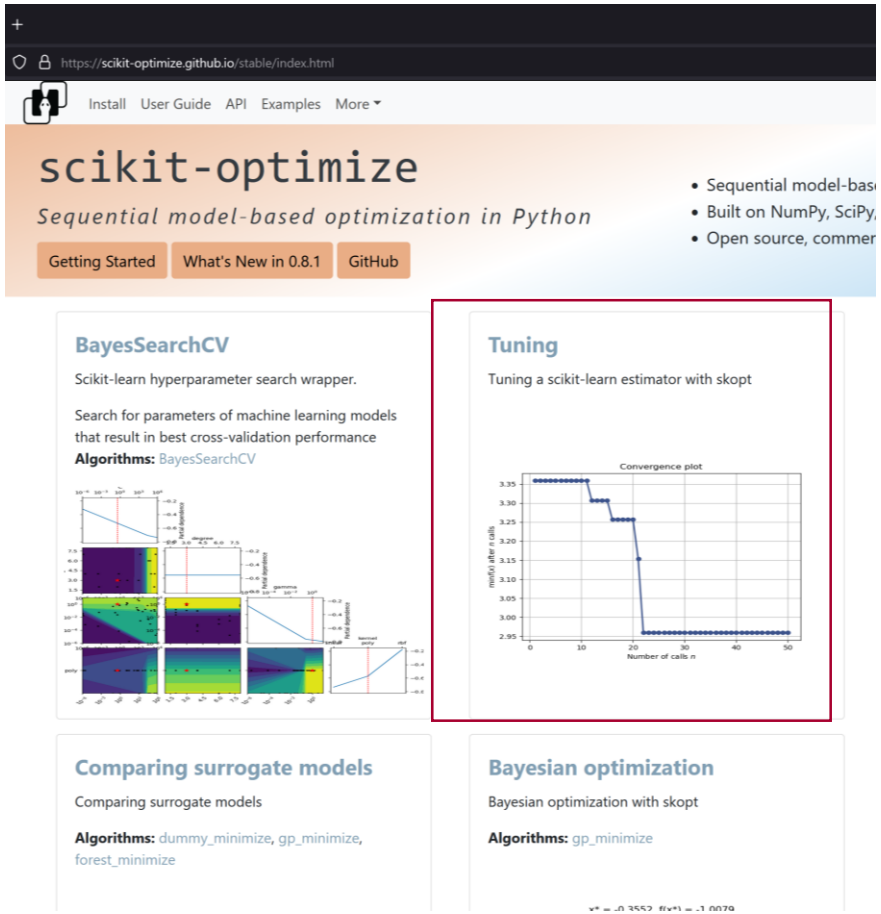


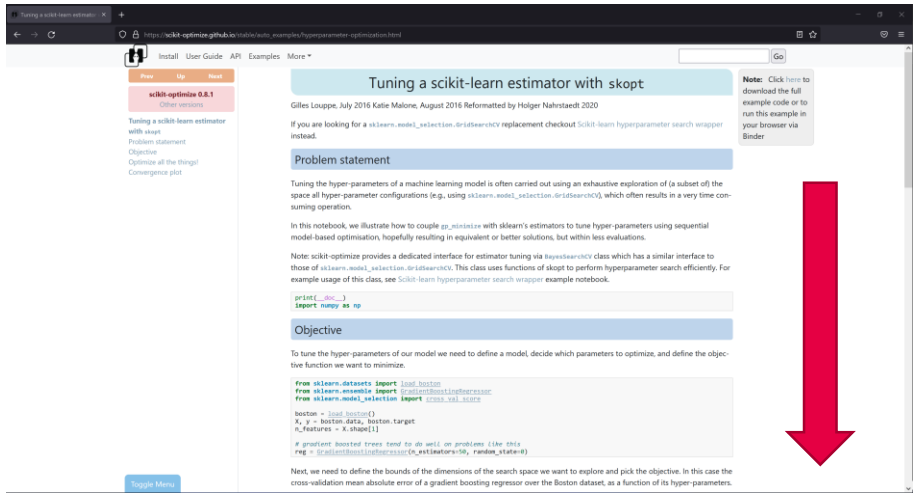
Convergence



Exercise 1: Bayesian Optimization (scikit-optimize)

(1) https://scikit-optimize.github.io/stable/index.html





scroll

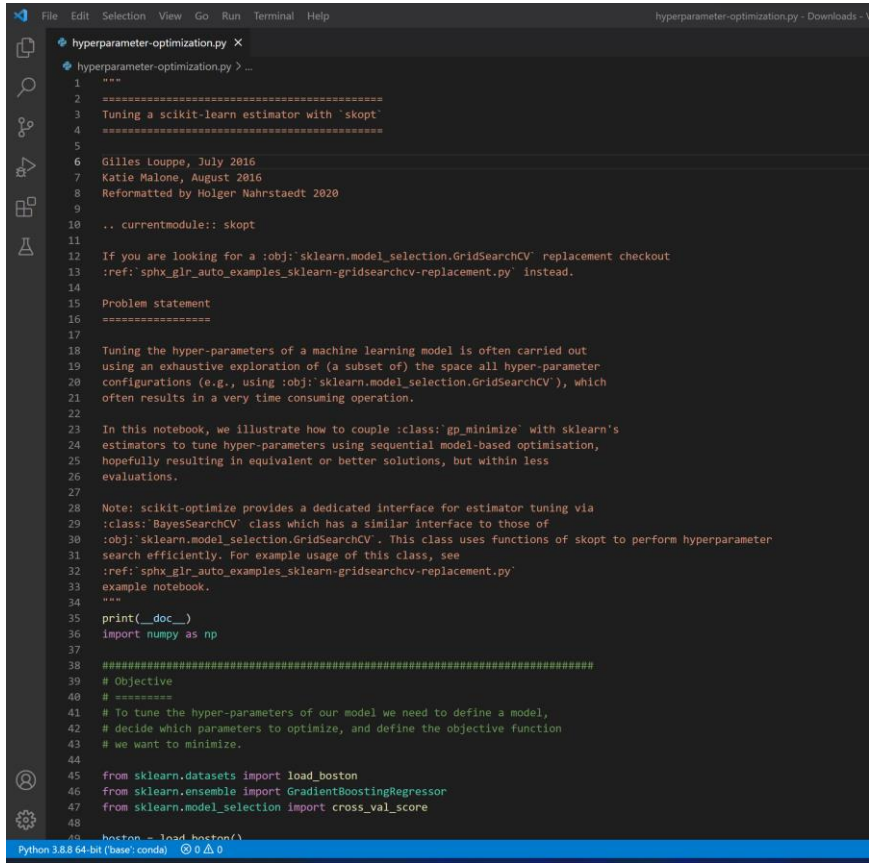
launch binder

(2)

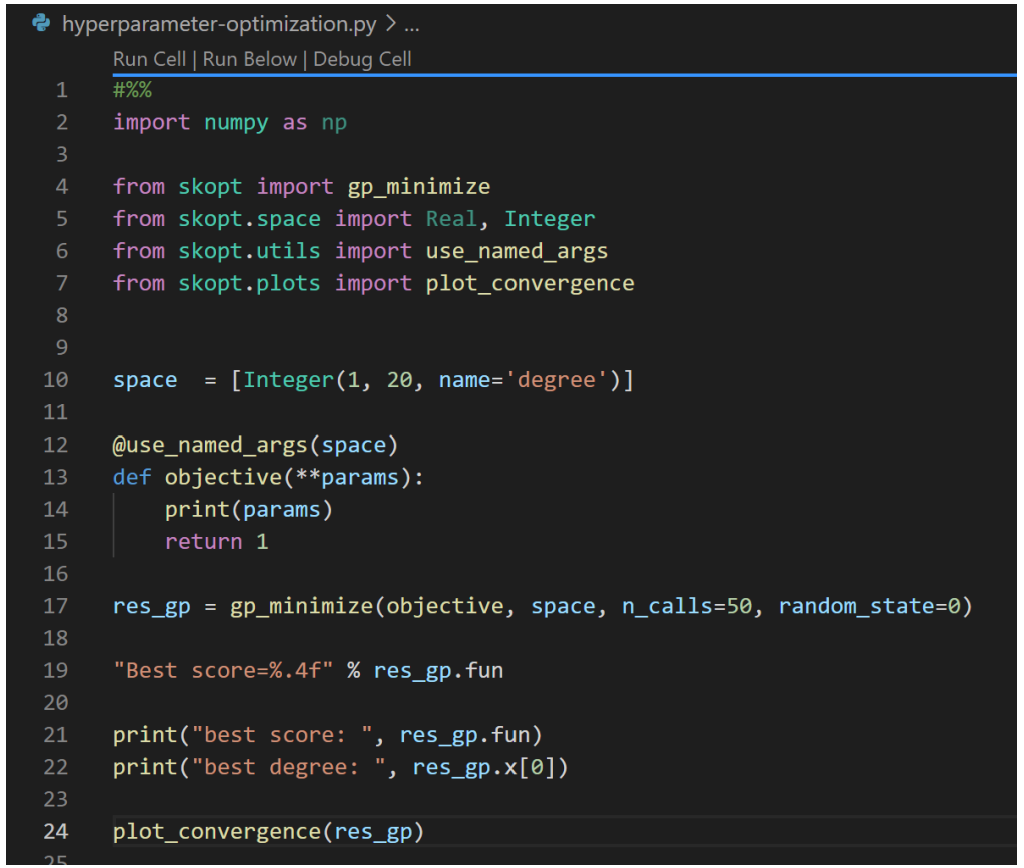
Download Python source code: hyperparameter-optimization.py

Download Jupyter notebook: hyperparameter-optimization.ipynb

Exercise 1: Bayesian Optimization (scikit-optimize)



Remove unnecessary code



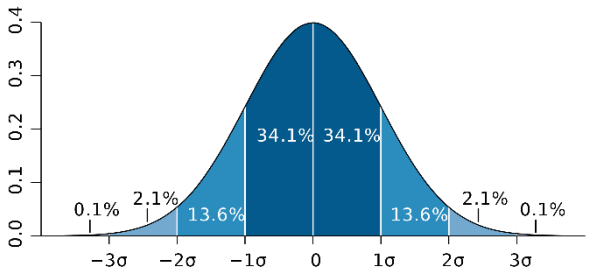
Exercise 2

Benchmark the Knapsack Problem
Open “Knapsack.py” and implement a generator for “v” and “w”. Increase items and measure the time to solve the problem. Repeat each benchmark (n-times) and calculate mean and variance time. Create a table and a plot to visualize your results.

https://realpython.com/python-timer/
https://docs.python.org/3/tutorial/datastructures.html#dictionaries
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mean.html
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.std.html
https://en.wikipedia.org/wiki/Standard_deviation

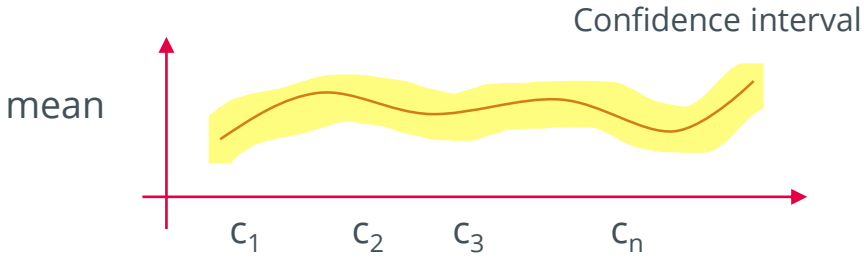
id	c1	c2	...
1	0.5s	0.6s	...
2	0.6s	0.7s	
...			
100	0.5s	0.8s	

Calculate **mean** and **std** on each column



Confidence interval
mean + 1.96*std
mean - 1.96*std
//std = standard deviation

https://en.wikipedia.org/wiki/1.96



Revision: Knapsack Problem (1/2)

Which items should be chosen to maximize the amount of money while keeping the overall weight under or equal to 6 kg?

Revision: Knapsack Problem (2/2)

Knapsack Problem (KP)

maximize: $\sum_{i=1}^{k} v_i \, b_i$

s.t.: $\sum_{i=1}^{k} w_i \, b_i \leq obj,$

$b_i \in \{0,1\}, i = 1, \ldots, k$

Pyomo is a Python-based, open-source optimization modeling language with a diverse set of optimization capabilities.

http://www.pyomo.org/

```
model = ConcreteModel()
model.elements = Set(initialize = v.keys())
model.x = Var(model.elements, within = Binary)
model.value = Objective(expr=sum(v[i]*model.x[i] for i in model.elements), sense=maximize)
model.weight = Constraint(expr=sum(w[i]*model.x[i] for i in model.elements) <= limit)
```

- https://www.gams.com/products/gams/gams-language/
- https://zimpl.zib.de/
- …

Revision: Multiple-Choice Knapsack Problem

Multiple-Choice Knapsack Problem (MCKP)

$$\text{maximize:} \quad \sum_{i=1}^{k} \sum_{j \in N_i} v_{ij}\, b_{ij}$$

$$\text{s.t.:} \quad \sum_{i=1}^{k} \sum_{j \in N_i} w_{ij}\, b_{ij} \leq obj,$$

$$\sum_{j \in N_i} b_{ij} = 1, i = 1, \ldots, k,$$

$$b_{ij} \in \{0,1\}, i = 1, \ldots, k, j \in N_i$$

Exercise 4: Implement a Decision Support System

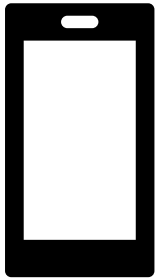
Knapsack?

Goal

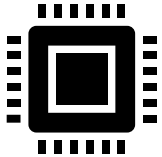
Original image (96,615 colors)

Quantized image (16 colors, K-Means)

Smartphone



CPU



?

Cloud

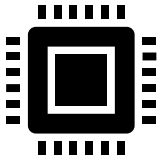
Exercise 4: General Idea

500€ 0.5kg

3000€ 0.2kg

30€ 5kg

CPU



600 MHz
800J 30s

900 MHz
1000J 20s

1,2 GHz
1800J 10s

Cloud

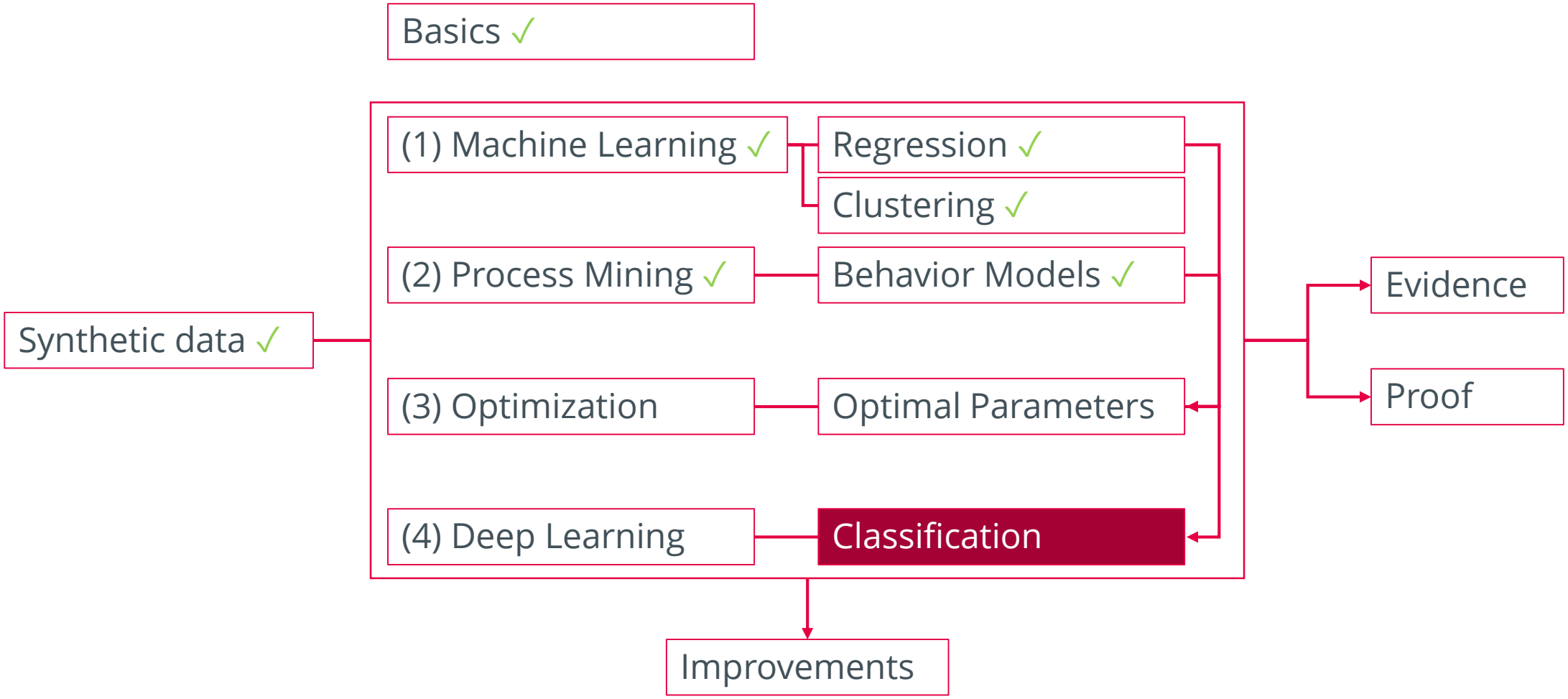
5G
1000J 10s

4G
1200J 50s

Edge
1300J 60s

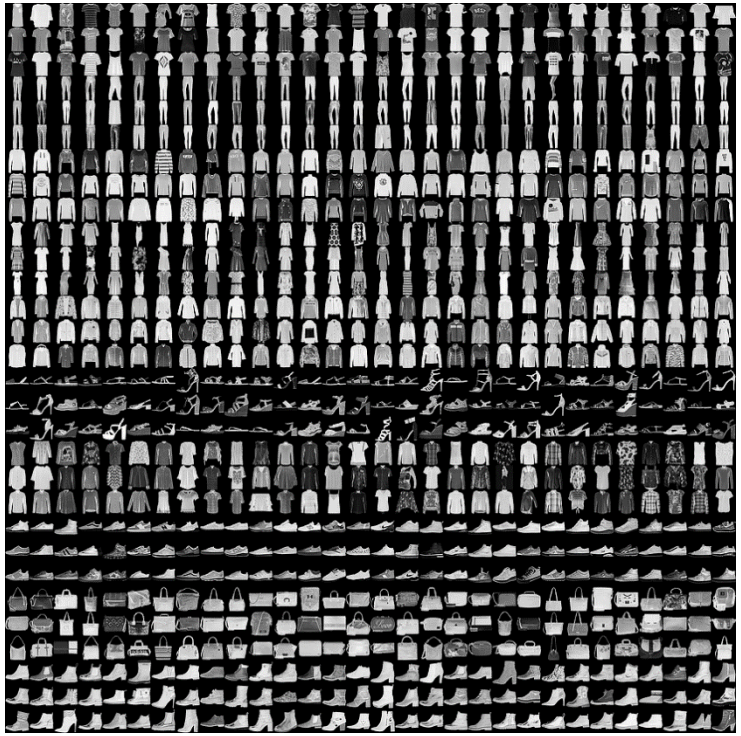
If the items are subdivided into k classes denoted N_i and exactly one item must be taken from each class, we get the multiple-choice knapsack problem

Summary: Advanced Topics in Algorithms

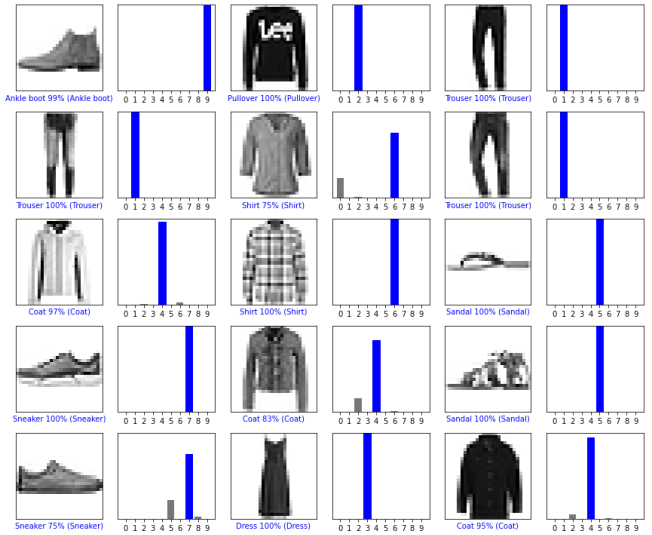


[Deep] Learning: Classification

Fashion-MNIST is a dataset of Zalando's article images



https://arxiv.org/pdf/1708.07747.pdf



XIAO, Han; RASUL, Kashif; VOLLGRAF, Roland. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

TH OWL

TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Thank you!