9 - Clustering Python

March 8, 2024

1 0.) Import and Clean data

[1]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

[2]:

```python
#drive.mount('/content/gdrive/', force_remount = True)
df = pd.read_csv("Country-data.csv", sep = ",")
```

[3]:

```python
df.head()
```

[3]:

```
               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

   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
```

[4]:

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

[5]:

```python
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
```

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

[6]:

```
kmeans = KMeans(n_clusters = 5).fit(X_scaled)
```

3 2.) Pick two features to visualize across

[7]:

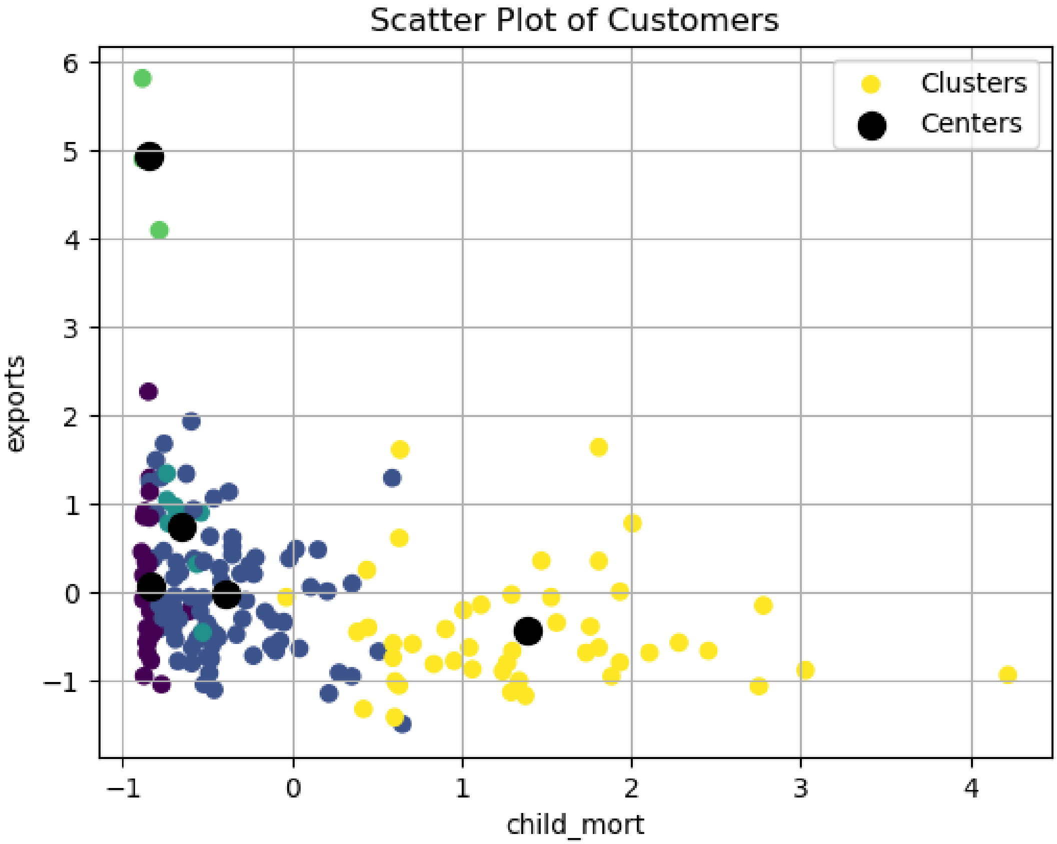
```
X.columns
```

[7]:

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

[8]:

```
import matplotlib.pyplot as plt

x1_index = 0
x2_index = 1


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

[9]:

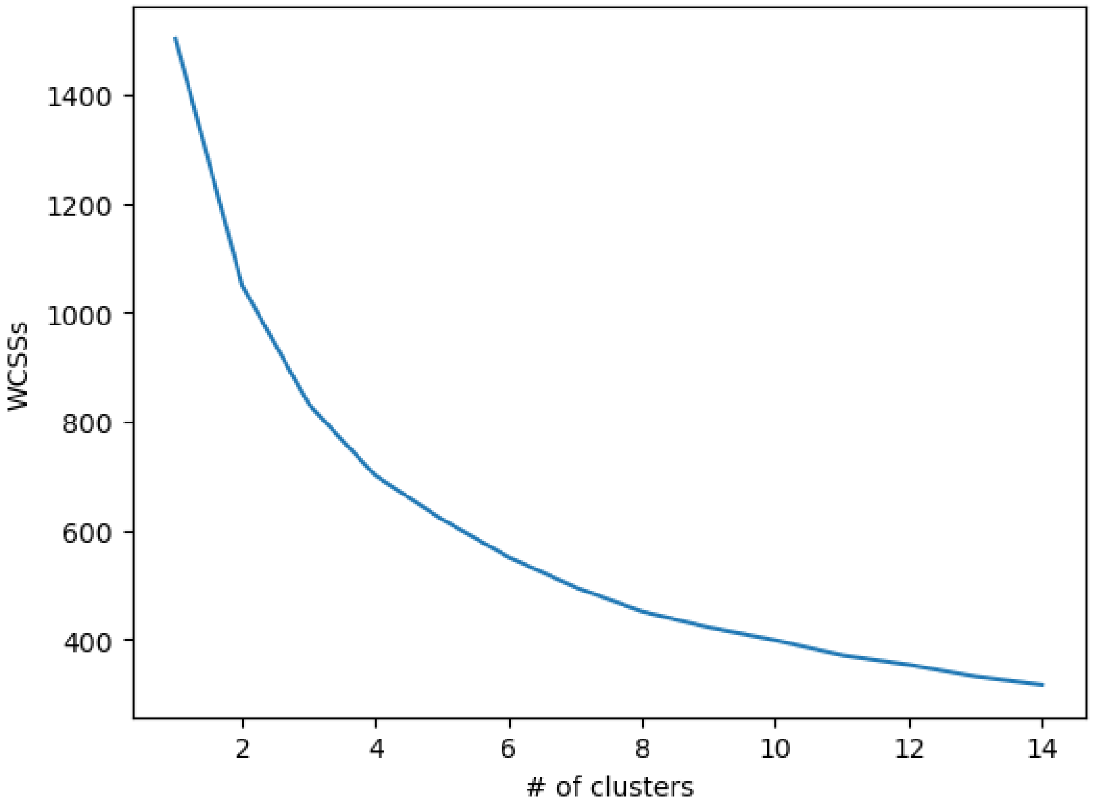
```
WCSSs = []
Ks = range(1,15)
for k in Ks:
    kmeans = KMeans(n_clusters = k, n_init = 30).fit(X_scaled)
    WCSSs.append(kmeans.inertia_)
```

[10]:

```
# OPTINIONAL DO IN 1 LINE OF CODE
WCSSs = [KMeans(n_clusters = k, n_init = 30).fit(X_scaled).inertia_ for k in␣
  ↪range (1,15)]
```

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.

[11]:
```
plt.plot(Ks,WCSSs)
plt.xlabel("# of clusters")
plt.ylabel("WCSSs")
plt.show()
```



The point on the graph (the WCSS against the number of clusters) where the slope of the curve becomes less steep indicates the elbow.

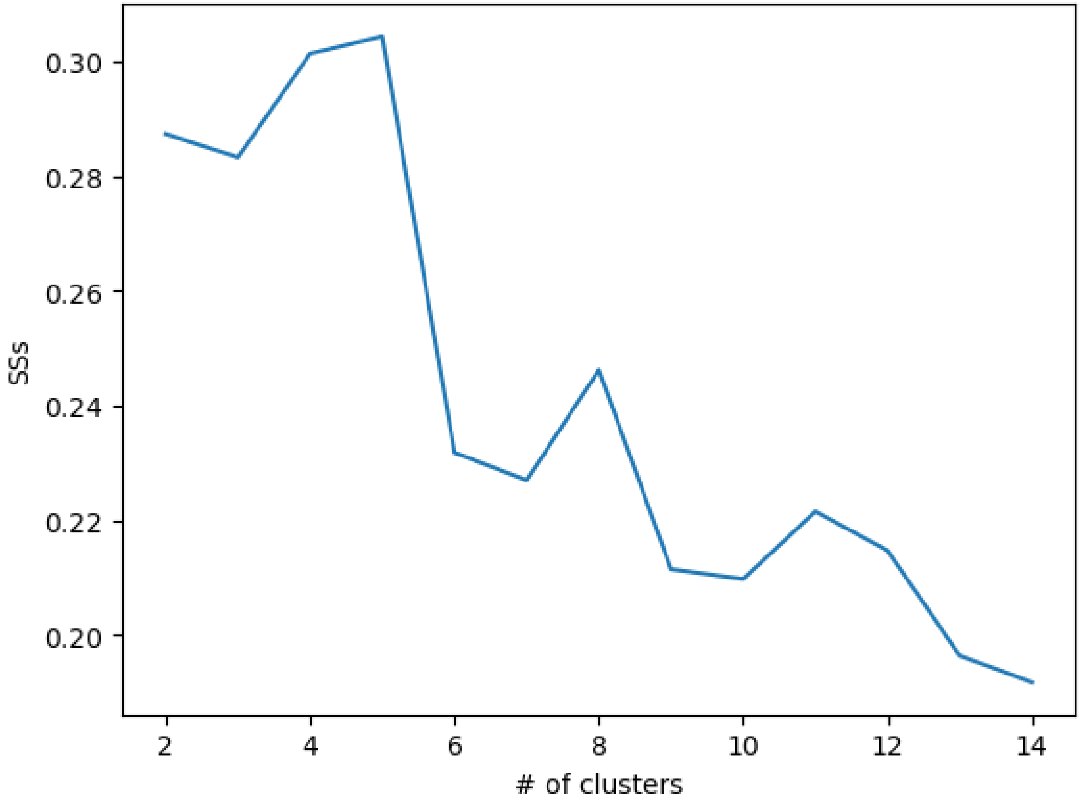
6 6.) Do the same for a silhoutte plot

[12]:
```
from sklearn.metrics import silhouette_score
```

[13]:
```
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)
```

[14]:

```
plt.plot(Ks,SSs)
plt.xlabel("# of clusters")
plt.ylabel("SSs")
plt.show()
```



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

[15]:

```
kmeans = KMeans(n_clusters = 2, n_init = 30).fit(X_scaled)
```

[16]:

```
preds = pd.DataFrame(kmeans.labels_)
```

[17]:

```
output = pd.concat([preds, df], axis =1)
```

[18]:

```
print("Cluster 1: ")
list(output.loc[output[0]==1,"country"])
```

```
    Cluster 1:
```

```
[18]: ['Albania',
       'Algeria',
       'Antigua and Barbuda',
       'Argentina',
       'Armenia',
       'Australia',
       'Austria',
       'Azerbaijan',
       'Bahamas',
       'Bahrain',
       'Barbados',
       'Belarus',
       'Belgium',
       'Belize',
       'Bhutan',
       'Bosnia and Herzegovina',
       'Brazil',
       'Brunei',
       'Bulgaria',
       'Canada',
       'Cape Verde',
       'Chile',
       'China',
       'Colombia',
       'Costa Rica',
       'Croatia',
       'Cyprus',
       'Czech Republic',
       'Denmark',
       'Dominican Republic',
       'Ecuador',
       'El Salvador',
       'Estonia',
       'Fiji',
       'Finland',
       'France',
       'Georgia',
       'Germany',
       'Greece',
       'Grenada',
       'Hungary',
       'Iceland',
       'Iran',
       'Ireland',
       'Israel',
       'Italy',
       'Jamaica',
       'Japan',
       'Jordan',
       'Kazakhstan',
       'Kuwait',
       'Latvia',
       'Lebanon',
       'Libya',
       'Lithuania',
       'Luxembourg',
       'Macedonia, FYR',
       'Malaysia',
       'Maldives',
       'Malta',
       'Mauritius',
       'Moldova',
       'Montenegro',
       'Morocco',
       'Netherlands',
       'New Zealand',
       'Norway',
       'Oman',
       'Panama',
       'Paraguay',
       'Peru',
       'Poland',
       'Portugal',
       'Qatar',
       'Romania',
       'Russia',
       'Saudi Arabia',
       'Serbia',
       'Seychelles',
       'Singapore',
       'Slovak Republic',
       'Slovenia',
       'South Korea',
       'Spain',
       'Sri Lanka',
       'St. Vincent and the Grenadines',
       'Suriname',
       'Sweden',
       'Switzerland',
       'Thailand',
       'Tunisia',
       'Turkey',
        'Ukraine',
        'United Arab Emirates',
        'United Kingdom',
        'United States',
        'Uruguay',
        'Venezuela',
        'Vietnam']
```

```
[19]: print("Cluster 2: ")
      list(output.loc[output[0]==0,"country"])
```

```
Cluster 2:
```

```
[19]: ['Afghanistan',
       'Angola',
       'Bangladesh',
       'Benin',
       'Bolivia',
       'Botswana',
       'Burkina Faso',
       'Burundi',
       'Cambodia',
       'Cameroon',
       'Central African Republic',
       'Chad',
       'Comoros',
       'Congo, Dem. Rep.',
       'Congo, Rep.',
       "Cote d'Ivoire",
       'Egypt',
       'Equatorial Guinea',
       'Eritrea',
       'Gabon',
       'Gambia',
       'Ghana',
       'Guatemala',
       'Guinea',
       'Guinea-Bissau',
       'Guyana',
       'Haiti',
       'India',
       'Indonesia',
       'Iraq',
       'Kenya',
       'Kiribati',
       'Kyrgyz Republic',
       'Lao',
 'Lesotho',
 'Liberia',
 'Madagascar',
 'Malawi',
 'Mali',
 'Mauritania',
 'Micronesia, Fed. Sts.',
 'Mongolia',
 'Mozambique',
 'Myanmar',
 'Namibia',
 'Nepal',
 'Niger',
 'Nigeria',
 'Pakistan',
 'Philippines',
 'Rwanda',
 'Samoa',
 'Senegal',
 'Sierra Leone',
 'Solomon Islands',
 'South Africa',
 'Sudan',
 'Tajikistan',
 'Tanzania',
 'Timor-Leste',
 'Togo',
 'Tonga',
 'Turkmenistan',
 'Uganda',
 'Uzbekistan',
 'Vanuatu',
 'Yemen',
 'Zambia']
```

Write an observation The output separates the countries into two distinct clusters. Cluster 1 includes countries that are developing countries, and Cluster 2 includes countries that are more developed with higher levels of income. In Cluster 2, we can also see the presence of countries with rapid economic growth such as China and India.

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 interprotation

```
[20]: output.drop("country",axis=1).groupby(0).mean()
```

```
[20]:    child_mort    exports    health    imports        income  inflation  \
      0
      0   76.280882  30.198515  6.090147  43.642146   4227.397059  11.098750
      1   12.161616  48.603030  7.314040  49.121212  26017.171717   5.503545

         life_expec  total_fer          gdpp
      0
      0   61.910294   4.413824   1981.235294
      1   76.493939   1.941111  20507.979798
```

```
[21]: output.drop("country",axis=1).groupby(0).std()
```

```
[21]:    child_mort    exports    health    imports        income  inflation  \
      0
      0   38.076068  18.201742  2.645319  19.323451   4890.581414  13.682630
      1    8.523122  30.116032  2.716652  26.928785  20441.749847   6.957187

         life_expec  total_fer          gdpp
      0
      0    6.897418   1.285590   2528.509189
      1    3.735757   0.486744  20578.727127
```

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

“0” is refered to cluster 2 (developed countries and emerging economies) and “1” is refered to cluster 1 (developing countries) in the previous discussion.

Mean:

In developed countries, indicators suh as income, healthcare spending, and life expectancy are higher. On the other hand, developing countries are struggling with higher child mortality, lower income, and lower gdpp.

Standard Deviation:

For developed countries, the standard deviations for indicators such as exports, income, and gdpp are relatively high. They suggest that there might be a notable range in the levels of economic prosperity. For developing countries, the standard deviations for indicators such as child mortality and inflation are relatively high. They indicate that a significant disparity within the overall living environment.

[ ]: