Robust Machine Learning for real-world challenges

Francesco Di Salvo, MSc.

Outline

- Introduction
- Agenda
- Expected outcome and grading
- Contacts
- Tools
- Project proposal

Introduction

Chair of Explainable Machine Learning (xAI)

What do we do?

Help building explainable, trustworthy, reliable AI that can help improve people's lives.

Why?

There is an overemphasis in the community on AI model development (accuracy) while critical areas for interpretability and trustworthiness lack understanding.

Chair of Explainable Machine Learning (xAI)

Areas of research

- Development of **robust AI algorithms** (classification, detection, segmentation)
- Translation of **explainable AI** into **impactful applications**. Focus on **Healthcare**.
- Data-driven analysis and **standardisation** of large and heterogeneous datasets with focus on **data-biases**.
- Development of statistical methods for **evaluating human-AI studies** and algorithms, sampling representative datasets, and the challenging definition of a ground truth.
- **Validation of AI systems** with focus on generalisability, data biases, hypothesis testing, human-AI interactions.

And you?

- Introduce yourself
  - Name
  - Academic background
  - Previous experience (if any) & interests in AI

Learning objective

- You will understand some of the current issues in Machine Learning
    - They mostly arise when we try to apply those models in real-word settings
- You will learn to formulate your own research questions
- You will learn how to properly design experiments and how to evaluate them
- You will learn to present and summarize your work

Expected outcome and grading

Expected outcome

- **Open source code**
  - Public repository hosted on GitHub
- **Final presentation**
  - One presentation per group
- **Final report**
  - One report per group

Requirements / Report

- 4-5 pages **per person** in the group, without references
  - Indicate who wrote which part of the report, everybody has to contribute
- Include link of the GitHub repository
- LaTeX template is in the VC course

Requirements / Report

- Possible structure
  1. Abstract (motivation, purpose, method, key findings, link to git repository)
  2. Introduction (motivation/background, related work, contribution)
  3. Methods (overview, model concept, details and purpose of contribution)
  4. Experiments & Results (used dataset, experimental setup, results)
  5. Discussion (discussion of results, positive & key findings, limitations)
  6. Conclusion (summary, outlook)

Requirements / Presentation

- Each group member should present ~10min
  - Three-people group: 30min
  - Two-people group: 20min
- ~10-15min questions (per group)

Requirements / Presentation

- Possible structure
  1. Introduction (motivation/background, related work, contribution)
  2. Methods (overview, model concept, details and purpose of contribution)
  3. Experiments & Results (used dataset, experimental setup, results)
  4. Discussion (discussion of results, positive & key findings, limitations)
  5. Conclusion (summary, outlook)

Requirements / GitHub Repository

- The code has to be reproducible
- The README has to be self-explanatory
  - Introduce and motivate your work
  - Explain how it works
    - Write a small tutorial
  - Contact information
    - Someone interested in your work might wanna reach out!

- Examples: link1, link2, link3, link4, link5

Grading

- Attendance of all meetings and presentations is mandatory
- All team members have to contribute to coding, presenting & writing
- Grades
  - 50% based on final presentation
  - 50% based on scientific report
- Every participant is graded individually

Grading / Presentation

- Formalities
  - Timing
  - Appropriate references
- Expertise/Content
  - Structure of content
  - Appropriate scope and complexity
- Presentation technique
  - Layout of slides
  - Quality of media
  - Key info on slides
  - Avoidable technical problems
  - Appropriate use of technical terms
  - Educational for audience
  - Knowledgeable Q&A
  - Posture/eye contact
  - Clear communication/voice
  - Respectful/non-defensive

Grading / Report

- Formalities
  - Appropriate length
  - Appropriate references
- Content
  - Abstract and introduction
  - Main body
- Presentation
  - Logic/structure
  - Appropriate use of tech. terms
  - Key info present
  - Scientific writing style
  - Discussion and summary
  - Technical understanding
  - Educational and clear to follow
  - Quality of layout and graphics
  - Readability and language

Contacts

Contacts & Office hours

- Available via email at francesco.di-salvo@uni-bamberg.de

- Available in WE5/04.091 every Wednesday from 10.00 to 14.00
  - *Please send me an email in advance in order to schedule individual meetings*

- Forum of the VC Course [link]

Tools

Tools

- Git and GitHub
- Python and Pytorch
- Google Colab
- Latex

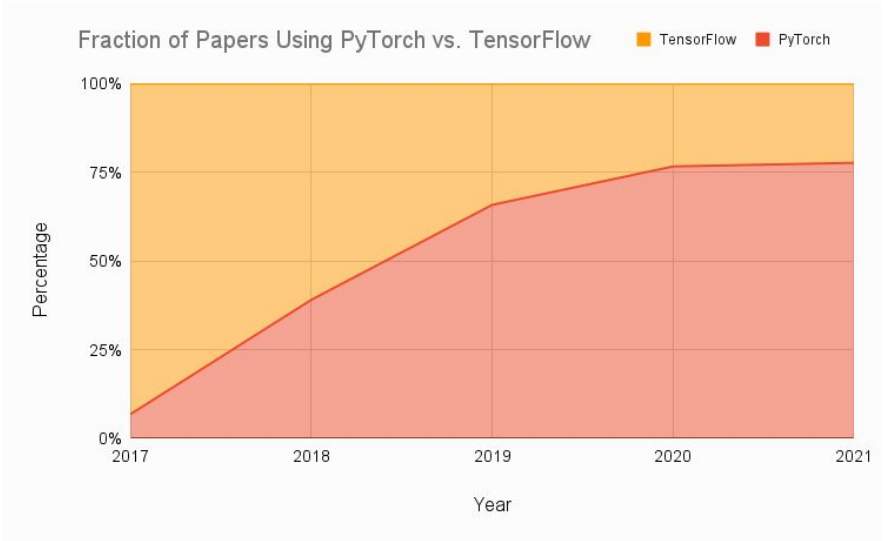
Git and GitHub

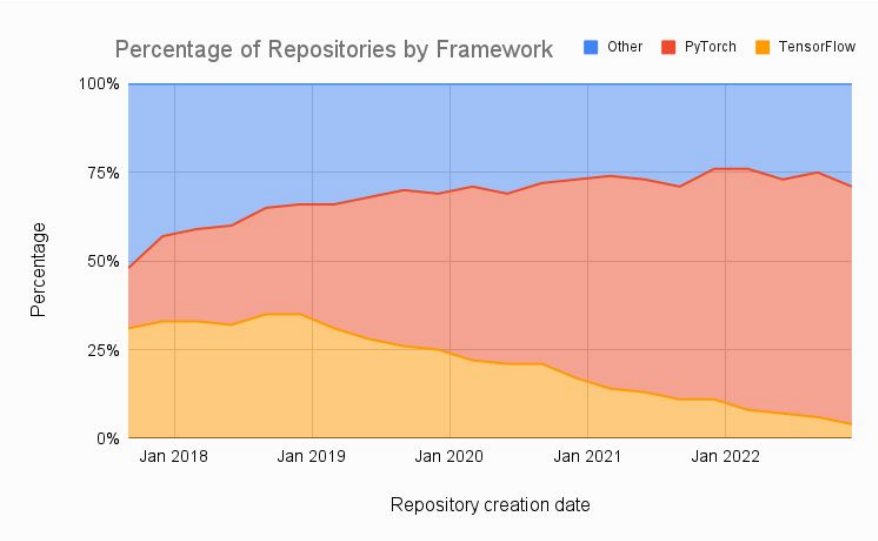
- Git
  - A free and open-source distributed version control system
  - Used to track changes in files, collaborate on projects, and revert to previous versions
  - Highly scalable and efficient, making it ideal for projects of all sizes
  - Popular among developers, but can be used by anyone who needs to track changes in files
- GitHub
  - GitHub is a web-based hosting service for software development projects that use Git

Git and GitHub

- Resources
  - https://www.w3schools.com/git/
  - https://www.freecodecamp.org/news/introduction-to-git-and-github/

Python and Pytorch





Pytorch Tutorial [YT] [GitHub]

https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/

Google Colab

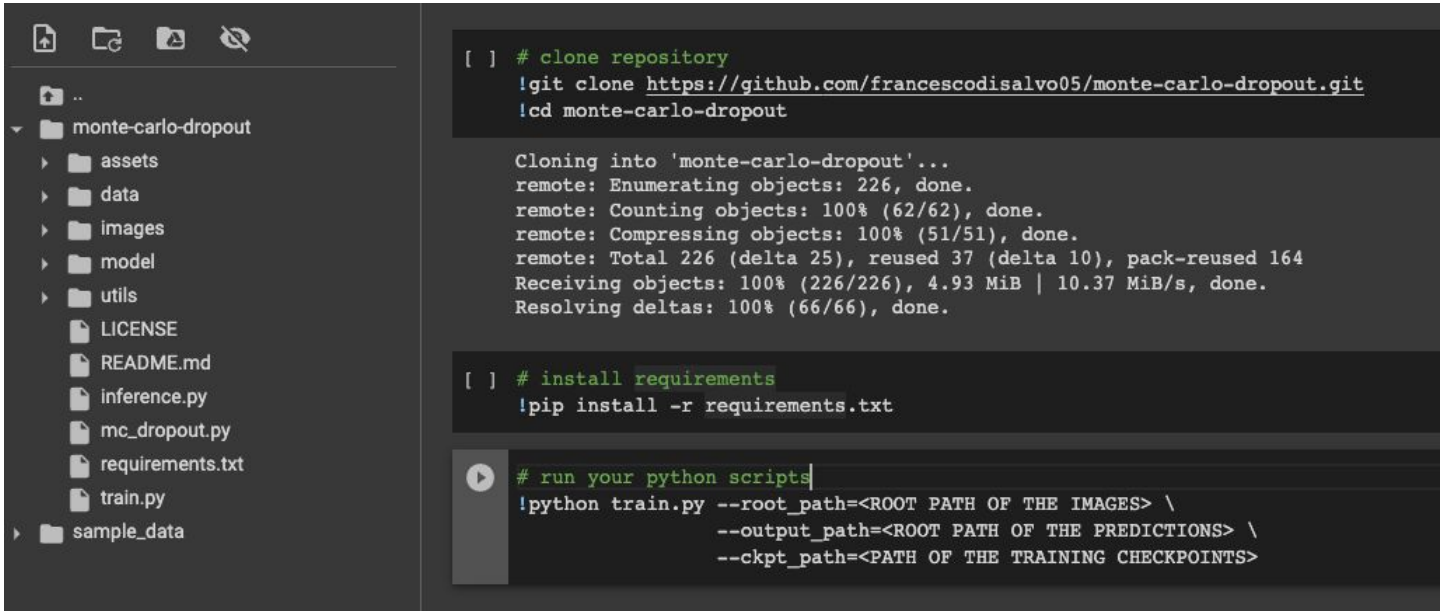
- Pros
  - Free GPU access
  - No setup required (no need to install any software or package)
  - Collaborative editing
  - You can use Jupyter Notebooks
- Cons
  - Limited storage space
  - Not persistent storage
    - Protip: copy data from Google Drive (avoid re-uploading multiple times!)
  - Session limits (up to 12 hours)

Google Colab

- Resources
  - Tutorial - Google Colab
  - PyTorch Tutorial on Google Colab
- Tips
  - How to use Google Colab with GitHub
  - Connect Google Drive to Google Colab
    - Useful for managing data, logs, and results

Google Colab

- Suggested workflow to use Colab only for the GPU



LaTeX

- "LaTeX is a tool for typesetting professional-looking documents" [src]
- It produces documents that are (hopefully) both beautiful and functional
- Free online compiler: https://www.overleaf.com/

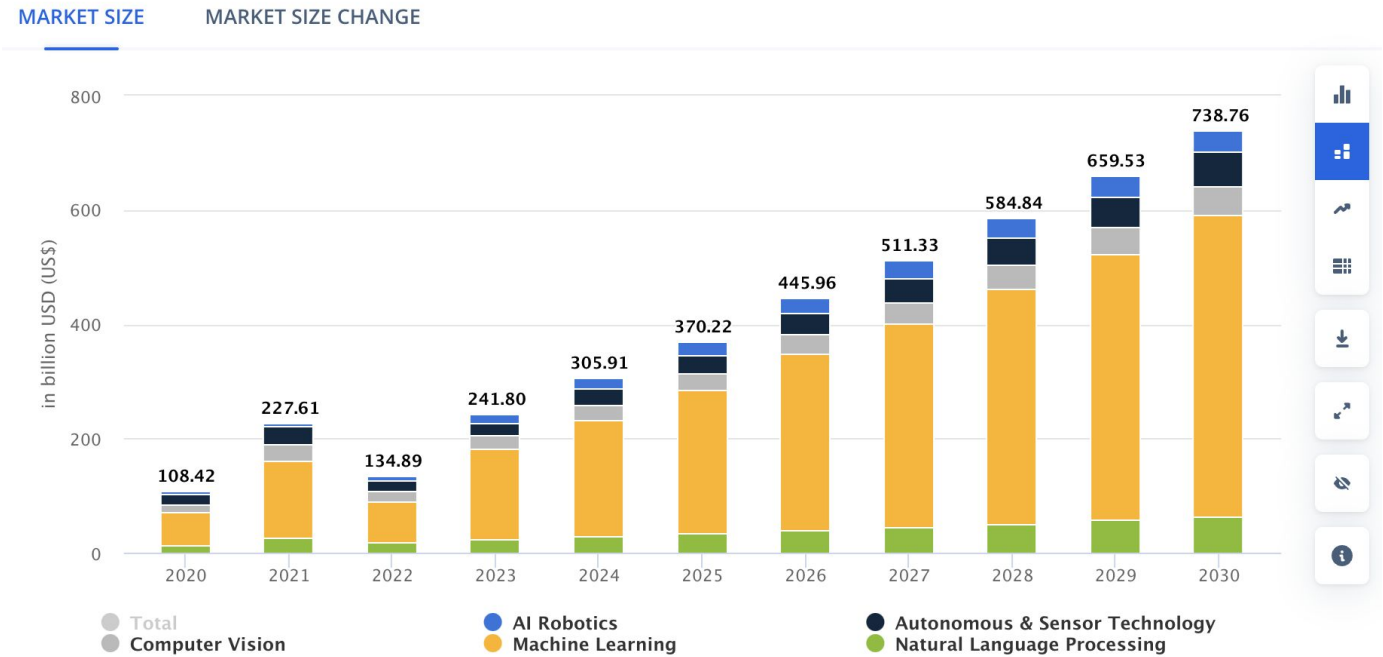
- Resources
  - https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
  - https://latex-tutorial.com/tutorials/

Background

Why are we here?

https://www.statista.com/outlook/tmo/artificial-intelligence/worldwide#market-size

Background



Notes: Data shown is using current exchange rates and reflects market impacts of the Russia-Ukraine war.

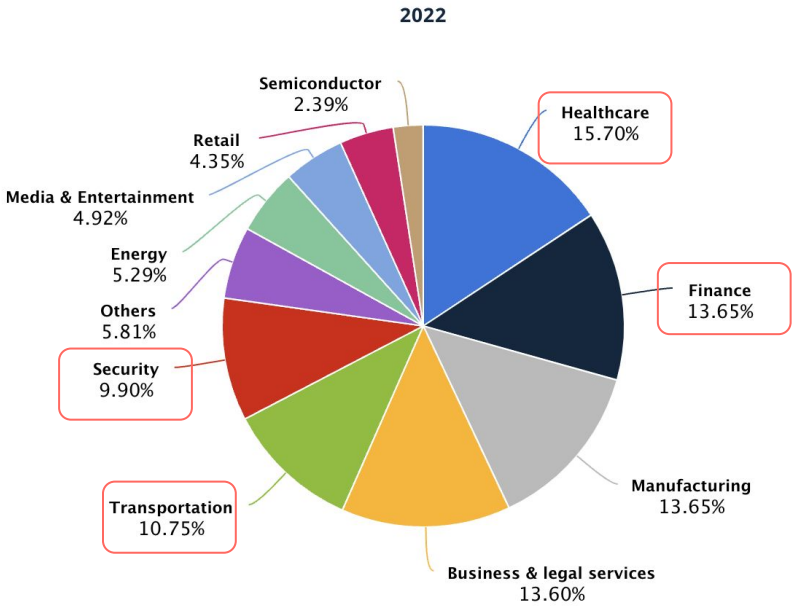
Most recent update: Aug 2023

https://www.statista.com/outlook/tmo/artificial-intelligence/worldwide#market-size

Background

MARKET SIZE SHARE BY INDUSTRY

in percent



Most recent update: Aug 2023

(Some) Critical challenges of traditional DL method

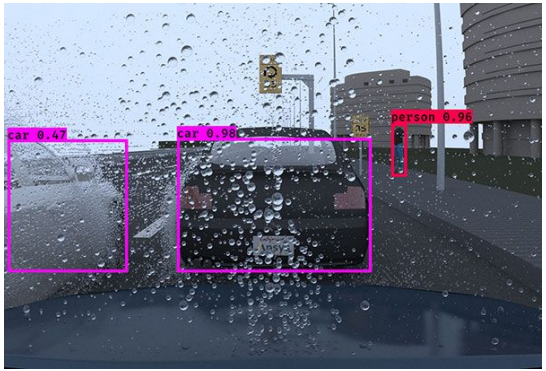
- Robustness toward image corruptions
- Robustness toward out-of-distribution data
- Explainability
- Model calibration
- Model- and data-efficiency

Robustness toward image corruptions

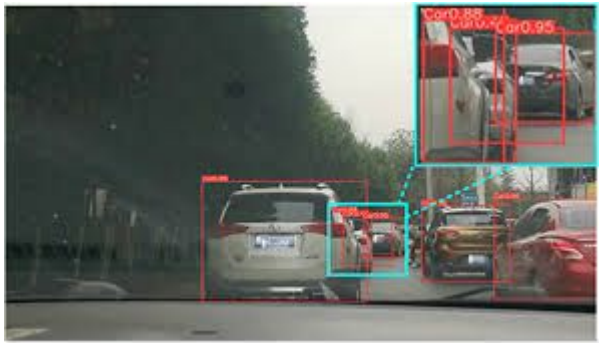
[source]

Robustness toward image corruptions

[source]

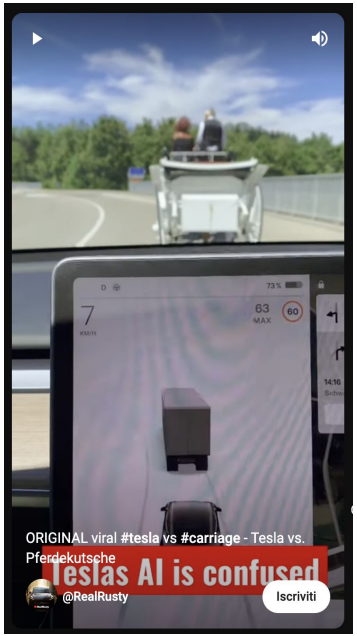


[source]



[source]

Robustness toward out of distribution



[source]

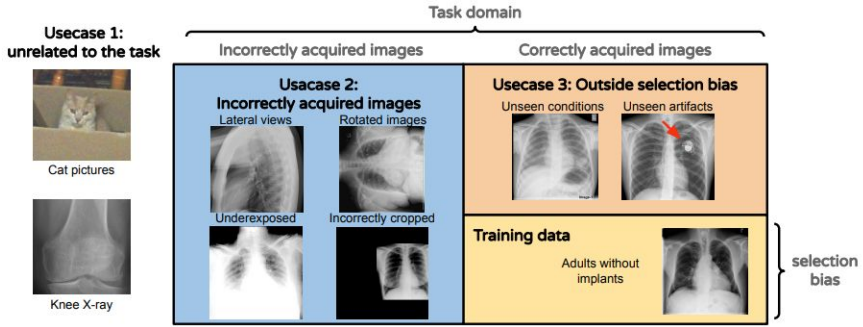
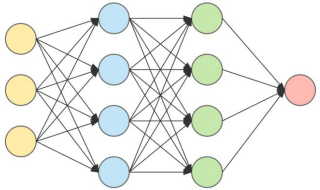


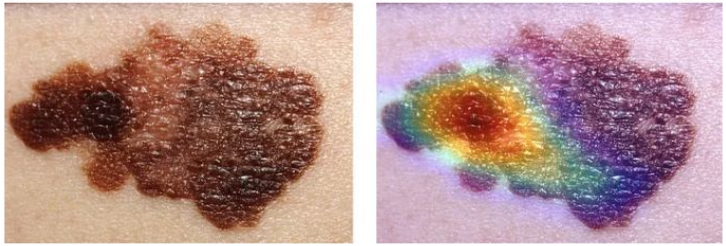
Figure 1: The three use-cases shown in relation to each other. The training data is sampled iid from the *In* data distribution. 1. Inputs that are unrelated to the task. 2. Inputs which are incorrectly prepared 3. Inputs that are unseen due to a selection bias in the training distribution.

[source]

Explainability



BENIGN / **MALIGNANT**



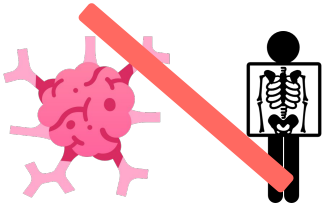
[source]

Also..

Carbon footprint

Requires Money

Requires a lot of data



Requires high computational resources

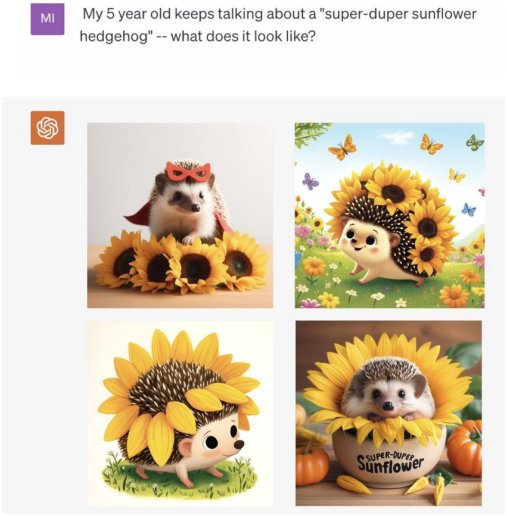
Some (!) of the causes

- Over-parametrization and subsequent high sensitivity
  - .. thus low generalization, high training cost, and high carbon footprint
- Task and dataset dependency
  - .. thus low generalization
- Possibly complex architectural design
  - .. thus it is difficult to understand what's really happening under the hood

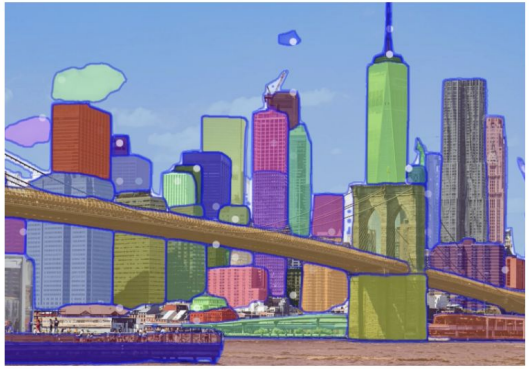
So what?

Foundation models

[ChatGPT]



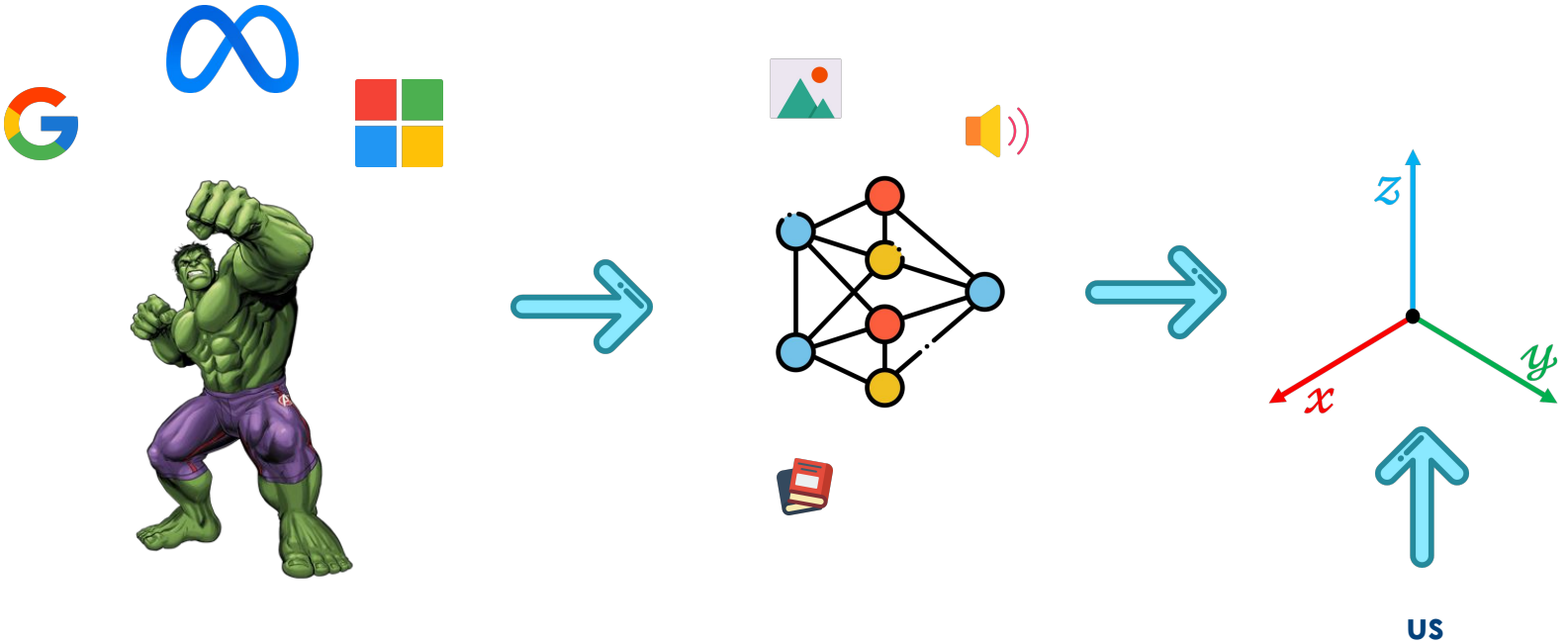
[DALL-E 3]



Automatically segment everything in an image.

[Segment Anything (SAM)]

Foundation models



What is an embedding?

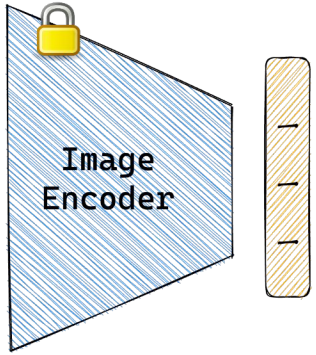


Image
Encoder

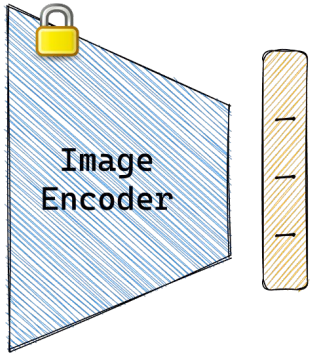
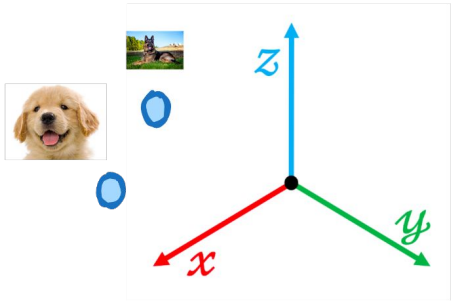


Image
Encoder


z
y
x

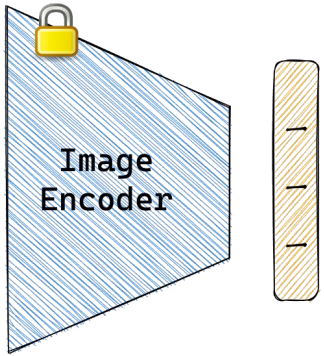
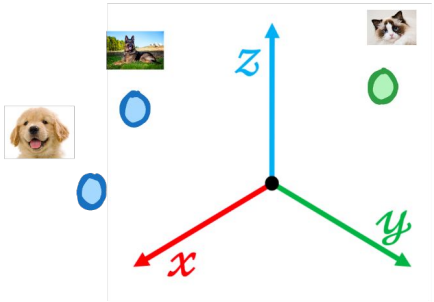
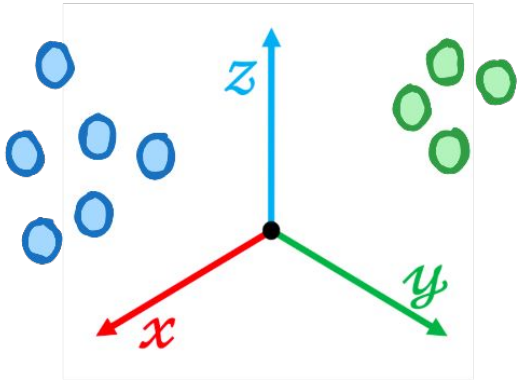


Image
Encoder


z
y
x

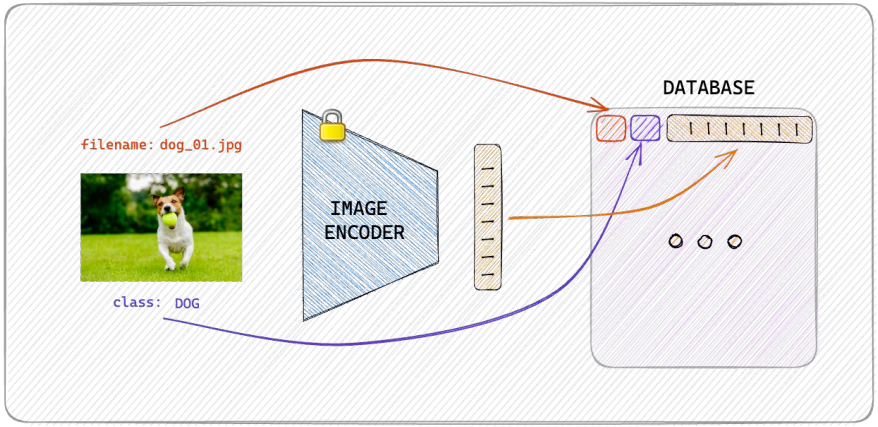

z
y
x

To summarize

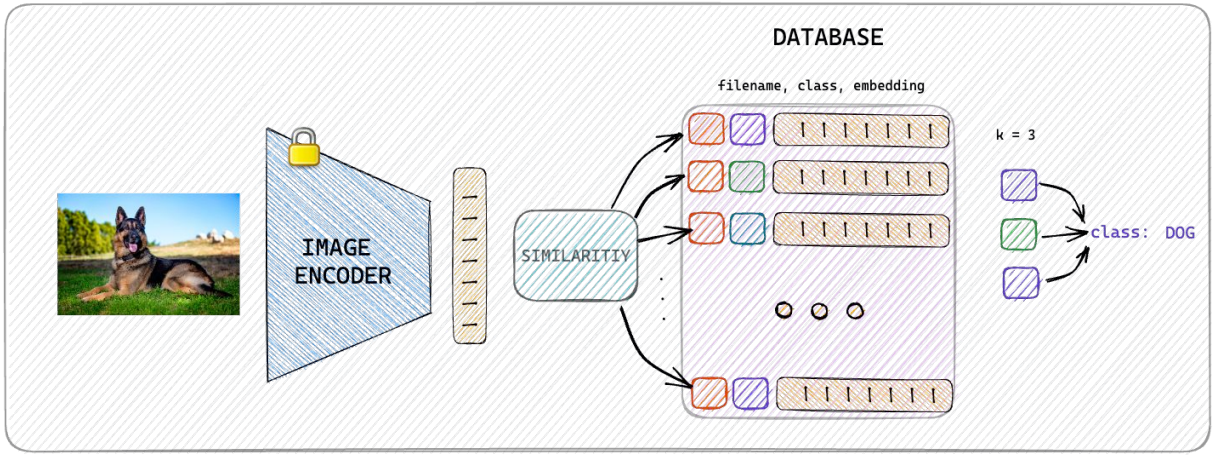
- An image embedding is a numerical representation of the image, encoded into a lower-dimensional vector representation
- It describes its location in a d-dimensional feature space
- Thanks to Foundation Models pre-training, images of the same class will be more likely to be close, whereas images of different classes will be far away

Let’s go to our project!

"Train"



Inference



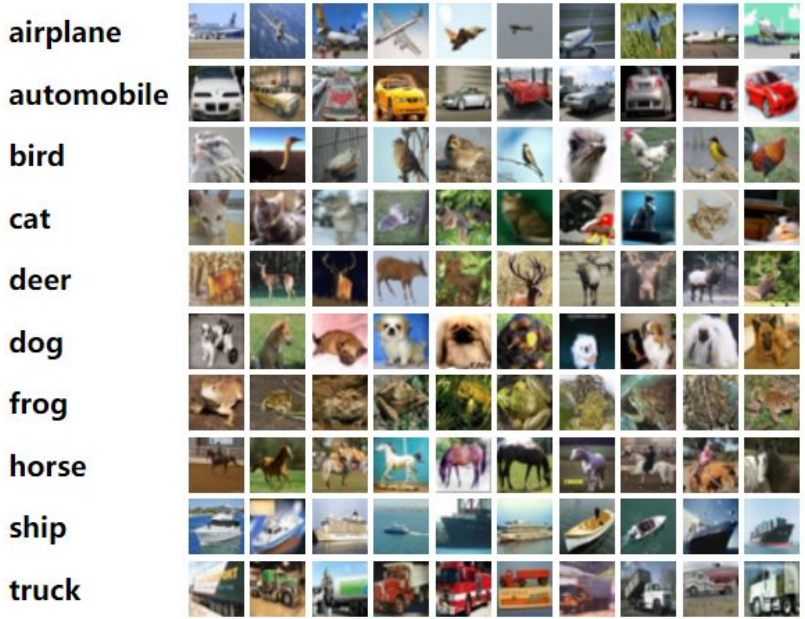
Project overview / Step 1

1. **Familiarize with vector-databases**
   a. They will be provided. Link in the VC course.
2. **Explore the latent space**
   a. Explore data distribution (e.g., through tSNE plots)
   b. How closest points look like? Plot the images!
      i. Is there any class-overlap in the neighborhood?
         e.g., a `dog` has three closest neighbours with class `cat`
   c. How further away points look like?
   d. Just try things out and have fun!

Project overview / Step 1

- **CIFAR10**
  - 60000 images 32x32
  - Divided in 10 classes

- **CIFAR100**
  - 60000 images 32x32
  - Divided in 100 classes



Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

Project overview / Step 1

- **DermaMNIST**
  - 10015 images 224x224
  - Divided in 7 classes

- **BreastMNIST**
  - 780 images 224x224
  - Divided in 2 classes

Yang, Jiancheng, et al. (2023) "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification."

Project overview / Step 1

DEMO!

Project overview / Step 2

1. **Performance evaluation**
    a. Simple k-Nearest Neighbour
        i. Implement your own kNN in PyTorch (to enable CUDA support)
        ii. How to determine *k? Does it work equally good/bad for all datasets?*
    b. Train a Linear Layer on top (known as Linear Probing)
        i. Find suitable hyperparameters for each dataset
2. **Summarize the results**

Alignment lecture

- Wrap up your results
- Feedback, ideas and brainstorming

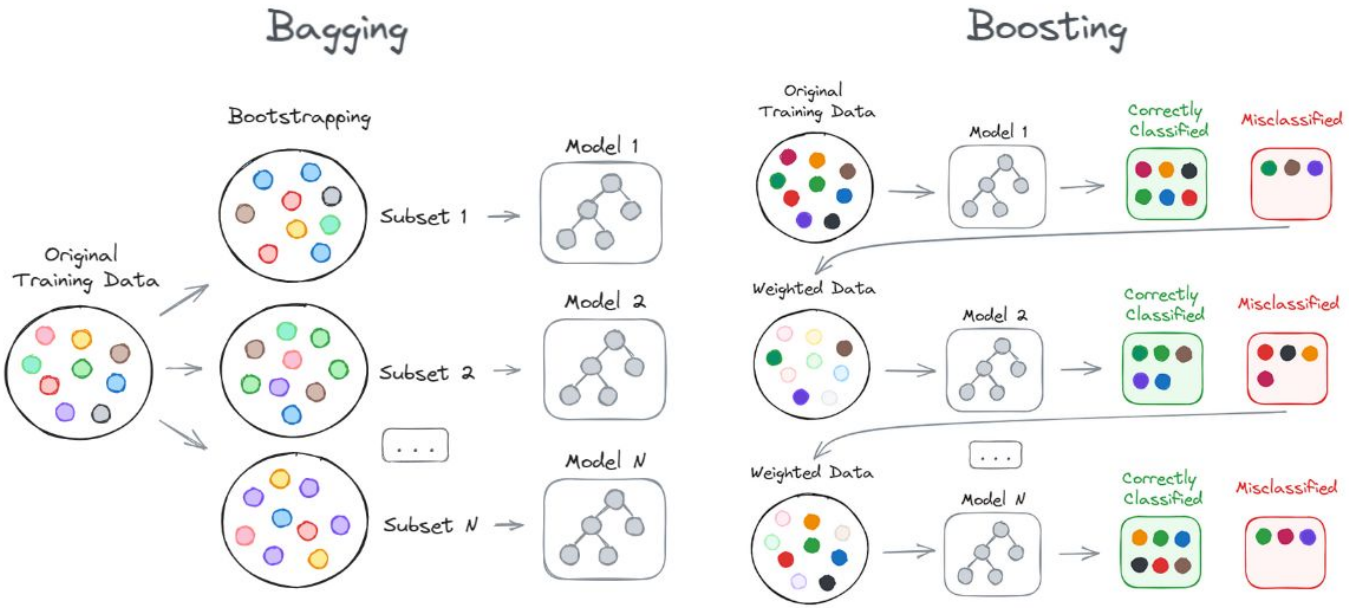
- Also, lecture about “How to”
  - Write a report
  - Make a scientific presentation
  - Organize your repository

Project overview / Step 3

Let’s improve kNN!

We’ll try to improve kNN performance using traditional ML concepts

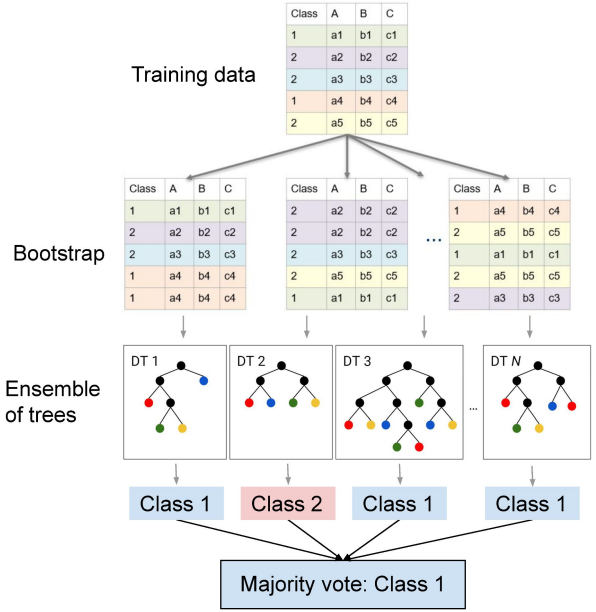
Project overview / Step 3 / Preliminaries



[source]

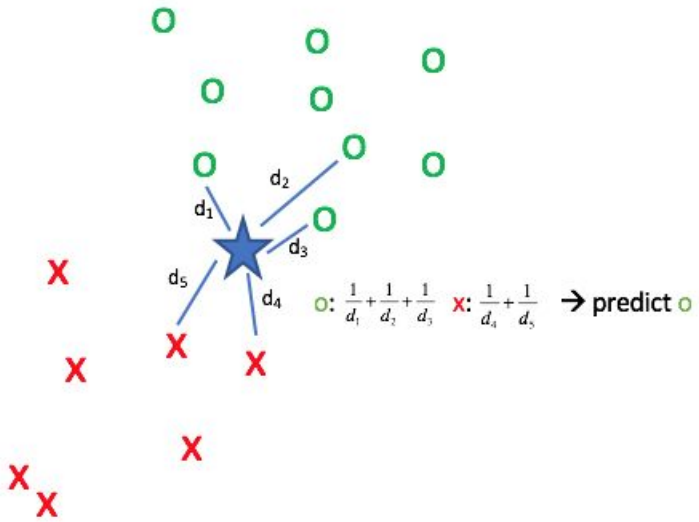
Project overview / Step 3 / **Option 1: Forest kNN**

- Translate the idea of RandomForest to kNN
  - Ensemble of kNN
  - Random subsets of samples
  - Random subset of features
    - .. different random projections?
  - Majority vote across weak classifiers



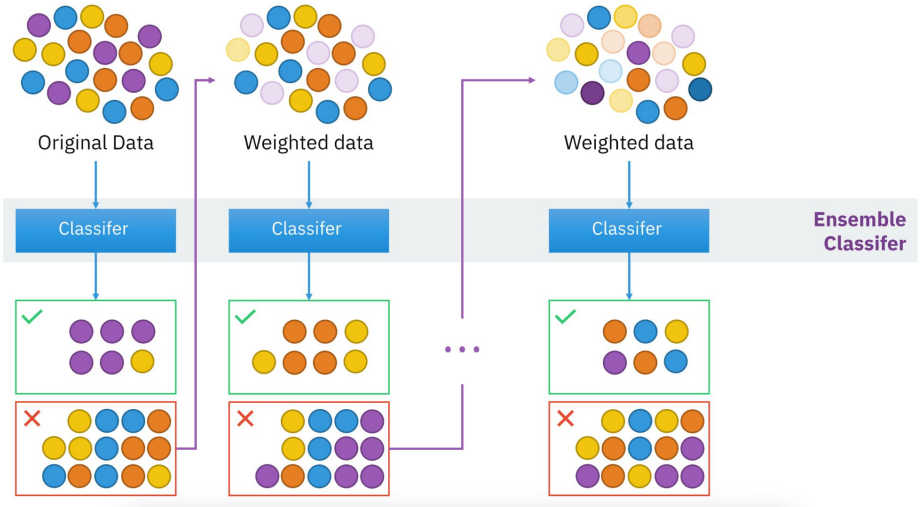
[source]

Project overview / Step 3 / **Option 2: Boost-WkNN**



Weighted kNN

[source]

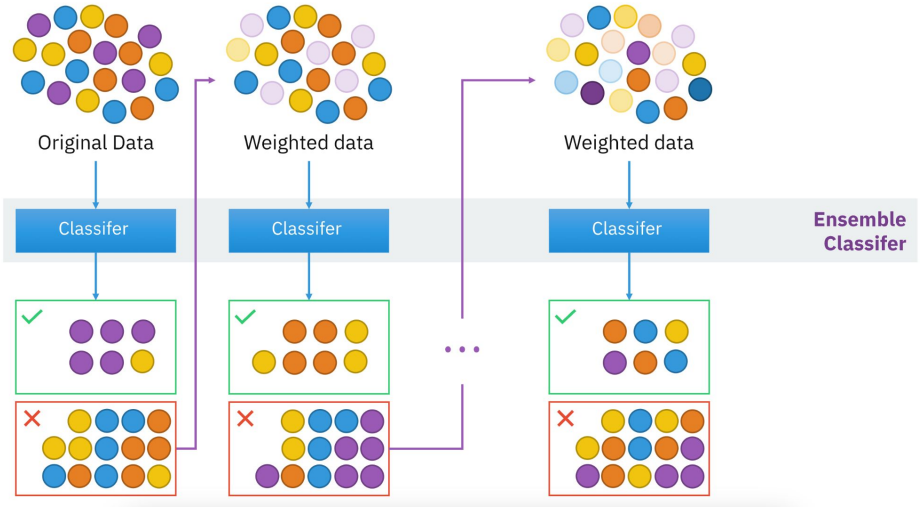


Boosting

[source]

Project overview / Step 3 / **Option 2: Boost-WkNN**

- Translate the idea of Boosting to Weighted-kNN
  - How to determine the weights?
  - Should a sample be simply removed or just re-weighted?
  - How to correctly aggregate the predictions? Weighted voting scheme or majority?



[source]

Project overview / Step 3

1. **Each group** will select and work on **one** of the proposed topics
   under the guidance of the instructor
   a. Familiarize with the resources of the assigned topic
      i. Some papers will be given, but you will also make your own literature search
   b. Implement the new kNN method within your codebase
   c. Compare the performance with the previously obtained results
      i. Why is it better or worse?
   d. Draw your conclusions!

… and that's it!

Agenda

24/04/2024	**[WE5/01.006]** Introduction | Project proposal | Group formation
30/04/2024	**[Deadline 30/04 23:59]** Email with group name, group members, and topics (order or preference)
-	Work on Step 1 and Step 2 | Literature review on the chosen topic | Set goals for Step 3
29/05/2024	**[WE5/01.006]** How to present your work (presentations, reports, and GitHub) [2hrs]
05/06/2024	**[WE5/04.091]** Mandatory group meeting with instructor | Updates
10/06/2024	**[Deadline 10/06 23:59]** Email with a two-pages summary (not graded!) Show your results from Step 1-2 and state your goals for Step 3
12/06/2024	**[WE5/01.006]** Alignment lecture | Brainstorming [2hrs]
-	Work on Step 3
24/07/2024	**[WE5/01.006]** Final presentations **[To be confirmed!]**
31/07/2024	**[Deadline: 02/08 23:59]** Submit final reports via email **[To be confirmed!]**

Make a group

Talk at the end of the lesson

Suggested literature

kNN and Foundation Models

- Nakata, K., Ng, Y., Miyashita, D., Maki, A., Lin, Y. C., & Deguchi, J. (2022). Revisiting a knn-based image classification system with high-capacity storage. In European Conference on Computer Vision (pp. 457-474). Cham: Springer Nature Switzerland.
- Doerrich, S., Archut, T., Di Salvo, F., & Ledig, C. (2024). Integrating kNN with Foundation Models for Adaptable and Privacy-Aware Image Classification. *21st IEEE International Symposium on Biomedical Imaging*.
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193.

Suggested literature

Option 1 (kNN-Forest)

- Biau, Gérard, and Erwan Scornet. "A random forest guided tour." Test 25 (2016): 197-227. [link]
- Li, Shengqiao, E. James Harner, and Donald A. Adjeroh. "Random KNN feature selection-a fast and stable alternative to Random Forests." BMC bioinformatics 12 (2011): 1-11. [link]
- Devi, R. Gayathri, and P. Sumanjani. "Improved classification techniques by combining KNN and Random Forest with Naive Bayesian classifier." 2015 IEEE international conference on engineering and technology (ICETECH). IEEE, 2015. [link]
- *Bonus: Statquest - Random Forest* [link1, link2]

Option 2 (Boost-WkNN)

- Yigit, Halil. "A weighting approach for KNN classifier." 2013 international conference on electronics, computer and computation (ICECCO). IEEE, 2013. [link]
- Dudani, Sahibsingh A. "The distance-weighted k-nearest-neighbor rule." IEEE Transactions on Systems, Man, and Cybernetics 4 (1976): 325-327. [link]
- García-Pedrajas, Nicolás, and Domingo Ortiz-Boyer. "Boosting k-nearest neighbor classifier by means of input space projection." Expert Systems with Applications 36.7 (2009): 10570-10582. [link]
- Piro, Paolo, et al. "Leveraging k-NN for generic classification boosting." Neurocomputing 80 (2012): 3-9. [link]
- *Bonus: Statquest - AdaBoost* [link1]

Suggested resources

- Git & GitHub
  - https://www.w3schools.com/git/
  - https://www.freecodecamp.org/news/introduction-to-git-and-github/
- Colab
  - Tutorial - Google Colab
  - PyTorch Tutorial on Google Colab
  - How to use Google Colab with GitHub
  - Connect Google Drive to Google Colab
- Pytorch Tutorial [YT] [GitHub]
- Latex
  - https://www.overleaf.com/ [compiler]
  - https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
  - https://latex-tutorial.com/tutorials/