Machine Learning Assignment - 4

- Swati Sharma (2021568)

Computational Part (Section C) [Theoretical Part is at the end]

a. EDA (Exploratory Data Analysis) on the country dataset

i.

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689	2.947964	12964.155689
std	40.328931	27.412010	2.746837	24.209589	19278.067698	10.570704	8.893172	1.513848	18328.704809
min	2.600000	0.109000	1.810000	0.065900	609.000000	-4.210000	32.100000	1.150000	231.000000
25%	8.250000	23.800000	4.920000	30.200000	3355.000000	1.810000	65.300000	1.795000	1330.000000
50%	19.300000	35.000000	6.320000	43.300000	9960.000000	5.390000	73.100000	2.410000	4660.000000
75%	62.100000	51.350000	8.600000	58.750000	22800.000000	10.750000	76.800000	3.880000	14050.000000
max	208.000000	200.000000	17.900000	174.000000	125000.000000	104.000000	82.800000	7.490000	105000.000000

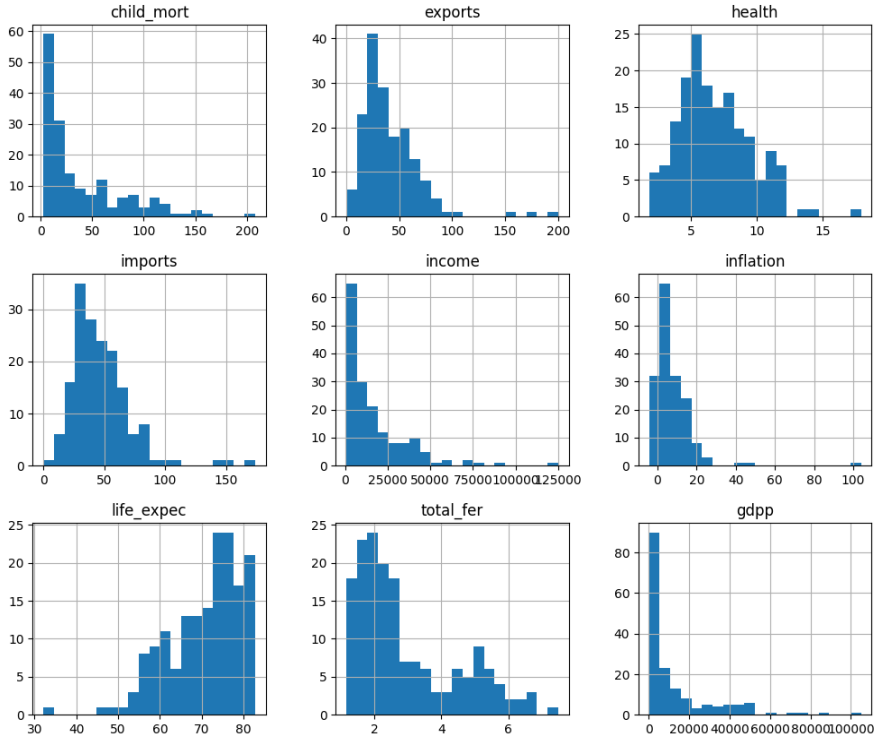
Not much deviation in 'health' and 'life_expec'

ii.

```
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

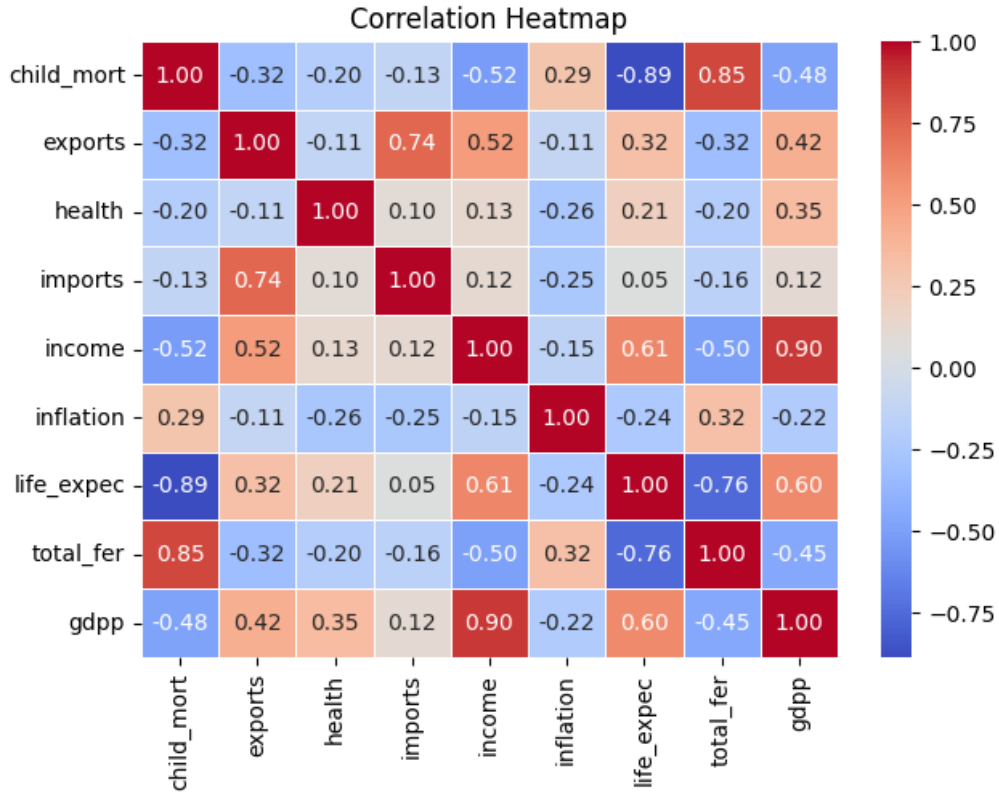
No missing values

iii.



Distribution of different column values

iv.



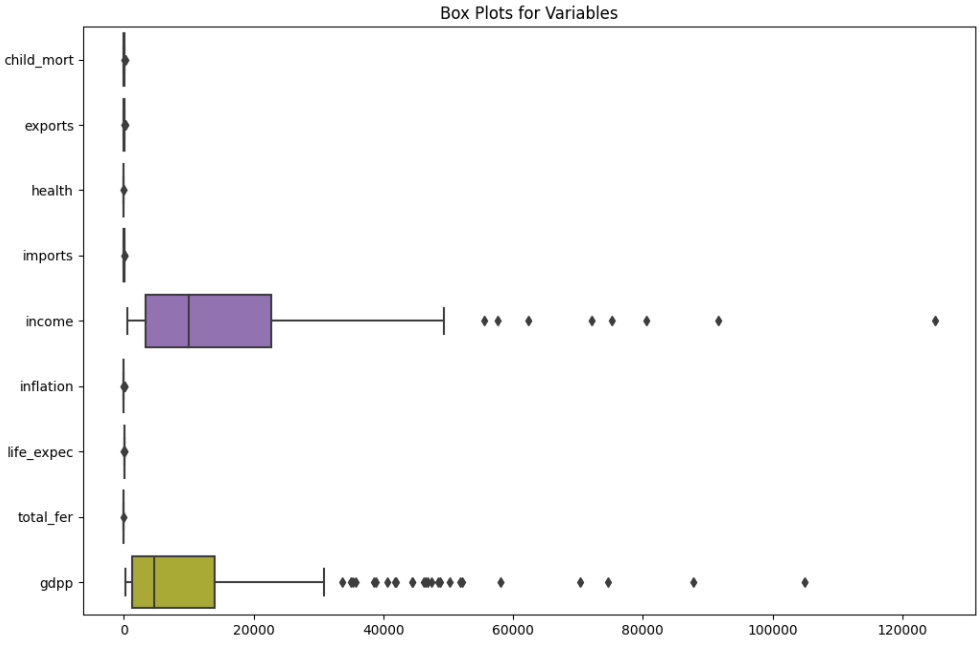
Strong Positive Correlation between:

1. gdpp and income (0.90)
2. total_fer and child_mort (0.85)
3. imports and exports (0.74)
4. income and life_expec (0.61)
5. gdpp and life_expec (0.60)

Strong Negative Correlation between:

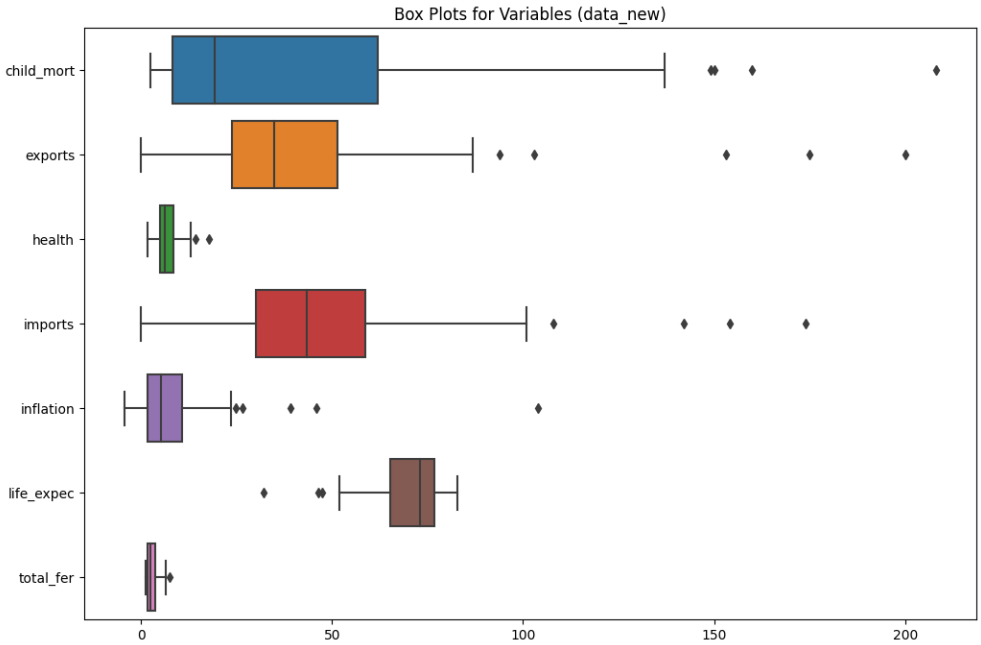
1. child_mort and life_expec (-0.89)
2. total_fed and life_expec (-0.76)

v.



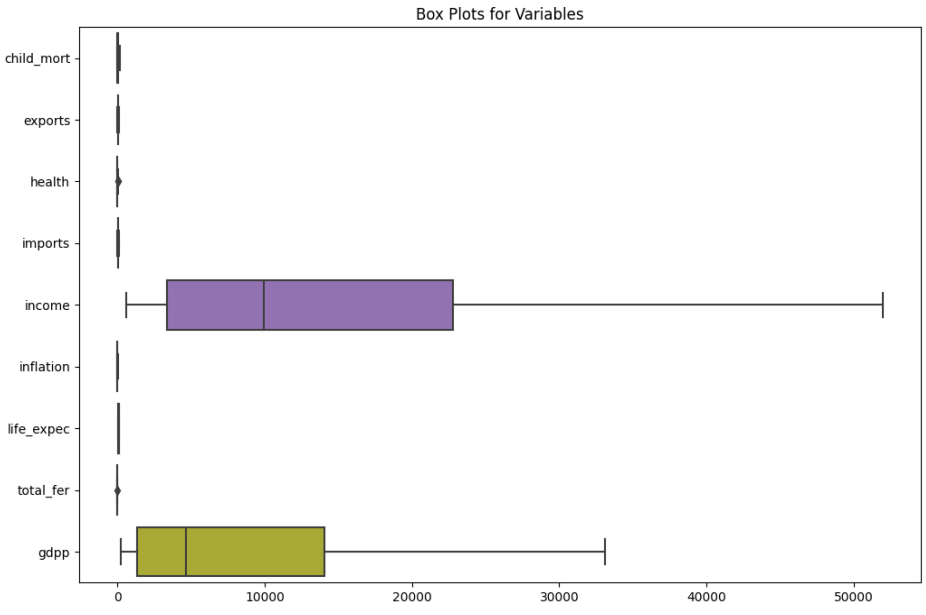
Many outliers are presented in the ‘income’ and ‘gdpp’ column.

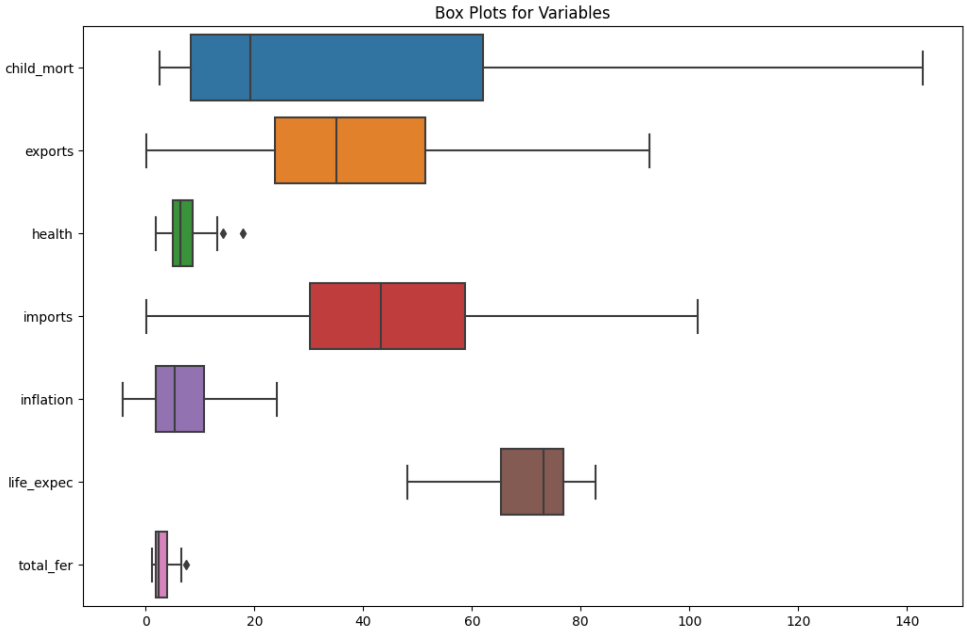
vi.



1. We removed the columns = [‘income’, ‘gdpp’] and plotted the box plot
2. Many outliers are presented in the 'child_mort', 'exports', 'imports' and 'inflation'

vii.

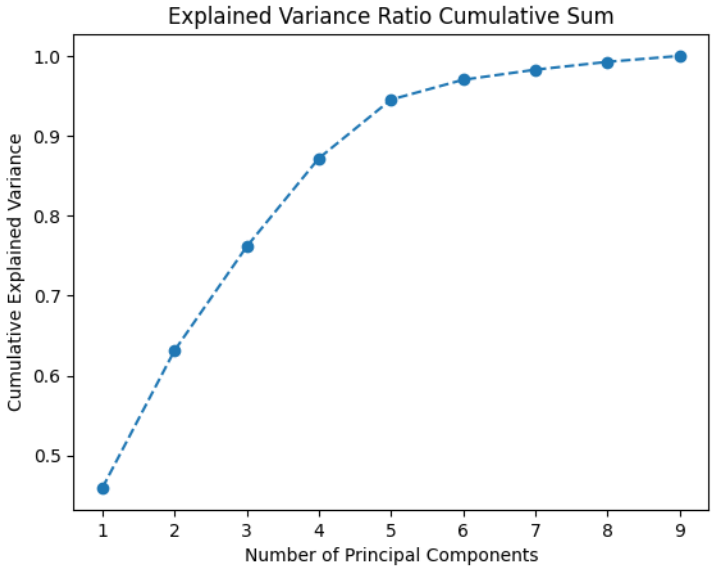




1. After removing the outliers using the IQR score, there were no outliers present.

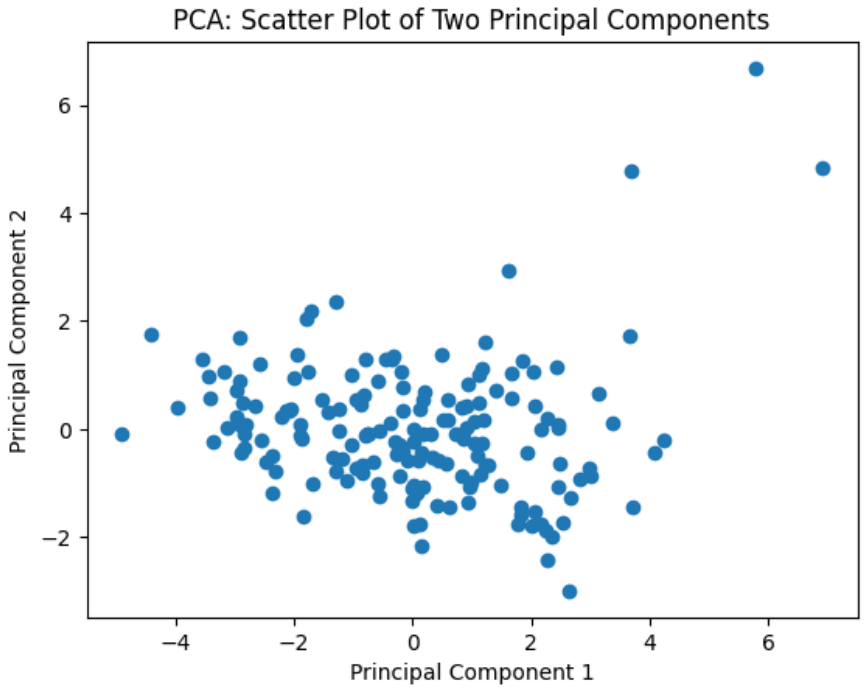
viii. Label Encoding was done for the ‘country’ column which gave every row a unique ID.

b. <u>PCA (Principal Component Analysis)</u>

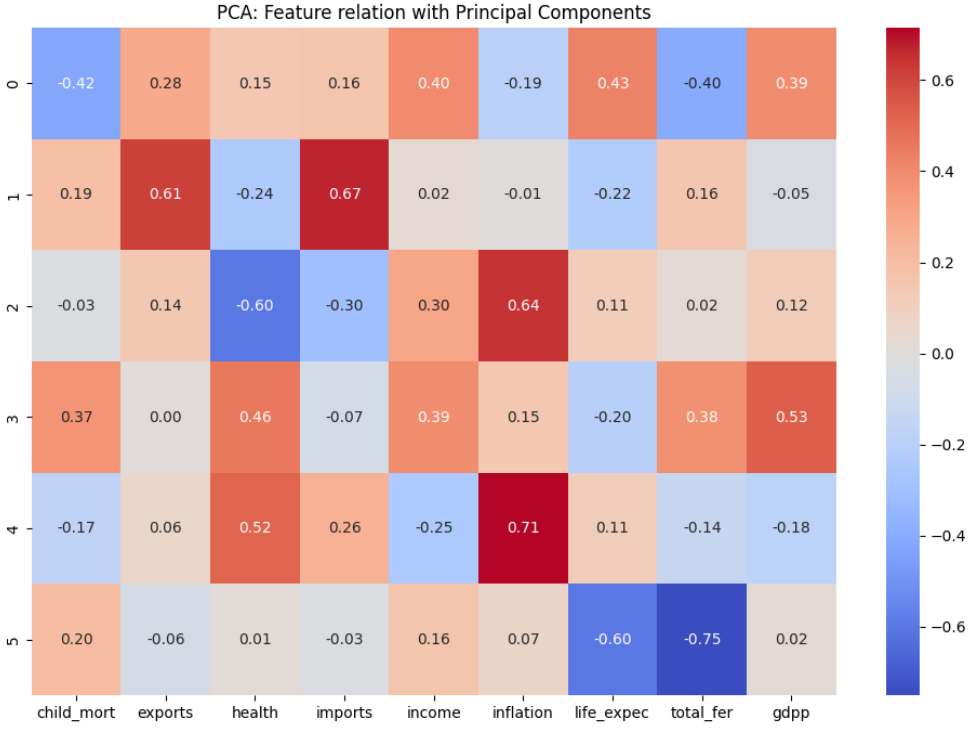


i. **Optimal number of components: 6**

ii.

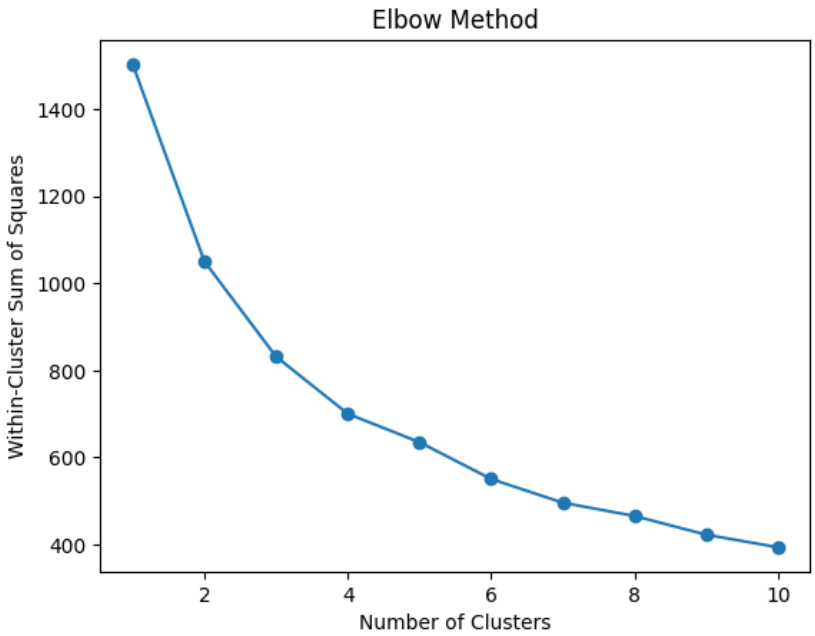


iii.

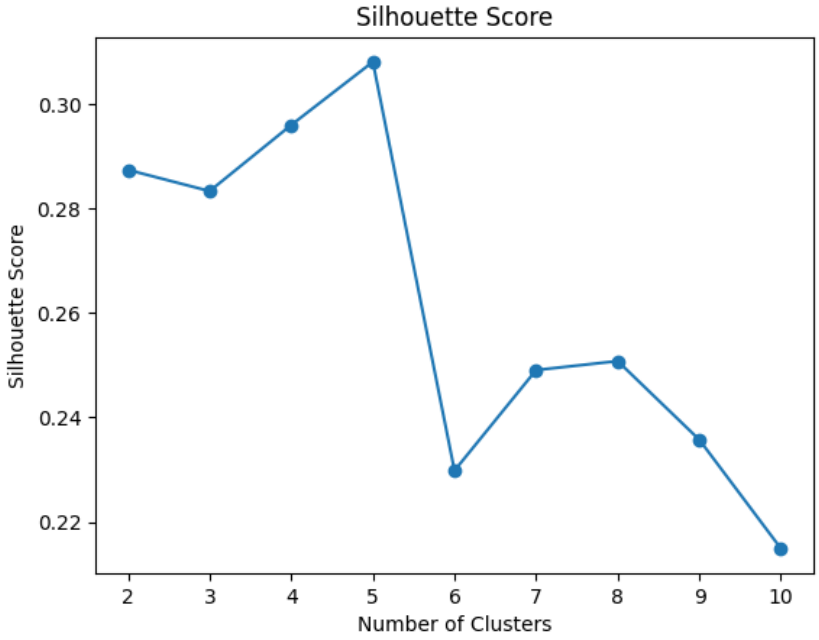


c. <u>K-means Clustering Algorithm</u>

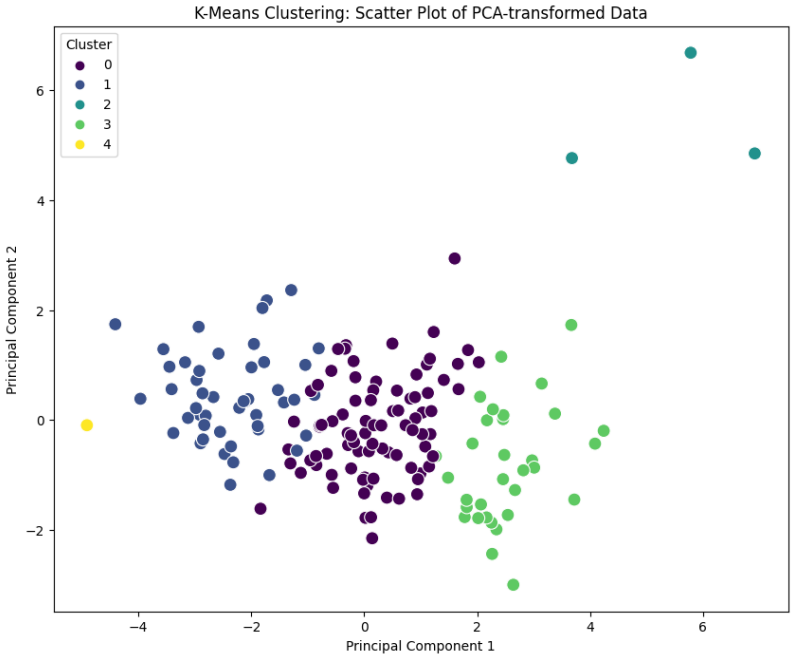
i.



ii.

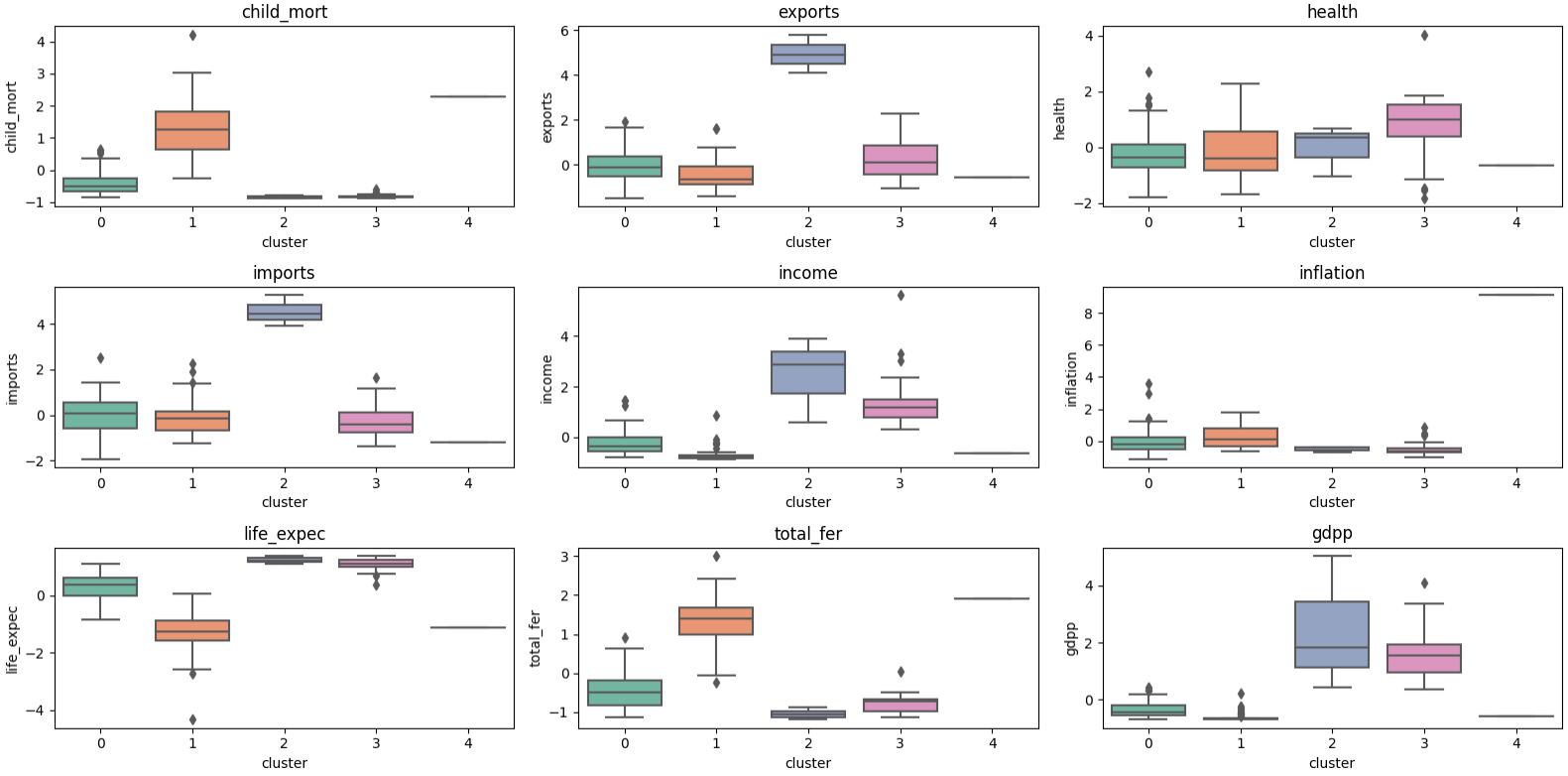


Optimal number of components: 5

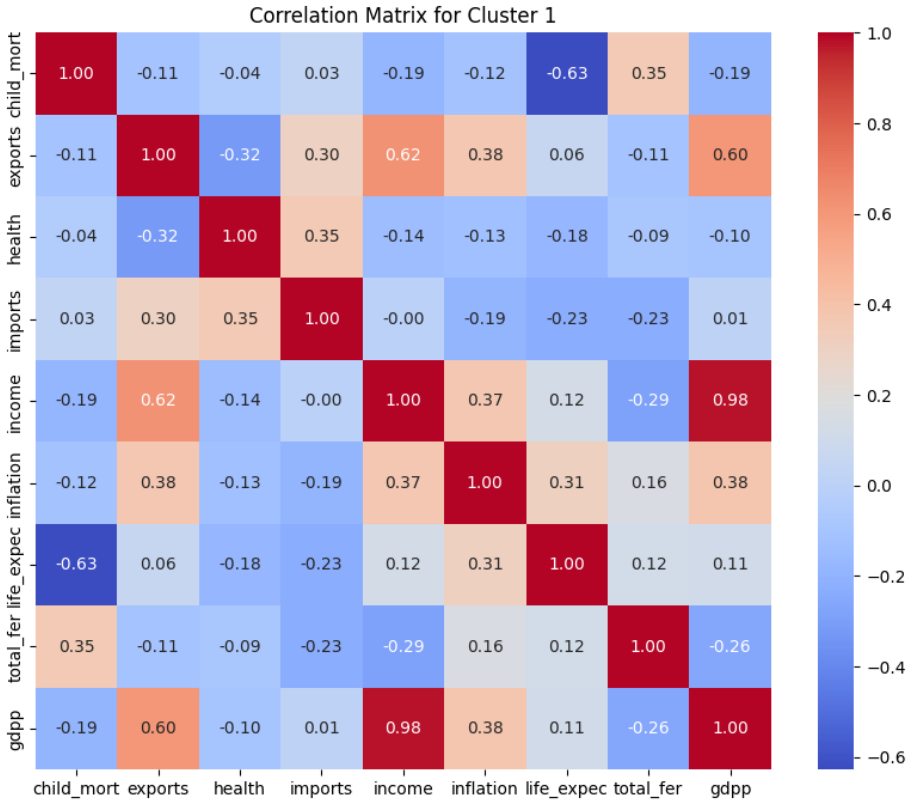


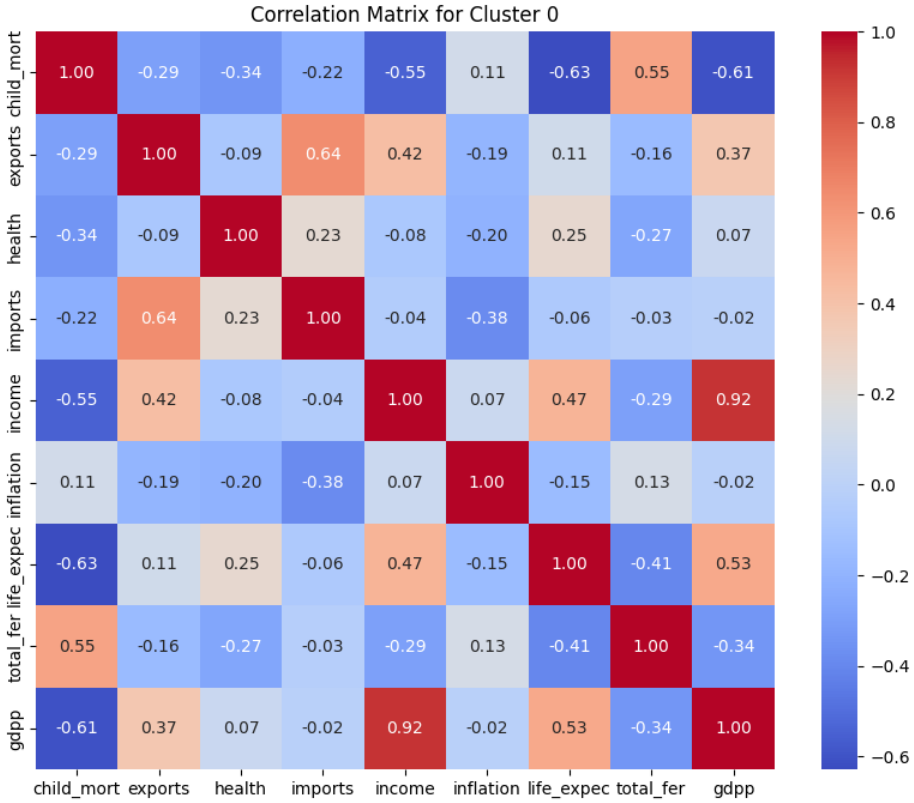
iii.

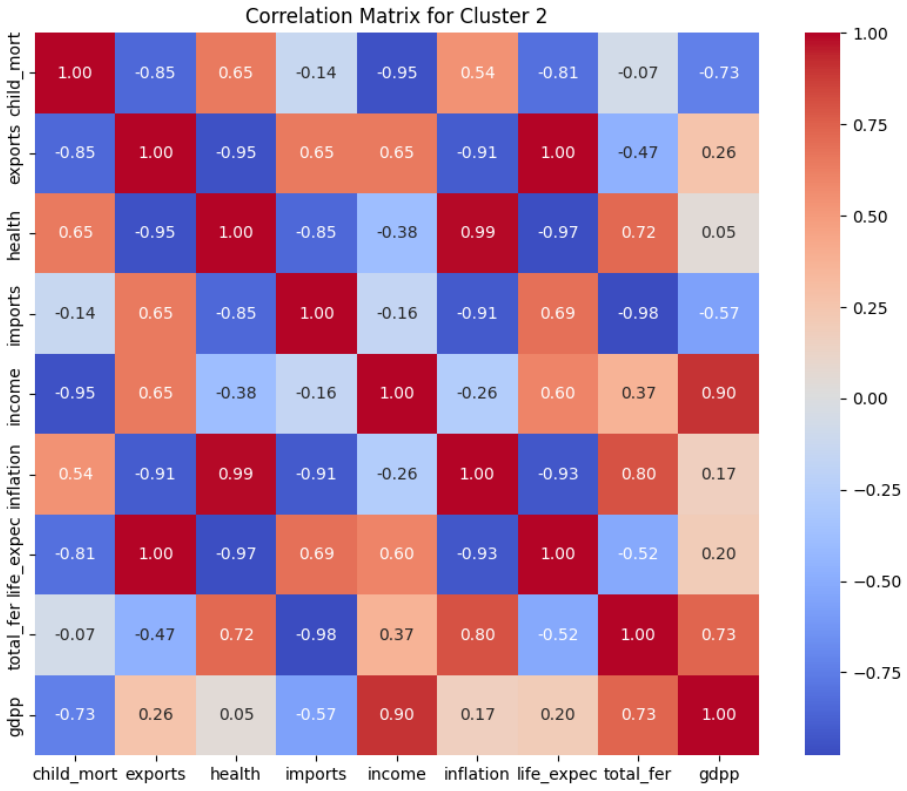
iv. Analysis part: We can see significant differences between the two clusters.

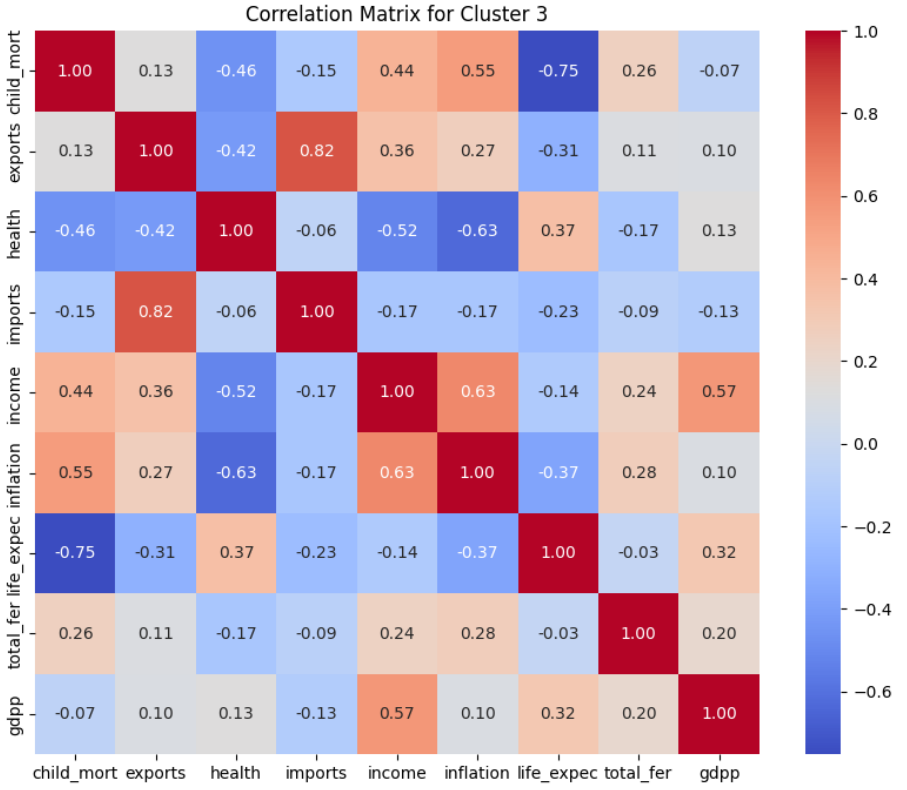


V.


Correlation Matrix for Cluster 1
child_mort 1.00 -0.11 -0.04 0.03 -0.19 -0.12 -0.63 0.35 -0.19
exports -0.11 1.00 -0.32 0.30 0.62 0.38 0.06 -0.11 0.60
health -0.04 -0.32 1.00 0.35 -0.14 -0.13 -0.18 -0.09 -0.10
imports 0.03 0.30 0.35 1.00 -0.00 -0.19 -0.23 -0.23 0.01
income -0.19 0.62 -0.14 -0.00 1.00 0.37 0.12 -0.29 0.98
inflation -0.12 0.38 -0.13 -0.19 0.37 1.00 0.31 0.16 0.38
life_expec -0.63 0.06 -0.18 -0.23 0.12 0.31 1.00 0.12 0.11
total_fer 0.35 -0.11 -0.09 -0.23 -0.29 0.16 0.12 1.00 -0.26
gdpp -0.19 0.60 -0.10 0.01 0.98 0.38 0.11 -0.26 1.00
child_mort exports health imports income inflation life_expec total_fer gdpp
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6


Correlation Matrix for Cluster 0
child_mort 1.00 -0.29 -0.34 -0.22 -0.55 0.11 -0.63 0.55 -0.61
exports -0.29 1.00 -0.09 0.64 0.42 -0.19 0.11 -0.16 0.37
health -0.34 -0.09 1.00 0.23 -0.08 -0.20 0.25 -0.27 0.07
imports -0.22 0.64 0.23 1.00 -0.04 -0.38 -0.06 -0.03 -0.02
income -0.55 0.42 -0.08 -0.04 1.00 0.07 0.47 -0.29 0.92
inflation 0.11 -0.19 -0.20 -0.38 0.07 1.00 -0.15 0.13 -0.02
life_expec -0.63 0.11 0.25 -0.06 0.47 -0.15 1.00 -0.41 0.53
total_fer 0.55 -0.16 -0.27 -0.03 -0.29 0.13 -0.41 1.00 -0.34
gdpp -0.61 0.37 0.07 -0.02 0.92 -0.02 0.53 -0.34 1.00
child_mort exports health imports income inflation life_expec total_fer gdpp
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6


Correlation Matrix for Cluster 2
child_mort 1.00 -0.85 0.65 -0.14 -0.95 0.54 -0.81 -0.07 -0.73
exports -0.85 1.00 -0.95 0.65 0.65 -0.91 1.00 -0.47 0.26
health 0.65 -0.95 1.00 -0.85 -0.38 0.99 -0.97 0.72 0.05
imports -0.14 0.65 -0.85 1.00 -0.16 -0.91 0.69 -0.98 -0.57
income -0.95 0.65 -0.38 -0.16 1.00 -0.26 0.60 0.37 0.90
inflation 0.54 -0.91 0.99 -0.91 -0.26 1.00 -0.93 0.80 0.17
life_expec -0.81 1.00 -0.97 0.69 0.60 -0.93 1.00 -0.52 0.20
total_fer -0.07 -0.47 0.72 -0.98 0.37 0.80 -0.52 1.00 0.73
gdpp -0.73 0.26 0.05 -0.57 0.90 0.17 0.20 0.73 1.00
child_mort exports health imports income inflation life_expec total_fer gdpp
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75


Correlation Matrix for Cluster 3
child_mort 1.00 0.13 -0.46 -0.15 0.44 0.55 -0.75 0.26 -0.07
exports 0.13 1.00 -0.42 0.82 0.36 0.27 -0.31 0.11 0.10
health -0.46 -0.42 1.00 -0.06 -0.52 -0.63 0.37 -0.17 0.13
imports -0.15 0.82 -0.06 1.00 -0.17 -0.17 -0.23 -0.09 -0.13
income 0.44 0.36 -0.52 -0.17 1.00 0.63 -0.14 0.24 0.57
inflation 0.55 0.27 -0.63 -0.17 0.63 1.00 -0.37 0.28 0.10
life_expec -0.75 -0.31 0.37 -0.23 -0.14 -0.37 1.00 -0.03 0.32
total_fer 0.26 0.11 -0.17 -0.09 0.24 0.28 -0.03 1.00 0.20
gdpp -0.07 0.10 0.13 -0.13 0.57 0.10 0.32 0.20 1.00
child_mort exports health imports income inflation life_expec total_fer gdpp
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6

vi.

Global Means:

	Feature	Mean
0	child_mort	-3.7229e-17
1	exports	2.12737e-16
2	health	5.50458e-16
3	imports	2.76559e-16
4	income	-7.97765e-17
5	inflation	-1.06369e-17
6	life_expec	3.69631e-16
7	total_fer	3.0448e-16
8	gdpp	5.85028e-17

Cluster Means:

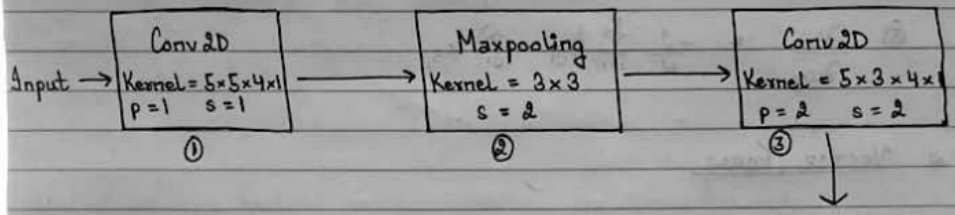
								Cluster	Feature	Mean
0	0	-0.41424	-0.00486343	-0.236121	0.0260358	-0.225992	-0.015464	0.276216	-0.444141	-0.349263
1	1	1.3063	-0.418849	-0.128947	-0.132965	-0.690625	0.205899	-1.2793	1.34239	-0.605453
2	2	-0.849003	4.93567	-0.00816303	4.54806	2.43954	-0.504206	1.22682	-1.03886	2.4408
3	3	-0.822941	0.183308	0.829894	-0.261252	1.39838	-0.499856	1.07433	-0.76825	1.59544
4	4	2.28139	-0.578452	-0.637438	-1.22178	-0.624065	9.12972	-1.13412	1.91613	-0.581936

Theoretical Part (Section C)

a. A

i.

1. a) Dimensions of input image = 15 × 15 × 4
↳ no. of channels



$$\text{Output of a kernel} = \left(\frac{\text{Input} + 2.\text{Padding} - \text{Height}}{\text{Stride}}\right) + 1$$

$$\therefore \text{Output height} = \left(\frac{\text{Input}_H + 2.\text{Padding} - \text{Kernel}_H}{\text{Stride}}\right) + 1$$

$$\text{Output width} = \left(\frac{\text{Input}_W + 2.\text{Padding} - \text{Kernel}_W}{\text{Stride}}\right) + 1$$

Output channel = Kernel output channel

① Output :

$$\text{Height} = \left(\frac{15 + 2.1 - 5}{1}\right) + 1 = 13$$

$$\text{Width} = \left(\frac{15 + 2.1 - 5}{1}\right) + 1 = 13$$

Channel = 1

② Output :

$$\text{Height} = \left(\frac{13 + 2.0 - 3}{2}\right) + 1 = 6$$

$$\text{Width} = \left(\frac{13 + 2.0 - 3}{2}\right) + 1 = 6$$

Channel = 1

③ Output :

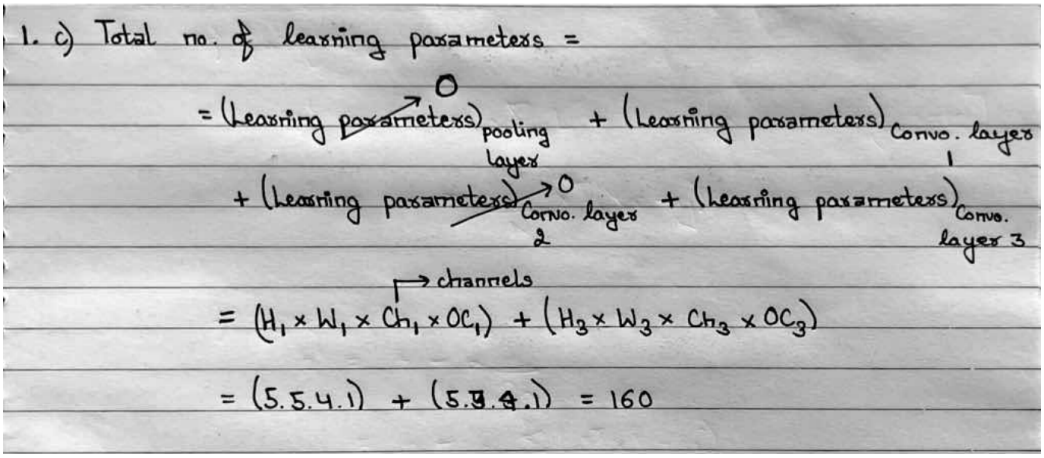
$$\text{Height} = \left(\frac{6 + 2.2 - 5}{2}\right) + 1 = 3$$

$$\text{Width} = \left(\frac{6 + 2.2 - 3}{2}\right) + 1 = 4$$

Channel = 1

ii. The purpose of the pooling layers is to reduce the dimensions of the hidden layer by combining the outputs of neuron clusters at the previous layer into a single neuron in the next layer.
    1. Translation Invariant: Pooling helps in achieving translation invariance by selecting the most appropriate information
    2. Parameter Reduction: Reduces the number of parameters, which helps prevent overfitting
    3. Downsampling: Reduces the dimensions, which helps in lowering the computational complexity

iii.



b. When the number of groups (k) is constant, the k-means algorithm may produce the same groupings based on the initial choice of group centers. If the initial centers are chosen from the same group where the algorithm previously converged, it will revisit that grouping. This occurs because the closest points to those initial centers stay consistent, regardless of how often the algorithm is run.
The algorithm minimizes a cost function, precisely the sum of squared distances between data points and their assigned group centers. The number of possible configurations (assignment points to groups and corresponding centers) is finite. The cost function decreases or remains the same with each algorithm iteration.
The algorithm cannot endlessly cycle through configurations due to the monotonic decrease in the cost function and the finite number of possible configurations. The k-means algorithm is guaranteed to converge in a finite number of steps, ensuring it eventually reaches the same grouping.

c. kNN, or k-Nearest Neighbors, is an algorithm that predicts outcomes by looking at the majority class of the closest K data points. It calculates distances between points without learning specific parameters or weights, which is a key difference from neural networks. Unlike neural networks, kNN doesn't create a model from the training data. Instead, it uses the entire dataset for predictions. It identifies the K-nearest neighbors and makes decisions based on their labels.
Neural networks, in contrast, are designed for learning intricate patterns by adjusting weights during training. They have input, hidden, and output layers, and their weights

are optimized using specific methods. This is a stark contrast to kNN, which doesn't involve weights or parameters.
Therefore, a neural network isn't suitable for modeling the kNN algorithm due to their differing approaches – kNN relies on distances and majority classes, while neural networks learn patterns through weight adjustments.

d.

LINEAR KERNEL	**NON-LINEAR KERNEL**
Used when there's a linear relationship between the input and output.	Used when there's no linear relationship between the input and output, consists of a more complex function
It may not capture the complexity; computational easy	Capture more complex patterns; might be computationally expensive
The decision boundary is a hyperplane	The decision boundary can have a complex, non-linear shape
It consists of an identity activation function	It consists of a non-linear activation function
Sensitive to outliers	More robust to outliers
e.g: y = 2x + 5	e.g.: RBF(Radial Based Kernel) kernel