# Lab 9 - Clustering

```
import pandas as pd
```

```
seed = 42
```

## Loading data

```
df = pd.read_excel('/content/Dataset - Clustering - RO,MD Indicators.xlsx')
```

```
df
```

| | Judet | I1 | I2 | I3 | I4 | I5 | I6 |
|---|---|---|---|---|---|---|---|
| **0** | ALBA | 0.761092 | 0.823034 | 0.828689 | 0.841842 | 0.780330 | 0.698279 |
| **1** | ARAD | 0.774407 | 0.624350 | 0.925624 | 0.806822 | 0.865146 | 0.855394 |
| **2** | ARGES | 0.877874 | 0.786144 | 0.902367 | 0.886611 | 0.870798 | 0.827686 |
| **3** | BACAU | 0.787105 | 0.831500 | 0.864372 | 0.882119 | 0.826954 | 0.881788 |
| **4** | BIHOR | 0.834993 | 0.839020 | 0.887317 | 0.856610 | 0.862951 | 0.829769 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **72** | Straseni | 0.807418 | 0.362986 | 0.622310 | 0.763022 | 0.314345 | 0.281002 |
| **73** | Taraclia | 0.862238 | 0.290542 | 0.414033 | 0.808750 | 0.037037 | 0.040909 |
| **74** | Telenești | 0.774472 | 0.364485 | 0.599848 | 0.713706 | 0.257873 | 0.217551 |
| **75** | Ungheni | 0.677080 | 0.320349 | 0.530009 | 0.745171 | 0.216930 | 0.209819 |
| **76** | UTA | 0.726712 | 0.316935 | 0.535275 | 0.767424 | 0.350520 | 0.179316 |

77 rows × 7 columns

```
df.describe()
```

| | I1 | I2 | I3 | I4 | I5 | I6 |
|---|---|---|---|---|---|---|
| **count** | 77.000000 | 77.000000 | 77.000000 | 77.000000 | 77.000000 | 77.000000 |
| **mean** | 0.813594 | 0.585419 | 0.748161 | 0.824512 | 0.591256 | 0.543367 |
| **std** | 0.077420 | 0.246386 | 0.166150 | 0.076760 | 0.299842 | 0.302364 |
| **min** | 0.547929 | 0.233676 | 0.414033 | 0.605172 | 0.037037 | 0.040909 |
| **25%** | 0.774472 | 0.349993 | 0.599848 | 0.763022 | 0.296262 | 0.217551 |
| **50%** | 0.807418 | 0.707392 | 0.828689 | 0.834153 | 0.753057 | 0.698279 |
| **75%** | 0.862238 | 0.807769 | 0.891533 | 0.882119 | 0.858490 | 0.818960 |
| **max** | 0.983779 | 0.948722 | 1.000000 | 0.998691 | 0.995141 | 0.999295 |

## KMeans Clustering

```
#https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
from sklearn.cluster import KMeans
X = df[['I1', 'I2', 'I3', 'I4', 'I5', 'I6']]
kmeans = KMeans(n_clusters=3, random_state=seed)
labels = kmeans.fit_predict(X)
labels
```

```
array([2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 0, 2,
       0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2,
       1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int32)
```

```
df['Cluster'] = labels
df
```

|  | Judet | I1 | I2 | I3 | I4 | I5 | I6 | Cluster |
|---|---|---|---|---|---|---|---|---|
| **0** | ALBA | 0.761092 | 0.823034 | 0.828689 | 0.841842 | 0.780330 | 0.698279 | 2 |
| **1** | ARAD | 0.774407 | 0.624350 | 0.925624 | 0.806822 | 0.865146 | 0.855394 | 0 |
| **2** | ARGES | 0.877874 | 0.786144 | 0.902367 | 0.886611 | 0.870798 | 0.827686 | 0 |
| **3** | BACAU | 0.787105 | 0.831500 | 0.864372 | 0.882119 | 0.826954 | 0.881788 | 0 |
| **4** | BIHOR | 0.834993 | 0.839020 | 0.887317 | 0.856610 | 0.862951 | 0.829769 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **72** | Straseni | 0.807418 | 0.362986 | 0.622310 | 0.763022 | 0.314345 | 0.281002 | 1 |
| **73** | Taraclia | 0.862238 | 0.290542 | 0.414033 | 0.808750 | 0.037037 | 0.040909 | 1 |
| **74** | Telenești | 0.774472 | 0.364485 | 0.599848 | 0.713706 | 0.257873 | 0.217551 | 1 |
| **75** | Ungheni | 0.677080 | 0.320349 | 0.530009 | 0.745171 | 0.216930 | 0.209819 | 1 |
| **76** | UTA | 0.726712 | 0.316935 | 0.535275 | 0.767424 | 0.350520 | 0.179316 | 1 |

77 rows × 8 columns

```
df[df['Cluster']==2]
```

|  | Judet | I1 | I2 | I3 | I4 | I5 | I6 | Cluster |
|---|---|---|---|---|---|---|---|---|
| **0** | ALBA | 0.761092 | 0.823034 | 0.828689 | 0.841842 | 0.780330 | 0.698279 | 2 |
| **11** | CALARASI | 0.833768 | 0.808304 | 0.805630 | 0.877343 | 0.747411 | 0.747363 | 2 |
| **12** | CARAS-SEVERIN | 0.893498 | 0.757473 | 0.879292 | 0.855632 | 0.714524 | 0.632179 | 2 |
| **15** | COVASNA | 0.944860 | 0.808655 | 0.902600 | 0.879889 | 0.753057 | 0.698381 | 2 |
| **16** | DAMBOVITA | 0.983779 | 0.740090 | 0.852888 | 0.920682 | 0.785742 | 0.704838 | 2 |
| **19** | GIURGIU | 0.713053 | 0.674217 | 0.850415 | 0.785774 | 0.839178 | 0.734970 | 2 |
| **21** | HARGHITA | 0.787475 | 0.764650 | 0.815257 | 0.874513 | 0.784484 | 0.649371 | 2 |
| **30** | OLT | 0.880856 | 0.707392 | 0.893666 | 0.858199 | 0.803397 | 0.702313 | 2 |
| **31** | PRAHOVA | 0.802523 | 0.759684 | 0.836550 | 0.842831 | 0.787242 | 0.730909 | 2 |
| **32** | SALAJ | 0.868655 | 0.834314 | 0.864974 | 0.843510 | 0.736056 | 0.669758 | 2 |
| **43** | Bălți | 0.547929 | 0.474625 | 0.742933 | 0.605172 | 0.604511 | 0.511424 | 2 |
| **50** | Chisinău | 0.672834 | 0.491344 | 0.757701 | 0.692758 | 0.648607 | 0.513248 | 2 |

```
df.to_excel('result.xlsx', index=False)
```

## Scores

- https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.htm

```
#https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
X = df[['I1', 'I2', 'I3', 'I4', 'I5', 'I6']]

kmeans = KMeans(n_clusters=3, random_state=seed)
labels = kmeans.fit_predict(X)
sil_score = silhouette_score(X, labels)
sil_score
```

```
np.float64(0.7902707860706932)
```

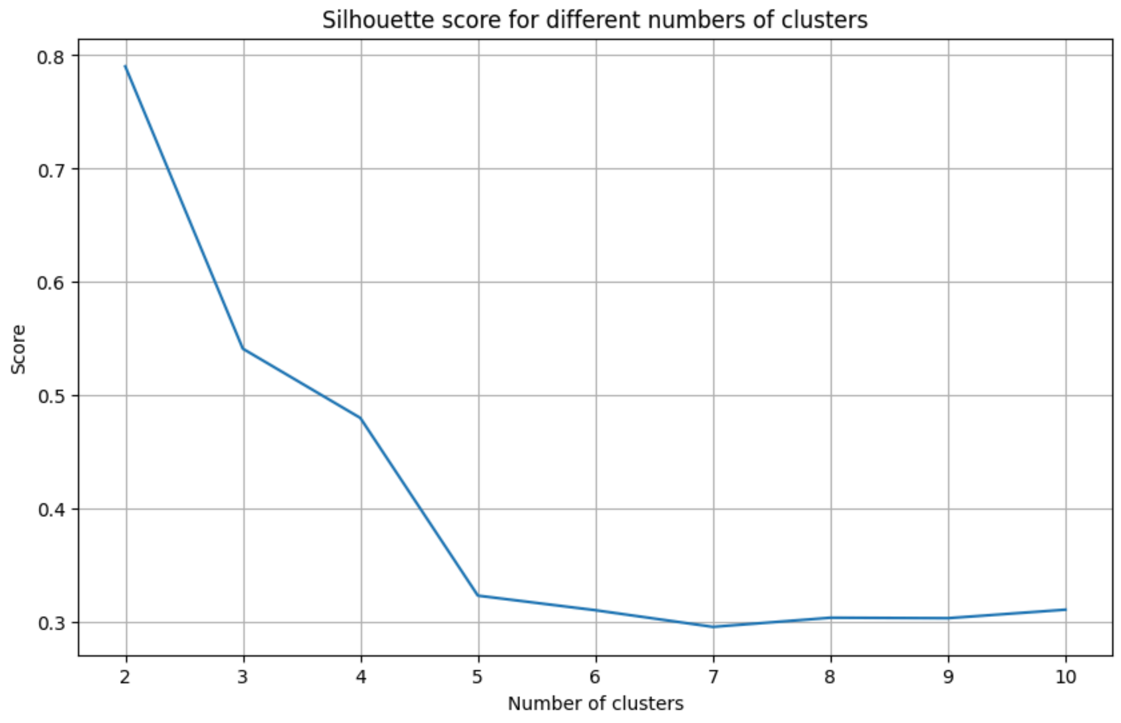
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score, calinski_harabasz_score

X = df[['I1', 'I2', 'I3', 'I4', 'I5', 'I6']]

scores = []
for n_clusters in range(2, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=seed)
    labels = kmeans.fit_predict(X)
    sil_score = silhouette_score(X, labels)
    db_score = davies_bouldin_score(X, labels)
    ch_score = calinski_harabasz_score(X, labels)
    scores.append([n_clusters, sil_score, db_score, ch_score])

scores = pd.DataFrame(scores, columns=['n_clusters', 'silhouette_score', 'davies_bouldin_score', 'calinski_harabasz_score'
scores
```

| | n_clusters | silhouette_score | davies_bouldin_score | calinski_harabasz_score |
|---|---|---|---|---|
| **0** | 2 | 0.790271 | 0.277278 | 681.575889 |
| **1** | 3 | 0.541230 | 0.919559 | 448.606424 |
| **2** | 4 | 0.480130 | 0.819642 | 405.239139 |
| **3** | 5 | 0.323396 | 1.077088 | 353.149875 |
| **4** | 6 | 0.310529 | 1.126234 | 327.643770 |
| **5** | 7 | 0.295835 | 1.169393 | 300.819487 |
| **6** | 8 | 0.303912 | 1.052773 | 287.269944 |
| **7** | 9 | 0.303493 | 1.059563 | 274.756459 |
| **8** | 10 | 0.311004 | 1.044359 | 266.985681 |

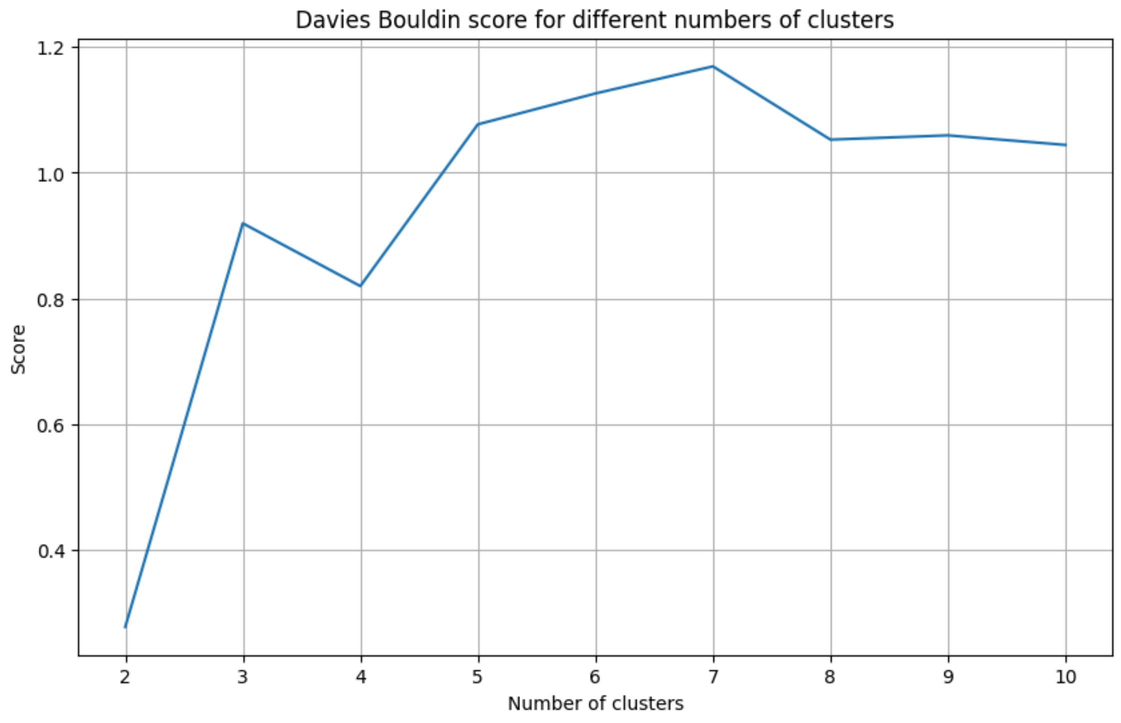
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(scores['n_clusters'], scores['silhouette_score'], label='silhouette_score')
plt.xlabel('Number of clusters')
plt.ylabel('Score')
plt.grid(True)
plt.title('Silhouette score for different numbers of clusters')
```

```
Text(0.5, 1.0, 'Silhouette score for different numbers of clusters')
```



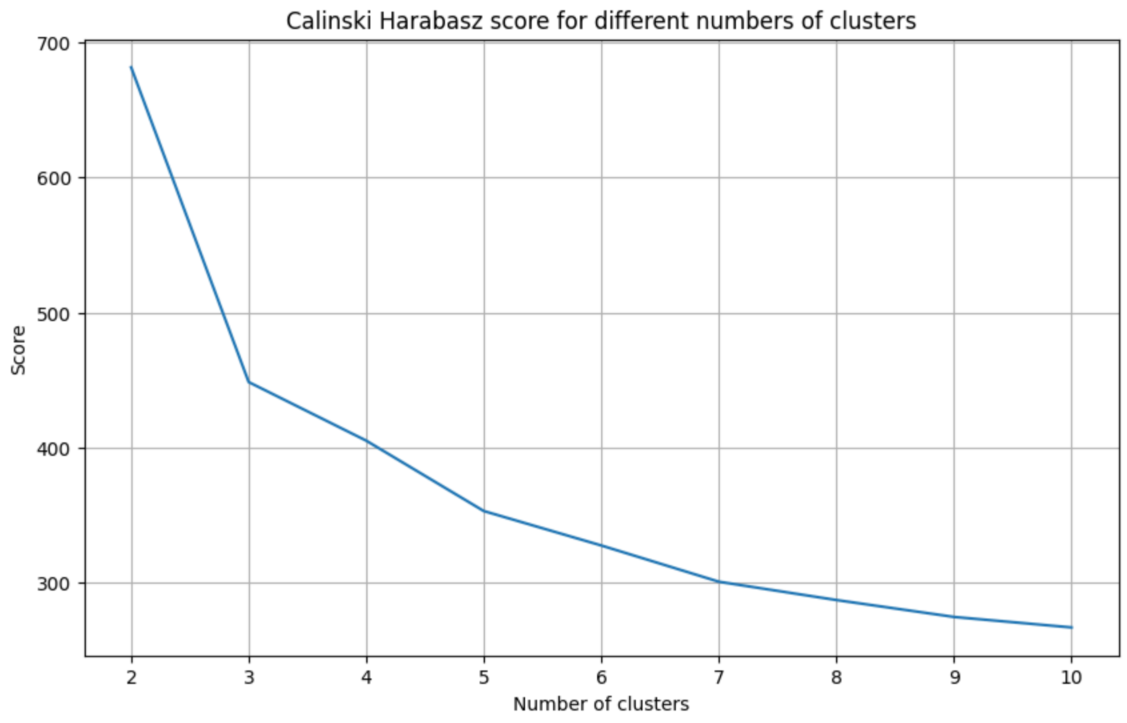
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(scores['n_clusters'], scores['davies_bouldin_score'], label='davies_bouldin_score')
plt.xlabel('Number of clusters')
plt.ylabel('Score')
plt.grid(True)
plt.title('Davies Bouldin score for different numbers of clusters')
```

```
Text(0.5, 1.0, 'Davies Bouldin score for different numbers of clusters')
```



```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(scores['n_clusters'], scores['calinski_harabasz_score'], label='calinski_harabasz_score')
plt.xlabel('Number of clusters')
plt.ylabel('Score')
plt.grid(True)
plt.title('Calinski Harabasz score for different numbers of clusters')
```

```
Text(0.5, 1.0, 'Calinski Harabasz score for different numbers of clusters')
```

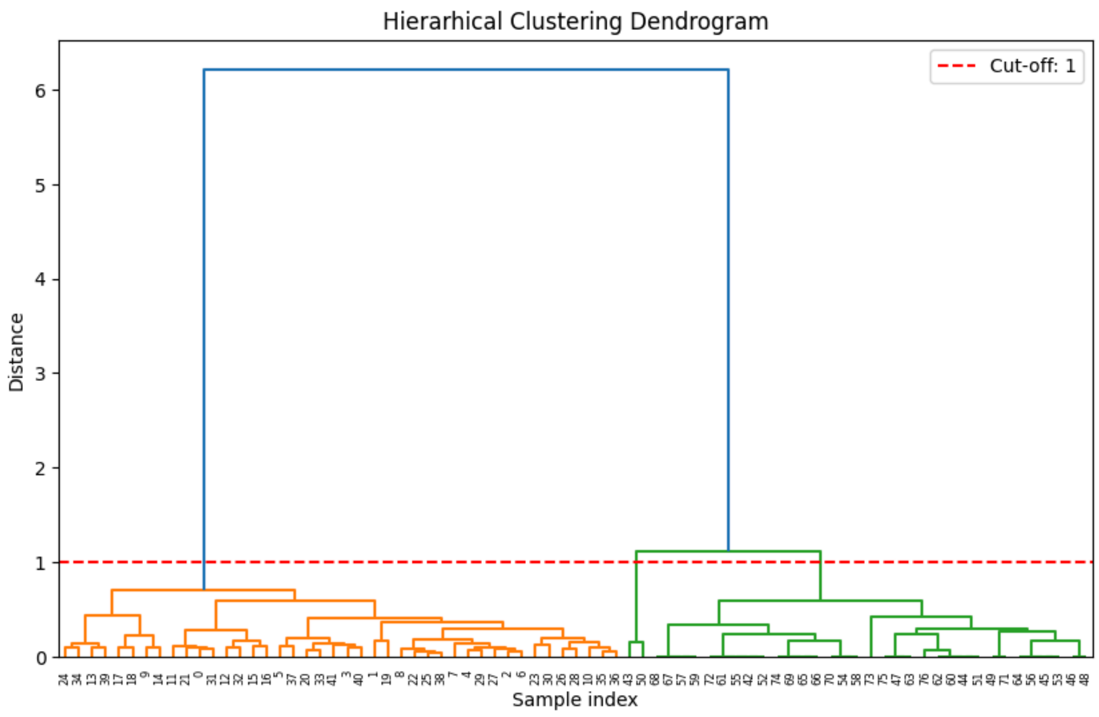


## Dendogram

> Documentation: https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html

```
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage

X = df[['I1', 'I2', 'I3', 'I4', 'I5', 'I6']]
```

```
Z = linkage(X, method='ward')

plt.figure(figsize=(10, 6))
dendrogram(Z)
plt.xlabel('Sample index')
plt.ylabel('Distance')
plt.title('Hierarhical Clustering Dendrogram')

cut_off = 1
#plt.axhline(y=cut_off, color='r', linestyle='--')
plt.axhline(y=cut_off, color='r', linestyle='--', label=f'Cut-off: {cut_off}')
plt.legend()
plt.show()
```


Hierarhical Clustering Dendrogram
Cut-off: 1
Distance
Sample index