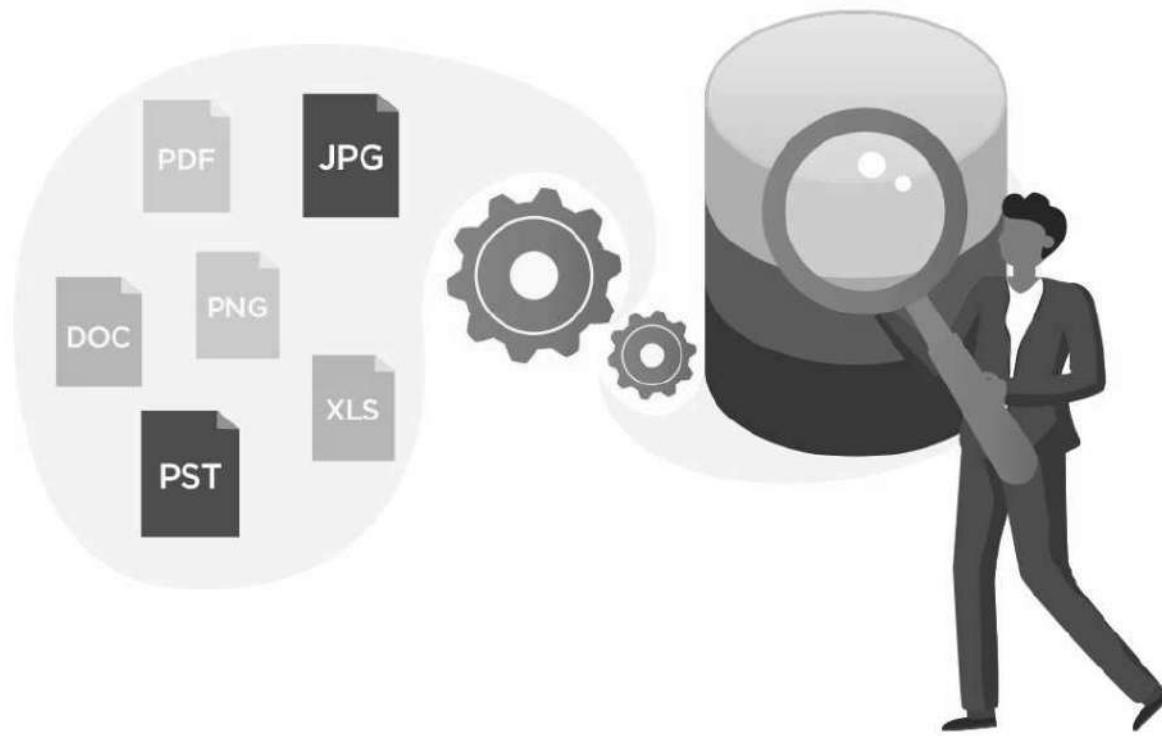


Data Mining (20CP306T)



Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

Syllabus

Unit-I: INTRODUCTION

- ❖ Introduction: What is Data Mining? Motivating Challenges; The origins of data mining; Data Mining Tasks. Types of Data; Data Pre-processing, Measures of Similarity and Dissimilarity.

Unit-II: SUPERVISED LEARNING

- ❖ Classification: Preliminaries; General approach to solving a classification problem; Decision tree induction; Rule-based classifier; Multilinear and Logistic Regression

UNIT 3 ASSOCIATION ANALYSIS

- ❖ Problem definition, Frequent item set generation; Rule Generation; Compact representation of frequent item sets; Alternative methods for generating frequent item sets. FP-Growth algorithm, Evaluation of association patterns, Effect of skewed support distribution, Sequential patterns.

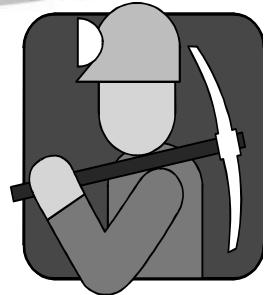
UNIT 4 UNSUPERVISED LEARNING & CLUSTERING

- ❖ Clustering, KNN, Clustering Review, Outlier Detection, Recent Trends in Data Mining

Necessity for Data Mining

- Large amounts of current and historical data being stored
 - Only small portion (~5-10%) of collected data is analyzed
 - Data that may never be analyzed is collected in the fear that something that may prove important will be missed
- As databases grow larger, decision-making from the data is not possible; need knowledge derived from the stored data
- Data sources
 - Health-related services, e.g., benefits, medical analyses
 - Commercial, e.g., marketing and sales
 - Financial
 - Scientific, e.g., NASA
- Desired analyses
 - Support for planning (historical supply and demand trends)
 - Yield management (scanning airline seat reservation data to maximize yield per seat)
 - System performance (detect abnormal behavior in a system)
 - Mature database analysis (clean up the data sources)

What Is Data Mining?



- Data mining (knowledge discovery from data)
 - Extraction of interesting (valid, significant previously unknown and potentially useful) patterns or knowledge from huge amount of data.
 - The process of discovering meaningful new correlations, patterns, and trends by sifting through large amounts of stored data, using pattern recognition technologies and statistical and mathematical techniques
- Alternative names
 - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, information harvesting, business intelligence, etc.

Data Mining—What's in a Name?

Information Harvesting

Data Mining

Knowledge Mining

**Knowledge Discovery
in Databases**

Data Dredging

Data Pattern Processing

Data Archaeology

Database Mining

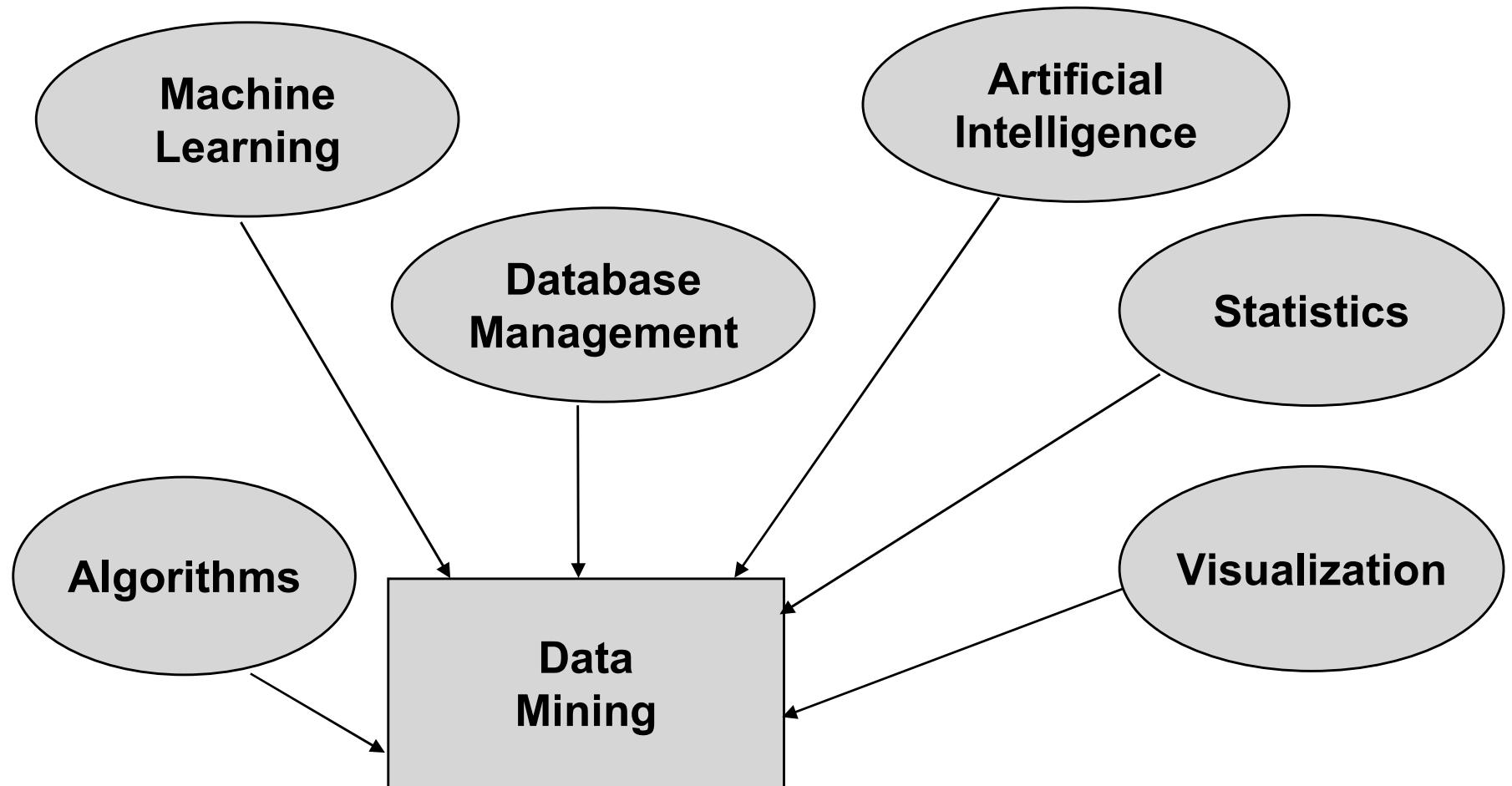
Knowledge Extraction

Siftware

Data Mining: History of the Field

- *The term “data mining” has been around since at least 1983 – as a pejorative term in the statistics community*
- Knowledge Discovery in Databases workshops started ‘89
 - Now a conference under the auspices of ACM SIGKDD
 - IEEE conference series started 2001
- Key founders / technology contributors:
 - Usama Fayyad, JPL (then Microsoft, then his own company, Digimine, now Yahoo! Research labs)
 - Gregory Piatetsky-Shapiro (then GTE, now his own data mining consulting company, Knowledge Stream Partners)
 - Rakesh Agrawal (IBM Research)

Integration of Multiple Technologies



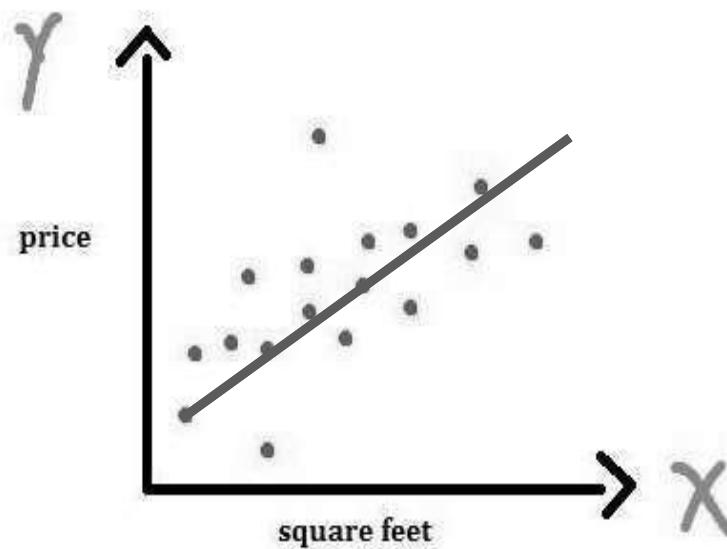
Data Mining: Classification Schemes

- General functionality
 - Descriptive data mining
 - Predictive data mining
- Different views, different classifications
 - Kinds of data to be mined
 - Kinds of knowledge to be discovered
 - Kinds of techniques utilized
 - Kinds of applications adapted

Regression, Classification, Clustering, Association Rule

Regression

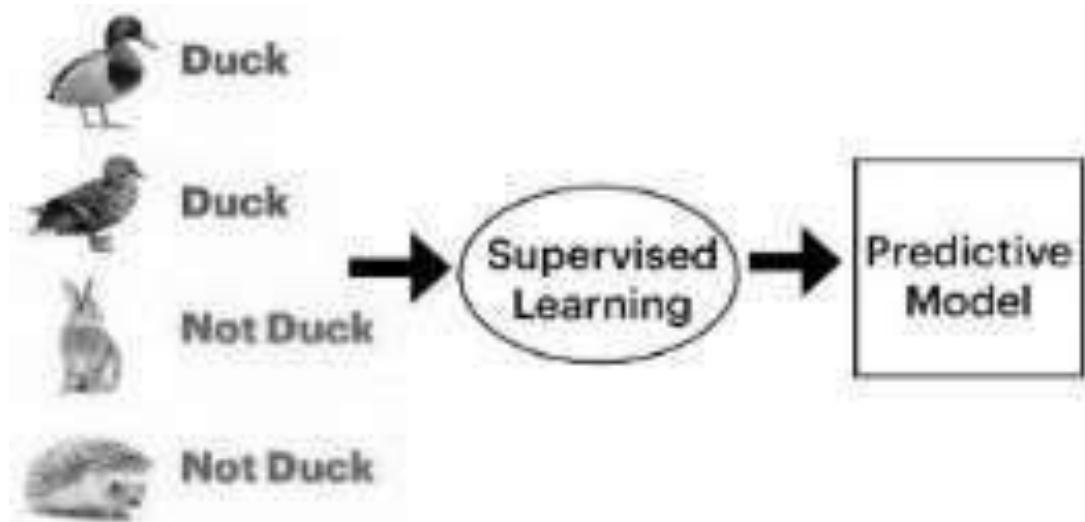
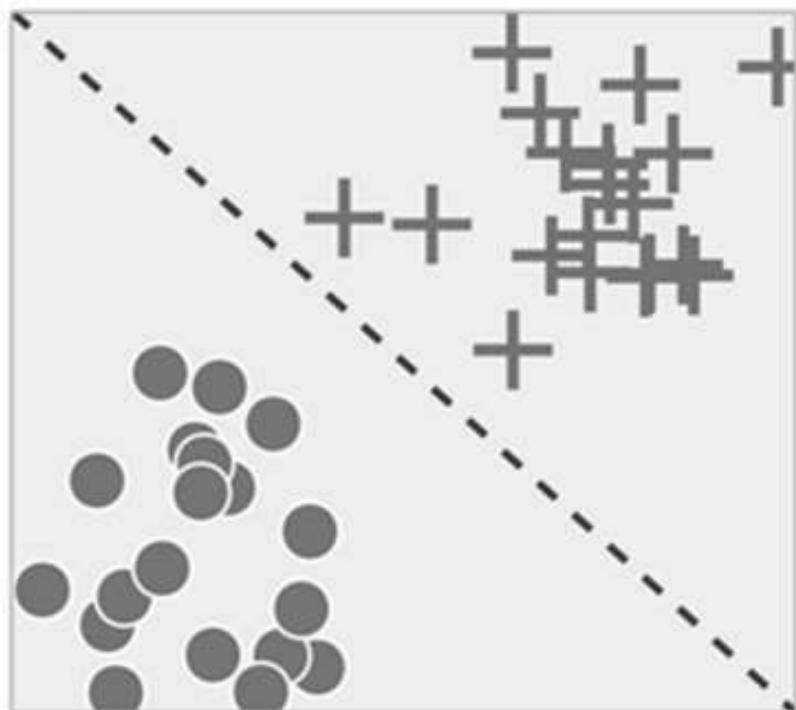
- If the output of the model is a continuous value.



- Ex.
 - ❖ House price prediction
 - ❖ Stock market prediction
 - ❖ Predicting age of a person
 - ❖ number of copies a music album will be sold next month

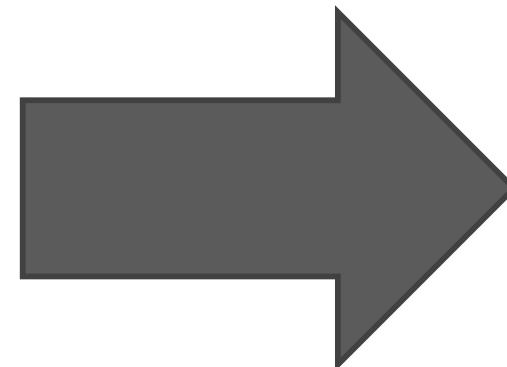
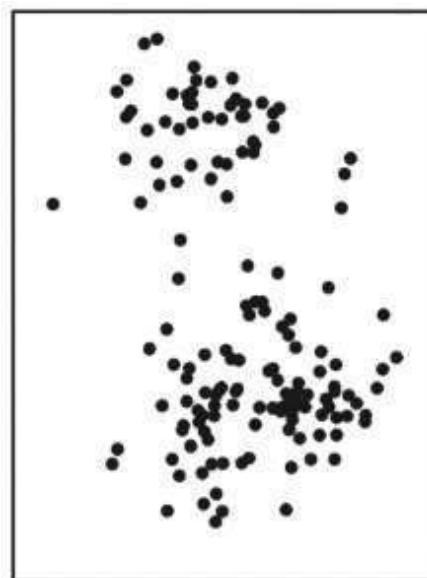
Classification

Classification

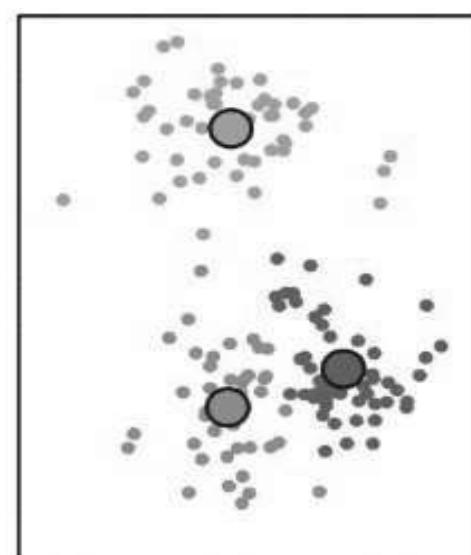


Cluster

Input Data



Final Result



Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection;
 - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Rules Discovered:

$\{\text{Milk}\} \rightarrow \{\text{Coke}\}$

$\{\text{Diaper, Milk}\} \rightarrow \{\text{Beer}\}$

Types of Data Mining

1. Predictive Data Mining Analysis
2. Descriptive Data Mining Analysis

1. Predictive Data Mining

- Predictive Data-Mining analysis works on the data that may help to know **what may happen later (or in the future) in business.**
- Predictive Data-Mining can also be further divided into four types that are listed below:
 - Classification Analysis
 - Regression Analysis
 - Time Serious Analysis
 - Prediction Analysis

2. Descriptive Data Mining

- The main goal of the Descriptive Data Mining tasks is to summarize or turn given data into relevant information.
- The Descriptive Data-Mining Tasks can also be further divided into four types that are as follows:
 - Clustering Analysis
 - Summarization Analysis
 - Association Rules Analysis
 - Sequence Discovery Analysis

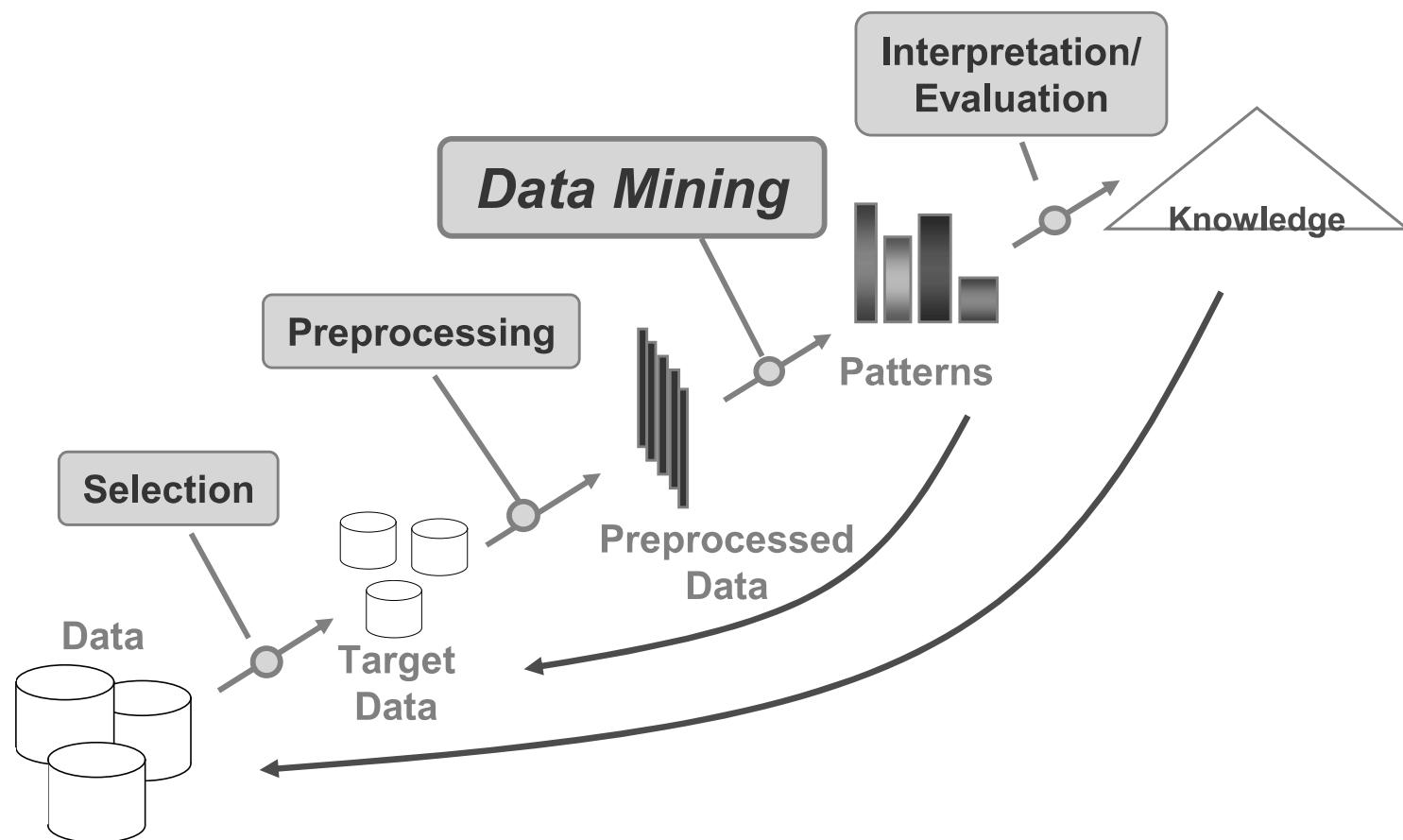
Multi-Dimensional View of Data Mining

- Data to be mined
 - Relational, data warehouse, transactional, stream, object-oriented/relational, active, spatial, time-series, text, multi-media, heterogeneous, legacy, WWW
- Knowledge to be mined
 - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
 - Multiple/integrated functions and mining at multiple levels
- Techniques utilized
 - Database-oriented, data warehouse (OLAP), machine learning, statistics, visualization, etc.
- Applications adapted
 - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, Web mining, etc.

What Can Data Mining Do?

- Cluster
- Classify
 - Categorical, Regression
- Summarize
 - Summary statistics, Summary rules
- Link Analysis / Model Dependencies
 - Association rules
- Sequence analysis
 - Time-series analysis, Sequential associations
- Detect Deviations

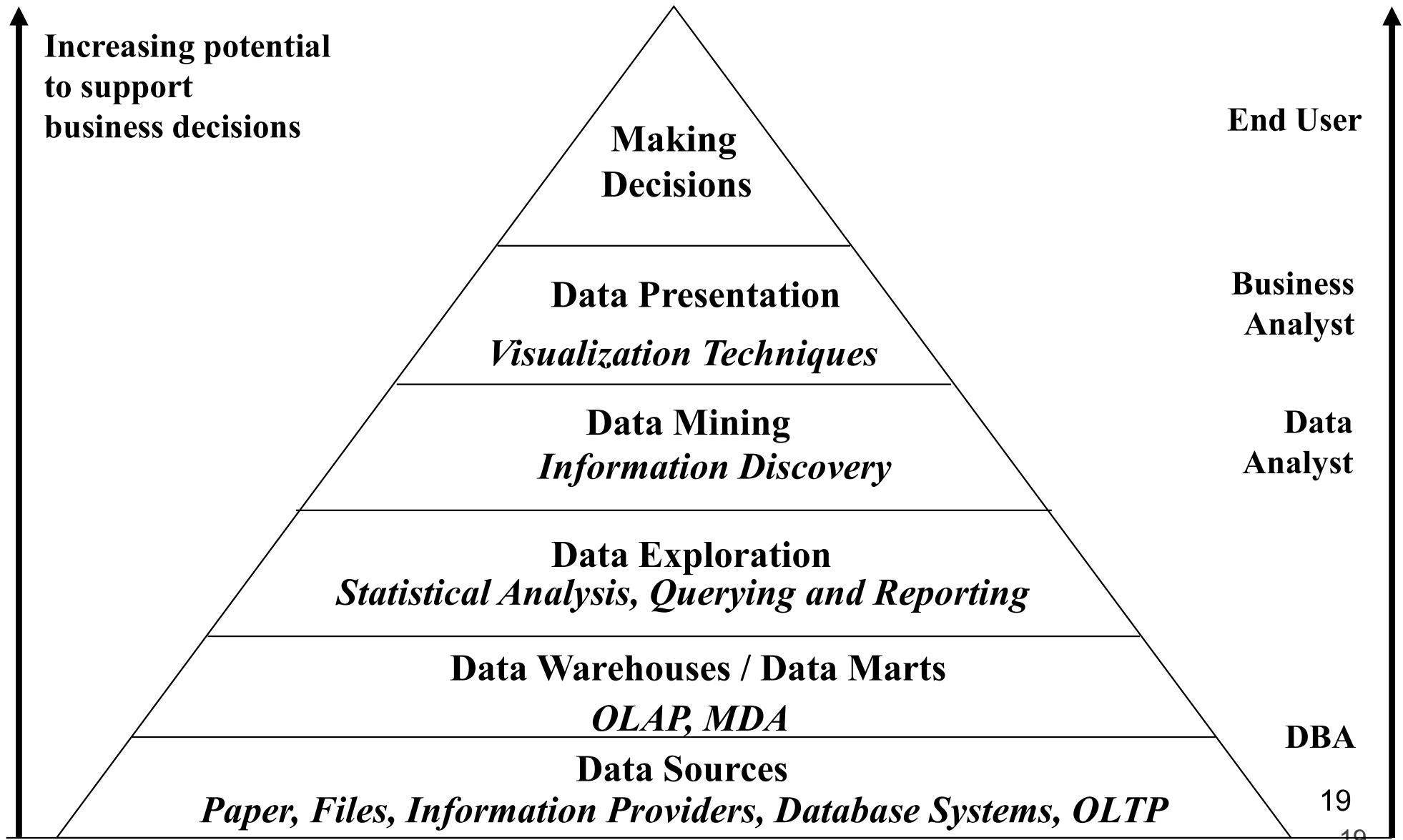
Knowledge Discovery in Databases (KDD): Process



Steps of a KDD Process

- Learning the application domain
 - relevant prior knowledge and goals of application
- Creating a target data set: data selection
- Data cleaning and preprocessing: (may take 60% of effort!)
- Data reduction and transformation
 - Find useful features, dimensionality/variable reduction, invariant representation.
- Choosing functions of data mining
 - summarization, classification, regression, association, clustering.
- Choosing the mining algorithm(s)
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation
 - visualization, transformation, removing redundant patterns, etc.
- Use of discovered knowledge

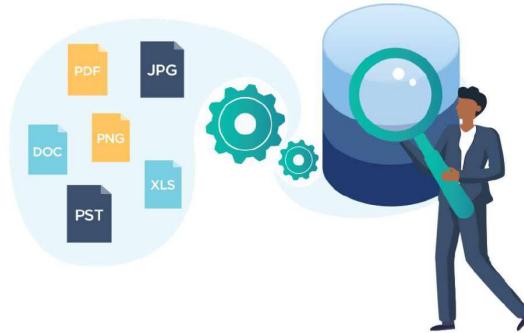
Data Mining and Business Intelligence



Types of Sources of Data in Data Mining

- 1) Flat Files
- 2) Relational Databases
- 3) DataWarehouse
- 4) Transactional Databases
- 5) Multimedia Databases
- 6) Spatial Databases
- 7) Time Series Databases
- 8) World Wide Web(WWW)

Data Mining (20CP306T)



Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

1

Types of Sources of Data in Data Mining

- 1) Flat Files
- 2) Relational Databases
- 3) DataWarehouse
- 4) Transactional Databases
- 5) Multimedia Databases
- 6) Spatial Databases
- 7) Time Series Databases
- 8) World Wide Web(WWW)

2

1. Flat Files

1. Flat files is defined as data files in text form or binary form with a structure that can be easily extracted by data mining algorithms.
2. Data stored in flat files have **no relationship or path among themselves**, like if a relational database is stored on flat file, then there will be no relations between the tables.
3. Flat files are represented by data dictionary. Eg: CSV file.

Application: Used in Data Warehousing to store data, Used in carrying data to and from server, etc.

3

2. Relational Databases

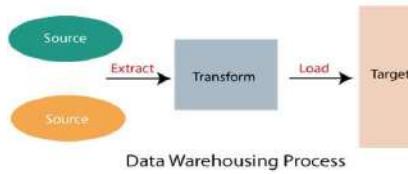
1. A Relational database is defined as the collection of data organized in tables with rows and columns.
2. **Physical schema** in Relational databases is a schema which defines the **structure of tables**.
3. **Logical schema** in Relational databases is a schema which defines the **relationship among tables**.
4. Standard API of relational database is SQL.

Application: Data Mining, ROLAP (Relational Online Analytical Processing) model, etc.

4

3. Data Warehouse

1. A Data Warehouse refers to a place where data can be stored for useful mining. It is defined as the **collection of data integrated from multiple sources** that will queries and decision making.



2. Two approaches can be used to integrate two different heterogeneous databases in Data Warehouse: **Query-driven Approach** and **Update-driven Approach**.

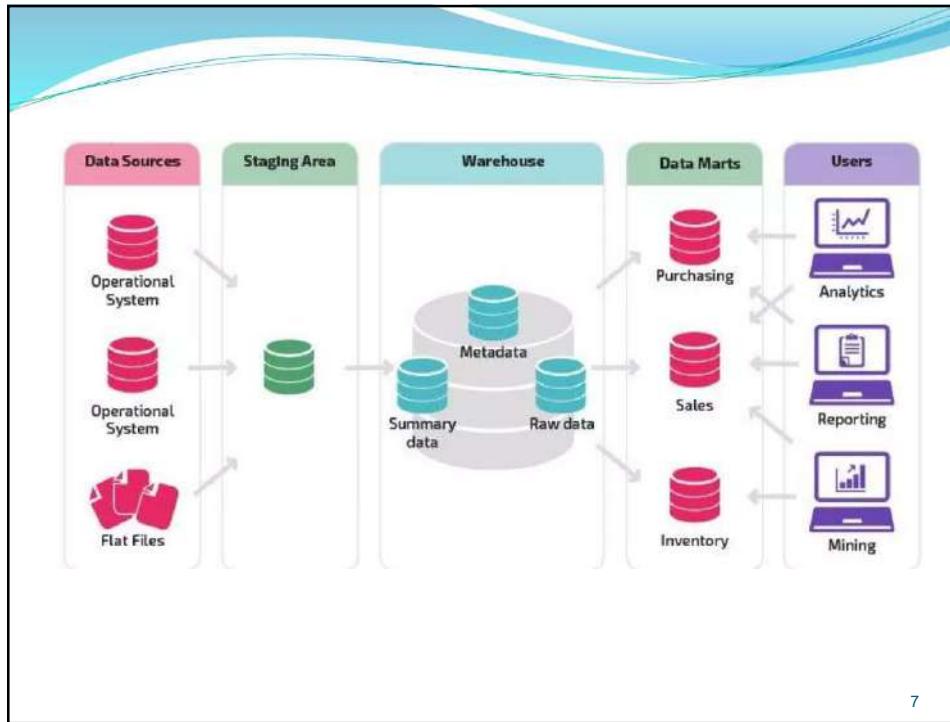
Application: Business decision making, Data mining, etc.

5

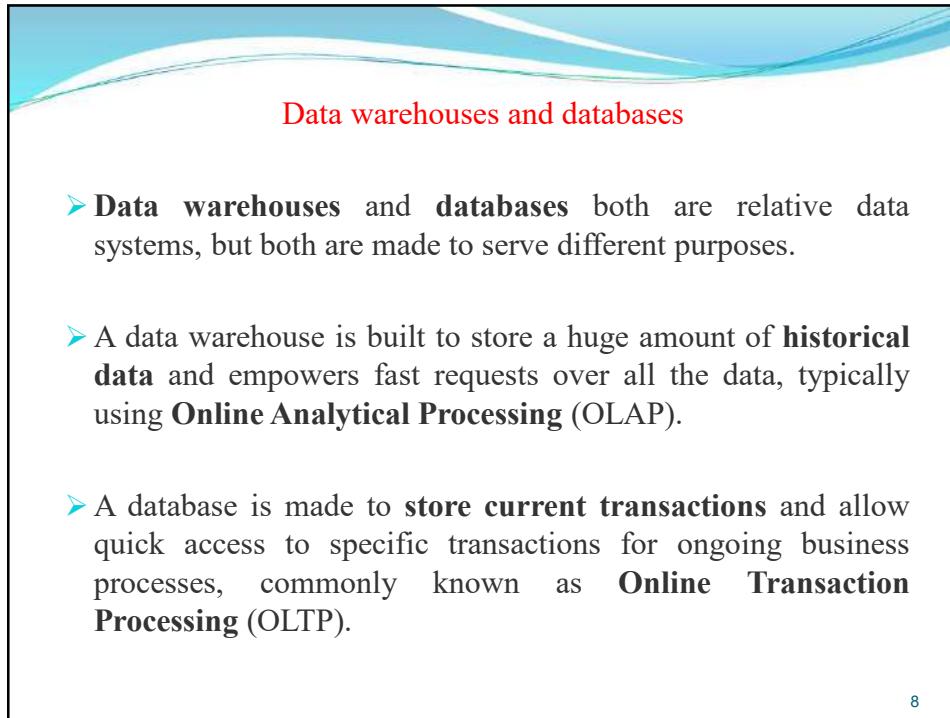
There are three types of data warehouse: **Enterprise** data warehouse **Data Mart** and **Virtual** Warehouse.

- 1) An **enterprise** data **warehouse** (EDW) is a database, or collection of databases, that centralizes a business's information from multiple sources and applications, and makes it available for analytics and use across the organization
- 2) A **data mart** is similar to a data warehouse, but it holds data only for a specific department or line of business, such as sales, finance, or human resources. A data warehouse can feed data to a data mart, or a data mart can feed a data warehouse.
- 3) A **virtual** **warehouse** is essentially a business database. The data found in a **virtual** **warehouse** is usually **copied from multiple sources** throughout a production system. This is done so related data can be searched quickly and without accessing the entire system.

6



7



8

Database Processing vs. Data Mining Processing

- | | |
|---|---|
| <ul style="list-style-type: none"> ● Query <ul style="list-style-type: none"> • Well defined • SQL
 ■ Data <ul style="list-style-type: none"> – Operational data
 ■ Output <ul style="list-style-type: none"> – Precise – Subset of database | <ul style="list-style-type: none"> ● Query <ul style="list-style-type: none"> • Poorly defined • No precise query language
 ■ Data <ul style="list-style-type: none"> – Not operational data
 ■ Output <ul style="list-style-type: none"> – Fuzzy – Not a subset of database |
|---|---|

Query Examples

- Database
 - Find all credit applicants with last name of Smith.
 - Identify customers who have purchased more than \$10,000 in the last month.
 - Find all customers who have purchased milk

- Data Mining
 - Find all credit applicants who are poor credit risks. (classification)
 - Identify customers with similar buying habits. (Clustering)

 - Find all items which are frequently purchased with milk. (association rules)

The Important features of Data Warehouse

1. Subject Oriented

- A data warehouse usually **focuses on modeling and analysis of data** that helps the business organization to make data-driven decisions.
- A data warehouse is subject-oriented. It provides useful data about a **subject instead of the company's ongoing operations**, and these subjects can be customers, suppliers, marketing, product, promotion, etc.

2. Time-Variant:

- The different data present in the data warehouse provides information for a specific period.

3. Integrated

- A data warehouse is built by joining data from heterogeneous sources, such as social databases, level documents, etc.

4. Non- Volatile

- As the name defines the data resided in data warehouse is permanent. It also means that data is not erased or deleted when new data is inserted.

11

4. *Transactional Databases*

1. Transactional databases is a collection of data **organized by time stamps**, date, etc to represent transaction in databases.
2. This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
3. Highly flexible system where users can modify information without changing any sensitive information.
4. Follows [ACID property](#) of DBMS.

Application: Banking, Distributed systems, Object databases, etc.

12

5. *Multimedia Databases*

1. Multimedia databases consists audio, video and images.
2. They can be stored on **Object-Oriented Databases** (MongoDB).
3. They are used to store complex information in a pre-specified formats.

Application: Digital libraries, video-on demand, news-on demand, musical database, etc.

6. *Spatial Database*

1. Store geographical information.
2. Stores data in the form of coordinates, topology, lines, polygons, etc.

Application: Maps, Global positioning, etc.

13

6. *Time-series Databases*

1. Time series databases contains stock exchange data and user logged activities.
2. Handles array of numbers indexed by time, date, etc.
3. It requires real-time analysis.

14

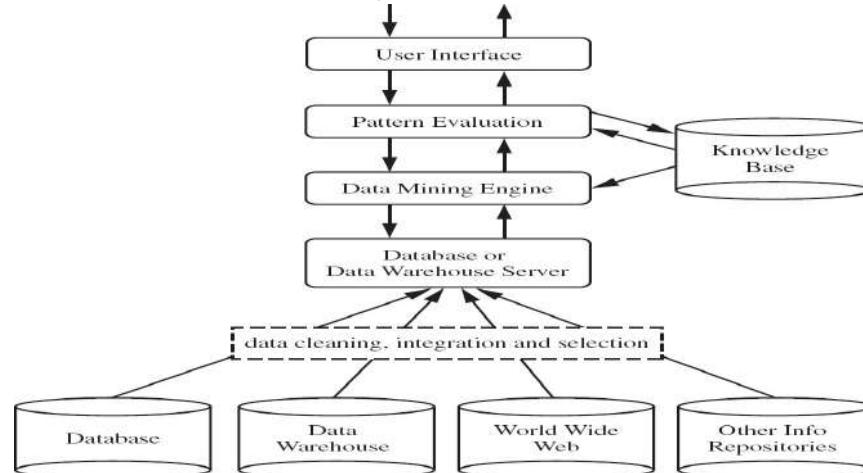
WWW

1. WWW refers to World wide web is a collection of documents and resources like audio, video, text, etc which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet network.
2. It is the most **heterogeneous repository** as it collects data from multiple resources.
3. It is **dynamic in nature** as Volume of data is continuously increasing and changing.

Application: Online shopping, Job search, Research, studying, etc.

15

Architecture: Typical Data Mining System



16

A typical DM System Architecture

- Database, data warehouse, WWW or other information repository (**store data**)
- Database or data warehouse server (**fetch and combine data**)
- Knowledge base (**turn data into meaningful groups according to domain knowledge**)
- Data mining engine (**perform mining tasks**)
- Pattern evaluation module (**find interesting patterns**)
- User interface (**interact with the user**)

17

Types of Data

➤ Numeric data

➤ Categorical data

➤ Text data

➤ Image data

➤ Video Data

➤ Audio Data

➤ Time Series Data

Name	Age	Height	Weight	M/F
Anil	50	5.6	70.2	M
Raju	25	5.4	75.8	M
Neetu	35	5.3	46	F
Meethi	8	3.4	24	F



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

18

Types of data

1) Numerical data

- It represents some quantifiable thing that you can measure
 - (a) Discrete data (b) Continuous data

2) Categorical data

- A categorical variable (sometimes called a **nominal** variable) is one that has two or more categories, but there is no intrinsic ordering to the categories.
- For example, gender is a categorical variable having two categories (male and female)

3) Ordinal data

- Mixture of numerical and categorical data.
- An ordinal variable is similar to a categorical variable. The difference between the two is that there is a clear ordering of the variables.
- For example, suppose you have a variable, economic status, with three categories (low, medium and high), moving rating.

Measures of Similarity and Dissimilarity

- In data science, the **similarity measure** is a way of measuring how data samples are related or closed to each other. On the other hand, the **dissimilarity measure** is to tell how much the data objects are distinct.
- Moreover, these terms are often used in clustering when similar data samples are grouped into one cluster.
- It is also used in classification(e.g. KNN), where the data objects are labeled based on the features' similarity.
- The similarity measure is usually expressed as a numerical value (0-1).
- Zero means low similarity(the data objects are dissimilar). One means high similarity(the data objects are very similar).



- Lets consider three data points A, B, and C. Each data sample can have a single value on one axis(because we only have one input feature); let's denote that as the x-axis.
- Let's take two points, A(0.5), B(1), and C(30). As you can tell, A and B are close enough to each other in contrast to C.
- Thus, the similarity between A and B is higher than A and C or B and C. In other terms, A and B have a strong correlation. Therefore, the smaller the distance is, the larger the similarity will get.

21

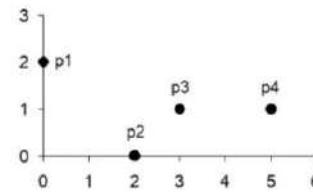
Similarity and Dissimilarity for Object with Single Attribute

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{ p-q }{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{ p-q }{n-1}$
Interval or Ratio	$d = p - q $	$s = -d, s = \frac{1}{1+d} \text{ or } s = 1 - \frac{d - \min.d}{\max.d - \min.d}$

22

Similarity and Dissimilarity for Object with Multiple Numeric Attribute

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1



	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

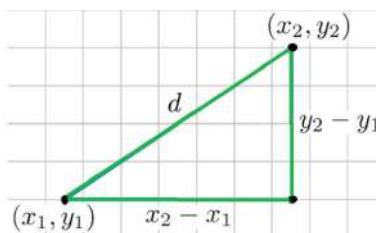
23

1. Euclidean distance

➤ Euclidean Distance represents the shortest distance between two points.

➤ It is one of the most used algorithms in the cluster analysis.

$$\begin{aligned}
 d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\
 &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}
 \end{aligned}$$



- ❖ Euclidean distance works great when you have low-dimensional data and the magnitude of the vectors is important to be measured.
- ❖ It is not working when there is an obstacle between two points

24

Example:

Find the distance between two points P(0, 4) and Q(6, 2).

Solution:

Given:

$$P(0, 4) = (x_1, y_1)$$

$$Q(6, 2) = (x_2, y_2)$$

The distance between the point PQ is

$$PQ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$PQ = \sqrt{(6 - 0)^2 + (2 - 4)^2}$$

$$PQ = \sqrt{(6)^2 + (-2)^2}$$

$$PQ = \sqrt{36+4}$$

$$PQ = \sqrt{40}$$

$$PQ = 2\sqrt{10}.$$

Therefore, the distance between two points P(0,4) and Q(6, 2) is $2\sqrt{10}$.

25

- ❖ Consider a dataset that contains two variables: height (cm) & weight (kg). Each point is classified as normal or underweight.

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	Normal
69	176	Normal
64	173	Normal
65	172	Normal
56	174	Underweight
58	169	Normal
57	173	Normal
55	170	Normal

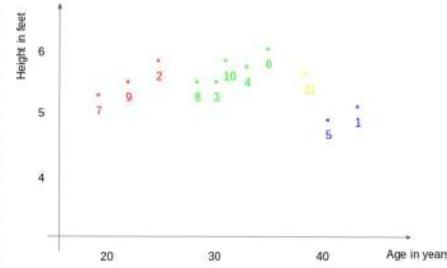
57 kg	170 cm	?
-------	--------	---

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

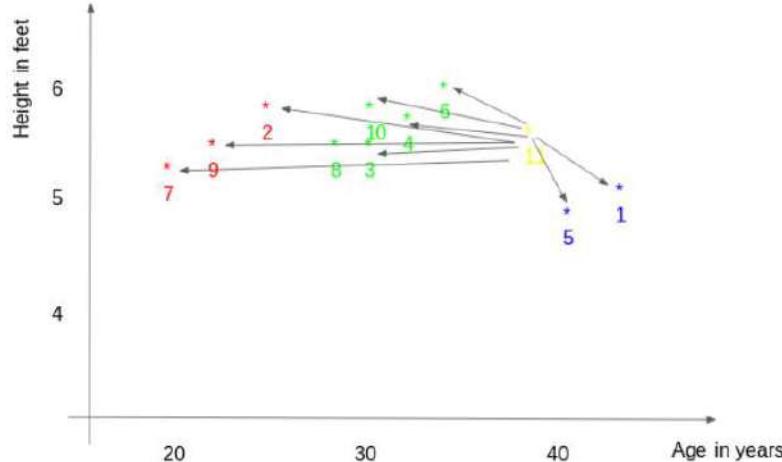
26

- ❖ Consider the following table – it consists of the height, age and weight (target) value for 10 people. As you can see, the weight value of ID11 is missing. We need to predict the weight of this person based on their height and age.

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?



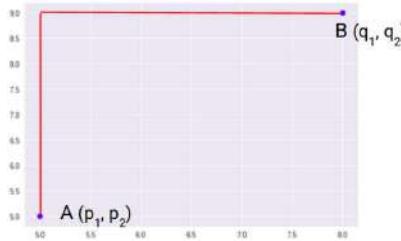
27



28

2. Manhattan Distance

- Manhattan Distance is the sum of absolute differences between points across all the dimensions.



$$d = |p_1 - q_1| + |p_2 - q_2| \quad D_m = \sum_{i=1}^n |p_i - q_i|$$

29

3. Minkowski Distance

- Minkowski Distance is the generalized form of Euclidean and Manhattan Distance.

$$D = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}$$

where n is number of dimension and p is an integer.

- The case where p = 1 is equivalent to the Manhattan distance and the case where p = 2 is equivalent to the Euclidean distance.
- Although p can be any real value, it is typically set to a value between 1 and 2.

30

4. Hamming Distance

- Hamming Distance measures the similarity between two strings of the same length.
- The Hamming Distance between two strings of the same length is the **number of positions at which the corresponding characters are different**.

A	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>3</td><td>2</td><td>7</td><td>4</td><td>6</td><td>5</td><td>8</td><td>1</td></tr></table>	3	2	7	4	6	5	8	1	Hamming Distance(A, B) = 4
3	2	7	4	6	5	8	1			
B	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td><td>7</td><td>3</td><td>9</td><td>5</td><td>1</td><td>8</td></tr></table>	1	2	7	3	9	5	1	8	
1	2	7	3	9	5	1	8			
A	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>"Geeks"</td><td>"for"</td><td>"Geeks"</td><td>"Java"</td><td>"C++"</td><td>"Swift"</td><td>"R"</td><td>"Python"</td></tr></table>	"Geeks"	"for"	"Geeks"	"Java"	"C++"	"Swift"	"R"	"Python"	Hamming Distance(A, B) = 2
"Geeks"	"for"	"Geeks"	"Java"	"C++"	"Swift"	"R"	"Python"			
B	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>"Geeks"</td><td>"for"</td><td>"Geeks"</td><td>"C++"</td><td>"C"</td><td>"Swift"</td><td>"R"</td><td>"Python"</td></tr></table>	"Geeks"	"for"	"Geeks"	"C++"	"C"	"Swift"	"R"	"Python"	
"Geeks"	"for"	"Geeks"	"C++"	"C"	"Swift"	"R"	"Python"			

31

“euclidean” and “manhattan”

- Here seven characters are different whereas two characters (the last two characters) are similar:
- ❖ Hence, the Hamming Distance here will be **7**. Note that larger the Hamming Distance between two strings, more dissimilar will be those strings (and vice versa).

32

Limitations

- Hamming distance is difficult to use when two vectors are not of equal length.
- It does not take the actual value into account as long as they are different or equal. Therefore, it is not advised to use this distance measure when the magnitude is an important measure.

Use Cases

- Typical use cases include error correction/detection when data is transmitted over computer networks. It can be used to determine the number of distorted bits in a binary word as a way to estimate error.
- Moreover, you can also use Hamming distance to measure the distance between categorical variables.

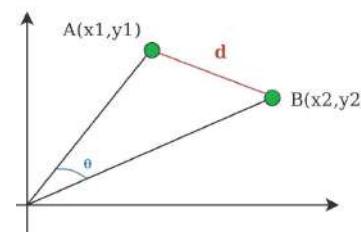
33

5. Cosine Similarity

- Cosine similarity has often been used as a way to counteract Euclidean distance's problem with high dimensionality.
- The cosine similarity is simply the cosine of the angle between two vectors.
- Two vectors with exactly the same orientation have a cosine similarity of 1, whereas two vectors diametrically opposed to each other have a similarity of -1.
- Note that their **magnitude is not of importance** as this is a measure of orientation.

$$D(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|}$$

Here (**theta**) gives the angle between two vectors and A, B are n-dimensional vectors.



Limitation

- One main disadvantage of cosine similarity is that the magnitude of vectors is not taken into account, merely their direction. In practice, this means that the differences in values are not fully taken into account.

Use Case

- We use cosine similarity often when we have **high-dimensional** data and when the **magnitude of the vectors is not of importance**.
- Cosine similarity is a metric used to measure how similar the documents are irrespective of their size.
- For text analyses, this measure is quite frequently used when the data is represented by word counts. For example, when a word occurs more frequently in one document over another this does not necessarily mean that one document is more related to that word. It could be the case that documents have uneven lengths and the magnitude of the count is of less importance.

35

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

$$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$$

$$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$$

Using the formula, $\cos(d_1, d_2) = (d_1 \cdot d_2) / ||d_1|| ||d_2||$,

$$d_1 \cdot d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$$

$$||d_1|| = (5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 1^2 + 0^2 + 1^2)^{0.5} = (42)^{0.5} = 6.481$$

$$||d_2|| = (3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2)^{0.5} = (17)^{0.5} = 4.12$$

$$\cos(d_1, d_2) = 0.94$$

The cosine similarity shows that the two documents are quite similar.

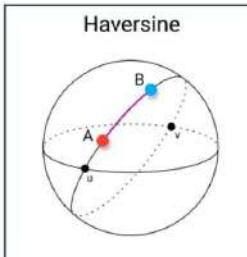
36

6. Haversine

- Haversine distance is the distance between two points on a sphere given their longitudes and latitudes.
- It is very similar to Euclidean distance in that it calculates the shortest line between two points.
- The main difference is that no straight line is possible since the assumption here is that the two points are on a sphere.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

where r is the radius of the earth(6371 km), d is the distance between two points, ϕ_1, ϕ_2 is the latitude of the two points, and λ_1, λ_2 is the longitude of the two points respectively.



37

	Places_X	lat_X	long_X		Places_Y	lat_Y	long_Y
0	Cafe	38.47	27.08	0	Airport	41.365	27.14
1	Restaurant	38.46	27.24	1	Hotel	41.300	26.61
2	Cathedral	38.41	27.16	2	Train_Station	41.040	26.63
3	Museum	41.06	28.99				

	Places_X	Places_Y	distance
0	Cafe	Airport	200.062763
1	Restaurant	Airport	200.798395
2	Cathedral	Airport	204.186115
3	Museum	Airport	98.443339
4	Cafe	Hotel	197.126628
5	Restaurant	Hotel	199.058375
6	Cathedral	Hotel	201.810501
7	Museum	Hotel	124.876004
8	Cafe	Train_Station	179.181669
9	Restaurant	Train_Station	181.191731
10	Cathedral	Train_Station	183.895345
11	Museum	Train_Station	122.961324

38

Disadvantages

- One disadvantage of this distance measure is that it is assumed the points lie on a sphere.

Use Cases

- As you might have expected, Haversine distance is often used in navigation. For example, you can use it to calculate the distance between two countries when flying between them.

39

7. Jaccard Index

- The Jaccard Similarity Index is a measure of the similarity between two sets of data.
- The index ranges from 0 to 1. The closer to 1, the more similar the two sets of data.
- The Jaccard index (or Intersection over Union) is a metric used to calculate the similarity and diversity of sample sets. It is the size of the intersection divided by the size of the union of the sample sets.
- To calculate the Jaccard distance we simply subtract the Jaccard index from 1:

$$D(x, y) = 1 - \frac{|x \cap y|}{|y \cup x|}$$

40

A = [0, 1, 2, 5, 6, 8, 9]

B = [0, 2, 3, 4, 5, 7, 9]

Number of observations in both: {0, 2, 5, 9} = 4

Number of observations in either: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} = 10

Jaccard Similarity: 4 / 10 = 0.4

The Jaccard Similarity Index turns out to be 0.4.

C = [0, 1, 2, 3, 4, 5]

D = [6, 7, 8, 9, 10]

Number of observations in both: {} = 0

Number of observations in either: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10} = 11

Jaccard Similarity: 0 / 11 = 0

The Jaccard Similarity Index turns out to be 0. This indicates that the two datasets share no common members.

41

E = ['cat', 'dog', 'hippo', 'monkey']

F = ['monkey', 'rhino', 'ostrich', 'salmon']

Number of observations in both: {'monkey'} = 1

Number of observations in either: {'cat', 'dog', 'hippo', 'monkey', 'rhino', 'ostrich', 'salmon'} = 7

Jaccard Similarity: 1 / 7 = 0.142857

The Jaccard Similarity Index turns out to be **0.142857**. Since this number is fairly low, it indicates that the two sets are quite dissimilar.

Jaccard distance = 1 – Jaccard Similarity

42

Disadvantages

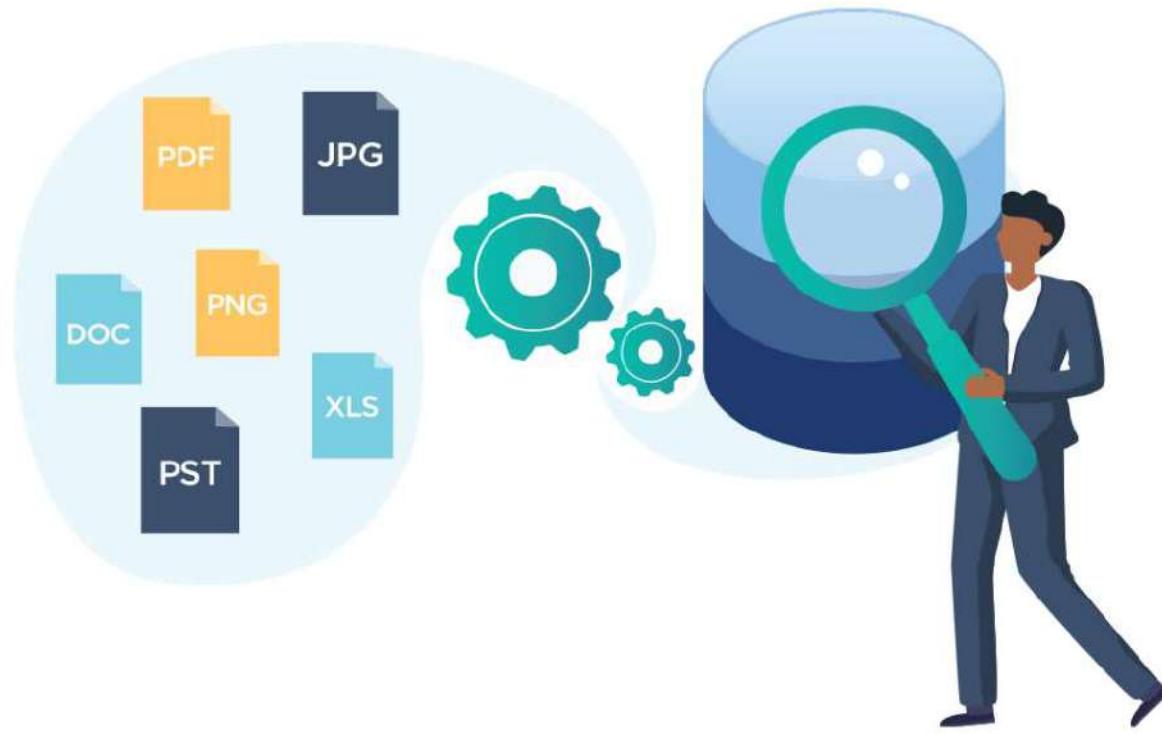
- A major disadvantage of the Jaccard index is that it is highly influenced by the size of the data. Large datasets can have a big impact on the index as it could significantly increase the union while keeping the intersection similar.

Use-Cases

- The Jaccard index is often used in applications where **binary or binarized data are used**. When you have a deep learning model predicting **segments of an image**, for instance, a car, the Jaccard index can then be used to calculate how accurate that predicted segment given true labels.
- Similarly, it can be used in text similarity analysis to measure **how much word choice overlap** there is between documents. Thus, it can be used to compare sets of patterns.

43

Data Mining (20CP306T)



Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

Why Data Preprocessing?

- Data in the real world is dirty
 - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - **noisy**: containing errors or outliers
 - **inconsistent**: containing discrepancies in codes or names
- No quality data, no quality mining results!
 - Quality decisions must be based on quality data
 - Data warehouse needs consistent integration of quality data
- A multi-dimensional measure of data quality:
 - A well-accepted multi-dimensional view:
 - accuracy, completeness, consistency, timeliness, believability, value added, interpretability, accessibility

Data Preprocessing



Major Tasks in Data Preprocessing

➤ Data cleaning

- Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

➤ Data integration

- Integration of multiple databases, data cubes, files, or notes

➤ Data transformation

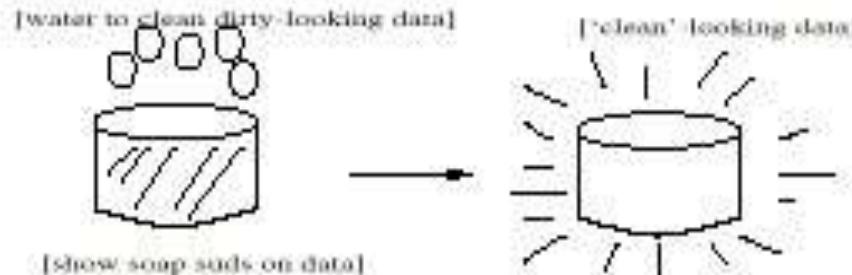
- Normalization (scaling to a specific range)
- Aggregation

➤ Data reduction

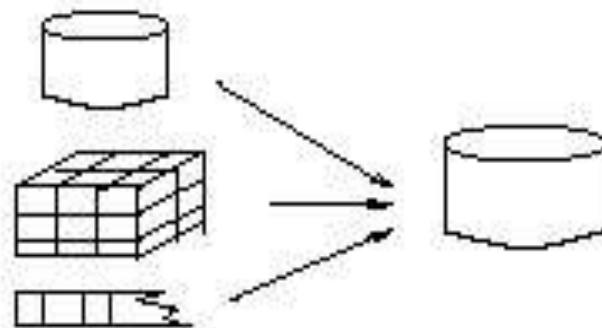
- Obtains reduced representation in volume but produces the same or similar analytical results
- Data aggregation, dimensionality reduction, data compression, generalization

Forms of data preprocessing

Data Cleaning



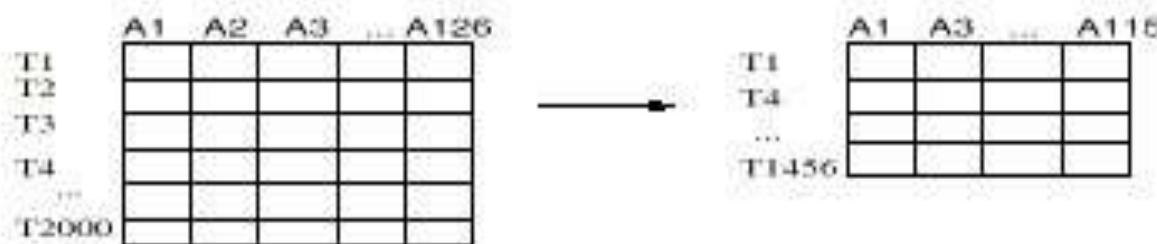
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



1. Data Cleaning

- Data cleaning tasks
 - Fill in missing values
 - Identify outliers
 - smooth out noisy data
 - Correct inconsistent data

Missing Data

- Data is not always available
 - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the task is classification—not effective in certain cases)
- Fill in the missing value **manually**: tedious + infeasible?
- Use a **global constant** to fill in the missing value: e.g., “unknown”, a new class?!
- Use the **attribute mean** to fill in the missing value
- Use the **attribute mean for all samples of the same class** to fill in the missing value: smarter
- Use the **most probable value** to fill in the missing value:
inference-based such as regression, Bayesian formula, decision tree

Noisy Data

- A Random error in a measured variable.
- Incorrect attribute values may be due to
 - faulty data collection instruments
 - data entry problems
 - data transmission problems
 - technology limitation
 - inconsistency in naming convention
- Other data problems which requires data cleaning
 - duplicate records
 - incomplete data
 - inconsistent data

How to Handle Noisy Data?

- Binning method:
 - first sort data and partition into (equi-depth) bins
 - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Semi-automated method: combined computer and human inspection
 - detect suspicious values and check manually
- Regression
 - smooth by fitting the data into regression functions
- Clustering
 - detect and remove outliers

Simple Discretization Methods: Binning

➤ Equal-width (distance) partitioning:

- It divides the range into N intervals of equal size: uniform grid
- if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B-A)/N$.
- But outliers may dominate presentation
- Skewed data is not handled well.
- Not suitable for categorical attributes

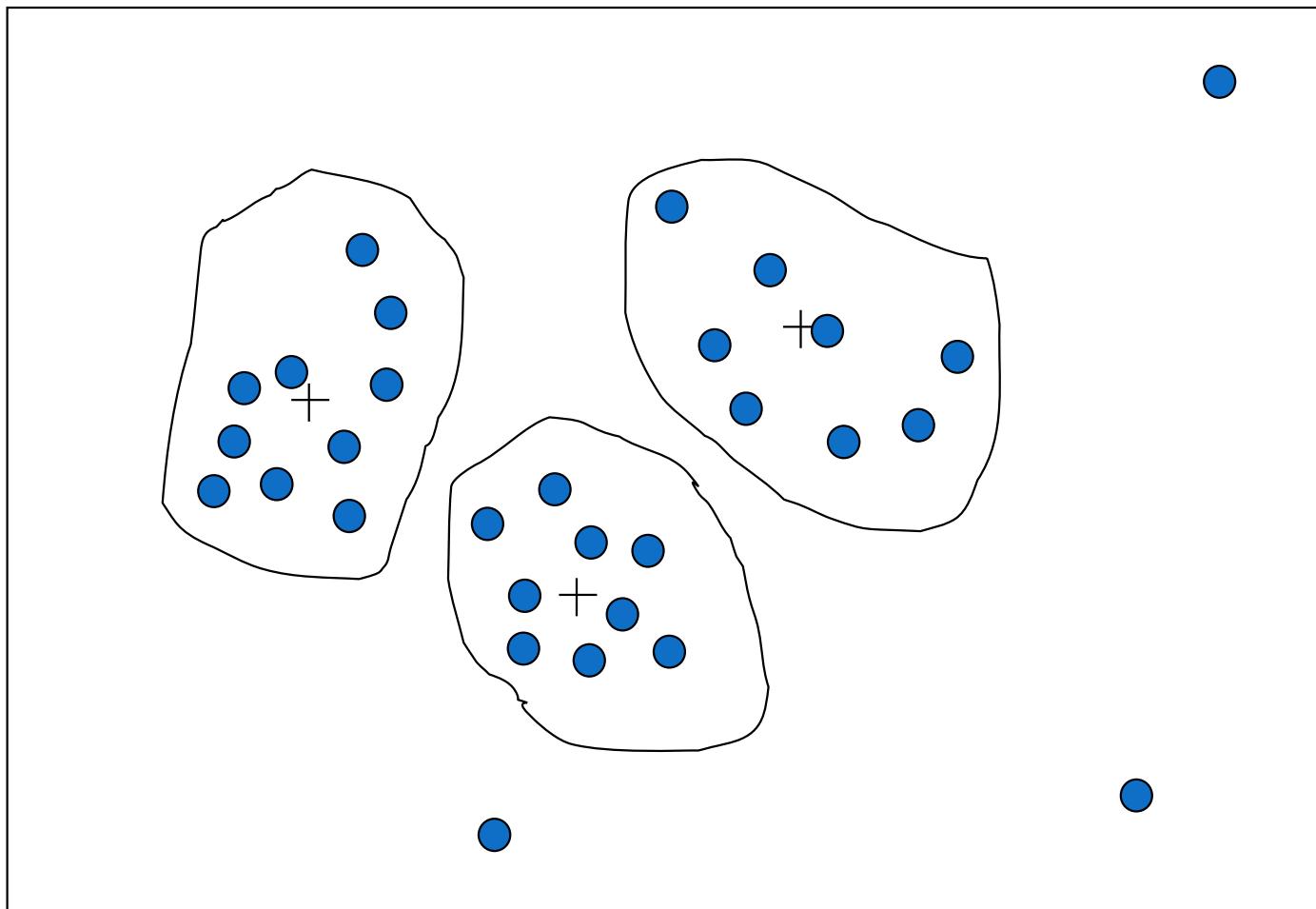
➤ Equal-depth (frequency) partitioning:

- It divides the range into N intervals, each containing approximately same number of samples
- Good data scaling
- Managing categorical attributes can be tricky.

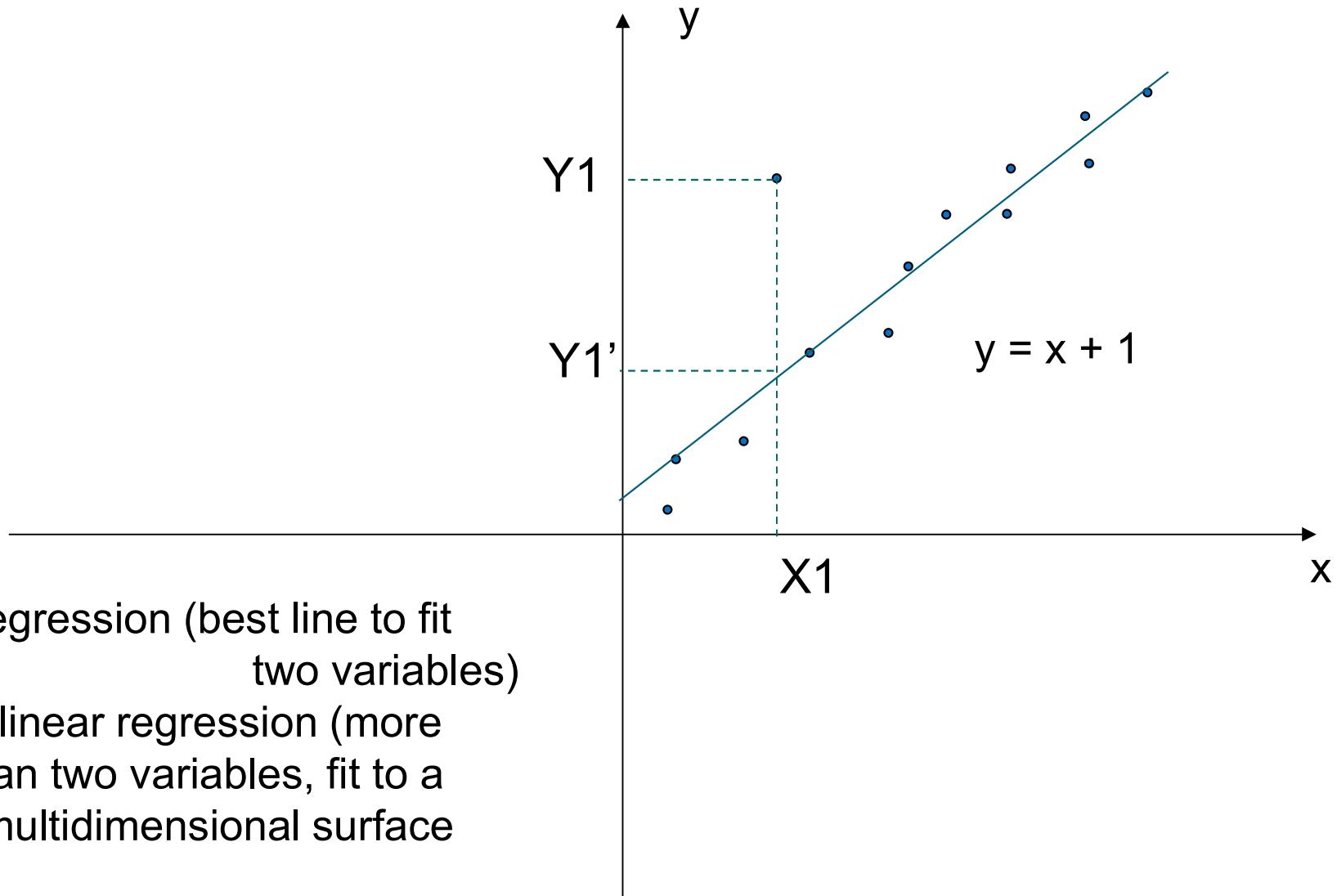
Binning Methods for Data Smoothing

- * Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- * Partition into (equi-depth) bins:
 - Bin 1: 4, 8, 9, 15
 - Bin 2: 21, 21, 24, 25
 - Bin 3: 26, 28, 29, 34
- * Smoothing by bin means:
 - Bin 1: 9, 9, 9, 9
 - Bin 2: 23, 23, 23, 23
 - Bin 3: 29, 29, 29, 29
- * Smoothing by bin boundaries:
 - Bin 1: 4, 4, 4, 15
 - Bin 2: 21, 21, 25, 25
 - Bin 3: 26, 26, 26, 34

Cluster Analysis



Regression



- Linear regression (best line to fit two variables)
- Multiple linear regression (more than two variables, fit to a multidimensional surface)

How to Handle Inconsistent Data?

- Manual correction using external references
- Semi-automatic using various tools
 - To detect violation of known functional dependencies and data constraints
 - To correct redundant data

2. Data Integration

- Data integration:
 - combines data from multiple sources into a coherent store
- Schema integration
 - integrate metadata from different sources
 - Entity identification problem: identify real world entities from multiple data sources
- Detecting and resolving data value conflicts
 - for the same real world entity, attribute values from different sources are different
 - possible reasons: different representations, different scales, e.g., metric vs. British units, different currency

Handling Redundant Data in Data Integration

- Redundant data occur often when integrating multiple DBs
 - The same attribute may have different names in different databases
 - One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant data may be able to be detected by **correlational analysis**

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

\bar{x} = mean of the values of the x-variable

y_i = values of the y-variable in a sample

\bar{y} = mean of the values of the y-variable

- Careful integration can help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

3. Data Transformation

- Smoothing: remove noise from data (binning, clustering, regression)
- Aggregation: summarization
- Normalization: scaled to fall within a small, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
- Attribute/feature construction
 - New attributes constructed from the given ones

Data Normalization or Standardization

- Data normalization or standardization is defined as the **process of rescaling original data without changing its behavior or nature.**
- We define new boundary (most common is $(0,1),(-1,1)$) and convert data accordingly.
- Standardization rescales data to have **a mean (μ) of 0 and standard deviation (σ) of 1 (unit variance).** The result of standardization (or Z-score normalization) is that the features will be rescaled so that they'll have the properties of a standard normal distribution with $\mu=0$ and $\sigma=1$

$$z = \frac{x - \mu}{\sigma}$$

z-score normalization

$$Z = \frac{x - \mu}{\sigma}$$

Diagram illustrating the components of the z-score formula:

- Score (x)
- Mean (μ)
- SD (Standard Deviation) (σ)

The diagram shows the Score and Mean boxes connected by arrows pointing to the subtraction operation. The SD box is connected by an arrow pointing to the division operation.

- A negative z-score indicates that the data point is below the mean.
- A positive z-score indicates that the data point is above the mean.

How to calculate Z-Score

marks
8
10
15
20

marks	marks after z-score normalization
8	-1.14
10	-0.7
15	0.3
20	1.4

$$s = \sqrt{\frac{\sum(x - \bar{x})^2}{n}}$$

Standard deviation = $\sqrt{\frac{\sum(\text{every individual value of marks} - \text{mean of marks})^2}{n}}$

Mean of marks = $8 + 10 + 15 + 20 / 4 = 13.25$

$$\begin{aligned} &= \sqrt{\frac{(8 - 13.25)^2 + (10 - 13.25)^2 + (15 - 13.25)^2 + (20 - 13.25)^2}{4}} \\ &= \sqrt{\frac{(-5.25)^2 + (-3.25)^2 + (1.75)^2 + (6.75)^2}{4}} \\ &= \sqrt{\frac{27.56 + 10.56 + 3.06 + 45.56}{4}} = \sqrt{\frac{86.74}{4}} = \sqrt{21.6} = 4.6 \end{aligned}$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{8 - 13.25}{4.6} = -1.14$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{10 - 13.25}{4.6} = -0.7$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{15 - 13.25}{4.6} = 0.3$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{20 - 13.25}{4.6} = 1.4$$

- Normalization rescales the values into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale. However, the outliers from the data set are lost.

Min-Max scaling

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- In some situations, we may prefer to map data to a range like [-1,1] with zero-mean.

$$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$

4. Data Reduction

➤ Problem:

- Data Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set

➤ Solution?

- Data reduction...

Data Reduction

- Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Data reduction strategies
 - Data cube aggregation
 - Dimensionality reduction
 - Data compression

Data Cube Aggregation

- Multiple levels of aggregation in data cubes
 - Further reduce the size of data to deal with
- Reference appropriate levels
 - Use the smallest representation capable to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

Dimensionality Reduction

- **Problem:** Feature selection (i.e., attribute subset selection):
 - Select a **minimum set of features** such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
- **Solution:** Heuristic methods (due to exponential # of choices) usually greedy:
 - step-wise forward selection
 - step-wise backward elimination
 - combining forward selection and backward elimination
 - decision-tree induction

Principal Component Analysis (PCA)

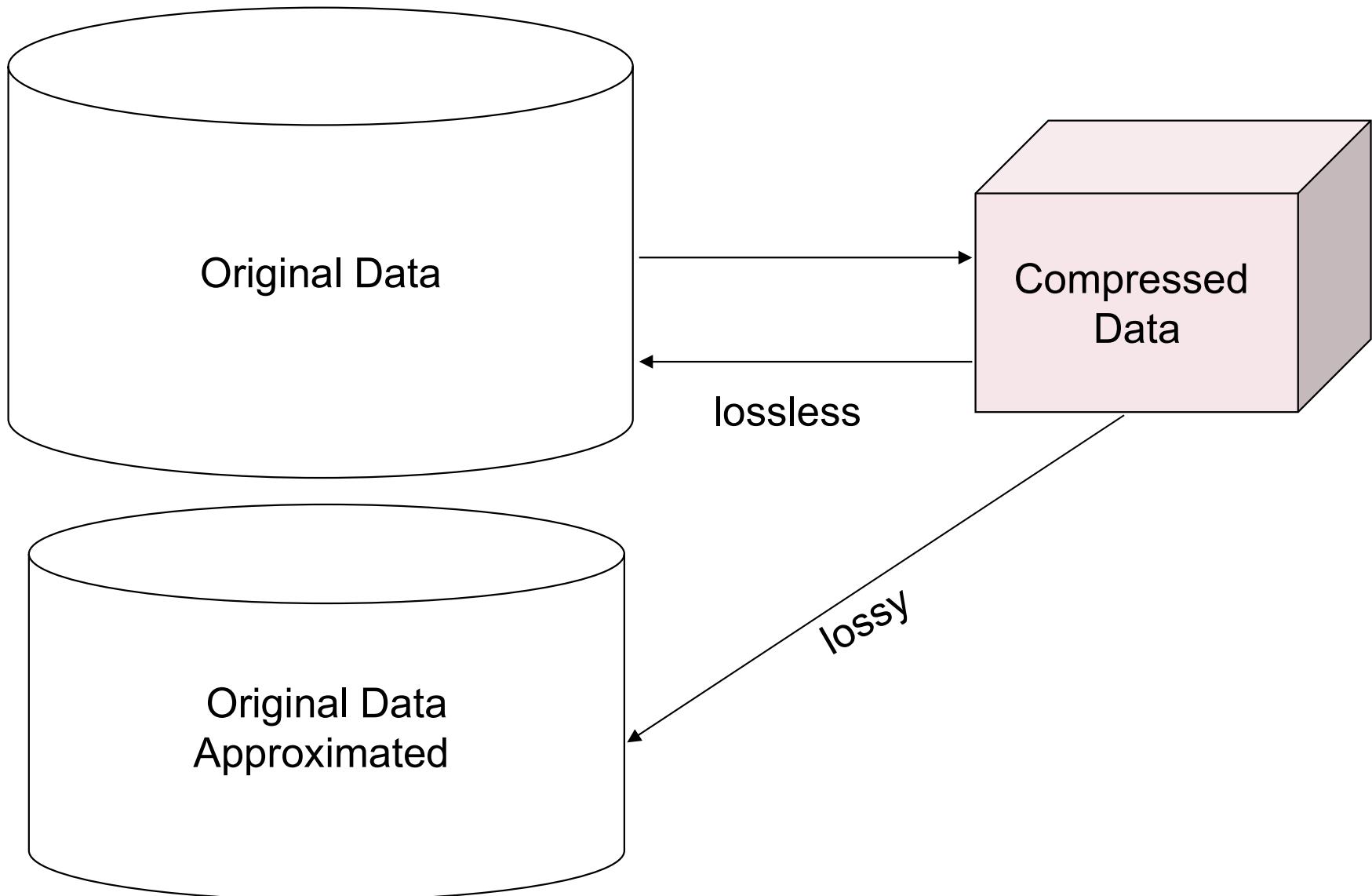
- Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets.

- PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.

Data Compression

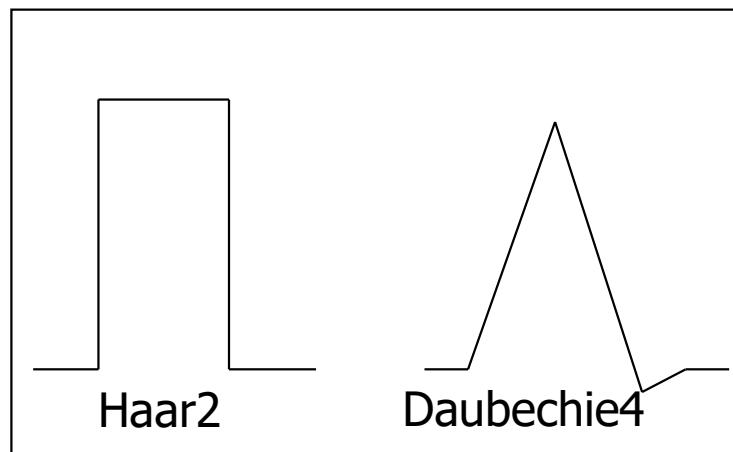
- String compression
 - There are extensive theories and well-tuned algorithms
 - Typically lossless
 - But only limited manipulation is possible without expansion
- Audio/video, image compression
 - Typically lossy compression, with progressive refinement
 - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
 - Typically short and vary slowly with time

Data Compression



Wavelet Transforms

- Discrete wavelet transform (DWT): linear signal processing
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space (conserves local details)



Statistics

- *Statistics is a branch of mathematics that deals with collecting, organization and interpretation of data.*

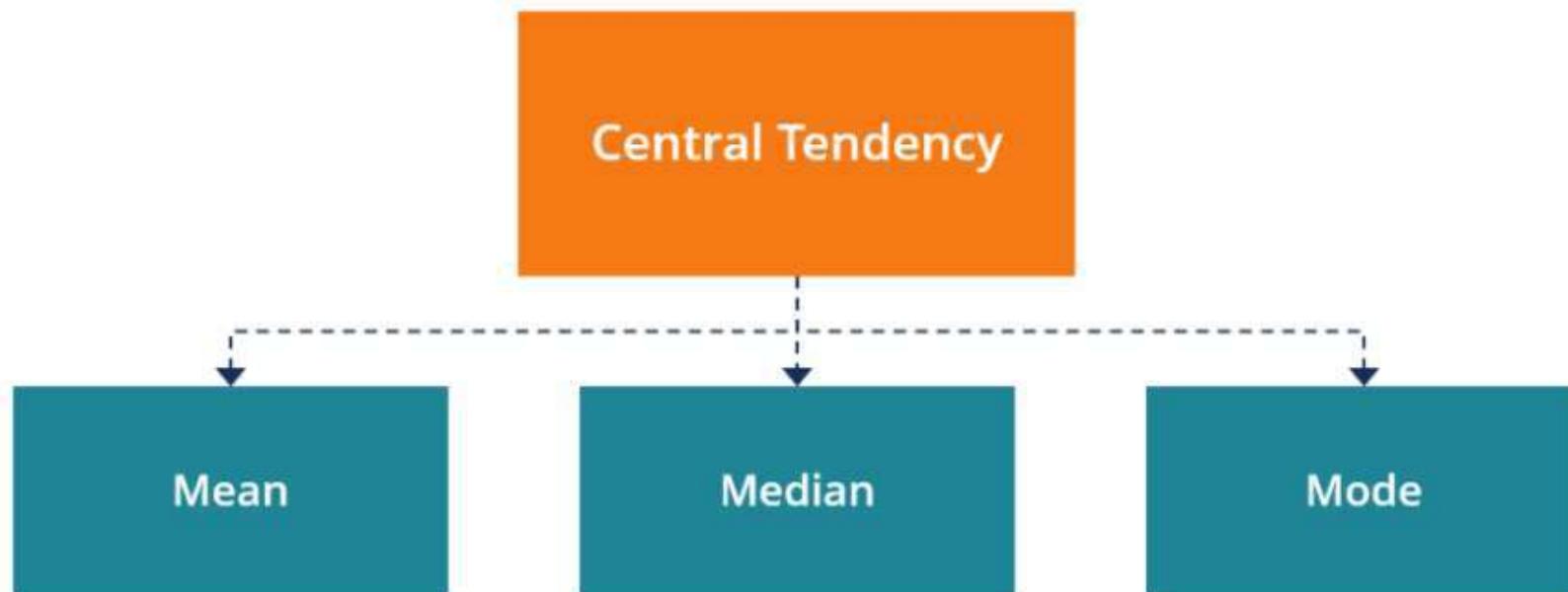
- **1. Descriptive Statistics:** In Descriptive Statistics you are describing, presenting, summarizing and organizing your data (population), either through numerical calculations or graphs or tables.

- **2. Inferential statistics:** Inferential Statistics are produced by more complex mathematical calculations, and allow us to infer trends and make **assumptions and predictions about a population based on a study of a sample taken from it.**

Central Tendency

- Central tendency is a **descriptive** summary of a dataset through a single value that reflects the center of the data distribution.

The three measures of the central tendency are commonly used is:-



Mean

- Mean is the average of central tendency and is the most commonly used measures.

Arithmetic mean: The arithmetic mean of a set of observations is defined to be their sum, divided by the number of observations.

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i$$

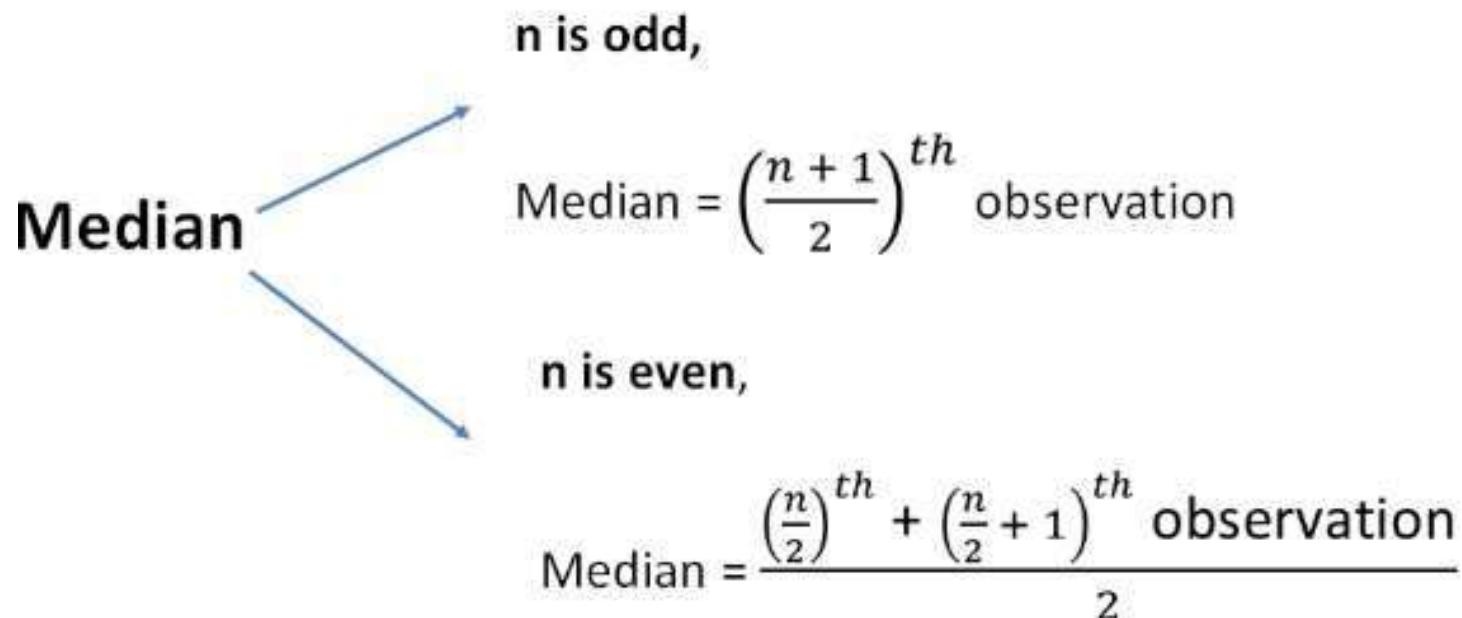
Weighted A.M.: For frequency distribution, where x_i have frequencies.
($i=1,2,3\dots$)

$$\bar{x} = \frac{f_1x_1 + f_2x_2 + \cdots + f_nx_n}{f_1 + f_2 + \cdots + f_n} = \frac{1}{N} \sum f_i x_i$$

- ❖ Mean is **highly susceptible** to outliers.

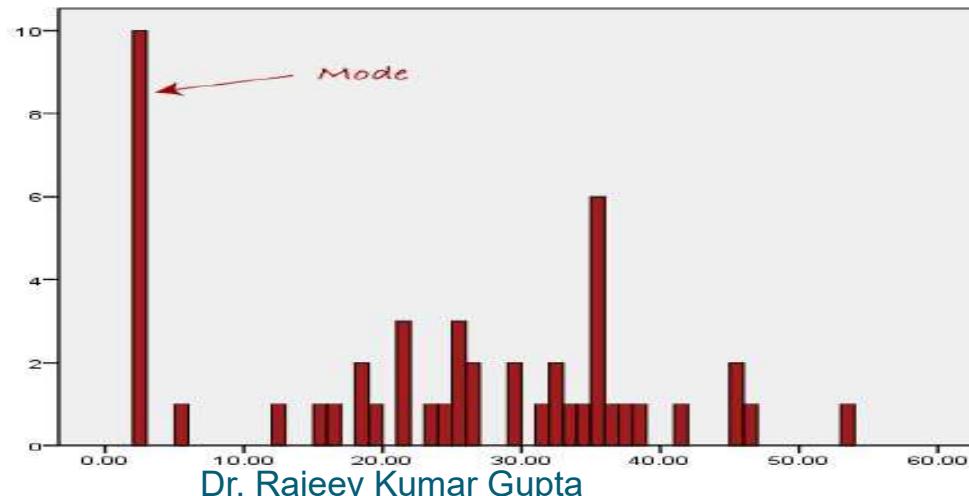
Median

- The way you compute the median of the dataset is by sorting all the values (in either ascending or descending order), and taking the one that ends up in the middle.
- Median is **less susceptible** to outliers than the mean.
- For example, you can talk about the mean or average household income in India



Mode

- The mode is a statistical term that refers to the most frequently occurring number found in a set of numbers.
- The mode can be the same value as the mean and/or median but this is not always the case.
- Mode is usually only relevant to **discrete numerical data**, and not to **continuous data**.
- It is not suitable for
 - For instance, the average frequency of people born with six fingers is something like 0.01%, but the mode is zero since the most common outcome is five fingers.
 - If most common mark is far away from the rest of the data in the data set



Effects of Outliers

1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 4

Mean = 2.58

Median = 2.5

Mode = 2

Standard Deviation = 1.08

$$s = \sqrt{\frac{\sum(X - \bar{x})^2}{n - 1}}$$

If we add an outlier to the data set:

1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 400

Mean = 35.38

Median = 2.5

Mode = 2

Standard Deviation = 114.74

- ❖ As you can see, having outliers often has a significant effect on your mean and standard deviation. Because of this, we must take steps to remove outliers from our data sets.
- ❖ If you have outliers, use median and mean.

Measure Variability

- A **population** includes all of the elements from a set of data.
- A **sample** consists one or more observations drawn from the population.
 - More than one sample can be derived from the same population.
 - A measurable characteristic of a population, such as a mean or standard deviation, is called a **parameter**; but a measurable characteristic of a sample is called a **statistic**.
- Statisticians use summary measures to describe the amount of variability or spread in a set of data. The most common measures of variability are the range, the interquartile range (IQR), variance, and standard deviation.

Range

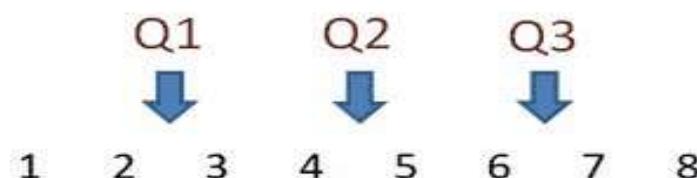
- The **range** is the difference between the largest and smallest values in a set of values.
 - For example, consider the following numbers: 1, 3, 4, 5, 5, 6, 7, 11. For this set of numbers, the range would be $11 - 1$ or 10.

Interquartile Range (IQR)

- The interquartile range (IQR) is a measure of variability, based on dividing a data set into quartiles.
- Quartiles divide a rank-ordered data set into four equal parts. The values that divide each part are called the first, second, and third quartiles; and they are denoted by Q1, Q2, and Q3, respectively.
- For non-normally distributed data and/or data with extreme outliers, the IQR and Mdn provide a better way to figure out the way scores are distributed.

- Q1 is the "middle" value in the first half of the rank-ordered data set.
- Q2 is the median value in the set.
- Q3 is the "middle" value in the second half of the rank-ordered data set.
- The interquartile range is equal to Q3 minus Q1.

$$\text{IQR} = 6.5 - 2.5 = 4.$$



34, 47, 1, 15, 57, 24, 20, 11, 19, 50, 28, 37.

- the median
- the range
- the upper and lower quartiles
- the interquartile range

1,11,15,19,20,24, 28,34,37,47,50,57

Median = (sixth + seventh observations) \div 2

$$= (24 + 28) \div 2 = 26$$

- Range = difference between the highest and lowest values
 $= 57 - 1 = 56$
- Lower quartile = value of middle of first half of data Q_1
 $= (15 + 19) \div 2 = 17$
- Upper quartile = value of middle of second half of data Q_3
 $=$ the median of 28, 34, 37, 47, 50, 57
 $= (\text{third} + \text{fourth observations}) \div 2$
 $= (37 + 47) \div 2 = 42$
- Interquartile range = $Q_3 - Q_1$
 $= 42 - 17 = 25$
- The interquartile range is the best measure of variability for **skewed distributions or data sets with outliers**. Because it's based on values that come from the middle half of the distribution, it's unlikely to be influenced by outliers.

Variance and Standard deviation

- **Variance** is defined and calculated as the average squared deviation from the mean. **Standard deviation** is calculated as the square root of variance.
- Not suitable for skewed numerical data. In this case median and interquartile range are used.

Population Variance

$$\sigma^2 = \frac{\sum_{i=1}^n (X_i - X_{avg})^2}{n}$$

Population Standard Deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - X_{avg})^2}{n}}$$

Sample Variance

$$s^2 = \frac{\sum_{i=1}^n (X_i - X_{avg})^2}{n - 1}$$

Sample Standard Deviation

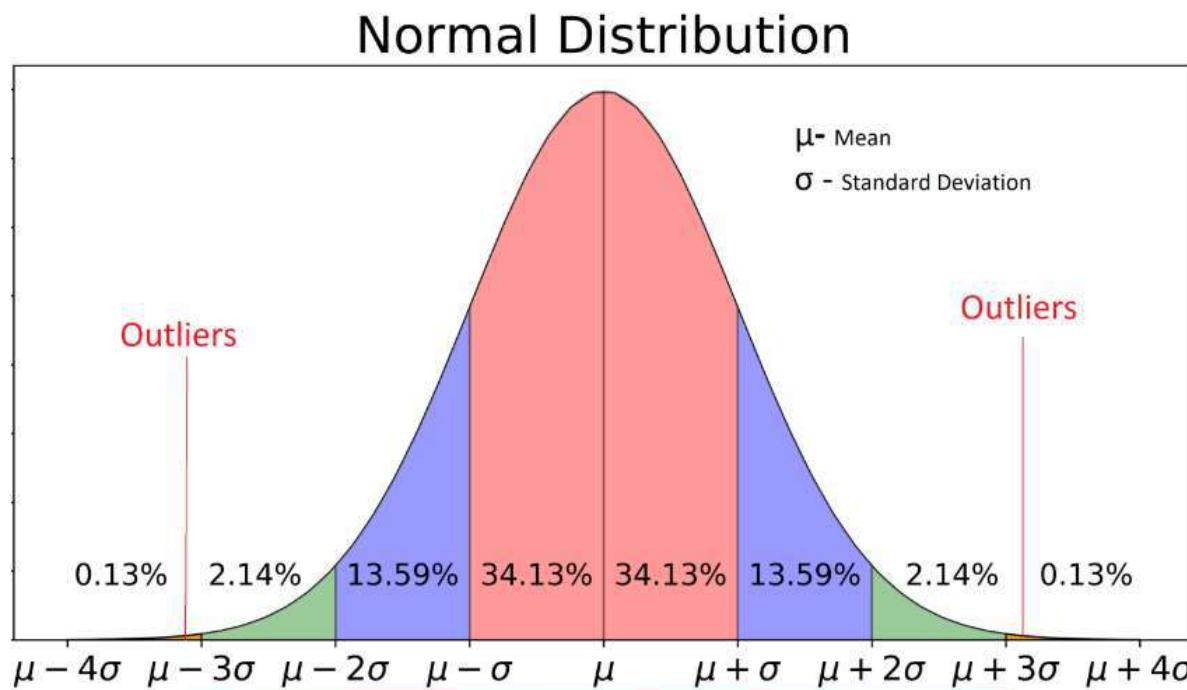
$$s = \sqrt{\frac{\sum_{i=1}^n (X_i - X_{avg})^2}{n - 1}}$$

Effect of Changing Units

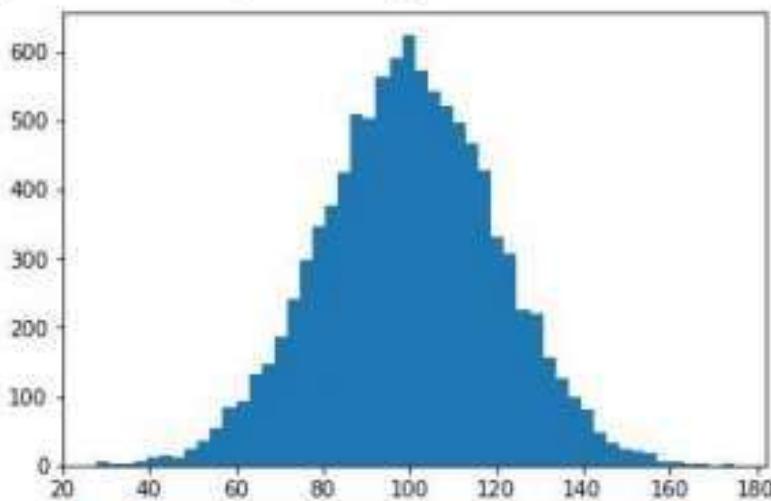
- If you add a constant to every value, the distance between values does not change. As a result, all of the measures of variability (range, interquartile range, standard deviation, and variance) and median **remain the same**.
- On the other hand, suppose you multiply every value by a constant. This has the effect of multiplying the range, interquartile range (IQR), standard deviation, mean and median **by that constant**.
- It has an even greater effect on the variance. It multiplies the variance by the square of the constant

Identify Outliers Using Standard Deviation

- **Standard Deviation :** If any data point that is more than 3 times of standard deviation, then those points are very likely to be treated as outliers, In general, if data distribution is approximately normal, then about **68%** of the data values lie within **one standard deviation of the mean**, and about **95%** are within **two standard deviations**, and about **99.7%** lie within **three standard deviations**.



Analysing standard deviation and variance on a histogram



This is a histogram which is centered around 100 (mean) with a standard deviation of 20 and 10,000 data points.

- Most common occurrence is around 100, and as we get further and further from that, things become less and less likely. The standard deviation point of 20 that we specified is around 80 and around 120. You can see in the histogram that this is the point where things start to fall off sharply.
- So we can say that things beyond that standard deviation boundary are unusual.

Effect of Mean and Standard Deviation on Curve

- Overall shape of a distribution of a large number of observations can be summarized by a smooth curve called a **density curve**.
- Know that **changing the mean** of a normal density curve **shifts the curve along the horizontal axis without changing its shape**.
- Know that **increasing the standard deviation** produces a **flatter and wider bell-shaped** curve and that decreasing the standard deviation produces a **taller and narrower curve**.
- <https://www.geogebra.org/m/QrNaGua4>

Data Mining Potential Applications

- Data analysis and decision support
 - Market analysis and management
 - Target marketing, customer relationship management (CRM), market basket analysis, cross selling, market segmentation
 - Risk analysis and management
 - Forecasting, customer retention, improved underwriting, quality control, competitive analysis
 - Fraud detection and detection of unusual patterns (outliers)
- Other Applications
 - Text mining (news group, email, documents) and Web mining
 - Stream data mining
 - DNA and bio-data analysis

Market Analysis and Management

- Where does the data come from?
 - Credit card transactions, loyalty cards, discount coupons, customer complaint calls, plus (public) lifestyle studies
- Target marketing
 - Find clusters of “model” customers who share the same characteristics: interest, income level, spending habits, etc.
 - Determine customer purchasing patterns over time
- Cross-market analysis
 - Associations/co-relations between product sales, & prediction based on such association
- Customer profiling
 - What types of customers buy what products (clustering or classification)
- Customer requirement analysis
 - identifying the best products for different customers
 - predict what factors will attract new customers
- Provision of summary information
 - multidimensional summary reports
 - statistical summary information (data central tendency and variation)

Corporate Analysis & Risk Management

- Finance planning and asset evaluation
 - cash flow analysis and prediction
 - contingent claim analysis to evaluate assets
 - cross-sectional and time series analysis (financial-ratio, trend analysis, etc.)
- Resource planning
 - summarize and compare the resources and spending
- Competition
 - monitor competitors and market directions
 - group customers into classes and a class-based pricing procedure
 - set pricing strategy in a highly competitive market

Fraud Detection & Mining Unusual Patterns

- Approaches: Clustering & model construction for frauds, outlier analysis
- Applications: Health care, retail, credit card service, telecomm.
 - Auto insurance: ring of collisions
 - Money laundering: suspicious monetary transactions
 - Medical insurance
 - Professional patients, ring of doctors, and ring of references
 - Unnecessary or correlated screening tests
 - Telecommunications: phone-call fraud
 - Phone call model: destination of the call, duration, time of day or week. Analyze patterns that deviate from an expected norm
 - Retail industry
 - Analysts estimate that 38% of retail shrink is due to dishonest employees
 - Anti-terrorism

Major Issues in Data Mining

- Volume of Data
 - Clever algorithms needed for reasonable performance
- Interest measures
 - How do we ensure algorithms select “interesting” results?
- “Knowledge Discovery Process” skill required
 - How to select tool, prepare data?
- Data Quality
 - How do we interpret results in light of low quality data?
- Data Source Heterogeneity
 - How do we combine data from multiple sources?
- Handling noisy or incomplete data
 - Require data cleaning methods and data analysis methods that can handle noise

- **Performance Issues**
 - Efficiency and scalability
 - Huge amount of data
 - Running time must be predictable and acceptable
 - Parallel, distributed and incremental mining algorithms
 - Divide the data into partitions and processed in parallel
 - Incorporate database updates without having to mine the entire data again from scratch
- **Diversity of Database Types**
 - Other database that contain complex data objects, multimedia data, spatial data, etc.
 - Expect to have different DM systems for different kinds of data
 - Heterogeneous databases and global information systems
 - Web mining becomes a very challenging and fast-evolving field in data mining

Unit –II

Unit –II: SUPERVISED LEARNING

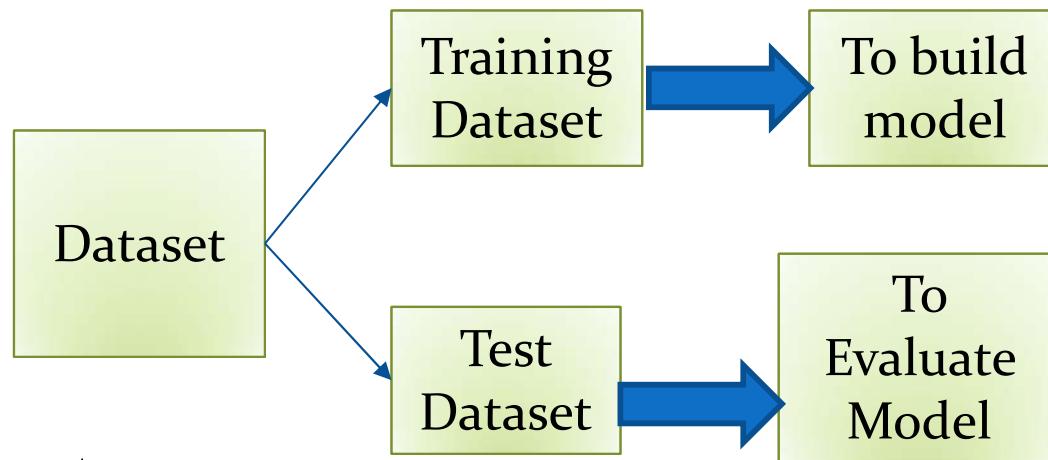
- ❖ Classification: Preliminaries; General approach to solving a classification problem; Decision tree induction; Rule-based classifier; Multilinear and Logistic Regression

Data Set

Datasets:

- A collection of instances
- Dataset consist of **feature matrix** and **target vector**

	Name	Age	Height	Weight	Class
1	Alfred	14	69	112.5	M
2	Alice	13	56.5	84	F
3	Barbara	13	65.3	98	F
4	Carol	14	62.8	102.5	F
5	Henry	14	63.5	102.5	M
6	James	12	57.3	83	M
7	Jane	12	59.8	84.5	F
8	Janet	15	62.5	112.5	F
9	Jeffrey	13	62.5	84	M
10	John	12	59	99.5	M
11	Joyce	11	51.3	50.5	F
12	Judy	14	64.3	90	F
13	Louise	12	56.3	77	F
14	Mary	15	66.5	112	F
15	Philip	16	72	150	M
16	Robert	12	64.8	128	M
17	Ronald	15	67	133	M
18	Thomas	11	57.5	85	M



Training set:

- Training set is used to build a model.
- It is used find relevant information on how to associate input data with output decision. The system is trained by applying these algorithms on the dataset, all the relevant information is extracted from the data and results are obtained.
- Generally, 70% of the data of the dataset is taken for training data.

Testing set:

- Testing data is used to test model. It is the set of data which is used to verify whether the system is producing the correct output after being trained or not. Generally, 30% of the data of the dataset is used for testing.

Iris Dataset



sepal_length	sepal_width	petal_length	petal_width	Iris_class
5	2	3.5	1	versicolor
6	2.2	4	1	versicolor
6.2	2.2	4.5	1.5	versicolor
6	2.2	5	1.5	virginica
4.5	2.3	1.3	0.3	setosa
5.5	2.3	4	1.3	versicolor
6.3	2.3	4.4	1.3	versicolor
5	2.3	3.3	1	versicolor
4.9	2.4	3.3	1	versicolor
5.5	2.4	3.8	1.1	versicolor
5.5	2.4	3.7	1	versicolor
5.6	2.5	3.9	1.1	versicolor
6.3	2.5	4.9	1.5	versicolor
5.5	2.5	4	1.3	versicolor
5.1	2.5	3	1.1	versicolor
4.9	2.5	4.5	1.7	virginica
6.7	2.5	5.8	1.8	virginica
5.7	2.5	5	2	virginica
6.3	2.5	5	1.9	virginica
5.7	2.6	3.5	1	versicolor
5.5	2.6	4.4	1.2	versicolor
5.8	2.6	4	1.2	versicolor

Numerical value

Categorical value

Face Recognition Dataset

Face Dataset



Training examples of a person

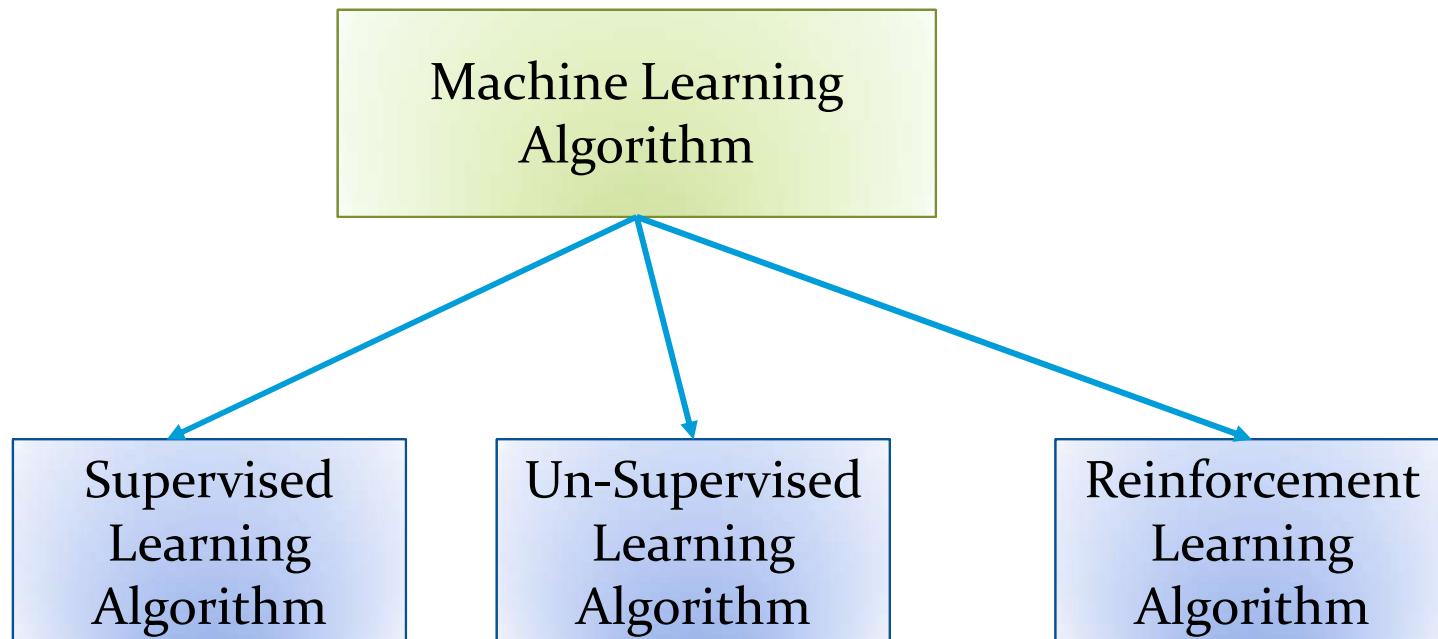


Test images



Learning Algorithm

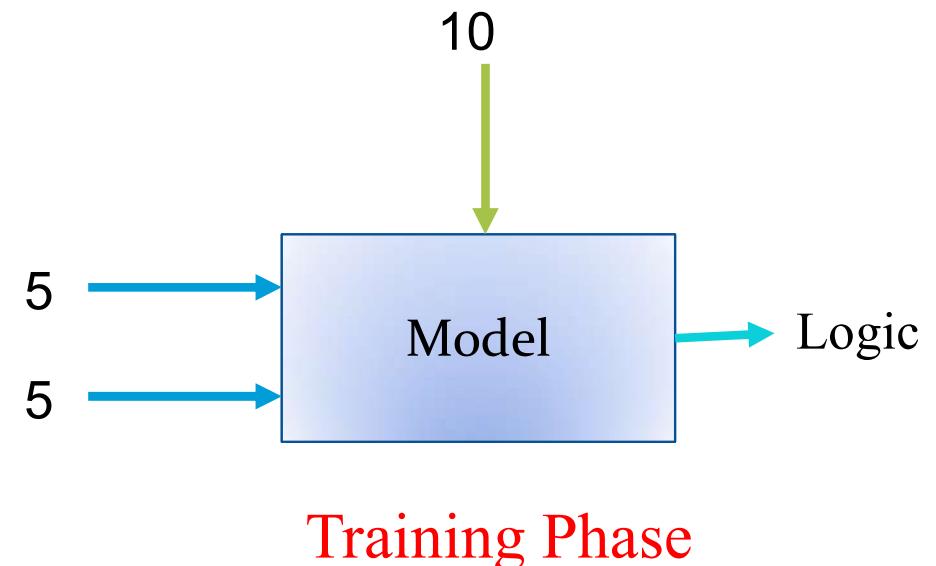
- Machine Learning is a concept which provides ability to the machine to automatically learn and improve from experience without being explicitly programmed.
- The process of learning begins with observations in order to find patterns in data and make better decisions in the future based on the examples that we provide.
- **The primary aim of learning algorithm is to allow the computers learn automatically without human intervention**



Supervised Learning

- Learning in the presence of **instructor**/supervisor/teacher
 - ❖ Ex. Classroom teaching
- Trained machine on a **labelled** dataset.
- Labelled dataset is one which have both **input** and **output parameters**.
- It is **task driven** because outcomes of a supervised learning algorithm are controlled by the task.

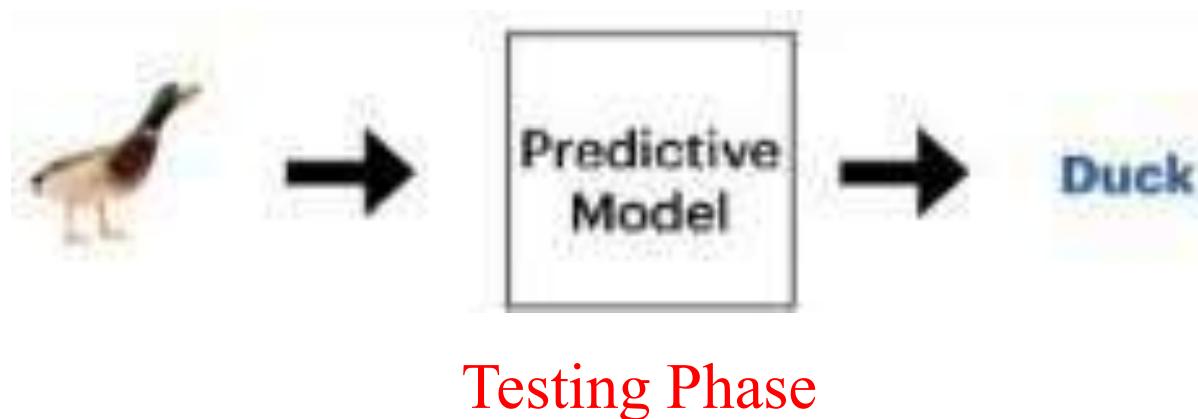
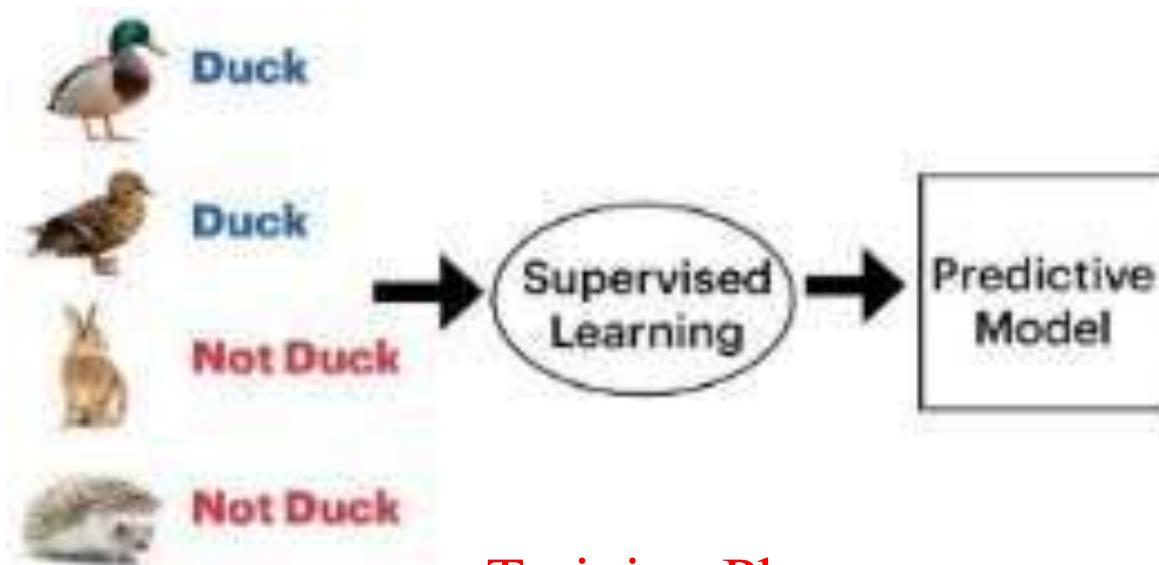
Num-1	Num-2	Sum
5	5	10
8	2	10
10	3	13
15	6	21
20	4	21
30	40	70



Training Phase

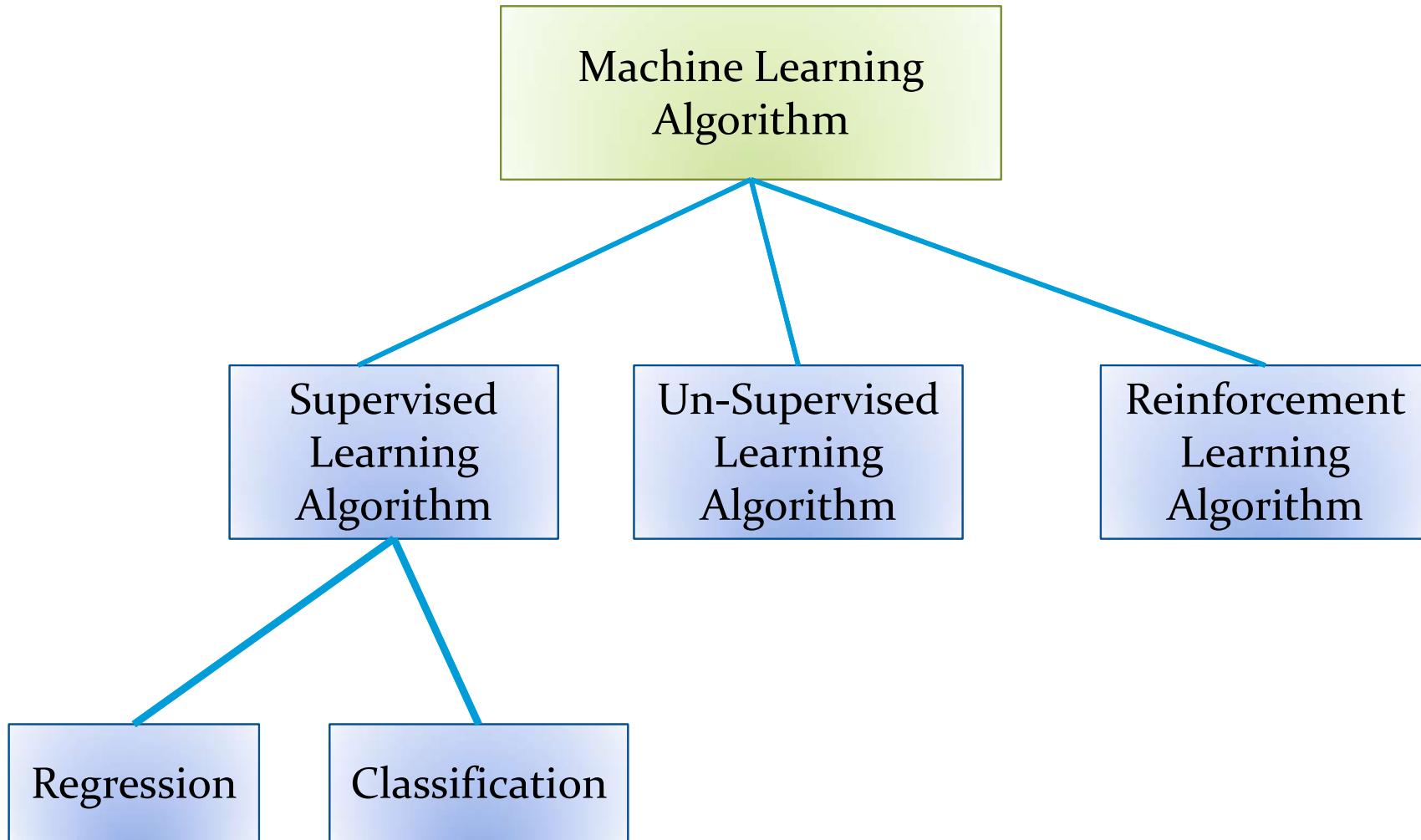


Testing Phase



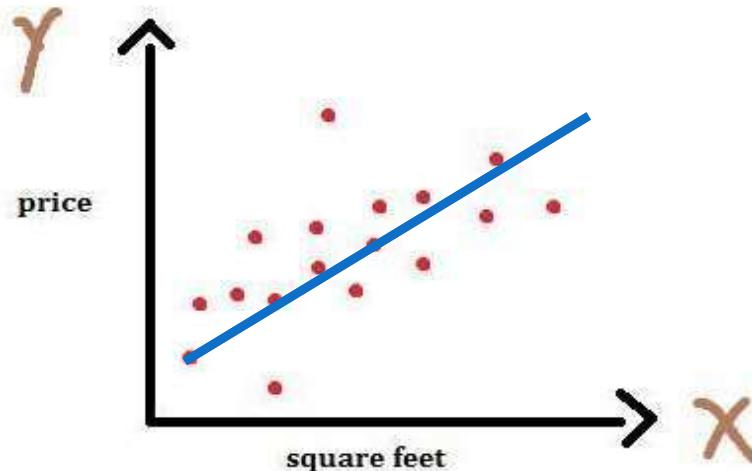
Source: <https://mc.ai/supervised-vs-unsupervised-learning/>

Types of Supervised Learning



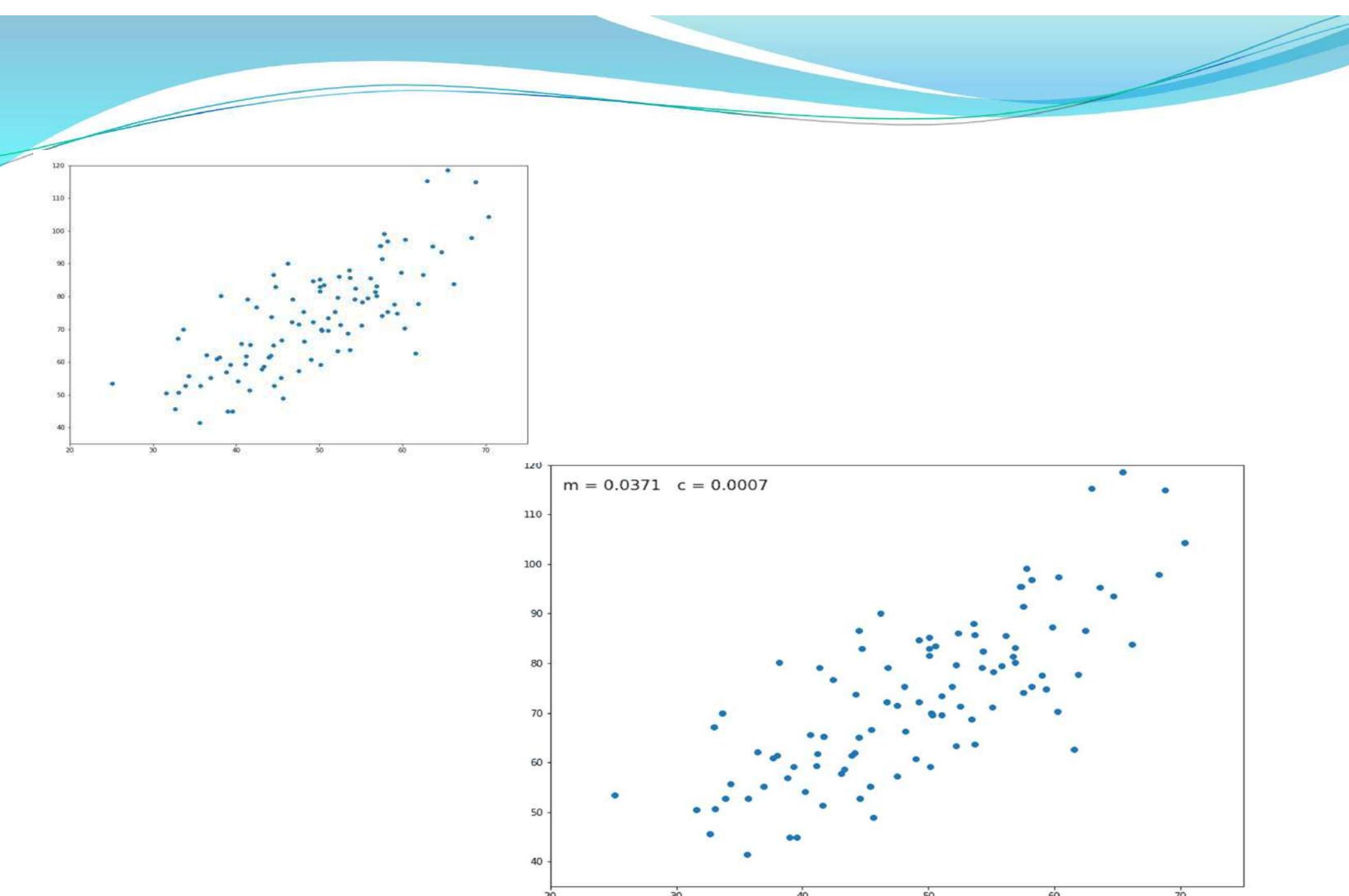
Regression

- If the output of the model is a continuous value.
- It is used to predict a continuous value.



Num-1	Num-2	Sum
5	5	10
8	2	10
10	3	13
15	6	21
20	4	21
30	40	70

- Ex.
 - ❖ House price prediction
 - ❖ Stock market prediction
 - ❖ Predicting age of a person
 - ❖ number of copies a music album will be sold next month



Types of Regression

1) Simple Linear Regression

- In this case, we only have a single independent variable and a single dependent variable.
- In linear regression, while developing the model we assume a linear relationship between the independent and dependent variable.
- In simple linear regression, we try to find a relationship between target variable and input variables by fitting a line, known as the regression line.

$$y = m * x + b$$

$$y(x) = w_0 + w_1 * x$$

where w's are the parameters of the model, x is the input, and y is the target variable.

► Multiple Linear Regression

- It is the extension of the simple linear regression model, to include more than one independent variable.

$$Y(x) = w_0 + w_1x_1 + w_2x_2 ----- + w_nx_n$$

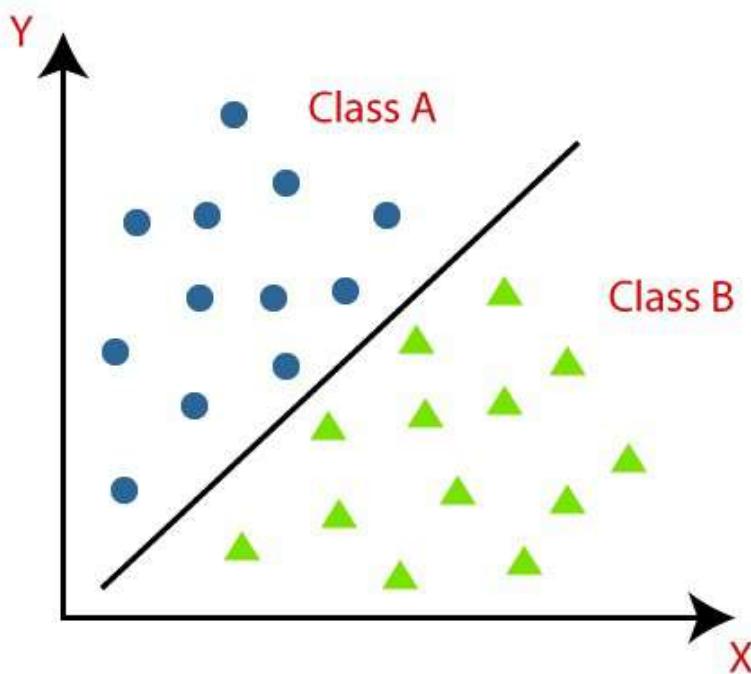
- This is the equation of a hyperplane. A linear regression model in **two** dimensions is a **straight line**; in **three dimensions** it is a **plane**, and in more than three dimensions, a **hyperplane**.

➤ Non Linear Regression

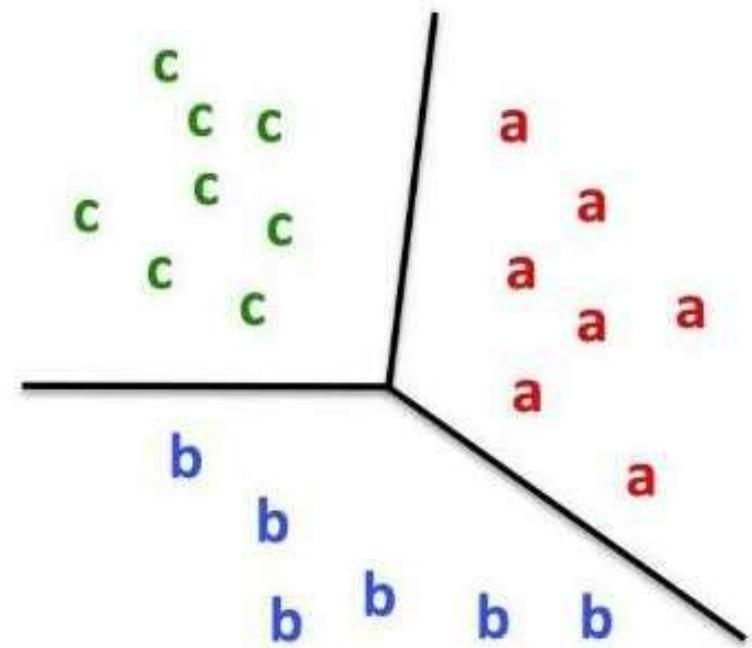
- A regression model, in which the dependent variable is dependent on nonlinear transformation of the parameters\coefficients, is termed a **nonlinear regression model**

Classification

- If the output of the model is *categorical*.
- In classification we are interested to predict the *categorical response* value where the data can be separated into specific “**classes**”.
- Ex.
 - ❖ Spam filtering
 - ❖ Cat dog classification
- Two types
 - ❖ Binary classification: Two class
 - ❖ Multiclass classification: More than two class



Binary Classification



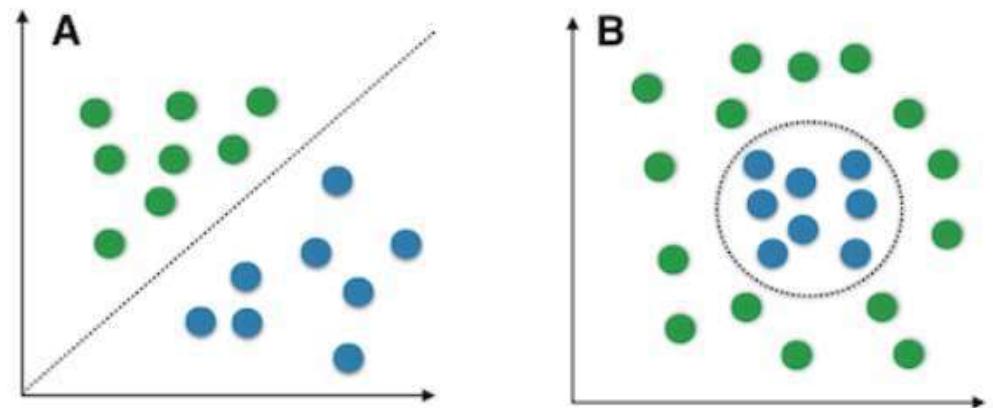
Multi-Class Classification

Source::

https://www.researchgate.net/publication/334612815_HEART_DISEASE_PREDICTION_SYSTEM_HDPS/figures?lo=1

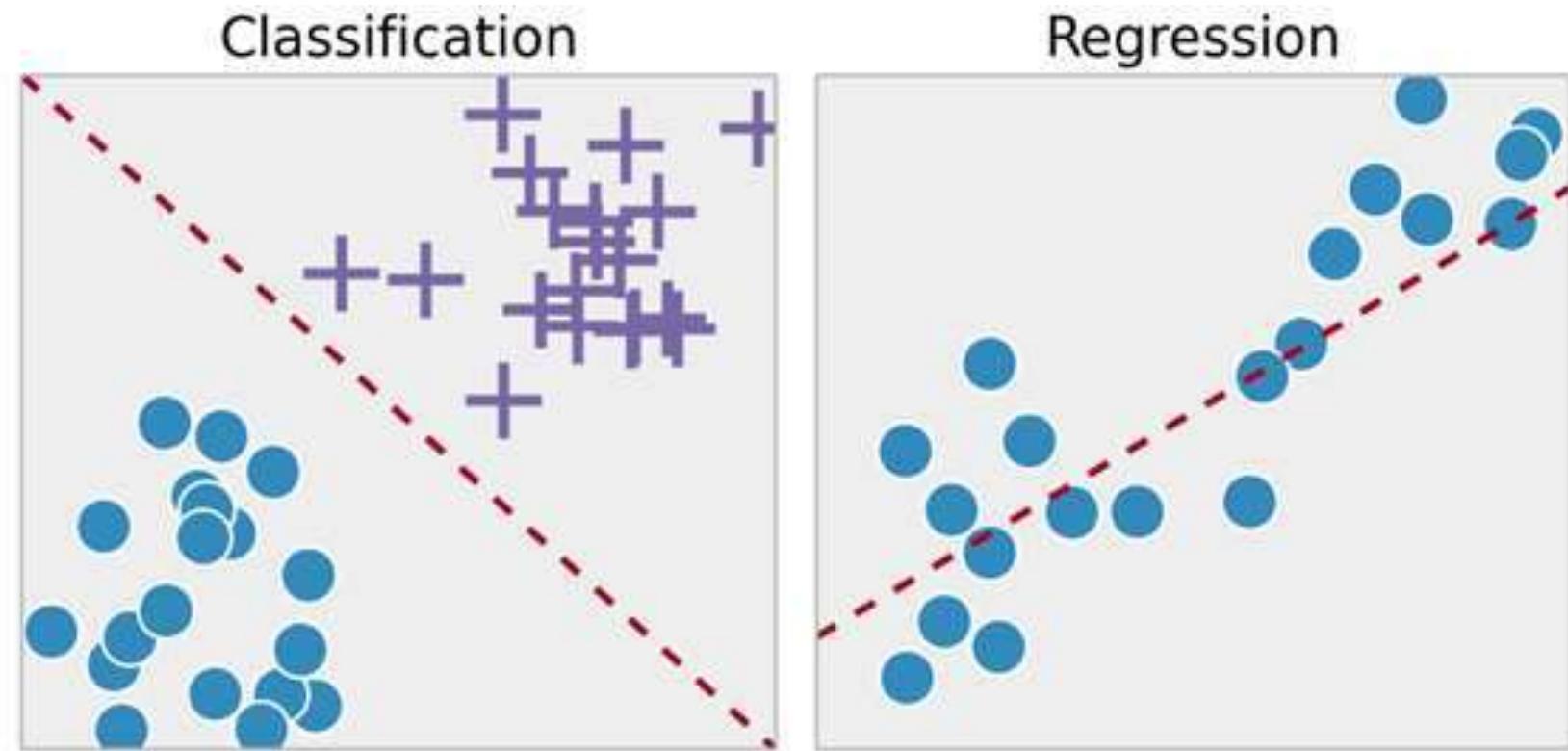
Types of ML Classification Algorithms:

- **Linear Models**
 - Logistic Regression
 - Support Vector Machines
- **Non-linear Models**
 - K-Nearest Neighbours
 - Kernel SVM
 - Naïve Bayes
 - Decision Tree Classification
 - Random Forest Classification



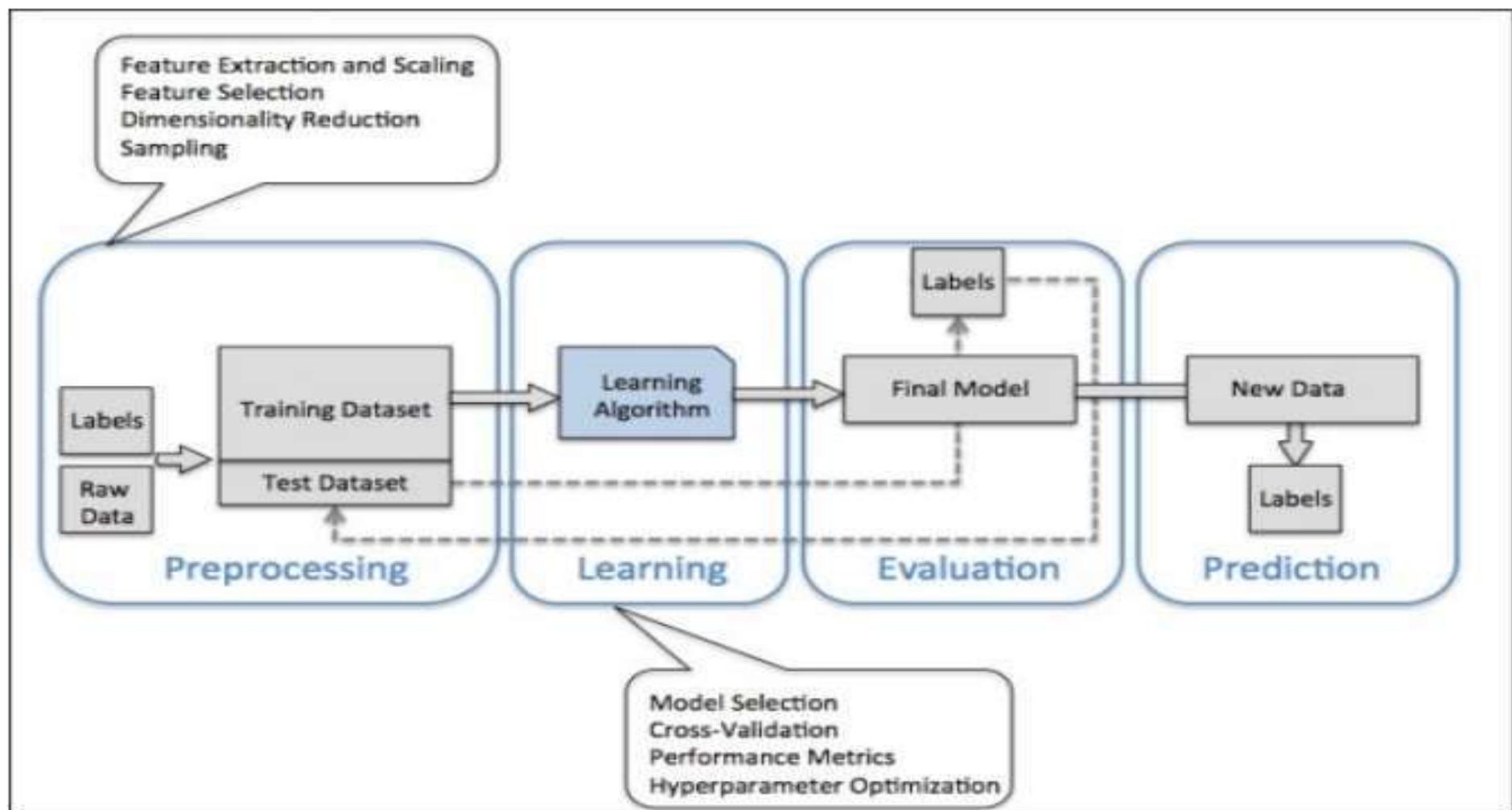
Source: <https://www.kdnuggets.com/2019/12/enabling-deep-learning-revolution.html>

Classification v/s Regression

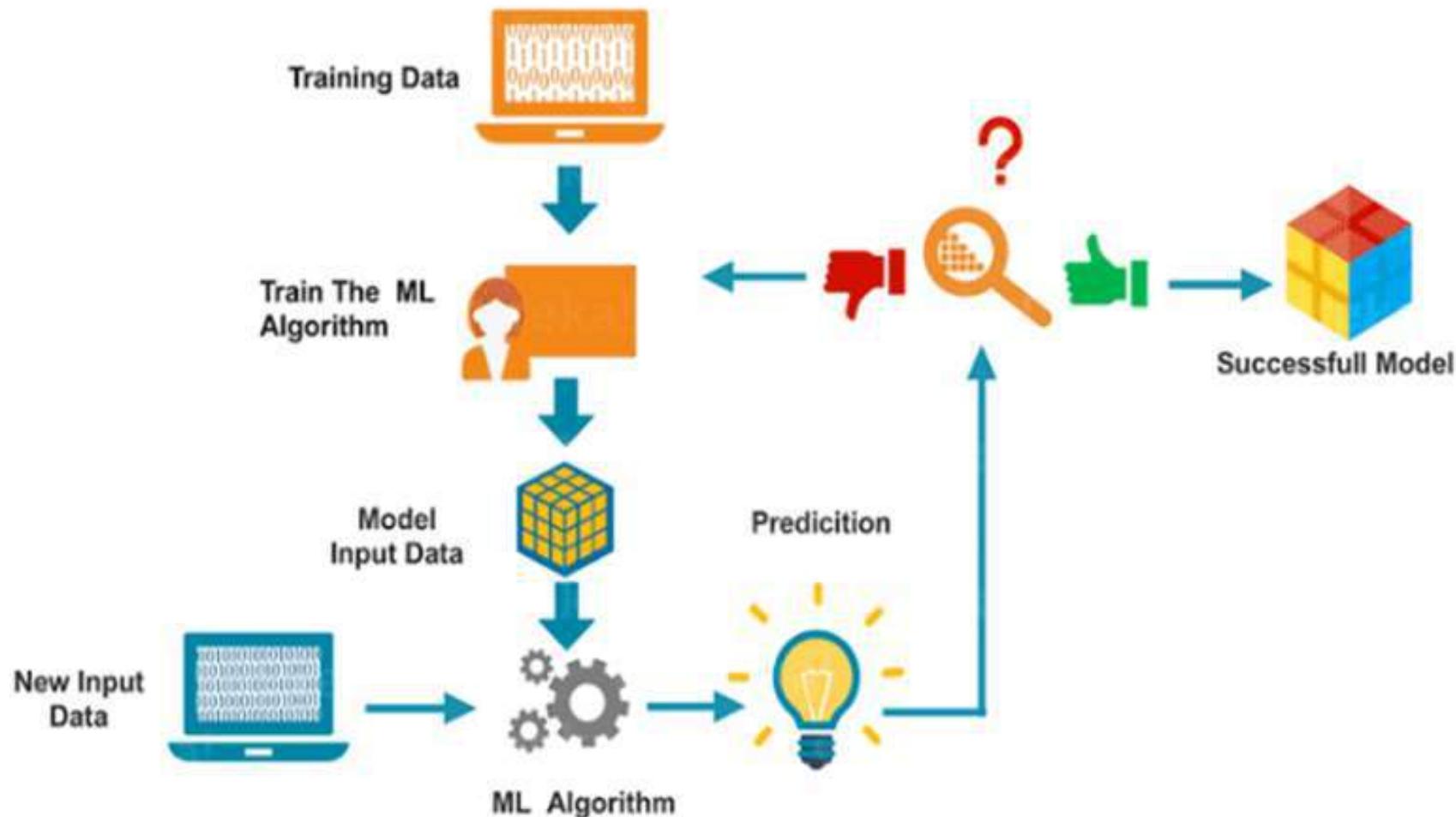


Source: <https://medium.com/deep-math-machine-learning-ai/different-types-of-machine-learning-and-their-types-34760b9128a2>

Supervised Machine Learning Process



How ML Algorithms Works?

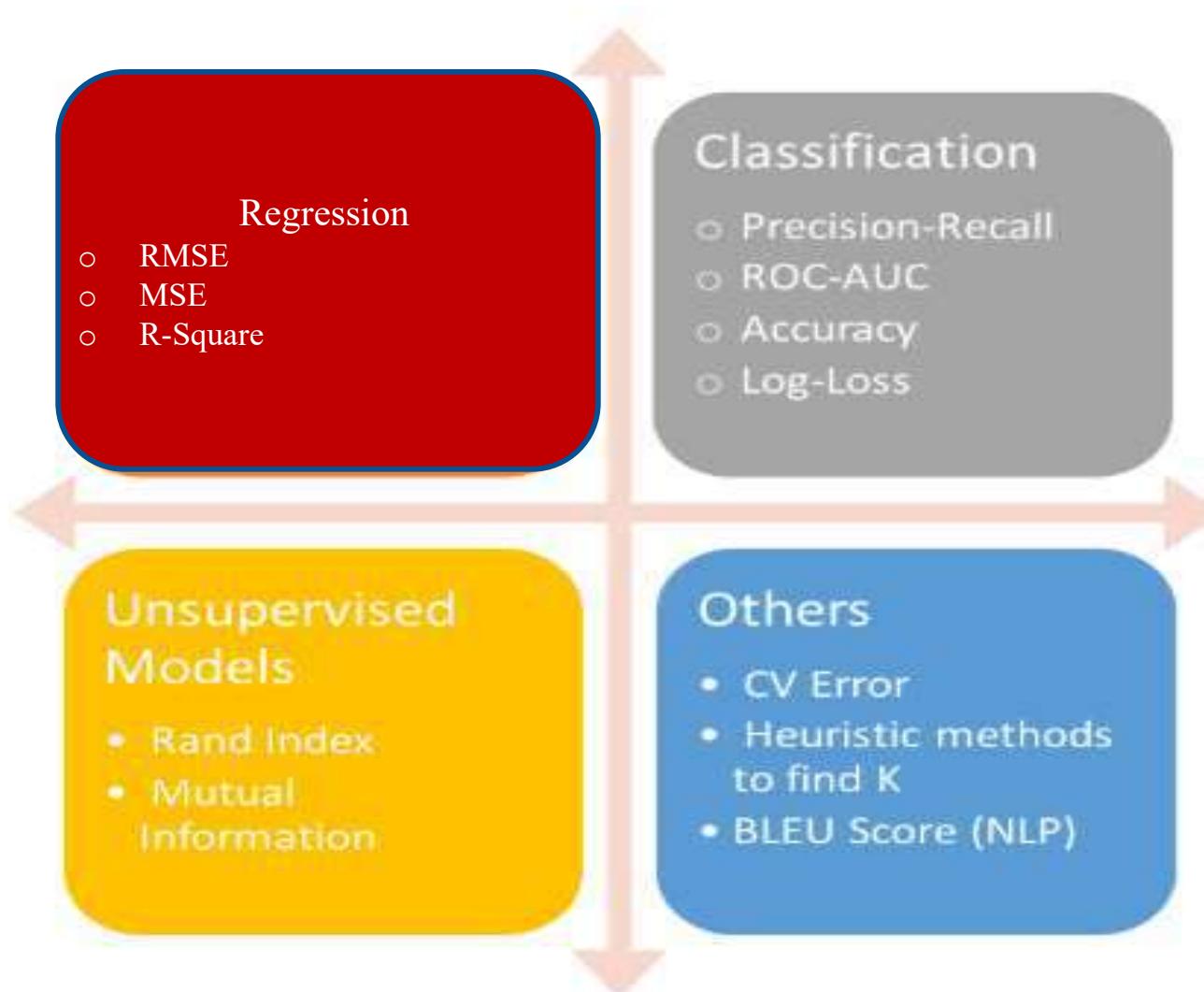


Source: <https://www.spaceotechnologies.com/machine-learning-app-development-complete-guide/>

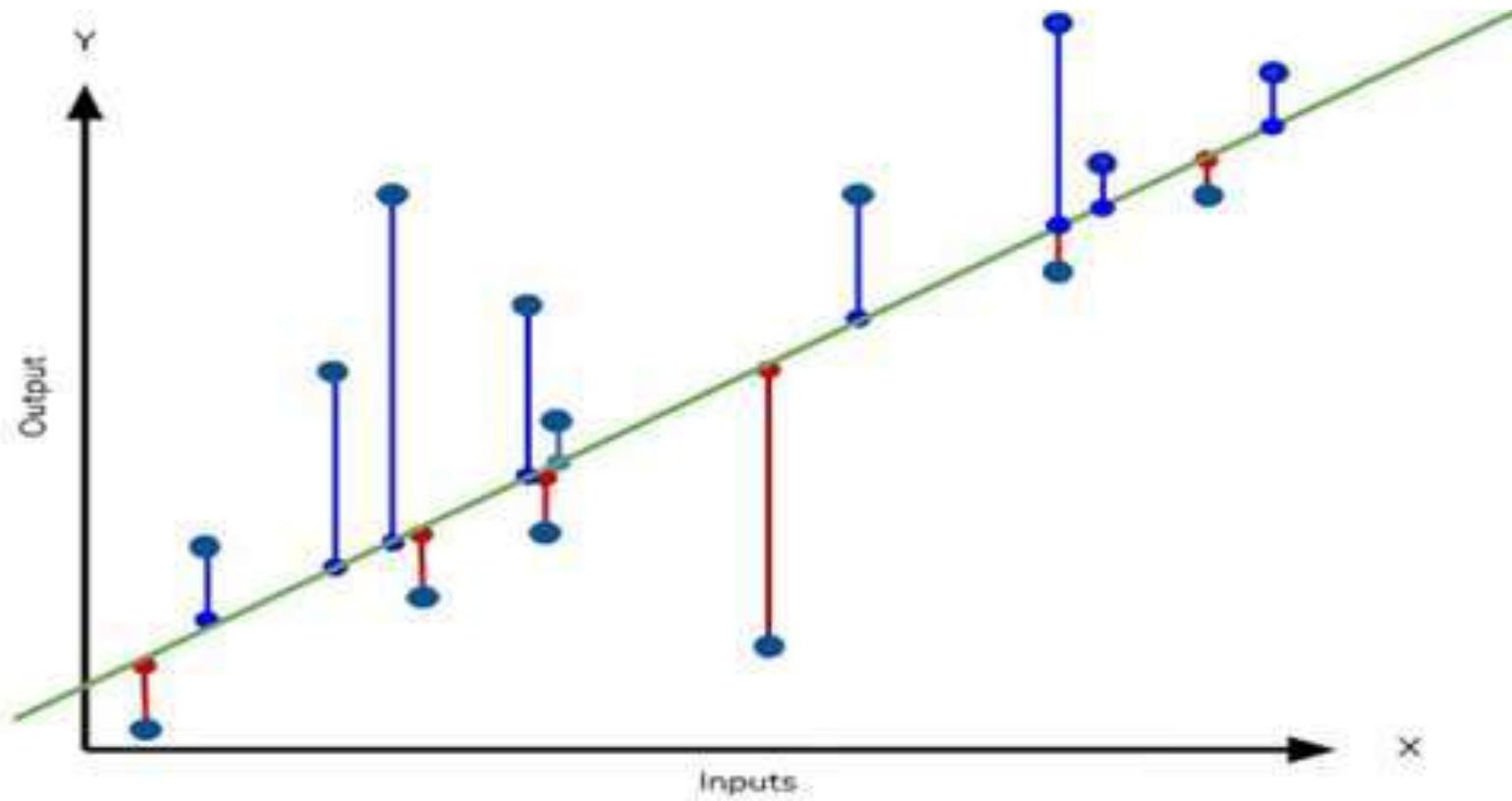


How to Evaluate the Performance of Regression Model

Evaluation Metric For Regression and Classifier



Regression



Regression Evaluation Metric

1) Mean Absolute Error

- This is the mean or average of absolute value of the errors, that is, the predicted - actual.

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

2) Root Mean Squared Error (RMSE)

- This is the square root of the mean of the squared errors. RMSE indicates how close the predicted values are to the actual values; hence a lower RMSE value signifies that the model performance is good.
- One of the key properties of RMSE is that the **unit will be the same** as the target variable.

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

3) Mean Squared Error

- Mean squared error calculates the average of the squares of the errors or deviation between the actual value and the predicted values, as predicted by a regression model.
- The mean squared error or MSE can be used to evaluate a regression model, with lower values meaning a better regression models with less errors.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

3) Coefficient of Determination or R² Score

- The coefficient of determination measures the proportion of variance in the dependent variable which is explained by the independent variable.
- The value of R² is between 0-1.
- A coefficient of determination score of 1 denotes a perfect regression model and indicating that all of the variance is explained by the independent variables.
- It also provides a measure of how well the future samples are likely to be predicted by the model.
- Higher the r² score better the model

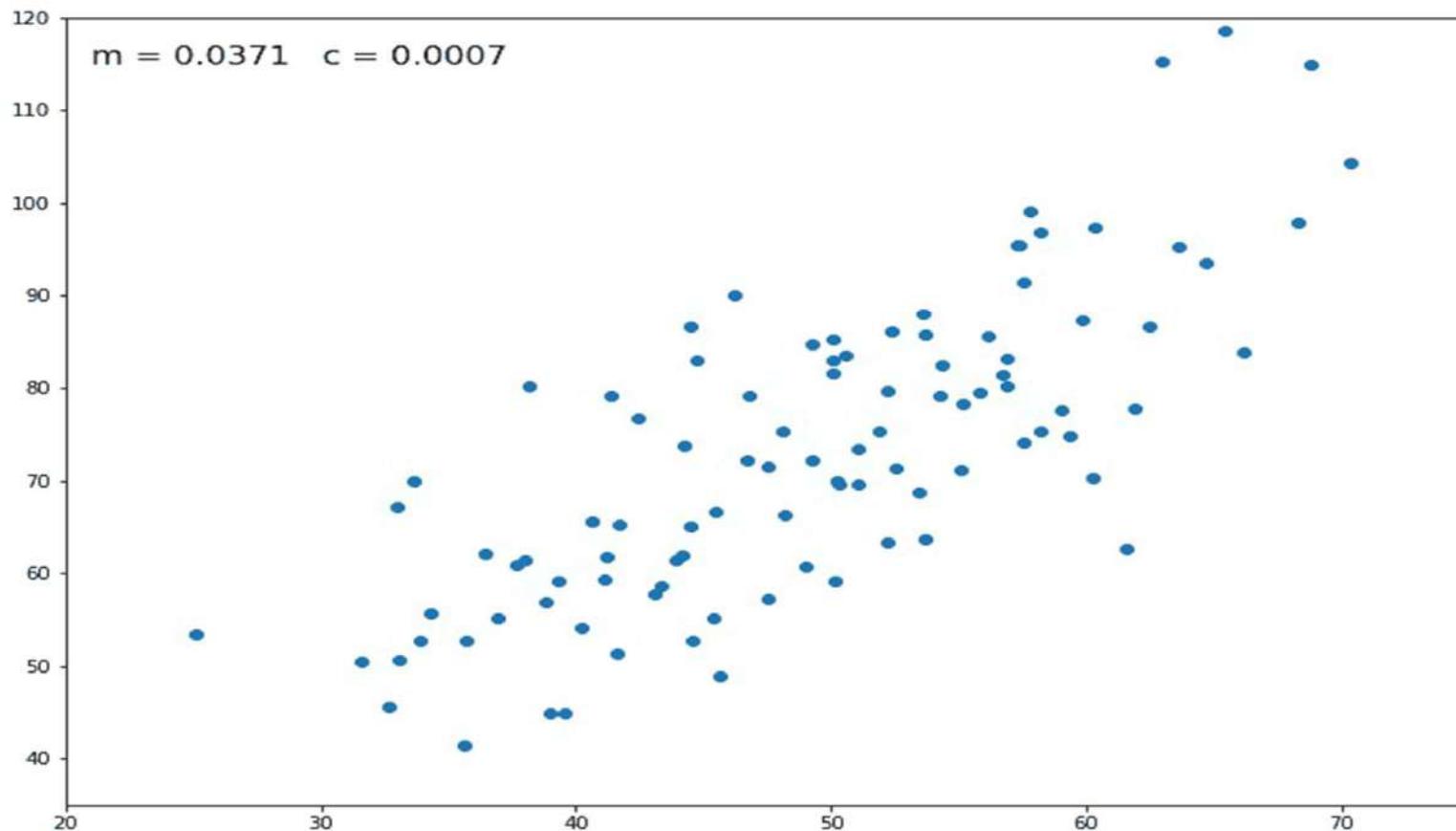
- **Total Sum of Squares (TSS)** : TSS is a measure of total variance in the response/ dependent variable Y and can be thought of as the **amount of variability inherent in the response before the regression is performed**.
- **Residual Sum of Squares (RSS)** : RSS measures the amount of variability that is left unexplained after performing the regression.
- $(TSS - RSS)$ measures the amount of variability in the response that is explained (or removed) by performing the regression

$$SS_t = \sum_{i=1}^m (y_i - \bar{y})^2$$

$$SS_r = \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

$$R^2 \equiv 1 - \frac{SS_r}{SS_t}$$

Linear Regression



$$y(x) = c + m * x$$

What will be the best of m and c

How to Calculate Coefficient

- There are three methods to calculate slope and intercept -one using the following formula

$$m = \frac{((n * \sum(X * Y)) - (\sum(X) * \sum(Y)))}{((n * \sum(X^2)) - (\sum(X))^2)}$$

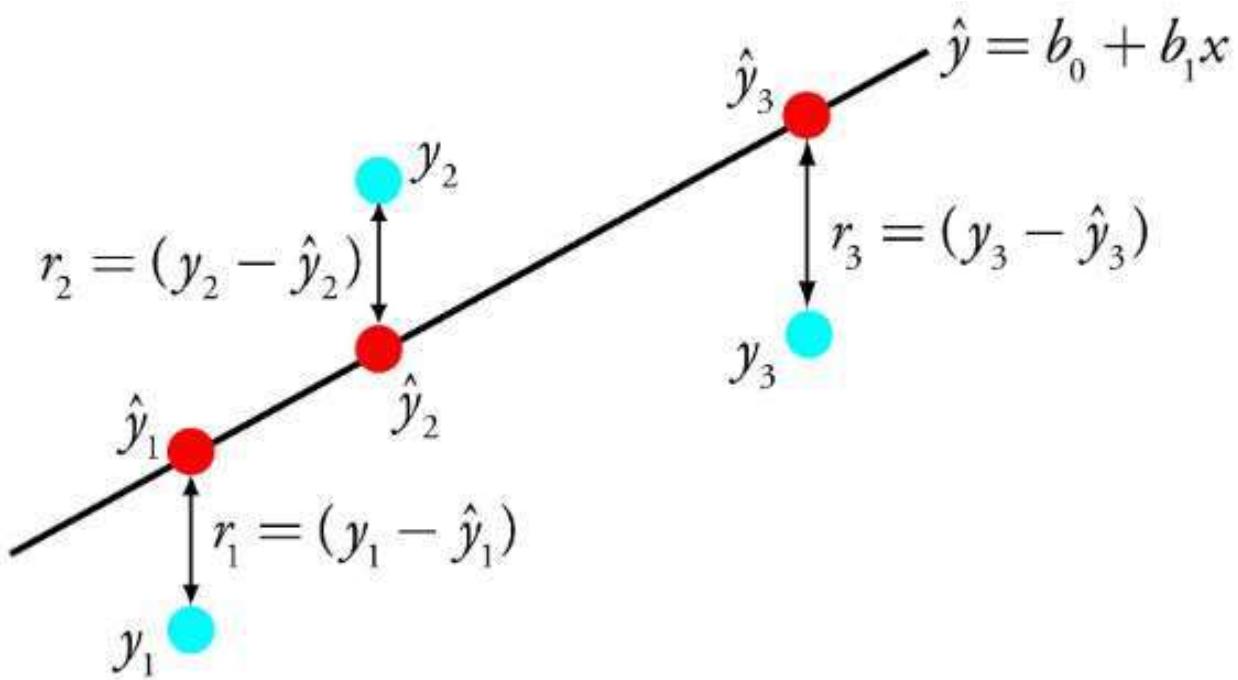
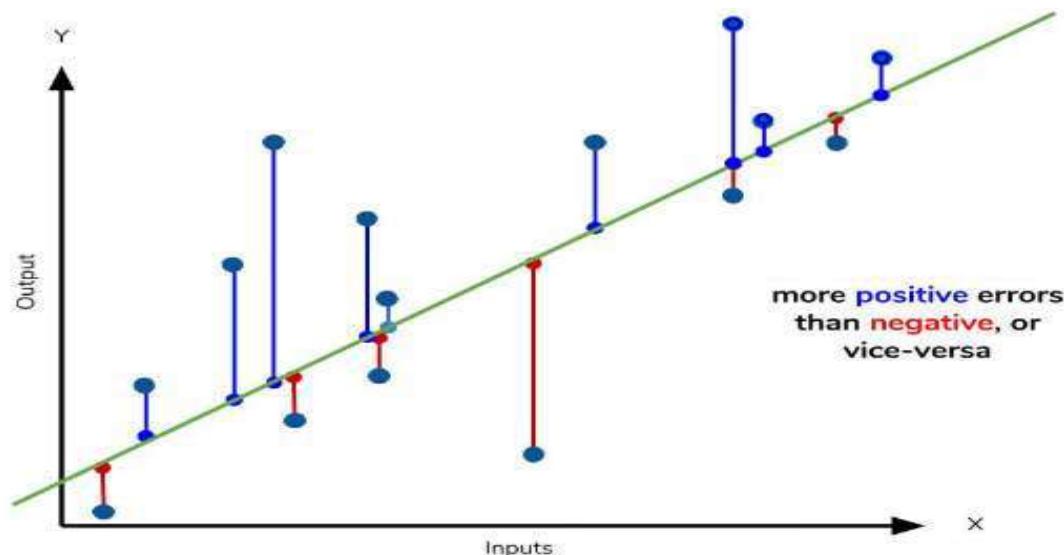
$$c = \frac{((\sum(Y) * \sum(X^2)) - (\sum(X) * \sum(X * Y)))}{((n * \sum(X^2)) - (\sum(X))^2)}$$

- using correlation & standard deviation (shortcut method).

$$m = \text{Correlation}(x,y) * (\text{Std. Dev. of } y / \text{Std. Dev. of } x)$$

$$c = \text{Mean}(Y) - m * \text{Mean}(X)$$

- Gradient decent algorithm



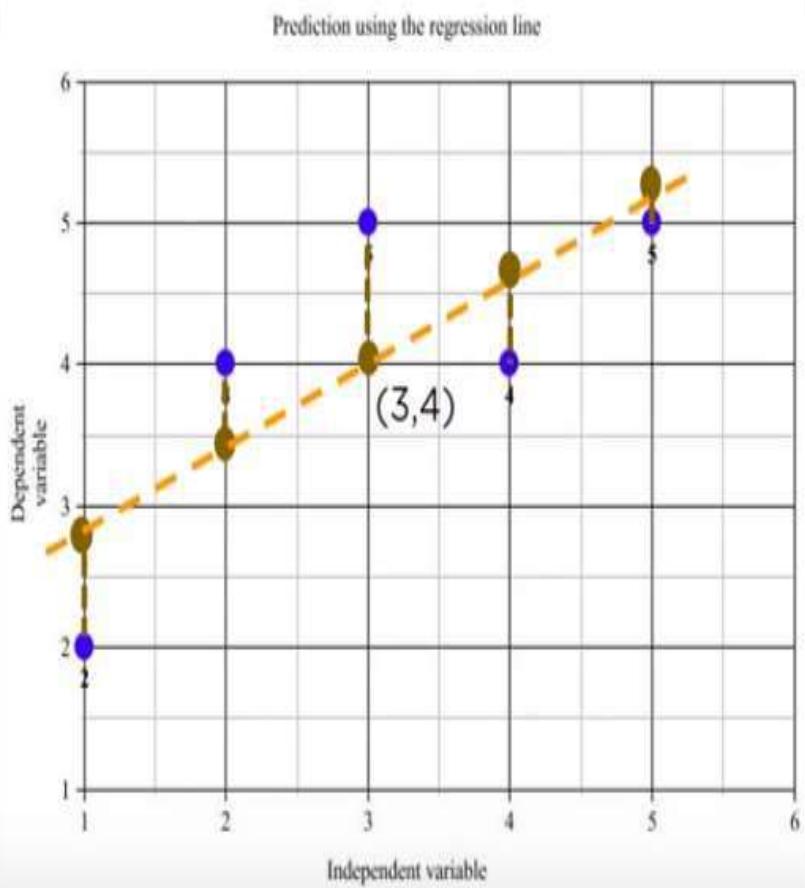
Intuition Behind Linear Regression

X	Y	(X ²)	(Y ²)	(X*Y)
1	2	1	4	2
2	4	4	16	8
3	5	9	25	15
4	4	16	16	16
5	5	25	25	25
$\sum = 15$	$\sum = 20$	$\sum = 55$	$\sum = 86$	$\sum = 66$

$$m = \frac{((n * \sum(X*Y)) - (\sum(X) * \sum(Y)))}{((n * \sum(X^2)) - (\sum(X)^2)} = \frac{((5 * 66) - (15 * 20))}{((5 * 55) - (225))} = 0.6$$

$$c = \frac{((\sum(Y) * \sum(X^2)) - (\sum(X) * \sum(X*Y))}{((n * \sum(X^2)) - (\sum(X)^2)} = 2.2$$

Lets find out the predicted values of Y for corresponding values of X using the linear equation where $m=0.6$ and $c=2.2$



y_{pred}

$$Y=0.6 * 1 + 2.2 = 2.8$$

$$Y=0.6 * 2 + 2.2 = 3.4$$

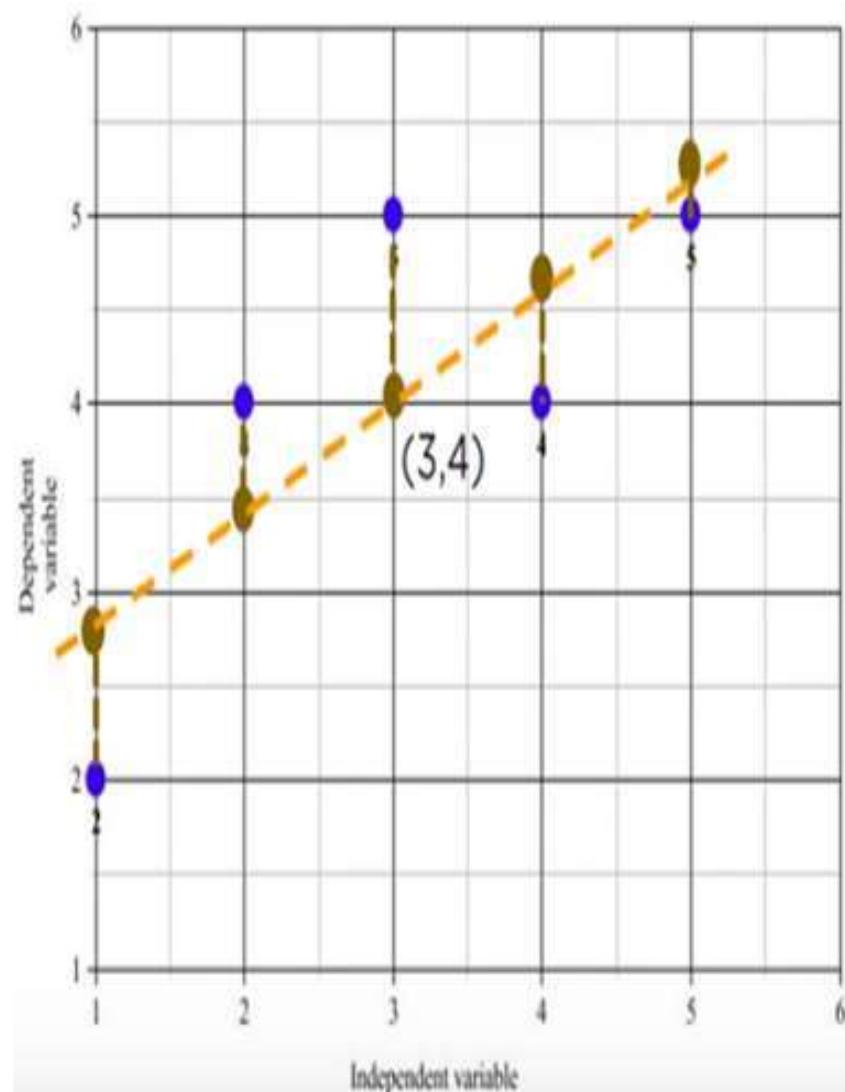
$$Y=0.6 * 3 + 2.2 = 4$$

$$Y=0.6 * 4 + 2.2 = 4.6$$

$$Y=0.6 * 5 + 2.2 = 5.2$$

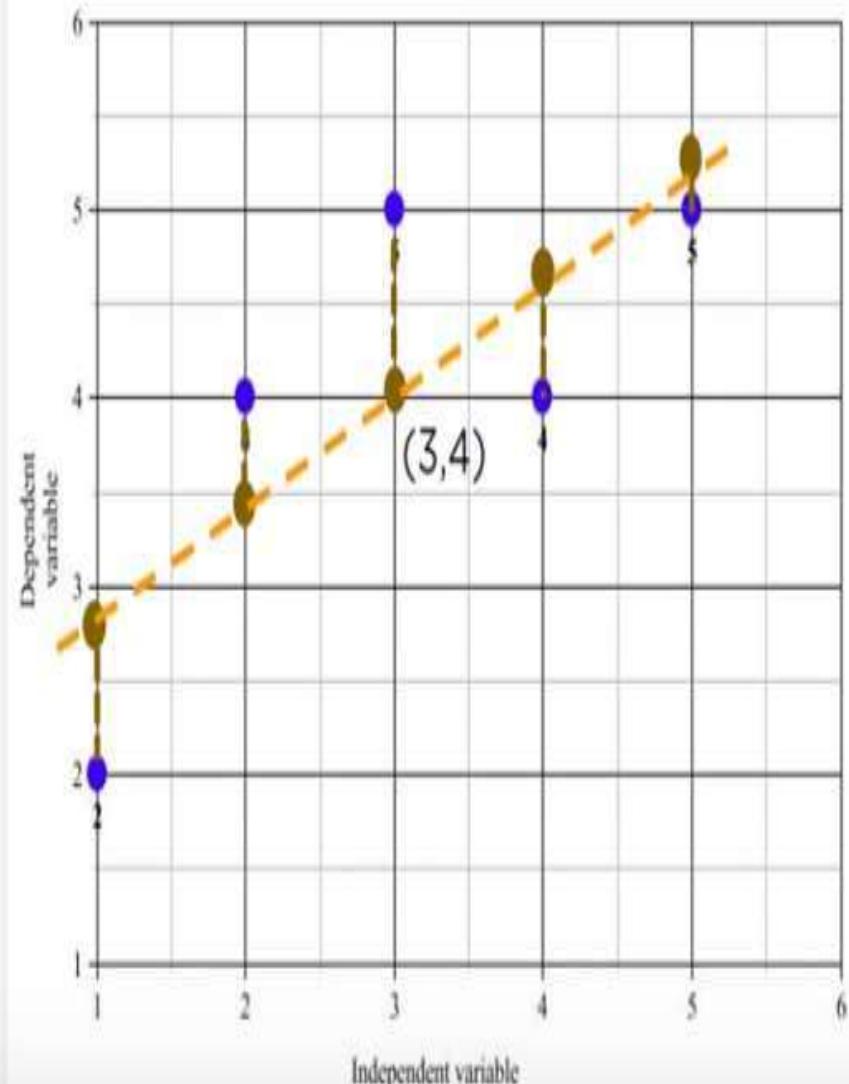
Here the blue points represent the actual Y values and the brown points represent the predicted Y values. The distance between the actual and predicted values are known as *residuals or errors*. The best fit line should have the least sum of squares of these errors also known as *e square*.

Prediction using the regression line



X	Y	Y _{pred}
1	2	2.8
2	4	3.4
3	5	4
4	4	4.6
5	5	5.2

Prediction using the regression line



X	Y	\hat{Y}_{pred}	$(Y - \hat{Y}_{pred})$	$(Y - \hat{Y}_{pred})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04

$$\sum = 2.4$$

The sum of squared errors for this regression line is 2.4. We check this error for each line and conclude the best fit line having the least square value.

Intuition Behind Multiple Linear Regression

SUBJECT	Y	X ₁	X ₂
1	-3.7	3	8
2	3.5	4	5
3	2.5	5	7
4	11.5	6	3
5	5.7	2	1
6	?	3	2

$$a = \bar{Y} - b_1 \bar{X}_1 - b_2 \bar{X}_2$$

$$b_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

$$b_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

$$\Sigma x_1^2 = \Sigma X_1 X_1 - \frac{(\Sigma X_1)(\Sigma X_1)}{N}$$

$$\Sigma x_2^2 = \Sigma X_2 X_2 - \frac{(\Sigma X_2)(\Sigma X_2)}{N}$$

$$\Sigma x_1 y = \Sigma X_1 Y - \frac{(\Sigma X_1)(\Sigma Y)}{N}$$

$$\Sigma x_2 y = \Sigma X_2 Y - \frac{(\Sigma X_2)(\Sigma Y)}{N}$$

$$\Sigma x_1 x_2 = \Sigma X_1 X_2 - \frac{(\Sigma X_1)(\Sigma X_2)}{N}$$

SUBJECT	Y	X ₁	X ₂	X ₁ X ₁	X ₂ X ₂	X ₁ X ₂	X ₁ Y	X ₂ Y
1	-3.7	3	8	9	64	24	-11.1	-29.6
2	3.5	4	5	16	25	20	14	17.5
3	2.5	5	7	25	49	35	12.5	17.5
4	11.5	6	3	36	9	18	69	34.5
5	5.7	2	1	4	1	2	11.4	5.7
Σ	19.5	20	24	90	148	99	95.8	45.6

$$b_1 = \frac{(\sum x_2^2)(\sum x_1y) - (\sum x_1x_2)(\sum x_2y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1x_2)^2} = \frac{32.8 * 17.8 - 3 * (-48)}{10 * 32.8 - 3 * 3} = 2.28$$

$$b_2 = \frac{(\sum x_1^2)(\sum x_2y) - (\sum x_1x_2)(\sum x_1y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1x_2)^2} = \frac{10 * (-48) - 3 * 17.8}{10 * 32.8 - 3 * 3} = -1.67$$

$$a = \bar{Y} - b_1 \bar{X}_1 - b_2 \bar{X}_2 = \frac{19.5}{5} - \frac{2.28 * 20}{5} - \frac{-1.67 * 24}{5} = 2.796$$

Final Regression equation or Model is:

$$Y = 2.796 + 2.28x_1 - 1.67x_2$$

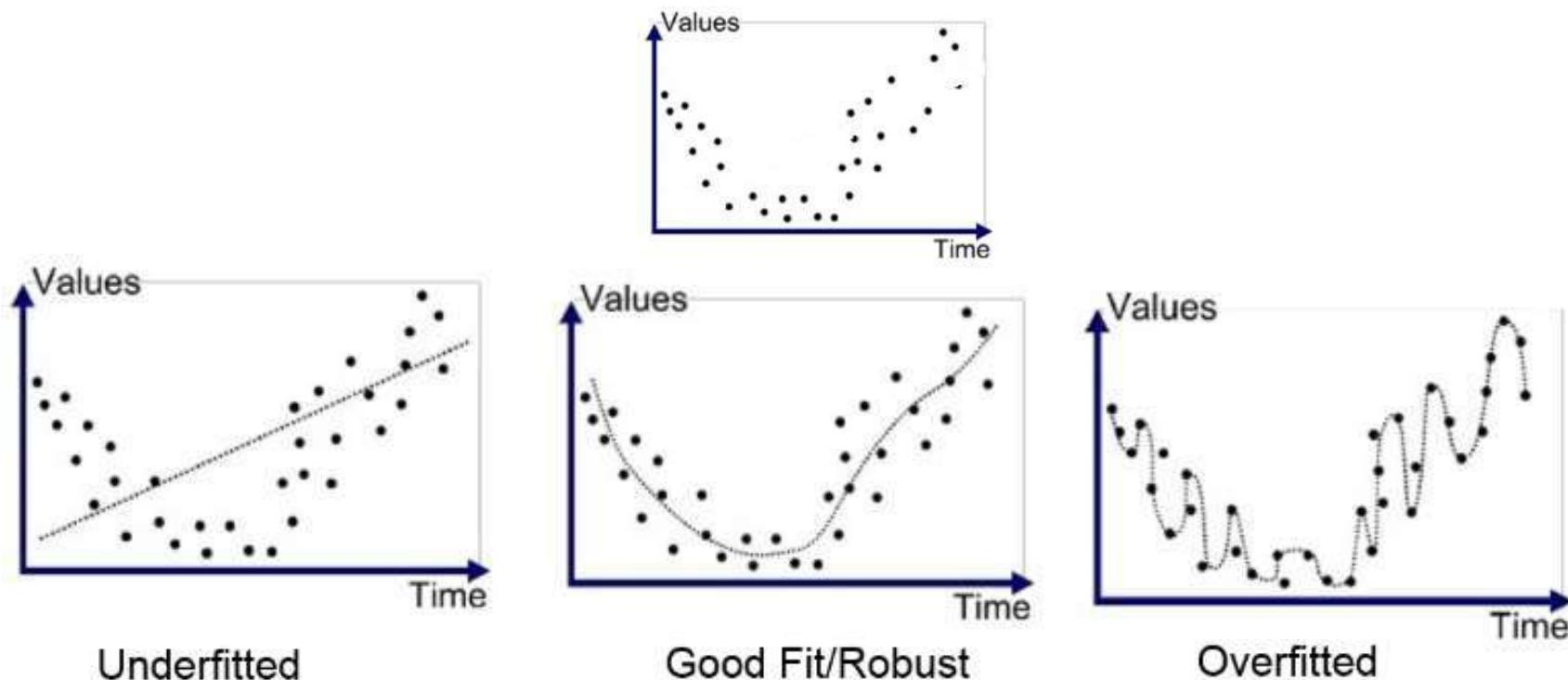
Now given $x_1 = 3$ and $x_2 = 2$ $Y = ?$

$$\begin{aligned} Y &= 2.796 + 2.28 * 3 - 1.67 * 2 \\ &= \mathbf{6.296} \end{aligned}$$

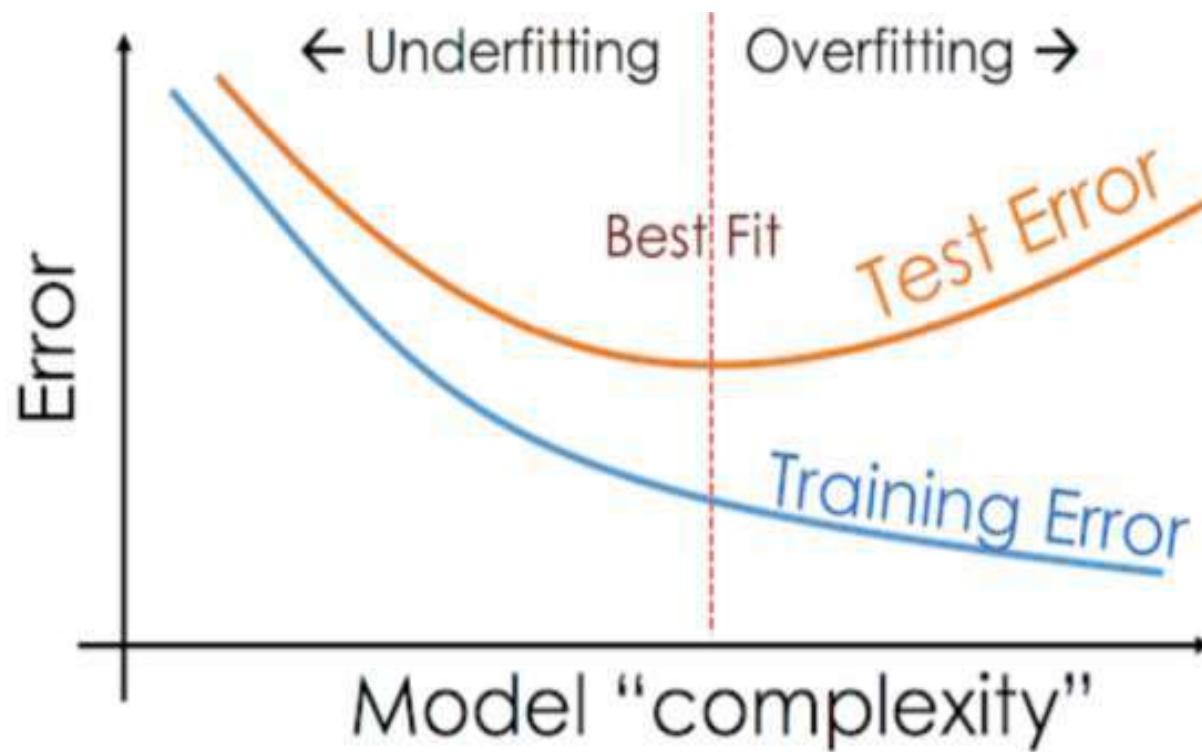
Underfitting and Overfitting

Underfitting: **Low** training and test accuracy

Overfitting: **High** training accuracy and **low test** accuracy

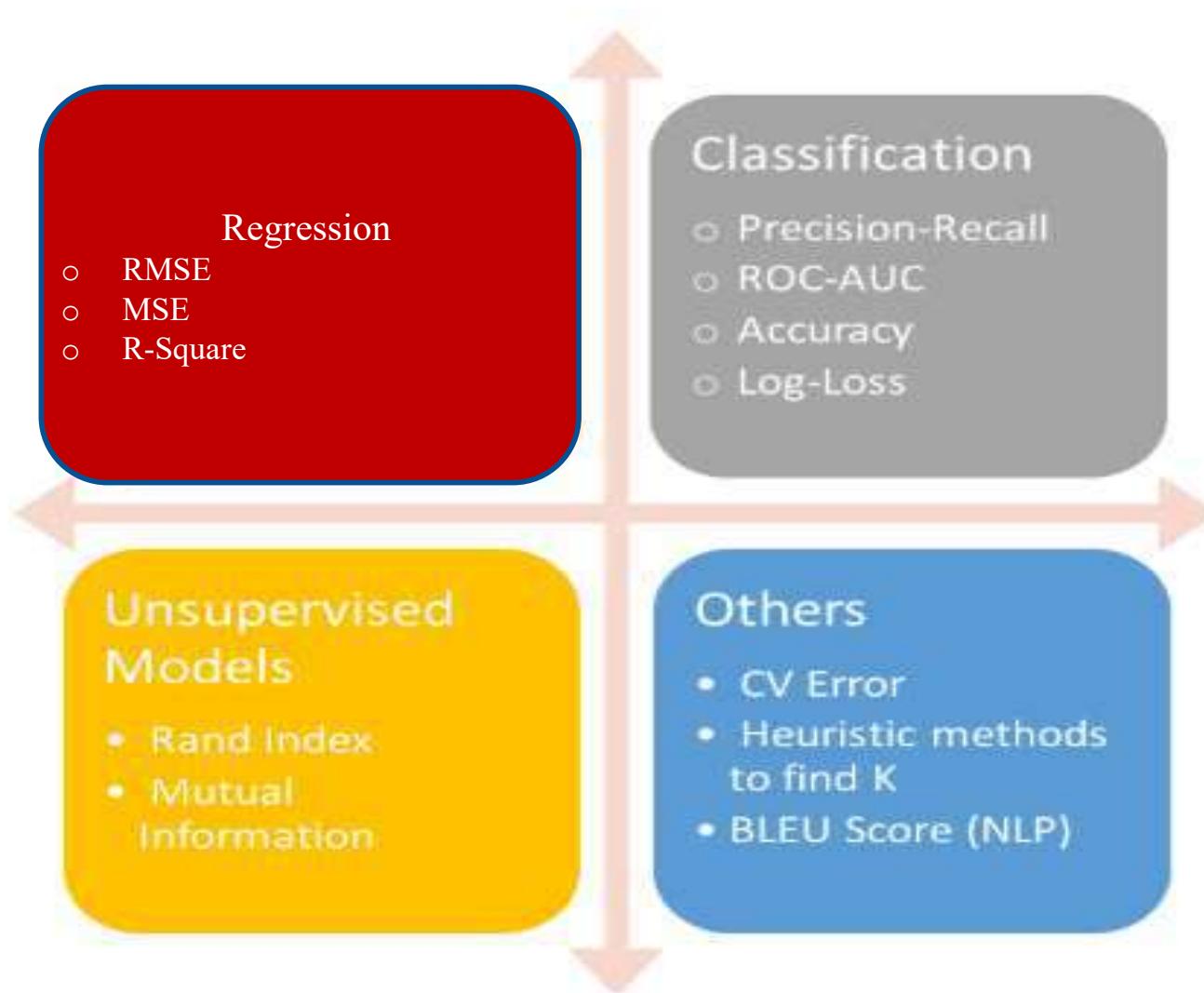


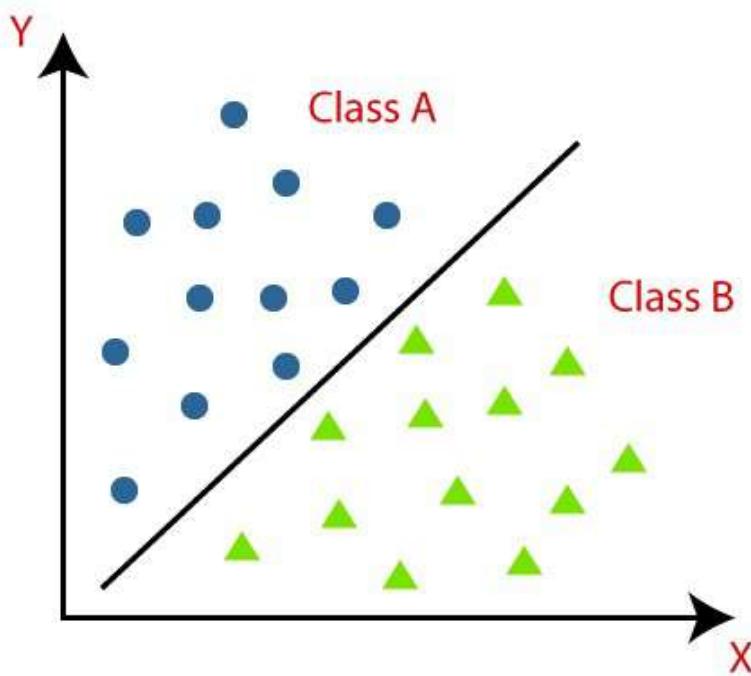
Source: <https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76/>



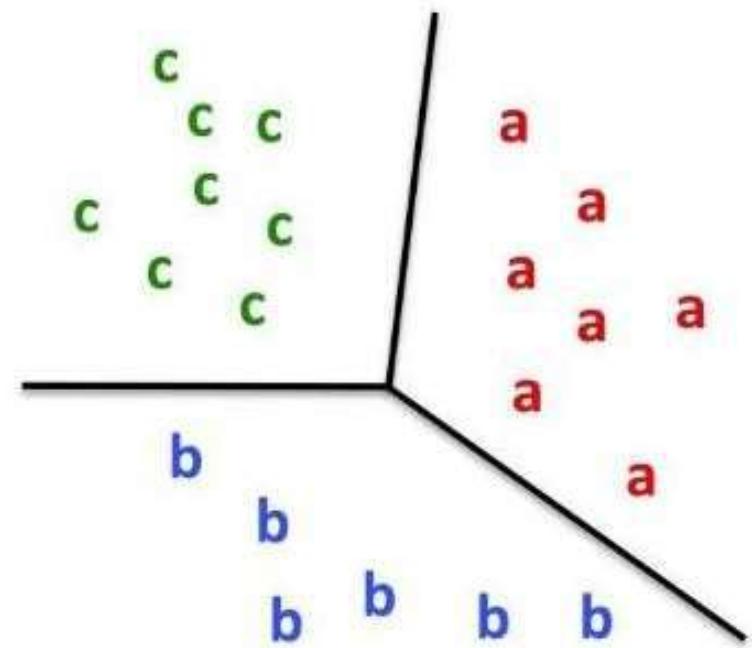
Source: <https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/>

Evaluation Metric for Regression and Classification





Binary Classification



Multi-Class Classification

Source::

https://www.researchgate.net/publication/334612815_HEART_DISEASE_PREDICTION_SYSTEM_HDPS/figures?lo=1

Classification Evaluation Metric

- True Positives (TP): Actual TRUE, which was predicted as TRUE
- True Negatives (TN): Actual FALSE, which was predicted as FALSE
- False Positives (FP): Actual FALSE, which was predicted as TRUE (Type I error)
- False Negatives (FN): Actual TRUE, which was predicted as FALSE (Type II error)

1) Confusion Matrix

- A confusion matrix is created by comparing the **predicted class** label of a data point with its **actual class label**.
- A confusion matrix can be created for a binary classification as well as a multi-class classification model.

Actual Output (y)	Predicted Output (y')
1	1
1	1
0	1
0	0
1	0
1	1

1	3	0
0	1	1

- It is a $n \times n$ matrix, where n is a number of classes.

		Predicted class	
		P	N
Actual Class		P	True Positives (TP)
		N	False Negatives (FN)
		P	False Positives (FP)
		N	True Negatives (TN)

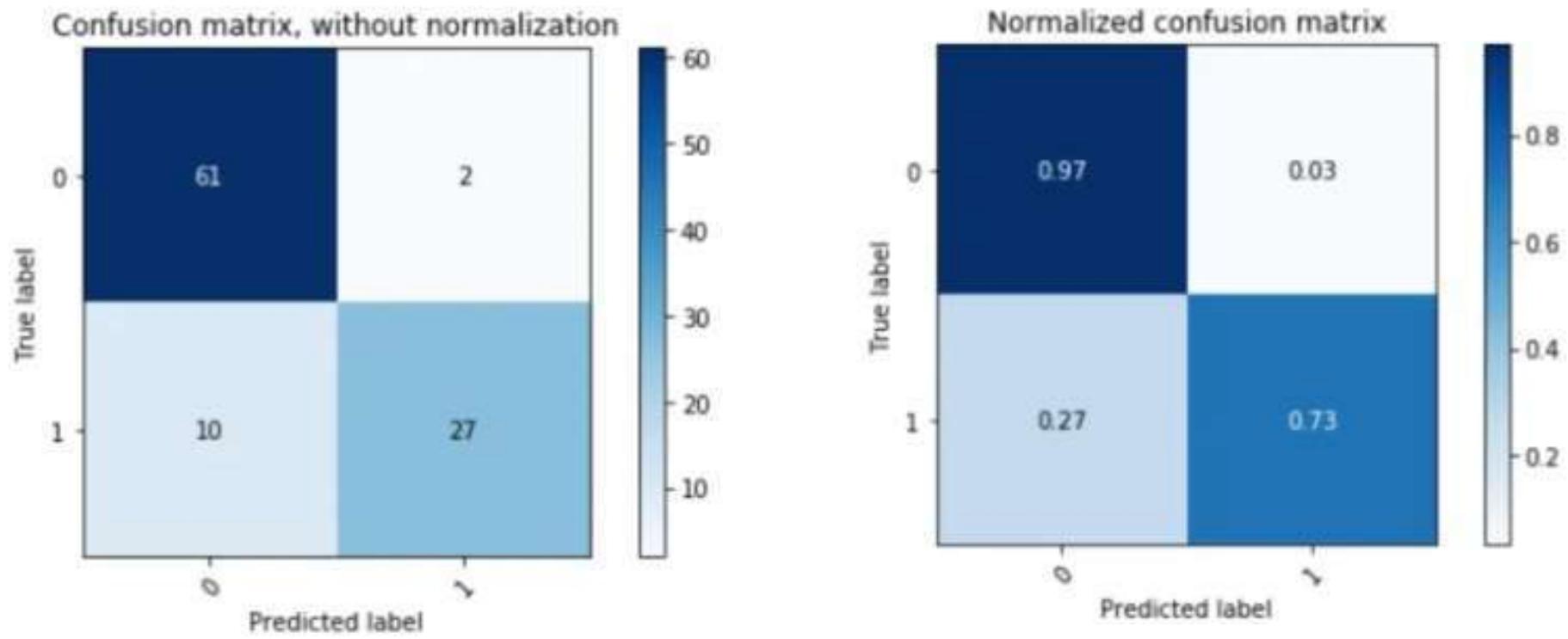
		Actual class	
		1	0
Predicted class		1	True positive
		0	False positive
		1	False negative
		0	True negative

		Predicted No:	Predicted Yes:
		Actual No	Actual Yes
Actual No	60	50	10
Actual Yes	40	5	35

		Prediction	
		Cat	Dog
Actual		Cat	15
		Dog	40

Total		Predicted		
		<i>Iris-setosa</i>	<i>Iris-versicolor</i>	<i>Iris-virginica</i>
Actual	<i>Iris-setosa</i>	12	1	1
	<i>Iris-versicolor</i>	0	16	0
	<i>Iris-virginica</i>	0	1	16

	Urban	Agriculture	Range	Forest	Water	
Urban	310	9	18	23	18	378
Agriculture	61	1051	92	147	12	1363
Range	12	32	561	86	17	708
Forest	23	87	218	1202	8	1538
Water	11	7	12	27	270	327
	417	1186	901	1485	325	3394



Why Confusion Matrix?

		Prediction	
		Cat	Dog
Actual	Cat	15	35
	Dog	40	10

Accuracy

- It is the proportion of the total number of predictions that are correct.

		Predicted Class	
		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

$$\text{Accuracy} = (45+30)/(45+20+5+30) = 75\%$$

The 75% of examples are correctly classified by the classifier.

		Normal	Fraud
Normal	Normal	940	10
	Fraud	40	10

Accuracy/Precision/ Recall

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

Source: <https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html>

Precision

- It is a ratio of total number of correctly classified positive examples and the total number of predicted positive examples. It shows correctness achieved in positive prediction.
- Precision becomes important in cases where we are more concerned about finding the **maximum number of positive class** even if the total accuracy reduces.

		Predicted Class	
		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP})$$

$$\text{Precision} = 45/(45+5) = 90\%$$

The 90% of examples are classified as spam are actually spam.

Recall / Sensitivity / True Positive Rate

- It is measure of positive examples labelled as positive by classifier. It should be higher.

		Predicted Class	
		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) = \text{TP}/\text{P}$$

$$\text{Sensitivity} = 45/(45+20) = 69.23\% .$$

The 69.23% spam emails are correctly Classify

Specificity / *True Negative Rate*

- It is measure of negative examples labeled as negative by classifier.
There should be high specificity.

$$\text{Specificity} = \text{TN}/(\text{TN}+\text{FP}) = \text{TN}/\text{N}$$

		Predicted Class	
		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

$$\text{specificity} = 30/(30+5) = 85.71\%$$

The 85.71% non-spam emails
are accurately classified

F1 Score

- It is a weighted average of the recall and precision.
- F1 score might be good choice when you seek to balance between Precision and Recall.

$$\text{F1 Score} = 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

Metric	Description	Formula
Accuracy	what % of predictions were correct?	$(TP+TN)/(TP+TN+FP+FN)$
Misclassification Rate	what % of prediction is wrong?	$(FP+FN)/(TP+TN+FP+FN)$
True Positive Rate OR Sensitivity OR Recall (completeness)	what % of positive cases did model catch?	$TP/(FN+TP)$
False Positive Rate	what % of 'No' were predicted as 'Yes'?	$FP/(FP+TN)$
Specificity	what % of 'No' were predicted as 'No'?	$TN/(TN+FP)$
Precision (exactness)	what % of positive predictions were correct?	$TP/(TP+FP)$
F1 score	Weighted average of precision and recall	$2*((precision * recall) / (precision + recall))$

How to Calculate Precision/ Recall

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

Source: <https://dev.to/overrideveloper/understanding-the-confusion-matrix-264i>

True Negatives for the Greyhound class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound			
	Mastiff	P_{MM}	P_{SM}	
	Samoyed	P_{GS}	P_{SS}	

True Negatives for the Mastiff class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P_{GG}		P_{SG}
	Mastiff			
	Samoyed	P_{GS}		P_{SS}

True Negatives for the Samoyed class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P_{GG}	P_{MG}	
	Mastiff	P_{GM}	P_{MM}	
	Samoyed			

False Positives for the Greyhound class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

False Positives for the Mastiff class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

False Positives for the Samoyed class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

Source: <https://dev.to/overrideveloper/understanding-the-confusion-matrix-264i>

False Negatives for the Greyhound class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

False Negatives for the Mastiff class:

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

False Negatives for the Samoyed class

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	P _{GG}	P _{MG}	P _{SG}
	Mastiff	P _{GM}	P _{MM}	P _{SM}
	Samoyed	P _{GS}	P _{MS}	P _{SS}

Source: <https://dev.to/overrideveloper/understanding-the-confusion-matrix-264i>

Actual

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	250	25	18
	Mastiff	21	320	24
	Samoyed	22	12	180



How to Calculate Accuracy, Precision and Recall for More than Two Classes

		Predicted		
		Greyhound	Mastiff	Samoyed
Actual	Greyhound	250	25	18
	Mastiff	21	320	24
	Samoyed	22	12	180

Predicted Class

		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

- **TN (Greyhound) = PMM + PSM + PMS + PSS = $320 + 24 + 12 + 180 = 536$**
- **TN (Mastiff) = PGG + PSG + PGS + PSS = $250 + 18 + 22 + 180 = 470$**
- **TN (Samoyed) = PGG + PMG + PGM + PMM = $250 + 25 + 21 + 320 = 616$**

False Positives

- **FP (Greyhound) = PGM + PGS = 21 + 22 = 43**
- **FP (Mastiff) = PMG + PMS = 25 + 12 = 37**
- **FP (Samoyed) = PSG + PSM = 18 + 24 = 42**

Actual	Predicted		
	Greyhound	Mastiff	Samoyed
Greyhound	250	25	18
Mastiff	21	320	24
Samoyed	22	12	180

False Negatives

- **FN (Greyhound) = PMG + PSG = 25 + 18 = 43**
- **FN (Mastiff) = PGM + PSM = 21 + 24 = 45**
- **FN (Samoyed) = PGS + PMS = 22 + 12 = 34**

True Positive Rate / Recall / Sensitivity

- **TPR (Greyhound) = TP / (TP + FN) = 250 / (250 + 43) = 250 / 293 = 0.8532423208 = 0.85**
- **TPR (Mastiff) = TP / (TP + FN) = 320 / (320 + 45) = 320 / 365 = 0.8767123288 = 0.88**
- **TPR (Samoyed) = TP / (TP + FN) = 180 / (180 + 34) = 180 / 214 = 0.8411214953 = 0.84**

Precision

- **Precision (Greyhound)** = $\text{TP} / (\text{TP} + \text{FP}) = 250 / (250 + 43) = 250 / 293 = 0.8532423208 = 0.85$

Therefore, the classifier has a precision of **0.85**, which is **85%**, in classifying images of Greyhounds.

- **Precision (Mastiff)** = $\text{TP} / (\text{TP} + \text{FP}) = 320 / (320 + 37) = 320 / 357 = 0.8963585434 = 0.90$

Therefore, the classifier has a precision of **0.90**, which is **90%**, in classifying images of Mastiffs.

- **Precision (Samoyed)** = $\text{TP} / (\text{TP} + \text{FP}) = 180 / (180 + 42) = 180 / 222 = 0.8108108108 = 0.81$

Therefore, the classifier has a precision of **0.81**, which is **81%**, in classifying images of Samoyeds.

True Negative Rate / Specificity

- **TNR (Greyhound)** = $\text{TN} / (\text{TN} + \text{FP}) = 536 / 536 + 43 = 536 / 579 = 0.9272419628 = 0.93$
- **TNR (Mastiff)** = $\text{TN} / (\text{TN} + \text{FP}) = 470 / (470 + 37) = 470 / 507 = 0.9270216963 = 0.93$
- **TNR (Samoyed)** = $\text{TN} / (\text{TN} + \text{FP}) = 616 / (616 + 42) = 616 / 658 = 0.9361702128 = 0.94$

Predicted Class

		Spam	Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

Predicted

	Greyhound	Mastiff	Samoyed	
Actual	Greyhound	250	25	18
	Mastiff	21	320	24
	Samoyed	22	12	180

Accuracy

$$\begin{aligned}
 AC &= (PGG + PMM + PSS) / (PGG + PMG + PSG + PGM + PMM + PSM + \\
 &\quad PGS + PMS + PSS) \\
 &= (250 + 320 + 180) / (250 + 25 + 18 + 21 + 320 + 24 + 22 + 12 + 180) \\
 &= 750 / 872 \\
 &= 0.8600917431 \\
 &= \mathbf{0.86}
 \end{aligned}$$

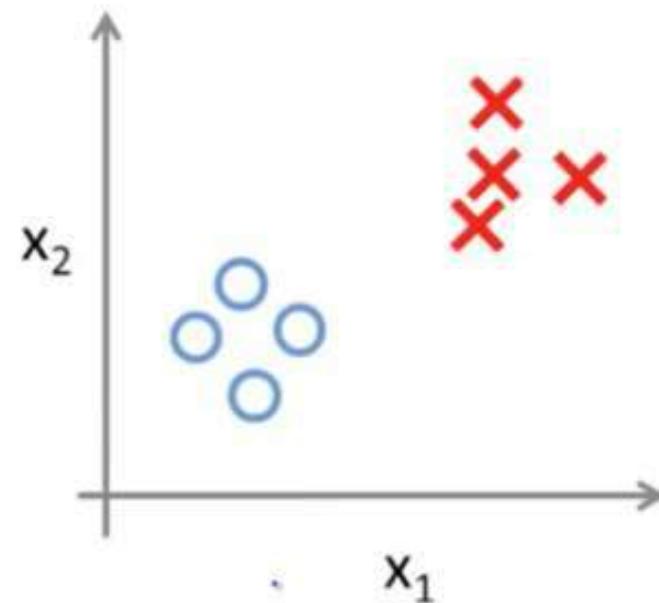
Therefore, the classifier has a total accuracy of **0.86** which is **86%**

Types of Classification

- 1) Binary Classification
- 2) Multi-Class Classification
- 3) Multi-Label Classification
- 4) Imbalanced Classification

Binary Classification

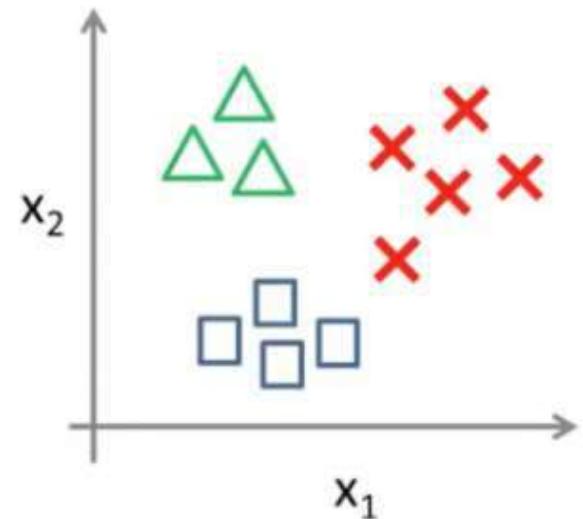
- Binary classification model categories given data into two class labels.
- label 0 is assigned to normal state and label 1 is assigned to abnormal state.
- Examples include:
 - Email spam detection (spam or not).
 - Cat- dog classification
 - Fraud and normal transaction



- Most of the Binary classification model predicts a **Bernoulli probability distribution** for each example.
- Popular algorithms that can be used for binary classification include:
 - Logistic Regression
 - k-Nearest Neighbors
 - Decision Trees
 - Support Vector Machine
 - Naive Bayes

Multi-Class Classification

- Multi-class classification model categories given data into more than two class labels.
- The number of class labels may be very large on some problems.
- Examples include:
 - Face classification.
 - Plant species classification.
 - Optical character recognition.



- Problems that involve predicting a sequence of words, such as text translation models, may also be considered a special type of multi-class classification.
- Each word in the sequence of words to be predicted involves a multi-class classification where the size of the vocabulary defines the number of possible classes that may be predicted and could be tens or hundreds of thousands of words in size.
- Most of the multi-class classification model predicts a **Multinoulli probability distribution** for each example.
- For classification, this means that the model predicts the probability of an example belonging to each class label.

➤ Multi-class classification algorithm are:

- k-Nearest Neighbors.
- Decision Trees.
- Naive Bayes.
- Random Forest.
- Gradient Boosting.

Multi-Label Classification

- Multi-label classification refers to those classification tasks that have **two or more class labels**, where one or more class labels may be predicted for each example.
- For example of photo classification, where a given photo may have multiple objects in the image.
- Classification algorithms used for binary or multi-class classification cannot be used directly for multi-label classification.
- Specialized versions of standard classification algorithms can be used
 - Multi-label Decision Trees
 - Multi-label Random Forests
 - Multi-label Gradient Boosting

Imbalanced Classification

- Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed.
- Typically, imbalanced classification tasks are binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

Examples include:

- Fraud detection.
- Outlier detection.
- Medical diagnostic tests.

Specialized techniques may be used to change the composition of samples in the training dataset by undersampling the majority class or oversampling the minority class. Examples include:

- Random Undersampling.
- SMOTE Oversampling

Specialized modeling algorithms may be used that pay more attention to the minority class when fitting the model on the training dataset, such as cost-sensitive machine learning algorithms.

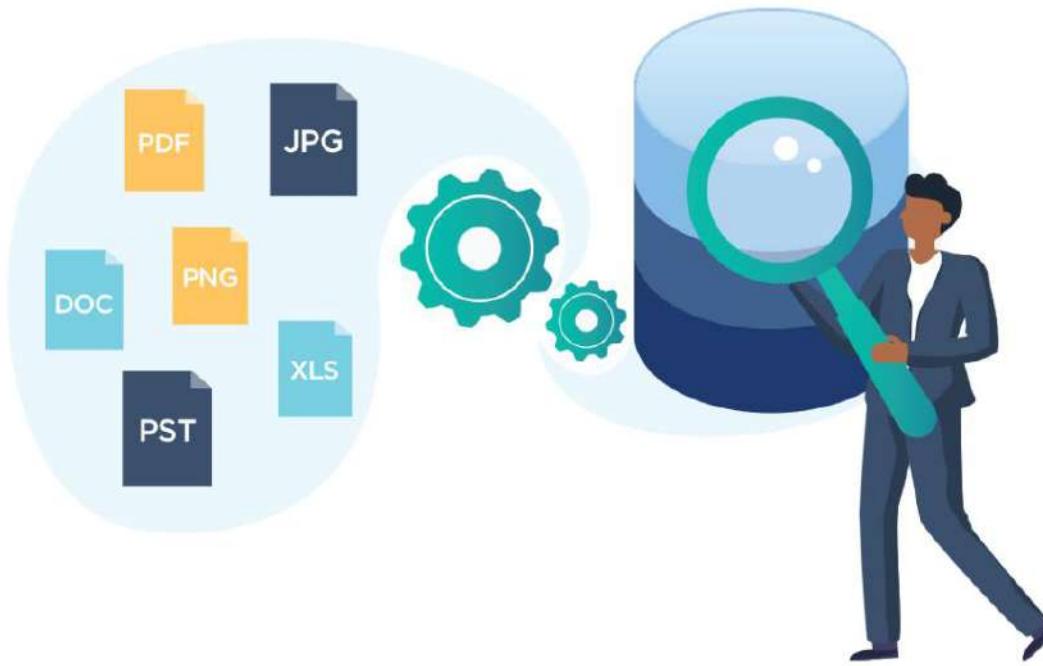
Examples include:

Cost-sensitive Logistic Regression.

Cost-sensitive Decision Trees.

Cost-sensitive Support Vector Machines.

Data Mining (20CP306T)



Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

Syllabus

Unit-I: INTRODUCTION

- ❖ Introduction: What is Data Mining? Motivating Challenges; The origins of data mining; Data Mining Tasks. Types of Data; Data Pre-processing, Measures of Similarity and Dissimilarity.

Unit –II: SUPERVISED LEARNING

- ❖ Classification: Preliminaries; General approach to solving a classification problem; Decision tree induction; Rule-based classifier; Multilinear and Logistic Regression

UNIT 3 ASSOCIATION ANALYSIS

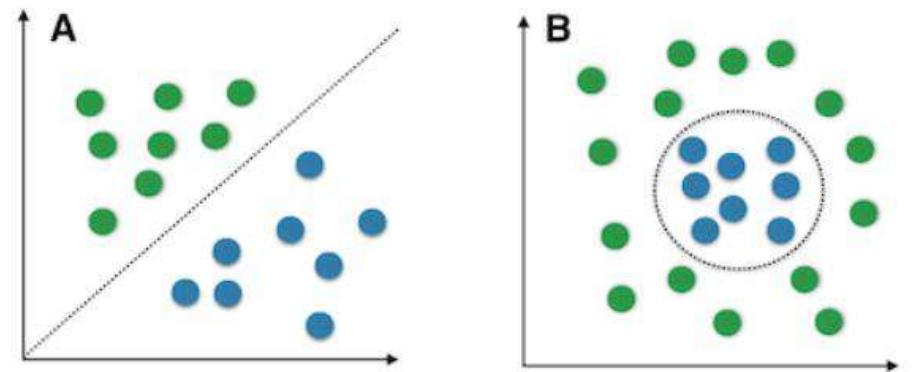
- ❖ Problem definition, Frequent item set generation; Rule Generation; Compact representation of frequent item sets; Alternative methods for generating frequent item sets. FP-Growth algorithm, Evaluation of association patterns, Effect of skewed support distribution, Sequential patterns.

UNIT 4 UNSUPERVISED LEARNING & CLUSTERING

- ❖ Clustering, KNN, Clustering Review, Outlier Detection, Recent Trends in Data

Types of ML Classification Algorithms:

- **Linear Models**
 - Logistic Regression
 - Support Vector Machines
- **Non-linear Models**
 - K-Nearest Neighbours
 - Kernel SVM
 - Naïve Bayes
 - Decision Tree Classification
 - Random Forest Classification



Source: <https://www.kdnuggets.com/2019/12/enabling-deep-learning-revolution.html>

Naïve Bayes Classifier

Random Variables

- A random variable is a variable that can take multiple values depending on the outcome of a random event.
- If the outcomes are finite (for example the 6 possibilities in a die throwing event) the random variable is said to be **discrete**.
- If the possible outcomes are not finite (for example, drawing a number between 0 and 1 can give an infinite number of values), the random variable is said to be **continuous**.
- Random variable can take multiple values so some values will be more often encountered than others. The description of the probability of each possible value that a random variable can take is called its **probability distribution**.

Probability

- Probability is simply how likely something is to happen.
- It is a branch of mathematics that deals with the occurrence of a random event.
- The value is expressed between zero and one.
- The probability of all the events in a sample space sums up to 1

Probability of an event happening = $\frac{\text{Number of ways it can happen}}{\text{Total number of outcomes}}$

Questions

- 1) A die is rolled, find the probability that an even number is obtained.
 - $S = \{1,2,3,4,5,6\}$
 - $E = \{2,4,6\}$
 - $P(E) = n(E) / n(S) = 3 / 6 = 1 / 2$
- 2) Two coins are tossed, find the probability that two heads are obtained.
 - The sample space S is given by.
 $S = \{(H,T), (H,H), (T,H), (T,T)\}$
 - Let E be the event "two heads are obtained".
 $E = \{(H,H)\}$
 - We use the formula of the classical probability.
 $P(E) = n(E) / n(S) = 1 / 4$

Conditional Probability

- Conditional probability is a measure of the probability of an event occurring given that another event has occurred.

$$P(A | B) = \frac{\text{Probability of } A \text{ and } B}{\text{Probability of } B}$$

Probability of
A given B

- What is the probability of person is a ‘Teacher’ given that he is a ‘Male’?

	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

$$P(\text{Teacher} | \text{Male}) = \frac{P(\text{Teacher} \cap \text{Male})}{P(\text{Male})} = 12/60 = 0.2$$

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

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

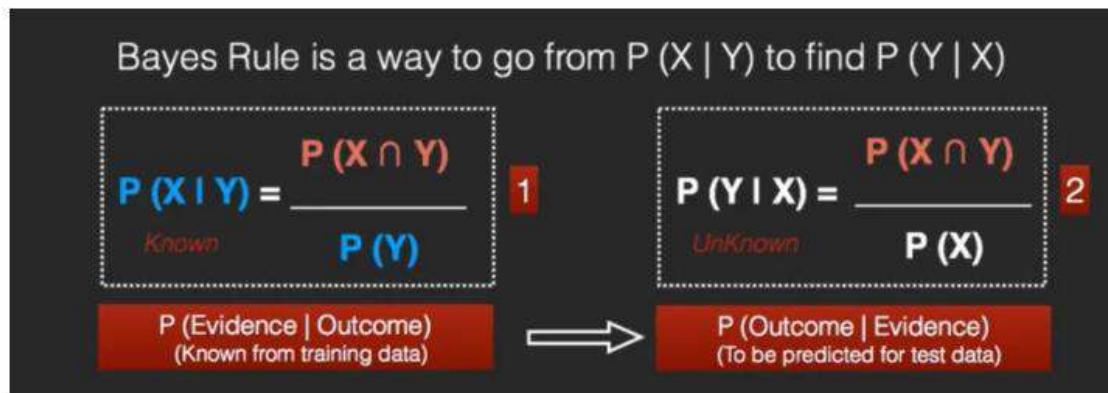
Naive Bayes classifier

- A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task.
- Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems.
- It is based on Bayes theorem which is given by

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood Class Prior Probability
↓ ↓
 $P(c|x)$ $\frac{P(x|c)P(c)}{P(x)}$
Posterior Probability Predictor Prior Probability

- $P(X|Y)$, known from the training dataset, to find $P(Y|X)$.
- For observations in test data, the X would be known while Y is unknown.
- For each row of the test dataset, you want to compute the probability of Y given the X has already happened.



Bayes Rule

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

The equation for Bayes Rule is shown in a red-bordered box: $P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$. To the left of the box, the words 'Bayes Rule' are written in red.

The Assumptions Made Here...

- 1) The predictors/features are independent. That is presence of one particular feature does not affect the other.
- 2) Each feature is given the same weight(or importance).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

$$X = (x_1, x_2, x_3, \dots, x_n)$$

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)}$$

- In the case of multiclass our task is to find the class y with maximum probability.

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y)$$

How Naïve Bayes Classifier Work

Example

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

- today = (Sunny, Hot, Normal, False), probability of playing golf is given by:

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
Sunny	Hot	Normal	False	Yes

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

$$P(Yes|today) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(today)}$$

$$P(No|today) = \frac{P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)}{P(today)}$$

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Outlook

	Yes	No	P(yes)	P(no)
Sunny	3	2	3/9	2/5
Overcast	4	0	4/9	0/5
Rainy	2	3	2/9	3/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Play	P(Yes)/P(No)	
Yes	9	9/14
No	5	5/14
Total	14	100%

Outlook

	Yes	No	P(yes)	P(no)
Sunny	3	2	3/9	2/5
Overcast	4	0	4/9	0/5
Rainy	2	3	2/9	3/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

So, probability of playing golf is given by:

$$P(\text{Yes}|\text{today}) = \frac{P(\text{SunnyOutlook}|\text{Yes})P(\text{HotTemperature}|\text{Yes})P(\text{NormalHumidity}|\text{Yes})P(\text{NoWind}|\text{Yes})P(\text{Yes})}{P(\text{today})}$$

and probability to not play golf is given by:

$$P(\text{No}|\text{today}) = \frac{P(\text{SunnyOutlook}|\text{No})P(\text{HotTemperature}|\text{No})P(\text{NormalHumidity}|\text{No})P(\text{NoWind}|\text{No})P(\text{No})}{P(\text{today})}$$

Since, $P(\text{today})$ is common in both probabilities, we can ignore $P(\text{today})$ and find proportional probabilities as:

$$P(\text{Yes}|\text{today}) \propto \frac{3}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \cdot \frac{9}{14} \approx 0.0141$$

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

$$P(\text{No}|\text{today}) \propto \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{5}{14} \approx 0.0068$$

Now, since

$$P(\text{Yes}|\text{today}) + P(\text{No}|\text{today}) = 1$$

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

$$P(\text{Yes}|\text{today}) = \frac{0.0141}{0.0141+0.0068} = 0.67$$

and

$$P(\text{No}|\text{today}) = \frac{0.0068}{0.0141+0.0068} = 0.33$$

Question

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes | X) = P(Rainy | Yes) \times P(Cool | Yes) \times P(High | Yes) \times P(True | Yes) \times P(Yes)$$

$$P(Yes | X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529$$

$$0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No | X) = P(Rainy | No) \times P(Cool | No) \times P(High | No) \times P(True | No) \times P(No)$$

$$P(No | X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057$$

$$0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

Naïve Bayes example for text

- <https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/>

Types of Naive Bayes model

- 1) **Gaussian**: In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution.
- 2) **Bernoulli Naive Bayes**: In the multivariate Bernoulli event model, features are **independent booleans** (binary variables) describing inputs.
 - Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence(i.e. a word occurs in a document or not) features are used rather than term frequencies(i.e. frequency of a word in the document).
- 3) **Multinomial Naive Bayes**: **Feature vectors** represent the **frequencies** with which certain events have been generated by a multinomial distribution.
 - This is mostly used for **document classification problem**, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

➤ Advantages of Naive Bayes

1. When assumption of independent predictors holds true, a Naive Bayes classifier performs better as compared to other models.
2. Naive Bayes requires a small amount of training data to estimate the test data. So, the training period is less.
3. Naive Bayes is also easy to implement.

➤ Disadvantages of Naive Bayes

1. Main limitation of Naive Bayes is the **assumption of independent predictors**. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it is almost impossible that we get a set of predictors which are completely independent.
2. If categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as **Zero Frequency**.

Applications of Naive Bayes Algorithms

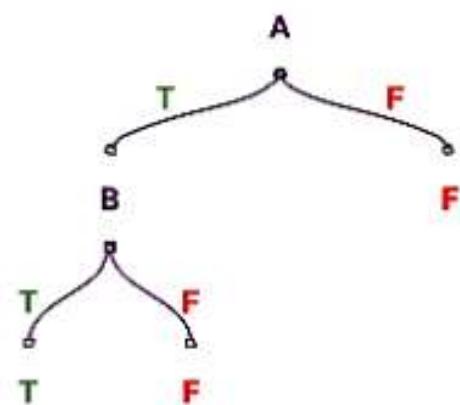
- 1) Real time Prediction
- 2) Multi class Prediction
- 3) Text classification/ Spam Filtering/ Sentiment Analysis:
- 4) Recommendation System

Decision Tree

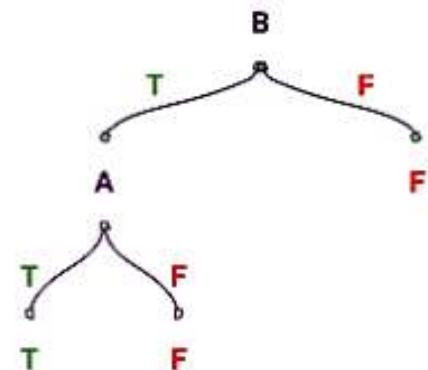
Decision Tree

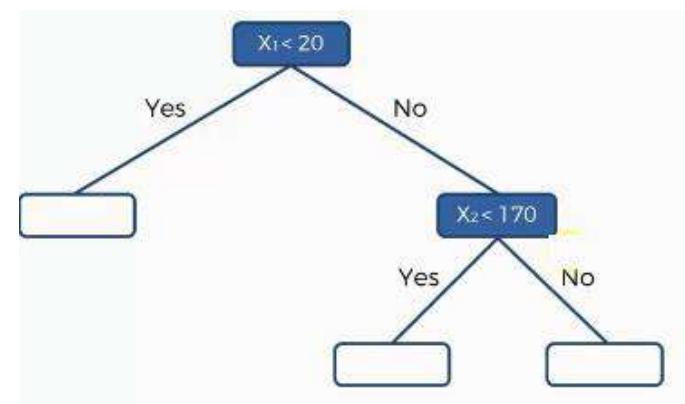
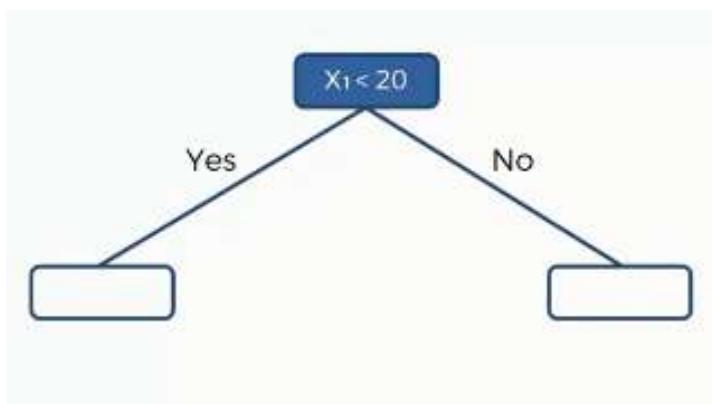
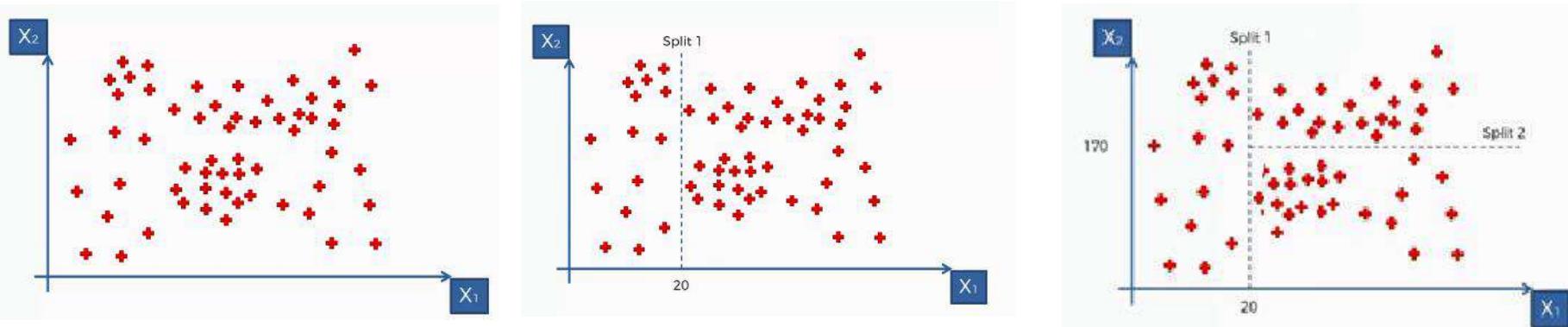
- It is a supervised learning and can be used to solve both regression and classification problems.
- A decision tree is a graphical representation of possible solutions to a decision based on certain conditions.
- A decision tree is a tree-like graph with nodes representing a question; edges represent the answers to the question; and the leaf node represent the actual output or class label.
- The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

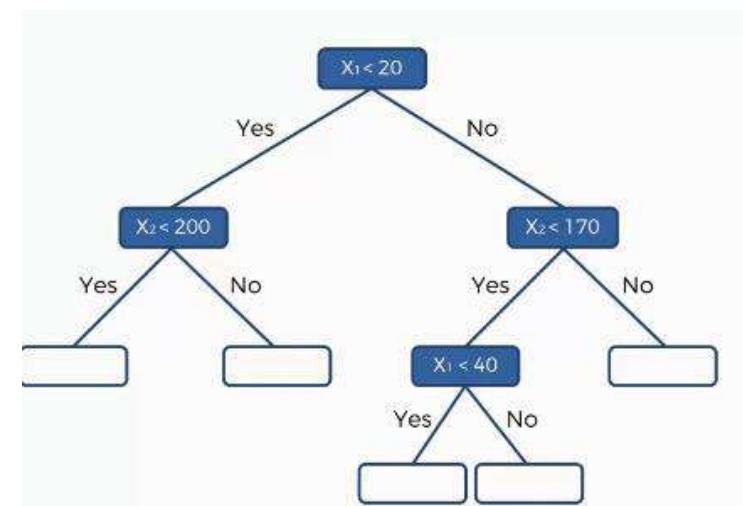
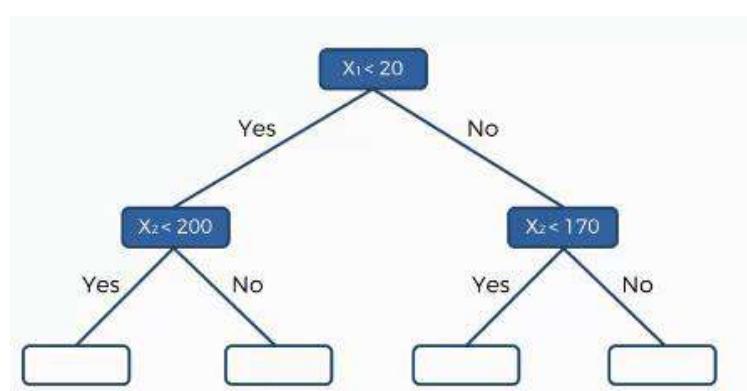
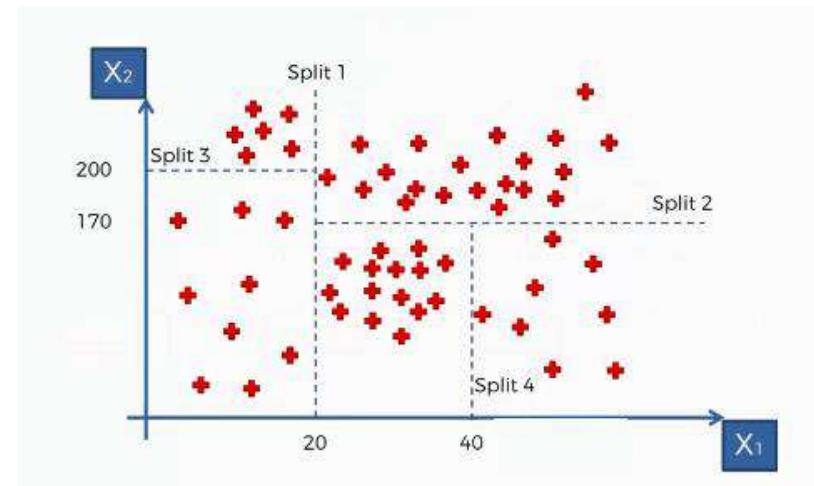
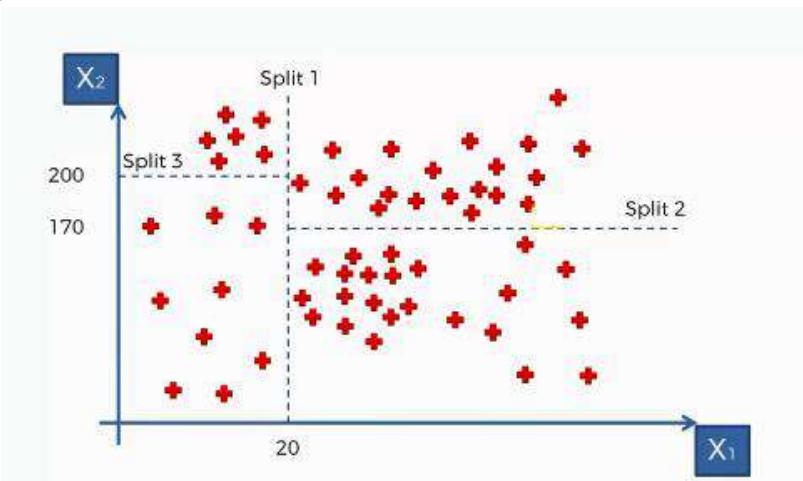
A	B	A AND B
F	F	F
F	T	F
T	F	F
T	T	T



OR





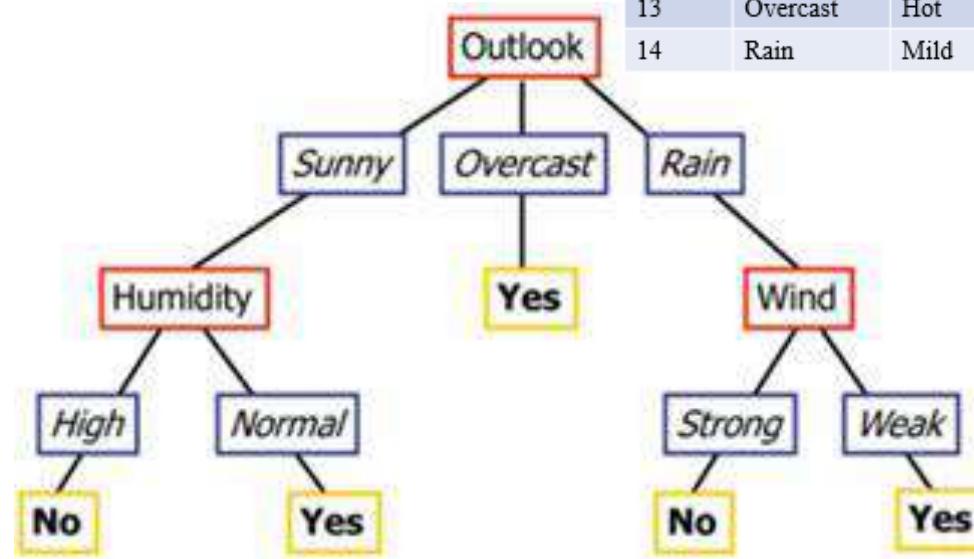


Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

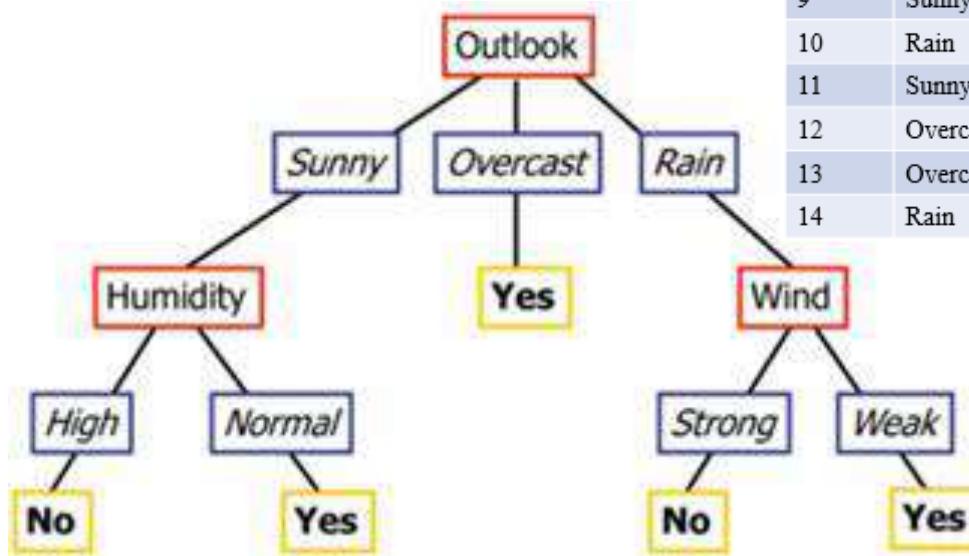
Outlook= sunny, temperature=Cool, Humidity=High, wind=strong

Outlook= sunny, temperature=Cool,
Humidity=High, wind=strong

Day	Outlook	Temperature	Humidity	Wind	Play Cricket
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



How to Decide Root Node



Day	Outlook	Temperature	Humidity	Wind	Play Cricket
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

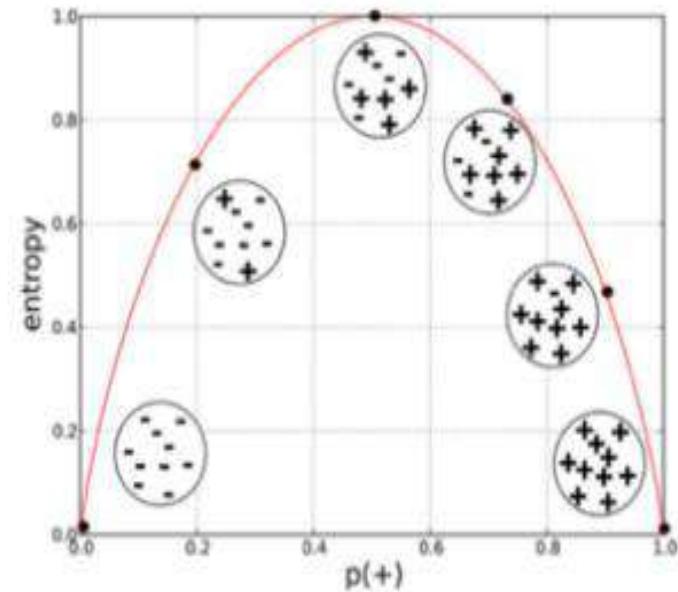
How to find Root Node?

- In Decision Tree the major challenge is to identification of the attribute for the root node in each level. This process is known as attribute selection.
 - 1) Information Gain
 - 2) Gini Index

Entropy

- Entropy is the measure of the homogeneity of a sample in a node. That means it measure the uncertainty in the data.
- If the sample is completely homogeneous the entropy is zero and if the sample is equally divided then it has entropy of one.

- Entropy is lowest at the extremes, when the bubble either contains no positive instances or only positive instances.
- That is, when the bubble is pure the disorder is 0.
- Entropy is highest in the middle when the bubble is evenly split between positive and negative instances.
- Entropy is a measure of disorder or uncertainty and the **goal of machine learning models** and Data Scientists in general is to **reduce uncertainty**.



<https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>

$$\text{Entropy} = - \sum p(X) \log p(X)$$

Entropy is measured **between** 0 and 1.

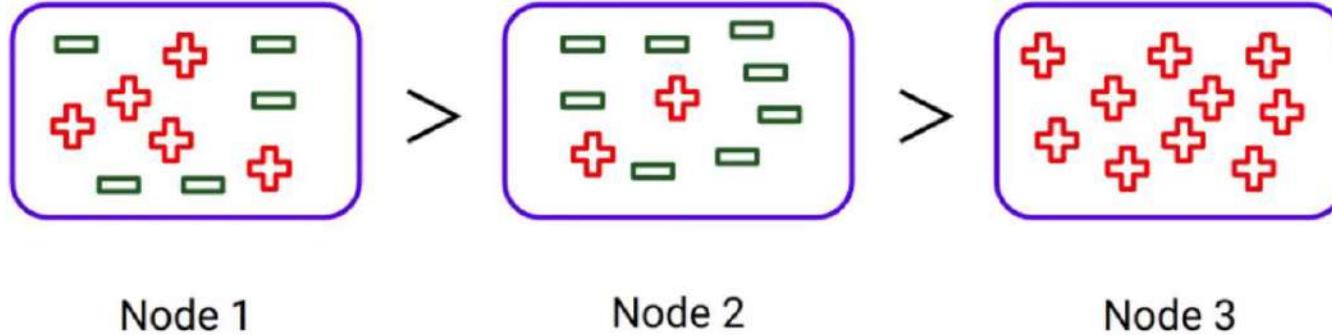


$$\text{Entropy} = - \left(\frac{3}{8} \log_2 \frac{3}{8} + \frac{2}{8} \log_2 \frac{2}{8} + \frac{2}{8} \log_2 \frac{2}{8} + \frac{1}{8} \log_2 \frac{1}{8} \right)$$

$$\text{Entropy} = 0.571$$

Information Gain

- This measure of purity is called the information. We can define information gain as a measure of how much information a feature provides about a class.
- It tells us how important a given attribute of the feature vectors is.
- Information gain can also be used for feature selection
- When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. **Information gain is a measure of this change in entropy.**



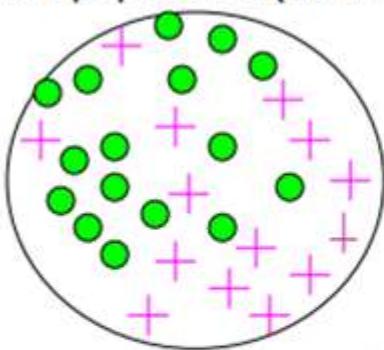
- So as the impurity of the node increases, we require more information to describe them. And that's why Node 1 will require more information as compared to the other nodes.
- So we can say that more impure nodes require more information to describe.

- Information gain is calculated by comparing the entropy of the dataset before and after a transformation.
- The greater the reduction in this uncertainty, the more information is gained.
- Lower probability events have more information, higher probability events have less information.
- **Skewed Probability Distribution (*unsurprising*):** Low entropy
- **Balanced Probability Distribution (*surprising*):** Higher entropy
- A larger information gain suggests a lower entropy

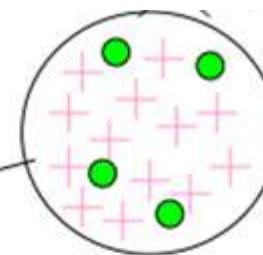
Information Gain = $\text{entropy}(\text{parent}) - [\text{average entropy}(\text{children})]$

child entropy $-\left(\frac{13}{17} \cdot \log \frac{13}{17}\right) - \left(\frac{4}{17} \cdot \log \frac{4}{17}\right) = 0.78$

Entire population (30 instances)

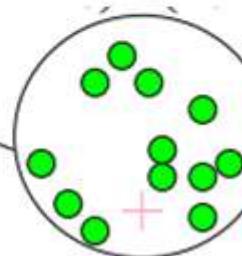


parent entropy $-\left(\frac{14}{30} \cdot \log \frac{14}{30}\right) - \left(\frac{16}{30} \cdot \log \frac{16}{30}\right) = 0.99$



17 instances

child entropy $-\left(\frac{1}{13} \cdot \log \frac{1}{13}\right) - \left(\frac{12}{13} \cdot \log \frac{12}{13}\right) = 0.39$

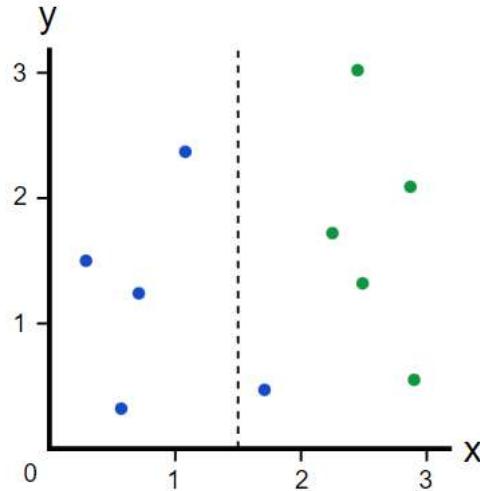
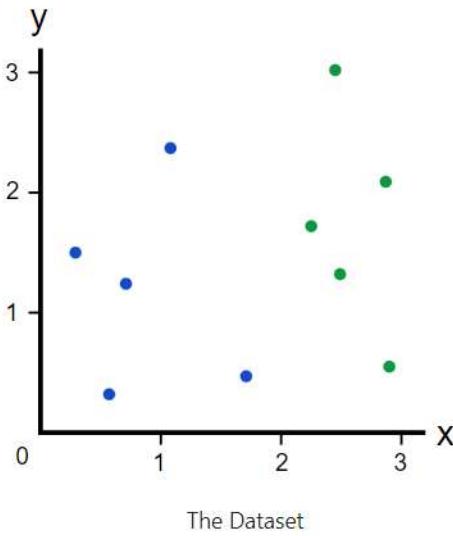


13 instances

(Weighted) Average Entropy of Children = $\left(\frac{17}{30} \cdot 0.787\right) + \left(\frac{13}{30} \cdot 0.391\right) = 0.615$

Information Gain= 0.996 - 0.615 = 0.38

if we made a split at $x = 1.5$? $x=1.5$?



Before the split, we had 5 blues and 5 greens, so the entropy was

$$\begin{aligned} E_{before} &= -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) \\ &= \boxed{1} \end{aligned}$$

Left Branch has 4 blues, so $E_{left} = \boxed{0}$ because it's a dataset of all one color.

Right Branch has 1 blue and 5 greens, so

$$\begin{aligned} E_{right} &= -\left(\frac{1}{6} \log_2\left(\frac{1}{6}\right) + \frac{5}{6} \log_2\left(\frac{5}{6}\right)\right) \\ &= \boxed{0.65} \end{aligned}$$

weighting the entropy

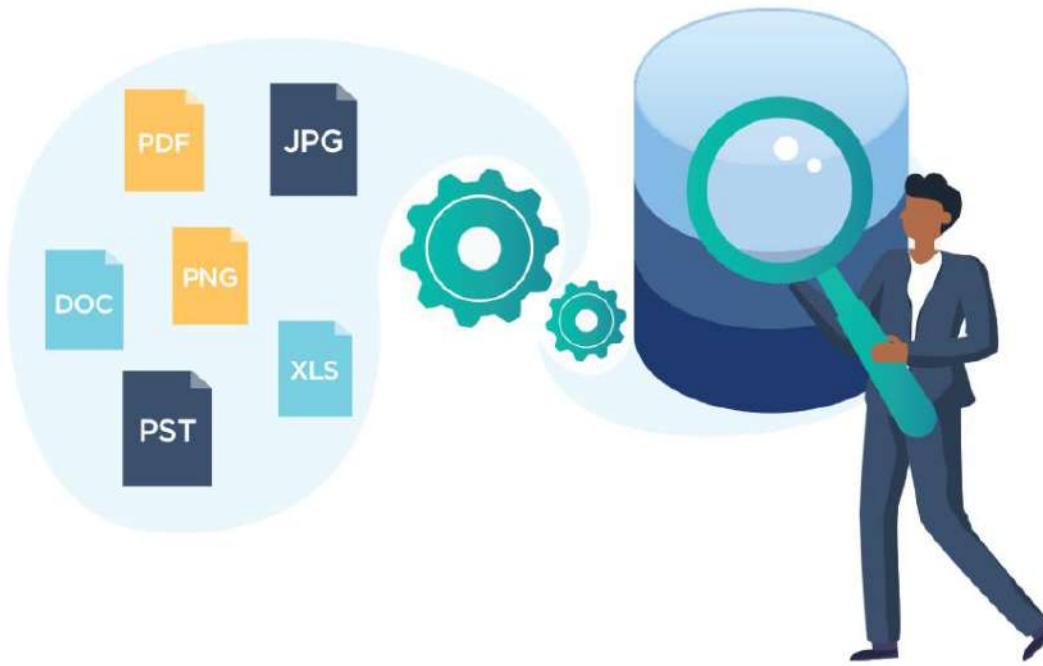
Information Gain = how much Entropy we removed

$$\text{Gain} = 1 - 0.39 = \boxed{0.61}$$

$$\begin{aligned} E_{split} &= 0.4 * 0 + 0.6 * 0.65 \\ &= \boxed{0.39} \end{aligned}$$

: higher Information Gain = more Entropy removed,

Data Mining (20CP306T)

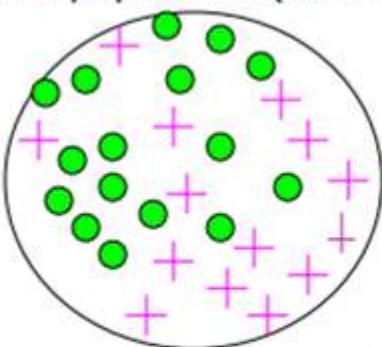


Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

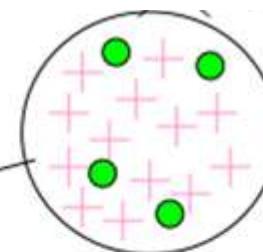
$$\text{Information Gain} = \text{entropy}(\text{parent}) - [\text{weighted_average entropy}(\text{children})]$$

child entropy $-\left(\frac{13}{17} \cdot \log \frac{13}{17}\right) - \left(\frac{4}{17} \cdot \log \frac{4}{17}\right) = 0.78$

Entire population (30 instances)

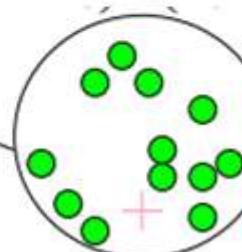


parent entropy $-\left(\frac{14}{30} \cdot \log \frac{14}{30}\right) - \left(\frac{16}{30} \cdot \log \frac{16}{30}\right) = 0.99$



17 instances

child entropy $-\left(\frac{1}{13} \cdot \log \frac{1}{13}\right) - \left(\frac{12}{13} \cdot \log \frac{12}{13}\right) = 0.39$

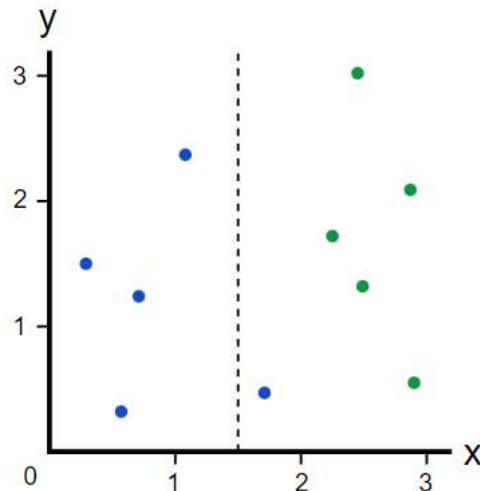
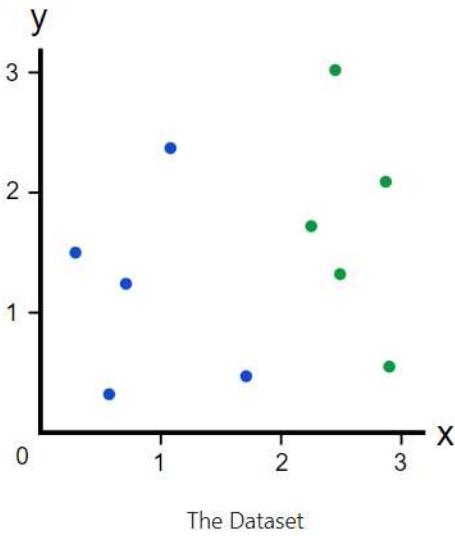


13 instances

(Weighted) Average Entropy of Children = $\left(\frac{17}{30} \cdot 0.787\right) + \left(\frac{13}{30} \cdot 0.391\right) = 0.615$

Information Gain= 0.996 - 0.615 = 0.38

if we made a split at $x = 1.5$? $x=1.5$?



Before the split, we had 5 blues and 5 greens, so the entropy was

$$\begin{aligned} E_{before} &= -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) \\ &= \boxed{1} \end{aligned}$$

Left Branch has 4 blues, so $E_{left} = \boxed{0}$ because it's a dataset of all one color.

Right Branch has 1 blue and 5 greens, so

$$\begin{aligned} E_{right} &= -\left(\frac{1}{6} \log_2\left(\frac{1}{6}\right) + \frac{5}{6} \log_2\left(\frac{5}{6}\right)\right) \\ &= \boxed{0.65} \end{aligned}$$

weighting the entropy

Information Gain = how much Entropy we removed

$$\text{Gain} = 1 - 0.39 = \boxed{0.61}$$

$$\begin{aligned} E_{split} &= 0.4 * 0 + 0.6 * 0.65 \\ &= \boxed{0.39} \end{aligned}$$

: higher Information Gain = more Entropy removed,

Gini Index

- Gini impurity is a function that determines how well a decision tree was split. Basically, it helps us to determine which splitter is best so that we can build a pure decision tree.
- Its measure the ‘impurity’. If all the elements belong to a single class, then it can be called pure.
- It means an attribute with **lower gini index** should be preferred.
- The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes.
- A Gini Index of 0.5 denotes equally distributed elements into some classes.

- Sklearn supports “gini” criteria for Gini Index and by default, it takes “gini” value.
- The degree of Gini index varies between 0 and 1

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

where p_i is the probability of an object being classified to a particular class.

- While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

<https://blog.quantinsti.com/gini-index/#:~:text=Gini%20index%20or%20Gini%20impurity,when%20it%20is%20randomly%20chosen.&text=A%20Gini%20Index>

%20of%200.5%20denotes%20equally%20distributed%20elements%20into%20some%20classes.

Gini Index for ‘Past Trend’.

- $P(\text{Past Trend}=\text{Positive})$: 6/10
- $P(\text{Past Trend}=\text{Negative})$: 4/10
- If (Past Trend = Positive & Return = Up), probability = 4/6
- If (Past Trend = Positive & Return = Down), probability = 2/6
- **Gini index = $1 - ((4/6)^2 + (2/6)^2) = 0.45$**
- If (Past Trend = Negative & Return = Up), probability = 0
- If (Past Trend = Negative & Return = Down), probability = 4/4
- **Gini index = $1 - ((0)^2 + (4/4)^2) = 0$**
- Weighted sum of the Gini Indices can be calculated as follows:
- Gini Index for Past Trend = $(6/10)0.45 + (4/10)0 = 0.27$

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

• Calculation of Gini Index for Open Interest

- $P(\text{Open Interest}=\text{High}) = 4/10$
- $P(\text{Open Interest}=\text{Low}) = 6/10$
- If ($\text{Open Interest} = \text{High}$ & $\text{Return} = \text{Up}$), probability = $2/4$
- If ($\text{Open Interest} = \text{High}$ & $\text{Return} = \text{Down}$), probability = $2/4$
- Gini index = $1 - ((2/4)^2 + (2/4)^2) = 0.5$

- If ($\text{Open Interest} = \text{Low}$ & $\text{Return} = \text{Up}$), probability = $2/6$
- If ($\text{Open Interest} = \text{Low}$ & $\text{Return} = \text{Down}$), probability = $4/6$
- Gini index = $1 - ((2/6)^2 + (4/6)^2) = 0.45$

- Weighted sum of the Gini Indices can be calculated as follows:
- Gini Index for Open Interest = $(4/10)0.5 + (6/10)0.45 = 0.47$

• Gini Index for Trading Volume

- $P(\text{Trading Volume}=\text{High})$: 7/10
- $P(\text{Trading Volume}=\text{Low})$: 3/10
- If ($\text{Trading Volume} = \text{High} \ \& \ \text{Return} = \text{Up}$), probability = 4/7
- If ($\text{Trading Volume} = \text{High} \ \& \ \text{Return} = \text{Down}$), probability = 3/7
- Gini index = $1 - ((4/7)^2 + (3/7)^2) = 0.49$
- If ($\text{Trading Volume} = \text{Low} \ \& \ \text{Return} = \text{Up}$), probability = 0
- If ($\text{Trading Volume} = \text{Low} \ \& \ \text{Return} = \text{Down}$), probability = 3/3
- Gini index = $1 - ((0)^2 + (1)^2) = 0$
- Weighted sum of the Gini Indices can be calculated as follows:
- Gini Index for Trading Volume = $(7/10)0.49 + (3/10)0 = 0.34$

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Attributes/Features	Gini Index
Past Trend	0.27
Open Interest	0.47
Trading Volume	0.34

- We observe that ‘Past Trend’ has the lowest Gini Index and hence it will be chosen as the root node.
- We will repeat the same procedure to determine the sub-nodes or branches of the decision tree.

- We will calculate the Gini Index for the ‘Positive’ branch of Past Trend as follows:

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Positive	Low	High	Up
Positive	High	High	Up
Positive	Low	Low	Down
Positive	Low	Low	Down
Positive	High	High	Up

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Positive	Low	High	Up
Positive	High	High	Up
Positive	Low	Low	Down
Positive	Low	Low	Down
Positive	High	High	Up

- **Gini Index of Open Interest for Positive Past Trend**

- P(Open Interest=High): 2/6
- P(Open Interest=Low): 4/6
- If (Open Interest = High & Return = Up), probability = 2/2
- If (Open Interest = High & Return = Down), probability = 0
- Gini index = $1 - (\text{sq}(2/2) + \text{sq}(0)) = 0$

- If (Open Interest = Low & Return = Up), probability = 2/4
- If (Open Interest = Low & Return = Down), probability = 2/4
- Gini index = $1 - (\text{sq}(2/4) + \text{sq}(2/4)) = 0.50$

- Weighted sum of the Gini Indices can be calculated as follows:
- Gini Index for Open Interest = $(2/6)0 + (4/6)0.50 = 0.33$

• Gini Index for Trading Volume

- $P(\text{Trading Volume}=\text{High})$: 4/6
- $P(\text{Trading Volume}=\text{Low})$: 2/6
- If ($\text{Trading Volume} = \text{High}$ & $\text{Return} = \text{Up}$), probability = 4/4
- If ($\text{Trading Volume} = \text{High}$ & $\text{Return} = \text{Down}$), probability = 0
- Gini index = $1 - (\text{sq}(4/4) + \text{sq}(0)) = 0$

- If ($\text{Trading Volume} = \text{Low}$ & $\text{Return} = \text{Up}$), probability = 0
- If ($\text{Trading Volume} = \text{Low}$ & $\text{Return} = \text{Down}$), probability = 2/2
- Gini index = $1 - (\text{sq}(0) + \text{sq}(2/2)) = 0$

- Weighted sum of the Gini Indices can be calculated as follows:
- Gini Index for Trading Volume = $(4/6)0 + (2/6)0 = 0$

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Positive	Low	High	Up
Positive	High	High	Up
Positive	Low	Low	Down
Positive	Low	Low	Down
Positive	High	High	Up

- We will split the node further using the ‘Trading Volume’ feature, as it has the minimum Gini index.

Attribute/Feature	Gini Index
Open Interest	0.33
Trading Volume	0

How Decision Tree Work

Information Gain

- This measure of purity is called the information. It represents the expected amount of information that would be needed to specify whether a new instance should be classified male or female.

Information Gain = $\text{entropy}(\text{parent}) - [\text{weighted entropy}(\text{children})]$

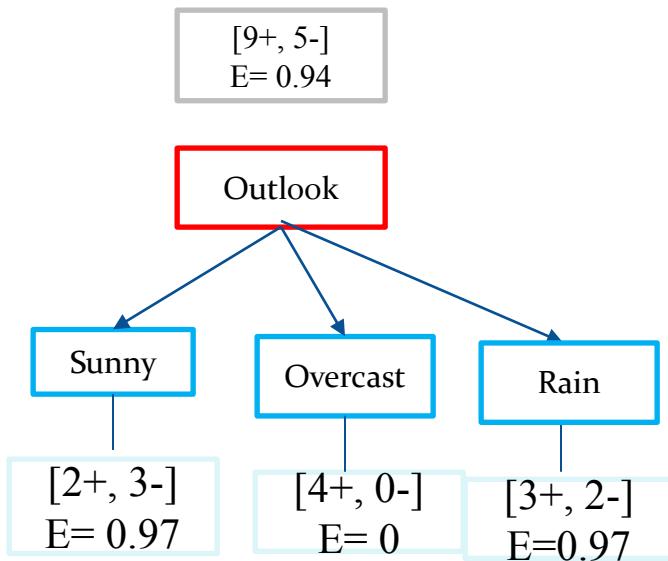
- Gini Index or Gini impurity is a metric to measure how often a randomly chosen element would be incorrectly identified.

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	Weak / (False)	No
2	Sunny	Hot	High	Strong / (True)	No
3	Overcast	Hot	High	Weak / (False)	Yes
4	Rain	Mild	High	Weak / (False)	Yes
5	Rain	Cool	Normal	Weak / (False)	Yes
6	Rain	Cool	Normal	Strong / (True)	No
7	Overcast	Cool	Normal	Strong / (True)	Yes
8	Sunny	Mild	High	Weak / (False)	No
9	Sunny	Cool	Normal	Weak / (False)	Yes
10	Rain	Mild	Normal	Weak / (False)	Yes
11	Sunny	Mild	Normal	Strong / (True)	Yes
12	Overcast	Mild	High	Strong / (True)	Yes
13	Overcast	Hot	Normal	Weak / (False)	Yes
14	Rain	Mild	High	Strong / (True)	No

Outlook= sunny, temperature=Cool, Humidity=High, wind=strong

Information Gain = $\text{entropy}(\text{parent}) - [\text{weighted entropy}(\text{children})]$



Play Golf	
Yes	No
9	5

$\text{Entropy}(\text{PlayGolf}) = \text{Entropy}(5,9)$
 $= \text{Entropy}(0.36, 0.64)$
 $= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64)$
 $= 0.94$

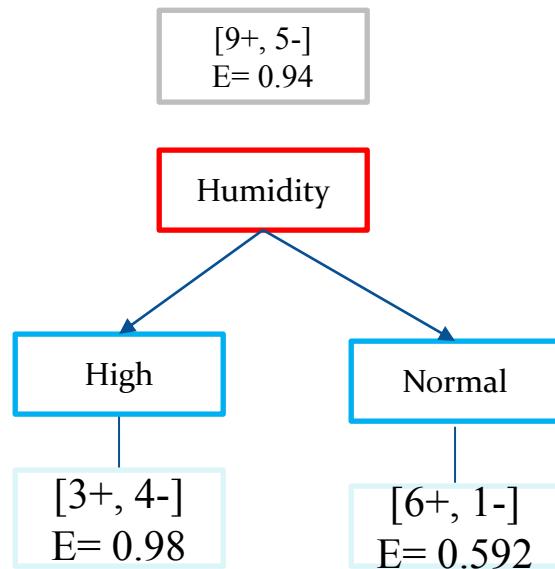
		Play Golf	
		Yes	No
Outlook	Sunny	2	3
	Overcast	4	0
	Rainy	3	2

$$\begin{aligned}
 \text{Gain}(S_O) &= \\
 0.94 - \frac{5}{14} * 0.97 - \frac{5}{14} * 0.97 & \\
 &= 0.247
 \end{aligned}$$

$$3/7 * \log(3/7) + 4/7 * \log(4/7)$$

$$E = 0.98$$

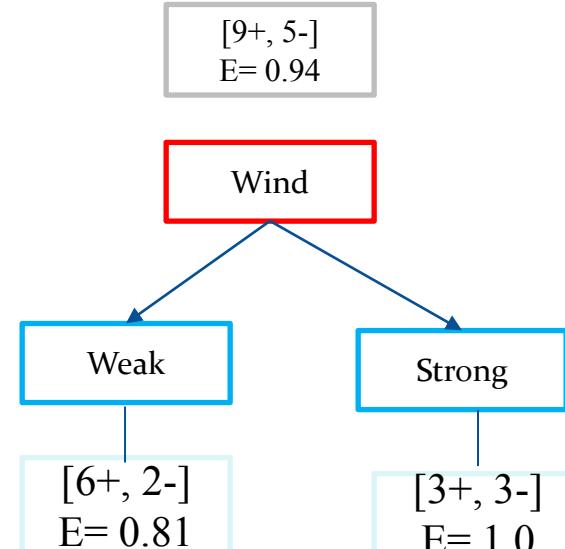
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1



$$\text{Gain}(S, \text{humidity}) =$$

$$0.94 - \frac{7}{14} * 0.98 - \frac{7}{14} * 0.59$$

$$= 0.152$$



$$\text{Gain}(S, \text{wind}) =$$

$$0.94 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1.0$$

$$= 0.048$$

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

		Play Golf	
		Yes	No
Outlook	Sunny	2	3
	Overcast	4	0
	Rainy	3	2
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

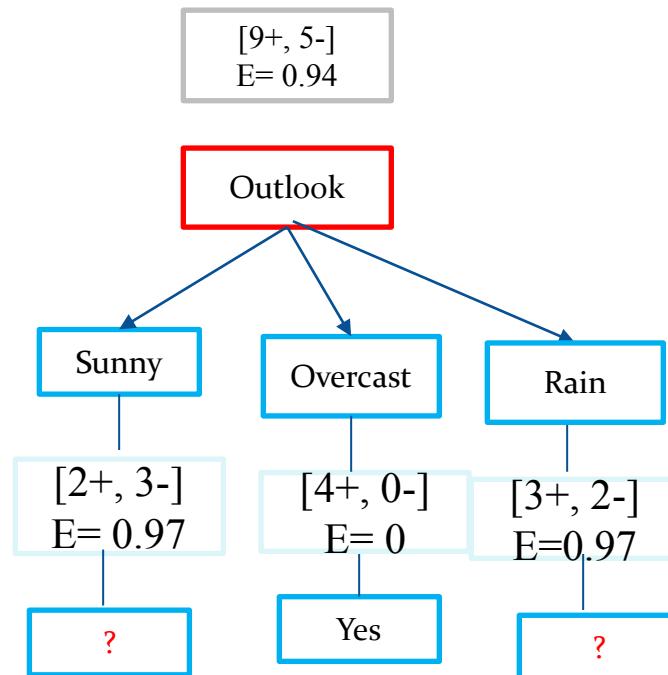
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Mild	High	Weak	No
4	Sunny	Cool	Normal	Weak	Yes
5	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Rain	Mild	High	Weak	Yes
2	Rain	Cool	Normal	Weak	Yes
3	Rain	Cool	Normal	Strong	No
4	Rain	Mild	Normal	Weak	Yes
5	Rain	Mild	High	Strong	No

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Overcast	Hot	High	Weak	Yes
2	Overcast	Cool	Normal	Strong	Yes
3	Overcast	Mild	High	Strong	Yes
4	Overcast	Hot	Normal	Weak	Yes



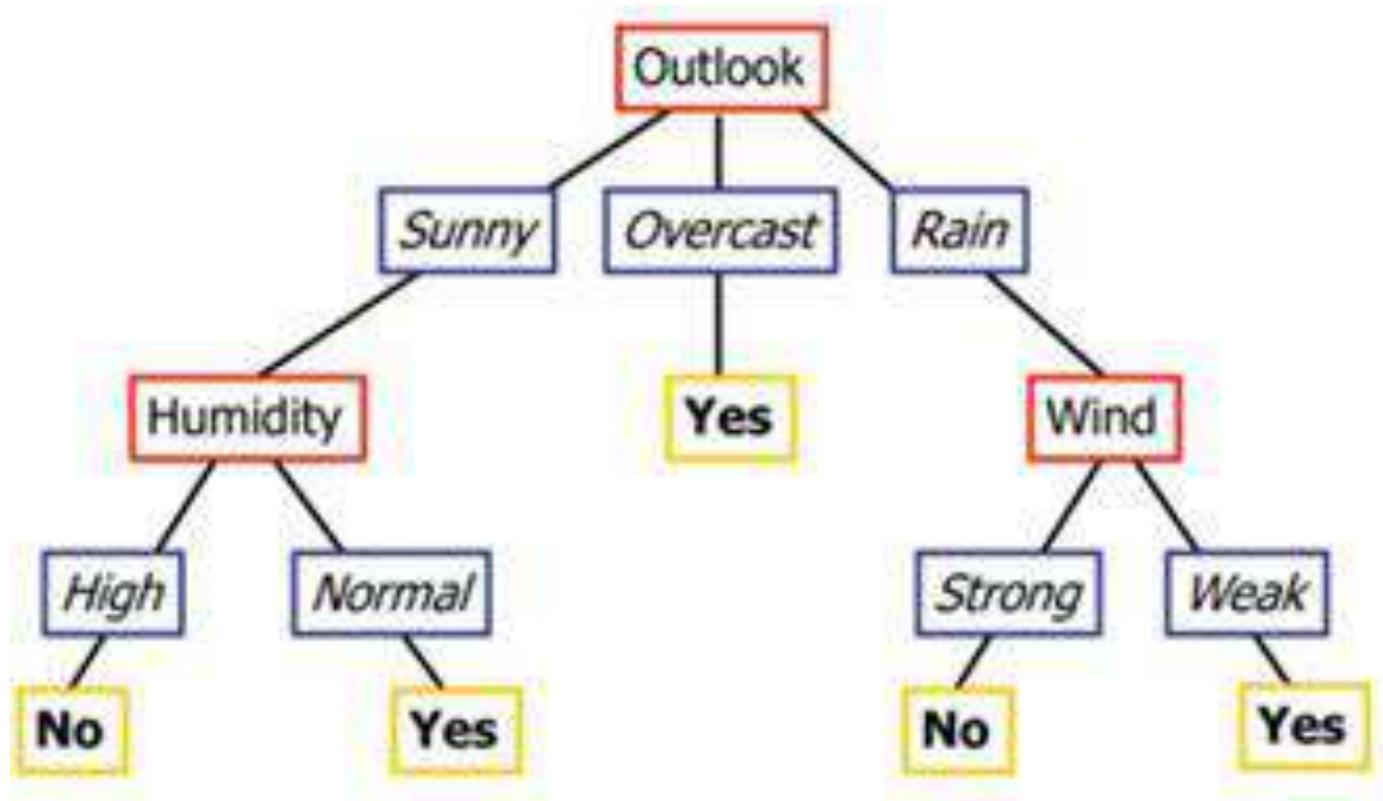
Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Mild	High	Weak	No
4	Sunny	Cool	Normal	Weak	Yes
5	Sunny	Mild	Normal	Strong	Yes

$$\text{Gain}(S_{sunny}, \text{humidity}) = 0.97 - \frac{3}{5} * 0.0 - \frac{2}{5} * 0.0 = 0.97$$

$$\text{Gain}(S_{sunny}, \text{temp}) = 0.97 - \frac{2}{5} * 0.0 - \frac{2}{5} * 1.0 - \frac{1}{5} * 0.0 = 0.57$$

$$\text{Gain}(S_{sunny}, \text{wind}) = 0.97 - \frac{2}{5} * 1.0 - \frac{3}{5} * 0.918 = 0.19$$

Outlook= sunny, temperature=Cool, Humidity=High, wind=strong



Applications

- 1) Predicting high occupancy dates for hotels
- 2) Identifying factors leading to better gross margins on a retail chain .
- 3) Demand forecasting

Strengths of Decision Tree

- 1) Decision trees are able to generate understandable rules.
- 2) Decision trees perform classification without requiring much computation.
- 3) Decision trees are able to handle both continuous and categorical variables.
- 4) Decision trees provide a clear indication of which fields are most important for prediction or classification.

Weaknesses of Decision Tree

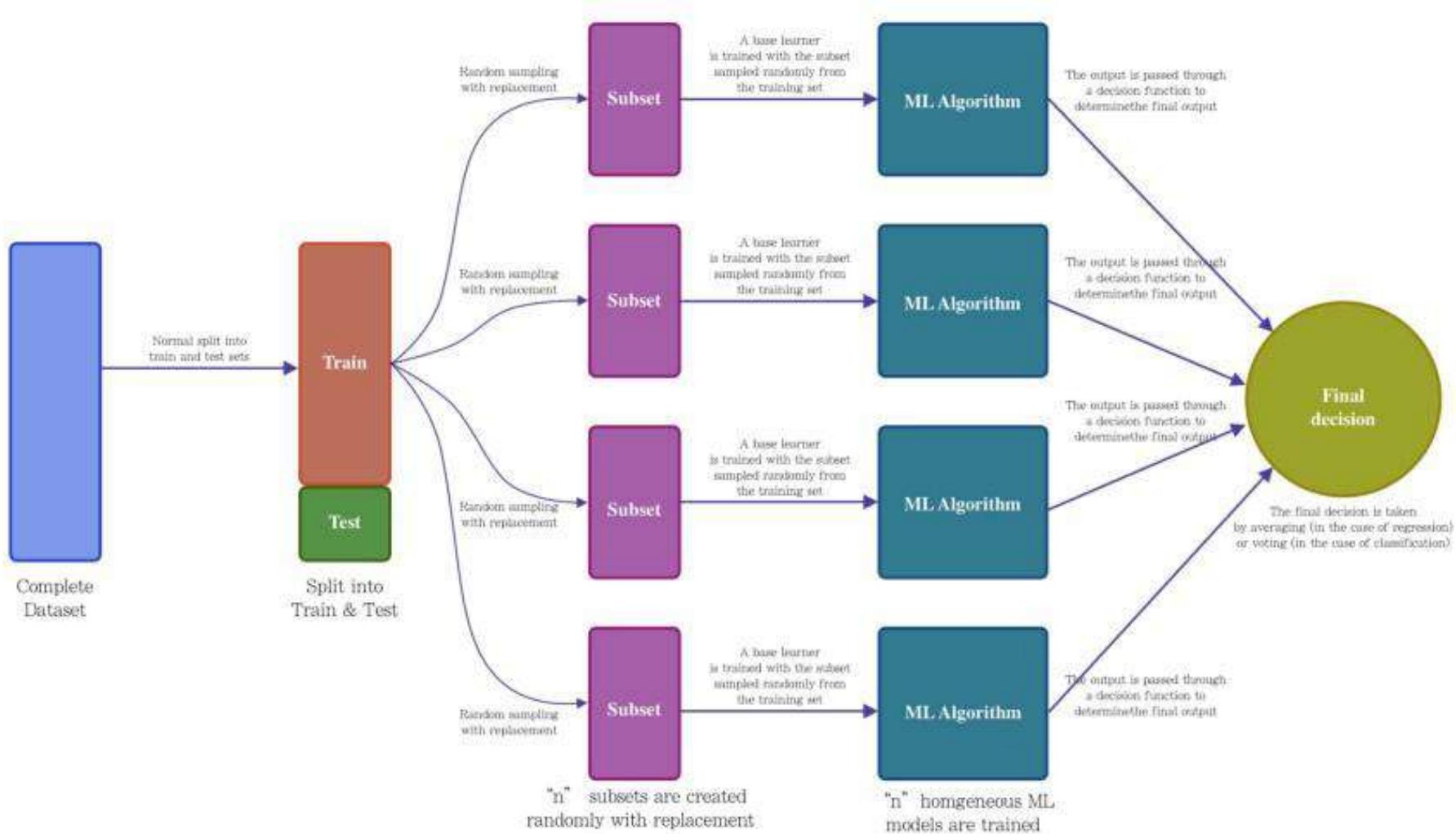
- 1) Unstable in nature. A small change in the data can result in a major change in the structure of the decision tree
- 2) Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a **continuous attribute**.
- 3) Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
- 4) Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive.
- 5) Prone to overfit

Solution to avoid Weaknesses of Decision Tree

- 1) Pruning. In some situations, stopping rules do not work well. An alternative way to build a decision tree model is to grow a large tree first, and then prune it to optimal size by removing nodes that provide less additional information.
- 2) The removal of irrelevant nodes can help reduce the chance of creating an over-fitting tree.
- 3) Bagging
- 4) Boosting

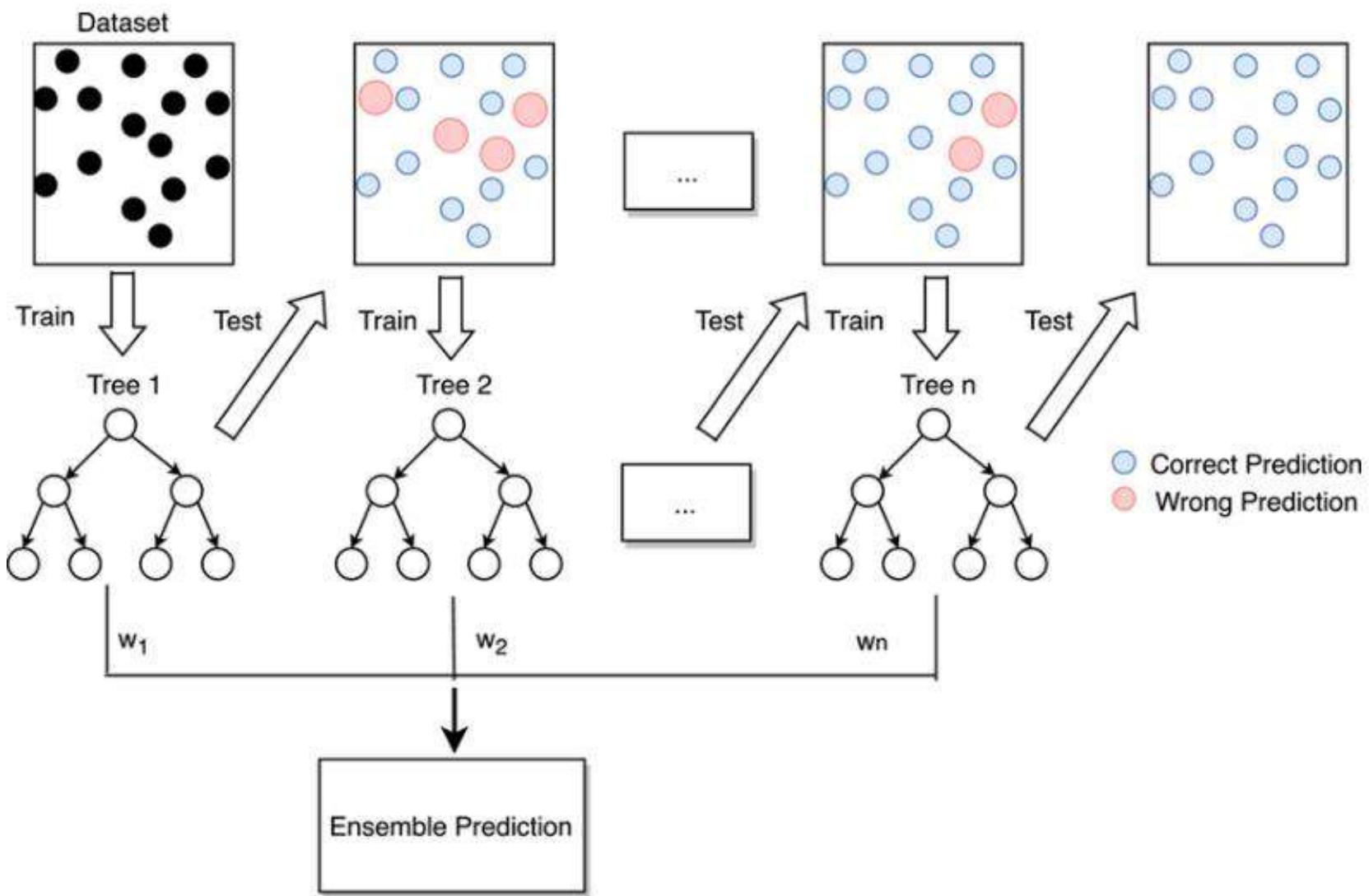
Bagging

- Bagging (Bootstrap Aggregation) is used when our goal is to **reduce the variance of a decision tree**.
- Here idea is to create several subsets of data from training sample chosen randomly with replacement. Now, each collection of subset data is used to train their decision trees.
- As a result, we end up with an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree.
- **Random Forest** is an extension over bagging.



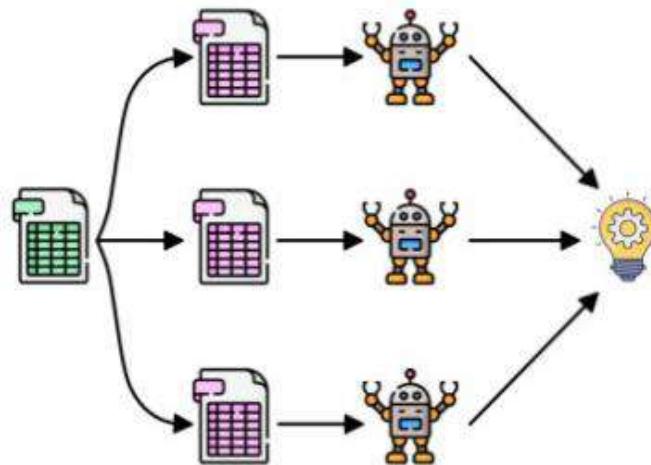
Boosting

- Boosting is another ensemble technique to create a collection of predictors.
- In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors.
- In other words, we fit consecutive trees (random sample) and at every step, the goal is to solve for net error from the prior tree.
- **AdaBoost (Adaptive Boosting), Gradient Boosting and XGBoost (Extreme Gradient Boosting).**
- The key differentiator between boosting-based techniques is the way in which errors are penalized (by modifying weights or minimizing a loss function) as well as how the data is sampled.

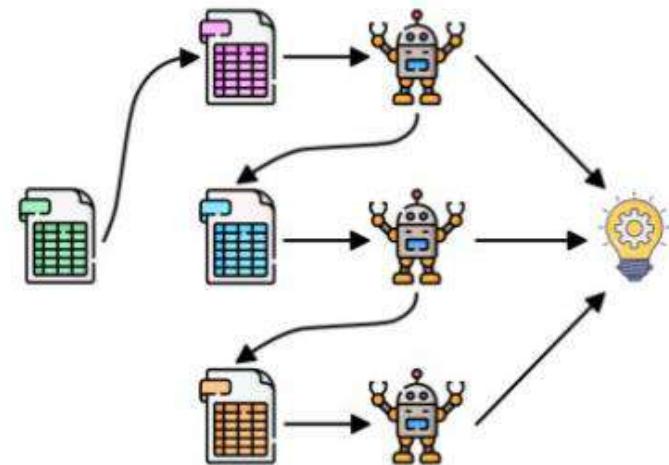


Bagging v/s Boosting

Bagging



Boosting



Parallel

Sequential

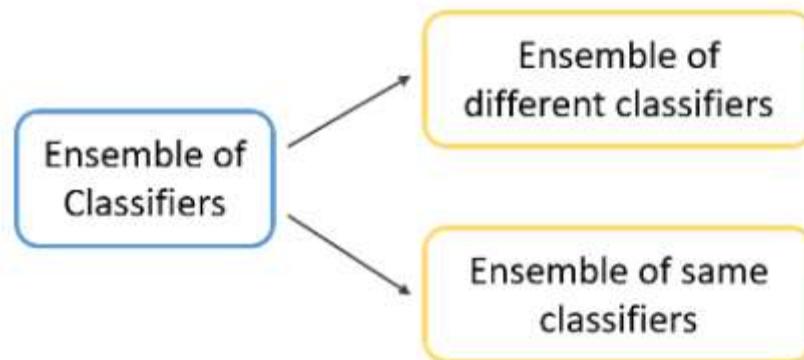
For more details:

<https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422>

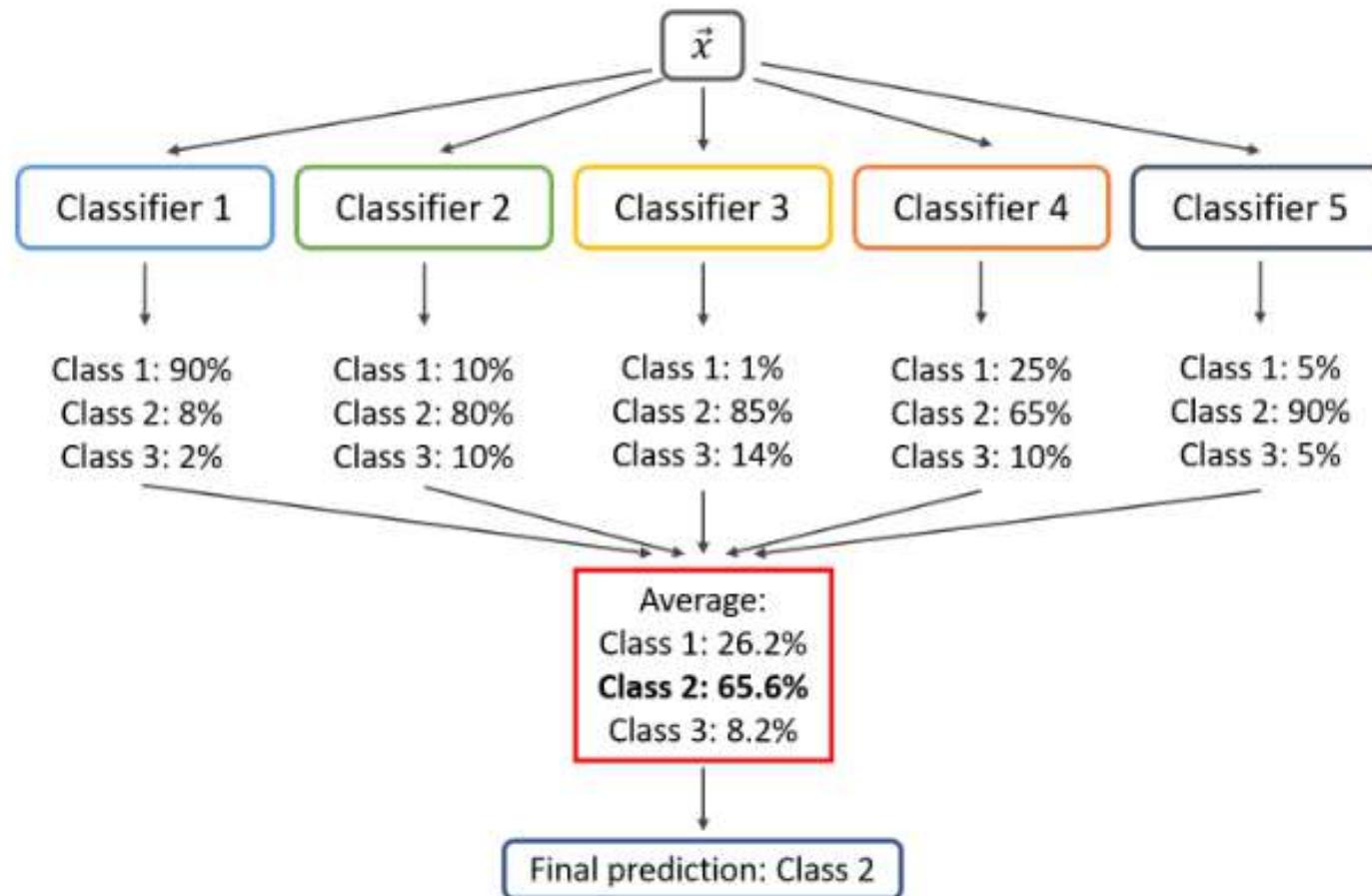
Ensemble Methods

Ensemble Methods

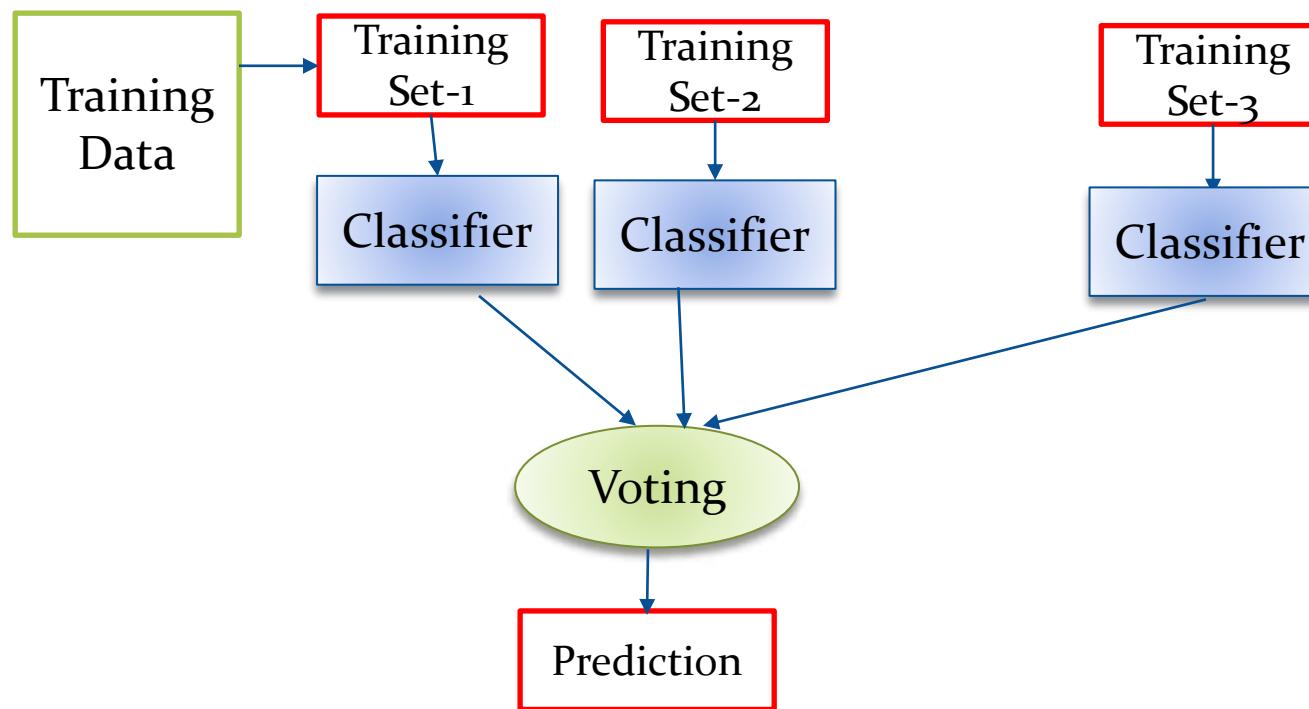
- Ensemble methods are meta-algorithms that combine several machine learning techniques into **one predictive model** in order to improve accuracy.
- Ensemble learning helps improve machine learning results by combining several models.



Ensemble of Different Classifiers



Ensemble of Same Classifiers



Simple Ensemble Techniques

- 1) Max Voting
- 2) Averaging
- 3) Weighted Averaging

Max Voting

- In this technique, multiple models are used to make predictions for each data point.
- The max voting method is generally used for **classification problems**.

	Model-1	Model-2	Model-3	Model-4	Model-5
Rating	5	4	5	4	4

Averaging

- In this method, we take an average of predictions from all the models and use it to make the final prediction.
- Averaging can be used for making predictions in **regression problems** or while calculating probabilities for classification problems.

	Model-1	Model-2	Model-3	Model-4	Model-5	Final Rating
Rating	5	4	5	4	4	4.4

$$(5+4+5+4+4)/5 = 4.4$$

Weighted Average

- This is an extension of the averaging method.
- All models are assigned different weights defining the importance of each model for prediction.

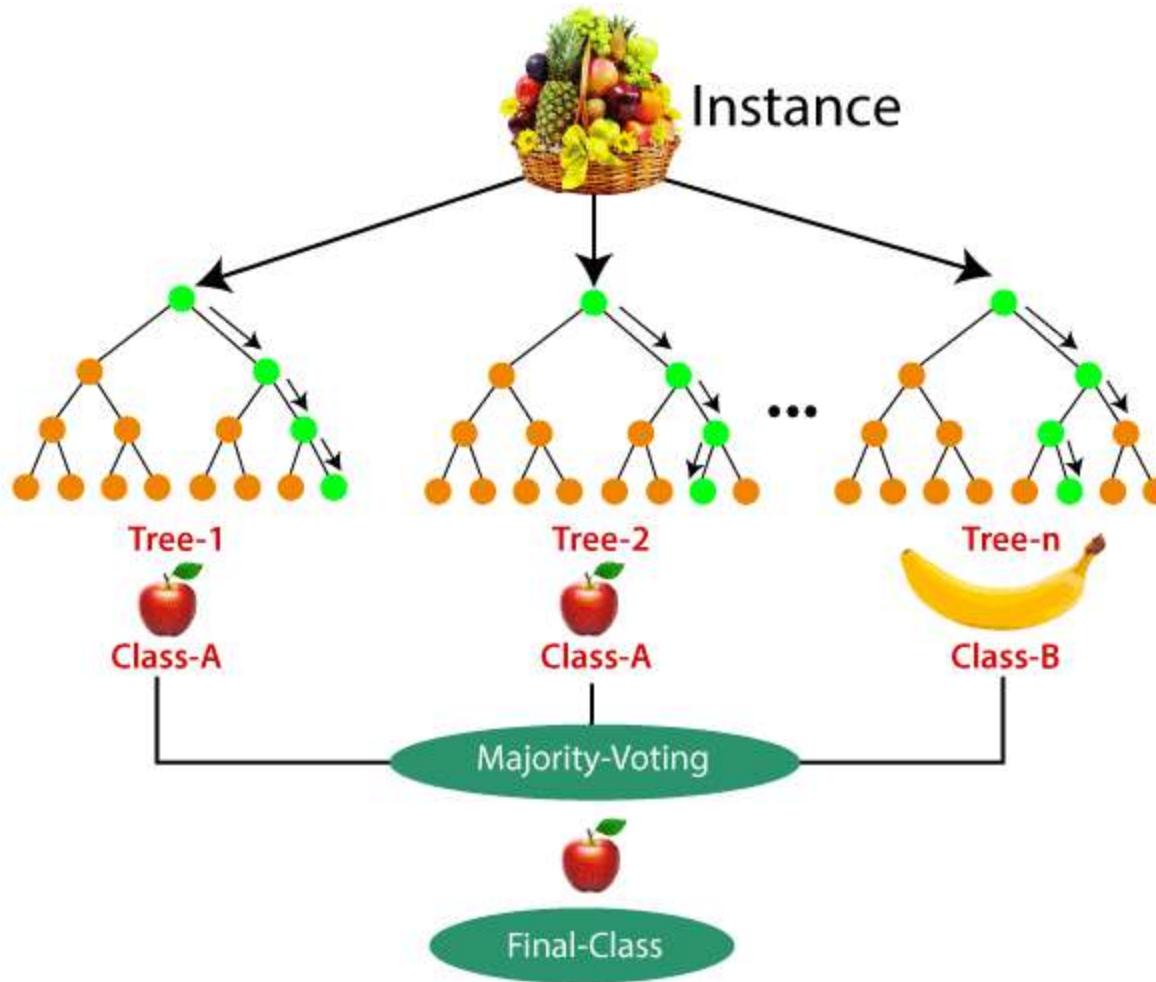
	Model-1	Model-2	Model-3	Model-4	Model-5	Final Rating
Weight	0.23	0.23	0.18	0.18	0.18	
Rating	5	4	5	4	4	4.41

$$[(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.$$

Random Forest Classifier

Random Forest Classifier

- Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems.
- A random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of **voting**.
- It is an **ensemble** method which is better than a single decision tree because it reduces the over-fitting by averaging the result.



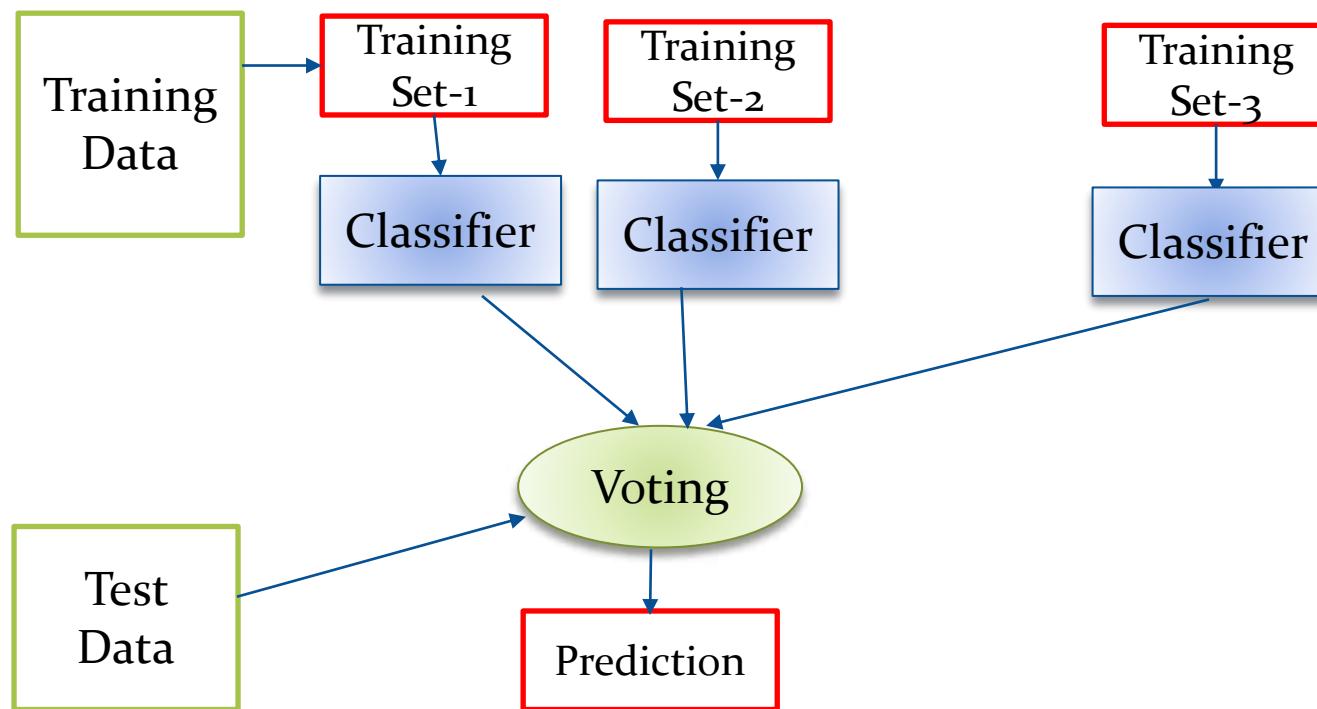
<https://www.javatpoint.com/machine-learning-random-forest-algorithm>

Step 1 – First, start with the selection of random samples from a given dataset.

Step 2 – Construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 – Voting will be performed for every predicted result.

Step 4 – Select the most voted prediction result as the final prediction result.



Advantages of Random Forest Tree

- 1) It overcomes the problem of overfitting by averaging or combining the results of different decision trees.
- 2) Random forests work well for a large range of data items than a single decision tree does.
- 3) Random forests are very flexible and possess very high accuracy.
- 4) Scaling of data does not require in random forest algorithm. It maintains good accuracy even after providing data **without scaling**.
- 5) Random Forest algorithms maintains good accuracy even a **large proportion of the data is missing**.

Disadvantages of Random Forest Tree

- 1) Complexity is the main disadvantage of Random forest algorithms.
- 2) Construction of Random forests are much harder and time-consuming than decision trees.
- 3) More computational resources are required to implement Random Forest algorithm.
- 4) High testing time as compare with other algorithms.

Applications of Random Forest Tree

Banking: Banking sector mostly uses this algorithm for the identification of loan risk.

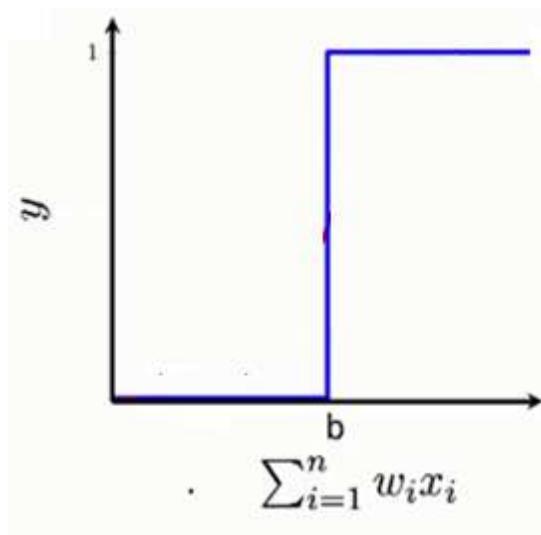
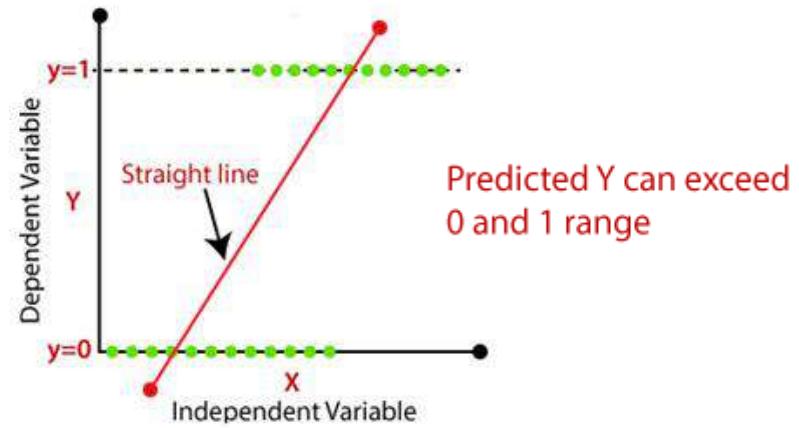
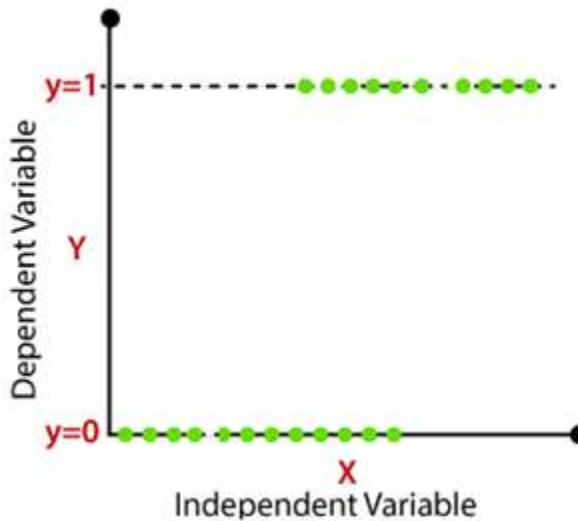
Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.

Land Use: We can identify the areas of similar land use by this algorithm.

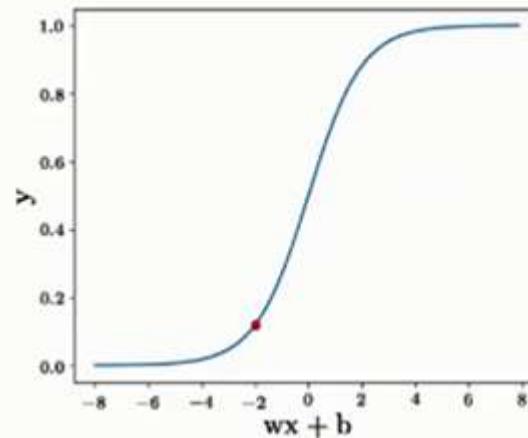
Marketing: Marketing trends can be identified using this algorithm.

Logistic Regression

Why Logistic Regression?



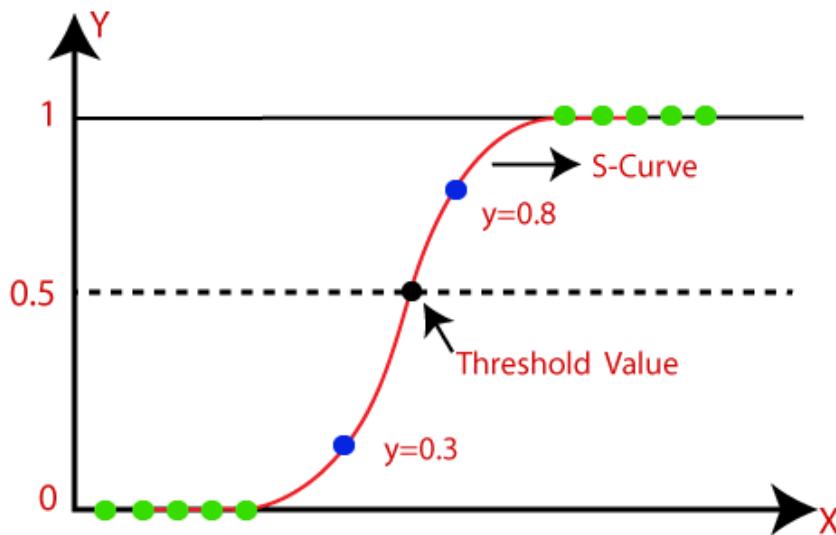
$$y = \frac{1}{1 + e^{-(wx+b)}}$$



Logistic Regression

- Logistic regression is a special case of **linear regression** when the outcome variable is categorical.
- Logistic regression is a **classification model**, used when the value of the target variable is *categorical* in nature.
- Logistic regression is best suited for instances of **binary classification**, it can be applied to multiclass classification problems.
- When using logistic regression, a threshold is usually specified that indicates at what value the example will be put into one class vs. the other class.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

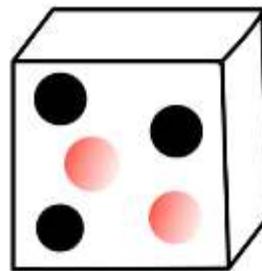


- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It gives the probabilistic values which lie between 0 and 1.

Probability v/s Odds

- The **odds** are defined as the **probability** that the event will occur divided by the **probability** that the event will not occur.

$$\text{Odds} = \frac{P}{1-P}$$



- What is the probability of choosing red ball.

PROBABILITY:

$$\frac{2}{5} = 40\%$$

ODDS:

$$\begin{aligned}\# \text{RED} : \# \text{NON-RED} \\ 2 : 3\end{aligned}$$

In logistic regression, the dependent variable is a ***logit***, which is the natural log of the odds

$$\text{log(odds)} = \text{logit}(Y) = \ln\left(\frac{P}{1-P}\right)$$

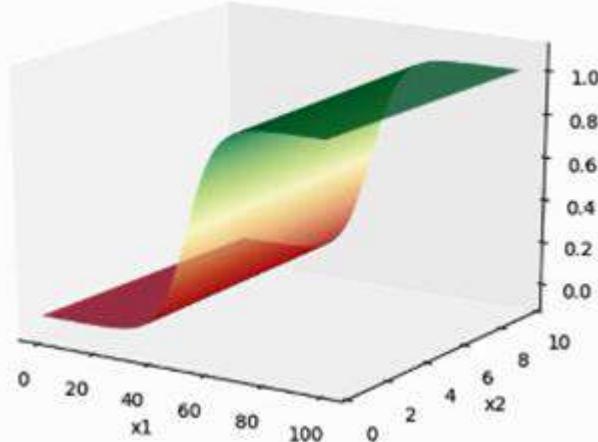
A logit is a log of odds and odds are a function of P, the probability

$$\ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

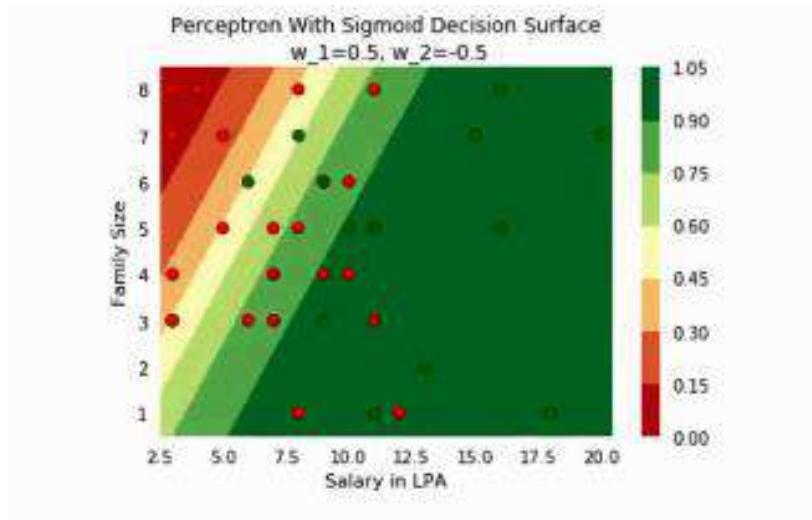
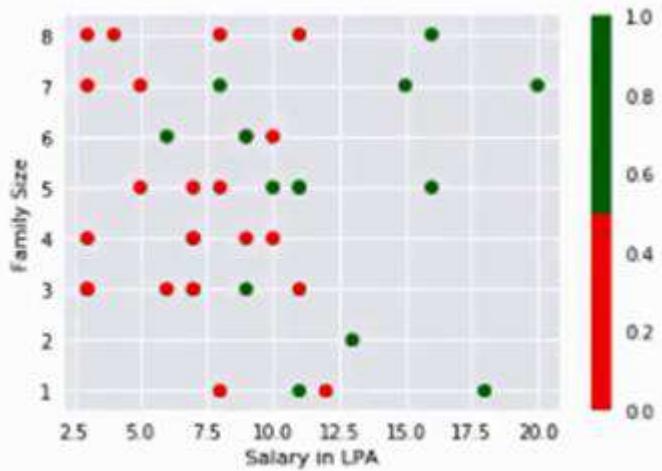
$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

Why Sigmoid Function?



$$y = \frac{1}{1 + e^{-(w_1x_1 + w_2x_2 + b)}}$$

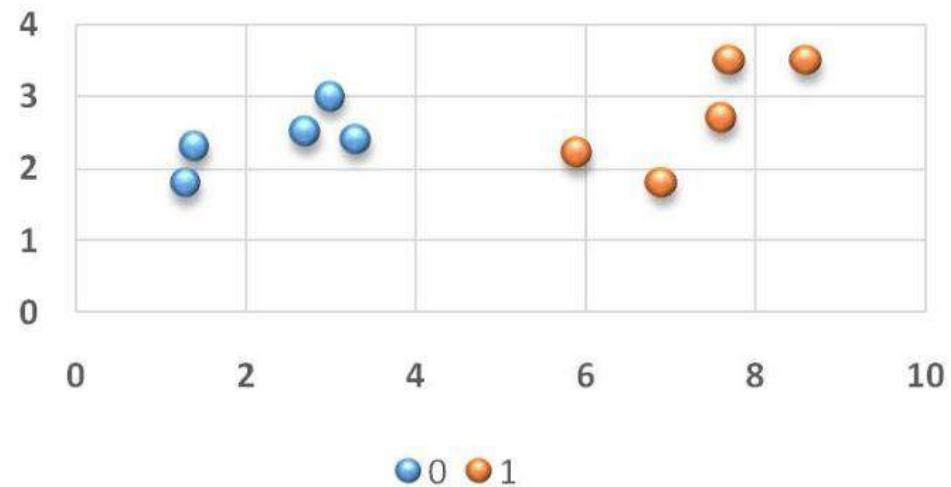
$$y = \frac{1}{1+e^{-(w^T x+b)}}$$



How Logistic Regression Work?

Time (Hr)	Sentences (1= 1000)	Article Type
2.7	2.5	0
1.4	2.3	0
3.3	2.4	0
1.3	1.8	0
3	3	0
7.6	2.7	1
5.9	2.2	1
6.9	1.8	1
8.6	3.5	1
7.7	3.5	1

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$



Some samples of two classes **Technical (1)** and **Non-technical(0)**

$$B_0 = -0.1068913$$

$$Z = B_0 + B_1 * X_1 + B_2 * X_2$$

$$B_1 = 0.41444855$$

$$Z = -0.1068913 + 0.41444855 * \text{Time} - 0.2486209 * \text{Sentences}$$

$$B_2 = -0.2486209$$

For, **X₁ = 1.9** and **X₂ = 3.1**, we get:

$$Z = -0.101818 + 0.41444855 * 1.9 - 0.2486209 * 3.1$$

$$Z = \mathbf{-0.085090545}$$

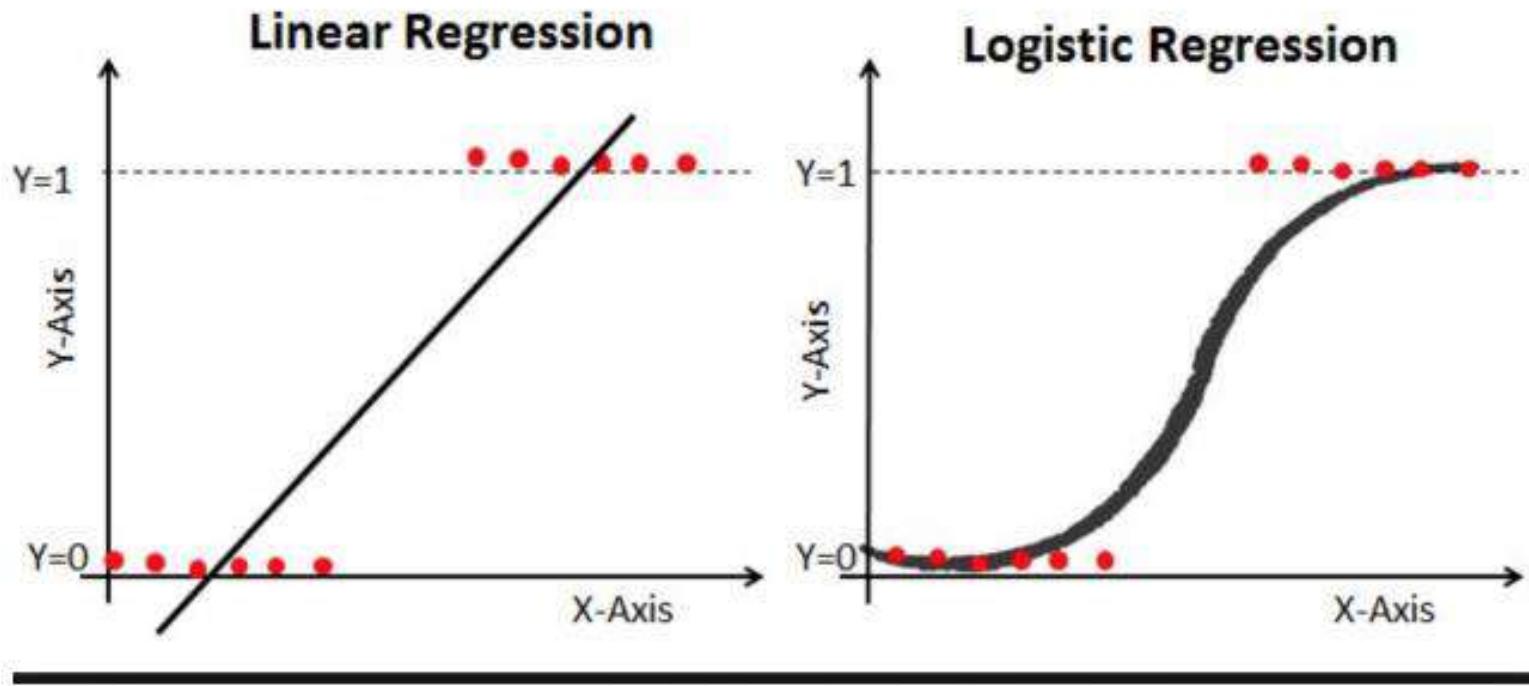
$$y = \frac{1}{1 + e^{-Z}}$$

$$Or, \quad y = \frac{1}{1 + e^{-(0.085090545)}}$$

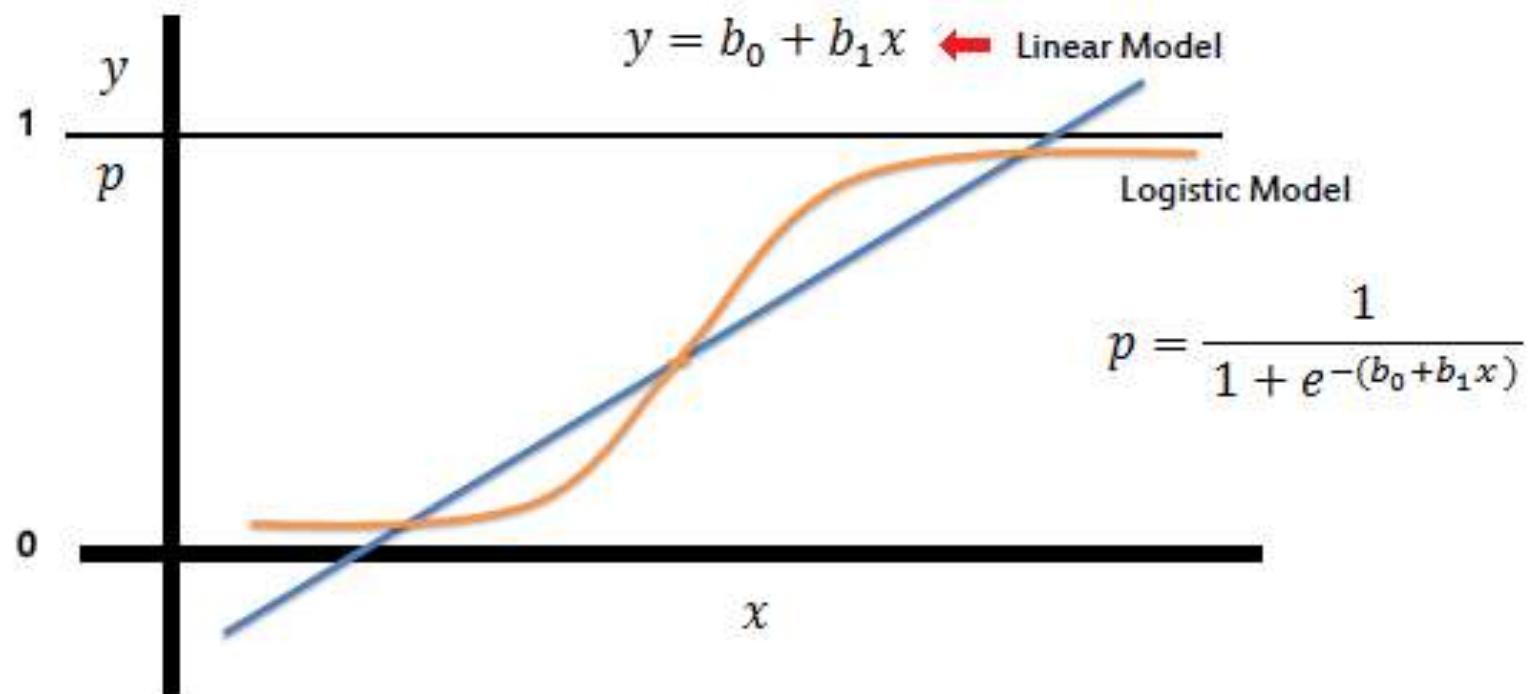
$$Or, \quad y = 0.47747429$$

As **y** is less than 0.5 ($y < 0.5$), we can safely classify given a sample to class **Non-technical**.

Linear Regression vs Logistic Regression

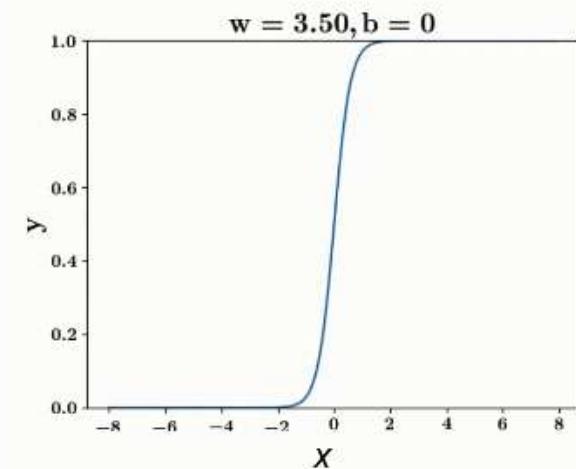
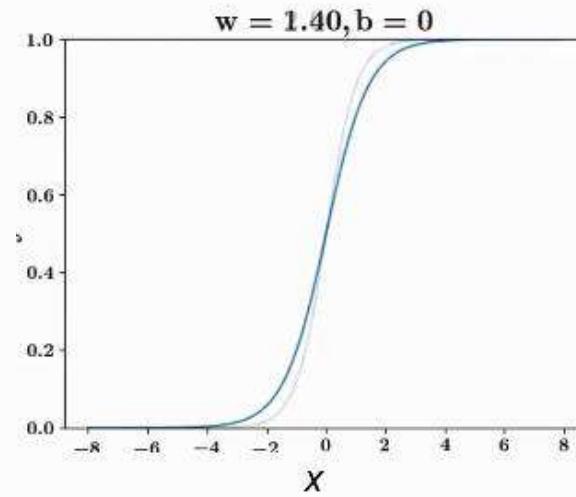
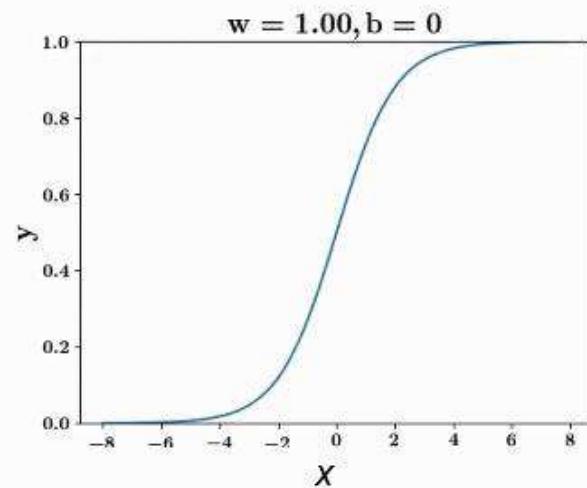
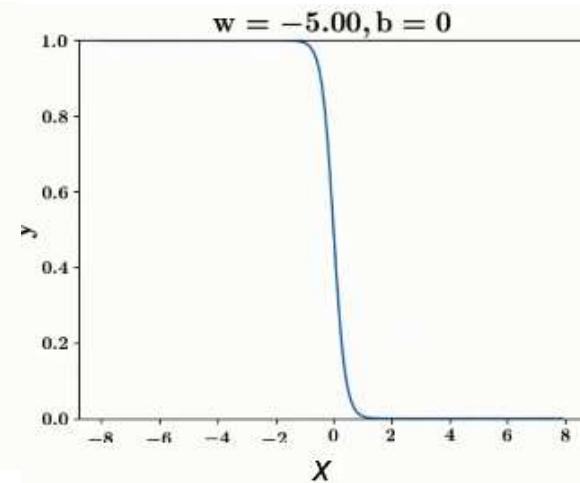
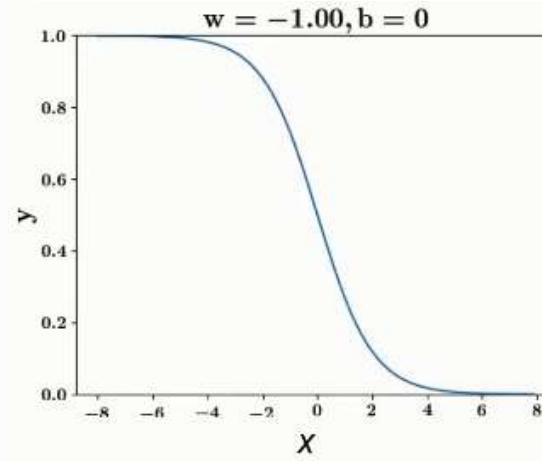
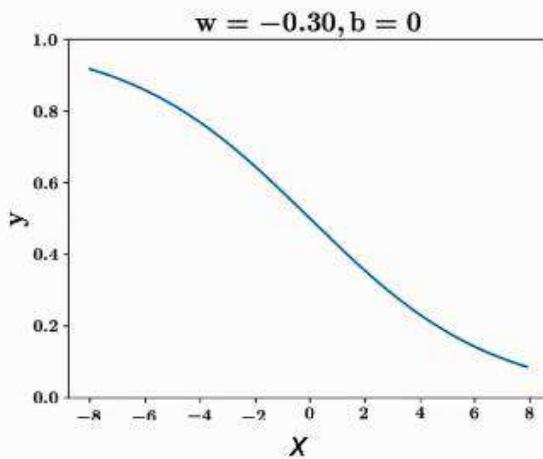


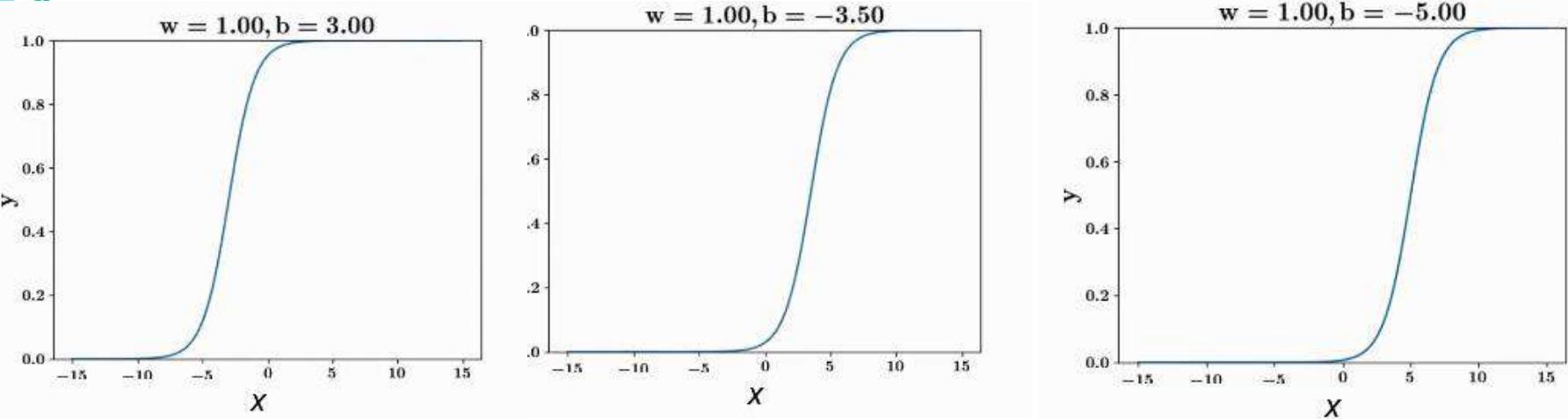
<https://medium.com/@ODSC/logistic-regression-with-python-ede39f8573c7>



Source: https://saedsayad.com/logistic_regression.htm

Function Behaviour when we Changed W and b?





1. w: (controls the slope)
 - a. Negative w, negative slope, mirrored s-shape, becomes more harsh(vertical/less smooth) the more negative it goes
 - b. Positive w, positive slope, normal s-shape, becomes more harsh(vertical/less smooth) the more positive it goes

2. b: (controls the midpoint)

$$y = 1/(1 + \exp(-(wx + b))) = \frac{1}{2} \text{ (for } w=1.00, b = -5\text{)}$$

$$\exp(-(wx + b)) = 1$$

$$wx + b = 0$$

$x = -b/w$ (As b becomes more -ve, boundary moves more to the right +ve, and vice versa)

Logistic Regression Assumptions

- There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
- We must include meaningful variables in our model.
- We should choose a large sample size for logistic regression.

Types of Logistic Regression

1) Binary or Binomial

- In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

2) Multinomial

- In such a kind of classification, dependent variable can have 3 or more possible *unordered* types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

3) Ordinal

- In such a kind of classification, dependent variable can have 3 or more possible *ordered* types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

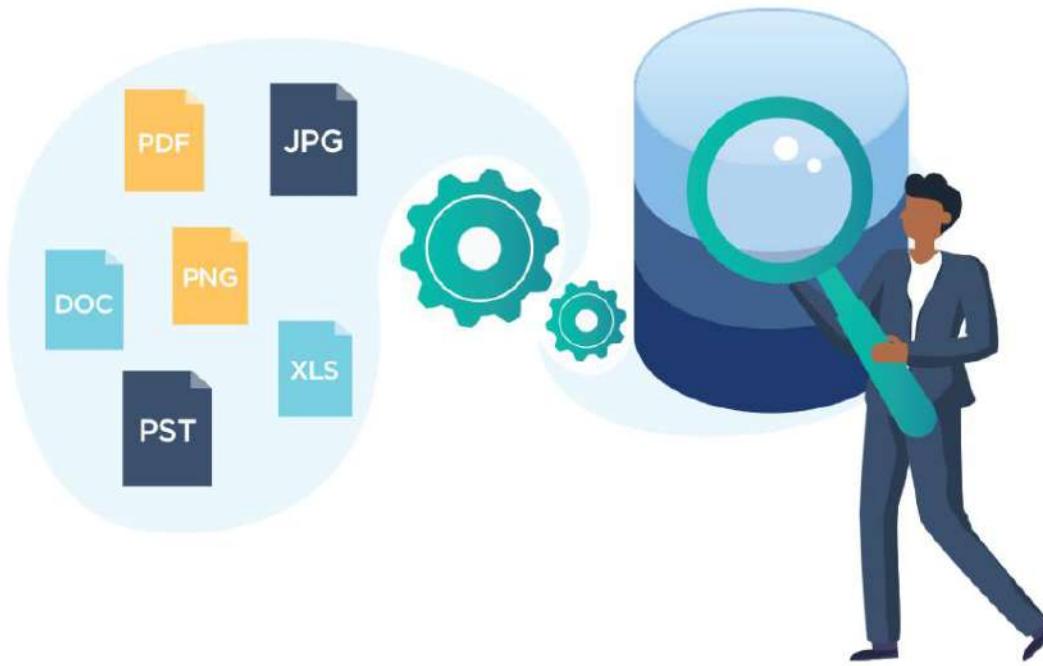
Advantages of Logistic Regression

- 1) Logistic regression is easier to implement, interpret and very efficient to train.
- 2) Logistic Regression performs well when the **dataset is linearly separable**.
- 3) Logistic regression is **less prone to over-fitting** for **low dimensional** datasets.
- 4) Logistic regression is easier to implement, interpret and very efficient to train.

Disadvantages of Logistic Regression

- 1) Main limitation of Logistic Regression is the **assumption of linearity** between the dependent variable and the independent variables.
- 2) Logistic Regression requires moderate or no multicollinearity between independent variables.
- 3) **Non linear problems can't be solved** with logistic regression since it has a **linear decision surface**.
- 4) It is **difficult to capture complex relationships** using logistic regression.
- 5) Logistic Regression **requires a large dataset**

Data Mining (20CP306T)



Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

UNIT- 3 Syllabus

UNIT 3 ASSOCIATION ANALYSIS

- ❖ Problem definition, Frequent item set generation; Rule Generation; Compact representation of frequent item sets; Alternative methods for generating frequent item sets. FP-Growth algorithm, Evaluation of association patterns, Effect of skewed support distribution, Sequential patterns.

Introduction

- **Data mining** is the discovery of knowledge and useful information from the large amounts of data stored in databases.
- **Association Rules:** Association rule learning is a type of unsupervised learning technique that checks for the **dependency of one data item on another data item** and **maps accordingly**.
- It tries to find **some interesting relations or associations among** the variables of dataset.
- It is employed in **Market Basket analysis, Web usage mining, continuous production, etc.** Here market basket analysis is a technique used by the various big retailer to discover the associations between items.
 - Ex. In a supermarket, all products that are purchased together are put together.

- For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby.



Association rule learning can be divided into three types of algorithms

- 1.Apriori**
- 2.Eclat**
- 3.F-P Growth Algorithm**

Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection;
 - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Rules Discovered:

$\{\text{Milk}\} \rightarrow \{\text{Coke}\}$

$\{\text{Diaper, Milk}\} \rightarrow \{\text{Beer}\}$

How does Association Rule Learning work?

- Association rule learning works on the concept of **If** and **Else** Statement, such as if A then B.



- Here the If element is called **antecedent**, and then statement is called as **Consequent**.
- These types of relationships where we can find out some association or relation between two items is known as *single cardinality*. It is all about creating rules, and if the number of items increases, then **cardinality also increases accordingly**.
- So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

- 1) **Support**
- 2) **Confidence**
- 3) **Lift**

Support

- ❖ Let N is the total number of transactions.
- Support of X is represented as the number of times X appears in the database divided by N
$$\text{Support}(X) = (\text{Number of times } X \text{ appears}) / N = P(X)$$
- Support for X and Y together is represented as the number of times they appear together divided by N as given below.
$$\text{Support}(XY) = (\text{Number of times } X \text{ and } Y \text{ appear together}) / N = P(X \cap Y)$$
- Thus, Support of X is the probability of X while the support of XY is the probability of $X \cap Y$

Sale database

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

$\text{Support}(\text{Bread}) = \text{Number of times Bread appears} / \text{total number of translations} = 4/5 = P(\text{Bread})$

$\text{Support}(\text{Milk}) = \text{Number of times Milk appears} / \text{total number of translations} = 4/5 = P(\text{Milk})$

$\text{Support}(\text{Diapers}) = \text{Number of times Diapers appears} / \text{total number of translations} = 4/5 = P(\text{Diapers})$

$\text{Support}(\text{Beer}) = \text{Number of times Beer appears} / \text{total number of translations} = 3/5 = P(\text{Beer})$

$\text{Support}(\text{Eggs}) = \text{Number of times Eggs appears} / \text{total number of translations} = 1/5 = P(\text{Eggs})$

$\text{Support}(\text{Cola}) = \text{Number of times Cola appears} / \text{total number of translations} = 2/5 = P(\text{Cola})$

Sale database

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

$\text{Support}(\text{Bread, Milk}) = \text{Number of times Bread, Milk appear together} / \text{total number of translations} = 3/5 = P(\text{Bread} \cap \text{Milk})$

$\text{Support}(\text{Diapers, Beer}) = \text{Number of times Diapers, Beer appears together} / \text{total number of translations} = 3/5 = P(\text{Diapers} \cap \text{Beer})$

- ❖ Support is very important metric because if a rule has **low support** then it may be the case that the **rule occurs by chance and it will not be logical to promote items** that customers seldom buy together. But if a rule has **high support** then that association becomes very important and if implemented properly **will result in increase in revenue, efficiency and customer satisfaction.**

Confidence

- To understand the concept of confidence, let us suppose that support for $X \rightarrow Y$ is 80%, then it means that $X \rightarrow Y$ is very frequent and there are 80% chances that X and Y will appear together in a transaction. This would be of interest to the sales manager.
- Let us suppose we have another pairs of items (A and B) and support for $A \rightarrow B$ is 50%. Of course it is not as frequent as $X \rightarrow Y$, but if this was higher, such as whenever A appears there is 90% chance that B also appears, then of course it would be of great interest.
- Thus, not only the probability that A and B appear together matters, but also the conditional probability of B when A has already occurred plays a significant role.
- This **conditional probability that B will follow when A has already been occurred** is considered during determining the confidence of the rule.

- Confidence for $X \rightarrow Y$ is defined as the **ratio of the support for X and Y together to the support for X.**

Confidence of $(X \rightarrow Y) = \text{Support}(XY) / \text{Support}(X) = P(X \cap Y) / P(X) = P(Y|X)$
where $P(Y|X)$ is the probability of Y once X has taken place

- Therefore if X appears much more frequently than X and Y appearing together, the confidence will be low.

Example of the support measure

TID	Items
1	ABC
2	ABD
3	BC
4	AC
5	BCD

TID	Items	Support = Occurrence / Total Support
1	ABC	
2	ABD	Total Support = 5
3	BC	Support {AB} = 2/5 = 40%
4	AC	Support {BC} = 3/5 = 60%
5	BCD	Support {ABC} = 1/5 = 20%

Example of the confidence measure

TID	Items	Given X \Rightarrow Y Confidence = Occurrence {X and Y} / Occurrence of (X)
1	ABC	
2	ABD	Confidence {A \Rightarrow B} = 2/3 = 66%
3	BC	Confidence {B \Rightarrow C} = 3/4 = 75%
4	AC	Confidence {AB \Rightarrow C} = 1/2 = 50%
5	BCD	

Database for identification of association rules

Antecedent	Consequent
A	0
A	0
A	1
A	0
B	1
B	0
B	1

There are two rules derived from the association of these combinations:

Rule 1: A implies 0, i.e., $A \rightarrow 0$

Rule 2: B implies 1, i.e., $B \rightarrow 1$

The support for Rule 1 is the Number of times A and 0 appear together / Total Number of transactions.

Support of Rule 1 = 3/7

The Support for Rule 2 is the Number of times B and 1 appear together / Total Number of transactions.

Support of Rule 2 = 2/7

- The Confidence for Rule 1 is Support of (A,0) / Support A, i.e., (Number of times A and 0 appear together / Total number of items) DIVIDED BY (Number of times A appears / Total number of items) Here, total number of items get cancelled. **Thus, the Confidence for Rule 1 is the Number of times A and 0 appear together / Number of times A appears.**

Confidence for Rule 1= 3/4

Confidence for Rule 2= 2/3

Database for identification of association rules

Antecedent	Consequent
A	0
A	0
A	1
A	0
B	1
B	0
B	1

Lift

- Let us suppose that Coke is a very common sales item in a store and that it usually appears in most of the transactions.
- Let us suppose that we have a rule of **Candle→Coke** which has a support of 20% and has a confidence of 90%. It is very logical to think that if coke is very popular and it appears in 95% of transactions, then obviously it also appears quite often with the candle as well. So, the rule for association of **candle and coke will not be all that useful**.
- But if we find that **Candle→ Matchbox** also has a support of 20% and a confidence of 90% then it is logical to suppose that the frequency of matchbox sales is very little as compared to the sale of coke. And the rule suggests that when we make a sale of candles, 90% chance indicates that a matchbox will also be sold in the same transaction. It is more effective and logical to conclude that when we sell a candle then we also sell a coke (coke is popular and will appear with every item not just with candle). **As support and confidence are unable to handle this case.**
- It is handled by the lift of the rule. In this case, the probability of Y is very low in case of **Candle→ Matchbox** (Here, Y is matchbox) and will be very high in case of **Candle→Coke** (Here, Y is coke). One can note that the low probability of Y, makes the **X→Y** rule more effective as compared to high probability of Y. Lift takes the note of this

Lift

- Confidence of the rule does not depend on the frequency of ‘Y’ appearing. But in order to identify the strength of the rule it is important to consider the **frequency of X and Y**.
- The Lift of the rule however considers the **whole dataset**, by taking into account the probability of Y also, in deciding the strength of the rule.
- Lift is the ratio of Confidence of $X \rightarrow Y$ divided by the probability of Y.

$$\text{Lift} = P(Y|X) / P(Y)$$

Or

$$\text{Lift} = \text{Confidence of } (X \rightarrow Y) / P(Y)$$

Or

$$\text{Lift} = (P(X \cap Y) / P(X)) / P(Y)$$

Rule 1: A implies 0, i.e., $A \rightarrow 0$

Rule 2: B implies 1, i.e., $B \rightarrow 1$

Lift for Rule1, i.e., $A \rightarrow 0 = P(0 | A) / P(0) = (P(A \cap 0) / P(A)) / P(0) = (3/4) / (4/7) = 1.3125$

Lift for Rule2, i.e., $B \rightarrow 1 = P(1 | B) / P(1) = (P(B \cap 1) / P(B)) / P(1) = (2/3) / (3/7) = 1.55$

Antecedent	Consequent
A	0
A	0
A	1
A	0
B	1
B	0
B	1

- As discussed earlier, the confidence for **Rule 1 is 3/4=0.75** and confidence for **Rule 2 is 2/3= 0.66**.
- It should be observed that although the Rule 1 has higher confidence as compared to Rule 2, but it has lower lift as compared to Rule 2.
- Sometime, confidence of the rule can be misleading if it is independent of the dataset. Lift, as a metric is important because **it considers both the confidence of the rule and the overall dataset.**

Rule 1: A implies 0, i.e., $A \rightarrow 0$

Rule 2: B implies 1, i.e., $B \rightarrow 1$

Support of $A \rightarrow 0 = 3/12$

Confidence of $A \rightarrow 0 = \text{Support}(A, 0) / \text{Support}(A) = 3/4 = 0.75$

Lift = Confidence ($A, 0$) / Support(0) = $(3/4) / (8/12) = 36/32 = 1.125$

For rule 2

Support of $B \rightarrow 1 = 2/12$

Confidence of $B \rightarrow 1 = \text{Support}(B, 1) / \text{Support}(B) = 2/3 = 0.66$

Lift = Confidence / Support(1) = $(2/3) / (4/12) = 2$

Antecedent	Consequent
A	0
A	0
A	1
A	0
B	1
B	0
B	1
C	0
D	0
C	0
C	1
E	0

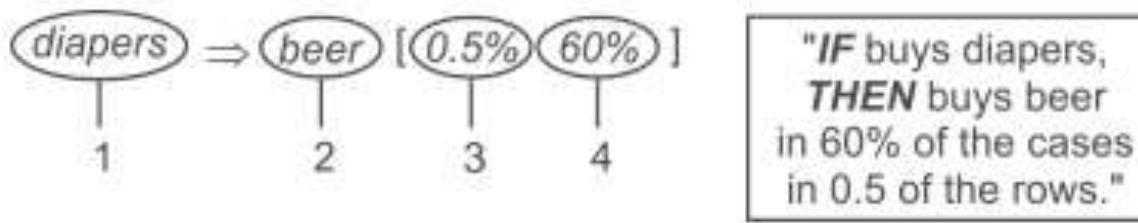
Calculate support, confidence and lift for the following rules.



Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

Another Representation of Association Rules

- Sometimes, the representation of association rules also includes support and confidence as shown



Representation of association rules

$\text{diapers} \Rightarrow \text{beer} [0.5\%, 60\%]$

Here,

1. Antecedent, left-hand side (LHS), body
2. Consequent, right-hand side (RHS), head
3. Support, frequency
4. Confidence, strength

Approaches for Transaction Database Storage

- There are three ways to store datasets of transactions.
 - 1) Simple Storage
 - 2) Horizontal Storage
 - 3) Vertical Storage

Let us suppose the number of items be five; $\{I_1, I_2, I_3, I_4, I_5\}$.

Let there be only seven transactions with transaction IDs $\{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$.

1) Simple transaction storage

- Consist of two columns
- Each row of the table shows the transaction ID and the purchased items

A simple representation of transactions as an item list

<i>Transaction ID</i>	<i>Items</i>
T1	I1, I2, I4
T2	I4, I5
T3	I1, I3
T4	I2, I4, I5
T5	I4, I5
T6	I2, I3, I5
T7	I1, I3, I4

2) Horizontal storage

- In this representation, each row is still a transaction, but columns have been created for each item.
- In the cell, 1 is filled against the item that occurs in a transaction and 0 against the rest.

Horizontal storage representation

TID	I1	I2	I3	I4	I5
T1	1	1	0	1	0
T2	0	0	0	1	1
T3	1	0	1	0	0
T4	0	1	0	1	1
T5	0	0	0	1	1
T6	0	1	1	0	1
T7	1	0	1	1	0

Horizontal storage representation

TID	I1	I2	I3	I4	I5
T1	1	1	0	1	0
T2	0	0	0	1	1
T3	1	0	1	0	0
T4	0	1	0	1	1
T5	0	0	0	1	1
T6	0	1	1	0	1
T7	1	0	1	1	0

- The advantage of this storage system is that we can count the frequency of each item by counting the ‘1’s in the given column.
 - Frequency of item I1 as 3 and item I4 as 5.
- We can also easily calculate the frequency of item pairs by counting the 1’s that result after an ‘AND’ operation on corresponding columns.
 - For example, the frequency of item pairs **I1 and I2 is 1** (By AND operation on column of I1 and I2 we will get only one 1, i.e., T1).

3) Vertical representation

- In this representation, the transaction list is turned around.
- Each row now represents an item and it indicates transactions in which the item appears. The columns now represent the transactions.
- This representation is also called a TID-list since for each item it provides a list of TIDs (Transition Ids).

Vertical storage representation

Item	TID						
	T1	T2	T3	T4	T5	T6	T7
I1	1	0	1	0	0	0	1
I2	1	0	0	1	0	1	0
I3	0	0	1	0	0	1	1
I4	1	1	0	1	1	0	1
I5	0	1	0	1	1	1	0

- A vertical representation also facilitates counting of items by counting the number of 1s in each row.
- In case of 2-itemsets, where you want to find out the frequency of occurrences of 2 items together in a transaction, you have to refer to the intersection of 2 rows corresponding to the items, and inspect one column at a time.
 - Example: If you want to find out frequency of item pairs (I1,I2) just look at rows I1 and I2 for each column if values of both I1 and I2 is 1 then this means I1 and I2 are occurring together in a transaction hence write their count as 1. similarly the count of item pair {I1, I4} is 2.

	T1	T2	T3	T4	T5	T6	T7
I1, I2	1	0	0	0	0	0	0
I1, I4	1	0	0	0	0	0	1

- The vertical representation is not storage efficient in case of very large number of transactions.

Naïve Algorithm for Finding Association Rules

- 1) Step-1: Find frequency of each items and all possible item pairs.
- 2) Step- 2: Identify item pairs that satisfy the threshold value of **support**.
- 3) Step- 3: Generate and identify rules that qualify threshold value of **confidence**.

For the given dataset find the association rules having minimum ‘support’ of 50% and minimum ‘confidence’ of 75%.

Sale record of grocery store

Transaction ID	Items
100	Bread, Cornflakes
101	Bread, Cornflakes, Jam
102	Bread, Milk
103	Cornflakes, Jam, Milk

- First step of the naïve algorithm is to list all the combinations of the items that are in stock and then identify the frequent combinations or combinations having frequency more than or equal to a specified support limit.
- Using this approach, the association rules that have the '**confidence**' **more than the threshold limit are identified.**

List of all itemsets and their frequencies

Itemsets	Frequency
Bread	3
Cornflakes	3
Jam	2
Milk	2
(Bread, Cornflakes)	2
(Bread, Jam)	1
(Bread, Milk)	1
(Cornflakes, Jam)	2

Itemsets	Frequency
(Cornflakes, Milk)	1
(Jam, Milk)	1
(Bread, Cornflakes, Jam)	1
(Bread, Cornflakes, Milk)	0
(Bread, Jam, Milk)	0
(Cornflakes, Jam, Milk)	1
(Bread, Cornflakes, Jam, Milk)	0

- Here, the minimum required support is 50% and we have to identify the frequent itemsets that appear **in at least two transactions**. The list of frequencies shows that all four items Bread, Cornflakes, Jam and Milk are frequent.

List of all itemsets and their frequencies

Itemsets	Frequency
Bread	3
Cornflakes	3
Jam	2
Milk	2
(Bread, Cornflakes)	2
(Bread, Jam)	1
(Bread, Milk)	1
(Cornflakes, Jam)	2
(Cornflakes, Milk)	1
(Jam, Milk)	1
(Bread, Cornflakes, Jam)	1
(Bread, Cornflakes, Milk)	0
(Bread, Jam, Milk)	0
(Cornflakes, Jam, Milk)	1
(Bread, Cornflakes, Jam, Milk)	0

The set of all frequent items

Itemsets	Frequency
Bread	3
Cornflakes	3
Jam	2
Milk	2
(Bread, Cornflakes)	2
(Cornflakes, Jam)	2

- We are interested in association rules that can only occur with **item pairs**, thus **individual frequent items** Bread, Cornflakes, Jam and Milk **are ignored**, and item pairs (**Bread, Cornflakes**) and (**Cornflakes, Jam**) are considered for association rule mining.
- The two 2-itemsets (**Bread, Cornflakes**) and (**Cornflakes, Jam**) are determined with a required confidence of 75%.
- Every 2-itemset (A, B) can lead to two rules $A \rightarrow B$ and $B \rightarrow A$, and we will consider rule which satisfy the required confidence.

Bread→**Cornflakes**

Confidence = $2/3 = 67\%$

Cornflakes→**Bread**

Confidence = $2/3 = 67\%$

Cornflakes→**Jam**

Confidence = Support of (Cornflakes, Jam) /
Support of (Cornflakes) = $2/3 = 67\%$

Jam→**Cornflakes**

Confidence = Support of (Jam, Cornflakes) /
Support of (Jam) = $2/2 = 100\%$

Rule **Jam**→**Cornflakes** has more than the minimum required confidence.

Limitation of Naïve

- It can we used when we have a small number of items in the database. Not suitable for the larger data items.
- The number of combinations becomes about a million with 20 items since the number of combinations is 2^n with n items.

Improved Naïve Algorithm to Deal with Larger Datasets

- Rather than counting all the possible item combinations in the stock it will be better to **focus only on the items that are sold in transactions**. Because, we may have hundreds of items in stock but we are concerned with finding associations only for the items that are being sold together.

Sale record of grocery store

Transaction ID	Items
100	Bread, Cornflakes
101	Bread, Cornflakes, Jam
102	Bread, Milk
103	Cornflakes, Jam, Milk

All possible combinations with nonzero frequencies

Transaction ID	Items	Combinations
100	Bread, Cornflakes	(Bread, Cornflakes)
200	Bread, Cornflakes, Jam	(Bread, Cornflakes), (Bread, Jam), (Cornflakes, Jam), (Bread, Cornflakes, Jam)
300	Bread, Milk	(Bread, Milk)
400	Cornflakes, Jam, Milk	(Cornflakes, Jam), (Cornflakes, Milk), (Jam, Milk), (Cornflakes, Jam, Milk)

Frequencies of all itemsets with nonzero frequencies

Itemsets	Frequency
Bread	3
Cornflakes	3
Jam	2
Milk	2
(Bread, Cornflakes)	2
(Bread, Jam)	1
(Cornflakes, Jam)	2
(Bread, Cornflakes, Jam)	1
(Bread, Milk)	1
(Cornflakes, Milk)	1
(Jam, Milk)	1
(Cornflakes, Jam, Milk)	1

With minimum required support is 50%

The set of all frequent items

Itemsets	Frequency
Bread	3
Cornflakes	3
Jam	2
Milk	2
(Bread, Cornflakes)	2
(Cornflakes, Jam)	2

- Ignore all single with **item pairs**.
- The two 2-itemsets (Bread, Cornflakes) and (Cornflakes, Jam) are determined with a required confidence of 75%.
- Every 2-itemset (A, B) can lead to two rules $A \rightarrow B$ and $B \rightarrow A$, and we will consider rule which satisfy the required confidence.

Bread→Cornflakes

Confidence = $2/3 = 67\%$

Cornflakes→Bread

Confidence = $2/3 = 67\%$

Cornflakes→Jam

Confidence = Support of (Cornflakes, Jam) /
Support of (Cornflakes) = $2/3 = 67\%$

Jam→Cornflakes

Confidence = Support of (Jam, Cornflakes) /
Support of (Jam) = $2/2 = 100\%$

Rule **Jam→Cornflakes** has more than the minimum required confidence.

- ❖ Still, naïve algorithm or any improvement of it is **not suitable for large problems**

Apriori Algorithm

- The Apriori algorithm was developed by two Indians Rakesh Agrawal and Ramakrishnan Srikant in 1994, to mine frequent itemsets for identifying association rules.
- The Apriori algorithm has been named so on the basis that it uses prior knowledge of frequent itemset properties.
- This algorithm consists of two phases.
 - 1) In the first phase, the frequent itemsets, i.e., the itemsets that exceed the minimum required support are identified.
 - 2) In the second phase, the association rules meeting the minimum required confidence are identified from the frequent itemsets.
- The second phase is comparatively straight forward – therefore, the major focus of research in this field is to improve the first phase.

- Identify association rules with 50% support and 75% confidence with the Apriori algorithm. The transactions are given in Table

Transactions database	
Transaction ID	Items
T1	Bread, Cornflakes, Eggs, Jam
T2	Bread, Cornflakes, Jam
T3	Bread, Milk, Tea
T4	Bread, Jam, Milk
T5	Cornflakes, Jam, Milk

- For 50% support each frequent item must appear in at least three transactions.

Phase 1: Identification of frequent itemsets

- It starts with the identification of candidate 1 itemsets, represented by C1 (one itemsets that may be frequent) and it is always the items in the stock which the store deals with, Here, we have six items, so C1 will be as shown below
 - From Candidate 1 itemsets, frequent one-itemsets are represented by L1 and found by calculating the count of each candidate item and selecting only those counts which are equal to or more than the threshold limit of support, i.e., 3

Candidate one itemsets C1

Item
Bread
Cornflakes
Eggs
Milk
Jam
Tea

Frequent items L1

Item	Frequency
Bread	4
Cornflakes	3
Milk	3
Jam	4

- To create C2 the following steps are applied:
 - 1) Perform Cartesian product of L1 with itself
 - 2) Some item pairs may have identical items. Keep just one. Remove the extras.
 - 3) Select only those item pairs in which items are in lexical order (so that if we have Milk, Jam then Jam, Milk should not appear).
- Then, the candidate 2-itemsets or C2 are determined per the process illustrated below:

$$C2 = L1 \text{ JOIN } L1$$

The diagram illustrates the Cartesian product of L1 with itself to form C2. It shows two tables of items, separated by a JOIN operator.

Bread
Cornflakes
Milk
Jam

JOIN

Bread
Cornflakes
Jam
Milk

In the Join operation the first step will be to perform a Cartesian product, i.e., to make all possible pairs between L1 and L1.

- 1) Bread, Bread
- 2) Bread, Cornflakes
- 3) Bread, Milk
- 4) Bread, Jam
- 5) Cornflakes, Bread
- 6) Cornflakes, Cornflakes
- 7) Cornflakes, Milk
- 8) Cornflakes, Jam
- 9) Milk, Bread
- 10) Milk, Cornflakes
- 11) Milk, Milk
- 12) Milk, Jam
- 13) Jam, Bread
- 14) Jam, Cornflakes
- 15) Jam, Milk
- 16) Jam, Jam

After removing items, the final C2 will be
Bread, Cornflakes
Bread, Milk
Bread, Jam
Cornflakes, Milk
Cornflakes, Jam
Milk, Jam

Candidate item pairs C2

Item pairs	Frequency
(Bread, Cornflakes)	2
(Bread, Milk)	2
(Bread, Jam)	3
(Cornflakes, Milk)	1
(Cornflakes, Jam)	3
(Milk, Jam)	2

- We therefore have only two frequent item pairs in L2.

Frequent two item pairs L2

Item pairs	Frequency
(Bread, Jam)	3
(Cornflakes, Jam)	3

- Generation of C3 from L2

- C3 is generated from L2 by carrying out a JOIN operation over L2

$$C3 = L2 \text{ JOIN } L2$$

- It will involve the same steps as performed for C2, but it has one important prerequisite for Join, i.e., two items are joinable **if their first item is common**.
- From {Bread, Jam} and {Cornflakes, Jam} two frequent 2-itemsets, we do not obtain a candidate 3-itemset since we do not have two 2-itemsets that have the same first item.
- This completes the first phase of the Apriori algorithm

Phase 2: Generation of rules

The two frequent 2-itemsets, following possible rules.

Bread→Jam

Jam→Bread

Cornflakes→Jam

Jam→Cornflakes

Confidence for each rule

Bread→Jam

$3/4 = 75\%$

Jam→Bread

$3/4 = 75\%$

Cornflakes→Jam

$3/3 = 100\%$,

Jam→Cornflakes

$3/4 = 75\%$

❖ Note

- One important pre-requisite for Join, i.e., two items are joinable if their first item is common.
- In a generalized case:

$$C_k = L_{k-1} \text{ JOIN } L_{k-1}$$

- And they are joinable if their **first k-2 items** are the same. So, in case of C3, the first item should be the same in L2, while in case of C2 there is no requirement of first item similarity because k-2 in the C2 case is 0.

- ❖ Identify rules for the given transaction database by using apriori algorithm. The threshold value of support is 50% and confidence is 70%.

Transaction database	
T1	1,3,4
T2	2,3,5
T3	1,2,3,5
T4	2,5

Step 1

Create C₁ candidate 1-itemsets. It is all the items in the transaction database

C ₁	Count
1	2
2	3
3	3
4	1
5	3

C₁, candidate 1-itemset and their count

Step 2

Create frequent 1-itemset, i.e., list of 1-itemsets whose frequency is more than the threshold value of support, i.e., 2

C ₁	Count	L ₁
1	2	1
2	3	2
3	3	3
4	1	5
5	3	

L₁, frequent 1-itemset

Candidate two itemsets, C2 with its frequency count as given in

Generation of C2

C2	Count
1, 2	1
1, 3	2
1, 5	1
2, 3	2
2, 5	3
3, 5	2

L2 is created by selecting those candidate pairs having support of 2 or more

L2 is created by selecting those candidate pairs having support of 2 or more

Generation L2

L2
1, 3
2, 3
2, 5
3, 5

From the given L2, the next step will be to generate C3 by using L2 JOIN L2
Thus, C3 :

Generation of C3	
C3	
	2, 3, 5

- The frequency of candidate three itemset 2, 3, 5 is 2, so C3 is qualified as L3. Thus, frequent three itemset is 2, 3, 5 for given dataset.
- This is the final frequent item list and it completes the first phase of the Aproiri algorithm to find the frequent itemsets

❖ Phase 2: Generating association rules from frequent itemset

- To understand the process of generation of association rules from the frequent itemset, let us consider frequent itemset l.
 - For each frequent itemset l generates all non-empty subsets of l.
 - For every non-empty subset s of l, output the rule $s \rightarrow l-s$
 - If the confidence of the rule is more than the threshold value of the confidence, then, this rule will be selected as the final association rule.

Let us generate the rules for the frequent itemset $(2, 3, 5)$ represented as I . Here, non-empty subsets are $\{\{2\}, \{3\}, \{5\}, \{(2, 3)\}, \{(2, 5)\}, \{(3, 5)\}\}$.

For every non-empty subset, the rule will be generated as follows.

$2 \rightarrow 3, 5$ [Here, (2) is s and $(3, 5)$ is $I-s$]

$3 \rightarrow 2, 5$

$5 \rightarrow 2, 3$

$2, 3 \rightarrow 5$

$2, 5 \rightarrow 3$

$3, 5 \rightarrow 2$

The next step will be to calculate the confidence for each rule

The next step will be to calculate the confidence for each rule

$2 \rightarrow 3, 5$; Confidence = $S(2 \cap 3 \cap 5) / S(2) = 2/3 = 0.67$

$3 \rightarrow 2, 5$; Confidence = $S(2 \cap 3 \cap 5) / S(3) = 2/3 = 0.67$

$5 \rightarrow 2, 3$; Confidence = $S(2 \cap 3 \cap 5) / S(5) = 2/3 = 0.67$

$2, 3 \rightarrow 5$; Confidence = $S(2 \cap 3 \cap 5) / S(2 \cap 3) = 2/2 = 1.0$

$2, 5 \rightarrow 3$; Confidence = $S(2 \cap 3 \cap 5) / S(2 \cap 5) = 2/3 = 0.67$

$3, 5 \rightarrow 2$; Confidence = $S(2 \cap 3 \cap 5) / S(3 \cap 5) = 2/2 = 1.0$

Here, the minimum threshold for confidence is 70%, thus selected association rules are

$2, 3 \rightarrow 5$;

$3, 5 \rightarrow 2$;

The possible rules from $2, 3 \rightarrow 5$ and $3, 5 \rightarrow 2$

- It is intuitive to create two new rules as $2 \rightarrow 5$ and $3 \rightarrow 5$ from the given rule. Since $2, 3 \rightarrow 5$ has confidence more than the threshold limit, in this case, it is not implicit or guaranteed that $2 \rightarrow 3$ and $3 \rightarrow 5$ will have confidence more than the threshold limit as there is no correlation between denominator and quotient of both the rules as shown below.

Rule	Confidence
$2, 3 \rightarrow 5$	1.0
$2 \rightarrow 5$	$S(2 \cap 5) / S(2) = 3/3 = 1.0$
$3 \rightarrow 5$	$S(3 \cap 5) / S(3) = 2/3 = 0.67$
$3, 5 \rightarrow 2$	1.0
$3 \rightarrow 2$	$S(3 \cap 2) / S(3) = 2/3 = 0.67$
$5 \rightarrow 2$	$S(5 \cap 2) / S(5) = 3/3 = 1.0$

Selected Association rules are

$2, 3 \rightarrow 5$

$2 \rightarrow 5$

$3, 5 \rightarrow 2$

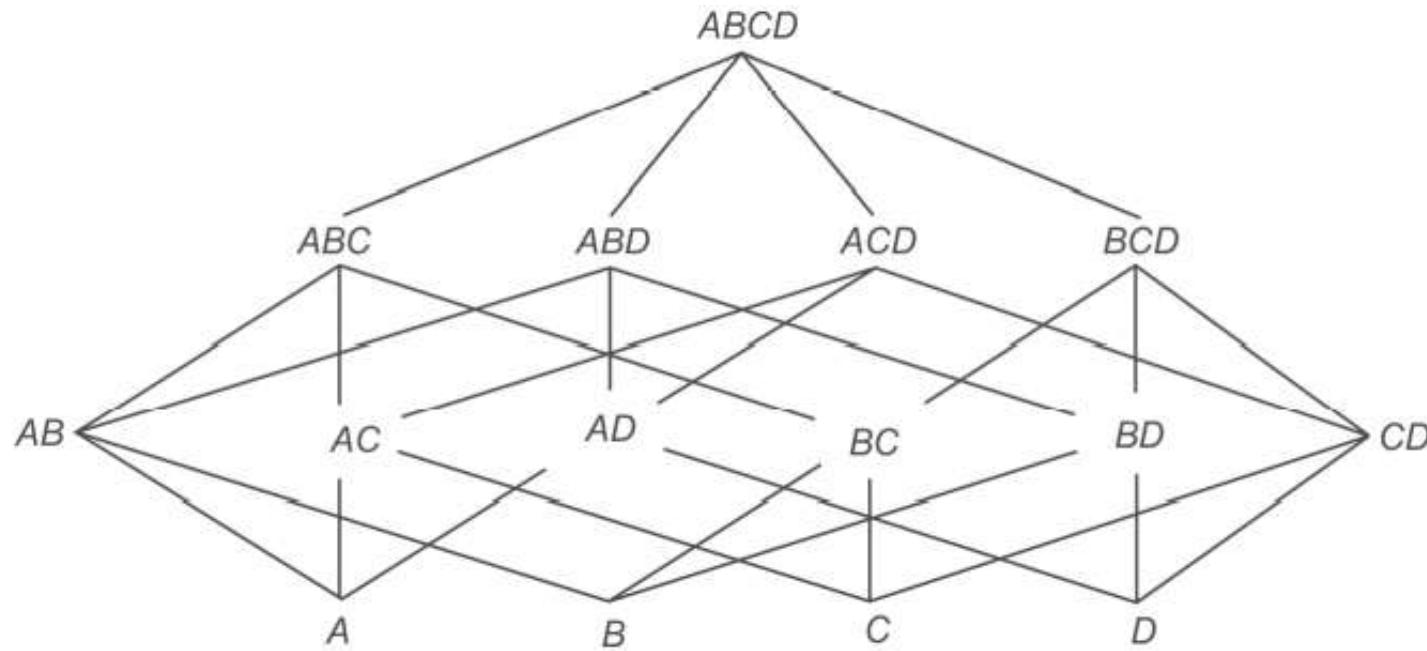
$5 \rightarrow 2$

Apriori property

- Apriori property states that all nonempty subsets of a frequent itemset must also be frequent.
- It means all the subsets of the candidate itemset should be frequent otherwise that itemset will be removed from the candidate itemset.
- This step is **called pruning** of the candidate itemset and it will help to reduce the search space and to improve the performance of the algorithm

Lattice Structure of Frequent Itemsets

- The Apriori property that defines an itemset can be frequent only if **all the subsets of the itemset are frequent.**
- This can be illustrated by a lattice structure.
 - Let us suppose there are four items A, B, C and D in the dataset. Then, the four-itemset ABCD will be frequent only if three itemsets ABC, ABD, ACD and BCD are frequent.
 - These three itemsets are frequent only if their subsets, i.e., AB, AC, BC, BD, AD and CD are frequent.
 - And these two itemsets are frequent only if their subsets, i.e., A, B, C and D items are also frequent



Lattice structure of frequent itemsets

- For example, Apriori property states that for a frequent three itemset (1, 2, 3) all of its non empty subsets, i.e., (1, 2), (2, 3) and (1, 3) must be frequent
- because, support of a superset is always less than or equal to support of its subset. It means that if (1, 3) is not frequent then (1, 2, 3) will also not be frequent.

- This Apriori property also explains the significance of having first k-2 items common in a joining principle.
- Let us suppose, we have L2 that is (1, 2) and (2, 3), then, one may suppose that it could produce a set (1,2,3) as C3. But as discussed earlier, (1, 2, 3) should be frequent only if all of its nonempty subsets, i.e., (1, 2), (2, 3) and (1, 3) are frequent.
- Here, as given in L2 (1, 3) is not frequent. So, itemsets given in L2 are considered as non joinable and the condition of having first k-2 items common is enforced to ensure it.
- Now, what would happen if L2 has (1, 2) and (1, 3) but not (2, 3)? Then, according to the joining principle C3 will be generated as (1, 2, 3), but it will be **discarded by the** Apriori property because one of its nonempty subset, i.e., (2, 3) is not frequent. Thus, the combination of joining principle and apriori property make the whole process complete

Improving of Apriori Algorithm by Pruning the Itemsets

- ❖ Find association rules for the transaction data given in Table for having support 15% and confidence 70%.

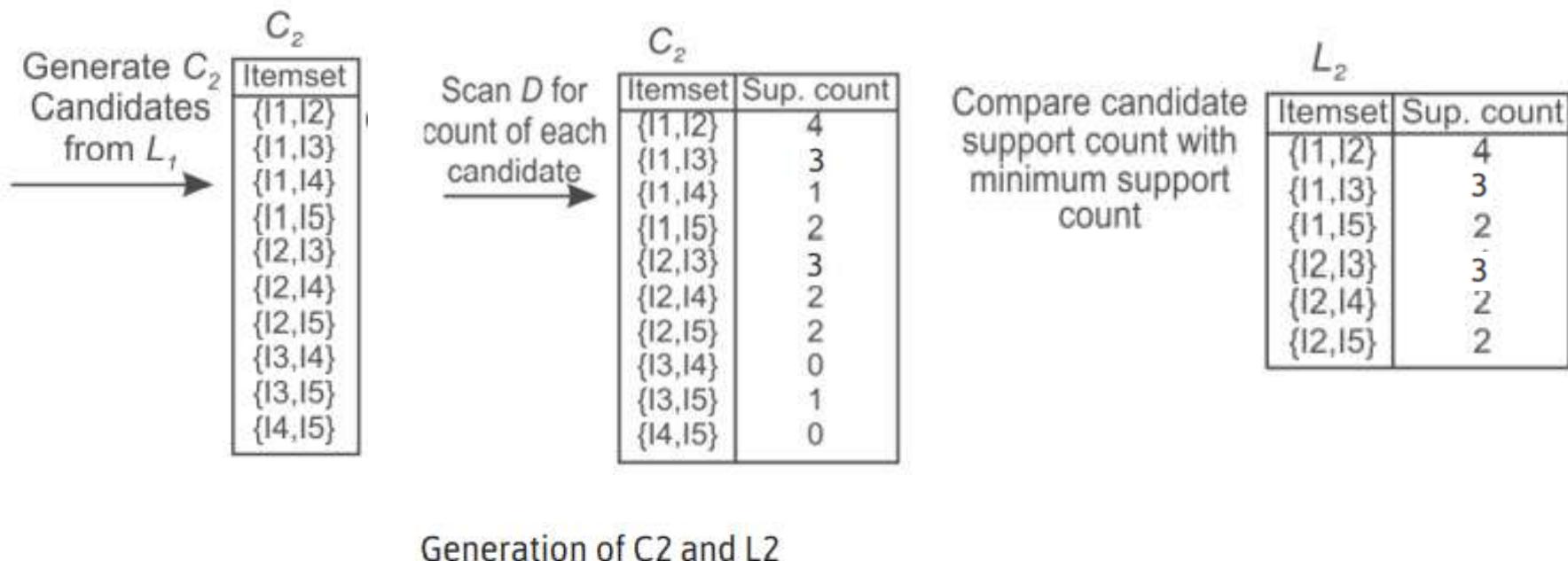
Transaction database for identification of association rules

TID	List of Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2,

The frequency count for each item, C1 is given in

C1	Count
I1	6
I2	7
I3	5
I4	2
I5	2

- All the items have frequency more than the specified support limit, thus all 1-item candidate sets given as C1 qualify as 1-item frequent itemset L1.
- The next step will be to generate C2. $C_2 = L_1 \text{ JOIN } L_1$ The process of creation of C2 and L2 is



- From the given L2, C3 is generated by considering those item pairs whose first k-2 items, i.e. first 1 item is common

Generation of C3

C3
I1, I2, I3
I1, I2, I5
I1, I3, I5
I2, I3, I4
I2, I3, I5
I2, I4, I5

- Before finding the frequency of each candidate item pair, the Apriori property will be applied to prune the candidate list.
- Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent

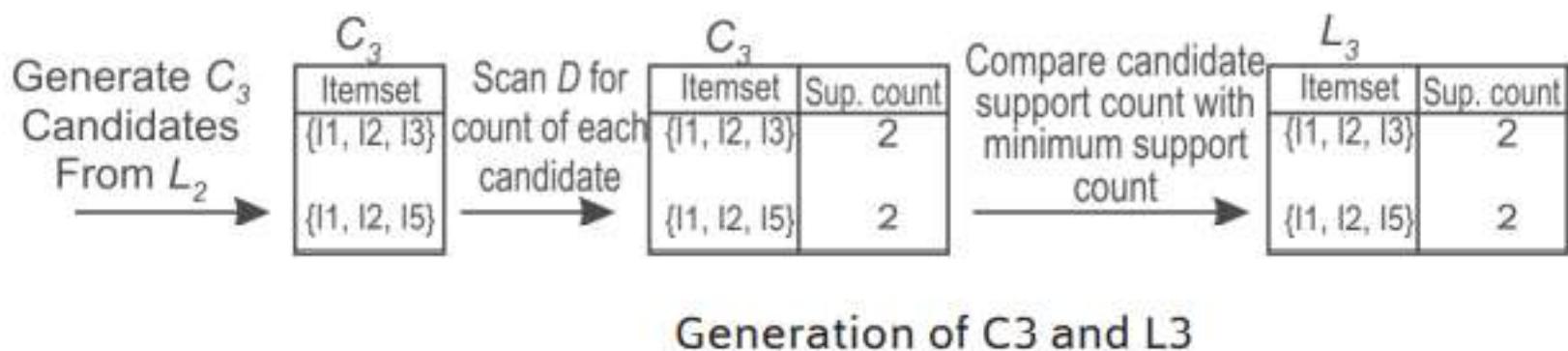
Pruning of candidate itemset C3

I1, I2, I3	The 2-item subsets of I1, I2 and I3 are (I1, I2), (I1, I3), and (I2, I3). All 2-item subsets are members of L2. Therefore, keep I1, I2 and I3 in C3.	Qualify
I1, I2, I5	The 2-item subsets of I1, I2 and I5 are (I1, I2), (I1, I5), and (I2, I5). All 2-item subsets are members of L2. Therefore, keep I1, I2 and I5 in C3.	Qualify
I1, I3, I5	The 2-item subsets of I1, I3 and I5 are (I1, I3), (I1, I5), and (I3, I5). Since, (I3, I5) is not a member of L2, and so it is not frequent. Therefore, remove I1, I3 and I5 from C3.	Does not qualify
I2, I3, I4	The 2-item subsets of I2, I3 and I4 are (I2, I3), (I2, I4), and (I3, I4). Since, (I3, I4) is not a member of L2, and so it is not frequent. Therefore, remove I2, I3 and I4 from C3.	Does not qualify
I2, I3, I5	The 2-item subsets of I2, I3 and I5 are (I2, I3), (I2, I5), and (I3, I5). Since, (I3, I5) is not a member of L2, so it is not frequent. Therefore, remove I2, I3 and I5 from C3.	Does not qualify
I2, I4, I5	The 2-item subsets of I2, I4 and I5 are (I2, I4), (I2, I5), and (I4, I5). Since, (I4, I5) is not a member of L2, it is not frequent. Therefore, remove I2, I4 and I5 from C3.	Does not qualify

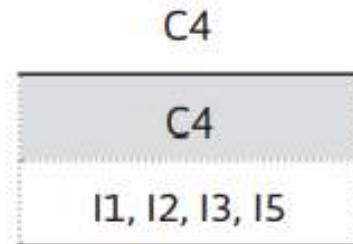
- Thus pruned C3 will be

Pruned C3
I1, I2, I3
I1, I2, I5

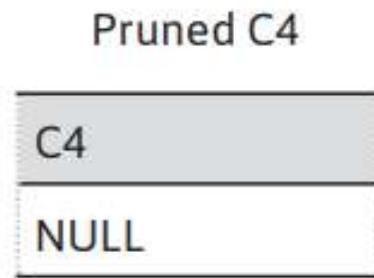
- The process of creation of L3 from C3 by identifying those 3 itemsets that has a frequency more than or equal to the given value of threshold value of support, i.e., 2



- The next step will be to generate C4 as shown below. $C4 = L3 \text{ JOIN } L3$
 $C4$ will be I1, I2, I3 & I5, because 2 items are commo



- But this itemset is pruned by the Apriori property because its subset (I2, I3, I5) is not frequent as it is not present in L3.
- Thus, C4 is null and the algorithm terminates at this point



- In this case, instead of finding the count of 1 four-item pair, pruning indicates that there is no need to find its count as all its subsets are not frequent.
- With this, the first phase of the Apriori algorithm has been completed.
- In this we will generate the rule for frequent 3-itemsets
 (I_1, I_2, I_3) and (I_1, I_2, I_5) .

Phase 2: Generating association rules from frequent itemsets

- Let us apply this rule to frequent 3-itemsets (I_1, I_2, I_3) and (I_1, I_2, I_5).
- For first frequent itemset (I_1, I_2, I_3), non-empty subsets are $\{\{I_1\}, \{I_2\}, \{I_3\}, \{(I_1, I_2)\}, \{(I_1, I_3)\}, \{(I_2, I_3)\}\}$

$I_1 \rightarrow I_2, I_3$ [Here, (I_1) is s and I_2 and I_3 are $l-s$]

$I_2 \rightarrow I_1, I_3$

$I_3 \rightarrow I_1, I_2$

$I_1, I_2 \rightarrow I_3$

$I_1, I_3 \rightarrow I_2$

$I_2, I_3 \rightarrow I_1$

The next will be to calculate the confidence for each rule as shown below.

$$I_1 \rightarrow I_2, I_3; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_1) = 2/6 = 0.3$$

$$I_2 \rightarrow I_1, I_3; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_2) = 2/7 = 0.28$$

$$I_3 \rightarrow I_1, I_2; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_3) = 2/5 = 0.4$$

$$I_1, I_2 \rightarrow I_3; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_1 \cap I_2) = 2/4 = 0.5$$

$$I_1, I_3 \rightarrow I_2; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_1 \cap I_3) = 2/4 = 0.5$$

$$I_2, I_3 \rightarrow I_1; \text{ Confidence} = S(I_1 \cap I_2 \cap I_3) / S(I_2 \cap I_3) = 2/4 = 0.5$$

- Since, the minimum threshold is 70%, there are no rules that qualify from the frequent itemset {1, 2, 3}.

- Now, let us apply this rule to second frequent 3-itemset (I_1, I_2, I_5).
- For this frequent itemset, non-empty subsets are $\{I_1\}$, $\{I_2\}$, $\{I_5\}$, $\{(I_1, I_2)\}$, $\{(I_1, I_5)\}$, $\{(I_2, I_5)\}$.

$I_1 \rightarrow I_2, I_5$
 $I_2 \rightarrow I_1, I_5$
 $I_5 \rightarrow I_1, I_2$
 $I_1, I_2 \rightarrow I_5$
 $I_1, I_5 \rightarrow I_2$
 $I_2, I_5 \rightarrow I_1$

The next step will be to calculate the confidence for each rule

$I_1 \rightarrow I_2, I_5$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_1) = 2/6 = 0.3$
 $I_2 \rightarrow I_1, I_5$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_2) = 2/7 = 0.28$
 $I_5 \rightarrow I_1, I_2$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_5) = 2/2 = 1$
 $I_1, I_2 \rightarrow I_5$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_1 \cap I_2) = 2/4 = 0.5$
 $I_1, I_5 \rightarrow I_2$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_1 \cap I_5) = 2/2 = 1$
 $I_2, I_5 \rightarrow I_1$; Confidence = $S(I_1 \cap I_2 \cap I_5) / S(I_2 \cap I_5) = 2/2 = 1$

- Now, there are three rules whose confidence is more than minimum threshold value of 70%, and these rules are

$I5 \rightarrow I1, I2$

$I1, I5 \rightarrow I2$

$I2, I5 \rightarrow I1$

- Find rules whose confidence is more than minimum threshold value of 70%, and these rules are

The final association rules

$I5 \rightarrow I1, I2$

$I2, I5 \rightarrow I1$

$I1, I5 \rightarrow I2$

$I5 \rightarrow I1$

$I5 \rightarrow I2$

- The possible rules from $I5 \rightarrow I1, I2$

$I5 \rightarrow I1$

$I5 \rightarrow I2$

- The possible rules from $I1, I5 \rightarrow I2$

$I1 \rightarrow I2$

$I5 \rightarrow I2$

- The possible rules from $I2, I5 \rightarrow I1$

$I2 \rightarrow I1$

$I5 \rightarrow I1$

- ❖ Consider the following group of frequent 3-itemsets and hence, identify C4 after pruning the following frequent three itemsets as L3.

$\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 2, 5\}$, $\{1, 3, 4\}$, $\{1, 3, 5\}$, $\{2, 3, 4\}$, $\{2, 3, 5\}$, $\{3, 4, 5\}$.

Frequent Pattern-Growth (FP Growth)

- As only frequent items are required for identifying the association rules, so it is best to identify the frequent items in the dataset and ignore all others.
- Instead of generating candidate itemsets as in the case of the Apriori algorithm, frequent pattern-growth (FP growth) only tests and generates frequent itemsets.
- The major difference between the Apriori algorithm and FP growth is that FP growth does not generate candidate itemsets, it only tests, whereas the Apriori algorithm generates candidate itemsets and then tests them.
- To improve the performance of the algorithm, it uses the principle to store frequent items in a compact structure which does not require the need to use the original transaction database repeatedly.

FP Growth aLGORITHM

- 1) Like the Apriori algorithm, the transaction database is scanned once to identify all the 1-itemset frequent items and their supports are noted down.
- 2) Then 1-itemset frequent items are sorted in descending order of their support.
- 3) After that the FP-tree is created with a ‘NULL’ root.
 - 1) Then the first transaction is obtained from the transaction database and all the non-frequent items are removed. And the remaining items are listed according to the order in the sorted frequent items.
 - 2) After this, the first branch of the tree is constructed using the transaction with each node corresponding to a frequent item and by setting the frequency of each item as 1 for the first transaction.

- 3) Then the next transaction is retrieved from the transaction database and all the non-frequent items are removed. And the remaining items are listed according to the order in the sorted frequent items as done in step 4.
 - 4) Now, the transaction is inserted in the tree using any common prefix that may appear and count of items is increased by 1. If no common prefix is found then the branch of the tree is constructed using the transaction; with each node corresponding to a frequent item and by setting the frequency of each item as 1.
 - 5) Step 6 is repeated until all transactions in the database have been processed.
- 4) Create the association rules, similar to the Apriori algorithm.

- ❖ Consider the below transaction database. The minimum support required is 50% and confidence is 75%.

100	Butter, Curd, Eggs, Jam
200	Butter, Curd, Jam
300	Butter, Eggs, Muffin, Nuts
400	Butter, Eggs, Jam, Muffin
500	Curd, Jam, Muffin

➤ First step will be to scan the transaction database to identify all the frequent items in the 1-itemsets and sort them in descending order of their support.

➤ **Nuts has been removed** as it has a support of 1 only

Frequency of each item in sorted order

Item	Frequency
Butter	4
Jam	4
Curd	3
Eggs	3
Muffin	3

- To create the FP-tree, the next step will be to exclude all the items that are not frequent from the transactions and sort the remaining frequent items in descending order of their frequency count.
- It is important to note that in case of **identical support**, items are arranged according to **lexicographic or sorted order**.

Updated database after eliminating the non-frequent items and reorganising it according to support

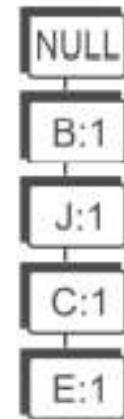
<i>Transaction ID</i>	<i>Items</i>
100	Butter, Jam, Curd, Eggs
200	Butter, Jam, Curd
300	Butter, Eggs, Muffin
400	Butter, Jam, Eggs, Muffin
500	Jam, Curd, Muffin

➤ It is important to note that in case of identical support, items are arranged according to lexicographic or sorted order.

➤ To built the FP-tree let us consider first transaction, i.e., 100 having Butter, Jam, Curd, Eggs as items. The tree is built by making a root node labeled NULL. A node is made for each frequent item in the first transaction and the count is set to 1.

100

Butter, Jam, Curd, Eggs



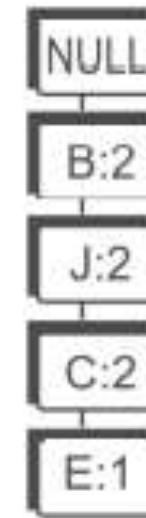
➤ For example, to build the FP-tree for the first transaction {B, J, C, E} is inserted in the empty tree with the root node labeled NULL. Each of these items is given a frequency count of 1.

FP-tree for first transaction,
i.e., 100 only

- Next, the tree is traversed for the next transaction, i.e., 200. If a path already exists then it will follow the same path and corresponding count of item is increased by 1.
- If a path does not exist then a new path will be created. Next the second transaction, which is {B, J, C} having all the items common with first, is inserted. It changes the frequency of each item, i.e., {B, J, C} to 2.

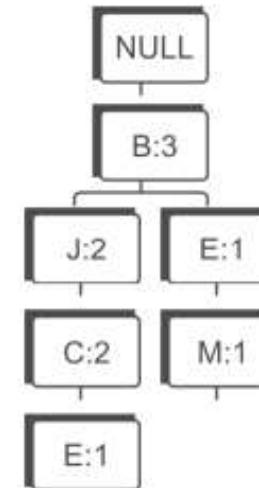
200

Butter, Jam, Curd

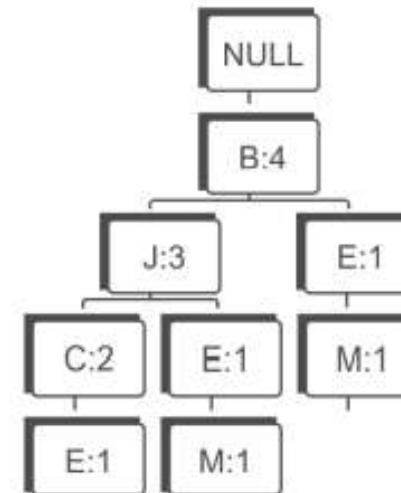


FP-tree for the first two transactions

- Similarly, third transaction is considered and {B, E, M} is inserted. This requires that nodes for E and M be created. The counter for B goes to 3 and the counter for E and M is set to 1.

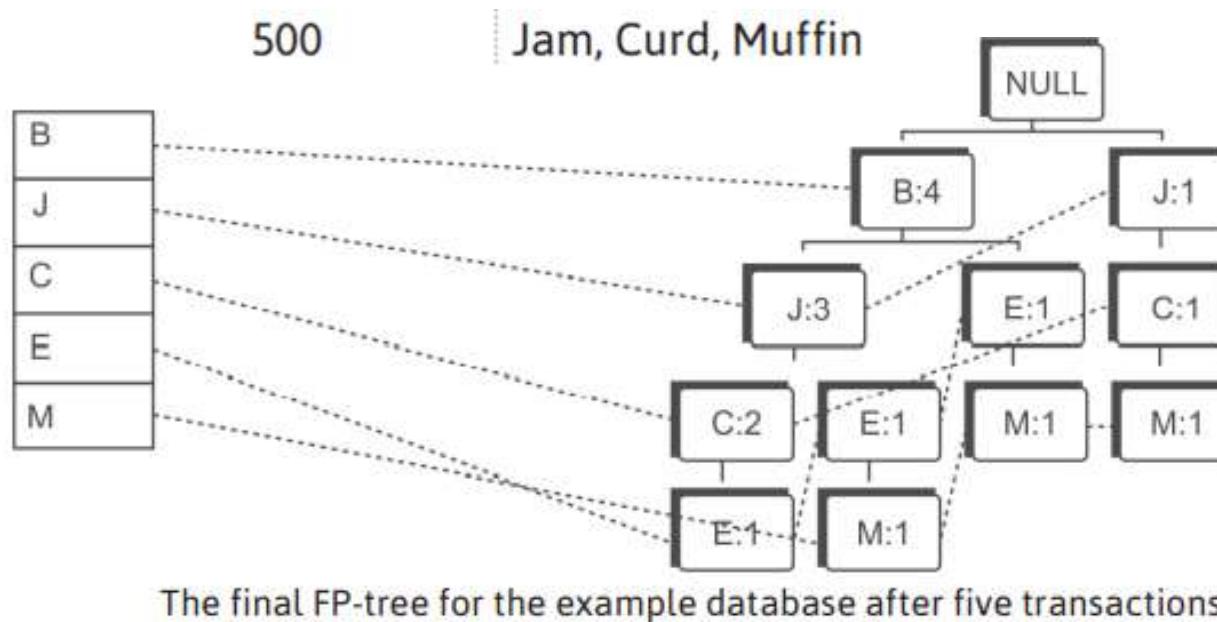


FP-tree for first three transactions



FP-tree for first four transactions

- The last transaction $\{J, C, M\}$ results in a brand new branch for the tree with a counter of 1 for each item.

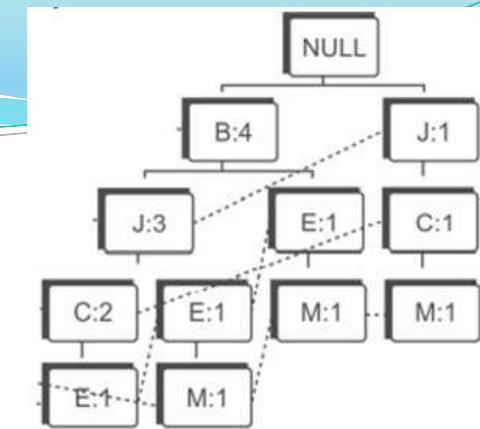


- The count tells the number of occurrences the path (constructed from the root node to this node) has in the transaction database.
- The FP-tree also consists of a header table with an entry for each itemset and a link to the first item in the tree with the same name

- The nodes near the root in the tree are more frequent than those further down the tree.
- Each identical path is only stored in the FP-tree once, which often makes the size of an FP-tree smaller than the corresponding transactional database.
- The height of an FP-tree is always equal to the **maximum number of itemsets** in a transaction excluding the root node.

Identification of frequent itemsets from FP-tree

- The mining on the FP-tree structure is performed using the Frequent Pattern growth (FP growth) algorithm. **This algorithm starts with the least frequent item**, i.e., the last item in the header table. Then the algorithm identifies all the paths from the root to this least frequent item and adjusts the count on the basis of the item's support count



❖ Identification of patterns for item M

BEM(1)

BJEM(1)

JCM(1)

- From this we can identify that, the support for BM is 2 (1 from BEM and 1 from BJEM), but its support is less than the threshold limit of 3, so this item pair will be **discarded** and we have identified no frequent item pairs by using item M.

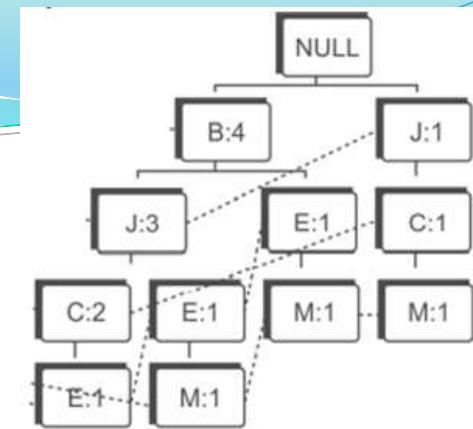
❖ Identification of patterns for item E

BJCE(1)

BJE(1)

BE(1)

- From this we can identify that the support for **BE is 3** (1 from BJCE, 1 from BJE and 1 from BE). However there is also a three item pair BJE having a support of 2 (1 from BJCE and 1 BJE), but its support is less than threshold value so will be discarded.
- So, we have identified **BE has two** item pairs having support more than the threshold limit.



❖ Identification of patterns for item C

BJC(2)

JC(1)

- From this we can identify that, the support for **JC** is **3** (2 from BJC and 1 from JC). So, we have identified JC has two item pairs

❖ Identification of patterns for item J

BJ(3)

J(1)

- From this we can identify that, the support for BJ is 3. There is no other item pair having support equal to or more than 3.

Frequent item pairs for database

Frequent Item pairs	Count
BE	3
JC	3
BJ	3

Finding association rules

- To find the association mining rules, **the same approach is followed as in Apriori algorithm**. For item pair BE, the possible rules are $B \rightarrow E$ and $E \rightarrow B$.
- The confidence for each rule will be calculated and if its value is more than given threshold value of confidence then corresponding rule will be selected otherwise it will be discarded.

Confidence of $(B \rightarrow E) = S(B \cap E) / S(B) = 3/4 = 0.75$

Confidence of $(E \rightarrow B) = S(E \cap B) / S(E) = 3/3 = 1.0$

Confidence of $(J \rightarrow C) = S(J \cap C) / S(J) = 3/4 = 0.75$

Confidence of $(C \rightarrow J) = S(C \cap J) / S(C) = 3/3 = 1$

Confidence of $(B \rightarrow J) = S(B \cap J) / S(B) = 3/4 = 0.75$

Confidence of $(J \rightarrow B) = S(J \cap B) / S(J) = 3/4 = 0.75$

- Consider given database to identify frequent item pairs having support 50% and confidence 70%.

Transaction database

TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Count for each data item

Item	Count
A	2
B	3
C	3
D	1
E	3

Frequency of each item in sorted order

Item	Count
B	3
C	3
E	3
A	2

- Now, the non-frequent items are removed from the transaction database and items are ordered on the basis of their frequency

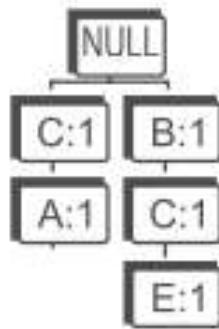
Modified database after eliminating the non-frequent items and reorganising

TID	Items
100	C A
200	B C E
300	B C E A
400	B E

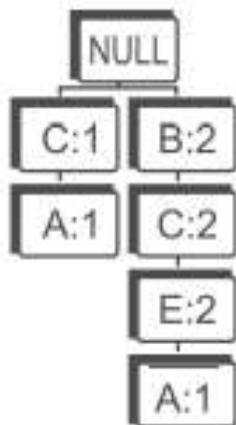
TID	Items
100	C A
200	B C E
300	B C E A
400	B E



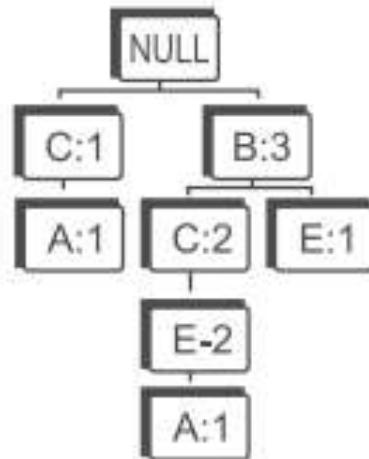
FP-tree for first transaction



FP-tree for first two transactions

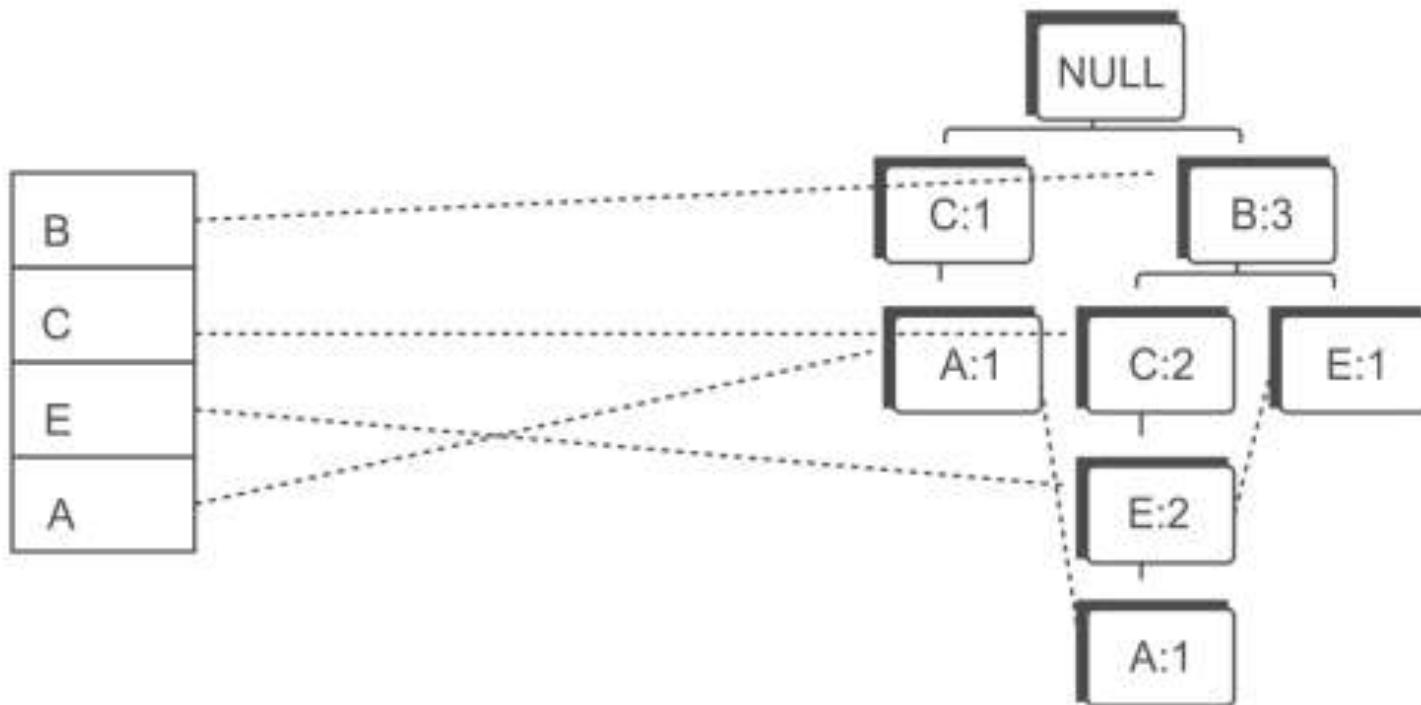


FP-tree for first three transactions



Full FP-tree for all four transactions

TID	Items
100	C A
200	B C E
300	B C E A
400	B E



Final FP-tree for database

Identification of Frequent Itemsets from the FP-tree

❖ Identification of patterns for item A

CA(1)

BCEA(1)

- The support for CA is 2. It is equal to the threshold limit thus **CA is** identified as a frequent item pair.

❖ Identification of patterns for item E

BCE(2)

BE(1)

- The support for BCE is 2 and it is equal to the threshold limit thus BCE is identified as a frequent 3-itemset.
- BE is identified as a frequent 2-item pair, since it has a support of 3.
- Thus, we have selected two frequent item pairs from E, i.e., **BCE and BE**.

❖ Identification of patterns for item C

BC(2)

- The support for BC is 2 and it is equal to the threshold limit thus BC is identified as a frequent item pair.

Frequent item pairs for the example database

Frequent Item pairs	Count
CA	2
BCE	2
BE	2
BC	2

❖ Finding association rules

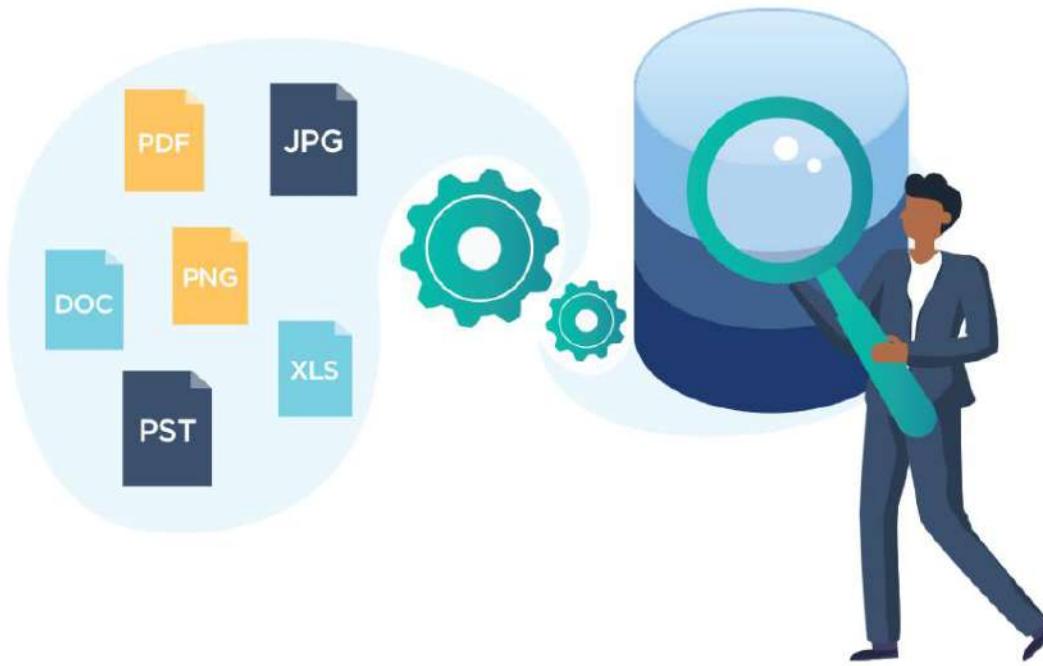
Calculation of confidence for identification of association rules

CA	$C \rightarrow A = S(C \cap A) / S(C) = 2/3 = 0.67$ $A \rightarrow C = S(A \cap C) / S(A) = 2/2 = 1.0$	Selected rule is $A \rightarrow C$
BCE	$B \rightarrow CE = S(B \cap C \cap E) / S(B) = 2/3 = 0.67$	Confidence is less than threshold value, so it will be discarded.
	$CE \rightarrow B = S(B \cap C \cap E) / S(CE) = 2/2 = 1.0$ Possible other non-implied rules are: $C \rightarrow E$, $E \rightarrow B$ $C \rightarrow E = S(C \cap E) / S(C) = 2/3 = 0.67$ $E \rightarrow B = S(E \cap B) / S(E) = 2/3 = 0.67$	Selected rule is $CE \rightarrow B$. Others are discarded
BE	$B \rightarrow E = S(B \cap E) / S(B) = 2/3 = 0.67$ $E \rightarrow B = S(E \cap B) / S(E) = 2/3 = 0.67$	Both the rules are discarded.
BC	$B \rightarrow C = S(B \cap C) / S(B) = 2/3 = 0.67$ $C \rightarrow B = S(C \cap B) / S(C) = 2/3 = 0.67$	Both the rules are discarded.

Thus, final association rules having confidence more than 70% are $CE \rightarrow B$ and $A \rightarrow C$.

*Thank you for your
attention!*

Data Mining (20CP306T)

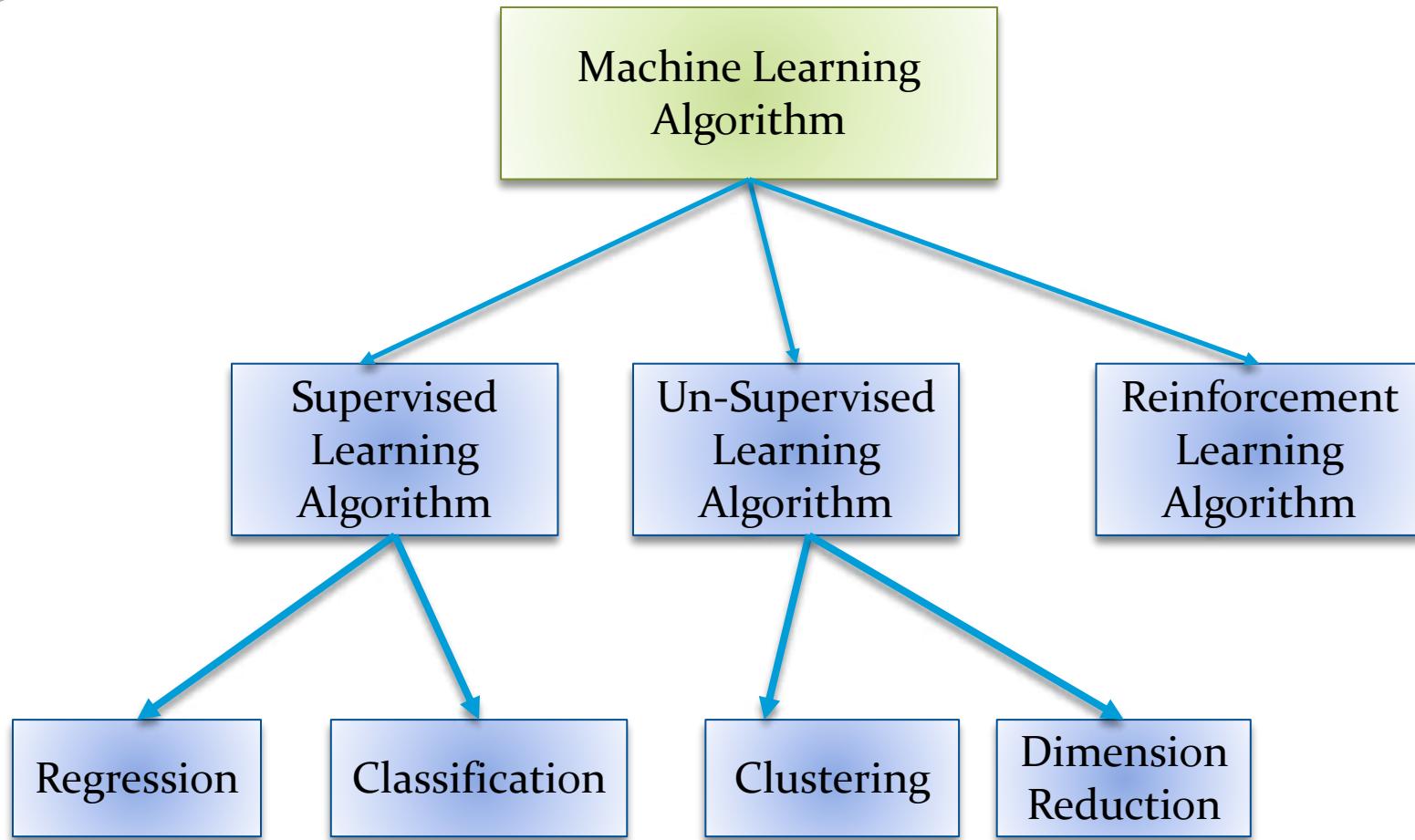


Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

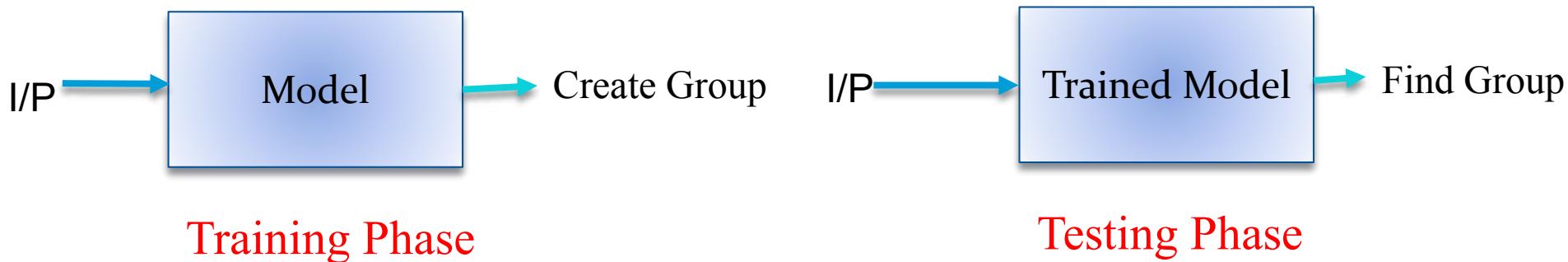
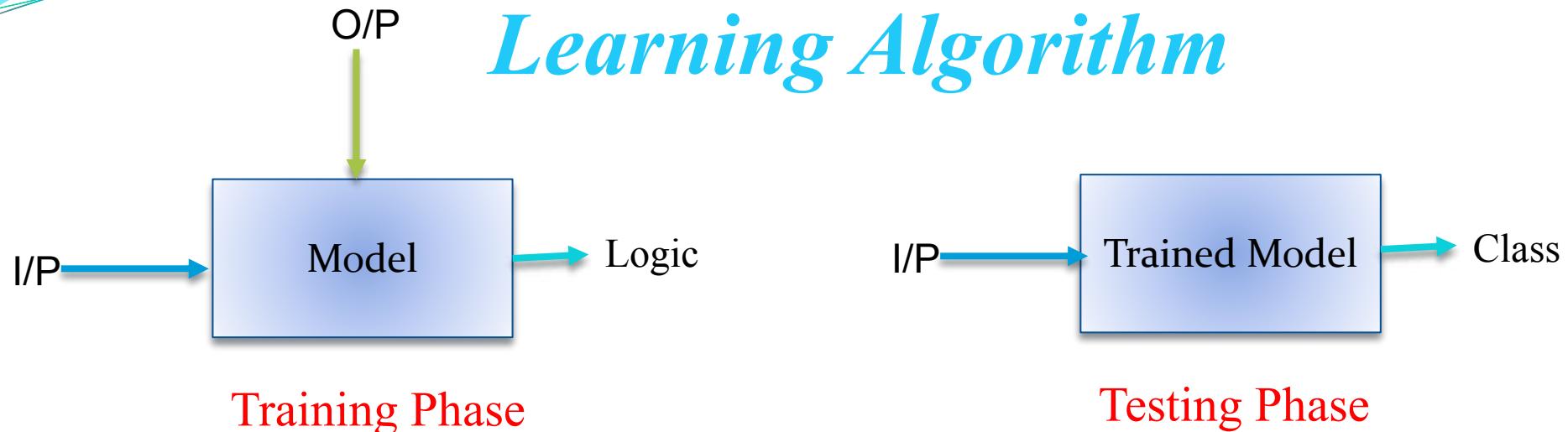
UNIT 4

UNSUPERVISED LEARNING & CLUSTERING

- ❖ Clustering, KNN, Clustering Review, Outlier Detection, Recent Trends in Data Mining

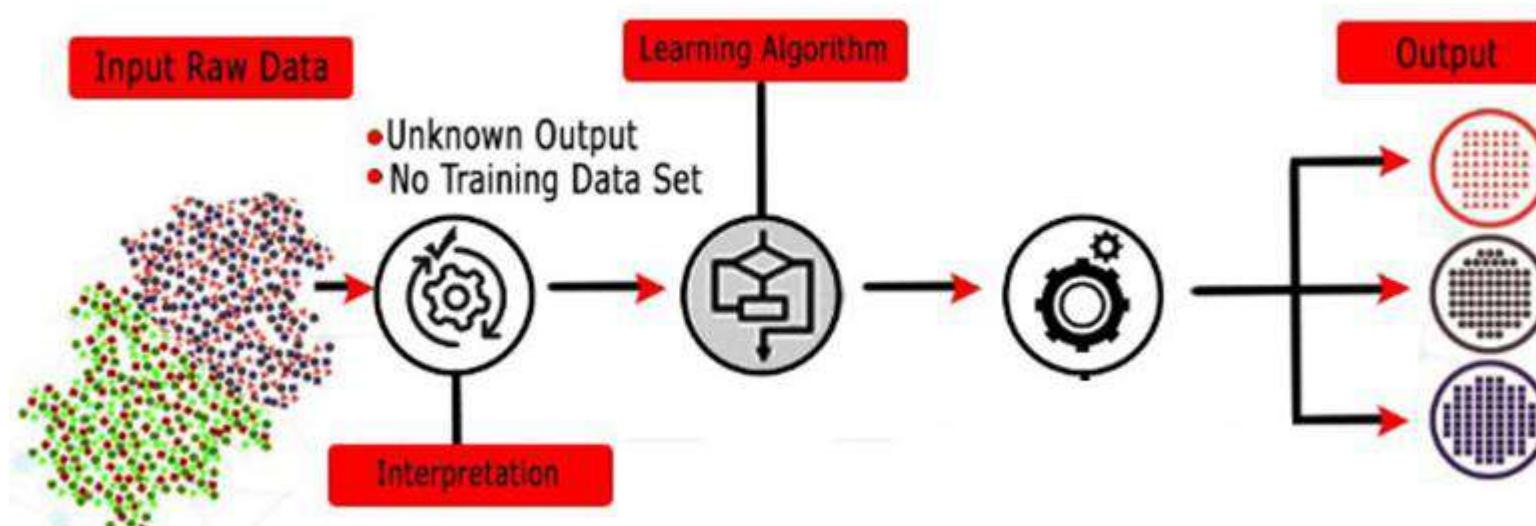


Learning Algorithm



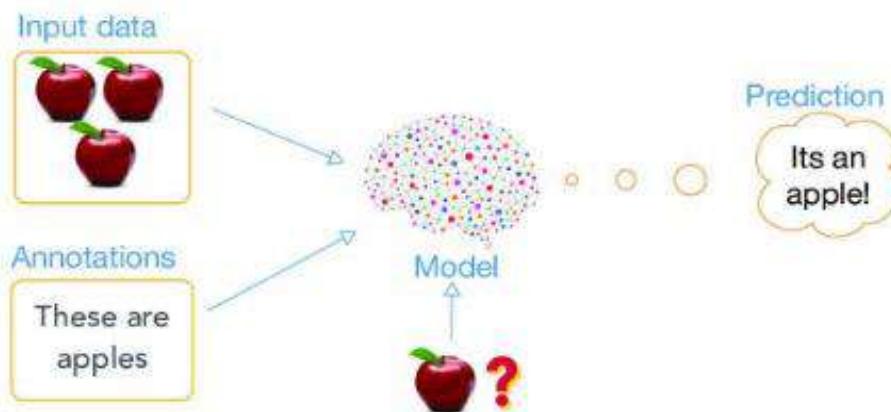
Unsupervised Learning

- Training model for unlabeled data.

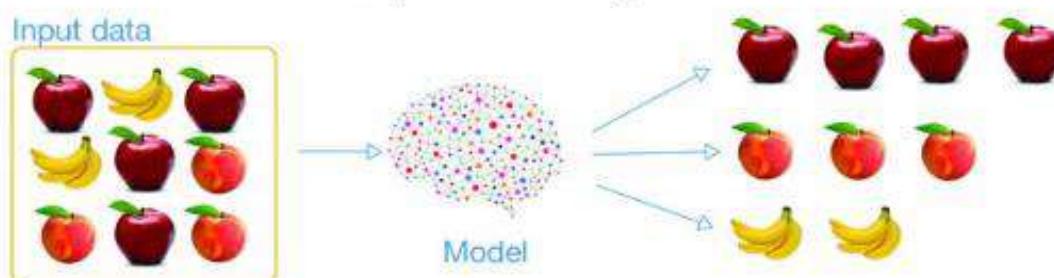


Supervised Learning v/s Unsupervised Learning

supervised learning

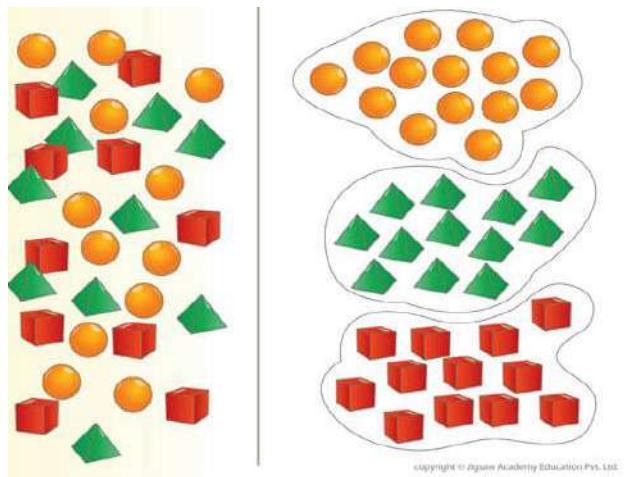


unsupervised learning

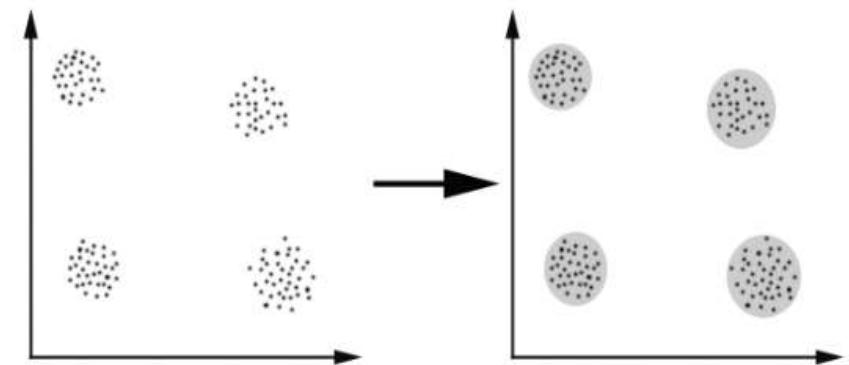


Clustering

- *Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.*
- In clustering, we do not have a target to predict. We look at the data and then try to club similar observations and form different groups.
- All the data points in a cluster should be similar to each other.
- The data points from different clusters should be as different as possible.

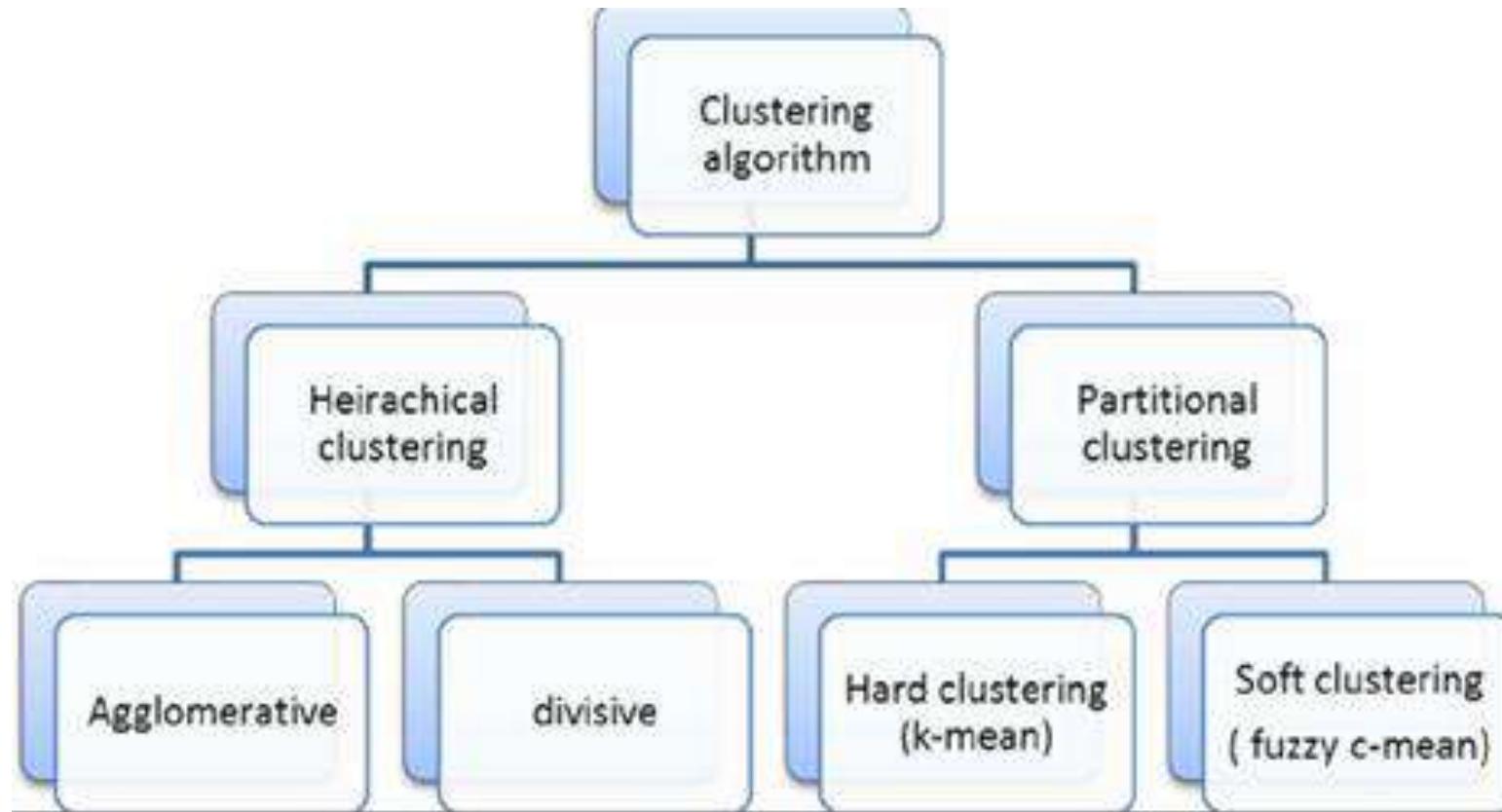


Source: <http://bampe08.blogspot.com/2015/03/cluster-analysis-for-business.html>



Source: <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eecb78b422a>

Types of Clustering Algorithm

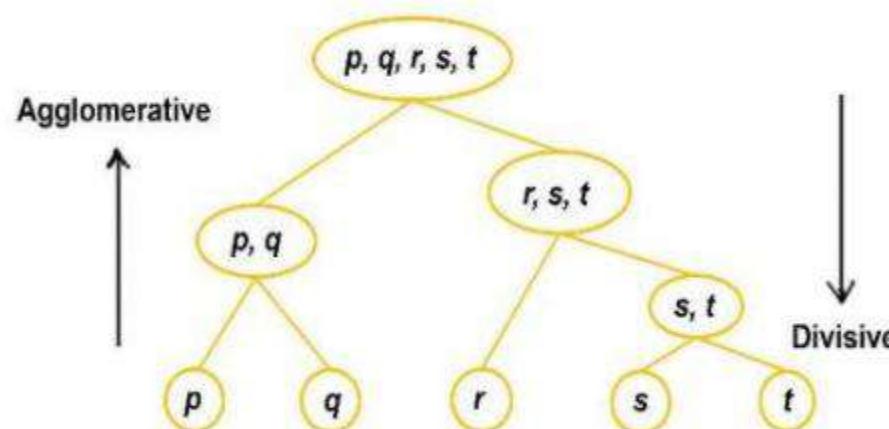


<https://medium.com/datadriveninvestor/clustering-algorithms-9fd35f34caa3>

Agglomerative and Divisive

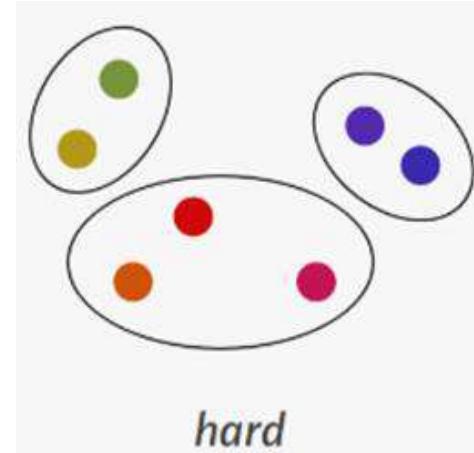
Agglomerative: This is a "bottom-up" approach: each observation starts in its own **cluster**, and pairs of **clusters** are merged as one moves up the hierarchy.

Divisive: This is a "top-down" approach: all observations start in one **cluster**, and splits are performed recursively as one moves down the hierarchy.



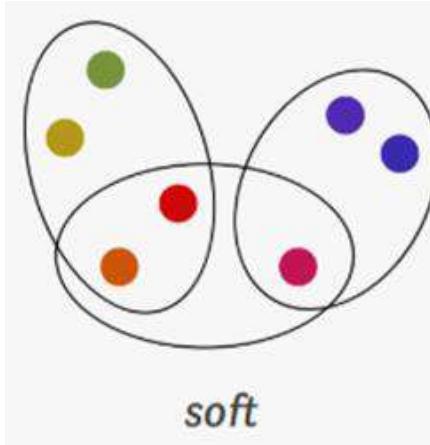
Hard Clustering: In hard clustering, each data point either belongs to a cluster completely or not.

Ex. K-mean clustering



Soft Clustering: In soft clustering given data point can belong to more than one cluster in soft clustering. This is also known as overlapping clustering.

Ex. Fuzzy C-mean clustering



Distance Metrics in Machine Learning

Distance Metrics in Machine Learning

- How will you define the similarity between different observations here?
- How can we say that two points are similar to each other?



To measure similarity between continuous or numerical variables

- 1.Euclidean Distance
- 2.Manhattan Distance
- 3.Minkowski Distance

To measure similarity between categorical variables

- 1.Hamming Distance

To measure similarities between different documents

- 1.Cosine distance

Which Distance Should Used?

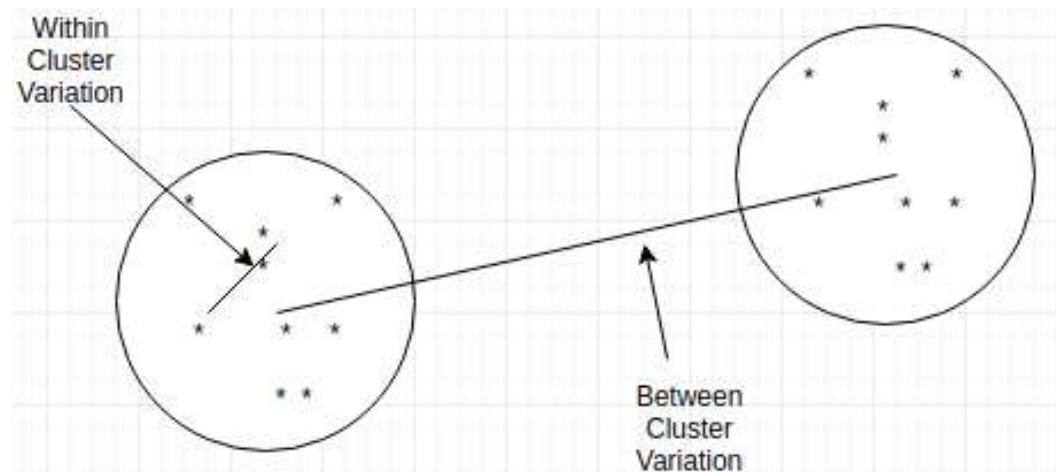
- Manhattan distance is usually preferred over the more common Euclidean distance when there is high dimensionality in the data.
- Hamming distance is used to measure the distance between categorical variables.
- Cosine distance metric is mainly used to find the amount of similarity between two data points.

Clustering Evaluation Metric

- Clusters can be evaluated with “**internal**” as well as “**external**” measures.
- Internal measures are related to the **inter/intra cluster** distance.
 - ❖ A good clustering is one where (Intra-cluster distance) the sum of distances between objects in the same cluster are minimized, (Inter-cluster distance) while the distances between different clusters are maximized.
- External measures are related to how representative are the current clusters to “true” classes.
 - ❖ Measured in terms of purity, entropy or F-measure

WCV (Within Cluster Variation): The variation in the data points that are present in the cluster.

BCV (Between Cluster Variation): The variation between 2 clusters.



<https://medium.com/datadriveninvestor/k-means-clustering-4a700d4a4720>

- The goal of K Means algorithm is to minimize the Within Cluster Variation and maximize the Between Cluster Variation.

Cluster Validation

1) Unsupervised Measure (No ground truth)

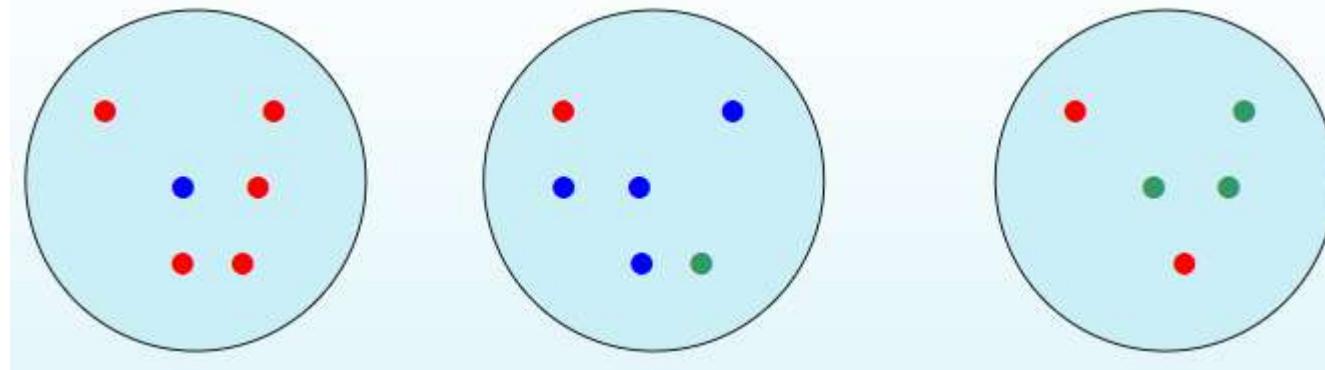
- Cluster Cohesion and cluster separation
- Sum square error (SSE)
- Silhouette Coefficient
- Internal indices

2) Supervised Measure (Have ground truth)

- External Indices
- Purity
- Rand Index
- Entropy
- Jaccard Coefficient

3) Relative Measure (Combination of supervised and unsupervised)

Cluster Purity (Supervised Measurement)



<https://slideplayer.com/slide/4838686/>

Cluster-1

Cluster-2

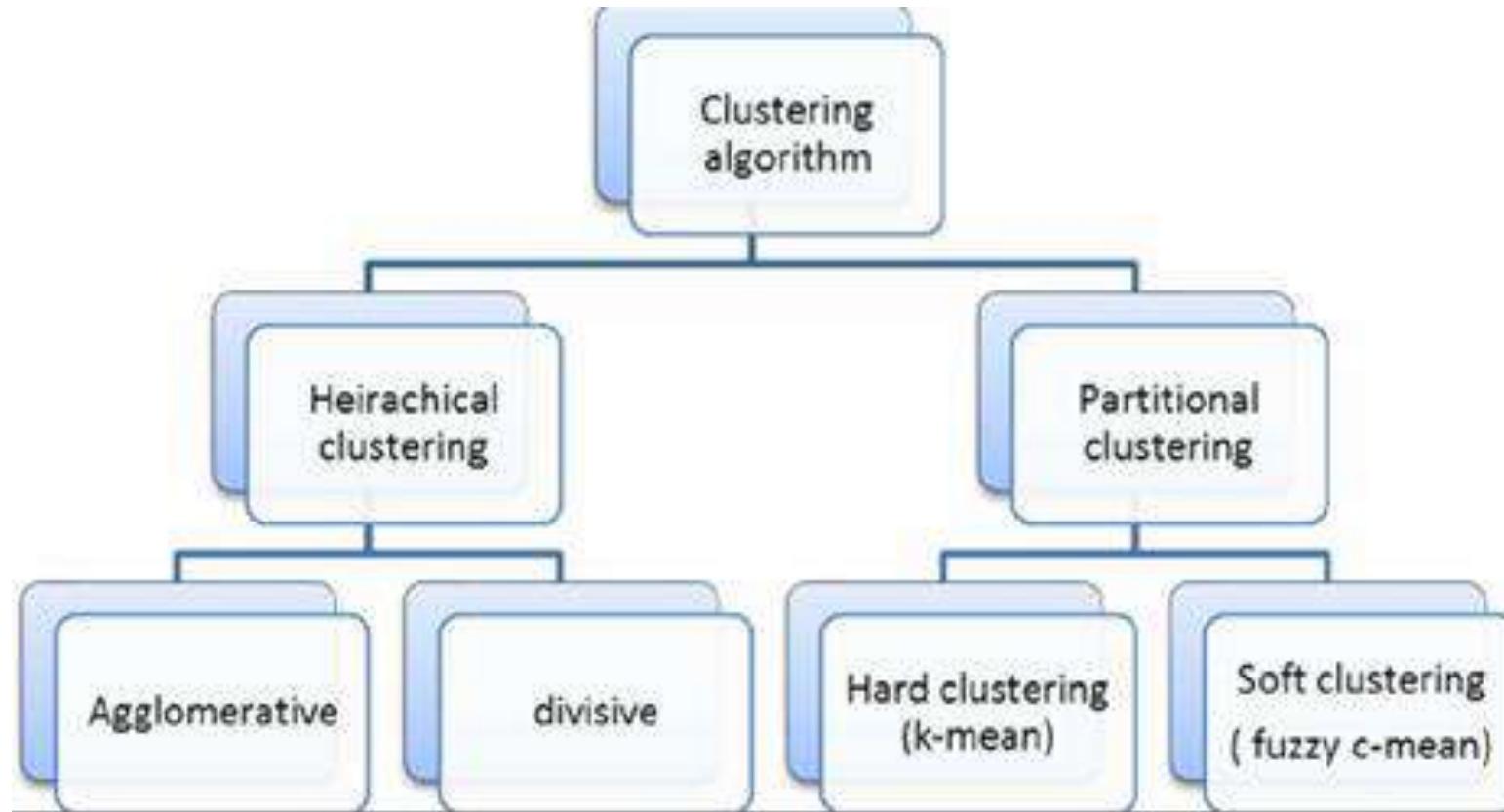
Cluster-3

$$\text{Purity of clustering} = \frac{\text{Sum of pure sizes of clusters}}{\text{Total number of elements across clusters}}$$

Pure size of a cluster = # elements from the majority class

$$= (5+4+3) / (6+6+5) = 12/17 = 0.71$$

Types of Clustering Algorithm

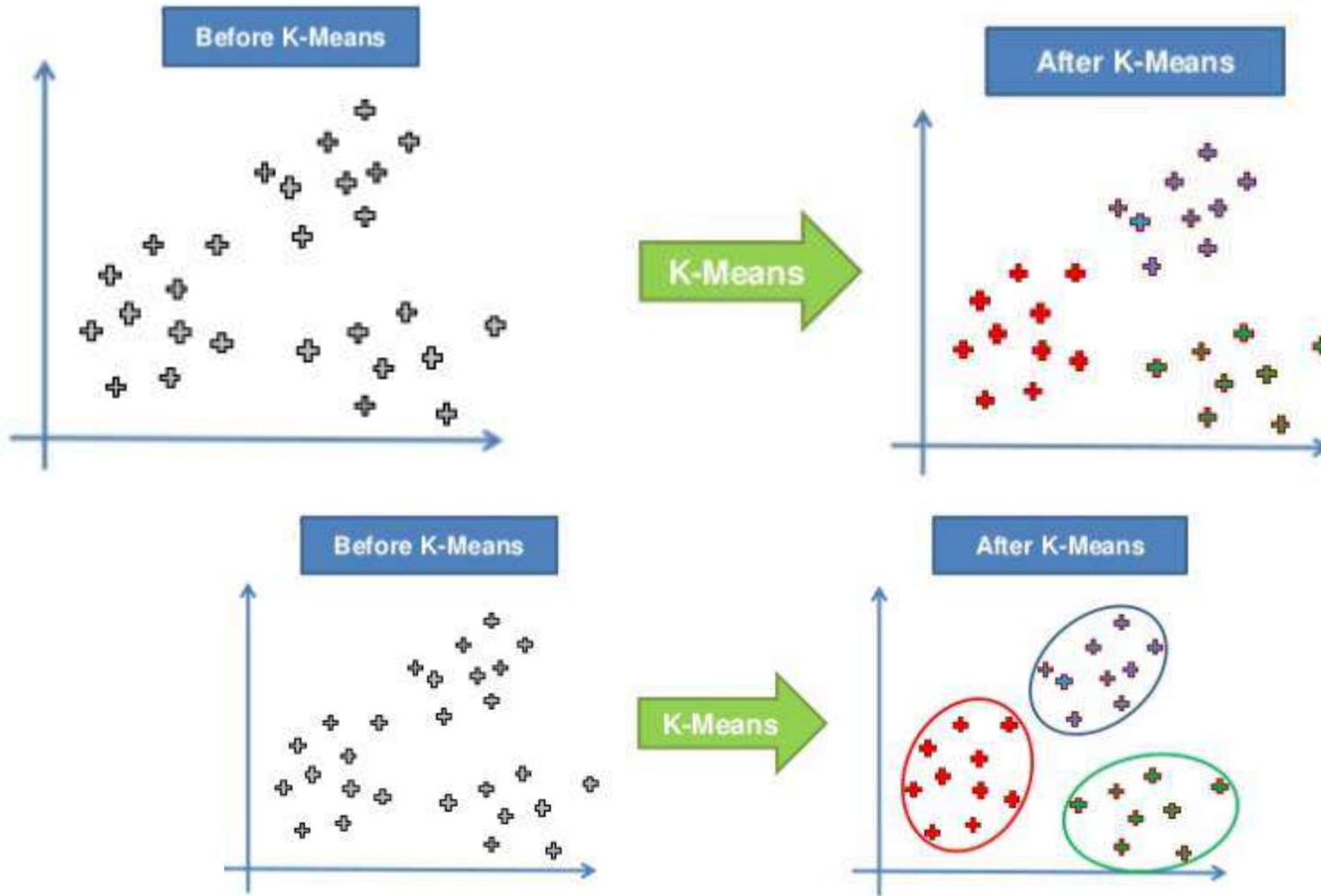


<https://medium.com/datadriveninvestor/clustering-algorithms-9fd35f34caa3>

K-Mean Clustering

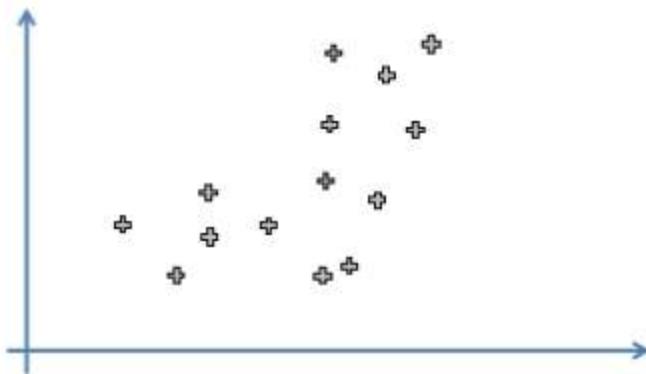
- K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups).
- The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K.
- The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

K-Mean Clustering

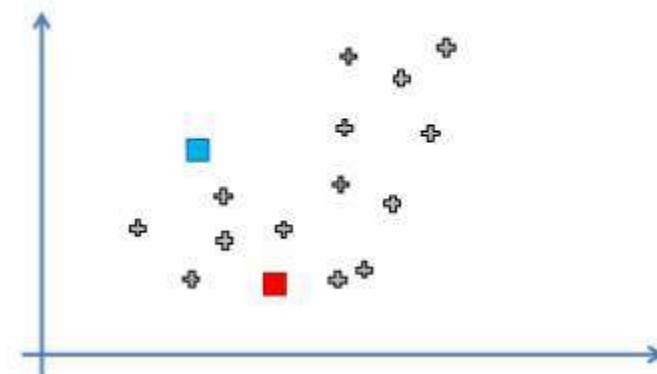


How K-Mean Clustering Work?

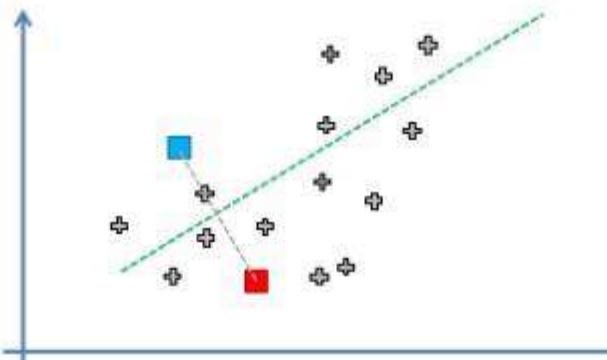
STEP 1: Choose the number K of clusters: K = 2



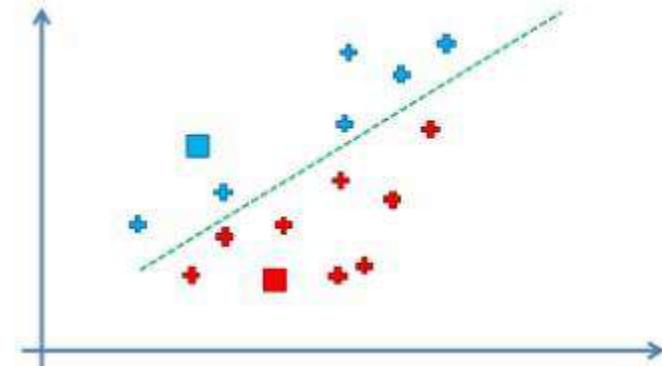
STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



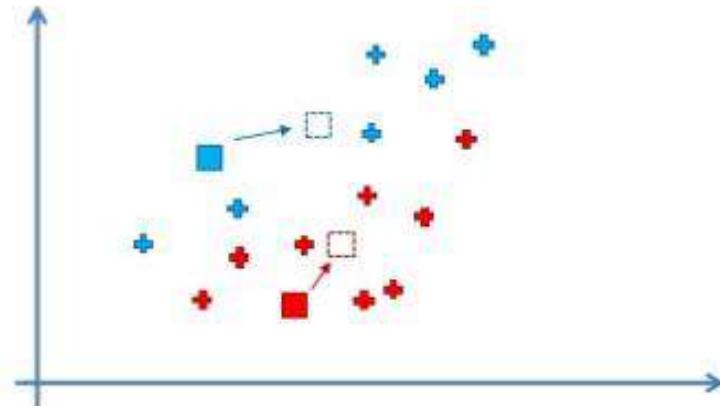
STEP 3: Assign each data point to the closest centroid



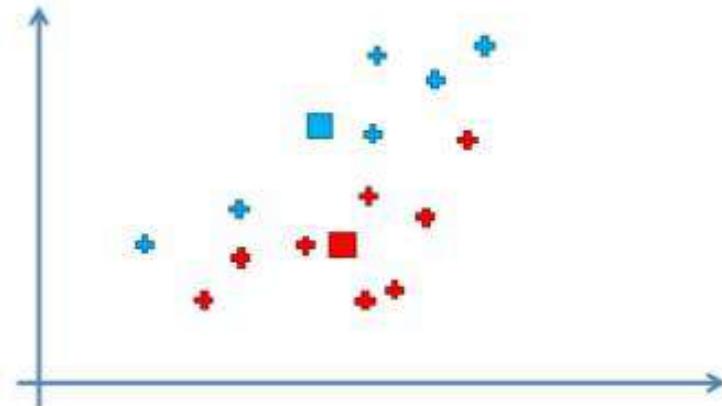
STEP 3: Assign each data point to the closest centroid



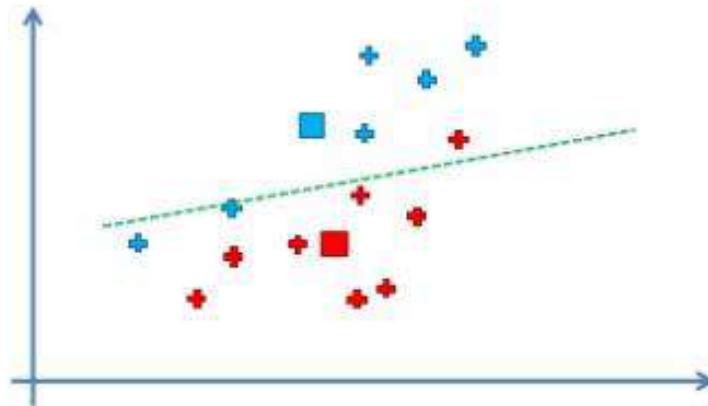
STEP 4: Compute and place the new centroid of each cluster



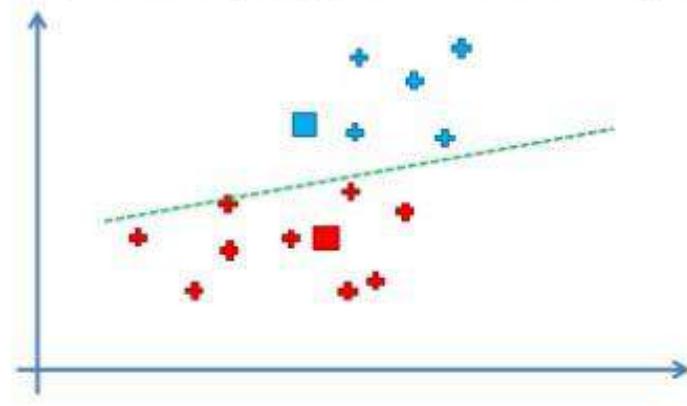
STEP 4: Compute and place the new centroid of each cluster



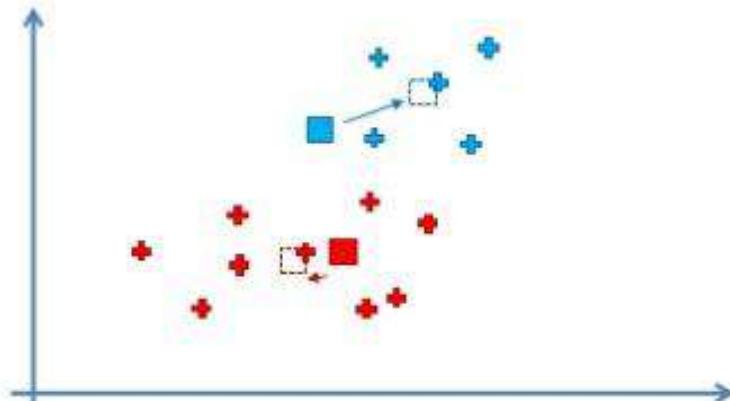
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



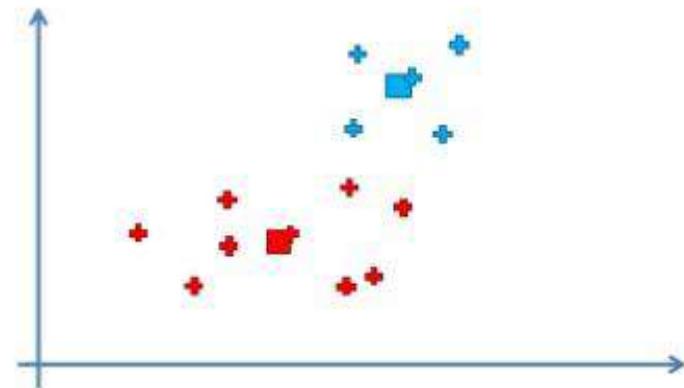
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



STEP 4: Compute and place the new centroid of each cluster

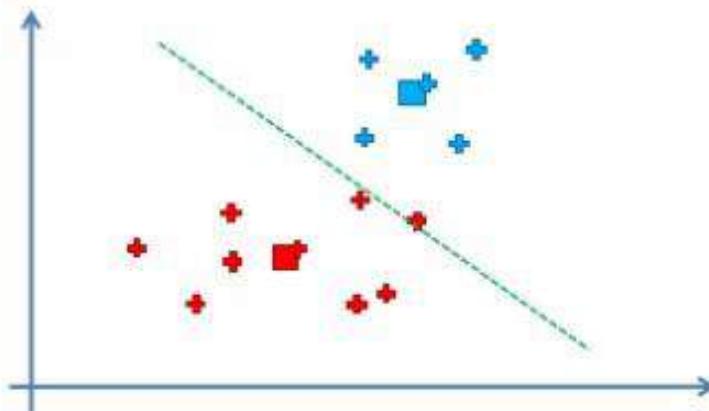


STEP 4: Compute and place the new centroid of each cluster

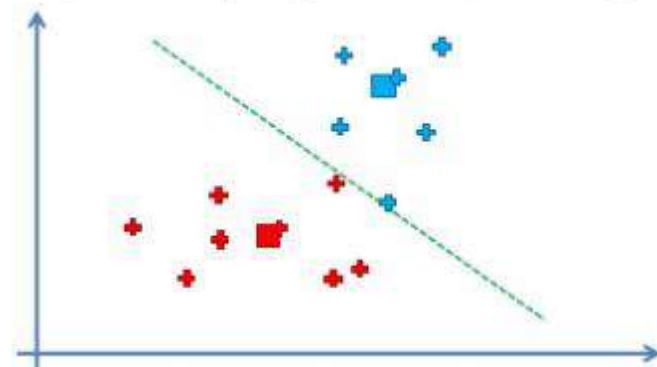


STEP 5: Reassign each data point to the new closest centroid.

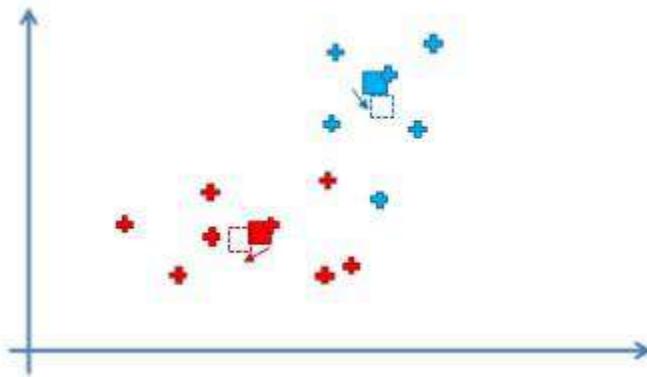
If any reassignment took place, go to STEP 4, otherwise go to FIN.



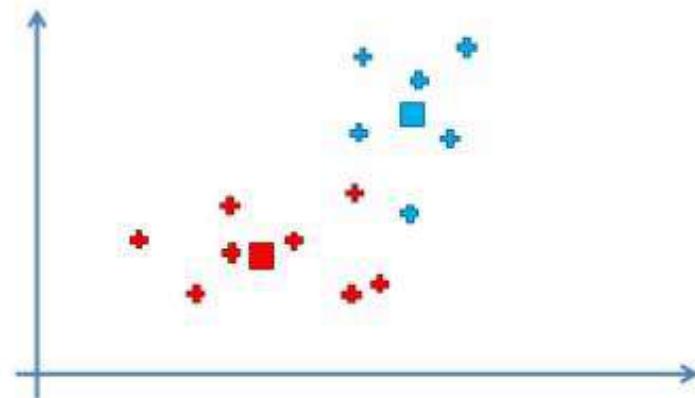
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



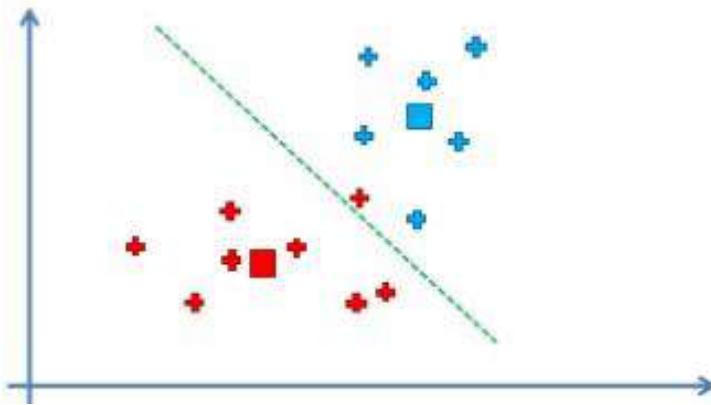
STEP 4: Compute and place the new centroid of each cluster



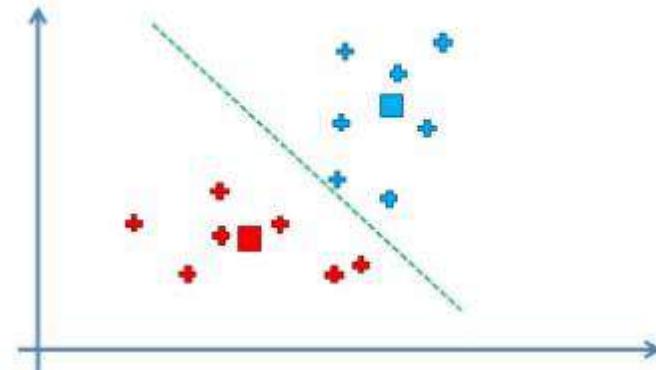
STEP 4: Compute and place the new centroid of each cluster



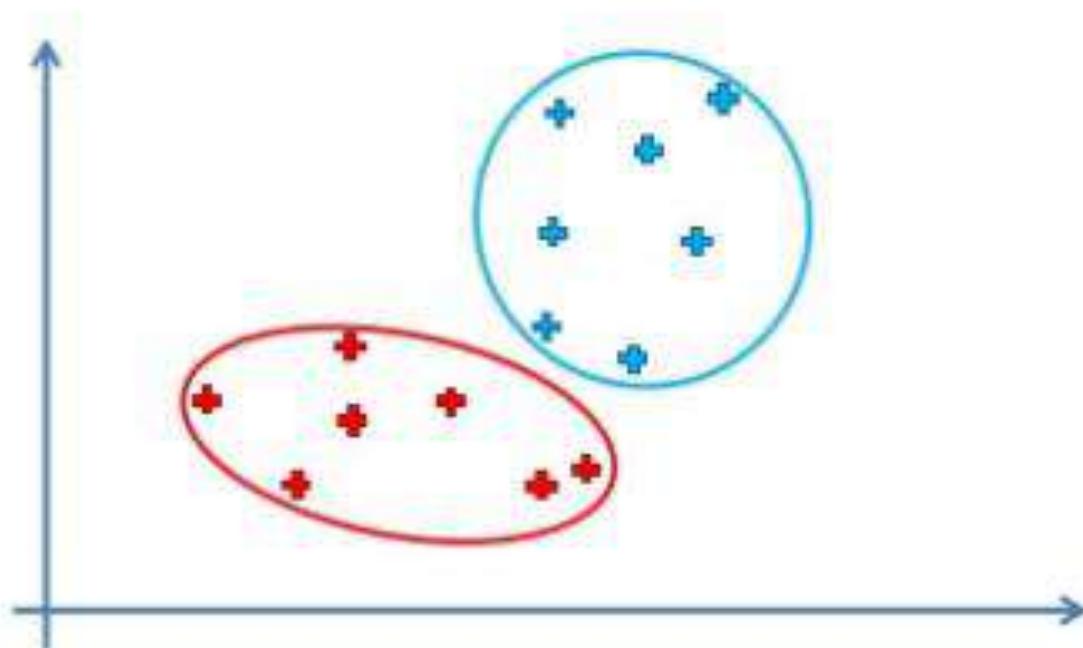
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

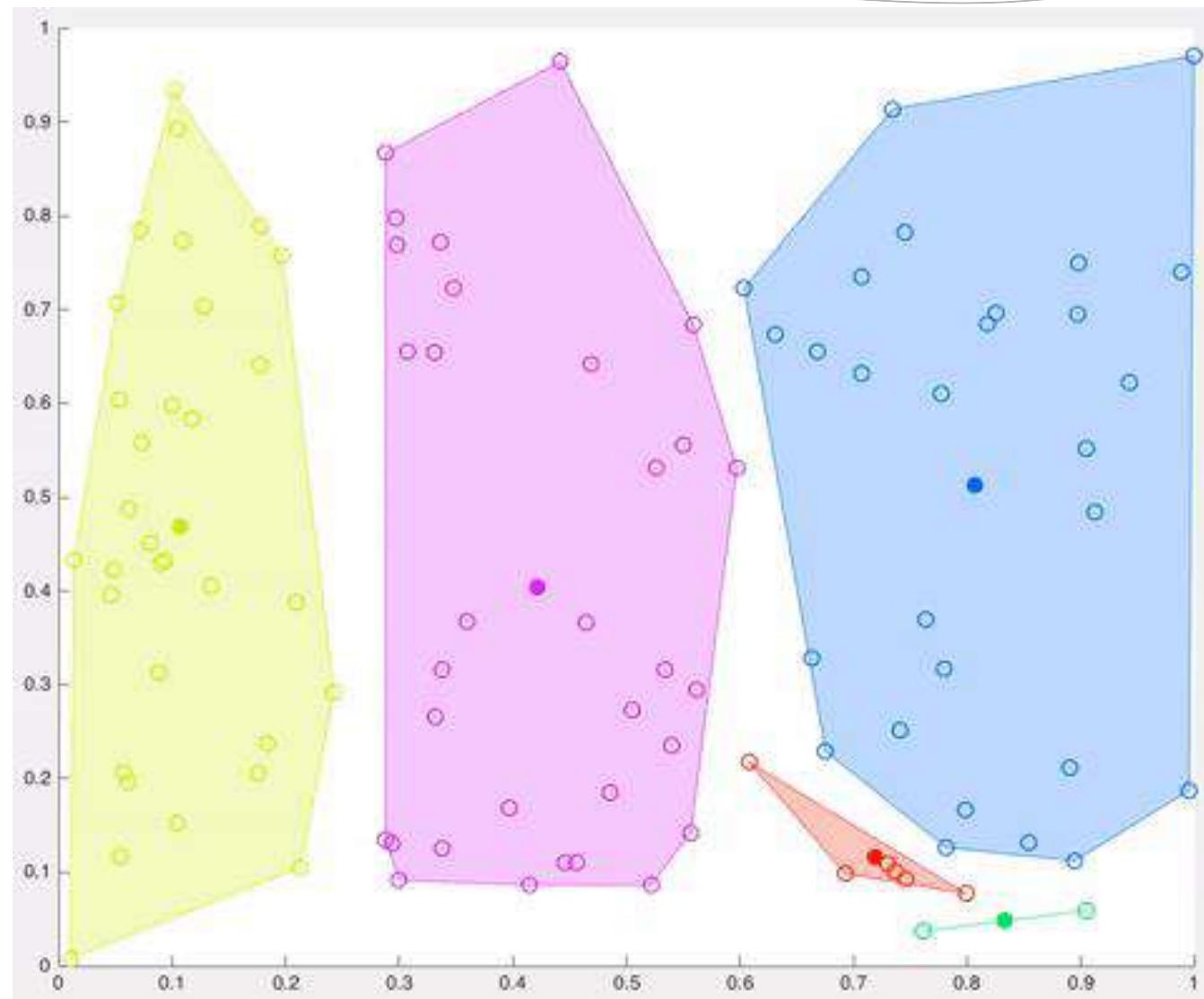


STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



FIN: Your Model Is Ready





<https://www.kdnuggets.com/2019/05/guide-k-means-clustering-algorithm.html>

K-Mean Clustering Visualization

<https://www.youtube.com/watch?v=5I3Ei69I4os>

K-Mean Clustering Algorithm

Step 1: Choose number of cluster (k)

Step 2: Randomly select k points.

Step 3: Assign each data point to the closest centroid.

Step 4: Compute new centroid and move point to the new centroid location.

Step 5: Reassigning each data point to the new closest centroid. If any reassignment, go to step 4 else goto finish.

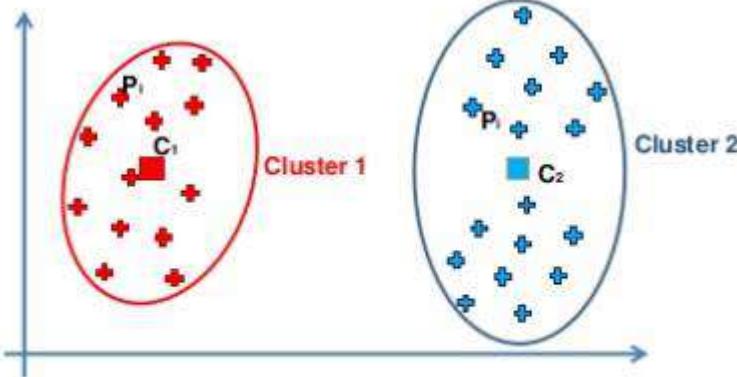
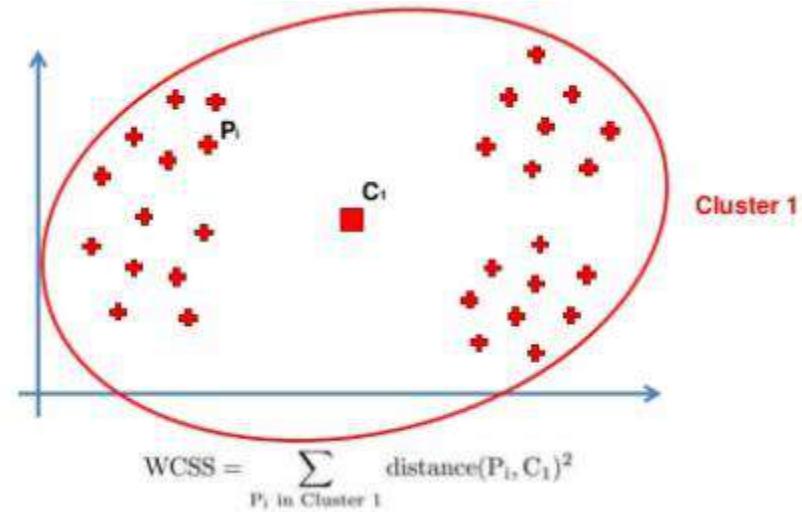
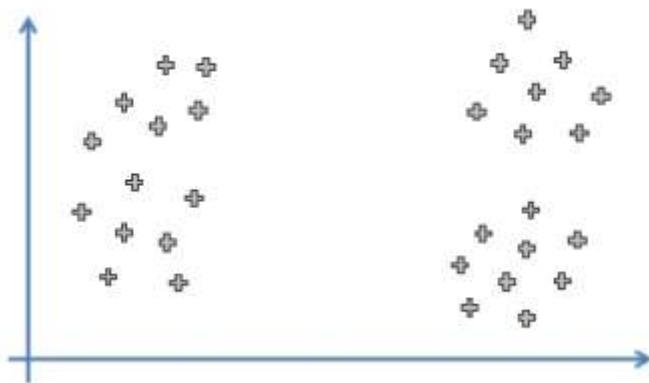
Stopping Condition for K-Mean Clustering

- 1) Centroids of newly formed clusters do not change
- 2) Points remain in the same cluster
- 3) Maximum number of iterations are reached

How to Find Optimal Number of Clusters

Elbow Method to Find Optimal Number of Clusters

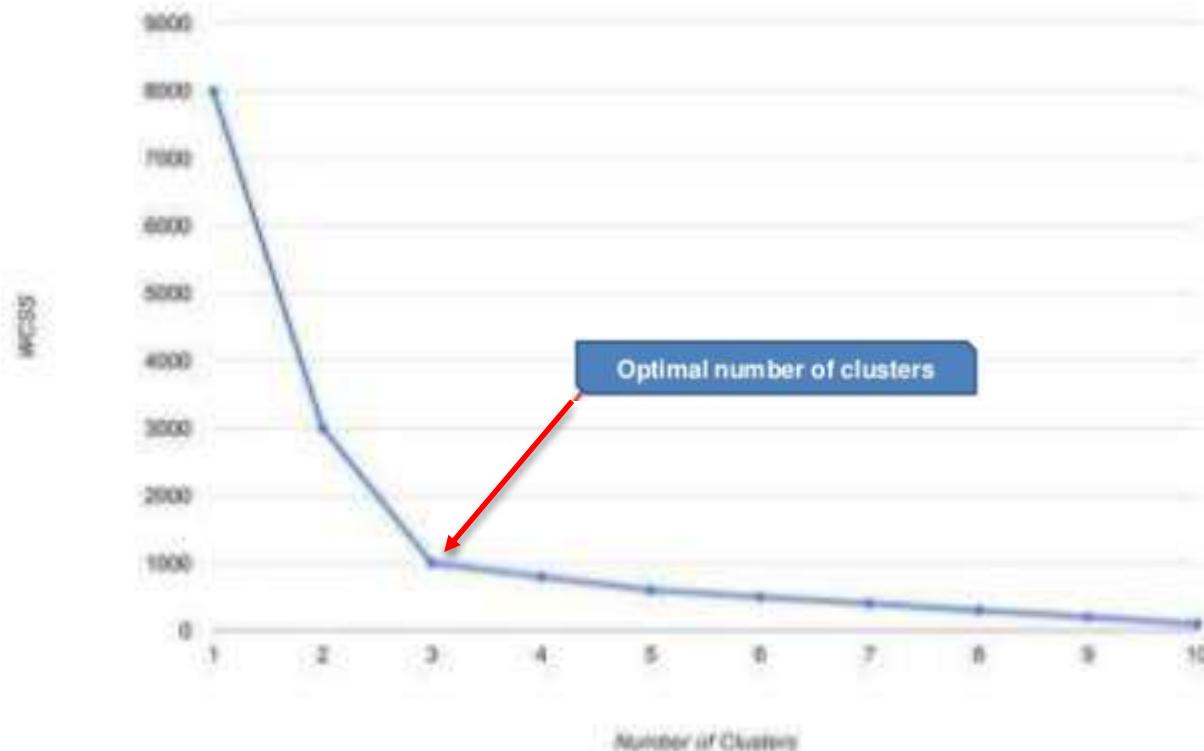
- 1) Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
- 2) For each k, calculate the total within-cluster sum of square (WCSS).
- 3) Plot the curve of WCSS according to the number of clusters k.
- 4) The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2$$

$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

The Elbow Method



<https://www.slideshare.net/KirillEremenko/deep-learning-az-self-organizing-maps-som-kmeans-clustering-part-3>

Silhouette Algorithm

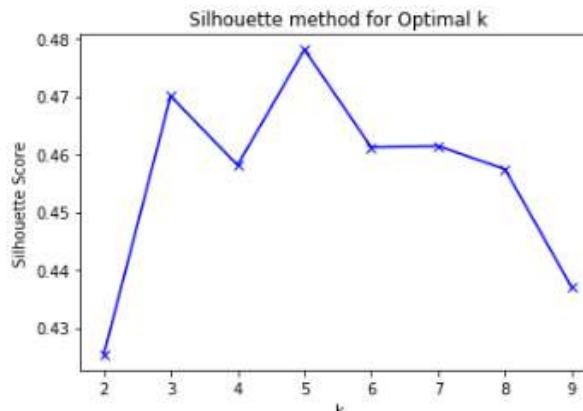
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- The silhouette ranges from -1 to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.
- A value close to **1 implies** that the instance is close to its cluster is a part of the **right cluster**. Whereas, a value close to **-1 means** that the value is assigned to the **wrong cluster**. Whereas **0 indicate** the point is on **boundary**.
- If most objects have a high value, then the clustering configuration is appropriate.
- If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

- It is calculated for each instance of dataset.

Silhouette Coefficient = $(x-y)/ \max(x,y)$

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$$

- where, y is the mean **intra cluster distance**: mean distance to the other instances in the same cluster. x depicts **mean nearest cluster distance** i.e. mean distance to the instances of the next closest cluster.



- $k=5$ should be chosen for the number of clusters.
- This method is better as it makes the decision regarding the optimal number of clusters more meaningful and clear. But this metric is computation expensive as the coefficient is calculated for every instance.

- In the Silhouette algorithm, we assume that the data has already been clustered into k clusters by a clustering technique(Typically K-Means Clustering technique).

```
Cluster 1 ={{1,0},{1,1}}
```

```
Cluster 2 ={{1,2},{2,3},{2,2},{1,2}},
```

```
Cluster 3 ={{3,1},{3,3},{2,1}}}
```

Take a point `{1,0} in cluster 1`

Calculate its **average distance** to all other points in **it's cluster**, i.e. cluster 1

```
So a1 = $\sqrt{(1-1)^2 + (0-1)^2} = \sqrt{0+1} = \sqrt{1} = 1$ 
```

Now for the object {1,0} in cluster 1 calculate its average distance from all the objects in cluster 2 and cluster 3. Of these take the **minimum** average distance.

So for cluster 2

$$\{1,0\} \rightarrow \{1,2\} = \text{distance} = \sqrt{(1-1)^2 + (0-2)^2} = \sqrt{0+4} = \sqrt{4} = 2$$

$$\{1,0\} \rightarrow \{2,3\} = \text{distance} = \sqrt{(1-2)^2 + (0-3)^2} = \sqrt{1+9} = \sqrt{10} = 3.16$$

$$\{1,0\} \rightarrow \{2,2\} = \text{distance} = \sqrt{(1-2)^2 + (0-2)^2} = \sqrt{1+4} = \sqrt{5} = 2.24$$

$$\{1,0\} \rightarrow \{1,2\} = \text{distance} = \sqrt{(1-1)^2 + (0-2)^2} = \sqrt{0+4} = \sqrt{4} = 2$$

Therefore, the average distance of point {1,0} in cluster 1 to all the points in cluster 2 =

$$(2+3.16+2.24+2)/4 = 2.325$$

Similarly, for cluster 3

$$\{1,0\} \rightarrow \{3,1\} = \text{distance} = \sqrt{(1-3)^2 + (0-1)^2} = \sqrt{4+1} = \sqrt{5} = 2.24$$

$$\{1,0\} \rightarrow \{3,3\} = \text{distance} = \sqrt{(1-3)^2 + (0-3)^2} = \sqrt{4+9} = \sqrt{13} = 3.61$$

$$\{1,0\} \rightarrow \{2,1\} = \text{distance} = \sqrt{(1-2)^2 + (0-1)^2} = \sqrt{1+1} = \sqrt{2} = 1.41$$

Therefore, the **average distance** of point {1,0} in cluster 1 to all the points in cluster 3 =

$$(2.24+3.61+1.41)/3 = 2.4$$

Now, the **minimum** average distance of the point {1,0} in cluster 1 to the other clusters 2 and 3 is,

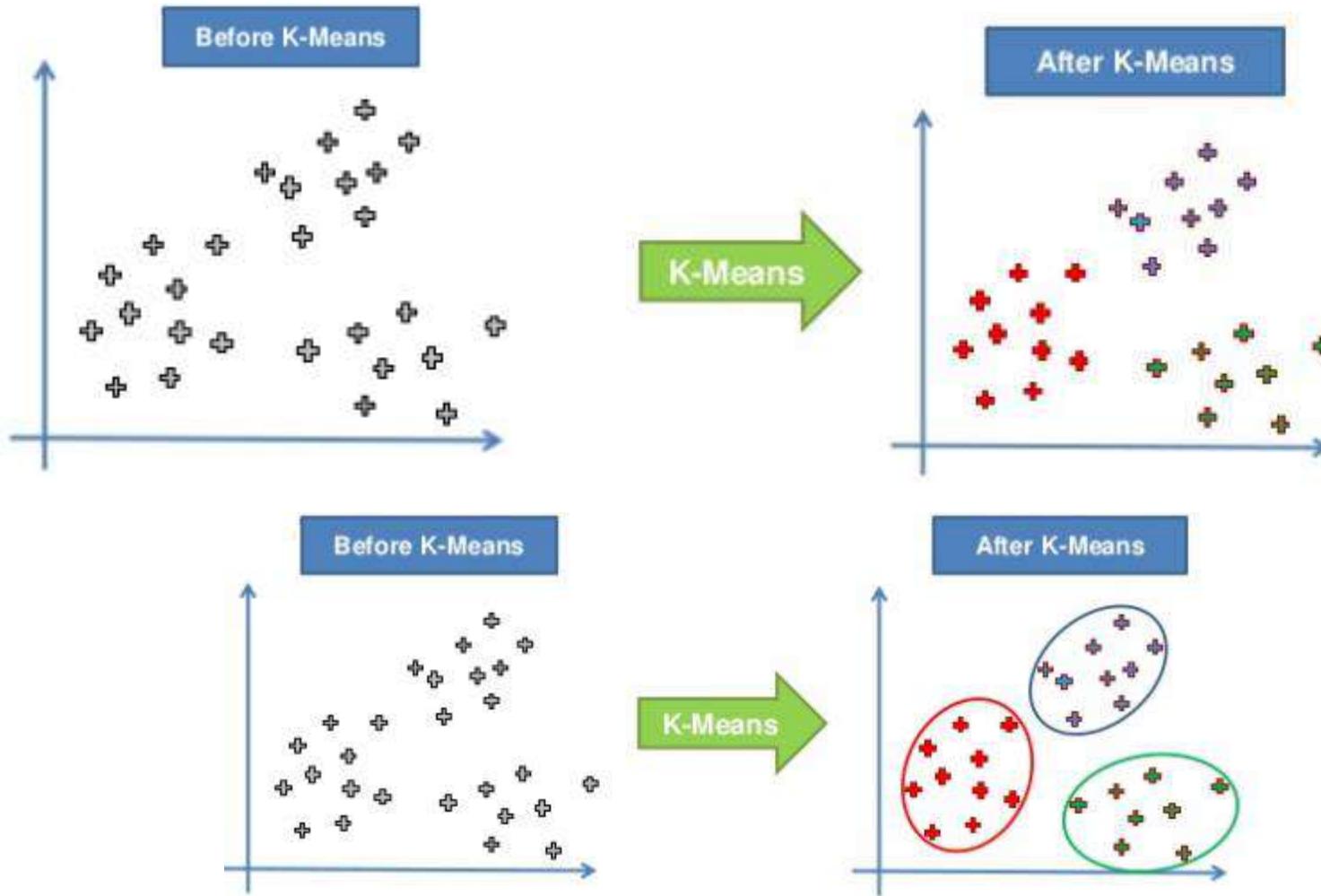
$$2.325$$

So the silhouette coefficient of cluster 1

$$s1 = 1 - (a1/b1) = 1 - (1/2.325) = 1 - 0.4301 = 0.5699$$

Random Initialization Trap

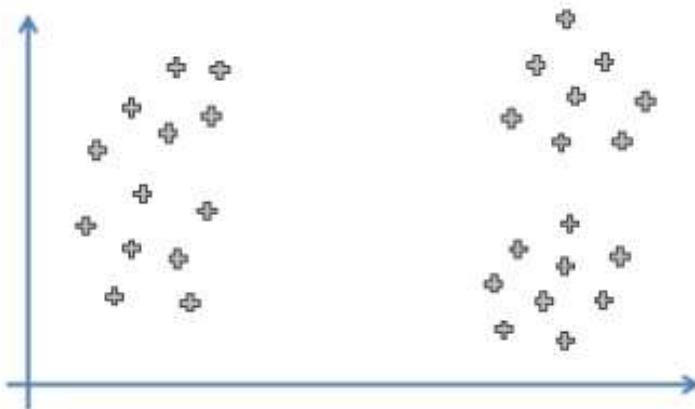
K-Mean Clustering



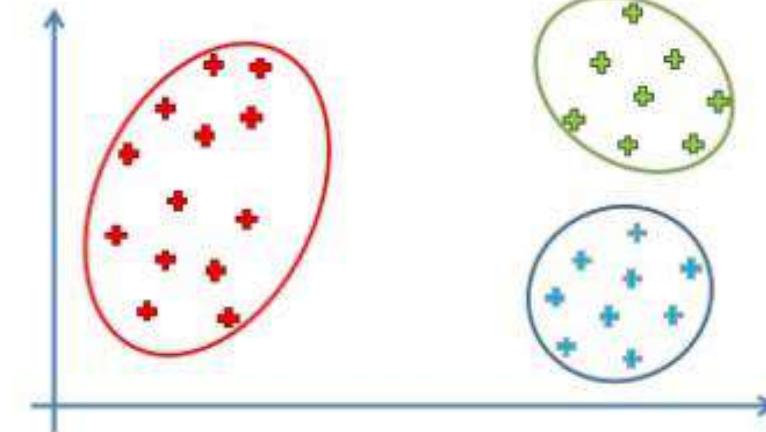
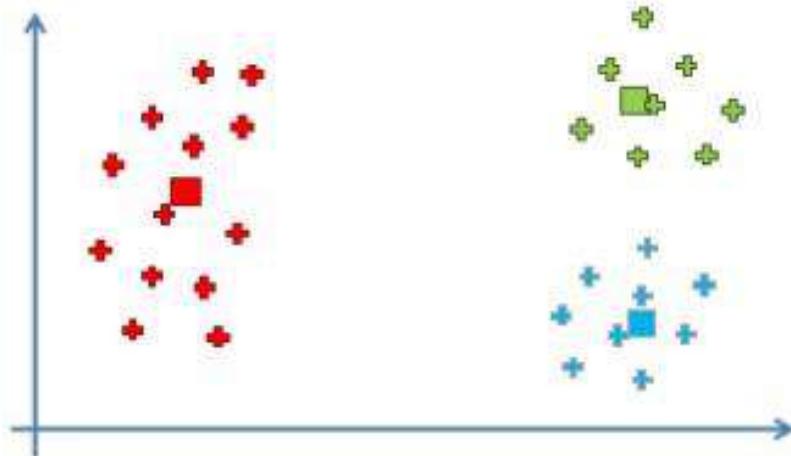
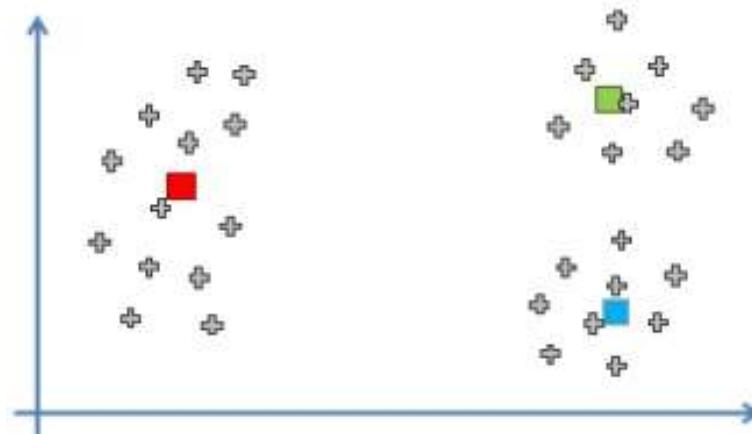
Random Initialization Trap

- Random initialization trap is a problem that can be occurred due to the random initialization of centroids in K mean clustering.
- Random initialization trap sometimes lead to generating wrong clusters in the dataset.

If random points are correctly placed

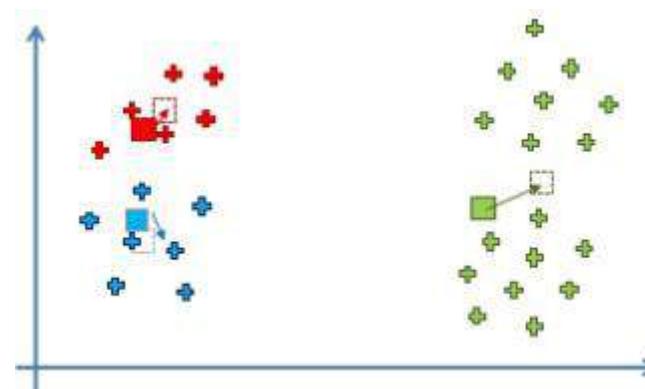
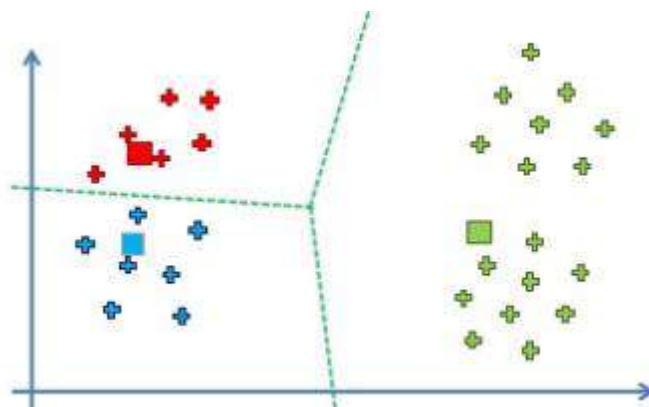
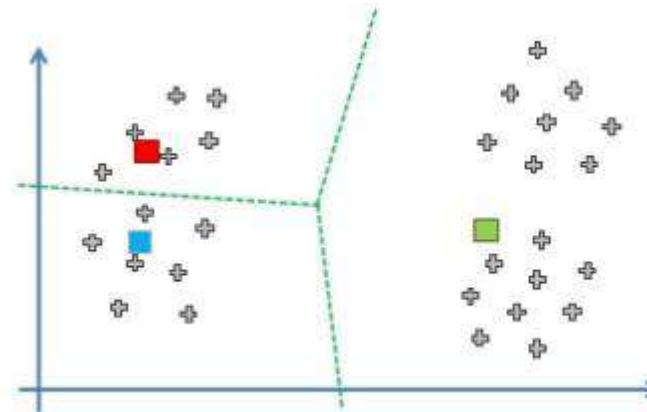
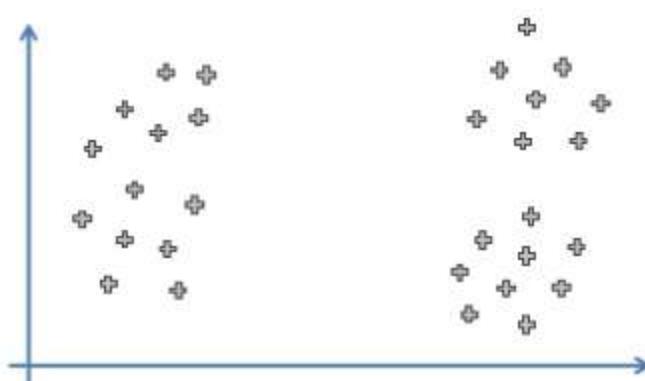


If we choose $K = 3$ clusters...

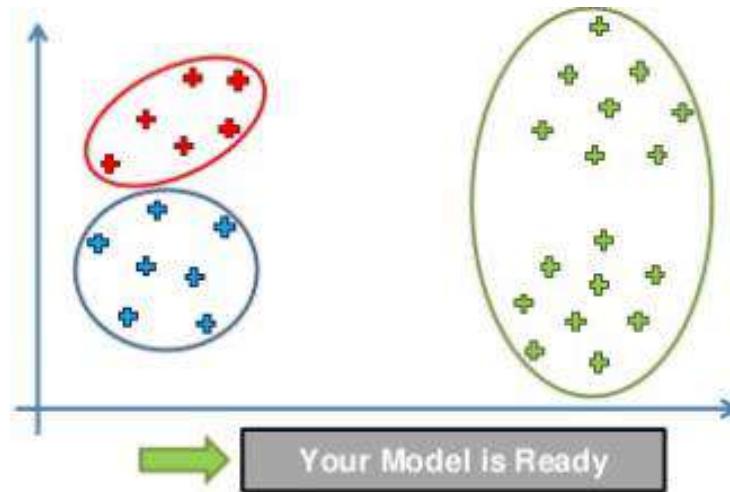
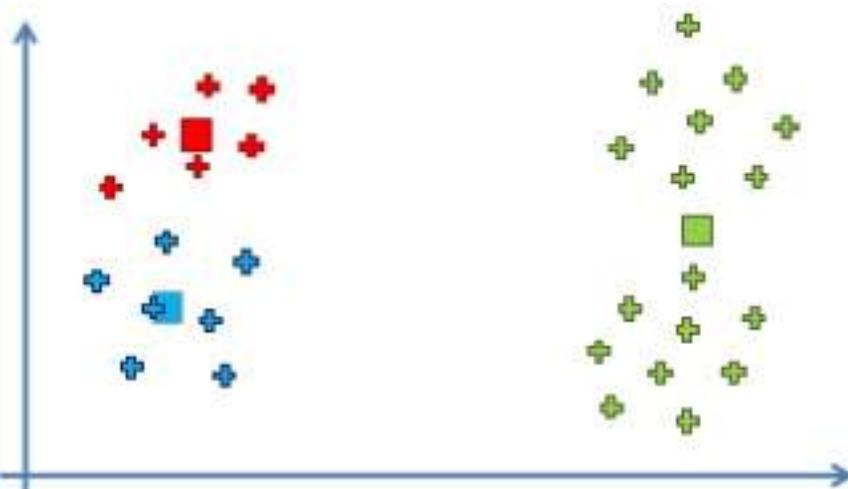
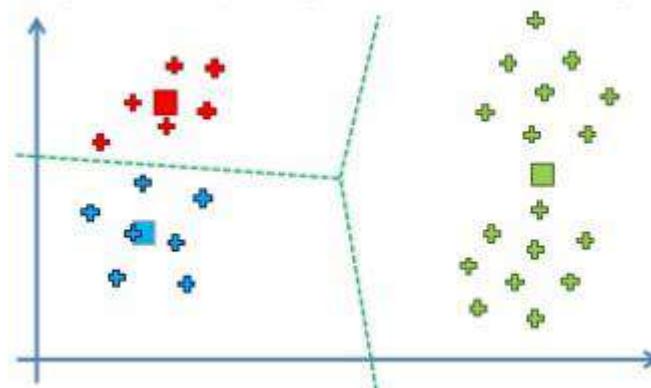
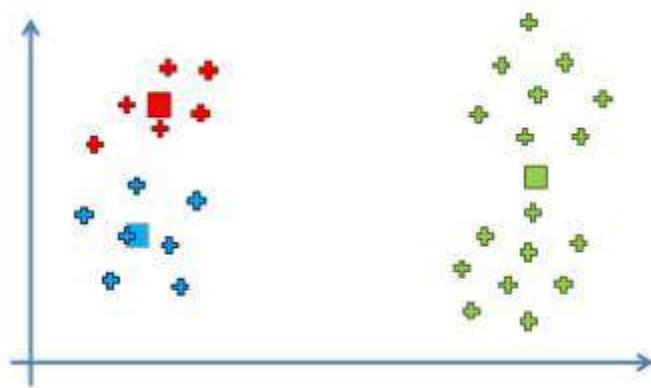


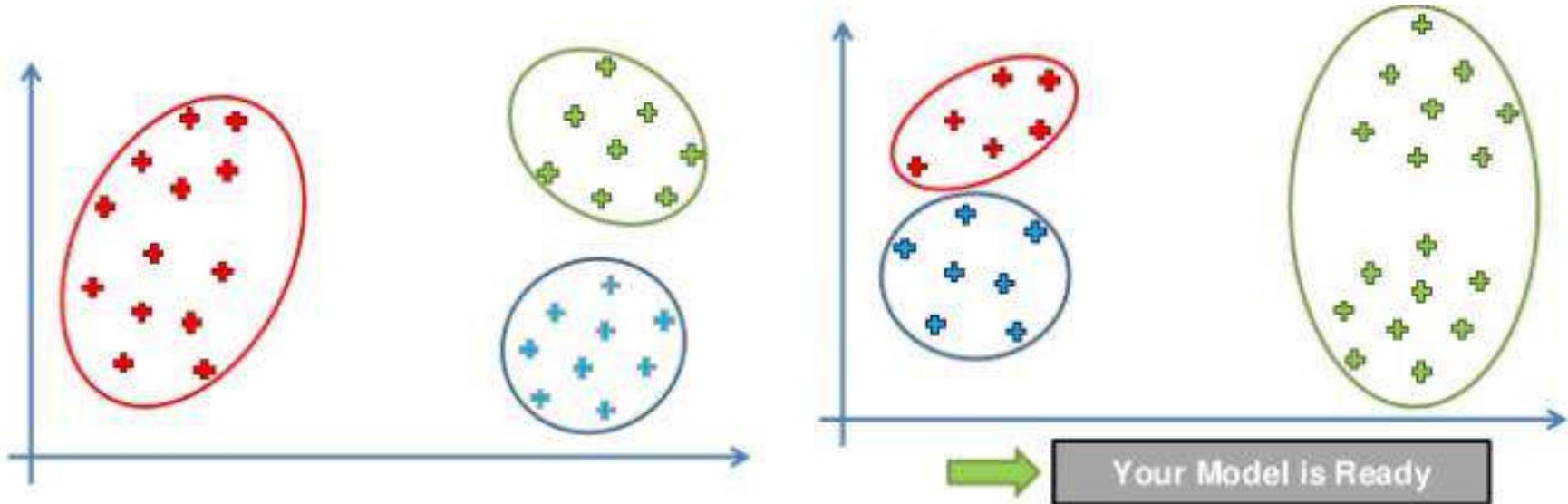
If random points are wrongly placed

STEP 1: Choose the number K of clusters: K = 3



<https://www.slideshare.net/KirillEremenko/deep-learning-az-self-organizing-maps-som-kmeans-clustering-part-2>





- K-means++ is a smart **centroid initialization** method for the K-mean algorithm. The goal is to spread out the initial centroid by assigning the **first centroid randomly** then selecting the rest of the centroids based on the maximum squared distance.

Limitations of K Mean Clustering

- We have to decide the **number of clusters** at the *beginning* of the algorithm. Ideally, we would not know how many clusters should we have, in the beginning of the algorithm and hence it a challenge with K-means.
- Being dependent on **initial values**.
- Centroids can be dragged by **outliers**, or outliers might get their own cluster instead of being ignored.
- It always tries to make clusters of the **same size**.

Understanding the Working of K-Mean Clustering using Numerical

How K-Mean Clustering Work

Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

$$n = 19$$

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

Assume

$$k = 2$$

$$\begin{aligned}c_1 &= 16 \\c_2 &= 22\end{aligned}$$

Distance functions

Euclidean $\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$

Manhattan $\sum_{i=1}^k |x_i - y_i|$

Minkowski $\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$

Iteration 1:

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	
15	16	22	1	7	1	
16	16	22	0	6	1	
19	16	22	3	3	2	
19	16	22	3	3	2	
20	16	22	4	2	2	
20	16	22	4	2	2	
21	16	22	5	1	2	
22	16	22	6	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

$$c_1 = 15.33$$

$$c_2 = 36.25$$

Source: https://www.saedsayad.com/clustering_kmeans.htm

Iteration 2:

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	18.56
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	
35	15.33	36.25	19.67	1.25	2	45.9
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

$$c_1 = 18.56$$

$$c_2 = 45.90$$

Source: https://www.saedsayad.com/clustering_kmeans.htm

Iteration 3:

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	19.50
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

$$c_1 = 19.50$$

$$c_2 = 47.89$$

Source: https://www.saedsayad.com/clustering_kmeans.htm

Iteration 4:

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	19.50
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

$$c_1 = 19.50$$

$$c_2 = 47.89$$

Source: https://www.saedsayad.com/clustering_kmeans.htm

Example-2

- Cluster the following eight points (with (x, y) representing locations) into three clusters:
A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

$$P(a, b) = |x_2 - x_1| + |y_2 - y_1|$$

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2)

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

Source: <https://www.gatevidyalay.com/tag/k-means-clustering-numerical-example-pdf/>

Cluster-01:

First cluster contains points-

- A1(2, 10)

Cluster-02:

Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

Cluster-03:

Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

For Cluster-01:

- We have only one point A1(2, 10) in Cluster-01.
- So, cluster center remains the same.

For Cluster-02:

Center of Cluster-02

$$= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)$$

$$= (6, 6)$$

For Cluster-03:

Center of Cluster-03

$$= ((2 + 1)/2, (5 + 2)/2)$$

$$= (1.5, 3.5)$$

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

Source: <https://www.gatevidyalay.com/tag/k-means-clustering-numerical-example-pdf/>

Cluster-01:

First cluster contains points-

- A1(2, 10)
- A8(4, 9)

Cluster-02:

Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)

Cluster-03:

Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

For Cluster-01:

Center of Cluster-01

$$= ((2 + 4)/2, (10 + 9)/2)$$

$$= (3, 9.5)$$

For Cluster-02:

Center of Cluster-02

$$= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)$$

$$= (6.5, 5.25)$$

For Cluster-03:

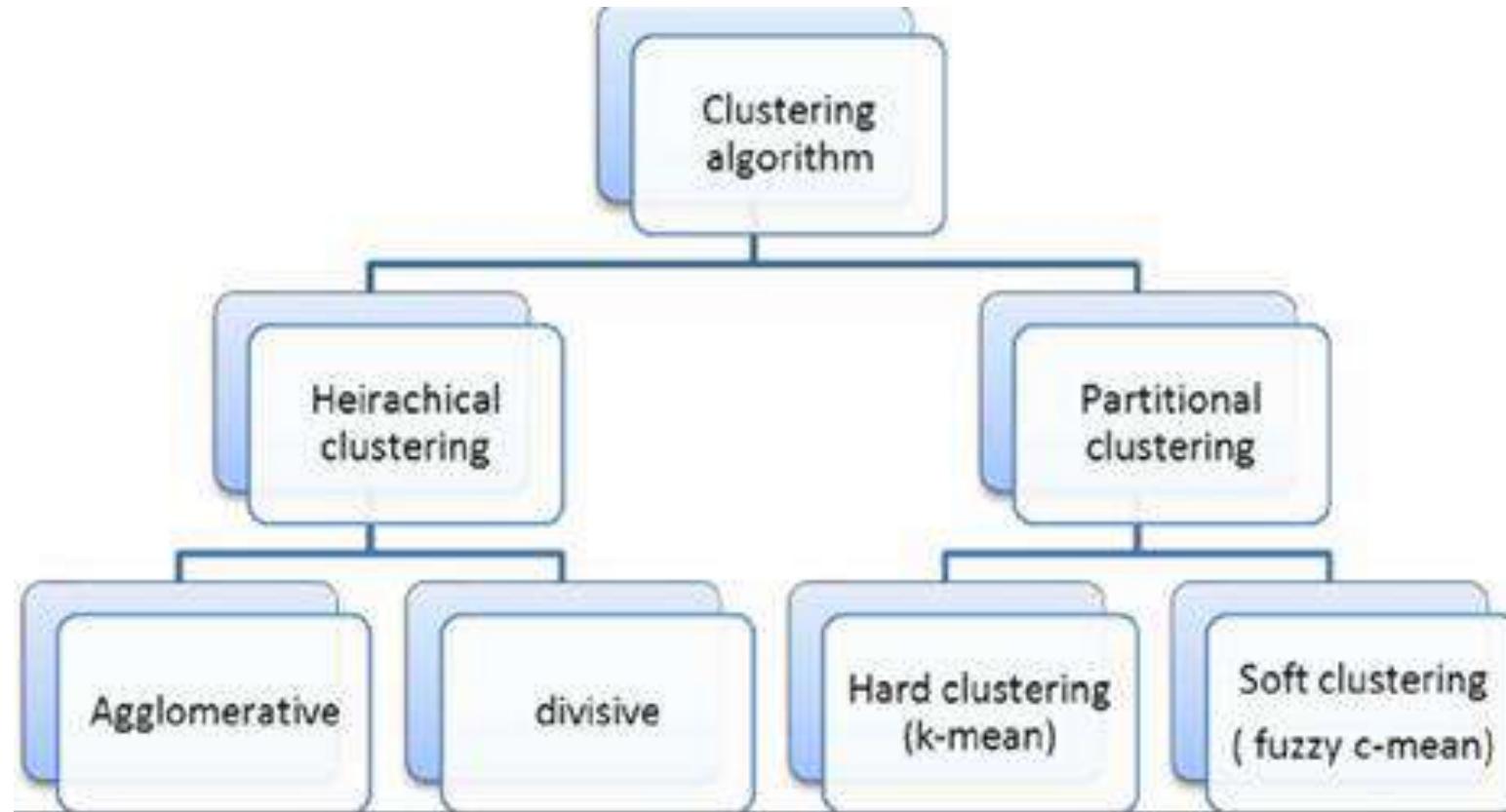
Center of Cluster-03

$$= ((2 + 1)/2, (5 + 2)/2)$$

$$= (1.5, 3.5)$$

Agglomerative Hierarchical Clustering

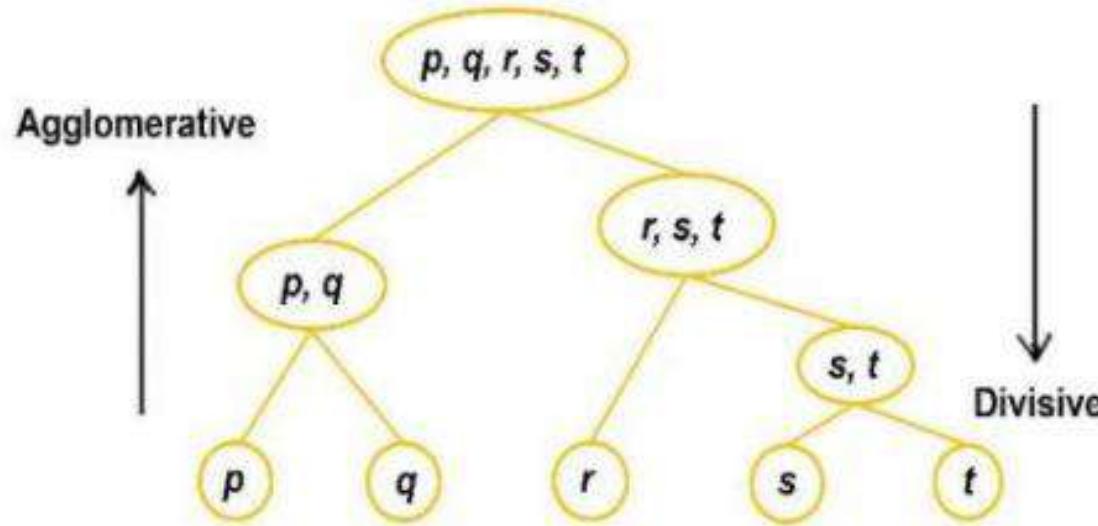
Types of Clustering Algorithm



<https://medium.com/datadriveninvestor/clustering-algorithms-9fd35f34caa3>

Agglomerative and Divisive

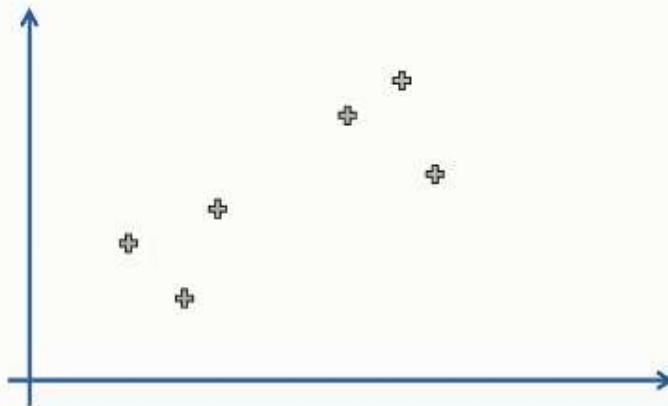
Agglomerative: This is a "bottom-up" approach: each observation starts in its own **cluster**, and pairs of **clusters** are merged as one moves up the hierarchy.



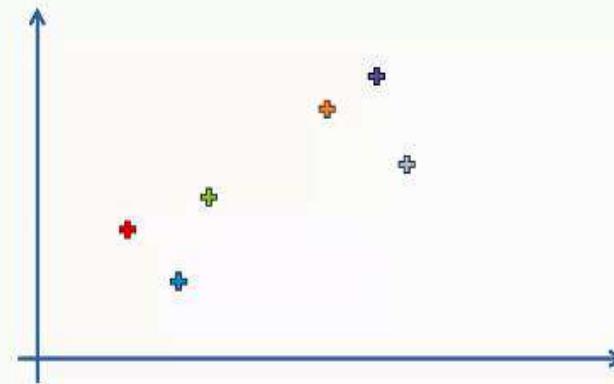
Agglomerative Hierarchical Clustering

- We assign each point to an individual cluster in this technique. Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left.
- It is a bottom-up approach.
- In this approach we are merging (or adding) the clusters at each step. So it is also known as **additive hierarchical clustering**.
- This clustering algorithm does not require us to prespecify the number of clusters.

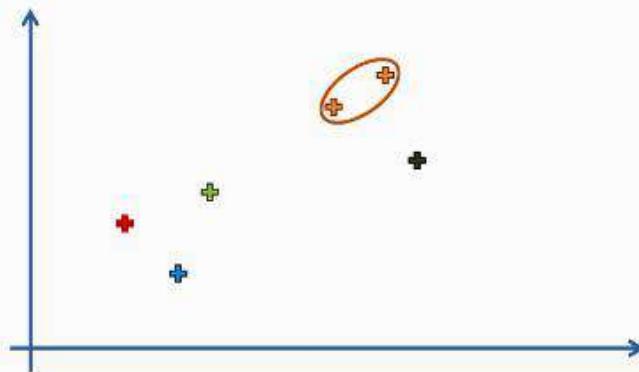
Consider the following dataset of $N = 6$ data points



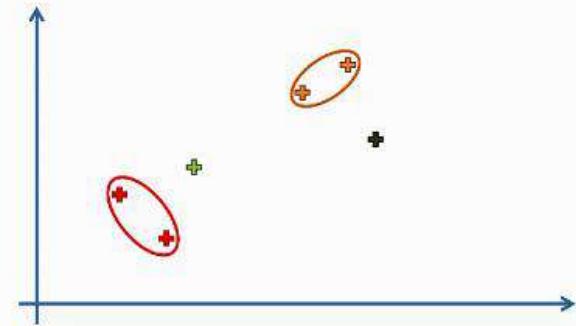
STEP 1: Make each data point a single-point cluster → That forms 6 clusters



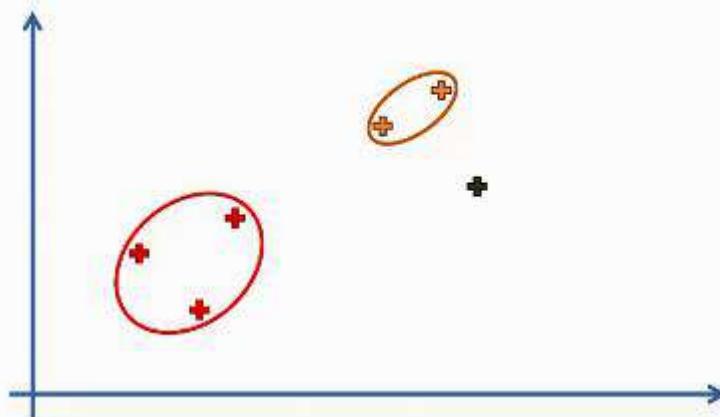
STEP 2: Take the two closest data points and make them one cluster
→ That forms 5 clusters



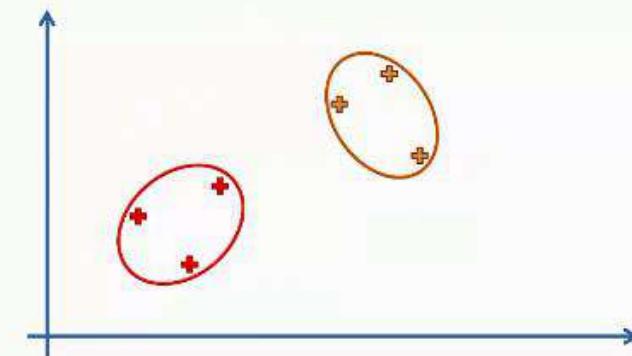
STEP 3: Take the two closest clusters and make them one cluster
→ That forms 4 clusters



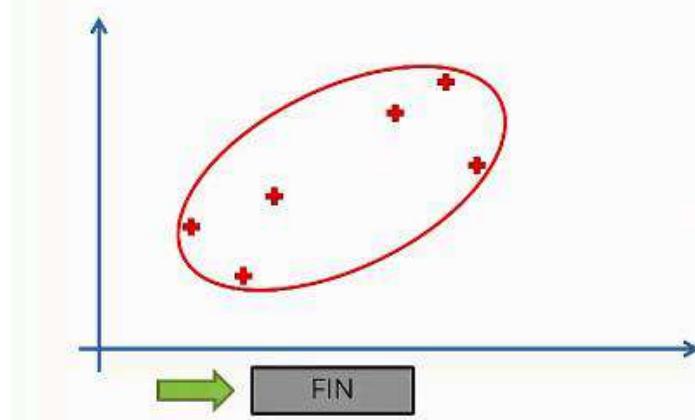
STEP 4: Repeat STEP 3 until there is only one cluster

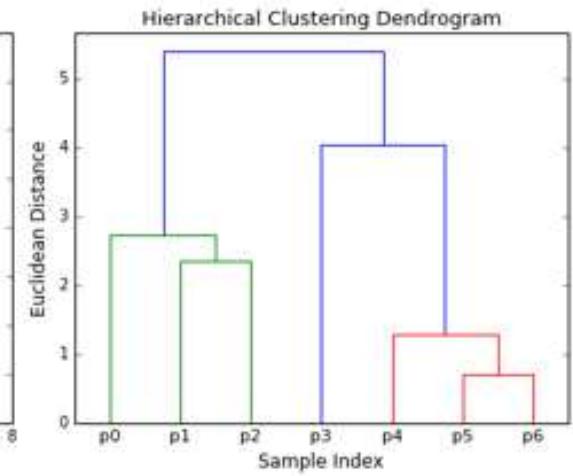
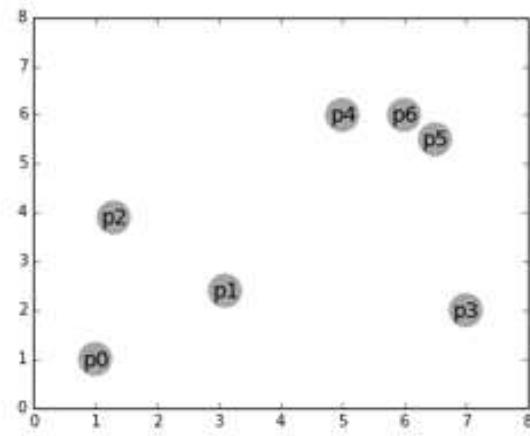
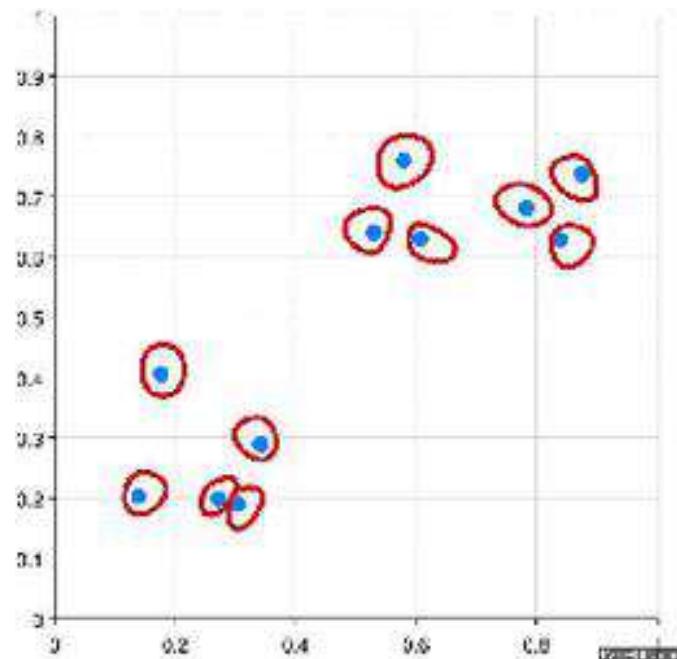


STEP 4: Repeat STEP 3 until there is only one cluster



STEP 4: Repeat STEP 3 until there is only one cluster





<https://www.kdnuggets.com/2019/09/hierarchical-clustering.html>

- Suppose a teacher wants to divide her students into different groups. She has the marks scored by each student in an assignment and based on these marks, she wants to segment them into groups.

Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

Creating a Proximity Matrix

- First, we will create a proximity matrix which will tell us the distance between each of these points.
- Since we are calculating the distance of each point from each of the other points, we will get a square matrix of shape n X n (where n is the number of observations).
- Euclidean distance will be used to calculate distance between two points.

Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

$$\sqrt{(10-7)^2} = \sqrt{9} = 3$$

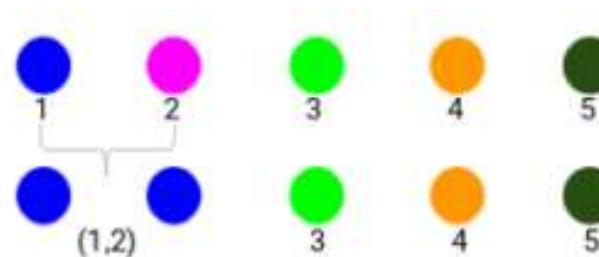
Steps to Perform Hierarchical Clustering

Step 1: First, we assign all the points to an individual cluster:



Step 2: Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. We then update the proximity matrix:

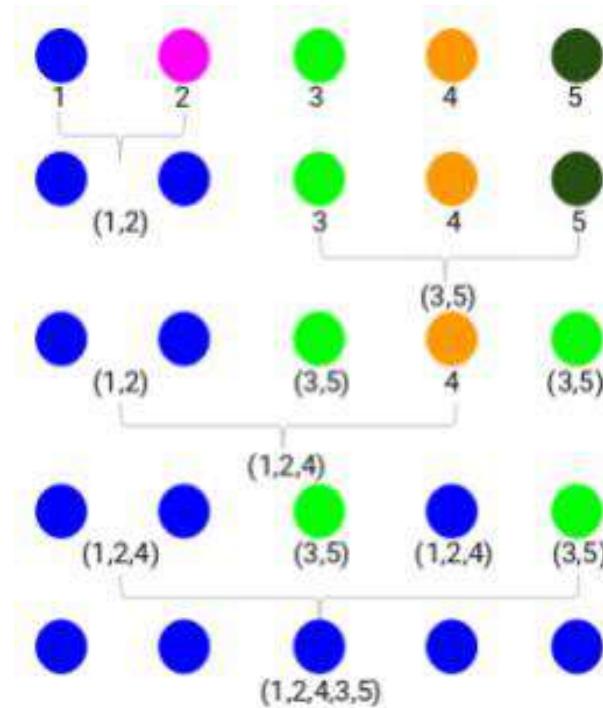
ID	1	2	3	4	5
1	0	(3)	18	10	25
2	(3)	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0



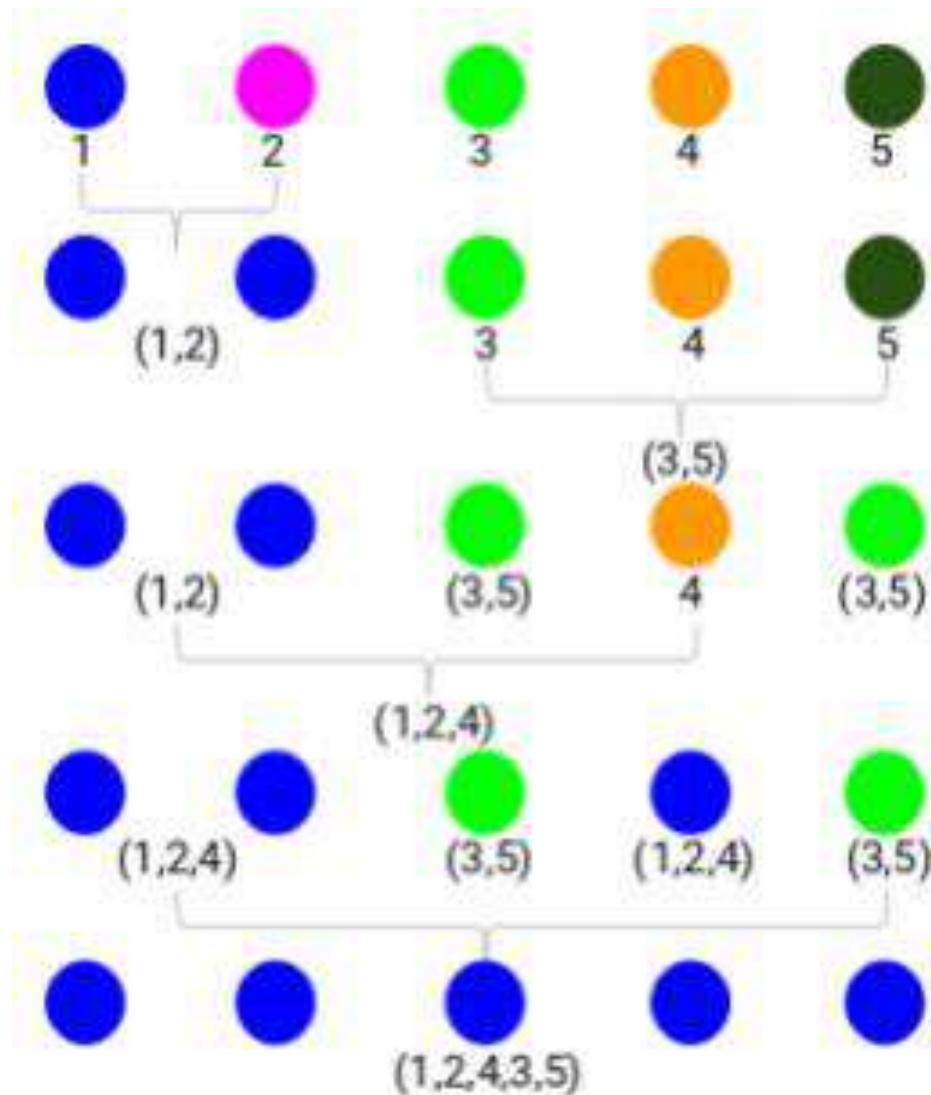
- Updated clusters and accordingly update the proximity matrix.
- Here, we have taken the maximum of the two marks (7, 10) to replace the marks for this cluster. Instead of the maximum, we can also take the minimum value or the average values as well.

ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

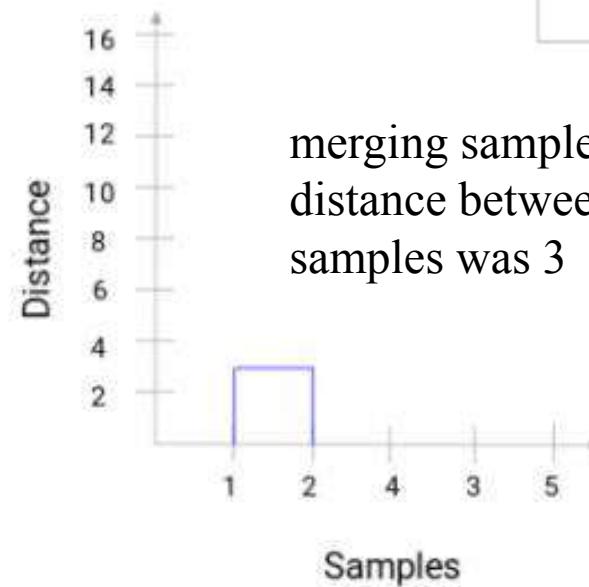
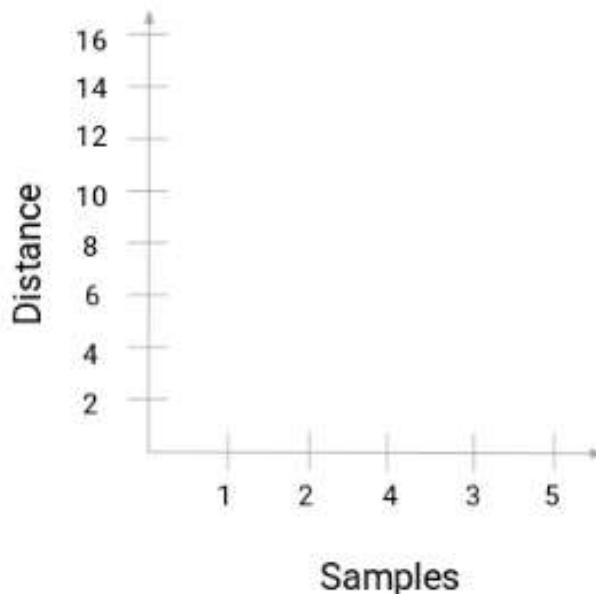


Choose the Optimal Number of Clusters in Hierarchical Clustering

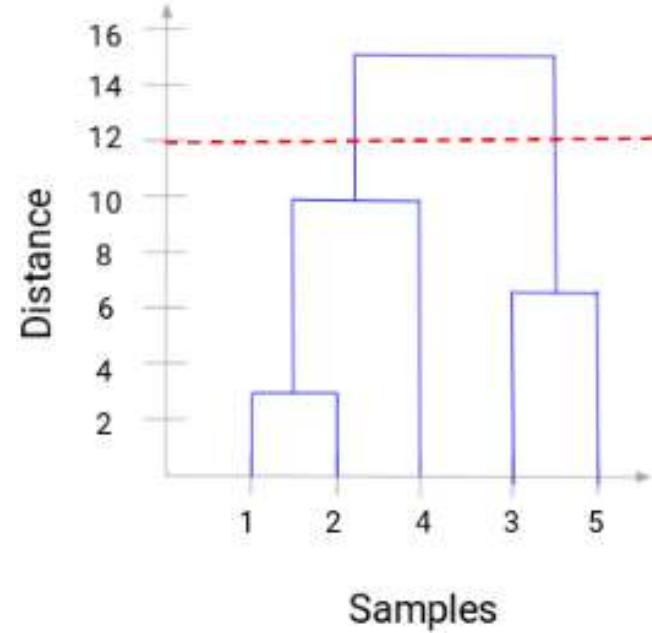
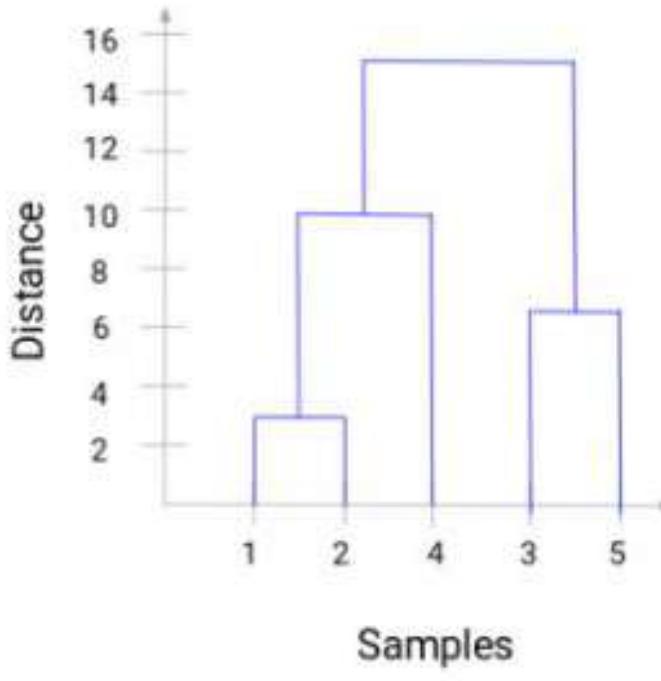


Choose the Number of Clusters in Hierarchical Clustering

- A dendrogram is a tree-like diagram that records the sequences of merges or splits.

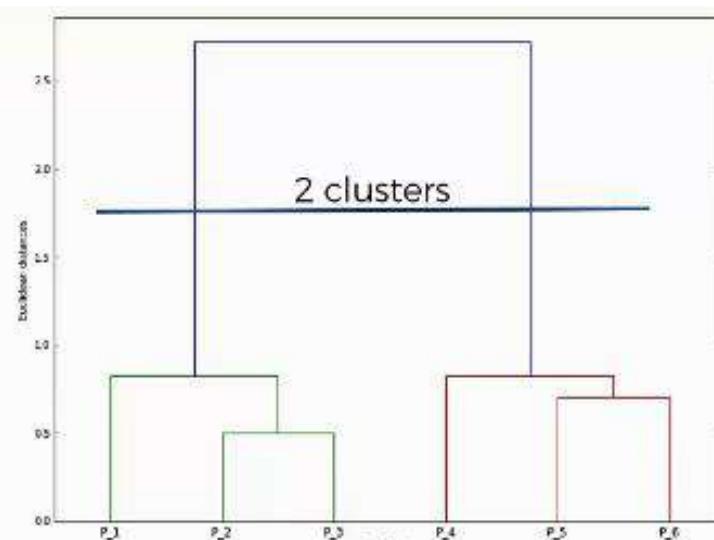
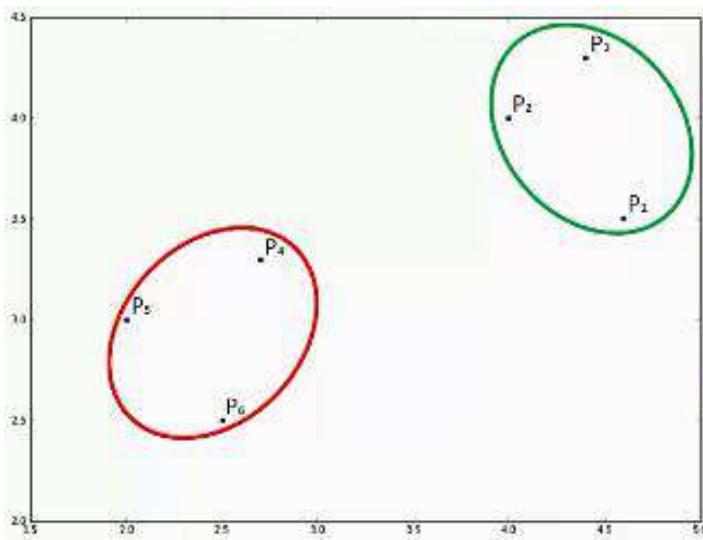
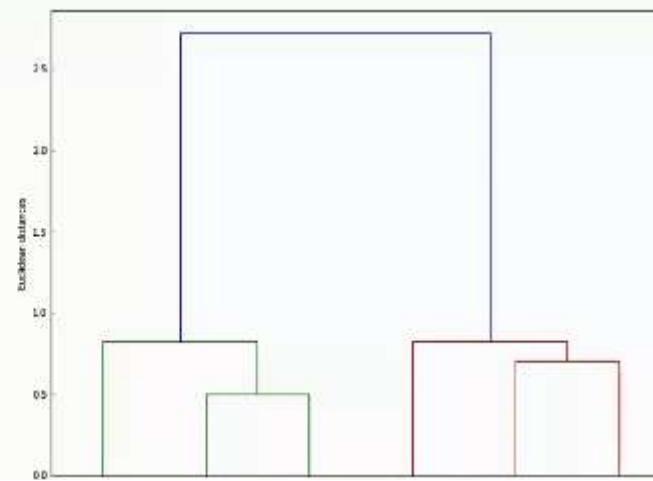
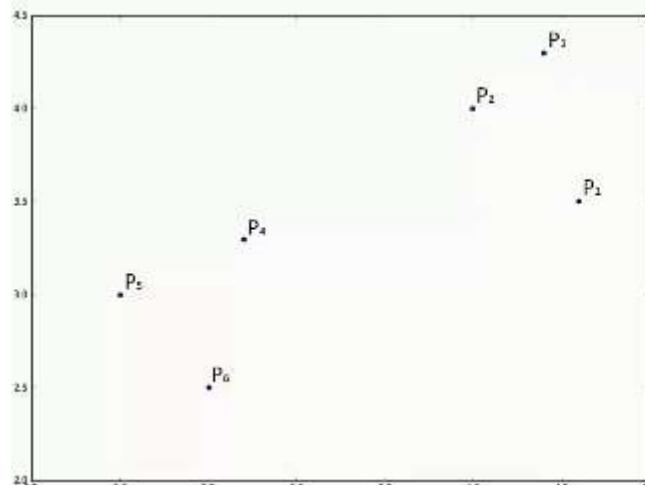


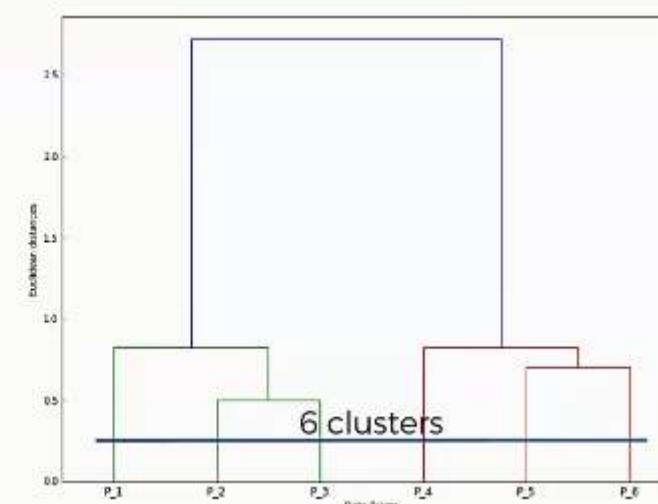
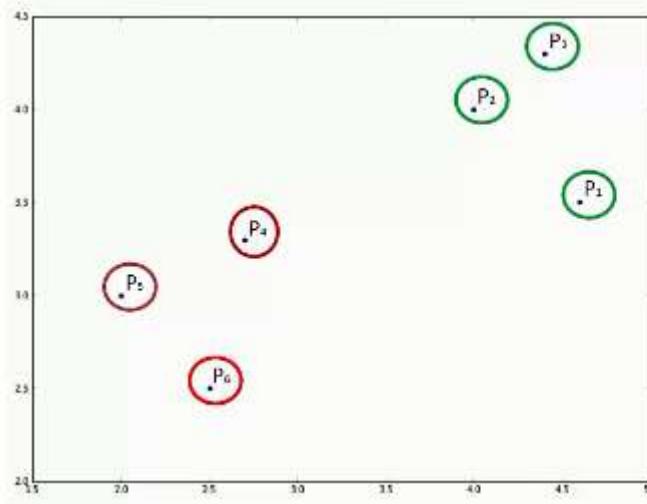
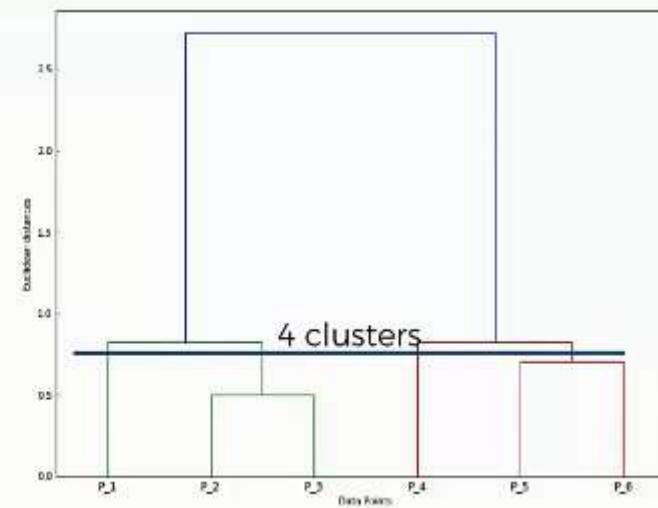
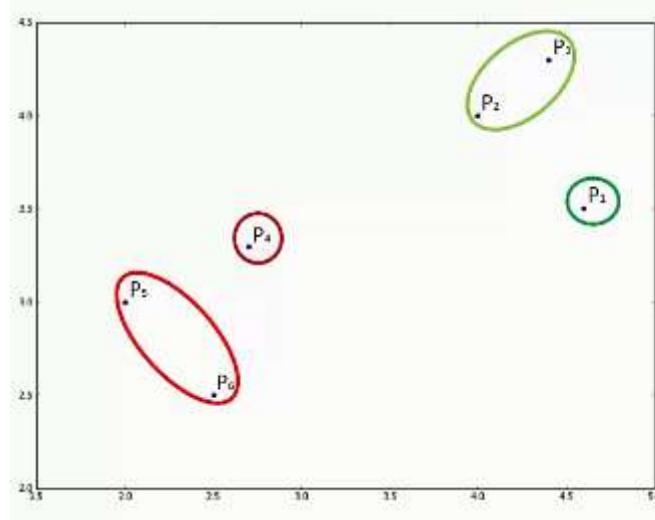
Student_ID	Marks
1	10
2	7
3	28
4	20
5	35



Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

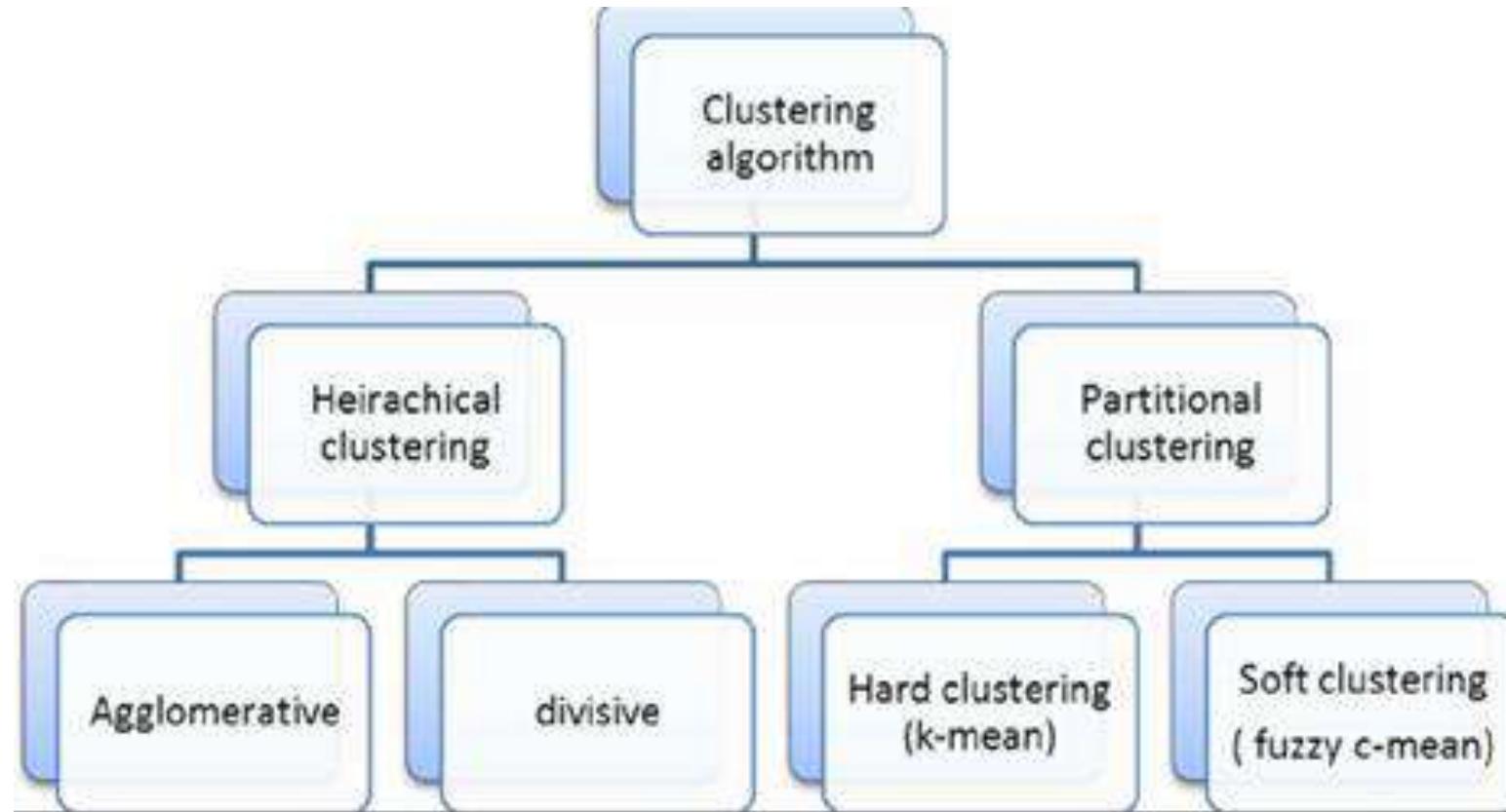
- More the distance of the vertical lines in the dendrogram, more the distance between those clusters.
- we can set a threshold distance and draw a horizontal line (*Generally, we try to set the threshold in such a way that it cuts the tallest vertical line*)





Divisive Hierarchical Clustering

Types of Clustering Algorithm

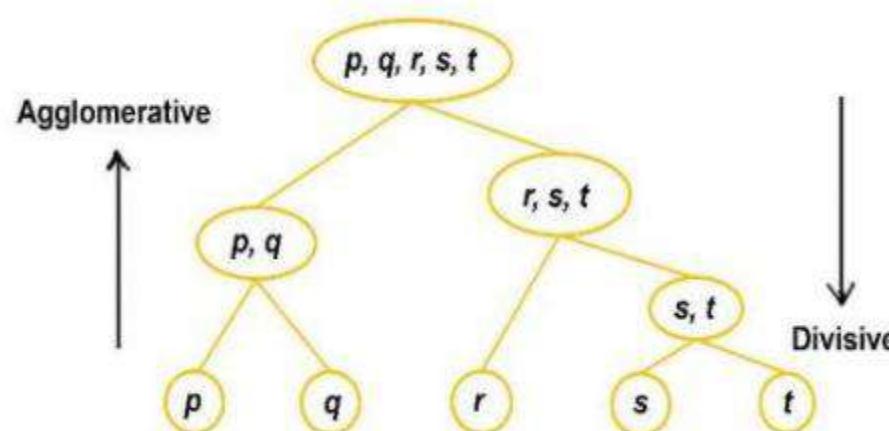


<https://medium.com/datadriveninvestor/clustering-algorithms-9fd35f34caa3>

Agglomerative and Divisive

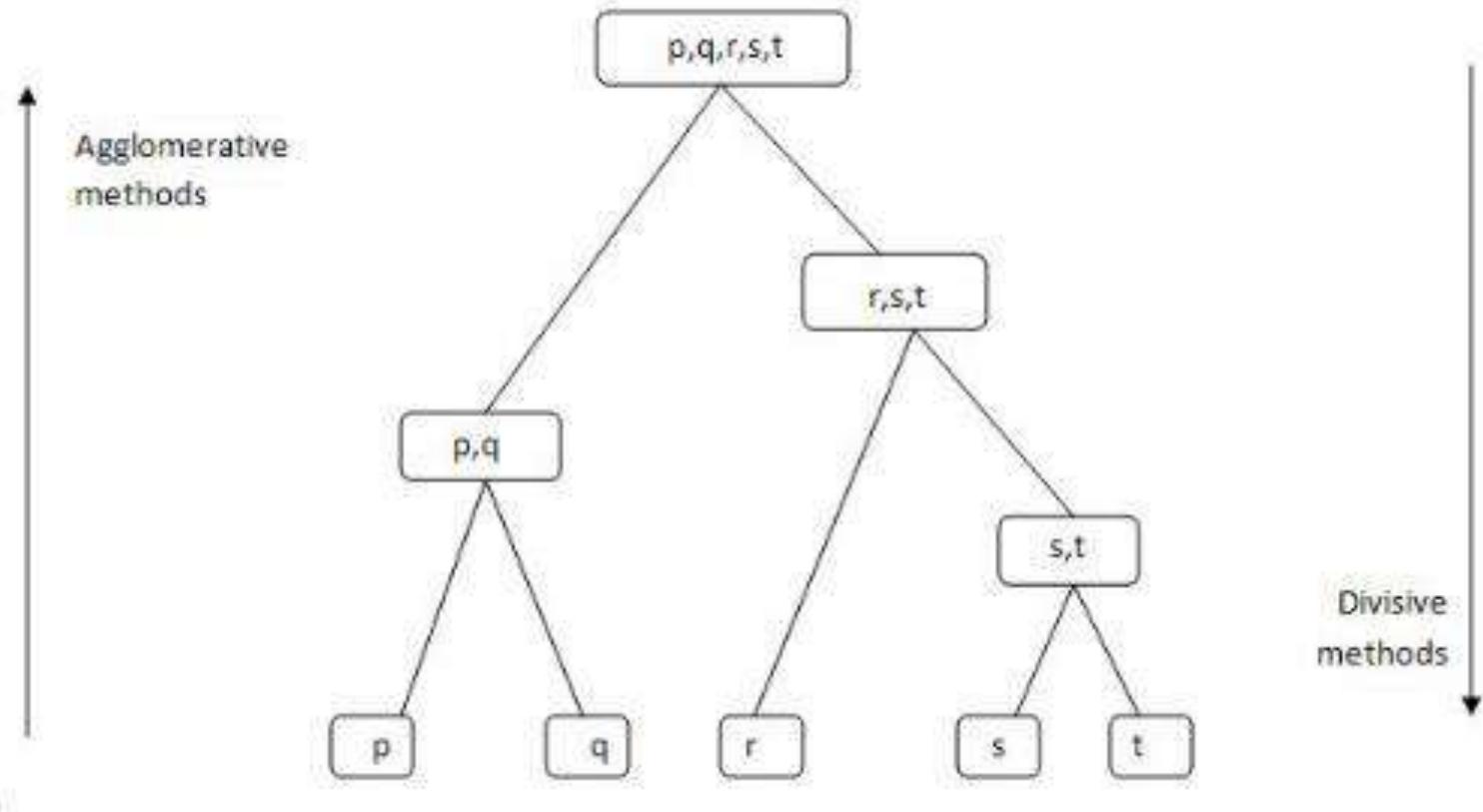
Agglomerative: This is a "bottom-up" approach: each observation starts in its own **cluster**, and pairs of **clusters** are merged as one moves up the hierarchy.

Divisive: This is a "top-down" approach: all observations start in one **cluster**, and splits are performed recursively as one moves down the hierarchy.



Divisive Hierarchical Clustering

- It is a top-down approach.
- Divisive hierarchical clustering works in the opposite way. Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster.
- Initially, all points will belong to the same cluster at the beginning.
- Now, at each iteration, **we split the farthest point in the cluster** and repeat this process until each cluster only contains a single point.
- This clustering algorithm **does not require us to prespecify the number of clusters**.



https://www.researchgate.net/publication/276332381_PERFORMANCE_OF_SELECTED_AGGLOMERATIVE_HIERARCHICAL_CLUSTERING_METHODS/figures?lo=1

1	2	3	4	5
1	0	4	4	5
2	4	0	4	3
3	4	4	0	5
4	5	3	5	0
5	7	3	7	2

The top level cluster is all points $\{1,2,3,4,5\}$.

Which point is most dissimilar? As you noted, we take the average distance to the other points.

for point 1

$$[d(1,2) + d(1,3) + d(1,4) + d(1,5)] / 4 \\ = (4 + 4 + 5 + 7) / 4 = 5$$

for point 2

$$[d(2,1) + d(2,3) + d(2,4) + d(2,5)] / 4 \\ = (4 + 4 + 3 + 3) / 4 = 3.5$$

The average distances for all points are:

1	2	3	4	5
5.00	3.50	5.00	3.00	4.75

A = {2,3,4,5} and B = {1}

Now we want to move any points that are closer to B than A

For point 2

$$[d(2,3) + d(2,4) + d(2,5)] / 3 - d(2,1) \\ = 10/3 - 4 = -2/3$$

For point 3

$$[d(3,2) + d(3,4) + d(3,5)] / 3 - d(3,1) \\ = 16/3 - 4 = 4/3$$

All of the differences are:

2	3	4	5
-2/3	4/3	-5/3	-3

1	2	3	4	5
1	0	4	4	5
2	4	0	4	3
3	4	4	0	5
4	5	3	5	0
5	7	3	7	2

Only point 3 is bigger than zero so we move it to cluster B. We now have

$$A = \{2,4,5\} \quad B = \{1,3\}$$

We check if any additional points should be moved. Again, we compute $d(x, A-x) - d(x, B)$ for each point in A.

			For point 2	
2	4	5	$[d(2,4) + d(2,5)] / 2 - [(d(2,1)+d(2,3))/2]$	
-1.0	-2.5	-4.5	$= 6/2 - 4 = -1$	

1	2	3	4	5
1	0	4	4	5
2	4	0	4	3
3	4	4	0	5
4	5	3	5	0
5	7	3	7	2
				0

All are negative (that is the remaining points in A are closer to A than to B), so we stop this division and we have the two clusters $\{2,4,5\}$ and $\{1,3\}$.

For the next step, we choose the cluster with the largest diameter, that is the cluster with the greatest distance between two points in the cluster.

$$\text{diameter}(\{1,3\}) = d(1,3) = 4$$

$$\text{diameter}(\{2,4,5\}) = \max(d(2,4), d(2,5), d(4,5)) = \max(3, 3, 2) = 3$$

So cluster $\{1,3\}$ has the largest diameter. Trivially, this will be split into $\{1\}$ and $\{3\}$. So now we have clusters $\{2,4,5\}$, $\{1\}$ and $\{3\}$.

At the next step, we must split the cluster $\{2,4,5\}$. As above, for each point we will find the point with the largest average distance to the rest of the points. For example we compute

$$\text{point 2: } [d(2,4) + d(2,5)] / 2 = [3+3]/2 = 3$$

The other averages are

$$\text{point 4: } 5/2$$

$$\text{point 5: } 5/2$$

1	2	3	4	5
1	0	4	4	5
2	4	0	4	3
3	4	4	0	5
4	5	3	5	0
5	7	3	7	2
				0

so point 2 is the most dissimilar.

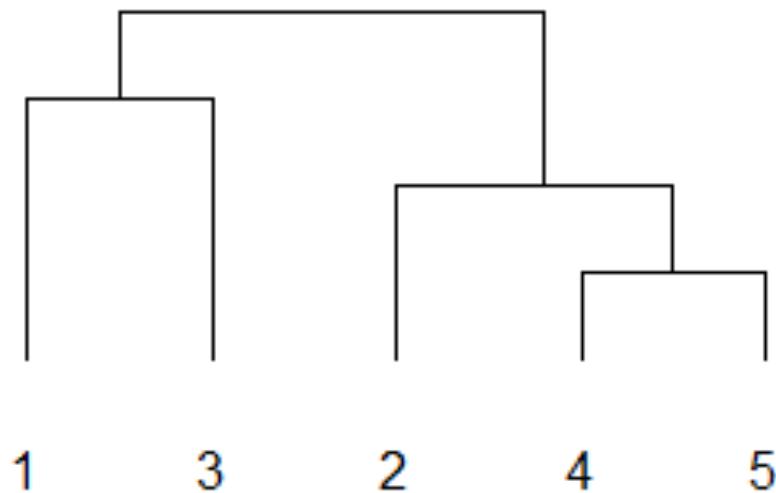
We split $\{2,4,5\}$ into $A=\{4,5\}$ and $B=\{2\}$. We need to check if some additional points should be moved into the set B.

$$d(4,5) - d(4,2) = 2-3 = -1$$

$$d(5,4) - d(5,2) = 2-3 = -1$$

So no additional points should be moved. We split $\{2,4,5\}$ into $\{2\}$ and $\{4,5\}$ giving us the partition of the full data set $\{1\}, \{2\}, \{3\}, \{4,5\}$

Finally, we divide the last cluster to get $\{1\} \{2\} \{3\} \{4\} \{5\}$ Thus, the full hierarchy looks like this:



Weaknesses of Hierarchical Clustering

- The weaknesses are that it rarely provides the best solution, it involves lots of arbitrary decisions
- It does not work with missing data
- It works poorly with mixed data types
- It does not work well on very large data sets
- Its main output, the dendrogram, is commonly misinterpreted.
- With many types of data, it is difficult to determine how to compute a distance matrix. There is no straightforward formula that can compute a distance where the variables are both numeric and qualitative.
 - For example, how can one compute the distance between a 45-year-old man, a 10-year-old-girl, and a 46-year-old woman?

Hierarchical Agglomerative vs Divisive Clustering

- Divisive clustering is **more complex** as compared to agglomerative clustering, as in case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.

- Divisive algorithm is **more accurate**. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without initially taking into account **the global distribution of data**.

Application of Clustering

- Customer Segmentation
- Market Segmentation
- Document Clustering
- Image Segmentation
- Recommendation Engines
- Social Network Analysis
- Search Result Grouping
- Anomaly Detection

Customer Segmentation



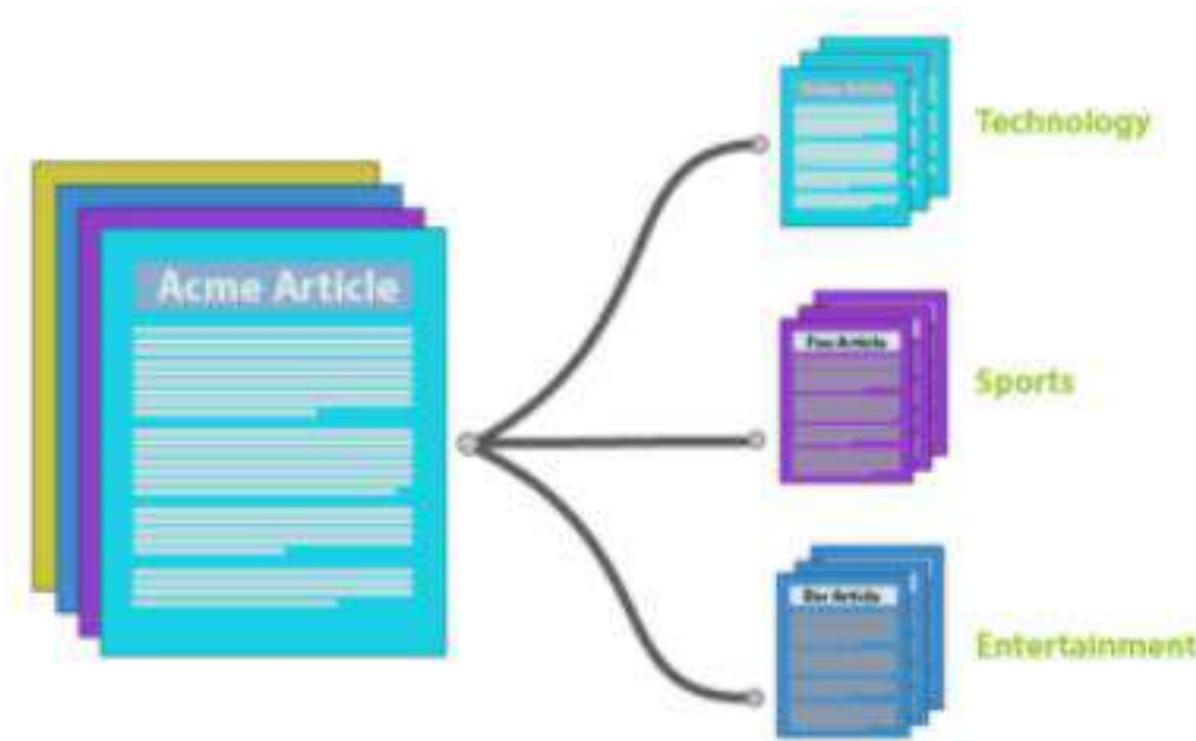
<https://blog.oxfordcollegeofmarketing.com/2014/07/30/cim-new-syllabus-focus-the-importance-of-market-segmentation/>

Market Segmentation



<https://blog.oxfordcollegeofmarketing.com/2014/07/30/cim-new-syllabus-focus-the-importance-of-market-segmentation/>

Document Clustering



<https://towardsdatascience.com/applying-machine-learning-to-classify-an-unsupervised-text-document-e7bb6265f52>

Image Segmentation



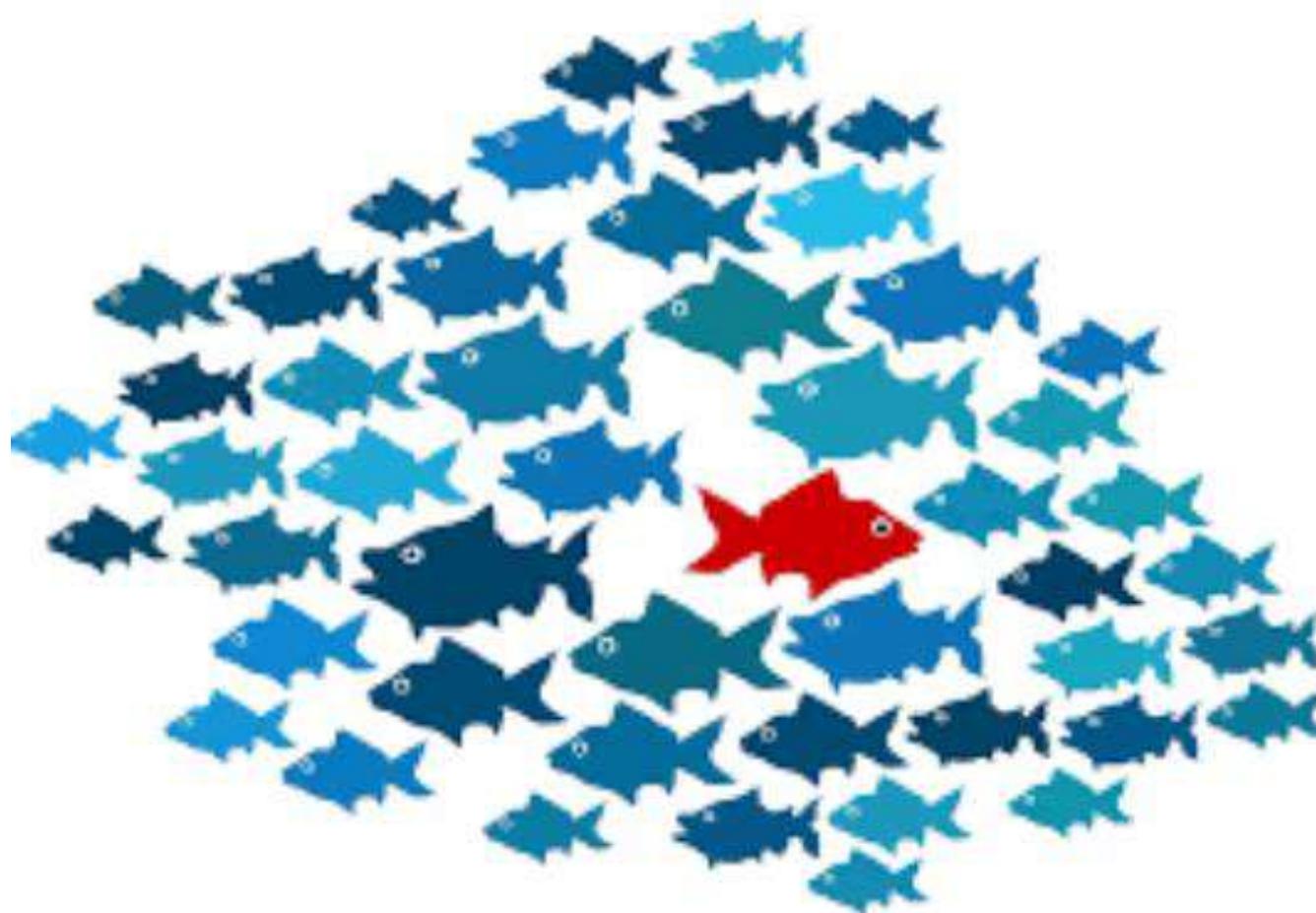
<https://in.pinterest.com/pin/668362400931054198/>

Social Network Analysis

- Social network analysis [SNA] is the mapping and measuring of relationships and flows between people, groups, organizations, computers, URLs, and other connected information/knowledge entities.



Anomaly Detection



<https://thedataScientist.com/anomaly-detection-why-you-need-it/>

Data Mining (20CP306T)

Dr. Rajeev Kumar Gupta
Assistant Professor
Pandit Deendayal Energy University
Gandhinagar, Gujarat

10 November 2022

1

This figure shows a data set containing nonconvex clusters and outliers/noises. Given such data, k-means algorithm has difficulties in identifying these clusters with arbitrary shapes.

10 November 2022

3

Density-based Spatial Clustering of Applications with Noise (DBSCAN)

Why DBSCAN?

- Partitioning methods and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are **suitable only for compact and well-separated clusters**.
- K-Means and Hierarchical Clustering both fail in creating clusters of **arbitrary shapes**. They are not able to form clusters based on varying densities.
- Severely affected by the **presence of noise and outliers in the data**.
- Specify number of cluster

What is DBSCAN?

- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- The most exciting feature of DBSCAN clustering is that it is **robust to outliers**. It also **does not require the number of clusters** to be told beforehand, unlike K-Means, where we have to specify the number of centroids.

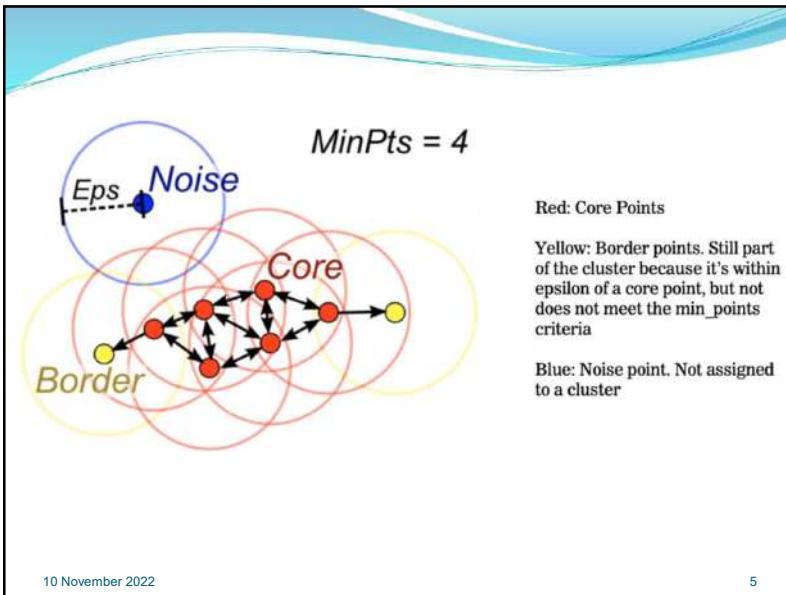
10 November 2022

2

- DBSCAN requires only two parameters: *epsilon* and *minPoints*.
- **Epsilon** is the radius of the circle to be created around each data point to check the density and **minPoints** is the minimum number of data points required inside that circle for that data point to be classified as a **Core point**.
- DBSCAN creates a circle of epsilon radius around every data point and classifies them into **Core point, Border point, and Noise**.
- A data point is a Core point if the circle around it contains **at least 'minPoints' number of points**. If the number of points is less than minPoints and it has at least one core point, then it is classified as Border Point, and if there are no other data points around any data point within epsilon radius, then it treated as Noise.

10 November 2022

4

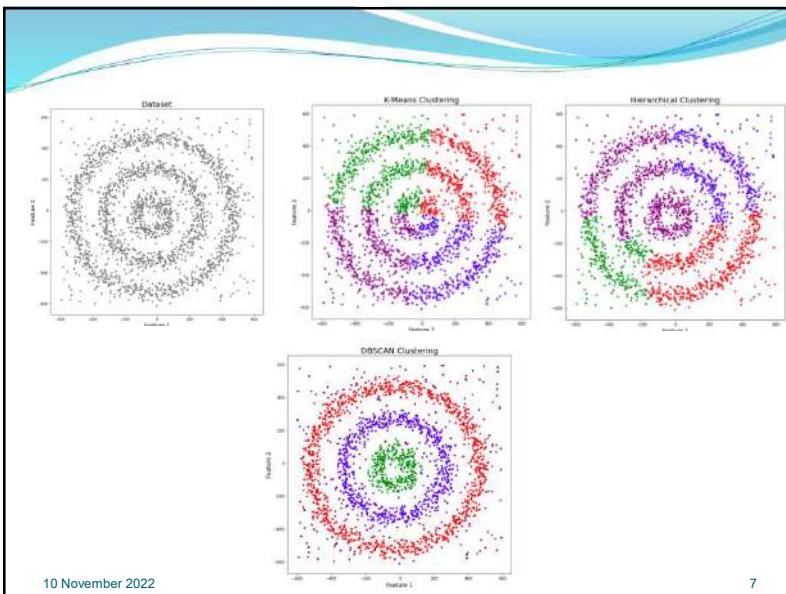


Parameter Selection in DBSCAN Clustering

- DBSCAN is very sensitive to the values of *epsilon* and *minPoints*. A slight variation in these values can significantly change the results produced by the DBSCAN algorithm.
- The value of *minPoints* should be at least one greater than the number of dimensions of the dataset, i.e., $\text{minPoints} \geq \text{Dimensions} + 1$.
- It does not make sense to take *minPoints* as 1 because it will result in each point being a separate cluster. Therefore, it must be at least 3. Generally, it is twice the dimensions. But domain knowledge also decides its value.
- The value of *epsilon* can be decided from the K-distance graph. The point of maximum curvature (elbow) in this graph tells us about the value of *epsilon*.
- If the value of *epsilon* chosen is too small then a higher number of clusters will be created, and more data points will be taken as noise. Whereas, if chosen too big then various small clusters will merge into a big cluster, and we will lose details.

10 November 2022

6



Cluster Validation

10 November 2022

8

Cluster Validation

1) Internal cluster validation, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.

2) External cluster validation, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the “true” cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.

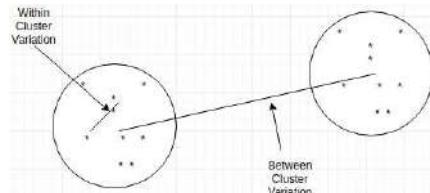
3) Relative cluster validation, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g., varying the number of clusters k). It's generally used for determining the optimal number of clusters.

10 November 2022

9

WCV (Within Cluster Variation): The variation in the data points that are present in the cluster.

BCV (Between Cluster Variation): The variation between 2 clusters.



<https://medium.com/datadriveneinvestor/k-means-clustering-4a700d4a4720>

- The goal of K Means algorithm is to minimize the Within Cluster Variation and maximize the Between Cluster Variation.

10 November 2022

10

Issues for Cluster Validation

- 1) Determining the clustering tendency of a set of data, i.e., identify whether non-random structure actually exists in the data.
- 2) Determining the correct number of clusters.
- 3) Evaluating how well the results of a cluster analysis fit the data **without reference to external information**.

Solutions

- 1) Comparing the results of a cluster analysis to **externally known results**, such as externally provided class labels.

10 November 2022

11

Cluster Validation

1) Unsupervised Measure (No ground truth)

- Cluster Cohesion and cluster separation
- Sum square error (SSE)
- Silhouette Coefficient
- Internal indices

2) Supervised Measure (Have ground truth)

- External Indices
- Purity
- Rand Index
- Entropy
- Jaccard Coefficient

3) Relative Measure (Combination of supervised and unsupervised)

10 November 2022

12

Similarity and Dissimilarity Measures

10 November 2022

13

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.
Webster's Dictionary



Similarity is hard to define, but...
"We know it when we see it"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

10 November 2022

14

Similarity and Dissimilarity Measures

- In clustering techniques, similarity (or dissimilarity) is an important measurement.
- Informally, **similarity** between two objects (e.g., two images, two documents, two records, etc.) is a **numerical measure** of the degree to which two objects are **alike**.
- The **dissimilarity** on the other hand, is another alternative (or opposite) measure of the degree to which two objects are **different**.
- Both similarity and dissimilarity also termed as **proximity**.
- Usually, similarity and dissimilarity are **non-negative numbers** and may range from **zero (highly dissimilar (no similar))** to some finite/infinite value (**highly similar (no dissimilar)**).

Note:

- Frequently, the term **distance** is used as a synonym for dissimilarity

10 November 2022

15

Proximity Calculation

- Proximity calculation to compute $p_{(i,j)}$ is different for different types of attributes.

Proximity calculation for Nominal attributes:

- For example, binary attribute, **Gender = {Male, female}** where **Male** is equivalent to **binary 1** and **female** is equivalent to **binary 0**.
- Similarity value is 1 if the two objects contains the same attribute value, while similarity value is 0 implies objects are not at all similar.

Object	Gender
Ram	Male
Sita	Female
Laxman	Male

Here, Similarity value let it be denoted by p , among different objects are as follows.

$$p(Ram, sita) = 0$$

$$p(Ram, Laxman) = 1$$

Note : In this case, if q denotes the **dissimilarity** between two objects i and j with single binary attributes, then $q_{(i,j)} = 1 - p_{(i,j)}$

10 November 2022

16

Proximity Calculation

- Now, let us focus on how to calculate proximity measures between objects which are defined by two or more binary attributes.
- Suppose, the number of attributes be b . We can define the contingency table summarizing the different matches and mismatches between any two objects x and y , which are as follows.

		Object y	
		1	0
Object x	1	f_{11}	f_{10}
	0	f_{01}	f_{00}

Here, f_{11} = the number of attributes where $x=1$ and $y=1$.

f_{10} = the number of attributes where $x=1$ and $y=0$.

f_{01} = the number of attributes where $x=0$ and $y=1$.

f_{00} = the number of attributes where $x=0$ and $y=0$.

Note : $f_{00} + f_{01} + f_{10} + f_{11} = b$, the total number of binary attributes.

Now, two cases may arise: symmetric and asymmetric binary attributes.

10 November 2022

17

Similarity Measure with Symmetric Binary

- To measure the similarity between two objects defined by symmetric binary attributes using a measure called symmetric binary coefficient and denoted as \mathcal{S} and defined below

$$\mathcal{S} = \frac{\text{Number of matching attribute values}}{\text{Total number of attributes}}$$

or

$$\mathcal{S} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

The dissimilarity measure, likewise can be denoted as \mathcal{D} and defined as

$$\mathcal{D} = \frac{\text{Number of mismatched attribute values}}{\text{Total number of attributes}}$$

or

$$\mathcal{D} = \frac{f_{01} + f_{10}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

Note that, $\mathcal{D} = 1 - \mathcal{S}$

18

Similarity Measure with Symmetric Binary

Example: Proximity measures with symmetric binary attributes

Consider the following two dataset, where objects are defined with symmetric binary attributes.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I},
Hobby = {T, C}, Job = {Y, N}

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

$$\mathcal{S}(\text{Hari}, \text{Ram}) = \frac{1+2}{1+2+1+2} = 0.5$$

10 November 2022

19