# MINOR PROJECT REPORT

Riya (B21EE085)

Khushi Maheshwari (B21AI052)

Jiya Kumawat (B21CS036)

**Importing dataset:**

We chose project 5(Country_data). First we mount the google drive and import the given dataset 'Country_Data'. We also import the given libraries.

**Initial data visualization:**

We look at all the information and description about all the features to have a better idea about the dataset.

**Data Preprocessing:**

We find out the missing values in our dataset and we get total missing values as 0.

We visualize the dataset using pairplots from the seaborn library and we also plot distributions of each feature using distplot() function.

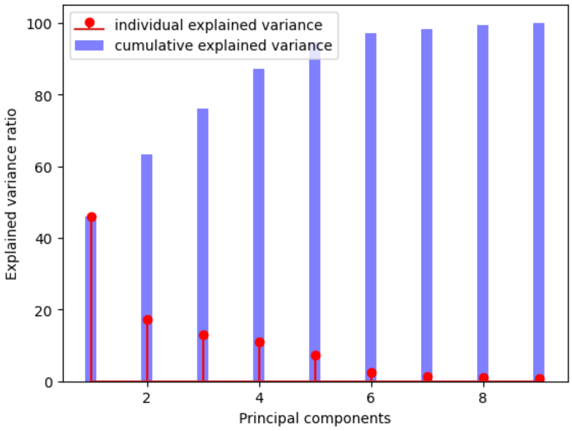
We also plot 6 values each for high, low and median values for each feature using the classifier function that we defined to get an idea of the ranges of the data and its max, min and median values. We also find the correlation matrix and plot it using heatmap.

We see that except country, all the features are numerical features. We make lists for continuous and categorical features and perform standardization on all the continuous features using StandardScaler.

**Dimensionality Reduction:**

We perform PCA for feature selection because it is preferrably used for an Unsupervised Learning Problem. We first create a function to centralize the data by subtracting mean and dividing by standard deviation. To find the covariance matrix, we make a function covar() that takes data as input and apply the formula (xTx)/(n-1) to return the covariance matrix. We centralize and find covariance for the given data. Next, we find the eigen values and eigen vectors using np.linalg.eig() function using the covariance matrix. We make eigen pairs.

We find the variance and cumulative variance expressed as we transverse each component. We also plot a bar graph to see the change in variance as we increase the number of components. Here the red stem indicate the individual explained variance by each component i.e. change in variance as we increase the number of components. We see that it decreases as number of components increase. For our dataset, we choose the number



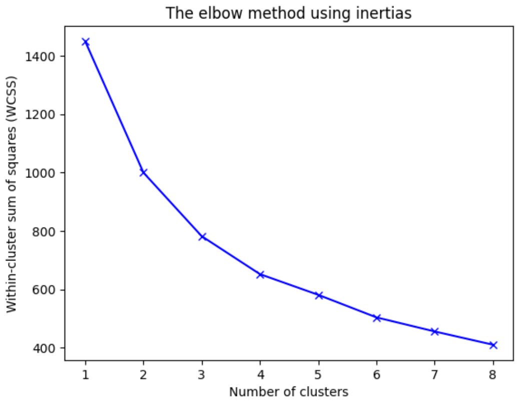
of components as 6 as >97% variance is explained up to 6 components so we transform the data using the top 6 eigen vectors using dot product. We finally get our PCA transformed dataset.

## Different clustering algorithms:

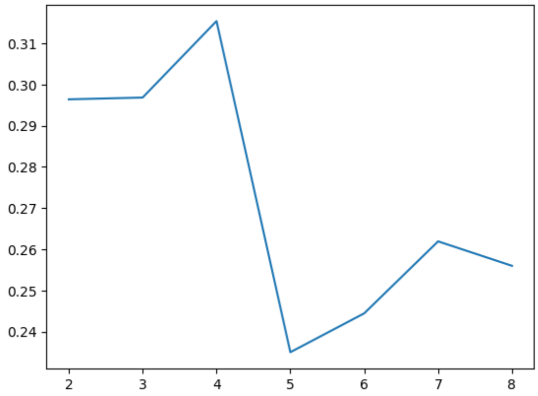
We try out various clustering algorithms to cluster the countries using socio-economic and health factors that determine the overall development of the country.

- **KMeans clustering:** we perform K means because it is the most preferred algorithm for dealing with Unsupervised Learning that finds clusters in the data. In this, we have to give the number od cluster, k as input. To find the optimal value of k, we use 2 tests,

  1. Elbow method: In the elbow method, we find the within cluster sum of squares for each value of k. we see that the slope till 4 is decreasing more sharply and then it decreases at a lower rate. If the plot looks like an arm, the the elbow is the optimum value therefore we find the optimum value of k as 4.

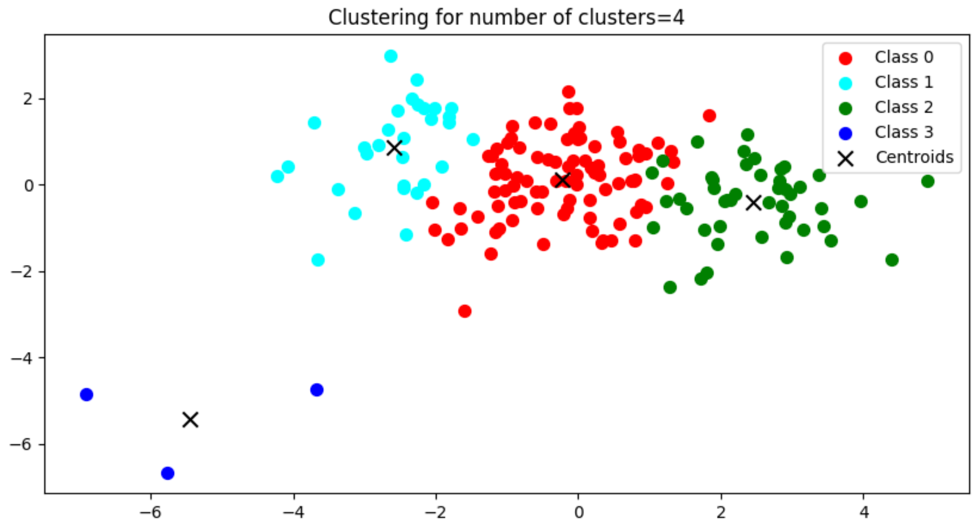


  2. Silhouette score: we vary k from 2 to 9 and note the value of k for which the silhouette score is maximum. The optimal value of k is 4 because it has the highest silhoutte score and silhoutte score tells how good is the clustering. A silhoutte score of 1 means perfect distinction of clusters.

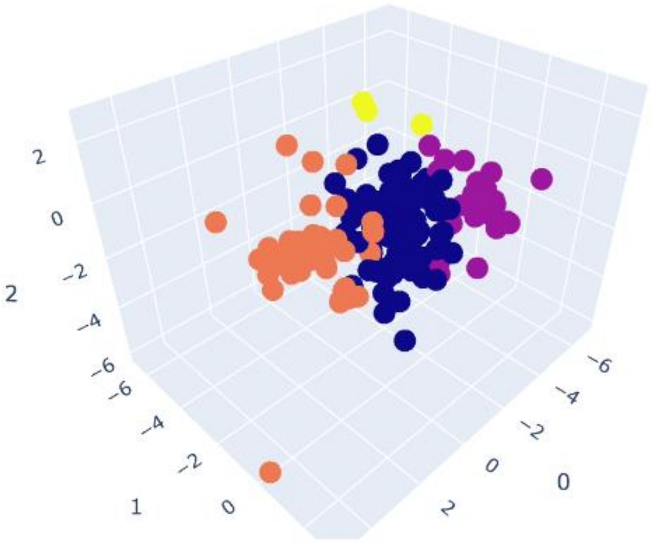


Using these two methods, we take the value of k as 4 and make an object for k means using k=4.

We fit our dataset in the kmeans object and find out the predicted class labels by using kmeans.lables_ and centroid by using kmeans.cluster_centers_. We then plot the clusters using scatter plot between the top 2 features using different colours for each cluster.



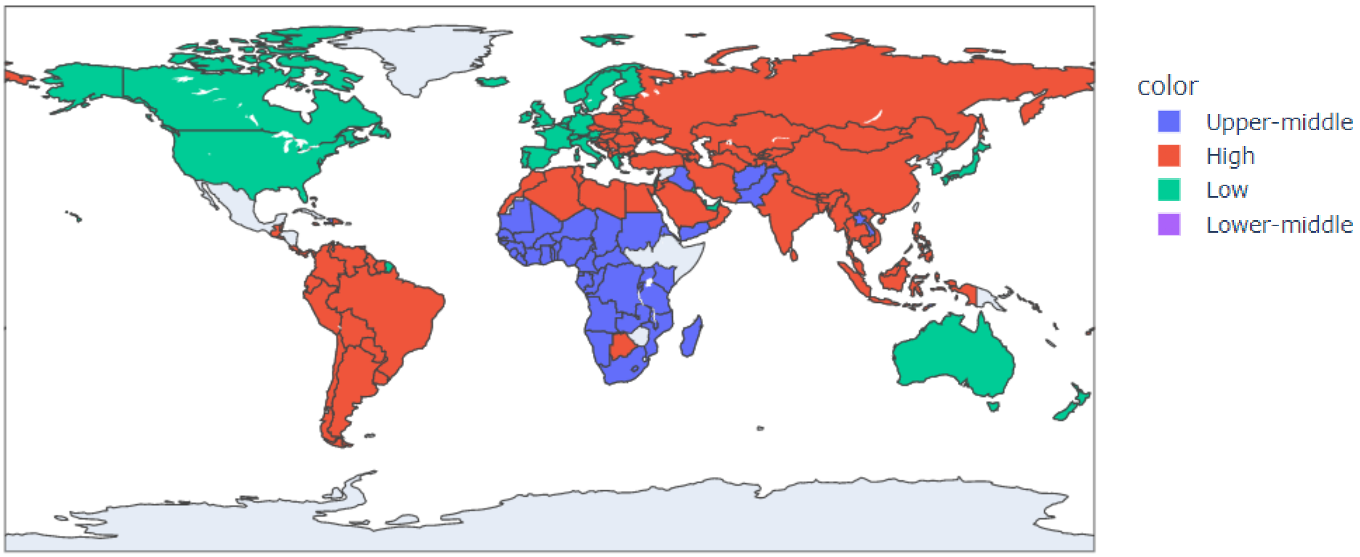
We also plot a 3-D scatter plot between the top 3 most important features using scatter_3d() function from the plotly.express library.



Now we have the 4 classes but we don't know which cluster corresponds to what level of overall development of the country i.e. which cluster represents highly developed, underdeveloped, developing etc countries. So to know this, we plot bar plots for each feature for each cluster.

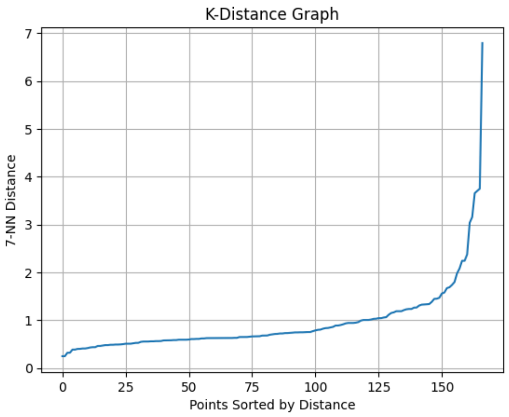
From these plots, we get to know that cluster with class label=0 indicates highly developed contires, label=1 indicates least or underdeveloped countries, label=2 indicated countries lying in the upper middle level. of development and label=3 indicated countries lying in the lower middle level of development.

Then we create a choropleth map using the px.choropleth method of the Plotly Express library. The map shows the overall development status of different countries based on their respective Class values, which were previously converted into categorical strings. The locationmode parameter is set to country names to enable the mapping of the country names in the Country column of the DataFrame.



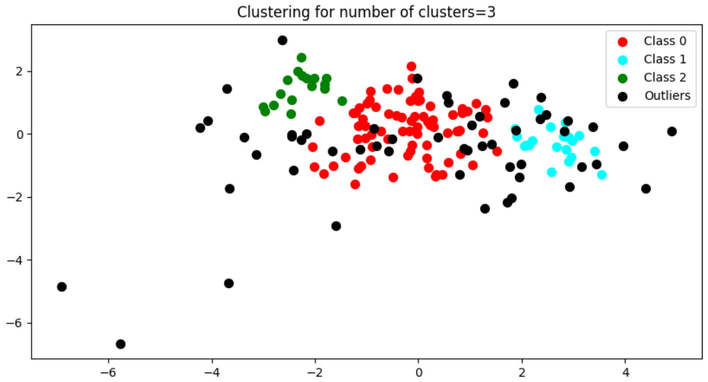
- **DBSCAN clustering:** next we apply dbscan which is a density based algorithm for clustering and it can also detect the outliers.

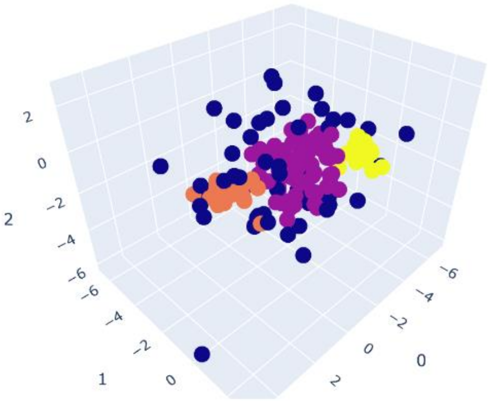
The two important hyoerparameters of DBSCAN are eps and min_samples. We take min_samples as >=2*dimensionality for smaller datasets so we take min_samples as 12 and to find the optimal value for eps, we make the K-distance graph using knn.



From the graph we see that at value 1.3, the graph changes sharply. So we take the value of eps as 1.3.

We create the object with the above metioned values of the hyperparameters and fit out data to find the predicted class labels. We know that in dbscan, the cluster with label=-1 is a cluster containing all the outliers. We then plot the 2-D and 3-D scattr plots of the clusters like we did in the previous algorithm.

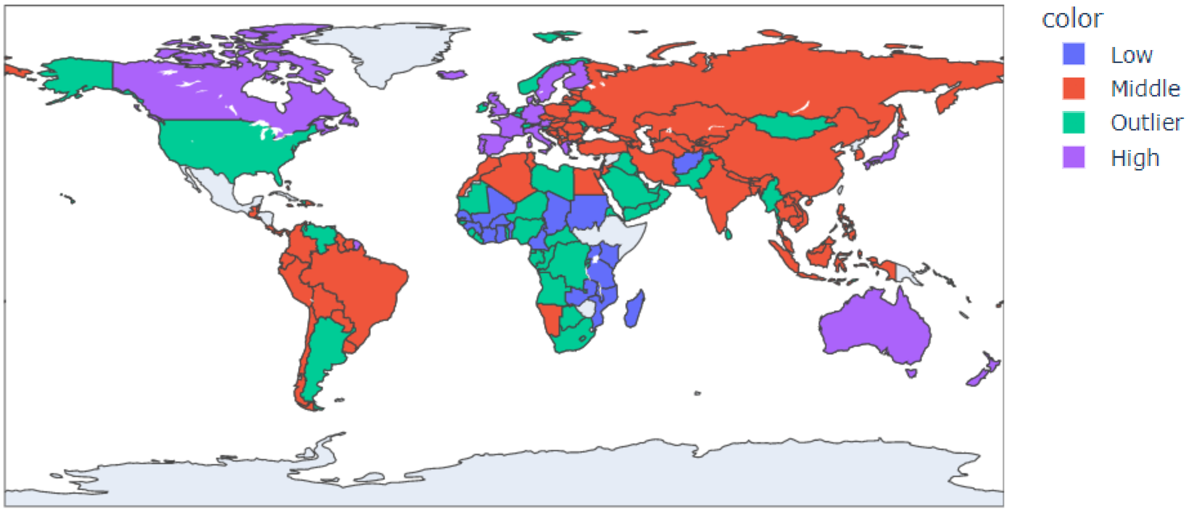




Again we find which clusters corresponds to what level of development of the countries by plotting bar graphs for all features corresponding to all the clsuters.

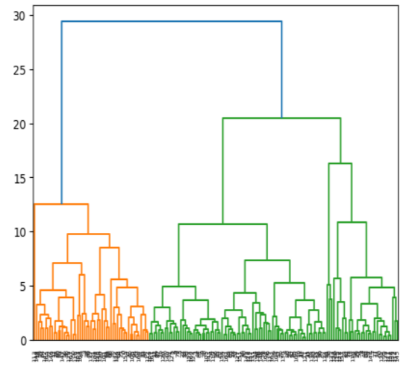
We find that, cluster with class label=0 indicates underdeveloped countries, label=1 indicates developing countries, label=2 indicates countries that are completly developed and and label=-1 indicates the outliers.

Then we create a choropleth map using the px.choropleth method of the Plotly Express library like we did previously.



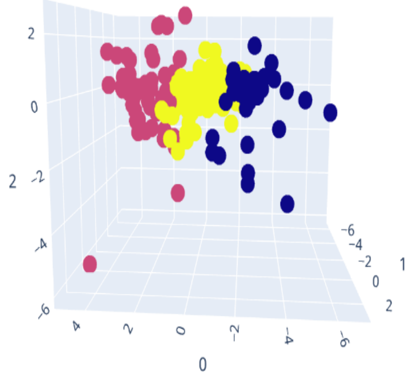
- **Hierarchical Clustering:** Next we apply Hierarchical clustering . We used it because produces clusters that are easy to interpret and understand . It has ability to handle different types of data so it can be used to cluster countries based on different types of data, including economic indicators, social indicators, and demographic data.

  We Plotted a dendogram of the data.



  Then we created an instance of Agglomerative Clustering ,set number of cluster as 3 and fit the data into it and find out the labels.

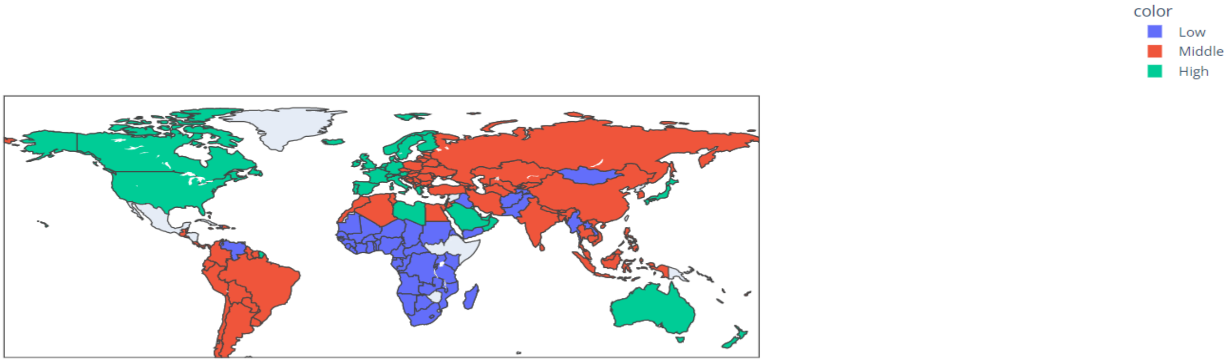
  We then plot the 2-D and 3-D scatter plots of the clusters like we did in the previous algorithm.



  Again we find which clusters corresponds to what level of development of the countries by plotting bar graphs for all features corresponding to all the clusters.

  We find that, cluster with class label=0 indicates countries that are completely developed, label=1 indicates underdeveloped countries,and label=2 indicates developing countries

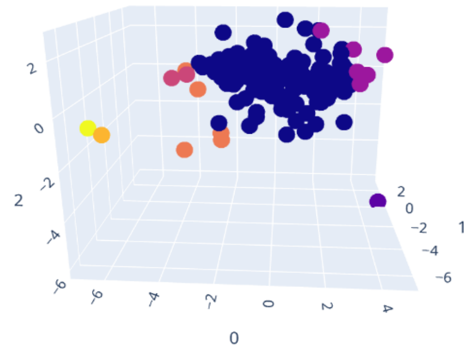
  Then we create a choropleth map using the px.choropleth method of the Plotly Express library like we did previously.

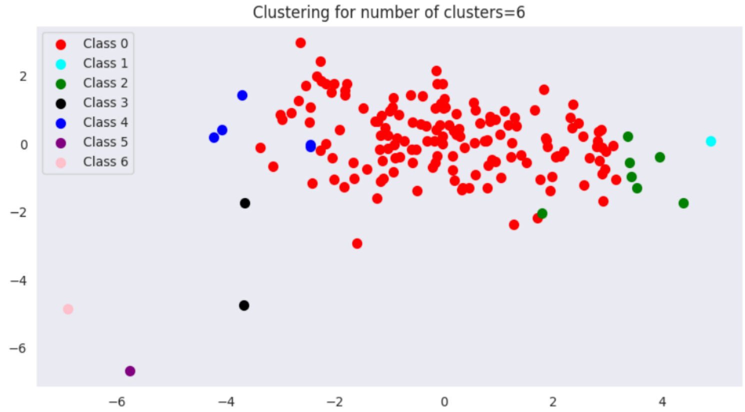


- **Mean Shift clustering**: Next we tried mean shift clustering . We used it because Mean shift clustering does not require the number of clusters to be specified a priori, which can be particularly useful when dealing with country data where the number of distinct groups may not be clear. Used estimate_bandwidth for calculating bandwidth then We created an instance of Mean shift with bandwidth equal to the calculated bandwidth . Fit the data in it . Find out the cluster centers and labels then calculated silhouette_score .

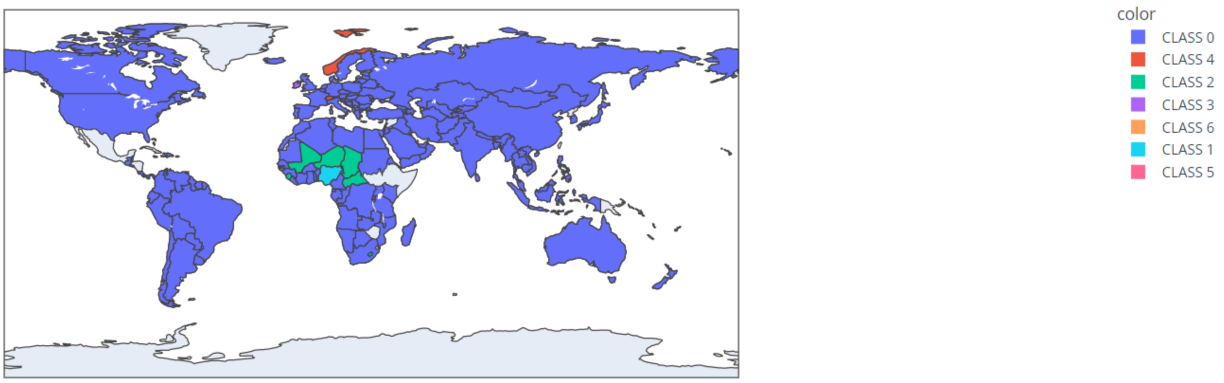
  We then plot the 2-D and 3-D scatter plots of the clusters like we did in the previous algorithm.

We got 6 clusters.



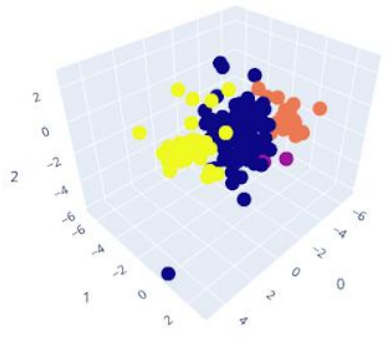


Then we create a choropleth map using the px.choropleth method of the Plotly Express library like we did previously.

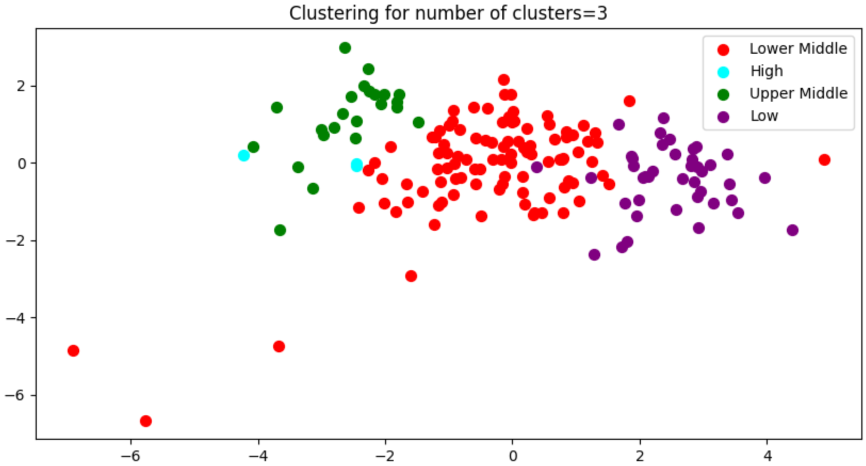


This clustering was not effective due to some reasons like we estimated the bandwidth using the estimate_bandwidth function with a quantile of 0.2. This approach may not always yield the optimal bandwidth value, and it may require some experimentation to find the best value or another reason can be that Mean Shift clustering is sensitive to outliers, which can affect the location of the cluster centroids and lead to suboptimal clustering results.

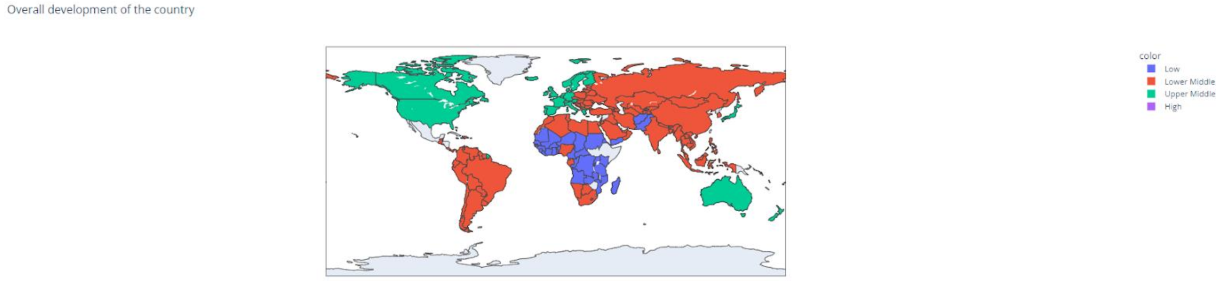
- **Spectral Clustering**: Spectral clustering can be used with different similarity metrics, allowing for the clustering of country data based on a wide range of features such as economic, social, and demographic indicators.Spectral clustering is robust to outliers.Spectral clustering produces clear and interpretable clusters that can be easily understood.s
  Next we apply spectral clustering algorithm on our dataset.
  We apply spectral clustering using sklearn library with n_clusters as 4 because we find using kmeans optimize value for the number of clusters k = 4.



We then plot the 2-D and 3-D scatter plots of the clusters like we did in the previous algorithm.


Clustering for number of clusters=3

Again we find which clusters corresponds to what level of development of the countries by plotting bar graphs for all features corresponding to all the clusters. From these plots, we get to know that cluster with class label=1 indicates highly developed contires, label=3 indicates least or underdeveloped countries, label=2 indicated countries lying in the upper middle level. of development and label=0 indicated countries lying in the lower middle level of development.
Then we create a choropleth map using the px.choropleth method of the Plotly Express library like we did previously.

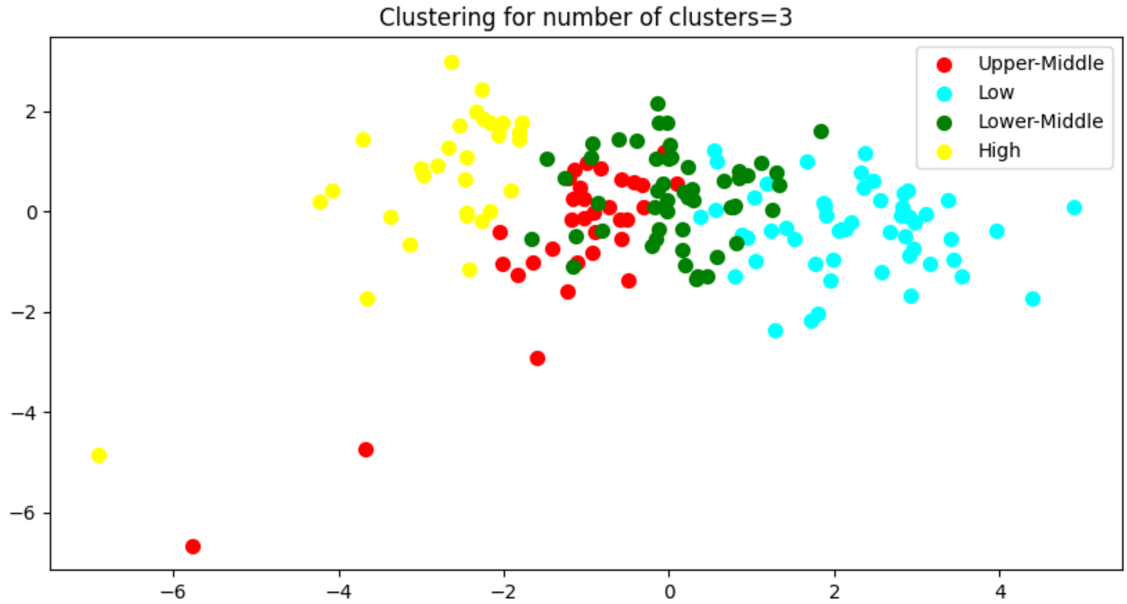

Overall development of the country

- **Gaussian Mixture:** GMM is a robust algorithm that can handle outliers. GMM provides a probability estimate for each data point to belong to each cluster. Next we apply gaussian mixture algorithm on our dataset.
  We apply gaussian mixture using sklearn library with n_components as 4 because we find using kmeans optimized value for the number of clusters k = 4.
  We then plot the 2-D and 3-D scatter plots of the clusters like we did in the previous algorithm.

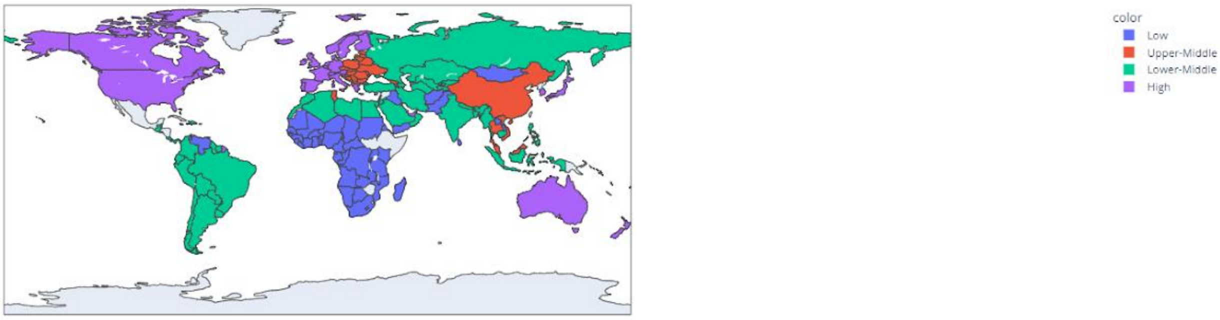
color
3
2.5
2
1.5
1
0.5
0

Again we find which clusters corresponds to what level of development of the countries by plotting bar graphs for all features corresponding to all the clusters.

From these plots, we get to know that cluster with class label=3 indicates highly developed contires, label=1 indicates least or underdeveloped countries, label=0 indicated countries lying in the upper middle level. of development and label=2 indicated countries lying in the lower middle level of development.



Then we create a choropleth map using the px.choropleth method of the Plotly Express library like we did previously.



**Conclusion:**

We get different results in all the clustering algorithms that we applied, namely kmeans, dbscan, hierarchical, mean shift, spectral and gaussian mixture because they are based on different techniques.

We see that mean shift clustering algorithm didn’t work for this dataset because it made 6 clusters while all others made 3-4 clusters and also it put the maximum countries in only one cluster and all the other clusters were scarce. This may be because it can be sensitive to the choice of bandwidth parameter and can struggle with datasets with varying densities.

All other algorithms fairly did a good job in clustering and had somewhat matching results but the exact efficiency of each method can’t be known because we don’t have the actual clusters to compare with.

| Clustering technique | Number of clusters |
|---|---|
| Kmeans | 4 |

| | |
|---|---|
| DBSCAN | 3+outliers |
| Hierarchical | 3 |
| Mean Shift | 6 |
| Spectral | 4 |
| Gaussian Mixture | 4 |

The exact number of clusters and the countries included in each cluster varied across different techniques. This shows that the choice of clustering technique can have an impact on the results.