Machine Learning System Design

Andrei Paleyes

Data Science Africa 2022, Arusha

About me

- PhD student in the University of Cambridge
- Interested in systems for ML and ML for systems
- Background in software engineering

About you!

Raise your hands if you ever:

About you!

Raise your hands if you ever:

- Programmed

About you!

Raise your hands if you ever:

- Built a website or a mobile app

About you!

Raise your hands if you ever:

- Trained an ML model for fun or learning

About you!

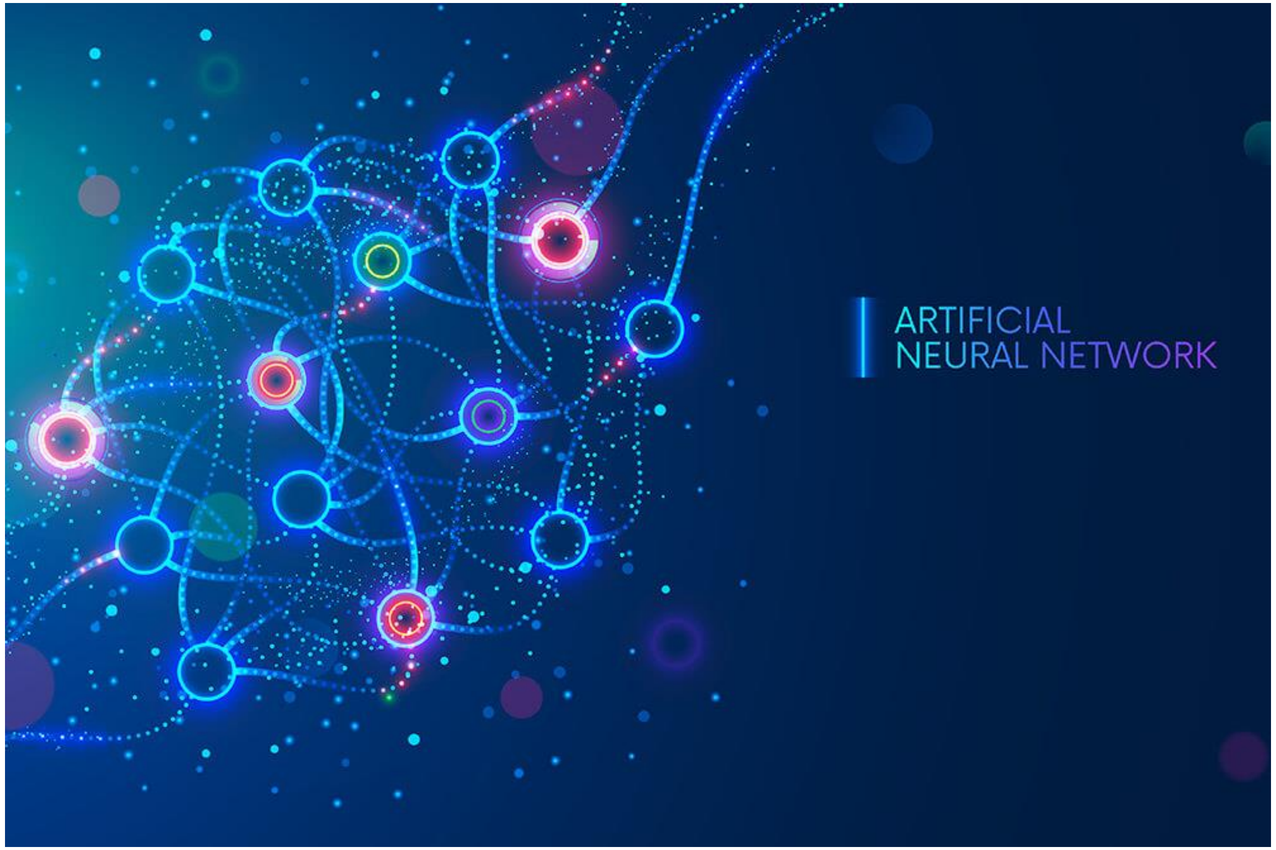
Raise your hands if you ever:

- Did something useful with an ML model

What is machine learning?

What is machine learning?

Machine learning (**ML**) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks.



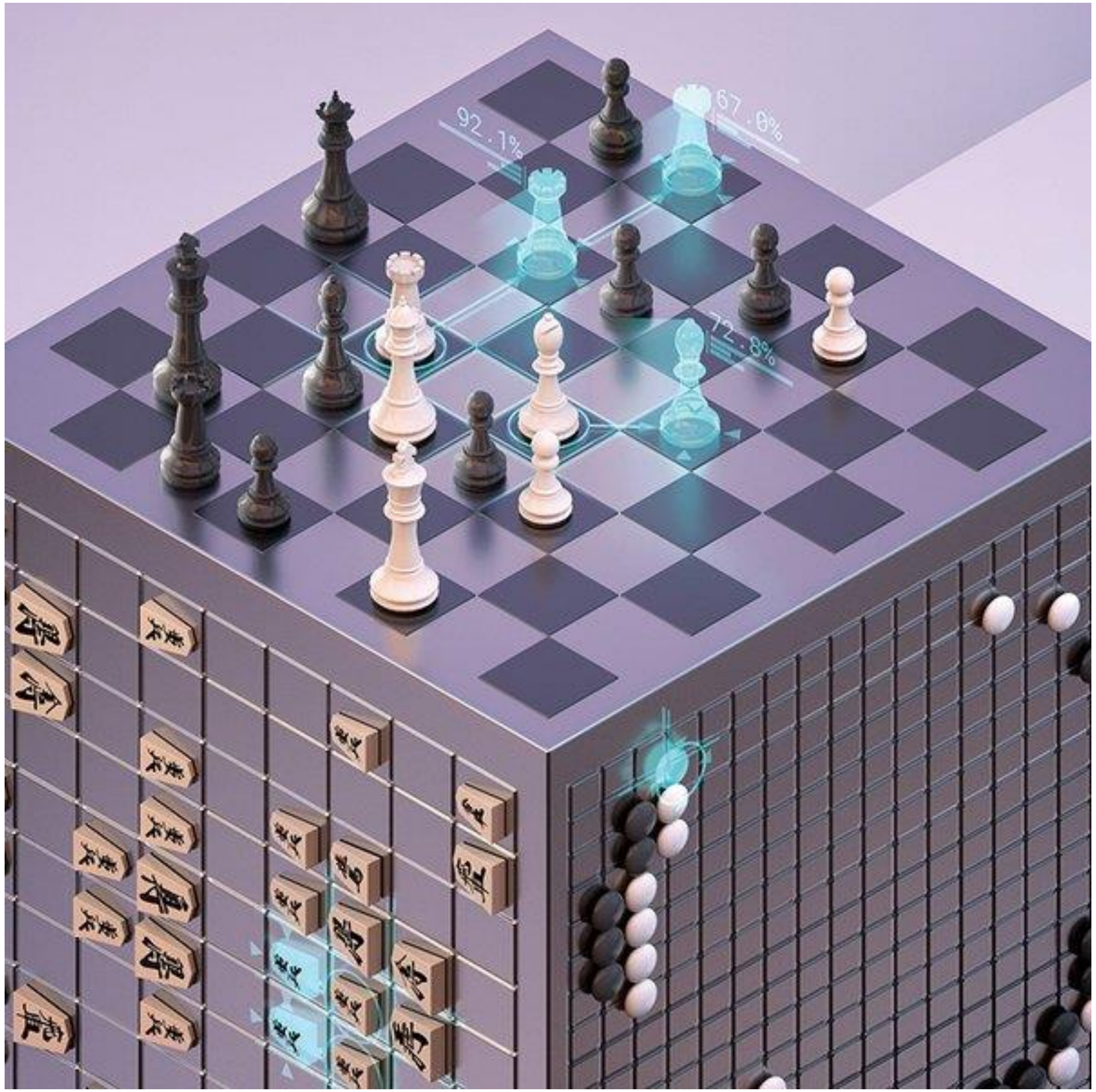
Sources: Wikipedia (text) and Bernard Marr (image)

?

What is machine learning?

What is machine learning?

Model + Data = Machine Learning


92.1%
67.0%
72.8%

What is machine learning?

Model + Data + System = Deployed Machine Learning

How to deploy ML systems

in 6 steps

How to deploy ML systems

in 7 steps

and maybe more

?

Step 1 – ask a question

First question you need to ask is …

“Do I really need ML?”

First question you need to ask is …

"Do I really need ML?"

because maybe you don't

Example 1: Amazon vs Strawberries

amazon fresh

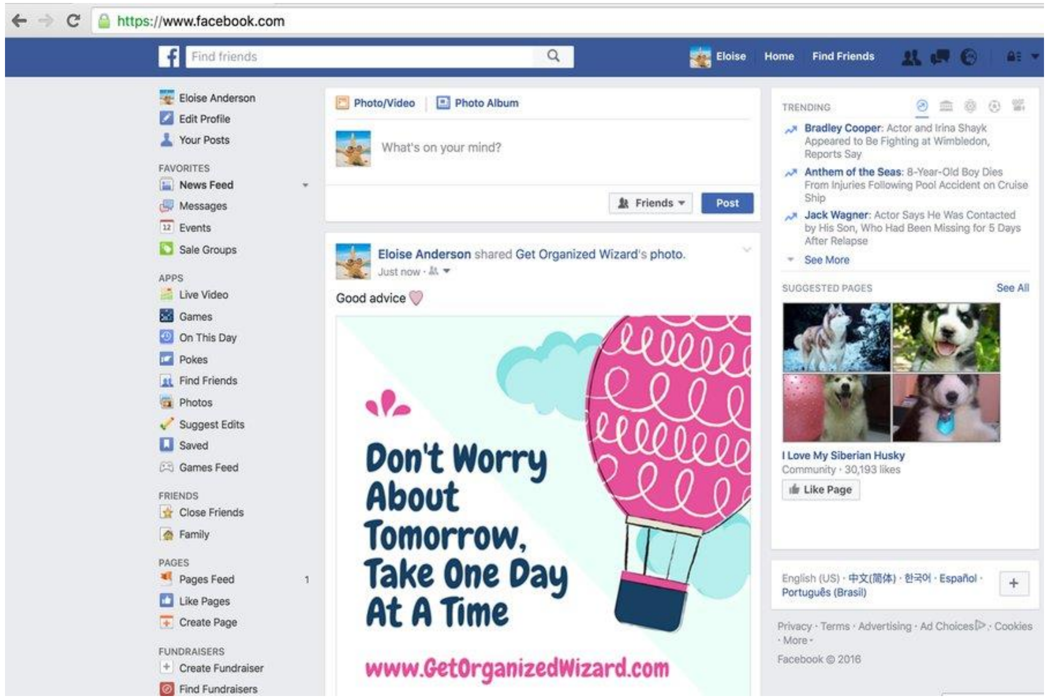
Example 1: Amazon vs Strawberries

"If you think that machine learning will give you a 100% boost, then a heuristic will get you 50% of the way there."

Martin Zinkevich, Google

?

Example 2: Facebook newsfeed



2006

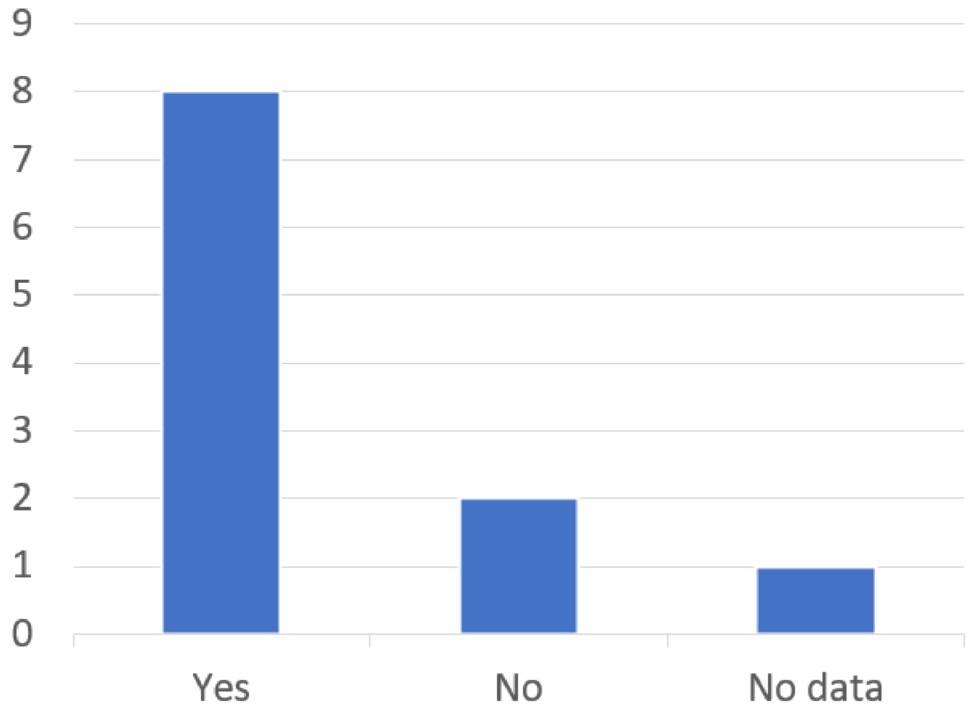
2022

Example 3: Temperature in Arusha

Will temperature in Arusha reach 20° C today?

Example 3: Temperature in Arusha

Will temperature in Arusha reach 20° C today?



Data from https://www.timeanddate.com/weather/tanzania/arusha/historic

Take home point

The best way to design an ML system often is not to design one.

Step 2 – how to measure success

Ask this early!

- No, seriously
- Ask this question early!

Ask this early!

- No, seriously
- Ask this question early!

- Remember the goal
- Don’t overfocus on ML

Other questions to consider

- Who are my end users?
- What are the biggest risks?
- Is there data? Where and in what form?

Check out ML Canvas, https://www.ownml.co/machine-learning-canvas

Step 3 – Find the data

Find the data

- Not a trivial problem!
- Unlike in exercises or academic research

Why getting data is hard?

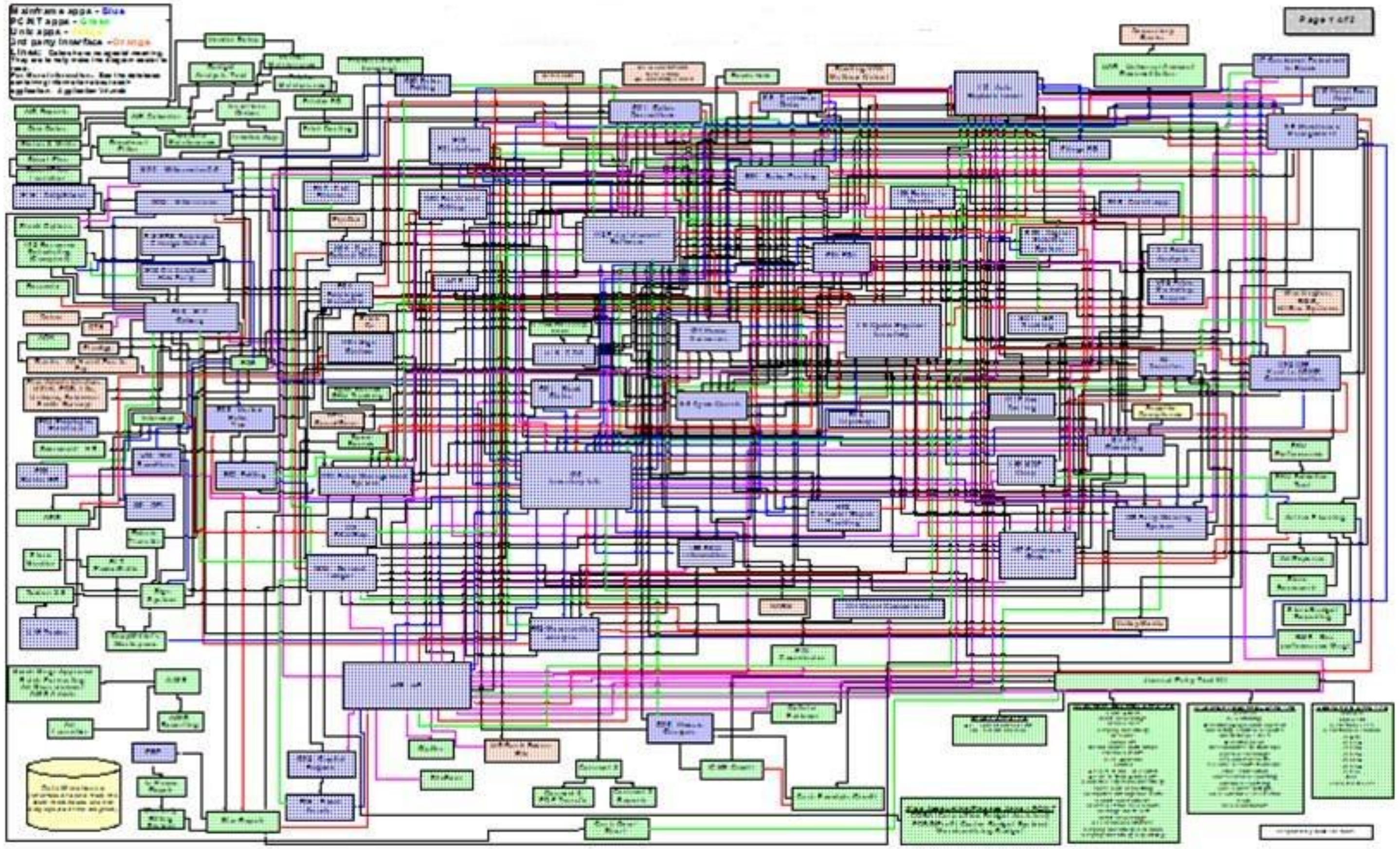
Data might be

- Split between sources
- Not saved
- Different between training and inference
- Unlabeled
- Not clean

?

Example 1: Twitter

Can you find all user data here?



Scaling Big Data Mining Infrastructure — The Twitter Experience. Jimmy Lin and Dmitriy Ryaboy, 2013

Example 2: Atlanta Fire Department, project Firebird

?

- 12 datasets
  - History of incidents
  - Business licenses
  - Households
  - etc.
- Join data on buildings by address
- Took weeks!

Example 2: Atlanta Fire Department, project Firebird

?

- P.O.Box 2798 Njiro Hill, Arusha Tanzania
- Institute of Accountancy in Arusha, Tanzania

Other questions to consider

- What generates the data?
- How to access it?
- In which format is it?
- Is it complete?

Step 4 – Store the data

Storage options

- Memory
- Text file(s)
- CSV file(s)
- JSON file(s)
- SQL database
- NoSQL database
- Data stream

How do I choose?

Depends on your use case! Two main considerations:

- Data modality
- Purpose

Modalities

- Tabular data
- Time series
- Images
- Video
- Free text

Purpose

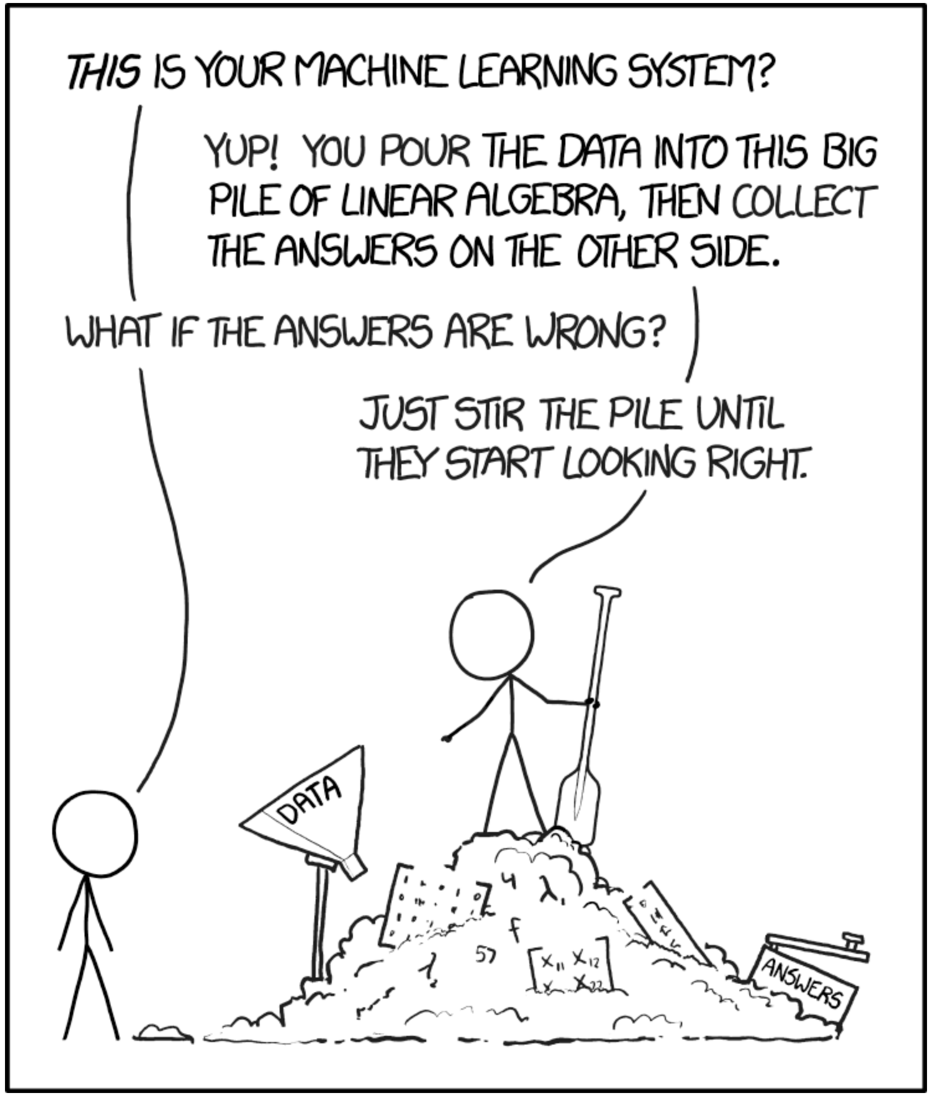
- To play with on your own
- For fellow data scientists
- For training
- For batch processing
- For online prediction

Other considerations

- Structure (e.g. nested)
- Size
- Performance

Step 5 – train a model

Model training

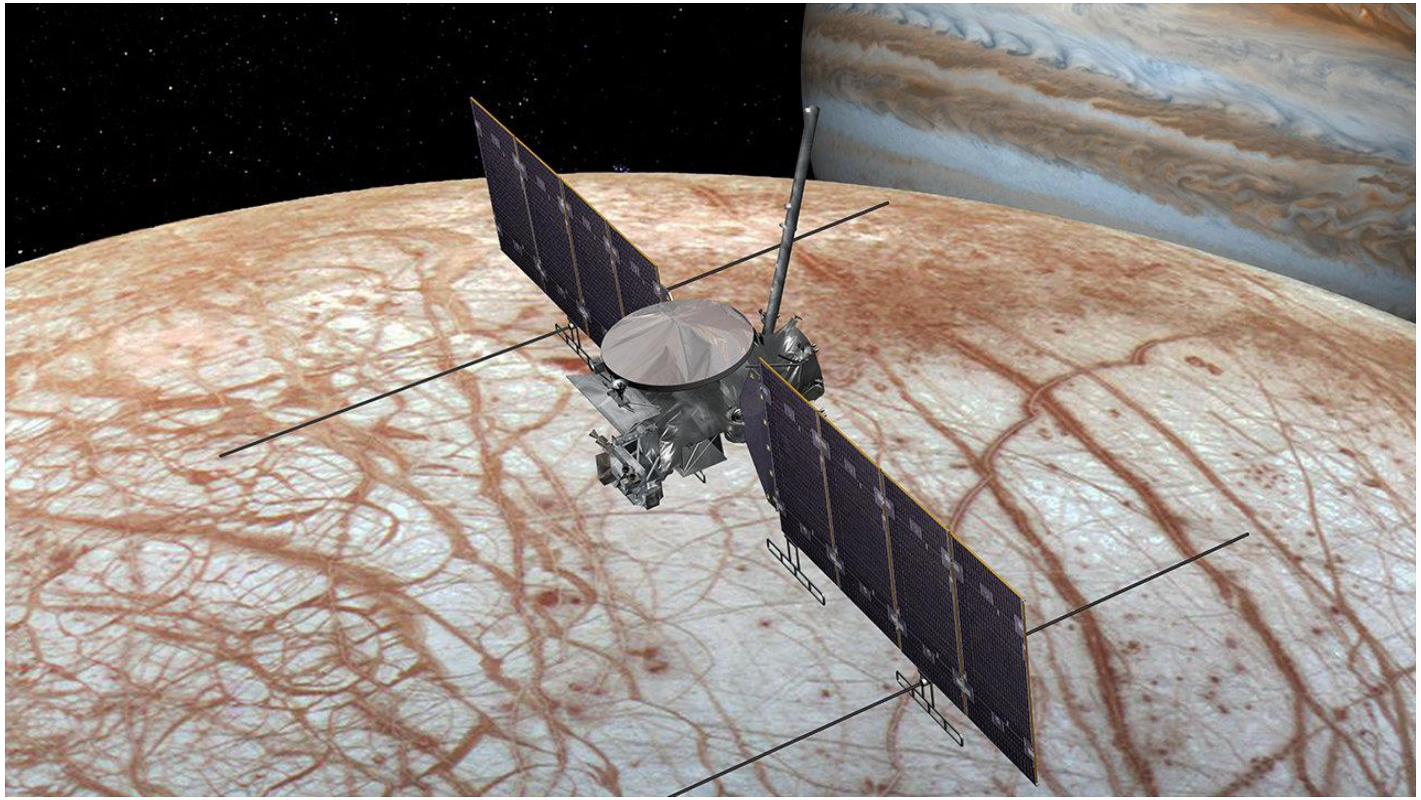


Model selection

Complex model is not necessarily the best!

Why? Let’s ask spacecrafts!

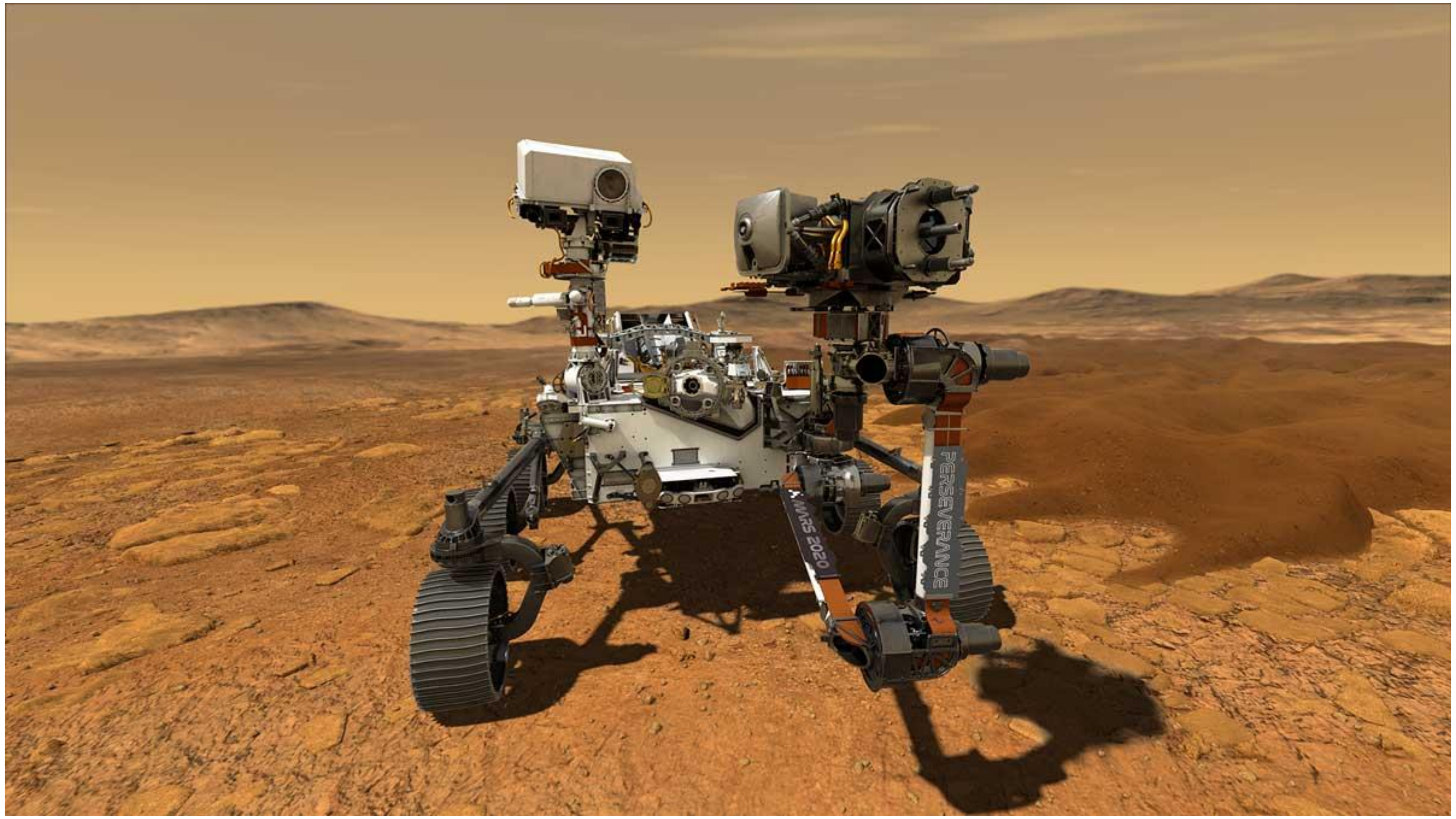
Example 1: Europa Clipper, 2024



PCA for anomaly detection!

Enabling onboard detection of events of scientific interest for the Europa Clipper spacecraft. Wagstaff et al, KDD 2019

Example 2: Perseverance, 2020



Random forest for landmark registration!

Precision instrument targeting via image registration for the Mars 2020 rover. Doran et al, IJCAI 2016

Why?

Why?

- Hardware constraints
- Reliability
- Interpretability
- Easy to use

Step 6 – Host the model

Where models can live

- Your machine
- Cloud server
- Mobile phone

Scenario 1

Learning about random forests with a Jupyter Notebook.

Where would the model live?

Scenario 1

Learning about random forests with a Jupyter Notebook.

Where would the model live?

On your own machine!

Scenario 2

Using simple linear regression in a mobile phone app

Where would the model live?

Scenario 2

Using simple linear regression in a mobile phone app

Where would the model live?

On the phone!

Scenario 3

Very deep neural net that uses a lot of memory and CPU to personalize website functions for users.

Where would the model live?

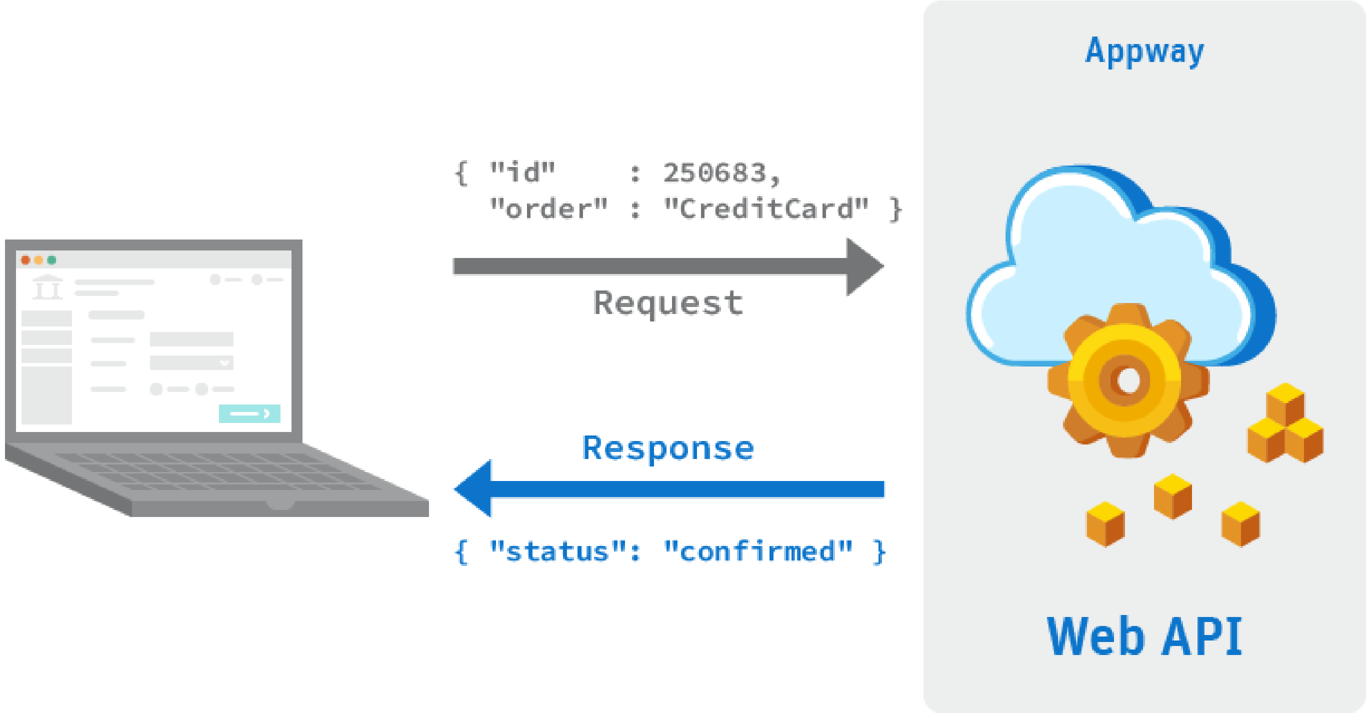
Scenario 3

Very deep neural net that uses a lot of memory and CPU to personalize website functions for users.

Where would the model live?

Dedicated host or cloud

A service!



Scenario 4

A model that updates book recommendations for users once a day.

Where would the model live?

Scenario 4

A model that updates book recommendations for users once a day.

Where would the model live?

Dedicated host or cloud

Step 7 - Monitoring and updating

A myth

“If we don’t do anything, model performance remains the same.”

Chip Huyen, Snorkel AI

Data tends to drift

- Seasonality
- Change of habits
- Unforeseen factors
- Unexpected events

Can happen both in features and in labels!

Monitor for drifts

- Distribution of feature values
- Distribution of model predictions

Drift example

Drift example: COVID-19

- Online shopping patterns changed – Stitch Fix
- New terminology affected translation models – Facebook
- Mobility patterns changed – Google

Business metrics too!

Model improvement ≠ business improvement

Model that improves clicks ≠ better conversion

150 successful machine learning models: 6 lessons learned at Booking.com. Bernardi et al, KDD 2019

When to update?

When to update?

To get started

- When you feel the need (use metrics!)
- On a set schedule, e.g. once a month

Eventually this can become automated

- AWS and Netflix deploy multiple times a day

Other things to consider

- Good software engineering practices
- Fairness, law and ethics
- Security
- Quality assurance
- User interface

A note on tools

- Their name is Legion
- Specific recommendations are impossible to give
- Focus on the goal and architecture
- Leverage available expertise

Where to go from here?

Learning by doing

- Practical lab!
- Try doing something yourself

Where to go from here?

Good resources

- Stanford MLSys Seminar series

https://mlsys.stanford.edu/

- Chip Huyen's blog and book

https://huyenchip.com/blog/

- Rules of ML

https://developers.google.com/machine-learning/guides/rules-of-ml/

- Coursera

https://www.coursera.org/specializations/machine-learning-engineering-for-production-mlops

- ML@CL website

https://mlatcl.github.io/

Where to go from here?

Papers

- Monitoring and explainability of models in production. Klaise et al., ICML DMML workshop 2020
- Hidden Technical Debt in Machine Learning Systems. Sculley et al., NeurIPS 2015
- Challenges in Deploying Machine Learning: a Survey of Case Studies. Paleyes et al, ACM Comp. Surv. 2022
- Scaling Big Data Mining Infrastructure — The Twitter Experience. Lin and Ryaboy, KDD 2013
- MLOps: A Taxonomy and a Methodology. Testi et al., IEEE Access 2022
- 150 successful machine learning models: 6 lessons learned at Booking.com. Bernardi et al, KDD 2019
- Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. Ashmore et al., ACM Comp. Surv. 2021
- Software engineering for machine learning: A case study. Amerishi et al., ICSE 2019
- Data lifecycle challenges in production machine learning: a survey. Polyzotis et al., ACM SIGMOD Record 2018

Summary

- Sometimes ML is not needed
- Get you success metric right
- Getting data can be hard
- So can be storing
- Simple models often work best
- Models live in different places
- Monitor for drift in data and metrics
- Also consider
  - Fairness, law and ethics
  - Privacy and security
  - Quality assurance
  - Good software engineering practices
  - User interface

Questions?

Appendix

Good software engineering practices

- Version control is good (e.g. git)
- Code reviews are good
- Unit tests are good
- Separation of concerns is good
- Naming is important

Data ethics

- Who owns the data?
- How was it collected?
- Do you have explicit permission to use the data?
- Can you identify individuals from the dataset?
- Can you use model trained on this dataset commercially?
- Can you use privacy techniques?

Fairness

- Can the training data contain biases?
  - Explicit biases
  - Hidden biases aka proxies
- Are the classes balanced?
- Is there a potential to aggravate bias?

Law

- Is your area highly regulated?
  - Healthcare
  - Finance
  - Judicial
- Will you operate somewhere with country-level laws?
  - EU – GDPR
  - Canada - PIPEDA
  - Kenya – Data Protection Act (similar laws exist in Uganda, Nigeria and South Africa)

Security

- Data poisoning
- Model reverse engineering aka model stealing
- Model inversion

Quality assurance

- Do you have acceptance criteria for the system?
- How does model performance translate to business value?
- Can you test model in real life, e.g. with A/B test?
- If not, maybe you can use a simulation?

User interface

- Good UX increases trust
- Bad UX limits adoption
- ML terms are not easy to understand
- Focus on what user wants to see, not on ML