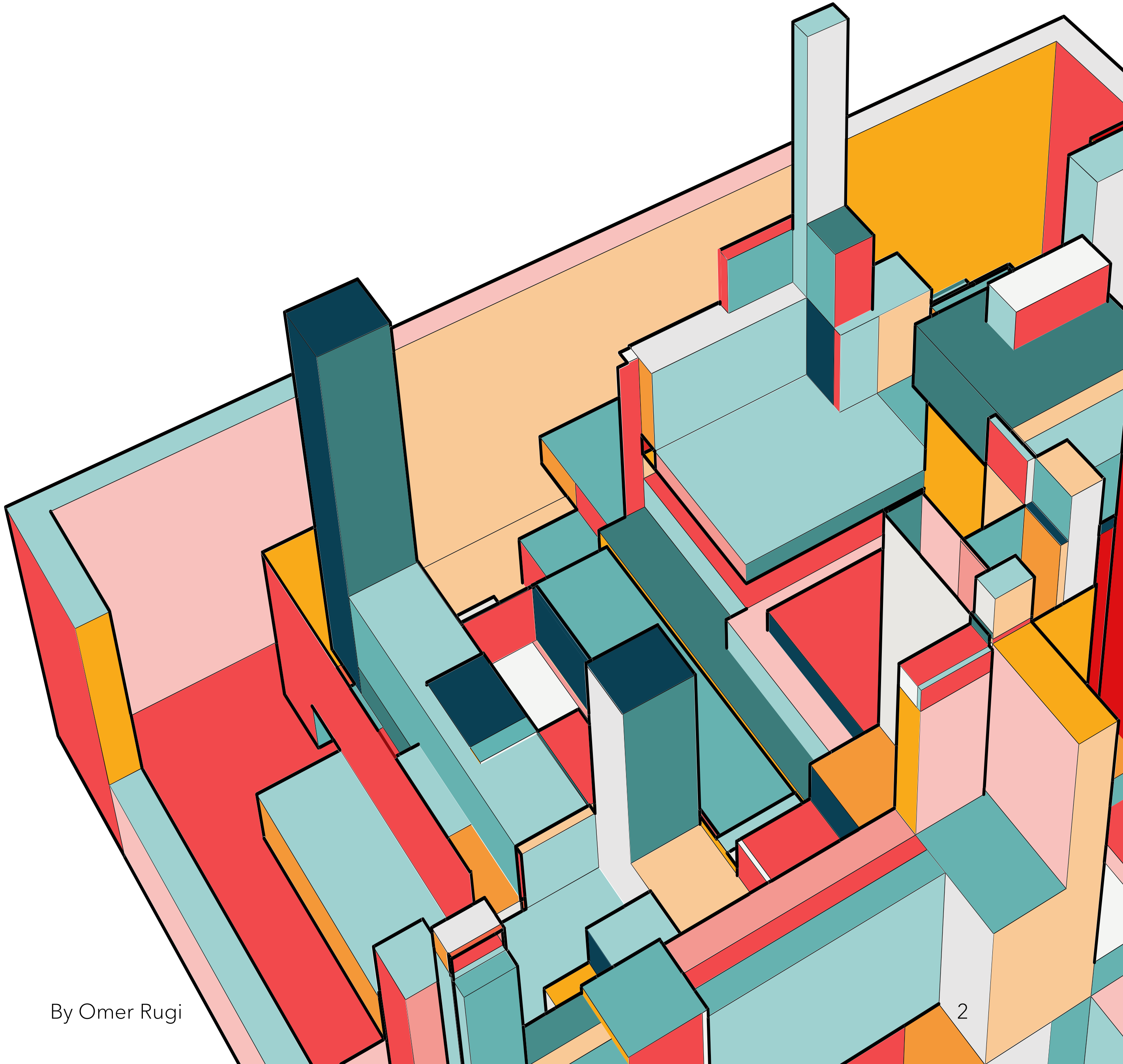
A.I, ML, DL

Brief introduction to the field of data.

Created by: Omer Rugi

MAIN IDEA

- Get you familiar with basic concepts.
- Enhance your terminology.
- Will not go in depth to models, math, or theory and will skip different fields to focus on the basics.

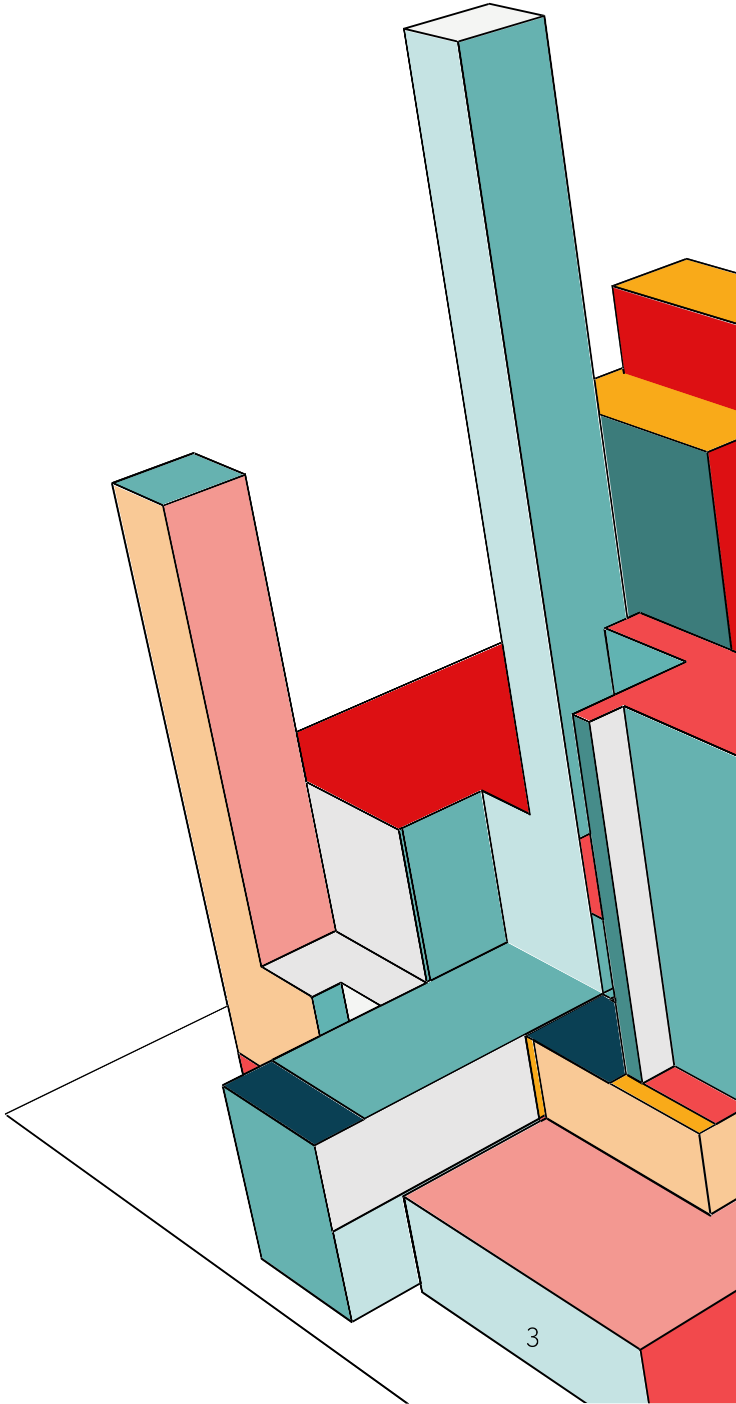


AI IN A NUTSHELL

AI, or Artificial Intelligence, refers to the development of computer systems that can perform tasks that usually require human intelligence. Like problem-solving, learning and understanding natural language for example.

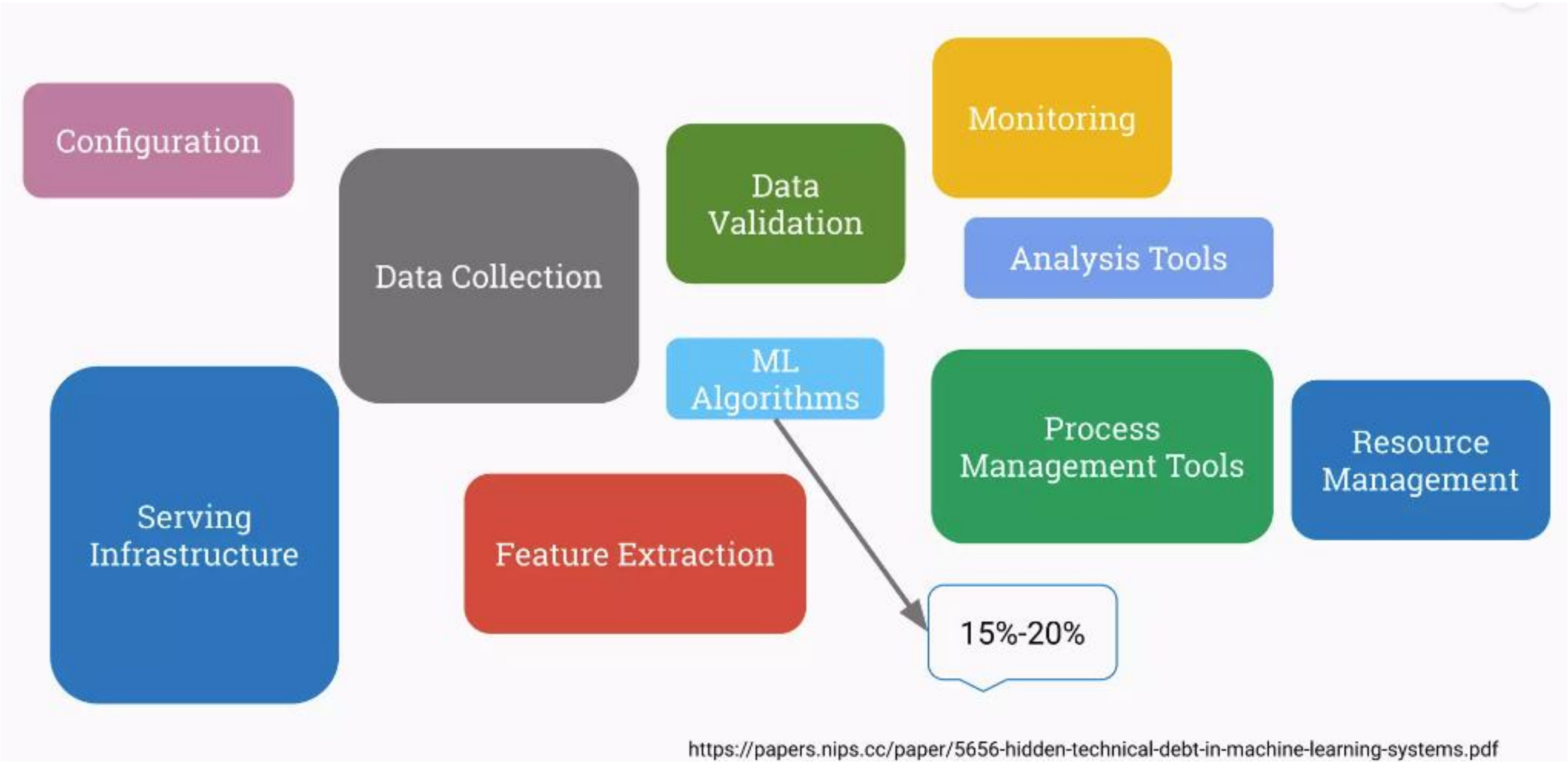
In simpler terms, AI is the field of study focused on creating machines that can think and learn like humans.

Fun fact: British mathematician Alan Turing proposes the “Turing Test” to evaluate a machine's ability to exhibit human-like intelligence and played with the concept of AI back in 1950.

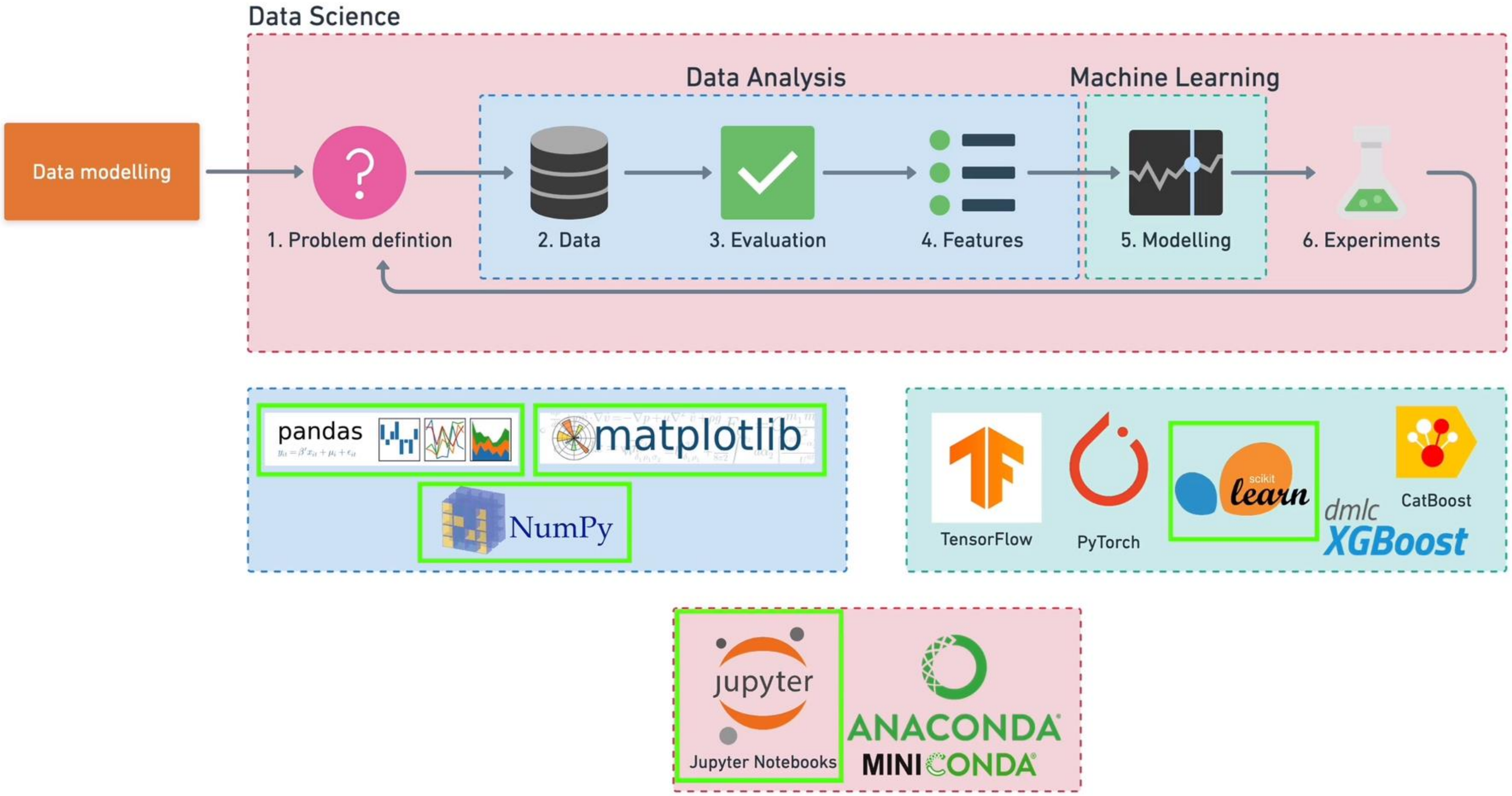


WHAT PEOPLE THINK

REALITY

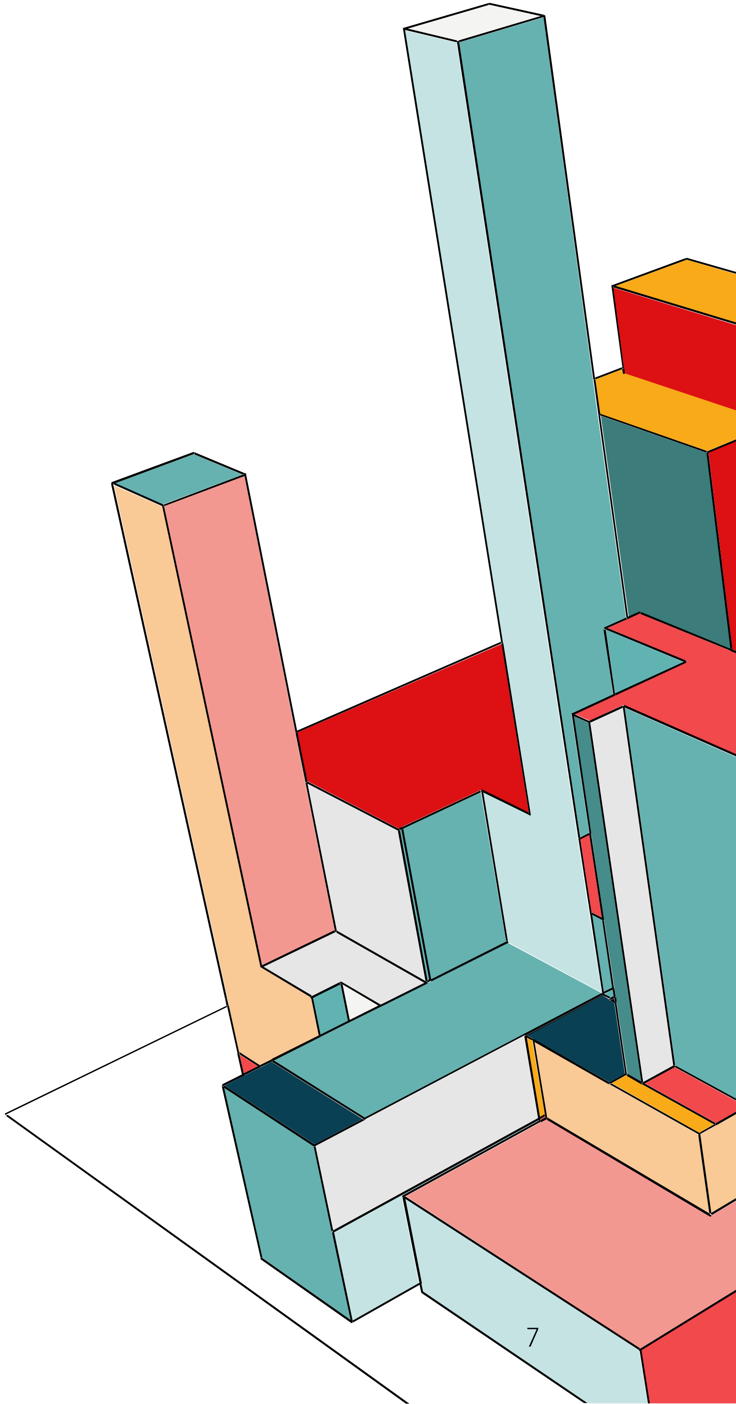

Configuration
Data Collection
Data Validation
Monitoring
Analysis Tools
ML Algorithms
Process Management Tools
Resource Management
Serving Infrastructure
Feature Extraction
15%-20%
https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Tools you can use



COMMON LIBS / FRAMEWORK

- NumPy: Used for high-performance mathematical operations, array processing, and serving as the backbone for numerical computations in Python.
- Pandas: Ideal for data manipulation and cleaning, especially when dealing with tabular data (like Excel spreadsheets), enabling easy data analysis.
- Matplotlib: Used for creating data visualizations, such as graphs and charts, to help in understanding data trends and patterns.



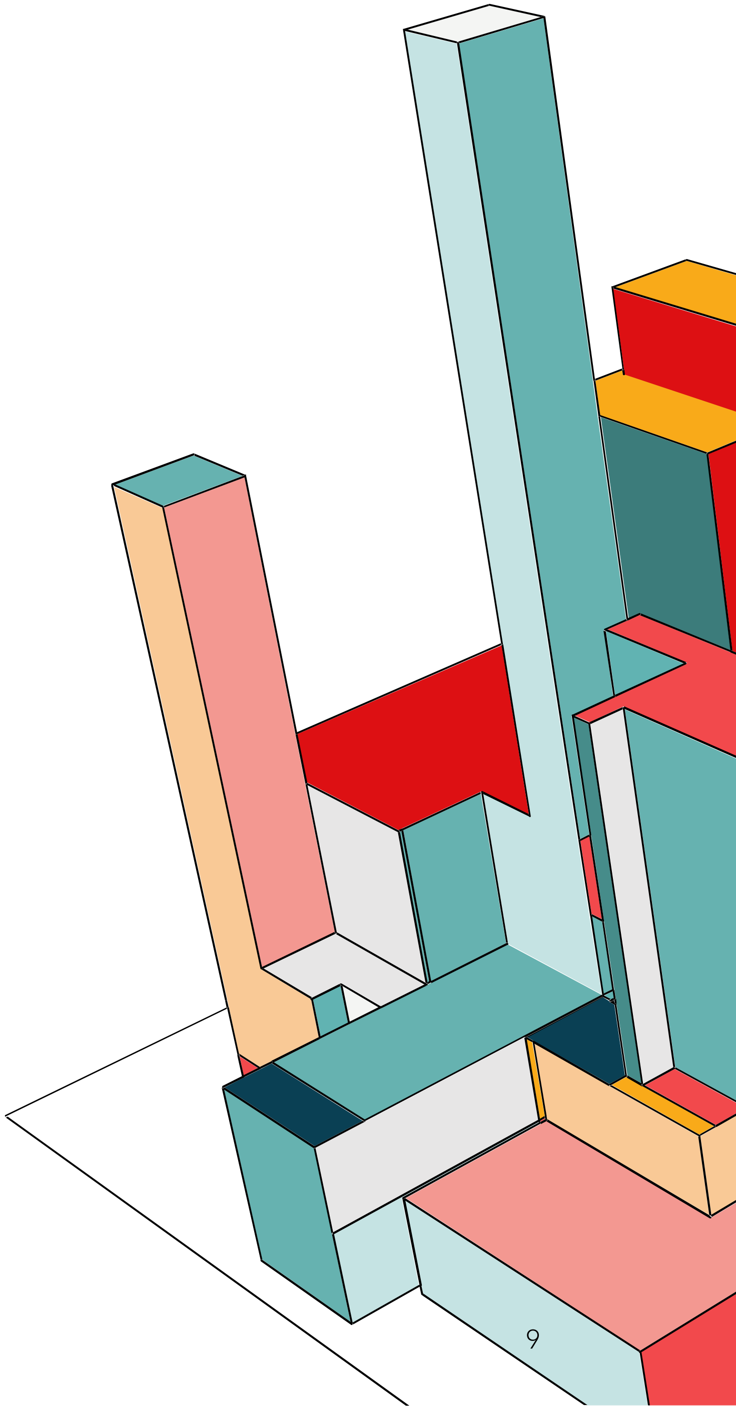
COMMON LIBS / FRAMEWORK

- Scikit-learn: Perfect for implementing machine learning algorithms, such as classification, regression, clustering, and dimensionality reduction, with a focus on model building and evaluation.

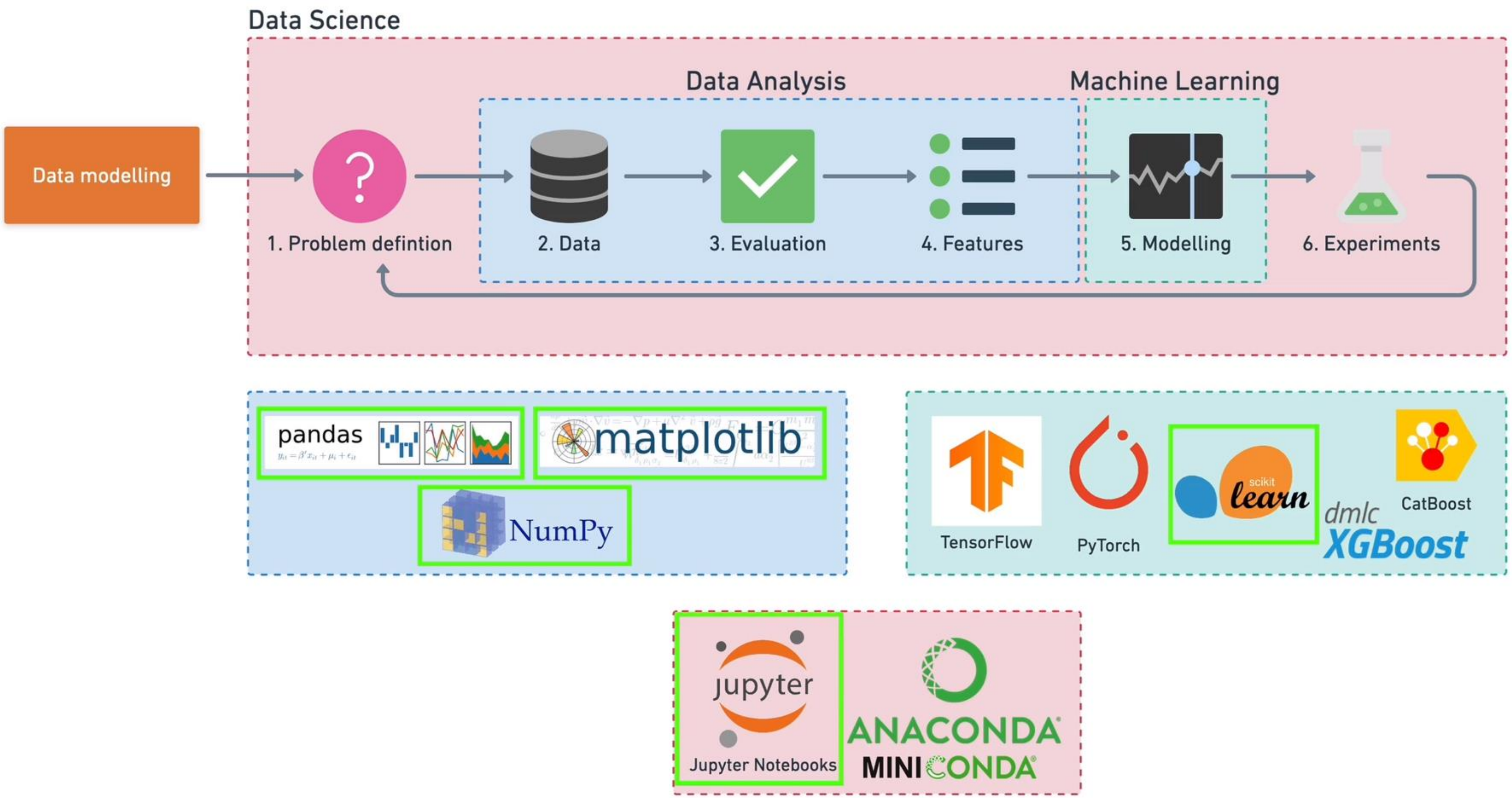
- TensorFlow: Utilized for building and training neural networks for deep learning applications, including image and speech recognition, and natural language processing tasks.

IDES / NOTEBOOKS

- PyCharm – from jbrains.
- Google Colab - colab.google.
- Jupyter Notebook.



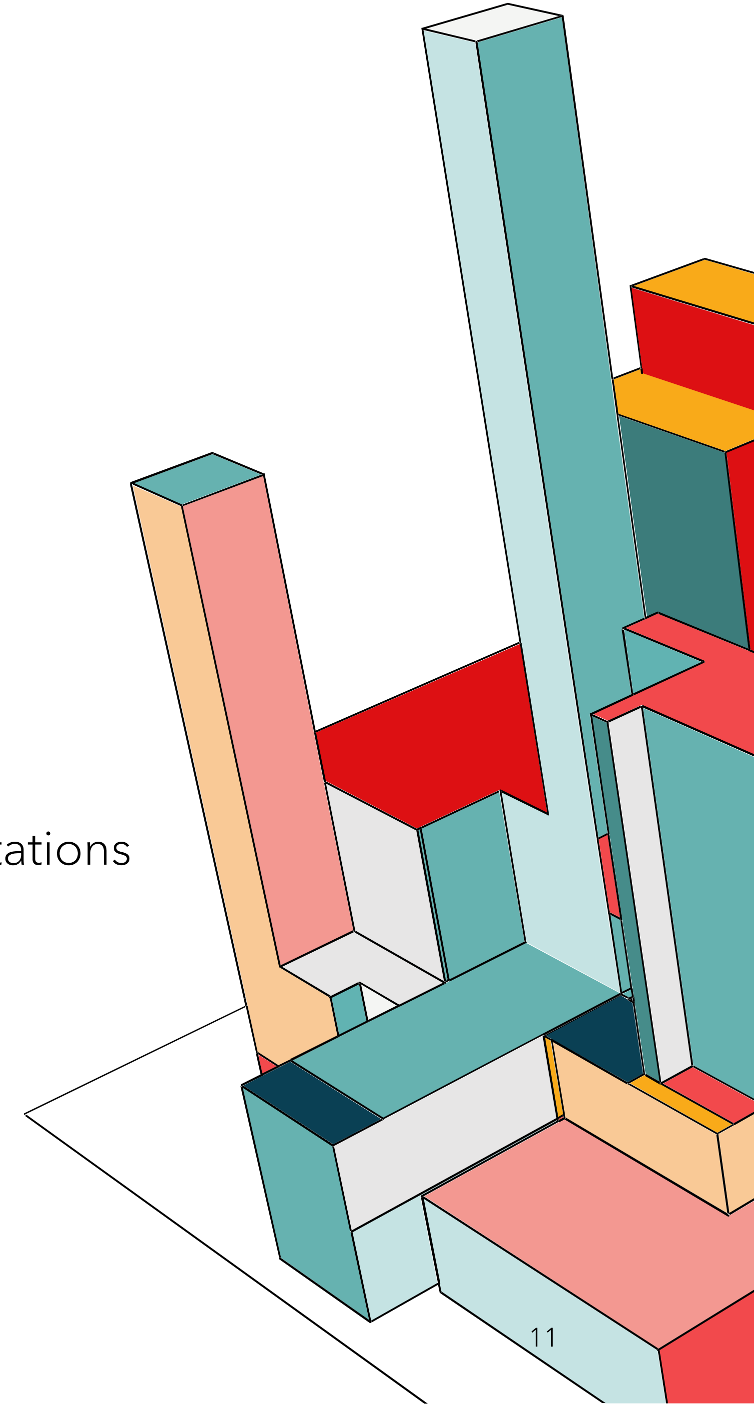
Tools you can use



HARDWARE

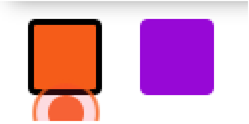
- CPU (Central Processing Unit):
  - The brain of the computer, handling general-purpose tasks.
  - Excels at sequential task processing.
  - Ideal for tasks requiring complex decision-making and flexibility.

- GPU (Graphics Processing Unit)
  - Originally designed for rendering graphics.
  - Specializes in parallel processing, handling thousands of tasks simultaneously.
  - Significantly accelerates deep learning tasks by distributing computations across many small cores.
  - Essential for training complex neural networks efficiently.

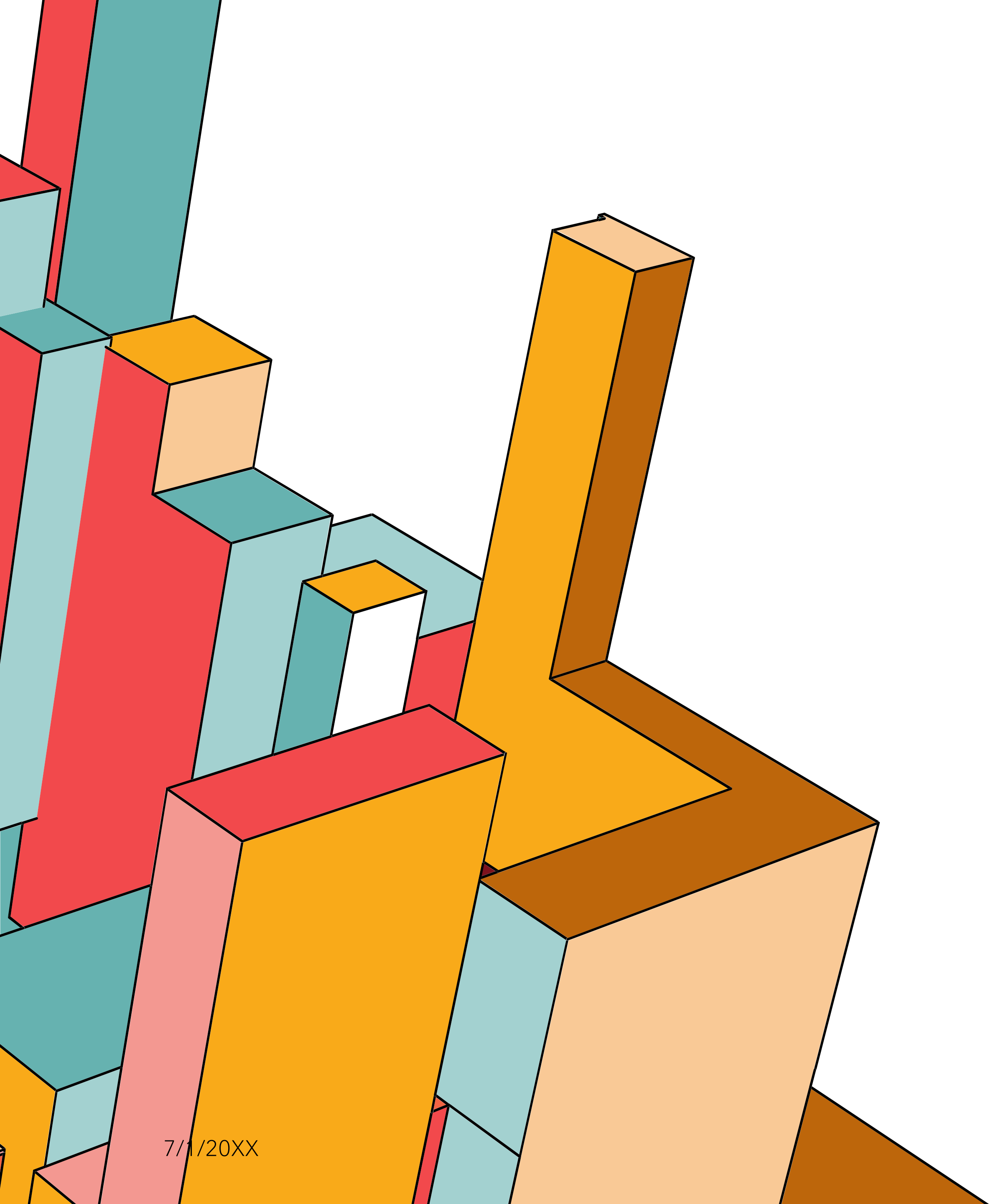


FUN TIME! HOW MODELS “SEE”

Machine Learning Playground (ml-playground.com)



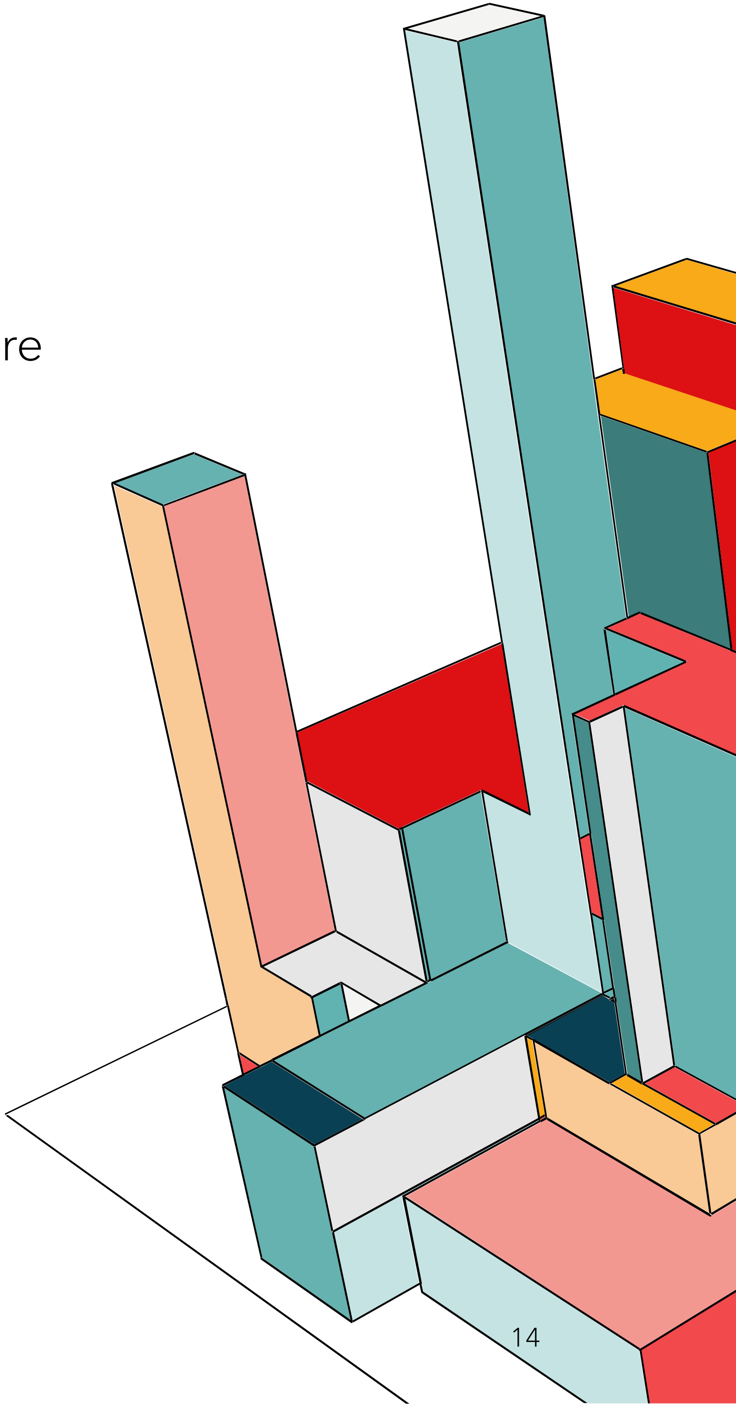
- Let’s play spam not spam!
- Orange will be non-span email, and purple spam email.
- The Y axis will be the email length and X will be number of links in the email.
- Place in the grid samples (span and non-spam).
- Select a model and see how the model “see” the data and how it splits it into two groups (spam and non-spam) by learning some pattern.

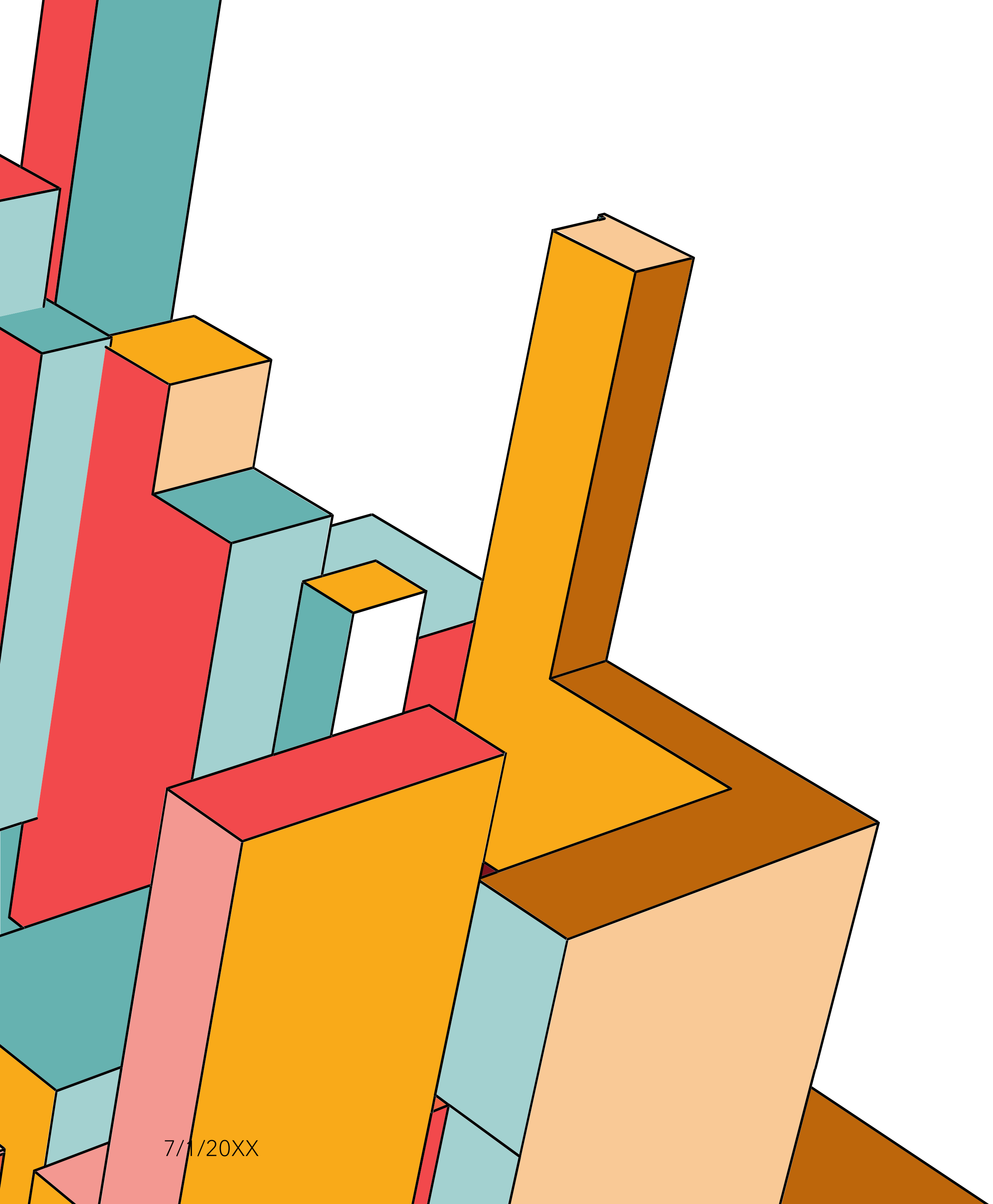


PROBLEM ANALYZATION

PROBLEM ANALYZATION

- Understand the Context: Know the objective of your analysis; what questions are you trying to answer?
- What is the solution options.
- What is needed to made the solution work.





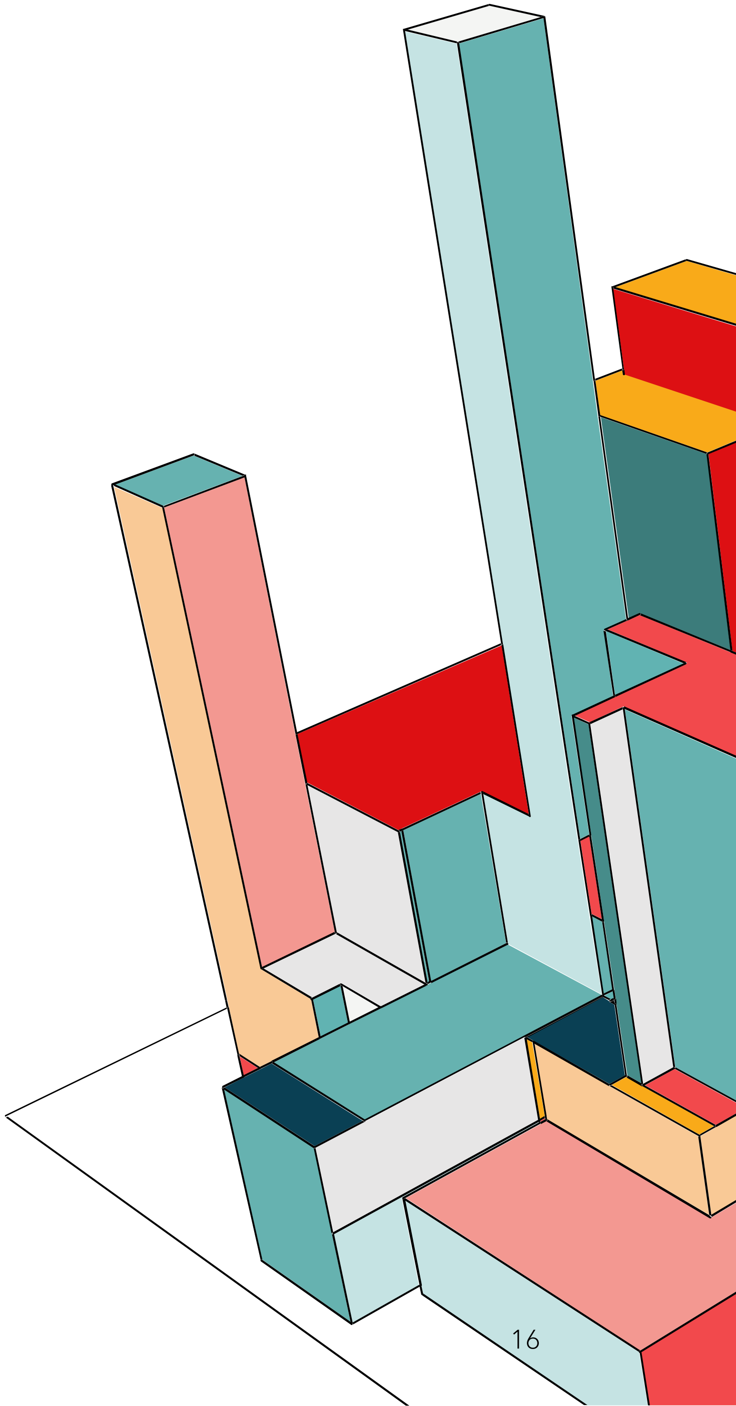
DATA

WHAT IS “DATA”?

- A collection of samples – like employees list, collection of pictures etc.
- Can be in variety of different formats – from CSVs to mp4.
- Can come from in variety of different sources – from signup forms to cameras.

[2]:

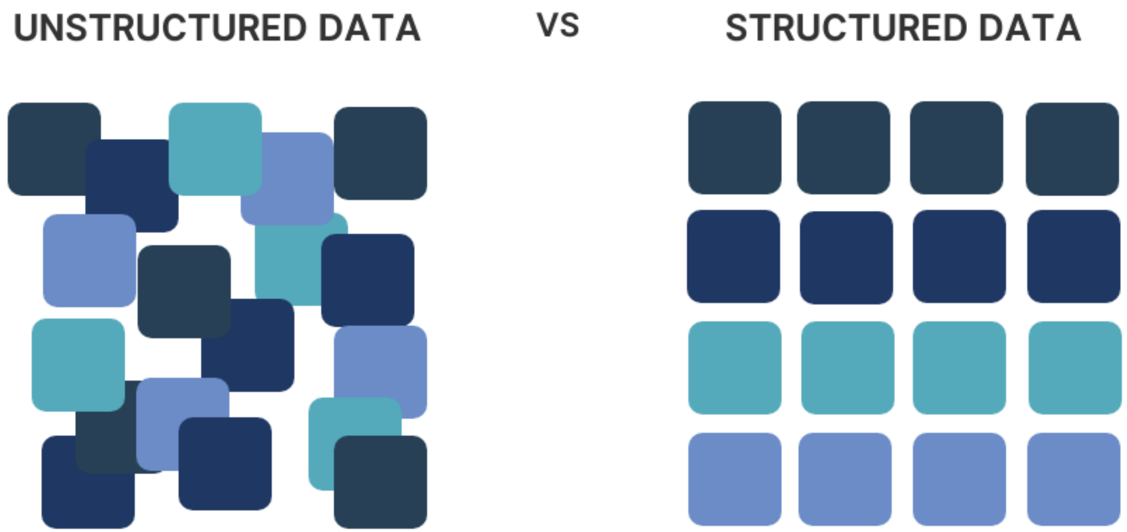
	id	age	name	last_name	sex	married	income	childrens	weight	height
266	267	31	Mary	Hodges	Male	Yes	36740.251100	2.0	79.544655	177.601313
201	202	30	Donald	Houston	Male	Yes	31389.980684	2.0	105.232239	177.720619
639	640	69	Cynthia	Ross	Male	Yes	34995.310579	1.0	73.408352	179.210756
567	568	48	Tim	Cruz	Male	Yes	23228.879691	2.0	68.674324	181.567296
419	420	77	April	Ward	Male	No	34289.140408	3.0	65.132407	177.575654
749	750	22	Donna	Harrington	Female	Yes	24007.373914	NaN	58.925293	NaN
572	573	21	Maria	Roach	Male	Yes	30712.367473	1.0	54.178356	191.492334
726	727	50	Kevin	Allen	Female	No	18856.159823	0.0	57.046813	158.417312
149	150	67	Dennis	Wallace	Male	No	42866.933323	1.0	52.699942	160.224293
683	684	61	Cindy	Robertson	Male	No	36844.367564	1.0	97.451476	176.770999



TYPES OF DATA

UNSTRUCTUED DATA

Unstructured data refers to information that doesn't follow a specific format or structure, such as text, images, and videos.

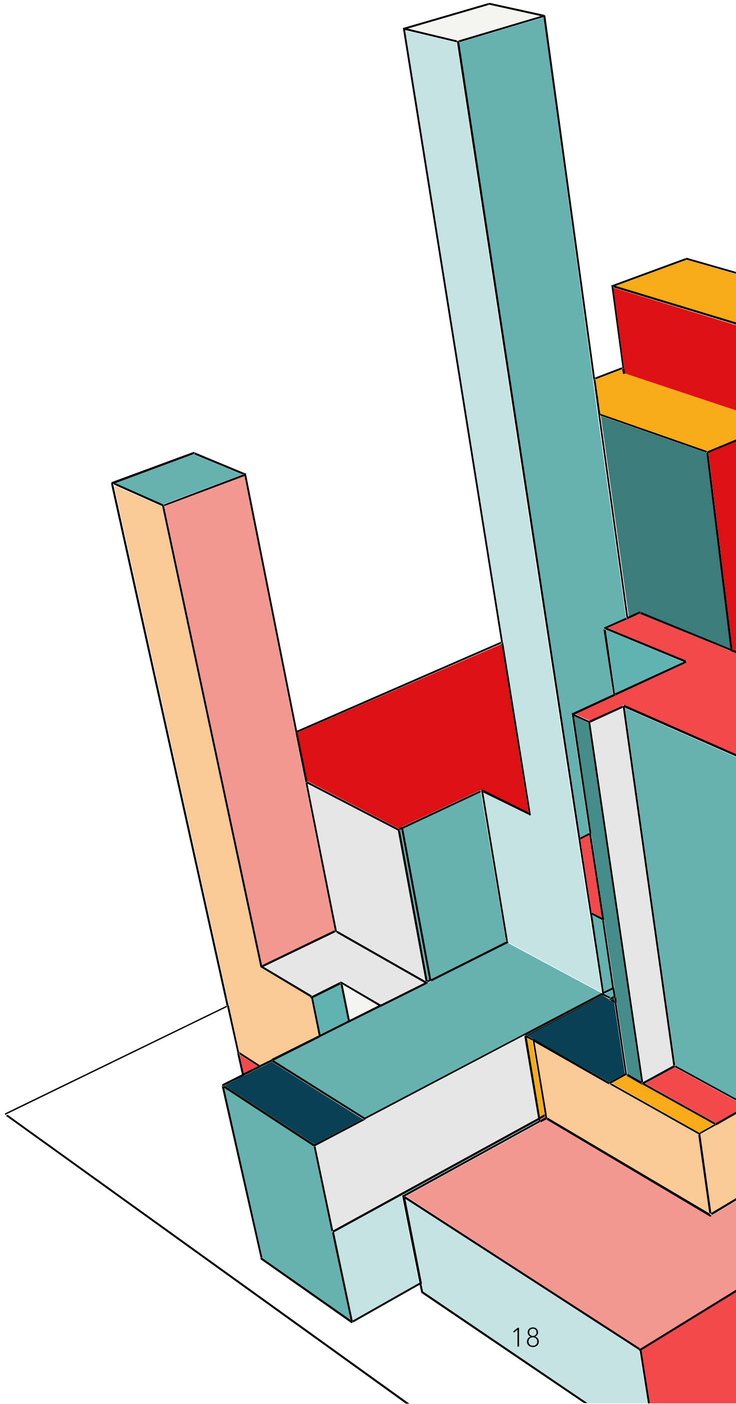


STRUCTUED DATA

Structured data is organized like databases or spreadsheets, featuring columns and rows.

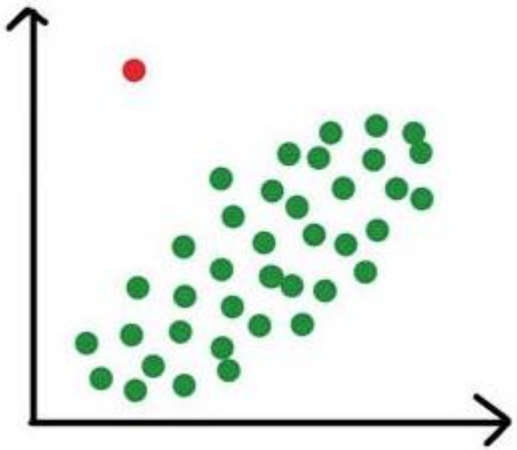
ANALYZE THE DATA

- Collect Data: Gather the necessary data from reliable sources.
- Clean Data: Remove inaccuracies and inconsistencies to ensure data quality.
- Explore Data: Use statistical methods and visualization to identify patterns and insights.
- Interpret Results: Draw conclusions based on the data exploration and statistical outcomes.



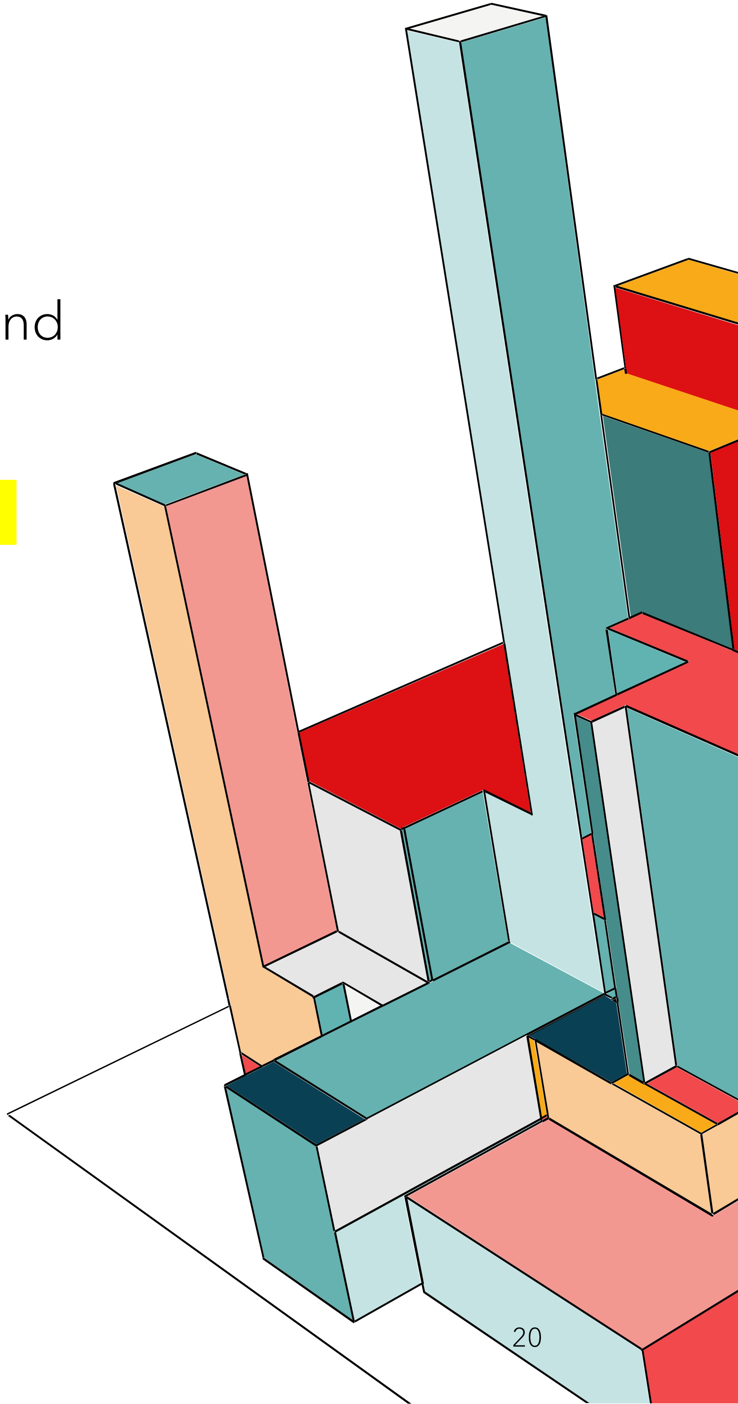
CLEAN DATA = HAPPY MODEL

- Clean data makes models learn better.
- Like humans, clean data without any mistakes helps us understand better.
- Dirty data can be – null values, an uneven number of samples, values that make no sense, useless data/features, outliers etc.

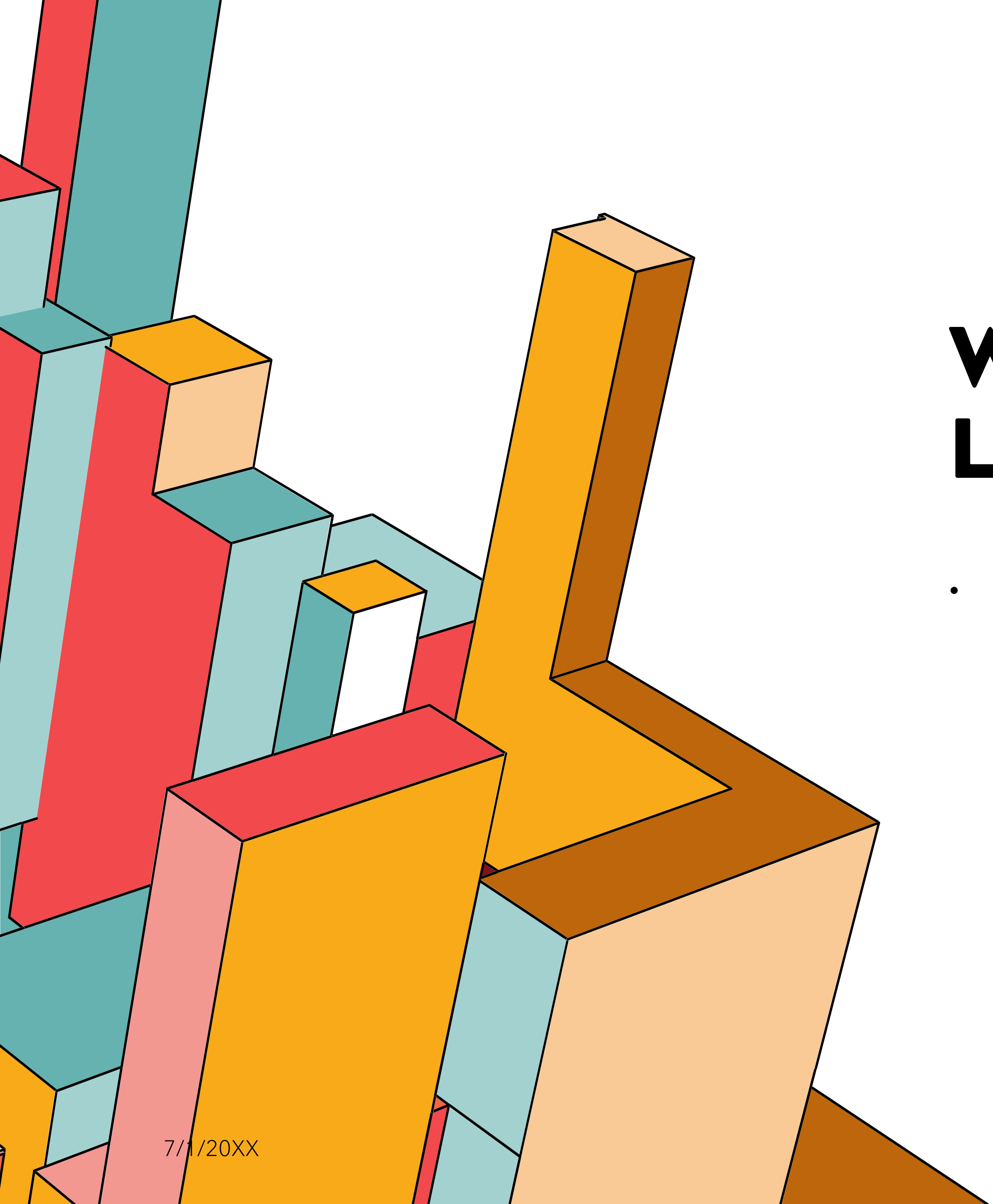


EXPLORING DATA

- Visualize Data: Create graphs and charts to see patterns, trends, and outliers.
- Summary Statistics: Calculate mean, median, mode, and standard deviation to understand data distribution.
- Identify Gaps: Check for missing values or data inconsistencies.



TASK 1 PART 1 & 2



WHAT IS MACHINE LEARNING?

- Focuses on developing wide range of algorithms and models to enable computers to “learn” and make decisions or predictions.

TYPES OF LEARNING

Supervised Learning

Unsupervised Learning

Reinforcement Learning

SUPERVISED LEARNING

- Involves training algorithms on labelled data, where both input features and desired output (label) are provided.
- Common tasks include classification and regression.

REAL-WORLD EXAMPLES OF ML APPLICATIONS:

Supervised Learning

Spam detection in email

Medical diagnosis

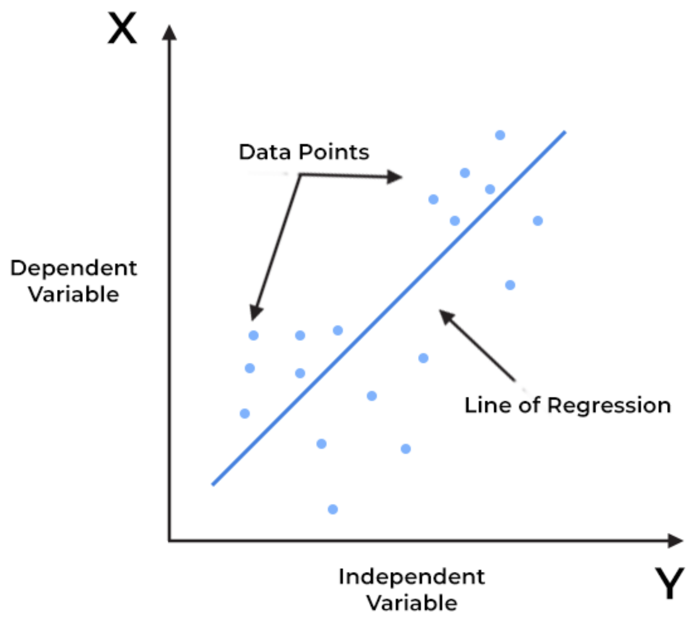
House price prediction

LINEAR REGRESSION

- What's the Idea?
  Think of it like predicting the price of a house based on its size. If you've ever seen a graph with a bunch of dots and a straight line that tries to fit through them, that's basically linear regression.

- Why Do We Use It?
  If you know the size of a house, you can make a good guess about how much it should cost. Linear regression helps us draw a line of "best fit" to make these guesses as accurate as possible.

- How Does It Actually Work?
  We take a ruler and try to draw a line among all the dots, which represent houses sold. This line is our formula, and we move it around until it's the best summary of all the dots together.



Best Fit Line for a Linear Regression Model

LINEAR REGRESSION

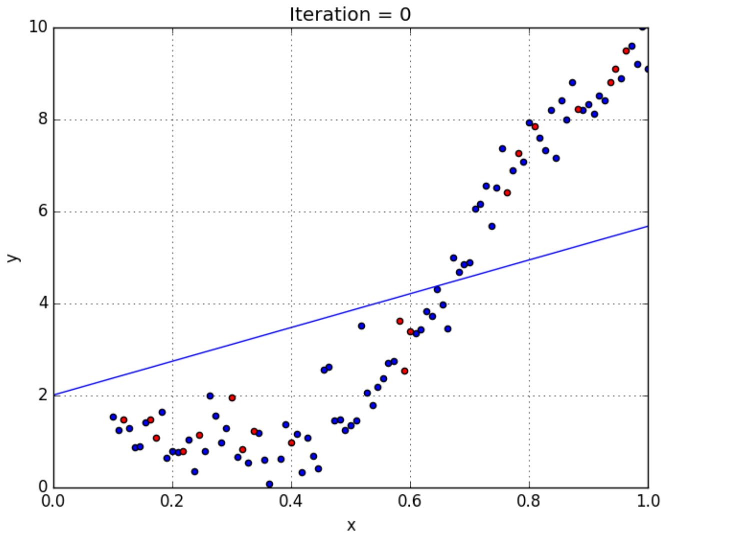
- Error: How Close Are We?
  The "error" is just a fancy term for the difference between our predicted price and what the house actually sold for. If we guess a house with 1,000 square feet sells for $200,000 but it sells for $220,000, our error is $20,000. We jiggle the ruler (our line) until these errors are as small as they can be across all the houses.

- Example: Predicting House Prices - Let's say you have a bunch of data on houses.

- Features (what we know): The size of the houses (in square feet).

- Target (what we're guessing): The selling price of the houses.

- You scatter all this data on a plot with house size on the horizontal line and price on the vertical one. Linear regression is like drawing a trend line on this plot to help us predict the price of a house we've never seen before based on its size.

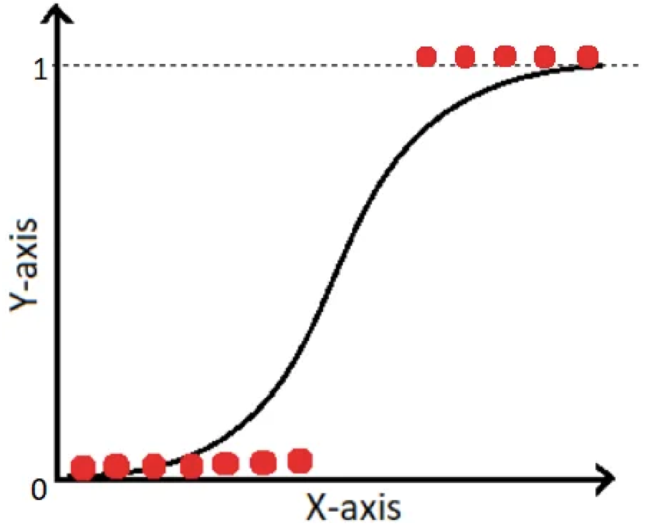


LOGISTIC REGRESSION

- What's it About?
  Imagine you're looking at people visiting an open house and trying to guess who will actually buy it. Logistic regression is like a magic scale that weighs the features of the house and the visitors' preferences to predict "will buy" or "won't buy."

- Why Use Logistic Regression?
  We use logistic regression when our question has a "yes" or "no" answer. In our case: "Will this visitor buy the house?" Yes or No.

- How Does it Function?
  It's like drawing a line that divides the visitors into two groups: those likely to buy and those who are not. This line is not straight but curved, perfect for sorting things into two buckets.



LOGISTIC REGRESSION

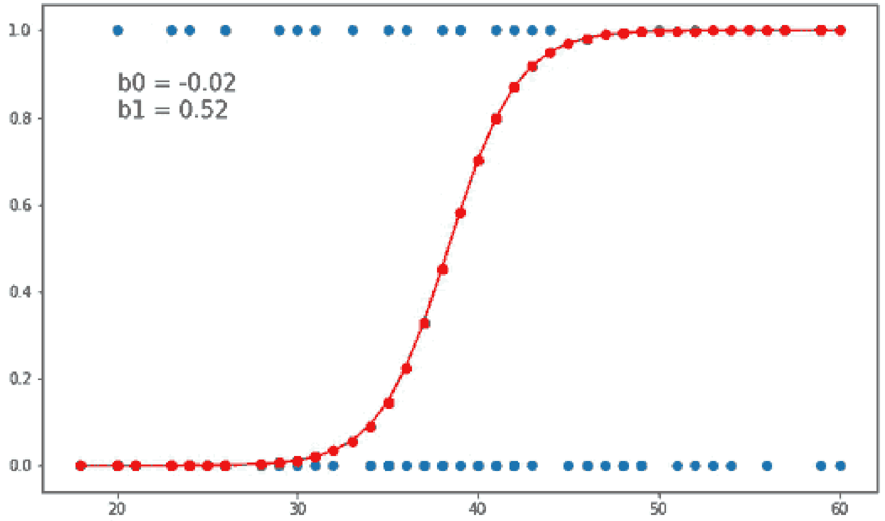
- Error: Checking Our Guesses
  The "error" here is about how many times we guess wrong. If we predict a visitor will buy, but they walk away, that's an error. We keep adjusting our magic scale (the curve) to be wrong less often.

- Example: Will They Buy the House?
  Consider you have data on past visitors:

- Features (what we know): Visitors' income, their current home size, number of children, distance from work.

- Target (what we're predicting): If they will buy the house or not.

- We plot visitors on a graph with these features and use logistic regression to draw our special curve (S-shaped). It predicts if new visitors with similar features will buy the house.



UNSUPERVISED LEARNING

- Unsupervised learning involves training algorithms on data without specific labels, aiming to find patterns or structures within the data itself.
- Common tasks include clustering (grouping similar data points) and dimensionality reduction (simplifying data to its most important features).

REAL-WORLD EXAMPLES OF ML APPLICATIONS:

Unsupervised Learning

Customer segmentation

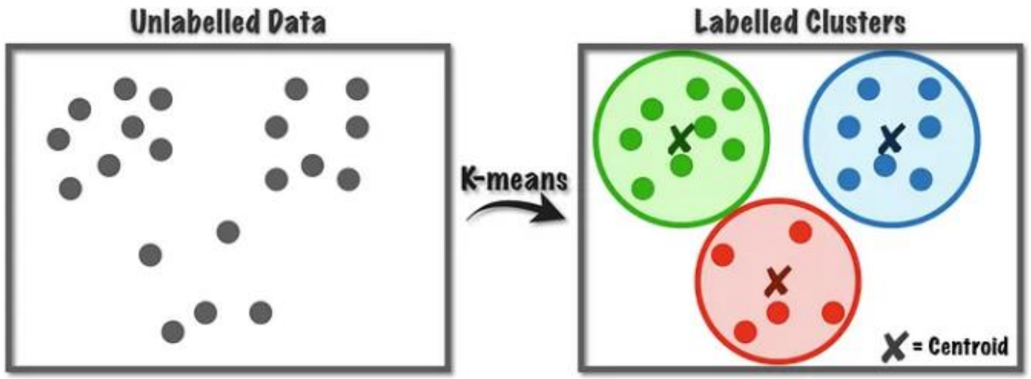
Anomaly detection

K-MEANS CLUSTERING

- What's the Idea?
  Imagine you have a huge list of houses with different features and you want to organize them into neat categories to make sense of the market. K-Means Clustering is like a smart organizer that helps us find the best categories automatically.

- Why Use K-Means Clustering?
  K-Means is perfect when we have a lot of data and we need to find a way to group similar things together. For example, it can help real estate agents to understand what kinds of houses appeal to different groups of buyers.

- How Does It Work?
  It's like throwing a bunch of colored balls on the floor (each color is a feature like size, price, or location) and then using a magnet (the K-Means algorithm) to pull together balls of similar colors into piles. Each pile represents a group of houses with similar features.



K-MEANS CLUSTERING

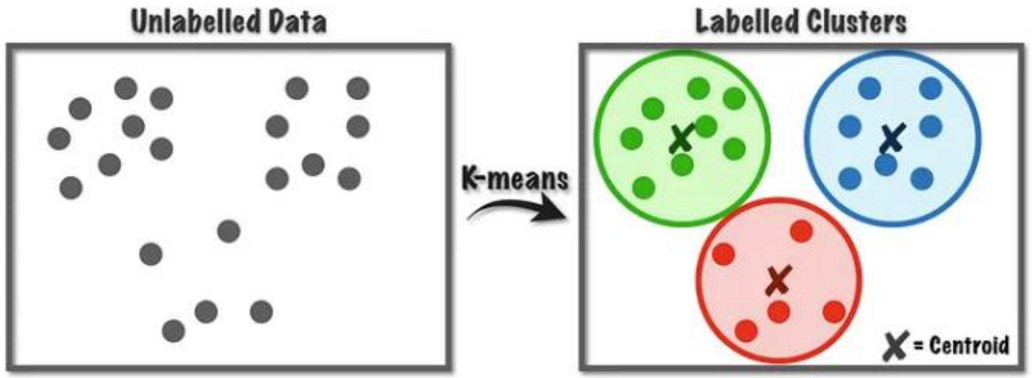
- Error: Ensuring Tight Groups.

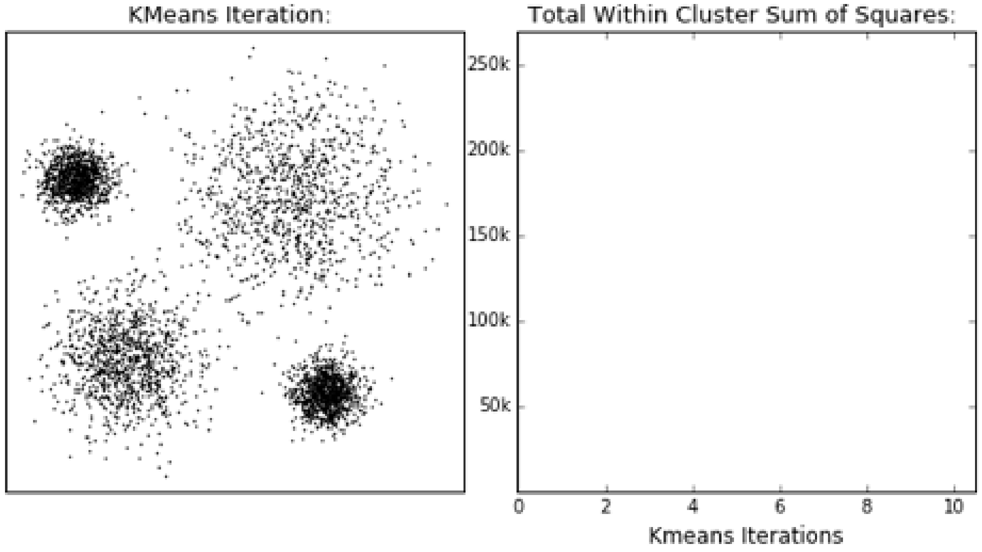
- Example: Categorizing Houses for Marketing
  Suppose a real estate company has a large database of houses and wants to market them effectively:

- Features (the 'colors'): Size, price, number of bedrooms, age of the house, etc.

- Goal (the 'piles'): To identify distinct groups of houses that can be targeted with specific marketing strategies.

- We use K-Means to find these groups. The algorithm looks at all the features and starts grouping houses that are alike into clusters.





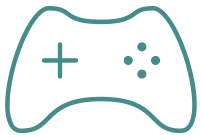
REINFORCMENT LEARNING

- Reinforcement learning involves training algorithms through a system of rewards and punishments. It learns to make decisions by taking actions in an environment to maximize some notion of cumulative reward.
- Common tasks include navigating mazes or playing games, where the algorithm learns from trial and error.

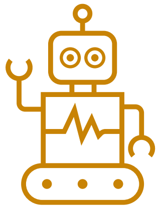
REAL-WORLD EXAMPLES OF ML APPLICATIONS:

Reinforcement Learning

Autonomous vehicles



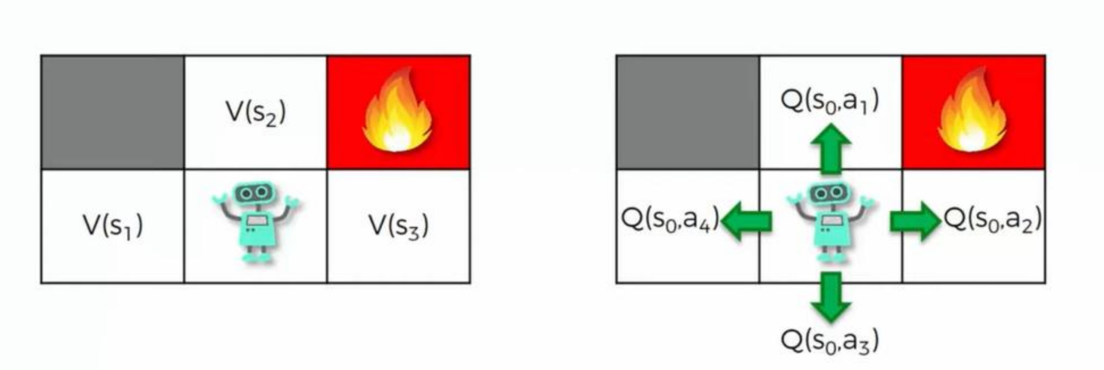
Game playing



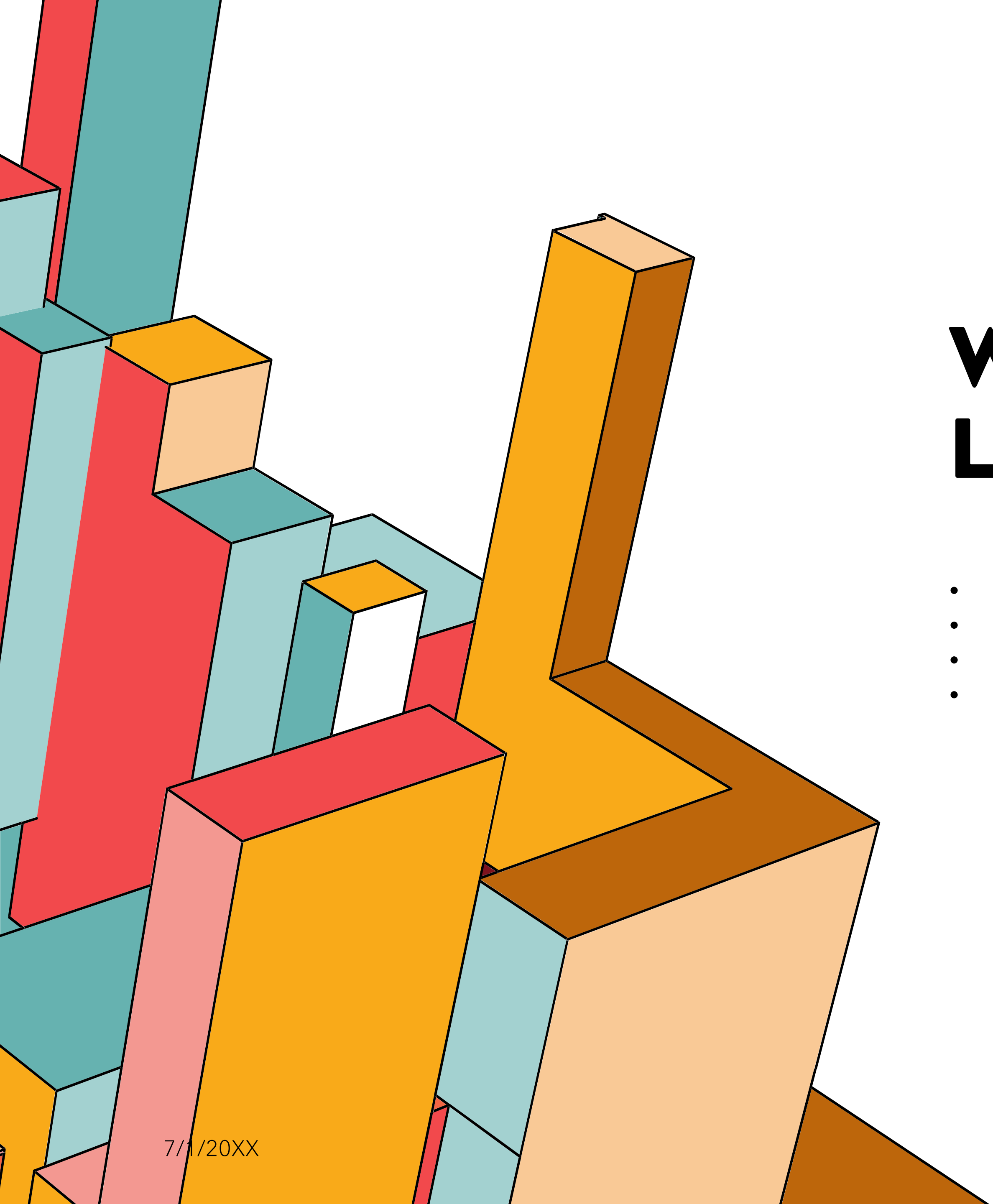
Robotics

Q-LEARNING

- Goal: Learn optimal decision-making in a given environment by estimating the value of taking an action in a particular state.
- Method: Learns a Q-function, which maps state-action pairs to expected future rewards, and uses it to select actions that maximize cumulative rewards.
- State-action pairs: A situation (state) and a choice (action) made by an agent. The agent learns which actions give better results in different situations.
- Reward: A score received after taking an action, guiding the agent's choices. Positive rewards are good, negative rewards are bad.



TASK 2



WHAT IS DEEP LEARNING?

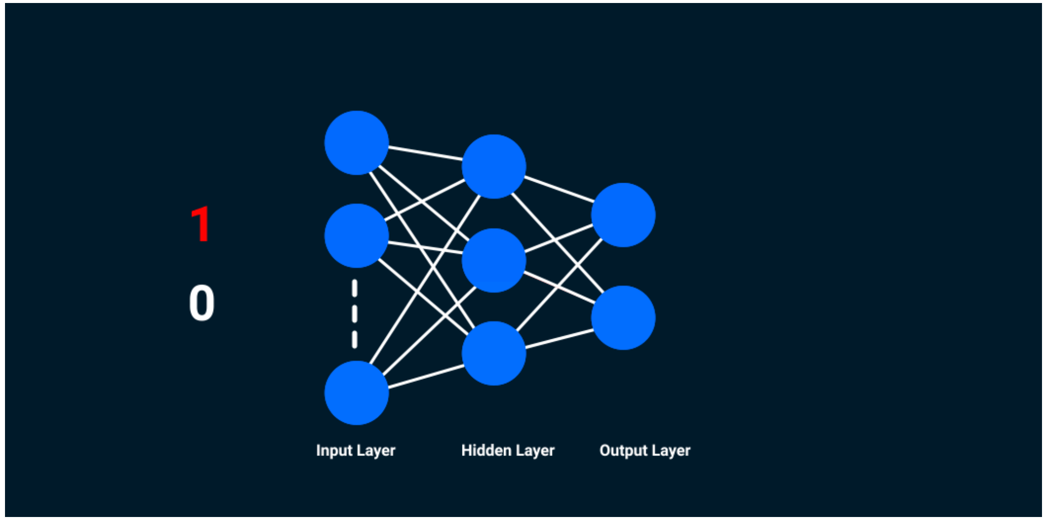
- A subfield of ML.
- Focuses on using neural networks.
- Solves verity of problems like ML, but usually gives better results.
- Can be “costly” on resources.

NEURAL NETWORK

- Neural Network: A computer system modelled on the human brain's network of neurons.

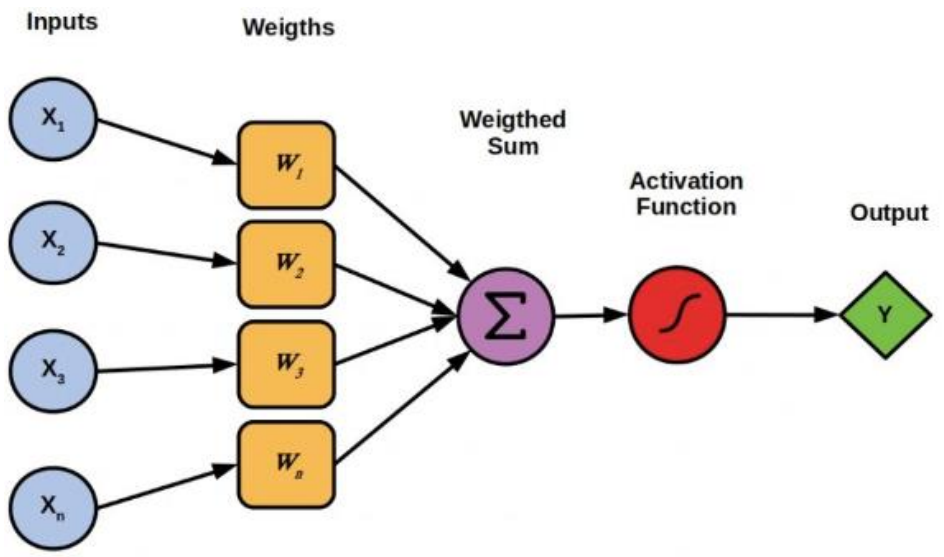
- Learning Method: Learns tasks by analysing examples, much like a student learns through practice and correction.

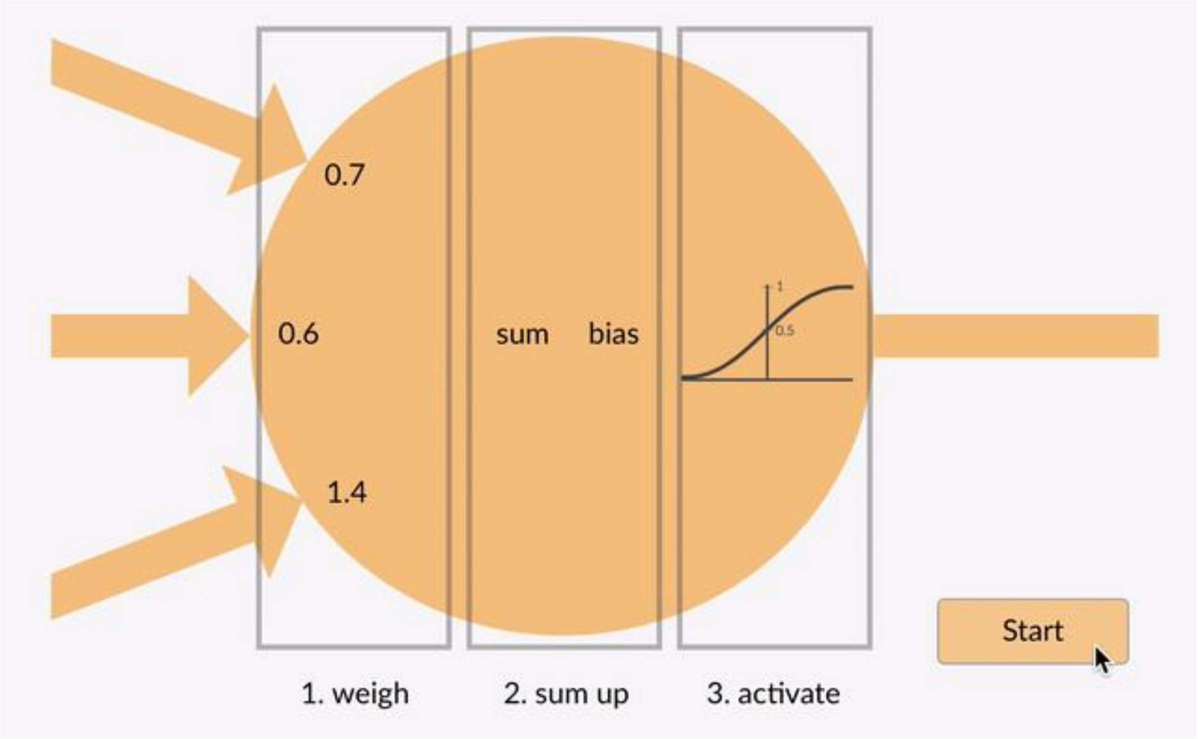
- Functionality:
  - Receives input (e.g., photos).
  - Makes a prediction (e.g., cat or dog).
  - Gets corrected if wrong.
  - Adjusts its "thought process" based on feedback.



THE PERCEPTRON

- Structure: A simple, single-layer neural network with an input layer and an output layer (no hidden layers).
- How it works: keeps wights for each feature + bias, do the multiplication and sum, then use step function to get result.





FEEDFORWARD NEURAL NETWORKS

- What is a Feedforward Neural Network? Imagine a factory assembly line: each worker (neuron) does a specific task and passes the work to the next person in line without ever going backward.

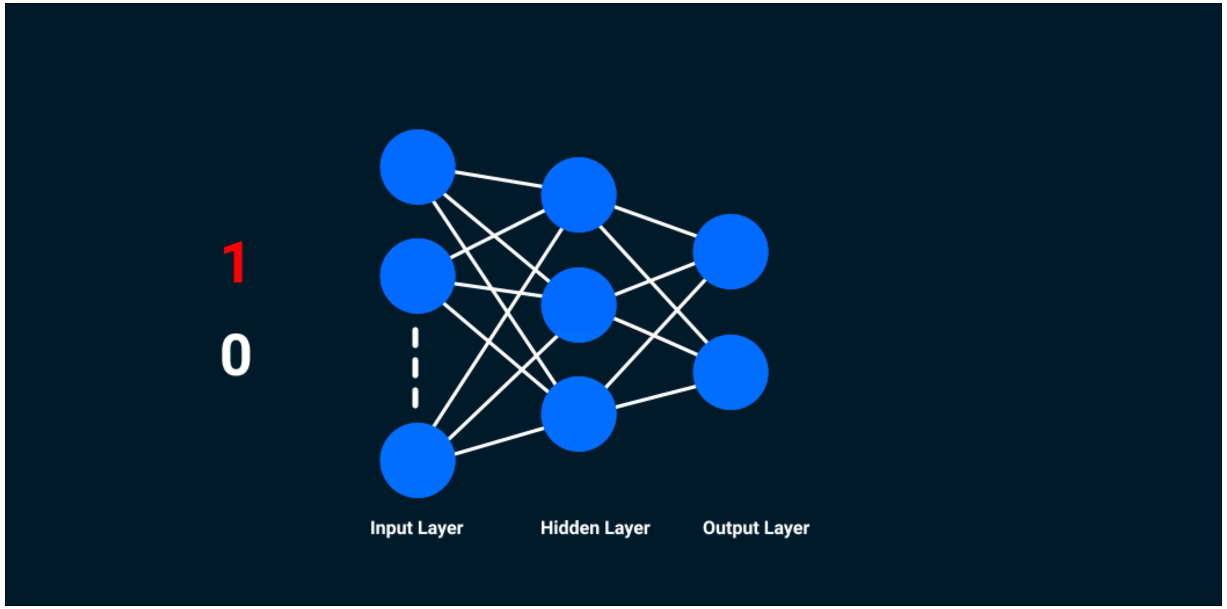
- A Feedforward Neural Network operates in a similar way. Information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and finally to the output nodes.

FEEDFORWARD NEURAL NETWORKS

- Layered Structure:
  Input Layer: Where you feed in the data, like the characteristics of a house (size, location, number of rooms).

- Hidden Layers: These are the 'workers' in between that do all the complex computations. They take in the data from the input layer, perform calculations, and pass the results forward. There can be one or many hidden layers.

- Output Layer: This produces the final result, such as predicting the price of a house.

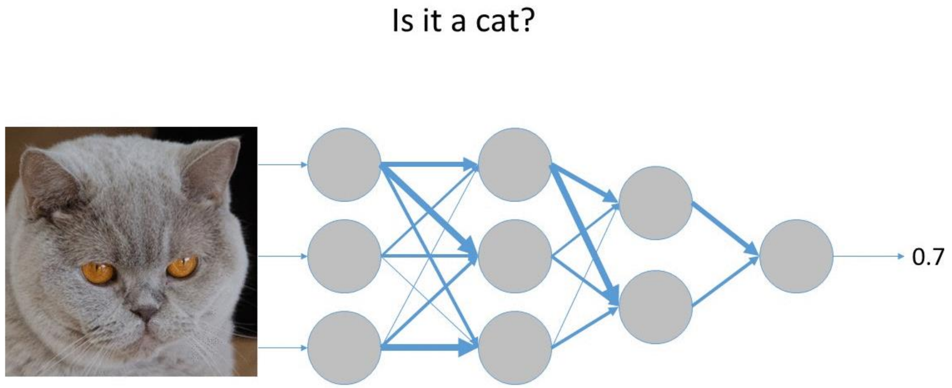


FEEDFORWARD NEURAL NETWORKS

- Neurons and Weights:
  Each neuron in a layer is connected to every neuron in the next layer through 'weights,' which are like the strength of the connection.

- Activation Functions:
  Each neuron applies an 'activation function’ to the data it receives before passing it on. This function can help decide how much to forward to the next layer, like a filter.

- Training the Network:
  Learning: FNNs learn by adjusting the weights based on the errors in predictions. This is done through a process called backpropagation, where the network adjusts its weights to minimize the error between the predicted output and the actual output.
  Epochs: This learning happens over many cycles, called epochs, where the network sees the data many times, getting a little better each time.



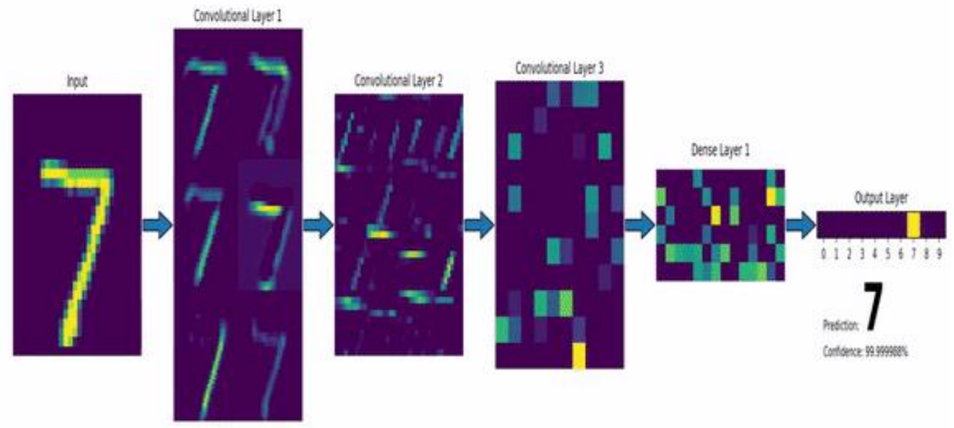
THE MAIN FIELDS OF DEEP LEARNING

Vision

NLP

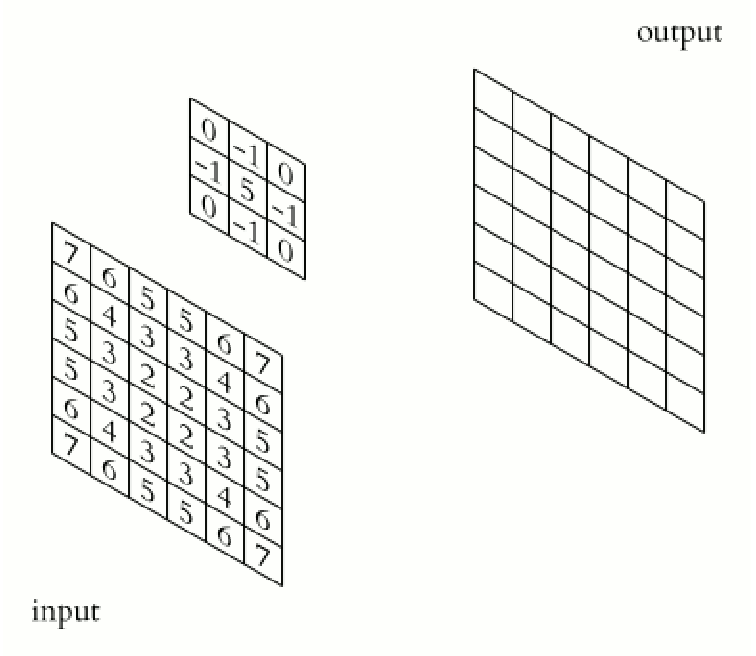
VISION

- What is it? Vision refers to the field focused on allowing machines to see or interpret the visual world as humans do. This includes recognizing objects, faces, and scenes in images or videos.
- How is it solved?
  - Convolutional Neural Networks (CNNs): The go-to neural network type for vision tasks. They work by applying filters that capture spatial hierarchies of features in images (like edges, textures, and objects).
- Example:
  - Image classification where the task is to categorize an image as one of many possible objects. Suppose we have a dataset of images labelled with different types of animals. A CNN can learn to recognize patterns (fur texture, shape of ears) that define each animal type and accurately classify unseen images.



CONVOLUTION

- Convolution: A mathematical operation that applies a “filter” to local area of the input to learn patterns.
- Filters: Small matrices that slide (convolve) across the input, performing element-wise multiplication and summing the results.
- Output: A feature map that represents the learned patterns.

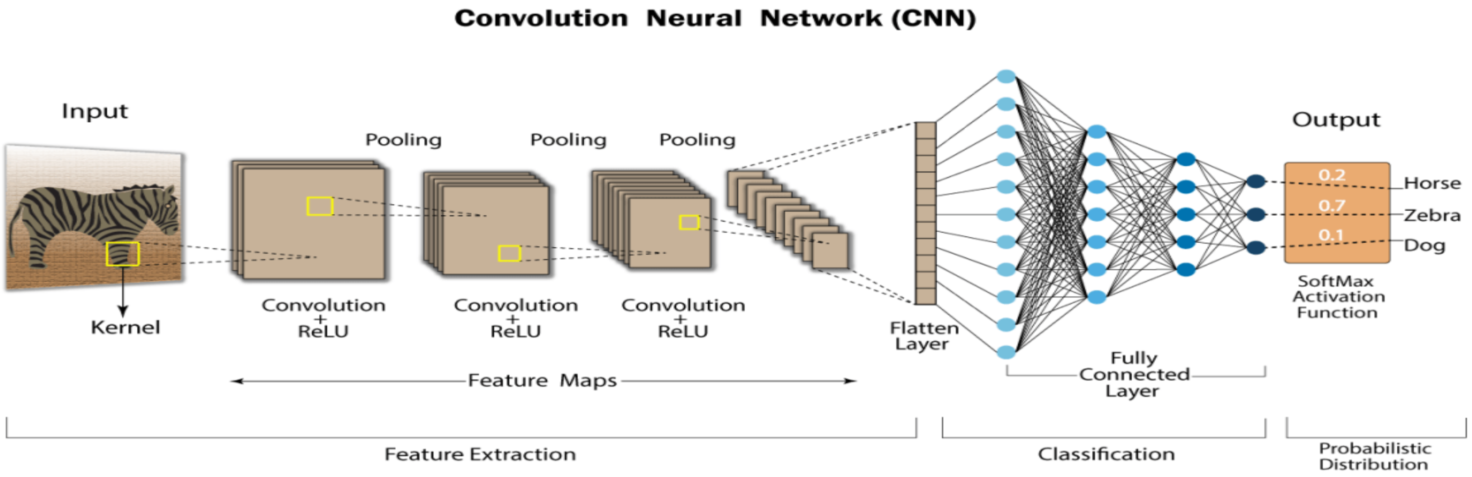


Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

CONVOLUTIONAL NEURAL NETWORKS

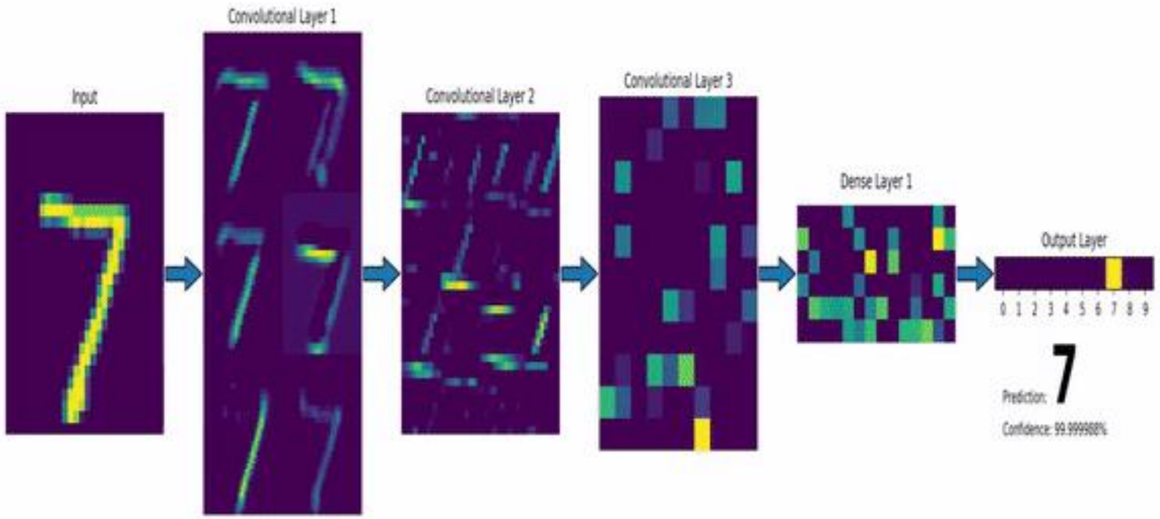
- What's a Convolutional Neural Network (CNN)?
  Imagine you have a pile of different photos, and you want a computer to sort them into categories, like photos of beaches, cities, and forests. A CNN is a smart tool for this job. It's like a detail detective that looks for specific clues in the pictures to figure out what category they belong to.

- How: The convolutions find patterns that are then flatten move into fully connected to perform classification or regression tasks.



CONVOLUTIONAL NEURAL NETWORKS

- How Does it Function?
  It's a bit like teaching someone to recognize a cat by showing them many pictures of cats. They notice features like whiskers, ears, and fur texture. A CNN does this with "filters" that learn to recognize these features. It scans through the image, focusing on small parts at a time, and remembers which features are important for distinguishing one category from another.



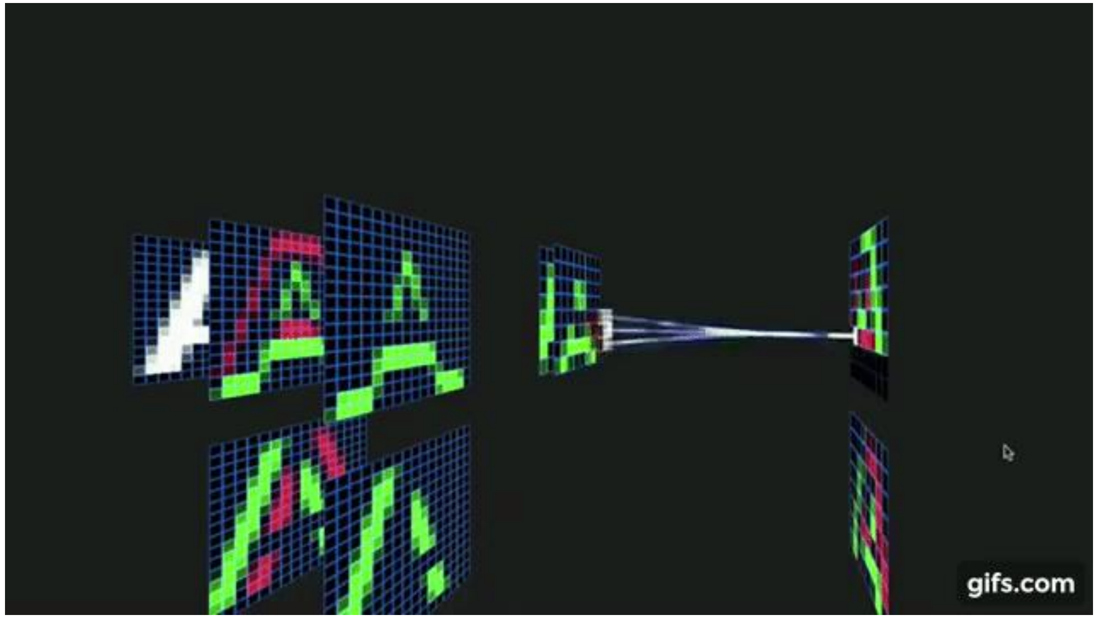
CONVOLUTIONAL NEURAL NETWORKS

- Example: Suppose you want to sort pictures of pets.

- Features: The CNN looks for features like fur patterns, ear shapes, and eye placement.

- Learning: Initially, it might not know what's important, but as you show it more pictures and tell it, "This is a cat," or "This is a dog," it starts to recognize which features are common to cats and which are common to dogs.

- Sorting: After learning, when you give the CNN a new picture it hasn't seen before, it uses what it's learned to say, "Ah, this picture has whiskers and pointy ears; it's probably a cat."

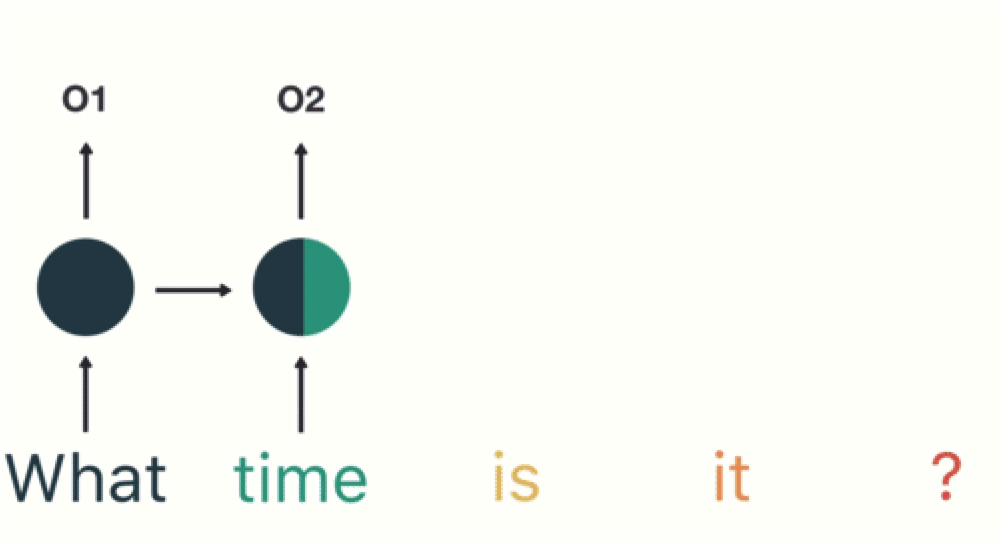


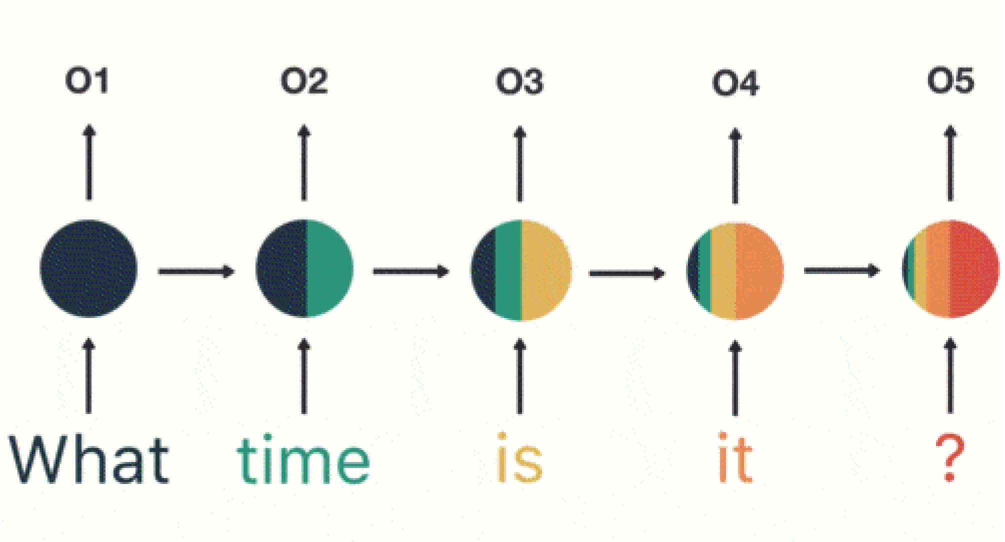
NLP (NATURAL LANGUAGE PROCESSING)

- What is it? NLP focuses on enabling machines to understand and generate human language. This can range from understanding the sentiment of text, translating languages, or even generating human-like text.
- How is it solved?
  - Recurrent Neural Networks (RNNs) and Transformers: RNNs were traditionally used for their ability to process sequential data (like sentences), by having loops within them, allowing information to persist. However, Transformers have largely taken over due to their ability to handle sequences of data in parallel, leading to more efficiency and effectiveness in tasks like translation and text generation.
- Example: Text generation where the task is to continue a given piece of text. Given the beginning of a story, a Transformer-based model like GPT (Generative Pretrained Transformer) can generate continuation of the story that feels natural and coherent.

RECURRENT NEURAL NETWORKS

- What is a Recurrent Neural Network (RNN)?
  Imagine having a conversation where you need to remember what was said before to understand the meaning of what's being said now. An RNN works similarly. It's like a smart note-taker that remembers previous information and uses it to make sense of new information.





RECURRENT NEURAL NETWORKS

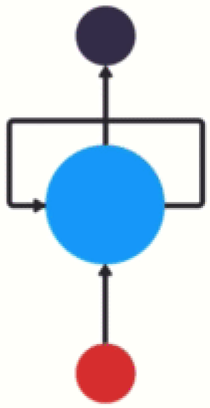
- How Does it Function?
  Think of RNNs as chain-linked compartments in a train. Each compartment (neuron) passes a message (information) to the next, but also keeps a copy of it. So when new data comes in, the RNN has the context from the previous data to make better predictions.

- Example: Predicting the Next Word
  Let's say you're trying to predict the next word in a text message. The RNN looks at the words you've typed so far:

- Previous Words (Features): The RNN remembers "I am feeling very..."

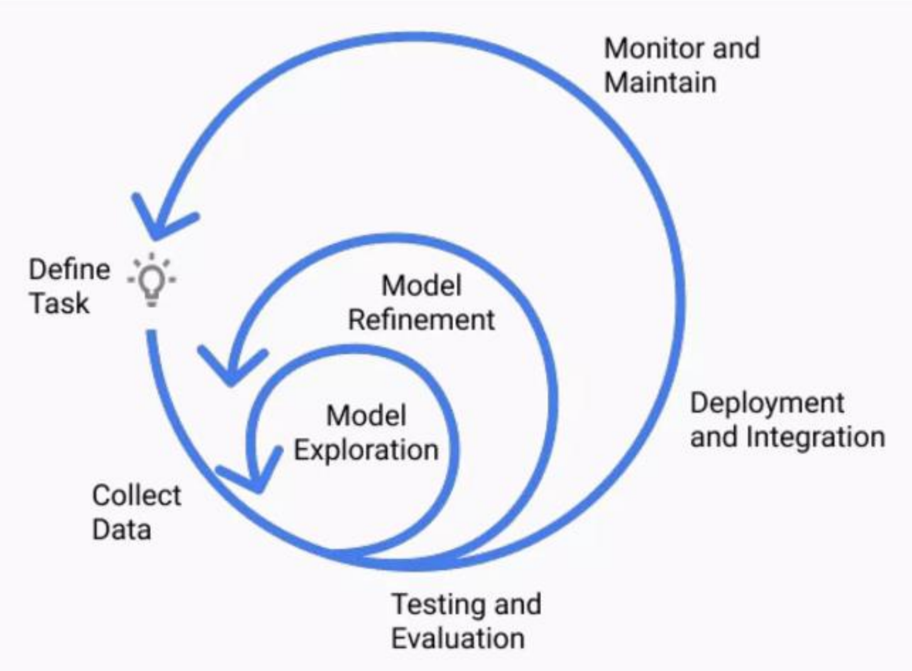
- Prediction (Target): Based on what it remembers, it predicts the next word might be "happy" or "tired."

- Output - Sequenced Prediction:
  Give it the start of a sentence, and it can often guess the next word or even complete the sentence for you.



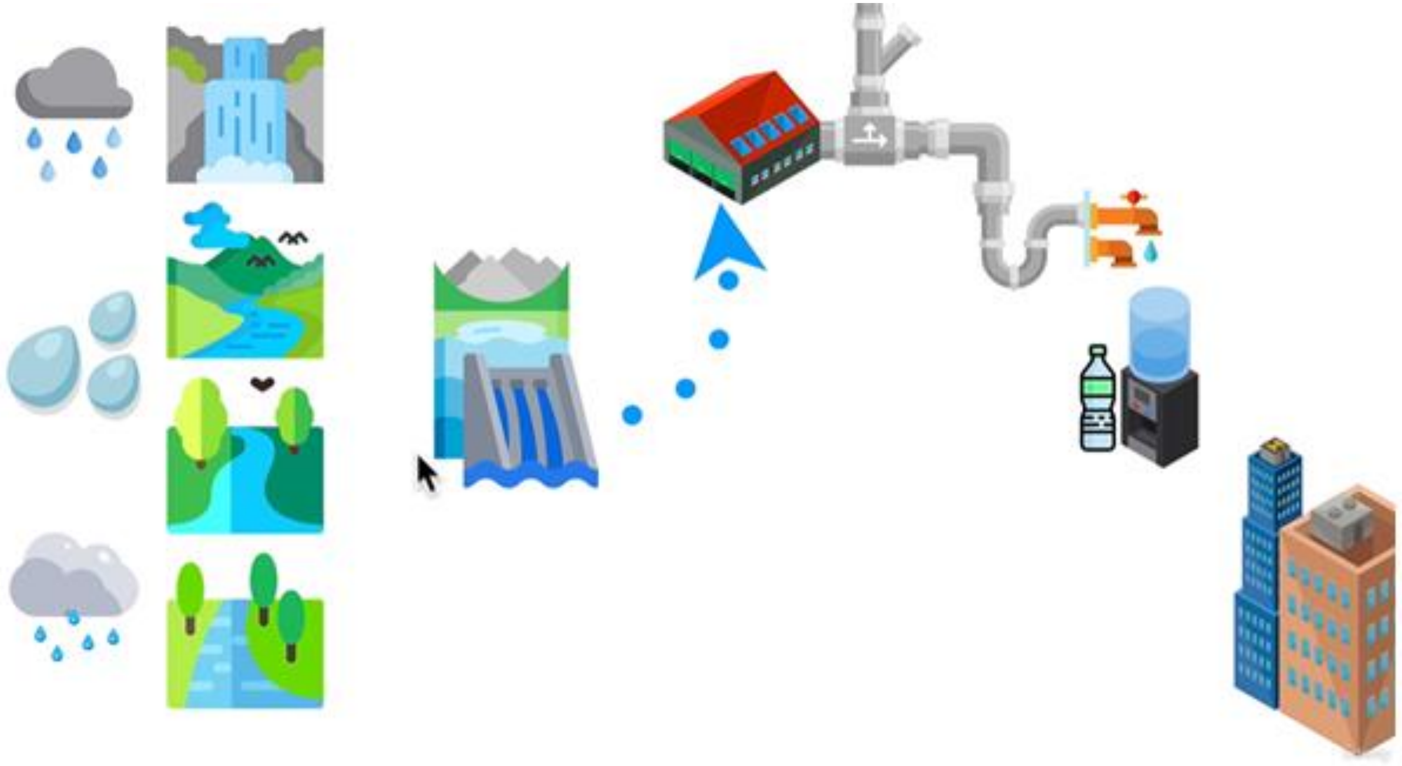
TASK 3

MACHINE/DEEP LEARNING LIFE CYCLE



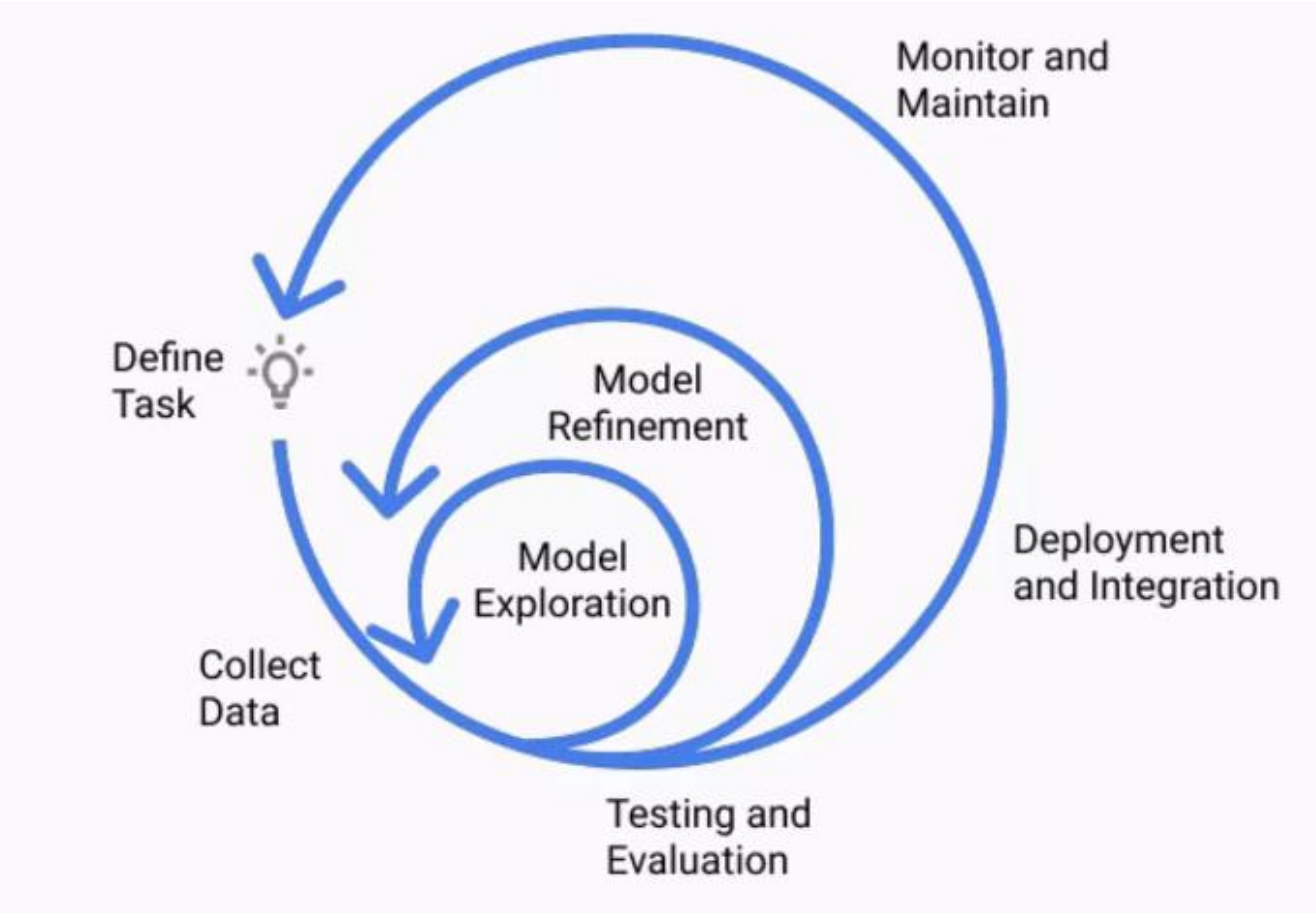
DATA STORAGE

- There are data lakes – where all the raw data get in.
- Then data warehouse – where the data is saved after processing and standardization.

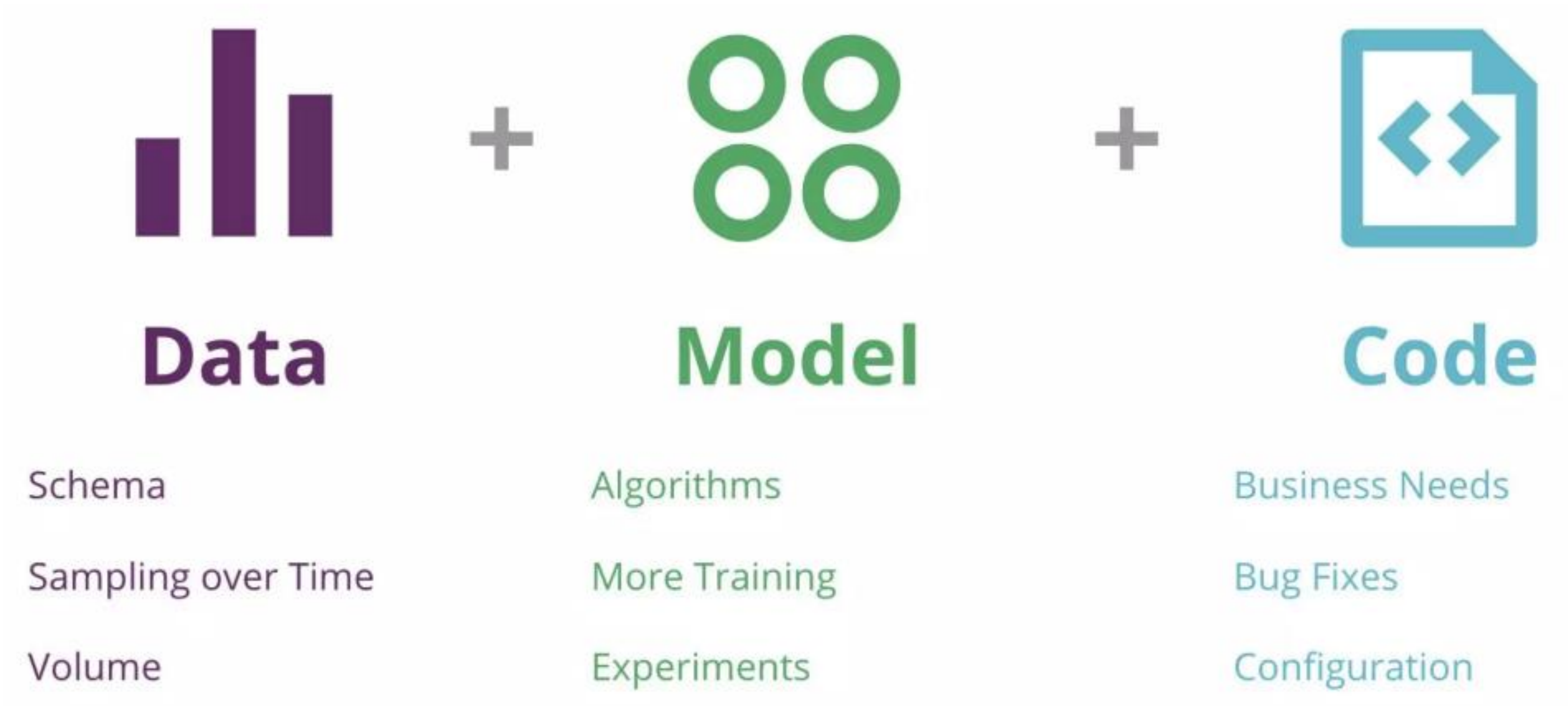


THE ENDLESS LIFE CYCLE

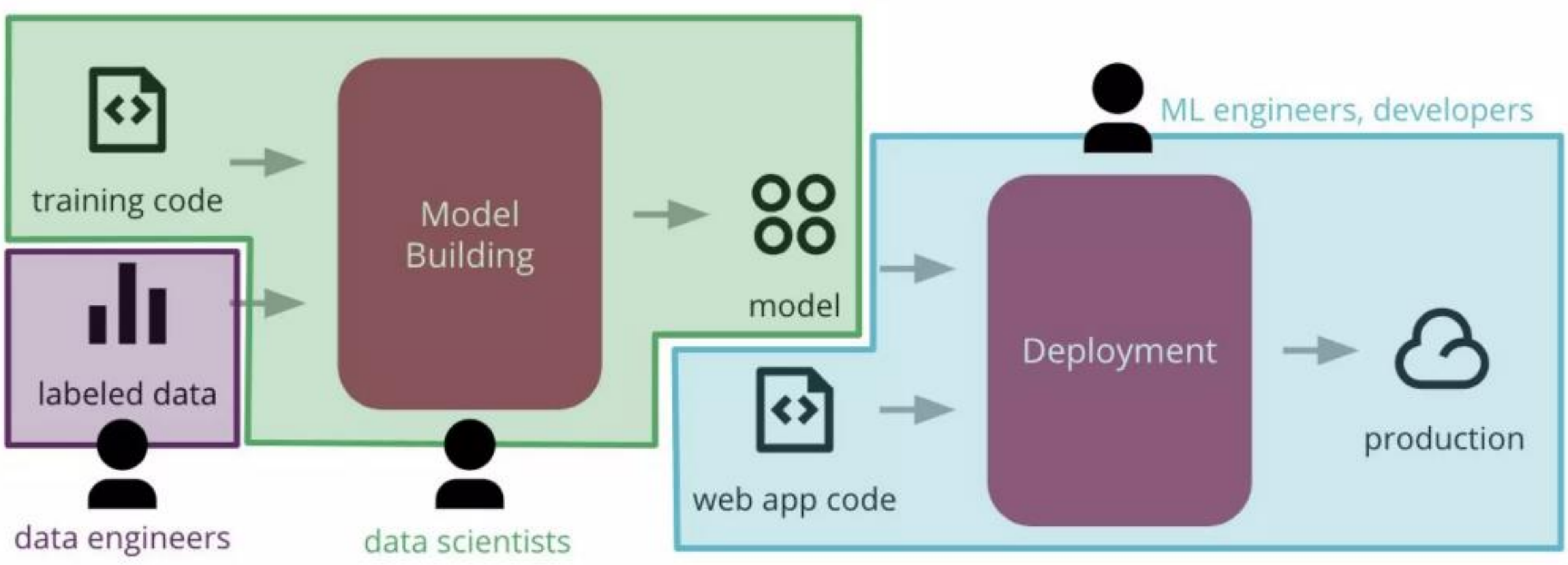
- Define Task.
- Collect Data – Always bring new data.
- Model Exploration - Keep searching for models.
- Model Refinement – Keep improving the model.
- Testing & Eval – Keep testing the model.
- Deploy & Integration – Deployment of versions.
- Monitoring % Maintain – Check production behaviours.
- And all over again.



HOW DOES IT SPLIT



ROLES


training code
Model Building
model
labeled data
data engineers
data scientists
ML engineers, developers
Deployment
production
web app code

THANK YOU