Designing an ML use case E2E

05/05/2023
About myself

UNamur

2008

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

ING

2014

International traineeship - Data scientist - Data scientist in Germany

Dataroots

2017

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.

**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.

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

**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.

**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.

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

Data Strategy & Governance
together solving challenges and creating value

Data Pipeline
robust & test-driven development

Data & AI Solutions
impact-driven analysis, optimization and prediction

End-user integration
users and system friendly consumption of results

Infrastructure
fit-for-purpose, maintainable and scalable
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
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?

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

How can we scale and manage it?
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?

Which data did you use? Why is it recommending me this?
value from innovation comes from the ability to mass produce and distribute your ideas
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
Anlyse and filter
Useful ideas,
Make decision, Based on data available

No value
The lifecycle of a project

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

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

Ideation

POC and validation

Feasibility and priority assessment

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

No value
The lifecycle of a project

Ideation

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

POC and validation

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

Feasibility and priority assessment

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

Scaling

- Fully leverage value
  - Deployment, automation, monitoring, alerting, scheduling, MIOps

Value
Operational challenges

Deploy model systems (not just one off solutions)

Models monitoring and (re)training

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

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
What is Agile?

https://agilemanifesto.org/
What is DevOps?

Wall of Confusion

**DEVELOPMENT**
- Deliver new features
- Product oriented
- Innovation

**OPERATIONS**
- Guarantee stability
- Service oriented
- Rationalization
DevOps engineer

develop code
build
test
plan
release
monitor
deploy
operate
# What is DevOps?

<table>
<thead>
<tr>
<th>Software delivery performance metric</th>
<th>Elite</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment frequency</strong></td>
<td>On-demand (multiple deploys per day)</td>
<td>Between once per week and once per month</td>
<td>Between once per month and once every 6 months</td>
<td>Fewer than once per six months</td>
</tr>
<tr>
<td><strong>Lead time for changes</strong></td>
<td>Less than one hour</td>
<td>Between one day and one week</td>
<td>Between one month and six months</td>
<td>More than six months</td>
</tr>
<tr>
<td><strong>Time to restore service</strong></td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Between one day and one week</td>
<td>More than six months</td>
</tr>
<tr>
<td><strong>Change failure rate</strong></td>
<td>0%-15%</td>
<td>16%-30%</td>
<td>16%-30%</td>
<td>16%-30%</td>
</tr>
</tbody>
</table>

DevOps + ML = MLOps?
MLOps purpose?

🔍 Unify the release cycle
 isKindOfClass
🔧 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.
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**.

<table>
<thead>
<tr>
<th>ID</th>
<th>Snapshot time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01/01/2022</td>
</tr>
<tr>
<td>2</td>
<td>01/01/2022</td>
</tr>
<tr>
<td>1</td>
<td>01/02/2022</td>
</tr>
<tr>
<td>2</td>
<td>01/02/2022</td>
</tr>
</tbody>
</table>

- 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**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Imputed feature (median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>null</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>null</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>Imputed feature (median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>2</td>
</tr>
<tr>
<td>null</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>null</td>
<td>2</td>
</tr>
</tbody>
</table>

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

---

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

Nice cake, can I have the recipe?
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
Quality assurance

Code quality includes

• Linting
• Unit tests
• Data pipeline tests
• ML model pipeline tests
• Application pipeline tests
Most common drift = Real data diverging from baseline data

- **Prediction drift** or change in $P(Y_{\text{prediction}}|X)$ is a shift in the model's predictions.
- **Concept drift** or change in $P(Y|X)$ is a shift in the actual relationship between the model inputs and the output.
- **Label drift** or change in $P(Y_{\text{ground truth}})$ is a shift in the model's output or label distribution.
- **Feature drift** or change in $P(X)$ is a shift in the model's input data distribution.

**Training data with decision boundary**

$P(Y|X)$

*Probability of y output given x input*

**Concept Drift**

$P(Y|X)$ Changes

- Reality/behavioral change
- Relationships change, not the input

**Data Drift**

- **Label Drift**
  - Output data shifts
  - $P(Y)$ Changes
- **Feature Drift**
  - Input data shifts
  - $P(X)$ Changes
- **Virtual Drift**
  - Data changes but boundary still works
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 $P(Y \text{ Ground Truth})$ 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
- Test for feature and data

ML Model Pipeline
- Tests for model development

Application Pipeline
- 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

Continuous training pipeline:
1. Data ingestion
2. Data validation
3. Data transformation
4. Model training/tuning
5. Model evaluation
6. Model validation
7. Model registration

Retraining trigger
ML metadata & artifact repository
Training pipeline metadata and artifacts

Dataset & feature repository
Trained model
Model registry

Data processing engine
Model training engine
Model evaluation engine
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

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>active</td>
<td>active</td>
<td>active</td>
<td>active</td>
<td>active</td>
</tr>
<tr>
<td>No churn</td>
<td>No churn</td>
<td>No churn</td>
<td>No churn</td>
<td>No churn</td>
<td>No churn</td>
</tr>
</tbody>
</table>

Terminal state

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>active</td>
<td>inactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No churn</td>
<td>No churn</td>
<td>Churn</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6
--- | --- | --- | --- | --- | ---
0 | 0 | 0 | 0 | 0 | 0
No churn | No churn | No churn | No churn | No churn | No churn

Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6
--- | --- | --- | --- | --- | ---
1 | active | inactive | Churn | Terminal state | Terminal state
No churn | No churn | Churn | Terminal state |
Target creation the first trade off - Option 2 - adding last 3 months as 1

Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6
---|---|---|---|---|---
0 | 0 | 0 | 0 | 0 | 0
No churn | No churn | No churn | No churn | No churn | No churn

Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6
---|---|---|---|---|---
1 | 1 | 1 | | | 
No churn | No churn | Churn | Terminal state | | 

Terminal state
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)

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)

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

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.

**Batch predictions**

Monthly Dataset - constructed on the historical monthly data.

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.

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

<table>
<thead>
<tr>
<th>Training set</th>
<th>Validation set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Independent validation</td>
<td></td>
</tr>
</tbody>
</table>

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

- Training set
- Validation set
- Test set

Independent validation

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

Check your output in detail
In depth analysis of the residuals, probabilities, etc
Mislabelling - disagreement in the target
The average American generates 16.4 tons of CO₂e emissions in a year.

The adult human brain runs continuously, whether awake or sleeping, on only about 12 watts (0.0000012 MWh).

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

Experimentation is not carbon neutral

<table>
<thead>
<tr>
<th>Model</th>
<th>Energy consumption, MWh</th>
<th>CO₂e emissions, tons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolved Transformer</td>
<td>7.5</td>
<td>3.2</td>
</tr>
<tr>
<td>T5</td>
<td>85.7</td>
<td>46.7</td>
</tr>
<tr>
<td>Meena</td>
<td>232</td>
<td>96.4</td>
</tr>
<tr>
<td>Gshard 600B</td>
<td>24.1</td>
<td>4.8</td>
</tr>
<tr>
<td>Switch Transformer</td>
<td>179</td>
<td>72.2</td>
</tr>
<tr>
<td>GPT-3</td>
<td>1,287</td>
<td>552.1</td>
</tr>
<tr>
<td>PaLM</td>
<td>3,181</td>
<td>271</td>
</tr>
</tbody>
</table>

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%)

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?

Recall = 56 * 10/12
Performance vs correlation vs causality

Predicting churn is not always enough to prevent it

Top 4 features

1. Days in contract
   - Small
   - Medium
   - High

2. First product
   - Small
   - High

3. Savings
   - Small
   - Medium
   - High

4. Frequency of usage
   - Small
   - Medium
   - High

Legend
- Towards non-churn
- Neutral
- Towards churn

- 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

<table>
<thead>
<tr>
<th></th>
<th>Will churn if targeted</th>
<th>Will churn if not targeted</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Persuadable</td>
<td>No need to persuade</td>
</tr>
<tr>
<td>Yes</td>
<td>Not persuadable</td>
<td>Better off not Persuading</td>
</tr>
</tbody>
</table>

Yes          No
Experiment design

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

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.

Different from a random trial, because we select a subpopulation with high chance of churn
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
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://www.optimizely.com/optimization-glossary/ab-testing/
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://www.price2spy.com/blog/dynamic-pricing-explained/
https://iq.opengenus.org/training-vs-inference/
https://ai.googleblog.com/2017/04/federated-learning-collaborative.html