

Pracadomowa5

May 16, 2021

1 Wczytanie danych

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
sns.set_theme(style="whitegrid")
sns.set_palette('Accent_r')

from matplotlib import pyplot as plt
```

```
[2]: data = pd.read_csv("../..//clustering.csv", header=None, names = ['X', 'Y'])
print(len(data))
data.head()
```

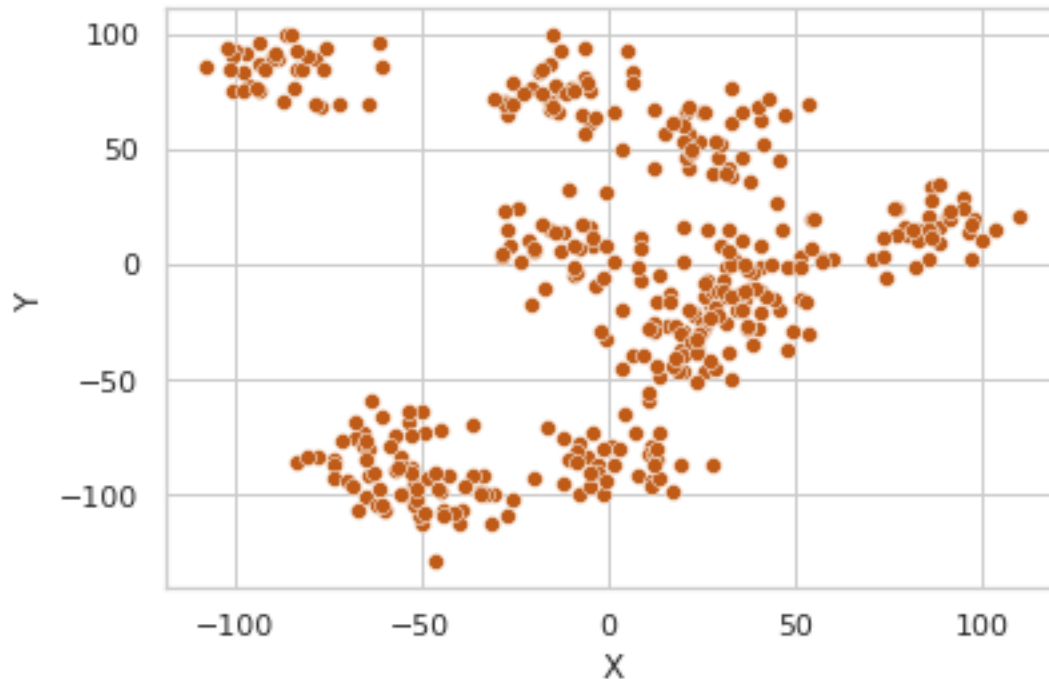
400

```
[2]:
```

	X	Y
0	41.788907	52.220182
1	-96.586516	90.957033
2	-54.143591	-99.153377
3	19.929231	-45.859779
4	-82.941076	84.099186

```
[3]: sns.scatterplot(data=data, x='X', y='Y')
```

```
[3]: <AxesSubplot:xlabel='X', ylabel='Y'>
```

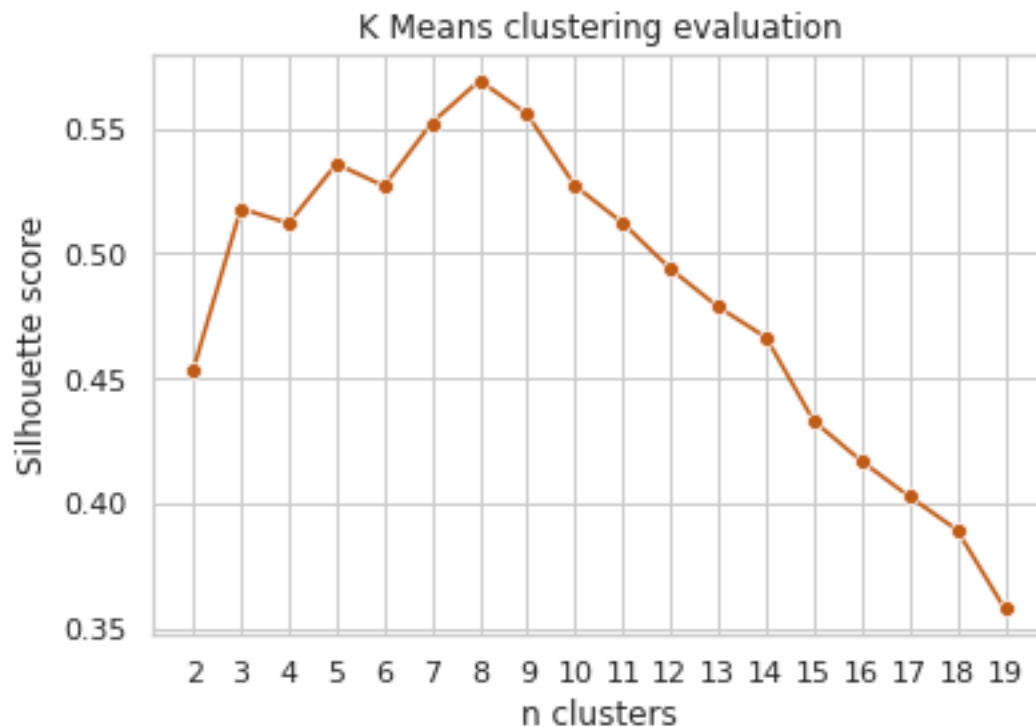


2 Klastrowanie metodą k-means oceniane współczynnikiem Silhouette

```
[4]: from sklearn.cluster import KMeans
      from sklearn.metrics import silhouette_score
```

```
[5]: n_clusters = [i for i in range(2,20)]
      silhouette_scores = []
      labels_list = []
      for n in n_clusters:
          kmeans = KMeans(n_clusters=n, random_state=123).fit(data)
          labels = kmeans.labels_
          score = silhouette_score(data, labels)
          labels_list.append(labels)
          silhouette_scores.append(score)
```

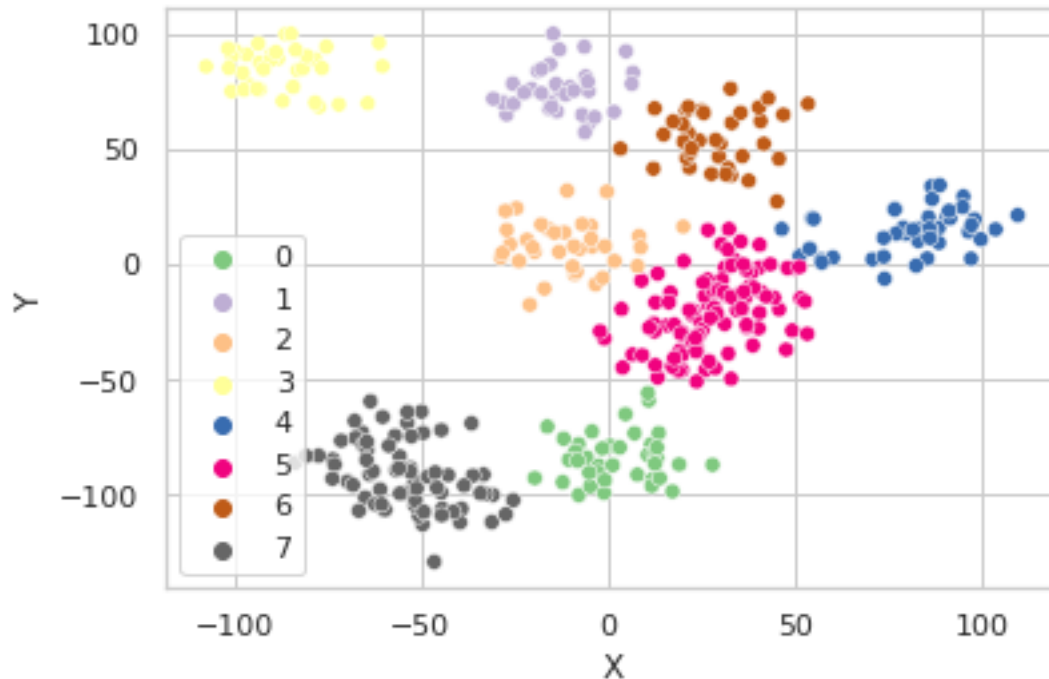
```
[6]: ax = sns.lineplot(x=n_clusters, y=silhouette_scores, marker='o', dashes=False)
      ax.set(xlabel='n clusters',
              ylabel='Silhouette score',
              title='K Means clustering evaluation',
              xticks=n_clusters)
      plt.show()
```



Na powyższym wykresie widzimy, że wybrana metryka przyjmuje największą wartość, gdy liczba klastrow jest równa 8. To oznacza, że wtedy klastry są najlepiej zdefiniowane.

```
[7]: sns.scatterplot(data=data, x='X', y='Y', hue=labels_list[6], palette='Accent')
```

```
[7]: <AxesSubplot:xlabel='X', ylabel='Y'>
```

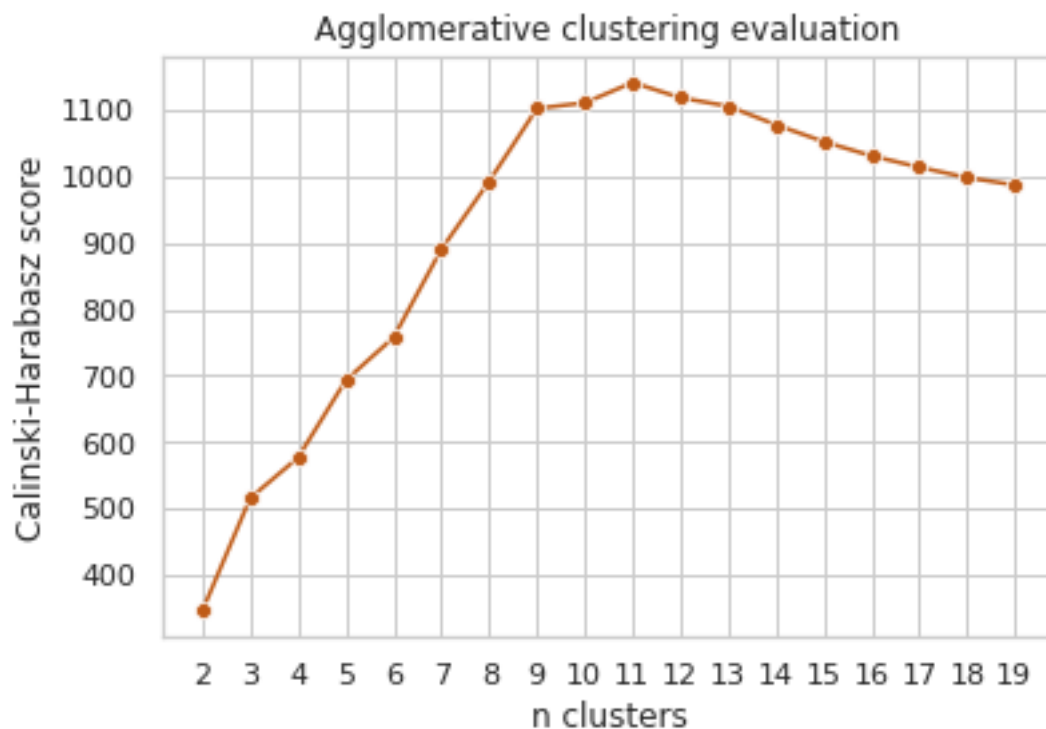


3 Klastrowanie metodą aglomeracyjną z użyciem metryki Calinski-Harabasz Index

```
[8]: from sklearn.cluster import AgglomerativeClustering
      from sklearn.metrics import calinski_harabasz_score
```

```
[9]: n_clusters = [i for i in range(2,20)]
      c_h_scores = []
      labels_list = []
      for n in n_clusters:
          agglomerative_clusters = AgglomerativeClustering(n_clusters=n).fit(data)
          labels = agglomerative_clusters.labels_
          score = calinski_harabasz_score(data, labels)
          labels_list.append(labels)
          c_h_scores.append(score)
```

```
[10]: ax = sns.lineplot(x=n_clusters, y=c_h_scores, marker='o', dashes=False)
       ax.set(xlabel='n clusters',
              ylabel='Calinski-Harabasz score',
              title='Agglomerative clustering evaluation',
              xticks=n_clusters)
       plt.show()
```



Największy wynik indeksu Calinski-Harabasz jest osiągany dla 11 klastrów. To najlepszy podział według tej metryki.

```
[11]: sns.scatterplot(data=data, x='X', y='Y', hue=labels_list[9], palette='Paired')
```

```
[11]: <AxesSubplot:xlabel='X', ylabel='Y'>
```

