BIG DATA ANALYTICS

MC-5101 (*After Mid-semester*)

Dr. Sumit Kumar Tetarave

Study Material- 02

Study Content from Mid-Sem to End-Sem

1. Classification Algorithms
   (Supervised algorithm)
2. Clustering Algorithms
   (Un-supervised algorithm)
3. Hypothesis Testing
4. Text Preprocessing

1. Classification Algorithms in Data Mining

- Classification is the operation of separating various entities into several classes.
  - These classes can be defined by
    - Business rules,
    - Class boundaries, or
    - Some mathematical function.
- Data mining has many classifiers/classification algorithms such as:

- ✓ Logistic regression
- ✓ **K-Nearest Neighbours Algorithm (kNN)**
- ✓ **Decision trees**
- ✓ Bayesian Classification
- ✓ Rule-based Classification
- ✓ Random Forest
- ✓ Support Vector Machines

Decision Tree

- A Decision Tree is a popular machine learning algorithm used for both classification and regression tasks.
- It is a tree-like structure that represents a series of decisions and their possible outcomes.
  - ✓Each internal node of the tree corresponds to a feature or attribute,
  - ✓each branch represents a decision based on that attribute, and
  - ✓each leaf node represents the final outcome or class label.
- Decision Trees are interpretable and easy to understand, making them useful for both analysis and prediction.

What is ID3 Algorithm?

- The ID3 (Iterative Dichotomiser 3) algorithm is one of the earliest and most widely used algorithms to create Decision Trees from a given dataset.
- It uses the concept of entropy and information gain to select the best attribute for splitting the data at each node.
  - ✓Entropy measures the uncertainty or randomness in the data, and
  - ✓Information gain quantifies the reduction in uncertainty achieved by splitting the data on a particular attribute.
- The ID3 algorithm recursively splits the dataset based on the attributes with the highest information gain until a stopping criterion is met,
  - resulting in a Decision Tree that can be used for classification tasks.

Steps to Create a Decision Tree using the ID3 Algorithm:

- **Step 1: Selecting the Root Node:**
  - ✓ Calculate the entropy of the target variable (class labels) based on the dataset.
  - ✓ The formula for entropy is: $Entropy(S) = -\Sigma (p_i * \log2(p_i))$
    where p_i is the proportion of instances belonging to class i.
- **Step 2: Calculating Information Gain:**
  - ✓ For each attribute in the dataset, calculate the information gain when the dataset is split on that attribute.
  - ✓ The formula: $Information\ Gain(S, A) = Entropy(S) - \Sigma ((|S_v| / |S|) * Entropy(S_v))$
    where S_v is the subset of instances for each possible value of attribute A, and $|S_v|$ is the number of instances in that subset.
- **Step 3: Selecting the Best Attribute:**
  - ✓ Choose the attribute with the highest information gain as the decision node for the tree.
- **Step 4: Splitting the Dataset:**
  - ✓ Split the dataset based on the values of the selected attribute.
- **Step 5: Repeat the Process:**
  - ✓ Recursively repeat steps 1 to 4 for each subset until a stopping criterion is met (e.g., the tree depth reaches a maximum limit or all instances in a subset belong to the same class).

Example: ID3 algorithm with a simple example of classifying whether to play tennis based on the conditions. Consider the following dataset:

Weather	**Temperature**	**Humidity**	**Wind**	**Play Tennis?**
Sunny	Hot	High	False	**No**
Sunny	Hot	High	True	**No**
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	**No**
Sunny	Cool	Normal	False	**Yes**
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	**Yes**
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Step 1: Calculating Entropy:

To calculate entropy, we first determine the proportion of positive and negative instances in the dataset:

- Positive instances (Play Tennis = Yes): 9
- Negative instances (Play Tennis = No): 5

- Entropy(S) = -[(9/14) * log2(9/14) + (5/14) * log2(5/14)] = -[(0.64*-0.44) + (0.35*-1.02)]= 0.28 + 0.35 = 0.63

Step 2: Calculating Information Gain:

We calculate the information gain for each attribute (Weather, Temperature, Humidity, Windy) and choose the attribute with the highest information gain as the root node.

- Information Gain(S, Weather) = Entropy(S) – [(5/14) * Entropy(Sunny) + (4/14) * Entropy(Overcast) + (5/14) * Entropy(Rainy)] = 0.63 –[.35*.66 + .28*0+ .35*.66] = 0.63 – [.462] = 0.17

[in Entropy(Sunny), (Play Tennis = Yes): 2 and (Play Tennis = No): 3. = -(2/5) * log2(2/5) – (3/5) * log2(3/5) = – 0.4*-0.91 -0.6*-0.51 = .36 + .30 = .66

[in Entropy(Overcast), (Play Tennis = Yes): 4 and (Play Tennis = No): 0. = -(4/4) * log2(4/4) – (0/5) * log2(0/5) = -0 – 0= 0

[in Entropy(Rainy), (Play Tennis = Yes): 3 and (Play Tennis = No): 2. = -(3/5) * log2(3/5) – (2/5) * log2(2/5) = -0.6*-0.51 – 0.4*-0.91= .30 + .36 = .66

Step 2: Calculating Information Gain: (continue…)

- Information Gain(S, Temperature) = Entropy(S) – [(4/14) * Entropy(Hot) + (4/14) * Entropy(Mild) + (6/14) * Entropy(Cool)] ≈ 0.029

- Information Gain(S, Humidity) = Entropy(S) – [(7/14) * Entropy(High) + (7/14) * Entropy(Normal)] ≈ 0.152

- Information Gain(S, Windy) = Entropy(S) – [(8/14) * Entropy(False) + (6/14) * Entropy(True)] ≈ 0.048

Step 4: Selecting the Best Attribute:

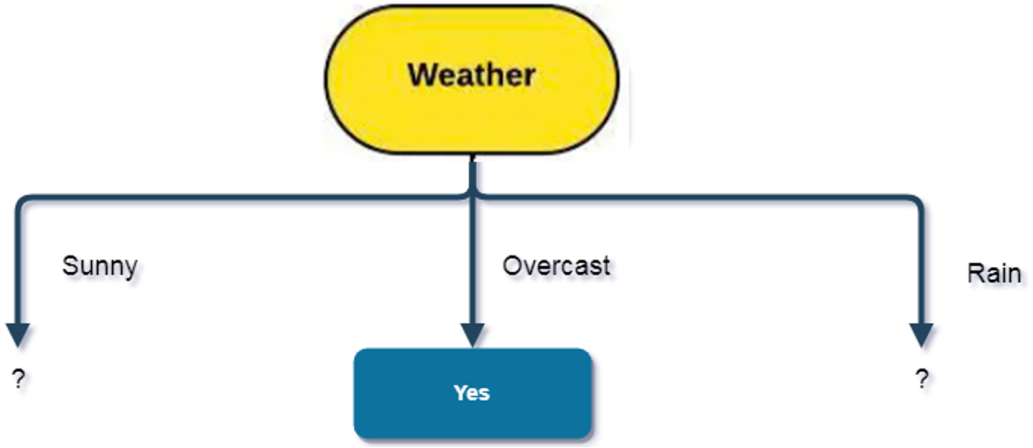
- The “Weather” attribute has the highest information gain, so we select it as the root node for our decision tree.

Step 5: Splitting the Dataset:

- We split the dataset based on the values of the “Weather” attribute into three subsets (Sunny, Overcast, Rainy).

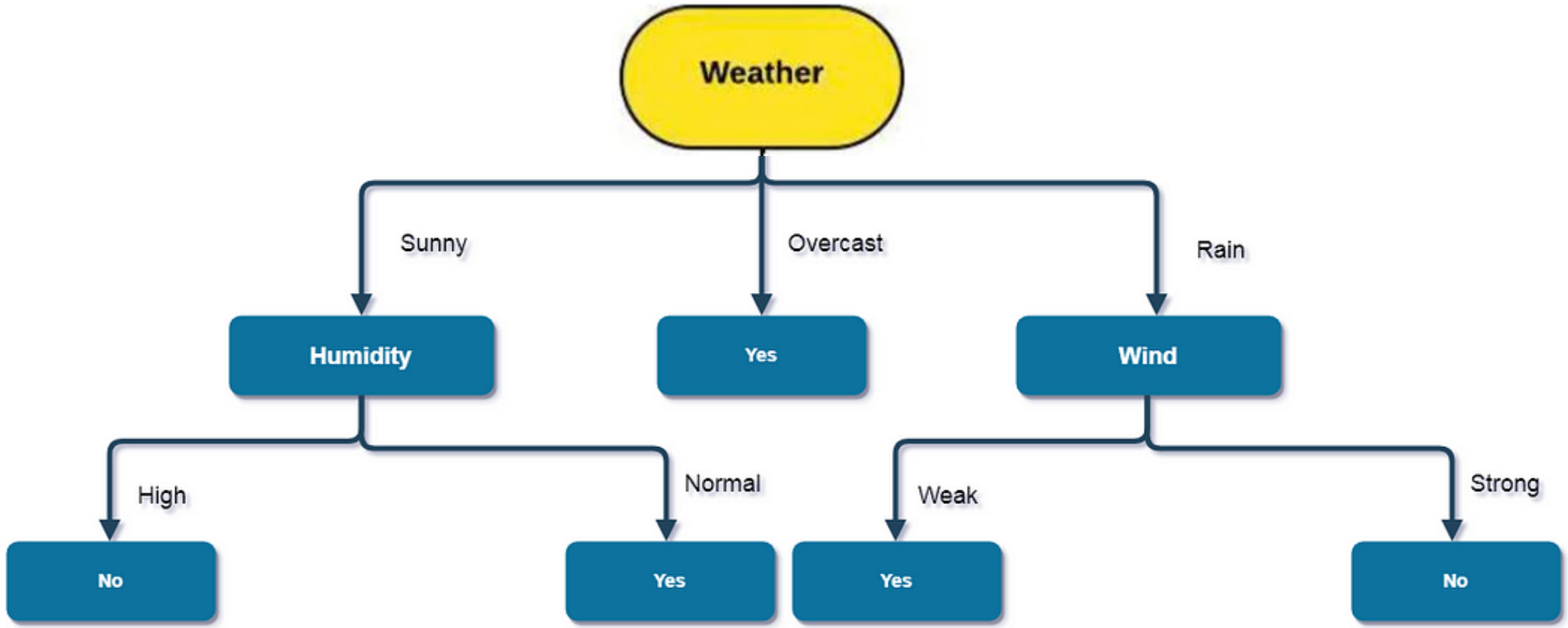
Step 6: Repeat the Process:

- If any value of the feature represents only one class (Ex. only rows with Play Tennis = ‘Yes’ or ‘No’) then we can say that the feature value represents a pure class. If the value does not represent a pure value, we have to extend it further until we find a pure class.



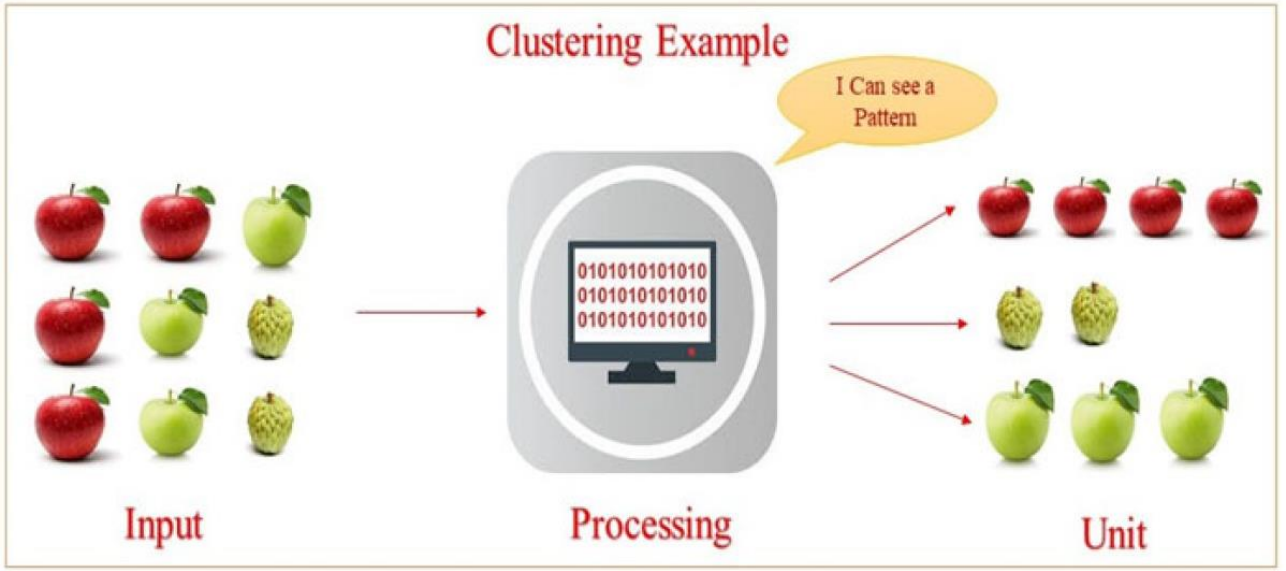
Weather	Temperature	Humidity	Windy	Play Tennis?
Sunny	Hot	High	False	**No**
Sunny	Hot	High	True	**No**
Sunny	Mild	High	False	**No**
Sunny	Cool	Normal	False	**Yes**
Sunny	Mild	Normal	True	**Yes**

Weather	Temperature	Humidity	Wind	Play Tennis?
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Rainy	Mild	Normal	False	Yes
Rainy	Mild	High	True	No


Weather
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Weak
Strong
No
Yes
Yes
No

2. CLUSTERING – AN OVERVIEW

- The clustering process, in general, is based on the approach that the data can be divided into an optimal number of "unknown" groups.
- The underlying stage of all the clustering algorithms is to find those hidden patterns and similarities,
  - without any intervention or predefined conditions.



2.1. General Approach to Clustering

- The quality assessment process of clustering results is regarded as cluster validation.
  - Cluster analysis is an iterative process of clustering and cluster verification by the user facilitated with
    - clustering algorithms,
    - cluster validation methods,
    - visualization and
    - domain knowledge to databases.

2.3. Applications of Cluster Analysis

- Clustering analysis is widely utilized in a variety of fields, including
  - data analysis, market research, pattern identification, and image processing.
- Earth observation databases use this data to identify
  - similar land regions and
  - to group houses in a city based on house type, value, and geographic position.
- It is the backbone of search engine algorithms,
  - where objects that are similar to each other must be presented together and dissimilar objects should be ignored.
  - Also, it is required to fetch objects that are closely related to a search term, if not completely related.
- Used in image segmentation in bioinformatics where
  - clustering algorithms have proven their worth in detecting cancerous cells from various medical imagery
    - – eliminating the prevalent human errors and other bias.
- Website network traffic can be divided into various segments and
  - heuristically when we can prioritize the requests and
  - also helps in detecting and preventing malicious activities.

2.4: Clustering Methods in Data Mining

- For a successful grouping there are two major goals –
  (i) Similarity between one data point with another
  (ii) Distinction of those similar data points with others which most certainly, heuristically differ from those points.
- To address the challenges such as scalability, attributes, dimensional, boundary shape, noise, and interpretation
  - there are various types of clustering methods to solve one or many of these problems.
- Various types of

Clustering methods:

- ✓ **Hierarchical Method:** Agglomerative and Divisive Approach
- ✓ **Density-based Method:** DBSCAN
- ✓ **Grid-Based Method:** STING and CLIQUE
- ✓ **Partitioning Method:** k-Means and k-Medoids
- ✓ **Model-Based Method:** Neural Network and other AI approaches
- ✓ **Constraint-based Method:** Tree-based algorithms with leaf clustering

2.6: Hierarchical Method: Agglomerative and Divisive Approach

- This method decomposes a set of data items into a hierarchy. Depending on how the hierarchical breakdown is generated, we can put hierarchical approaches into different categories. Following are the two approaches;

Agglomerative Approach

- This Algorithm is also referred as Bottom-up approach.
  - This approach treats each and every data point as a single cluster and
  - then merges each cluster by considering the similarity (distance) in each individual cluster
  - until a single large cluster is obtained or when some condition is satisfied.

Advantages

- Easy to identify nested clusters.
- Gives better results and ease in implementation.
- They are suitable for automation.
- Reduces the effect of initial values of cluster on the clustering results.
- Reduces the computing time and space complexity.

Disadvantages

- It can never undo what was done previously.
- Difficulty in handling different sized clusters and convex shapes lead to increase in time complexity
- There is no direct minimization of objective function.
- Sometimes there is difficulty in identifying the exact number of clusters by the Dendrogram.

Divisive Approach

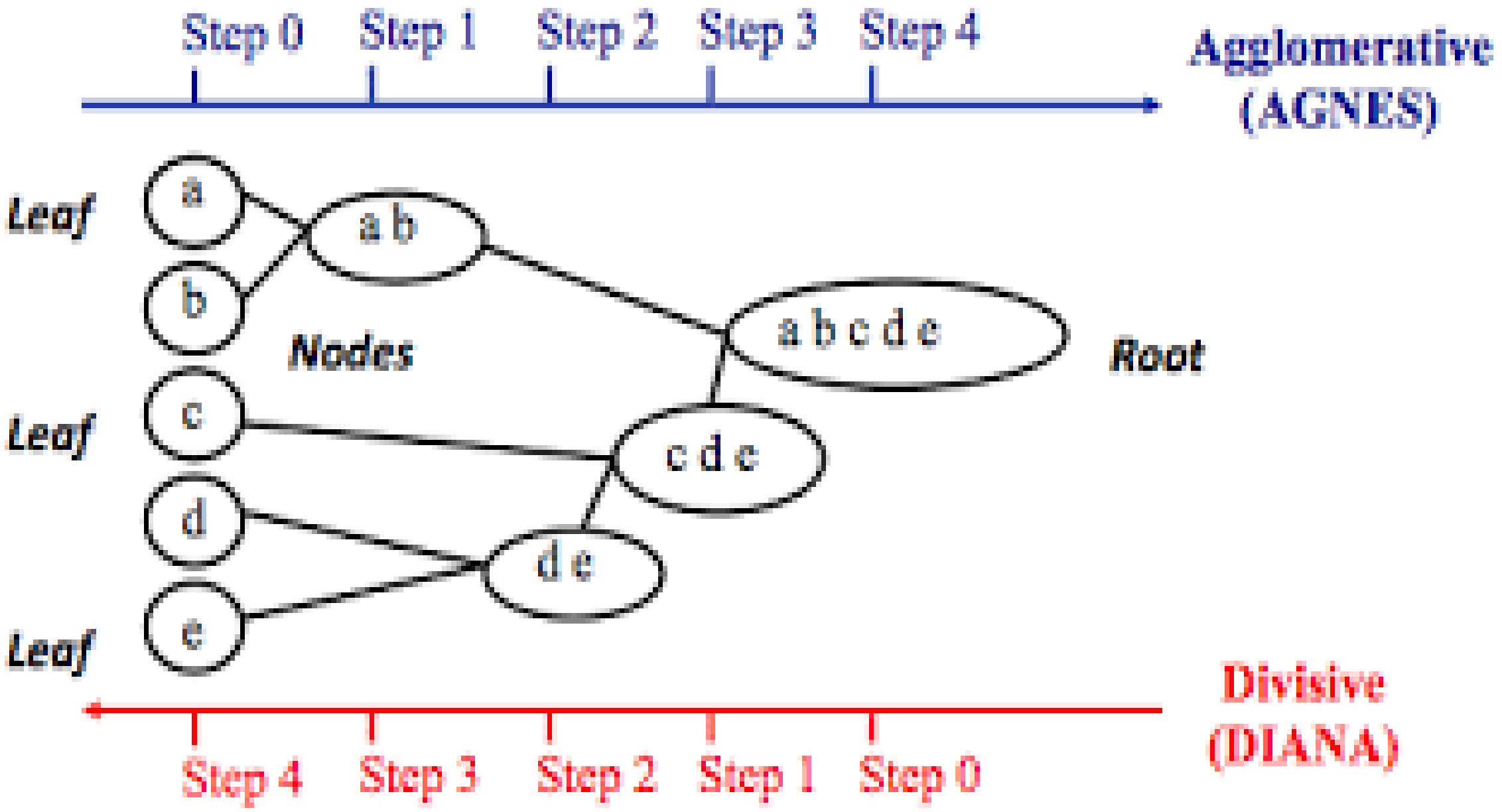
- This approach is also referred as the top-down approach.
- In this, we consider the entire data sample set as one cluster and
  - continuously splitting the cluster into smaller clusters iteratively.
  - It is done until each object in one cluster or the termination condition holds.
- This method is rigid, because once a merging or splitting is done, it can never be undone.

Advantage

- It produces more accurate hierarchies than bottom-up algorithm in some circumstances.

Disadvantages

- Top down approach is computationally more complex than bottom up approach because we need a second flat clustering algorithm.
- Use of different distance metrics for measuring distance between clusters may generate different results.


Step 0
Step 1
Step 2
Step 3
Step 4
Agglomerative
(AGNES)
Leaf
Leaf
Leaf
a
b
c
d
e
a b
c d e
d e
a b c d e
Nodes
Root
Divisive
(DIANA)
Step 4
Step 3
Step 2
Step 1
Step 0

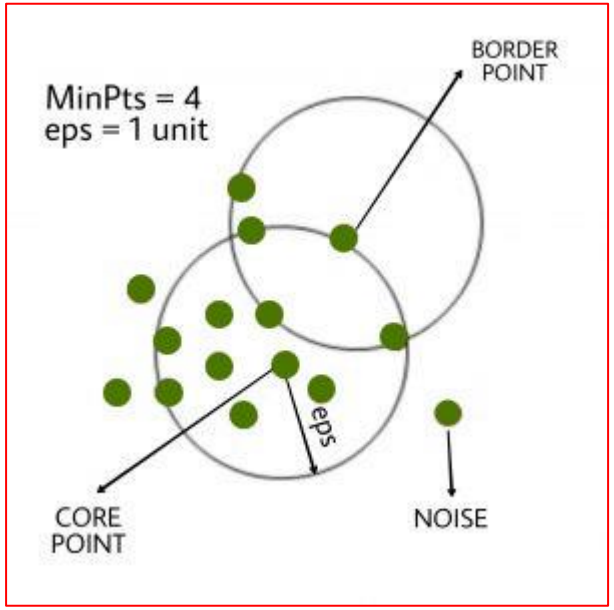
Density based Methods

- Density-based clustering identifies clusters based on the density of data points in the feature space.
- A cluster is defined as a high-density region separated by low-density areas.
- The most well-known algorithm in this category is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Parameters Required For DBSCAN Algorithm

1. **eps**: It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered neighbors.
   - ✓ If the eps value is chosen too small then a large part of the data will be considered as an outlier.
   - ✓ If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters.
   - ✓ One way to find the eps value is based on the ***k-distance graph***.
2. **MinPts**: Minimum number of neighbors (data points) within eps radius.
   - ✓ The larger the dataset, the larger value of MinPts must be chosen.
   - ✓ As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1.
   - ✓ The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.
Core Point: *A point is a core point if it has more than MinPts points within eps.*
Border Point: *A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.*
Noise or outlier: *A point which is not a core point or border point.*



Steps of DBSCAN:

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density-connected points and assign them to the same cluster as the core point.
4. A point a and b are said to be density connected
   - ✓if there exists a point c which has a sufficient number of points in its neighbors and both points a and b are within the eps distance.
   - ✓This is a chaining process. So, if b is a neighbor of c, c is a neighbor of d, and d is a neighbor of e, which in turn is neighbor of a implying that b is a neighbor of a.
5. Iterate through the remaining unvisited points in the dataset.
   - ✓Those points that do not belong to any cluster are **noise**.

Example:

- Consider the following points in a 2D plane: {(1,1),(1,2),(2,2),(8,8),(8,9),(25,25)}
- Algorithm parameters:ϵ =2 , min points=1
- Result:
  - Cluster1:{(1,1),(1,2),(2,2)}
  - Cluster2:{(8,8),(8,9)}
  - Outlier:(25,25)

Grid-Based Clustering:

- Grid-based clustering divides the data space into a grid of cells (a structured, grid-like representation), and then clustering operations are performed on these grids instead of individual data points.
- Two popular grid-based algorithms:
  1. STING (Statistical Information Grid Clustering Algorithm) and
  2. OPTICS (Ordering Point To Identify Clustering Structure Clustering Algorithm)

<u>How it works:</u>

- The data space is divided into cells (grids).
- For each cell, the density of points is calculated
  - ✓ (how many data points fall into the cell).
- Cells with a density above a certain threshold are considered to be part of a cluster.
- Clusters are formed by merging adjacent dense cells.

Partitioning Method

- Given a data set D of n objects and k no of clusters to form.
- A Partitioning Method organizes the objects into k partitions (k≤n).

Partition based algorithms determine all clusters at once. They include:

- ***K*-means and derivatives**
- Fuzzy *c*-means clustering

K-MEANS CLUSTERING

- k-means clustering is an algorithm
  - to cluster or
  - to group

the objects based on attributes/features into K number of group.

(K is positive integer number)

- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

Step 1: Begin with a decision on the value of k = number of clusters .

Step 2: Put any initial partition that classifies the data into k clusters.

✓ We may assign the training samples randomly, or systematically as the following:

1. Take the first k training sample as single-element clusters
2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

Step 3: Take each sample in sequence and compute its distance from the centroid of each of the clusters.

✓If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4 . Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

A Simple example showing the implementation of k-means algorithm (using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2,

we obtain two clusters containing:

{1,2,3} and {4,5,6,7} using minimum Euclidean distance between each point and the centroids.

$$d(m_1,2)=\sqrt{|1.0-1.5|^2+|1.0-2.0|^2}=1.12$$
$$d(m_2,2)=\sqrt{|5.0-1.5|^2+|7.0-2.0|^2}=6.10$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

we will find the mean of the x and y coordinates of each point in the cluster. Their new centroids for Iteration 2 are:

$$m_1=(\frac{1}{3}(1.0+1.5+3.0),\frac{1}{3}(1.0+2.0+4.0))=(1.83,2.33)$$

$$m_2=(\frac{1}{4}(5.0+3.5+4.5+3.5),\frac{1}{4}(7.0+5.0+5.0+4.5))=(4.12,5.38)$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

Step 4 :

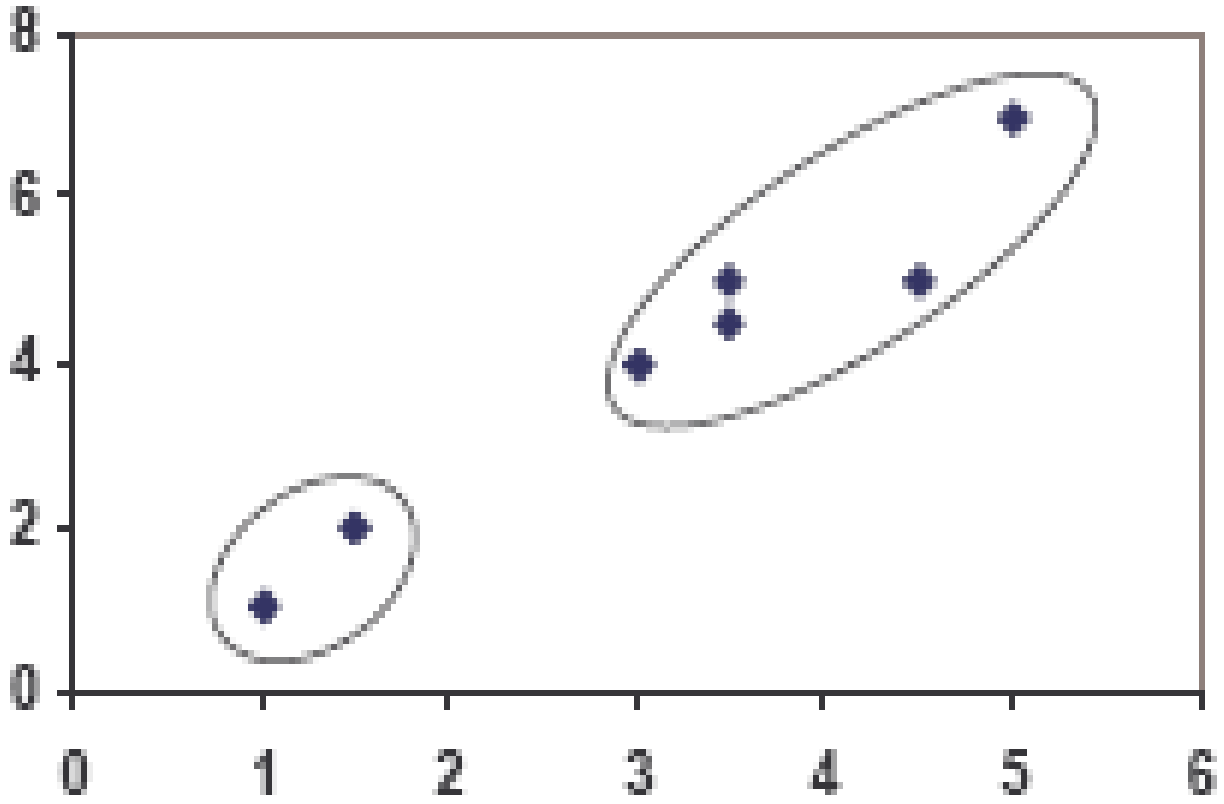
The clusters obtained are:

{1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



Limitations of K-means

- Choosing the value of k.
- Sensitive to outlier .

  Ex.

  D={1,2,3,8,9,10,25}, K=2

  c1=(1,2,3} c2={8,9,10,25}

3. Hypothesis Testing

- A hypothesis is an assumption or idea, specifically a statistical claim about an unknown population parameter.

    - For example, a judge assumes a person is innocent and verifies this by reviewing evidence and hearing testimony before reaching a verdict.

Examples:

- It is a statement about one or more populations.
- It is usually concerned with the parameters of the population.
  - e.g. the hospital administrator may want to test the hypothesis that the average length of stay of patients admitted to the hospital is 5 days.

What is Hypothesis Testing?

- Hypothesis testing is a statistical method that is used to make a statistical decision using experimental data.
- Hypothesis testing is basically an assumption that we make about a population parameter.
- It evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data.

- To test the validity of the claim or assumption about the population parameter:
  - A sample is drawn from the population and analyzed.
  - The results of the analysis are used to decide whether the claim is true or not.
    - Example:
      - You say an average height in the class is 30 or a boy is taller than a girl.
      - All of these are assumptions that we are assuming, and we need some statistical way to prove these.
      - We need some mathematical conclusion whatever we are assuming is true.

Defining Hypotheses

- Null hypothesis (H_0):
  - In statistics, the null hypothesis is a general statement or default position that there is no relationship between two measured cases or no relationship among groups.
  - In other words, it is a basic assumption made based on the problem knowledge.
- Example: A company's mean production is 50 units/per day
  - **H_0: $\mu = 50$**

- Alternative hypothesis (H_1): The alternative hypothesis is the hypothesis used in hypothesis testing that is contrary to the null hypothesis.
- Example: A company's production is not equal to 50 units/per day
    - i.e. $H_1: \mu \neq 50$.

Type-I and Type-2 Error

Possible Action

Condition of NULL Hypothesis

	True	**False**
Fail to Reject H_0	Correct Action	**Type-II Error**
Reject H_0	**Type-I Error**	Correct Action

NOTE: If $\mathbf{H_0}$ is rejected then we conclude $\mathbf{H_A}$ is true. If $\mathbf{H_0}$ is not rejected we conclude $\mathbf{H_0}$ may be is true.

Key Terms of Hypothesis Testing-1

- Level of significance(α):
- Probability of Type-I Error.
  - It refers to the degree of significance in which we accept or reject the null hypothesis.
  - This is normally denoted with α and generally, it is 0.05 or 5%, which means your output should be 95% confident to give a similar kind of result in each sample.

Key Terms of Hypothesis Testing-2

- **Power of a hypothesis test: 1-β**
  - Probalility of Type − II error: β

    This occurs when you fail to reject the null hypothesis (H0) even though it is false.
  - Power is the probability of correctly rejecting the null hypothesis when it is false
    - A high power means that the test is sensitive and has a low chance of a Type II error

One tail test and Two Tail test

- A one-tailed test (or one-sided test) in hypothesis testing is used when we are interested in determining whether a parameter (such as a mean or proportion) is either greater than or less than a specific value, but not both.
  - A one-tailed test focuses on one direction of interest in the distribution.
- A two-tailed test (or two-sided test) in hypothesis testing is used when we are interested in detecting any significant difference in either direction — whether a parameter is greater than or less than a specific value.

Critical region: Key Terms of Hypothesis Testing-3

- The critical region in hypothesis testing is the range of values for a test statistic that leads to the rejection of the null hypothesis (H_o).
  - the critical region is determined based on the significance level α.

Example :

- Testing a coin for fairness.

Formulate the hypotheses

- H_o= the coin is fair
- H_1= The coin is not fair.

✓H_o= probability of heads is 0.5

✓H_1= probability of heads is different from 0.5.

Let $\alpha = 0.05$, meaning there is a 5% chance of rejecting null hypothesis when it is true.

1. Flip the coin 20 times and count the number of heads.
   - Let we get head 7 times.
2. critical values for two sided test for $\alpha=0.05$ are 5 and 15.

➢We got 7 heads, which is within the range of 5 to 15.

➢Since 7 is not in the critical region, you fail to reject the null hypothesis.

➢This means there is no strong evidence that the coin is unfair.

Critical region in one tell test

- In a one-tailed test, the entire significance level (alpha, α) is placed on one side of the distribution.
- The critical region is determined either in the right tail (for a right-tailed test) or the left tail (for a left-tailed test).
- For example, if $\alpha = 0.05$, the critical region lies in the extreme 5% of the distribution on one side (either the right or left).

Z-Test

- The Z-test is a statistical test used to determine whether there is a significant difference between sample and population means or between the means of two samples.
- Type of Test:

  - One-Sample Z-Test: Used to compare the sample mean to a known population mean.
  - Two-Sample Z-Test: Used to compare the means of two independent samples.
  - Z-Test for Proportions: Used to compare sample proportions to known proportions or between two proportion

Assumption:

- Normality: The data should be approximately normally distributed. For large sample sizes (typically n> 30)

- Known Population Variance: The population standard deviation (σ) must be known. In practice, this is often assumed or approximated using sample data.

One-Sample Z-Test:

- Purpose: To test if the sample mean is significantly different from a known population mean.
- Hypotheses:
  - Null Hypothesis (H_0): $\mu = \mu_0$ (the sample mean is equal to the population mean)
  - Alternative Hypothesis (H_1): $\mu \neq \mu_0$ (the sample mean is different from the population mean) - for a two-tailed test

  - Test Statistics:

$$Z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$$

Where $\bar{x}$: sample mean

μ_0 : Population mean

σ : population standard deviation

n : Sample size

Example:

- You are a quality control manager for a company that produces light bulbs. The company claims that the average lifetime of their light bulbs is 1,000 hours. You believe that the actual average lifetime might be different due to recent changes in the production process.

- Objective:
  - Test whether the average lifetime of the light bulbs is significantly different from 1,000 hours using a one-sample Z-test.

Given Data:

- Population Mean: 1,000 hours (claimed average lifetime)
- Sample Mean: 1,020 hours (observed average lifetime from a sample)
- Population Standard Deviation (σ): 50 hours
- Sample Size (n): 30
- Significance Level (α): 0.05 (two-tailed test)

Steps:

- Null Hypothesis (H_o): The average lifetime of the light bulbs is 1,000 hours.
  - H_o : μ=1000
- Alternative Hypothesis (H_1): The average lifetime of the light bulbs is different from 1,000 hours.
  - H_1: $\mu \neq$ 1,000

- $Z \approx 2.19$

- Determine the Critical Values:
- For a two-tailed test with $\alpha = 0.05$, the critical Z-values are , ± 1.96 (from standard Z-tables).
- Make the Decision:

  - If the absolute value of the calculated Z-value is greater than the critical value, reject the null hypothesis.

Two sample Z-test

- A two-sample Z-test is a statistical test used to compare the means of two independent samples to determine whether they are significantly different from each other.
- Assumptions:
  - Data from each sample should be independent.
  - Each sample should be normally distributed (or approximately normally distributed, especially for large sample sizes).
  - Population variances should be known.
  - If population variances are unknown and sample sizes are small, a t-test might be more appropriate.

- Test Statistic:
  - The Z-test statistic for two independent samples is calculated as:

$$Z = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Where,

$\overline{x_1}$, $\overline{x_2}$, : are the sample means of the two groups.

σ_1, σ_2: are the population standard deviations of the two groups.

n_1, n_2 : are the sample sizes of the two groups.

Decision Rule:

- Calculate the p-value associated with the test statistic Z.
- Compare the p-value to the significance level (usually denoted as α, typically 0.05).
- If the p-value is less than α, reject the null hypothesis and conclude that there is a statistically significant difference between the means of the two samples. If not, fail to reject the null hypothesis.

4: Text Mining and its Applications

- Almost 80% of data in the world resides in an unstructured format
  - Therefore, text mining is an extremely valuable practice within organizations.
- Text mining tools and Natural Language Processing (NLP) techniques, like information extraction,
  - allow us to transform unstructured documents into a structured format
    - to enable analysis and the generation of high-quality insights.
- This, in turn, improves the decision-making of organizations, leading to better business outcomes.

4.1) Text Mining and its Applications…

Text mining often includes the following techniques:

- **Information extraction** is a technique for extracting domain specific information from texts.
  - Text fragments are mapped to field or template lots that have a definite semantic technique.
- **Text summarization** involves identifying, summarizing and organizing related text so that users can efficiently deal with information in large documents.
- **Text categorization** involves organizes documents into a taxonomy, thus allowing for more efficient searches.
  - It involves the assignment of subject descriptors or classification codes or abstract concepts to complete texts.
- **Text clustering** involves automatically clustering documents into groups where documents within each group share common features.

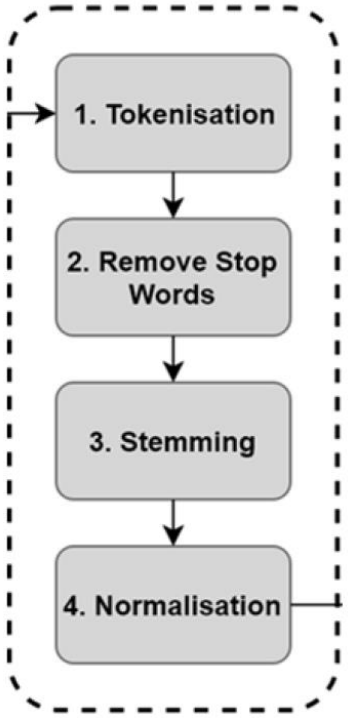
4.2) Text Mining and its Applications…

Following are some of the applications of Text Mining:

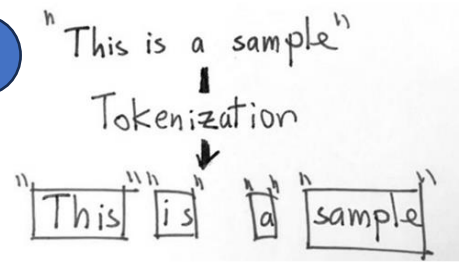
- **Customer service:** There are various ways in which we invite customer feedback from our users.
  - When combined with text analytics tools, feedback systems such as chatbots, customer surveys, Net-Promoter Scores, online reviews, support tickets, and social media profiles, enable companies to improve their customer experience with speed.
  - Text mining and sentiment analysis can provide a mechanism for companies to prioritize key pain points for their customers, allowing businesses to respond to urgent issues in realtime and increase customer satisfaction.
- **Risk management:** Text mining also has applications in risk management.
  - It can provide insights around industry trends and financial markets by monitoring shifts in sentiment and by extracting information from analyst reports and whitepapers.
  - This is particularly valuable to banking institutions as this data provides more confidence when considering business investments across various sectors.
- **Maintenance:** Text mining provides a rich and complete picture of the operation and functionality of products and machinery.
  - Over time, text mining automates decision making by revealing patterns that correlate with problems and preventive and reactive maintenance procedures.
  - Text analytics helps maintenance professionals unearth the root cause of challenges and failures faster.
- **Healthcare:** Text mining techniques have been increasingly valuable to researchers in the biomedical field, particularly for clustering information.
  - Manual investigation of medical research can be costly and time-consuming; text mining provides an automation method for extracting valuable information from medical literature.
- **Spam filtering:** Spam frequently serves as an entry point for hackers to infect computer systems with malware.
  - Text mining can provide a method to filter and exclude these e-mails from inboxes, improving the overall user experience and minimizing the risk of cyber-attacks to end users.

4.3: Text Preprocessing

- Text preprocessing is an approach for cleaning and preparing text data for use in a specific context.
- The ultimate goal of cleaning and preparing text data is to reduce the text to only the words that you need for your NLP goals.
- Once you have a clear idea of the type of application you are developing and the source and nature of text data, you can decide on which preprocessing stages can be added to your NLP pipeline.
- Most of the NLP toolkits on the market include options for all of the preprocessing stages such as:



1



2 "Stop words" are frequently occurring words used to construct sentences.

Sample Text with Stop Words	Without Stop Words
TextMining – A technique of data mining for analysis of web data	TextMining, technique, datamining, analysis, web, data
The movie was awesome	Movie, awesome
The product quality is bad	Product, quality, bad

3

4 **Normalization**

- Upper or lowercasing
- Stopword removal
- Stemming – bluntly removing prefixes and suffixes from a word
- Lemmatization – replacing a single-word token with its root

4.4: BoW and TF-IDF For Creating Features from Text

- We understand that sentence in a fraction of a second, such as
  - Review 1: This movie is very scary and long
  - Review 2: This movie is not scary and is slow
  - Review 3: This movie is spooky and good
- But machines simply cannot process text data in raw form. They need us to break down the text into a numerical format that’s easily readable by the machine.
- This is where the two concepts come into play
  - Bag-of-Words (BoW) and
  - Term Frequency-Inverse Document Frequency (TF-IDF).
- Both BoW and TF-IDF are techniques that help us convert text sentences into numeric vectors.

4.5: Bag-of-Words (BoW)

- We will first build a vocabulary from all the unique vocabulary, which consists of these 11 words:
- 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'.
- We can now take each of these words and mark their occurrence in the three movie reviews above with 1s and 0s.
- This will give us 3 vectors for 3 reviews as a Vector Representation:

Review 1: This movie is very scary and long
Review 2: This movie is not scary and is slow
Review 3: This movie is spooky and good

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the Review (in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	1	1	0	1	1	0	0	8
Review 3	1	1	1	0	0	1	0	0	0	1	1	6

- Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]
- Vector of Review 2: [1 1 2 0 1 1 0 1 1 0 0]
- Vector of Review 3: [1 1 1 0 0 1 0 0 0 1 1]

And that's the core idea behind a Bag of Words (BoW) model.

Drawbacks of using a BoW

- If the new sentences contain new words, then
  - our vocabulary size would increase and
  - ✓ thereby, the length of the vectors would increase too.
- Additionally,
  - the vectors would also contain many 0s,
  - ✓ thereby resulting in a sparse matrix (which is what we would like to avoid)
- We are retaining no information
  - on the grammar of the sentences nor
  - on the ordering of the words in the text.

4.6: Term Frequency-Inverse Document Frequency (TF-IDF)

- Term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- **Term Frequency (TF)**: is a measure of how frequently a term, t, appears in a document, d:

$$tf_{td} = \frac{n_{t,d}}{number\ of\ terms\ in\ a\ document}$$

- Here, in the numerator, n is the number of times the term "t" appears in the document "d".
- Thus, each document and term would have its own TF value.
- Example: How to calculate the TF for

Review #2: *This movie is not scary and is slow.*

Here,
- Vocabulary: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'
- Number of words in Review 2 = 8
- TF for the word 'this' = (number of times 'this' appears in review 2)/(number of terms in review 2) = 1/8

Similarly,
- TF('movie') = 1/8
- TF('is') = 2/8 = 1/4
- TF('very') = 0/8 = 0
- TF('scary') = 1/8
- TF('and') = 1/8
- TF('long') = 0/8 = 0
- TF('not') = 1/8
- TF('slow') = 1/8
- TF('spooky') = 0/8 = 0
- TF('good') = 0/8 = 0

Term	Review 1	Review 2	Review 3	TF (Review1)	TF (Review2)	TF (Review3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

Inverse Document Frequency (IDF)

- Computing just the TF alone is not sufficient to understand the importance of words, thus, we need the IDF value
- IDF is a measure of how important a term is.

$$idf_t = \log \frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'}$$

- Example: We can calculate the IDF values for the all the words in Review 2:
  - IDF('this') =

= log(number of documents/number of documents containing the word 'this')

= log(3/3) = log(1) = 0

Similarly,

- IDF('movie',) = log(3/3) = 0
- IDF('is') = log(3/3) = 0
- IDF('not') = log(3/1) = 0.48
- IDF('scary') = log(3/2) = 0.18
- IDF('and') = log(3/3) = 0
- IDF('slow') = log(3/1) = 0.48

✓ Hence, we see that words like "is", "this", "and", etc., are reduced to 0 and have little importance;

✓ while words like "scary", "long", "good", etc. are words with more importance and thus have a higher value.

Term	Review 1	Review 2	Review 3	IDF
This	1	1	1	0.00
movie	1	1	1	0.00
is	1	2	1	0.00
very	1	0	0	0.48
scary	1	1	0	0.18
and	1	1	1	0.00
long	1	0	0	0.48
not	0	1	0	0.48
slow	0	1	0	0.48
spooky	0	0	1	0.48
good	0	0	1	0.48

Compute the TF-IDF score

- We can now calculate the TF-IDF score for every word in Review 2:
  - TF-IDF('this', Review 2) = TF('this', Review 2) * IDF('this') = 1/8 * 0 = 0

Similarly,
- TF-IDF('movie', Review 2) = 1/8 * 0 = 0
- TF-IDF('is', Review 2) = 1/4 * 0 = 0
- TF-IDF('not', Review 2) = 1/8 * 0.48 = 0.06
- TF-IDF('scary', Review 2) = 1/8 * 0.18 = 0.023
- TF-IDF('and', Review 2) = 1/8 * 0 = 0
- TF-IDF('slow', Review 2) = 1/8 * 0.48 = 0.06

- Words with a higher score are more important, and
  - those with a lower score are less important
- TF-IDF gives larger values for less frequent words.
- It also gives large value for frequent words in a single document but, rare in all the documents combined
  - Means, both IDF and TF values are high

Similarly, we can calculate the TF-IDF scores for all the words with respect to all the reviews:

Term	Review 1	Review 2	Review 3	IDF	TF (Review1)	TF (Review2)	TF (Review3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.025	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

Practices

Practice 1 : Implementation of k-means algorithm to create 2-cluster for the given data

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

Solution:

1. First, we will randomly choose 3 centroids from the given data.

Let us consider A2 (2,6), A7 (5,10), and A15 (6,11) as the centroids of the initial clusters.

Hence, we will consider that

- Centroid 1 = (2,6) is associated with cluster 1.
- Centroid 2 = (5,10) is associated with cluster 2.
- Centroid 3 = (6,11) is associated with cluster 3.

2. Now we will find the Euclidean distance between each point and the centroids. Based on the minimum distance of each point from the centroids, we will assign the points to a cluster.

Point	Distance from Centroid 1 (2,6)	Distance from Centroid 2 (5,10)	Distance from Centroid 3 (6,11)	Assigned Cluster
A1 (2,10)	4	3	4.123106	Cluster 2
A2 (2,6)	0	5	6.403124	Cluster 1
A3 (11,11)	10.29563	6.082763	5	Cluster 3
A4 (6,9)	5	1.414214	2	Cluster 2
A5 (6,4)	4.472136	6.082763	7	Cluster 1
A6 (1,2)	4.123106	8.944272	10.29563	Cluster 1
A7 (5,10)	5	0	1.414214	Cluster 2
A8 (4,9)	3.605551	1.414214	2.828427	Cluster 2
A9 (10,12)	10	5.385165	4.123106	Cluster 3
A10 (7,5)	5.09902	5.385165	6.082763	Cluster 1
A11 (9,11)	8.602325	4.123106	3	Cluster 3
A12 (4,6)	2	4.123106	5.385165	Cluster 1
A13 (3,10)	4.123106	2	3.162278	Cluster 2
A14 (3,8)	2.236068	2.828427	4.242641	Cluster 1
A15 (6,11)	6.403124	1.414214	0	Cluster 3

3. Now, we will calculate the new centroid for each cluster.

- In cluster 1, we have 6 points i.e. A2 (2,6), A5 (6,4), A6 (1,2), A10 (7,5), A12 (4,6), A14 (3,8).
  - ✓ To calculate the new centroid for cluster 1, we will find the mean of the x and y coordinates of each point in the cluster. Hence, the new centroid for cluster 1 is (3.833, 5.167).
- In cluster 2, we have 5 points i.e. A1 (2,10), A4 (6,9), A7 (5,10) , A8 (4,9), and A13 (3,10).
  - ✓ Hence, the new centroid for cluster 2 is (4, 9.6)
- In cluster 3, we have 4 points i.e. A3 (11,11), A9 (10,12), A11 (9,11), and A15 (6,11).
  - ✓ Hence, the new centroid for cluster 3 is (9, 11.25).

4. Stop step 3 if we observe that no point has changed its cluster compared to the previous iteration.

Practice 2: The quality-control manager at a light bulb factory needs to determine whether the mean life of a large shipment of light bulbs is equal to 375 hours. The population standard deviation is 120 hours. A random sample of 64 light bulbs indicates a sample mean life of 350 hours. (Significance Level (α): 0.05 and the critical Z-values are , ± 1.96)
Objective:

a) At the 0.05 level of significance, is there evidence that the mean life is different from 375 hours?
b) Compute the z-value and interpret its meaning.

Find Z for Test Statistics:

$$Z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}$$

Where $\bar{x}$: sample mean

μ_0 : Population mean

σ : population standard deviation

n : Sample size

Practice 3: Suppose we are looking for documents using the query *Q* and our database is composed of the documents *D1*, *D2,* and *D3*.

- *Q*: The cat.
- *D1*: The cat is on the mat.
- *D2*: My dog and cat are the best.
- *D3*: The locals are playing.

Let's compute the TF scores of the words "the" and "cat".

TF("the", D1) = 2/6 = 0.33
TF("the", D2) = 1/7 = 0.14
TF("the", D3) = 1/4 = 0.25
TF("cat", D1) = 1/6 = 0.17
TF("cat", D2) = 1/7 = 0.14
TF("cat", D3) = 0/4 = 0

Let's compute the IDF scores of the words "the" and "cat".

IDF("the") = log(3/3) = log(1) = 0
IDF("cat") = log(3/2) = 0.18

Let's compute the TF-IDF scores of the words "the" and "cat".

TF-IDF("the", D1) = 0.33 * 0 = 0
TF-IDF("the, D2) = 0.14 * 0 = 0
TF-IDF("the", D3) = 0.25 * 0 = 0
TF-IDF("cat", D1) = 0.17 * 0.18= 0.0306
TF-IDF("cat, D2) = 0.14 * 0.18= 0.0252
TF-IDF("cat", D3) = 0 * 0 = 0

- ✓ The next step is to use a ranking function to order the documents according to the TF-IDF scores of their words.
- ✓ We can use the average TF-IDF word scores over each document to get the ranking of *D1*, *D2,* and *D3* with respect to the query *Q*.

Average TF-IDF of D1 = (0 + 0.0306) / 2 = 0.0153
Average TF-IDF of D2 = (0 + 0.0252) / 2 = 0.0126
Average TF-IDF of D3 = (0 + 0) / 2 = 0

Practice 4: A binary classifier was evaluated using a set of 1,00 test examples in which 50% of all examples are positive. It was found that the classifier has 50% sensitivity and 60% accuracy. Write the confusion matrix. Compute the classifier's precision and F-measure.

Solution:

- Actual –ve and +ve = 100 and given, actual +ve (i.e., TP+FN) = 50, therefore, actual –ve (i.e., TN+FP) = 100 – 50 = 50.
- Sensitivity (i.e., Recall) = TP/(TP+ FN) = 50, TP/50 = 50/100, TP= 25, then FN = 25.
- Accuracy = (TP+TN)/(TP+FP+TN+FN) = (20+TN)/(100)= 60/100,

  TN= 40 and FP = 10

Finally, TP= 25, FP = 10 , TN= 40 and FN = 25

✓Precision= TP/TP+FP = 25/35= 5/7

✓F-Measure= (2* Recall * Precision)/ (Recall+ Precision) = 2*50*(5/7)/((350+5)/7)= 500/355

All the Best!!!