Data Science & Big Data Analytics

Subject Code: 310251

T. E. Computer (2019 Pattern)

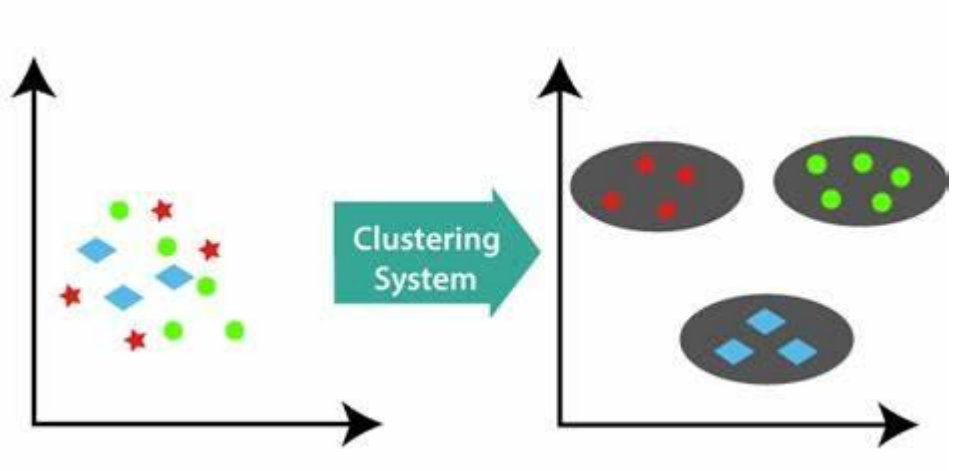
UNIT V

Unit V	Big Data Analytics and Model Evaluation	07 Hours
Clustering Algorithms: K-Means, Hierarchical Clustering, Time-series analysis. **Introduction to Text Analysis**: Text-preprocessing, Bag of words, TF-IDF and topics. Need and Introduction to social network analysis, Introduction to business analysis. **Model Evaluation and Selection**: Metrics for Evaluating Classifier Performance, Holdout Method and Random Sub sampling, Parameter Tuning and Optimization, Result Interpretation, Clustering and Time-series analysis using Scikit-learn, sklearn. metrics, Confusion matrix, AUC-ROC Curves, Elbow plot.		
#Exemplar/Case Studies	Use IRIS dataset from Scikit and apply K-means clustering methods	
***Mapping of Course Outcomes for Unit V**	CO4, CO2	

Clustering

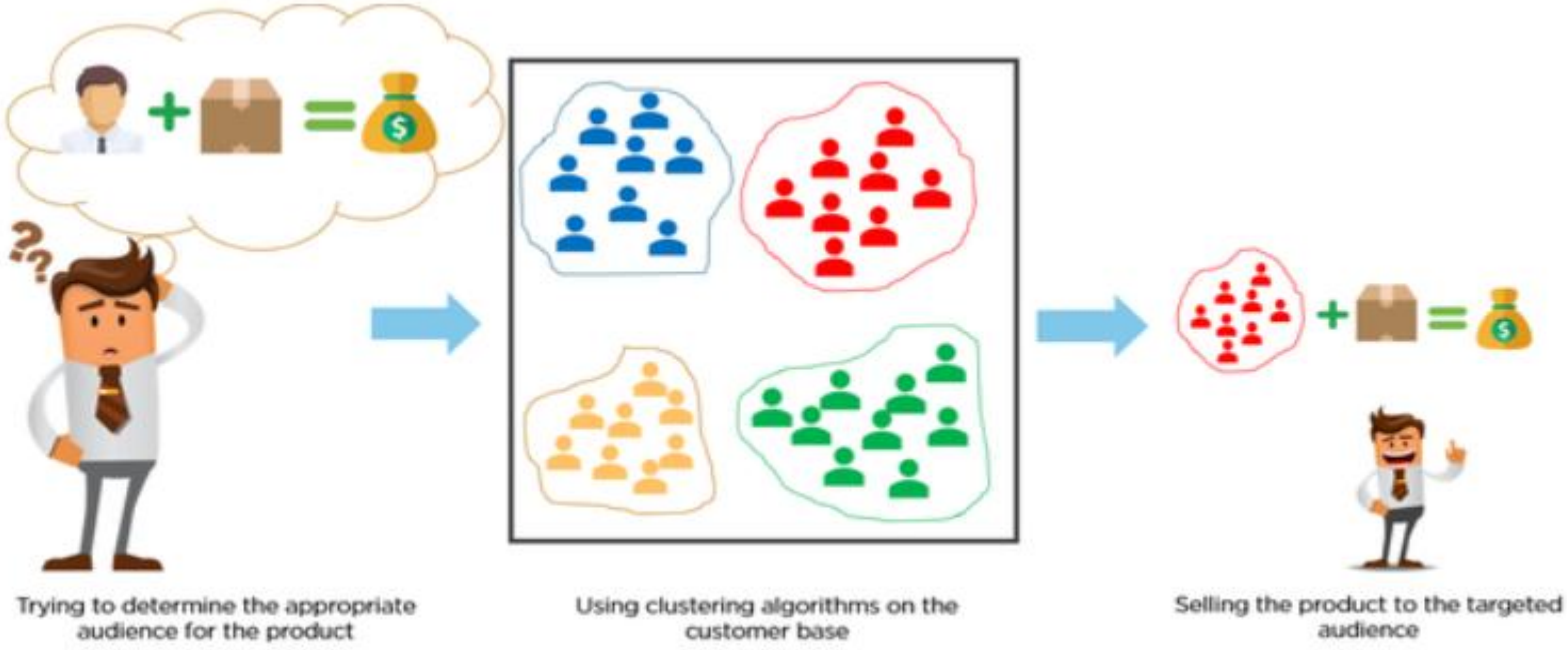
What is Clustering | Cluster analysis

Cluster analysis is a statistical classification technique in which a set of objects or points with similar characteristics are grouped together in clusters.



Clustering

Need of Clustering Algorithms



Trying to determine the appropriate audience for the product

Using Clustering algorithms on the customer base

Selling the products to the targeted audience

Clustering Algorithms

- K-Means 
- Hierarchical Clustering 
- Time-series analysis 

Clustering Algorithms

❑ K-Means

Unsupervised learning algorithm

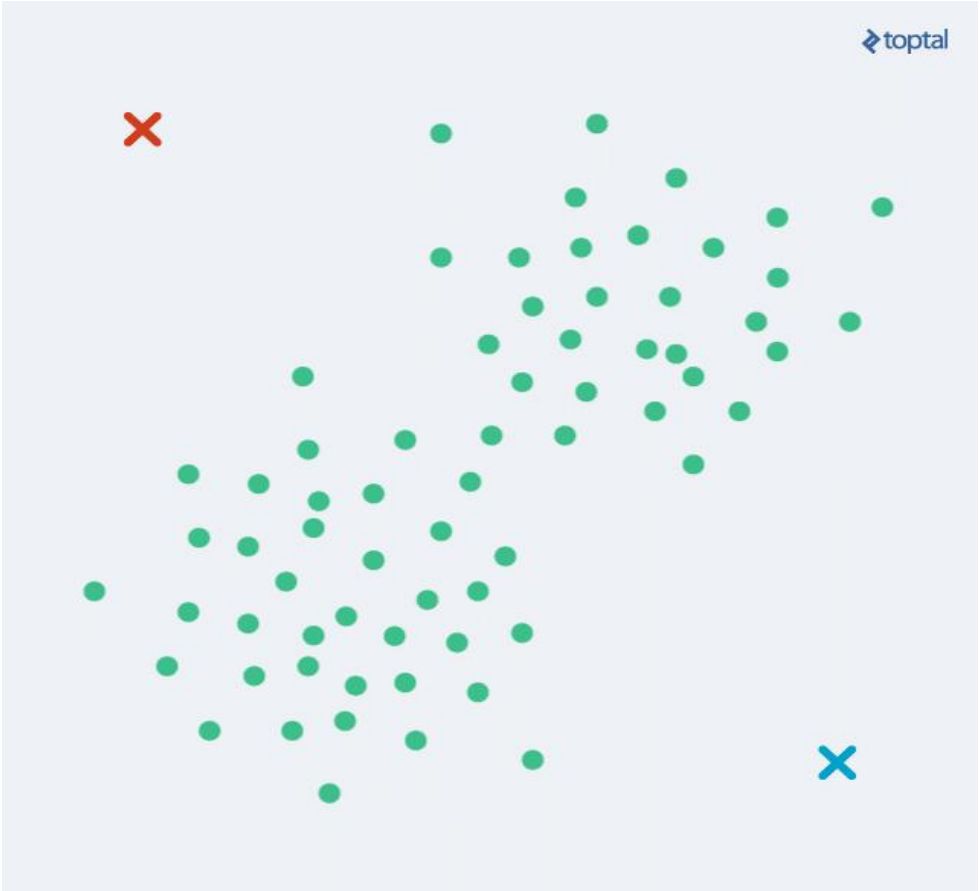
Used to solve the clustering problems

Which groups, unlabeled dataset into different clusters.

Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and
for K=3, there will be three clusters, and so on.

Clustering Algorithms

- K-Means



Clustering Algorithms

❑ K-Means

- It is an iterative algorithm that **divides the unlabeled dataset into k different clusters**
- in such a way that each **dataset belongs only one group that has similar properties.**

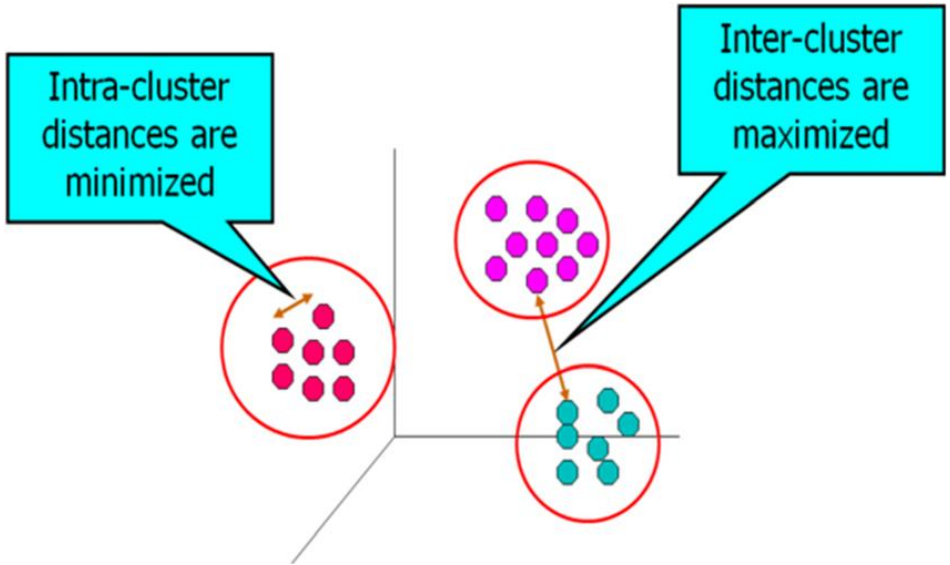
- It allows us to cluster the data into different groups and also provides a convenient way to discover the **categories of groups in the unlabeled dataset on its own without the need for any training.**

Clustering Algorithms

❑ K-Means

- It is a centroid-based algorithm, where each cluster is associated with a centroid.

- The main aim of this algorithm is **to minimize the sum of distances between the data point and their corresponding clusters.**



Clustering Algorithms

❑ K-Means

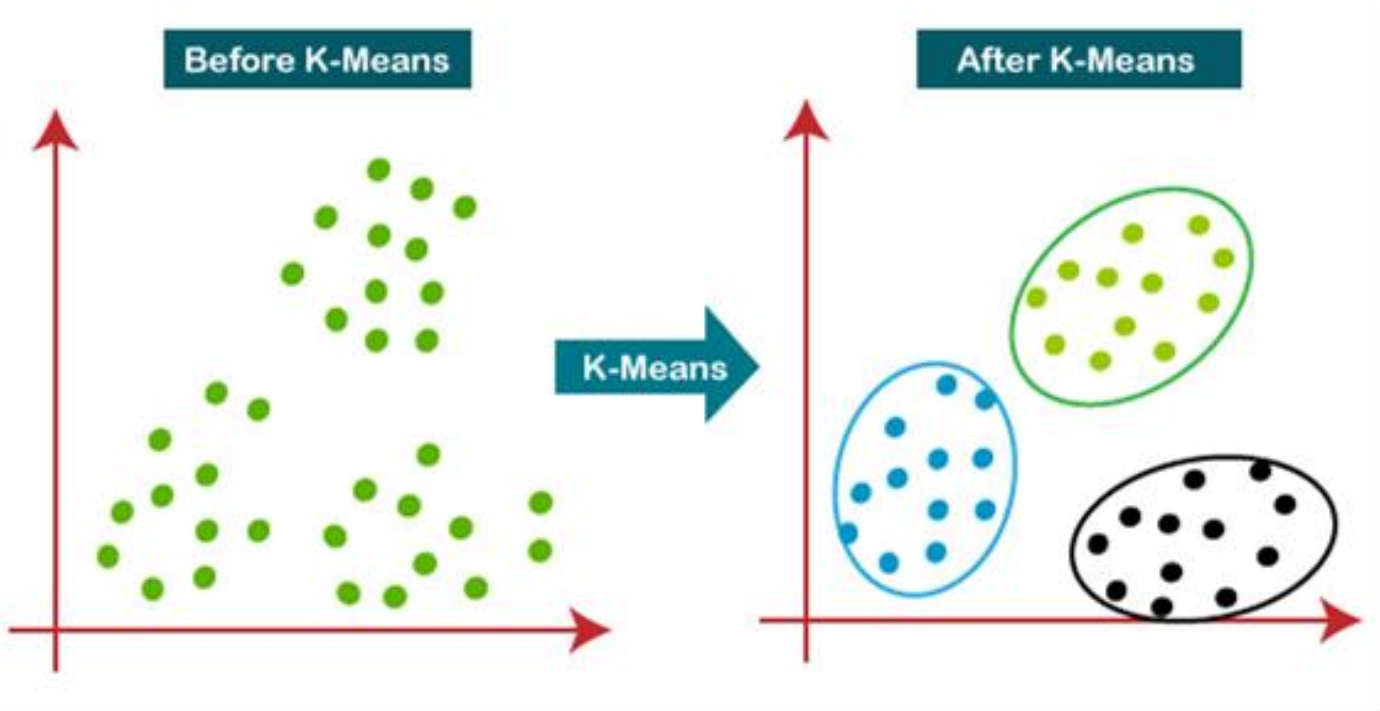
The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.

- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

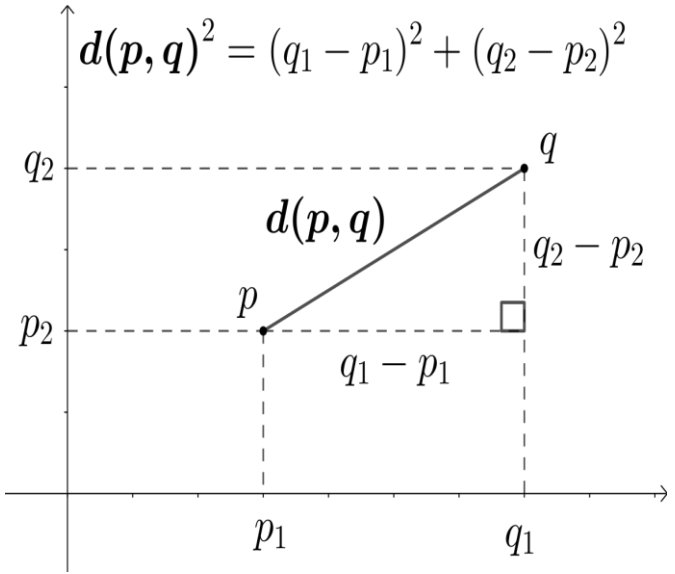
Clustering Algorithms

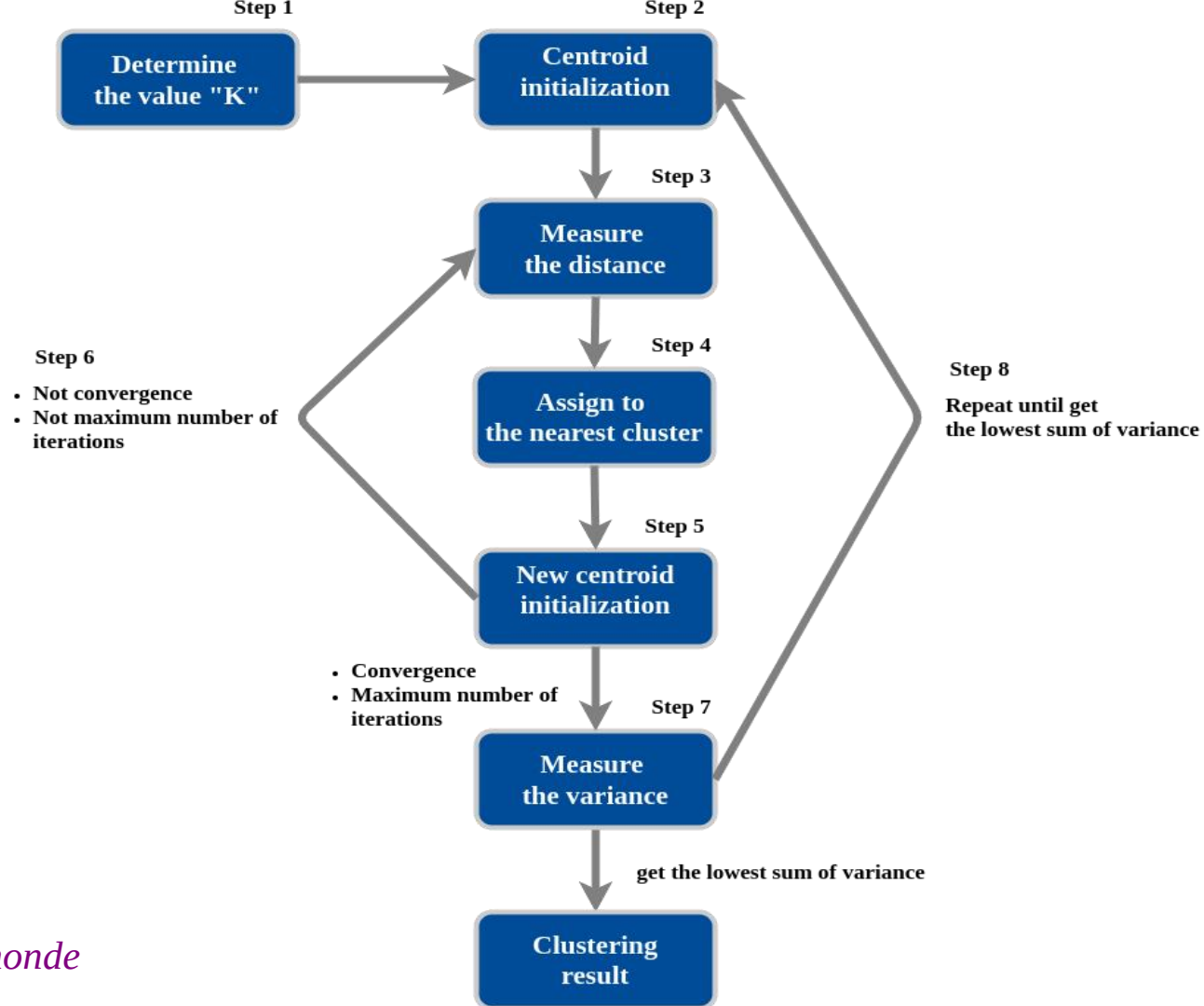
❑ K-Means



❑ K-Means

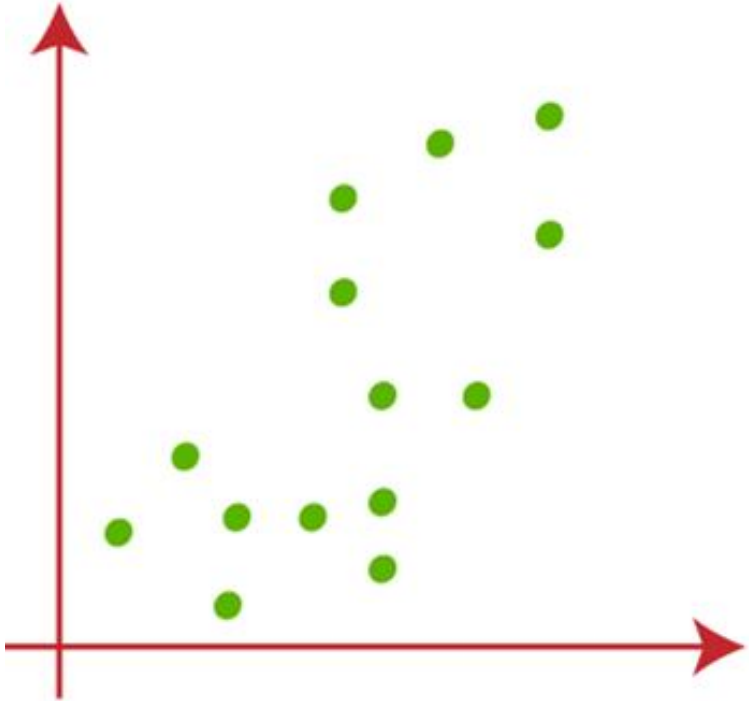
- Basically K-Means runs on distance calculations, which uses “Euclidean Distance” to calculate the distance between two given instances.
- For given instances (X1, Y1) and (X2, Y2), the formula is




Step 1
Determine the value "K"
Step 2
Centroid initialization
Step 3
Measure the distance
Step 4
Assign to the nearest cluster
Step 5
New centroid initialization
Step 6
Not convergence
Not maximum number of iterations
Convergence
Maximum number of iterations
Step 7
Measure the variance
Step 8
Repeat until get the lowest sum of variance
get the lowest sum of variance
Clustering result

Clustering Algorithms

- K-Means



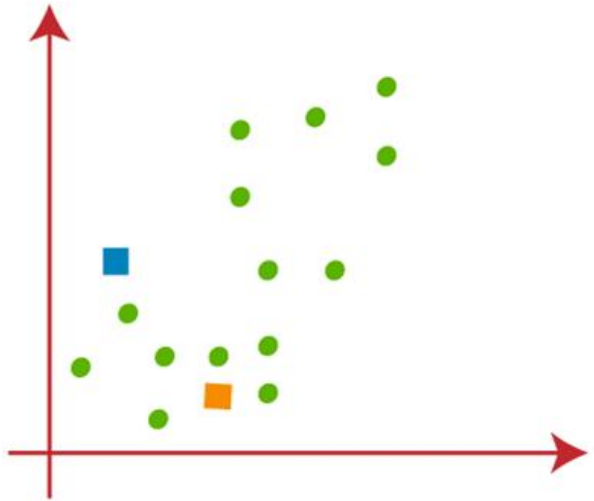
❑ K-Means

How does the K-Means Algorithm Work?

Step 1

Select the number K to decide the number of clusters.

- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters.
- It means here we will try to group these datasets into two different clusters.

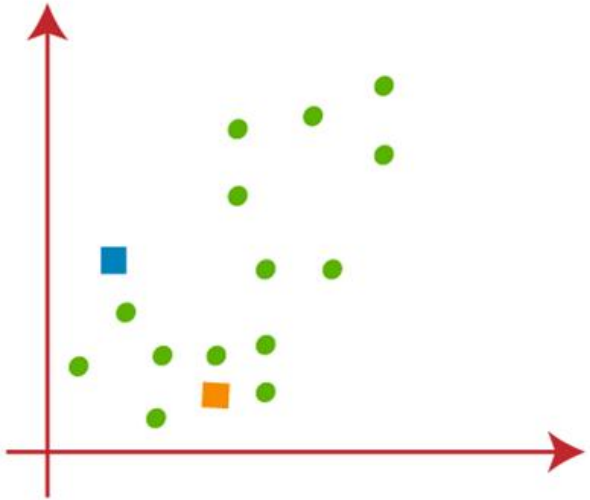


❑ K-Means

How does the K-Means Algorithm Work?

Step 2

Select random K points or centroids.
(It can be other from the input dataset).

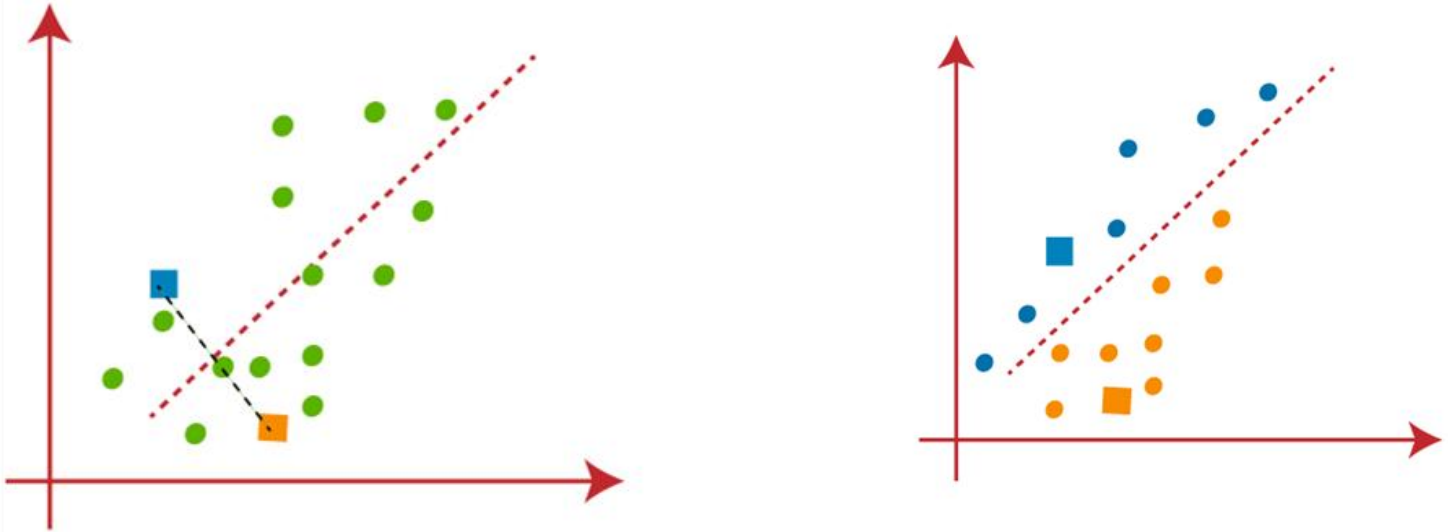


❑ K-Means

How does the K-Means Algorithm Work?

Step 3

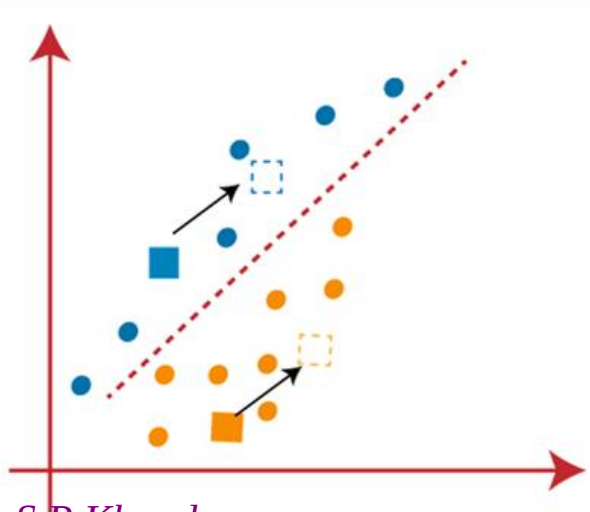
Assign each data point to their closest centroid, which will form the predefined K clusters.

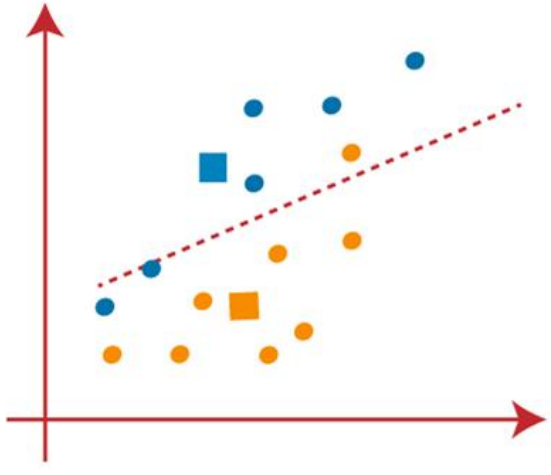


❑ K-Means

How does the K-Means Algorithm Work?

Step 4 Calculate the variance and place a new centroid of each cluster.

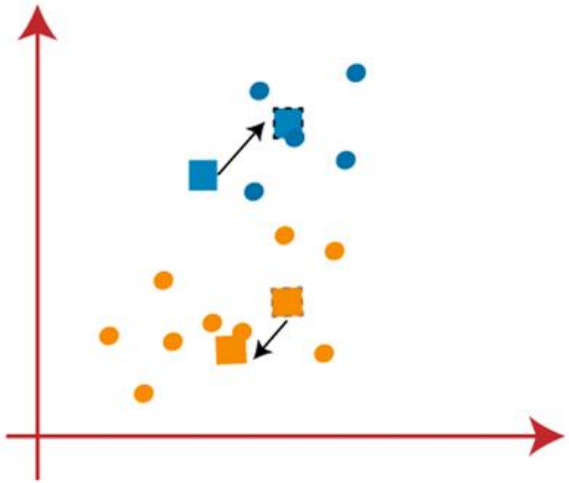


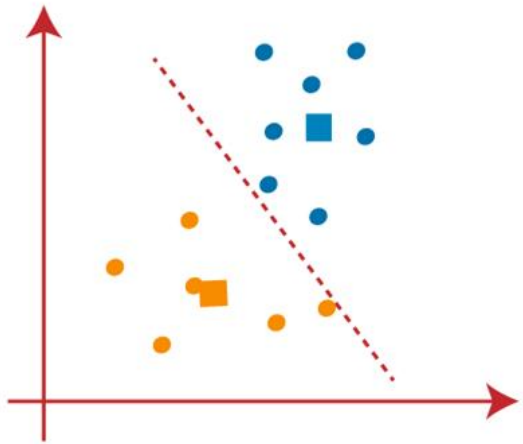


❑ K-Means

How does the K-Means Algorithm Work?

Step 5 Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

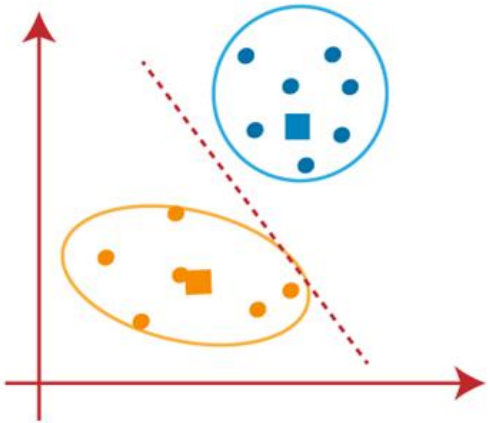




❑ K-Means

How does the K-Means Algorithm Work?

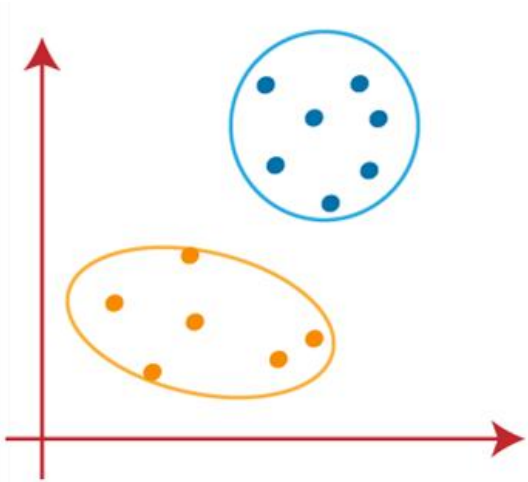
Step 6 If any reassignment occurs, then go to step-4 else go to FINISH.



❑ K-Means

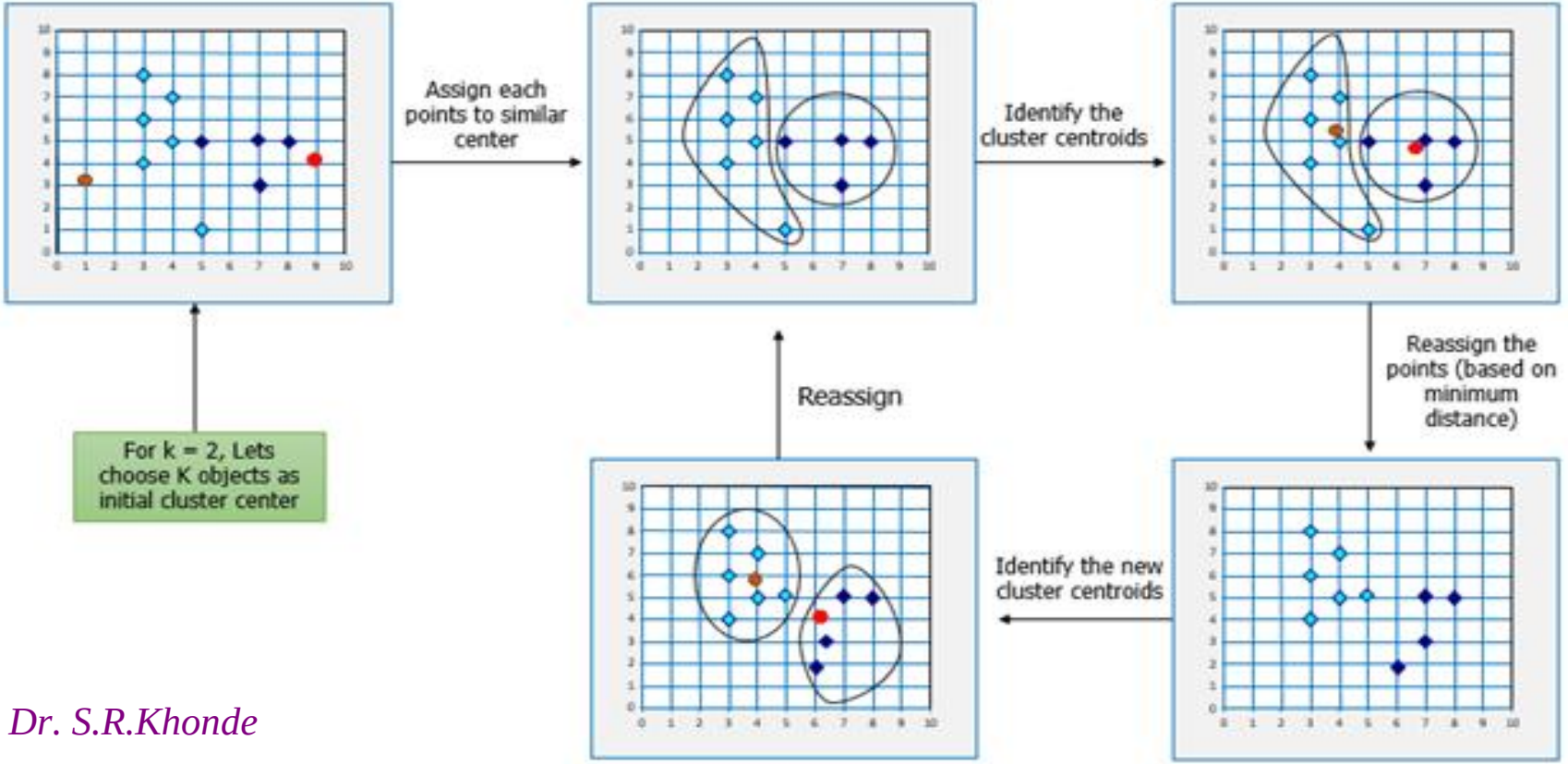
How does the K-Means Algorithm Work?

Step 7 The model is ready.

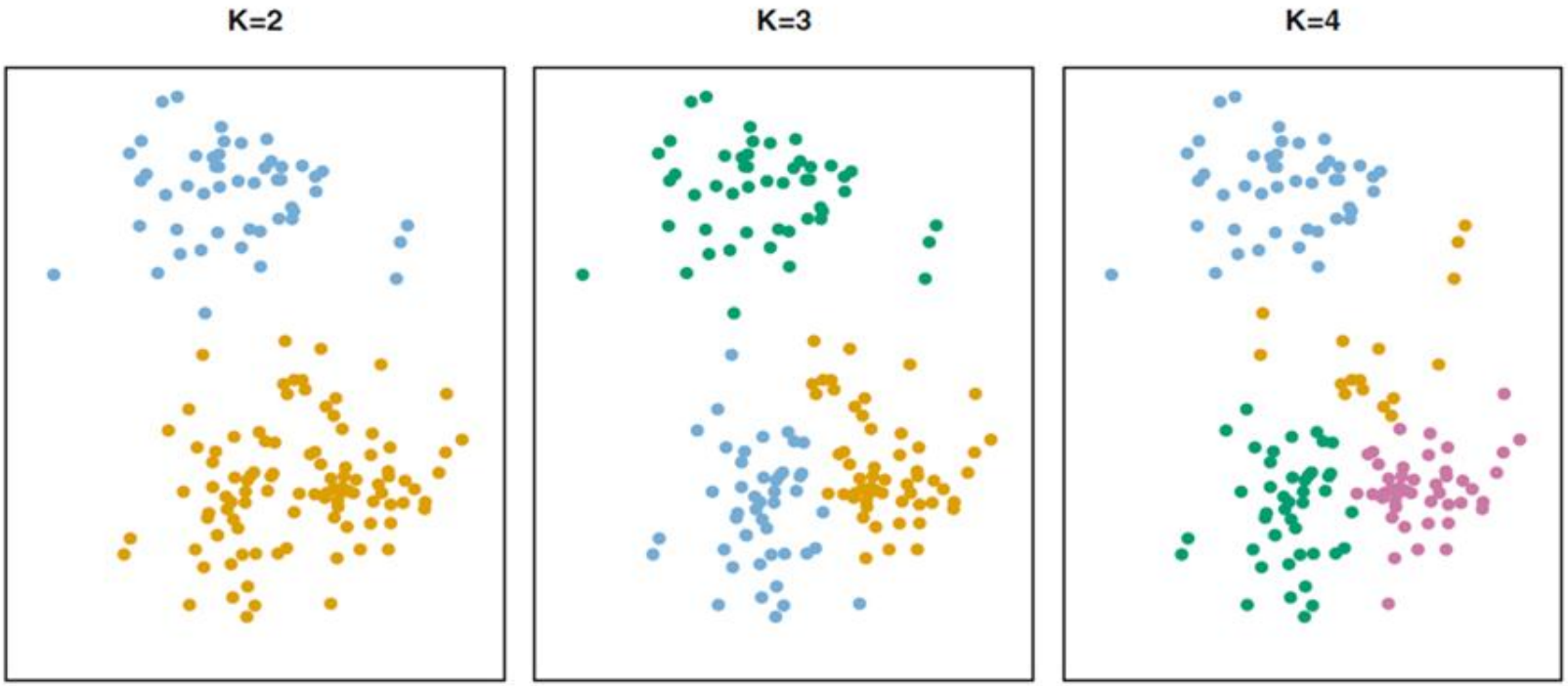


Clustering Algorithms

How does the K-Means Algorithm Work?



Effect for Number of Cluster: K



Clustering Algorithms

❑ Hierarchical Clustering | hierarchical cluster analysis

- unsupervised machine learning algorithm
- used to group the unlabeled datasets into a cluster

❑ Hierarchical Clustering

- we develop the hierarchy of clusters in the form of a tree
- this tree-shaped structure is known as the **dendrogram**.

❑ Hierarchical Clustering

- The hierarchical clustering technique has two approaches:

1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

- The agglomerative hierarchical clustering algorithm is a popular example of HCA.
- To group the datasets into clusters, it follows the bottom-up approach.
- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together.
- It does this until all the clusters are merged into a single cluster that contains all the datasets.

❑ Hierarchical Clustering

Divisive Hierarchical clustering

- This is top Down Strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster.

- It subdivides the clusters into smaller & smaller pieces, until each object from a cluster on its own or until it satisfies certain termination conditions.

Like , a desired number of cluster or the diameter of each cluster is within a certain threshold

❑ Hierarchical Clustering

Why hierarchical clustering?

- we can opt for the hierarchical clustering algorithm
- because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

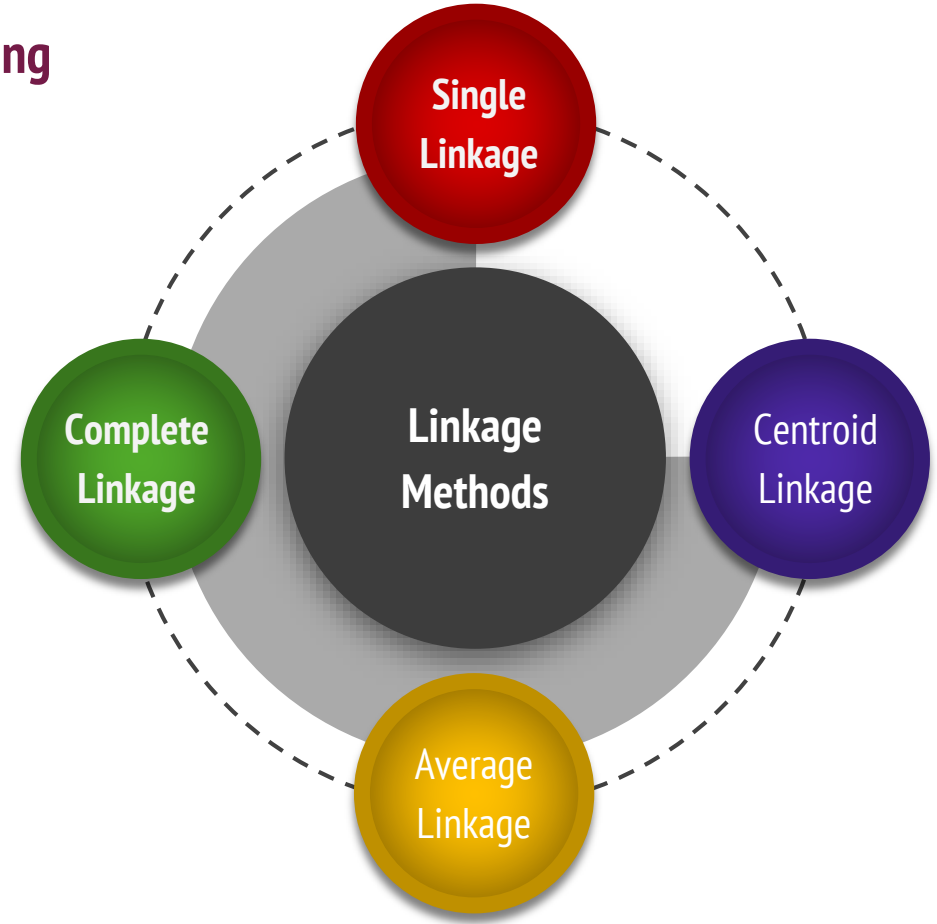
Measure for the distance between two clusters

- the closest distance between the two clusters is crucial for the hierarchical clustering.
- There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering.
- These measures are called **Linkage methods**.

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods



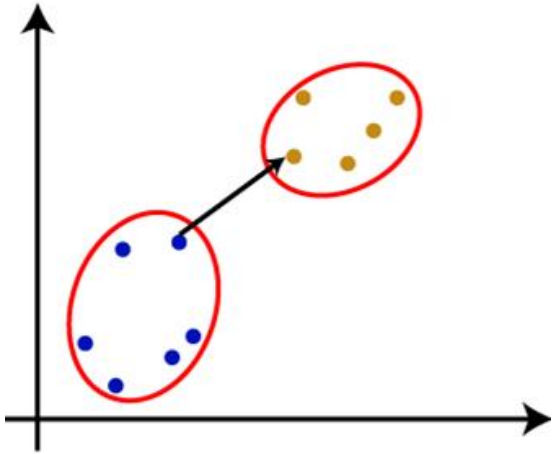
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

Single Linkage

- It is the Shortest Distance between the closest points of the clusters.



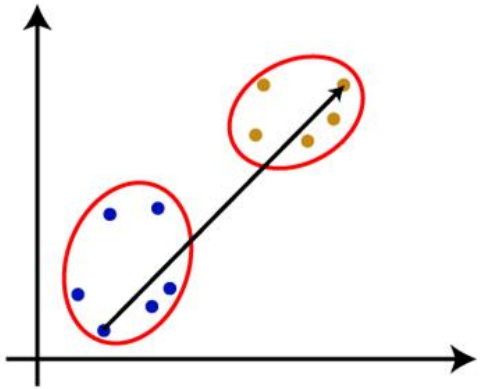
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

Complete Linkage

- It is the farthest distance between the two points of two different clusters.
- It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

- It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters.

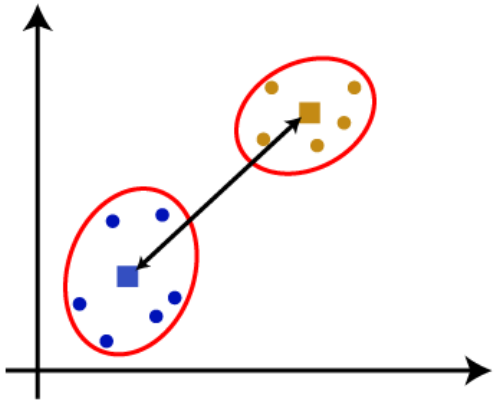
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

Centroid Linkage

- It is the linkage method in which the distance between the centroid of the clusters is calculated.



❑ Hierarchical Clustering Example

How the Agglomerative Hierarchical clustering Work?

- Task is to divide students into different groups.
- Each student evaluated on assignment and based on the given marks.
- There's no fixed target here as to how many groups to have.
- No clear idea about what type of students should be assigned to which group, it cannot be solved as a supervised learning problem.
- So, we will try to apply hierarchical clustering here and segment the students into different groups.

❑ Hierarchical Clustering Example

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 1 Creating a Proximity Matrix

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

$\sqrt{(10-7)^2} = \sqrt{9} = 3$

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

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 2

assign all the points to an individual cluster

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 3

look at the smallest distance in the proximity matrix and merge the points with the smallest distance

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

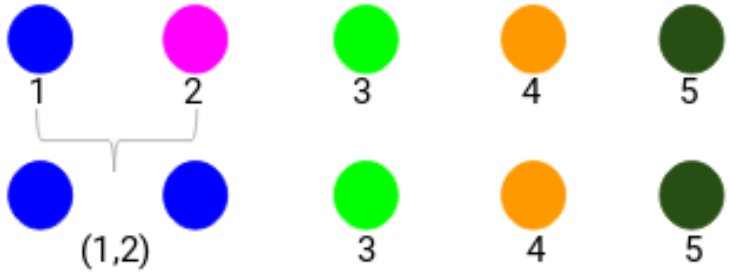
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 3

the smallest distance is 3 and hence we will merge point 1 and 2



❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 3

look at the updated clusters and accordingly update the proximity matrix

As per the selected linkage function update the proximity matrix

Student_ID
(1,2)
3
4
5

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 3

ID	(1,2)	3	4	5
(1,2)	0			
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

To decide the distance between (1,2)->3

- Check the proximity matrix

min((1,3),(2,3))
=min(18,21)
=18

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

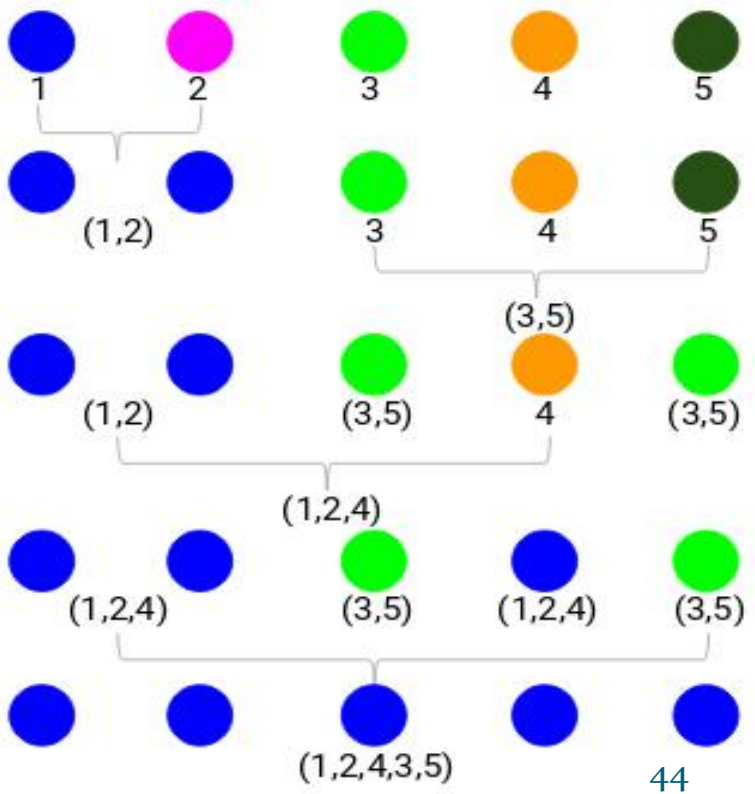
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 4

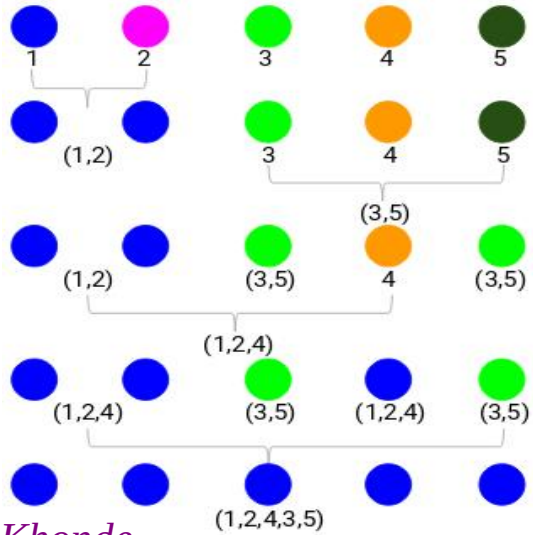
Repeat step 2 until only a single cluster is left.

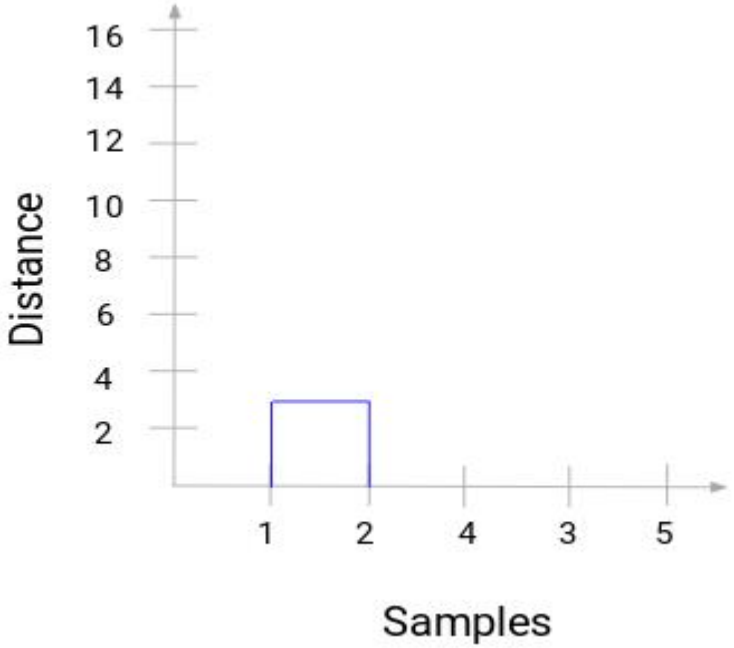


❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Dendrogram Representation

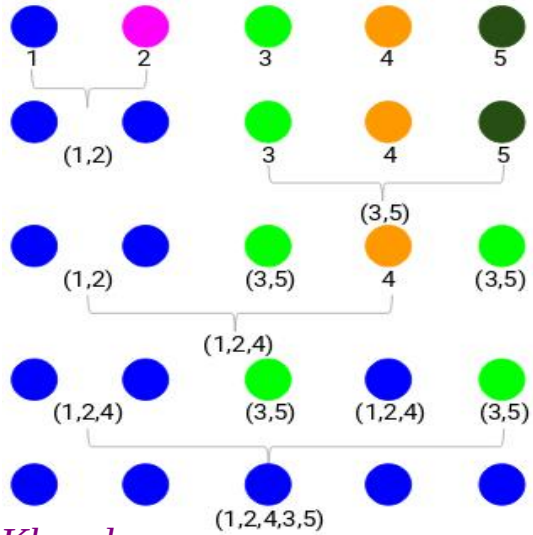


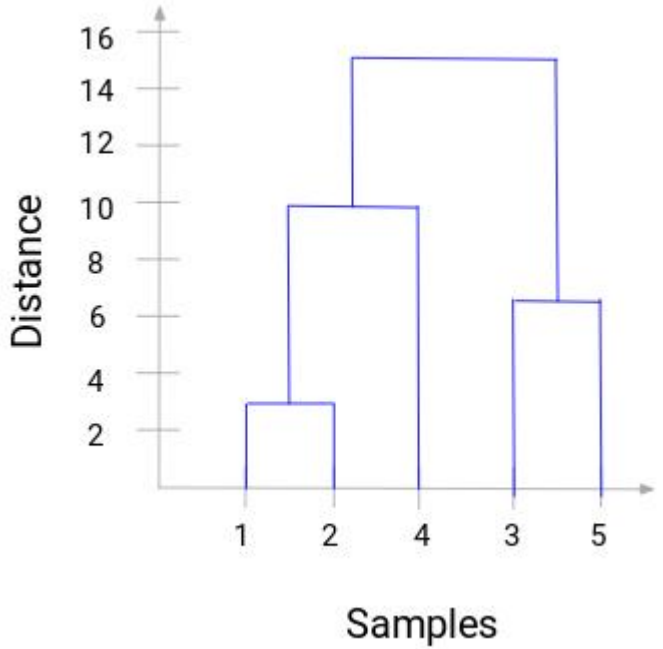


❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Dendrogram Representation



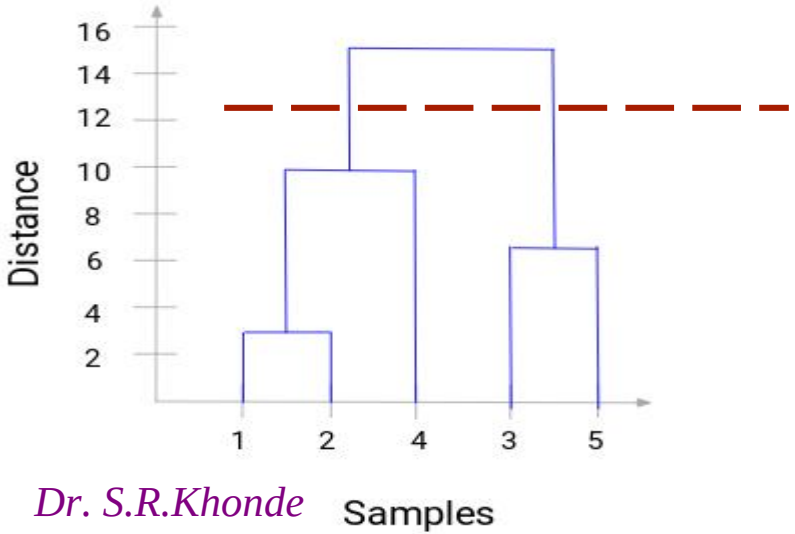


More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Number of Cluster



- Decide threshold
- Consider Threshold =12
- The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold.
- The red line intersects 2 vertical lines
- we will have 2 clusters.
- One cluster will have a sample (1,2,4) and the other will have a sample (3,5)

❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

- Initially each item in its own cluster
- Iteratively cluster are merged together
- Bottom up

Divisive Hierarchical clustering

- Initially each item in one cluster
- Large clusters are successively divided
- Top Down

❑ Time-series analysis

- Time series is a sequence of data points in chronological sequence, most often gathered in regular intervals.
- It can be applied to any variable that changes over time
- It is the way of studying the characteristics of the response variable with respect to time, as the independent variable
- To estimate the target variable in the name of predicting or forecasting, use the time variable as the point of reference

Clustering Algorithms

❑ Time-series analysis

Example

stock price

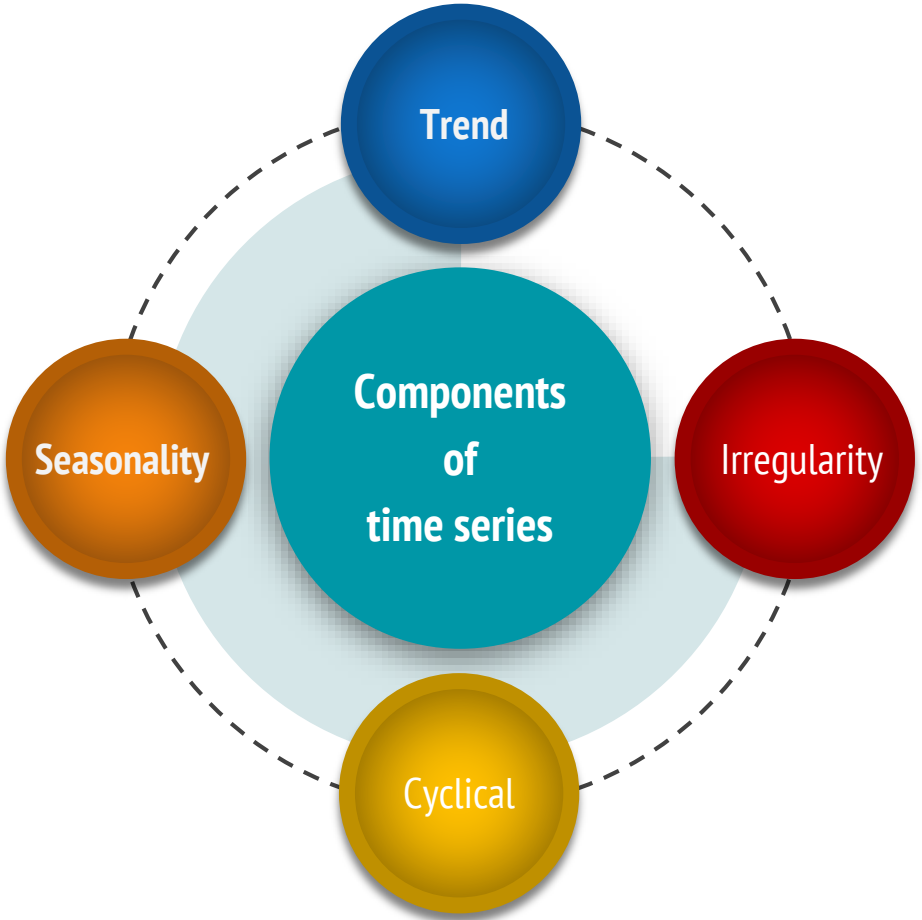
Timestamp	Stock - Price
2015-10-11 09:00:00	100
2015-10-11 10:00:00	110
2015-10-11 11:00:00	105
2015-10-11 12:00:00	90
2015-10-11 13:00:00	120

Basic structure of time series data

Observations are recorded every hour.

- **Time-series analysis**

  Components of time series



❑ Time-series analysis

Components of time series

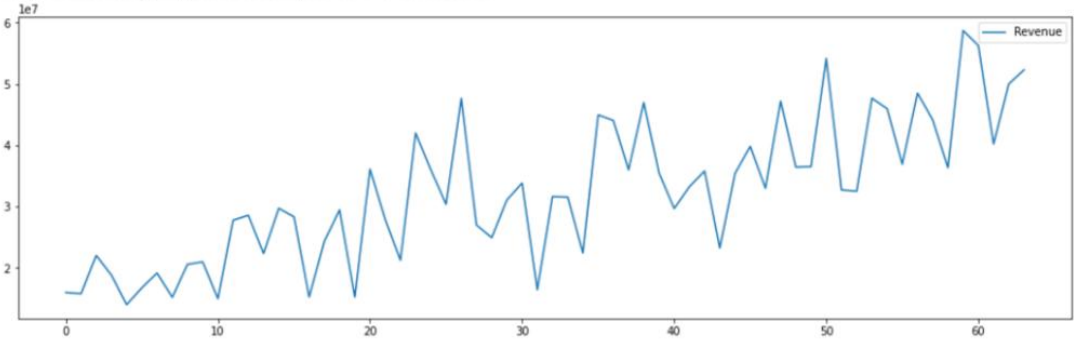
Trend

In which there is no fixed interval and any divergence within the given dataset is a continuous timeline.

The trend would be negative or positive of null trend

```
data.plot.line(x=None,y=None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f23cec9da90>
```



❑ Time-series analysis

Components of time series

In which regular or fixed interval shifts within the dataset in a continuous timeline.

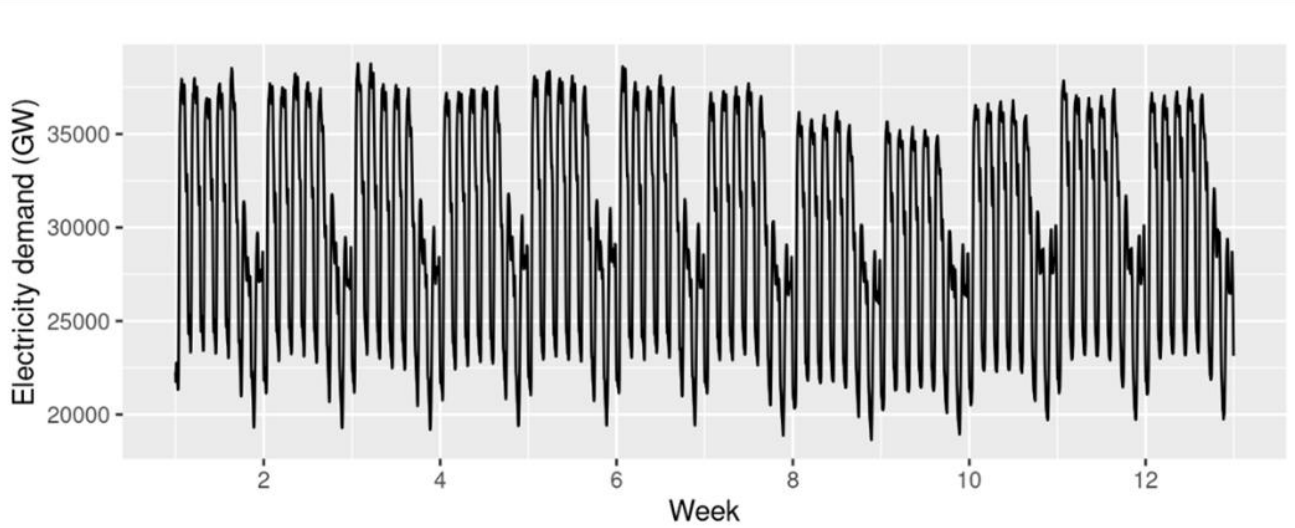
Would be bell curve or saw tooth.

- Identifying seasonality in time series data is important for the development of a useful time series model.

❑ Time-series analysis

Components of time series

Cyclical



In which there is no fixed interval, uncertainty in movement and its pattern

Source

❑ Time-series analysis

Components of time series

Irregularity

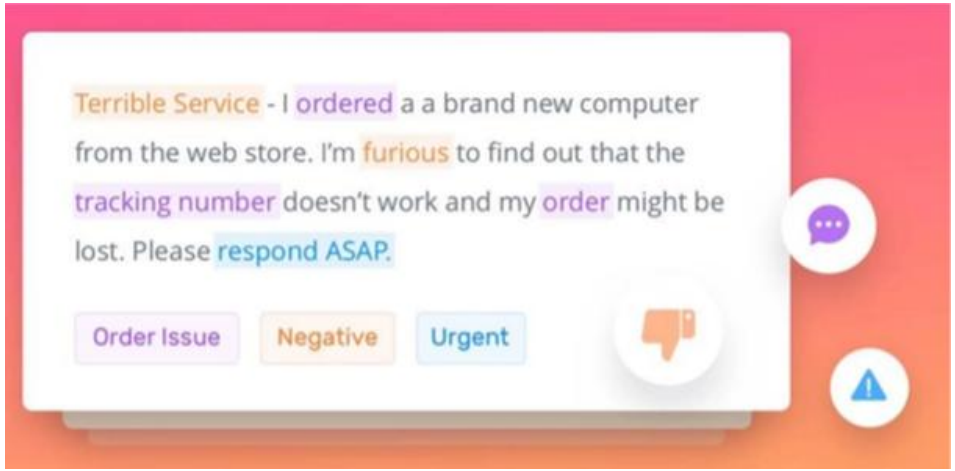
Unexpected situations/events/scenarios and spikes in a short time span

❑ Time-series analysis

Components of time series

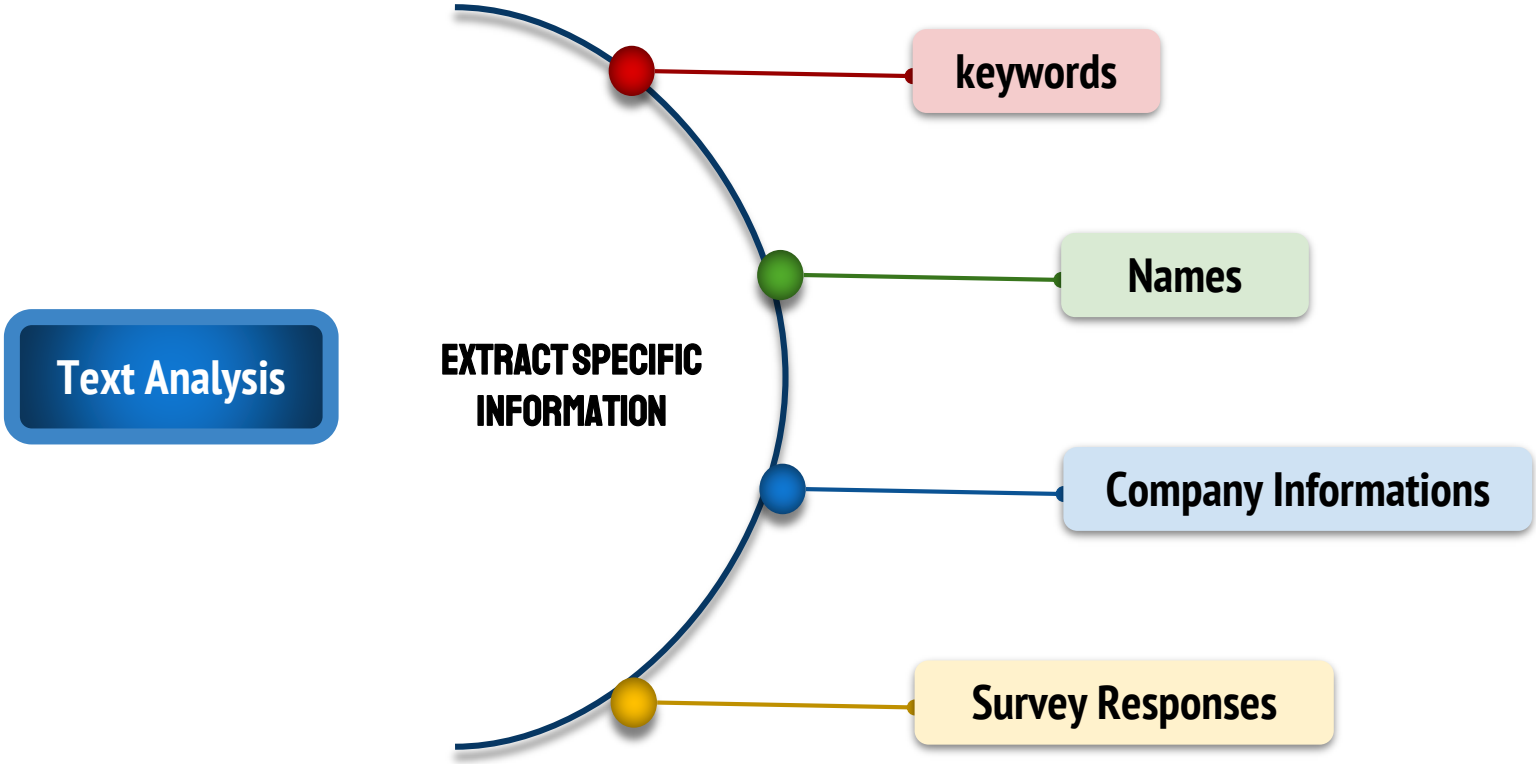
	Trend	Seasonality	Cyclical	Irregularity
Time	Fixed Time Interval	Fixed Time Interval	Not Fixed Time Interval	Not Fixed Time Interval
Duration	Long and Short Term	Short Term	Long and Short Term	Regular/Irregular
Visualization				
Nature - I	Gradual	Swings between Up or Down	Repeating Up and Down	Errored or High Fluctuation
Nature – II	Upward/Down Trend	Pattern repeatable	No fixed period	Short and Not repeatable
Prediction Capability	Predictable	Predictable	Challenging	Challenging

Text Analysis



It is a machine learning technique used to automatically extract valuable insights from unstructured text data.

Text Analysis



Text Analysis

Text Analysis Operations using natural language toolkit

- **Tokenization**
- **Stop Words Removal**
- **Stemming and Lemmatization**
- **POS Tagging**

Text Analysis

Text Analysis Operations using natural language toolkit

Tokenization

- the first step in text analytics
- The process of breaking down a text paragraph into smaller chunks such as words or sentences is called Tokenization.
- Token is a single entity that is the building blocks for a sentence or paragraph.

Text Analysis

Text Analysis Operations using natural language toolkit

Tokenization

Sentence Tokenization

- split a paragraph into **list of sentences** using **sent_tokenize()** method

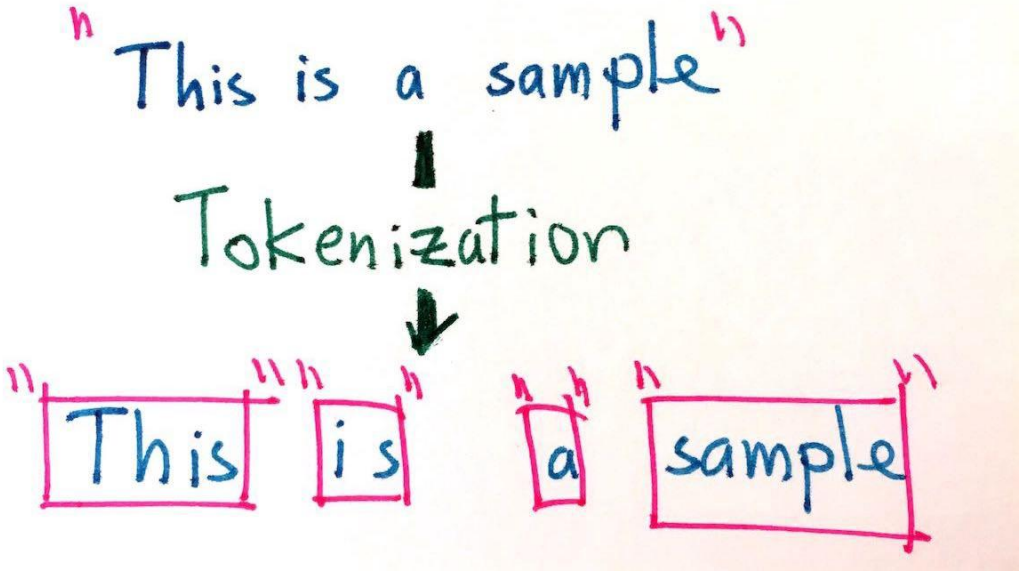
Word Tokenization

- split a sentence into **list of words** using **word_tokenize()** method

Text Analysis

Text Analysis Operations using natural language toolkit

Tokenization



Text Analysis Operations using natural language toolkit

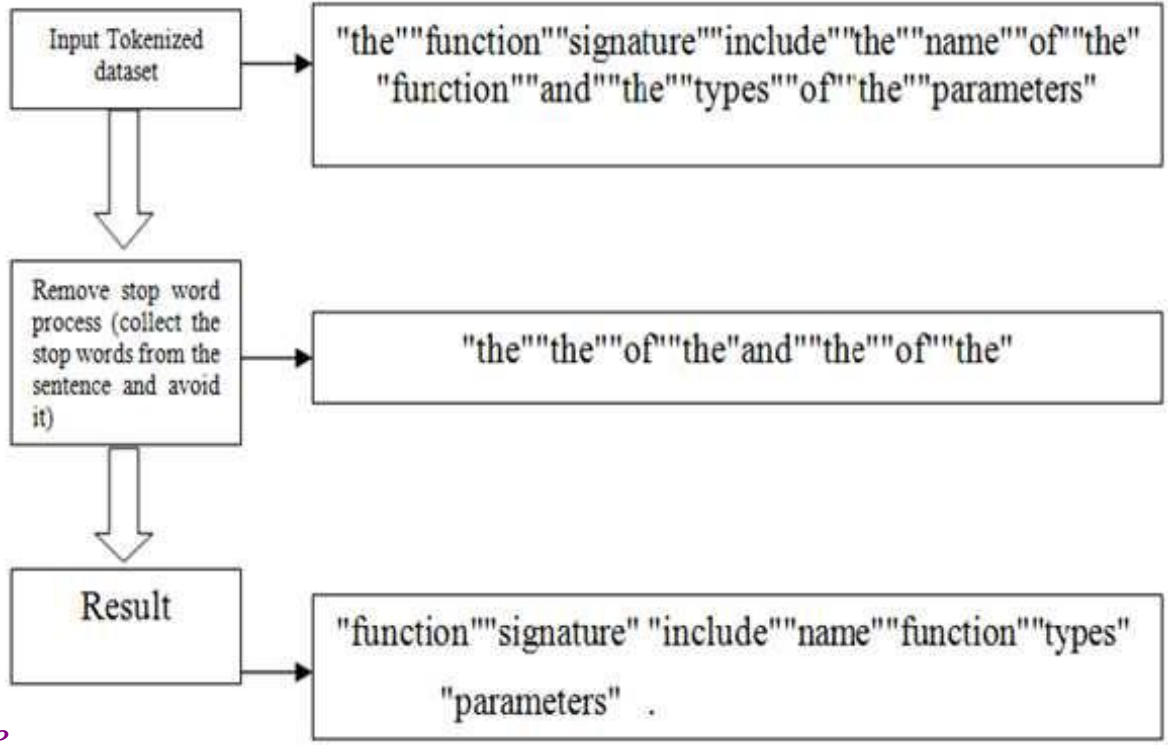
Stop Words Removal

- Stopwords considered as noise in the text.
- Text may contain stop words such as is, am, are, this, a, an, the, etc.

Text Analysis

Text Analysis Operations using natural language toolkit

Stop Words Removal



Text Analysis Operations using natural language toolkit

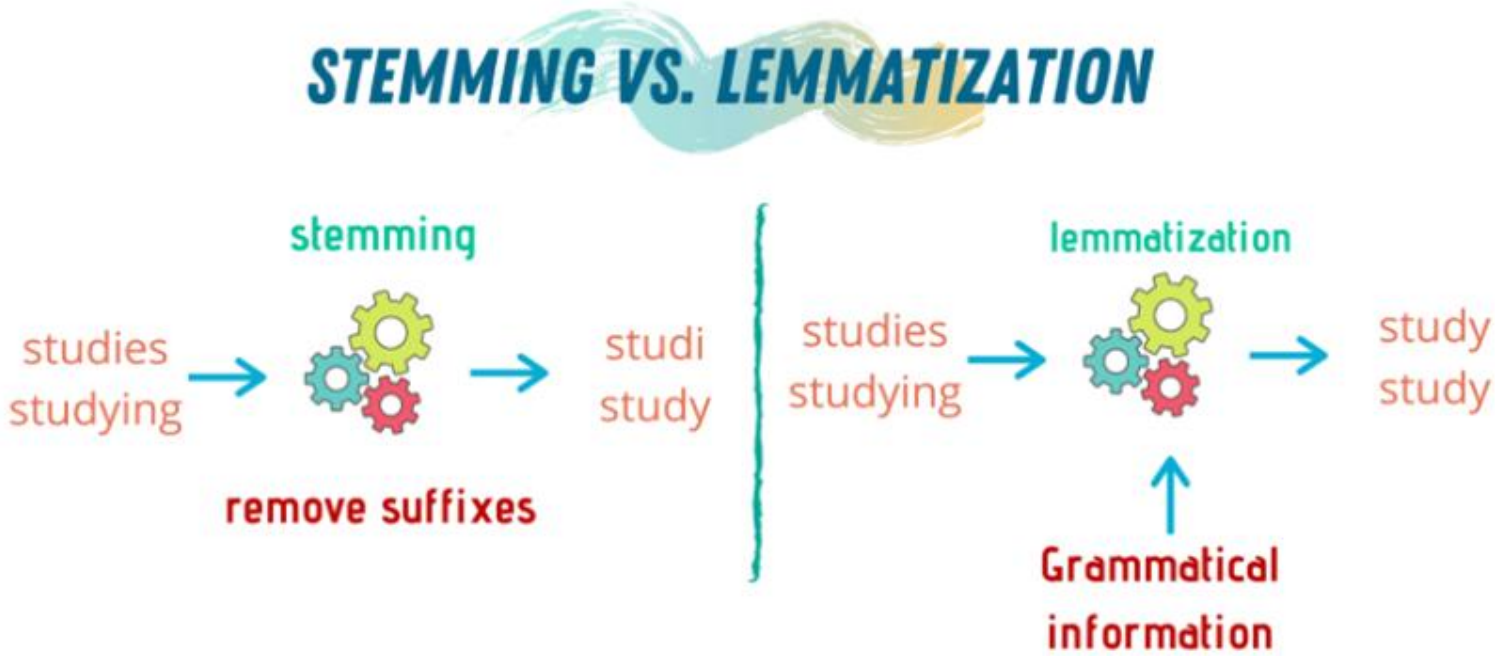
Stemming and Lemmatization

- **Stemming** is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy.

- **Lemmatization** in NLTK (Natural Lang. Toolkit) is the algorithmic process of finding the lemma of a word depending on its meaning and context.

Text Analysis Operations using natural language toolkit

Stemming and Lemmatization



Text Analysis Operations using natural language toolkit

POS Tagging

- **POS (Parts of Speech)** tell us about grammatical information of words of the sentence by assigning specific token as tag to each words.

Part of Speech	Tag
Noun	n
Verb	v
Adjective	a
Adverb	r

Text Analysis Model using TF-IDF

- Term frequency–inverse document frequency(TFIDF)

- is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

Term Frequency

- It is a measure of the frequency of a word (w) in a document (d).

- TF is defined as the ratio of a word's occurrence in a document to the total number of words in a document.

Term Frequency

Formula

$$TF(w,d) = \frac{occurences\ of\ w\ in\ document\ d}{total\ number\ of\ words\ in\ document\ d}$$

Term Frequency

Example

Documents	Text	Total number of words in a document
A	Jupiter is the largest planet	5
B	Mars is the fourth planet from the sun	8

Inverse Document Frequency

- It is the measure of the importance of a word.
- Term frequency **(TF)** does not consider the importance of words.
- Some words such as' of', 'and', etc. can be most frequently present but are of little significance.
- **IDF** provides weightage to each word based on its frequency in the corpus D.

Inverse Document Frequency

Formula

$$IDF(w, D) = \ln\left(\frac{Total\ number\ of\ documents\ (N)\ in\ corpus\ D}{number\ of\ documents\ containing\ w}\right)$$

Inverse Document Frequency

Example

In our example, since we have two documents in the corpus, N=2.

Documents	Text	Total number of words in a document
A	Jupiter is the largest planet	5
B	Mars is the fourth planet from the sun	8

Words	TF (for A)	TF (for B)	IDF
Jupiter	1/5	0	ln(2/1) = 0.69
Is	1/5	1/8	ln(2/2) = 0
The	1/5	2/8	ln(2/2) = 0
largest	1/5	0	ln(2/1) = 0.69
Planet	1/5	1/8	ln(2/2) = 0
Mars	0	1/8	ln(2/1) = 0.69
Fourth	0	1/8	ln(2/1) = 0.69
From	0	1/8	ln(2/1) = 0.69
Sun	0	1/8	ln(2/1) = 0.69

Term Frequency — Inverse Document Frequency (TFIDF)

- It is the product of **TF** and **IDF**.
- **TFIDF** gives more weightage to the word that is rare in the corpus (all the documents).
- **TFIDF** provides more importance to the word that is more frequent in the document.

Term Frequency – Inverse Document Frequency (TFIDF)

Formula

$$TFIDF\ (w, d, D) = TF(w, d) * IDF(w, D)$$

Term Frequency – Inverse Document Frequency (TFIDF)

Example

Documents	Text	Total number of words in a document
A	Jupiter is the largest planet	5
B	Mars is the fourth planet from the sun	8

Words	TF (for A)	TF (for B)	IDF	TFIDF (A)	TFIDF (B)
Jupiter	1/5	0	ln(2/1) = 0.69	0.138	0
Is	1/5	1/8	ln(2/2) = 0	0	0
The	1/5	2/8	ln(2/2) = 0	0	0
largest	1/5	0	ln(2/1) = 0.69	0.138	0
Planet	1/5	1/8	ln(2/2) = 0	0.138	0
Mars	0	1/8	ln(2/1) = 0.69	0	0.086
Fourth	0	1/8	ln(2/1) = 0.69	0	0.086
From	0	1/8	ln(2/1) = 0.69	0	0.086
Sun	0	1/8	ln(2/1) = 0.69	0	0.086

Term Frequency — Inverse Document Frequency (TFIDF)

Disadvantage of **TF IDF**

- It is unable to capture the semantics.

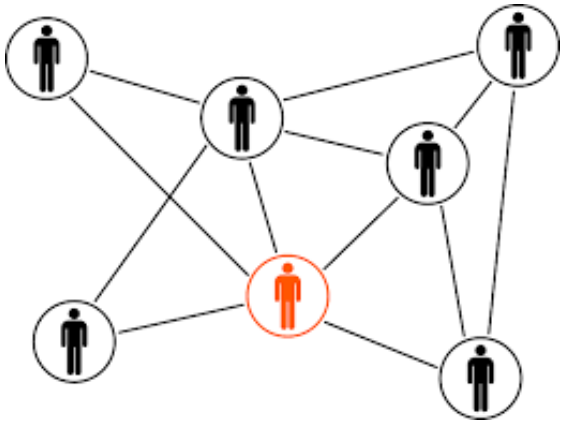
Introduction to social network analysis

Social network analysis (SNA)

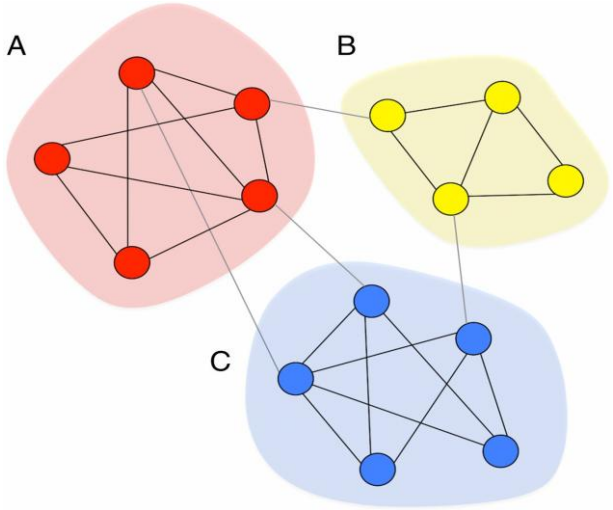
- is the process of investigating social structures in terms of nodes and edges that connect them through the use of networks and graph theory.

Introduction to social network analysis

Link prediction

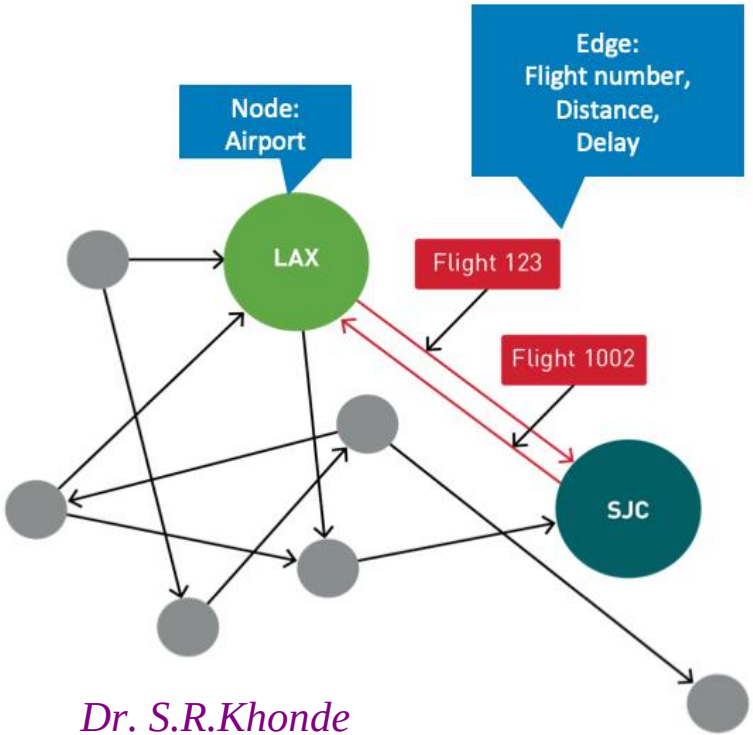


Classification



Graph Theory for social network analysis

Graph:



A graph is made up of **vertices(also called nodes)** that are c**onnected by edges(also called links or relationships**).

Graph Theory for social network analysis

Edges:

Here are **three different edges relationships:**

- **Symmetric and Asymmetric (Directionality)**
- Binary and Valued (Weight)

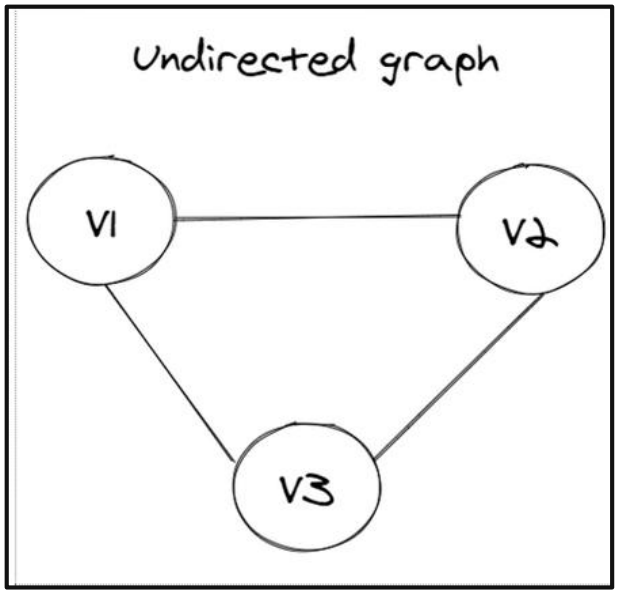
- The relationship **"working together" is a symmetric** relationship
- If A is related to B, B is also related to A.

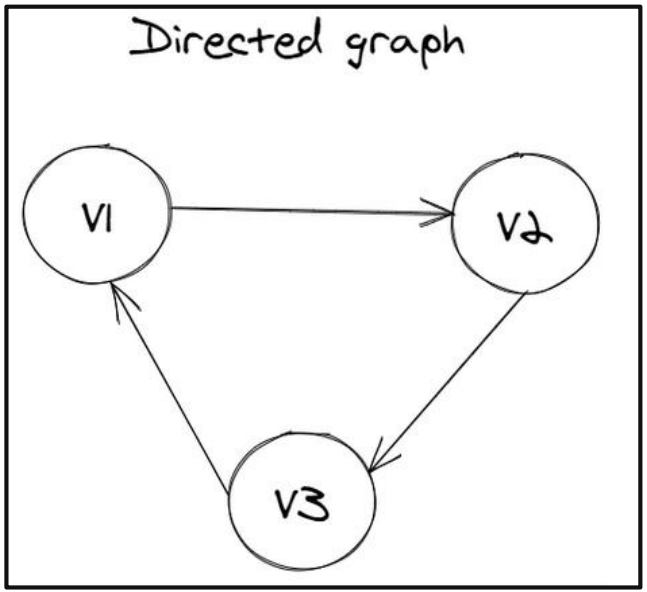
Graph Theory for social network analysis

Edges:

Here are **three different edges relationships:**

- **Symmetric and Asymmetric (Directionality)**
- Binary and Valued (Weight)





Graph Theory for social network analysis

Edges:

Here are **three different edges relationships:**

- **Symmetric and Asymmetric (Directionality)**
- Binary and Valued (Weight)

- The relationship between n**odes is 'child of'**, then the relationship is asymmetric.
- This is the case if someone **follows someone else on Twitter**.
- If A is the child of B, then B is not a child of A. Such a network where the relationship is asymmetric

Graph Theory for social network analysis

Edges:

Here are **three different edges relationships:**

- **Symmetric and Asymmetric (Directionality)**
- **Binary and Valued (Weight)**

- Relationships can be binary or valued.
- "Max follows Alex on Twitter" is a binary relationship
- "Max retweeted 4 tweets from Alex" is valued.
- In the Twitter world, such relationships are easily quantified
- in the "softer" social world it's very hard to determine and quantify the quality of an interpersonal relationship.

Graph Theory for social network analysis

Density

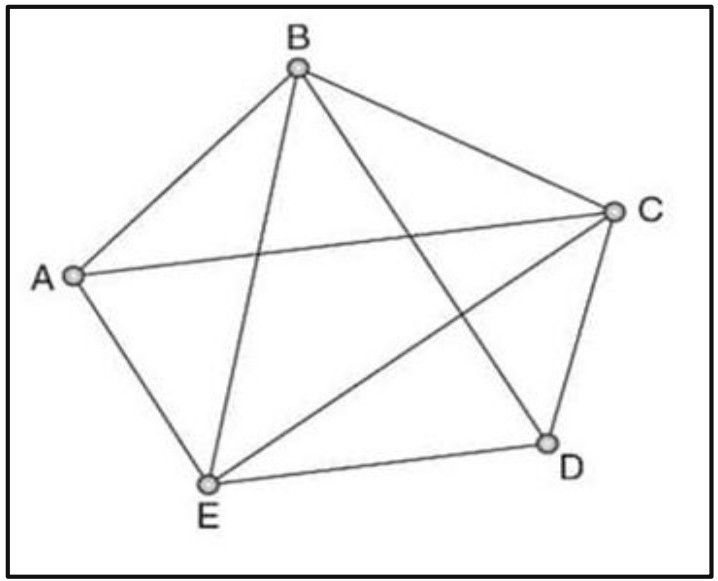
- The relation between **the number of existing connections in a network and all possible connections**

$$\text{Density} = \frac{\text{Actual Connections}}{\text{Potential Connections}}$$

$$\text{Potential Connections} = \frac{n*(n-1)}{2}$$

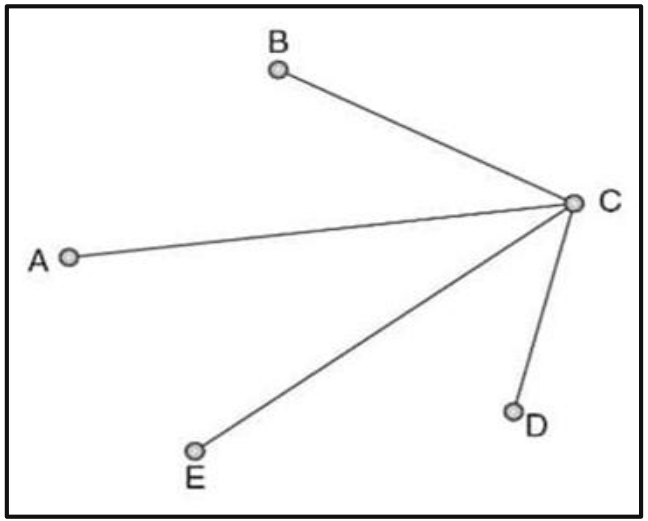
and n = number of nodes in the network

Graph Theory for social network analysis | Density



- 5 Nodes
- Potential edges= 5(5-1)/2 = 5*4 =20/2=10
- Actual Edges= 9
- Density= 9/10= 90%
- Hence **it is a high-density network.**

Graph Theory for social network analysis | Density

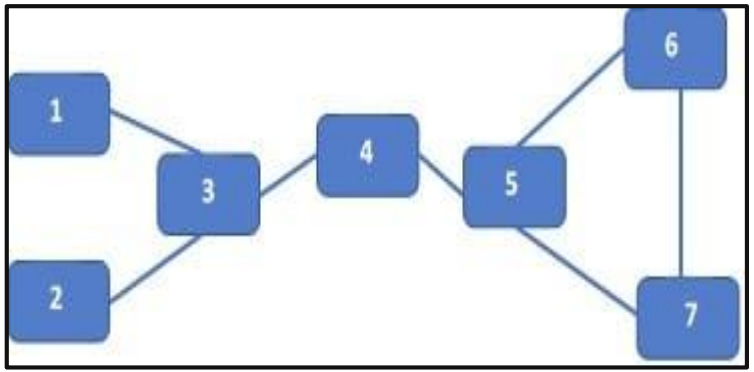


- 5 Nodes
- Potential edges= 5(5-1)/2 = 5*4 /2 =20/2=10
- Actual Edges= 5
- Density= 5/10= 50%
- Hence **it is a low-density network.**

Graph Theory for social network analysis

Centrality Measures

Degree Cardinality Measures the number of direct ties to a node; this will indicate the most connected node in the group.



Node	Score	Standardized Score
1	1	1/6
2	1	1/6
3	3	3/6 = ½
4	2	2/6 = 1/3
5	3	3/6 = ½
6	2	2/6 = 1/3
7	2	2/6 = 1/3

The standardized score is calculated by dividing the score by (n-1), where n is the number of nodes in the network.

Nodes 3 and 5 have a high degree centrality of 0.5, i.e., they are the most well-connected nodes in the network.

Graph Theory for social network analysis

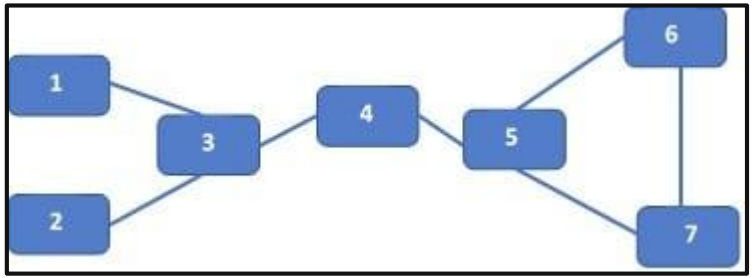
Closeness Cardinality

- **Closeness Cardinality** Closeness measures how close a node is to the rest of the network. It is the ability of the node to reach the other nodes in the network.
- It is calculated as the inverse of the sum of the distance between a node and **other nodes in the network.**

Graph Theory for social network analysis

Centrality Measures

- **Closeness Cardinality**



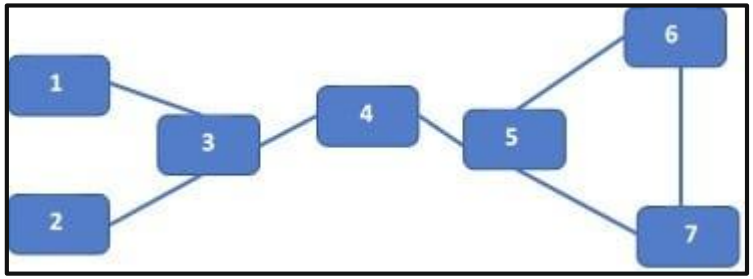
- Hence the Closeness score for node 1 will be 1/16.
- The standardized score is calculated by multiplying the score by (n-1).

Destination Nodes ->	Node 2	Node3	Node 4	Node 5	Node 6	Node 7	Total Distance
Distance from Node 1 to Destination Node	2	1	2	3	4	4	16

Graph Theory for social network analysis

Centrality Measures

- **Closeness Cardinality**



Node	Score	Standardized Score
1	1/16	6/16 = 3/8
2	1/16	6/16 = 3/8
3	1/11	6/11
4	1/10	6/10 = 3/5
5	1/11	6/11
6	1/15	6/15 = 2/5
7	1/15	6/15 = 2/5

Node 4 is the closest/central node in the network with the highest closeness score of 0.6.

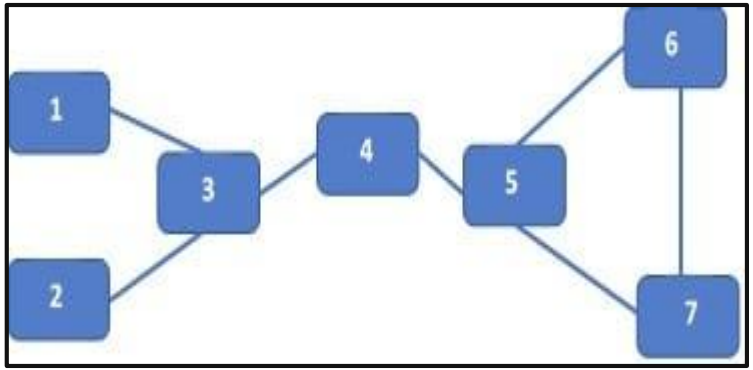
Graph Theory for social network analysis

Centrality Measures

- **Betweenness Centrality** is a measure of how often a node appears in the shortest path connecting two other nodes.

Graph Theory for social network analysis

Centrality Measures



Node pairs	Path value of Node 5
1,5	½
1,6	1/3
1,7	1/3
2,5	½
2,6	1/3
2,7	1/3
3,5	1
3,6	½
3,7	½
Total Score for Node 5	13/3

Let us take node 5 in *Figure 4*. Node 5 occurs in 9 shortest paths between a pair of nodes

Nodes with high betweenness centrality are critical in controlling and maintaining flow in the network; hence these are critical nodes in the network

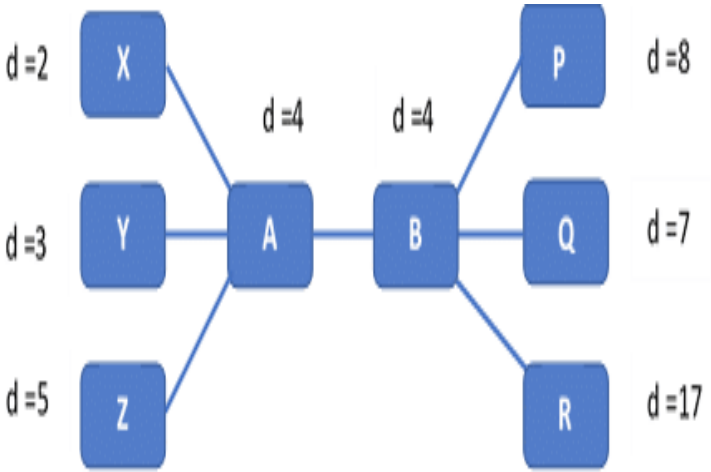
Graph Theory for social network analysis

Centrality Measures

- **Eigen Centrality** is a relative measure of **the importance of the node in the network.**
- Each node is assigned a value or score depending upon **the number of other prominent/ high scoring nodes it is connected to.**

Graph Theory for social network analysis

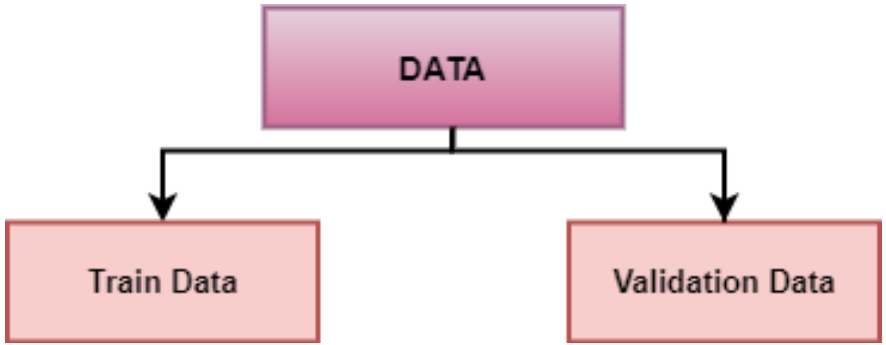
Why Centrality Measures ?



- Here **'d' represents the degree centrality score.**
- Nodes **A and B are connected to 4 nodes each,** and hence both **have a degree centrality score of 4.**
- But when we look at their neighbors, **we can see that node B is connected to nodes with a high degree**.
- Hence, node **B can be preferred over node A when we have to choose based on connectivity**.

Why Cross Validation is important

- Data needs to split into:



- **Training data:** Used for model development
- **Validation data:** Used for validating the performance of the same model
- In simple terms cross-validation allows us to utilize our data even better.

Cross Validation

- **Cross-Validation** also referred to **as sampling technique** which is an essential element of a data science project.

- It is a resampling procedure used to evaluate machine learning models and access how the model will perform for an independent test dataset.

Cross Validation

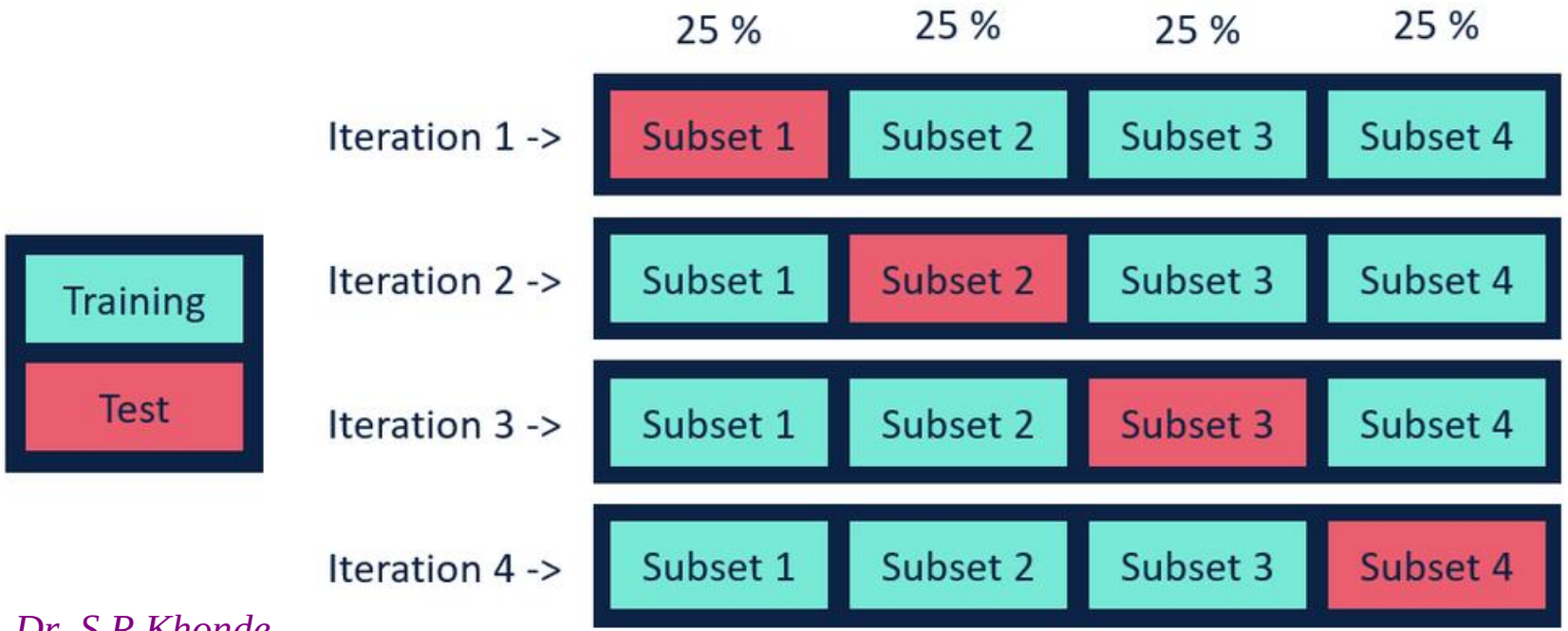
8 different cross-validation techniques

1. **Leave p out cross-validation**
2. **Leave one out cross-validation**
3. **Holdout cross-validation**
4. **Repeated random subsampling validation**
5. **k-fold cross-validation**
6. **Stratified k-fold cross-validation**
7. **Time Series cross-validation**
8. **Nested cross-validation**

Model Evaluation & Selection

Cross Validation

Hold Out Cross Validation

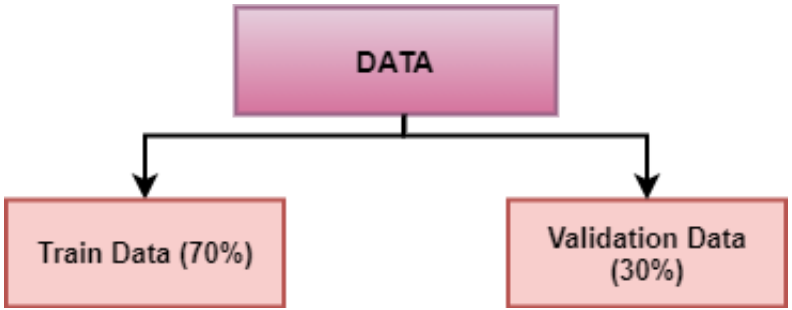


Model Evaluation & Selection

Cross Validation

Hold Out Cross Validation

- The holdout technique is an exhaustive cross-validation method, that randomly splits the dataset into train and test data depending on data analysis.



70:30 split of Data into training and validation data respectively

Model Evaluation & Selection

Cross Validation

Hold Out Cross Validation

- In the case of holdout cross-validation, the dataset is randomly split into training and validation data.
- Generally, the split of training data is more than test data.
- The training data is used to induce the model and validation data is evaluates the performance of the model.
- The more data is used to train the model, the better the model is.

Model Evaluation & Selection

Cross Validation

Hold Out Cross Validation

- For the holdout cross-validation method, a good amount of data is isolated from training.

Pros

1. Simple, easy to understand, and implement.

Cons

1. Not suitable for an imbalanced dataset.
2. A lot of data is isolated from training the model.

Model Evaluation & Selection

Cross Validation

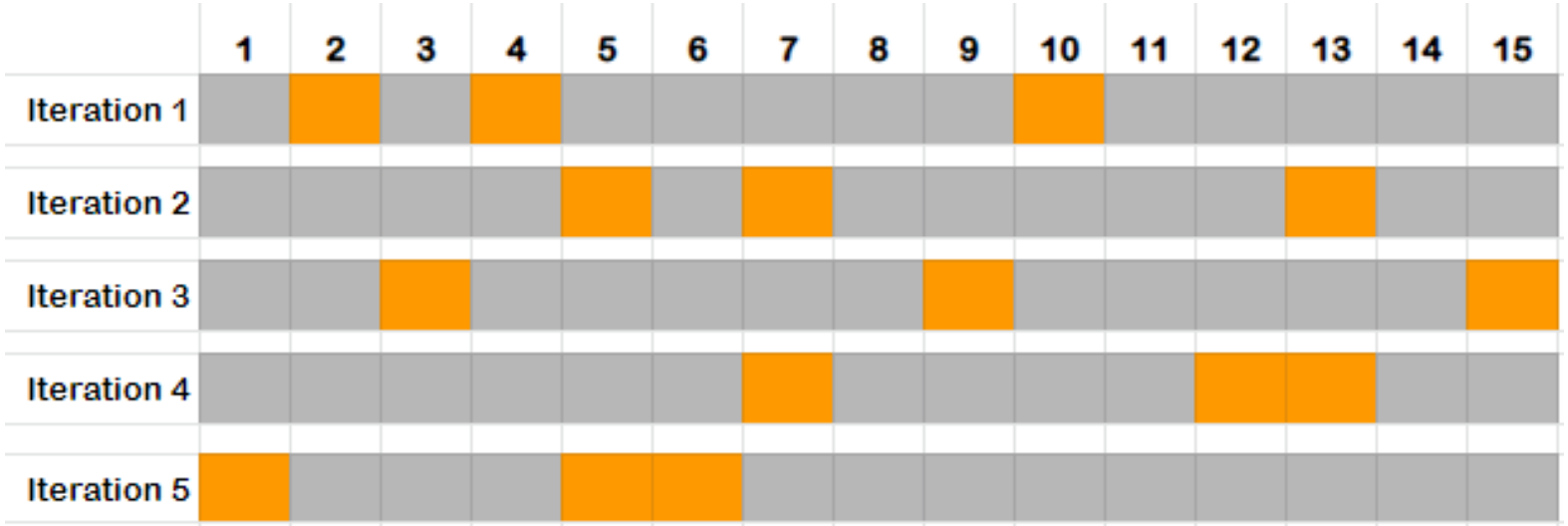
Random subsampling Cross Validation

- Repeated random subsampling validation also referred to as Monte Carlo cross-validation splits the dataset randomly into training and validation.
- Unlikely k-fold cross-validation split of the dataset into not in groups or folds but splits in this case in random.
- The number of iterations is not fixed and decided by analysis.
- the results are then averaged over the splits.

Model Evaluation & Selection

Cross Validation

Random subsampling Cross Validation



Repeated random subsampling validation

Model Evaluation & Selection

Cross Validation

Random subsampling Cross Validation

Pros

1. The proportion of train and validation splits is not dependent on the number of iterations or partitions.

Cons

1. Some samples may not be selected for either training or validation.

1. Not suitable for an imbalanced dataset.

Parameter Tuning and Optimization

- There is a list of different machine learning models.
- They all are different in some way or the other, but what makes them different is nothing but input parameters for the model.
- These input parameters are named as **Hyperparameters.**
- These hyperparameters will define the architecture of the model
- the best part about these is that you get a choice to select these for your model.

Parameter Tuning

- we are not aware of optimal values for hyperparameters which would generate the best model output.
- what we tell the model is to explore and select the optimal model architecture automatically.
- This selection procedure for hyperparameter is known as **Hyperparameter Tuning.**

Model Evaluation & Selection

Why we need Parameter Tuning

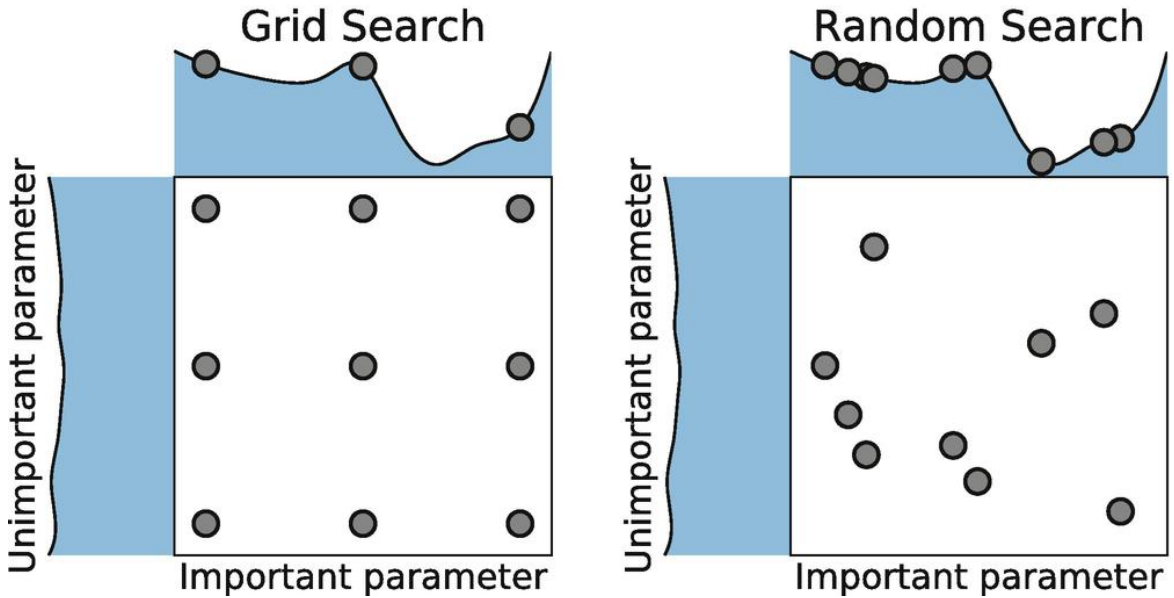
- here we would discuss what questions this hyperparameter tuning will answer for us

  - What should be the value for the maximum depth of the Decision Tree?
  - How many trees should I select in a Random Forest model?
  - Should use a single layer or multiple layer Neural Network, if multiple layers then how many layers should be there?
  - How many neurons should I include in the Neural Network?
  - What should be the minimum sample split value for Decision Tree?
  - What value should I select for the minimum sample leaf for my Decision Tree?

Model Evaluation & Selection

approaches to Hyperparameter tuning

- Manual Search
- Random Search
- Grid Search



approaches to Hyperparameter tuning

- **Manual Search**

  - ❏ we select some hyperparameters for a model based on our gut feeling and experience.
  - ❏ Based on these parameters, the model is trained, and model performance measures are checked.
  - ❏ This process is repeated with another set of values for the same hyperparameters until optimal accuracy is received, or the model has attained optimal error.
  - ❏ This might not be of much help as human judgment is biased, and here human experience is playing a significant role.

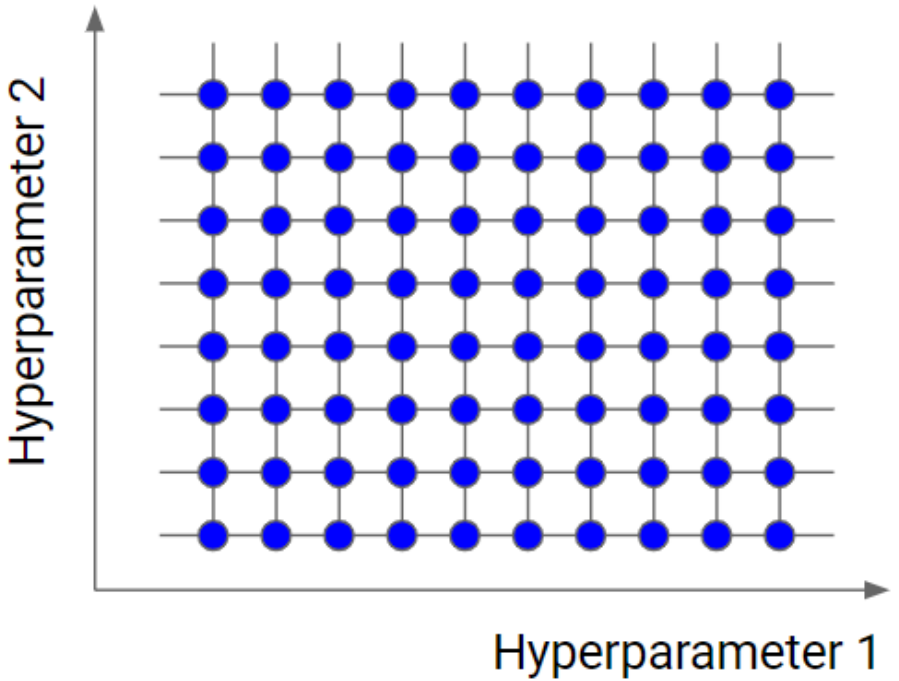
approaches to Hyperparameter tuning

- **Random Search**

  - ❏ doing multiple rounds of this process, it would be better to give multiple values for all the hyperparameters in one go to the model and let the model decide which one best suits.

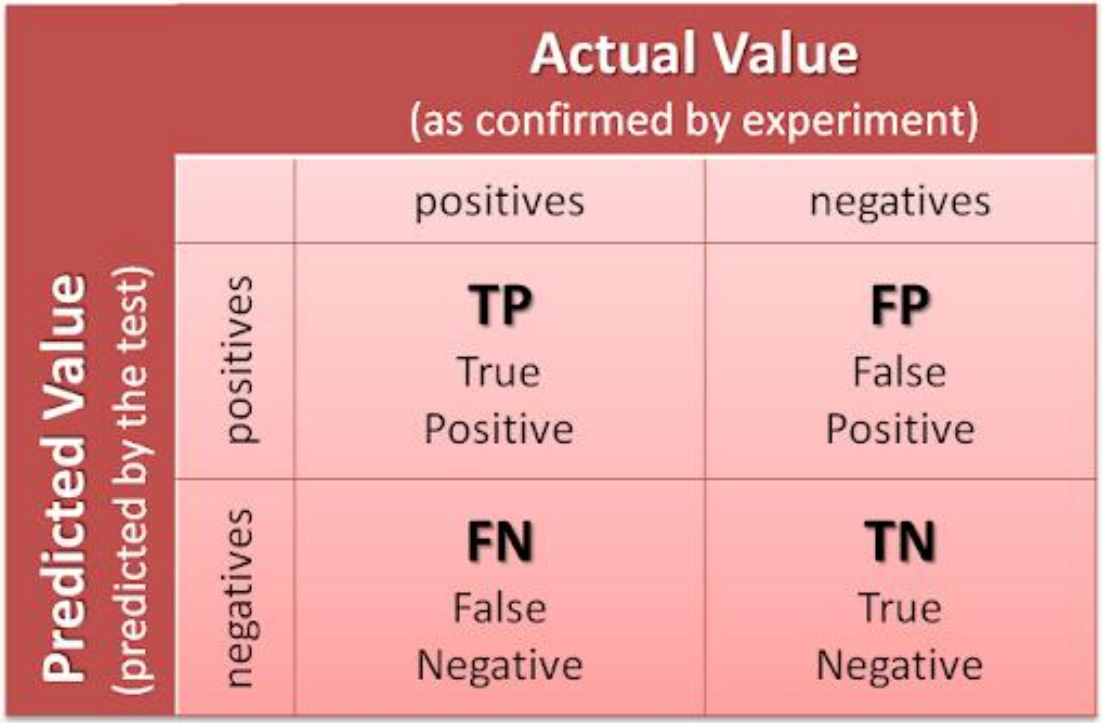
approaches to Hyperparameter tuning

- **Grid Search**
  - ❑ This method is quite an expensive method in terms of computation power and time, but this is the most efficient method as there is the least possibility of missing out on an optimal solution for a model.



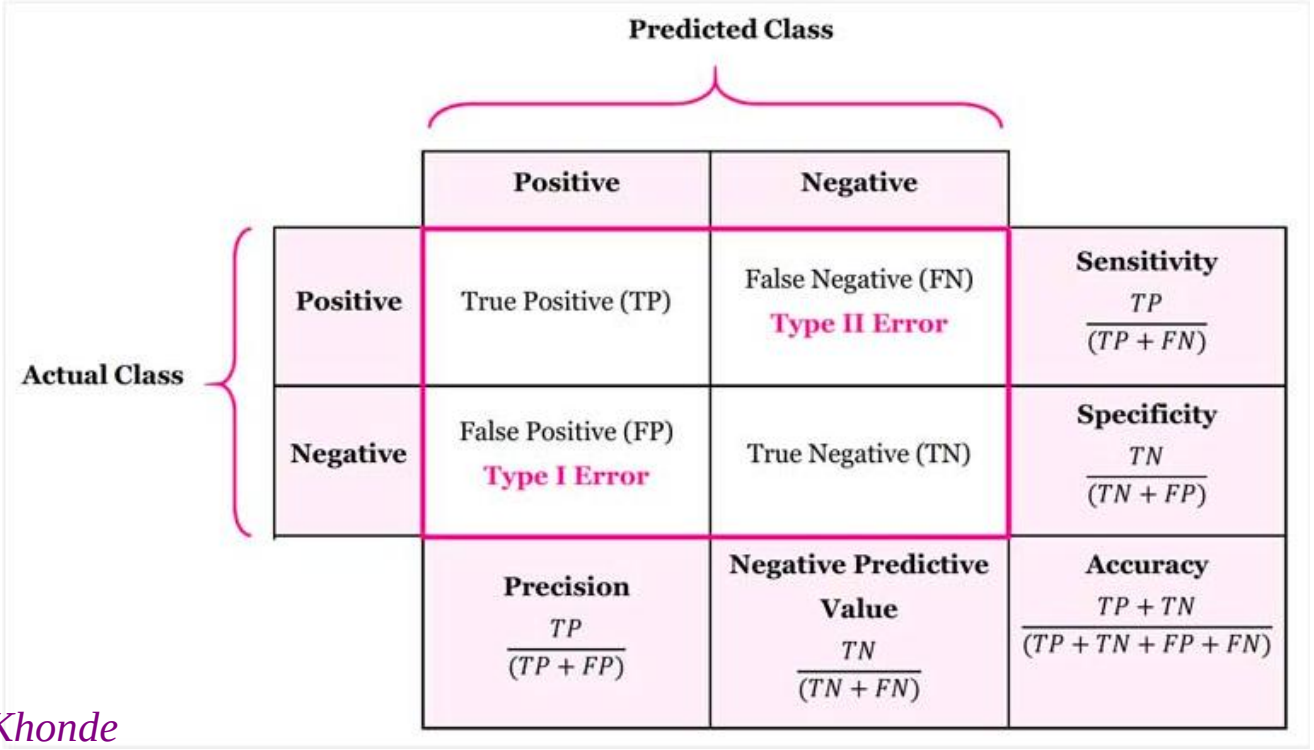
Model Evaluation & Selection

Confusion Matrix



Model Evaluation & Selection

Confusion Matrix

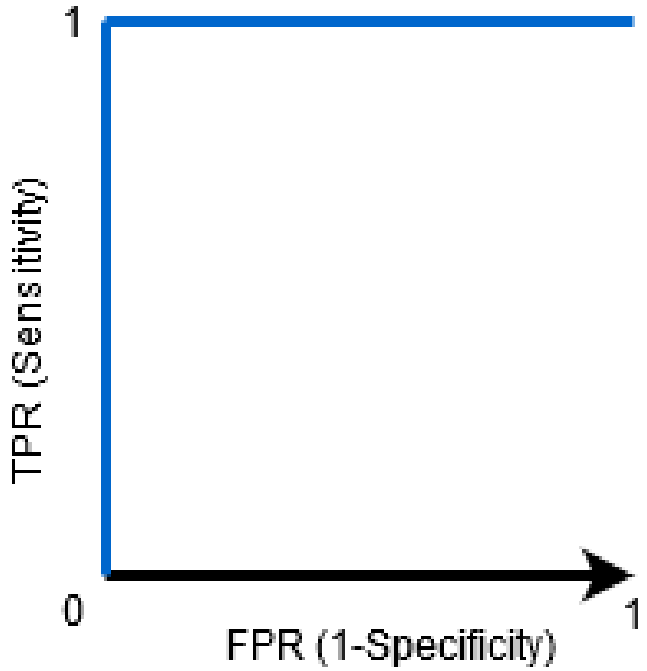


ROC-AUC Curve

- The Receiver Operator Characteristic (ROC) curve **is an evaluation metric for binary classification problems.**
- It is a probability curve **that plots the TPR against FPR at various threshold values** and essentially **separates the 'signal' from the 'noise'.**
- **The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.**
- The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

Model Evaluation & Selection

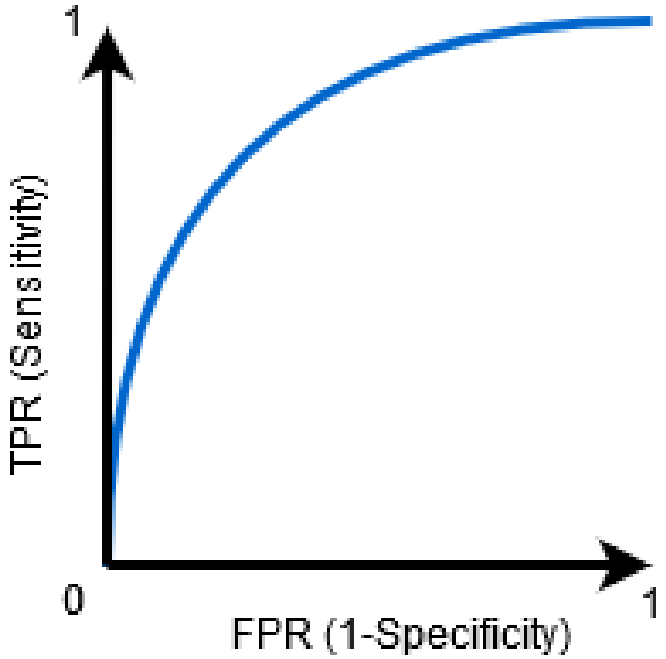
ROC-AUC Curve



- When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly.
- If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.

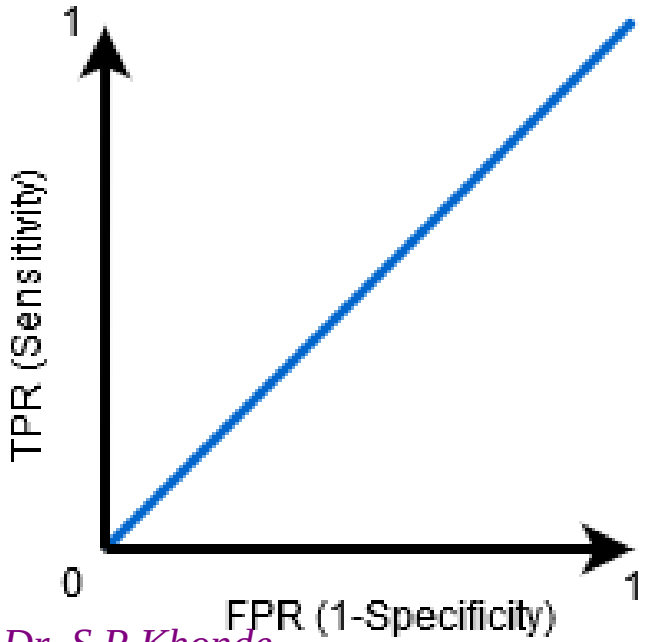
Model Evaluation & Selection

ROC-AUC Curve



- When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values.
- This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.

ROC-AUC Curve



- When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.


THANK
YOU!