Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

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

Anna Krause
Daniel Schlör

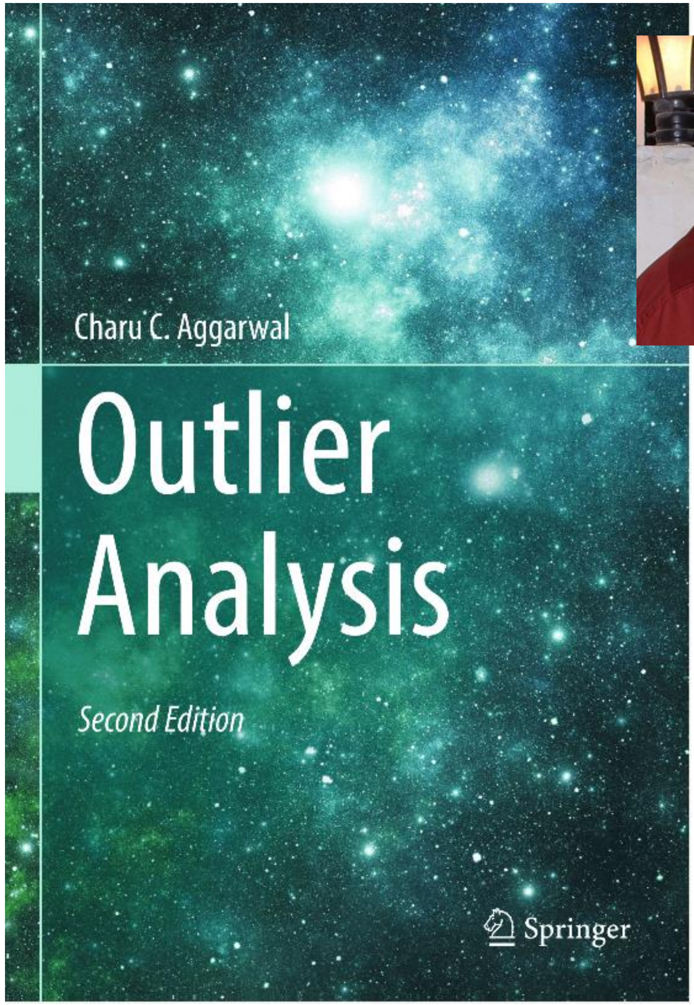
AD Lecture Outline (tentative)

1. Introduction
2. Overview over model families
3. Evaluation
4. Linear and kernel models
5. Distance- and density-based models
6. (Deep-Learning-based models)

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
- E-book can be accessed from the university for free!



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

Lecture 1: Introduction to Anomaly Detection

Anomalies and Outliers

- Terms **anomaly** and **outlier** are often used interchangeably

Douglas M. Hawkins

"The data come from some heavy tailed distribution such as Student's t. There is no question that any observation is in any way erroneous."

D. Hawkins. Identification of Outliers, Chapman and Hall, 1980

Anomalies and Outliers

- Terms **anomaly** and **outlier** are often used interchangeably

Douglas M. Hawkins

"The data come from some heavy tailed distribution such as Student's t. There is no question that any observation is in any way erroneous."

"The data arise from two distributions. One of these, the 'basic distribution', generates 'good' observations, while another, the 'contaminating distribution', generates 'contaminants'."

D. Hawkins. Identification of Outliers, Chapman and Hall, 1980

Anomalies and Outliers

- Terms **anomaly** and **outlier** are often used interchangeably

> "…are patterns in data that do not conform to a well defined notion of normal behavior."

Varun Chandola

Anomalies and Outliers

- Terms **anomaly** and **outlier** are often used interchangeably

"…are patterns in data that do not conform to a well defined notion of normal behavior."

Douglas M. Hawkins

"…are observation which deviates so much from the other observation as to arouse suspicions that it was generated by a different mechanism."

Varun Chandola

Chandola et al., Anomaly detection: A survey, 2009

Anomalies and Outliers

"Anomalies might be induced in the data for a variety of reasons, such as malicious activity, for example, credit card fraud, cyber-intrusion, terrorist activity or break-down of a system, but all of the reasons have the common characteristic that they are interesting to the analyst."

Chandola et al., Anomaly detection: A survey

Anomalies and Outliers

"Anomalies might be induced in the data for a variety of reasons, such as malicious activity, for example, credit card fraud, cyber-intrusion, terrorist activity or break-down of a system, but all of the reasons have the common characteristic that they are interesting to the analyst."

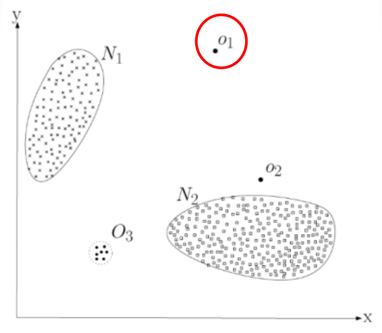
Chandola et al., Anomaly detection: A survey

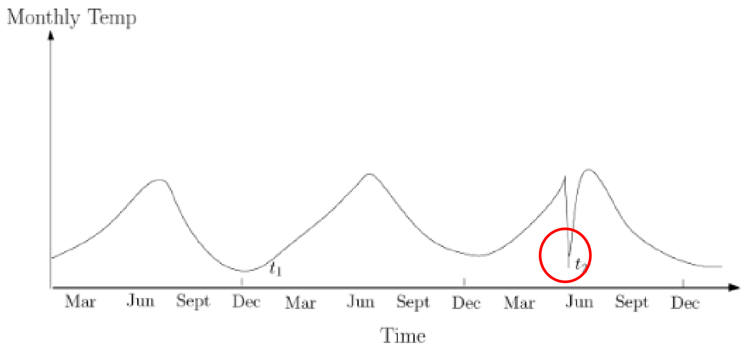
Application examples

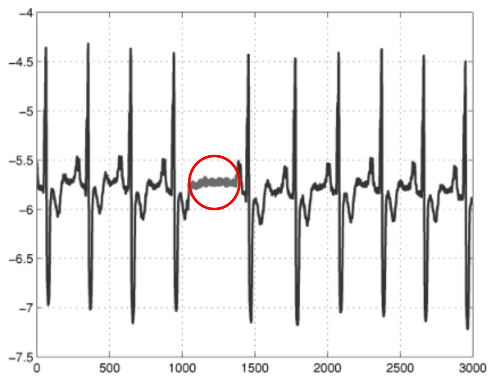
- Intrusion detection
- Credit-card fraud
- Sensor events
- Medical diagnosis
- Law enforcement
- Earth science

Types of Anomalies

- Point anomalies: present or absent with respect to all other data points
- Contextual anomalies: not conspicuous in isolation but in context
  - Temporal context
  - Spatial context
  - Other dependencies
- Collective anomalies: a group or sequence of data points is anomalous



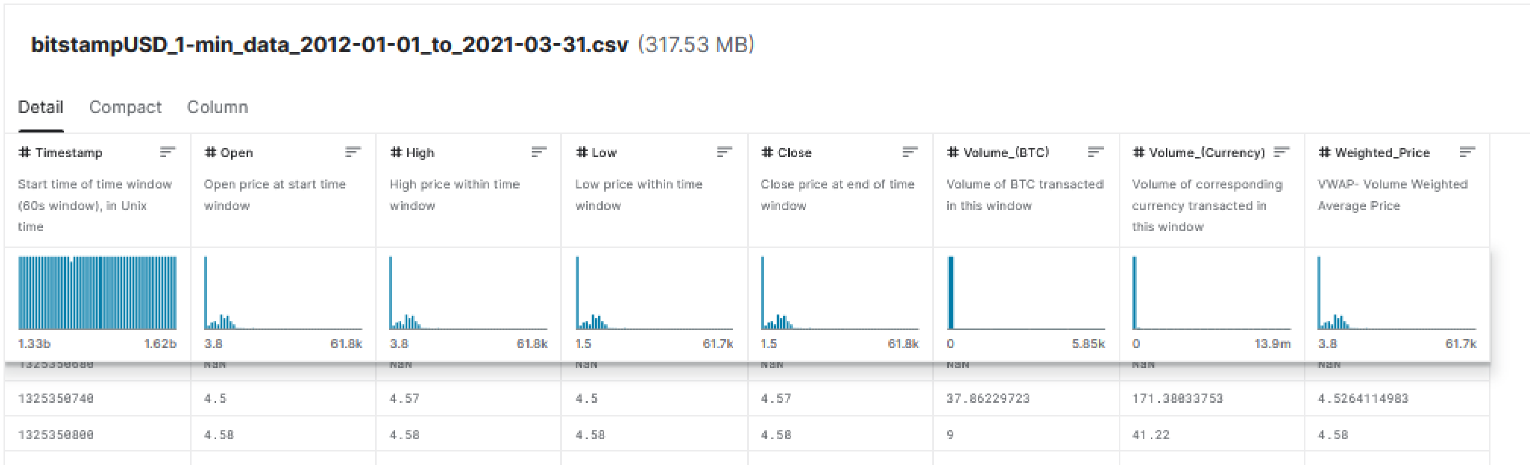




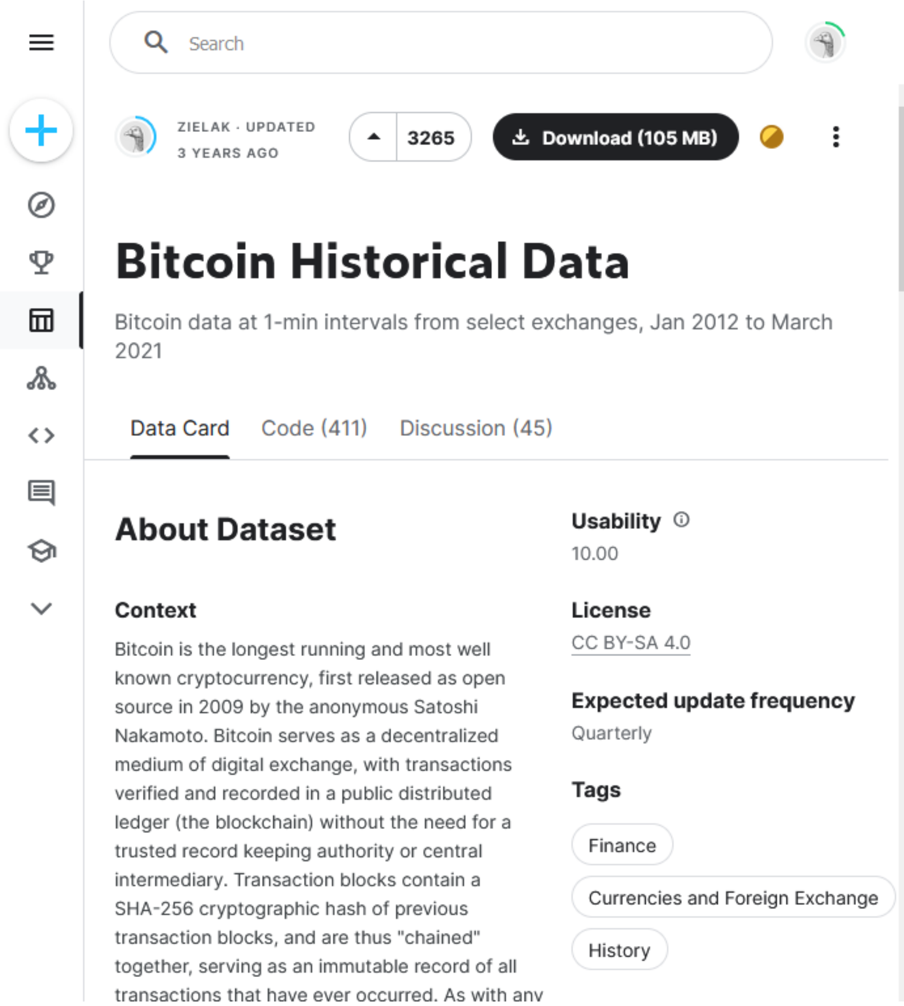
 from https://dl.acm.org/doi/pdf/10.1145/1541880.1541882

Example: Types of Anomalies

- Kaggle: Bitcoin Historical Data
- https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data/



# Timestamp	# Open	# High	# Low	# Close	# Volume_(BTC)	# Volume_(Currency)	# Weighted_Price
Start time of time window (60s window), in Unix time	Open price at start time window	High price within time window	Low price within time window	Close price at end of time window	Volume of BTC transacted in this window	Volume of corresponding currency transacted in this window	VWAP- Volume Weighted Average Price
1.33b – 1.62b	3.8 – 61.8k	3.8 – 61.8k	1.5 – 61.7k	1.5 – 61.8k	0 – 5.85k	0 – 13.9m	3.8 – 61.7k
[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1325350740	4.5	4.57	4.5	4.57	37.86229723	171.38033753	4.5264114983
1325350800	4.58	4.58	4.58	4.58	9	41.22	4.58



Example: Types of Anomalies (cont.)

```
# download the bitcoin history file if needed
import os
from urllib.request import urlretrieve

url = ("https://www.dropbox.com/scl/fi/ufra7gag7g1l5ktbtk43w/bitstampUSD_1-min_data_2012-01-
01_to_2021-03-31.csv?rlkey=bcj9imqn8muw29urg8gt1tda9&dl=1")
filename = "bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv"

if not os.path.exists("bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv"):
    urlretrieve(url, filename)
```

Example: Types of Anomalies (cont.)

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.dates import date2num
from datetime import datetime, timedelta

# Load the Data
df = pd.read_csv('bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv')

df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')
df.set_index(df['Timestamp'], inplace=True)
df.drop('Timestamp', axis=1, inplace=True)

# We look at a specific time frame
data_slice = df[(df.index > '2012-01-01 00:00') & (df.index < '2017-01-28 23:59')]

# Plot data of the "Low" column
ax = data_slice['Low'].plot()

plt.show()
```

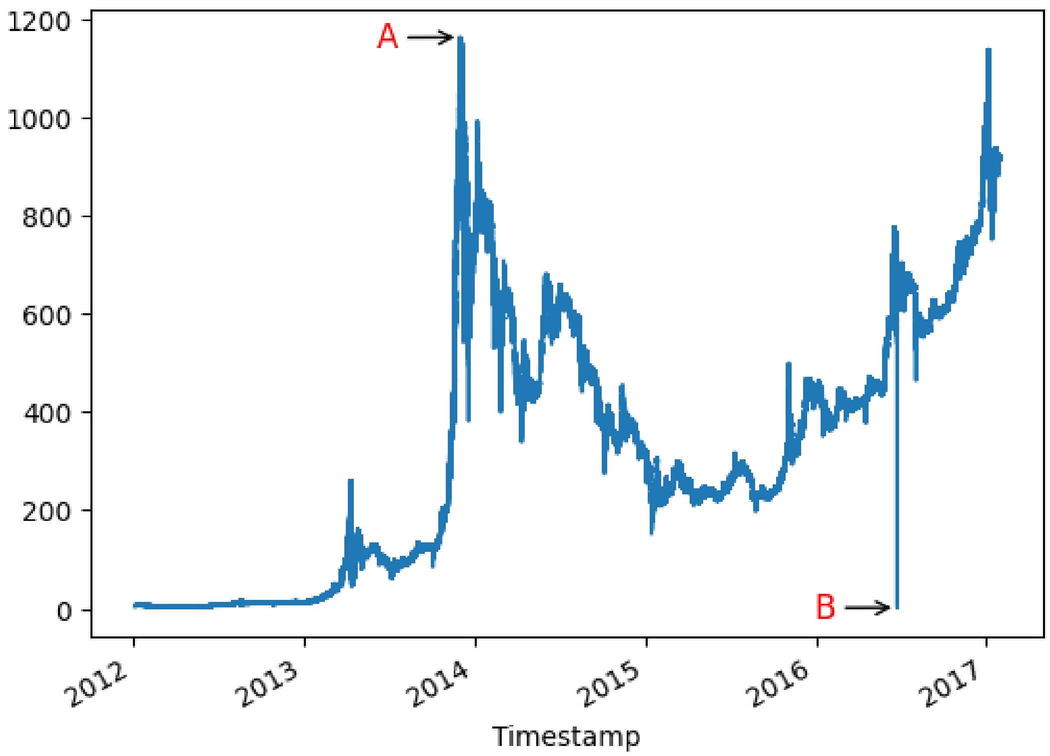
Example: Types of Anomalies (cont.)

```
# Annotate two specific points
annotate_times = {'A':'2013-11-30 03:38:00', 'B': '2016-06-23 12:36:00'}
for name, annotate_time in annotate_times.items():
    annotate_value = data_slice.loc[annotate_time, 'Low']
    annotate_time_dt = datetime.strptime(annotate_time, '%Y-%m-%d %H:%M:%S') - 
timedelta(days=180)
    ax.annotate(name, xy=(annotate_time, annotate_value), 
xytext=(date2num(annotate_time_dt), annotate_value - 20),
                arrowprops=dict(facecolor='black', arrowstyle='->'),
                fontsize=12, color='red')
```

Example: Types of Anomalies (cont.)

Question:

What type of anomalies are A and B?



Example: Types of Anomalies (cont.)

```python
# Point A:
# Get the maximum 'Low' value and the corresponding date
print("Point A:")
max_low_row = data_slice.loc[data_slice['Low'].idxmax()]

max_low_value = max_low_row['Low']
max_low_date = max_low_row.name

print(f"The maximum 'Low' value is {max_low_value} on {max_low_date}")

display(df[(df.index > '2013-11-30 03:34:00') & (df.index < '2013-11-30 03:42:00')]['Low'])
```

```
Point A:
The maximum 'Low' value is 1162.99 on 2013-11-30 03:38:00

Timestamp
2013-11-30 03:35:00    1162.00
2013-11-30 03:36:00    1162.00
2013-11-30 03:37:00    1162.97
2013-11-30 03:38:00    1162.99
2013-11-30 03:39:00    1162.99
2013-11-30 03:40:00    1162.67
2013-11-30 03:41:00    1162.67
Name: Low, dtype: float64
```

Example: Types of Anomalies (cont.)

```python
# Point B:
print("Point B:")
min_low_row = data_slice.loc[data_slice['Low'].idxmin()]

# Get the minimum 'Low' value and the corresponding date
min_low_value = min_low_row['Low']
min_low_date = min_low_row.name

print(f"The minimum 'Low' value is {min_low_value} on {min_low_date}")

display(df[(df.index > '2016-06-23 12:32') & (df.index < '2016-06-23 12:40')]['Low'])
```

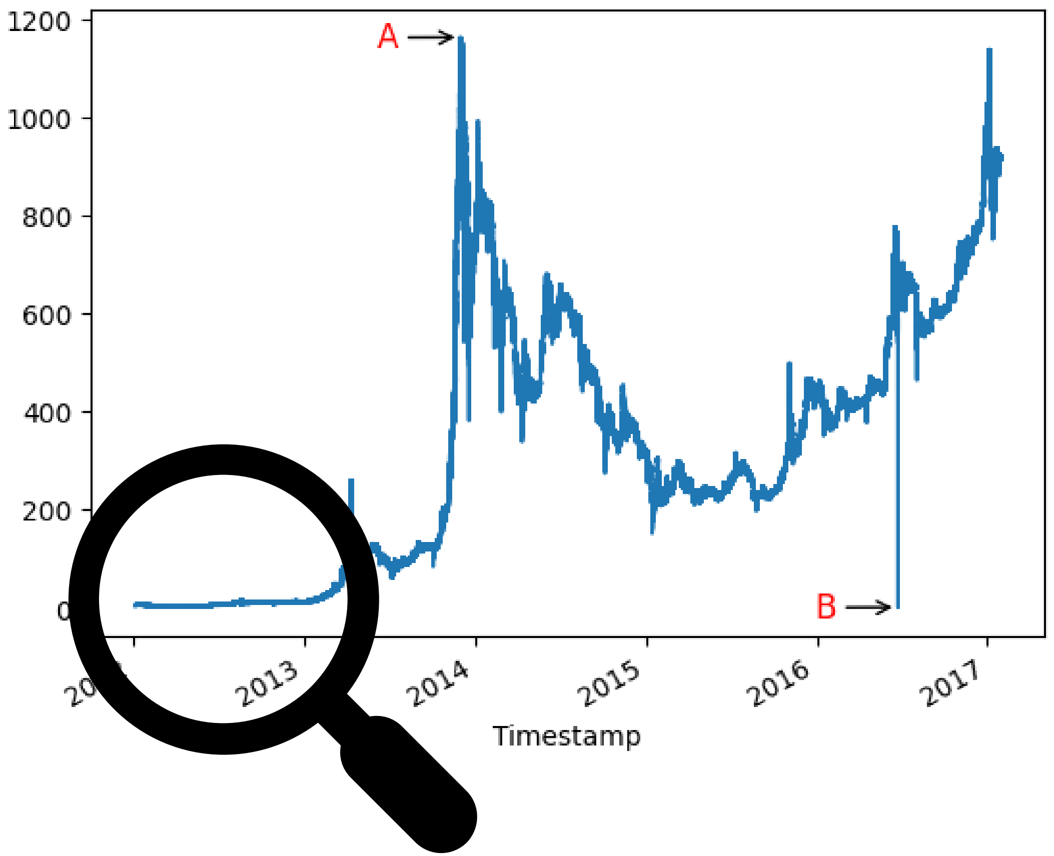
```
Point B:
The minimum 'Low' value is 1.5 on 2016-06-23 12:36:00
Timestamp
2016-06-23 12:33:00    590.54
2016-06-23 12:34:00    590.40
2016-06-23 12:35:00    586.71
2016-06-23 12:36:00      1.50
2016-06-23 12:37:00    584.06
2016-06-23 12:38:00    587.15
2016-06-23 12:39:00    586.43
Name: Low, dtype: float64
```

Example: Types of Anomalies (cont.)

Question:

What type of anomalies are A and B?

- A seems to be a collective anomaly (anomalous group of points)
- B might be a point or contextual anomaly



Example: Types of Anomalies (cont.)

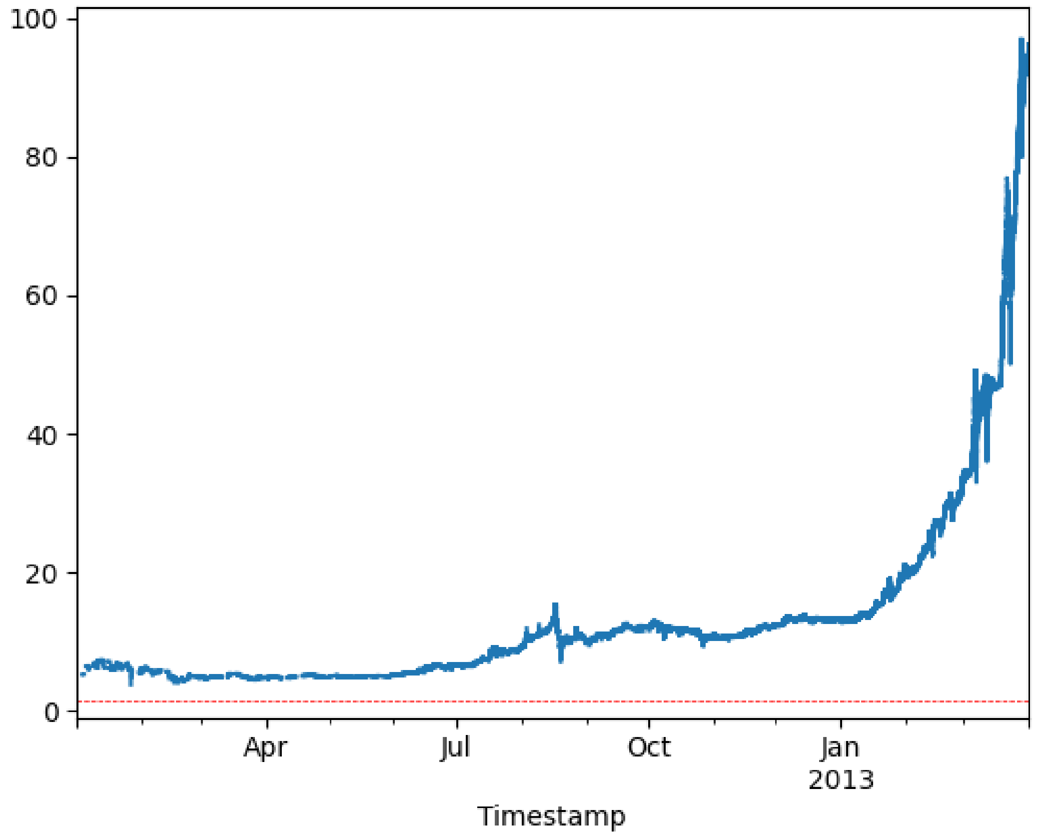
```
Point B:
The minimum 'Low' value is 1.5 on 2016-06-23
12:36:00
Timestamp
2016-06-23 12:33:00    590.54
2016-06-23 12:34:00    590.40
2016-06-23 12:35:00    586.71
2016-06-23 12:36:00      1.50
2016-06-23 12:37:00    584.06
2016-06-23 12:38:00    587.15
2016-06-23 12:39:00    586.43
Name: Low, dtype: float64
```

```python
data_slice = df[(df.index > '2012-01-01 00:00')
& (df.index < '2013-03-31 23:59')]

ax = data_slice['Low'].plot()

ax.axhline(y=1.5, color='red', linestyle='--',
linewidth=0.5)

plt.show()
```

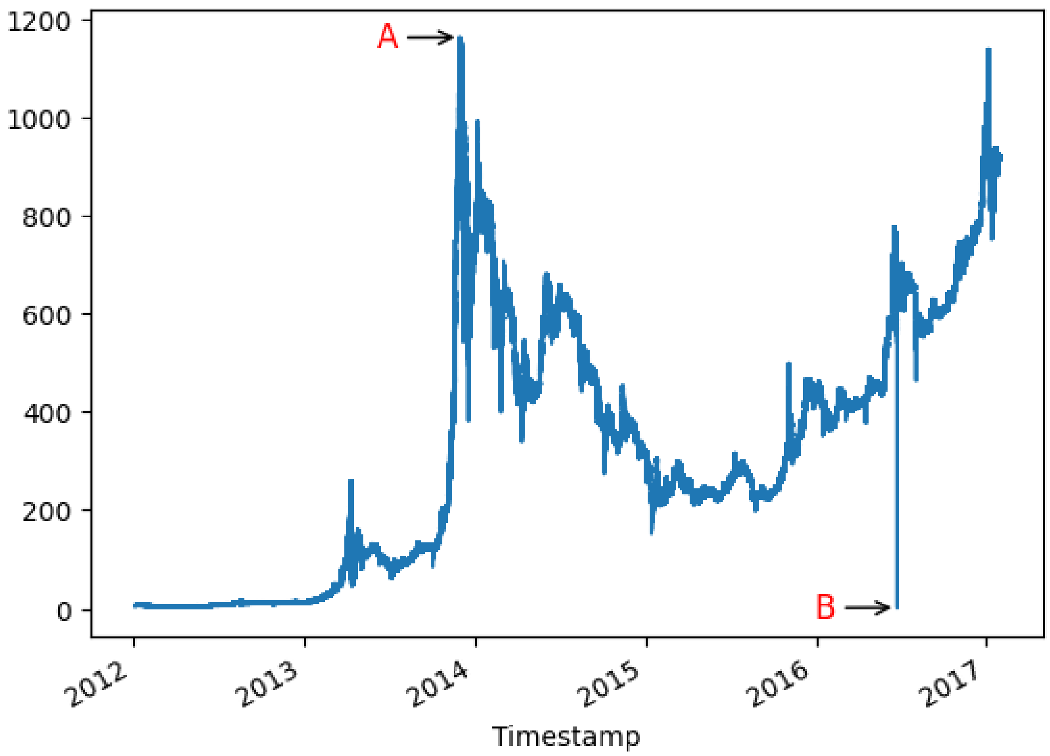


Example: Types of Anomalies (cont.)

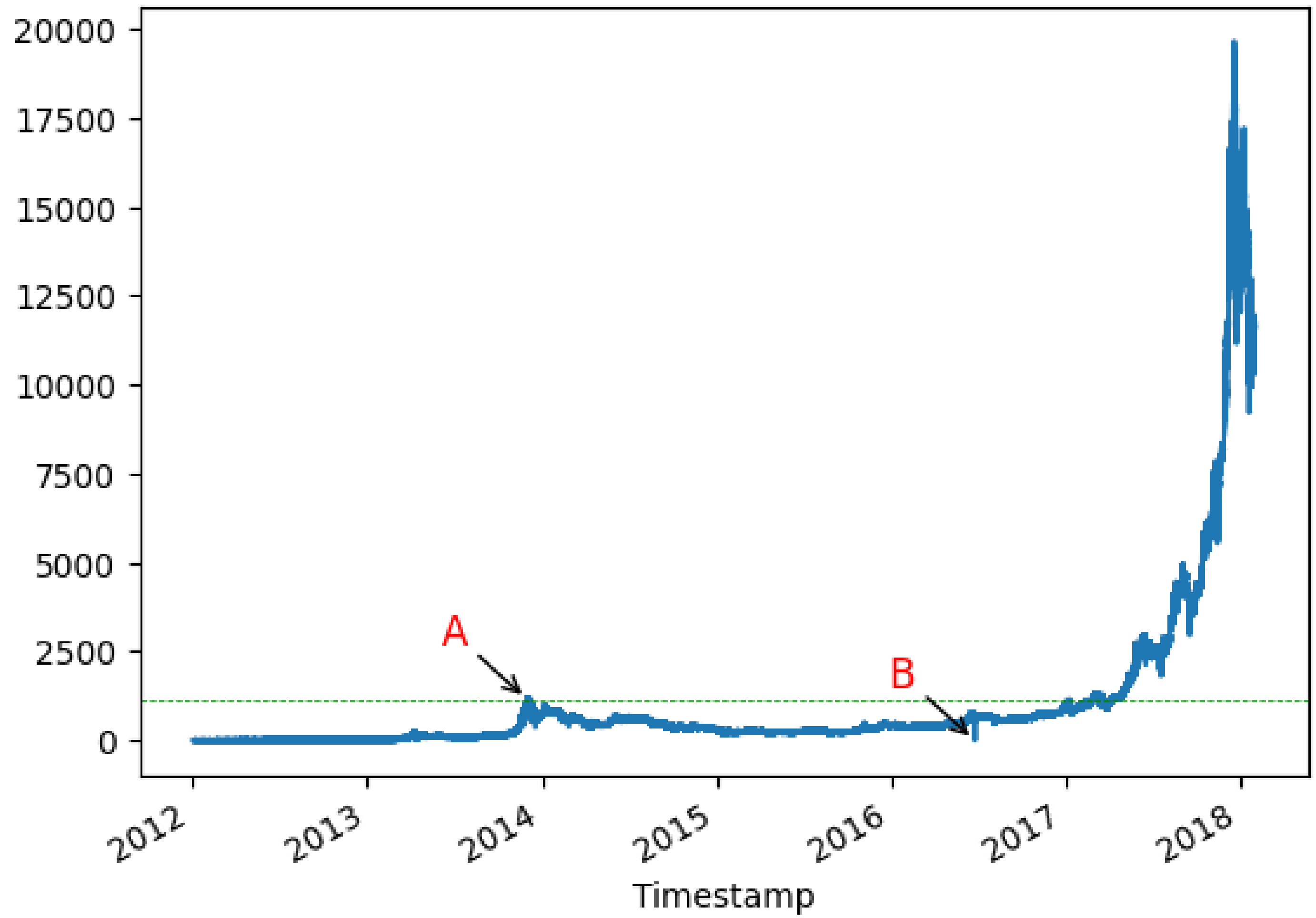
Question:

What type of anomalies are A and B?

- A seems to be a collective anomaly (anomalous group of points)
- B is a point anomaly



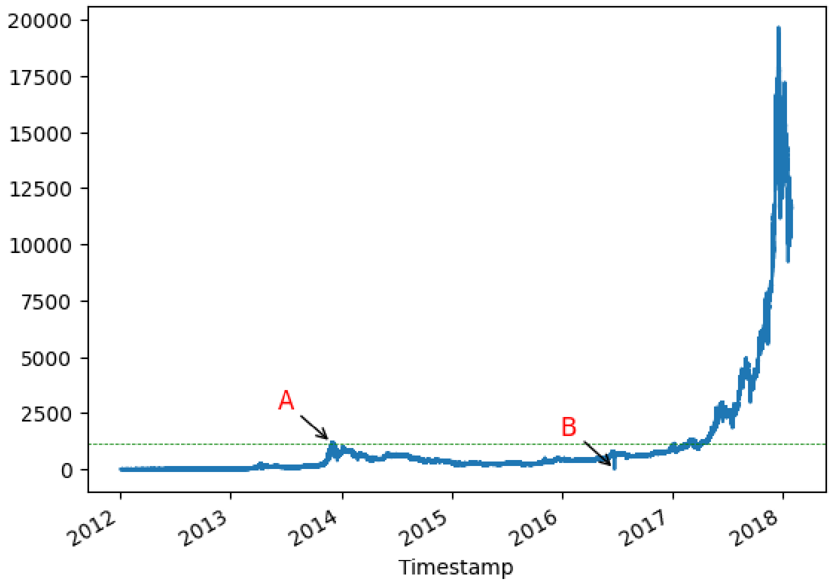
Example: Types of Anomalies (cont.)



Example: Types of Anomalies (cont.)

```
data_slice = df[(df.index > '2012-01-01 00:00') & (df.index < '2018-01-28 23:59')]

ax = data_slice['Low'].plot()
for name, annotate_time in annotate_times.items():
    annotate_value = data_slice.loc[annotate_time, 'Low’]
    annotate_time_dt = datetime.strptime(annotate_time, '%Y-%m-%d %H:%M:%S’)
    annotate_time_dt -= timedelta(days=180)
    ax.annotate(name, xy=(annotate_time, annotate_value),
                xytext=(date2num(annotate_time_dt), annotate_value + 1500),
                arrowprops=dict(facecolor='black', arrowstyle='->'),
                fontsize=12, color='red')
ax.axhline(y=1162.99, color='green', linestyle='--', linewidth=0.5)
plt.show()
```

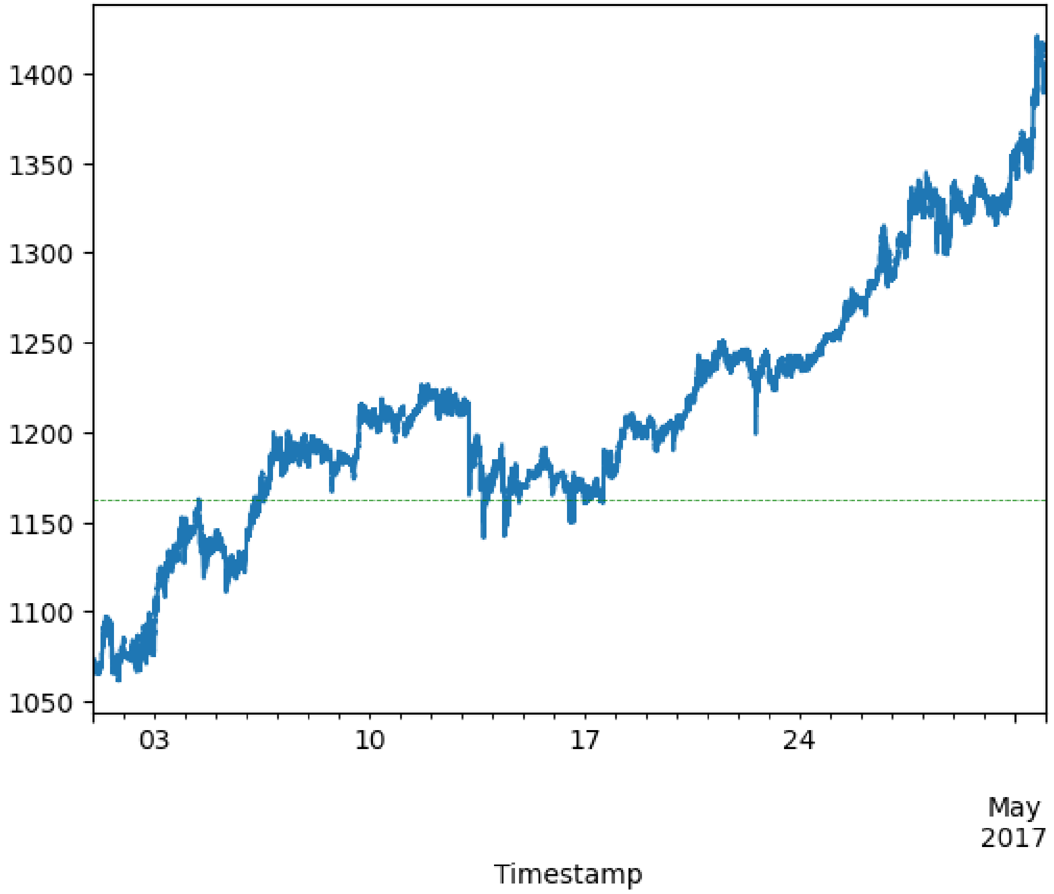


Example: Types of Anomalies (cont.)

Question:

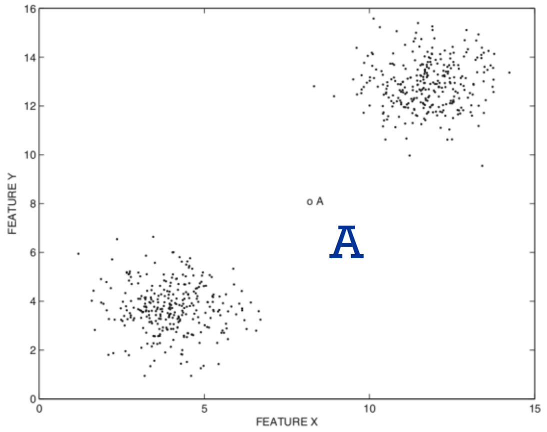
What type of anomalies are A and B?

- A is a contextual and collective anomaly (anomalous group of points in their context)
- B is a point anomaly



Anomalies vs. Noise

- In real applications: Noise in data
- Noise typically also "generated by a different mechanism"
- Noise may *not* be of interest to the analyst



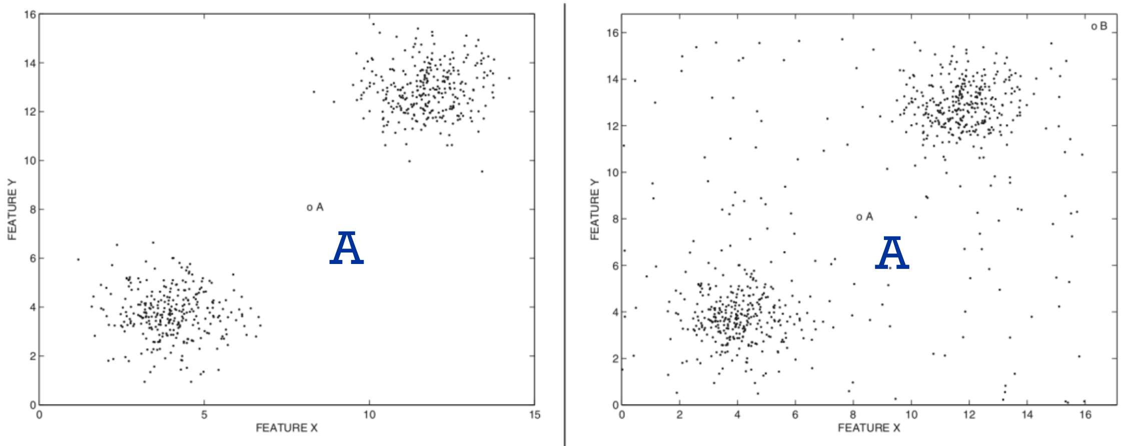
(a) No noise

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

- Main patterns are identical (two clusters)
- (a): Point A is anomaly

Anomalies vs. Noise

- In real applications: Noise in data
- Noise typically also "generated by a different mechanism"
- Noise may *not* be of interest to the analyst



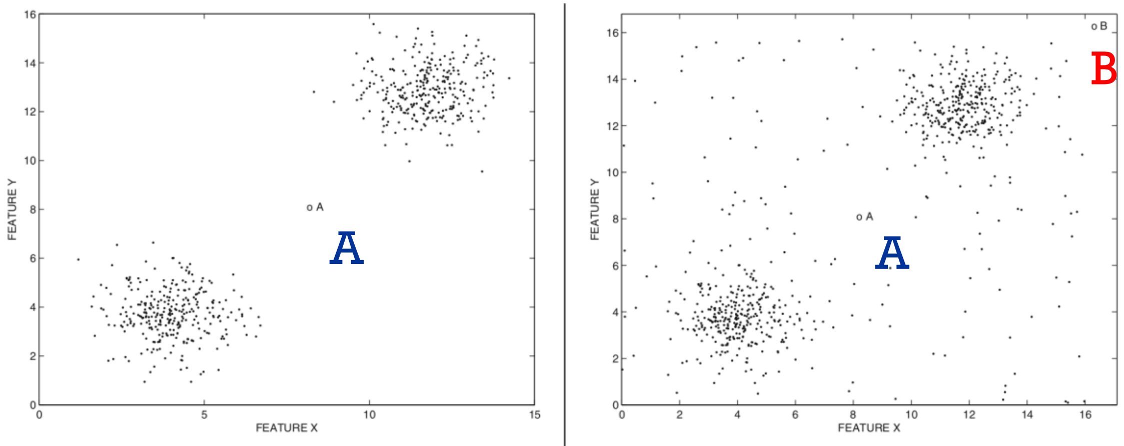
(a) No noise (b) With noise

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

- Main patterns are identical (two clusters)
- (a): Point A is anomaly
- (b): Point A is quite likely noise (fits pattern of random noise)

Anomalies vs. Noise

- In real applications: Noise in data
- Noise typically also "generated by a different mechanism"
- Noise may *not* be of interest to the analyst



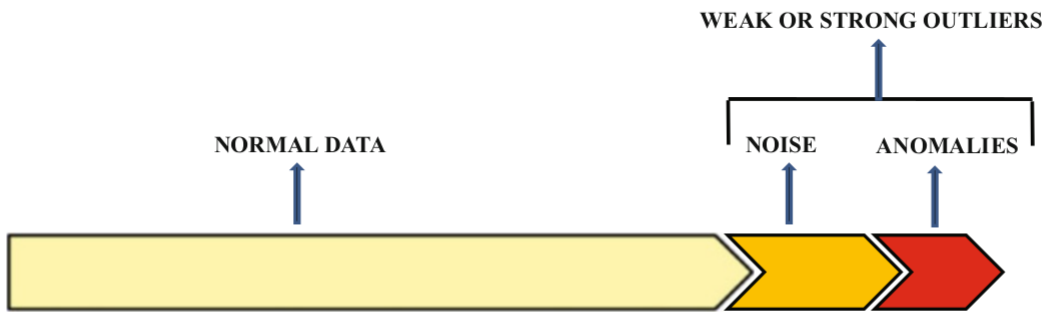
(a) No noise
(b) With noise

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

- Main patterns are identical (two clusters)
- (a): Point A is anomaly
- (b): Point A is quite likely noise (fits pattern of random noise)
- Anomaly vs. outlier: is it of interest to an analyst (A vs. B)?
- Noise: semantic boundary between normal data and true anomalies

Anomalies vs. Noise

- In real applications: Noise in data
- Noise typically also "generated by a different mechanism"
- Noise may *not* be of interest to the analyst



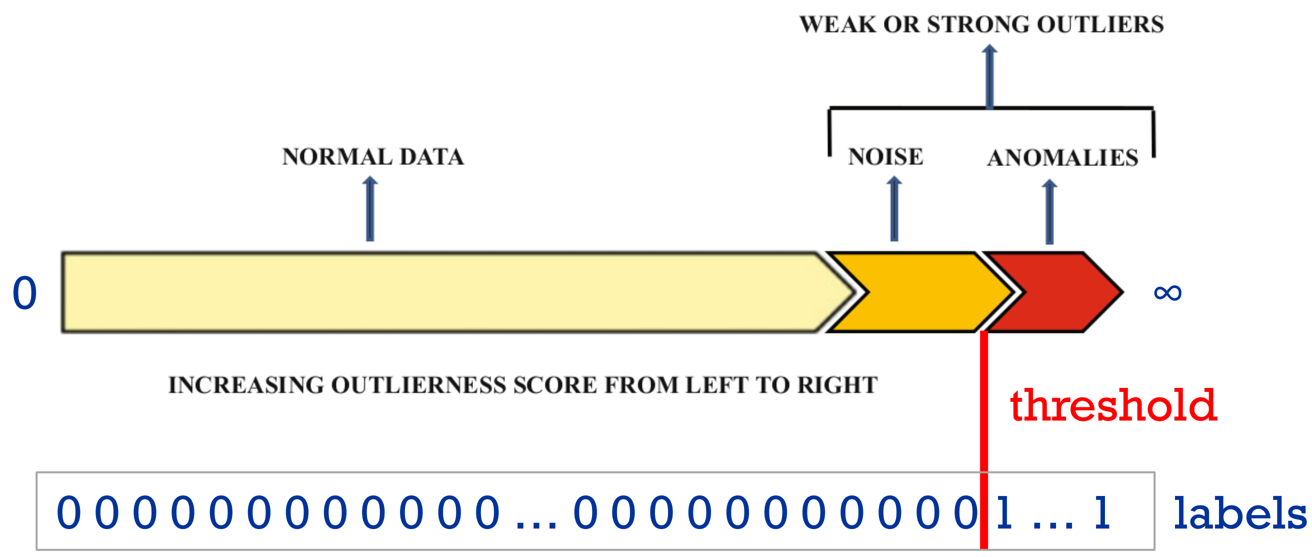
from https://link.springer.com/book/10.1007/978-3-319-47578-3

- Main patterns are identical (two clusters)
- (a): Point A is anomaly
- (b): Point A is quite likely noise (fits pattern of random noise)
- Anomaly vs. outlier: is it of interest to an analyst (A vs. B)?
- Noise: semantic boundary between normal data and true anomalies

Outlier Detection Output

- Binary labels (is an outlier or not)
- Outlier scores (level of outlierness)
- Outlier score can be converted to binary
  - By threshold
  - By extreme value analysis
  - As a machine learning task
- Often important: Are known anomalies available / is training data available

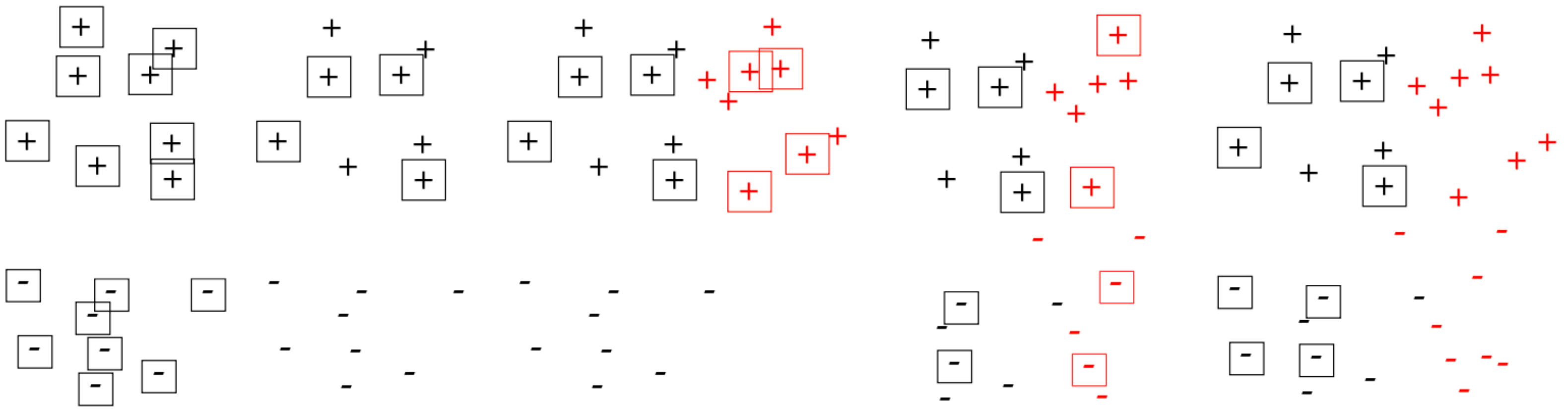
=> supervision vs. unsupervised



Supervised vs. Unsupervised Outlier Detection

- Unsupervised methods
  - Either for noise removal or anomaly detection
  - Often used in an exploratory setting
- Supervised methods: application-specific anomaly detection

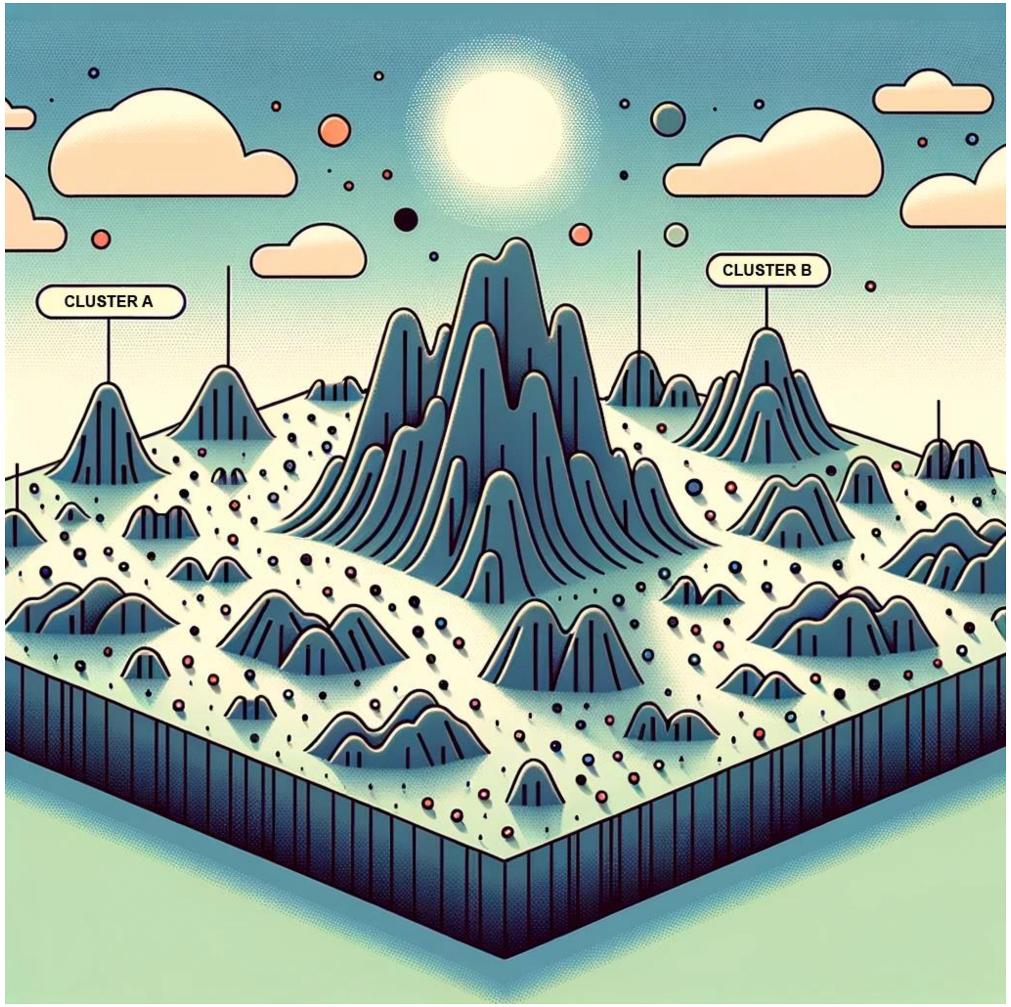
- Different levels of supervision:
  - Fully supervised: normal and abnormal data available
  - PU learning / contaminated supervised: examples of outliers given, normal data may contain outliers
  - Semi-supervised: only normal / only outliers available
  - Semi-unsupervised: some but not all classes are known



Supervised Learning | Positive Unlabeled | Multi-Positive Unlabeled | Semi-Supervised | Semi-UnSupervised

The Data Model is Everything

- Outlier detection algorithms:
  learn a model of **normal** data
- Outlier score of a data:
  deviations from this model
- Models make different assumptions
  about the data

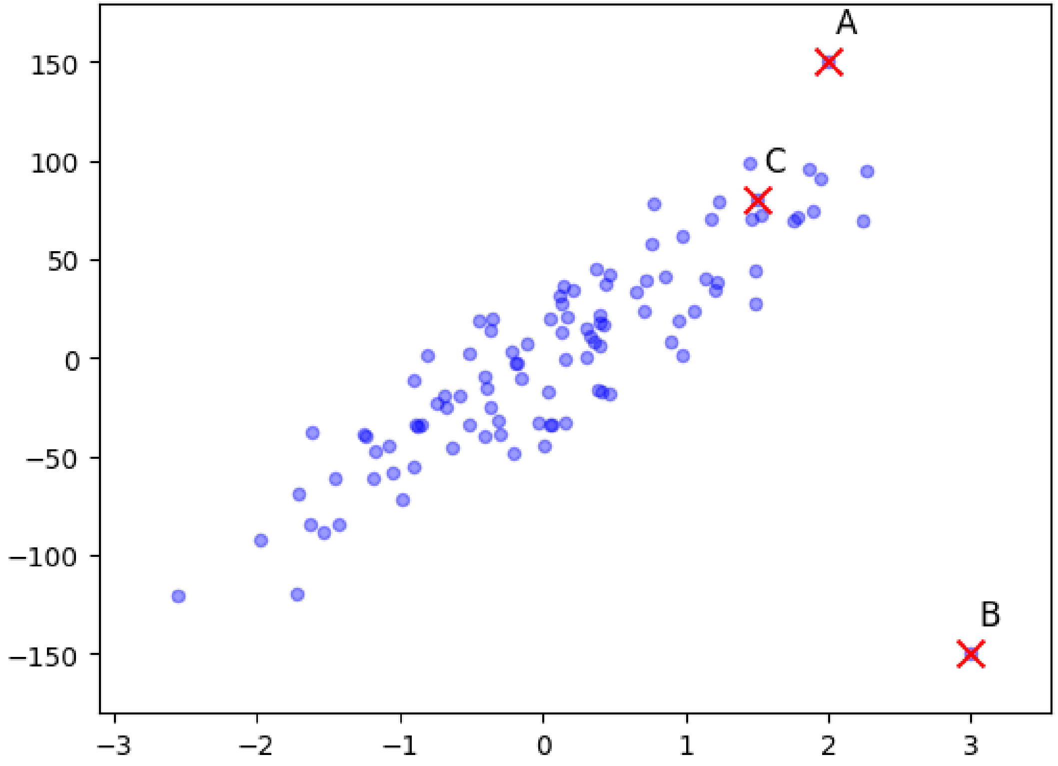


The Data Model is Everything (cont.)

Example 1

Question:

How would you model the following examples, and can you make up a criterion on the “outlierness”?



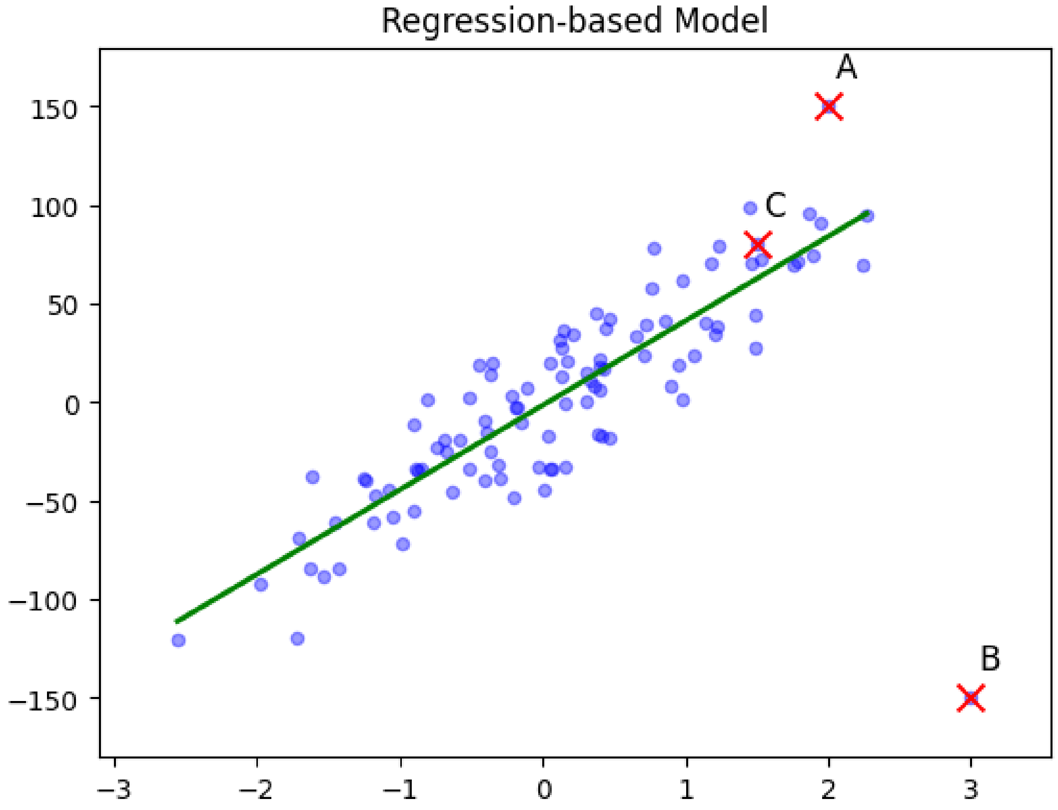
The Data Model is Everything (cont.)

Example 1

Question:

How would you model the following examples, and can you make up a criterion on the "outlierness"?

Linear model, distance to the line

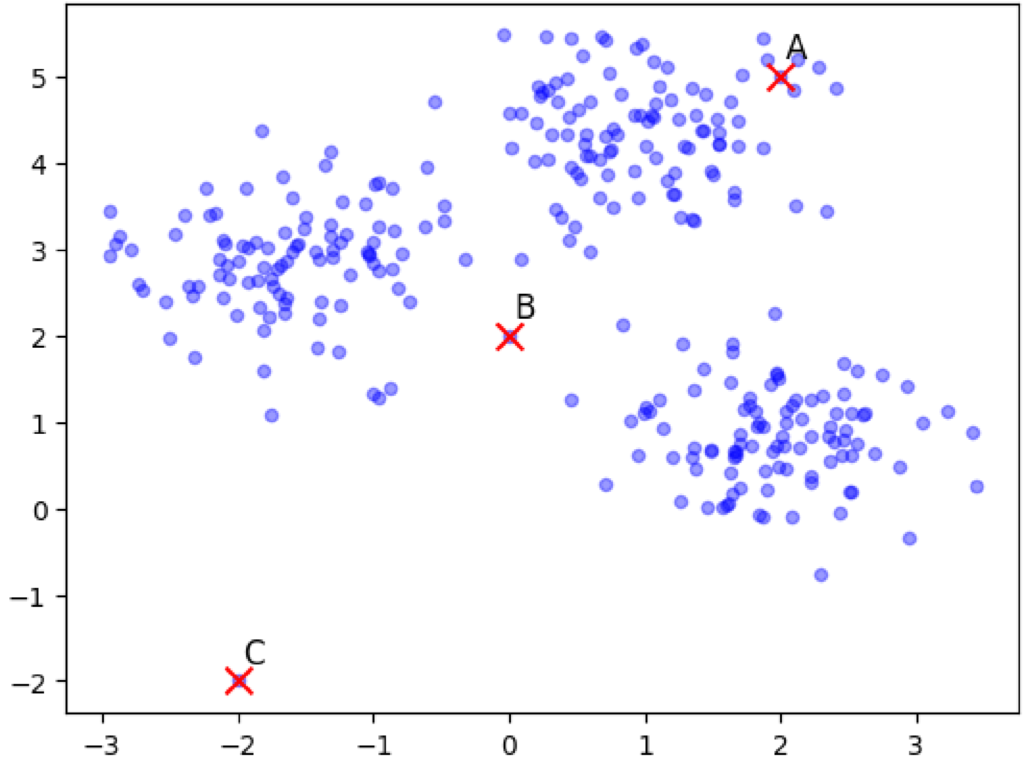


The Data Model is Everything (cont.)

Example 2

Question:

How would you model the following examples, and can you make up a criterion on the “outlierness”?



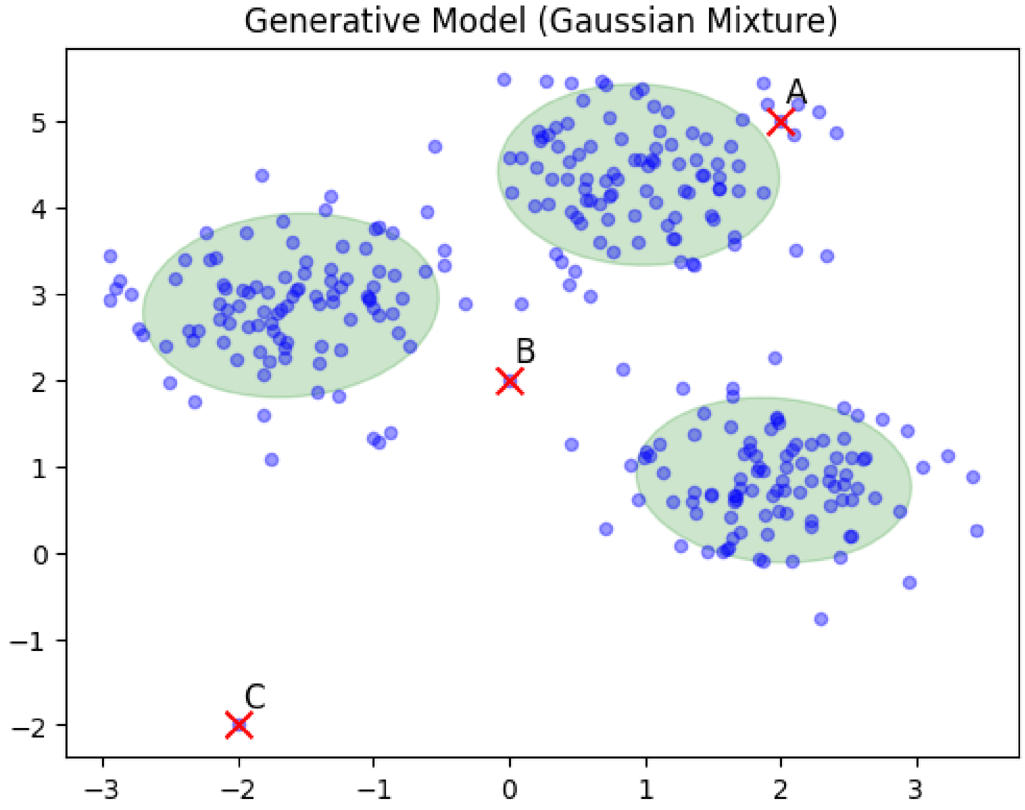
The Data Model is Everything (cont.)

Example 2

Question:

How would you model the following examples, and can you make up a criterion on the "outlierness"?

Clustering based model, distance to cluster centroid

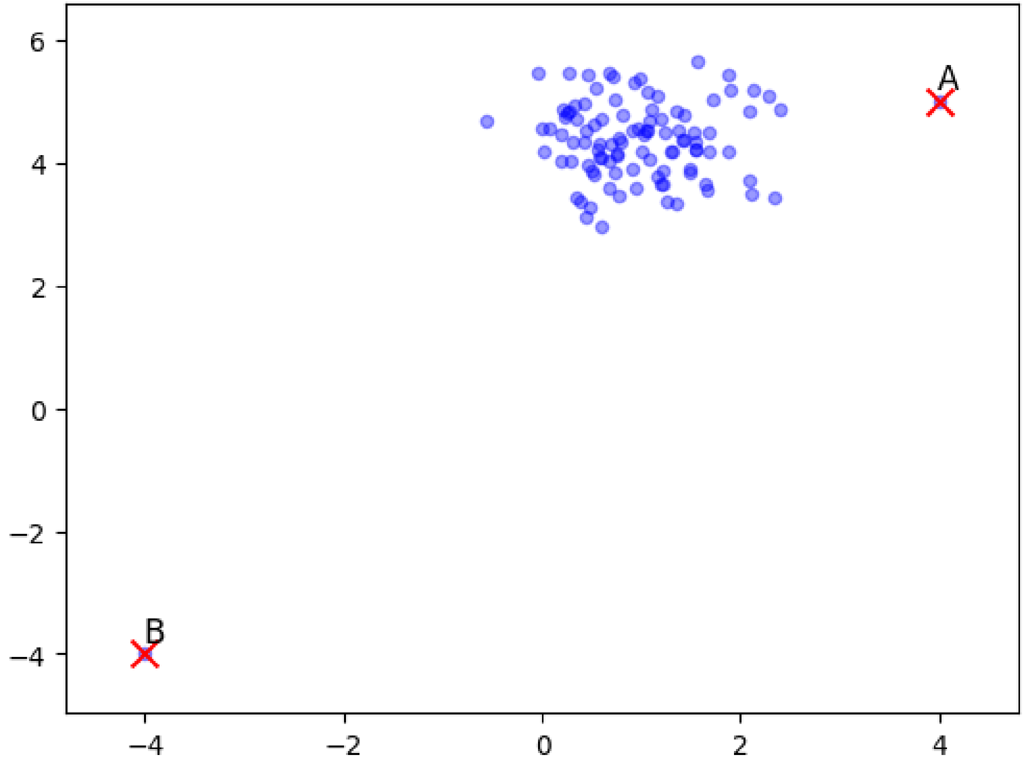


The Data Model is Everything (cont.)

Example 3

Question:

How would you model the following examples, and can you make up a criterion on the “outlierness”?



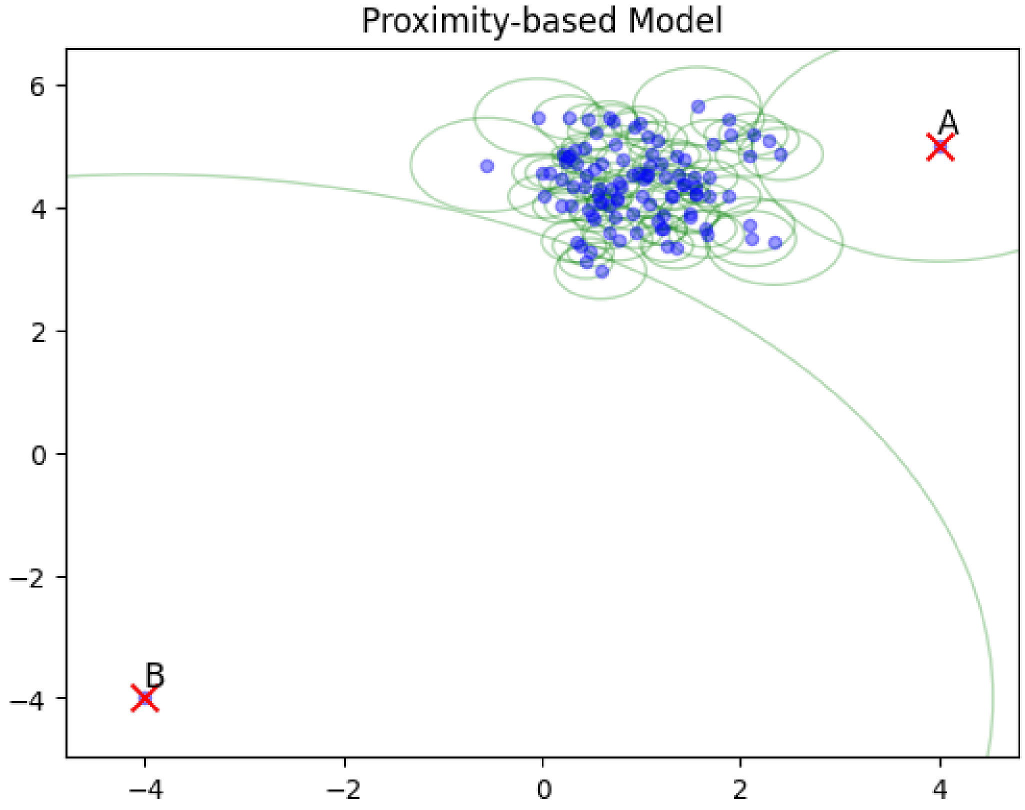
The Data Model is Everything (cont.)

Example 3

Question:

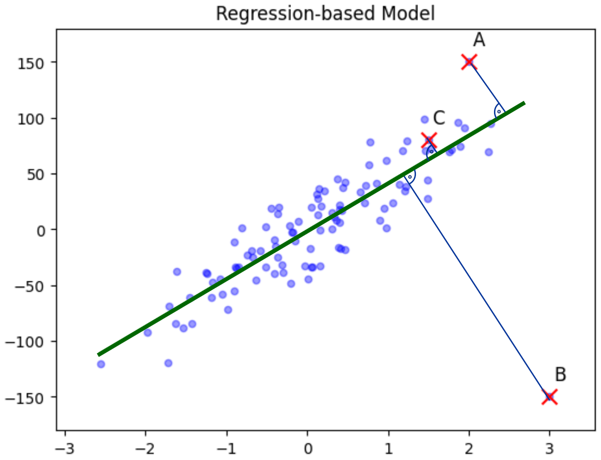
How would you model the following examples, and can you make up a criterion on the “outlierness”?

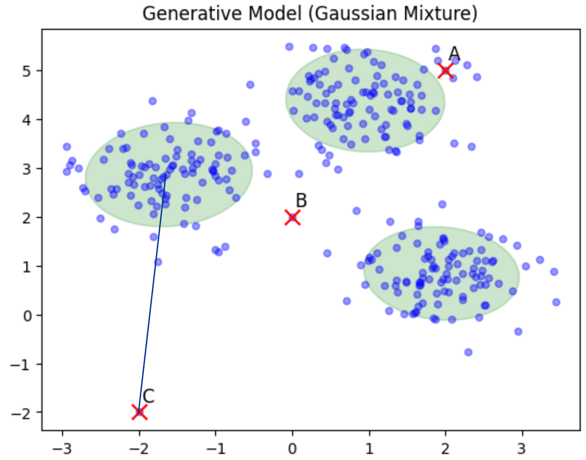
Density- / proximity-based models, distance to *n* neighbors

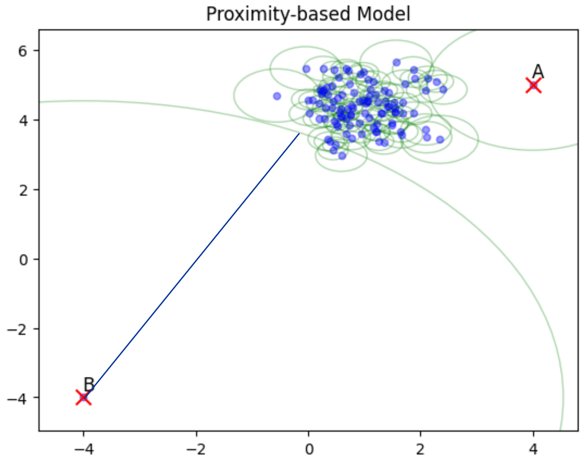


The Data Model is Everything (cont.)

- Outlier score corresponds to the fit between data point and model
- Choice of data model is crucial
- Anomaly detection is **typically an unsupervised problem**
  - Examples of outliers are not given to learn the best model
  - Understanding of data and data deviations important







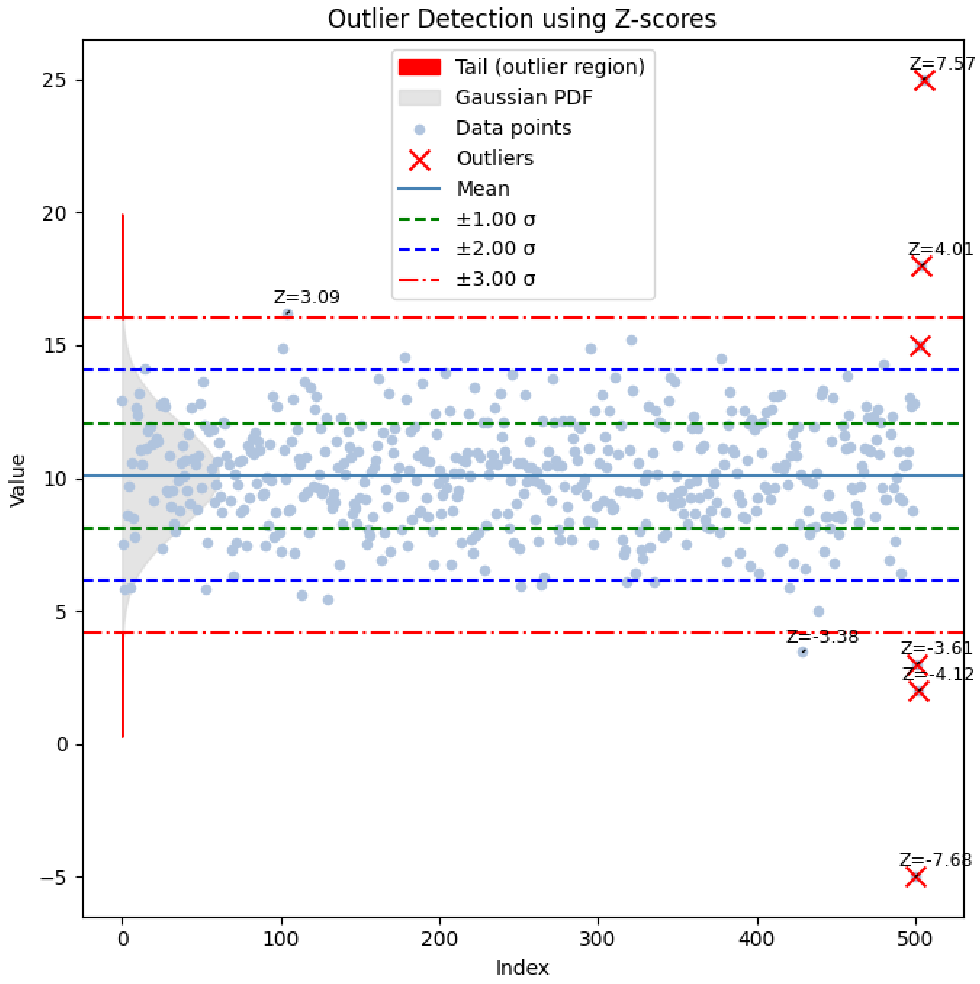
Overview of Anomaly Detection Methods

Z-value Test for Outlier Detection

- Simple model for outlier detection
- One-dimensional data $X_i, \ldots, X_N$ with mean μ and standard deviation σ
- Z-value for a data point X_i:

$$z_i = \frac{|x_i - \mu|}{\sigma}$$

- Z-value denotes the number of standard deviations to mean
- **Implicit assumption: data follows normal distribution**



Z-value Test for Outlier Detection

- “3σ rule-of-thumb”: $z_i \geq 3$ as decision criterion for anomalies:

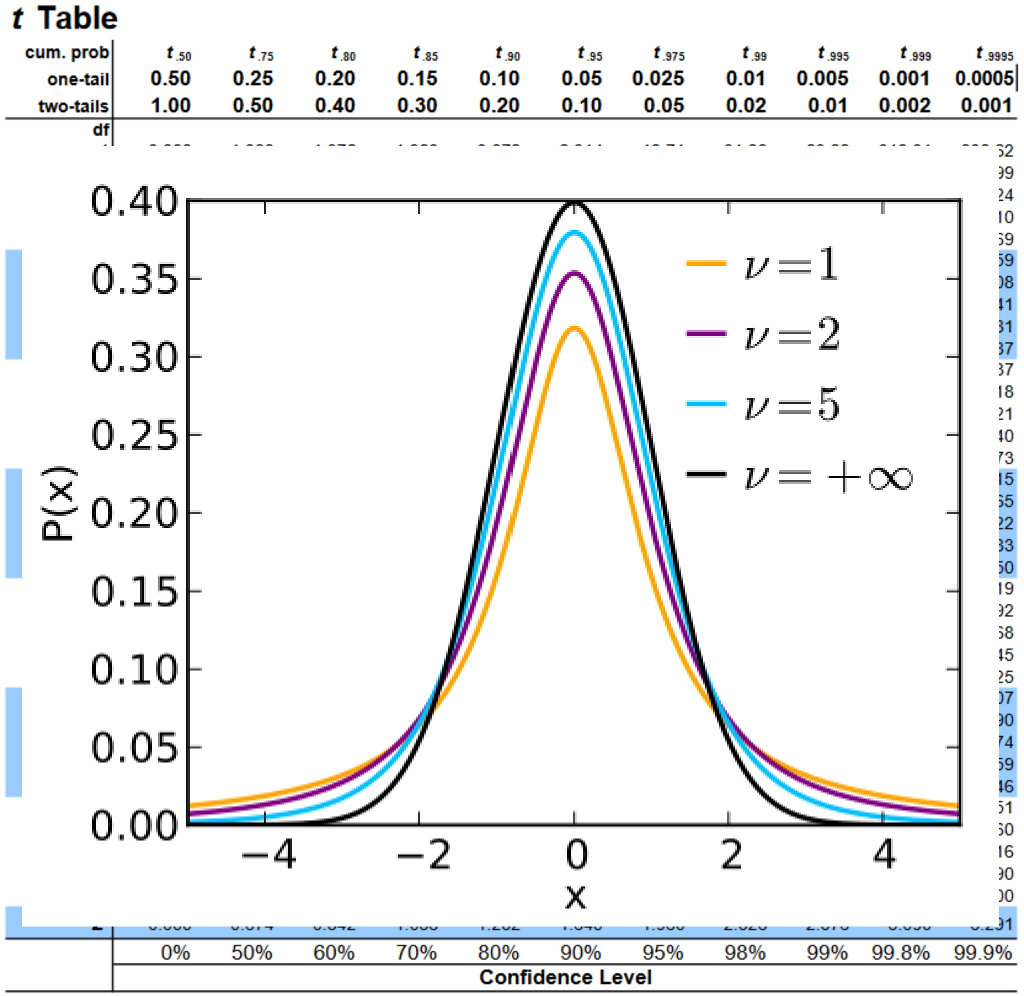
$$P(z < 3) = 0.9973$$

- Typically: μ and σ not explicitly known
  - For enough data (n > 30) assumption of normality
  - For few data interpretation by *Student's t-distribution* and the (absolute of the) *t-value*

$$t_i = \left| \frac{x_i - \mu}{\sigma / \sqrt{n}} \right|$$

for sample size n

$$\text{unbiased } \sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{n-1}}$$



https://www.sjsu.edu/faculty/gerstman/StatPrimer/t-table.pdf

Example: Z-value Test

Given 100 samples S with estimated

mean: 7.13

std.dev: 1.86

including $x = \{1, 2, 13, 14\} \subset S$.

1. Calculate the Z-values for these points.
2. Decide, which will be labeled as outliers according to the "3σ rule-of-thumb".

Calculate: $z = |x-\mu| / \sigma$

Data point (1):

z-score = 6.13 / 1.86 =

= 3.2957

Data point (2):

z-score = 5.13 / 1.86

= 2.7581

Data point (13):

z-score = 5.87 / 1.86

= 3.1559

Data point (14):

z-score = 6.87 / 1.86

= 3.6935

Outliers: 1, 13, 14 z>=3

Example: Z-value Test

Given 100 samples S with estimated

mean: 7.13

std.dev: 1.86

including $x = \{1, 2, 13, 14\} \subset S$.

1. Calculate the Z-values for these points.
2. Decide, which will be labeled as outliers according to the “3σ rule-of-thumb”.

```python
mean = 7.13
std = 1.86
p = [1, 2, 13, 14]

z_val = [abs(x - mean) / std for x in p]
print("z-values", z_val)

# > z-values [3.2956989247311825, 2.758064516129032,
3.1559139784946235, 3.693548387096774]

out = [p[i] for i, z in enumerate(z_val) if abs(z) > 3]
print("Outliers:", out)

# > Outliers: [1, 13, 14]
```

Example: Z-value Test

Given the following data points:

3, 8, 6, 15, 13, 7

1. Calculate the mean and sample standard deviation.
2. Compute the t-values for each data point.
3. Identify if any points are outliers for a tail probability mass of 0.05.

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587

Example: Z-value Test

Given the following data points:

3, 8, 6, 15, 13, 7

1. Calculate the mean and sample standard deviation.
2. Compute the t-values for each data point.
3. Identify if any points are outliers for a tail probability mass of 0.05.

```python
import numpy as np
data = [3, 8, 6, 15, 13, 7]
mean = np.mean(data)
print("Mean:", mean)

# > Mean: 8.666666666666666

std = np.std(data, ddof=1)
print("Sample Standard Deviation:", std)

# > Sample Standard Deviation: 4.501851470969102

n = len(data)
t_val = [(x - mean) / (std / np.sqrt(n)) for x in data]
print("t-values:", t_val)

# > t-values: [-3.083274062967549, -
0.3627381250500547, -1.4509525002200228,
3.4460121880225554, 2.357797812857538, -
0.9068453126375141]

t_critical = 2.571 # df = 5 and alpha = 0.05
outliers = [data[i] for i, t in enumerate(t_val) if
abs(t) > t_critical]
print("Outliers:", outliers)

# > Outliers: [3, 15]
```

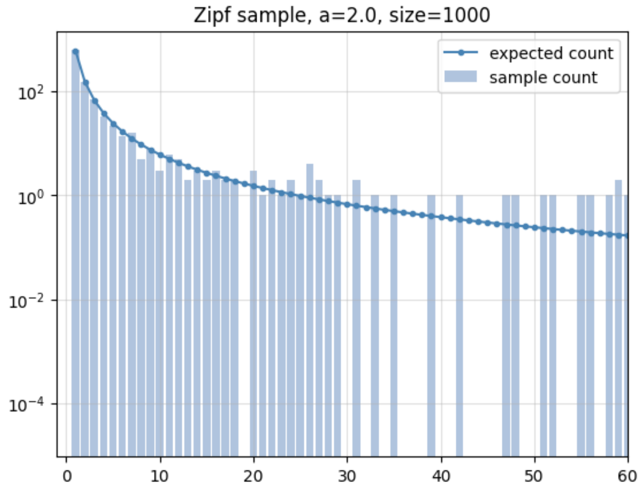
Z-value Test for Outlier Detection (cont.)

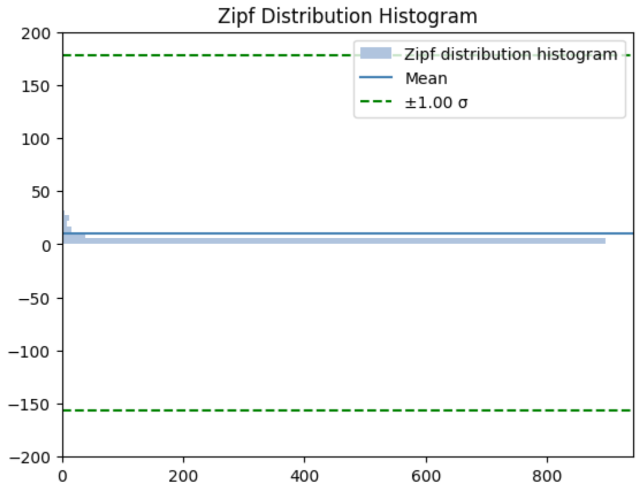
- What if data is not normal distributed?
- E.g. zipf with $k \geq 1, a > 1$

$$\mathrm{zipf}(k; a) = \frac{k^{-a}}{\zeta(a)}$$

with the Riemann Zeta function

$$\zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s}$$

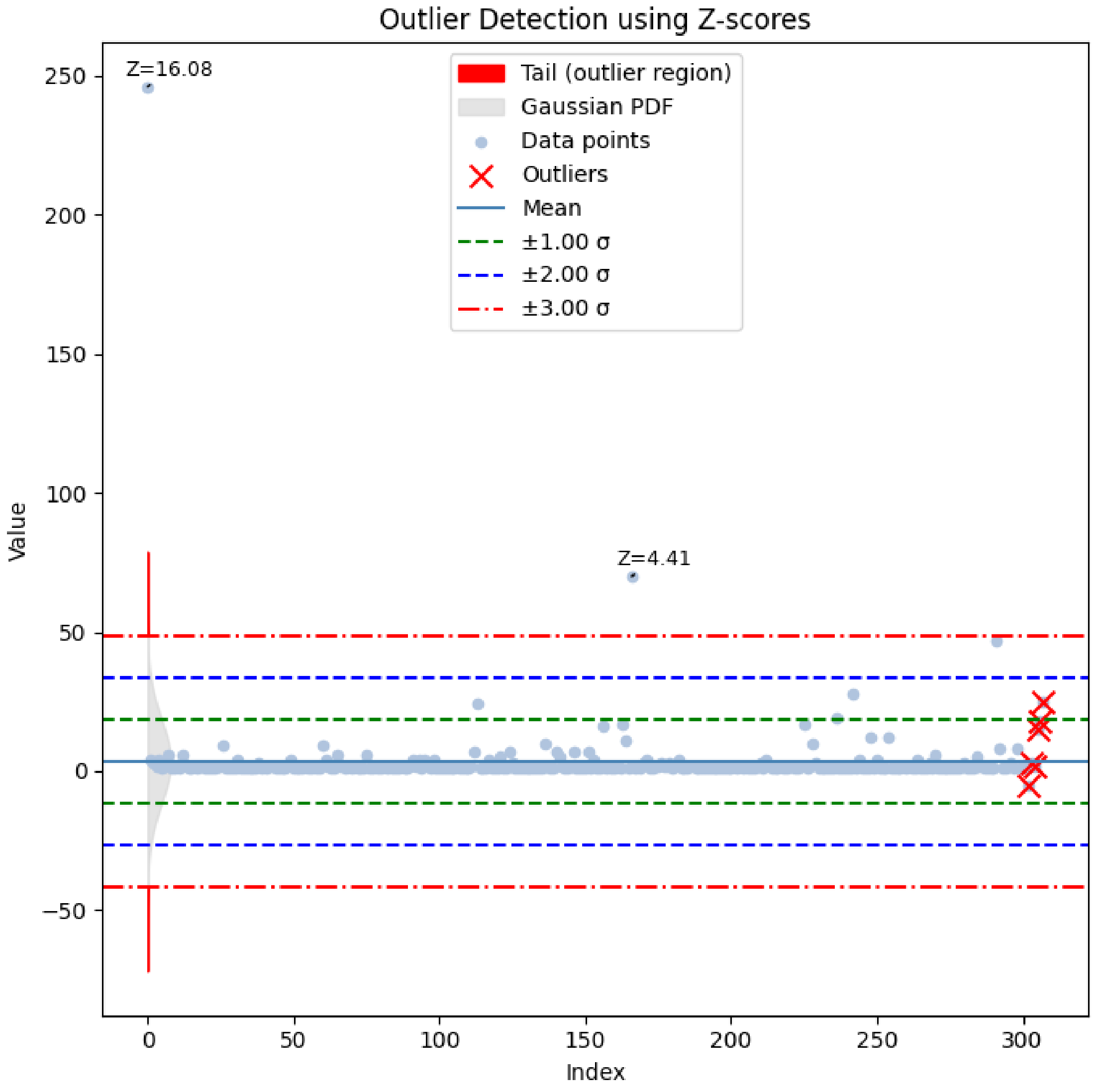




Z-value Test for Outlier Detection (cont.)

- What if data is not normal distributed?

Standard deviation and mean (and thus Z-value) are not meaningful



The Data Model is Everything (cont.)

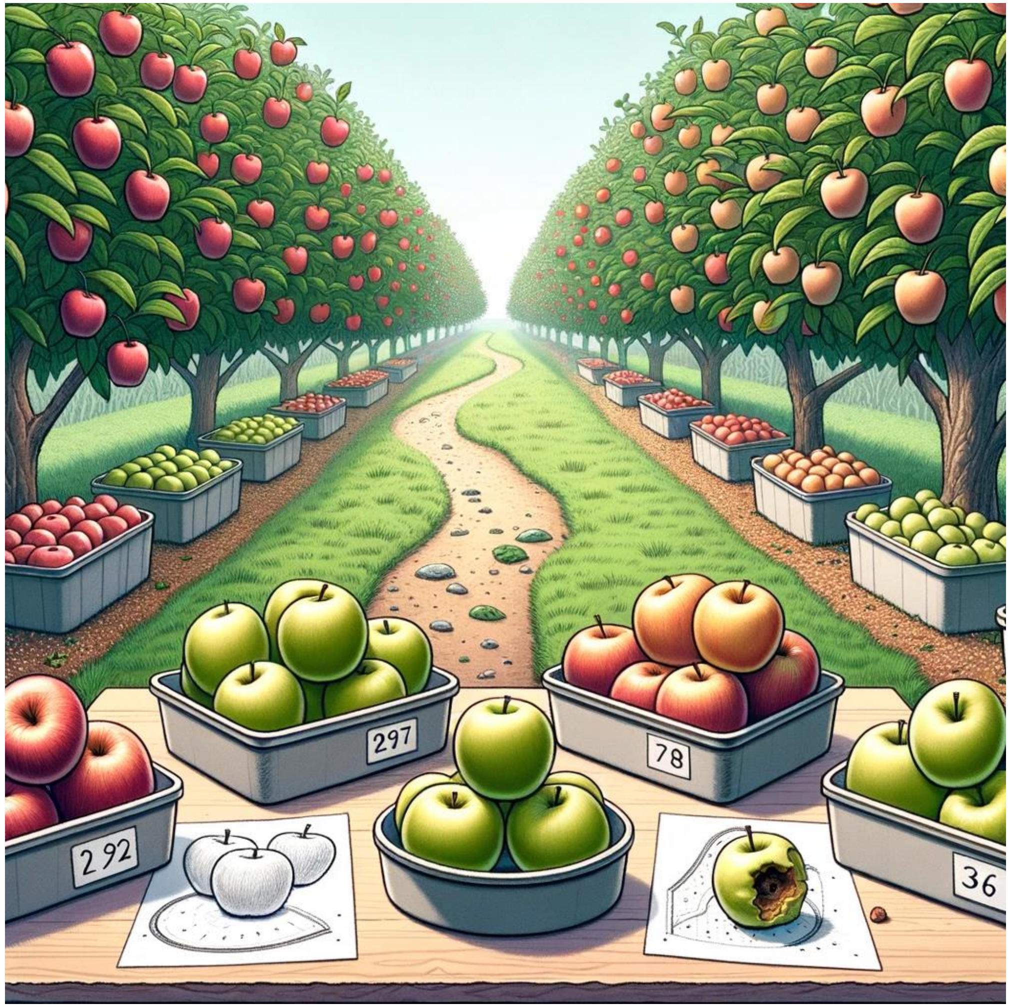
- Mistakes made at the modeling stage can result in incorrect understanding of the data
- Best choice of a model is often data-specific
- Core principle based on assumptions about the structure of normal patterns
- Choice of the “normal” model: understanding of data patterns in that particular domain

Connection with Supervised Models

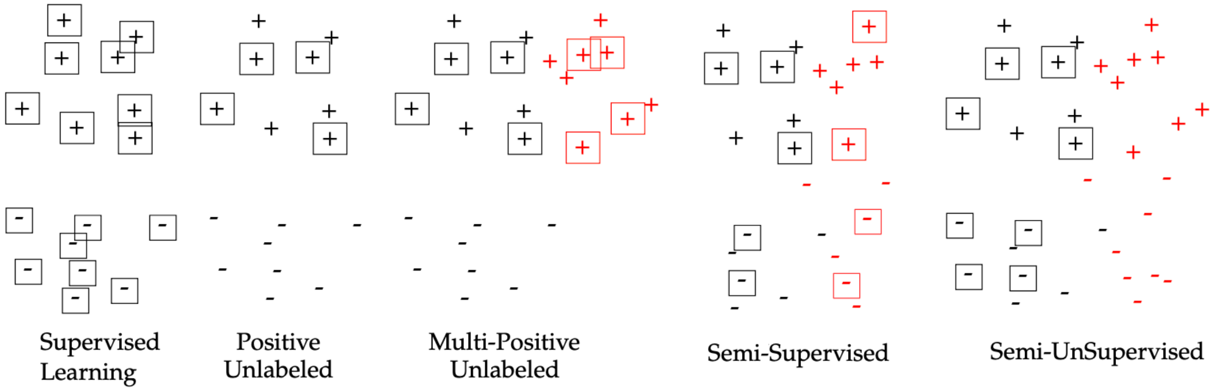
- Anomaly detection: classification problem for which (one) class label is unobserved
- As #normal samples $\gg$ #anomalies
  - Treat whole unlabeled (contaminated) data as normal and create a (noisy) model
  - Deviation from normal model treated as outliers

=> Theory and methods from classification can generalize to anomaly detection

Connection with Supervised Models (cont.)



- One-class analog of multi-class classification
- Unobserved nature of labels (or outlier scores) typically "unsupervised" perspective
- If labels are given, anomaly detection simplifies to imbalanced classification



Connection with Supervised Models (cont.)

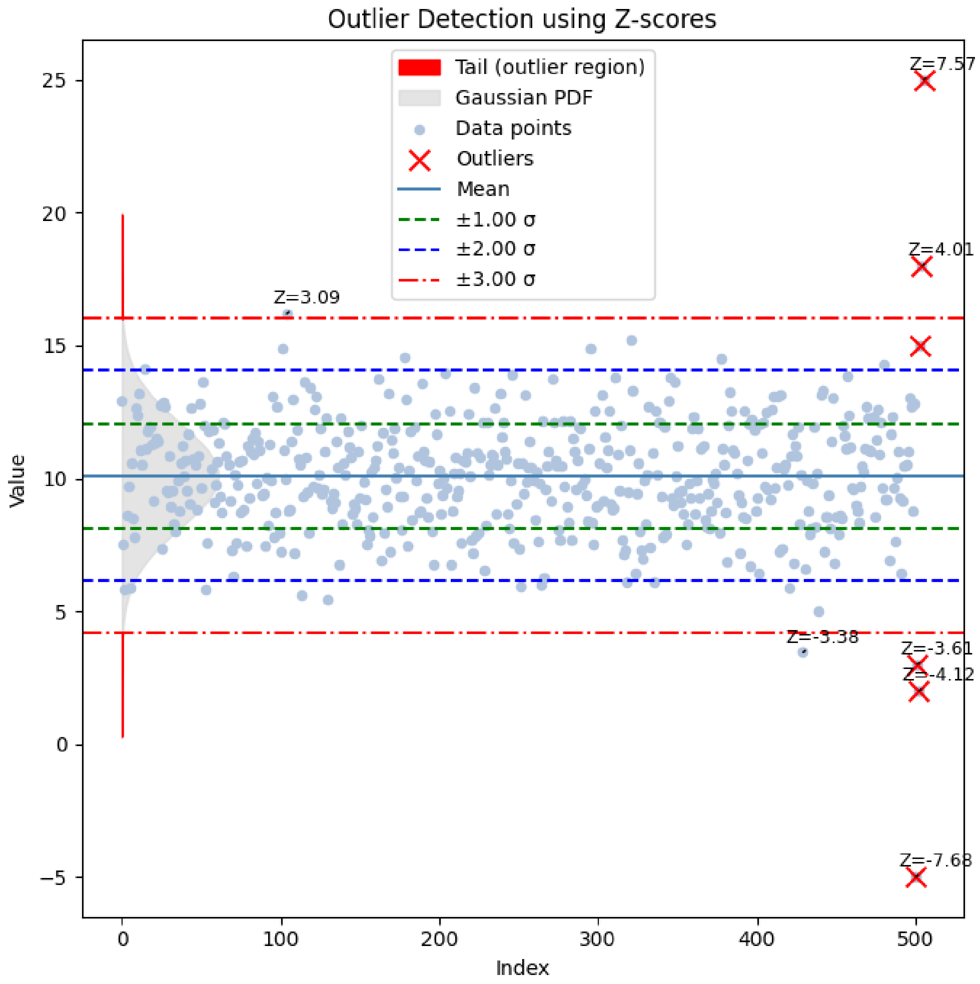
- Differentiation between instance-based and explicit generalization to model normal behavior
- Instance-based: No model constructed up front, most relevant instance from training used to make predictions
- Explicit Generalization: (Summarizing) model created up front, outlierness scored based on this model of normal behavior

Supervised Model	Unsupervised Analog	Type
k-nearest neighbor	k-NN distance, LOF, LOCI	Instance-based
Linear Regression	Principal component Analysis	Explicit Generalization
Naive Bayes	Expectation-maximization	Explicit Generalization
Rocchio	Mahalanobis method	Explicit Generalization
Decision Trees	Isolation Trees	Explicit Generalization
Random Forest	Isolation Forest	Explicit Generalization
Rule-based	FP-Outlier	Explicit Generalization
Support Vector Machines	One-class Support Vector Machines	Explicit Generalization
Neural Networks	Replicator Neural Networks (Auto-Encoder)	Explicit Generalization
Matrix Factorization	Principle Component Analysis, Matrix Factorization	Explicit Generalization

Connection with Supervised Models (cont.)

Question:

- Is z-value test for outlier detection instance-based or explicit generalization?



Overview of Anomaly Detection Methods

Anomaly Detection Methods

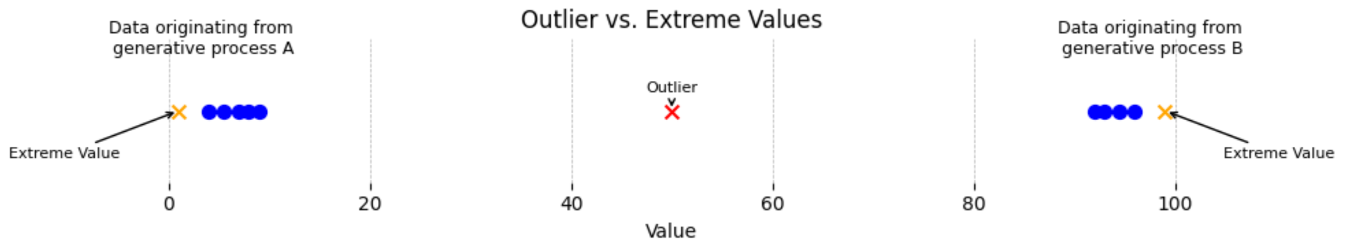
- Extreme-Value Analysis
- Probabilistic and Statistical Models
- Linear Models
- Proximity-Based Models
- Information-Theoretic Models
- Outlier Ensembles
- High-Dimensional Outlier Detection

Extreme-Value Analysis

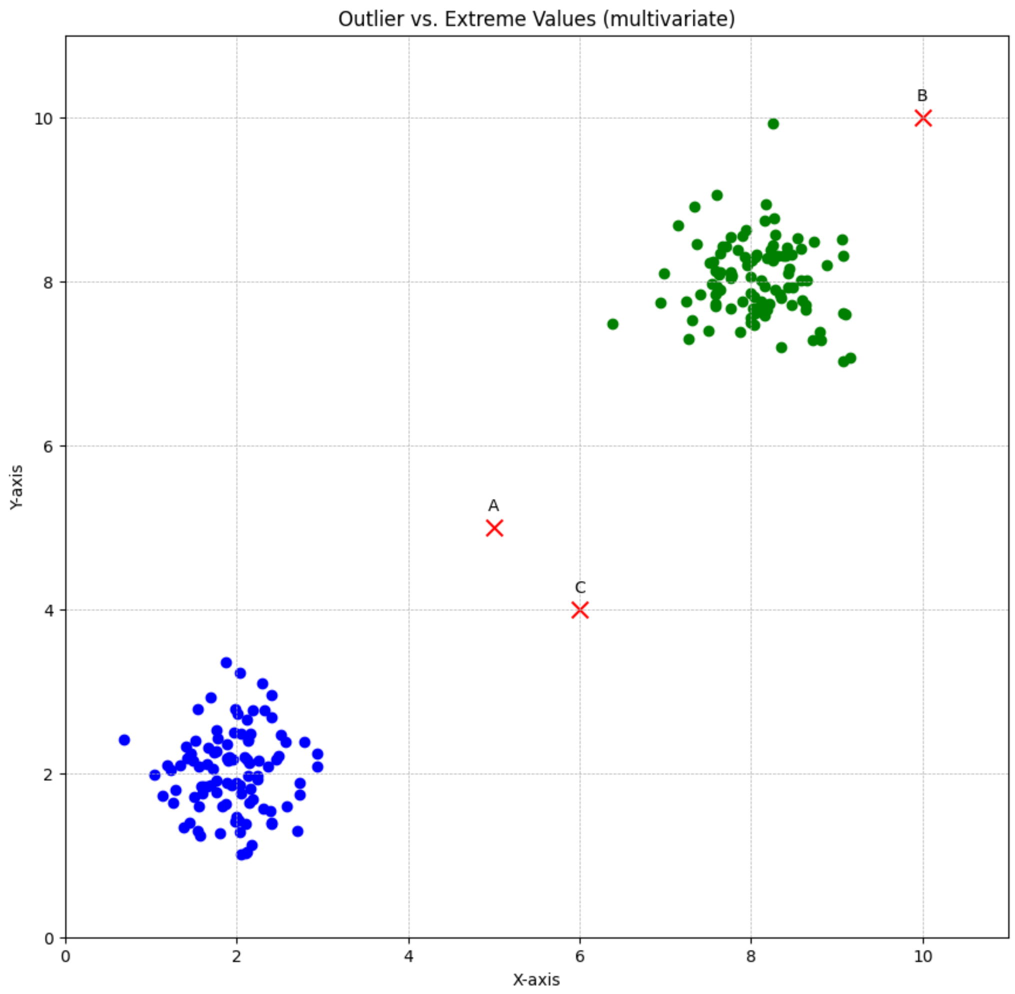
- Most basic form of outlier detection
- Assumption: Values that are too large or too small are outliers
- Determining the statistical tails of the underlying distribution
- z-value Test:
  - Normal distributed data: statistical interpretation
  - Arbitrary distributions: rather "heuristic idea of the outlier score"
- Extreme-value statistics is different from the general definition of outliers
- generative probabilities vs. extremity in value

Extreme-Value Analysis (cont.)

- [1, 4, 5.5, 7, 8, 9, 50, 92, 93, 94.5, 96, 99]

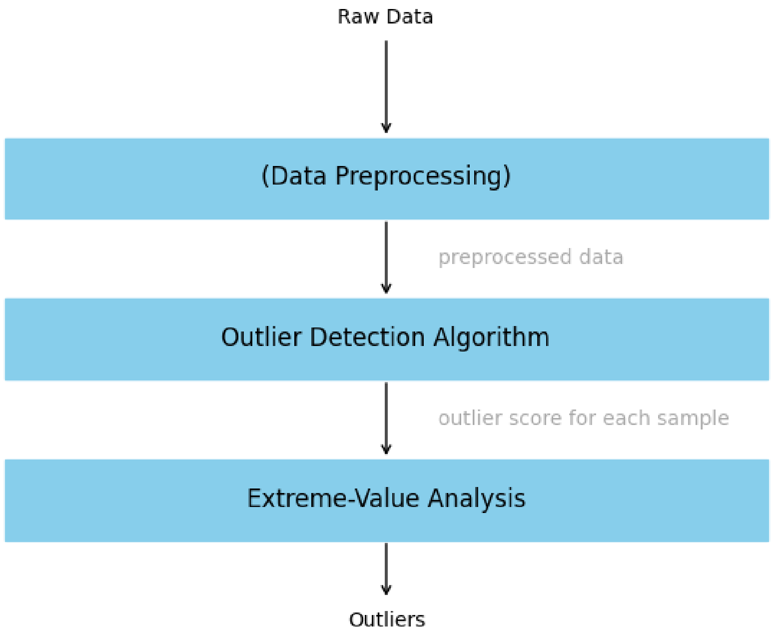


- Probabilistic and density-based models: Outlier is 50 (in line with outlier definition)
- Extreme-Value Analysis: 1 and 99 are outliers from an extreme-value perspective
- Special kinds of outliers even in the multivariate case



Extreme-Value Analysis (cont.)

- Extreme-value analysis for outlier detection algorithms as final step
- Algorithms quantify deviations of data from normal patterns as (univariate) outlier score
- Outlier scores can be analyzed with (multivariate) extreme-value methods



Probabilistic and Statistical Models

- Data is modeled as closed-form probability distribution
- The parameters of this model are learned
- Key assumption: choice of data distribution
- The likelihood fit of a data point to a generative model is the outlier score

Example:

- Gaussian PDF:

$$g(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- „Fit“ the parameters μ, σ to the data x_i

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \,, \sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{n-1}}$$

Probabilistic and Statistical Models (cont.)

Example (cont.):

Approach:

1. Choose Gaussian (Normal) Distribution Probability Density Function
2. "Fit" parameters to dataset
3. Calculate the likelihood fit for a data point
4. Convert the likelihood fit to an outlier score (e.g. negative log likelihood)

Example:

1. Gaussian PDF:

$$g(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

2. „Fit" the parameters μ, σ to the data x_i

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \,, \sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{n-1}}$$

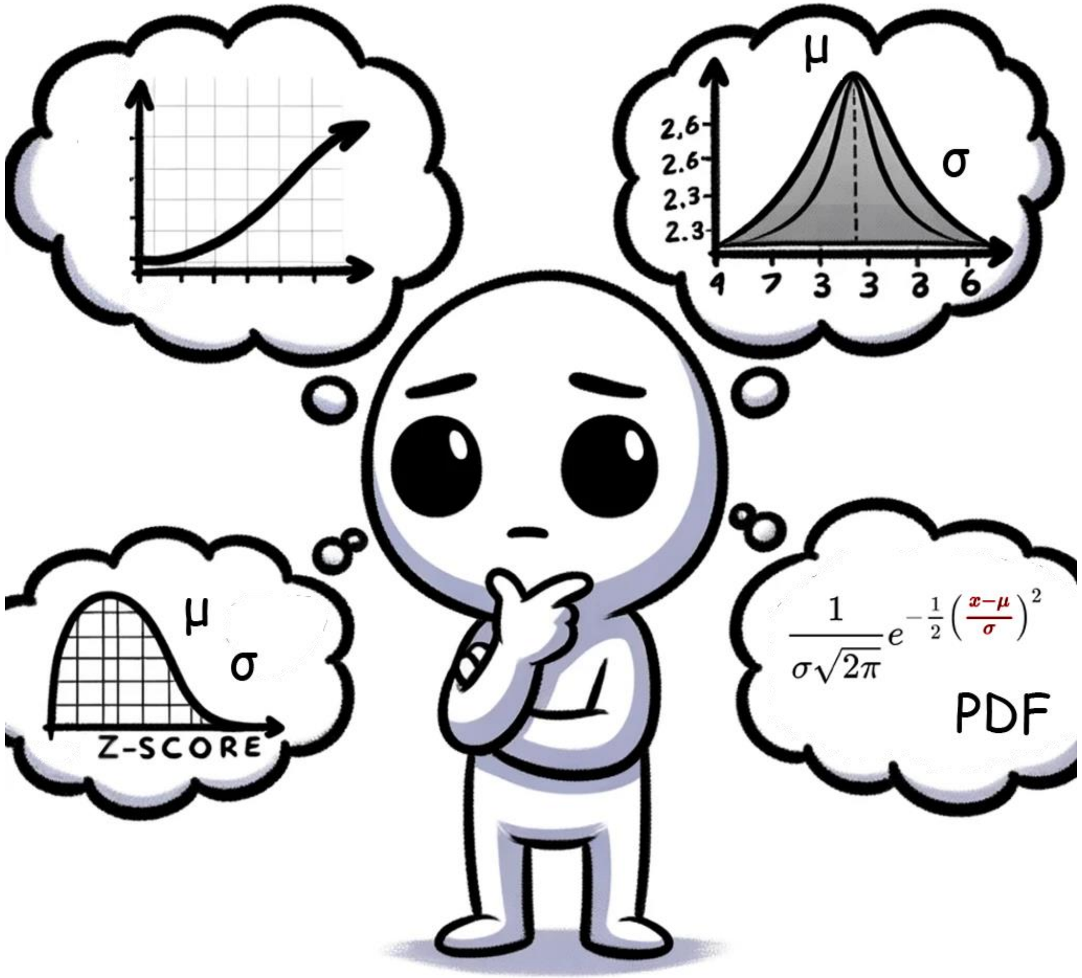
3. Likelihood for $\hat{x}$: $g(\hat{x}; \mu, \sigma)$
4. $\mathrm{NLL}(\hat{x}) = -\log(g(\hat{x}; \mu, \sigma))$

Probabilistic and Statistical Models (cont.)

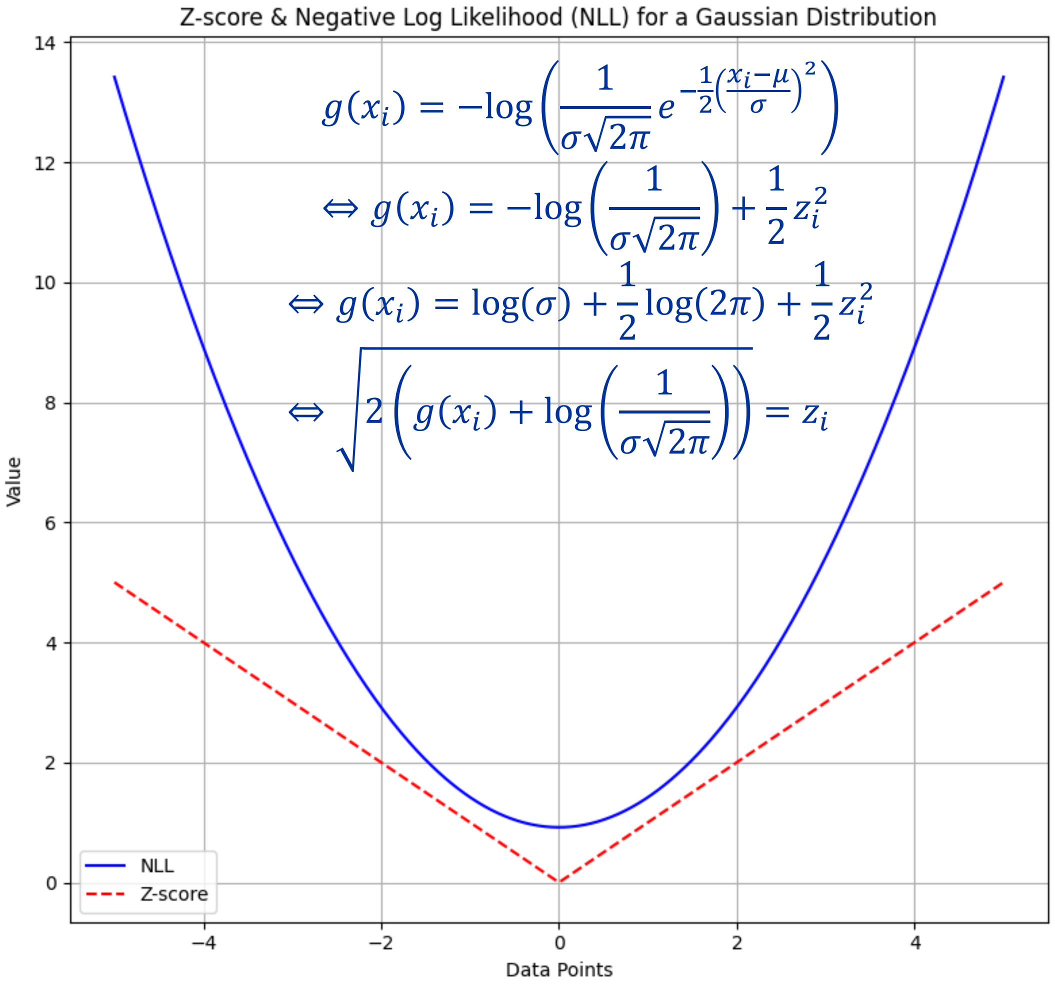
Example (cont.):

Approach:

1. Choose Gaussian (Normal) Distribution Probability Density Function
2. “Fit” parameters to dataset
3. Calculate the likelihood fit for a data point
4. Convert the likelihood fit to an outlier score (e.g. negative log likelihood)



Probabilistic and Statistical Models (cont.)



$$g(x_i) = -\log\left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2}\right)$$

$$\Leftrightarrow g(x_i) = -\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \frac{1}{2} z_i^2$$

$$\Leftrightarrow g(x_i) = \log(\sigma) + \frac{1}{2}\log(2\pi) + \frac{1}{2} z_i^2$$

$$\Leftrightarrow \sqrt{2\left(g(x_i) + \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)\right)} = z_i$$

```
import numpy as np
import matplotlib.pyplot as plt

mu = 0
sigma = 1

x = np.linspace(-5, 5, 400)      # Generate data points

# Calculate Z-scores for these data points
z_scores = abs(x - mu) / sigma

# Calculate the PDF likelihood for these data points
pdf_likelihood = (1 / (np.sqrt(2 * np.pi) * sigma)) * \
           np.exp(-0.5 * ((x - mu) / sigma)**2)

plt.figure(figsize=(8, 7.5))
plt.plot(x, -np.log(pdf_likelihood), label='NLL', color='blue’)
plt.plot(x, z_scores, label='Z-score', color='red', linestyle='--’)
plt.xlabel('Data Points’)
plt.ylabel('Value’)
plt.title('Z-score & Negative Log Likelihood (NLL) for a Gaussian
Distribution’)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

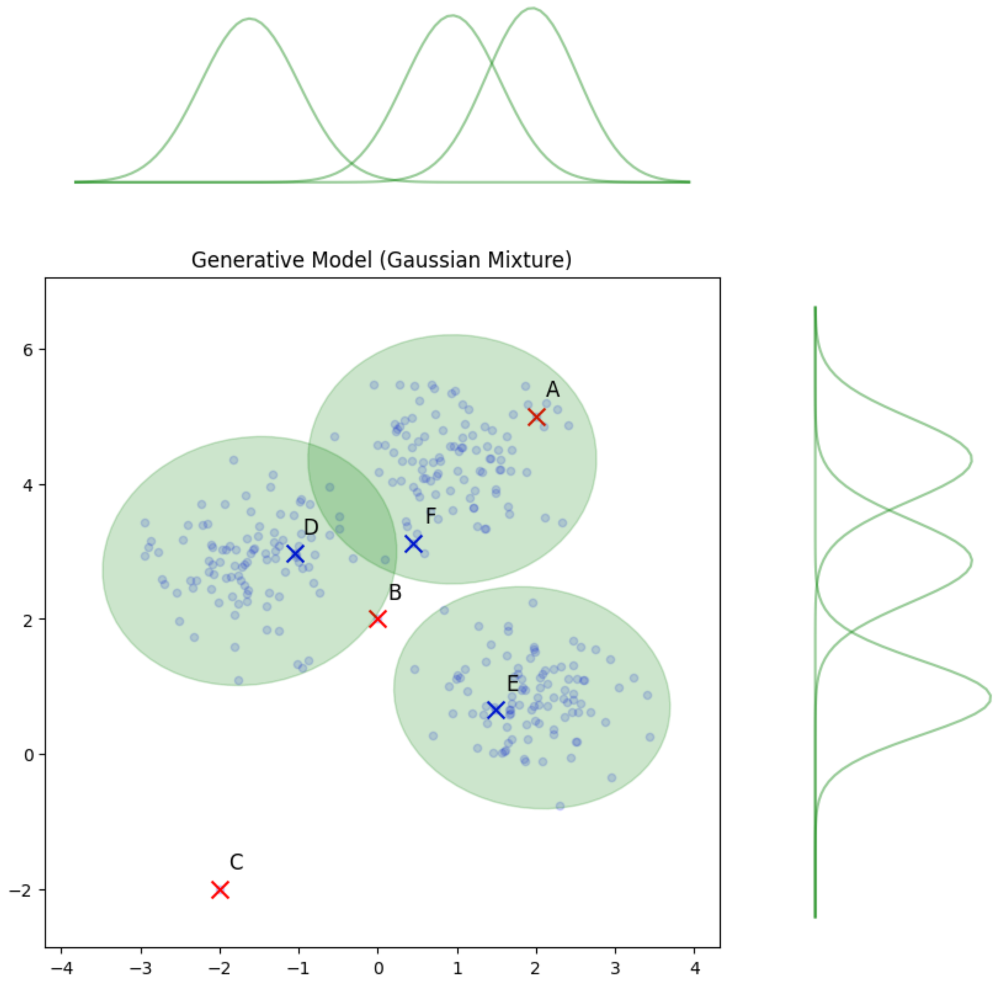
Probabilistic and Statistical Models (cont.)

Gaussian Mixture Models

- Probabilistic model
- Assumption: Data is generated from a mixture of Gaussians
- Data is described as combination of K Gaussians

$$p(x;\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k g(x;\mu_k,\Sigma_k)$$

- Parameters are learned via EM algorithm



Reynolds, Douglas A. "Gaussian mixture models." Encyclopedia of biometrics 741.659-663 (2009).

Gaussian Mixture Modelling Example:
http://localhost:8888/notebooks/AD02-S94-GMM_Example.ipynb