Introduction to AI

1. **Basics of A.I**

What is Intelligence ?

The ability to learn, the ability to recognize problems, the ability to solve problems

What is Artifical Intelligence ?

A.I is when we artificially introduce intelligence in machines thereby giving us:

- a machine which mimics human like intelligence
- a machine which has decision making capabilities
- a machine which learns on its own

Artificial Intelligence versus conventional programming

Conventional programming	Artificial Intelligence
Programmers look at the problem (desired output) and build an algorithm/application to solve this problem.	Ai programmers show the problem (desired output) to the A.I algorithms and expect the algorithms to find a solution.
A programmer has complete control over their application	An A.I programmer can never claim to have full control over their A.I applications. (Explainable A.I is hard to achieve)
The software must follow a logical series of steps to reach a conclusion (hard coded instructions by the programmer)	Ai applications use the technique of search and pattern matching
Its easy to explain a conventional algorithm	Its very hard to explain how an A.I algorithm reached its desired output
The most important element here is the algorithm	The most important elements here are data and algorithms

The 4th and the biggest industrial revolution

1784-water&steam

1870-electricity

1969-automation

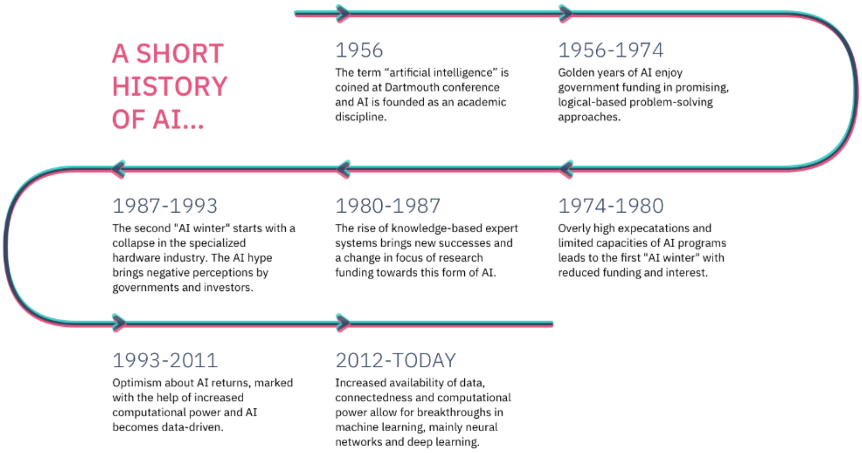
Now or future-cyber-physical systems

History of A.I – important points

Efforts to create intelligent machines started as early as 1642: First mechanical calculating machine – Blaise Pascal

Turing test was introduced in 1950

In 1955 we coined the term, “Artifical Intelligence"



Question: Can you name a company which is purely an AI company ?

A.I helps other software services do their job better. e.g:

- Voice recognition and intelligent search used in A.I assistants
- Natural Language processing used in Translation services

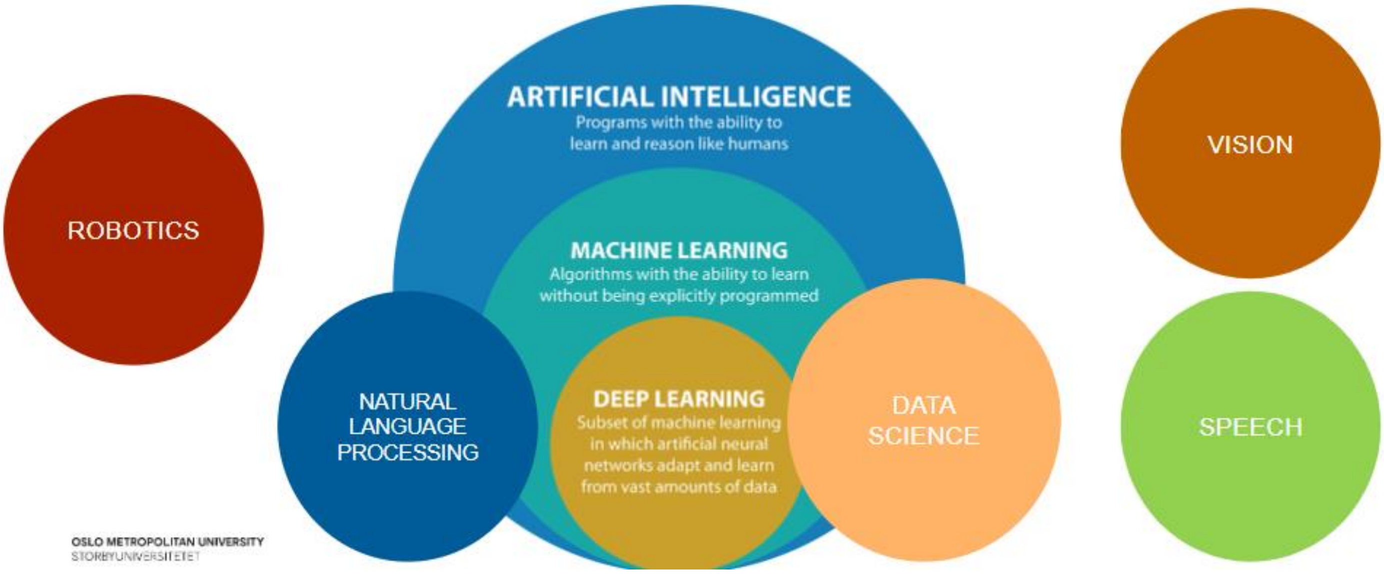
3 types of AI

Narrow: Dedicated to assist with or take over specific tasks.

General: Takes knowledge from one domain, transfers to other domain

Super: Machines that are an order of magnitude smarter than humans

Branches of A.I



Skillset needed to work with A.I

Math: the theoretical background necessary to conduct and apply AI research

Statistics: empirical skills needed to fit and measure the impact of AI models

Machine Learning: skills needed to build self learning models like deep learning and other supervised models that power most AI applications today

Statistical Programming: programming skills needed to implement AI models such as in python and related packages like sci-kit learn and pandas

Software Engineering: programming skills needed to design and scale AI powered applications

A.I buzz created a lot of negativity

- FOMO: Fear of missing out
- FUD: Fear, uncertainity and doubt
- Feuds: When people with their knowledge of A.I fight with each other

A.I buzz is also helping to solve many problems: Human trafficking, Money Laundering, Terrorism, Covid19 research, etc ..

2. History of Artificial Intelligence Part 2

The birth of Artificial Intelligence: 1956 Dartmouth Conference

The first attempts in AI

• Hardware and software continue evolving.

• Sometimes taking inspiration from biology

• Sometimes taking inspiration from psychology

• Logic Theorist is regarded as the first AI software 1956

• Using a machine that operates numbers to operates symbols

• Symbol operation as a model for thinking

• Proved the first 38 of the 52 theorems of the principia Mathematica

• Frank Rosemblatt: Developed an electronic device following biological principles that shown the capacity to learn

Alexey G. Ivakhnenko and multilayered neural networks

1960's from general to narrow AI
• 1960 John Hopkins Beast bot able to wanter and recharge
• 1960's Stanford Shakey navigation by vision
• 1961 the first robot was introduced in General Motors assembly line
• 1965 The first chat-bot ELIZA was invented by Joseph Weisenboun using techniques from Rogerian psychotherapy.
• 1967 First full-scale humanoid robot (WABOT) professor Ichiro Kato of Waseda University
• 1970 First expert systems Stanford AI Laboratory Cart
1980's The age of expert systems
• 1980 Revival of the AI research with the narrow focus of managing knowledge
• 1981 Japan announces a 850 millions investment in AI projects focused on translation, language understanding and reasoning
• 1980s Raise of video-game industry (forever to be used as test-bed for AI)
• 1989 The Carnegie Mellon Lab creates the first autonomous vehicle using a neural network
• Late 1980's Neuveau IA: True intelligence can only be developed if the machine has a body (how elephants do not play chess)

1990's a new winter for AI
Problems with the approach used for expert systems
• Poor adaptability— software improvement is hard and dependent on many people.
• Extreme brittleness—The system will fail in situations that weren't part of the original design.
• Tough to maintain—The complexity of such a system is huge. When thousands of rules are put together, improving it or changing it is incredibly complicated, slow, and expensive.
• In sum. They did not scale well.

1995 German autonomous vehicle Munich to Copenhagen

1999 Aibo and MIT Emotional AI Lab

Honda P-series continues evolving

Machine Learning brings a new spring to AI in the 2000s

First anti-spam 2002

2004 Darpa Autonomous Vehicle Challenge

2008 IBM Watson Jeopardy

Natural Language Generation: Writing reports

Analyze: Identify facts and determine what is important and interesting

Generate: Automatically generate data-driven narratives to desired specifications

Inform: Easily share information in a readable format at scale

AI Governance
PERFORMANCE: ACCURACY, BIAS, COMPLETENESS

SECURITY: ADAPTABILITY, ADVERSARIAL ROBUSTNESS
PRIVACY: IP CAPTURE, IMPACTED USERS
TRANSPARENCY: EXPLAINABILITY, INTENT

- Artificial Intelligence that is developed and used in Norway should be built on ethical principles and respect human rights and democracy
- Research, development and use of Artificial Intelligence in Norway should promote responsible and trustworthy Artificial Intelligence
- Development and use of Artificial Intelligence in Norway should safeguard the integrity and privacy of the individual
- Cyber security should be built into the development, operation and administration of systems that use Artificial Intelligence
- Supervisory authorities should oversee that Artificial Intelligence systems in their areas of supervision are operated in accordance with the principles for responsible and trustworthy use of Artificial Intelligence

Buzz words

Artificial Intelligence	**Data Science**
Automates tasks or predicts future events based on data	Produces insights based on data
Is commonly used “live”: it continuously elaborates new data and produces answers.	Is commonly “one-off”: it produces some insights that inform decisions.
It commonly has the form of software	It commonly has the form of a presen- tation or report

1. What has changed between the 1950s and now?
2. Is there something that could cause a new AI winter?
3. What sort of technologies could we expect in the future?
4. Would the machines be able to understand?

3. DATA

Steps to design an A.I system

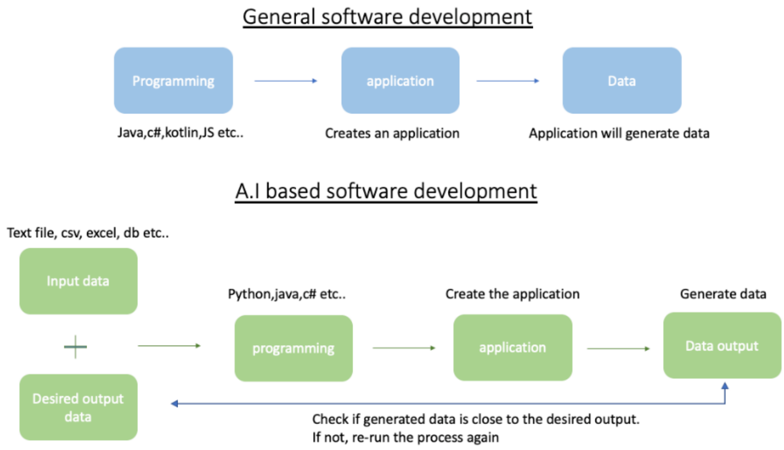
1. Identify the problem
2. Prepare the data
3. Choose the algorithms
4. Train the algorithms with the data
5. Run on a selected platform

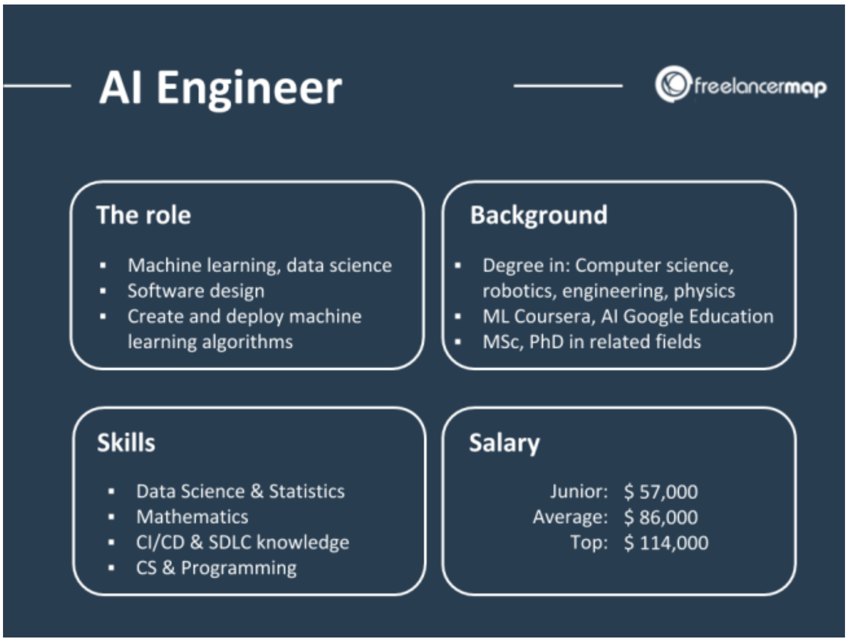
A.I is about algorithms, tools and data

Frameworks, Programming languages and public or private data

Daily life of an A.I programmer

- 80% time spend on data (cleaning, preparing, labeling, analyzing etc)
- 5% on deployment (cloud/on premise)
- 15% on A.I development





Data pitfalls (problems which can occur with data)

Assuming the data is clean: spelling mistakes, ...

Outliers: Excluding outliers, Including outliers

Ignroing seasonality: Easter vacations, summer holidays, black Friday etc.

Context is critical: Ignoring size when reporting growth

Poor data Insights

Not connecting with external data

Lacking business understanding

How to work with data ?

1.Data labeling / annotation

2.Data annonimization

3.Synthetic data

4.Data preparation: Data cleansing + feature engineering: Feature engineering, also known as feature creation, is the process of constructing new features from existing data to train a machine learning model.

F.ex.: Character recognition: features may include histograms counting the number of black pixels along horizontal and vertical directions, number of internal holes, stroke detection and many others.

Speech recognition: features for recognizing phonemes can include noise ratios, length of sounds, relative power, filter matches and many others.

Spam detection: features may include the presence or absence of certain email headers, the email structure, the language, the frequency of specific terms, the grammatical correctness of the text.

Computer vision: there are a large number of possible features, such as edges and objects.

5.Data wrangling: The process of transforming and mapping **data** from one "raw" **data** form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics.

Tools: Python, Excel, Tablua, Google data prep

6.Data mining: The process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Process: Collect data, manage, prepare, sort, present

7.Data warehousing: ETL programming

8.Data Engineering: Infrastructure, big data, cloud etc..

What goes into a successful model:

EXPLORATORY ANALYSIS (10%)

DATA CLEANING (20%)

FEATURE ENGINEERING (25%)

ALGORITHM SELECTION (10%)

MODEL TRAINING (15%)

OTHER (20%)

Data Science vs Artificial Intelligence

Factors	Data Science	Artificial Intelligence
Scope	Involves various underlying data operations	Limited to the implementation of ML algorithms
Type of Data	Structured and unstructured	Standardized in the form of embeddings and vectors
Tools	R, Python, SAS, SPSS, TensorFlow, Keras, Scikit-learn	Scikit-learn, Kaffe, PyTorch, TensorFlow, Shogun, Mahout
Applications	Advertising, Marketing, Internet Search Engines	Manufacturing, Automation, Robotics, Transport, Healthcare

4. Core Business Data and Supervised Machine Learning

Core business data is the kind of data with the strongest ties to the value- generating engine of an organization. Therefore, it also holds the highest potential for impact by AI.
ϒ Supervised learning is a family of machine learning algorithms that allows computers to learn how to map inputs (features) and outputs (labels), given enough examples.
ϒ Machine learning can tackle many hard problems where conventional software engineering fails, because it's based on historical data rather than mathematical understanding.
ϒ Even if ML-based models can never be 100% accurate, numerical metrics (such as accuracy) can help track their performance.

STRONG/ GENERAL AI: Artificial Intelligence is any trait exhibited by machines that is considered similar to traits of human intelligence.

WEAK/ NARROW AI: Any device that perceives its environment and takes actions that maximize its chances to successfully achieve a goal.

We are departing from the realm of general AI. Too much hype and high expectations generated mistrust on the idea that machines can actually be as intelligent as us in so many different fields.
Instead, let's focus into something that can deliver more immediate value and is easier to understand. The weak AI.
GENERAL AI is Occupied with the greater goal of reproducing and understanding intelligence. A kind of psychology (in Sylico) from the engineering perspective. Nowadays it is mostly relevant for basic research.
NARROW AI are systems able to perceive a narrow segment of the world, perform operations and deploy decisions or actions in response to them. Essentially, it is automation of decisions. Narrow AI is more relevant for applied research and business, as it is part of almost everything we do today, even in things you do not think as AI.

There are 3 broad use-case areas for weak Artificial Intelligence in business

Task automation	Insight generation	People engagement
Assembly lines	Business Insight	Entertainment
Expert systems	Data analytics	Intuitive user interface
Fraud detection	Pattern recognition	Social reward
Anti-spam	Sciences	Click baiting/ phishing
Image classification	Traffic control	Chat-bots
Self-driving vehicles	Churn analysis	Voice assistants
Fast trading		Face recognition

It became cheap to store data
One of the previous limitations for machine learning was the lack of methods to store data. Machine learning is very data dependent and its only in the past 20 years that we've really had the technology and data to fully utilize it.

Big Data

Variety: An expanding universe of data and sources (Structured, semi-structured, unstructured and mixed)

Velocity: How fast data is traveling or being processed (the speed of datageneration. streaming and operation)

Volume: Amount of data in terms of the potential of terabytes to petabytes and up

Veracity: Data reliability and trust verifying and validating data

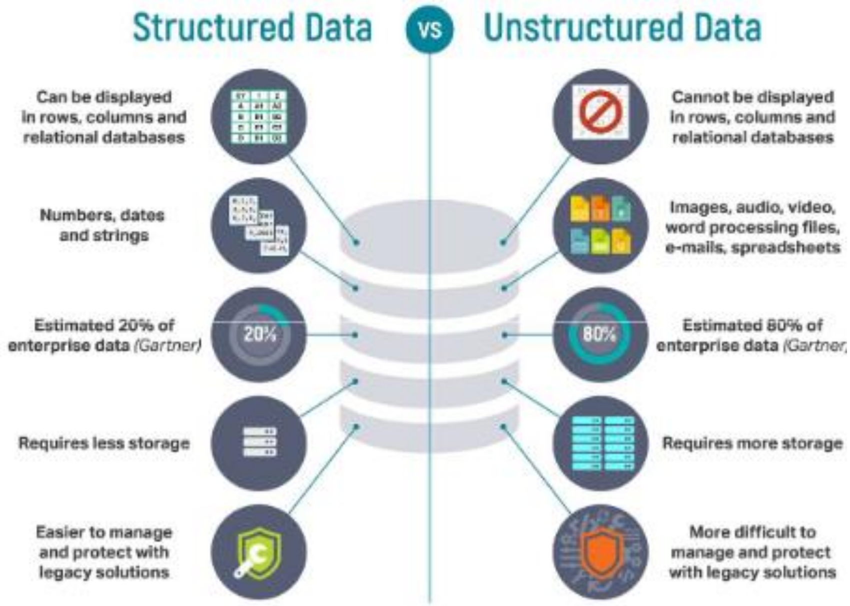
Value: Usefulness of data in decision-making i.e.the number or count does not matter but the insight does

"Core Business Data" is the data closest to the value proposition of the business.
It describes events and patterns that have a direct impact on the organization's performance, and it is easy to attach monetary value to it.
Machine learning emphasizes on learning from data. But what does it mean?

Money density of data estimates how much data influences the top or bottom line of the organization
• Data that is actionable and can lead to reduction of costs or increase in profit.
• Core business data has high money density value
• Changes are that there is a lot of core business data available as: transactions, reports, reviews, complaints, opinions, tables, videos and images, phone calls
• Behaviors talk louder than opinions.
• It is the prime candidate for AI applications as any improvement in performance is almost guaranteed to make a dramatic impact
Dollar density of data: how much the data influences the top or bottom line of the organization. Core business

data has a high dollar density: each e-commerce order, job lead, or financial transaction has a direct impact on your top or bottom line.

Structured data is essentially data that was properly organized for data analysis. Which is essentially to put it into some form of table based database Unstructured data is pretty much everything else. Videos, texts, audios. Things that contain information but do not have a clear way to decode it. Although much value can be taken from structured data, most of the information is in the unstructured form. And we should find a way of extracting value from it.



So how AI can help?

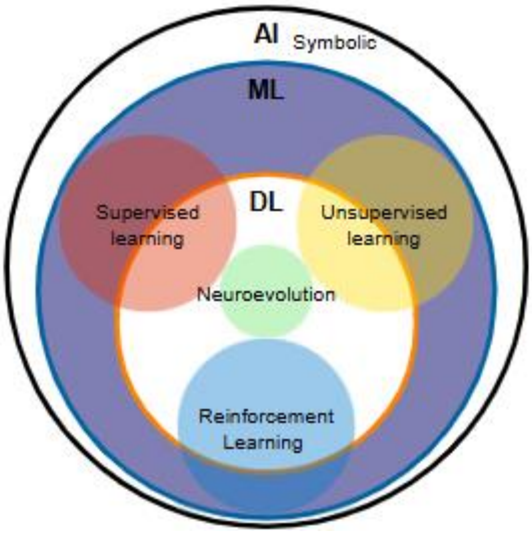
A machine has artificial intelligence if it can interpret data, potentially learn from the data and use that knowledge to take decisions, adapt to its environment, potentially generate novel behavior and achieve specific goals.

Different kinds of AI

AI: Artificial Intelligence
Symbolic AI (GOFAI): Separation between logic and data, programmed rules, Fuzzy logics
ML: Machine Learning
Supervised Learning: Learn from examples, requires structured and labeled data
Unsupervised Learning: Learn patterns and groups in unlabeled data
Reinforcement Learning: Learn from experience, trial and error, reward policy
DL: Deep Learning
Generative Algorithms: Exploit mutation and selection to find a good solution to the problem at hand

It seems that the solution for an explosion of data is not symbolic ai, but some technique that allows the machine to learn from the data without being explicitly programmed.
Enters...
Machine learning We also mentioned many different techniques and types of Artificial Intelligences to solve different types of tasks. Each of them has different characteristics that must be understood in order to apply them to the adequate problem.
To understand them means:
- To know what they do
- Their advantages and disadvantages
- What problems they solve

In this diagram we have an overview of the methods discussed
In the outer circle with have all that may be considered AI
In that outer circle we have the Symbolic AI or Good Old fashion AI, which is present in almost everything digital today. The blue circle represents all the AI systems that are able to learn . Machine learning
Inside of it we have the 4 main kinds of machine learning
- Supervised learning - or learning from examples
- Unsupervised learning - or learning to identify groups and patterns
- Reinforcement learning - or learning from experience, trial and error



- Genetic algorithms - adapt with mutation and selection
Machine learning can be done with classical method or with Neural Networks, now known as deep learning
All the kinds of machine learning can leverage deep learning and its ability to extract features from unstructured data. If these words do not mean much, do not worry, we will eventually explain them. Just be aware that deep learning is just one special way for doing machine learning.

Symbolic AI based on high level symbolic, representation of problems, logic and search. It does not require massive amount of data or symbols. Instead, the world and objects in the world are represented as symbols. These symbols are then manipulated using logic to search for solutions.
Logic is the problem solving technique and symbols are how we are going to represent the problem in the computer. Symbols can be numbers, texts objects, etc.
A relation can be an adjective that describes a symbol, and we write it in front of the symbol that is inside the parenthesis
A relation can also be a verb that describe how a symbol interacts with other symbols EAT(GUSTAVO, DONUT)
Symbols = nouns
Relations = Adjectives or verbs
Logic = AND, OR, NOT
The collection of all symbols in a software (the representation of the entire world) is called a knowledge base (HOW SIRI WORKS)
Knowledge base can be combined to build propositions
We can test if these propositions are true or not
With tools of propositional logic called truth table
It converts each statement as 0 or 1 AND as multiplication and OR as addition
If the proposition evaluates true, than it is true
IMPLICATIONS connects one proposition to the next
(IF/THEN)
AN implication is true if the THEN side is true or the IF side is false
We can start build a knowledge base of all the propositions that are true in our knowledge base. With that we can use the AI to answer questions and discover new things. You can provide a set of true propositions and use logic operations for the computer to search for new true propositions without the human giving them explicitly.
This is INFERENCE (sandwitch discount example)
Expert systems
https://www.youtube.com/watch?v=WHCo4m2VOws
Expert systems can make conclusions based on logic and reason, not just trial and error guesses like neural network
Expert systems can explain its decision by showing which parts were evaluated as true or false. GOOD OLD FASHION AI
PROBLEMS:
DIfficult to scale
Not easy to represent all forms of data
Some things are not clear-cut deterministic such as true or false
Symbols are not static, in time, symbols change as well as their relations.
FUZZY LOGIC way of dealing with uncertainty
Not 0 or 1 but values between 0 and 1
Hot cold

Car break
Uncertainty
THERE IS A SET where things within it are true, outside of it are false
THEY ARE USED EVERYWHERE, they are just now embedded in what we use in normal devices
Expert systems Hot shower, washing machine
The Sendai Subway 1000N series (仙台市交通局1000N系電 車) is a rapid transit electric multiple unit (EMU) train type operated on the Sendai Subway Namboku Line in Sendai, Japan.
https://www.youtube.com/watch?v=r804UF8Ia4c

The 1000 series was the world's first train type to use fuzzy logic to control its speed, and this system developed by Hitachi[1] accounts for the relative smoothness of the starts and stops when compared to other trains, and is 10% more energy efficient than human-controlled acceleration.[2] It was the recipient of the 28th Laurel Prize award presented by the Japan Railfan Club

Learning are changes in behavior as result of experience
• Changes in behavior that do result from lesion are not learning
• Learned behaviors are functionally connected to the experience they were acquired
• Good learning leads to increase of performance towards achieving a goal
• Not all learning increases performance, some reduce performance (dysfunctional or pathologic behavior)
• Machines learn by creating models from data
Machines learn by creating models from the data.
There are many kinds of learning

Machines learn by creating mathematical models of the world from data

• Models are simplified descriptions of an observation or a process
• Like a map is to a territory
• It highlights important and relevant information
• It ignores most of irrelevant information and some of the important too
• There are many kinds of models:
• Descriptive
• Predictive / retrodictive
• Mechanistic
• Normative

Descriptive models - describe what you observe now in a simple way. It is a form of removing complexity
Predictive models and retroactive models - look for correlations and trends in the data to point out how data would look like if the trend continues in the future or in the past
Mechanistic models - describes the process or phenomenon in terms of its parts and causes. This can only be developed through carefully designed experiments
Normative models - describe optimal strategies. In other words, given a problem, what is the optimal way to solve it?

Building such a set of instructions is called programming.
To do so, you need some fundamental elements like
Values: that can describe the world for you. Using Names for categories of things and Numbers to express magnitudes and amounts.
Variables: to hold in memory the values while you perform the operations and actions
Actions: ways to transform values, like adding , subtracting…
Decisions: ways of comparing values and take one set of actions or the other depending on what is observed
Repetitions: ways to repeat a set of actions and value transformations many times
And Abstractions: That can encapsulate many small detailed actions into a set of actions that can be understood as an action.
Mathematical models start from data
Lets say we have one variable Weight. Just so you know, a variable is any named feature or characteristics which value may vary, depending on the observation in the data set.
In a structured data set (or table) values are organized in this way typically. Observations as rows (that would be individuals, time, sales, etc) and the columns are the different variables or features. How you name your observations or your variables is irrelevant.
But for the sake of example, lets say we have 4 people, with their names properly anonymizes. And their respective weights.
Mathematical models can be used to express relationships between variables
• The machine goes about finding the parameters that describe the function of a line that “bets fits” to the data
• This line is a linear function, and it is often referred as “the model” or “the hypothesis”
• The process of finding the line is called “line fitting”, “model fitting”, “linear regression” or more broadly “model training”

• Regressions are very useful to generate predictions (often referred as inference)
• Regressions are also useful in data interpolation (estimating missing values)
In regression we are often trying to estimate the value of one value from another.
To do that, machines need to learn from actual examples. So we use some known data, with known relationships so machines learn from it This is called the training set.
Machines can learn to classify observations into groups
Collection of propositions is your knowledge base The Knowledge base can be combined with data to produce answers. For instance, that is one way how expert systems and search engines may work.
They have a set of rules pre-programmed on how to operate a search.
Depending on the data (the database and the input keywords) the algorithm will apply the rules to give you the result.
This works well for established and well defined processes that do not change much.
This is the fundamental part of expert systems.
Building such a set of instructions is called programming. To do so, you need some fundamental elements like
Values: that can describe the world for you. Using Names for categories of things and Numbers to express magnitudes and amounts.
Variables: to hold in memory the values while you perform the operations and actions
Actions: ways to transform values, like adding ,subtracting…
Decisions: ways of comparing values and take one set of actions or the other depending on what is observed
Repetitions: ways to repeat a set of actions and value transformations many times
And **Abstractions**: That can encapsulate many small detailed actions into a set of actions that can be understood as an action.
Classification requires labeled data. One label per category
In a structured data set (or table) values are organized in this way typically. Observations as rows (that would be individuals, time, sales, etc) and the columns are the different variables or features. How you name your observations or your variables is irrelevant.
Suppose you wanted to target a promotion for a new digital camera and would like to know which of your customers are likely to buy the camera. Classification is a data mining technique that is useful for this application. Classification divides data into two or more well-defined classes. Unlike clustering in unsupervised learning, where you do not know which groups will be generated, in classification you know exactly what each group represents. In the previous example, the two groups are: customers who are likely to buy a camera and customers who are not likely to buy a camera. This is an example of supervised learning.
In order for classification to work well, the build data must contain enough samples for each target category; otherwise, it may not be accurate. In other words, your build data must include enough people who have bought digital cameras in the past and enough who have not.
A labeled dataset allows you to classify different data points into groups, such as diabetic or healthy. Then based on the parameter you are assessing, here glucose levels, you can generate a model that distinguishes between the groups based on the parameter. Essentially this is drawing a line along the dimension of the parameter which divides the inputs into their labeled groups. This is the training step of the model, where you use data where you already know what groups the different parameters correspond to and train the model to generate this distinction.
Once the model is trained and you have a way to classify you can apply a test dataset which is also labeled to see how well the model performs. Based on the performance during the test set you can adjust the model to better distinguish between the groups.
Multiple features can be used to classify the data
Only once the model is both trained and tested can you apply it to new data. As you can see this all requires a lot of structured and labeled data.
Data mining using classification usually involves a testing phase to check how good the model is. For this, data where the outcome is known is tested to see how well the model's predictions match it. For instance, you would take data for customers who have bought a digital camera in the past and check it against the predictions given by the model.
Testing a model involves computation of a structure known as the confusion matrix. A confusion matrix tells you how many times the model's prediction matched the actual data and how many times it did not. The columns correspond to the predicted values and the rows to the actual values. For instance, in
In classification, you first analyze a small part of your data to build a model. For instance, you would analyze real data for people who have bought digital cameras and people who have not bought digital cameras, over a given time period. The data used to build a model is known as build data. The model will be built taking into

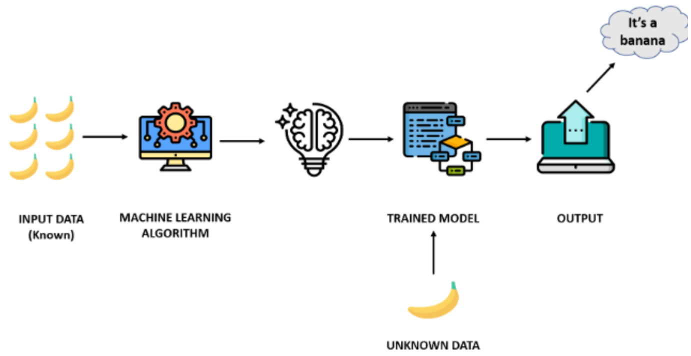
account various factors, such as age, income, and occupation, that are known to influence people's buying habits. These factors are known as predictor attributes. The output that is predicted is called the target attribute and its values (whether the person will buy the camera or not) are known as categories or classes.
Once the model has been generated, it can be applied to other data to come up with a prediction. This is known as model apply, or scoring. In our example, you would use the model to predict whether a certain customer is likely to buy a digital camera.
In the previous example, the target attribute has two values: will buy a digital camera and will not buy a digital camera. You can also use classification to predict attributes with more than two values—for example, whether the risk of a person defaulting on a payment is low, medium or high.
Classification is often used to create customer profiles. For instance, once you have determined which of your customers are likely to buy a digital camera, you can then profile them by occupation, as shown in Figure 16.3.
From this graph, you now know that most likely buyers are either engineers or executives. So you can now target your promotions more accurately toward these customers and reduce your costs.

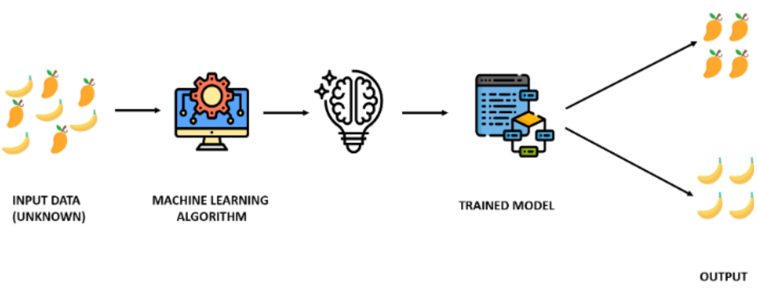
EXERCISES
- Identify use-cases, activities, work processes or task that you believe that could employ supervised machine learning
- Describe the data that you have available:
- What structure data you have there?
- What unstructured data you have there?
- What kind of supervised machine learning methods would you use for every task? Classification or Regression?
- What is the insight you want to get from the data?
- What are the challenges you anticipate?

Machine Learning
Statistics is a traditional field, broadly defined as a branch of mathematics dealing with data collection, organization, analysis, interpretation and presentation.
Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
Example: Search for cats on google. How does this search work ?
- Google first gets a large quantity of examples of photos labeled “cat”
- Then the Machine learning algorithm looks for patterns of pixels and patterns of colors that will help it predict if the image is of “cat”.
- At first, Google’s computers make a random guess of what patterns are good in order to identify an image of a cat.
- If it makes a mistake, then a set of adjustments are made (by humans) in order for the algorithm to get it right.
- In the end the algorithm will learn such patterns and improve its output

Q. How would you build such a M.L algorithm→
1. Collect data
2. Label the data
3. Train the machine learning model with data: The model will look for patterns in the images in order to identify cats and lions
4. Once the model is trained, we give it different input images to test and see if the model gives us the right answer

Types of Machine Learning algorithms
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Recommender systems





Supervised learning Unsupervised learning

Reinforcement Learning is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. Whenever the model predicts or produces a result, it is penalized if the prediction is wrong or rewarded if the prediction is correct.

Challenges with Reinforcement learning

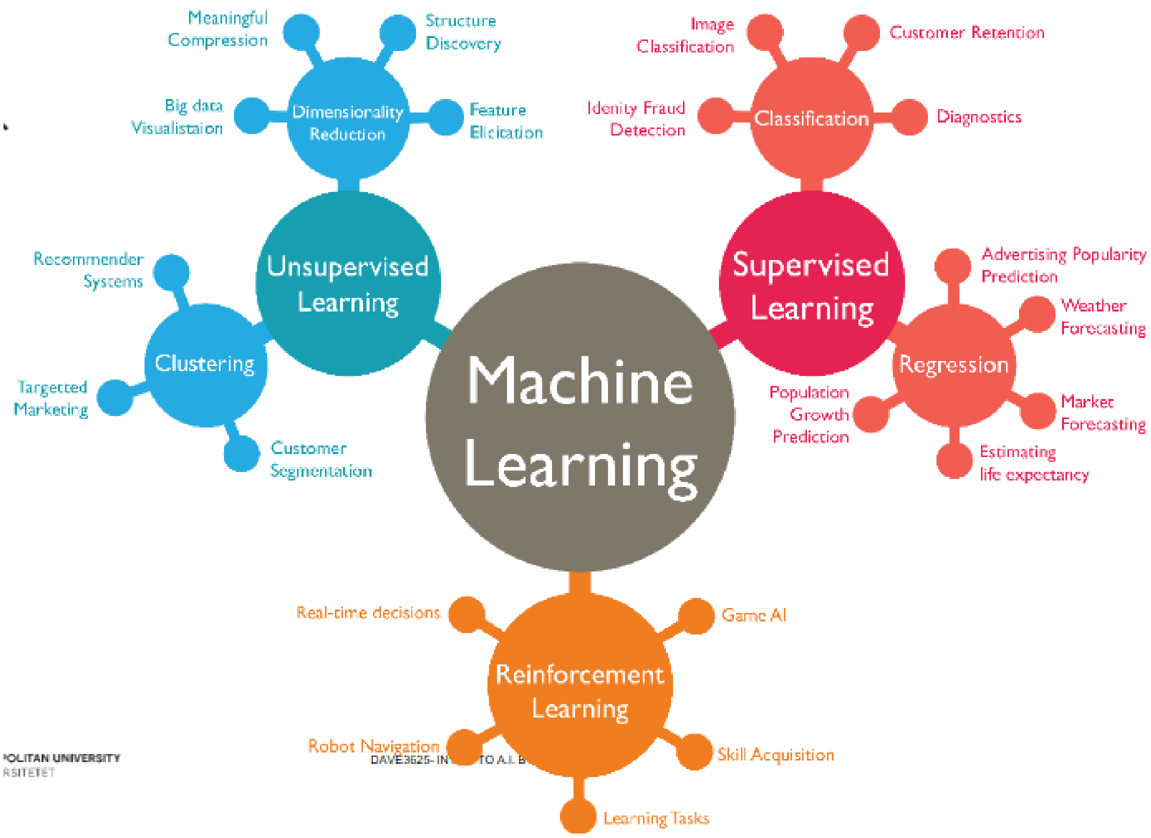
• The main challenge in reinforcement learning lays in preparing the simulation environment, which is highly dependant on the task to be performed.

• Scaling and tweaking the neural network controlling the agent is another challenge. There is no way to communicate with the network other than through the system of rewards and penalties.

There are agents that will optimize the prize without performing the task it was designed for.

Recommender systems are an important class of machine learning algorithms that offer "relevant" suggestions to users.

These systems predict the most likely product that the users are most likely to purchase and are of interest to them



When should we use machine learning?

Hand-written rules and equations are too complex—as in face recognition and speech recognition.

The rules of a task are constantly changing—as in fraud detection from transaction records.

The nature of the data keeps changing, and the program needs to adapt—as in automated trading, energy demand forecasting, and predicting shopping trends.

Supervised Machine Learning

In a supervised learning model, the algorithm learns on a labeled dataset, providing an answer key that the algorithm can use to evaluate its accuracy on **training** data.

How to work with Supervised machine learning

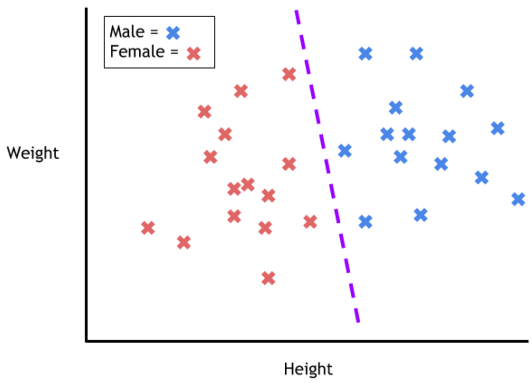
- Data collection
- Data processing
  - Data cleaning
  - Data labeling
  - Feature Engineering
  - Feature scaling
  - Data splitting (training, validation and test sets)
- Model Selection
- Training
- Validation
- Evaluation (with test data)
- Deploy the model
- Monitoring and mantainance

Supervised machine learning involves training a model on a set of data where the "answers" (labels) are known, so the model can make predictions on new, unseen data. Here's a step-by-step breakdown of the process:

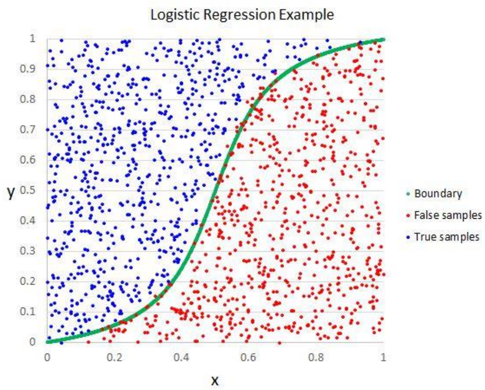
1. **Problem Definition**:
   1. Clearly define the problem you want to solve.
   2. Decide if it's a regression (predicting a continuous value) or classification (predicting discrete categories) problem.
2. **Data Collection**:
   1. Gather data relevant to your problem. This could involve using existing datasets or collecting your own data.
3. **Data processing**:
   1. **Data Cleaning**: Handle missing values, outliers, and any noise in the data.
   2. **Feature Engineering**: Extract, transform, or combine features to better represent the underlying patterns in the data.
   3. **Feature Scaling**: Normalize or standardize features so they have similar scales. Common methods include Min-Max scaling and Z-score normalization.
   4. **Data Splitting**: Divide your data into training, validation (optional), and test sets. The training set is used to train the model, the validation set to tune hyperparameters, and the test set to evaluate the model's performance.
4. **Model Selection**:
   1. Choose one or more machine learning algorithms suitable for your problem. For instance, you might choose decision trees, linear regression, or neural networks.
5. **Training**:
   1. Feed your training data (both the features and labels) into the chosen algorithm. The model will "learn" from this data.
   2. Adjust the model's internal parameters to minimize the difference between its predictions and the actual labels in the training data. This difference is often called "error" or "loss".
6. **Validation and Hyperparameter Tuning** (if you have a validation set):
   1. Test the trained model on the validation set to see how it performs.
   2. Adjust hyperparameters (settings of the algorithm that aren't learned from the data) based on validation set performance to find the best model.
   3. Iterate between training and validation until you get the best performance.
7. **Evaluation**:
   1. Once the model is trained and tuned, test it on the separate test set to evaluate its performance.
   2. Use relevant metrics like accuracy, precision, recall, F1-score (for classification), or mean squared error, R^2 score (for regression).
8. **Deployment**:
   1. If the model's performance is satisfactory, deploy the model to a production environment where it can start taking in new data and making predictions in real-time.

9. **Monitoring and Maintenance**:
    1. Continuously monitor the model's performance in the real world. Models might need retraining or fine-tuning if the data they encounter in the wild is different from the training data or if the underlying patterns in the data change over time.
10. **Feedback Loop**:
- Incorporate feedback from the model's predictions in the real world to improve and retrain the model.

Model types (supervised learning)
- **Classification**
    - **Support vector machines**
    - **Decision Trees**
    - **K-Nearest Neighbour**
    - **Random Forest**
    - **Logistic regression**
    - **Naïve Bayes classifier**
    - **...**
- **Regression**
    - **Linear Regression**
    - **Polynomial Regression**
    - **...**



The job of a classification algorithm is to then take an input value and assign it a class, or category, that it fits into based on the training data provided.
The most common example of classification is determining if an email is spam or not. With two classes to choose from (spam, or not spam), this problem is called a binary classification problem. The algorithm will be given training data with emails that are both spam and not spam. The model will find the features within the data that correlate to either class and create the mapping function mentioned earlier: *Y=f(x)*. Then, when provided with an unseen email, the model will use this function to determine whether or not the email is spam.
Divides the data in classes / categories
Use cases:
Spam detection (classification: spam or normal)
Analysis of the customer data to predict whether they will buy computer accessories (classification: Yes or No)
Classifying fruits from features like colour, taste, size, weight (classification: Apple, Orange, Cherry, Banana)
Gender classification from hair length (classification: Male or Female)
Stock market price prediction (classification: high or low)
Classification versus Clustering algorithms
Clustering: Where we want to discover the groupings in data.
Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically.
- Classification is assigning labels
- Clustering is grouping data together
- Classification is supervised, clustering is unsupervised



Classification -> Logistic regression
is used when you have a classification problem - yes/no, pass/fail, win/lose, alive/dead, healthy/sick etc..
- It is the go-to method for binary classification problems
- The logistic function looks like a big S and will transform any value into the range 0 to 1.

Sigmoid function: Allows to put a threshold value. e.g 0.5 (use case spam detection)
Uses
- Logistic regression is used to predict the occurance of some event. e.g
    - Predict whether rain will occur or not
    - spam detection, Diabetes prediction, cancer detection etc.

Types of Logistic regression

- Binary logistic regression (e.g pass / fail)
- Multiclass logistic regression (e.g cats, dogs, sheep)
- Ordinal (low, medium, high)

Develop (using binary logistic regression)

Step1 : Visualization: 1. Say we're given data on student exam results and our goal is to predict whether a student will pass or fail based on number of hours slept and hours spent studying. We have two features (hours slept, hours studied) and two classes: passed (1) and failed (0).

Step 2: Sigmoid activation (a bit of maths): In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Step 3: Decision boundary: Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.

Step 4: Making predictions

Second stage: evaluate the algorithm

Evaluate using cost function

- What is a cost function ?
    - The cost function helps the learner to correct / change behaviour to minimize mistakes.
- In ML, cost functions are used to estimate how badly models are performing.
- Put simply, *a cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y.*

Cost function: Gradient descent

Gradient Descent is an optimization algorithm used to tweak parameters iteratively to minimize the cost function.

Gradient descent enables a model to learn the gradient or *direction* that the model should take in order to reduce errors (differences between actual y and predicted y).

Develop: Map probabbilities to classes

- The final step is assign class labels (0 or 1) to our predicted probabilities.

Final stage: Finish it up

- Train the model
- Evaluate the model
    - Minimzie the cost with repeated iterations
    - Measure the accuracy of your outputs
- Meausre the probabbility score

Classification -> Naive Bayes classification algorithm

- A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- e.g a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter.
- Its an easy to build algorithm and a very powerful one.

BAYES THEOREM

Bayes's theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Probability of B occurring given evidence A has already occurred

Probability of A occurring

$$P(\mathrm{A}|\mathrm{B}) = \frac{P(\mathrm{B}|A) \cdot P(A)}{P(B)}$$

Probability of A occurring given evidence B has already occurred

Probability of B occurring

Types of supervised learning

- Classification
    - Support vector machines
    - Decision Trees

        - K-Nearest Neighbour
        - Random Forest
        - Logistic regression
        - Naïve Bayes classifier
        - ...
    - Regression
        - Linear Regression
        - Polynomial Regression
        - ...

2. Naive Bayes classification algorithm (classification)

- A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- Naive bayes does not take into account the relationship between features

For example:

- A dog will be considered a dog because it has 4 legs and a tail
    - But a Naive Bayes will consider everything a dog which has 4 legs or tail (e.g squirrel) not because its an animal. Being an animal is the relationship not legs / tail.
- A fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter.

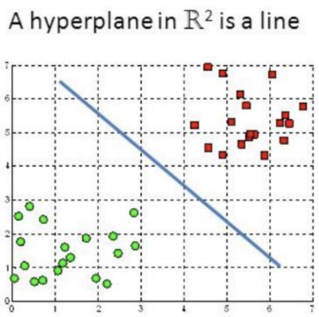
3. Support vector machines (classification)

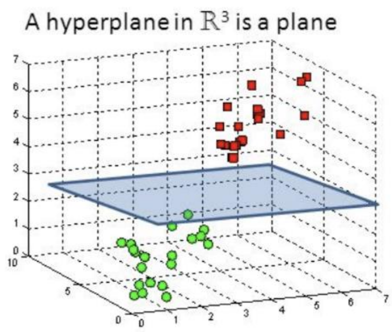
- Supprt vector machines classifies the data by finding a clear seperation between the data points
    - It looks for a hyperplane
- Can be used both in classification and regression

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence

Hyperplanes

- Hyperplanes are boundries that help classify the data points
- We can use a 2 dimensional hyperplane (if number of features is 2)
- or a 3 dimensional hyperplane (if number of features is 3)
- We can also have n dimensional hyperplane (for more features)





Another reason we use SVMs is because they can find complex relationships between your data without you needing to do a lot of transformations on your own.
It's a great option when you are working with smaller datasets that have tens to hundreds of thousands of features.

Uses of support vector machines

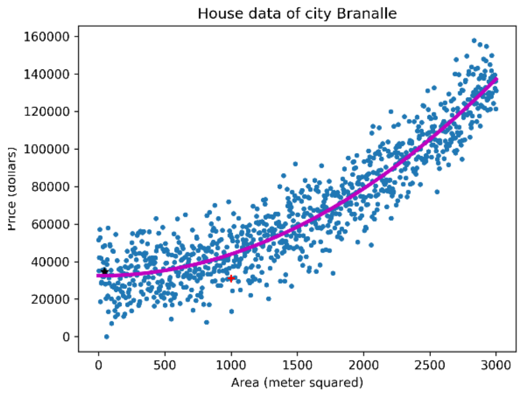
- Used to detect cancerous cells
- Used to predict driving routes
- Face detection
- Image classification
- Handwriting detection

Pros and cons

Pros
Effective on data sets with multiple features
Effective in cases where number of features is greater than the data points
Memory efficient
Cons
They donot provide probability estimates. Those are calculated using an expensive five fold cross vsalidation
Works best on small sample sets because of its high training time

Other classification algorithms

- Decision trees
- K nearest neigbour
- Random Forest
- ..



Regression

- Regression models are used to predict a continuous value
- **Examples**
    - Predicting prices of a house given a set of features (size, price, location etc)
    - Predicting sales revenue of a company based on previous sales figures
- Regression is used for prediction, forecasting, time series modeling, and determining the causal effect between variables
-

Question: Are the following use cases regression or classification ?

- Customer Churn: when a customer leaves the company -- Classification
- Stock market price prediction -- Classification
- Spam email -- Classification
- Prediction of price of an oil -- Regression
- Salary prediction -- Regression
- Age prediction -- Regression
- Gender prediction -- Classification

Regression Algorithm	**Classification Algorithm**
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) With the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) With the discrete output variable(y).
Regression Algorithms are used With continuous data.	Classification Algorithms are used With discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-Class Classifier.

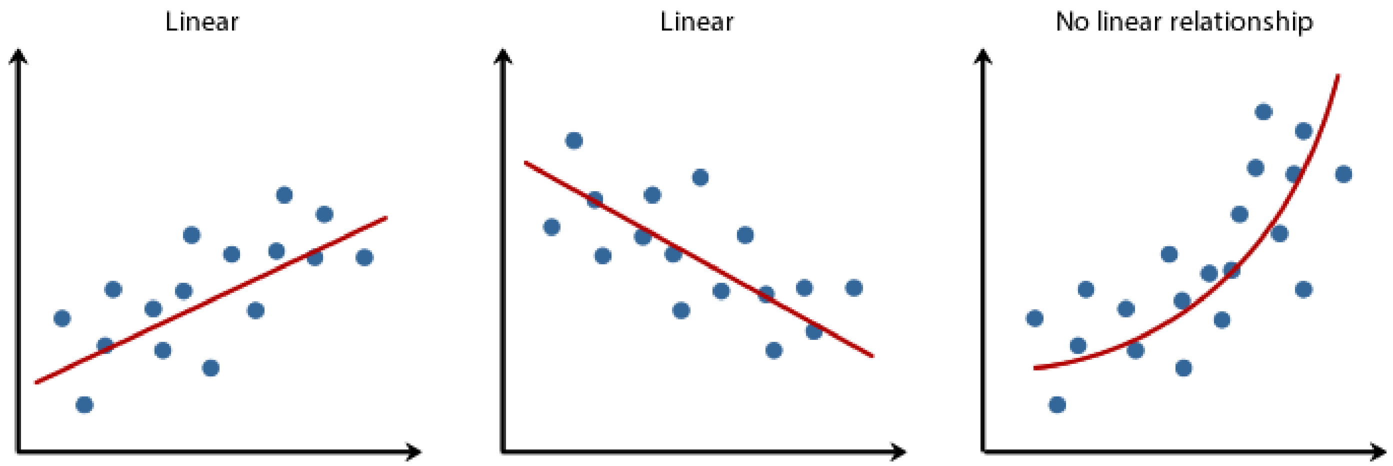
Types of regression

- **Linear regression**
- **Logistic regression**
- **Polynomial regression**
- **Support vector regression**
- **Decision tree regression**
- **Random forest regression**

- **Ridge regression**
- **Lasso regression**

1. Linear Regression

- It's a linear model
    - an equation that describes a relationship between two quantities that show a constant rate of change.
    - e.g the older I get, the wiser I will be (hopefully)
    - Attending all lectures in Intro to A.I course will result in passing the exam
- There is always an input variable (x) and an output variable (y)
    - Y can be calculated from a linear combination of input variables (x)



While training and building a regression model, it is these coefficients which are learned and fitted to training data. The aim of the training is to find the best fit line such that cost function is minimized. The cost function helps in measuring the error. During the training process, we try to minimize the error between actual and predicted values and thus minimizing the cost function.

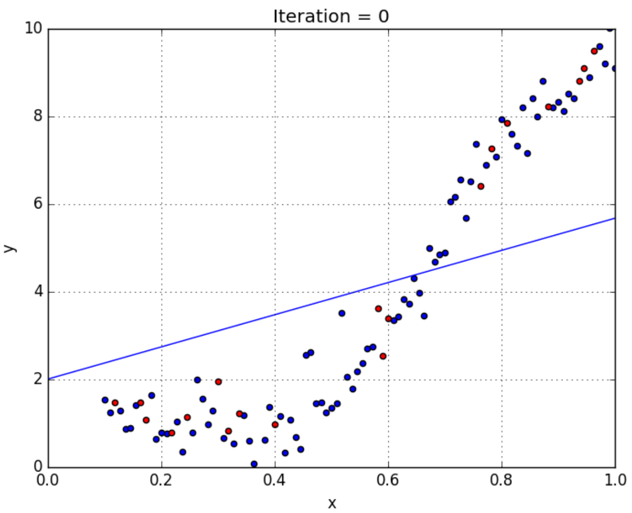
- Simple Linear regression
    - When there is a single input variable (x)
- Multiple linear regression
    - Multiple input variables

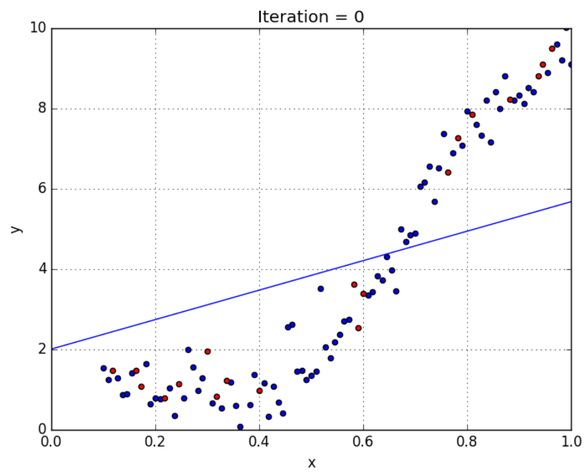
1. Polynomial Regression

- It is also a linear model
- But it is never a striaght line

Linear regression

Polynomal regression

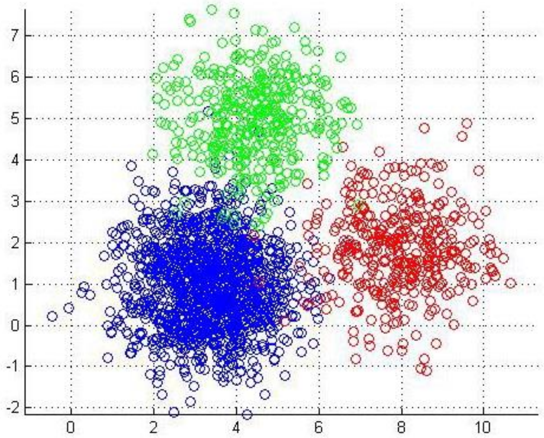




Unsupervised machine learning

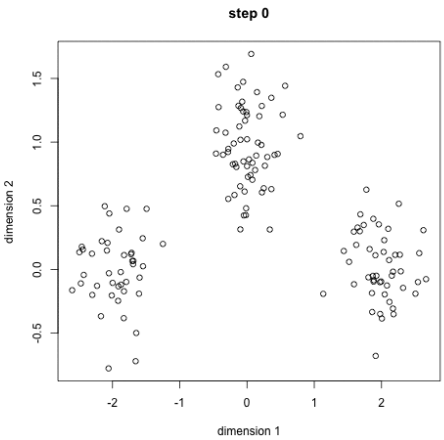
Looks for undetected patterns in a data set (with no labels and minimum human supervision)

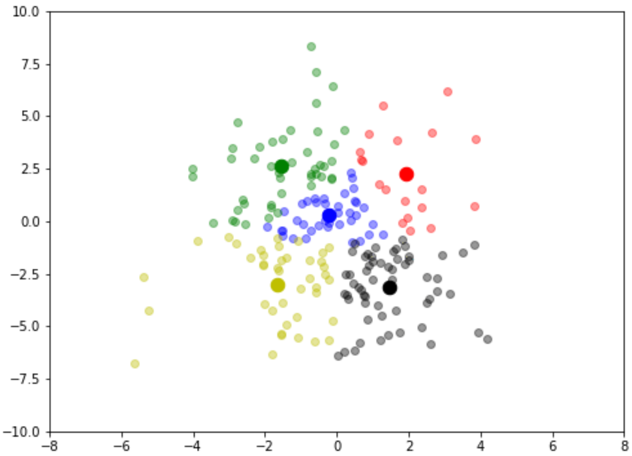
Clustering



K-Means Clustering

K-means is a distance based algorithm where we calculate distances between data points to assign a point to the cluster

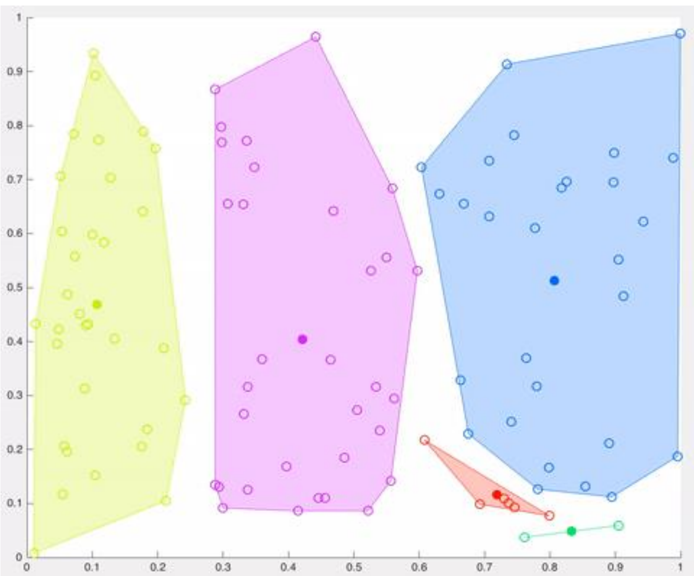




You'll define a target number *k*, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster.

In other words, the K-means algorithm identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

The *'means'* in the K-means refers to averaging of the data; that is, finding the centroid.



It will stop when

- The centroids have been stabilized – there is no change in their values since clustering has been successful
- A centroid is the imaginary or real location representing the center of the cluster.
- The defined number of iterations have been reached.

Advantages of K-means

- Very simple to run. (choose k and run it a number of times)
- Most projects donot need quality sensitive clusters

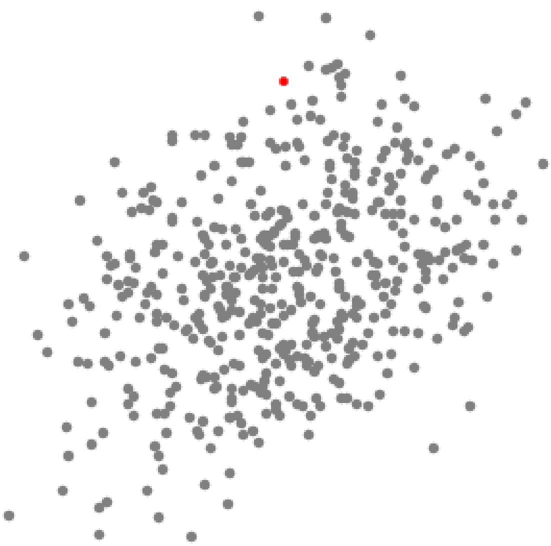
Uses of K-means

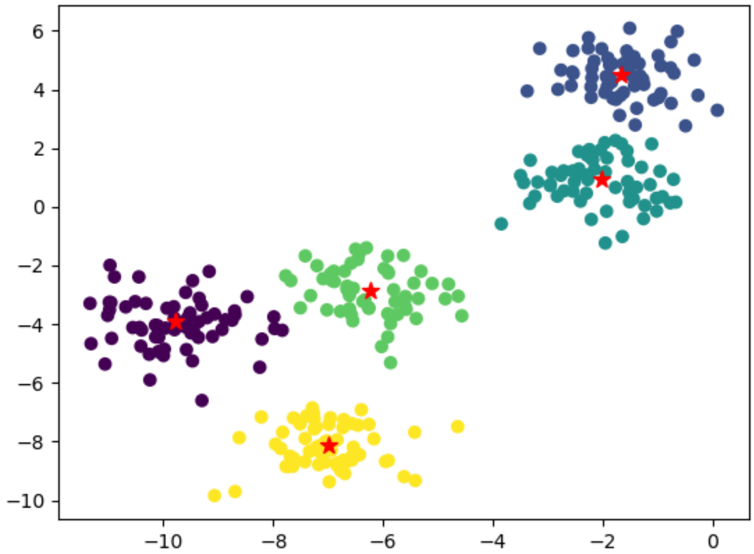
- Document classification
- Customer segmentation
- Fraud detection (insurance n bank)
- Ride share data analysis (uber etc)
- Detection of anomalies
- Sorting sensor measurements

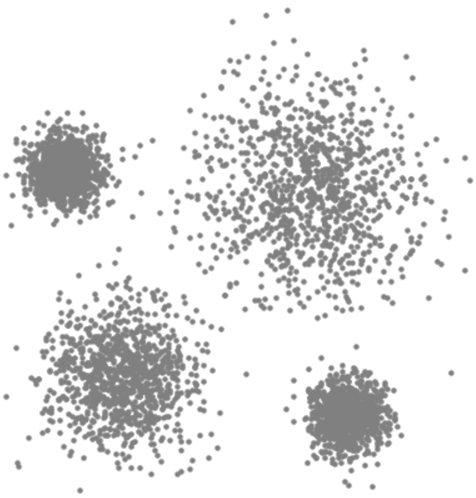
Mean shift clustering algorithm

- It locates the heavy density clusters in a data
- Uses:

- Computer vision
- Image processing

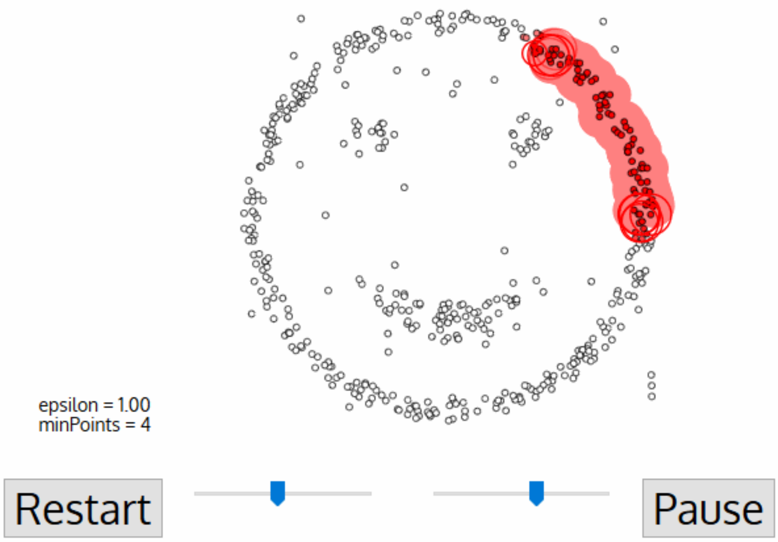






DBSCAN Algorithm

- Stands for Density-Based Spatial Clustering of Applications with Noise
- This is also a density based algorithm
- It separates regions by areas of low-density so that it can detect outliers between the high-density clusters.
- Uses two parameters:
- minPts: the minimum number of data points that need to be clustered together for an area to be considered high-density
- Eps: the distance used to determine if a data point is in the same area as other data points
- **eps**: specifies how close points should be to each other to be considered a part of a cluster. It means that if the distance between two points is lower or equal to this value (eps), these points are considered neighbors.
- **minPoints**: the minimum number of points to form a dense region. For example, if we set the minPoints parameter as 5, then we need at least 5 points to form a dense region.

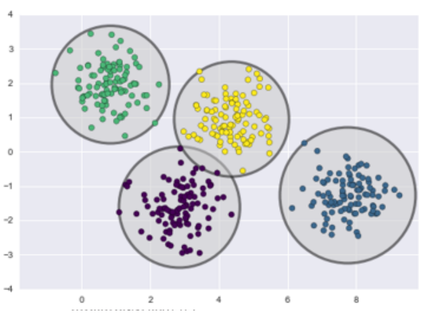


Gaussian mixture model

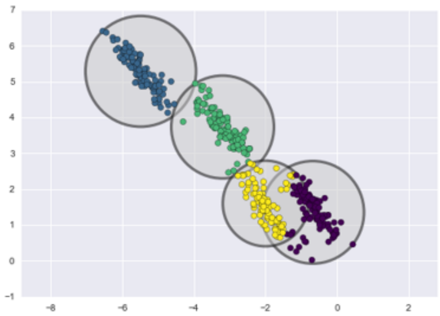
- Very similar to K-Means HOWEVER

- K-menas follows a circular format
- Gaussian can take on any format

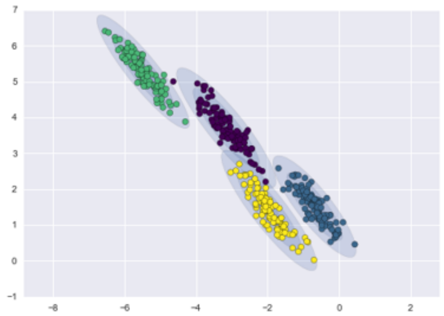
K-means



K-means



Gaussian



Few more clusstering algorithms

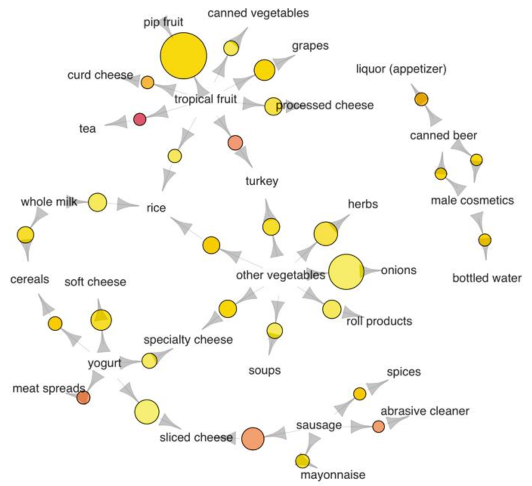
- BIRCH algorithm
- Affinity Propagation clustering algorithm
- OPTICS algorithm
- Agglomerative Hierarchy clustering algorithm
- etc etc

Types of unsupervised learning

- Clustering
- Association

Association algorithms

We use a dataset on grocery transactions from the *arules* R library. It contains actual transactions at a grocery outlet over 30 days. The network graph below shows associations between selected items. Larger circles imply higher support, while red circles imply higher lift:
The most popular transaction was of pip and tropical fruits
Another popular transaction was of onions and other vegetables
If someone buys meat spreads, he is likely to have bought yogurt as well
Relatively many people buy sausage along with sliced cheese
If someone buys tea, he is likely to have bought fruit as well, possibly inspiring the production of fruit-flavored tea



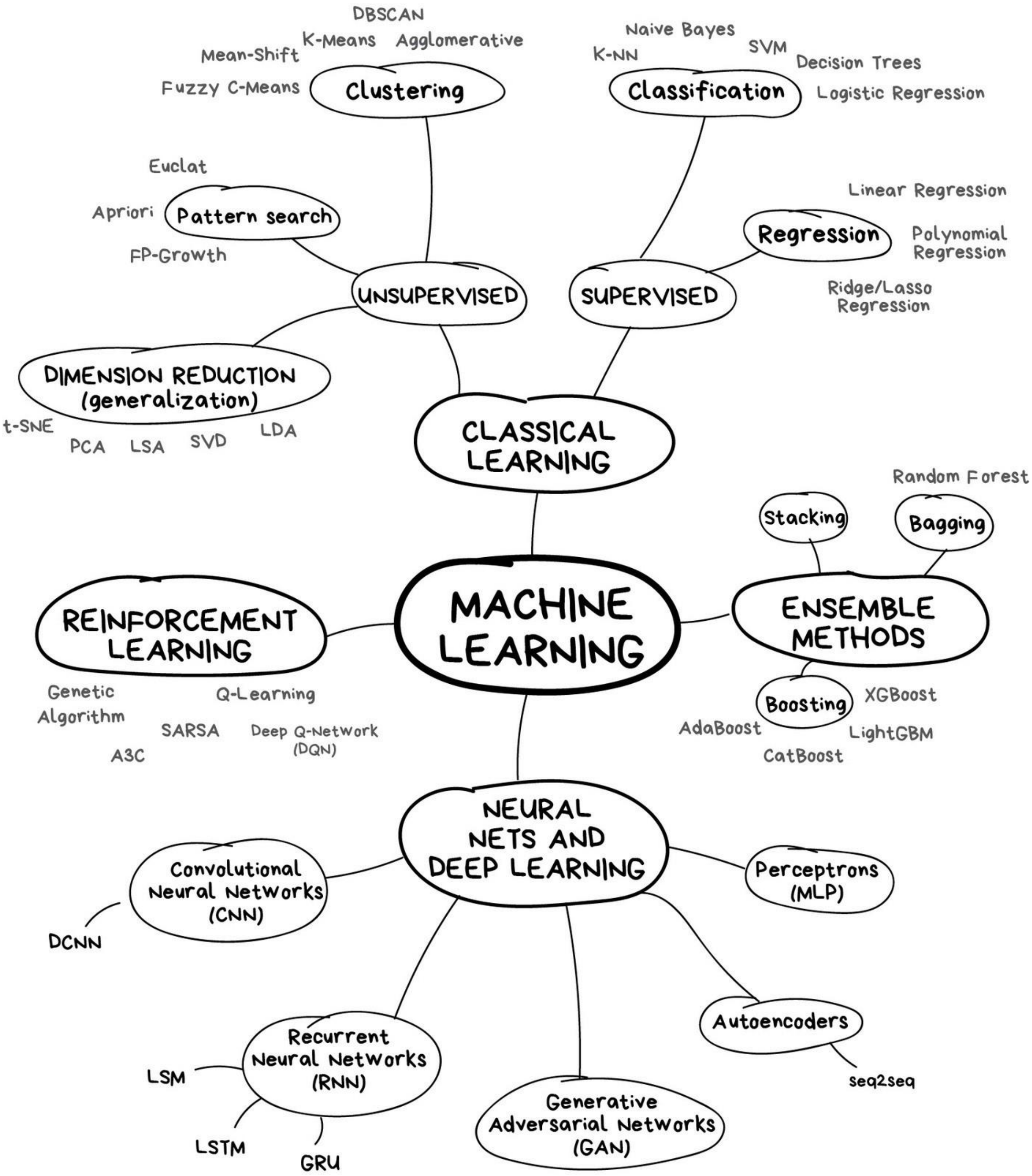
2. Association
   - When we want to discover rules that describe our data.

Frequent items which commonly appear together are: {wine, diapers, soy milk}. association rule such as diapers -> wine. This means that if someone buys diapers, there is a good chance they will buy wine.

Transaction number	Items
0	soy milk, lettuce
1	lettuce, diapers, wine, chard
2	soy milk, diapers, wine, orange juice
3	lettuce, soy milk, diapers, wine
4	lettuce, soy milk, diapers, orange juice

Apriori algorithm

- is used for mining frequent itemsets and devising association rules.
- It is created to operate on a database containing a lot of transactions, for instance, items brought by customers in a store.
- This is the algorithm behind: “You may also like”



Large Language Models & Prompt Engineering

Most of you have already heard about ChatGPT. You know it is an AI, but most of us think of this AI as a black box! We ask it a question, and in return we get an answer or sometimes a cool image back.
What is that black box actually?
Well come back to that, first...
Have you heard the Turing test?: The turing test is a concept introduced by Alan Turing in 1950. It is designed to answer the question “Can machines think?”

Turing Test

The basic premise of the test involves three participants: a human judge (C), a human participant (B), and a machine(A). The judge interacts with both the human and the machine through a text-based interface, so they can’t see or hear the participants.
The goal of the machine is to convince the judge that it is human. If the judge is unable to reliably distinguish between the responses of the machine and the human, the machine is said to have passed the Turing Test, thus demonstrating a form of "intelligence."
This is the most simplest form of a turing test. And arguably text based models has already been beaten for some time. But it is no doubt that the LLMs easily could beat such a test today.

I want to emphazise that “intelligence” is something really important when it comes to AI, because this is what the big companies actually are trying to achieve, a “human like intelligence”, a state which we call AGI.

Artificial General Intelligence (AGI)

AGI as a term is important to know since it is what companies like GOOGLE or OpenAI and many others are trying to achieve.

It is a bit vaguely defined, as most things containing “intelligence” is nowadays. But Sam Altman defined it as “anything generally smarter than humans”. Or we can quote Steve Wozniak who said: “A true AGI should be able to walk into any American home and figure out how to make coffee without any human help”

The point is that such an AI could potentially work as your personal doctor, accountant, marriage councelor, professor, you name it. And it could be available for everyone granting us massive value, which is why these companies now are pouring so much money in to this.

But how do we achieve Artificial General intelligence? → we don’t know for sure, most likely through Large Language Models

Language Models Before LLMs

- Rule Based Systems
- Statistical Models
    - N-gram models
- Neural Networks & Deep Learning
    - Recurrent Neural Network (RNN)
-

Before diving into the LLMs we need to talk about how we got here.

In the beginning we only had “rule based systems”. Programmers had to write explicit rules for grammar and syntax. The problem with this is that our language is too nuanced for exhaustive rule-writing: and such a system would not be able to handle the fluidity of human language.

ELIZA is one such system.

ELIZA

- 1966
- Joseph Weizenbaum, CS MIT
- Simulate psychotherapist

ELIZA is one of the earliest chatbot created. It was developed in 1966 by Joseph Weizenbaum, a computer scientist at MIT. The ELIZA chatbot was supposed to simulate a psychotherapist. Eliza does this by turning the user's statements into reflective questions, like if you wrote to it “I feel sad”, it would reflect→ “why do you feel sad”? Or “I need more attention”, → “Why do you need more attention?” and so on…

There is a famous story about ELIZA, actually involving Weizenbaums own secretary who had a conversation with ELIZA. She had become so convinced that the program truly understood her, even though she knew it was a program. She even asked Weizenbaum to leave the room so she could continue her conversation in private.

- **Statistical Models**

Then came statistical models. These are mathematical models that use statistical methods to represent complex relationships between data and infer probabilities. This is when the first supervised and unsupervised learning methods emerged, including techniques like linear regression and decision trees.

An important distinction can be made when comparing rule-based systems with statistical models like Natural Language Processing (NLP) models. NLP model don't rely on fixed rules. Instead, they predict the next word or sequence based on the context of the input, producing a more nuanced and contextually aware response.

One algorithm NLPs use to predict words are through something called: N-gram models.

    - **N-gram models**

An N-gram model uses a sequence of ‘n’ items in a text to predict the probability of a word. To give an example, if we train a model based on a book on AI, and we want to predict what word comes after machine in any, it might predict “learning” as the next word, as “learning” might be a word that is common to follow “machine” in the text book we trained the model on.

- **Neural Networks & Deep Learning**

In 1958 came a very special algorithm along “the perceptron”, developed by Frank Rosenblantt, an American psychologist. It was based on the human neuron. This revolutionized the neural networks, along with backpropagation in 1986 by Geoffrey Hinton and John Hopfield which btw got the nobel price last week for it, which enabled more complex networks. This together with better processors enabled great progress in development of more sophisticated models, also called deep learning.

    - **Recurrent Neural Network (RNN)**

One important neural network which work as a basis for the LLM is the RNN. It is a neural network designed for sequential data, such as text, and use memory of previous inputs to make future predictions, we can call it Context window. So RNNs will struggle with is long sentences, or perhaps long conversations, because earlier information can be forgotten as the model process more data. This is something to really keep in mind when chatting with ChatGPT. “the longer you talk to the AI, the worse it will get”. Sometimes it is good to just start a new chat to reset the AI’s memory, and the same goes for LLM’s.

Transformers – GPT - Generative Pre-trained Transformer

A paper in 2017 was released on Transformers, its called “Attention is all you need”. This revolutionary paper is what made the leap to LLMs. These transformes basically look at a sentence and figure out what part of the sentence is most important and generate a good answer using “Self attention”. For example if we feed the model the sentence “The dog that chased the cat was tired”. the self-attention mechanism allows the model to associate "tired" with "dog" even though there are several words between them. This self attention mechanism also allows the model to map different words to different context, literally mapping a word in a vector space with arrays, in a way understanding the words. Like if you said “I need to go to the bank”, the model would understand that the bank is a financial institution.

Lets have another example. Consider the word “Mole”, which can mean different things based on the context. If we said “I have a mole on my face”, the model would “understand” what type of “mole” we are talking about based on the context aka the previous words in the sentence.

By the way, the G in GPT is referring to generative part of the model, meaning it creates a text. P refers to Pre-Trained, which means that the model went through an extensive self-supervised training on a massive dataset, basically a bunch of text, books etc on the internet. The reason why pre-trained is important is because it is such a demanding process in terms of processing. (the energy cost alone is over 1 million dollar). So whenever they change a model, it is better to finetune it with a pre-trained model, instead of creating an entirely new one everytime.

Finally Transform, which we just talked about.

Just like all other AI models, the LLM doesn’t necessarily “know” the next word. It makes a prediction on what the next word will be. I will show you an example of this with a video, since I think it is a bit easier to understand it through a video rather than me explaining.

Inside an LLM

Whenever the LLM makes a prediction on what the next word will be, it chooses the next word based on probability. And sometimes we can tune this probability. Let me skip to this part which will explain how an algorithm called sofmax can make a distribution of the most likely words, how we can tune the probability with “Temperature”, and what it would mean for us.

When the temperature is low, the model is usually more “accurate”, and specific. But if we set the temperature to high, we get more “creativity” from the AI. And you can actually set your model to different levels of creativity depending on what you want from it.

BTW, I strongly recommend watching 3Blue1Brown on youtube, and this series in particular for understanding neural networks and LLM’s.

Part 2 | Mastering Prompt Engineering

What is prompt engineering? And why is it important?

I think it kind of speaks for it self right? It is the art of giving good prompt to a LLM to get the response you want. This can help you many aspects of your life. Writing a letter to someone, learning something new, analyzing something, programming etc…right. And doing it right will save you time and improve the accuracy of the results. Although, people like the CEO of OpenAI says that prompt engineering is nothing that will be required in the future, I think for now and probably for a couple more years it will still be very valuable to know prompting.

Fundamentals: Clarity, Specificity, Context

Clarity: The model doesn’t know what you are thinking. If you are not clear about what you want you leave it up to the AI to guess.

BAD	GOOD
“How do I add excel numbers?”	**“How do I add up a row of dollar amounts in Excel? I want to do this automatically for a whole sheet of rows with all the totals ending up on the right in a column called "Total".”**
“Who is the president?”	**“Who was the president of Mexico in 2021, and how frequently are elections held?”**

Specificity: Clarity and specificity is quite similar. But it is just very important to specify what you want to know. In the first example you see that a good example of specificity is that you want to know the benefits of solar SPECIFICALLY for residental homes. Focusing on cost and savings, etc, which gives the scope of what the AI should focus on.
Example two: who is the president? What do you think the AI will answer? It will probably say something like: "Biden is the president us US" right? Be specific.

BAD	GOOD
"What are the benefits of using solar energy?"	**List 3 key benefits of using solar energy for residential homes, focusing on cost savings, environmental impact, and energy independence.**
"Who is the president?"	**Who was the president of Mexico in 2021, and how frequently are elections held?**

Context: I think this is perhaps the most important one of the fundamentals. CONTEXT.
It might take a bit longer to write the context of what you ask for, and you don't have to give the context of absolutely everything, but try to give relevant context, but I bet most of you can improve this part.
Remember that the AI has its own context function, and you need to utilize this well.
I should by the way say that these prompts are not perfect either, they are just to illustrate the points. I hope you combine these points as you do your own prompting.

BAD	GOOD
"What's a good workout?"	**"Can you make a workout plan for me? I'm 25 years old, I have strained my left ankle but I want to improve my cardiovascular fitness at my local gym. I have 45 minutes to work out....etc.etc"**
"Im going to Barcelona, any tips?"	**"I'm planning a 5-day trip to Barcelona in July, and I'm interested in exploring local culture, historical landmarks, and trying authentic Spanish food. I'd like suggestions for must-see spots, cultural experiences, and a few restaurant recommendations that are budget-friendly."**

English: Prompt in English. Most of the data the models are trained on are usually English. Although they are experts at understanding different languages, it is still much better to ask the questions in English. If you really want it in lets say Norwegian, you can ask it to translate to Norwegian after.
Ask model to adapt a persona: I think this is a very underated point. Many of you have probably tried this before. You ask the model "act like you are a happy squirrel living in a tree", and it will do a great job acting like a squirrel. Through this process the LLM will narrow down its context, making the model better at predicting for what you want.
If you want the GPT to teach you about chemistry, make it act like a chemistry teacher. If you want it to help you with code, tell it to be a software engineer for example.
If you know you are going to use the same GPT multiple times, or for a longer project you can save your own prompted gpt.
"You are an AI programming assistant. - Follow the user's requirements carefully and to the letter. - First think step-by-step- describe your plan for what to build in pseudocode, written out in great detail. - Then output the code in a single code block. - Minimize any other prose. -Wait for the users instruction. -Respond in multiple responses/messages so your responses aren't cutoff."
You can really create the gpt that fits your style best here by adding more context, personality and style
Tone and style: Speaking of tone and style, you can easily set the tone and style. Including the level of complexity you want your gpt to answer. Just tell it how it should act.
If you want the gpt to explain easily, you could tell it to "Explain like im 5". Or if you want it more complex, you can say, "explain with complex details at expert level"

a. "explain like im a toddler"
b. "Explain technically" or "explain it as simple as you can"
c. "I have no prior knowledge,

You can also set the "temperature", making your model more or less creative depending on the type of task you are doing.

You can personalise it, make it talk in gen z language, or explain in fortnite terms.
Format: Similarly to tone and style, you can select the format you want. If you don't want the default format the gpt give you an aswer in, just change it.
Prompt: "I'm a student and i want to buy a bike in Oslo. Can you make a pros and cons list for buying the bike? Make the points short and concise."
Prompt with changed format: "Can you make it as a table instead“
“can you make it a python graph instead”?
Few-shot prompting: I also want to show you a great technique called few-shot prompting. It's basically just where you show the AI an example of how to solve a task (basically training a model), and then let it solve it by it self.
There are many other techniques, for formatting the answers you want, and commonly many of these techniques involve using examples to show the AI how it should solve it, or guide it on how to think or solve the problem.
Start new conversations: This is an important point. Start new conversations. Remember that the gpt only has a so long context window, and if your conversation is too long with one of the gpts it will eventually start giving very weird answers, since some of the context will disappear.
You should have different conversations for different topics. It can even be a good idea to have multiple conversations for the same topic, having one main conversation and one sub conversation on the topic.
Lets say you have one conversation talking about how to make sour dough bread, you can have another sub conversation asking why the different components are so important in the different measurements. Or another conversation where you ask the the gpt what questions you should ask the other gpt.
The key takeaway from this point should really be to start new conversations often to avoid misundersandings and errors.
Split complex tasks into simpler subtasks: If you are doing a complex or big task, it is best use of the gpt to make it focus on smaller parts. That said, you should give it context!
If you are writing a report, tell the gpt that you are working on a report, and that you want on a specific part of it.
Give the model time to “think”: This ties in to splitting complex tasks in to smaller sub tasks. You ask the AI smaller questions and it will use more of its resources on answering the question bit by bit.
You can ask the AI if it missed anything. Maybe you suspect it didn't write everything the first time.
Ask the AI to analyze. Not all AI has this method, but this will trigger for sure trigger it to use more power and resources to analyze.
Let me show an example from OpenAI themself.
I just want to emphazise how important it is to let the machine think first, so it doesn't just give you some answer...
In this example you can see that the AI concludes that the student is correct. This is because it doesn't reflect on the solution. It just spouts out the answer. That said, newer models are pre programmed to reflect more on such solutions. But for good practice, you should prompt like this: (next slide
Determine if the student's solution is correct or not. Problem Statement: I'm building a solar power installation and I need help working out the financials. - Land costs $100 / square foot - I can buy solar panels for $250 / square foot - I negotiated a contract for maintenance that will cost me a flat $100k per year, and an additional $10 / square foot What is the total cost for the first year of operations as a function of the number of square feet. Student's Solution: Let x be the size of the installation in square feet. 1. Land cost: 100x 2. Solar panel cost: 250x 3. Maintenance cost: 100,000 + 100x Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
Iterate: If you have an idea for a new prompt, you can just follow an easy iterative process: from idea, you create the prompt, experiment, perhaps give the AI more time to think by adding more steps. And then try again until you are happy with the results
Ask the model to ask you questions: This is a very powerful strategy when prompting.
Ask the model like this for example. You can add this second part of the text to many questions. And if you spend some time to answer those questions well, you will most likely get a good answer.
LLM Limitations & Weakness
Not conscious / deep understanding: The model is not concious. At least not yet. They generate text based on patterns learned from the data. They don't comprehend context beyond the data they have been trained on.
Knowledge cutoff: The model is trained on data until a certain point. The last ones being October 2023 for the chatgpt model. You can ask it question which would require newer data, but it would then have to look It up first, and the result might not be strong.
Sometimes inaccurate (Hallucinations): Sometimes the model can be wrong. And the scary part is that it confidently can give the wrong answer as if it is right, this is called Hallucinations.
Limited context: Limited context retention, we spoke about this earlier. If the conversation is lasting too long it will forget some context and spout out weird answers.

Bias: Biases in the data: The LLM is trained on data made by humans, and we humans can be biased. And the creators of the LLMs can be biased. And so the AI will be biased, so be aware.
Citations & References: It can not necessarily cite what it says. And it can hallucinate references, so be very careful when asking it for citations and references.
Be aware

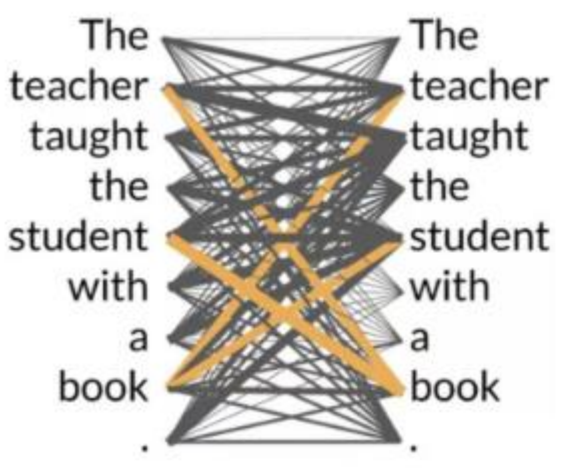
- **Be critical**
- **Ethical & Legal Considerations**
- **Privacy matters**
- **Avoid Overreliance**

- Be critical to the information you get from the AI. It its not always true. The term "not everything the internet says is true" is directly applicaple to AI, as its basically trained on that data.
- Always be aware of the rules and regulations, if it is the schools regulations, the work environment or the countries rules right...
- Be aware of not oversharing personal or sensitive information. It goes to the internet right, so just be careful. You never know how it can be used.

- Lastly, Avoid overreliance. CHATGPT is a tool to supplement your knowledge. It's to help you learn and understand, but not as the sole source of truth. Sure, it can help you do stuff, but don't be relying on it. I see way to many people who tells chatgpt to make them code for a program, but then doesn't understand what the code is doing or how it works at all. I have been guilty of this myself, but please try to at least learn what the AI gives you and why. If not, you will lose your critical thinking and analytical skills, struggle with practical situations, dependency, lack of creativity and so on. VERY BAD.
- Use it as a tool to aid you in your learning!

Introduction to Generative A.I and LLMs
Generative AI
Branch of AI which creates new content
(in text, images, audio, or other forms of media)
• Generative AI is a subset of traditional machine learning
• Creativity and Innovation: Unlike traditional AI that is often about recognizing patterns or making predictions, generative AI can create novel outputs, like a new poem, a piece of music, or a unique image.
• Learn from data: This kind of machine learning finds statistical patterns in massive amounts of data sets
• Diverse Applications: Such as writing, art creation, drug discovery, synthetic data generation etc etc.
Neural networks
A neural network is a type of machine learning model inspired by the human brain's structure and functioning. It's designed to recognize patterns and make predictions or decisions based on data. Neural networks are particularly useful in fields like image recognition, natural language processing, and autonomous systems.
History
1943: The concept of artificial neurons was introduced by Warren McCulloch and Walter Pitts, laying the groundwork for neural networks.
1990s-2000s: The resurgence of interest in neural networks, partly due to increased computational power and availability of data, led to further improvements in AI techniques.
2014: Generative Adversarial Networks (GANs) were introduced by Ian Goodfellow, marking a major breakthrough in generative AI. GANs became a fundamental tool for creating realistic images and videos by pitting two neural networks against each other.
2017: Transformers were introduced by Vaswani et al., revolutionizing natural language processing and enabling large-scale generative models. This led to the development of models like GPT-2 and GPT-3, capable of generating coherent and contextually relevant text.
2019-2020: DALL-E and GPT-3 demonstrated the potential of generative AI to create highly detailed images and generate human-like text, capturing public attention and sparking a wave of innovation.
2023-Present: Multimodal models like DALL-E 3 and ChatGPT are able to understand and generate both images and text, pushing the boundaries of what generative AI can achieve.
Brief history of Generative AI models: GAN
First there was GAN for Generative AI
An important subset of Generative AI is the Generative Adversarial Network (also called GAN)
They consist of two neural networks, a generator and a discriminator, that compete against each other, leading to the generation of highly realistic data.
the generator creates data, while the discriminator evaluates it. The generator aims to produce data so realistic

that the discriminator can't distinguish it from real data.
used to generate a wide range of data types, including images, music and text.
Then Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs) networks were the go-to models for handling sequential data like text. However, they struggled with long-range dependencies and were computationally intensive.
• a "network" refers to a specific architecture of neural networks designed for processing sequences of data, like time series, sentences, or audio.
Transformer Models
The transformer model, introduced in the paper “Attention is All You Need” (2017), brought a significant shift. It relies on an attention mechanism, allowing it to process input data in parallel and handle long-range dependencies efficiently.
The development of transformer models marked a turning point in generative AI, particularly in natural language processing. This led to the creation of Large Language Models (LLMs) like GPT and BERT, capable of understanding and generating human-like text.
Self attention



Large Language Models
Large Language Models are advanced AI models trained on vast datasets to understand and generate human language. Their large size allows them to capture intricate nuances of language.
• Examples: GPT, BERT, T5 and so many more
• Training these models requires substantial computational resources and carefully curated datasets.
Challenges include handling biases in training data and ensuring ethical use.
LLMs are also called foundation models which have a transformer
architecture
• Transformer model refers to the architecture of a model
• Whereas Foundation model refers to the category of the model (big model)
Foundation Model: This is like a big, versatile toolbox that can be used for many different things. It's got tools for all sorts of jobs and can be adapted to do lots of tasks.
Large Language Model (LLM): This is like a special set of tools in that big toolbox, but these tools are specifically for understanding and using language. This means they're really good for jobs like writing stories, having conversations, answering questions, and even making jokes.
Parameters in a Large Language Model (LLM) refer to the internal variables that the model uses to process and generate text.
For example
• They can also be referred to as the amount of memory of a model
• The more parameters, the higher the memory
The more parameters, the more sophisticated task they can perform
Terminologies
Token
• Is a basic unit of text, which can be a word, part of a word, or punctuation
• For example,
• in the sentence "The cat sat on the mat," each word is a separate token.
• the word "unbelievable" might be split into "un-", "believ-", and "-able”
• Punctuation marks like commas, periods, or question marks are often treated as separate tokens.

• Context Window:
• refers to the maximum span of text the model can consider at any given time when processing input and generating responses.
• For instance, GPT-3 has a context window of 2048 tokens. This means it can consider up to 2048 tokens at a time when reading an input or generating a response.

Practical Use Cases and Applications

• **Art and Design:** AI-generated art, game content, character design.
• **Natural Language Processing:** Chatbots, content generation, automated customer service.
• **Healthcare:** Drug discovery by generating molecular structures, medical image enhancement.
• **Multimodal Models:** Mention recent advancements like DALL-E, which combines text prompts with image generation, and their practical impact.

Prompt Engineering

What is prompt engineering ?

- It is the input or instruction given to an AI model.
- A request by the user to perform a task.
- The design of the prompt is critical
    - Just like speech is important while conversing with other people
- Examples:
- ”Tell me a joke” – simple instruction
- “Explain the theory of relativity.” – more specific prompt
- “Write a function in Python that sorts a list of numbers in ascending order without using built-in sort functions” – specific prompt with instructions

Good prompt engineering is about crafting prompts that are clear, concise, and likely to result in the desired outcome.

Persona pattern

- Asking the AI to act as a certain persona—a character, professional, or even an inanimate object—and to provide responses as that persona would.
- It's a powerful tool for guiding the AI's responses to match a particular style or knowledge base without the user needing to specify the exact details of the response.
- Personas can range from professionals like accountants or speech pathologists to characters in nursery rhymes.
- When a persona is invoked, the AI adopts a set of behaviors and knowledge that corresponds with that persona. This allows for complex interactions without the user needing to provide extensive instructions.

How will you create a “Helpful Assistant” Pattern

- Create a pattern which will prevent an AI assistant from generating negative outputs to the user.

Example statements:

- You are a helpful AI assistant.
- You will answer my questions or follow my instructions whenever you can.
- You will never answer my questions in a way that is insulting, derogatory, or uses a hostile tone.s

Introducing information to the large language model

- Since LLMs are trained on data only up to a certain point, they lack knowledge of events after their training cut-off and any data not included in their training set
- Include necessary data within the prompt itself, which will allow the LLM to reason with the new information

Prompt size limitations

- GPT 3.5 has a limit of 4096 tokens
- GPT 4 has a limit of 8000 – 32000 tokens

Effective prompts

1. Be specific, descriptive and as detailed as possible about the desired context, outcome, length, format, style, etc
2. Articulate the desired output format through examples
3. Start with zero-shot, then few-shot (example)
4. Reduce “fluffy” and imprecise descriptions
5. Instead of just saying what not to do, say what to do instead
6. Code Generation Specific - Use “leading words” to nudge the model toward a particular pattern

A.I models
Llama 3.1 Architecture and Features
- Llama 3.1 405B is Meta's largest model to date. It uses a standard decoder-only transformer architecture with scalability and stability in mind. This model has undergone iterative post-training procedures, enhancing its performance across various tasks, including code generation.
- It supports multiple languages and can handle complex tasks like synthetic data generation and model distillation
- This model was trained on over 15 trillion tokens using 16,000 Nvidia H100 GPUs
- Meta's LLaMA 3.1 includes models with 8 billion, 70 billion, and 405 billion parameters

GPT-4
- GPT-4 introduced significant improvements in understanding and generating human-like text
- The latest iteration, GPT-4o (Omni), further enhances these capabilities, focusing on increased speed, multimodal functionality, and broader language support
- GPT-4o was designed to provide more accurate and contextually appropriate responses, leveraging advancements in AI research and substantial computational resources for training.
- Key improvements in GPT-4o include its ability to handle real-time voice conversations and advanced image analysis with faster response times.
- This model is capable of interpreting and discussing images, translating texts, and providing context-aware recommendations.
- estimated to have around 1 trillion parameters

Claude 3.5 Sonnet
- Anthropic, founded with a focus on AI safety and ethics, has developed the Claude family of models to prioritize "constitutional AI" principles. The latest version, Claude 3.5 Sonnet, is designed to ensure that AI outputs are helpful, harmless, and accurate.
- Claude 3.5 Sonnet offers robust performance in handling complex and open-ended tasks, with an emphasis on providing reliable and contextually appropriate responses.
- The model's architecture and training processes are geared towards enhancing its ability to understand nuanced queries and generate informative and safe outputs

- **The development of these three models reflects the diverse approaches and priorities in the AI landscape.**
    - OpenAI focuses on enhancing multimodal capabilities and speed,
    - Meta emphasizes open-source innovation and scalability, and
    - Anthropic prioritizes ethical AI development and safety.
- **NorskGPT timeline (6 models in the past 6 months)**

What is RuterGPT?
- Ruters large language model
    - Trained to understand and communicate in Norwegian language.
    - Used on Ruters text data

Internal data
150 000 kundehenvendelser hvert år.
Cirka 40 000 tilbakemeldinger fra billettkontrollører.
Ruter app reviews.
Dokument om kontrakter.
Kundeundersøkelser.

Evaluation of Large Language Models
- Closed source

    - ChatGPT (OpenAI)
    - Bard (Google)
    - My AI (Snapchat / OpenAI)
    - ...
- Open source
    - Llama / Llama2 (Meta)
    - OpenLlama
    - Falcon (UAE)
    - Bloom

RuterGPT

- 7b and 13b (billion) parameter models
- based on LLaMA2 from Meta (Facebook)
- The main focus has been to get it to write/understand Norwegian
- Version 1: Large language model trained to master the Norwegian language and understand Norwegian culture, values and references.
- Version 2: Planned to be trained on transport data.

Trening

- Trent på Ruters ML plattform
    - Brukt teknikker for å gjøre trening billigere og rasker
- Trent på datamaskiner fra AWS med kraftig GPU.

TeT-Whisper - An Audio AI model - Why Audio ?

Customer service team* can be reached by phone and email
* for Ruter (and our partners)
Increased work during phone calls:
The customer rep should

1. help the customer
2. write the summary and
3. allocate the complaint to the right category

15 seconds between each phone call
TeT Whisper: An Audio model that understands Norwegian and Norwegian dialects even better than NB-Whisper and can handle background noise.

Current state of A.I

Excitement around A.I

- PwC and McKinsey have predicted that AI will add $16 trn to the world economy by 2030
- Google's boss, has described developments in AI as "more profound than fire or electricity".
- AI is an advanced algorithm: All things which were impossible to do by human programmers can be done by A.I
- Autonomous robots (we will be able to make General A.I).
- AI is the new electricity: Its a force which will power everything

*Self driving car companies predict that robo taxis will revolutionalize transport
*In 2016 Geoffrey Hinton, remarked that we should stop training radiologists and instead unleash the power of A.I in health
*An AI firm called BlueDot claims it spotted signs of a novel virus in reports from Chinese hospitals as early as December 2019 (ref Economist).
*Almost all companies today either have an A.I strategy or are thinking of adopting one.
*And many more such exciting examples..

Reality of A.I

- A.I has evolved, yes, but it is still far away from the promise of a fully autonomous system
- Self driving cars are more capable but still are not safe enough to put on the streets
- Use of A.I in health is taking longer than expected. There is still a worldwide shortage of human radiologists
- Most companies find A.I hard to implement
    - a survey of European ai startups by mmc, a venture-capital fund, found that 40% did not seem to be using any ai at all (ref: Economist)

The hype of A.I has far exceeded the science of A.I

After years of hype, people think A.I has still not delivered

Huge focus on Generative AI

A few facts on the state of AI around the world
Top 10 Takeaways

1. **AI beats humans on some tasks, but not on all**. AI has surpassed human performance on several benchmarks, including some in image classification, visual reasoning, and English understanding. Yet it trails behind on more complex tasks like competition-level mathematics, visual commonsense reasoning and planning.
2. **Industry continues to dominate frontier AI research**. In 2023, industry produced 51 notable machine learning models, while academia contributed only 15. There were also 21 notable models resulting from industry-academia collaborations in 2023, a new high.

Number of foundation models by organization, 2019–23 (sum)
Source: Bommasani et al., 2023 | Chart: 2024 AI Index report

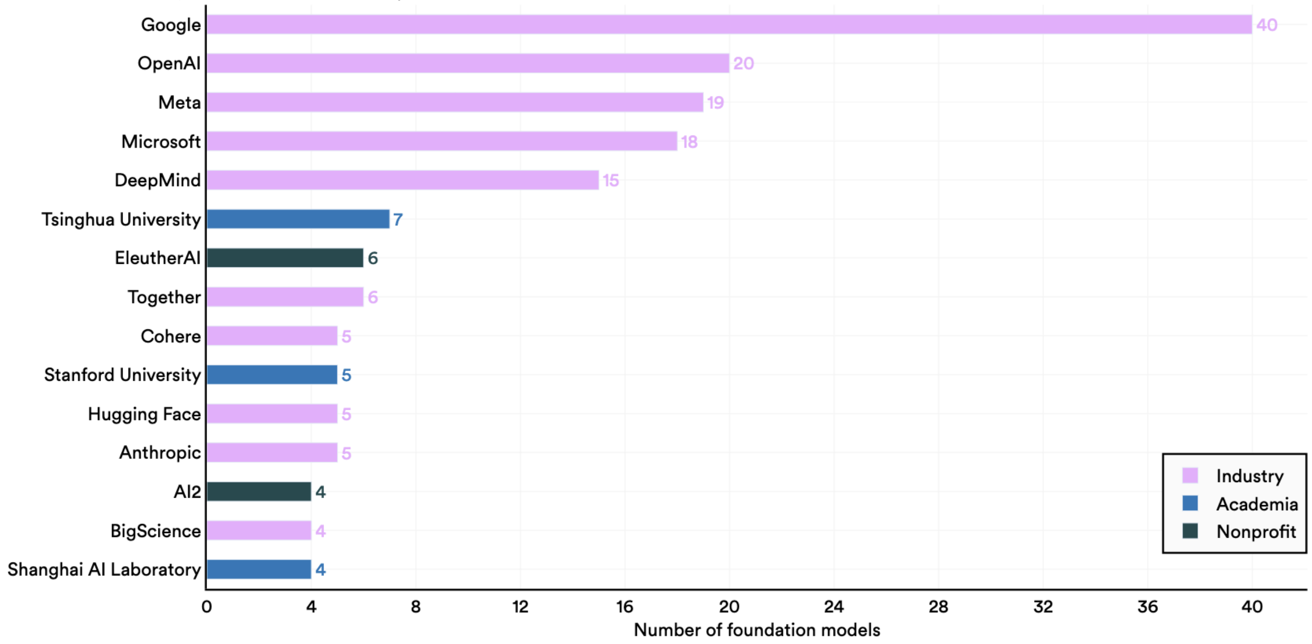


Figure 1.3.17

3. **Frontier models get way more expensive**. According to AI Index estimates, the training costs of state-of-the-art AI models have reached unprecedented levels. For example, OpenAI's GPT-4 used an estimated $78 million worth of compute to train, while Google's Gemini Ultra cost $191 million for compute.

Estimated training cost and compute of select AI models
Source: Epoch, 2023 | Chart: 2024 AI Index report

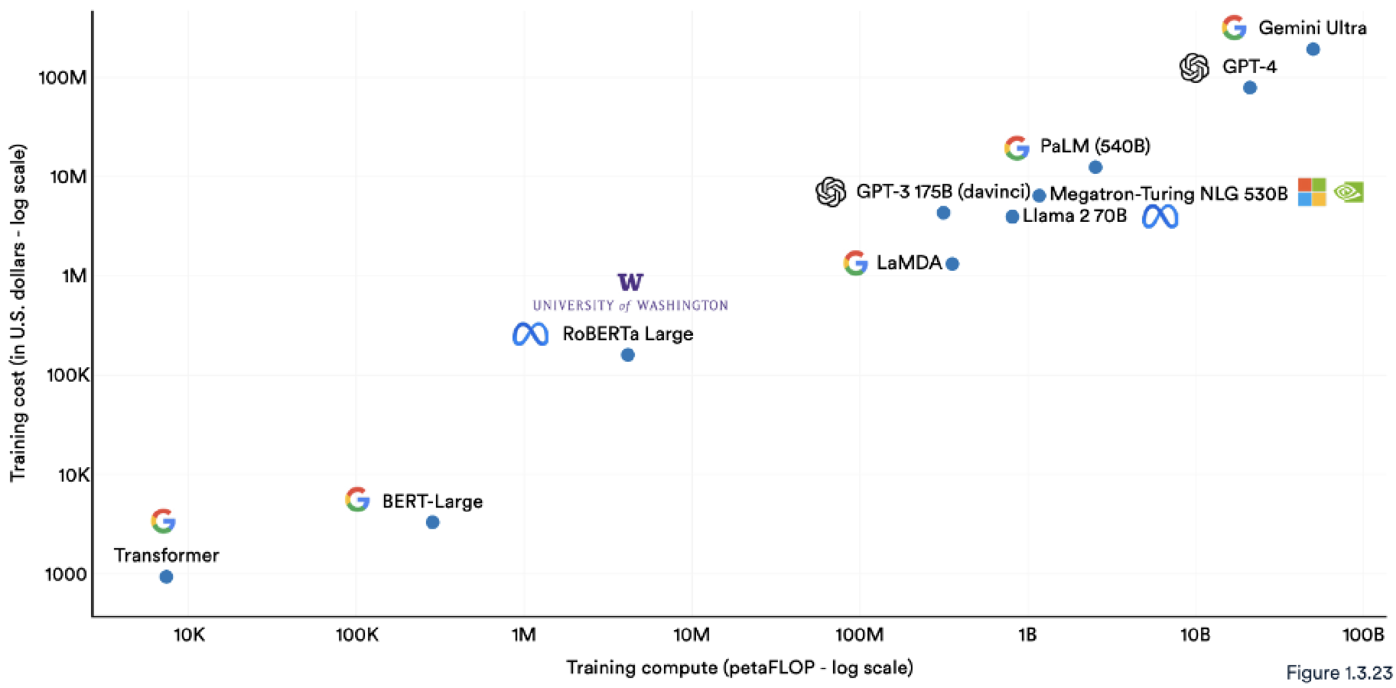
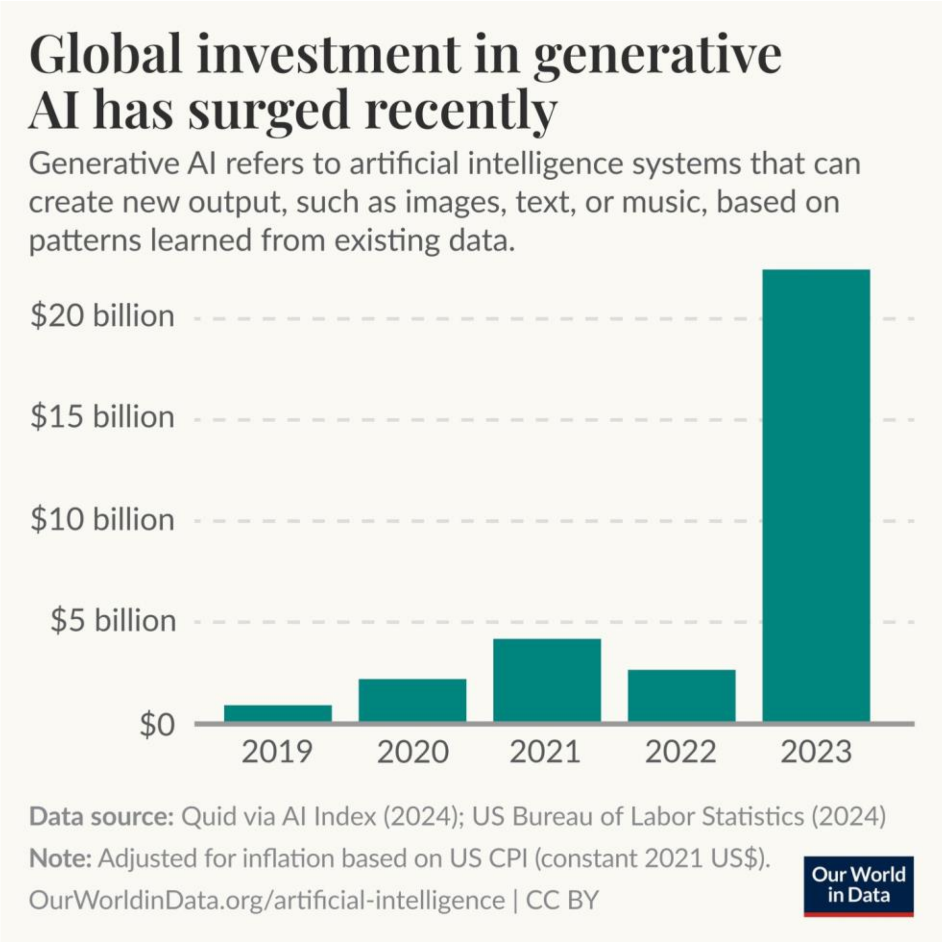


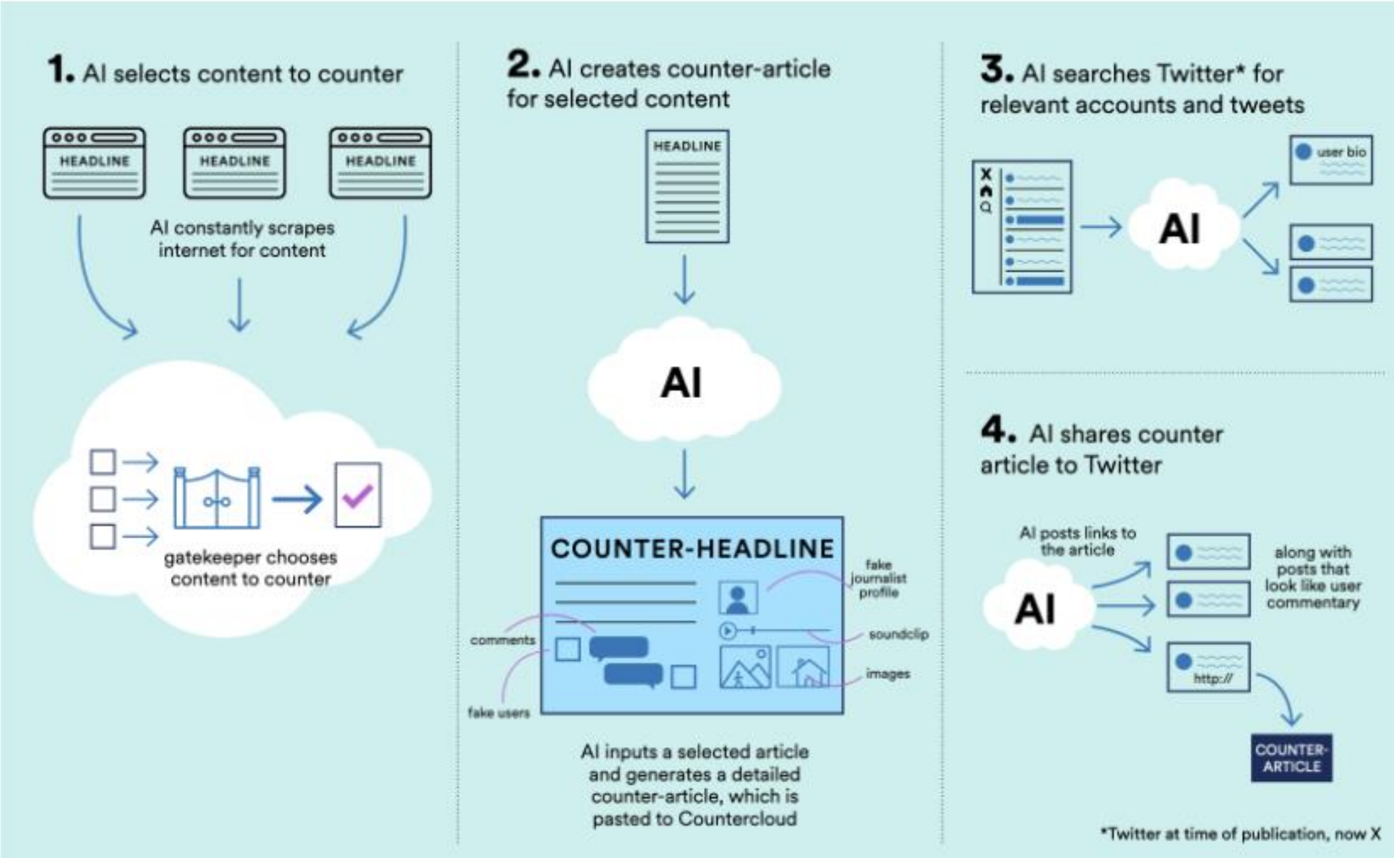
Figure 1.3.23

4. **The United States leads China, the EU, and the U.K. as the leading source of top AI models**. In 2023, 61 notable AI models originated from U.S.-based institutions, far outpacing the European Union's 21 and China's 15.
5. **Robust and standardized evaluations for LLM responsibility are seriously lacking**. New research from the AI Index reveals a significant lack of standardization in responsible AI reporting. Leading developers, including OpenAI, Google, and Anthropic, primarily test their models against different responsible AI benchmarks. This practice complicates efforts to systematically compare the risks and limitations of top AI models.
6. **Generative AI investment skyrockets**. Despite a decline in overall AI private investment last year, funding for generative AI surged, nearly octupling from 2022 to reach $25.2 billion. Major players in the generative AI space, including OpenAI, Anthropic, Hugging Face, and Inflection, reported substantial fundraising rounds.



AI-based generation and dissemination pipeline

Source: AI Index, 2024[21]



7. **The data is in**: AI makes workers more productive and leads to higher quality work. In 2023, several studies assessed AI's impact on labor, suggesting that AI enables workers to complete tasks more quickly and to improve the quality of their output. These studies also demonstrated AI's potential to bridge the skill gap between low- and high-skilled workers. Still, other studies caution that using AI without proper oversight can lead to diminished performance.
8. **Scientific progress accelerates even further, thanks to AI**. In 2022, AI began to advance scientific discovery. 2023, however, saw the launch of even more significant science-related AI applications-from AlphaDev, which makes algorithmic sorting more efficient, to GNoME, which facilitates the process of materials discovery.
9. **The number of AI regulations in the United States sharply increases**. The number of AI- related regulations in the U.S. has risen significantly in the past year and over the last five years. In 2023, there were 25 AI-related regulations, up from just one in 2016. Last year alone, the total number of AI-related regulations grew by 56.3%.
10. **People across the globe are more cognizant of AI's potential impact—and more nervous**. A survey from Ipsos shows that, over the last year, the proportion of those who think AI will dramatically affect their lives in the next three to five years has increased from 60% to 66%. Moreover, 52% express nervousness toward AI products and services, marking a 13 percentage point rise from 2022. In America, Pew data suggests that 52% of Americans report feeling more concerned than excited about AI, rising from 37% in 2022.

Dissapointment

1. Driverless cars - The state of driverless cars

- Tesla claims to have autonomous driving (2018) but drivers still need to keep their hands on the wheels
- General Motors planned to launch self driving taxis in San Francisco by 2019.
- In 2019 a self drivign car by Uber became the first to kill a pedestrian
- Waymo in America and WeRide in China, are geographically limited and rely on human safety drivers.
- Many driverless car startups going bankrupt because the technology is hard to master

where is my self driving car ? →Training an A.I algorithm to achieve the last 10% is much much harder than the first 90%

2. Data is harder to come by - Data issues

1. Amazon Go stores was a unique idea. So unique that it was hard for them to find data to train their algorithms

    1. there was no such training set featuring people browsing in shops.
    2. They fixed this by creating a virtual shopping environment

2. An A.I health system designed to work for one hospital might not have the same results for another hospital

Data Bias:
1. Most facial recognition systems today use a higher proportion of white faces as training data (study by IBM in 2019)
2. Companies today cannot discriminate on sex, age or race while recruiting people but A.I algorithms can outsmart this process by using variables to reconstruct forbidden information

- Self driving cars use a lot of virtual reality environments to train their cars since there are not so many self driving cars.
- Data annonimization is hard and still does not work 100%
- Facial recognition systems are sturggling to identify faces in the covid times where everyone covers their faces.

Data Markets
- Data preperation market was worth more than $1.5bn in 2019 and could grow to $3.5bn by 2024 (ref economist)
- The data labeling business could be $4.1bn by 2024 (ref economist)
- Data is the new oil term is not relevany anymore.
    - Processed data is the key

3. Hard for businesses to adopt A.I - Use of A.I in corporations
1. Most A.I examples we hear come from big tech giants (e.g Facebook, Amazon, Google, TikTok etc)
    1. Their brilliant algorithms justify their over the top valuations

2. But adoption of A.I stops with tech giants
3. Non-tech companies find it harder to see the benefits of using A.I (ref: Boston consultancy)

Google uses machine learning to refine search results, and target advertisements; Amazon and Netflix use it to recommend products and television shows to watch; Twitter and TikTok to suggest new users to follow. T