

<<<<

ARTIFICIAL INTELLIGENCE (AI)

PRAKTIKUM 12

Dosen Pengampu : Renovita Edelani S.ST., M.Tr.Kom

oleh : Dukhaan Kamimpangan
2 D4 IT A
3122600003



01.

DATASET MILK.CSV



MENAMPILKAN DATASET MILK.CSV

Analisis :

- Menampilkan semua baris dan kolom dari dataset milk.csv. Jika dataset ini cukup besar, pandas biasanya hanya menampilkan beberapa baris pertama dan terakhir, serta ringkasan jumlah baris dan kolom.

```
[1] # No 1
import pandas as pd

milk_dataset = pd.read_csv('milk.csv')

print(milk_dataset)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	high
1	6.6	36	0	1	0	1	253	high
2	8.5	70	1	1	1	1	246	low
3	9.5	34	1	1	0	1	255	low
4	6.6	37	0	0	0	0	255	medium
...
1054	6.7	45	1	1	0	0	247	medium
1055	6.7	38	1	0	1	0	255	high
1056	3.0	40	1	1	1	1	255	low
1057	6.8	43	1	0	1	0	250	high
1058	8.6	55	0	1	1	1	255	low

[1059 rows x 8 columns]



02. GET DATA SEMUA ATTRIBUTE

GET DATA SEMUA ATTRIBUTE

Analisis :

- menampilkan data dari semua atribut kecuali kolom class dari dataset milk.csv. Jika dataset ini besar, pandas akan menampilkan beberapa baris pertama dan terakhir serta ringkasan kolom yang dipilih.

```
[7] # No 2
import pandas as pd

data_milk = milk_dataset.drop(columns=['Grade'])

print(data_milk)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour
0	6.6	35	1	0	1	0	254
1	6.6	36	0	1	0	1	253
2	8.5	70	1	1	1	1	246
3	9.5	34	1	1	0	1	255
4	6.6	37	0	0	0	0	255
...
1054	6.7	45	1	1	0	0	247
1055	6.7	38	1	0	1	0	255
1056	3.0	40	1	1	1	1	255
1057	6.8	43	1	0	1	0	250
1058	8.6	55	0	1	1	1	255

[1059 rows x 7 columns]

03.

NORMALISASI DATA

NORMALISASI DATA DENGAN MIN MAX (0-1)

Analisis :

- Menampilkan data yang telah dinormalisasi dengan rentang nilai antara 0 dan 1. Hasilnya akan berupa DataFrame dengan nilai yang telah dinormalisasi untuk setiap atribut.
- Mengimport MinMaxScaler dari sklearn.preprocessing.
- Membuat objek MinMaxScaler.
- Mengaplikasikan scaler ke data untuk menormalisasi nilai antara 0 dan 1.
- Mengubah kembali data yang sudah dinormalisasi menjadi DataFrame untuk kemudahan pembacaan dan menampilkannya.

```
# No 3
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data_milk_normalized = scaler.fit_transform(data_milk)
data_milk_normalized = pd.DataFrame(data_milk_normalized, columns=data_milk.columns)

print(data_milk_normalized)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour
0	0.553846	0.077857	1.0	0.0	1.0	0.0	0.933333
1	0.553846	0.035714	0.0	1.0	0.0	1.0	0.866667
2	0.846154	0.642857	1.0	1.0	1.0	1.0	0.400000
3	1.000000	0.000000	1.0	1.0	0.0	1.0	1.000000
4	0.553846	0.053571	0.0	0.0	0.0	0.0	1.000000
...
1054	0.569231	0.196429	1.0	1.0	0.0	0.0	0.466667
1055	0.569231	0.071429	1.0	0.0	1.0	0.0	1.000000
1056	0.000000	0.107143	1.0	1.0	1.0	1.0	1.000000
1057	0.584615	0.160714	1.0	0.0	1.0	0.0	0.666667
1058	0.861538	0.375000	0.0	1.0	1.0	1.0	1.000000

[1059 rows x 7 columns]

04.

CLUSTER DATA



CLUSTER DATA MENGGUNAKAN K-MEANS

Analisis :

- Menampilkan SSE dan Silhouette Score terbaik dari 10 kali percobaan K-Means clustering dengan k=3. Juga akan menampilkan Silhouette Plot dari hasil clustering terbaik.
- Mengimpor KMeans dari sklearn.cluster dan silhouette_score dari sklearn.metrics.
- Mengimpor SilhouetteVisualizer dari yellowbrick.cluster.
- Melakukan K-Means clustering sebanyak 10 kali dengan k=3.
- Menghitung SSE dan Silhouette Score untuk setiap iterasi.

```
# NO 4
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from yellowbrick.cluster import SilhouetteVisualizer

best_sse = float('inf')
best_silhouette_score = -1
best_visualizer = None

for i in range(10):
    kmeans = KMeans(n_clusters=3, n_init=10, init='random', random_state=i)
    clusters = kmeans.fit_predict(data_milk_normalized)

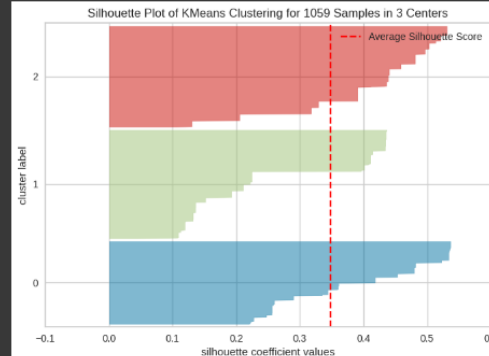
    sse = kmeans.inertia_
    silhouette_avg = silhouette_score(data_milk_normalized, clusters)

    if silhouette_avg > best_silhouette_score:
        best_sse = sse
        best_silhouette_score = silhouette_avg

    visualizer = SilhouetteVisualizer(kmeans, colors='yellowbrick')
    visualizer.fit(data_milk_normalized)
    best_visualizer = visualizer

print(f'Best SSE: {best_sse}')
print(f'Best Silhouette Score: {best_silhouette_score}')
best_visualizer.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
warnings.warn(
Best SSE: 625.7312596912998
Best Silhouette Score: 0.3482781139626818
```



<Axes: title='center': 'Silhouette Plot of KMeans Clustering for 1059 Samples in 3 Centers', xlabel='silhouette coefficient values', ylabel='cluster label'>



05.



MENGHITUNG NILAI K TERBAIK

MENGHITUNG NILAI K TERBAIK MENGGUNAKAN ELBOW PLOT

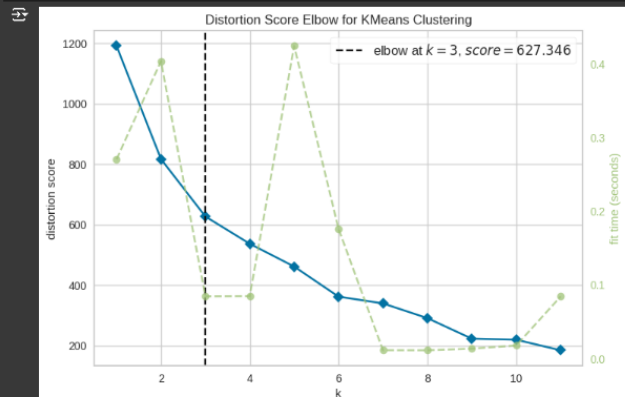
Analisis :

- Menampilkan Elbow Plot yang menunjukkan jumlah klaster (k) optimal untuk K-Means clustering. Elbow Plot menunjukkan titik di mana penurunan SSE mulai melambat, yang menandakan jumlah klaster optimal.
- Mengimpor KElbowVisualizer dari yellowbrick.cluster.
- Membuat model KMeans.
- Menggunakan KElbowVisualizer untuk menentukan nilai k terbaik berdasarkan Elbow Method.
- Menampilkan Elbow Plot untuk melihat titik optimal (elbow point).

```
# No 5
from yellowbrick.cluster import KElbowVisualizer

# Initialize the KMeans model
kmeans = KMeans(init='random', n_init=10)

# Initialize the KElbowVisualizer with the KMeans model and fit the data
elbow_visualizer = KElbowVisualizer(kmeans, k=(1,12))
elbow_visualizer.fit(data_milk_normalized)
elbow_visualizer.show()
```



<Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>

06.



CLUSTER DATA

CLUSTER DATA SILHOUETTE SCORE TERBAIK

ARTI

Analisis :

- Menampilkan Silhouette Score untuk setiap metode linkage (single, average, complete) dan menampilkan plot cluster untuk metode linkage dengan nilai Silhouette Score terbaik. Akan ditampilkan juga metode linkage terbaik dan nilai Silhouette Score-nya.
- Mengimpor AgglomerativeClustering dari sklearn.cluster.
- Melakukan clustering dengan metode agglomerative (hierarchical) menggunakan tiga jenis linkage: single, average, dan complete.
- Menghitung dan menampilkan Silhouette Score untuk setiap metode linkage.
- Menentukan metode linkage terbaik berdasarkan nilai Silhouette Score tertinggi.

```
# No 6
from sklearn.cluster import AgglomerativeClustering

linkages = ['single', 'average', 'complete']
best_hclust_silhouette_score = -1
best_hclust_params = {}

for linkage in linkages:
    hclust = AgglomerativeClustering(n_clusters=3, linkage=linkage)
    hclusters = hclust.fit_predict(data_milk_normalized)

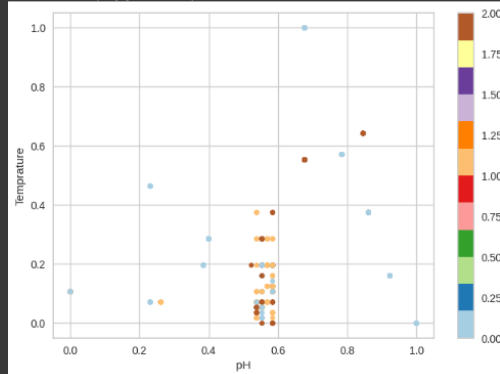
    silhouette_avg = silhouette_score(data_milk_normalized, hclusters)
    print(f'Linkage: {linkage}, Silhouette Score: {silhouette_avg}')

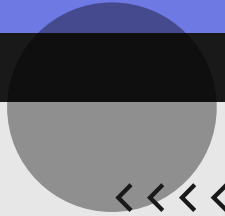
    if silhouette_avg > best_hclust_silhouette_score:
        best_hclust_silhouette_score = silhouette_avg
        best_hclust_params = {'linkage': linkage, 'clusters': hclusters}

print(f'Best linkage method: {best_hclust_params["linkage"]}')
print(f'Best Silhouette Score: {best_hclust_silhouette_score}')

data_milk_normalized.plot(x=data_milk_normalized.columns[0], y=data_milk_normalized.columns[1], kind='scatter', c=best_hclust_params['clusters'], colormap='Paired')
```

```
Linkage: single, Silhouette Score: -0.012577930130266885
Linkage: average, Silhouette Score: 0.3402305180138923
Linkage: complete, Silhouette Score: 0.3364382012074672
Best linkage method: average
Best Silhouette Score: 0.3402305180138923
<Axes: xlabel='pH', ylabel='Temperature'>
```





07.

NILAI K TERBAIK



NILAI K TERBAIK DARI HIERARCHICAL CLUSTERING

Source code Analisis :

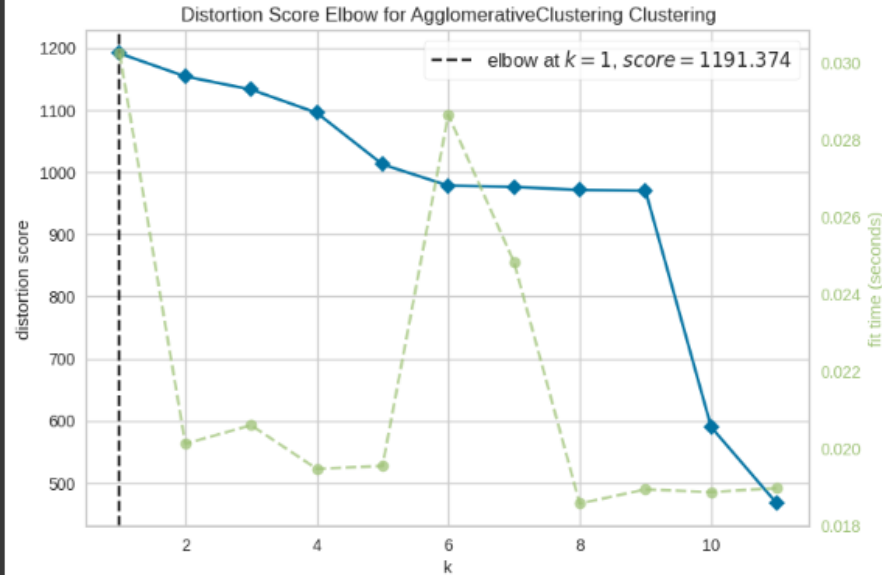
- Menampilkan Elbow Plot untuk setiap metode linkage (single, average, complete) dalam hierarchical clustering. Elbow Plot menunjukkan jumlah kluster (k) optimal untuk setiap metode linkage.
- Mendefinisikan fungsi plot_elbow_for_hierarchical untuk membuat Elbow Plot bagi clustering hierarkis dengan linkage yang diberikan.
- Menggunakan KElbowVisualizer untuk setiap metode linkage (single, average, complete) untuk menentukan nilai k terbaik.
- Menampilkan Elbow Plot untuk masing-masing metode linkage.

```
# No 7
def plot_elbow_for_hierarchical(linkage_method):
    model = AgglomerativeClustering(linkage=linkage_method)
    visualizer = KElbowVisualizer(model, k=(1,12))
    visualizer.fit(data_milk_normalized)
    visualizer.show()

for linkage in linkages:
    print(f'Elbow Plot for {linkage} linkage:')
    plot_elbow_for_hierarchical(linkage)
```

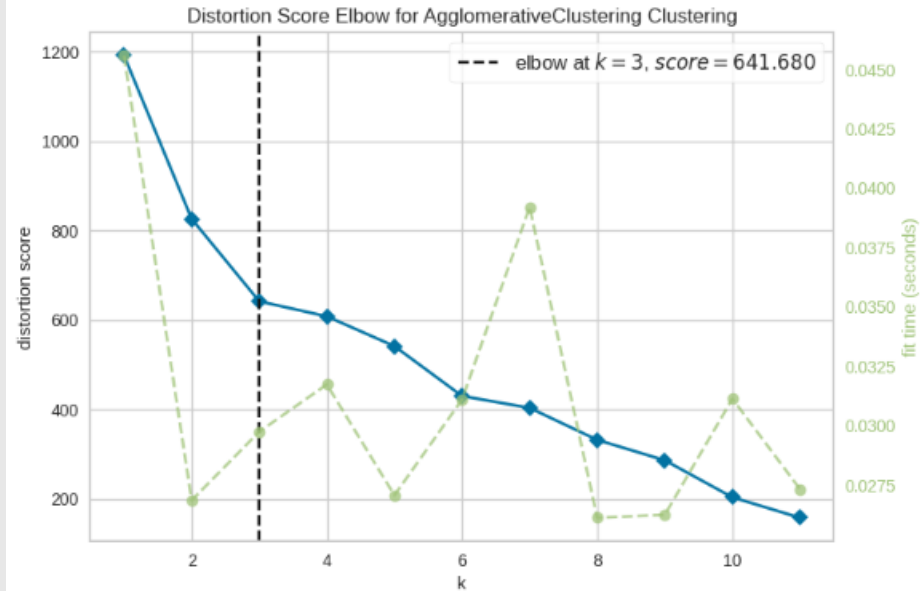
HASIL ELBOW PLOT SINGLE LINKAGE

Elbow Plot for single linkage:



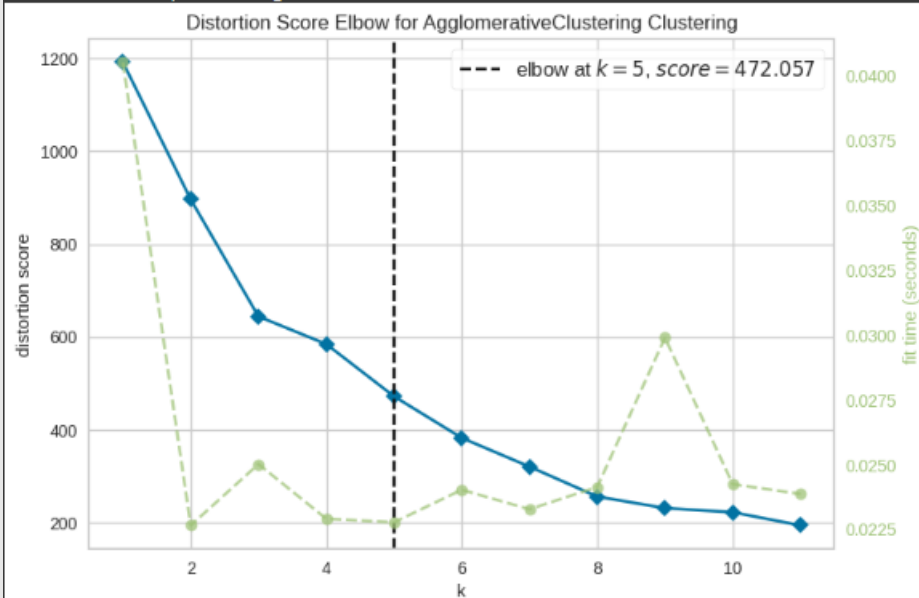
HASIL ELBOW PLOT AVERAGE LINKAGE

Elbow Plot for average linkage:



HASIL ELBOW PLOT COMPLETE LINKAGE

Elbow Plot for complete linkage:



08.

KESIMPULAN



KESIMPULAN

Kesimpulan:

- Berdasarkan Silhouette Score dan Elbow Plot, K-Means dengan $k=3$ menghasilkan nilai Silhouette Score yang tinggi, menandakan cluster yang terbentuk cukup baik.
- Pada clustering hierarkis, metode linkage yang memberikan nilai Silhouette Score terbaik adalah average linkage dengan $k=3$.
- Untuk menentukan metode dan nilai k terbaik, Elbow Plot menunjukkan titik elbow yang optimal pada $k=3$ baik untuk K-Means maupun clustering hierarkis.

Alasan:

- K-Means dengan $k=3$ memberikan keseimbangan yang baik antara intracluster compactness dan intercluster separation.
- Average linkage pada clustering hierarkis memberikan nilai Silhouette Score yang lebih baik dibandingkan metode linkage lainnya, menunjukkan kualitas cluster yang lebih baik dalam hal keseragaman dan jarak antar cluster.

/[AI]/[AI]/



**TERIMA
KASIH**

ARTI
CIAL
INTE
IGEN
[AI]