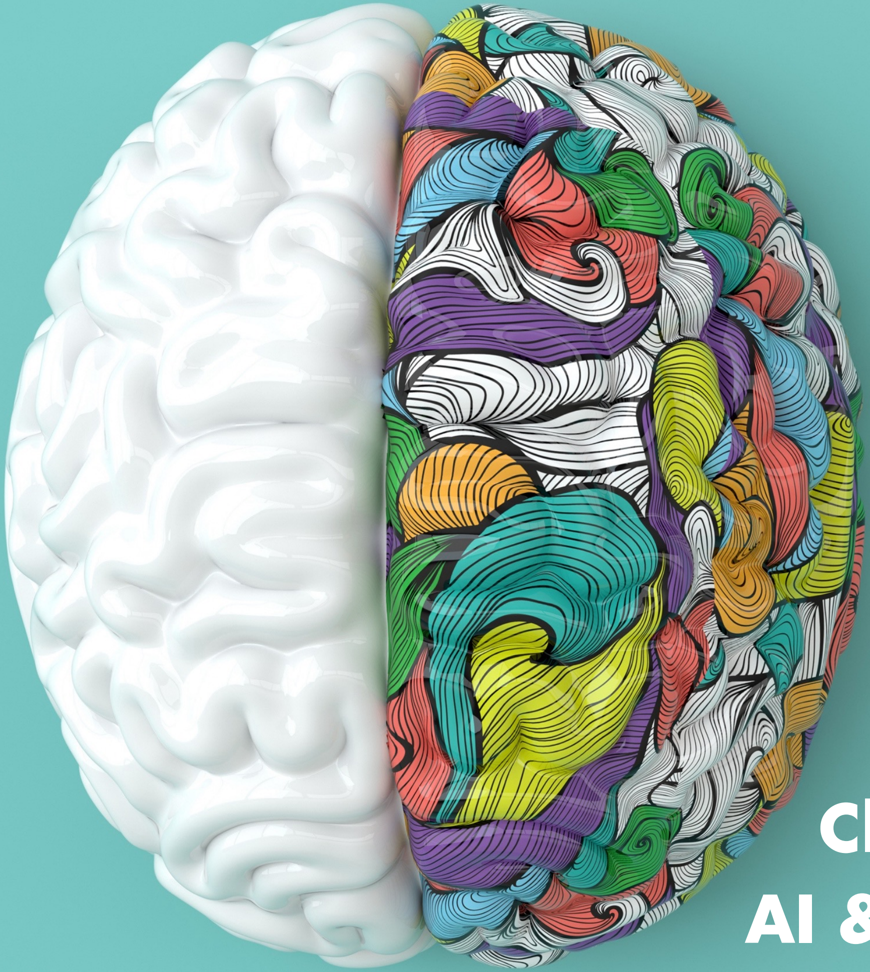
#WISSENTEILEN

@_openKnowledge | @mobileLarson

AI and Architecture

Lars Röwekamp | open knowledge GmbH



Cloud <

AI & ML <

Architecture <

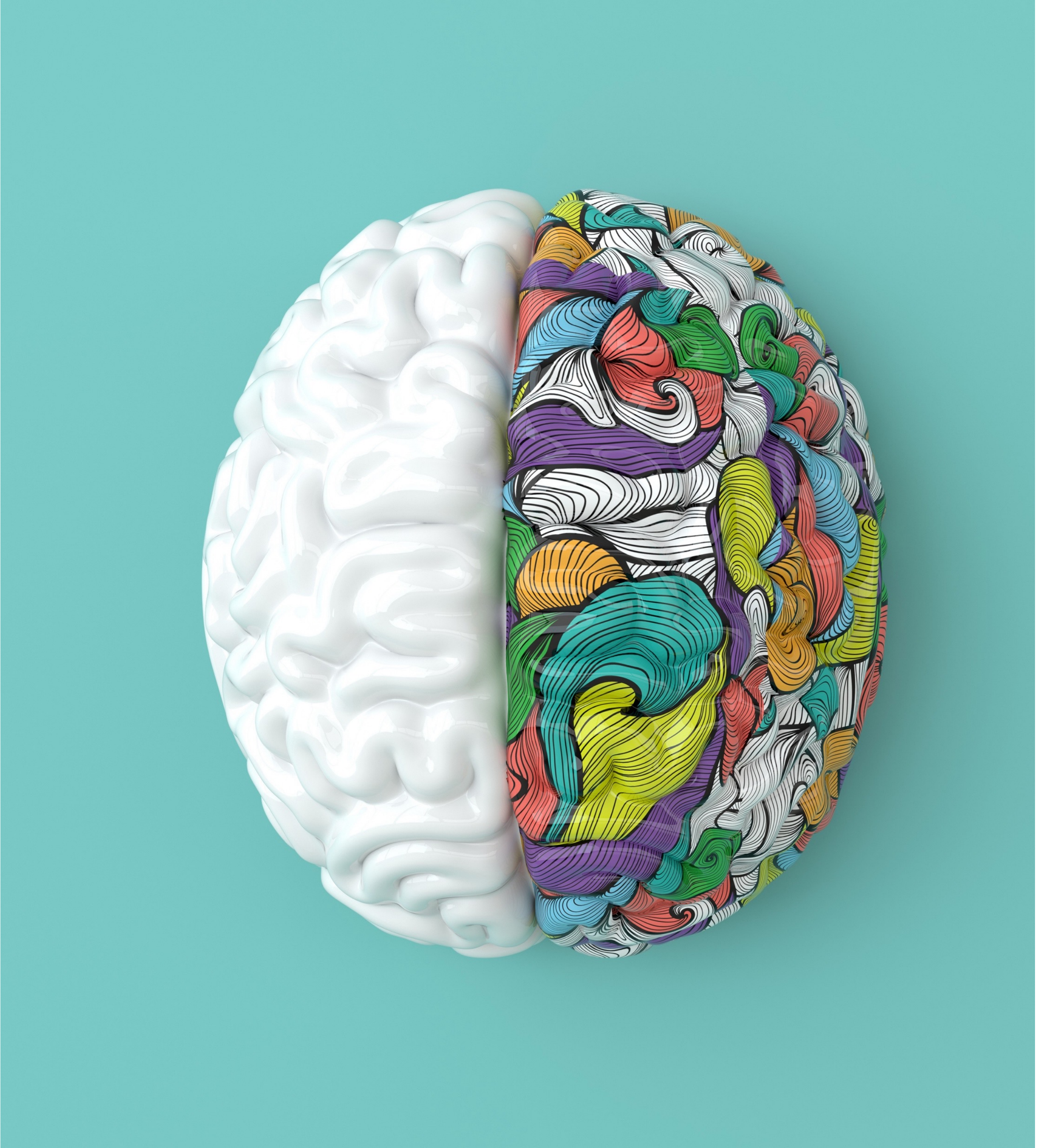
Microservices <

Lars Röwekamp

@mobile Larson

CIO New Technologies
OPEN KNOWLEDGE

OPEN
KNOW
LEDGE



How to use Artificial Intelligence?

OPEN
KNOW
LEDGE

ML Voodoo

Data

Data Data

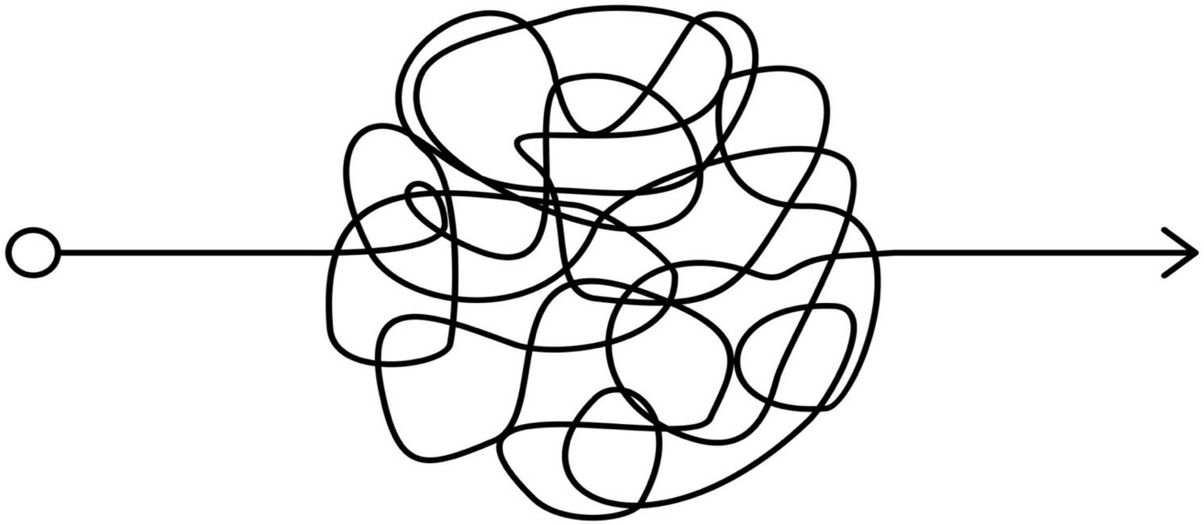
Data Daten

Data

Data

Data

Data



42

ML Voodoo Maturity Level

Data

Data

Data

Data

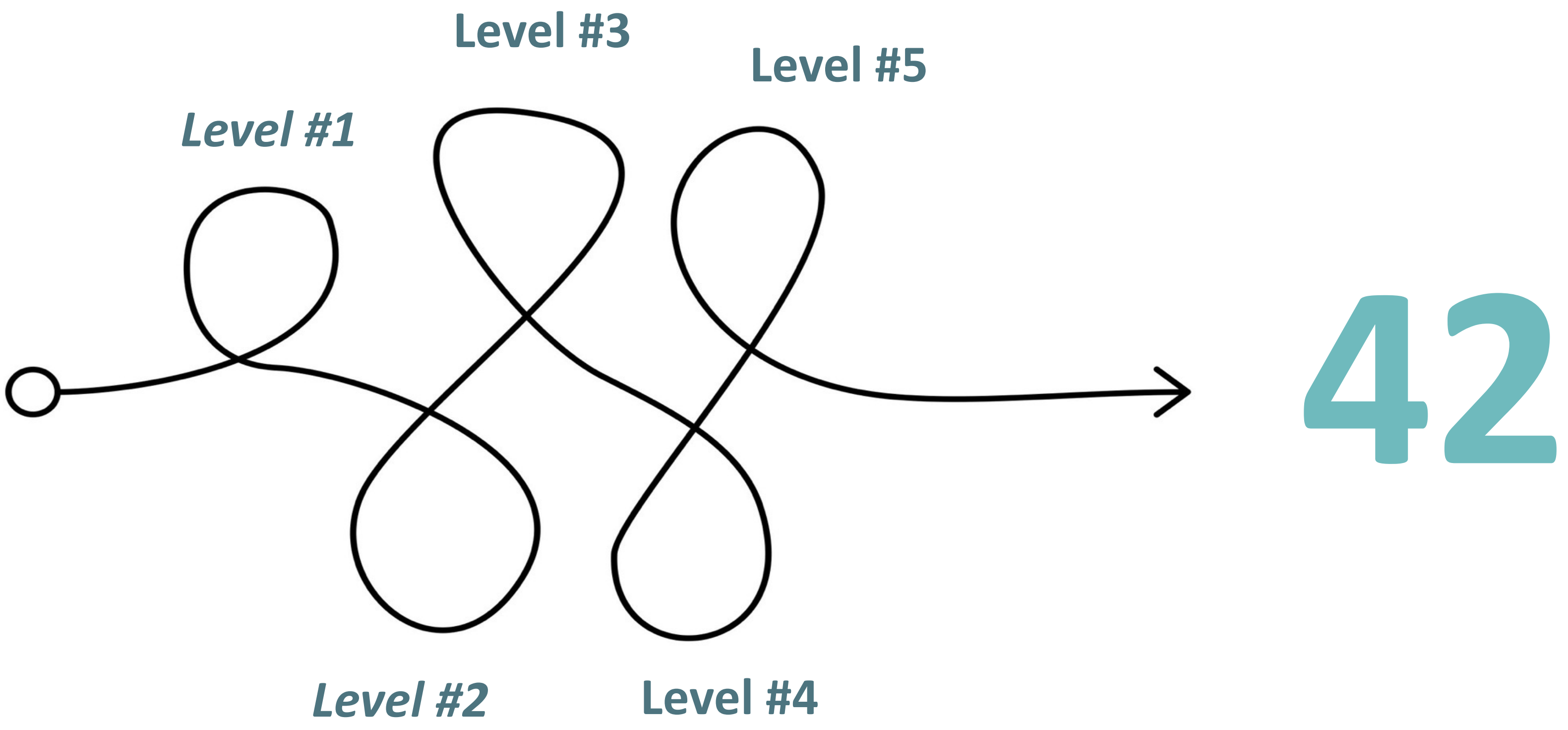
Daten

Data

Data

Data

Data



Level #1

Level #2

Level #3

Level #4

Level #5

42

Level #1:

Integration of 3rd Party APIs

Level #2:

Usage of 3rd Party ML Models

Level #3:

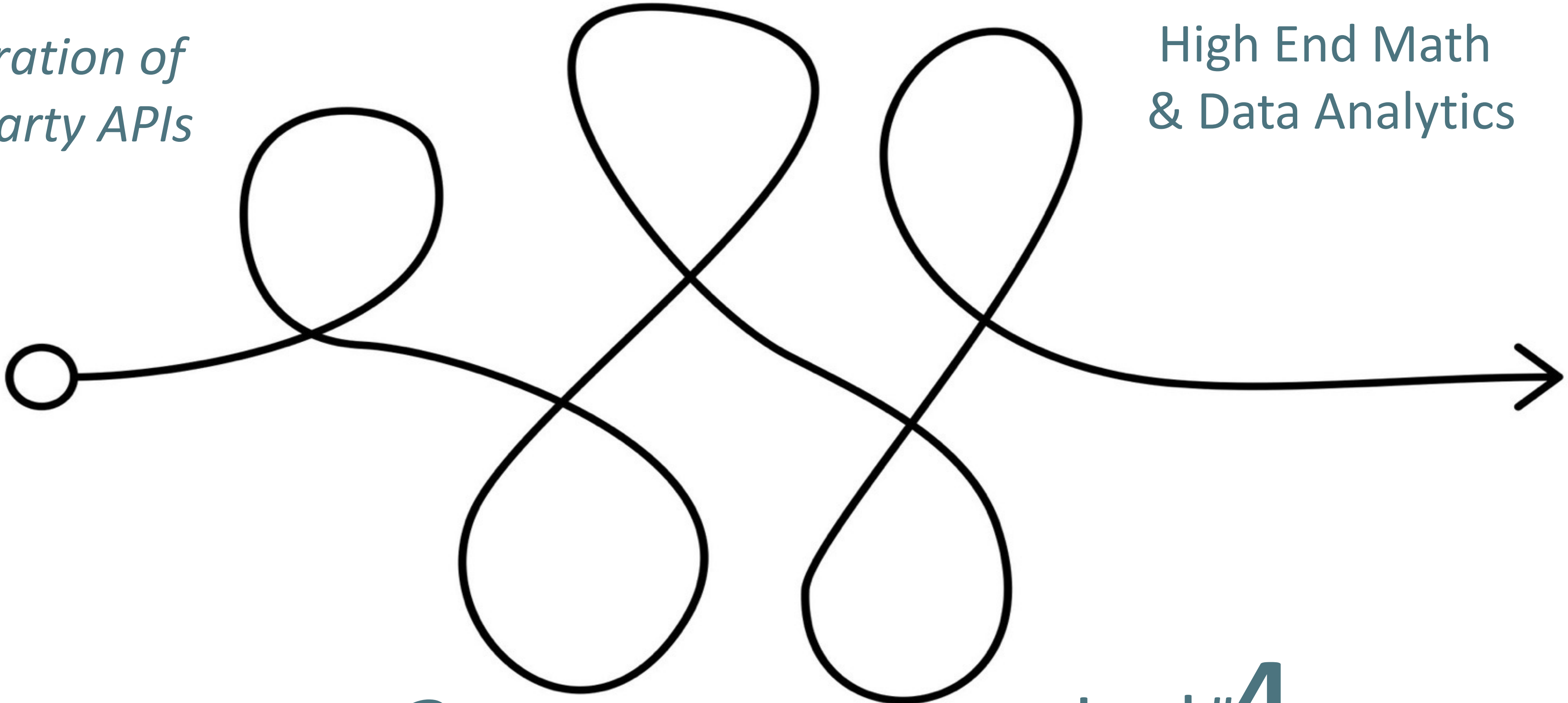
Productive use of ML

Level #4:

Design & Implementation of own ML Models

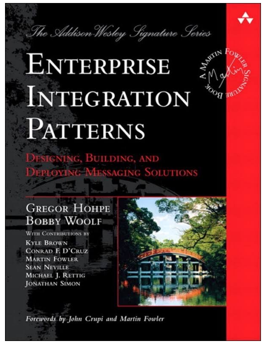
Level #5:

High End Math & Data Analytics



Level #1:

Integration of 3rd Party APIs



The Addison-Wesley Signature Series

ENTERPRISE INTEGRATION PATTERNS

DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS

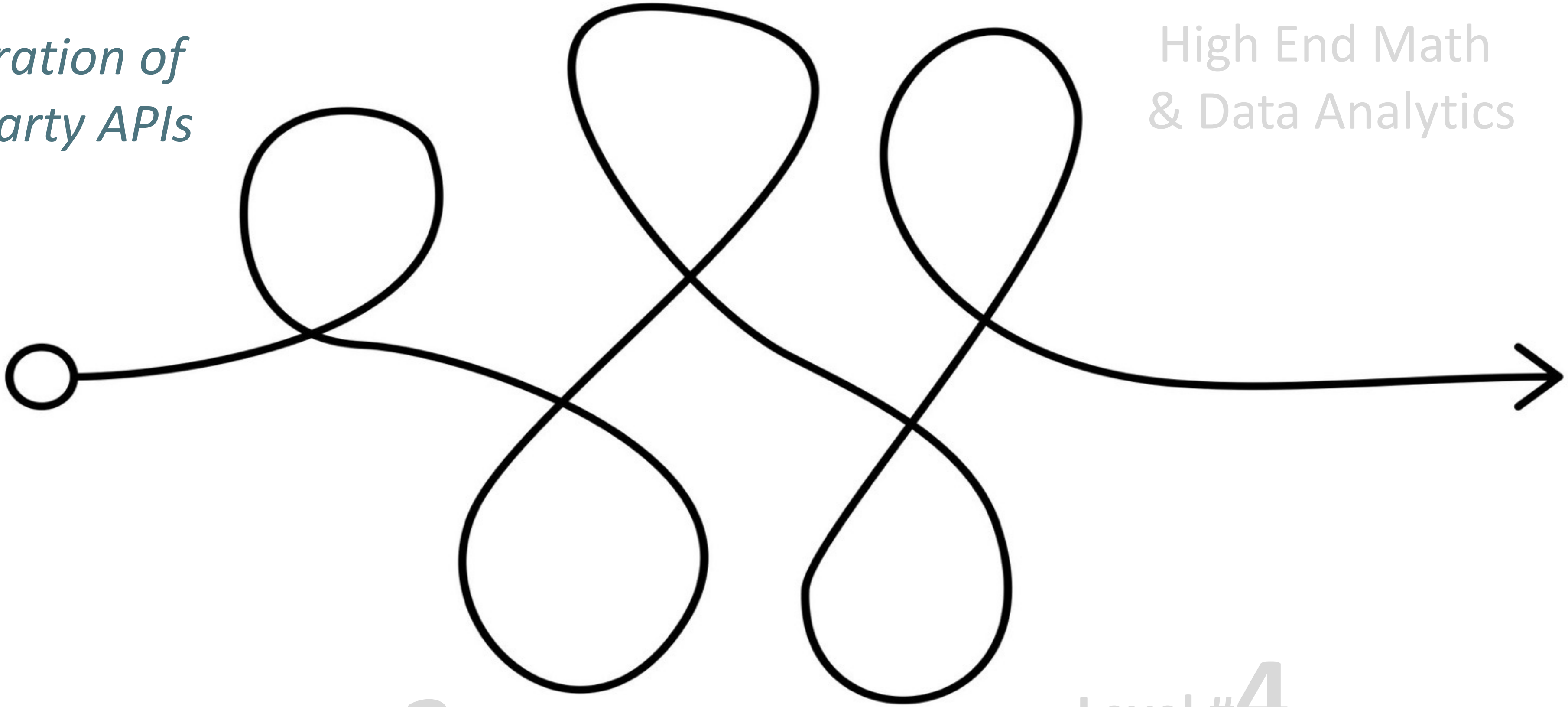
GREGOR HOHPE
BOBBY WOOLF

WITH CONTRIBUTIONS BY
KYLE BROWN
CONRAD F. D'CRUZ
MARTIN FOWLER
SEAN NEVILLE
MICHAEL J. RETTIG
JONATHAN SIMON

Forewords by John Crupi and Martin Fowler

A MARTIN FOWLER SIGNATURE BOOK

„Read this!“



Level #2:

Usage of 3rd Party ML Models

Level #3:

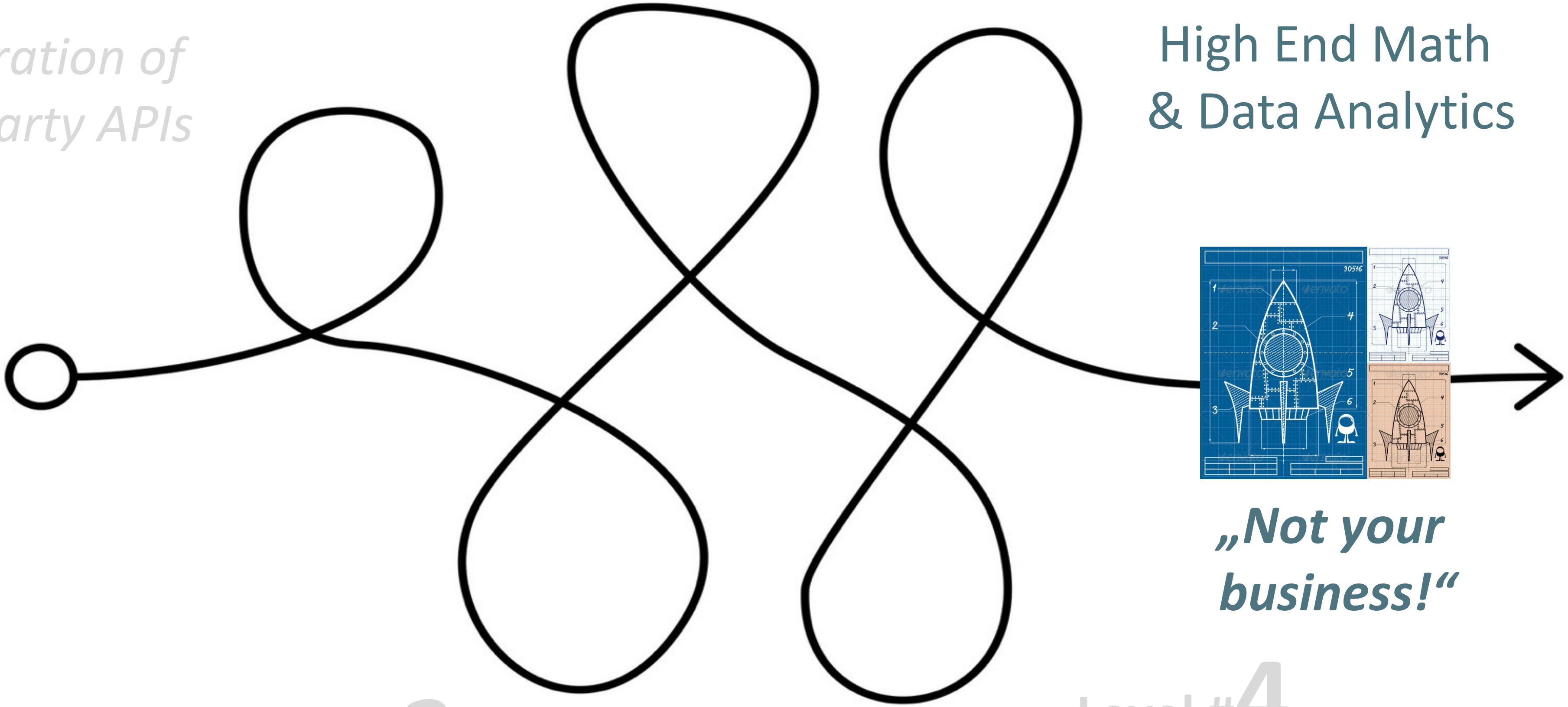
Productive use of ML

Level #4:

Design & Implementation of own ML Models

Level #5:

High End Math & Data Analytics



Level #1:

Integration of 3rd Party APIs

Level #2:

Usage of 3rd Party ML Models

Level #3:

Productive use of ML

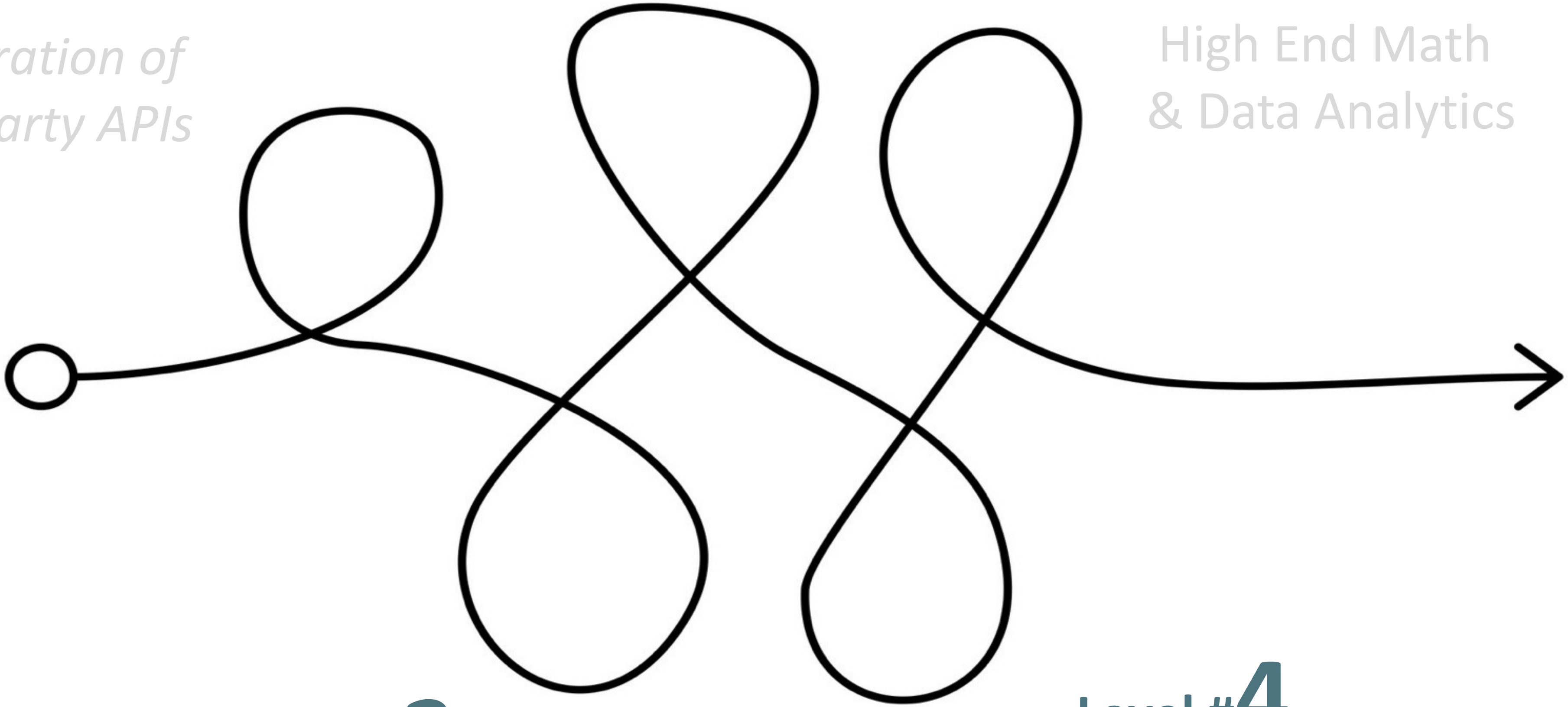
Level #4:

Design & Implementation of own ML Models

Level #5:

High End Math & Data Analytics

„Not your business!“



Level #1:

Integration of 3rd Party APIs

Level #2:

Usage of 3rd Party ML Models

Level #3:

Productive use of ML

Level #4:

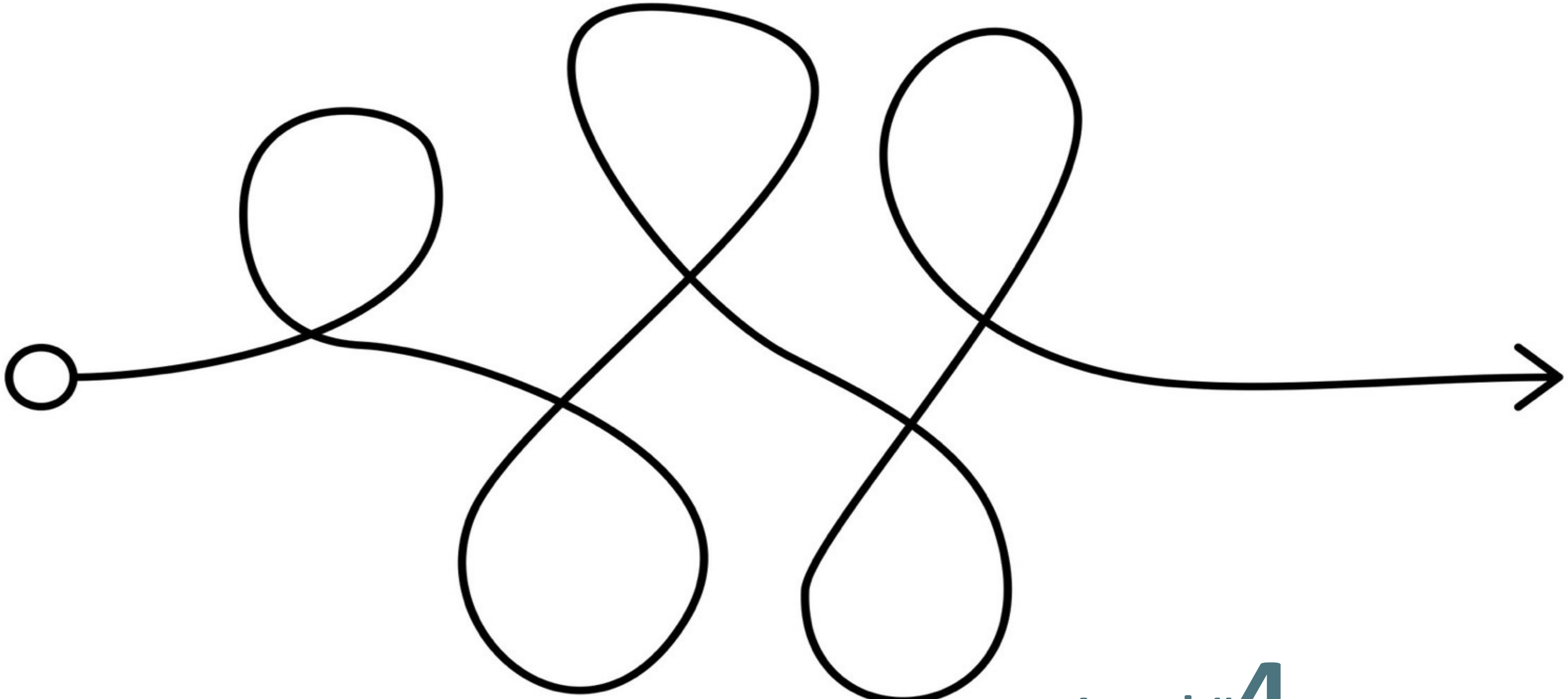
Design & Implementation of own ML Models

Level #5:

High End Math & Data Analytics

Level #3:

PRODUCTION



Level #2:

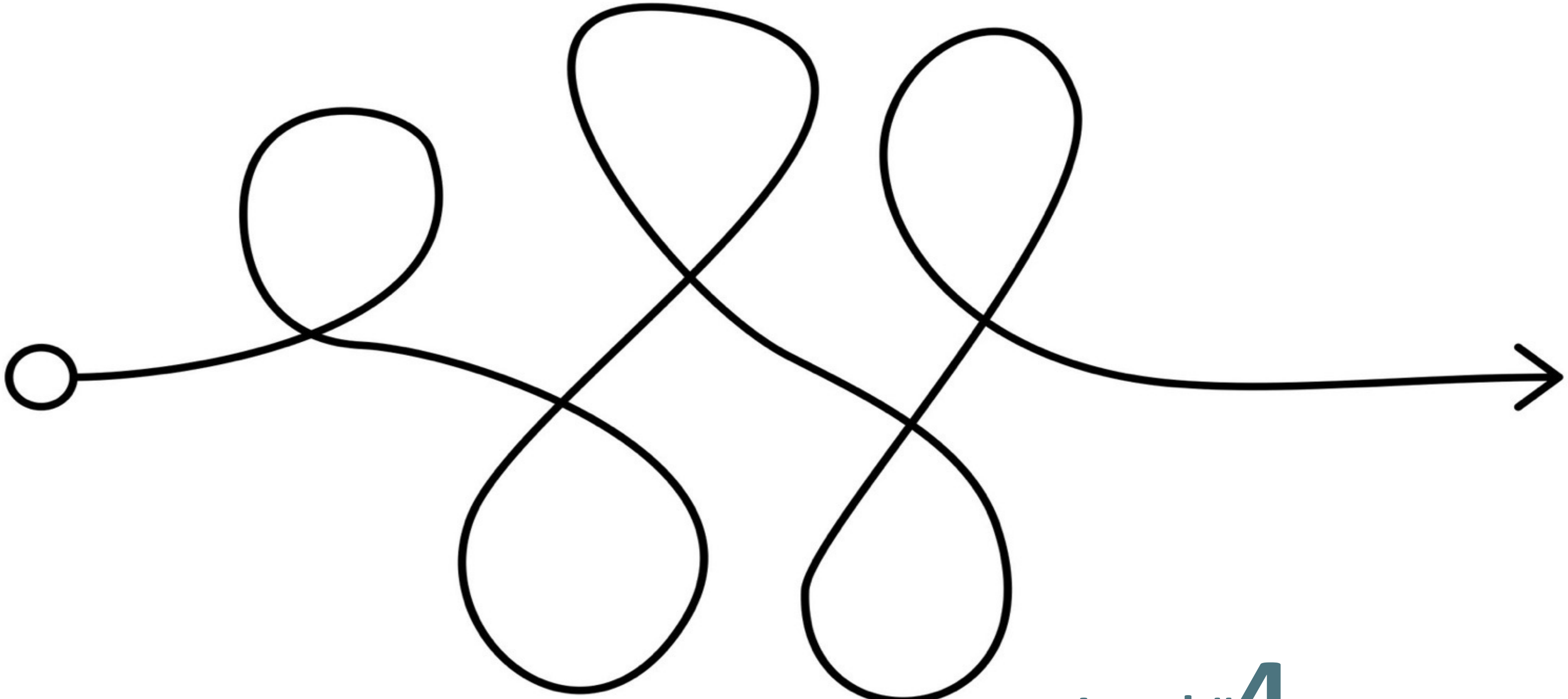
EXPLORATION

Level #4:

RESEARCH

Level #3:

PRODUCTION



Level #2:

EXPLORATION

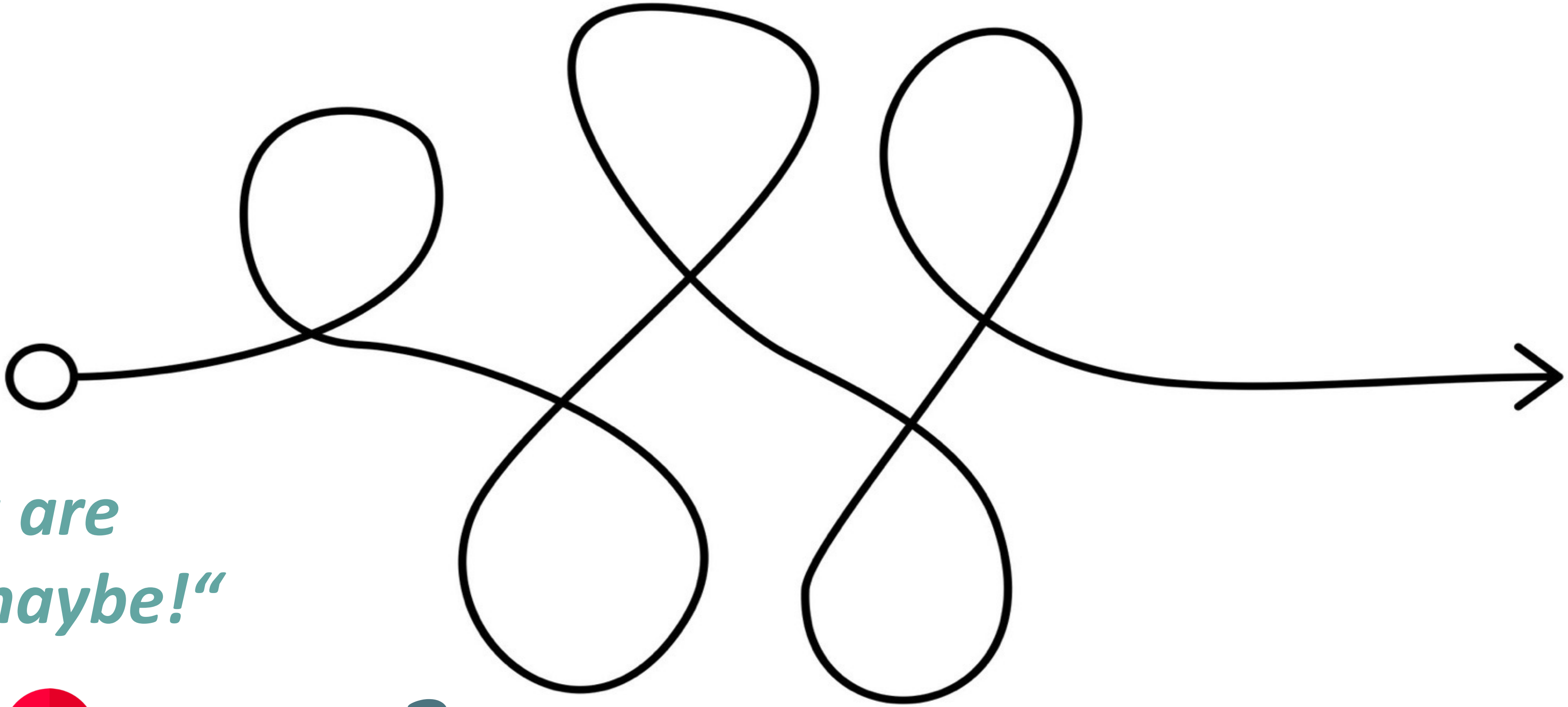
Level #4:

~~RESEARCH~~

not really

Level #3:

PRODUCTION



„You are here, maybe!“

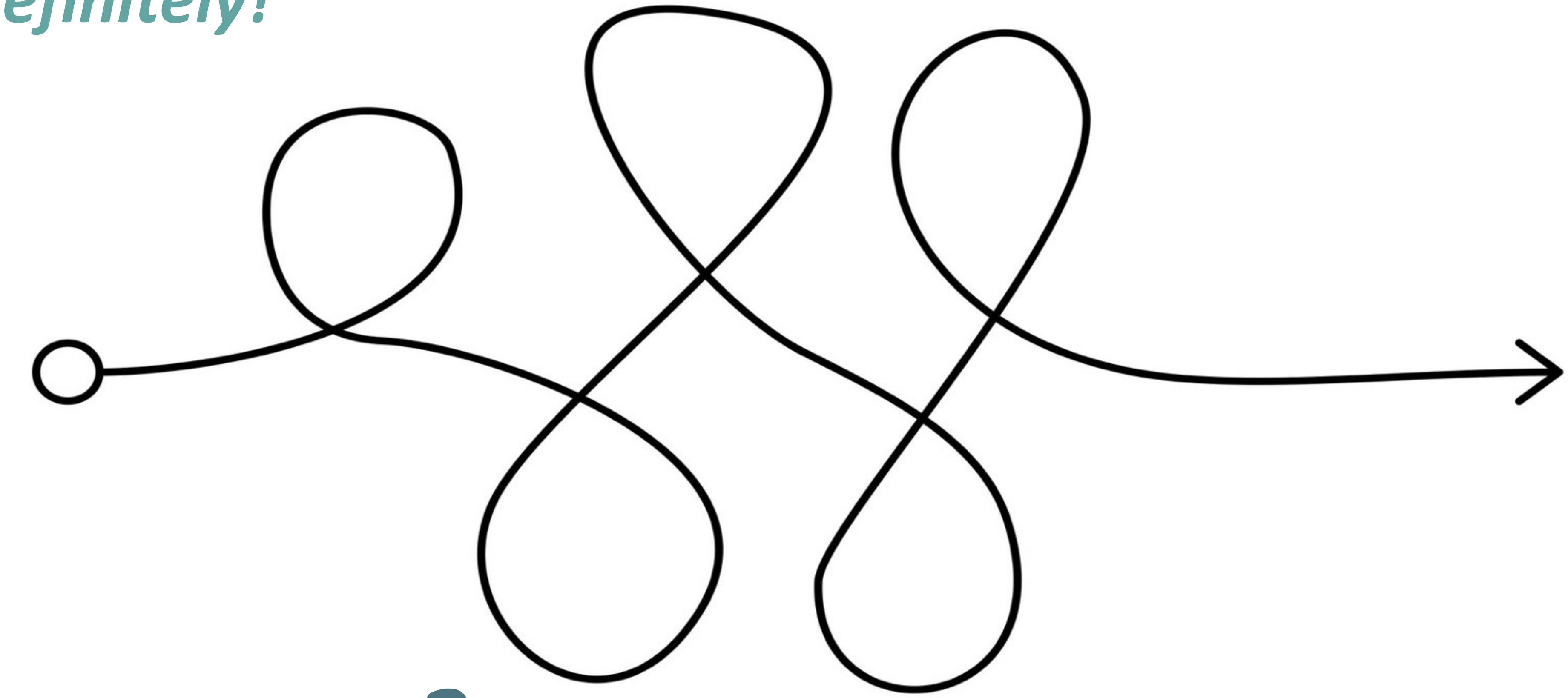
Level #2:

EXPLORATION

„But, you want to be here, definitely!“

Level #3:

PRODUCTION



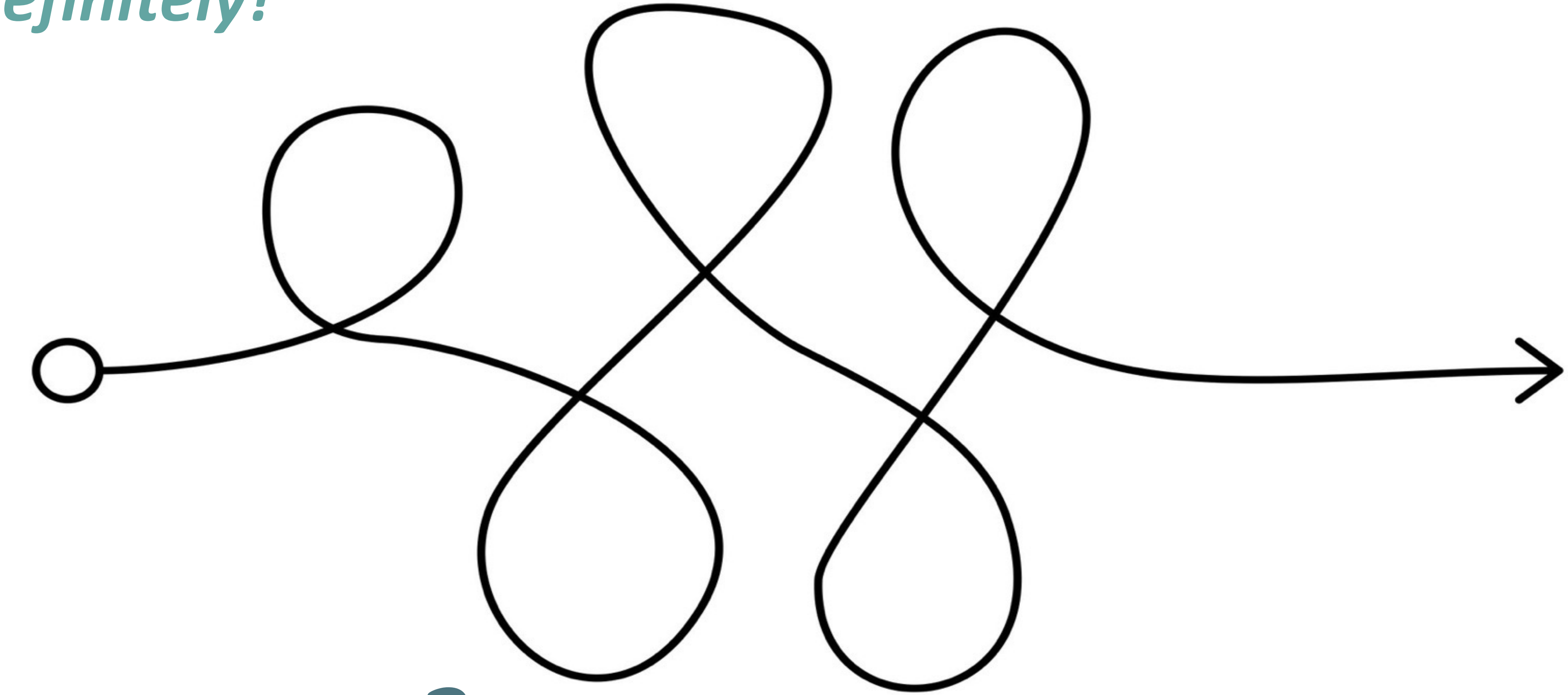
Level #2:

EXPLORATION

„But, you ~~want~~ have to be here, definitely!“

Level #3:

PRODUCTION



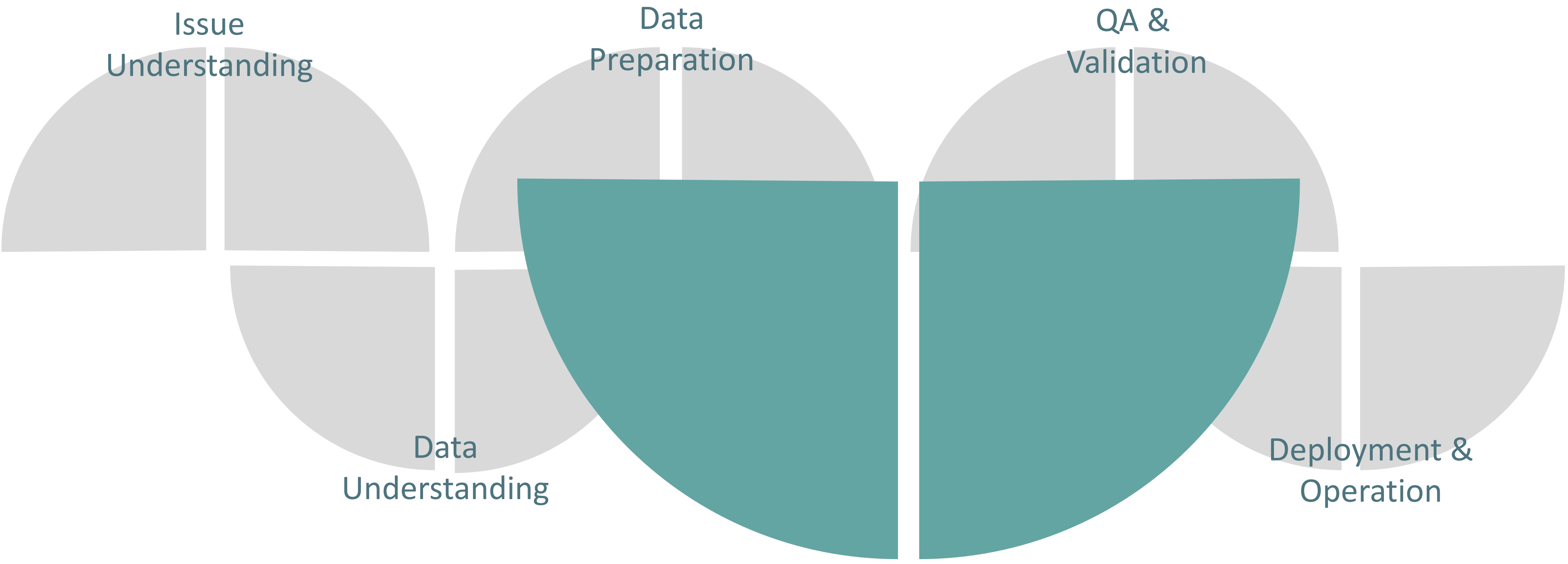
Level #2:

EXPLORATION

OFFENKUNDIGGUT

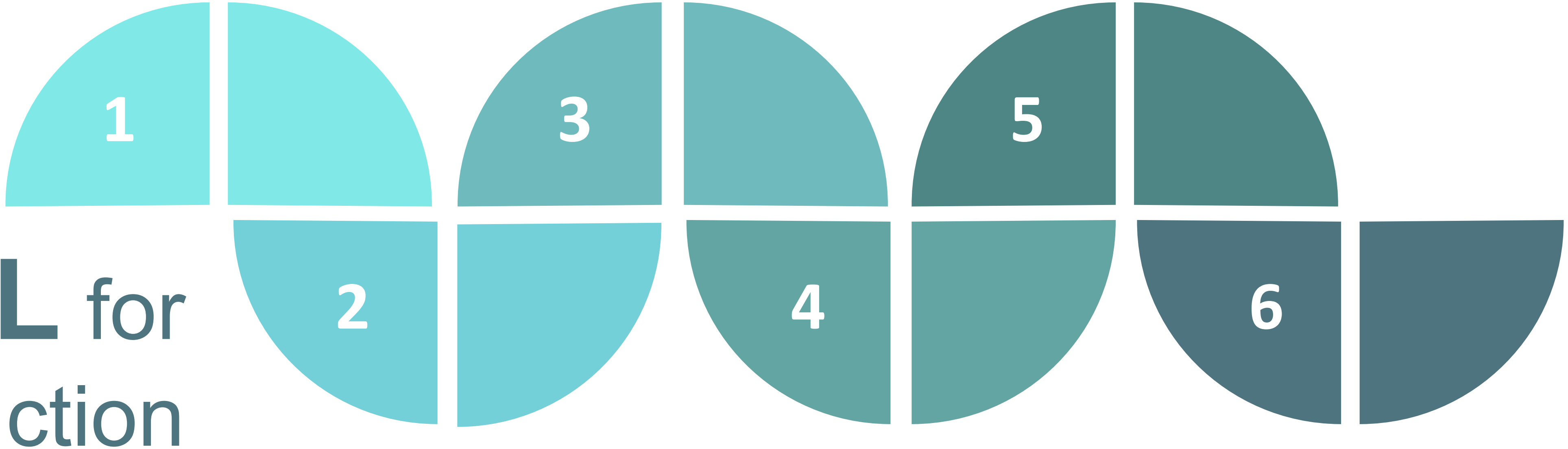
OPEN KNOW LEDGE

ML Voodoo Level 2 „only“



Modeling

ML for Production



Business Understanding

Data Understanding

Data Preparation

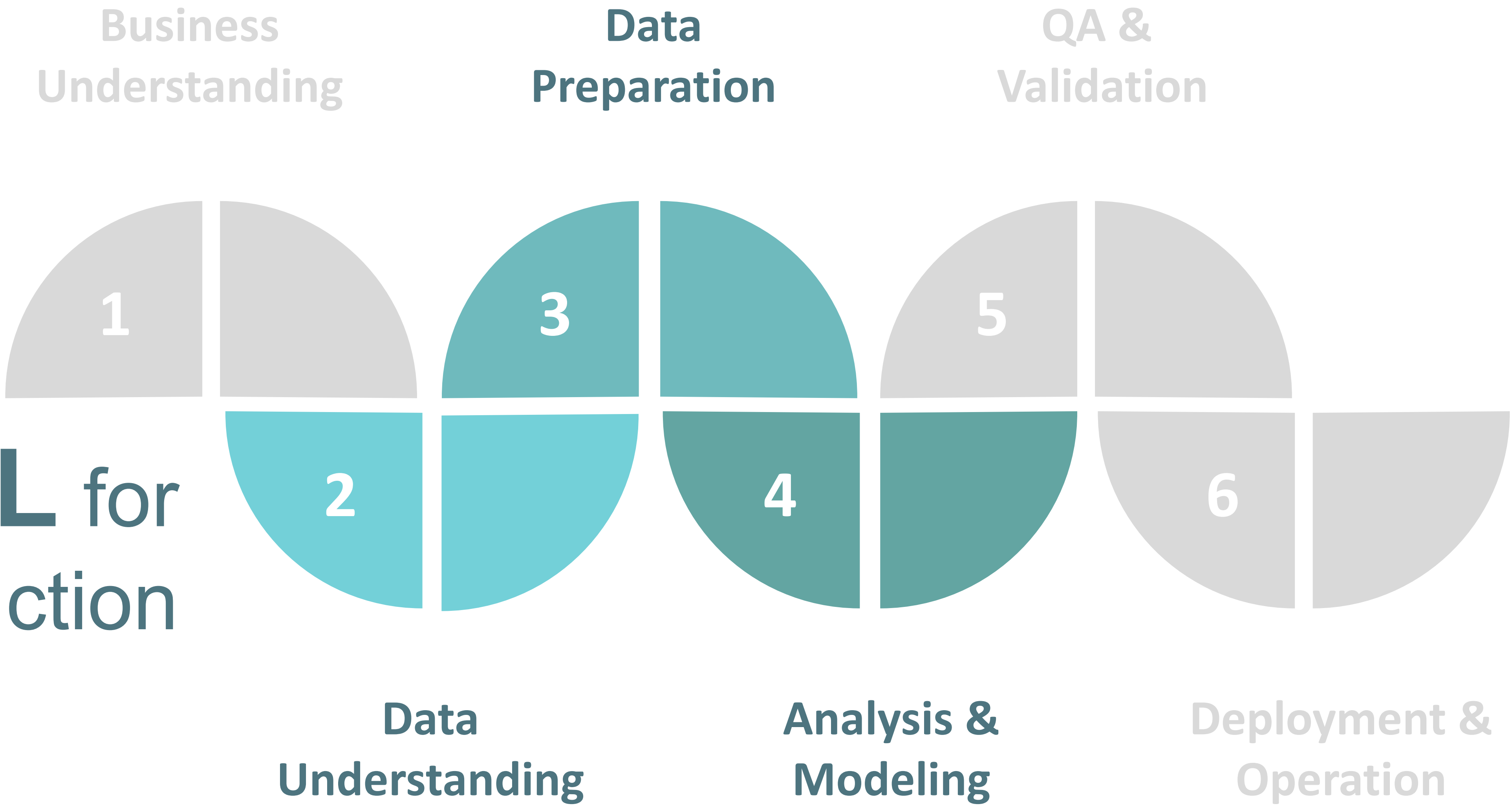
Analysis & Modeling

QA & Validation

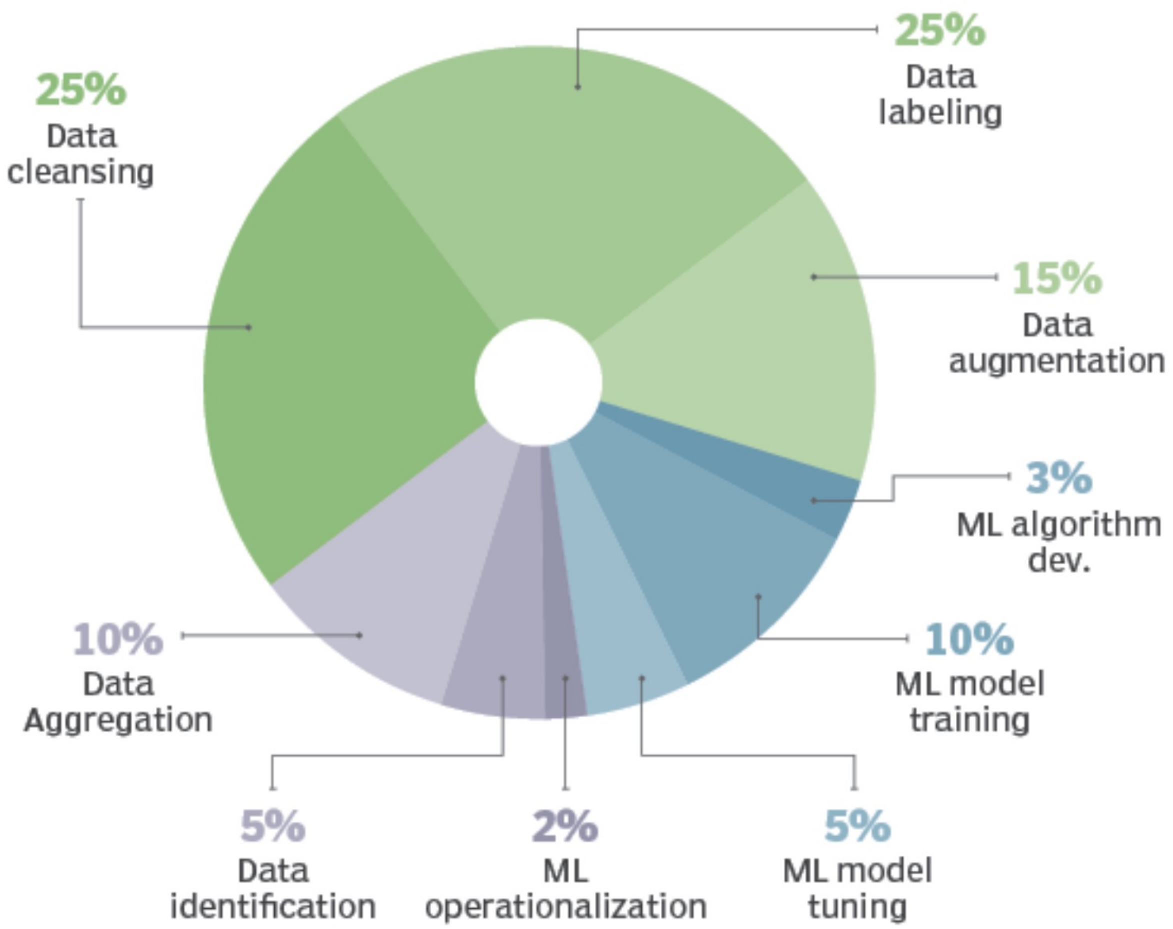
Deployment & Operation

*CRISP-DM: CRoss Industry Standard Process for Data Mining

ML for Production



*CRISP-DM: CRoss Industry Standard Process for Data Mining



Quelle: https://www.techtarget.com/searchenterpriseai/feature/Data-preparation-for-machine-learning-still-requires-humans

ML for Production by Example: **Sales Forecast***

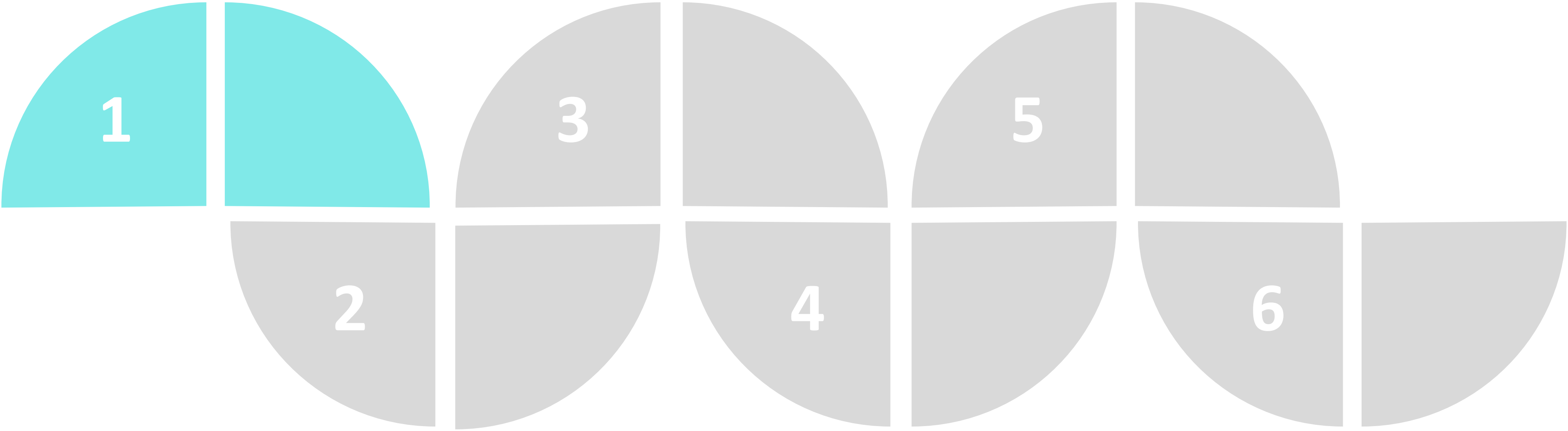
*Grocery Shopping Chain

ML for Production by Example …

SALES FORECAST

Based on **historical data** from several stores, sales figures are to be **predicted for the future**.

Business Understanding



„Sorry, what was the Question?“

Problem Statement & Target Definition

Business Understanding
by Example

Problem Statement - SALES FORECAST

*„ Often there is **too much or too little goods** in stock.“*

*„Lots of perishable **goods are thrown away** in the evening.“*

*„Marketing **campaigns are not effective** (enough).“*

*„**Turnover** is okay, but it's **not predictable**.“*

*„**Staffing** is often **less than optimal**.“*

…

Business Understanding
by Example

Business Goals- SALES FORECAST

A **sales forecast** is to be created, which supports the **planning of sales** per store*, as well as the **planning of marketing campaigns****.

* ... and thus indirectly also procurement and warehousing

** ... such as promotions or introduction of new product

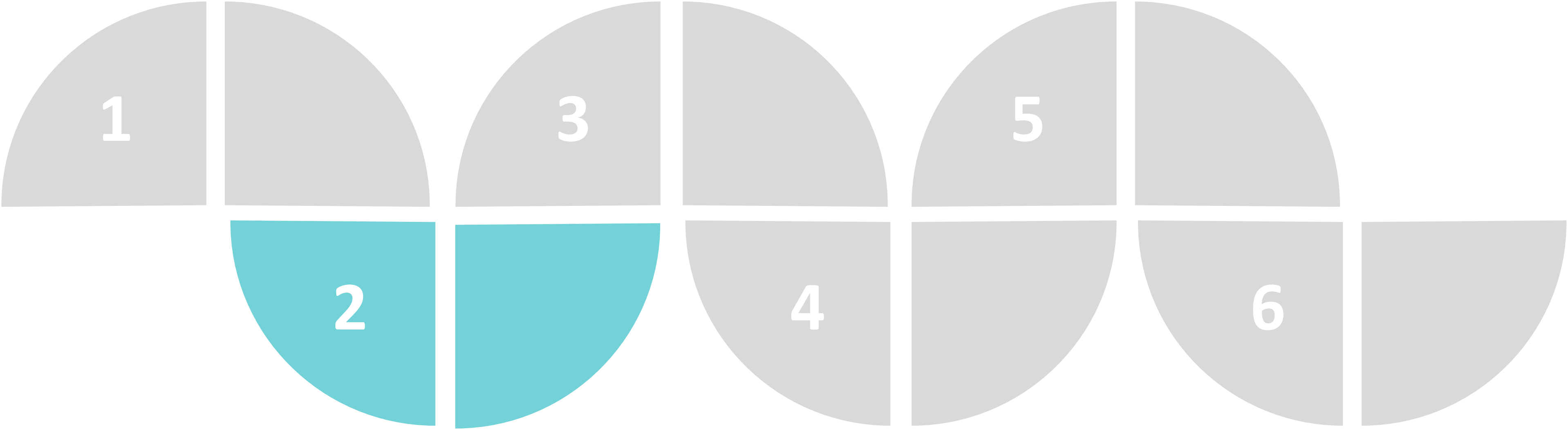
Business Understanding
by Example

Analytical Goals - SALES FORECAST

A machine learning solution should predict the **sales per store** for the **period of 6 weeks**, with an **accuracy of [X]**%.

ATTENTION:
A concrete value is important here. Unfortunately, "as accurate as possible" is of little help for later validation and quality analysis of the model!

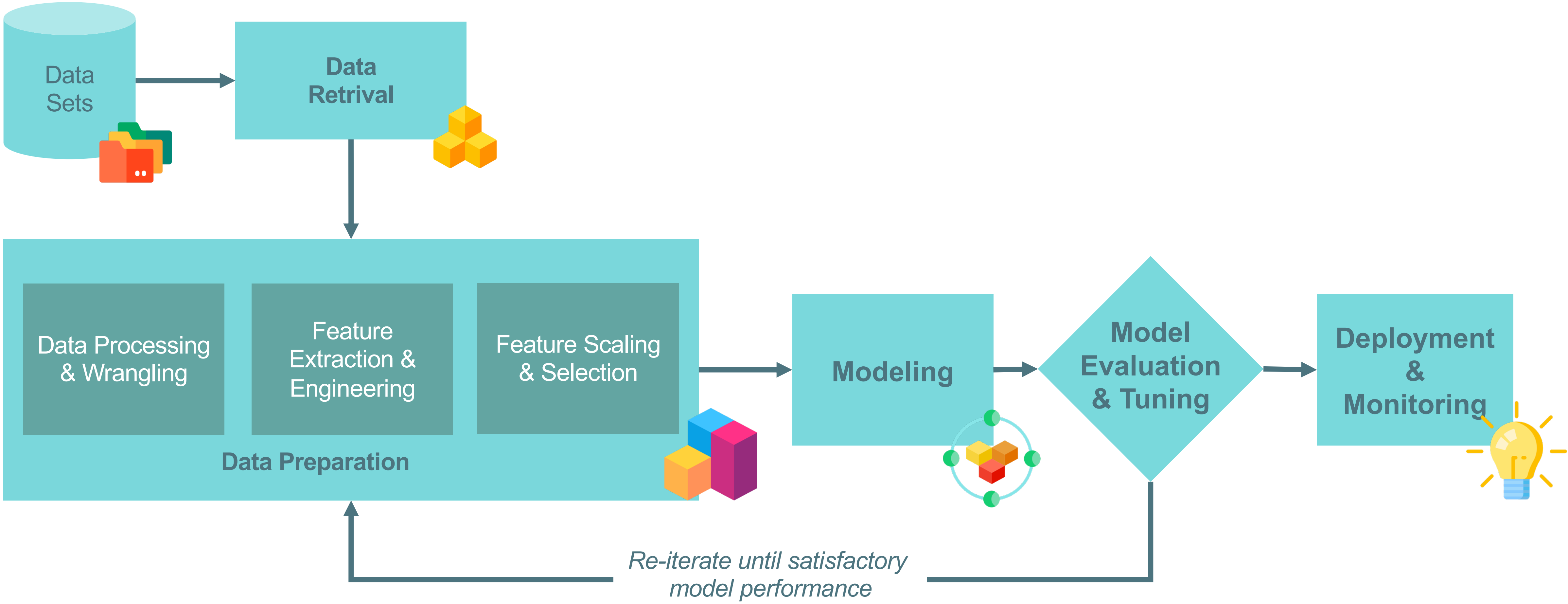
Data Understanding



„The Truth is in the Data!“

Collect, examine & evaluate

ML4Prod Process



Data Sets

Data Retrival

Data Processing & Wrangling

Feature Extraction & Engineering

Feature Scaling & Selection

Data Preparation

Modeling

Model Evaluation & Tuning

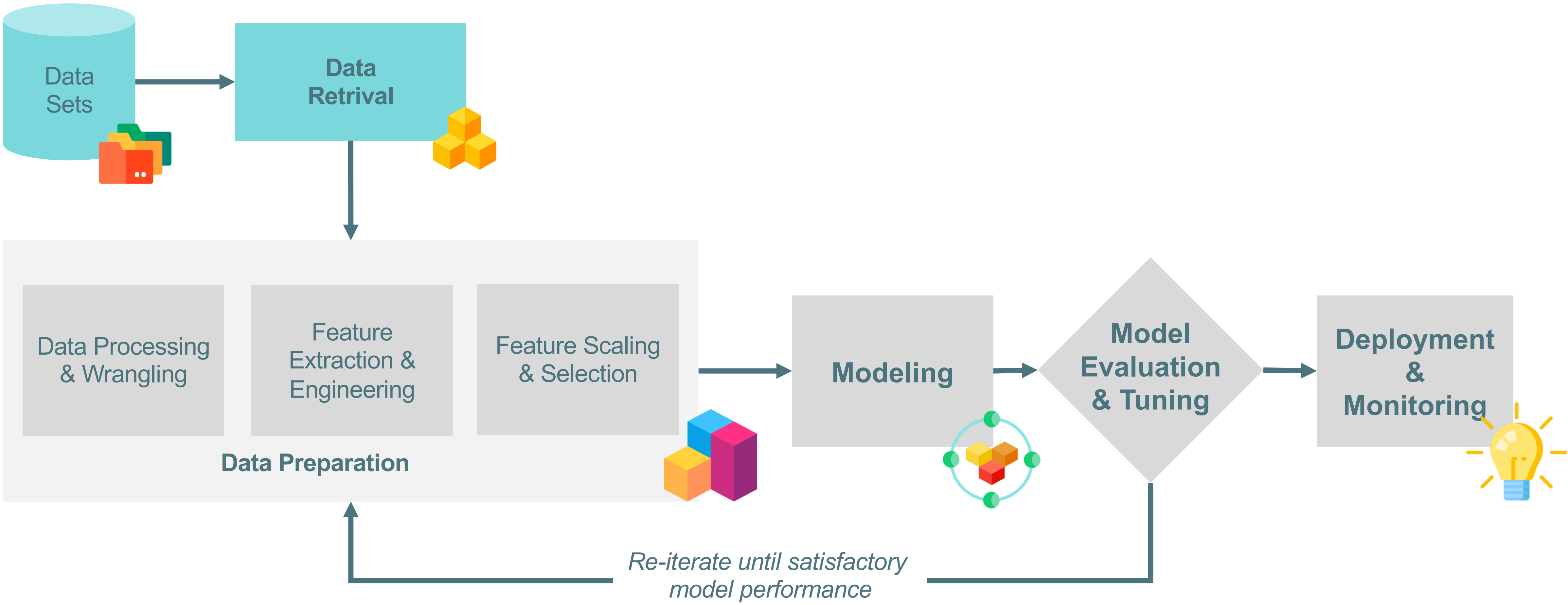
Deployment & Monitoring

Re-iterate until satisfactory model performance

OFFENKUNDIGGUT

OPEN KNOWLEDGE

Data Understanding


Data Sets
Data Retrival
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate until satisfactory model performance

Data Understanding
by Example

Collect & describe data - SALES FORECAST

Sales reports* and **other data**** on the POS from the last three years serve as the basis for the initial analysis..

* Sales per store per day

** Location, size, employees, distance to competitors, etc.

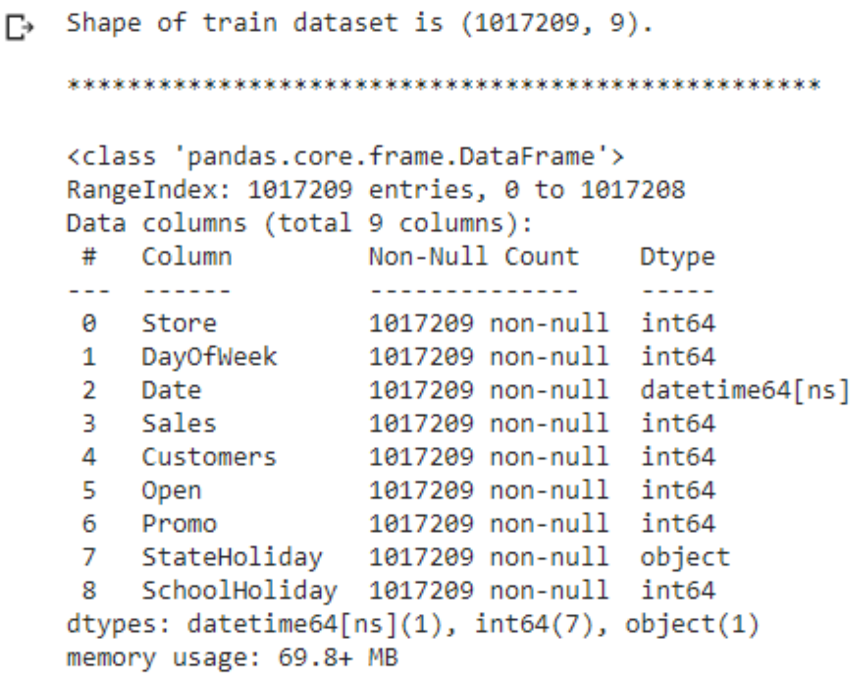
Data Understanding
by Example

Collect & describe data

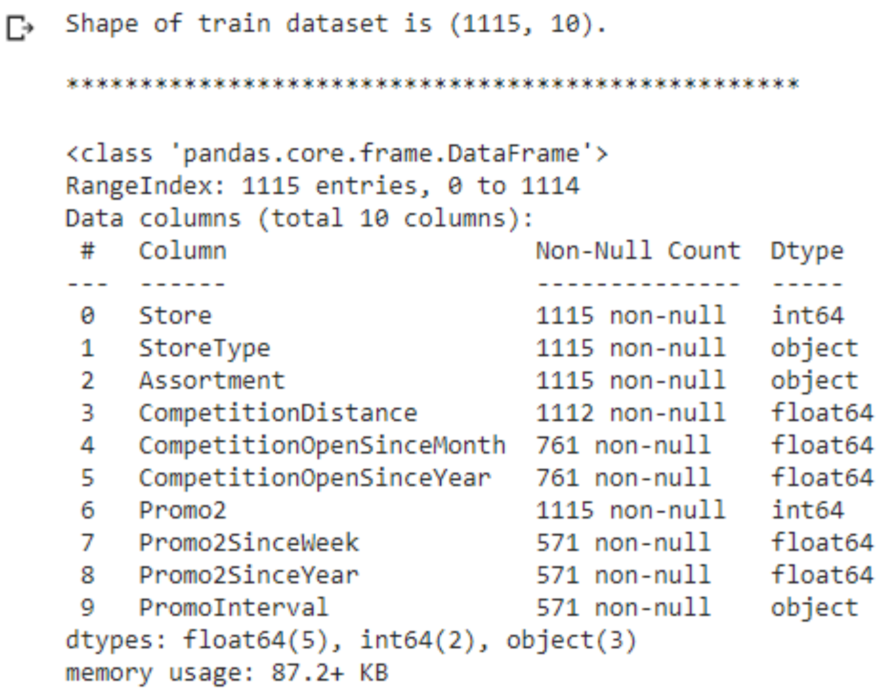
The sales report and POS data are examined in more detail with the help of an **exploratory data analysis** (EDA).

The focus is on the question of **which variables have a direct or indirect influence** on sales.

Data Understanding
by Example



```
Shape of train dataset is (1017209, 9).

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   Store          1017209 non-null  int64
 1   DayOfWeek      1017209 non-null  int64
 2   Date           1017209 non-null  datetime64[ns]
 3   Sales          1017209 non-null  int64
 4   Customers      1017209 non-null  int64
 5   Open           1017209 non-null  int64
 6   Promo          1017209 non-null  int64
 7   StateHoliday   1017209 non-null  object
 8   SchoolHoliday  1017209 non-null  int64
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 69.8+ MB
```



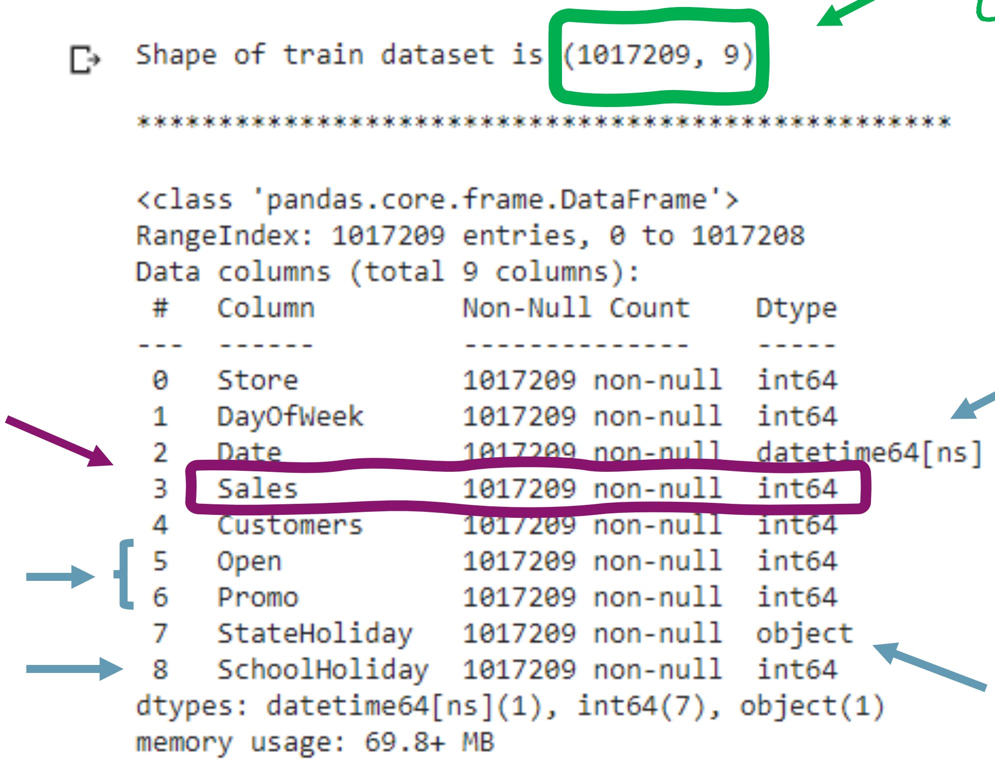
```
Shape of train dataset is (1115, 10).

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Store                      1115 non-null   int64
 1   StoreType                  1115 non-null   object
 2   Assortment                 1115 non-null   object
 3   CompetitionDistance        1112 non-null   float64
 4   CompetitionOpenSinceMonth  761 non-null    float64
 5   CompetitionOpenSinceYear   761 non-null    float64
 6   Promo2                     1115 non-null   int64
 7   Promo2SinceWeek            571 non-null    float64
 8   Promo2SinceYear            571 non-null    float64
 9   PromoInterval              571 non-null    object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

Sales Data
(>1 mio data records)

Store Data
(ca. 1100 data records)

Data Understanding
by Example

Sales Data
(>1 Mio Data Records)



```
Shape of train dataset is (1017209, 9)

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   Store          1017209 non-null  int64
 1   DayOfWeek      1017209 non-null  int64
 2   Date           1017209 non-null  datetime64[ns]
 3   Sales          1017209 non-null  int64
 4   Customers      1017209 non-null  int64
 5   Open           1017209 non-null  int64
 6   Promo          1017209 non-null  int64
 7   StateHoliday   1017209 non-null  object
 8   SchoolHoliday  1017209 non-null  int64
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 69.8+ MB
```

> 1 MIO Data Records
Lots of data! very nice.

SALES!
Target variable

DATETIME?
For sorting the "Time Series". But is there any more info in there?

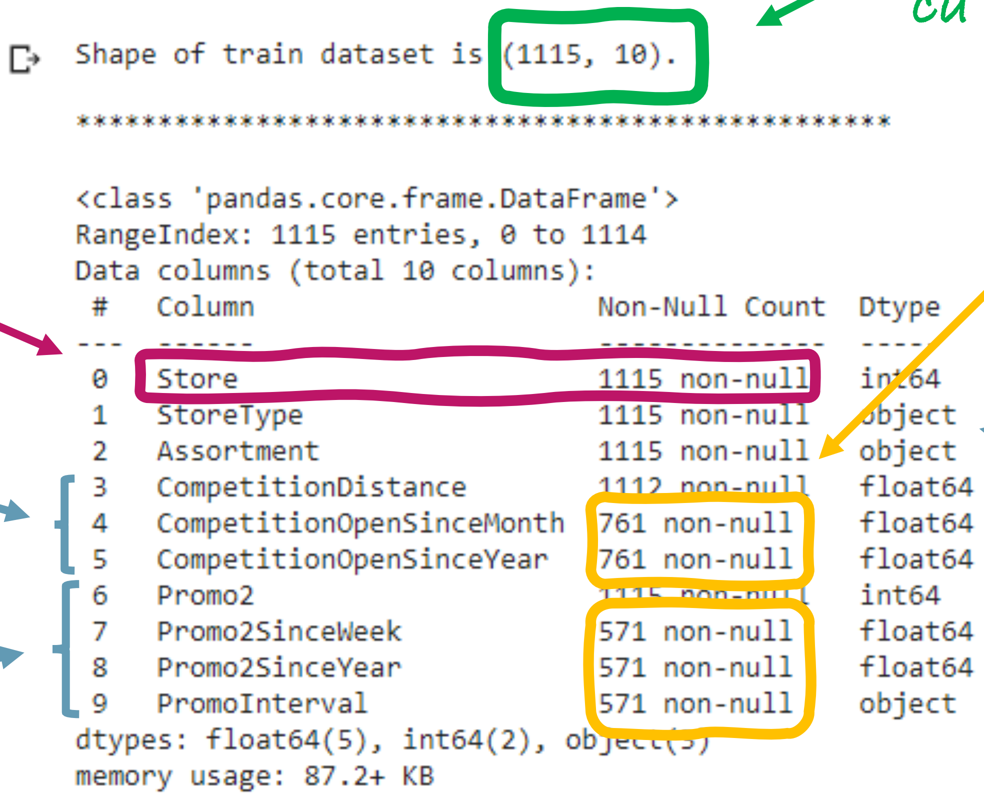
INT64?
Which values/ranges are possible? And what do they mean??

OBJECT?
Inconvenient for ML!

Data Understanding
by Example

Store Data

(ca. 1100 Data Records)



```
Shape of train dataset is (1115, 10).

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Store                      1115 non-null   int64
 1   StoreType                  1115 non-null   object
 2   Assortment                 1115 non-null   object
 3   CompetitionDistance        1112 non-null   float64
 4   CompetitionOpenSinceMonth  761 non-null    float64
 5   CompetitionOpenSinceYear   761 non-null    float64
 6   Promo2                     1115 non-null   int64
 7   Promo2SinceWeek            571 non-null    float64
 8   Promo2SinceYear            571 non-null    float64
 9   PromoInterval              571 non-null    object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

> 1000 DATENSÄTZE
ca 1 Mio of 1000 Stores

STORE!
Merge Reference

NULL?
Why are there no values?

COMPETITION?
What semantics?

OBJECT?
Inconvenient for ML!

PROMO2?
What semantics?

Data Understanding
by Exa

Store D

(ca. 1100 Data R

Me

COMP

What s

P

What se

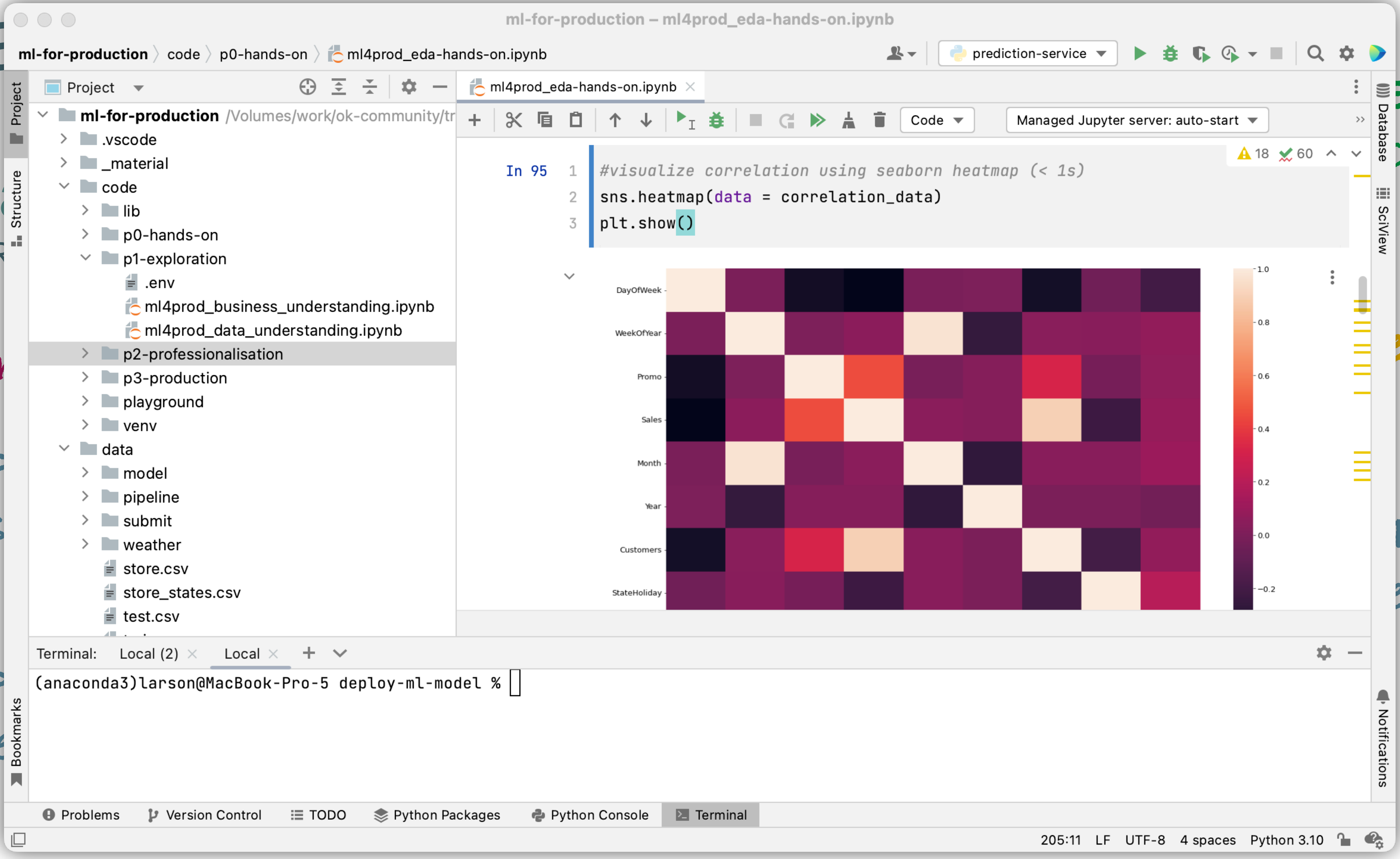
ENSÄTZE

00 Stores

e there no

T?

enient for ML!



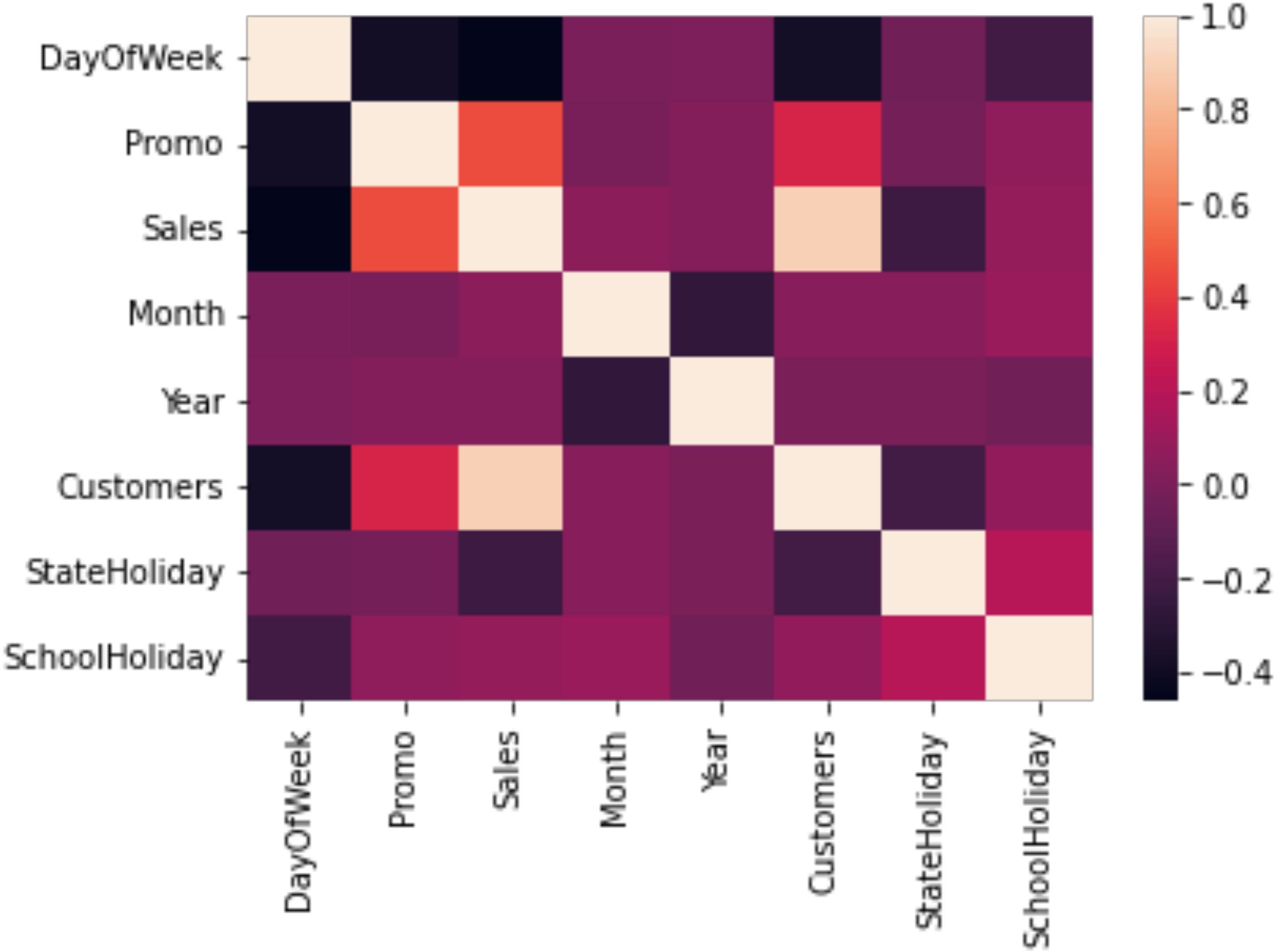
```
#visualize correlation using seaborn heatmap (< 1s)
sns.heatmap(data = correlation_data)
plt.show()
```

```
(anaconda3)larson@MacBook-Pro-5 deploy-ml-model %
```

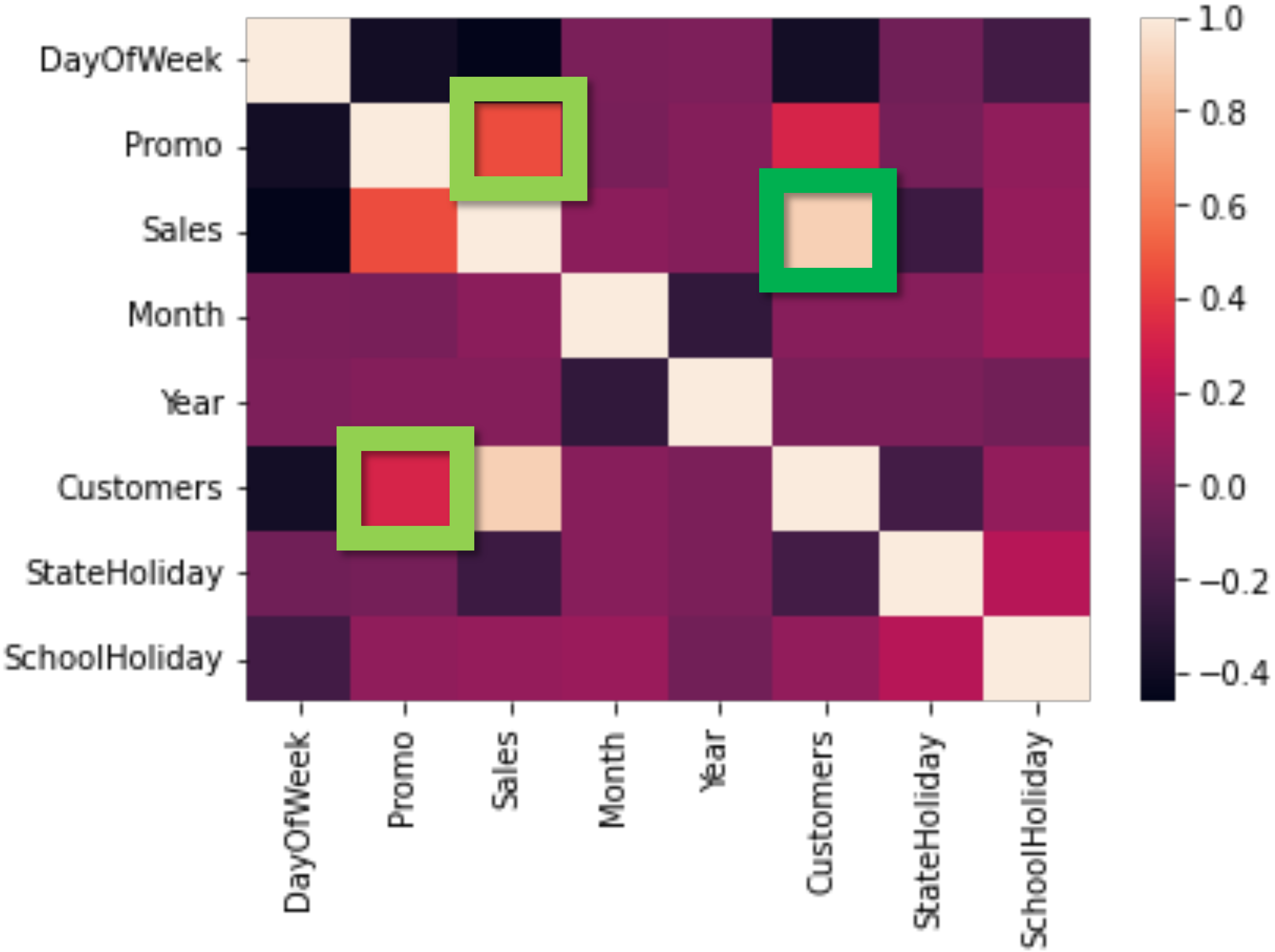
OFFENKUNDIGGUT

OPEN KNOW LEDGE

Product Sales “Correlations“

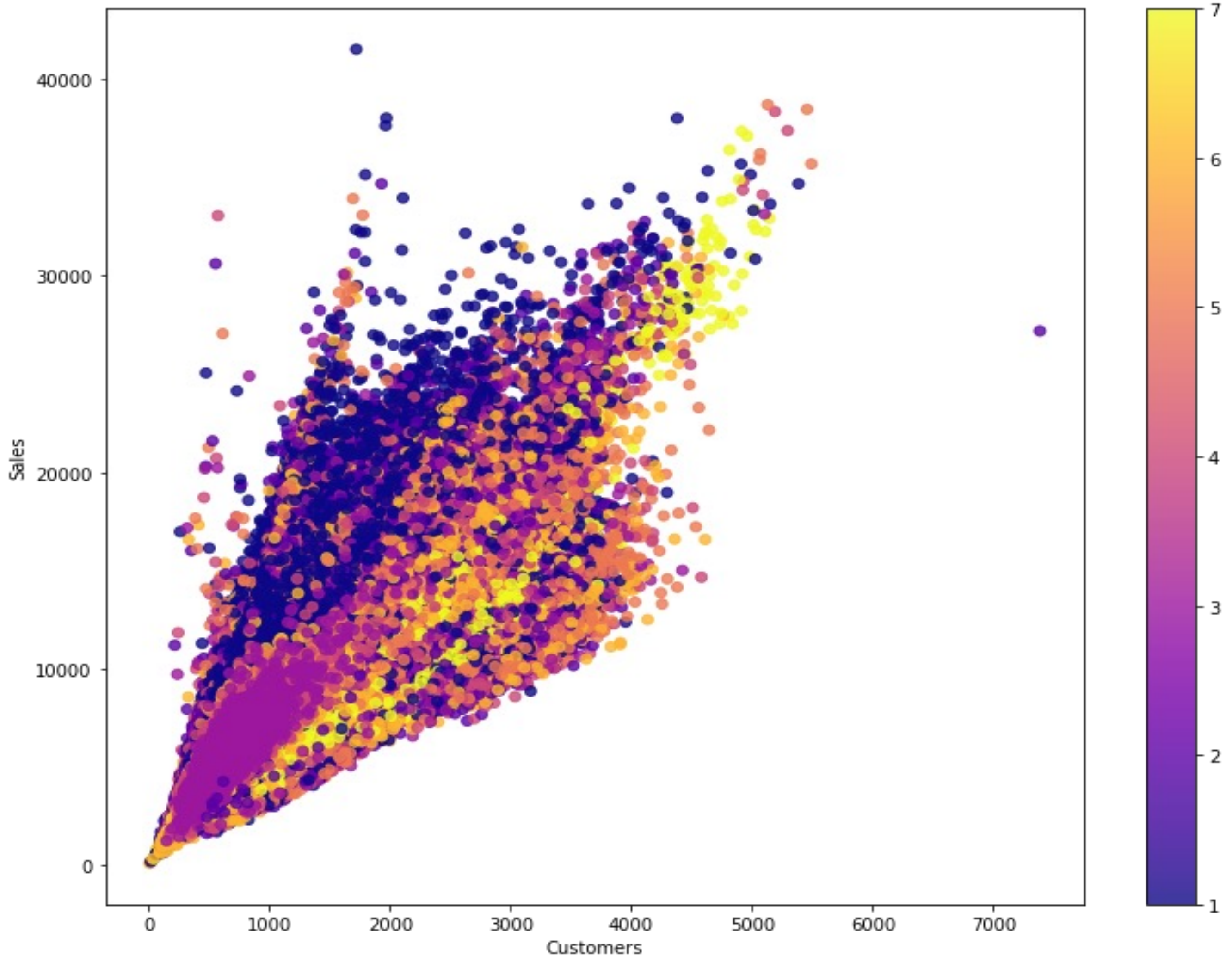


Product Sales “Correlations“

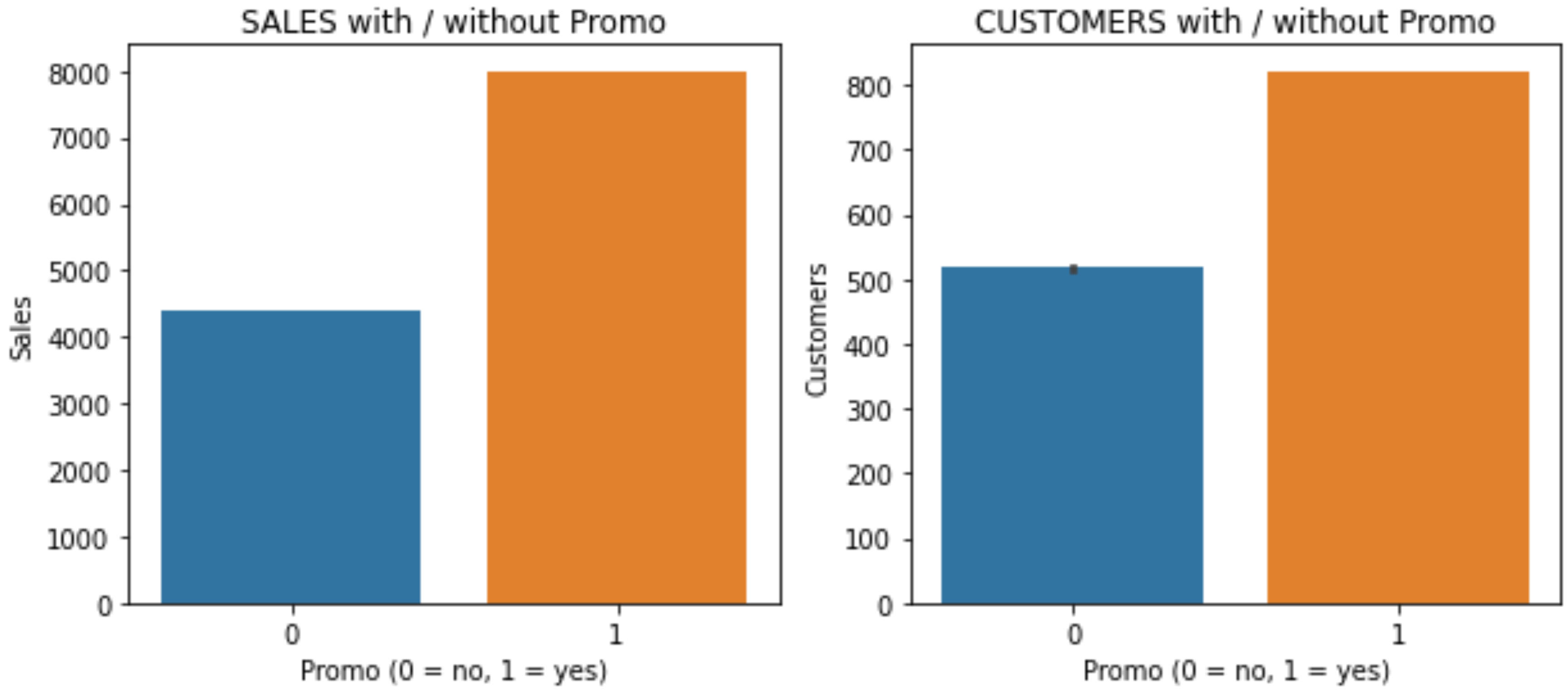


Product Sales “Customer“

As suspected, there is a high correlation between Sales and Customer.

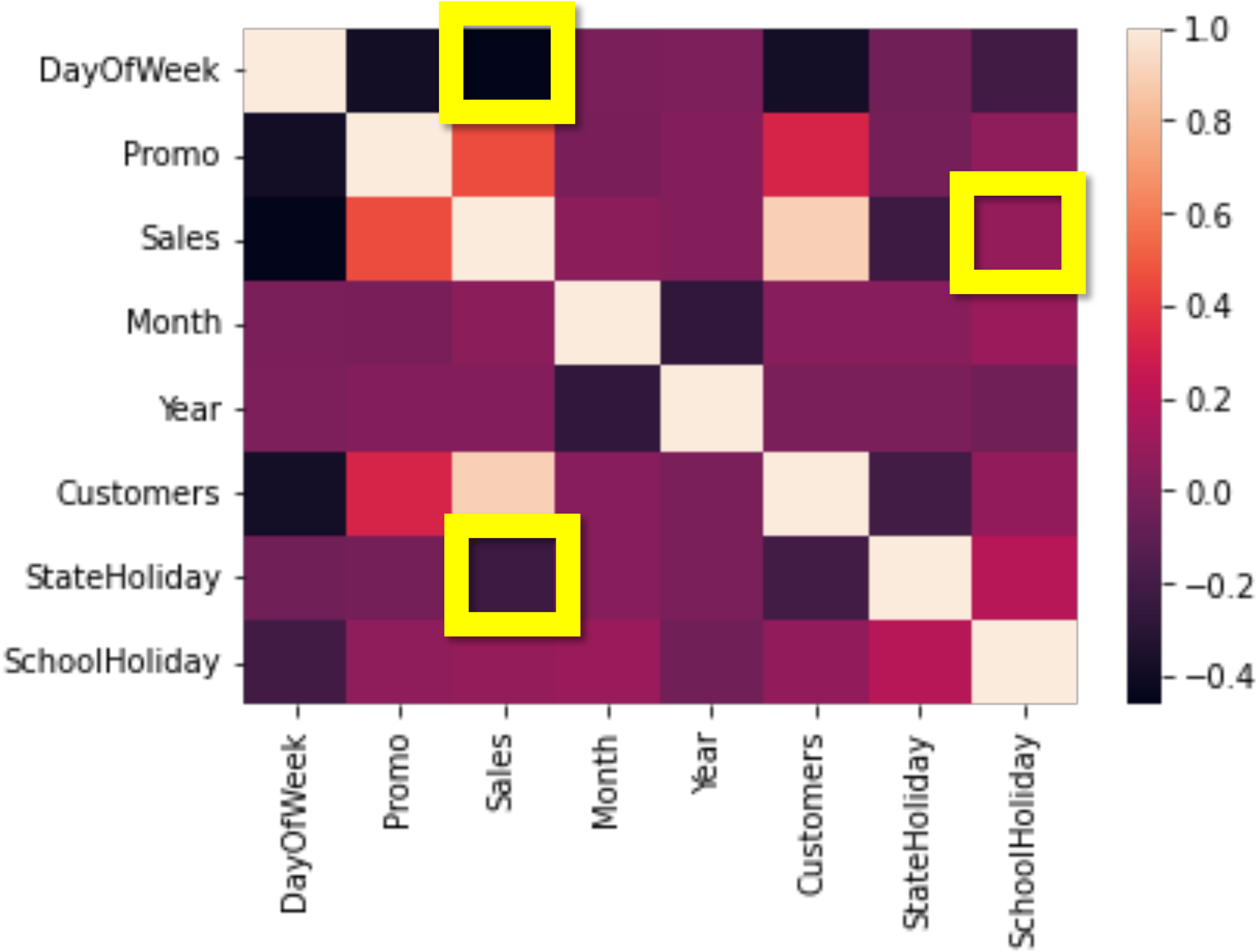


Product Sales “Promo“



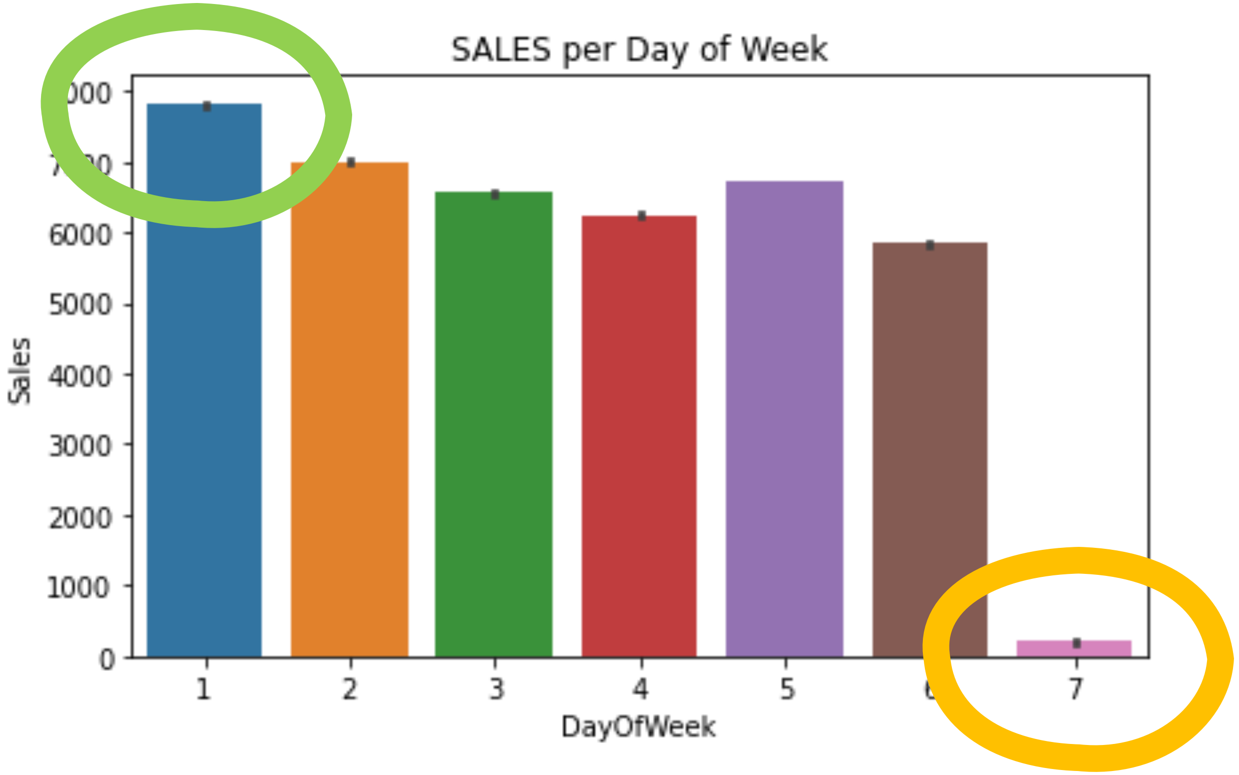
Average value for SALES and CUSTOMERS is significantly higher for PROMO = 1 than for PROMO = 0.

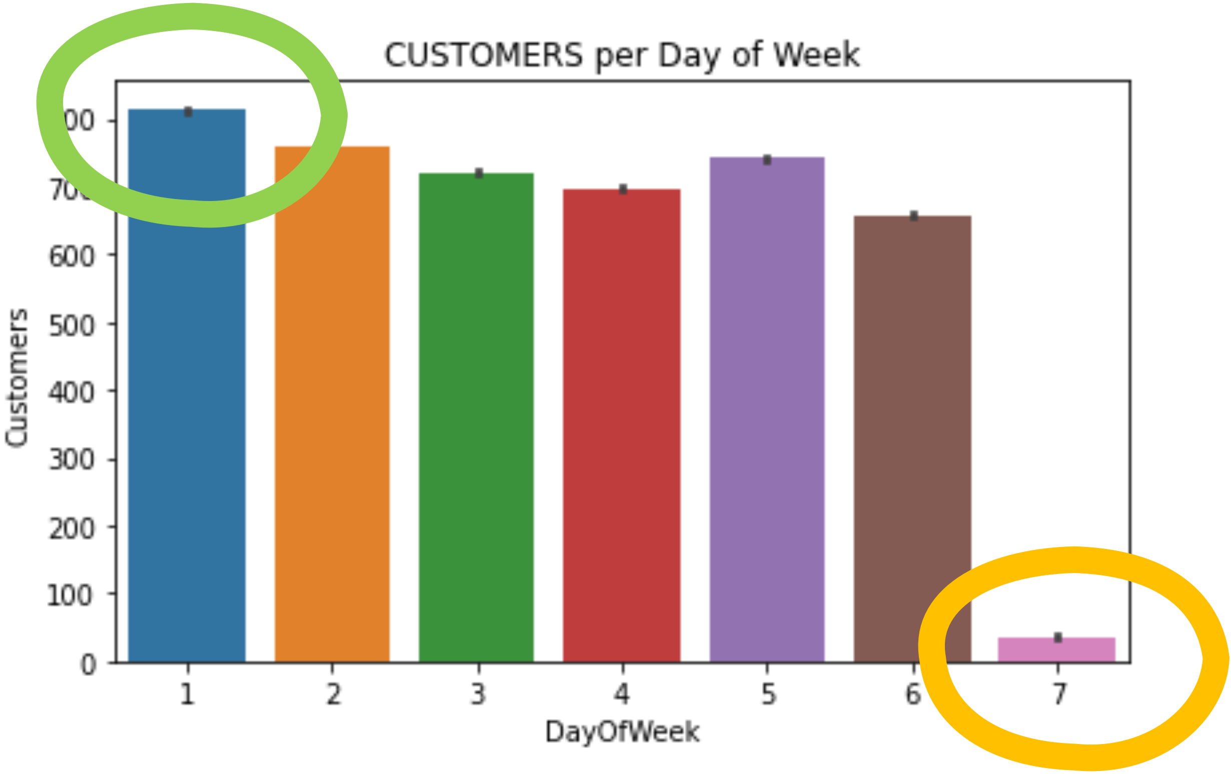
Product Sales “Correlations“



Product Sales “Weekday“

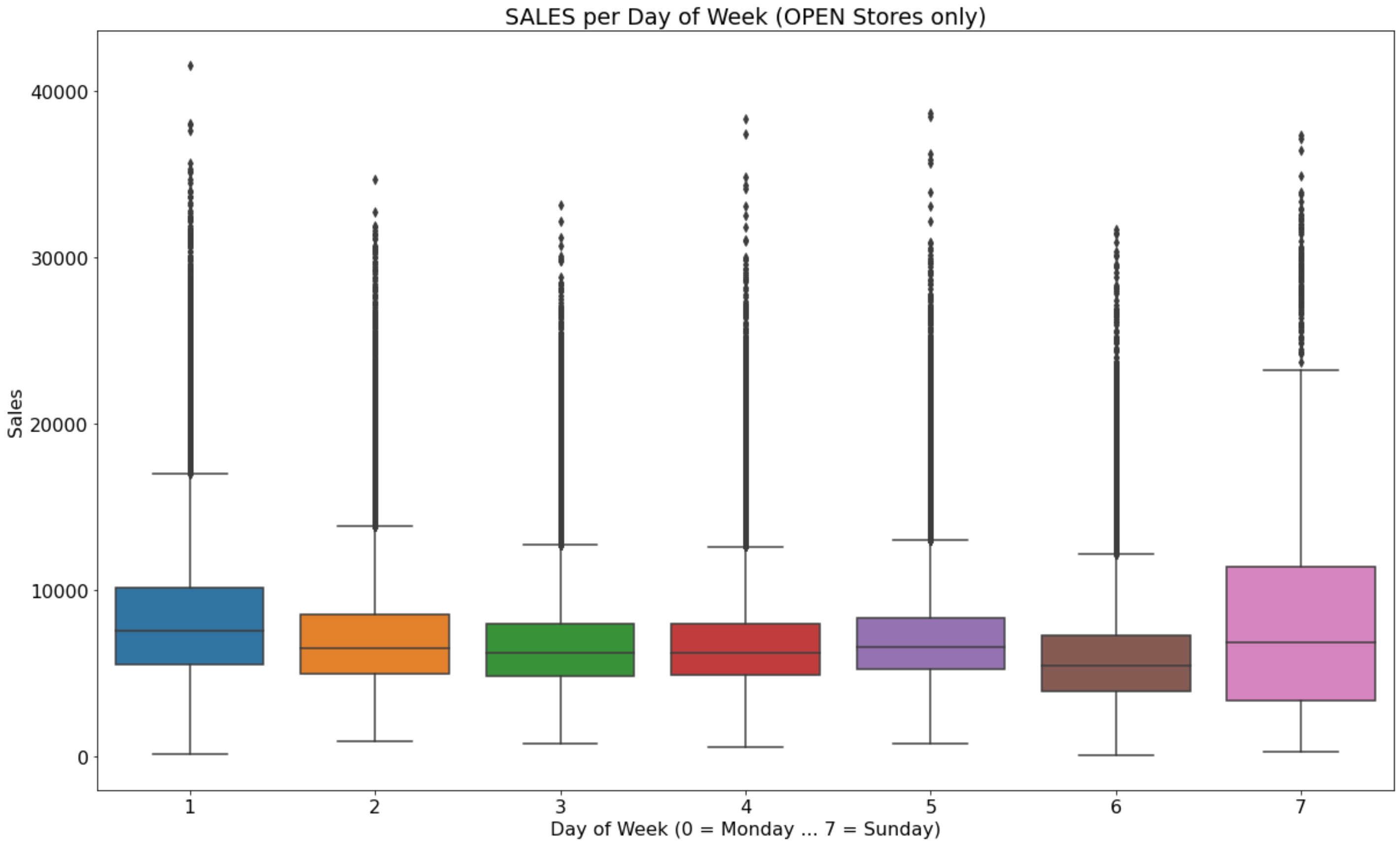
Increased visits / purchases
after the weekend

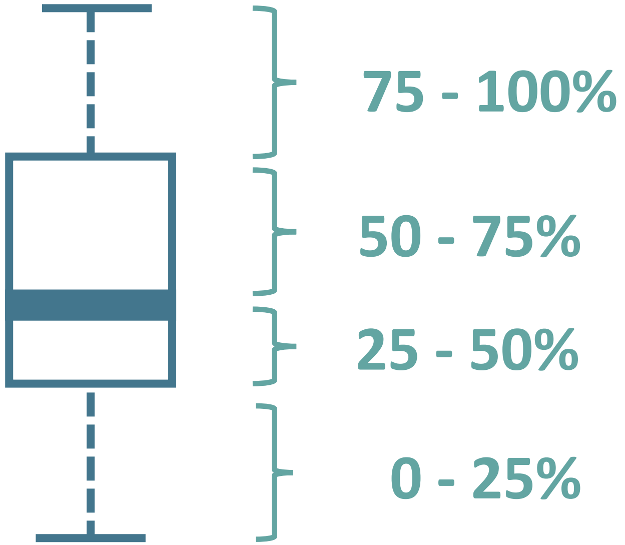




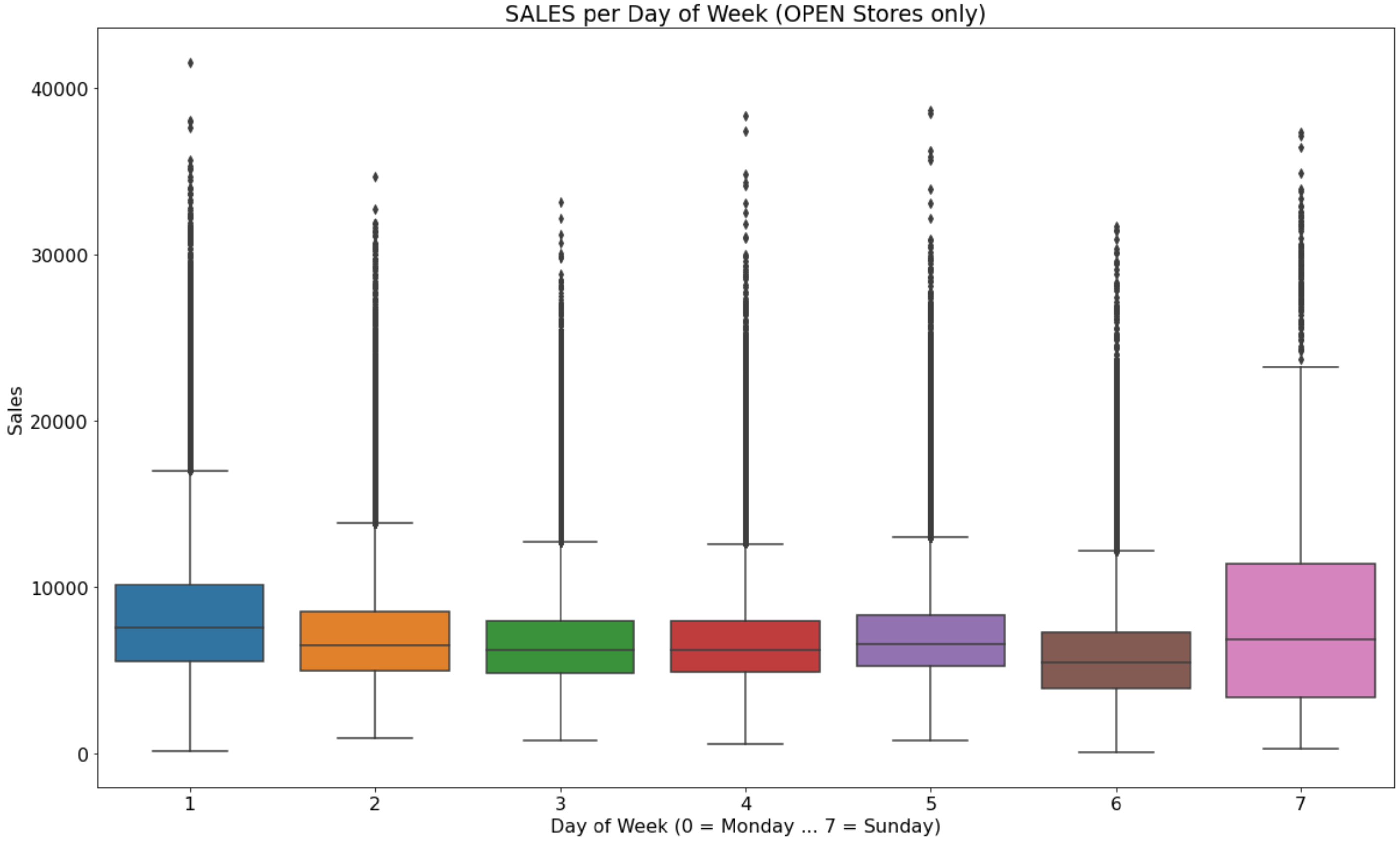
Probably only a few stores
are open on Sunday open.

Product Sales “Weekday“





Product Sales “Weekday“



TUE to FRI

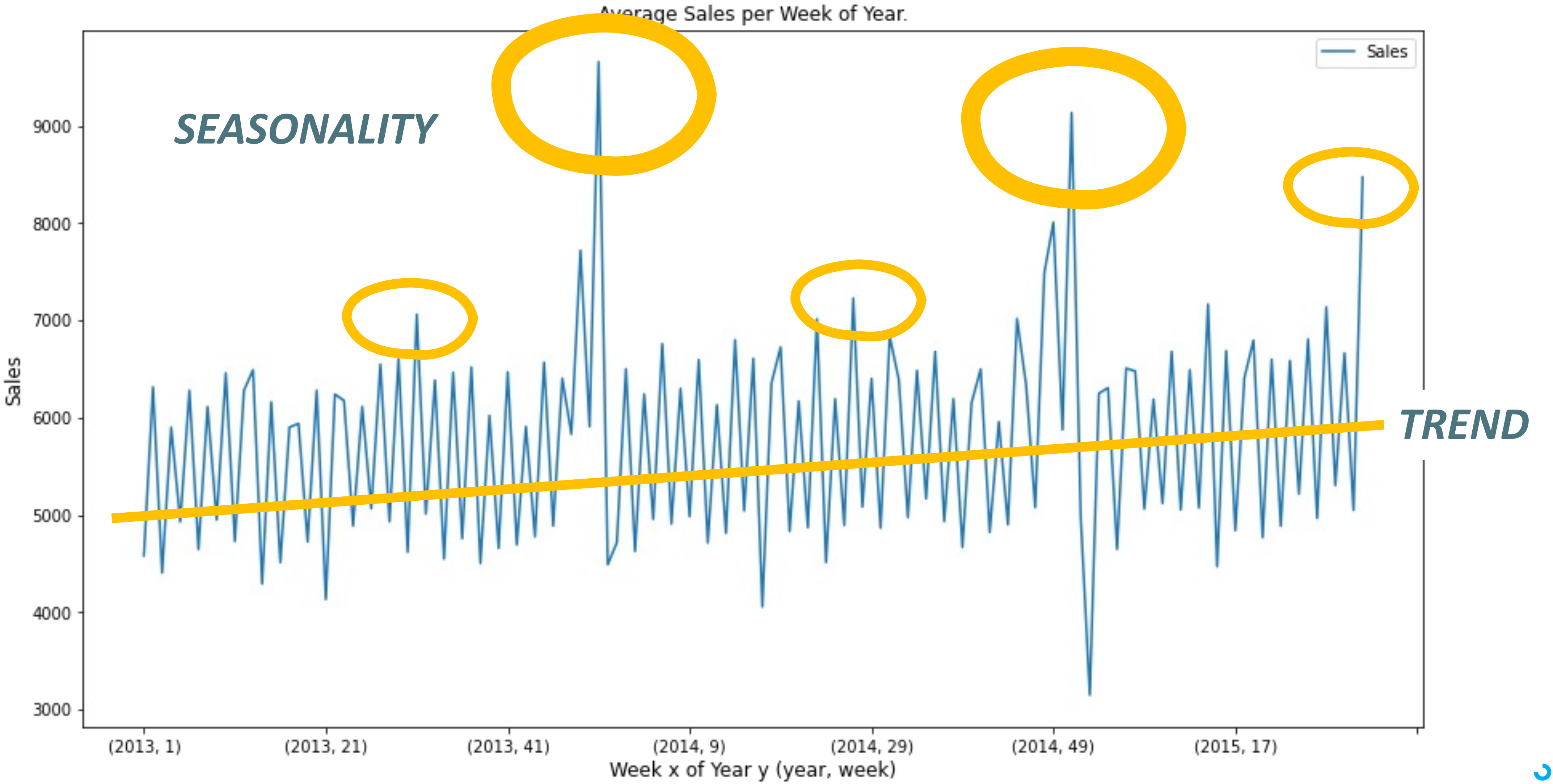
similar pattern

SAT, SUN, MON

Individual pattern*

*weekday as own Feature

Product Sales “Seasons“

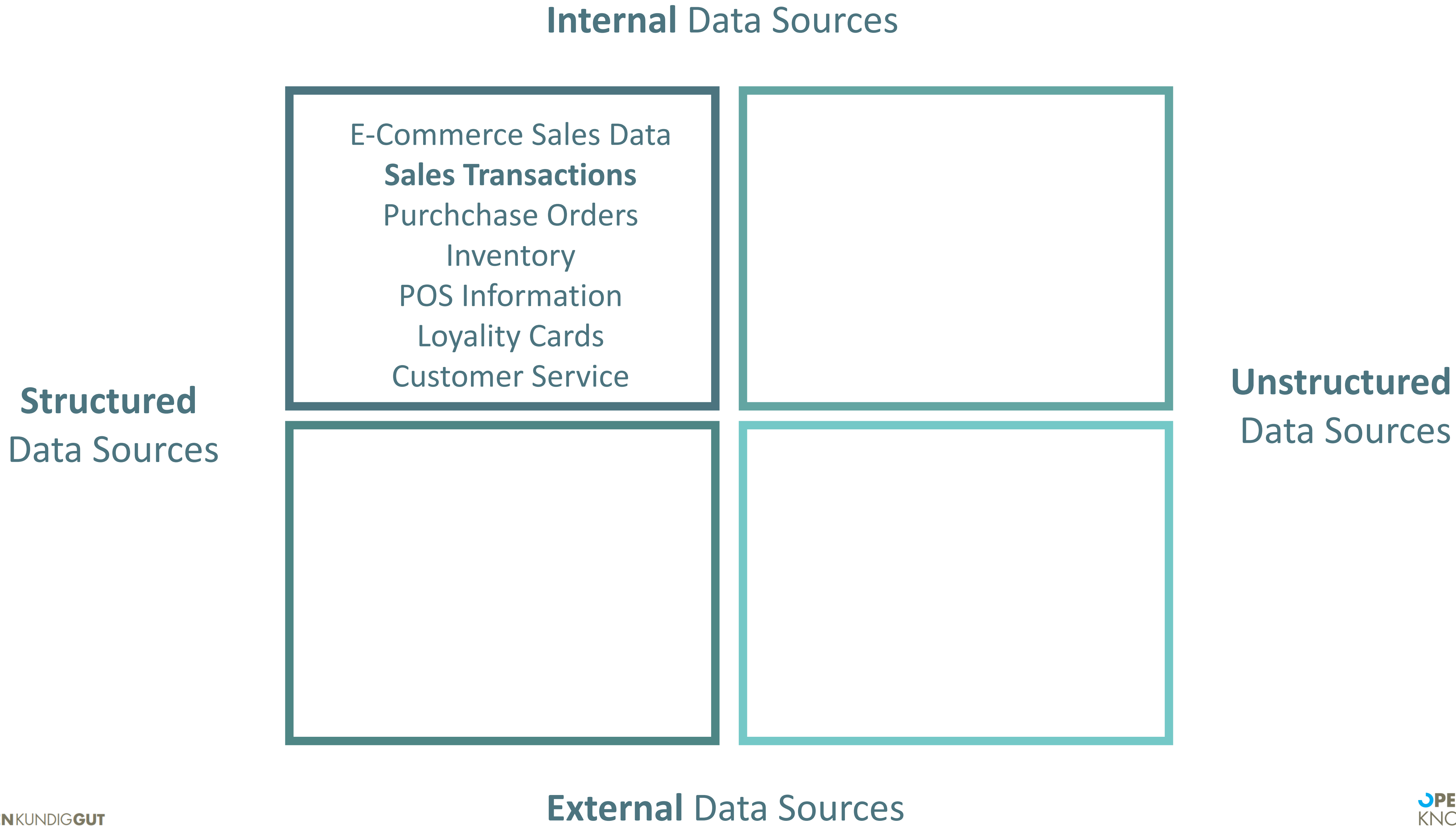


Data Understanding
by Example

Collect & describe data - SALES FORECAST

In addition to **sales reports** from individual POS, there are **numerous other sources** that can be used for ML-based forecasting to paint a more accurate picture.

	Structured Data Sources	**Unstructured** Data Sources
Internal Data Sources	**Sales Transactions**	
External Data Sources		



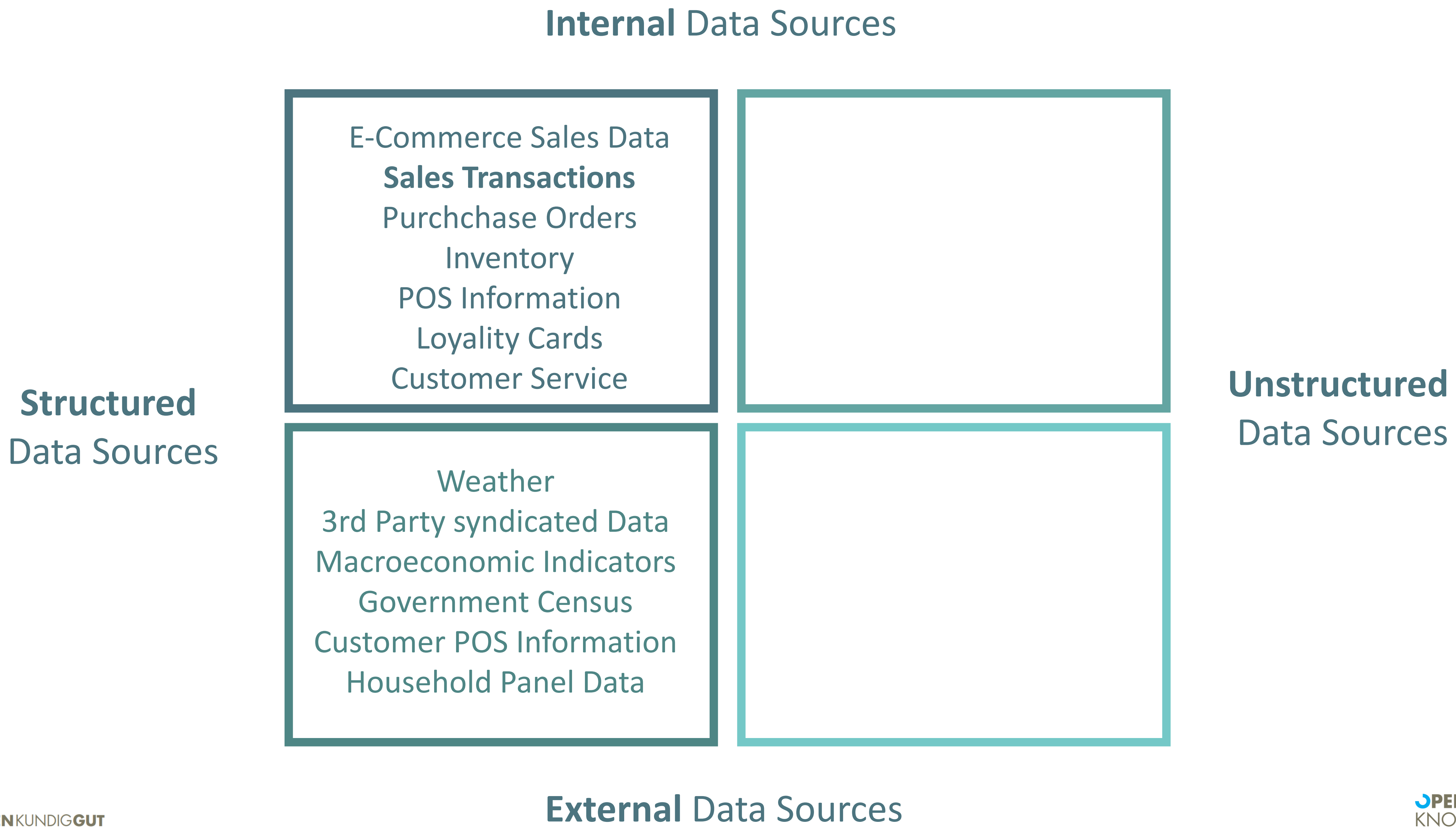
Internal Data Sources

Structured
Data Sources

E-Commerce Sales Data
Sales Transactions
Purchchase Orders
Inventory
POS Information
Loyality Cards
Customer Service

Unstructured
Data Sources

External Data Sources



Internal Data Sources

Structured Data Sources

Unstructured Data Sources

External Data Sources

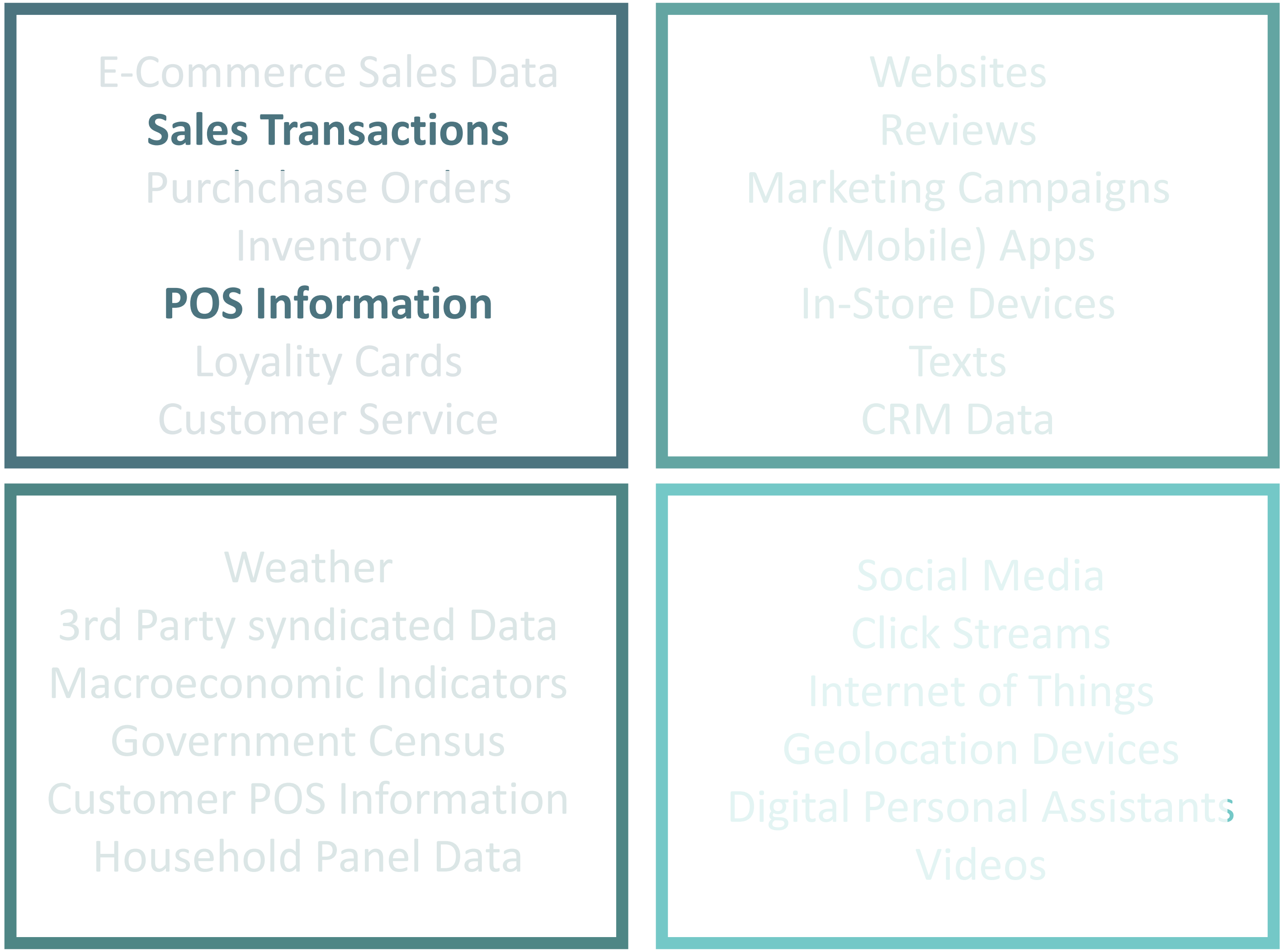
E-Commerce Sales Data
Sales Transactions
Purchchase Orders
Inventory
POS Information
Loyality Cards
Customer Service

Weather
3rd Party syndicated Data
Macroeconomic Indicators
Government Census
Customer POS Information
Household Panel Data

	Structured Data Sources	**Unstructured** Data Sources
Internal Data Sources	E-Commerce Sales Data **Sales Transactions** Purchchase Orders Inventory POS Information Loyality Cards Customer Service	Websites Reviews Marketing Campaigns (Mobile) Apps In-Store Devices Texts CRM Data
External Data Sources	Weather 3rd Party syndicated Data Macroeconomic Indicators Government Census Customer POS Information Household Panel Data	

	Structured Data Sources	**Unstructured** Data Sources
Internal Data Sources	E-Commerce Sales Data **Sales Transactions** Purchchase Orders Inventory POS Information Loyality Cards Customer Service	Websites Reviews Marketing Campaigns (Mobile) Apps In-Store Devices Texts CRM Data
External Data Sources	Weather 3rd Party syndicated Data Macroeconomic Indicators Government Census Customer POS Information Household Panel Data	Social Media Click Streams Internet of Things Geolocation Devices Digital Personal Assistants Videos

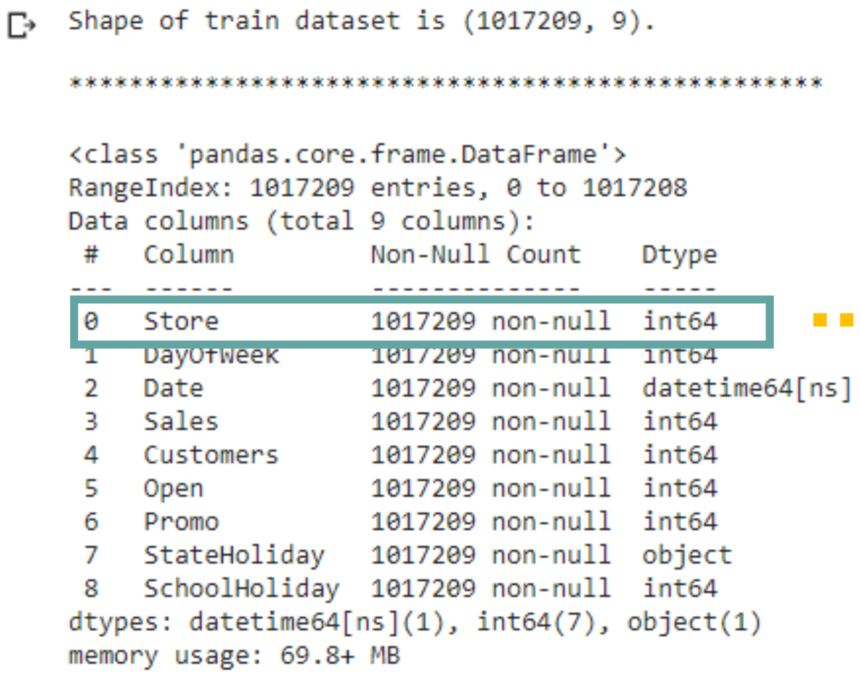
Internal Data Sources



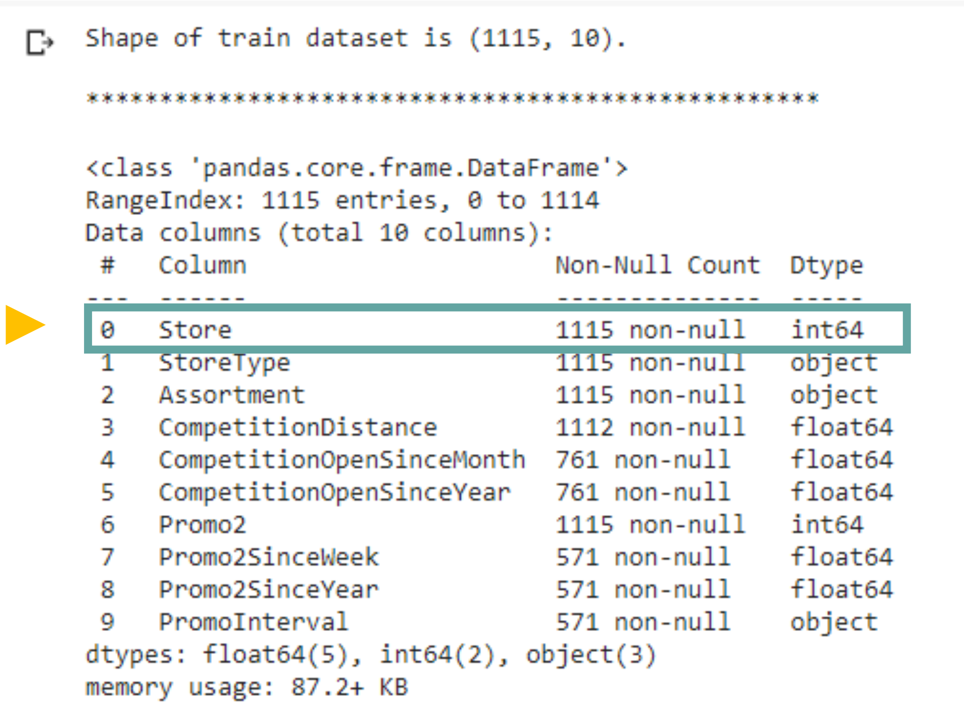
Structured Data Sources	**Unstructured** Data Sources
E-Commerce Sales Data **Sales Transactions** Purchchase Orders Inventory **POS Information** Loyality Cards Customer Service	Websites Reviews Marketing Campaigns (Mobile) Apps In-Store Devices Texts CRM Data
Weather 3rd Party syndicated Data Macroeconomic Indicators Government Census Customer POS Information Household Panel Data	Social Media Click Streams Internet of Things Geolocation Devices Digital Personal Assistants Videos

External Data Sources

Product Sales “Merge Data “



```
Shape of train dataset is (1017209, 9).

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   Store          1017209 non-null  int64
 1   DayOfWeek      1017209 non-null  int64
 2   Date           1017209 non-null  datetime64[ns]
 3   Sales          1017209 non-null  int64
 4   Customers      1017209 non-null  int64
 5   Open           1017209 non-null  int64
 6   Promo          1017209 non-null  int64
 7   StateHoliday   1017209 non-null  object
 8   SchoolHoliday  1017209 non-null  int64
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 69.8+ MB
```



```
Shape of train dataset is (1115, 10).

**************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Store                       1115 non-null   int64
 1   StoreType                   1115 non-null   object
 2   Assortment                  1115 non-null   object
 3   CompetitionDistance         1112 non-null   float64
 4   CompetitionOpenSinceMonth   761 non-null    float64
 5   CompetitionOpenSinceYear    761 non-null    float64
 6   Promo2                      1115 non-null   int64
 7   Promo2SinceWeek             571 non-null    float64
 8   Promo2SinceYear             571 non-null    float64
 9   PromoInterval               571 non-null    object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

Sales Data

(>1 Mio Data Records)

Store Data

(ca. 1100 Data Records)

Data Understanding
by Example

Collect & describe data - SALES FORECAST

The data must be analyzed in advance for many **quality factors**:

Availability, Completeness, Accuracy,

Validity, Consistency, Relevance,

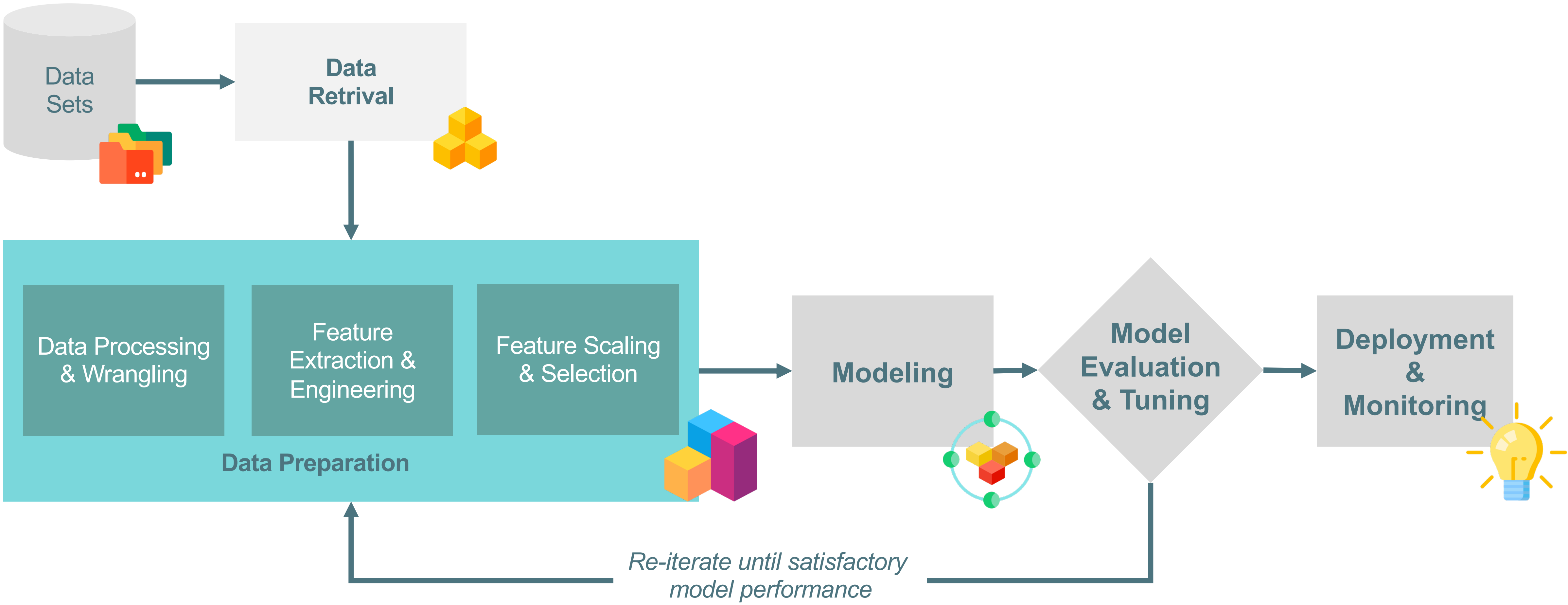
Granularity, Cost

Data Preparation

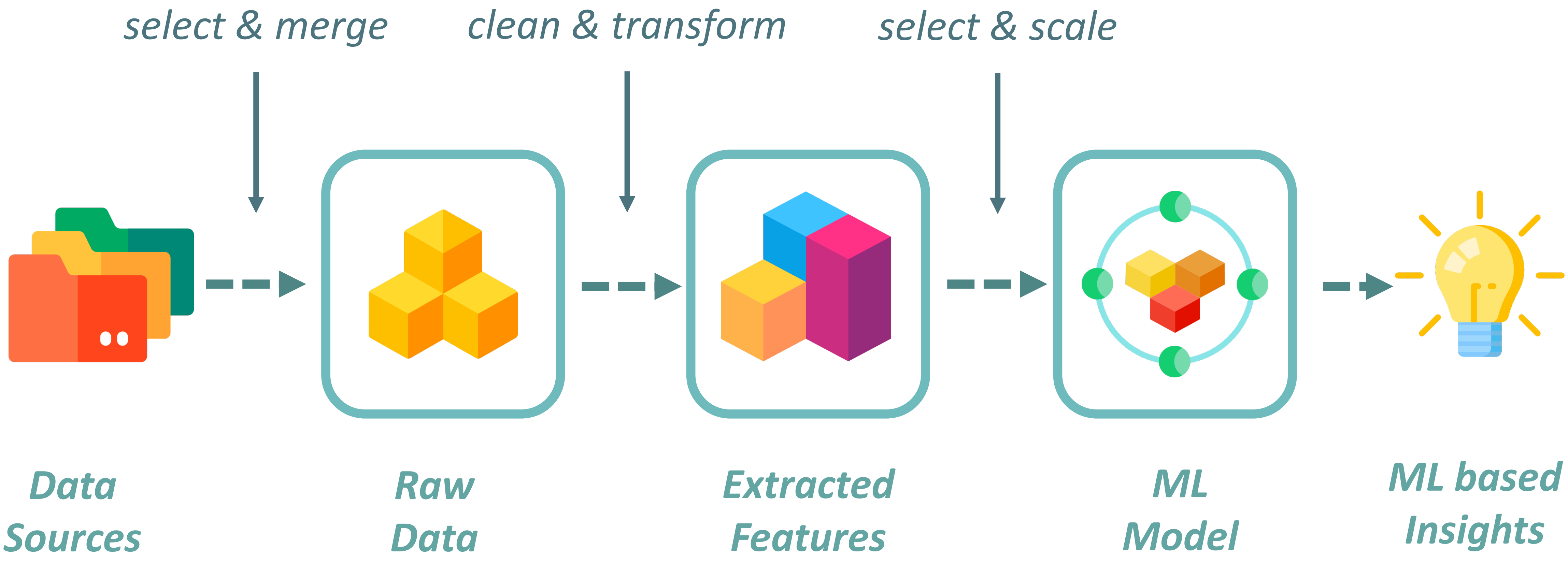
„ Quantity is not equal to Quality!“

Select, transform & clean data

Data Preparation Process


Data Sets
Data Retrival
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate until satisfactory model performance

Data Preparation Process



select & merge

clean & transform

select & scale

Data Sources

Raw Data

Extracted Features

ML Model

ML based Insights

Data Preparation Process

Data Collection

- name problem
- define data
- collect and combine data

Data Preprocessing

- format
- validate
- clean
- sample
- type
- refine

Data Transformation*

*aka Feature Engineering / Selection

- scale
- normalize
- split
- aggregate
- encode
- …

Data Preparation

What exactly are these **features**?

What does **Feature Engineering** mean?

And what is the difference to **Feature Selection**?

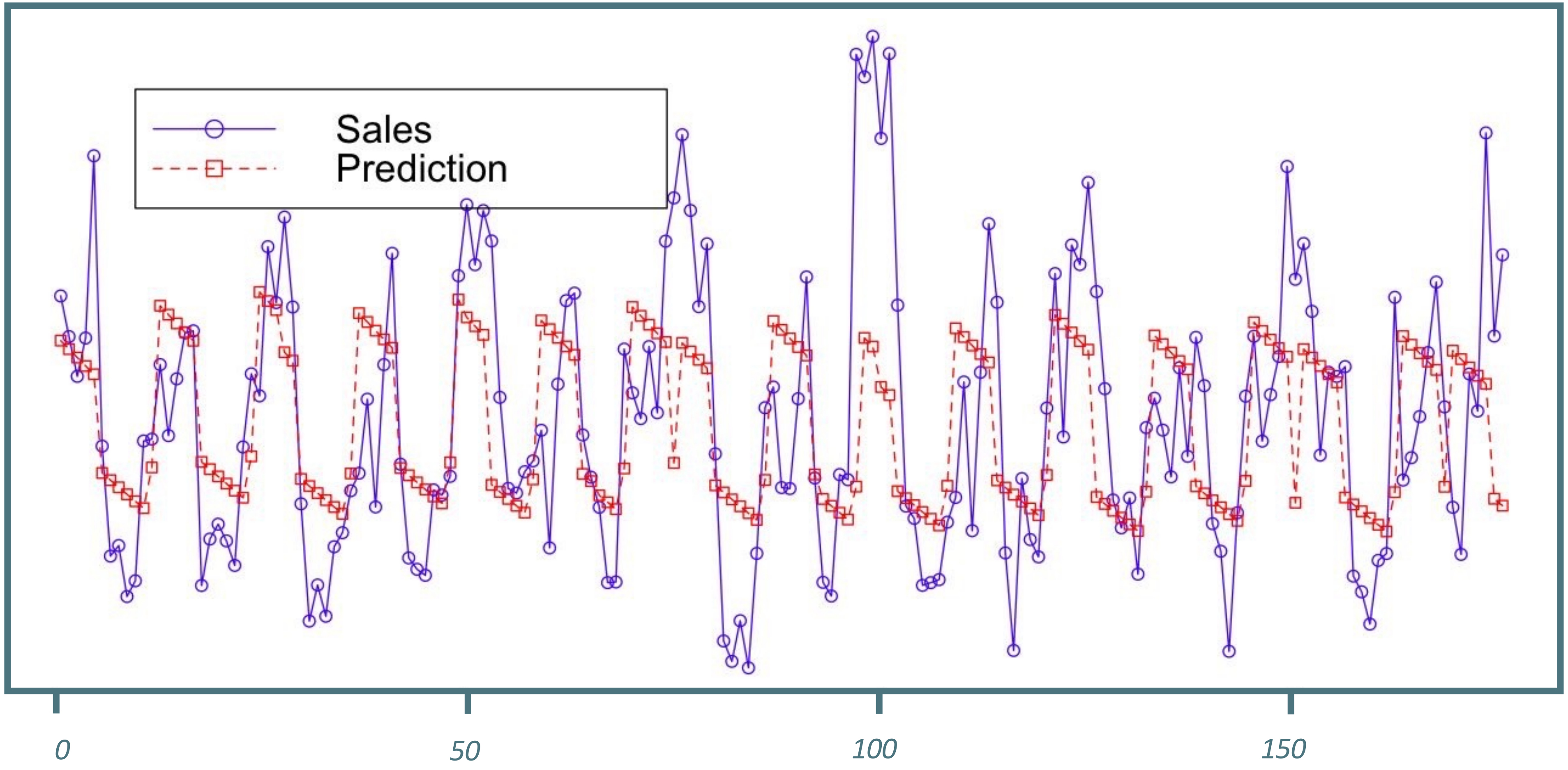
Data Preparation

Feature Engineering / Selection

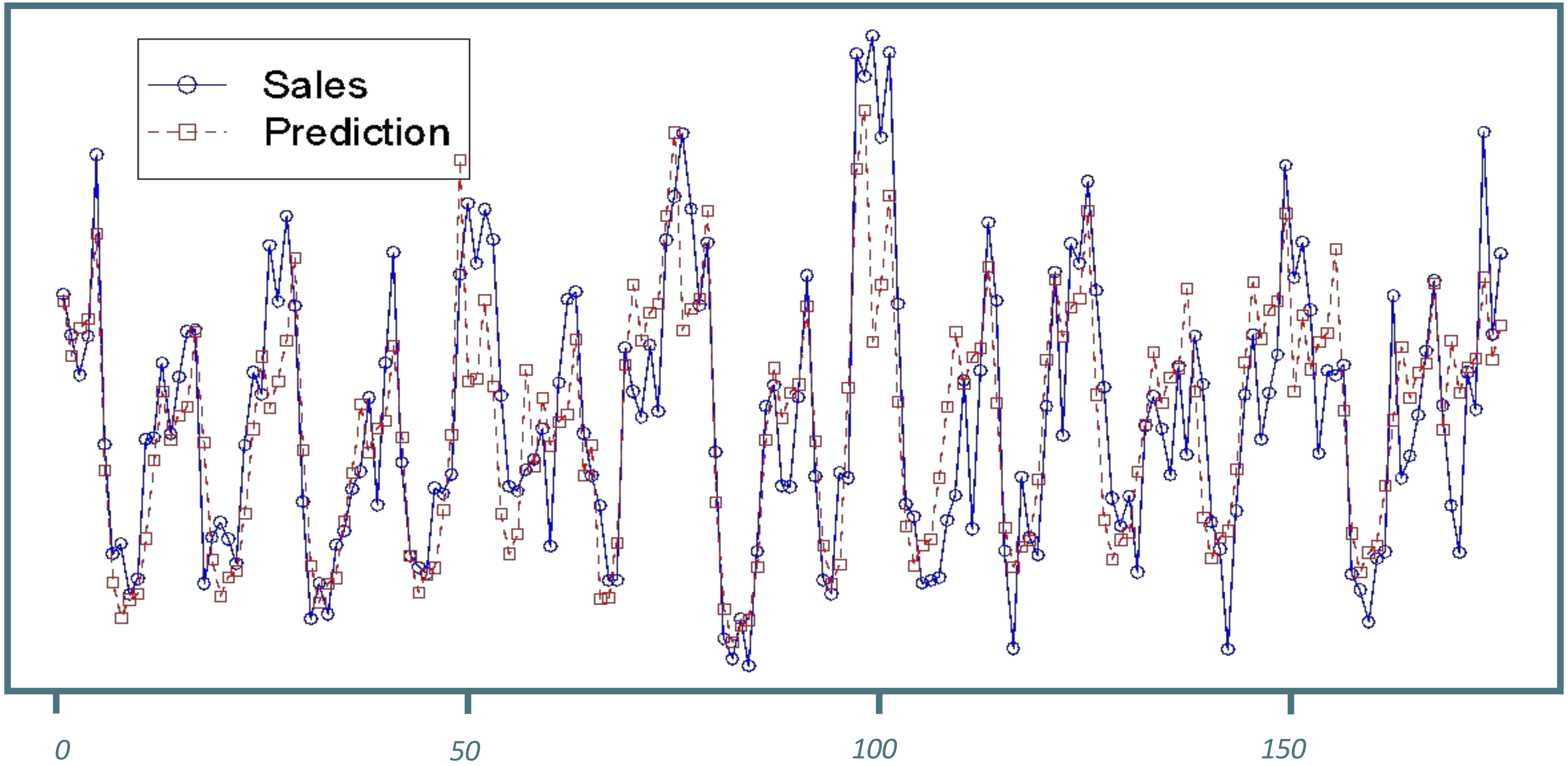
*YOU: "But that sounds **damn elaborate**! Does it have to be? We're dealing with **artificial intelligence**, after all! Surely the model can do that on its own, right?“*

*ME: “YES and NO! Feature Engineering / Selection is an **essential part of the ML4Prod pipeline** to optimize the model. In this way, we give the model hints on what it should pay attention to."*

Sales Prediction **without additional Features**



Sales Prediction **with additional Features**



Data Preparation

The Art of Feature Engineering & Selection

„As many as necessary and as few as possible.“

- as many as necessary leads to **good results**
- as few as possible leads to **good performance**

Data Preparation

Feature Selection for everyone?

„There is a tool for it …“

- FeatureTools*
- FeatureSelector**

*https://www.featuretools.com

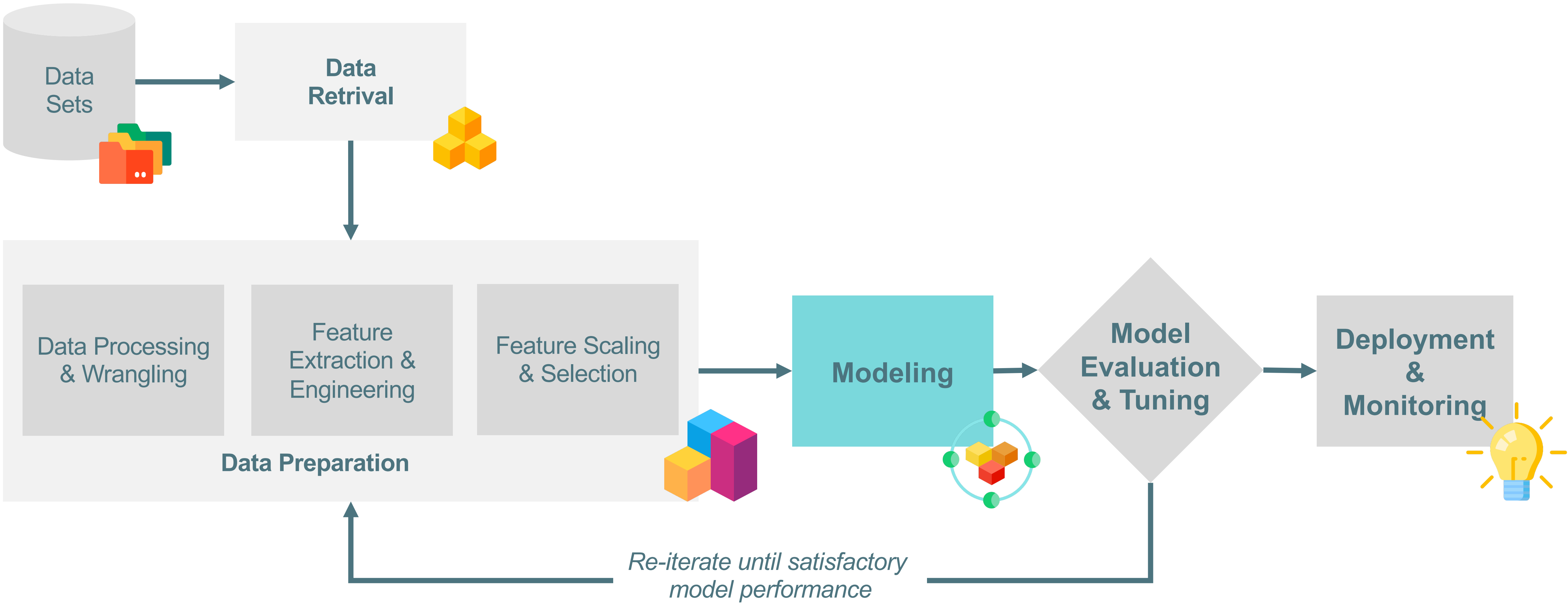
**https://github.com/WillKoehrsen/feature-selector

Analysis & Modeling

„Which model suits me best?“

Select, train & evaluate model

Analysis & Modeling

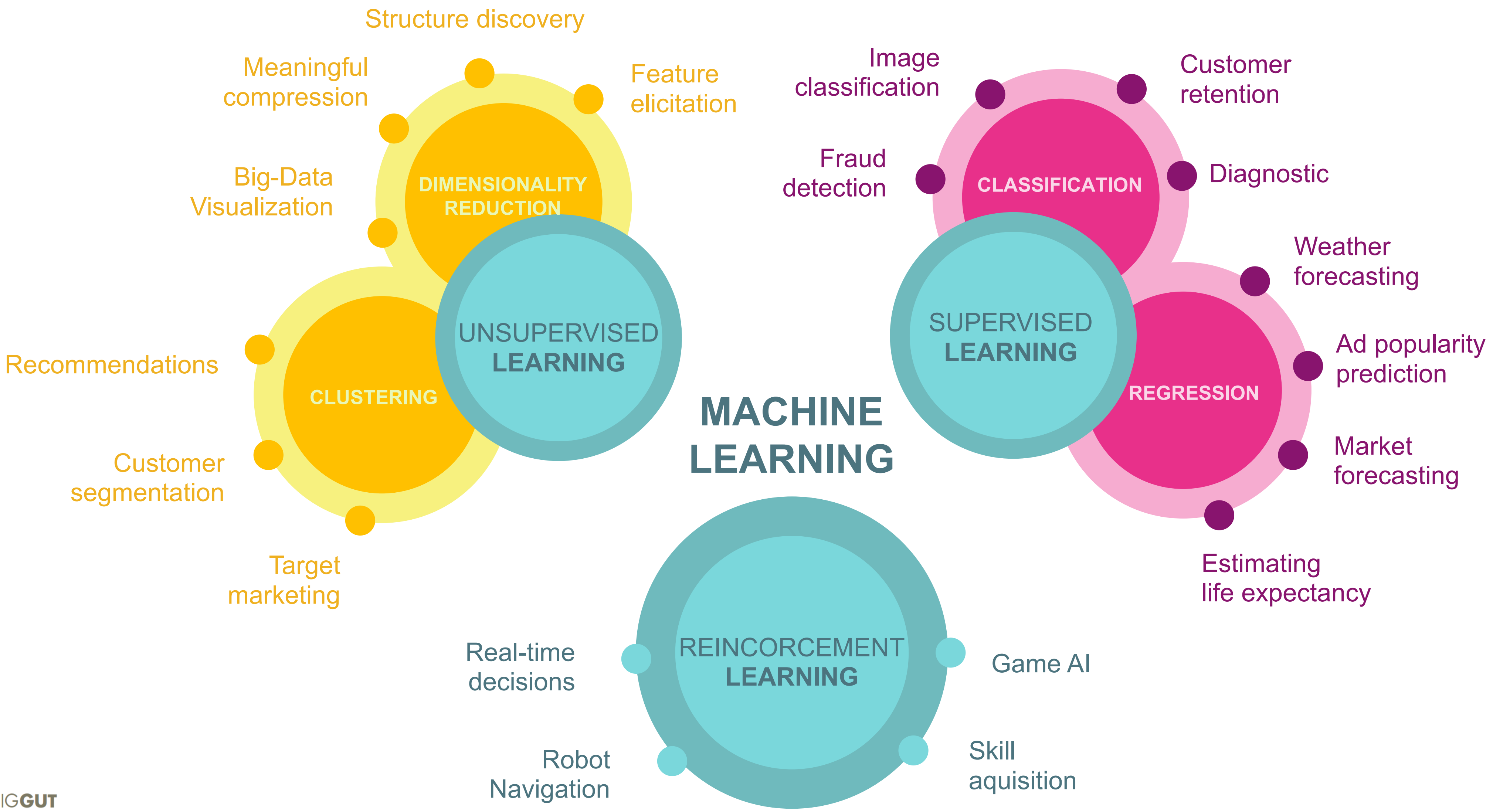

Data Sets
Data Retrival
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate until satisfactory model performance

Analysis & Modeling

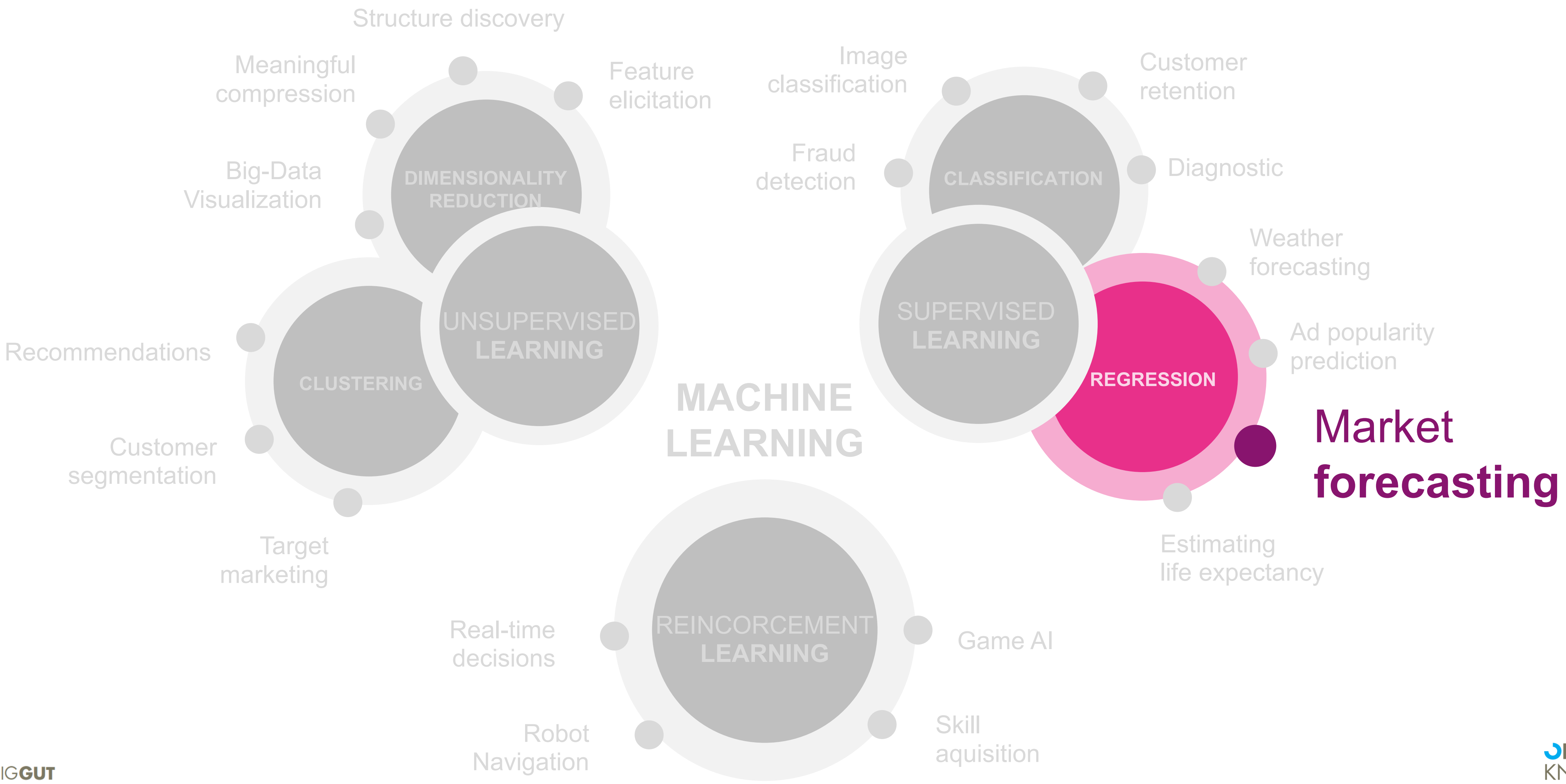
Germanys Next Top Model?

Easier said than done!
Where is the best place to start?
And how?

Analysis & Modeling



Analysis & Modeling



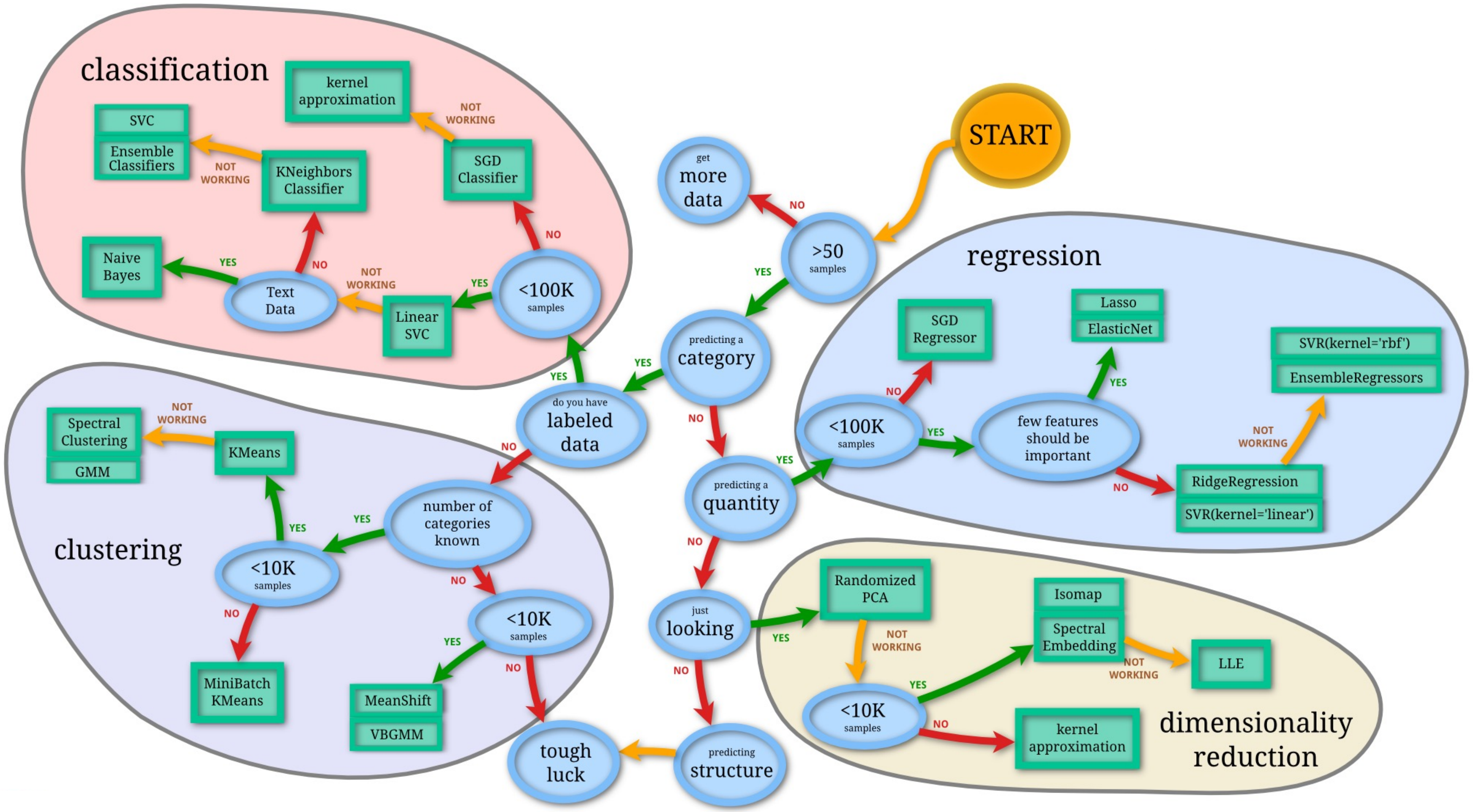
Analysis & Modeling

ML Regressors

If **values for the future** are to be predicted on the basis of values from the past, **regression algorithms** are usually used.

Analysis & Modeling

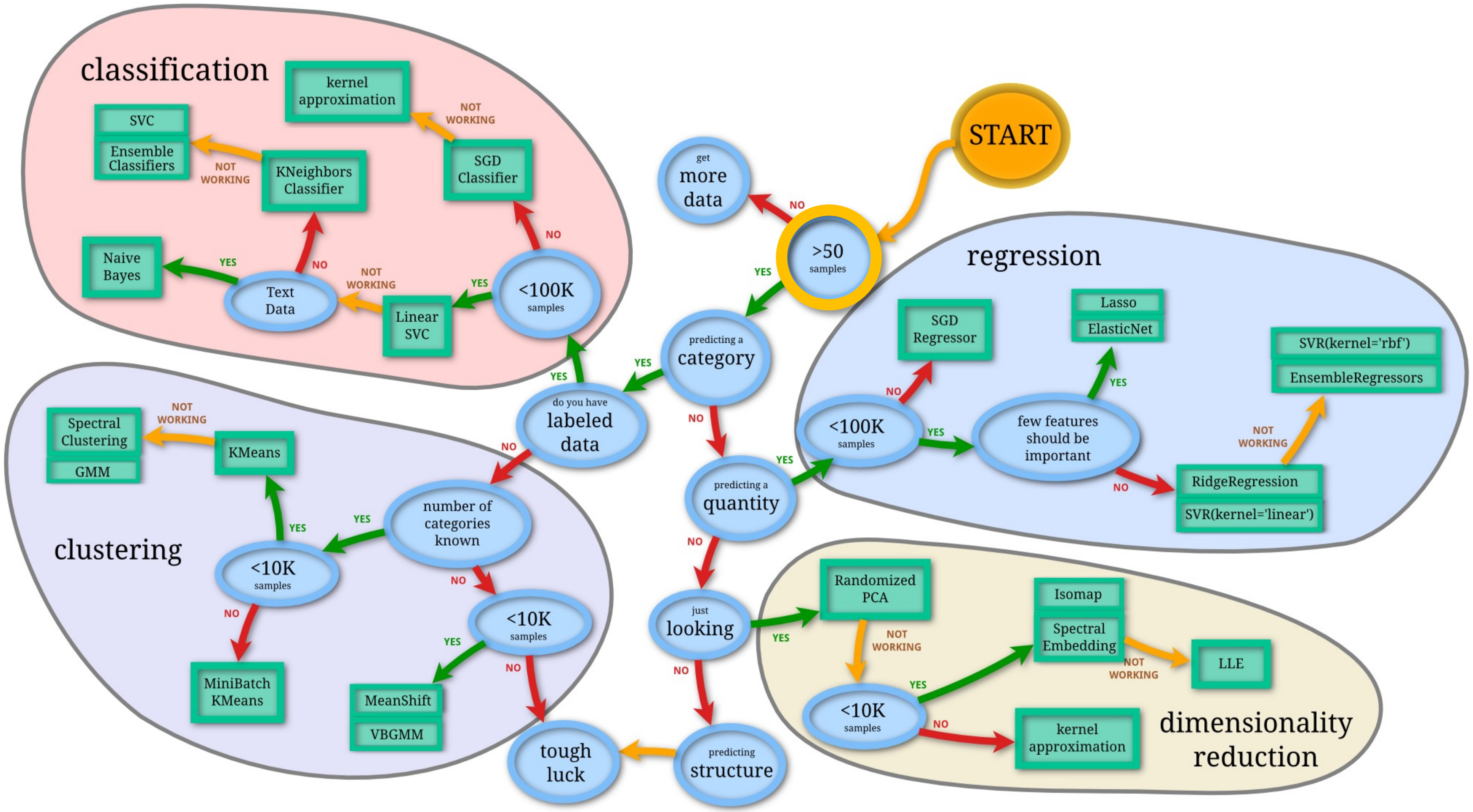
scikit-lean
Algorithm
Cheat-Sheet



Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling

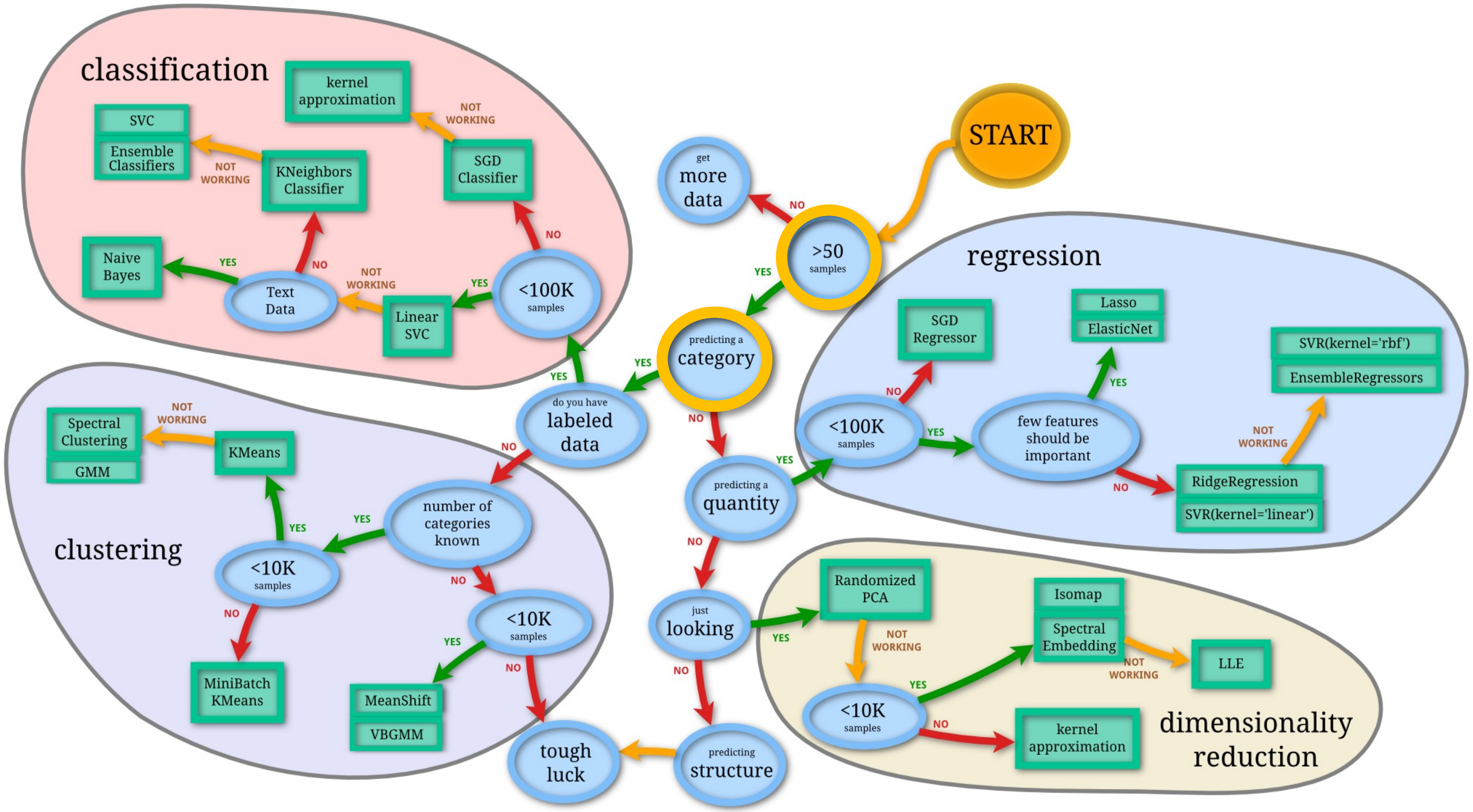
scikit-lean
Algorithm
Cheat-Sheet



Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling

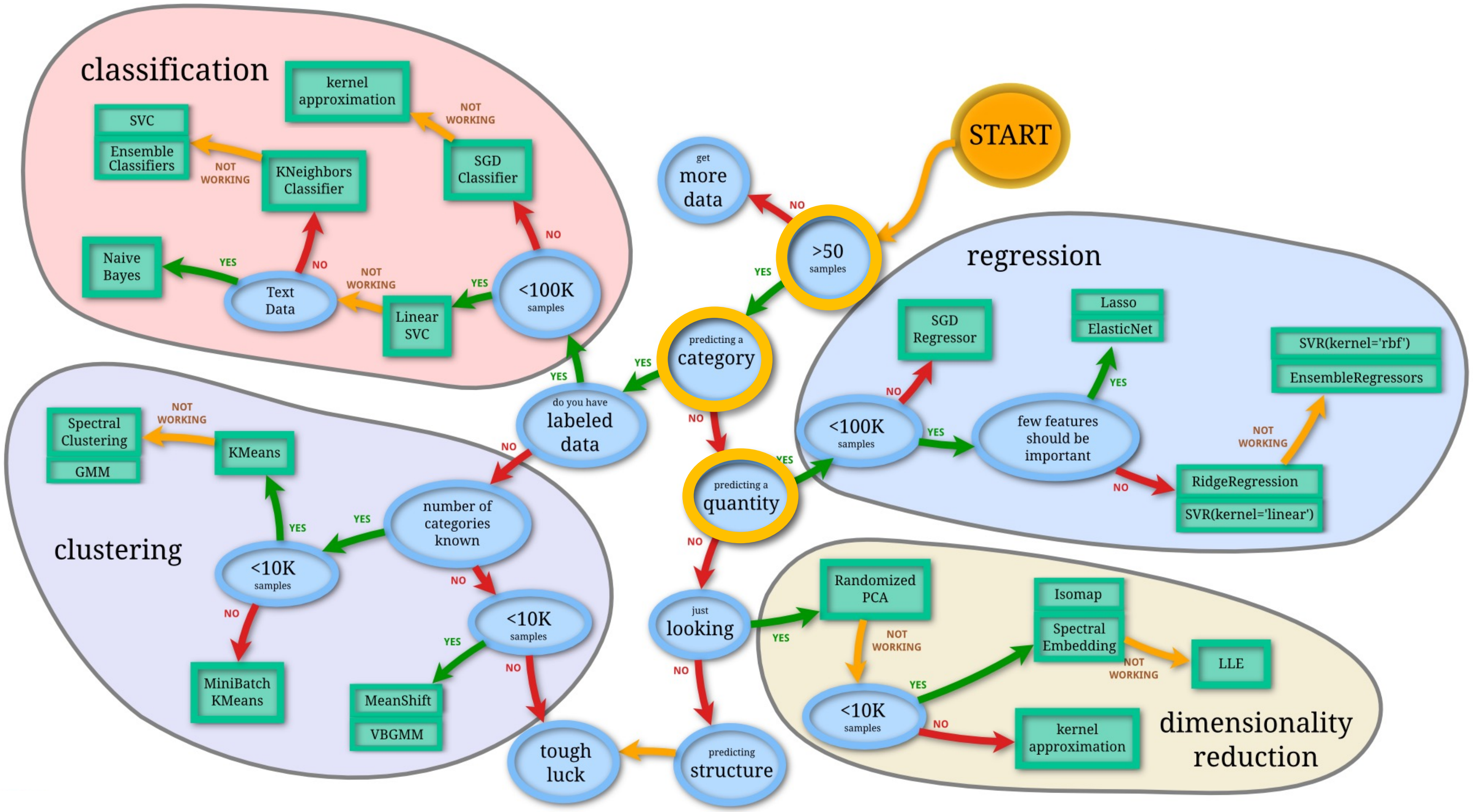
scikit-lean
Algorithm
Cheat-Sheet



Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling

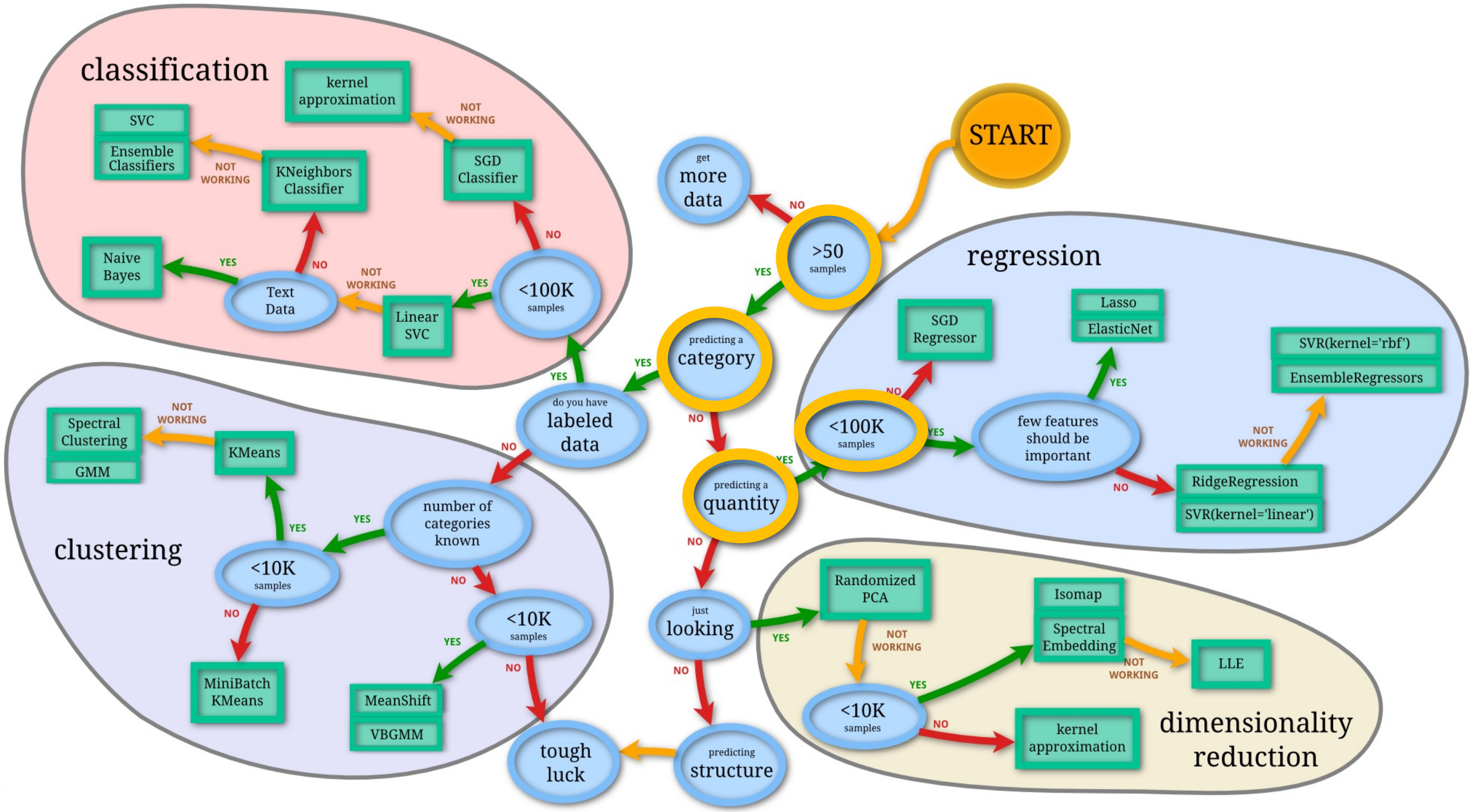
scikit-lean
Algorithm
Cheat-Sheet



Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling

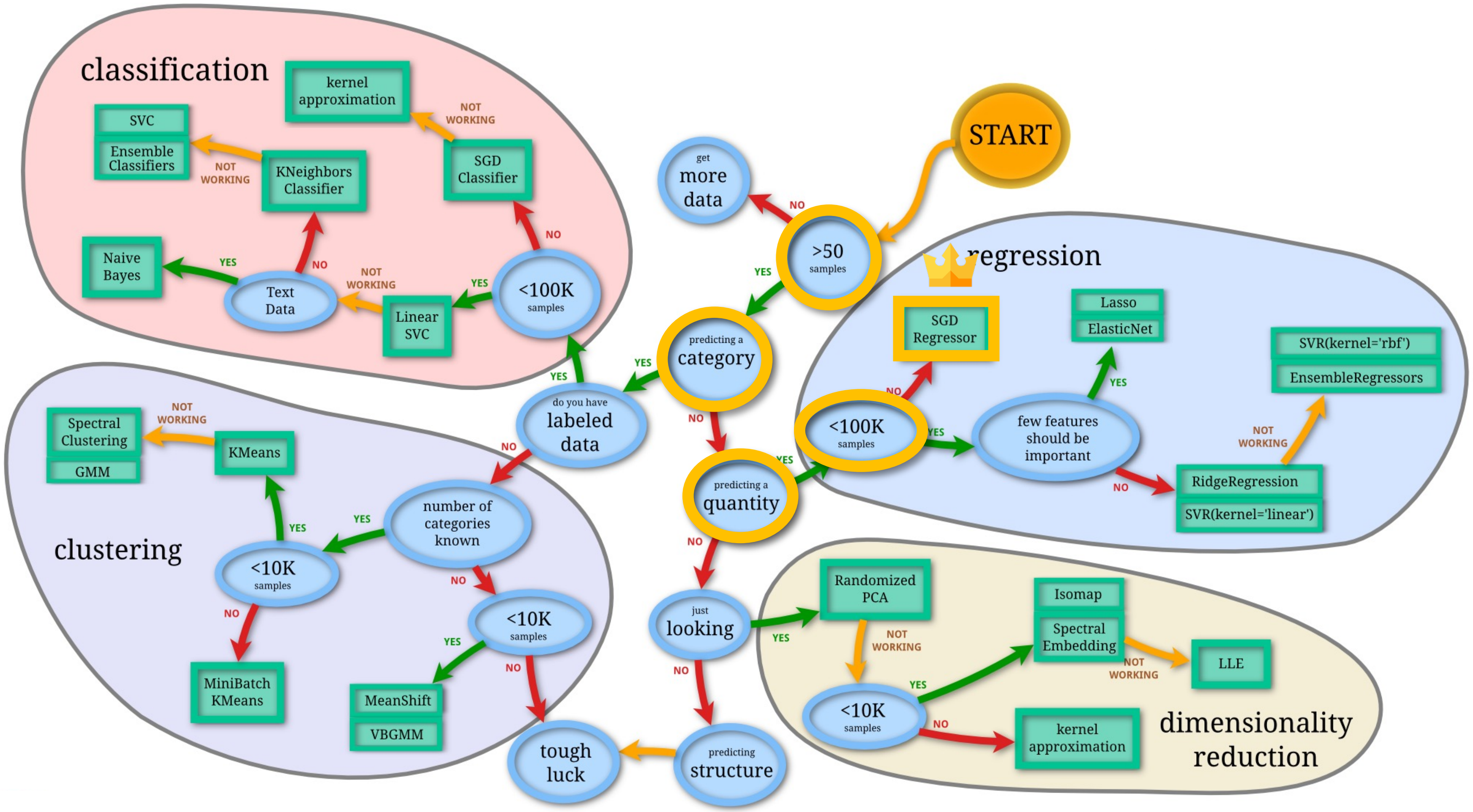
scikit-lean
Algorithm
Cheat-Sheet



Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling

scikit-lean
Algorithm
Cheat-Sheet



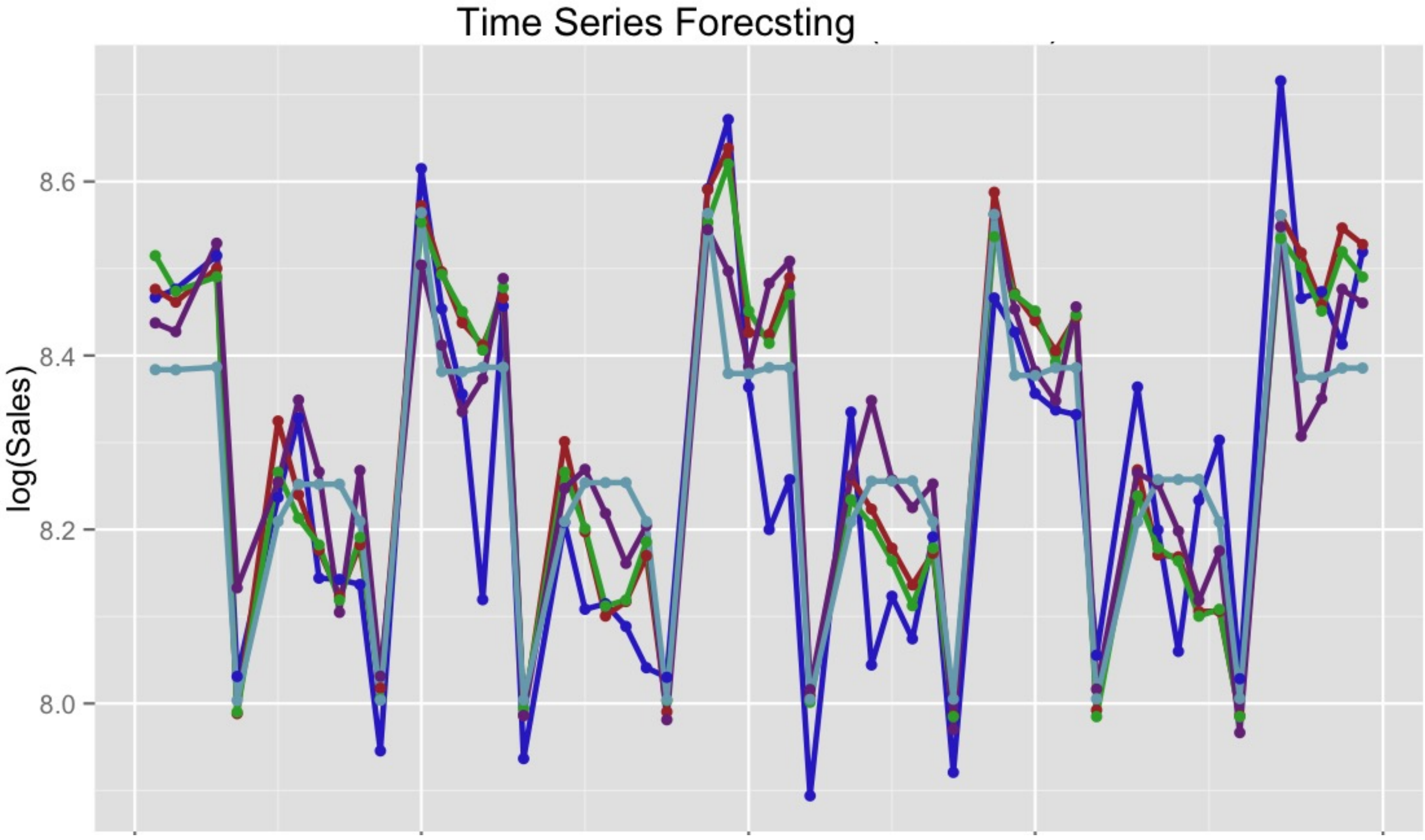
Quelle: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Analysis & Modeling
by Example

Models on the shortlist (Regressors)

- SGD Regressor (linear Regressor)
- Decission Tree Regressor (single Decision Tree)
- Random Forest Regressor (multiple parallel Decision Trees)
- xgBoost Regressor (multiple sequential Decision Trees)

Analysis & Modeling
by Example

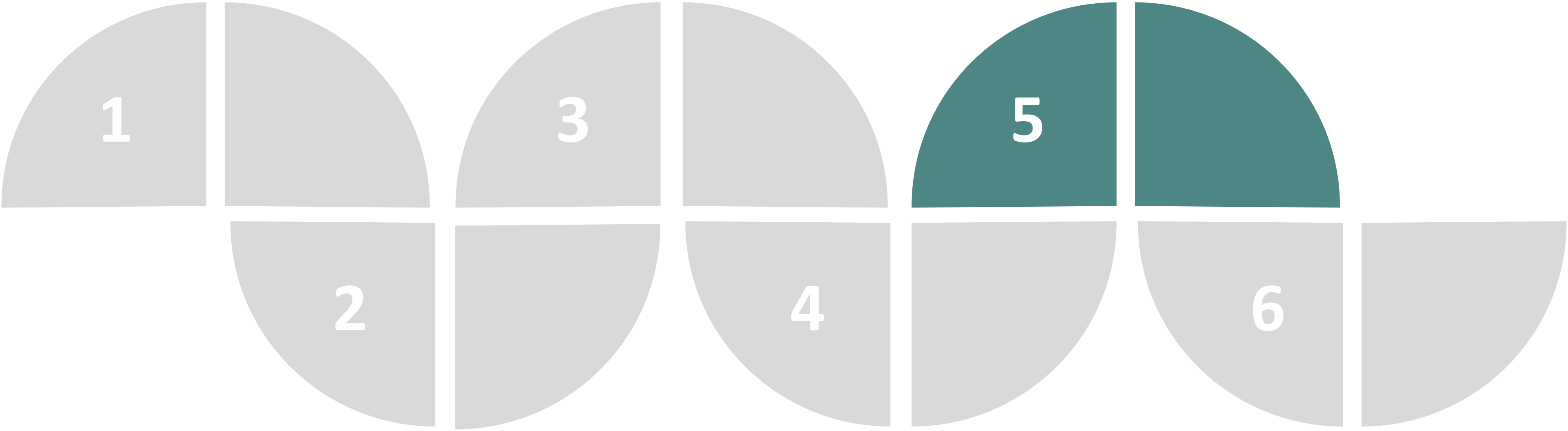


- **Time Series**
- **SGD** (RMSPE = 0.234)
- **Decission Tree** (RMSPE = 0.164)
- **Random Forest** (RMSPE = 0.139)
- **xgBoost** (RMSPE = 0.121)

RMSPE
Root Mean Square Percentage Error

Quelle: https://www.datasciencecentral.com/linear-machine-learning-and-probabilistic-approaches-for-time/

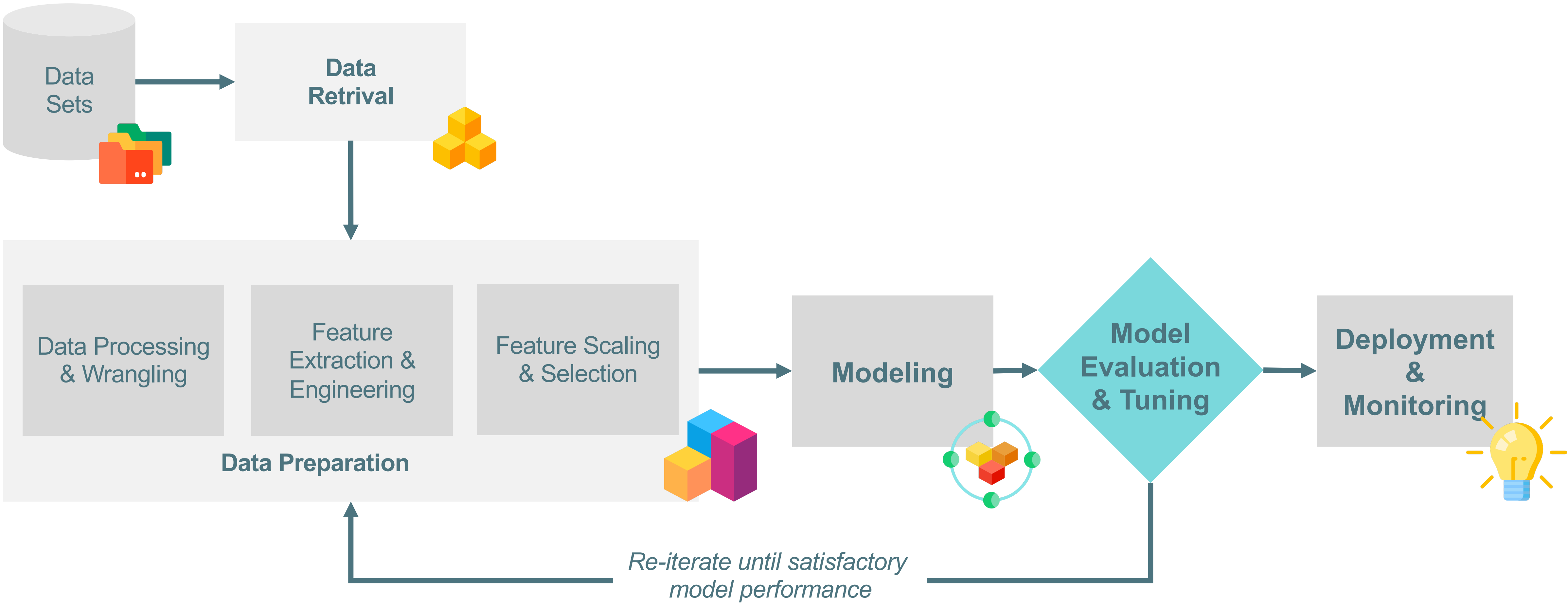
QA & Validation



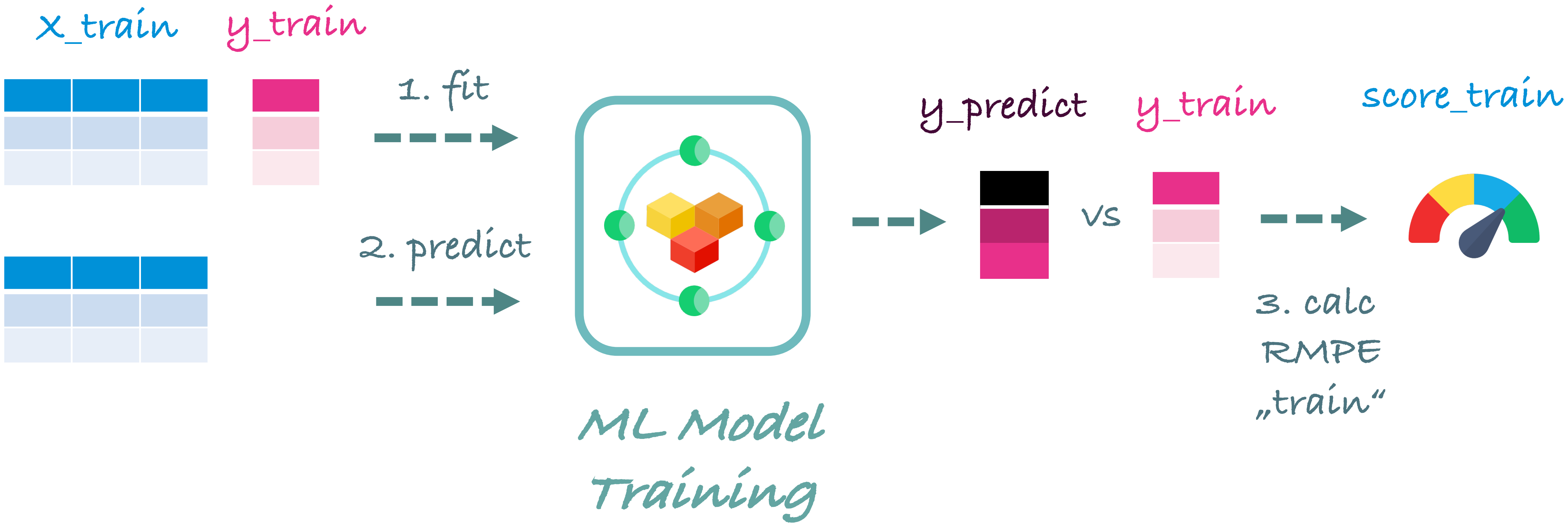
„ Is GOOD also GOOD enough?“

Model Quality vs. Project Goal

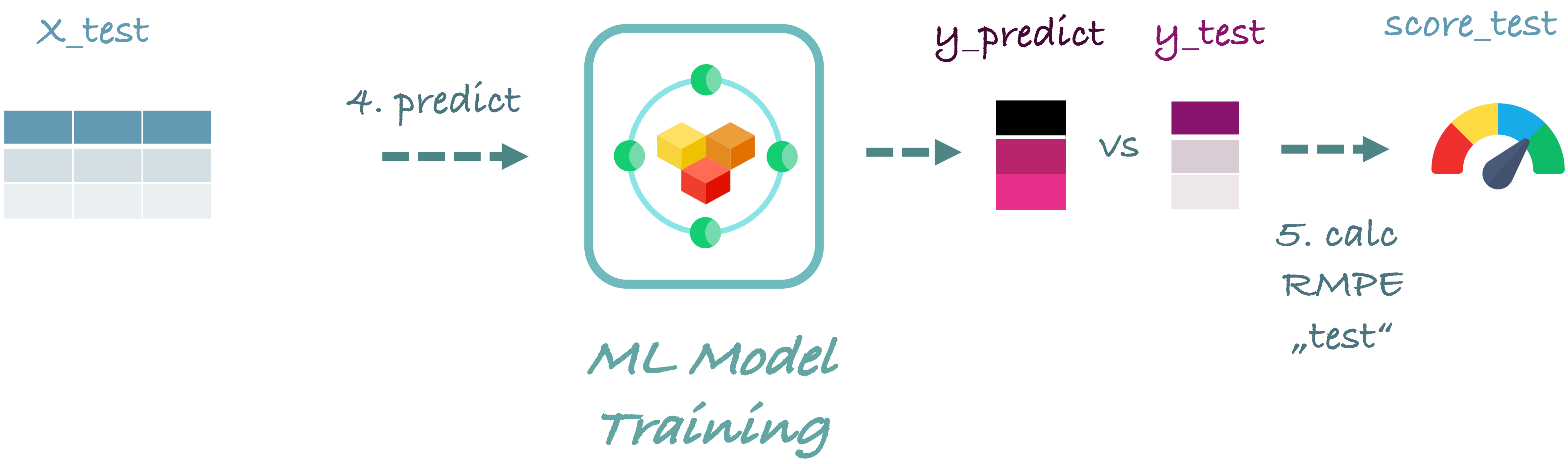
Analysis & Modeling


Data Sets
Data Retrival
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate until satisfactory model performance

QA & Validation
by Example



QA & Validation
by Example



QA & Validation

by Example

score_train ≈ score_test < X

QA & Validation

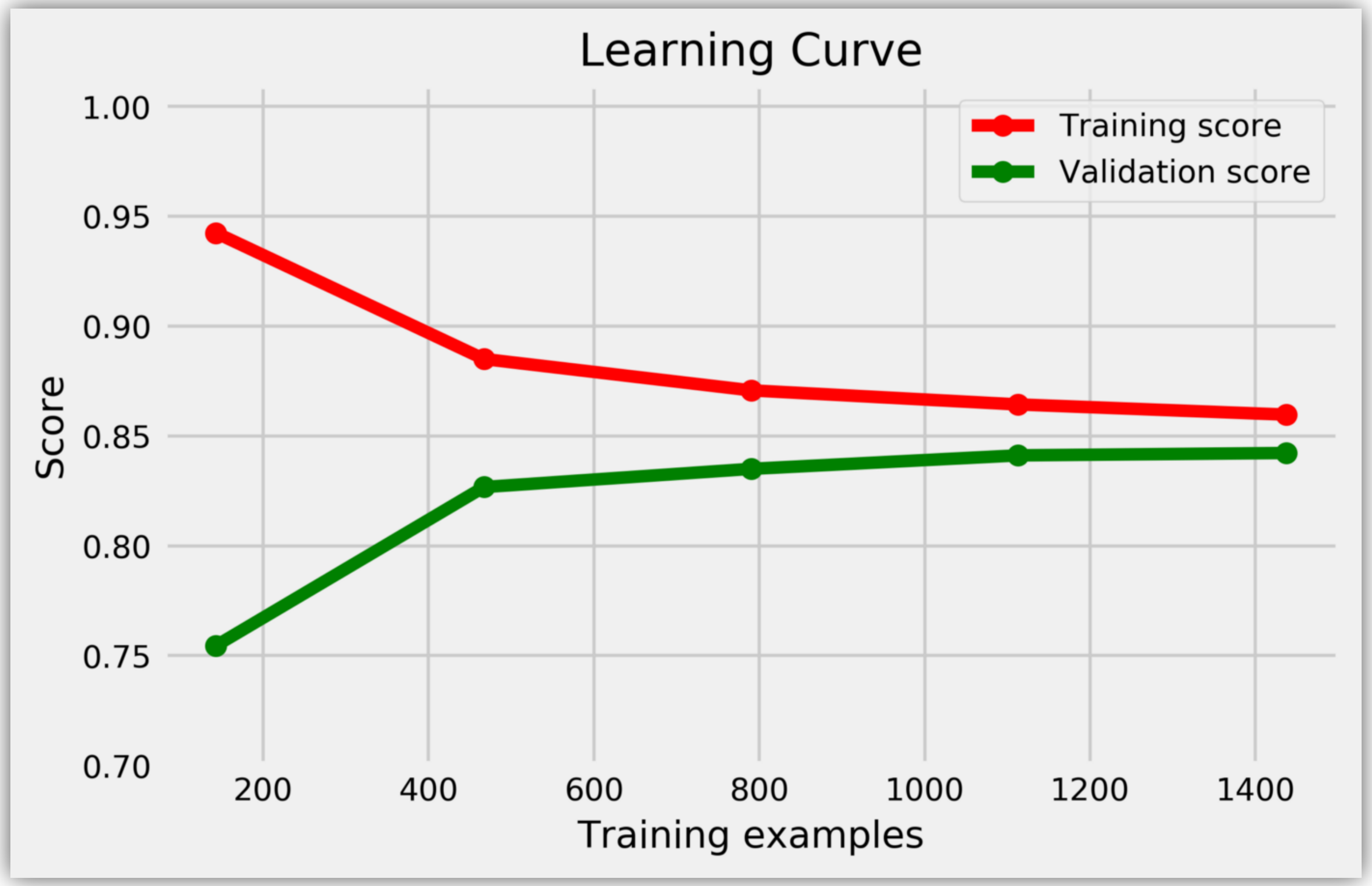
by Example

score_train ≈ score_test < X

analytisches Ziel

"A machine learning solution should predict sales per store for the period of 6 weeks, with an accuracy of [X]%."

QA & Validation



Quelle: https://towardsdatascience.com/a-short-introduction-to-model-selection-bb1bb9c73376

QA & Validation

Challenge of the "right" data split

To **evaluate the prediction quality** (aka performance) of the trained model, the training score and the test score are compared.

To ensure that the selected data split did not lead to a good result by accident, several **cross-validations** are performed.

QA & Validation

Cross-Validation

Testing the ML model performance with different splits :

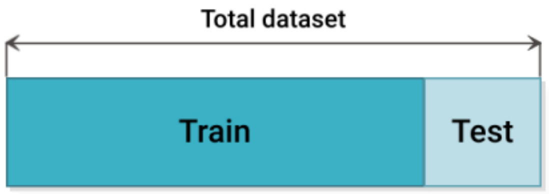
- Hold-out
- K-folds*
- Leave-x-out**
- Time Series CV

* x = „one“ oder „p“

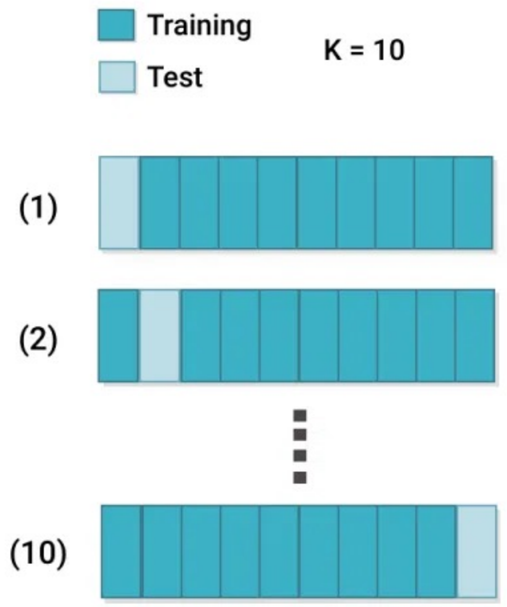
** x = „Stratified …“, „Repeated …“ oder „Nested…“

QA & Validation

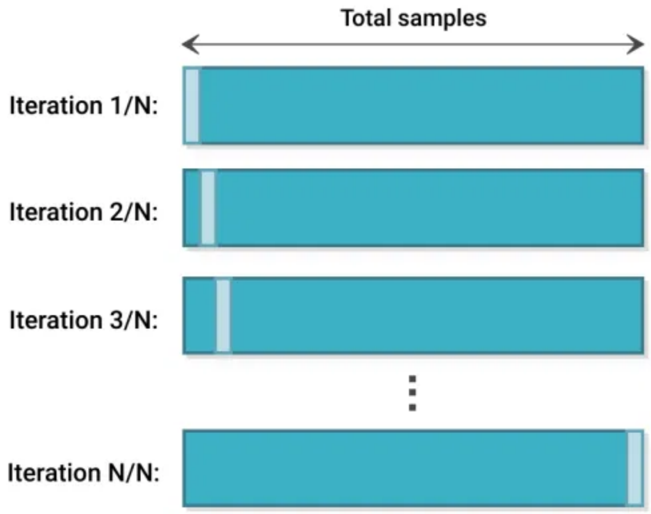
Cross-Validation



Hold-out



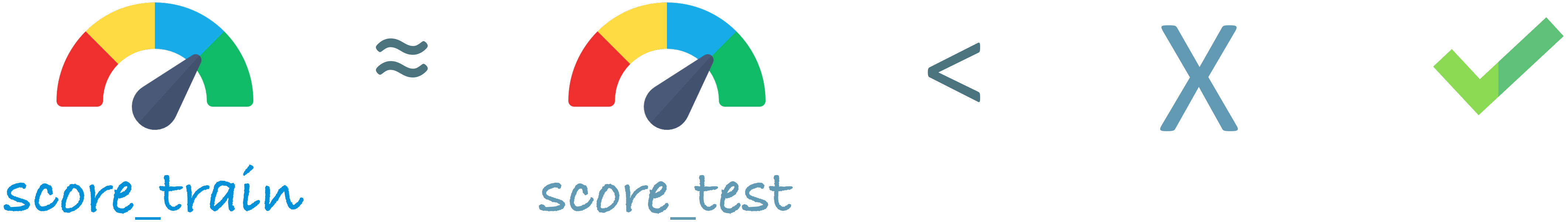
K-Folds



Leave-out

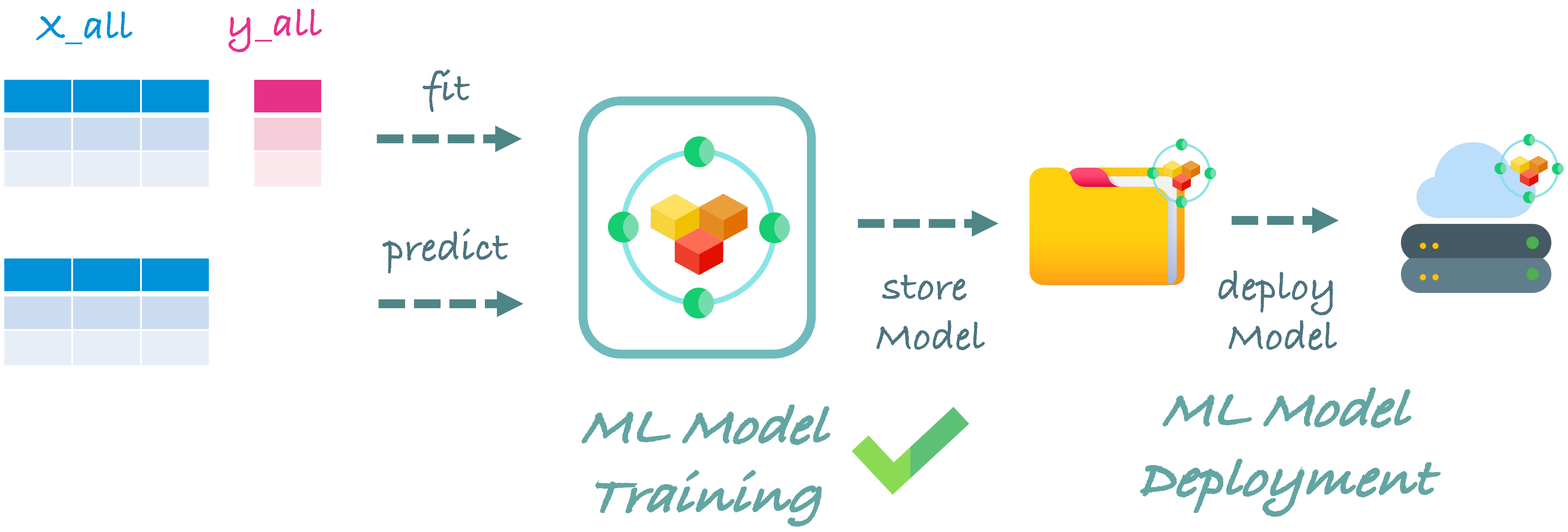
QA & Validation

by Example

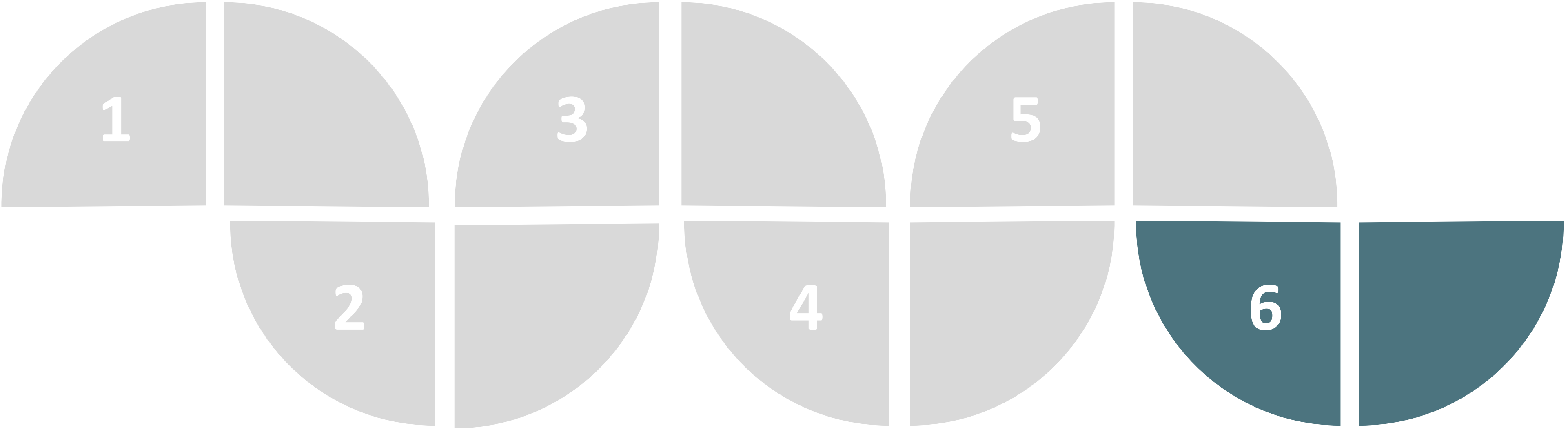


score_train ≈ score_test < X ✓

QA & Validation
by Example



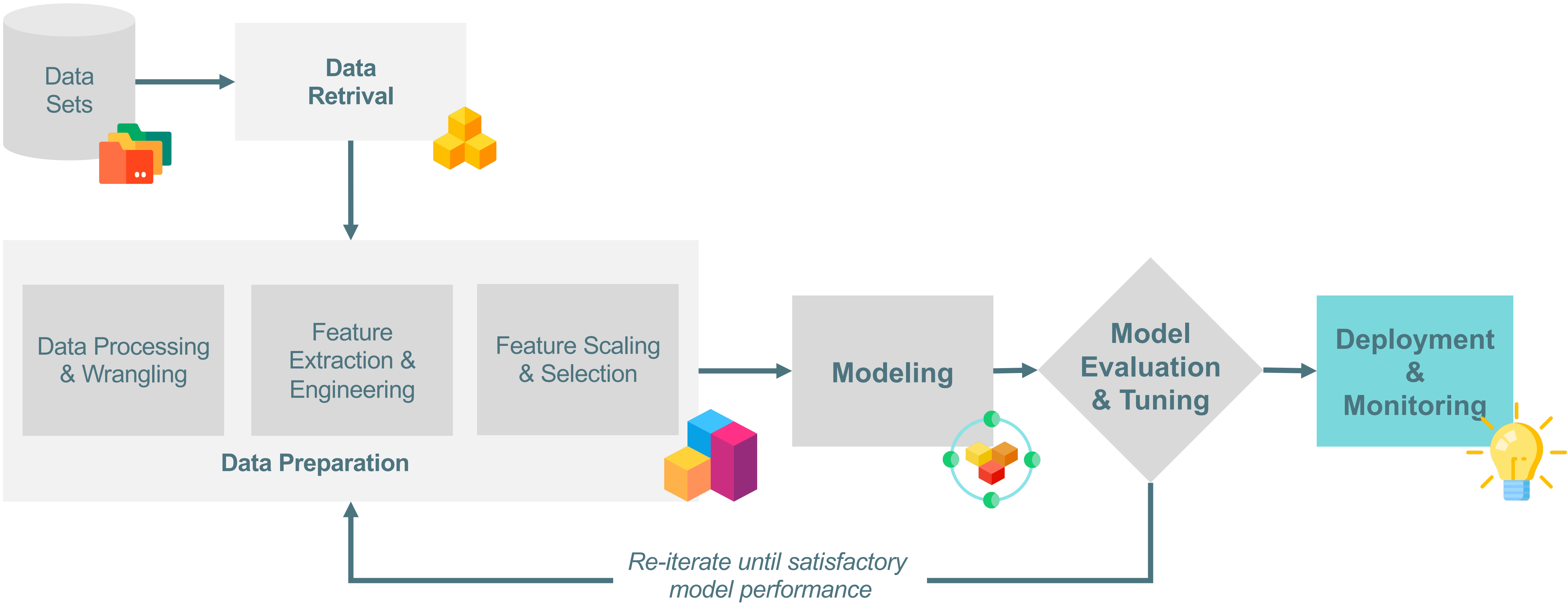
Deployment & Operation



„ ATTENTION! This is not an exercise!“

Model Deployment & Performance Monitoring

Deployment & Operation



Data Sets
Data Retrival
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate until satisfactory model performance

OFFENKUNDIGGUT

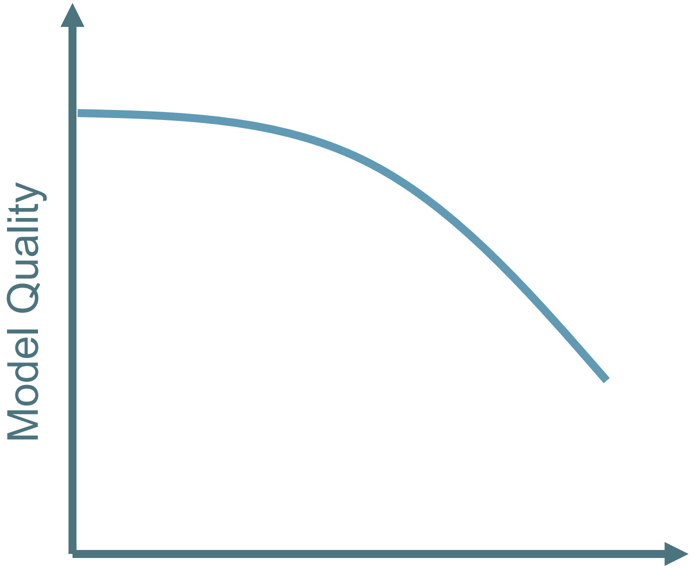
OPEN KNOWLEDGE

*“**Change** is the only constant in life.“*

Heraclitus, Greek philosopher

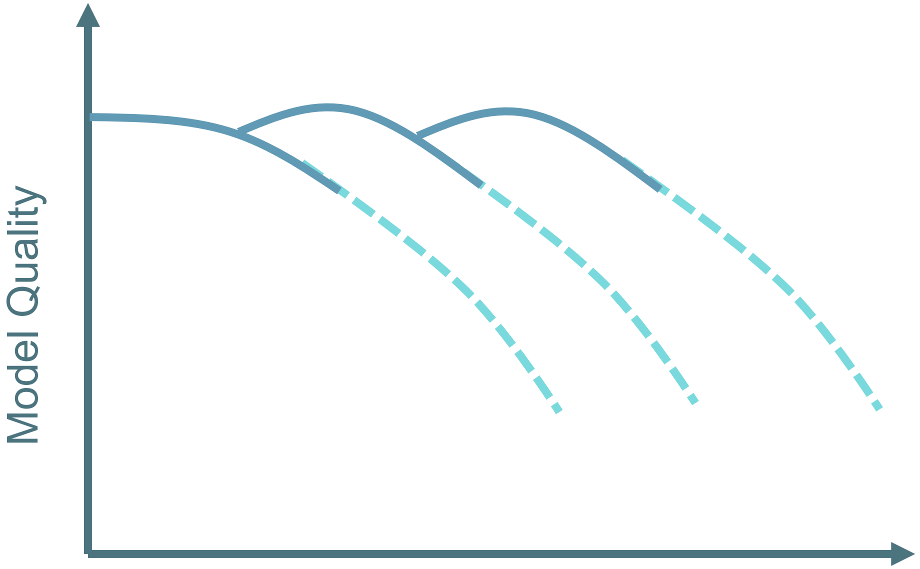
Deployment & Operation

Static Model



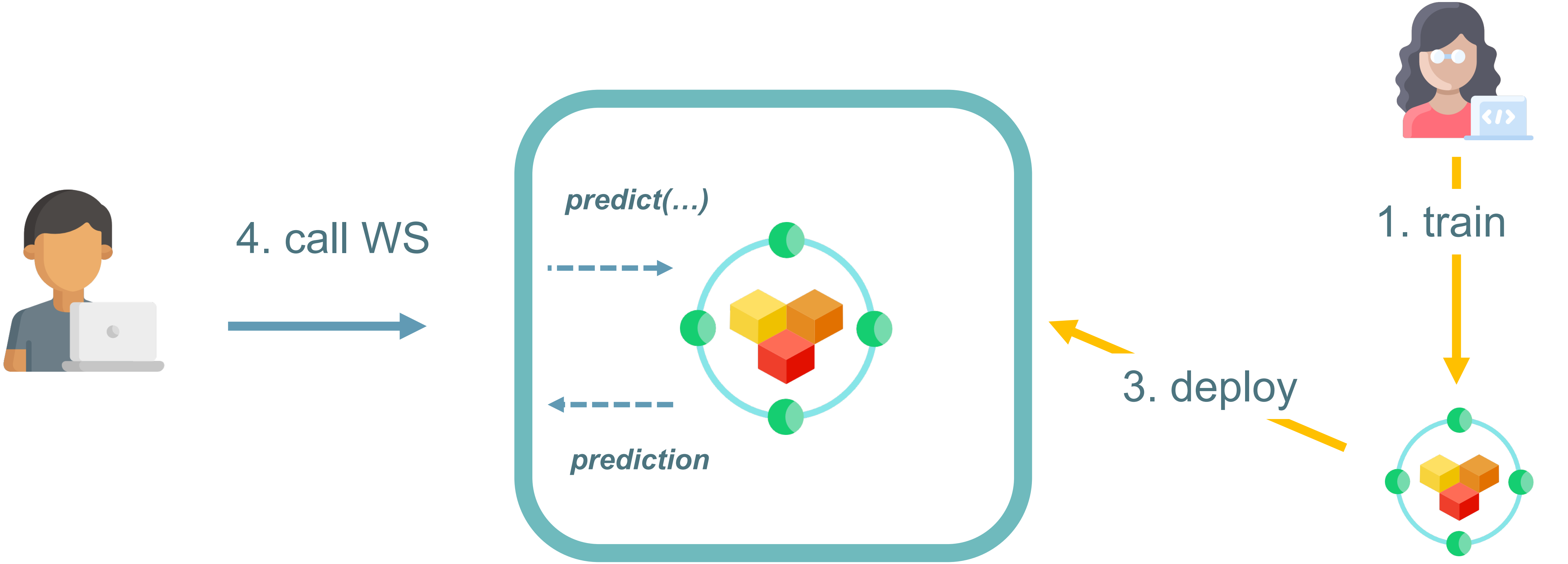
Model decay over time

Refreshed Model



Regulary updated model

Deployment & Operation



4. call WS

predict(…)

prediction

1. train

3. deploy

2. validate

ML based Web Service

CONTINUOUS DELIVERY

*“…the ability to **get changes of all types** — including new features, configuration changes, bug fixes, and experiments — **into production**, or into the hands of users, safely and quickly in a sustainable way.“*

Jez Humble & Dave Farley

Deployment & Operation
by Example

MLops aka CI / CD Pipeline for Changes

Data

Schema
Sampling over Time
Volume

Model

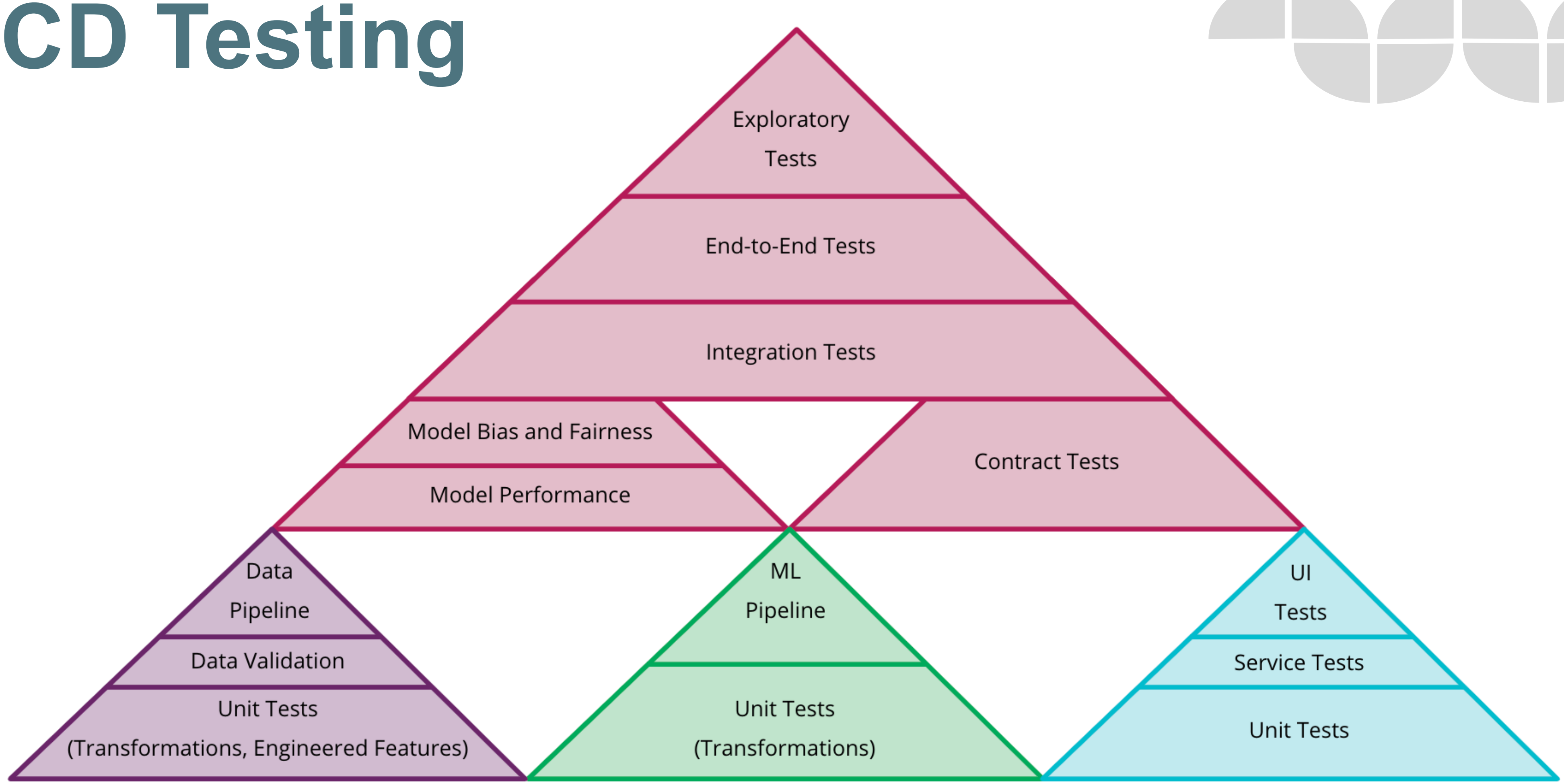
Algorithms
More Training
Experiments

Code

Business Needs
Bug Fixes
Configuration

The 3 axis of change in ML apps – data, model, and code – and a few reasons for them to change.

CI/CD Testing



Stack of Test Pyramids
Example of how to combine different test pyramids for data, model, and code in CD4ML

Source: https://martinfowler.com/articles/cd4ml.html

Deployment & Operation
by Example

Monitoring the Model

Model Performance

Model Input/Output Distribution

Model Learning Curves

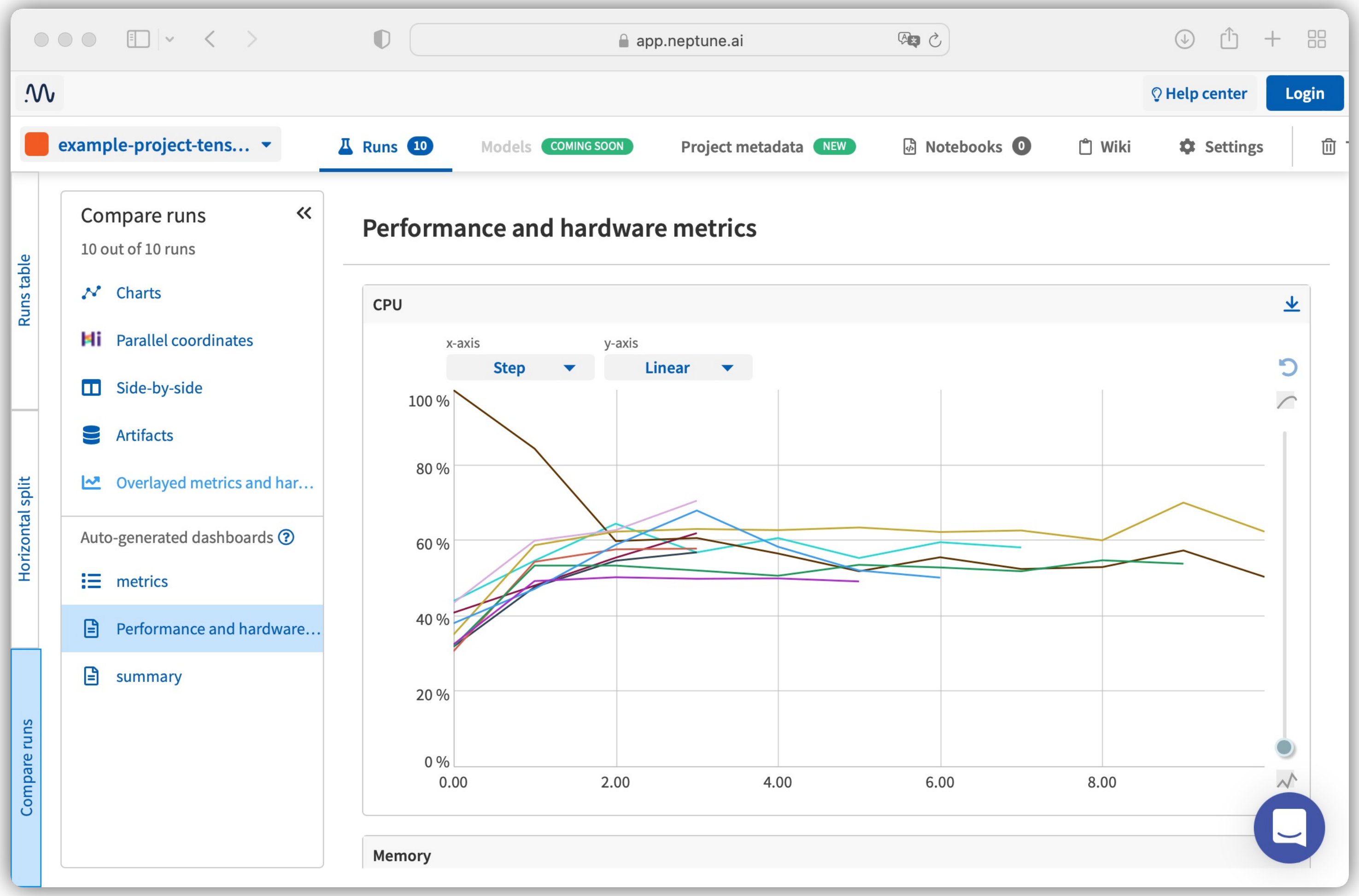
Model Evaluation Metrics

Model QA Results

Hardware Metrics

CI/CD Pipeline for ML

ML Monitoring



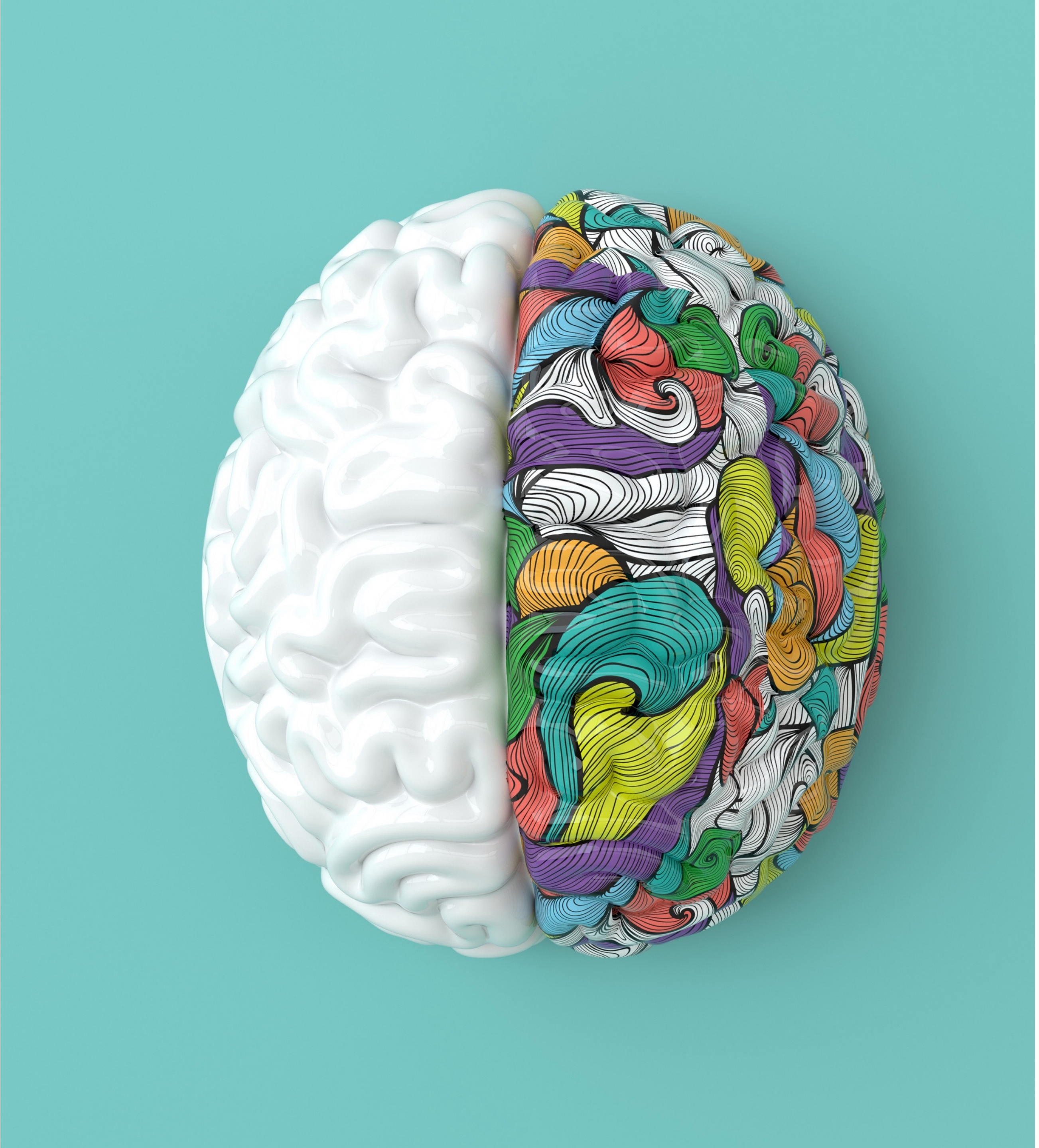
Performance and hardware metrics

CPU

x-axis Step

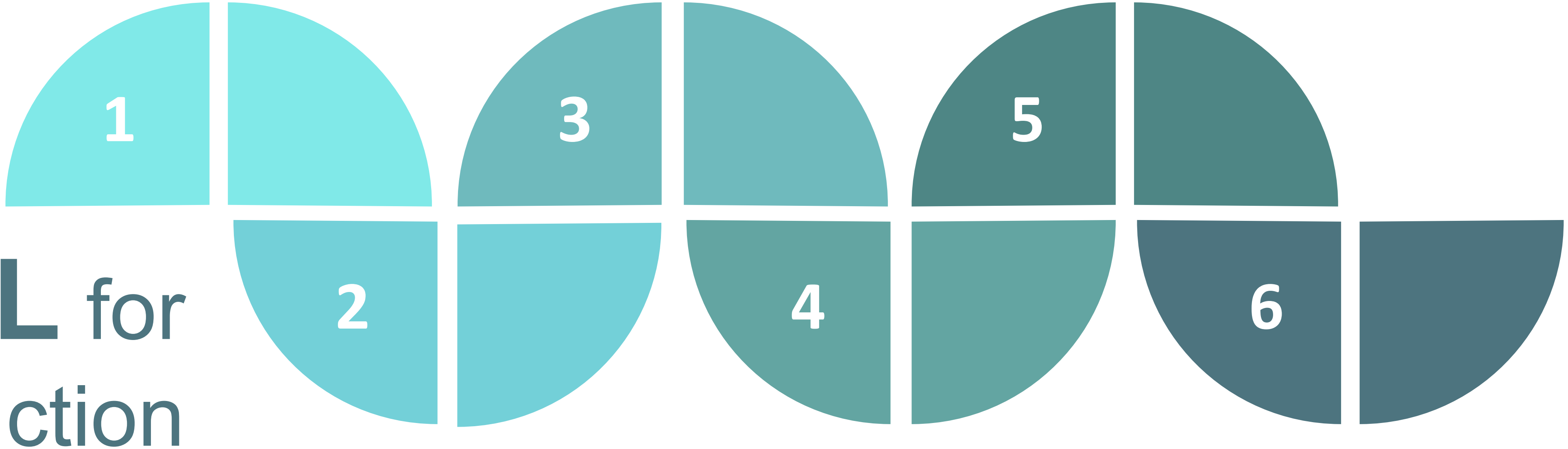
y-axis Linear

Memory



Take aways!

ML for Production



Business Understanding

Data Understanding

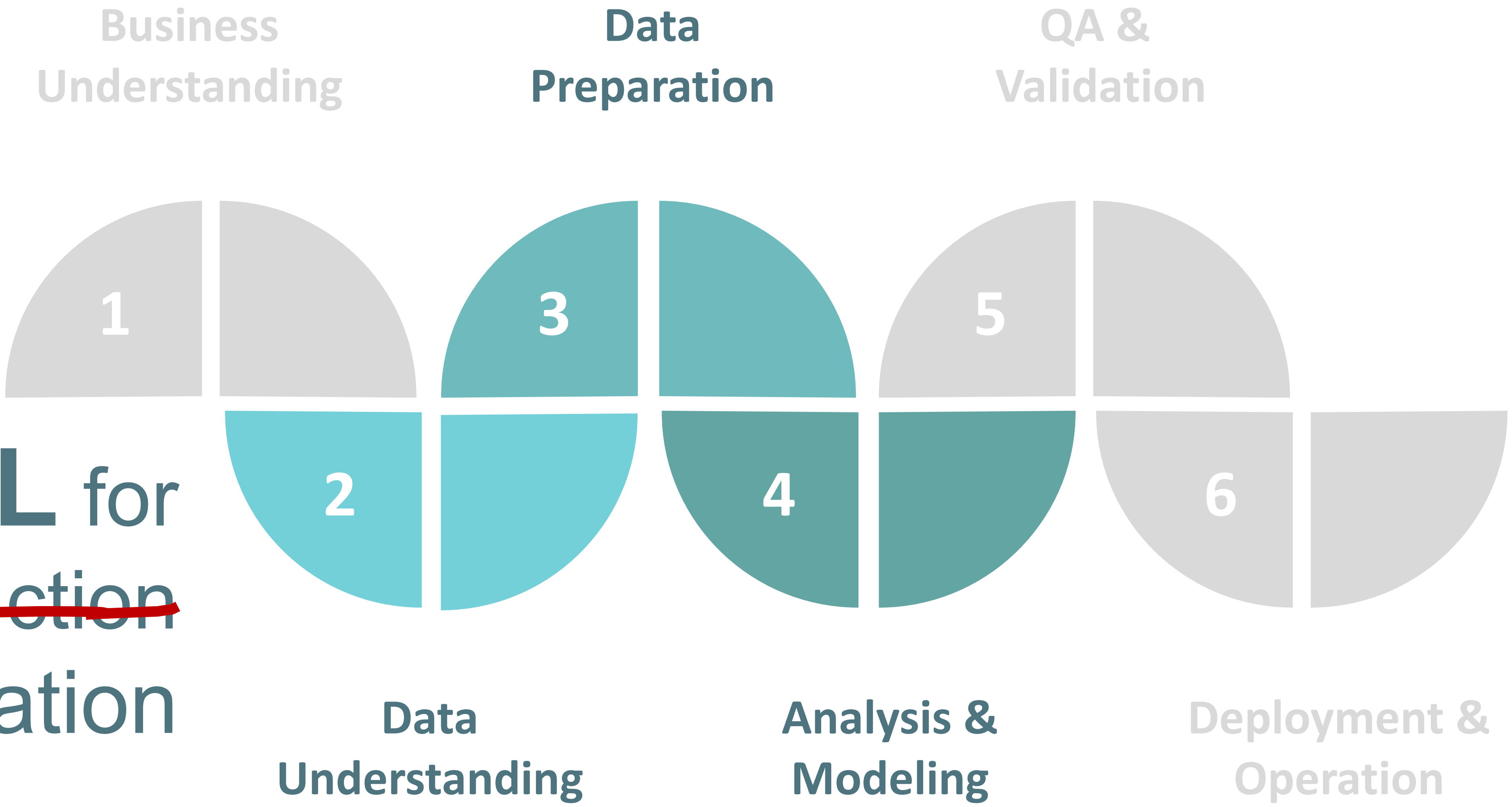
Data Preparation

Analysis & Modeling

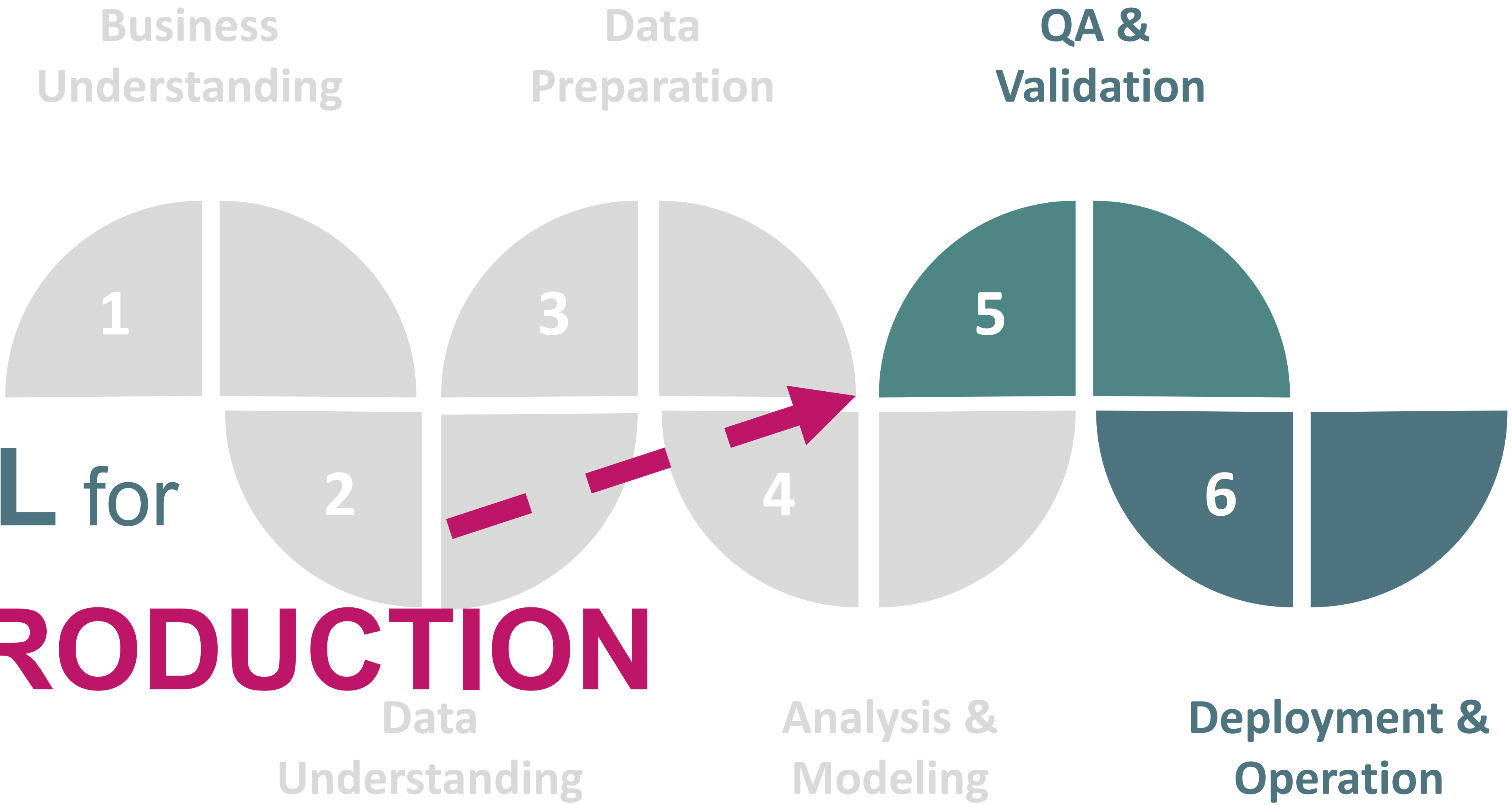
QA & Validation

Deployment & Operation

ML for ~~Production~~ Exploration



ML for PRODUCTION



Business Understanding

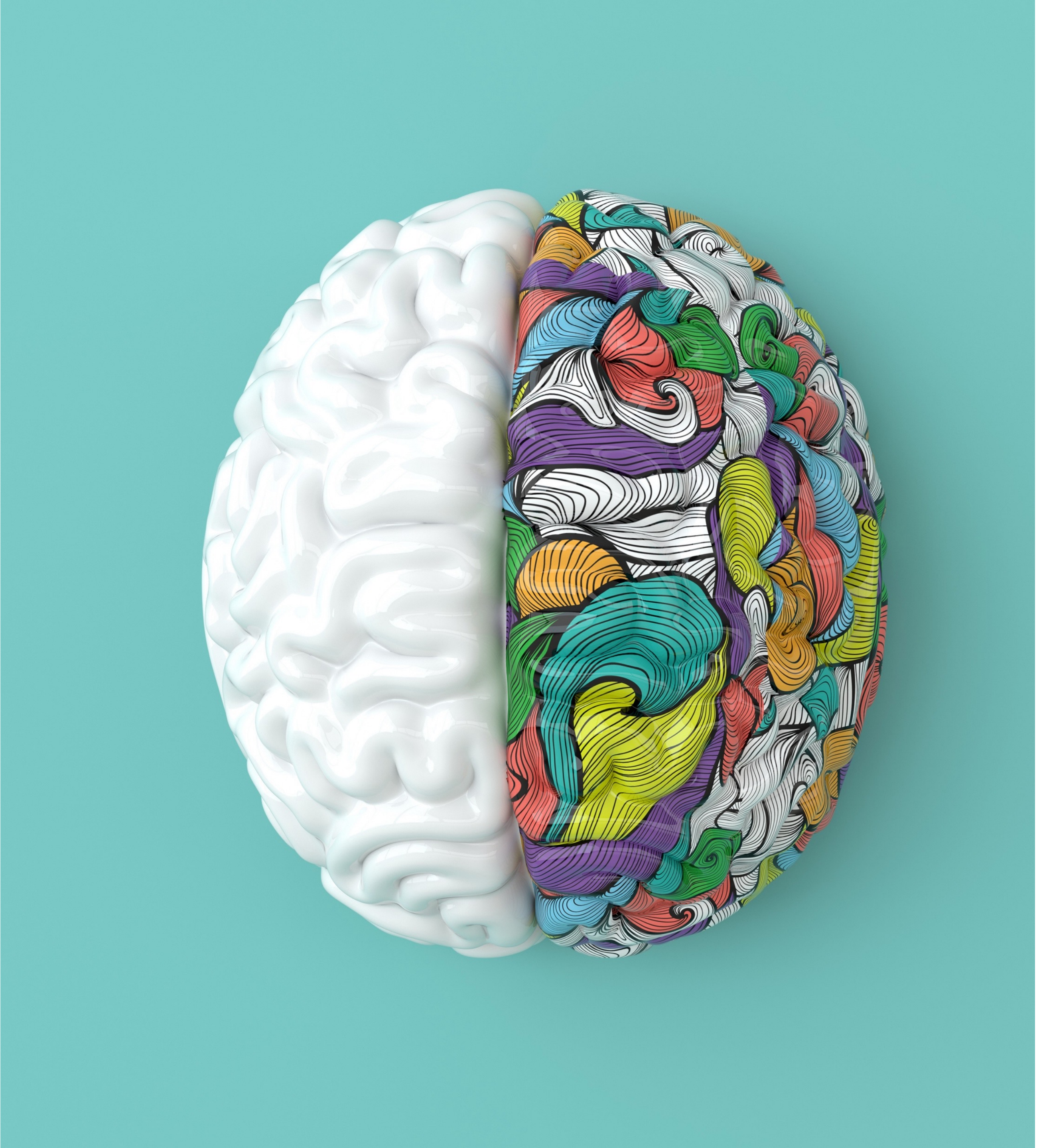
Data Preparation

QA & Validation

Data Understanding

Analysis & Modeling

Deployment & Operation



Time for Questions? YES!

OPEN KNOW LEDGE

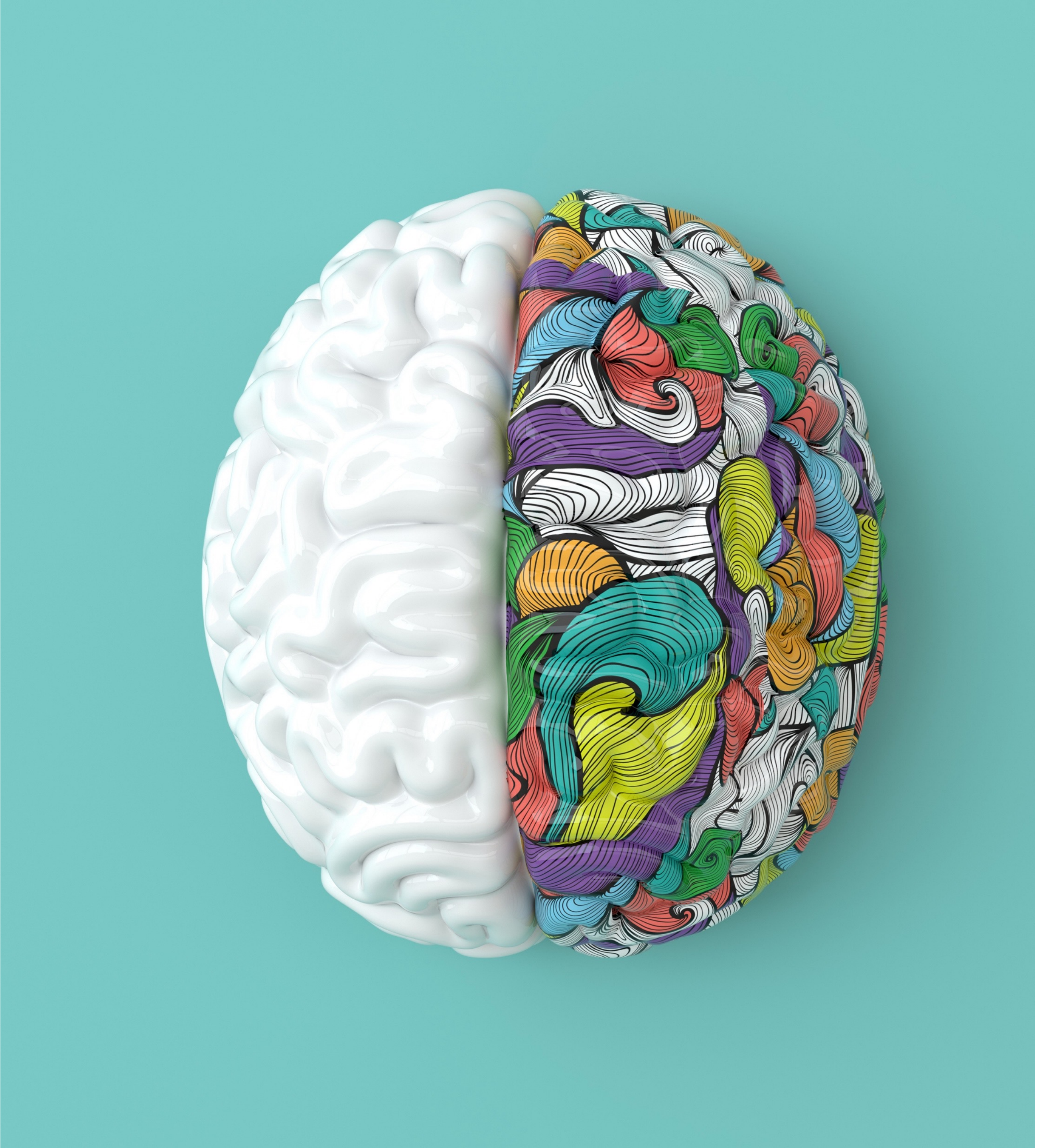
#WISSENTEILEN

by **open** knowledge **GmbH**

Thank You!

Lars Röwekamp, CIO New Technologies

@_**open**Knowledge | @**mobile**Larson



USED IMAGES

Folie 01: © Merena, iStockphoto.com

Folie 20: © Photoplotnikov, iStockphoto.com

All other pictures, drawings and icons originate from

- **pexels.com**,
- **pixabay.com**,
- **unsplash.com**,
- **flaticon.com**

or were made by my own.