

Climate Change AI Virtual Summer School

Introduction to AI

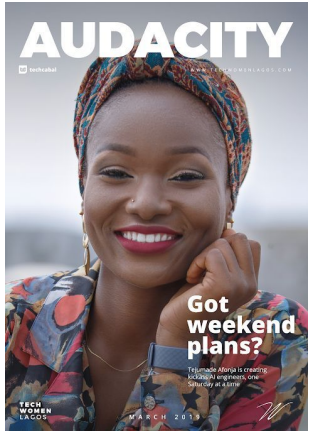
Tejumade Afonja

Slides adapted from AI Climate Institute Pilot Workshop

Disclaimer

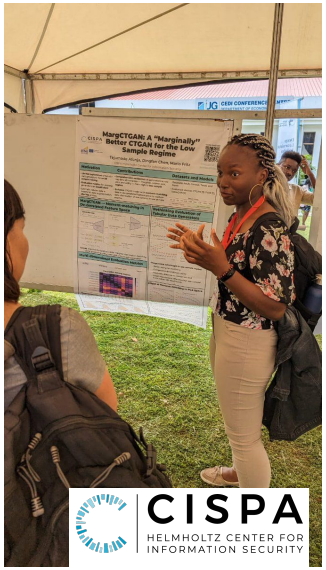
- This slide is adapted from the AI Climate Institute Pilot Workshop.
- I have modified the slides slightly to include some classwork - but not much.
- I think the slide deck is excellent and chose to use the material. Please always cite/reference the original slides.
- I am a ML researcher but not a climate researcher (yet), so most of the examples used that are climate related are not my expertise, but the core ML fundamentals are my forte.

About Me





















- Educator & Researcher
  - Co-founder, TRI AI & AI Saturdays Lagos (now TRI AI Saturdays)
  - Doctoral Researcher, CISPA Helmholtz Center for Information Security
- General Chair, Deep Learning Indaba 2025, Ex Student Researcher, Google DeepMind, 2022 DSSG Fellow

Lecture Outline

What is Machine Learning?

Types of Machine Learning

Data

Q&A

Understanding Linear Regression Algorithm

Lecture Outline

What is Machine Learning?

Types of Machine Learning

Data

Q&A

Understanding Linear Regression Algorithm

ML vs. other approaches

Strengths & limitations

What is AI?

Artificial intelligence (AI): Any algorithm that allows a machine to perform a complex task (e.g., tasks associated with human intelligence)

AI represents the developments of computer systems capable of emulating human intelligence. These systems can perform tasks that typically require human cognitive abilities, such as problem solving, learning from data, and decision making.

https://runsystem.net/en/news/learn-about-ai-terminology-2024-ani-agi-and-asi

What is AI?

Artificial intelligence (AI): Any algorithm that allows a machine to perform a complex task (e.g., tasks associated with human intelligence)

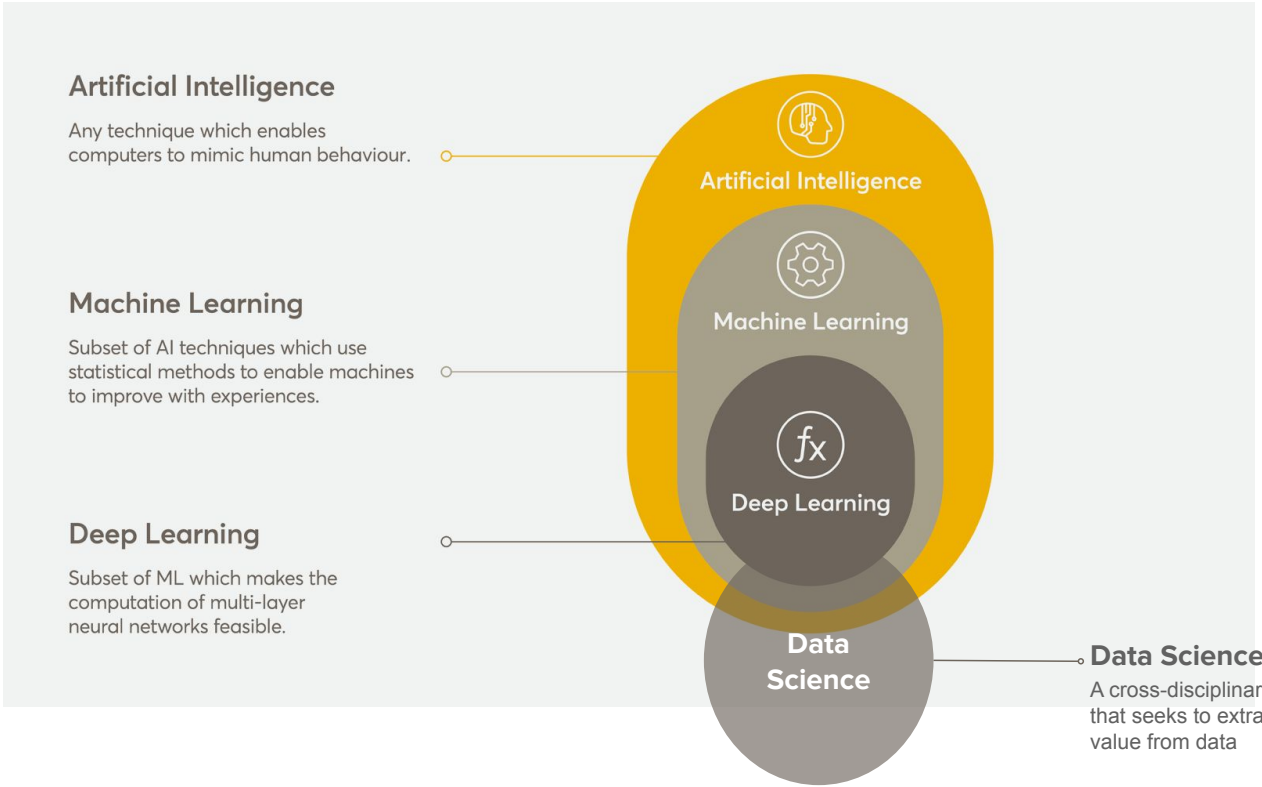
Rule-based systems:
Techniques that automatically reason over an explicit set of rules



Machine learning (ML):
Techniques that automatically extract patterns from data



Automatic Extraction of Patterns from Data



https://en.idei.club/27516-iot-artificial-intelligence.html

Class Work: Rule-based Systems



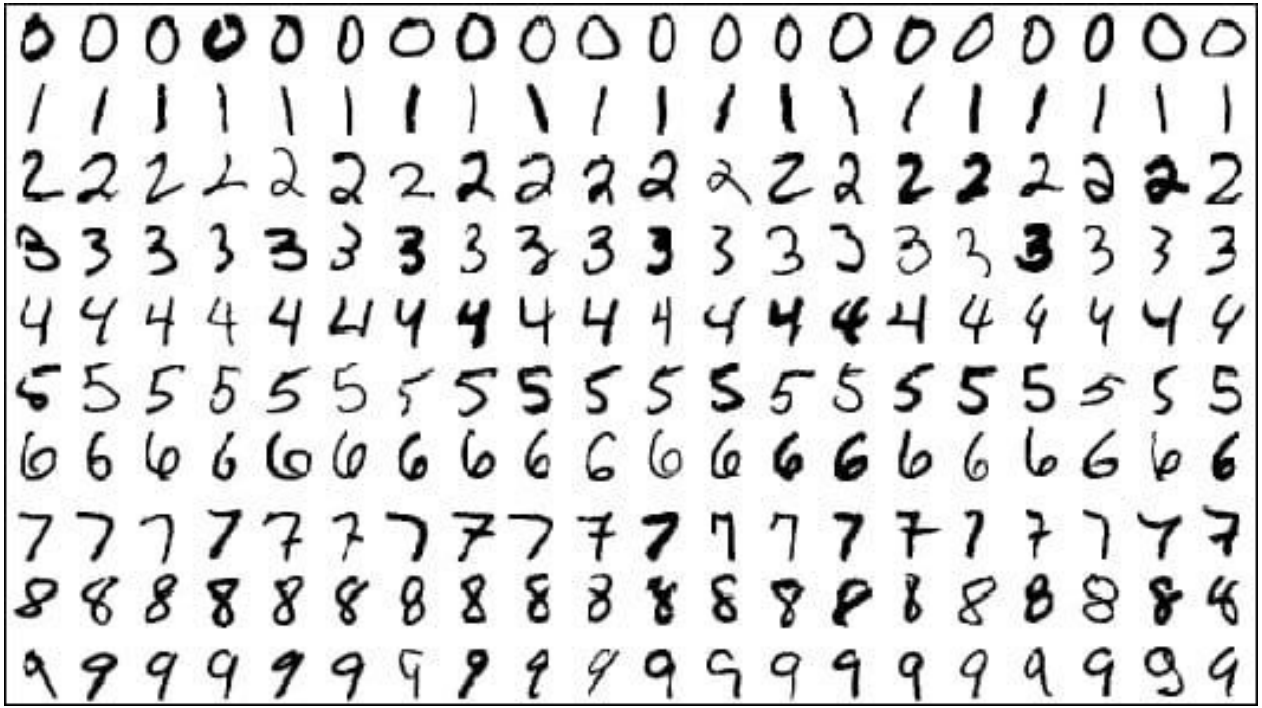
Write a simple program to recognise the digit 9 from image

Class Work: Rule-based Systems



does your program still work? if not, think about modifying it

Machine Learning: Learning General Patterns from Data



What is Machine Learning (ML)?

“Machine learning is a field of study that gives computer the ability to learn without being explicitly programmed ”

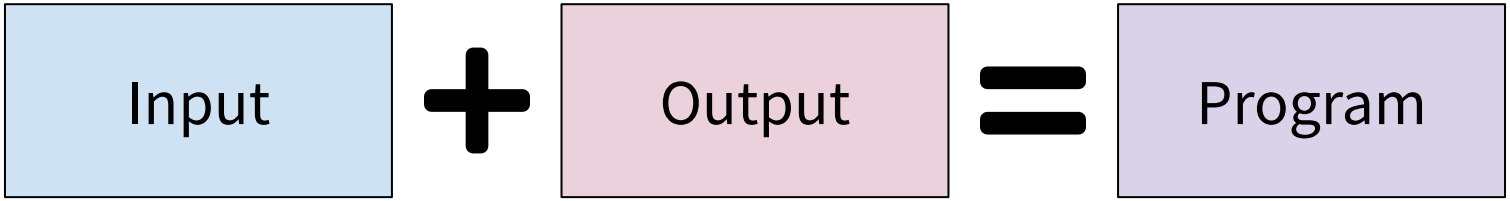
— Arthur Samuel (1959)

Traditional Programming vs. ML

Traditional Programming



Machine Learning (ML)



Some Strengths & Limitations of ML

STRENGTHS

- Leveraging patterns in data
- Performing tasks at scale
- Optimising complex systems
- Ability to be “always on”
- Combining heterogeneous inputs
- Integrating with other methods

LIMITATIONS

- It’s a tool – not the right fit for every task
- Can be (overconfidently) incorrect
- Can be data- and/or compute-hungry
- “Garbage in, garbage out”
- Inherits biases in data/design/use
- Assumes rules/patterns are persistent

When Should You Use ML?

- Not always: Not all problems can or should be solved with machine learning!
  - Use ML in situations where it makes more sense to infer the rules from examples, rather than writing the rules down
- When you have access to data which “matches” the task
  - Appropriate resolution, coverage, representative sample of a distribution, etc.
- When estimates are useful and errors are allowable
  - “*All Models Are Wrong, Some are Useful*” – George Box
- When it is a cost-effective solution
  - Costs include (i) dataset creation and processing and (ii) model development, deployment, and maintenance

Machine Learning

Machine learning (ML): Automatically extract patterns from data

MODELS

Neural networks (deep learning)
Support vector machines
Decision trees
Bayesian models
…

DATA / USE CASE TYPES

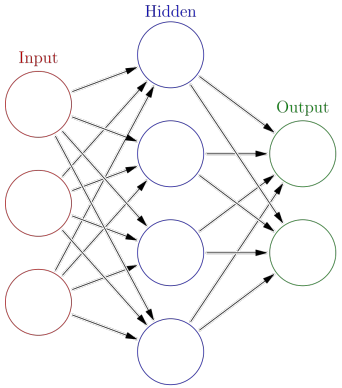
Computer vision
Natural language processing
Robotics
Signal processing
…

PARADIGMS

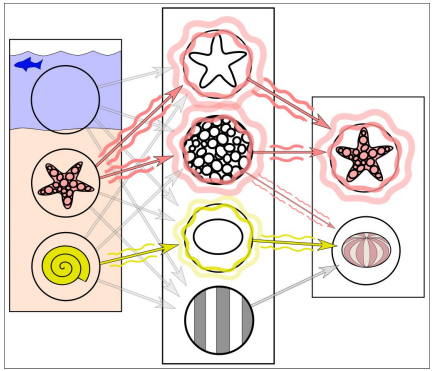
Generative vs discriminative models
Physics-informed & robust
Interpretable
Task specific vs general-purpose models
…

What is an ML Model?

The **model** consists of parameters and a structure that together transform inputs into outputs



The “learning” part means these parameters get adjusted through **training** to produce a “learned model” that can achieve a task



“Large” vs. “Small” Models

- **Wide spectrum** of models in terms of number of parameters
- Small models can run in seconds on most laptops
- Larger models can require using a server (e.g., one powerful GPU running for half a day)
- Very large models can require 10,000+ GPUs running for several months

```
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier

# Load data
X, y = load_iris(return_X_y=True)

# Create, train, and predict
model = RandomForestClassifier()
model.fit(X, y)

predictions = model.predict(X[:5])
print(predictions)
```

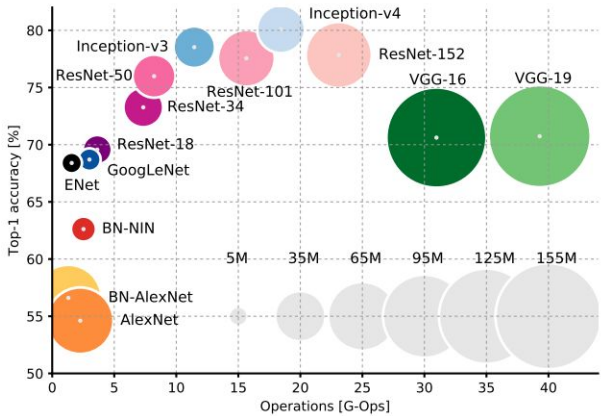
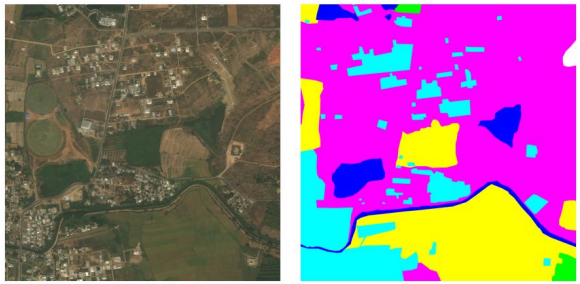


Image: https://arxiv.org/pdf/1605.07678

Task-Specific vs. General-Purpose Models

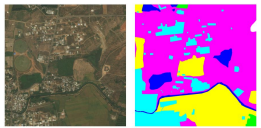
Task-specific model

- Trained only for one task
- Often trained with one data modality
- Example: Segmenting by land use type in satellite imagery

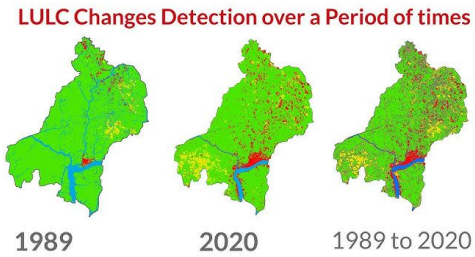


General-purpose models

- Aims to handle many different types of tasks across various domains
- Trained on broad, diverse data
- Example: Geospatial foundation models for segmenting by land use type, land use change detection, etc.

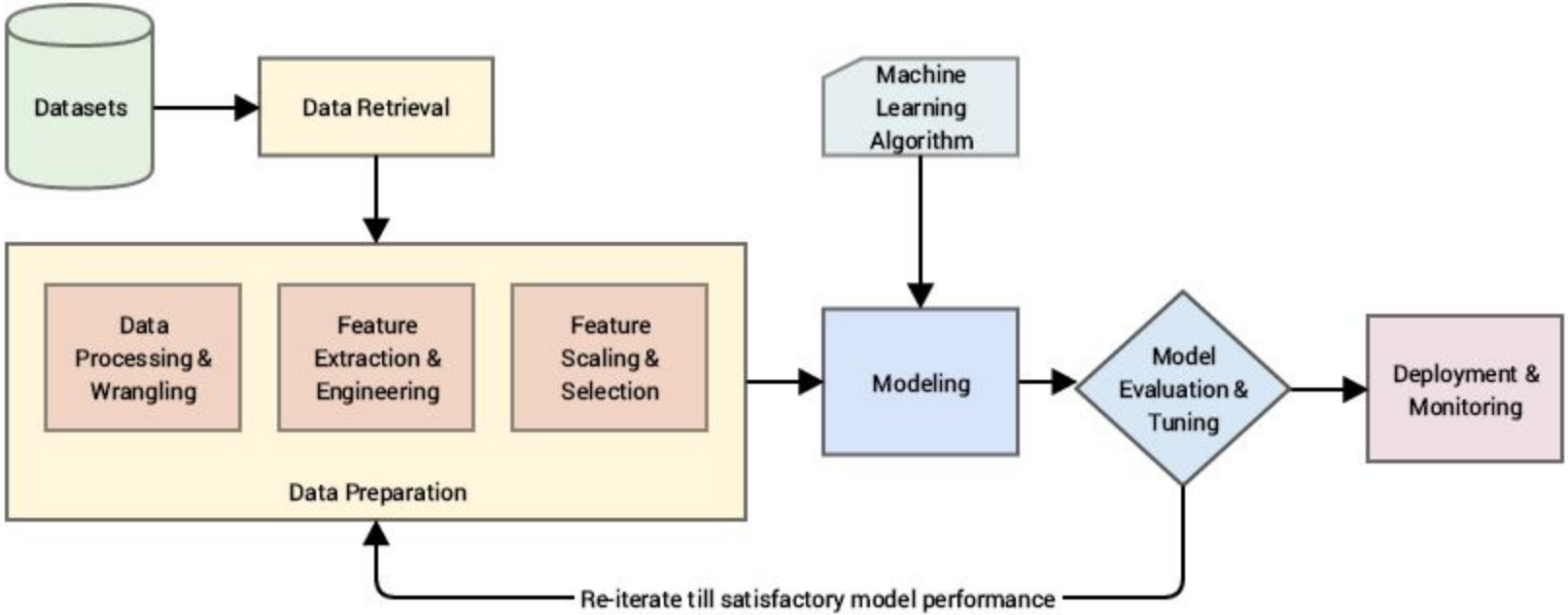


+



+ ...

ML In Practice



Practical Machine Learning with Python. Bali, Sarkar, and Sharma.

Development → training → inference

Development

Setting up the data pipeline infrastructure

Deciding how to evaluate model performance

Testing different algorithms and hyperparameters

Choosing which model will be used for final training

Training

“Fitting” model parameters to improve performance on task

Choose training data to be representative of inference setting(s)

Can take minutes to months

Needs to be done again if data distribution evolves

Inference

Producing predictions for “unseen” data inputs

Uses the “frozen” model parameters from training

Assumes “unseen” inputs have similar distribution to training data

Typically runs much faster than training (milliseconds to seconds)

Class Work: ML Development Workflow

Development

Setting up the data pipeline infrastructure

Deciding how to evaluate model performance

Testing different algorithms and hyperparameters

Choosing which model will be used for final training

Which of these steps do you think takes the most time in a real ML project?

Lecture Outline

What is Machine Learning?

Types of Machine Learning

Data

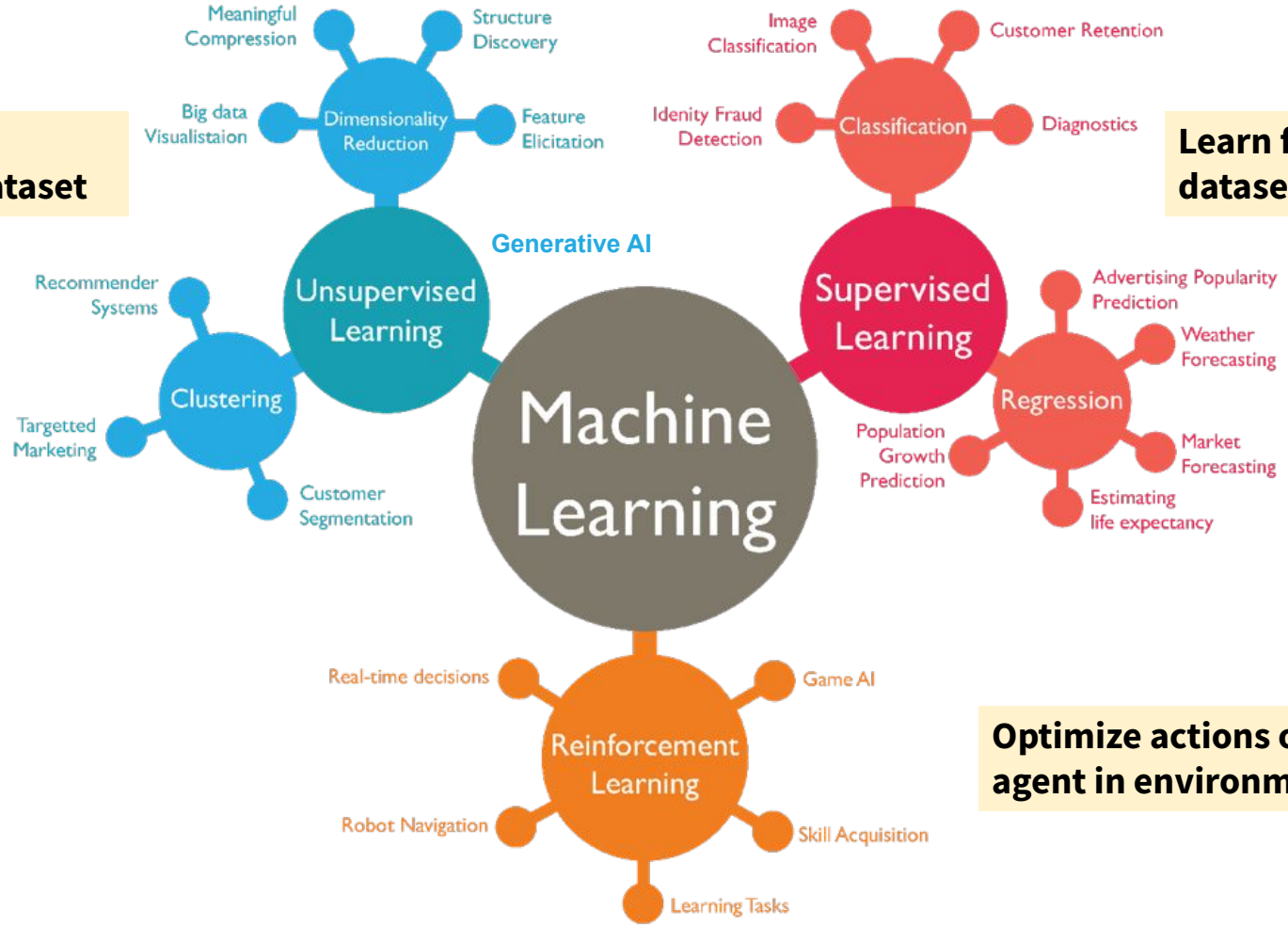
Q&A

Understanding Linear Regression Algorithm

Main paradigms

Climate-relevant ML subfields

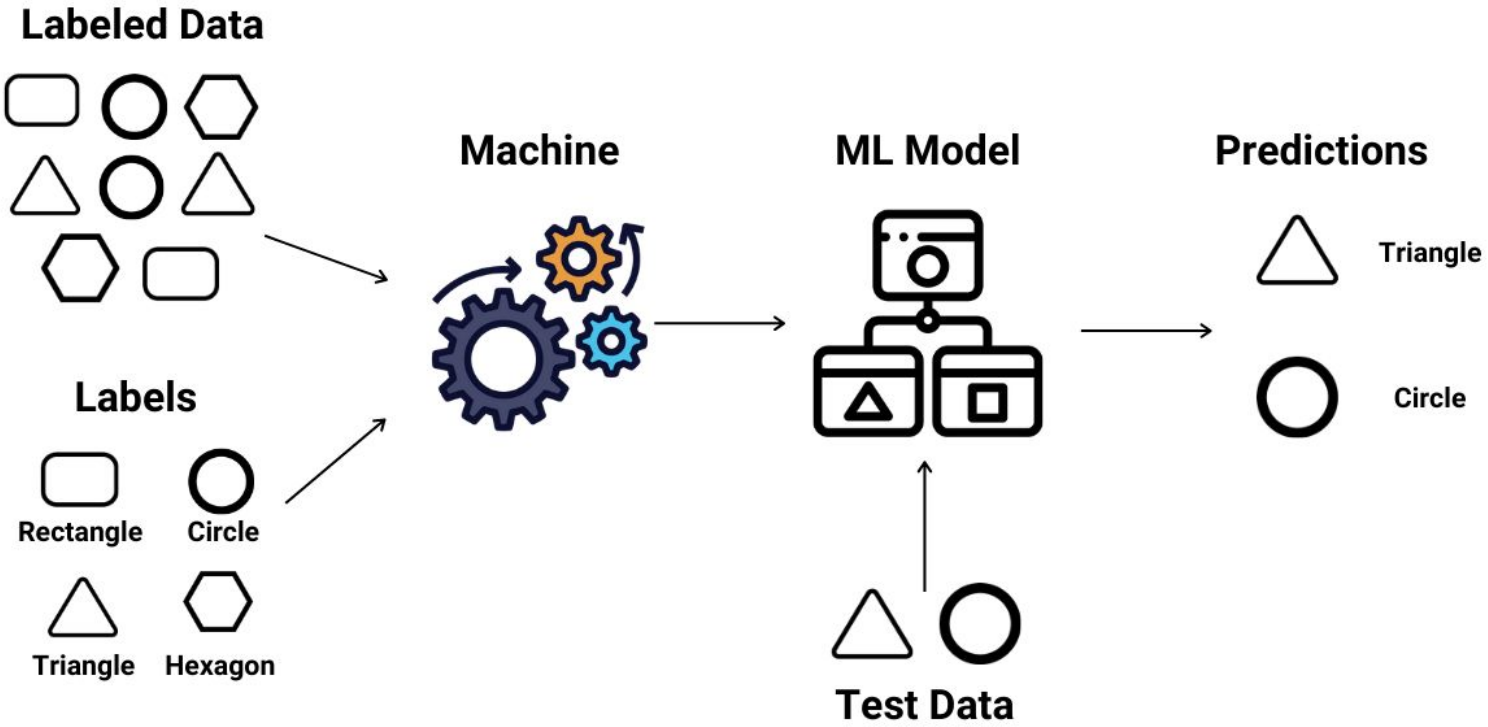
Learn from unlabeled dataset



Learn from labeled dataset

Optimize actions of agent in environment

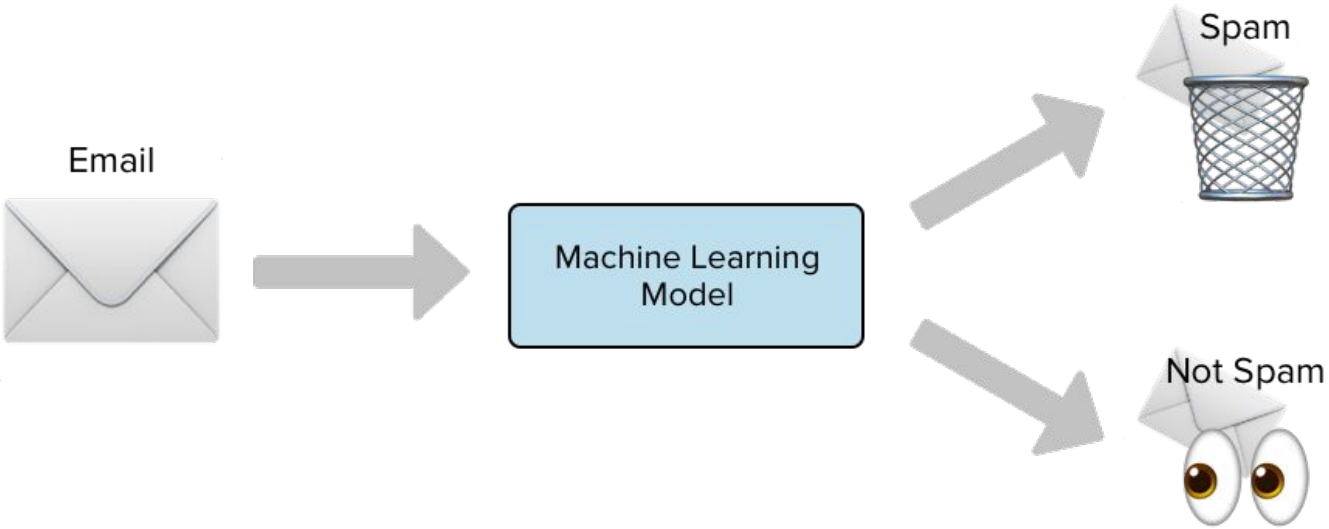
Class Work: is this a regression or classification task?



Class Work: is this a regression or classification task?

Distance to nearest market	Presence of borehole	House price (₦)
5 km (near balogun market, lagos)	Yes	₦75, 000, 000
15 km (outskirt of Abuja)	No	₦30, 000, 000

Class Work: is this a regression or classification task?



How to Choose a Learning Approach?



Input data:

- Can comprise **inputs**, plus **target outputs** (commonly referred to as **labels**) when known
  when we say a dataset is ***unlabeled****, it means we do not have the target outputs a priori*

Model output(s):

- Can be categorical/numerical, discrete/continuous, single/multiple outputs

Learning approach:

Supervised Learning

- **Task-driven**
- **Labeled data**
- **Performance feedback**

Unsupervised Learning

- **Data-driven**
- **Unlabeled data**

Reinforcement Learning

- **Reward-driven**
- **Dynamically-collected state/action data**
- **Performance feedback**

Supervised Machine Learning Terminologies

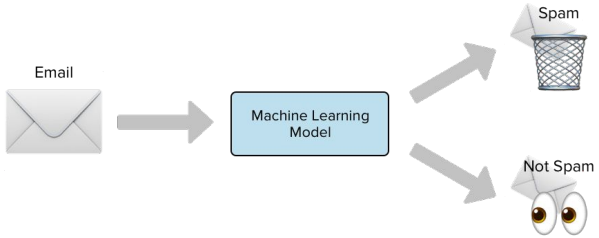
- **Label**: A label is the thing we're predicting. It is often denoted by y. Also called *dependent* or *target* variable

- **Feature:** A feature is an input variable that we use for our task. It is often denoted as x. Also called *independent* variable

- **Model:** A model encodes the relationship between features and label

- **Training:** is the process of teaching the model how to do well on the data or we say *learning* the model

- **Inference:** applying the trained model to *unseen* data. This is also called *testing* data



Supervised Learning

Given a *labeled* dataset, i.e. inputs $\mathbf{X}^{(i)}$ with labels $\mathbf{y}^{(i)}$:

$$\{(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{X}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{X}^{(n)}, \mathbf{y}^{(n)})\}$$

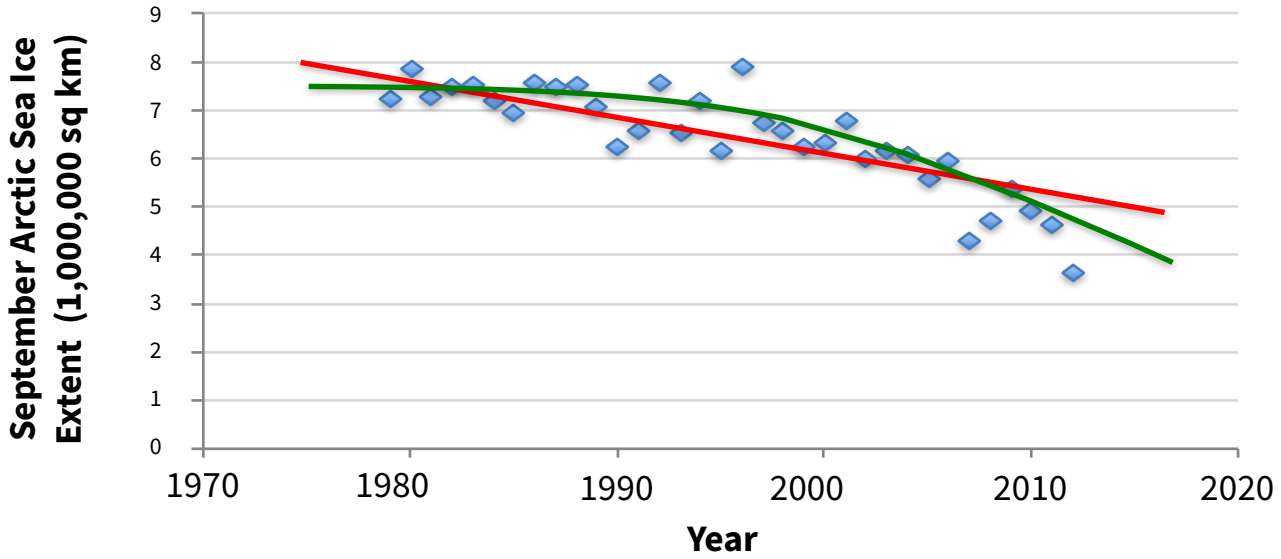
learn a function **f** to predict **y** given **X**, i.e. $\mathbf{f(X)} \cong \mathbf{y}$.

- **X** can have multiple elements, where each element represents some attribute ("**feature**")
- *Regression*: **y** is real-valued
- *Classification:* **y** is categorical

Supervised Learning: Regression

Learn a function **f** to predict **y** given **X**, i.e. **f(X) ≅ y**.

- *Regression*: **y** is real-valued

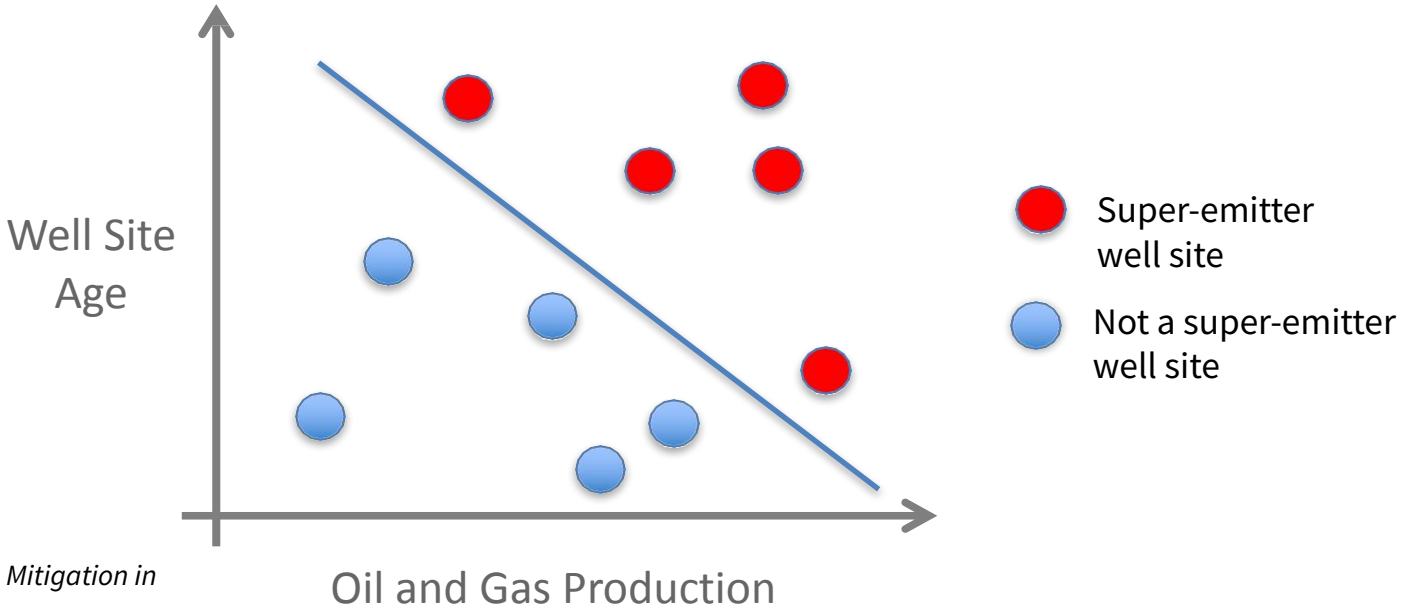


Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Supervised Learning: Classification

Learn a function **f** to predict **y** given **X**, i.e. **f(X) ≅ y**.

- *Classification:* **y** is categorical



A ML Approach to Methane Emissions Mitigation in the O&G Industry. Wang et al. 2020.

Supervised Learning: Common Models/Methods

- **Classification:** Support vector machines, random forests, logistic regression, naive Bayes, neural networks

- **Regression:** Linear regression, ridge regression, Gaussian processes, neural networks

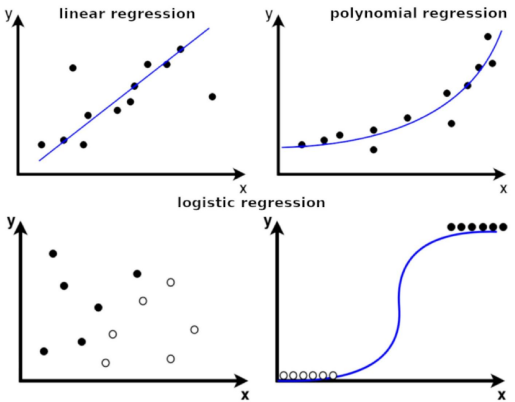


Image: Analytics Vidhya

Supervised Learning

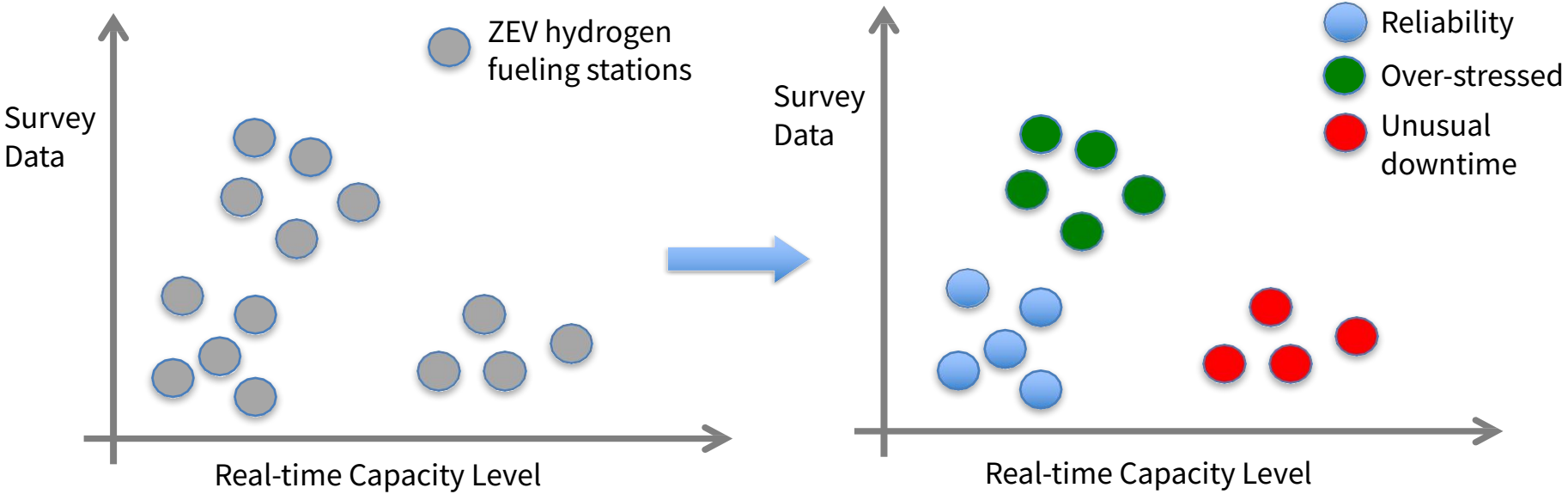
Criteria to choose an algorithm:

- Type of input and output data
- Data distribution, quality, and feature correlations
- Desired performance and robustness guarantees

Example use case: Classifying a well site as a super-emitter or not based on its age and oil & gas production as features, with super-emitter labels existing for training

Unsupervised Learning

Given an *unlabeled* dataset, i.e. inputs $\mathbf{X}^{(i)}$ with no labels, learn hidden structure in the data.



Unsupervised Temporal Clustering to Monitor the Performance of Alternative Fueling Infrastructure. Ramea 2020.

Unsupervised Learning: Common Models/Methods

- **Learning Patterns in the Data:** Principal Component Analysis (PCA), clustering, anomaly detection algorithms
- **Generative Modeling** (generate synthetic data): Normalizing flows, variational autoencoders, generative adversarial networks
- **Some Transfer Learning Tasks:** E.g., domain alignment

Unsupervised Learning

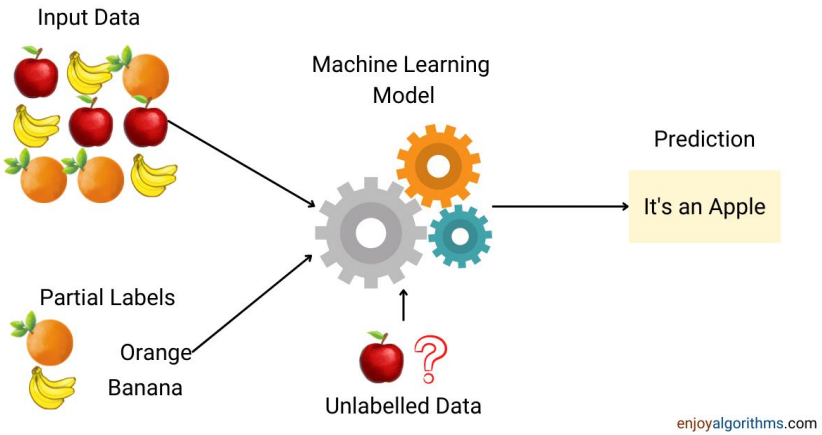
Criteria to choose an algorithm/model:

- What is the primary goal?
  - Clustering?
  - Synthetic data generation?
  - Domain adaptation?
- Data distribution, quality, and feature correlations

Example use case: Synthetic generation of building energy load profiles

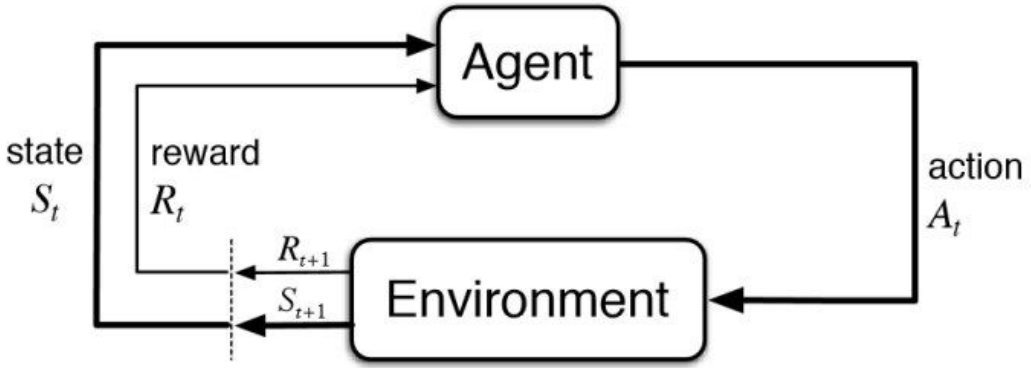
Semi-Supervised Learning

Semi-supervised learning falls between supervised learning and unsupervised learning. The dataset contains both **labelled** and **unlabelled** data points



https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning

Reinforcement Learning



Goal: Find the best policy that maximizes the total expected reward over time

RL is often used in settings where **control** is used

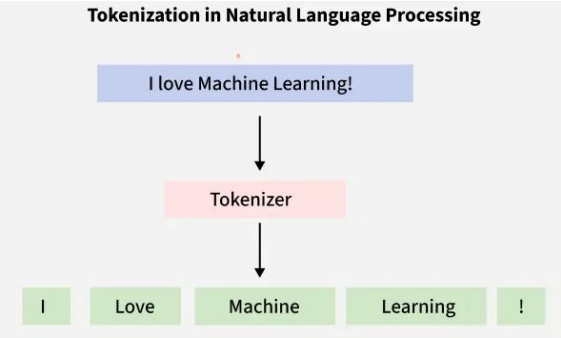
Examples:

- Game playing
- Energy optimization in a building
- Pan and zoom on satellite imagery

Figure from 5 Things You Need to Know about Reinforcement Learning

Some ML Approaches Associated with Data Modalities

Natural Language Processing (NLP)

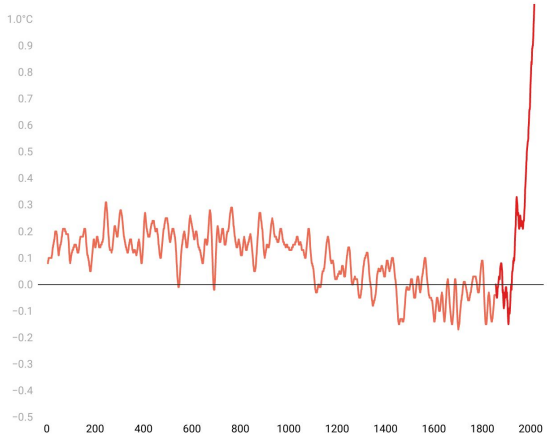


E.g., recurrent neural networks (RNNs), hidden Markov models (HMMs), large language models (LLMs)

Computer vision



Time series analysis

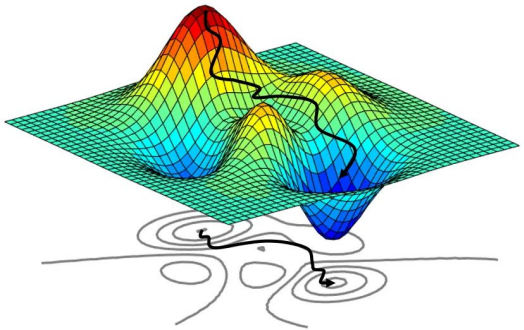


Images: Geeksforgeeks, Medium, TIME

ML for Surrogate Modeling

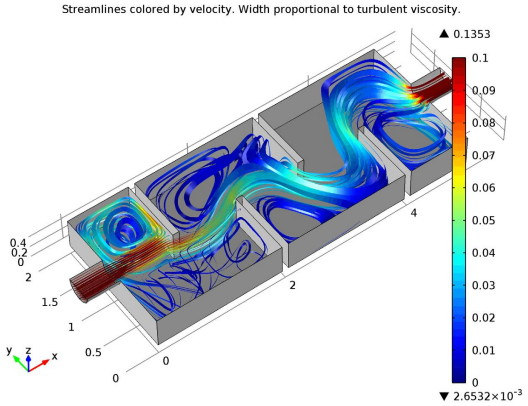
Optimization models

Approximate expensive objective functions and constraints to accelerate search for optimal solutions

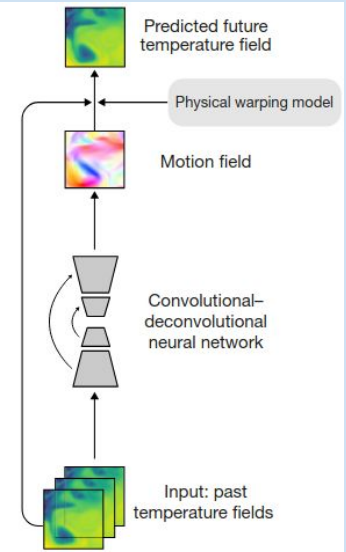


Physics models

Emulate complex physics simulations (like fluid dynamics or structural mechanics)



Physics-informed models



Images: Medium, Comsol, Reichstein et al 2019

Interpretable & Explainable ML

Most of AI is **not** causal!

- **Explainable AI** provides reasons for its decisions after the fact, even if its internal workings are complex or opaque.
- **Interpretable AI** has transparent internal logic that humans can directly understand by examining how the model itself works.

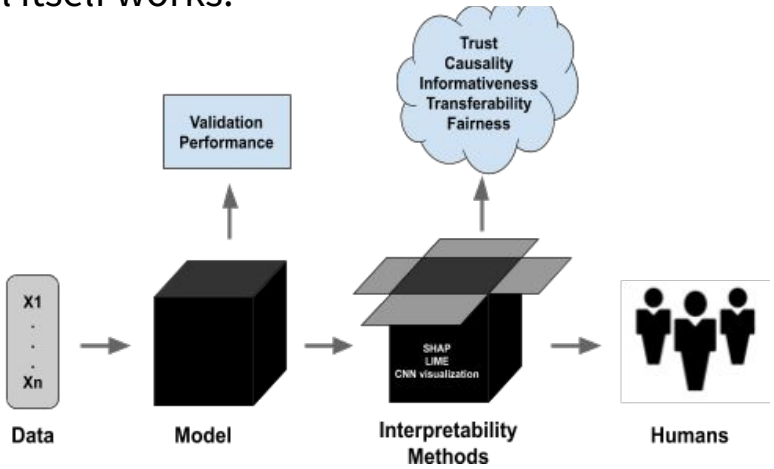
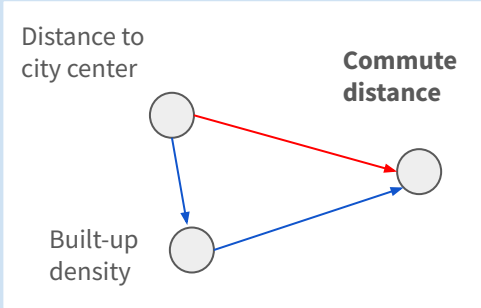


Image: ML CMU

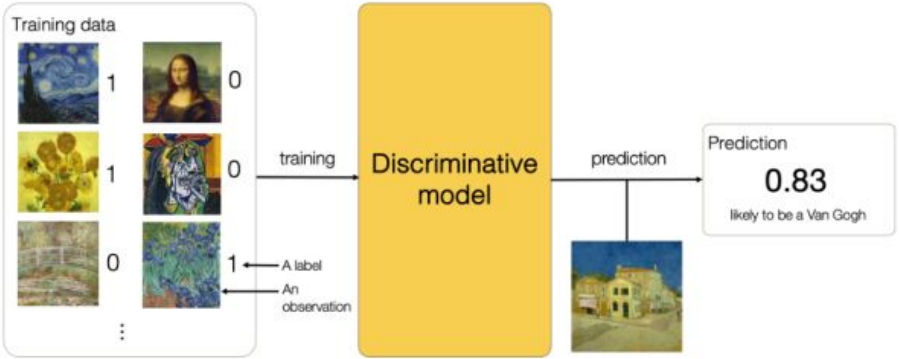
Causal ML

- Identify and quantify cause-and-effect relationships (instead of correlations)
- Enables prediction of intervention outcomes and counterfactual

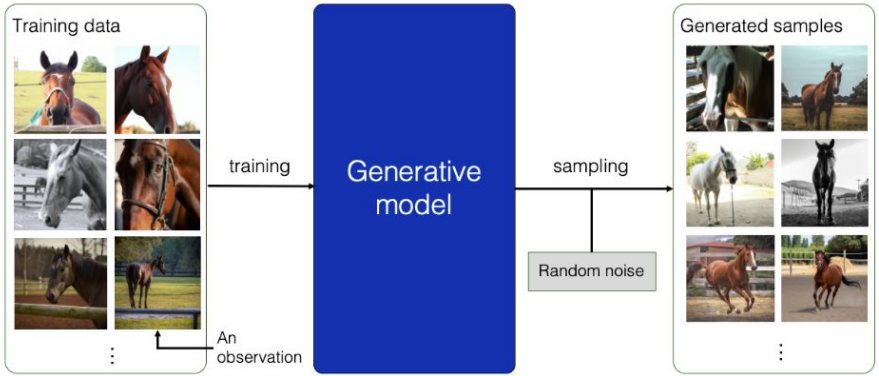


Generative vs. Discriminative ML

Discriminative AI models the ***conditional*** probability distribution, learning to map directly from inputs to outputs.



Generative AI ("GenAI") explicitly models the ***joint*** probability distribution of the data, learning how features and labels occur together.



Images: Analytics yogi

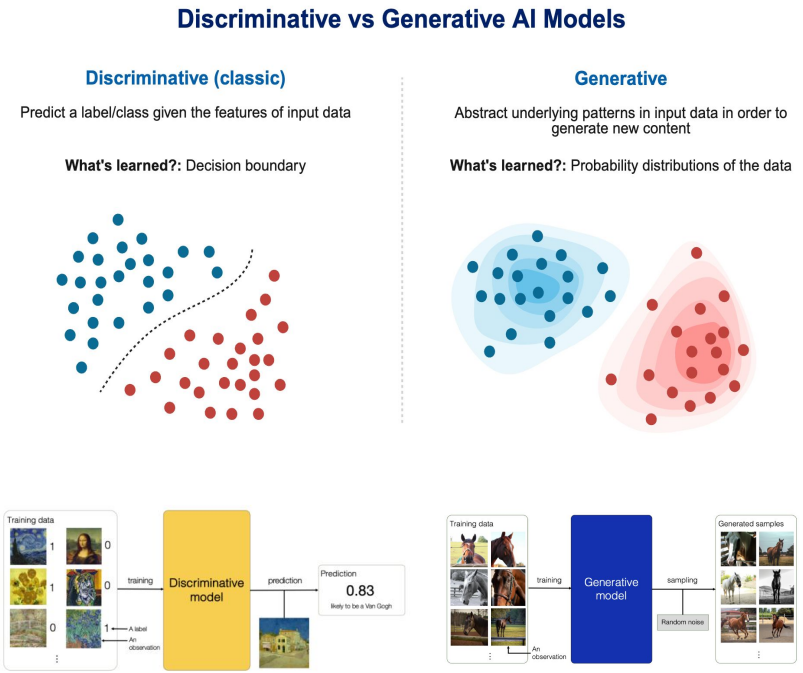
Discriminative vs. Generative Models

Discriminative models

- Used to separate (“discriminate” between) different outputs based on the value of the input.
- Learn the conditional probability distribution P(Y|X) of targets Y conditioned on inputs X
- E.g, predicts a label or category (e.g., spam or not spam)

Generative models

- Learn the joint probability distribution P(X,Y) over the inputs X and target Y
- Learns to create (“generate”) new content sampled from the learned distribution — text, images, audio, code



Example: Generative Pre-trained Transformer (GPT)

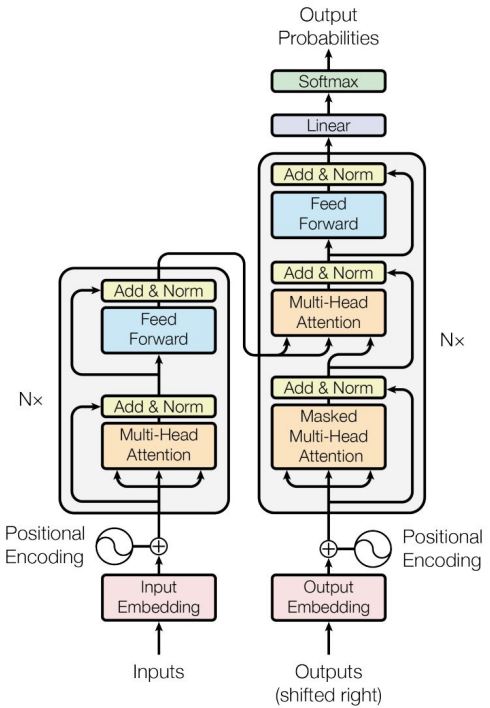
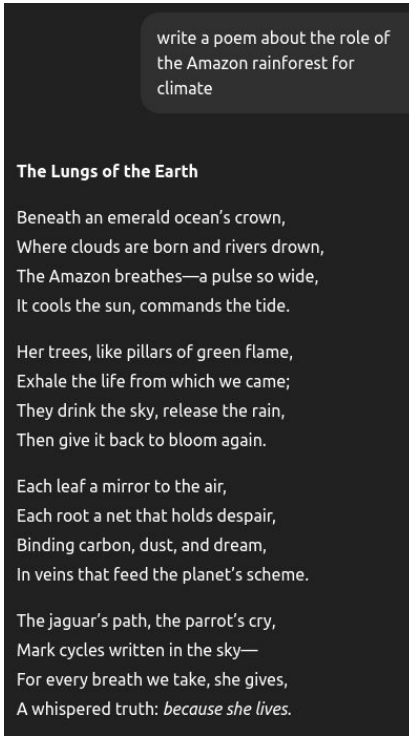


Figure 1: The Transformer - model architecture.

Figure source: Vaswani et al. (2017)



Core Idea: Predicts next token from previous ones *(unsupervised)*

Strengths: Analyzes patterns in diverse data at scale, often with impressive results

Limitations:

- Requires massive data & compute
- Inherits biases and inaccuracies from training data
- Can hallucinate without uncertainty estimates

Key Characteristics:

- Interpolates patterns
- Pattern matching vs. direct fact retrieval

Class Work: Synthetic Data Generation

If a dataset is synthetic (artificially generated), would you consider it "safe" to share publicly? Why or why not?

Lecture Outline

What is Machine Learning?

Types of Machine Learning

Data

Q&A

Understanding Linear Regression Algorithm

Data collection

Data preparation for ML

The Importance of Data

- Data is the core of machine learning
- The quality of your model is dependent on the quality of your data
- How much data is enough?
    - Problem-dependent
    - There are some okay heuristics, but almost always comes down to how things work empirically



Figure from What is Big Data? A Quick Introduction for Analytics and Data Engineering Beginners

Data Collection and Preparation

1. Define the task
2. Select data which aligns with that task
    a. Is the data labeled? If not, can you label the data?
3. Explore and clean the data
    a. Handle missing values (exclude data or impute)
    b. Investigate label quality
4. Split the data into training, validation, and test

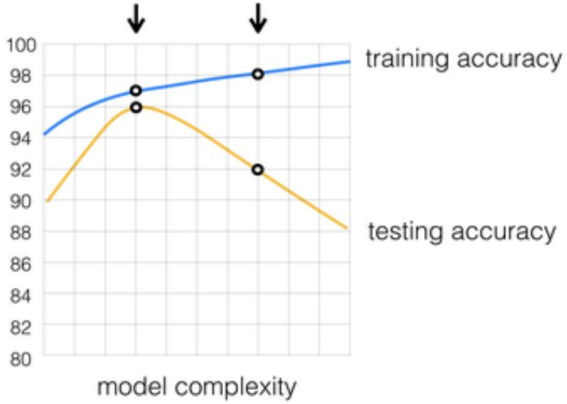
Splitting the Dataset

- Why do we need to split the dataset?
  - To avoid *overfitting* and assess model performance on new data



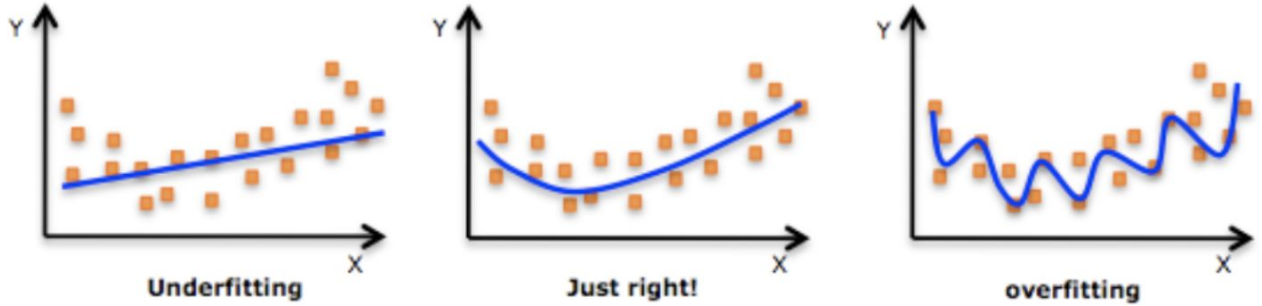
- Training: Learn model parameters
- Validation: Select hyperparameters and compare models
- Testing: Evaluate the best model to assess generalization performance on *unseen* data

Overfitting vs. Generalization



Solution: **k-fold Cross-Validation**

Assess how well a predictive model will perform on "unseen" data by repeatedly splitting the dataset into *k* training and testing subsets.



Define the Task

1. Outline the **requirements** of the **problem** to be solved:
   - Find the temporal, spatial constraints of the problem (making the ML solution valuable)

     e.g., weather forecasting can be useful to optimize renewable energy production, but this can happen for optimizing existing infrastructure (short timescale), or for scoping the placement of new production sites (long timescale)
2. Establish the **specifications** your ML model has to comply with:
   - What are the specifications for **training**? And for **deployment**?
3. Frame the **ML task** – some examples:

Classification

Regression

Clustering

Image segmentation

Data augmentation

Anomaly Detection

On-line Optimization

Data Selection

- Select data which aligns with the ML task
  - By reviewing **state-of-the-art** applications
  - By asking **domain experts** (identify researchers/practitioners working in this area)
  - Carefully assessing the **data requirements** for the ML task

    Note: You may need to review/relax some of the previous **specifications** based on the data that is actually **available**.

- Is the **dataset labeled**? If not, can you label the data? Is it worth it to annotate to solve the problem, considering alternative approaches?

Explore and Clean the Data

1. Remove duplicate or irrelevant observations
2. Fix structural errors
3. Filter unwanted outliers
4. Handle missing data
5. Validate and quality-assure

From Data Cleaning: Definition, Benefits, And How-To - Tableau

Class Work: Data Ownership

Who should own the data used to train AI models–the user, the company, or someone else?

Summary so far

📝 Machine learning is useful for particular problems.

- It is not a silver bullet.

📝 There are many different types of machine learning.

- Learning approaches: Supervised, unsupervised, reinforcement learning
- Data modalities: NLP, computer vision, time series analysis
- Requirements: e.g., interpretable and explainable, physics-informed
- Other choices: Generative vs. discriminative, large vs. small, general-purpose vs. task-specific

📝 Data is the core of machine learning!

Class Work: Yes or No

From a technical perspective, artificial intelligence (AI) primarily refers to large, general-purpose generative models such as large language models.

https://partici.fi/53603205

Q&A

Lecture Outline

What is Machine Learning?

Types of Machine Learning

Data

Q&A

Understanding Linear Regression Algorithm

- Equation of a straight line
- Loss function
- Optimising the parameter

Equation of a Straight Line

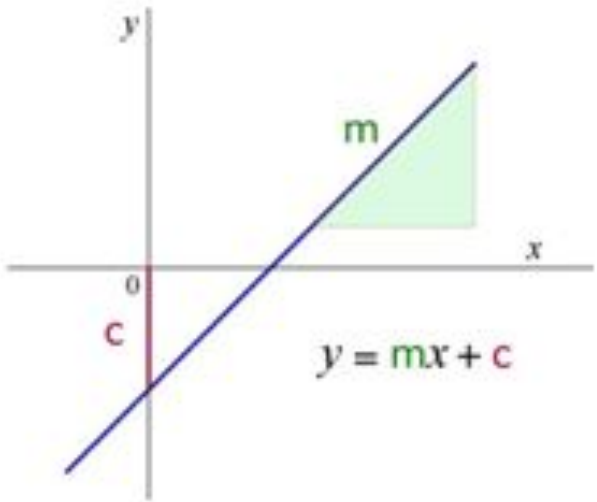
$$y = mx + c$$

Gradient **y-intercept**

Equation of a Straight Line

$$y = mx + c$$

Gradient **y-intercept**

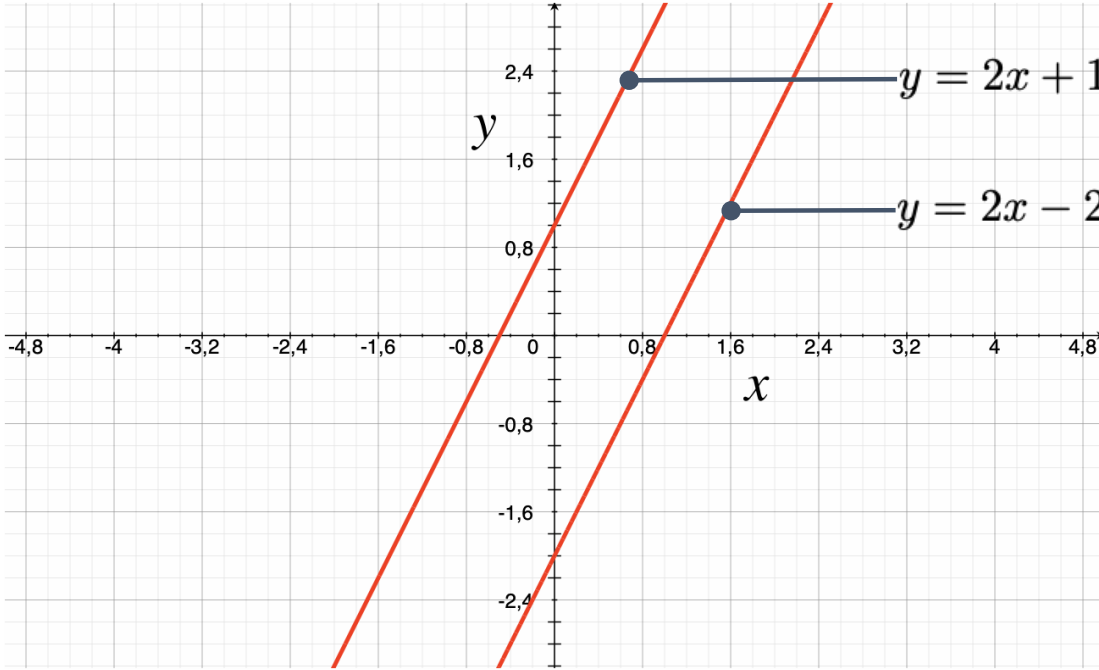


https://www.easycalculation.com/maths-dictionary/equation_of_a_straight_line.html

Equation of a Straight Line

$$y = mx + c$$

Gradient **y-intercept**

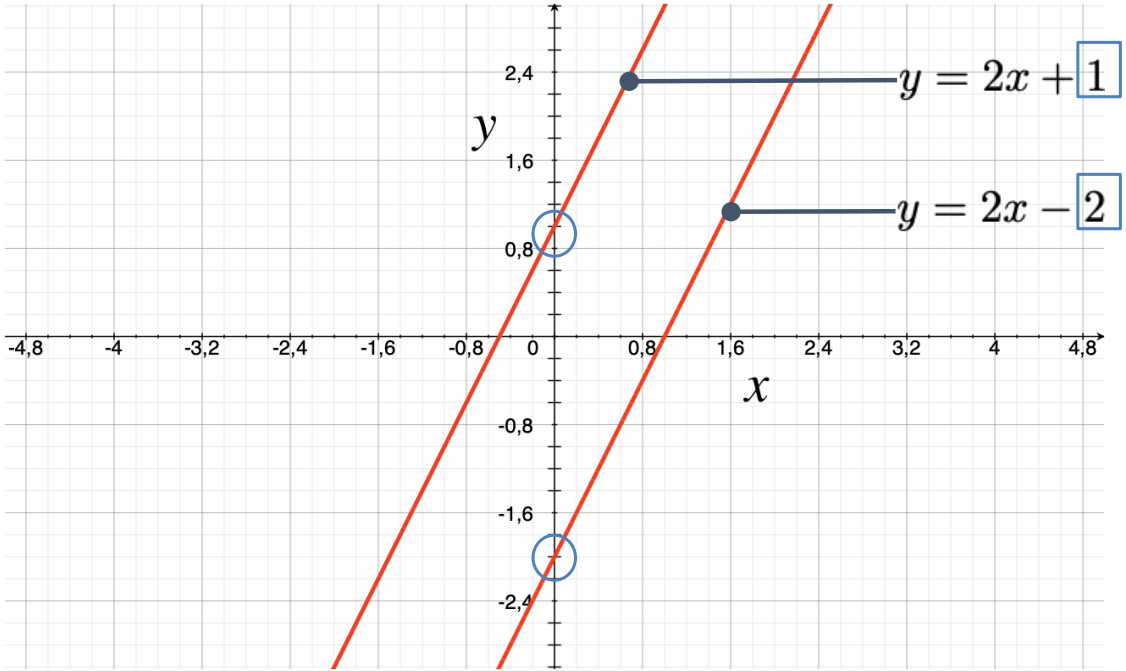


https://www.easycalculation.com/maths-dictionary/equation_of_a_straight_line.html

Equation of a Straight Line

$$y = mx + c$$

y-intercept

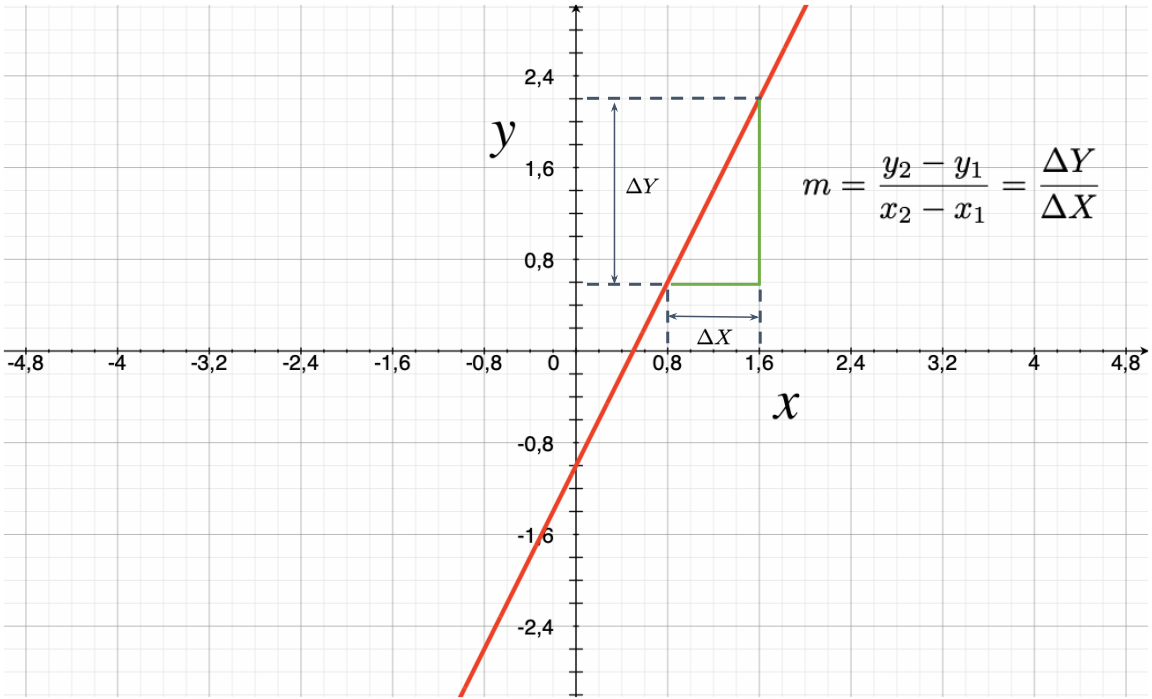


https://www.easycalculation.com/maths-dictionary/equation_of_a_straight_line.html

Equation of a Straight Line

$$y = mx + c$$

Gradient

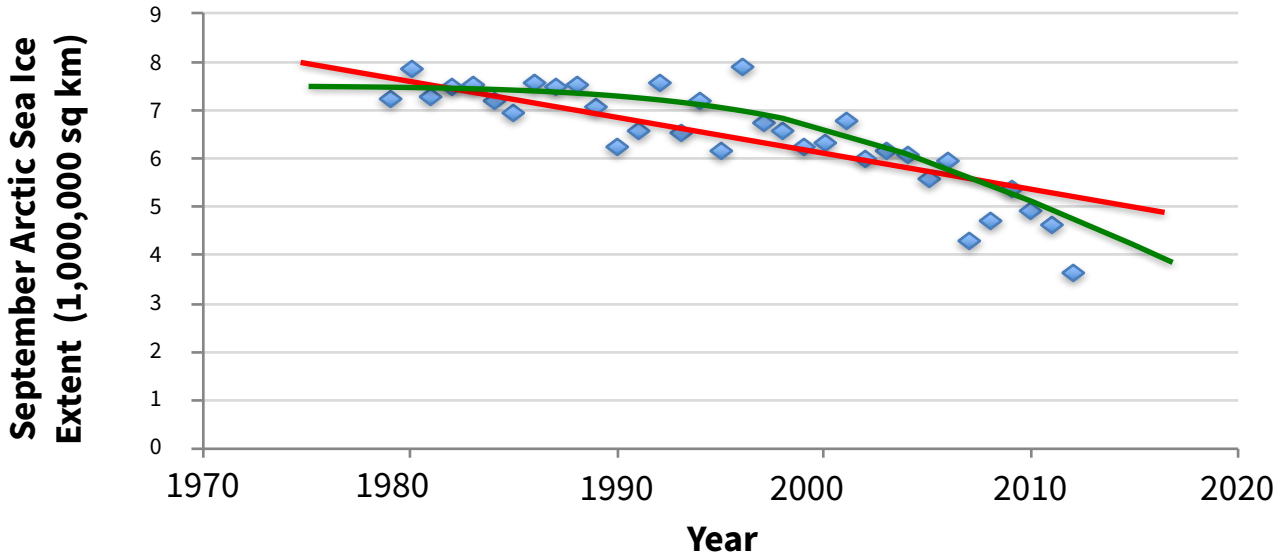


https://www.easycalculation.com/maths-dictionary/equation_of_a_straight_line.html

Predicting Arctic Sea Ice Extent

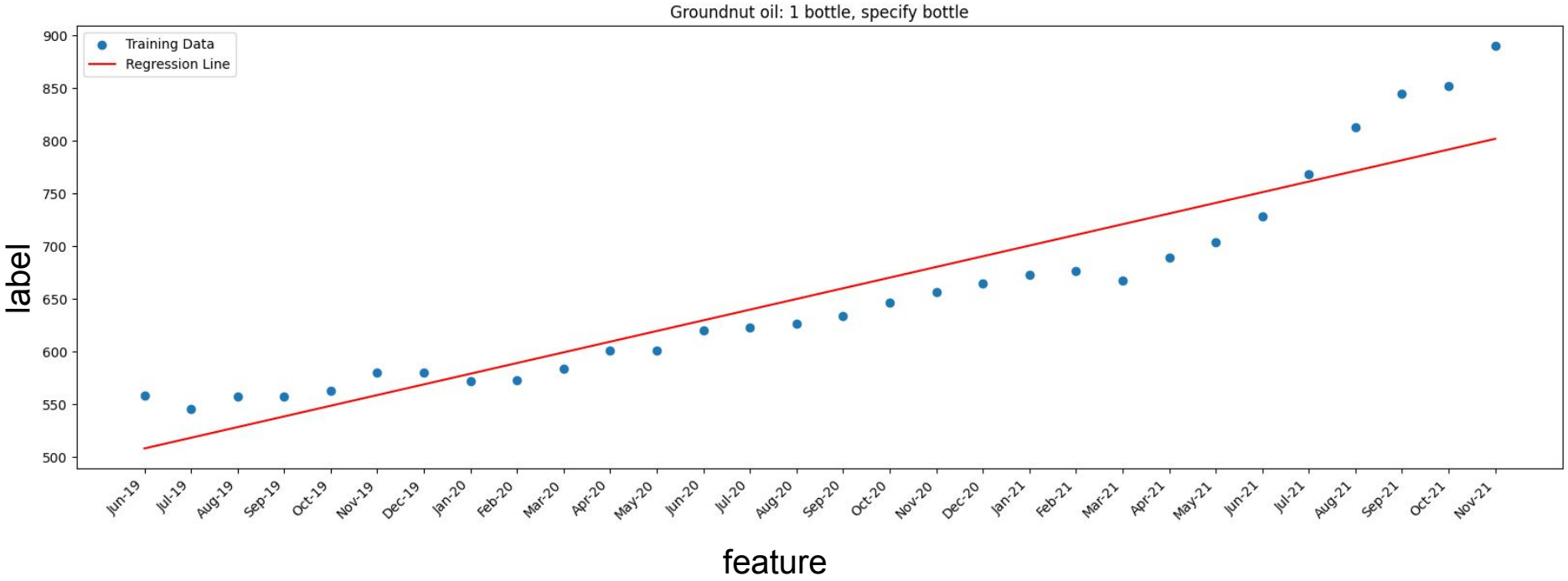
Learn a function **f** to predict **y** given **X**, i.e. **f(X) ≅ y**.

- *Regression*: **y** is real-valued



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Predicting Price of Groundnut Oil

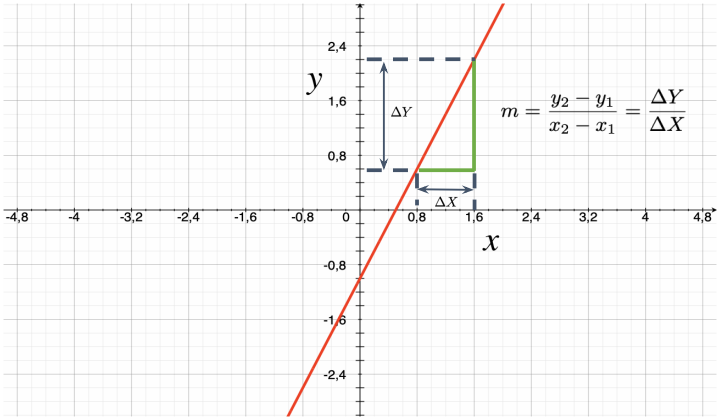


Practise Notebook:
https://colab.research.google.com/drive/1lS_k0gfkzz8lXZqA9WscHuh9QJsEqDcG?usp=sharing#scrollTo=3a4aSRo6T8hi

Linear Regression

Linear Regression assumes that there is a **linear** relationship between the features and the numerical label.

$$y = mx + c$$



In ML, we usually write the equation as:

$$y = b + w_1 x_1$$

where:

is the label

is the bias (y-intercept), sometimes written as w_0

is the weight of feature x_1

Linear Regression Model

The linear regression model predicts an output $\hat{y}$ based on input features $x_1, x_2, \cdots, x_n$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

where:

- $\hat{y}$ is the prediction
- w_0 is the y-intercept (bias)
- $w_1, w_2, \cdots, w_n$ are the weights (coefficients)
- $x_1, x_2, \cdots, x_n$ are the input features

Linear Regression Model: Loss Function

To measure the difference between the predictions $\hat{y}$ and the actual y values (labels)

we use the **Mean Squared Error** (MSE)

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

where

m is the number of training examples

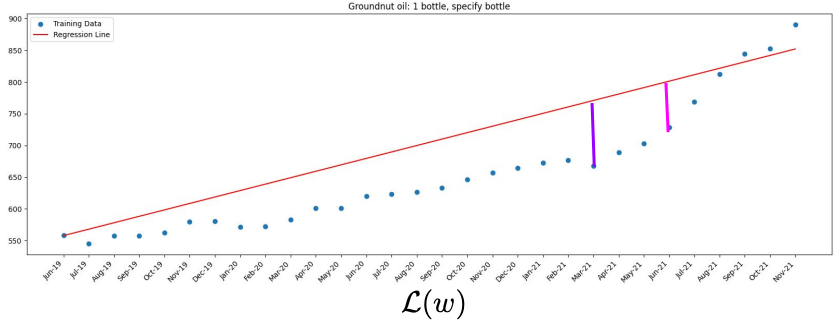
$\mathcal{L}(w)$ is the loss function

Linear Regression Model: Loss Function

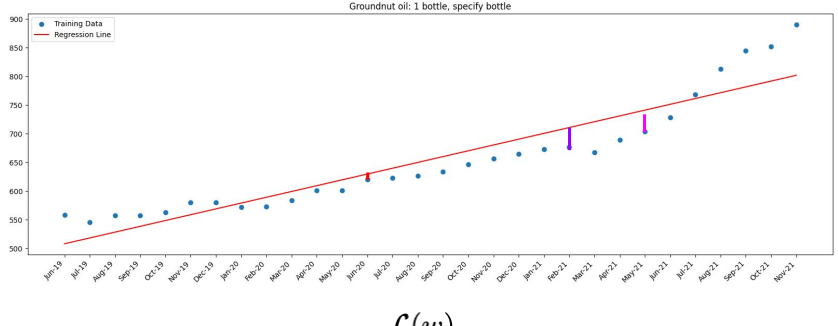
- **Loss** is a penalty for a *bad* prediction - it is a number indicating how **wrong** the model's prediction was on a single data point. If the model's prediction is perfect, the loss is *zero*, otherwise, the loss is *non-zero*

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

- Training a linear regression model means learning a *set of* values for the weights from labelled examples that reduces the loss



$\mathcal{L}(w)$



$\mathcal{L}(w)$

Interactive illustration: https://developers.google.com/machine-learning/crash-course/linear-regression/parameters-exercise

Linear Regression Model: Optimal Weight (minimising the loss)

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

substitute

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(y_i - (w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in}))^2$$

The goal is to find $w_0, w_1, w_2, \cdots, w_n$ that minimises $\mathcal{L}(w)$

Linear Regression Model: Optimal Weight (minimising the loss)

We can achieve this in two ways:

- **Closed form solution (normal equation)**

- **Gradient descent optimisation**

Optimising the Parameters: Normal Equation

If X is the *design* matrix (including the bias term), and y is the target vector, the optimal weights are given by

$$w = (X^T X)^{-1} X^T y$$

This only works when $X^T X$ is **invertible** and for small datasets

Optimising the Parameters: Gradient Descent

The weights are updated iteratively using gradient of the loss function

$$w_j := w_j - \alpha \frac{\partial \mathcal{L}}{\partial w_j}$$

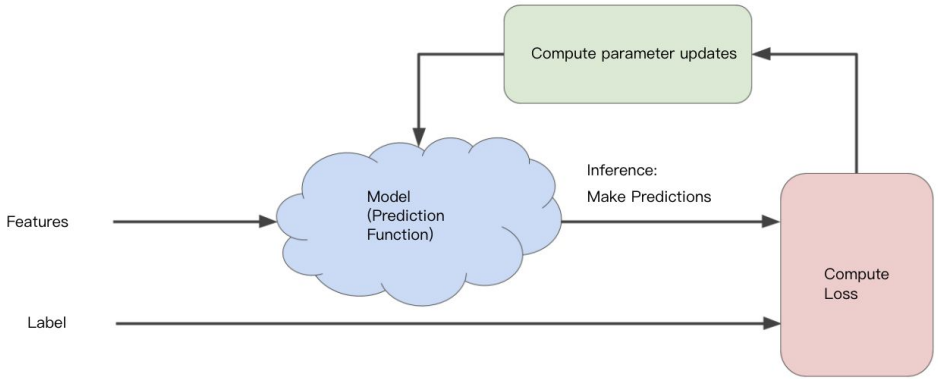
where α is the learning rate and

$$\frac{\partial \mathcal{L}}{\partial w_j} = -\frac{2}{m} \sum_{i=1}^{m} x_{ij}(y_i - \hat{y}_i)$$

This method is used for large datasets

Finding the Minimum of a Function

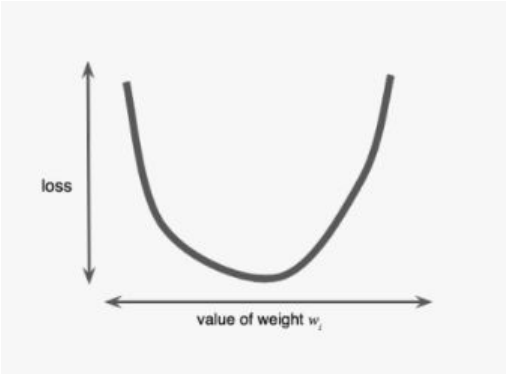
We begin our training by starting with an initial guess for the weights and biases and iteratively adjust the numbers until we get weights and bias with the lowest loss

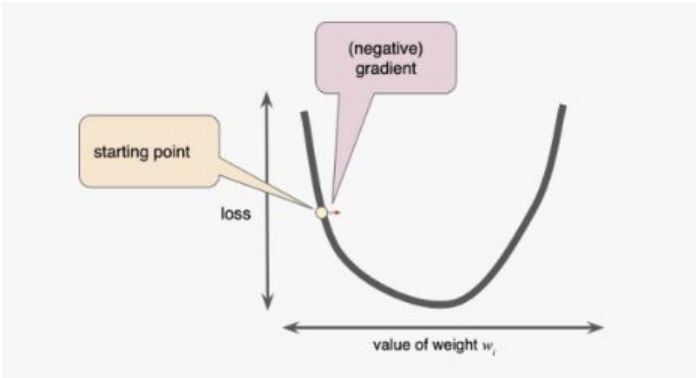


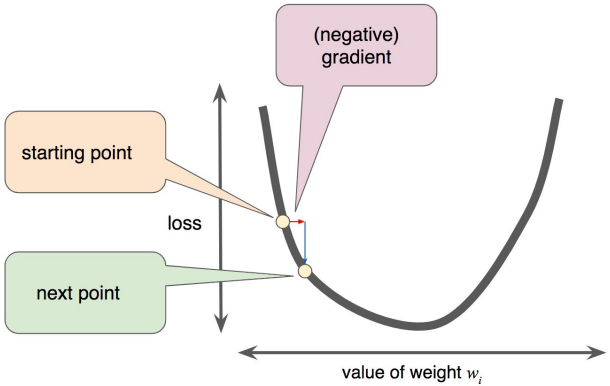
The gradient descent way!

https://github.com/litaotao/machine-learning-crash-course

Finding the Minimum of a Function





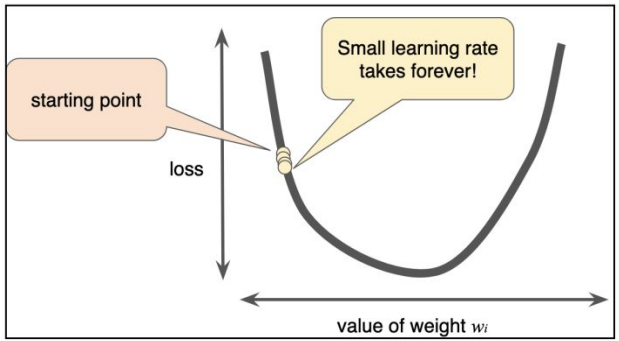


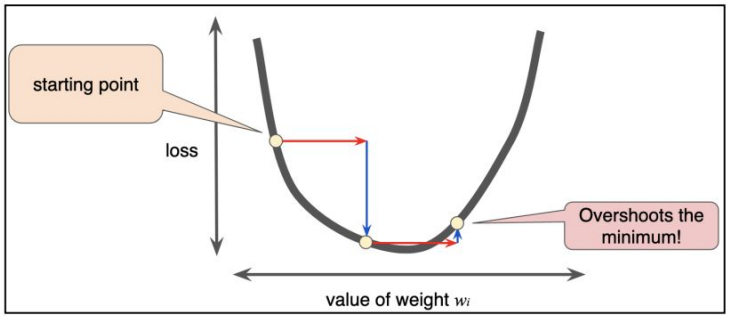
The gradient descent way!

$$w_j := w_j - \alpha \frac{\partial \mathcal{L}}{\partial w_j}$$

The Learning Rate Matters

Learning rate is a **hyperparameter** that determines the size of the steps taken at each iteration when updating the model's parameter during training

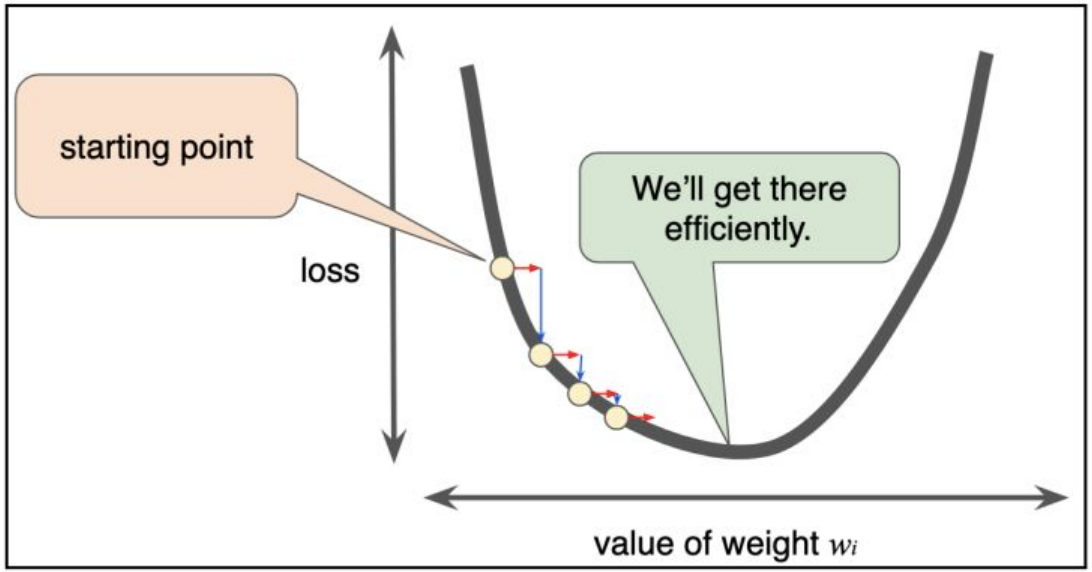




$$w_j := w_j - \alpha \frac{\partial \mathcal{L}}{\partial w_j}$$

https://cedar.buffalo.edu/~srihari/CSE676/8.3%20BasicOptimizn.pdf

The Learning Rate Matters



https://cedar.buffalo.edu/~srihari/CSE676/8.3%20BasicOptimizn.pdf

We'll Stop here today!
Thank You

You can contact me:

- Email: **tejumade@tri-ai.org**
- Google Scholars, LinkedIn: **tejumade Afonja**
- X,Bluesky,Github: **tejuafonja**
- Website: **tejuafonja.com**
- **My NGO website: tri-ai.org**