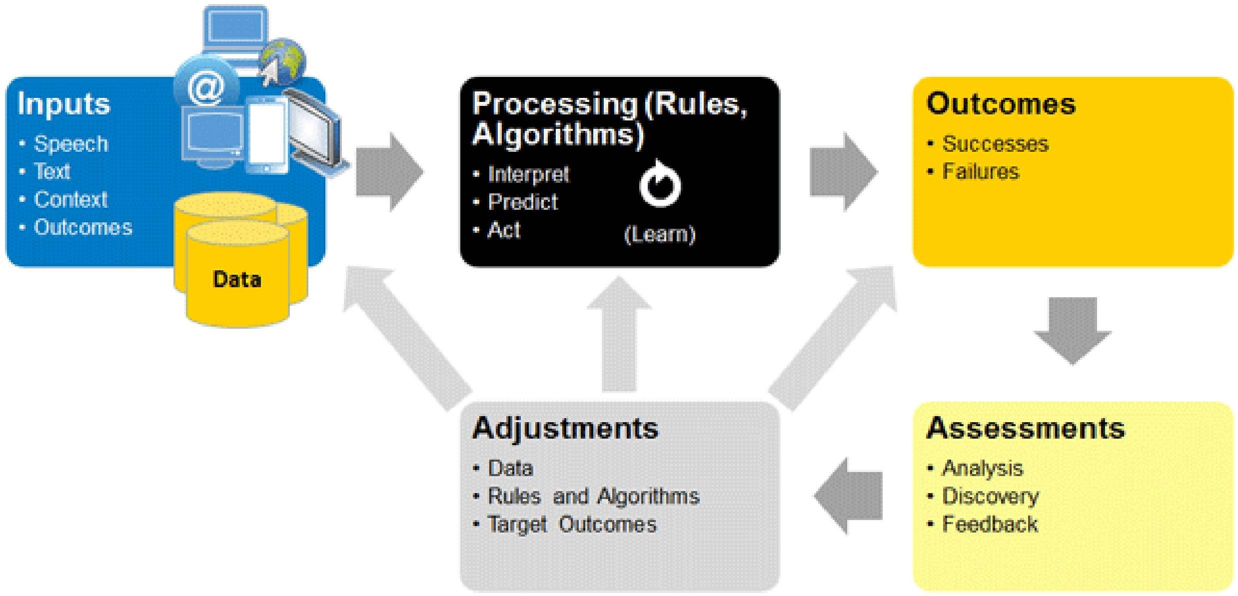
Artificial Intelligence

It’s a new beginning…

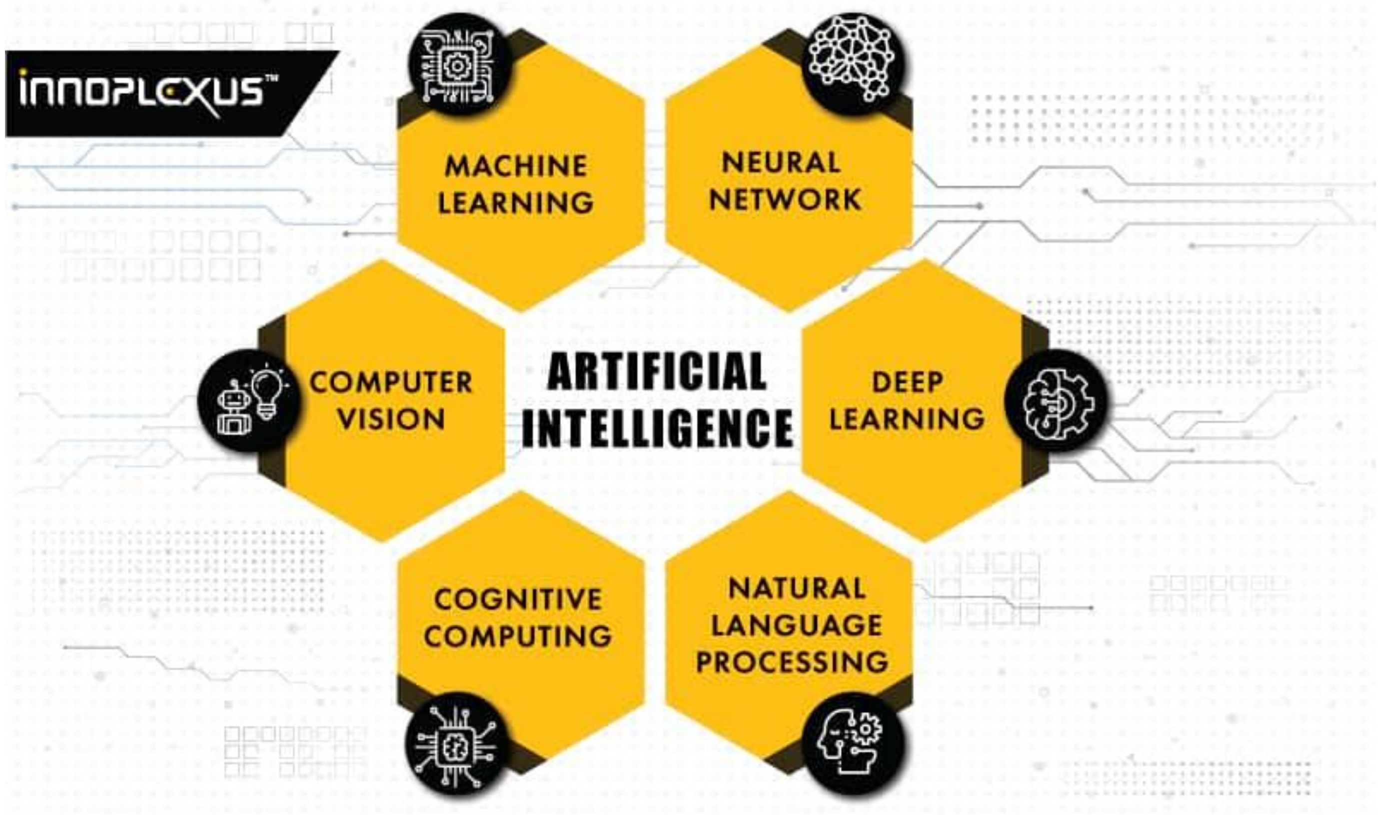
Machine Intelligence is the last invention that humanity will ever need to make

How AI Works?



AI works by combining large amounts of data with fast, iterative processing and intelligent algorithms, allowing the software to learn automatically from patterns or features in the data.

AI - Term



- Machine learning model
- Deep learning model
- Prediction model
- Neural network
- Cognitive computing
- Natural language processing
- Autonomous system
- Computer vision

Example of Ai Terms

Machine Learning

Example: Facial recognition

Deep learning

Example: Image Captioning

Predictions

Example: Predict human life expectancy

Autonomous

Example: Self Driving Car, Self-Flying Drone

Computer vision

Example: Facial Recognition, Identifying the things by just looking at

Cognitive computing

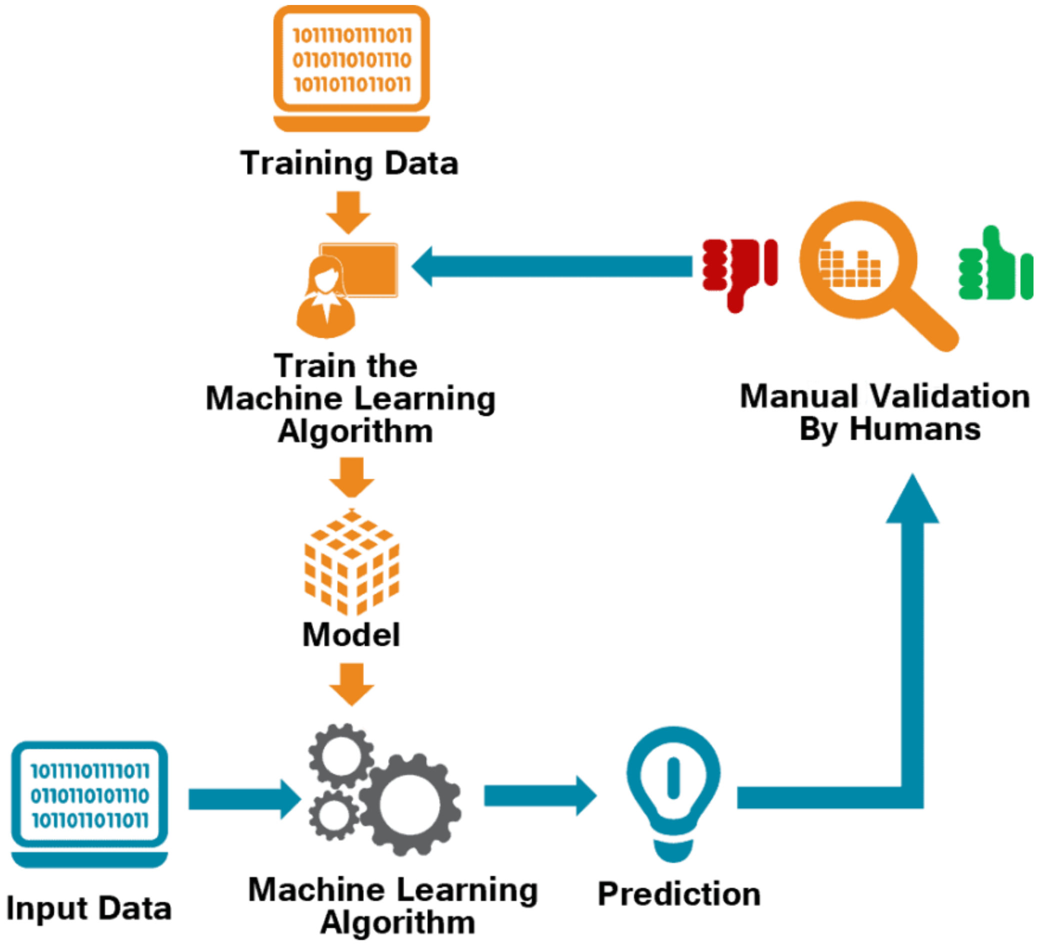
Example: Faster Thinking & suggestions.

Neural networks

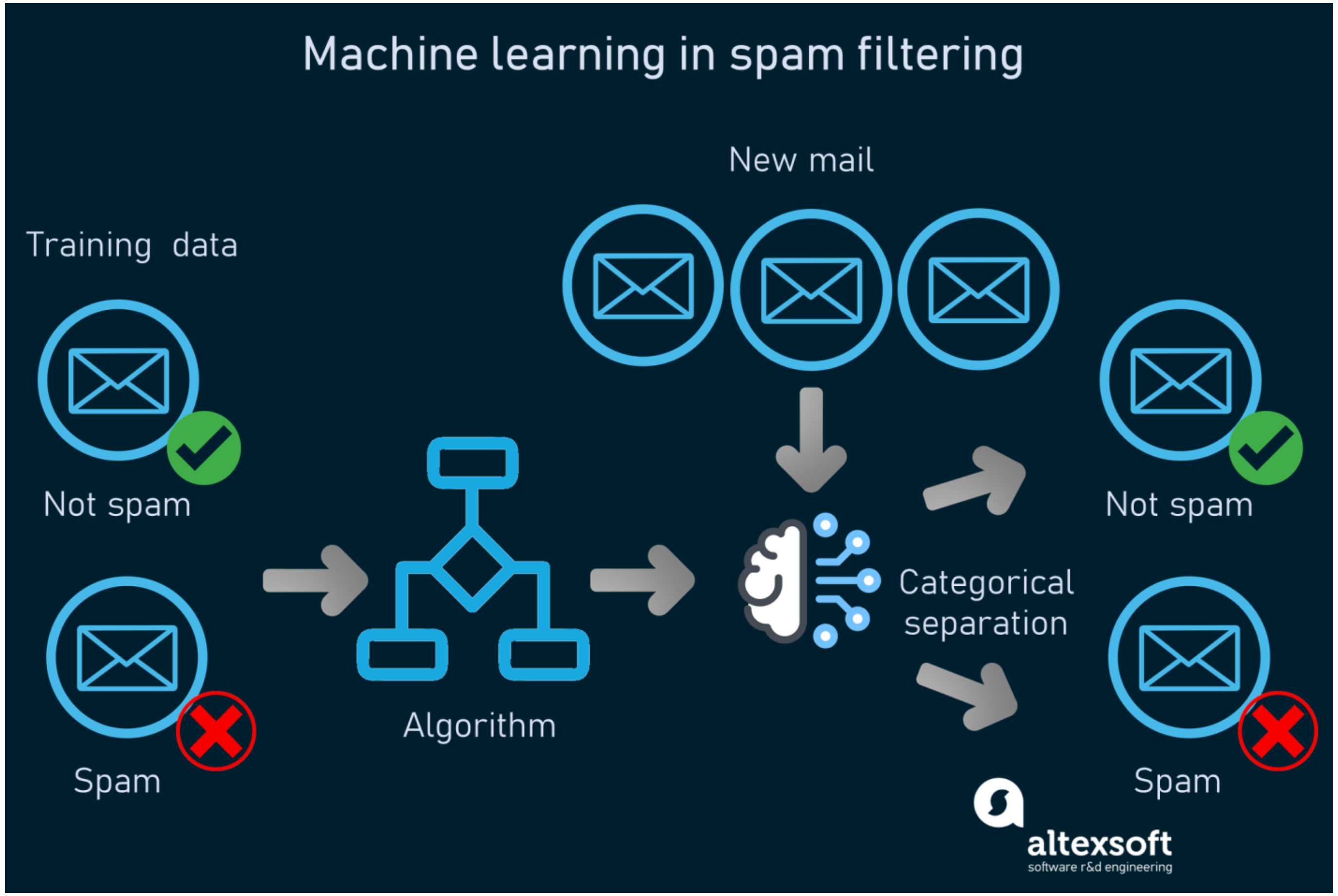
Example: Nodes connection & doing multiple jobs

Machine Learning Model
How it's works

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data

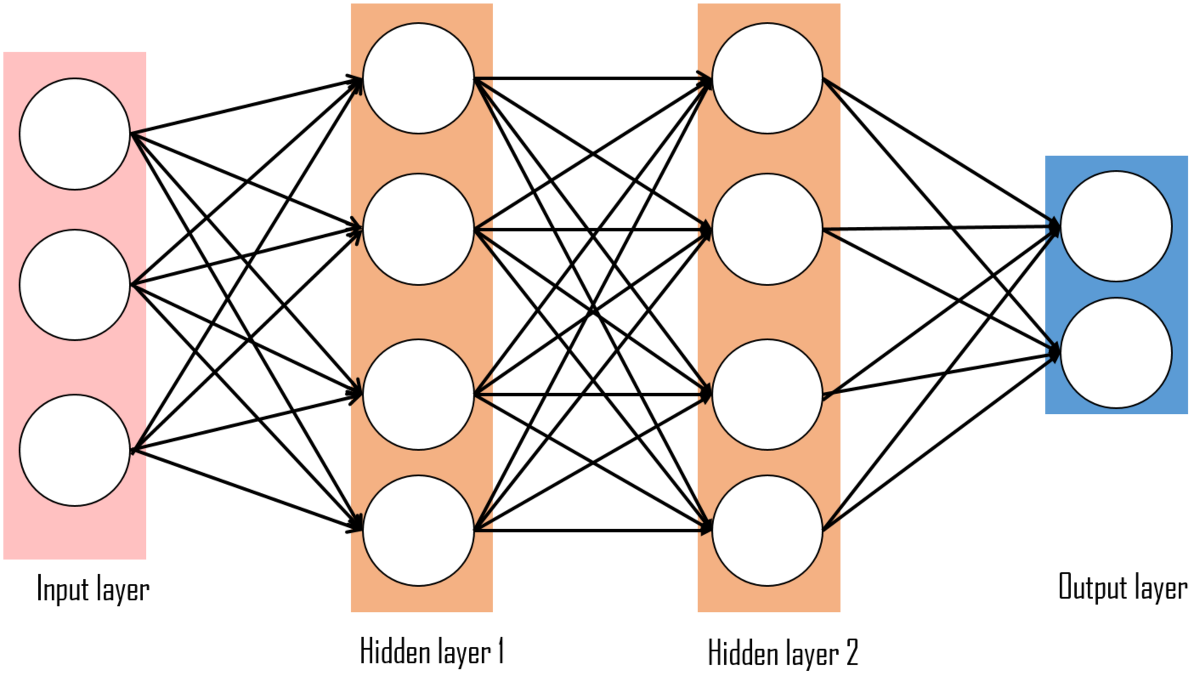


Example of ML

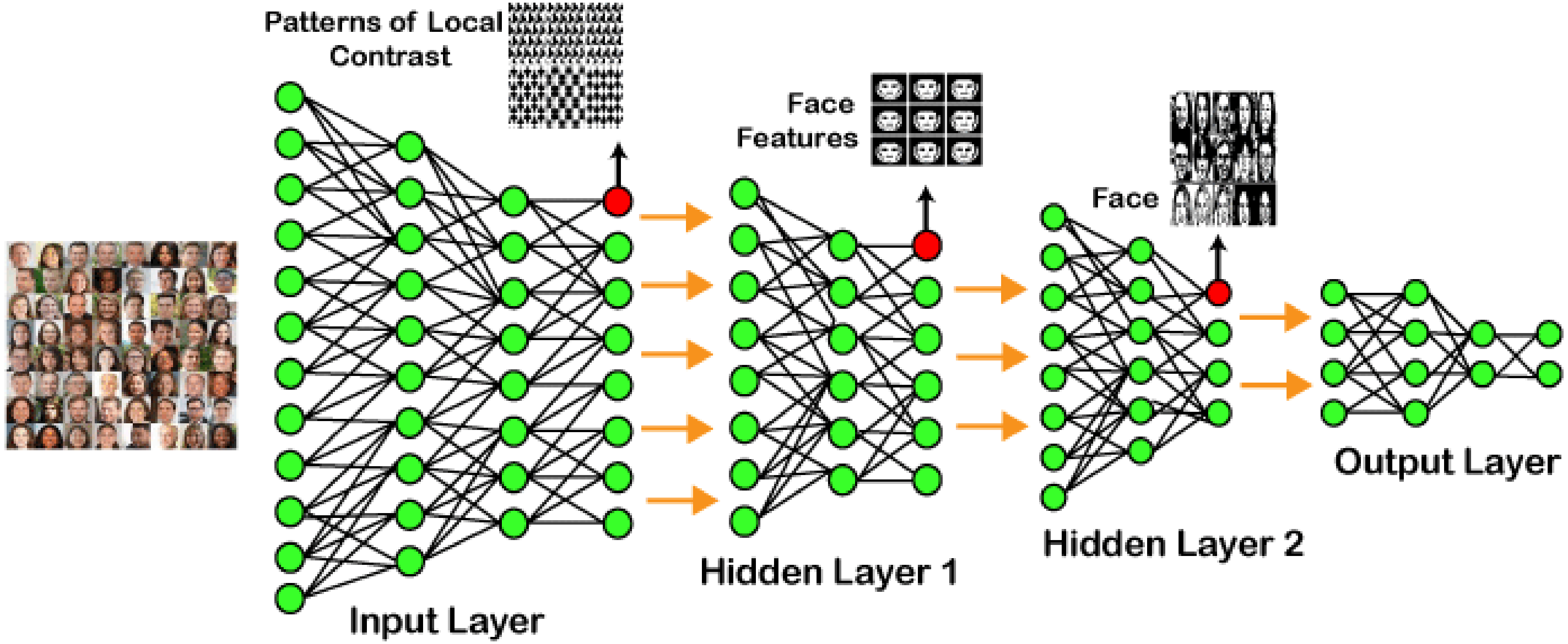


Deep learning
How it’s works

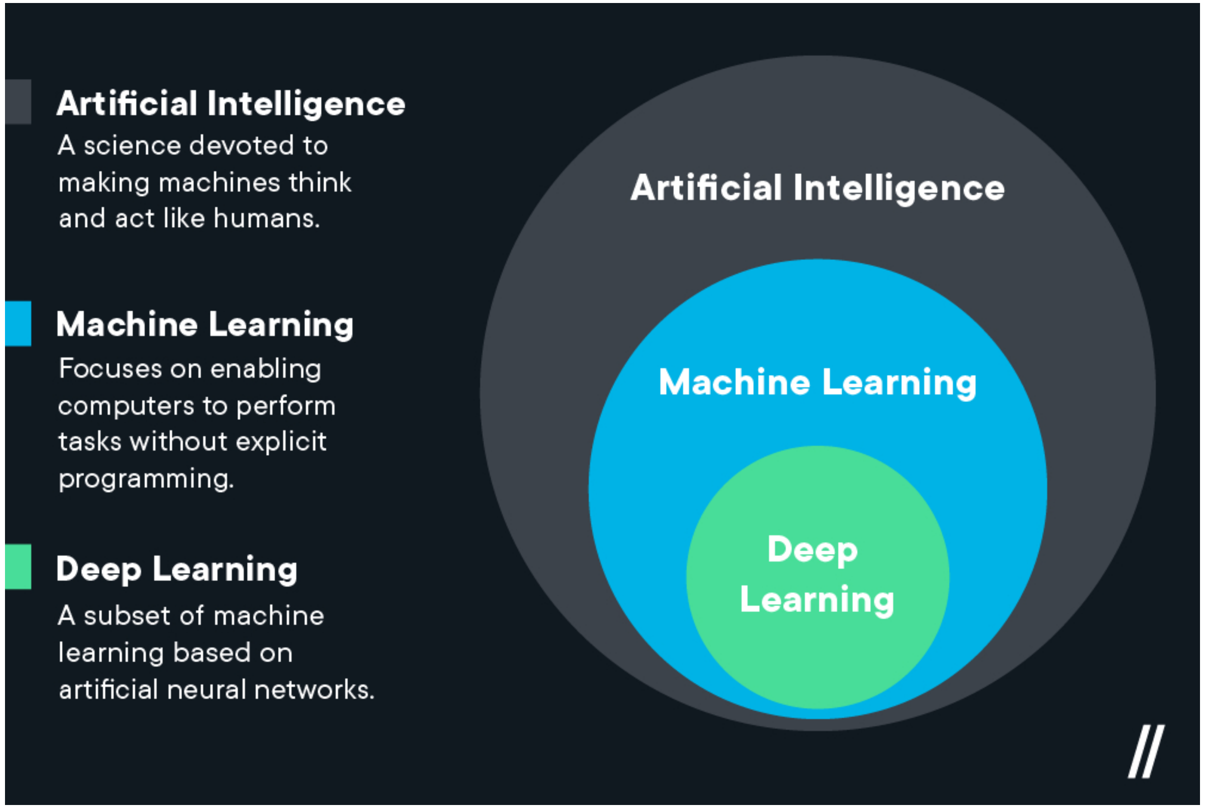
Deep learning networks learn by discovering intricate structures in the data they experience. By building computational models that are composed of multiple processing layers, the networks can create multiple levels of abstraction to represent the data

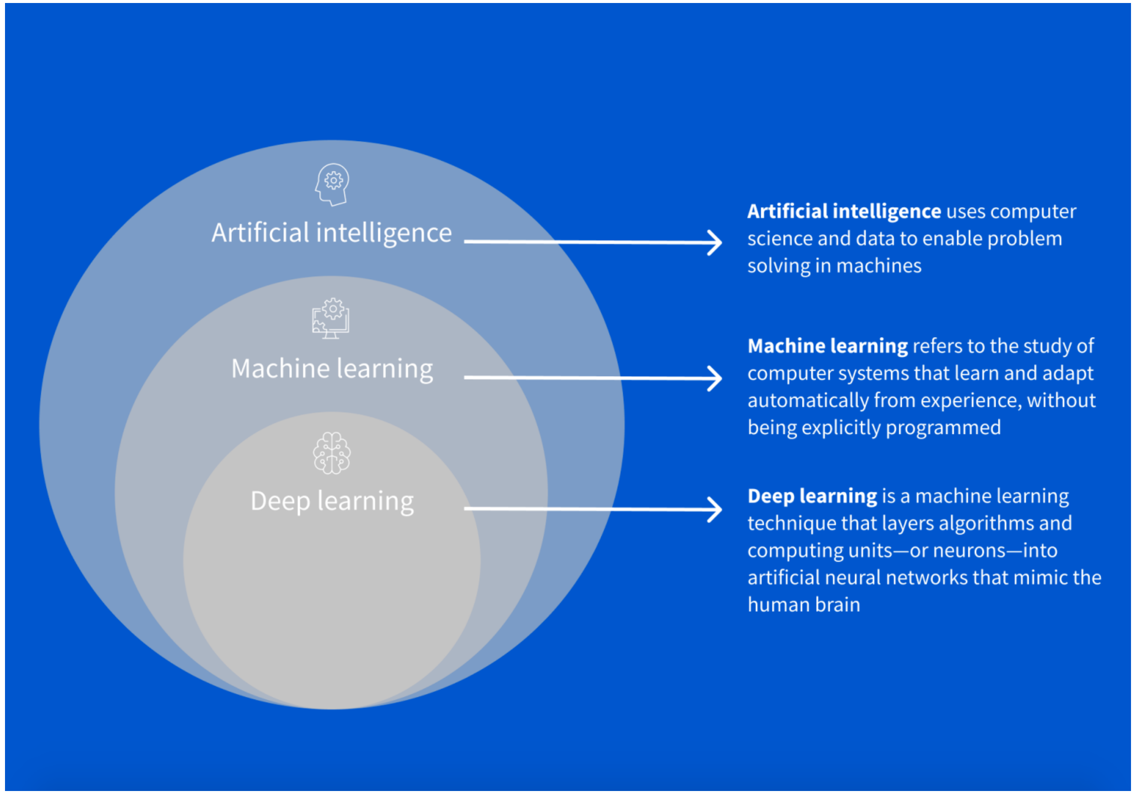


Example of Deep learning



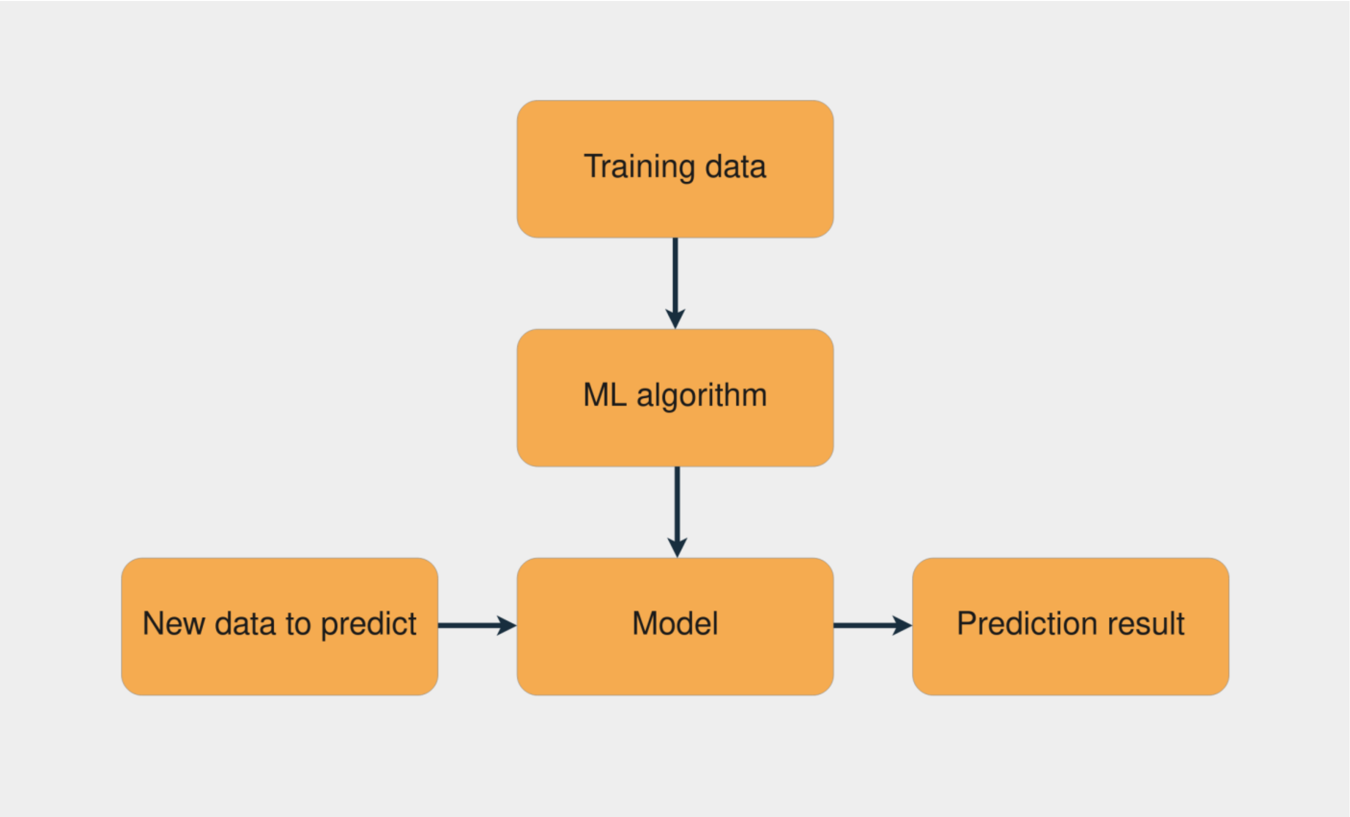
Deep Leaning V/S Machine Learning



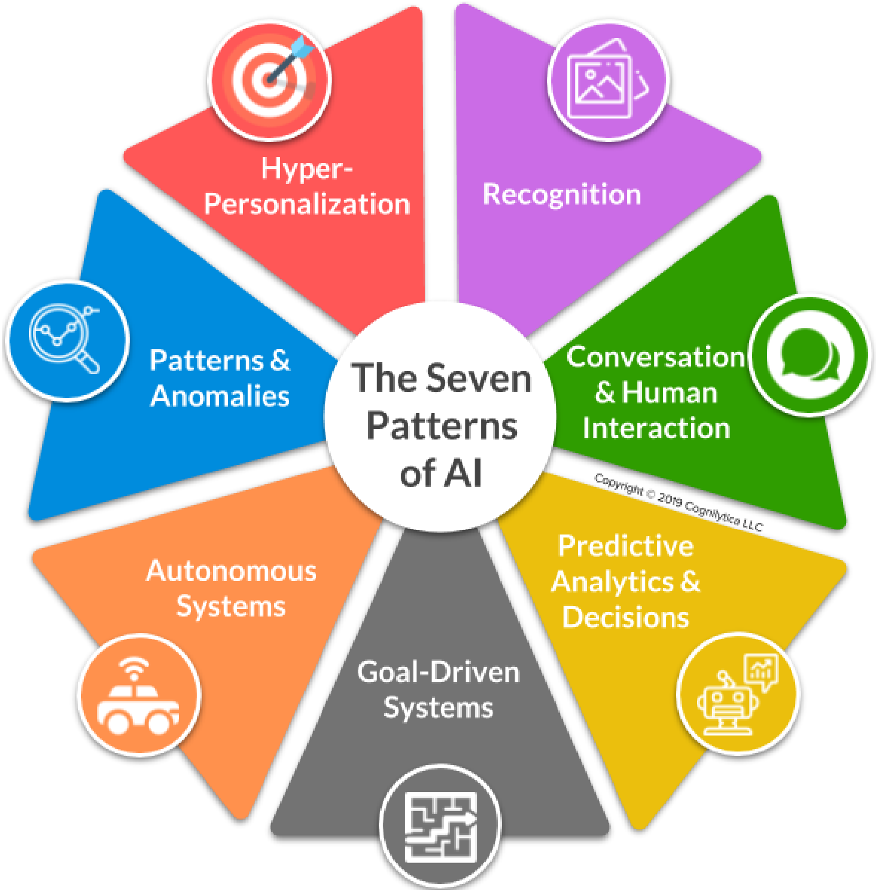


Predictions

Using individual customer data, AI could be used to predict customer needs and expectations better, reducing handling times and helping to identify future trends. AI can also be used to help optimize supply chains by analyzing huge volumes of data to predict demand across multiple product segments and geographies.

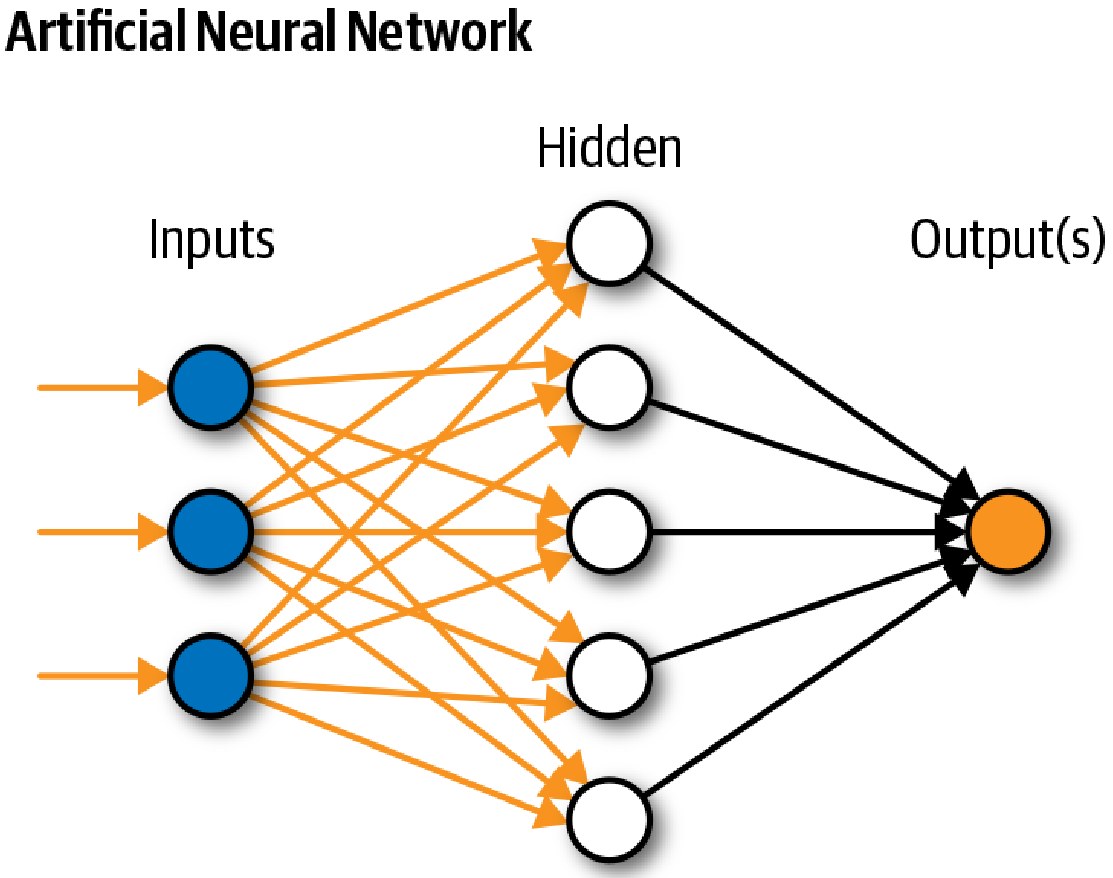


Example of Predictions



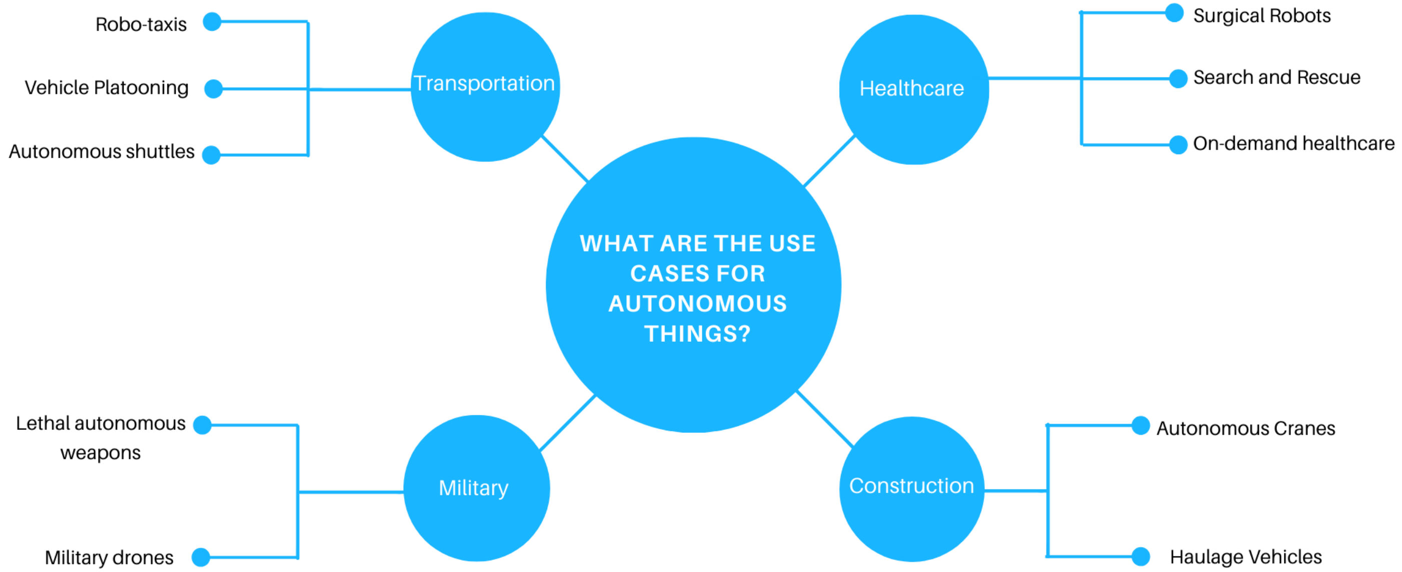
Neural networks

- A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.



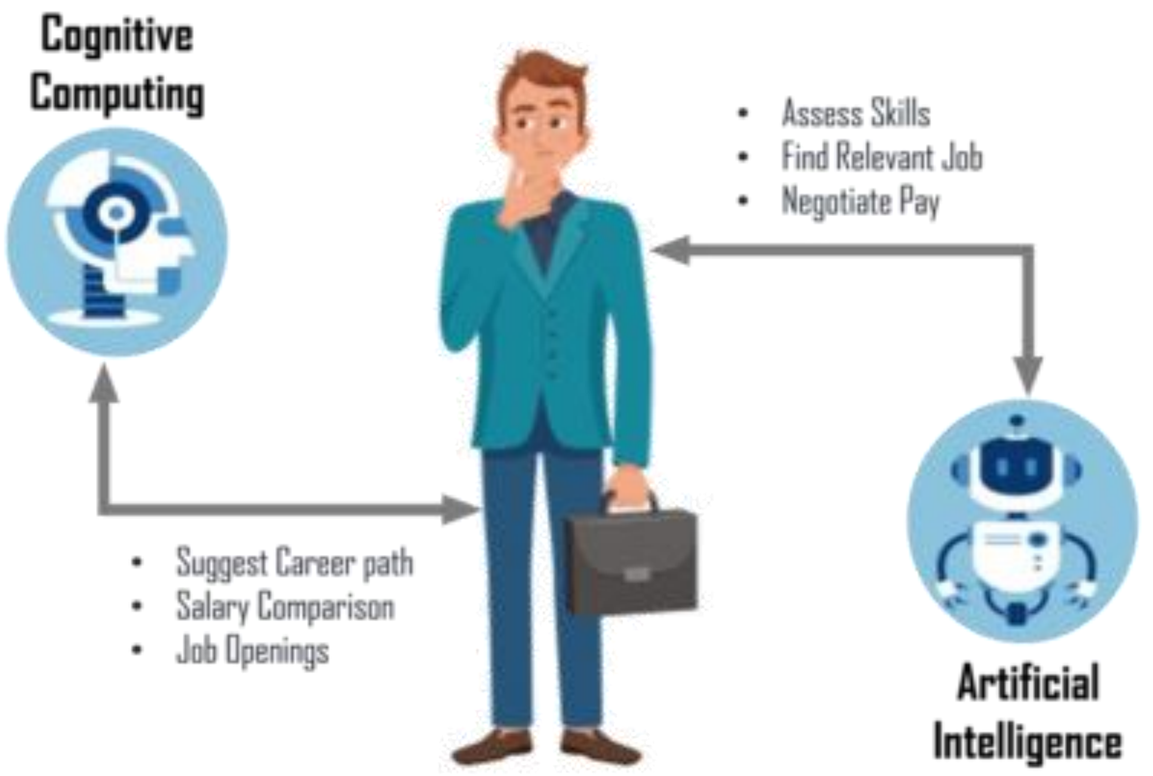
Autonomous AI

Autonomous Intelligent Systems are AI software systems that act independently of direct human supervision, e.g., self-driving cars, UAVs, smart manufacturing robots, care robots for the elderly and virtual agents for training or support.



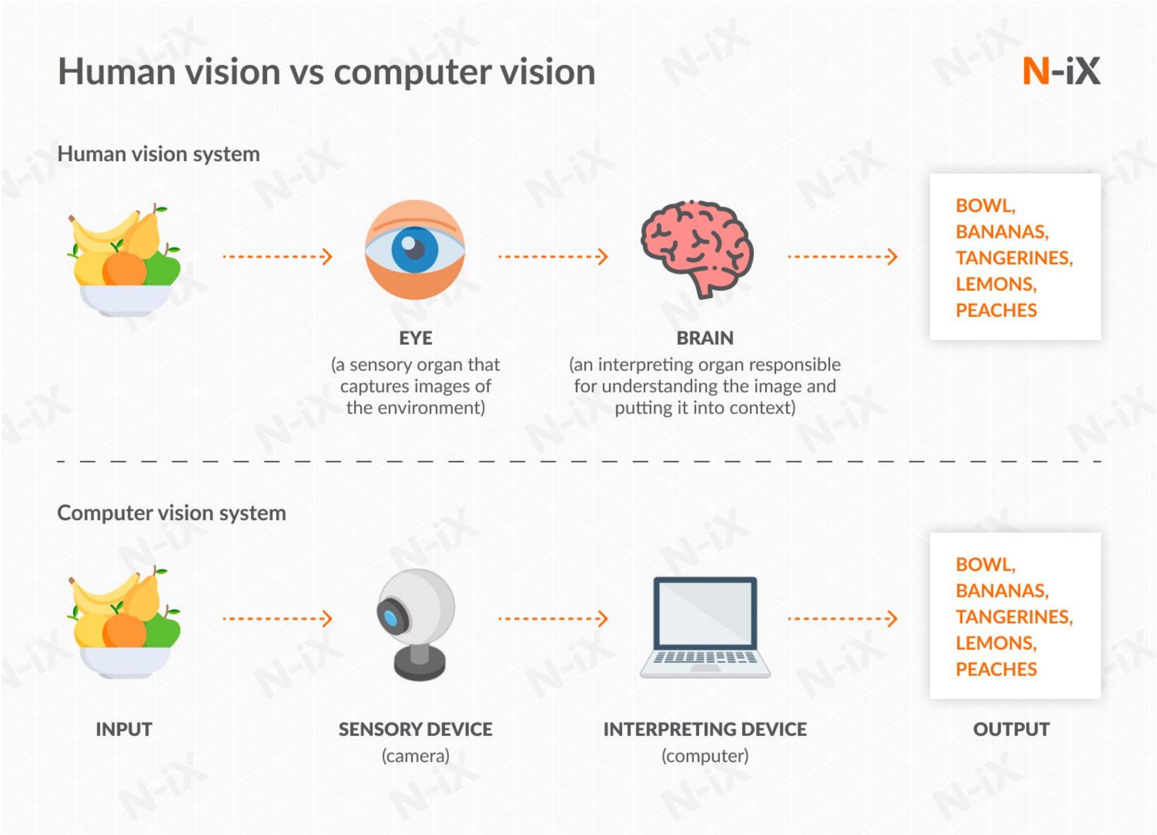
Cognitive computing

Cognitive computing refers to technology platforms influenced by cognitive science to simulate the human thought process and encompass artificial intelligence and signal processing.



Computer vision

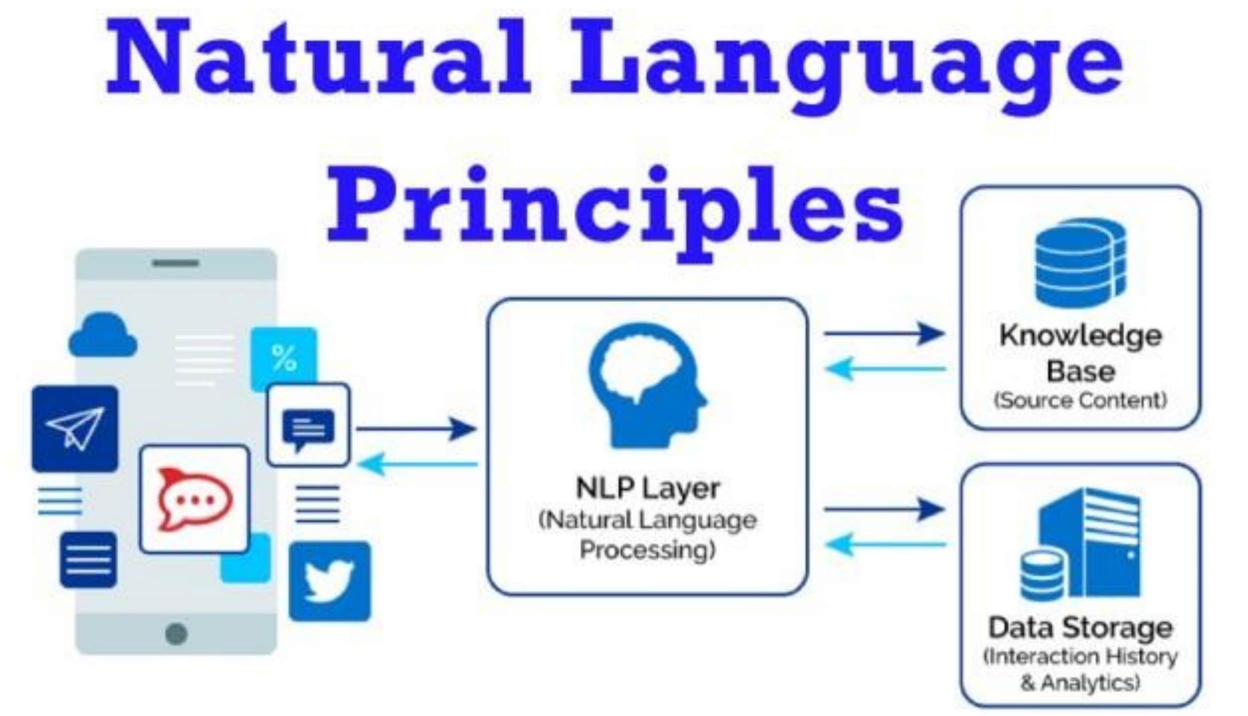
Computer vision, a type of artificial intelligence, enables computers to interpret and analyze the visual world, simulating the way humans see and understand their environment. It applies machine learning models to identify and classify objects in digital images and videos, then lets computers react to what they see.



Example Computer vision

Natural language processing

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.



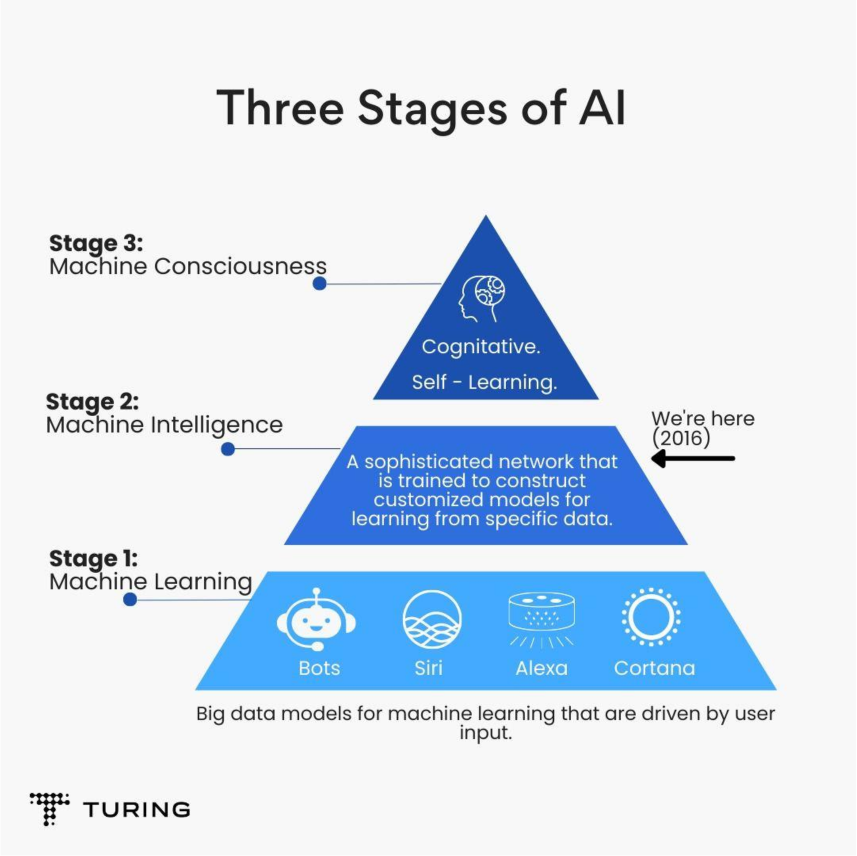
Example of Natural language processing

Artificial intelligence stages

Stage 1: Artificial Narrow Intelligence (ANI)

Stage 2: Artificial General Intelligence (AGI)

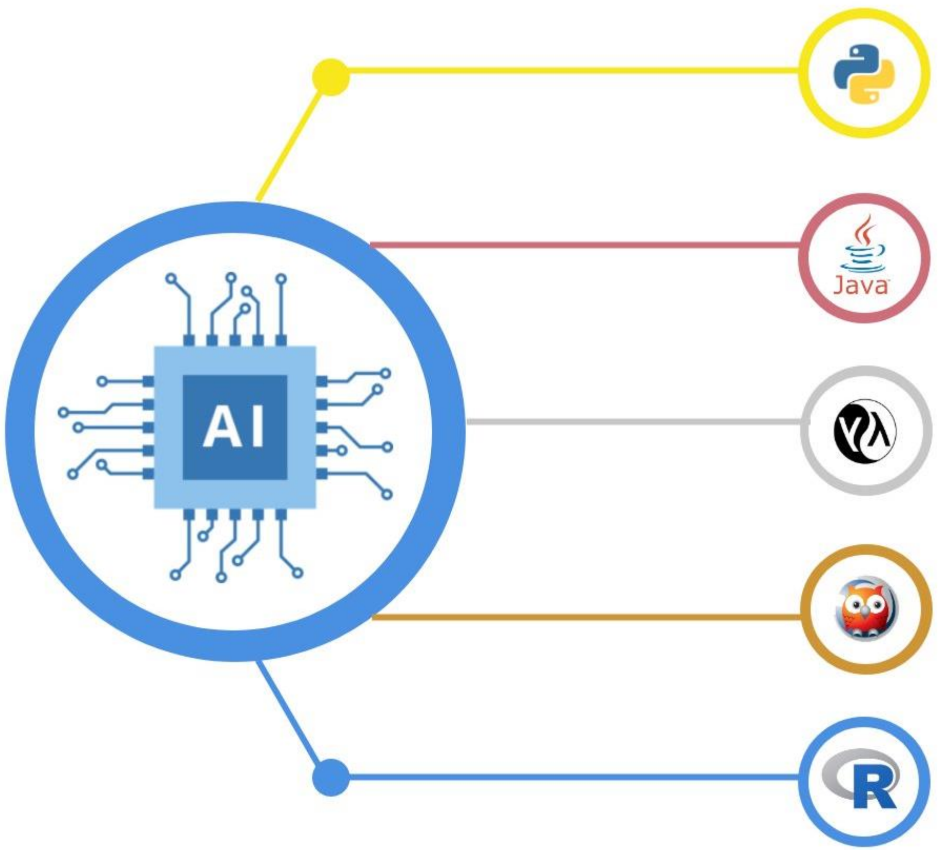
Stage 3: Artificial Super Intelligence (ASI)



Best Programming Language for AI

- PYTHON
- R
- JAVA
- LISP - By Artificial Intelligence father developed
- Prolog
- C++
- JAVA SCRIPT

Which language is best for Ai



Python
Easy to learn, simple syntax and a lot of frameworks and libraries.

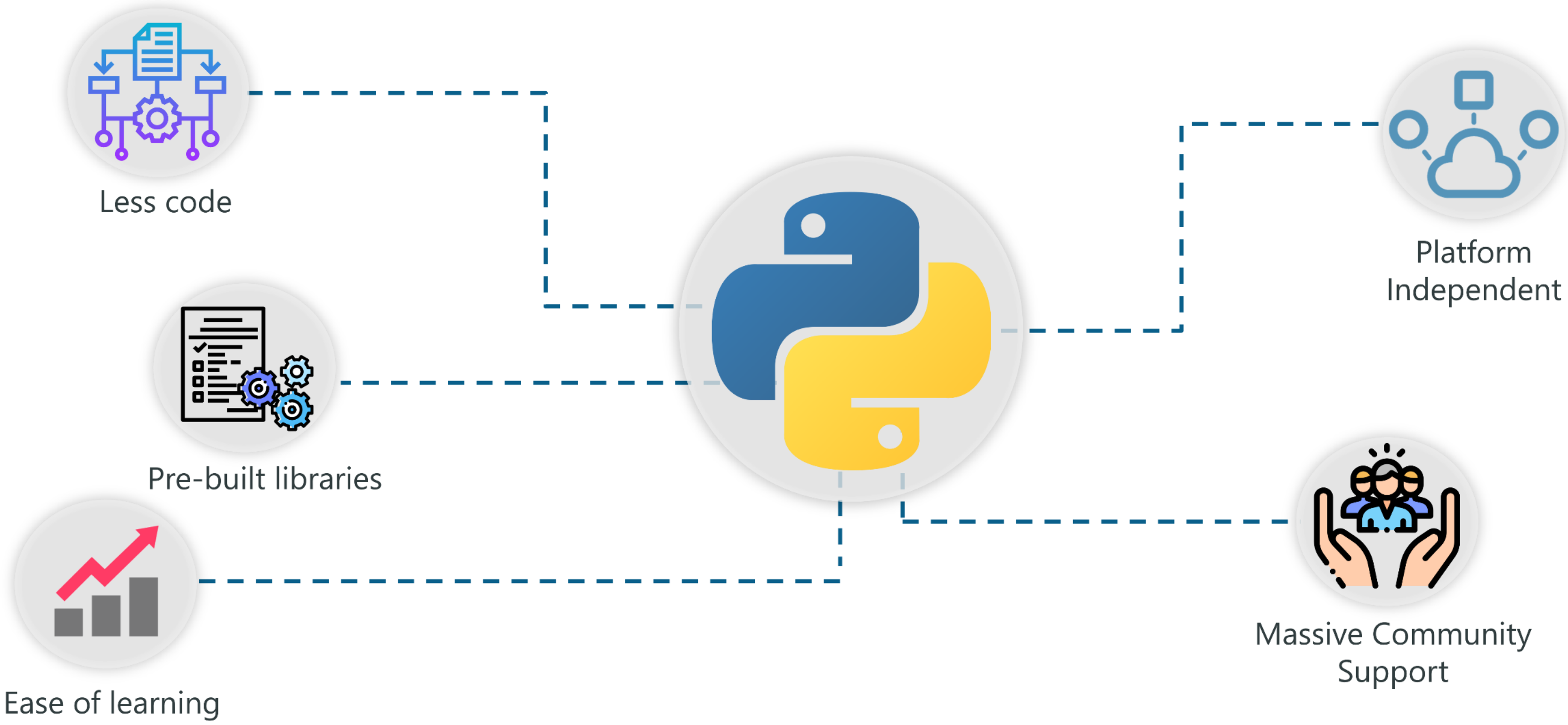
Java
Java is also be considered as a good choice For AI development

Lisp
One of the oldest and the most suited languages for the development in AI

Prolog
One of the logic programming language specifically designed for AI development.

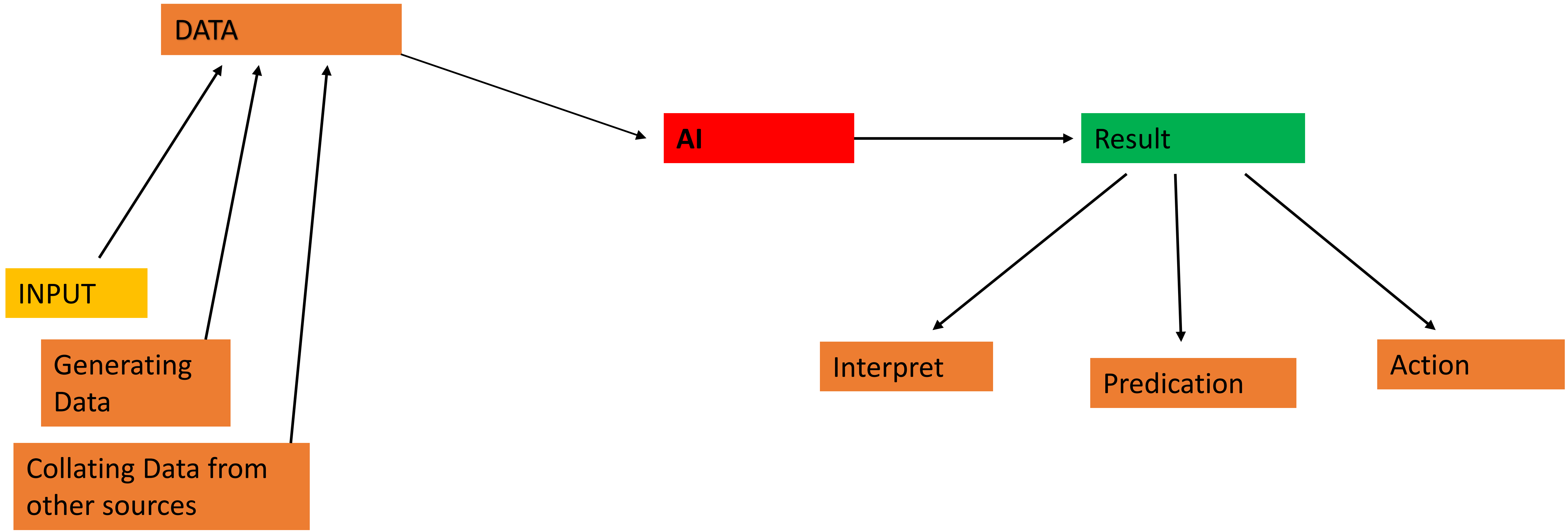
R Programming
Comprehensive statistical language that encourages the development of new ideas.

Python language is best for Ai

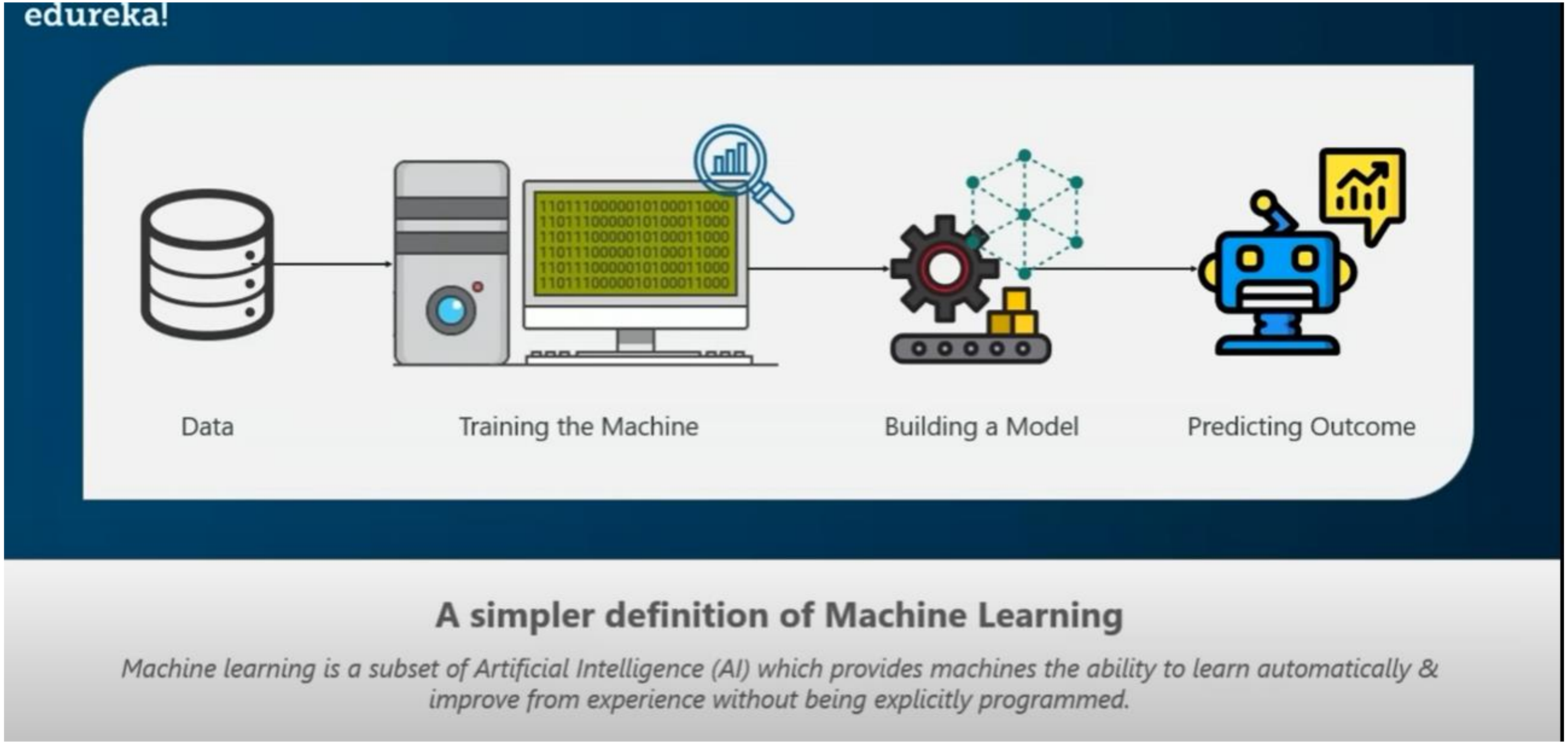


Summary Of AI

For AI Data is everything
If you want to make your Ai you have to have data as much possible.



Summary Of AI



edureka!

MACHINE LEARNING DEFINITIONS

Algorithm: *A set of rules and statistical techniques used to learn patterns from data*

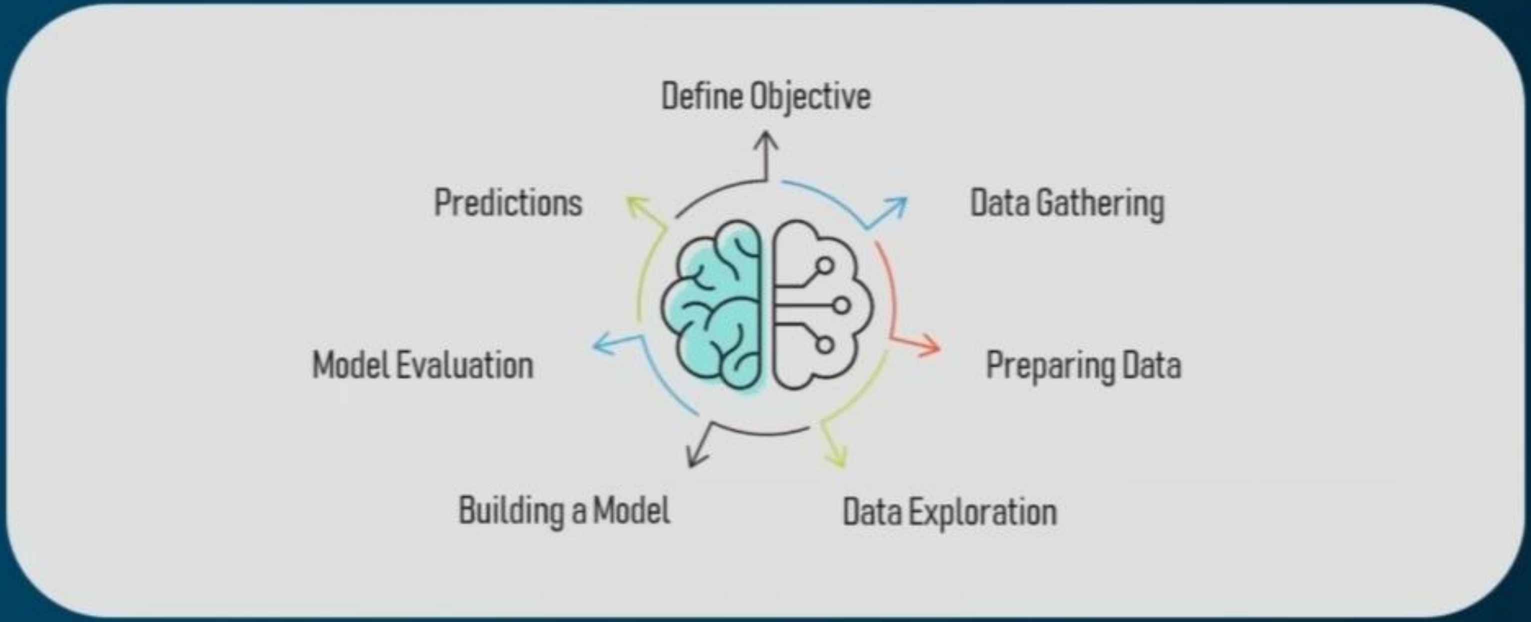
Model: *A model is trained by using a Machine Learning Algorithm.*

Predictor Variable: *It is a feature(s) of the data that can be used to predict the output.*

Response Variable: *It is the feature or the output variable that needs to be predicted by using the predictor variable(s).*

Training Data: *The Machine Learning model is built using the training data.*

Testing Data: *The Machine Learning model is evaluated using the testing data.*



MACHINE LEARNING PROCESS

*The Machine Learning process involves building a **Predictive model** that can be used to find a **solution** for a **Problem Statement.***

edureka!

- What are we trying to predict?
- What are the target features?
- What is the input data?
- What kind of problem are we facing? Binary classification? Clustering?

Step 1: Define the objective of the Problem

To predict the possibility of rain by studying the weather conditions.

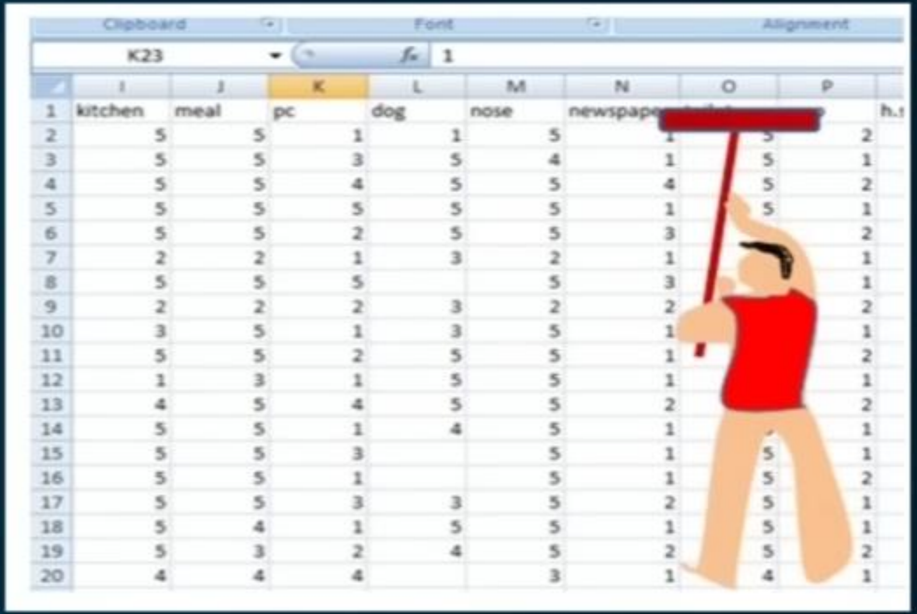
edureka!

Step 2: Data Gathering

Data such as weather conditions, humidity level, temperature, pressure, etc are either collected manually or scarped from the web.

edureka!

- Transform data into desired format
- Data cleaning
  - Missing values
  - Corrupted data
  - Remove unnecessary data



Step 3: Preparing Data

Data Cleaning involves getting rid of inconsistencies in data such as missing values or redundant variables.

edureka!

Step 4: Exploratory Data Analysis

Data Exploration involves understanding the patterns and trends in the data. At this stage all the useful insights are drawn and correlations between the variables are understood.

edureka!

- Machine Learning model is built by using the training data set
- The model is the Machine Learning algorithm that predicts the output by using the data fed to it

Step 5: Building a Machine Learning Model

At this stage a Predictive Model is built by using Machine Learning Algorithms such as Linear Regression, Decision Trees, etc.

edureka!

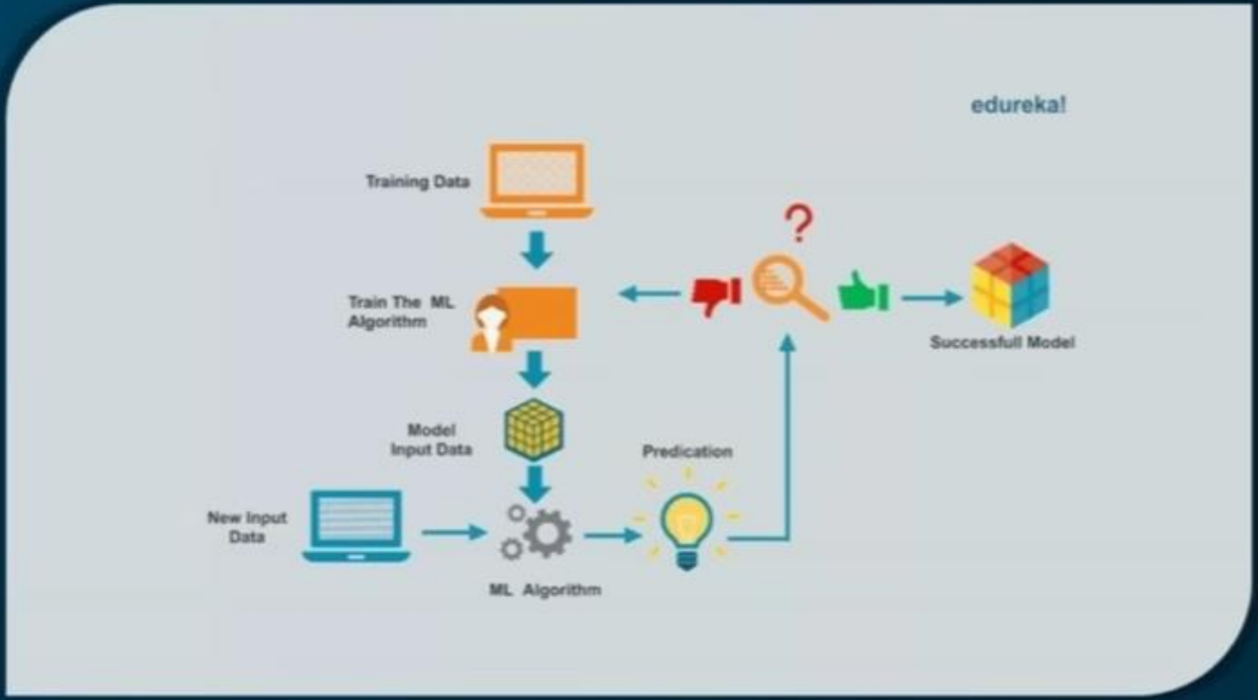
- Machine Learning model is evaluated by using the testing data set
- The accuracy of the model is calculated
- Further improvement in the model are done by using techniques like Parameter tuning

Machine Learning Model

Step 6: Model Evaluation & Optimization

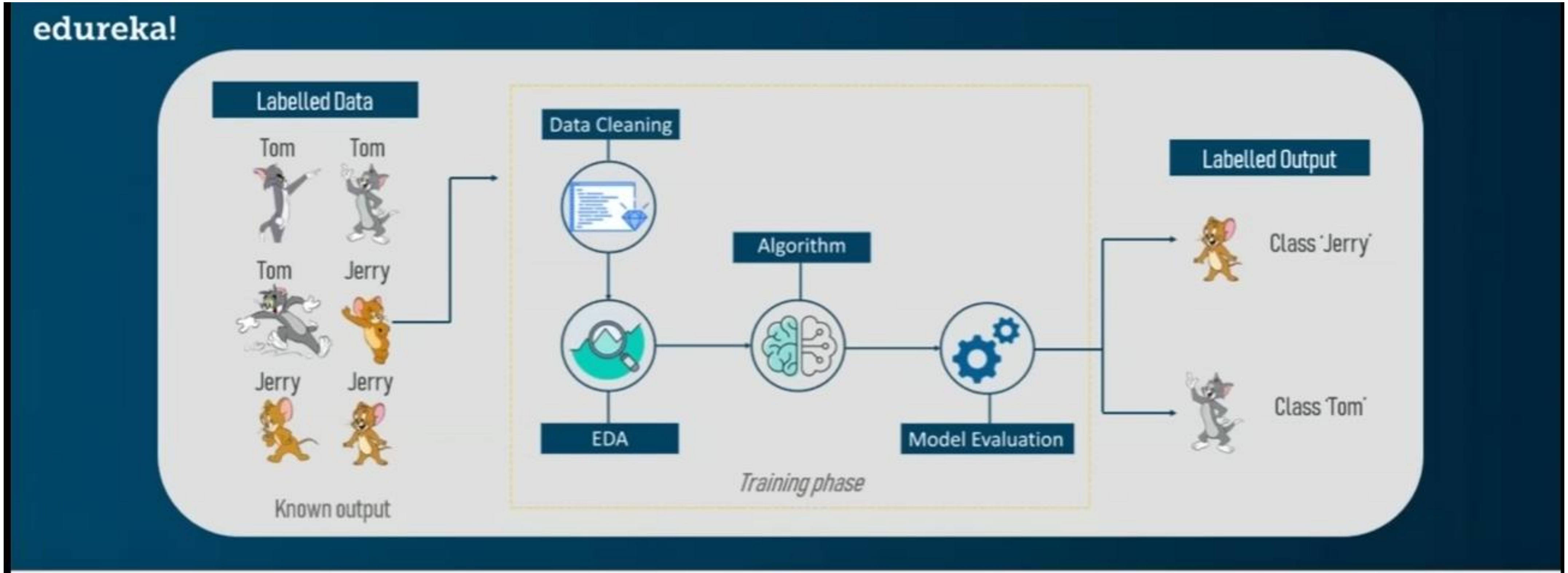
The efficiency of the model is evaluated and any further improvement in the model are implemented.

edureka!



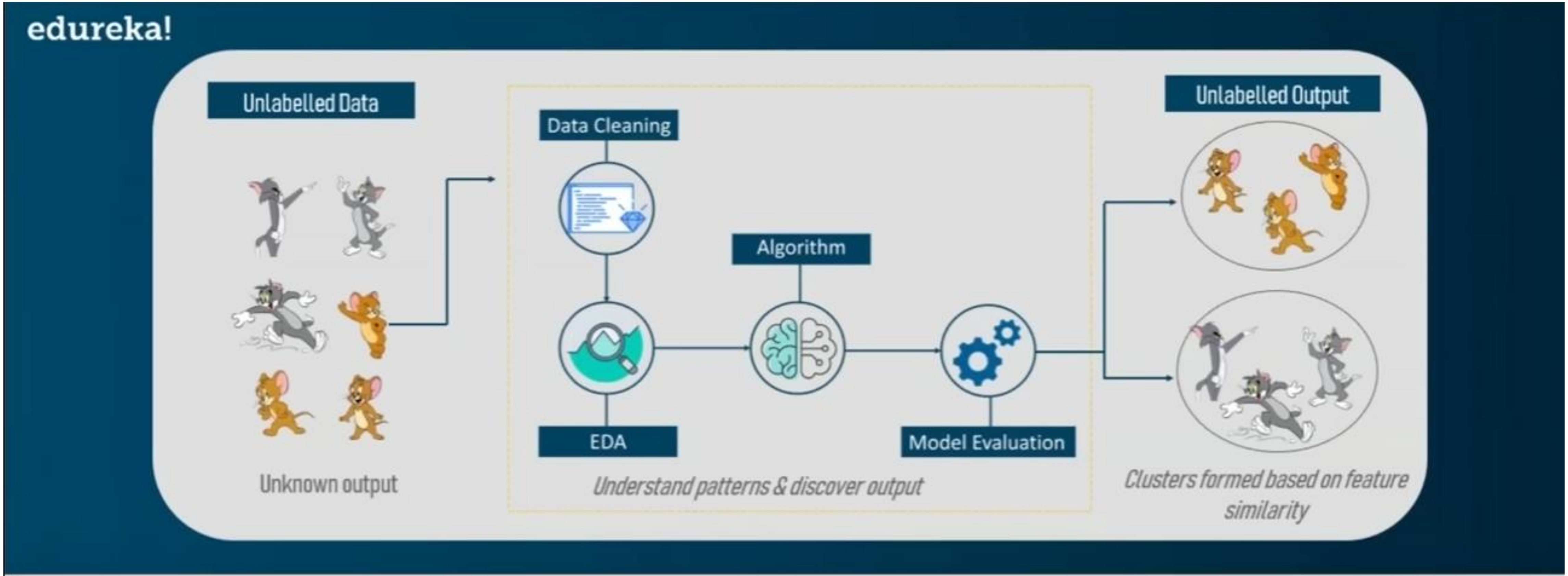
Step 7: Predictions

The final outcome is predicted after performing parameter tuning and improving the accuracy of the model.



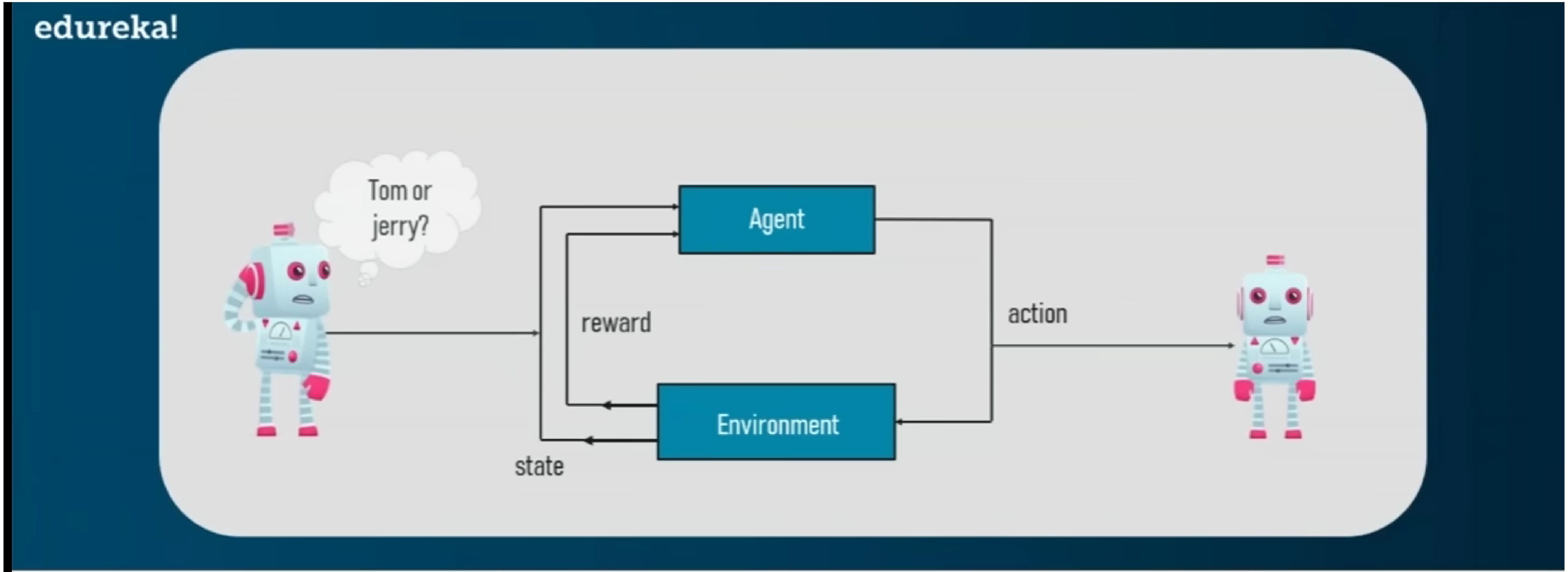
Supervised Learning

Supervised learning is a technique in which we teach or train the machine using data which is well labelled.



Unsupervised Learning

Unsupervised learning is the training of machine using information that is unlabeled and allowing the algorithm to act on that information without guidance.



Reinforcement Learning

Reinforcement Learning is a part of Machine learning where an agent is put in an environment and he learns to behave in this environment by performing certain actions and observing the rewards which it gets from those actions.

edureka!

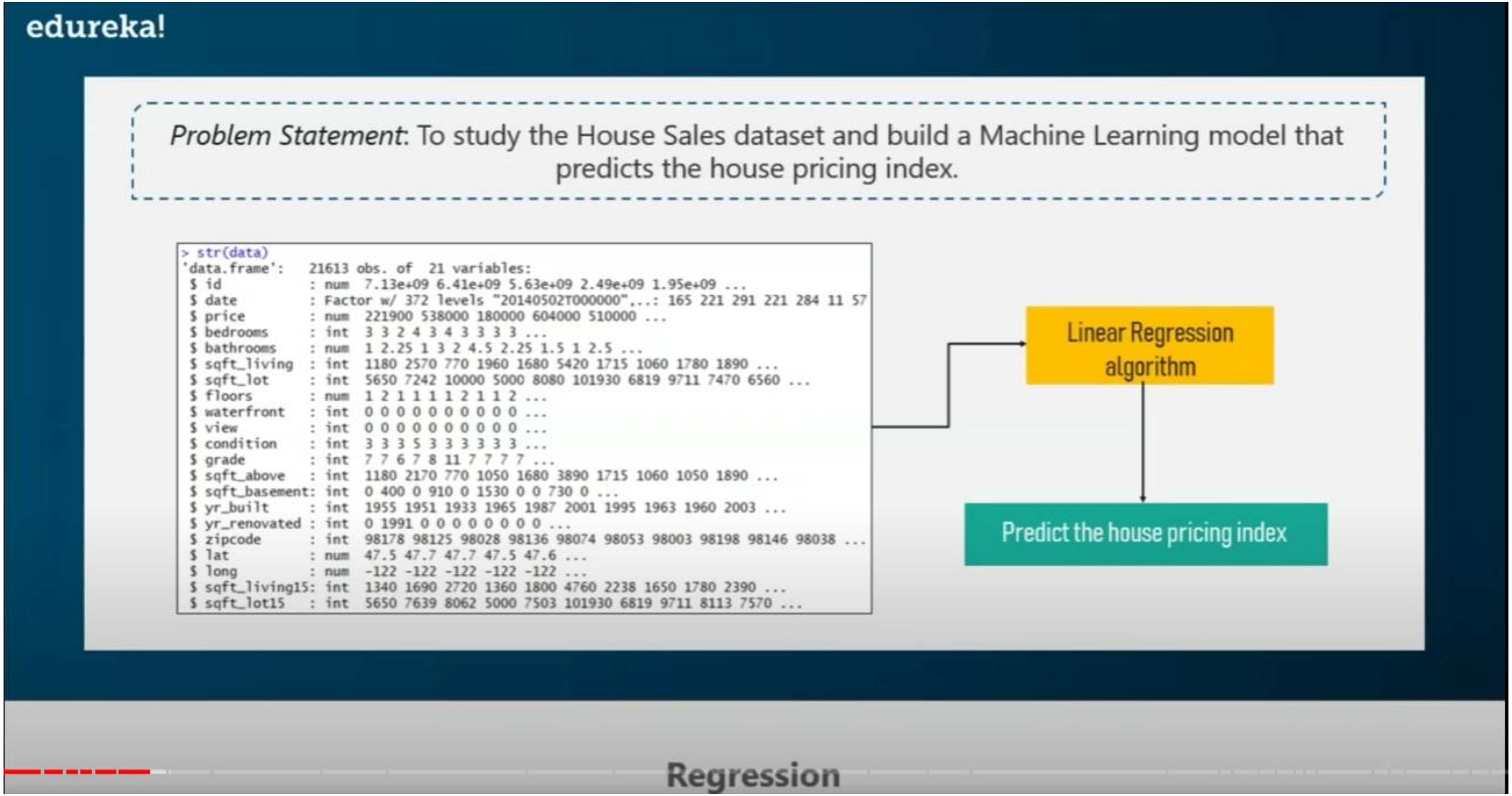
	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	The machine learns by using labelled data	The machine is trained on unlabelled data without any guidance	An agent interacts with its environment by producing actions & discovers errors or rewards
Type of problems	Regression & Classification	Association & Clustering	Reward based
Type of data	Labelled data	Unlabelled data	No pre-defined data
Training	External supervision	No supervision	No supervision
Approach	Map labelled input to known output	Understand patterns and discover output	Follow trail and error method
Popular algorithms	Linear regression, Logistic regression, Support Vector Machine, KNN, etc	K-means, C-means, etc	Q-Learning, SARSA, etc

Supervised vs Unsupervised vs Reinforcement Learning

edureka!

Regression	Classification	Clustering
• Supervised Learning	• Supervised Learning	• Unsupervised Learning
• Output is a continuous quantity	• Output is a categorical quantity	• Assigns data points into clusters
• Main aim is to forecast or predict	• Main aim is to compute the category of the data	• Main aim is to group similar items clusters
• Eg: Predict stock market price	• Eg: Classify emails as spam or non-spam	• Eg: Find all transactions which are fraudulent in nature
• Algorithm: Linear Regression	• Algorithm: Logistic Regression	• Algorithm: K-means

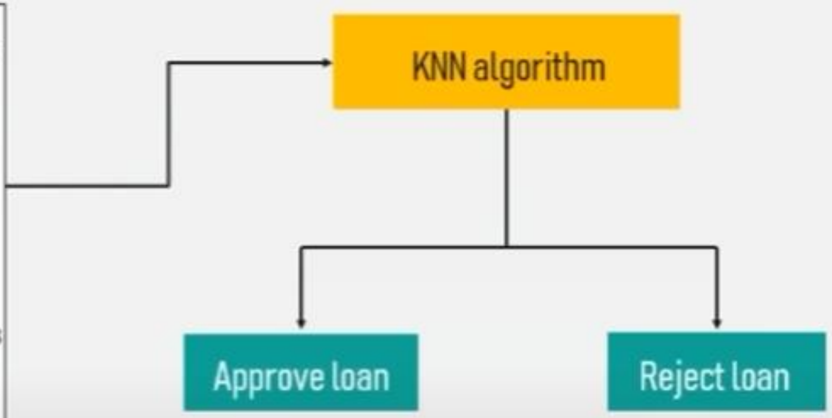
Regression vs Classification vs Clustering


edureka!
Problem Statement: To study the House Sales dataset and build a Machine Learning model that predicts the house pricing index.
> str(data)
'data.frame': 21613 obs. of 21 variables:
$ id : num 7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
$ date : Factor w/ 372 levels "20140502T000000",..: 165 221 291 221 284 11 57
$ price : num 221900 538000 180000 604000 510000 ...
$ bedrooms : int 3 3 2 4 3 4 3 3 3 3 ...
$ bathrooms : num 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
$ sqft_living : int 1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
$ sqft_lot : int 5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
$ floors : num 1 2 1 1 1 1 2 1 1 2 ...
$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...
$ view : int 0 0 0 0 0 0 0 0 0 0 ...
$ condition : int 3 3 3 5 3 3 3 3 3 3 ...
$ grade : int 7 7 6 7 8 11 7 7 7 7 ...
$ sqft_above : int 1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
$ sqft_basement: int 0 400 0 910 0 1530 0 0 730 0 ...
$ yr_built : int 1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
$ yr_renovated : int 0 1991 0 0 0 0 0 0 0 0 ...
$ zipcode : int 98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
$ lat : num 47.5 47.7 47.7 47.5 47.6 ...
$ long : num -122 -122 -122 -122 -122 ...
$ sqft_living15: int 1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
$ sqft_lot15 : int 5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
Linear Regression algorithm
Predict the house pricing index
Regression

edureka!

Problem Statement: Study a bank credit dataset and make a decision about whether to approve the loan of an applicant based on his profile

```
$ Account.Balance                       : int  1 1 2 1 1 1 1 1 4 2 ...
$ Duration.of.Credit..month.            : int  18 9 12 12 12 10 8 6 18 24 ..
$ Payment.Status.of.Previous.Credit: int  4 4 2 4 4 4 4 4 4 2 ...
$ Purpose                               : int  2 0 9 0 0 0 0 0 3 3 ...
$ Credit.Amount                         : int  1049 2799 841 2122 2171 2241
$ Value.Savings.Stocks                  : int  1 1 2 1 1 1 1 1 1 3 ...
$ Length.of.current.employment          : int  2 3 4 3 3 2 4 2 1 1 ...
$ Instalment.per.cent                   : int  4 2 2 3 4 1 1 2 4 1 ...
$ Sex...Marital.Status                  : int  2 3 2 3 3 3 3 3 2 2 ...
$ Guarantors                            : int  1 1 1 1 1 1 1 1 1 1 ...
$ Duration.in.Current.address           : int  4 2 4 2 4 3 4 4 4 4 ...
$ Most.valuable.available.asset         : int  2 1 1 1 2 1 1 1 3 4 ...
$ Age..years.                           : int  21 36 23 39 38 48 39 40 65 23
$ Concurrent.Credits                    : int  3 3 3 3 1 3 3 3 3 3 ...
$ Type.of.apartment                     : int  1 1 1 1 2 1 2 2 2 1 ...
$ No.of.Credits.at.this.Bank            : int  1 2 1 2 2 2 2 1 2 1 ...
$ Occupation                            : int  3 3 2 2 2 2 2 2 1 1 ...
$ No.of.dependents                      : int  1 2 1 2 1 2 1 2 1 1 ...
$ Telephone                             : int  1 1 1 1 1 1 1 1 1 1 ...
$ Foreign.Worker                        : int  1 1 1 2 2 2 2 2 1 1 ...
```

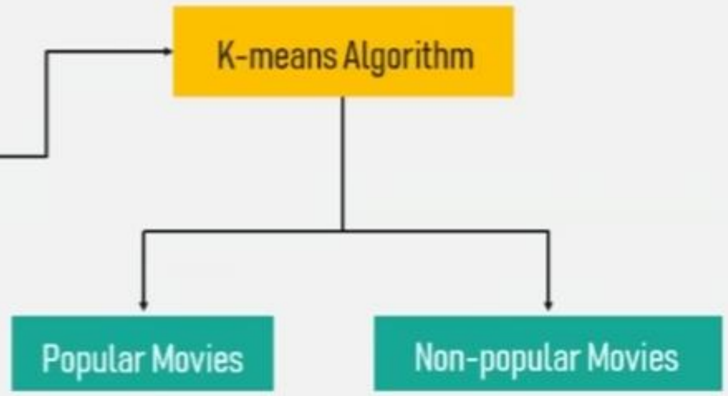


Classification

edureka!

Problem Statement: To cluster a set of movies as either good or average based on their social media out reach

	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes	cast_total_facebook_likes
AvatarÂ	0	855	1000	4834
Pirates of the C...	563	1000	40000	48350
SpectreÂ	0	161	11000	11700
The Dark Knigh...	22000	23000	27000	106759
John CarterÂ	475	530	640	1873
Spider-Man 3Â	0	4000	24000	46055
TangledÂ	15	284	799	2036
Avengers: Age ...	0	19000	26000	92000
Harry Potter an...	282	10000	25000	58753
Batman v Super...	0	2000	15000	24450
Superman Retur...	0	903	18000	29991
Quantum of Sol...	395	393	451	2023
Pirates of the C...	563	1000	40000	48486



Clustering

edureka!

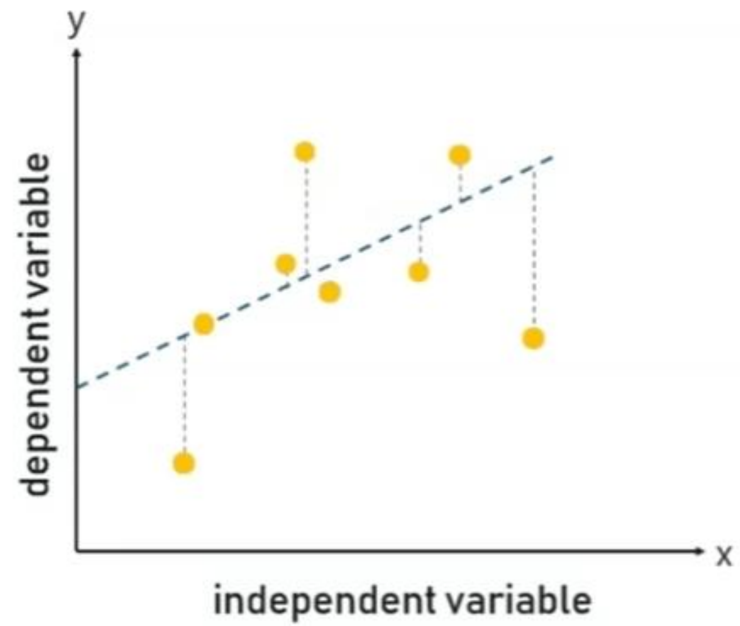
SUPERVISED LEARNING ALGORITHMS

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Naïve Bayes Classifier
- K Nearest Neighbour
- Support Vector Machines

edureka!

LINEAR REGRESSION

Linear Regression is a method to predict dependent variable (Y) based on values of independent variables (X). It can be used for the cases where we want to predict some continuous quantity.



- *Dependent variable (Y):*

The response variable who's value needs to be predicted.

- *Independent variable (X):*

The predictor variable used to predict the response variable.

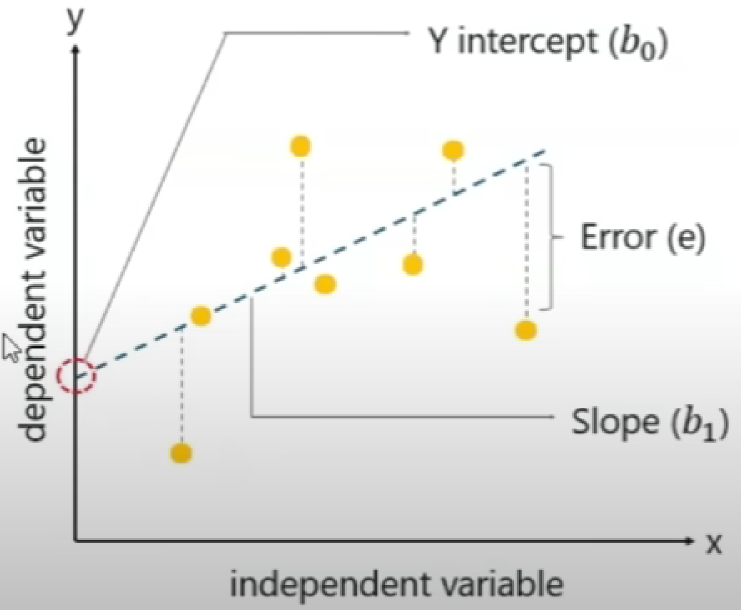
The following equation is used to represent a linear regression model:

$$Y = \beta o + \beta 1X + \epsilon$$

Download

edureka!

LINEAR REGRESSION



$$Y = \beta o + \beta 1X + \epsilon$$

Error

independent variable

Slope

Y intercept

dependent variable

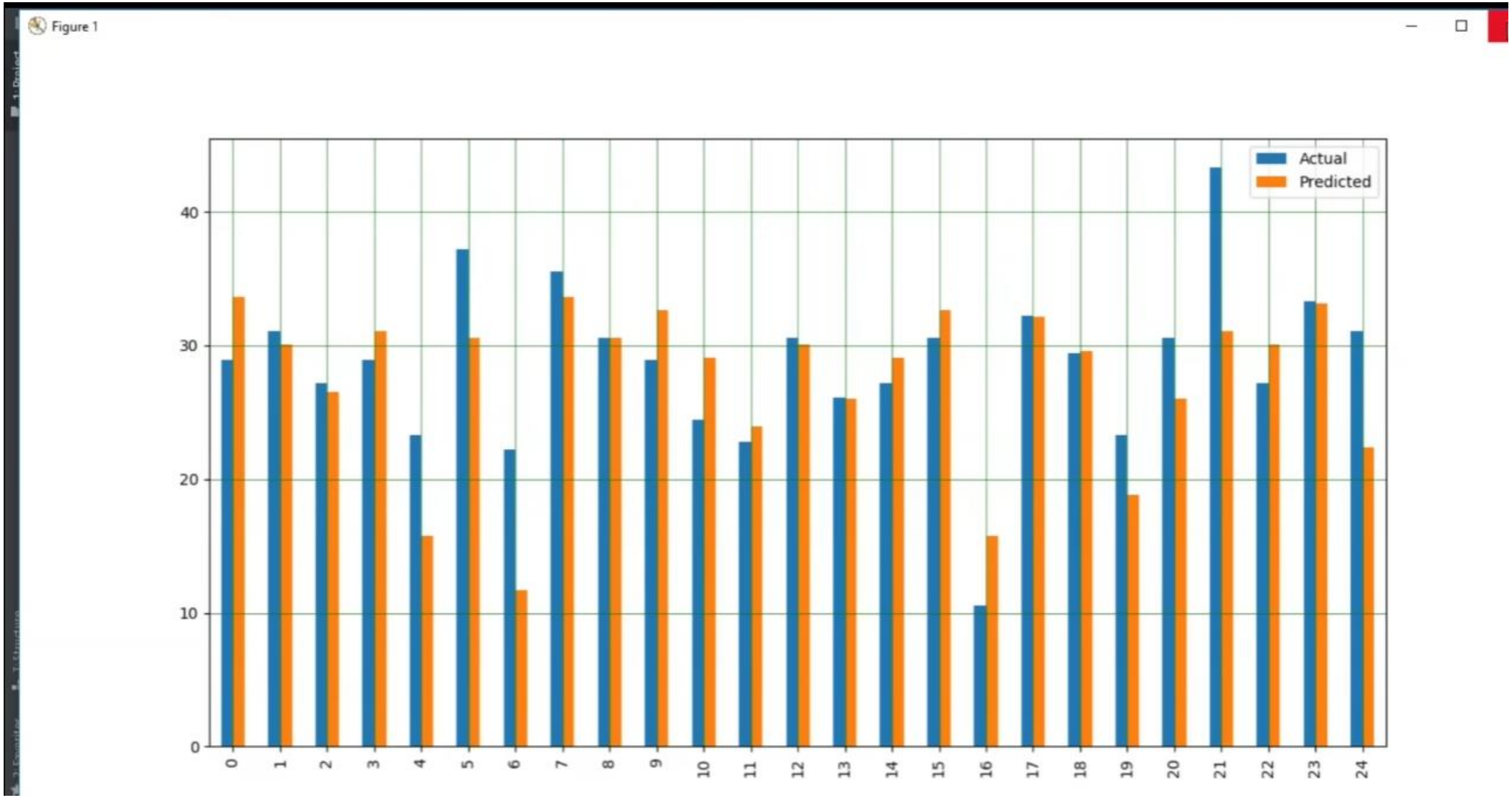
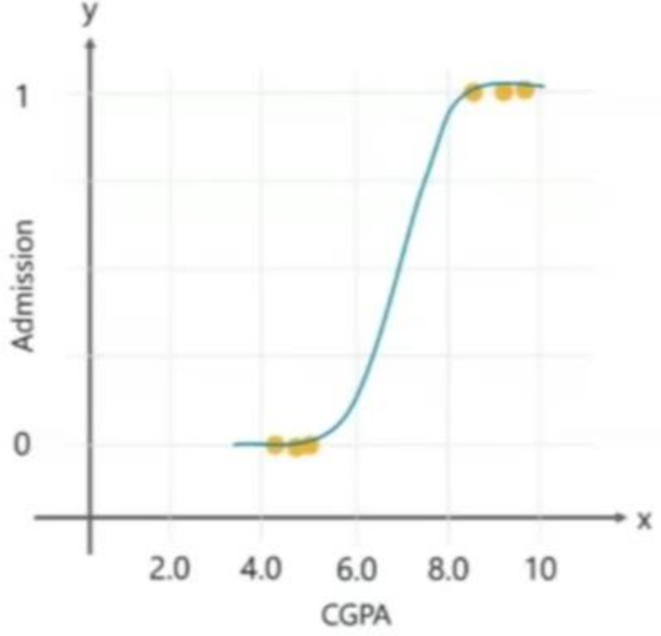


Figure 1
Actual
Predicted
0
10
20
30
40
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

edureka!

LOGISTIC REGRESSION

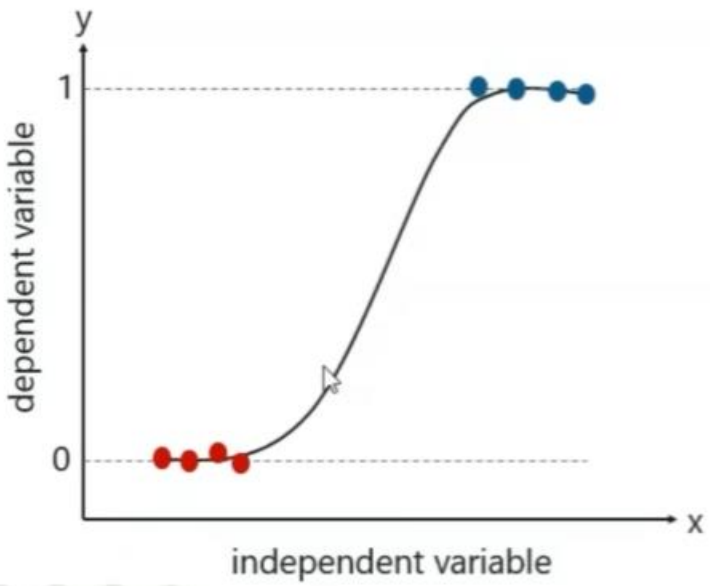
Logistic Regression is a method used to predict a dependent variable, given a set of independent variables, such that the dependent variable is categorical.



edureka!

LOGISTIC REGRESSION

Logistic Regression is a method used to predict a dependent variable, given a set of independent variables, such that the dependent variable is categorical.



Linear Regression equation: **Y = βo + β1X + ∈**

Represent a relationship between p(X)=Pr(Y=1|X) and X?

- Take the exponent of the equation, since the exponential of any value is a positive number
- Secondly, a number divided by itself + 1 will always be less than 1

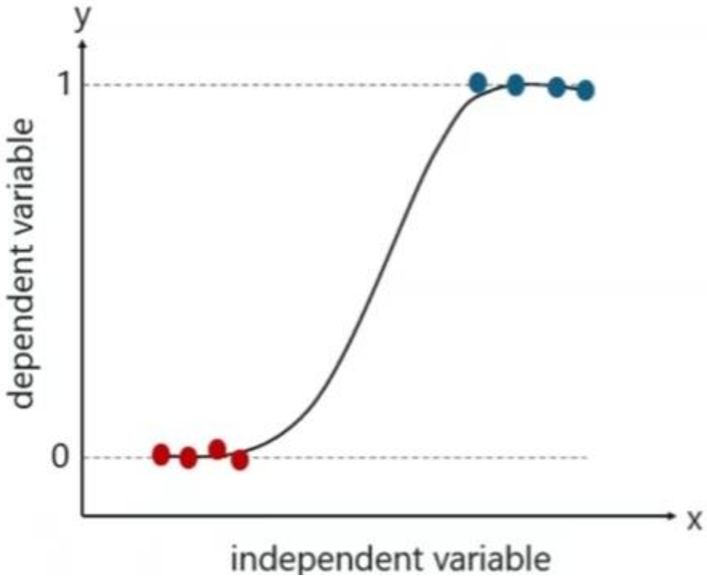
Hence, the formula:

$$P(X) = \frac{e^{(\beta_0+\beta_1 x)}}{e^{(\beta_0+\beta x)} + 1}$$

edureka!

LOGISTIC REGRESSION

Logistic Regression is a method used to predict a dependent variable, given a set of independent variables, such that the dependent variable is categorical.



$$P(X) = \frac{e^{(\beta_0+\beta_1 x)}}{e^{(\beta_0+\beta x)}+1}$$

$$\Longrightarrow \; p(e^{(\beta_0+\beta x)}+1) = e^{(\beta_0+\beta_1 x)}$$

$$\Longrightarrow \; p.\,e^{(\beta_0+\beta x)} + p = e^{(\beta_0+\beta_1 x)}$$

$$\Longrightarrow \; p = e^{(\beta_0+\beta_1 x)} - p.\,e^{(\beta_0+\beta x)}$$

$$\Longrightarrow \; p = e^{(\beta_0+\beta_1 x)}\,(1-p)$$

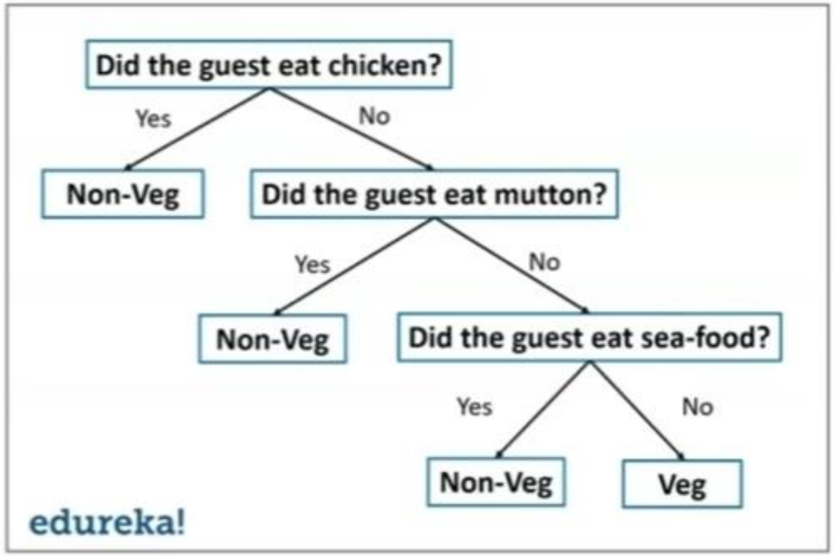
$$\Longrightarrow \; \frac{p}{(1-p)} = e^{(\beta_0+\beta_1 x)}$$

$$\Longrightarrow \; \ln[\frac{p}{(1-p)}] = (\beta_0 + \beta_1 x)$$

edureka!

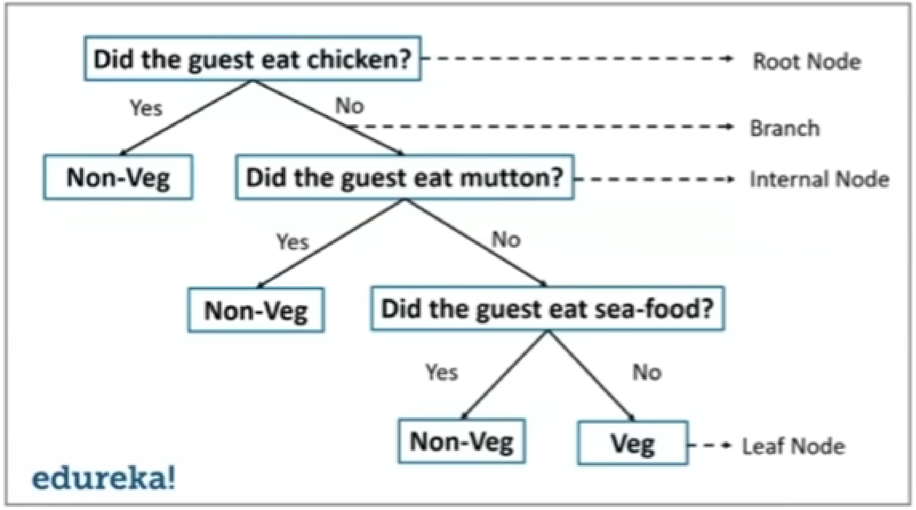
DECISION TREE

A Decision Tree is a Supervised Machine Learning algorithm which looks like an inverted tree, wherein each node represents a **predictor variable** (feature), the link between the nodes represents a **Decision** and each leaf node represents an **outcome** (response variable).



edureka!

DECISION TREE



Root Node: The root node is the starting point of a tree. At this point, the first split is performed.

Internal Nodes: Each internal node represents a decision point (predictor variable) that eventually leads to the prediction of the outcome.

Leaf/ Terminal Nodes: Leaf nodes represent the final class of the outcome and therefore they're also called terminating nodes.

Branches: Branches are connections between nodes, they're represented as arrows. Each branch represents a response such as yes or no.

edureka!

DECISION TREE

Step 1: Select **Best Attribute (A)**

Step 2: Assign A as a decision variable for the root node.

Step 3: For each value of A, build a descendant of the node

Step 4: Assign classification labels to the leaf node.

Step 5: If data is correctly classified: Stop.

Step 6: Else: Iterate over the tree.

edureka!

INFORMATION GAIN & ENTROPY

Problem Statement: To study the data set and create a Decision Tree that classifies the speed of a car as either slow or fast,

Road type	Obstruction	Speed limit	Speed
steep	yes	yes	slow
steep	no	yes	slow
flat	yes	no	fast
steep	no	no	fast

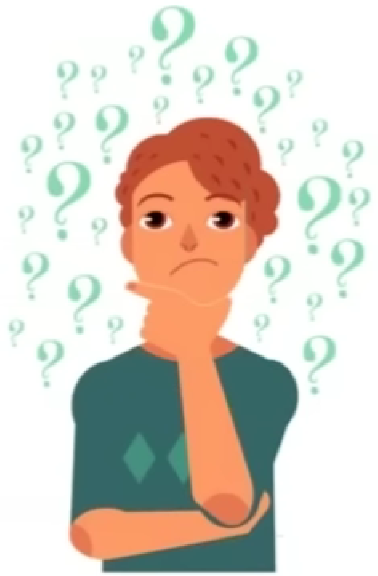
edureka!

INFORMATION GAIN & ENTROPY

Step 1: Select **Best Attribute (A)**

Q. How do you know which variable best separates the data?

Ans: The variable with the highest Information Gain best divides the data into the desired output classes.



Calculate the following measures:

1. Entropy
2. Information Gain (IG)

edureka!

INFORMATION GAIN & ENTROPY

Entropy

Entropy measures the impurity or uncertainty present in the data.

Information Gain (IG)

IG indicates how much "information" a particular feature/ variable gives us about the final outcome.

INFORMATION GAIN & ENTROPY

Calculating IG of parent node (Speed of car)

Find out the fraction of the two classes (slow & fast) present in the parent node:

- P(slow) -> fraction of 'slow' outcomes in the parent node
- P(fast) -> fraction of 'fast' outcomes in the parent node

The formula to calculate P(slow) is:

p(slow) = no. of 'slow' outcomes in the parent node / total number of outcomes

Road type	Obstruction	Speed limit	Speed
steep	yes	yes	slow
steep	no	yes	slow
flat	yes	no	fast
steep	no	no	fast

INFORMATION GAIN & ENTROPY

Calculating IG of parent node (Speed of car)

Similarly for p(fast),

p(fast) = no. of 'fast' outcomes in the parent node / total number of outcomes

$$P_{\text{fast}} = \frac{2}{4} = 0.5$$

Road type	Obstruction	Speed limit	Speed
steep	yes	yes	slow
steep	no	yes	slow
flat	yes	no	fast
steep	no	no	fast

Therefore, the entropy of the parent node is:

$$Entropy_{\text{parent}} = -\Sigma p_{slow} log_2(p_{slow}) + p_{fast} log_2(p_{fast})$$

Entropy(parent) = – {0.5 log2(0.5) + 0.5 log2(0.5)} = – {-0.5 + (-0.5)} = 1

INFORMATION GAIN & ENTROPY

Calculating IG of child node (Road Type)

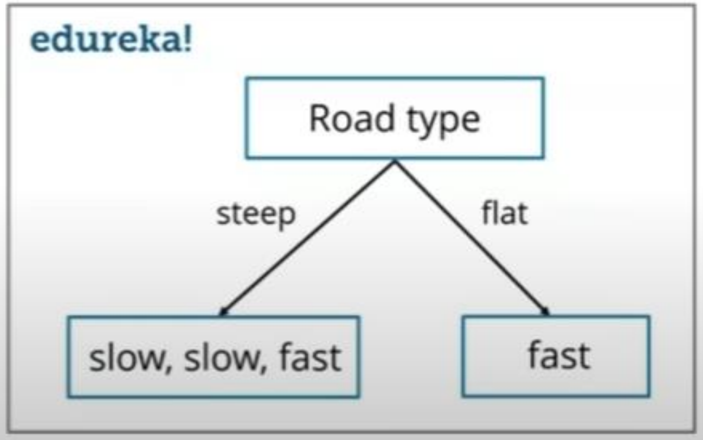
Entropy of right side child node (fast):

Entropy of left side child node (slow, slow, fast):

P(slow) = 2/3 = 0.667
P(fast) = 1/3 = 0.334

Therefore, the entropy is:

Entropy(left child node) = – {0.667 log2(0.667) + 0.334 log2(0.334)}
= – {-0.38 + (-0.52)} = 0.9



INFORMATION GAIN & ENTROPY

Calculating IG of child node (Road Type)

*Information Gain = entropy(parent) – [weighted average] * entropy(children)*

INFORMATION GAIN & ENTROPY

Calculating IG of child node (Road Type)

calculate the Entropy(children) with weighted average:

- Total number of outcomes in parent node: 4
- Total number of outcomes in left child node: 3
- Total number of outcomes in right child node: 1

*[Weighted avg]Entropy(children) = (no. of outcomes in left child node) / (total no. of outcomes in parent node) * (entropy of left node) + (no. of outcomes in right child node)/ (total no. of outcomes in parent node) * (entropy of right node)*

Entropy(children) with weighted avg. is = 0.675

INFORMATION GAIN & ENTROPY

Calculating IG of child node (Road Type)

calculate the Entropy(children) with weighted average:

- Total number of outcomes in parent node: 4
- Total number of outcomes in left child node: 3
- Total number of outcomes in right child node: 1

*[Weighted avg]Entropy(children) = (no. of outcomes in left child node) / (total no. of outcomes in parent node) * (entropy of left node) + (no. of outcomes in right child node)/ (total no. of outcomes in parent node) * (entropy of right node)*

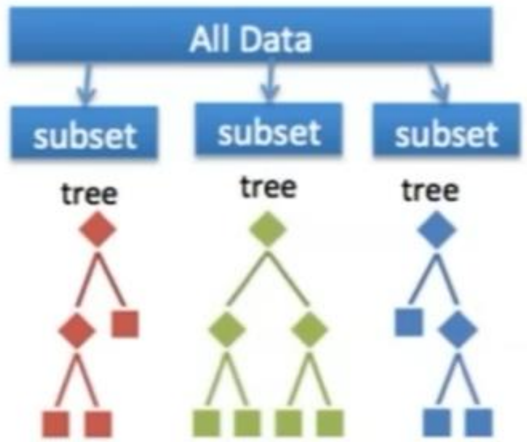
Entropy(children) with weighted avg. is = 0.675

RANDOM FOREST

Random forest builds multiple decision trees (called the forest) and glues them together to get a more accurate and stable prediction.

Why Random Forest?

- More Accuracy
- Avoid Overfitting
- Bagging



RANDOM FOREST

We're going to use this data set to create a Random Forest that predicts if a person has heart disease or not.

Blood Flow	Blocked Arteries	Chest Pain	Weight	Heart Disease
Abnormal	No	No	130	No
Normal	Yes	Yes	195	Yes
Normal	No	Yes	218	No
Abnormal	Yes	Yes	180	Yes

CREATING A RANDOM FOREST

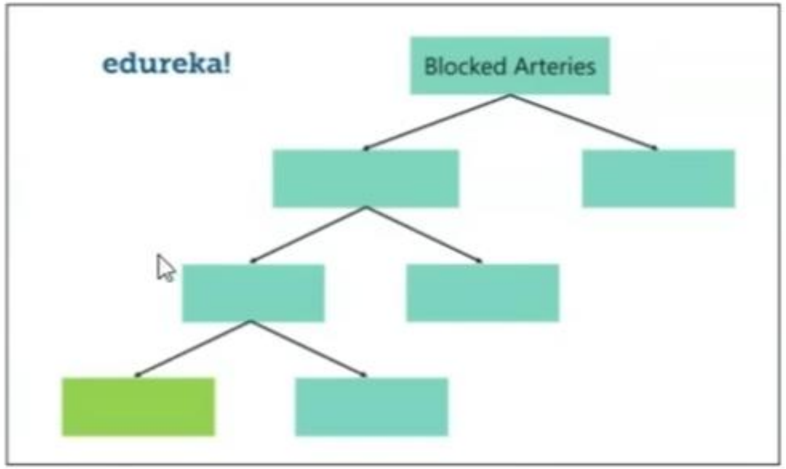
Step 1: Create a Bootstrapped Data Set

Blood Flow	Blocked Arteries	Chest Pain	Weight	Heart Disease
Normal	Yes	Yes	195	Yes
Abnormal	No	No	130	No
Abnormal	Yes	Yes	180	Yes
Abnormal	Yes	Yes	180	Yes

CREATING A RANDOM FOREST

Step 2: Creating Decision Trees

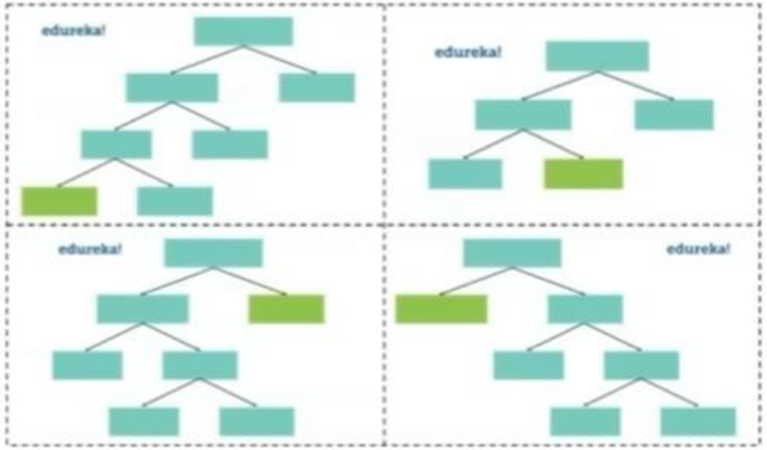
- Build a Decision Tree by using the bootstrapped data set
- Begin at the root node & choose the best attribute to split the data set
- Repeat the same process for each of the upcoming branch nodes



CREATING A RANDOM FOREST

Step 3: Go back to Step 1 and Repeat

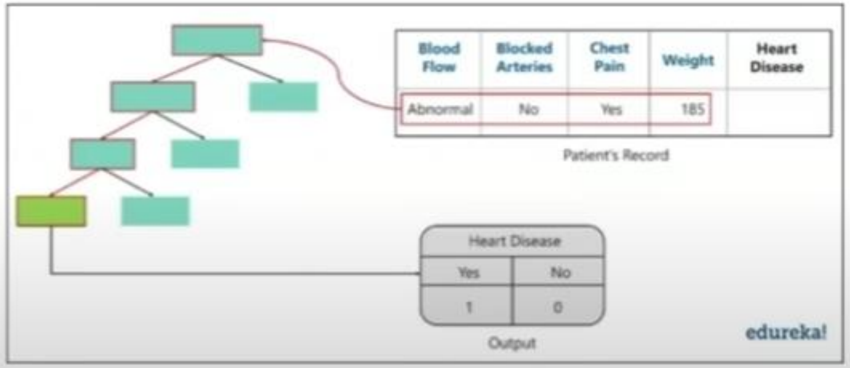
- Each Decision Tree predicts the output class based on the respective predictor variables used in that tree.
- Go back to step 1, create a new bootstrapped data set and then build a Decision Tree by considering only a subset of variables at each step.
- This iteration is performed 100's of times, creating multiple decision trees



CREATING A RANDOM FOREST

Step 4: Predicting the outcome of a new data point

- To predict whether a new patient has heart disease or not, run the new data down the decision trees
- After running the data down all the trees in the Random Forest, we check which class got the majority votes.
- In our case, the class 'Yes' received the most number of votes, hence it's clear that the new patient has heart disease.



CREATING A RANDOM FOREST

Step 5: Evaluate the Model

- In a real-world problem, about 1/3rd of the original data set is not included in the bootstrapped data set.
- This sample data set that does not include in the bootstrapped data set is known as the Out-Of-Bag (OOB) data set.
- we can measure the accuracy of a Random Forest by the proportion of OOB samples that are correctly classified.

Blood Flow	Blocked Arteries	Chest Pain	Weight	Heart Disease
Normal	No	Yes	218	No

NAÏVE BAYES

- Naïve Bayes is based on the Bayes Theorem that is used to solve classification problems by following a probabilistic approach.
- It is based on the idea that the predictor variables in a Machine Learning model are independent of each other.

- P(A|B): Conditional probability of event A occurring, given the event B
- P(A): Probability of event A occurring
- P(B): Probability of event B occurring
- P(B|A): Conditional probability of event B occurring, given the event A

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

NAÏVE BAYES

Type	Swim	Wings	Green	Sharp teeth
Cat	450/500	0	0	500/500
Parrot	50/500	500/500	400/500	0
Turtle	500/500	0	100/500	50/500

The class of type cats shows that:

- Out of 500, 450 (90%) cats can swim
- 0 number of cats have wings
- 0 number of cats are of Green color
- All 500 cats have sharp teeth

The class of type Parrot shows that:

- 50 (10%) parrots hold true value for swim
- All 500 parrots have wings
- 400 (80%) parrots are green in color
- No parrots have sharp teeth

The class of type Turtle shows that:

- All 500 turtles can swim
- 0 number of turtles have wings
- 100 (20%) turtles are green in color
- 50 (10%) turtles have sharp teeth

NAÏVE BAYES

To predict whether the animal is a Cat, Parrot or a Turtle based on the defined predictor variables (swim, wings, green, sharp teeth).

	Swim	Wings	Green	Sharp Teeth
Observation	True	False	True	False

To solve this, we will use the Naive Bayes approach:

P(H|Multiple Evidences) = P(C1| H)* P(C2|H) ......*P(Cn|H) * P(H) / P(Multiple Evidences)

NAÏVE BAYES

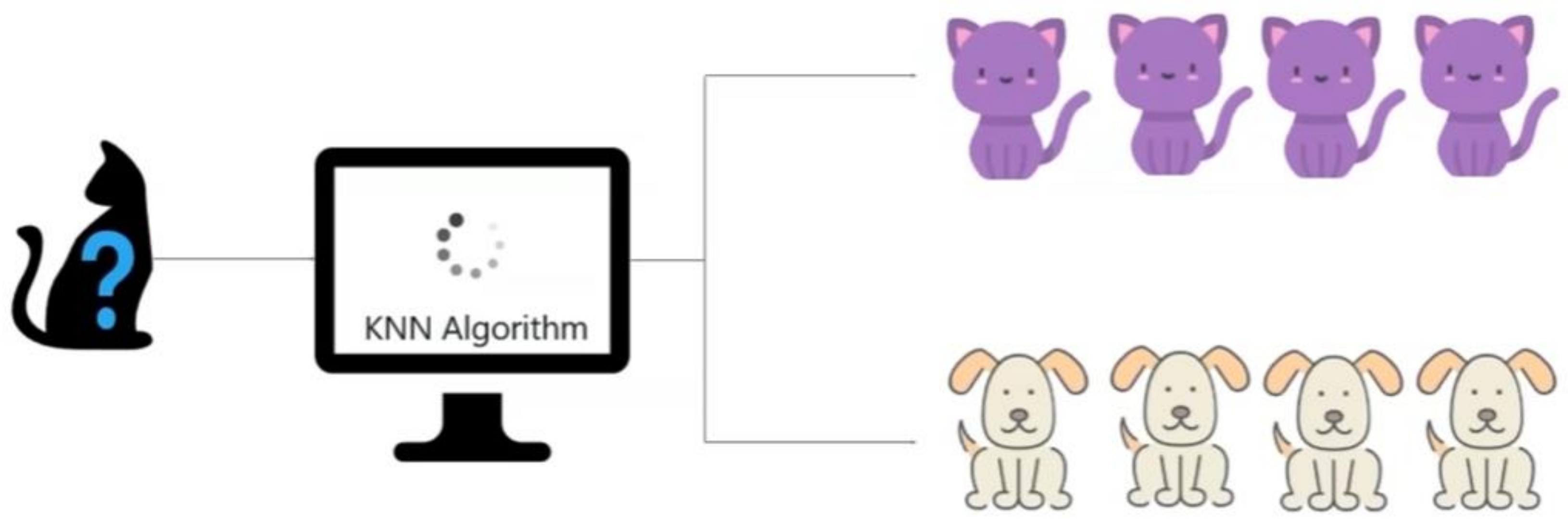
In the observation, the variables Swim and Green are true and the outcome can be any one of the animals (Cat, Parrot, Turtle).

P(H|Multiple Evidences) = P(C1| H)* P(C2|H) ......*P(Cn|H) * P(H) / P(Multiple Evidences)

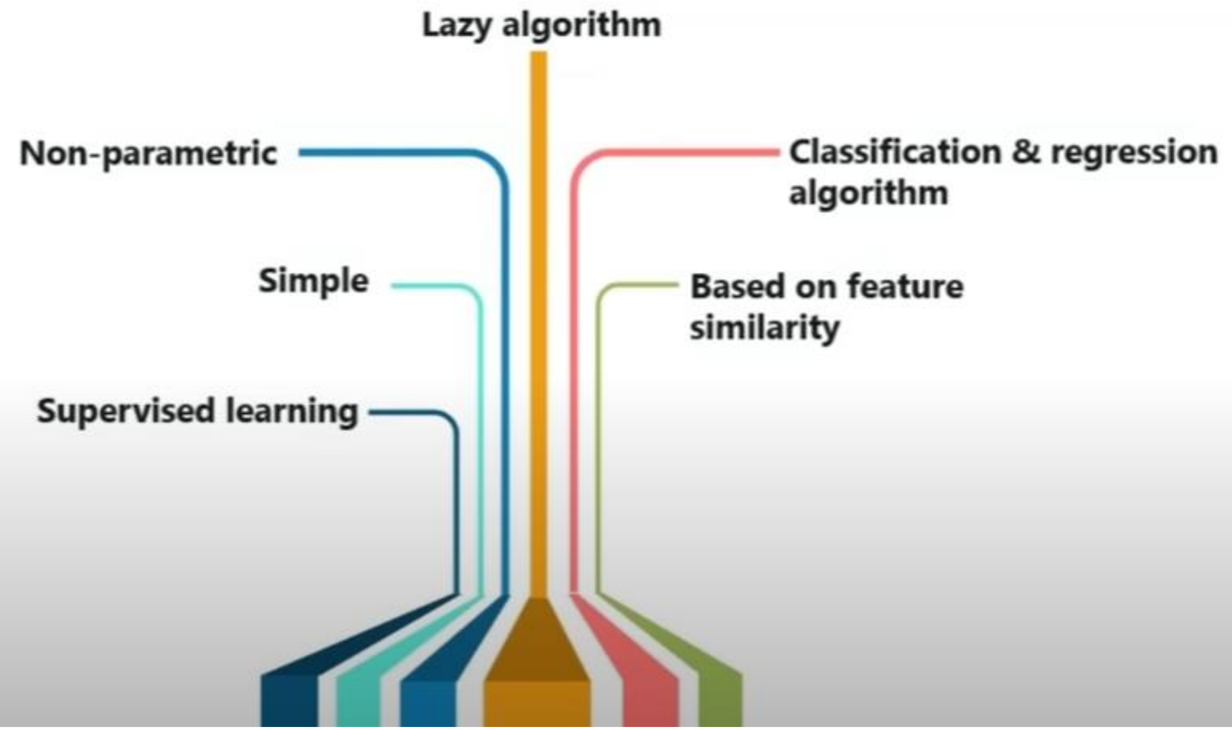
To check if the animal is a Turtle:
*P(Turtle| Swim, Green) = P(Swim|Turtle) * P(Green|Turtle) * P(Turtle) / P(Swim, Green)*
= 1 * 0.2 * 0.333 / P(Swim, Green)
= **0.0666/ P(Swim, Green)**

K NEAREST NEIGHBOUR (KNN)

K Nearest Neighbour is a Supervised Learning algorithm that classifies a new data point into the target class, depending on the features of it's neighbouring data points.

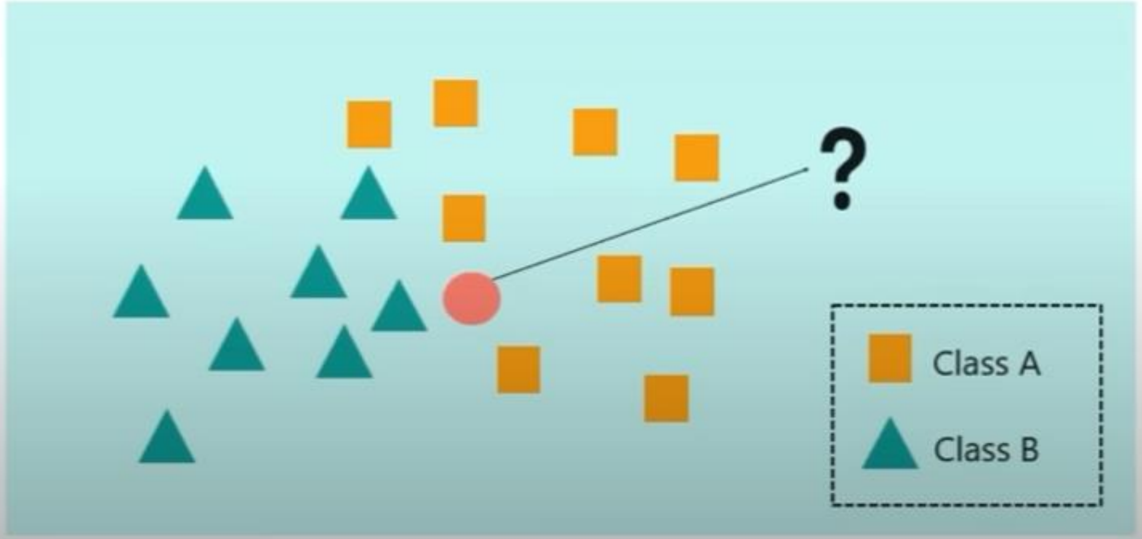


K NEAREST NEIGHBOUR (KNN)



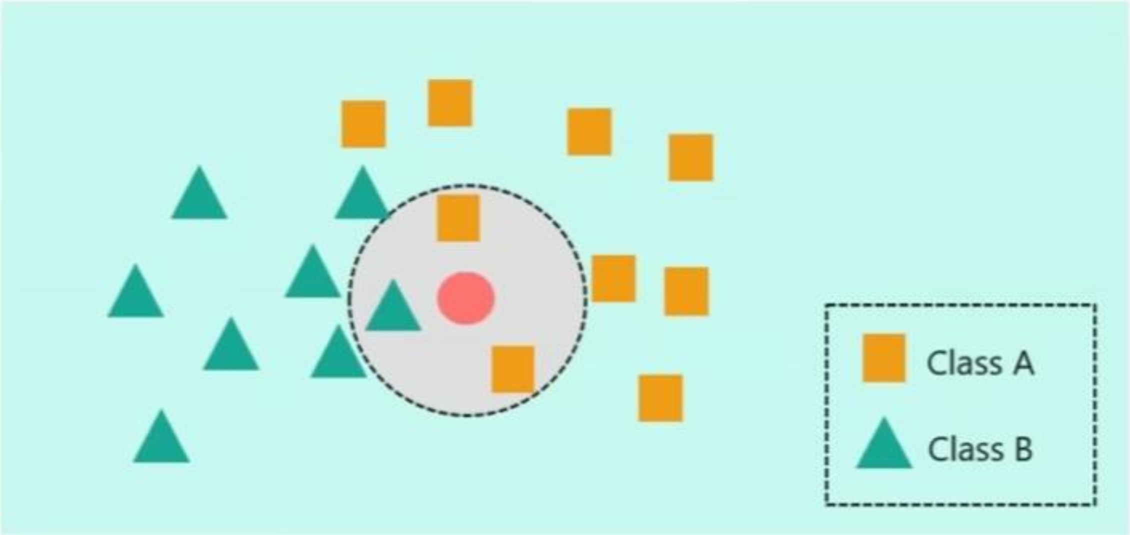
K NEAREST NEIGHBOUR (KNN)

Problem Statement: Assign the new data point into one of the two clusters



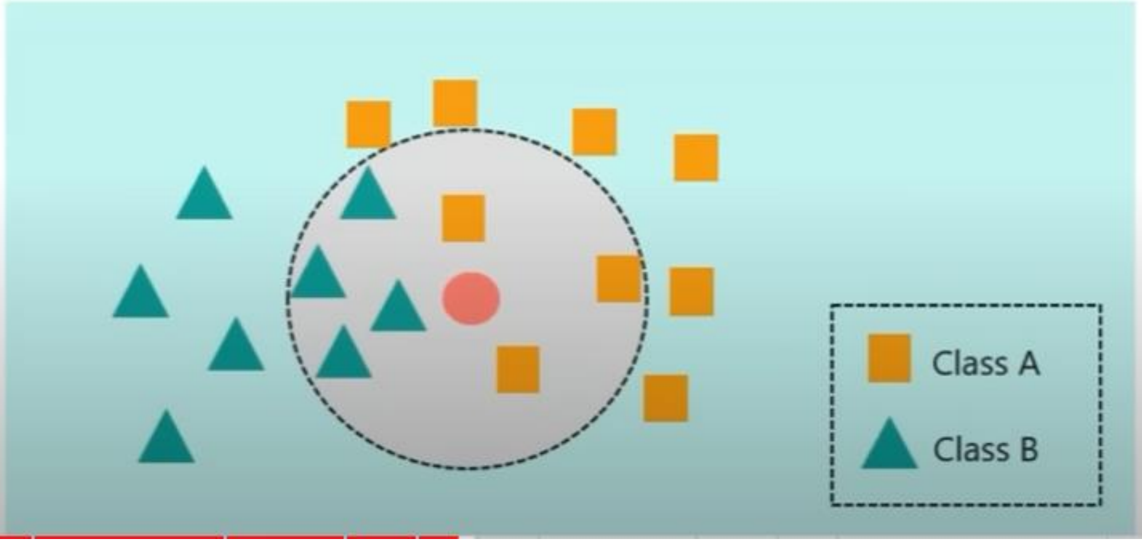
K NEAREST NEIGHBOUR (KNN)

Choose the value of 'k', here k=3 and find the 3 nearest neighbours

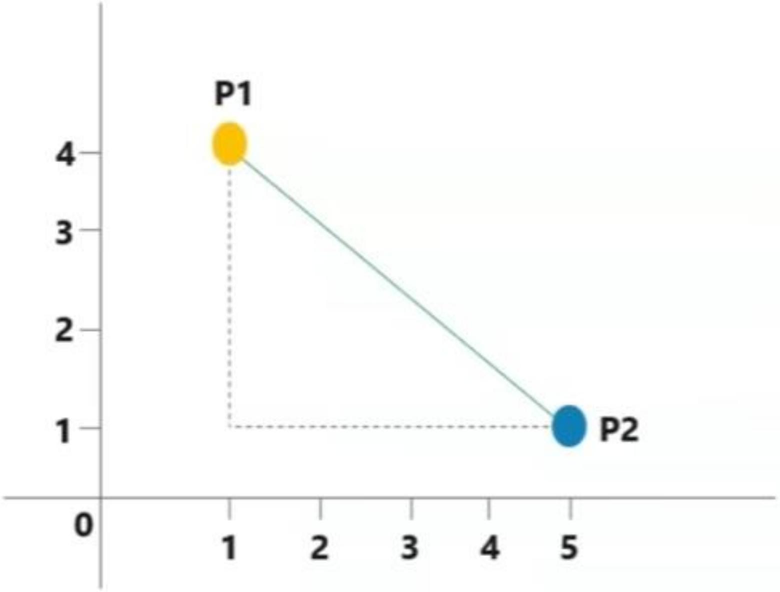


K NEAREST NEIGHBOUR (KNN)

Here k=7, find the 7 nearest neighbours



EUCLIDEAN DISTANCE



Calculations

Point P1 = (1,1)

Point P2 = (5,4)

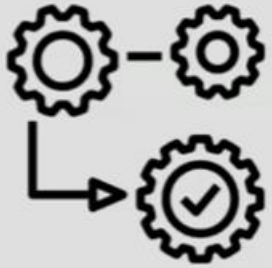
Euclidian distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.

Supervised machine learning algorithm

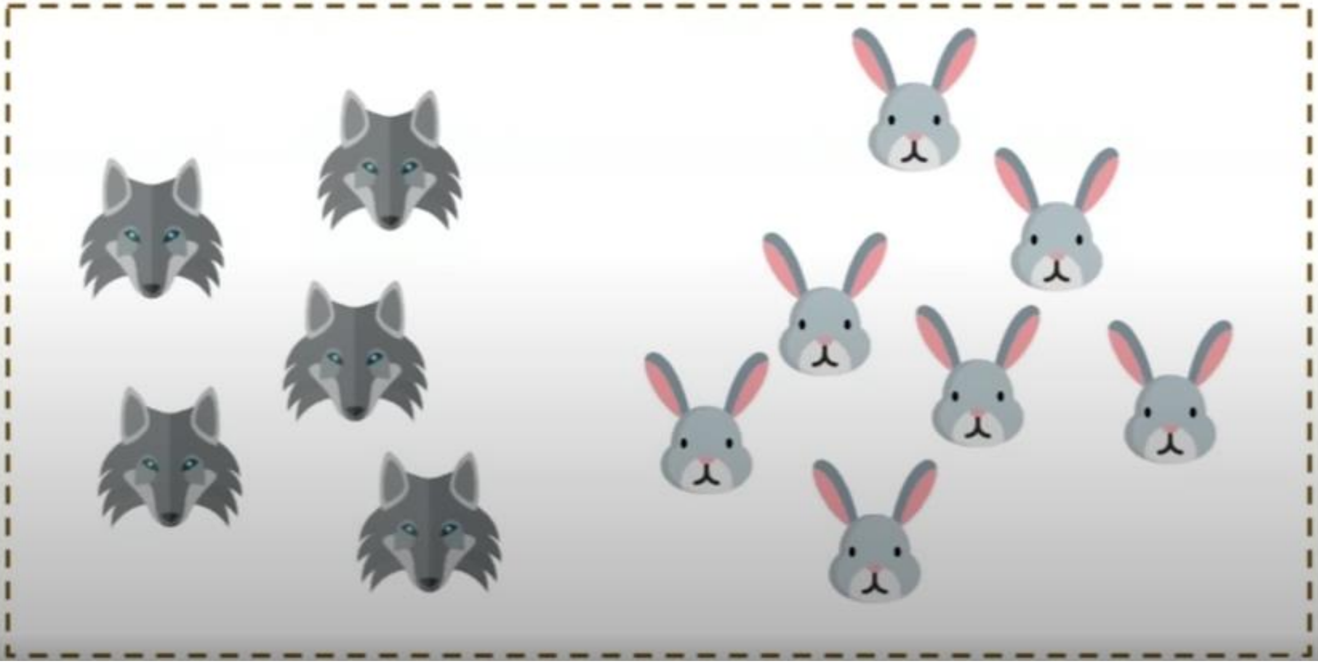
Classification & Regression algorithm



SVM kernel functions

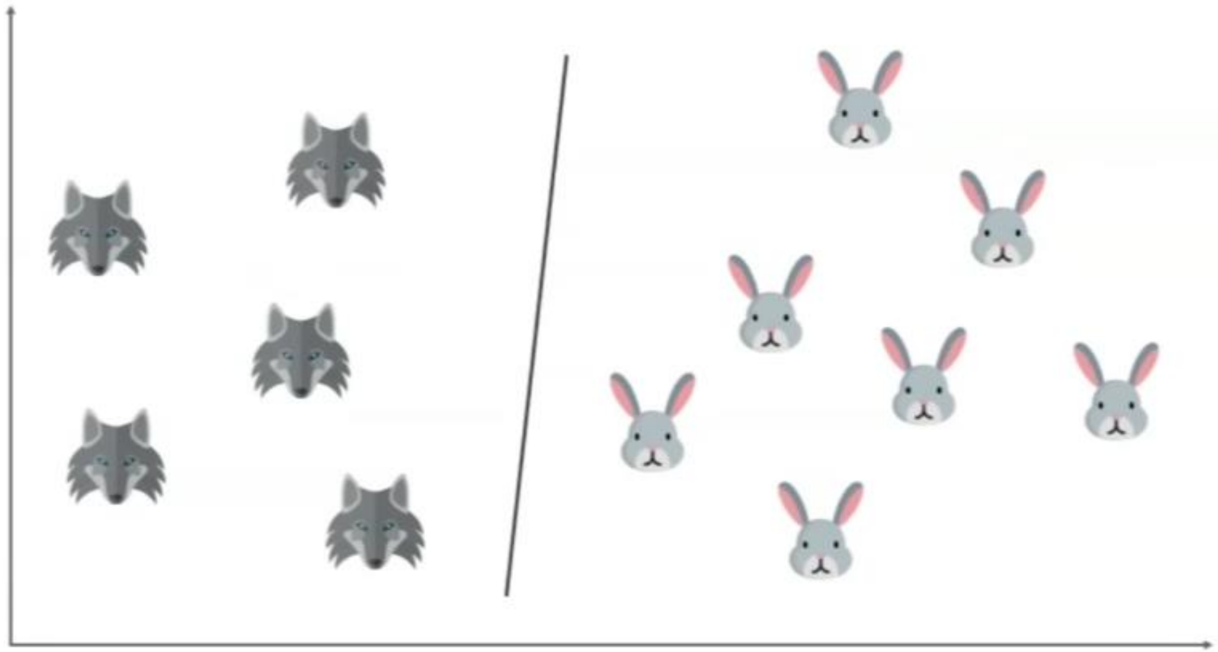
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.



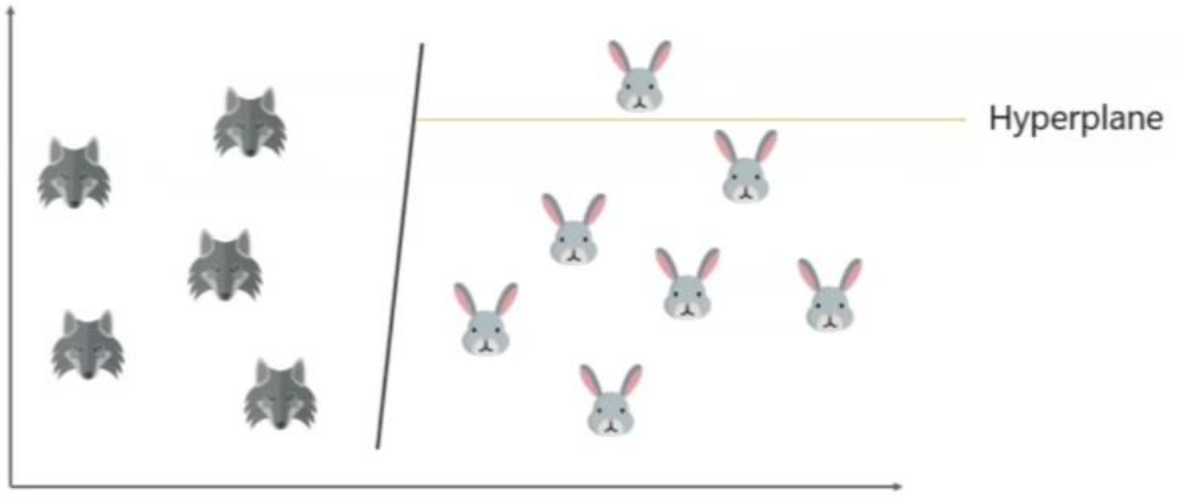
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.



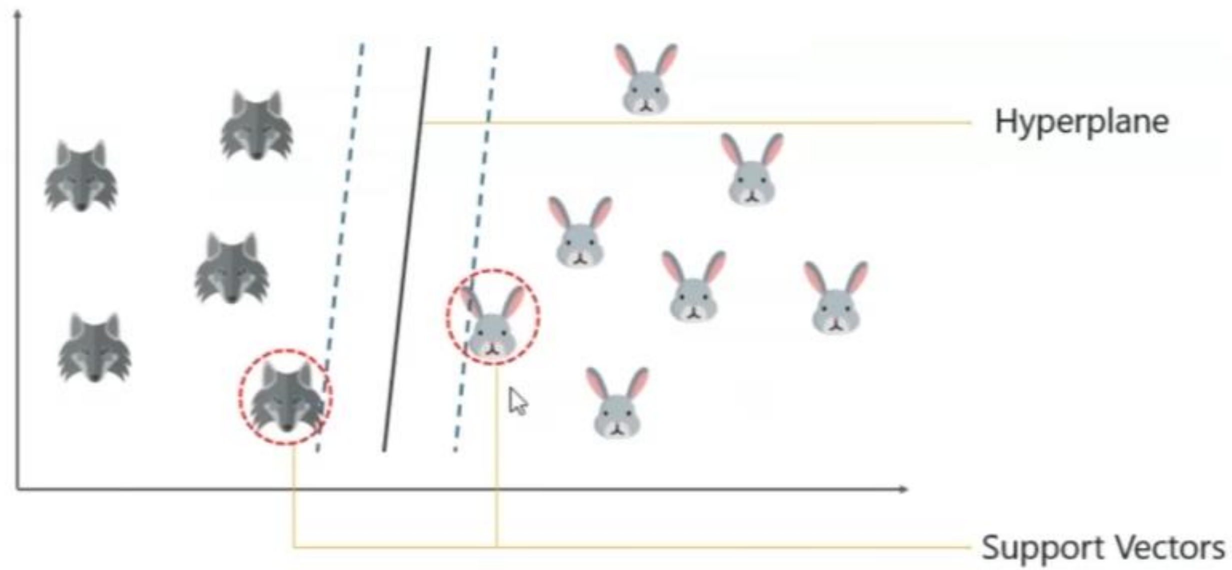
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.



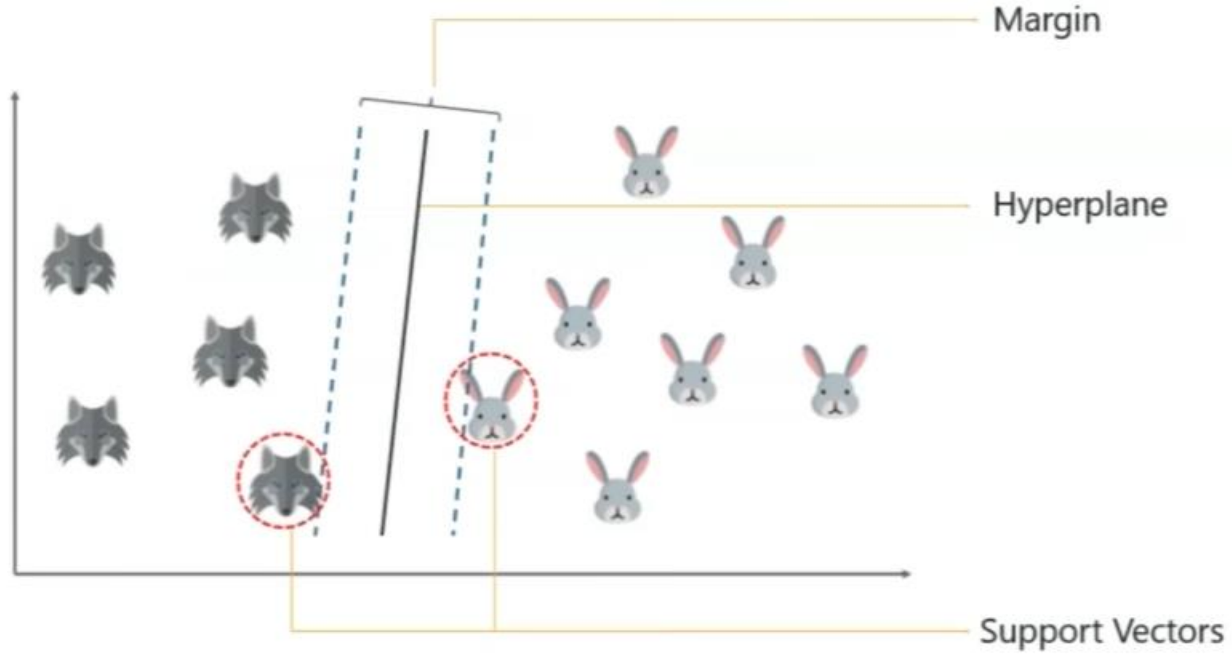
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.



SUPPORT VECTOR MACHINE (SVM)

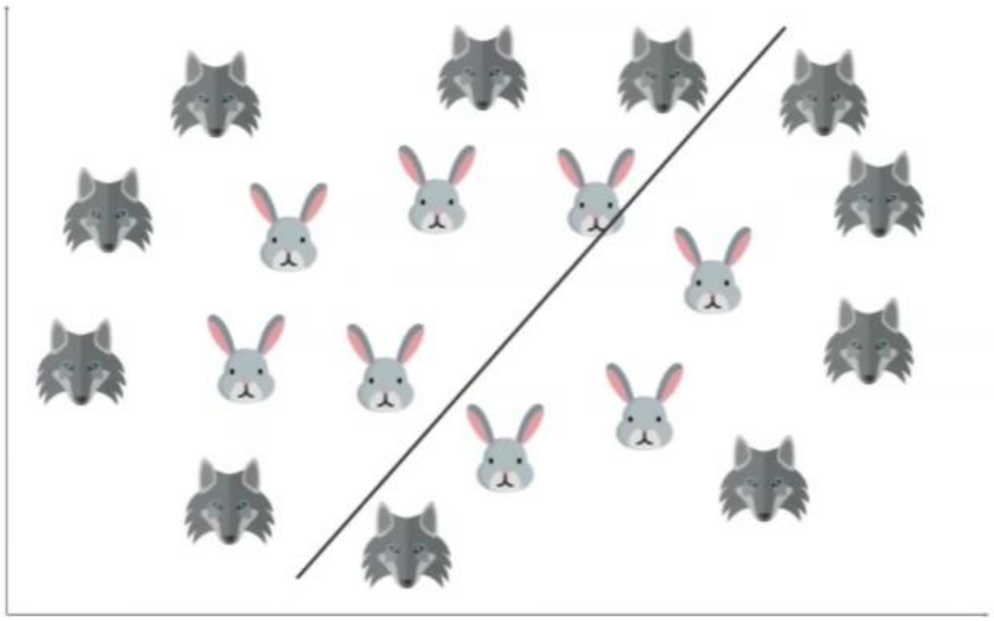
Support Vector Machine (SVM) is a supervised classification method that separates data using hyperplanes.



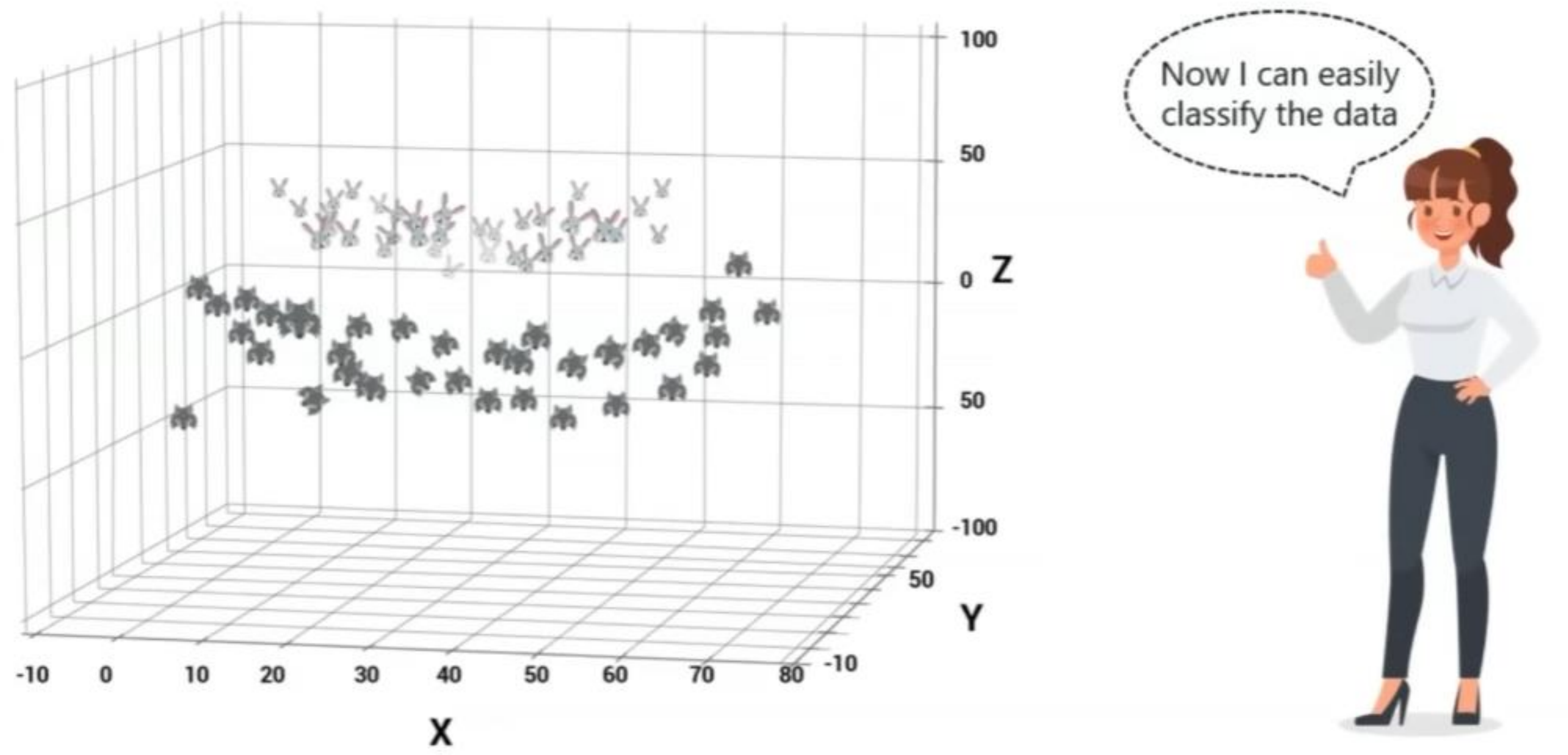
www.edureka.co

INTRODUCTION TO NON-LINEAR SVM

Non-linear SVM is used when the data can't be separated using a straight line

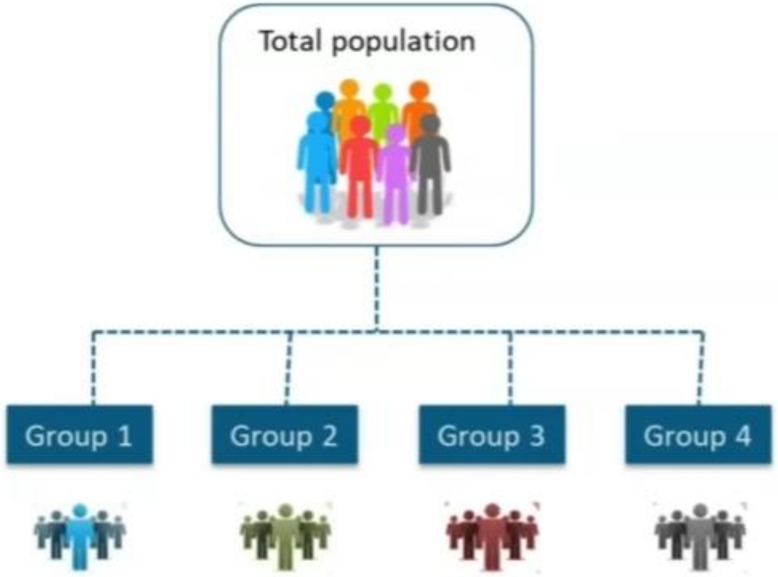


INTRODUCTION TO NON-LINEAR SVM

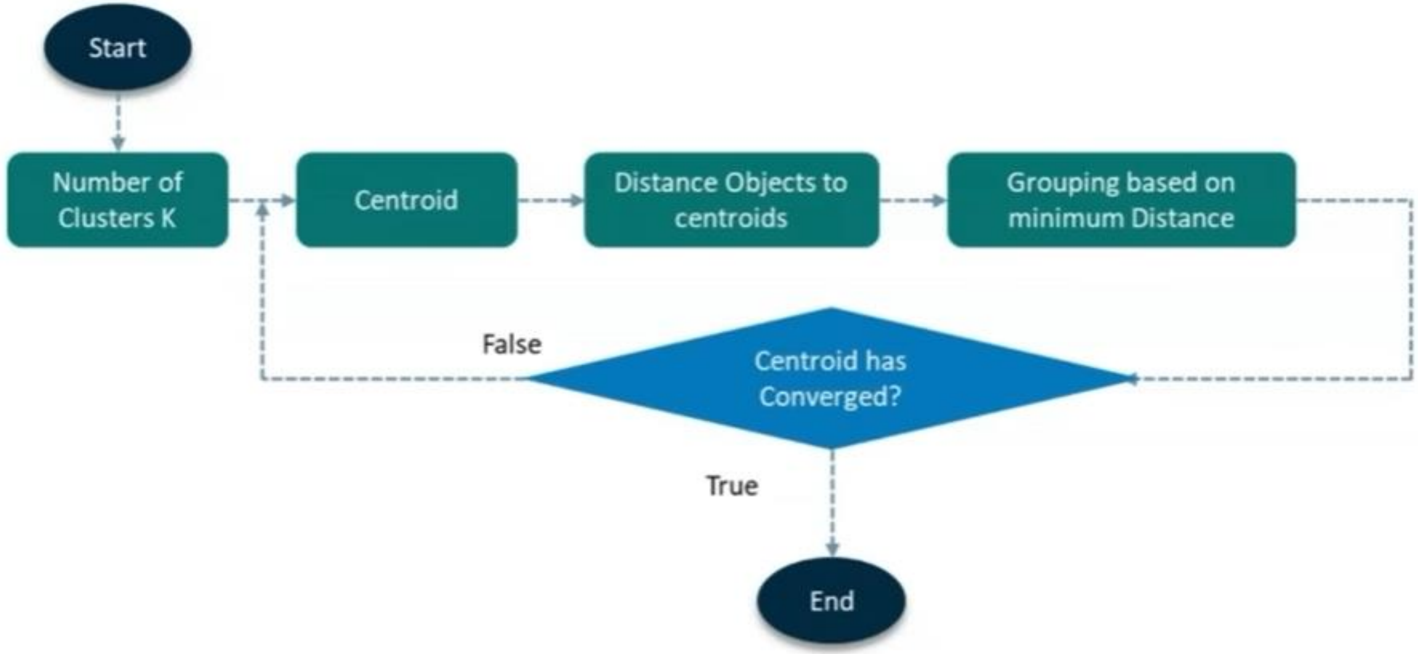


K-MEANS CLUSTERING

The process by which objects are classified into a predefined number of groups so that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group.

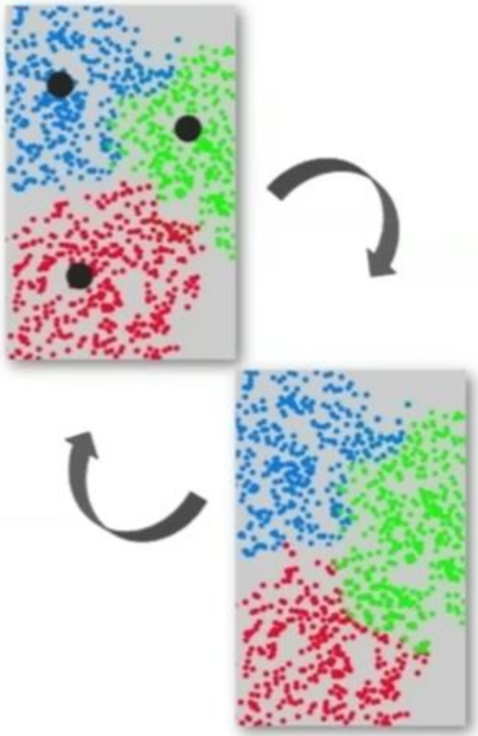


K-MEANS CLUSTERING



K-MEANS CLUSTERING

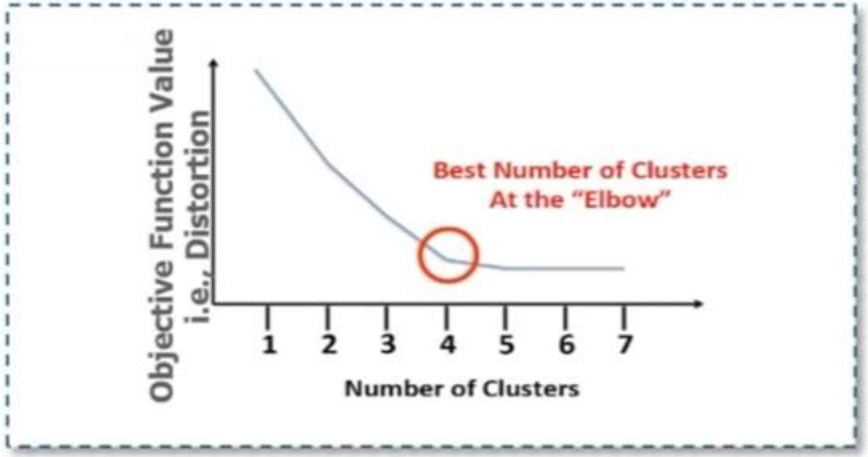
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assigns the point to the closest cluster.
4. Next the Centroids are calculated again, when we have our new cluster.
5. The distance of the points from the centre of clusters are calculated again and points are assigned to the closest cluster.
6. And then again the new centroid for the cluster is calculated.
7. These steps are repeated until we have a repetition in centroids or new centroids are very close to the previous ones.



K-MEANS CLUSTERING

The Elbow Method:

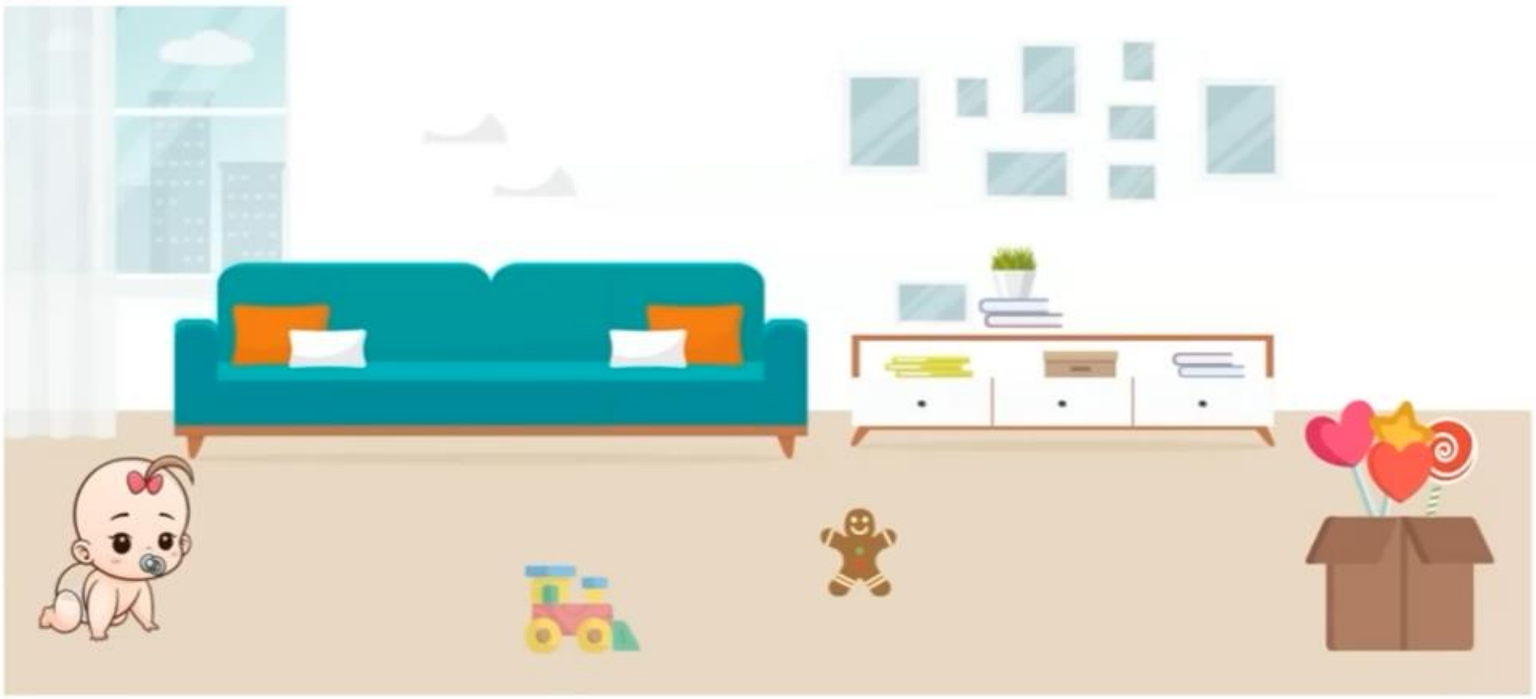
First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically:



$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} dist(x, c_i)^2$$

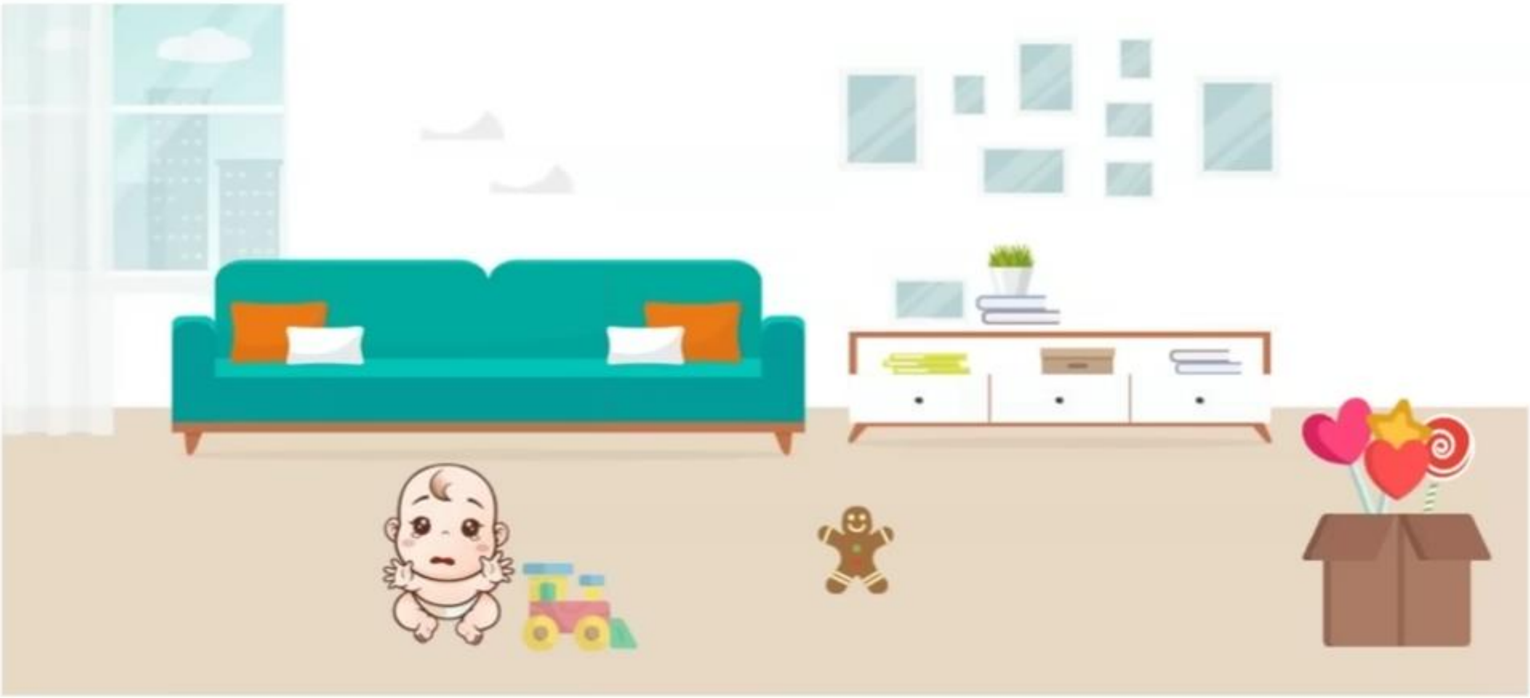
REINFORCEMENT LEARNING ANALOGY

Scenario 1: Baby starts crawling and makes it to the candy



REINFORCEMENT LEARNING ANALOGY

Scenario 2: Baby starts crawling but falls due to some hurdle in between



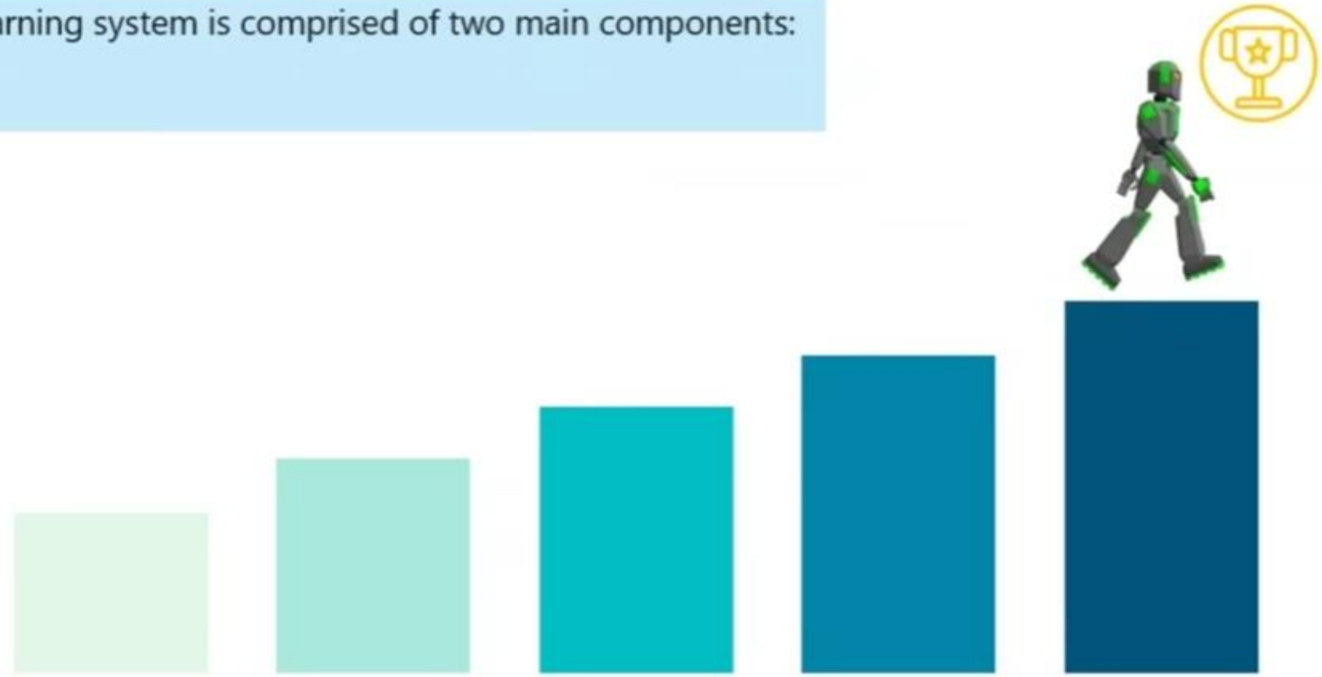
REINFORCEMENT LEARNING

Reinforcement Learning system is comprised of two main components:

- Agent
- Environment

REINFORCEMENT LEARNING

Reinforcement Learning system is comprised of two main components:
- Agent
- Environment



COUNTER STRIKE EXAMPLE

1. The RL Agent (Player1) collects state S^0 from the environment
2. Based on the state S^0, the RL agent takes an action A^0, initially the action is random
3. The environment is now in a new state S^1
4. RL agent now gets a reward R^1 from the environment
5. The RL loop goes on until the RL agent is dead or reaches the destination

BASIC TERMINOLOGIES IN RL

Agent: The RL algorithm that learns from trial and error

Environment: The world through which the agent moves

Action (A): All the possible steps that the agent can take

State (S): Current condition returned by the environment

BASIC TERMINOLOGIES IN RL

Reward (R): An instant return from the environment to appraise the last action

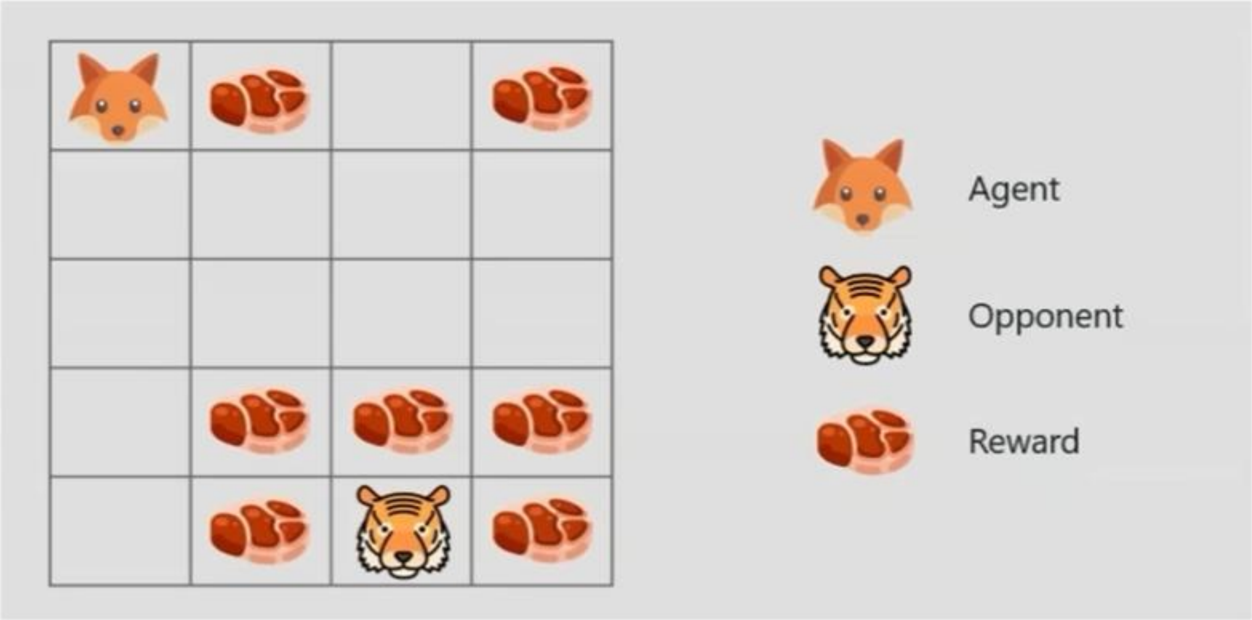
Policy (π): The approach that the agent uses to determine the next action based on the current state

Value (V): The expected long-term return with discount, as opposed to the short-term reward R

Action-value (Q): This similar to Value, except, it takes an extra parameter, the current action (A)

REWARD MAXIMIZATION

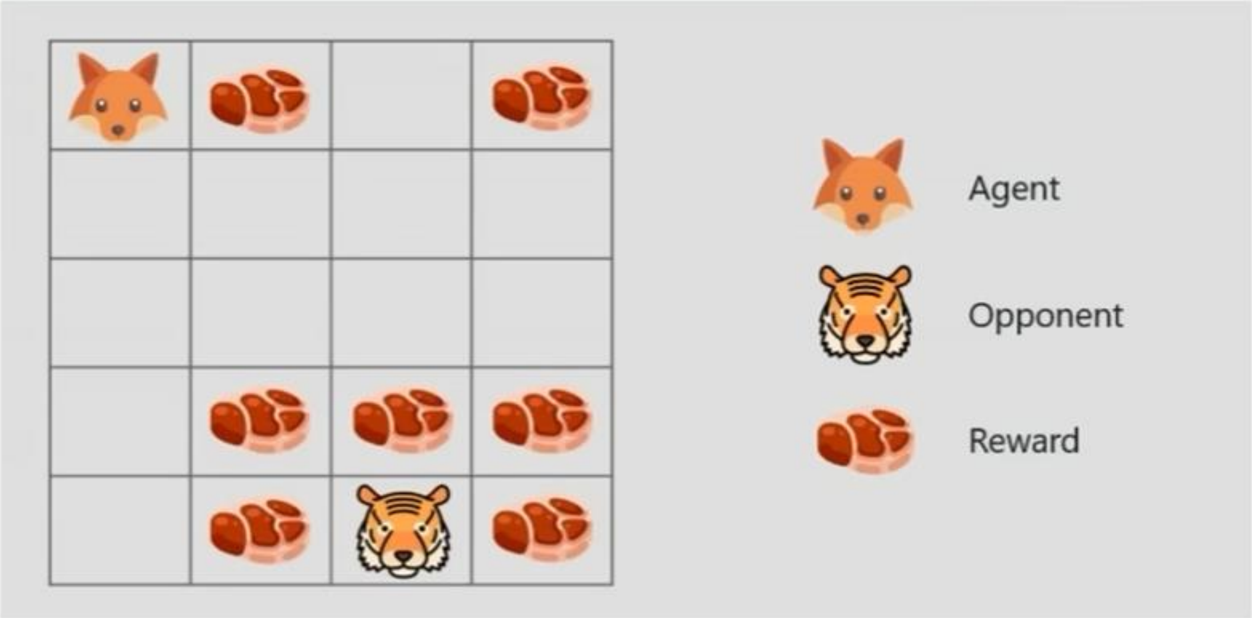
Reward maximization theory states that, *a RL agent must be trained in such a way that, he takes the best action so that the reward is maximum.*



EXPLORATION & EXPLOITATION

Exploitation is about using the already known exploited information to heighten the rewards

Exploration is about exploring and capturing more information about an environment

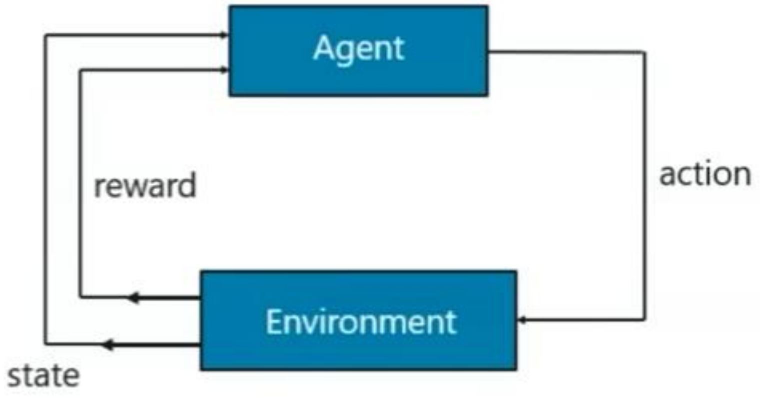


MARKOV'S DECISION PROCESS

The mathematical approach for mapping a solution in reinforcement learning is called *Markov Decision Process* (MDP)

The following parameters are used to attain a solution:

- Set of actions, A
- Set of states, S
- Reward, R
- Policy, π
- Value, V

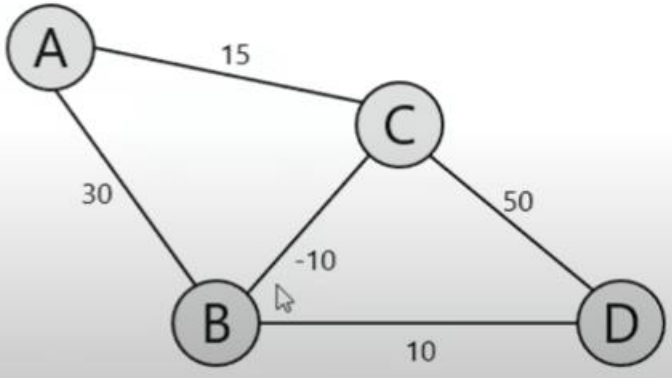


MARKOV'S DECISION PROCESS

Goal: Find the shortest path between A and D with minimum possible cost

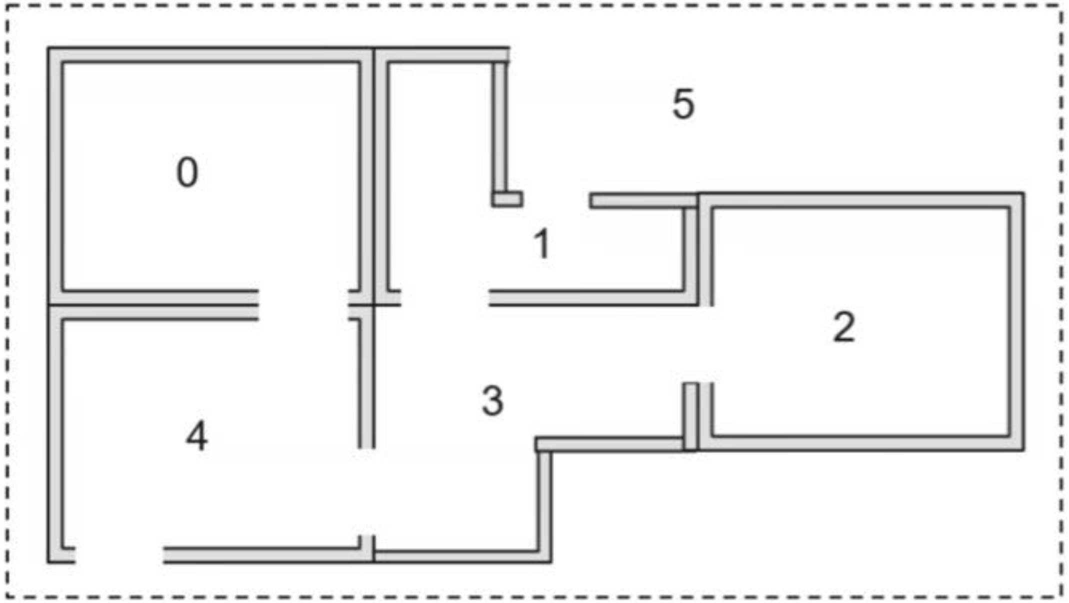
In this problem,

- Set of states are denoted by nodes i.e. {A, B, C, D}
- Action is to traverse from one node to another {A -> B, C -> D}
- Reward is the cost represented by each edge
- Policy is the path taken to reach the destination {A -> B -> D}



UNDERSTANDING Q-LEARNING

Place an agent in any one of the rooms (0,1,2,3,4) and the goal is to reach outside the building (room 5)

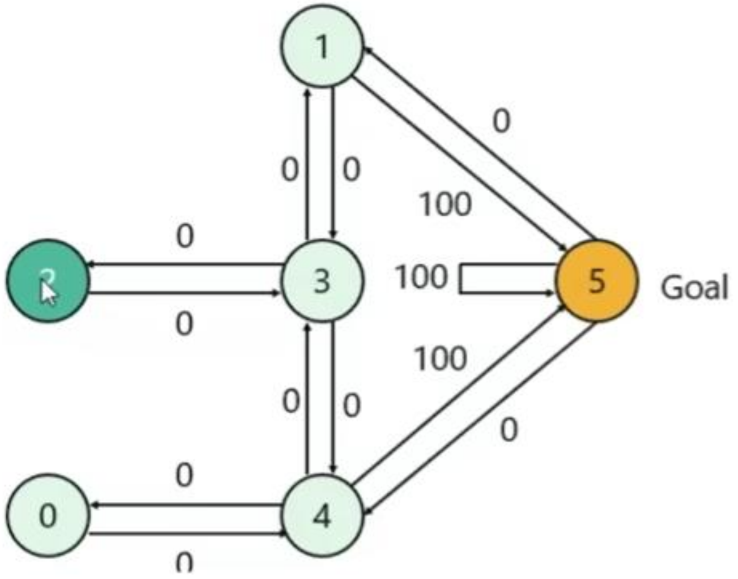


- 5 rooms in a building connected by doors
- each room is numbered 0 through 4
- The outside of the building can be thought of as one big room (5)
- Doors 1 and 4 lead into the building from room 5 (outside)

UNDERSTANDING Q-LEARNING

The terminology in Q-Learning includes the terms state and action:

- Room (including room 5) represents a state
- agent's movement from one room to another represents an action
- In the figure, a state is depicted as a node, while "action" is represented by the arrows

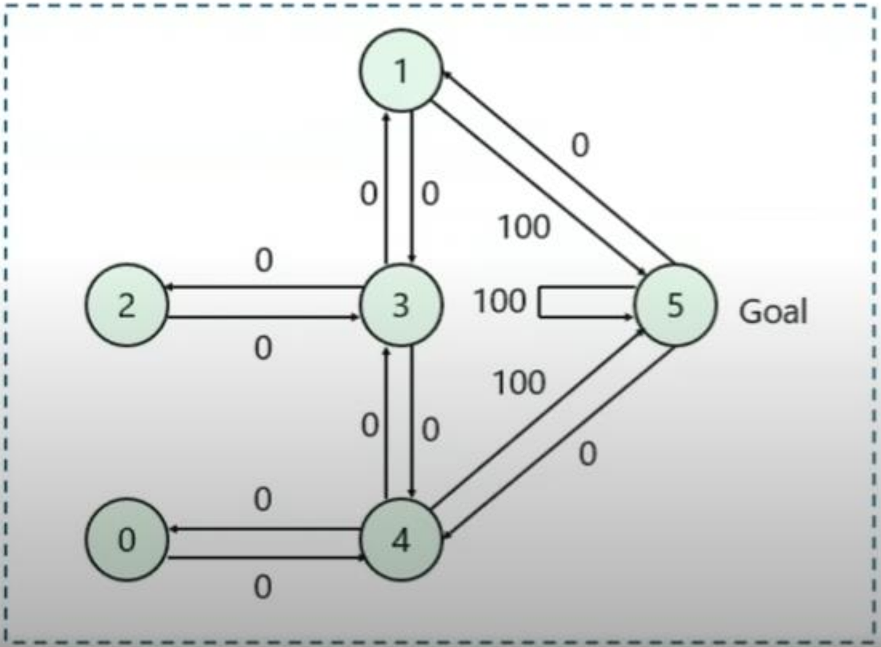


Example (Agent traverse from room 2 to room5):

1. Initial state = state 2
2. State 2 -> state 3
3. State 3 -> state (2, 1, 4)
4. State 4 -> state 5

UNDERSTANDING Q-LEARNING

We can put the state diagram and the instant reward values into a reward table, matrix R.



$$
R = \begin{array}{c c}
 & \text{Action} \\
\text{State} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix}
\end{array}
$$

The -1's in the table represent null values

UNDERSTANDING Q-LEARNING

Add another matrix Q, representing the memory of what the agent has learned through experience.

- The rows of matrix Q represent the current state of the agent
- columns represent the possible actions leading to the next state
- Formula to calculate the Q matrix:

*Q(state, action) = R(state, action) + Gamma * Max [Q(next state, all actions)]*

Note

The Gamma parameter has a range of 0 to 1 (0 <= Gamma > 1).

- If Gamma is closer to zero, the agent will tend to consider only immediate rewards.
- If Gamma is closer to one, the agent will consider future rewards with greater weight

Q-LEARNING ALGORITHM

1. Set the gamma parameter, and environment rewards in matrix R
2. Initialize matrix Q to zero
3. Select a random initial state
4. Set initial state = current state
5. Select one among all possible actions for the current state
6. Using this possible action, consider going to the next state
7. Get maximum Q value for this next state based on all possible actions
8. Compute: Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]
9. Repeat above steps until current state = goal state

Q-LEARNING EXAMPLE

First step is to set the value of the learning parameter Gamma = 0.8, and the initial state as Room 1.

Next, initialize matrix Q as a zero matrix:

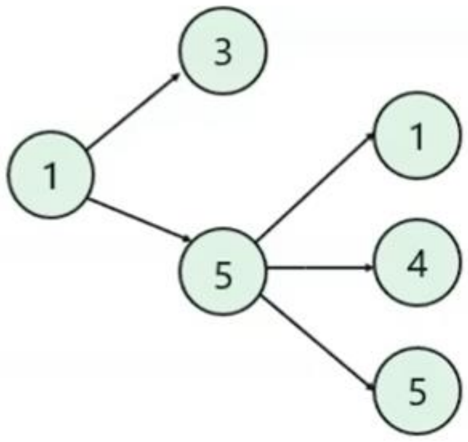
- From room 1 you can either go to room 3 or 5, let's select room 5.
- From room 5, calculate maximum Q value for this next state based on all possible actions:

*Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]*

Q(1,5) = R(1,5) + 0.8 * Max[Q(5,1), Q(5,4), Q(5,5)] = 100 + 0.8 * 0 = 100

$$Q = \begin{array}{c|c} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$R = \begin{array}{c|c} & \text{Action} \\ \text{State} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$



Q-LEARNING EXAMPLE

For the next episode, the next state, 1, now becomes the current state. We repeat the inner loop of the Q learning algorithm because state 1 is not the goal state.

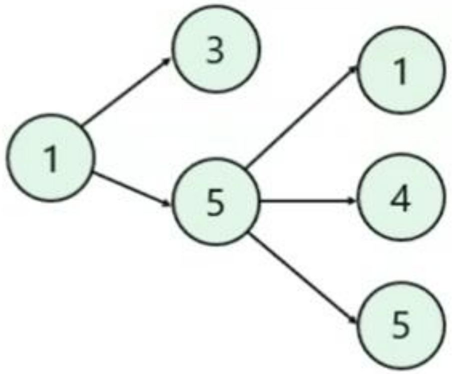
- From room 1 you can either go to room 3 or 5, let's select room 5.
- From room 5, calculate maximum Q value for this next state based on all possible actions:

*Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]*

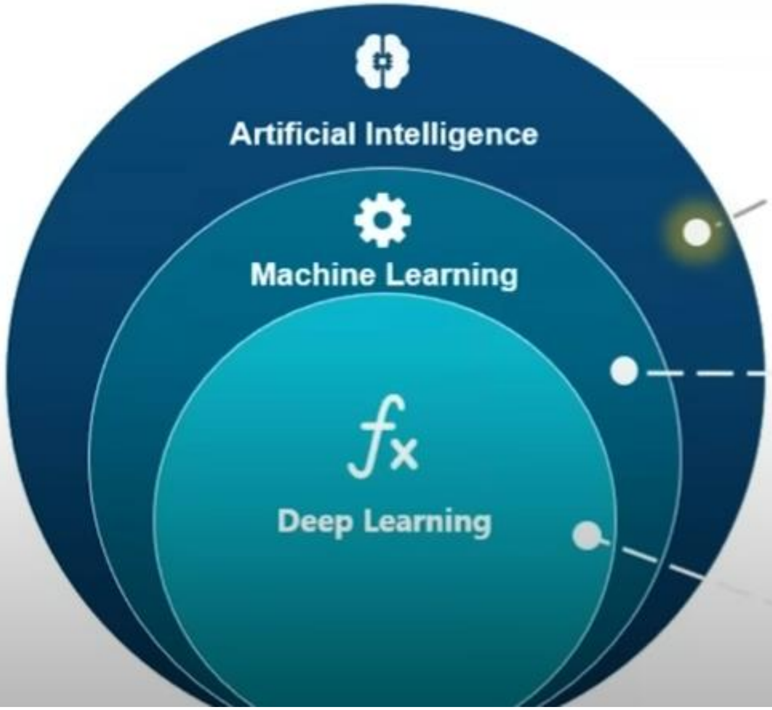
Q(1,5) = R(1,5) + 0.8 * Max[Q(5,1), Q(5,4), Q(5,5)] = 100 + 0.8 * 0 = 100
The matrix Q remains the same since, Q(1,5) is already fed to the agent

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$R = \begin{array}{c} \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$



AI VS ML VS DL

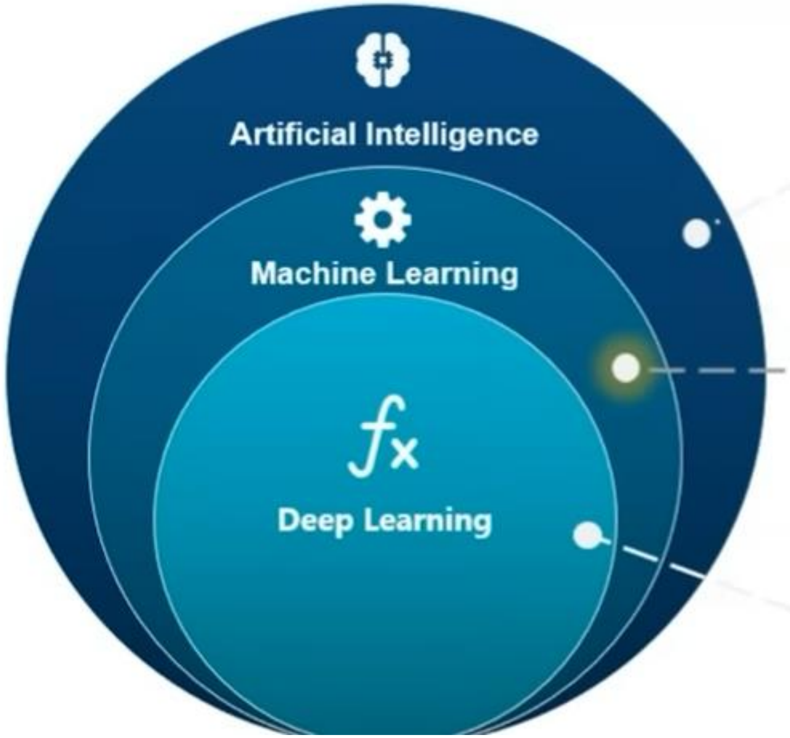


ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the

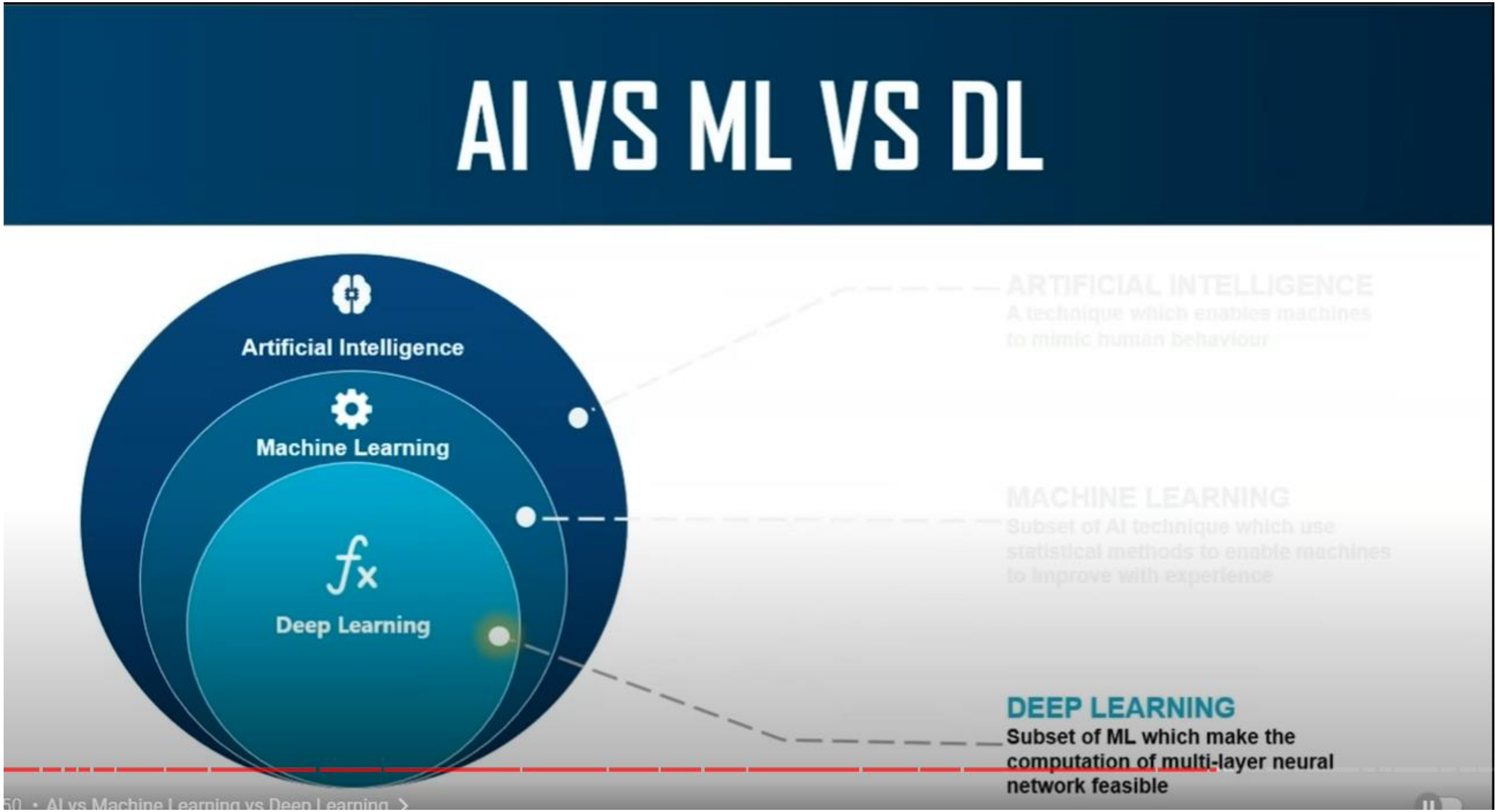
AI VS ML VS DL



ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the computation of multi-layer neural


AI VS ML VS DL
Artificial Intelligence
Machine Learning
Deep Learning
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible
AI vs Machine Learning vs Deep Learning

LIMITATIONS OF ML

High Dimensional data

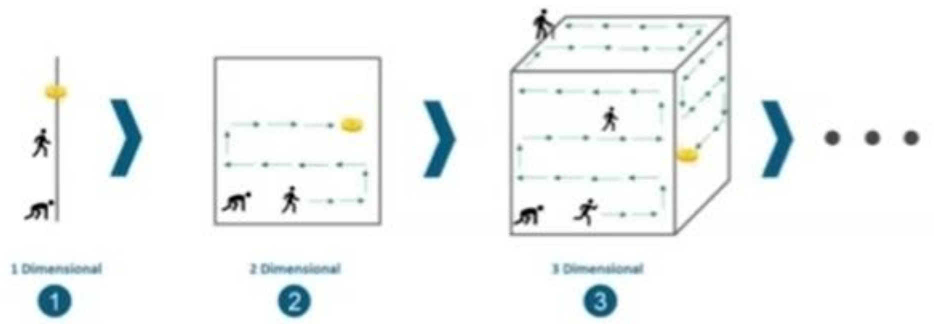
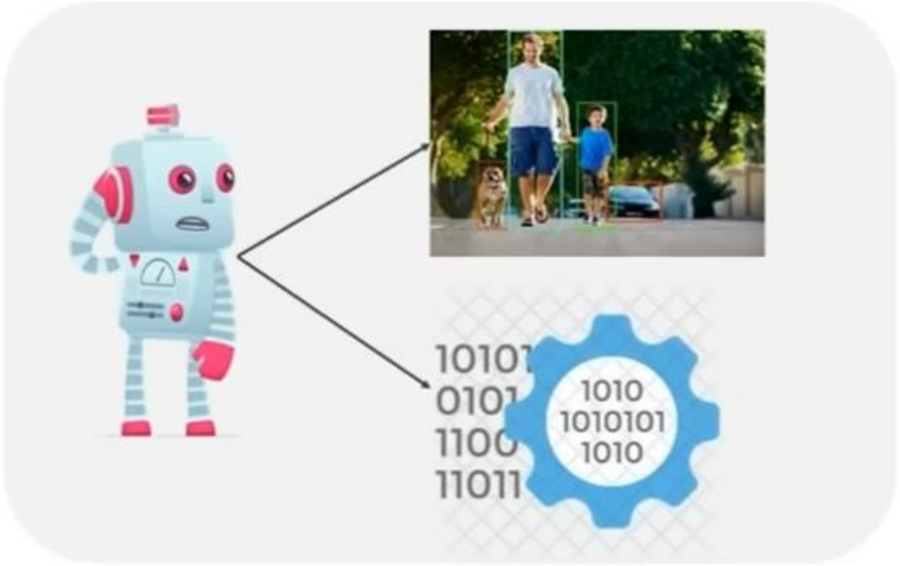
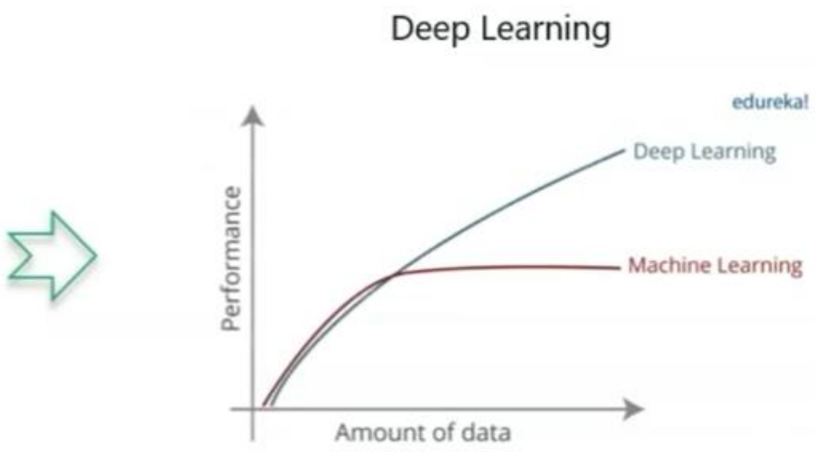


Image Recognition

LIMITATIONS OF ML

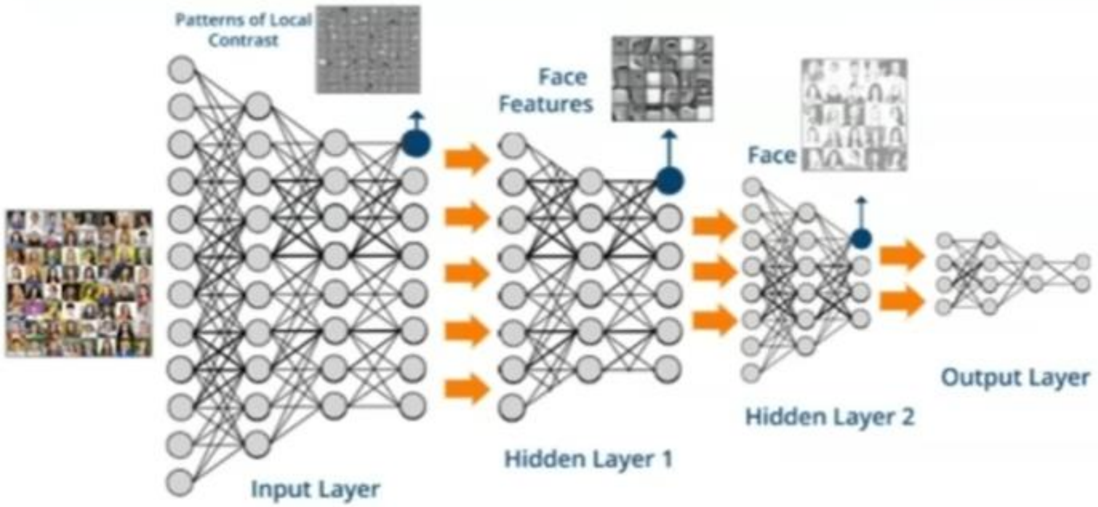
One of the big challenges with traditional Machine Learning models is a process called feature extraction. For complex problems such as object recognition or handwriting recognition, this is a huge challenge.





WHY DEEP LEARNING?

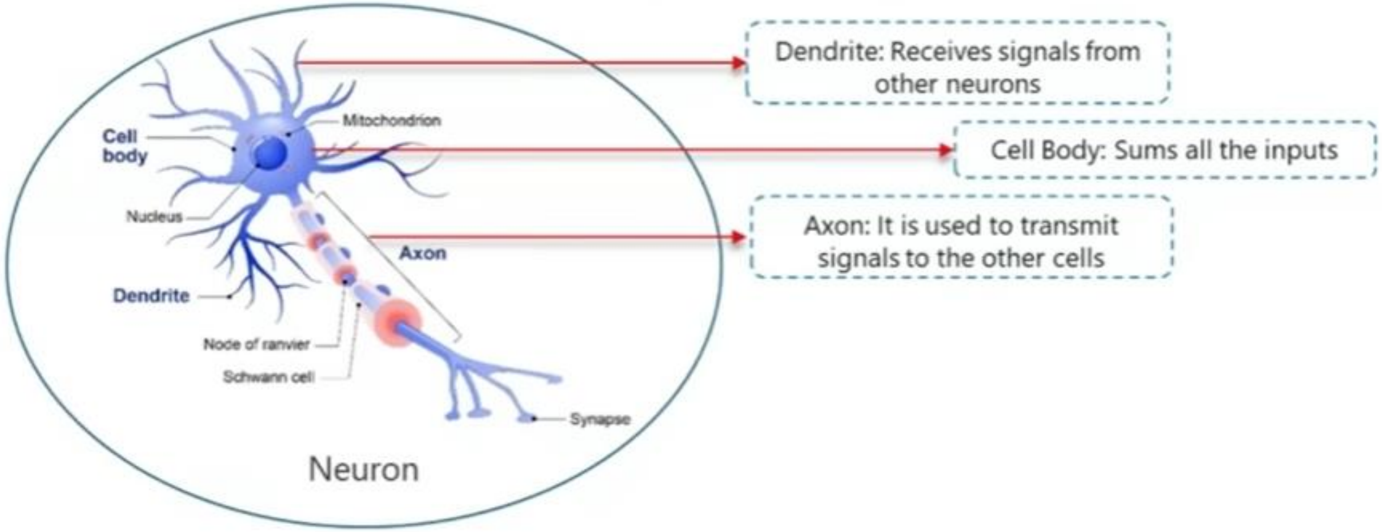
Deep Learning models are capable to focus on the right features by themselves, requiring little guidance from the programmer. These models also partially solve the dimensionality problem.



The idea behind Deep Learning is to build learning algorithms that mimic brain.

HOW DEEP LEARNING WORK?

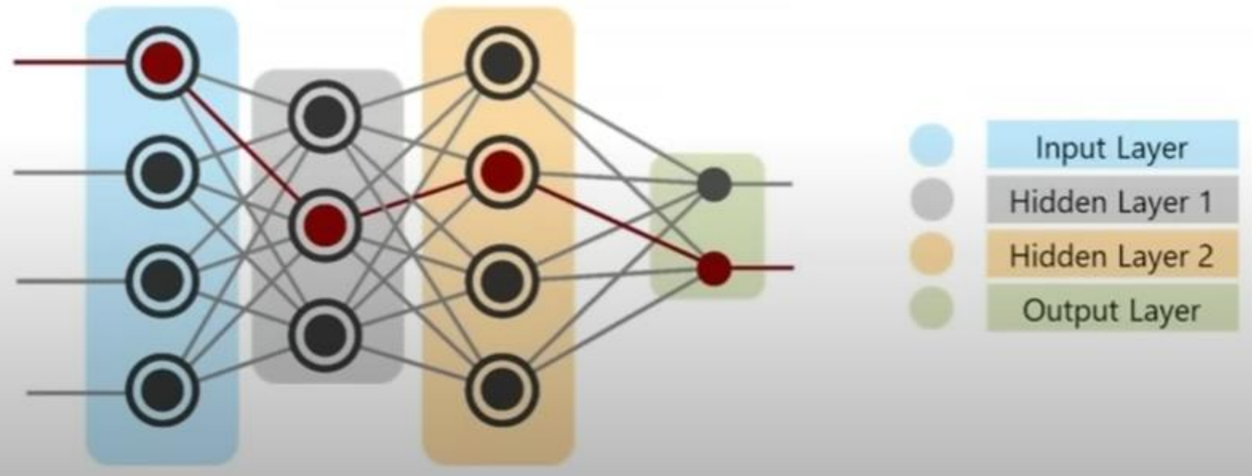
Deep learning is a form of machine learning that uses a model of computing that's very much inspired by the structure of the brain, so lets understand that first.



Biological Neuron

WHAT IS DEEP LEARNING?

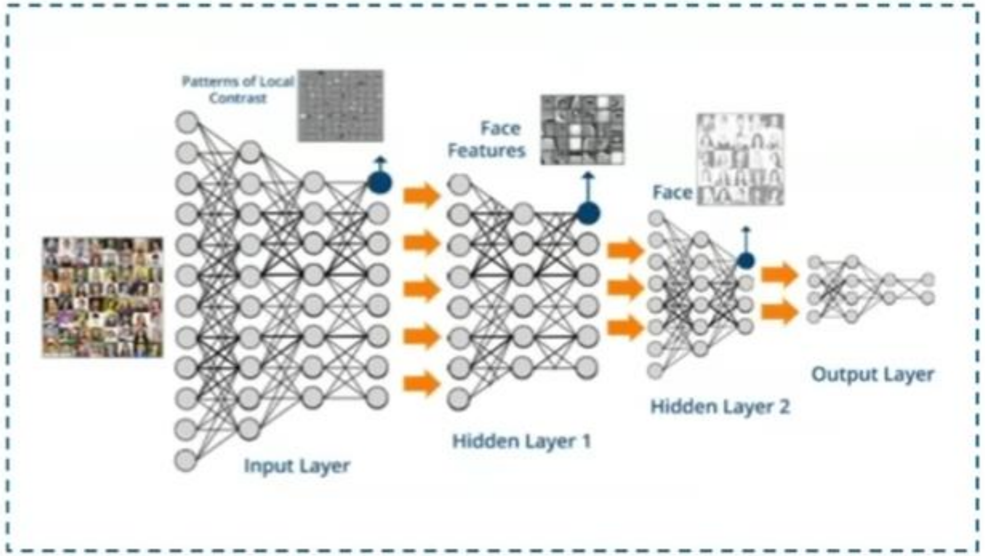
Deep Learning is a collection of statistical machine learning techniques used to learn feature hierarchies based on the concept of artificial neural networks.



UNDERSTANDING DEEP LEARNING

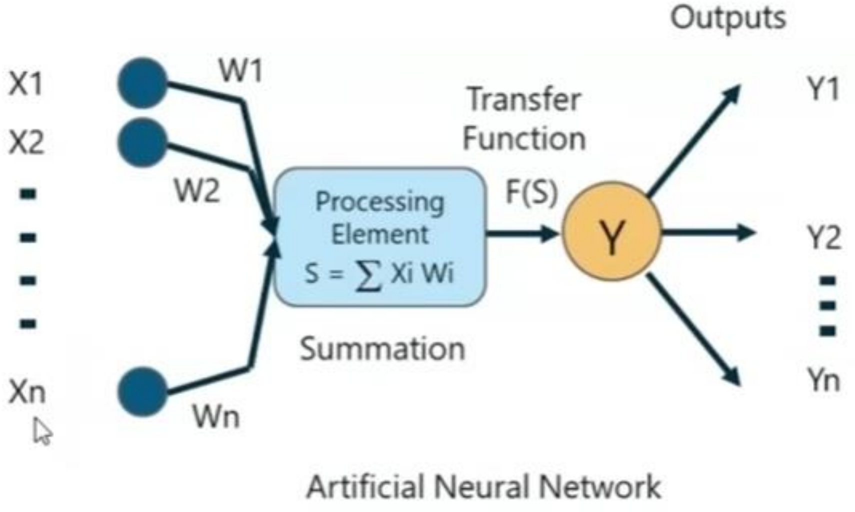
Image recognition using Deep Networks:

1. Pass the high dimensional data to the input layer
2. Output received from the input layer contains patterns which are extracted
3. Output will be fed to the Hidden layer 1
4. Hidden layer 2 will able to form the entire faces
5. The output layer performs classification



A PERCEPTRON

An Artificial Neuron or a Perceptron is a linear model used for binary classification. It models a neuron which has a set of inputs, each of which is given a specific weight. The neuron computes some function on these weighted inputs and gives the output.



Perceptron or Artificial Neuron

PERCEPTRON LEARNING ANALOGY

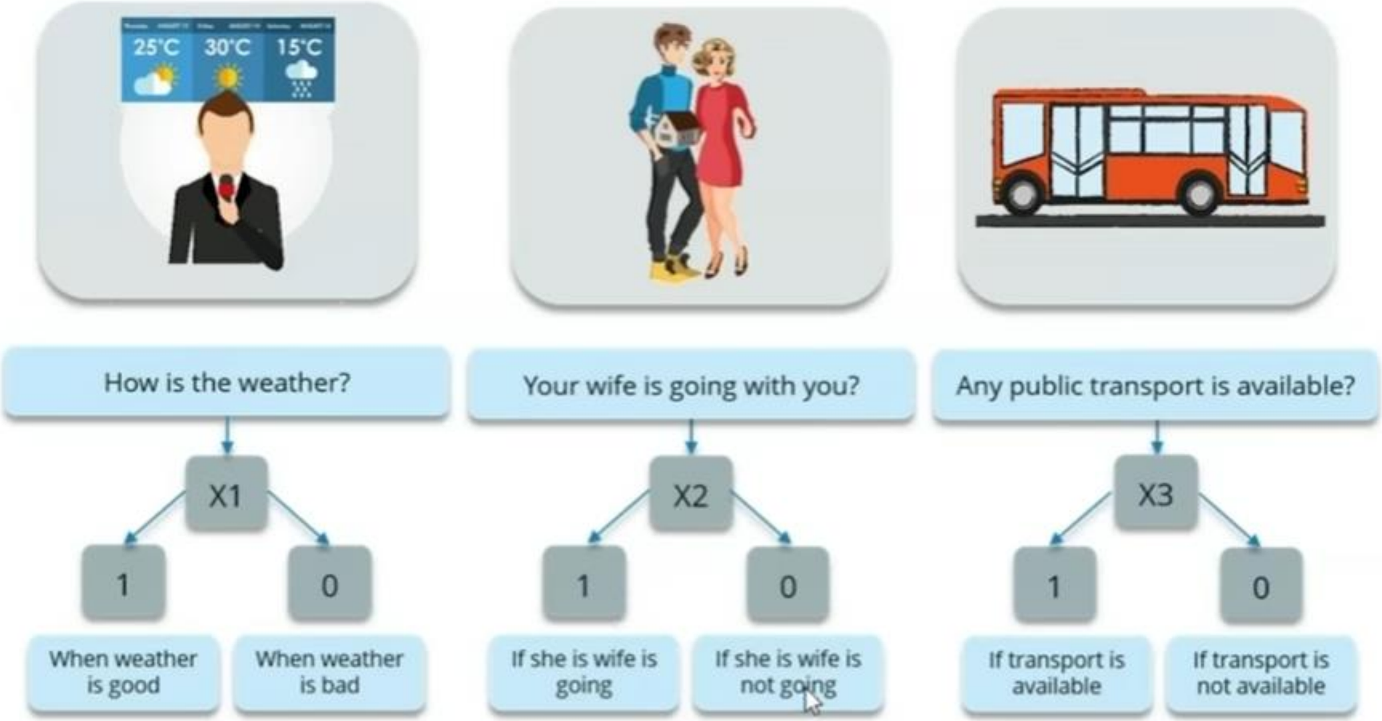
Suppose you want to go to a party happening near your house. And your decision depends on multiple factors:

1.How is the weather?

2.Your wife is going with you?

3.Any public transport is available?

PERCEPTRON LEARNING ANALOGY



PERCEPTRON LEARNING ANALOGY

Suppose for you the most important factor is weather.

Weather -> Good = Party

Weather -> Not Good = Sit home

X1 = 1

Output =1

PERCEPTRON LEARNING ANALOGY

W1 = Weight associated with input X1
W2 = Weight associated with input X2
W3 = Weight associated with input X3

W1 = 6, W2 = 2, W3 = 2

Threshold = 5

It will fire when weather is good and won't fire if weather is bad irrespective of the other inputs

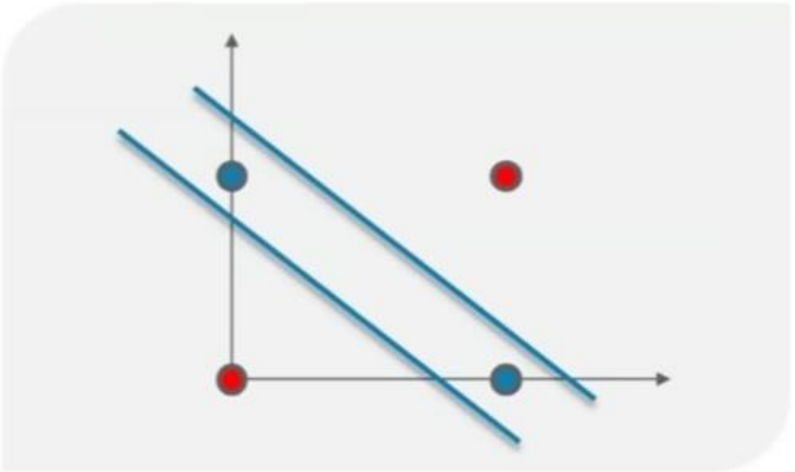
W1 = 6, W2 = 2, W3 = 2

Threshold = 3

It will fire when either x1 is high or the other two inputs are high

LIMITATIONS OF A PERCEPTRON

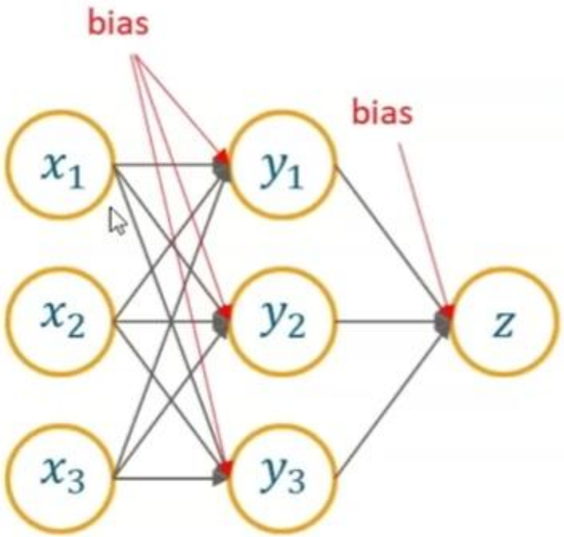
Dealing with non-linearly separable data:

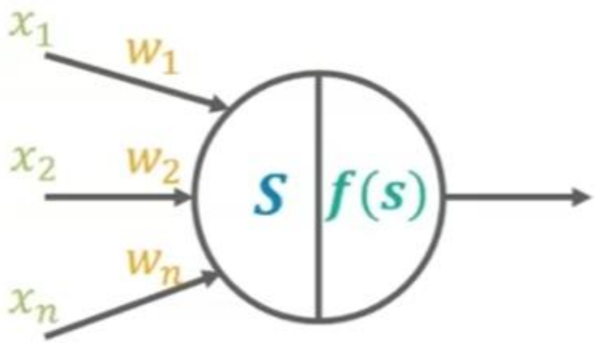


A Multilayer Perceptron with backpropagation can be used to solve this problem.

MULTILAYER PERCEPTRON

A Multi-layer Perceptron has the same structure of a single layer perceptron but with one or more hidden layers and is thus considered a deep neural network.





Summation:

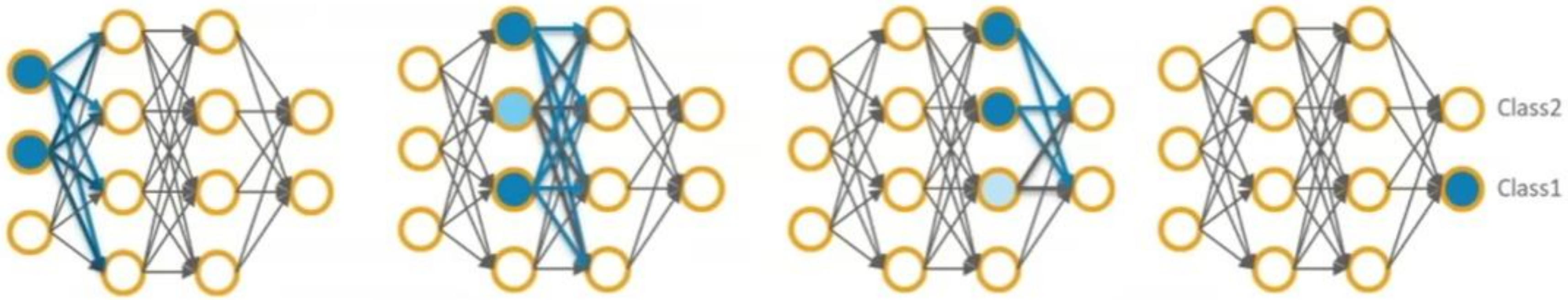
$$S = \sum_{i=1}^{n} w_i * x_1$$

Transformation:

$$f(x) = \frac{1}{1+e^{-\beta x}}$$

MULTILAYER PERCEPTRON

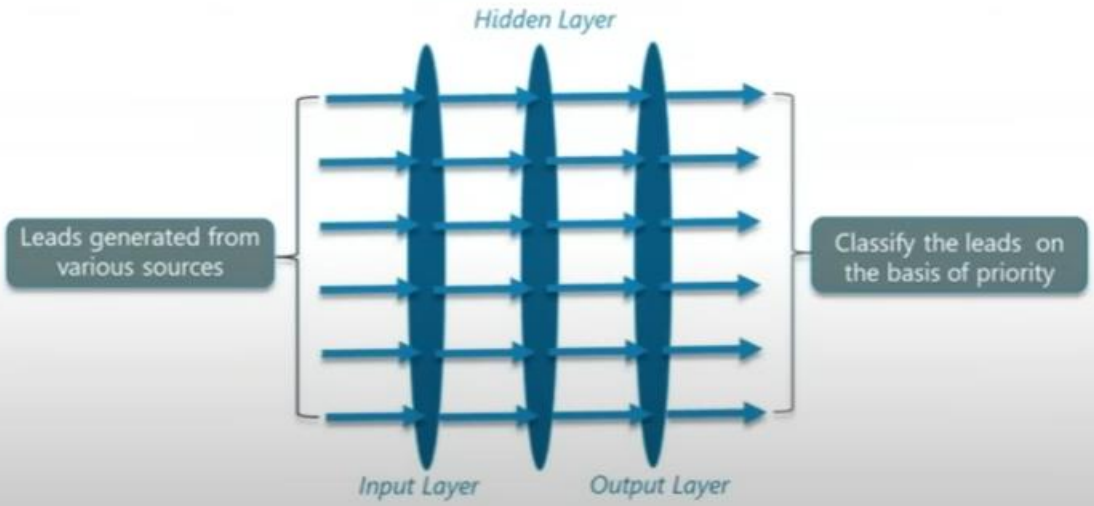
- The weights between the units are the primary means of long-term information storage in neural networks
- Updating the weights is the primary way the neural network learns new information



A set of inputs are passed to the first hidden layer, the activations from that layer are passed to the next layer and so on, until you reach the output layer.

BACKPROPAGATION

The Backpropagation algorithm is a supervised learning method for Multilayer Perceptron.



Maximum weight is assigned to the most important lead/input.

TRAINING A NEURAL NETWORK

The most common deep learning algorithm for supervised training of the multi-layer perceptrons is known as *backpropagation*. After calculating the weighted sum of inputs and passing them through the activation function we propagate backwards and update the weights to reduce the error (desired output – model output). Consider the below example:

Input	Desired Output
0	0
1	1
2	4

TRAINING A NEURAL NETWORK

Let's consider the initial value of the weight as 3 and see the model output:

Input	Desired Output	Model Output (W=3)
0	0	0
1	2	3
2	4	6

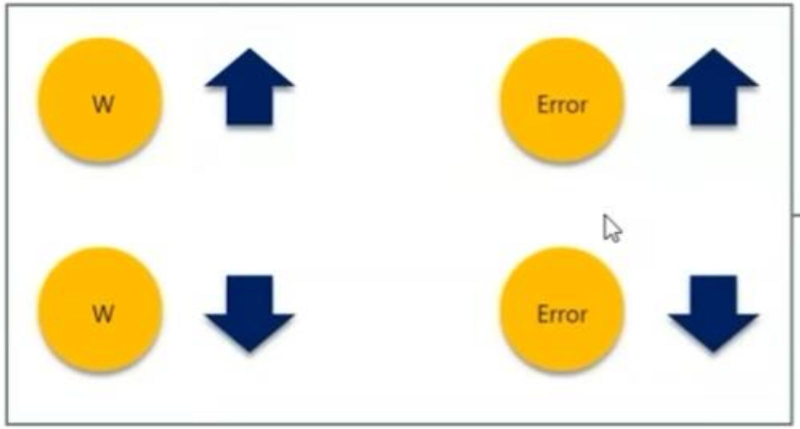
TRAINING A NEURAL NETWORK

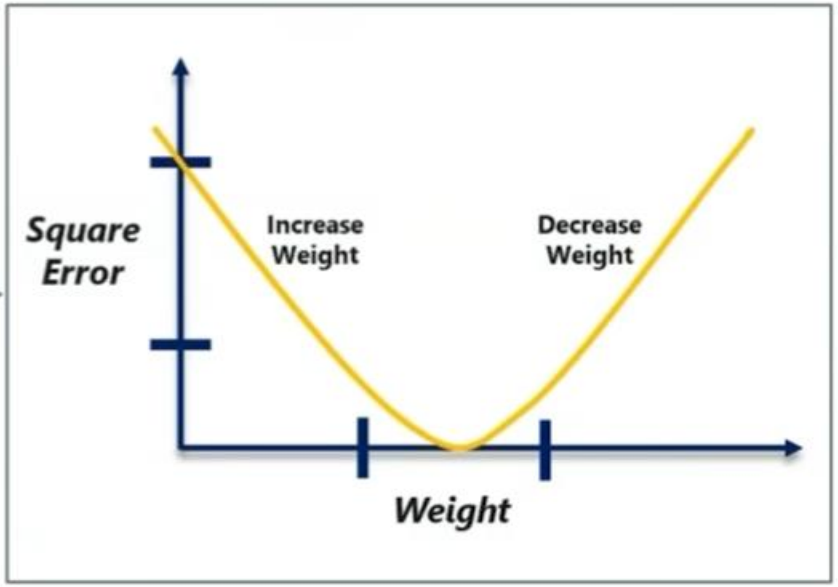
Notice that the error has increased

Input	Desired Output	Model Output (W=3)	Absolute Error	Square Error	Model Output (W=4)	Square Error
0	0	0	0	0	0	0
1	2	3	1	1	4	4
2	4	6	2	4	8	16

TRAINING A NEURAL NETWORK

Relationship between the assigned weight and the error





REDUCING THE ERROR/LOSS

Step 1: Calculate the error

Network Output

Error/Loss $J(\mathbf{w}) = \frac{1}{2}\sum_i (\text{target}^{(i)} - \text{output}^{(i)})^2$

Actual Output

Step 2: Calculate the rate of change of error w.r.t change in the weights

Learning Rate

Change in Weight $\Delta w_j = -\eta \frac{\partial J}{\partial w_j},$

Gradient

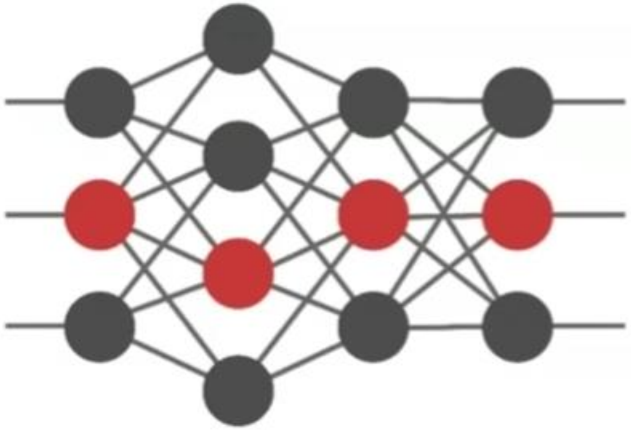
Step 3: Based on change in weight, update the weight value.

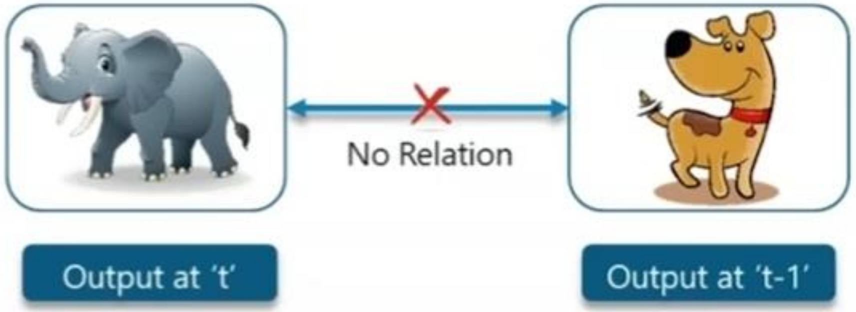
$$w := w + \Delta w$$

Updated weight value

WHY NOT FEED FORWARD NETWORK?

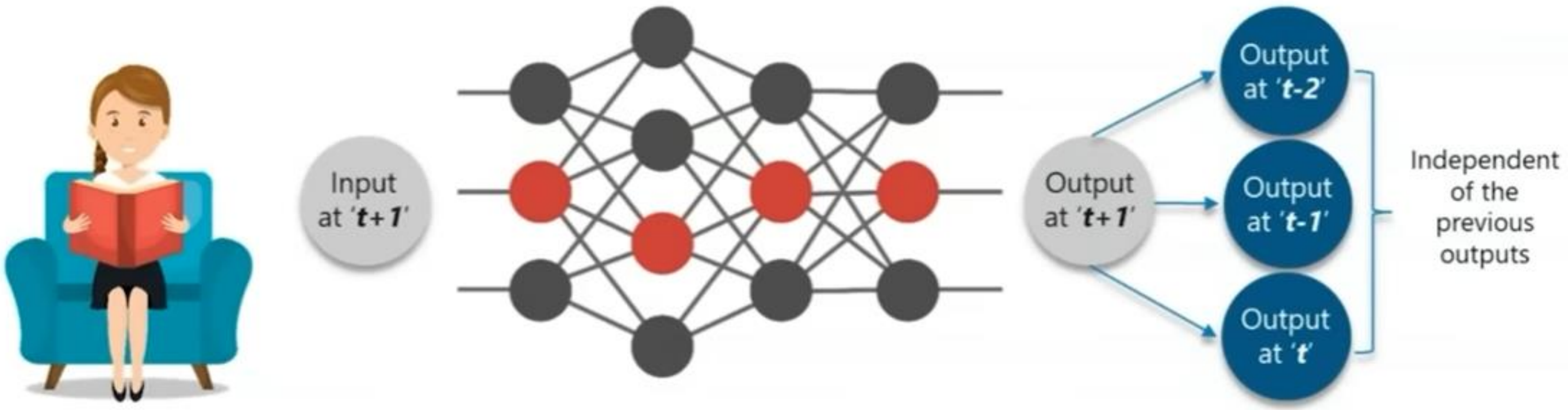
A trained feedforward network can be exposed to any random collection of photographs, and the first photograph it is exposed to will not necessarily alter how it classifies the second one.



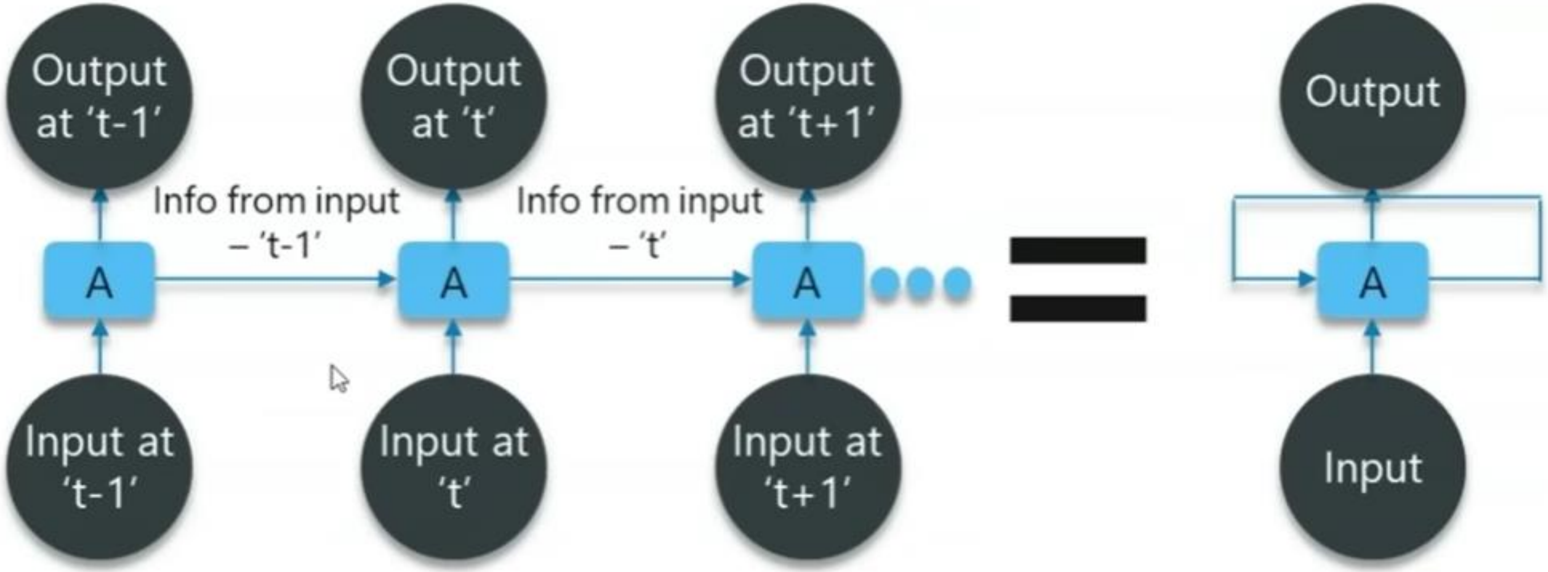


WHY NOT FEED FORWARD NETWORK?

When you read a book, you understand it based on your understanding of previous words



SOLUTION

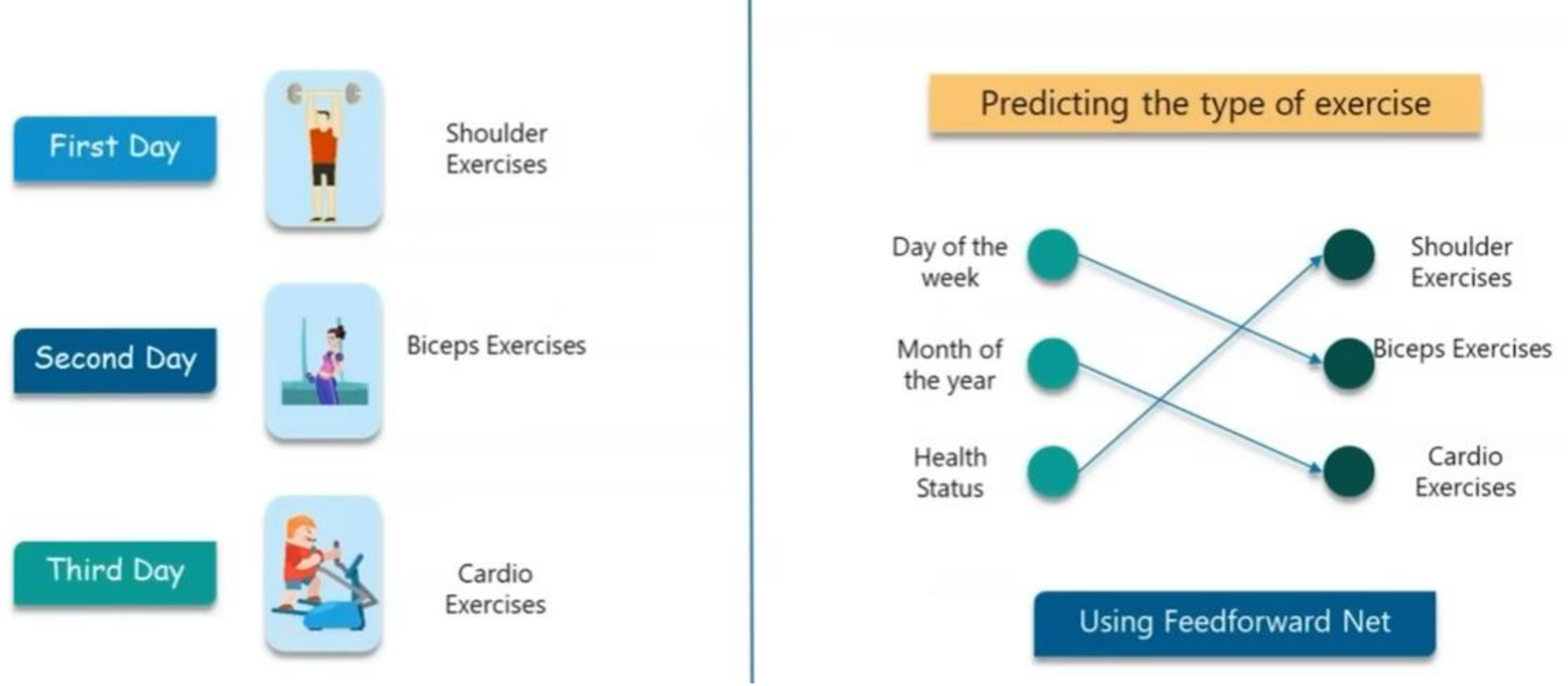


Output at 't-1'
Output at 't'
Output at 't+1'
Output
Info from input – 't-1'
Info from input – 't'
A
A
A
A
Input at 't-1'
Input at 't'
Input at 't+1'
Input

RECURRENT NEURAL NETWORK

Recurrent Networks are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or numerical times series data emanating from sensors, stock markets and government agencies.

RECURRENT NEURAL NETWORK



RECURRENT NEURAL NETWORK

First Day

Shoulder Exercises

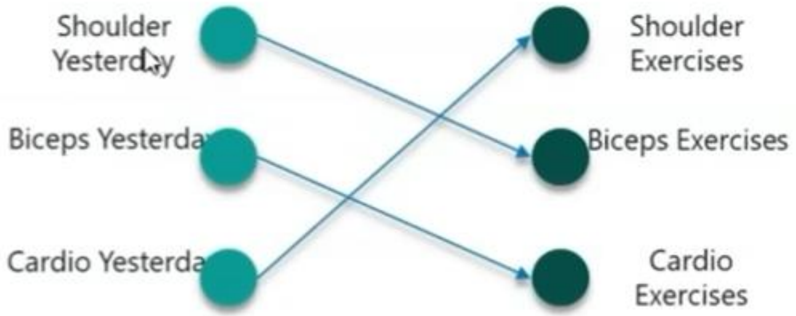
Second Day

Biceps Exercises

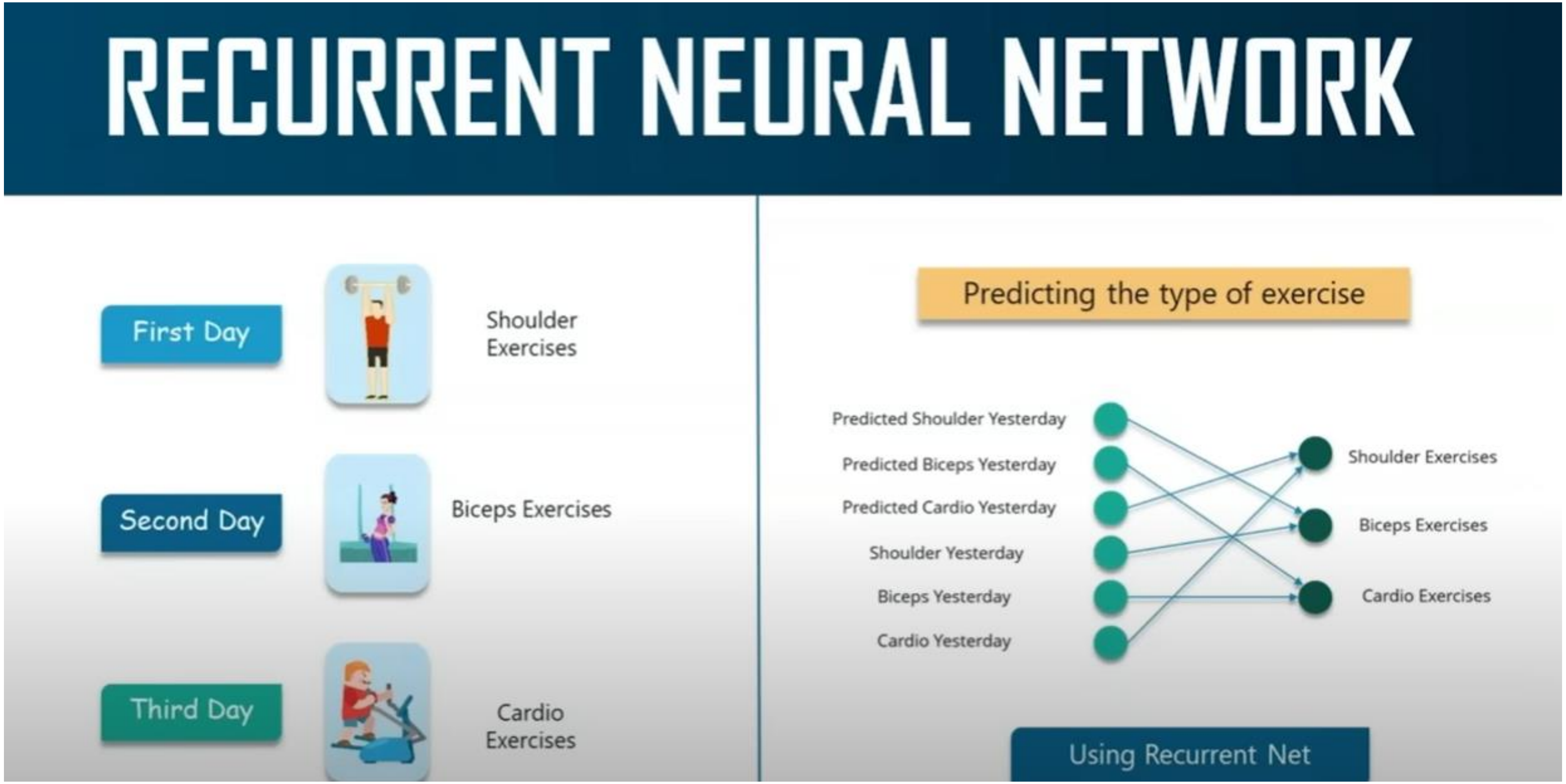
Third Day

Cardio Exercises

Predicting the type of exercise

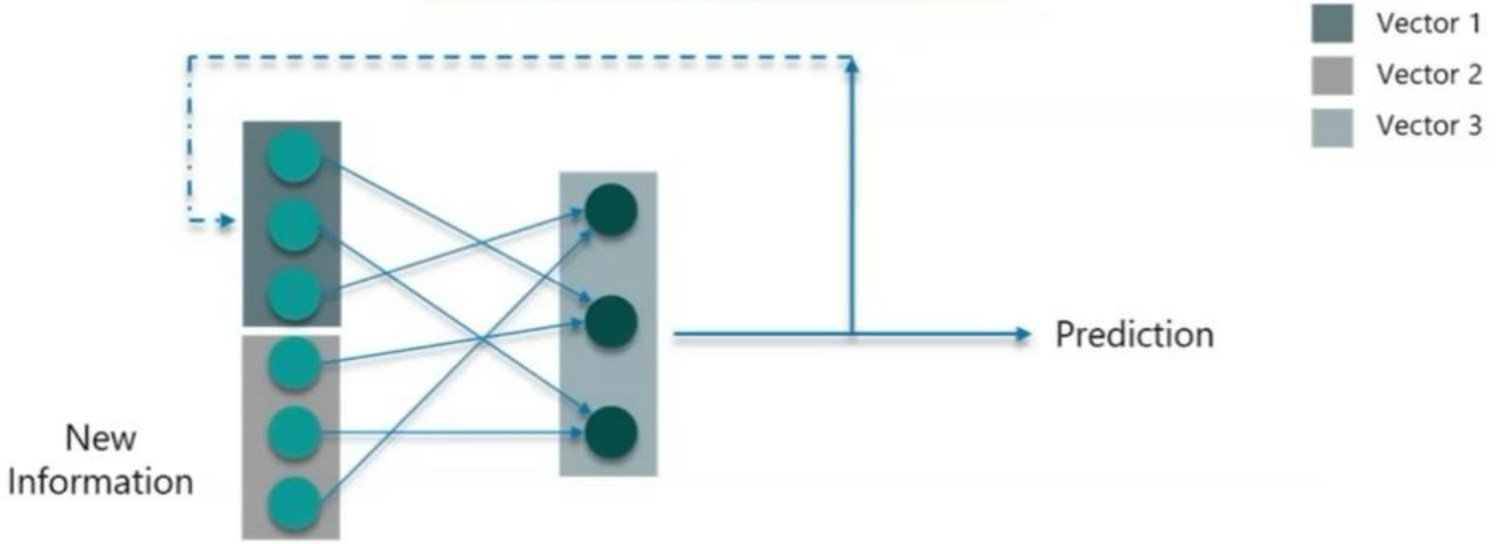


Using Recurrent Net


RECURRENT NEURAL NETWORK
First Day
Shoulder Exercises
Second Day
Biceps Exercises
Third Day
Cardio Exercises
Predicting the type of exercise
Predicted Shoulder Yesterday
Predicted Biceps Yesterday
Predicted Cardio Yesterday
Shoulder Yesterday
Biceps Yesterday
Cardio Yesterday
Shoulder Exercises
Biceps Exercises
Cardio Exercises
Using Recurrent Net

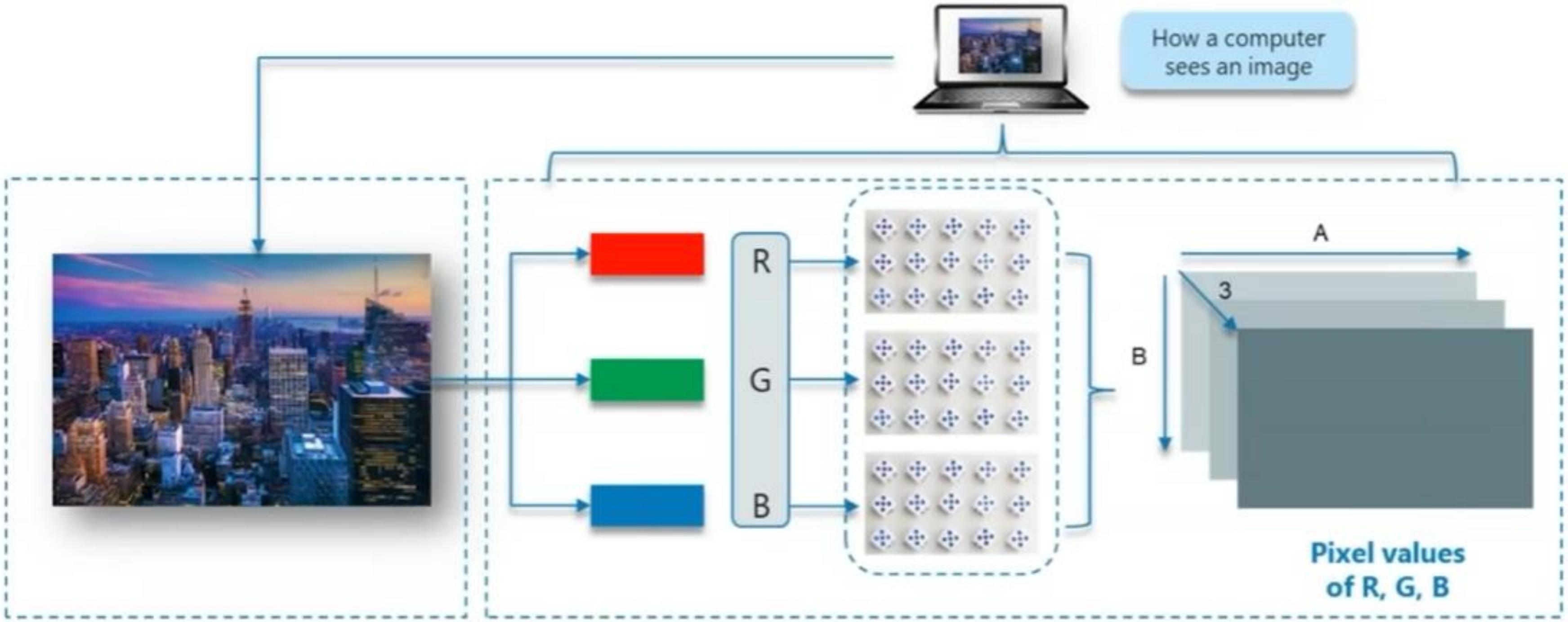
RECURRENT NEURAL NETWORK

Predicting the type of exercise



Using Recurrent Net

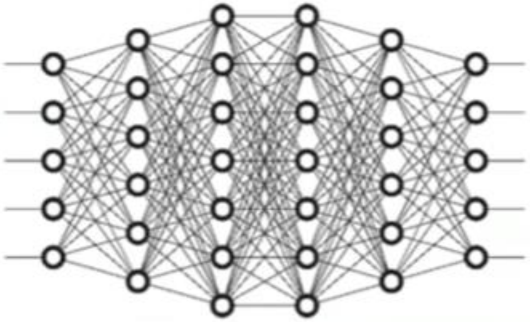
WHY CNN?



Size of the image will be – B x A x 3

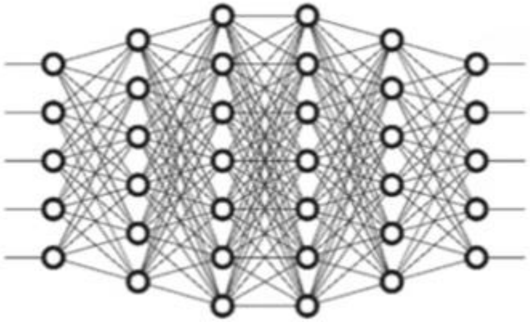
WHY CNN?

Image with
28 x 28 x 3
pixels



Number of weights in the first hidden layer will be 2352

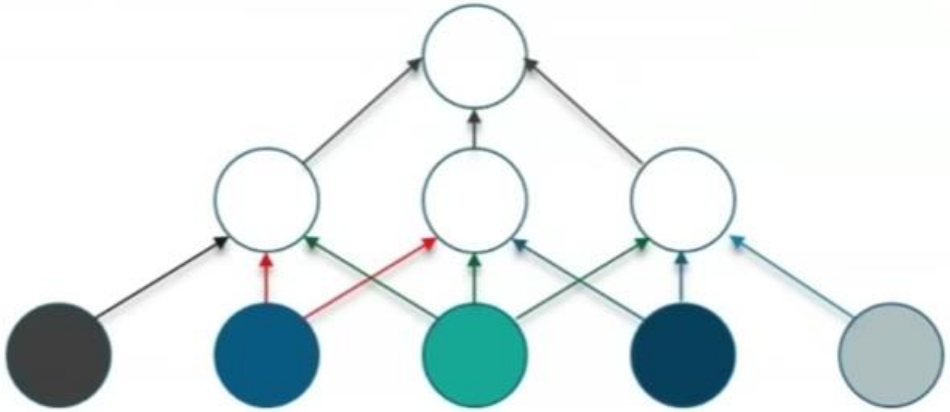
Image with
200 x 200 x 3
pixels



Number of weights in the first hidden layer will be 120,000

CONVOLUTIONAL NEURAL NETWORKS

In case of CNN the neuron in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner.



NEED FOR TEXT MINING & NLP

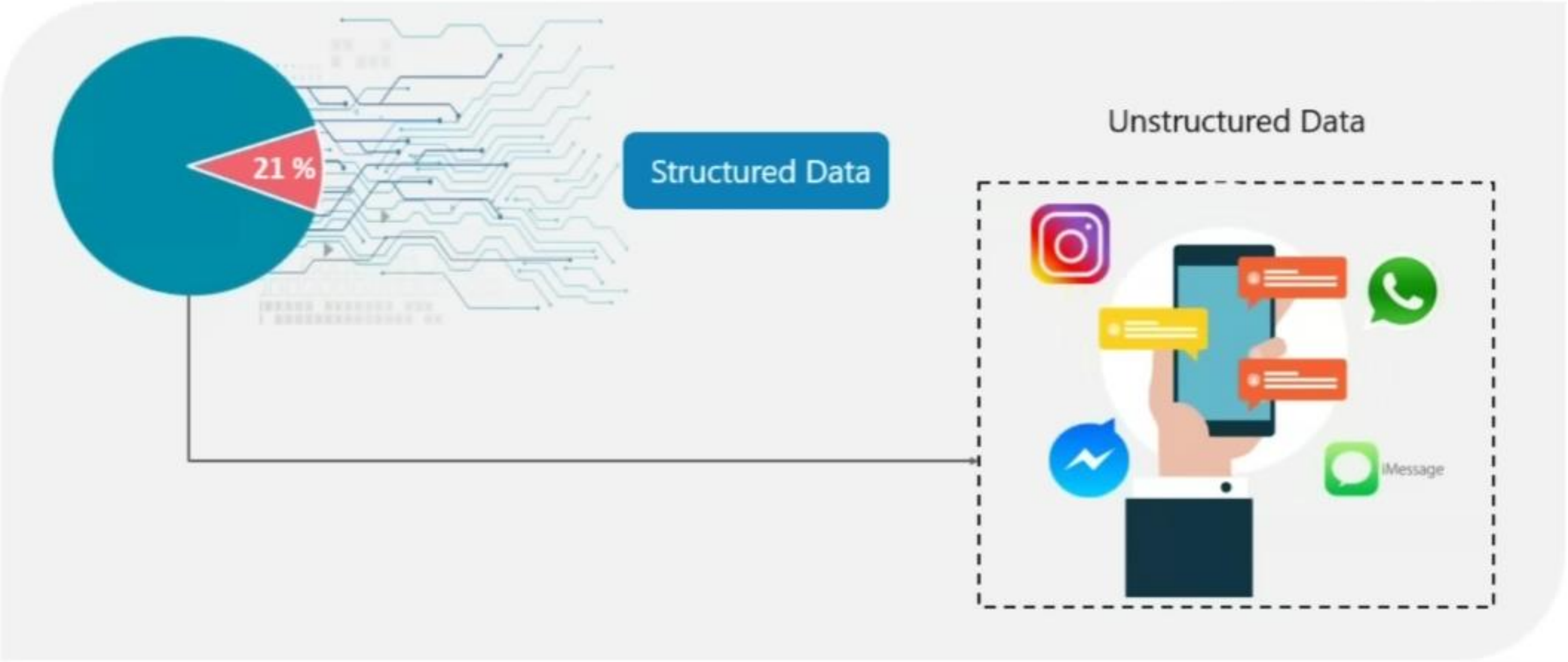
1,736,111 pictures

347,222 tweets

204,000,000 emails

4,166,667 likes &
200,000 photos

NEED FOR TEXT MINING & NLP



WHAT IS TEXT MINING?

Text Mining / Text Analytics is the process of deriving meaningful information from natural language text.

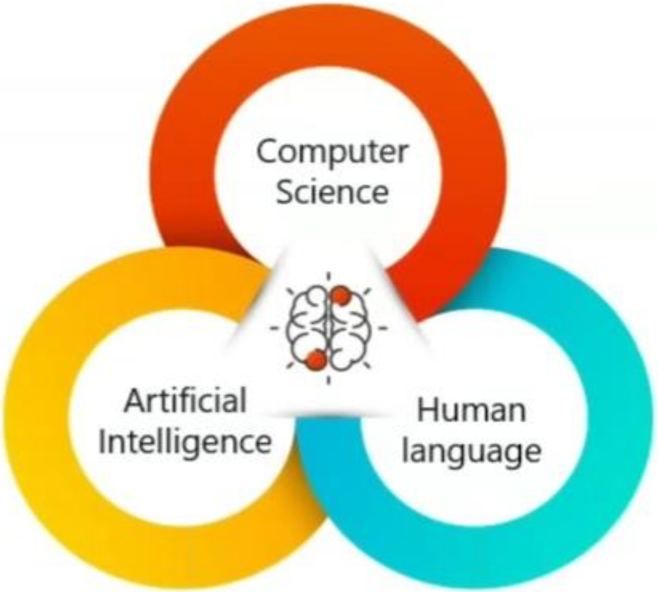
TEXT MINING AND NLP

Text Mining is
the process of deriving high quality information from the text .

The overall goal is, to turn text into data for analysis, via application of Natural Language Processing (NLP)

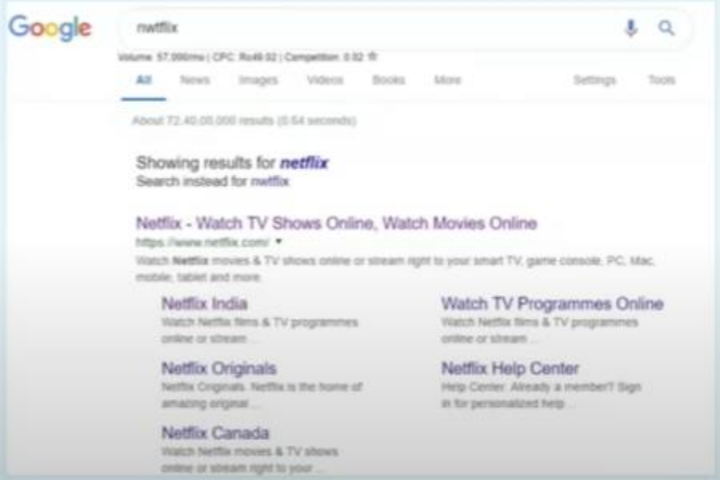
WHAT IS NLP?

Natural Language Processing is a part of computer science and artificial intelligence which deals with human languages.



WHERE IS TEXT MINING USED?

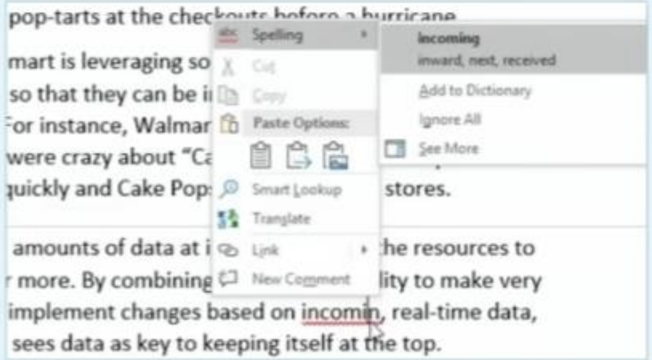
Autocomplete



Spam Detection

WHERE IS TEXT MINING USED?

Predictive typing



Spell checker

APPLICATIONS OF NLP

Sentimental Analysis

Chatbot

Speech Recognition

Machine Translation

www.e

APPLICATIONS OF NLP

Spell checking

Information extraction

Keyword search

Advertisement matching

TOKENIZATION

The process of splitting the whole data (corpus) into smaller chunks is known as tokenization

01 Break a complex sentence into words

02 Understand the importance of each of the words with respect to the sentence

03 Produce a structural description on an input sentence

TOKENIZATION

Tokens | are | simple

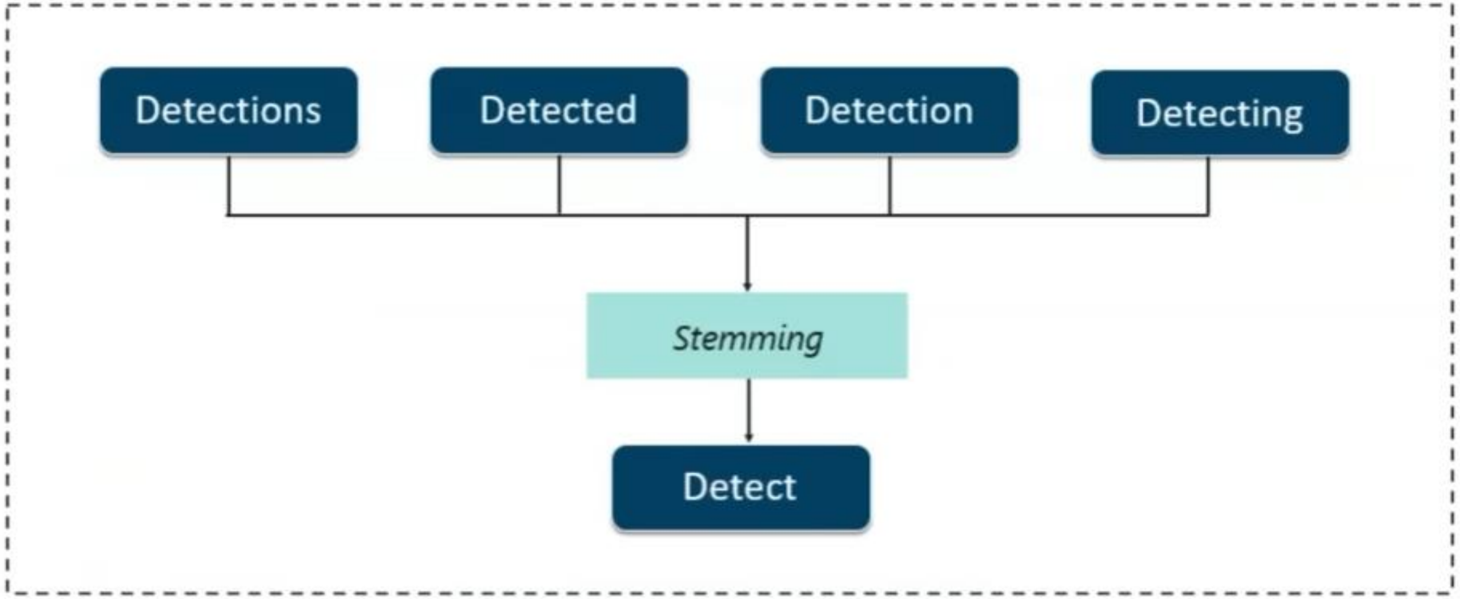
01 Break a complex sentence into words

02 Understand the importance of each of the words with respect to the sentence

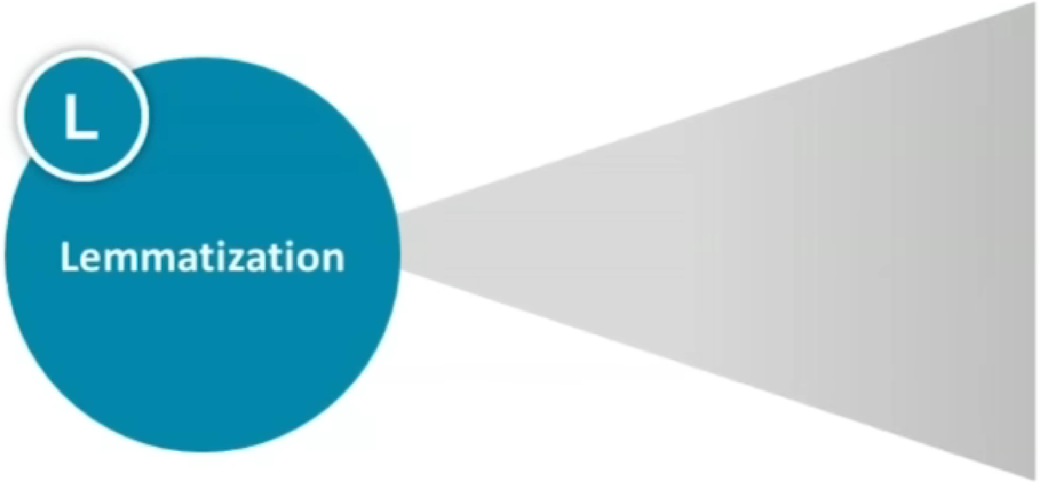
03 Produce a structural description on an input sentence

STEMMING

Normalize words into its base form or root form



LEMMITIZATION



- Groups together different inflected forms of a word, called Lemma
- Somehow similar to Stemming, as it maps several words into one common root
- Output of Lemmatisation is a proper word
- For example, a Lemmatiser should map ***gone***, ***going*** and went into ***go***

STOP WORDS

Are stop words helpful?

DOCUMENT TERM MATRIX

Documents

	This	is	fun
Doc 1	1	1	1
Doc 2	1	1	0
Doc 3	0	1	0
Doc 4	1	0	0

Document Term Matrix

Thank You


"Artificial intelligence will have a more profound impact on humanity than fire, electricity and the internet."
Sundar Pichai, the CEO of Alphabet

Resources

https://youtu.be/JMUxmLyrhSk