An Introduction to Machine Learning

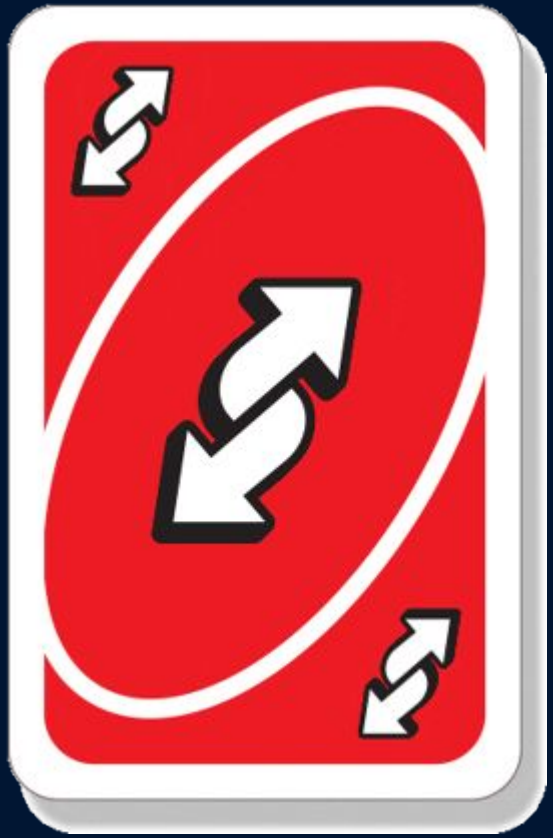
These slides may or may not been prepared last-minute.

Hi, I’m Vijay.

Hackerabadi;
Tech Lead@GDSC-IIITB;
Dev@Zense;
Pun Master@Blahajgang

Before we go any further...

What do *you* think Machine Learning is all about?



Those are some great insights!

So, what actually is *ML*?

What is AI? Are they the same?

Is it a scam?

What is life?

Let’s start with the most pertinent question.

What is life?

PTSD?

- How would you approach this problem?
- Are there solid and defined rules?

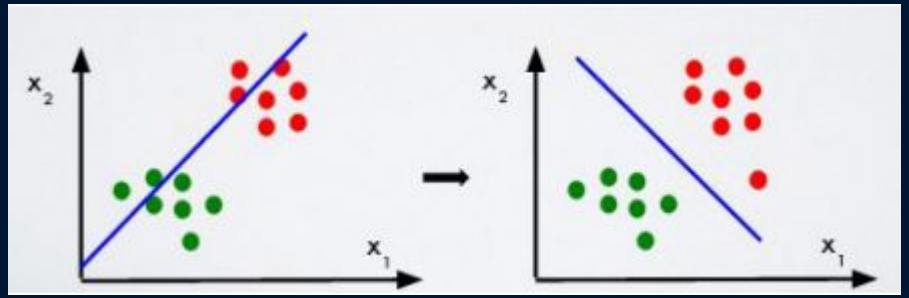
Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

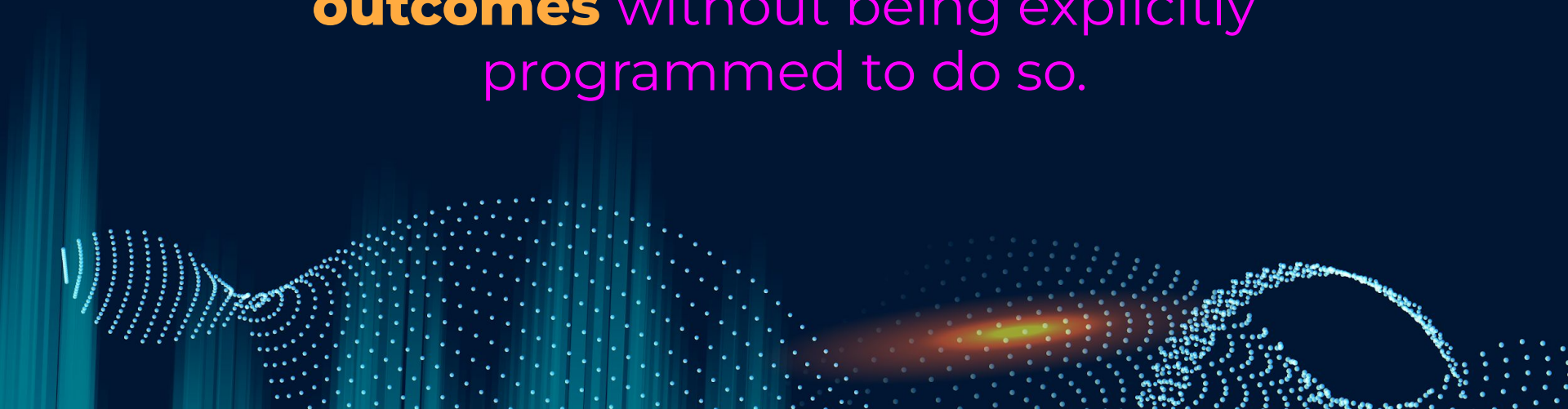
Extrapolate those ideas.

- What rules can we define for *this* problem?
- Where do we stop
- Test cases



What if someone or something can analyse this data and devise the rules for separation?

ML consists of a set of algorithms that allow software applications to become more accurate at predicting outcomes without being explicitly programmed to do so.



ML はどこで見ることができますか**?**

Hint: Google Lens!

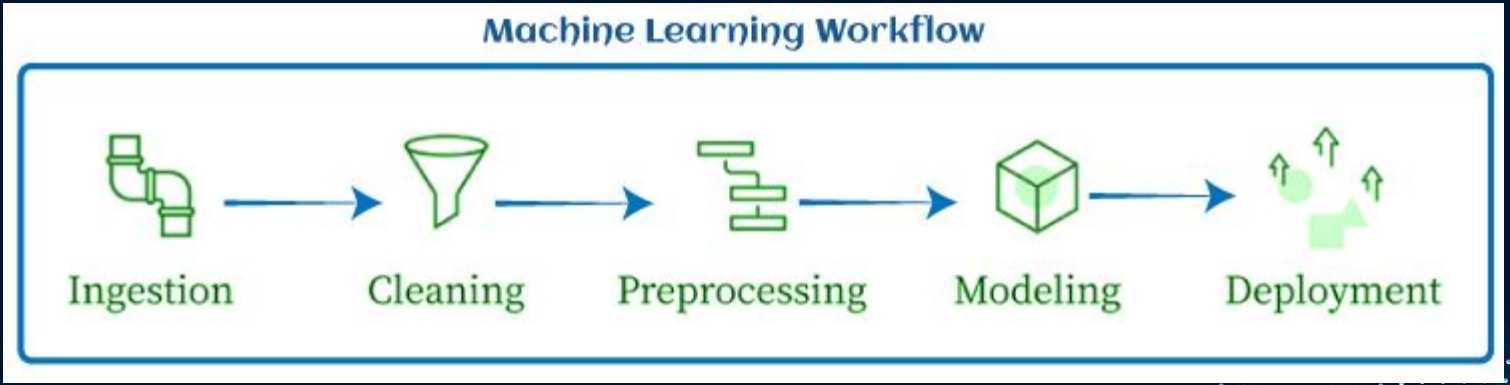
ML is used everywhere*

* pretty much

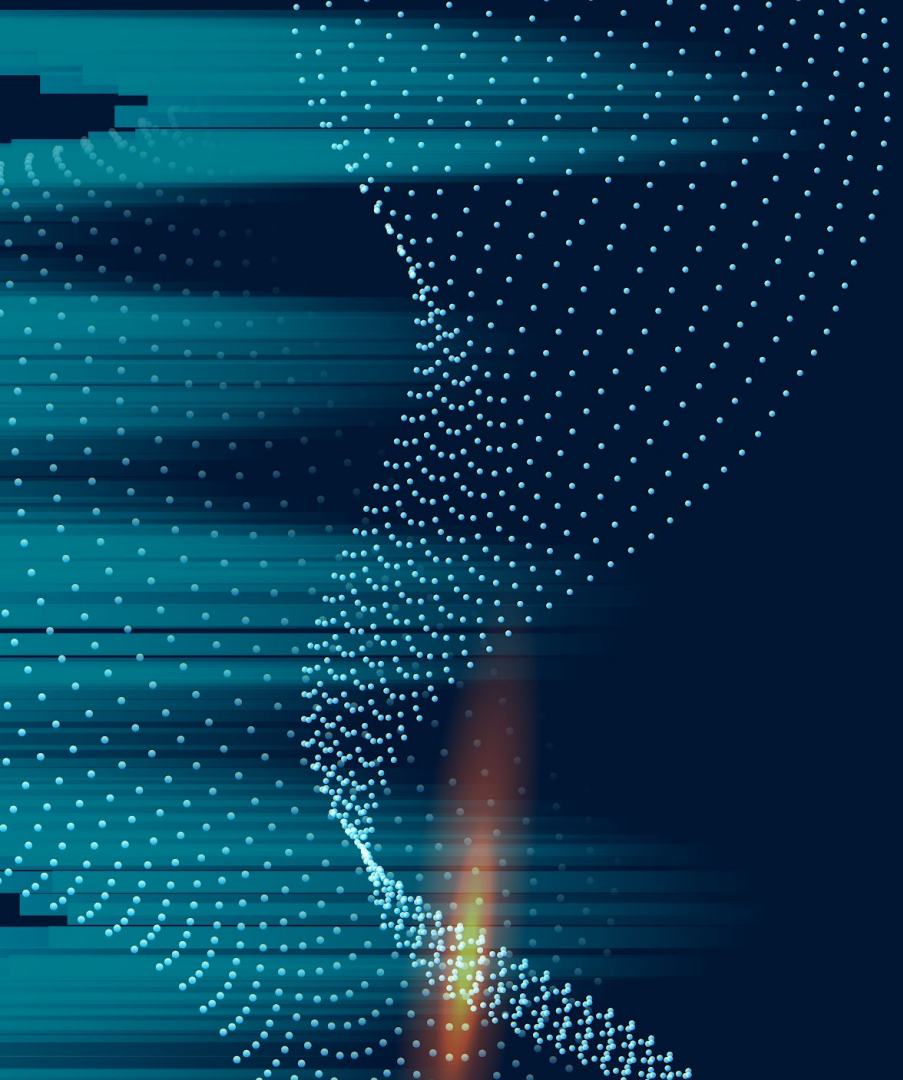
Basic Components

- Tasks
  - Problems you wish to apply ML techniques on; clear declaration and definition of inputs and outputs
- Models
  - Algorithms run on data that generate insights
- Features
  - Features and processed inputs
- Datasets
  - “Raw” data

ML Pipeline



Source:
https://www.javatpoint.com/machine-learning-pipeline



Classification

Categorise a set of (labelled) data into a set of classes

Regression

Predict a continuous value from given data pairs

Clustering

Grouping unlabelled examples into bins

SHEESH.

That’s a lot of theory.

Let’s do something fun!

Have you

Heard of Blahaj?

Do you know

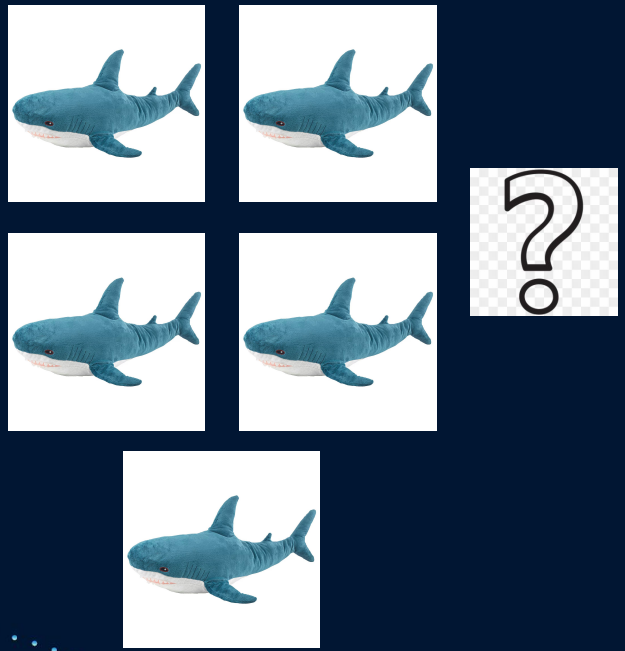
Why this animal is (in?)famous in the MLH circles?

Let’s make a classifier to classify Blahajs from Corgis!

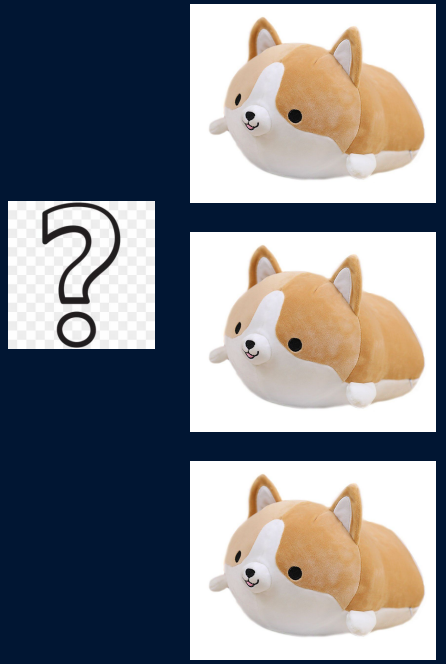
Recall: What is a classifier?

Any volunteers?

What object do you think the question mark is?



What object do you think the question mark is?



What object do you think the question mark is?

?

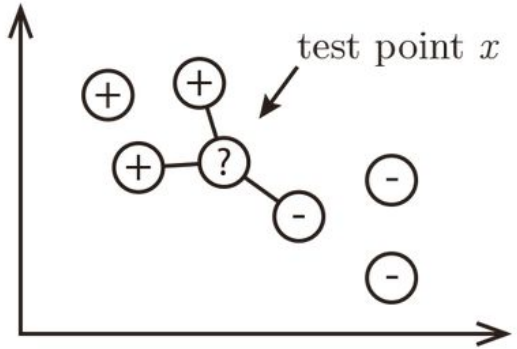
Congratulations!

You have discovered the KNN algorithm!

KNN Algorithm

Source: https://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote02_kNN.html

Assumption: Similar Inputs have similar outputs
Classification rule: For a test input x, assign the most common label amongst its k most similar training inputs



Neighbors' labels are $2\times\oplus$ and $1\times\ominus$ and the result is $\oplus$.

Formal (and borderline incomprehensible) definition of k-NN:
Test point: $\mathbf{x}$
Define the set of the k nearest neighbors of $\mathbf{x}$ as $S_\mathbf{x}$. Formally $S_\mathbf{x}$ is defined as $S_\mathbf{x} \subseteq D$ s.t. $|S_\mathbf{x}| = k$ and $\forall(\mathbf{x}', y') \in D \backslash S_\mathbf{x}$,

$$\text{dist}(\mathbf{x}, \mathbf{x}') \geq \max_{(\mathbf{x}'', y'') \in S_\mathbf{x}} \text{dist}(\mathbf{x}, \mathbf{x}''),$$

(i.e. every point in D but *not* in $S_\mathbf{x}$ is at least as far away from $\mathbf{x}$ as the furthest point in $S_\mathbf{x}$). We can then define the classifier $h()$ as a function returning the most common label in $S_\mathbf{x}$:

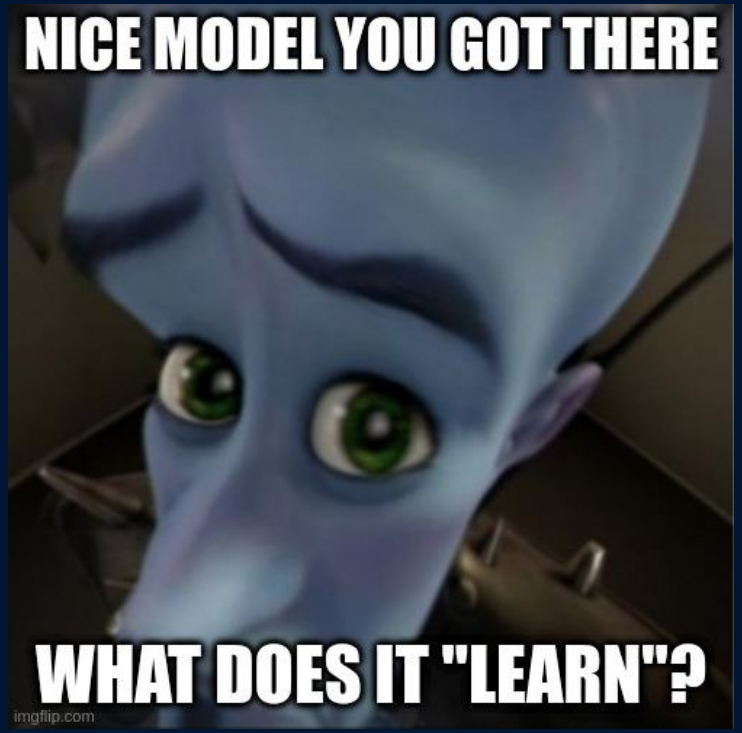
$$h(\mathbf{x}) = \text{mode}(\{y'' : (\mathbf{x}'', y'') \in S_\mathbf{x}\}),$$

where $\text{mode}(\cdot)$ means to select the label of the highest occurrence.
(Hint: In case of a draw, a good solution is to return the result of k-NN with smaller k)

In English, please!

- Find the k nearest neighbours of the target point given the input data
- Assign the target point based on the majority class of the nearest neighbours
- Tie-breakers based on nearest distance

Megamind’s got a point!



Recap: Classification

- Labelled data - features and models
- **Training phase:** Learning classification rules/ decision surfaces
- **Testing phase:** Checking the “accuracy” of the model using (previously) unseen data

What’s on the menu today?

Some code?

Make your copy!

shorturl.at/nJQSV

Some things to keep in mind

- Overfitting and underfitting
- Performance and processing time
- Data bias and size

FAQs.

Let’s ask ChatGPT!

What is the difference between Machine Learning, Data Science, and Artificial Intelligence?

Machine Learning, Data Science, and Artificial Intelligence are related but distinct fields of study. Here is a brief overview of each one:

1. Machine Learning (ML): Machine Learning is a subfield of Artificial Intelligence that focuses on creating systems that can learn from data and improve their performance on a specific task over time. Machine Learning algorithms are designed to identify patterns in data, make predictions, or take actions based on input data.
2. Data Science: Data Science is an interdisciplinary field that combines statistical analysis, machine learning, and domain expertise to extract insights and knowledge from data. Data Science includes a wide range of techniques for collecting, cleaning, processing, and analyzing data, as well as visualizing and communicating insights.
3. Artificial Intelligence (AI): Artificial Intelligence is a broad field that encompasses many subfields, including Machine Learning, Natural Language Processing, Computer Vision, Robotics, and more. AI refers to the development of systems that can perform tasks that typically require human intelligence, such as learning, reasoning, perception, and decision making.

In summary, Machine Learning is a subset of Artificial Intelligence that focuses on building systems that can learn from data, while Data Science is an interdisciplinary field that includes Machine Learning as one of its core components. Artificial Intelligence, on the other hand, is a much broader field that encompasses many different techniques and subfields, including Machine Learning.

What mathematics should I learn to get started in Machine Learning?

To get started in Machine Learning, you will need to have a solid understanding of linear algebra, calculus, probability, and statistics. These mathematical concepts are used extensively in developing machine learning models and understanding how they work.

Here is a brief overview of what you should focus on:

1. Linear Algebra: Linear algebra is an essential mathematical tool in machine learning, as it deals with vectors, matrices, and linear transformations. It provides a way to represent and manipulate data in a more efficient and compact way.
2. Calculus: Calculus is used to optimize machine learning algorithms and is particularly important in the training of neural networks. Understanding concepts such as derivatives, integrals, and optimization is crucial.
3. Probability: Probability theory is used to model uncertain events and measure the likelihood of their occurrence. It is essential in understanding statistical models and Bayesian networks.
4. Statistics: Statistics is used to analyze data, identify patterns, and make predictions. It is essential in evaluating machine learning models and understanding their performance.

In summary, having a strong foundation in linear algebra, calculus, probability, and statistics is critical in understanding and developing machine learning models.

How to contribute to open-sourced machine learning projects?

Contributing to open-source machine learning projects is a great way to learn, gain experience, and give back to the community. Here are some steps you can follow to contribute to open-sourced machine learning projects:

1. Identify a project that interests you: Look for open-source machine learning projects that align with your interests and skill level. You can find such projects on platforms like GitHub, GitLab, or Bitbucket.
2. Explore the project: Get familiar with the project by reading its documentation, issues, and codebase. This will help you identify areas where you can contribute.
3. Look for open issues: Look for open issues that you can work on. These issues could range from documentation to bug fixes to new features.
4. Discuss with the community: Once you identify an issue, discuss it with the community on the project's forum or chat channels. This will help you understand the issue better, get feedback on your ideas, and avoid duplicating efforts.
5. Fork the project: Fork the project's repository, create a branch, and start working on the issue.
6. Make changes and test: Make changes to the code and test them thoroughly. Make sure that your changes align with the project's guidelines and standards.
7. Submit a pull request: Once you are confident with your changes, submit a pull request to the project. This will trigger a review process where other contributors will review your changes and provide feedback.
8. Address feedback: Address the feedback from the project's contributors and make necessary changes to your code.
9. Get your changes merged: Once your changes pass the review process, they will be merged into the project's codebase.

Contributing to open-source machine learning projects can be a fulfilling experience. It allows you to work with experienced developers, learn new skills, and contribute to the community.

“No, but seriously though.

Where can I learn more?”

Resources

- GDSC-IIITB's ML study jams: https://github.com/GDSC-IIITB/ML-Study-Jams-2022

- Curated list: https://github.com/vijay-jaisankar/ML_TA_IIITB_2022/blob/main/RESOURCES.md
    - Find a nice resource? Raise a PR!

Thank you!

Any questions?