Artificial Intelligence

Biggest confusion

Artificial Intelligence:

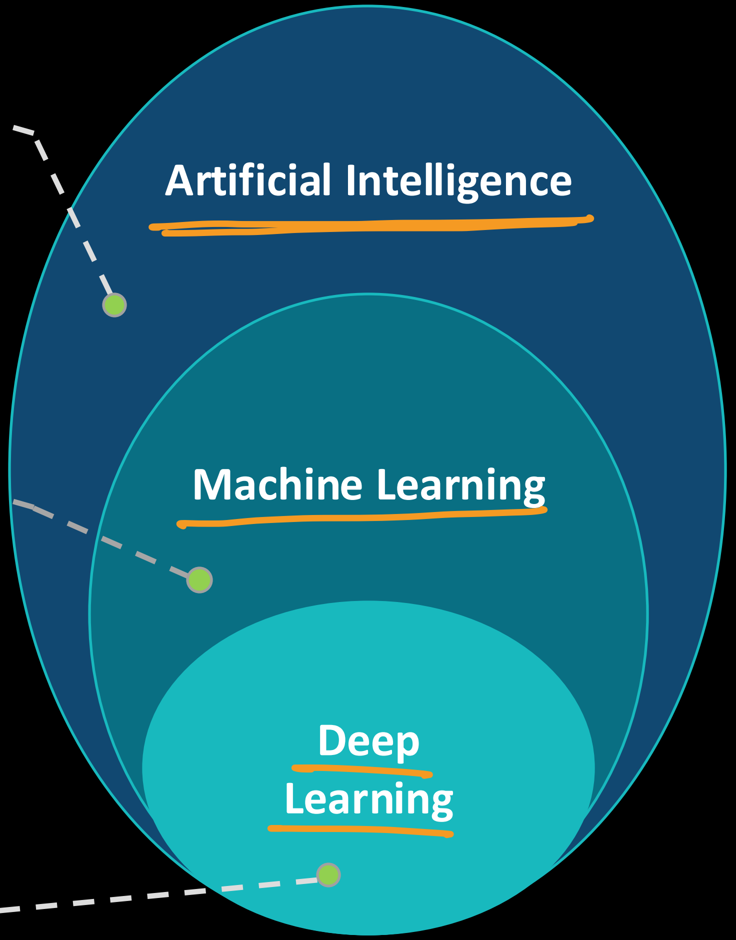
- A technique which enables machine to mimic human behavior

Machine Learning:

- Subset of AI which uses statistical methods to enable machines to improve the experience

Deep Learning:

- Subset of ML which makes the computation of multi-layer neural network feasible



What is AI?

Cognitive thinking → Experienced based thinking

- Artificial Intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans
- Any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals
- The theory and development of computer system able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision making and translation
- Often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving”

The Evolution of Artificial Intelligence

Pre-1950
Early Foundations

1950s
The Birth of AI

1960s-1970s
Rise and Fall of Early Enthusiasm

1980s
Expert Systems and the AI Renaissance

1990s-2000s
The Statistical Revolution

2010s-Present
The Deep Learning Era

GenAI → text/ audio/ video/ images

History

- 1940s–1950s: The Birth of AI
- 1943: First neural network concept by Warren McCulloch and Walter Pitts.
- 1950: Alan Turing proposes the Turing Test to evaluate machine intelligence.
- 1956: The term "Artificial Intelligence" is coined by John McCarthy at the Dartmouth Conference, recognized as the official birth of AI research.
- 1961: Unimate, the first industrial robot, starts working on a General Motors assembly line.
- 1964: ELIZA, the first chatbot, demonstrates early natural language understanding.
- 1966: Shakey, the first mobile intelligent robot, navigates its environment.
- 1970s: First "AI winter" as progress slows and funding declines.

History

- 1980: Development of expert systems like XCON shows promise for business uses.
- 1987–1993: Second "AI winter" due to limited capabilities and high expectations.
- 1997: IBM’s Deep Blue defeats world chess champion Garry Kasparov, proving AI’s strategic capabilities.
- 1998: Kismet, the first robot able to recognize and simulate emotions, is built at MIT.
- 2002: Roomba, the autonomous vacuum cleaner, brings simple AI into homes.
- 2008: Introduction of speech recognition features in consumer tech, such as Apple’s Siri prototype.

Deep Learning Evolution

- 2011: IBM Watson beats human champions on Jeopardy!, showcasing advances in NLP.
- 2012: AlexNet wins ImageNet competition with deep learning, igniting neural network revolution. → CNN
- 2014: GANs (Generative Adversarial Networks) and chatbots like Eugene Goostman arrive.
- 2014: Amazon releases the Echo with Alexa, bringing voice assistants mainstream. → NLP
- 2016: AlphaGo defeats Go champion Lee Sedol, achieving a feat thought decades away.

Transformers and Generative AI

- 2017: Google introduces the transformer architecture in "Attention Is All You Need," revolutionizing NLP.
- 2018: BERT and GPT mark the rise of large language models.
- 2020: OpenAI releases GPT-3, with 175B parameters, prompting excitement about generative AI.
- 2021: OpenAI launches DALL-E for text-to-image generation.
- 2022: OpenAI debuts ChatGPT, bringing conversational AI into the spotlight.

AI Mainstreaming and Regulation

understands multiple types of inputs,
→ text / image / audio / video

- 2023: GPT-4 demonstrates multimodal abilities; Google releases Bard, and global discussions on AI risks intensify.
- 2023: The US and UK host the first global summits and executive orders on AI safety.
- 2024: Google DeepMind releases Gemini 1.5; OpenAI announces Sora, a text-to-video model.
- 2024: The 2024 Nobel Prize in Chemistry is awarded for AlphaFold's protein structure prediction breakthroughs.
- 2025: France hosts the Artificial Intelligence Action Summit; new AI assistants and data center investments announced.

Why are we talking about it now ?

process huge and unstructured data

More Computational Power

GPU + TPU

more data = more accuracy

More Data

extreme Gradient Boosting (XGB)

Better algorithms

$$$

Broad investment

Applications of AI

→ CNN

- Healthcare Diagnostics & Drug Discovery → Cancer / diabetes
  - AI analyzes medical images, predicts patient outcomes, and accelerates drug discovery, revolutionizing disease detection and treatment
- Personalized Customer Experiences → Recommendation Engine
  - Recommendation engines and conversational AI personalize product suggestions and automate customer support in e-commerce and services
- Autonomous Vehicles (Self-Driving Cars) → Reinforcement Learning
  - AI powers real-time decision-making for self-driving cars, enhancing safety and efficiency in transport and logistics.
- Natural Language Processing (NLP) & AI Assistants
  - Virtual assistants like ChatGPT, Gemini, and Claude power chatbots, customer service, and productivity tools with advanced language understanding.
- Financial Services & Fraud Detection
  - AI identifies fraudulent activities, automates financial planning, and personalizes investment advice, reshaping banking and investment

Applications of AI

- Generative AI (Content Creation)
  - AI tools like DALL-E, Runway, and GPT-4o generate human-like text, images, music, and video, fueling rapid innovation in media and design.
- Cybersecurity
  - Platforms such as Darktrace use AI to detect, respond to, and prevent sophisticated cyber threats in real-time.
- Education and Personalized Learning
  - Adaptive learning platforms like Khan Academy and AI tutors tailor content to student needs, improving engagement and performance.
- Supply Chain & Retail Optimization
  - AI forecasts demand, manages inventory, and automates logistics in real-time, streamlining operations across supply chains and retail.
- Smart Cities & IoT
  - AI manages urban infrastructure—traffic, energy, security—by analyzing sensor data, optimizing resources, and improving quality of life

Machine Learning

main Requirement → Data / Resources

What is machine learning ?

- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E
  - Tom Mitchell, 1997

- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
  - Arthur Samuel, 1959

- Machine Learning is the science (and art) of programming computers so they can learn from data

* traditional programming
↳ Explicit programming

→ Detect if email is a spam

→ understand all conditions which can mark an email as a spam.

→ source ip

→ predefined words

⋮

```
if email source == "..." then
    mark spam
else if email contains ["lottery"...] then
    mark spam
    ⋮
else
    mark not spam / ham
```

---

language

input → [implement logic] → output

formula / Logic is already known

* ML programming
↳ Existing data with required labels

email content	type
You have won a lottery..	spam
you have a new task...	ham

Observed data

patterns / Relationship

labelled

type ↑ label / class

data → [ML program] ← python, R

[ML program] → [formula] → Model

input → [formula] → output

Logic / formula is unknown

Where to use machine learning ?

- Problems for which existing solutions require a lot of fine-tuning or long lists of rules:
  - one Machine Learning algorithm can often simplify code and perform better than the traditional approach

→ huge data / unstructured data → Extract patterns

- Complex problems for which using a traditional approach yields no good solution:
  - the best Machine Learning techniques can perhaps find a solution
- Fluctuating environments:
  - a Machine Learning system can adapt to new data
- Getting insights about complex problems and large amounts of data

→ generation → text / audio / video / image

→ text classification

image

Examples of Applications

- Analyzing images of products on a production line to automatically classify them
  - This is image classification, typically performed using convolutional neural networks (CNN)
- Detecting tumors in brain scans
  - This is semantic segmentation, where each pixel in the image is classified (typically use CNNs)
- Automatically classifying news articles
  - This is natural language processing (NLP), and more specifically text classification
- Automatically flagging offensive comments on discussion forums
  - This is also text classification, using the same NLP tools → sentiment analysis < +ve, neutral, -ve
- Forecasting your company's revenue next year, based on many performance metrics
  - This is a regression task (i.e., predicting values) that may be tackled using any regression model
- Making your app react to voice commands
  - This is speech recognition, which requires processing audio samples: since they are long and complex sequences, they are typically processed using RNNs, CNNs, or Transformers

long patterns

Examples of Applications

- Detecting credit card fraud
  - This is anomaly detection example

Rare behavior

- Segmenting clients based on their purchases so that you can design a different marketing strategy for each segment

market / basket analysis

  - This is clustering example

column / feature / variable / random variable

- Representing a complex, high-dimensional dataset in a clear and insightful diagram
  - This is data visualization, often involving dimensionality reduction techniques
- Recommending a product that a client may be interested in, based on past purchases
  - This is a recommender system
- Building an intelligent bot for a game
  - This is often tackled using Reinforcement Learning

Types of Machine Learning

Supervised

unsupervised

Reinforcement

x	y
1	3
2	5
3	7
4	9
5	11
8	?

$$\boxed{y = f(x)}$$

$$\boxed{y = 2x + 1}$$

formula | model

prediction → Regression

X → independent variable

y → dependent variable

Supervised Learning → prediction → labelled data → data with the answers

- The majority of practical machine learning uses supervised learning
- Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output

$$\boxed{Y = f(X)}$$ model / formula

- The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data
- It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process
- We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher
- Learning stops when the algorithm achieves an acceptable level of performance (80%)

Supervised Learning – Problems

- **Regression** → dependent variable is a non-categorical variable
  - Related to predicting future values
  - E.g.
    - Population growth prediction
    - Expecting life expectancy
    - Market forecasting/prediction
    - Advertising Popularity prediction
    - Stock prediction
  - Algorithms
    - Linear and multi-linear regression
    - Logistic regression
    - Naïve Bayes
    - Support Vector Machine

Supervised Learning – Problems

- **Classification** → dependent variable is a categorical variable
  - Related to classify the records
  - E.g.
    - Find whether an email received is a spam or ham
    - Identify customer segments
    - Find if a bank loan is granted
    - Identify if a kid will pass or fail in an examination
  - Algorithms
    - Logistic Regression
    - Decision Tree
    - Random Forest
    - Support Vector Machine
    - K-nearest neighbor

test	result
70	1
35	0
40	1
25	0
80	1
.	
.	

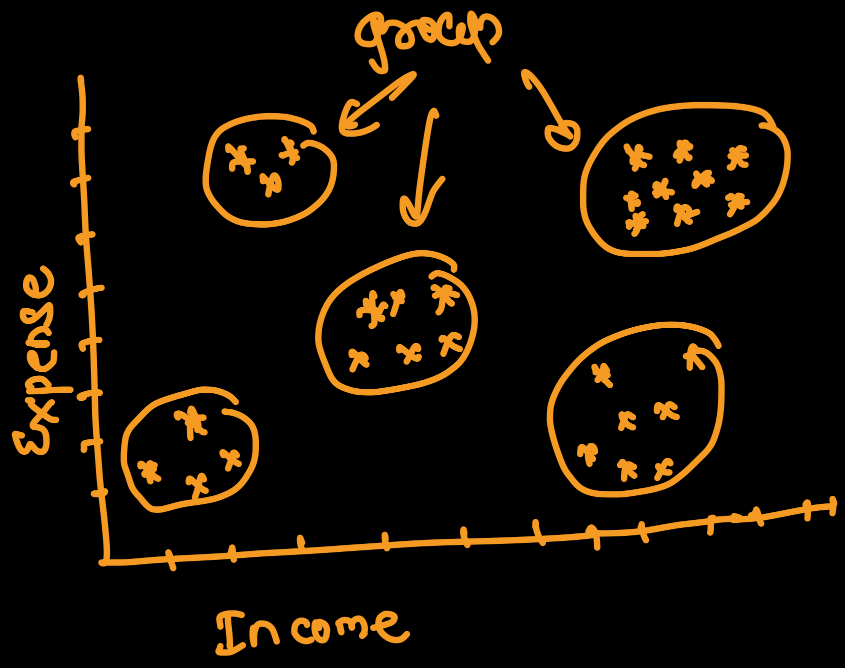
> category / class

Unsupervised Learning → • No prediction → unlabelled data ↳ No label

- Unsupervised learning is where you only have input data (X) and no corresponding output variables
- The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data → Exploratory Data Analysis
- These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher
- Algorithms are left to their own devises to discover and present the interesting structure in the data

Unsupervised Learning - Problems

- **Clustering** → find out clusters / groups
  - discover the inherent groupings in the data, such as grouping customers by purchasing behaviour
  - E.g.
    - Batsman vs bolwer
    - Customer spending more money vs less money
  - Algorithms
    - K-means clustering
    - Hierarchical clustering



Unsupervised Learning - Problems

- **Association Rules Mining**
  - An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y
  - E.g.
    - Market basket analysis
  - Algorithms
    - Apriori
    - Eclat

name	price
butter	24
milk	70
shampoo	10

Unsupervised Learning - Problems

Dimension = column = variable

- Dimensionality Reduction
  - Used to reduce the dimensions of the dataset to reduce the resource requirements, visualize the result, get the results in short time
  - Algorithms
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - Independent Component Analysis (ICA)

Reinforcement Learning → No data → Experience

- It is about taking suitable action to maximize reward in a particular situation
- It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation → auto driving vehicles
- Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task
- In the absence of training dataset, it is bound to learn from its experience

Reinforcement Learning

- Examples
  - Resources management in computer clusters
  - Traffic Light Control
  - Robotics
  - Web system configuration
  - Chemistry
- Algorithms
  - Q-Learning
  - Deep Q-Learning

End to End Process

Steps

- Look at the big picture
- Get the data
- Discover and visualize the data to gain insights
- Prepare the data for Machine Learning algorithms
- Select a model and train it
- Fine-tune your model
- Present your solution
- Launch, monitor, and maintain your system

Look at the Big Picture

- Frame the Problem
  - The first question to ask your boss is what exactly the business objective is
  - Building a model is probably not the end goal
  - How does the company expect to use and benefit from this model?
  - Knowing the objective is important because it will determine
    - how you frame the problem
    - which algorithms you will select
    - which performance measure you will use to evaluate your model
    - how much effort you will spend tweaking it
- Select a Performance Measure
  - Your next step is to select a performance measure
  - A typical performance measure for regression problems is the Root Mean Square Error (RMSE)
  - It gives an idea of how much error the system typically makes in its predictions, with a higher weight for large errors

Get the data

- Decide the data source
- Download the data and make it available for the further learning
- Take a Quick Look at the Data Structure
  - Understand the data set and features
  - Evaluate the features and decide which one(s) are needed
- Create a Test Set
  - Keep some records aside for testing and validation

Discover and Visualize the Data to Gain Insights

- Visualize the data
  - Use libraries like matplotlib or seaborn
  - Understand the pattern and relationship
- Look for correlation
- Experiment with attribute combinations

Prepare the Data for Machine Learning Algorithms

- Data Cleaning
  - Process of cleaning the data set to prepare it for ML algorithm
  - Steps
    - Check for the missing data
    - Check for wrong data types
    - Add features if needed
    - Remove unwanted features
- Feature Scaling
  - ML algorithms don’t perform well when the input numerical attributes have very different scale
  - Scale the features to bring all of them to a single scale
- Handle categorical / text data
  - Use transformers to convert categorical to numerical

Select and Train a Model

- Training the model using train data set
  - Create a model using selected algorithm
  - Save the model for future use
- Evaluation the model
  - Evaluate the model to see if there is any chance to improve the accuracy
  - Techniques
    - Cross Validation

Fine-Tune Your Model

- Grid Search
  - One option would be to fiddle with the hyperparameters manually, until you find a great combination of hyperparameter values
  - This would be very tedious work, and you may not have time to explore many combinations
  - You can also automate this process using libraries like sci-kit
- Randomized Search
  - The grid search approach is fine when you are exploring relatively few combinations
  - But when the hyperparameter search space is large, it is often preferable to use randomized search
- Ensemble Methods
  - Another way to fine-tune your system is to try to combine the models that perform best
  - The group (or “ensemble”) will often perform better than the best individual model, especially if the individual models make very different types of errors.
- Analyze the Best Models and Their Errors
- Evaluate Your System on the Test Set

Launch, Monitor, and Maintain Your System

- Deploy the application for the end users
- Monitor the application's performance
- If the data keeps evolving, update your datasets and retrain your model regularly
- You should probably automate the whole process as much as possible
  - Collect fresh data regularly and label it
  - Write a script to train the model and fine-tune the hyperparameters automatically. This script could run automatically, for example every day or every week, depending on your needs
  - Write another script that will evaluate both the new model and the previous model on the updated test set, and deploy the model to production if the performance has not decreased (if it did, make sure you investigate why)

Summary

Data Collection
Data Modelling
Deployment
What problem are we trying to solve ?
Problem definition
What data do we have ?
Data
What defines success ?
Evaluation
What features should we model ?
Features
What model should we use ?
Modelling
What have we tried/ what else can we try ?
Experiments