

# part\_1\_green

January 23, 2025

## 1 Imports

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from scipy.stats import zscore
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN, \
    ↪SpectralClustering
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score, davies_bouldin_score, \
    ↪calinski_harabasz_score
from sklearn.neighbors import kneighbors_graph
from sklearn.decomposition import PCA
```

## 2 Load Data

```
[2]: # Load the data
file_path = "fic_epita_kantar_codes.csv"
data_codes = pd.read_csv(file_path, sep=';')
```

```
[3]: # Select relevant columns
columns_of_interest = [
    "A11", "A12", "A13", "A14", "A4", "A5", "A5bis",
    "A8_1_slice", "A8_2_slice", "A8_3_slice", "A8_4_slice",
    "B1_1_slice", "B1_2_slice", "B2_1_slice", "B2_2_slice",
    "B3", "B4", "B6",
    "C1_1_slice", "C1_2_slice", "C1_3_slice", "C1_4_slice",
    "C1_5_slice", "C1_6_slice", "C1_7_slice", "C1_8_slice", "C1_9_slice"
]
subset_data = data_codes[columns_of_interest]
```

```
[4]: # Check for missing values
missing_values = subset_data.isnull().sum()
print("Missing Values:\n", missing_values)
```

```

# Data type analysis
data_types = subset_data.dtypes
print("\nData Types:\n", data_types)

# Summary statistics
summary_stats = subset_data.describe()
print("\nSummary Statistics:\n", summary_stats)

# Check for outliers
outliers = subset_data.apply(zscore).abs() > 3
print("\nOutliers:\n", outliers)

```

Missing Values:

|            |      |
|------------|------|
| A11        | 0    |
| A12        | 0    |
| A13        | 0    |
| A14        | 0    |
| A4         | 0    |
| A5         | 1443 |
| A5bis      | 3557 |
| A8_1_slice | 0    |
| A8_2_slice | 0    |
| A8_3_slice | 0    |
| A8_4_slice | 0    |
| B1_1_slice | 0    |
| B1_2_slice | 0    |
| B2_1_slice | 0    |
| B2_2_slice | 0    |
| B3         | 0    |
| B4         | 0    |
| B6         | 0    |
| C1_1_slice | 0    |
| C1_2_slice | 0    |
| C1_3_slice | 0    |
| C1_4_slice | 0    |
| C1_5_slice | 0    |
| C1_6_slice | 0    |
| C1_7_slice | 0    |
| C1_8_slice | 0    |
| C1_9_slice | 0    |

dtype: int64

Data Types:

|     |       |
|-----|-------|
| A11 | int64 |
| A12 | int64 |
| A13 | int64 |
| A14 | int64 |
| A4  | int64 |

```

A5          float64
A5bis       float64
A8_1_slice  int64
A8_2_slice  int64
A8_3_slice  int64
A8_4_slice  int64
B1_1_slice  int64
B1_2_slice  int64
B2_1_slice  int64
B2_2_slice  int64
B3          int64
B4          int64
B6          int64
C1_1_slice  int64
C1_2_slice  int64
C1_3_slice  int64
C1_4_slice  int64
C1_5_slice  int64
C1_6_slice  int64
C1_7_slice  int64
C1_8_slice  int64
C1_9_slice  int64
dtype: object

```

Summary Statistics:

|       | A11         | A12         | A13         | A14    | A4 \        |
|-------|-------------|-------------|-------------|--------|-------------|
| count | 5000.000000 | 5000.000000 | 5000.000000 | 5000.0 | 5000.000000 |
| mean  | 0.781800    | 0.225000    | 0.402200    | 0.0    | 1.421600    |
| std   | 0.413065    | 0.417624    | 0.490391    | 0.0    | 0.714112    |
| min   | 0.000000    | 0.000000    | 0.000000    | 0.0    | 1.000000    |
| 25%   | 1.000000    | 0.000000    | 0.000000    | 0.0    | 1.000000    |
| 50%   | 1.000000    | 0.000000    | 0.000000    | 0.0    | 1.000000    |
| 75%   | 1.000000    | 0.000000    | 1.000000    | 0.0    | 2.000000    |
| max   | 1.000000    | 1.000000    | 1.000000    | 0.0    | 3.000000    |

|       | A5          | A5bis       | A8_1_slice  | A8_2_slice  | A8_3_slice  | ... \ |
|-------|-------------|-------------|-------------|-------------|-------------|-------|
| count | 3557.000000 | 1443.000000 | 5000.000000 | 5000.000000 | 5000.000000 | ...   |
| mean  | 2.326399    | 1.821206    | 5.122200    | 5.521400    | 3.791200    | ...   |
| std   | 1.243436    | 0.987393    | 6.738067    | 7.001067    | 5.524971    | ...   |
| min   | 1.000000    | 1.000000    | 0.000000    | 0.000000    | 0.000000    | ...   |
| 25%   | 1.000000    | 1.000000    | 1.000000    | 1.000000    | 1.000000    | ...   |
| 50%   | 2.000000    | 2.000000    | 3.000000    | 3.000000    | 2.000000    | ...   |
| 75%   | 3.000000    | 2.000000    | 6.000000    | 7.000000    | 5.000000    | ...   |
| max   | 5.000000    | 4.000000    | 50.000000   | 50.000000   | 50.000000   | ...   |

|       | B6          | C1_1_slice  | C1_2_slice  | C1_3_slice  | C1_4_slice \ |
|-------|-------------|-------------|-------------|-------------|--------------|
| count | 5000.000000 | 5000.000000 | 5000.000000 | 5000.000000 | 5000.000000  |
| mean  | 3.269000    | 4.192400    | 4.131200    | 4.550600    | 4.614200     |

|     |          |          |          |          |          |
|-----|----------|----------|----------|----------|----------|
| std | 0.875438 | 0.876091 | 0.897635 | 0.773278 | 0.784781 |
| min | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 25% | 3.000000 | 4.000000 | 4.000000 | 4.000000 | 4.000000 |
| 50% | 4.000000 | 4.000000 | 4.000000 | 5.000000 | 5.000000 |
| 75% | 4.000000 | 5.000000 | 5.000000 | 5.000000 | 5.000000 |
| max | 4.000000 | 5.000000 | 5.000000 | 5.000000 | 5.000000 |

|       |             |             |             |             |             |
|-------|-------------|-------------|-------------|-------------|-------------|
|       | C1_5_slice  | C1_6_slice  | C1_7_slice  | C1_8_slice  | C1_9_slice  |
| count | 5000.000000 | 5000.000000 | 5000.000000 | 5000.000000 | 5000.000000 |
| mean  | 4.493000    | 4.682200    | 4.541000    | 4.336600    | 4.352400    |
| std   | 0.884368    | 0.726711    | 0.788951    | 0.822336    | 0.849446    |
| min   | 1.000000    | 1.000000    | 1.000000    | 1.000000    | 1.000000    |
| 25%   | 4.000000    | 5.000000    | 4.000000    | 4.000000    | 4.000000    |
| 50%   | 5.000000    | 5.000000    | 5.000000    | 5.000000    | 5.000000    |
| 75%   | 5.000000    | 5.000000    | 5.000000    | 5.000000    | 5.000000    |
| max   | 5.000000    | 5.000000    | 5.000000    | 5.000000    | 5.000000    |

[8 rows x 27 columns]

Outliers:

|      |       |       |       |       |       |       |       |            |            |
|------|-------|-------|-------|-------|-------|-------|-------|------------|------------|
|      | A11   | A12   | A13   | A14   | A4    | A5    | A5bis | A8_1_slice | A8_2_slice |
| \    |       |       |       |       |       |       |       |            |            |
| 0    | False | False | False | False | False | False | False | False      | False      |
| 1    | False | False | False | False | False | False | False | False      | False      |
| 2    | False | False | False | False | False | False | False | False      | False      |
| 3    | False | False | False | False | False | False | False | False      | False      |
| 4    | False | False | False | False | False | False | False | False      | False      |
| ...  | ...   | ...   | ...   | ...   | ...   | ...   | ...   | ...        | ...        |
| 4995 | False | False | False | False | False | False | False | False      | False      |
| 4996 | False | False | False | False | False | False | False | False      | False      |
| 4997 | False | False | False | False | False | False | False | False      | False      |
| 4998 | False | False | False | False | False | False | False | False      | False      |
| 4999 | False | False | False | False | False | False | False | False      | False      |

|      |            |     |       |            |            |            |            |   |
|------|------------|-----|-------|------------|------------|------------|------------|---|
|      | A8_3_slice | ... | B6    | C1_1_slice | C1_2_slice | C1_3_slice | C1_4_slice | \ |
| 0    | False      | ... | False | False      | False      | True       | True       |   |
| 1    | False      | ... | False | False      | False      | False      | False      |   |
| 2    | False      | ... | False | False      | False      | False      | False      |   |
| 3    | False      | ... | False | False      | False      | False      | False      |   |
| 4    | False      | ... | False | False      | False      | False      | False      |   |
| ...  | ...        | ... | ...   | ...        | ...        | ...        | ...        |   |
| 4995 | False      | ... | False | False      | False      | False      | False      |   |
| 4996 | False      | ... | False | False      | False      | False      | False      |   |
| 4997 | False      | ... | False | False      | False      | False      | False      |   |
| 4998 | False      | ... | False | False      | False      | False      | False      |   |
| 4999 | False      | ... | False | False      | False      | False      | False      |   |

C1\_5\_slice C1\_6\_slice C1\_7\_slice C1\_8\_slice C1\_9\_slice

|      |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|
| 0    | False | True  | True  | False | False |
| 1    | False | False | False | False | False |
| 2    | False | False | False | False | False |
| 3    | False | False | False | False | False |
| 4    | False | False | False | False | False |
| ...  | ...   | ...   | ...   | ...   | ...   |
| 4995 | False | False | False | False | False |
| 4996 | False | False | False | False | False |
| 4997 | False | False | False | False | False |
| 4998 | False | False | False | False | False |
| 4999 | False | False | False | False | False |

[5000 rows x 27 columns]

```
[5]: # Handle NaN values in columns A5 and A5bis
subset_data['A5'].fillna(subset_data['A5'].mean(), inplace=True)
subset_data['A5bis'].fillna(subset_data['A5bis'].mean(), inplace=True)
```

/tmp/ipykernel\_21221/3626455305.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
subset_data['A5'].fillna(subset_data['A5'].mean(), inplace=True)
/tmp/ipykernel_21221/3626455305.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
subset_data['A5'].fillna(subset_data['A5'].mean(), inplace=True)
/tmp/ipykernel_21221/3626455305.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
```

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

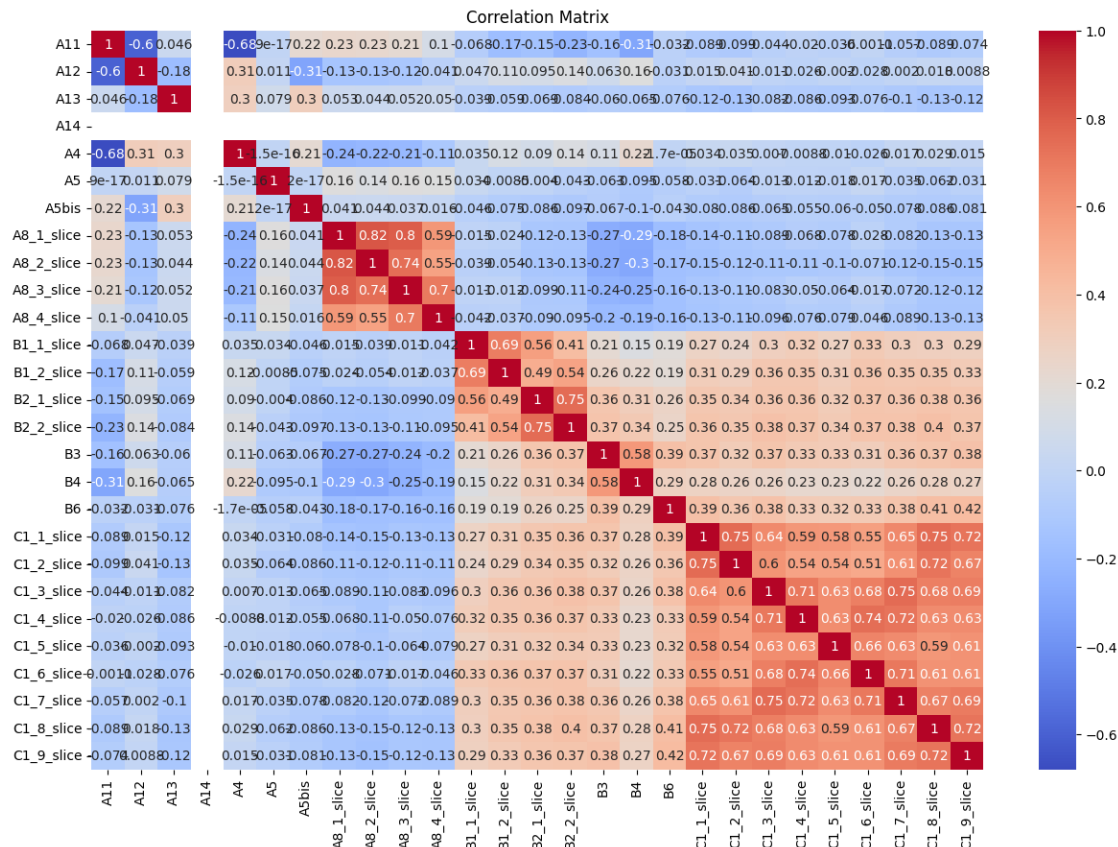
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
subset_data['A5bis'].fillna(subset_data['A5bis'].mean(), inplace=True)
/tmp/ipykernel_21221/3626455305.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

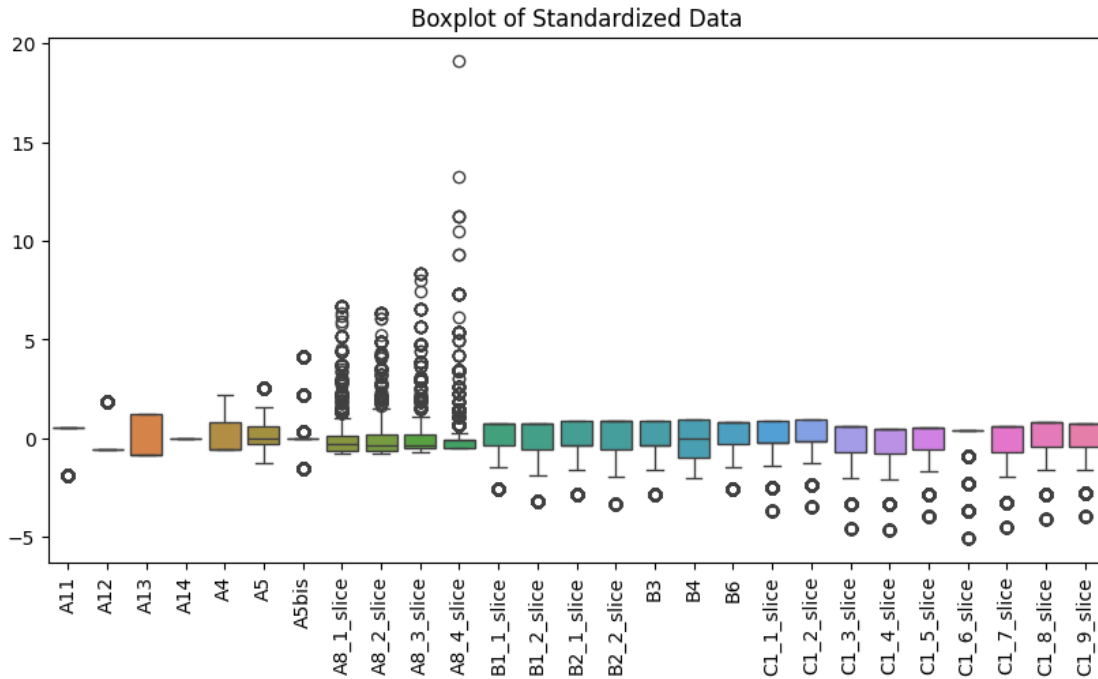
```
subset_data['A5bis'].fillna(subset_data['A5bis'].mean(), inplace=True)
```

```
[6]: # Correlation heatmap
plt.figure(figsize=(15, 10))
sns.heatmap(subset_data.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```



```
[7]: # Standardize data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(subset_data)
```

```
[8]: # Check if clustering is appropriate (variance and distribution)
plt.figure(figsize=(10, 5))
sns.boxplot(data=pd.DataFrame(scaled_data, columns=columns_of_interest))
plt.xticks(rotation=90)
plt.title("Boxplot of Standardized Data")
plt.show()
```



### 3 Clustering

```
[9]: # Function to evaluate clustering
def evaluate_clustering(labels, data):
    if len(set(labels)) > 1: # At least 2 clusters
        silhouette = silhouette_score(data, labels)
        davies_bouldin = davies_bouldin_score(data, labels)
        calinski_harabasz = calinski_harabasz_score(data, labels)
    else:
        silhouette, davies_bouldin, calinski_harabasz = None, None, None
    return silhouette, davies_bouldin, calinski_harabasz

results = {}
```

### 3.1 Kmeans

```
[10]: from sklearn.model_selection import ParameterGrid
      from sklearn.decomposition import PCA

      def grid_search_kmeans(data, param_grid):
          results = []
          for params in ParameterGrid(param_grid):
              kmeans = KMeans(**params, random_state=42)
              kmeans_labels = kmeans.fit_predict(data)
              score = evaluate_clustering(kmeans_labels, data)
              results.append({'params': params, 'score': score})
          return results

      # Try multiple values of PCA components
      pca_results = {}
      pca_components = [2, 3, 4, 5] # List of PCA components to try

      for n_components in pca_components:
          pca = PCA(n_components=n_components)
          reduced_data = pca.fit_transform(scaled_data)

          # Define the parameter grid for KMeans
          param_grid = {
              'n_clusters': [3, 4, 5, 6, 7, 8, 9],
              'init': ['k-means++', 'random'],
              'max_iter': [100, 300],
              'n_init': [10, 20]
          }

          pca_results[n_components] = grid_search_kmeans(reduced_data, param_grid)

      results['K-Means'] = pca_results
```

### 3.2 Hierarchical Clustering

```
[11]: from sklearn.model_selection import ParameterGrid
      from sklearn.decomposition import PCA

      def grid_search_agglo(data, param_grid):
          results = []
          for params in ParameterGrid(param_grid):
              agglo = AgglomerativeClustering(**params)
              agglo_labels = agglo.fit_predict(data)
              score = evaluate_clustering(agglo_labels, data)
              results.append({'params': params, 'score': score})
          return results
```



```

# Try multiple values of PCA components
pca_results = {}
pca_components = [2, 3, 4, 5] # List of PCA components to try

for n_components in pca_components:
    pca = PCA(n_components=n_components)
    reduced_data = pca.fit_transform(scaled_data)

    # Define the parameter grid for Agglomerative Clustering
    param_grid = {
        'n_clusters': [3, 4, 5, 6, 7, 8, 9],
        'linkage': ['ward', 'complete', 'average', 'single']
    }

    pca_results[n_components] = grid_search_agglo(reduced_data, param_grid)

results['Agglomerative'] = pca_results

```

### 3.3 DBSCAN

```

[12]: from sklearn.model_selection import ParameterGrid

def grid_search_dbscan(data, param_grid):
    results = []
    for params in ParameterGrid(param_grid):
        dbscan = DBSCAN(**params)
        dbscan_labels = dbscan.fit_predict(data)
        score = evaluate_clustering(dbscan_labels, data)
        results.append({'params': params, 'score': score})
    return results

# Try multiple values of PCA components
pca_results = {}
pca_components = [2, 3, 4, 5] # List of PCA components to try

for n_components in pca_components:
    pca = PCA(n_components=n_components)
    reduced_data = pca.fit_transform(scaled_data)

    # Define the parameter grid for Agglomerative Clustering
    param_grid = {
        'eps': [0.5, 1.0, 1.5, 2.0],
        'min_samples': [3, 5, 10, 15],
    }

    pca_results[n_components] = grid_search_dbscan(reduced_data, param_grid)

```

```
results['DBSCAN'] = pca_results
```

### 3.4 Gaussian Mixture Model

```
[13]: from sklearn.model_selection import ParameterGrid

def grid_search_gmm(data, param_grid):
    results = []
    for params in ParameterGrid(param_grid):
        gmm = GaussianMixture(**params)
        gmm_labels = gmm.fit_predict(data)
        score = evaluate_clustering(gmm_labels, data)
        results.append({'params': params, 'score': score})
    return results

# Try multiple values of PCA components
pca_results = {}
pca_components = [2, 3, 4, 5] # List of PCA components to try

for n_components in pca_components:
    pca = PCA(n_components=n_components)
    reduced_data = pca.fit_transform(scaled_data)

    # Define the parameter grid for Agglomerative Clustering
    param_grid = {
        'n_components': [3, 4, 5, 6, 7, 8, 9],
        'covariance_type': ['full', 'tied', 'diag', 'spherical'],
        'random_state': [42]
    }

    pca_results[n_components] = grid_search_gmm(reduced_data, param_grid)

results['GMM'] = pca_results
```

### 3.5 Spectral Clustering

```
[14]: from sklearn.model_selection import ParameterGrid

def grid_search_spectral(data, param_grid):
    results = []
    for params in ParameterGrid(param_grid):
        spectral = SpectralClustering(**params)
        spectral_labels = spectral.fit_predict(data)
        score = evaluate_clustering(spectral_labels, data)
        results.append({'params': params, 'score': score})
    return results
```

```

# Try multiple values of PCA components
pca_results = {}
pca_components = [2, 3, 4, 5] # List of PCA components to try

for n_components in pca_components:
    pca = PCA(n_components=n_components)
    reduced_data = pca.fit_transform(scaled_data)

    # Define the parameter grid for Agglomerative Clustering
    param_grid = {
        'n_clusters': [3, 4, 5, 6, 7, 8, 9],
        'affinity': ['nearest_neighbors'],
        'random_state': [42]
    }

    pca_results[n_components] = grid_search_spectral(reduced_data, param_grid)

results['Spectral'] = pca_results

```

```

/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(

```



```

/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(
/home/floflo/Documents/epita/epita-ml-scia/lib/python3.12/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
    warnings.warn(

```

```
[15]: print(results)
```

```

{'K-Means': {2: [{'params': {'init': 'k-means++', 'max_iter': 100, 'n_clusters':
3, 'n_init': 10}, 'score': (0.4024973580346682, 0.9384735512109831,
3911.7411719820284)}, {'params': {'init': 'k-means++', 'max_iter': 100,
'n_clusters': 3, 'n_init': 20}, 'score': (0.4024973580346682,
0.9384735512109831, 3911.7411719820284)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 10}, 'score': (0.43696184323503295,
0.800807992091704, 4382.46484845105)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 20}, 'score': (0.43670553196911205,
0.8015655276876479, 4382.643219886097)}, {'params': {'init': 'k-means++',

```

```

'max_iter': 100, 'n_clusters': 5, 'n_init': 10}, 'score': (0.36501089678069776,
0.8390424953834879, 4390.695531815)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 20}, 'score': (0.3650631940423554,
0.8385143445254073, 4390.769580451708)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3672003381326133,
0.8606640254820307, 4258.283855955983)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 20}, 'score': (0.36710487712169415,
0.8610894763220047, 4258.506419909315)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 10}, 'score': (0.35041818070513703,
0.8615897208514317, 4246.255439297727)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 20}, 'score': (0.35041818070513703,
0.8615897208514317, 4246.255439297727)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 10}, 'score': (0.35971353908934023,
0.8173797205725045, 4195.7811724495095)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 20}, 'score': (0.35971353908934023,
0.8173797205725045, 4195.7811724495095)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 10}, 'score': (0.3620741389656808,
0.82632282549468, 4198.23639189435)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 20}, 'score': (0.3620741389656808,
0.82632282549468, 4198.23639189435)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 10}, 'score': (0.4024973580346682,
0.9384735512109831, 3911.7411719820284)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 20}, 'score': (0.4024973580346682,
0.9384735512109831, 3911.7411719820284)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 10}, 'score': (0.43696184323503295,
0.800807992091704, 4382.46484845105)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 20}, 'score': (0.43670553196911205,
0.8015655276876479, 4382.643219886097)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 10}, 'score': (0.36501089678069776,
0.8390424953834879, 4390.695531815)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 20}, 'score': (0.3650631940423554,
0.8385143445254073, 4390.769580451708)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3672003381326133,
0.8606640254820307, 4258.283855955983)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 6, 'n_init': 20}, 'score': (0.36710487712169415,
0.8610894763220047, 4258.506419909315)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 7, 'n_init': 10}, 'score': (0.35041818070513703,
0.8615897208514317, 4246.255439297727)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 7, 'n_init': 20}, 'score': (0.35041818070513703,
0.8615897208514317, 4246.255439297727)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 8, 'n_init': 10}, 'score': (0.35971353908934023,
0.8173797205725045, 4195.7811724495095)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 8, 'n_init': 20}, 'score': (0.35971353908934023,
0.8173797205725045, 4195.7811724495095)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 9, 'n_init': 10}, 'score': (0.3620741389656808,
0.82632282549468, 4198.23639189435)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 9, 'n_init': 20}, 'score': (0.3620741389656808,
0.82632282549468, 4198.23639189435)}, {'params': {'init': 'random', 'max_iter':

```

```

100, 'n_clusters': 3, 'n_init': 10}, 'score': (0.4024973580346682,
0.9384735512109829, 3911.741171982029)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 3, 'n_init': 20}, 'score': (0.4024973580346682,
0.9384735512109829, 3911.741171982029)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 4, 'n_init': 10}, 'score': (0.43667379498650555,
0.8021676084420011, 4382.62133573297)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 4, 'n_init': 20}, 'score': (0.43671150785107965,
0.8017722125562746, 4382.637282685561)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 5, 'n_init': 10}, 'score': (0.3653927323242895,
0.8373494426318651, 4390.185391321916)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 5, 'n_init': 20}, 'score': (0.3650281322068442,
0.8390619876512234, 4390.681723646808)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3633490931581044,
0.8618224672361846, 4257.726613802008)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 6, 'n_init': 20}, 'score': (0.3633490931581044,
0.8618224672361846, 4257.726613802008)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 7, 'n_init': 10}, 'score': (0.350345362858592,
0.8628326794857513, 4246.46684001746)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 7, 'n_init': 20}, 'score': (0.350345362858592,
0.8628326794857513, 4246.46684001746)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 8, 'n_init': 10}, 'score': (0.35650238704006987,
0.8296564987620684, 4192.372051339794)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 8, 'n_init': 20}, 'score': (0.3609008830375155,
0.8158347602853306, 4195.030786701272)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 9, 'n_init': 10}, 'score': (0.3622686679729114,
0.8252837299284188, 4190.645782182066)}, {'params': {'init': 'random',
'max_iter': 100, 'n_clusters': 9, 'n_init': 20}, 'score': (0.35874525598195856,
0.8314646478711967, 4192.285516370003)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 3, 'n_init': 10}, 'score': (0.4024973580346682,
0.9384735512109829, 3911.741171982029)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 3, 'n_init': 20}, 'score': (0.4024973580346682,
0.9384735512109829, 3911.741171982029)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 4, 'n_init': 10}, 'score': (0.43667379498650555,
0.8021676084420011, 4382.62133573297)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 4, 'n_init': 20}, 'score': (0.43671150785107965,
0.8017722125562746, 4382.637282685561)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 5, 'n_init': 10}, 'score': (0.3653927323242895,
0.8373494426318651, 4390.185391321916)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 5, 'n_init': 20}, 'score': (0.3650281322068442,
0.8390619876512234, 4390.681723646808)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3633490931581044,
0.8618224672361846, 4257.726613802008)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 6, 'n_init': 20}, 'score': (0.3633490931581044,
0.8618224672361846, 4257.726613802008)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 7, 'n_init': 10}, 'score': (0.350345362858592,
0.8628326794857513, 4246.46684001746)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 7, 'n_init': 20}, 'score': (0.350345362858592,
0.8628326794857513, 4246.46684001746)}, {'params': {'init': 'random',

```

```

'max_iter': 300, 'n_clusters': 8, 'n_init': 10}, 'score': (0.35650238704006987,
0.8296564987620684, 4192.372051339794)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 8, 'n_init': 20}, 'score': (0.3609008830375155,
0.8158347602853306, 4195.030786701272)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 10}, 'score': (0.3622686679729114,
0.8252837299284188, 4190.645782182066)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 20}, 'score': (0.35874525598195856,
0.8314646478711967, 4192.285516370003)}], 3: [{'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 3, 'n_init': 10}, 'score': (0.3146723964912577,
1.1673403389199912, 2667.9191772578215)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 3, 'n_init': 20}, 'score': (0.3146723964912577,
1.1673403389199912, 2667.9191772578215)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 10}, 'score': (0.32675022901375905,
1.0221101967742494, 2798.385138654622)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 20}, 'score': (0.3271610330765948,
1.0218235738304982, 2798.425710536509)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 10}, 'score': (0.366651607835907,
0.8983927266111819, 3117.9828186507357)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 20}, 'score': (0.36663709585315823,
0.8984726428922378, 3117.9825532279933)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 10}, 'score': (0.31865162299871197,
1.0219255768415039, 2928.185501330087)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 20}, 'score': (0.31865162299871197,
1.0219255768415039, 2928.185501330087)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 10}, 'score': (0.3391987704106205,
1.0085373212323778, 2820.8667934088962)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 20}, 'score': (0.32493987622758386,
1.014717616528749, 2822.451133709708)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 10}, 'score': (0.3306013807808123,
0.976483558681713, 2758.4690217620987)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 20}, 'score': (0.3306013807808123,
0.976483558681713, 2758.4690217620987)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 10}, 'score': (0.31753891841522963,
1.0272526630694507, 2707.597818681012)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 20}, 'score': (0.31753891841522963,
1.0272526630694507, 2707.597818681012)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 10}, 'score': (0.3146723964912577,
1.1673403389199912, 2667.9191772578215)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 20}, 'score': (0.3146723964912577,
1.1673403389199912, 2667.9191772578215)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 10}, 'score': (0.32675022901375905,
1.0221101967742494, 2798.385138654622)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 20}, 'score': (0.3271610330765948,
1.0218235738304982, 2798.425710536509)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 10}, 'score': (0.366651607835907,
0.8983927266111819, 3117.9828186507357)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 20}, 'score': (0.36663709585315823,
0.8984726428922378, 3117.9825532279933)}, {'params': {'init': 'k-means++',

```



'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.31865162299871197,  
 1.0219255768415039, 2928.185501330087)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.31865162299871197,  
 1.0219255768415039, 2928.185501330087)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.3391987704106205,  
 1.0085373212323778, 2820.8667934088962)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.32493987622758386,  
 1.014717616528749, 2822.451133709708)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.3306013807808123,  
 0.976483558681713, 2758.4690217620987)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.3306013807808123,  
 0.976483558681713, 2758.4690217620987)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.31753891841522963,  
 1.0272526630694507, 2707.597818681012)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.31753891841522963,  
 1.0272526630694507, 2707.597818681012)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.31470871395515626,  
 1.1705942320755616, 2667.559853439986)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.31470871395515626,  
 1.1705942320755616, 2667.559853439986)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.324944668055596,  
 1.025080068789065, 2797.0393973945975)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.32875084476405675,  
 1.0200095349332794, 2797.9836552361435)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.36664815167892184,  
 0.8984393617441452, 3117.9796212053743)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.36664815167892184,  
 0.8984393617441452, 3117.9796212053743)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.3179626999494151,  
 1.0194581781743723, 2927.9472382235135)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.31835343287741014,  
 1.0199424726808808, 2928.07243839394)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.32481064808478616,  
 1.0146897700892454, 2822.390243950722)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.3253804756780898,  
 1.0147367886700278, 2822.4956204370023)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.3308068859788606,  
 0.9768536079545369, 2758.549284249129)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.3308068859788606,  
 0.9768536079545369, 2758.549284249129)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.31757468095390945,  
 1.028281562402153, 2707.6256869230065)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.31757468095390945,  
 1.028281562402153, 2707.6256869230065)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.31470871395515626,  
 1.1705942320755616, 2667.559853439986)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.31470871395515626,  
 1.1705942320755616, 2667.559853439986)}, {'params': {'init': 'random',

```

'max_iter': 300, 'n_clusters': 4, 'n_init': 10}, 'score': (0.324944668055596,
1.025080068789065, 2797.0393973945975)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 4, 'n_init': 20}, 'score': (0.32875084476405675,
1.0200095349332794, 2797.9836552361435)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 5, 'n_init': 10}, 'score': (0.36664815167892184,
0.8984393617441452, 3117.9796212053743)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 5, 'n_init': 20}, 'score': (0.36664815167892184,
0.8984393617441452, 3117.9796212053743)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3179626999494151,
1.0194581781743723, 2927.9472382235135)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 6, 'n_init': 20}, 'score': (0.31835343287741014,
1.0199424726808808, 2928.07243839394)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 7, 'n_init': 10}, 'score': (0.32481064808478616,
1.0146897700892454, 2822.390243950722)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 7, 'n_init': 20}, 'score': (0.3253804756780898,
1.0147367886700278, 2822.4956204370023)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 8, 'n_init': 10}, 'score': (0.3308068859788606,
0.9768536079545369, 2758.549284249129)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 8, 'n_init': 20}, 'score': (0.3308068859788606,
0.9768536079545369, 2758.549284249129)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 10}, 'score': (0.31757468095390945,
1.028281562402153, 2707.6256869230065)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 20}, 'score': (0.31757468095390945,
1.028281562402153, 2707.6256869230065)}], 4: [{'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 3, 'n_init': 10}, 'score': (0.2689085119744404,
1.320727618540326, 2150.354501996648)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 3, 'n_init': 20}, 'score': (0.2689085119744404,
1.320727618540326, 2150.354501996648)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 10}, 'score': (0.2774735529305711,
1.17525318007525, 2134.757962260248)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 4, 'n_init': 20}, 'score': (0.2773768127932139,
1.175483514173317, 2134.7995438088246)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 10}, 'score': (0.3214844571100403,
1.046932789504725, 2309.089442397961)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 20}, 'score': (0.3214844571100403,
1.046932789504725, 2309.089442397961)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 10}, 'score': (0.27105124680596526,
1.1028615165848052, 2200.328653591736)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 20}, 'score': (0.27105124680596526,
1.1028615165848052, 2200.328653591736)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 10}, 'score': (0.2740862144191113,
1.0827736098061733, 2066.722697396891)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 20}, 'score': (0.2740862144191113,
1.0827736098061733, 2066.722697396891)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 10}, 'score': (0.27751851705354663,
1.1149184963460654, 1993.3615793323427)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 20}, 'score': (0.27751851705354663,
1.1149184963460654, 1993.3615793323427)}, {'params': {'init': 'k-means++',

```

'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.27167810425211075,  
 1.1408969278947447, 1920.0477648698834)}, {'params': {'init': 'k-means++',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.27167810425211075,  
 1.1408969278947447, 1920.0477648698834)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.2689085119744404,  
 1.320727618540326, 2150.354501996648)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.2689085119744404,  
 1.320727618540326, 2150.354501996648)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.2774735529305711,  
 1.17525318007525, 2134.757962260248)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.2773768127932139,  
 1.175483514173317, 2134.7995438088246)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.3214844571100403,  
 1.046932789504725, 2309.089442397961)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.3214844571100403,  
 1.046932789504725, 2309.089442397961)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.27105124680596526,  
 1.1028615165848052, 2200.328653591736)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.27105124680596526,  
 1.1028615165848052, 2200.328653591736)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.2740862144191113,  
 1.0827736098061733, 2066.722697396891)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.2740862144191113,  
 1.0827736098061733, 2066.722697396891)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.27751851705354663,  
 1.1149184963460654, 1993.3615793323427)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.27751851705354663,  
 1.1149184963460654, 1993.3615793323427)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.27167810425211075,  
 1.1408969278947447, 1920.0477648698834)}, {'params': {'init': 'k-means++',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.27167810425211075,  
 1.1408969278947447, 1920.0477648698834)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.2689232530367428,  
 1.3207697658845652, 2150.354787475825)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.2689232530367428,  
 1.3207697658845652, 2150.354787475825)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.27521952924028126,  
 1.1796111048052484, 2135.033736632836)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.27521952924028126,  
 1.1796111048052484, 2135.033736632836)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.3203944601559695,  
 1.0493745531896788, 2309.0860532938595)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.3203944601559695,  
 1.0493745531896788, 2309.0860532938595)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.2709313525304655,  
 1.1035115154597548, 2200.3388827300614)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.2709313525304655,  
 1.1035115154597548, 2200.3388827300614)}, {'params': {'init': 'random',

'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.27343953738760374,  
 1.0828377740716335, 2066.2388126433225)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.2740862144191113,  
 1.0827736098061733, 2066.7226973968905)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.28243149479805296,  
 1.1533904226948744, 1974.4708816343182)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.27682909695695757,  
 1.1144692820235584, 1993.2399305707684)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.2712187857077789,  
 1.1420430406866675, 1920.3354699273646)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.2712187857077789,  
 1.1420430406866675, 1920.3354699273646)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.2689232530367428,  
 1.3207697658845652, 2150.354787475825)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.2689232530367428,  
 1.3207697658845652, 2150.354787475825)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.27521952924028126,  
 1.1796111048052484, 2135.033736632836)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.27521952924028126,  
 1.1796111048052484, 2135.033736632836)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.3203944601559695,  
 1.0493745531896788, 2309.0860532938595)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.3203944601559695,  
 1.0493745531896788, 2309.0860532938595)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.2709313525304655,  
 1.1035115154597548, 2200.3388827300614)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.2709313525304655,  
 1.1035115154597548, 2200.3388827300614)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.27343953738760374,  
 1.0828377740716335, 2066.2388126433225)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.2740862144191113,  
 1.0827736098061733, 2066.7226973968905)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.28243149479805296,  
 1.1533904226948744, 1974.4708816343182)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.27682909695695757,  
 1.1144692820235584, 1993.2399305707684)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.2712187857077789,  
 1.1420430406866675, 1920.3354699273646)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.2712187857077789,  
 1.1420430406866675, 1920.3354699273646)}], 5: [{'params': {'init': 'k-means++',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.24037802041160808,  
 1.4574170780114333, 1813.216722398052)}, {'params': {'init': 'k-means++',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.23968190613932766,  
 1.4567887235554553, 1813.2264263733598)}, {'params': {'init': 'k-means++',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.2428860675196842,  
 1.3236801035274266, 1741.030795655103)}, {'params': {'init': 'k-means++',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.29814639005934335,  
 1.2248032014422265, 1779.6261102371404)}, {'params': {'init': 'k-means++',

```

'max_iter': 100, 'n_clusters': 5, 'n_init': 10}, 'score': (0.2896384820482563,
1.1960287811729469, 1841.7540210856853)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 5, 'n_init': 20}, 'score': (0.29001112132136797,
1.1956591016733342, 1841.7491504823342)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3134107587579036,
1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 6, 'n_init': 20}, 'score': (0.3134107587579036,
1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 10}, 'score': (0.2763060598138407,
1.1265497041593515, 1751.7409311747224)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 7, 'n_init': 20}, 'score': (0.2763060598138407,
1.1265497041593515, 1751.7409311747224)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 10}, 'score': (0.2760822543710182,
1.1945360352586754, 1671.3681818121393)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 8, 'n_init': 20}, 'score': (0.2760822543710182,
1.1945360352586754, 1671.3681818121393)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 10}, 'score': (0.28024406404359314,
1.195777274435692, 1613.716371188193)}, {'params': {'init': 'k-means++',
'max_iter': 100, 'n_clusters': 9, 'n_init': 20}, 'score': (0.2808131227234657,
1.1960959244494975, 1613.7633982546754)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 10}, 'score': (0.24037802041160808,
1.4574170780114333, 1813.216722398052)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 3, 'n_init': 20}, 'score': (0.23968190613932766,
1.4567887235554553, 1813.2264263733598)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 10}, 'score': (0.2428860675196842,
1.3236801035274266, 1741.030795655103)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 4, 'n_init': 20}, 'score': (0.29814639005934335,
1.2248032014422265, 1779.6261102371404)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 10}, 'score': (0.2896384820482563,
1.1960287811729469, 1841.7540210856853)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 5, 'n_init': 20}, 'score': (0.29001112132136797,
1.1956591016733342, 1841.7491504823342)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 6, 'n_init': 10}, 'score': (0.3134107587579036,
1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 6, 'n_init': 20}, 'score': (0.3134107587579036,
1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 7, 'n_init': 10}, 'score': (0.2763060598138407,
1.1265497041593515, 1751.7409311747224)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 7, 'n_init': 20}, 'score': (0.2763060598138407,
1.1265497041593515, 1751.7409311747224)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 8, 'n_init': 10}, 'score': (0.2760822543710182,
1.1945360352586754, 1671.3681818121393)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 8, 'n_init': 20}, 'score': (0.2760822543710182,
1.1945360352586754, 1671.3681818121393)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 9, 'n_init': 10}, 'score': (0.28024406404359314,
1.195777274435692, 1613.716371188193)}, {'params': {'init': 'k-means++',
'max_iter': 300, 'n_clusters': 9, 'n_init': 20}, 'score': (0.2808131227234657,
1.1960959244494975, 1613.7633982546754)}, {'params': {'init': 'random',

```

'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.23965424726587725,  
 1.4566873281175547, 1813.223433327099)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.24013630075022047,  
 1.4571565486192462, 1813.2314488625311)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.29806285367903657,  
 1.2247211393421311, 1779.626555420164)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.29806285367903657,  
 1.2247211393421311, 1779.626555420164)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.2894551466275514,  
 1.1962806457249855, 1841.7133886329582)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.28981444017333197,  
 1.196044207283825, 1841.7451877584235)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.3134107587579036,  
 1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.3134107587579036,  
 1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.2762763252778659,  
 1.127684008889972, 1751.7429881356268)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.2762763252778659,  
 1.127684008889972, 1751.7429881356268)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 10}, 'score': (0.27361911389226146,  
 1.1875370785557804, 1671.2462130963318)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 8, 'n\_init': 20}, 'score': (0.2761884411555232,  
 1.1965163785317408, 1671.320851868272)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 10}, 'score': (0.2807528194014574,  
 1.196948473982624, 1613.7767720363036)}, {'params': {'init': 'random',  
 'max\_iter': 100, 'n\_clusters': 9, 'n\_init': 20}, 'score': (0.2807528194014574,  
 1.196948473982624, 1613.7767720363036)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 10}, 'score': (0.23965424726587725,  
 1.4566873281175547, 1813.223433327099)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 3, 'n\_init': 20}, 'score': (0.24013630075022047,  
 1.4571565486192462, 1813.2314488625311)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 10}, 'score': (0.29806285367903657,  
 1.2247211393421311, 1779.626555420164)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 4, 'n\_init': 20}, 'score': (0.29806285367903657,  
 1.2247211393421311, 1779.626555420164)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 10}, 'score': (0.2894551466275514,  
 1.1962806457249855, 1841.7133886329582)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 5, 'n\_init': 20}, 'score': (0.28981444017333197,  
 1.196044207283825, 1841.7451877584235)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 10}, 'score': (0.3134107587579036,  
 1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 6, 'n\_init': 20}, 'score': (0.3134107587579036,  
 1.1027545929089277, 1807.1647964033464)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 10}, 'score': (0.2762763252778659,  
 1.127684008889972, 1751.7429881356268)}, {'params': {'init': 'random',  
 'max\_iter': 300, 'n\_clusters': 7, 'n\_init': 20}, 'score': (0.2762763252778659,  
 1.127684008889972, 1751.7429881356268)}, {'params': {'init': 'random',

```

'max_iter': 300, 'n_clusters': 8, 'n_init': 10}, 'score': (0.27361911389226146,
1.1875370785557804, 1671.2462130963318)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 8, 'n_init': 20}, 'score': (0.2761884411555232,
1.1965163785317408, 1671.320851868272)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 10}, 'score': (0.2807528194014574,
1.196948473982624, 1613.7767720363036)}, {'params': {'init': 'random',
'max_iter': 300, 'n_clusters': 9, 'n_init': 20}, 'score': (0.2807528194014574,
1.196948473982624, 1613.7767720363036)}}], 'Agglomerative': {2: [{'params':
{'linkage': 'ward', 'n_clusters': 3}, 'score': (0.39587370471001304,
1.0227350248178448, 3314.1425025851768)}, {'params': {'linkage': 'ward',
'n_clusters': 4}, 'score': (0.406964452256109, 0.802978038805655,
3651.3797615710223)}, {'params': {'linkage': 'ward', 'n_clusters': 5}, 'score':
(0.2987079272790559, 0.9640290817731934, 3614.1656768502594)}, {'params':
{'linkage': 'ward', 'n_clusters': 6}, 'score': (0.2803993840382627,
0.9120157438951967, 3409.7487721036546)}, {'params': {'linkage': 'ward',
'n_clusters': 7}, 'score': (0.28374591091619283, 0.9709643411843917,
3341.5751905351613)}, {'params': {'linkage': 'ward', 'n_clusters': 8}, 'score':
(0.2829868392356227, 0.9949983482981348, 3406.677814607097)}, {'params':
{'linkage': 'ward', 'n_clusters': 9}, 'score': (0.2994879624548921,
0.9340271054738712, 3470.531866047693)}, {'params': {'linkage': 'complete',
'n_clusters': 3}, 'score': (0.48432280324659793, 0.7566003571698333,
2556.5216910510767)}, {'params': {'linkage': 'complete', 'n_clusters': 4},
'score': (0.48151750861003395, 0.6113644237862588, 1723.2972496089199)},
{'params': {'linkage': 'complete', 'n_clusters': 5}, 'score':
(0.42895262115822863, 0.6342974764054585, 1729.0561369481343)}, {'params':
{'linkage': 'complete', 'n_clusters': 6}, 'score': (0.40906965218156527,
0.6423695104160392, 1510.005631672537)}, {'params': {'linkage': 'complete',
'n_clusters': 7}, 'score': (0.39469690538127333, 0.6406089043227502,
1304.1933297635487)}, {'params': {'linkage': 'complete', 'n_clusters': 8},
'score': (0.3911037268309153, 0.6800313592671685, 1128.6391022141493)},
{'params': {'linkage': 'complete', 'n_clusters': 9}, 'score':
(0.2617909969427613, 0.8141945638930166, 1645.2514013033935)}, {'params':
{'linkage': 'average', 'n_clusters': 3}, 'score': (0.6075651987527403,
0.3794640915932333, 140.65063737202522)}, {'params': {'linkage': 'average',
'n_clusters': 4}, 'score': (0.5713579268823568, 0.4794482245402226,
1224.3934191535716)}, {'params': {'linkage': 'average', 'n_clusters': 5},
'score': (0.558573609198812, 0.5962922558259496, 965.4148220240168)}, {'params':
{'linkage': 'average', 'n_clusters': 6}, 'score': (0.5528146082933071,
0.5396017243354762, 777.8616720934473)}, {'params': {'linkage': 'average',
'n_clusters': 7}, 'score': (0.46599002500371045, 0.5698543565321962,
740.7720129096535)}, {'params': {'linkage': 'average', 'n_clusters': 8},
'score': (0.37482638982939925, 0.7001836308889408, 1166.40892140422)},
{'params': {'linkage': 'average', 'n_clusters': 9}, 'score':
(0.37269178226300176, 0.6974437965576766, 1038.0313618043128)}, {'params':
{'linkage': 'single', 'n_clusters': 3}, 'score': (0.7144848175386341,
0.16920588504384718, 24.292275620344395)}, {'params': {'linkage': 'single',
'n_clusters': 4}, 'score': (0.6865753720312757, 0.2591891161343916,
27.116797743850686)}, {'params': {'linkage': 'single', 'n_clusters': 5},

```

'score': (0.6544502695991152, 0.24981534349873571, 23.96712018652606)},  
{'params': {'linkage': 'single', 'n\_clusters': 6}, 'score': (0.6452873401391809,  
0.19238894769107964, 19.20742101156293)}, {'params': {'linkage': 'single',  
'n\_clusters': 7}, 'score': (0.593777862587913, 0.21006764269419934,  
17.5664639553682)}, {'params': {'linkage': 'single', 'n\_clusters': 8}, 'score':  
(0.5612683737491476, 0.22271107117992442, 16.1872381819564)}, {'params':  
{'linkage': 'single', 'n\_clusters': 9}, 'score': (0.5607874972521522,  
0.22732098189339217, 15.333990349057395)}], 3: [{'params': {'linkage': 'ward',  
'n\_clusters': 3}, 'score': (0.307570903425625, 1.1873680918814722,  
2491.6544202056716)}, {'params': {'linkage': 'ward', 'n\_clusters': 4}, 'score':  
(0.3014982942358188, 1.0415145229185447, 2510.229673593349)}, {'params':  
{'linkage': 'ward', 'n\_clusters': 5}, 'score': (0.3290551670059458,  
0.9778143488133452, 2746.868117304966)}, {'params': {'linkage': 'ward',  
'n\_clusters': 6}, 'score': (0.32794784056416154, 1.0511133968674764,  
2597.7221156061414)}, {'params': {'linkage': 'ward', 'n\_clusters': 7}, 'score':  
(0.28190873575070935, 1.1412218498853621, 2446.3128985660437)}, {'params':  
{'linkage': 'ward', 'n\_clusters': 8}, 'score': (0.28465174841660834,  
1.0837149463912854, 2362.0439880976537)}, {'params': {'linkage': 'ward',  
'n\_clusters': 9}, 'score': (0.28278343004218415, 1.1444332904667844,  
2294.898353060376)}, {'params': {'linkage': 'complete', 'n\_clusters': 3},  
'score': (0.5846994901828863, 0.4882959641891609, 765.4115797364749)},  
{'params': {'linkage': 'complete', 'n\_clusters': 4}, 'score':  
(0.44013834614560127, 0.8676148580188907, 1125.184666906897)}, {'params':  
{'linkage': 'complete', 'n\_clusters': 5}, 'score': (0.3306371276809357,  
0.9638526504311449, 1237.7109611859119)}, {'params': {'linkage': 'complete',  
'n\_clusters': 6}, 'score': (0.3168004196558596, 1.005289337757399,  
1095.4555655181716)}, {'params': {'linkage': 'complete', 'n\_clusters': 7},  
'score': (0.3159527357204902, 0.8913614827969729, 917.444926120594)}, {'params':  
{'linkage': 'complete', 'n\_clusters': 8}, 'score': (0.23773083686168742,  
0.9942854957837757, 1213.2595679094582)}, {'params': {'linkage': 'complete',  
'n\_clusters': 9}, 'score': (0.23518604126127537, 1.03223640498443,  
1074.6963020556757)}, {'params': {'linkage': 'average', 'n\_clusters': 3},  
'score': (0.6635462363651364, 0.3484533455839458, 43.25363759335528)},  
{'params': {'linkage': 'average', 'n\_clusters': 4}, 'score':  
(0.4939719331988341, 0.6318657120947816, 813.898318225025)}, {'params':  
{'linkage': 'average', 'n\_clusters': 5}, 'score': (0.48449330001481267,  
0.6908006174904181, 1029.3482711312793)}, {'params': {'linkage': 'average',  
'n\_clusters': 6}, 'score': (0.4689998210096886, 0.7441602547642648,  
864.2946022472306)}, {'params': {'linkage': 'average', 'n\_clusters': 7},  
'score': (0.42136262142005115, 0.770492081767818, 829.1779965701951)},  
{'params': {'linkage': 'average', 'n\_clusters': 8}, 'score':  
(0.4202066700550922, 0.6712937306478028, 711.2402560174874)}, {'params':  
{'linkage': 'average', 'n\_clusters': 9}, 'score': (0.40422339061035684,  
0.7057509706046576, 637.2887609713332)}, {'params': {'linkage': 'single',  
'n\_clusters': 3}, 'score': (0.5647307646900226, 0.2613625532566788,  
17.144211901480823)}, {'params': {'linkage': 'single', 'n\_clusters': 4},  
'score': (0.5633872361094945, 0.24936042138218187, 16.65392105303912)},  
{'params': {'linkage': 'single', 'n\_clusters': 5}, 'score': (0.5557148385706466,



0.27241344108165855, 23.82406433841305)}, {'params': {'linkage': 'single', 'n\_clusters': 6}, 'score': (0.5554600279569282, 0.2691881398948368, 21.320278115046914)}, {'params': {'linkage': 'single', 'n\_clusters': 7}, 'score': (0.554472641837936, 0.28346770181372405, 20.295045847477752)}, {'params': {'linkage': 'single', 'n\_clusters': 8}, 'score': (0.5533360286675926, 0.2656921989261483, 17.473696738883707)}, {'params': {'linkage': 'single', 'n\_clusters': 9}, 'score': (0.5431731647845863, 0.27796711989384837, 17.56153394301605)}], 4: [{'params': {'linkage': 'ward', 'n\_clusters': 3}, 'score': (0.2620123746028213, 1.1515725534580563, 1755.622413885215)}, {'params': {'linkage': 'ward', 'n\_clusters': 4}, 'score': (0.29169938253257977, 1.032509861292593, 1759.3033715336621)}, {'params': {'linkage': 'ward', 'n\_clusters': 5}, 'score': (0.28397776232477595, 1.0723737563638651, 1788.0367707728437)}, {'params': {'linkage': 'ward', 'n\_clusters': 6}, 'score': (0.23288867404226882, 1.2705525753488118, 1710.3325651036757)}, {'params': {'linkage': 'ward', 'n\_clusters': 7}, 'score': (0.21645744478251344, 1.2188684569272972, 1676.6957856793135)}, {'params': {'linkage': 'ward', 'n\_clusters': 8}, 'score': (0.21527781821393402, 1.1633827972330768, 1643.7819401923466)}, {'params': {'linkage': 'ward', 'n\_clusters': 9}, 'score': (0.223182755128608, 1.2491604197993158, 1599.7454991297052)}, {'params': {'linkage': 'complete', 'n\_clusters': 3}, 'score': (0.43748731220821835, 0.8756802979028583, 1581.8990251012124)}, {'params': {'linkage': 'complete', 'n\_clusters': 4}, 'score': (0.4318477315238128, 0.792953613267751, 1068.0318168343138)}, {'params': {'linkage': 'complete', 'n\_clusters': 5}, 'score': (0.3899735228454676, 0.8810535887910369, 919.4340813587435)}, {'params': {'linkage': 'complete', 'n\_clusters': 6}, 'score': (0.37513126016598, 0.9632127923034424, 773.5069053446681)}, {'params': {'linkage': 'complete', 'n\_clusters': 7}, 'score': (0.36284972159564743, 0.9794742952538763, 658.0980740764417)}, {'params': {'linkage': 'complete', 'n\_clusters': 8}, 'score': (0.18050345489810773, 1.1930474955087709, 784.1763741549098)}, {'params': {'linkage': 'complete', 'n\_clusters': 9}, 'score': (0.18049370647279844, 1.215258391215838, 820.1863532113184)}, {'params': {'linkage': 'average', 'n\_clusters': 3}, 'score': (0.5603066650263009, 0.5393860860911791, 111.7190529379311)}, {'params': {'linkage': 'average', 'n\_clusters': 4}, 'score': (0.5403702142905685, 0.5552060460014795, 620.7044994162085)}, {'params': {'linkage': 'average', 'n\_clusters': 5}, 'score': (0.5307512253656957, 0.5075256794770655, 467.4932287264358)}, {'params': {'linkage': 'average', 'n\_clusters': 6}, 'score': (0.5198407607461262, 0.7714741869443427, 382.35932848196114)}, {'params': {'linkage': 'average', 'n\_clusters': 7}, 'score': (0.41385940777157765, 0.781111832737608, 320.5755816689694)}, {'params': {'linkage': 'average', 'n\_clusters': 8}, 'score': (0.4080479001895096, 0.7522122778389133, 275.9279434850206)}, {'params': {'linkage': 'average', 'n\_clusters': 9}, 'score': (0.3982233990552501, 0.7947307158542731, 250.28611670912878)}, {'params': {'linkage': 'single', 'n\_clusters': 3}, 'score': (0.5373168712700958, 0.28017527171233575, 15.666393846740638)}, {'params': {'linkage': 'single', 'n\_clusters': 4}, 'score': (0.5360808701263778, 0.267878309991624, 15.11756812902321)}, {'params': {'linkage': 'single', 'n\_clusters': 5}, 'score': (0.5346885029031482, 0.2692262881907833, 13.826384760189633)}, {'params':

```
{'linkage': 'single', 'n_clusters': 6}, 'score': (0.5306633801362435,
0.2698683250354302, 13.076403915853248)}, {'params': {'linkage': 'single',
'n_clusters': 7}, 'score': (0.5269856005405307, 0.28817370316644014,
17.768381577709654)}, {'params': {'linkage': 'single', 'n_clusters': 8},
'score': (0.5186288315717092, 0.30579303292697846, 17.620694691442043)},
{'params': {'linkage': 'single', 'n_clusters': 9}, 'score': (0.5187159717768017,
0.30688859209521435, 17.163568131208255)}], 5: [{'params': {'linkage': 'ward',
'n_clusters': 3}, 'score': (0.22762027868253887, 1.5834037637479337,
1443.115019068764)}, {'params': {'linkage': 'ward', 'n_clusters': 4}, 'score':
(0.23010720458469078, 1.381986860044806, 1396.4709494023657)}, {'params':
{'linkage': 'ward', 'n_clusters': 5}, 'score': (0.2492535683252509,
1.297390607660709, 1492.6284501563575)}, {'params': {'linkage': 'ward',
'n_clusters': 6}, 'score': (0.26073513872522125, 1.188005355190138,
1415.3832540840285)}, {'params': {'linkage': 'ward', 'n_clusters': 7}, 'score':
(0.21014904241901833, 1.228499410023195, 1392.4090676350104)}, {'params':
{'linkage': 'ward', 'n_clusters': 8}, 'score': (0.21423363602344006,
1.2530259960664252, 1353.406171285247)}, {'params': {'linkage': 'ward',
'n_clusters': 9}, 'score': (0.22563773230610895, 1.3546672965387971,
1311.6604963491372)}, {'params': {'linkage': 'complete', 'n_clusters': 3},
'score': (0.5330775927422162, 0.6377724941828203, 687.0190692922246)},
{'params': {'linkage': 'complete', 'n_clusters': 4}, 'score':
(0.5217172872519237, 0.96699412207622, 478.27224109679673)}, {'params':
{'linkage': 'complete', 'n_clusters': 5}, 'score': (0.2583243017089808,
1.2022461252582914, 594.7661653462397)}, {'params': {'linkage': 'complete',
'n_clusters': 6}, 'score': (0.2668634198004871, 1.1800700414432705,
889.9962269215669)}, {'params': {'linkage': 'complete', 'n_clusters': 7},
'score': (0.2658171784738916, 1.0475422548148015, 744.94940104482)}, {'params':
{'linkage': 'complete', 'n_clusters': 8}, 'score': (0.2555178456533447,
1.1248163961062705, 654.3626144789207)}, {'params': {'linkage': 'complete',
'n_clusters': 9}, 'score': (0.25462574023177187, 1.1118077537232325,
576.6869834299181)}, {'params': {'linkage': 'average', 'n_clusters': 3},
'score': (0.5439686012119248, 0.5130424044795684, 622.1196301171643)},
{'params': {'linkage': 'average', 'n_clusters': 4}, 'score':
(0.5320927103291411, 0.5485607503534613, 419.5267490216868)}, {'params':
{'linkage': 'average', 'n_clusters': 5}, 'score': (0.41585991284437784,
0.6649761818244174, 512.1806445744346)}, {'params': {'linkage': 'average',
'n_clusters': 6}, 'score': (0.4107623763405669, 0.8802755782727868,
415.0233542808496)}, {'params': {'linkage': 'average', 'n_clusters': 7},
'score': (0.3482250040413306, 1.0036465419564489, 539.1462977524179)},
{'params': {'linkage': 'average', 'n_clusters': 8}, 'score':
(0.3465856582460124, 1.0318098946797827, 464.7377337695666)}, {'params':
{'linkage': 'average', 'n_clusters': 9}, 'score': (0.32961423095163445,
1.0807150708970064, 428.5327767360073)}, {'params': {'linkage': 'single',
'n_clusters': 3}, 'score': (0.5116165643436573, 0.29760744353669916,
14.867148637751187)}, {'params': {'linkage': 'single', 'n_clusters': 4},
'score': (0.49641791615642905, 0.3033179278153528, 12.441155138022612)},
{'params': {'linkage': 'single', 'n_clusters': 5}, 'score':
(0.49279463291126396, 0.29634214384118823, 12.03232841400174)}, {'params':
```

```
{'linkage': 'single', 'n_clusters': 6}, 'score': (0.4929838404867554,
0.29567691955581893, 11.451139440233947)}, {'params': {'linkage': 'single',
'n_clusters': 7}, 'score': (0.49215032700930916, 0.3011224015568771,
12.700648408299148)}, {'params': {'linkage': 'single', 'n_clusters': 8},
'score': (0.4908047737923336, 0.29407518876607097, 12.748995034013578)},
{'params': {'linkage': 'single', 'n_clusters': 9}, 'score': (0.3381961010570384,
0.3290064506618544, 11.581749700476196)}}], 'DBSCAN': {2: [{'params': {'eps':
0.5, 'min_samples': 3}, 'score': (0.3240172644151518, 1.4585195895095129,
70.06875238137512)}, {'params': {'eps': 0.5, 'min_samples': 5}, 'score':
(0.3184295420144574, 3.6073598771408912, 122.5941364005826)}, {'params': {'eps':
0.5, 'min_samples': 10}, 'score': (0.3884607849404637, 0.9540137213234943,
542.7707962325118)}, {'params': {'eps': 0.5, 'min_samples': 15}, 'score':
(0.46333163987176096, 1.4460277501868042, 1066.951440907399)}, {'params':
{'eps': 1.0, 'min_samples': 3}, 'score': (0.6357523600838892,
0.8681154614944683, 73.15381886903455)}, {'params': {'eps': 1.0, 'min_samples':
5}, 'score': (0.5649030622483121, 1.7267752554085418, 104.9755443473632)},
{'params': {'eps': 1.0, 'min_samples': 10}, 'score': (0.6564463512071658,
1.008975325105257, 458.54498867194906)}, {'params': {'eps': 1.0, 'min_samples':
15}, 'score': (0.6438013923553175, 1.1049907829314387, 595.0968540576275)},
{'params': {'eps': 1.5, 'min_samples': 3}, 'score': (0.7074571165276579,
0.667399273190541, 85.1900225647041)}, {'params': {'eps': 1.5, 'min_samples':
5}, 'score': (0.708145297890045, 0.7230745322550047, 93.99532346511484)},
{'params': {'eps': 1.5, 'min_samples': 10}, 'score': (0.6550437343022281,
0.7226360219652678, 184.9401369872646)}, {'params': {'eps': 1.5, 'min_samples':
15}, 'score': (0.6668383411070625, 0.9605552483875771, 299.697594401739)},
{'params': {'eps': 2.0, 'min_samples': 3}, 'score': (0.7214246011996667,
0.6788983701172336, 76.81002251018943)}, {'params': {'eps': 2.0, 'min_samples':
5}, 'score': (0.7214246011996667, 0.6788983701172336, 76.81002251018943)},
{'params': {'eps': 2.0, 'min_samples': 10}, 'score': (0.7200098884776146,
0.7556012718861121, 85.13614765868454)}, {'params': {'eps': 2.0, 'min_samples':
15}, 'score': (0.6909472823190721, 0.7163470877488993, 119.63419890431193)}}], 3:
[{'params': {'eps': 0.5, 'min_samples': 3}, 'score': (-0.20508010180561637,
1.7903798049155706, 45.37092254738408)}, {'params': {'eps': 0.5, 'min_samples':
5}, 'score': (-0.2565940595651856, 1.497881220903671, 124.26357107513307)},
{'params': {'eps': 0.5, 'min_samples': 10}, 'score': (0.10387290902939897,
1.67242407109763, 631.6412768736878)}, {'params': {'eps': 0.5, 'min_samples':
15}, 'score': (0.07797023121329028, 1.5705625799440714, 869.9952513372999)},
{'params': {'eps': 1.0, 'min_samples': 3}, 'score': (0.13265755595829332,
1.306913010621511, 78.27514931870041)}, {'params': {'eps': 1.0, 'min_samples':
5}, 'score': (0.3936785303860722, 2.2188530402611044, 251.75937195906295)},
{'params': {'eps': 1.0, 'min_samples': 10}, 'score': (0.40653404385499103,
1.2665081168792065, 519.9851103437114)}, {'params': {'eps': 1.0, 'min_samples':
15}, 'score': (0.5092537042902239, 1.1858045786074798, 859.127132457352)},
{'params': {'eps': 1.5, 'min_samples': 3}, 'score': (0.5887885042808141,
0.8689612808348304, 82.07438026574842)}, {'params': {'eps': 1.5, 'min_samples':
5}, 'score': (0.5365854354216303, 2.895124705701278, 164.71643784767483)},
{'params': {'eps': 1.5, 'min_samples': 10}, 'score': (0.611821603969022,
1.0982436782768201, 486.9909701619563)}, {'params': {'eps': 1.5, 'min_samples':
```

15}, 'score': (0.6087747023149197, 1.1209135692616858, 557.1461223278501)),  
 {'params': {'eps': 2.0, 'min\_samples': 3}, 'score': (0.6499379557576712,  
 0.8027415335929355, 108.28755287798843)}, {'params': {'eps': 2.0, 'min\_samples':  
 5}, 'score': (0.638231726210603, 0.981274861064108, 119.30279969045998)},  
 {'params': {'eps': 2.0, 'min\_samples': 10}, 'score': (0.6105674130585514,  
 0.8732668468816362, 249.98449935596457)}, {'params': {'eps': 2.0, 'min\_samples':  
 15}, 'score': (0.6239913183314697, 1.0337372632022162, 320.78637138842123)}], 4:  
 [{'params': {'eps': 0.5, 'min\_samples': 3}, 'score': (-0.375832252619525,  
 1.4684013909455302, 29.424966185093762)}, {'params': {'eps': 0.5, 'min\_samples':  
 5}, 'score': (-0.31906601132882706, 1.5856802834024988, 62.71768599650236)},  
 {'params': {'eps': 0.5, 'min\_samples': 10}, 'score': (-0.2035694974234666,  
 1.2394473096879137, 173.08751769181643)}, {'params': {'eps': 0.5, 'min\_samples':  
 15}, 'score': (-0.20195908368942134, 1.6704871716435672, 216.37727188990087)},  
 {'params': {'eps': 1.0, 'min\_samples': 3}, 'score': (0.00953278625317219,  
 1.6098848544362683, 96.8323280414639)}, {'params': {'eps': 1.0, 'min\_samples':  
 5}, 'score': (0.22345837599846746, 1.9607549011147798, 316.2393552972082)},  
 {'params': {'eps': 1.0, 'min\_samples': 10}, 'score': (0.23328908736934376,  
 1.4879616153864152, 414.1697319610226)}, {'params': {'eps': 1.0, 'min\_samples':  
 15}, 'score': (0.4077331540626646, 1.614462595666646, 1416.1322401737048)},  
 {'params': {'eps': 1.5, 'min\_samples': 3}, 'score': (0.3471466717419147,  
 1.4305349924548107, 79.27586716748213)}, {'params': {'eps': 1.5, 'min\_samples':  
 5}, 'score': (0.4898723689674858, 1.212214445601911, 243.56825510215697)},  
 {'params': {'eps': 1.5, 'min\_samples': 10}, 'score': (0.5381344324237283,  
 1.3374650718277439, 629.4008711766946)}, {'params': {'eps': 1.5, 'min\_samples':  
 15}, 'score': (0.5144748767187549, 1.3605600441935433, 749.6609402509467)},  
 {'params': {'eps': 2.0, 'min\_samples': 3}, 'score': (0.4566621225945644,  
 1.0586201622378435, 60.70181011169553)}, {'params': {'eps': 2.0, 'min\_samples':  
 5}, 'score': (0.49286237993610904, 2.9347552771792755, 141.64840751873362)},  
 {'params': {'eps': 2.0, 'min\_samples': 10}, 'score': (0.5750382592378072,  
 1.2193981937577978, 363.17209644676257)}, {'params': {'eps': 2.0, 'min\_samples':  
 15}, 'score': (0.5769957279938744, 1.2228779519639947, 449.7263131261176)}], 5:  
 [{'params': {'eps': 0.5, 'min\_samples': 3}, 'score': (-0.31442378846136815,  
 1.4569848662622042, 25.67901560402031)}, {'params': {'eps': 0.5, 'min\_samples':  
 5}, 'score': (-0.30990953157190315, 1.466675174142465, 46.17362589974499)},  
 {'params': {'eps': 0.5, 'min\_samples': 10}, 'score': (-0.303433327884952,  
 1.6169786697319943, 76.2535547146881)}, {'params': {'eps': 0.5, 'min\_samples':  
 15}, 'score': (-0.2328462964696697, 1.5742189825118496, 150.3177616190073)},  
 {'params': {'eps': 1.0, 'min\_samples': 3}, 'score': (-0.22782004140908096,  
 1.6354133701044813, 57.157928837957385)}, {'params': {'eps': 1.0, 'min\_samples':  
 5}, 'score': (-0.03800183329245565, 1.5719286092333205, 184.59891209544202)},  
 {'params': {'eps': 1.0, 'min\_samples': 10}, 'score': (-0.06538620575170774,  
 1.568172224760517, 214.02482759625917)}, {'params': {'eps': 1.0, 'min\_samples':  
 15}, 'score': (-0.05459215538951181, 1.4571753848307138, 256.7682247006956)},  
 {'params': {'eps': 1.5, 'min\_samples': 3}, 'score': (-0.022484413665163464,  
 1.1187547190841927, 64.373114232658)}, {'params': {'eps': 1.5, 'min\_samples':  
 5}, 'score': (0.26863722394594663, 2.187014681000047, 228.72889906126488)},  
 {'params': {'eps': 1.5, 'min\_samples': 10}, 'score': (0.44468396678785704,  
 1.541398190647205, 867.7118352692869)}, {'params': {'eps': 1.5, 'min\_samples':

```

15}, 'score': (0.4094984783263266, 1.3820641248016352, 668.025973853923)},
{'params': {'eps': 2.0, 'min_samples': 3}, 'score': (0.33056568257243946,
1.6561324959105928, 77.64584206157551)}, {'params': {'eps': 2.0, 'min_samples':
5}, 'score': (0.4444974110025688, 1.8553784290782516, 130.19774174476183)},
{'params': {'eps': 2.0, 'min_samples': 10}, 'score': (0.4770773378179487,
1.3956803178689412, 259.75124923583206)}, {'params': {'eps': 2.0, 'min_samples':
15}, 'score': (0.5072267785931093, 1.44252616408996, 561.4174235658277)}}],
'GMM': {2: [{'params': {'covariance_type': 'full', 'n_components': 3,
'random_state': 42}, 'score': (0.39095456562326947, 0.9881910081535367,
3551.2784668189456)}, {'params': {'covariance_type': 'full', 'n_components': 4,
'random_state': 42}, 'score': (0.40416372221730473, 0.8267834137742408,
4118.365200894689)}, {'params': {'covariance_type': 'full', 'n_components': 5,
'random_state': 42}, 'score': (0.32658999592457716, 0.915721292772387,
3990.385507001125)}, {'params': {'covariance_type': 'full', 'n_components': 6,
'random_state': 42}, 'score': (0.3109387504147275, 0.99418089657051,
3602.881725014636)}, {'params': {'covariance_type': 'full', 'n_components': 7,
'random_state': 42}, 'score': (0.26757867653551864, 1.0292806569596846,
3338.8162958553244)}, {'params': {'covariance_type': 'full', 'n_components': 8,
'random_state': 42}, 'score': (0.31291308546989494, 0.8274338167851039,
3363.5142091726498)}, {'params': {'covariance_type': 'full', 'n_components': 9,
'random_state': 42}, 'score': (0.33864559072592754, 0.8549330123896448,
3824.551600413693)}, {'params': {'covariance_type': 'tied', 'n_components': 3,
'random_state': 42}, 'score': (0.40833591418506837, 0.8439603213532516,
2883.9661222054838)}, {'params': {'covariance_type': 'tied', 'n_components': 4,
'random_state': 42}, 'score': (0.4375197819795308, 0.7369848464880445,
3188.3251681475836)}, {'params': {'covariance_type': 'tied', 'n_components': 5,
'random_state': 42}, 'score': (0.36494404321469026, 0.836439942552966,
4047.185757369932)}, {'params': {'covariance_type': 'tied', 'n_components': 6,
'random_state': 42}, 'score': (0.4233751628610233, 0.8354523442016119,
2795.9355968184313)}, {'params': {'covariance_type': 'tied', 'n_components': 7,
'random_state': 42}, 'score': (0.3941742607402815, 0.9004067563705496,
2844.6189227511004)}, {'params': {'covariance_type': 'tied', 'n_components': 8,
'random_state': 42}, 'score': (0.39663257131258595, 0.8240649112336106,
2702.3600062295864)}, {'params': {'covariance_type': 'tied', 'n_components': 9,
'random_state': 42}, 'score': (0.32626432324796845, 0.9746493916893199,
2739.9402632319634)}, {'params': {'covariance_type': 'diag', 'n_components': 3,
'random_state': 42}, 'score': (0.4018443033555822, 0.9556221911428263,
3755.692769673924)}, {'params': {'covariance_type': 'diag', 'n_components': 4,
'random_state': 42}, 'score': (0.36553761300524823, 0.8611689301071893,
3767.7604933328557)}, {'params': {'covariance_type': 'diag', 'n_components': 5,
'random_state': 42}, 'score': (0.3452697802707615, 0.8548886908372776,
4114.40494912166)}, {'params': {'covariance_type': 'diag', 'n_components': 6,
'random_state': 42}, 'score': (0.3330349732998816, 0.9448875375008067,
3644.0911862791404)}, {'params': {'covariance_type': 'diag', 'n_components': 7,
'random_state': 42}, 'score': (0.31278623134006295, 0.8667559214142916,
3472.1789222281745)}, {'params': {'covariance_type': 'diag', 'n_components': 8,
'random_state': 42}, 'score': (0.3222959122793512, 0.8232111074421984,
3541.7827575786823)}, {'params': {'covariance_type': 'diag', 'n_components': 9,

```

```

'random_state': 42}, 'score': (0.340962733107502, 0.8552718617622883,
3889.235710112541)), {'params': {'covariance_type': 'spherical', 'n_components':
3, 'random_state': 42}, 'score': (0.39026125830547936, 0.9348541331056331,
3457.160485994782)}, {'params': {'covariance_type': 'spherical', 'n_components':
4, 'random_state': 42}, 'score': (0.4287835620499266, 0.8140597070646715,
4259.2761663651045)}, {'params': {'covariance_type': 'spherical',
'n_components': 5, 'random_state': 42}, 'score': (0.35831855922135986,
0.8368526663907538, 4257.487472313238)}, {'params': {'covariance_type':
'spherical', 'n_components': 6, 'random_state': 42}, 'score':
(0.3071984422159643, 0.8103924516094536, 3482.3257735449474)}, {'params':
{'covariance_type': 'spherical', 'n_components': 7, 'random_state': 42},
'score': (0.31303113915821773, 0.8679003257561205, 3687.235807005993)},
{'params': {'covariance_type': 'spherical', 'n_components': 8, 'random_state':
42}, 'score': (0.30086362300956127, 0.8084902823604136, 3491.3111702056317)},
{'params': {'covariance_type': 'spherical', 'n_components': 9, 'random_state':
42}, 'score': (0.3165891507219381, 0.8597828350799971, 3609.7724039925515)}], 3:
[{'params': {'covariance_type': 'full', 'n_components': 3, 'random_state': 42},
'score': (0.2761604300244667, 1.3491048186340073, 1967.925715316944)},
{'params': {'covariance_type': 'full', 'n_components': 4, 'random_state': 42},
'score': (0.23436594142642989, 1.2292833174580986, 2040.097899959354)},
{'params': {'covariance_type': 'full', 'n_components': 5, 'random_state': 42},
'score': (0.2745516449322517, 1.0501811059873638, 2098.835486230293)},
{'params': {'covariance_type': 'full', 'n_components': 6, 'random_state': 42},
'score': (0.2290713097486788, 1.1684984670437646, 1829.108023956888)},
{'params': {'covariance_type': 'full', 'n_components': 7, 'random_state': 42},
'score': (0.13732348462158325, 1.3096871416250977, 1607.4612310518512)},
{'params': {'covariance_type': 'full', 'n_components': 8, 'random_state': 42},
'score': (0.16760936065118398, 1.191778215389808, 1795.2881285065002)},
{'params': {'covariance_type': 'full', 'n_components': 9, 'random_state': 42},
'score': (0.13974514698226476, 1.449969198211288, 1477.4157231953304)},
{'params': {'covariance_type': 'tied', 'n_components': 3, 'random_state': 42},
'score': (0.34203844704416764, 1.1091035618400855, 2032.8612574697083)},
{'params': {'covariance_type': 'tied', 'n_components': 4, 'random_state': 42},
'score': (0.33373379792159613, 0.9879215730199297, 2590.7260057023595)},
{'params': {'covariance_type': 'tied', 'n_components': 5, 'random_state': 42},
'score': (0.31780889007857266, 0.9493092520892438, 2339.6101538667144)},
{'params': {'covariance_type': 'tied', 'n_components': 6, 'random_state': 42},
'score': (0.32516487032687746, 1.059874967714211, 2435.8087636248492)},
{'params': {'covariance_type': 'tied', 'n_components': 7, 'random_state': 42},
'score': (0.3083549157268952, 1.0212199004007696, 2102.9334999314324)},
{'params': {'covariance_type': 'tied', 'n_components': 8, 'random_state': 42},
'score': (0.32812685674822845, 1.0934453106802136, 2213.196971710516)},
{'params': {'covariance_type': 'tied', 'n_components': 9, 'random_state': 42},
'score': (0.3376236907104415, 0.9715721808001361, 2241.632209537098)},
{'params': {'covariance_type': 'diag', 'n_components': 3, 'random_state': 42},
'score': (0.31181795617459884, 1.206382560689101, 2561.844100583126)},
{'params': {'covariance_type': 'diag', 'n_components': 4, 'random_state': 42},
'score': (0.3021304633379264, 1.2186816967654743, 1996.0190677963471)},

```

```

{'params': {'covariance_type': 'diag', 'n_components': 5, 'random_state': 42},
'score': (0.2931429178814123, 1.2831460957638563, 2115.3021426253354)},
{'params': {'covariance_type': 'diag', 'n_components': 6, 'random_state': 42},
'score': (0.2517733436035507, 0.991189147037546, 2078.976760613973)}, {'params':
{'covariance_type': 'diag', 'n_components': 7, 'random_state': 42}, 'score':
(0.2766532902695155, 1.0901890078465812, 2083.8324701424863)}, {'params':
{'covariance_type': 'diag', 'n_components': 8, 'random_state': 42}, 'score':
(0.29325801888821895, 1.0617401417665655, 2313.259188041016)}, {'params':
{'covariance_type': 'diag', 'n_components': 9, 'random_state': 42}, 'score':
(0.2770732921515457, 1.0726340352433619, 2165.728957774773)}, {'params':
{'covariance_type': 'spherical', 'n_components': 3, 'random_state': 42},
'score': (0.2663737076108886, 1.1215918593324812, 1670.8150726667586)},
{'params': {'covariance_type': 'spherical', 'n_components': 4, 'random_state':
42}, 'score': (0.3196848791824092, 1.0252584788349268, 2677.2501809247196)},
{'params': {'covariance_type': 'spherical', 'n_components': 5, 'random_state':
42}, 'score': (0.2801533844520979, 0.9776893294191499, 2306.160294942312)},
{'params': {'covariance_type': 'spherical', 'n_components': 6, 'random_state':
42}, 'score': (0.2574334017671572, 1.079764645027932, 2345.8299141063967)},
{'params': {'covariance_type': 'spherical', 'n_components': 7, 'random_state':
42}, 'score': (0.2697456346036244, 1.0962248507042485, 2206.4978571770234)},
{'params': {'covariance_type': 'spherical', 'n_components': 8, 'random_state':
42}, 'score': (0.31109344303149933, 1.0226143323870698, 2509.992701609847)},
{'params': {'covariance_type': 'spherical', 'n_components': 9, 'random_state':
42}, 'score': (0.30105205973762167, 1.0071149789113607, 2351.9057746149288)}],
4: [{'params': {'covariance_type': 'full', 'n_components': 3, 'random_state':
42}, 'score': (0.19420868027825788, 1.4520262213905095, 1441.1756188672693)},
{'params': {'covariance_type': 'full', 'n_components': 4, 'random_state': 42},
'score': (0.21319763220383847, 1.61847708143143, 955.5250850677919)}, {'params':
{'covariance_type': 'full', 'n_components': 5, 'random_state': 42}, 'score':
(0.19951240934613002, 1.4706925034700775, 1126.7745754407647)}, {'params':
{'covariance_type': 'full', 'n_components': 6, 'random_state': 42}, 'score':
(0.09146743450364896, 1.3401324722717556, 1058.4516645511392)}, {'params':
{'covariance_type': 'full', 'n_components': 7, 'random_state': 42}, 'score':
(0.08975652234281292, 1.478651535919348, 976.6780787667151)}, {'params':
{'covariance_type': 'full', 'n_components': 8, 'random_state': 42}, 'score':
(0.08873172158359106, 1.5098913280903652, 945.4859821960617)}, {'params':
{'covariance_type': 'full', 'n_components': 9, 'random_state': 42}, 'score':
(0.09546184545463755, 1.55001695860957, 917.063685436746)}, {'params':
{'covariance_type': 'tied', 'n_components': 3, 'random_state': 42}, 'score':
(0.3057992128037759, 1.2495969484538458, 1685.1712367008677)}, {'params':
{'covariance_type': 'tied', 'n_components': 4, 'random_state': 42}, 'score':
(0.32758466692711463, 1.056019921532776, 1719.7517536283328)}, {'params':
{'covariance_type': 'tied', 'n_components': 5, 'random_state': 42}, 'score':
(0.2970885571057133, 1.0565860503146218, 1657.4850138628258)}, {'params':
{'covariance_type': 'tied', 'n_components': 6, 'random_state': 42}, 'score':
(0.2348798970965236, 1.220802895366555, 1695.0738341312947)}, {'params':
{'covariance_type': 'tied', 'n_components': 7, 'random_state': 42}, 'score':
(0.20898801575530174, 1.5589237114112922, 1242.3764173504137)}, {'params':

```

```

{'covariance_type': 'tied', 'n_components': 8, 'random_state': 42}, 'score':
(0.21424849132643184, 1.3443938129236925, 1237.1154984117147)}, {'params':
{'covariance_type': 'tied', 'n_components': 9, 'random_state': 42}, 'score':
(0.24471458210170266, 1.260065118343325, 1298.9596401280714)}, {'params':
{'covariance_type': 'diag', 'n_components': 3, 'random_state': 42}, 'score':
(0.2617356693871496, 1.3818948246784835, 2011.0359114565451)}, {'params':
{'covariance_type': 'diag', 'n_components': 4, 'random_state': 42}, 'score':
(0.25986141763464676, 1.3510293353344098, 1214.227764299998)}, {'params':
{'covariance_type': 'diag', 'n_components': 5, 'random_state': 42}, 'score':
(0.2343515366156703, 1.3951635984937054, 1622.837584044457)}, {'params':
{'covariance_type': 'diag', 'n_components': 6, 'random_state': 42}, 'score':
(0.17700620454266547, 1.4345956849195254, 1462.2017698415495)}, {'params':
{'covariance_type': 'diag', 'n_components': 7, 'random_state': 42}, 'score':
(0.15580444669802132, 1.5679441692516518, 1237.0166962246567)}, {'params':
{'covariance_type': 'diag', 'n_components': 8, 'random_state': 42}, 'score':
(0.1812238801531975, 1.5341255048239613, 1314.635722706089)}, {'params':
{'covariance_type': 'diag', 'n_components': 9, 'random_state': 42}, 'score':
(0.14014057475174457, 1.3244156390042643, 1188.325512363161)}, {'params':
{'covariance_type': 'spherical', 'n_components': 3, 'random_state': 42},
'score': (0.23743447644213767, 1.624897364786088, 1644.5286399711083)},
{'params': {'covariance_type': 'spherical', 'n_components': 4, 'random_state':
42}, 'score': (0.27415149182303405, 1.1734953545956008, 2032.705344489196)},
{'params': {'covariance_type': 'spherical', 'n_components': 5, 'random_state':
42}, 'score': (0.19016209794787592, 1.1807182336827169, 1533.3332635861484)},
{'params': {'covariance_type': 'spherical', 'n_components': 6, 'random_state':
42}, 'score': (0.11914852861641698, 1.233664045331519, 1324.7904376001197)},
{'params': {'covariance_type': 'spherical', 'n_components': 7, 'random_state':
42}, 'score': (0.19575958575715838, 1.2460772332611807, 1716.795041880433)},
{'params': {'covariance_type': 'spherical', 'n_components': 8, 'random_state':
42}, 'score': (0.2104414123427196, 1.177065352320935, 1677.6518893198536)},
{'params': {'covariance_type': 'spherical', 'n_components': 9, 'random_state':
42}, 'score': (0.21273904858731812, 1.1727701424520254, 1616.8507261858617)}],
5: [{'params': {'covariance_type': 'full', 'n_components': 3, 'random_state':
42}, 'score': (0.22759746119373053, 1.7104646687770872, 1039.9141643869343)},
{'params': {'covariance_type': 'full', 'n_components': 4, 'random_state': 42},
'score': (0.2123828486027197, 1.9671066902733698, 820.2306664911905)},
{'params': {'covariance_type': 'full', 'n_components': 5, 'random_state': 42},
'score': (0.1801964992135855, 2.2833754710298964, 739.7402053314252)},
{'params': {'covariance_type': 'full', 'n_components': 6, 'random_state': 42},
'score': (0.16595698229262365, 2.138195451124142, 687.1497483024817)},
{'params': {'covariance_type': 'full', 'n_components': 7, 'random_state': 42},
'score': (0.07335834569253794, 2.071381969208873, 683.1688737869931)},
{'params': {'covariance_type': 'full', 'n_components': 8, 'random_state': 42},
'score': (0.052220597596502806, 1.6574116192454236, 757.7463958701226)},
{'params': {'covariance_type': 'full', 'n_components': 9, 'random_state': 42},
'score': (0.059185569387614836, 1.9972879721660088, 738.9564696830748)},
{'params': {'covariance_type': 'tied', 'n_components': 3, 'random_state': 42},
'score': (0.2638774489722794, 1.3972254286516843, 1444.7017706352372)},

```



```

{'params': {'covariance_type': 'tied', 'n_components': 4, 'random_state': 42},
'score': (0.2902777320007168, 1.2445489104635077, 1263.3649065283998)},
{'params': {'covariance_type': 'tied', 'n_components': 5, 'random_state': 42},
'score': (0.2806338044156119, 1.6608040446610581, 1203.3720218571873)},
{'params': {'covariance_type': 'tied', 'n_components': 6, 'random_state': 42},
'score': (0.2694353016139269, 1.223812535512118, 1039.159759591033)}, {'params':
{'covariance_type': 'tied', 'n_components': 7, 'random_state': 42}, 'score':
(0.2548042142073531, 1.2884670895372337, 967.9116843399216)}, {'params':
{'covariance_type': 'tied', 'n_components': 8, 'random_state': 42}, 'score':
(0.2746497112289144, 1.3116793867449528, 1134.7253755183958)}, {'params':
{'covariance_type': 'tied', 'n_components': 9, 'random_state': 42}, 'score':
(0.2713713360220058, 1.3291425409368187, 1076.3417774894672)}, {'params':
{'covariance_type': 'diag', 'n_components': 3, 'random_state': 42}, 'score':
(0.1673700877524557, 1.8846048936927833, 668.638658952802)}, {'params':
{'covariance_type': 'diag', 'n_components': 4, 'random_state': 42}, 'score':
(0.193909901972992, 1.8600959191122335, 1130.6735553222793)}, {'params':
{'covariance_type': 'diag', 'n_components': 5, 'random_state': 42}, 'score':
(0.23625158821937667, 1.5263271169311015, 1197.3262805845916)}, {'params':
{'covariance_type': 'diag', 'n_components': 6, 'random_state': 42}, 'score':
(0.18691052863025143, 1.3995170638506347, 1102.49705605993)}, {'params':
{'covariance_type': 'diag', 'n_components': 7, 'random_state': 42}, 'score':
(0.15708674620764568, 1.4967375420634828, 1047.7045034125742)}, {'params':
{'covariance_type': 'diag', 'n_components': 8, 'random_state': 42}, 'score':
(0.14158358799473122, 1.482422442185053, 988.8618117492729)}, {'params':
{'covariance_type': 'diag', 'n_components': 9, 'random_state': 42}, 'score':
(0.1382910827837623, 1.4264579257060577, 889.8557823463981)}, {'params':
{'covariance_type': 'spherical', 'n_components': 3, 'random_state': 42},
'score': (0.23490955132381755, 1.640517839744107, 1520.0224120191215)},
{'params': {'covariance_type': 'spherical', 'n_components': 4, 'random_state':
42}, 'score': (0.13805746508266115, 1.467382626236996, 1061.2395911086744)},
{'params': {'covariance_type': 'spherical', 'n_components': 5, 'random_state':
42}, 'score': (0.16975204112907127, 1.2566558213261865, 1303.1212101597864)},
{'params': {'covariance_type': 'spherical', 'n_components': 6, 'random_state':
42}, 'score': (0.227451099255996, 1.1562163075262326, 1395.3607776540048)},
{'params': {'covariance_type': 'spherical', 'n_components': 7, 'random_state':
42}, 'score': (0.1718120712662137, 1.2964364663895172, 1252.8773411451778)},
{'params': {'covariance_type': 'spherical', 'n_components': 8, 'random_state':
42}, 'score': (0.22127477489420472, 1.4368332234307881, 1301.740742408087)},
{'params': {'covariance_type': 'spherical', 'n_components': 9, 'random_state':
42}, 'score': (0.22520831083546522, 1.4152515178031484, 1254.0927178159043)}}],
'Spectral': {2: [{'params': {'affinity': 'nearest_neighbors', 'n_clusters': 3,
'random_state': 42}, 'score': (0.08685794141280594, 0.8671966783160222,
1219.9871720858157)}, {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
4, 'random_state': 42}, 'score': (0.137818496601079, 0.9259509537049319,
1219.9336025599953)}, {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
5, 'random_state': 42}, 'score': (0.1492709660861394, 1.0519457320756715,
1224.0920827350965)}, {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
6, 'random_state': 42}, 'score': (0.15137657655893272, 1.0157350837233665,

```

```

1126.6513132064683)), {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
7, 'random_state': 42}, 'score': (0.2025093367159689, 0.9011720995719121,
1702.3151644956567)}, {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
8, 'random_state': 42}, 'score': (0.16005523599544824, 0.8749133114201284,
1890.6491289160663)}, {'params': {'affinity': 'nearest_neighbors', 'n_clusters':
9, 'random_state': 42}, 'score': (0.17749064312589377, 0.8123119172203954,
1740.4248969166804)}}], 3: [{'params': {'affinity': 'nearest_neighbors',
'n_clusters': 3, 'random_state': 42}, 'score': (0.018621792106440806,
1.1191930073174445, 549.5310026296637)}, {'params': {'affinity':
'nearest_neighbors', 'n_clusters': 4, 'random_state': 42}, 'score':
(0.1330021326868381, 1.0729691842685878, 952.9402336257388)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 5, 'random_state': 42}, 'score':
(0.14218096806622701, 1.0672935301233388, 790.6086042729972)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 6, 'random_state': 42}, 'score':
(0.017419181929286112, 1.0637660730083631, 816.4585878809143)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 7, 'random_state': 42}, 'score':
(0.024029686559107144, 1.3275899507613151, 699.5673355649217)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 8, 'random_state': 42}, 'score':
(0.07376368414159024, 0.9906443013465221, 1036.173576872354)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 9, 'random_state': 42}, 'score':
(0.09210665119551663, 1.0237254970422571, 1358.131733373495)}}], 4: [{'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 3, 'random_state': 42}, 'score':
(0.07733170100914676, 0.7238370514096976, 20.006251502275727)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 4, 'random_state': 42}, 'score':
(0.09906614114155038, 1.1086317774572028, 220.6549647800551)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 5, 'random_state': 42}, 'score':
(0.10099779610128987, 1.1038401507844493, 165.66228409220628)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 6, 'random_state': 42}, 'score':
(-0.11451322536618705, 1.1366100614999732, 177.38033679733516)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 7, 'random_state': 42}, 'score':
(-0.049076349023641025, 1.2101105921333304, 543.2040779772422)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 8, 'random_state': 42}, 'score':
(-0.0312456078036606, 1.1982895967643372, 504.8035966472851)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 9, 'random_state': 42}, 'score':
(-0.0011722287092773924, 1.27120891831037, 582.1189276920579)}}], 5: [{'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 3, 'random_state': 42}, 'score':
(0.06065175026075889, 0.7721659776151265, 48.409139015353084)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 4, 'random_state': 42}, 'score':
(0.06475904932056055, 1.1392347425625005, 190.22496146748426)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 5, 'random_state': 42}, 'score':
(-0.061616663014659276, 1.1972298865133908, 229.57458241379769)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 6, 'random_state': 42}, 'score':
(-0.018508359990280682, 1.3630704700560552, 331.2737825762419)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 7, 'random_state': 42}, 'score':
(-0.013266993467488214, 1.2949544116090814, 283.42834938184967)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 8, 'random_state': 42}, 'score':
(-0.081024109984228, 1.3903815005650437, 264.4333699848559)}, {'params':
{'affinity': 'nearest_neighbors', 'n_clusters': 9, 'random_state': 42}, 'score':

```

```
(-0.026673499654258664, 1.436150147722933, 426.5839943247175))]]}]}
```

### 3.6 Results

```
[16]: # Display best score and hyperparameters for each method
best_hyperparameters = {}
for method, pca_comps in results.items():
    for pca in pca_comps:
        best_score = max(pca_comps[pca], key=lambda x: x['score'][0]) # Get
        ↳ the score with the highest first value
        if method not in best_hyperparameters:
            best_hyperparameters[method] = {'pca': pca, 'params': {}
        ↳ best_score['params'], 'score': best_score['score']}
        else:
            current_best_score = best_hyperparameters[method]['score']
            if best_score['score'] > current_best_score:
                best_hyperparameters[method] = {'pca': pca, 'params': {}
        ↳ best_score['params'], 'score': best_score['score']}

print("Best Hyperparameters:")
for method, params in best_hyperparameters.items():
    print(f"Method: {method}, PCA Components: {params['pca']}, Parameters:
    ↳ {params['params']}, Score: {params['score']}")
```

Best Hyperparameters:

Method: K-Means, PCA Components: 2, Parameters: {'init': 'k-means++',  
'max\_iter': 100, 'n\_clusters': 4, 'n\_init': 10}, Score: (0.43696184323503295,  
0.800807992091704, 4382.46484845105)

Method: Agglomerative, PCA Components: 2, Parameters: {'linkage': 'single',  
'n\_clusters': 3}, Score: (0.7144848175386341, 0.16920588504384718,  
24.292275620344395)

Method: DBSCAN, PCA Components: 2, Parameters: {'eps': 2.0, 'min\_samples': 3},  
Score: (0.7214246011996667, 0.6788983701172336, 76.81002251018943)

Method: GMM, PCA Components: 2, Parameters: {'covariance\_type': 'tied',  
'n\_components': 4, 'random\_state': 42}, Score: (0.4375197819795308,  
0.7369848464880445, 3188.3251681475836)

Method: Spectral, PCA Components: 2, Parameters: {'affinity':  
'nearest\_neighbors', 'n\_clusters': 7, 'random\_state': 42}, Score:  
(0.2025093367159689, 0.9011720995719121, 1702.3151644956567)

```
[17]: import matplotlib.pyplot as plt

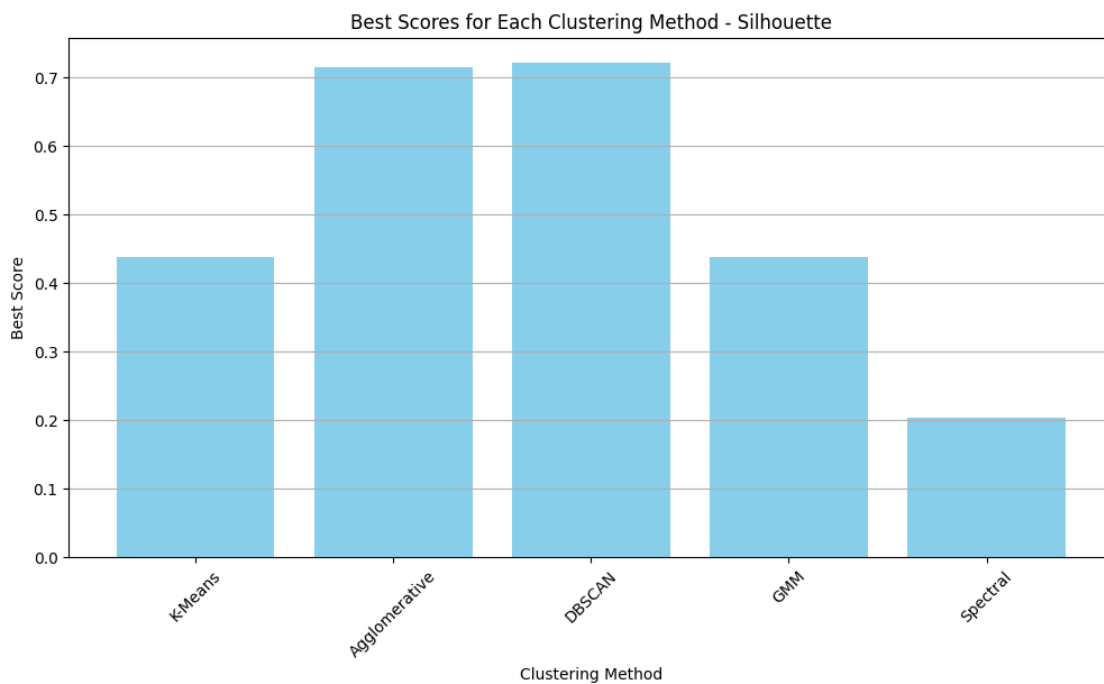
# Data to plot
methods = list(best_hyperparameters.keys())
pca_components = [best_hyperparameters[method]['pca'] for method in methods]
scores = [best_hyperparameters[method]['score'][0] for method in methods] #
    ↳ Using the first score for plotting
```

```

# Create a bar plot
plt.figure(figsize=(12, 6))
plt.bar(methods, scores, color='skyblue')
plt.title('Best Scores for Each Clustering Method - Silhouette')
plt.xlabel('Clustering Method')
plt.ylabel('Best Score')
plt.xticks(rotation=45)
plt.grid(axis='y')

# Show the plot
plt.show()

```



```

[18]: import matplotlib.pyplot as plt

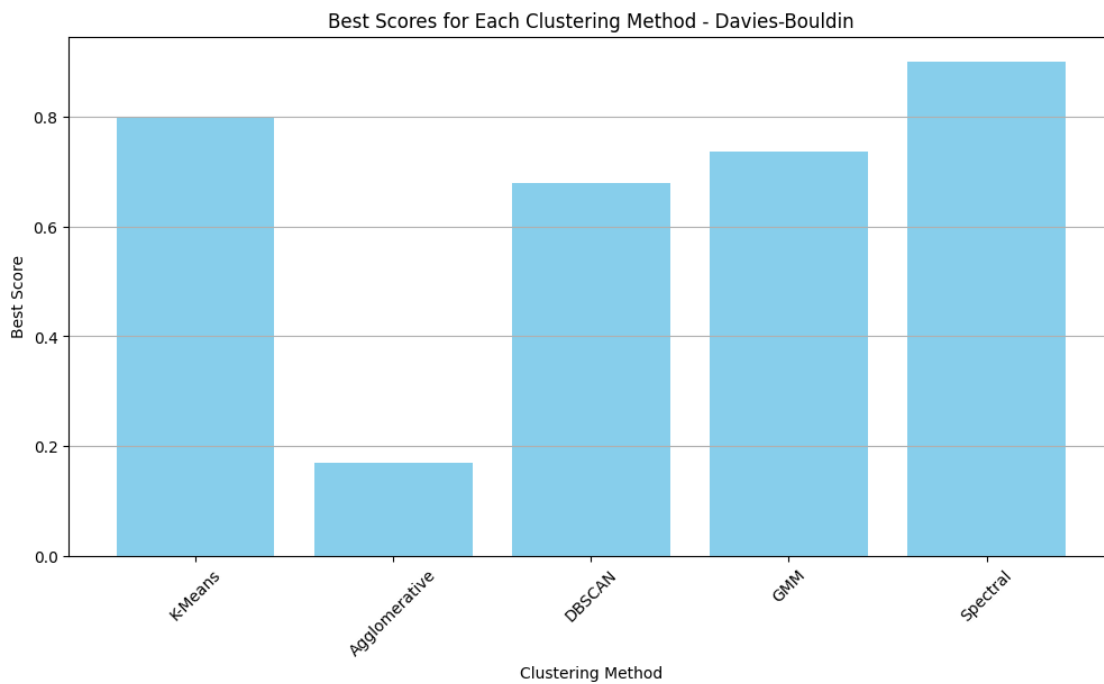
# Data to plot
methods = list(best_hyperparameters.keys())
pca_components = [best_hyperparameters[method]['pca'] for method in methods]
scores = [best_hyperparameters[method]['score'][1] for method in methods] # Using the first score for plotting

# Create a bar plot
plt.figure(figsize=(12, 6))
plt.bar(methods, scores, color='skyblue')
plt.title('Best Scores for Each Clustering Method - Davies-Bouldin')

```

```
plt.xlabel('Clustering Method')
plt.ylabel('Best Score')
plt.xticks(rotation=45)
plt.grid(axis='y')

# Show the plot
plt.show()
```

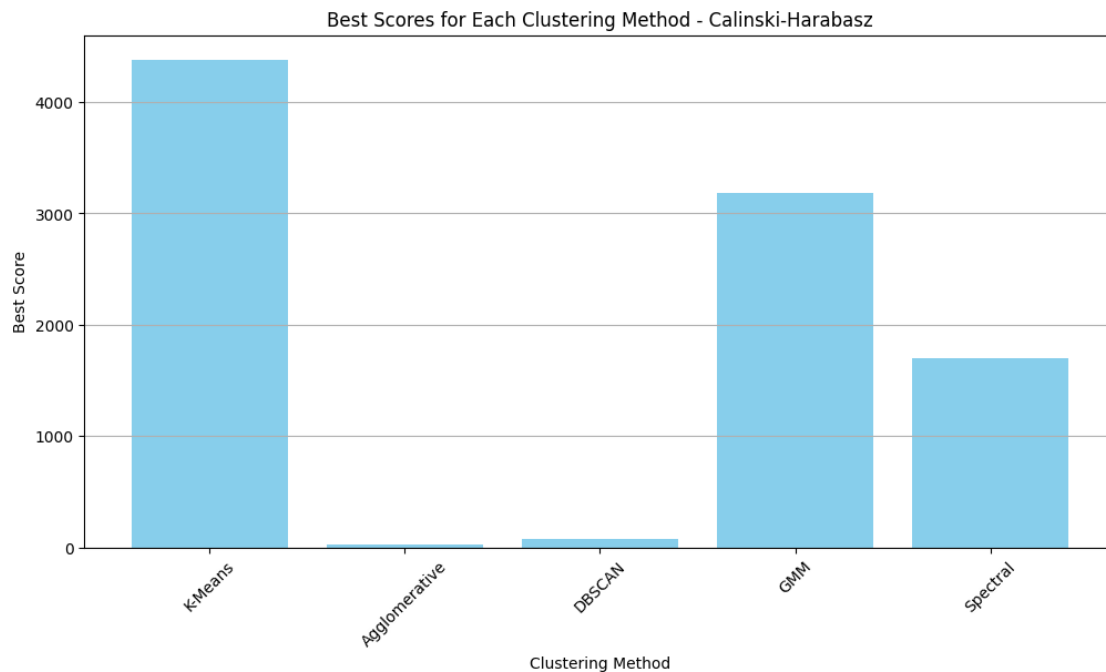


```
[19]: import matplotlib.pyplot as plt

# Data to plot
methods = list(best_hyperparameters.keys())
pca_components = [best_hyperparameters[method]['pca'] for method in methods]
scores = [best_hyperparameters[method]['score'][2] for method in methods] # Using the first score for plotting

# Create a bar plot
plt.figure(figsize=(12, 6))
plt.bar(methods, scores, color='skyblue')
plt.title('Best Scores for Each Clustering Method - Calinski-Harabasz')
plt.xlabel('Clustering Method')
plt.ylabel('Best Score')
plt.xticks(rotation=45)
plt.grid(axis='y')
```

```
# Show the plot
plt.show()
```



```
[20]: # Best clustering method: K-Means
best_method = "K-Means"
best_pca_components = 2
best_parameters = {'init': 'k-means++', 'max_iter': 100, 'n_clusters': 4,
                  ↪ 'n_init': 10}
best_score = (0.43696184323503295, 0.800807992091704, 4382.46484845105)

# Apply K-Means with the best parameters
kmeans = KMeans(init=best_parameters['init'],
                ↪ max_iter=best_parameters['max_iter'],
                    n_clusters=best_parameters['n_clusters'],
                ↪ n_init=best_parameters['n_init'], random_state=42)
labels = kmeans.fit_predict(reduced_data)

# Calculate intra-cluster variances
intra_variances = []
n_clusters = best_parameters['n_clusters'] # Use the number of clusters from
                ↪ parameters
for k in range(n_clusters):
    cluster_data = reduced_data[labels == k]
```

```

    intra_variance = np.var(cluster_data, axis=0).mean() # Mean variance of
    ↪ features
    intra_variances.append(intra_variance)

# Intra-group variance
variance_intra = np.mean(intra_variances)

# Calculate cluster centers
centers = kmeans.cluster_centers_

# Inter-group variance
mean_center = np.mean(centers, axis=0)
variance_inter = np.var(centers, axis=0).mean() # Mean variance of centers

# Calculate the ratio
ratio = variance_inter / variance_intra

print(f"Variance Intra-Groupes: {variance_intra}")
print(f"Variance Inter-Groupes: {variance_inter}")
print(f"Ratio: {ratio}")

```

Variance Intra-Groupes: 2.1261409538972154

Variance Inter-Groupes: 3.32245739214457

Ratio: 1.562670332865047

```

[21]: import matplotlib.pyplot as plt

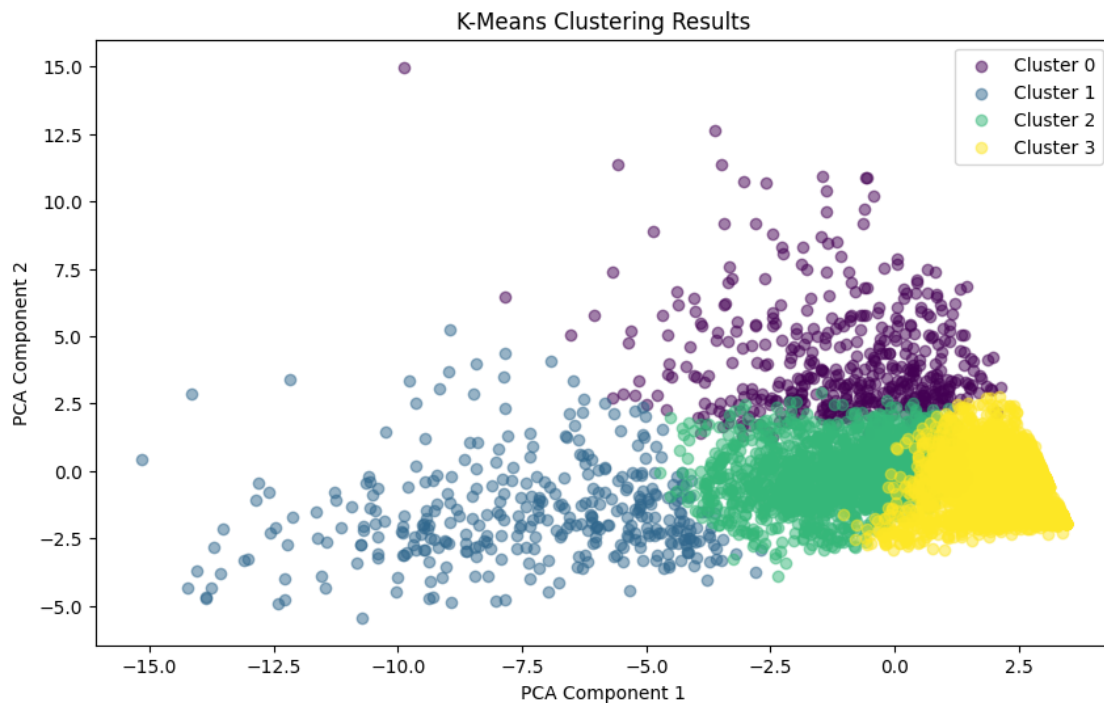
# Plot the reduced data with cluster labels
plt.figure(figsize=(10, 6))
unique_labels = set(labels)
colors = plt.cm.viridis(np.linspace(0, 1, len(unique_labels))) # Updated to
    ↪ use viridis directly

for i, label in enumerate(unique_labels):
    if label == -1: # Noise points
        color = 'k' # Black for noise
    else:
        color = colors[i] # Use the index to get the color
    plt.scatter(reduced_data[labels == label, 0], reduced_data[labels == label,
    ↪ 1],
                color=color, label=f'Cluster {label}' if label != -1 else
    ↪ 'Noise', alpha=0.5)

plt.title('K-Means Clustering Results')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.legend()

```

```
plt.show()
```



## 4 Descriptif

```
[23]: # Assuming 'data' is your DataFrame and 'labels' are the cluster labels
subset_data['Cluster'] = labels # Convert labels to a list before adding to
↳ the DataFrame

# Group by cluster and calculate summary statistics
cluster_summary = subset_data.groupby('Cluster').agg(['mean', 'std', 'count'])

# Display the summary statistics for each cluster
print(cluster_summary)

# Calculate the importance of each feature for each cluster
importance_summary = {}

for label in unique_labels:
    if label != -1: # Skip noise points
        cluster_data = subset_data[subset_data['Cluster'] == label]
        importance = cluster_data.drop(columns=['Cluster']).mean() # Calculate
↳ mean for each feature
        importance_summary[label] = importance
```



```

# Convert the summary to a DataFrame for better visualization
importance_df = pd.DataFrame(importance_summary).T

# Create a summary of the most important features for each cluster
importance_summary_str = {}

for label in unique_labels:
    if label != -1: # Skip noise points
        # Get the importance of features for the current cluster
        importance = importance_df.loc[label]
        # Sort features by importance
        sorted_importance = importance.sort_values(ascending=False)
        # Create a summary string for the top features
        summary = ', '.join([f'{feature} : {value:.1f}' for feature, value in
sorted_importance.items()])
        importance_summary_str[label] = summary

# Display the importance summary for each cluster
print("Importance Summary for Each Cluster:")
for label in importance_summary_str.keys():
    # Get the top 5 features for the current cluster
    top_features = importance_df.loc[label].nlargest(5)
    # Check the most common value for these features in the cluster
    most_common_values = {feature: subset_data[subset_data['Cluster'] ==
label][feature].mode()[0] for feature in top_features.index}
    print(f'Cluster {label}: {top_features.index.tolist()} with most common
values: {most_common_values}')

```

|         | A11      |          |       | A12      |          |       | A13 \    |
|---------|----------|----------|-------|----------|----------|-------|----------|
|         | mean     | std      | count | mean     | std      | count | mean     |
| Cluster |          |          |       |          |          |       |          |
| 0       | 0.976395 | 0.151979 | 466   | 0.107296 | 0.309822 | 466   | 0.450644 |
| 1       | 0.763587 | 0.425457 | 368   | 0.277174 | 0.448212 | 368   | 0.546196 |
| 2       | 0.921525 | 0.268994 | 1784  | 0.117713 | 0.322358 | 1784  | 0.446749 |
| 3       | 0.641898 | 0.479543 | 2382  | 0.320319 | 0.466697 | 2382  | 0.337112 |

|         | A14      |       |      | ... | C1_6_slice | C1_7_slice | \        |       |
|---------|----------|-------|------|-----|------------|------------|----------|-------|
|         | std      | count | mean | ... | count      | mean       | std      | count |
| Cluster |          |       |      | ... |            |            |          |       |
| 0       | 0.498093 | 466   | 0.0  | ... | 466        | 4.628755   | 0.643870 | 466   |
| 1       | 0.498539 | 368   | 0.0  | ... | 368        | 2.644022   | 0.916644 | 368   |
| 2       | 0.497296 | 1784  | 0.0  | ... | 1784       | 4.409753   | 0.636112 | 1784  |
| 3       | 0.472823 | 2382  | 0.0  | ... | 2382       | 4.915197   | 0.287548 | 2382  |

| C1_8_slice |     |       | C1_9_slice |           |
|------------|-----|-------|------------|-----------|
| mean       | std | count | mean       | std count |

| Cluster |          |          |      |          |          |      |
|---------|----------|----------|------|----------|----------|------|
| 0       | 4.283262 | 0.698092 | 466  | 4.317597 | 0.757750 | 466  |
| 1       | 2.532609 | 0.831283 | 368  | 2.532609 | 0.824702 | 368  |
| 2       | 4.122197 | 0.628970 | 1784 | 4.119395 | 0.691900 | 1784 |
| 3       | 4.786314 | 0.426069 | 2382 | 4.814861 | 0.415650 | 2382 |

[4 rows x 81 columns]

Importance Summary for Each Cluster:

Cluster 0: ['A8\_1\_slice', 'A8\_2\_slice', 'A8\_3\_slice', 'A8\_4\_slice', 'C1\_6\_slice'] with most common values: {'A8\_1\_slice': 20, 'A8\_2\_slice': 20, 'A8\_3\_slice': 10, 'A8\_4\_slice': 5, 'C1\_6\_slice': 5}

Cluster 1: ['A8\_2\_slice', 'A8\_1\_slice', 'A8\_3\_slice', 'C1\_6\_slice', 'C1\_3\_slice'] with most common values: {'A8\_2\_slice': 1, 'A8\_1\_slice': 1, 'A8\_3\_slice': 1, 'C1\_6\_slice': 2, 'C1\_3\_slice': 3}

Cluster 2: ['A8\_2\_slice', 'C1\_6\_slice', 'C1\_4\_slice', 'C1\_3\_slice', 'C1\_7\_slice'] with most common values: {'A8\_2\_slice': 2, 'C1\_6\_slice': 5, 'C1\_4\_slice': 5, 'C1\_3\_slice': 5, 'C1\_7\_slice': 5}

Cluster 3: ['C1\_6\_slice', 'C1\_4\_slice', 'C1\_3\_slice', 'C1\_7\_slice', 'C1\_5\_slice'] with most common values: {'C1\_6\_slice': 5, 'C1\_4\_slice': 5, 'C1\_3\_slice': 5, 'C1\_7\_slice': 5, 'C1\_5\_slice': 5}

/tmp/ipykernel\_21221/2211454646.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
subset_data['Cluster'] = labels # Convert labels to a list before adding to
the DataFrame
```

**Cluster 0 :**

### Features principales

- **A8\_1\_slice** (Temps passé à l'entretien au printemps) : *20 heures*
- **A8\_2\_slice** (Temps passé à l'entretien en été) : *20 heures*
- **A8\_3\_slice** (Temps passé à l'entretien en automne) : *10 heures*
- **A8\_4\_slice** (Temps passé à l'entretien en hiver) : *5 heures*
- **C1\_6\_slice** (Fréquence consultation : Profils Instagram sur le jardinage) : *Jamais (5)*

**Description** Ce cluster regroupe des individus qui :

- Consacrent un temps significatif à l'entretien de leurs espaces extérieurs, surtout au printemps et en été, mais beaucoup moins en automne et hiver.
- Ne s'intéressent pas à des contenus numériques liés au jardinage sur Instagram ou n'en consultent jamais.

---

**Cluster 1 :**

### Features principales

- **A8\_2\_slice** (Temps passé à l'entretien en été) : *1 heure*
- **A8\_1\_slice** (Temps passé à l'entretien au printemps) : *1 heure*
- **A8\_3\_slice** (Temps passé à l'entretien en automne) : *1 heure*
- **C1\_6\_slice** (Fréquence consultation : Profils Instagram sur le jardinage) : *Au moins une fois par semaine (2)*
- **C1\_3\_slice** (Fréquence consultation : Blogs d'experts en aménagement) : *Une à trois fois par mois (3)*

**Description** Ce cluster regroupe des individus qui :

- Passent très peu de temps à l'entretien de leurs espaces extérieurs, quelle que soit la saison.
  - Consultent régulièrement des profils Instagram dédiés au jardinage.
  - Lisent des blogs d'experts en aménagement quelques fois par mois pour s'inspirer ou s'informer.
- 

### Cluster 2 :

#### Features principales

- **A8\_2\_slice** (Temps passé à l'entretien en été) : *2 heures*
- **C1\_6\_slice** (Fréquence consultation : Profils Instagram sur le jardinage) : *Jamais (5)*
- **C1\_4\_slice** (Fréquence consultation : Pages/groupes Facebook sur le jardinage) : *Jamais (5)*
- **C1\_3\_slice** (Fréquence consultation : Blogs d'experts en aménagement) : *Jamais (5)*
- **C1\_7\_slice** (Fréquence consultation : Forums de discussion sur le jardinage) : *Jamais (5)*

**Description** Ce cluster regroupe des individus qui :

- Passent très peu de temps à entretenir leurs espaces extérieurs, même en été.
  - Ne consultent jamais de contenus numériques liés au jardinage ou à l'aménagement des espaces extérieurs (Instagram, Facebook, blogs, forums).
  - Manifestent peu d'intérêt pour les échanges ou discussions autour de l'entretien des espaces extérieurs.
- 

### Cluster 3 :

#### Features principales

- **C1\_6\_slice** (Fréquence consultation : Profils Instagram sur le jardinage) : *Jamais (5)*
- **C1\_4\_slice** (Fréquence consultation : Pages/groupes Facebook sur le jardinage) : *Jamais (5)*
- **C1\_3\_slice** (Fréquence consultation : Blogs d'experts en aménagement) : *Jamais (5)*
- **C1\_7\_slice** (Fréquence consultation : Forums de discussion sur le jardinage) : *Jamais (5)*
- **C1\_5\_slice** (Fréquence consultation : Pages Pinterest sur le jardinage) : *Jamais (5)*

**Description** Ce cluster regroupe des individus qui :

- Ne consultent jamais de contenus numériques en rapport avec le jardinage, que ce soit sur Instagram, Facebook, Pinterest, des blogs ou des forums.
- Peuvent être caractérisés par un désintérêt total pour les ressources numériques ou en ligne dédiées à l'aménagement ou à l'entretien des espaces extérieurs.

## 5 Conclusion

Dans le cadre de notre analyse de clustering, nous avons réalisé un benchmark en utilisant plusieurs métriques de performance : Silhouette, Davies-Bouldin et Calinski-Harabasz. Nous avons appliqué différentes méthodes de clustering, notamment KMeans, Agglomerative Clustering, DBSCAN, Gaussian Mixture Model et Spectral Clustering.

Les résultats de cette évaluation ont montré que le Kmeans, encore une fois offrait les meilleures performances globales selon les métriques analysées (surtout pour 4 clusters). En conséquence, nous avons décidé de nous concentrer sur cette méthode.

En termes de variance, nous avons obtenu les résultats suivants : - Variance Intra-Groupes: 2.126  
- Variance Inter-Groupes: 3.322 - Ratio: 1.562

Ces résultats soulignent l'efficacité du KMeans dans notre analyse.