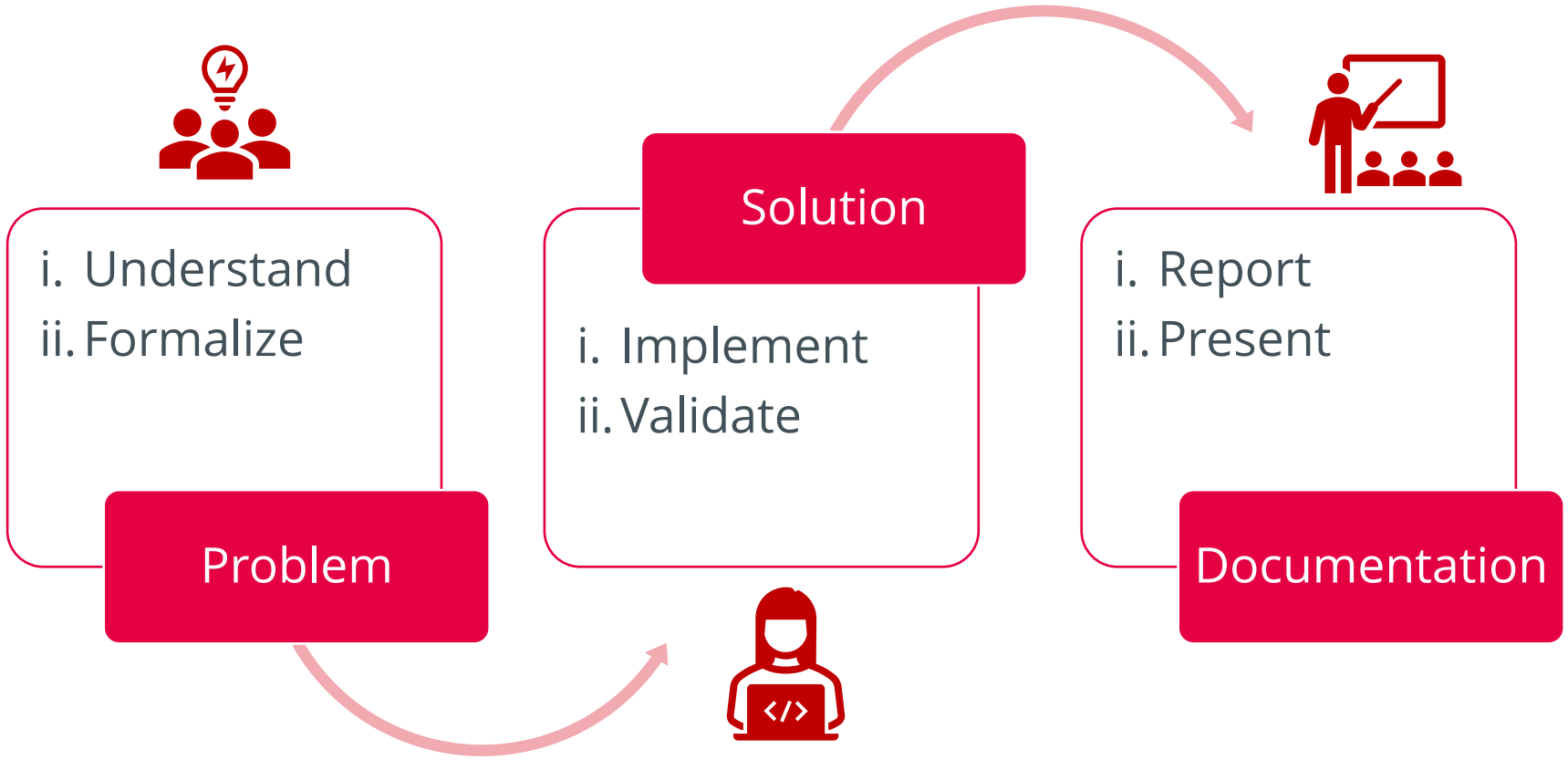
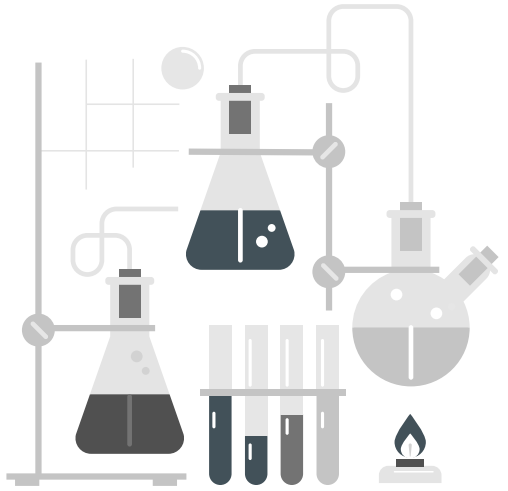
TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Welcome

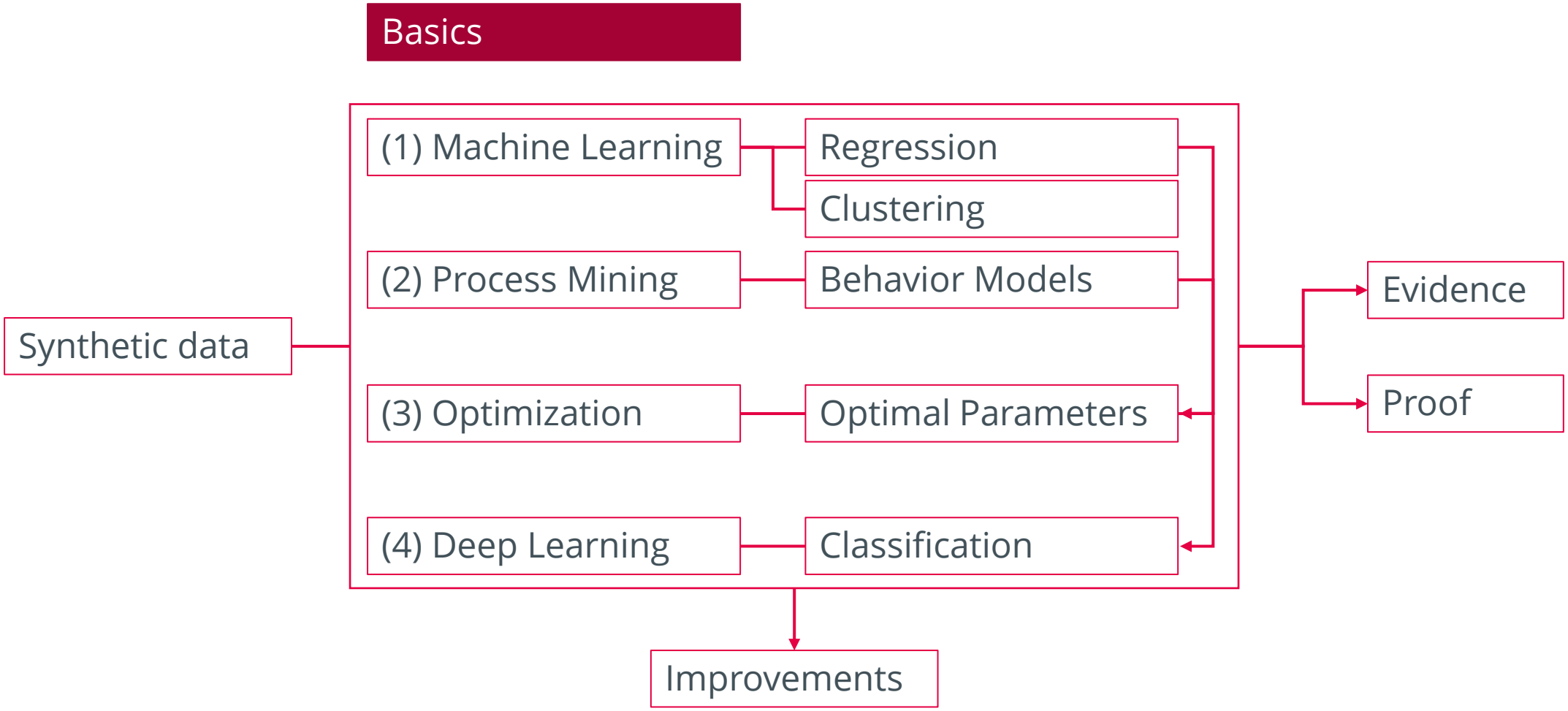
to Advanced Topics in Algorithms

Learning Objective: Become a Scientist

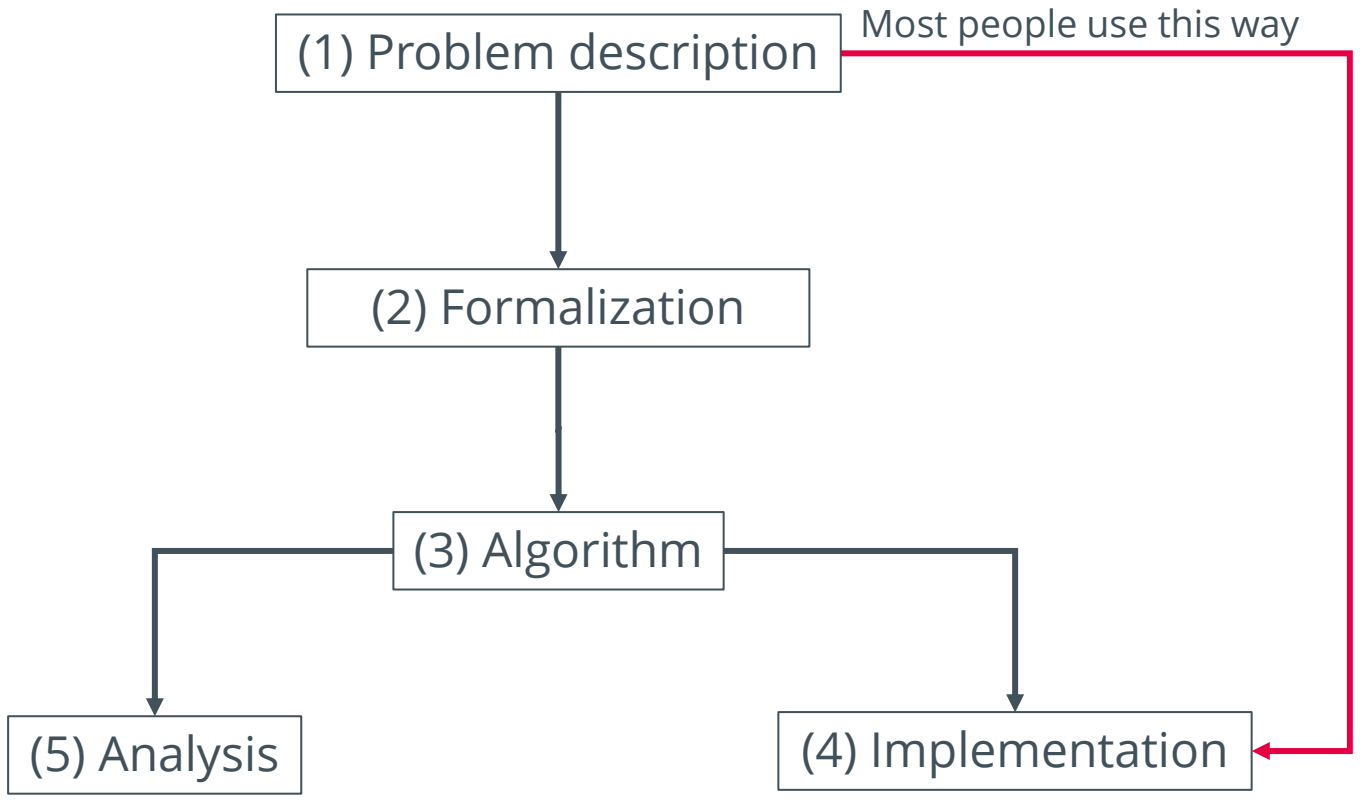


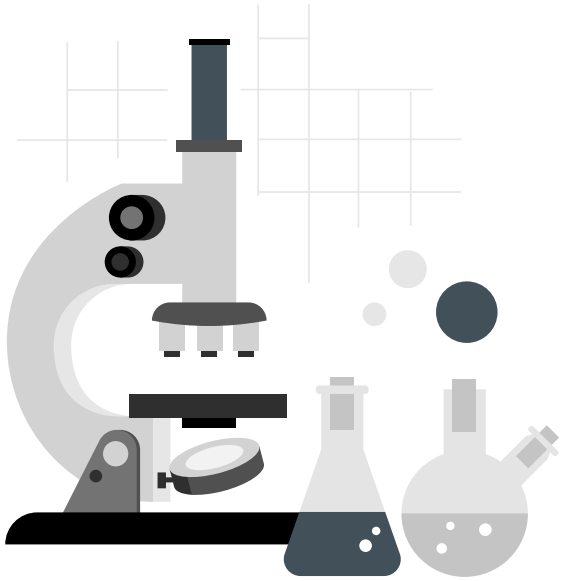


Summary: Advanced Topics in Algorithms



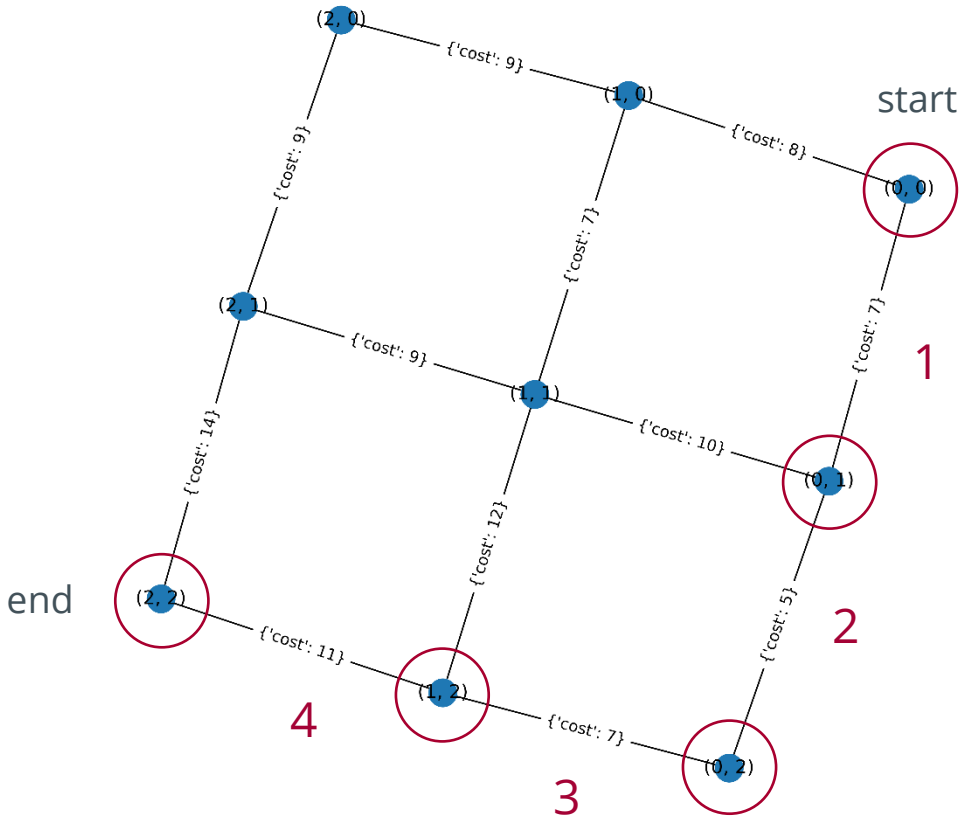
Concept: Algorithm Design





It's not rocket science

Algorithm Design: A*-Algorithm



cost of the path from the start node

$$f(n) = g(n) + h(n)$$

next node

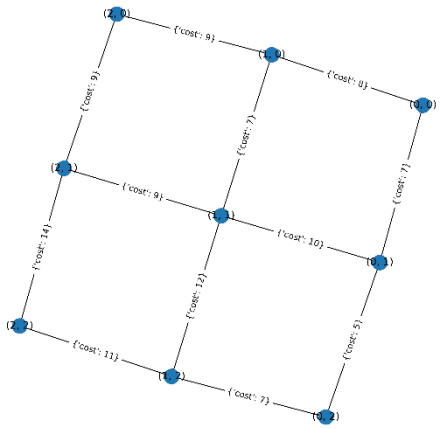
next	end	Euclidean distance
(1, 0)	(2, 2)	2.2360
(0, 1)	(2, 2)	2.2360
(0, 0)	(2, 2)	2.8284
(1, 1)	(2, 2)	1.4142
(0, 2)	(2, 2)	2.0
(2, 0)	(2, 2)	2.0
(1, 2)	(2, 2)	1.0
(2, 1)	(2, 2)	1.0
(2, 2)	(2, 2)	0.0

Algorithm Design: (1) Problem Description

The shortest path of a directed graph should be calculated.

Textual description of the problem

Input



directed graph

Algorithm

shortest path

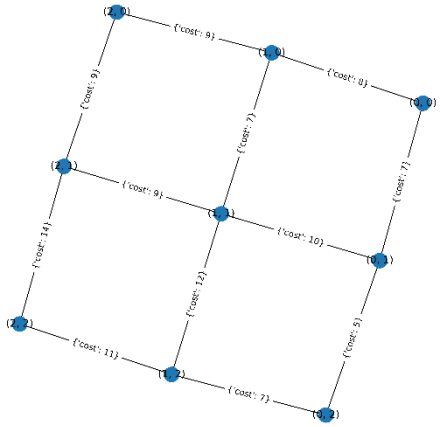
[(0, 0), (0, 1), (0, 2), (1, 2), (2, 2)]

Output

https://networkx.org

Algorithm Design: (2) Formalization

Input



directed graph

Algorithm

shortest path

Output

[(0, 0), (0, 1), (0, 2), (1, 2), (2, 2)]

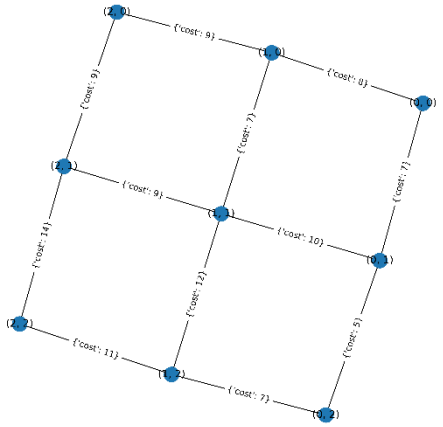
LaTeX

The directed graph D is defined as quadruple $D = (L, l_0, l_e, T)$. The finite set L describes the nodes, where l_0 describes the start node and l_e describes the end node. $T \subseteq L \times L$ describes the transitions between nodes.

$P = ((p_1, p_2)_i \mid i \in \{1, \ldots, m\})$ is defined as an n-tuple of...

Algorithm Design: (3) Algorithm

Input



directed graph

Algorithm

shortest path

Output

[(0, 0), (0, 1), (0, 2), (1, 2), (2, 2)]

LaTeX

```
\usepackage[linesnumbered,ruled,vlined]{algorithm2e}
```

Algorithm 1: The A^* ...

Input: Directed graph D
Output: n-tuple of point tuple

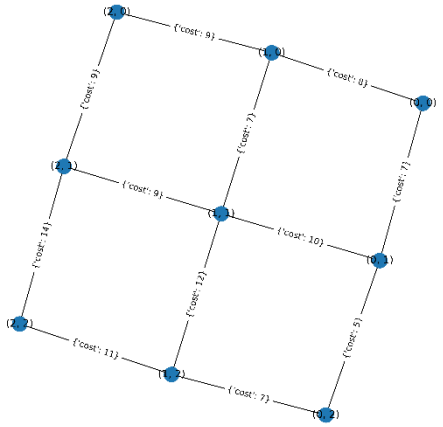
```
// Step 1:  (Calculate shortest path):
```

1 $P \leftarrow A^*(D)$

2 **return** P

Algorithm Design: (4) Implementation

Input



directed graph

Algorithm

shortest path

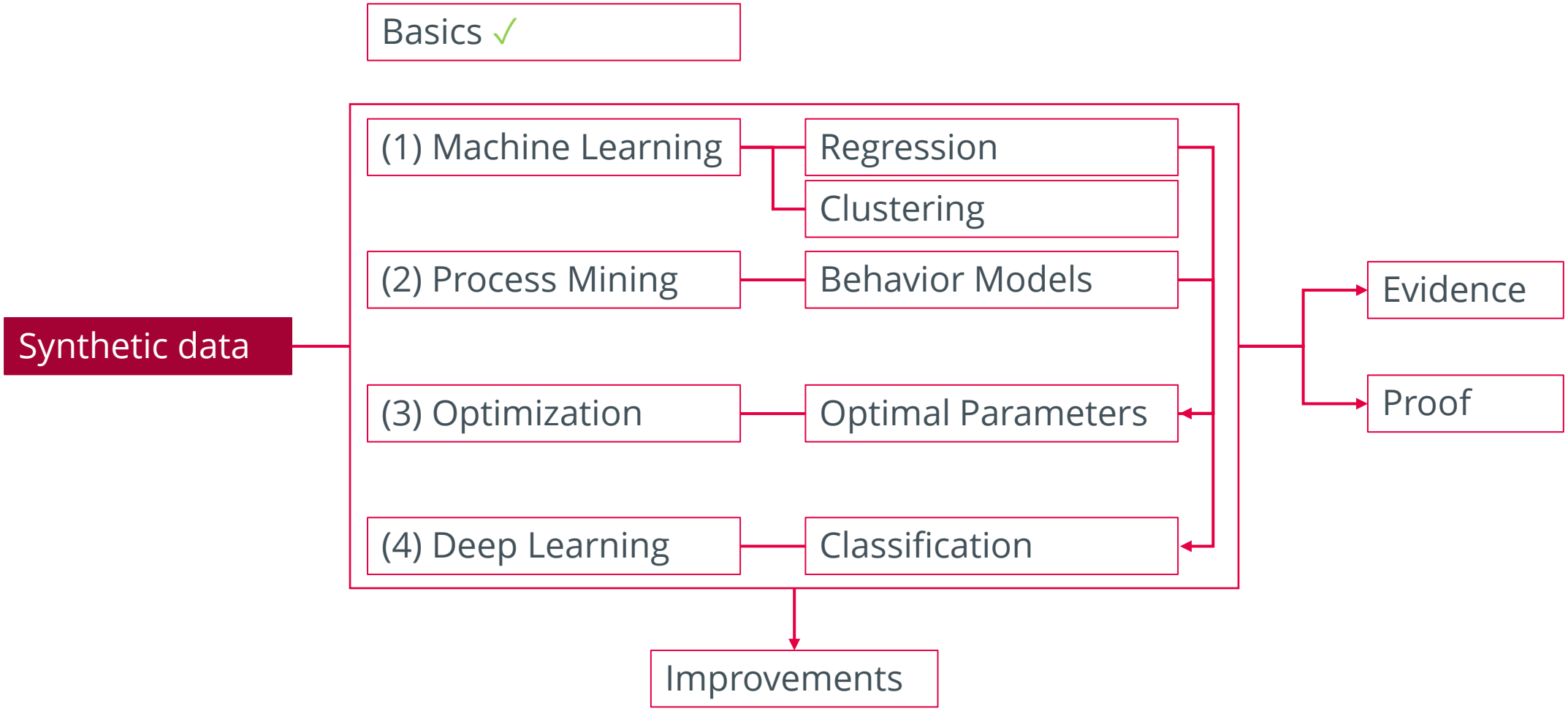
Output

[(0, 0), (0, 1), (0, 2), (1, 2), (2, 2)]

Python

```
Run Cell | Run Below | Debug Cell
# %%
import random
import math
import networkx as nx
import matplotlib.pyplot as plt
import matplotlib

from networkx.classes.graph import Graph
from networkx.algorithms.shortest_paths.weighted import _weight_function
from heapq import heappush, heappop
from itertools import count

matplotlib.rcParams.update({'font.size': 22, 'font.family': 'Arial'})

def star(G, source, target, heuristic=None, weight="weight"):

    push = heappush
    pop = heappop
    weight = _weight_function(G, weight)
```

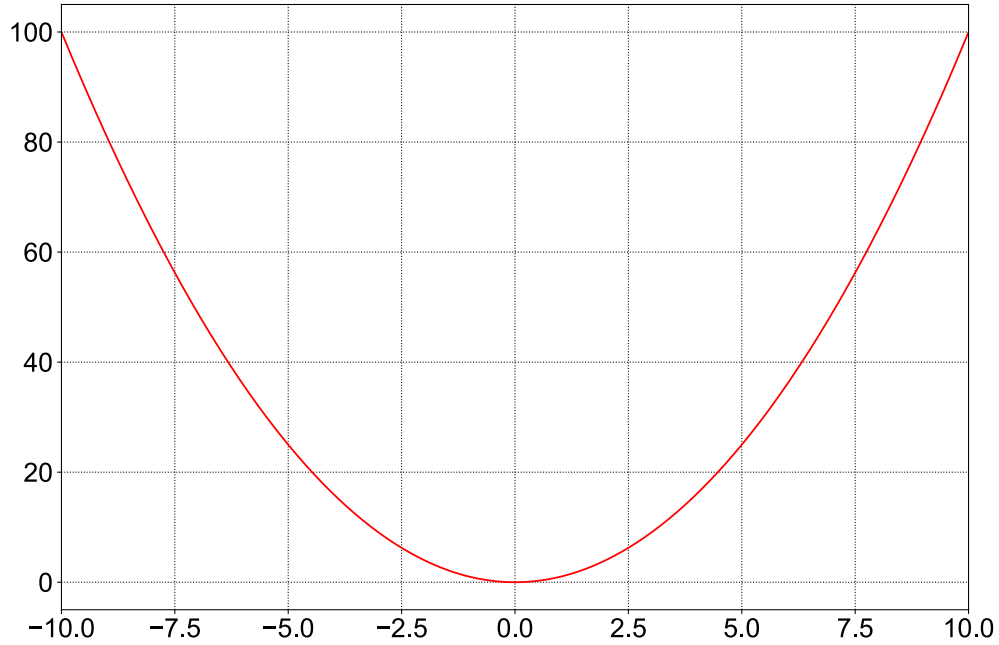
Summary: Advanced Topics in Algorithms



Synthetic Data 1/2

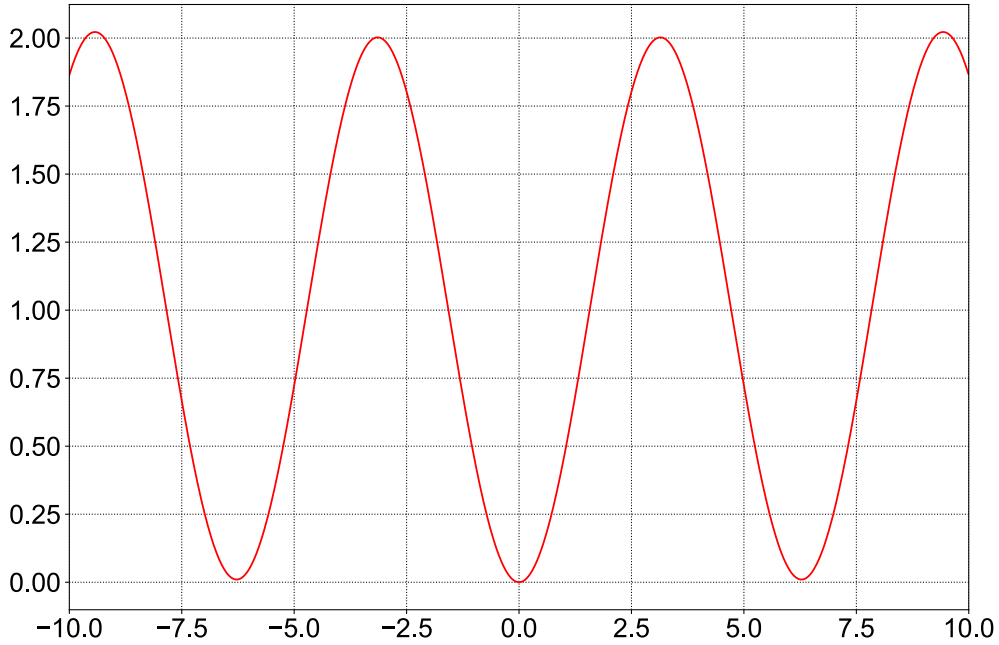
Sphere

$$f_s(x) = \sum_{i=1}^{n} x_i^2$$



Griewank

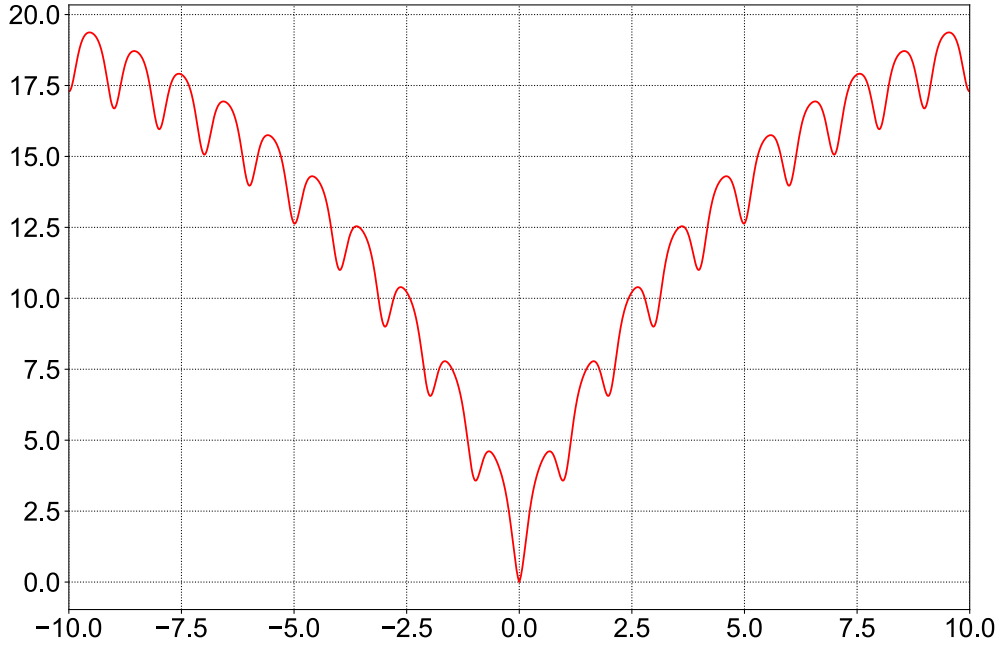
$$f_g(x) = 1 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{(i)}}\right) + \sum_{i=1}^{n} \frac{x_i^2}{4000}$$



Synthetic Data 2/2

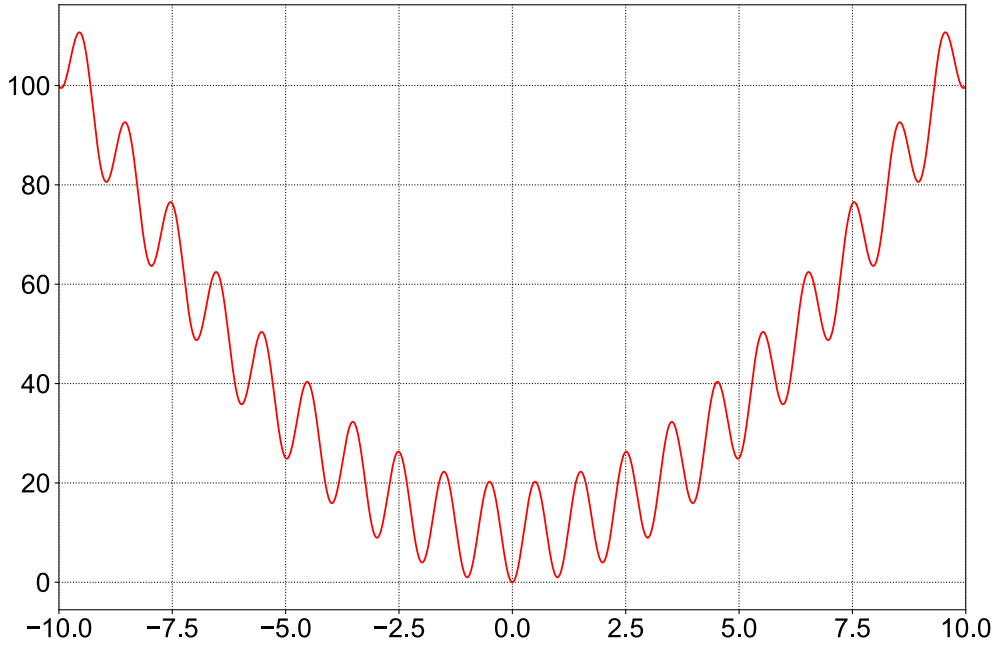
Ackley

$$f_{\mathrm{a}}(x) = 20 - 20 * \exp\left[-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}\right] + \exp(1) - \exp\left[\frac{\sum_{i=1}^{n}\cos(2\pi x_i)}{n}\right]$$



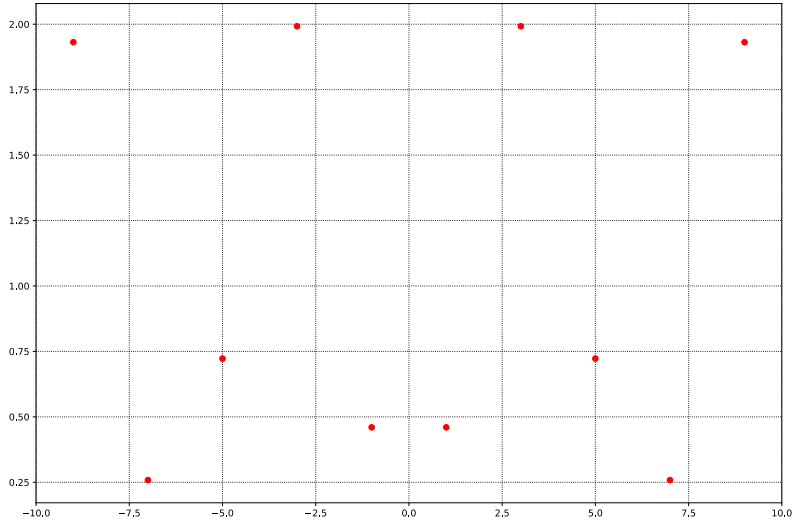
Rastrigin

$$f_{\mathrm{r}}(x) = 10 \cdot \left[n - \sum_{i=1}^{n}\cos(2\pi x_i)\right] + \sum_{i=1}^{n} x_i^2$$

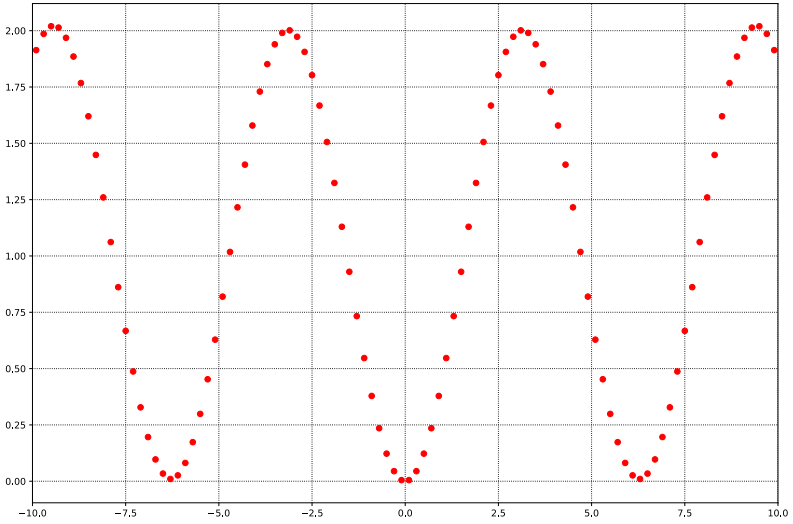


Synthetic Data: Latin Hypercube Sampling

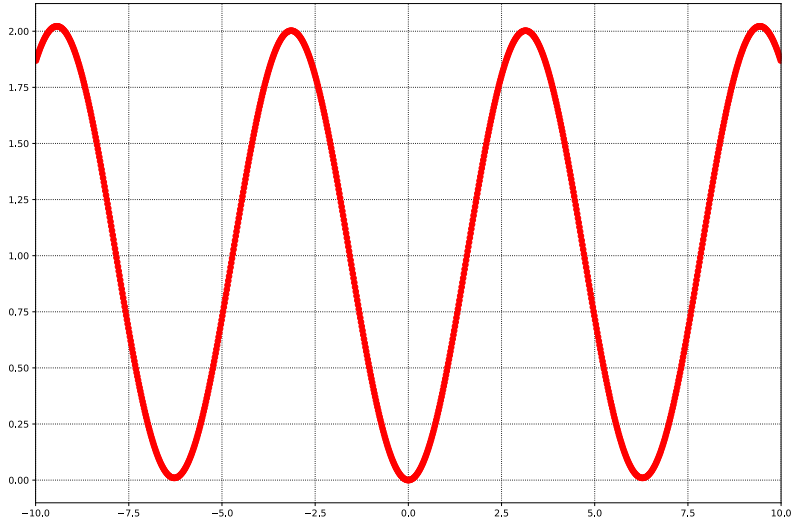
10



100



1000

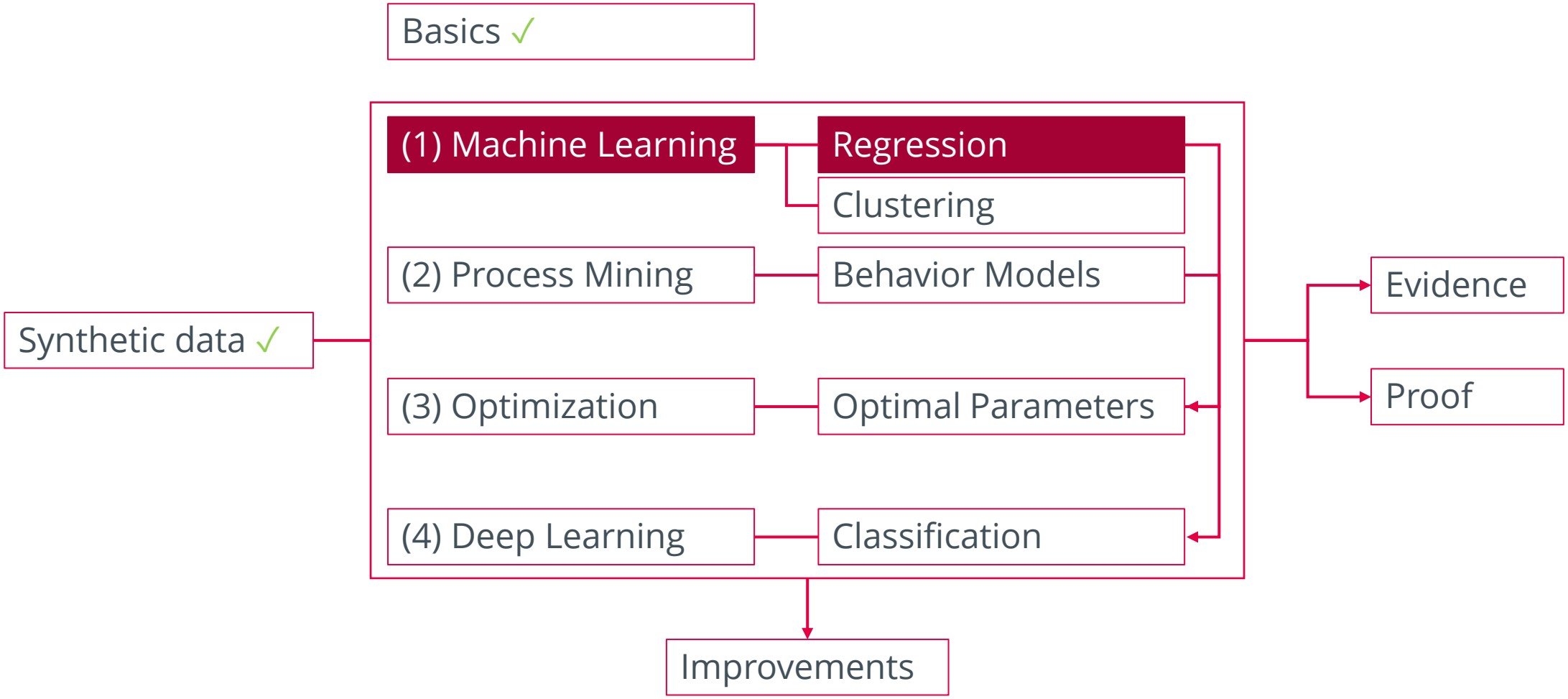


It is among the most popular sampling techniques in computer experiments thanks to its simplicity and projection properties with high-dimensional problems.

Jin, R. and Chen, W. and Sudjianto, A. (2005), “An efficient algorithm for constructing optimal design of computer experiments.” Journal of Statistical Planning and Inference, 134:268-287.

https://smt.readthedocs.io/en/latest/_src_docs/sampling_methods/lhs.html

Summary: Advanced Topics in Algorithms



Symbolic Regression 1/3

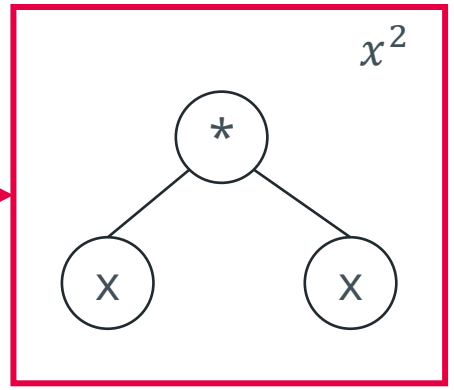
Input

Time	Energy
09:59	0
10:00	21.16
…	
10:05	126.97
10:06	0
…	

- mean squared error
- root mean squared error

Symbolic Regression

Output

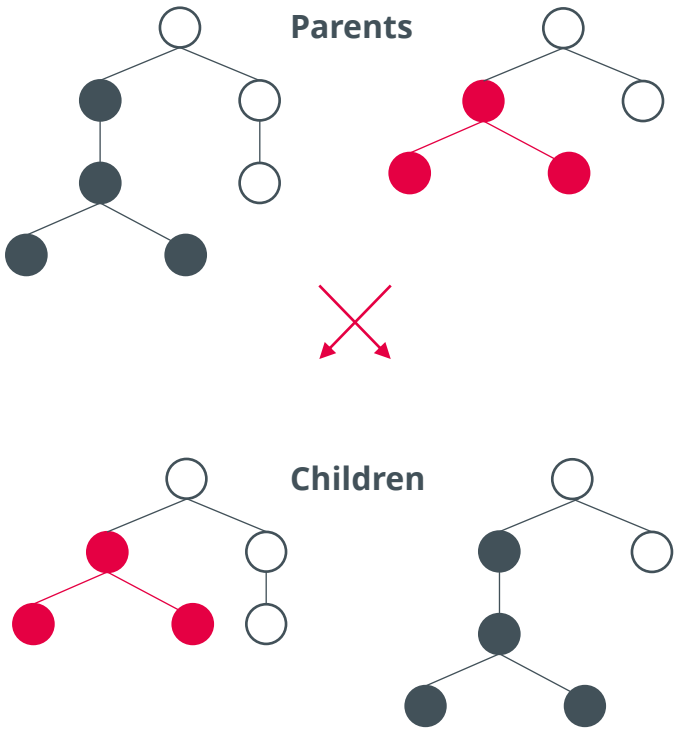


Primitives

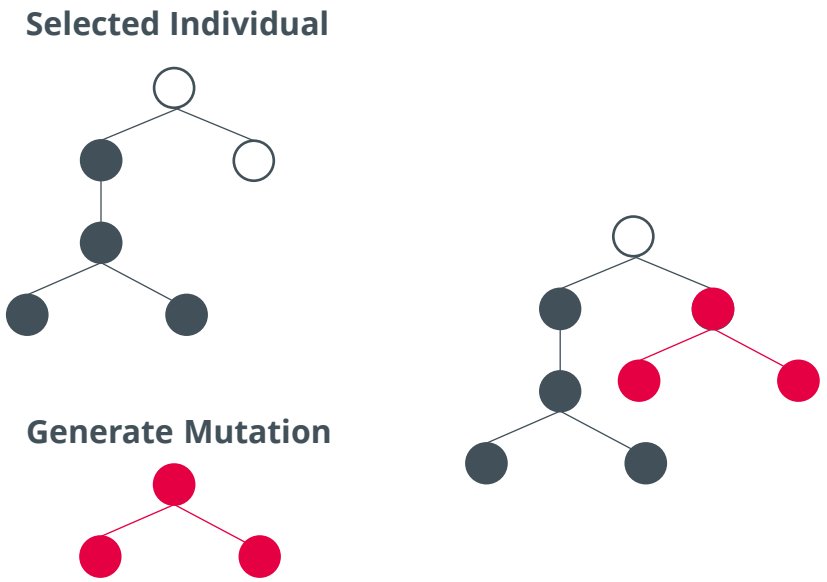
- add
- sub
- mult
- …

Symbolic Regression 2/3

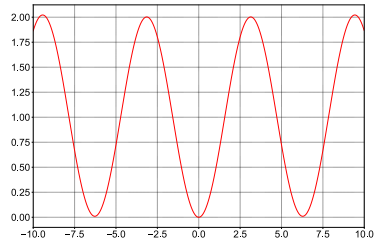
Crossover



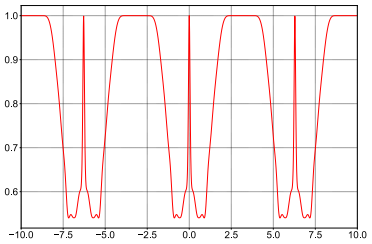
Mutation



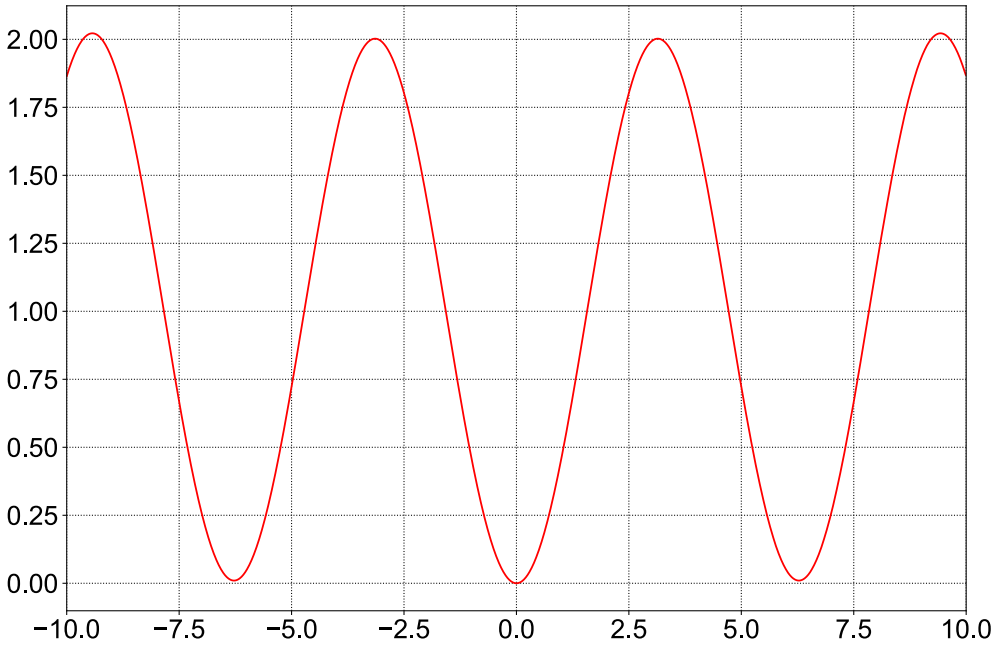
Evaluate

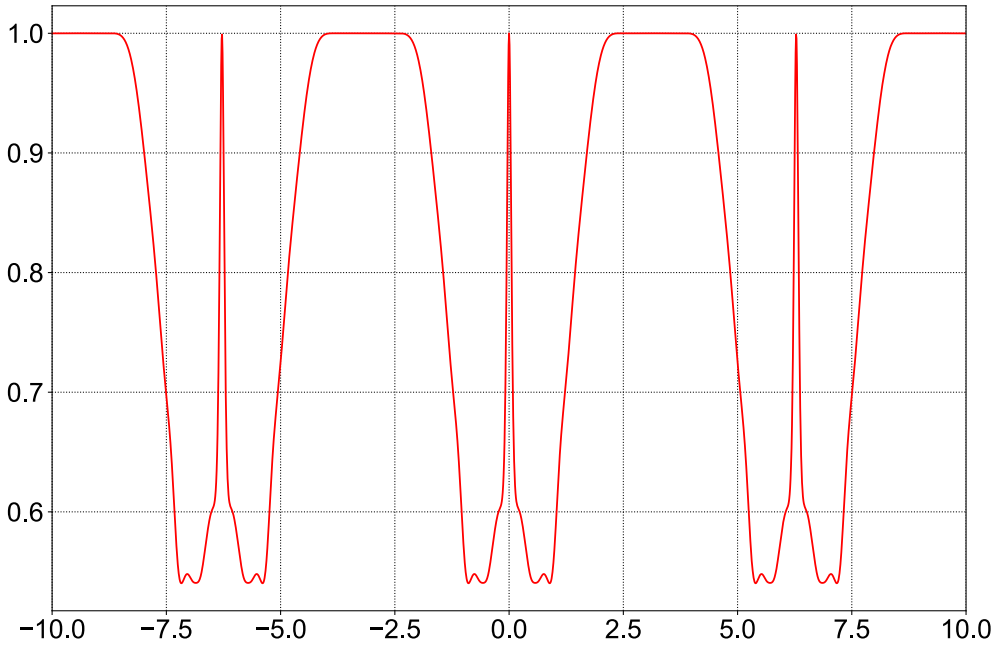


e.g. mean squared error

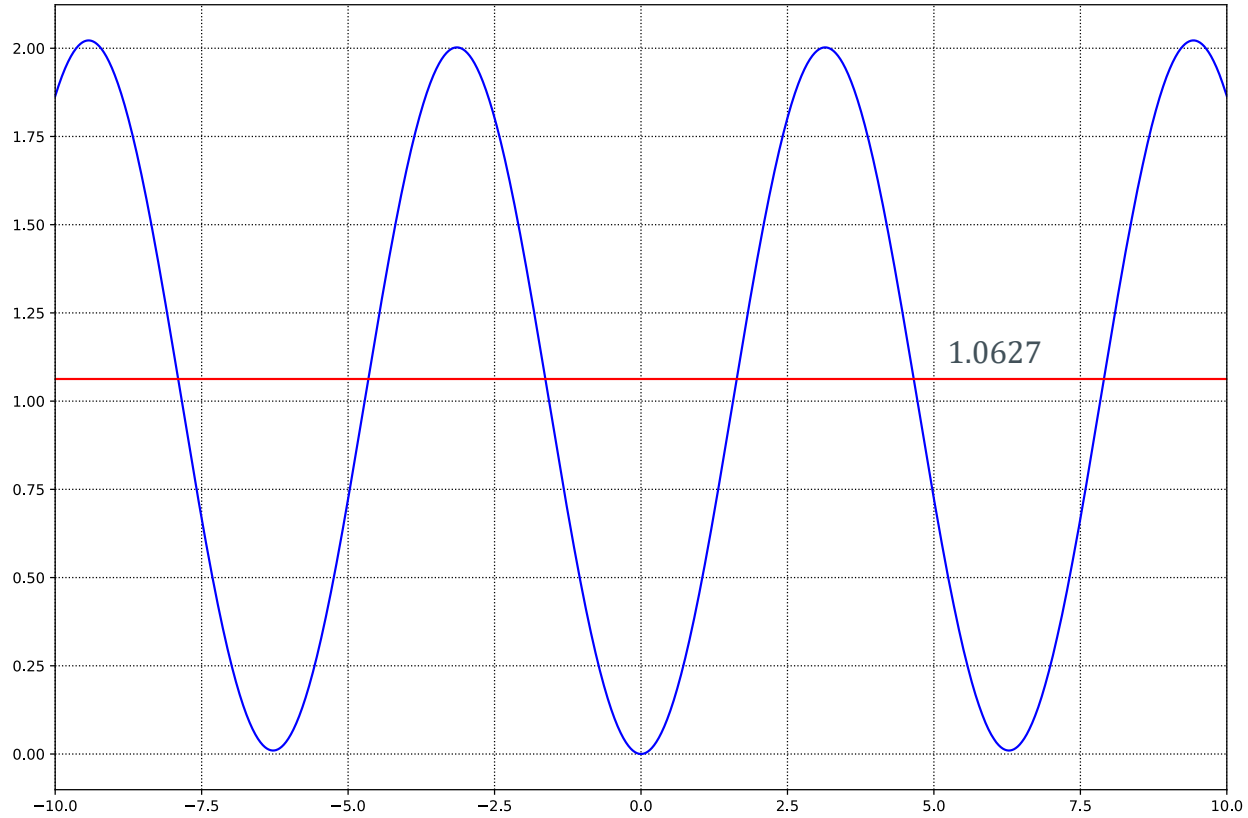


Symbolic Regression 3/3

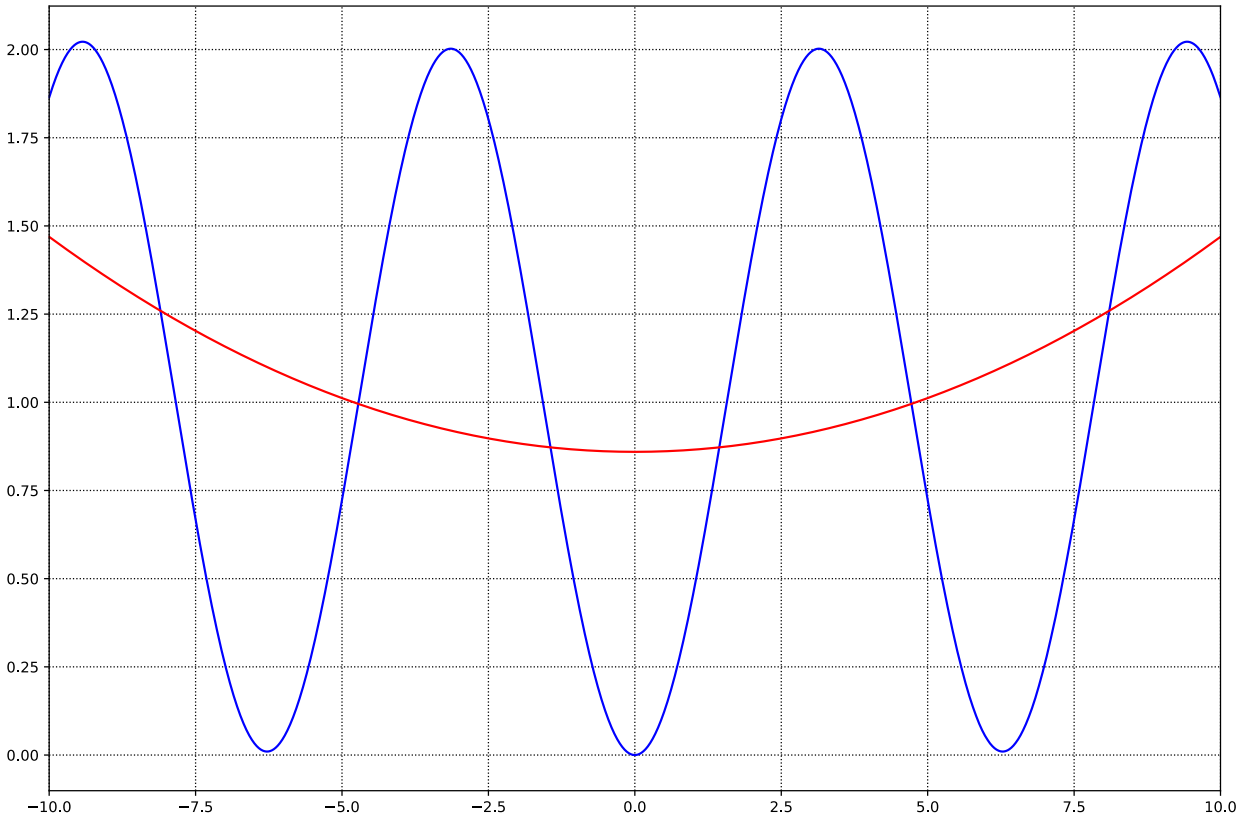




Regression with Polynomial Features: Step 1

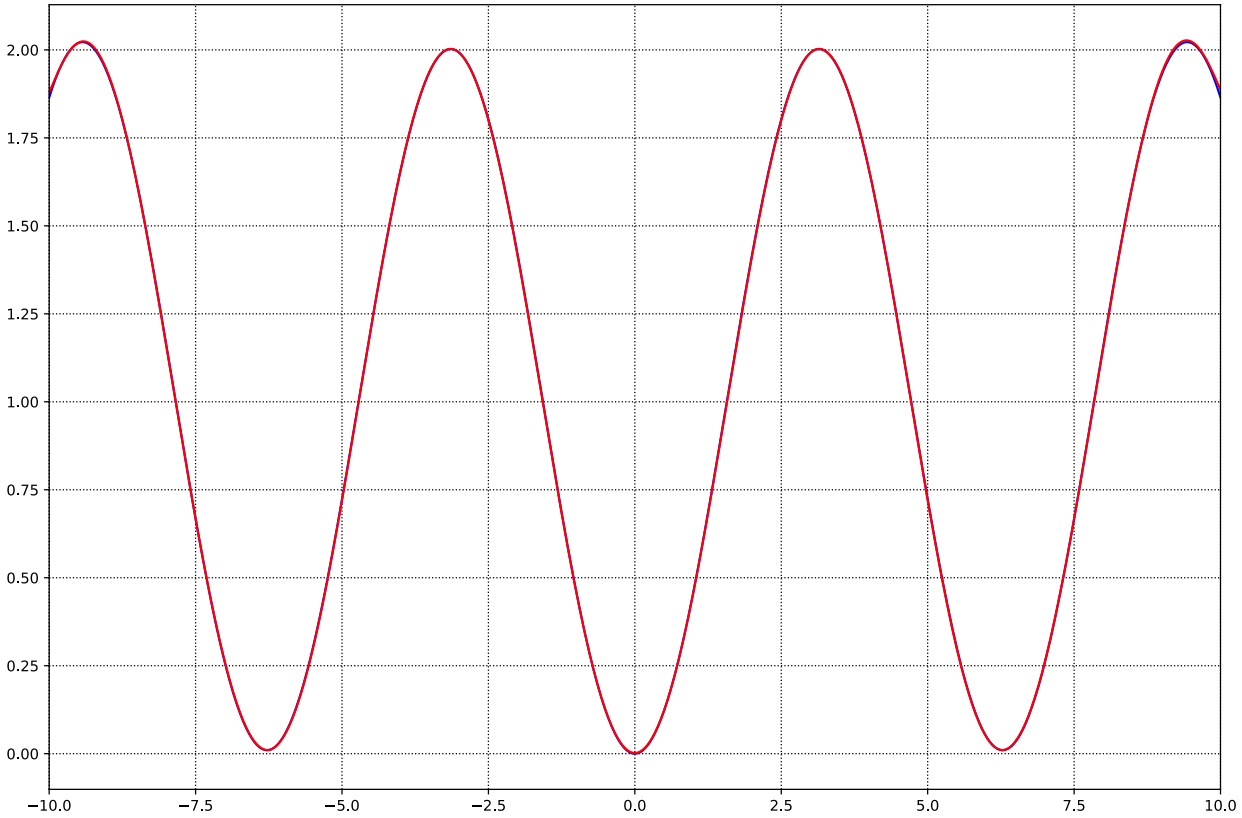


Regression with Polynomial Features: Step 2



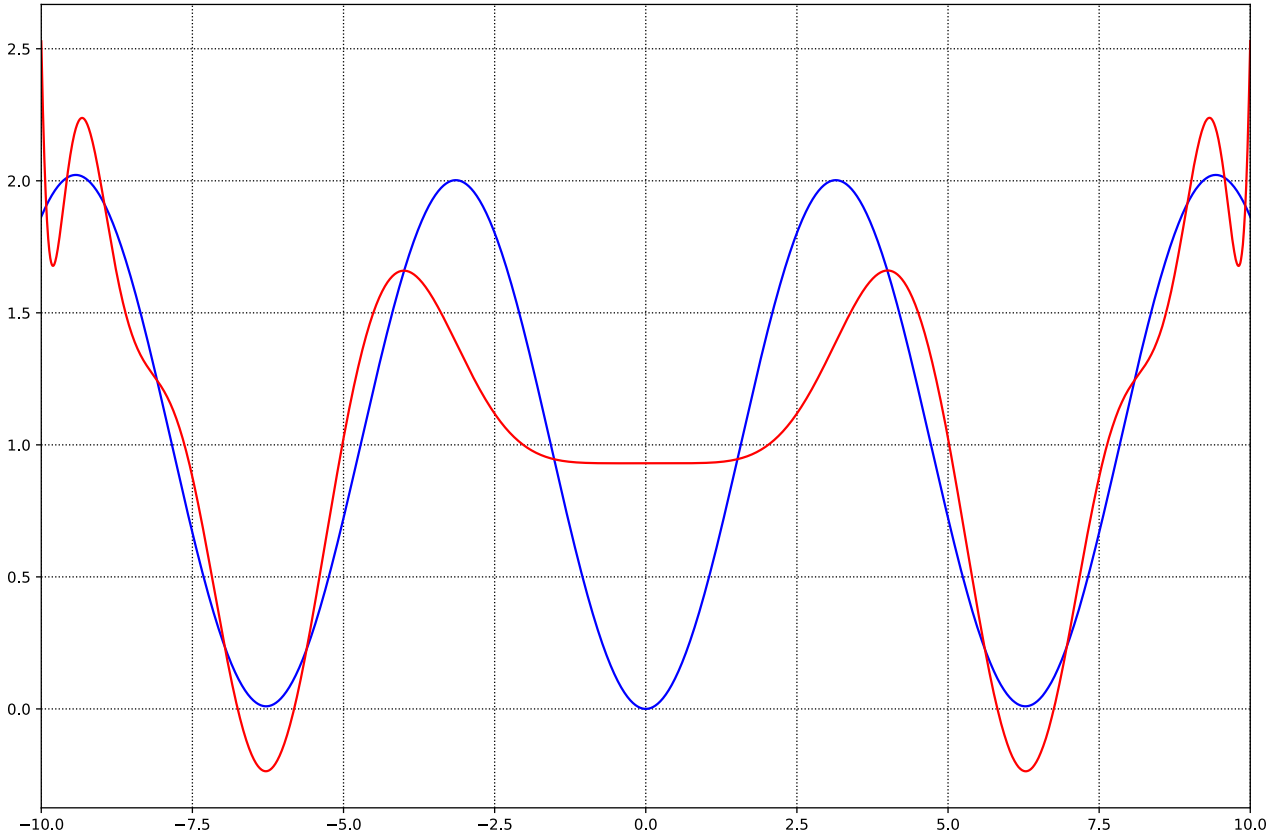
$$0.00609567x_0^2 + 0.8595$$

Regression with Polynomial Features: Step 14



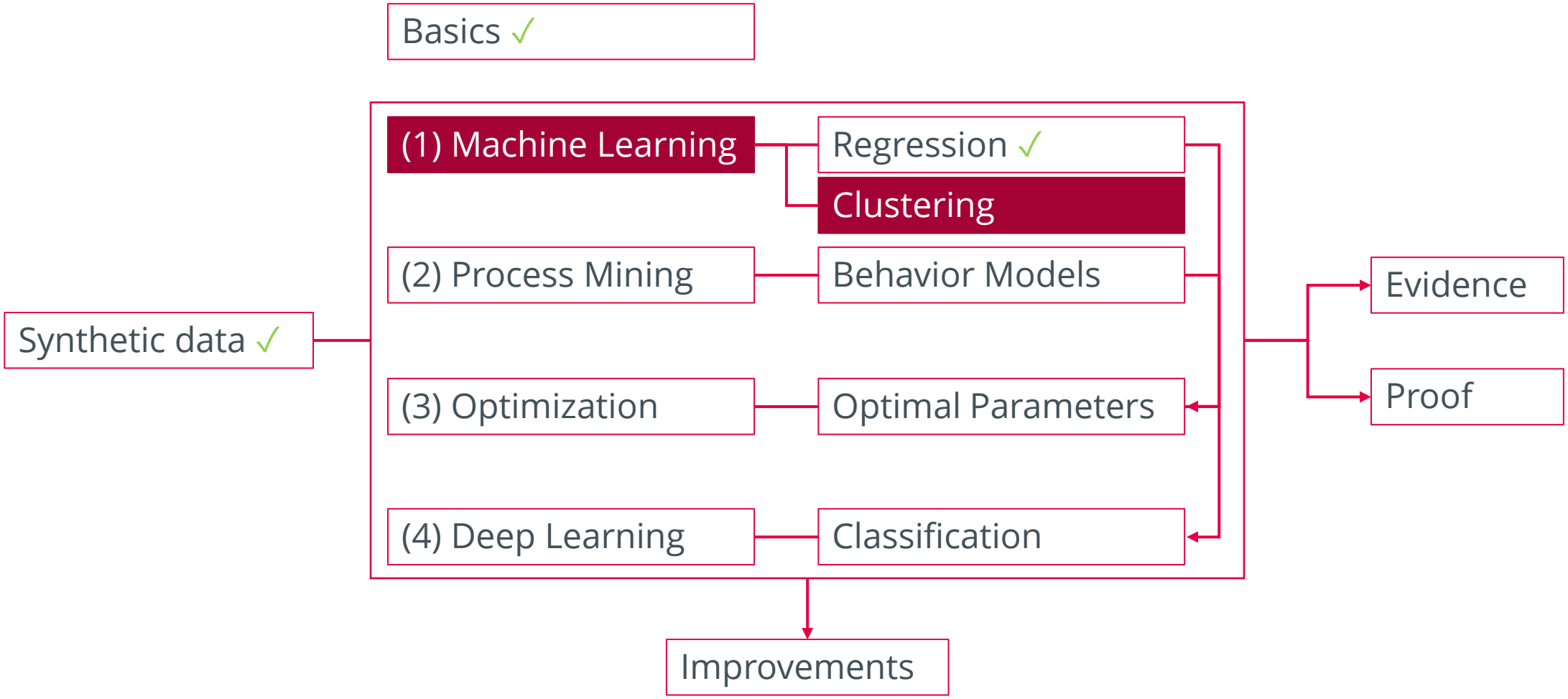
$$2.0 \cdot 10^{-7} x_0^{10} - 2.167 \cdot 10^{-5} x_0^8 + 0.00131727 x_0^6 - 4.0 \cdot 10^{-8} x_0^5 - 0.04085166 x_0^4 + 1.78 \cdot 10^{-6} x_0^3 + 0.49669338 x_0^2 - 3.354 \cdot 10^{-5} x_0 + 0.0025$$

Regression with Polynomial Features: Step 18

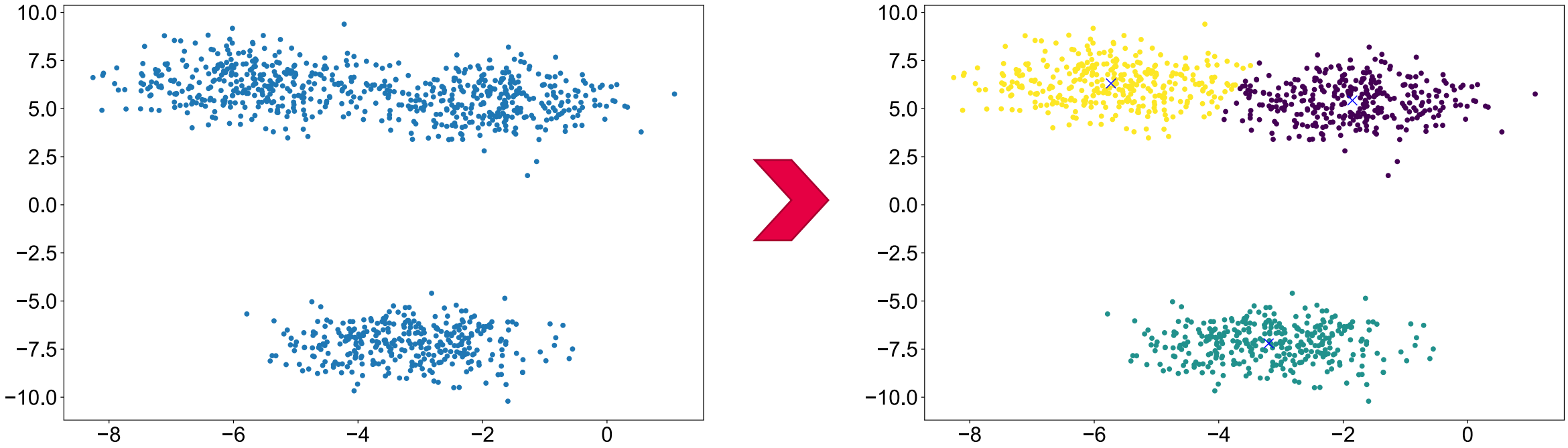


$$-1.4 \cdot 10^{-7} x_0^{12} + 6.5 \cdot 10^{-6} x_0^{10} + 2.0 \cdot 10^{-8} x_0^9 - 0.0001587 x_0^8 - 4.2 \cdot 10^{-7} x_0^7 + 0.00150314 x_0^6 + 3.51 \cdot 10^{-6} x_0^5 + 0.00022478 x_0^4 + 6.6 \cdot 10^{-7} x_0^3 + 1.824 \cdot 10^{-5} x_0^2 + 1.589 \cdot 10^{-5} x_0 + 0.9302$$

Summary: Advanced Topics in Algorithms

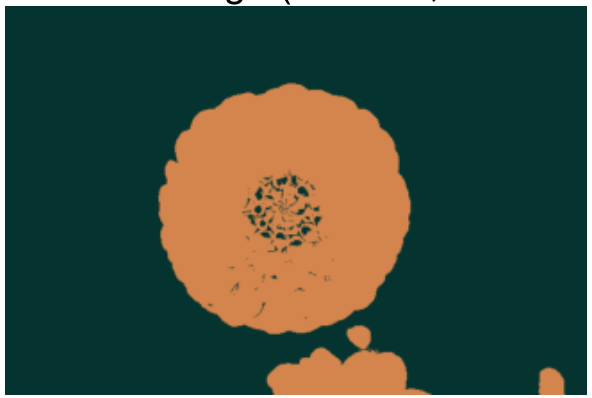


K-Means



K-Means

Quantized image (2 colors, K-Means)



Quantized image (16 colors, K-Means)

Quantized image (32 colors, K-Means)

Quantized image (64 colors, K-Means)

Quantized image (128 colors, K-Means)

Original image (96,615 colors)

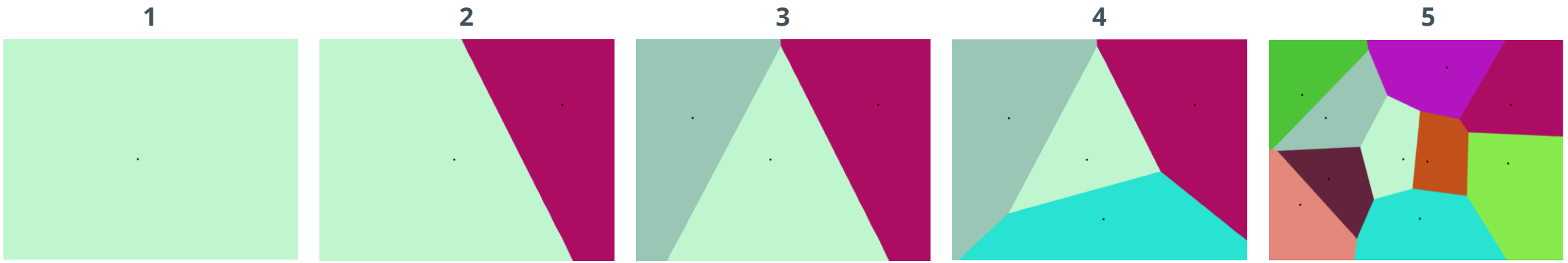
K-Means

The algorithm can also be understood through the concept of Voronoi diagrams [1]:

1. First the Voronoi diagram of the points is calculated using the current centroids. Each segment in the Voronoi diagram becomes a separate cluster.
2. Secondly, the centroids are updated to the mean of each segment.

The algorithm then repeats this until a stopping criterion is fulfilled.

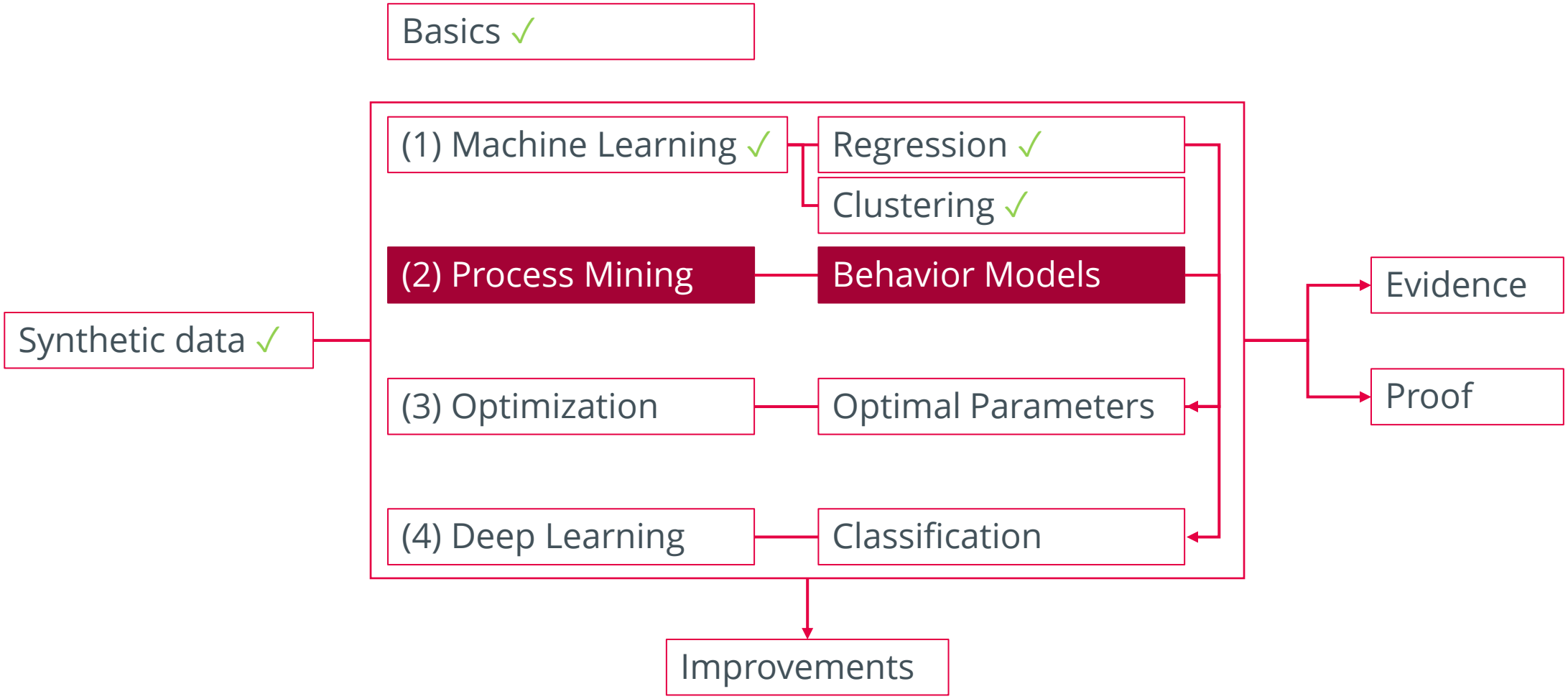
Voronoi Diagram [2]



Fortune's algorithm: O(n log n) time and O(n) space

[1] https://scikit-learn.org/stable/modules/clustering.html#k-means
[2] http://alexbeutel.com/webgl/voronoi.html

Summary: Advanced Topics in Algorithms

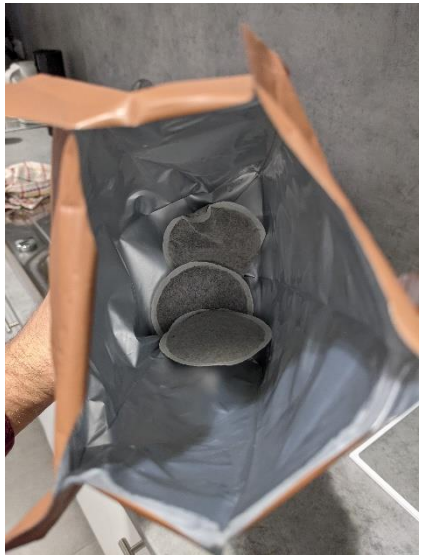


Produce Coffee

(1) Power on

Activity	Timestamp
power on	12:00:00

(2) Take Pad



Activity	Timestamp
power on	12:00:00
take pad	12:05:00

(3) Add Pad

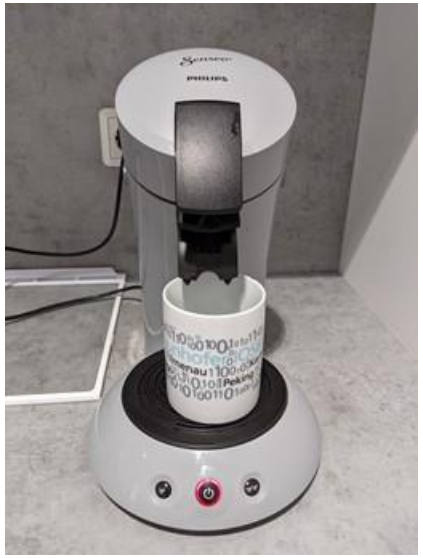
Activity	Timestamp
power on	12:00:00
take pad	12:05:00
add pad	12:05:30

(4) Close

Activity	Timestamp
power on	12:00:00
take pad	12:05:00
add pad	12:05:30
close	12:05:40

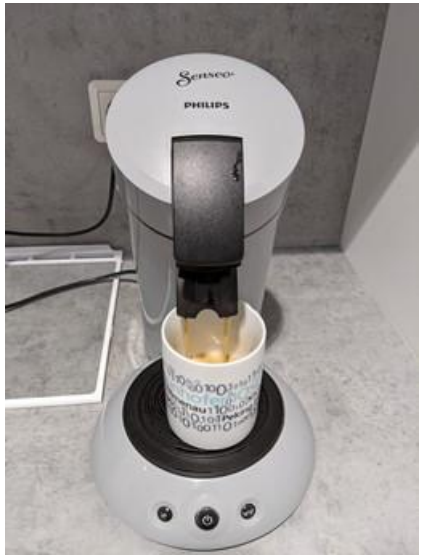
Produce Coffee

(5) Start



Activity	Timestamp
start	12:05:45

(6) Produce

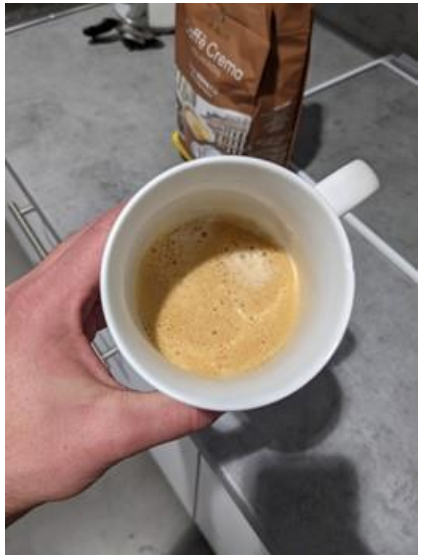


Activity	Timestamp
start	12:05:45
produce	12:07:00

(7) Remove Pad

Activity	Timestamp
start	12:05:45
produce	12:07:00
remove pad	12:07:30

(8) Enjoy



Activity	Timestamp
start	12:05:45
produce	12:07:00
remove pad	12:07:30
enjoy	12:20:00

Process Mining

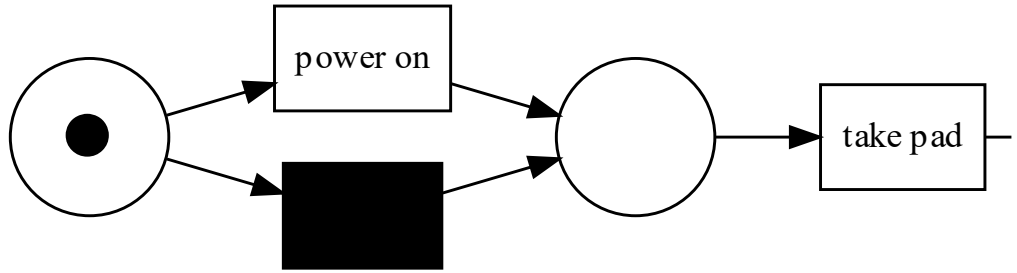
Event log*

Case	Activity	Timestamp
1	power on	12:00:00
1	take pad	12:05:00
1	add pad	12:05:30
1	close	12:05:40
1	start	12:05:45
1	produce	12:07:00
1	remove pad	12:07:30
1	enjoy	12:20:00
2	power on	12:45:00
…		

- Alpha miner
- Heurist miner
- Inductive miner

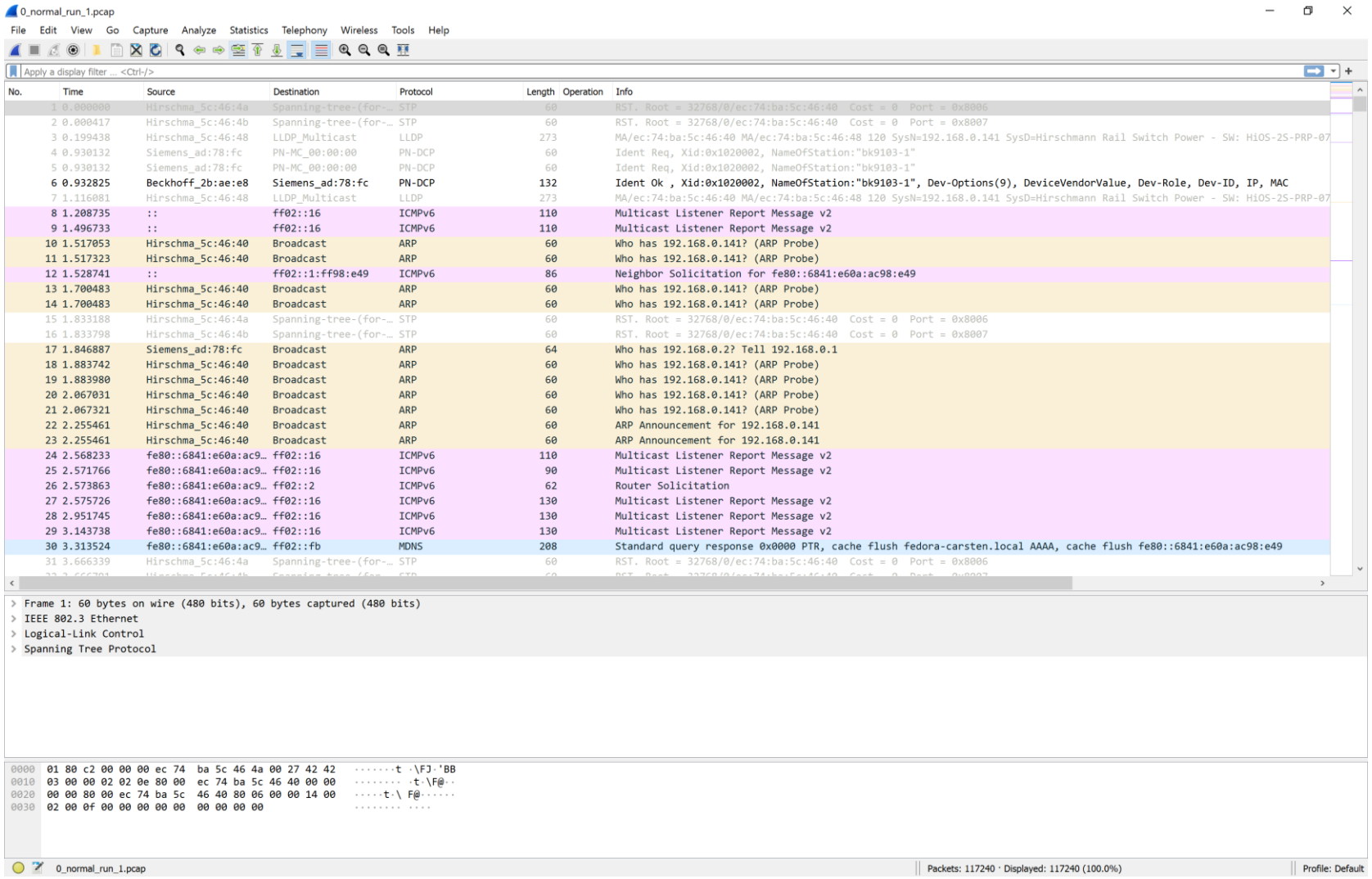
Process discovery

Petri Net



*IEEE XES is a standard format describing how event logs are stored
http://www.xes-standard.org/

Excurse: Deep Packet Inspection



Excurse: Deep Packet Inspection

Pcap

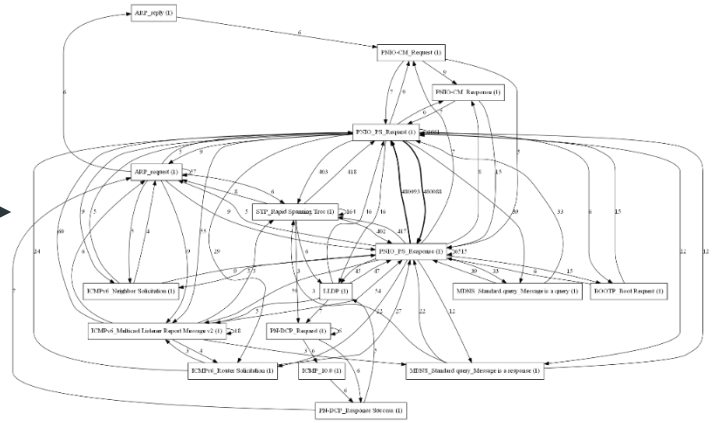
(1) Attack
(2) Normal

Event log

case_id:	enumerated scenario
activity:	what action is performed
Start_time:	when action started
Ressource:	who started the action

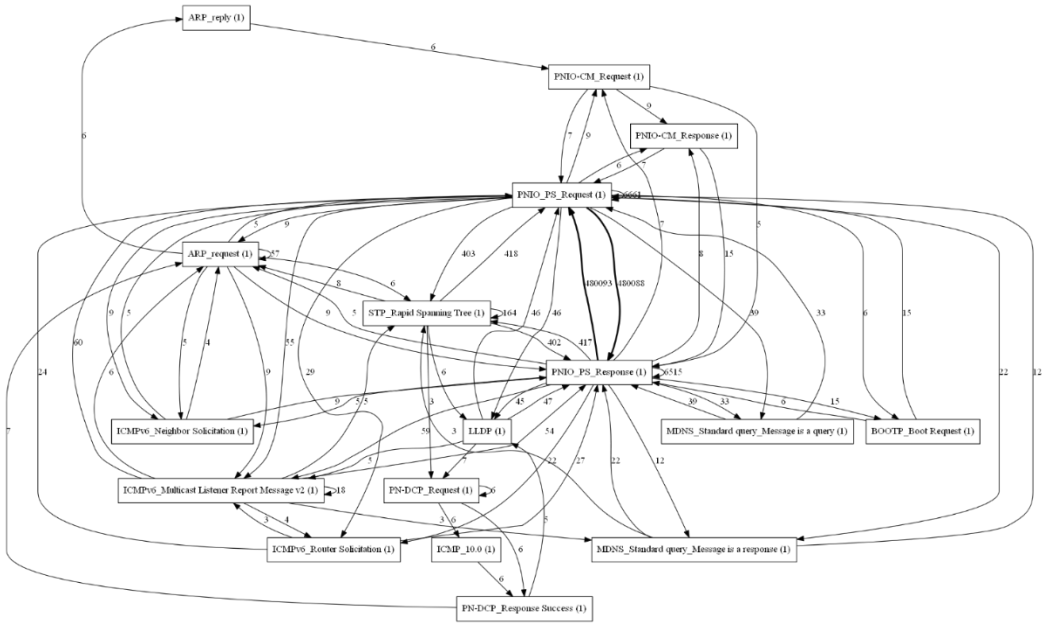
Action (e.g. request / response)

pm4py

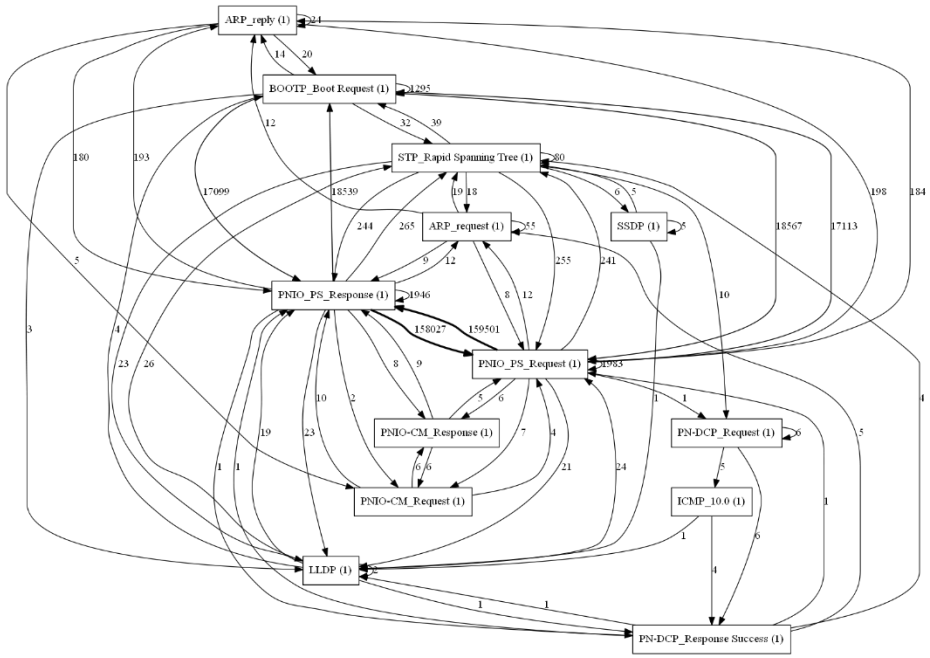


Excurse: Deep Packet Inspection

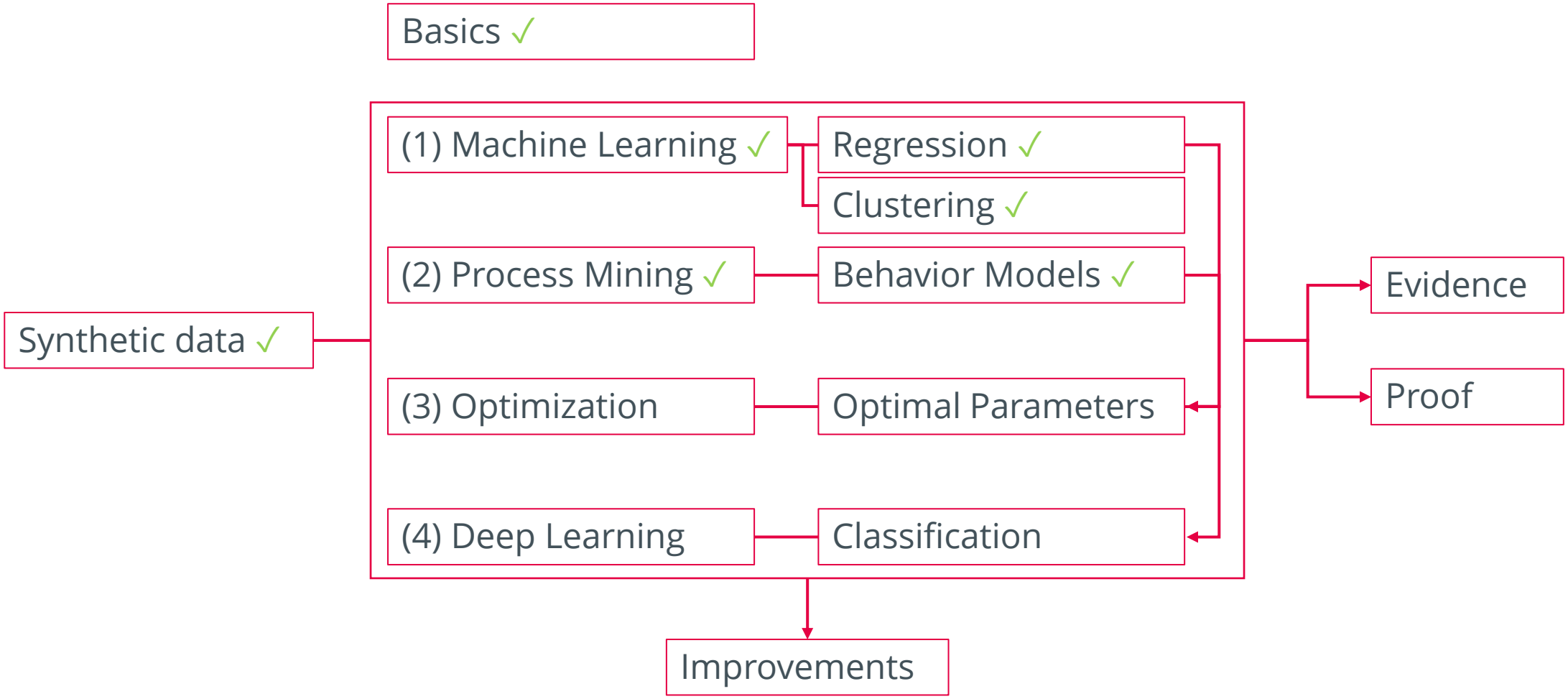
Normal



ARP CP



Summary: Advanced Topics in Algorithms



TH OWL

TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Thank you!