

## HIERARCHICAL CLUSTERING:

Also called as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters.

→ The endpoint is a set of clusters, where each cluster is distinct from each other cluster.

→ The objects within each cluster are broadly similar to each other.

## TYPES OF HIERARCHICAL CLUSTERING:

### 1. AGGLOMERATIVE CLUSTERING:

- Also known as AGNES (Agglomerative Nesting).

- It works in a bottom-up manner.

(19)

## 2. DIVISIVE HIERARCHICAL CLUSTERING:

- Also known as DIANA (Divise Analysis)

- It works in a top-down manner.

### NOTE:

- In hierarchical clustering, one can stop at any number of clusters.
- One can find appropriate by interpreting the dendrogram.
- One can use median or mean as a cluster centre to represent each cluster.

$\therefore$  Hierarchical clusters (or) the hierarchy

of clusters is represented as a dendrogram or tree structure.

\* On top of the output obtained, we try

to implement <sup>the</sup> K-Means - <sup>IN</sup> INDUSTRY.

(20)

This algorithm works by grouping the data one by one on the basis of the nearest distance measured of all the pairwise distance between the data point

→ Again the distance between the data point is recalculated but which distance to consider when the groups has been performed.

To perform this, there are many methods available. They are

1. SINGLE - NEAREST DISTANCE (OR) SINGLE LINKAGE:

It is the shortest distance between the closest points of the clusters.

2. COMPLETE - FARTHEST DISTANCE (OR) COMPLETE LINKAGE:

It is the farthest distance between the two points of the different clusters.

\* Complete linkage is one of the popular linkage methods as it forms tighter clusters than single-linkage.

### 3. AVERAGE - AVERAGE DISTANCE (OR) AVERAGE LINKAGE:

It is the distance between each pair of observations in each cluster are added up & divided by the number of pairs to get an average inter-cluster distance.

\* Average <sup>linkage</sup> & complete linkage are the two most popular distance metrics in hierarchical clustering.

\* Average linkage returns the value of the Arithmetic mean.

(22)

#### 4. CENTROID LINKAGE:

It is the distance between the centroids of two clusters.

\* As the centroids move with new observation it is possible that the smaller clusters are more similar to the new larger cluster than to their individual clusters causing an inversion in the dendrogram.

#### 5. WARD'S LINKAGE METHOD:

It is to specify the distance between two clusters computed as the increase in the

"Error sum of squares (ESS)" after fusing two clusters into a single cluster.

The purpose of them<sup>(linkage)</sup> is to find the distance.

→ So, these are called as Inter-cluster distance computation.

→ This is the way we go on grouping the data until one cluster is formed.

\* Now on the basis of dendrogram graph we can calculate how many number of clusters should be actually present.

#### STEPS TO BE FOLLOWED TO PERFORM

#### HIERARCHICAL CLUSTERING:

• Let  $X = \{x_1, x_2, \dots, x_n\}$  be the set of data points.

1. Begin with the disjoint clustering

having level  $\lambda(0) = 0$  & sequence number

$$m = 0.$$

2. Find the least distance pair of clusters

(24)

In the current clustering, say pair(r), (s) according to the  $d[(r), (s)] = \min d[(i), (j)]$  where the minimum is over all pairs of clusters in current -ng

3. Increment the sequence number i.e.,

$$m = m + 1.$$

Merge clusters (r) & (s) into a single cluster to form the next clustering m. Set the level of this clustering to  $L(m) = d[(r), (s)]$ .

4. Update the distance matrix D, by deleting the rows and columns corresponding to the clusters (r) & (s) & adding a row & column corresponding to the newly formed cluster.

Then, the distance between the new cluster denoted (r,s) and old cluster (k) is defined

(25)

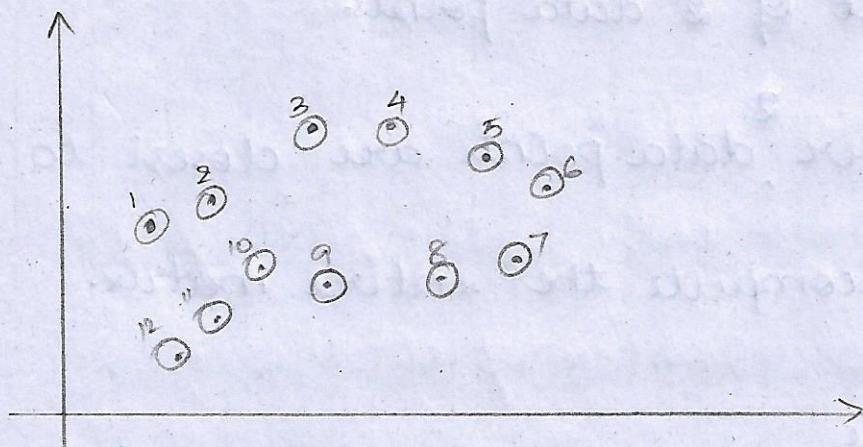
as  $d[(k), (r, s)] = \min(d[(k), r], d[(k), s])$ .

5. If all the data points are in one cluster then stop else repeat from step 2.

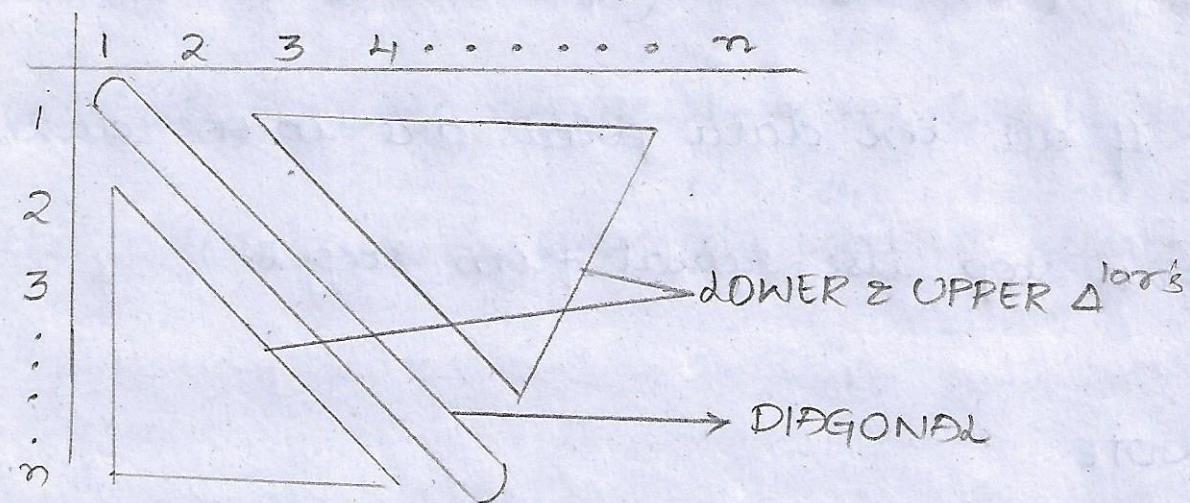
NOTE:

The divisive Hierarchical clustering is just the reverse of Agglomerative Hierarchical clustering.

TASK : TO cluster the data,



→ Now try to create a distance matrix and compute the distance matrix.



NOTE:

We will consider or take values from upper or lower triangular matrices

→ Always try to pick the lower value from the closest of 2 data points.

→ Which ever <sup>2</sup> data points are closest to the cluster, recompute the entire matrix.

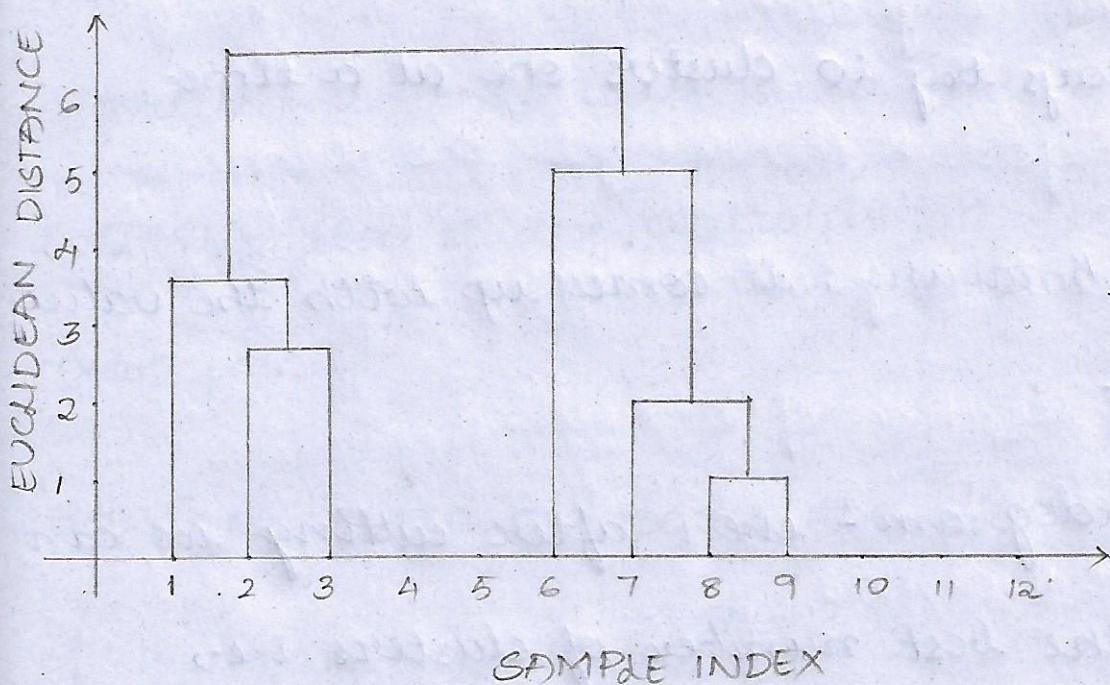
so, over here

- ↳ Computing distance matrix is very complex.
- ↳ We should be aware of the linkage.
- ↳ Dendrogram is also considered.

## DENDOGRAM :

A dendrogram is a type of tree diagram showing the hierarchical relationships between different sets of data.

→ So, just by looking at the Dendrogram we can tell how the clusters are formed.



### NOTE:

→ Distance between the data points represents the dissimilarities.

→ Height of the blocks represents the distance between clusters.

- ↳ We can cut the dendrogram tree with a horizontal line at a height where the line can ~~traverse~~<sup>traverse</sup> the maximum distance up & down without intersecting the merging point.
- Always try to cluster one at a time.

\* In K-means : it comes up with the value of K.

\* Dendrogram - even after cutting we can see the best number of clusters i.e., allocating the objects to the clusters.

## K-MEANS:

The main objective of k-means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

## DRAWBACKS:

The only one disadvantage of k-means algorithm is that it is sensitive to the initialization of the centroids (or) the mean points.

So, in order to overcome the draw back of k-means we use  $k_{\text{means}}^{++}$ .

## K-MEANS++:

This algorithm ensures a smarter initialization of the centroids and improves the quality of the clustering.

\*→ Apart from initialization, the rest of the algorithm is the same as the standard k-means algorithm.

→ Therefore, K-means++ is the standard k-means algorithm coupled with a smarter initialization of centroids.

## STEPS INVOLVED:

1. Randomly select the first centroid from the data points.
2. For each data point compute its distance from the nearest, previously chosen

centroid.

3. select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid.

Ex:-

the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid.

4. Repeat steps 2 and 3 until 'k' centroids have been sampled.

## PRINCIPAL COMPONENT ANALYSIS (PCA):

It is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction of large datasets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set in machine learning.

→ PCA can also be used to filter noisy datasets, such as image compression.

### WHEN IS PCA USED?

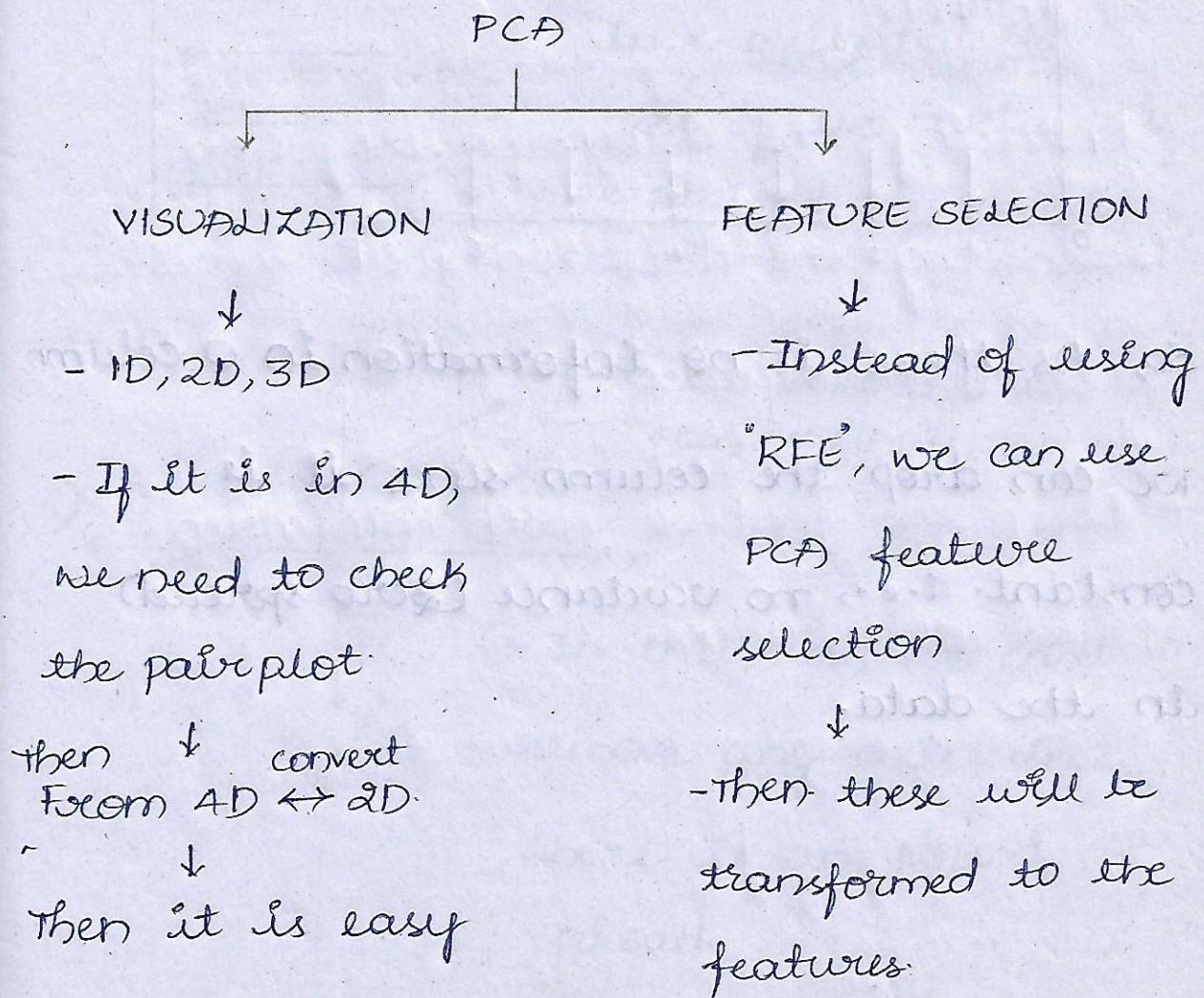
PCA forms the basis of multi-variate data analysis based on projection methods.

→ The most important use of PCA is to

represent a multivariate data table as smaller set of variables (summary indices) in order to observe trends, jumps, clusters and outliers.

→ So, by creating new uncorrelated variables that successively maximize the variance.

### USES OF PCA :



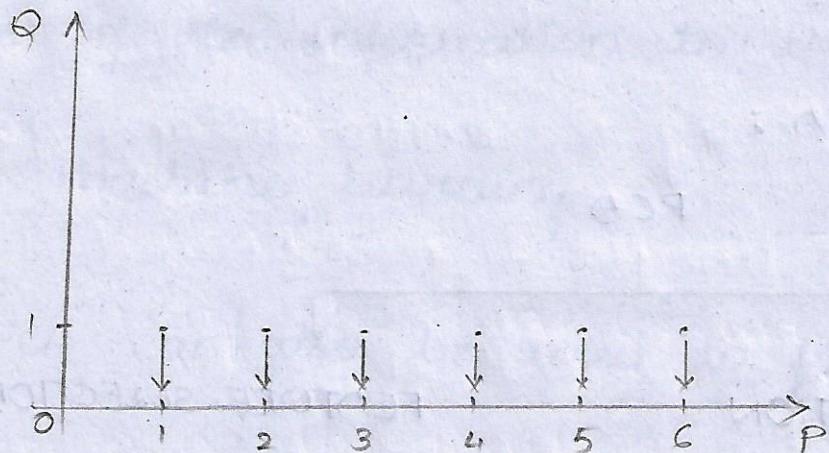
Example:

P	1	2	3	4	5.3	.....	.....
Q	1	1	1	1	1	.....	.....

In the above data,

$P \rightarrow$  CONTINUOUS,  $Q \rightarrow$  DISCRETE

If we try to plot the data, then



So, As there is no information in Q column,  
we can drop the column since it is  
constant i.e., no variance (zero spread)  
in the data.

(35)

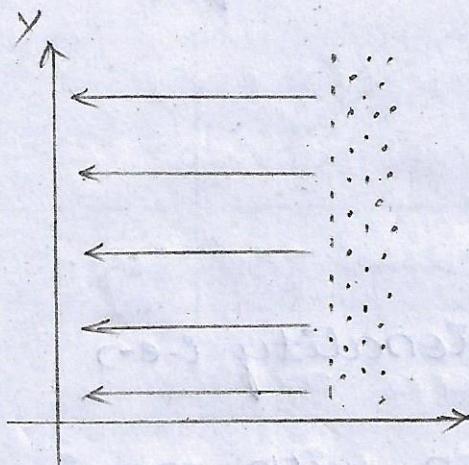
→ As we are dropping down the data i.e.,

9 column, now it is reduced = the

dimension is 1D after reducing.

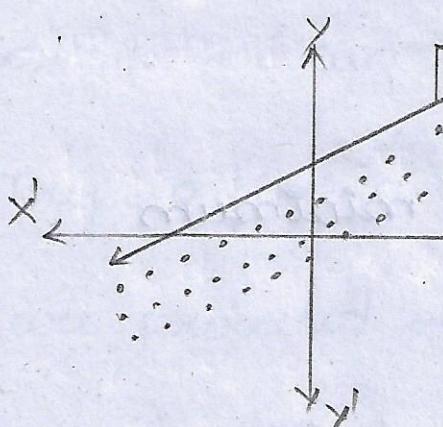
→ So, from the above graph we can say

that the data is projected only towards  
the x-axis.



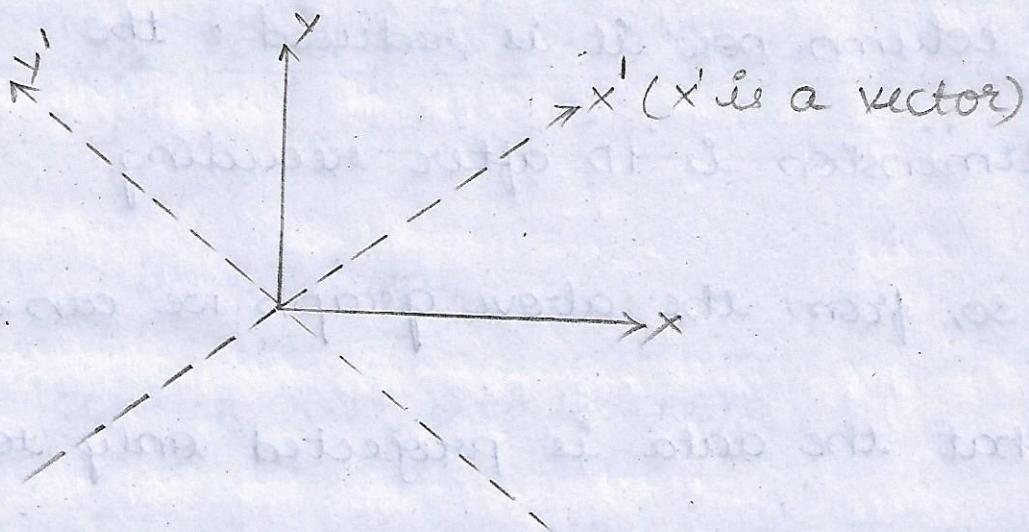
→ In this case,  
we have variability  
in x-axis, but <sup>its</sup> less.

Most of the variance is  
explained by y-axis.



→ In this case, we can't  
remove any axis since  
there is an equal  
spread.

In such cases, try to rotate the axis, then



Now simply drop the y-axis after the transformation.

So, In PCA;

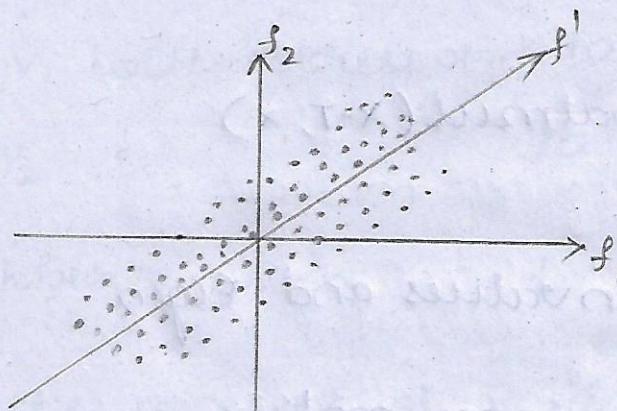
→ Reducing the dimensionality i.e.,

Preserving the direction with maximum spread (or) variance

→ Drop the feature with minimum spread (or) variance.

## xen HOW DO PCA SELECT FEATURES?

The basic idea when using PCA as a tool for feature selection is "to select variables according to the magnitude i.e., & from largest to smallest in absolute values of their coefficients.



→ TASK: Try to find that direction where the variance is maximized and simply drop the least variance.

So, now we can say that,

Try to find a direction  $f'$  such that if  $x_i$ 's are projected on  $f'$ , the variance should be maximized.

⇒  $x'_i = \text{Project } x_i \text{ on } f' \text{ i.e., we get new data points}$

⇒ Try to max. {variance( $\text{proj}_f(x_i)$ )}

STEPS INVOLVED IN PCA:

1. Standardize (or) Normalize the data.
2. Calculate the covariance matrix for the features in the dataset i.e.,

$$X^T \cdot X = S_{d \times d}$$

In Numpy - `np.matmul(X.T, X)`

3. Calculate the Eigenvalues and Eigenvectors for the covariance matrix.

so, if we have  $d \times d$  covariance matrix then,

Eigen vectors -  $v_1, v_2, \dots, v_d$

Eigen values -  $\lambda_1, \lambda_2, \dots, \lambda_d$

4. Sort the eigenvalues and their corresponding eigenvectors.

(39)

$$\Rightarrow \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$$

so, whenever we have a matrix, we get the vectors, i.e.,

it gives all the vectors which are orthogonal (+ve).

+ If we keep eigen values sorted then,  
 $\nu_1$ , will correspond to the max. variance.

$\nu_2$  " " " to the 2<sup>nd</sup> " "

where  $\lambda_1, \lambda_2, \dots, \lambda_d$  are basically give the amount of variance explained by

$$\nu_1, \nu_2, \dots, \nu_d$$

∴ We can say that, In real world application, Eigen values & Eigen vectors tells the maximum variance in the data.

(40)

So, 90% of the variance is to be preserved;

then we try to summate i.e.,

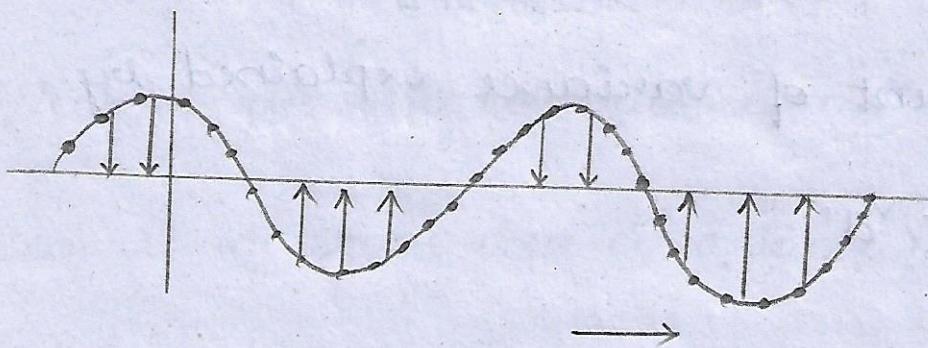
We reduce d to 4d

$$\Rightarrow X_{n \times d} \xrightarrow{\text{PCA}} X'_{n \times d'} \text{ where } d' = d.$$

5. Pick 'k' eigen values and form a matrix

of eigen vectors.

6. Transform the original matrix.



In the above case,

If we try to implement PCA,

\* 1<sup>st</sup> variance is on x-axis & 2<sup>nd</sup> variance  
on y-axis.

(21)

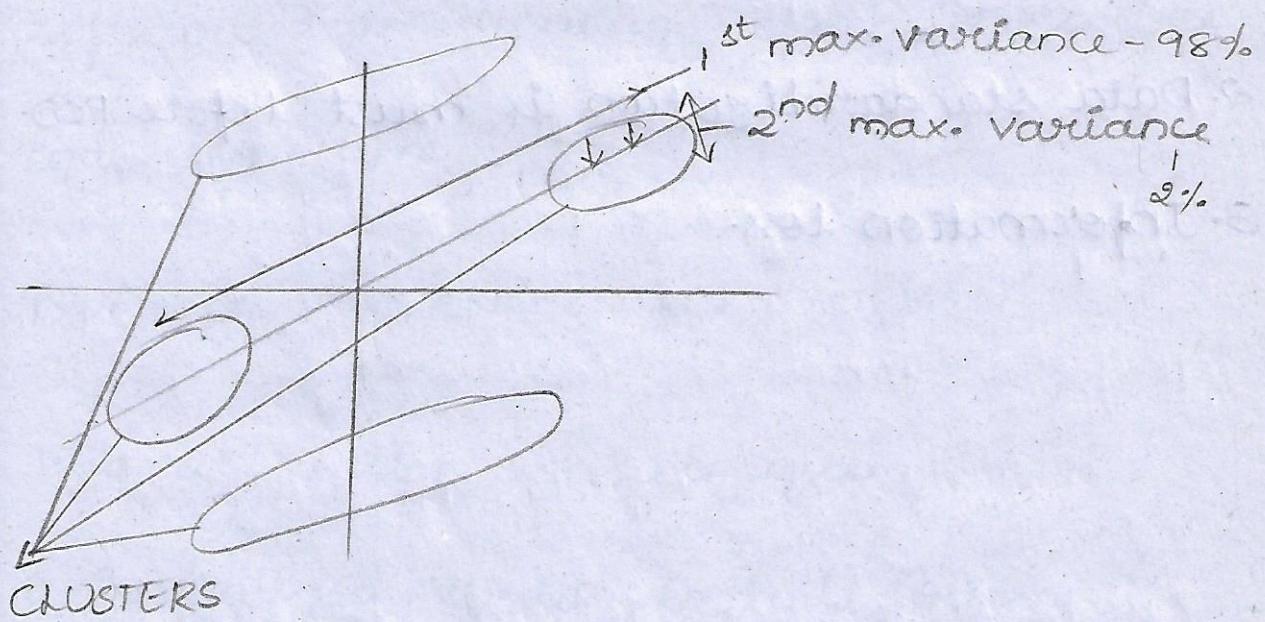
So, we project on X-axis and remove Y-axis.

→ Then the output is



⇒ X-axis has 98% variance and Y-axis has 2% variance.

We lost the pattern by doing PCA; we removed some information.



In the above case, here we mixed 2 clusters i.e., we missed the cluster information if we perform PCA.

## ADVANTAGES OF PCA :

1. Removes correlated features.
2. Improves Algorithm performance.
3. Reduces Overfitting
4. Improves visualization.

## DISADVANTAGES OF PCA :

1. Independent variables become less interpretable.
2. Data standardization is must before PCA.
3. Information loss.