# U-3 (DS Honors)
# Clustering Methods.

cluster → a no. of similar things that occur together

→ Technique in which data points are arranged in similar groups dynamically without any pre-assign of groups

Appln → • Image Processing • Insurance • Recommendation engine

* properties of clustering algo -
  - scalability
  - ability to deal with different data types
  - minimal to requirements for domain knowledge for determining input parameters
  - Interpretability & usability

* Typical Requirements → • clustering in datamining.
  • Discovery of clusters with arbitary slope.

* Problems with Clustering →
  - Not address all requirn.
  - dealing with large no. of dimension & data items → problematic
  - effectiveness depends on distance
  - distance measure shouldn't define it.
  - result of clustering → different ways of interpretation

Clustering Methods → 1) Partioning Methods → KMeans, K Medoids, K-Medians

2) Quality & choosing cluster → SSE, Silhouette, Elbow, Index

3) KNN

4) DBSCAN

5) K-Modes.

# K-Means.

- Heuristic Method
- supervised learning algo.
- solves clustering problems
- group of unlabeled dataset in diff clusters
- k → defines no. of predefined clusters needs to be created.
  K=2 → 2 clusters.
- each dataset belongs to one group has similar project.
- minimizes sum of b/w data points & clusters
- distance calculation → Euclidean distance.

Properties →
- Alway k-clusters
- Atleast one item in cluster
- clusters are non-hierarcial & don't overlap
- every member of cluster is closer to cluster

Adv →
- efficient in computation
- easy to implement

Disadv →
- only when mean is defined
- need to specify k, no. of clusters in advance
- Trouble with noisy data & outliers
- not suitable to discover cluster with non complex shapes.

Algo → 1) define k 2) Select k random points as centroid.
3) compute distance from centroid → $d = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$
4) Recompute centroids of clusters as → $(x_c, y_c) = \left(\frac{\Sigma x}{m}, \frac{\Sigma y}{m}\right)$
5) Repeat 3 & 4 until one of follow is meet:
   a) Centroid don't change
   b) Points remain in same cluster
   c) Max no. of iterations are reached

# * Assessing Quality & Choosing no. of clusters

1) Internal Evaluation Measures ~~→ a) SSE (Sum of Squared Errors)~~
   - a) Sum of Squared Errors (SSE) –
     - Measures total distance from data points to assigned cluster centr
     - Lower SSE → better clustering
   - b) Silhouette Coefficient –
     - measure how well data points fits its assigned cluster compared to o
     - Value close to 1 → better clustering.
2) External evaluation Measures
   - a) Adjusted Rand Index (ARI)
   - b) Normalized Mutal information (NMI)

Choosing no. of clusters

1) Elbow Method – • determine no. of clusters
   • Inertia is sum of all distances for the datapoint from centroid of that cluster.
2) Silhouette Analysis
3) Gap Statistics
4) K-Means Centroid distance
5) Domain knowledge.

# KNN

- K-Nearest Neighbour
- Supervised learning approach
- assumes similarity b/w new case/data & available test case
- regression & classification (more preffered)
- lazy learning algo → not learning from training set immediately
  → stores dataset & time of classification → action

Adv → • simple to implement
• robust → noisy training data
• more effective → training data → large

Disadv → • need to determine k value
• computation cost high
• sensitive → choice of distance

Why KNN? → discovery category or class of selected dataset without difficulty

Working → • select no. of k of neighbors
• Calculate Euclidean distance of K no. of neighbours
• Take K nearest neighbours as per calculated Euclean distance
• k-neighbour, count no. of data point in each categories.
• Assign new data point to category which no. neighbour is max.
• Our model is ready

Applⁿ → • Image recognition • Customer segmentation • Anomaly detection.
• Recommender System

# 1-NN (1-Nearest Neighbor):

Classify a datapoint by assigning it class of its closest neighbor in training data.

- Pros → • easy to implement & understand
• Robust to outliers
- Cons → • sensitive → noise in data
• computationally expensive → large datasets.

## * K-Medians.

- clustering algo.
- similar to K-Means, instead of mean to calculate
- Uses median to calculate.
- Datasets → outliers or non spherical clusters.

Adv → • Robust to Outliers
• suitable for non-spherical clusters
• Intuitive meaning of medoids.

Disadv → • expensive
• sensetive to initial medoid selection
• Not suitable for high dimensional data

Appln → • Customer segmentation • Image segmentation
• Anomaly detection • Document Clustering

## * Density Based Spatial Clustering

## * Hierarchial Clustering.

Agglomerative

Divisive

# DBSCAN-

- Density Based Spatial Clustering of ~~clustering~~ Appln with Noise.
- groups together closely packed points
- unsupervised learning.

$\epsilon$ → radius of neighbourhood around point x.
minpts → required to form a density cluster.

Adv → • don't need to specify no. of clusters.
• flexible in shape & size of clusters
• able to deal with noise & outliers
• easy → someone who knows dataset → to set parameters.

Disadv → • Input parameters → difficult to determine
• some situation very sensative to input parameter
• confused → border points belong to two cluster
• result → distance metric
• hard to guess parameters.

# Hierarchical Clustering.

- method of cluster analysis in which data points are arranged in hierarchy of clusters.

Adv → • simple to implement
• easy & results in hierarchy (more info)
• doesn't need prespecify no. of clusters.

Disadv → • large clusters breaks
• difficult to handle diff sized clusters & shapes.
• sensitive to noise & outliers
• can't be changed or deleted once done.

Hierarchical Clustering
Types

**Agglomerative clustering**

- bottoms up approach
- each item → own cluster
- identifies small cluster
- Iteratively clustered are merged together
- also known as AGNES

**Divisive Clustering**

- top down approach
- all item → one cluster
- large cluster
- large clusters are successively divided.
- also known as DIANA.

# * Roles of Dendrograms →

- diagram representing tree of hierachy.
- The similar two objects are less in height of that joins them.
- Major info lost here

→ 1) Visualization
2) Cluster analysis
3) Understanding Cluster Relationship
4) Anomaly Detection
5) Model Interpretation
6) Communication & Collaboration

# * Choosing number clusters in Hierarchial clustering

↳ 1) Elbow Method
2) Silhouette Analysis
3) Gap Statistic
4) Dendogram Analysis
5) Domain Knowledge.

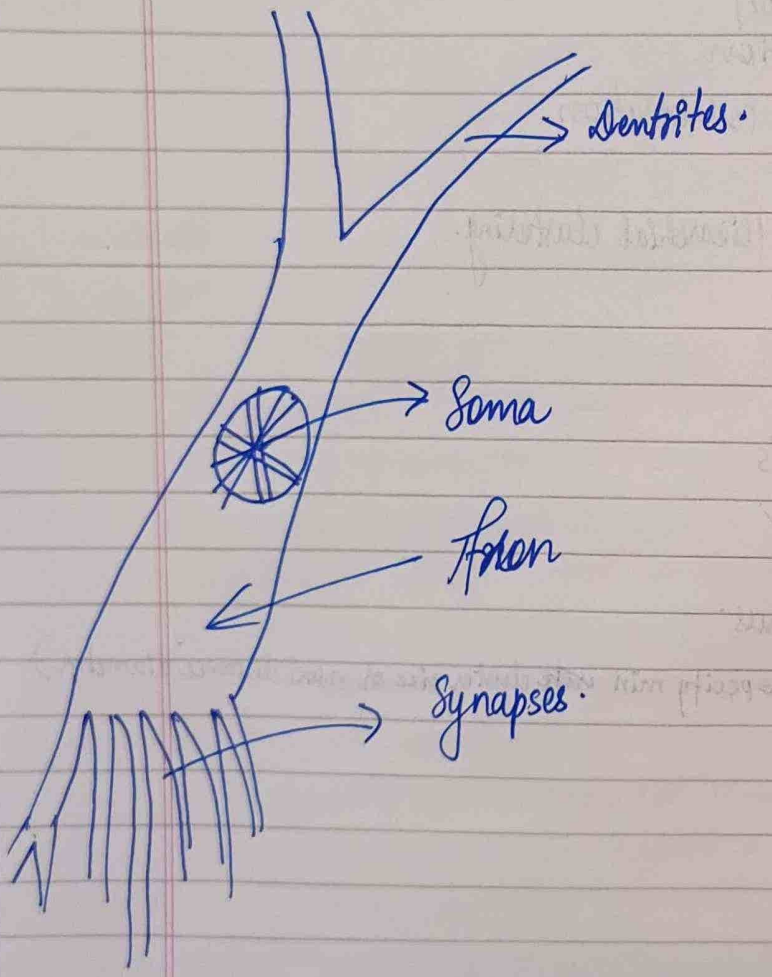
# * Divisive Clustering Techniques.

1) Stopping Criteria. (specify min cluster size or max distance/diameter).
2) Cluster quality metrics
3) Domain Knowledge

# U-4 (DS Honors)
# Artifical Neural Network.

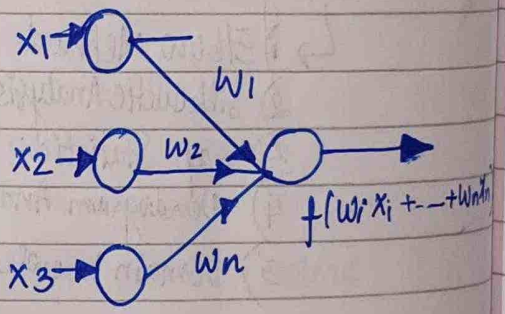
## Biological Neuron Model

- Dendarities → accepts inputs.
- soma → proces input.
- Axon → Turns processed inputs into output.
- Synapses → electrochemical contact b/w neurons




## Artifical Neuron Model

- ANN
- Edge or connection or link
- Weight or connect in s



## * Models of a Neuron -

→ Input Layer → similar to dendrites → input layer receives various inputs or features

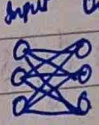
→ Weights → determine significance of each input.

→ Activation Function → determine output of ANN based on weighted sum of inputs.

→ Outputs → calculates using activation function & passed to next layer

## * Network Architectures.

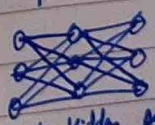
1) feedforward Neural Network (FNN) -




Adv → • simple & easy to implement & train.
• efficient for learning static patterns & mappings.
• widely used for introductory learning & basic tasks.
• unidirectional

Multilayered feed-forward / Multilayer Perceptron -

Adv → • complex decision making → create multiple layers of perceptron.
• intricate info is too complex for layer to handle
• multilayer go beyond limitation of single layer.
• atleast one layer b/w input & output layers.




Radial Basis Function Network (RBFN) -

Adv → • Fast learning • High efficiency • Good generalization
• Adaptability • Parametric control • Robustness • Overfitting

Functional Link Neural Network (FLNN) -

Adv → • Reduced complexity • Improved interpretability • Flexibility
• Data efficiency • Computational efficiency

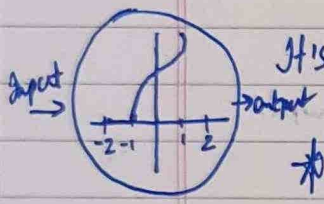
2) Feedback Network -

suggests feedback network has feedback paths.

- Recurrent networks - feedback networks with closed loops
- Fully recurrent network - all nodes connected → all other nodes & each node works as both input & output.
- Jordan Network - closed loop network → output will goto input ↳ feedback.

* Activation Function -

decides whether artificial neural network fire for a given set of inputs.




It's crucible in determing accuracy & computationaly efficient of model.

* Types of activation function -

1) Identify func$^n$ = $g(x) = x$
2) Binary step func$^n$ = $g(x) = 1$ when $x > 0$ otherwise 0.
3) Logistic/Sigmoid = $(s(x) = \frac{1}{1+e^{-x}}$
4) Tan H = Tan H$(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
5) Rectified Linear Unit (ReLu) =
$R(x) = \max(0, x)$

## Perceptron

- simple artifical neural network.
- single layer of processing units → neurons.
- fundamental building block of more complex neural networks.

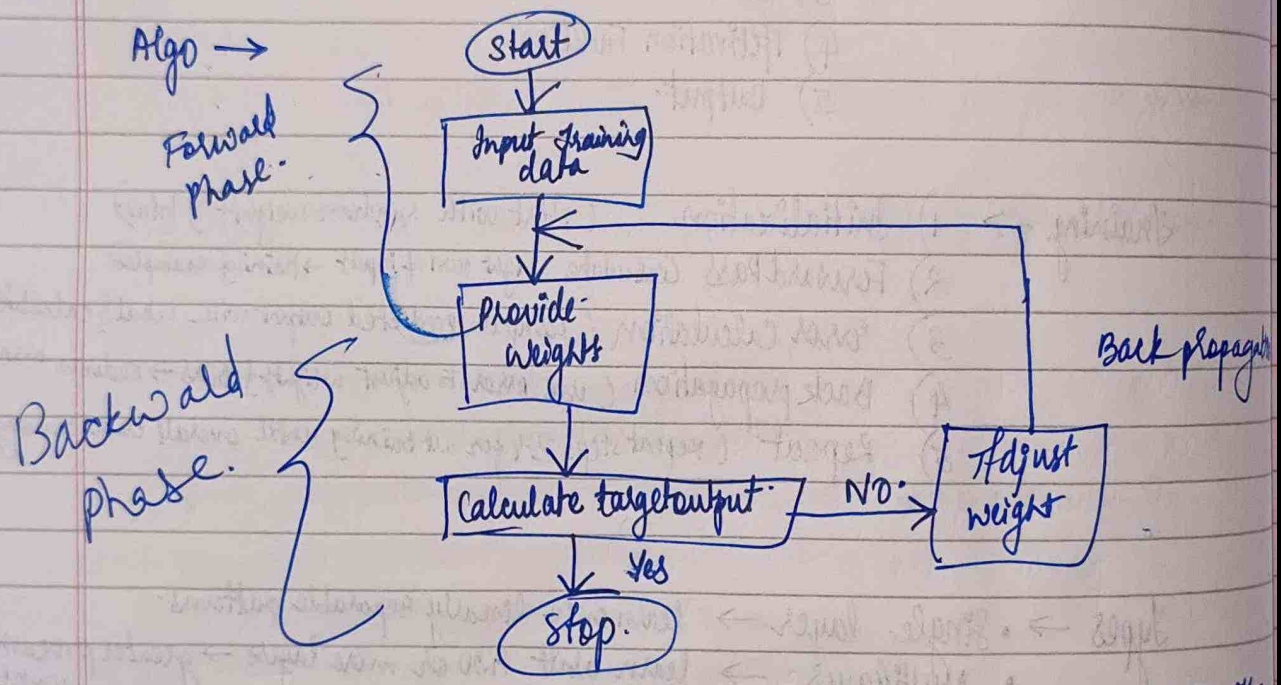
Key components →
1) Inputs
2) Weights
3) Bias
4) Activation Function
5) Output.

Training →
1) Initialization (start with random weights & bias)
2) Forward Pass (calculate weight sum of inputs → training example)
3) Error Calculation (compare predicted output with actual & calculate error)
4) Back propagation (use error to adjust weights & bias → reduces error)
5) Repeat (repeat steps 2-4 for all training until overall error converges)

Types →
- Single layer → learn only linearly separable patterns.
- Multilayer → learn about two or more layers → greater processing power

# Back Propagation -

- algo for supervised learning for ANN
- keeps adjusting weights of connecting neurons with an object to reduce deviation of output signal with target output.
- reach global loss minising backpropagation.
- consists of multiple iterations known as epochs.




Features → gradient descent method → case of single perceptrone network with diff unit
→ weights are calculate in learning period of network.
→ feedforward of input traing pattern.
calculation & backpropagation of error
updation of weight.

Adv → • simple, fast & easy
• only no. of input architecture are tuned, not any parameters
• flexible & efficient.
• No need for user to learning any special function.

Disadv → • Sensitive to noisy data & irregularities
• performance → dependent → data
• too much time taking
• matrix based approach preffred over min batch

* Generalized Delta Learning Rule -
- most common method for training backpropagation network.
- Used to train ANN & MLPs.
- network to learn & adjust its weights & biases based on error it makes.
- extends concept of delta rule used for training single layer perceptron.

Steps → 1) Forward Pass 2) Cost Calculation 3) Propogation
4) Weight Update 5) Repeat.

Benefits →
- efficiently learns complex relationships
- Flexible & adaptable
- Widely used & well studied.

Limitations →
- gets stuck in local minima
- computational cost
- sensitive to initialization.

Powerful training tool for ANNs. but limitations require careful consideration & exploration of alternate methods.

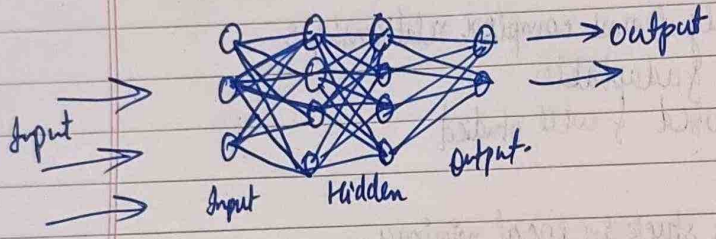
* Limitations of MLP -
- Limited Representation Power
- Local Minima.
- Sensitivity to Initialization.
- Overfitting
- Computational Cost
- Interpretability
- Black Box Nature.

# U-5 (DS Honors)
# Convolutional Neural Network.

## Convolutional Neural Network - (CNN)

- deep learning neural network.
- doesn't require preparation → can operate on raw data.
- feed forward NN with up to 20 to 30 layers

Key features → • Convolutional layers • Activation fuctions
• Pooling layers • Fully connected layers




Adv → • detecting patterns & features in img, audio & video
• Robust to translation, rotation & scaling invariance
• end to end training, no need for manual feature extraction
• handeles large amt of data & high accuracy

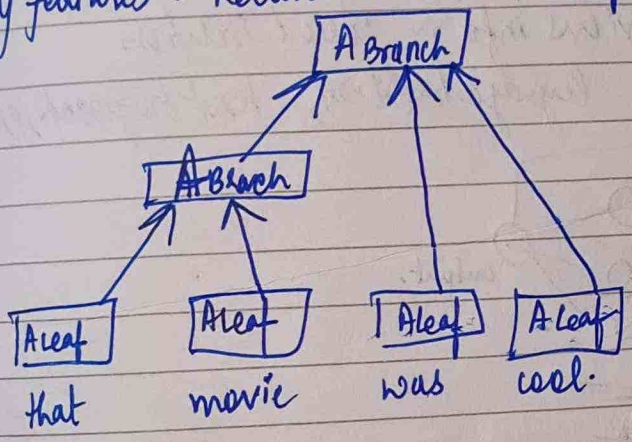
Disadv → • computationaly expensive
• Lot of money needed
• prone to overfitting
• large amt of labelled data for training
• Interpretability is limited, hard to understand what network has learnt.

Appln → Image Recognition
Object detection.
Image Segmentation.
Video analysis.
Medial image analysis.

# Recursive Neural Network (RvNNs)

- Type of ANN that handles hierarchial, structed data like NLP, parse tree etc
- Excel at tasks that involve understanding relationship & dependencies.

Key features → • Recursive structure • Compositionality • Gated activation fuction



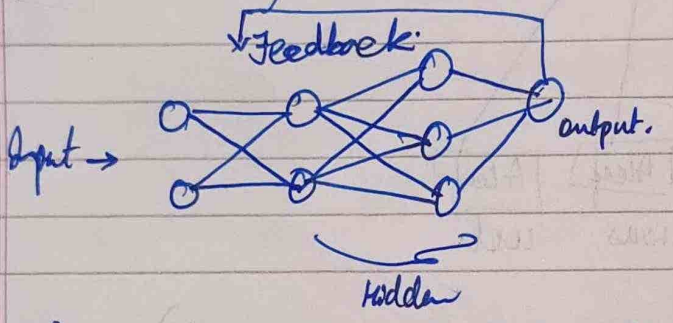

Adv → • NLP → (Text summarization, machine transln).
• Code analysis & generation → (fuctionality of code, identify bugs)
• Biological sequence analysis → (analyzing biological sequence like DNA)

Disadv → • Training Complexity
• Internal decision making
• Data sparsity (large amt of training data)

Applⁿ → • Sentiment analysis
• Machine Translation
• Text Summarization.
• Code generation
• Protein structure prediction

# Recurrent Neural Network (RNN)

- Feed Forward Neural Network
- Preceptors are arranged in layers, hidden layers are not connect with outside world.
- all nodes fully connected, some layers aren't
- need to access previous info in current iteration
- commonly used in language translation, text to speech, etc.




Adv → • remember each info throughout time
• useful in time series prediction → feature to remember input
• extend effective pixel neighborhood
• handling variable length sequence
• Generative Modelling

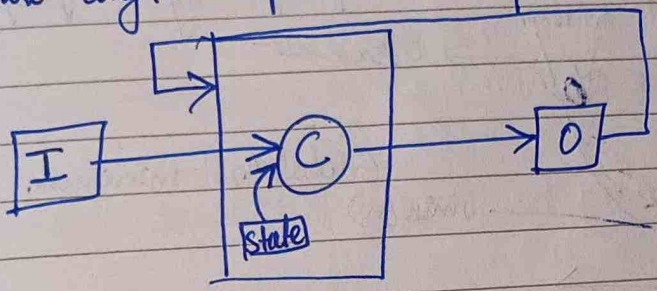
Disadv → • gradient vanishing & explode problems
• ~~training~~ is difficult task to train
• can't process very long sequence if using tanh or relu as activation fuch.

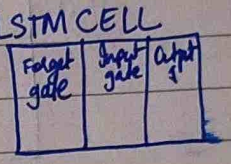
Applⁿ → • NLP • speech recognition • time series forecasting
• Music generation • Image captioning

Types of RNN → • one to one
• one to many
• many to one
• many to many

# Long Short Term Memory (LSTM)

- advanced RNN
- sequential network.
- capture long term dependencies in sequential data



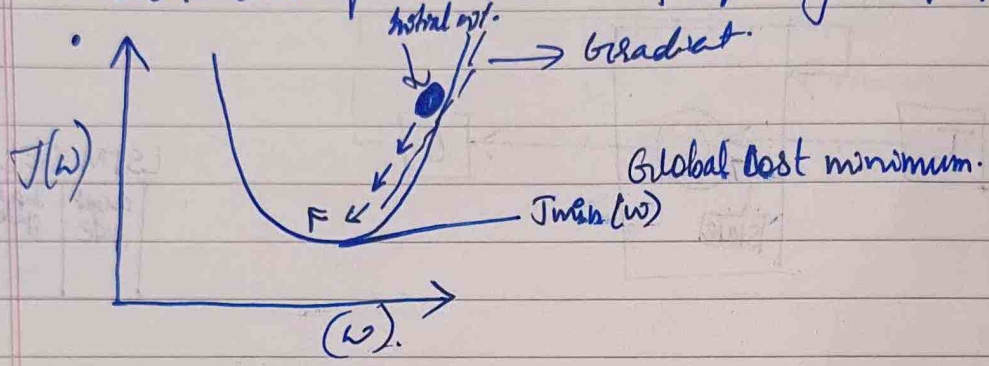

Adv →
- Capture long term dependencies
- Overcomes vanishing gradient problems.
- Wide range of appln
- Flexible architecture.
- State of art perfolmace (leading result in various tasks).

Disadv →
- computationally expensive
- Less interpretable
- Hyperparameter tuning
- Large data requirements
- Susceptible to overfitting

Appln →
- Speech Recognition
- Reccomender System
- Video Analysis
- Time Series
- Language Modelling.

# * Gradient Descent -

- finds best fit line for giving training dataset.
optimization algo → min. cost func
- locate least possible value → fulfil a given fee function.




Working → 1) Initialization 2) calculate gradient
3) Update parameters 4) Repeat (until loss func converges)

Benefits → • Simple & easy to implement
• Flexible & adaptable
• widely used & studied.

Limitations → • Might get stuck at local minima
• sensitive to initialization
• Expensive cost
• Tuning Parameters.

~~Optimization~~ →

Types → 1) Batch → process all training examples for each iteration
2) Stochastic → process 1 traing eg for per iteration - Faster than BATCH
3) Mini Batch → More examples out of total m training examples are processed per iteration.

# V-6 (DS Honors).

## Appln Perspective.

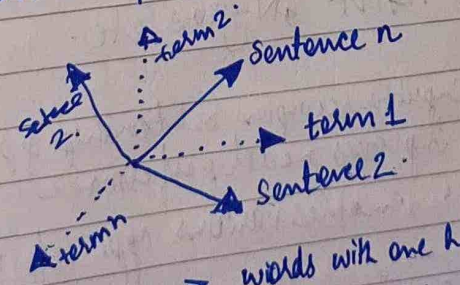
* **Text Preprocessing** → first step of NLP → preparing text data for further processing.

1) **Tokenization**
   - → process of dividing text into meaningful pieces.
   - → pieces are known as tokens
   - → whitespace / unigram tokenization → entire text is split into words by splitting them at whitespaces.
   - → regular expression tokenization → regular expression pattern to get token

example → "hi the sky is blue." becomes → ["hi", "sky", "is", "blue".]

2) Normalization → text to consistent format for easier processing
3) Stop word removal → eliminating the, a, is, etc
4) Part of speech tagging → identifies grammatical role of each word
5) Feature engineering → Bag of words, n grams, TF-IDF, etc

* **Document Representation.**
  - In Vector space model → each term/word is an axis/dimension.
  - Text/ document is represented as a vector in multi-dimensional space




Ways → 1) 1 Hot encoding → words with one hot vector i.e we associate each word with index

2) n-grams → n-words sequence, we take n words from sentence & consider it as ↳ estimate following words probability given previous words.

3) Bag of words → uses term frequency & used in sentiment analysis.

* Feature Selection / Extraction –

↓

subset of relevant features from original set based on their importance

Objectives → • Reduce Dimensionality
• Improve performance
• Avoid overfitting

Benefits → Reducing Train. time & computational cost –
→ Improves model interpretable → Focus on feature
→ Prevent overfitting.

Limitation → May discard potentially useful info
→ Require careful selection of feature selection technique

* Feature Extraction –

Creating new features from original set that captures more info aspects of that

Objective → enhances representation of text data.

Techniques → BOW, TF-IDF, N-gram, etc.

Benefits → • Capture complex relationship & pattern in text data
• Improves model performance
• enables various appln of ML algorithms.

Limitations → • Costs high
• Require careful selection of feature extraction technique
• May introduce irrelevant or redundant information.

7.

* Techniques -

1) Bag of Words → Keep frequency count of all unique words & consider it feature.
   Steps → i) Identify unique word from document.
   ii) Find frequency of ~~words~~ unique words from a single sentence.

2) Term Freqparency - Inverse Document Frequency (TF-IDF).

TF → Frequency of each word in document
IDF → Assign lower weight of words that appear more frequency, basically depicts rarity of word in document.

$TF \rightarrow \frac{\text{Term i frequency in document j}}{\text{Total no of words in document j}}$

$IDF = \log_e \left[ \frac{\text{Total documents}}{\text{document with term i}} \right]$

* Topic Modelling Algorithms - Latent Dirichlet Allocation (LDA) -

- LDA → helps extract topics from a given corpus.
- classifies data into documents & words per topic, these are modelled based on dirichlet distributions & processes
- Assumptions by LDA → i) Documents are an mixture of topics
  ii) Topics are an mixture of tokens.

Working → 1) Preprocessing (remove noise, tokenization & text → numerical)
2) Latent variable • each topic → probability distribution over words
   • each document → probability distribution over topics.
3) Generative process → choose topic mixture.
4) Inference → Infer latent variable given observed data.
   (topic distribution) (documents & words).

LDA

* Adv → • Unsupervised
  • Probabilistic
  • Scalable
  • Interpretable.

* Disadv → • Topic coherence
  • No. of topics
  • Overfitting.

Appln → • Document Clustering
  • Recommendation system
  • Topic summarization
  • Trend analysis

* Stemming & Lemmatization

• technique used in NLP to reduce words to their base or root form
• normalize words & reduce inflectional or derivational variations

• Stemming

• process of removing prefixes & suffixes from word to obtain word's base form
• resulting stem may not always be a valid word.
• Way faster than Lemmatization
• ex → original word → fishing, fishes, fished
  stemmed words → fishi, fishe, fish.

Lemmatization

• transform words to their base form or lemma, considering word's meaning & part of speech
• uses linguistic rules & morphological analysis to achieve this
• resulting lemma is a valid word represent base form
• more accurate results
• example original words
  ↳ walking, walks, walk
  lemmatized → walk, walk, walk

# Text Similarity Measures-

- Used to find similar text.
- Google, Quora, Stackoverflow use it to find similar questions.
- Important in NLP for task like information retrieval, document clustering & machine translation.

Feature to consider when choosing text similarity measure
- Task at hand
- Nature of text data
- Computational resource
- Interpretability

Types → 1) Lexical — Jaccard Similarity, Levenshtein Distance (Focus on words & characters, not on meaning)
2) Semantic — Cosine Similarity (Capture meaning, intent, relationship between words)
3) Featurebased — Kernals, Tree Kernals (Utilize extracted features like structure, topics)
4) Neural-Network based — sentence BERT, Siamese Network (Learn complex semantic relationship & capture non-linear similarities).

Appln → Info Retrieval, document clustering, machine translation, plagiarism detection, chatbots etc

* Jaccard Similarity → Ratio of common words to total words. (no imp to duplication of words)

$$JS = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

* Cosine Similarity →

$$\text{Similarity}(A, B) = \frac{A \cdot B}{|A| \times |B|} = \frac{\sum (A_i \times B_i)}{\sqrt{\sum A_i^2} \times \sqrt{\sum B_i^2}}$$

- suitable → words → repeated & important
- ratio of dot product of two vectors of words to their product of magnitude.