Designing an ML use case E2E

05/05/2023

About myself

2008

UNamur

Bachelor and master in Mathematics
Master in Economics in Turing-Erasmus in Alicante & Padua

2014

ING

International traineeship -
Data scientist -
Data scientist in Germany

2017

Dataroots

Machine learning engineer -
AI tech lead -
Head of research

Highlights about Dataroots

Experienced niche player

Dataroots is a fast growing scale-up, specialized in end-to-end **data solutions**. We are team of +100 data experts.

Open-source mindset

Active and trusted open-source contributor to share knowledge and accelerate innovation via **frameworks** and **templates**. We don't start from zero.

AI Partner

We delivered more than **+200** successful projects for **+60** different clients.

Co-creation

Collaborative and co-creation with our clients in an **agile** way. We work both project and staffing-based. We **train** your people, actively share knowledge.

Cloud & technology agnostic

We have knowledge and experience in setting up cloud infrastructure and services from both **GCP**, **Azure** and **AWS**. Through this way, we remain impartial in every situation.

Azure amazon webservices Google Cloud

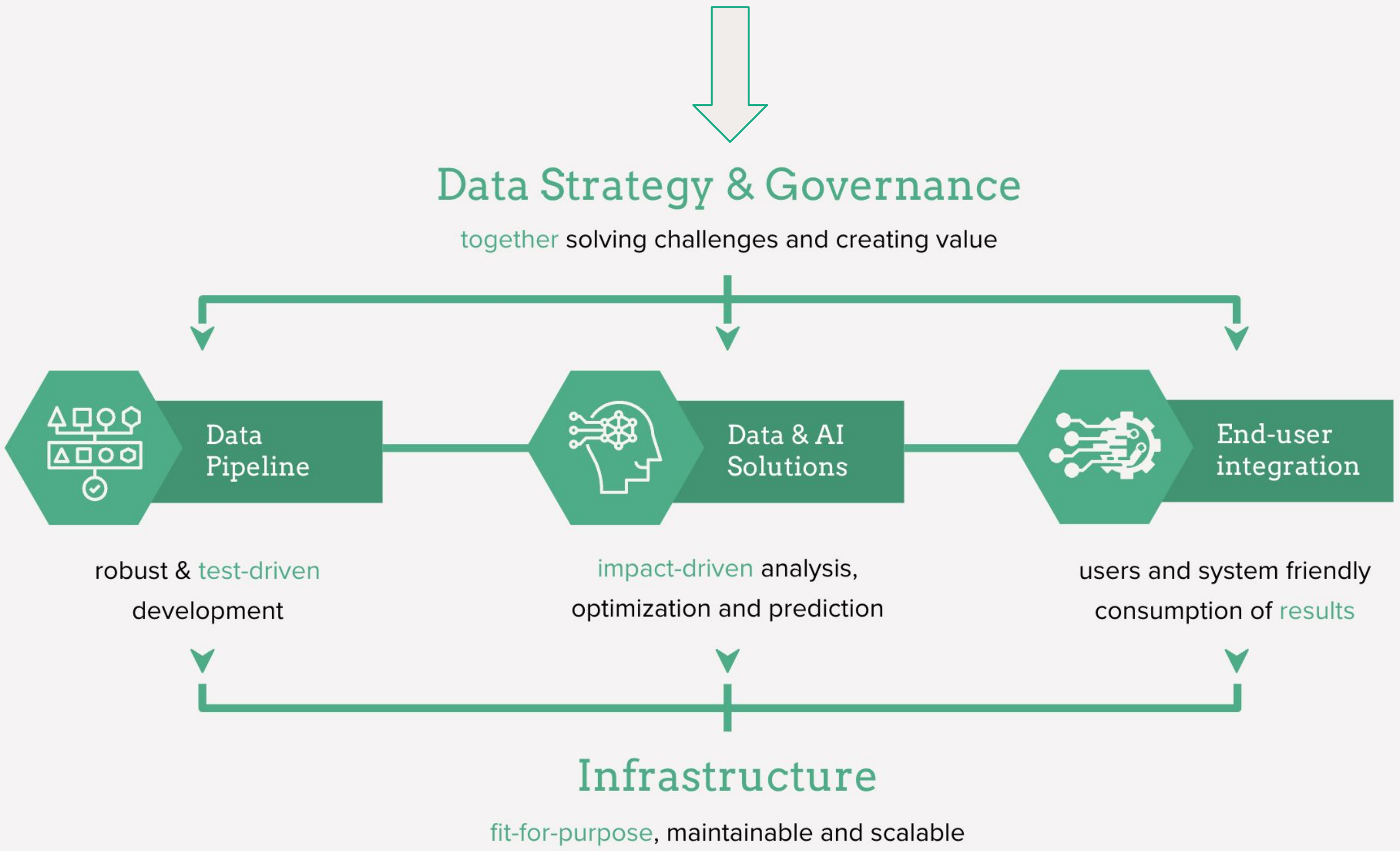
Dataroots research

By reinvesting 10% of the yearly revenue in applied **R&D** we assure our consultants have expertise and are opinionated on the **latest** cutting-edge machine learning, AI and cloud **tech stack.**

Our mission

On a mission

to deliver data-driven solutions with unrivalled longevity and business impact for our clients.



Innovation at dataroots

Objective

Monitor state of the art

Be a center of excellence in data & AI

10%

Of revenues invested in R&D

How

Continuous Training

Innovative projects

Community growth

Results

Internal knowledge

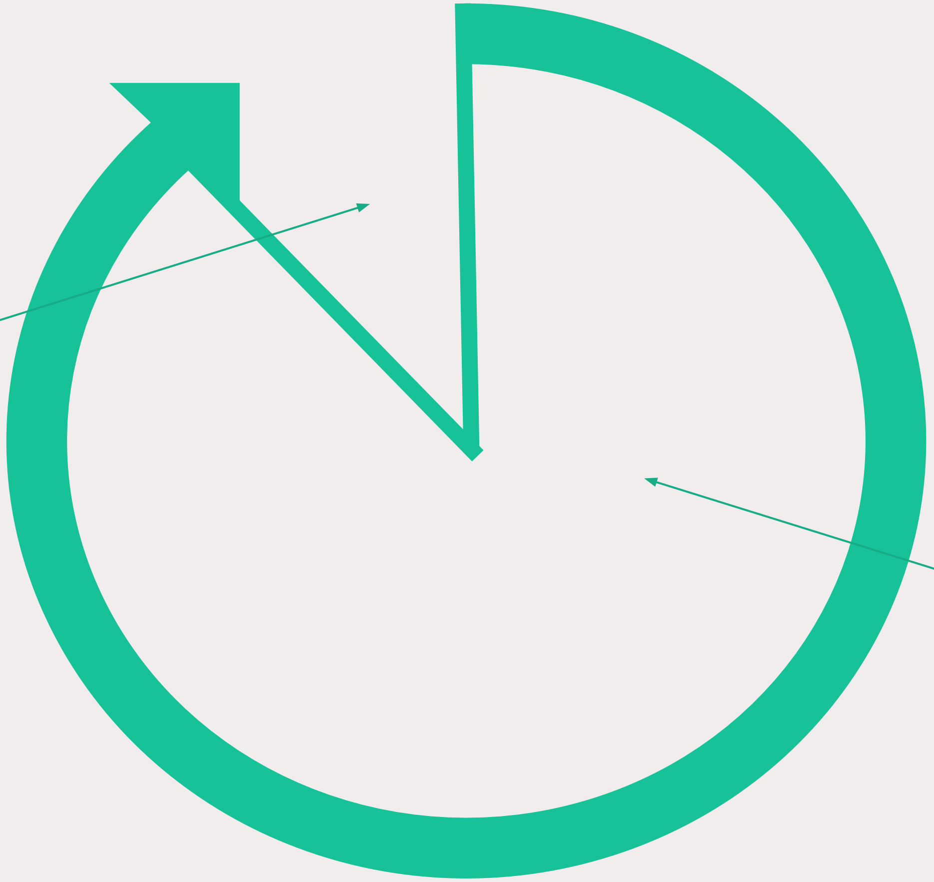
Innovative solutions

Open-source & data for good

The last 20% takes 80% of the work

In production AI

Mind blowing
State Of the Art
Easy to demo

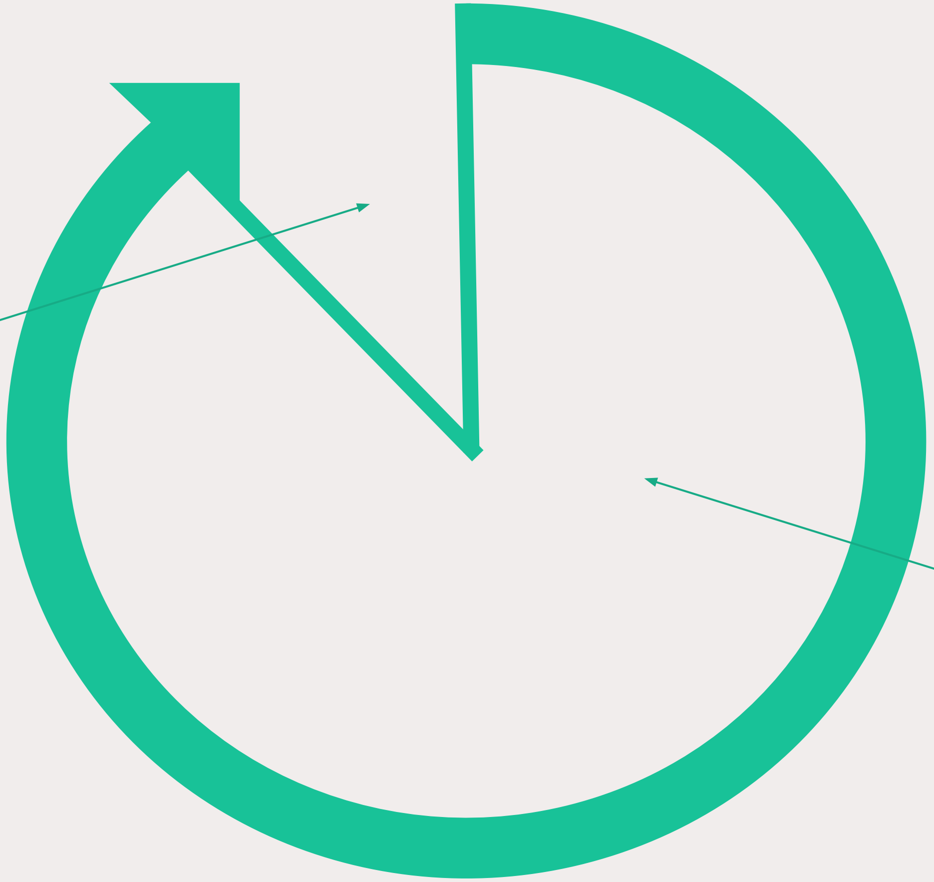


Depends on data
Human in the loop
Mix of heuristics

The last 20% takes 80% of the work

In production AI

Mind blowing
State Of the Art
Easy to demo



Depends on data
Human in the loop
Mix of heuristics

The agenda for today

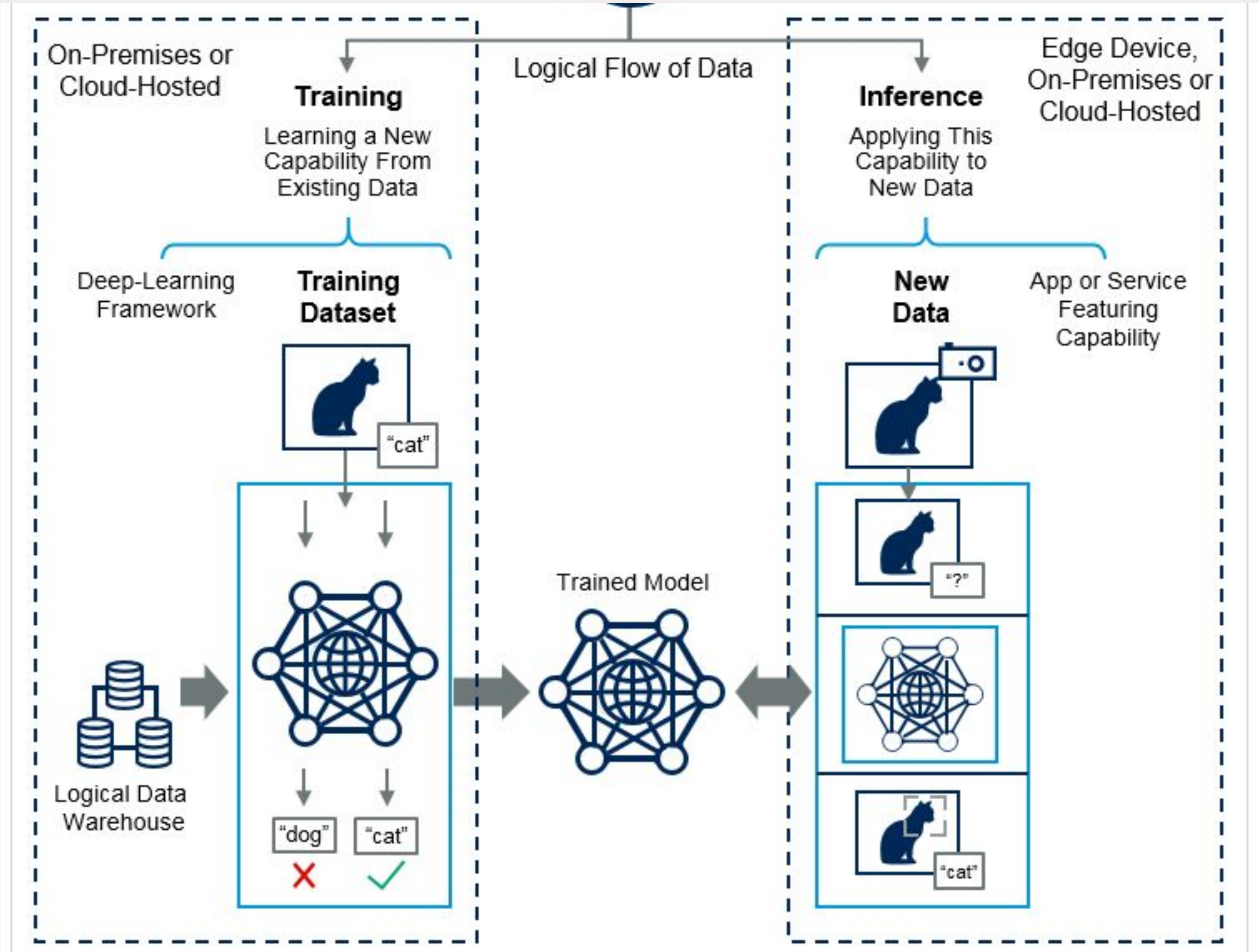
1. A brief introduction to MLOps
2. A simple yet dangerous journey through a churn use case

The usual bottleneck? deployment & operations

Why so many AI projects fail?

Some typical ML problems

- Model **decay** & no retraining of the model
- **Data** availability
- **Wrong** initial **assumptions** (problem definition)
- **Changing** business objectives
- Data **quality**
- **Locality** of the data (distributional shift)



Modellers don't think about it

Situation

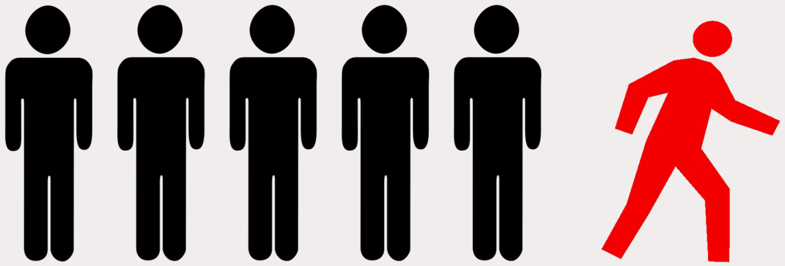
A churn model

- Work in an e-commerce company
- Data scientist
- Know all about
    - How to create a model
    - Optimize parameters
    - Evaluate model’s performance
- Tasked to
    - Build a churn model
    - Prevent customers from churning

Understanding of the problem

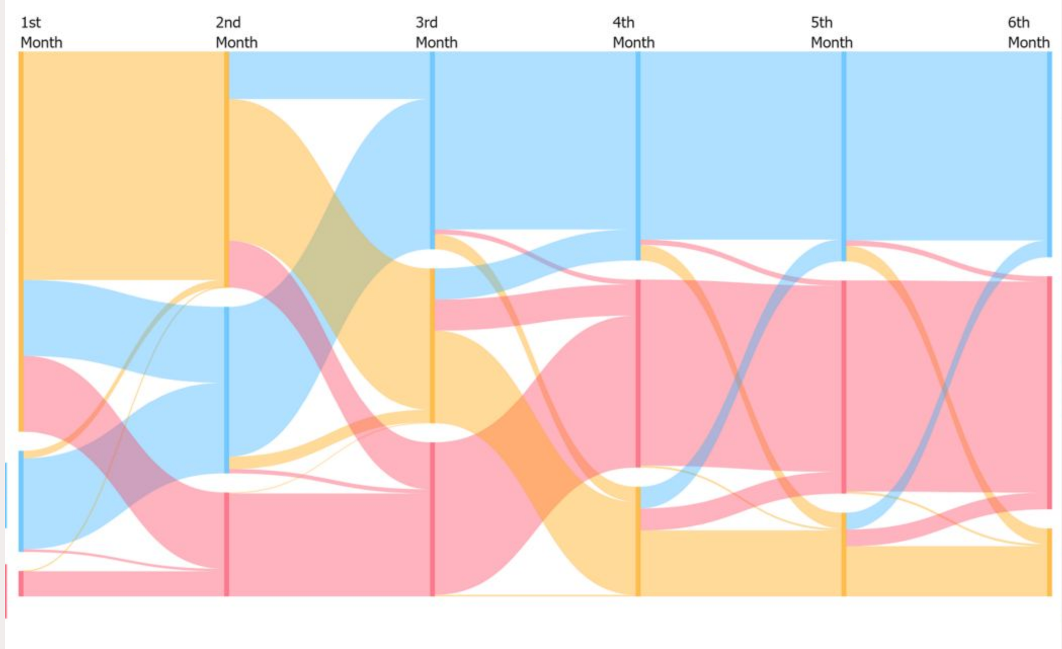
Customers acquisition costs more than customer retention

- Customers are not using the product
- Customers are leaving and unsubscribing
- +- 10% of customers churn every month
- This is a huge loss of customers

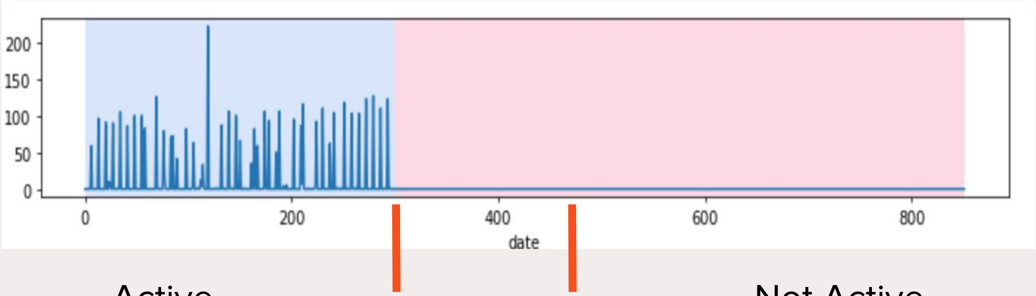


Data collected

Define churn - Analyse change in behavior



- Collected data about platform usage
- Determine when a customer is no longer active
- Check change in status over time
- Using time series regularity
- Some customers have very regular patterns of usage
- Other customers have less regular patterns
- When no interactions after expected next pattern
- Customer has churned



Active

Churn

Cancel Subscription

Not Active

Solution proposed

Predict churn

- When the customer stops his subscription, it is too late
- When the customer stops using the product, it is too late
- Predict change in behavior at least 3 months in advance
- In order to target the customer before the churn moment
- Use customer past behavior
- Use customer profile
- Use customer usage type
- Extract information from the time series data
- Use ensemble of trees

Target
= a customer will churn in the next 3 months
= will become inactive

Classification problem

The engineer manages to make a churn model

😎

👏👏👏

The engineer manages to make a churn model

How can we use your model?

The engineer manages to make a churn model

😎
👏👏👏

How can we use your model?

👨‍💼

Which models did you try? Which one are we using?

👩‍🔬

The engineer manages to make a churn model

😎

👏👏👏

How can we use your model?

👨‍💼

How can we scale and manage it?

👩‍🔧

Which models did you try? Which one are we using?

👩‍🔬

datarøøts

The engineer manages to make a churn model

How can we use your model?

Which models did you try? Which one are we using?

How can we scale and manage it?

Which data did you use? Why is it recommending me this?

dataroots

value from innovation comes from the ability to mass produce and distribute your ideas

No value

The lifecycle of a project

Ideas and R&D
Creating choices, quantity over quality, novelty

No value

The lifecycle of a project

Ideas and R&D
Creating choices, quantity over quality, novelty

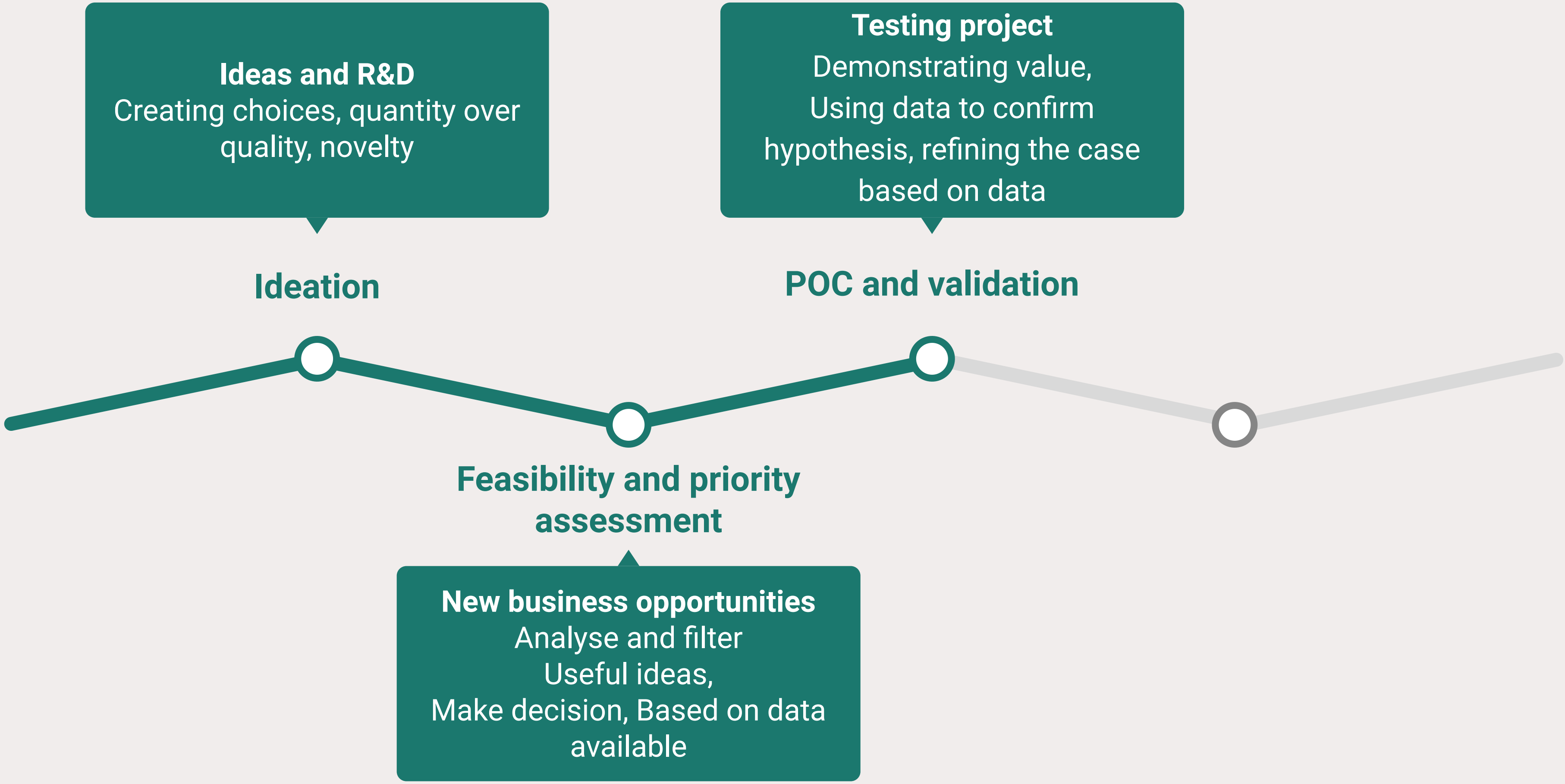
Ideation

Feasibility and priority assessment

New business opportunities
Analyse and filter
Useful ideas,
Make decision, Based on data available

The lifecycle of a project

No value



The lifecycle of a project

Value

Ideas and R&D
Creating choices, quantity over quality, novelty

Ideation

Feasibility and priority assessment

New business opportunities
Analyse and filter
Useful ideas,
Make decision, Based on data available

Testing project
Demonstrating value,
Using data to confirm hypothesis, refining the case based on data

POC and validation

Scaling

Fully leverage value
Deployment, automation, monitoring, alerting, scheduling, MlOps

Operational challenges

⚙️ Deploy model systems (not just one off solutions)

🏋️ Models monitoring and (re)trainining

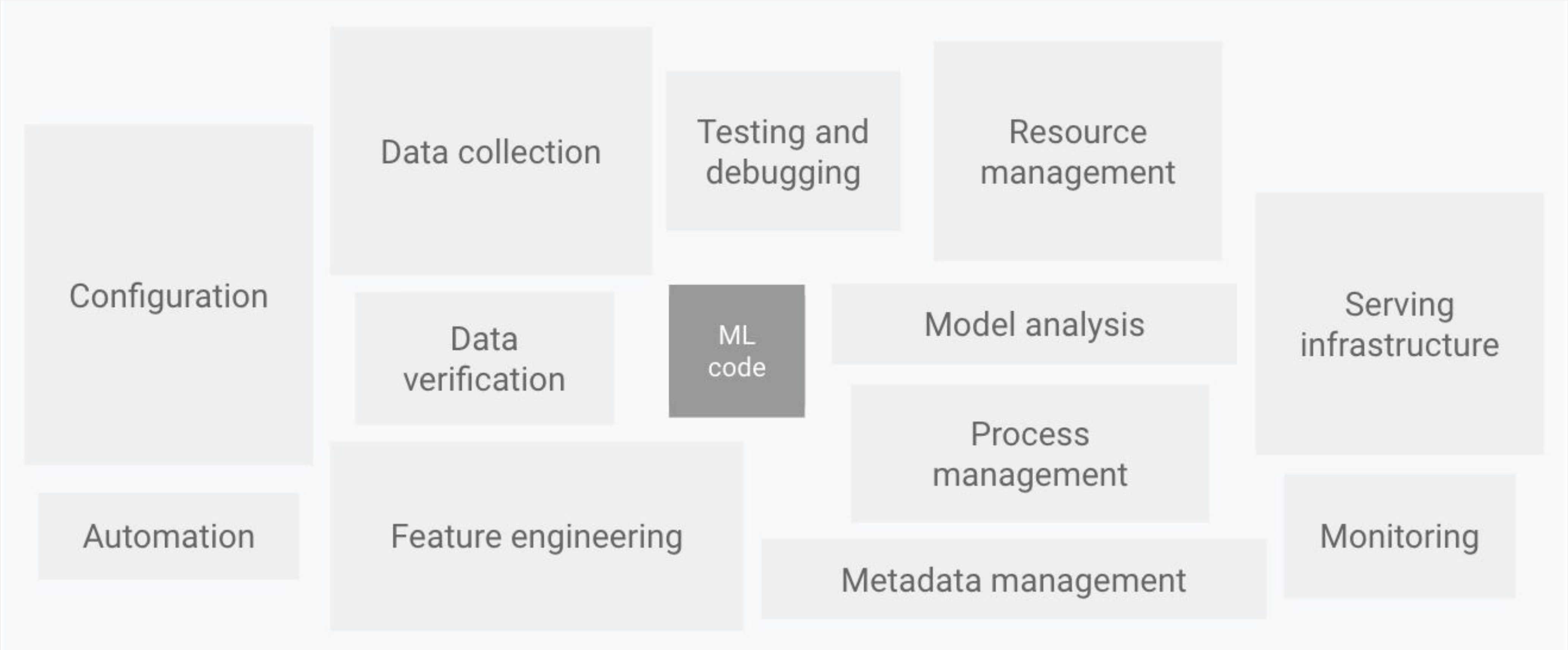
🏋️Models and experiments are not properly **tracked**

📦 Consistent project structure

📦 Code and dependencies tracking

💡 Auditability and regulations - reproducibility and explainability

Typical ML systems problems



Who does need to adopt ML engineering best practices?

A. Data Scientist
B. Data Engineers
C. Business Owners
D. All the above

Machine learning challenges

- 📉 Model decay
- ♾️ Retraining
- 💼 Wrong initial assumptions (problem definition)
- 💼 Changing business objectives
- 🔎 Data availability
- 🔎 Data quality
- 🔎 Locality of the data (distributional shift)

Any ideas how to help?

Discover MLOps

MlOps is the extension of the DevOps methodology to include Machine Learning and Data Science assets as first-class citizens within the DevOps ecology

https://www.tekna.no/en/events/mlops---what-you-need-to-know-43062/

dataroots

What do you know about DevOps?

What is DevOps?

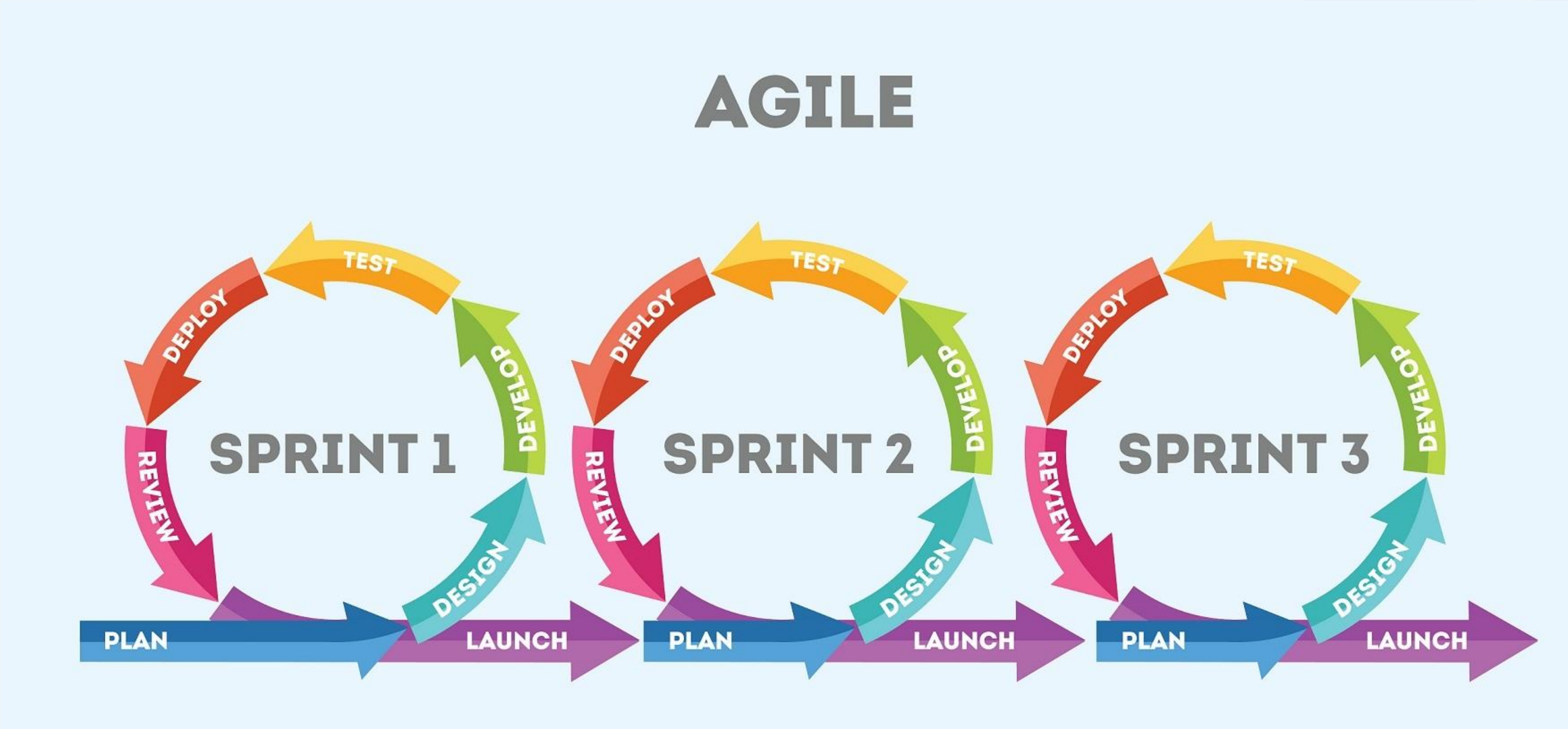
*"A set of **practices** intended to reduce the time between committing a **change** to a system and the change being placed into normal production, while ensuring **high quality**"*

DevOps usually involves using Agile methodologies.

The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win

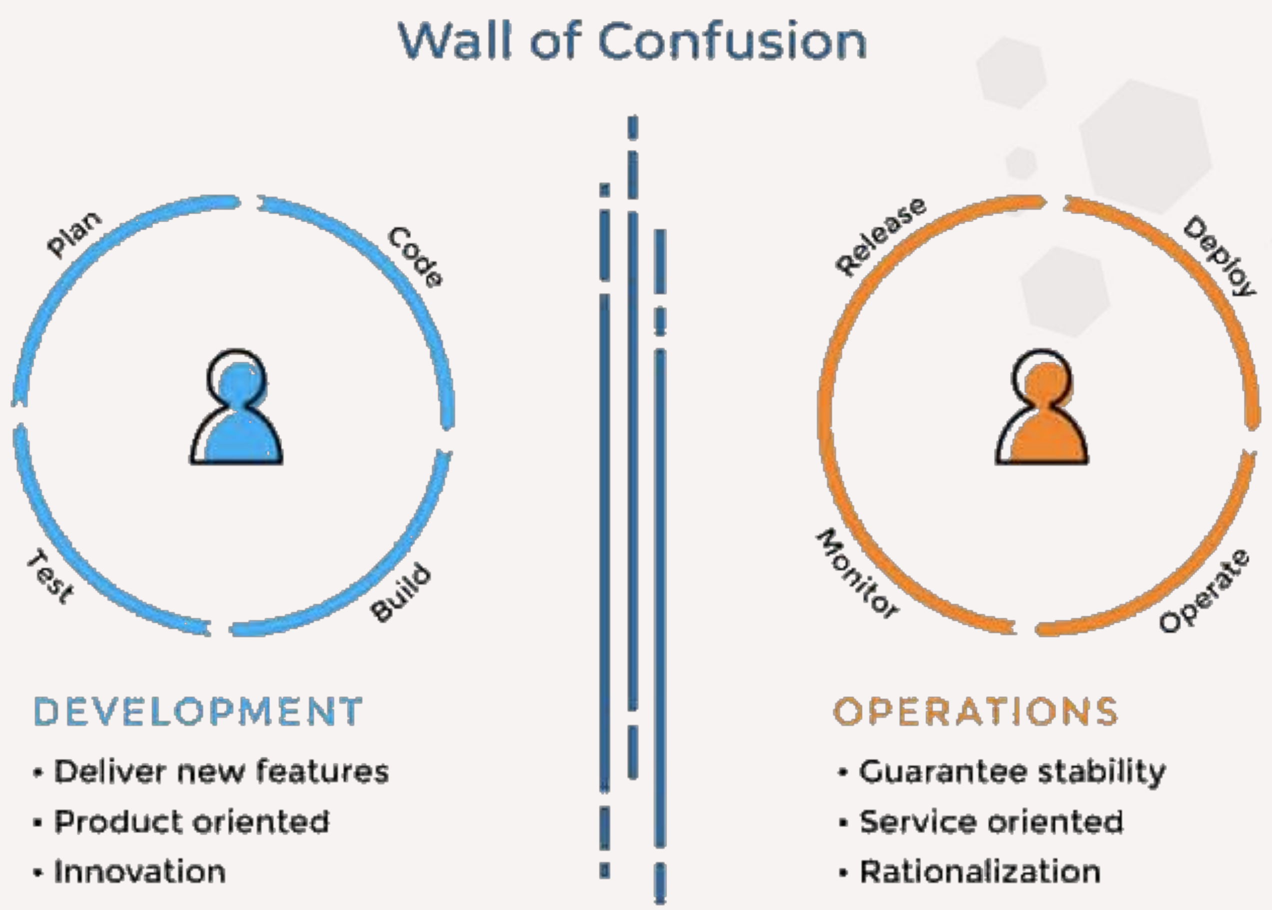
dataroots

What is Agile?



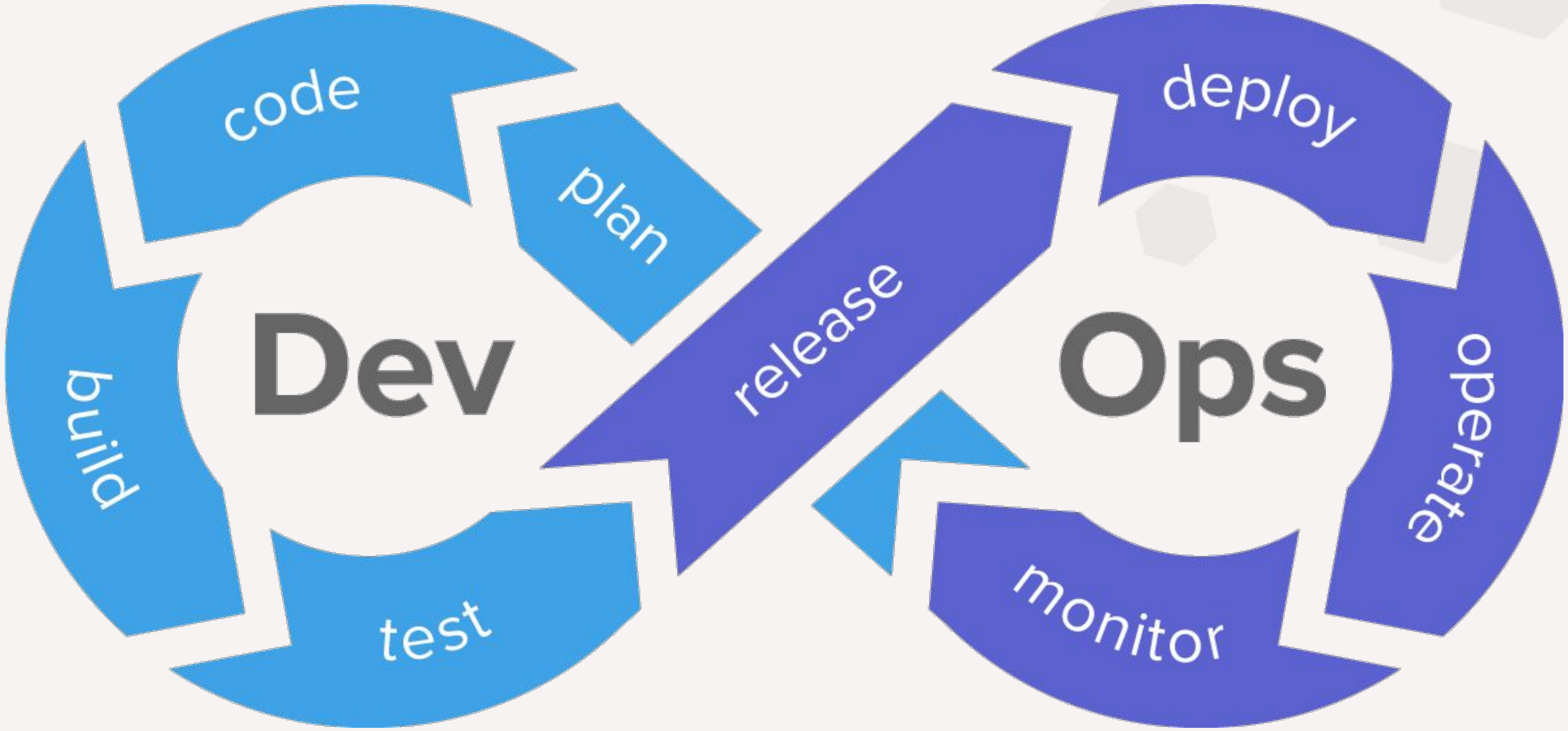
https://agilemanifesto.org/

What is DevOps?



dataroots

DevOps engineer



dataroots

What is DevOps?

Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

https://services.google.com/fh/files/misc/state-of-devops-2021.pdf

DevOps + ML = MLOps?

MLOps purpose?

- 🚀 Unify the release cycle
- 🔨 Automated testing (e.g. data validation, model testing, ...)
- ♾️ Apply agile principles in ML
- ⛓️ Include ML as first-class citizens within CI/CD systems
- 💸 Reduce technical debt

MLOps must be a language-, framework-, platform-, infrastructure- and people practice.

~~Dev~~MLOps

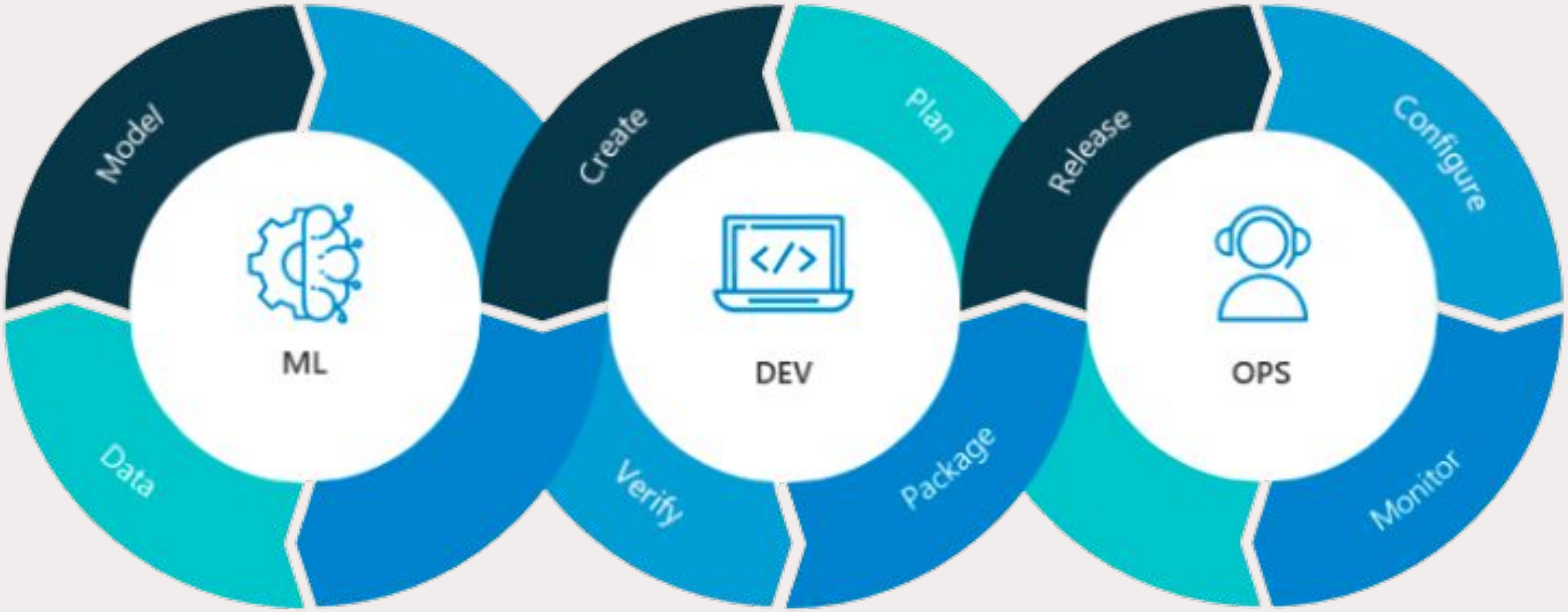


Image from NVIDIA

MlOps 6 Principles

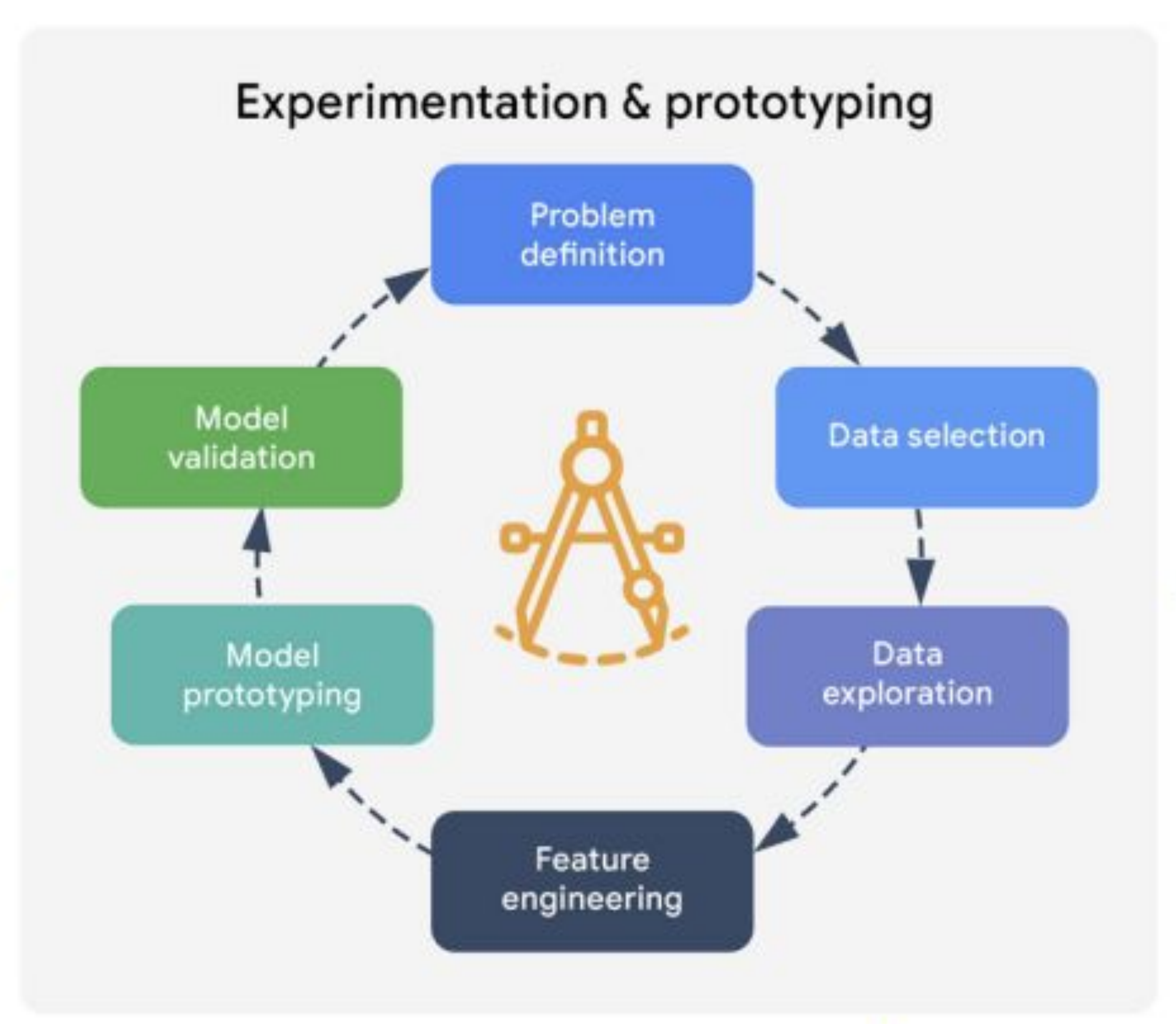
- **Reproducibility**
- Versioning
- **Testing**
- **Deployment**
- Monitoring
- **Automation**

Is it not exactly the same as Devops?
How does it differ?

- Model
- Features
- Data

Let's go into details for each of these principles

ML development process



Reproducibility - collecting data

- Always **backup** your data.
- Saving a **snapshot** of the data set
- Data sources should be designed with **timestamps** so that a view of the data at any point can be retrieved.
- Data **versioning**.

ID	Snapshot time
1	01/01/2022
2	01/01/2022
1	01/02/2022
2	01/02/2022

- Manage evolution of data (also detect drift)
- Avoid data leakage
- Reproduce even in the past

Reproducibility - Features engineering

- Feature generation code should be under **version** control.
- Often features generation follows the same mechanism as a model

-> **save the complex features like a model / in model step**

Train

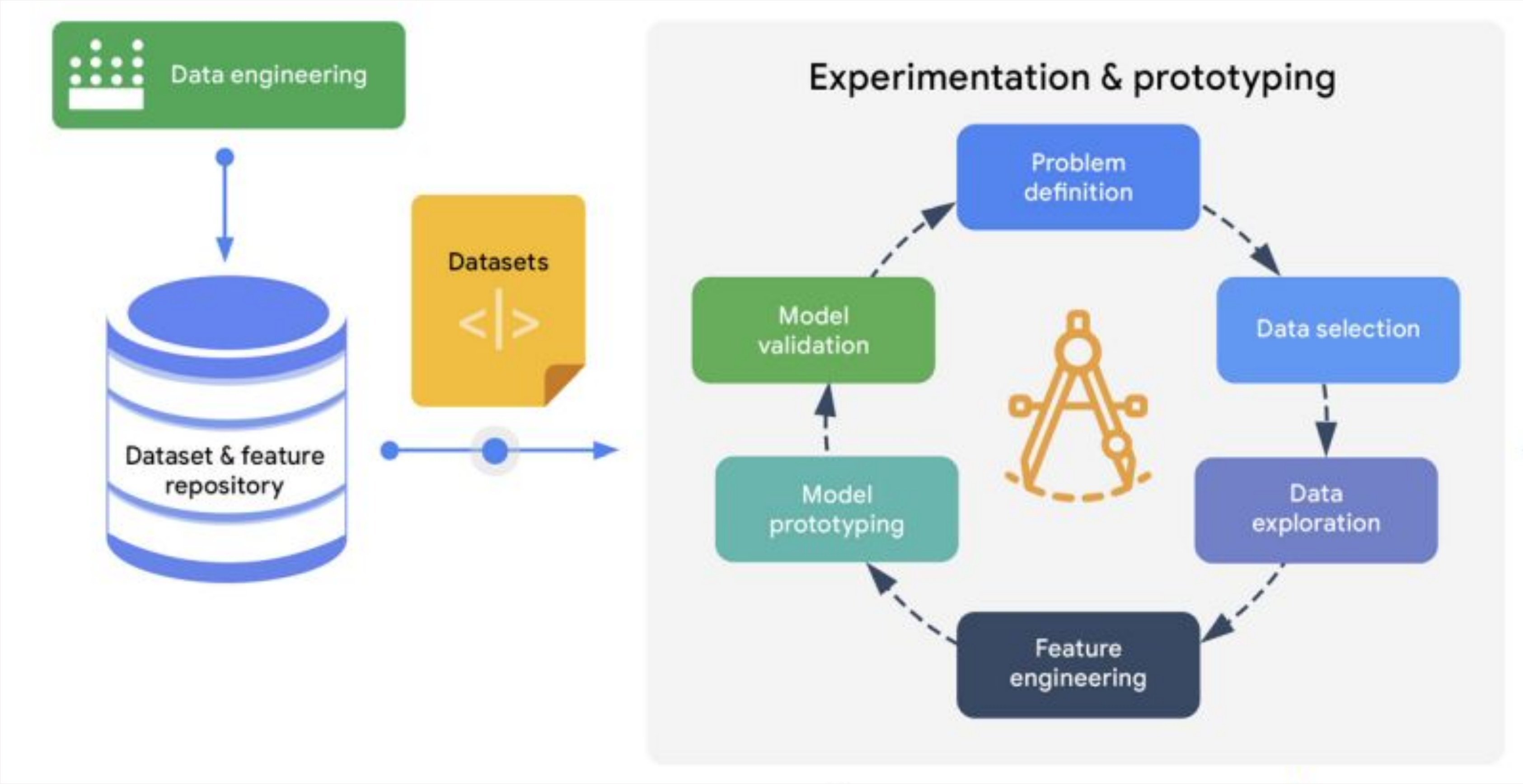
Feature	Imputed feature (median)
1	1
null	1
1	1
2	2
null	1
1	1

Test

Feature	Imputed feature (median)
null	2
null	2
1	1
2	2
2	2
null	2

Data drift caused by not saving the median value

ML development process



Reproducibility - model (re)training

- Ensure the **order** of features is always the same
- **Document** and automate hyperparameters
- Document and automate the architecture of ML models.
- **Track** experiments
- Set seeds and save all parameters

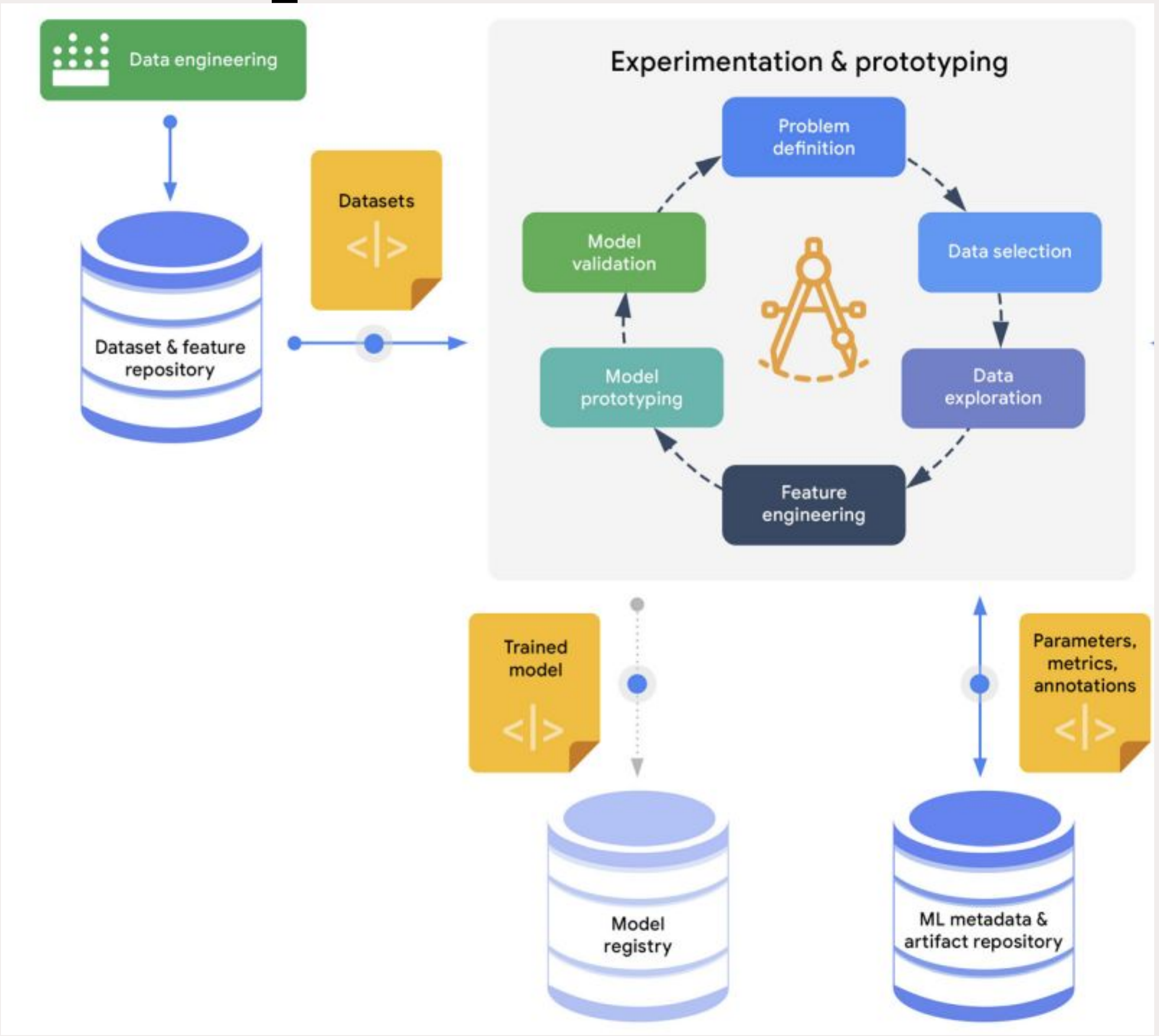
Nice cake, can I have the recipe?

Euh, did a lot of experiments, did not write down the proportions…

TensorBoard

mlflow

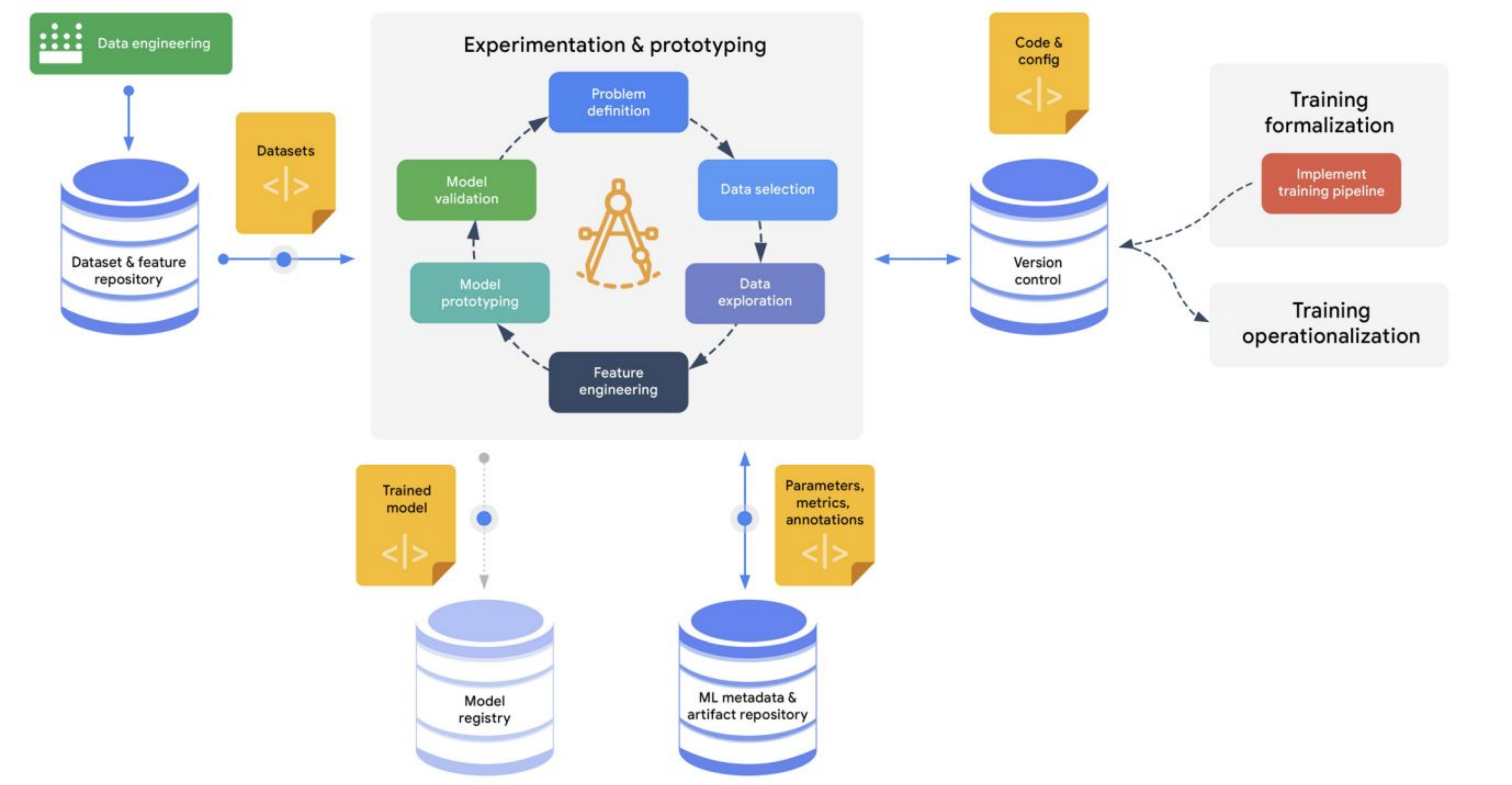
ML development process



Reproducibility - Versioning & deployment

- Software **versions** and dependencies should match the production environment.
- Use a **container** (Docker) and document its specification, such as image version.
- Register the model in a **registry**
- Ideally, the **same programming language** is used for training and deployment.

ML development process



Testing,
Automation

Quality assurance

Code quality includes

- **Linting**
- **Unit tests**
- **Data pipeline tests**
- **ML model pipeline tests**
- **Application pipeline tests**

Black

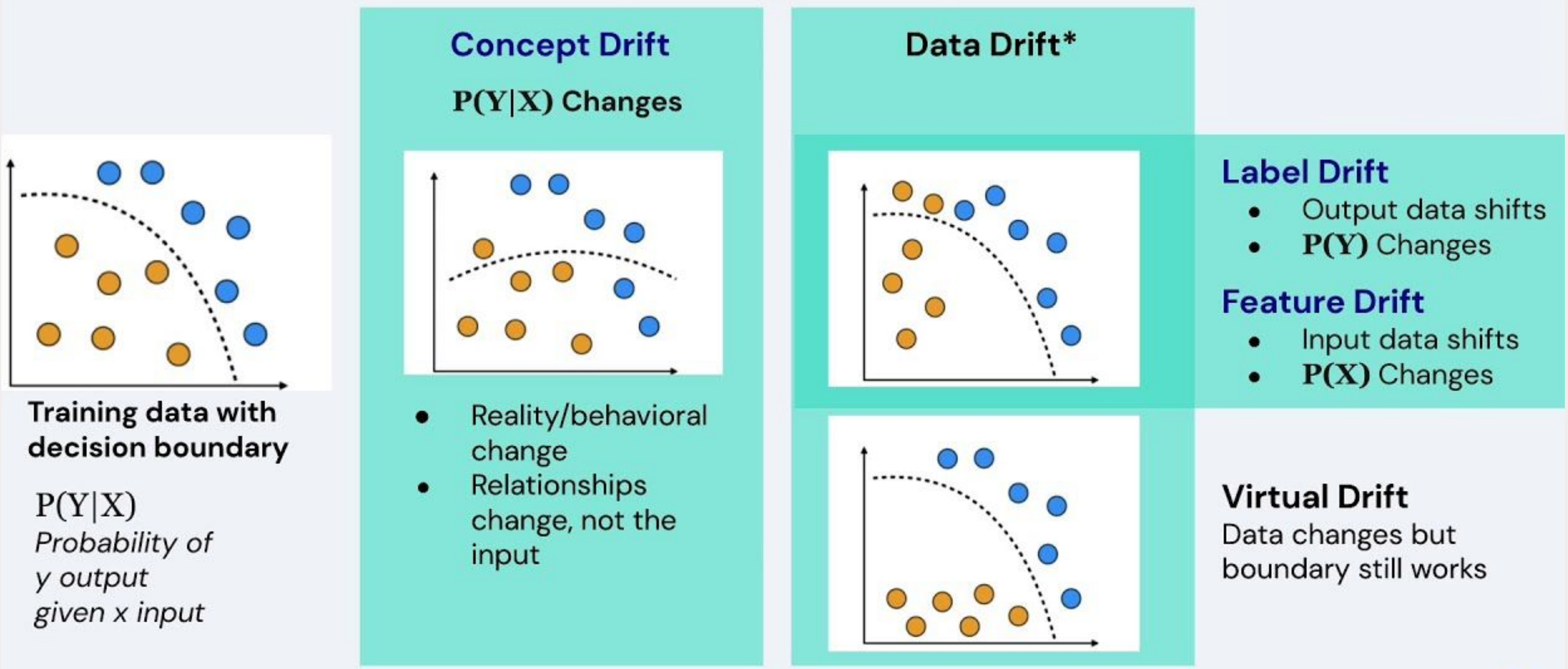
TextAttack

pytest

CI/CD

great_expectations

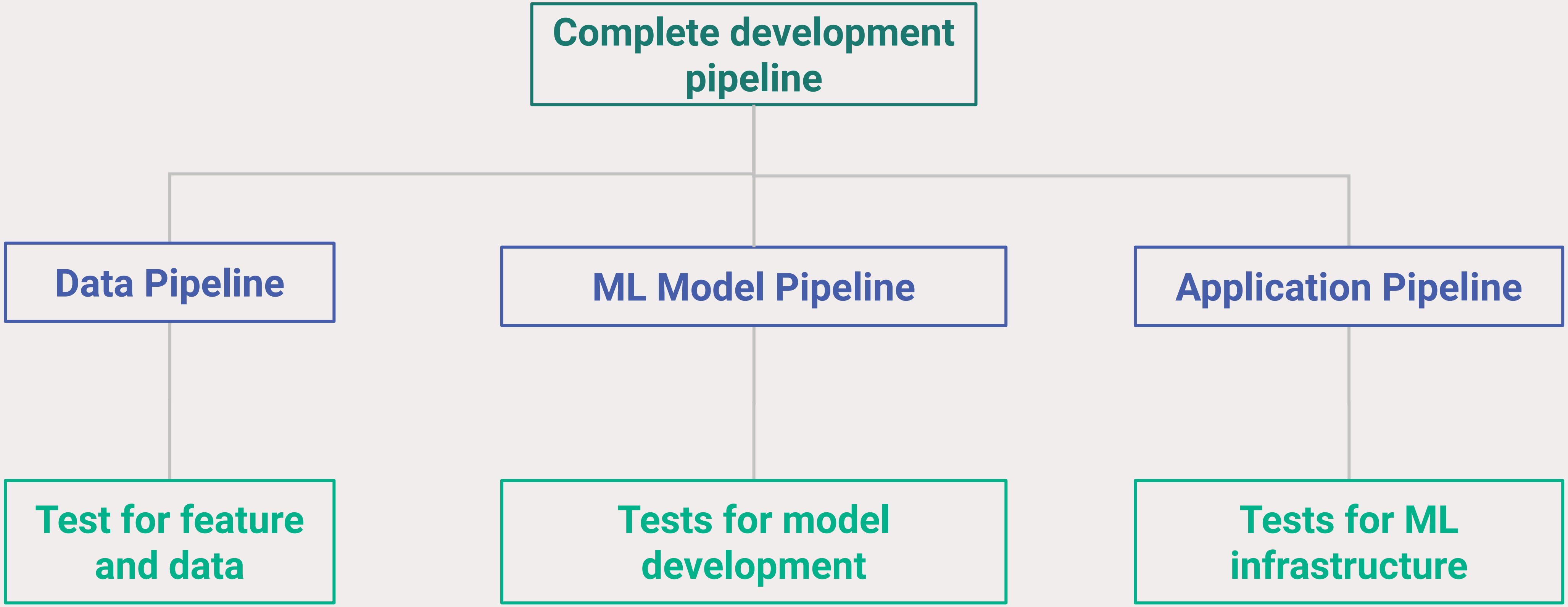
Most common drift = Real data diverging from baseline data



Understanding Data drift = **Real data diverging from baseline data**

- **Concept drift** or change in P(Y|X) is a shift in the actual relationship between the model inputs and the output.
  - The behavior of buyers wrt a product
- **Label drift** or change in P(Y Ground Truth) is a shift in the model's output or label distribution
  - The price of houses in the market, given the same features (inflation)
- **Feature drift** or change in P(X) is a shift in the model's input data distribution.
  - The lightning of a picture of a product (ex e-commerce)

Testing

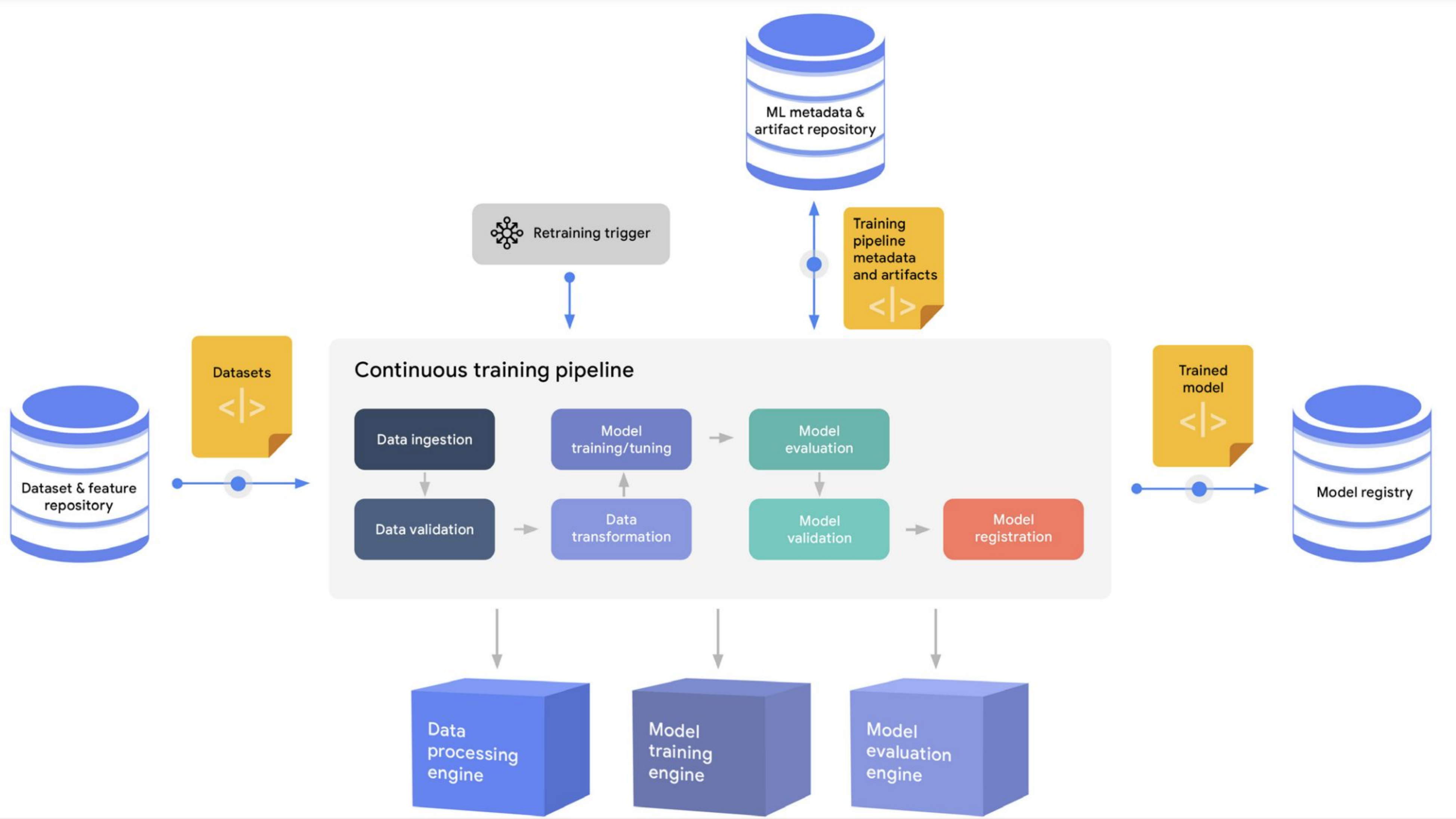

Complete development pipeline
Data Pipeline
ML Model Pipeline
Application Pipeline
Test for feature and data
Tests for model development
Tests for ML infrastructure

Tooling enablers : CICD

- CI = Continuous Integration
- CD
  - Continuous Delivery (phase 1)
  - Continuous Deployment (Phase 2)

A pipeline is a set of stages, containing steps. Each step will run specific checks or actions to ensure that the change that was introduced is able to go to production, and subsequently prepare it to be deployed

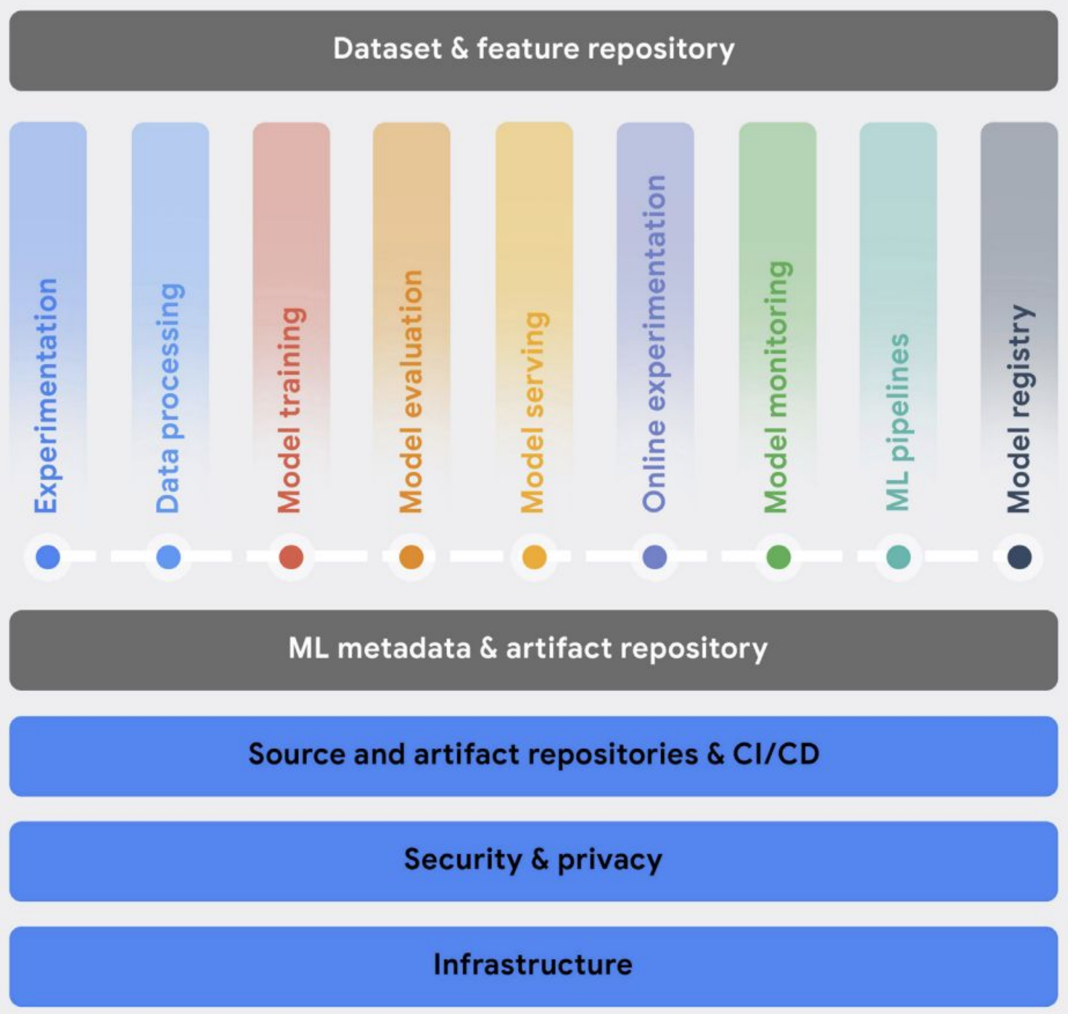
Continuous training



MLOps Infrastructure Stack

The MLOps technology stack should include tooling for the following:

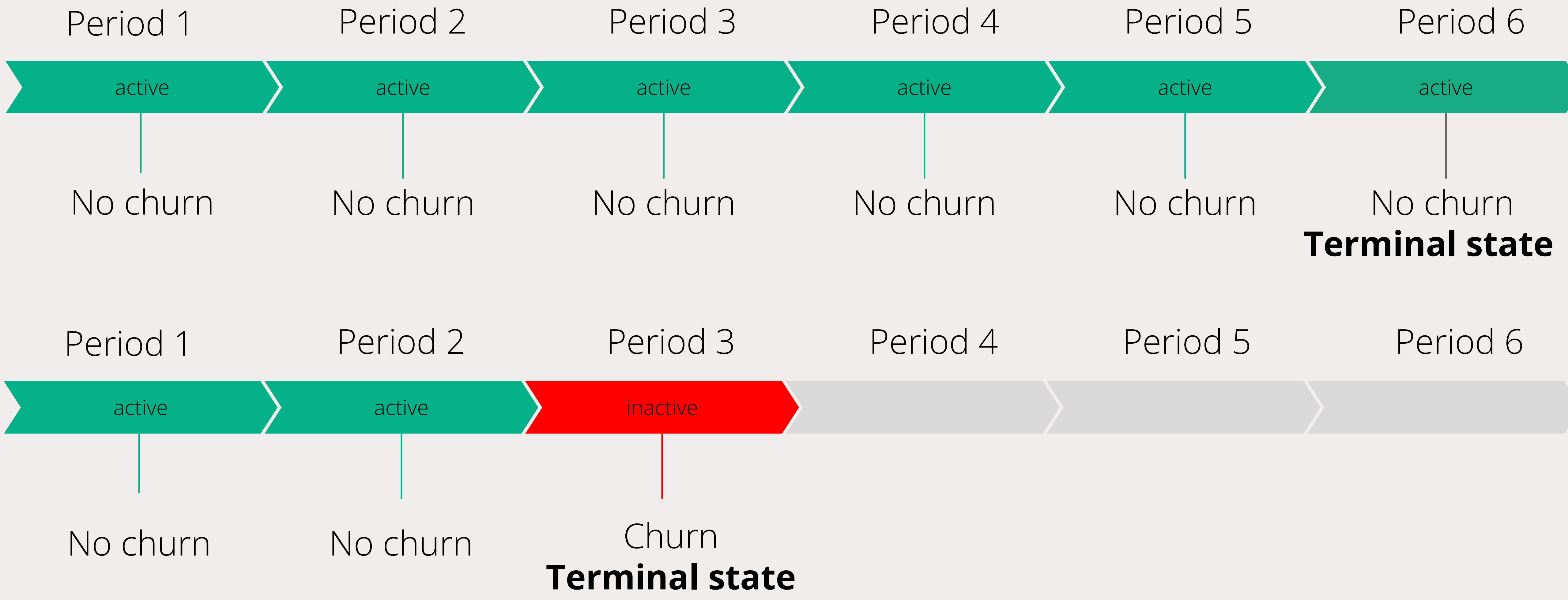
- data processing,
- version control of data, models & code,
- CI/CD ML pipelines,
- automate deployments & experiments,
- model performance assessment, and
- model monitoring



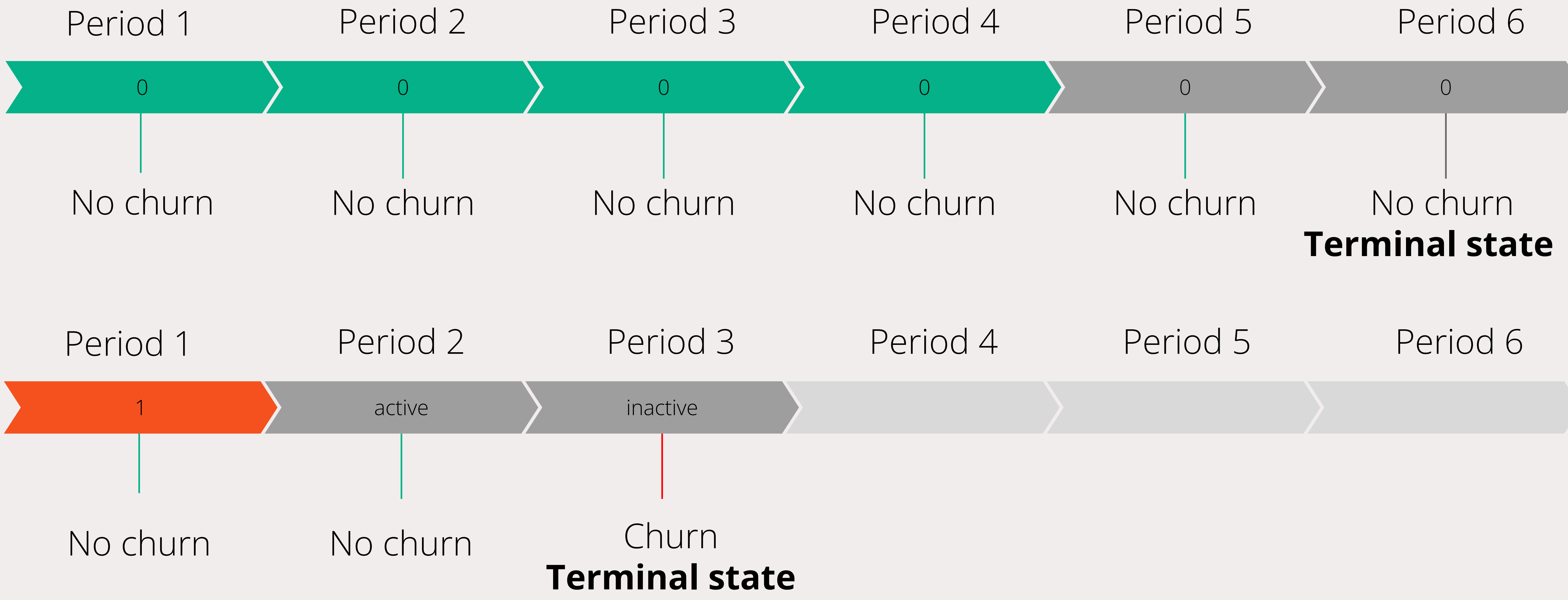
Avoid pitfalls and start thinking since the development about MLOps

Marketing

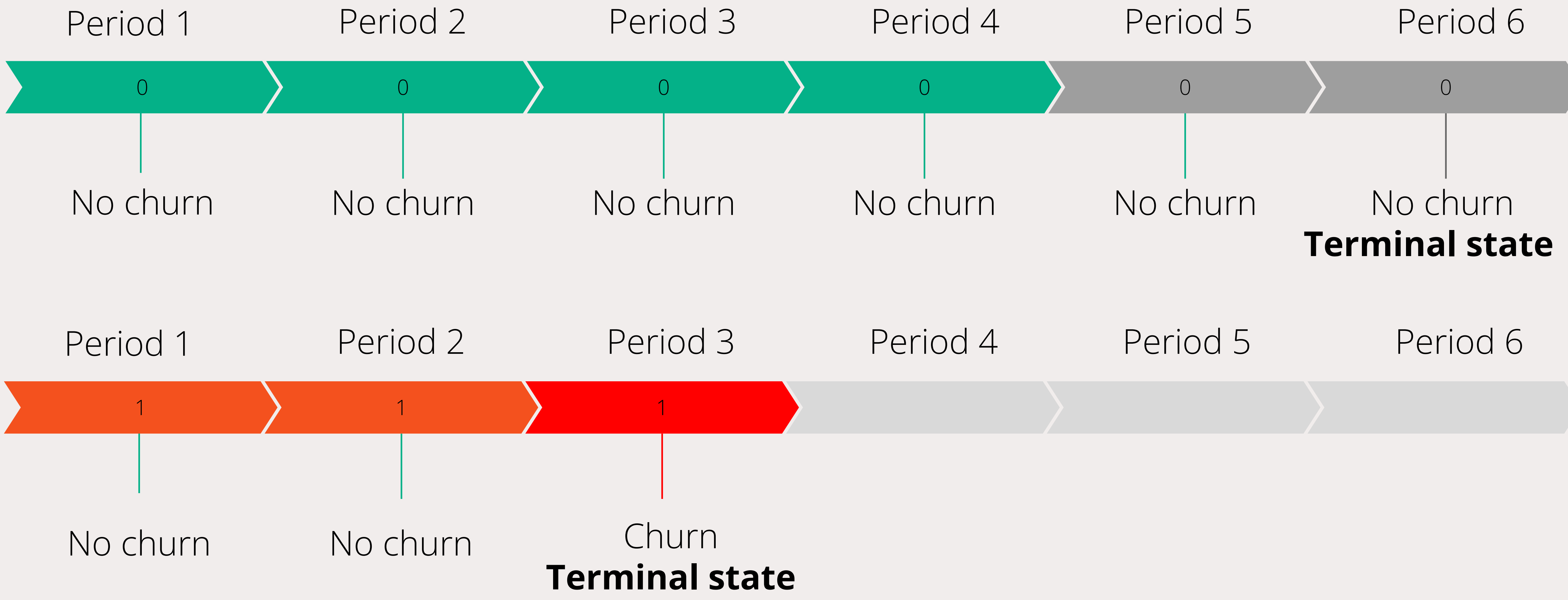
Target creation the first trade off



Target creation the first trade off - option 1 - 3 months before



Target creation the first trade off - Option 2 - adding last 3 months as 1



Designing the target - Trade offs

Will churn in 3 months

+ the question is more clear cut
+ Gives more margin to the campaign team to act
- the signal could be happening 2 months before the churning moment
- Less target points for class 1 (unbalanced dataset)

Will churn within the next 3 months

- the question is more vague
- The signal will be increasing towards the churn moment
- It can be difficult to have a clear cut of the features correlation with the target
- gives less margin to Marketing to act
+ You have more data points for the class 1 (unbalanced dataset)

Designing the target - Trade offs

Will churn in 3 months

+ the question is more clear cut
+ Gives more margin to the campaign team to act
- the signal could be happening 2 months before the churning moment
- Less target points for class 1 (unbalanced dataset)

Whatever you choose, always keep it in mind, it's very important for the productionalisation of the project

Will churn within the next 3 months

- the question is more vague
- The signal will be increasing towards the churn moment
- It can be difficult to have a clear cut of the features correlation with the target
- gives less margin to Marketing to act
+ You have more data points for the class 1 (unbalanced dataset)

The productionalisation manner influences the target design choice (streaming - batch)

Don’t make the mistake to deploy as an API something running as batch

Data is not processed in the same way

Streaming predictions

Is the customer going to churn in the next 3 months, starting from any day in the month, any time of the day

The historical dataset needs to be created in this fashion - otherwise, if you construct your dataset monthly but query it daily - it doesn’t make sense

The data is anyway not more granular than monthly -> your predictions will look the same for the same month

The data is daily but the model has learned patterns on monthly features created on the whole month and not in between months -> performance cannot be guaranteed

Batch predictions

Monthly Dataset - constructed on the historical monthly data

Also fine to use an API for batch prediction, as long as you inform the data consumers of the way you intended your API predictions

Train test - validation set & MLOps

Reproducing results for future runs

Training model and Fine Tuning parameters

Independent validation

Training set

Validation set

Test set

Out of Time, Out of Sample

Out of Time, Out of Sample

Train test - validation set & MLOps

Reproducing results for future runs

Training model and Fine Tuning parameters

Independent validation

Training set	Validation set	Test set
	Out of Time, Out of Sample	**Out of Time, Out of Sample**

Out of sample

- First hashing all the users
- Then selecting the sets based on the hash:

in training, all users with hash < 0.70
in validation, all users with 0.7<=hash < 0.85
in test, all users with hash >= 0.85

Out of time

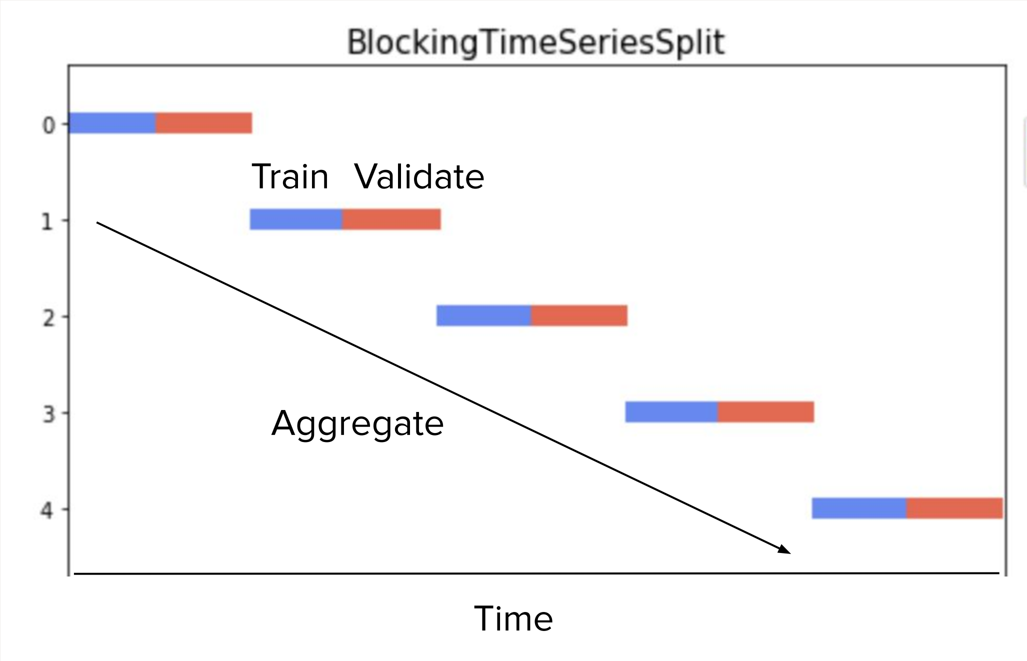
- split based on time for training

January -> model 1
February -> model 2

How can I know which model to deploy?
-> compare the models
On which data?
-> unseen data
By which model?
-> both models
how can I make sure that the users that I have in my test sets are the same?

Sampling methods & society's evolution

Behavior of men is something changing



Challenges:

- Few churners in sample set (3-10%)
- Behavior of churners evolves over time

Oversampling - SMOTE

- ok if you have enough variability between the churners
- Careful not to extrapolate too much
- Categorical vs continuous variables - creating weird effects

Undersampling - remove the active people

Leakage if all timeXuser are in the same training set

-> training independent models allow more robust results

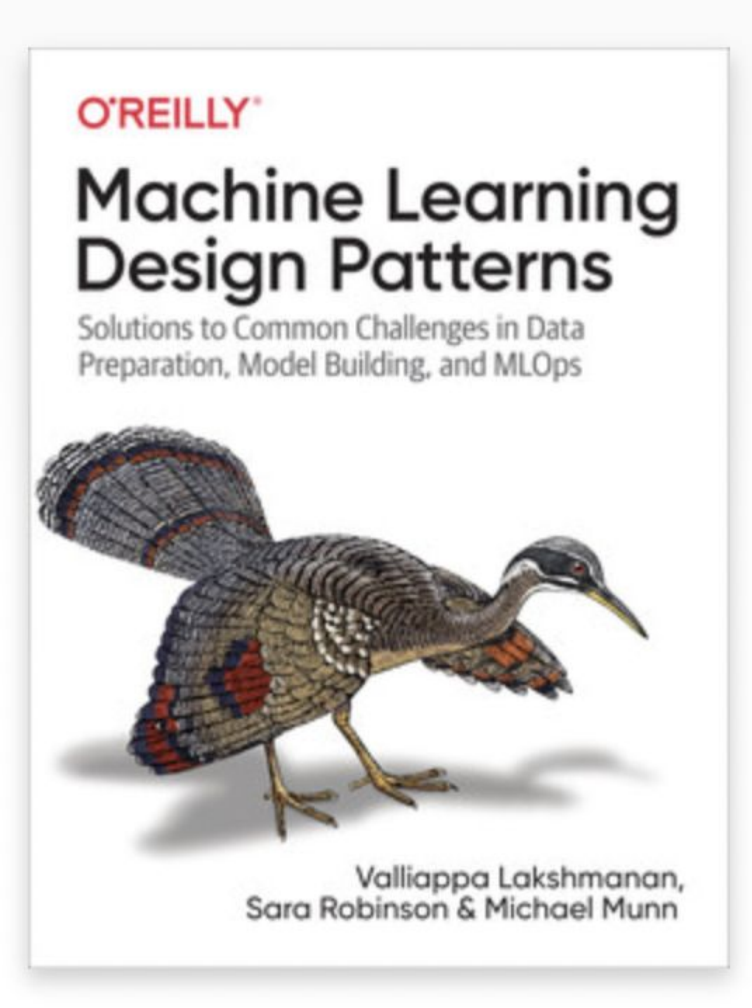
weight over time

Always keep the test set intact

Feature engineering - Thinking about the outcome

Imputing for robust pipelines

- What happens when imputing missing values, removing outliers and normalizing features
- Thinking about non existing values - how will you treat in production?
- Use the right encoding method for the right values - Zip code vs favorite color
- Normalizing categorical vs continuous variable
- Time series processing is not so straightforward
- **Be smart - most gains lie in this step**



Modelling

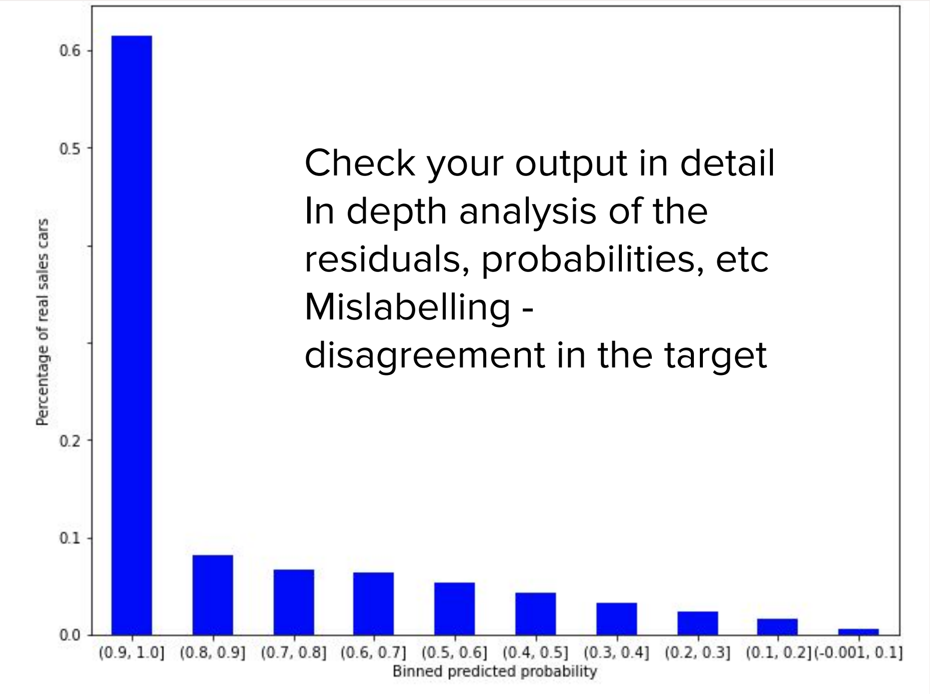
Machine learning

Key questions

- What **type** of problem is it?
- Is this a **balanced** problem (i.e. the outcomes occur with similar frequency)?

Steps

- Optimize the right **metric** & determine maximum attainable
- Determine the **baseline** of a random/dummy model or a business **heuristic**
- Is ML even needed?
- Develop model on the **train** and **validation**
- **Check performance** on the test set
- **Calibrate** your model



THE MODEL - frugal development

Experimentation is not carbon neutral

The average American generates

16.4 tons of CO_2e emissions in a year

The adult human brain runs continuously,

whether awake or sleeping,

on only about 12 watts (0,0000012 MWh)

Model	Energy consumption, MWh	CO2e emissions, tons
Evolved Transformer	7.5	3.2
T5	85.7	46.7
Meena	232	96.4
Gshard 600B	24.1	4.8
Switch Transformer	179	72.2
GPT-3	1,287	552.1
PaLM	3,181	271

- CPU = energy to power the calculations
- Storage = dematerialized data centers ->cooling demand
- Servers = rare metal component

The demand for **metal**, will increase by **8 times** in 2050, material used to produce AI hardware and components

Churn- how to measure the model?

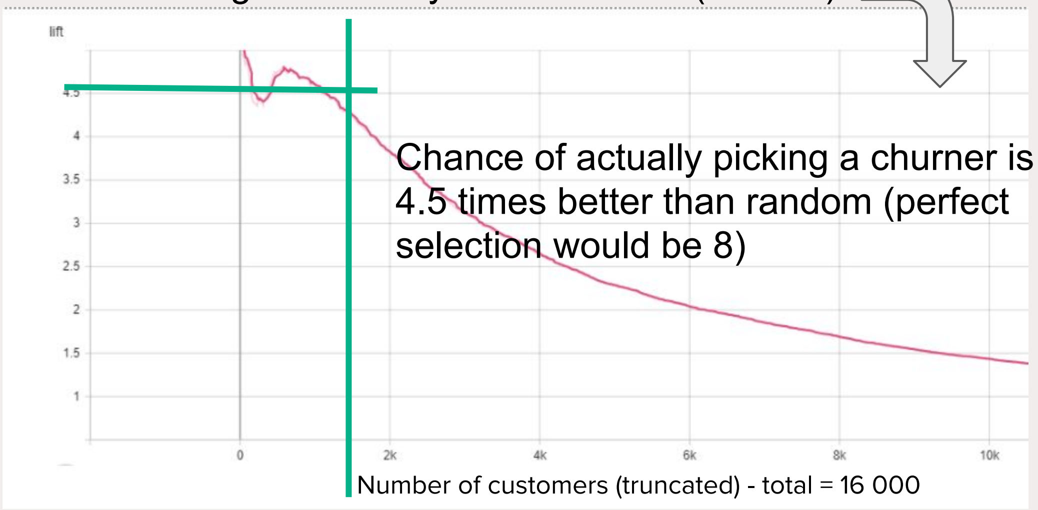
The lift curve a metric that is easy to understand

Lift = **How much better than random?**

If you were to pick randomly some customers, then you'd have a very small probability of actually targeting a future churner (= 12% in this case)

If you use the model's ranking, there is a much higher chance that you are actually targeting a churner (=4.5x 12% = **54%**)

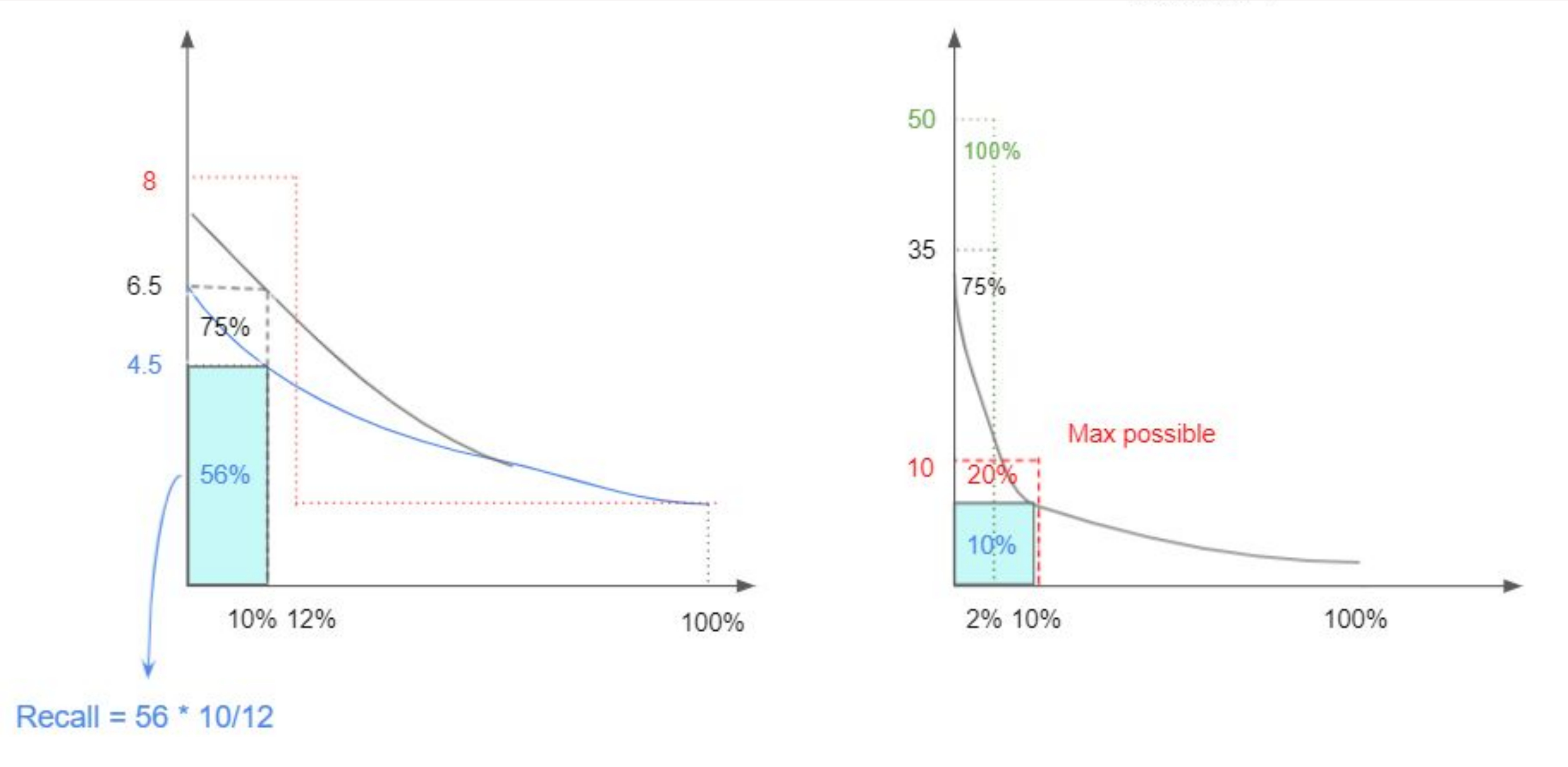
You target 1600 of your customers (+- 10%)



The lift curve tells you **how much better** than random the model picks the future churners

Don’t overshoot

What is the best model capable of achieving?



Performance vs correlation vs causality

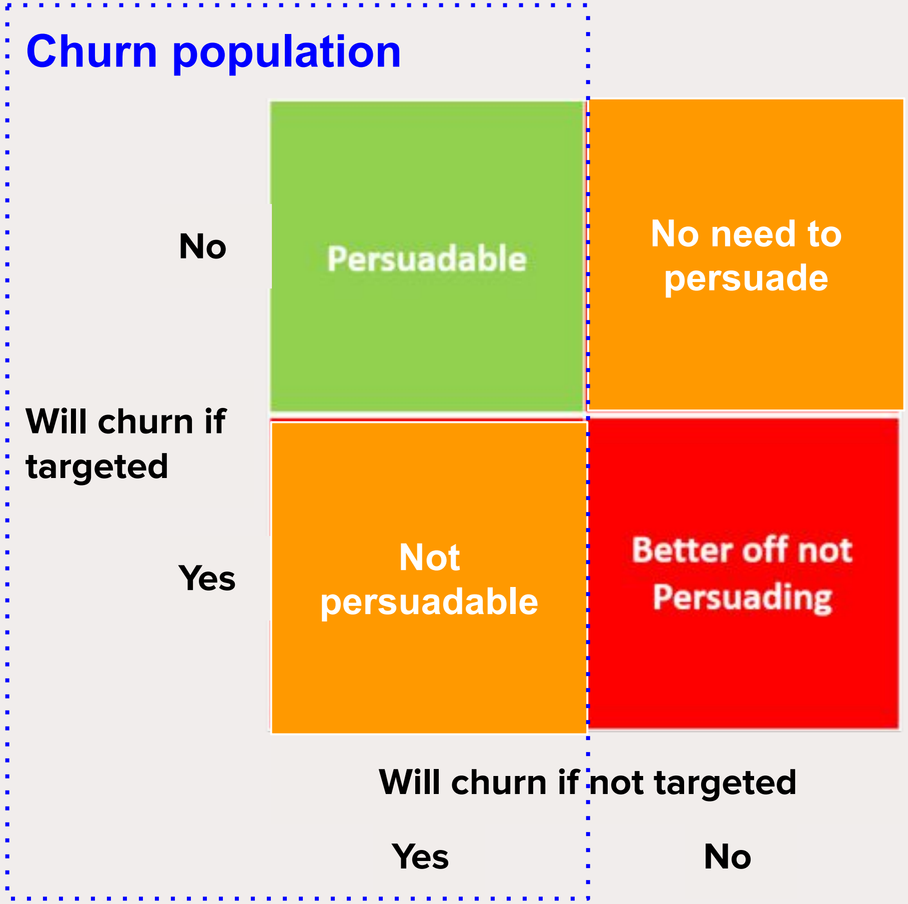
Predicting churn is not always enough to prevent it

- Global feature importance are same for everyone - no non-linearity
    - Partial dependence plots
    - Regressive feature selection
- Local feature importance - much more personal - allow for strong non linearity
    - Shape
    - Lime
- Still no causality - correlation at best
- Univariate - not multivariate
- Usually the feature names don’t mean much to the stakeholders - complex features are hard to understand, even harder to make actionable
- Causal inference is even better - needs some causal modelling - a lot of domain knowledge - hard to pull through and to automate

The customer will churn in the next 3 months, is it the right question?

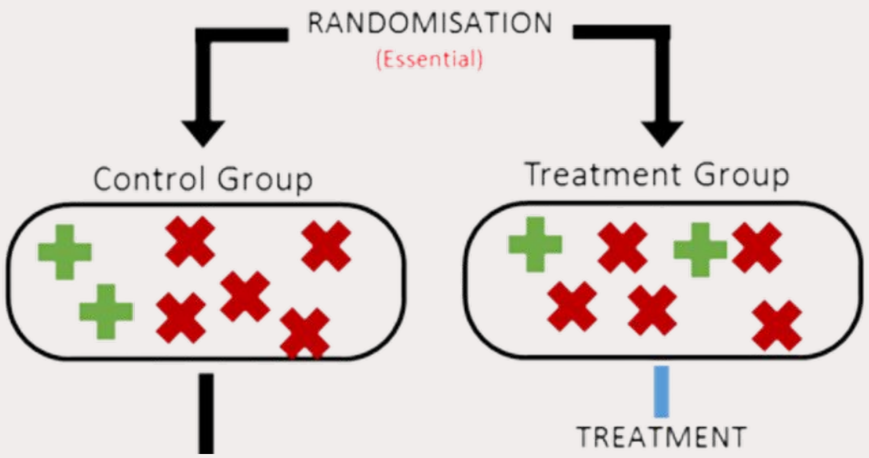
Which customers if targeted will react to a marketing campaign?

- When first model, this information is not available
- Maybe use proxies on other campaigns
- Use the high probability churners
- Collecting data for the next marketing campaigns



Experiment design

Establishing a targeting strategy - A/B testing - causal inference RCT



Different from a random trial, because we select a subpopulation with high chance of churn

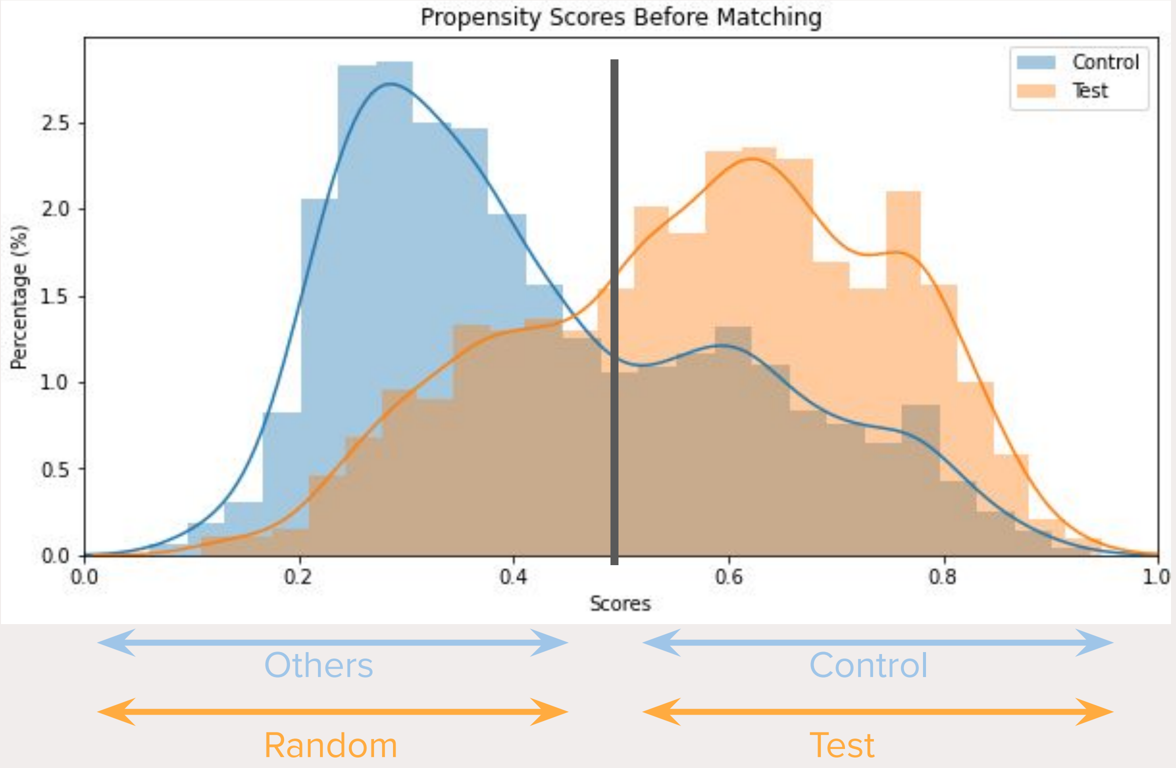
Once churn is predicted, **what can you do?**

- **Treatment** = target group, on which marketing launches an **action**
- **Control** = target group, on which no action is applied. This group serves to measure the effect of the campaign in a reliable way
- **Random** = a randomly selected group that serves to explore other possible scenarii, by being targeted by marketing
- **Others** = remainder of the population, which receives no action and should have a lower churn rate.

What if you don't have the experiment design?

Synthetic population & causal inference

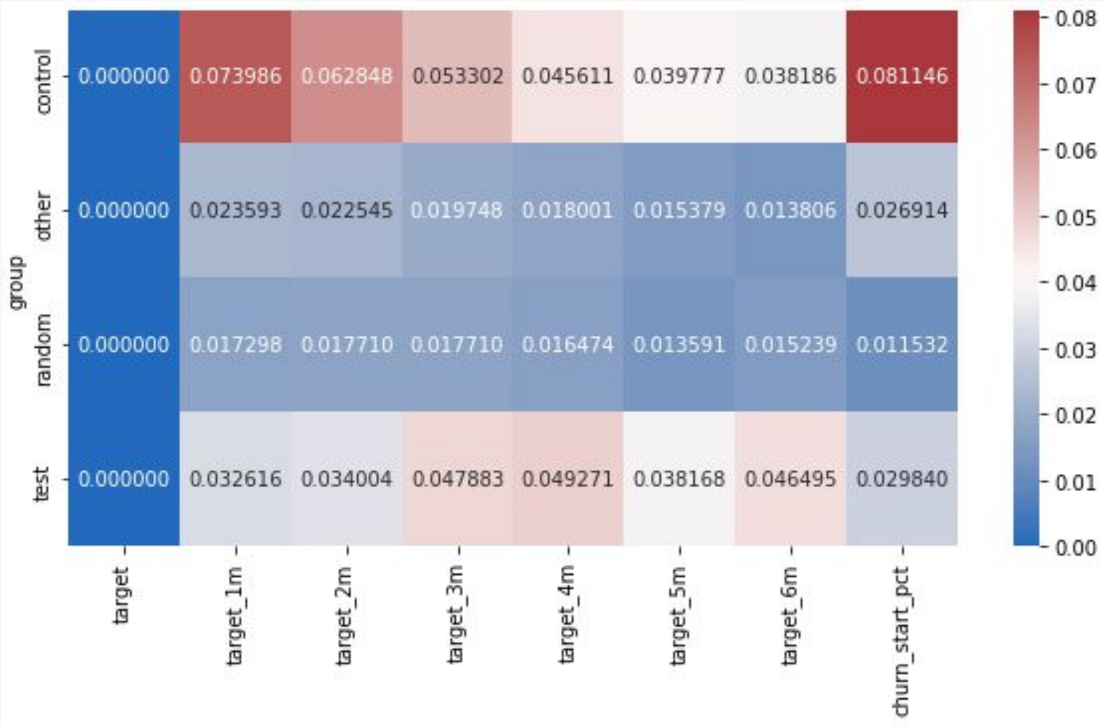
- Identify the groups targeted by the campaign by creating a logistic regression model (1 = targeted & 0 = not targeted)
- Separating the groups based on the propensity score
- Not as good as for a real experiment - not always showing great results



From results to actions

With the marketing department - elaborate campaign

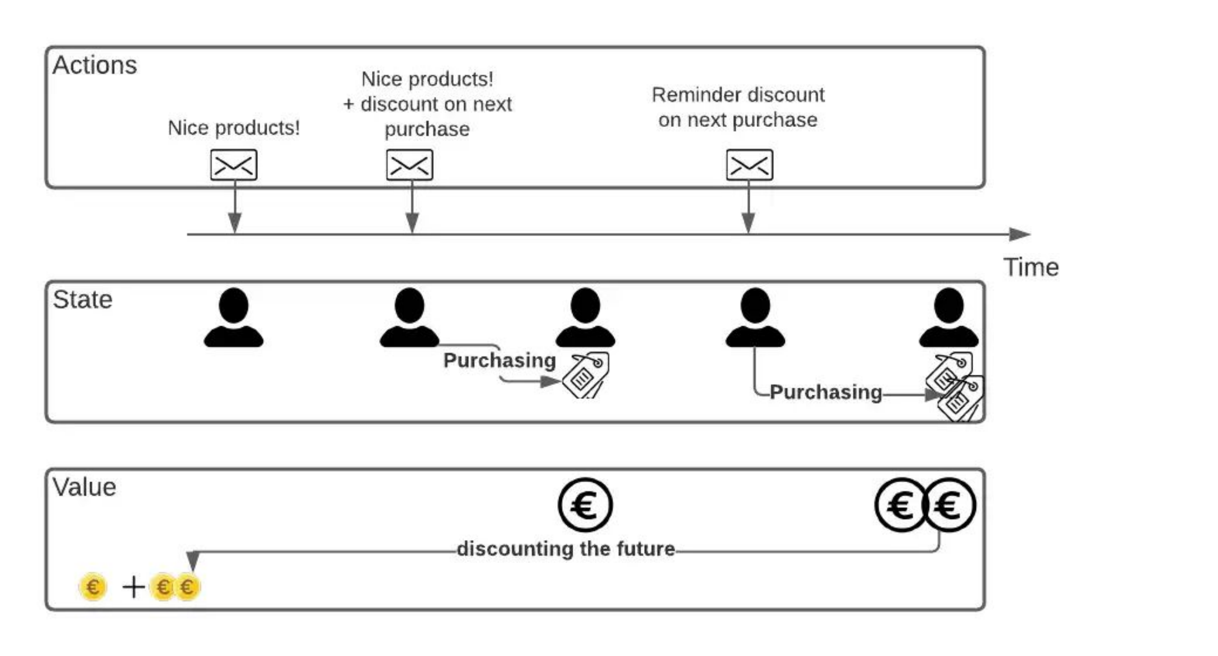
- Based on the sub segment of the customer
- **Personalised mail**
- Information about the new features of the product
- Reminder of how to use the platform
- **Monitoring** of emails opening
- Evaluation of churn rate on the different groups
- Automatic run of the prediction
- **Dashboard** with results
- Close follow up of data quality



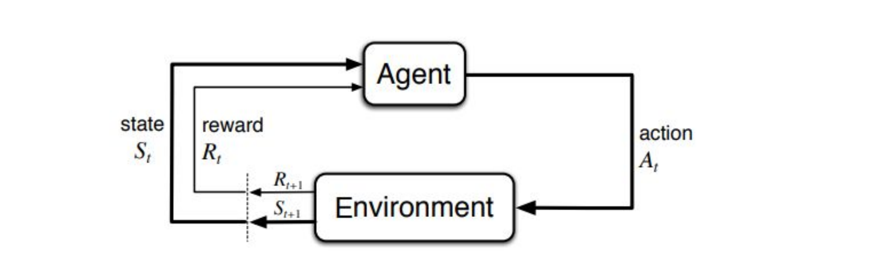
An embedded approach

A data-driven strategy

- Propose the **right message**
- At the **right moment** (check the web behavior, trend and customer profile to determine the right moment)
- With the **right customer**
- Combine with the other models



Link

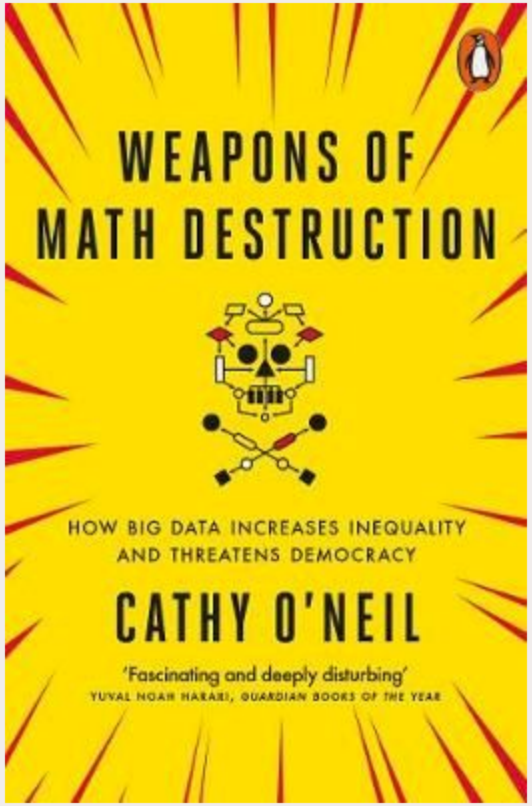


Bias and fairness

With great power comes great responsibility

Data (science) doesn't exist (happen) in a vacuum

- Data isn't objective
  - Inherits social biases and systematic, oppressive patterns
- Convenient samples
  - Not everyone is equally represented
- Lack of sociotechnical engagement and understanding
  - Construct validity, proxies, measurement, causality,...
- Lack of knowledge around historical and scientific faux-pas
  - Eugenics, Phrenology, pseudoscience
- Prediction as the sole thing so strive for, and basis for decision making



Conclusion

What to remember from this presentation

Creating a machine learning model is easy, creating a responsible, ethical and sustainable machine learning project is not so easy

- Focus on the **business goal**
- Always keep in mind how to **industrialize** the project - in order to minimize **refactoring**
- Start small and keep **improving**
- The **big picture** is important - The **model's target** is not always necessarily what you are aiming at, it might only be a **proxy** in a bigger problem
- Be creative, be a **problem solver**
- Keep **everyone on board**
- Keep the **AI act** in mind

References images

https://blogs.gartner.com/paul-debeasi/2019/02/14/training-versus-inference/

https://ml-ops.org/content/mlops-principles

https://python.plainenglish.io/sankeying-with-plotly-90500b87d8cf

https://innovationgrowthlab.org/blog/what-are-trials

https://www.outspot.be/nl

https://twitter.com/mtairas/status/978925669374668801?lang=bg

https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243

https://www.optimizely.com/optimization-glossary/ab-testing/

https://link.springer.com/article/10.1007/s11263-020-01400-4

https://www.pinecone.io/learn/transfer-learning/

https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

https://www.iese.fraunhofer.de/blog/change-point-detection/

https://link.springer.com/article/10.1007/s10489-021-02321-6

https://www.price2spy.com/blog/dynamic-pricing-explained/

https://ai.googleblog.com/2020/04/off-policy-estimation-for-infinite.html

https://iq.opengenus.org/training-vs-inference/

https://ai.googleblog.com/2017/04/federated-learning-collaborative.html