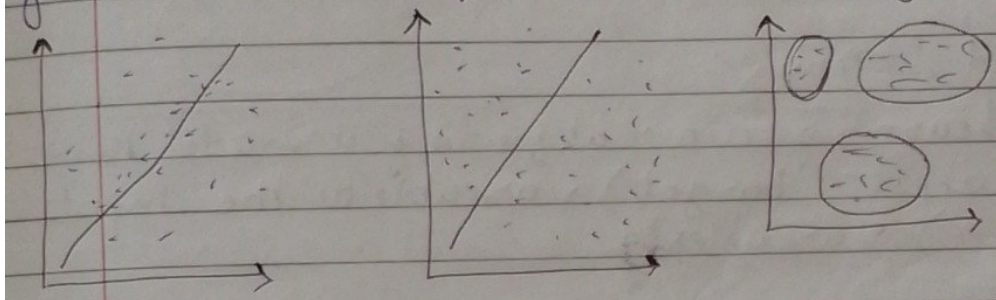
Unsupervised learning

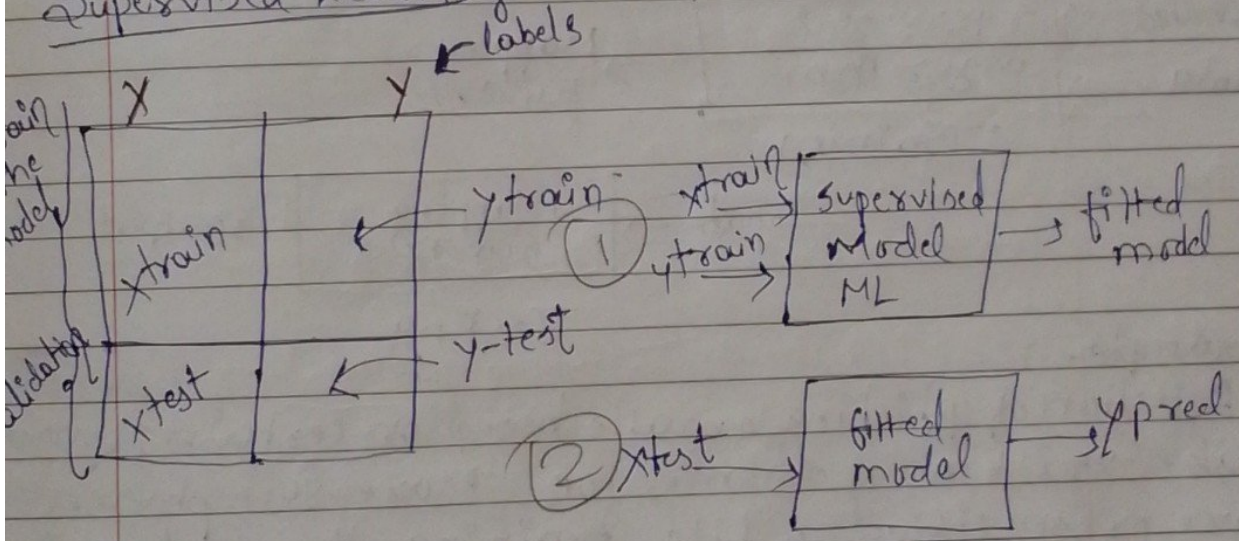
Date / /
Page No.

Machine learning

- Supervised Learning
  - Regression
  - Classification
- Unsupervised Learning
  - Clustering
  - Association



Supervised learning

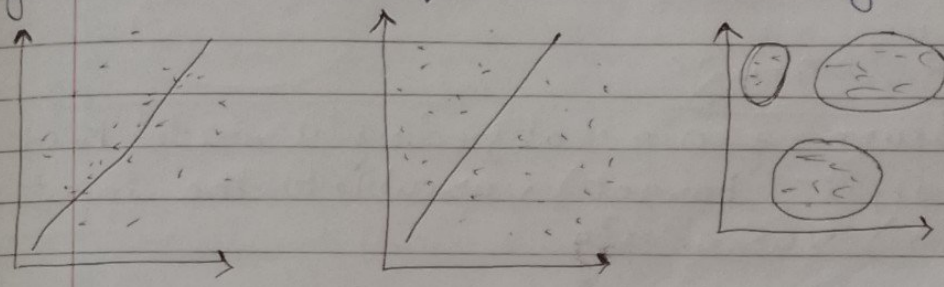


Unsupervised learning

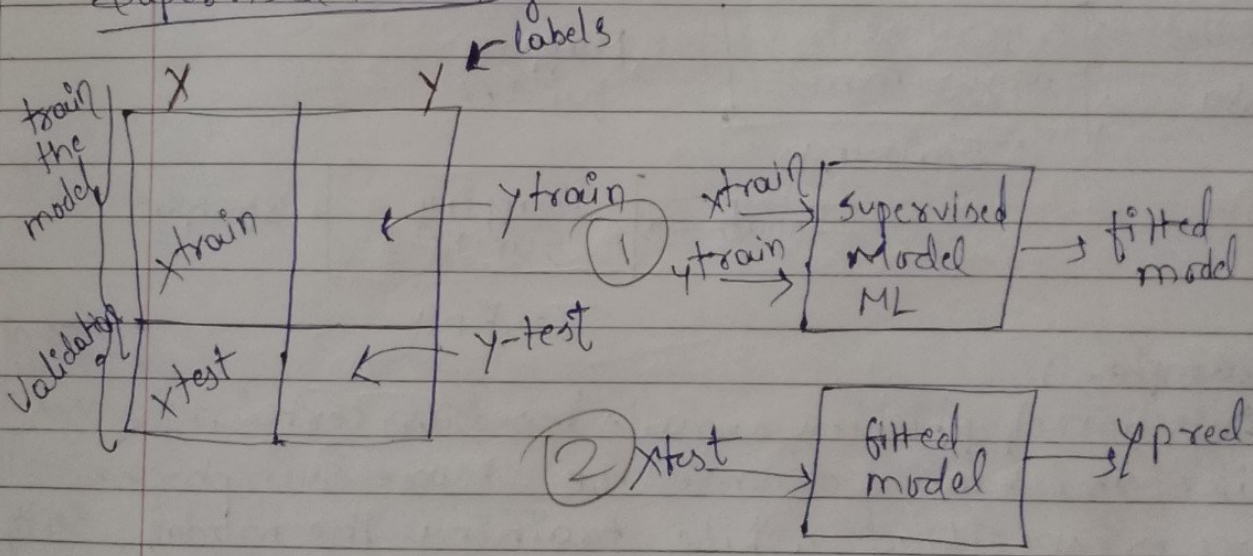
Date / /
Page No.

Machine learning

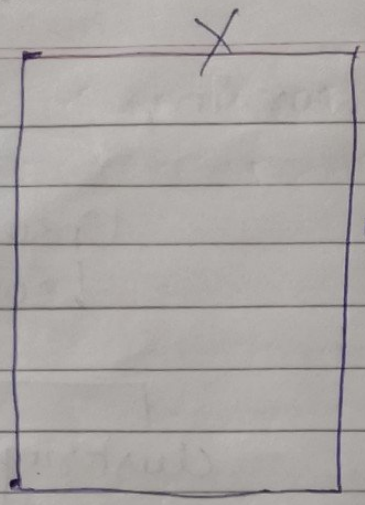
- Supervised Learning
  - Regression
  - Classification
- Unsupervised Learning
  - clustering
  - Association



Supervised learning



Unsupervised Learnings

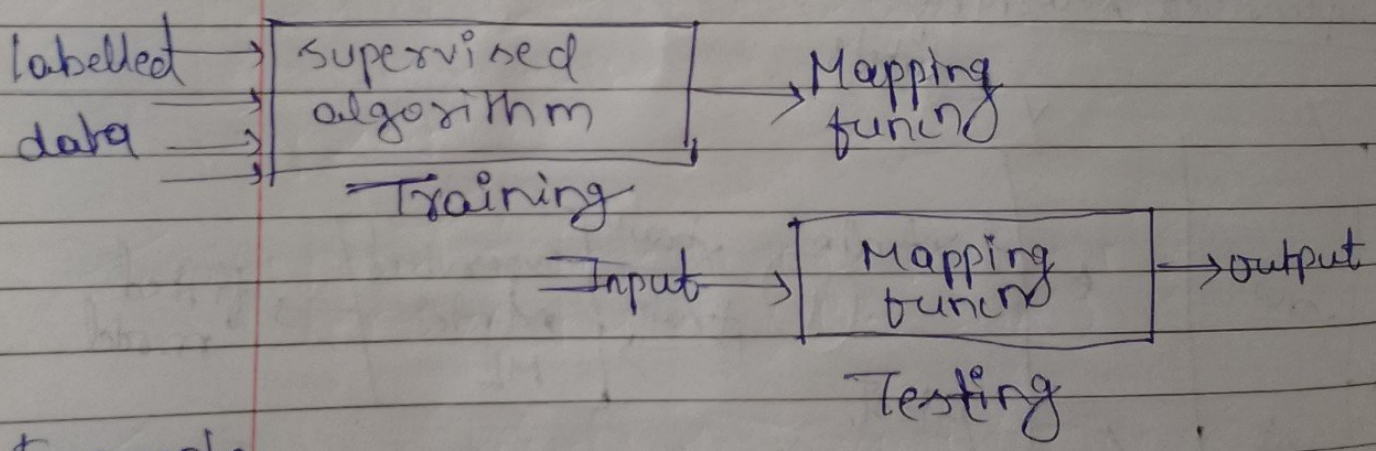


Do not have any label.

objective : To give labels to the data

Supervised learning

Supervised learning aims at finding a model that maps the output (target) variable to the input (predictor) variables



Example.
Detection of phishing emails based on certain phrases like 'you have won million'. More such phrases are prespecified while training the model. So if a new email also contains a similar phrase such emails can directly be tagged as spam.

Shot on AWESOME A05s

Unsupervised Learning

Date / /
Page No.

- Unsupervised learning aims to learn more about the given data

The data used for unsupervised learning has no labels, i.e. there is no desired outcome or correct answer given.

Example
Consider a dataset with information about flowers
We know the data has records of flowers and their different characteristics
Using unsupervised learning, we groups flowers with similar characteristics and try to find if they belong to a particular species.

Bussiness example : group the data.

Consider the data of flowers petal length and petal width in millimeters for different flowers.

Petal length	9.5	17	16.5	4.8	11.5	5.8	10	4.6	15.5	13	5.1	12
Petal width	7.5	15	15	8.4	10	8.6	9	8.8	14	12	9	11

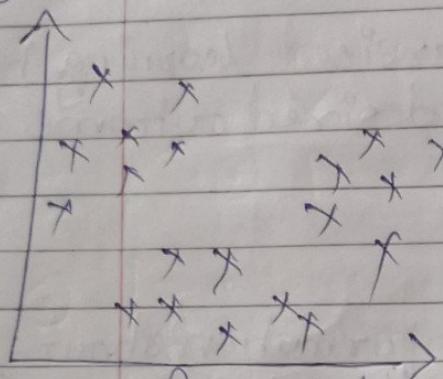
Can we find the data that belongs to the same kind of flower ?
Why Clustering ?
Yes, it is possible to find the data that belongs to same kind of flower.
However, it is not possible to name the type of flower, since no information on its 'label' is available.

Shot on AWESOME A05s

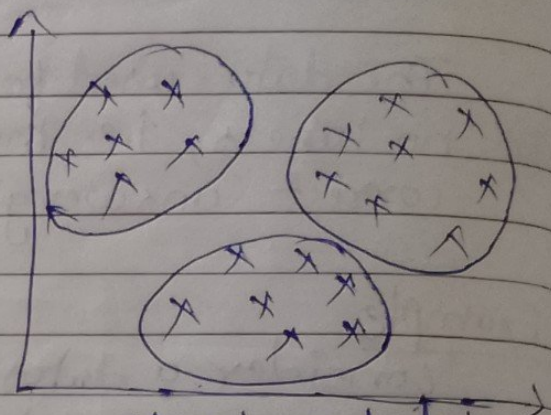
Clustering

Date / /
Page No.

clustering is a technique in which data is grouped based on the similarities in them



Raw data



clustered data.

* clustering partitions the data into natural groups such that
  - ✓ (*) Points in the cluster are as similar as possible
  - ✓ (*) Points across the clusters are as dissimilar as possible.

① ~~Density~~ based : Interchange Partition Based with Density based (DBSCAN)

② ~~Hierarchical~~ based: Hierarchy of clusters are formed based on the distances between the observations (Hierarchical clustering)

③ Graph based : Clusters are formed either by dividing a set of graphs or dividing the nodes of a graph (K- Spanning tree)

④ ~~Partition~~based: Obeservations are partitioned into predetermined number of clusters based on distance from centroids (K-means clustering)

Shot on AWESOME A05s

Model based : Assumes that the data is a mixture of distributions and tries to fit a model such that each distributions represents a cluster (Gaussian Mixture Model)

Clustering approach.

Clustering Approach

Hard clustering	Soft clustering
Each point is assigned to only one cluster	The probability each cluster for each of points is obtained
e.g. Kmeans clustering	e.g. Gaussian Mixture Model

Classification V/s Clustering

- In the classification problem, the target variable is known and is categorical.

  A model is trained on this information and new data is classified accordingly

  In clustering groups of data are formed based on similarity in observations.

Shot on AWESOME A05s

Proximity Measures

Date / /
Page No.

- The proximity measures find the distance between two instances
- Proximity measures include
  - Similarity measures
  - Dissimilarity measures
- Depending upon the data types, we choose the proximity measure.

Distance Measures for numeric data

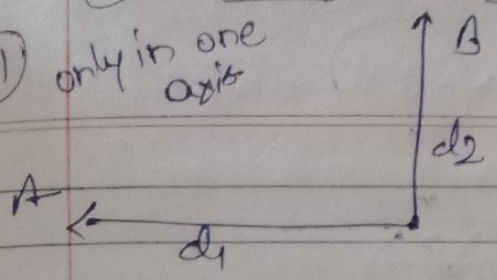
The x_i and y_i are the values taken by variables X and Y respectively

Distance Measures	Formula
Euclidean dist.	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
Squared dist.	$\sum_{i=1}^{n} (x_i - y_i)^2$
Manhattan dist.	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Minkowski dist.	$\sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p}$
Chebyshev's dist.	$\max_{i=1}^{n} \lvert x_i - y_i \rvert$

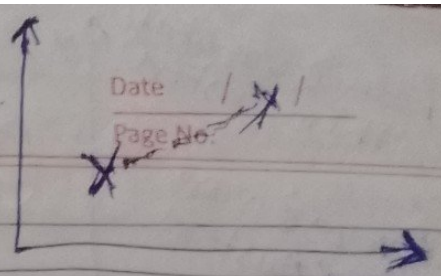
Shot on AWESOME A05s

(Manhattan distance)

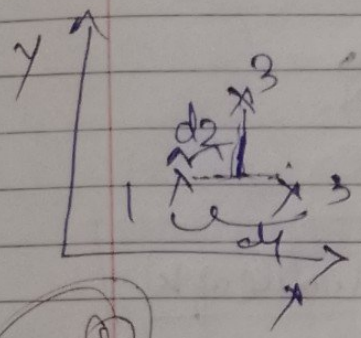
(P=1) only in one axis



(P=2)



$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$



$D_{12} = d_2 + d_3$

$D_{13} = d_1$ $\quad d_1 < d_2$

Session 2

day 1

"K-means algorithm"

K — specifies the no. of clusters

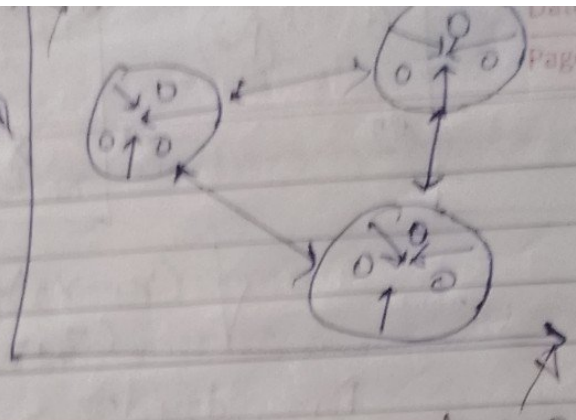
means — specifies avg. (centroid) of a cluster

- Used when data is numeric
- Recursive technique
- Can not train a model
- Based on proximity measures
- Greedy algorithm
- Minimizes the squared error of points in its cluster
- Non deterministic algorithm

Shot on AWESOME A05s

① Reduce variation within clusters

② Increase variation b/w clusters



K-means algorithm - procedure

[choose a distance measure and value of k]

↓

[Initial centroid assignment]

↓

[calculate dist b/w centroids and observations]

↓

[Reassign the cluster based on distance]

↓

[calculate centroids of newly formed cluster]

↓

[Repeat the steps 3, 4, and 5, until the cluster centroids do not change]

① lets choose k=3 और k=3 choose करलिया and ② step 2 it randomly assign 3 centroids in the data because we choose k=3 ③ हर point का, हर centroid से distance निकाला जाएगा ④ जिस से centroid से min dist. होगा उसका उसका उस centroid के साथ cluster बन जाएगा ⑤ फिर अब जो cluster बना है, उस single cluster में fix किया जाएगा, फिर new cluster का उसका, same हमा दोनों cluster के साथ होगा ⑥ again new centroids को हमारा हर data पर

Initial Centroid assignment methods

Forgy Main मेरा k = 3 लेंगे तो उनही datapoints में से कोई 3 data point लेके centroid assign कर देता and Random Main Randomly देता है।

Two methods for Initializing cluster centroids:

1. Forgy
2. Random Partition

Forgy method assigns k random observations as cluster centroids for k clusters

In Random Partition method, a cluster is randomly assigned to each data point and the mean of data in each cluster is considered as Initial cluster centroid.

K-means algorithm

1. For the k means algorithm, we mostly prefer the forgy method to initialize the cluster centroids
2. The centroids initialization step affects the formation of final clusters.
3. The clusters centroid represents the center of a clusters
4. If the data is grouped using 'n' variables, then the cluster centroid is an n-dimensional vector representing the average of all the observation for each variable.

Shot on AWESOME A05s

// Pythons sklearn library provides the k Means to cluster the data in pre-specified number of clusters //

```
#import the function
from sklearn.cluster import KMeans

# build a k-Means model for a specific value of k
# 'random_state' preserves the cluster
    labels over multiple code runs
model = KMeans(n_clusters=k, random_state=1)

# fit and predict the cluster labels
model.fit_predict(data)
```

Data Scaling

Consider a data with 3 features, of which 2 features have a small range (say between 0 to 25) and the third features ranges from -100 to 2000. The clustering would be majorly based on the feature with a high range. Since its contribution to the distance measure would be high with very little or no effect of the other variables; thus we scale the data such that each feature will have equal weight

Merits

- Easy to understand
- Simple Implementation

Date / /
Page No.

Demerits

- Finding the optimal value of k can be computationally expensive
- Initial centroid assignment affects the final output
- Not efficient in presence of outliers

Notes

- In case of categorical data use k-modes algorithm
- For mixed (numerical and categorical) data type use k-prototypes algorithm

Optimal value of (k)

In the previous example, it was obvious that the observations were divided into 3 groups thus we consider k = 3.

But in general, it is not easy to decide the optimal value of k

In this session, we study two common techniques to find the optimal value of k :

One of the methods is using the Elbow plot and the other one is the Silhouette Methods

Shot on AWESOME A05s

Elbow Method

K-means clustering aims to reduce the within-cluster variation

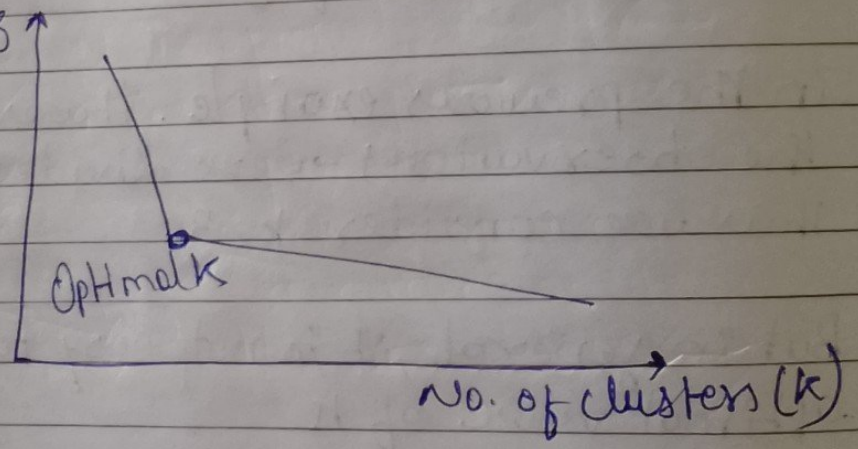
The elbow (or scree) plot is used to plot the within cluster sum of squares (WCSS) for different values of K

Optimal K is the value corresponding to the elbow point.

$$WCSS = \sum_{C_j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

where,

C_j is the jth cluster and μ_j is the centroid of the jth cluster



It is used to measures the variation within the cluster.

Higher the value of WCSS **Higher** is the variation within the cluster.

lower the value of WCSS **lower** is the variation within the cluster.

~~WCSS → जैसे अपने पास तीन cluster हों, जो final 3 cluster में जहाँ data points stops changing their positions from cluster to cluster~~

final cluster का तीनों centroids 1,2,3 का centroid का sum and square distance centroid से centroids and adding all 3 we get the WCSS.

Shot on AWESOME A05s

- The plot shows that the wcss is decreasing rapidly for k-values less than the optimal k value

Page No.

- After the elbow point, the wcss is steadily decreasing which implies that more clusters are formed by dividing the clusters into subgroups

- Selecting the k-greater than optimal k leads to overfitting

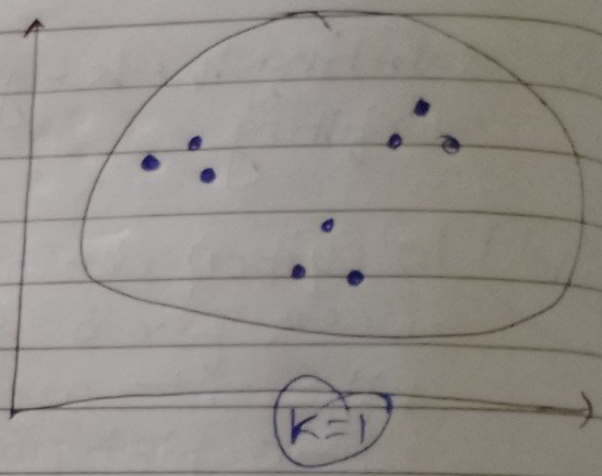
Note: In python, the attribute 'inertia_' returns the wcss for a specific value of k. we use the different wcss and k values to plot the elbow plot.

```
# consider an empty list to store wcss for each k
wcss = []

# perform the k means with different k-values.
for i in list_k:
    model = kMeans(n_clusters = i, random_state=10)
    # fit the model
    model.fit(df_data)
    # 'inertia_' returns the wcss for a specific value of k
    wcss.append(model.inertia_)

# plot the elbow plot
plt.plot(k, wcss)
```

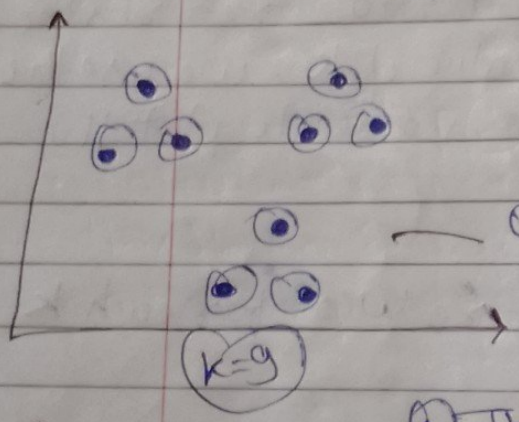
Shot on AWESOME A05s

→

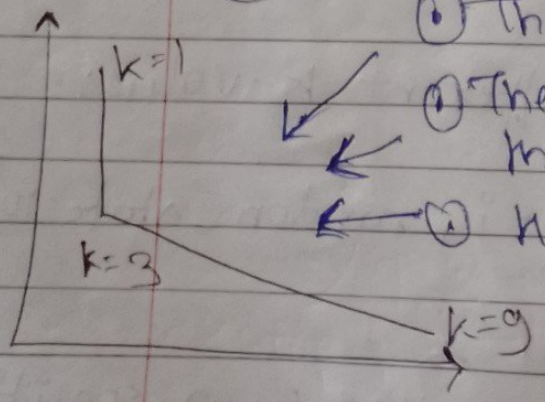
- Suppose there are 9 observations
- The minimum value of k that we can have is one
- The variation within the cluster is maximum as all the observations are grouped inside a single cluster. But our aim is to reduce this variation.



underfitted
wcss is higher



— overfitted
wcss is less.



- The optimal value of k lies b/w 1 and 9
- The elbow plot shows that k=3 is the optimal value
- Here, the variation within the cluster is minimum and the variation b/w the clusters is maximum

Silhouette Coefficient

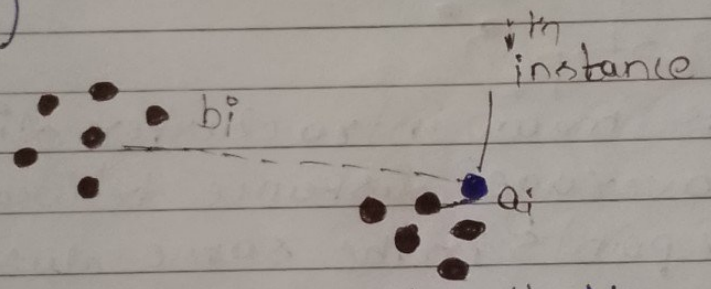
Date / /
Page No.

Silhouette coefficient of a single instance (observation) is given by

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

a_i = mean distance between ith point and other points in the same cluster (mean intra cluster distance)

b_i = mean distance between ith point and points of the next closest cluster (mean inter-cluster distance)

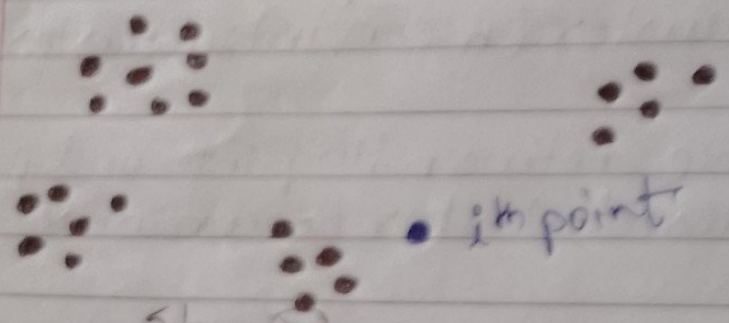


- Silhouette score is used to find the optimal no. of clusters
- It is the mean silhouette coefficient over all the instances
- The value of the silhouette score lies between -1 to +1
- We plot the silhouette score for different values of 'k' and select the k with the highest score
- It is a computationally expensive method
- It is also used to validate the quality of clusters

Shot on AWESOME A05s

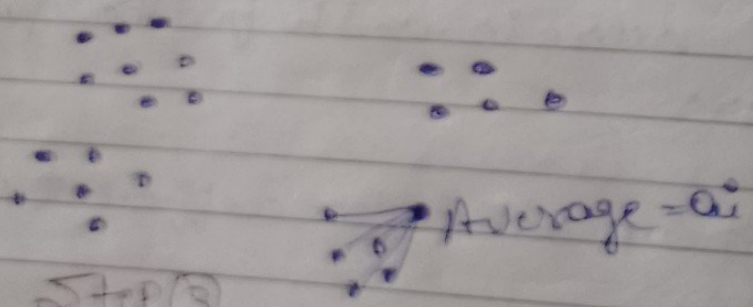
- For each point, we have to calculate the silhouette coefficient

- Lets consider the ith point for which we will calculate the silhouette coefficient



Step 2

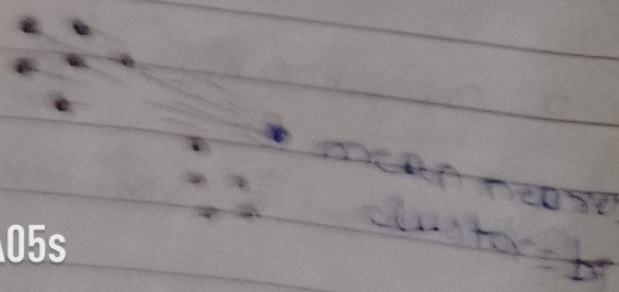
- Calculate mean intra cluster distance (a_i)
- It is the average distance between i^{th} point and other points in the same cluster



Step 3

- For same point, calculate mean inter cluster distance (b_i)

- It is the average distance between i^{th} point and points of the next closest cluster



Shot on AWESOME A05s

Step 4

- calculate s_i using the obtained values of a_i and b_i
- Similarly calculate silhouette coefficient for each observations
- The average silhouette coefficient of all the observations is the silhouette score.

	Qual	gender	State	Course
1	2	M	Delhi	BBA
2	1	F	Delhi	BBA
3	2	M	Delhi	B.com
4	2	M	Pune	B.com

	Qual	gender	State	Course		
1-2	0	0	1	1	2/4	0.5
1-3	1	1	1	1	3/4	0.75

	1	2	3	4
1	1	0.5	0.75	
2	0.5	1		0
3	0.75		1	
4		0		1

Similarity and dissimilarity measures.

A similarity measure for two objects return the value b/w 0 and 1. For the completely similar objects the similarity measure is 1

A dissimilarity measure for two objects return the value between 0 and 1 For completely similar objects the dissilarity measure is 0

Shot on AWESOME A05s

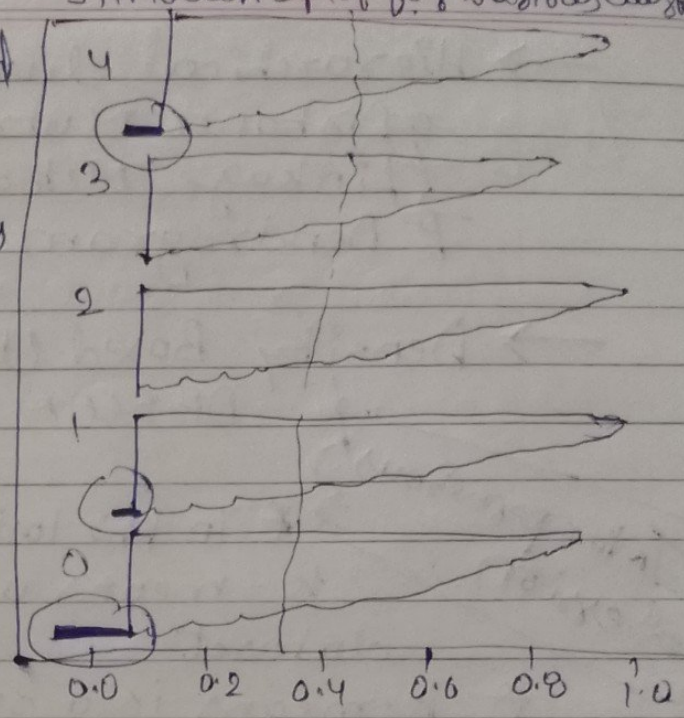
Identify outliers using Silhouette plot

- Any silhouette coefficient that is less than 0 is considered as an outlier

- In this case there are outliers in the 4th, 1st and 0th cluster

- Thus, we can say that the value of k=5 is not a good choice of k

Silhouette plot for various clusters



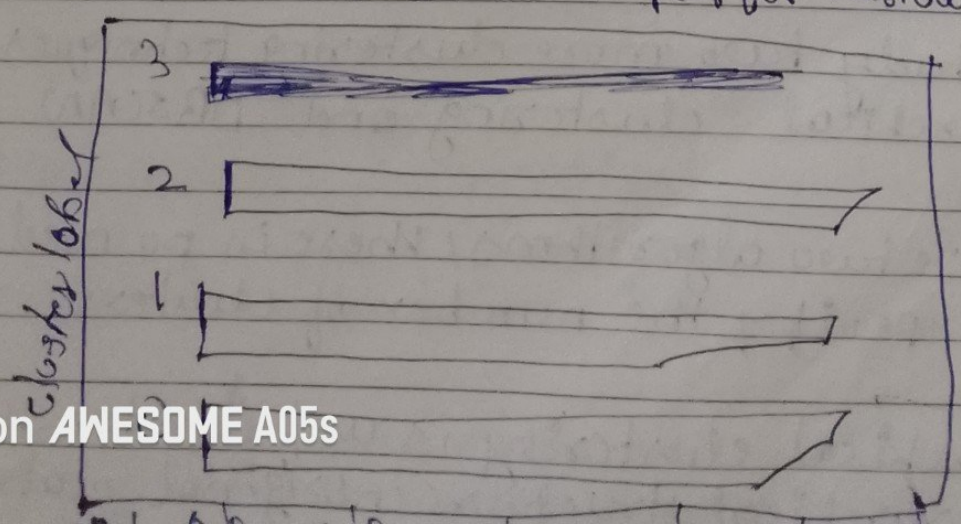
Silhouette coefficient less than average score

- Generally the silhouette score of each cluster should be greater than the average score

- In this figure, the average score is shown by a red dotted vertical line (---)

- The silhouette coefficient for the 1st and 3rd cluster is less than the average score; thus k=4 is not a good choice of k

The silhouette plot for various clusters



Shot on AWESOME A05s

Hierarchical and DBSCAN Clustering

→ Hierarchical clustering
- Distance Matrix
- Linkage Methods
- Dendrogram

→ Density Based Clustering
- DBSCAN

In the previous session

* In the last session, we studied the k-means algorithm to cluster the dataset.
* k-means is a simple algorithm to understand and implement. However, this algorithm has several flaws
* we need to provide the number of clusters (k) to the algorithm, which is not always easy to to determine.
* The initial centroid assignment affects the formation of final clusters
* The algorithm is sensitive to the presence of outlier
* we study two more clustering techniques: Hierarchical clustering and DBSCAN
* For these two algorithms, there is no need to pre-specify the number of clusters
* Hierarchical clustering is useful when there is a hierarchical structure in original data

Shot on AWESOME A05s

* DBSCAN identifies the noise/outliers in the data

* The DBSAN algorithm can form clusters of any arbitary shape.

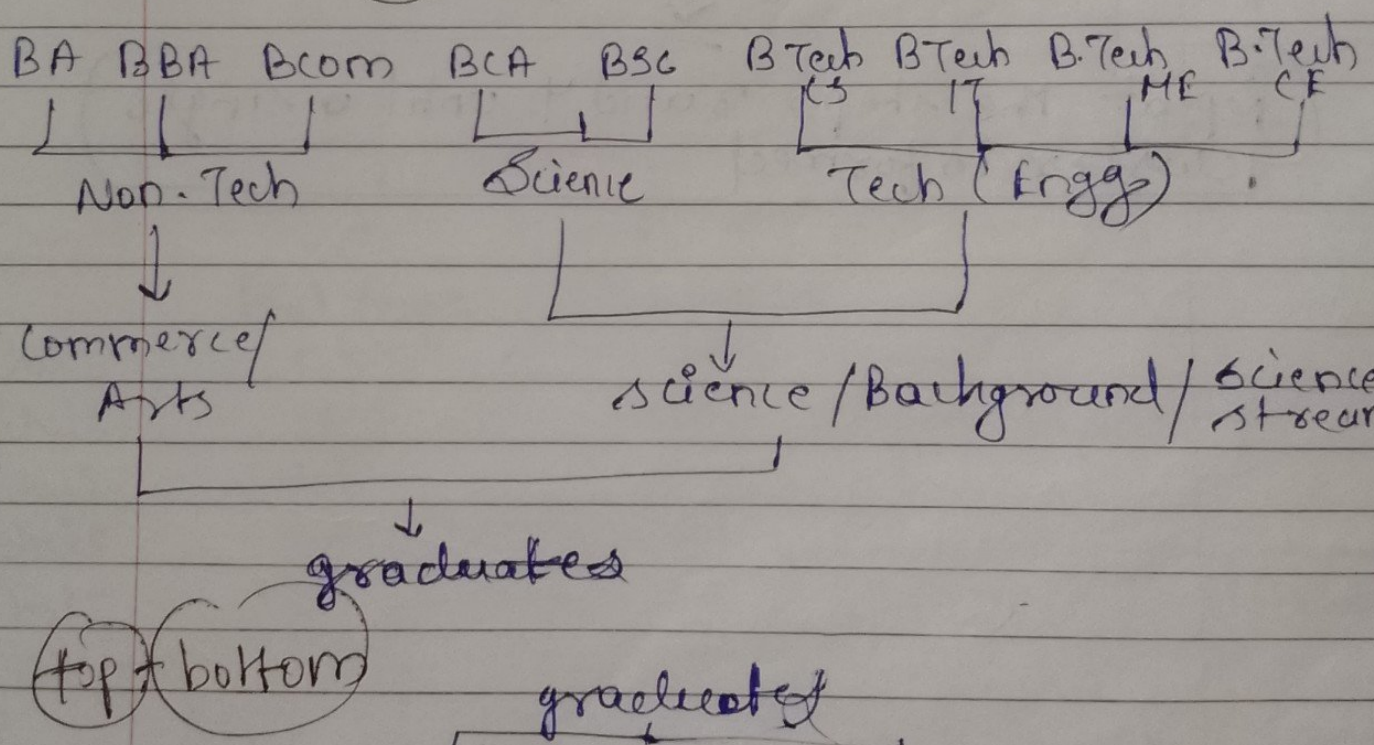
Hierarchical Clustering

* Hierarchical based clustering method.

* The two main types: Agglomerative (bottom to top approach) and Divisive (top to bottom approach)

* No need to pre-define the numbers of clusters

Bottom to top approach



* Most popular hierarchical clustering method.

* It considers the bottom to top approach

* The similar observations are clustered together to form a bigger cluster, considering each observations as a

Shot on AWESOME A05s

* The process continues till all the observations are fused in a single cluster

Date / /

Page No.

Agglomerative Clustering - Procedure

Consider each observation as unique cluster

↓

Calculate the pairwise distance b/w all clusters

↓

Combine the two nearest cluster into a single cluster

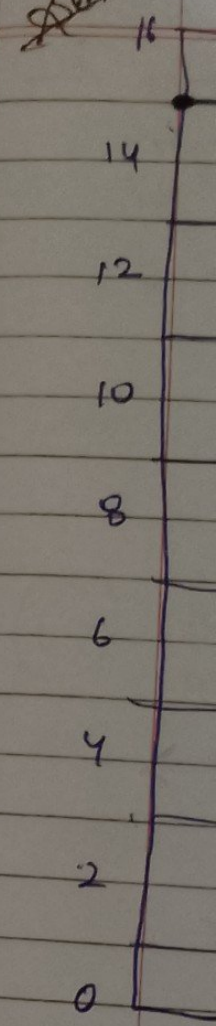
↓

Calculate distance between newly formed cluster and remaining clusters

↓

Repeat the steps 3 and 4 until a single cluster is formed

Dendrogram (Session 13)



Petal1	
Petal2	
Petal3	
Petal4	
Petal5	

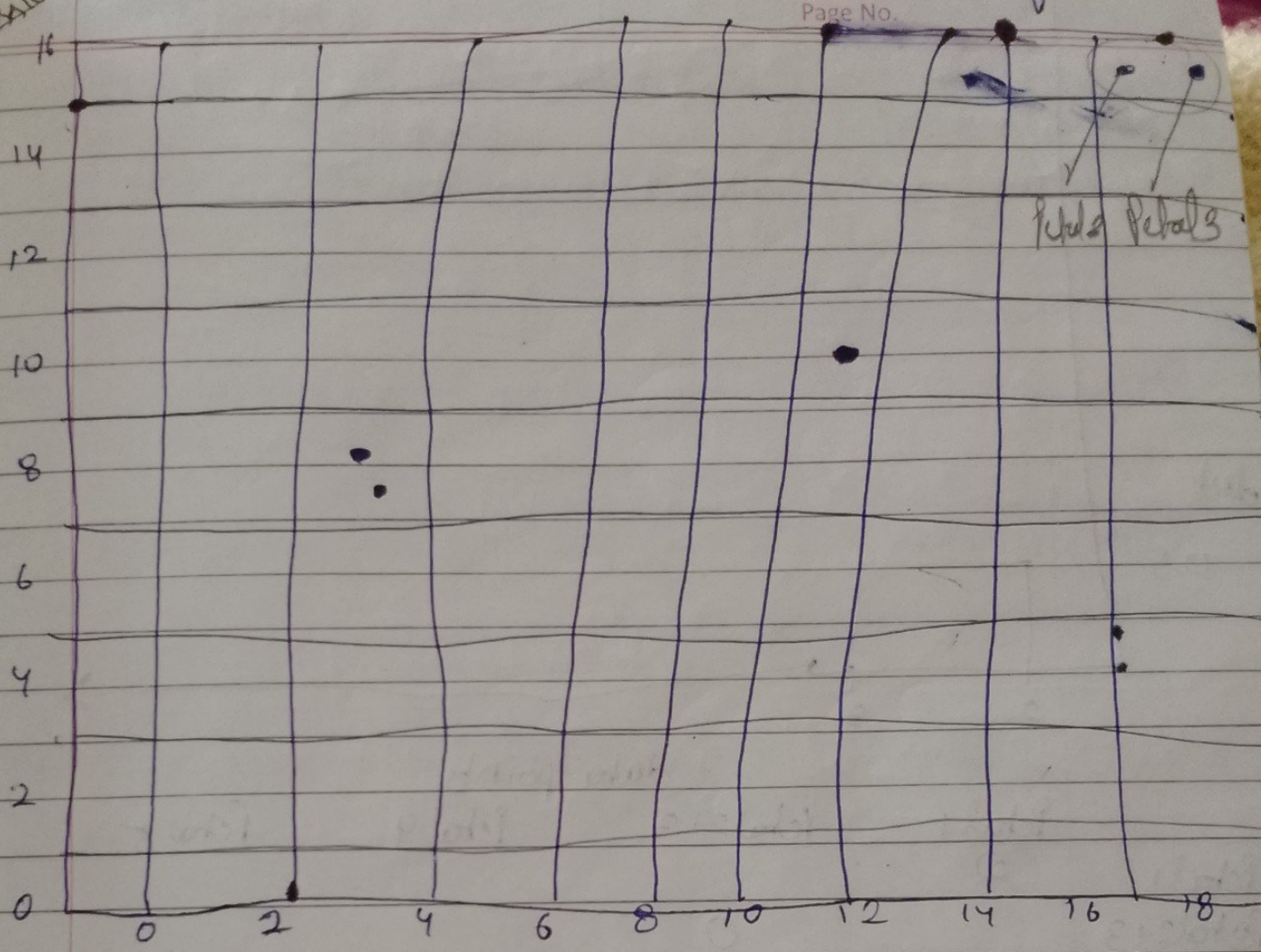
Distance

	Pe
Petal1	
Petal2	
Petal3	
Petal4	6.14
Petal5	6.4

Shot on AWESOME A05s

Agglomerative Clustering

bottom to top



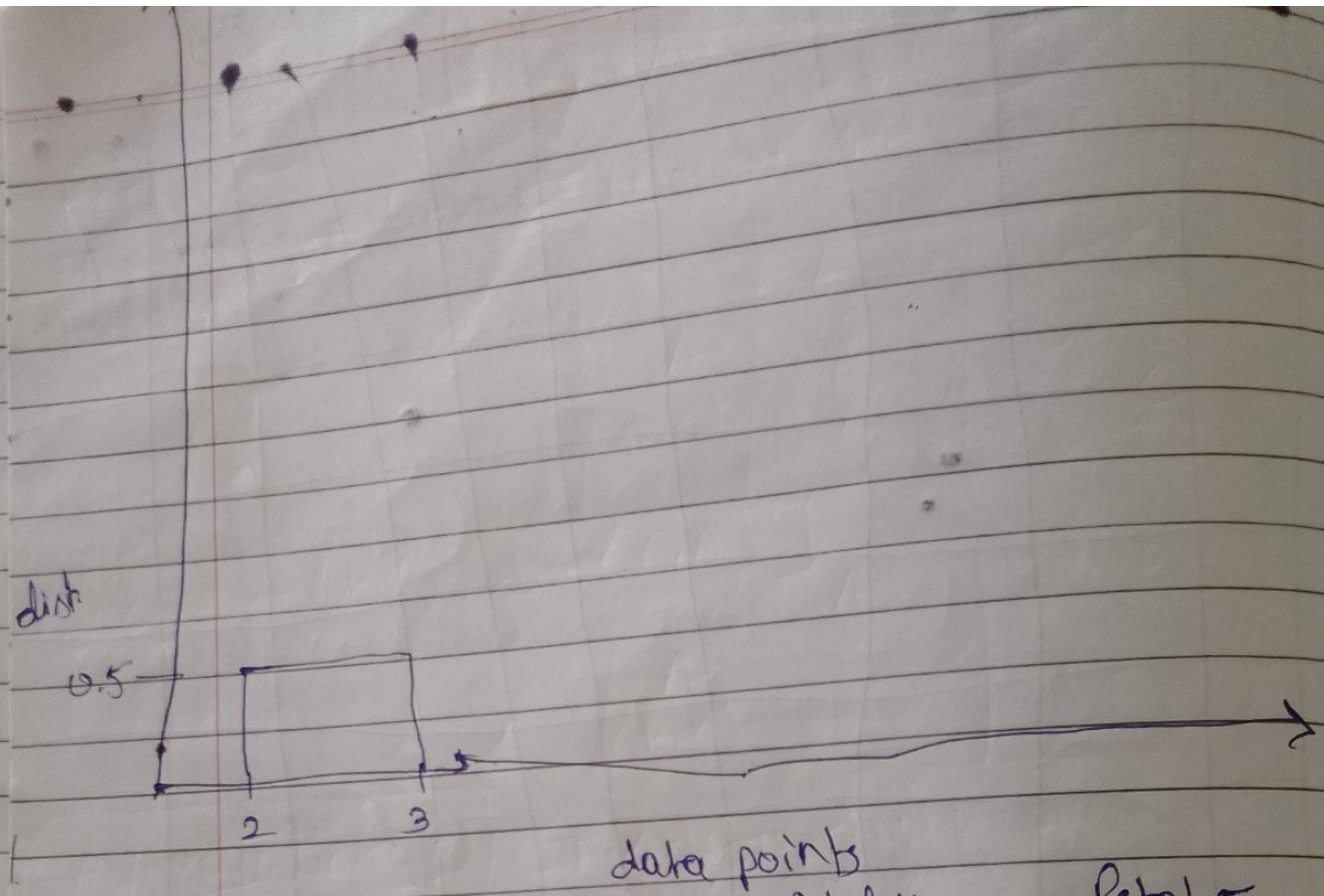
	Petal Length	Petal Width
Petal1	5.5	7.5
Petal2	17	15
Petal3	16.5	15
Petal4	4.8	8.4
Petal5	11.5	10

$\sqrt{(\text{Petal1L}-\text{Petal2L})^2+(\text{Petal1W}-\text{Pet}$

Distance Matrix

	Petal1	Petal2	Petal3	Petal4	Petal5
Petal1	0	13.7259	13.3135	1.14018	6.5
Petal2		0	0.5	13.8708	7.4330
Petal3			0	13.4332	7.0710
Petal4	6.140174			0	6.8884
Petal5	6.5	7.4330344			0

Shot on AWESOME A05s



	Petal 1	Petal 2+3	Petal 4	Petal 5
Petal 1	0			
Petal 2+3		0		
Petal 4			0	
Petal 5				0

इसके बाद का table, Linkage explanation के बाद मिलेगा

Different Linkage Methods.

→ Simple Linkage
→ Complete linkage
→ Average linkage
→ Centroid linkage
→ Wards linkage

SINGLE LINKAGE.

This defined as the minimum distance between the points two clusters

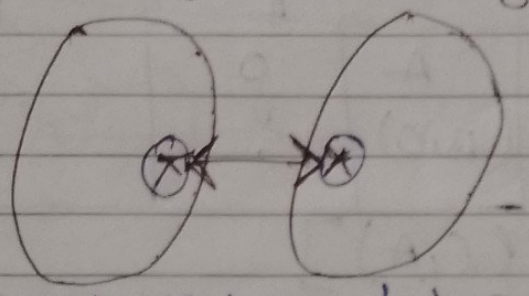
$$d(X,Y) = \min\{d(x,y) \mid x \in X, y \in Y\}$$

Shot on AWESOME A05s

→ The method can create the non elliptical clusters

→ It can produce undesirable results in the presence of outliers

→ It causes a (chaining effect) where clusters have merged since at least one point in a cluster is closest to a point in another cluster. This forms a long and elongated cluster



(Ex) the single linkage between the cluster (B, C) and A is

Min [dist (B, A), dist (C, A)]

Min [0.17, 0.28]

0.17

	A	B, C
A	0	?
B, C	?	0

	A	B, C
A	0	0.17
(B, C)	0.17	0

Complete Linkage

→ It is defined as the maximum distance between the points of the two different clusters

$$d(X, Y) = \max\{d(x, y) \mid x \in X, y \in Y\}$$

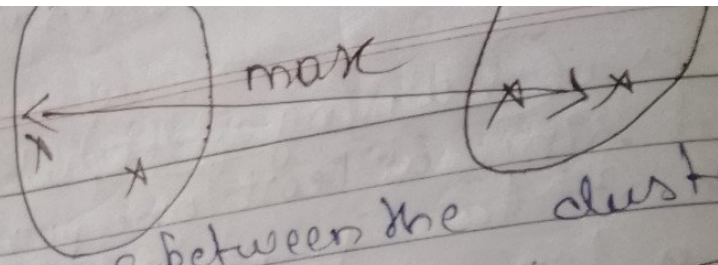
→ The method returns more stable clusters, with

→ nearly equal diameter

→ It avoids the chaining effect

→ It is sensitive to outliers

Shot on AWESOME A05s



Ex:-

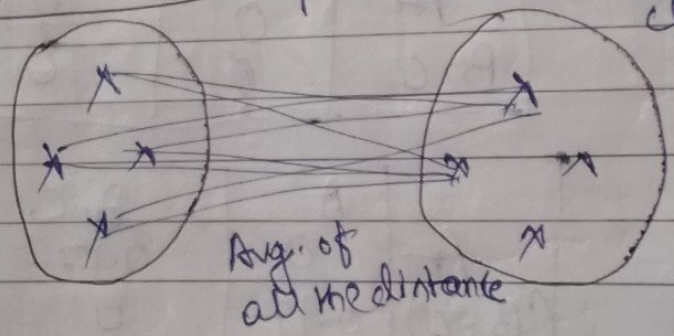
The complete linkage between the clusters (B, C) and A is

	A	B, C
A	0	?
(B, C)	?	0

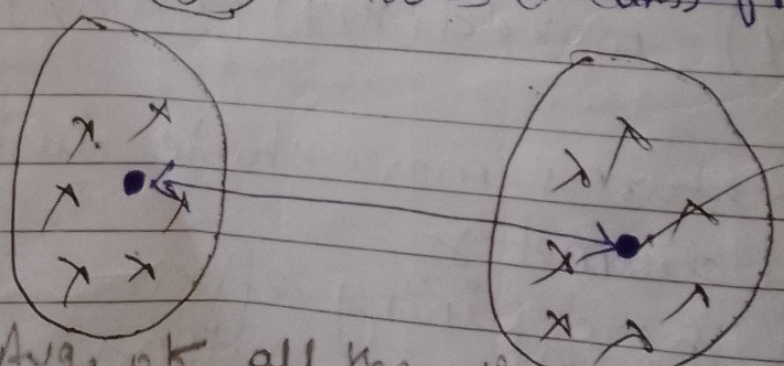
Max [dist (B, A), dist (C, A)]

	A	B, C
A		
(B, C)		

Average Linkage It is defined as the average of all the pairwise distances between the two clusters



Centroid Linkage It is defined as the distance between the centroids (means) of the two clusters



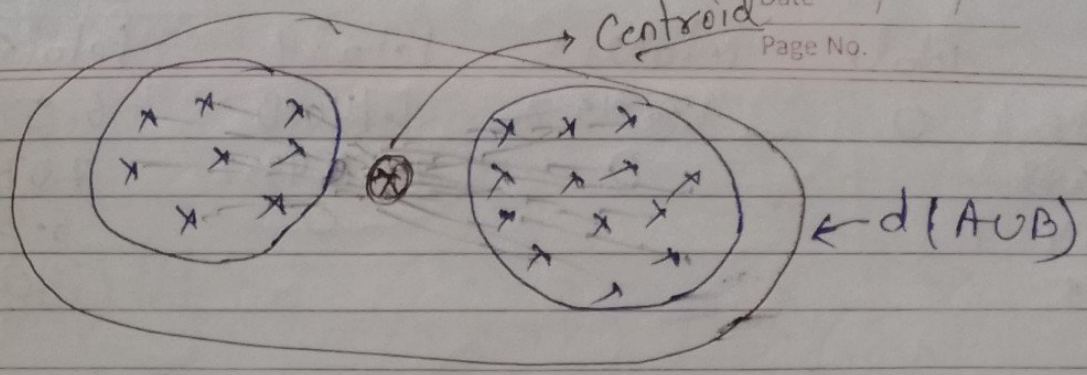
Centroid = Avg. of all the distances in a cluster

Shot on AWESOME A05s

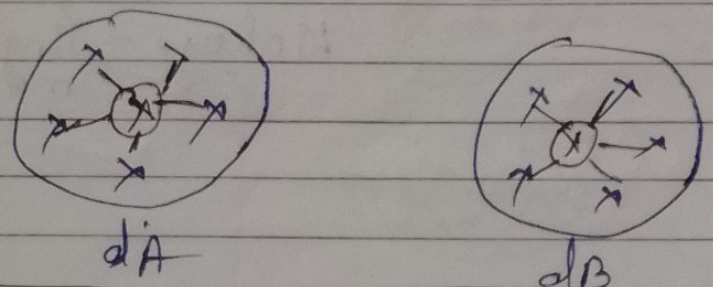
Wards Linkage

Date / /
Page No.

step ①



step ②



$$\boxed{\text{wards} = d_{A\cup B} - d_A - d_B}$$

$$\Delta(A,B) = \sum_{i\in A\cup B} |x_i - m_{A\cup B}|^2 - \sum_{i\in A} |x_i - m_A|^2 - \sum_{i\in B} |x_i - m_B|^2$$

By default, the scikit learn library of python considers the 'ward' linkage

The clusters are merged, if the new cluster minimizes the variance

It is a computationally intensive method.

It is given by the formula

$$d(X,Y) = \sqrt{\frac{2n_X \cdot n_Y}{n_X + n_Y}} \cdot d(X_c, Y_c)$$

n_X = No. of elements in the cluster X

n_Y = No. of elements in the cluster Y

X_c = Centroid of the cluster X

Y_c = Centroid of the cluster Y

Shot on AWESOME A05s

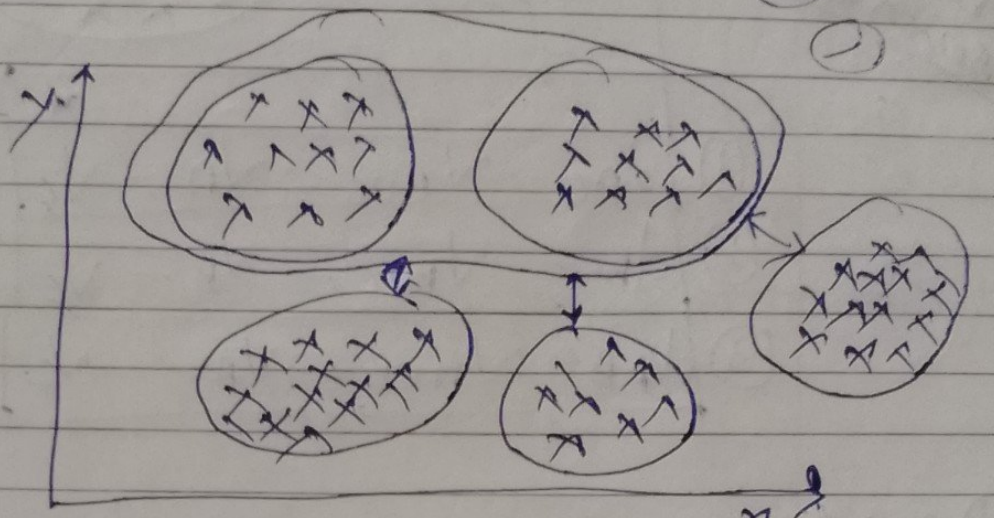
Petal 1+4+5

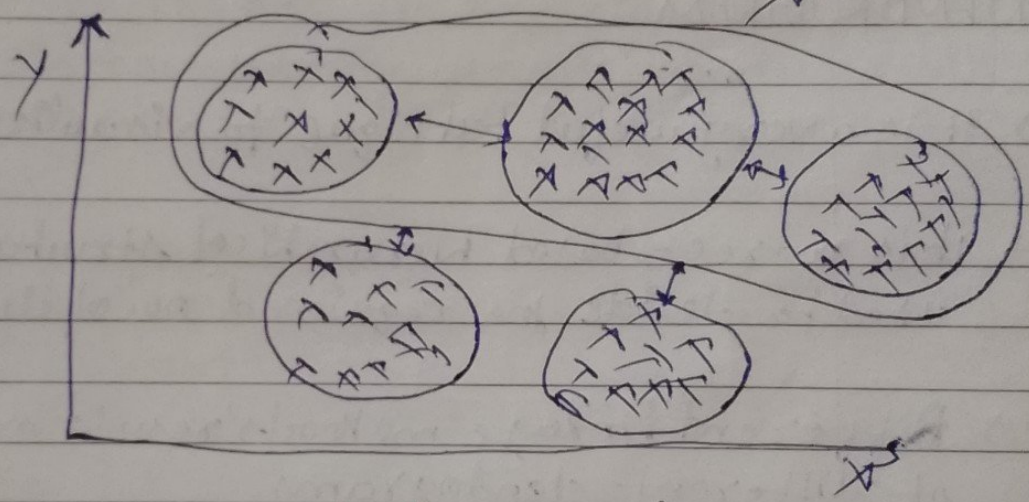
Petal 2+3

Date / /

Page No.

Day 3 Session 1





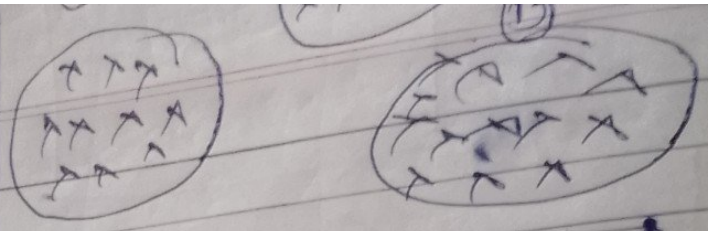
Single Linkage	Complete Linkage	Average Linkage
→ Can create non elliptical clusters	→ Creates more compact clusters	→ Balances b/w single and complete linkage
→ Sensitive to outliers	→ Biased towards globular clusters	→ Robust to outliers
→ Prone to chaining effect	→ Breaks large clusters	→ Biased towards globular clusters

Centroid Linkage	Ward Linkage
→ Cluster formation is similar to average linkage	→ Most effective in presence outliers
→ Can cause inversion	→ Biased towards globular clusters.

Shot on AWESOME A05s

Ward Linkage

Day (3) Session (2)



① $d_{AC} = V_{A \cup C} - V_A - V_C$

② $d_{BC} = V_{B \cup C} - V_B - V_C$

③ $d_{AB} = V_{A \cup B} - V_A - V_C$

DENDROGRAM

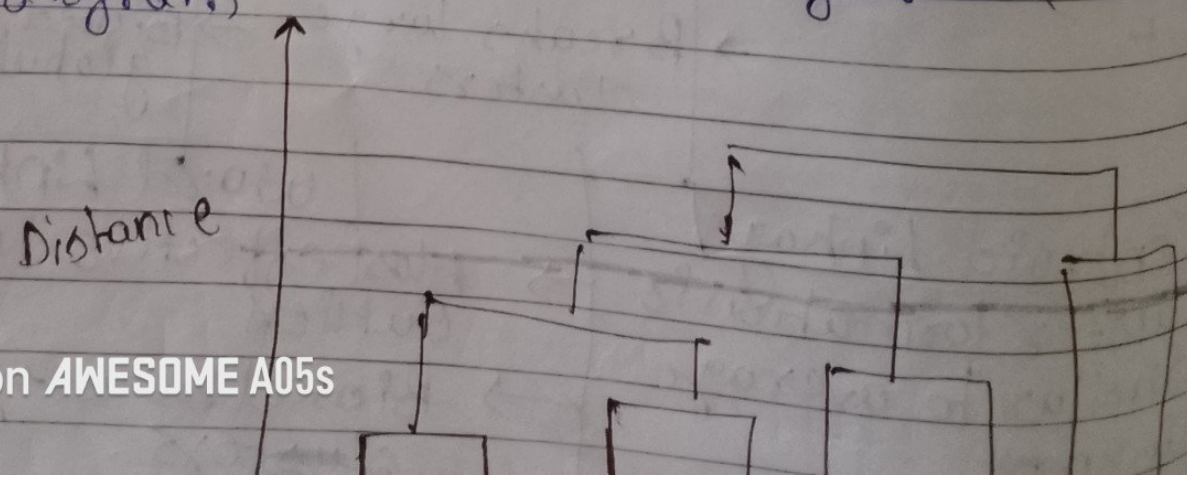
→ It is a very useful technique to visualize the clusters

→ It is a tree-based hierarchical structure that can be used to decide the required no. of clusters

→ Different linkage methods result in the formation of different dendograms

→ Observations linked at a low height represents more similar observations

→ Dissimilar observations fuse at a higher level in the dendrogram



Shot on AWESOME A05s

point, each considered as a single clusters and the distance is given on the Y-axis.

Date / /
Page No.

→ Each single cluster is known as 'leaf'
→ The horizontal line is known as 'clade' which represents the merging of clusters
→ The black line shows the height at which the dendogram is being cut

→ The line intersects the dendogram at 4 distinct points, which gives 4 clusters namely (P_1, P_2, P_3, P_4), (P_5, P_6), P_7 and P_8

→ The clusters P_7 and P_8 are clustered at higher distance than the remaining observations, which suggests more dissimilarity between these points

COPHENETIC CORRELATION COEFFICIENT

big dedogram best hai represent ~ hai data ko

→ Quantifies how the dendrogram has represented the dissimilarities between the observations.

→ It is defined as the correlation coefficient between cophenetic distances and the actual distance b/w the observations

→ The cophenetic distance between the points P_i and P_j is the height represented on the Y-axis of dendrogram at which P_i and P_j are first linked together

→ The actual distance is the pairwise distance b/w the observations represented in the distance matrix.

Shot on AWESOME A05s

then it shows dendogram is good

→ The value close to 1 represents the best linkage quality

→ It is mostly used in biostatistics to evaluate the cluster models

In phythong the linkage matrix provides the cophenetic distance

```
#import the function
from scipy.cluster.hierarchy import cophenet
#pass the linkage matrix and actual distance
#1st output of the cophenet() is the correlation
    coefficient
coeff, cophnet_dist=cophenet(linkage_matrix, actual_dist)
#print the cophnetic correlation coefficient
print(coeff)
```

Dendograms

Merits

→ Does not require a pre-specified number of clusters

→ Hierarchical relation between the clusters can be identified

Dendrogram provides a clear representation of clusters

Demerits

→ Different dendograms are produced for different linkage methods

→ Selecting an optimal number of clusters using dendogram is sometimes difficult

→ Complexity is high

Shot on AWESOME A05s

Parameters in DBSCAN

Date / /
Page No.

→ There are two parameters in the DBSCAN algorithm: epsilon (ε) and the minimum number of samples (min_samples or min_pts)

→ epsilon (ε) is the radius of the neighbourhood for a datapoint.

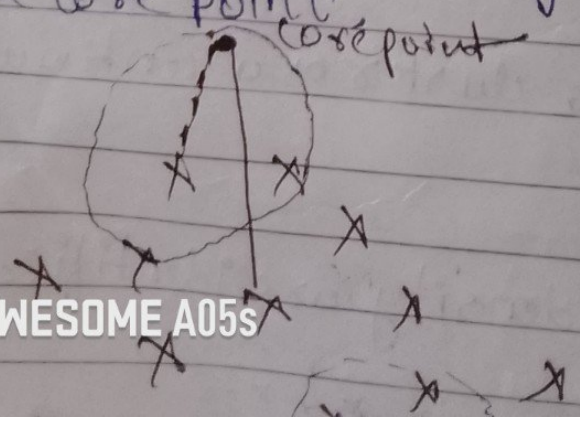
→ Minimum number of samples is the lower bound for the count of data points in the neighbourhood of a core point.

Terminologies

→ A core point is a data point which has at least a minimum number of samples in its ε-neighbourhood (including itself) or otherwise; it is a non core point

→ A data point is 'directly density reachable' from a core point if it is in the ε-neighbourhood of a core point

→ A border point is a point which is not a core point but, it is directly density reachable from a core point



$\varepsilon = 1.5$

min_sample = 5

Shot on AWESOME A05s

→ A point s is 'density reachable' from a core point(c) if there is a path $p_1 = c, p_2, \ldots, p_n = s$, such that p_{i+1} is directly density reachable from p_i

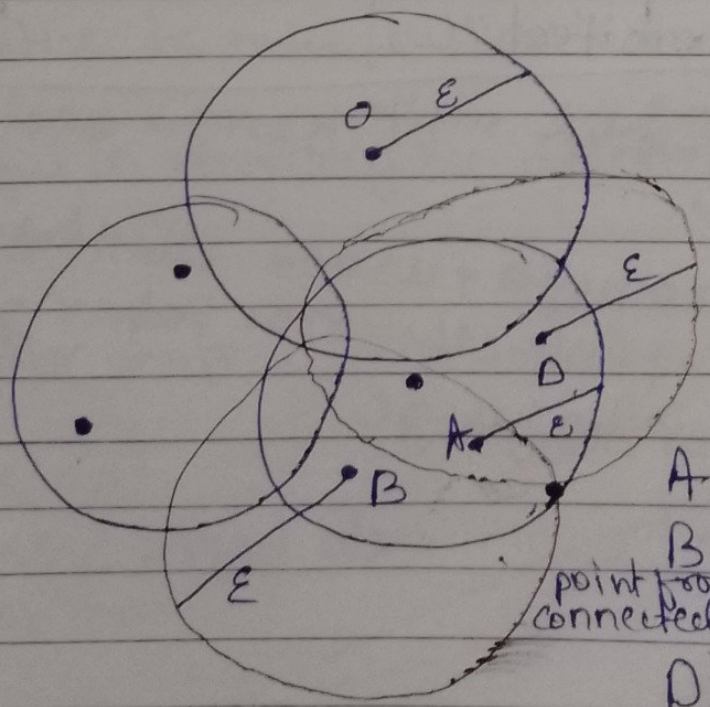
→ The definition of density reachable indicates that the points on the path must be the core points (the last point can be an exception!)

→ Two points are 'density connected' if there is a third point from which both the points are density reachable

→ A point which is not density reachable from any point is known as 'noise' point or outlier

Please note

All the non core points are directly reachable from core points and not vice-versa



A : Core point

B : Directly density reachable point from A, A and C are density connected and density reachable

D : Border point

O : Noise point

Shot on AWESOME A05s

DBSCAN PROCEDURE

Date / /
Page No.

Decide the parameters ε and min_samples

↓

choose a starting point (P) randomly and find its ε-neighbourhood

↓

If P is a core point, find all density-reachable points from P and form a cluster else mark P as noise point.

↓

Find next unvisited point and follow the same step as P

↓

Repeat this procedure till all the points are marked visited

Ek points ~~ko~~ jo ~~k~~ visit karega and circle bana dega.

Circle bana rahega, usose clusters ban jaaye, jo points clusters se bahar rahegaa

No outlier ho jaaye

Shot on AWESOME A05s

Clustering Techniques

Kmeans

→ Partition Based most widely used algorithm

→ Need to specify the number of clusters (k)

→ sensitive to noise / outliers in the data

→ creates more circular clusters

→ Cluster formation depends on the initial centroid assignment.

Hierarchical

→ Hierarchy-based algorithm

→ No need to specify the number of clusters

→ Sensitive to noise / outliers in the data

→ Useful in recovering the underlying Hierarchical st. in the data

→ Cluster shape depends on the distance metric

DBSCAN

→ Density based algorithm. 2nd most widely used algorithm

→ No need to specify the number clusters.

→ Need to specify epsilon (ε) and the minimum no. of samples

Day (4)
Session (8)

Dimension Reduction Techniques

→ The real world dataset may contain a large number of features under study

→ A dataset with a large no. of features needs more time for model training

→ As the no. of variables increase, the data becomes more sparse. Overfitting can occur when a model is built on such data

→ To avoid the sparsity of the data, we require more observations (i.e rows) which may not be easily available in most cases

→ The dist. b/w the diff. points starts converging to a single value with a increase in dimensions. This is known as 'Distance Concentration'

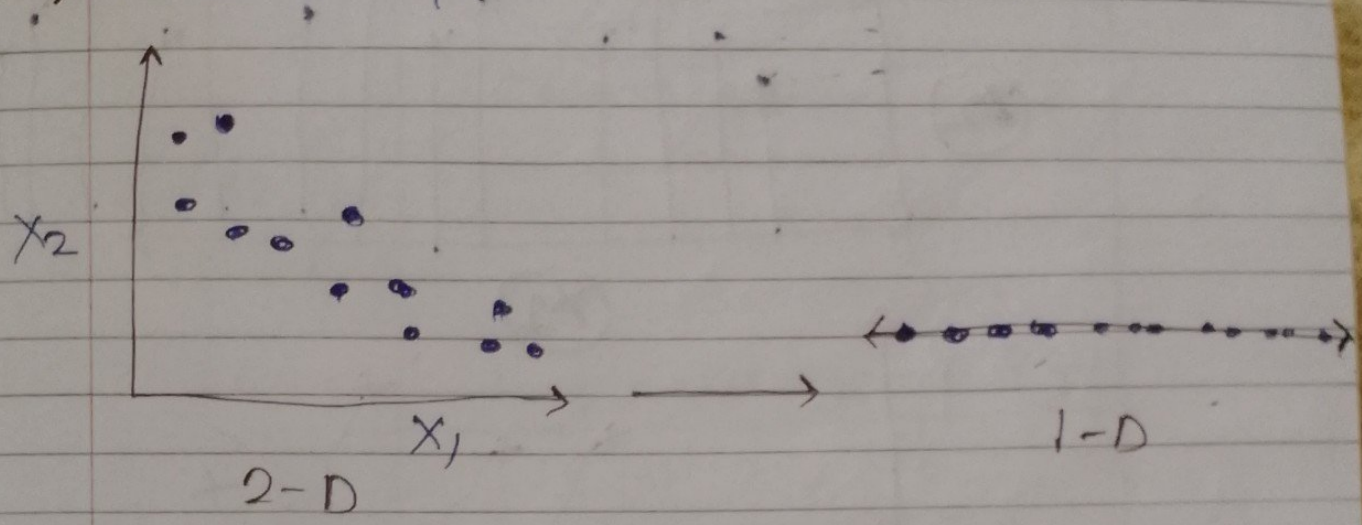
→ To avoid such issues, one can reduce the dimensions of the dataset

→ The dimension reduction techniques remove the redundant variables / noise in the original data, which reduces the training time

→ Reducing the dataset to 2 to 3 dimensions helps in visualization of the data

Shot on AWESOME A05s

→ Various dimension reduction techniques
→ PCA - Principal Component Analysis
→ LDA - Linear Discriminant Analysis
→ Factor Analysis



Approaches for dimension reduction

Two different approaches can be used for dimension reduction: Projection, Manifold learning

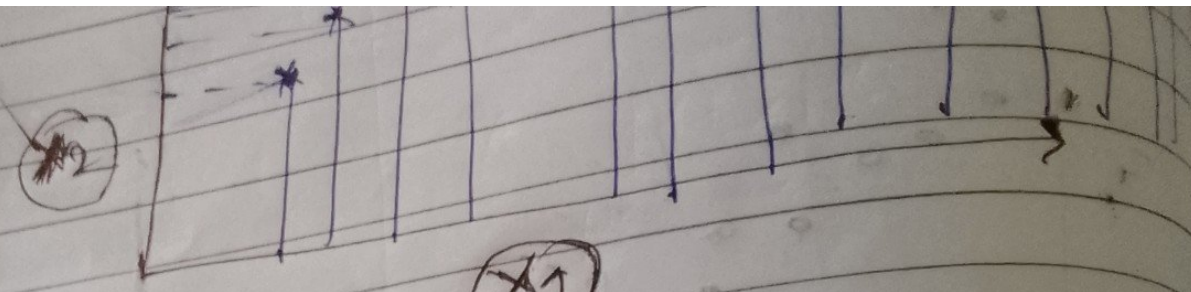
In the projection approach, the original dataset is projected onto the lower-dimensional plane

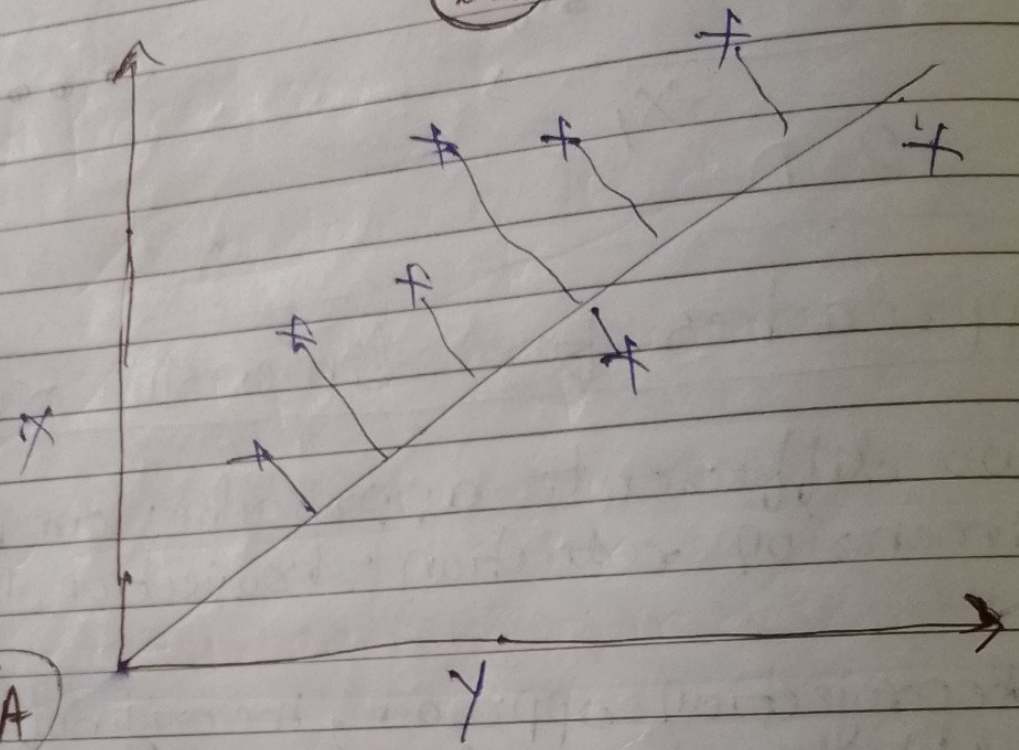
PCA uses the projection approach for dimension reduction

This method is not effective if the dataset has different layers in the higher dimensions

In manifold learning, a manifold is created on which the dataset lies

Shot on AWESOME A05s





PCA

→ It is one of the dimensionality reduction techniques that is used to reduce the dimensions of the large datasets

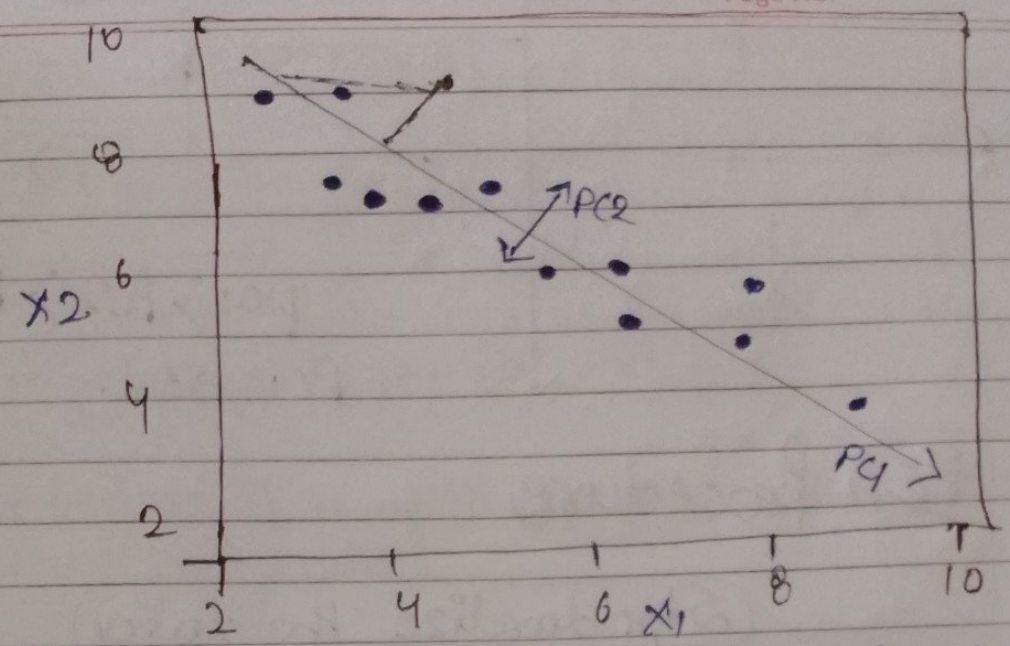
→ It transforms the large set of features into small set such that it will contain the maximum information in the original data.

→ The no. of components is less than or equal the no. of independent variables

→ PCA projects the original dataset on lower dimensional plane

Shot on AWESOME A05s

uncorrelated points

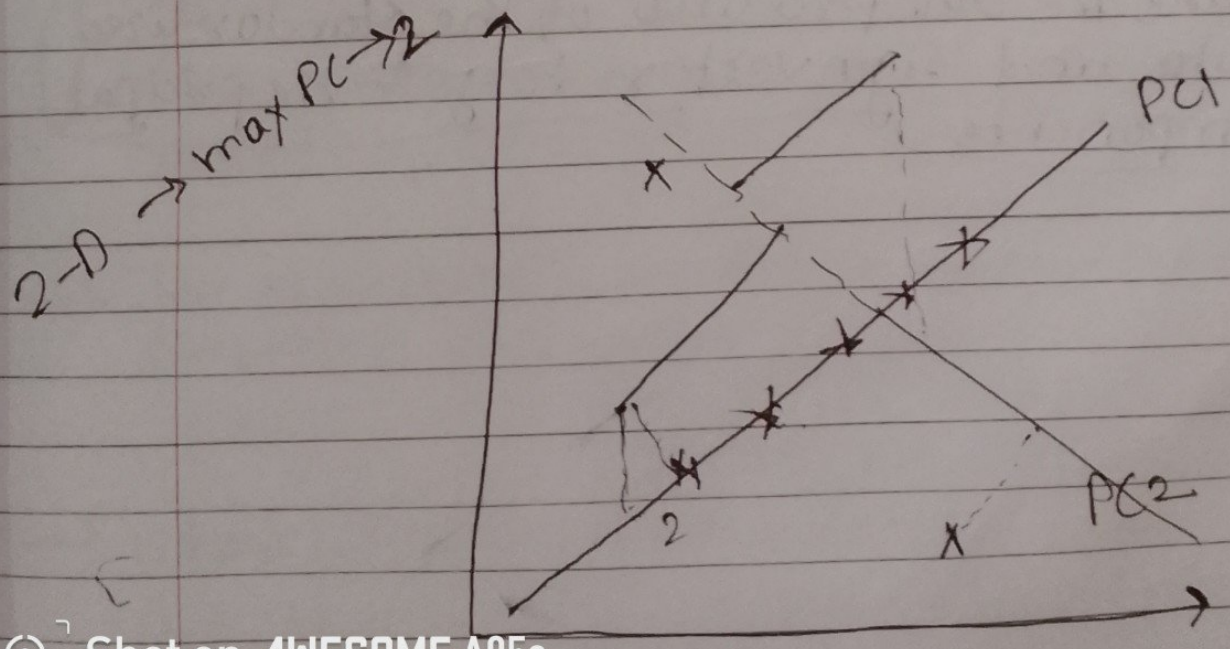


→ The first Principal component (PC1) exhibits the direction of maximum variance in the data.

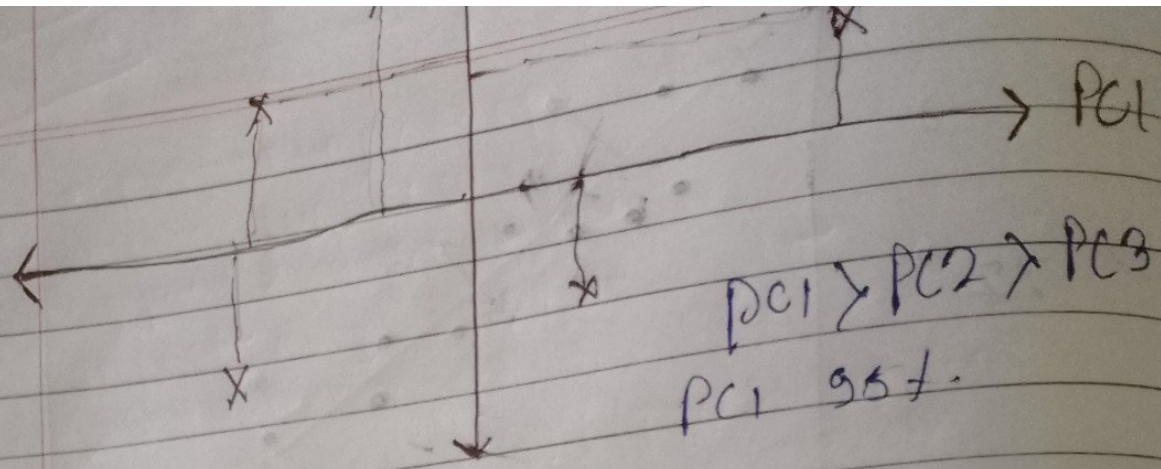
→ It is used to remove the redundancy in data

→ PCA reduces the multicollinearity (if present) in the original data.

→ Principal components are always orthogonal to each other.



maximum data PC1 में है



PC1 > PC2 > PC3 ...

PC1 95%

PCA Procedure

- Standardize the data
- ↓
- Compute the covariance matrix
- ↓
- Calculate the eigenvalues and eigenvectors
- ↓
- Sort the eigenvalues in descending order
- ↓
- Select the eigenvectors that explains the maximum variance in the data
- ↓
- Take the dot product of the standardized data and eigenvectors to get the principal components

Shot on AWESOME A05s

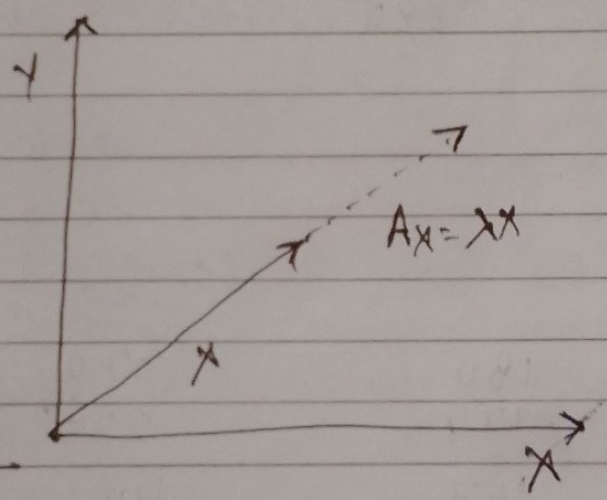
Date / /
Page No.

→ For each eigenvalue λ of a matrix A, there exist a non-zero vector X, which satisfy the equation:
$(A-\lambda I)X=0$ i.e. $AX=\lambda X$

→ The vector X is known as the eigen vector corresponding to the eigenvalue λ

→ Eigenvectors of the distinct eigen values are always linearly independent

→ The eigen vector is a vector that does not changes its direction, after transformation by matrix A



Eigen values

→ In PCA, the eigen value represents the total variance explained by the principal component

→ The percentage of variation explained by the ith component is given as

$$\left(\frac{\lambda_i}{\sum_{i=1}^{n}\lambda_i}\right)*100$$

→ where λ_i is the ith eigen value

→ The eigenvalues are considered in selecting the number of principal components

Shot on AWESOME A05s

Day (5) Session (2)

Selecting principal Components

Date / /
Page No.

→ The principal components are obtained by taking the dot product of the scaled data and the eigen vectors

→ We use the eigenvalues of the covariance matrix to select the optimal number of principal components.

→ In order to reduce the data dimension, we consider first few principal components that explains most of the variance in the data.

→ Different Criteria to select the principal components

- Kaiser Criterion ✓
- Scree Plot ✓

Kaiser Criterion

→ It is an easy criterion to choose the principal components.

→ It selects the number of principal components for which the eigenvalue is greater than (1)

→ Higher eigenvalues explain most of the variance in the data

Shot on AWESOME A05s

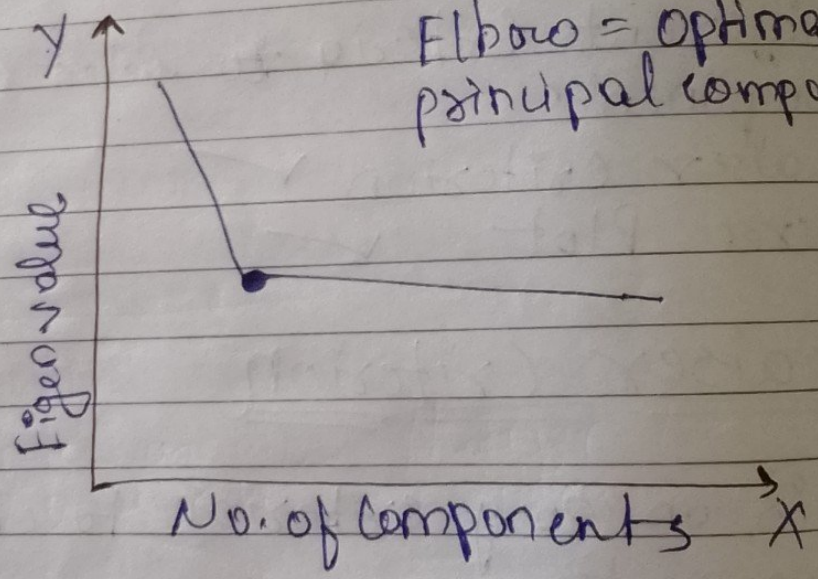
Scree Plot

→ This method plots the eigen values (Y-axis) aganist the number of principal components (X-axis)

→ The elbow point in the scree plot corresponds the optimal number of components

→ After the elbow point, the components do not contribute much to the variance in the data

→ This method fails if there is no explicit elbow point in the Scree plot



Elbow = optimal no. of principal components

Percentage Of Total Variance

One can decide the number of principal components based on the percentage of variance explained by the variables

In most of the cases, the components explaining 80% of variance can be considered as the principal components

Shot on AWESOME A05s

Summary

→ PCA is an unsupervised dimension reduction technique that uses the projection method to reduce the data dimension

It finds the principal components that best represents the data in much lower dimensions

The first principal component explains most of the variation in the data, the second principal component explains 2nd most variation in the data and so on.

The principal components are always orthogonal to one another

Reducing the dimension of the data to 2 to 3 dimensions aids in data visualization.

PCA through Sklearn

```
from sklearn.decomposition import PCA

df_s
```

	AccelSec	TopSpeed_kmH	Range_km
0	-0.998198	1.540888	0.954
1	1.408172	-0.690954	-0.63
2	-0.953636	0.837705	0.5141
3	-0.017825	-0.079490	0.161590
4	1.185360	-1.149551	-1.512807

Shot on AWESOME A05s

LDA

→ LDA was formulated by Ronald A Fisher in 1936

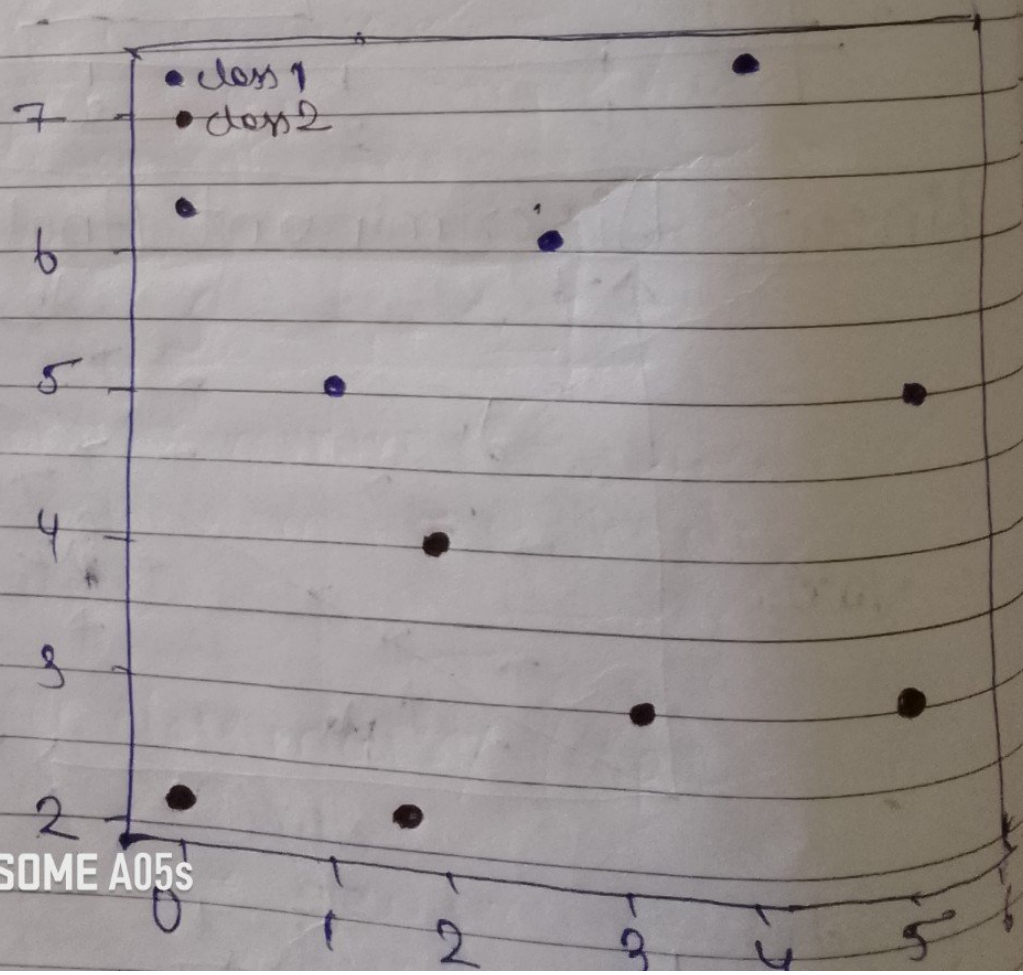
→ It is linear transformation technique and is mostly widely used for dimensionality reduction

→ It is used as a pre-processing stage for pattern classification

→ The purpose of LDA is to lower the dimension space with a good separability between the classes

→ It is known to have a practical use as a binary as well as multiclass classifier

LDA



Shot on AWESOME A05s

known classes/categories with 'n' independent features

- In LDA, we project the original data on the k (k < C) vectors that maximizes the separation between the classes

- For the given figure, we project the data on the 1-D line that maximizes the separational between class 1 and class 2.

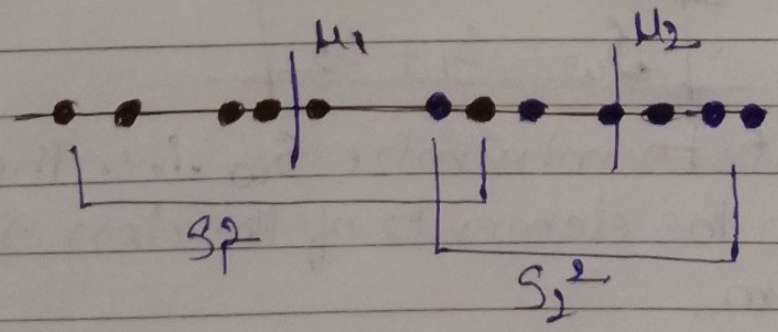
LDA

Goal of LDA is to:

Maximize the distance between the means (i.e. between μ_1 and μ_2) of classes

Minimize the variance within each class.

In the given figure μ_1 and μ_2 are the means and S_1^2 and S_2^2 are the variances of the two classes.



Shot on AWESOME A05s

Standardize the data
↓
Compute the within class matrix
↓
Compute the b/w class matrix
↓
Find the projection vectors and transform the data

Within class matrix

→ It captures how precisely the data is scattered within the class

→ Consider the data divided into two classes C_1 and C_2, the within class matrix is given by the summation of the covariance matrix (S_1) of the class C_1 and the covariance matrix (S_2) of the class C_2

$$S_w = S_1 + S_2$$

→ we want to minimize S_w i.e. the distance between the elements of the class should be minimum

Between class matrix (S_B)

→ It represents how precisely the data is scattered across the classes

→ Suppose the means of classes C_1 and C_2 are

Shot on AWESOME A05s

→ μ_1 and μ_2 respectively. The formula to find S_B is given as

$$\boxed{S_B = (\mu_1 - \mu_2) \cdot (\mu_1 - \mu_2)^T}$$

→ We want to maximize S_B. i.e. the distance between the two classes should be higher.

Projection Vector

→ The data samples are to be projected on the projection vectors with lower dimension

→ These vectors are the eigen vectors of the matrix

$$\boxed{S_W^{-1} \cdot S_B}$$

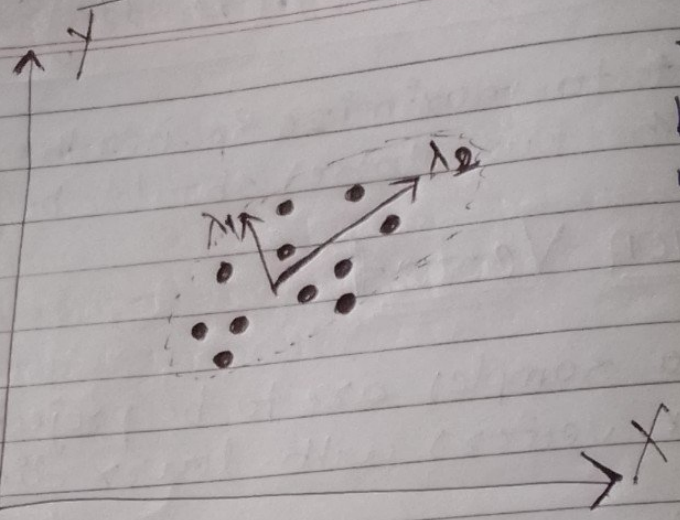
→ The eigenvector associated with highest eigenvalue represents the direction of highest separability between the classes

→ The projection vectors are the coefficients of the linear discriminants that exhibit the direction of maximum separability of classes

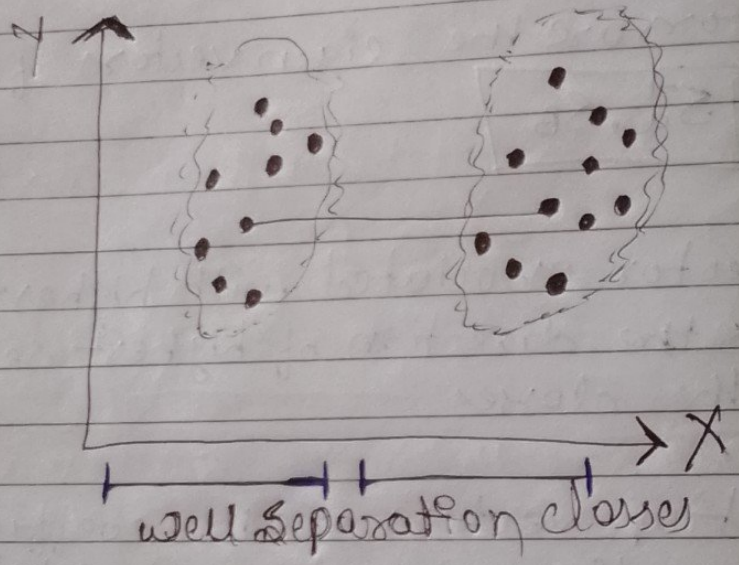
→ Like PCA, transform the original data by taking a dot product of original data with projection vectors

Shot on AWESOME A05s

PCA and LDA



In PCA the components maximize the variance.



In LDA the component axes maximizes the classes.

Imp

PCA	LDA
① It is a unsupervised machine learning technique	It is a supervised machine learning technique
Finds the components that maximize the variance in the data.	Calculates the linear discri to obtain the maximum Separation between the clas of the dependent variable
It obtains the components that best represents the dataset into lower dimensions	It obtains the discriminant that best discriminants b/w the classes

Shot on AWESOME A05s