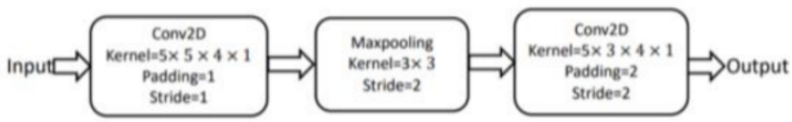
CSE343 : Machine Learning
Assignment-4
CNN, PCA, K-means clustering

2021337 Megha

Q1.



A.a

Kernel =(5x5x4x1) Padding =1, stride =1
Input = 15x15x4
Output1 :
W = (15-5+2*1)/1 +1 = 13
H = (15-5+2*1)/1 +1 = 13
Input2 = (13x13x1)

Maxpooling
Kernel =(3x3) stride =2
Input2 = 13x13x1
Output2 :
W = (13-3)/2 +1 = 6
H = (13-3)/2 +1 =6
Input3 = 6x6x1

Kernel =(5x3x4x1) Padding =2, stride=2
Output : (6-5+2*2)/2 +1 = 3

Final = (3x3x1)

A.b Pooling is used to reduce the dimension of feature maps without losing much information, as it decreases the number of parameters to learn.It shows the features in a concise way.

A.c Total learnable parameters
Layer1:
Kernel size : 5x5x4x1 = 100 = No. of parameters
Layer2:
Max Pooling layers do not have learnable parameters.

Layer3:
Kernel size : 5x3x4x1 = 60 =No. Of parameters

Total =100+60 = 160

B.Yes, it is possible to revisit a configuration during iterations in K-means.There are two steps : Assigning data points to nearest centroid and then updating centroids by calculating the assigned points .
So, it will reach a point when no centroids are updated on further iterations. This is the point when the algorithm has converged.

It will always converge because :
1.It minimizes the sum of squared distances between data points and their assigned centroids.So, it is non increasing with each iteration, and it reaches a local minimum when no change in centroid occurs.And this always happen in finite steps.

C.KNN(K-nearest neighbors) used for classification and regression analysis,whereas Neural networks are used for complex functions which deal with recognizing patterns.
Neural networks are said to be a universal function approximator, so yes we can express KNN prediction function as a neural network.
Layer1 : Distance calculation
y=wx+b
x=input

Layer2 : Hidden Layer
Layer 3 : Output

D.

Feature	**Linear kernels**	**Non-Linear kernels**
Operation	Linear during convolution operation	Non-Linear activation function
Computational rate	Simpler, computationally efficient	Complex
	Can’t capture complex patterns	Capture complex patterns
Applications	Smoothing,Blurring, edge detection	Feature detection,pattern recognition,image classification
Ex:	Mean filter,Gaussian filter,Sobel filter	Sigmoid filter,ReLU filter,Max pooling filter

Q3. Section C
Clustering Analysis using PCA and K-Means

a. Data

```
data = pd.read_csv('Country-data.csv')
data.head()
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

EDA

Except ‘country’ all others are numeric columns

```
data.info()
#No null values. 'Country' is object
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     167 non-null    object 
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64  
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64  
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

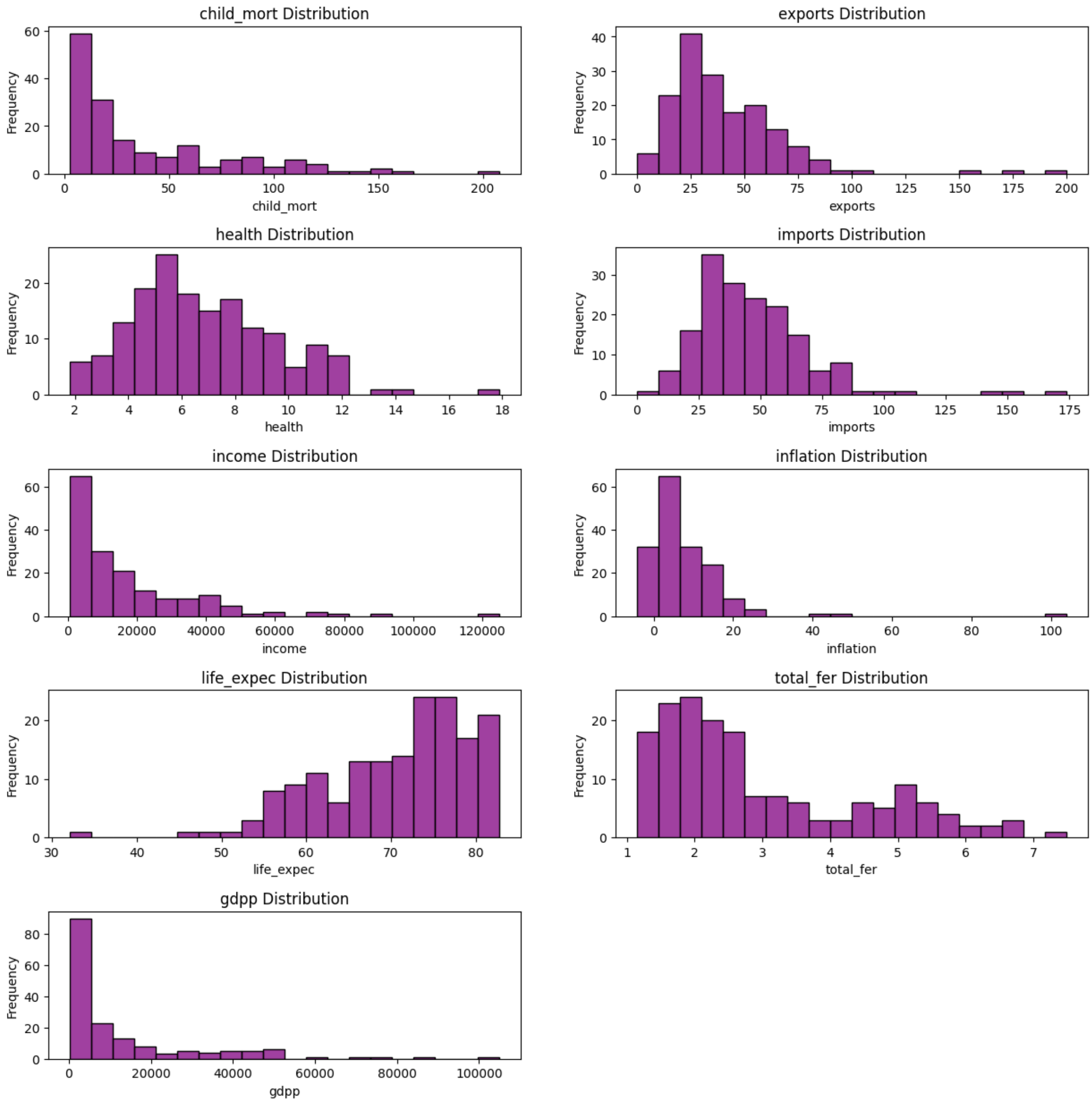
Data description

```
data.describe()
```

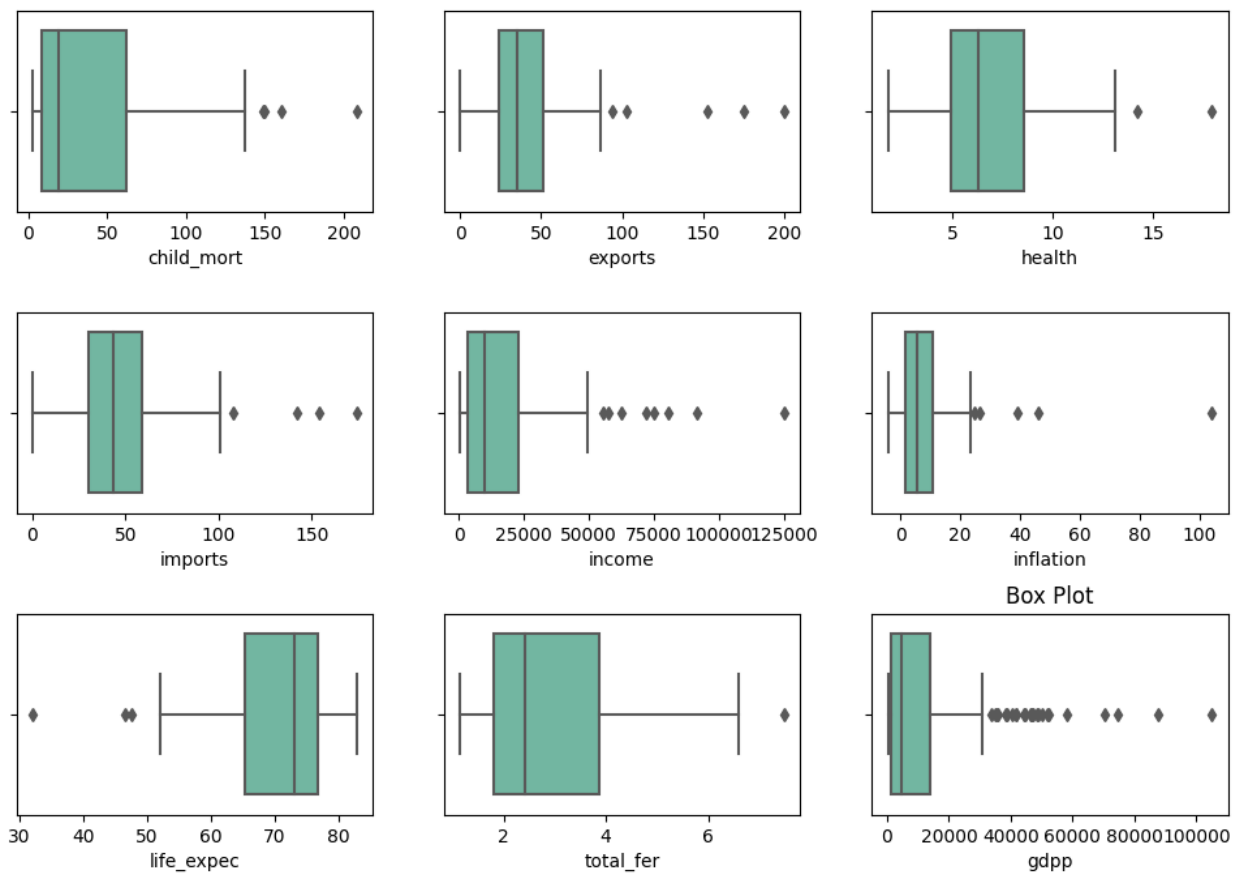
	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689	2.947964	12964.155689
std	40.328931	27.412010	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

Data Visualization :

1.Histogram

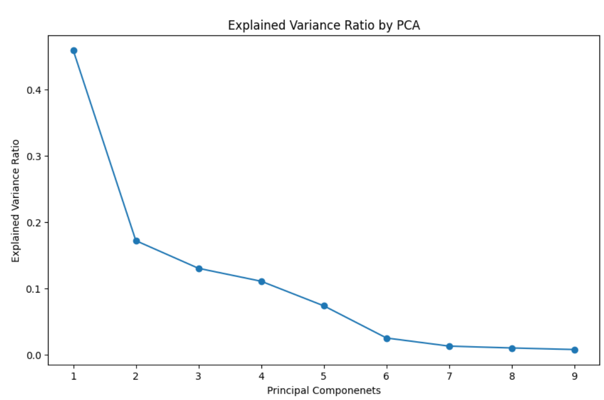


2.Box Plot

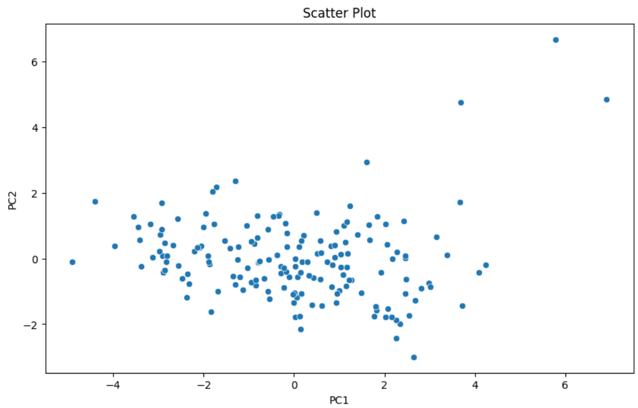


Clear outliers in child_mort ,exports ,imports, income, gdpp.But as our dataset is not small, we can’t remove them.

b. PCA

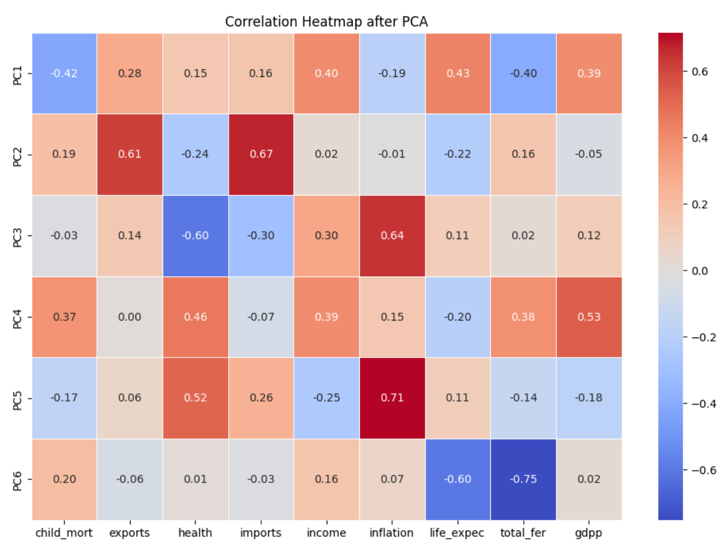


This signifies we need only 6 features



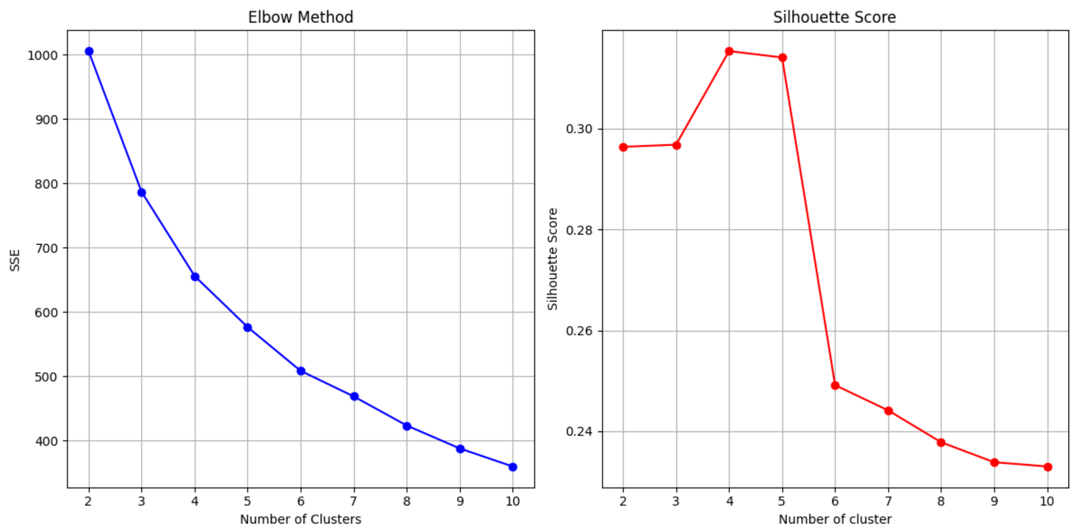
	Feature	PC1	PC2	PC3	PC4	PC5	PC6
0	child_mort	-0.419519	0.192884	-0.029544	0.370653	-0.168970	0.200628
1	exports	0.283897	0.613163	0.144761	0.003091	0.057616	-0.059333
2	health	0.150838	-0.243087	-0.596632	0.461897	0.518000	0.007276
3	imports	0.161482	0.671821	-0.299927	-0.071907	0.255376	-0.030032
4	income	0.398441	0.022536	0.301548	0.392159	-0.247150	0.160347
5	inflation	-0.193173	-0.008404	0.642520	0.150442	0.714869	0.066285
6	life_expec	0.425839	-0.222707	0.113919	-0.203797	0.108220	-0.601127
7	total_fer	-0.403729	0.155233	0.019549	0.378304	-0.135262	-0.750689
8	gdpp	0.392645	-0.046022	0.122977	0.531995	-0.180167	0.016779

Correlation Heatmap after PCA



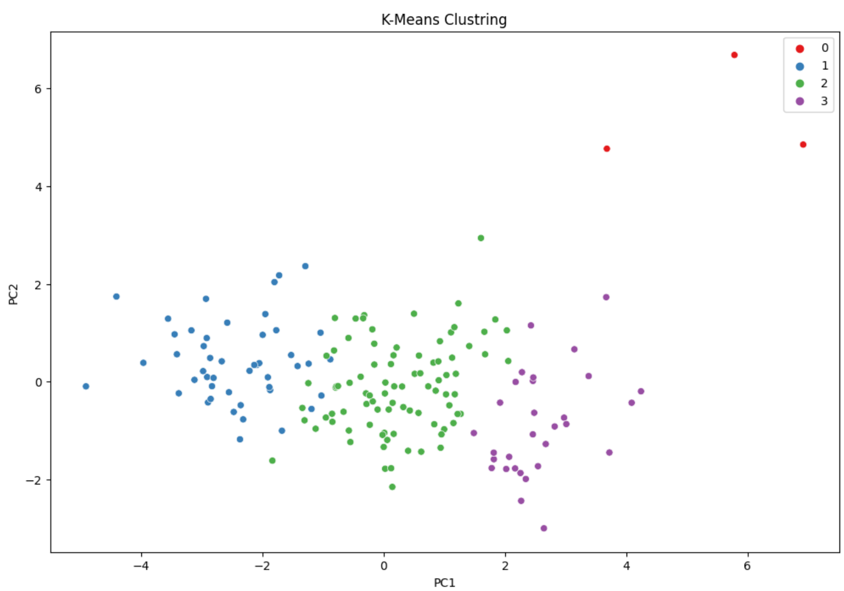
There's strong relationship between PC4 and gdpp ,inflation and PC5,PC3, Imports and PC2 and so on.

c. K-Means clustering algorithm

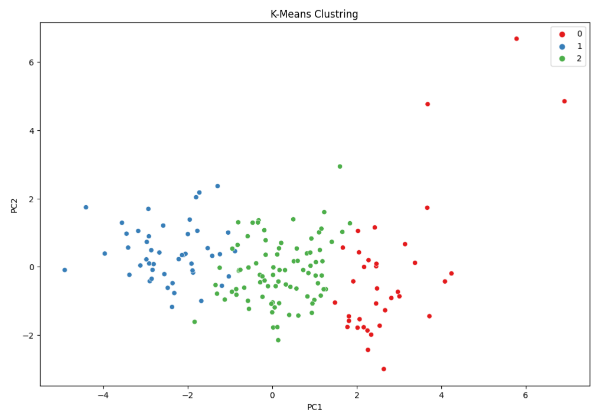


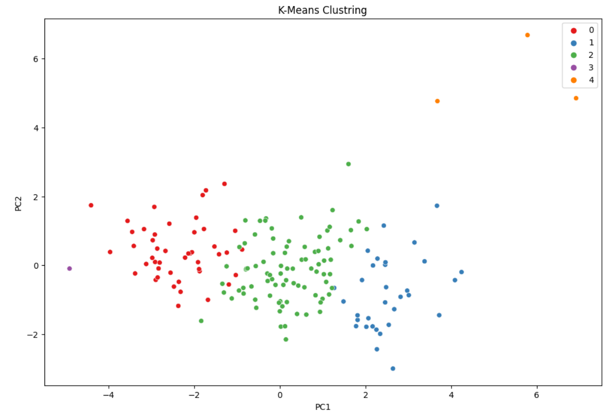
We can see that ‘k’==4 is giving by both the graphs.

Applying K Means using k==4



After analyzing the graph, I found that the clusters are not separated too much,and some are lying in other clusters as well.
So, I tried with clusters=3 and 5 also.
But 4 is giving the best result.





K == 3(Only some data points get Added to cluster 3)

k==5 (Not any better performance terms of segregation)

After applying K means clustering, the assigned clusters are

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	Cluster
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553	1
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090	2
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460	2
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530	1
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200	2
...	...	...	...	...	...	...	...	...	...	...	...
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970	2
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500	2
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310	2
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310	1
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460	1

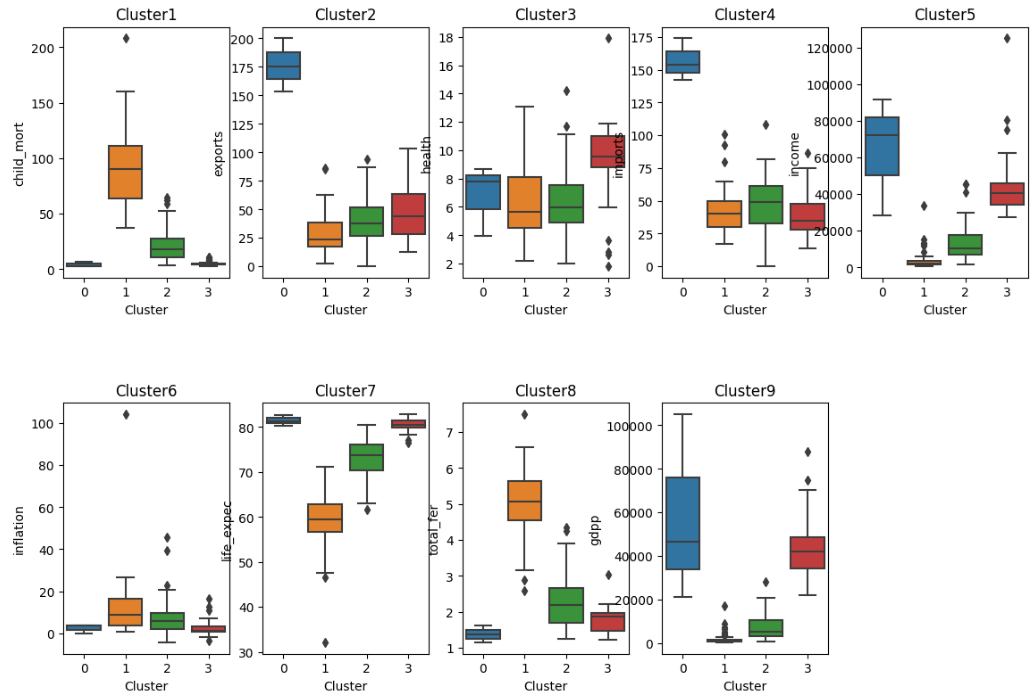
167 rows × 11 columns

The cluster wise inputs are.

```
2    87
1    47
3    30
0     3
Name: Cluster, dtype: int64
```

Analyzing clusters

1.BoxPlot



2.Mean ,Median ,std of each feature cluster wise

	Cluster	0	1	2	3
child_mort	mean	4.133333	92.961702	21.389655	4.953333
	median	2.800000	90.200000	18.100000	4.200000
	std	2.309401	33.375229	13.821462	2.159140
exports	mean	176.000000	29.151277	41.290678	45.826667
	median	175.000000	23.800000	37.700000	44.250000
	std	23.515952	18.160597	19.523129	21.736255
health	mean	6.793333	6.388511	6.235862	9.168667
	median	7.770000	5.660000	5.980000	9.535000
	std	2.492877	2.662015	2.158742	3.266299
imports	mean	156.666667	42.323404	48.038689	39.736667
	median	154.000000	40.300000	49.200000	35.000000
	std	16.165808	17.732741	20.083366	17.455134
income	mean	64033.333333	3942.404255	12968.620690	45250.000000
	median	72100.000000	1870.000000	10500.000000	40550.000000
	std	32460.642836	5641.790360	8870.376775	19785.308900
inflation	mean	2.468000	12.019681	7.413460	2.742200
	median	3.620000	8.920000	5.730000	1.190000
	std	2.179718	15.509958	7.808165	4.266366
life_expec	mean	81.433333	59.187234	72.935632	80.376667
	median	81.300000	59.500000	73.800000	80.400000
	std	1.205543	6.443521	3.947474	1.440231
total_fer	mean	1.380000	5.008085	2.286552	1.795333
	median	1.360000	5.060000	2.200000	1.865000
	std	0.240624	1.041382	0.696392	0.369293
gdpp	mean	57566.666667	1922.382979	6919.103448	43333.333333
	median	46600.000000	897.000000	5020.000000	41850.000000
	std	43011.665084	2956.103925	5453.932294	15040.114942

Cluster 0:

```
        country  child_mort  exports  health  imports  income  inflation  \
91   Luxembourg         2.8    175.0    7.77    142.0   91700      3.620   
98        Malta         6.8    153.0    8.65    154.0   28300      3.830   
133   Singapore         2.8    200.0    3.96    174.0   72100     -0.046   

     life_expec  total_fer    gdpp  Cluster  
91         81.3       1.63  105000        0  
98         80.3       1.36   21100        0  
133        82.7       1.15   46600        0  
```

Cluster 1:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	Cluster
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.440	56.2	5.82	553	1
3	Angola	119.0	62.3	2.85	42.9	5900	22.400	60.1	6.16	3530	1
17	Benin	111.0	23.8	4.10	37.2	1820	0.885	61.8	5.36	758	1
21	Botswana	52.5	43.6	8.30	51.3	13300	8.920	57.1	2.88	6350	1
25	Burkina Faso	116.0	19.2	6.74	29.6	1430	6.810	57.9	5.87	575	1

Cluster 2:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	Cluster
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090	2
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460	2
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200	2
5	Argentina	14.5	18.9	8.10	16.0	18700	20.90	75.8	2.37	10300	2
6	Armenia	18.1	20.8	4.40	45.3	6700	7.77	73.3	1.69	3220	2

Cluster 3:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	Cluster
7	Australia	4.8	19.8	8.73	20.9	41400	1.160	82.0	1.93	51900	3
8	Austria	4.3	51.3	11.00	47.8	43200	0.873	80.5	1.44	46900	3
15	Belgium	4.5	76.4	10.70	74.7	41100	1.880	80.0	1.86	44400	3
23	Brunei	10.5	67.4	2.84	28.0	80600	16.700	77.1	1.84	35300	3
29	Canada	5.6	29.1	11.30	31.0	40700	2.870	81.3	1.63	47400	3