Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

data science

CAIDAS

Machine Learning for Time Series and Anomaly Detection

Anna Krause
Daniel Schlör

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Machine Learning for <u>**Time Series**</u> and Anomaly Detection

<u>Anna Krause</u>
Daniel Schlör

Machine Learning for Time Series and **Anomaly Detection**

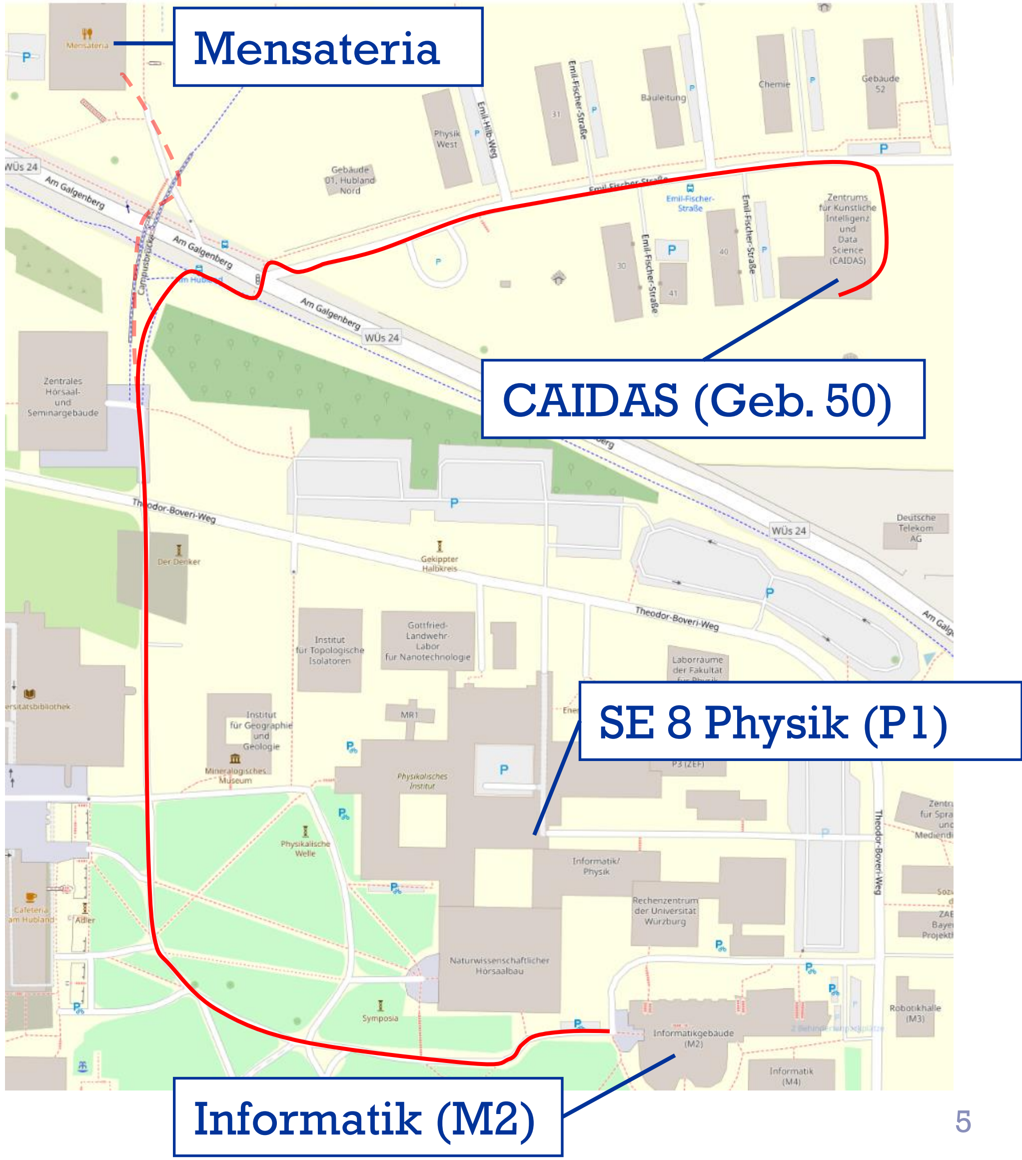
Anna Krause
Daniel Schlör

Organizational Matters

Lecture Times and Location

- Lectures will take place on **Wednesday, 12-14h**
- Exercises will take place on **Thursday, 16-18h**

in Seminarraum 3, **CAIDAS Building (Geb. 50), Hubland Nord**



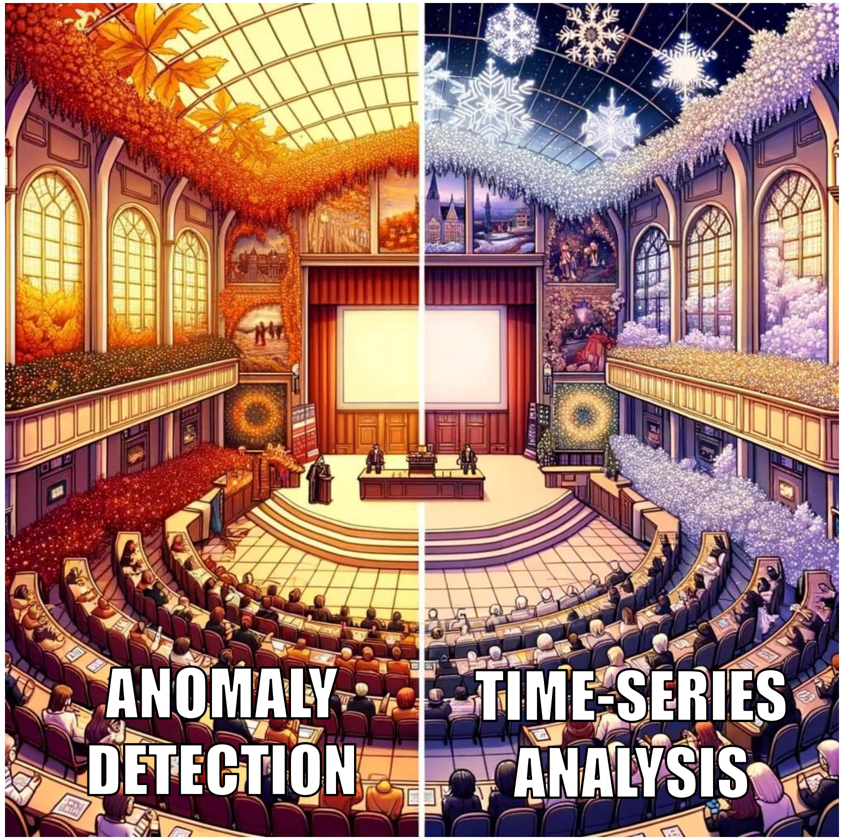
Lecture Format

- Two independent topics
  - ~7 lectures on Anomaly Detection
  - ~7 lectures on Time-Series Analysis
- Rough plan (subject to change):
  - Now till end of November: Anomaly Detection

14.10. - 20.10.2024	Lecture
21.10. - 27.10.2024	Lecture + Exercise
28.10. - 03.11.2024	Lecture + Exercise
04.11. - 10.11.2024	Lecture + Exercise
11.11. - 17.11.2024	Lecture + Exercise
18.11. - 24.11.2024	Lecture + Exercise
25.11. - 01.12.2024	Lecture + Exercise

… [Time-Series part] …

  - December till end of January: Time-Series Analysis
  - 05.02.2024 Exam
  - Exam will cover both topics, AD and TS!



Exercises Format

- Each exercise (AD / TS) will be split into two parts
  - Specific tasks to practice lecture content
    - (Preparation for exam)
  - Project to explore concepts from lecture in practice
    - (Gaining hands-on experience)
- First lectures (per block) more focused on specific tasks
- Later lectures more focused working on the project
- Exercise in teams! 2-3 students per team
- In the exercise session: Solving tasks / working on projects together

Exercises Format (cont.)

- You can earn a grade bonus (one step if passed, e. g. 1.3 => 1.0) if you prepare and present your tasks and project work
- Specific exercise tasks: Report at the end / beginning of next session your results as group
- Project: Project presentation at the end of each topic block
  - Tentative plan: 05.12 or 12.12 (exercise slot) for AD
23.01 or 30.01. (exercise slot) for TS
  - Topic: A small data-science experiment on AD / TS that you build in your team

Project Overview

1. Project Proposal
   - Your own idea for anomaly detection
   - Examples: Credit card fraud, malfunctioning sensors, network intrusion or suspicious activities in server logs, anomalous social media activity, etc.
2. Data Collection
3. Approach
4. Implementation
5. Evaluation
6. Presentation

Project Overview

1. Project Proposal
2. Data Collection
   - Gather or generate relevant data
   - Datasets from Kaggle, UCI, etc.
3. Approach
4. Implementation
5. Evaluation
6. Presentation

Project Overview

1. Project Proposal
2. Data Collection
3. Approach
   - Search for related work
   - Own idea for a new approach or adaptation of approach from lecture
   - Scikit-learn, PyOD, PyTorch, etc.
4. Implementation
5. Evaluation
6. Presentation

Project Overview

1. Project Proposal
2. Data Collection
3. Approach
4. Implementation
   - Implementation of the approach
   - Preprocessing
   - Feature extraction
   - Baselines
5. Evaluation
6. Presentation

Project Overview

1. Project Proposal
2. Data Collection
3. Approach
4. Implementation
5. Evaluation
   - Exploratory evaluation
   - Quantitative evaluation (metrics)
   - Comparison with baselines
6. Presentation

Project Overview

1. Project Proposal
2. Data Collection
3. Approach
4. Implementation
5. Evaluation
6. Presentation
   - Prepare slides
   - Introduce task, dataset and approach
   - Share findings and challenges

Modules this lecture can be credited

- 10-I=AKDS Ausgewählte Kapitel des Data Science (only newest PO)
- 10-I=AKIS Ausgewählte Kapitel der Intelligenten Systeme
- 10-I=AKII Ausgewählte Kapitel der Informatik

- Do not forget to register for the exam (the module to be credited) on WueStudy in time! (31.01.? check!)
- We have no ability to late-register and can not post grades without registration!

Opportunities @LSX

- Master Praktikum / Thesis on
  - AD / TS methodology
  - Application domains:
    - Network intrusion detection, fraud, audio, sensor data, text, etc.
  - Other topics (see website)
    - Natural Language Processing
    - Ecosystems and Climate Modeling
    - Publication Data
    - Product and Chat Recommendation
    - Physics Informed Deep Learning
    - Medical & Biological Data

https://www.informatik.uni-wuerzburg.de/datascience/

Agenda for Today

- Organizational matters ✓
- Pre-course quiz
- Data Science and Machine Learning Foundations
  - Data, Information, Knowledge
  - Definitions
  - ML categories and tasks
  - Terminology
  - Preprocessing

- https://www.menti.com/alq91fjk4iyj

Pre-course quiz

Data Science and Machine Learning Foundations

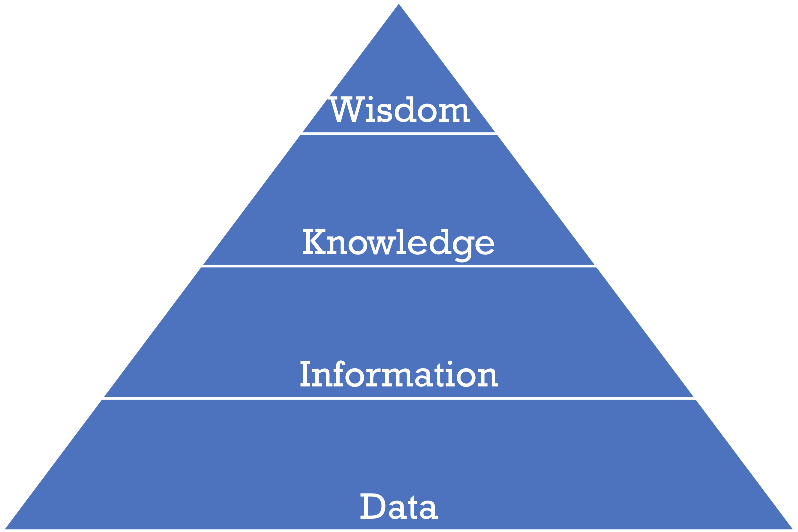
Data - Information - Knowledge

- Data
  Raw data (measurements, "facts")

- Information
  Significant, summarized data for a specific purpose

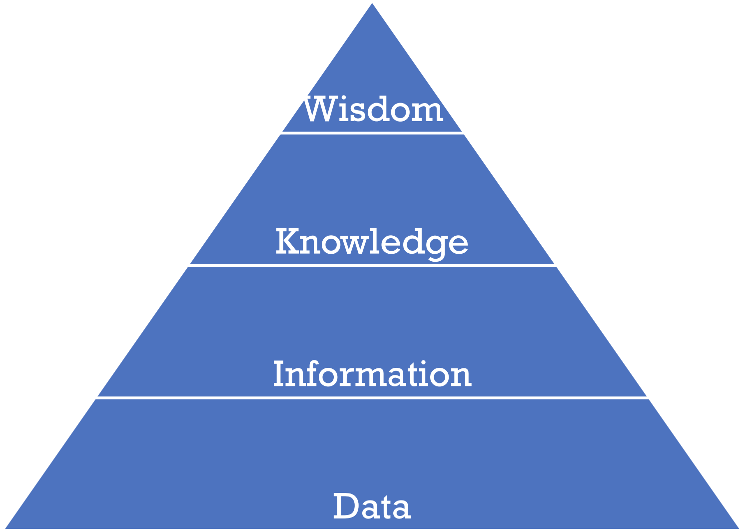
- Knowledge
  Knowledge that people are aware of

Be aware:
Many contradictory definitions exist



DIKW
Pyramid

Data - Information - Knowledge



DIKW Pyramid

There's a vast amount of data available over computer networks —far beyond megabytes, gigabytes, or terabytes, it's well into the petabyte region.

Yet how much common-sense information is this? Data isn't

information, any more than fifty tons of cement is a skyscraper. A string of bits might represent a draft treaty between two nations, a slice from a rock video, a thousand digits of π, or random noise. Data is just bits and bytes . . . grains of sand without a concrete aggregate. Information has utility. It has meaning.

Most important: information is not knowledge. Back to my central thesis: my computer can access the Swiss molecular-biology archive, yet I still know squat about DNA transcription. Everyone has access to quotes from the New York Stock Exchange, yet who can predict what'll happen tomorrow? And having the latest Jupiter images sure doesn't mean you understand planetary atmospheres. Professor Tomasko, my dissertation adviser, would single me out as living proof.

There's a relationship between data, information, knowledge, understanding, and wisdom.

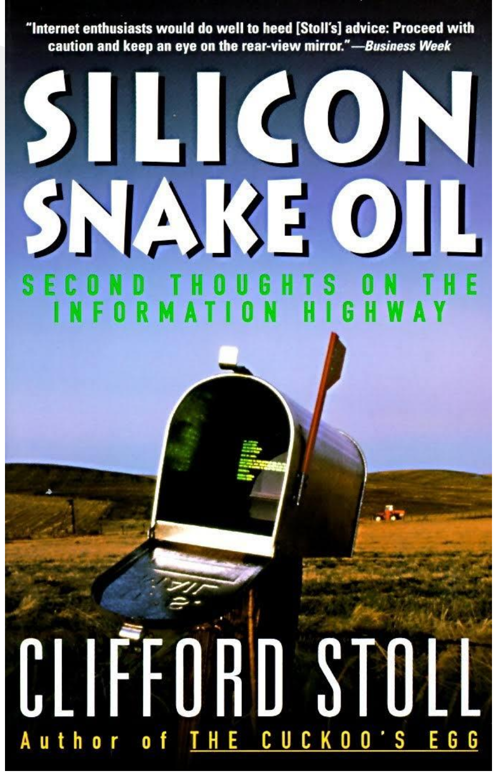
Our networks are awash in data. A little of it's information. A smidgen* of this shows up as knowledge. Combined with ideas, some of that is actually useful. Mix in experience, context, compassion, discipline, humor, tolerance, and humility, and perhaps knowledge becomes wisdom.

Minds think with ideas, not information. No amount of data, bandwidth, or processing power can substitute for inspired thought.

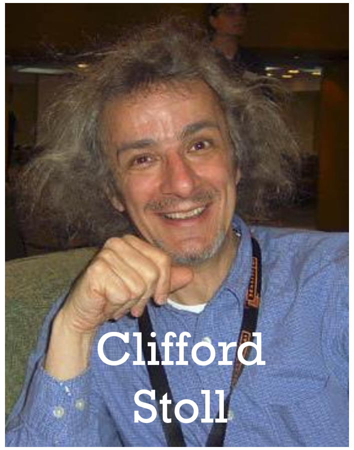
Dazzled by computers and communications theory, we've been misled into thinking that experience can be broken down into bits and bytes. High schools teach us that knowledge is power. The movie *The Net* transformed that cliché into "information is power." Today, those with the most information have the most power.

This is patently false. The powerful aren't informed. And who has the most information? Librarians. Hardly a powerful group.

The Internet, that great digital Dumpster, confers not power, not prosperity, not perspicacity.



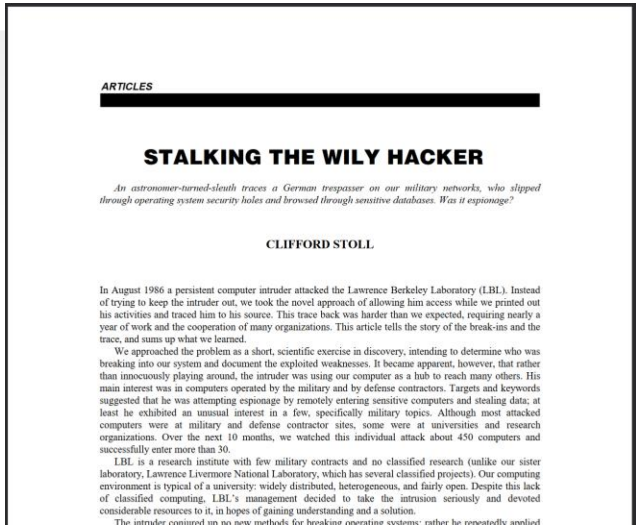
https://archive.org/details/isbn_9780785794943
1995



From https://www.pro-linux.de/artikel/2/241/interview-mit-clifford-stoll.html

Clifford Stoll

- In 1986, Cliff Stoll at Lawrence Berkeley National Labs was assigned to investigate a 75-cent accounting discrepancy in the lab's computer network.
- Stoll discovered the source of the anomaly was a hacker infiltrating the system.
- He spent a year tracing the hacker's activities, uncovering intrusions into military and government networks.
- The hacking was carried out by young German hackers working for the Soviet KGB.
- Stoll's investigation led to the first known case of state-sponsored hacking, which he detailed in his 1989 book The Cuckoo's Egg.



ARTICLES

STALKING THE WILY HACKER

An astronomer-turned-sleuth traces a German trespasser on our military networks, who slipped through operating system security holes and browsed through sensitive databases. Was it espionage?

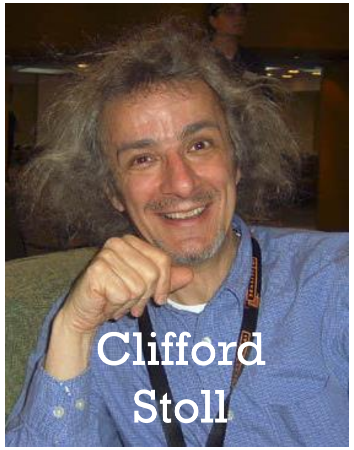
CLIFFORD STOLL

In August 1986 a persistent computer intruder attacked the Lawrence Berkeley Laboratory (LBL). Instead of trying to keep the intruder out, we took the novel approach of allowing him access while we printed out his activities and traced him to his source. This trace back was harder than we expected, requiring nearly a year of work and the cooperation of many organizations. This article tells the story of the break-ins and the trace, and sums up what we learned.

We approached the problem as a short, scientific exercise in discovery, intending to determine who was breaking into our system and document the exploited weaknesses. It became apparent, however, that rather than innocuously playing around, the intruder was using our computer as a hub to reach many others. His main interest was in computers operated by the military and by defense contractors. Targets and keywords suggested that he was attempting espionage by remotely entering sensitive computers and stealing data; at least he exhibited an unusual interest in a few, specifically military topics. Although most attacked computers were at military and defense contractor sites, some were at universities and research organizations. Over the next 10 months, we watched this individual attack about 450 computers and successfully enter more than 30.

LBL is a research institute with few military contracts and no classified research (unlike our sister laboratory, Lawrence Livermore National Laboratory, which has several classified projects). Our computing environment is typical of a university: widely distributed, heterogeneous, and fairly open. Despite this lack of classified computing, LBL's management decided to take the intrusion seriously and devoted considerable resources to it, in hopes of gaining understanding and a solution.

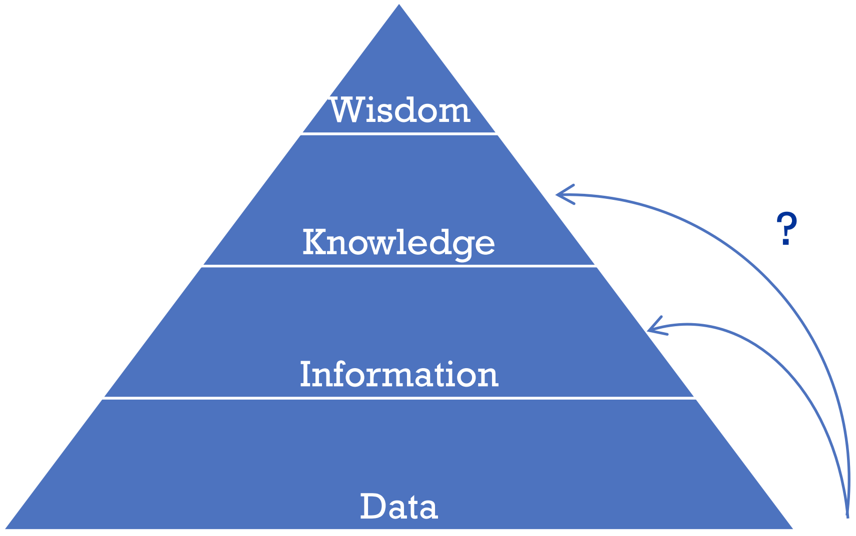
http://pdf.textfiles.com/academics/wilyhacker.pdf



From https://www.pro-linux.de/artikel/2/241/interview-mit-clifford-stoll.html

How to get from data to information, knowledge, wisdom?

- Knowledge Discovery in Databases (KDD)
- Data Mining
- Data Science



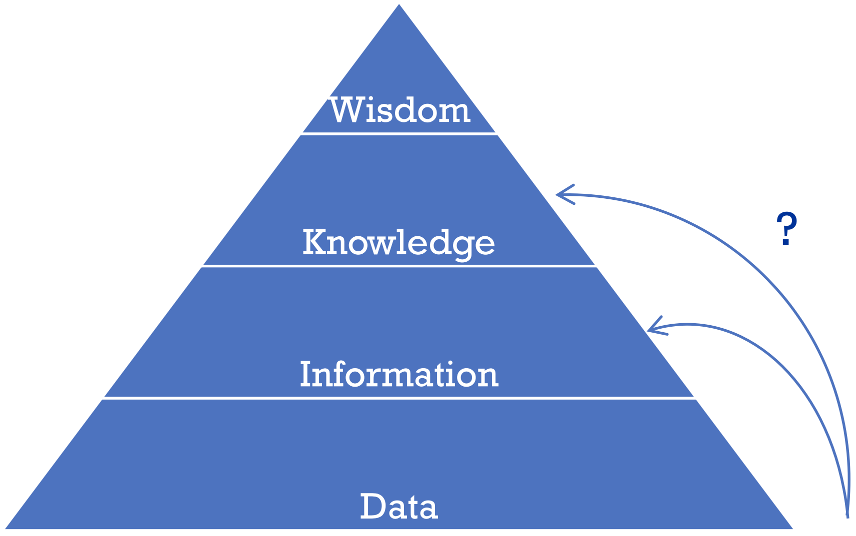
DIKW Pyramid

How to get from data to information, knowledge, wisdom?

- Knowledge Discovery in Databases (KDD)
    - Fayyad et al.* define KDD in 1996 as

*The nontrivial process of identifying **valid**, **novel**, potentially **useful**, and ultimately **understandable** patterns in data.*

- Data Mining
- Data Science



DIKW Pyramid

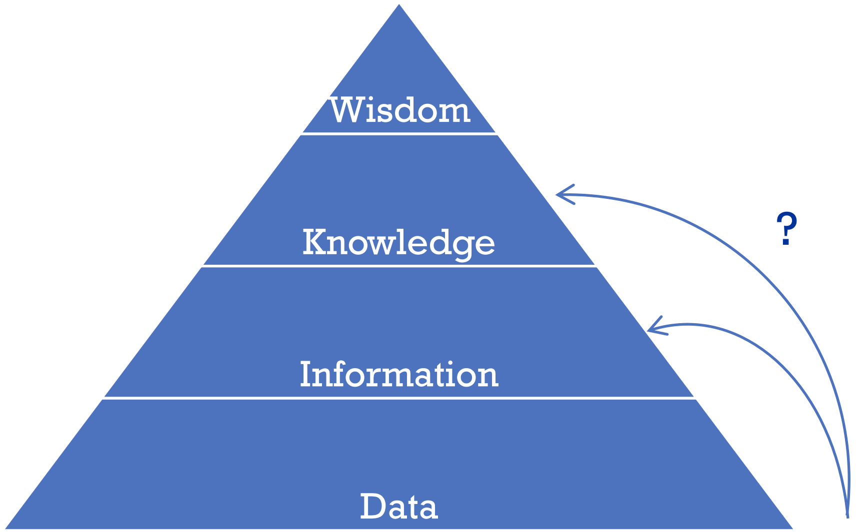
* The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, Vol. 39, No. 11, 1996

How to get from data to information, knowledge, wisdom?

- Knowledge Discovery in Databases (KDD)
- Data Mining
  - Aggarwal defines Data Mining in 2015 as

Data Mining is the study of ***collecting, cleaning, processing, analyzing, and gaining useful insights*** *from data. […] "Data mining" is a broad umbrella term that is used to describe these different aspects of data processing.*

- Data Science



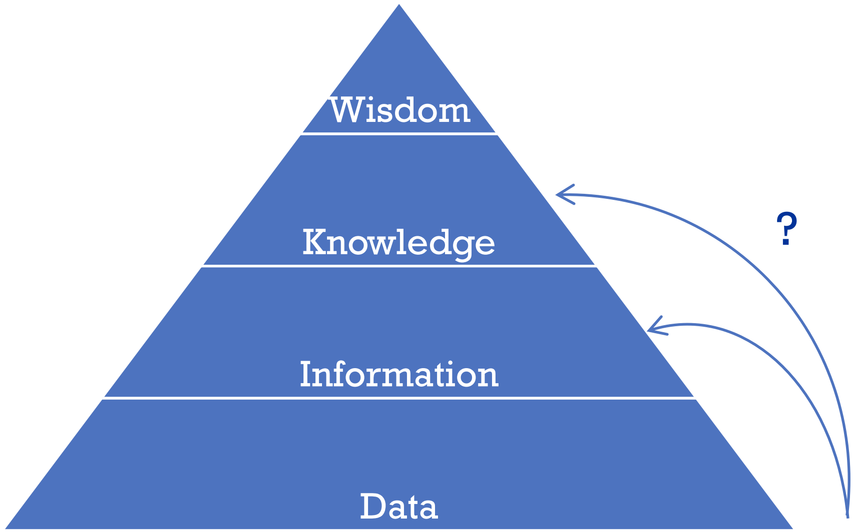
DIKW Pyramid

Aggarwal, Data Mining: The Textbook, Springer, 2015

How to get from data to information, knowledge, wisdom?

- Knowledge Discovery in Databases (KDD)
- Data Mining
- Data Science
  - Provost & Fawcett in 2013 connect both

"At a high level, data science is a set of ***fundamental principles that support and guide the principled extraction of information and knowledge from data****. Possibly the most closely related concept to data science is data mining - the actual extraction of knowledge from data via technologies that incorporate these principles. "*



DIKW
Pyramid

Provost, Foster, and Tom Fawcett. "Data science and its relationship to big data and data-driven decision making." *Big data* 1.1 (2013): 51-59.

Data Science

What do you imagine data science to involve?

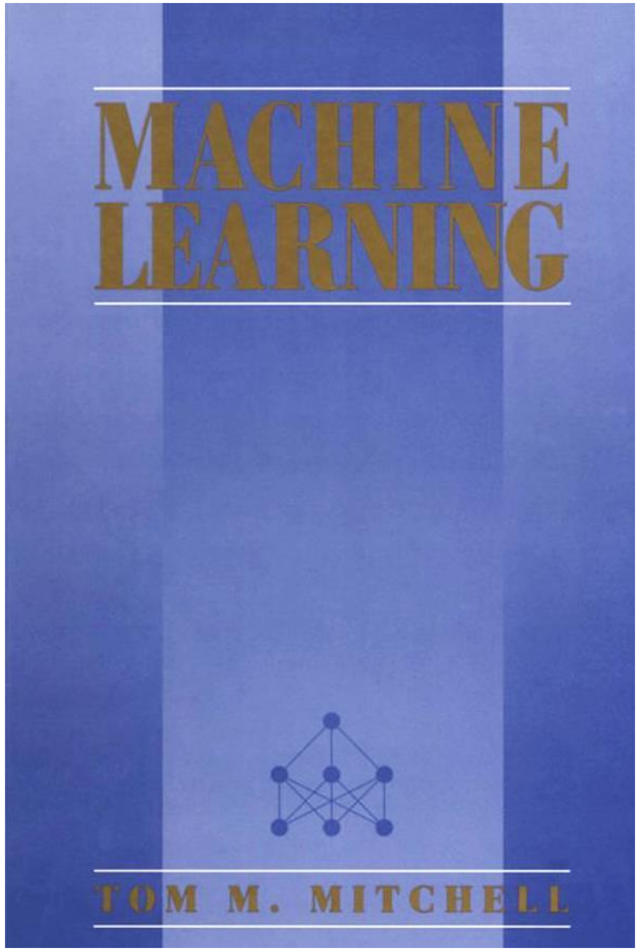
Data Science

Data Science

What is Machine Learning?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

— Tom Mitchell, 1997

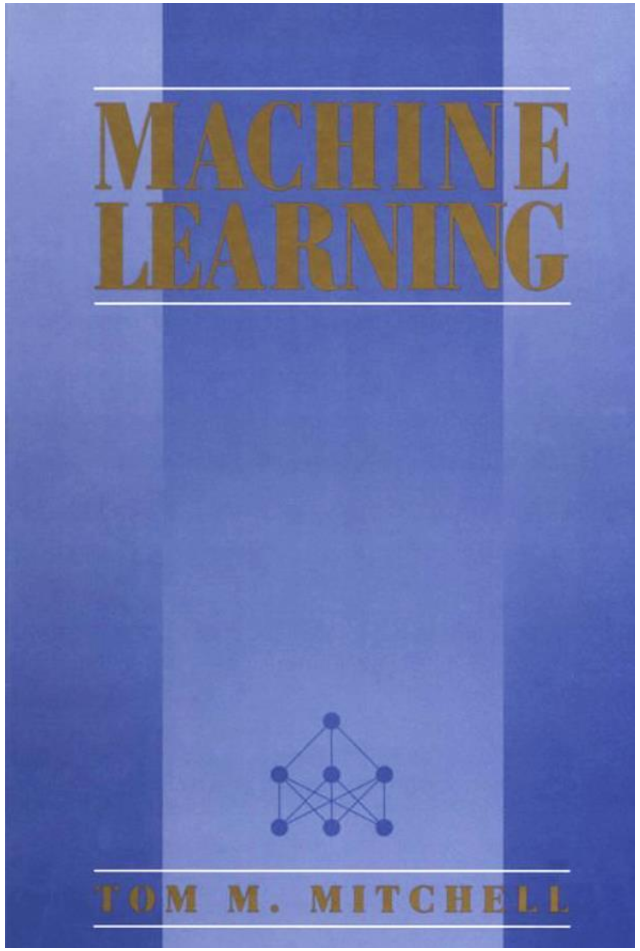


Machine Learning, Tom Mitchell, McGraw Hill, 1997

What is Machine Learning?

data “A computer program is said to learn from ~~experience E~~ with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with ~~experience E~~”

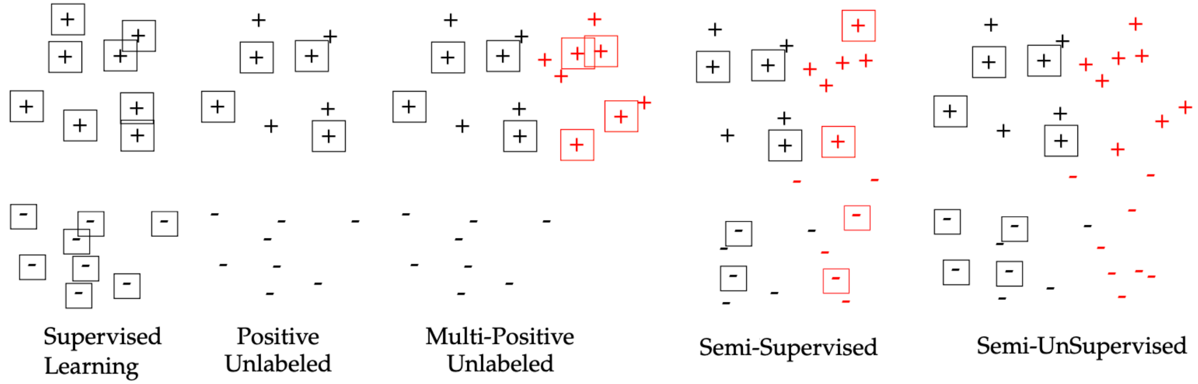
iterative training



Machine Learning, Tom Mitchell, McGraw Hill, 1997

ML approaches

- Main Categories:
  - Supervised ML
  - Unsupervised ML
  - Semi-Supervised ML
  - …



ML approaches

- Main Categories:
  - Supervised ML
  - Unsupervised ML
  - Semi-Supervised ML
  - …

- ML-Tasks
  - Classification
  - Regression
  - Clustering
  - Anomaly Detection
  - Synthesis und Sampling
  - …

ML approaches

- Main Categories:
  - Supervised ML
  - Unsupervised ML
  - Semi-Supervised ML
  - …

- ML-Tasks
  - Classification
  - Regression
  - Clustering
  - Anomaly Detection
  - Synthesis und Sampling

Can you come up with examples that you could address with each ML task?

Terminology

- Data, Datasets
- Data-point, Sample, Instance
- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

Terminology

- Data, Datasets
- Data-point, Sample, Instance
- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

- ML-Tasks

- Classification
- Regression
- Clustering
- Anomaly Detection
- Synthesis and Sampling
- …

Terminology

- Data, Datasets $\mathcal{D}$
- Data-point, Sample, Instance $\boldsymbol{x}_i$
- Features, Attributes $\boldsymbol{X}_j$
- Label, Target, Class $\boldsymbol{Y}, \boldsymbol{y}_i$
- Model f_Θ
- Prediction $\hat{\boldsymbol{y}}_i$
- Objective, Loss, Criterion $\mathcal{L}$
- Training
- Metric, Measure, Evaluation Metrics

Terminology

$\mathcal{D}$

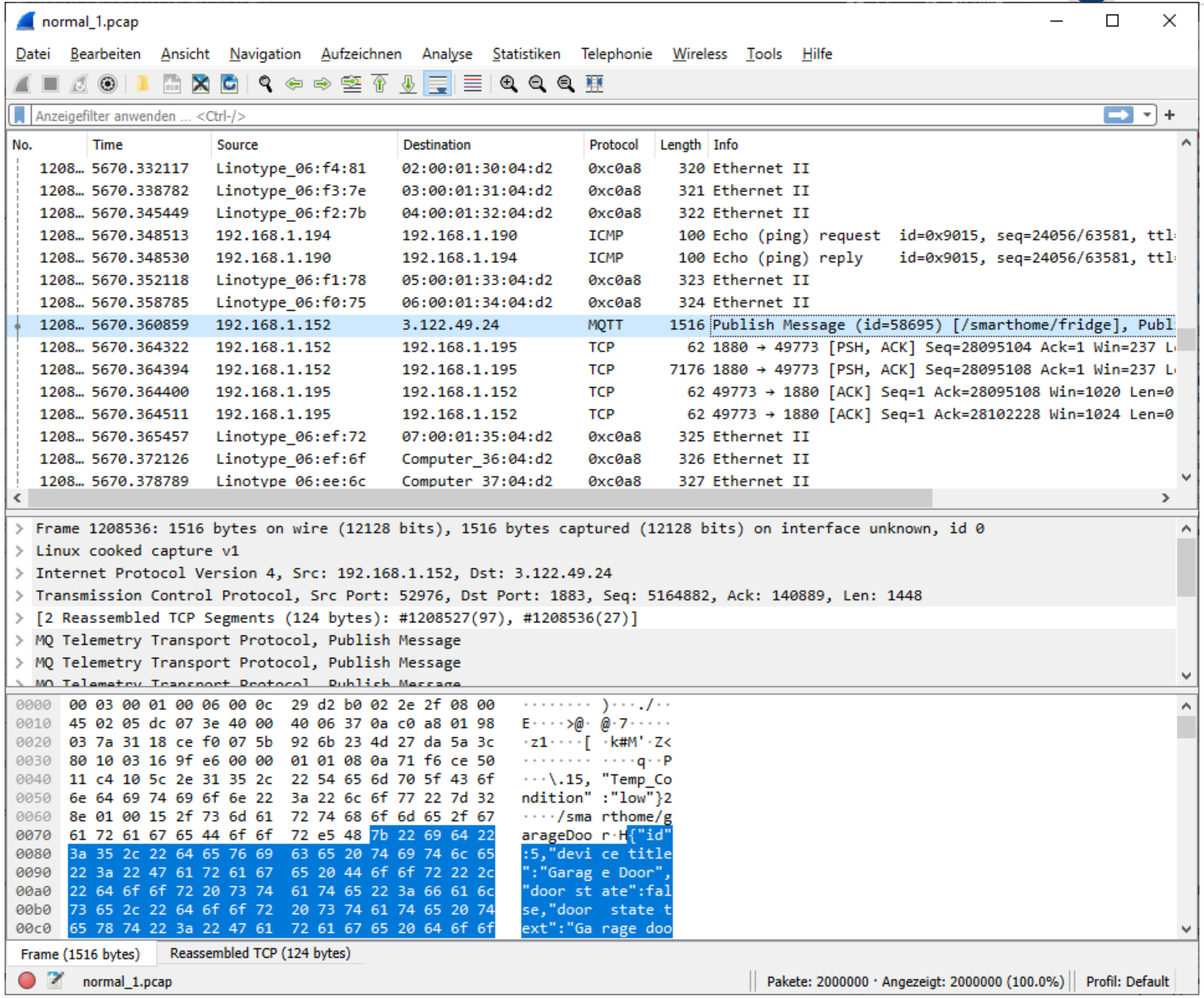
- **Data, Datasets**
- Data-point, Sample, Instance
- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

```
[root@server log]# ls
anaconda           cron-20240602          dnf.rpm.log         maillog-20240519   secure-20240519
audit              cron-20240609          dnf.rpm.log.1       maillog-20240526   secure-20240526
boot.log           dnf.librepo.log        dnf.rpm.log.2       maillog-20240602   secure-20240602
boot.log-20220114  dnf.librepo.log.1      dnf.rpm.log.3       maillog-20240609   secure-20240609
boot.log-20220614  dnf.librepo.log.2      fail2ban.log        messages           spooler
boot.log-20220625  dnf.librepo.log-20210829  docker-clean     messages-20240519  spooler-20240519
boot.log-20220902  dnf.librepo.log-20210905  firewalld        messages-20240526  spooler-20240526
boot.log-20230322  dnf.librepo.log-20210912  hawkey.log       messages-20240602  spooler-20240602
boot.log-20230429  dnf.librepo.log-20210919  hawkey.log-20240519  messages-20240609  spooler-20240609
boot.log-20230623  dnf.librepo.log.3      hawkey.log-20240526  migrate2rocky.log  sssd
btmp               dnf.librepo.log.4      hawkey.log-20240602  private          tuned
btmp-20240601      dnf.log                hawkey.log-20240609  puppetlabs       wtmp
chrony             dnf.log.1              journal             qemu-ga            wtmp-20240609
cron               dnf.log.2              kdump.log           rhsm
cron-20240519      dnf.log.3              lastlog             samba
cron-20240526      dnf.log.4              maillog             secure
[root@server log]# tail secure
Jun 10 15:24:29 server sshd[685182]: pam_ldap(sshd:auth): unknown option: forward_pass
Jun 10 15:24:29 server sshd[685182]: Accepted password for myuser from 10.85.10.10 port 5046 ssh2
Jun 10 15:24:29 server sshd[685182]: pam_unix(sshd:session): session opened for user myuser by (uid=0)
Jun 10 15:25:03 server sudo[685449]: pam_ldap(sudo:auth): unknown option: forward_pass
Jun 10 15:25:06 server sudo[685449]: myuser: TTY=pts/45 ; PWD=/var/log ; USER=root ; COMMAND=/bin/su
Jun 10 15:25:06 server sudo[685449]: pam_systemd(sudo:session): Cannot create session: Already running in
Jun 10 15:25:06 server sudo[685449]: pam_unix(sudo:session): session opened for user root by myuser(uid=0)
Jun 10 15:25:06 server su[685452]: pam_systemd(su:session): Cannot create session: Already running in a se
Jun 10 15:25:06 server su[685452]: pam_unix(su:session): session opened for user root by myuser(uid=0)
Jun 10 15:25:10 server sshd[685478]: Connection closed by 10.187.14.15 port 57378 [preauth]
[root@server log]# grep Ban fail2ban.log
2024-06-09 00:46:08,405 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.68.104.130
2024-06-09 01:54:56,052 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.19.118.77
2024-06-09 03:44:20,579 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.33.213.28
2024-06-09 10:24:40,827 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.108.46.118
2024-06-09 19:37:43,656 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.124.163.235
2024-06-09 23:25:54,097 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.134.141.56
2024-06-10 07:09:42,580 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.68.68.30
2024-06-10 12:16:35,575 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.217.12.0
2024-06-10 14:28:01,120 fail2ban.actions        [17931]: NOTICE  [sshd] Ban 10.183.49.87
```

Terminology

$\mathcal{D}$

- **Data, Datasets**
- Data-point, Sample, Instance
- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics



Ter

```
import pandas as pd
```

[6] ✓ 0.0s Python

```
# load csv files into pandas DataFrame
data = pd.read_csv('NF-ToN-IoT.csv')
data
```

[7] ✓ 1.4s Python

	IPV4_SRC_ADDR	L4_SRC_PORT	IPV4_DST_ADDR	L4_DST_PORT	...	TCP_FLAGS	FLOW_DURATION_MILLISECONDS	Label	Attack
0	192.168.1.195	63318	52.139.250.253	443	...	24	327	0	Benign
1	192.168.1.79	57442	192.168.1.255	15600	...	0	0	0	Benign
2	192.168.1.79	57452	239.255.255.250	15600	...	0	0	0	Benign
3	192.168.1.193	138	192.168.1.255	138	...	0	0	0	Benign
4	192.168.1.79	51989	192.168.1.255	15600	...	0	0	0	Benign
...	...	...	...	...	...	...	...	...	...
1379269	192.168.1.31	58032	192.168.1.194	80	...	18	9433	1	ddos
1379270	192.168.1.31	58034	192.168.1.194	80	...	18	9221	1	ddos
1379271	192.168.1.31	58036	192.168.1.194	80	...	18	9656	1	ddos
1379272	192.168.1.31	58038	192.168.1.194	80	...	18	10046	1	ddos
1379273	192.168.1.31	58040	192.168.1.194	80	...	18	10485	1	ddos

1379274 rows × 14 columns

$\mathcal{D}$

- **Data, Datasets**
- Data-point, Sample, Instance
- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Ter

```python
import pandas as pd
```

[6] ✓ 0.0s Python

```python
# load csv files into pandas DataFrame
data = pd.read_csv('NF-ToN-IoT.csv')
data
```

[7] ✓ 1.4s Python

	IPV4_SRC_ADDR	L4_SRC_PORT	IPV4_DST_ADDR	L4_DST_PORT	...	TCP_FLAGS	FLOW_DURATION_MILLISECONDS	Label	Attack
0	192.168.1.195	63318	52.139.250.253	443	...	24	327	0	Benign
1	192.168.1.79	57442	192.168.1.255	15600	...	0	0	0	Benign
2	192.168.1.79	57452	239.255.255.250	15600	...	0	0	0	Benign
3	192.168.1.193	138	192.168.1.255	138	...	0	0	0	Benign
4	192.168.1.79	51989	192.168.1.255	15600	...	0	0	0	Benign
...	...	...	...	...	...	...	...	...	...
1379269	192.168.1.31	58032	192.168.1.194	80	...	18	9433	1	ddos
1379270	192.168.1.31	58034	192.168.1.194	80	...	18	9221	1	ddos
1379271	192.168.1.31	58036	192.168.1.194	80	...	18	9656	1	ddos
1379272	192.168.1.31	58038	192.168.1.194	80	...	18	10046	1	ddos
1379273	192.168.1.31	58040	192.168.1.194	80	...	18	10485	1	ddos

1379274 rows × 14 columns

```python
# show sample 1379271
data.iloc[1379271]
```

[8] ✓ 0.0s Python

```
IPV4_SRC_ADDR                  192.168.1.31
L4_SRC_PORT                           58036
IPV4_DST_ADDR                 192.168.1.194
L4_DST_PORT                              80
PROTOCOL                                  6
L7_PROTO                                7.0
IN_BYTES                                216
OUT_BYTES                               180
IN_PKTS                                   4
OUT_PKTS                                  3
TCP_FLAGS                                18
FLOW_DURATION_MILLISECONDS             9656
Label                                     1
Attack                                 ddos
Name: 1379271, dtype: object
```

$\mathcal{D}$ • Data, Datasets

x_i • **Data-point, Sample, Instance**

- Features, Attributes
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

Terminology

- $\mathcal{D}$ • Data, Datasets
- x_i • Data-point, Sample, Instance
- X_j • **Features, Attributes**
- Label, Target, Class
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

```python
import pandas as pd
```
[2] ✓ 0.0s Python

```python
# load csv files into pandas DataFrame
data = pd.read_csv('NF-ToN-IoT.csv')
data
```
[3] ✓ 1.4s Python

	IPV4_SRC_ADDR	L4_SRC_PORT	IPV4_DST_ADDR	L4_DST_PORT	...	TCP_FLAGS	FLOW_DURATION_MILLISECONDS	Label	Attack
0	192.168.1.195	63318	52.139.250.253	443	...	24	327	0	Benign
1	192.168.1.79	57442	192.168.1.255	15600	...	0	0	0	Benign
2	192.168.1.79	57452	239.255.255.250	15600	...	0	0	0	Benign
3	192.168.1.193	138	192.168.1.255	138	...	0	0	0	Benign
4	192.168.1.79	51989	192.168.1.255	15600	...	0	0	0	Benign
...	...	...	...	...	...	...	...	...	...
1379269	192.168.1.31	58032	192.168.1.194	80	...	18	9433	1	ddos
1379270	192.168.1.31	58034	192.168.1.194	80	...	18	9221	1	ddos
1379271	192.168.1.31	58036	192.168.1.194	80	...	18	9656	1	ddos
1379272	192.168.1.31	58038	192.168.1.194	80	...	18	10046	1	ddos
1379273	192.168.1.31	58040	192.168.1.194	80	...	18	10485	1	ddos

1379274 rows × 14 columns

```python
# list dataset freatures
data.columns
```
[4] ✓ 0.0s Python

```
Index(['IPV4_SRC_ADDR', 'L4_SRC_PORT', 'IPV4_DST_ADDR', 'L4_DST_PORT',
       'PROTOCOL', 'L7_PROTO', 'IN_BYTES', 'OUT_BYTES', 'IN_PKTS', 'OUT_PKTS',
       'TCP_FLAGS', 'FLOW_DURATION_MILLISECONDS', 'Label', 'Attack'],
      dtype='object')
```

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Terminology

- $\mathcal{D}$ • Data, Datasets
- x_i • Data-point, Sample, Instance
- X_j • Features, Attributes
- Y, y_i • **Label, Target, Class**
- Model
- Prediction
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

```python
import pandas as pd
```

[2] ✓ 0.0s — Python

```python
# load csv files into pandas DataFrame
data = pd.read_csv('NF-ToN-IoT.csv')
data
```

[3] ✓ 1.4s — Python

	IPV4_SRC_ADDR	L4_SRC_PORT	IPV4_DST_ADDR	L4_DST_PORT	...	TCP_FLAGS	FLOW_DURATION_MILLISECONDS	Label	Attack
0	192.168.1.195	63318	52.139.250.253	443	...	24	327	0	Benign
1	192.168.1.79	57442	192.168.1.255	15600	...	0	0	0	Benign
2	192.168.1.79	57452	239.255.255.250	15600	...	0	0	0	Benign
3	192.168.1.193	138	192.168.1.255	138	...	0	0	0	Benign
4	192.168.1.79	51989	192.168.1.255	15600	...	0	0	0	Benign
...	...	...	...	...	...	...	...	...	...
1379269	192.168.1.31	58032	192.168.1.194	80	...	18	9433	1	ddos
1379270	192.168.1.31	58034	192.168.1.194	80	...	18	9221	1	ddos
1379271	192.168.1.31	58036	192.168.1.194	80	...	18	9656	1	ddos
1379272	192.168.1.31	58038	192.168.1.194	80	...	18	10046	1	ddos
1379273	192.168.1.31	58040	192.168.1.194	80	...	18	10485	1	ddos

1379274 rows × 14 columns

```python
# list dataset freatures
data.columns
```

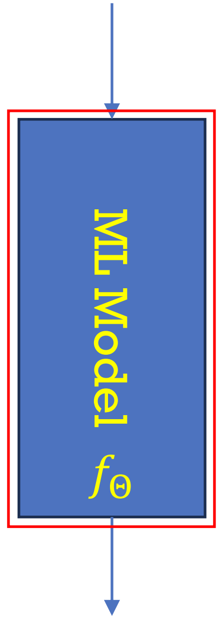
[4] ✓ 0.0s — Python

```
Index(['IPV4_SRC_ADDR', 'L4_SRC_PORT', 'IPV4_DST_ADDR', 'L4_DST_PORT',
       'PROTOCOL', 'L7_PROTO', 'IN_BYTES', 'OUT_BYTES', 'IN_PKTS', 'OUT_PKTS',
       'TCP_FLAGS', 'FLOW_DURATION_MILLISECONDS', 'Label', 'Attack'],
      dtype='object')
```

Terminology

- $\mathcal{D}$ • Data, Datasets
- $\boldsymbol{x_i}$ • Data-point, Sample, Instance
- $\boldsymbol{X_j}$ • Features, Attributes
- $\boldsymbol{Y}, \boldsymbol{y_i}$ • Label, Target, Class
- f_Θ • **Model**
- $\hat{y}_i$ • **Prediction**
- Objective, Loss, Criterion
- Training
- Metric, Measure, Evaluation Metrics

$\boldsymbol{x_i} = (\underbrace{x_{i,1}}_{\in \boldsymbol{X_1}}, \dots, \underbrace{x_{i,d}}_{\in \boldsymbol{X_d}}) \in \mathcal{D}$



$f_\Theta : \mathcal{D} \to Y$

$f_\Theta(x) = \hat{y}$

$\boxed{\hat{y}_i} \in \boldsymbol{Y}$

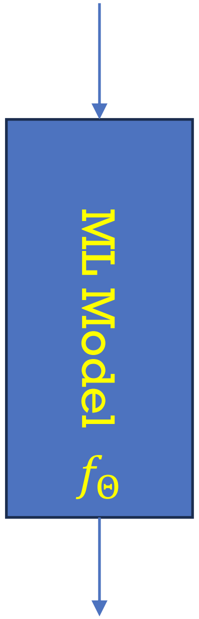
$\overset{?}{=}$

$y_i \in \boldsymbol{Y}$

Terminology

- $\mathcal{D}$ • Data, Datasets
- x_i • Data-point, Sample, Instance
- X_j • Features, Attributes
- Y, y_i • Label, Target, Class
- f_Θ • Model
- $\hat{y}_i$ • Prediction
- $\mathcal{L}$ • **Objective, Loss, Criterion**
- Training
- Metric, Measure, Evaluation Metrics

$$x_i = (\underset{\in X_1}{x_{i,1}}, \dots, \underset{\in X_d}{x_{i,d}}) \in \mathcal{D}$$



$$f_\Theta : \mathcal{D} \to Y$$

$$f_\Theta(x) = \hat{y}$$

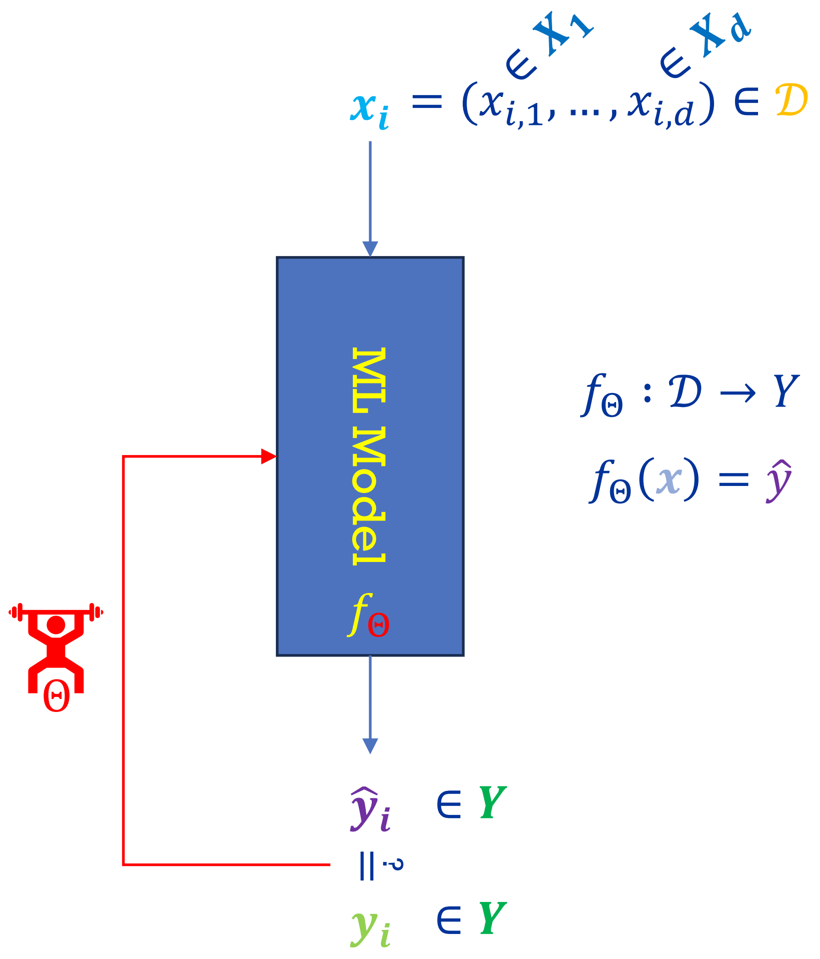
$$\hat{y}_i \in Y$$

$$\boxed{\overset{?}{=}}$$

$$y_i \in Y$$

Terminology

- $\mathcal{D}$ • Data, Datasets
- x_i • Data-point, Sample, Instance
- X_j • Features, Attributes
- Y, y_i • Label, Target, Class
- f_Θ • Model
- $\hat{y}_i$ • Prediction
- $\mathcal{L}$ • Objective, Loss, Criterion
- • **Training**
- • Metric, Measure, Evaluation Metrics



Terminology

- $\mathcal{D}$ • Data, Datasets
- x_i • Data-point, Sample, Instance
- X_j • Features, Attributes
- Y, y_i • Label, Target, Class
- f_Θ • Model
- $\hat{y}_i$ • Prediction
- $\mathcal{L}$ • Objective, Loss, Criterion
- Training
- **Metric, Measure, Evaluation Metrics**

$$x_i = (\overset{\in X_1}{x_{i,1}}, \dots, \overset{\in X_d}{x_{i,d}}) \in \mathcal{D}$$

ML Model f_Θ

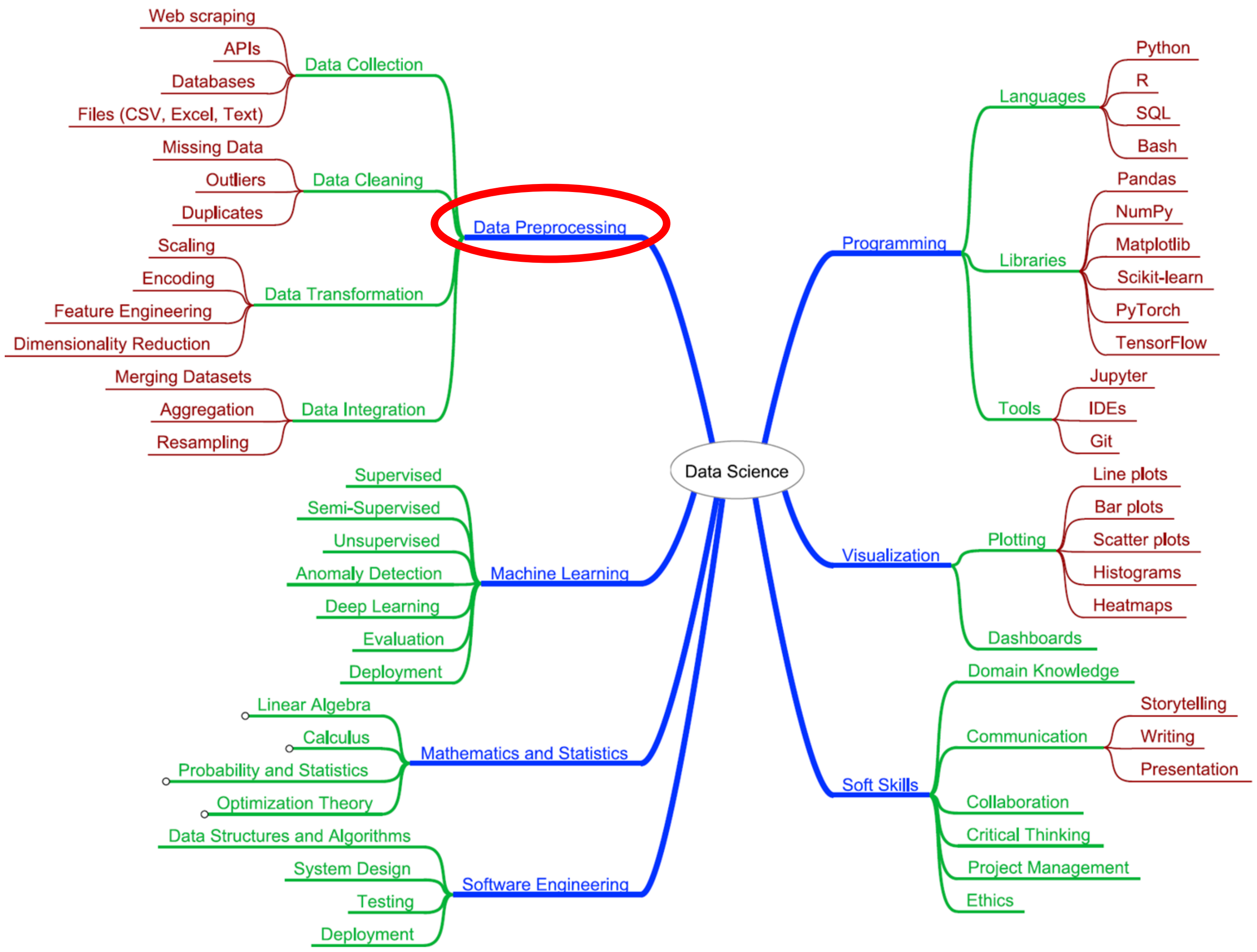
$$f_\Theta : \mathcal{D} \to Y$$

$$f_\Theta(x) = \hat{y}$$

$$\boxed{\frac{1}{\mathcal{D}} \sum_{i=0}^{|\mathcal{D}|} \begin{array}{c} \hat{y}_i \in Y \\ \stackrel{?}{=} \\ y_i \in Y \end{array}}$$

Accuracy

Data Science



Wrap up

- Data, Information, Knowledge
- Data Science
- Machine Learning
- Terminology

Data Preprocessing

- Selecting the data to be used
  - Creation of a data table
  - Reduction of the amount of data, e.g. through sampling
- Data cleaning
  - Data consistency
  - Removing incorrect values / instances
  - Missing values
  - Duplicates (redundant features)
- Data transformation
  - Discretization
  - Normalization
  - Feature Selection
  - Feature Extraction

Data Preprocessing - Example

ID	Name	Color	Quality control necessary?	Production Time [sec]	Production Frequency [1/h]
I1	Product1	Red	No	10	360
I2	Product2	Green	Yes	120	30
I3	Product3	Green	Yes	30	120
I4	Product4	Blue	No	90	40
I5	Product5	Red	Yes	60	60
…	…	…	…	…	…

Data Preprocessing - Propositionalization

- Data Science methods typically use a single table with
  - Rows: Cases, Propositions (instances)
  - Columns: Properties (features)

- To achieve a single table, we have to perform propositionalization
  - Transformation of a relational database into a propositional dataset (single table)
  - Features are usually aggregated (average, min, max, existance, etc.)
  - The proportionalization is typically performed by the user (domain knowledge necessary)

Data Preprocessing - Propositionalization

ID	Name	Color	Quality control necessary?	Production Time [sec]	Production Frequency [1/h]
I1	Product 1	Red	No	10	360
I2	Product 2	Green	Yes	120	30
…	…	…	…	…	…

ID	Product ID	Raw Material	Price [€/unit]
R1	I1	Material1	0.05
R2	I1	Material2	1
R3	I2	Material2	10
R4	I2	Material3	5
R5	I2	Material4	25
…		…	.

ID	Name	Color	Quality control necessary?	Production Time [sec]	Production Frequency [1/h]	Number of raw materials	Raw Material Cost [€]
I1	Product1	Red	No	10	360	2	1.05
I2	Product2	Green	Yes	120	30	3	40
…	…	…	…	…	…	…	…

Data Preprocessing – What is wrong?

ID	Name	Color	Quality control necessary?	Production Time [sec]	Production Frequency [1/h]
I1	Product1	Red	No	10	360
I2	Product2	G	Yes	120	120
I3	Product3	Green	Yes	30	120
I3	Product3	Green	Yes	30	120
I4	Product4	Blue		90	40
I5	Product5	Red	Yes	-10	-360
…	…	…	…	…	…

Data Preprocessing – What is wrong?

ID	Name	Color	Quality control necessary?	Production Time [sec]	Production Frequency
I1	Product1	Red	No	10	360
I2	Product2	G	Yes	120	120
I3	Product3	Green	Yes	30	120
I3	[illegible]	Green	Yes	30	[illegible]
[illegible]	[illegible]	Blue		90	40
[illegible]	Product5	Red	Yes	-10	-360
...	...	...	...	...	...

Inconsistency *

Incorrectly written value

Out of range

Duplicate

Missing Value

* If 120 sec. are necessary to produce the product, only 30 products can be produced per hour

Data Preprocessing – Erroneous Values

- Typical errors:
  - Missing values
  - Duplicates
  - Values are outside of a specified range
  - Incorrectly written feature values (especially for strings)
  - Inconsistency (values are mathematically, physically, etc. impossible)
  - Redundancy (features can be constructed / calculated by other features)

- Possible Solutions
  - Removing
    - How much information is removed?
    - Do we remove the samples or the feature?
  - Correcting
    - Is it possible to correct the erroneous values?
    - Which values do we insert?

Data Preprocessing – Erroneous Values

- Erroneous / Missing values may be corrected by
  - Inserting a default value
  - Inserting the most common value (for categorical data)
  - Inserting the average value (for continuous data)
  - Inserting the prediction of an already fitted model
  - Using the error value as is
    - Can conclusions be drawn from the absence of the value?

Data Preprocessing - Data Consistency

- Syntactic errors in input files
  For example:
  - Values containing commas in a comma-separated file format
  - German vs. English decimal separator (comma vs. dot), …
- Consistent unit for a concept (gram vs. kilogram vs. ton)
- Same concepts that were recorded with different names
- Different concepts, which were recorded with the same name

Data Preprocessing - Outlier Detection

- Outlier = instance that is “far away” from other instances (regarding one or more features)

- An error or important information?
  - If the outlier is actually erroneous, the instance must be removed

- Possible method for outlier detection:
  - Apply clustering methods
  - Find instances that are difficult to sort into clusters
  - Apply **anomaly detection** methods

Data Preprocessing - Discretization

- Some data science methods require ordinal values, but data is often numerical
- Discretization describes the conversion of numerical feature into ordinal
- Interval in old feature corresponds to one value in new feature

ID	Raw Material Cost [€/unit]
I1	1.05
I2	42.0
I3	0.20
I4	60.0
I5	25.0
…	…

>50: High
10-50: Medium
<10: Low

ID	Raw Material Cost [€/unit]
I1	Low
I2	Medium
I3	Low
I4	High
I5	Medium
…	…

- The intervals can be selected either by hand (domain knowledge) or automatically (see next slides)

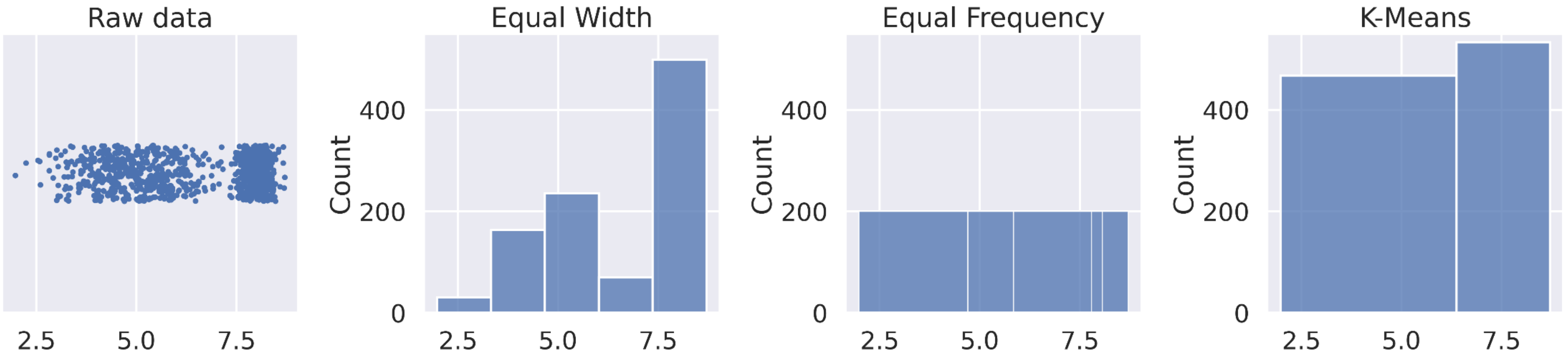
Data Preprocessing - Automatic Discretization

- Equal-Width Discretization
  All intervals are the same size

- Equal-Frequency Discretization
  All intervals contain the same number of instances

- Discretization by Clustering
  The intervals are determined by a clustering

Data Preprocessing - Automatic Discretization



- Equal Width
  - Simple method but generates imbalanced bins
- Equal Frequency
  - Ensures equal number of samples per bin but bin edges are not well interpretable
- Clustering
  - Bins are generated by a clustering method which reflects the structure of the data

Data Preprocessing - Automatic Discretization

- Alternative: Supervised Discretization
  Include another (e.g. binary) feature in the discretization!
  If possible (in the binary case), an interval should contain "only positive" or "only negative" examples
  - Top-down:
    - Start with an interval
    - Successively divide into parts that belong to (if possible) the same class
  - Bottom-up:
    - Start with each value as a single interval
    - Combine intervals with similar class distribution
  - Objective for "class homogeneity" e.g.:
    - Entropy
    - Statistical significance test (Chi^2 Test)

Data Preprocessing - Encoding

- Example: Feature “color” with values {red, green, blue}
- **Don't**: Introduction of “unnatural” structures (e.g. an order)

Color
Red
Green
Blue

⇨

Colo r
1
2
3

(Red < Green) ???

- **Do**: One boolean feature for each colour

Color
Red
Green
Blue

⇨

Red	Gree n	Blue
1	0	0
0	1	0
0	0	1

One-Hot Encoding

Data Preprocessing - Encoding

- Other approaches:
  - Learn *embeddings* as dense representation of fixed dimensionality
    - Classical (word2vec, GloVe, FastText, …)
    - (pretrained) LM based (BERT, RoBERTa, …)
    - Sentence embeddings (SentenceBERT, ELMo, InferSent, …)
  - Approaches from the Category Encoders library

```
pip install category_encoders
```

```
import category_encoders as ce

encoder = ce.BackwardDifferenceEncoder(cols=[...])
encoder = ce.BaseNEncoder(cols=[...])
encoder = ce.BinaryEncoder(cols=[...])
encoder = ce.CatBoostEncoder(cols=[...])
encoder = ce.CountEncoder(cols=[...])
encoder = ce.GLMMEncoder(cols=[...])
encoder = ce.GrayEncoder(cols=[...])
encoder = ce.HashingEncoder(cols=[...])
encoder = ce.HelmertEncoder(cols=[...])
encoder = ce.JamesSteinEncoder(cols=[...])
encoder = ce.LeaveOneOutEncoder(cols=[...])
encoder = ce.MEstimateEncoder(cols=[...])
encoder = ce.OneHotEncoder(cols=[...])
encoder = ce.OrdinalEncoder(cols=[...])
encoder = ce.PolynomialEncoder(cols=[...])
encoder = ce.QuantileEncoder(cols=[...])
encoder = ce.RankHotEncoder(cols=[...])
encoder = ce.SumEncoder(cols=[...])
encoder = ce.TargetEncoder(cols=[...])
encoder = ce.WOEEncoder(cols=[...])

encoder.fit(X, y)
X_cleaned = encoder.transform(X_dirty)
```

http://contrib.scikit-learn.org/category_encoders/index.html

Data Preprocessing - Feature Scaling

- Some (numerical) features have small value ranges (e.g.: 0.0 - 0.01), others large value ranges (e.g.: 0 - 100,000)
- The features should be normalized by a suitable method
  - Rescaling (Min-Max Normalization):

$$f(x) = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - Standardization (Z-score Normalization):

$$f(x) = \frac{x - \mu}{\sigma}$$

(With μ and σ being the mean and standard deviation of x for the dataset (train set))

Data Preprocessing - Instance Selection

- Some methods require the selection of random subsets
- Random sampling
  - (Random) selection of a subset of the data
- Stratified sampling
  - Increase the proportion of instances in the sample for the rare class compared to a random sample (especially for “imbalanced data")
  - Correlations between features should also be found in the random subsets

Data Preprocessing - Feature Selection

- Are features …
  - relevant? (the feature is related to the quantity of interest)
  - irrelevant? (the feature is not related to the quantity of interest)
  - redundant? (the feature can be replaced/constructed by other features)

- You may filter features that …
  - are anachronisms (features are unknown at the time of prediction)
  - are monotonically increasing (time, ID, …)
  - only have few non-default values
  - have many different values (e.g. number of samples = number of values)

- There exist a vast amount of heuristic methods for finding the “optimal” subsets of features (2n possible subsets exist!), e.g.
  - Filtering (remove features with low scores according to a suitable measure)
  - Sequential Forward/Backward Selection
  - Embedded Feature Selection (e.g. L1 Regularization)
  - Genetic algorithms

Data Preprocessing - Feature Extraction

- Creating new features from given features
  - e.g. transform the postal code to geographical coordinates (longitude & latitude)

- Combining features by any mathematical mapping, e.g.

$$x_{new} = 5 \cdot x_0^2 + e^{x_1} - 2 \cdot \sin x_2$$

- Feature Extraction often uses background knowledge
  $\Rightarrow$ Linking with further datasets ("cross-domain mining")

Wrap up

- Data, Information, Knowledge
- Data Science
- Machine Learning
- Terminology

Wrap up

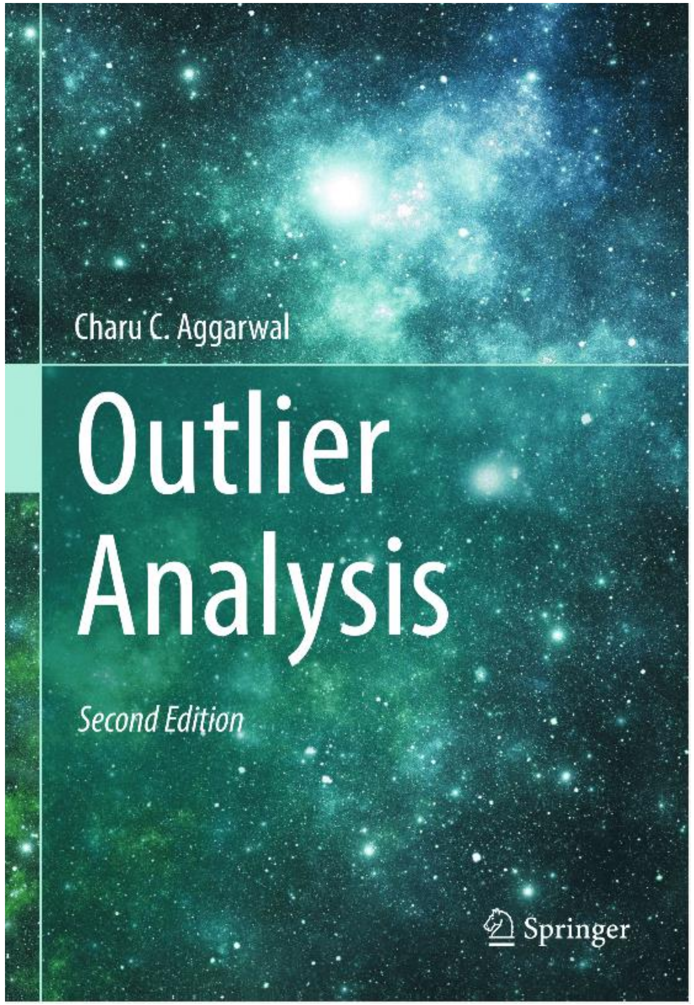
- Data, Information, Knowledge
- Data Science
- Machine Learning
- Terminology

- Data Preprocessing
  - Propositionalization
  - Errors and Data consistency
  - Discretization
  - Encoding
  - Feature Scaling
  - Instance Selection
  - Feature Selection
  - Feature Extraction

Text Book

https://link.springer.com/book/10.1007/978-3-319-47578-3

- Parts of the lecture are based on: Outlier Analysis by Aggarwal



Aggarwal, Charu C., and Charu C. Aggarwal. An introduction to outlier analysis. Springer International Publishing, 2017.