Question 1: Assignment Summary

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(why you took that many numbers of principal components, which type of Clustering produced a better result and so on)

Note: You don't have to include any images, equations or graphs for this question. Just text should be enough.

Ans:

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.

Solution Methodology:

- Importing the Data (Country-data.csv) file.
- Understanding the Data set
    - Converting the columns Exports, health and imports variables which are given in % of GDP to Absolute values.
    - Check for any missing values in data set

- Dropping the country column from data set before proceeding to Scaling step.

- Scaling the data set
  - Perform the standard scaling before proceeding for PCA.
  - Go ahead for PCA step by dropping the country column

- Dimensionality Reduction
  - Perform PCA step on the scaled dataset.
  - Get the cumulative sum of the explained variance ratio after performing the PCA step.
  - Plot Scree plot and it clearly tells us almost 90 % of cumm explained variance is covered by 3 principal components in the data set.
  - Perform the PCA with 3 Components

- Outlier Treatment
  - We have identified the outliers on all the 3 components
  - Removed the outliers on all the 3 components based on Statistical Technique
  - Proceed further to Clustering after removing outliers from the dataset.

- Determine Optimal value of K for K-means Clustering
  - Check for Hopkins measure (0.81) which is good to perform clustering on Data set.
  - Plot elbow curve to determine the optimal value of K to perform the K-Means Clustering Algorithm
    - Plot clearly shows that K=3 is the optimal value
  - Lets check the silhouette analysis on different clusters.
    - For n_clusters=2, the silhouette score is 0.45946256207584163
    - For n_clusters=3, the silhouette score is 0.5171734664552139
    - For n_clusters=4, the silhouette score is 0.4374615104958549
    - It clearly shows that k=3 is the Optimal value
  - Perform K-means Clustering with K=3 on the data set and check for results.
  - Scatter Plot clearly shows the clear division of clusters (0,1,2) on PCA components.
  - Box plots clearly shows the variations how each Principal Components (PC1,PC2,PC3) varies against each cluster in the dataset (0,1,2)

- K means Clustering Observations
  - Developed Countries:

- The countries under Cluster id ‘0’ is having low child_mort and High GDPP and High Income.
- Developing Countries:
- The countries under Cluster id ‘1’ is having medium child_mort,medium GDPP and medium Income.
- Under Developed Countries:
- The countries under Cluster id ‘2’ is having high child_mort, low GDPP and low Income.

Hierarchal Clustering Approach

- Perform Hierarchical Clustering with complete linkage and plot dendrogram.
- Cut the tree with 3 clusters and assign the labels.
- Scatter Plot clearly shows the clear division of clusters (0,1,2) on PCA components.
- Box plots clearly shows the variations how each Principal Components (PC1,PC2,PC3) varies against each cluster in the dataset (0,1,2)
    - Developed Countries:
    - The countries under Cluster id ‘2’ is having low child_mort and High GDPP and High Income.
    - Developing Countries:
    - The countries under Cluster id ‘1’ is having medium child_mort,medium GDPP and medium Income.
    - Under Developed Countries:
    - The countries under Cluster id ‘0’ is having high child_mort, low GDPP and low Income.
- I am going with K-means as both the clustering algorithms showing the same count for the under developed countries which are in direst aid which is 44.
- The countries which are in dire need of aid based on K- Means clustering are.
- 'Afghanistan',
- 'Angola',
- 'Benin',
- 'Botswana',
- 'Burkina Faso',
- 'Burundi',
- 'Cameroon',

- 'Chad',
- 'Comoros',
- 'Congo, Dem. Rep.',
- 'Congo, Rep.',
- "Cote d'Ivoire",
- 'Equatorial Guinea',
- 'Eritrea',
- 'Gabon',
- 'Gambia',
- 'Ghana',
- 'Guinea',
- 'Guinea-Bissau',
- 'Iraq',
- 'Kenya',
- 'Kiribati',
- 'Lao',
- 'Lesotho',
- 'Liberia',
- 'Madagascar',
- 'Malawi',
- 'Mali',
- 'Mauritania',
- 'Mozambique',
- 'Namibia',
- 'Niger',
- 'Pakistan',
- 'Rwanda',
- 'Senegal',
- 'Sierra Leone',

- 'Solomon Islands',
- 'South Africa',
- 'Sudan',
- 'Tanzania',
- 'Togo',
- 'Uganda',
- 'Yemen',
- 'Zambia'

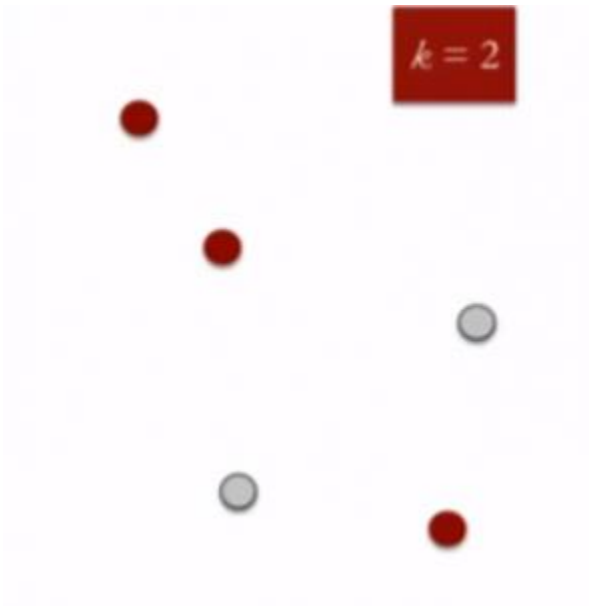
Question 2: Clustering

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

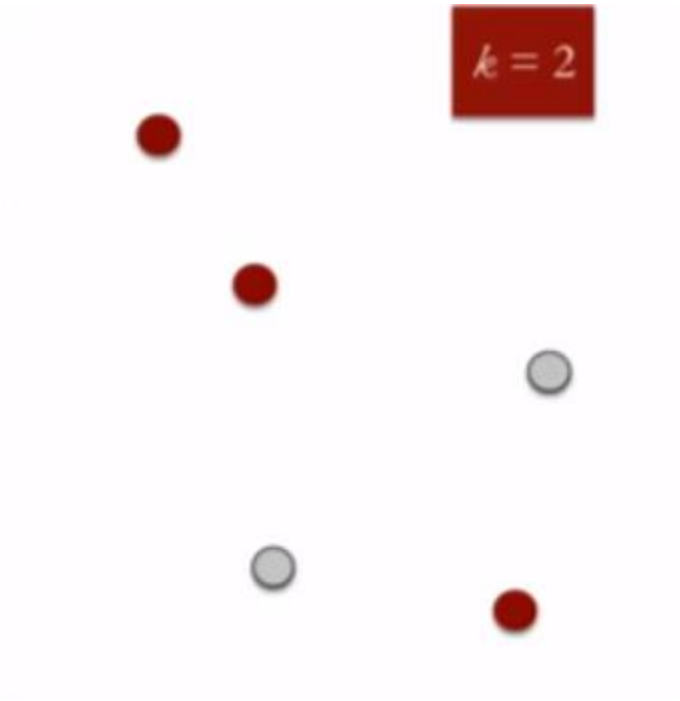
Ans: K Means Clustering

K means is an iterative clustering algorithm that aims to find local maxima in each iteration. This algorithm works in these 5 steps :

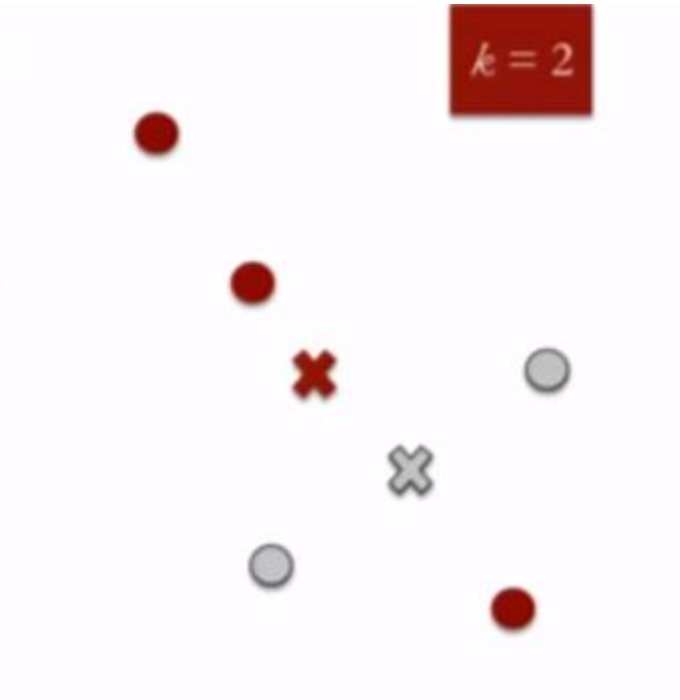
1. Specify the desired number of clusters K : Let us choose k=2 for these 5 data points in 2-D space.



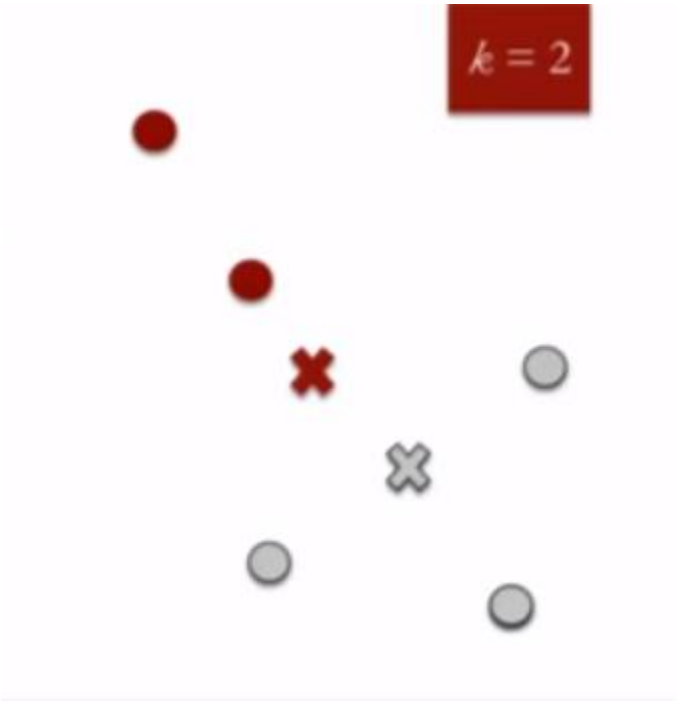
2. Randomly assign each data point to a cluster : Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.



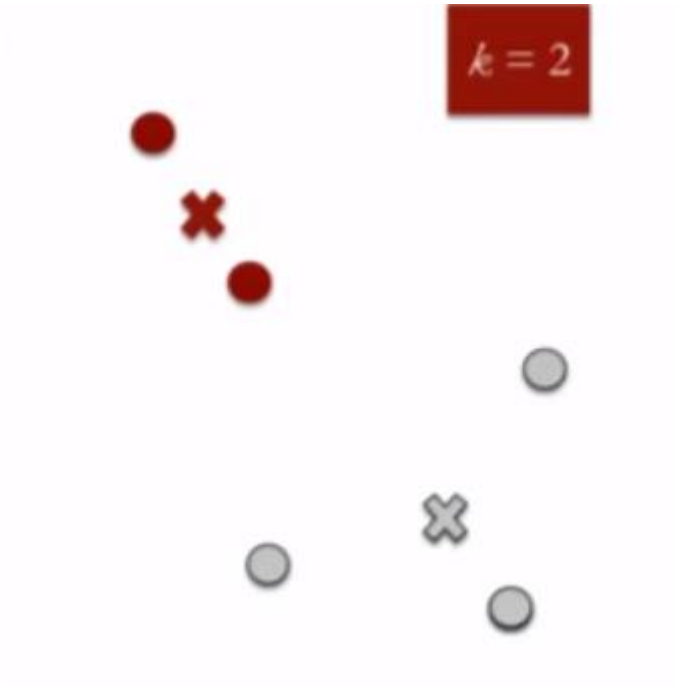
3. Compute cluster centroids : The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. Re-assign each point to the closest cluster centroid : Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster



5. Re-compute cluster centroids : Now, re-computing the centroids for both the clusters.

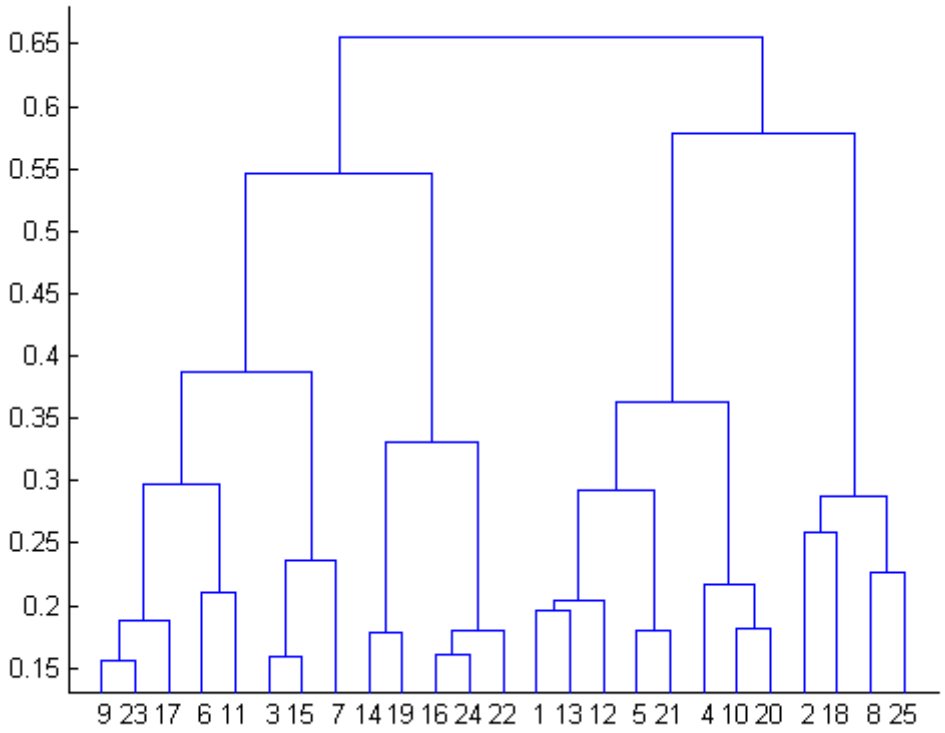


6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

Hierarchical Clustering

Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

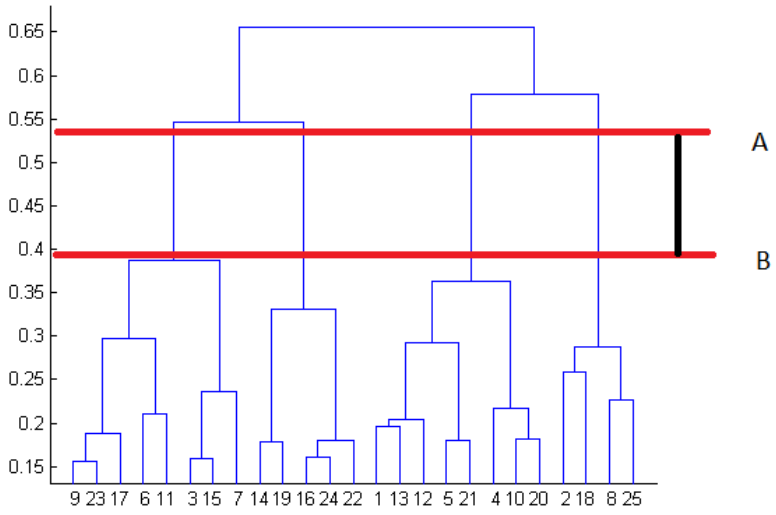
The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:



At the bottom, we start with 25 data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.

The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram. The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.

In the above example, the best choice of no. of clusters will be 4 as the red horizontal line in the dendrogram below covers maximum vertical distance AB.



Two important things that you should know about hierarchical clustering are:

- This algorithm has been implemented above using bottom up approach. It is also possible to follow top-down approach starting with all data points assigned in the same cluster and recursively performing splits till each data point is assigned a separate cluster.
- The decision of merging two clusters is taken on the basis of closeness of these clusters. There are multiple metrics for deciding the closeness of two clusters :
    - Euclidean distance: $||a-b||_2 = \sqrt{(\Sigma(a_i-b_i))}$
    - Squared Euclidean distance: $||a-b||_2^2 = \Sigma((a_i-b_i)^2)$
    - Manhattan distance: $||a-b||_1 = \Sigma|a_i-b_i|$
    - Maximum distance:$||a-b||_{INFINITY} = max_i|a_i-b_i|$
    - Mahalanobis distance: $\sqrt{((a-b)^T S^{-1} (-b))}$ {where, s : covariance matrix}

Difference between K Means and Hierarchical clustering

- Hierarchical clustering can't handle big data well but K Means clustering can. This is because the time complexity of K Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. $O(n^2)$.
- In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram

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

Ans:

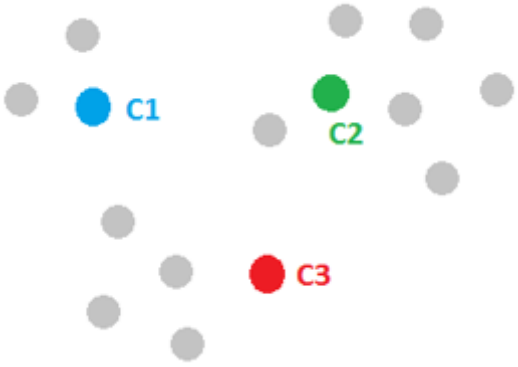
Clustering via K-means

Among all the unsupervised learning algorithms, clustering via k-means might be one of the simplest and most widely used algorithms. Briefly speaking, k-means clustering aims to find the set of k clusters such that every data point is assigned to the closest center, and the sum of the distances of all such assignments is minimized.

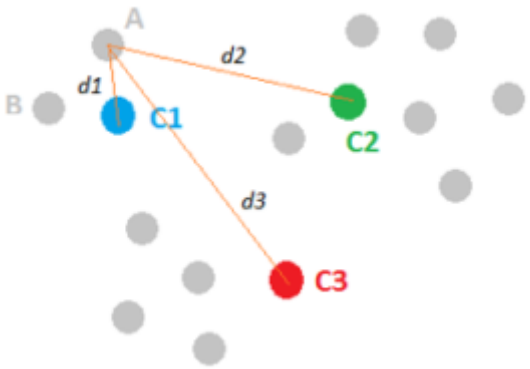
Let's walk through a simple 2D example to better understand the idea. Imaging we have these gray points in the following figure and want to assign them into three clusters. K-means follows the four steps listed below.

Step one: Initialize cluster centers

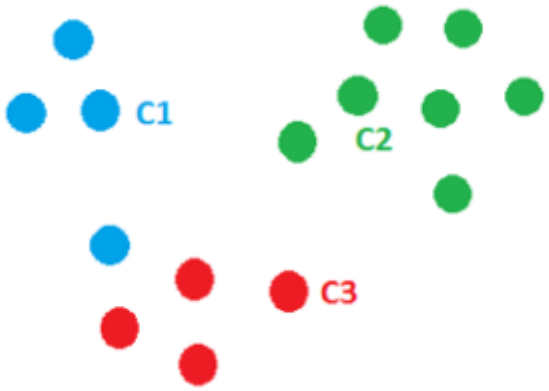
We randomly pick three points C1, C2 and C3, and label them with blue, green and red color separately to represent the cluster centers.



Step two: Assign observations to the closest cluster center

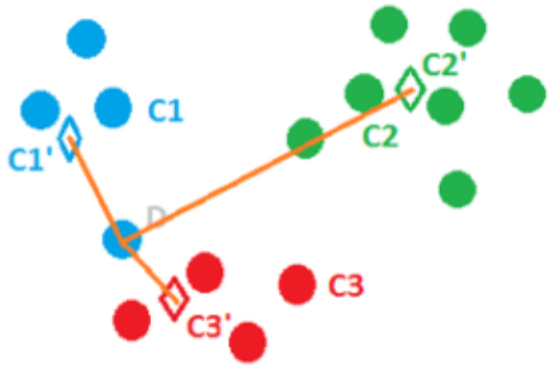


Once we have these cluster centers, we can assign each point to the clusters based on the minimum distance to the cluster center. For the gray point A, compute its distance to C1, C2 and C3, respectively. And after comparing the lengths of *d1*, *d2* and *d3*, we figure out that *d1* is the smallest, therefore, we assign point A to the blue cluster and label it with blue. We then move to point B and follow the same procedure. This process can assign all the points and leads to the following figure.



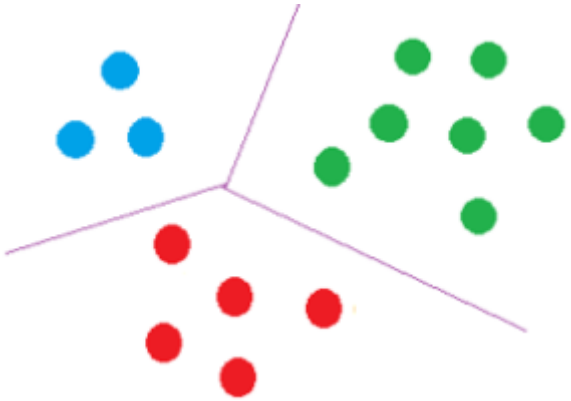
Step three: Revise cluster centers as mean of assigned observations

Now we've assigned all the points based on which cluster center they were closest to. Next, we need to update the cluster centers based on the points assigned to them. For instance, we can find the center mass of the blue cluster by summing over all the blue points and dividing by the total number of points, which is four here. And the resulted center mass C1', represented by a blue diamond, is our new center for the blue cluster. Similarly, we can find the new centers C2' and C3' for the green and red clusters.



Step four: Repeat step 2 and step 3 until convergence

The last step of k-means is just to repeat the above two steps. For example, in this case, once C1', C2' and C3' are assigned as the new cluster centers, point D becomes closer to C3' and thus can be assigned to the red cluster. We keep on iterating between assigning points to cluster centers, and updating the cluster centers until convergence. Finally, we may get a solution like the following figure. Well done!



Some Additional Remarks about K-means

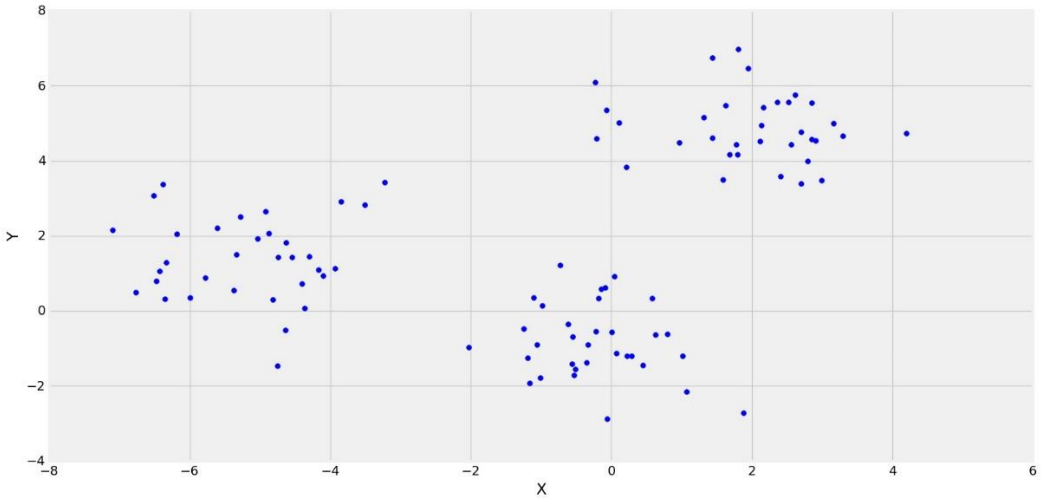
- The k-means algorithm converges to local optimum. Therefore, the result found by K-means is not necessarily the most optimal one.
- The initialization of the centers is critical to the quality of the solution found. There is a smarter initialization method called K-means++ that provides a more reliable solution for clustering.
- The user has to select the number of clusters ahead of time.

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

Statitical Aspect:

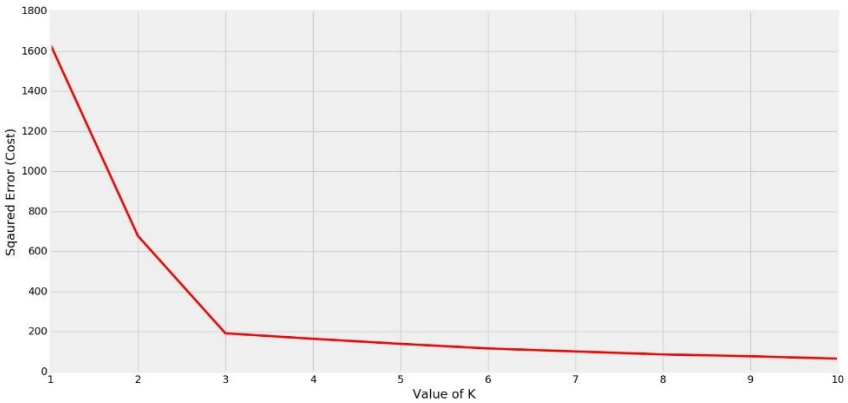
Elbow Curve:

There is a popular method known as **elbow method** which is used to determine the optimal value of K to perform the K-Means Clustering Algorithm. The basic idea behind this method is that it plots the various values of cost with changing *k*. As the value of *K* increases, there will be fewer elements in the cluster. So average distortion will decrease. The lesser number of elements means closer to the centroid. So, the point where this distortion declines the most is the **elbow point**.



3 clusters are forming

In the above figure, its clearly observed that the distribution of points are forming 3 clusters. Now, let's see the plot for the squared error(Cost) for different values of K.



Elbow is forming at K=3

Clearly the elbow is forming at K=3. So the optimal value will be 3 for performing K-Means.

Silhouette Algorithm to determine the optimal value of k

One of the fundamental steps of an unsupervised learning algorithm is to determine the number of clusters into which the data may be divided. The silhouette algorithm is one of the many algorithms to determine the optimal number of clusters for an unsupervised learning technique.

In the Silhouette algorithm, we assume that the data has already been clustered into k clusters by a clustering technique(Typically K-Means Clustering technique). Then for each data point, we define the following:-

C(i) -The cluster assigned to the ith data point

|C(i)| – The number of data points in the cluster assigned to the ith data point

a(i) – It gives a measure of how well assigned the ith data point is to it's cluster

$$a(i) = \frac{1}{|C(i)|-1} \sum_{C(i), i \neq j} d(i,j)$$

b(i) – It is defined as the average dissimilarity to the closest cluster which is not it's cluster

$$b(i) = min_{i \neq j} \left(\frac{1}{|C(j)|} \sum_{j \epsilon C(j)} d(i,j) \right)$$

The silhouette coefficient s(i) is given by:-

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

We determine the average silhouette for each value of k and for the value of k which has the **maximum value of s(i)** is considered the optimal number of clusters for the unsupervised learning algorithm.
Let us consider the following data:-

S.NO	X1	X2
1.	-7.36	6.37
2.	3.08	-6.78

S.NO	X1	X2
3.	5.03	-8.31
4.	-1.93	-0.92
5.	-8.86	6.60

We now iterate the values of k from 2 to 5. We assume that no practical data exists for which all the data points can be optimally clustered into 1 cluster.

We construct the following tables for each value of k:-

k = 2

S.NO	A(I)	B(I)	S(I)
1.	5.31	14.1	0.62
2.	2.47	13.15	0.81
3.	2.47	14.97	0.84
4.	9.66	8.93	-0.076
5.	5.88	19.16	0.69

Average value of s(i) = 0.58
k = 3

S.NO	A(I)	B(I)	S(I)
1.	1.52	9.09	0.83

S.NO	A(I)	B(I)	S(I)
2.	2.47	7.71	0.68
3.	2.47	10.15	0.76
4.	0	7.71	1
5.	1.52	17.93	0.92

Average value of s(i) = 0.84
k = 4

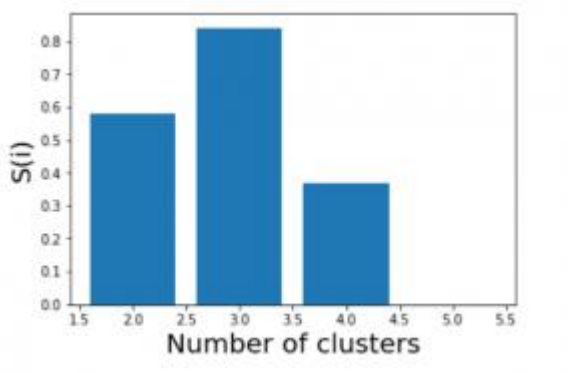
S.NO	A(I)	B(I)	S(I)
1.	1.52	9.09	0.83
2.	infinite	2.47	0
3.	infinite	2.47	0
4.	infinite	7.71	0
5.	1.52	10.23	0.85

Average value of s(i) = 0.37
k = 5

S.NO	A(I)	B(I)	S(I)
1.	infinite	1.52	0
2.	infinite	2.47	0

S.NO	A(I)	B(I)	S(I)
3.	infinite	2.47	0
4.	infinite	7.71	0
5.	infinite	1.52	0

Average value of s(i) = 0



We see that the highest value of s(i) exists for k = 3. Therefore we conclude that the optimal number of clusters for the given data is 3.

Business Aspect:

In addition to Statastical aspect we need to consider the business aspect as well in taking the number of clusters let see in the countries that are in direst need of the funding in the assignment, some times we take business consideration of identifying the nations based on developed, developing and under developed. It's not only just taking the statistical observations we just need to take care of business aspect as well and business aspect will be having the upper edge over the statistical aspect while considering the number of clusters.

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

Ans:

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

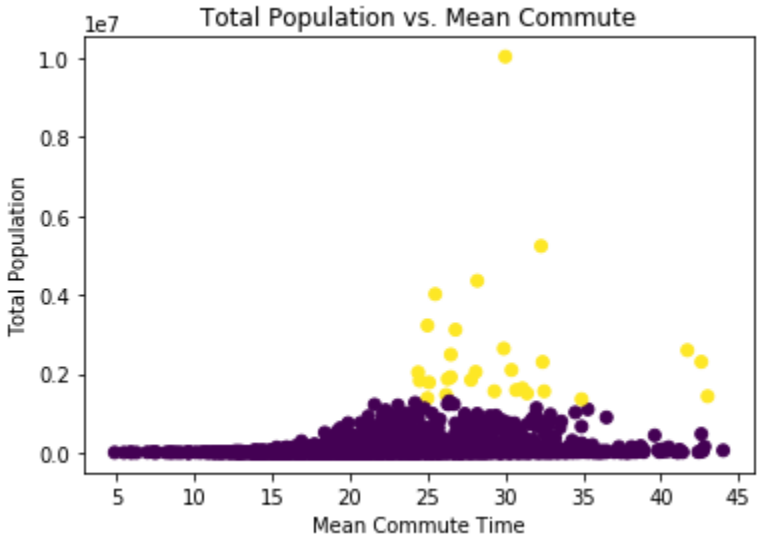
analysis is because groups are defined based on the distance between points in mathematical space.

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.

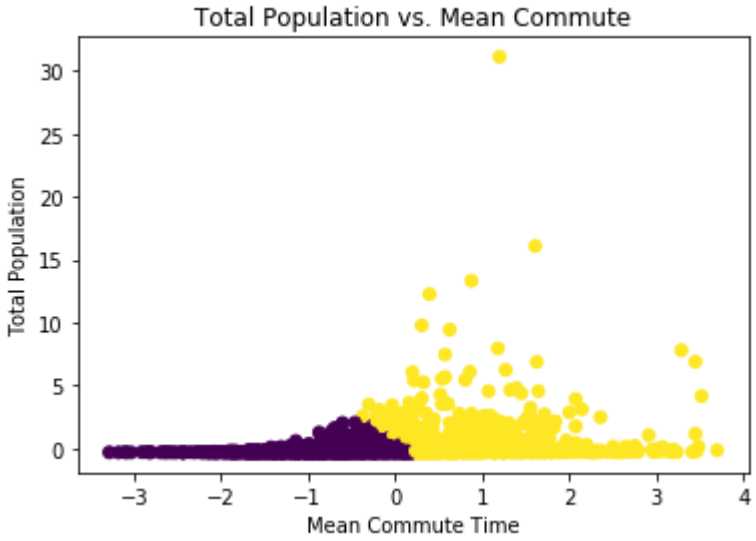
What follows is a couple examples demonstrating how standardization may impact a clustering solution, using the 2015 US Census Demographic Dataset, downloaded from Kaggle. This dataset includes different demographic variables for counties in the United States, including population, race, income, poverty, commute distance, commute method, as well as variables describing employment.

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:

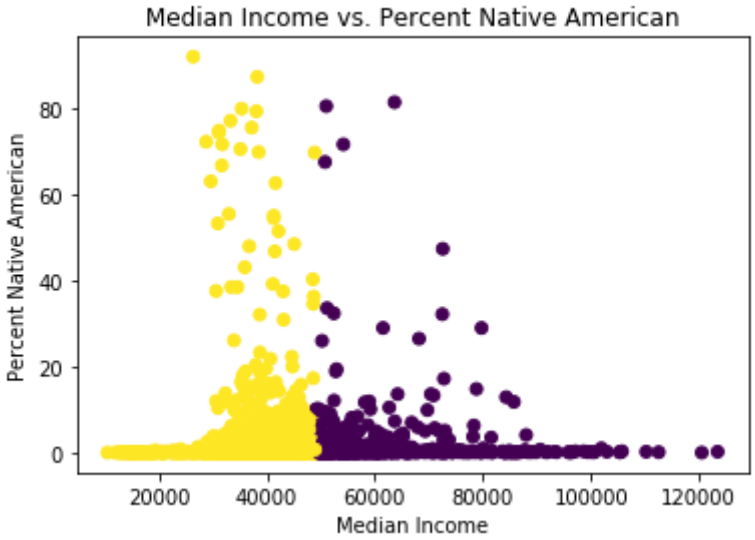


However, after standardization, both Total Population and Mean Commute seem to have an influence on how the clusters are defined.

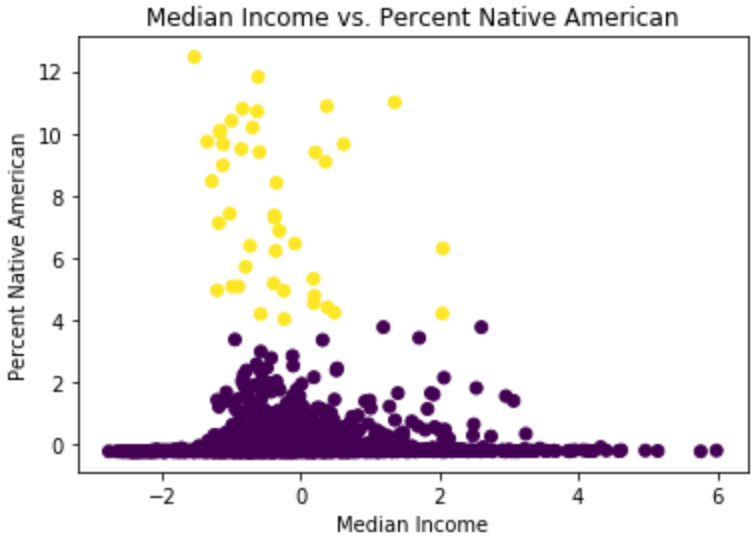


In this next example, we are interested in clustering on Median Income and Percent of the Population that is Native American (by county). Median Income is measured in dollars and represents the "middle" income for a household in a given county, and Native American is a percentage of the total population 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.

Although standardization is considered best practice for cluster analysis, there are circumstances where standardization may not be appropriate for your data (e.g., Latitude and Longitude). If you'd like to read more there are a few great discussions on this topic on the Statistics and Data Science forums of Stack Exchange, as well as this academic article on Standardization and Its Effects on K-Means Clustering Algorithm by Ismail Bin Mohamad and Dauda Usman.

As always, the golden rule is to know thy data. Only you will know if standardization is right for your use case.

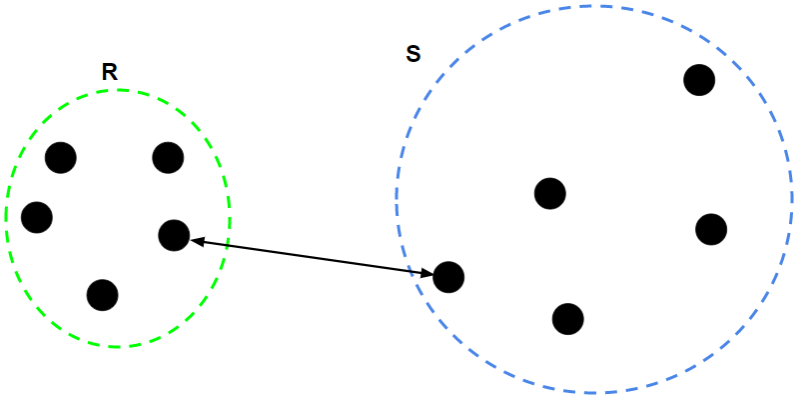
e) Explain the different linkages used in Hierarchical Clustering.

Ans:

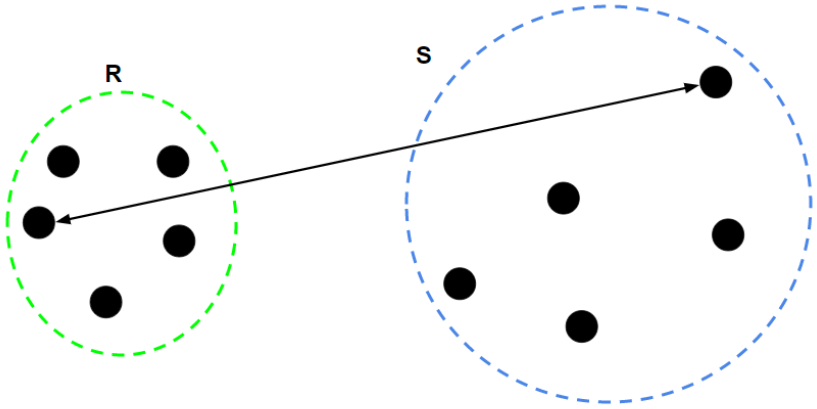
The process of Hierarchical Clustering involves either clustering sub-clusters(data points in the first iteration) into larger clusters in a bottom-up manner or dividing a larger cluster into smaller sub-clusters in a top-down manner. During both the types of hierarchical clustering, the distance between two sub-clusters needs to be computed. The different types of linkages describe the different approaches to

measure the distance between two sub-clusters of data points. The different types of linkages are:-

1. **Single Linkage:** For two clusters R and S, the single linkage returns the minimum distance between two points i and j such that i belongs to R and j belongs to S.



2. **Complete Linkage:** For two clusters R and S, the single linkage returns the maximum distance between two points i and j such that i belongs to R and j belongs to S.

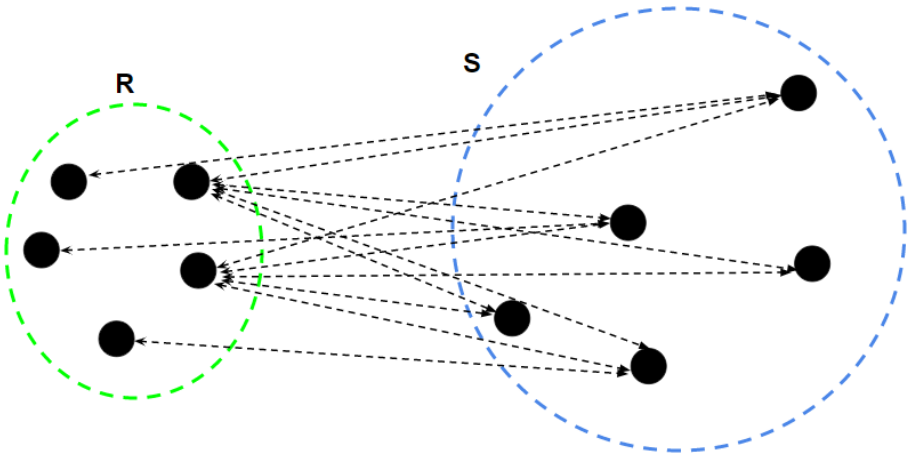


3. **Average Linkage:** For two clusters R and S, first for the distance between any data-point i in R and any data-point j in S and then the arithmetic mean of these distances are calculated. Average Linkage returns this value of the arithmetic mean.

where

– Number of data-points in R

– Number of data-points in S

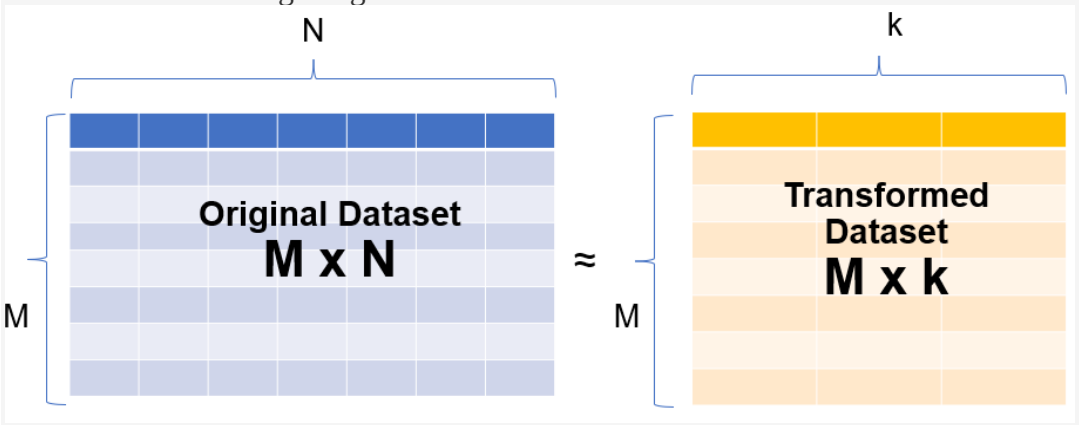


Question 3: Principal Component Analysis

a) Give at least three applications of using PCA.

Ans:

Fundamentally, PCA is a dimensionality reduction technique, i.e., it approximates the original data set to a smaller one containing fewer dimensions. To understand it visually, take a look at the following image.



In the image above, you can see that a data set having N dimensions has been approximated to a smaller data set containing 'k' dimensions. And this simple manipulation helps in several ways such as follows:

- For data visualisation and EDA
- For creating uncorrelated features that can be input to a prediction model: With a smaller number of uncorrelated features, the modelling process is faster and more stable as well.
- Finding latent themes in the data: If you have a data set containing the ratings given to different movies by Netflix users, PCA would be able to find latent themes like genre and, consequently, the ratings that users give to a particular genre.
- Noise reduction

b) Briefly discuss the 2 important building blocks of PCA - Basis transformation and variance as information.

Ans:

Change Basis:

So the given dataset A is as follows

X	Y
2	2
3	4
4	5
5	8
6	11
7	12
8	15
9	16
10	19
11	21

and the associated covariance matrix is given by

$$cov(A)=[9.1619.6119.6142.23]$$

In this case,
we've $cov(x,x)=9.16$, $cov(y,y)=42.23$ and $cov(x,y)=cov(y,x)=19.61$

From here, we can observe that there is some correlation between the X and Y columns. Let's see what happens to the covariance matrix when we take another basis to represent the same points

Covariance Matrix in New Basis

Let's say that the same friend, who earlier gave you the smart suggestion in the roadmap example comes around again and asks you to represent your data in the following basis

$$New\ Basis=[0.420\ 0.906\ 0.906\ -0.420]$$

He tells you that you'll observe something interesting with the covariance matrix when the data is represented in this basis. Let's check out your friend's claims.

So as per your friend's suggestion, you went ahead and transformed the dataset and computed the covariance matrix for the new dataset A'. The new covariance matrix is as follows:

$$cov(A') = [51.30.030.030.049]$$

As you can see now the covariance matrix has changed a lot. $cov(x_{new}, x_{new})$ is much higher than $cov(y_{new}, y_{new})$. One important thing that has happened here is that $cov(x_{new}, y_{new})$ is now ~0. What does that mean? This means that x and y columns have very less correlation.

Well, now that the 2 features are **nearly uncorrelated**, there is no dependence or correlation of one direction on the other. This process of converting the covariance matrix with only non-zero diagonal elements and 0 values everywhere else is also known as **diagonalisation**.

Creating Uncorrelated Features

Now you must be wondering "How does finding new basis vectors where the covariance matrix only has non-zero values along the diagonal and 0 elsewhere help us?"

Well now,

- Your new basis vectors are all uncorrelated and independent of each other.
- Since variance is now explained only by the new basis vectors themselves, you can find the directions of maximum variance by checking which value is higher than the rest numerically. There is no correlation to take into account this time. All the information in the dataset is explained by the columns themselves.

So now, your new basis vectors are **uncorrelated**, **hence linearly independent**, and **explain the directions of maximum variance.**

As a matter of fact, these basis vectors are the **Principal Components** of the original matrix. The algorithm of PCA seeks to find those new basis vectors that diagonalise the covariance when the same data is represented on this new basis. And then these vectors would have all the above properties that we require and therefore would help us in the process of dimensionality reduction.

Variance as Information

As mentioned in the video, you need to do find something known as **eigenvectors** of the covariance matrix using a process called **Eigendecomposition**. These eigenvectors would be the new set of basis vectors in whose representation, the covariance matrix will be diagonalized. Therefore these would be the new principal components.

Here's a brief overview of what eigendecomposition is and how it is applied on the covariance matrix to give us the eigenvectors. Please note that though it is recommended that you understand the text below you can perform the eigendecomposition in a couple of steps in Python.

An overview of Eigendecomposition

Basically, for any square matrix A, its **eigenvectors** are all the vectors v which satisfy the following equation:

$$Av=\lambda v$$

λ is some constant also known as the **eigenvalue** for that particular eigenvector.

For example, for the square matrix $A=[2005]$, its eigenvectors and corresponding eigenvalues are given by as follows

$$[2005][10]=2[10]$$

$$[2005][01]=5[01]$$

Here A has 2 sets of eigenvectors/eigenvalues -

$$v_1=[10],\ \lambda_1=2 \text{ and } v_2=[01],\ \lambda_2=5$$

The process of finding the eigenvalues and eigenvectors of a matrix is known as eigendecomposition.

In general, if the size of matrix A is n (i.e. it is a n x n matrix) then, there will be at most n eigenvectors that can be formed. As you saw in the above case, A is a 2 x 2 matrix and hence it had 2 eigenvectors.

Note: This is a brief overview of what eigenvectors and eigenvalues are and it is sufficient for you right now in the context of PCA. If you want to learn further about eigenvectors and how they're found out, please go through the additional link mentioned here.

Eigendecomposition of Covariance Matrix

Now, continuing from the previous segment, we wanted to find a new set of basis vectors where the covariance matrix gets diagonalised. It turns out that these new set of basis vectors are in fact the eigenvectors of the Covariance Matrix. And therefore these eigenvectors are the Principal Components of our original dataset. In other words, these eigenvectors are the directions that capture maximum variance.

But what about the eigenvalues? What do they signify?

Well, the **eigenvalues** are indicators of **the variance explained by that particular direction** or eigenvector. So higher is the eigenvalue, higher is the variance explained by that eigenvector and hence that direction is more important for us.

Therefore to summarise,

- The eigendecomposition of the covariance matrix C yields us the eigenvectors $V1, V2, V3....$ with their corresponding eigenvalues as $\lambda_1, \lambda_2, \lambda_3....$ ...
- When you use the eigenvectors as the new set of basis vectors and transform the original dataset to this new basis, your covariance matrix will now be diagonalised.
- These **eigenvectors** are the **Principal Components** of the original dataset. $V1$ is **Principal Component 1**, $V2$ is **Principal Component 2** and so on.
- The eigenvectors are ordered on the basis of their eigenvalues, to signify the variance explained by them. Or you can say $\lambda_1 > \lambda_2 > \lambda_3....$. Higher the eigenvalue, higher is the amount of variance captured by the eigenvector. Hence the maximum variance is explained by Principal Component 1, the second-highest variance is explained by Principal Component 2 and so on.

c) State at least three shortcomings of using Principal Component Analysis.

Ans:

Some important shortcomings of PCA:

- PCA is limited to linearity, though we can use **non-linear techniques such as t-SNE** as well (you can read more about t-SNE in the optional reading material below).

- PCA needs the components to be perpendicular, though in some cases, that may not be the best solution. The alternative technique is to use **Independent Components Analysis.**
- PCA assumes that columns with low variance are not useful, which might not be true in prediction setups (especially classification problem with a high class imbalance).