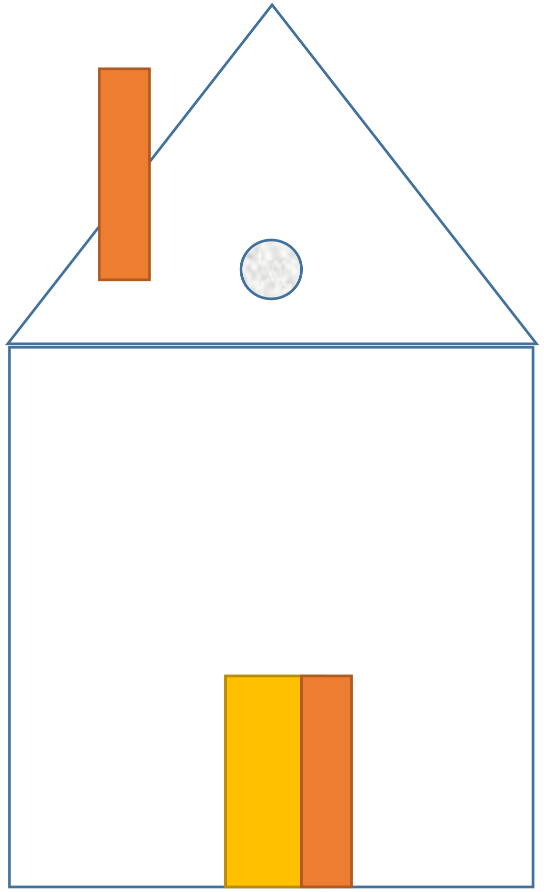
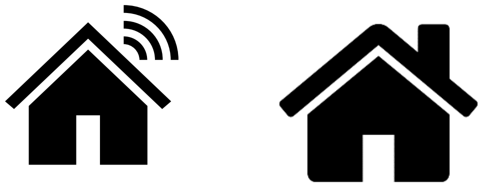
Simple Game

Draw a house on a paper

90% of people have drawn a house like





Question:

How many of your houses are like this?

10% people raised their hands

Another Game

Rapid Fire Round: Quiz

What does it represent?

$$A = lb$$

Answer (100%): Rectangle

What does it represent?

$$W = mg$$

Answer (90%):

Weight = mass times gravity

What does it represent?

$$F = ma$$

Answer (90%):

Newton’s Second Law

What does it represent?

$$y = mx$$

Answer (100%):

Equation of Straight Line

What does it represent?

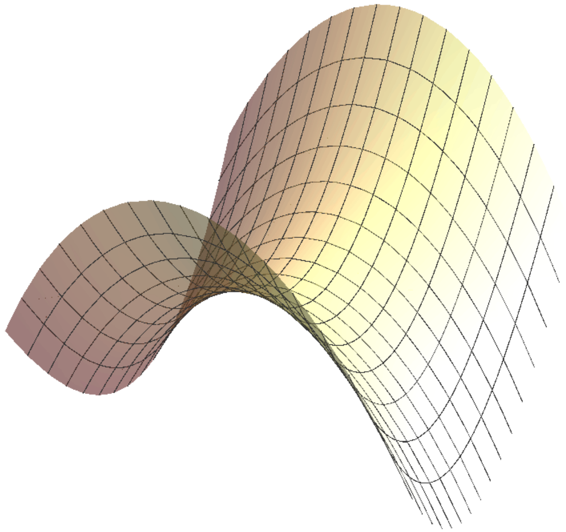
$$z = xy$$

Answer (100%):

Complete Silence

What does it represent?

$$z = xy$$



hyperbolic paraboloid

What does it represent?

$$A = \pi r^2$$

Answer (100%):

Area of a Circle

What does it represent?

$$E = mc^2$$

Answer (100%):

Einstein Equation

What does it represent?

$$y = ax^2$$

Answer (80%):

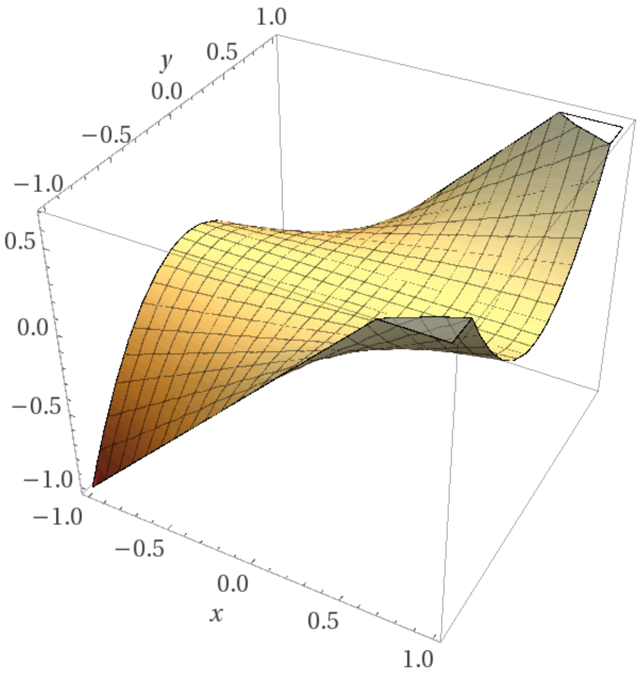
Parabola

What does it represent?

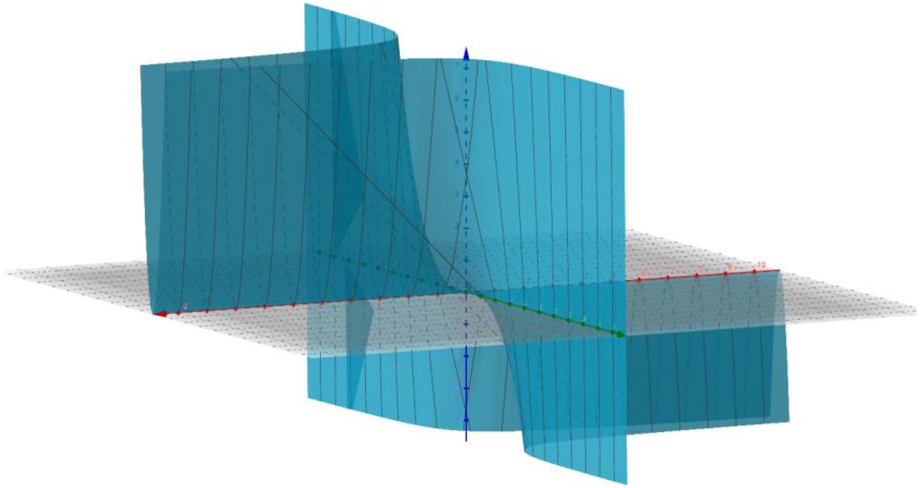
$$z = xy^2$$

Answer (100%):

Complete Silence

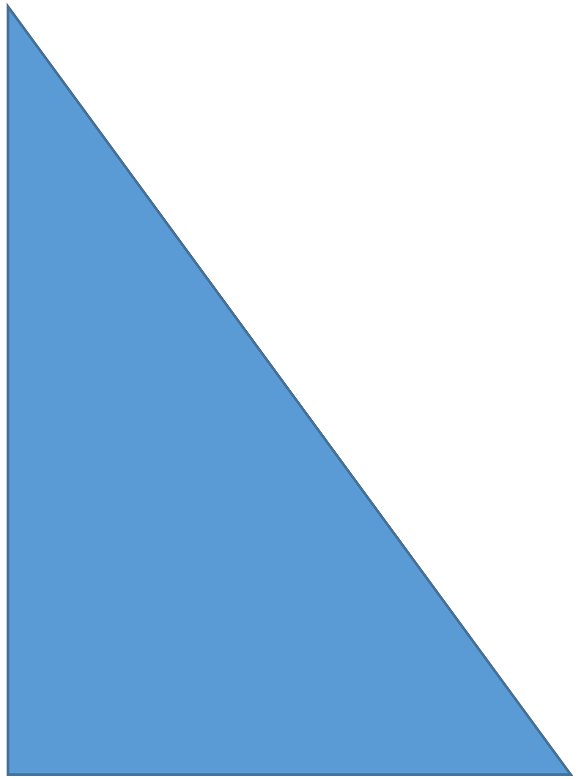


$$z = xy^2$$



What does it represent?

$$a^2 + b^2 = c^2$$



Answer (100%):

Pythagoras Theorem

What does it represent?

$$x^2 + y^2 = r^2$$

Answer (70%):

Equation of Circle

What does it represent?

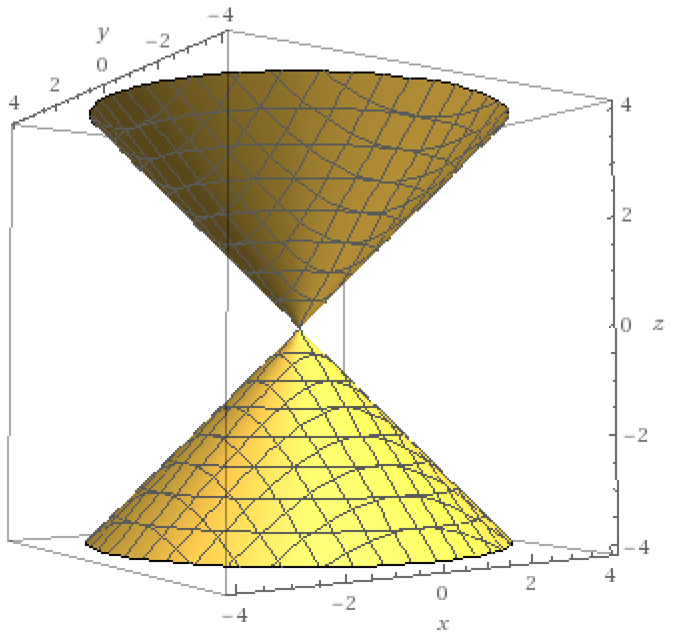
$$x^2 + y^2 = z^2$$

Answer (90%):

Some 3D equation

What does it represent?

$$x^2 + y^2 = z^2$$



What does it represent?

$$\left(x^2 + \frac{9}{4}y^2 + z^2 - 1\right)^3 - x^2z^3 - \frac{9}{80}y^2z^3 = 0$$

Answer (90%):

Pin drop silence

$$\left(x^2+\frac{9}{4}y^2+z^2-1\right)^3-x^2z^3-\frac{9}{80}y^2z^3=0$$

$$y = mx$$

$$A = lb$$

$$F = ma$$

$$W = mg$$

$$z = xy$$

$$y = ax^2$$

$$A = \pi r^2$$

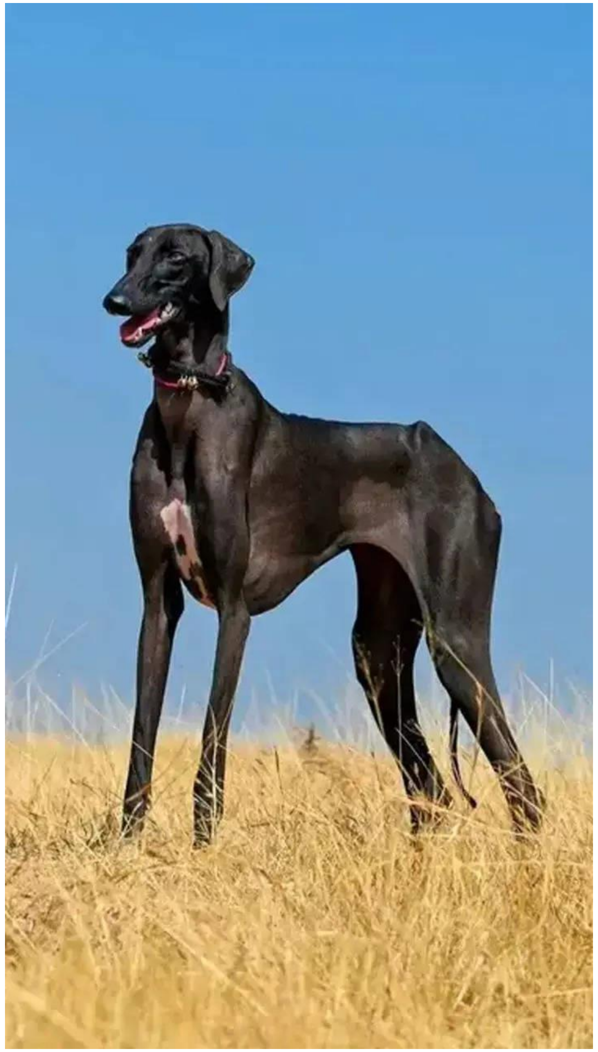
$$E = mc^2$$

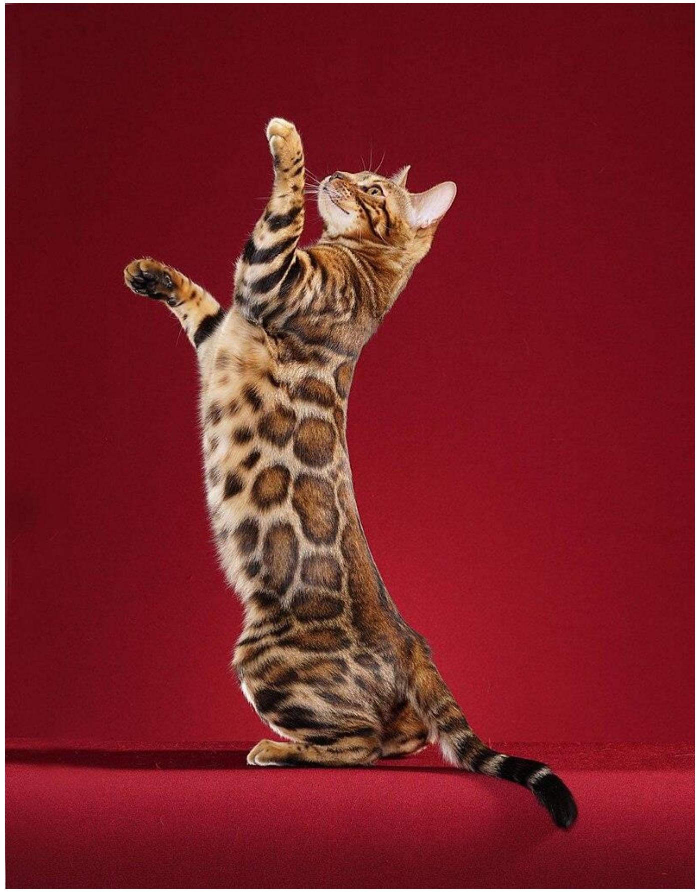
$$z = xy^2$$

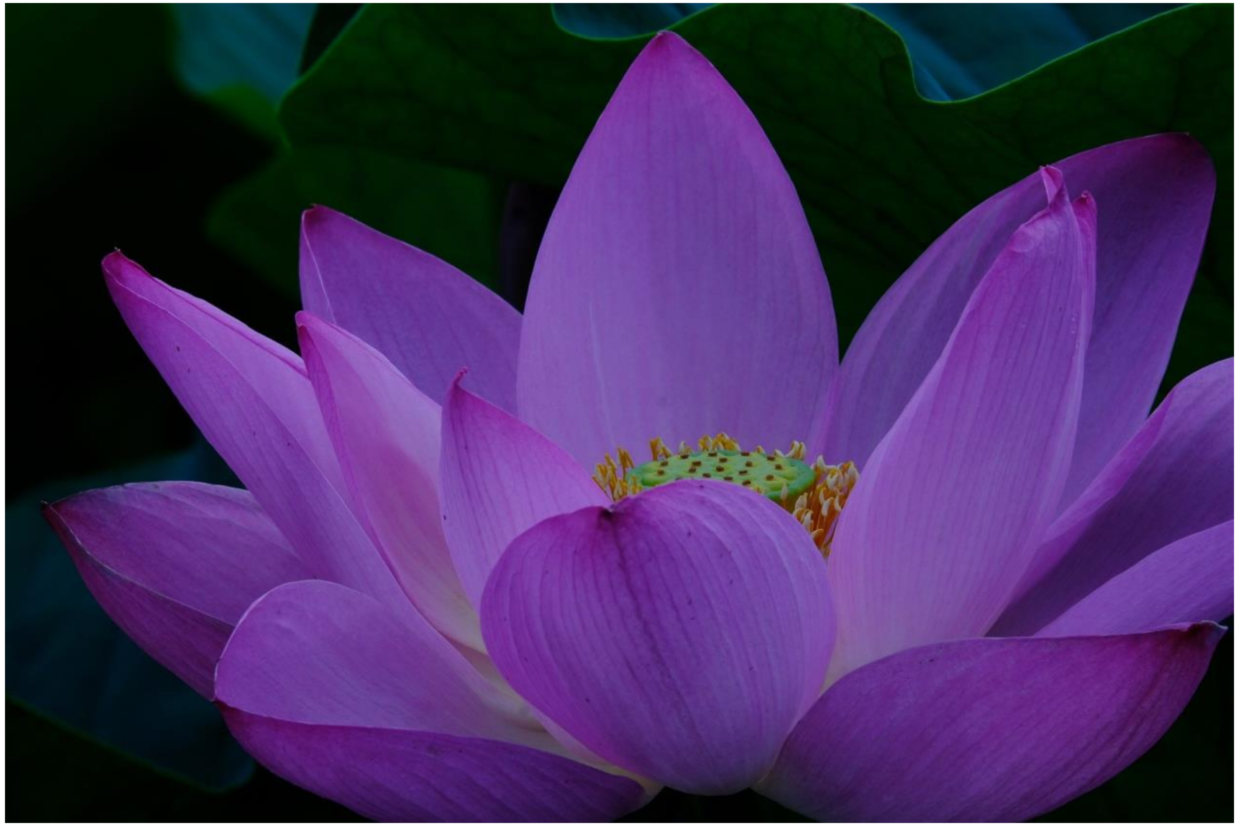
$$x^2 + y^2 = z^2$$

$$x^2 + y^2 = r^2$$

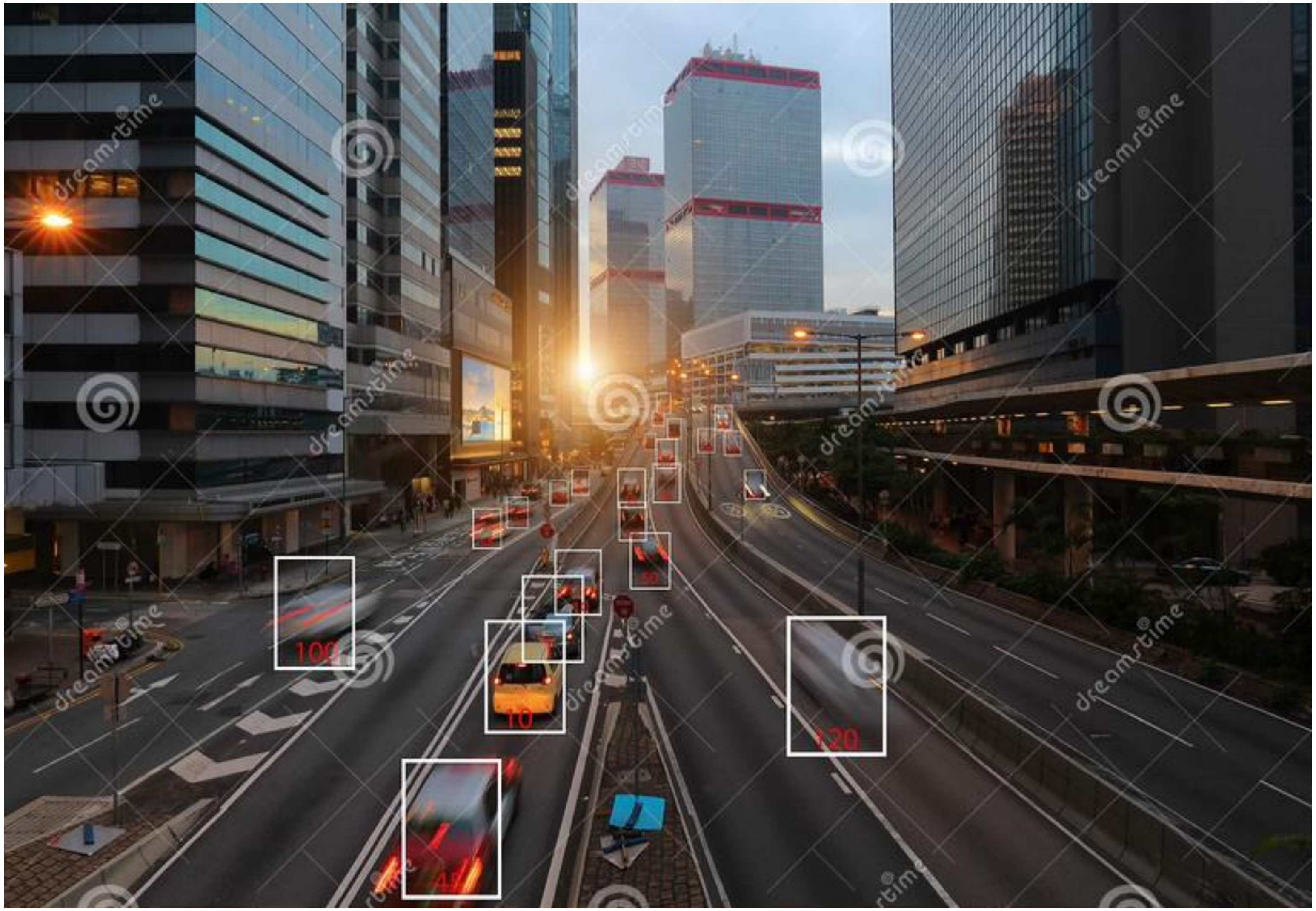
$$a^2 + b^2 = c^2$$







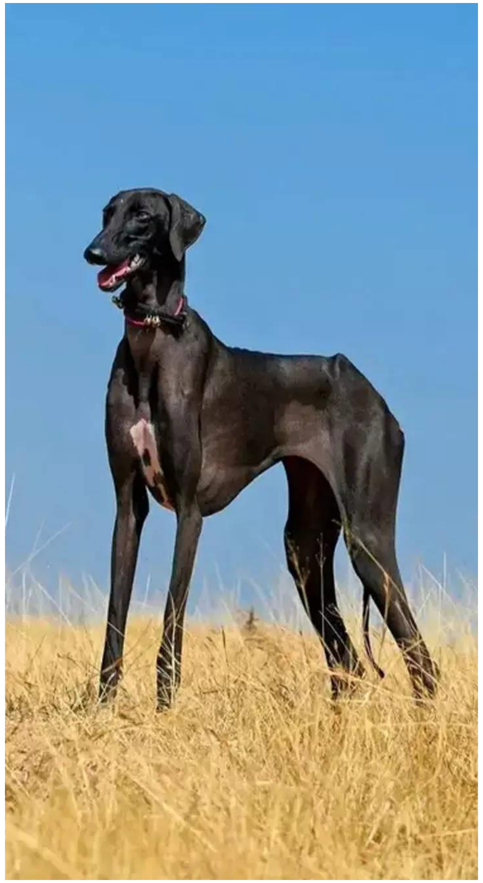

THE ART BOOK
THE GREAT MASTERS
WILLIAM SAFIRE
AMY TAN


100
10
120

German Shepherd

Labrador Retriever

360 Globally Recognized Breeds



Rajapalayam Dog

French Bulldogs

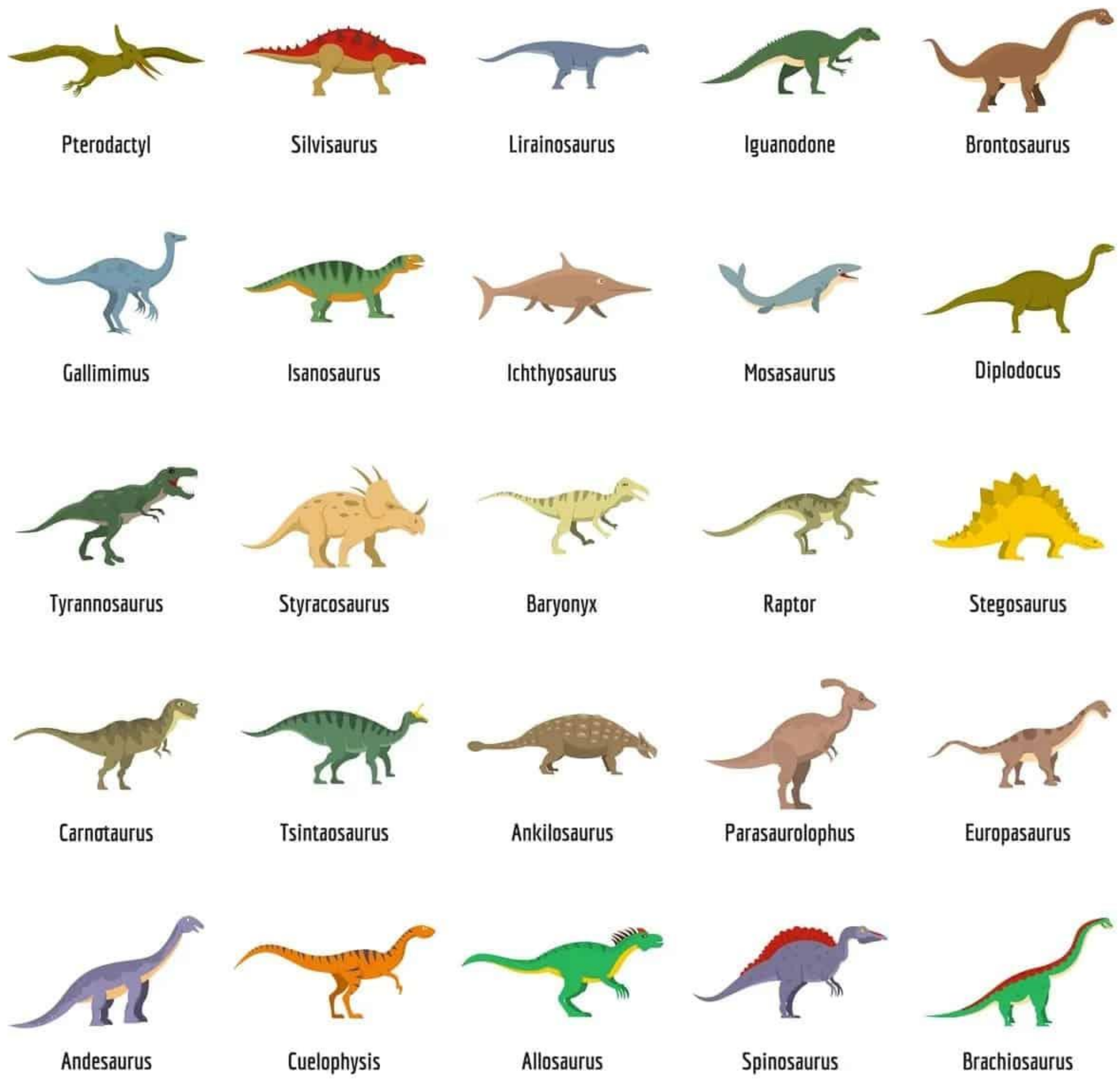
The American Eskimo Dog

rhebamny.com

73 Standardized Breeds

Approximately 400,000 Flowering Plants

https://nayturr.com/types-of-dinosaurs/



Congratulations! You have done the labelling job well.

I mean you are fit to learn Machine Learning concepts

Let us explore more details with mathematics https://www.youtube.com/watch?v=n_1apYo6-Ow

Basics of Machine Learning

Panchatcharam Mariappan

Associate Professor

Department of Mathematics and Statistics, IIT Tirupati

References

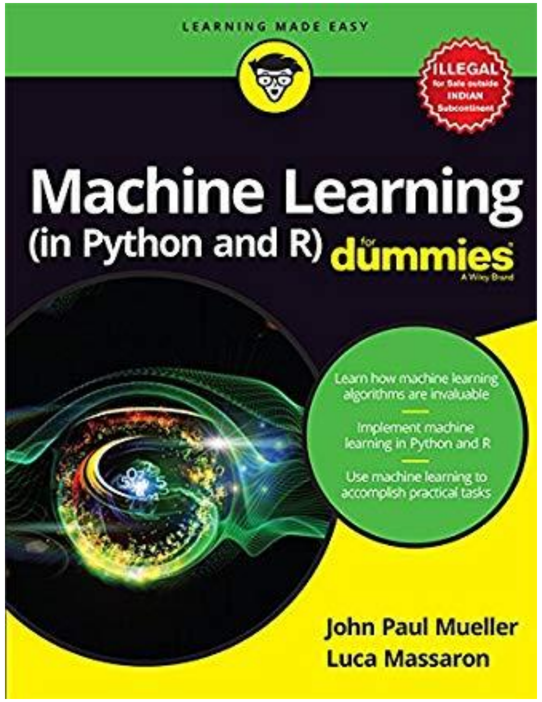
Website

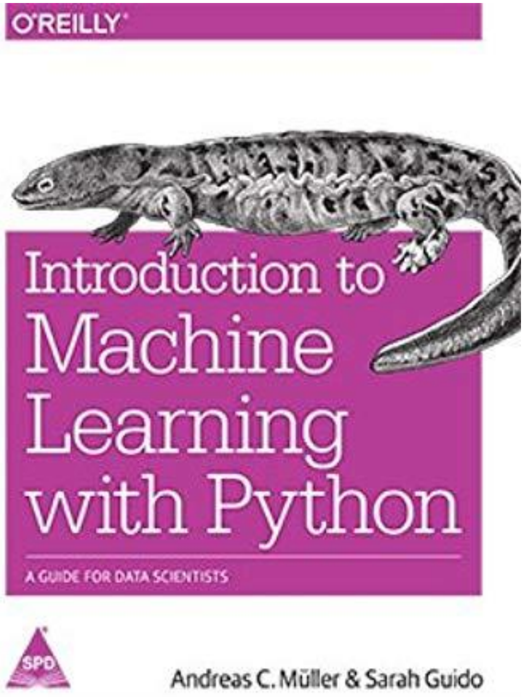
- I. Goodfellow, Y. Bengio, A. Courville, https://www.deeplearningbook.org/
- T. Renelle, Machine Learning Guide Podcast, http://ocdevel.com/mlg
- M. Nielsen, Neural Networks and Deep Learning, http://neuralnetworksanddeeplearning.com/
- J. Brownlee, Examples of Linear Algebra and Machine Learning, https://machinelearningmastery.com/examples-of-linear-algebra-in-machine-learning/
- M. P. Deisenroth, A. A. Faisal. C. S. Ong, Mathematics for Machine Learning, https://mml-book.github.io/book/mml-book_printed.pdf

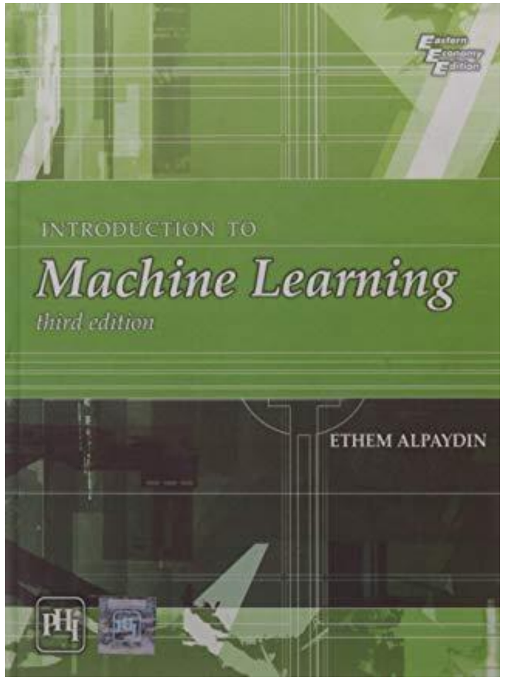
References

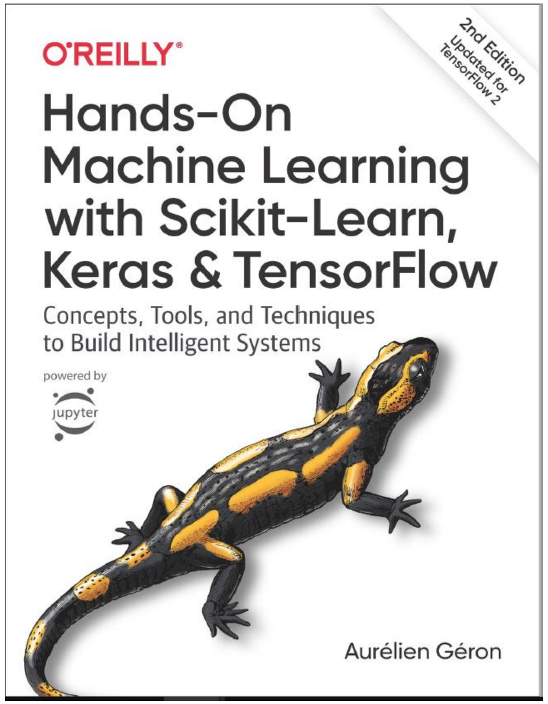
Books

- J. P. Mueller, Machine Learning in Python and R for Dummies, 2016
- A. Muller, Introduction to Machine Learning with Python: A Guide for Data scientists, 2016
- A. Ethem, Introduction to Machine Learning, 2015
- A. Geron, Hands-On Machine Learning with Scikit-Learn Keras & TensorFlow, Oreilly, 2019



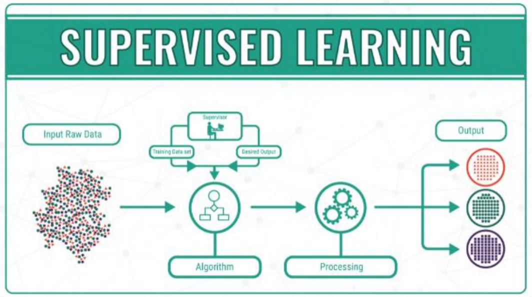


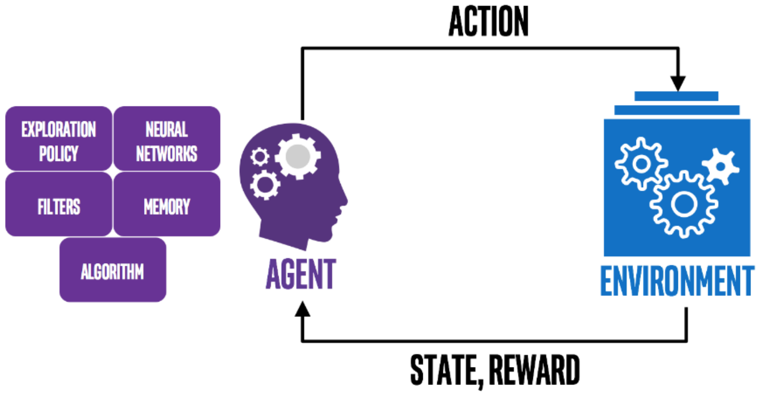


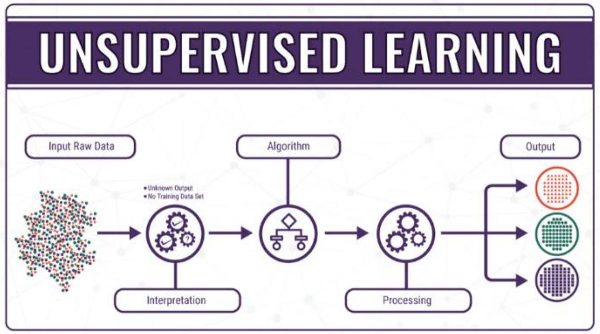


Journey

- ❑Early Concepts 1950-1960s: Perceptions, Basic Building blocks of Neural Networks
- ❑Rule Based Systems (1970s-1980s)
- ❑First AI Winter (1980s-1990s)
- ❑Statistical Learning (1990s): Decision tree, SVM, Bayesian
- ❑Rise of Neural Networks (1990s-2000s). Neural Network, DL, Limited to computational power
- ❑Big Data and Computational Power (2010s): CNN, RNNs
- ❑Deep Learning Dominance (2010-2020): NLP
- ❑Transfer Learning, Pre-trained models(2020s)
- ❑Generative Adversarial Networks, Quantum Computing (ongoing)

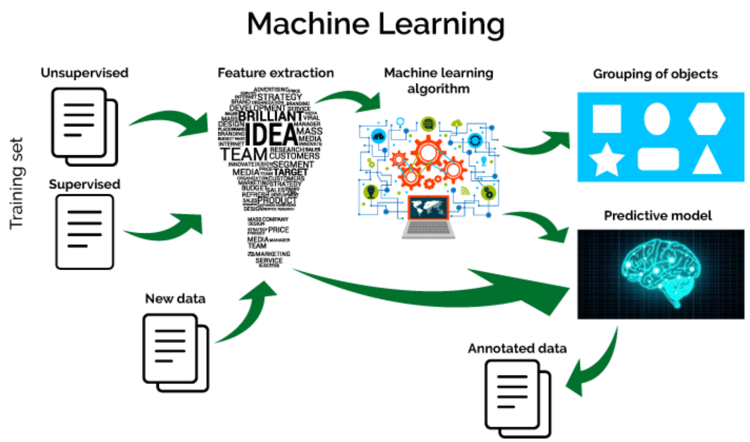






Source: Educative.io

Machine Learning



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

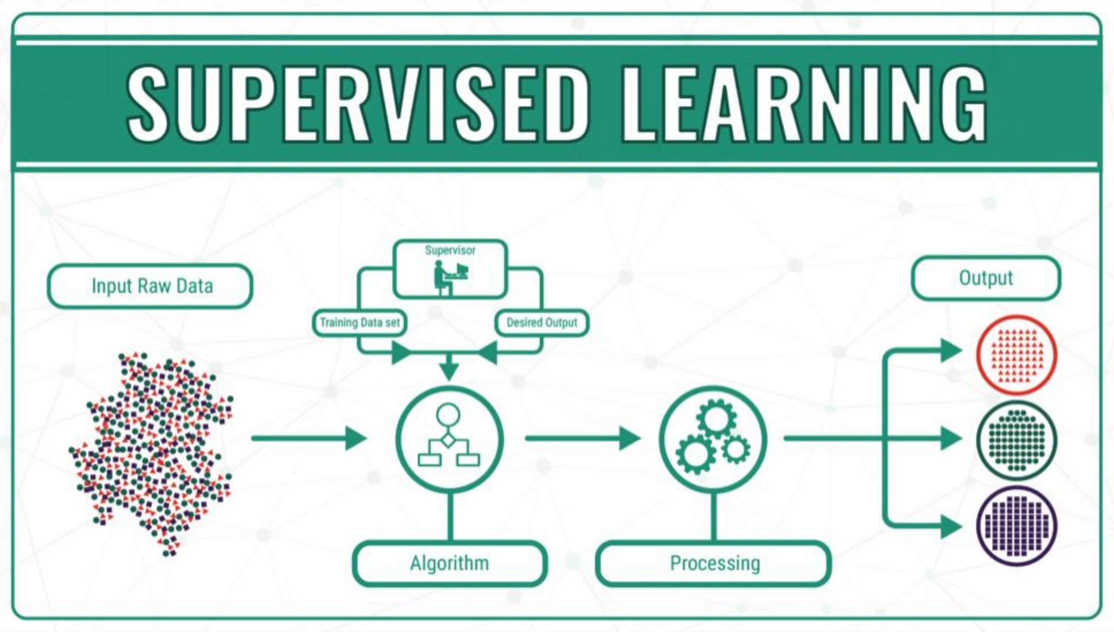
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
—Tom Mitchell, 1997

"Algorithms that parse data, learn from that data, and then apply what they've learned to make informed decisions"
https://www.zendesk.com/

Source: Educative.io

Supervised Learning

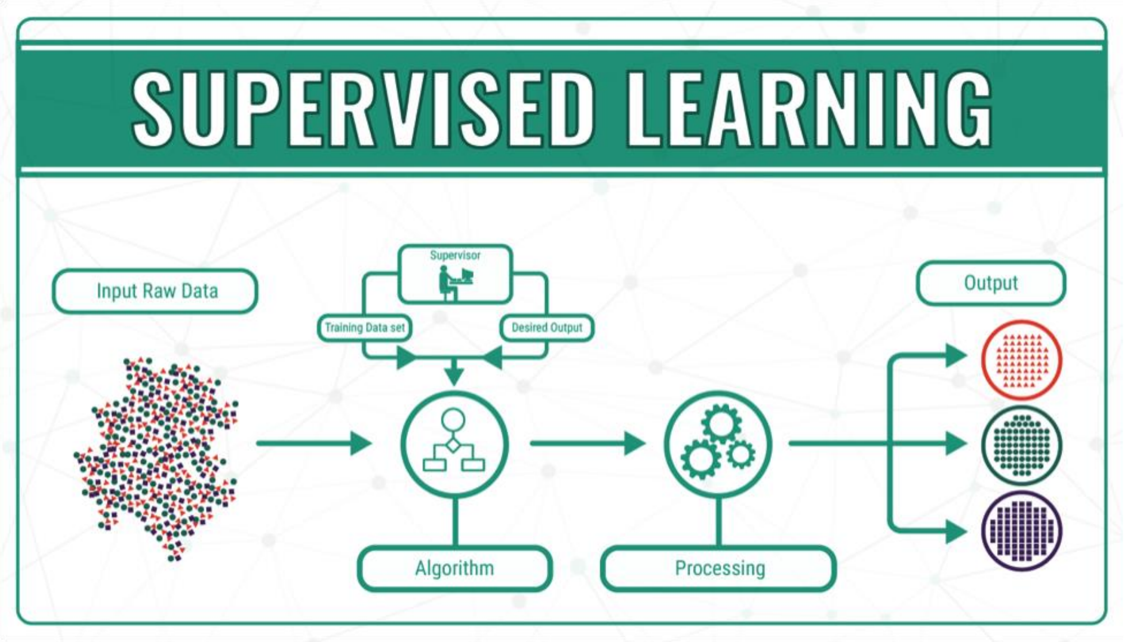
- Builds a Mathematical Model
- Contains input and output data: Training Data
- Relations : Supervisor Signals, F(x)
- Each training example: Array or Vector
- Training Data: Matrix
- Iterative optimization



Source: Educative.io

Supervised Learning

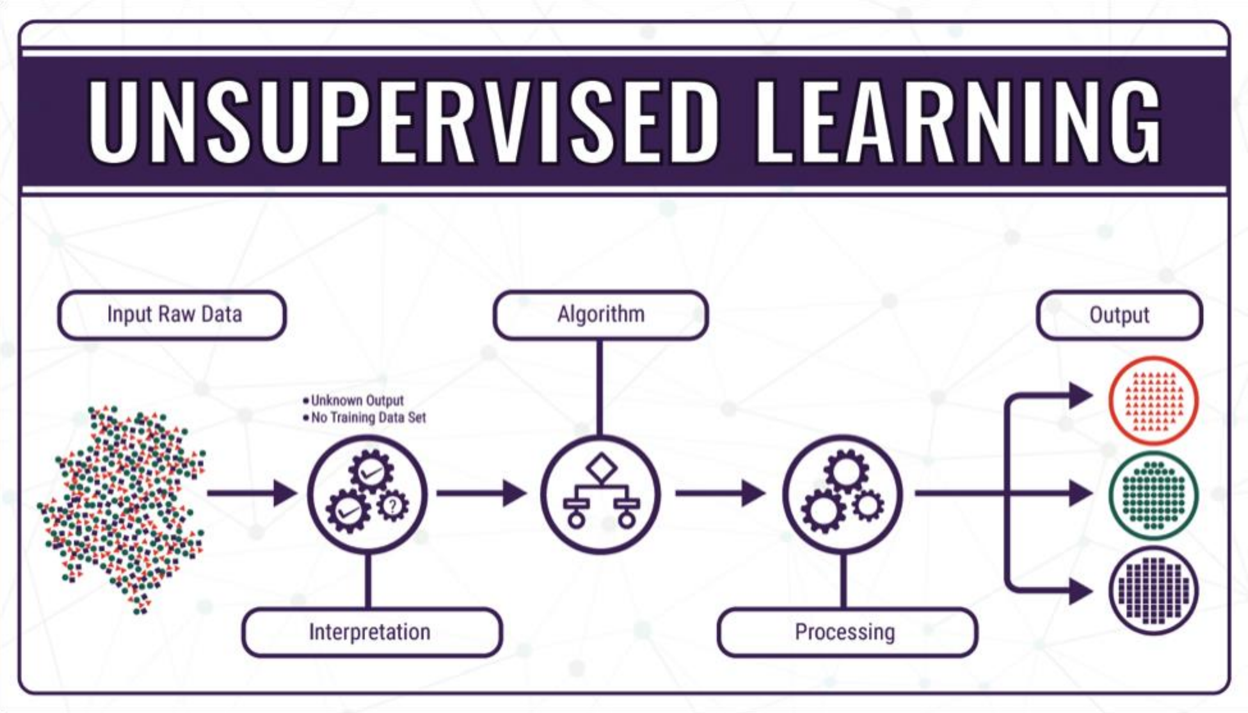
- k-nearest neighbours
- Linear Regression
- Logistic Regression
- SVM
- Decision Trees
- Random Forests
- Neural Networks



Source: Educative.io

Unsupervised Learning

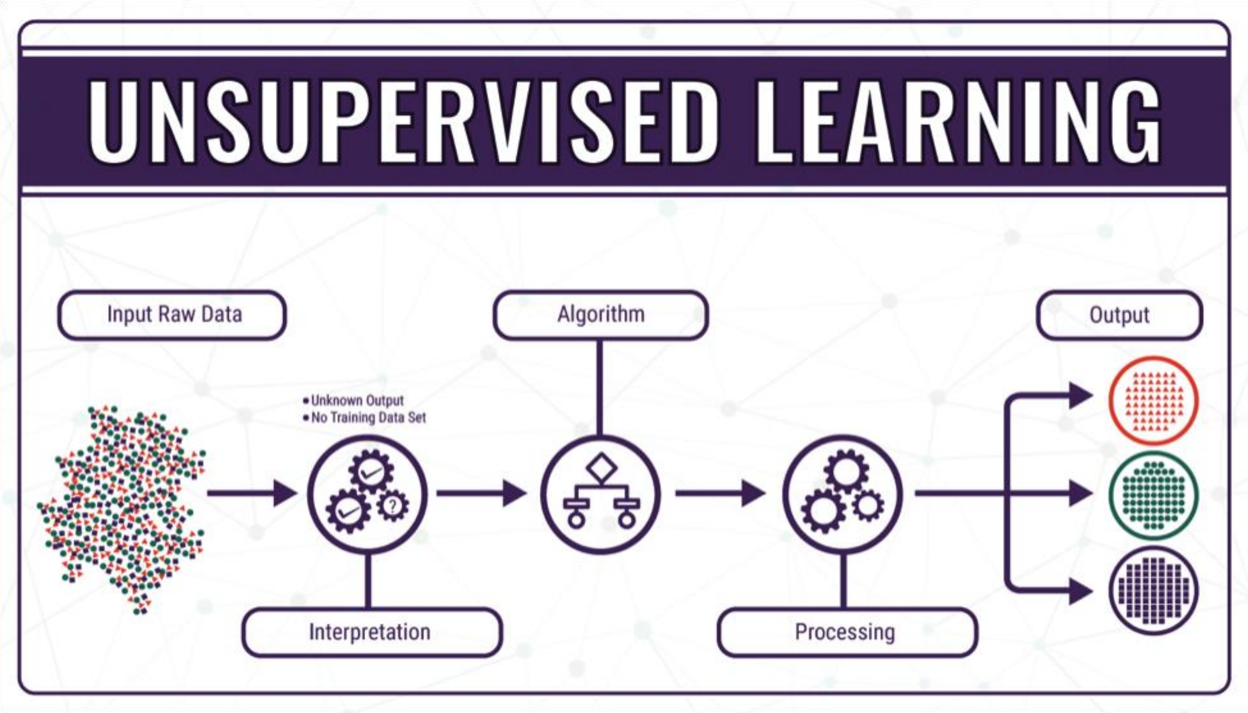
- Takes only input
- Finds the structure in the data
- Groups/Clustering data
- Classifies
- React based on the presence of such commonalities in each new piece of data
- Statistical analysis (density estimation function)
- Weighted to finding probabilities of outcomes (conditional probability)



Source: tecnative.io

Unsupervised Learning

- Clustering
    - k-means
    - DBSCAN
    - HCA
- Anomaly/Novelty detection
    - One-Class SVM
    - Isolation Forest
- Visualization
    - PCA
    - Kernel PCA
    - LLE
    - t-SNE



Source: tecnative.io

What is “learning” in ML?

Hard question to answer. Let us give a fuzzy answer at a enough high level of abstraction

1. **Algorithms that solve some kind of inference problems**
2. **Models for datasets**

Does the image have only books?

Statistical Inference

Statistical inference is the process of using data analysis to deduce properties of an underlying probability distribution.

Inferential statistical analysis infers properties of a population, for example by testing hypotheses and deriving estimates.

Does the image have only books?

What does a ML algorithm do?

Machine learning algorithms are not algorithms for performing inference. Rather, they are algorithms for building inference algorithms from examples. An inference algorithm takes a piece of data and outputs a decision (or a probability distribution over the decision space).

Second type of problem associated to ML

"Given a dataset how I can succinctly describe it (in a quantitative, mathematical manner"

Example: Regression Analysis

Geometric Models:

The general problem is that we have example data points

$$x_1, x_2, \cdots, x_n \in \mathbb{R}^D$$

We want to find some kind of geometric structure that (approximately) describes them.

Probabilistic Models:

The basic task here is to find a probability distribution that describes the dataset $\{x_n\}$

ML Examples

- **Classification: From data to discrete classes**
  - Spam Filtering
  - Object Detection
  - Weather Prediction (rain, snow)
- **Regression: predict a numeric value**
  - Stock Market
  - Weather Prediction (Temperature)
- **Ranking: comparing items**
  - Web Search
  - Find Similar images
- **Collaborative Filtering: Recommendation**

ML Examples

Clustering: discovers structure in data

- Cluster Point or images
- Cluster web search

Embedding: Visualize data

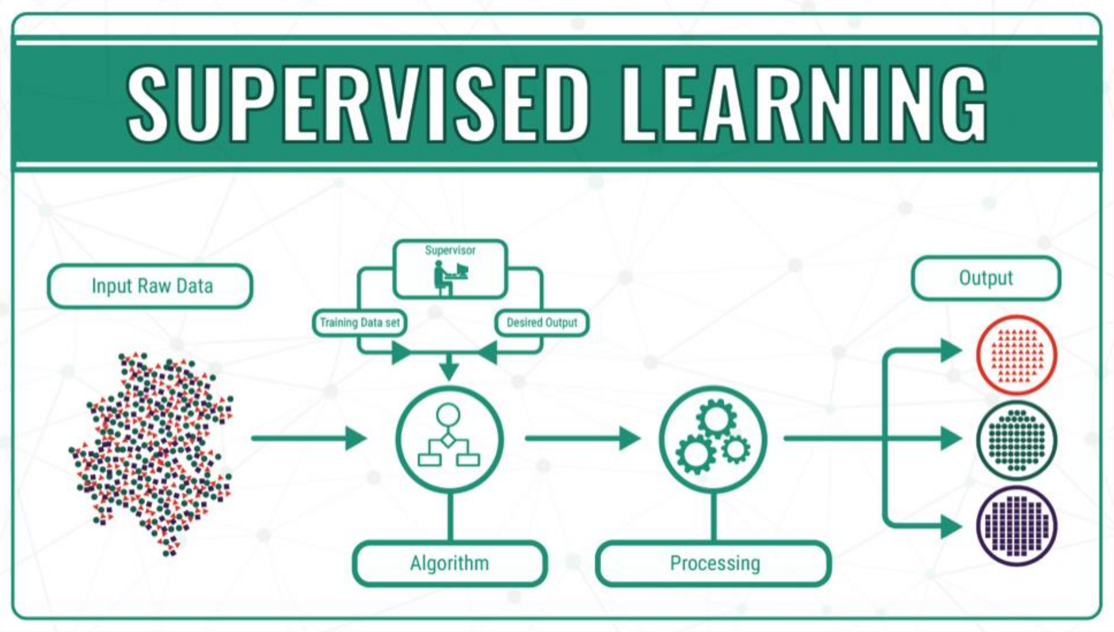
- Images words

Structured Prediction: from data to discrete classes

- Speech Recognition
- NLP

Supervised Learning

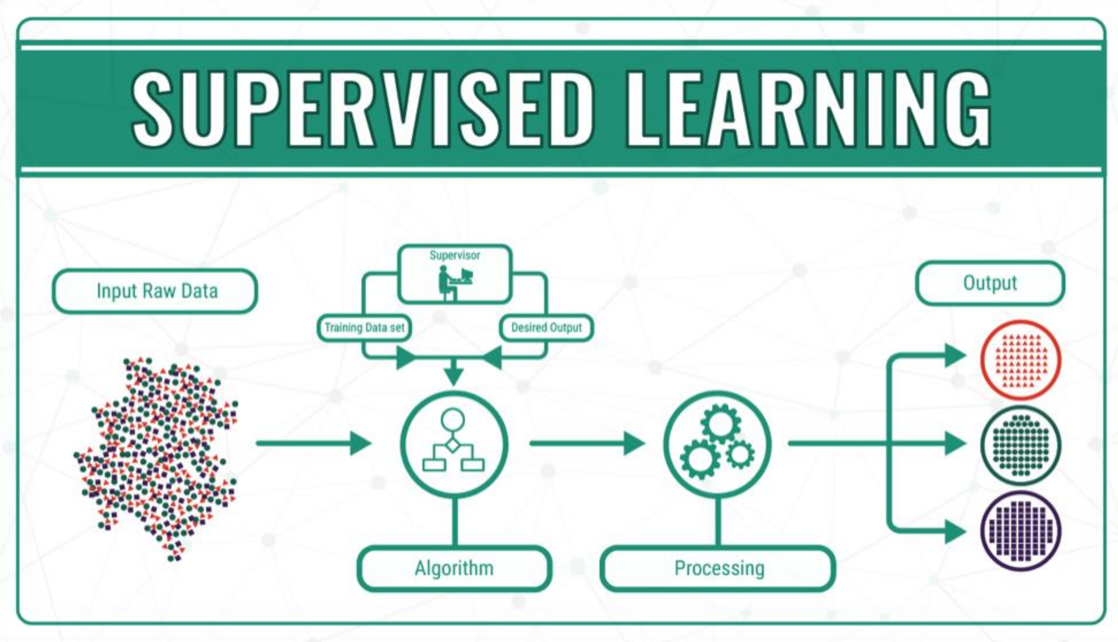
- Builds a Mathematical Model
- Contains input and output data: Training Data
- Relations : Supervisor Signals, F(x)
- Each training example: Array or Vector
- Training Data: Matrix
- Iterative optimization



Source: Educative.io

Supervised Learning

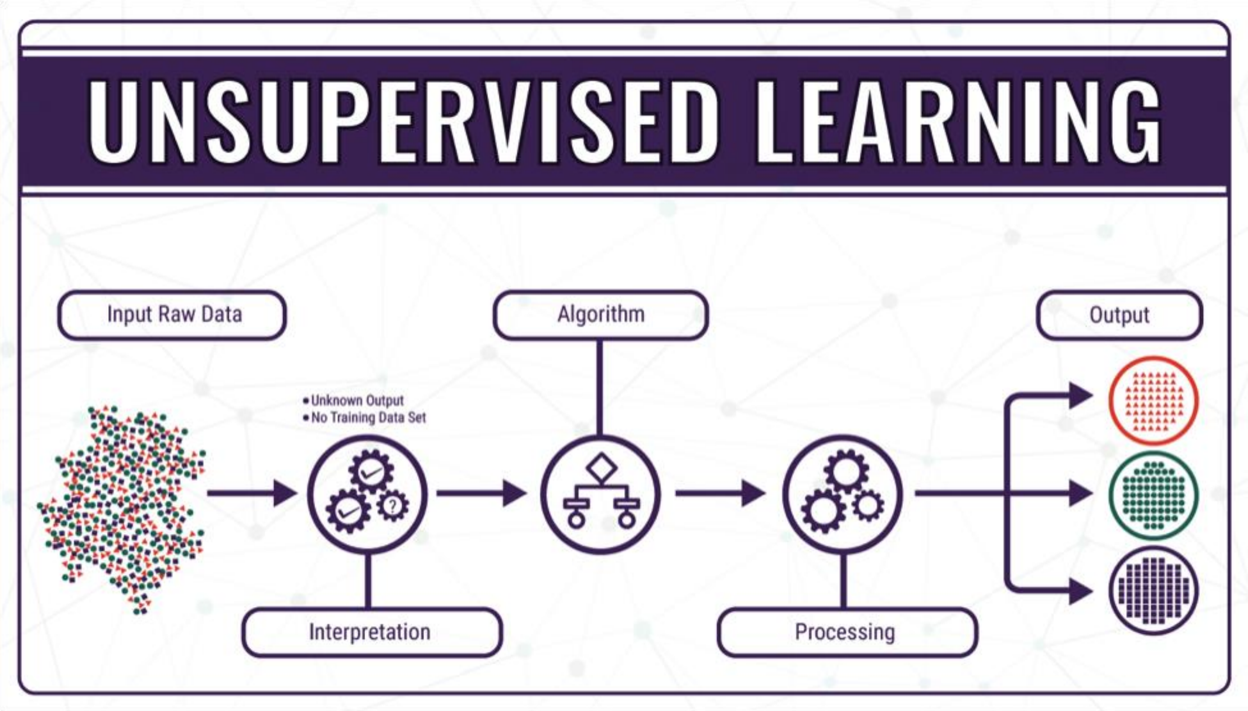
- k-nearest neighbours
- Linear Regression
- Logistic Regression
- SVM
- Decision Trees
- Random Forests
- Neural Networks



Source: https://how.dev/answers/supervised-learning-algorithms

Unsupervised Learning

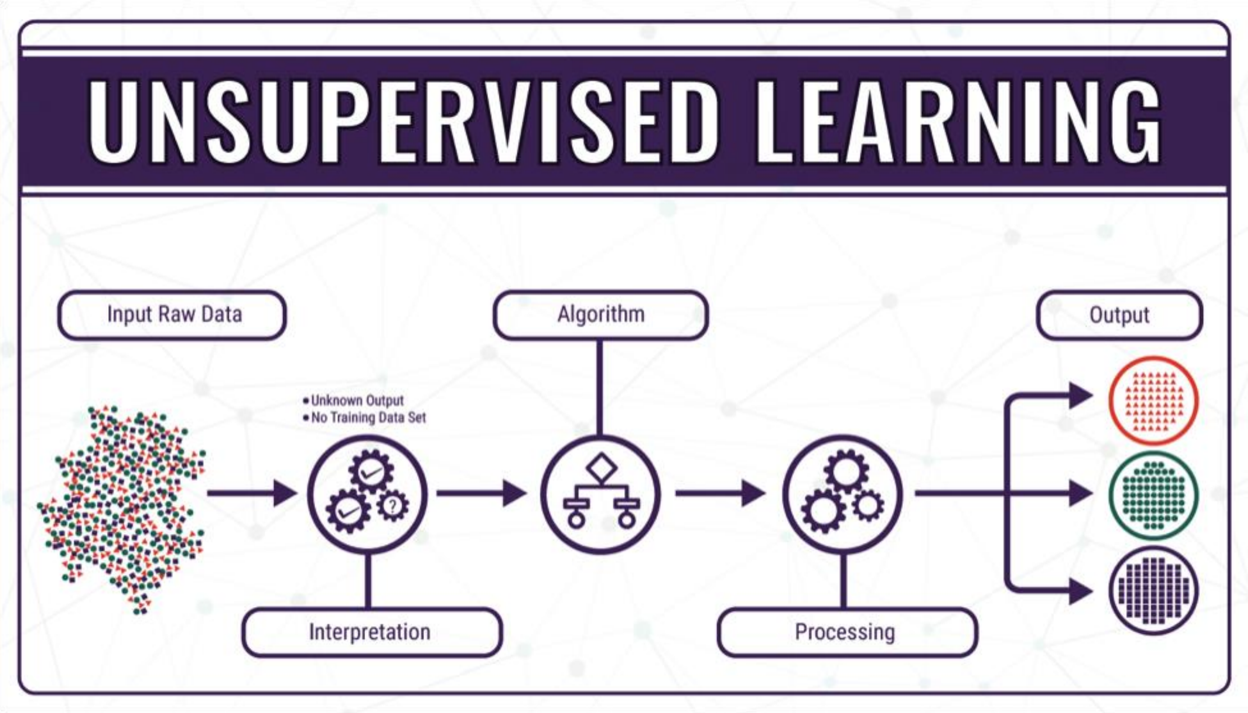
- Takes only input
- Finds the structure in the data
- Groups/Clustering data
- Classifies
- React based on the presence of such commonalities in each new piece of data
- Statistical analysis (density estimation function)
- Weighted to finding probabilities of outcomes (conditional probability)



Source: tecnative.io

Unsupervised Learning

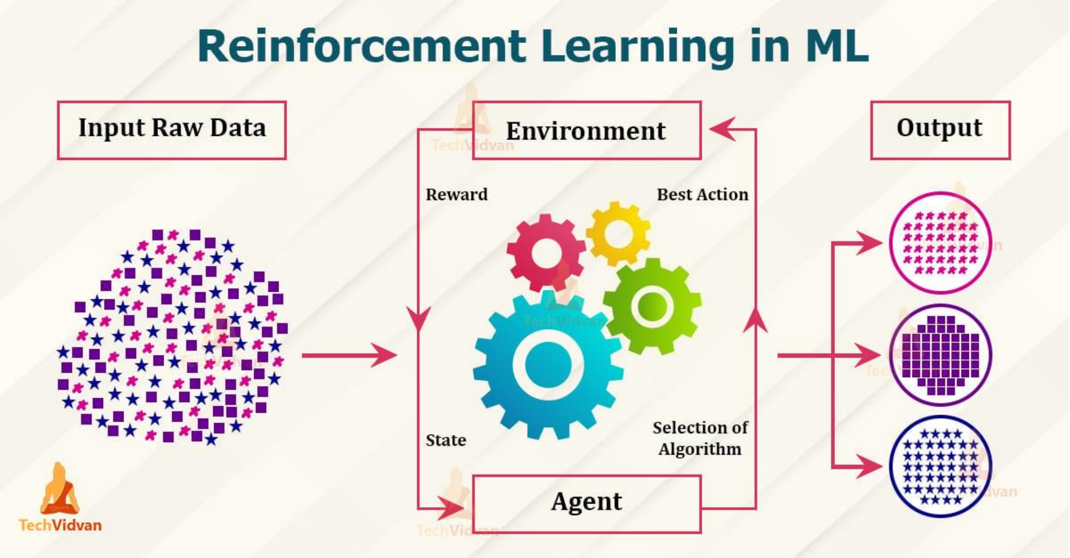
- Clustering
  - k-means
  - DBSCAN
  - HCA
- Anomaly/Novelty detection
  - One-Class SVM
  - Isolation Forest
- Visualization
  - PCA
  - Kernel PCA
  - LLE
  - t-SNE



Source: tecnative.io

Reinforcement Learning

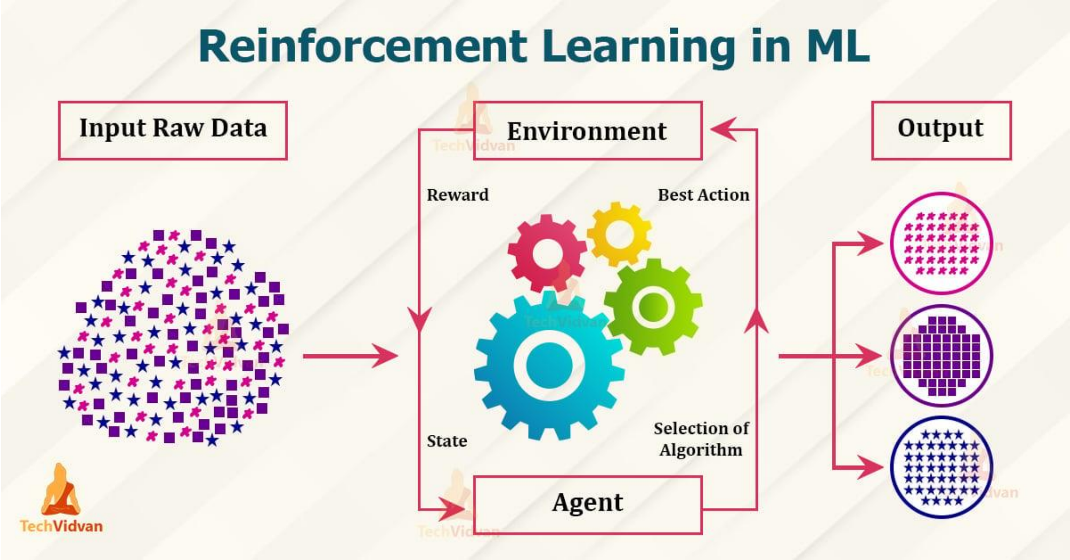
- Give rewards for every positive result and make based on an algorithm
- **Agent-Based Learning:** Learns by interacting with the environment
- **Trial-and-Error**: Receives rewards or penalties for actions
- **Objective**: Maximize cumulative rewards over time
- **Decision Process**: Uses **Markov Decision Process (MDP)** framework
- **Exploration vs. Exploitation**: Balances between trying new actions and using learned knowledge
- **Optimization**: Iterative improvement of policies (e.g., **Q-learning, Policy Gradient** methods)



Source:https://techvidvan.com/

Reinforcement Learning

- Model Free-RL (Value Based)
  - Q-Learning
  - Deep Q-Networks (DQN)
  - SARSA
- Model Free-RL (Policy Based)
  - REINFORCE
  - Policy Gradient
  - Actor-Critic
- Model Based-RL
  - Dynamic Programming
  - Model Predictive Control (MPC)



Source:https://techvidvan.com/

Three Different Types of ML

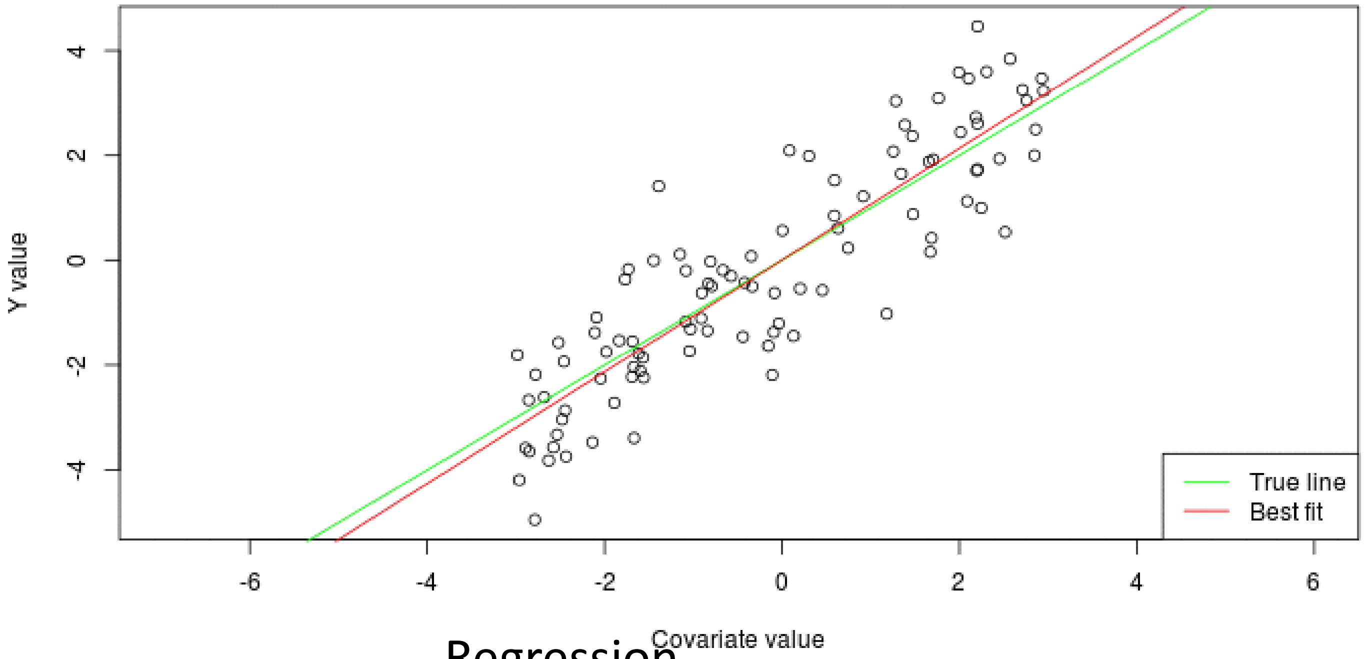
Supervised Learning
- Labeled Data
- Direct Feedback
- Predict Outcome/Future

Unsupervised Learning
- No Labels
- No Feedback
- Find Hidden Structures in Data

Reinforcement Learning
- Decision Process
- Reward System
- Learn from Series of Actions

Machine Learning in Mathematical Way

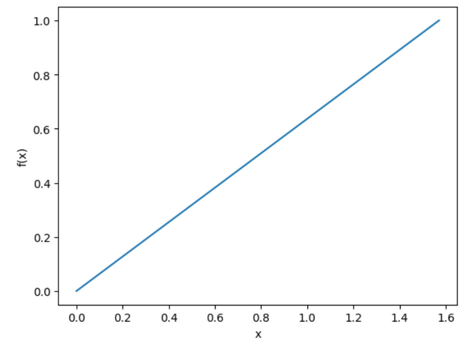
- **Assumption**: Given a data set $\{(x_i, y_i)\}$, $\exists$ a relation $\mathrm{f}: \mathrm{X} \rightarrow \mathrm{Y}$
- **Supervised Learning:**
  - Given: Training Set $\{(x_i, y_i) | i = 1,2,\cdots, N\}$
  - Find: $\hat{f}: \mathrm{X} \rightarrow \mathrm{Y}$ a good approximation to f



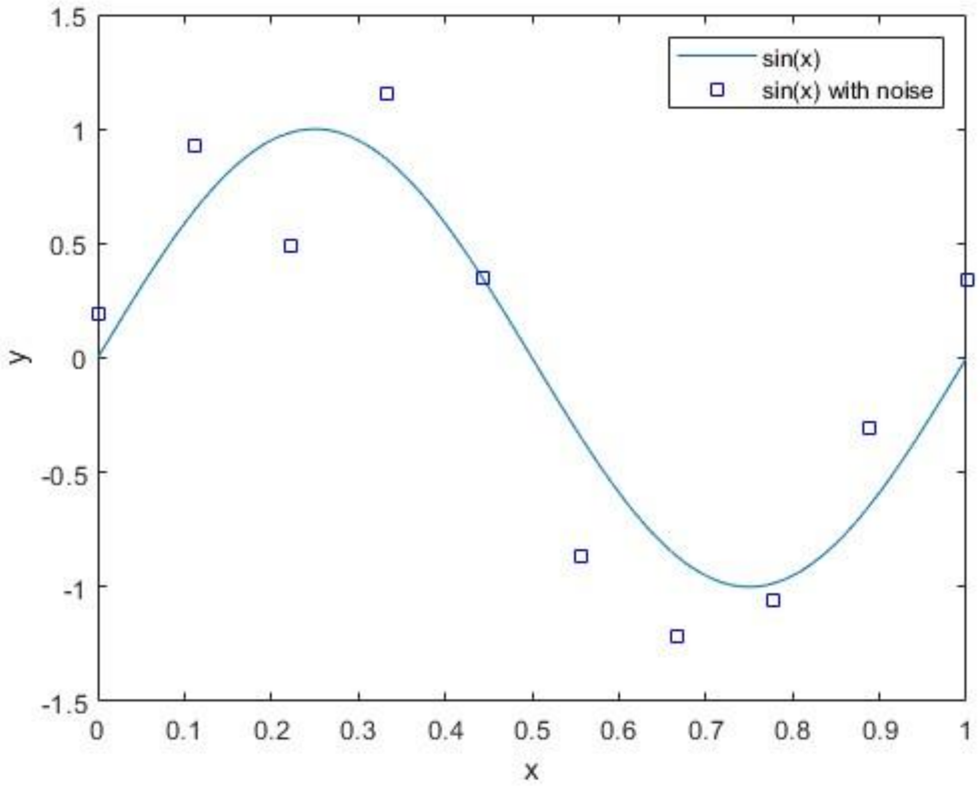
Regression

- Girls vs Boys

x	0	$\frac{\pi}{2}$
$f(x)$	0	1

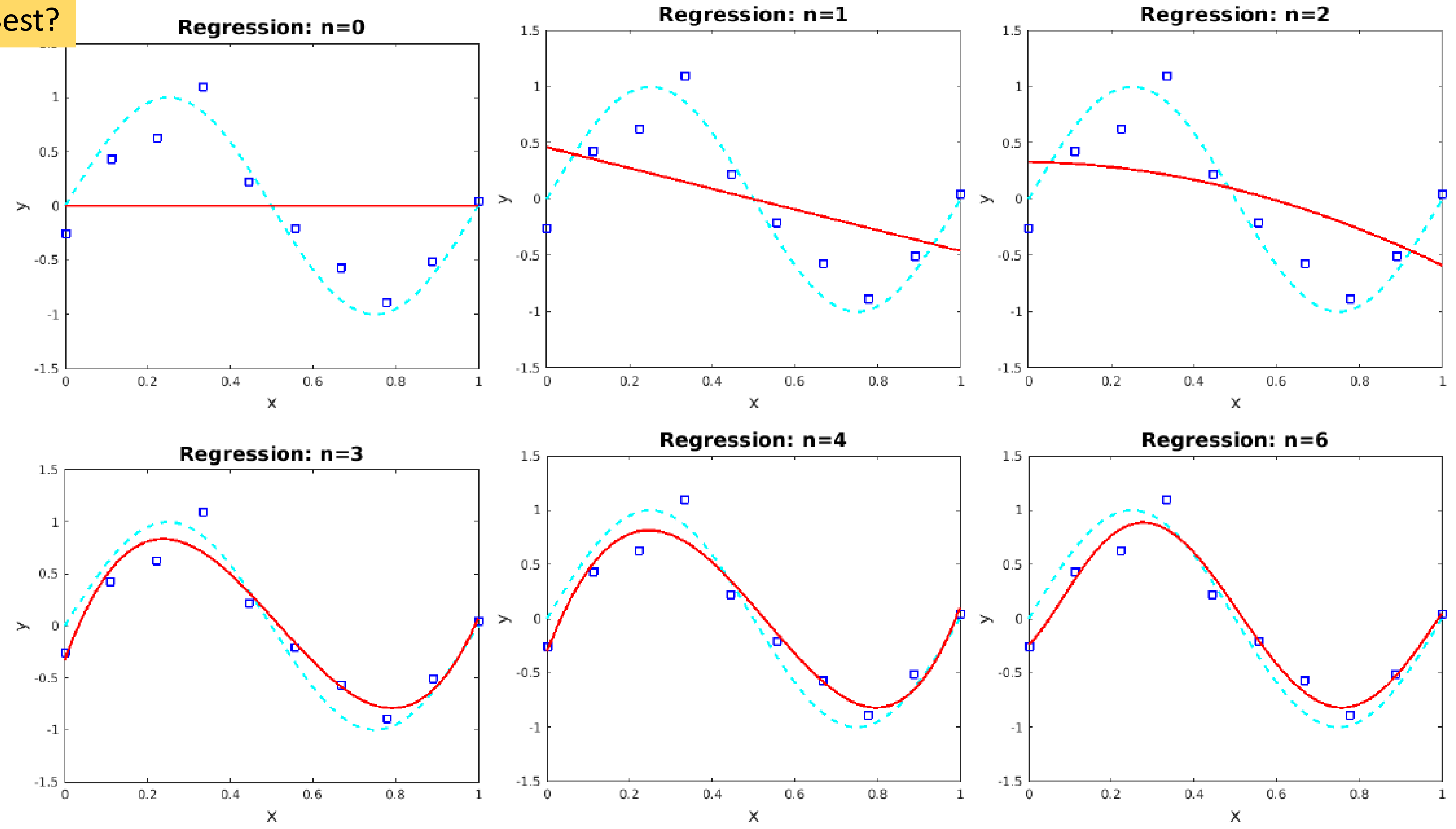


Consider 10 points generated from a sine function with noise



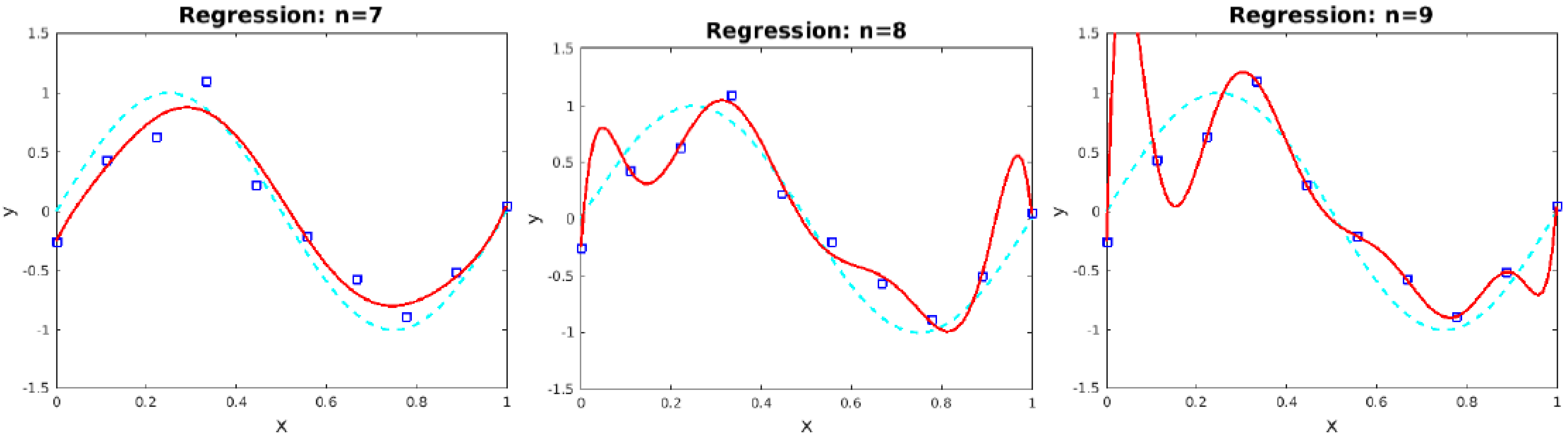
Simple Example

Which is Best?



Simple Example

Which is Best?



MS or LS Error

How do you measure it?

Given several models with similar explanatory ability, the simplest is most likely to be the best choice

Least Square Error

Given a dataset $\{(x_i, y_i) | i = 1,2,\cdots,m\}$ and the model P_n, define the LS Error as

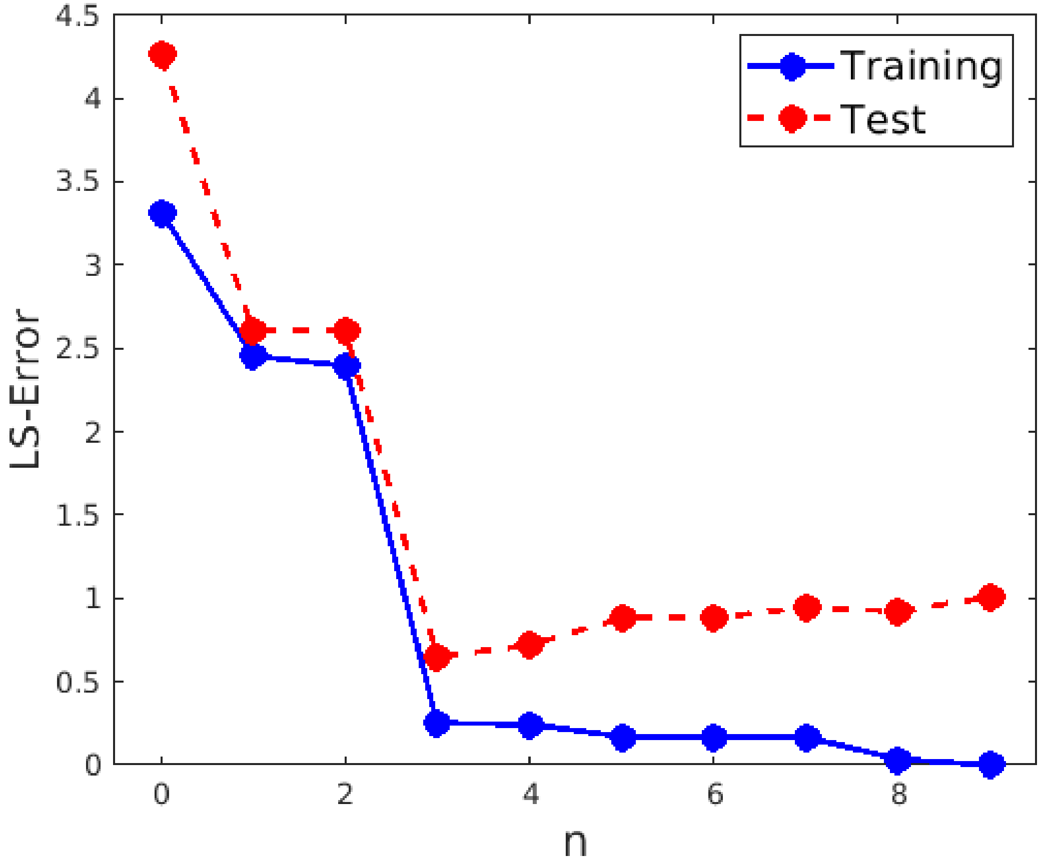
$$E_n = \sum_{i=1}^{m} \left(y_i - P_n(x_i)\right)^2$$

It is also called the mean square error

MS or LS Error

The best choice is P_3

Least Square Error



Occam’s Razor Principle

Law of Parsimony

One should not increase, beyond what is necessary, the number of entities required to explain anything

- When many solutions are available for a given problem, we should select the simplest one
- What do you mean by simple?
  - Use prior knowledge of the problem to solve to define what is a simple solution.

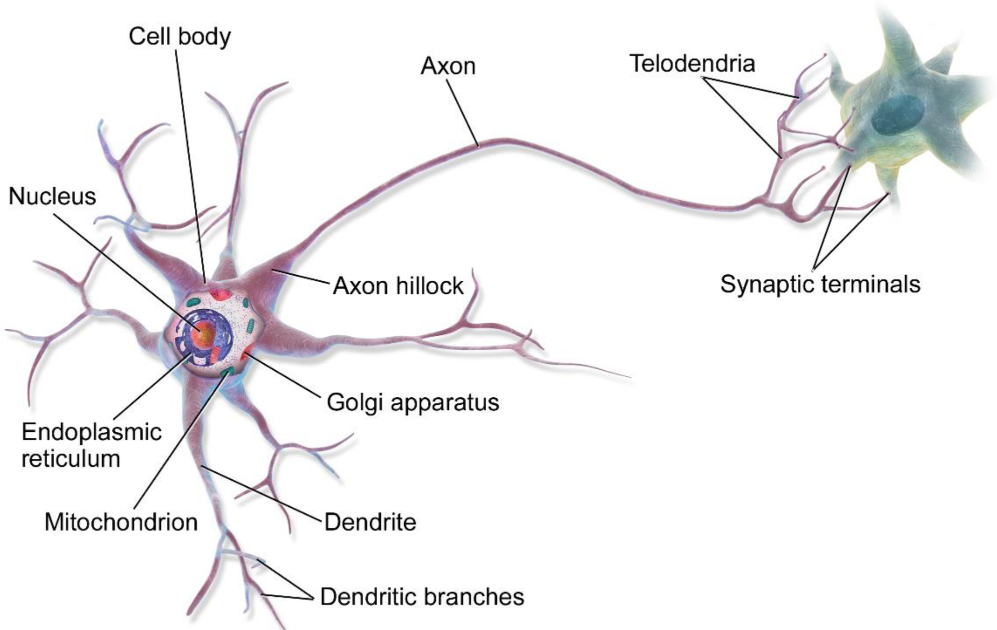
Binary Classifiers

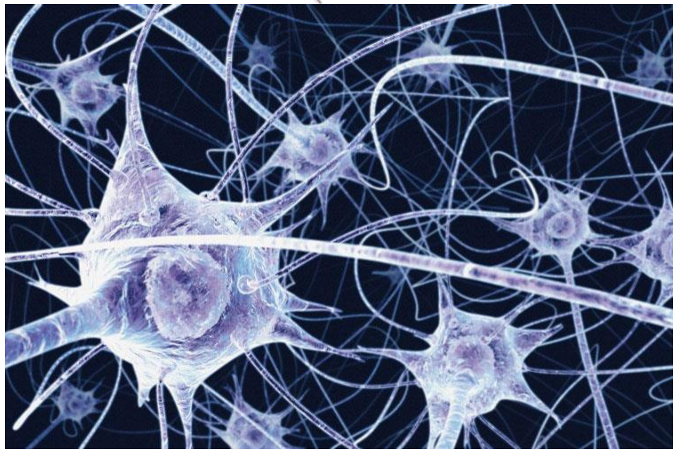
Binary Classifier

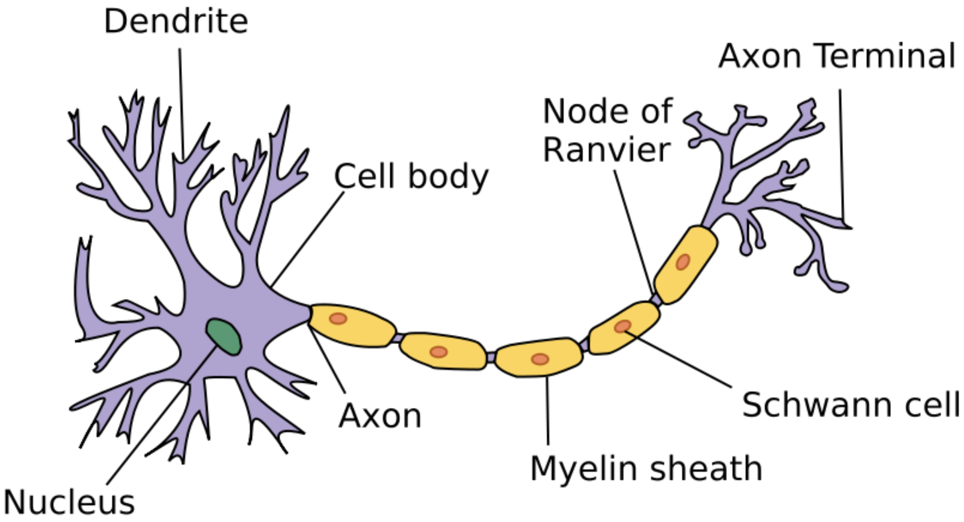
A function which can decide whether given input vector belongs to some specific class or not.

- It refers to those classification tasks that have two class labels
- A type of linear classifier
- A classification algorithm that makes its prediction based on a linear predictor function combining a set of weights with the feature vector
- Linear classifiers are artificial neurons

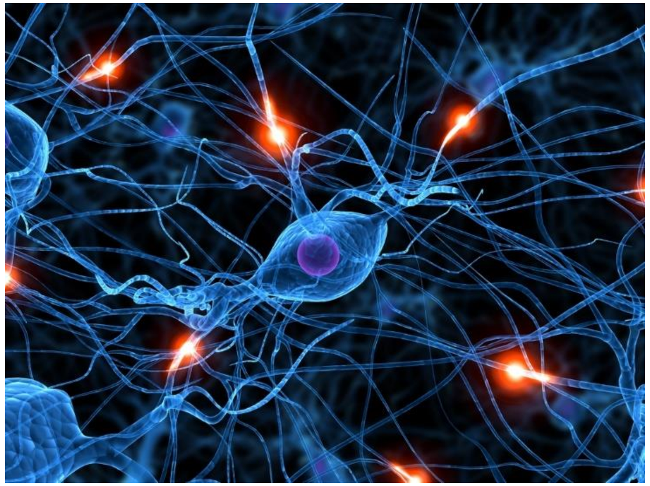
Human Brain: Mystique and Mystery

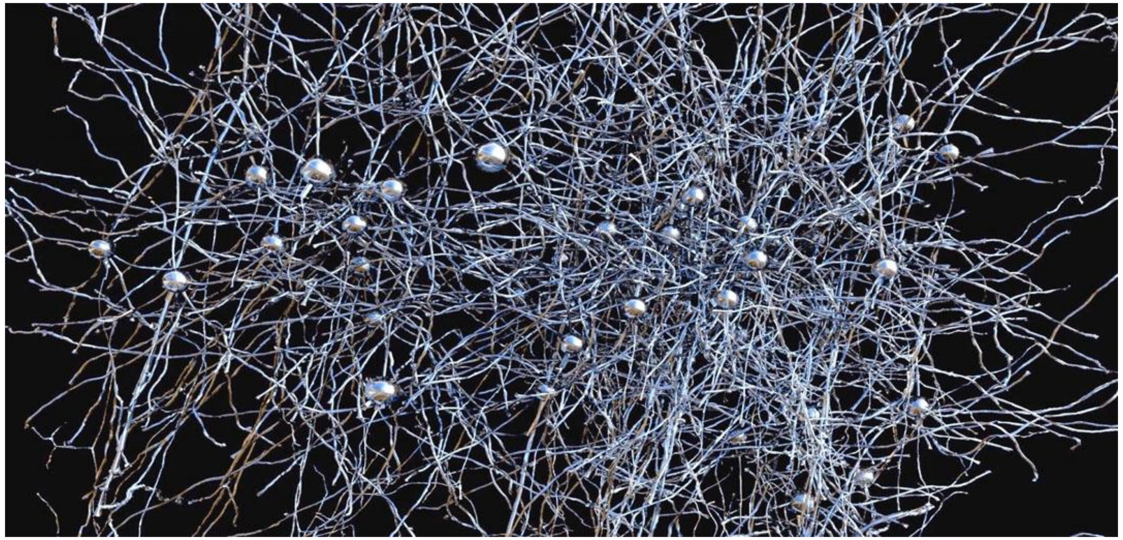






Human Brain: Mystique and Mystery





Linear Classifier

As artificial neurons, Linear classifiers have the following characteristics

- **Inputs**
  - Feature Values x
- **Weight**
  - Each Feature has a weight w
- **Activation Function**
  - Weighted sum
  - $\phi_w(x) = \sum_j w_j x_j = w^T x$
- **Decision/Output**
  - Class 1, if $\phi_w(x) > 0$
  - Class 2, if $\phi_w(x) < 0$

Perceptron

Invented by Frank Rosenblatt (1957), Built on work of Hebbs (1949), Improved by Widrown-Hoff (1960),
Learning Methods for two-layer neural networks (1970)

Perceptron

Inputs

$x = (x_1, x_2, \cdots, x_n)^T$

Weight Vector

$w = (w_1, w, \cdots, w_n)^T$

Net Input

$z = \sum_j w_j x_j = w^T x$

Activation Function

$\phi(z) = 1$
If $z \geq \theta$

$\phi(z) = -1$
otherwise

θ is a threshold

Perceptron

Inputs

$x = (1, x_1, x_2, \cdots, x_n)^T$

Weight Vector

$w = (w_0, w_1, w, \cdots, w_n)^T$

Net Input

$z = w_0 + \sum_j w_j x_j = w^T x$

Activation Function

$\phi(z) = 1$
If $z \geq 0$

$\phi(z) = -1$
otherwise

$w_0 = -\theta$, called bias in ML

Mathematical view of Perceptron

$$z = w_0 + \sum_j w_j x_j = w^T x$$

Let us take $n = 1$ and see, $z = w_0 + w_1 x_1 \Rightarrow y = ax + b$

The equation $z = w^T x$ represents a hyperplane in $\mathbb{R}^n$, whereas w_0 decides the intercept

What is unknown here?

1. Initialize weights to 0 or small random numbers
2. For each training sample x^i
   a) Find the output value $y^i = \phi(z^i)$
   b) Update the weights

Update Weight Vector

$$w = w + \Delta w, \Delta w = \eta(y^i - \bar{y}^i)x^i$$

η is the learning rate,
$0 < \eta < 1,$
y^i is the true class label of the i^{th} training sample,
$\bar{y}^i$ is the predicted class label of the i^{th} training sample

What will be Δw?

1. If prediction is correct
2. What will be it if the prediction is wrong

Separable Dataset

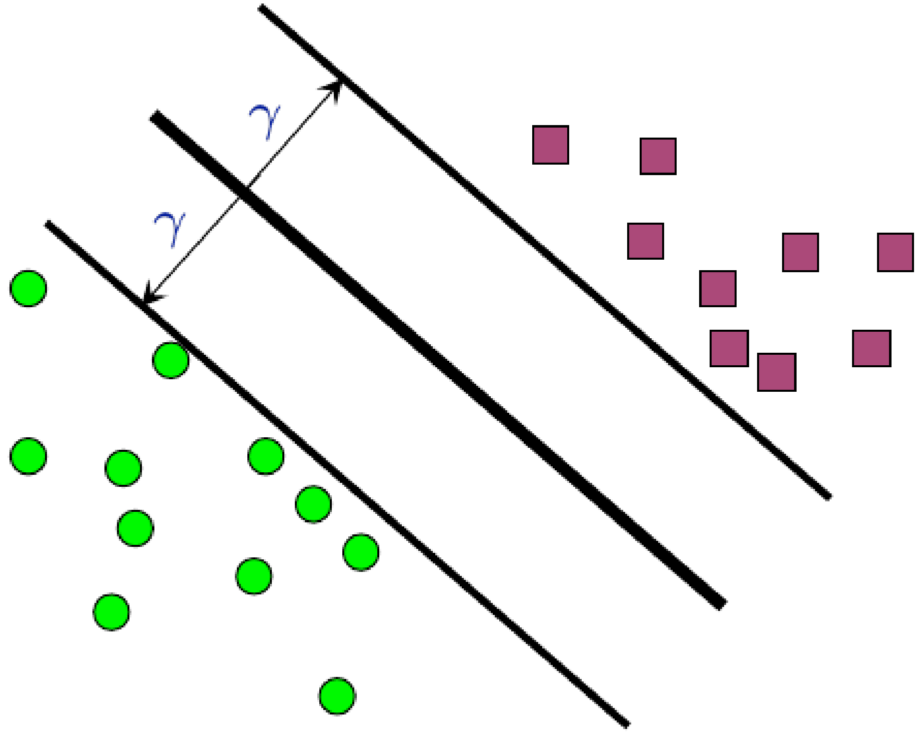
A dataset $\{(x^i, y^i)\}$ is linearly separable if there exists $\hat{w}$ and γ such that

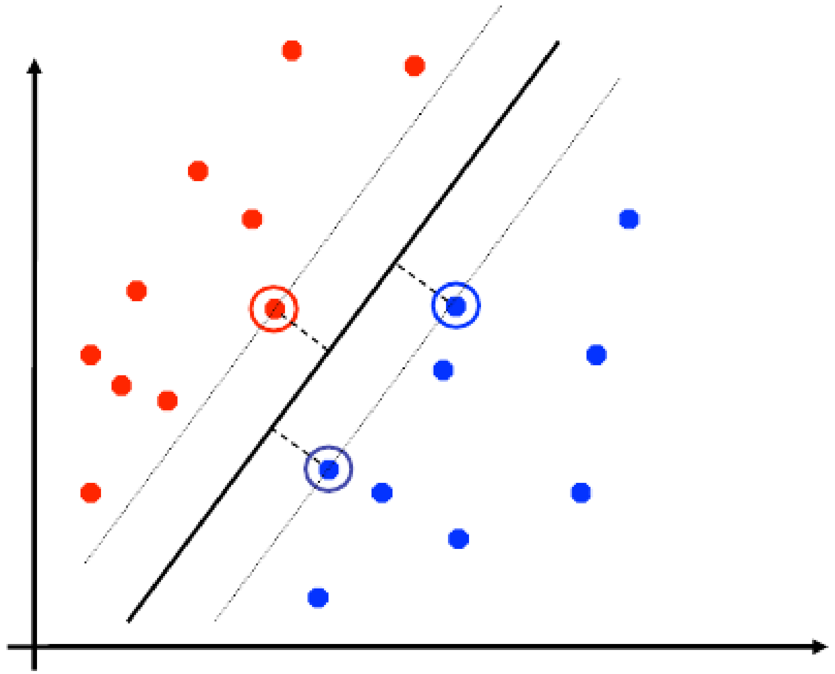
$$y^i \hat{w}^T x^i \geq \gamma > 0, \forall i$$

where γ is called the margin

Let X and Y be two sets of points in an $\mathbb{R}^n$. Then X and Y are linearly separable if there exists $w \in \mathbb{R}^n$ and $k \in \mathbb{R}$ such that every point $x \in X$ satisfies $w^T x > k$ and every point $y \in Y$ satisfied

$$w^T y < k$$





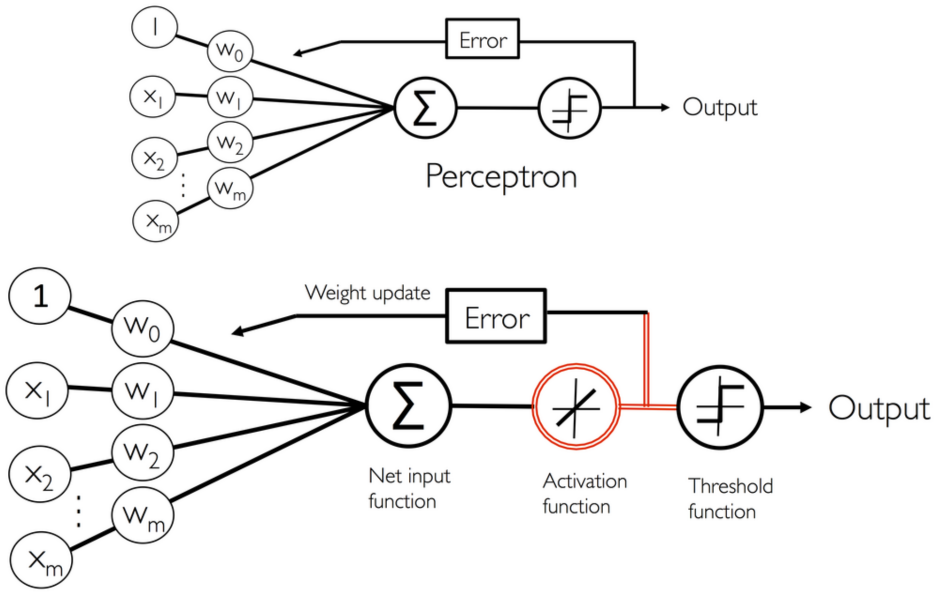
For linearly separable training dataset

1. Perceptron always converge
2. **Separability:** Some weights get the training set perfectly correct

Support Vector Machine (SVM) chooses the linear separator with the largest margin.

Adaline Algorithm

1. Weights are updates based on $\phi(z)$
2. Suppose $\phi(z) = z$ (Identity Function)
3. This algorithm is interested to define a cost function and minimize it
4. Continuous cost function allow the ML optimization problem to Calculus Problem



Adaptive Linear Neuron (Adaline)

Given a dataset $\{(x^i, y^i), i = 1,2,\cdots,N\}$

Learn the weights w_i and bias $b = w_0$

Activation Function

$$\phi(z) = z$$

Cost Function (SSE)

$$\mathcal{J}(w,b) = \frac{1}{2}\sum_i \left(y^i - \phi(z^i)\right)^2$$

$$z^i = w^T x^i + b$$

Dominant algorithm for the minimization of the cost function

Compute $-\nabla\mathcal{J}$ for the search direction (update direction)

$$w = w + \Delta w \ = w - \eta\nabla_w\mathcal{J}(w,b)$$

$$b = b + \Delta b = b - \eta\nabla_b\mathcal{J}(w,b)$$

Where $\eta > 0$ is the step length (learning rate)

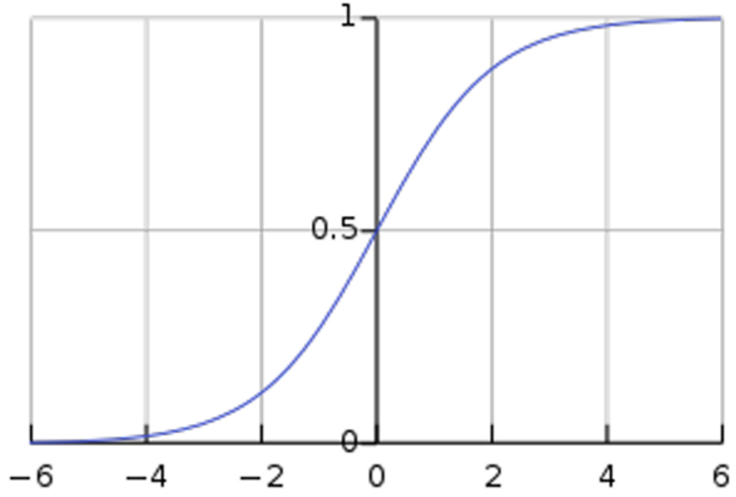
$$\Delta w \ = -\eta\nabla_w\mathcal{J}(w,b) = \eta\sum_i \left(y^i - \phi\left(z^i\right)\right)x^i$$

$$\Delta w \ = -\eta\nabla_b\mathcal{J}(w,b) = \eta\sum_i \left(y^i - \phi\left(z^i\right)\right)$$

$$\phi(z) = \begin{cases} 1 & if\ z \geq 0 \\ -1 & otherwise \end{cases}$$

$$\phi(z) = \frac{1}{1 + e^{-z}} = \frac{e^{z}}{1 + e^{z}}$$

This helps to identify the probability of individual classes



- **Assumption**: Given a unlabeled dataset $\{x_i\}$,
- **Unsupervised Learning:**
  - Given: Training Set $\{x_i | i = 1,2,\cdots,N\}$
  - Find a similar cluster or density estimation or dimensionality reduction

🌡 **Assumption**: Given a unlabeled dataset $\{x_i\}$,

$$\min_{\mathcal{C}} \sum_{i} \sum_{c \in \mathcal{C}} \mathbb{I}(i, c) \|x_i - \mu_c\|^2$$

$\mathcal{C}$: set of clusters

$\mathbb{I}(i, c)$: indicator function

$$\mathbb{I}(i, c) = \begin{cases} 1 & x_i \in c \\ 0 & x_i \notin c \end{cases}$$

μ_c: Centroid of the cluster

🌡 **Assumption**: Given a unlabeled dataset $\{x_i\}$, estimate the probability distribution (MLE)

$$\hat{p}(x) = \arg\max_{p(x)} \prod_i p(x_i)$$

$p(x)$: Probability density functions of the data

Find the distribution that maximizes the MLE.

It is the science of decision-making combining ML and Optimal Control

- Learning the optimal behavior in a dynamic environment - maximum reward.
- Optimal behavior is learned through interactions with the environment and observations of how it responds
- No need for labeled input/output pairs
- In the absence of a supervisor, the learner must independently discover the sequence of actions that maximize the reward.
- This discovery process is similar to a trial-and-error

🌡 **Assumption**: Given a unlabeled dataset $x_i \in \mathbb{R}^d$, reduce to a low dimensional space $z_i \in \mathbb{R}^k$. Principal Component Analysis (PCA) can be formulated as finding the projection

$$z_i = W^T x_i$$

$W \in \mathbb{R}^{d \times k}$ is a projection matrix that maximizes the variance in the reduced space

$$\max_W \sum_i \|W^T x_i\|^2$$

Reinforcement Learning

It is the science of decision-making combining ML and Optimal Control

- Learning the optimal behavior in a dynamic environment - maximum reward.
- Optimal behavior is learned through interactions with the environment and observations of how it responds
- No need for labeled input/output pairs
- In the absence of a supervisor, the learner must independently discover the sequence of actions that maximize the reward.
- This discovery process is similar to a trial-and-error

Agent: The learner or decision maker

Environment: The external system with which the agent interacts

State (s_t):The representation of the current system if the environment at time step t

Action (a_t):The action taken by the agent at time step t

Reward (r_t):The scalar feedback received after taking action (a_t) at time step t in state s_t

Policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$, where $\mathcal{S}$ set of all states, $\mathcal{A}$ set of all actions

Value Function (V^{π}):Estimates how good a particular state

Action-Value Function (Q^{π}):Estimates the expected cumulative reward

Markov Decision Process

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$$

$\mathcal{S}$: Possible States

$\mathcal{A}$: Possible actions

$P(s_{t+1}|s_t, a_t)$: probability of moving from state s_t to s_{t+1} when action a_t is taken

r_t: reward function, immediate reward after taking action a_t

$\gamma \in [0,1]$: discount factor, helps to identify future rewards relative to immediate rewards

Bellman Optimality Equation

Value Function and Bellman Equation

$$V^{\pi}(s_t, a_t) = \mathbb{E}^{\pi}[r_t + \gamma V^{\pi}(s_{t+1})]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}^{\pi}[r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1})]$$

$$V^*(s_t\ ) = \max_{\pi} V^{\pi}(s_t)\ \ and\ Q^*(s_t, a_t) = \max_{\pi} Q^{\pi}(s_t, a_t)$$

Optimal Action-Value Function

$$Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1}}\left[r_t + \gamma \max_{a_{t+1}} Q^*(s_t, a_t)\right]$$

Self-supervised Learning

Self-supervised learning is a type of machine learning where a model learns from unlabeled data by creating its own supervision signal. In other words, the model generates pseudo-labels or uses part of the data to predict another part, which allows it to learn useful representations of the data without requiring human-provided labels.

- **Assumption**: Given a unlabeled dataset $\{x_i\}$,
- **Self-supervised Learning:**
  - Given: Training Set $\{x_i | i = 1,2,\cdots,N\}$
  - Define pretext task to generate a supervisory signal from the data
  - Corrupted or masked input x_i^m
  - Target $y_i = x_i$

- **Define $f(x)$** model (neural network) that learns the transformation of the input data xx into a useful representation.
- Learned embedding of the input x_i

$$\mathcal{L}(\theta) = \sum_i \mathcal{L}_{task}(f(x_i^m), y_i)$$

What is ChatGPT?

GPT?

Generative Pretrained Transformers

LLM

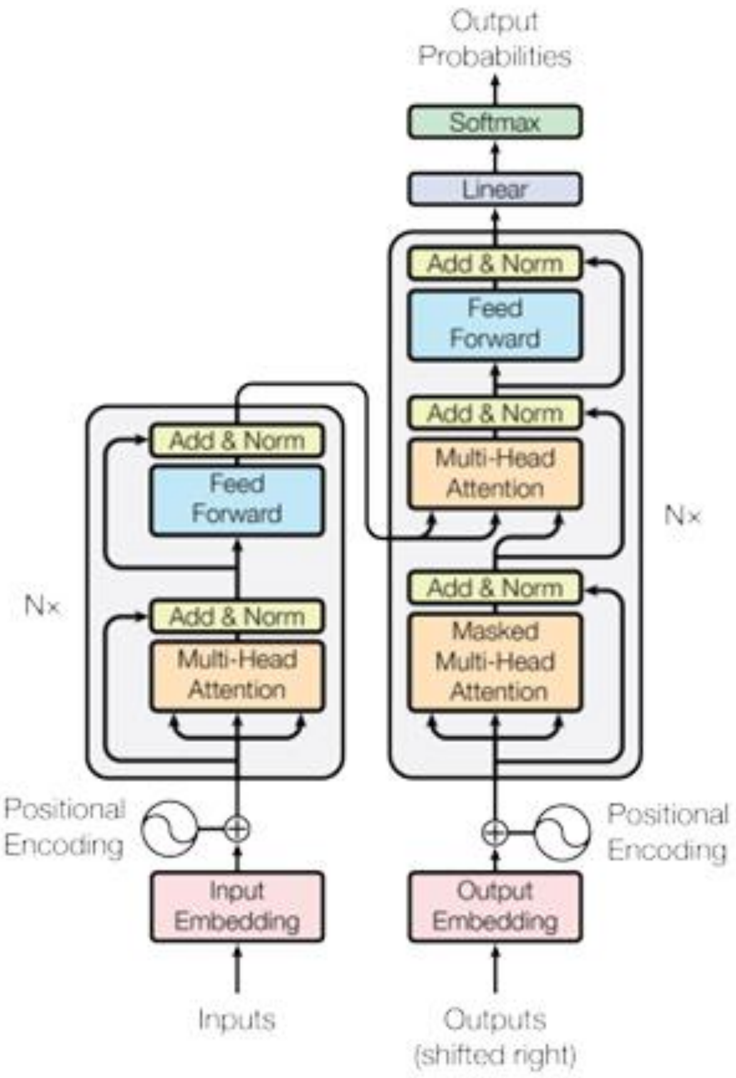
1. **A type of machine learning model designed for natural language processing (NLP) tasks such as language generation.**
2. **Language models with many parameters, and are trained with self-supervised learning on a vast amount of text.**

Large Language Models: Examples

1. ChatGPT(1,2,3,4,J,Neo,lite)
2. BERT
3. T5
4. XLNet
5. Claude
6. Gopher
7. LaMDA
8. LLaMA (2,3.1
9. DeepSeek(v1,v2,R1..)
10. Gemini (1,2,1.5,Ultra..)
11. PanGu-∑
12. Amazon Nova
13. BloombergGPT
14. T5

Attention

1. One pivotal development in this area is the transformer architecture, introduced in Vaswani et al.'s groundbreaking paper *"Attention Is All You Need"* in 2017 [Source: Medium.com]
2. RNNs and CNNs has struggles with parallelization
3. Transformer marked a significant departure from RNNs and CNNs
4. Built entirely on attention mechanisms
5. Enables models to efficiently process and generate language
6. Achieves state-of-the-art results across a range of NLP tasks


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

LLM: Key Features

1. Understanding Context
2. Generative Capabilities
3. Versatility across Domains
4. Continuous Learning
5. Efficiency
6. Scalability

Tokens

The cat sat on the mat

$$V = [The, cat, sat, on, the\ mat]$$

$$W = [T, h, e, c, a, t, s, a, t, o, n, t, h, e, m, a, t]$$

Token can be a word or part of a word or even a single character

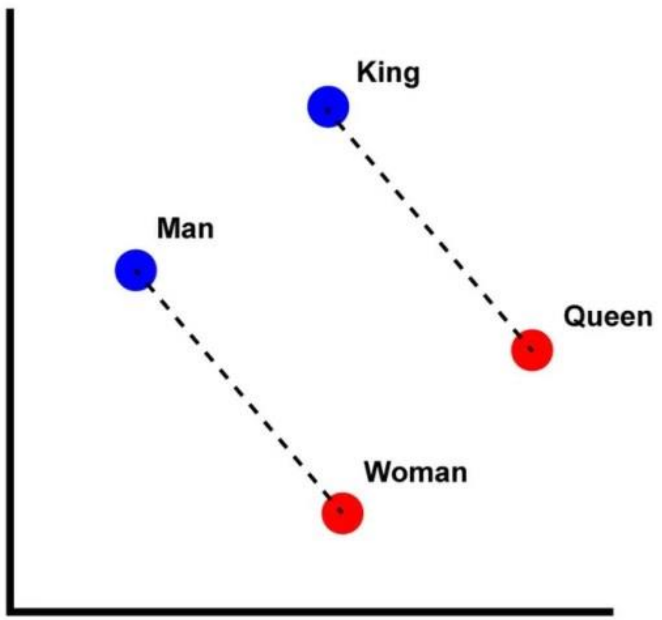
Why Tokenization

- **It transforms raw text into a format the model can understand**
- **Convert Tokens to Numbers**
- **Play with this numbers, embeddings**

Embeddings

- Numerical Representation of Tokens
- It will help the model to understand their learnings and relationships
- Each token is converted into a vector (a list of numbers) in a high-dimensional space.

Word2vec Project



Word2vec Project

$$king - man + woman = queen$$
$$v_{king} = [2, 5, 1]$$
$$v_{man} = [1, 2, 0]$$
$$v_{woman} = [0, 2, 3]$$
$$v_{queen} = [1, 5, 4]$$
$$[2, 5, 1] - [1, 2, 0] + [0, 2, 3] = [1, 5, 4]$$

Embedding

$$\boldsymbol{Embedding\ f: X^n \to \mathbb{R}^m}$$
$$\boldsymbol{f(x) = y}$$

x: **represents the input token**

y: **numerical vector**

n: **number of dimensions in the input space**

m: **number of dimensions in the embedding space**

Token: sparrow

Embedding: [0.25,0.78,0,45,…]

Self-Attention

One of the key innovations that allow LLMs to understand language so effectively is the **attention mechanism**

It finds which words (or tokens) in a sentence are most relevant to each other when generating responses.

Recall: $o \rightarrow r, o \rightarrow w$

Self-Attention

The cat chased the mouse because it was hungry

What does it refers here?

Self-Attention

Input Representation: Each word (token) in a sentence is first represented as a vector (its embedding)

Attention Scores: For each word by comparing it to every other word in the sentence. This is done using queries, keys, and values

Query (Q): A representation of the focused

Key (K): A representation of all other words

Value (V): The actual information we want to keep from each word

Attention Score

- The dot product of the query vector with the key vectors of all other words
- Measure of how relevant each word is to the query word.

$$AS = QK^T$$

Softmax function

- Commonly used in machine learning, especially for classification problems, as it transforms raw scores (logits) into probabilities.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Weighted Sum

- Finally, each word's value vector is multiplied by its attention score, and the results are summed to create a new representation of the word

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q: matrix of queries

K: matrix of keys

V: matrix of values

d_k: dimension of the keys

Sentence: The cat chased the mouse because it was hungry

Tokens: [The, cat, chased, the, mouse, because, it, was, hungry]

Embeddings: Convert each of these tokens to a vector

Query (Q): Embedding for the token it

Keys (K): The embeddings for all vectors

Values(V): The same embeddings

Attention Scores: Compute how well the query “it” relates to each of the keys from the other words

Weighting: The word “cat” would likely receive a higher score than “mouse” because “it” refers back to “cat”

Resulting Representation: The resulting representation for “it” would be a weighted sum of the value embeddings, emphasizing the context provided by the word “cat.”

Note: Attention mechanism gives the relationship between words, but not the position of the words

You require Positional Encoding (PE) for this

$$PE(\mathrm{pos}, 2\mathrm{i}) = \sin\left(\frac{\mathrm{pos}}{1000^{\frac{2\mathrm{i}}{\mathrm{d_{model}}}}}\right)$$

$$PE(\mathrm{pos}, 2\mathrm{i}+1) = \cos\left(\frac{\mathrm{pos}}{1000^{\frac{2\mathrm{i}}{\mathrm{d_{model}}}}}\right)$$

pos: position of the token in the sequence

i: dimension of the embedding vector

dmodel: total number of dimension in the embedding