MOD-VII

① Linear Discriminant Function Based Classifier.

→ to classify data points by drawing a str. decision boundary (hyperplane) that separates classes.

Linear Discriminant function (LDF)

to assign a class label to an input vector x using a linear function.

Function form :- $g(x) = w^T x + w_0$

w : weight vector
x : input vector
w_0 : bias (threshold)

Decision rule (for 2-class problem):

If $g(x) > 0$, assign to class 1
If $g(x) < 0$, assign to class 2

Ex:- If i/p has 2 features (x_1, x_2), the decision boundary is:-

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

→ This is the eqn. of a line.

Properties → Simple & fast
→ can't model non-linear reln.
→ works best when data is linearly separable.

Simple mathematical tool, used to classify data into diff. categories by drawing a St. line (2D) or a flat plane that seperates the data.

Ex:- You have 2 types of fruits - Apples
Bananas

Each fruit has 2 features - weight, color.
[W]e can plot them on a graph. If a str. line can [s]eparate apple from bananas then you can use [li]near discrimina...

② Perceptron

- Simplest type of neural network
- One of the earliest algo. used for classifi.ⁿ
- tries to mimic how a human neuron works & is mainly used to classify data into 2 grps. (binary classification)
- A perceptron takes I/P, multiplies them with weights, adds them up, and then decides whether the O/P should be 1 (yes) or 0 (no).
- formula :- $\text{Output} = \begin{cases} 1 & \text{if } (w \cdot x + b) > 0 \\ 0 & \text{otherwise} \end{cases}$

where :- x = I/P features (ht., wt...)
w = weights (imp. of each I/P)
b = bias.
w·x = weighted sum of I/P.

- Perceptron draws a st. line (2D) to separate 2 classes.
- Ex :- Want to separate cat (1) from Dog (0)
Based on their height & weight.
Perceptron will try to draw a line on a graph so that
. cats are on one side
. Dogs are on other side.

Adv.:-
simple
fast
works well for linearly separable data.

Limitations
- Only works if data can be separated by a st. line
- Doesn't handle multi-class problems well.

(3) Support Vector Machines (SVM)

- powerful ML algo.
- used for classific^n. & regression.
- mostly used to classify data into 2 classes. (Spam or not spam, cat or dog)
- tries to find the best boundary that separates 2 classes with largest possible gap b/w them.

- Ex:- Imagine you have 2 types of shoes
  - Sports shoes
  - Party shoes.

You want to separate them on a graph based on 2 features: color & heel size.
SVM draws a line b/w the 2 types but not just any line.
It picks the best possible line that is far away from both classes.
That way, new shoes can be placed on the correct side of the line.

Adv.

- Effective when clear margin of separation exists.
- Can be used for both linear & non-linear classification.
- works well with high-D. data.

Disadv.

- Slow with large datasets.
- Hard to choose the right kernel sometimes
- Doesn't perform perform well if classes overlap a lot.

Real-life uses of SVM:

- Handwriting recog.
- face detection.
- Spam email filtering
- Bioinformatics-

MOD - VIII

① Non-metric methods for pattern ~~recogn~~ classification

Non-metric methods

→ Classifⁿ methods that don't rely on dist. or geometric measurements.

→ Not based on metrices like Euclidean dis.

→ Useful when data is non-numeric.

→ Nominal data - data that represents categories not numbers.

→ Ex:- Colors: {R, G, B}
  Weather: {Sunny, Rainy, Cloudy}
  Gender: {Male, Female}
  Blood Types: {A, B, AB, O}

→ Why Metric methods don't work?
You can't say "Red" is closer to "Blue" than "Green".
Thus non-metric method is needed.

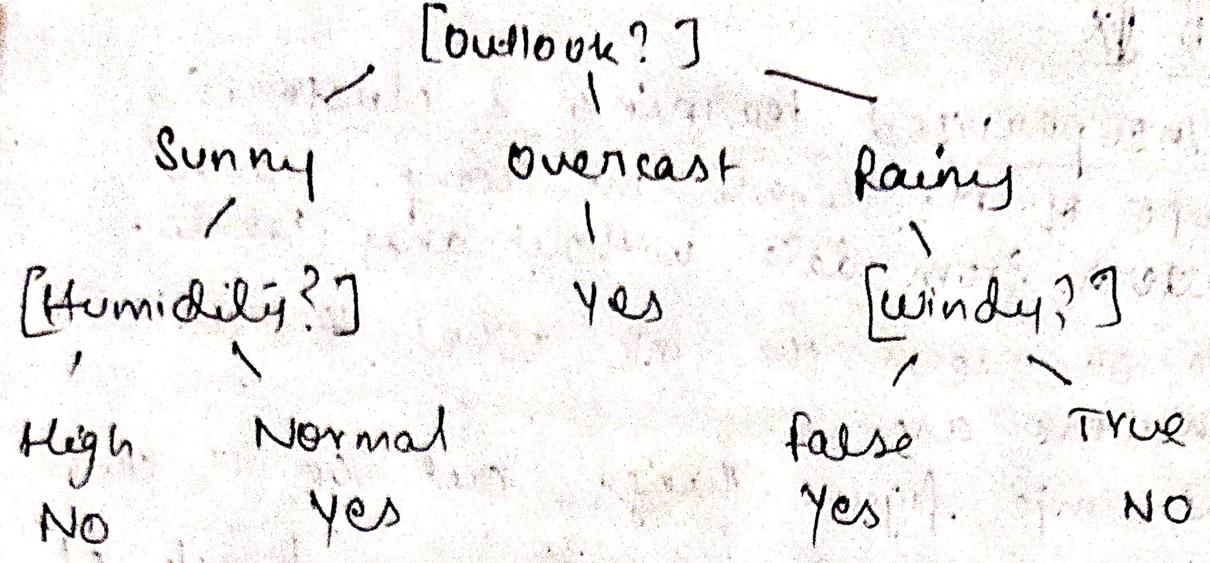
② Decision Trees

A tree-like str. where:

- Internal nodes represent decision rules on attributes.
- Branches rep. outcomes of decisions.
- Leaves rep. class labels (final decision)

Ex:- For a weather Dataset:-

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



Adv.

- Easy to visualize & interpret.
- No need for feature scaling
- can handle non-numeric & categorical data directly.

Disadv.

- can become complex if not pruned.
- Sensitive to small changes in data
- prone to overfitting.

MOD. IX

① Unsupervised learning & clustering

- Type of ML where the comp. learns pattern from data without any labels.
- No one tells the m/c what the correct answers are.
- The m/c figure things out on its own.
- Ex:- Imagine you ~~have~~ given a bunch of photos of animals, but no one tells you which one is a dog, cat, or rabbit.

Your task is to group similar-looking animals together. That's exactly what unsupervised learning does!

- Unsupervised Algorithms
  - K-Means clustering
  - Principal component Analysis (PCA)

Clustering:-

- an unsupervised learning method where we group similar data points into clusters.

Ex:- Given customer data (age, income), clustering can grp. them into segments like:- Low income, young age
High income, middle age
Retired, fixed income.

Application → Market segmentation
→ Doc. / Image classification
→ Anomaly detection.

② Criterion functions for clustering

- technique in unsupervised learning
- where we grp. similar data points together into clusters.
- Ex:- Grouping students by similar marks.
  Grouping customers based on shopping habits.

- Criterion funⁿ. → like rule / formula
  → helps comp. measure how good a clustering is.

goals → points in the same cluster should be close together.
→ points in diff. clusters should be far apart.

common criterion functions:- → Intra-cluster Distance

• Within-Cluster Distance (WCD)
- measures how close the points are inside each cluster.
make this as small as possible.
smaller value - better grouping.
Ex - group friends by hobbies.

→ Inter-cluster Distance

Between-Cluster Distance (BCD)
- measures how far apart the clusters are from each other.
- Goal - make this as large as possible.
- Bigger dist. → better separation b/w clusters.

- Ex - you want grp. of stu. with very diff. study habits in separate clusters.

- Sum of squared errors (SSE)

- Adds up the sq. dist. of all points from their cluster center.
- minimize SSE
- used in K-Means clustering

(3) Algorithms for clustering.

A) K-Means clustering

1. Choose a no. of clusters K
2. Initialize K random centroids
3. Assign each data point to nearest centroid
4. Update centroids by averaging points in each clusters.
5. Repeat steps 3-4 until convergence.

→ unsupervised learning algo.
→ used to grp. data points into 'K' clusters based on how similar they are
→ like putting similar items into same box.
→ Imagine you have a bag of colourful balls & you want to group them by color, but the balls are not labeled.
→ K-means will automatically find groups based on color without you telling it.

→ K = no. of clusters you want
you choose this no. before the algo. runs.

For ex: K = 3
→ it will make 3 groups from the data.

→ works?

1. Choose k
2. Place k random points on data
3. Assign each point to nearest centroid
4. Move the centroids to the centre of their assigned points
5. Repeat 3 & 4 until centroids don't move much.

→ Real-Life use → Image Compression.
→ Market Segmentation.

(4) Hierarchical Clustering

→ method of unsupervised learning
→ grp. data into tree-like str. called dendogram.
→ doesn't need you to choose the no. of clusters at the start like k-Means.
Instead, it builds a hierarchy of clusters - from small to big or big to small.

→ Ex - Organizing a family tree.
You 1st grp. close relatives
Then you merge them into bigger family groups
Eventually you get 1 big family tree.

Other methods

1> Density Based Spatial Clustering of Applications with Noise.

→ Groups points are are closely packed

2> Mean Shift

3> Gaussian Mixture Models (GMM).