HW 9

March 8, 2024

1 0.) Import and Clean data

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

[2]:
```
data = pd.read_csv("Country-data.csv", sep = ",")
data
```

[2]:
```
                 country  child_mort  exports  health  imports  income  \
0            Afghanistan        90.2     10.0    7.58     44.9    1610
1                Albania        16.6     28.0    6.55     48.6    9930
2                Algeria        27.3     38.4    4.17     31.4   12900
3                 Angola       119.0     62.3    2.85     42.9    5900
4    Antigua and Barbuda        10.3     45.5    6.03     58.9   19100
..                   …           …       …  …    …      …
162              Vanuatu        29.2     46.6    5.25     52.7    2950
163            Venezuela        17.1     28.5    4.91     17.6   16500
164              Vietnam        23.3     72.0    6.84     80.2    4490
165                Yemen        56.3     30.0    5.18     34.4    4480
166               Zambia        83.1     37.0    5.89     30.9    3280

     inflation  life_expec  total_fer   gdpp
0         9.44        56.2       5.82    553
1         4.49        76.3       1.65   4090
2        16.10        76.5       2.89   4460
3        22.40        60.1       6.16   3530
4         1.44        76.8       2.13  12200
..         …           …       …    …
162       2.62        63.0       3.50   2970
163      45.90        75.4       2.47  13500
164      12.10        73.1       1.95   1310
165      23.60        67.5       4.67   1310
166      14.00        52.0       5.40   1460
```

```
[167 rows x 10 columns]
```

[3]:
```python
names = data[["country"]].copy()
X = data.drop("country", axis =1)
```

[4]:
```python
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
```

2 1.) Fit a kmeans Model with any Number of Clusters

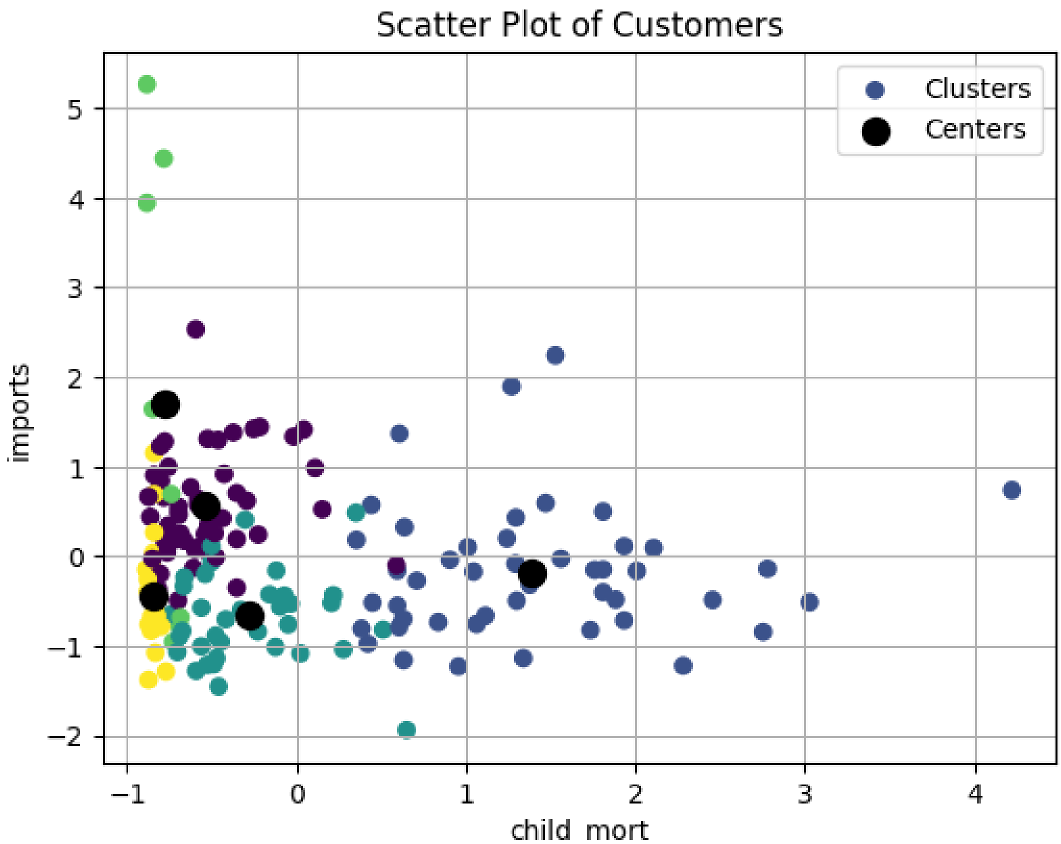
[5]:
```python
kmeans = KMeans(n_clusters = 5).fit(X_scaled)
```

3 2.) Pick two features to visualize across

[6]:
```python
X.columns
```

[6]:
```
Index(['child_mort', 'exports', 'health', 'imports', 'income', 'inflation',
       'life_expec', 'total_fer', 'gdpp'],
      dtype='object')
```

[7]:
```python
import matplotlib.pyplot as plt

x1_index = 0
x2_index = 3


scatter = plt.scatter(X_scaled[:, x1_index], X_scaled[:, x2_index], c=kmeans.
 ↪labels_, cmap='viridis', label='Clusters')


centers = plt.scatter(kmeans.cluster_centers_[:, x1_index], kmeans.
 ↪cluster_centers_[:, x2_index], marker='o', color='black', s=100,␣
 ↪label='Centers')

plt.xlabel(X.columns[x1_index])
plt.ylabel(X.columns[x2_index])
plt.title('Scatter Plot of Customers')

# Generate legend
plt.legend()

plt.grid()
plt.show()
```



4 3.) Check a range of k-clusters and visualize to find the elbow. Test 30 different random starting places for the centroid means

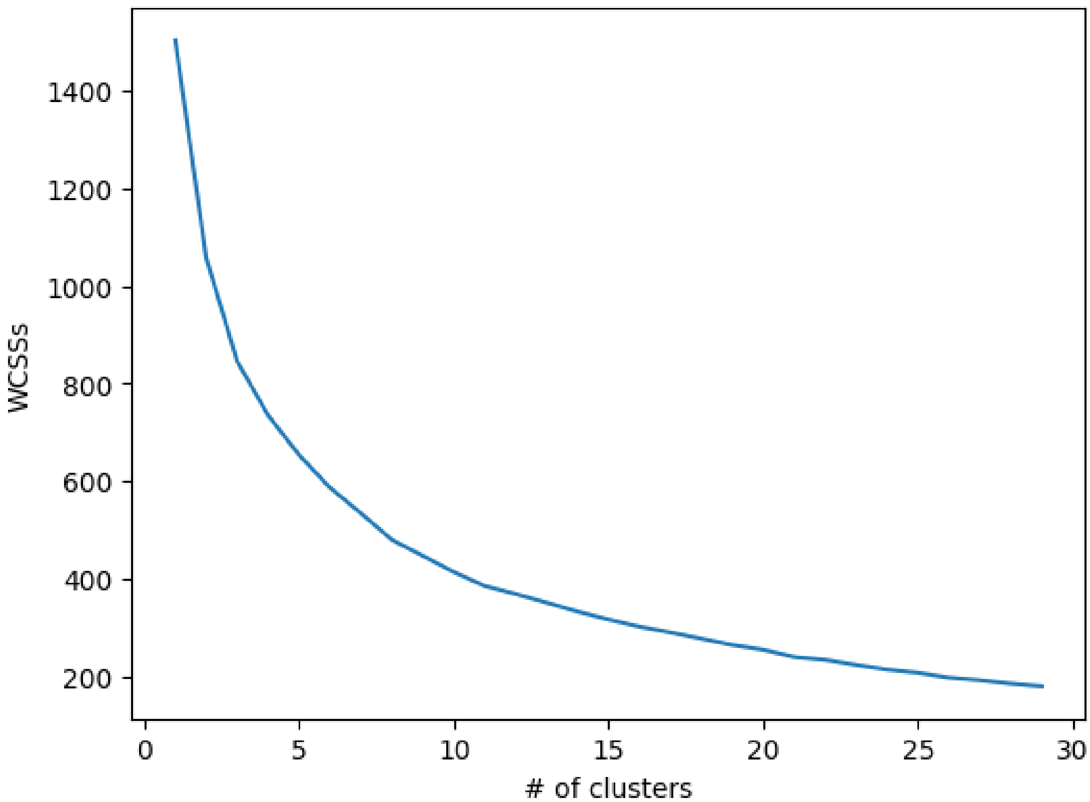
[8]:
```
WCSSs = []
Ks = range(1,30)
for k in Ks:
    kmeans = KMeans(n_clusters = k,n_init = 30).fit(X_scaled)
    WCSSs.append(kmeans.inertia_)

# WCSSs = [KMeans(n_clusters = 5,n_init = 30).fit(X_scaled).inertia_ for k in␣
  ↪range(1,15)]
```

[9]:
```
WCSSs = []
Ks = range(1, 30)
for k in Ks:
    wcss = []
    for _ in range(30):  # Test 30 different random starting places
```

```
        kmeans = KMeans(n_clusters=k, n_init=1).fit(X_scaled)  # Set n_init to␣
↪1 to test different random starts
        wcss.append(kmeans.inertia_)
    WCSSs.append(sum(wcss) / len(wcss))  # Average WCSS over the 30 runs
```

5 4.) Use the above work and economic critical thinking to choose a number of clusters. Explain why you chose the number of clusters and fit a model accordingly.

[10]:
```
plt.plot(Ks,WCSSs)
plt.xlabel("# of clusters")
plt.ylabel("WCSSs")
plt.show()
```



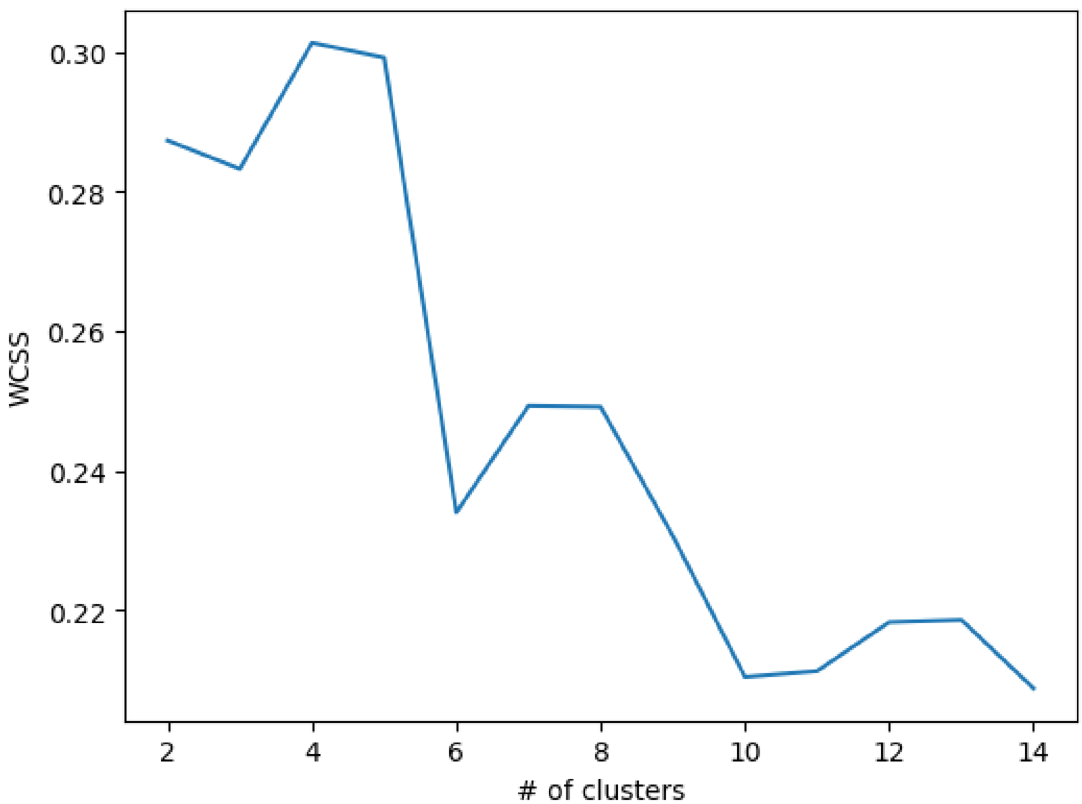
We chose 6 clusters based on the "elbow point" observed in the plot of within-cluster sum of squares (WCSS) versus the number of clusters. The elbow point signifies the point at which the rate of decrease in WCSS slows down significantly. 6 clusters is the optimal number to balance model complexity and explanatory power, aiming to capture meaningful patterns in the data without overfitting. Beyond 8 clusters, the reduction in WCSS didn't justify the addition of more clusters in explaining the variance in the data.

6 6.) Do the same for a silhoutte plot

[11]:
```
from sklearn.metrics import silhouette_score
```

[12]:
```
SSs = []
Ks = range(2,15)
for k in Ks:
    kmeans = KMeans(n_clusters = k,n_init = 30).fit(X_scaled)
    sil = silhouette_score(X_scaled,kmeans.labels_)
    SSs.append(sil)
```

[13]:
```
plt.plot(Ks,SSs)
plt.xlabel("# of clusters")
plt.ylabel("WCSS")
plt.show()
```



We choose 5 clusters based on the silhouette score analysis, as this number corresponds to the configuration where the data points are most appropriately grouped into clusters. The highest silhouette score around 5 clusters indicates better-defined clusters and greater separation between them. By selecting 5 clusters, we ensure that each cluster captures a meaningful and distinct subset of the data. Moreover, having fewer clusters simplifies the model and enhances interpretability, which is beneficial in various applications.

7 7.) Create a list of the countries that are in each cluster. Write interesting things you notice.

[14]:
```
# Fit KMeans model with the chosen number of clusters

kmeans_final = KMeans(n_clusters=5, n_init=30).fit(X_scaled)
```

[15]:
```
preds = pd.DataFrame(kmeans.labels_)
```

[16]:
```
output = pd.concat([preds,data],axis=1)
```

[17]:
```
from collections import defaultdict

# Create a dictionary to store countries in each cluster
clusters_countries = defaultdict(list)

# Iterate through each country and its corresponding cluster label
for country, label in zip(data["country"], kmeans_final.labels_):
    clusters_countries[label].append(country)

# Print the countries in each cluster
for cluster, countries_list in clusters_countries.items():
    print(f"Cluster {cluster + 1}: {countries_list}")
```

```
Cluster 1: ['Afghanistan', 'Angola', 'Benin', 'Botswana', 'Burkina Faso',
'Burundi', 'Cameroon', 'Central African Republic', 'Chad', 'Comoros', 'Congo,
Dem. Rep.', 'Congo, Rep.', "Cote d'Ivoire", 'Equatorial Guinea', 'Eritrea',
'Gabon', 'Gambia', 'Ghana', 'Guinea', 'Guinea-Bissau', 'Haiti', 'Iraq', 'Kenya',
'Kiribati', 'Lao', 'Lesotho', 'Liberia', 'Madagascar', 'Malawi', 'Mali',
'Mauritania', 'Mozambique', 'Namibia', 'Niger', 'Pakistan', 'Rwanda', 'Senegal',
'Sierra Leone', 'South Africa', 'Sudan', 'Tanzania', 'Timor-Leste', 'Togo',
'Uganda', 'Yemen', 'Zambia']
Cluster 3: ['Albania', 'Algeria', 'Antigua and Barbuda', 'Argentina', 'Armenia',
'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus',
'Belize', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Brazil', 'Bulgaria',
'Cambodia', 'Cape Verde', 'Chile', 'China', 'Colombia', 'Costa Rica', 'Croatia',
'Czech Republic', 'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador',
'Estonia', 'Fiji', 'Georgia', 'Grenada', 'Guatemala', 'Guyana', 'Hungary',
'India', 'Indonesia', 'Iran', 'Jamaica', 'Jordan', 'Kazakhstan', 'Kyrgyz
Republic', 'Latvia', 'Lebanon', 'Libya', 'Lithuania', 'Macedonia, FYR',
'Malaysia', 'Maldives', 'Mauritius', 'Micronesia, Fed. Sts.', 'Moldova',
'Mongolia', 'Montenegro', 'Morocco', 'Myanmar', 'Nepal', 'Oman', 'Panama',
'Paraguay', 'Peru', 'Philippines', 'Poland', 'Romania', 'Russia', 'Samoa',
'Saudi Arabia', 'Serbia', 'Seychelles', 'Slovak Republic', 'Solomon Islands',
'Sri Lanka', 'St. Vincent and the Grenadines', 'Suriname', 'Tajikistan',
```

```
'Thailand', 'Tonga', 'Tunisia', 'Turkey', 'Turkmenistan', 'Ukraine', 'Uruguay',
'Uzbekistan', 'Vanuatu', 'Venezuela', 'Vietnam']
Cluster 2: ['Australia', 'Austria', 'Belgium', 'Brunei', 'Canada', 'Cyprus',
'Denmark', 'Finland', 'France', 'Germany', 'Greece', 'Iceland', 'Ireland',
'Israel', 'Italy', 'Japan', 'Kuwait', 'Netherlands', 'New Zealand', 'Norway',
'Portugal', 'Qatar', 'Slovenia', 'South Korea', 'Spain', 'Sweden',
'Switzerland', 'United Arab Emirates', 'United Kingdom', 'United States']
Cluster 4: ['Luxembourg', 'Malta', 'Singapore']
Cluster 5: ['Nigeria']
```

Cluster 1: This cluster represents a mix of developed and developing nations from various regions. It includes countries like Australia, Canada, Japan, and the United States, indicating a blend of advanced economies with high standards of living and diverse industrial bases.

Cluster 2: This cluster is predominantly composed of African countries, such as Angola, Burundi, and Zambia, among others. These nations likely share common challenges like poverty, political instability, and limited access to healthcare and education, reflecting socio-economic disparities within the continent.

Cluster 3: Luxembourg, Malta, and Singapore are the countries in this cluster, known for their high levels of prosperity, economic stability, and business-friendly environments. These nations serve as global financial centers and boast high per capita incomes, indicating advanced economies with favorable living conditions.

Cluster 4: The countries in this cluster are a mix of emerging and established economies, including Brazil, China, India, and Russia. This diversity suggests varying levels of development and economic growth potential among these nations, with some transitioning towards becoming major global players.

Cluster 5: Nigeria stands alone in this cluster, highlighting its unique socio-economic challenges compared to other countries in the dataset. As one of the most populous countries in Africa, Nigeria faces issues like political instability, economic inequality, and infrastructure deficits, requiring targeted interventions to address its specific needs.

8 8.) Create a table of Descriptive Statistics. Rows being the Cluster number and columns being all the features. Values being the mean of the centroid. Use the nonscaled X values for interpretotation

[18]:
```
output
```

[18]:
```
      0              country  child_mort  exports  health  imports  income  \
0    11          Afghanistan        90.2     10.0    7.58     44.9    1610
1     9              Albania        16.6     28.0    6.55     48.6    9930
2     6              Algeria        27.3     38.4    4.17     31.4   12900
3    12               Angola       119.0     62.3    2.85     42.9    5900
4     8  Antigua and Barbuda        10.3     45.5    6.03     58.9   19100
..   ..                  …           …        …       …        …       …
```

```
162   0            Vanuatu        29.2      46.6     5.25      52.7     2950
163   6          Venezuela        17.1      28.5     4.91      17.6    16500
164   8            Vietnam        23.3      72.0     6.84      80.2     4490
165   1              Yemen        56.3      30.0     5.18      34.4     4480
166  11             Zambia        83.1      37.0     5.89      30.9     3280

     inflation  life_expec  total_fer   gdpp
0         9.44        56.2       5.82    553
1         4.49        76.3       1.65   4090
2        16.10        76.5       2.89   4460
3        22.40        60.1       6.16   3530
4         1.44        76.8       2.13  12200
..         …           …         …     …
162       2.62        63.0       3.50   2970
163      45.90        75.4       2.47  13500
164      12.10        73.1       1.95   1310
165      23.60        67.5       4.67   1310
166      14.00        52.0       5.40   1460

[167 rows x 11 columns]
```

[19]:
```python
# Get centroids from the KMeans model
centroids = kmeans_final.cluster_centers_

# Create a DataFrame to store descriptive statistics
cluster_stats = pd.DataFrame(centroids, columns=X.columns)

# Add a column for cluster number
cluster_stats['Cluster'] = range(1, len(centroids) + 1)

# Set the cluster number as the index
cluster_stats.set_index('Cluster', inplace=True)

# Display the table
cluster_stats
```

[19]:
```
         child_mort   exports    health   imports    income  inflation  \
Cluster
1          1.340192 -0.434470 -0.145518 -0.166756 -0.688260   0.212380
2         -0.828609  0.172621  0.859190 -0.296373  1.462275  -0.478189
3         -0.419827  0.006648 -0.211724  0.047581 -0.217274  -0.034953
4         -0.849003  4.935673 -0.008163  4.548058  2.439542  -0.504206
5          2.281385 -0.578452 -0.637438 -1.221785 -0.624065   9.129718

         life_expec  total_fer      gdpp
Cluster
1         -1.285398   1.352961 -0.604727
```

```
2          1.107649  -0.763681  1.661902
3          0.268420  -0.438222 -0.330805
4          1.226824  -1.038863  2.440797
5         -1.134121   1.916133 -0.581936
```

9 9.) Write an observation about the descriptive statistics.

Cluster 1: This cluster exhibits relatively low values for child mortality, income, inflation, and GDP per capita, indicating regions with moderate development levels. However, it shows higher values for life expectancy, suggesting better access to healthcare and overall well-being compared to other clusters.

Cluster 2: Countries in this cluster demonstrate high values for child mortality and inflation, suggesting significant socio-economic challenges such as poverty and inadequate healthcare. Additionally, they exhibit low values for income, life expectancy, and GDP per capita, indicating underdeveloped economies with limited resources.

Cluster 3: This cluster is distinguished by exceptionally high values in exports, imports, and GDP per capita. These countries likely have strong trade economies with robust economic activity and development, despite moderate values for other indicators like child mortality and health.

Cluster 4: Countries in this cluster show moderate to high values for health, income, life expectancy, and GDP per capita, suggesting relatively developed and prosperous regions with adequate healthcare and economic stability.

Cluster 5: This cluster is characterized by extremely high values for child mortality and inflation, indicating significant socio-economic challenges and potential instability. Despite the high inflation rates, GDP per capita remains relatively low, reflecting economic struggles in these countries.