

## ASSIGNMENT II

**Briefly describe the "Clustering of Countries" assignment that you just completed within 200-300 words. Mention the problem statement and the solution methodology that you followed to arrive at the final list of countries. Explain your main choices briefly**

### **Problem Statement -**

HELP International is an international humanitarian NGO that is committed to fighting poverty and providing the people of backward countries with basic amenities and relief during the time of disasters and natural calamities. It runs a lot of operational projects from time to time along with advocacy drives to raise awareness as well as for funding purposes.

After the recent funding programmes, they have been able to raise around \$ 10 million. Now the CEO of the NGO needs to decide how to use this money strategically and effectively. The significant issues that come while making this decision are mostly related to choosing the countries that are in the direst need of aid.

### **Objective-**

Our job is to categorise the countries using some socio-economic and health factors that determine the overall development of the country. Then you need to suggest the countries which the CEO needs to focus on the most.

### **Steps-**

- Calculate the derived metrics for a column if needed.
- Rescale the features and Calculate the PCA components for the dataset. Decide on the variance ratio which has been explained by principal components. From the plots of variance ratio Vs PCA components, we noticed that 80% of variance has been explained by the first two components
- A Scree plot is a diagnostic tool to check whether PCA works well on your data or not. From Scree plot shows how much variation is explained by each cumulative sum of PCA components variance ratio.
- From the Elbow curve method, For each value of  $k$ , we calculate the sum of squared errors. If the line chart looks like an arm, then the 'elbow' on the arm is the value of  $k$  that is best. The idea is that we want a small SSE, but that the SSE tends to decrease toward 0 as we increase  $k$  (the SSE is 0 when  $k$  is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster).
- The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like the number of clusters visually. For each cluster  $k$ , we will calculate the silhouette score, if they are close to 1 indicates that the data point is very similar. If -1 indicates that the data point is not similar.
- We need to check whether data has the clustering tendency or not. These can be decided by using the 'Hopkins' Statistic test. If the value close to 1 tends to indicate the

*data is highly clustered, random data will tend to result in values around 0.5, and uniformly distributed data will tend to result in values close to 0.*

- *Hierarchical clustering - Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom.*

- *Different type of Linkage –*

1. *Single Linkage - In single linkage hierarchical clustering, the distance between two clusters is defined as the shortest distance between two points in each cluster*

2. *Complete Linkage - In complete linkage hierarchical clustering, the distance between two clusters is defined as the longest distance between two points in each cluster.*

- *From the both the linkage steps, we will try to cut the cluster at 3 or 4 and decide on the clusters visually.*

- *Finally, we will decide on the clusters which we need to pay heed too.*

- *Then based on mean income and gdpp, we take all countries that have less than this income and gdpp.*

- *Finally, after filtering this, we obtain final cities we have to concentrate on.*

- *We have used PCA above to reduce the variables involved and then done the clustering of countries based on those Principal components and then later we identified few factors like child mortality, income, etc which plays a vital role in deciding the development status of the country and built clusters of countries based on that. Based on those clusters we have identified the list of countries that are in dire need of aid. The list of countries is subject to change as it is based on the few factors like the number of components chosen, the Number of Clusters chosen, the Clustering method used, etc. which we have used to build the model.*

## **Question 2: Clustering**

a) *Compare and contrast K-means Clustering and Hierarchical Clustering.*

b) *Briefly explain the steps of the K-means clustering algorithm.*

c) *How is the value of 'k' chosen in K-means clustering? Explain both the statistical as well as the business aspect of it.*

d) *Explain the necessity for scaling/standardisation before performing Clustering.*

e) *Explain the different linkages used in Hierarchical Clustering.*

## **Comparison of K-Means Clustering and Hierarchical clustering-**

- *Clustering is a process of keeping similar data into groups. Clustering is an unsupervised learning technique as every other problem of this kind; it deals with finding a structure in a collection of unlabeled data.*

- *The goal of clustering is to provide measures and criteria that are used for determining whether two objects are similar or dissimilar. The aim of clustering is descriptive, and classification is predictive.*

- **K- Means clustering:**

- It is a partition method, a technique which finds mutual exclusive clusters of spherical shape. A specific number of disjoint, flat (non-hierarchical) clusters are generated.
  - The statistical method can be used to cluster to assign rank values to the cluster categorical data. Categorical data have been converted into numeric by assigning rank value. K-Means algorithm organizes objects into  $k$  – partitions
  - Where each partition represents a cluster. Researchers start out with the initial set of means and classify cases based on their distances to their centers.
  - Next, Researchers compute the cluster means again, using the cases that are assigned to the clusters; then, Researchers reclassify all cases based on the new set of means.
  - Finally, Researchers calculate the means of cluster once again and assign the cases to their permanent clusters.
  - **Hierarchical Clustering:**
    - In hierarchical clustering, Researchers assign each item to a cluster such that if Researchers have  $N$  items then Researchers have  $N$  clusters.
    - Find closest pair of clusters and merge them into a single cluster. Compute distance between new cluster and each of old clusters. Researchers have to repeat these steps until all items are clustered into  $K$  no. of clusters.
- Types of clustering –
- Hierarchical Clustering Agglomerative (bottom up) - Starts with all documents belong to the same cluster. Eventually all documents belong to the same cluster
  - Divisive (top down) - Start with each document being a single cluster Eventually each node forms a cluster on its own.

Properties	K-Means	Hierarchical Clustering
<b>Definition</b>	K Means Clustering generates a specific number of disjoint, flat (non-hierarchical) Clusters.	Hierarchical Clustering method construct a hierarchy of Clustering, not just a single partition of objects.
<b>Clustering Criteria</b>	Clustering Criteria It is well suited to generating globular Cluster.	Use a distance matrix as Clustering Criteria. A termination Condition can be used .Example –A number of Clusters.

<b>Performance</b>	<i>The performance of K-mean algorithm is better than Hierarchical Clustering Algorithm.</i>	<i>Hierarchical Clustering Algorithm performance is less as compare to K-mean algorithm.</i>
<b>Category Data</b>	<i>K- Means can be used in categorical data is first converted into numeric by assigning rank.</i>	<i>Hierarchical algorithm was adopted for categorical data, and due to its complexity a new approach for assigning rank value to each categorical attribute.</i>
<b>Sensitive To Noise</b>	<i>K-Means is very sensitive to noise in the dataset.</i>	<i>It is less sensitive to noise in the dataset</i>
<b>Cluster</b>	<i>There are always K.</i>	<i>The number of Clusters k is not required as an input.</i>
<b>Execution Time</b>	<i>K -mean algorithm also increases its time of execution.</i>	<i>Hierarchical algorithm its performance is better.</i>
<b>Quality</b>	<i>K-Means algorithms Shows less quality.</i>	<i>Hierarchical algorithm shows more quality.</i>
<b>Data Set</b>	<i>k -mean algorithm is good for large dataset</i>	<i>Hierarchical is good for small datasets.</i>

### **Steps of the K-means clustering algorithm –**

- **K means** algorithm is an iterative algorithm that tries to partition the dataset into Kpre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**

- It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum.
- The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way k means algorithm works is as follows:

1. Specify number of clusters  $K$ .
2. Initialize centroids by first shuffling the dataset and then randomly selecting  $K$  data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
  - Compute the sum of the squared distance between data points and all centroids.
  - Assign each data point to the closest cluster (centroid).
  - Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach kmeans follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.

Below is a break down of how we can solve it mathematically.

The objective function is:

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|x^i - \mu_k\|^2 \quad (1)$$

where  $w_{ik}=1$  for data point  $x_i$  if it belongs to cluster  $k$ ; otherwise,  $w_{ik}=0$ . Also,  $\mu_k$  is the centroid of  $x_i$ 's cluster.

It's a minimization problem of two parts. We first minimize  $J$  w.r.t.  $w_{ik}$  and treat  $\mu_k$  fixed. Then we minimize  $J$  w.r.t.  $\mu_k$  and treat  $w_{ik}$  fixed. Technically speaking, we differentiate  $J$  w.r.t.  $w_{ik}$  first and update cluster assignments (E-step). Then we differentiate  $J$  w.r.t.  $\mu_k$  and recompute the centroids after the cluster assignments from previous step (M-step). Therefore, E-step is:

$$\begin{aligned} \frac{\partial J}{\partial w_{ik}} &= \sum_{i=1}^m \sum_{k=1}^K \|x^i - \mu_k\|^2 \\ \Rightarrow w_{ik} &= \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (2)$$

In other words, assign the data point  $x_i$  to the closest cluster judged by its sum of squared distance from cluster's centroid.

And M-step is:

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{i=1}^m w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^m w_{ik}x^i}{\sum_{i=1}^m w_{ik}} \quad (3)$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments.

**Few things to note here:**

Since clustering algorithms including kmeans use distance-based measurements to determine the similarity between data points, it's recommended to standardize the data to have a mean of zero and a standard deviation of one since almost always the features in any dataset would have different units of measurements such as age vs income.

Given kmeans iterative nature and the random initialization of centroids at the start of the algorithm, different initializations may lead to different clusters since kmeans algorithm may stuck in a local optimum and may not converge to global optimum. Therefore, it's recommended to run the algorithm using different initializations of centroids and pick the results of the run that that yielded the lower sum of squared distance.

Assignment of examples isn't changing is the same thing as no change in within-cluster variation:

$$\frac{1}{m_k} \sum_{i=1}^{m_k} \|x^i - \mu_{c^k}\|^2 \quad (4)$$

**Determining K value using following methods:**

**Elbow method**

Recall that, the basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation [or total within-cluster sum of square (WSS)] is minimized. The total WSS measures the compactness of the clustering and we want it to be as small as possible.

The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve much better the total WSS.

The optimal number of clusters can be defined as follow:

- Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
- For each k, calculate the total within-cluster sum of square (wss).
- Plot the curve of wss according to the number of clusters k.
- The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

**Average silhouette method**

The average silhouette approach we'll be described comprehensively in the chapter cluster validation statistics. Briefly, it measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering.

Average silhouette method computes the average silhouette of observations for different values of  $k$ . The optimal number of clusters  $k$  is the one that maximize the average silhouette over a range of possible values for  $k$  (Kaufman and Rousseeuw 1990).

The algorithm is similar to the elbow method and can be computed as follow:

- Compute clustering algorithm (e.g.,  $k$ -means clustering) for different values of  $k$ . For instance, by varying  $k$  from 1 to 10 clusters.
- For each  $k$ , calculate the average silhouette of observations (avg.sil).
- Plot the curve of avg.sil according to the number of clusters  $k$ .
- The location of the maximum is considered as the appropriate number of clusters.

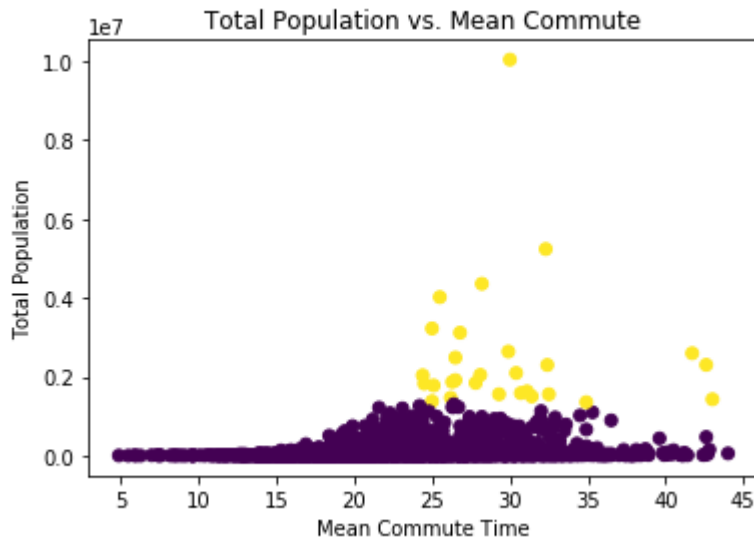
### **The necessity for scaling/standardisation before performing Clustering.**

- In statistics, standardization (sometimes called data normalization or feature scaling) refers to the process of rescaling the values of the variables in your data set so they share a common scale. Often performed as a pre-processing step, particularly for cluster analysis, standardization may be important if you are working with data where each variable has a different unit (e.g., inches, meters, tons and kilograms), or where the scales of each of your variables are very different from one another (e.g., 0-1 vs 0-1000). The reason this importance is particularly high in cluster analysis is because groups are defined based on the distance between points in mathematical space.

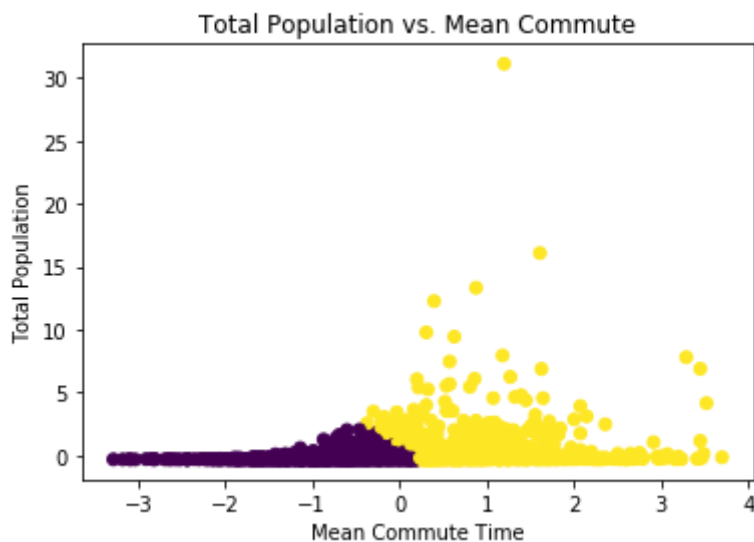
- When you are working with data where each variable means something different, (e.g., age and weight) the fields are not directly comparable. One year is not equivalent to one pound, and may or may not have the same level of importance in sorting a group of records. In a situation where one field has a much greater range of value than another (because the field with the wider range of values likely has greater distances between values), it may end up being the primary driver of what defines clusters. Standardization helps to make the relative weight of each variable equal by converting each variable to a unitless measure or relative distance.

- In our first example, we are interested in performing cluster analysis on Total Population and Mean Commute Time. We would like to use these two variables to split all of the counties into two groups. The units (number of people vs. minutes) and the range of values (85 - 10038388 people vs. 5 - 45 minutes) of these attributes are very different. It is also worth noting that Total Population is a sum, and Mean Commute Time is an average.

When we create clusters with the raw data, we see that Total Population is the primary driver of dividing these two groups. There is an apparent population threshold used to divide the data into two clusters:



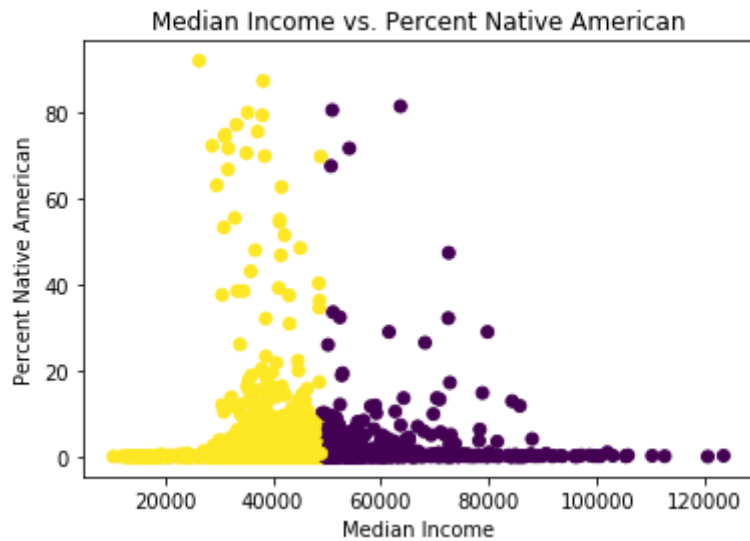
*However, after standardization, both Total Population and Mean Commute seem to have an influence on how the clusters are defined.*



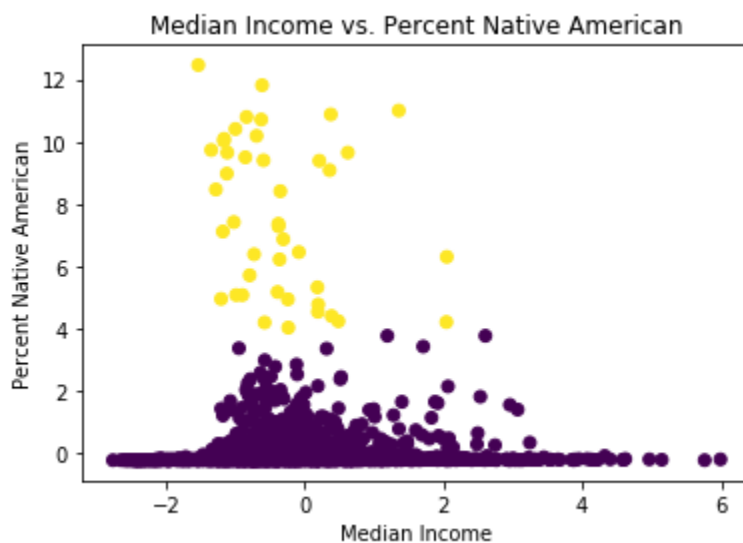
*In this next example, we are interested in clustering on Median Income and Percent of the Population that is Native American (by county). Median Income is measured in dollars and represents the "middle" income for a household in a given county, and Native American is a percentage of the total population for that county. Again, the units and ranges of these variables are very different from one another.*

*When we perform cluster analysis with these two variables without first standardizing, we see that the clusters are primarily split on Income. Income, being measured in dollars, has greater separation in points than percentages, therefore it is the dominant variable.*





*When we standardize the data prior to performing cluster analysis, the clusters change. We find that with more equal scales, the Percent Native American variable more significantly contributes to defining the clusters.*



*Standardization prevents variables with larger scales from dominating how clusters are defined. It allows all variables to be considered by the algorithm with equal importance.*

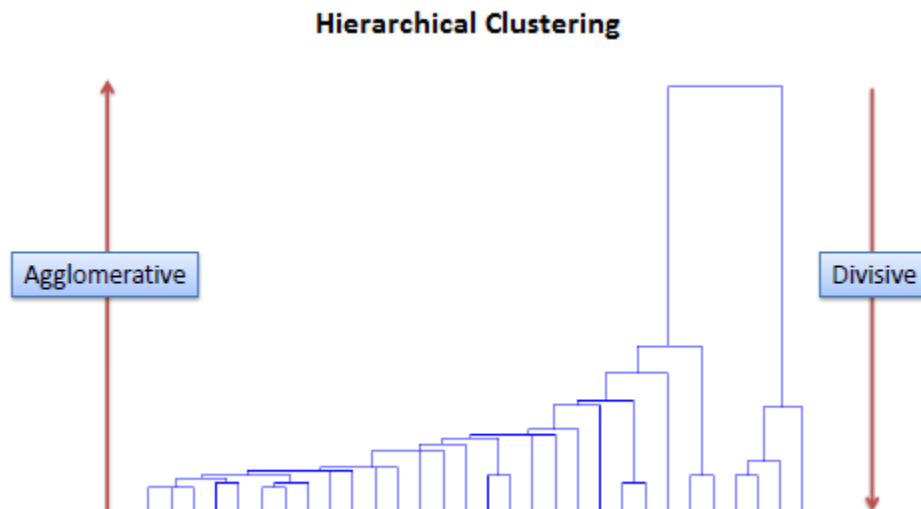
*There are a few different options for standardization, but two of the most frequently used are z-score and unit interval:*

1. [Z-score](#) transforms data by subtracting the mean value for each field from the values of the file and then dividing by the standard deviation of the field, resulting in data with a mean of zero and a standard deviation of one.

2. Unit interval is calculated by subtracting the minimum value of the field and then dividing by the range of the field (maximum minus minimum) which results in a field with values ranging from 0 to 1.

### **Hierarchical Clustering**

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. For example, all files and folders on the hard disk are organized in a hierarchy. There are two types of hierarchical clustering, Divisive and Agglomerative.



#### *Divisive method*

In divisive or top-down clustering method we assign all of the observations to a single cluster and then partition the cluster to two least similar clusters. Finally, we proceed recursively on each cluster until there is one cluster for each observation. There is evidence that divisive algorithms produce more accurate hierarchies than agglomerative algorithms in some circumstances but is conceptually more complex.

#### *Agglomerative method*

In agglomerative or bottom-up clustering method we assign each observation to its own cluster. Then, compute the similarity (e.g., distance) between each of the clusters and join the two most similar clusters. Finally, repeat steps 2 and 3 until there is only a single cluster left. The related algorithm is shown below.

**Given:**

A set  $X$  of objects  $\{x_1, \dots, x_n\}$

A distance function  $dist(c_1, c_2)$

**for**  $i = 1$  to  $n$

$c_i = \{x_i\}$

**end for**

$C = \{c_1, \dots, c_n\}$

$l = n+1$

**while**  $C.size > 1$  **do**

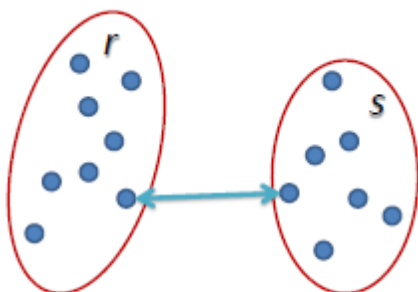
- $(c_{min1}, c_{min2}) = \text{minimum } dist(c_i, c_j) \text{ for all } c_i, c_j \text{ in } C$
- remove  $c_{min1}$  and  $c_{min2}$  from  $C$
- add  $\{c_{min1}, c_{min2}\}$  to  $C$
- $l = l + 1$

**end while**

Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function. Then, the matrix is updated to display the distance between each cluster. The following three methods differ in how the distance between each cluster is measured.

### Single Linkage

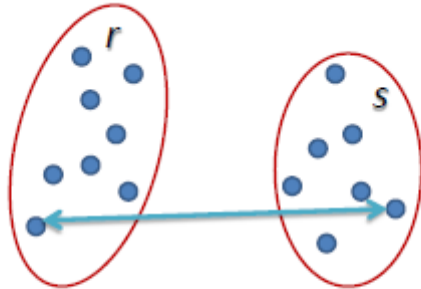
In single linkage hierarchical clustering, the distance between two clusters is defined as the shortest distance between two points in each cluster. For example, the distance between clusters "r" and "s" to the left is equal to the length of the arrow between their two closest points.



$$L(r, s) = \min(D(x_{ri}, x_{sj}))$$

### Complete Linkage

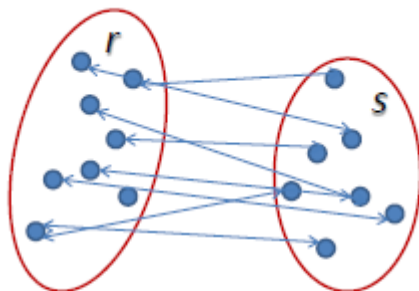
In complete linkage hierarchical clustering, the distance between two clusters is defined as the longest distance between two points in each cluster. For example, the distance between clusters "r" and "s" to the left is equal to the length of the arrow between their two furthest points.



$$L(r, s) = \max(D(x_{ri}, x_{sj}))$$

### **Average Linkage**

In average linkage hierarchical clustering, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster. For example, the distance between clusters "r" and "s" to the left is equal to the average length each arrow between connecting the points of one cluster to the other.



$$L(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

### **Question 3: Principal Component Analysis**

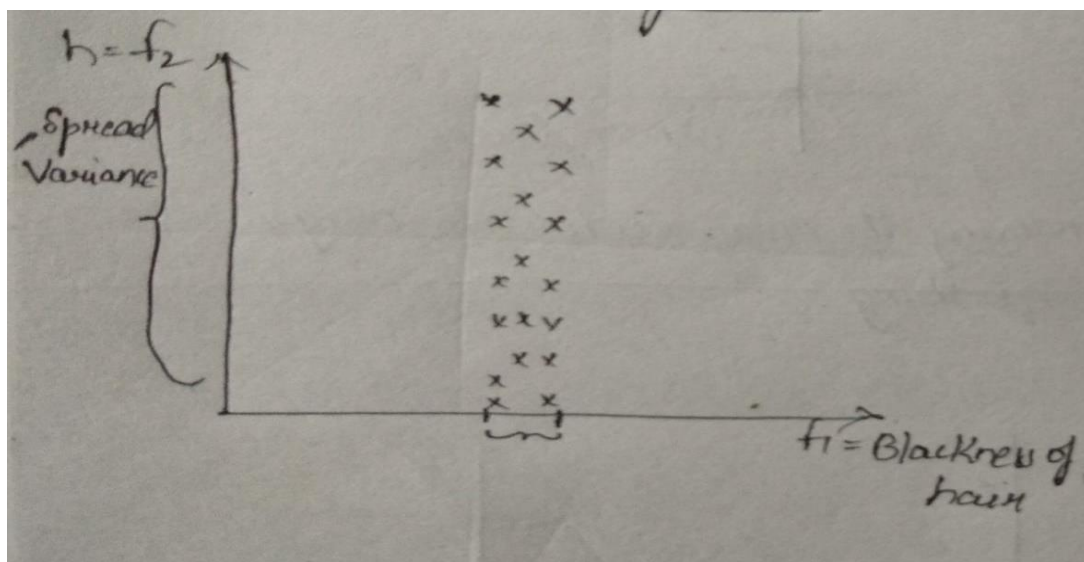
- Give at least three applications of using PCA.
- Briefly discuss the 2 important building blocks of PCA - Basis transformation and variance as information.
- State at least three shortcomings of using Principal Component Analysis.

### **Application of PCA –**

- The primary application of PCA is dimension reduction. If you have high dimensional data, PCA allows you to reduce the dimensionality of your data so the majority of the variation that exists in your data across many high dimensions is captured in fewer dimensions.
- Data visualization is central to the Principal component analysis (PCA), It equally allows you to understand the data and reduce the dimension of data (this is why you have discriminant PCA).
- This method combines highly correlated variables together to form a smaller number of an artificial set of variables which is called "principal components" that account for most variance in the data. Due to which we can combine correlated to one and other will be kept as such.
- PCA is used to remove the least beneficial features so you have a smaller data set, but without losing too much predictive power. That's not to say that there aren't examples where PCA improves accuracy by reducing overfitting.

### PCA building blocks:

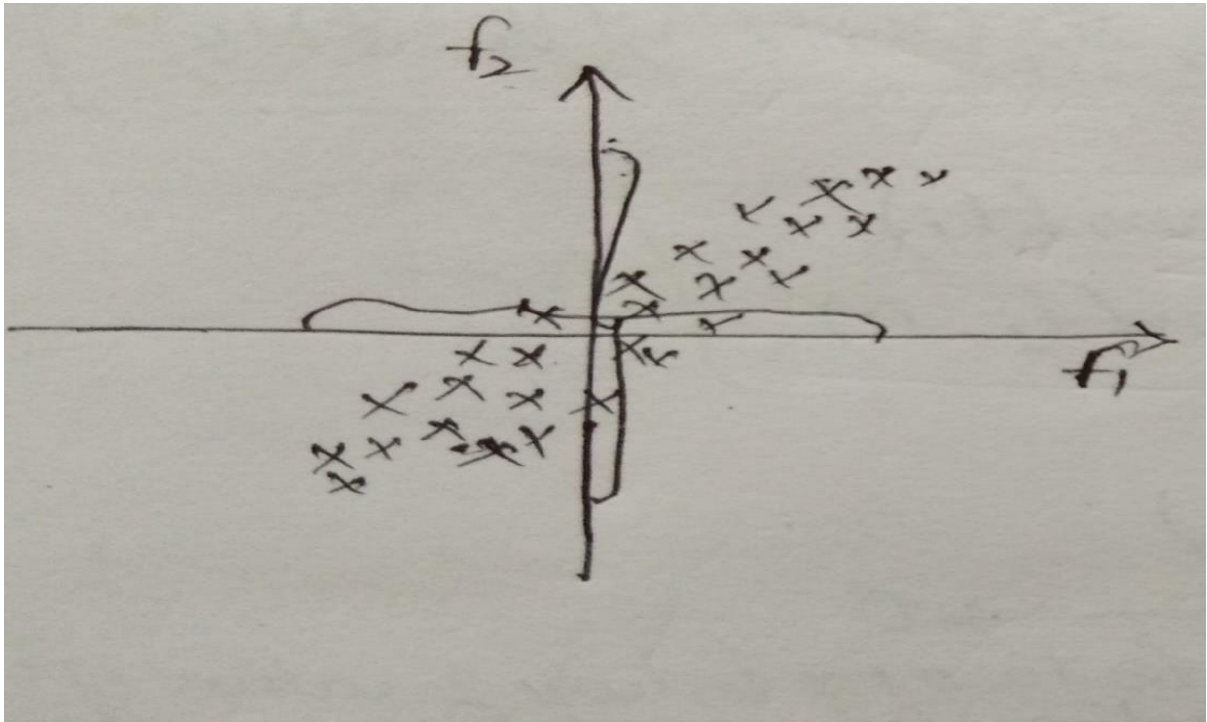
- If we have a very high dimensional dataset like MNIST dataset where there are 784 dimensions, and if we have to visualise it, Of course, we humans can't visualize more than 3 dimensions. This is where PCA comes into play.
- For explanation purpose, we are taking a 2D dataset instead of 784. Let's say we have a dataset with two features  $f_1$  and  $f_2$ .  $f_1$  represents Blackness of hair and  $f_2$  represents the height
- This data is in 2-Dimensions because we are having 2 features. As you can see more data points are spread across feature  $f_2$  than  $f_1$ .



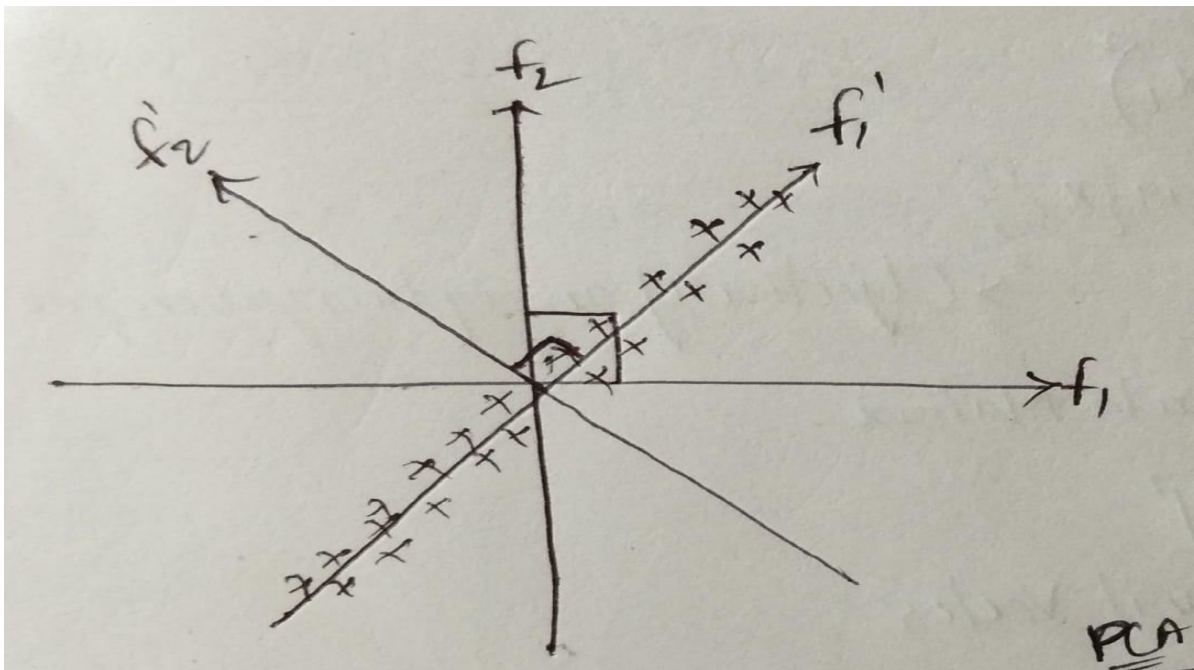
If we are forced to go from 2D to 1D, we can project these points on the feature  $f_2$  and simply say this is the 1D representation of the data since **spread/variance**

As you can see, in such a case both  $f_1$  and  $f_2$  preserve the same spread/variance.

If you choose either  $f_1$  or  $f_2$ , there will be 50% of the information lost.



One idea is that we rotate the axis's  $f_1$  and  $f_2$  as  $f_1'$  and  $f_2'$  respectively, where  $f_1' \perp f_2'$ , such that  $f_1'$  has maximum spread than  $f_2'$  (See the image below):

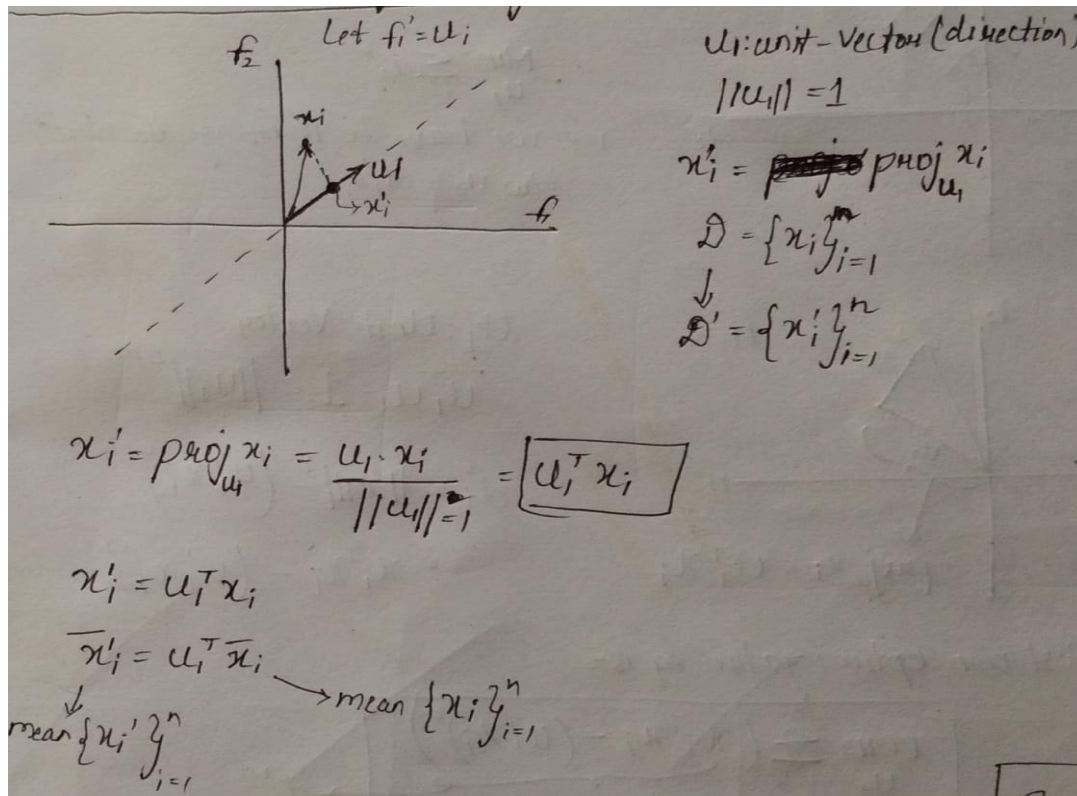




As we found out the axis which has the maximum spread  $f_1'$ . Now we can drop  $f_2'$  and project  $X_i$ 's (datapoints) completely onto  $f_1'$ . Our objective is achieved now i.e going from 2 dimensions to 1 dimension.

This is what essentially PCA does. It tries to find the right direction in which we can preserve more information about the data and dropping unnecessary dimensions.

Mathematical Objective function of PCA



Notations and prerequisite formulas:

1. We'll represent  $f_1'$  i.e the maximum variance axis as " $u_1$ " as most of the explanations online would prefer this notation.
2. So, we want to find the unit vector  $u_1$  i.e  $\|u_1\| = 1$  which preserves maximum variance. i.e., it reduces the interdependency between the variables and increases the variability within the variables.
3. Projection of  $X_i$ 's on  $u_1 = u_1^T X_i$  (This is what we want to find out.)

We assume that data is Column standardized which means,  $\text{mean}(\mu) = 0$  and Standard deviation( $\sigma$ ) = 1.

Now we'll relate the above formula with our problem:

We want to find out  $\mathbf{u}_1$  such that the spread/variance of projected  $X_i$ 's onto  $\mathbf{u}_1$  is maximal. We are now trying to create the principal components which is the combination of variables.

So the function we want to find is as follows (See the equation on the left).

\* find  $\mathbf{u}_1$  s.t  $\text{var}\{\text{proj}_{\mathbf{u}_1} x_i\}_{i=1}^n$  is maximal

$$\text{var}\{\mathbf{u}_1^T x_i\}_{i=1}^n = \underbrace{\frac{1}{n} \sum_{i=1}^n}_{\text{Avg}} \left( \underbrace{\mathbf{u}_1^T x_i}_{x_i} - \underbrace{\mathbf{u}_1^T \bar{x}}_{\text{Mean}\{x_i\}_{i=1}^n} \right)^2$$

~~$x_i$~~   $x_i$ : Col Standardized }  
 $\bar{x} = [0, 0, 0, \dots, 0]$

Since we assumed that data is Column standardized which means,  $\text{mean}=0$ , the  $\mathbf{u}_1^T \bar{x}$  part in the above equation becomes 0.

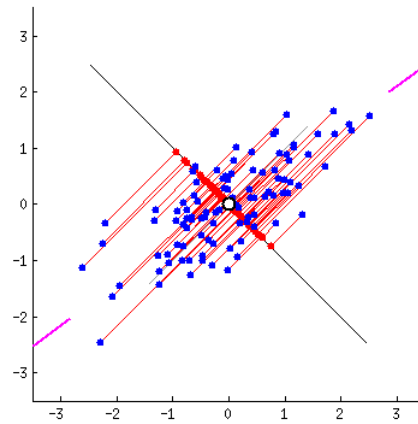
$$\text{var}\{x_i'\}_{i=1}^n = \frac{1}{n} \sum_{i=1}^n (\mathbf{u}_1^T x_i)^2$$

So, this will be the final optimization function that we will solve where we want to find  $\mathbf{u}_1$  where the variance is maximal with a constraint that  $\mathbf{u}_1$  is a unit vector.



$$\max_{u_1} \frac{1}{n} \sum_{i=1}^n (u_1^T x_i)^2$$

$\rightarrow \text{var}\{x_i\}$   
 $\rightarrow \text{Objective of an optimization problem}$   
 $\rightarrow \text{Data Matrix}$   
 $\text{s.t. } u_1^T u_1 = 1 = \|u_1\|^2$   
 $\hookrightarrow \text{Constraint: } u_1 \text{ is a unit vector}$   
 $\rightarrow \text{i.e. Find } u_1 \text{ which maximizes projected variance.}$



As you can see from the above visual, we have to minimize the distance of points  $x_i$ 's to  $u_1$  in this distance formulation.

$$u_1: \text{Unit Vector}$$

$$u_1^T u_1 = 1 = \|u_1\|^2$$

$$d_i^2 = \|x_i\|^2 - (u_1^T x_i)^2$$

$$= x_i^T x_i - (u_1^T x_i)^2$$

The adjacent distance is nothing but projection of  $X_i$  onto  $u_1$ , the hypotenuse is the length of  $X_i$ . Now we can easily find out the distance b/w  $X_i$  and  $u_1$  i.e  $d_i$  from Pythagoras' theorem.

Now the final distance optimization function is as follows, where we want to find  $u_1$  by minimizing the distance of a point  $X_i$  to  $u_1$ .

So, the dist min optimization eq is:-

$$\min_{u_1} \sum_{i=1}^n \left( x_i^T x_i - (u_1^T x_i)^2 \right) \rightarrow d_i^2$$

s.t  $u_1^T u_1 = 1$

The previous optimization function is Variance maximization, while this one is distance minimization.

### **Eigen Values and Eigen Vectors(Basis and variance transformation)**

Solution to our optimization function can be attained by using eigen values and vectors represented by  $(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_d)$  and  $(V_1, V_2, V_3, \dots, V_d)$  respectively. Remember Eigen values are scalars. Eigen vector define the basis.

There's a simple function in Sklearn library, wherein you have to give the Covariance matrix( $S$ ) to it and it returns you eigen values and vectors correspondingly where  $\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_d$ .

eigen-values of  $S = \lambda_1, \lambda_2, \dots, \lambda_d$

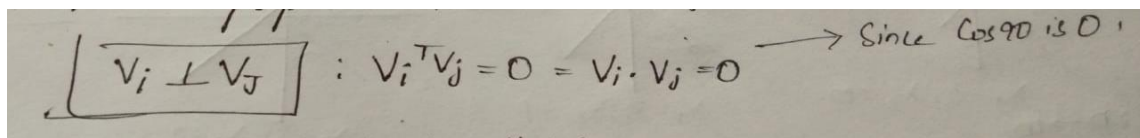
eigen-vectors of  $S = v_1, v_2, \dots, v_d$

where;  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \geq \lambda_d$    
→ Maximal eigen-value

That means the vector corresponding to  $\lambda_1$  i.e  $V_1$  has the highest variance explained, then  $\lambda_2$  i.e  $V_2$  which is the second most variance explained vector and so on.

So the  $u_1$  which we are trying to obtain from the optimization function is nothing but  $V_1$  here.

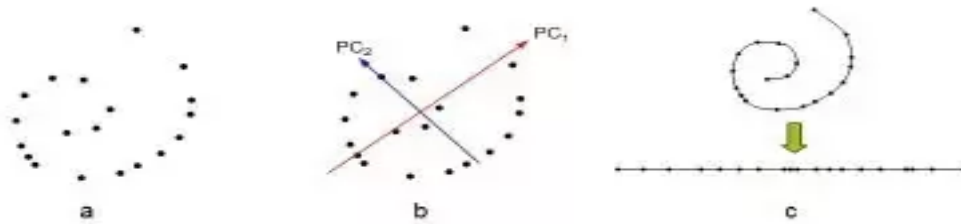
One of the very nice properties is that every pair of eigen vectors are perpendicular to each other.


$$\boxed{V_i \perp V_j} : V_i^T V_j = 0 = V_i \cdot V_j = 0 \rightarrow \text{Since } \cos 90 \text{ is } 0$$

That means if we take  $V_1$  and  $V_2$  i.e the top two vectors corresponding to top eigen values, it is similar to obtaining  $f_1'$  and  $f_2'$ . Similarly, we can mention 'd' as 700 or 300 or 100 or 2 or 1 or whatever number of dimensions you want to reduce to and it will return the top dimensions of your dataset.

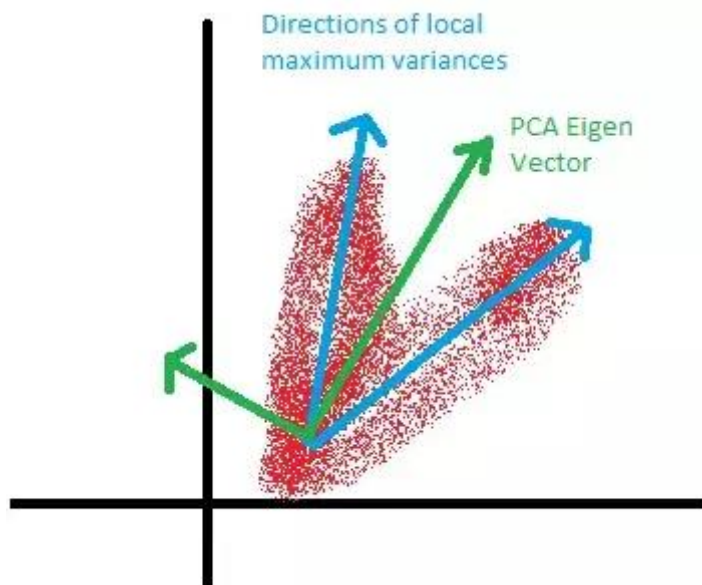
Shortcomings of using Principal Component Analysis.

- Assume that, we have two-dimensional data (i.e., two features) and the joint distribution of the data follows multivariate normal distribution. One of the important properties of multivariate normal distribution is that, if the correlation between the features is zero, it means that features are orthogonal. The main job of PCA is to represent the data in lower dimensional by removing the redundant features. It achieves that through finding **orthogonal** principal components. The above property is not applicable if the joint distribution of data (not individual distribution of feature) follows other distribution instead of multivariate normal distribution. We also use a covariance matrix (covariance matrix is a function of correlation matrix) to find the principal components. The only one distribution (zero-mean probability distribution) which allows us to represent the whole data in a compact form is Gaussian distribution. It shows that PCA make an implicit assumption that data should follows Gaussian distribution. If data didn't follow Gaussian distribution, it would be difficult to extract independent statistical components by PCA.
- The standard PCA always finds linear principal components to represent the data in lower dimension. Sometime, we need non-linear principal components.



If we apply standard PCA for the above data, it will fail to find good representative direction.

- As I said before, PCA always finds orthogonal principal components. Sometimes, our data demands non-orthogonal principal components to represent the data.



The green color vectors are principal components. But, the actual maximum variance directions are blue color vectors. PCA fails to find that vectors. But, Independent Component Analysis (ICA) works well for the above data and it gives the blue color vectors as independent components

- PCA always considered the low variance components in the data as noise and recommend us to throw away those components. But, sometimes those components play a major role in supervised learning task.