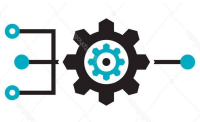
Outcome

Implement data visualization using visualization tools in Python Programming

Outline

Clustering Algorithms 

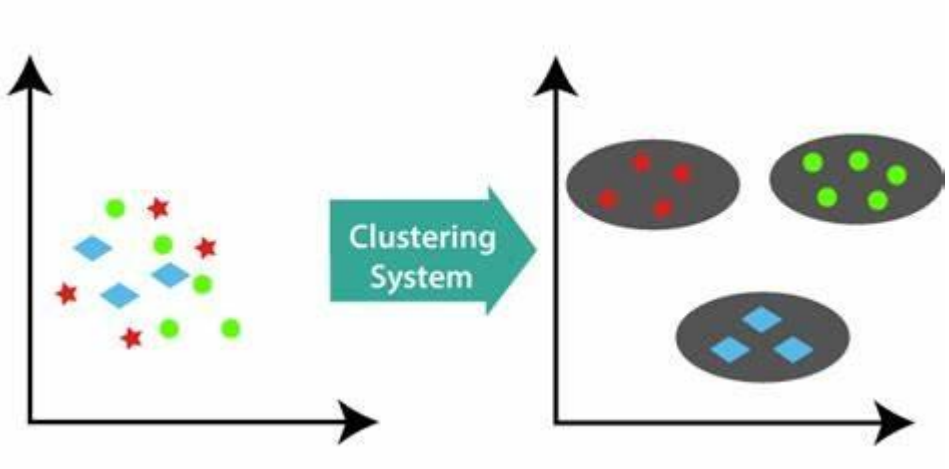
Text Analysis

Model Evaluation and Selection

Clustering

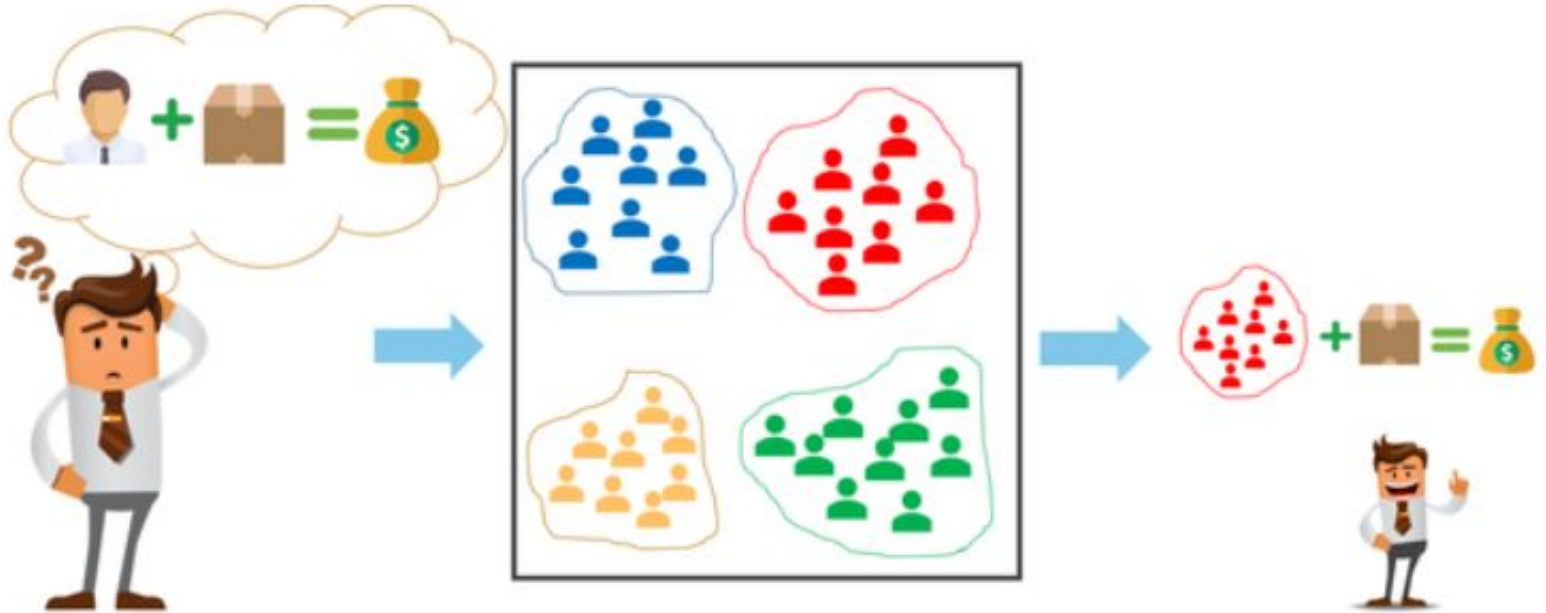
What is Clustering | Cluster analysis

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

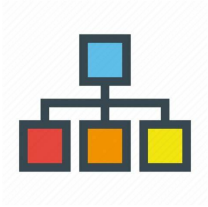
Using Clustering algorithms on the customer base

Selling the products to the targeted audience

Clustering Algorithms

- K-Means

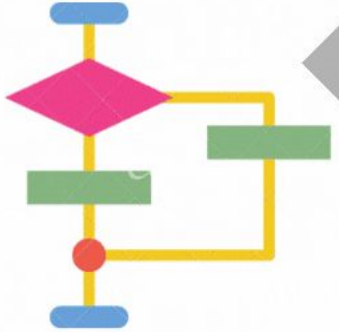
- Hierarchical Clustering



- Time-series analysis

K-MEANS

Clustering Algorithm



Dr. Mahesh R. Sanghavi, SNJB KBJ COE

Clustering Algorithms

K-Means

Unsupervised learning algorithm

Used to solve the clustering problems

Which groups are 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

K defines,

- the number of predefined clusters that need to be created in the process,

  as if K=2, there will be two clusters,

  for K=3, there will be three clusters,

  and so on.

Clustering Algorithms

❑ K-Means

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

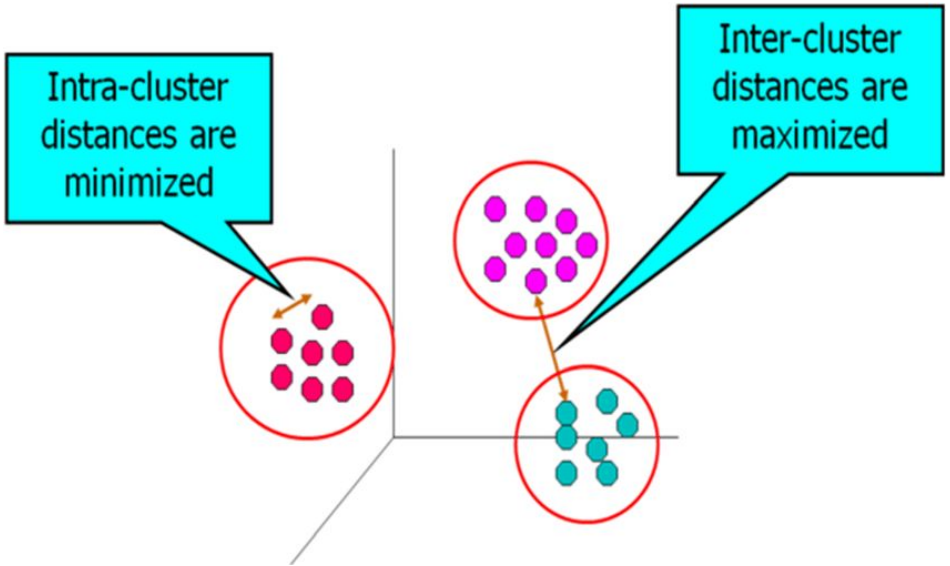
- It allows us to cluster the data into different groups and a convenient way to discover the **categories of groups in the unlabeled dataset on its own without the need for any training.**

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

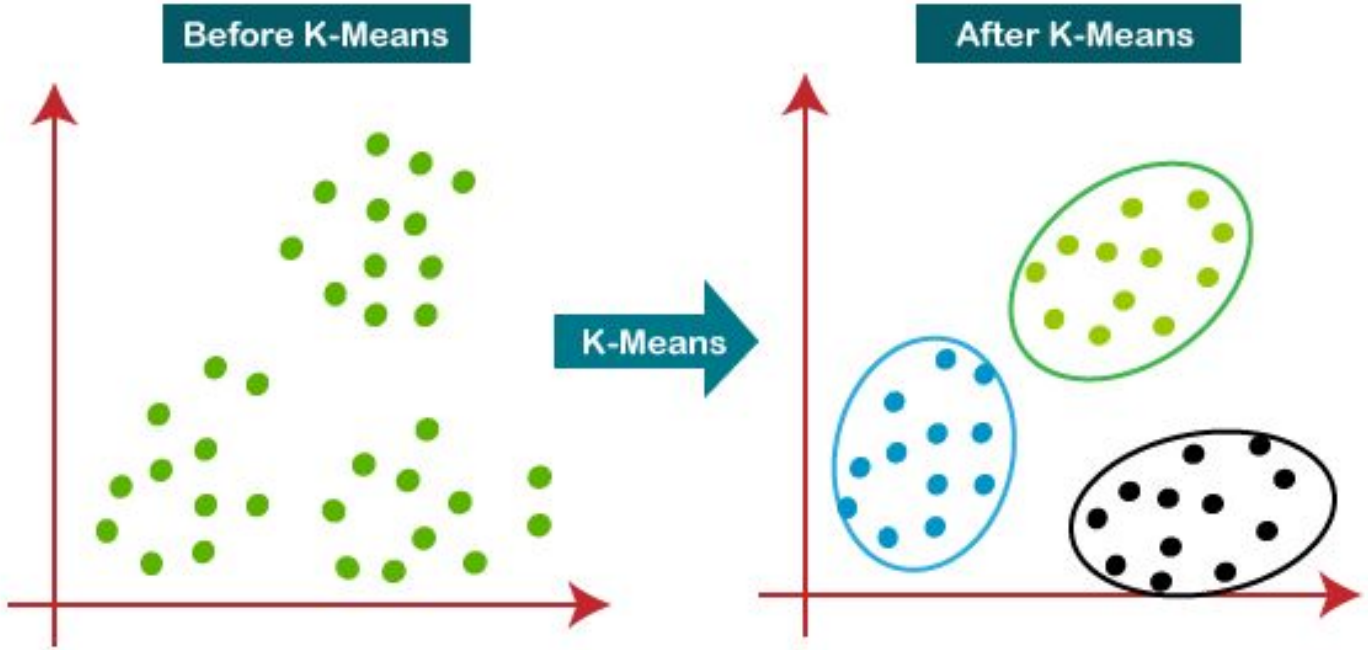
K-Means

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

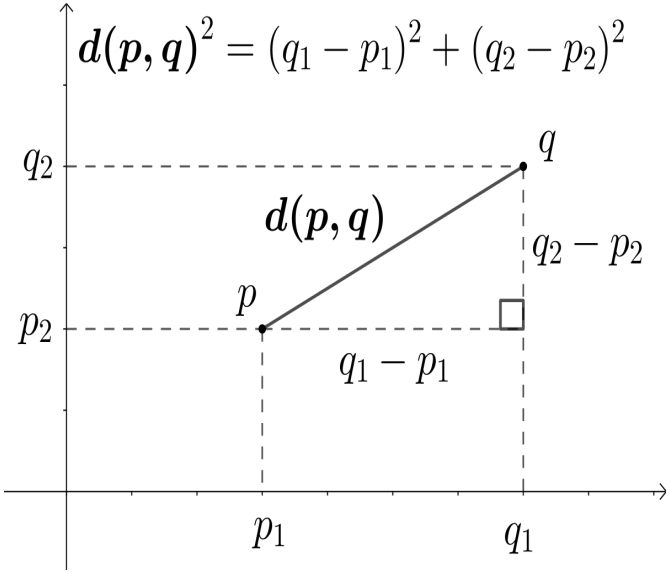
❑ K-Means



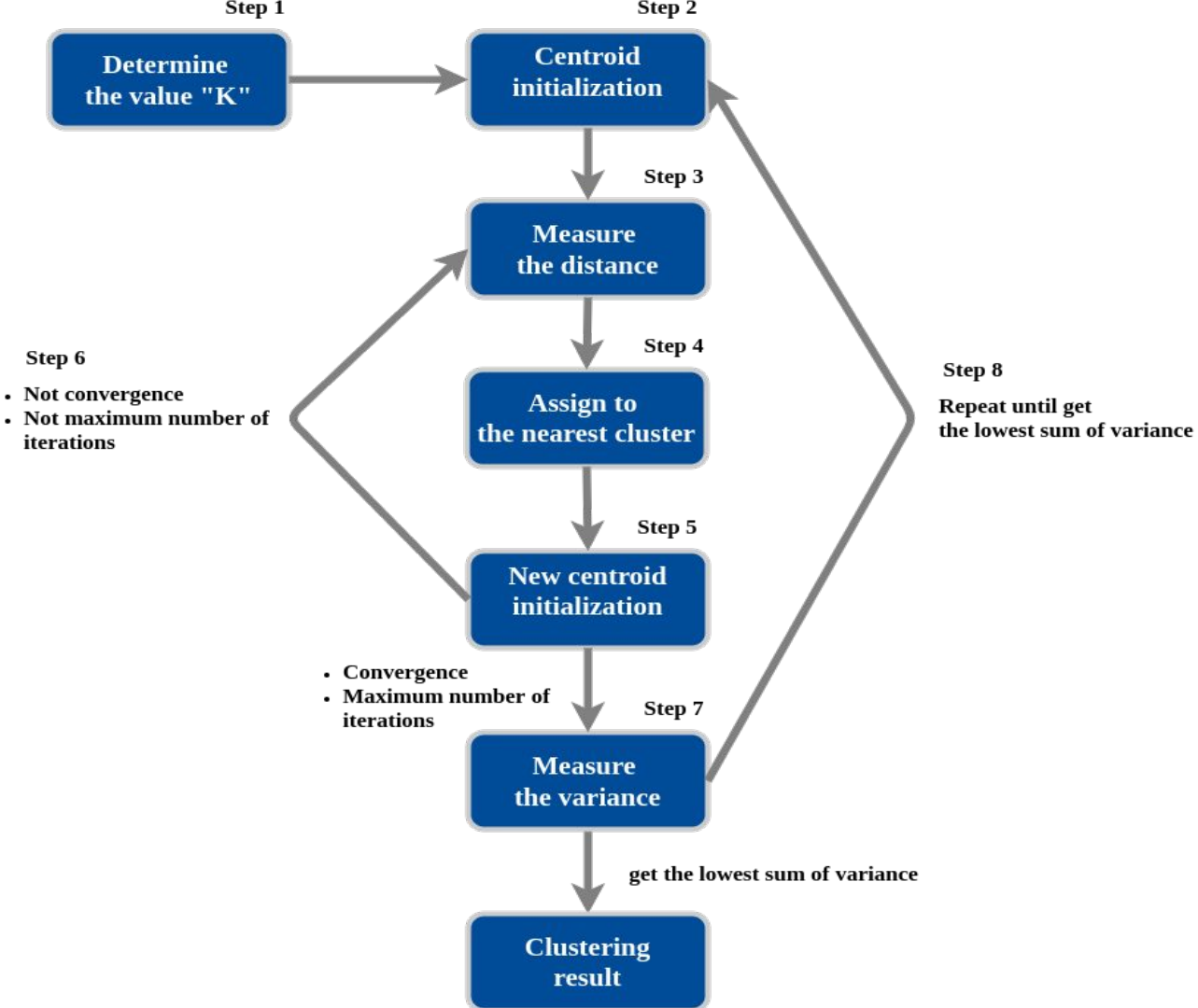
K-Means

- 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

$$d(p,q)^2 = (q_1 - p_1)^2 + (q_2 - p_2)^2$$

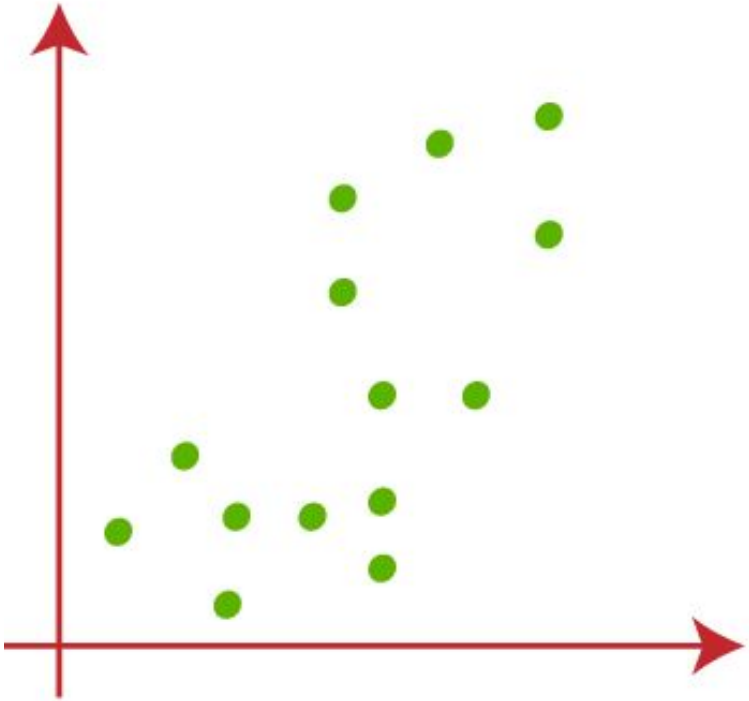


- Link for Solved Example
- Link for Python Code


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



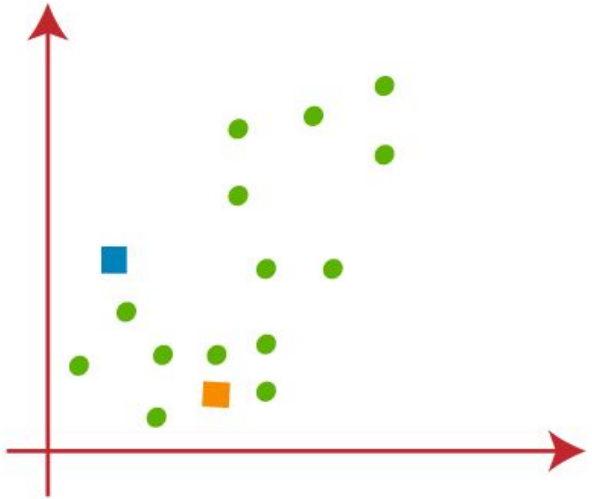
❑ K-Means

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.

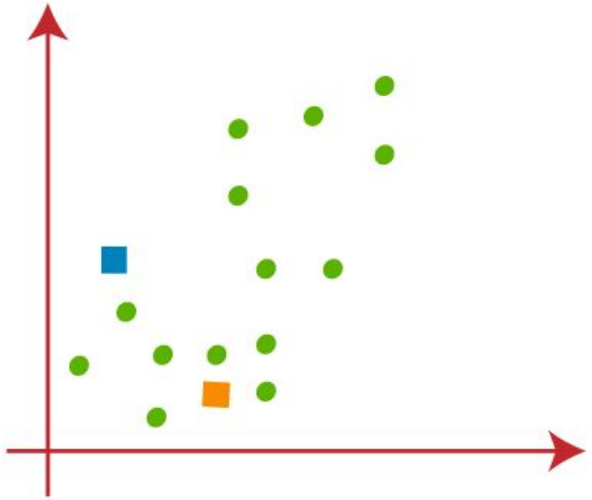


❑ K-Means

How does the K-Means Algorithm Work?

Step 2

Select random K points or centroids. (It can be other from the i

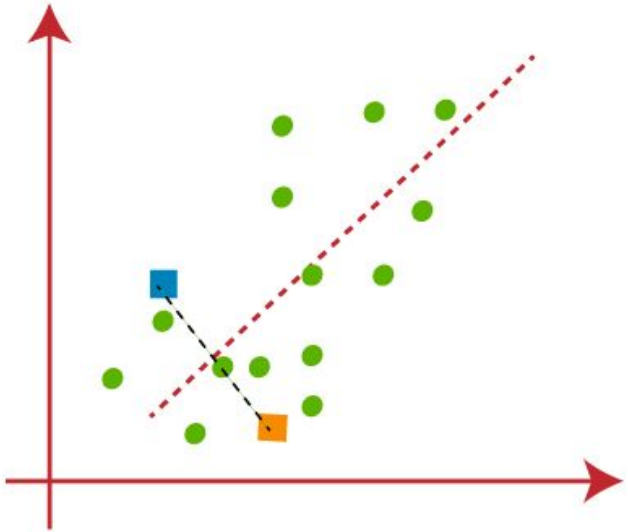


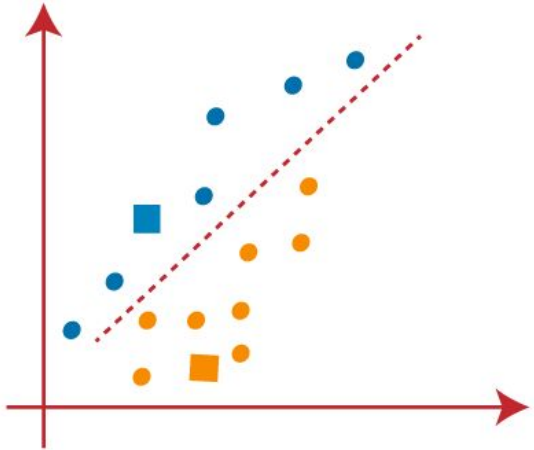
❑ K-Means

How does the K-Means Algorithm Work?

Step 3

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

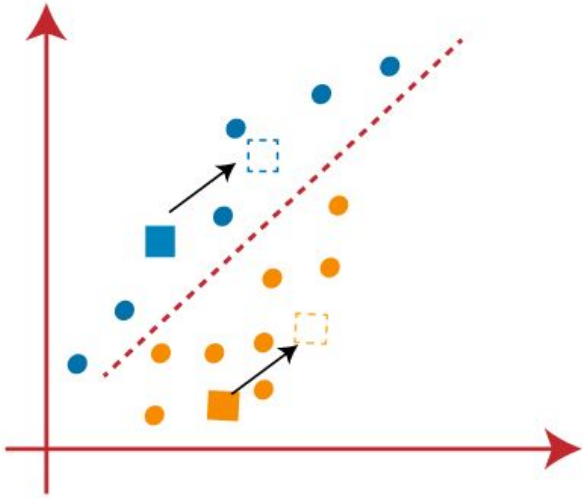


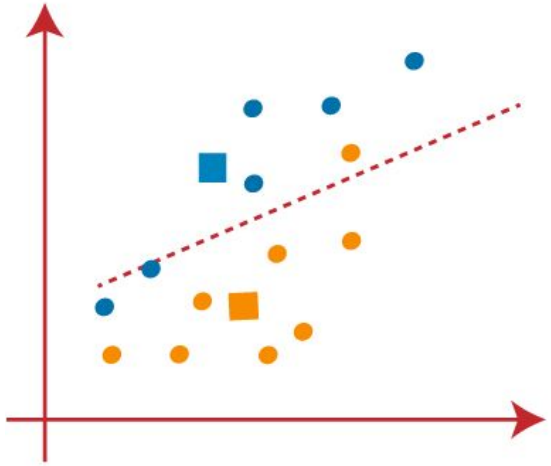


❑ K-Means

How does the K-Means Algorithm Work?

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

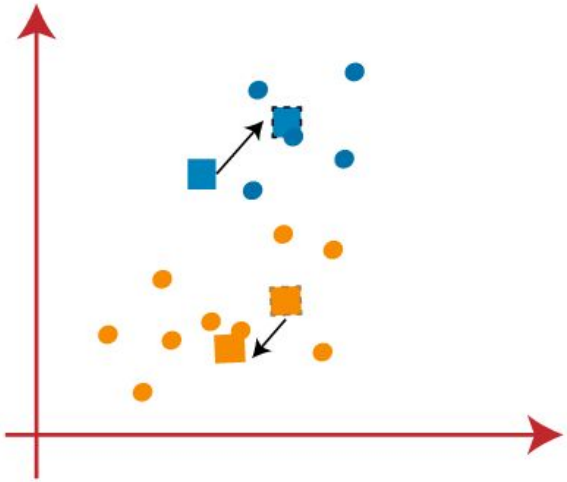


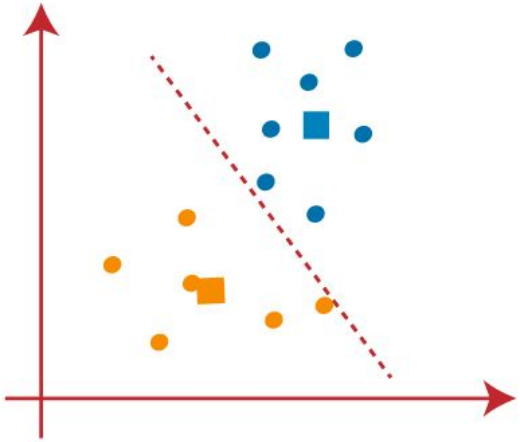


❑ K-Means

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.

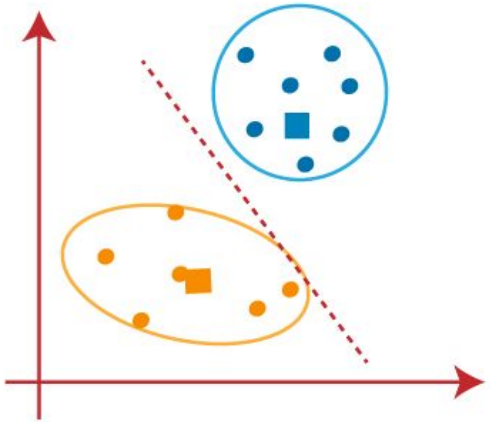




❑ K-Means

How does the K-Means Algorithm Work?

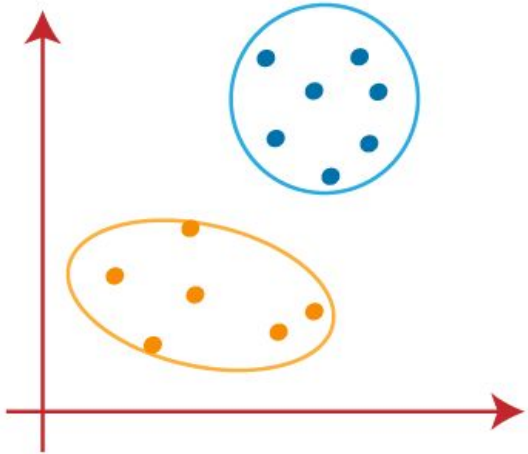
Step 6 If any reassignment occurs, then go to step-4 else go to FINISH.



❑ K-Means

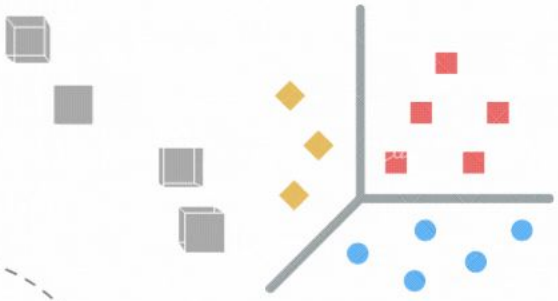
How does the K-Means Algorithm Work?

Step 7 The model is ready.



HIERARCHICAL

Clustering Algorithm



Dr. Mahesh R. Sanghavi, SNJB KBJ COE

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

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

- 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

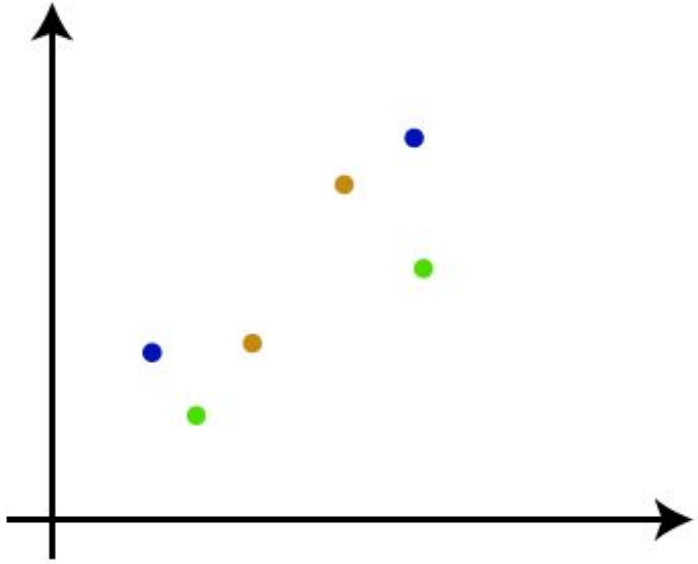
Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 1

Create each data point as a single cluster.

Let's say there are N data points, so the number of clusters will also be N.



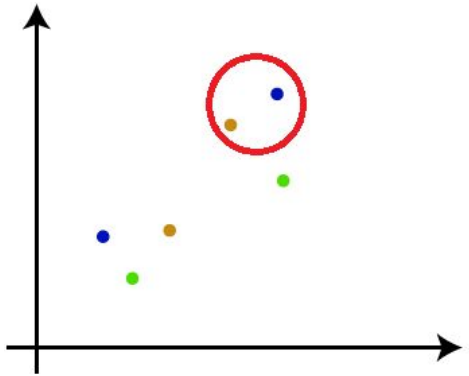
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 2

Take two closest data points or clusters and merge them to form one cluster.



So, there will now be N-1 clusters.

❑ Hierarchical Clustering

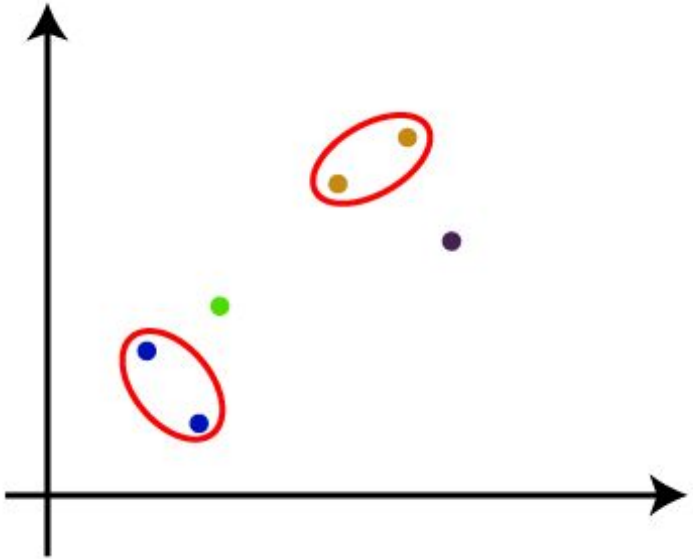
Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 3

Again, take the two closest clusters and merge them together to form one cluster.

There will be N-2 clusters.

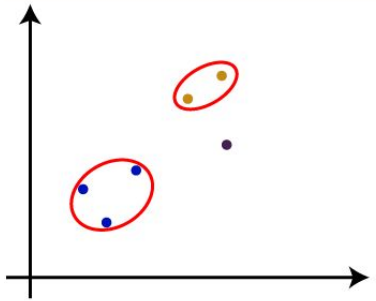


❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

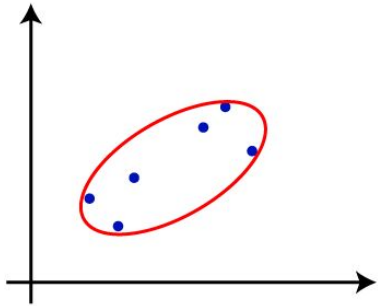
How the Agglomerative Hierarchical clustering Work?

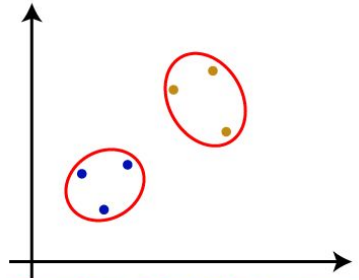
Step 4



Repeat Step 3 until only one cluster left. So, we will get the following clusters.

Consider the images:





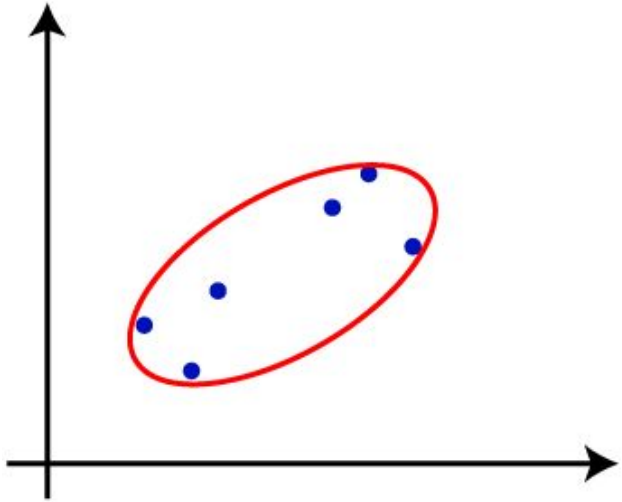
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

How the Agglomerative Hierarchical clustering Work?

Step 5

Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.



❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

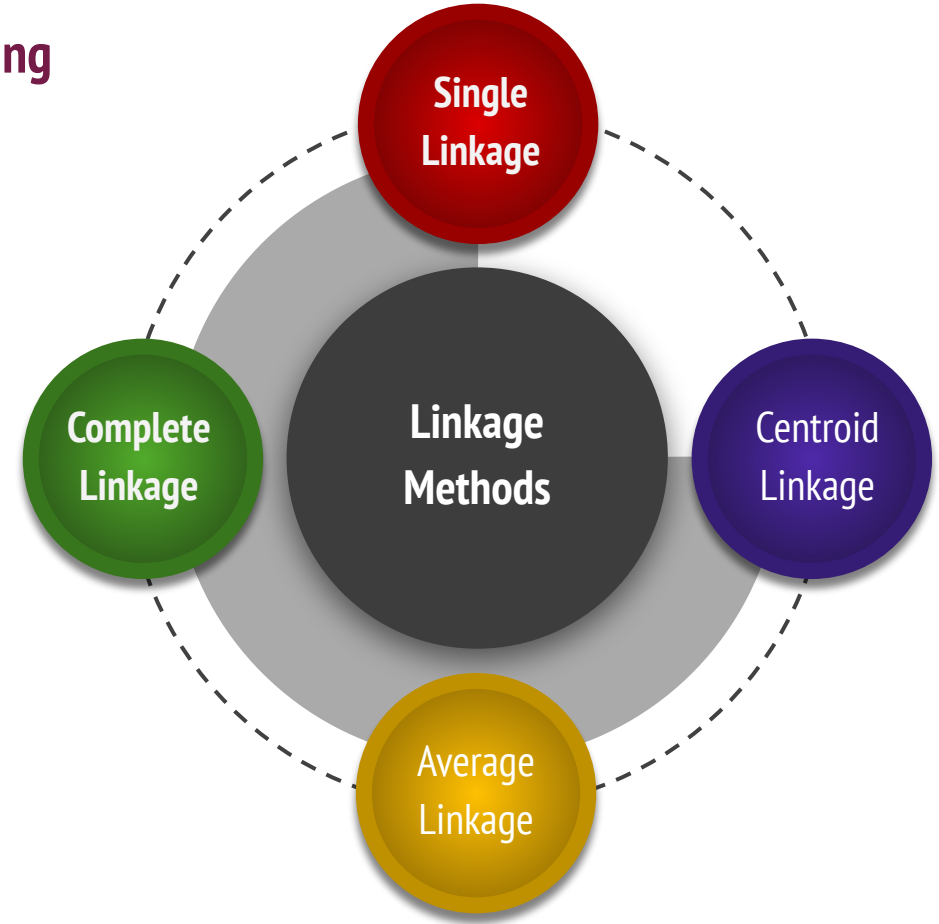
Measure for the distance between two clusters

- 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



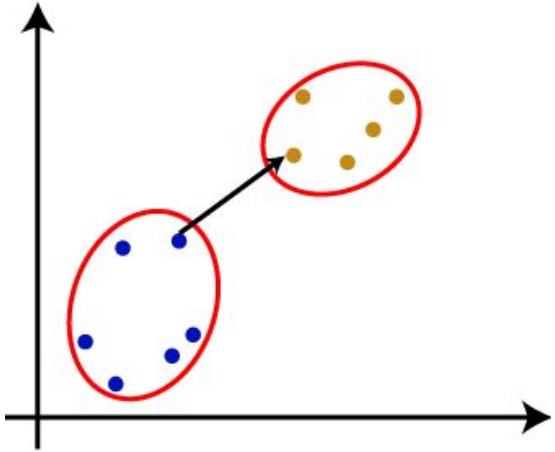
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

Single Linkage

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



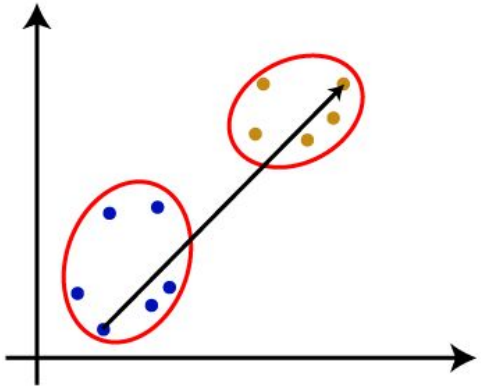
❑ Hierarchical Clustering

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

Average Linkage

- 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.

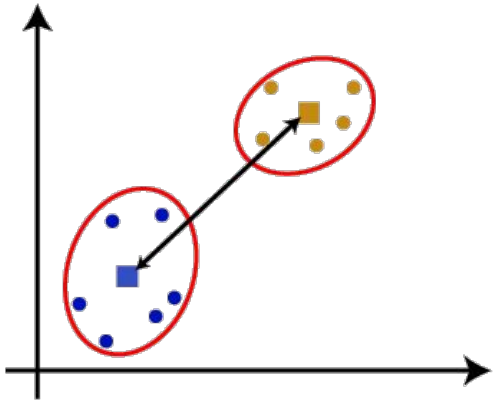
❑ Hierarchical Clustering

Agglomerative Hierarchical clustering

Linkage Methods

Centroid Linkage

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



Reference

❑ 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

Clustering Algorithms

❑ 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 its one cluster
- Large clusters are successively divided
- Top Down

Clustering Algorithms

❑ 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 and generally speaking, usually data points that are closer together are more similar than those further apart
- 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

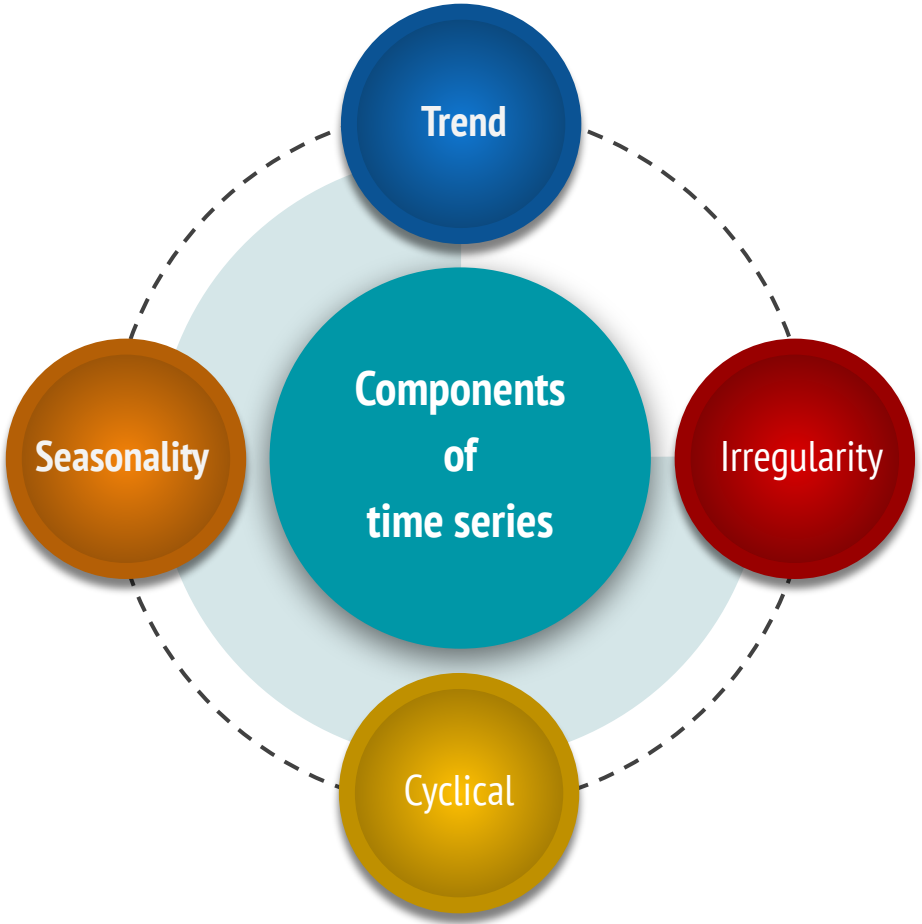
Basic structure of time series data

Observations are recorded every hour.

Clustering Algorithms

❑ Time-series analysis

Components of time series



Time-series analysis

Components of time series

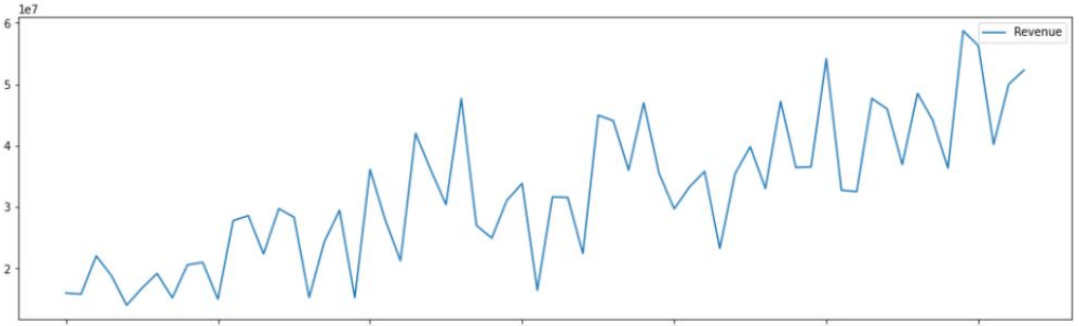
Trend

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

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

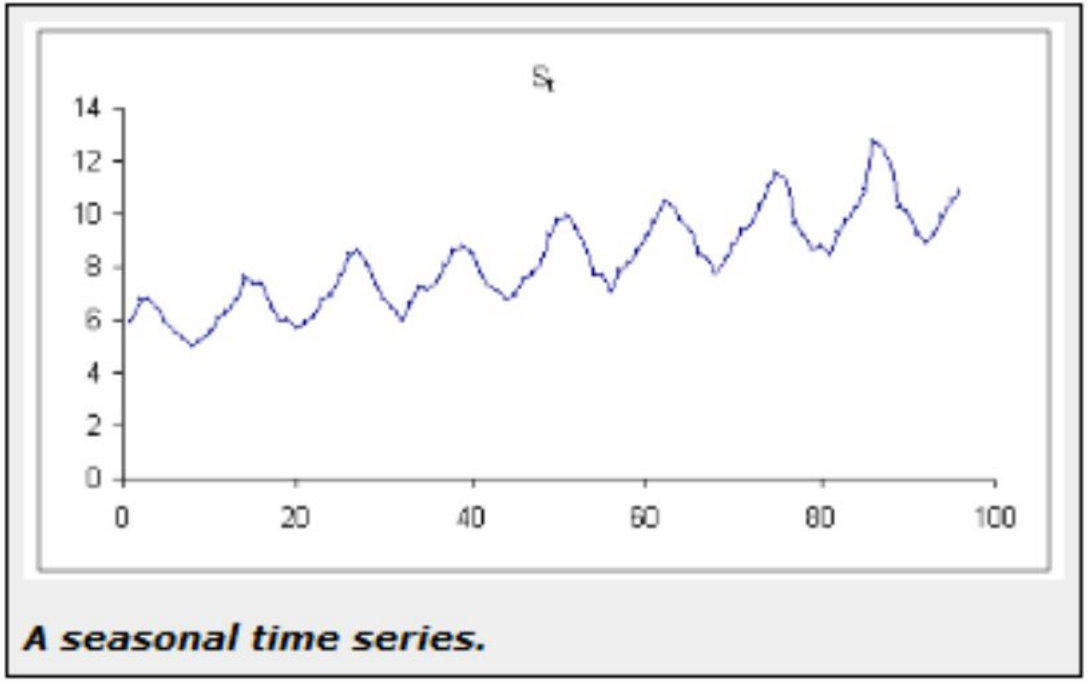
❑ tools for detecting seasonality in time series data

- tools that are useful for detecting seasonality in time series data
- Time series plots
- Statistical analysis and tests

Time-series analysis

Components of time series

Seasonality



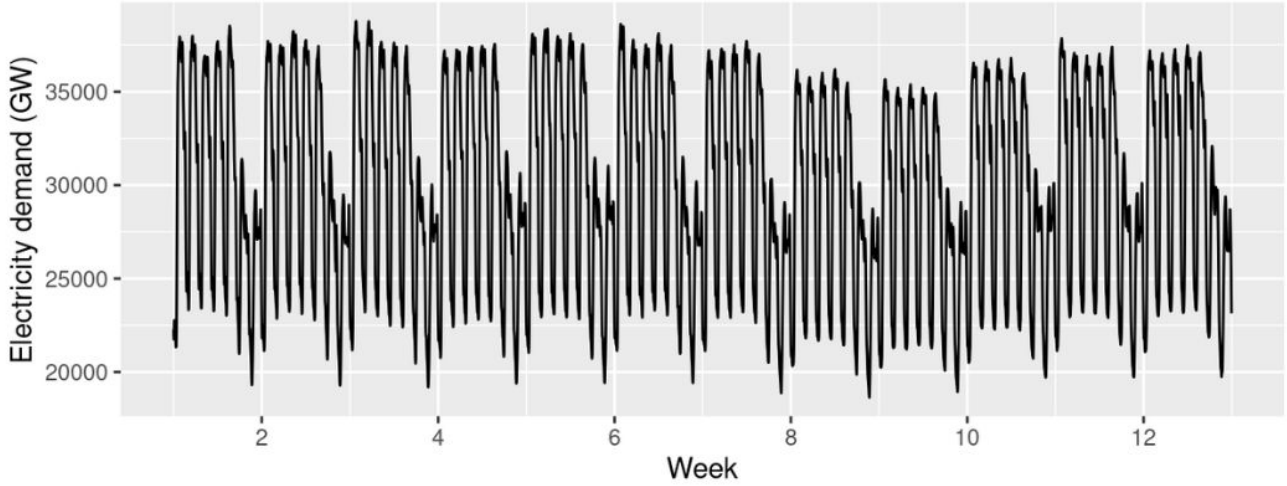
A seasonal time series.

Clustering Algorithms

❑ Time-series analysis

Components of time series

Cyclical



Source

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

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

Clustering Algorithms

Time-series analysis

Time Series analysis can be classified as :

- Parametric & Non Parametric
- Linear & Non Linear
- Univariate & Multivariate

Time-series analysis

Techniques used for time series analysis

- ARIMA Models
- Box-Jenkins multivariate models
- Holt winters exponential smoothing

Clustering Algorithms

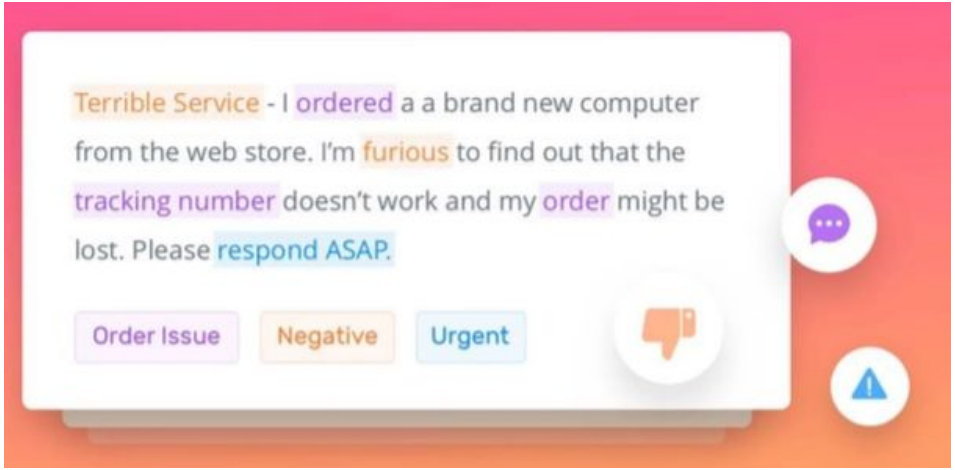
❑ Time-series analysis

Techniques used for time series analysis

ARIMA Models

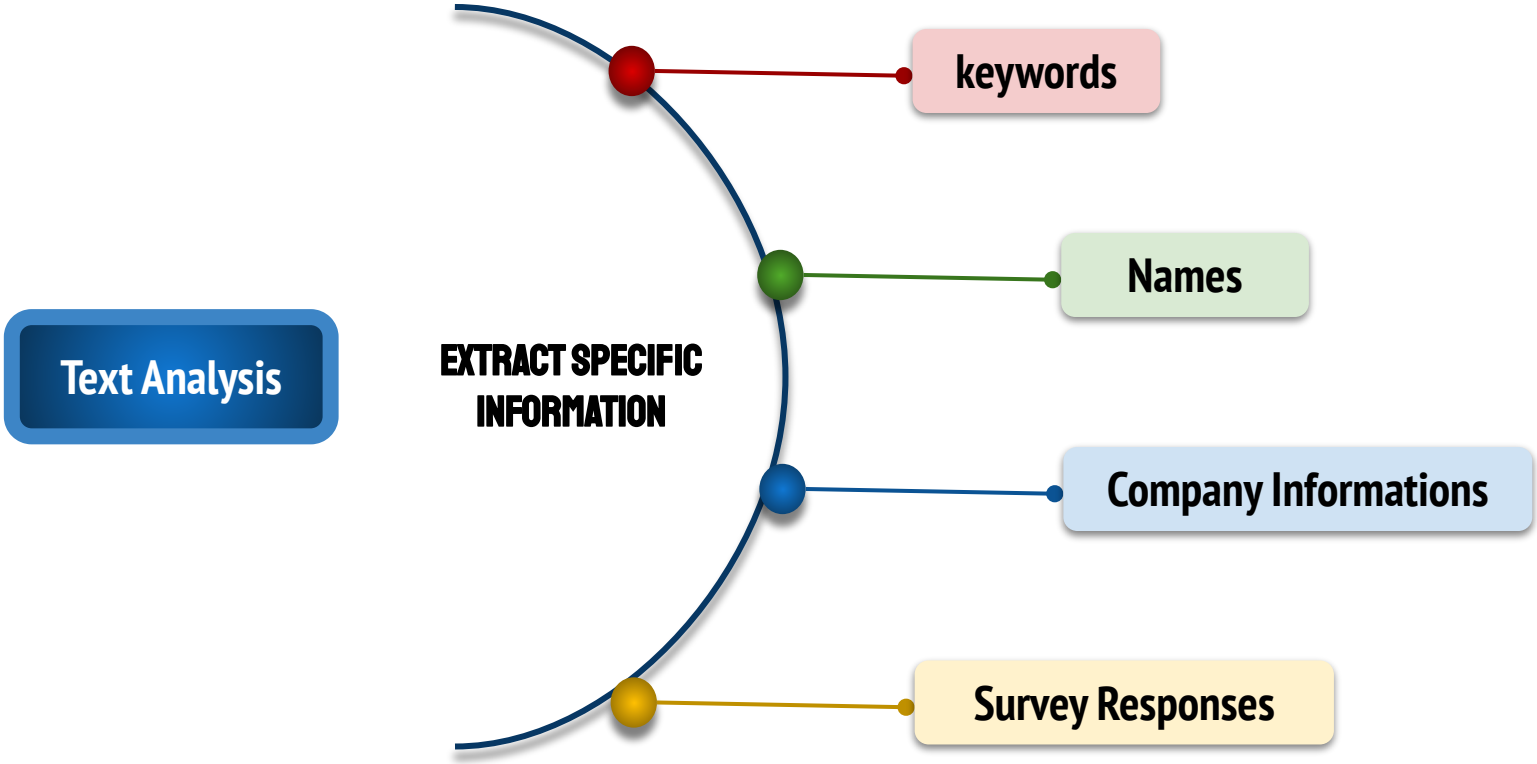
ARIMA stands for AutoRegressive Integrated Moving Average.

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

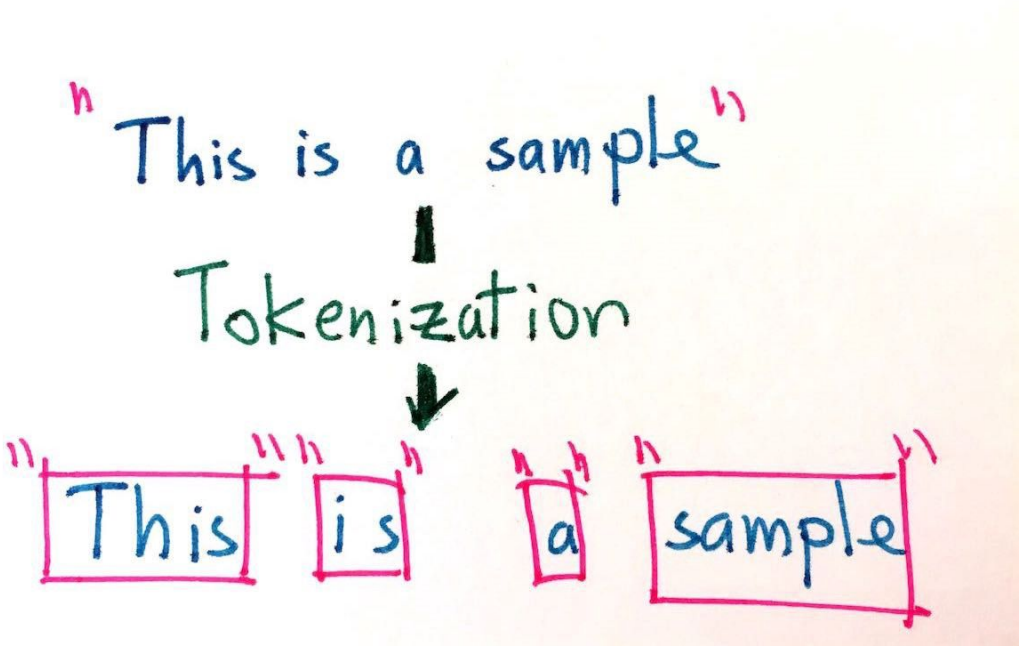
Word Tokenization

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

Text Analysis

Text Analysis Operations using natural language toolkit

Tokenization



Text Analysis

Text Analysis Operations using natural language toolkit

Stop Words Removal

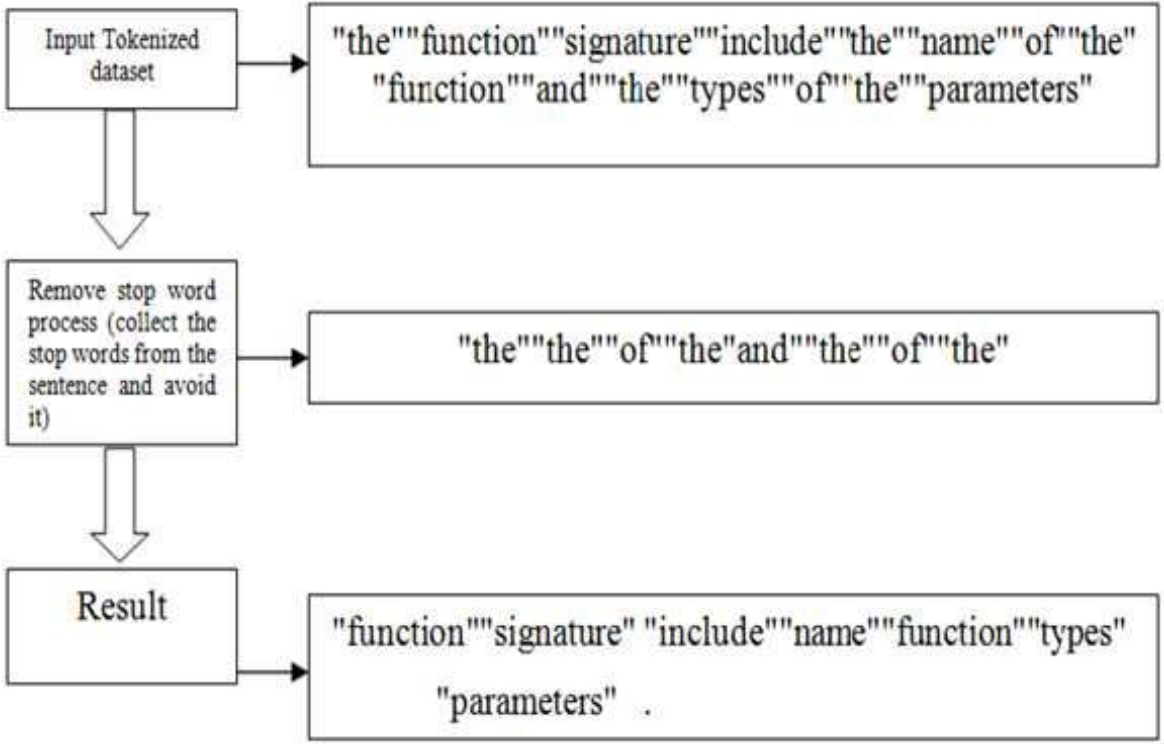
- Stopwords considered as noise in the text.

- 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

Text Analysis Operations using natural language toolkit

Stemming and Lemmatization

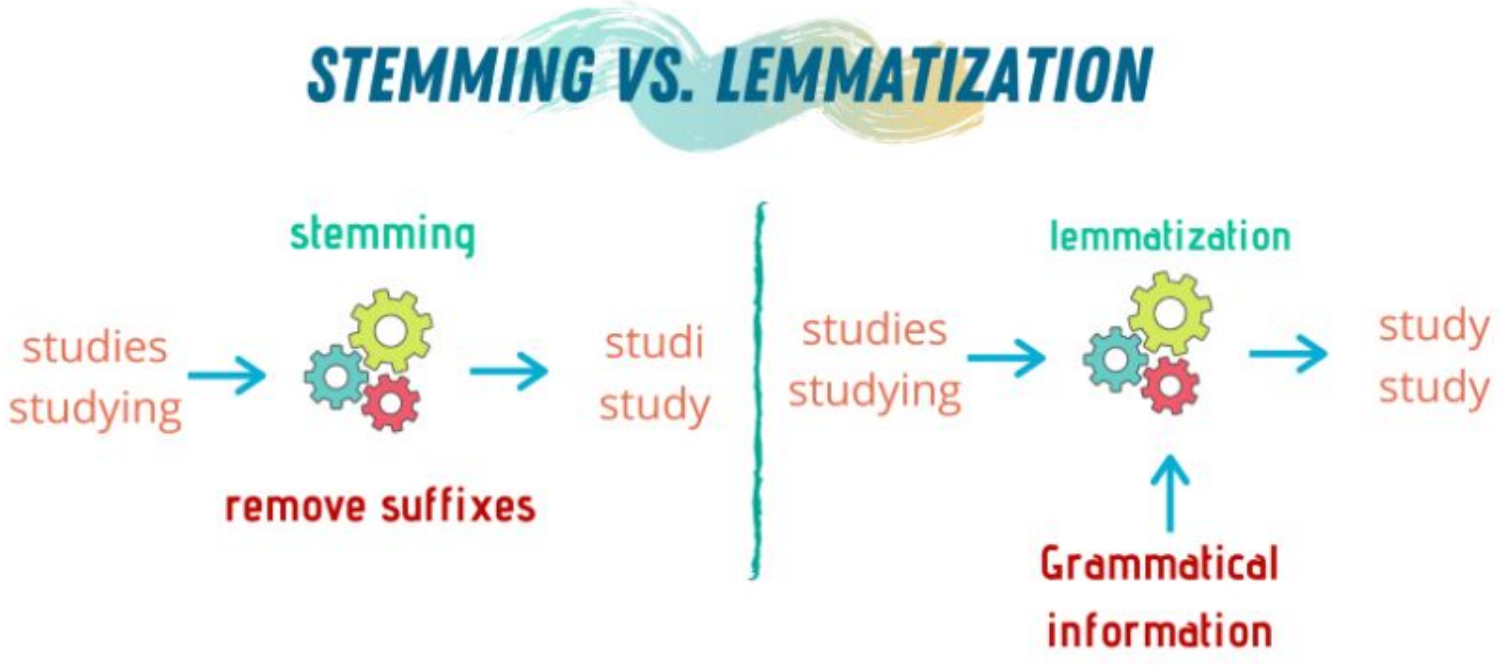
- **Stemming** is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy.

- **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

Text Analysis Operations using natural language toolkit

Stemming and Lemmatization



Text Analysis

Text Analysis Operations using natural language toolkit

Stemming and Lemmatization

Example

Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

Text Analysis

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

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

Text Analysis

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

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

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

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)

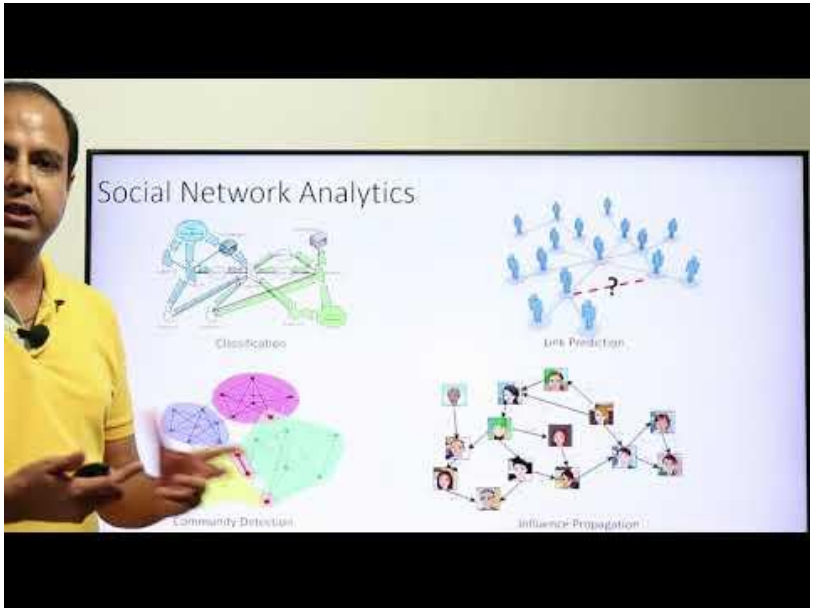
- is the process of investigating social structures in terms of nodes and edges that connect them through the use of networks and graph theory.

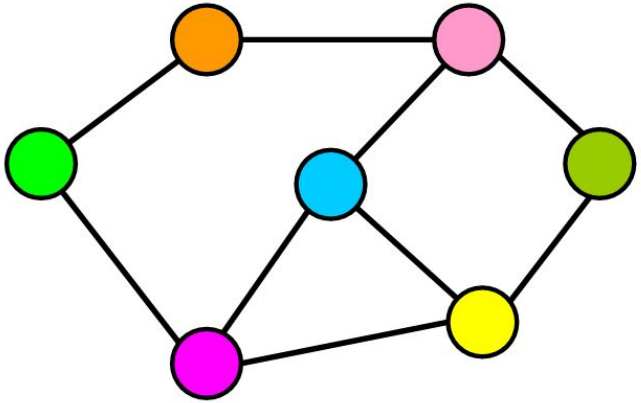
Source

,

Text Analysis

Introduction to social network analysis

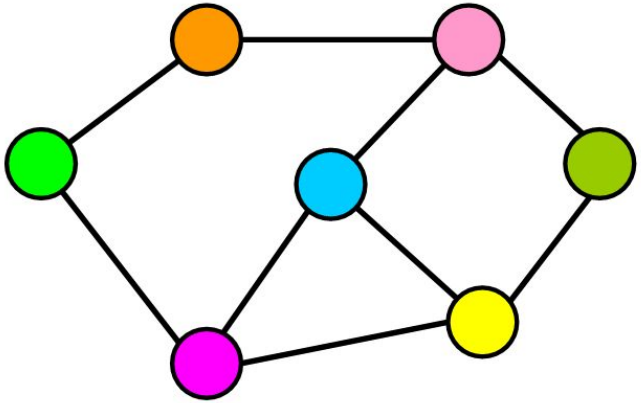




Introduction to social network analysis

Graph Theory

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



Text Analysis

Introduction to business analysis

Business analysis

- "**Business analysis** is the practice of enabling change in an enterprise by defining needs and recommending solutions that deliver value to stakeholders.

- It enables an enterprise to articulate needs and the rationale for change, and to design and describe solutions that can deliver value."

,

Model Evaluation & Selection

Cross Validation

- **Cross-Validation** also referred to **as out of sampling technique** 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.

,

Model Evaluation & Selection

Why Cross Validation is important

- During the development of an ML model using the training data, the model performance needs to be evaluated.

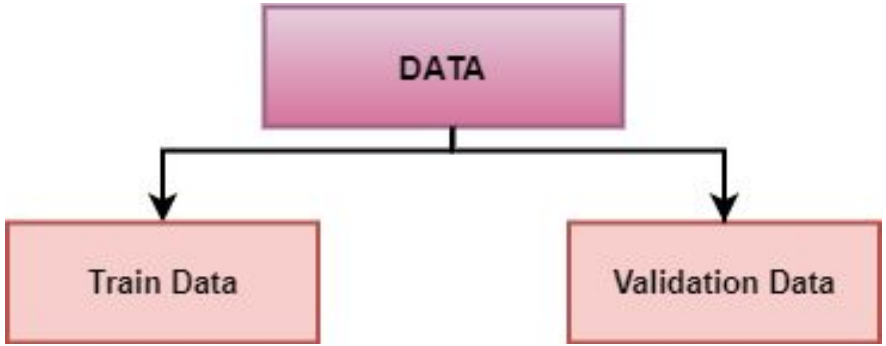
- Here's the importance of cross-validation data comes into the picture.

,

Model Evaluation & Selection

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.

,

Model Evaluation & Selection

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**

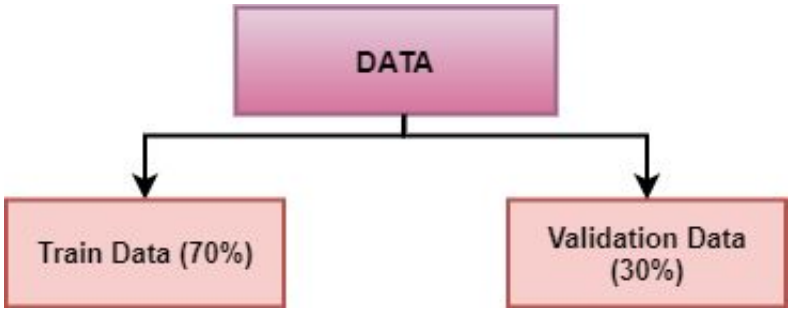
,

Model Evaluation & Selection

Cross Validation

Hold Out Cross Validation

- 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

Hold Out Cross Validation

- **Hold out Approach in Sklearn**

  - ❏ The hold-out approach can be applied by using train_test_split module of sklearn.model_selection

  - ❏ In the below example we have split the dataset to create the test data with a size of 30% and train data with a size of 70%. The random_state number ensures the split is deterministic in every run.

,

Model Evaluation & Selection

Cross Validation

Random subsampling Cross Validation

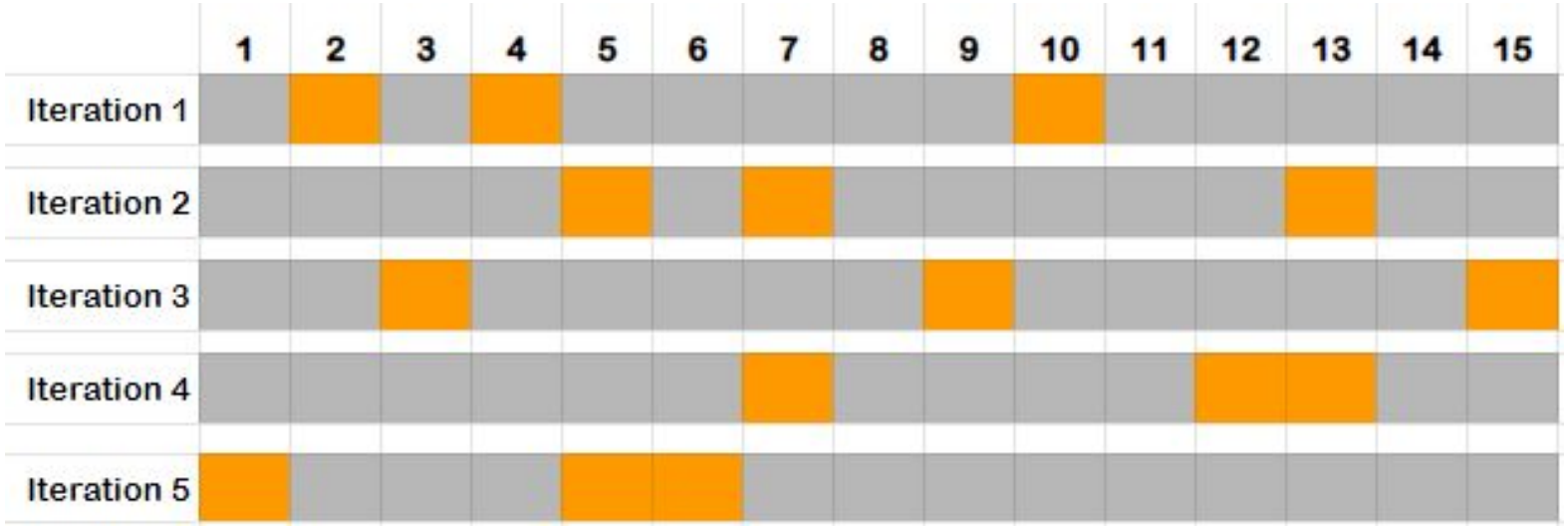
- Repeated random subsampling validation also referred to as Monte Carlo cross-validation splits the dataset randomly into training and validation.
- Unlikely k-fold cross-validation split of the dataset into not in groups or folds but splits in this case in random.
- The number of iterations is not 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.
2. Not suitable for an imbalanced dataset.

Model Evaluation & Selection

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.

Model Evaluation & Selection

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

Parameter Tuning

Hyperparameters

Parameters

Score

n_layers = 3 n_neurons = 512 learning_rate = 0.1	→	Weights optimization	→	85%
n_layers = 3 n_neurons = 1024 learning_rate = 0.01	→	Weights optimization	→	80%
n_layers = 5 n_neurons = 256 learning_rate = 0.1	→	Weights optimization	→	92%

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

Why we need Parameter Tuning

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

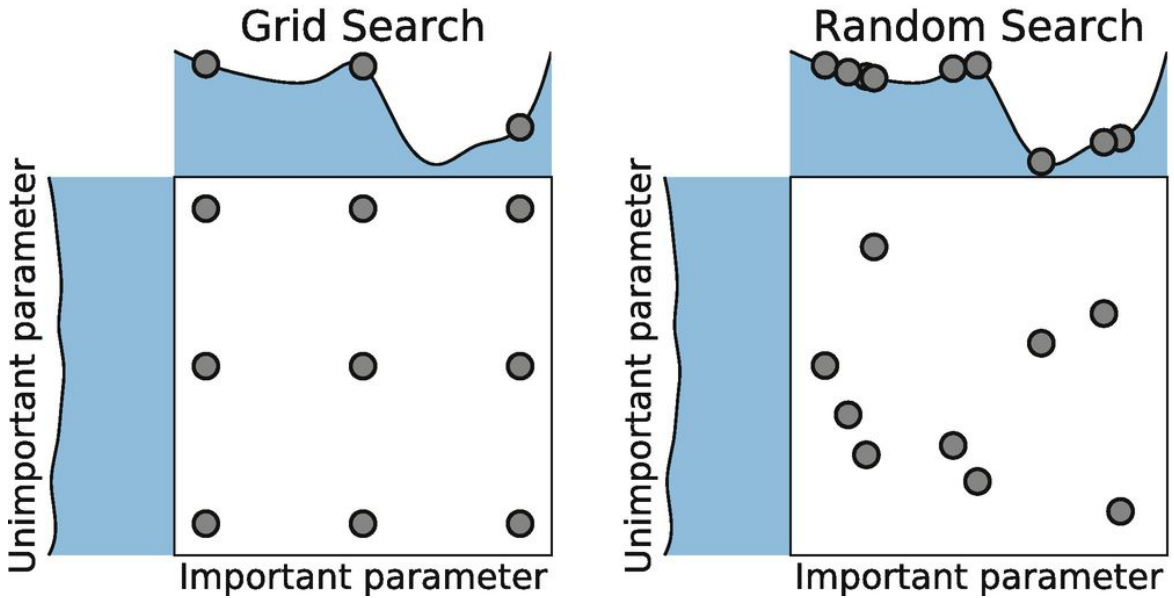
- How many iterations should I select for Neural Network?
- What should be the value of the learning rate for gradient descent?
- Which solver method is best suited for my Neural Network?
- What is the K in K-nearest Neighbors?
- What should be the value for C and sigma in Support Vector Machine?

Note : this a few questions which could be answered by hyperparameter tuning.

Model Evaluation & Selection

approaches to Hyperparameter tuning

- Manual Search
- Random Search
- Grid Search



Model Evaluation & Selection

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.

Model Evaluation & Selection

approaches to Hyperparameter tuning

- **Random Search**

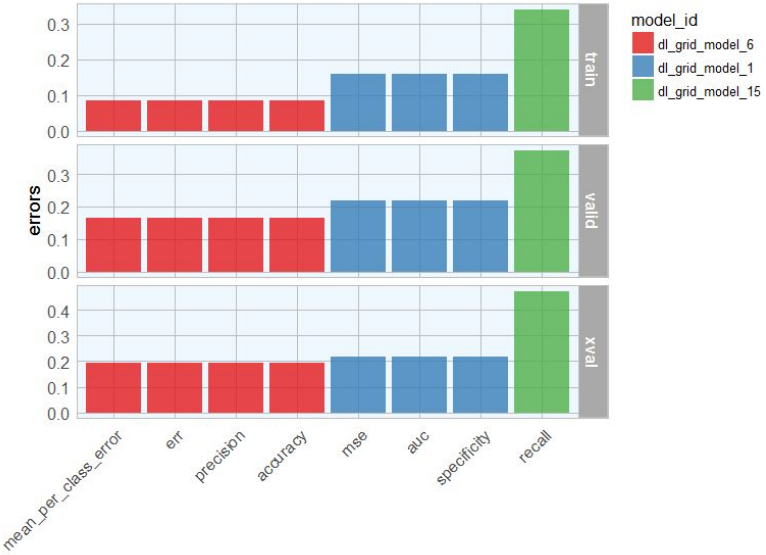
  - ❏ doing multiple rounds of this process, it would be better to give multiple values for all the hyperparameters in one go to the model and let the model decide which one best suits.

Model Evaluation & Selection

approaches to Hyperparameter tuning

- **Grid Search**

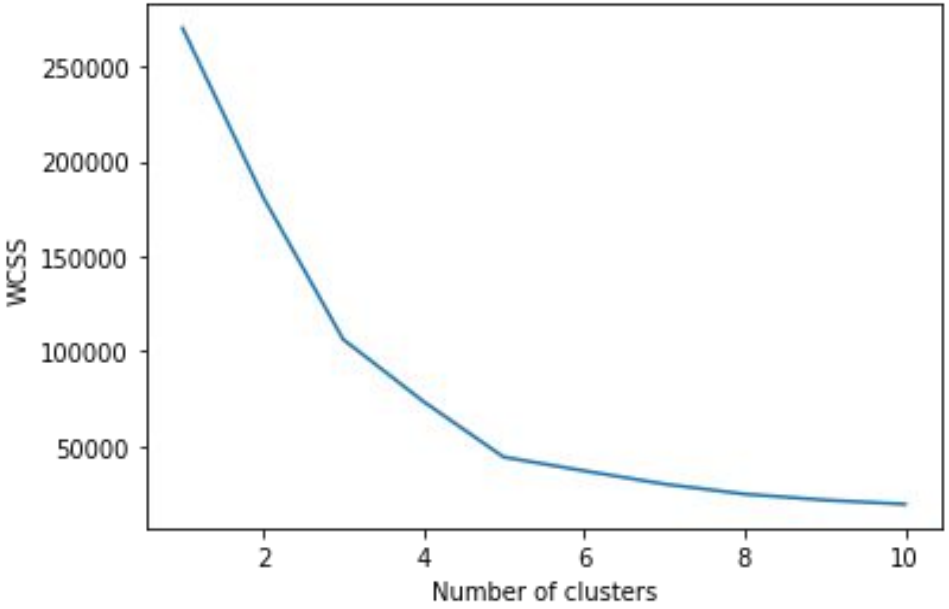
❑ This method is quite an expensive method in terms of computation power and time, but this is the most efficient method as there is the least possibility of missing out on an optimal solution for a model.

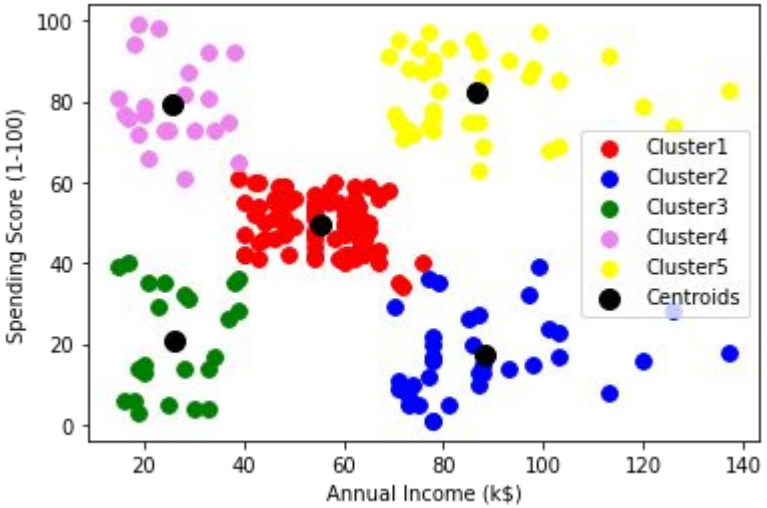


Model Evaluation & Selection

Elbow Method

Click here for demonstration





Model Evaluation & Selection

https://www.mygreatlearning.com/blog/hyperparameter-tuning-explained/

https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/

,

Model Evaluation & Selection

Metrics for Evaluating Classifier Performance

https://blog.ineuron.ai/Hold-Out-Method-Random-Sub-Sampling-Method-3MLDEXAZML#:~:text=Hold%2DOut%20Method%20is%20a,your%20model%20works%20on%20it.

https://vitalflux.com/hold-out-method-for-training-machine-learning-model/

https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d

https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html