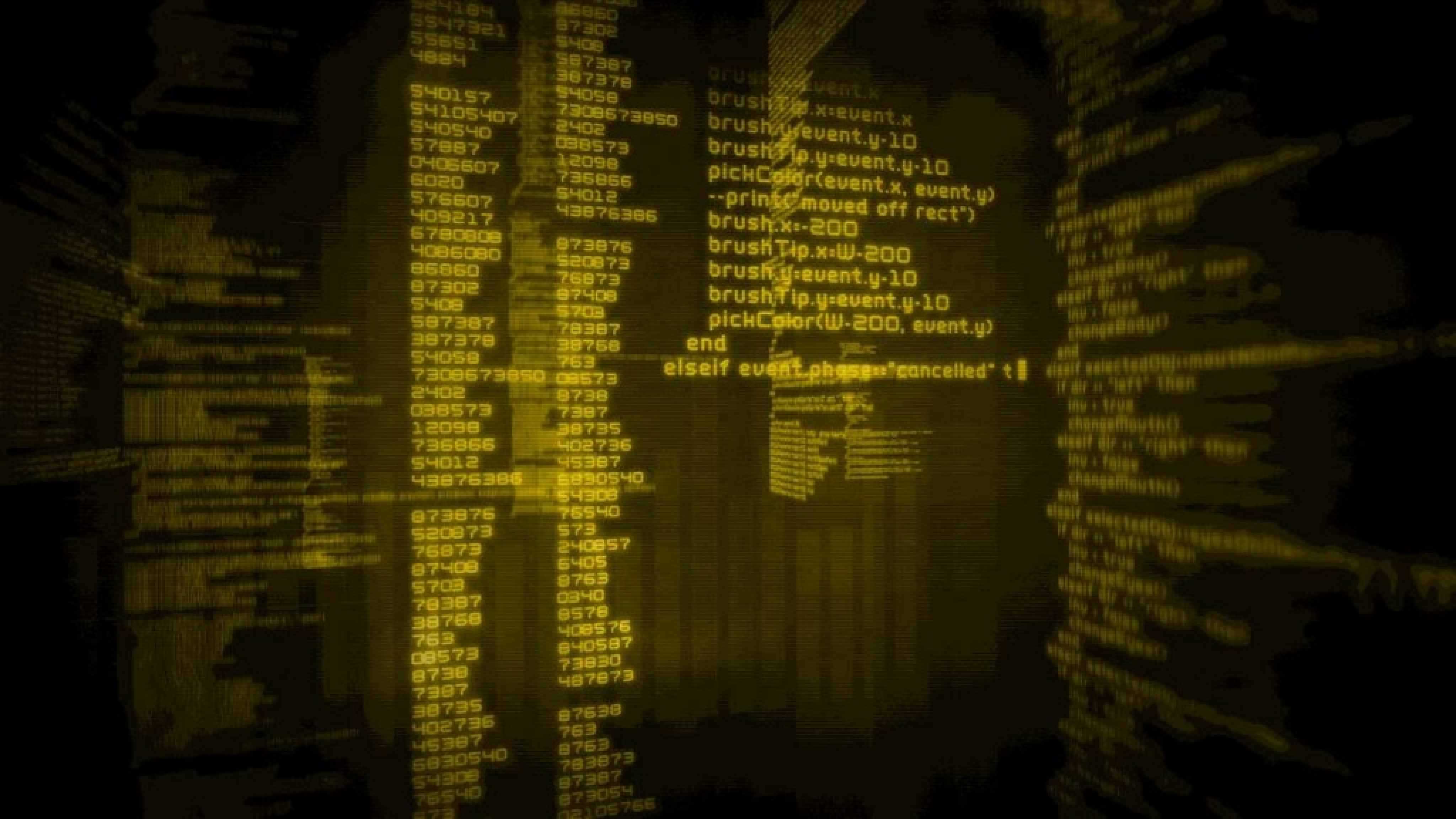
Machine Learning Full Course

edureka!

www.edureka.co/machine-learning-certification-training

Agenda

- Introduction To Data Science
- What is Machine Learning?
- Applications of Machine Learning
- Various Types Of Machine Learning
- Supervised Learning in Depth
- Unsupervised Learning in Depth
- Reinforcement Learning in Depth


brushTip.x=event.x
brush.y=event.y-10
brushTip.y=event.y-10
pickColor(event.x, event.y)
--print("moved off rect")
brush.x=-200
brushTip.x=W-200
brush.y=event.y-10
brushTip.y=event.y-10
pickColor(W-200, event.y)
end
elseif event.phase=="cancelled" t


Which Provisioner do you use the most ?
Shell
Ansible
Puppet (Agent, Apply)
Chef (Solo, Client, Zero, Apply)
Salt
Docker
Other

What is Data Science?

Everyday

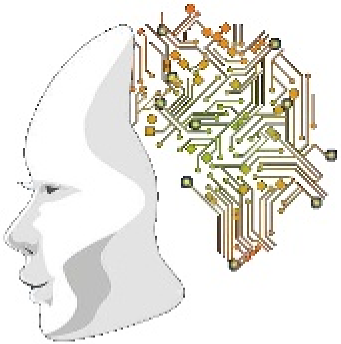
2.5×10^{18} Bytes

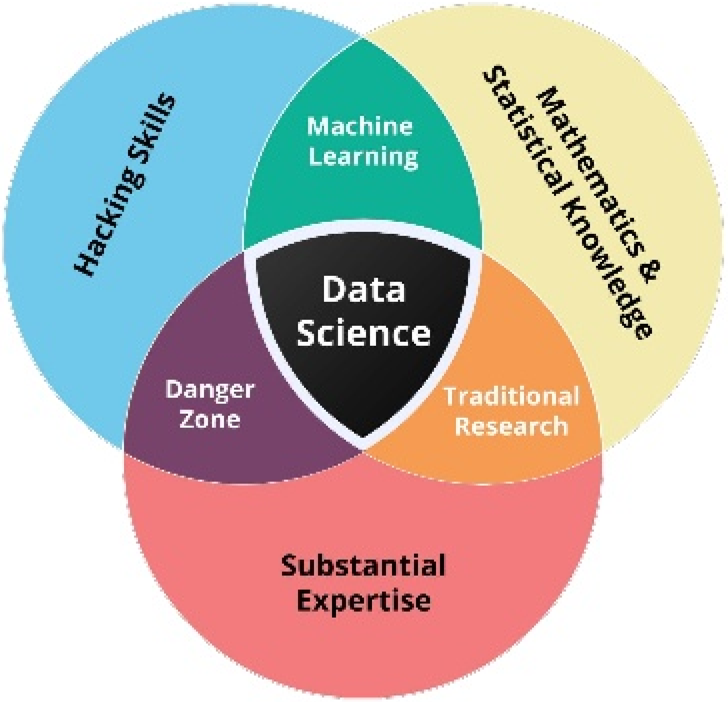
McK

The United States alone faces a shortage of 140,000 to 190,000 people with analytical expertise and 1.5 million managers and analysts with the skills to understand and make decisions based on the analysis of big data

What is Data Science?

Data science, also known as **data-driven science**, is an interdisciplinary field about scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured.





Data Science Peripherals

Statistics

Data Science Peripherals

Statistics

Prog Languages

Data Science Peripherals

Statistics

Prog Languages

Software

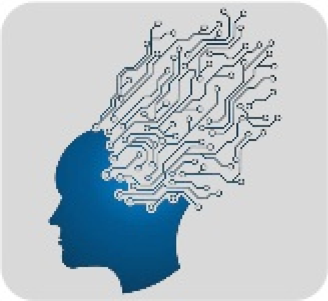
Data Science Peripherals

Statistics

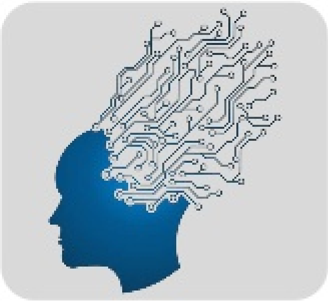
Prog Languages

Software

Machine Learning

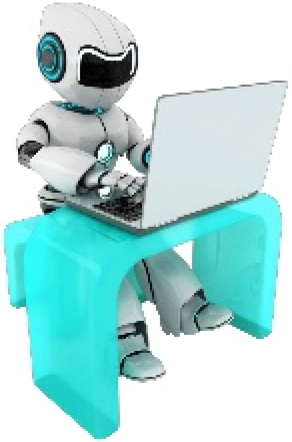


Data Science Peripherals

Statistics	Prog Languages	Software	Machine Learning	Big Data
				

What is Machine Learning?

Machine Learning is a class of algorithms which is data-driven, i.e. unlike "normal" algorithms it is the data that "tells" what the "good answer" is



Getting computers to program themselves and also teaching them to make decisions using data

“Where writing software is the bottleneck, let the data do the work instead.”

Features of Machine Learning

01

It uses the data to *detect patterns* in a dataset and *adjust program actions accordingly*

It *focuses on* the *development of computer programs* that can teach themselves to *grow and change* when *exposed* to *new data*

02

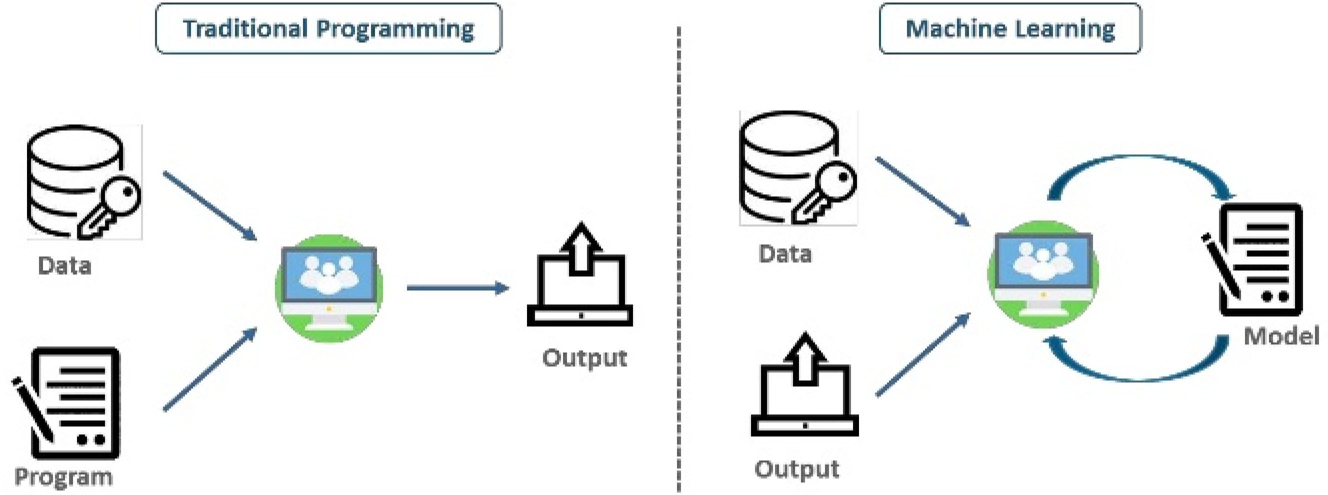
03

It enables computers to *find hidden insights using iterative algorithms without being explicitly programmed*

Machine learning is a *method* of *data analysis* that *automates analytical model building*

04

How It Works?

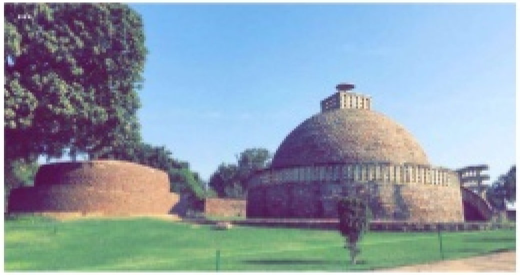


Learn from Data

Find Hidden Insights

Train and Grow

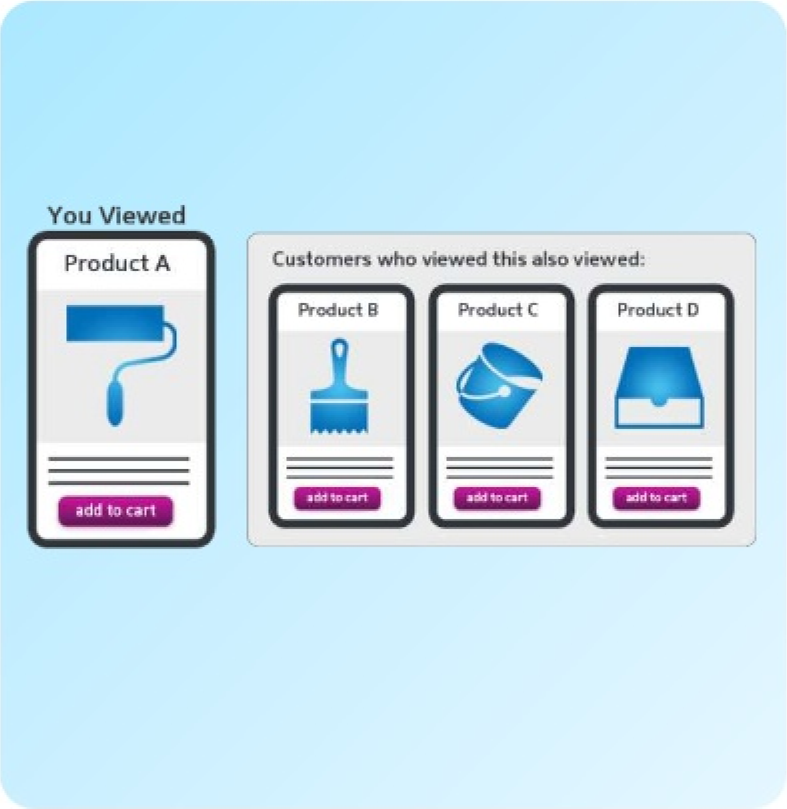
Applications of Machine Learning

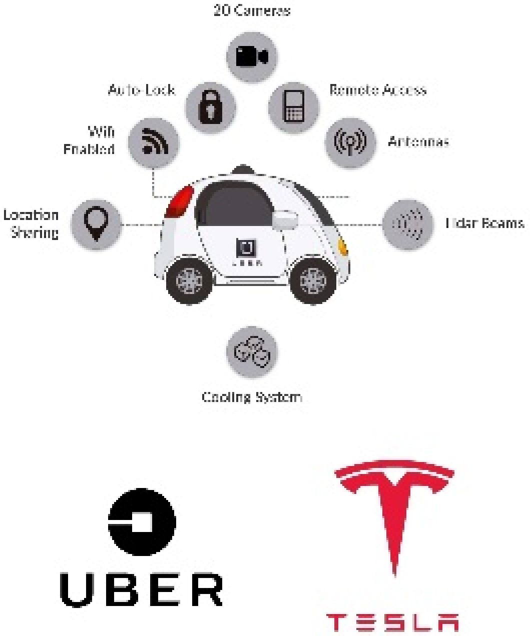


```
<img class="spotlight" alt="Image may contain: sky, grass, outdoor and nature">
```

UBER

Applications of Machine Learning





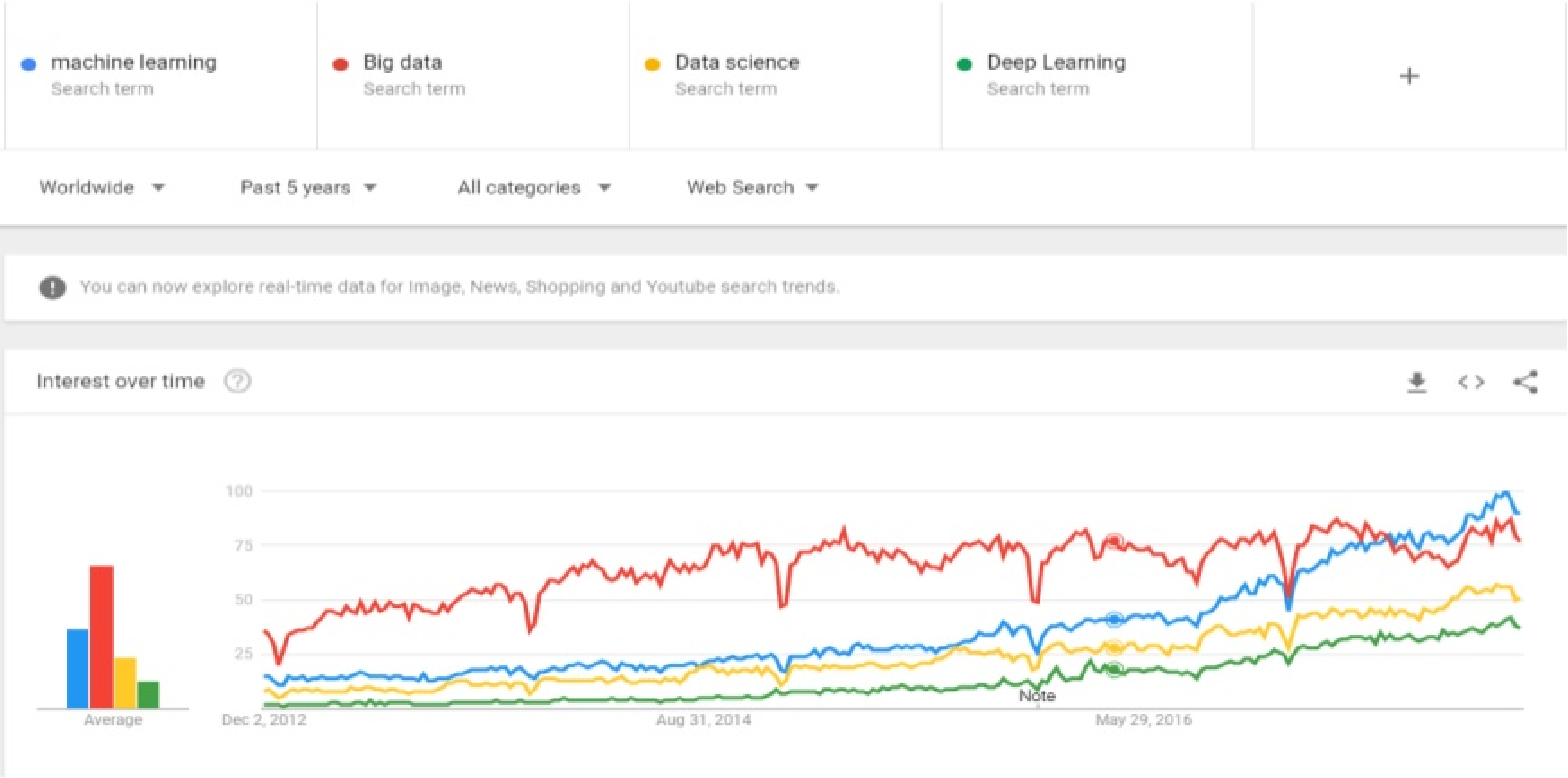
Applications of Machine Learning

NETFLIX

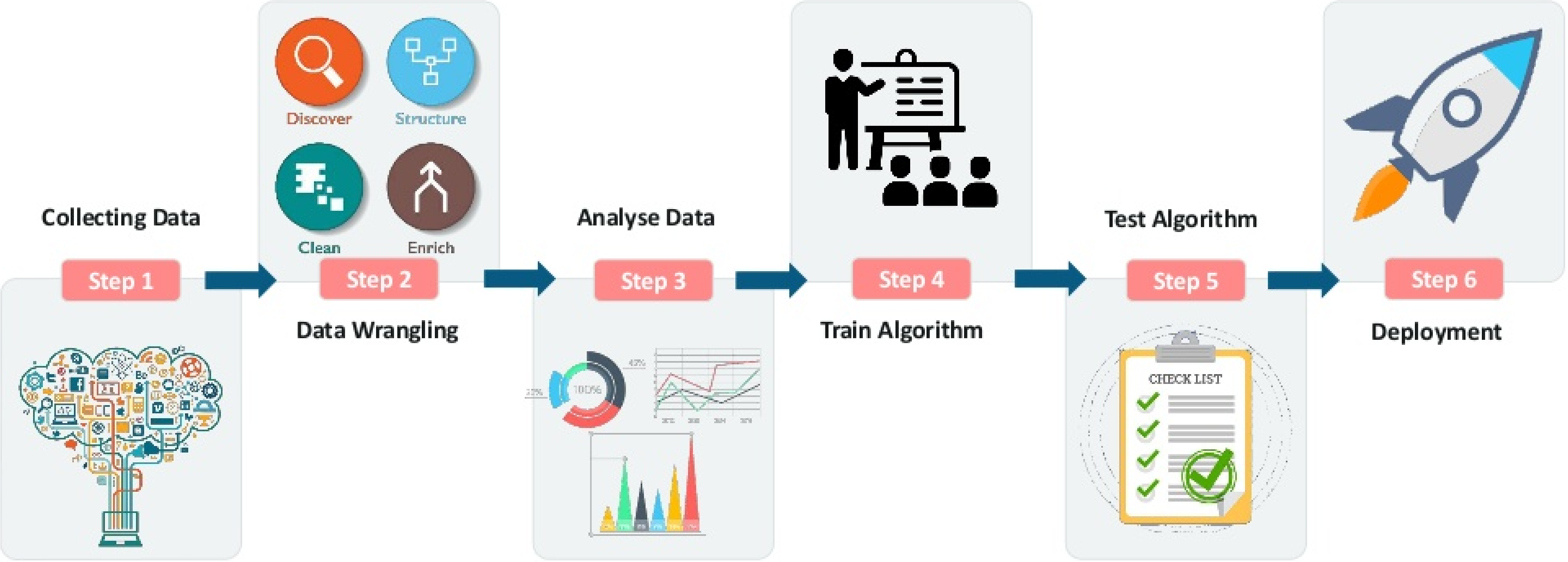
amazon video

hulu

Market Trend: Machine Learning

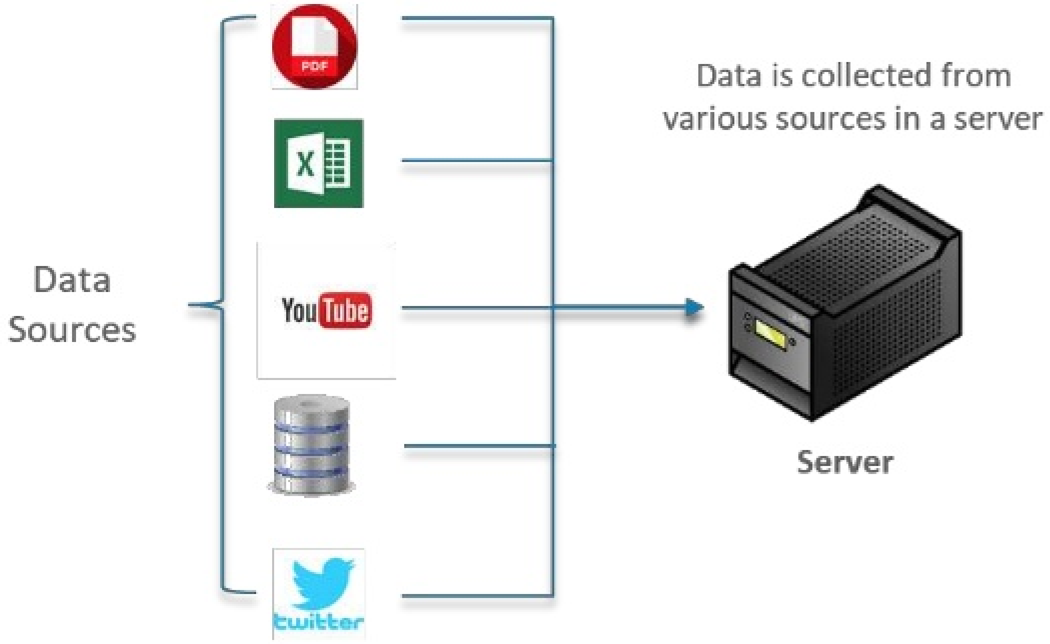


Machine Learning Life Cycle



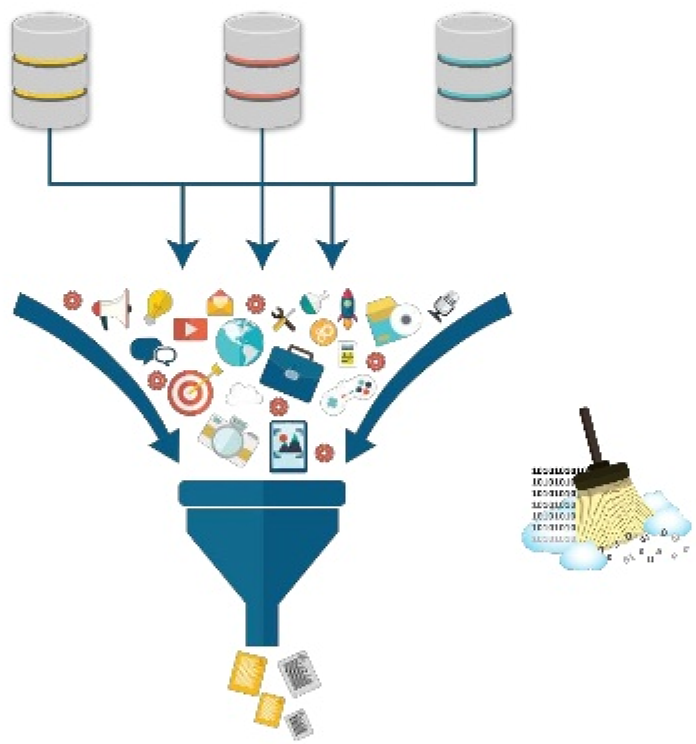
Step 1: Collecting Data

1
2
3
4
5
6



Data Wrangling

1
2
3
4
5
6



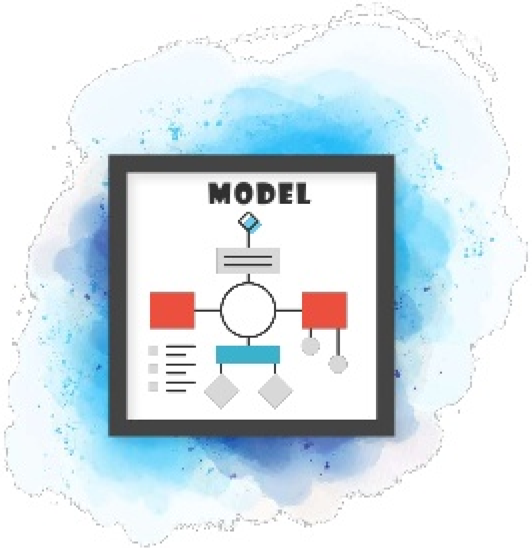
Data acquired from sources

Data filtering

Clean Data

Analyse Data

1
2
3
4
5
6



Train Algorithm

1

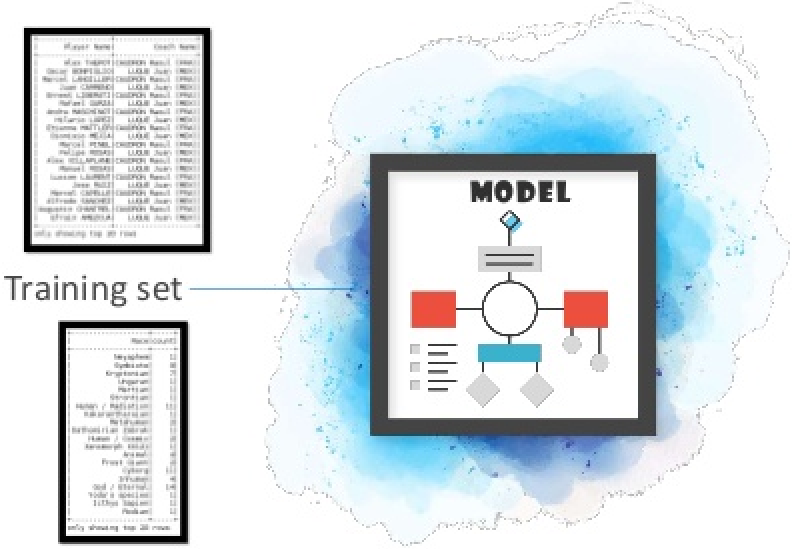
2

3

4

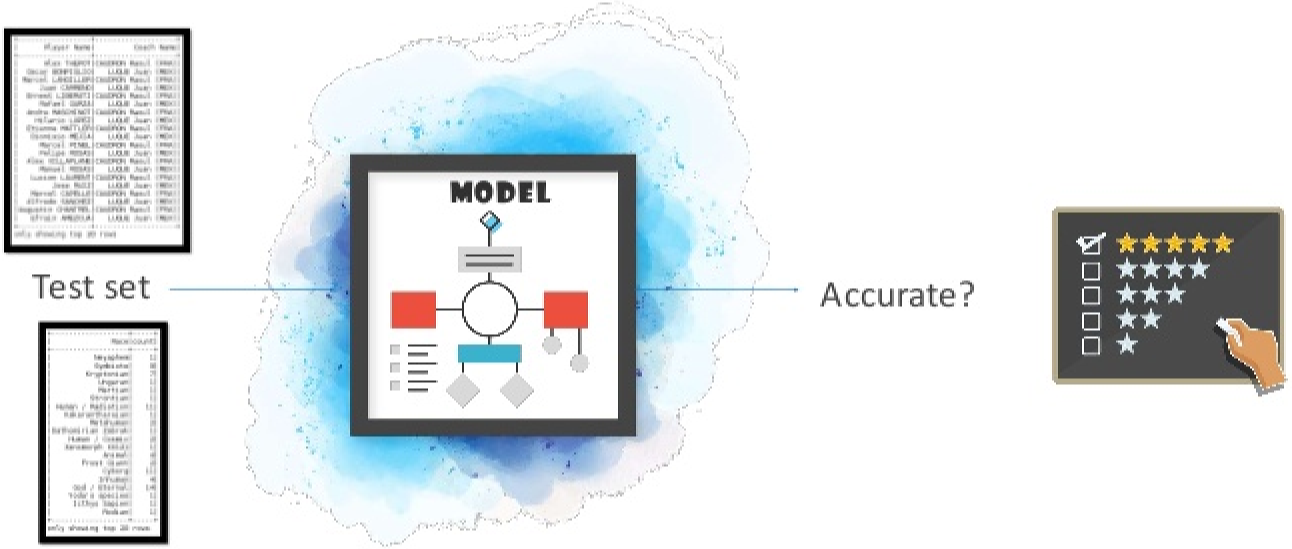
5

6



Test Algorithm

1
2
3
4
5
6



Operation and Optimization

1

2

3

4

5

6

Important Python Libraries

Scikit-Learn

Simple and efficient or data mining and data analysis, Built on NumPy and matplotlib, Open source

Seaborn

Focused on the visual of statistical models which include heat maps and depict the overall distributions

Pandas

Perfect tool for data wrangling, designed for quick and easy data manipulation, aggregation, and visualization

Matplotlib

It enables you to make-
Bar charts, Scatter plots, Line Charts, Histograms, Pie charts, Contour plots, Quiver plots

Numpy

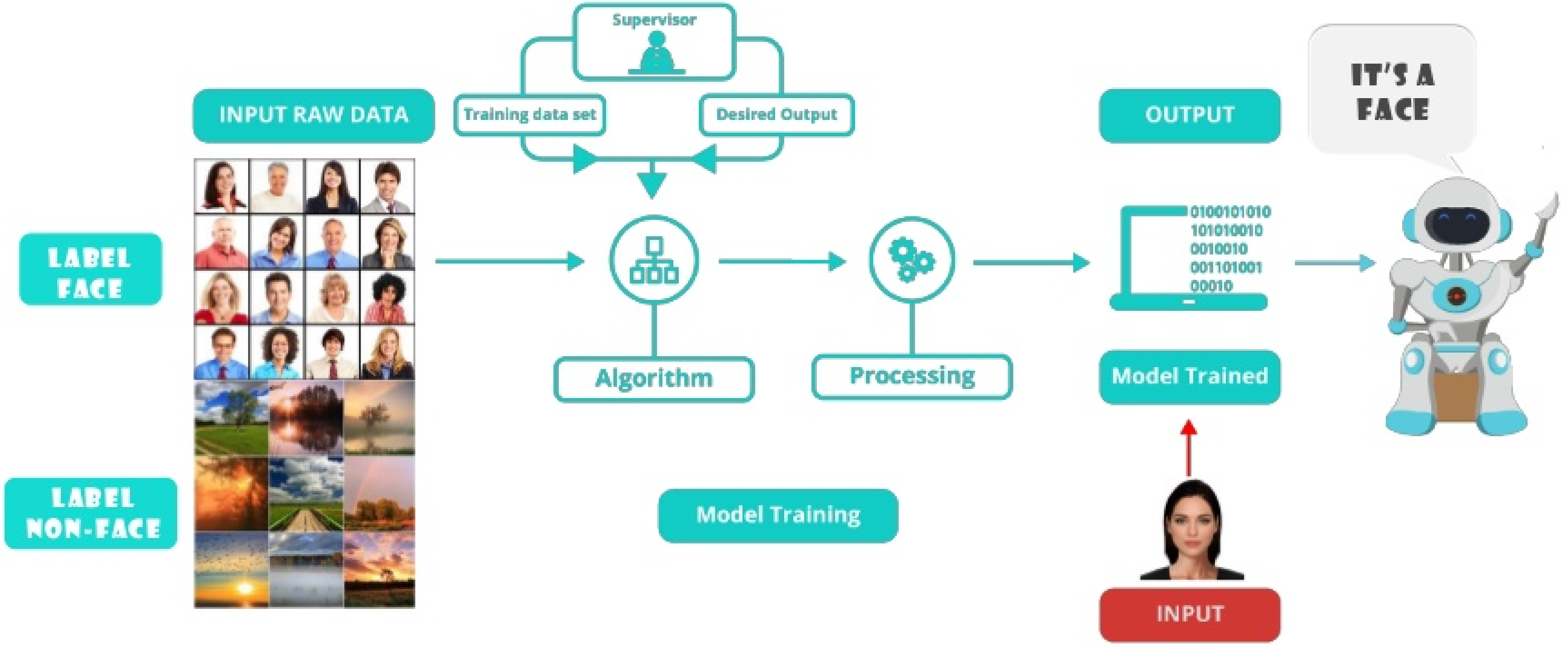
Stands for Numerical Python, provides an abundance of useful features for operations on n-arrays and matrices in Python

Types of Machine Learning

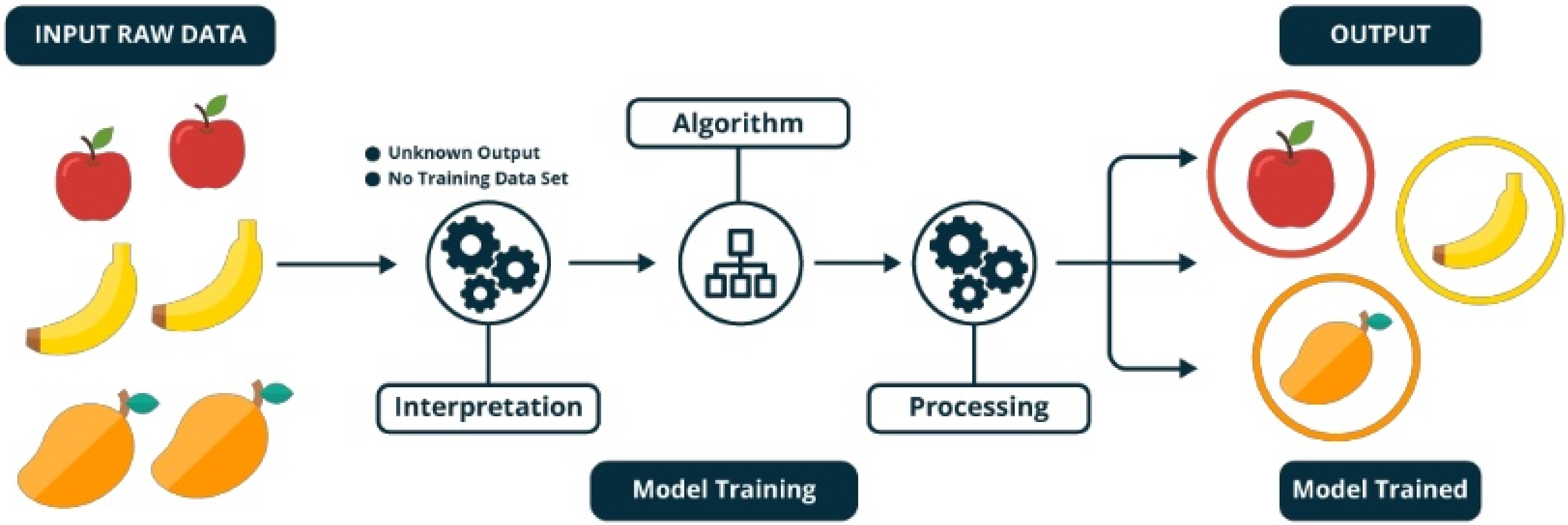
Types of Learning

- Supervised Learning
- Reinforcement Learning
- Unsupervised Learning

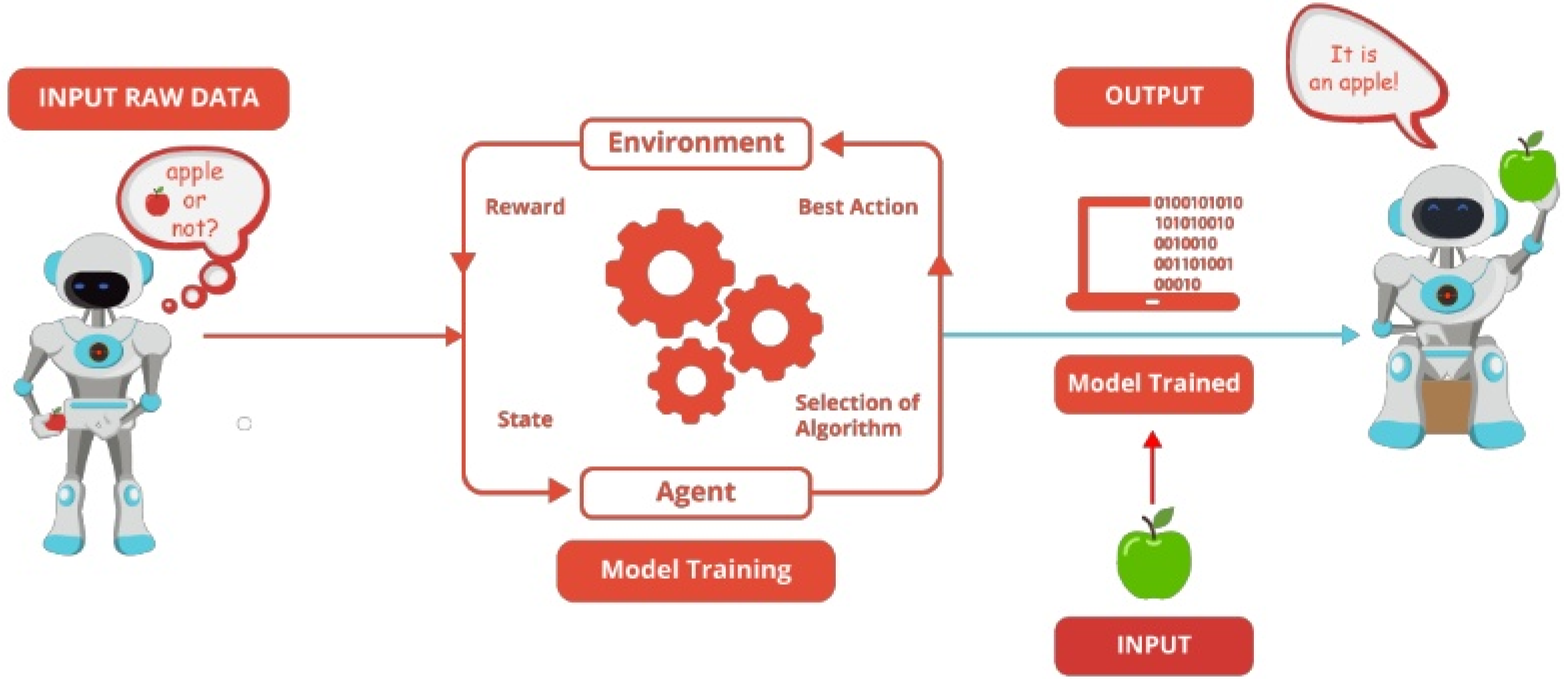
Supervised Learning



Unsupervised Learning

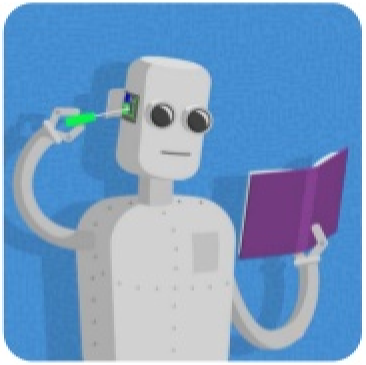


Reinforcement Learning



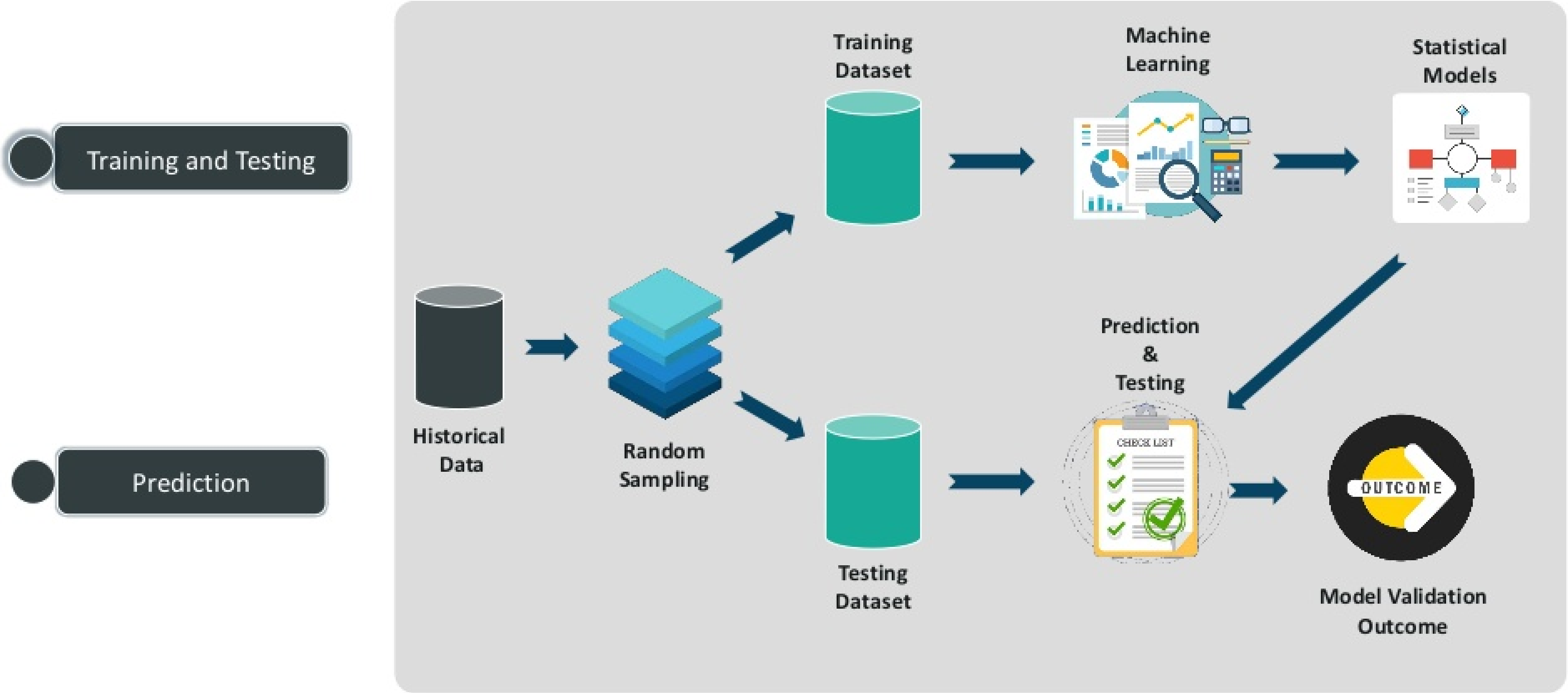
Supervised Learning

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output



It is called Supervised Learning because the process of an algorithm learning from the training dataset can be thought as a teacher supervising the learning process

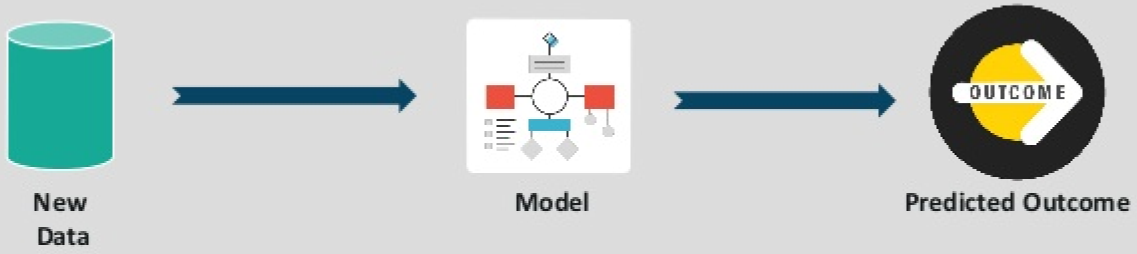
Supervised Learning



Supervised Learning

Training and Testing

Prediction



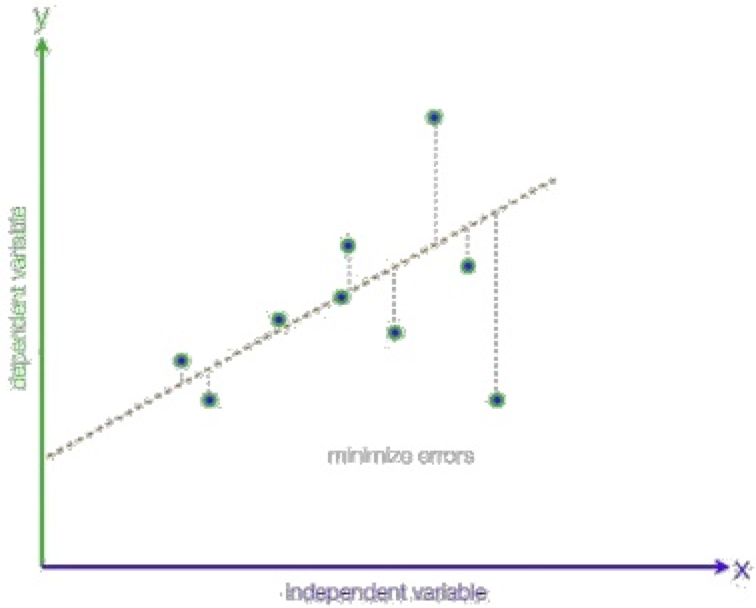
Supervised Learning Algorithms

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Naïve Bayes Classifier

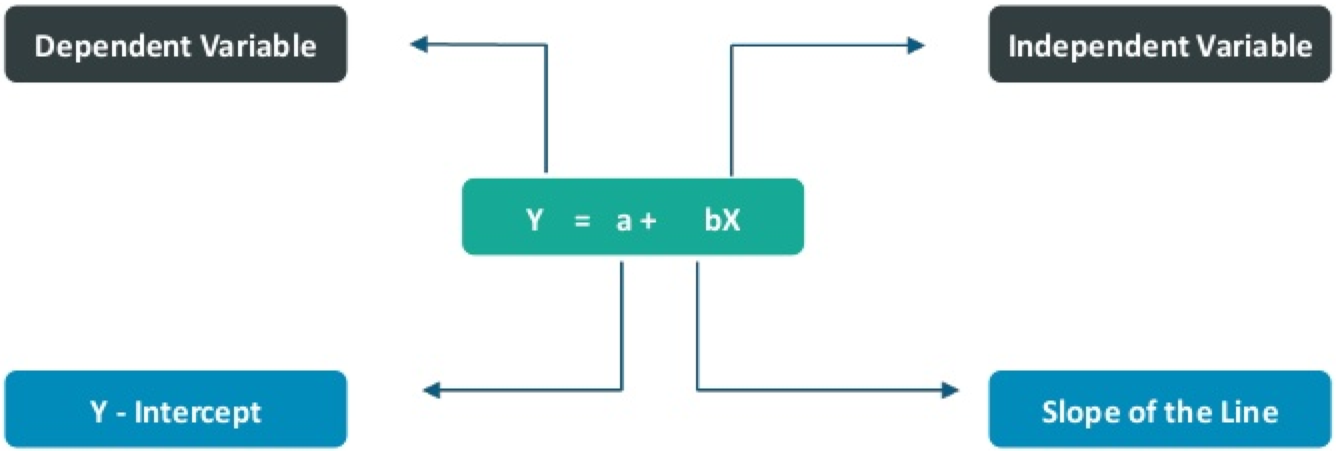
Linear Regression

Linear Regression Analysis is a powerful technique used for predicting the unknown value of a variable (**Dependent Variable**) from the known value of another variables (**Independent Variable**)

- A **Dependent Variable(DV)** is the variable to be predicted or explained in a regression model
- An **Independent Variable(IDV)** is the variable related to the dependent variable in a regression equation

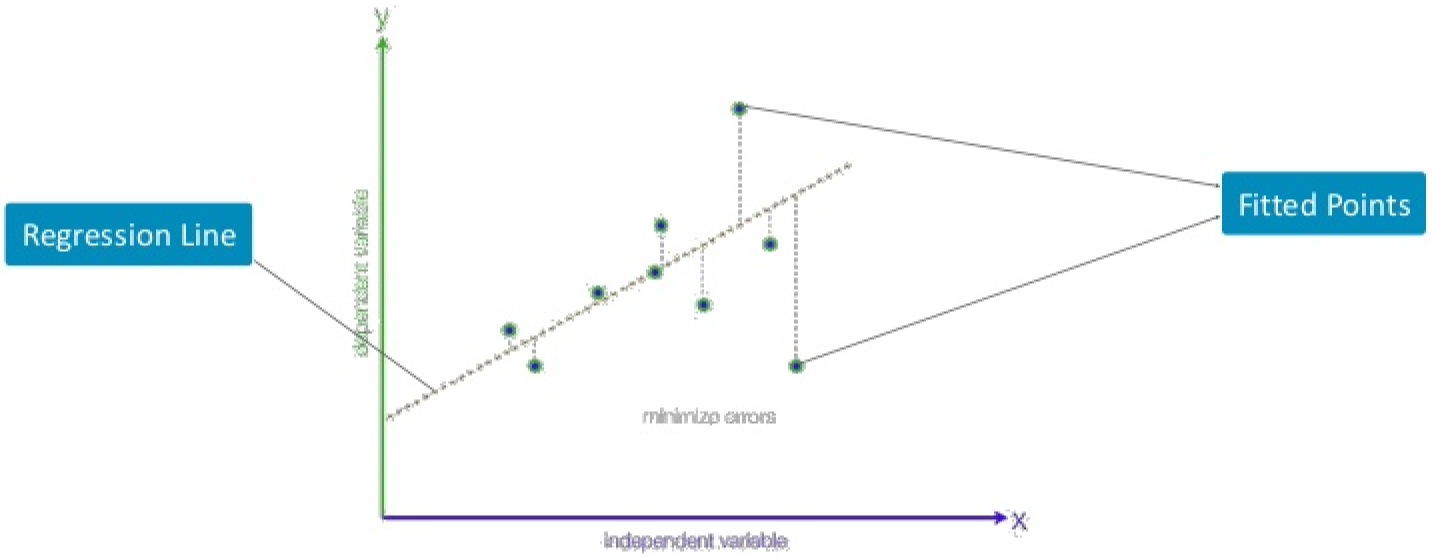


Simple Linear Regression



Regression Line

Linear Regression Analysis is a powerful technique used for predicting the unknown value of a variable (**Dependent Variable**) from The regression line is simply a single line that best fits the data (In terms of having the smallest overall distance from the line to the points)



Demo

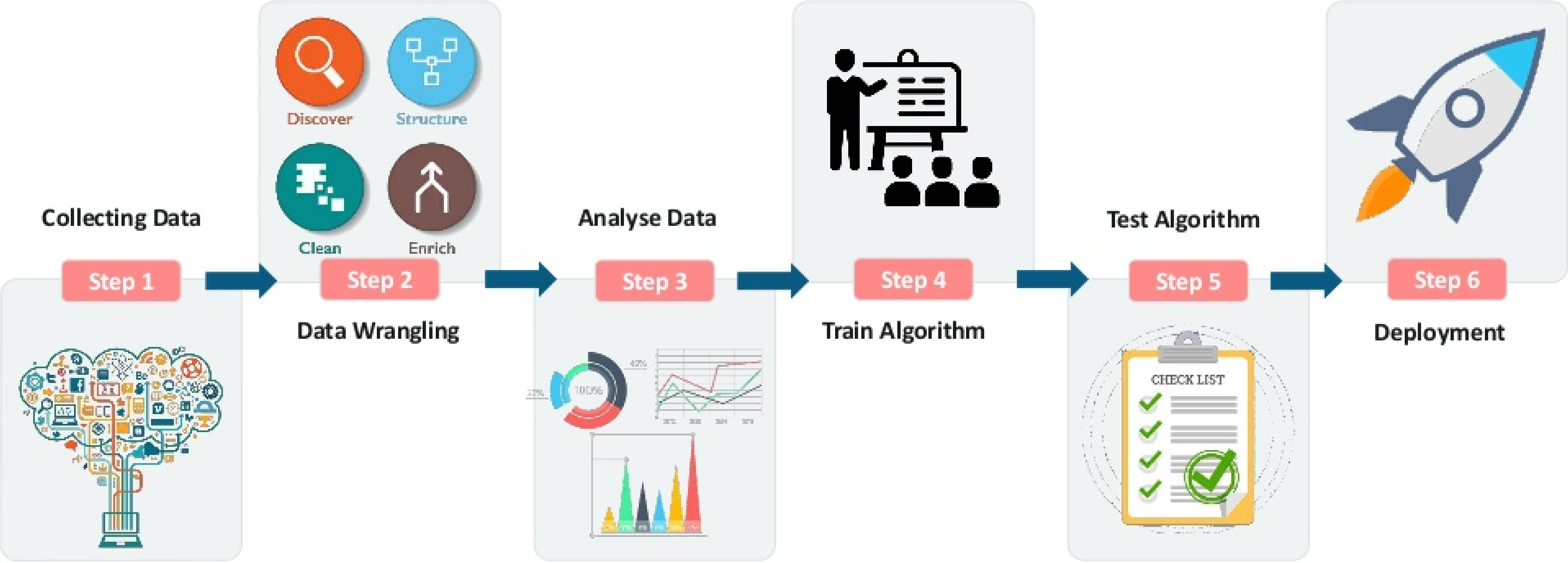
Real Estate Company Use Case

Hi I am john, I need some baseline for pricing my Villas and Independent Houses

Dataset Description

Column	Description
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

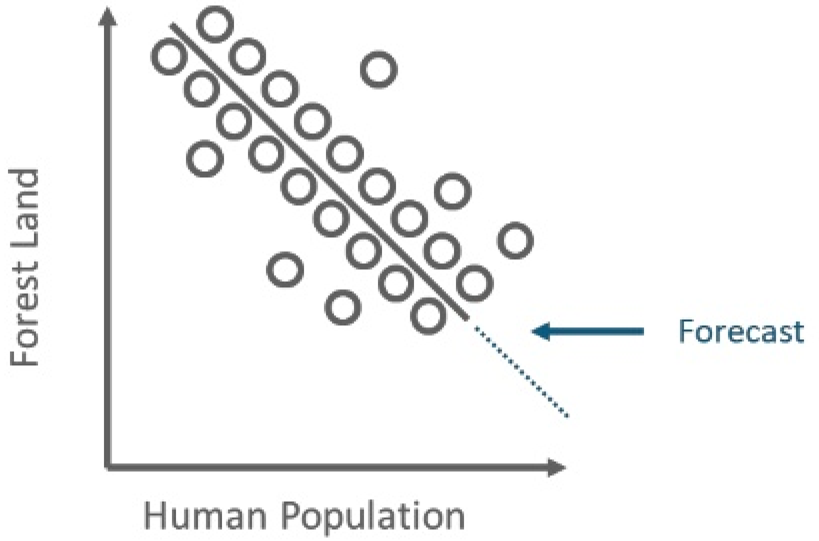
Steps


Collecting Data
Step 1
Discover
Structure
Clean
Enrich
Step 2
Data Wrangling
Analyse Data
Step 3
Step 4
Train Algorithm
Test Algorithm
Step 5
CHECK LIST
Step 6
Deployment

Model Fitting

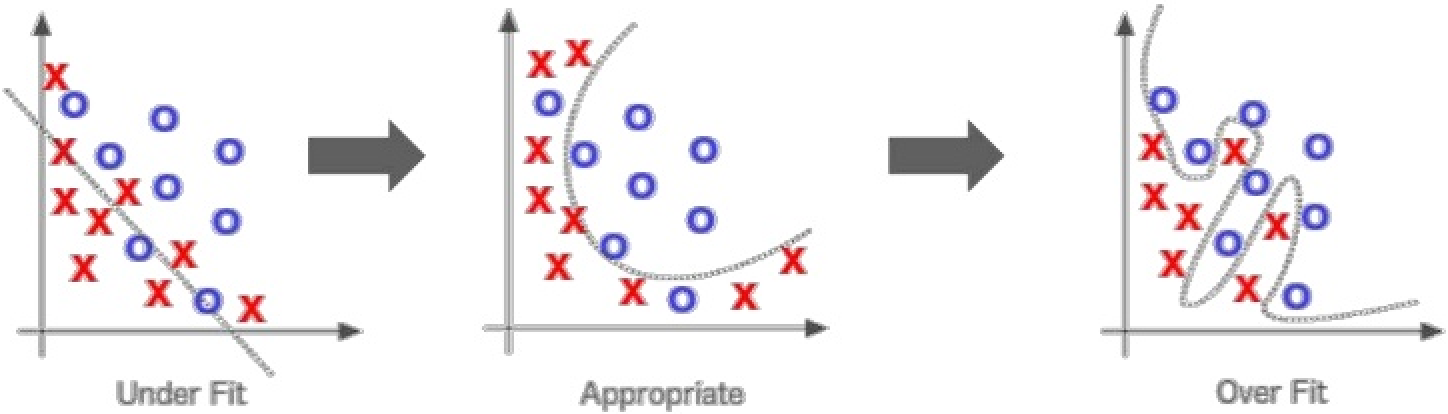
Fitting a model means that you're making your algorithm learn the relationship between predictors and outcome so that you can predict the future values of the outcome .

So the best fitted model has a specific set of parameters which best defines the problem at hand



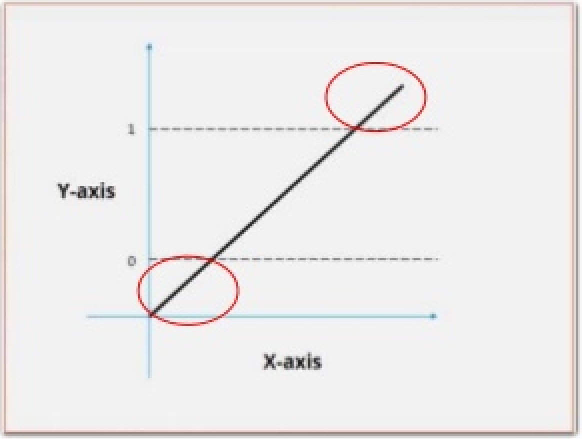
Types of Fitting

Machine Learning algorithms first attempt to solve the problem of under-fitting; that is, of taking a line that does not approximate the data well, and making it to approximate the data better.



Need For Logistic Regression

Here, the best fit line in linear regression is going below 0 and above 1



WHO WILL WIN ?

What is Logistic Regression?

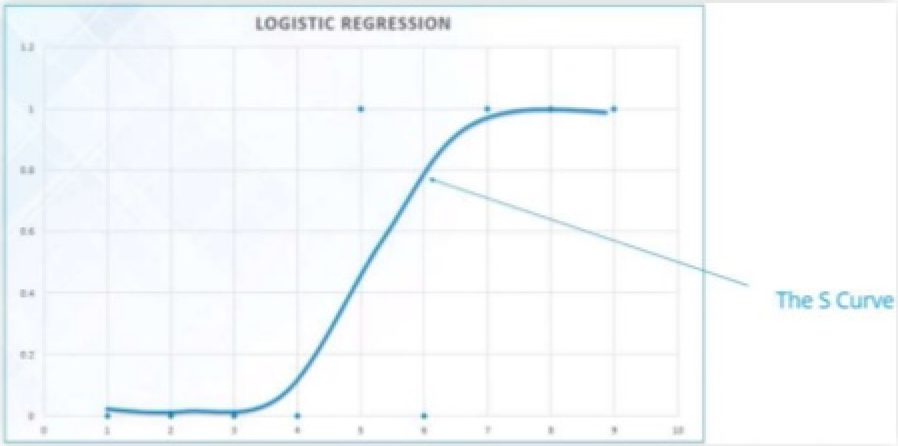
Logistic Regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome.
Outcome is a binary class type.

The outcome(result) will be binary(0/1)

0- If malignant
1- If benign

What is Logistic Regression?

The Logistic Regression Curve is called as "Sigmoid Curve", also known as S-Curve

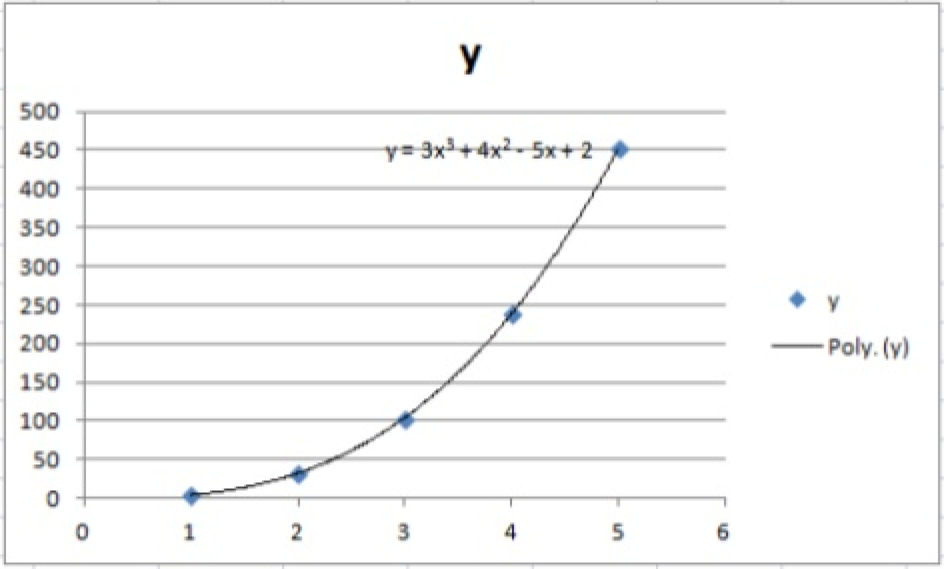


Based on the threshold value set, we decide the output from the function

What is Polynomial Regression?

When we have non linear data, which can't be predicted with a linear model. We switch to Polynomial Regression.

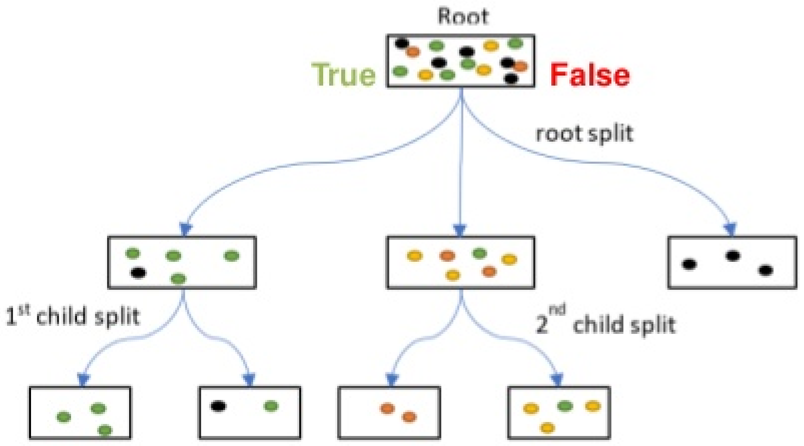
Such a scenario is shown in the below graph



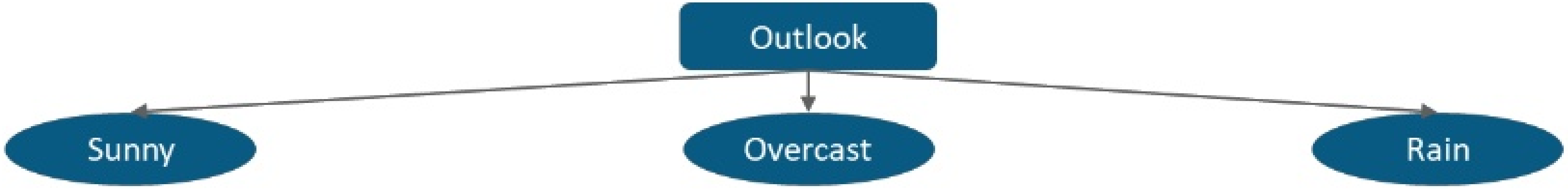
What is a Decision Tree?

A decision tree is a tree-like structure in which internal node represents test on an attribute

- Each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes)
- A path from root to leaf represents classification rules.



Building a Decision Tree



Day	Outlook	Humidity	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

2 Yes / 3 No

Split further

Day	Outlook	Humidity	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

Pure subset

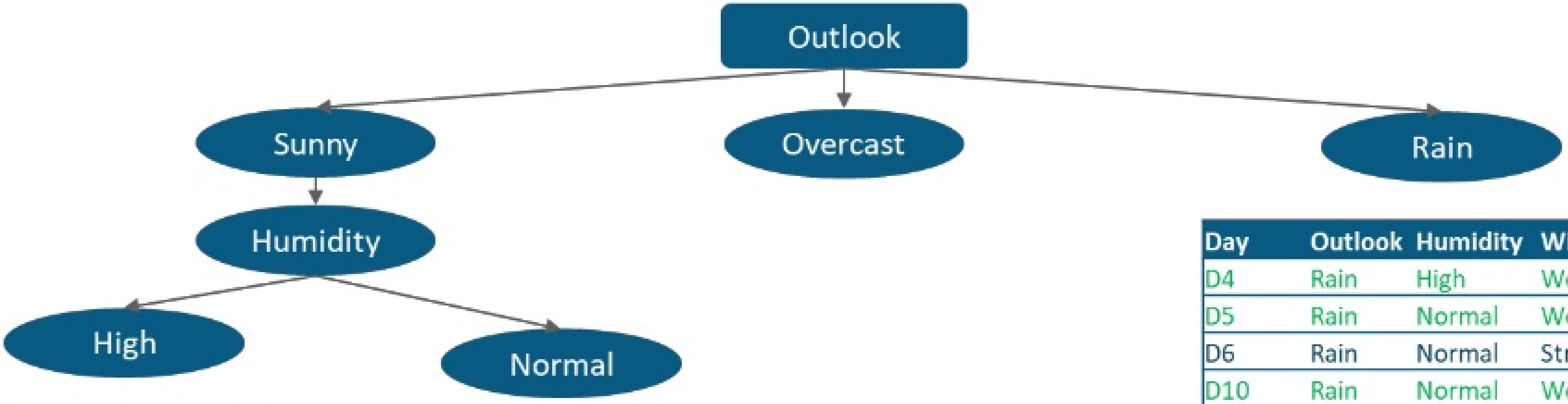
Yes(will play)

Day	Outlook	Humidity	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 Yes / 2 No

Split further

Building a Decision Tree



Day	Humidity	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Pure subset
NO(will not play)

Day	Humidity	Wind
D9	Normal	Weak
D11	Normal	Strong

Pure subset

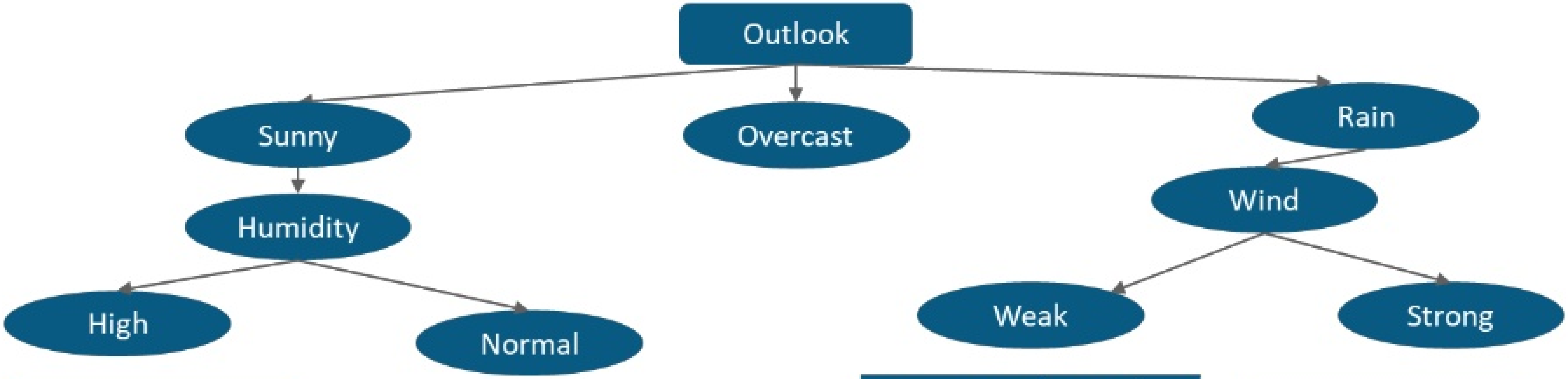
Yes(will play)

Day	Outlook	Humidity	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 Yes / 2 No

Split further

Building a Decision Tree



Day	Humidity	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Pure subset
NO(will not play)

Day	Humidity	Wind
D9	Normal	Weak
D11	Normal	Strong

Pure subset
Yes(will play)

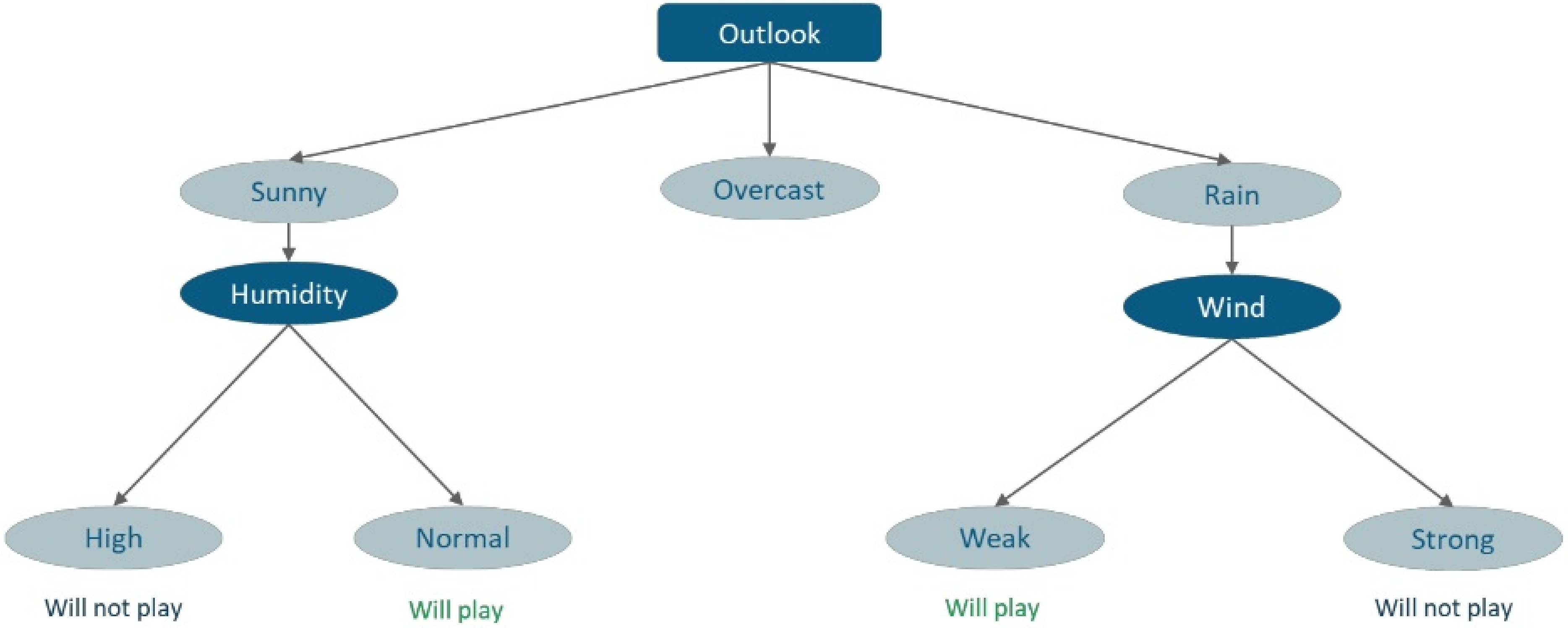
Day	Humidity	Wind
D4	High	Weak
D5	Normal	Weak
D10	Normal	Weak

Pure subset
Yes(will play)

Day	Humidity	Wind
D6	Normal	Strong
D14	High	Strong

Pure subset
NO(will not play)

Building a Decision Tree

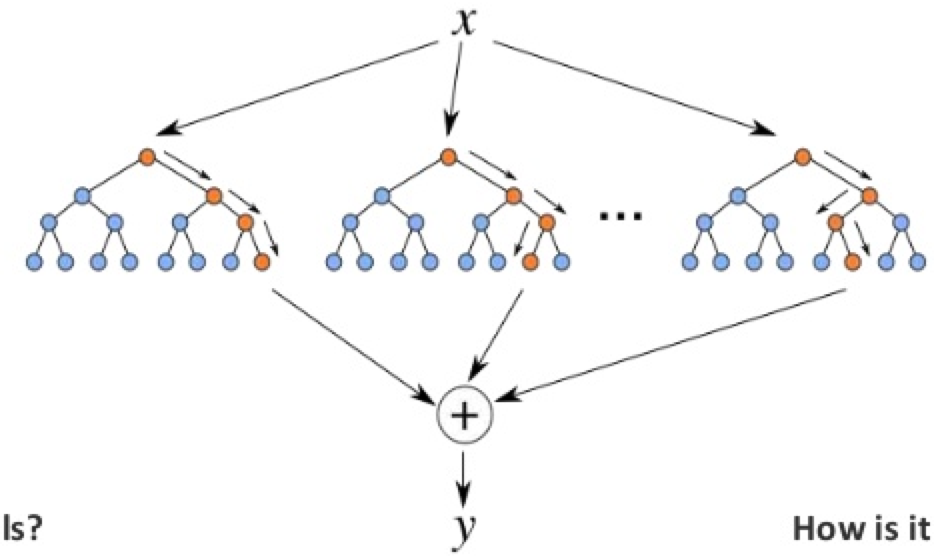


Demo

edureka!

What is Random Forest?

Random Forest is an ensemble classifier made using **many Decision tree models**



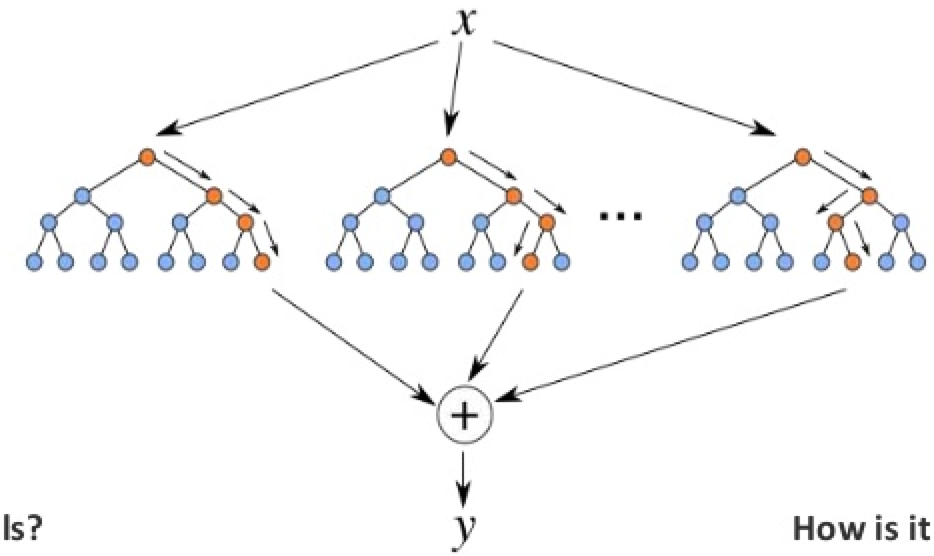
What are Ensemble models?

How is it better from Decision Trees ?

Demo

What is Naïve Bayes?

Random Forest is an ensemble classifier made using **many Decision tree models**



What are Ensemble models?

How is it better from Random Forest?

What is Naïve Bayes ?

Naive Bayes is a simple but surprisingly powerful algorithm
for predictive modeling.

Classification Technique

Bayes' Theorem

Given a hypothesis H and evidence E , Bayes' theorem states that the relationship between the probability of the hypothesis before getting the evidence P(H) and the probability of the hypothesis after getting the evidence P(H|E) is

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

Bayes' Theorem Example

Bayes' Theorem Example

P(King) = 4/52 =1/13

Bayes' Theorem Example

$$P(King) = 4/52 = 1/13$$

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

Bayes' Theorem Example

P(King) = 4/52 =1/13

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

P(Face|King) = 1

Bayes' Theorem Example

$$P(King) = 4/52 = 1/13$$

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

$$P(Face|King) = 1$$

$$P(Face) = 12/52 = 3/13$$

Bayes' Theorem Example

$$P(King) = 4/52 = 1/13$$

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

$$P(Face|King) = 1$$

$$P(Face) = 12/52 = 3/13$$

$$= \frac{1.(1/13)}{3/13} = 1/3$$

Bayes' Theorem Example

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

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

$$P(A \cap B) = P(A|B).P(B) = P(B|A).P(A)$$

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

Bayes' Theorem Proof

Likelihood
How probable is the evidence
Given that our hypothesis is true?

Prior
How probable was our hypothesis
Before observing the evidence?

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

Posterior
How probable is our Hypothesis
Given the observed evidence?
(Not directly computable)

Marginal
How probable is the new evidence
Under all possible hypothesis?

Naïve Bayes: Working

Classification Steps

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Frequency Table		Play	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	3	2

Frequency Table		Play	
		Yes	No
Humidity	High	3	4
	Normal	6	1

Frequency Table		Play	
		Yes	No
Wind	Strong	6	2
	Weak	3	3

Classification Steps

Likelihood Table		Play: Yes	Play: No	
Outlook	Sunny	3/10	2/4	5/14
	Overcast	4/10	0/4	4/14
	Rainy	3/10	2/4	5/14
		10/14	4/14	

$P(x|c) = P(Sunny|Yes) = 3/10 = 0.3$

$P(x) = P(Sunny) = 5/14 = 0.36$

$P(c) = P(Yes) = 10/14 = 0.71$

Likelihood of 'Yes' given Sunny is

$P(c|x) = P(Yes|Sunny) = P(Sunny|Yes)^* \; P(Yes) \; / \; P(Sunny)$ = (0.3 x 0.71) /0.36 = 0.591

Similarly Likelihood of 'No' given Sunny is

$P(c|x) = P(No|Sunny) = P(Sunny|No)^* \; P(No) \; / \; P(Sunny)$ = (0.4 x 0.36) /0.36 = 0.40

Classification Steps

Likelihood table for Humidity

Likelihood Table		Play: Yes	Play: No	
Humidity	High	3/9	4/5	7/14
	Normal	6/9	1/5	7/14
		9/14	5/14	

Likelihood table for Wind

Likelihood Table		Play: Yes	Play: No	
Wind	Weak	6/9	2/5	8/14
	Strong	3/9	3/5	6/14
		9/14	5/14	

$P(Yes|High)$ = 0.33 x 0.6 / 0.5 = 0.42

$P(No|High)$ = 0.8 x 0.36 / 0.5 = 0.58

$P(Yes|Weak)$ = 0.67 x 0.64 / 0.57 = 0.75

$P(No|Weak)$ = 0.4 x 0.36 / 0.57 = 0.25

Classification Steps

Suppose we have a day with the following values

Outlook	=	Rain
Humidity	=	High
Wind	=	Weak
Play	=	?

Likelihood of 'Yes' on that Day = P(Outlook = Rain|Yes)*P(Humidity= High|Yes)* P(Wind= Weak|Yes)*P(Yes)

= 2/9 * 3/9 * 6/9 * 9/14 = 0.0199

Likelihood of 'No' on that Day = P(Outlook = Rain|No)*P(Humidity= High|No)* P(Wind= Weak|No)*P(No)

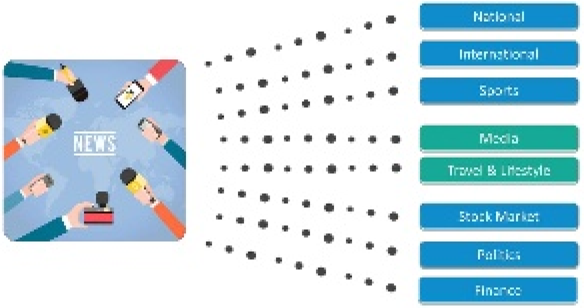
= 2/5 * 4/5 * 2/5 * 5/14 = 0.0166

Classification Steps

P(Yes) = 0.0199 / (0.0199+ 0.0166) = 0.55

P(No) = 0.0166 / (0.0199+ 0.0166) = 0.45

Industrial Use Cases



News Categorization

Weather Predictions

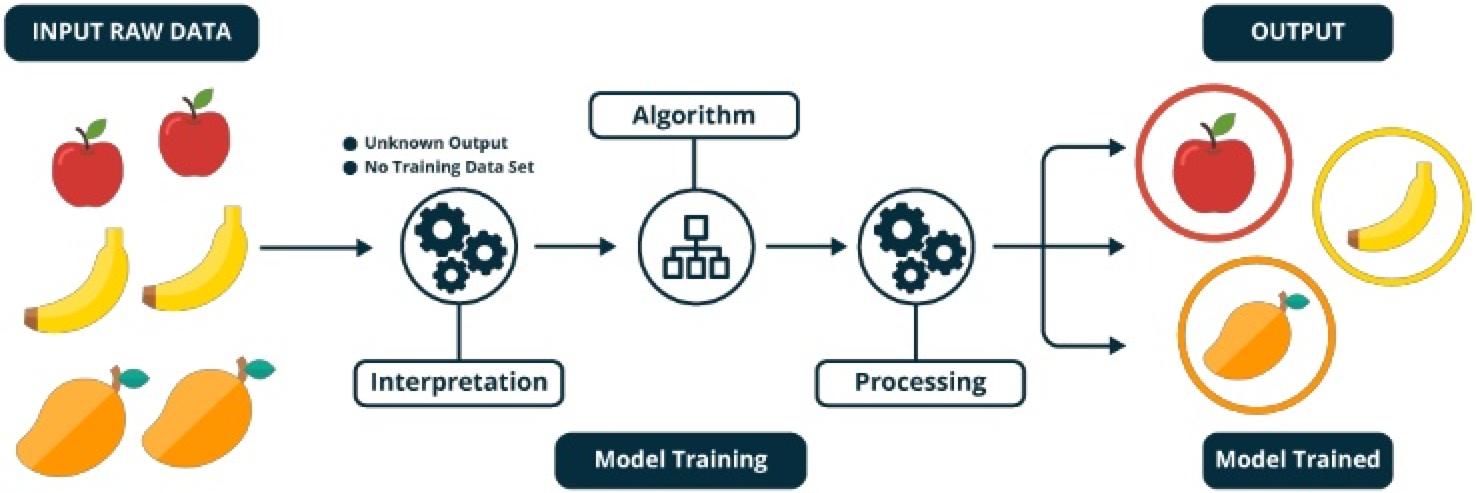
Spam Filtering

Demo

edureka!

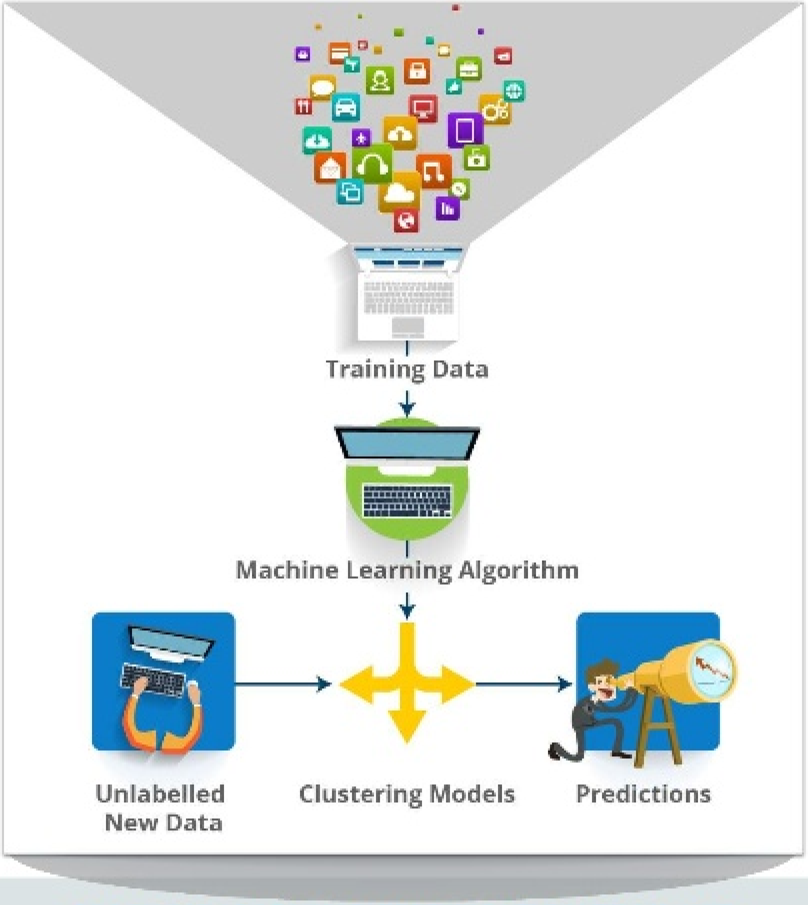
Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data **Without labelled responses**



Unsupervised Learning: Process Flow

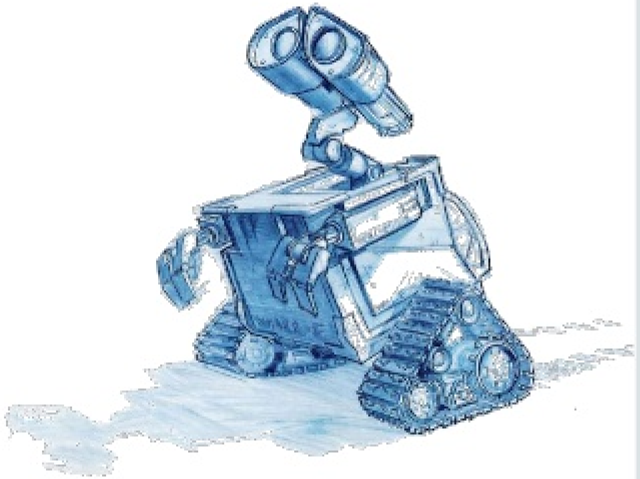
Training data is collection of information without any label



What is Clustering?

"**Clustering** is the process of dividing the datasets into groups, consisting of similar data-points"

It means grouping of objects based on the information found in the data, describing the objects or their relationship



Why is Clustering Used?

The goal of clustering is to determine the intrinsic grouping in a set of **Unlabelled Data**

Sometimes, Partitioning is the goal

Where is it used?

Retail Store

Banking

Insurance Companies

amazon

NETFLIX

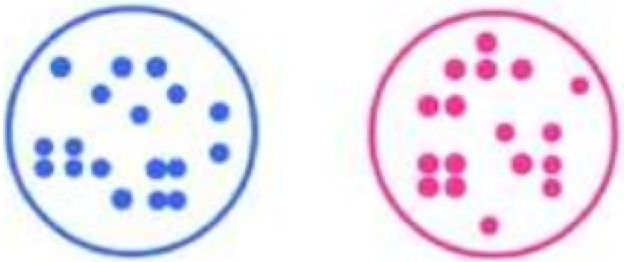
Recommended Movies

flickr

Flickr's Photos

Types of Clustering

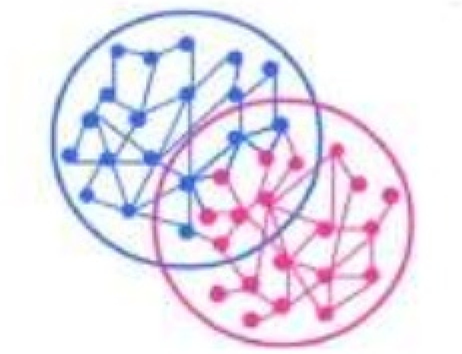
- **Exclusive Clustering**
- Overlapping Clustering
- Hierarchical Clustering



K-Means Clustering

Types of Clustering

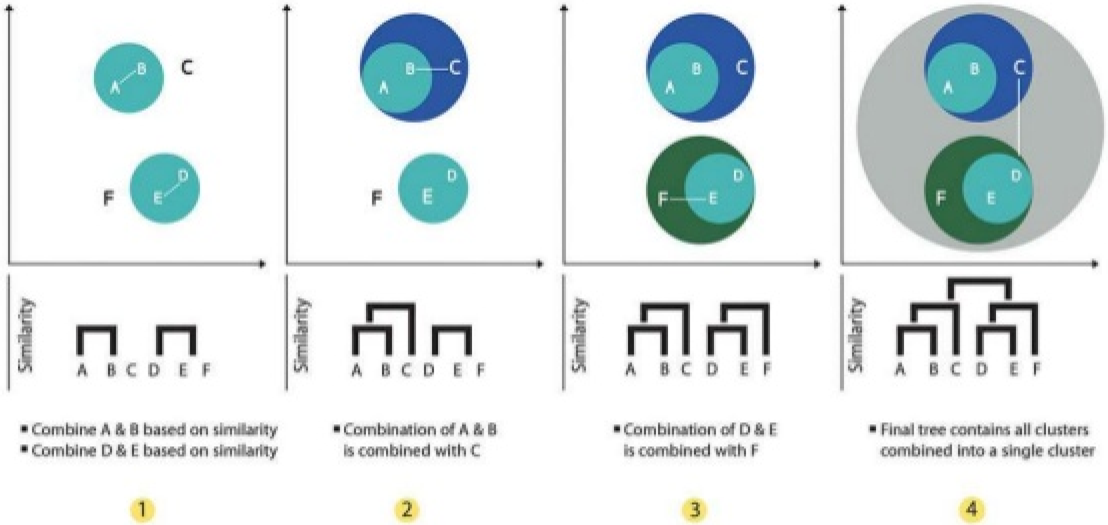
- Exclusive Clustering
- **Overlapping Clustering**
- Hierarchical Clustering



C-Means Clustering

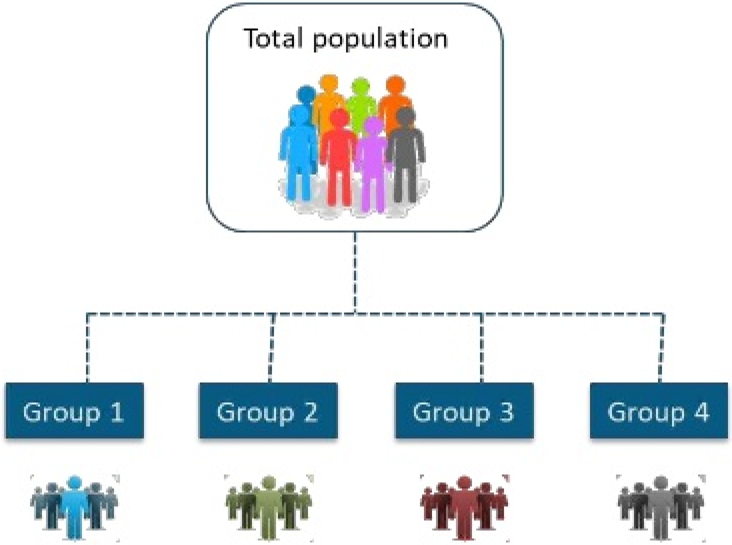
Types of Clustering

- Exclusive Clustering
- Overlapping Clustering
- **Hierarchical Clustering**

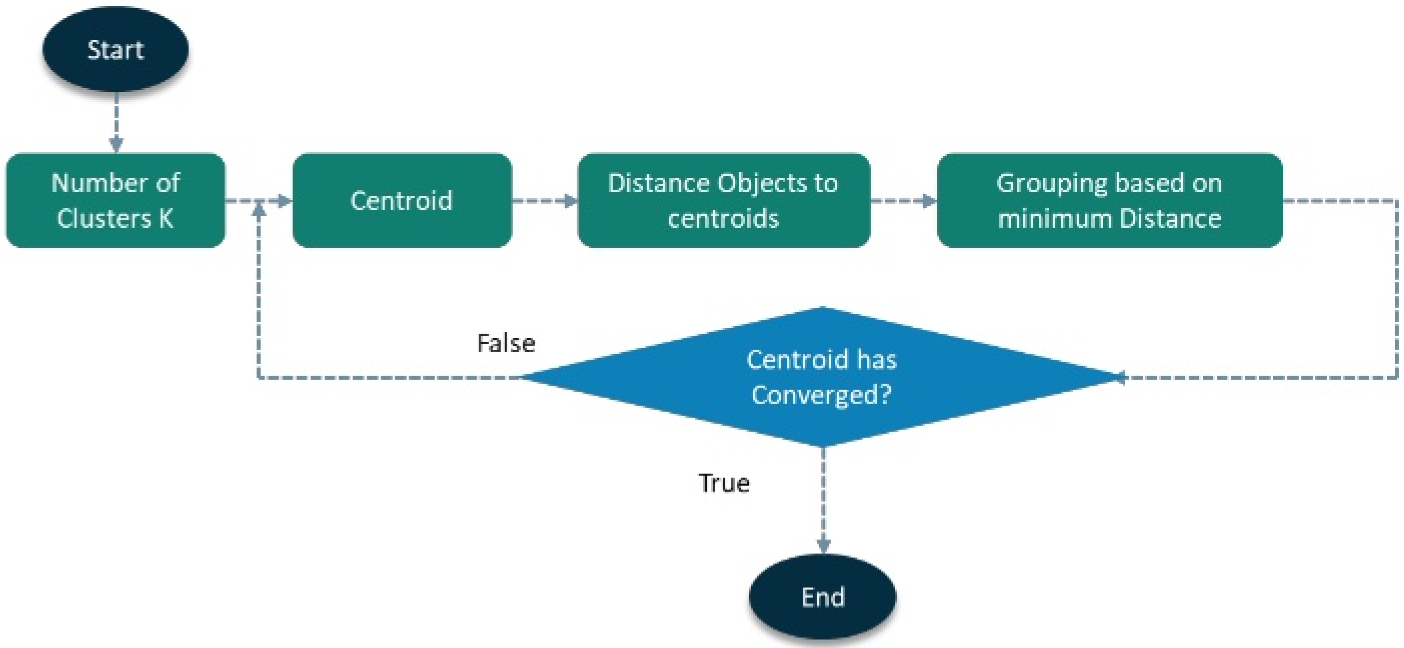


K-Means Clustering

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 Algorithm Working



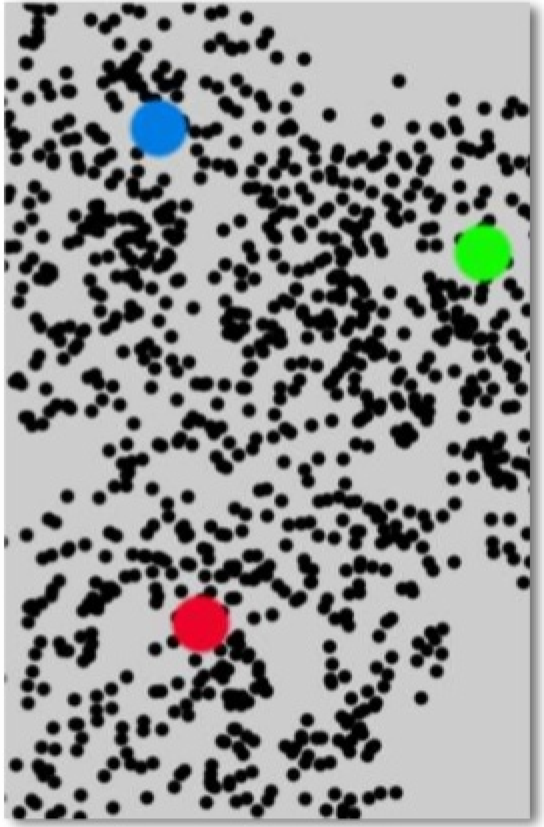
K-Means Clustering : Steps

1. First we need to decide the number of clusters to be made. (Guessing)

Let's assume , Number of clusters = 3

K-Means Clustering : Steps

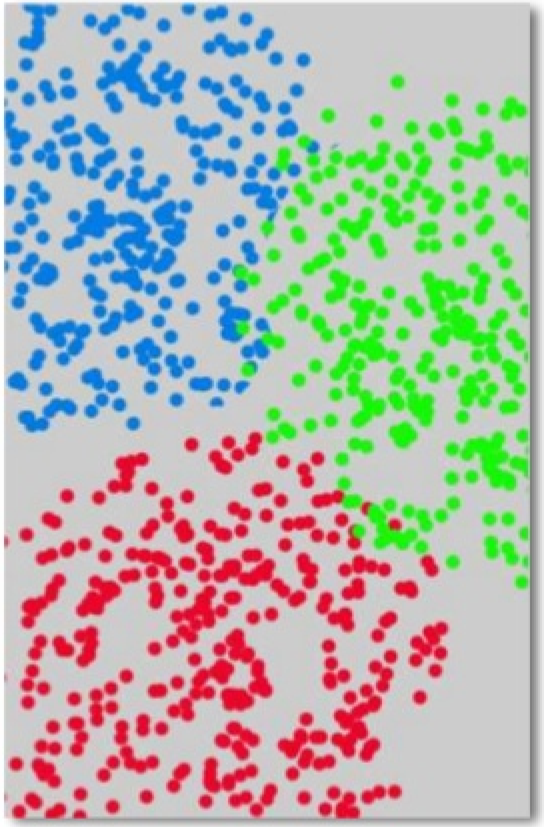
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)



edureka!

K-Means Clustering : Steps

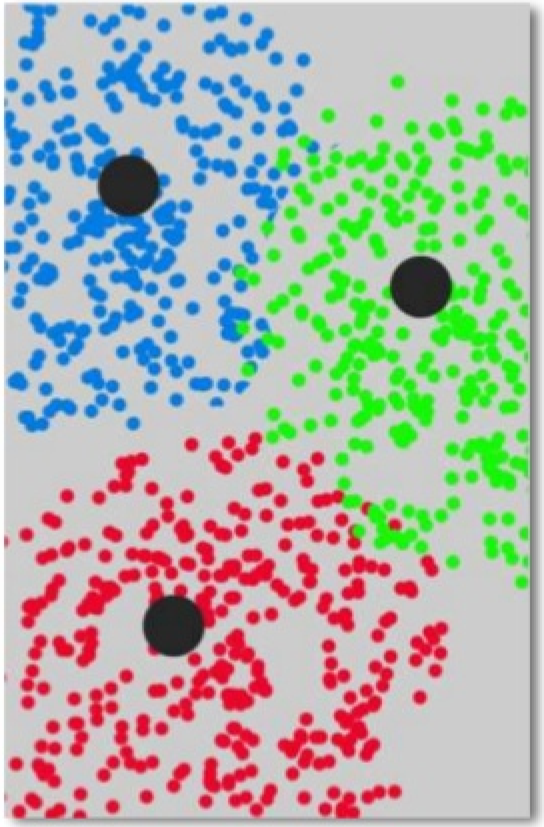
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assigns the point to the closest cluster.



edureka!

K-Means Clustering : Steps

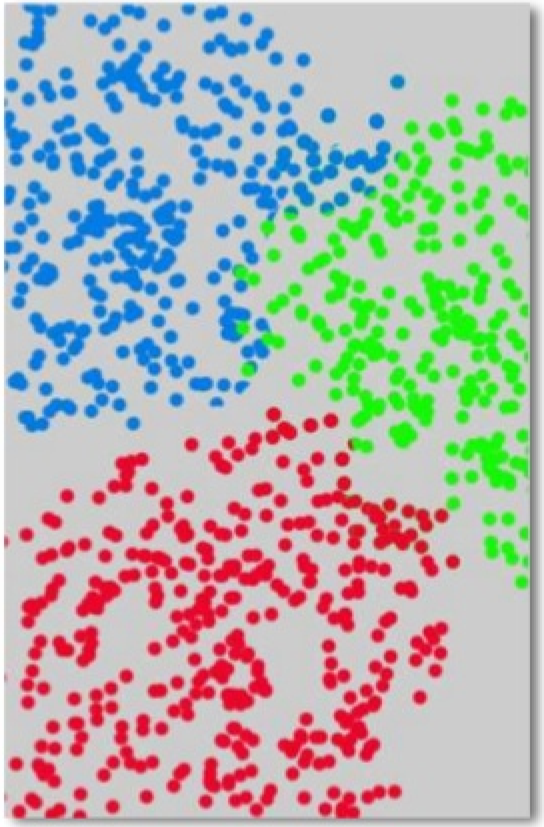
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assigns the point to the closest cluster.
4. Next the Centroids are calculated again, when we have our new cluster.



edureka!

K-Means Clustering : Steps

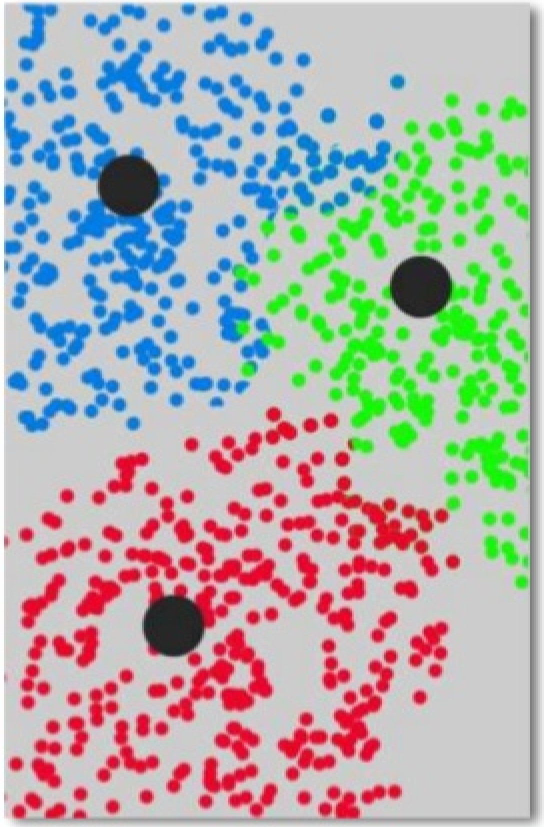
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assigns the point to the closest cluster.
4. Next the Centroids are calculated again, 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.



edureka!

K-Means Clustering : Steps

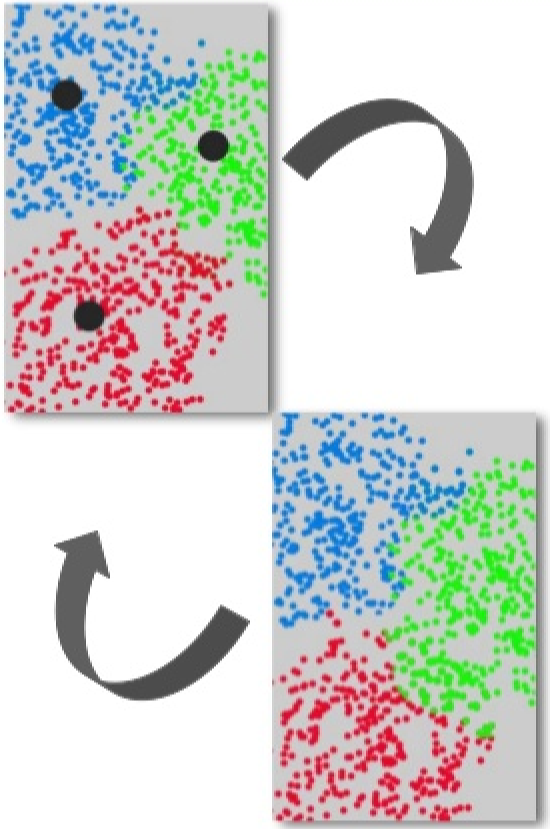
1. First we need to decide the number of clusters to be made. (Guessing)
2. Then we provide centroids of all the clusters. (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assigns the point to the closest cluster.
4. Next the Centroids are calculated again, 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.



edureka!

K-Means Clustering : Steps

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.

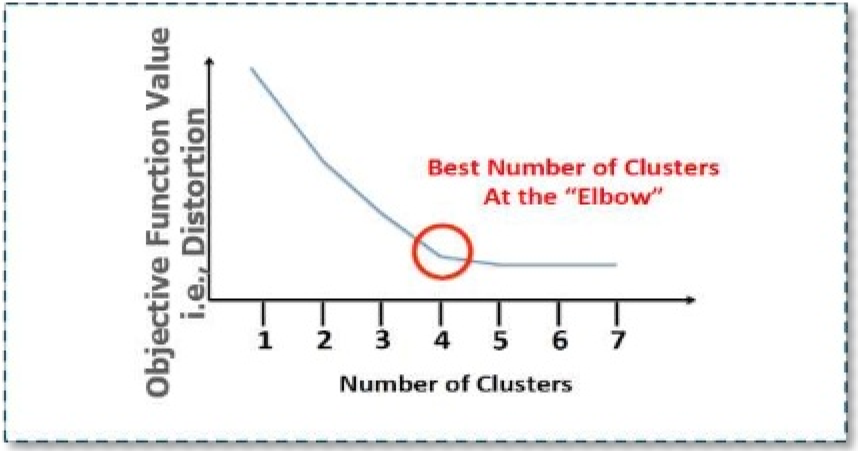


edureka!

How to Decide the number of Clusters

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:



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

Pros and Cons: K-Means Clustering

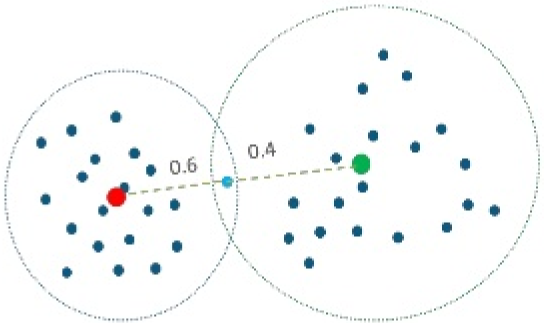
- Simple, understandable
- Items automatically assigned to clusters

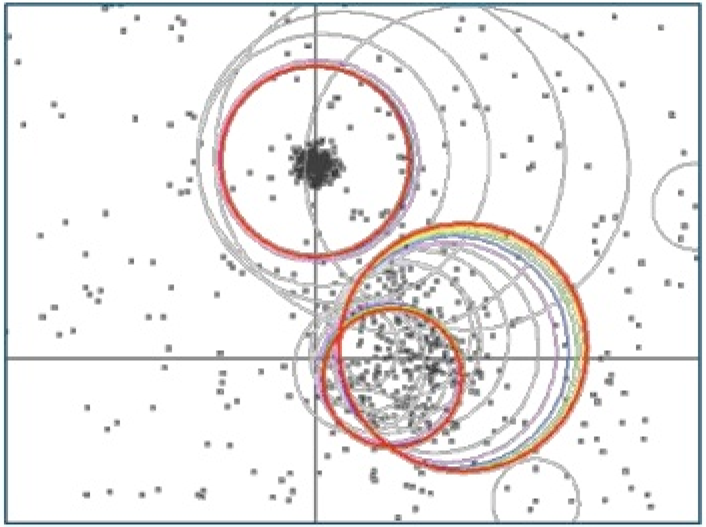
- Must define number of clusters
- All items forced into clusters
- Unable to handle noisy data and outliers

Fuzzy C – Means Clustering

Fuzzy C-Means is an extension of K-Means, the popular simple clustering technique

Fuzzy clustering (also referred to as soft clustering) is a form of Clustering in which each data point can belong to more than one cluster





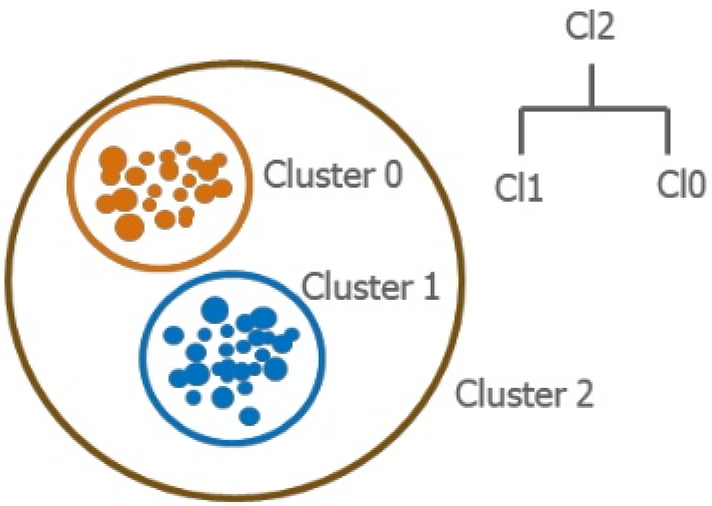
Pros and Cons: C-Means Clustering

- Allows a data point to be in multiple clusters
- A more natural representation of the behaviour of genes
- Genes usually are involved in multiple functions

- Need to define c, the number of clusters
- Need to determine membership cut-off value
- Clusters are sensitive to initial assignment of centroids
- Fuzzy c-means is not a deterministic algorithm

Hierarchical Clustering

Hierarchical clustering is an alternative approach which builds a hierarchy from the bottom-up, and doesn't require us to specify the number of clusters beforehand



Pros and Cons: Hierarchical Clustering

- No assumption of a particular number of clusters
- May corresponds to meaningful taxonomies

- Once a decision is made to combine two clusters, it can’t be undone
- Too slow for large datasets

Demo

Market Basket Analysis

Market basket analysis explains the combinations of products that frequently co-occur in transactions.

Market Basket Analysis algorithms

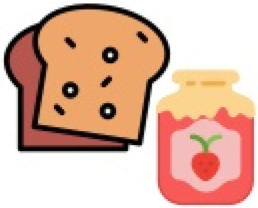
1. Association Rule Mining
2. Apriori

Association Rule Mining

Association rule mining is a technique that shows how items are associated to each other.

Example:

Customer who purchase bread have a 60% likelihood of also purchasing jam.



Customer who purchase laptops are more likely to purchase laptop bags.

Association Rule Mining

Example of Association rule

$$A \Rightarrow B$$

- It means that if a person buys item A then he will also buy item B
- Three common ways to measure association:

- Support
- Confidence
- Lift

Association Rule Mining

Support gives fraction of transactions which contains the item A and B

$$Support = \frac{freq(A,B)}{N}$$

Confidence gives how often the items A & B occur together, given no. of times A occurs

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

Lift indicates the strength of a rule over the random co-occurrence of A and B

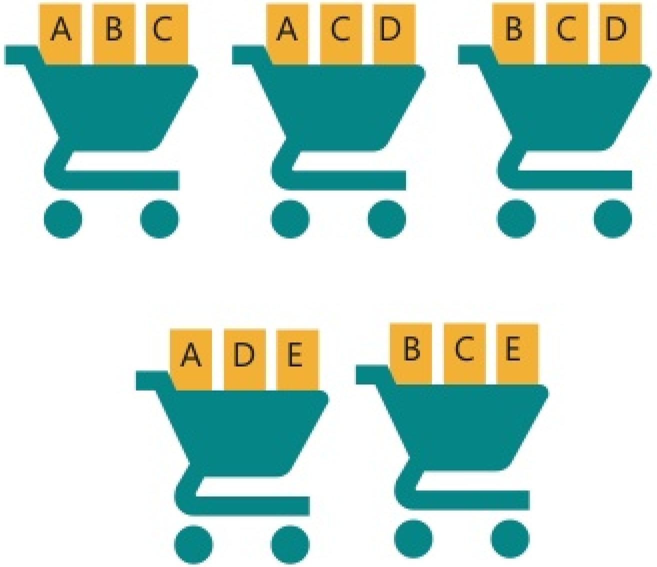
$$Lift = \frac{Support}{Supp(A)\ x\ Supp(B)}$$

Association Rule Mining Example

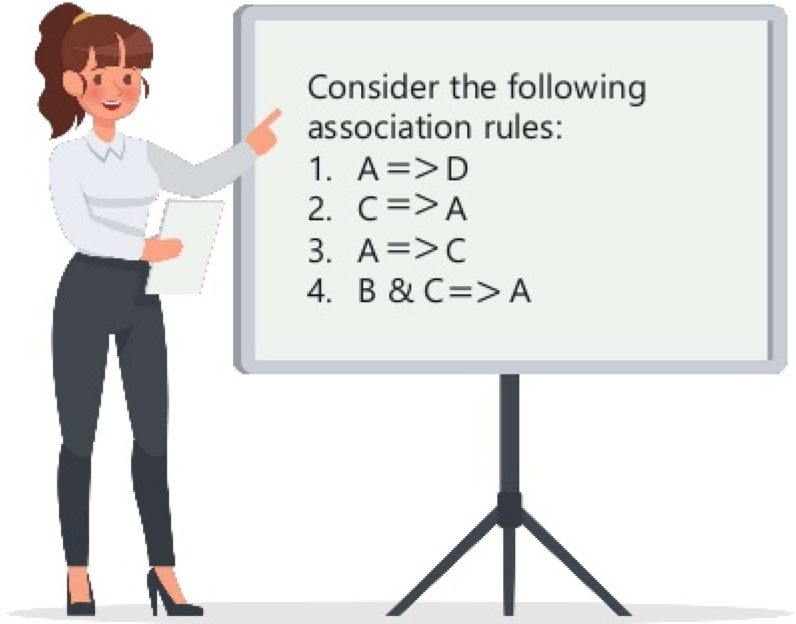
Set of items {A, B, C, D, E}
Set of transactions {T1, T2, T3, T4, T5}

Transactions at a local store

T1	A	B	C
T2	A	C	D
T3	B	C	D
T4	A	D	E
T5	B	C	E



Association Rule Mining Example



Calculate support, confidence and lift for these rules:

Rule	Support	Confidence	Lift
A =>D	2/5	2/3	10/9
C => A	2/5	2/4	5/6
A => C	2/5	2/3	5/6
B, C=>A	1/5	1/3	5/9

Apriori Algorithm

Apriori algorithm uses frequent item sets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.

Frequent Itemset: an itemset whose support value is greater than a threshold value.

Example: If {A,B} is a frequent item set, then {A} and {B} should be frequent item sets

Apriori Algorithm

Consider the following transactions:

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Note:

Min. support count = 2

Apriori Algorithm – First Iteration

Step 1: Create item sets of size one & calculate their support values

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

Table: F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Item sets with support value less than min. support value (i.e. 2) are eliminated

Apriori Algorithm – Second Iteration

Step 2: Create item sets of size two & calculate their support values.
All the combinations of item sets in F1| are used in this iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C2|

Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Table: F2|

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Item sets with support less than 2 it are eliminated

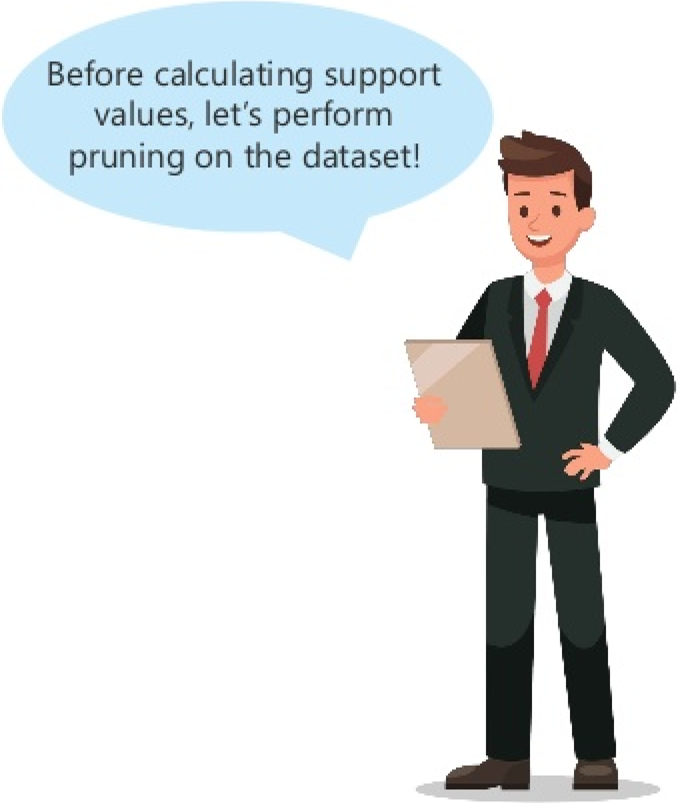
Apriori Algorithm – Third Iteration

Step 3: Create item sets of size three & calculate their support values.
All the combinations of item sets in F2| are used in this iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C3|

Itemset	Support
{1,2,3}	
{1,2,5}	
{1,3,5}	
{2,3,5}	



Apriori Algorithm – Pruning

After the combinations are made, divide C3| item sets to check if there any other subsets whose support is less than min support value.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C3|

Itemset	In F2 |?
{1,2,3} {1,2},{1,3},{2,3}	NO
{1,2,5} {1,2},{1,5},{2,5}	NO
{1,3,5} {1,5},{1,3},{3,5}	YES
{2,3,5} {2,3},{2,5},{3,5}	YES

Table: F2|

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

If any of the subsets of these item sets are not there in FI2 then we remove that itemset

Apriori Algorithm – Fourth Iteration

Using the item sets of C3|, create new itemset C4|.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: F3|

Itemset	Support
{1,3,5}	2
{2,3,5}	2

Table: C4|

Itemset	Support
{1,2,3,5}	1

Since support of C4| is less than 2, stop and return to the previous itemset, i.e. CI3

Apriori Algorithm – Subset Creation

Frequent Item set F3|

Itemset	Support
{1,3,5}	2
{2,3,5}	2

- Generate all non empty subsets for each frequent item set
    - ❖ For I = {1,3,5}, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}
    - ❖ For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}
- For every subsets S of I, output the rule:

S → (I-S) (S recommends I-S)

if ***support(I)/support(S) >= min_conf value***

Apriori Algorithm – Applying Rules

Applying rules to item sets of F3|:

1. {1,3,5}

- ✓ Rule 1: **{1,3} → ({1,3,5} - {1,3})** means 1 & 3 → 5
  Confidence = support(1,3,5)/support(1,3) = 2/3 = 66.66% > 60%
  Rule 1 is selected
- ✓ Rule 2: **{1,5} → ({1,3,5} - {1,5})** means 1 & 5 → 3
  Confidence = support(1,3,5)/support(1,5) = 2/2 = 100% > 60%
  Rule 2 is selected
- ✓ Rule 3: **{3,5} → ({1,3,5} - {3,5})** means 3 & 5 → 1
  Confidence = support(1,3,5)/support(3,5) = 2/3 = 66.66% > 60%
  Rule 3 is selected

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Apriori Algorithm – Applying Rules

Applying rules to item sets of F3|:

1. {1,3,5}

- ✓ Rule 4: **{1} → ({1,3,5} - {1})** means 1 → 3 & 5
  Confidence = support(1,3,5)/support(1) = 2/3 = 66.66% > 60%
  Rule 4 is selected
- ✓ Rule 5: **{3} → ({1,3,5} - {3})** means 3 → 1 & 5
  Confidence = support(1,3,5)/support(3) = 2/4 = 50% <60%
  Rule 5 is rejected
- ✓ Rule 6: **{5} → ({1,3,5} - {5})** means 5 → 1 & 3
  Confidence = support(1,3,5)/support(3) = 2/4 = 50% < 60%
  Rule 6 is rejected

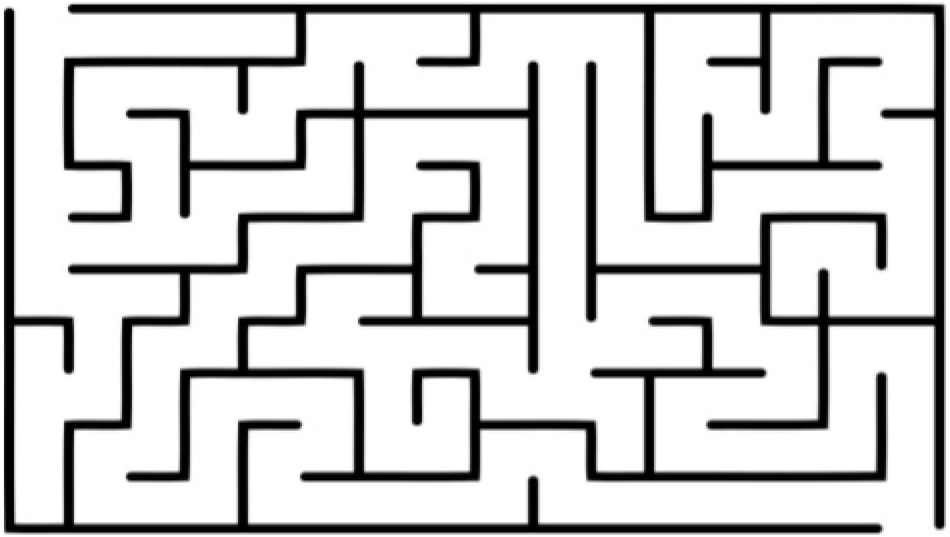
TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Demo

edureka!

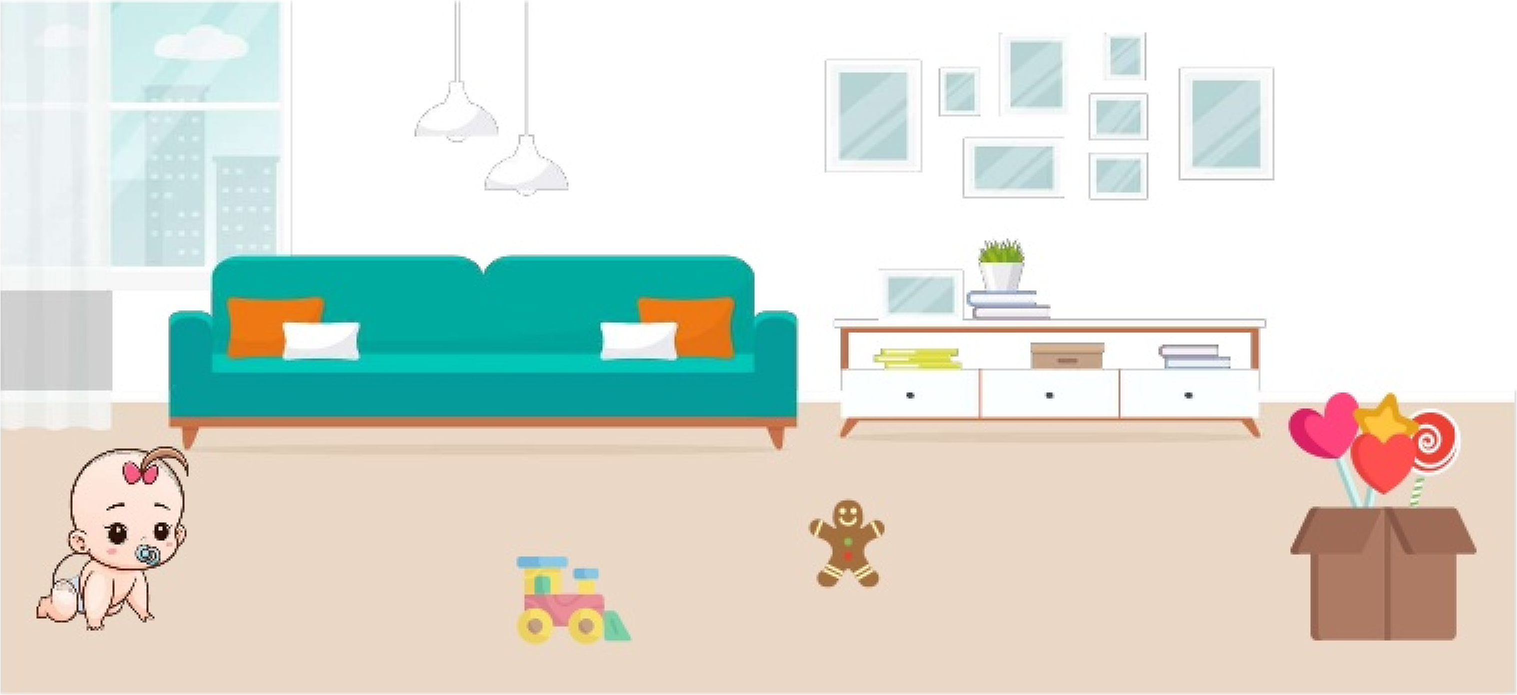
What is Reinforcement Learning?

Reinforcement learning is a type of Machine Learning where an agent learns to behave in an environment by performing actions and seeing the results



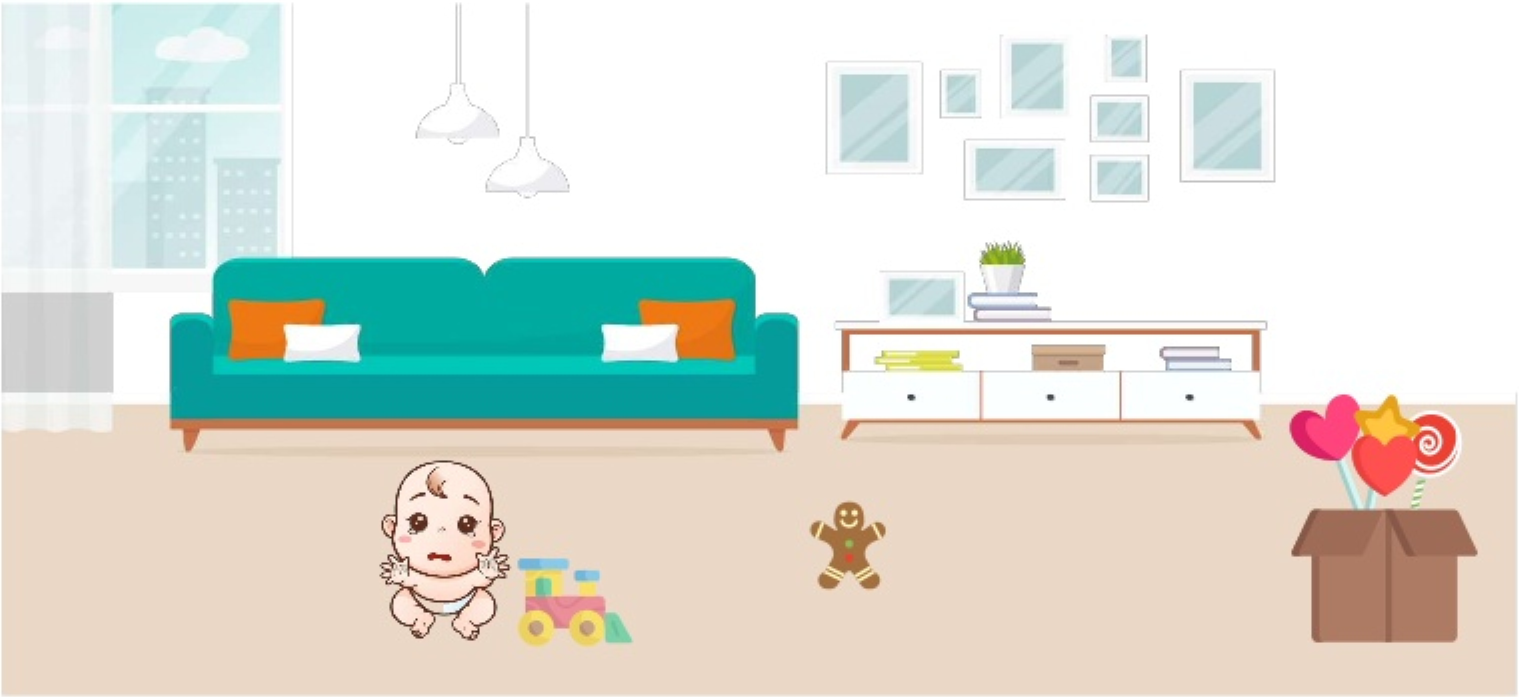
Analogy

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



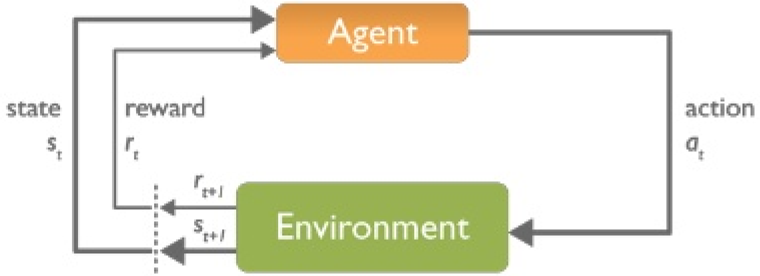
Analogy

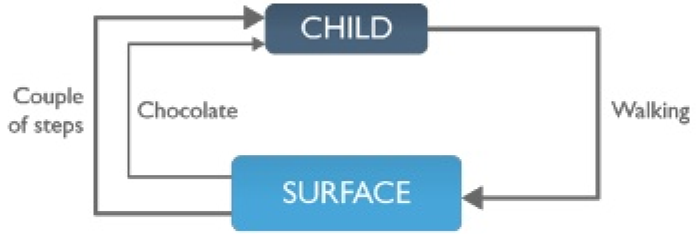
Scenario 2: Baby starts crawling but falls due to some hurdle in between



Analogy

- **The child is an agent** trying to manipulate the **environment (which is the surface on which it walks)**
- **Taking actions (viz walking)** and he/she tries to go from one **state (viz each step he/she takes)** to another
- The child gets a **reward (let's say chocolate)** when he/she accomplishes a **submodule of the task (viz taking couple of steps)**





- Will not receive any chocolate **(negative reward)** when he/she is not able to walk

Reinforcement Learning Definitions

Agent

Intelligent programs

Model of the Environment

Used for planning & if know the current state and action then predict the resultant next state and next reward

Environment

An external condition

Value Function

Value of a state is the total amount of reward an agent can expect to accumulate over the future

Policy

A mapping from state to actions defining agent's behavior at a particular time

Reward Function

Could be +1 or any other value, indicating, what's good in an immediate sense

Reinforcement Learning Definitions

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

Reinforcement Learning Definitions

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

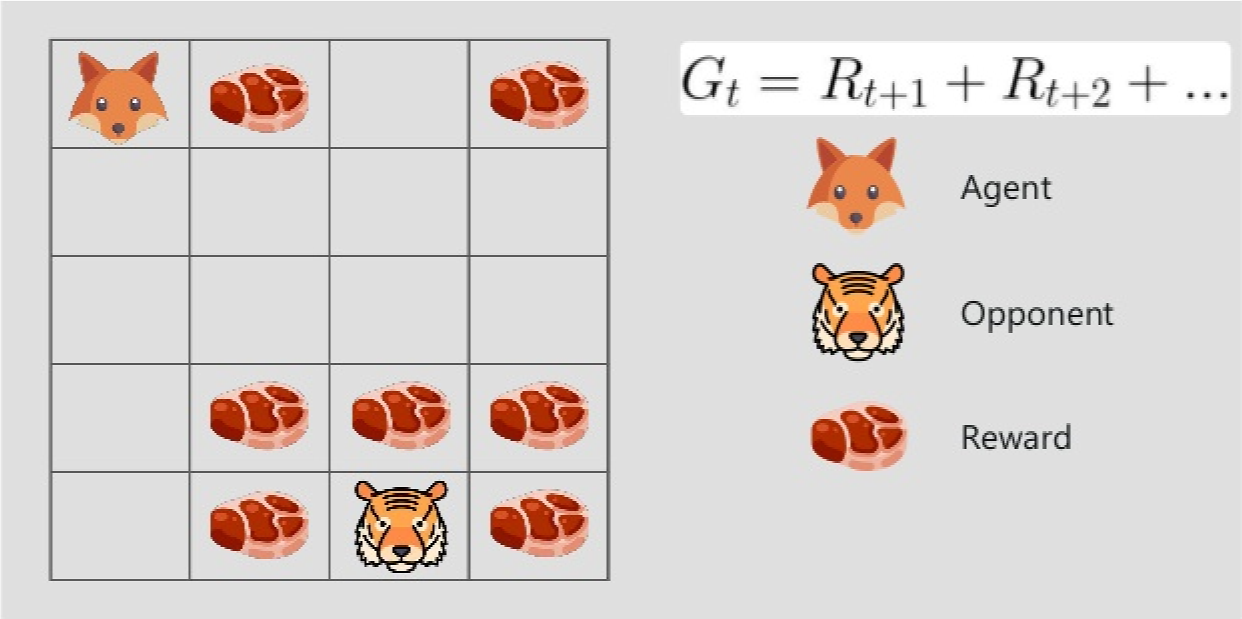
Policy (π): The approach that the agent uses to determine the next action based on the current state

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

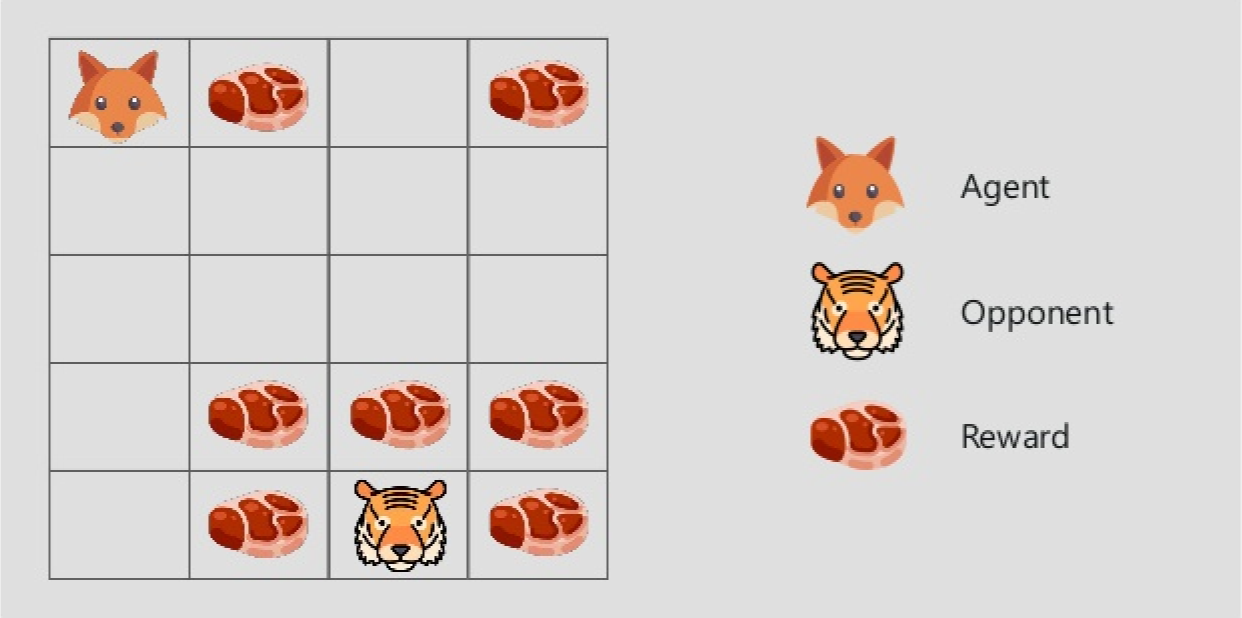
Reward maximization theory states that, *a RL agent must be trained in such a way that, he takes the best action so that the reward is maximum.*



$$G_t = R_{t+1} + R_{t+2} + \ldots$$

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



$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} where \gamma \in [0, 1)$$

$$R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots$$

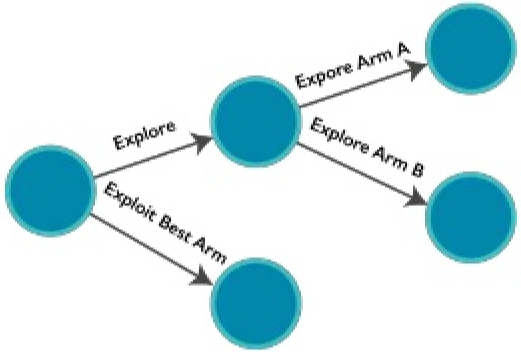
The K-Armed Bandit Problem

- The K-armed bandit is a metaphor representing a casino slot machine with k pull levers (or arms)
- The user or customer pulls any one of the levers to win a predefined reward
- The objective is to select the lever that will provide the user with the highest reward

The Epsilon Greedy Algorithm

01 Tries to be fair to the two opposite goals of exploration and exploitation by using a mechanism of flipping a coin

02 Just like, if you flip a coin and it comes up head you should explore for a moment but if it comes up tails, you should exploit



Takes whatever action seems best at the present moment

The Epsilon Greedy Algorithm

01 With probability 1 – epsilon, the Epsilon-Greedy algorithm exploits the best known option

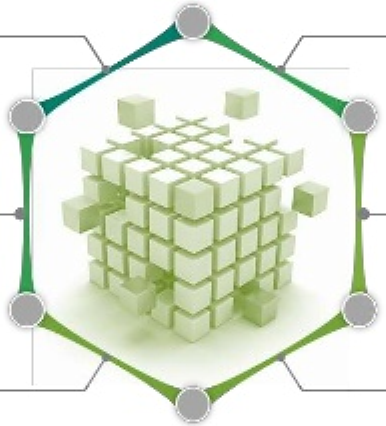
02 With probability epsilon / 2, the Epsilon-Greedy algorithm explores the best known option

03 With probability epsilon / 2, the Epsilon-Greedy algorithm explores the worst known option

Markov 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:

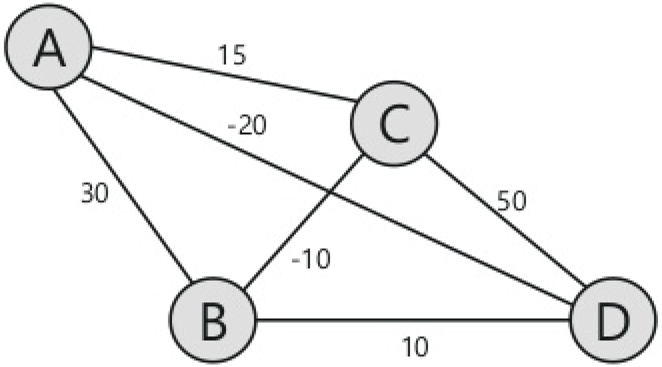


Markov Decision Process- Shortest Path Problem

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

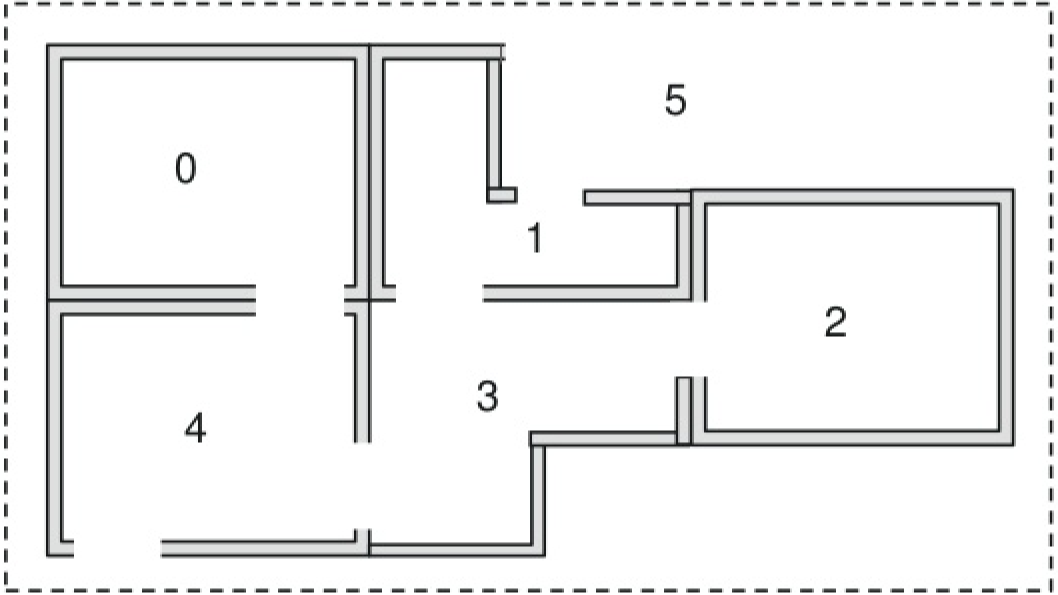
In this problem,

- 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 -> C -> D}



Understanding Q-Learning With An Example

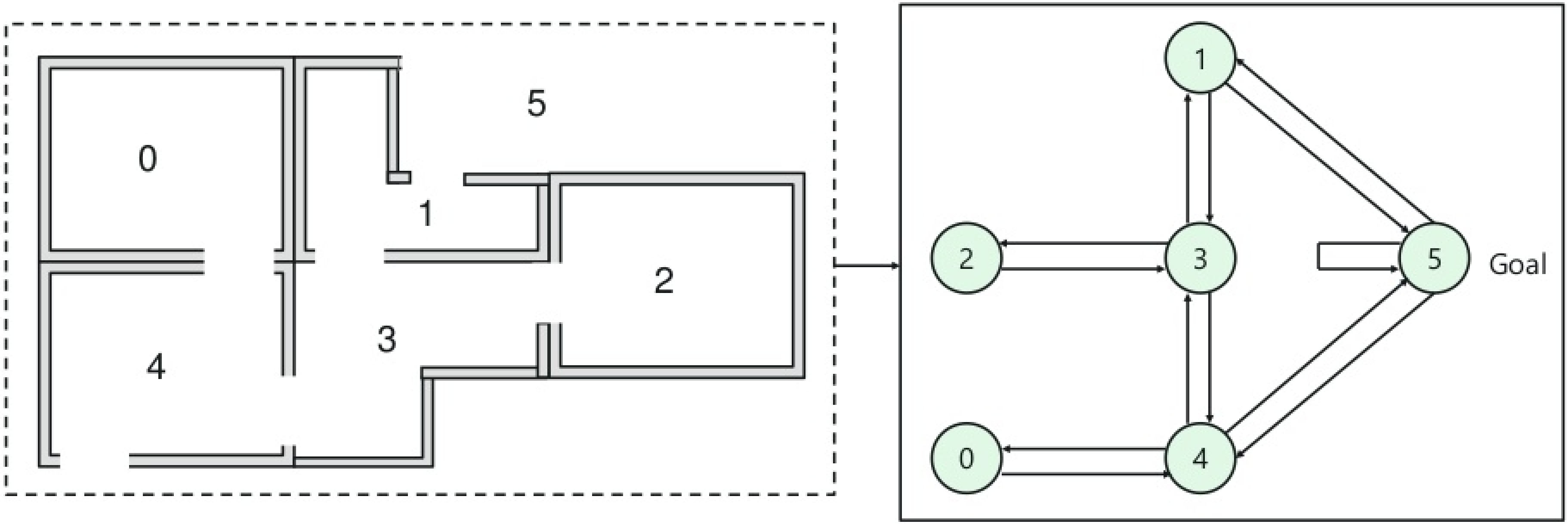
Place an agent in any one of the rooms (0,1,2,3,4) and the goal is to reach outside the building (room 5)



- 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 With An Example

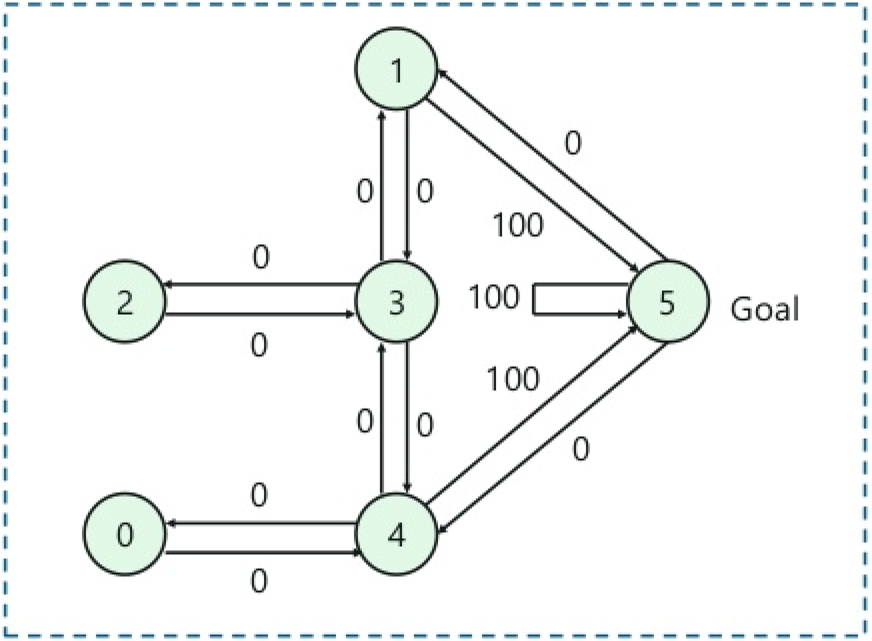
Let's represent the rooms on a graph, each room as a node, and each door as a link



Understanding Q-Learning With An Example

Next step is to associate a reward value to each door:

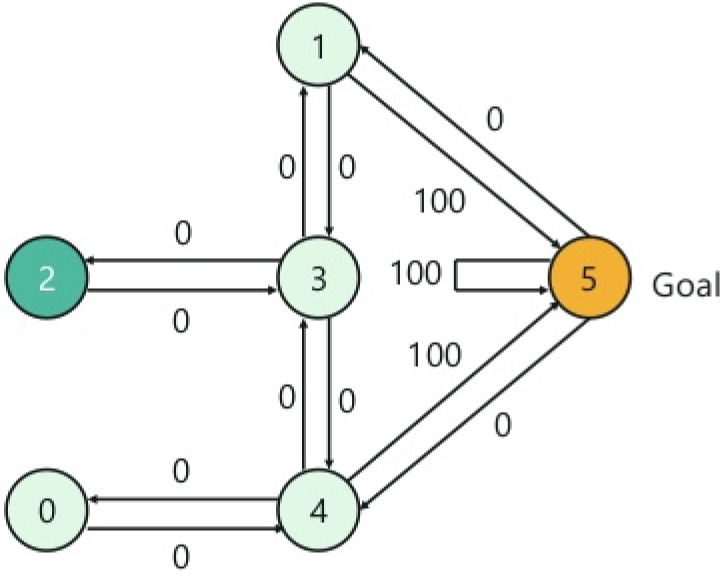
- doors that lead directly to the goal have a reward of 100
- Doors not directly connected to the target room have zero reward
- Because doors are two-way, two arrows are assigned to each room
- Each arrow contains an instant reward value



Understanding Q-Learning With An Example

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

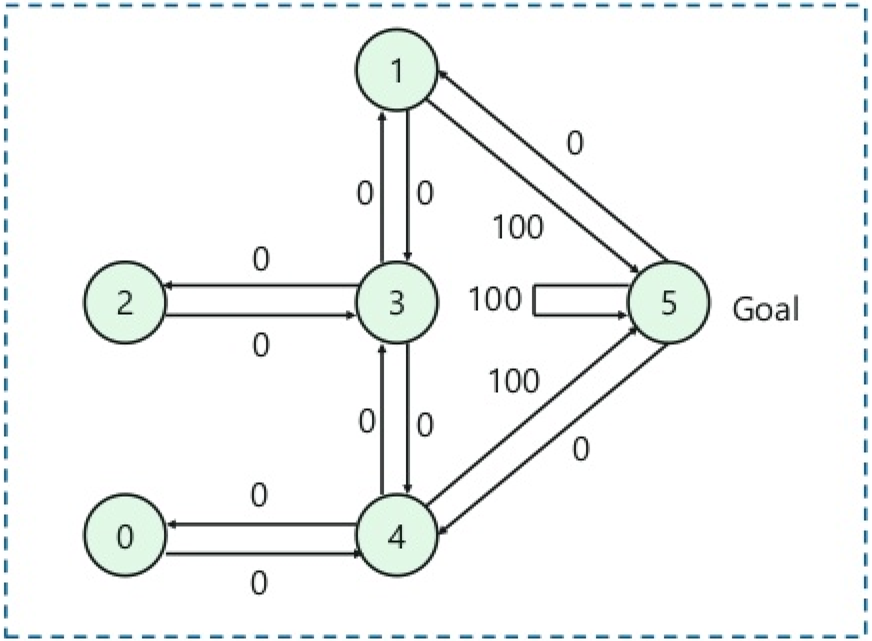


Example (Agent traverse from room 2 to room5):

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 With An Example

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 With An Example

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.

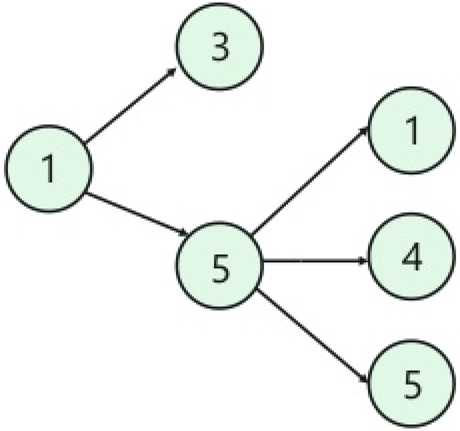
Next, initialize matrix Q as a zero matrix:
- From room 1 you can either go to room 3 or 5, let's select room 5.
- From room 5, calculate maximum Q value for this next state based on all possible actions:

*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

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

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



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:

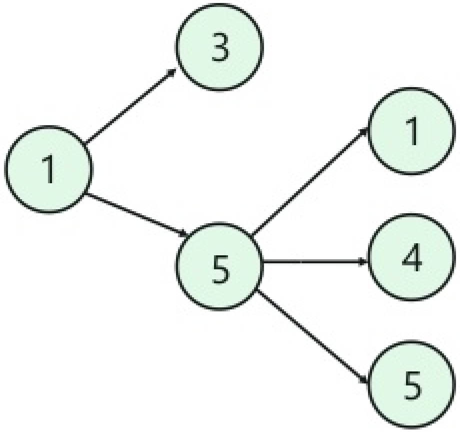
- From room 1 you can either go to room 3 or 5, let's select room 5.
- From room 5, calculate maximum Q value for this next state based on all possible actions

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

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

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

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



Q – Learning Example

For the next episode, we start with a randomly chosen initial state, i.e. state 3

- From room 3 you can either go to room 1,2 or 4, let's select room 1.
- From room 1, 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(3,1) = R(3,1) + 0.8 * Max[Q(1,3), Q(1,5)] = 0 + 0.8 * [0, 100] = 80

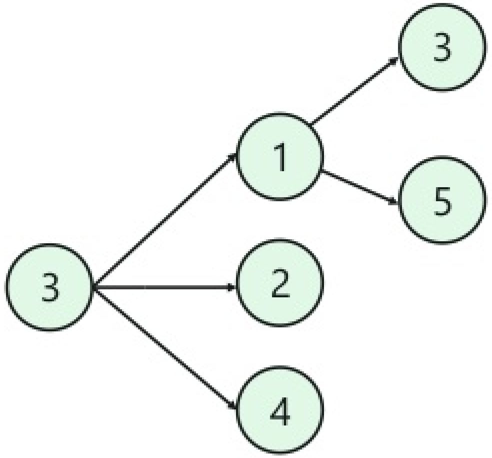
The matrix Q get's updated

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

Action

State

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



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)]*

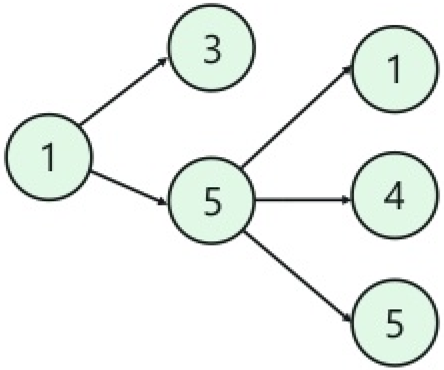
Q(1,5) = R(1,5) + 0.8 * Max[Q(5,1), Q(5,4), Q(5,5)] = 100 + 0.8 * 0 = 100

The matrix Q remains the same since, Q(1,5) is already fed to the agent

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

Action

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



Demo

edureka!

Thank You

For more information please visit our website
www.edureka.co