Introduction to Machine Lecture Engineering

BECM33MLE — Machine Learning Engineering

Dr. Tomáš Báča[1], Dr. Jan Brabec[2], Jan Lukány, MSc[3]

[1]Multi-Robot Systems group, CTU in Prague
[2]CISCO
[3]Datamole

What is Machine Learning Engineering?

Research (ML)

- ideas & experiments
- exploration
- feature design
- modeling
- evaluation & benchmarking

Engineering (E)

- robust software
- system design
- ML pipelines
- data management
- scalability, testing

Operations (MLOps)

- production lifecycle
- deployment
- monitoring & alerting
- automation & infrastructure
- user interface

Lecturers

Dr. Tomáš Báča (TB)

- Assistant professor @ Multi-robot Systems Group (MRS)
- Co-founder of Fly4Future, and Eagle.one
- PhD on Multi-UAV Distributed Sensing
- Research in Aerial robotics, Control, Realtime Mapping and Localization

Dr. Jan Brabec (JB)

- Principal AI Researcher @ Cisco
- Building systems detecting cybersecurity threats
- Ph.D. on ML in Cybersecurity @ CTU in Prague

Jan Lukány, M.Sc. (JL)

- ML/Data Engineering Team Lead @ Datamole
- Graduated at CTU FIT (Machine Learning)
- Main technical focus: "Engineering" part of MLE
- 10 years of experience from industry (agritech, biotech)

Lectures

- lectures are optional, no need to contact us in case you can not come
- slides and other materials will be available on CourseWare at the day of the lecture

Time plan

Week	**Date** (lecturer)	**Topic**
1	Sep, 23 (TB)	Introduction
2	Sep, 30 (TB)	Classical MLE tools and methods, datasets
3	Oct, 07 (TB)	Deep MLE tools and methods, datasets
4	Oct, 14 (JB)	ML System Design and Architecture
5	Oct, 21 (JL)	Data storage frameworks
-	Oct, 28 (-)	Canceled - National holiday
6	Nov, 04 (JL)	Machine learning model execution paradigms
7	Nov, 11 (JB+TB)	Ground truth management
8	Nov, 18 (JB)	Production metrics and observability
9	Nov, 25 (JB)	ML and AI technical debt
10	Dec, 02 (TB)	AI engineering, MCP
11	Dec, 09 (TB)	Containerization (Docker, Apptainer)
12	Dec, 16 (TB)	Development workflows (git, CI-CD, BDD)
13	Jan, 06 (TB)	MLE and AI on the "edge"

Practicals

- practicals are **compulsory**
- slides and other materials will be available on CourseWare

Time plan

Week	Date	Topic
1	Sep, 23	Introduction
2	Sep, 30	GUI + Project setup
3	Oct, 07	Datasets
4	Oct, 14	Reinforcement learning in a virtual environment
5	Oct, 21	Machine learning basics
-	Oct, 28	Canceled - National holiday
6	Nov, 04	Implementation details
7	Nov, 11	Machine learning advanced
8	Nov, 18	Deployment
9	Nov, 25	Going to market
10	Dec, 02	Workshop
11	Dec, 09	Workshop
12	Dec, 16	Workshop
13	Jan, 06	Project presentations

Semestral project

- practicals will revolve around **team semestral project**
- students will pick their own project topic: (**ML-equipped service or product development**)
- students work on the project throughout the semester, reporting on the progress regularly
- presentation of the project in the final weeks
- practical mini-lectures and tips during the semester

Vaguely inspired *by How to make (almost) anything*

- MIT course by prof. Neil Gershenfeld
- Successfully replicated at CTU by Dr. Jiri Zemanek as JVC (Jak Vyrobit (temer)Cokoliv)

Evaluation

- final evaluation in the form of **graded assessment** (klasifikovany zapocet)
- **no final exam**
- the activity in the **practicals** is what defines if & what mark you obtain
- up to 100 points from the semester

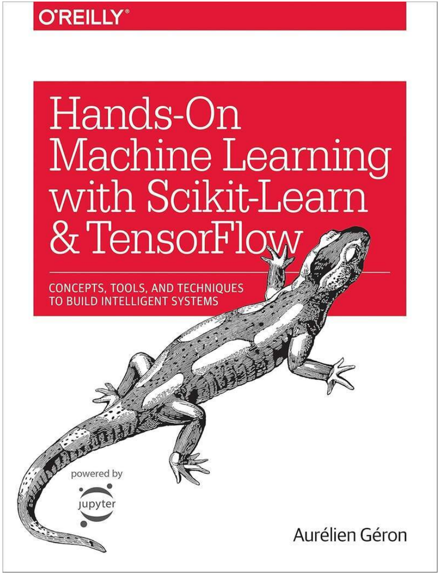
ECTS score

The total amount of points is the summation of

- ints for the project (up to 50 points),
- The points for homeworks (up to 15 points),
- The points for documentation (up to 15 points),
- The points for the project presentation (up to 20 points),

Points	[0,50)	[50,60)	[60,70)	[70,80)	[80,90)	[90,100]
Mark	F	E	D	C	B	A

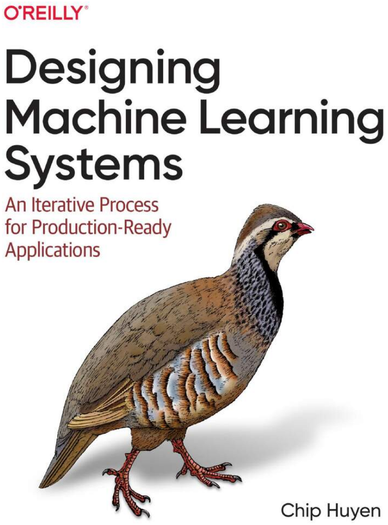
Recommended literature



- comprehensive introduction the ML concepts and common problems faced in ML (the *research* part of MLE)
- practical examples for classical methods shown in Scikit-Learn
- deep learning examples in TensorFlow
- project-oriented approach
- **basis for Lectures 1–2**
- 1st edition (but technically a 4th edition)

[1] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022

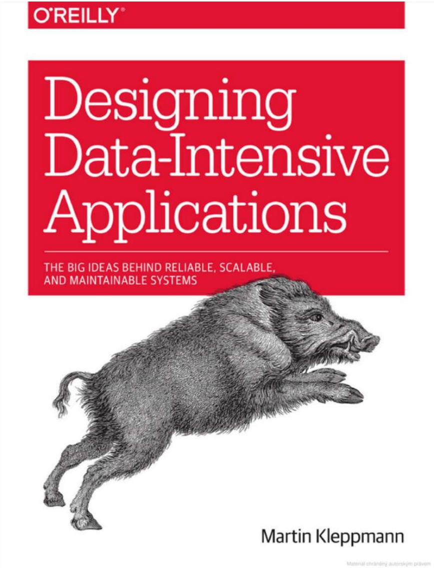
Recommended literature



- comprehensive overview of the whole MLE field
- includes:
  - data engineering
  - feature engineering
  - model deployment
  - continual learning and testing in production

[2] C. Huyen, *Designing machine learning systems*. " O'Reilly Media, Inc.", 2022

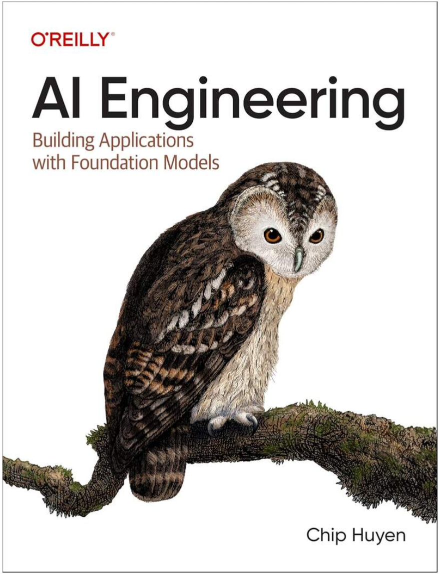
Recommended literature



- going beyond MLE
- **Engineering** of robust, reliable software systems
- databases, data models

[3] M. Kleppmann, *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems.* " O'Reilly Media, Inc.", 2017

Recommended literature



- Going from ML Engineering to *AI* engineering
- LLMs and beyond
- RAG
- Prompt engineering
- Foundation models

[4] C. Huyen, *AI Engineering: Building Applications with Foundation Models*. O'Reilly Media, Incorporated, 2024

Relation to other courses

Deep Learning Essentials (BECM33DPL0), 1^{st} semester

Course going *deep* into the underlying theory of Deep Learning, focusing on ANL structures, learning algorithms, reinforcement learning.

Computer Vision Methods (BE4M33MPV), 2^{st} semester

Course focusing solely in computer vision techniques and algorithms, of which many are ML-based.

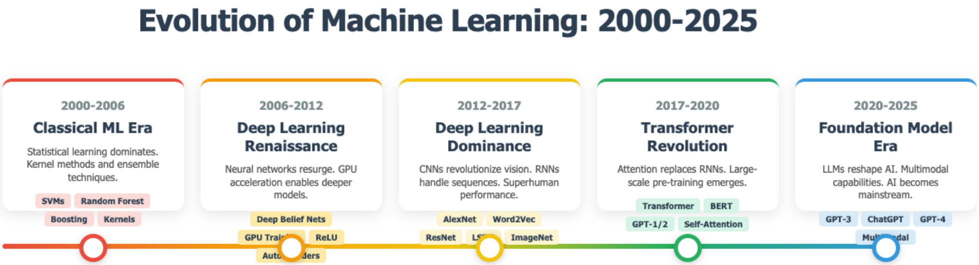
Machine Learning Fundamentals (BECM33MLF), 2^{st} semester

Course going into the underlying theory of machine learning, why and how it even works (VC dimension, PAC learning), and theory behind some fundamental ML models.

Machine Learning Methods (BECM36MLM), 2^{st} semester

Course about advanced ML algorithms, ML working with relationship databases, graph neural networks, neuro-symbolic networks, model interpretability, RL.

Machine Learning Evolution



What is machine learning?

Example of the "traditional" approach

Fabrication of spam-filter by hand:

1. gaining expert knowledge on the particular problem, i.e., finding what words are common in spam
2. fabricating a complex tree of nested conditions and rules
3. evaluating your program
4. tweaking the set of rules and conditions to work on new data

Is it good enough?

- For many problems, yes!
- For many problems, not at all!

Traditional problem-solving approach

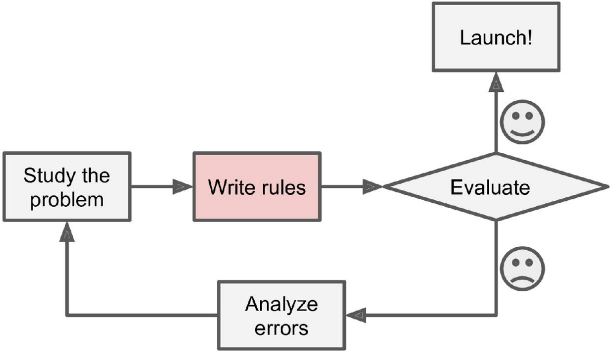


Figure 1: Traditional software development feedback loop [1].

What is machine learning?

The core of ML

- Data (or simulated environment in reinforcement learning)
- A model
- A training process

When to use it?

- The problems that have many parameters that need to be tuned.
- The problems where real-time adaptation to change is crucial.
- The problems that too complex for handcrafting a solution.
- The problems where even the experts don't know how the solution would even look like.

ML problem-solving approach

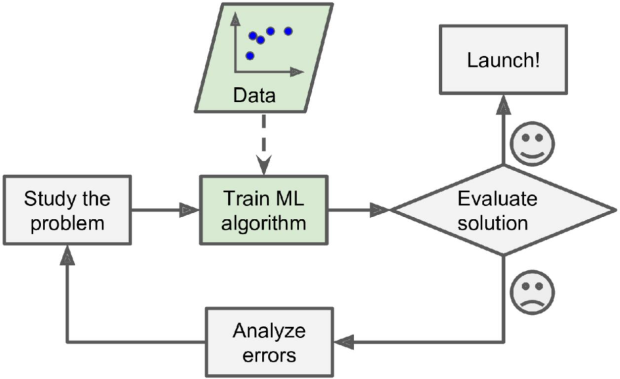


Figure 2: Machine Learning development feedback loop [1].

What is machine learning?

Continuous deployment

- new data might come in from the users
- e.g., new emails being flagged as *spam*
- the ML algorithm will rel-learn on the fly to flag those emails automatically

ML as an automated problem-solving approach

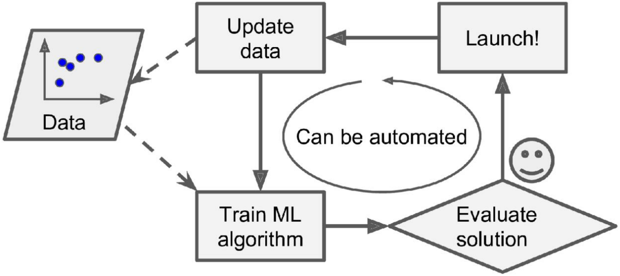


Figure 3: Machine Learning automated development feedback loop [1].

What is machine learning?

Complex problems & data science

- some ML algorithms can be inspected to see what they have learned
  - spam filters' *most common patterns contained in spam email*
  - pattern recognition and segmentation: finding patterns, relationship and clusters in data where humans would not be able to
  - feature recognition: finding hidden relationships in data, finding which features are responsible for outcomes

'

ML helping understand the problem

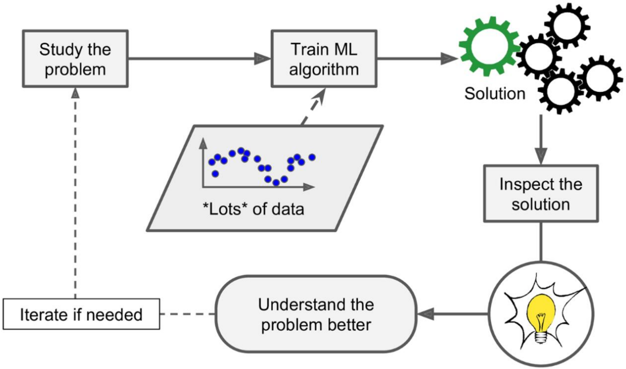


Figure 4: Machine Learning can help understand the problem [1].

Types of Machine Learning Tasks - Classification

- deciding if a data sample belongs to a class
  - spam detection (spam or ham)
  - medical diagnosis (positive on disease)
  - image recognition (dog detected)
- classical methods: KNN, SVM, Decision trees, Random forests

Classification

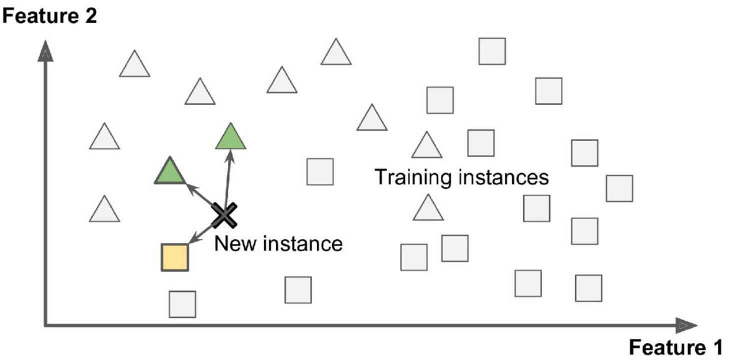


Figure 5: Illustration of classification [1].

Types of Machine Learning Tasks - Regression

- prediction a continuous variable
  - commodity price prediction
  - medical diagnosis and medical analysis
  - financial forecasting
- classical methods: Linear regression, SVM, Random forests

Regression

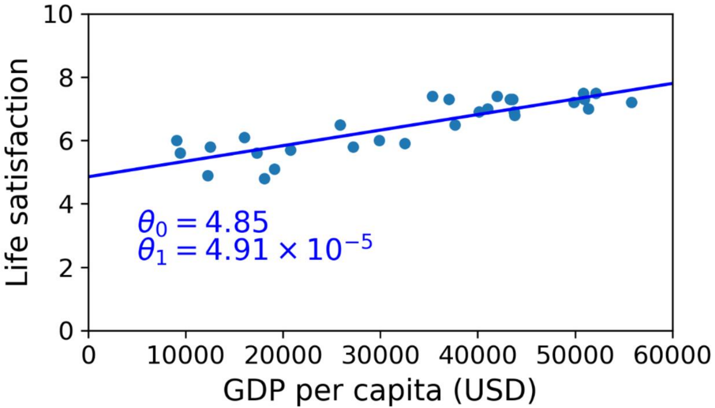


Figure 6: Illustration of linear regression [1].

Types of Machine Learning Tasks - Clustering

- finding groups in data with common properties
  - data exploration and visualization
  - customer segmentation
  - anomaly detection
- classical methods: K-Means, Gaussian mixture models, DBSCAN

Clustering

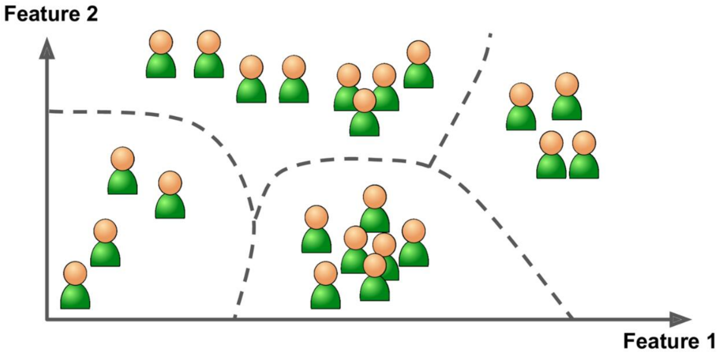


Figure 7: Illustration of clustering [1].

Types of Machine Learning Tasks - Natural language processing

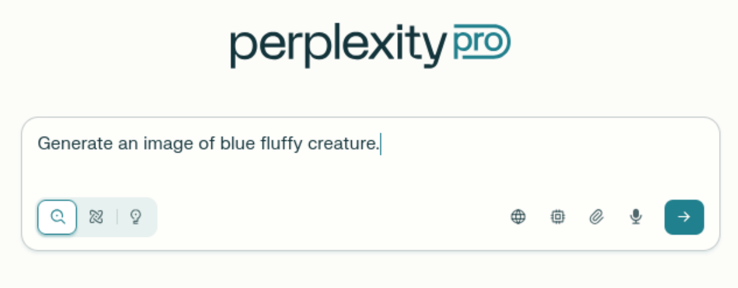
- working with text
  - translation
  - summarization
  - sentiment extraction (financial sector)
  - classification (spam filtering)
- classical methods: not much
- deep methods: transformers, RAG

Natural language processing

Types of Machine Learning Tasks - Content creation

- generating content
  - image generation
  - synthetic data generation
- classical methods: not much
- deep methods: transformers, generative adversarial networks, RAG

Content creation



Types of Machine Learning Tasks - Content creation

- generating content
  - image generation
  - synthetic data generation
- classical methods: not much
- deep methods: transformers, generative adversarial networks, RAG

Content creation

Types of Machine Learning Systems - Supervised/Unsupervised

- requires labelled dataset
- manual labelling
- nowadays almost gone
- ML-aided labelling
  - **Segment Anything 2** as an aid for labelling videos [5]
- example problems:
  - classification
  - pattern recognition
  - speech recognition
  - image classification
  - spam detection

Supervised learning approach

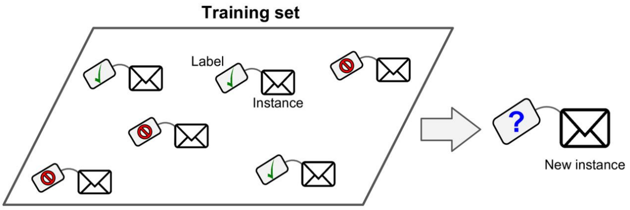


Figure 8: Supervised learning illustration in the form of spam filter classification [1].

Types of Machine Learning Systems - Supervised/Unsupervised

- unlabelled dataset
- we often want to understand what is in the dataset / what is in the data coming into a live system
- typical problems:
  - clustering
  - data visualization
  - dimensionality reduction
  - anomaly detection (cybersecurity, network traffic patterns)
  - association mining (people who buy X also buy Y)

Unsupervised learning approach

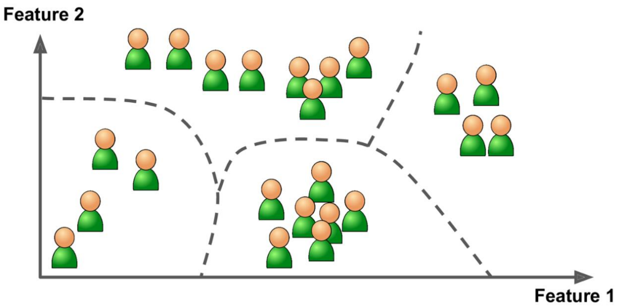


Figure 9: Unsupervised learning illustration [1].

Clustering for visualization

- clustering and dimensionality reduction can help understanding the contents of a dataset
- often used in conjunction with semi-supervised learning (at least something is known to help guide the process)

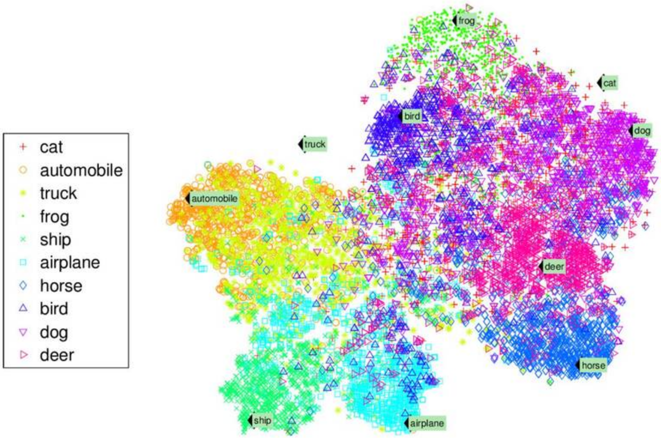


Figure 10: Clustering for visualization [1], [6].

Types of Machine Learning Systems - Semi-supervised learning

- some labels are know, but most are unknown
- the labelled data does not cover the whole space, but other unlabelled data do
- labeling everything would be impractical or impossible

Self-supervised learning

- auto-generating labels from unlabelled data
- no human in the loop (later for fine tuning)
- learning of LLMs

Semi-supervised learning

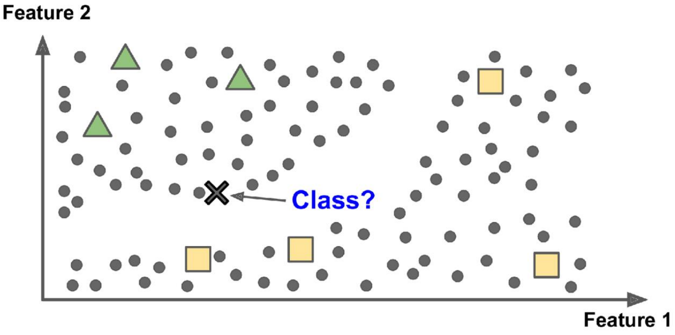


Figure 11: Semi-supervised learning [1].

Types of Machine Learning Systems - Semi-supervised learning

Example

- dataset of partially-labelled images
- unlabelled open-vocabulary language corpus
- learned mapping between space of images and text
- when classifying a new image as an unknown class, a connection can be made between the vector in the image space into a vector in the language space

[6] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," *Advances in neural information processing systems*, vol. 26, 2013

Clustering for visualization

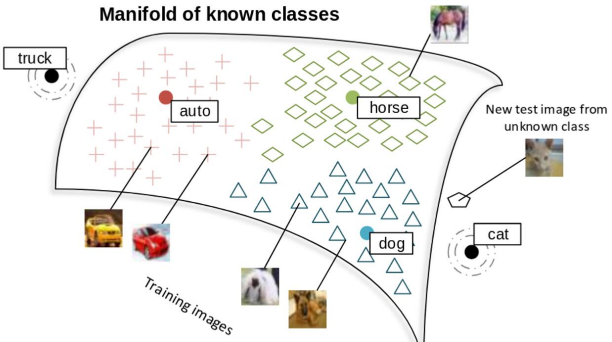


Figure 12: Classification on unseen data [6].

Types of Machine Learning Systems

- an agent lives in an environment (real or simulated)
- action can be made in the environment by using a **policy**
- actions can lead to positive **reward** or negative **penalty**
- the policy is updated as a part of the **learning step**
- both **supervised** and **unsupervised** learning are special cases of reinforcement learning

Reinforcement learning

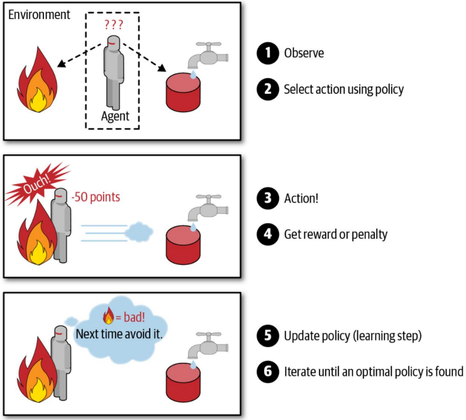


Figure 13: Reinforcement learning [1].

Types of Machine Learning Systems - Batch/Online learning

Batch learning

- ML algorithm is learned once (often a model)
- dataset is obtained, processed and used once
- suitable in some situations:
  - the domain does not change in time
  - the dataset is good, it covers the domain well

Examples

- image classification and object detection in industrial (controlled) environment
- natural language processing (all the common language transformers are initially batch-learned, and then)

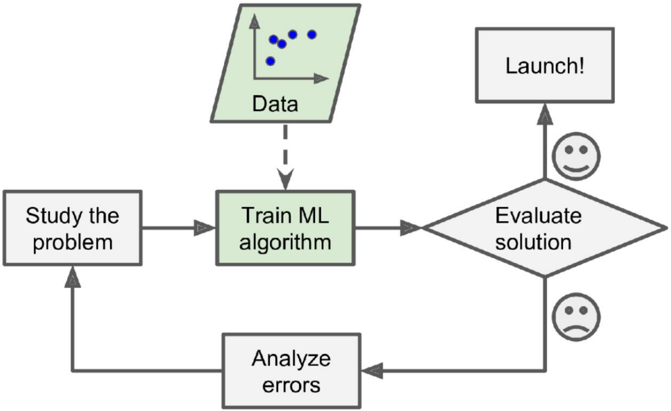


Figure 14: Batch learning approach [1].

Types of Machine Learning Systems - Batch/Online learning

Online learning

- the ML algorithm / model is initially trained using a priming dataset
- the learning is run continuously with new data

Examples

- spam email filtering
- financial forecasting (check FreqTrade and FreqAI)
- fraud detection
- smart counters in Albert ;-)

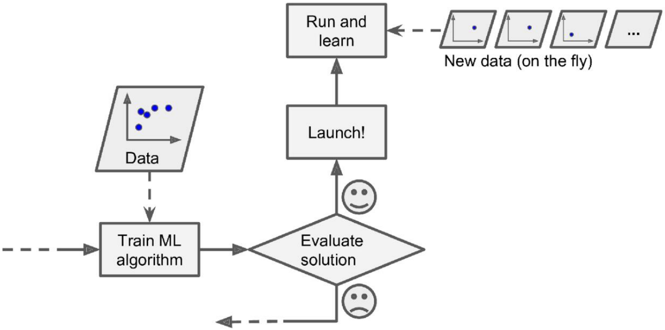


Figure 15: Online learning [1].

Types of Machine Learning Systems - Batch/Online learning

Learning with extra large datasets

- when dataset can not be efficiently stored in memory
- distills to online learning
- large dataset is chopped into small pieces

Examples

- social media feed moderation
- financial transactions analysis

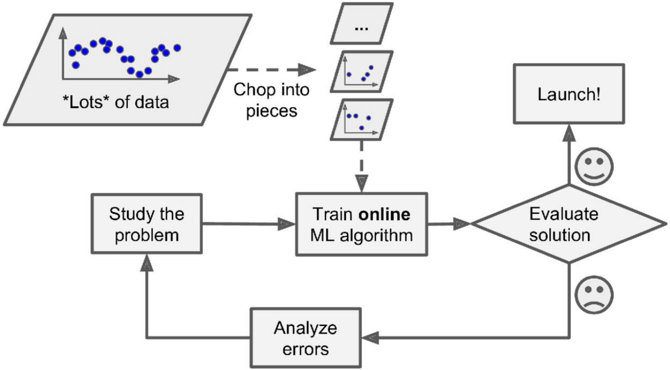


Figure 16: Online learning for large datasets [1].

Types of Machine Learning Systems - Instance/Model-based learning

Instance learning

- data samples from the dataset are used directly in the algorithm
- Most commonly: K-nearest neighbour (KNN), Case-base reasoning

Examples

- Classical image classification (KNN)
- Handwritten OCR
- Medical diagnosis

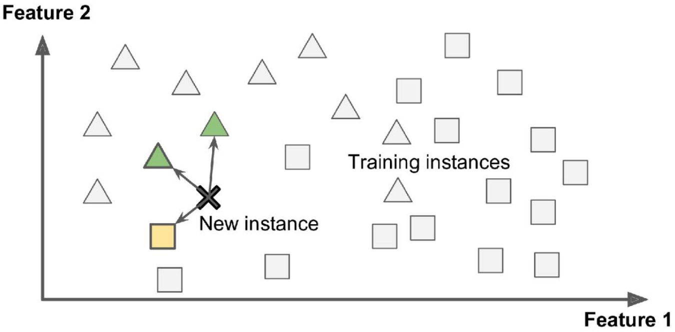


Figure 17: Instance-based learning [1].

Types of Machine Learning Systems - Instance/Model-based learning

- the dataset is
- SVM classifier is the staple in model-based learning
- Decision Tree-based methods
- All the Deep Neural Network algorithms
- Bayesian models
- Reinforcement learning

Examples

- Medical diagnostics
- Image recognition, natural language processing

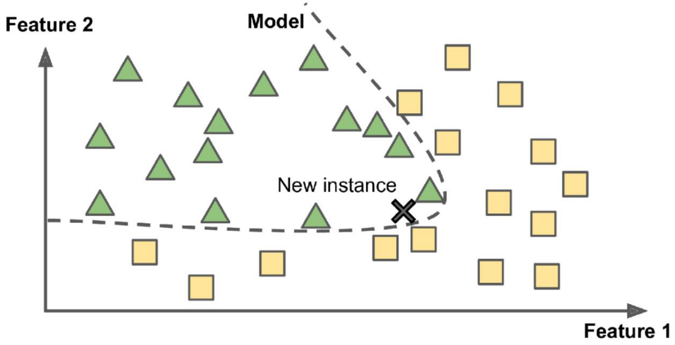


Figure 18: Model-based learning [1].

Types of Machine Learning Systems - Model-based learning in detail

- learning to estimate a relationship between *GDP per capita* of a country and *life satisfaction* $[0, 10]$
- instance-based learning (e.g., KNN) does not work well
  - it only reflects the neighbourhood but not the trend)
  - the data is sparse (max 195 samples) and noisy
  - is more susceptible to noise in the local neighbourhood
- we observe **linear trend**
- let's try to find an **affine function** that fits the data

Model-based learning

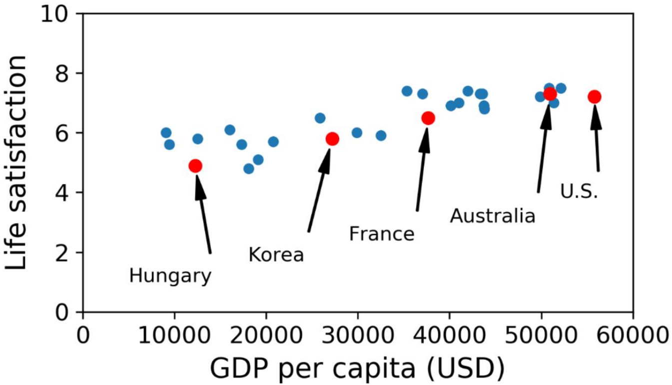


Figure 19: GDP per capita of selected countries [1].

Types of Machine Learning Systems - Model-based learning in detail

The approach:

1. model selection
2. finding parameters
3. evaluating the model
4. possibly finding a different model

Finding the parameters

- Optimization problem
- Optimization criterion (fitness function, performance measure):
  - sum of squares of distance
  - max error

Model-based learning

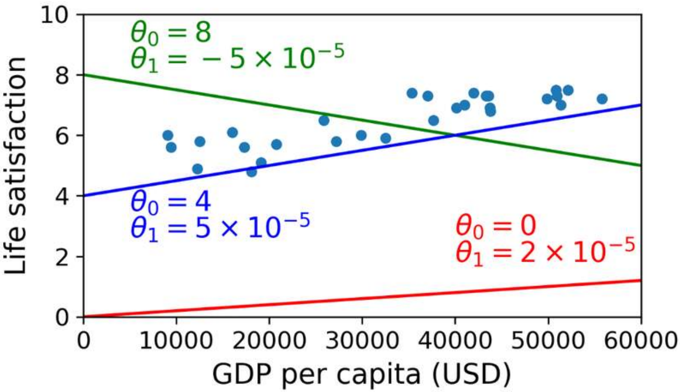


Figure 20: GDP per capita various affine fits [1].

Types of Machine Learning Systems - Model-based learning in detail

- least squares fit given the training dataset
- we can use the model to create predictions for new data
- you can compare the results of the *LinearRegression* to *KNN*, KNN will be worse

Model-based learning

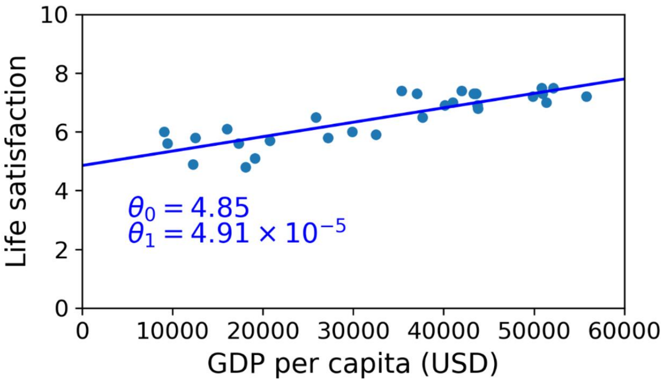


Figure 21: GDP per capita least-squares affine fit [1].

Challenges in ML - Quality of training data

The Unreasonable Effectiveness of Data

- with poor dataset, various ML models perform **equally bad**
- the ML engineer must do a clever decision:
  - if spend more time on finding the right model,
  - if spending time on getting more data or improving the dataset,
  - if spending time on tuning the training process,
- however, getting more data is often very expensive, so in the end, finding the right model might be worth it

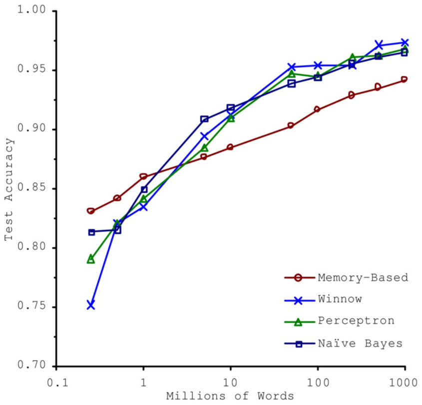


Figure 22: With bad data, various ML models perform equally bad [1], [7].

Challenges in ML - Non-representative data

Non-representative data

- outliers or purely missing coverage of the whole domain
- the original fit we had suddenly does not look so good

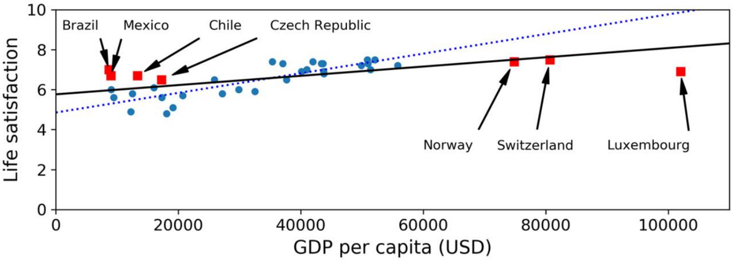


Figure 23: Non-representative training data [1].

Challenges in ML - Irrelevant features

- proper feature selection and feature design is key
  - improves model accuracy
  - reduces overfitting,
  - leads to faster training,
  - enhances interpretability,
  - might make collecting data easier,

Example 1: Medical diagnosis

- dataset may include hundreds of features, including non-medical data
- predicting the risk of diabetes based on
  - eye color,
  - ZIP code,
  - shoe size,
  - parents' shoe size,
- might not yield good results.

Example 2: Image processing

- using pure pixel values
  - histogram normalization
- using color space when color is irrelevant
- using a wrong color space, e.g., RGB
- learning resolution-dependent models

Challenges in ML - Overfitting

- the model has too much free parameters
- the data is too sparse for the dimensionality of the model
- is hard to spot during training: the error on the training dataset is smaller with more complex model
- can be detected with cross-validation

Example

- fitting a high-order polynomial to data that follow a linear trend

Overfitting data with improper model

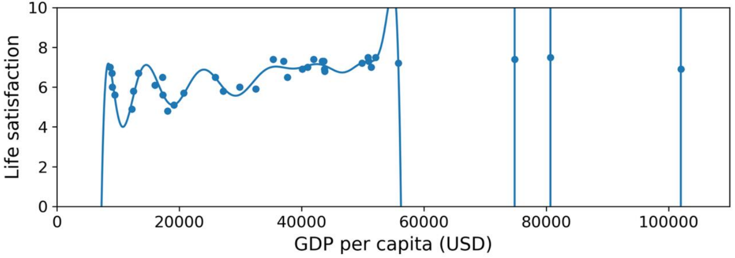


Figure 24: Overfitting illustration [1].

Challenges in ML - Underfitting

- the model is too simple for the data and it does not generalize well
- learning process converges too soon with fitness being bad
- bringing it more data does not help

Deep learning

- Finding the right shape and size of a neural net is a field of study of its own
- too complex and it easily overfits the problem, practically learning the whole dataset
- too simple and it underfits the dataset and generalizes poorly

Underfitting

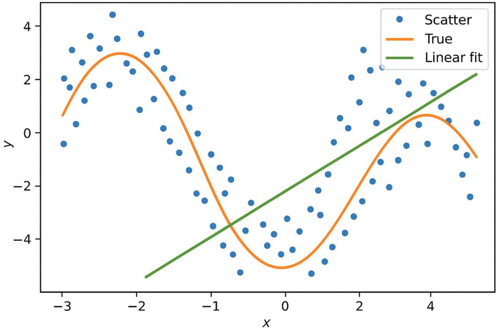


Figure 25: Underfitting illustration.

Challenges in ML - Labelling

- for many problems, labelling is tricky
- image recognition of specialized object

ML can help you with labelling

- by leveraging ML to automate a simpler-subtask of the labelling
- e.g., image segmentation and tracking to automatically label a part of a Video-based dataset

Ground truth management

- **Lecture 7**
- sometimes, knowing the truth from the real-world data is difficult
- building simulators / emulators for initial training might help

Labelling objects in images

Figure 26: How to label objects in an image?

Real-world examples from the aerial robotics world

What ML-related (research-related) real-world problems are we solving in aerial robotics?

Note

- Realworld examples from other fields will be provided in future lectures by the experts from the industry.

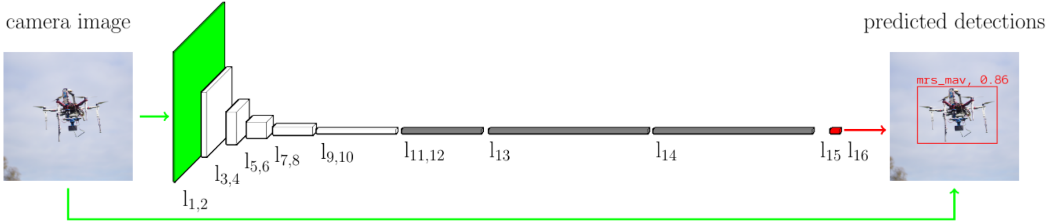
Marker-less drone detection in camera images

- motivated by eagle.one (`http://eagle.one`)
- detection of a flying drone from another drone
- requirements: real-time (max $100\,\mathrm{ms}$ execution), onboard computation, RGB camera input
- related tasks: 3D pose estimation, outlier rejection, tracking

[8] M. Vrba and M. Saska, "Marker-less micro aerial vehicle detection and localization using convolutional neural networks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2459–2466, 2020

Marker-less drone detection in camera images

Typical CNN architecture for object detection



Typical input data

Marker-less drone detection in camera images

Engineering problems:

- gathering custom dataset is costly and difficult
- labelling the dataset is impractical
- estimating size of the drone? small & close = large & far away
- how to make a dataset with precise 3D relative pose

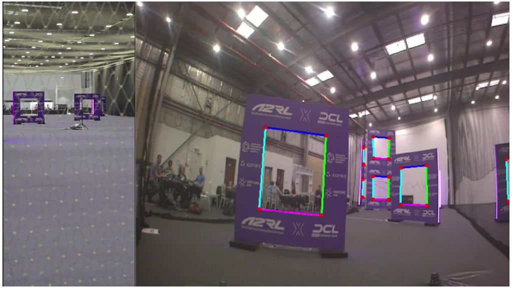
Possible solutions

- flying with a custom drones which know where they are (not ideal, not scalable)
- using onboard-markers for automatic labelling (also not scalable)
- generating large simulated datasets (Generative models)

[9] V. Walter, M. Vrba, and M. Saska, "On training datasets for machine learning-based visual relative localization of micro-scale uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 674–10 680

Autonomous drone racing

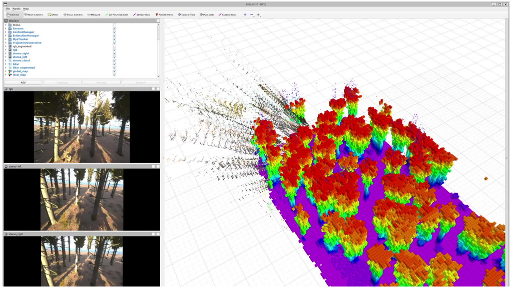
- real-time detection of *gates* under extreme motion
- feedback and feedforward control of a drone on the edge of the control envelope
- computer vision, reinforcement learning
- lucrative research field
  - ≈\$500,000 grant for fundamental research
  - (almost) winners of the Autonomous Racing League (2025)
- contact Dr. Robert Penicka if you are interested (`https://mrs.fel.cvut.cz/members/postdocs/penicka`)



Video: `https://youtu.be/6f062lZTTBA`

Autonomous semantic mapping and localization

- real-time processing of multi-modal data
- dense open-vocabulary object detection
- dense mapping of semantic data
- abstracting concepts into semantic graphs
- localization by matching semantic graphs with prior data
- challenge by SPRIN-D (`https://www.sprind.org/en/actions/challenges/funke-fully-autonomous-flight-2.0`)
- lucrative research field
  - ≈$500,000 grant for fundamental research
  - ≈$350,000 funding from SPRIN-D agency
  - cooperation with *Lockheed Martin*
- contact Dr. Tomas Baca if you are interested



Video: `https://youtu.be/JtxhlhZRs1A`

SPRIN-D challenge 2.0: Task query

`Land near a red car parked on a drive way of a house no. 5.`

Be ready to tackle problems that are seemingly unsolvable.

Quotation from a client (anonymized)

We would like your support in developing a drone system to help improve safety for surfers and swimmers. The idea is for the drone to patrol sea areas and **use computer vision to detect sharks**. If a shark is identified, the drone should send an alert signal to a wearable device, such as a smartwatch, through radio connection. Specifically, we would need your team to develop:

- The computer vision algorithm
- The alert signal transmission
- A patrolling function (ideally a circular pattern around the designated area)
- An automatic landing feature for low-battery situations.

Conclusion

Thanks for your attention

References I

[1] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", 2022.

[2] C. Huyen, *Designing machine learning systems.* " O'Reilly Media, Inc.", 2022.

[3] M. Kleppmann, *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems.* " O'Reilly Media, Inc.", 2017.

[4] C. Huyen, *AI Engineering: Building Applications with Foundation Models.* O'Reilly Media, Incorporated, 2024.

[5] N. Ravi, V. Gabeur, Y.-T. Hu, *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[6] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," *Advances in neural information processing systems*, vol. 26, 2013.

[7] M. Banko and E. Brill, "Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing," in *Proceedings of the first international conference on Human language technology research*, 2001.

[8] M. Vrba and M. Saska, "Marker-less micro aerial vehicle detection and localization using convolutional neural networks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2459–2466, 2020.

[9] V. Walter, M. Vrba, and M. Saska, "On training datasets for machine learning-based visual relative localization of micro-scale uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 674–10 680.