INTRODUCTION TO DATA ANALYSIS_ UNDERSTANDING GROUPS

2nd Sem, MCA

CONTENT

- Understanding Groups in Data Analysis
  - Clustering
    - Hierarchical Clustering
    - K-Means Clustering
  - Association rules
    - Market Basket Analysis
    - Recommendation system
    - Apriori algorithm
    - FP Growth Algorithm
  - Decission Tree
    - Impurity
    - Random Forest

DATA ANALYSIS - GROUPING

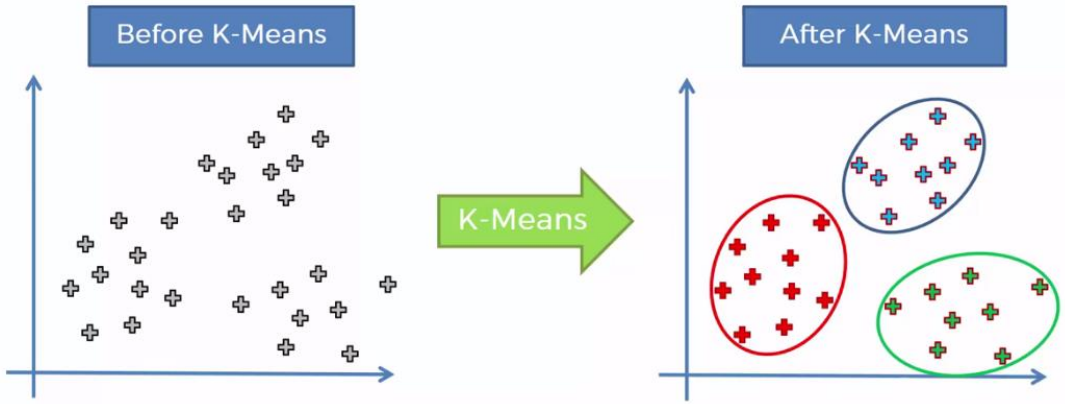
- Grouping and classification techniques are very important methods in data analysis.
- **Grouping Analysis** methods helps to determine natural groupings in data.
- Useful to decompose data set into simpler subsets → helps to make sense of entire collection of observations.
- For each group summary statistics, variety of graphs may help in better analysis
- Different ways to visualize and group observations,
  - *Clustering:* based on similarities of overall set of variables of interest.
  - *Association rule:* identify groups based on interesting combinations of predefined categories
  - *Decision tree:* groups observation based on combination of ranges of continuous variables or of specific categories.

DATA ANALYSIS – CLUSTERING

- **Cluster**: group of (similar) objects that belongs to same class.
- **Clustering**: process of making a group of abstract objects into classes of similar objects.
- Given a data set of items, with certain features, and values for these features; the task is to categorize those items into groups.
    - Used to find similarity as well as relationship patterns among data samples and then cluster those samples into groups having similarity based on features.
    - Clustering is important because it determines the intrinsic grouping among the present unlabeled data.

Clustering methods –

- Partitioning Method
- Hierarchical Method; Agglomerative Approach, Divisive Approach
- Constraint-based Method
- Density-based Method
- Grid-Based Method
- Model-Based Method



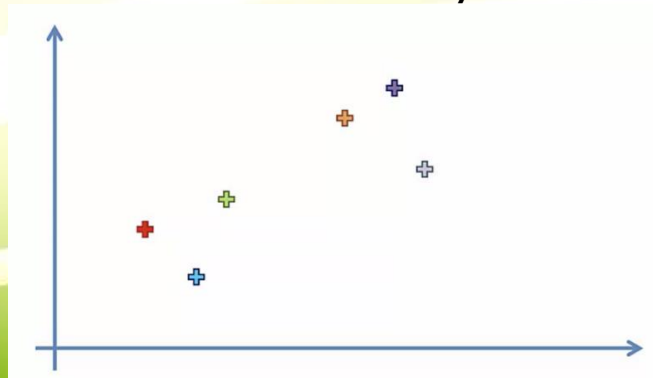
Clustering

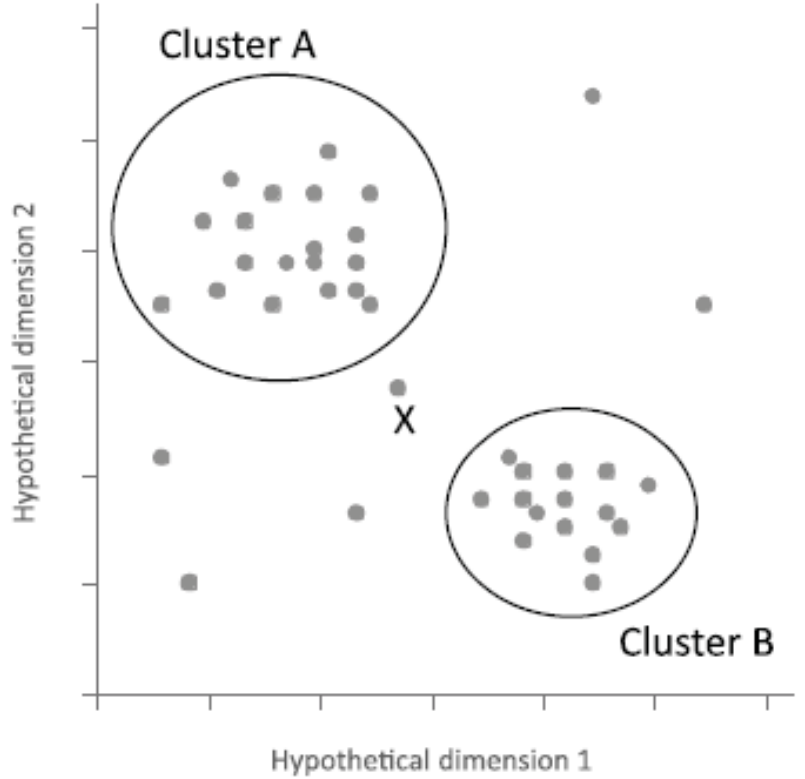
Applications of Cluster Analysis

- *Collaborative systems and customer segmentation:* help marketers discover distinct groups in their customer base → characterize customer groups based on purchasing patterns.
- helps in classifying documents on the web for information discovery.
- used in outlier detection applications; example detection of credit card fraud.
- *Biological data analysis:* used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Data summarization and compression
- Trend detection in dynamic data
- Social network analysis

DATA ANALYSIS – CLUSTERING

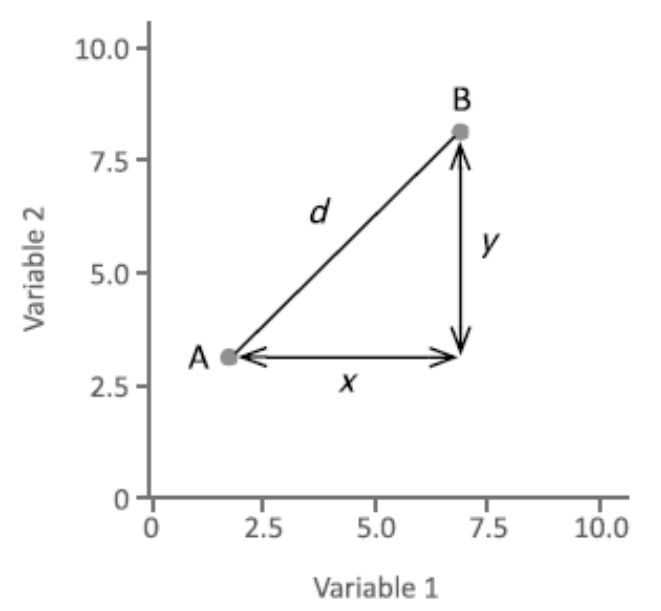
- Clustering is an *unsupervised* method for grouping.
- **Unsupervised**: groups are not known in advance.
- Clustering method chosen to subdivide data into groups applies automated procedure to discover groups based on some criteria.
- Many clustering methods.
- Each method will group data differently based on criteria it uses.
- For clustering, there is no way to measure accuracy (usefulness matters).
- **Distance** between two observations defines how similar they are to be in same cluster or not.





DATA ANALYSIS – CLUSTERING

- Clustering needs a way to measure how similar the observations are to each other.
- To calculate similarity, distance between observations is computed.
- Simple distance between two observations can be calculated using simple trigonometry.



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

d: Euclidean distance
(x1, y1) → coordinate of first point
(x2, y2) → coordinate of second point.

$$x = 7 - 2 = 5$$
$$y = 8 - 3 = 5$$
$$d = \sqrt{x^2 + y^2} = \sqrt{25 + 25} = 7.07$$

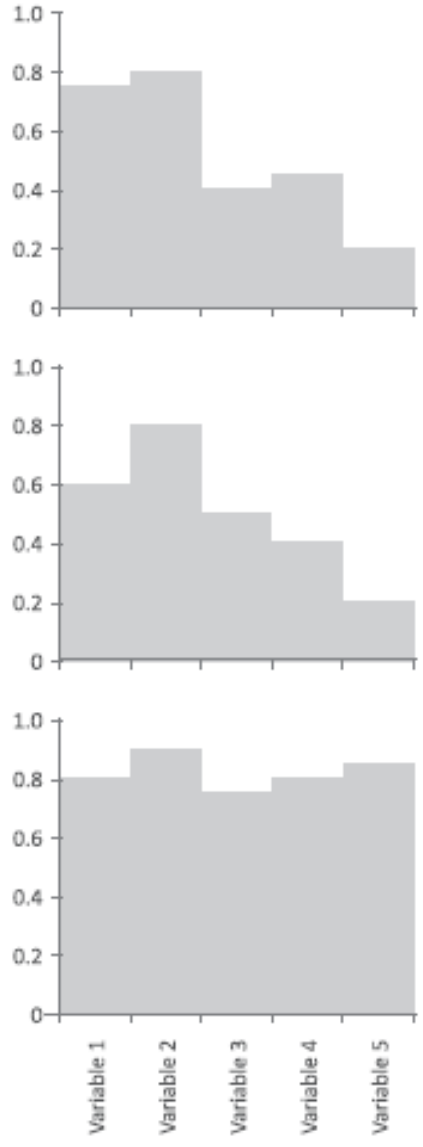
- Distance metrics: **Euclidean, Jaccard**, City Block, Minkowski, Cosine, Spearman, Hamming, Mahalanobis etc.

DATA ANALYSIS – CLUSTERING

- **Euclidian Distance**: calculate distances between observations with more than two variables.

$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

ID	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	0.7	0.8	0.4	0.5	0.2
B	0.6	0.8	0.5	0.4	0.2
C	0.8	0.9	0.7	0.8	0.9



DATA ANALYSIS – CLUSTERING

- **Euclidian Distance**: $d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

ID	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	0.7	0.8	0.4	0.5	0.2
B	0.6	0.8	0.5	0.4	0.2
C	0.8	0.9	0.7	0.8	0.9

- Used to calculate distance between two observations p and q where each observation has n variables.
- Euclidean distance between A and B; A and C; B and C.

$$d_{A-B} = \sqrt{(0.7-0.6)^2 + (0.8-0.8)^2 + (0.4-0.5)^2 + (0.5-0.4)^2 + (0.2-0.2)^2}$$
$$d_{A-B} = 0.17$$

- More similarity between observations A-B than A-C.
- C is not closely related to either A or B.

The Euclidean distances between A and C is

$$d_{A-C} = \sqrt{(0.7-0.8)^2 + (0.8-0.9)^2 + (0.4-0.7)^2 + (0.5-0.8)^2 + (0.2-0.9)^2}$$
$$d_{A-C} = 0.83$$

The Euclidean distance between B and C is

$$d_{B-C} = \sqrt{(0.6-0.8)^2 + (0.8-0.9)^2 + (0.5-0.7)^2 + (0.4-0.8)^2 + (0.2-0.9)^2}$$
$$d_{B-C} = 0.86$$

DATA ANALYSIS – CLUSTERING

- Euclidean distance metric can be used only for numerical variables.
- **Jaccard distance**: for binary variables.

	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	1	1	0	0	1
B	1	1	0	0	0
C	0	0	1	1	1

- This approach is based on number of common or different (0/1) values between corresponding variables across each pair of observations.
  - *Count11: Count of all variables that are 1 in "Observation 1" and 1 in "Observation 2."*
  - *Count10: Count of all variables that are 1 in "Observation 1" and 0 in "Observation 2."*
  - *Count01: Count of all variables that are 0 in "Observation 1" and 1 in "Observation 2."*
  - ~~*Count00: Count of all variables that are 0 in "Observation 1" and 0 in "Observation 2."*~~
- **Jaccard distance (d):**

$$d = \frac{\text{Count}_{10} + \text{Count}_{01}}{\text{Count}_{11} + \text{Count}_{10} + \text{Count}_{01}}$$

DATA ANALYSIS – CLUSTERING

- **Jaccard distance**: $$d = \frac{\text{Count}_{10} + \text{Count}_{01}}{\text{Count}_{11} + \text{Count}_{10} + \text{Count}_{01}}$$

	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	1	1	0	0	1
B	1	1	0	0	0
C	0	0	1	1	1

  - *Count11: Count of all variables that are 1 in "Observation 1" and 1 in "Observation 2."*
  - *Count10: Count of all variables that are 1 in "Observation 1" and 0 in "Observation 2."*
  - *Count01: Count of all variables that are 0 in "Observation 1" and 1 in "Observation 2."*
  - ~~*Count00: Count of all variables that are 0 in "Observation 1" and 0 in "Observation 2."*~~

Jaccard distance (d):

- between A and B $d_{A-B} = (1 + 0)/(2 + 1 + 0) = 0.33$
- between A and C $d_{A-C} = (2 + 2)/(1 + 2 + 2) = 0.8$
- between B and C $d_{B-C} = (2 + 3)/(0 + 2 + 3) = 1.0$

DATA ANALYSIS – CLUSTERING

Example: Calculate the Jaccard distance (replacing **None with 0, Mild with 1,** and **Severe with 2**) using the variables: Fever, Headaches, General aches, Weakness, Exhaustion, Stuffy nose, Sneezing, Sore throat, Chest discomfort, for the following pairs of patient observations:

(a) 1326 and 398
(b) 1326 and 1234
(c) 6377 and 2662

$$d = \frac{\text{Count}_{10} + \text{Count}_{01}}{\text{Count}_{11} + \text{Count}_{10} + \text{Count}_{01}}$$

Patient ID	Fever	Headaches	General Aches	Weakness	Exhaustion	Stuffy Nose	Sneezing	Sore Throat	Chest Discomfor
1326	None	Mild	None	None	None	Mild	Severe	Severe	Mild
398	Severe	Severe	Severe	Severe	Severe	None	None	Severe	Severe
6377	Severe	Severe	Mild	Severe	Severe	Severe	None	Severe	Severe
1234	None	None	None	Mild	None	Severe	None	Mild	Mild
2662	Severe	Severe	Mild	Severe	Severe	Severe	None	Severe	Severe
9477	None	None	None	Mild	None	Severe	Severe	Severe	None
7286	Severe	Severe	Severe	Severe	Severe	None	None	None	Severe
1732	None	None	None	None	None	Severe	Severe	None	Mild
1082	None	Mild	Mild	None	None	Severe	Severe	Severe	Severe
1429	Severe	Severe	Severe	Mild	Mild	None	Severe	None	Severe
14455	None	None	None	Mild	None	Severe	Mild	Severe	None
524	Severe	Mild	Severe	Mild	Severe	None	Severe	None	Mild
1542	None	None	Mild	Mild	None	Severe	Severe	Severe	None
8775	Severe	Severe	Severe	Severe	Mild	None	Severe	Severe	Severe
1615	Mild	None	None	Mild	None	Severe	None	Severe	Mild
1132	None	None	None	None	None	Severe	Severe	Severe	Severe
4522	Severe	Mild	Severe	Mild	Mild	None	None	None	Severe

DATA ANALYSIS – CLUSTERING

Euclidean distance needs first the data to normalize before bring using. Also, as dimensionality increases, it becomes more complex and less useful.

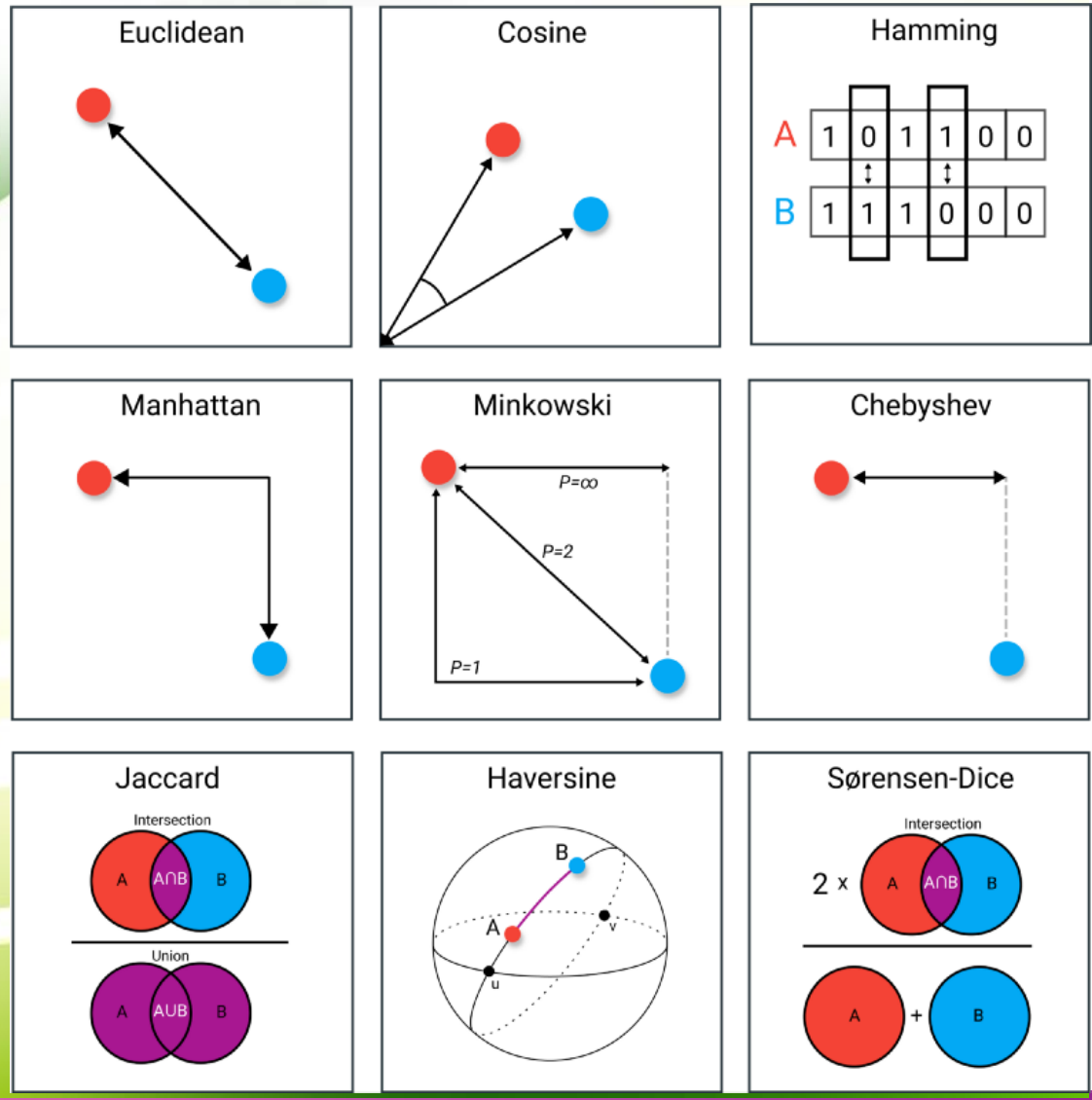
$$D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2)}$$

Manhattan distance

- Taxicab distance or City Block distance

$$D(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

Hamming distance used to compare two binary strings of equal length (compares similarity by calculating number of characters that are different from each other).



DATA ANALYSIS – CLUSTERING

Chebyshev distance (Chessboard distance)**:** maximum distance along one axis.

$$D(x,y) = \max_i \left(|x_i - y_i|\right)$$

Minkowski distance

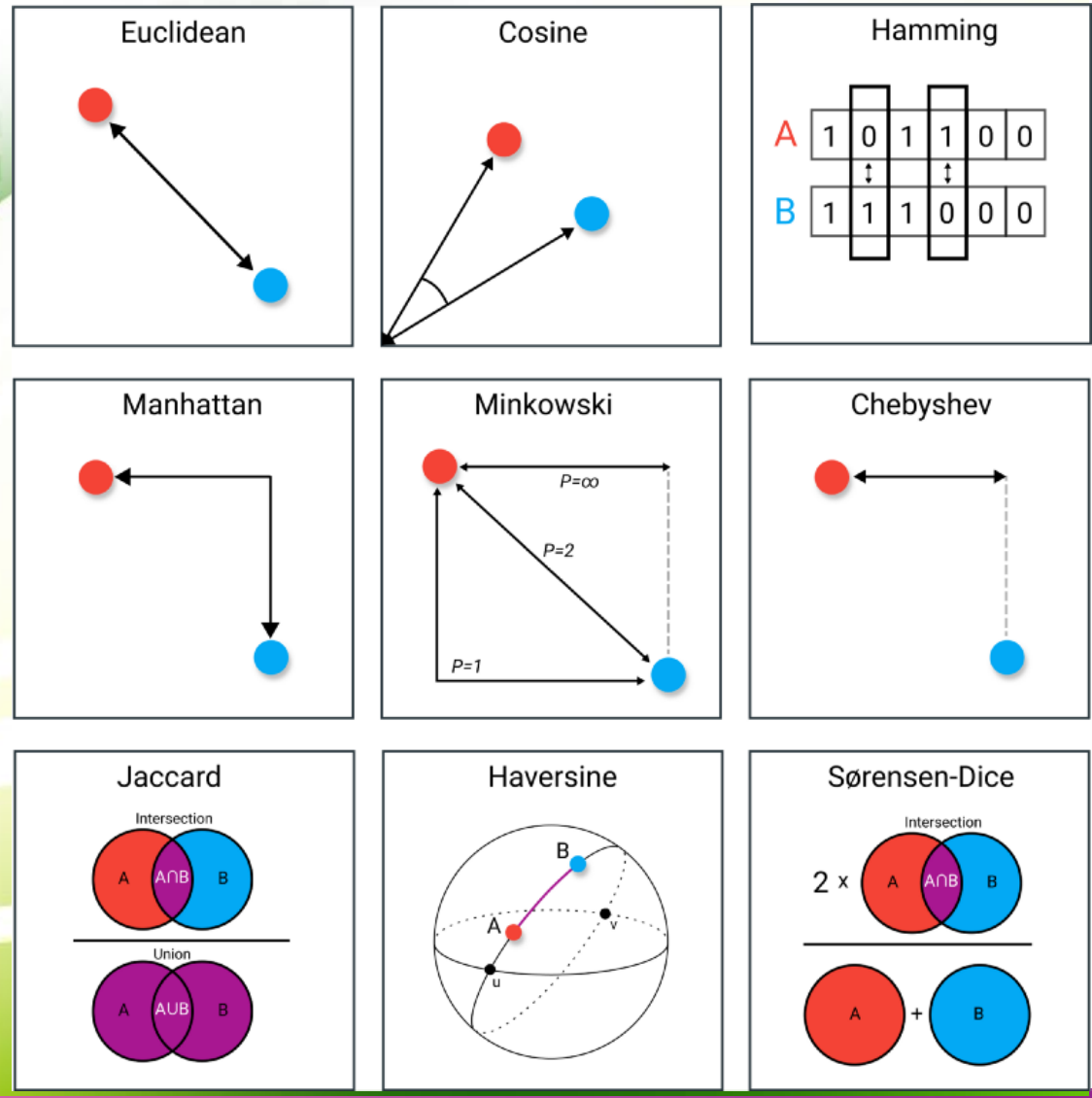
$$D(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

P = 1 → Manhattan distance
P = 2 → Euclidean distance
P = ∞ → Chebyshev distance

Jaccard index calculates similarity and diversity as size of intersection divided by size of union.

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

$$d = \frac{\text{Count}_{10} + \text{Count}_{01}}{\text{Count}_{11} + \text{Count}_{10} + \text{Count}_{01}}$$



DATA ANALYSIS – HIERARCHICAL CLUSTERING

- **Hierarchical Clustering**: creates hierarchical decomposition of given set of data objects.
- Two approaches;
- **Agglomerative Approach**: (bottom-up approach) "*AGNES*" (*Agglomerative Nesting*)
  - Start with each object forming a separate/singleton group/cluster.
  - Keeps on merging objects or groups that are **close/similar to one another**. *(Euclidian distance)*
  - Keep on doing so until all of groups are merged into one or until termination condition holds.
  - normally limited to data sets with fewer (< 10,000 observations) → computational cost to generate hierarchical tree can be high for larger numbers of observations
  - result is a tree-based representation of the objects, named *dendrogram*.
- **Divisive Approach**: (top-down approach) "DIANA" (*Divise Analysis*)
  - Start with all of objects in same cluster.
  - In continuous iteration, a cluster is split up into smaller clusters.
  - Keep doing until each object in one cluster or termination condition holds.

Hierarchical clustering

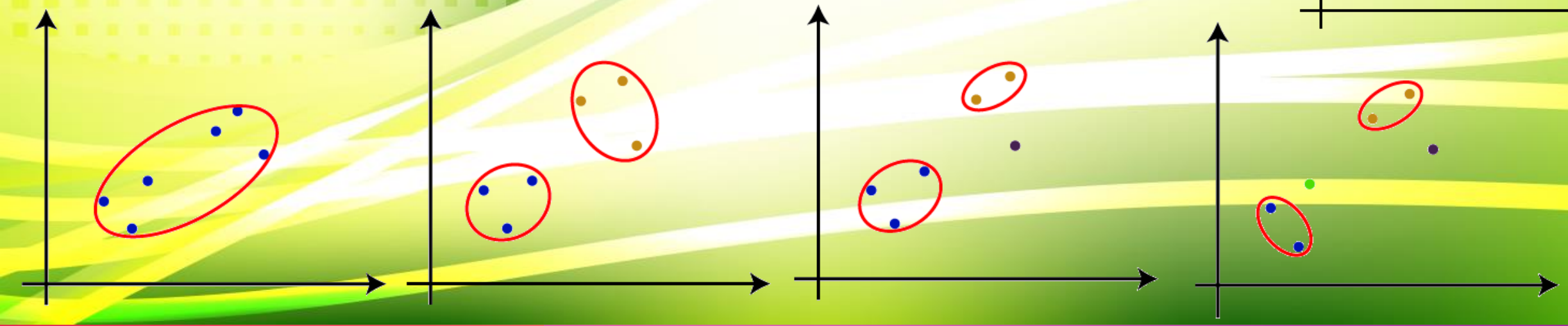
Step-1: Create each data point as a single cluster (for N data points, number of clusters will also be N).

Step-2: Take two closest data points/clusters and merge them to form one cluster (N-1 clusters remains).

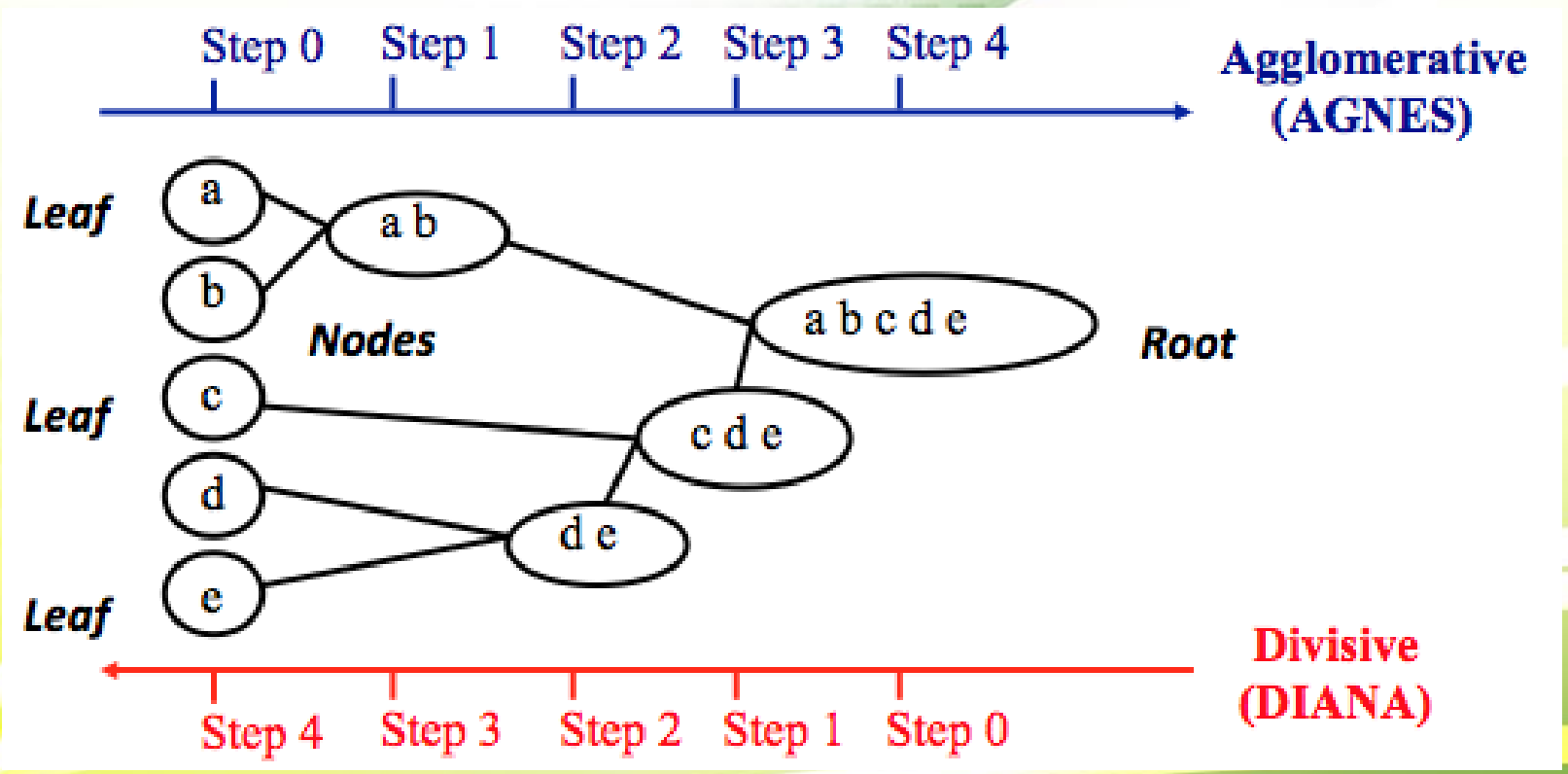
Step-3: Again, take two closest clusters and merge them together to form one cluster (remaining N-2 clusters).

Step-4: **Repeat Step 3** until only one cluster left *(or termination condition meets).*

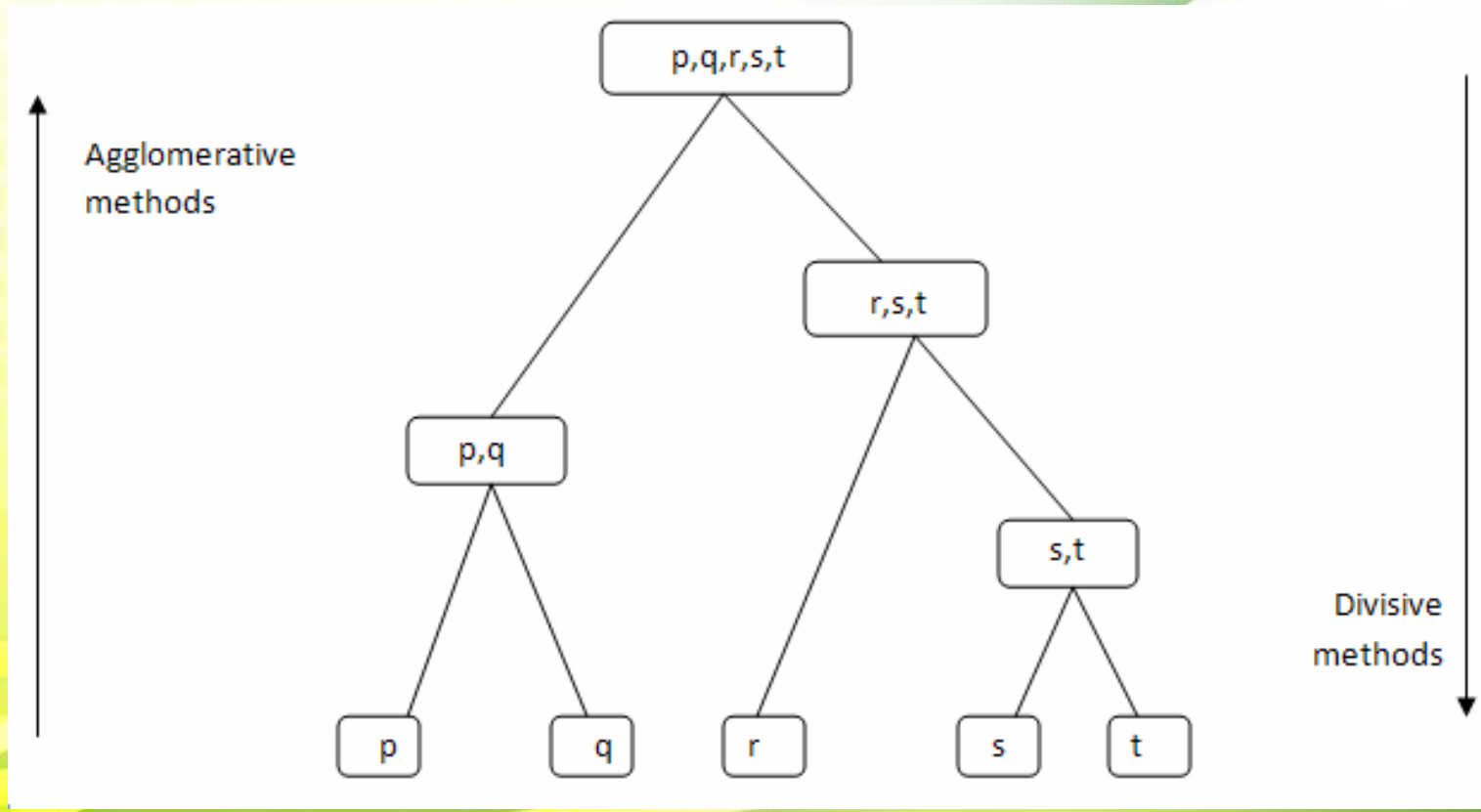
Step-5: Once all clusters are combined into one big cluster, develop the dendrogram to find required clusters.

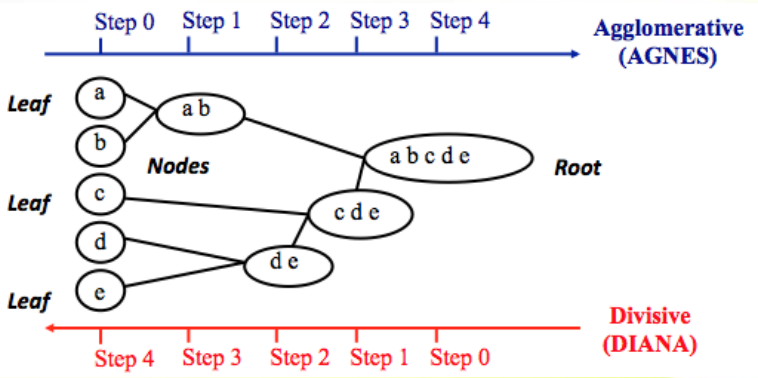


DATA ANALYSIS – HIERARCHICAL CLUSTERING

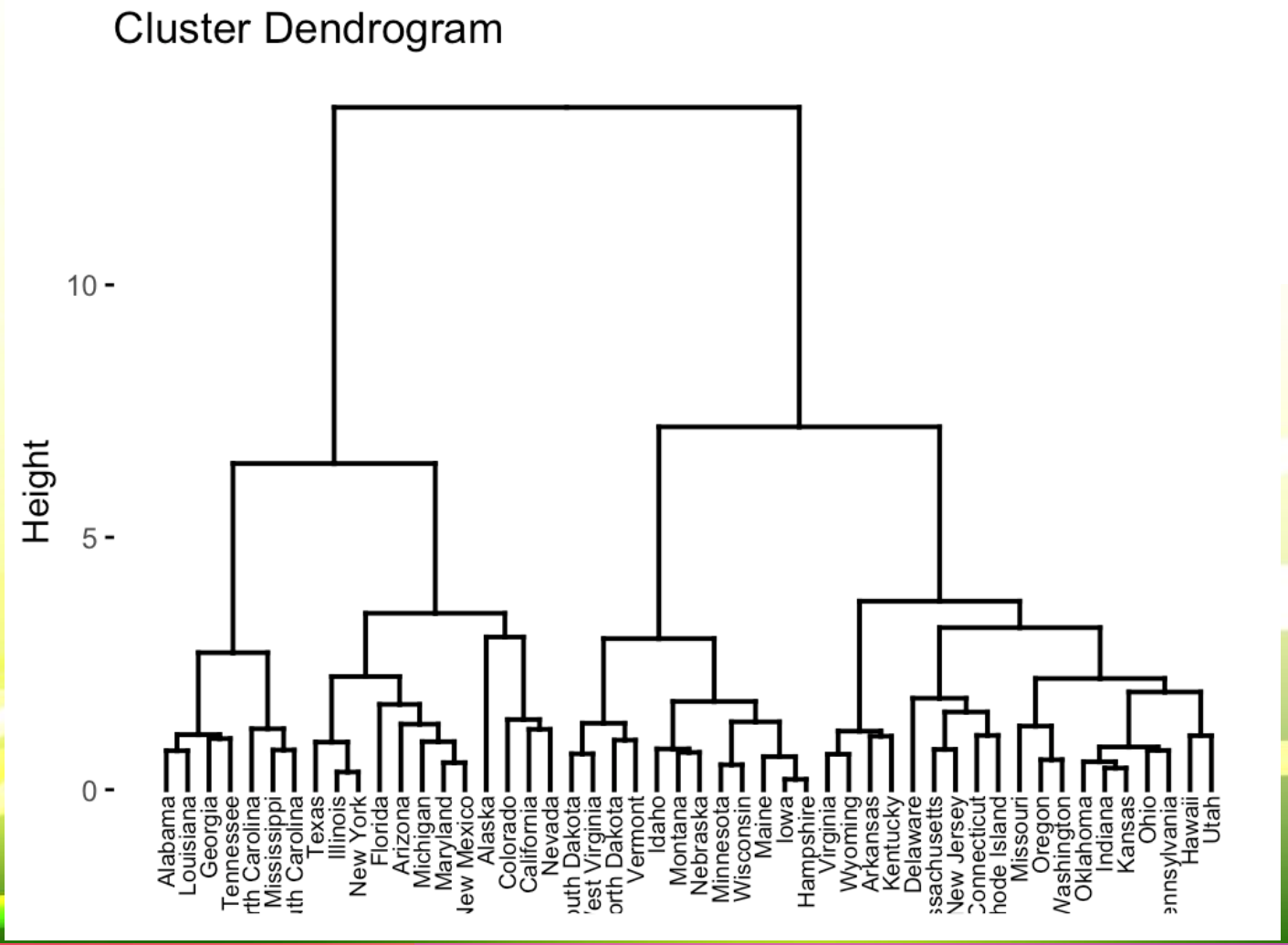


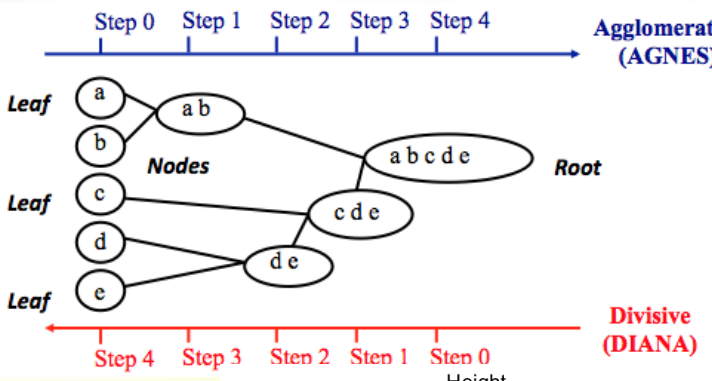
DATA ANALYSIS – HIERARCHICAL CLUSTERING

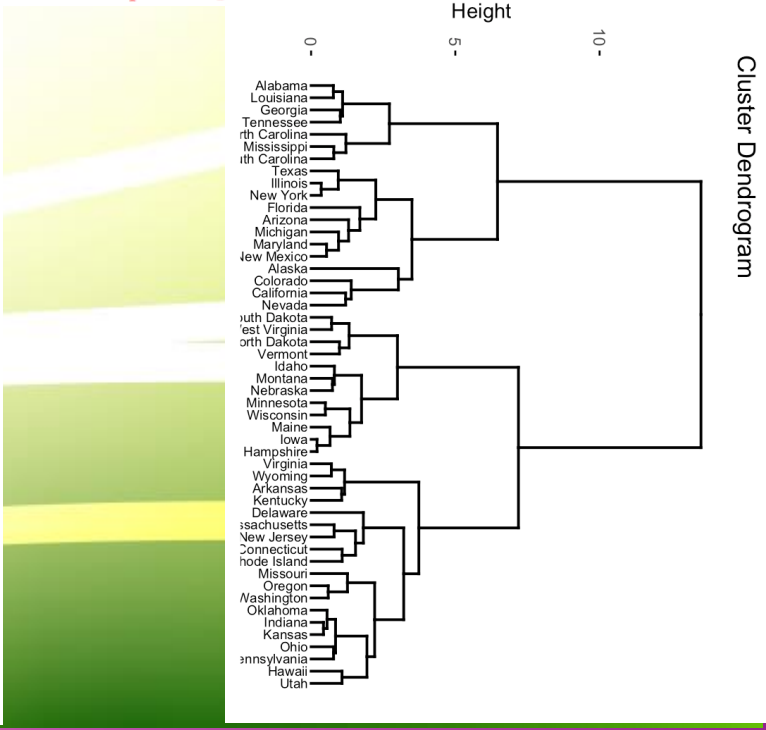




DATA ANALYSIS – HIERARCHICAL CLUSTERING







DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

Data

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

Proximity/distance matrix

Iteration - 2

Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

Data

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

Proximity/distance matrix

DATA ANALYSIS – HIERARCHICAL CLUSTERING

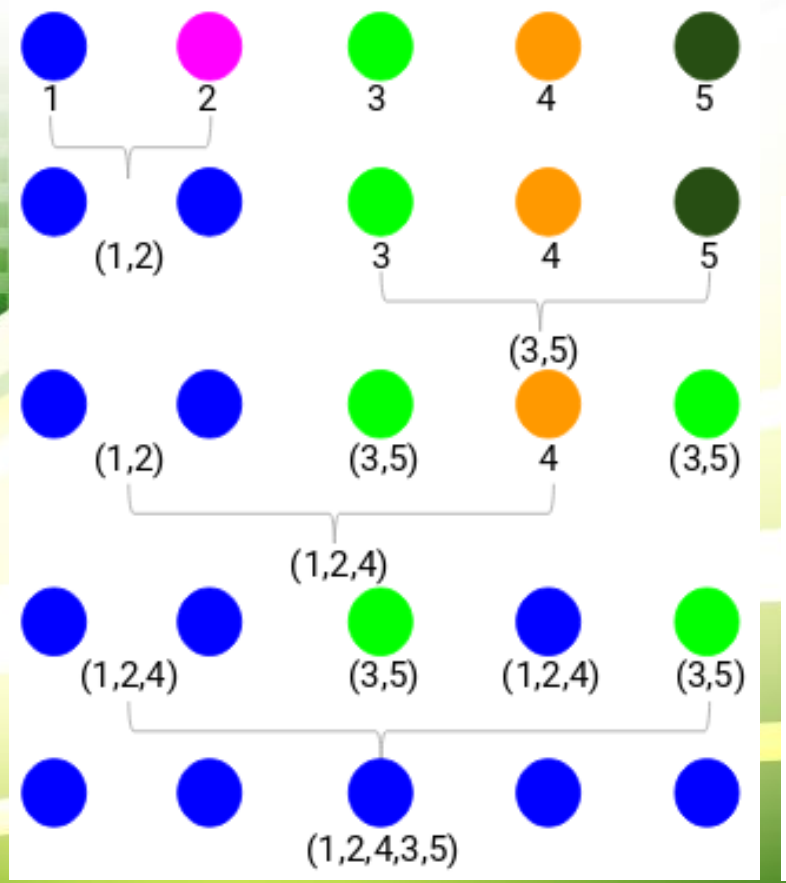
Agglomerative Hierarchical Clustering

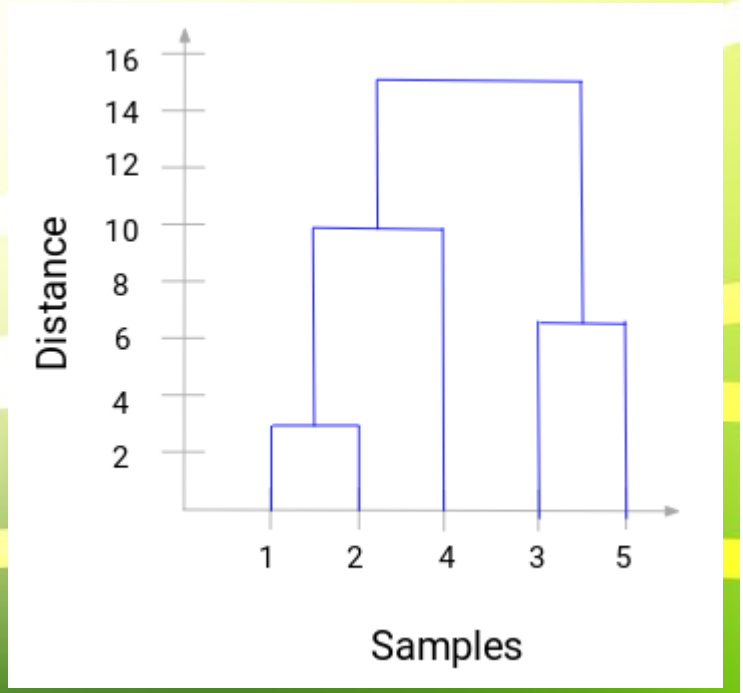
Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

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

Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

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





DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

- distances between all combinations of observations are summarized in **distance matrix**.
- diagonal values are excluded (pairs of same observation).
- distance matrix is usually symmetrical about diagonal *(distance between A & B is same as distance between B & A)*.
- The two closest observations are identified and are merged into a single cluster.
- Next iteration starts considering these two observations as single group (n-1 of observations).

Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	7.9	8.6	4.4	5.0	2.5
B	6.8	8.2	5.2	4.2	2.2
C	8.7	9.6	7.5	8.9	9.8
D	6.1	7.3	7.9	7.3	8.3
E	1.5	2.0	5.1	3.6	4.2
F	3.7	4.3	5.4	3.3	5.8
G	7.2	8.5	8.6	6.7	6.1
H	8.5	9.7	6.3	5.2	5.0
I	2.0	3.4	5.8	6.1	5.6
J	1.3	2.6	4.2	4.5	2.1
K	3.4	2.9	6.5	5.9	7.4
L	2.3	5.3	6.2	8.3	9.9
M	3.8	5.5	4.6	6.7	3.3
N	3.2	5.9	5.2	6.2	3.7

	A	B	C	D	...
A		$d_{A,B}$	$d_{A,C}$	$d_{A,D}$	...
B	$d_{B,A}$		$d_{B,C}$	$d_{B,D}$	...
C	$d_{C,A}$	$d_{C,B}$		$d_{C,D}$	...
D	$d_{D,A}$	$d_{D,B}$	$d_{D,C}$		...
...	...	...	...	...	

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

DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	7.9	8.6	4.4	5.0	2.5
B	6.8	8.2	5.2	4.2	2.2
C	8.7	9.6	7.5	8.9	9.8
D	6.1	7.3	7.9	7.3	8.3
E	1.5	2.0	5.1	3.6	4.2
F	3.7	4.3	5.4	3.3	5.8
G	7.2	8.5	8.6	6.7	6.1
H	8.5	9.7	6.3	5.2	5.0
I	2.0	3.4	5.8	6.1	5.6
J	1.3	2.6	4.2	4.5	2.1
K	3.4	2.9	6.5	5.9	7.4
L	2.3	5.3	6.2	8.3	9.9
M	3.8	5.5	4.6	6.7	3.3
N	3.2	5.9	5.2	6.2	3.7

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

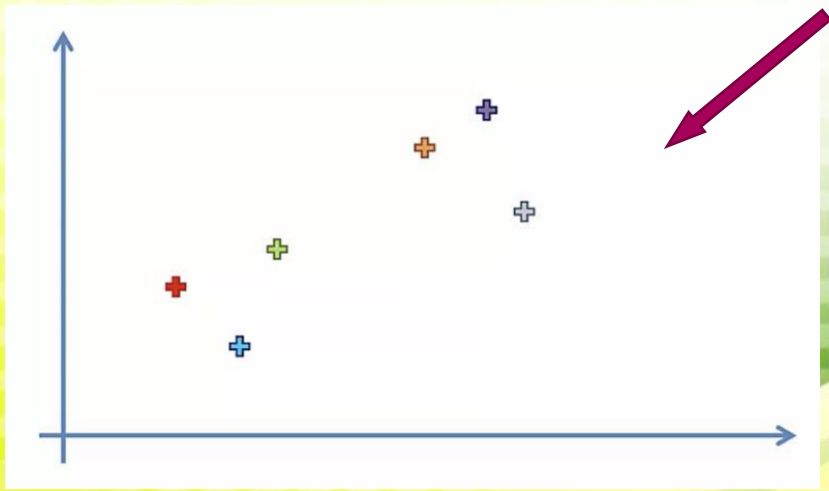
Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

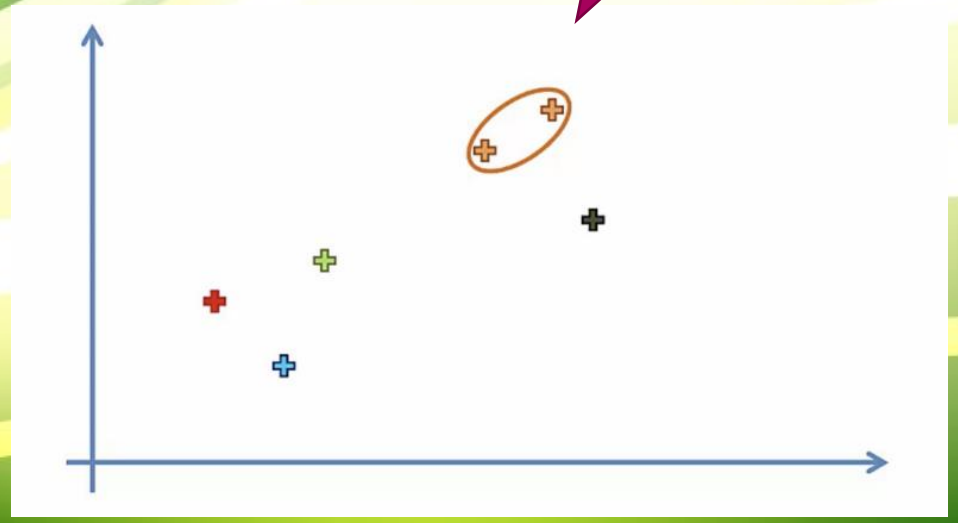
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A		0.282	1.373	1.2	1.272	0.978	1.106	0.563	1.178	1.189	1.251	1.473	0.757	0.793
B	0.282		1.423	1.147	1.113	0.82	1.025	0.56	1.064	1.065	1.144	1.44	0.724	0.7
C	1.373	1.423		0.582	1.905	1.555	0.709	0.943	1.468	1.995	1.305	1.076	1.416	1.378
D	1.2	1.147	0.582		1.406	1.092	0.403	0.808	0.978	1.543	0.797	0.744	1.065	0.974
E	1.272	1.113	1.905	1.406		0.476	1.518	1.435	0.542	0.383	0.719	1.223	0.797	0.727
F	0.978	0.82	1.555	1.092	0.476		1.191	1.039	0.57	0.706	0.595	1.076	0.727	0.624
G	1.106	1.025	0.709	0.403	1.518	1.191		0.648	1.163	1.624	1.033	1.108	1.148	1.051
H	0.563	0.56	0.943	0.808	1.435	1.039	0.648		1.218	1.475	1.169	1.315	0.984	0.937
I	1.178	1.064	1.468	0.978	0.542	0.57	1.163	1.218		0.659	0.346	0.727	0.553	0.458
J	1.189	1.065	1.995	1.543	0.383	0.706	1.624	1.475	0.659		0.937	1.344	0.665	0.659
K	1.251	1.144	1.305	0.797	0.719	0.595	1.033	1.169	0.346	0.937		0.64	0.774	0.683
L	1.473	1.44	1.076	0.744	1.223	1.076	1.108	1.315	0.727	1.344	0.64		0.985	0.919
M	0.757	0.724	1.416	1.065	0.797	0.727	1.148	0.984	0.553	0.665	0.774	0.985		0.196
N	0.793	0.7	1.378	0.974	0.727	0.624	1.051	0.937	0.458	0.659	0.683	0.919	0.196	

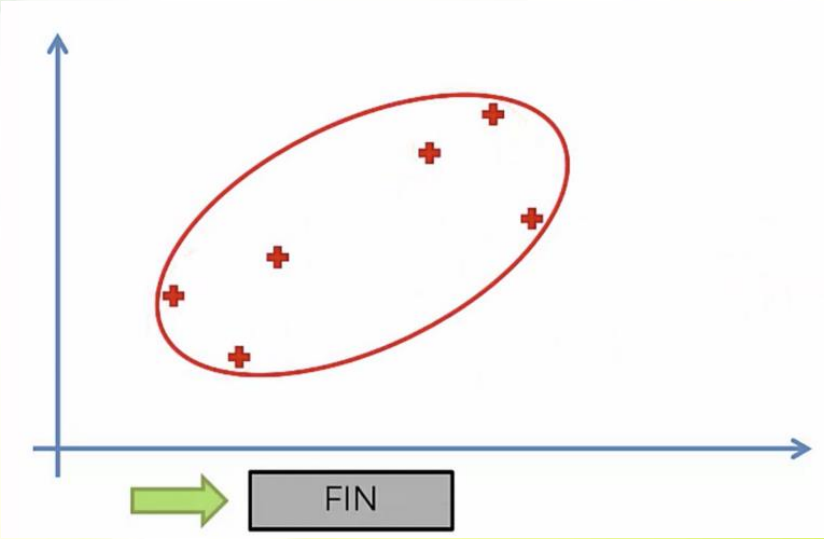
DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

- Euclidean distance for distance between 2 observations.
- For distance between individual observations and clusters, a **joining or linkage rule** is used



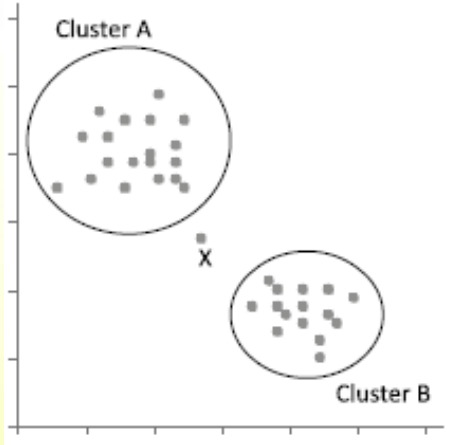


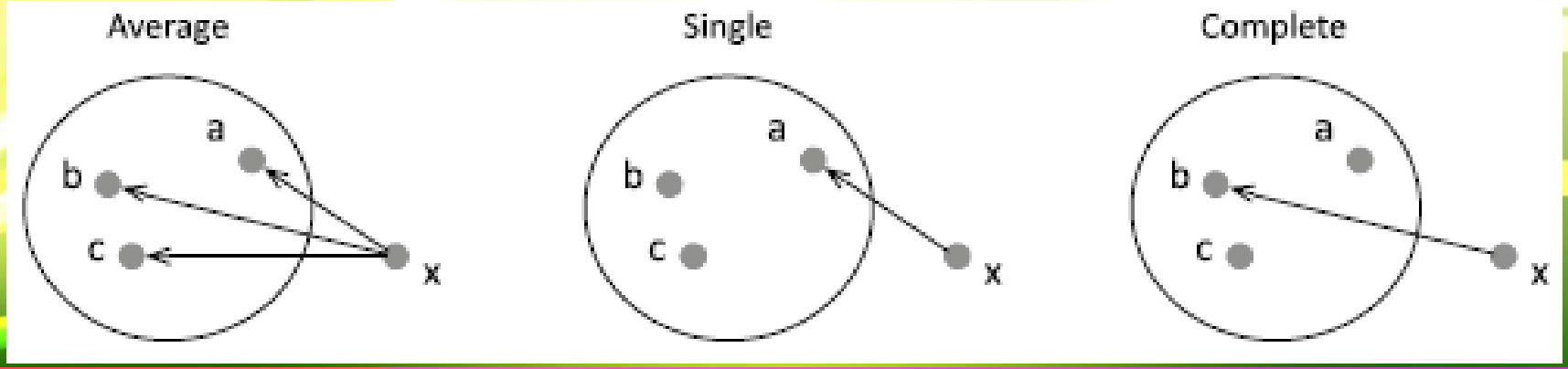


DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering **Linkage Rule** between single observation & a cluster:

- **Average linkage**: distance between all members of cluster and the observation under consideration are calculated → **average** is used for overall distance.
- **Single linkage**: distance between all members of cluster and the observation under consideration are calculated → **smallest** is selected.
- **Complete linkage**: distance between all members of cluster and the observation under consideration are calculated → **highest** is selected.

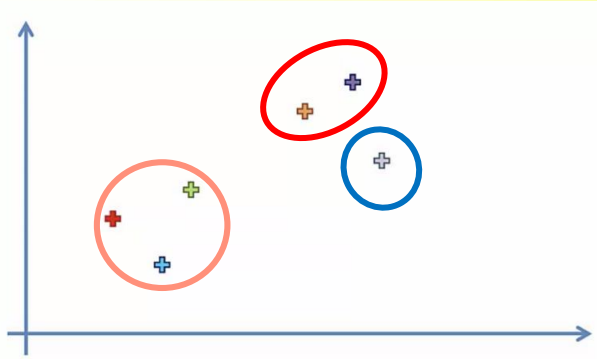


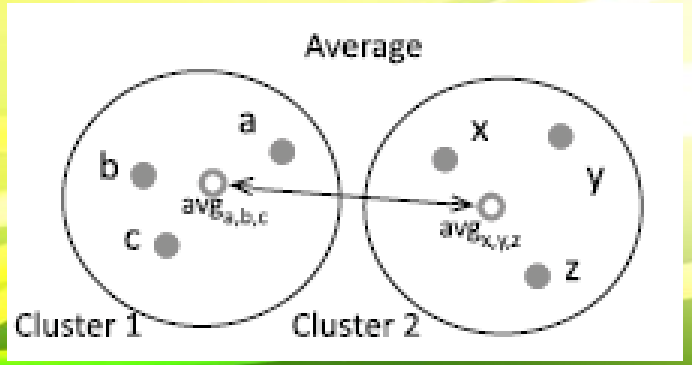


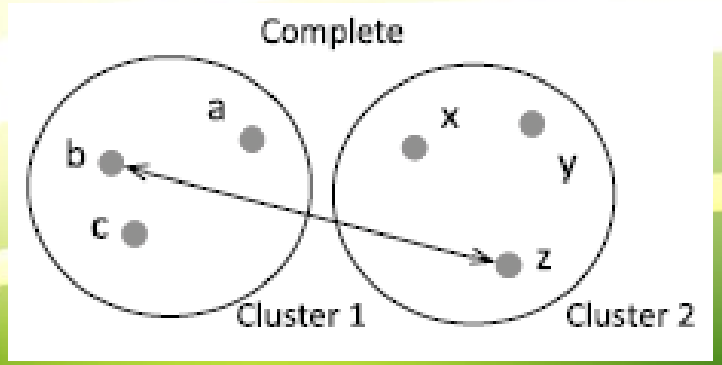
DATA ANALYSIS – HIERARCHICAL CLUSTERING

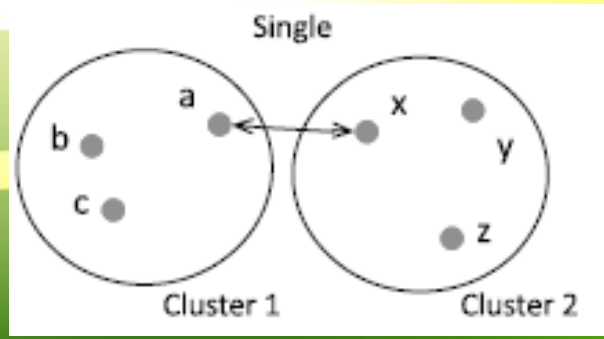
Agglomerative Hierarchical Clustering **Linkage Rule** between two clusters:

- **Average linkage**: distance between **average/centroid** values of both clusters.
- **Single linkage**: distance between all members of both clusters are calculated → **smallest** is selected.
- **Complete linkage**: distance between all members of both clusters are calculated → **highest** is selected.









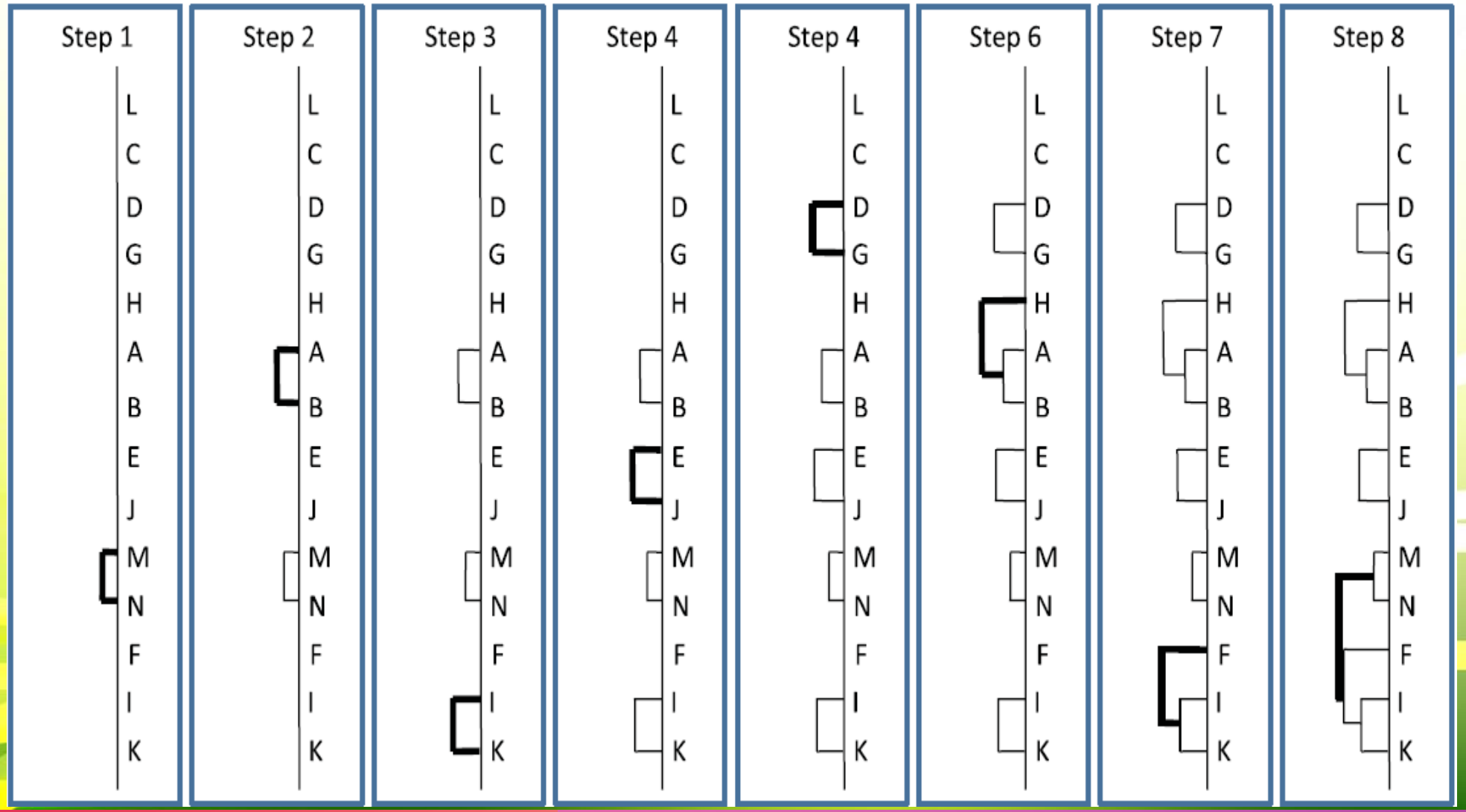
DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	7.9	8.6	4.4	5.0	2.5
B	6.8	8.2	5.2	4.2	2.2
C	8.7	9.6	7.5	8.9	9.8
D	6.1	7.3	7.9	7.3	8.3
E	1.5	2.0	5.1	3.6	4.2
F	3.7	4.3	5.4	3.3	5.8
G	7.2	8.5	8.6	6.7	6.1
H	8.5	9.7	6.3	5.2	5.0
I	2.0	3.4	5.8	6.1	5.6
J	1.3	2.6	4.2	4.5	2.1
K	3.4	2.9	6.5	5.9	7.4
L	2.3	5.3	6.2	8.3	9.9
M	3.8	5.5	4.6	6.7	3.3
N	3.2	5.9	5.2	6.2	3.7

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A		0.282	1.373	1.2	1.272	0.978	1.106	0.563	1.178	1.189	1.251	1.473	0.757	0.793
B	0.282		1.423	1.147	1.113	0.82	1.025	0.56	1.064	1.065	1.144	1.44	0.724	0.7
C	1.373	1.423		0.582	1.905	1.555	0.709	0.943	1.468	1.995	1.305	1.076	1.416	1.378
D	1.2	1.147	0.582		1.406	1.092	0.403	0.808	0.978	1.543	0.797	0.744	1.065	0.974
E	1.272	1.113	1.905	1.406		0.476	1.518	1.435	0.542	0.383	0.719	1.223	0.797	0.727
F	0.978	0.82	1.555	1.092	0.476		1.191	1.039	0.57	0.706	0.595	1.076	0.727	0.624
G	1.106	1.025	0.709	0.403	1.518	1.191		0.648	1.163	1.624	1.033	1.108	1.148	1.051
H	0.563	0.56	0.943	0.808	1.435	1.039	0.648		1.218	1.475	1.169	1.315	0.984	0.937
I	1.178	1.064	1.468	0.978	0.542	0.57	1.163	1.218		0.659	0.346	0.727	0.553	0.458
J	1.189	1.065	1.995	1.543	0.383	0.706	1.624	1.475	0.659		0.937	1.344	0.665	0.659
K	1.251	1.144	1.305	0.797	0.719	0.595	1.033	1.169	0.346	0.937		0.64	0.774	0.683
L	1.473	1.44	1.076	0.744	1.223	1.076	1.108	1.315	0.727	1.344	0.64		0.985	0.919
M	0.757	0.724	1.416	1.065	0.797	0.727	1.148	0.984	0.553	0.665	0.774	0.985		0.196
N	0.793	0.7	1.378	0.974	0.727	0.624	1.051	0.937	0.458	0.659	0.683	0.919	0.196	

DATA ANALYSIS – HIERARCHICAL CLUSTERING

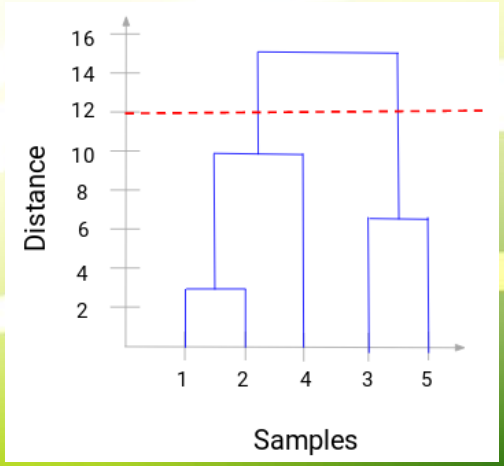


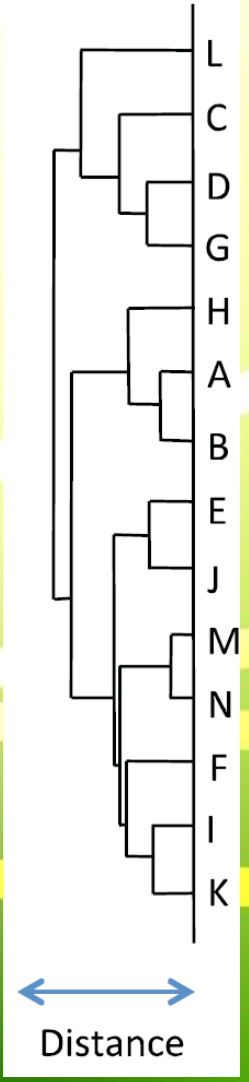
DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

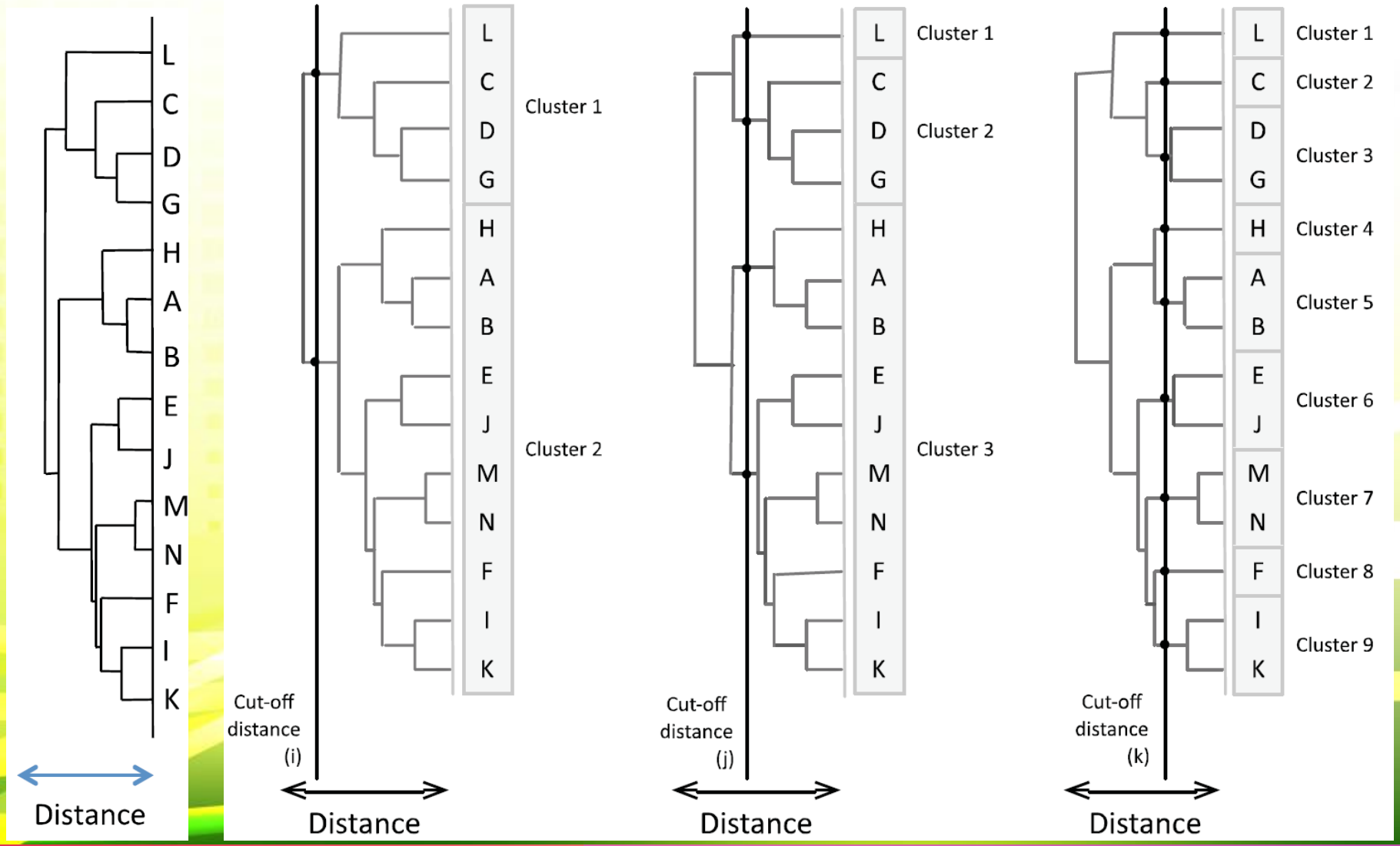
- When clustering completes, a tree called a dendrogram is generated showing similarity between observations and clusters.
- Horizontal length of lines reflects distance at which the cluster was formed.
- To divide dataset into series of distinct clusters, **threshold distance (cut-off) is selected**.
  - Where this distance intersects with a horizontal line on tree, a separate cluster is formed.

Student_ID	Marks
1	10
2	7
3	28
4	20
5	35





DATA ANALYSIS – HIERARCHICAL CLUSTERING



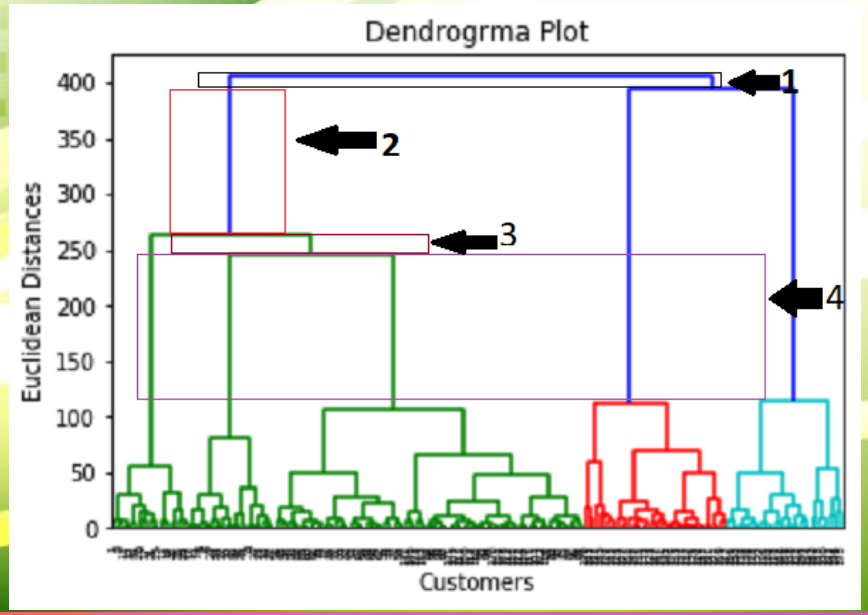
Distance cut-offs toward left result in fewer clusters with more diverse observations within each cluster.

Cut-offs toward right result in greater number of clusters with more similar observations within each cluster.

DATA ANALYSIS – HIERARCHICAL CLUSTERING

Finding optimal number of clusters using Dendrogram:

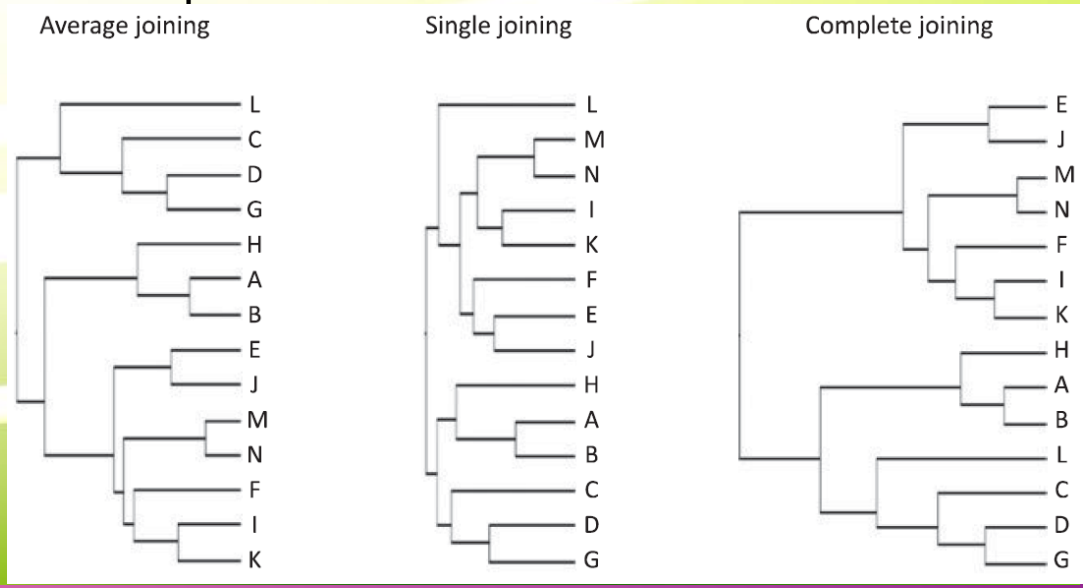
- Find **maximum vertical distance** that does not cut any horizontal bar.
- The 4th distance looks maximum; so, **number of clusters will be 5** (vertical lines in this range).
- The 2nd number also approximately equals the 4th distance *(but K-means algorithm also gives 5 clusters as result).*
- So, optimal number of clusters will be 5.



DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering

- Different joining/linkage rules change final hierarchical clustering.
- Hierarchical clustering of same set of observations using average, single, and complete linkage rules would be completely different.
  - barrier for merging observations and clusters is lowest with single linkage approach → clustering dendrogram may contain chains of clusters as well as clusters that are spread out.
  - barrier to joining clusters is highest with complete linkage → possible that an observation is closer to observations in other clusters than the cluster to which it has been assigned.
  - average linkage approach moderates the tendencies of single or complete linkage approaches

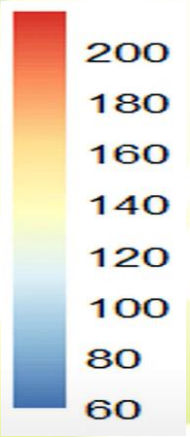


DATA ANALYSIS – HIERARCHICAL CLUSTERING

Agglomerative Hierarchical Clustering **Heatmap**

- **Heatmap** is a graphical representation of data where values are depicted by color.
- Cluster heatmaps is a heatmap where rows and columns of data matrix have been ordered according to the output from clustering.

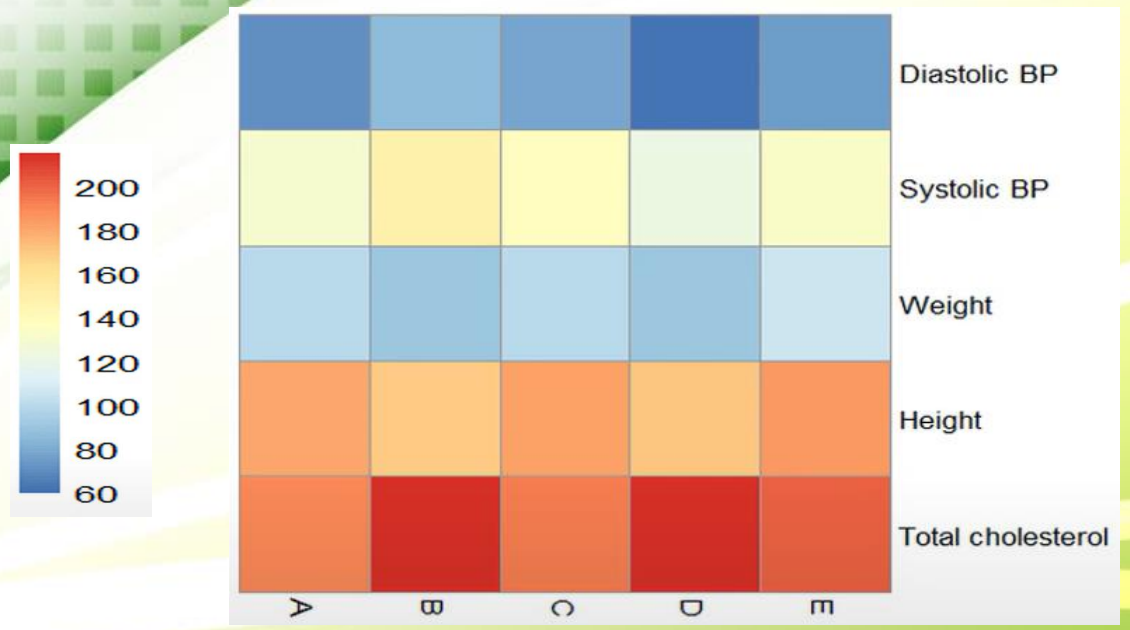
	A	B	C	D	E
Diastolic BP (mmHg)	70	86	78	60	75
Systolic BP (mmHg)	130	147	137	121	134
Weight (kg)	100	90	100	90	106
Height (cm)	180	170	181	171	185
Total cholesterol (mg/dl)	190	215	192	215	200



	A	B	C	D	E
Diastolic BP (mmHg)	70	86	78	60	75
Systolic BP (mmHg)	130	147	137	121	134
Weight (kg)	100	90	100	90	106
Height (cm)	180	170	181	171	185
Total cholesterol (mg/dl)	190	215	192	215	200

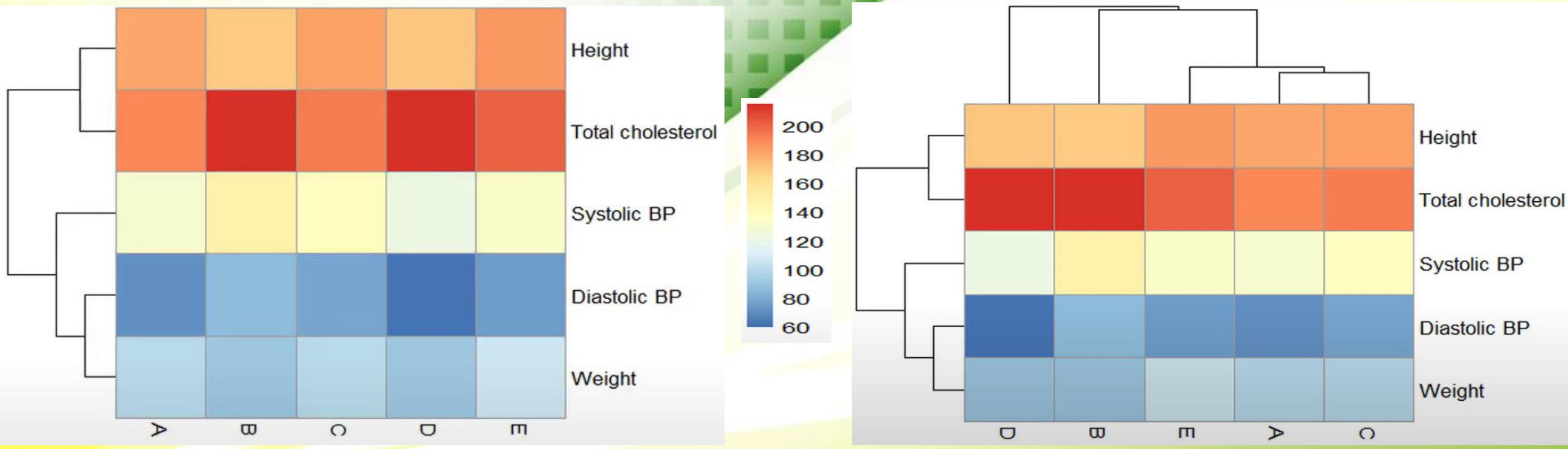
DATA ANALYSIS – HIERARCHICAL CLUSTERING

	A	B	C	D	E
Diastolic BP (mmHg)	70	86	78	60	75
Systolic BP (mmHg)	130	147	137	121	134
Weight (kg)	100	90	100	90	106
Height (cm)	180	170	181	171	185
Total cholesterol (mg/dl)	190	215	192	215	200



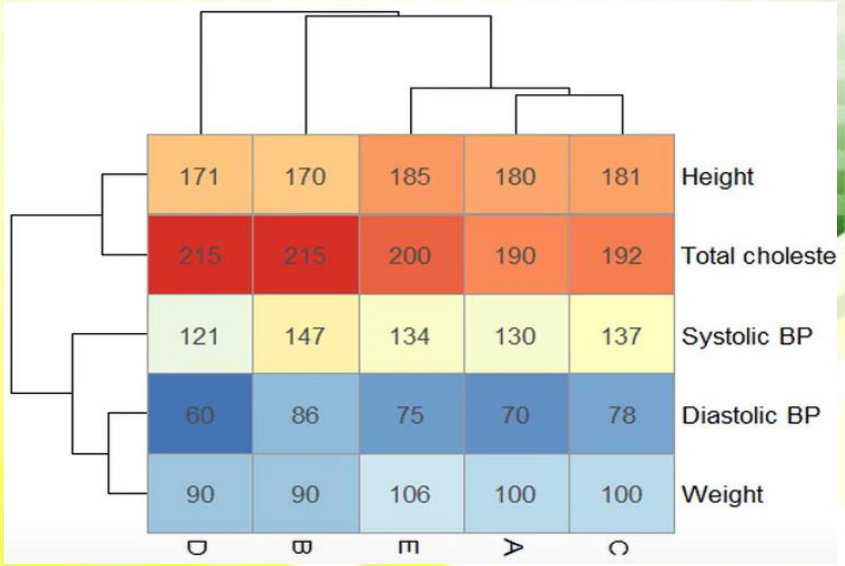
DATA ANALYSIS – HIERARCHICAL CLUSTERING

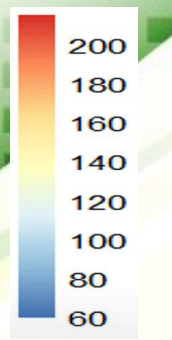
Agglomerative Hierarchical Clustering Heatmap



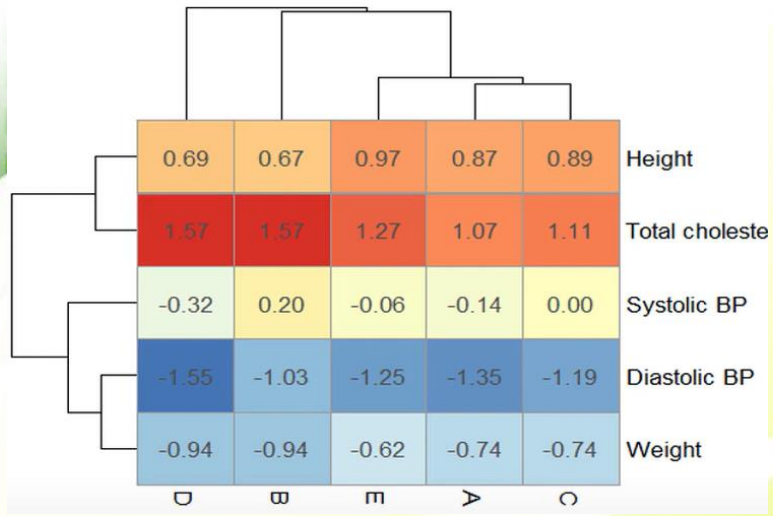
DATA ANALYSIS – HIERARCHICAL CLUSTERING

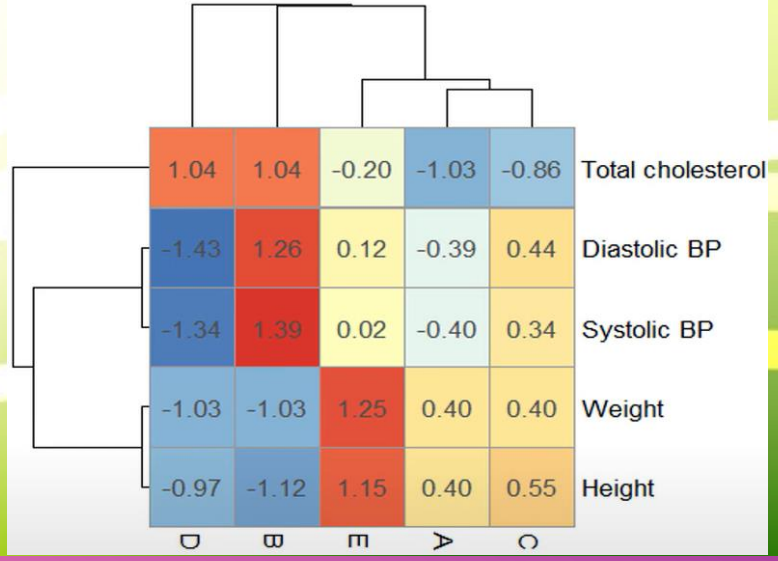
Normalization

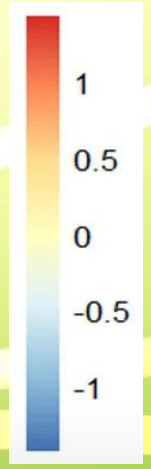




$$Z = \frac{x_i - \bar{x}}{SD}$$







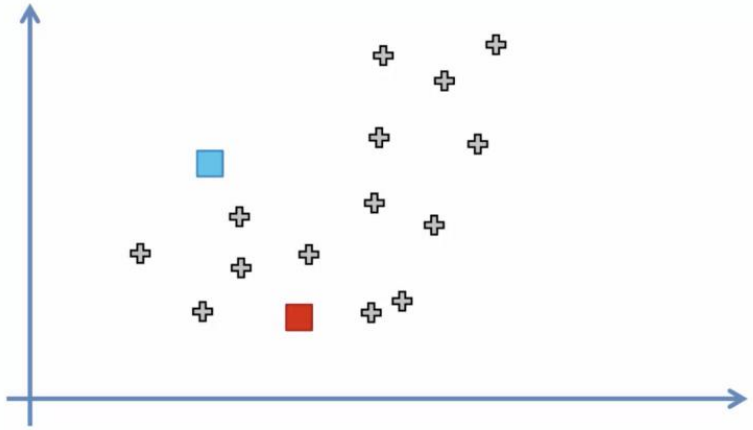
DATA ANALYSIS – K-MEANS CLUSTERING

- K-Means Clustering is an unsupervised machine learning algorithm.
- Non-hierarchical method for grouping a data set.
- K-Means attempts to classify data without having first been trained with labeled data.
  - Once algorithm has been run and groups are defined, any new data can be easily assigned to most relevant group.
- *Top-down* approach: starts with predefined number of clusters and assigns all observations to each of them.
- Computationally faster and can handle greater numbers of observations than agglomerative hierarchical clustering.

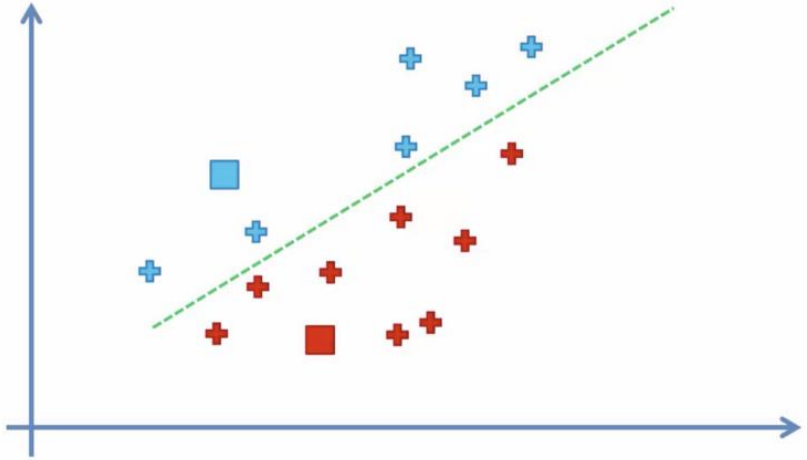
<u>Disadvantages:</u>

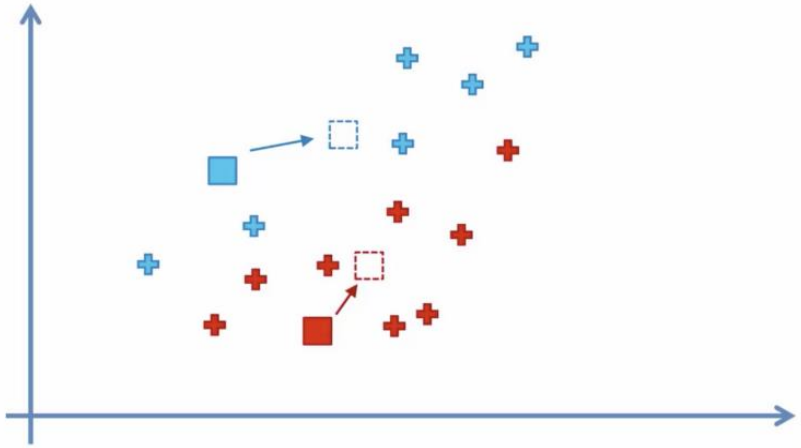
- Number of groups (K) must be specified before creating clusters and this number is not guaranteed to provide best partitioning of observations.
- when dataset contains many outliers, k-means may not create an optimal grouping; because reassignment of observations is based on closeness to cluster center and outliers pull cluster center in their direction *(WRONG)*.
- No hierarchical organization is generated using k-means clustering and hence there is no ordering of individual observations.

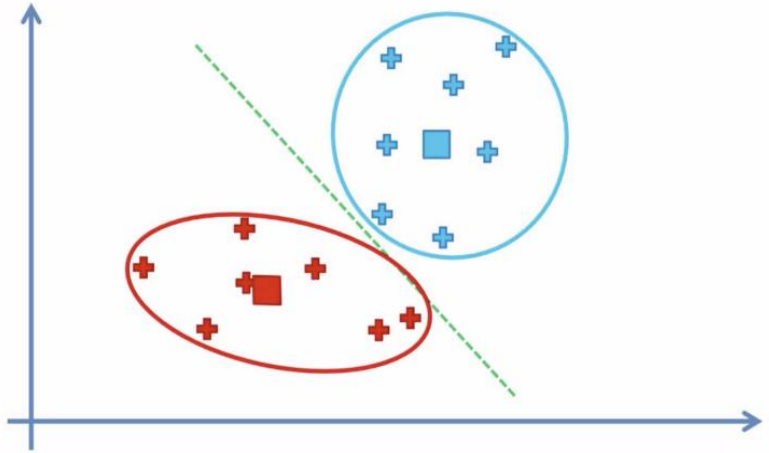
DATA ANALYSIS – K-MEANS CLUSTERING



1. Select **K** (= 2) random points as cluster centers called centroids.
2. Assign each data point to the closest cluster by calculating its distance with respect to each centroid.
3. Determine the new cluster center by computing the average of the assigned points.
4. Repeat steps 2 and 3 until none of the cluster assignments change.

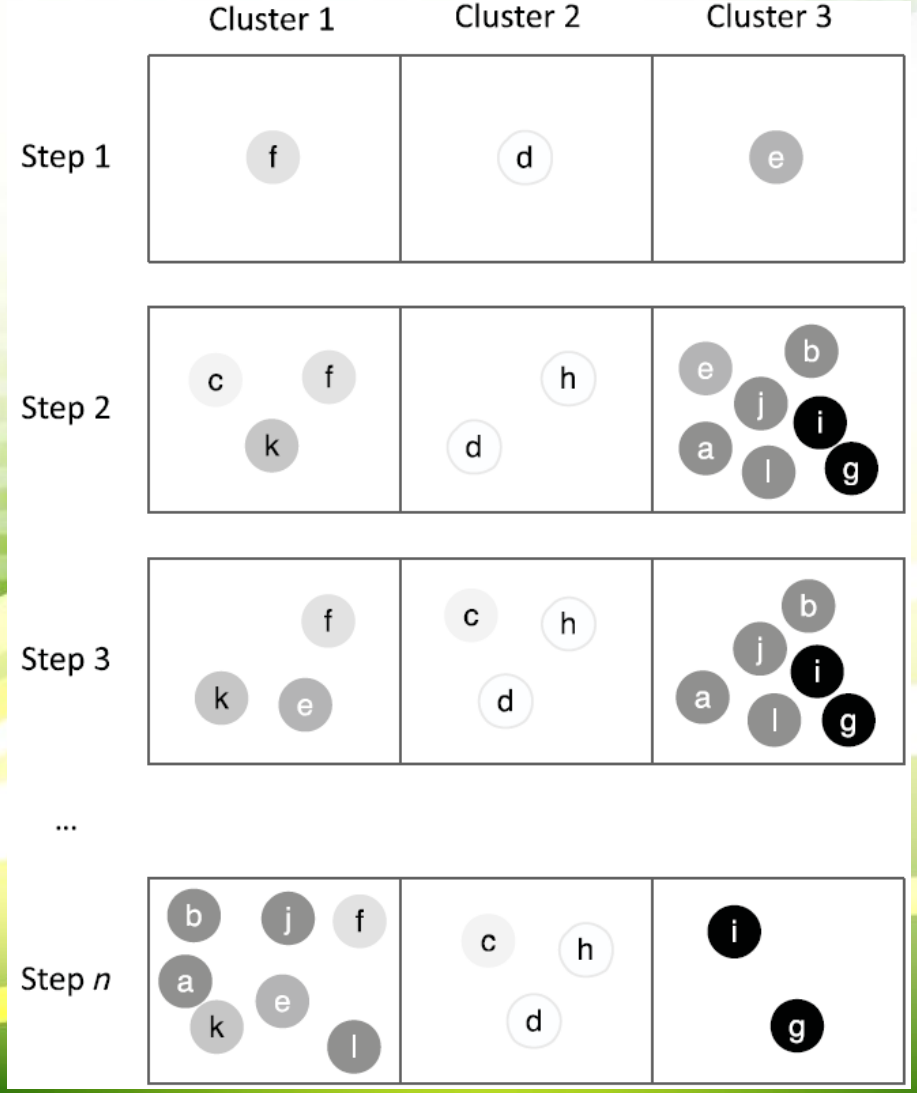






DATA ANALYSIS – K-MEANS CLUSTERING

Example

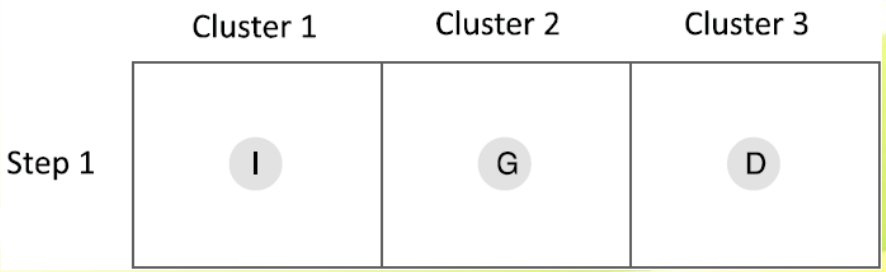


DATA ANALYSIS – K-MEANS CLUSTERING

Example

Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	7.9	8.6	4.4	5.0	2.5
B	6.8	8.2	5.2	4.2	2.2
C	8.7	9.6	7.5	8.9	9.8
D	6.1	7.3	7.9	7.3	8.3
E	1.5	2.0	5.1	3.6	4.2
F	3.7	4.3	5.4	3.3	5.8
G	7.2	8.5	8.6	6.7	6.1
H	8.5	9.7	6.3	5.2	5.0
I	2.0	3.4	5.8	6.1	5.6
J	1.3	2.6	4.2	4.5	2.1
K	3.4	2.9	6.5	5.9	7.4
L	2.3	5.3	6.2	8.3	9.9
M	3.8	5.5	4.6	6.7	3.3
N	3.2	5.9	5.2	6.2	3.7

1. Select **K** (= 3) random points (I, G, D) as cluster centers called centroids.

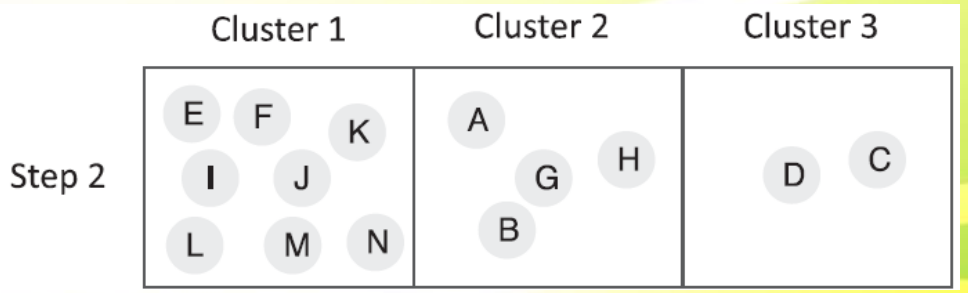


DATA ANALYSIS – K-MEANS CLUSTERING

Example

Name	Cluster 1	Cluster 2	Cluster 3	Cluster Assignment
A	1.178	1.106	1.2	2
B	1.064	0.025	1.147	2
C	1.468	0.709	0.582	3
E	0.542	1.518	1.406	1
F	0.57	1.191	1.092	1
H	1.218	0.648	0.808	2
J	0.659	1.624	1.543	1
K	0.346	1.033	0.797	1
L	0.727	1.108	0.744	1
M	0.553	1.148	1.065	1
N	0.458	1.051	0.974	1

2. Assign each data point to the closest cluster by calculating its distance with respect to each centroid.



DATA ANALYSIS – K-MEANS CLUSTERING

Example

Cluster 1 Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
E	1.5	2	5.1	3.6	4.2
F	3.7	4.3	5.4	3.3	5.8
I	2	3.4	5.8	6.1	5.6
J	1.3	2.6	4.2	4.5	2.1
K	3.4	2.9	6.5	5.9	7.4
L	2.3	5.3	6.2	8.3	9.9
M	3.8	5.5	4.6	6.7	3.3
N	3.2	5.9	5.2	6.2	3.7
Average (Center)	2.65	3.99	5.38	5.58	5.25

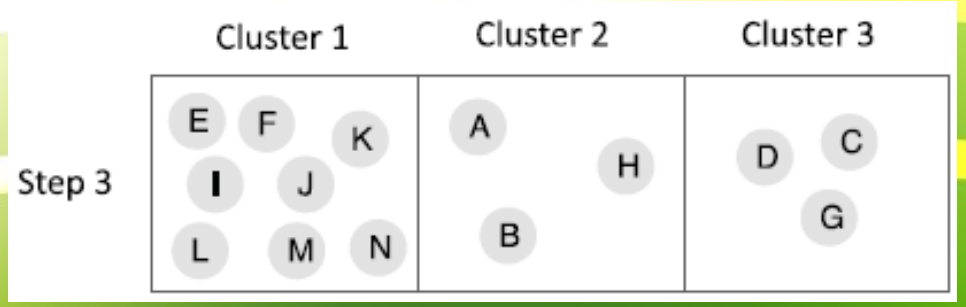
Cluster 2 Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
A	7.9	8.6	4.4	5	2.5
B	6.8	8.2	5.2	4.2	2.2
G	7.2	8.5	8.6	6.7	6.1
H	8.5	9.7	6.3	5.2	5
Average (Center)	7.60	8.75	6.13	5.28	3.95

Cluster 3 Name	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
C	8.7	9.6	7.5	8.9	9.8
D	6.1	7.3	7.9	7.3	8.3
Average (Center)	7.40	8.45	7.70	8.10	9.05

3. Determine the new cluster center by computing the average of the assigned points.

New center of cluster 1:
{ Variable 1 = 2.65; Variable 2 = 3.99;
Variable 3 = 5.38; Variable 4 = 5.58;
Variable 5 = 5.25 }

4. Repeat steps 2 and 3 until none of the cluster assignments change.



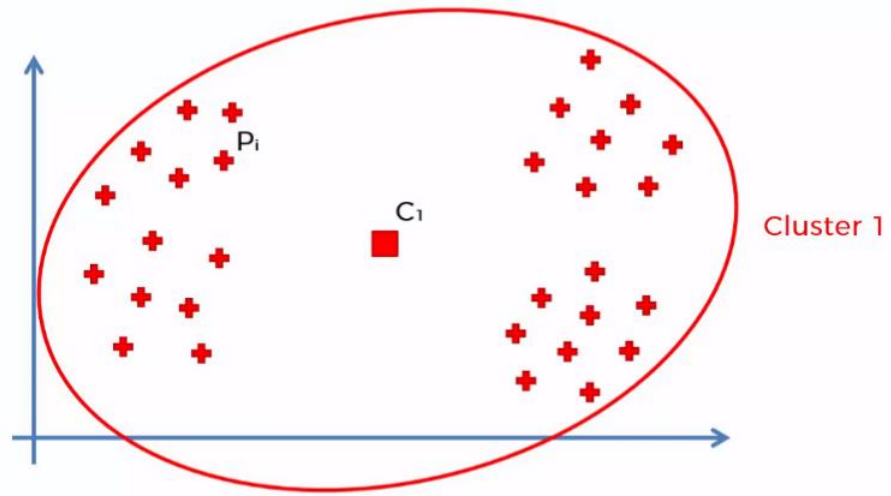
DATA ANALYSIS – K-MEANS CLUSTERING

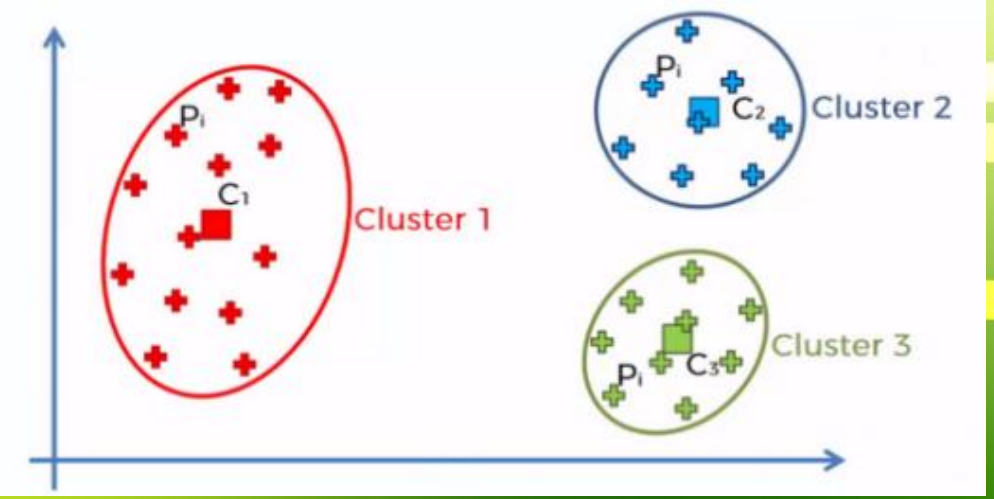
Choosing the right number of clusters

- Correct idea of number of cluster is very important for model performance.
- Choosing right number of cluster is very difficult, at first.
- Within Cluster Sum of Squares (WCSS) helps finding 'K' value.
  - WCSS: sum of squares of distances of data points in each and every cluster from its centroid.
  - Main idea is to minimize distance between data points and centroid of clusters.
- **Example**, computed WCSS for K=1 is greater than WCSS calculated for *K=3*.

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

Where x_i = data point and c_i = closest point to centroid

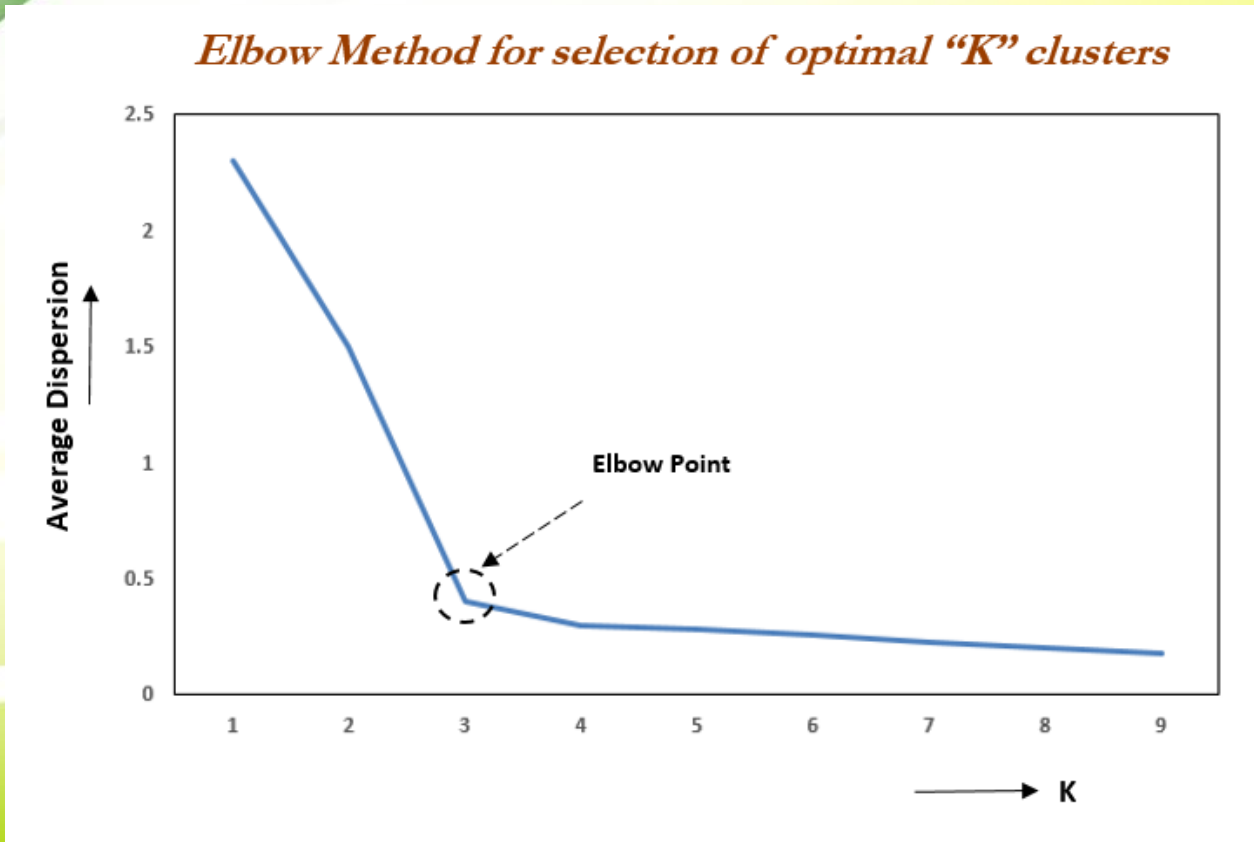




DATA ANALYSIS – K-MEANS CLUSTERING

Choosing right number of clusters

- Most common technique is **elbow method**.
- Elbow method is used to determine optimal number of clusters in K-means clustering.
- Elbow method plots the value of cost function produced by different values of K.
- Value of K at which improvement in distortion declines the most is called the elbow.
- At this point, STOP division of data into further clusters.



DATA ANALYSIS –CLUSTERING

Given here are 5 students marks for two subjects. Appropriately Cluster them into two groups of "good performer" and "average performer".

1. **For Hierarchical clustering, consider average measure for linkage rule.**
2. **For K-means, consider Student S2 and S3 marks for initial centroids.**

	Marks (Out of 10)	
Student	**Sub-A**	**Sub-B**
S1	8	5
S2	6	4
S3	5	6
S4	9	7
S5	4	3

DATA ANALYSIS –CLUSTERING

1. For Hierarchical clustering, consider average measure for linkage rule.

	Marks (Out of 10)	
Student	Sub-A	Sub-B
S1	8	5
S2	6	4
S3	5	6
S4	9	7
S5	4	3

	S1	S2	S3	S4	S5
S1	0				
S2		0			
S3			0		
S4				0	
S5					0

Proximity/distance matrix

DATA ANALYSIS –CLUSTERING

2. For K-means, consider Student S2 and S3 marks for initial centroids.

	Marks (Out of 10)	
Student	**Sub-A**	**Sub-B**
S1	8	5
S2	6	4
S3	5	6
S4	9	7
S5	4	3

	Distance		Cluster
	Mean-1	**Mean-2**	
S1			
S2			
S3			
S4			
S5			

Proximity/distance matrix

DATA ANALYSIS - GROUPING

- Association rules method groups observations and attempts to discover links or associations between different attributes of the group.
- unsupervised grouping method
- **Association rule learning**: procedure to check for dependency of one data item on another data item and maps accordingly so that it can help in be more profitable analysis.
  - *If a customer buys bread, (s)he's 70% likely of buying milk.*
- **Association Rule:** simple If/Then statements that help discover relationships between seemingly independent relational databases or other data repositories.
  - Ways to find patterns in data; Helps in finding features (dimensions) which occur together (correlated).
  - Employed in *Market Basket analysis*, Web usage mining, Medical Diagnosis, etc.

IF the customer is age 18 AND
the customer buys paper AND
the customer buys a hole punch
THEN the customer buys a binder

A package of products can be created for college students.

DATA ANALYSIS - GROUPING

Advantages

- rules are easy to understand

Limitations

- forces to either restrict analysis to variables that are categorical or convert continuous variables to categorical variables.
- Generating rules can be computationally expensive (especially when dataset has many variables or many possible values per variable)
- There are ways to make analysis run faster but they often compromise final results.
- can generate large numbers of rules that must be prioritized and interpreted.

DATA ANALYSIS - GROUPING

Grouping by Combinations of Values

- Increasing number of variables or number of possible values for each variable or both → increases number of groups
- When number of groups becomes too large → impractical to generate all combinations

Customer ID	Gender	Purchase
932085	Male	Television
596720	Female	Camera
267375	Female	Television

Group Number	Count	Gender	Purchase
Group 1	16,099	Male	Camera or Television
Group 2	15,513	Female	Camera or Television
Group 3	16,106	Male or Female	Camera
Group 4	15,506	Male or Female	Television
Group 5	7,889	Male	Camera
Group 6	8,210	Male	Television
Group 7	8,217	Female	Camera
Group 8	7,296	Female	Television

Group Number	Count	Gender	Purchase	Income
Group 1	16,099	Male	Camera or Television	Below $50K or Above $50K
Group 2	15,513	Female	Camera or Television	Below $50K or Above $50K
Group 3	16,106	Male or Female	Camera	Below $50K or Above $50K
Group 4	15,506	Male or Female	Television	Below $50K or Above $50K
Group 5	15,854	Male or Female	Camera or Television	Below $50K
Group 6	15,758	Male or Female	Camera or Television	Above $50K
Group 7	7,889	Male	Camera	Below $50K or Above $50K
Group 8	8,210	Male	Television	Below $50K or Above $50K
Group 9	8,549	Male	Camera or Television	Below $50K
Group 10	7,550	Male	Camera or Television	Above $50K
Group 11	8,217	Female	Camera	Below $50K or Above $50K
Group 12	7,296	Female	Television	Below $50K or Above $50K
Group 13	7,305	Female	Camera or Television	Below $50K
Group 14	8,208	Female	Camera or Television	Above $50K
Group 15	8,534	Male or Female	Camera	Below $50K
Group 16	7,572	Male or Female	Camera	Above $50K
Group 17	7,320	Male or Female	Television	Below $50K
Group 18	8,186	Male or Female	Television	Above $50K
Group 19	4,371	Male	Camera	Below $50K
Group 20	3,518	Male	Camera	Above $50K
Group 21	4,178	Male	Television	Below $50K
Group 22	4,032	Male	Television	Above $50K
Group 23	4,163	Female	Camera	Below $50K
Group 24	4,054	Female	Camera	Above $50K
Group 25	3,142	Female	Television	Below $50K
Group 26	4,154	Female	Television	Above $50K

DATA ANALYSIS – ASSOCIATION RULE

- Association rule learning works on the concept of If (**antecedent)** and Else Statement (**Consequent).**
  - *If a customer buys bread, (s)he's 70% likely of buying milk.*

- ***Single cardinality***: Association or relation between two items
  - If number of items increases, then cardinality also increases accordingly.

- **Important association metrics:**
  - **Support** (frequency)
  - **Confidence** (paired/conditional occurrence)
  - **Lift** (strength of rule)

DATA ANALYSIS – ASSOCIATION RULE

- **Support:** frequency of an event in dataset.
  - ***Supp(e) = Freq(e) / Total transaction***
- **Confidence:** indicates how often the rule has been found to be true *(how often items X and Y occur together in dataset when occurrence of X is already given).*
  - ***Confidence(X,Y) = Freq(X,Y) / Freq(X)*** **Confidence = Group support / IF-part support**
- **Lift:** strength of any rule. *Ratio of observed support measure and expected support if X and Y are independent of each other.*
  - ***Lift(R) = Supp(X,Y) / (Supp(X)*Supp(Y))*** **Lift = Confidence / THEN-part support**
  - **Lift= 1**: probability of occurrence of antecedent and consequent is independent of each other.
  - **Lift>1**: determines the degree to which two itemsets are dependent to each other.
  - **Lift<1**: tells that rule body and rule head appear less often together than expected (negative effect on occurrence).

DATA ANALYSIS – ASSOCIATION RULE

Supp(e) = Freq(e) / Total transaction

Confidence(X,Y) = Freq(X,Y) / Freq(X)

- **Confidence = Group support / IF-part support**

*Lift(R) = Supp(X,Y) / (Supp(X)*Supp(Y))*

- **Lift = Confidence / THEN-part support**

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Rule 1

IF Class of work is Private AND
Education is Doctorate
THEN Income is <=50K

Rule 2

IF Class of work is Private AND
Education is Doctorate
THEN Income is >50K

Total observations: 32,561
Class of work is Private: 22,696 observations
Education is Doctorate: 413 observations
Class of work is private and Education is Doctorate: 181 observations
Income is <=50K: 24,720 observations
Income is >50K: 7841 observations

Association Rule Summary Table

	Rule 1	Rule 2
Count	49	132
Support	0.0015	0.0041
Confidence	0.27	0.73
Lift	0.36	3.03

DATA ANALYSIS - ASSOCIATION RULE

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

	Beer	Bread	Milk	Diaper	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

One-hot encoded Table

	Rule 1	Rule 2
Count	49	132
Support	0.0015	0.0041
Confidence	0.27	0.73
Lift	0.36	3.03

Association Rule Summary Table

$$Support = \frac{frq(X,Y)}{N}$$

$$Rule: X \Rightarrow Y \rightarrow Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

{Diaper, Beer} → Milk
- Support = 2/5, Confidence = 2/3

{Milk} → {Diaper, Beer}
- Support = 2/5, Confidence = 2/4

{Milk, Diaper} → Bread
- Support = 2/5, Confidence = 2/3

DATA ANALYSIS - ASSOCIATION RULE

Example: For the computer accessories purchase transaction, prepare the Association Rule Summary table by calculating the Support, Confidence & Life for the following Association rules.

1. **If customer purchases Laptop, then (s)he also buys Monitor.**
2. **If customer purchases Monitor & Tablet, then (s)he also buys headset.**
3. **If customer purchases Laptop & Monitor, then (s)he also buys headset.**
4. **If customer purchases Laptop & Monitor, then (s)he also buys Printer.**

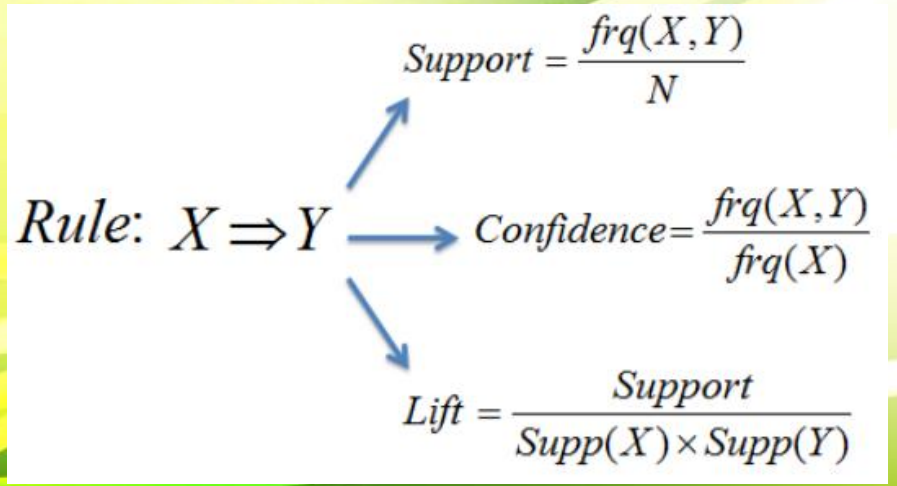


Table 1. Market basket transactions

Transaction ID	*Items Bought*
1	{*Laptop, Printer, Tablet, Headset*}
2	{*Printer, Monitor, Tablet*}
3	{*Laptop, Printer, Tablet, Headset*}
4	{*Laptop, Monitor, Tablet, Headset*}
5	{*Printer, Monitor, Tablet, Headset*}
6	{*Printer, Tablet, Headset*}
7	{*Monitor, Tablet*}
8	{*Laptop, Printer, Monitor*}
9	{*Laptop, Tablet, Headset*}
10	{*Printer, Tablet*}

DATA ANALYSIS - ASSOCIATION RULE

Transaction ID	Items List
1	Cookies, Egg, Milk, Sandwich
2	Bottled Water, Burger, Chicken, Egg, Pizza, Salad
3	Beacon, Bottled Water, Egg, Sandwich, Yogurt
4	Burger, Pie, Pizza, Salad, Soda
5	Burger, Ice Cream, Pie, Pizza, Salad, Soda
6	Chocolate Shake, Cookies, Egg, Milk, Sandwich
7	Beacon, Chocolate Shake, Cookies, Milk, Yogurt
8	Bottled Water, Burger, Chicken, Chocolate Shake, Egg, Pie, Pizza, S
9	Beacon, Bottled Water, Egg, Milk, Pizza, Salad, Yogurt
10	Chocolate Shake, Cookies, Egg, Milk, Sandwich
11	Beacon, Burger, Salad
12	Cookies, Egg, Milk, Sandwich, Yogurt
13	Beacon, Bottled Water, Egg, Pie, Pizza, Sandwich
14	Cookies, Egg, Milk, Sandwich
15	Bottled Water, Burger, Chicken, Egg, Pie, Pizza, Salad

LHS	RHS	rules	support	confidence	lift
Ice Cream	Soda	{Ice Cream} => {Soda}	0.07	1.00	5.00
Soda	Ice Cream	{Soda} => {Ice Cream}	0.07	0.33	5.00
Ice Cream	Pie	{Ice Cream} => {Pie}	0.07	1.00	3.00
Pie	Ice Cream	{Pie} => {Ice Cream}	0.07	0.20	3.00
Ice Cream	Burger	{Ice Cream} => {Burger}	0.07	1.00	2.50
Burger	Ice Cream	{Burger} => {Ice Cream}	0.07	0.17	2.50
Ice Cream	Salad	{Ice Cream} => {Salad}	0.07	1.00	2.14
Salad	Ice Cream	{Salad} => {Ice Cream}	0.07	0.14	2.14
Ice Cream	Pizza	{Ice Cream} => {Pizza}	0.07	1.00	2.14
Pizza	Ice Cream	{Pizza} => {Ice Cream}	0.07	0.14	2.14
Soda	Chicken	{Soda} => {Chicken}	0.07	0.33	1.67
Chicken	Soda	{Chicken} => {Soda}	0.07	0.33	1.67
Soda	Chocolate Shake	{Soda} => {Chocolate Shake}	0.07	0.33	1.25
Chocolate Shake	Soda	{Chocolate Shake} => {Soda}	0.07	0.25	1.25
Soda	Pie	{Soda} => {Pie}	0.20	1.00	3.00
Pie	Soda	{Pie} => {Soda}	0.20	0.60	3.00
Soda	Burger	{Soda} => {Burger}	0.20	1.00	2.50
Burger	Soda	{Burger} => {Soda}	0.20	0.50	2.50
Soda	Bottled Water	{Soda} => {Bottled Water}	0.07	0.33	0.83
Bottled Water	Soda	{Bottled Water} => {Soda}	0.07	0.17	0.83
Soda	Salad	{Soda} => {Salad}	0.20	1.00	2.14
Salad	Soda	{Salad} => {Soda}	0.20	0.43	2.14
Soda	Pizza	{Soda} => {Pizza}	0.20	1.00	2.14

DATA ANALYSIS – ASSOCIATION RULE

- **Market Basket Analysis (MBA)**: popular examples and applications of association rule mining.
- Technique used by big retailers to determine association between items, as a marketing strategy.
  - *If a customer buys bread, (s)he most likely can also buy butter, eggs, or milk → these products are stored within a shelf or mostly nearby.*
- **Steps in MBA:**
  - *Establish possible Rules.*
  - *Calculate support, confidence, lift for each rule.*
  - *Validate the Rule(s).*

$$Rule: X \Rightarrow Y \quad Support = \frac{frq(X,Y)}{N} \quad Confidence = \frac{frq(X,Y)}{frq(X)} \quad Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

DATA ANALYSIS – ASSOCIATION RULE

BENEFITS OF MARKET BASKET ANALYSIS

- Customer Behavior analysis
- Optimization of in-store operations & stock management.
- Campaigns and promotions
- Item Recommendations
- Increasing market share

DATA ANALYSIS – ASSOCIATION RULE

- **(Item) Recommendation system** makes prediction for user consumption.
- **Content-based** approach focused on information of items' own features, rather than using users' interactions and feedbacks.
  - *Example, movie attributes: genre, year, director, actor etc.*
- **Collaborative Filtering:** focused on users' historical preference.
  - Based on user's own past preference.
    - Basic assumption: *users who have agreed in past tend to also agree in future.*
    - *User regularly watches scientific videos → (s)he's recommended other such videos.*
  - Based on other users' (majority) past preference.
    - Collecting preferences or taste information from many users (collaborating).
    - Basic assumption: *if many have agreed in past, others will also agree in future.*
    - *Many users are watching budget analysis video → many other users also recommended same/such videos.*

DATA ANALYSIS – ASSOCIATION RULE

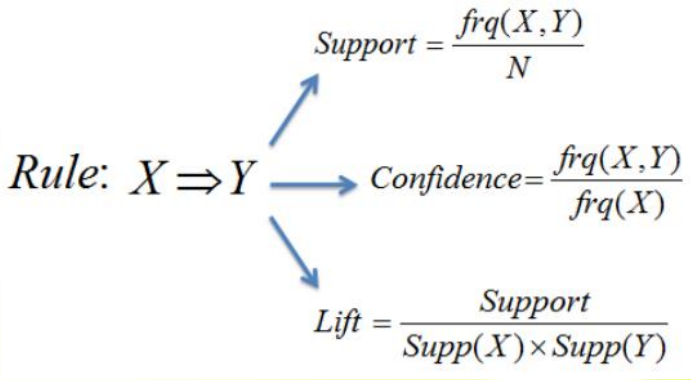
- Collaborative Filtering focuses on users' historical preference, for Item Recommendation.
- User preference usually expressed by two categories.

- **Explicit Rating** given by user to an item on a sliding scale.
  - Example: 5 stars for a movie
  - Most direct feedback from users to show how much they like an item.

- **Implicit Rating** suggests users preference indirectly.
  - Example: page views, clicks, purchase records, whether or not listen to music track, etc.
  - Shows user's involvement/attention with the "content", time spent, etc. → value of the "content"

DATA ANALYSIS – ASSOCIATION RULE

- **Steps in Grouping Analysis**:
  - Association rule learning → *Establish possible Rules.*
  - *Calculate support, confidence, lift for each rule.*
  - *Validate the Rule(s) with threshold/acceptance level.*



- **Types of Association rule learning**:
  - *Apriori Algorithm*
  - *F-P (Frequent Pattern) Growth Algorithm*
  - *Eclat (Equivalence Class Transformation) algorithm*

DATA ANALYSIS - APRIORI

- **Apriori algorithm** uses frequently purchased item-sets to **generate association rules**.
  - Used in market basket analysis.
  - Helps to find frequent item-sets in transactions and identifies association rules between these items.
- Named Apriori because it uses prior knowledge of frequent itemset properties.
- Limitation is *frequent itemset generation* → needs to scan database many times leading to increased time and reduce performance (computationally costly step).
- Basic Assumption:
  - *All subsets of a frequent itemset must be frequent.*
  - *If an itemset is infrequent, all its supersets will be infrequent.*

TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Apriori Algorithm

TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum Support= 2, Minimum Confidence= 50%

Step-1: Calculating C1 and F1:

- Create a table that contains each itemset's support count (frequency of each itemset individually in dataset). This table is called the **Candidate set or C1.**
- Take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the **frequent itemset F1.**

Itemset	Support_Count
A	6
B	7
C	5
D	2
~~E~~	~~1~~

Step-2: Candidate Generation C2, and F2:

- Create the pair of the itemsets of F1 in the form of subsets.
- Again find support count for these pairs from the main transaction table of datasets **(C2).**
- Compare the C2 Support count with minimum support count, and eliminate the itemsets with less support count **(in F2).**

Itemset	Support_Count
{A, B}	4
{A,C}	4
~~{A, D}~~	~~1~~
{B, C}	4
{B, D}	2
~~{C, D}~~	~~0~~

Apriori Algorithm

TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum Support= 2, Minimum Confidence= 50%

Step-3: Candidate generation C3, and F3:

- Repeat the same two processes, but with subsets of three itemsets together.
- Create C3 and F3 accordingly.

Step-4: Finding the association rules for the subsets:

- To generate the association rules, first create a new table with all possible rules from the occurred combination {A, B.C}.
- For all the rules, calculate the Confidence using required formula.
- After calculating the confidence value for all rules, exclude the rules that have less confidence than the minimum threshold (50%).

Itemset	Support_Count
{A, B, C}	2
~~{B, C, D}~~	~~1~~
~~{A, C, D}~~	~~0~~
~~{A, B, D}~~	~~0~~

- First three rules **A ^B → C, B^C → A, and A^C → B** can be considered as the strong association rules for the given problem.

Rules	Support	Confidence
A ^B → C	2	Sup{(A ^B) ^C}/sup(A ^B)= 2/4=0.5=50%
B^C → A	2	Sup{(B^C) ^A}/sup(B ^C)= 2/4=0.5=50%
A^C → B	2	Sup{(A ^C) ^B}/sup(A ^C)= 2/4=0.5=50%
C→ A ^B	2	Sup{(C^(A ^B)}/sup(C)= 2/5=0.4=40%
A→ B^C	2	Sup{(A^(B ^C)}/sup(A)= 2/6=0.33=33.33%
B→ B^C	2	Sup{(B^(B ^C)}/sup(B)= 2/7=0.28=28%

DATA ANALYSIS - APRIORI

Transaction ID	Items bought
1	(Apple x 3), (Cabbage x 1), (Donut x 2)
2	(Bread x 2), (Cabbage x 3), (Egg x 1)
3	(Apple x 1), (Bread x 1), (Cabbage x 1), (Egg x 2)
4	(Bread x 3), (Egg x 4)
5	(Apple x 2), (Cabbage x 2), (Egg x 1)

Transaction ID	Items bought
1	A A A C D D
2	B B C C C E
3	A B C E E
4	B B B E E E E
5	A A C C E

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

Iteration-1

C1	
Item-Set	**Support**
{A}	4
{B}	4
{C}	5
{D}	2
{E}	5

F1	
Item-Set	**Support**
{A}	4
{B}	4
{C}	5
{E}	5

- **Fix a threshold support level.**
- **Generally, 50% of total number of transaction = 2.5 (3)**
- **Discard item/itemset with frequency < threshold (3)**

DATA ANALYSIS - APRIORI

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

F1	
Item-Set	Support
{A}	4
{B}	4
{C}	5
{E}	5

Discard item/itemset with frequency < threshold (3)

Basic Assumption:

- *All subsets of a frequent itemset must be frequent.*
- ***If an itemset is infrequent, all its supersets will be infrequent.***

C1	
Item-Set	Support
{A}	4
{B}	4
{C}	5
{D}	2
{E}	5

Iteration-2

			Only items in F1				
			C2			F2	
Transaction ID	Items bought		Item-Set	Support		Item-Set	Support
1	A C D	→	{A,B}	2	→	{A,C}	4
2	B C E		{A,C}	4		{A,E}	3
3	A B C E		{A,E}	3		{B,C}	3
4	B E		{B,C}	3		{B,E}	4
5	A C E		{B,E}	4		{C,E}	4
6	A B C D E		{C,E}	4			

DATA ANALYSIS - APRIORI

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

F1	
Item-Set	**Support**
{A}	4
{B}	4
{C}	5
{E}	5

F2	
Item-Set	**Support**
{A,C}	4
{A,E}	3
{B,C}	3
{B,E}	4
{C,E}	4

Discard item/itemset with frequency >= threshold (3)

F3	
Item-Set	**Support**
{A,C,E}	3
{B,C,E}	3

Iteration-3

- Grouping is done in a way that each item-set contains three items in them.
- Further, these will be divided into their **sub-sets.**
- Also, those with support value less than threshold (3), will be omitted → This process is known as **Pruning**.

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

→

C3	
Item-Set	**In F2 ?**
{A,B,C}, {A,B}, {A,C}, {B,C}	No
{A,B,E}, {A,B}, {A,E}, {B,E}	No
{A,C,E}, {A,E}, {A,C}, {C,E}	Yes
{B,C,E}, {B,C}, {B,E}, {C,E}	Yes

DATA ANALYSIS - APRIORI

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

F1	
Item-Set	Support
{A}	4
{B}	4
{C}	5
{E}	5

F2	
Item-Set	Support
{A,C}	4
{A,E}	3
{B,C}	3
{B,E}	4
{C,E}	4

Discard item/itemset with frequency < threshold (3)

F3	
Item-Set	Support
{A,C,E}	3
{B,C,E}	3

Iteration-4

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

→

C4	
Item-Set	Support
{A,B,C,E}	2
{A,B,C,D}	1
{B,C,D,E}	1
{A,C,D,E}	1

→

Omitting items that are already omitted	
C4	
Item-Set	Support
{A,B,C,E}	2

In iteration-4, support of the only item-set having 4 items is less than threshold value (3) → Stop iterations → Take final item set to be F3.

DATA ANALYSIS - APRIORI

From F3,

- If I = {A,C,E}, then subsets are {A,C}, {A,E}, {C,E}, {A}, {C} and {E}.
- If I = {B,C,E}, then subsets are {B,C}, {B,E}, {C,E}, {B}, {C} and {E}.

Association Rules: In order to filter out relevant item-sets, create association rules and apply them to subsets. (assume minimum confidence value = 60%)

- For every subset S of I, association rule is:

$$S \longrightarrow (I - S) \text{ if } \frac{\text{Support (I)}}{\text{Support (S)}} \geq \text{Minimum confidence value i.e., 60\%}$$

'S' recommends 'I - S'

Consider {A,C,E} from F3.
Rule 1: {A,C} →({A,C,E} — {A,C})
which is **{A,C} → {E}**

F3	
Item-Set	**Support**
{A,C,E}	3
{B,C,E}	3

F2	
Item-Set	**Support**
{A,C}	4
{A,E}	3
{B,C}	3
{B,E}	4
{C,E}	4

F1	
Item-Set	**Support**
{A}	4
{B}	4
{C}	5
{E}	5

DATA ANALYSIS - APRIORI

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

F1

Item-Set	Support
{A}	4
{B}	4
{C}	5
{E}	5

F2

Item-Set	Support
{A,C}	4
{A,E}	3
{B,C}	3
{B,E}	4
{C,E}	4

F3

Item-Set	Support
{A,C,E}	3
{B,C,E}	3

Consider {A,C,E} from F3.

Rule 1: {A,C} →({A,C,E} — {A,C}) which is **{A,C} → {E}**
Confidence = Support{A,C,E} / Support{A,C} = 3/4 = 75% > 60%
So rule 1 i.e., {A,C} →{E} is valid.

Rule 2: {A,E} →({A,C,E} — {A,E}) which is **{A,E} → {C}**
Confidence = Support{A,C,E} / Support{A,E} = 3/3 = 100% > 60%
So rule 2 i.e., {A,E} →{C} is valid.

Rule 3: {C,E} →({A,C,E} — {C,E}) which is **{C,E} → {A}**
Confidence = Support{A,C,E} / Support{C,E} = 3/4 = 75% > 60%
So rule 3 i.e., {C,E} →{A} is valid.

Rule 4: {A} →({A,C,E} — {A}) which is **{A} → {C,E}**
Confidence = Support{A,C,E} / Support{A} = 3/4 = 75% > 60%
So rule 4 i.e., {A} →{C,E} is valid.

Rule 5: {C} →({A,C,E} — {C}) which is **{C} → {A,E}**
Confidence = Support{A,C,E} / Support{C} = 3/5 = 60% !> 60% (not greater than 60%)
So rule 5 i.e., {C} →{A,E} is **rejected**.

$$Support = \frac{frq(X,Y)}{N}$$

$$Rule: X \Rightarrow Y \rightarrow Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

$S \rightarrow (I - S)$ if $\frac{\text{Support (I)}}{\text{Support (S)}} \geq$ Minimum confidence value i.e., 60%

'S' recommends 'I - S'

Rule 6: {E} →({A,C,E} — {E}) which is **{E} → {A,C}**
Confidence = Support{A,C,E} / Support{E} = 3/5 = 60% !> 60% (not greater than 60%)
So rule 6 i.e., {E} →{A,C} is **rejected**.

DATA ANALYSIS - APRIORI

FOR {A,C,E} in F3, association rules generated earlier.

Calculate strength of each one.

Rule 1: Lift = Support{A,C,E} / (Support{A,C} x Support{E}) = 3/(4x5) = 3/20 = **0.15**

Rule 2: Lift = Support{A,C,E} / (Support{A,E} x Support{C}) = 3/(3x5) = 3/15 = **0.20**

Rule 3: Lift = Support{A,C,E} / (Support{C,E} x Support{A}) = 3/(4x4) = 3/16 = **0.1875**

Rule 4: Lift = Support{A,C,E} / (Support{A} x Support{C,E}) = 3/(4x4) = 3/16 = **0.1875**

Rule 5: Lift = Support{A,C,E} / (Support{C} x Support{A,E}) = 3/(5x3) = 3/15 = **0.20**

Rule 6: Lift = Support{A,C,E} / (Support{E} x Support{A,C}) = 3/(5x4) = 3/20 = **0.15**

- Same steps can be applied to item-set {B,C,E}.
- Very small sample → Lift values do not vary much here .
- All Lift < 1 → None of association rule is strong to be accepted.
- For **larger data**; analysis is more evident.

F1	
Item-Set	Support
{A}	4
{B}	4
{C}	5
{E}	5

F2	
Item-Set	Support
{A,C}	4
{A,E}	3
{B,C}	3
{B,E}	4
{C,E}	4

F3	
Item-Set	Support
{A,C,E}	3
{B,C,E}	3

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Transaction ID	Items bought
1	A C D
2	B C E
3	A B C E
4	B E
5	A C E
6	A B C D E

DATA ANALYSIS – APRIORI

Example: For the purchase transaction given below, establish the association rules with Support threshold=50%, Confidence= 60%.

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

1-2	4
1-3	3
2-3	4
1-2-3	3

DATA ANALYSIS – F-P GROWTH

Shortcomings Of Apriori Algorithm

- Needs generation of large number of candidate itemsets (for huge database).
- Needs multiple scans of database to check support of each itemset generated and this leads to high costs.

Frequent Pattern (F-P) growth algorithm

- Improved version of Apriori Algorithm (overcome these shortcomings)
- No need for candidate generation to generate frequent pattern.
- Represents database in form of tree structure (F-P tree) to extract most frequent patterns.
- F-P tree structure maintains the association (frequency patterns) between itemsets.
- Database is fragmented using one frequent item. This fragmented part is called “pattern fragment”.
- Itemsets of these fragmented patterns are analyzed → Thus search for frequent itemsets is reduced comparatively.

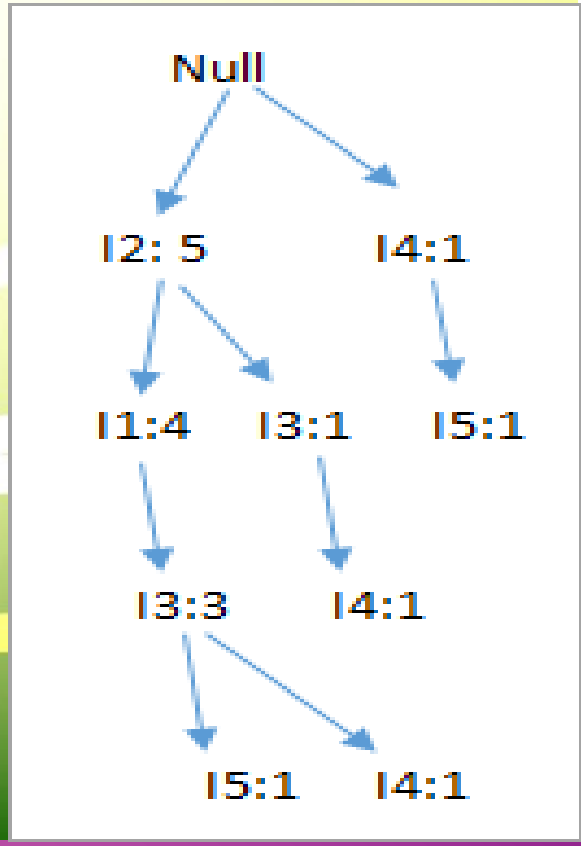
DATA ANALYSIS – F-P GROWTH

Frequent Pattern Tree

- Tree-like structure that is made with initial itemsets of database.
- Every node of FP tree represents an item of that itemset.
- Root node represents null value whereas lower nodes represent itemsets of the data.
- Association of these nodes with lower nodes that is between itemsets is maintained while creating the tree.

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Item	Count
I1	4
I2	5
I3	4
I4	3
I5	2



DATA ANALYSIS – F-P GROWTH

F-P Algorithm Steps

1. Scan database to find occurrences of itemsets.
2. Construct FP tree by creating root (null) of the tree.
3. Scan database again. Tree Branch is constructed with transaction itemsets in descending order of count.
   - Examine first transaction and find out itemset in it. Itemset with max count is taken at top, the next itemset with lower count and so on.
4. Next transaction in database is examined. Itemsets are ordered in descending order of count.
   - If any itemset of this transaction is already present in another branch, then this transaction branch would share a common prefix to root.
   - i.e. common itemset is linked to new node of another itemset in this transaction.
5. Count of itemset is incremented as it occurs in transactions. Both common node and node count is increased by 1 as they are created and linked according to transactions.
6. Once all the transactions are scanned iteratively → **FP tree is created.**

7. **Mine the created FP Tree.** i.e. lowest node is examined first along with the links of lowest nodes.
   - Lowest node represents frequency pattern length 1. From this, traverse the path in FP Tree. This path(s) are called conditional pattern base (a sub-database consisting of prefix paths in FP tree occurring with lowest node/suffix).
8. Construct a Conditional FP Tree, which is formed by a count of itemsets in the path (itemsets meeting threshold support are considered in Conditional FP Tree).
9. Frequent Patterns are generated from the Conditional FP Tree.

DATA ANALYSIS – F-P GROWTH

Support threshold=50% → min_sup=3, Confidence= 60%

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Item	Count
I1	4
I2	5
I3	4
I4	3
I5	**2**

Freq count

Item	Count
I2	5
I1	4
I3	4
I4	3

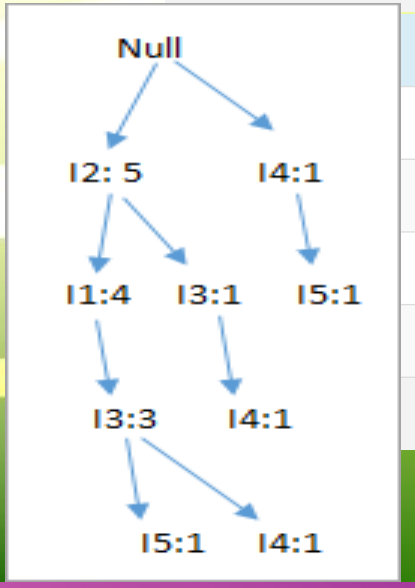
Threshold filter & Sort

DATA ANALYSIS – F-P GROWTH

Build FP Tree

1. Considering the **root node** null.
2. First scan of Transaction **T1: I1, I2, I3** contains three items {I1:1}, {I2:1}, {I3:1}, where I2 is linked as a child to root, I1 is linked to I2 and I3 is linked to I1.
3. **T2: I2, I3, I4** contains I2, I3, and I4, where I2 is linked to root, I3 is linked to I2 and I4 is linked to I3. But this branch would share I2 node as common as it is already used in T1.
4. Increment the count of I2 by 1 and I3 is linked as a child to I2, I4 is linked as a child to I3. The count is {I2:2}, {I3:1}, {I4:1}.
5. **T3: I4, I5**. Similarly, a new branch with I5 is linked to I4 as a child is created.
6. **T4: I1, I2**. The sequence will be I2, and I1. I2 is already linked to the root node, hence it will be incremented by 1. Similarly I1 will be incremented by 1 as it is already linked with I2 in T1, thus {I2:3}, {I1:2}.
7. **T5:I1, I2, I3, I5**. The sequence will be I2, I1, I3, and I5. Thus {I2:4}, {I1:3}, {I3:2}, {I5:1}.
8. **T6: I1, I2, I3, I4**. The sequence will be I2, I1, I3, and I4. Thus {I2:5}, {I1:4}, {I3:3}, {I4 1}.

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4



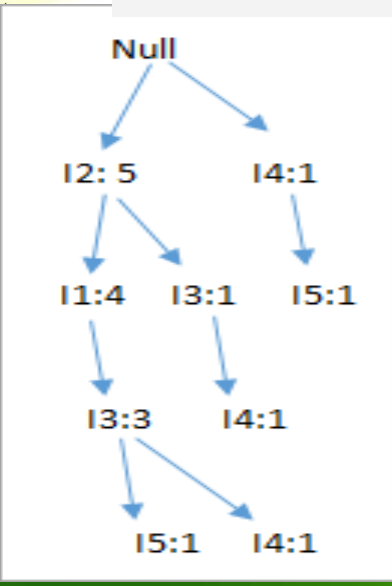
Item	Count
I2	5
I1	4
I3	4
I4	3
I5	2

DATA ANALYSIS – F-P GROWTH

Mining of FP-tree:

- Lowest node item I5 is deleted (threshold).
- Next lower node I4 occurs in 2 branches, {I2,I1,I3:1},{I2,I3:1}. This forms the conditional pattern base.
- Conditional pattern base is considered a transaction database & conditional FP-tree is constructed. This will contain {I2:2, I3:2, I1:1}. I1 is not considered as it does not meet the min support count.
- This path will generate all combinations of frequent patterns : {I2,I4:2},{I3,I4:2},{I2,I3,I4:2}
- For I3, prefix path is: {I2,I1:3},{I2:1}, this will generate a 2 node FP-tree : {I2:4, I1:3} and frequent patterns are generated: {I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}.
- For I1, prefix path is: {I2:4} this generates single node FP-tree: {I2:4} and frequent patterns are generated: {I2, I1:4}.

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4



Item	Count
I2	5
I1	4
I3	4
I4	4

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I4	{I2,I1,I3:1},{I2,I3:1}	~~{I2:2, I3:2}~~	~~{I2,I4:2},{I3,I4:2},{I2,I3,I4:2}~~
I3	{I2,I1:3},{I2:1}	{I2:4, I1:3}	{I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}
I1	{I2:4}	{I2:4}	{I2,I1:4}

DATA ANALYSIS – F-P GROWTH

- Considering last column (frequent Pattern generation) & Support threshold=50% → min_sup=3
  - *3-item frequent set generated {I2,I1,I3:3}*
  - *2-Item frequent sets generated {I2,I3:4}, {I1,I3:3}, {I2,I1:4}.*
- All these set (association rules) are distinct sets.
- Once association rules are generated; lift value can be calculated to further analysis *(same like apriori).*
- In comparison to Apriori Algorithm, only the frequent patterns are generated, NOT all combinations of different items & keep analyzing each (computationally costly).

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I4	{I2,I1,I3:1},{I2,I3:1}	{I2:2, I3:2}	{I2,I4:2},{I3,I4:2},{I2,I3,I4:2}
I3	{I2,I1:3},{I2:1}	{I2:4, I1:3}	{I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}
I1	{I2:4}	{I2:4}	{I2,I1:4}

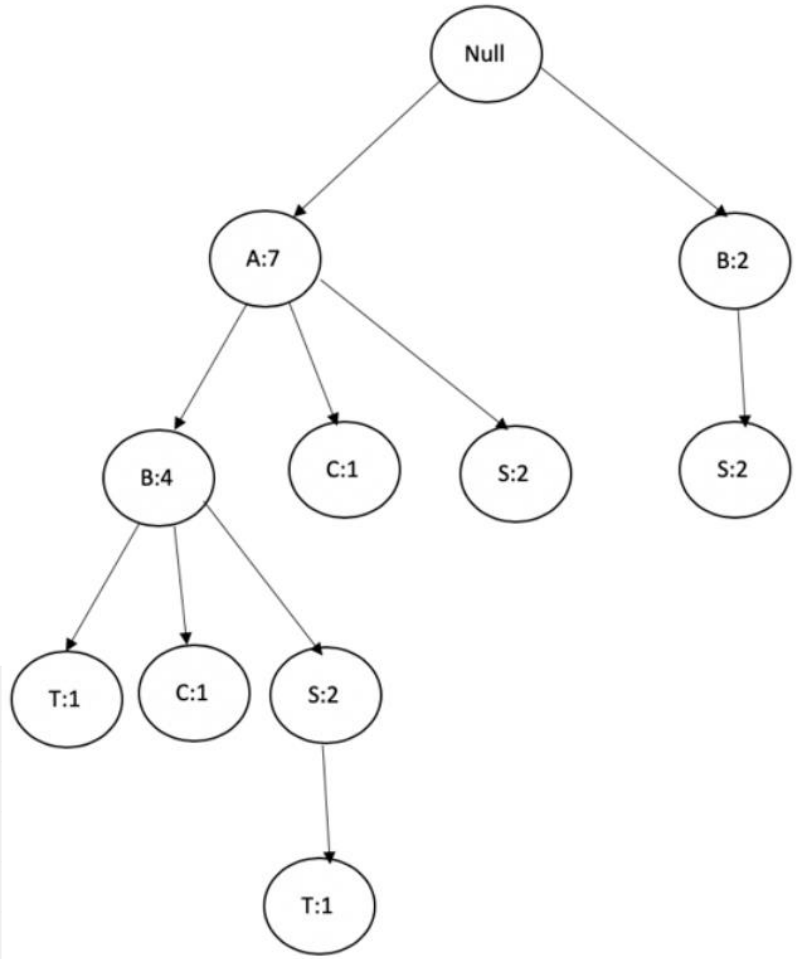
Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

DATA ANALYSIS – F-P GROWTH

- Asparagus (A), Corn (C), Beans (B), Tomatoes (T) & Squash (S)
- minimum count=2

Transaction ID	List of items in transaction
T1	B , A , T
T2	A , C
T3	A , S
T4	B , A , C
T5	B , S
T6	A , S
T7	B , S
T8	B , A , S , T
T9	B , A , S

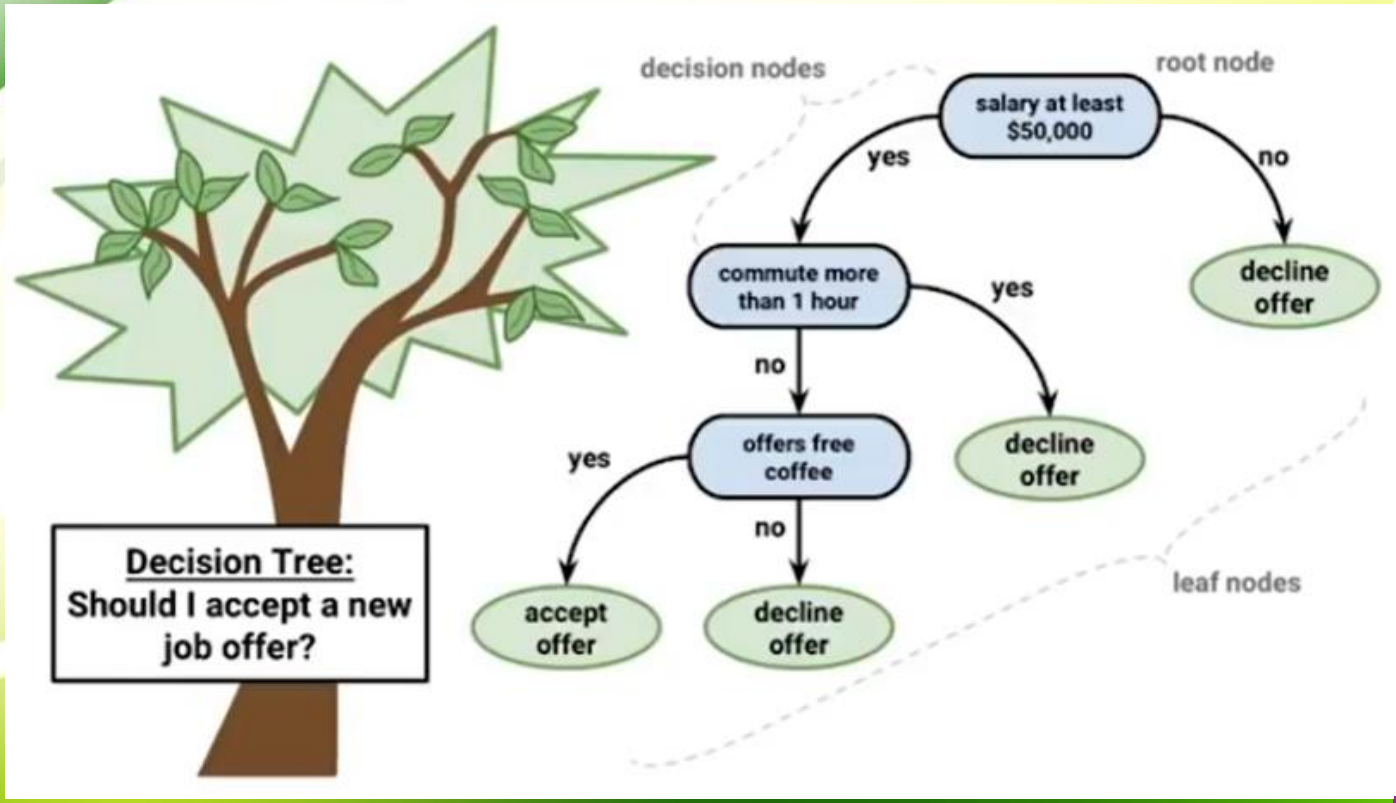
Item	Support Count
Asparagus (A)	7
Beans (B)	6
Squash (S)	6
Corn (C)	2
Tomatoes (T)	2



Item	Conditional Pattern base	Conditional FP tree	Frequent Pattern Generation
Tomatoes (T)	{{A,B:1},{A,B,S:1}}	<A:2,B:2>	{A,T:2},{B,T:2},{A,B,T:2}
Corn (C)	{{A,B:1},{A:1}}	<A:2>	{A,C:2}
Squash (S)	{{A,B:2},{A:2},{B:2}}	<A:4,B:2>,<B:2>	{A,S:4},{B,S:4},{A,B,S:2}
Bean (B)	{{A:4}}	<A:4>	{A,B:4}

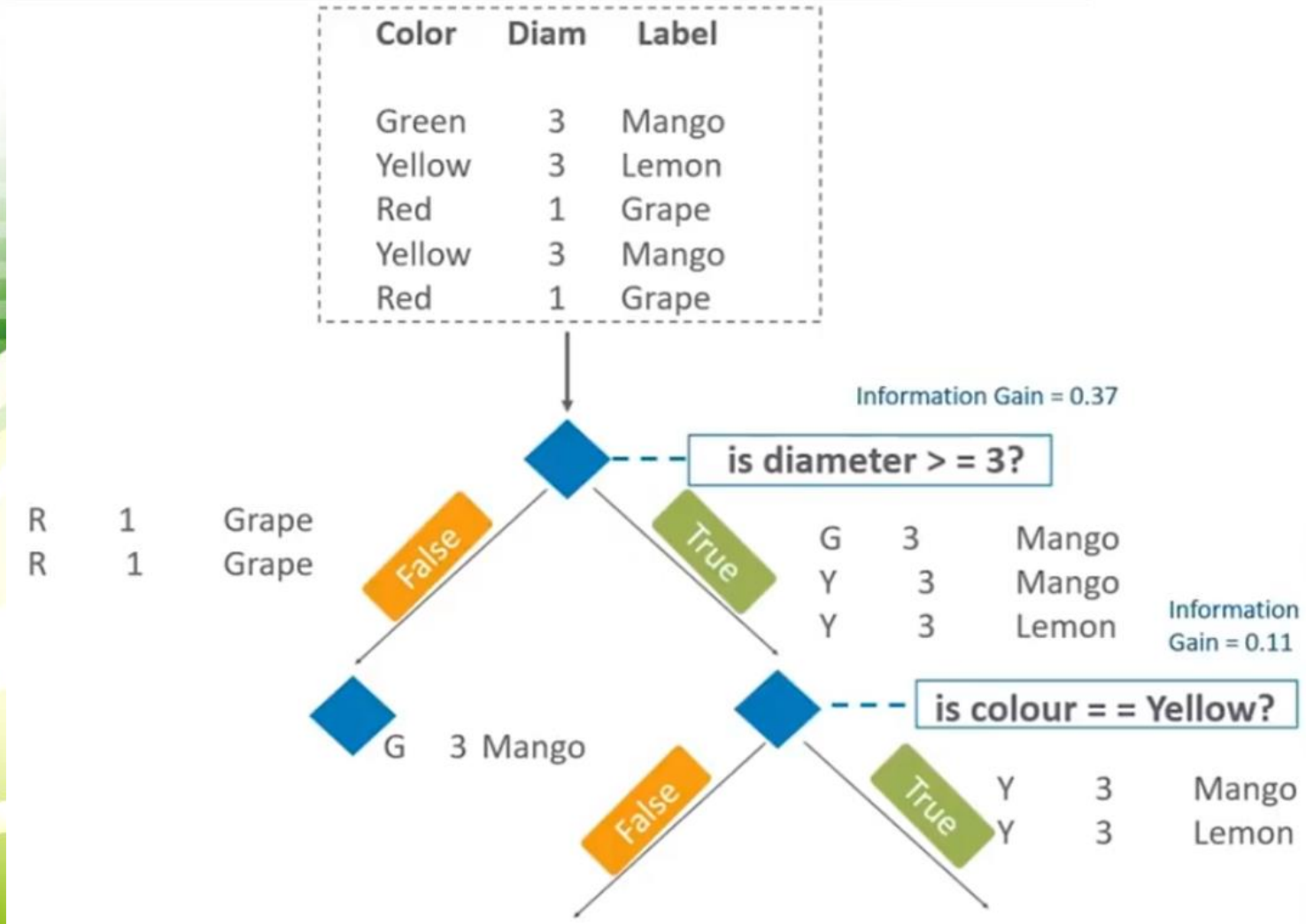
DATA ANALYSIS – DECISION TREE

- Series of questioning is common to many decision-making processes.
- Can be shown visually as decision tree.
- Often generated by hand to precisely and consistently define a decision-making process.
- Can also be generated automatically from data.
- Consist of series of decision points based on certain selected variables.



DATA ANALYSIS – DECISION TREE

- Supervised learning algorithm.
- Supervised methods attempt to place (classify) each observation into interesting groups (based on selected variable).
- These methods iterate over training set of observations and adjust parameters as the classifier correctly or incorrectly classifies each observation.



Decision Tree - Types

- Graphical representation of all possible solutions to a decision.
- Easy to understand how decisions reached based on multiple criteria
- Works for both continuous as well as categorical output variables.

Regression Tree

- Used when the dependent variable is continuous.
- Value obtained by leaf nodes in training data is the mean response of observation falling in that region.
- If an unseen data observation falls in that region, its prediction is made with the mean value.

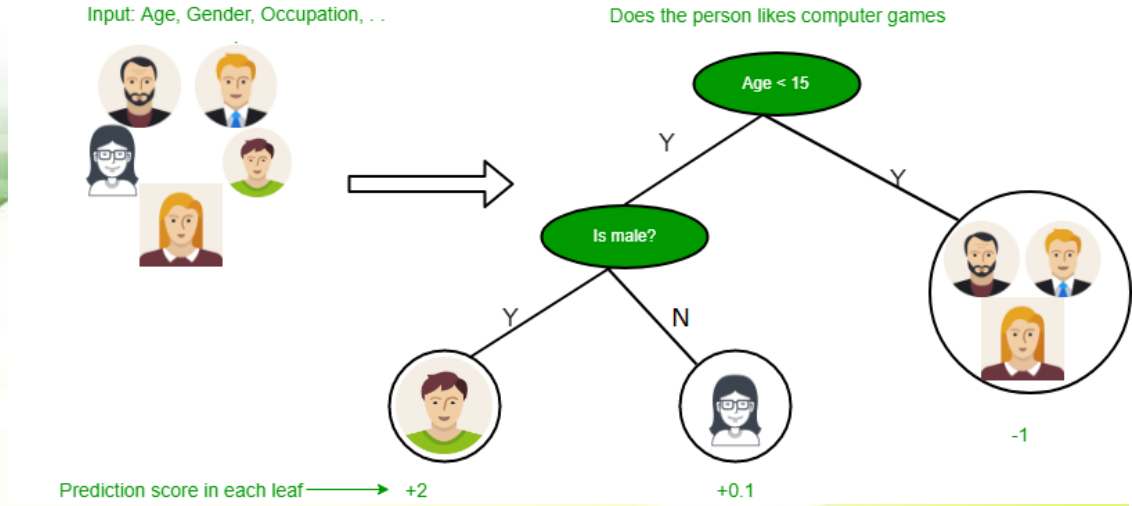
Classification Tree

- Used when dependent variable is categorical.
- Value obtained by leaf nodes in training data is the mode response of observation falling in that region.

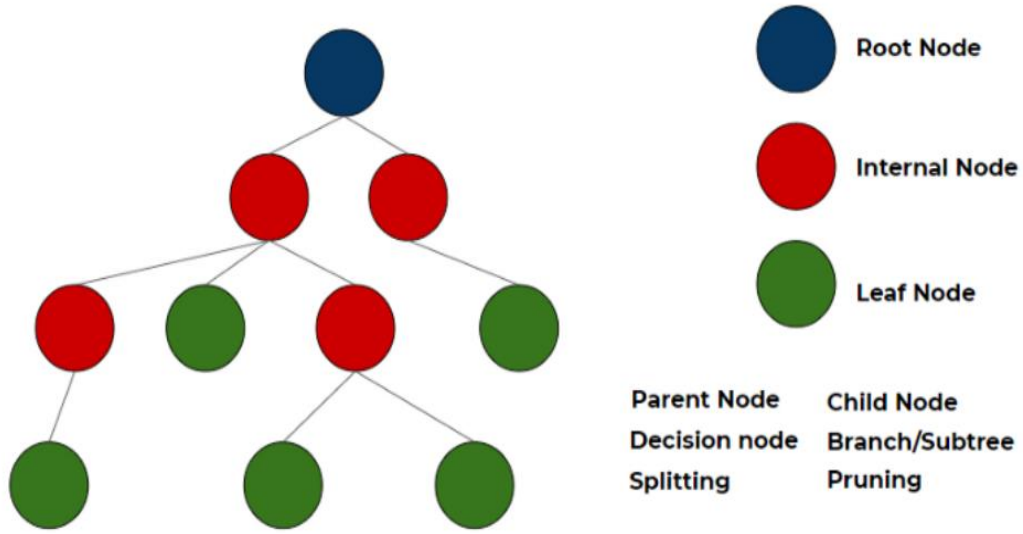
- Together they are called as CART(classification and regression tree)

Decision Tree

- Building decision trees is computationally expensive for large data set with many continuous variables
- Decision tree uses tree representation to solve problem.
- Leaf node corresponds to a class label and attributes are represented on internal node of the tree.
- *parent–child relationship*: relationship between two connected nodes
- *splitting variable:* used as potential decision points
- *response* variable: used to guide construction of tree.
- Response variable used to guide which splitting variables are selected and at what value the split is made.
- DT splits data set into increasingly smaller, nonoverlapping subsets.
- *Root node* contains all observations.
- Splitting Condition; Termination/stop condition.



Terminologies associated with decision tree



Decision Tree

Advantages of a decision tree

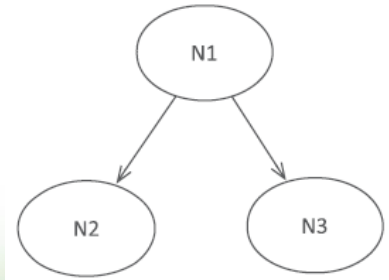
- **Easy to visualize and interpret**: Its graphical representation is very intuitive to understand and it does not require any knowledge of statistics to interpret it.
- **Useful in data exploration**: We can easily identify the most significant variable and the relation between variables with a decision tree. It can help us create new variables or put some features in one bucket.
- **Less data cleaning required**: It is fairly immune to outliers and missing data, hence less data cleaning is needed.
- **The data type is not a constraint**: It can handle both categorical and numerical data.

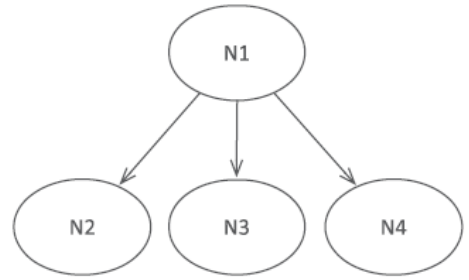
Disadvantages of decision tree

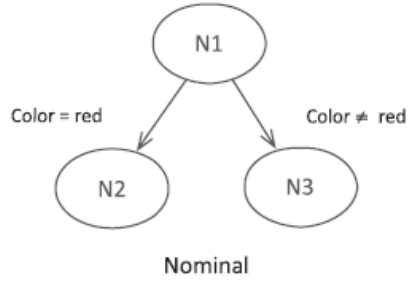
- **Overfitting**: single decision tree tends to overfit the data which is solved by setting constraints on model parameters and pruning.
- **Not exact fit for continuous data**: It losses some of the information associated with numerical variables when it classifies them into different categories.

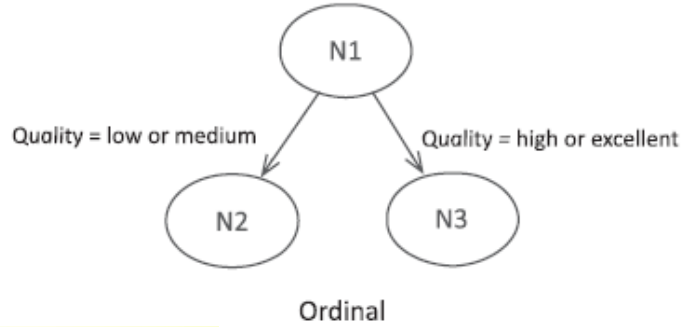
Decision Tree

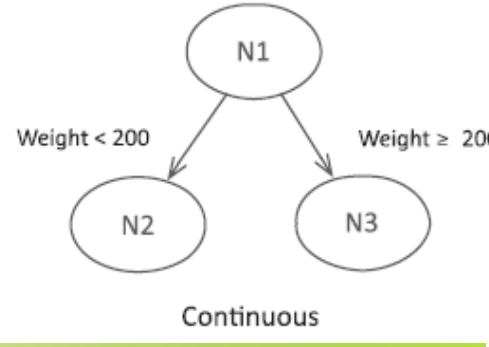
- **Dichotomous**: Two-way split at each level is common.
  - *Temperature* may have only two values: "hot" and "cold."
  - More than two ways possible; can be complex.
- **Nominal**: discrete values with no order
  - *Color* can take values "red," "green," "blue," and "black".
  - two-way split: "red" subset and non-red subset.
- **Ordinal**: discrete values are ordered.
  - *Quality* with possible values "low," "medium," "high," and "excellent".
  - *Quality* (low / medium) one subset; (high / excellent) in another.
- **Continuous**: For variables with continuous values to be split two-ways, specific cut-off value needs to be determined
  - *Weight* between 0 and 1,000 with a selected cut-off of 200.
  - Left subset with *Weight* below 200 and right subset >= 200.





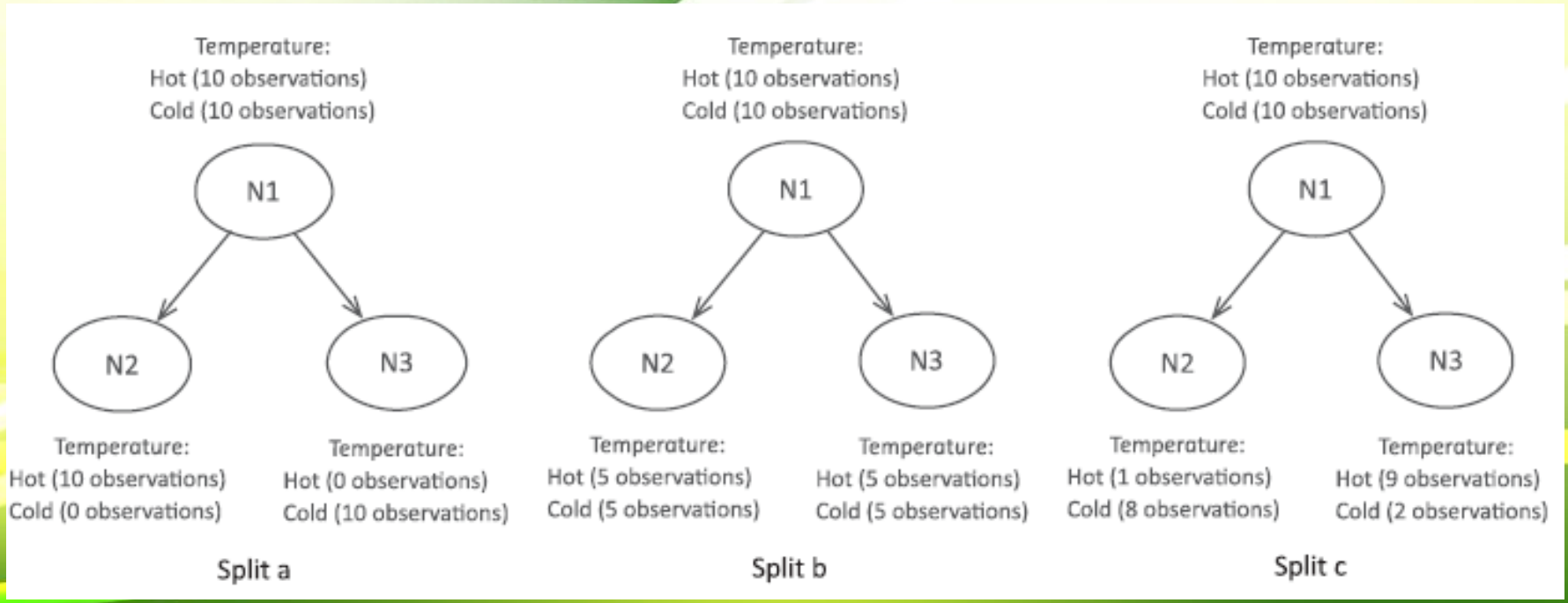






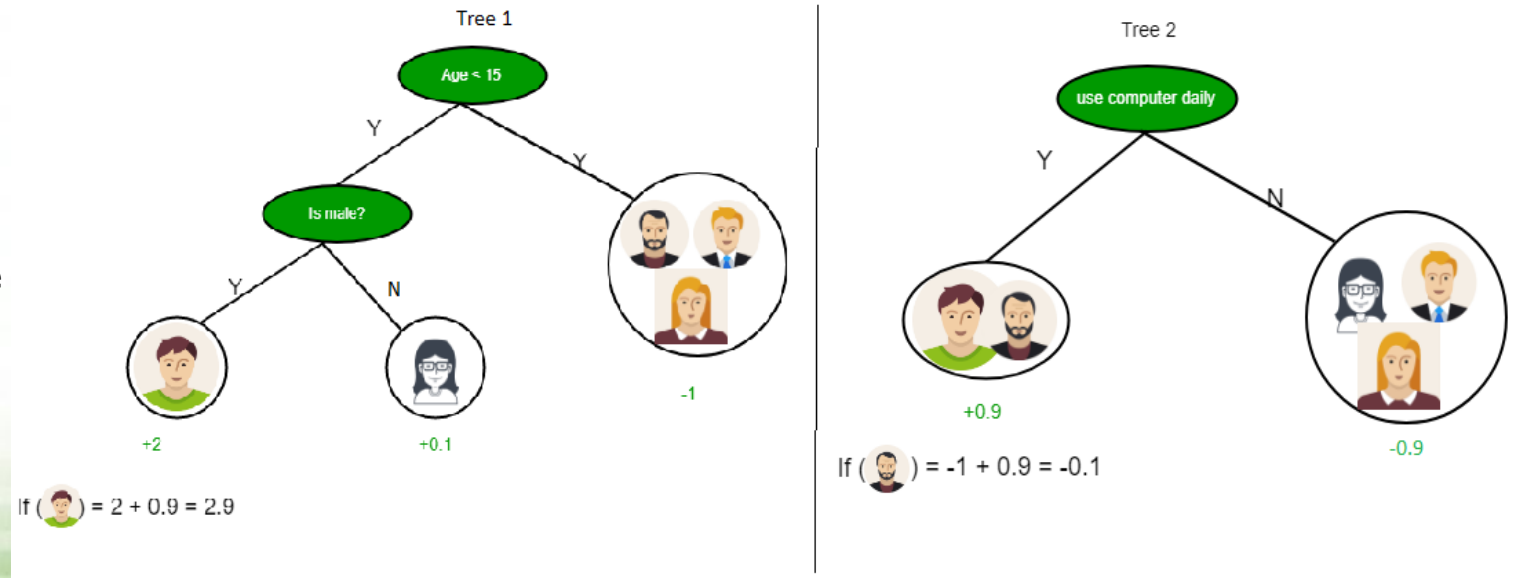
Decision Tree

- Data can be split in n-number of ways.
- To determine best split (condition), a ranking is made of all possible splits using scores calculated for each split.
  - Split a best split; each node is all of same category.
  - Split b (50% "hot," 50% "cold") not a good split.
  - Split c is good split (though not as clean as A); Good proportion of similarity with minimal impurity



Decision Tree

- Major challenge is to identification of the attribute for root node in each level (attribute selection).
    - Popular approaches of selection measures:
        - Information Gain
        - Gini Index



Information Gain

- 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.
- Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.
- The higher the entropy more the information content.

Gini Index

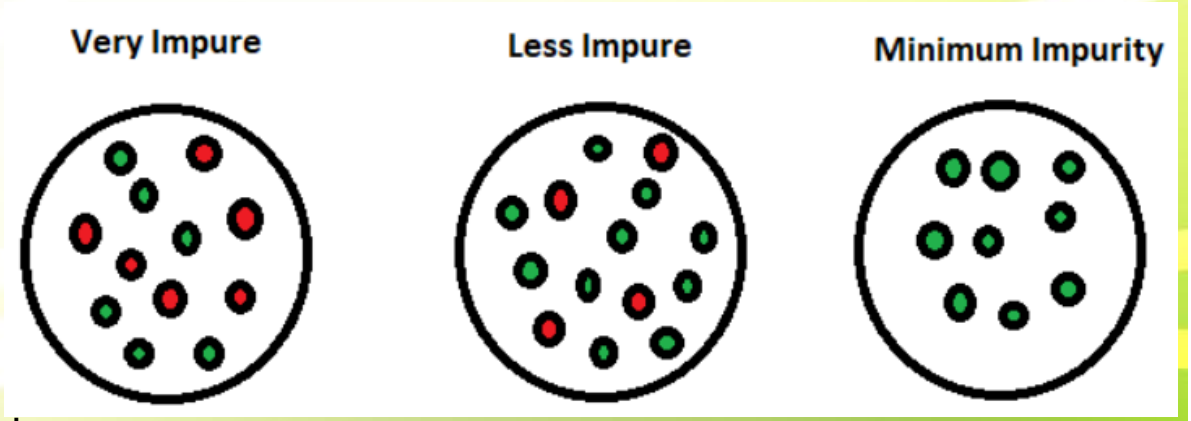
- Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified.
- It means an attribute with lower Gini index should be preferred.
- Most of the time Gini impurity is used as it gives good results for splitting and its computation is inexpensive.

Decision Tree

- "Goodness" of splitting criteria is determined by how clean each split is.
- **Impurity**: proportion of different categories of response variable.
- Cleaner splits result in lower scores.
- As the tree is being generated, it is desirable to decrease level of impurity until ideally there is only one category at a terminal node (a node with no children).
- Three primary methods for calculating impurity:
  - Misclassification,
  - Gini,
  - Entropy.

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

  - S: set of observations.
  - Pi : fraction of observations that belong to a particular value
  - C : number of different possible values of response variable.



Decision Tree

Split a

Entropy (N1) = $-(10/20)\log_2(10/20) - (10/20)\log_2(10/20) = 1$
Entropy (N2) = $-(10/10)\log_2(10/10) - (0/10)\log_2(0/10) = 0$
Entropy (N3) = $-(0/10)\log_2(0/10) - (10/10)\log_2(10/10) = 0$

Split b

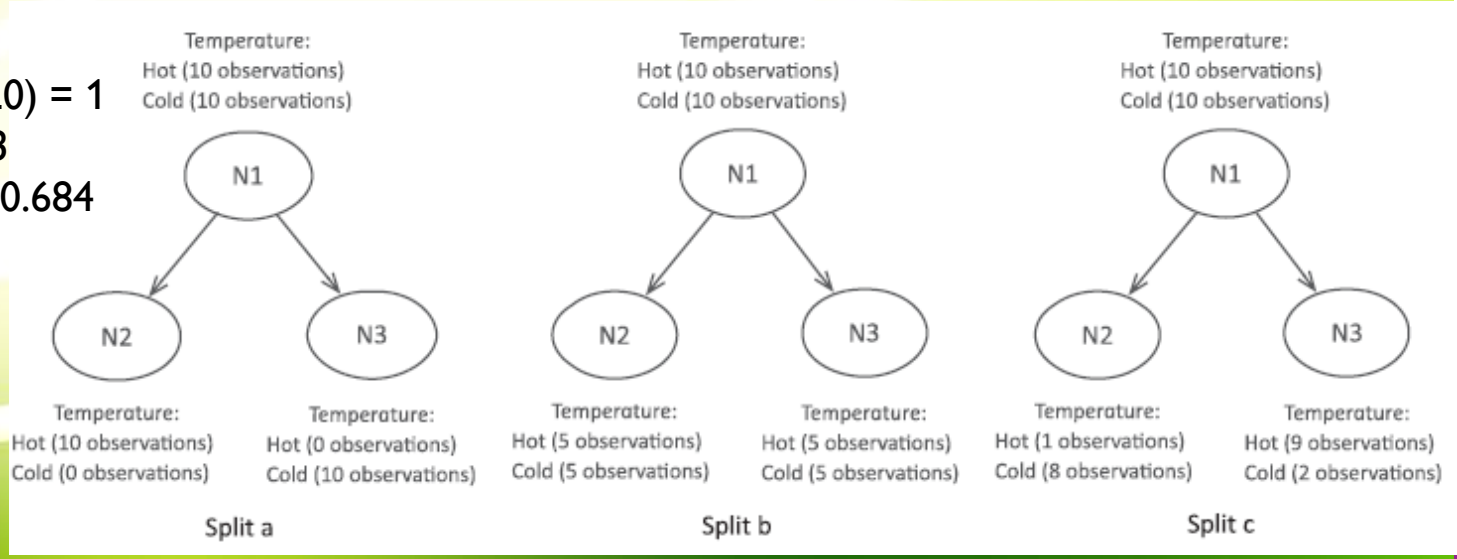
Entropy (N1) = $-(10/20)\log_2(10/20) - (10/20)\log_2(10/20) = 1$
Entropy (N2) = $-(5/10)\log_2(5/10) - (5/10)\log_2(5/10) = 1$
Entropy (N3) = $-(5/10)\log_2(5/10) - (5/10)\log_2(5/10) = 1$

Split c

Entropy (N1) = $-(10/20)\log_2(10/20) - (10/20)\log_2(10/20) = 1$
Entropy (N2) = $-(1/9)\log_2(1/9) - (8/9)\log_2(8/9) = 0.503$
Entropy (N3) = $-(9/11)\log_2(9/11) - (2/11)\log_2(2/11) = 0.684$

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

S: set of observations.
Pi : fraction of observations that belong to a particular value
C : number of different possible values of response variable



Decision Tree

- *N:* number of observations in parent node,
- *K:* number of possible resulting nodes,
- *N(vj)* : number of observations for each of the *j* child nodes,
- *vj* : set of observations for the *j*th node.

$$Gain = E_{parent} - E_{children}$$

$$\text{Gain} = \text{Entropy(parent)} - \sum_{j=1}^{k} \frac{N(v_j)}{N} \text{Entropy}(v_j)$$

- Gain formula can be used with other impurity metrics by replacing the entropy calculation.

  *Gain(Split a) = 1 − (((10/20) *0) + ((10/20) *0)) = 1*

  *Gain(Split b) = 1 − (((10/20) *1) + ((10/20) *1)) = 0*

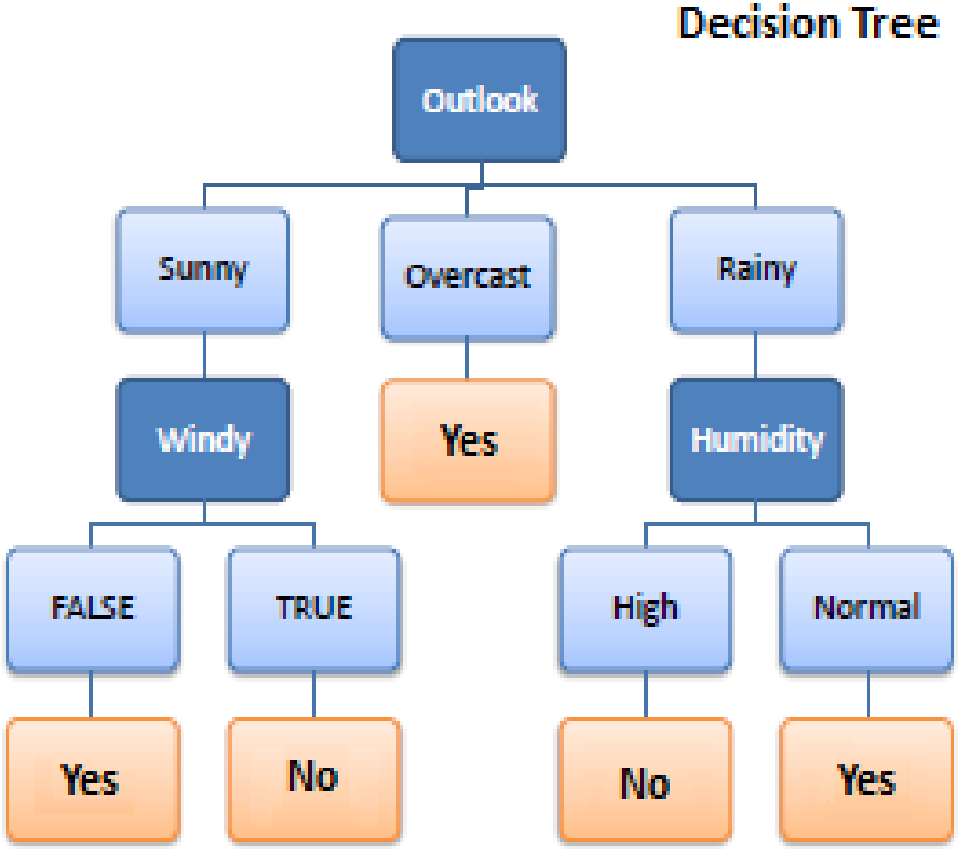
  *Gain(Split c) = 1 − (((9/20) *0.503) + ((11/20)* 0.684)) = 0.397*

- Condition used in **Split a** is selected as best splitting criteria.
- During tree generation process, algorithm examines all possible splitting values for all splitting variables, calculates a gain function, and selects best splitting criterion.

Decision Tree

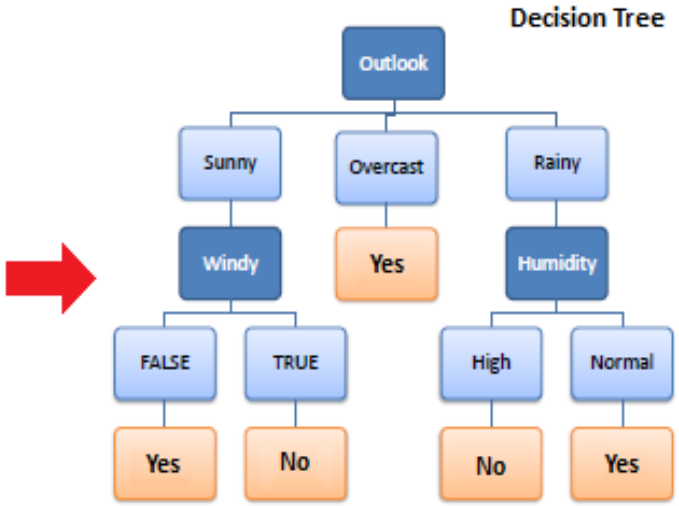
Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Decision Tree

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

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

Decision Tree

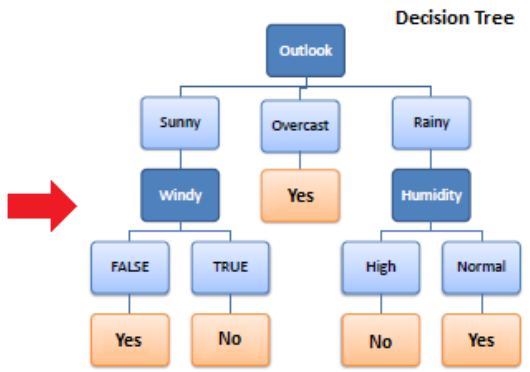
$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

$$Gain = E_{parent} - E_{children}$$

$$\text{Gain} = \text{Entropy(parent)} - \sum_{j=1}^{k} \frac{N(v_j)}{N} \text{Entropy}(v_j)$$

Predictors | Target

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

E(PlayGolf, Outlook) = **P**(Sunny)*__E__(3,2) + **P**(Overcast)*__E__(4,0) + **P**(Rainy)*__E__(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Decision Tree

Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Decision Tree

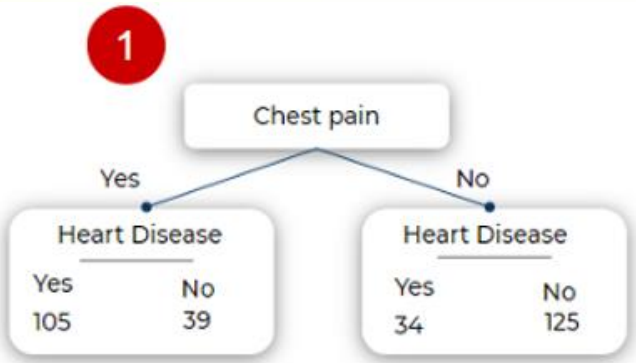
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

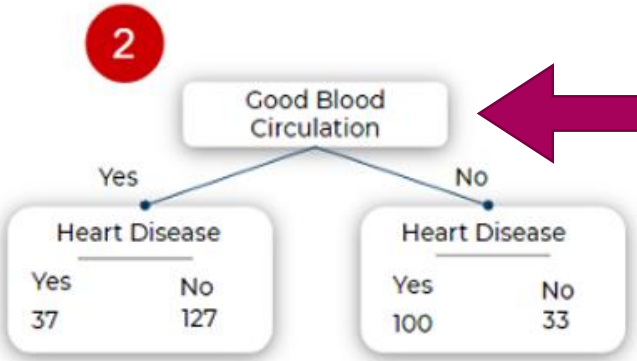
		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

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

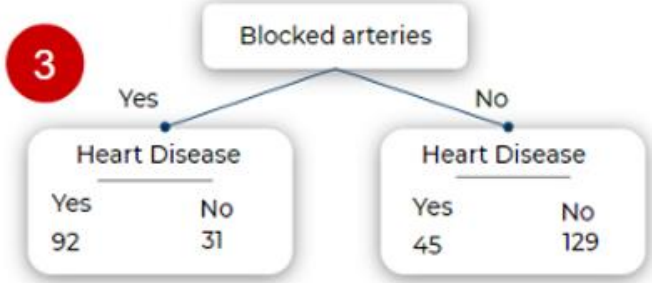
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Decision Tree



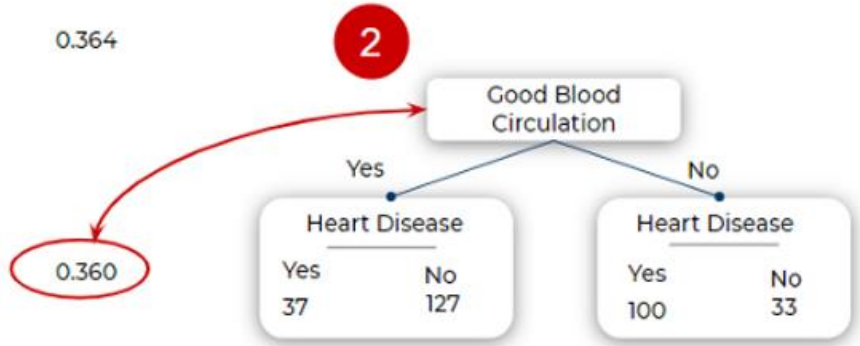


Chest pain	Good blood circulation	Blocked arteries	Heart disease
yes	no	yes	no
yes	yes	yes	yes
no	no	yes	no
yes	yes	no	yes
...	...	...	...



Gini Impurity

1	Chest pain	0.364
2	Good Blood Circulation	0.360
3	Blocked arteries	0.381



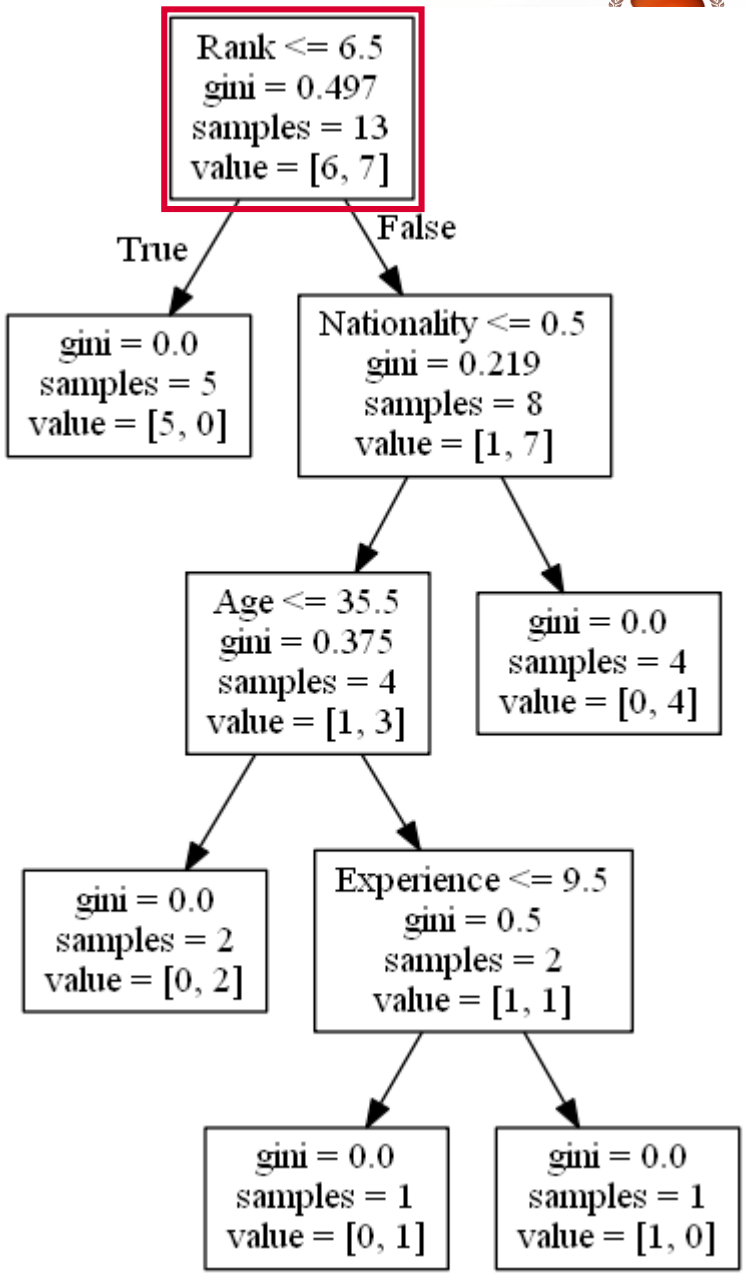
Decision Tree

- **Rank** <= 6.5 means that every candidate with a rank of 6.5 or lower will follow the True arrow (to the left), and the rest will follow the False arrow (to the right).

- **gini** = 0.497 refers to the quality of the split,
  - always a number between 0.0 and 0.5,
  - where 0.0 would mean all of the samples got the same result,
  - 0.5 would mean that the split is done exactly in the middle.

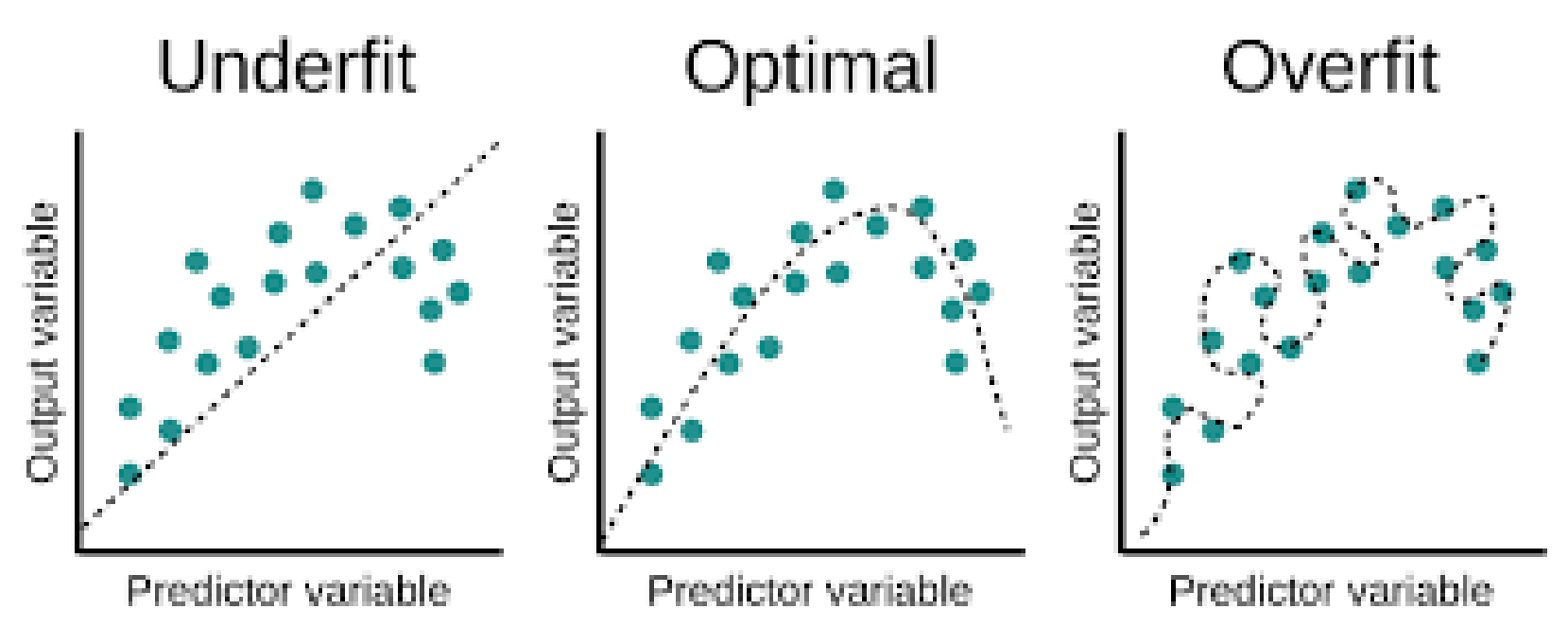
- **samples** = 13 means that there are 13 candidate left at this point in the decision, which is all of them since this is the first step.

- **value** = [6, 7] means that of these 13 candidate, 6 will get "NO", and 7 will get "GO".



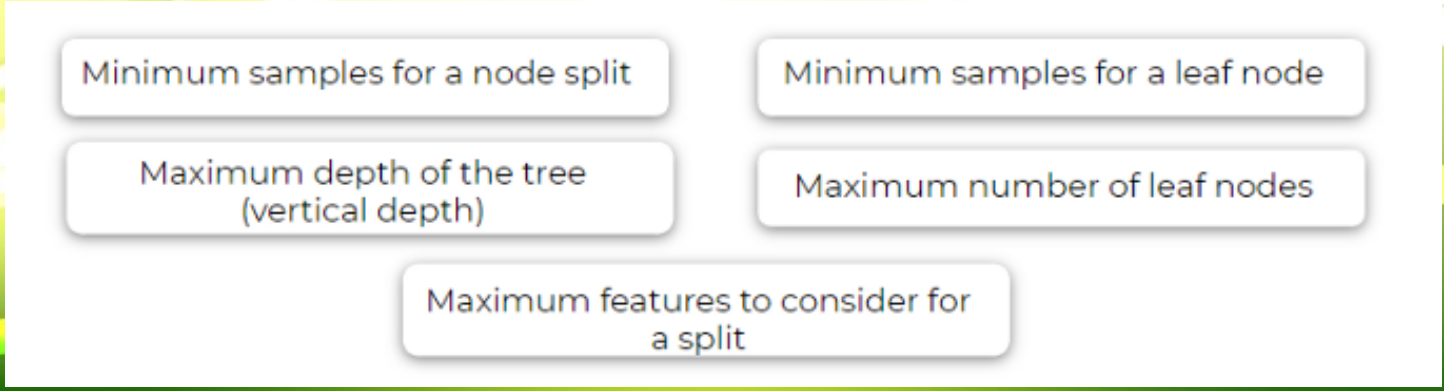
Decision Tree

- Overfitting is one of the key challenges in a tree-based algorithm.
- If no limit is set, it will give 100% fitting.
  - In worst-case scenario, it will end up making a leaf node for each observation.



Decision Tree

- We need to take some precautions to avoid overfitting.
- It is mostly done in two ways:
  - **Setting constraints on tree size:**
    - Parameters play important role in tree modeling.
    - Various parameters can help avoid overfitting.
  - **Tree pruning:**
    - Pruning is something opposite to splitting.
    - To overcome the overfitting issue of tree, we decide to merge two segments in the middle which means removing nodes from the tree.



Decision Tree

DT can be used for grouping &

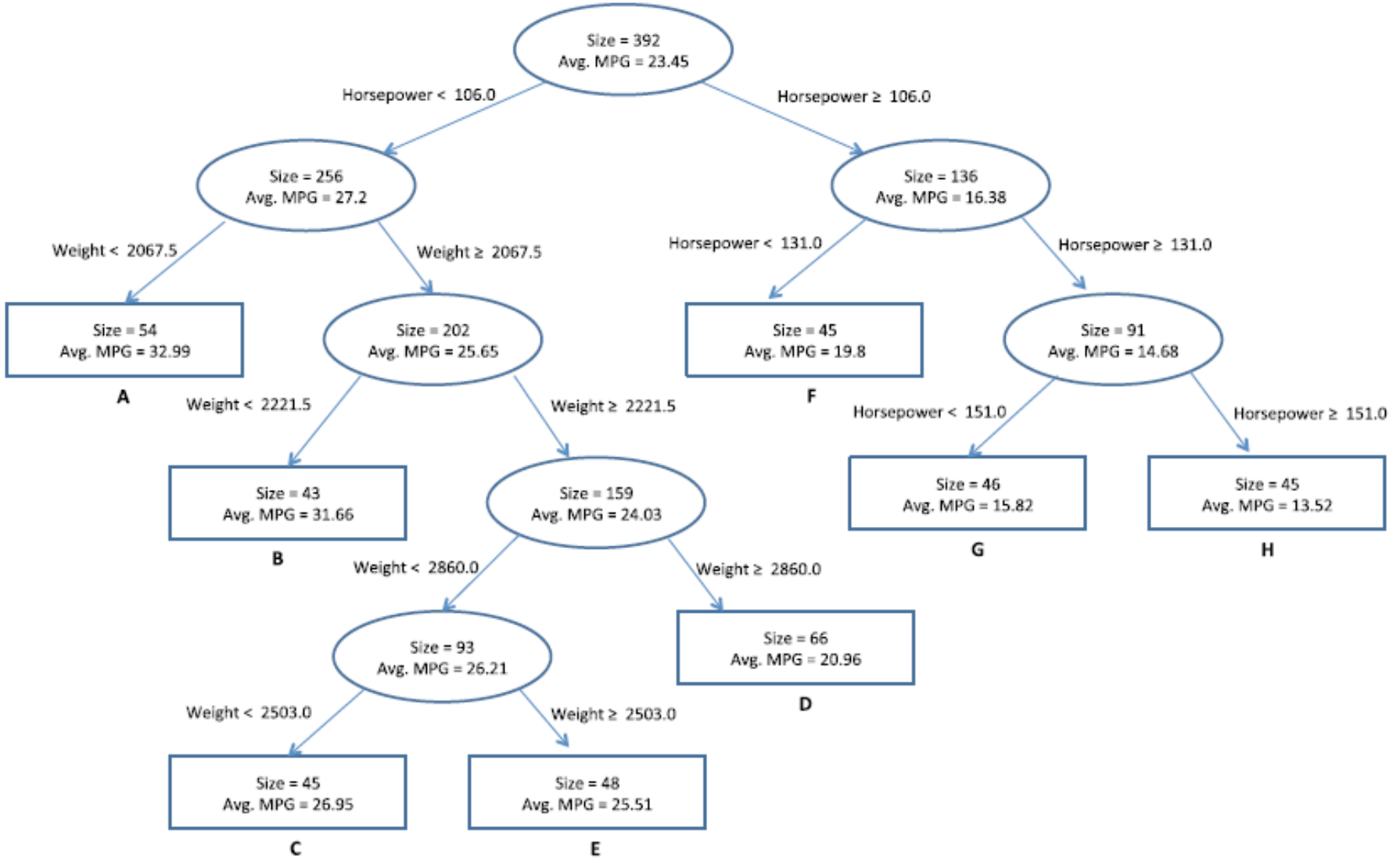
Generating association rule.

Node A

IF *Horsepower* <106 AND
Weight <2067.5
THEN *MPG* is high

Node B

IF *Horsepower* <106 AND
Weight 2067.5 – 2221.5
THEN *MPG* is high



Decision Tree

- Sum of squares of error (SSE): When response variable is continuous.
- resulting split should ideally result in sets where response values are close to group mean.
- The lower group's SSE value is, the closer that group's values are to the mean of the set.
- For each potential split, a SSE value is calculated for each resulting node.
- A score for the split is calculated by summing the SSE values of each resulting node.
- Once all splits for all variables are computed, then the split with the lowest score is selected.

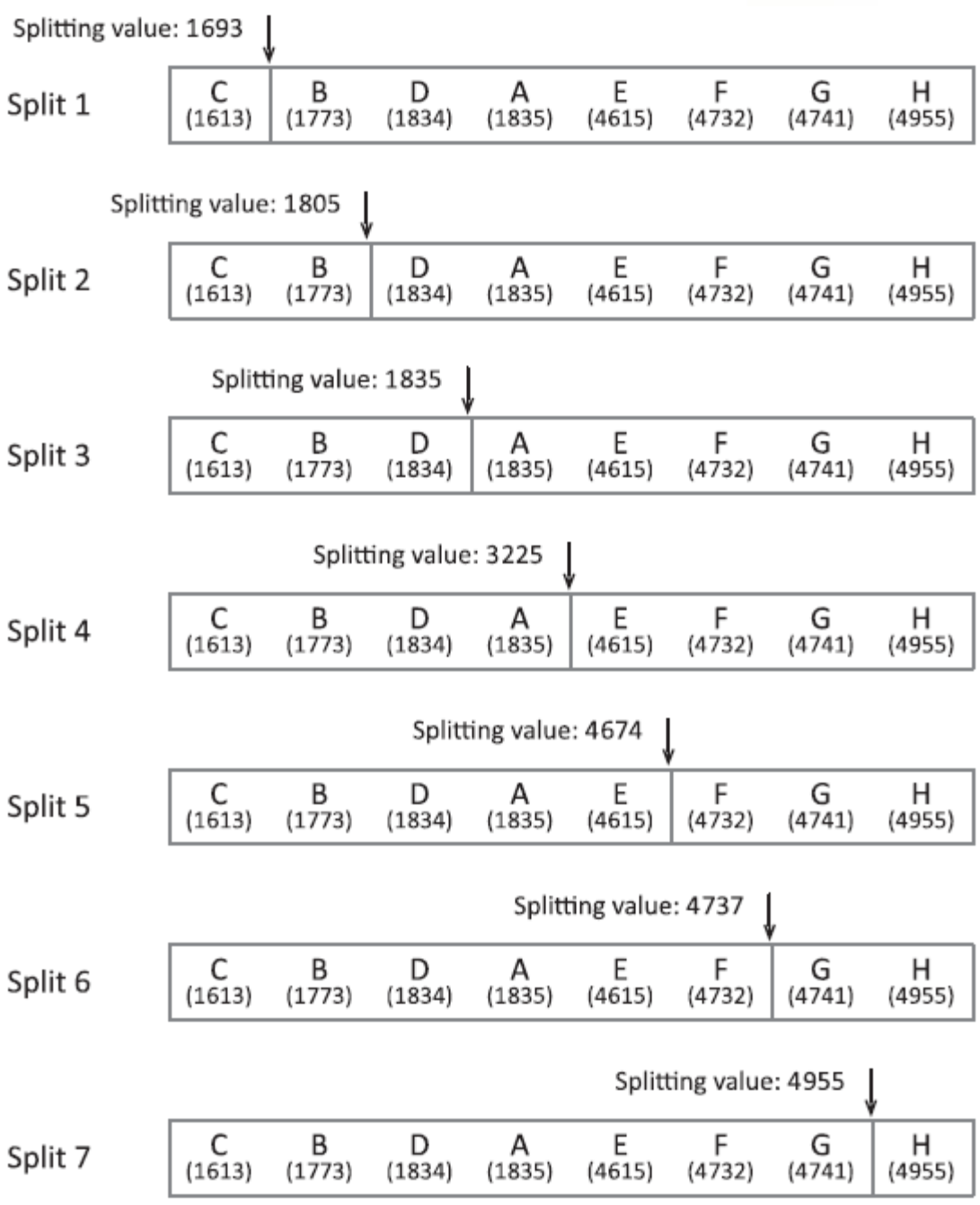
$$SSE = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

N: number of observations
y_i : individual value for the response
$\bar{y}$: average value for the subset.

Decision Tree

Observations	Weight	MPG
A	1,835	26
B	1,773	31
C	1,613	35
D	1,834	27
E	4,615	10
F	4,732	9
G	4,955	12
H	4,741	13

- *Weight* is assigned as a splitting variable and *MPG as response variable.*
- Series of values used to split *Weight*: 1,693, 1,805, 1,835, 3,225, 4,674, 4,737, and 4,955.
- These values are midpoint between each pair of values (after sorting) and were selected because they divided data set into all possible two-ways splits.
- Calculate only a score for splits which result in three or more observations → Split 3, Split 4, and Split 5.



Decision Tree

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

N: number of observations
y_i: individual value for the response
$\bar{y}$: average value for the subset.

Observations	Weight	MPG
A	1,835	26
B	1,773	31
C	1,613	35
D	1,834	27
E	4,615	10
F	4,732	9
G	4,955	12
H	4,741	13

Split 3

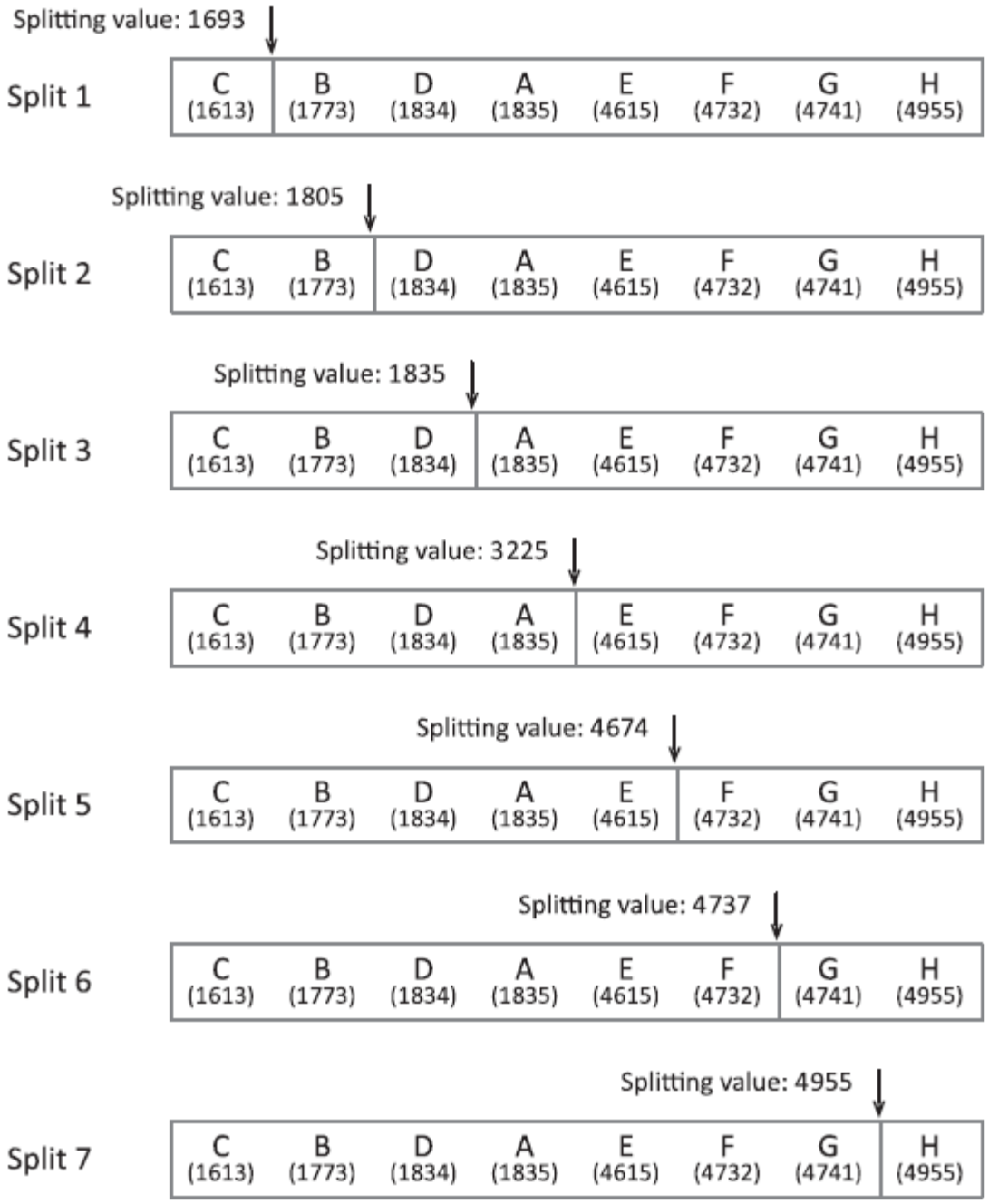
For the subset where *Weight* is less than 1835 (C, B, D):

$$\text{Average} = (35 + 31 + 27)/3 = 31$$
$$\text{SSE} = (35 - 31)^2 + (31 - 31)^2 + (27 - 31)^2 = 32$$

For the subset where *Weight* is greater than or equal to 1835 (A, E, F, H, G):

$$\text{Average} = (26 + 10 + 9 + 13 + 12)/5 = 14$$
$$\text{SSE} = (26 - 14)^2 + (10 - 14)^2 + (9 - 14)^2 + (13 - 14)^2 + (12 - 14)^2 = 190$$

Split score = 32 + 190 = 222



Decision Tree

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

N: number of observations
y_i: individual value for the response
$\bar{y}$: average value for the subset.

Observations	Weight	MPG
A	1,835	26
B	1,773	31
C	1,613	35
D	1,834	27
E	4,615	10
F	4,732	9
G	4,955	12
H	4,741	13

Split 4

For the subset where *Weight* is less than 3225 (C, B, D, A):

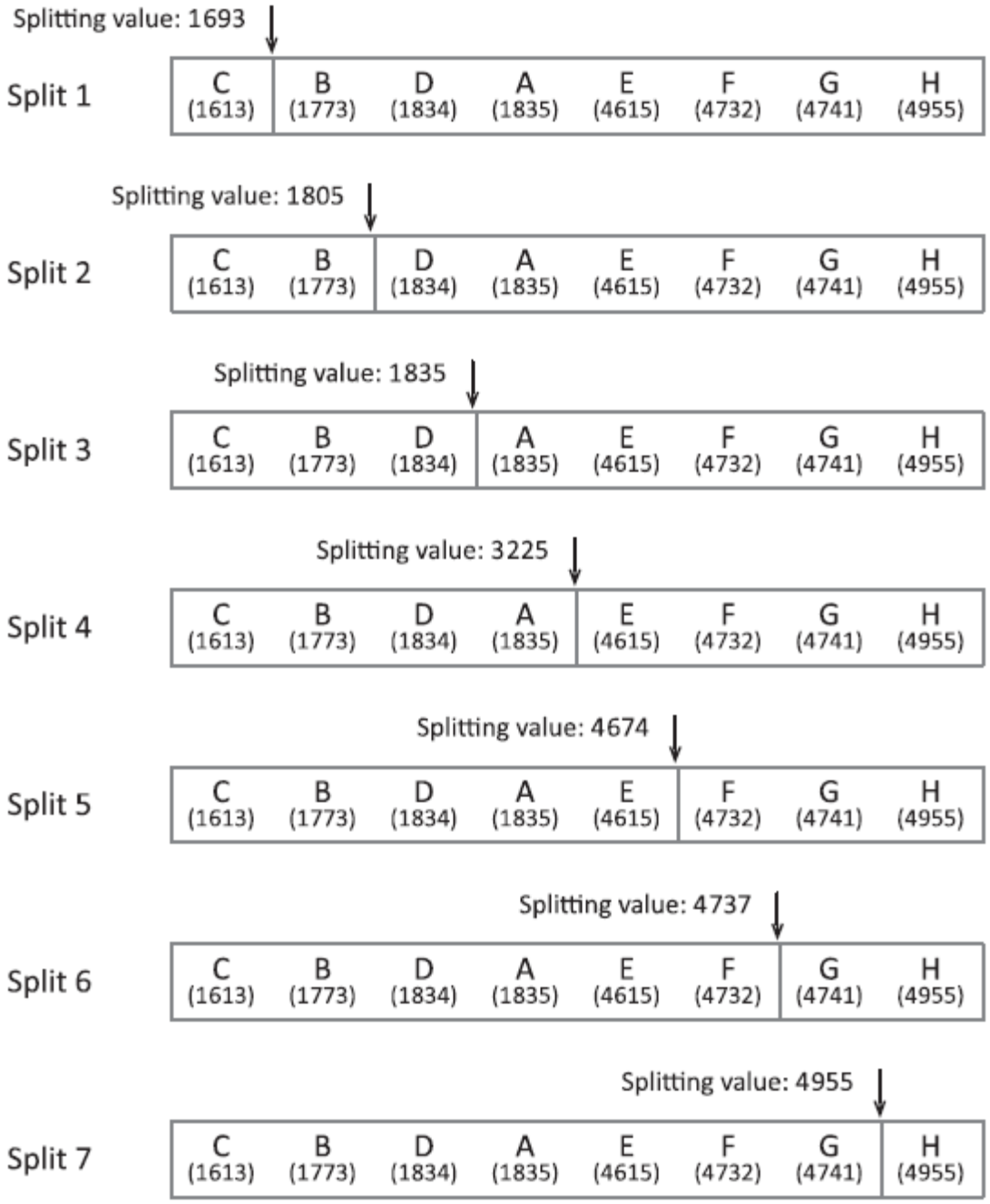
Average = (35 + 31 + 27 + 26)/4 = 29.75
SSE = $(35 - 29.75)^2 + (31 - 29.75)^2 + (27 - 29.75)^2 + (26 - 29.75)^2 = 50.75$

For the subset where *Weight* is greater than or equal to 3225 (E, F, H, G):

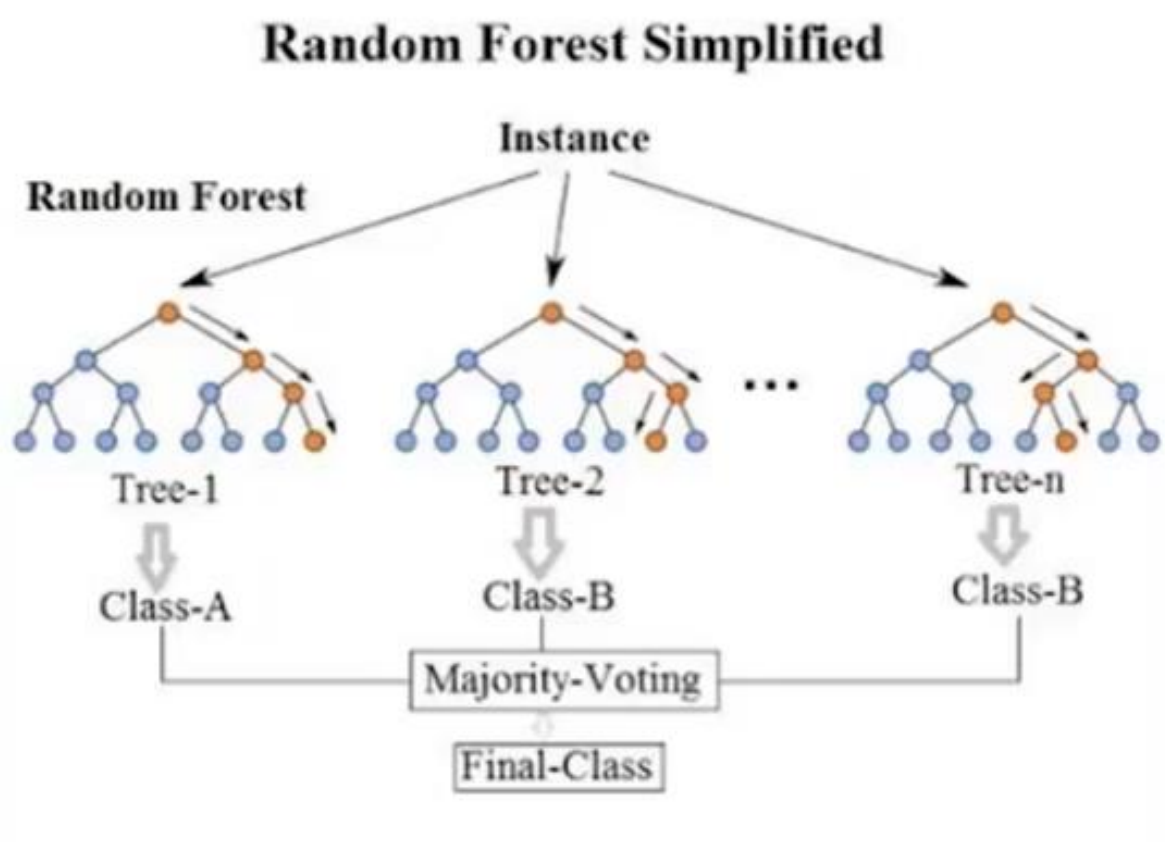
Average = (10 + 9 + 13 + 12)/4 = 11
SSE = $(10 - 11)^2 + (9 - 11)^2 + (13 - 11)^2 + (12 - 11)^2 = 10$

Split score = 50.75 + 10 = 60.75

Split 5 score = 362.8 + 8.67 = 371.47

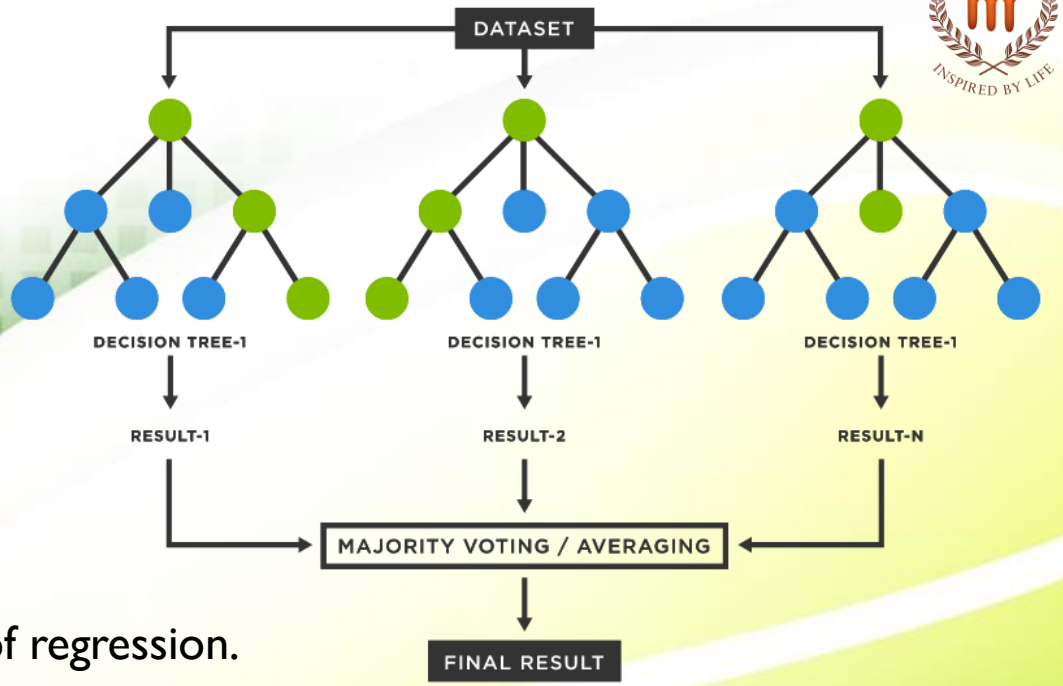


Random Forest



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

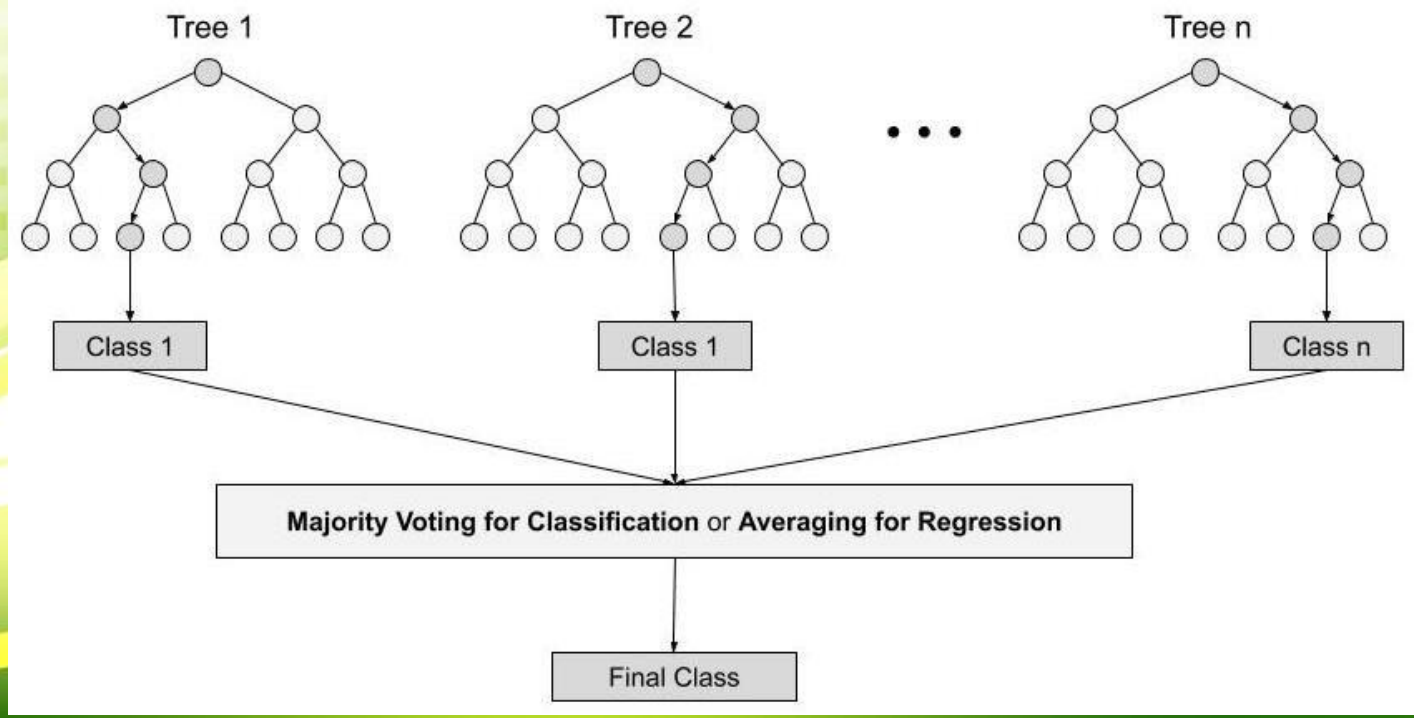
Random Forest



- *Supervised Learning Algorithm.*
- *used for Classification and Regression problems.*
- Random Forest has multiple decision trees as base learning models.
- builds decision trees on different samples
  - takes their majority vote for classification and average in case of regression.
- One important features of Random Forest Algorithm is that it can handle data set containing ***continuous variables*** as in case of regression and ***categorical variables*** as in case of classification.
  - performs better results for classification problems.
- basic idea behind this is to combine multiple decision trees in determining final output rather than relying on individual decision trees.

Random Forest

- Every decision tree has high variance, but when combine all of them together in parallel then resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence output doesn't depend on one decision tree but multiple decision trees.
  - In case of a classification problem, final output is taken by using the majority voting classifier.
  - In case of a regression problem, final output is the mean of all outputs. (Aggregation)
  - randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model (Bootstrap)



Random Forest

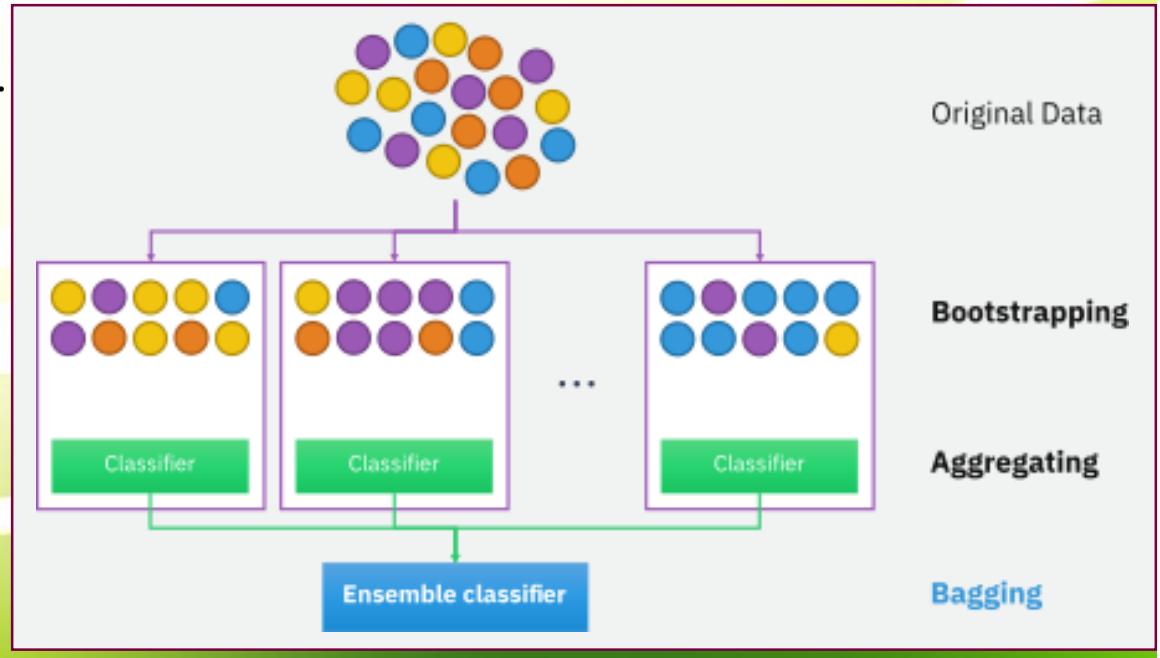
- A Random Forest is an ensemble technique.
- ***Ensemble*** simply means combining multiple models; thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. *Random Forest.*
2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. *ADA BOOST, XG BOOST*

Random Forest

- Bootstrap Sampling is a method that involves drawing of sample data repeatedly with replacement from a data source to estimate a population parameter.
- Bagging (***Bootstrap Aggregation)*** is the ensemble technique used by random forest.
  - Bagging chooses a random sample from the data set.
  - Each model is generated from samples (Bootstrap Samples) provided by Original Data with replacement (***row sampling)***.
  - This step of row sampling with replacement is called ***bootstrap***.
  - Now each model is trained independently which generates results.
  - The final output is based on majority voting after combining the results of all models.
  - This step which involves combining all the results and generating output based on majority voting is known as ***aggregation***.



Random Forest