

.1

۱. تعداد دسته های نامشخص: اگر تعداد واقعی دسته ها در داده ها نامشخص باشد یا دسته ها به طور متقاضی متفاوت نباشند، k-means ممکن است نتایج نادرستی تولید کند یعنی اگر تعداد مشخصی از دسته ها برای الگوریتم مشخص نشده باشد ممکن است الگوریتم به دسته هایی که معنای زیادی ندارند تقسیم شود.

۲. توزیع ناهمگن اندازه ها: اگر اندازه دسته ها یا انحراف معیار آنها از هم متفاوت باشد الگوریتم k-means ممکن است دسته هایی را به طور نادرست تعیین کند، به طوری که دسته های با اندازه بزرگتر یا پراکندگی بیشتر، به طور ناعادلانه تر تاثیرگذار باشند.

۳. داده های پرت: حضور داده های پرت یا outliers میتواند تاثیر مخربی بر عملکرد k-means داشته باشد. این داده های پرت ممکن است باعث تشکیل دسته های اضافی یا تاثیرات غیرمنتظره دیگری بر روی مرکزهای دسته ها شود.

۴. توزیع های نامتقارن و غیرمنظم: اگر توزیع داده ها نامتقارن یا غیرمنظم باشد الگوریتم k-means ممکن است دسته هایی را تشکیل دهد که معنای آماری یا شناختی نداشته باشند. مثلاً اگر داده ها در فضای دو بعدی به صورت خطی جداپذیر نباشند، الگوریتم ممکن است به نتایج نادرستی برسد.

.2

زیاد بودن پارامتر eps در DBSCAN:

اگر مقدار eps بسیار زیاد باشد همه نقاط به عنوان همسایه شناخته میشوند و به یک دسته بزرگ تبدیل میشوند در این حالت، الگوریتم DBSCAN به عنوان یک الگوریتم خوش بندی مفید عمل نخواهد کرد و تمایل به تشخیص یک خوش بزرگ و یا حتی تشخیص یک خوش وحد خواهد داشت.

کم بودن پارامتر eps در DBSCAN:

اگر مقدار eps بسیار کم باشد بسیاری از نقاط به عنوان نویز شناخته میشوند و خوش ها به صورت مجزا تشخیص داده نمیشوند در این حالت، الگوریتم ممکن است بسیار زیاد نقاط را به عنوان نویز تشخیص داده و خوش های مهم را به طور نادرست جدا کند.

.3

(الف)

درست است.

(ب)

نادرست است چون انتروپی یکی از معیارهای ارزیابی خوش بندی است اما همواره بهترین معیار نیست.
استفاده از انتروپی برای ارزیابی خوش بندی وابسته به ماهیت داده و اهداف مسئله است.

همچنین انتروپی معمولاً برای ارزیابی خوش بندی هایی که داده ها دارای برچسب هستند استفاده میشود.
انتروپی میزان ناپایداری در یک مجموعه داده را اندازه‌گیری می‌کند. در مواردی که داده‌ها بدون برچسب هستند،
انتروپی معنای چندانی ندارد زیرا ما نمی‌توانیم ناپایداری برچسب‌ها را اندازه‌گیری کنیم. بنابراین در حالت بدون
نظرات (unsupervised) که برچسب ها در دسترس نیستند، ما نمی‌توانیم از انتروپی به عنوان معیار ارزیابی
خوش بندی استفاده کنیم.

(پ)

نادرست است چون اگر از معیار group average در خوش بندی سلسله مراتبی استفاده کنیم ممکن است
به نویز بیشتری حساس شویم به علت اینکه با محاسبه میانگین، نقاط پرت می‌توانند تاثیر بیشتری بر خوش
بندی داشته باشند. علاوه بر این، این معیار ممکن است به سمت خوش های با شکل کروی کج شود. با این
حال کاهش حساسیت به نویز و کج شدن به سمت خوش های کروی به ویژگی های خاص داده ها و شرایط
مختلف بستگی دارد.

(ت)

درست است.

.4

.1

```
[1] from google.colab import drive
drive.mount('/content/drive')

↳ Mounted at /content/drive

----- Q4 -----
```

▼ 4-1

```
[2] import warnings
warnings.filterwarnings('ignore')

[3] import pandas as pd

[4] dataframe=pd.read_csv('/content/drive/MyDrive/Wholesale customers data.csv')
dataframe.head()
```

↳ Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776

```
dataframe=pd.read_csv('/content/drive/MyDrive/Wholesale customers data.csv')
dataframe.head()
```

↳ Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

Next steps:  View recommended plots

```
[5] dataframe.info()
```

↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 8 columns):
 # Column Non-Null Count Dtype
--- -- -- -- --
 0 Channel 440 non-null int64
 1 Region 440 non-null int64
 2 Fresh 440 non-null int64
 3 Milk 440 non-null int64
 4 Grocery 440 non-null int64
 5 Frozen 440 non-null int64
 6 Detergents_Paper 440 non-null int64
 7 Delicassen 440 non-null int64

```
dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Channel     440 non-null    int64  
 1   Region      440 non-null    int64  
 2   Fresh       440 non-null    int64  
 3   Milk        440 non-null    int64  
 4   Grocery     440 non-null    int64  
 5   Frozen      440 non-null    int64  
 6   Detergents_Paper 440 non-null    int64  
 7   Delicassen  440 non-null    int64  
dtypes: int64(8)
memory usage: 27.6 KB
```

.2

4-2

```
[6] from sklearn.decomposition import PCA
     from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
data_scaled = scaler.fit_transform(dataframe)

pca = PCA(n_components=2)
principal_components = pca.fit_transform(data_scaled)

df_pca = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2'])

df_pca.to_csv('wholesale_customers_pca.csv', index=False)

df_pca.head()
```

	PC1	PC2
0	0.843939	-0.515351
1	1.062676	-0.484601
2	1.269141	0.682055
3	-1.056782	0.610821
4	0.634030	0.974199

.3

4-3

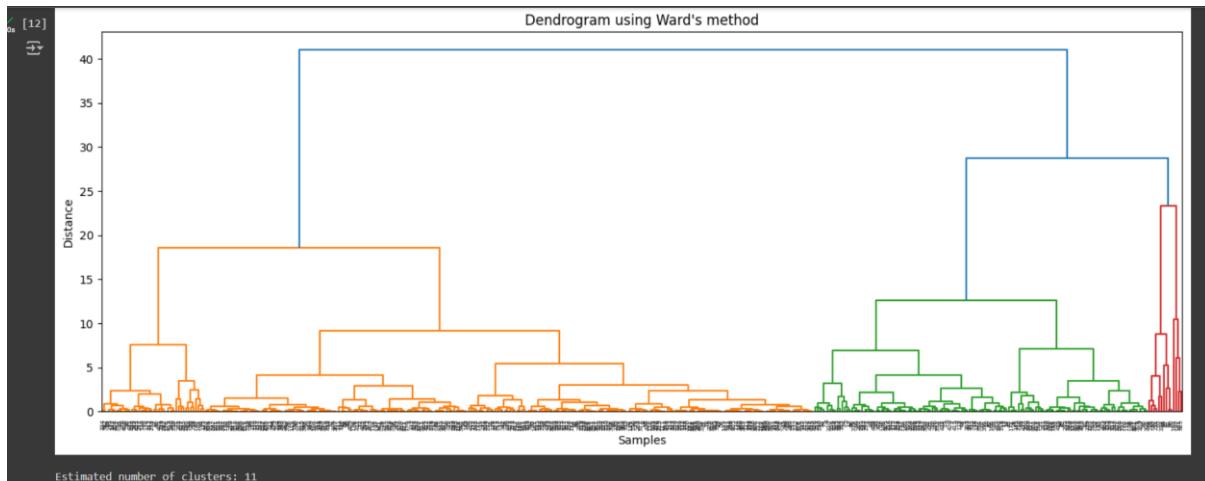
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
from scipy.cluster.hierarchy import fcluster

Z = linkage(df_pca, method='ward')

plt.figure(figsize=(17, 6))
dendrogram(Z)
plt.title('Dendrogram using Ward\'s method')
plt.xlabel('Samples')
plt.ylabel('Distance')
plt.show()

max_d = 7
clusters = fcluster(Z, max_d, criterion='distance')

print(f'\nEstimated number of clusters: {len(np.unique(clusters))}')
```



.4

4-4

```

0s  from sklearn.cluster import AgglomerativeClustering
1s  from sklearn.metrics import silhouette_score

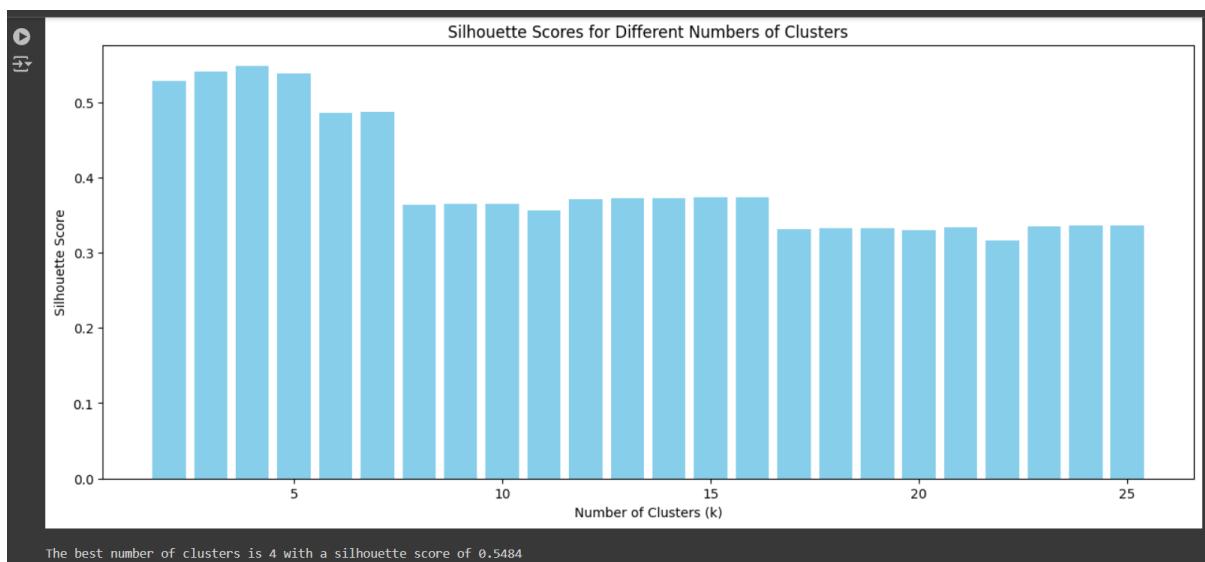
0s  silhouette_scores = []

for k in range(2, 26):
    clustering = AgglomerativeClustering(n_clusters=k)
    labels = clustering.fit_predict(df_pca)
    silhouette_avg = silhouette_score(df_pca, labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(15, 6))
plt.bar(range(2, 26), silhouette_scores, color='skyblue')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Scores for Different Numbers of Clusters')
plt.show()

best_k = np.argmax(silhouette_scores) + 2
print(f'\nThe best number of clusters is {best_k} with a silhouette score of {silhouette_scores[best_k-2]:.4f}')

```



The best number of clusters is 4 with a silhouette score of 0.5484

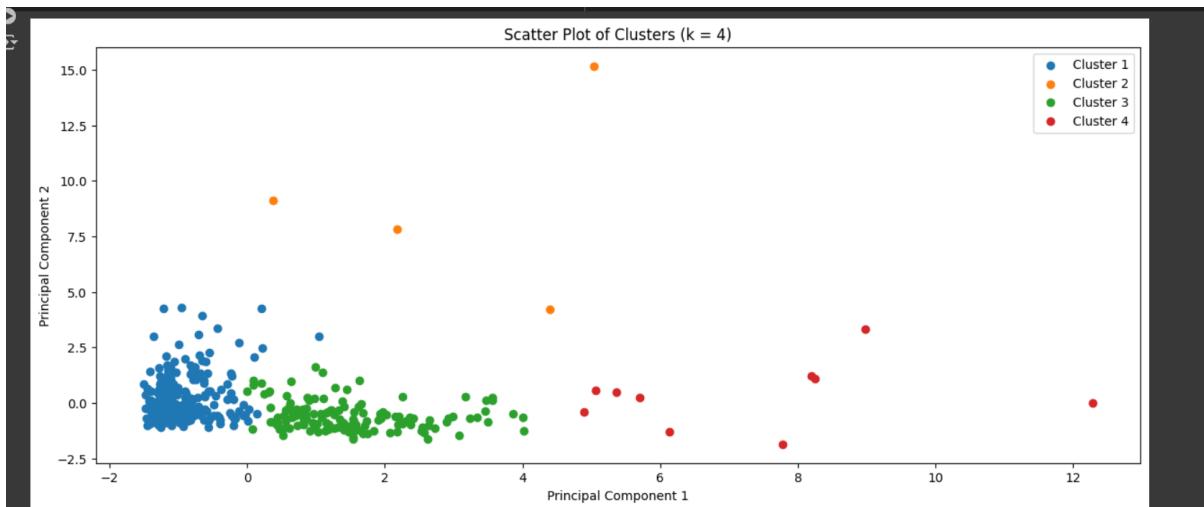
.5

4-5

```
2s best_k = 4
clustering = AgglomerativeClustering(n_clusters=best_k)
labels = clustering.fit_predict(df_pca)

pca = PCA(n_components=2)
data_pca = pca.fit_transform(df_pca)

plt.figure(figsize=(15, 6))
for cluster in np.unique(labels):
    plt.scatter(data_pca[labels == cluster, 0], data_pca[labels == cluster, 1], label=f'Cluster {cluster + 1}')
plt.title('Scatter Plot of Clusters (k = {best_k})')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.show()
```

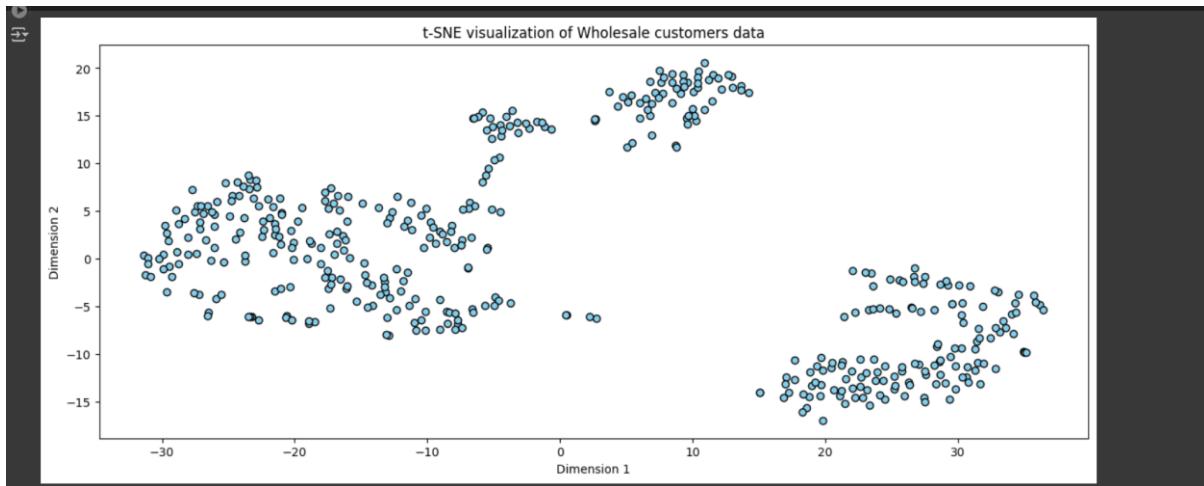


.6

4-6

```
[20] from sklearn.manifold import TSNE
3s tsne = TSNE(n_components=2, random_state=42)
data_tsne = tsne.fit_transform(data_scaled)

plt.figure(figsize=(15, 6))
plt.scatter(data_tsne[:, 0], data_tsne[:, 1], c='skyblue', edgecolor='k')
plt.title('t-SNE visualization of Wholesale customers data')
plt.xlabel('Dimension 1')
plt.ylabel('Dimension 2')
plt.show()
```



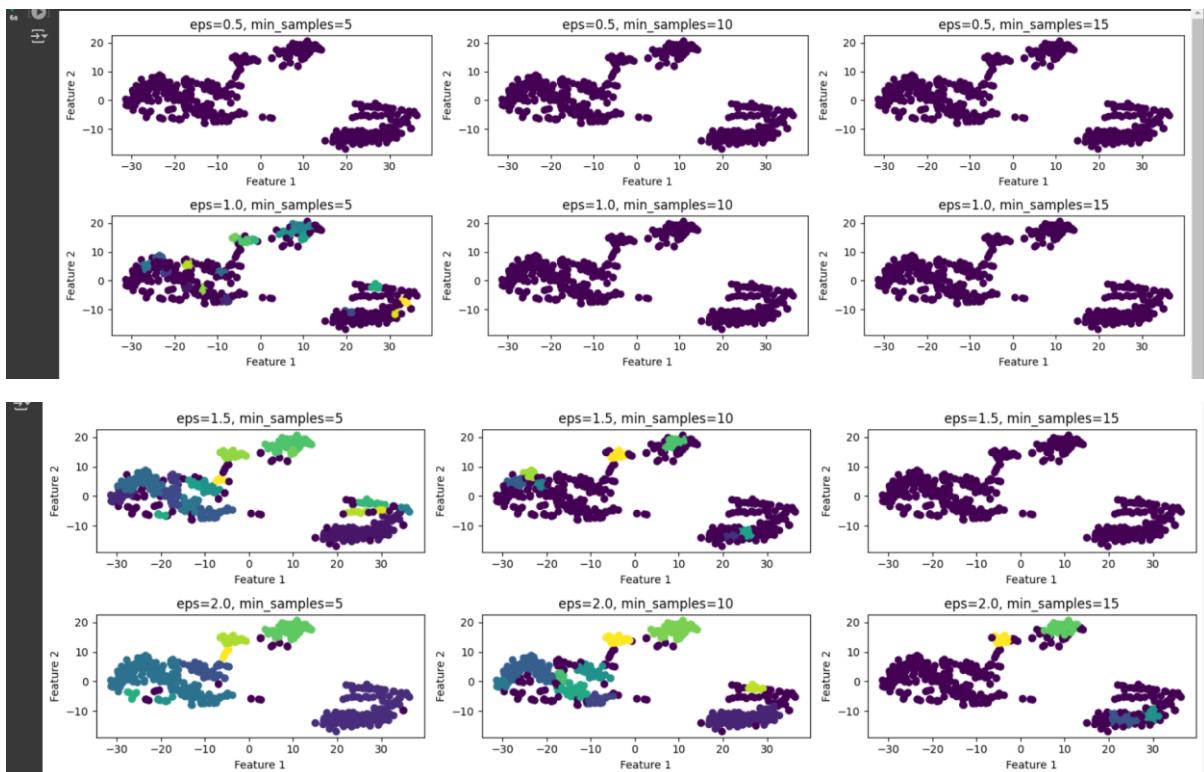
.7

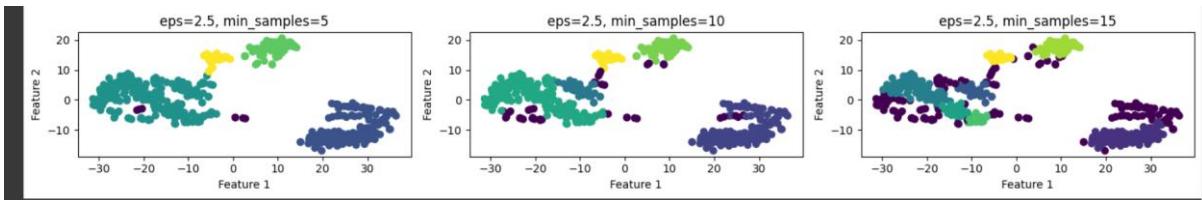
4-7

```

5s  from sklearn.cluster import DBSCAN
6s
7s  eps_values = np.arange(0.5,3,0.5)
8s  min_samples_values = range(5,20,5)
9s
10s fig, axs = plt.subplots(len(eps_values), len(min_samples_values), figsize=(15, 12))
11s
12s for i, eps in enumerate(eps_values):
13s     for j, min_samples in enumerate(min_samples_values):
14s         dbSCAN = DBSCAN(eps=eps, min_samples=min_samples)
15s         cluster_labels = dbSCAN.fit_predict(data_tsne)
16s
17s         axs[i, j].scatter(data_tsne[:, 0], data_tsne[:, 1], c=cluster_labels, cmap='viridis')
18s         axs[i, j].set_title(f'eps={eps}, min_samples={min_samples}')
19s         axs[i, j].set_xlabel('Feature 1')
20s         axs[i, j].set_ylabel('Feature 2')
21s
22s plt.tight_layout()
23s plt.show()

```





```

dbscan_default = DBSCAN()
cluster_labels_default = dbscan_default.fit_predict(data_tsne)
best_avg_points = np.mean(np.bincount(cluster_labels_default + 1))

best_eps = None
best_min_samples = None
best_avg_points = 0

for eps in eps_values:
    for min_samples in min_samples_values:
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
        cluster_labels = dbscan.fit_predict(data_tsne)
        unique_labels, counts = np.unique(cluster_labels, return_counts=True)
        avg_points = np.mean(counts)

        if avg_points > best_avg_points:
            best_avg_points = avg_points
            best_eps = eps
            best_min_samples = min_samples

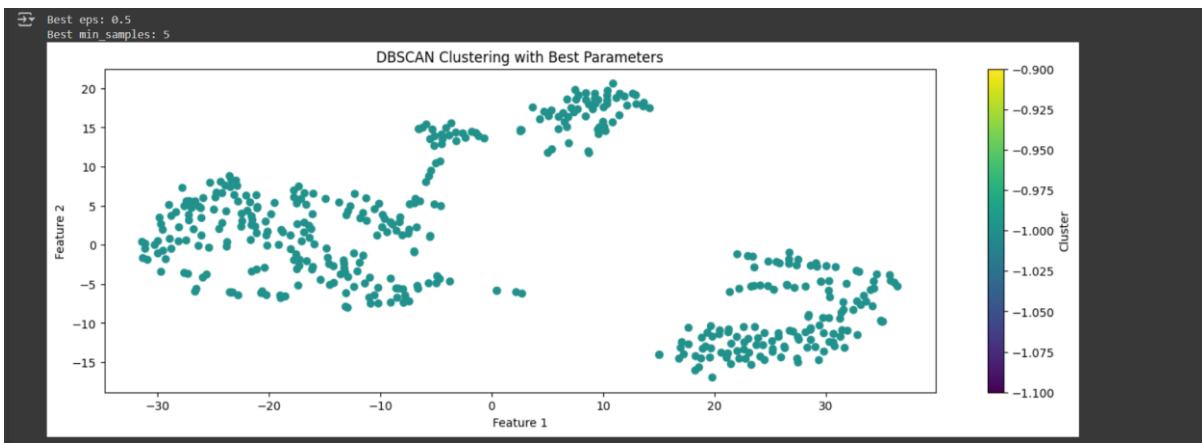
print("Best eps:", best_eps)
print("Best min_samples:", best_min_samples)

dbscan_best = DBSCAN(eps=best_eps, min_samples=best_min_samples)
cluster_labels_best = dbscan_best.fit_predict(data_tsne)

plt.figure(figsize=(16, 5))
plt.scatter(data_tsne[:, 0], data_tsne[:, 1], c=cluster_labels_best, cmap='viridis')
plt.title('DBSCAN Clustering with Best Parameters')
plt.xlabel('Feature 1')

plt.figure(figsize=(16, 5))
plt.scatter(data_tsne[:, 0], data_tsne[:, 1], c=cluster_labels_best, cmap='viridis')
plt.title('DBSCAN Clustering with Best Parameters')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar(label='Cluster')
plt.show()

```



4-8

```
[47] from sklearn.neighbors import NearestNeighbors

[48] nearest_neighbors = NearestNeighbors(n_neighbors=5)
nearest_neighbors.fit(data_tsne)
distances, indices = nearest_neighbors.kneighbors(data_tsne)

distances = np.sort(distances[:, 4], axis=0)

plt.figure(figsize=(16, 5))
plt.plot(distances)
plt.title('k-NN Distance Graph')
plt.xlabel('Points sorted by distance')
plt.ylabel('5th Nearest Neighbor Distance')
plt.show()

best_eps = None
best_min_samples = None
best_avg_points = 0

eps_values = distances[np.linspace(0, len(distances) - 1, num=10).astype(int)]
min_samples_values = range(3, 10)

for eps in eps_values:
    for min_samples in min_samples_values:
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
        cluster_labels = dbscan.fit_predict(data_tsne)

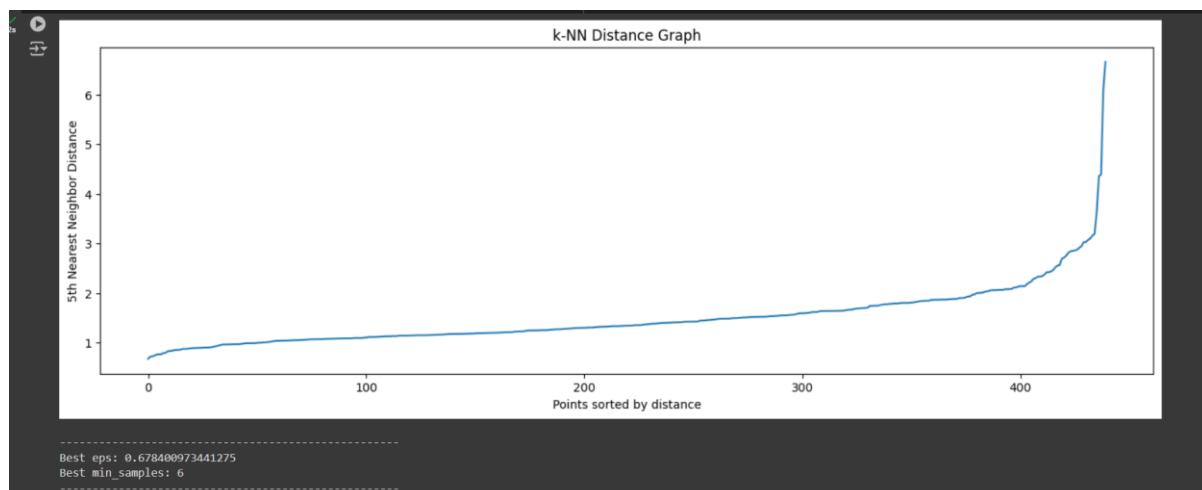
        unique_labels, counts = np.unique(cluster_labels, return_counts=True)
        avg_points = np.mean(counts)

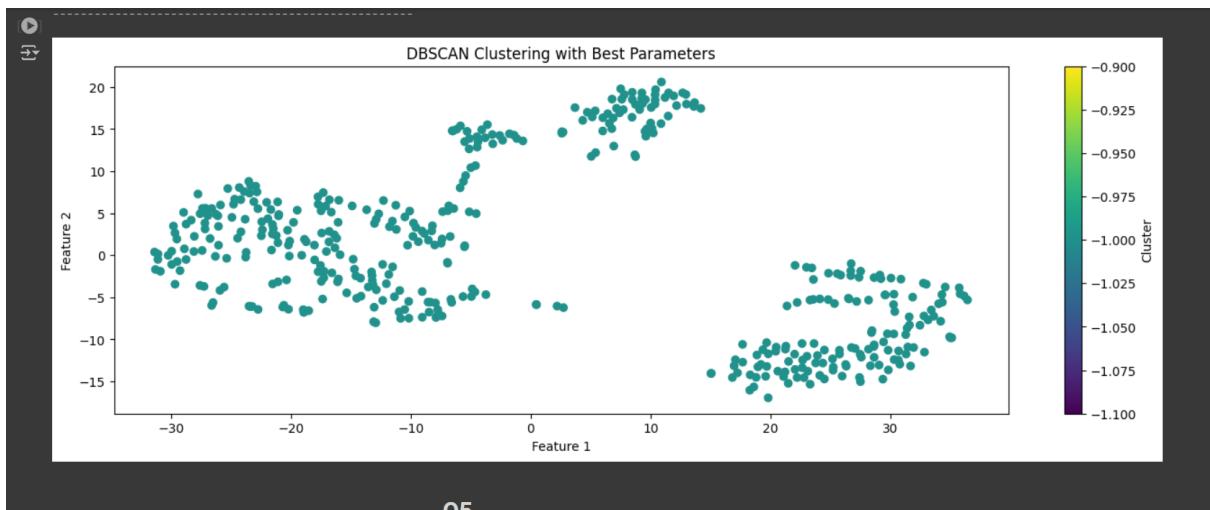
        if avg_points > best_avg_points:
            best_avg_points = avg_points
            best_eps = eps
            best_min_samples = min_samples
```

```
print("\n-----")
print("Best eps:", best_eps)
print("Best min_samples:", best_min_samples)
print("-----\n")

dbscan_best = DBSCAN(eps=best_eps, min_samples=best_min_samples)
cluster_labels_best = dbscan_best.fit_predict(data_tsne)

plt.figure(figsize=(16, 5))
plt.scatter(data_tsne[:, 0], data_tsne[:, 1], c=cluster_labels_best, cmap='viridis')
plt.title('DBSCAN Clustering with Best Parameters')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar(label='Cluster')
plt.show()
```





.5

.1

5-1

```
[103]: datafram2=pd.read_csv('/content/drive/MyDrive/EastWestAirlines.csv')
dataframe.head()
```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

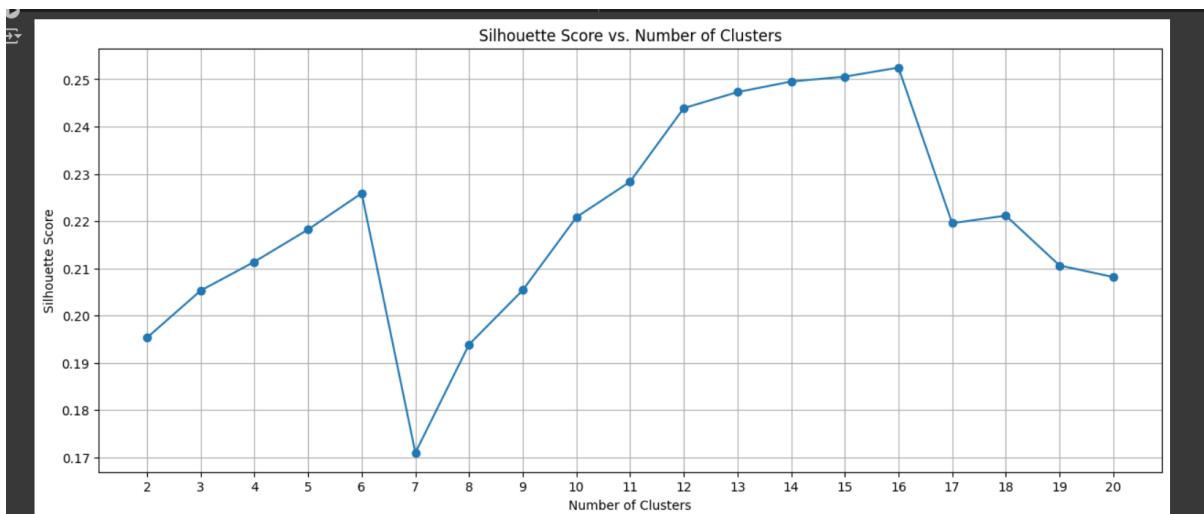
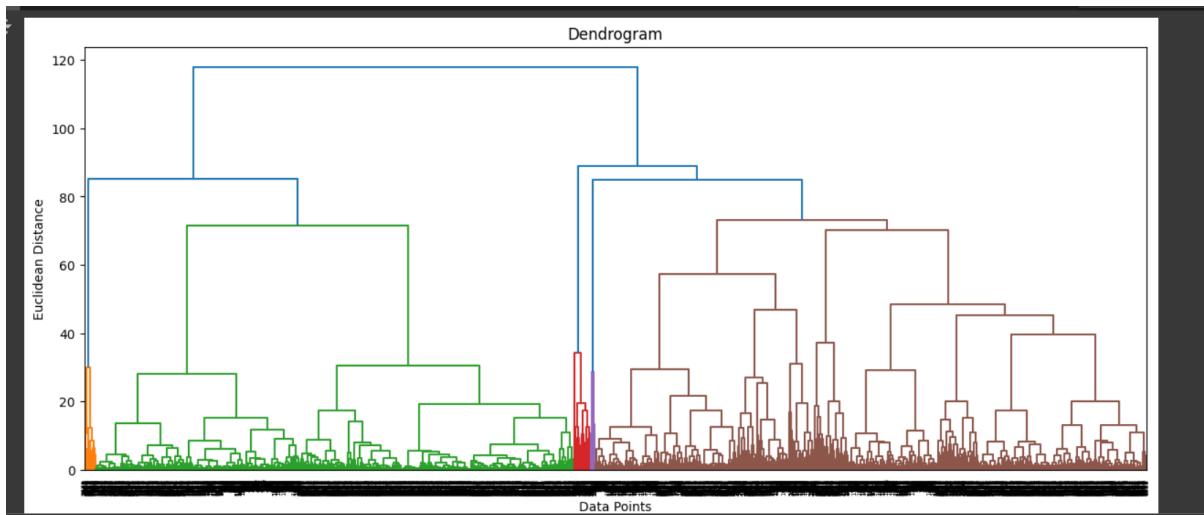
Next steps: [View recommended plots](#)

```
[114]: from sklearn.metrics import silhouette_score
from sklearn.cluster import AgglomerativeClustering
```

```
[115]: scaler = StandardScaler()
data_scaled = scaler.fit_transform(dataframe2)

Z = linkage(data_scaled, method='ward', metric='euclidean')

plt.figure(figsize=(15, 6))
dendrogram(Z)
plt.title('Dendrogram')
plt.xlabel('Data Points')
plt.ylabel('Euclidean Distance')
plt.show()
```



Number of clusters: 16

.2

اگر بدون نرمال سازی داده ها، خوشه بندی انجام می شد ممکن بود مشکلاتی در نتایج به وجود بیاد از قبیل:

مقیاس متغیرها: اگر متغیرها مقیاس های مختلفی داشته باشند می توانند تاثیر متفاوتی در خوشه بندی داشته باشند. برای مثال اگر یکی از ویژگی ها دارای مقیاس بزرگی باشد و دیگری دارای مقیاس کوچک، خوشه بندی ممکن است توسط ویژگی با مقیاس بزرگ تعیین شود و ویژگی های دیگر نادیده گرفته شوند.

تفاوت‌های واریانس: متغیرهایی که واریانس های متفاوتی دارند می‌توانند واریانس در نتایج خوش بندی ایجاد کنند. در این صورت خوش بندی ممکن است تمایل داشته باشد به تشکیل خوش هایی که بیشترین واریانس را دارند.

مقیاس های مختلف: وجود مقیاس های مختلف برای ویژگی ها میتواند منجر به تاثیر بیش از حد بر وزن ویژگی هایی با مقیاس بزرگتر شود و باعث تغییر نتایج خوش بندی شود.

تاثیر پرتو: اگر ویژگی هایی در داده ها وجود داشته باشند که مقادیر آنها بسیار بزرگ یا بسیار کوچک نسبت به مقادیر بقیه ویژگی ها باشد، این ویژگی ها می‌توانند به طور نامناسبی بر خوش بندی تاثیر گذارند.

عدم پایداری: بدون نرمال سازی، الگوریتم های خوش بندی ممکن است به طور ناپایدار عمل کنند و نتایج متفاوتی در هر بار اجرا داشته باشند.

توزيع نرمال: در برخی الگوریتم های خوش بندی، فرض شده است که داده ها از یک توزیع نرمال پیروی می‌کنند یا حداقل باید نرمال شوند. بدون نرمالایز کردن داده ها، این فرض را نقض می کنیم و ممکن است الگوریتم ها به درستی عمل نکنند.

.3

5-3

```
[123] from scipy.cluster.hierarchy import fcluster

max_clusters = 20
best_num_clusters = 16

clustering = AgglomerativeClustering(n_clusters=best_num_clusters, linkage='ward', affinity='euclidean')
cluster_labels = clustering.fit_predict(data_scaled)

cluster_centers = []
for cluster_label in range(best_num_clusters):
    cluster_center = data_scaled[cluster_labels == cluster_label].mean(axis=0)
    cluster_centers.append(cluster_center)

cluster_labels = fcluster(z, t=best_num_clusters, criterion='maxclust')

for i, center in enumerate(cluster_centers):
    print(f"ویژگی های مرکز خوش {i+1}: \n{center}")
    print(f"برحسب خوش {i+1}: \n{cluster_labels[i]}")
    print("-----")
```

رویزگی‌های مرکز خوشه 1:

$$[1.05090618 \quad -0.37591741 \quad -0.16795597 \quad -0.57353448 \quad -0.09824189 \quad -0.06276658 \\ -0.51624148 \quad -0.50461177 \quad -0.25234575 \quad -0.26540554 \quad -1.05540263 \quad -0.7669193]$$

برچسب خوشه 1:
2

رویزگی‌های مرکز خوشه 2:

$$[0.21932485 \quad -0.04688966 \quad -0.1562356 \quad -0.66822727 \quad 9.03825361 \quad -0.06276658 \\ -0.10166533 \quad 0.61785114 \quad 0.0875494 \quad 0.22034681 \quad -0.07246398 \quad 0.05178388]$$

برچسب خوشه 2:
2

رویزگی‌های مرکز خوشه 3:

$$[-0.65223354 \quad -0.31796521 \quad -0.18233325 \quad -0.7195829 \quad -0.09824189 \quad -0.06276658 \\ -0.60412923 \quad -0.63070834 \quad -0.24071579 \quad -0.25498995 \quad 0.6320257 \quad -0.7669193]$$

برچسب خوشه 3:
2

رویزگی‌های مرکز خوشه 4:

$$[-0.48014468 \quad 0.33774024 \quad -0.17904594 \quad 0.9578827 \quad -0.09824189 \quad -0.06276658 \\ 0.48650198 \quad 0.57111995 \quad -0.22688958 \quad -0.24564197 \quad 0.47427393 \quad -0.7669193]$$

برچسب خوشه 4:
2

رویزگی‌های مرکز خوشه 5:

$$[-1.15845877 \quad -0.08926323 \quad -0.17064462 \quad 0.2705201 \quad -0.09824189 \quad -0.06276658 \\ 0.0879954 \quad 0.22539412 \quad -0.14808356 \quad -0.16012907 \quad 1.14985887 \quad 1.30391816]$$

برچسب خوشه 5:
13

رویزگی‌های مرکز خوشه 6:

$$[0.22607501 \quad 0.14738844 \quad 0.01671559 \quad -0.25410436 \quad -0.09824189 \quad -0.06276658 \\ -0.1207324 \quad 0.45935459 \quad 1.76269144 \quad 1.62807413 \quad -0.19071581 \quad 0.58246511]$$

برچسب خوشه 6:
2

رویزگی‌های مرکز خوشه 7:

$$[-0.3059493 \quad 0.8769957 \quad 0.57504338 \quad 0.32610178 \quad -0.09824189 \quad -0.06276658 \\ 0.99493949 \quad 2.09951383 \quad 4.91973141 \quad 5.07647186 \quad 0.27433002 \quad 1.02312664]$$

برچسب خوشه 7:
7

رویزگی‌های مرکز خوشه 8:

$$[0.32529244 \quad -0.15580117 \quad -0.1739266 \quad 0.89036224 \quad -0.09824189 \quad -0.06276658 \\ 0.46926448 \quad 0.595519 \quad -0.23395776 \quad -0.2489601 \quad -0.30999011 \quad 1.30391816]$$

برچسب خوشه 8:
13

رویزگی‌های مرکز خوشه 9:

$$[-0.99012042 \quad 2.93749694 \quad -0.16963131 \quad 0.6734365 \quad -0.09824189 \quad -0.06276658 \\ 0.79236024 \quad 0.21224434 \quad 0.04388291 \quad 0.06672645 \quad 0.99009027 \quad 0.66886134]$$

برچسب خوشه 9:
1

رویزگی‌های مرکز خوشه 10:

$$[-2.34680769e-01 \quad 5.59233305e-01 \quad -1.01410550e-01 \quad 9.65590528e-01 \\ -9.82418871e-02 \quad 1.38818752e+01 \quad 2.86214972e+00 \quad 1.52253632e+00 \\ -2.70233003e-02 \quad -1.06151515e-02 \quad 1.79293251e-01 \quad 2.68499430e-01]$$

برچسب خوشه 10:
13

رویزگی‌های مرکز خوشه 11:

$$[-0.61937242 \quad 0.55843919 \quad -0.13790858 \quad 1.90368833 \quad -0.09824189 \quad -0.06276658 \\ 2.4386943 \quad 1.48490098 \quad 0.36058835 \quad 0.51852605 \quad 0.6307746 \quad 1.17582512]$$

برچسب خوشه 11:
7

رویزگی‌های مرکز خوشه 12:

$$[-0.072268 \quad 0.1271357 \quad 2.84906345 \quad 0.07918988 \quad -0.09824189 \quad -0.06276658 \\ 0.20162352 \quad -0.02523792 \quad -0.00467054 \quad -0.01259017 \quad 0.06797596 \quad 0.53607955]$$

برچسب خوشه 12:
8

رویزگی‌های مرکز خوشه 13:

$$[0.5640066 \quad -0.28008674 \quad -0.14859376 \quad -0.71287452 \quad -0.09824189 \quad -0.06276658 \\ -0.50429672 \quad -0.6445395 \quad -0.09377626 \quad -0.07595952 \quad -0.54245123 \quad 1.30391816]$$

برچسب خوشه 13:
7

رویزگی‌های مرکز خوشه 14:

$$[-0.18809551 \quad 0.79387166 \quad 7.70716294 \quad -0.04322862 \quad -0.09824189 \quad -0.06276658 \\ 0.03775436 \quad 0.24465423 \quad 0.41209299 \quad 0.58316596 \quad 0.18607463 \quad 0.39477001]$$

برچسب خوشه 14:
2

رویزگی‌های مرکز خوشه 15:

$$[0.95912641 \quad 0.57955932 \quad 0.26227263 \quad 0.31994627 \quad -0.09824189 \quad -0.06276658 \\ 2.00801342 \quad 6.00337852 \quad 13.92819244 \quad 12.62331067 \quad -0.92901903 \quad 1.30391816]$$

برچسب خوشه 15:
2

رویزگی‌های مرکز خوشه 16:

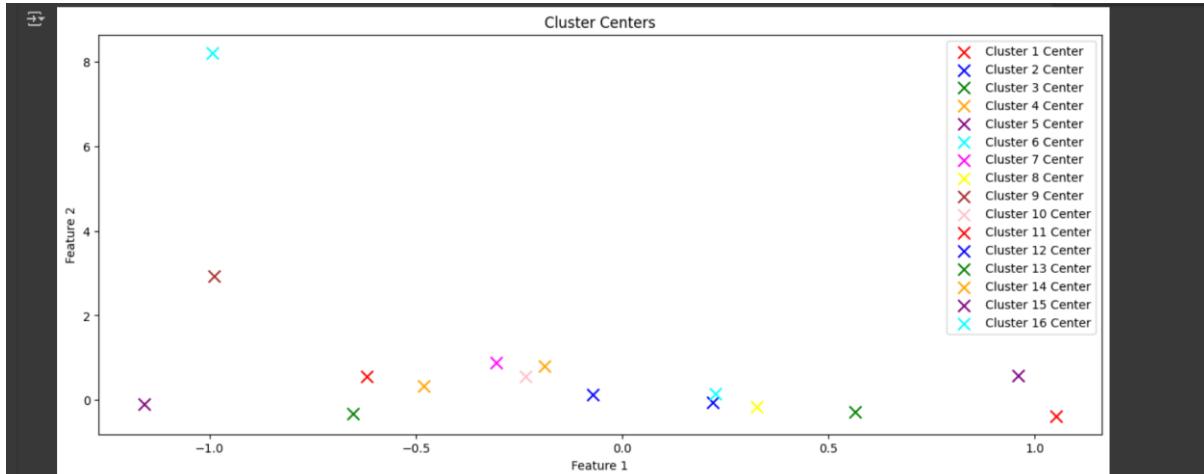
$$[0.99359709 \quad 8.2086291 \quad 0.54374382 \quad 0.91249478 \quad -0.09824189 \quad -0.06276658 \\ 1.39115214 \quad 1.13765069 \quad 0.73501746 \quad 1.33083601 \quad 1.21002205 \quad 1.19492671]$$

برچسب خوشه 16:
13

```
import matplotlib.pyplot as plt

colors = ['red', 'blue', 'green', 'orange', 'purple', 'cyan', 'magenta', 'yellow', 'brown', 'pink']

plt.figure(figsize=(15, 6))
for i, center in enumerate(cluster_centers):
    plt.scatter(center[0], center[1], label=f'Cluster {i+1} Center', marker='x', s=100, c=colors[i % len(colors)])
plt.title('Cluster Centers')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```



.4

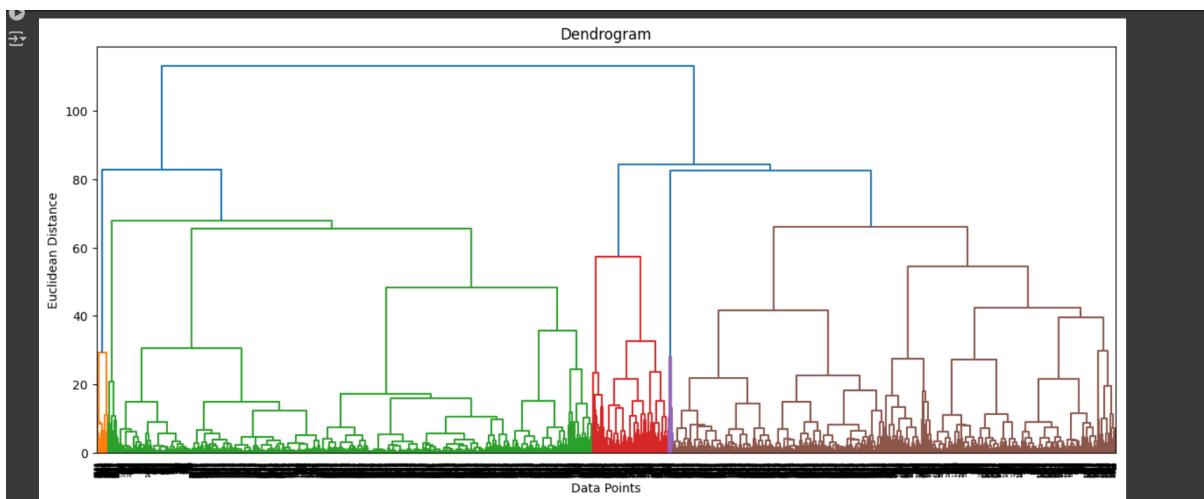
5-4

```
47s  data_random_5percent = dataframe2.sample(frac=0.05, random_state=42)
data_remainder_95percent = dataframe2.drop(data_random_5percent.index)

scaler = StandardScaler()
data_scaled2 = scaler.fit_transform(data_remainder_95percent)

Z = linkage(data_scaled2, method='ward', metric='euclidean')

plt.figure(figsize=(15, 6))
dendrogram(Z)
plt.title('Dendrogram')
plt.xlabel('Data Points')
plt.ylabel('Euclidean Distance')
plt.show()
```



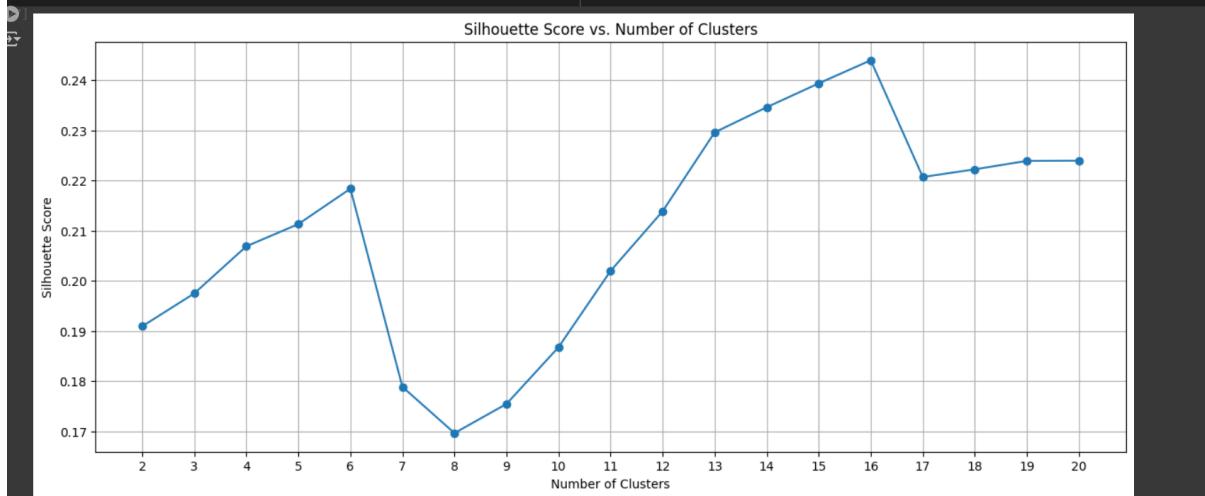
```

20s
max_clusters = 20
silhouette_scores = []

for n_clusters in range(2, max_clusters+1):
    clustering = AgglomerativeClustering(n_clusters=n_clusters, linkage='ward', affinity='euclidean')
    cluster_labels = clustering.fit_predict(data_scaled2)
    silhouette_avg = silhouette_score(data_scaled2, cluster_labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(15, 6))
plt.plot(range(2, max_clusters+1), silhouette_scores, marker='o')
plt.title('Silhouette Score vs. Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.xticks(range(2, max_clusters+1))
plt.grid(True)
plt.show()

```



```

max_clusters = 20
best_num_clusters = 16

clustering = AgglomerativeClustering(n_clusters=best_num_clusters, linkage='ward', affinity='euclidean')
cluster_labels = clustering.fit_predict(data_scaled2)

cluster_centers2 = []
for cluster_label in range(best_num_clusters):
    cluster_center = data_scaled2[cluster_labels == cluster_label].mean(axis=0)
    cluster_centers2.append(cluster_center)

cluster_labels = fcluster(Z, t=best_num_clusters, criterion='maxclust')

for i, center in enumerate(cluster_centers2):
    print(f"پوزیگی‌های مرکز خوشه {i+1}: \n{center}")
    print(f"برچسب خوشه {i+1}: \n{cluster_labels[i]}")
    print("-----")

```

```

1 پوزیگی‌های مرکز خوشه 1:
[-0.62713017 -0.37218069 -0.18532139 -0.72250169 -0.09963817 -0.06440426
-0.59899349 -0.62252038 -0.26259379 -0.27662063 0.60493358 -0.76911475]
1 برچسب خوشه 1:
-----
2 پوزیگی‌های مرکز خوشه 2:
[ 0.26429906 -0.14003757 -0.155462 -0.70243537 8.91287146 -0.06440426
-0.093040412 0.56044641 0.04099322 0.16819664 -0.10709165 0.01919415]
2 برچسب خوشه 2:
-----
3 پوزیگی‌های مرکز خوشه 3:
[-8.86258602e-01 3.72219494e+00 2.24288326e-02 -2.00465769e-03
-9.96381691e-02 -6.44042551e-02 2.73213601e-02 -1.60564654e-03
1.63009198e-01 2.92842398e-01 8.95632153e-01 4.85013043e-01]
3 برچسب خوشه 3:
-----
4 پوزیگی‌های مرکز خوشه 4:
[ 0.07018818 0.20443449 2.57163036 -0.44573996 -0.09963817 -0.06440426
-0.23694744 -0.09912697 0.42948039 0.54952261 -0.05220495 0.53292354]
4 برچسب خوشه 4:
-----
5 پوزیگی‌های مرکز خوشه 5:
[-2.35896173e-01 5.49942290e-01 -1.01467354e-01 9.59816053e-01
-9.96381691e-02 1.35284716e+01 2.82745769e+00 1.53353680e+00
-2.13214833e-02 -4.79304169e-03 1.78980916e-01 2.65540679e-01]
5 برچسب خوشه 5:
14

```

```

2s
6: وزیرگی‌های مرکز خوشه 6:
[ -1.14310057 -0.08877565 -0.12556247 0.19848443 -0.09963817 -0.06440426
 0.03201977 0.11810566 -0.14395686 -0.15790352 1.1341255 1.28769272]
7: برجسب خوشه 6:
3

-----
```

7: وزیرگی‌های مرکز خوشه 7:

```
[ 0.62464036 0.06103525 -0.17836069 0.60410195 -0.09963817 -0.06440426
 0.20938999 0.60020085 -0.26552794 -0.27933096 -0.61357199 -0.72893395]
```

7: برجسب خوشه 7:

```
15
```

8: وزیرگی‌های مرکز خوشه 8:

```
[ -0.13632537 0.84902589 -0.01457682 1.36646372 -0.09963817 -0.06440426
 1.3971887 2.14635107 1.63380546 1.74544604 0.16644406 0.36900622]
```

8: برجسب خوشه 8:

```
14
```

9: وزیرگی‌های مرکز خوشه 9:

```
[ -0.0314103 0.83045276 0.5837291 0.08809379 -0.09963817 -0.06440426
 0.74297393 2.39904285 6.29027578 6.3324998 -0.0111352 1.01145506]
```

9: برجسب خوشه 9:

```
14
```

10: وزیرگی‌های مرکز خوشه 10:

```
[ -0.74396134 0.97269767 0.09040202 2.03078912 -0.09963817 -0.06440426
 2.64796151 1.11820753 0.02598238 0.05949078 0.76123033 1.30019611]
```

10: برجسب خوشه 10:

```
3
```

```

11: وزیرگی‌های مرکز خوشه 11:
[ 1.11476737 -0.44467734 -0.18394491 -0.77145619 -0.09963817 -0.06440426
-0.63085511 -0.76225224 -0.27124104 -0.277159 -1.12199243 -0.76911475]
11: برجسب خوشه 11:
13

-----
```

12: وزیرگی‌های مرکز خوشه 12:

```
[ 2.22383597e-01 2.55183576e-01 7.52987128e+00 -8.29735092e-02
-9.96381691e-02 -6.44042551e-02 -6.46557787e-02 -8.64800393e-02
-5.01341766e-02 -7.03550047e-03 -2.74381159e-01 6.87682272e-03]
```

12: برجسب خوشه 12:

```
3
```

13: وزیرگی‌های مرکز خوشه 13:

```
[ 0.41569577 -0.16817753 -0.18100888 0.87563157 -0.09963817 -0.06440426
0.45114025 0.52957087 -0.2174191 -0.2268804 -0.38987445 1.30019611]
```

13: برجسب خوشه 13:

```
3
```

14: وزیرگی‌های مرکز خوشه 14:

```
[ -0.9920122 0.47778206 -0.18392834 0.97342382 -0.09963817 -0.06440426
0.49797279 0.44169577 -0.27380577 -0.29889112 0.98060797 -0.76911475]
```

14: برجسب خوشه 14:

```
14
```

15: وزیرگی‌های مرکز خوشه 15:

```
[ 0.51166787 -0.25783668 -0.17856483 -0.76800515 -0.09963817 -0.06440426
-0.54125184 -0.78775329 -0.15877063 -0.11499387 -0.50720433 1.30019611]
```

15: برجسب خوشه 15:

```
14
```

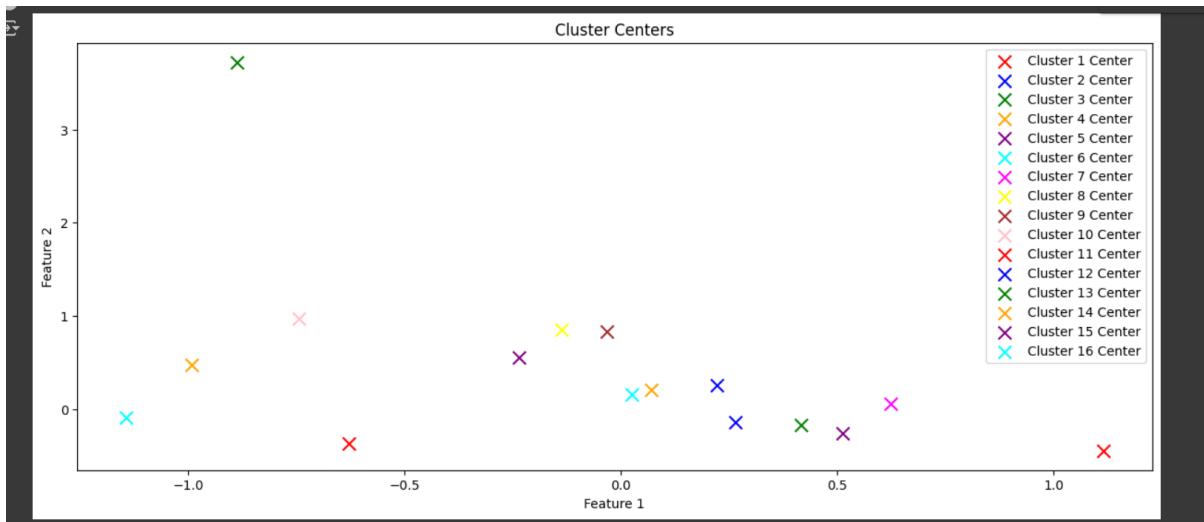
```

16: وزیرگی‌های مرکز خوشه 16:
[ 0.02592389 0.15980406 -0.121393 -0.4706296 -0.09963817 -0.06440426
-0.24075649 0.33217476 1.80981884 1.7045529 -0.00332625 0.45425142]
16: برجسب خوشه 16:
14
```

```

17: colors = ['red', 'blue', 'green', 'orange', 'purple', 'cyan', 'magenta', 'yellow', 'brown', 'pink']

plt.figure(figsize=(15, 6))
for i, center in enumerate(cluster_centers2):
    plt.scatter(center[0], center[1], label=f'Cluster {i+1} Center', marker='x', s=100, c=colors[i % len(colors)])
plt.title('Cluster Centers')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```



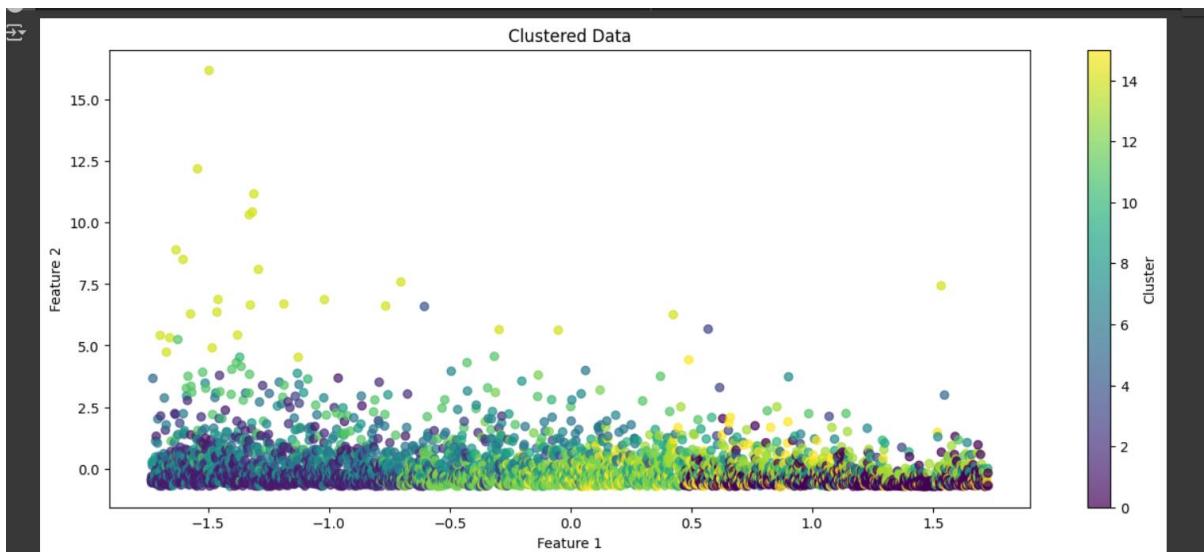
ترکیب خوش بندی یکسان شده است ولی مراکز خوش ها فرق کرده است

.5

5-5

```
[131] from sklearn.cluster import KMeans
[132] num_clusters = 16
[133] kmeans = KMeans(n_clusters=num_clusters, random_state=42)
[134] cluster_labels = kmeans.fit_predict(data_scaled)

[135] plt.figure(figsize=(15, 6))
[136] plt.scatter(data_scaled[:, 0], data_scaled[:, 1], c=cluster_labels, cmap='viridis', marker='o', alpha=0.7)
[137] plt.title('Clustered Data')
[138] plt.xlabel('Feature 1')
[139] plt.ylabel('Feature 2')
[140] plt.colorbar(label='cluster')
[141] plt.show()
```



```
▶ num_clusters = 16

kmeans = KMeans(n_clusters=num_clusters, random_state=42)
cluster_labels = kmeans.fit_predict(data_scaled)

cluster_centers = kmeans.cluster_centers_

for i, center in enumerate(cluster_centers):
    print(f"\tوبیزگی‌های مرکز خوش {i+1}: {center}")
    print("-----")
```

```
→ 1: ووبیزگی‌های مرکز خوش 1:
[ 1.1728225 -0.45477502 -0.13114863 -0.74753877 -0.09824189 -0.06276658
-0.61661939 -0.72749032 -0.24385692 -0.26094573 -1.16268952 -0.7669193 ]-----
```

```
→ 2: ووبیزگی‌های مرکز خوش 2:
[-1.21927046 -0.22173736 -0.15356178 -0.66814302 -0.09824189 -0.04999221
-0.58277297 -0.55298822 -0.22831714 -0.24907485 1.22910357 -0.7669193 ]-----
```

```
→ 3: ووبیزگی‌های مرکز خوش 3:
[-1.02881219 -0.0389872 -0.03888866 -0.30902223 -0.09824189 -0.06276658
-0.30154899 -0.17503209 -0.10826162 -0.11430657 1.01403202 1.30391816]-----
```

```
→ 4: ووبیزگی‌های مرکز خوش 4:
[-0.06539659 0.90442565 0.68960154 0.29049966 -0.09824189 -0.06276658
1.07398045 2.79127304 6.87710931 6.58574903 0.03522628 1.191981 ]-----
```

```
→ 5: ووبیزگی‌های مرکز خوش 5:
[-0.30152248 0.63971926 -0.08443292 1.0220844 -0.09824189 15.64629931
3.17969131 1.71461374 0.03329269 0.05969539 0.23987261 0.33752735]-----
```

```
→ 6: ووبیزگی‌های مرکز خوش 6:
[ 0.21932485 -0.04688966 -0.1562356 -0.66822727 9.03825361 -0.06276658
-0.10166533 0.61785114 0.0875494 0.22034681 -0.07246398 0.05178388]-----
```

```
→ 7: ووبیزگی‌های مرکز خوش 7:
[-0.82439127 0.54221846 -0.15999921 1.12200321 -0.09824189 -0.06276658
0.57356733 0.50992274 -0.20531752 -0.23630987 0.81062892 -0.7669193 ]-----
```

```
→ 8: ووبیزگی‌های مرکز خوش 8:
[ 0.01916407 0.37510804 0.16911763 -0.23227856 -0.09824189 -0.06276658
0.06671387 0.91878745 2.33739836 2.53994798 -0.01568859 0.69401398]
```

```
→ 9: ووبیزگی‌های مرکز خوش 9:
[-0.9665507 0.20817222 -0.09049007 1.5702864 -0.09824189 -0.06276658
1.24503985 0.80462637 -0.10637356 -0.11504317 0.95308087 1.30391816]-----
```

```
→ 10: ووبیزگی‌های مرکز خوش 10:
[ 0.55756592 -0.1467829 -0.0978408 1.02493283 -0.09824189 -0.04602631
0.55734137 0.64722959 -0.14648584 -0.17002018 -0.52130419 1.30391816]-----
```

```
→ 11: ووبیزگی‌های مرکز خوش 11:
[ 0.03660825 0.43216289 7.2878929 -0.05719688 -0.09824189 -0.06276658
0.058050676 0.07149734 0.3419436 0.39333244 -0.07623932 0.26849943]-----
```

```
→ 12: ووبیزگی‌های مرکز خوش 12:
[-0.69500337 1.58520568 0.08116118 2.00803698 -0.09824189 -0.06276658
3.28892243 1.63246015 0.48723666 0.61436746 0.7398057 0.90125532]-----
```

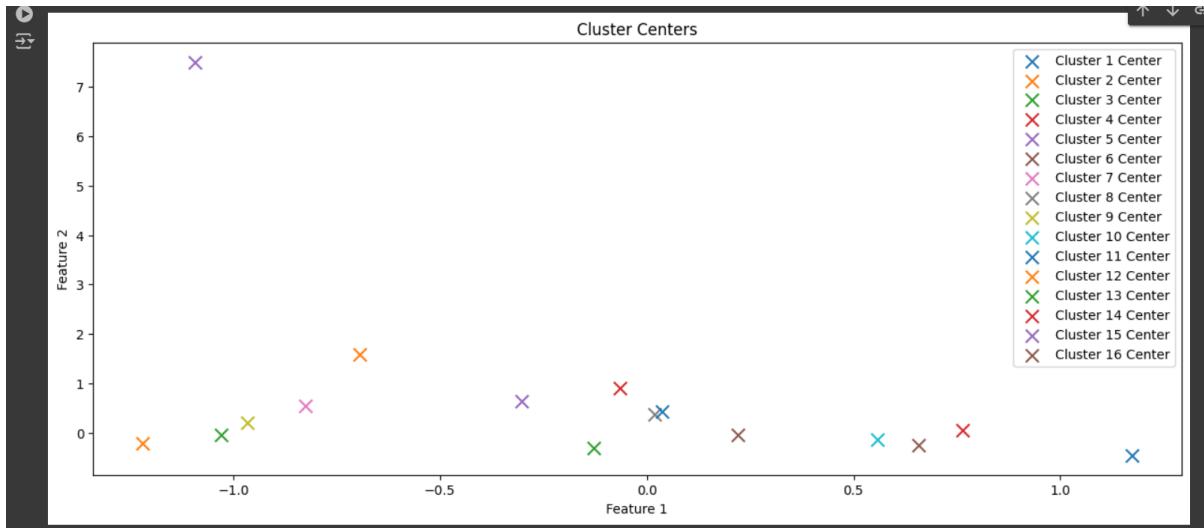
```
→ 13: ووبیزگی‌های مرکز خوش 13:
[-0.12735004 -0.31668372 -0.14590259 -0.71543233 -0.09824189 -0.06276658
-0.59483439 -0.64055415 -0.22677717 -0.23848523 0.06907245 -0.7669193 ]-----
```

```
→ 14: ووبیزگی‌های مرکز خوش 14:
[ 0.7645114 0.04286025 -0.15515876 0.59362821 -0.09824189 -0.04808887
0.20888849 0.68478781 -0.23757191 -0.26924312 -0.75432788 -0.7669193 ]-----
```

```
→ 15: ووبیزگی‌های مرکز خوش 15:
[-1.09289491 7.50055461 0.34719392 0.96248647 -0.09824189 -0.06276658
1.45753744 1.01074524 0.65738922 1.11840993 1.25016718 0.98532778]-----
```

```
→ 16: ووبیزگی‌های مرکز خوش 16:
[ 0.65663903 -0.24310958 -0.08252382 -0.67155888 -0.09824189 -0.06276658
-0.46848766 -0.48067573 -0.09516356 -0.07022098 -0.63234375 1.30391816]
```

```
plt.figure(figsize=(15, 6))
for i, center in enumerate(cluster_centers):
    plt.scatter(center[0], center[1], label=f'Cluster {i+1} Center', marker='x', s=100)
plt.title('Cluster Centers')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```

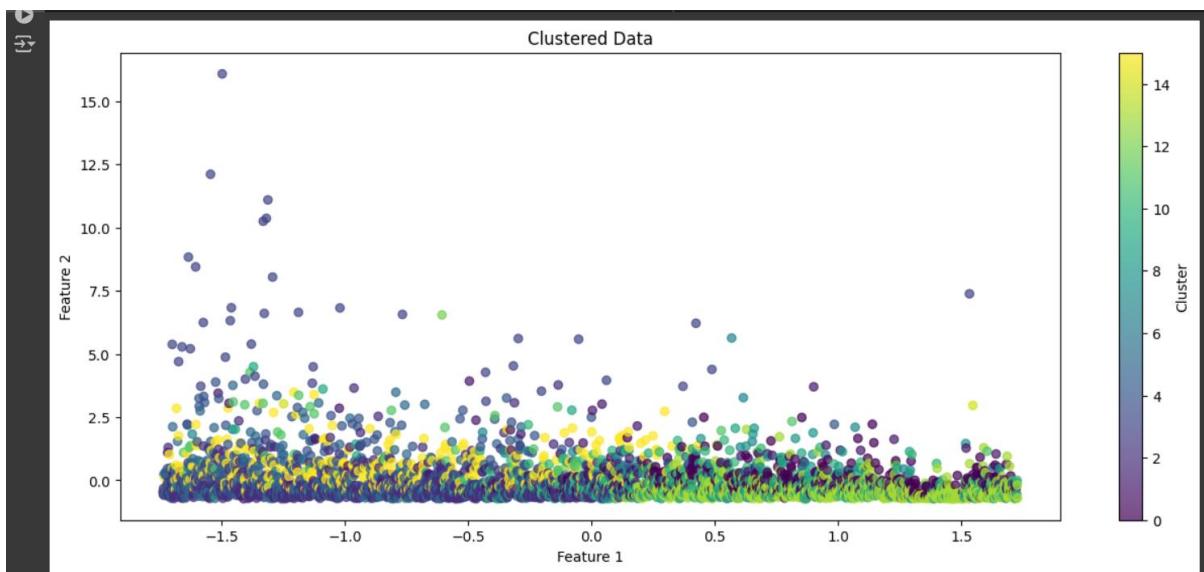


The second mode

```
num_clusters = 16

kmeans = KMeans(n_clusters=num_clusters, random_state=42)
cluster_labels = kmeans.fit_predict(data_scaled2)

plt.figure(figsize=(15, 6))
plt.scatter(data_scaled2[:, 0], data_scaled2[:, 1], c=cluster_labels, cmap='viridis', marker='o', alpha=0.7)
plt.title('Clustered Data')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar(label='Cluster')
plt.show()
```



```
0s num_clusters = 16

kmeans = KMeans(n_clusters=num_clusters, random_state=42)
cluster_labels = kmeans.fit_predict(data_scaled2)

cluster_centers2 = kmeans.cluster_centers_

for i, center in enumerate(cluster_centers2):
    print(f"ویژگی‌های مرکز خوش {i+1}: \n{center}")
    print("-----")
```

```
0s
→ 1: ویژگی‌های مرکز خوش 1:
[ 0.77565313  0.01250103 -0.14374539  0.4870495 -0.09963817 -0.05025893
  0.13934375  0.70436792 -0.24928334 -0.26829783 -0.76553117 -0.76911475]
-----
→ 2: ویژگی‌های مرکز خوش 2:
[ 0.15431282  0.34281071  0.15700793 -0.5166723 -0.09963817 -0.06440426
 -0.18292971  0.57855941  2.27067028  2.37361481 -0.1289312  0.60503699]
-----
→ 3: ویژگی‌های مرکز خوش 3:
[-0.76311158 -0.27304534 -0.14472885 -0.70602332 -0.09963817 -0.05757347
 -0.5926409 -0.61720189 -0.22987888 -0.23842596  0.73571371 -0.76911475]
-----
→ 4: ویژگی‌های مرکز خوش 4:
[-0.95399603  6.33656794  0.18791692  1.06972083 -0.09963817 -0.06440426
  1.59030786  0.97825518  0.56926095  0.94552463  1.06927417  0.96463218]
-----
→ 5: ویژگی‌های مرکز خوش 5:
[-0.87960409  0.53611626 -0.04932768  1.82184661 -0.09963817 -0.06440426
  1.99737541  1.0273765 -0.031313875 -0.03123397  0.87566301  1.16893385]
-----
→ 6: ویژگی‌های مرکز خوش 6:
[-1.08012881 -0.0147024 -0.06633863 -0.15566768 -0.09963817 -0.06440426
 -0.22296171 -0.03750001 -0.11226592 -0.11925474  1.06819053  1.30019611]
-----
→ 7: ویژگی‌های مرکز خوش 7:
[ 0.16853096 -0.21632308 -0.10017738 -0.77145619 13.18195497 -0.06440426
  0.12877858  0.77035472  0.1826785  0.49751773 -0.02112461  0.05860959]
-----
→ 8: ویژگی‌های مرکز خوش 8:
[-0.01087736  0.80933654  0.84232099 -0.04673758 -0.09963817 -0.06440426
  0.68104341  2.41422099  7.33749611  6.65082464 -0.03728177  1.09326502]
```

```

▶ 9: ویژگی‌های مرکز خوشه 9:
[ 0.60690117 -0.26738015 -0.07854457 -0.67193497 -0.09963817 -0.06440426
-0.47153676 -0.5170972 -0.12180754 -0.10301905 -0.58766818 1.30019611]

▶ 10: ویژگی‌های مرکز خوشه 10:
[ 0.05184306 0.42678398 7.31593976 -0.1071308 -0.09963817 -0.06440426
-0.01393079 0.02476887 0.27692059 0.31475734 -0.08660266 0.26554068]

▶ 11: ویژگی‌های مرکز خوشه 11:
[ 0.46340995 -0.14024463 -0.08330733 1.05748668 -0.09963817 -0.04865252
0.5580764 0.63893966 -0.16279327 -0.17683278 -0.43776604 1.30019611]

▶ 12: ویژگی‌های مرکز خوشه 12:
[-0.67209562 1.22392037 0.28329091 1.68628519 -0.09963817 -0.06440426
2.63560587 2.67567965 2.87553517 3.14634905 0.72172134 1.16524105]

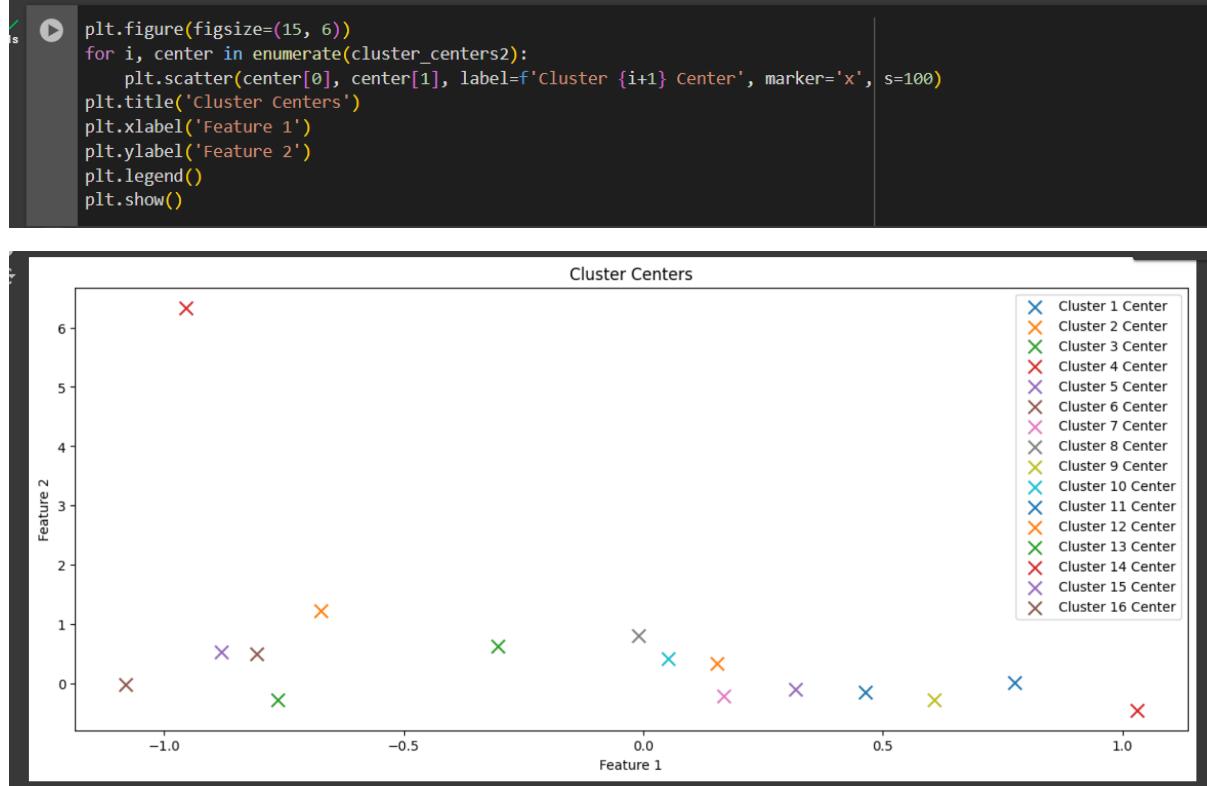
▶ 13: ویژگی‌های مرکز خوشه 13:
[-0.30271374 0.62990139 -0.08452569 1.01618306 -0.09963817 15.2483865
3.14171119 1.72665909 0.04237613 0.06696564 0.23955276 0.33451771]

▶ 14: ویژگی‌های مرکز خوشه 14:
[ 1.02990277 -0.44600619 -0.13533342 -0.75327329 -0.09963817 -0.06440426
-0.61893827 -0.7669678 -0.2629921 -0.26617211 -1.03196018 -0.76911475]

▶ 15: ویژگی‌های مرکز خوشه 15:
[ 3.17503563e-01 -9.76567281e-02 -1.86175680e-01 -6.64090472e-01
6.54115840e+00 -6.44042551e-02 -2.16278948e-01 4.43830679e-01
-3.77208249e-02 -1.47595252e-02 -1.54851122e-01 -2.70332012e-03]

▶ 16: ویژگی‌های مرکز خوشه 16:
[-0.80612581 0.49564394 -0.16422697 1.0758236 -0.09963817 -0.06440426
0.52472972 0.50236828 -0.21340043 -0.2430127 0.79780137 -0.76911475]

```



ترکیب خوشه بندی یکسان شده است ولی مراکز خوشه ها فرق کرده است

5-6

```

k_values = range(1, 26)

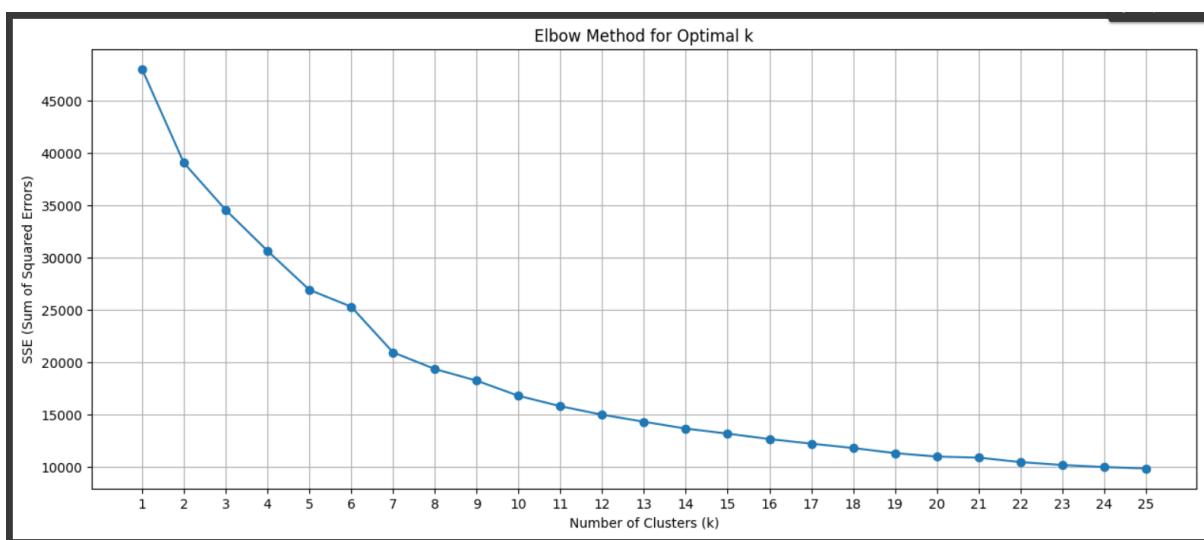
sse = []

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(15, 6))
plt.plot(k_values, sse, marker='o', linestyle='-' )
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('SSE (Sum of Squared Errors)')
plt.xticks(range(1, 26))
plt.grid(True)
plt.show()

# slope_changes = [sse[i] - sse[i-1] for i in range(1, len(sse))]
# elbow_index = slope_changes.index(max(slope_changes)) + 1
# best_k = elbow_index + 1
# print(f"\nThe optimal value of k is: {best_k}")
# -----
# elbow_index = sse.index(min(sse)) + 1
# best_k = elbow_index
# print(f"The optimal value of k is: {best_k}")

```



K = این نقطه جایی است که کاهش SSE با افزایش تعداد خوشه‌ها به صورت چشمگیری کاهش پیدا می‌کند و پس از آن کاهش SSE کمتر احساس می‌شود --> کاهش قابل توجهی در SSE تا حدود K=7 or K=8 مشاهده می‌شود و پس از آن کاهش به تدریج کمتر دیده می‌شود. این به این معنی است که Elbow Point در حدود مقادیر 7 یا 8 قرار دارد.

5-7

```
[43] from sklearn.metrics import silhouette_score

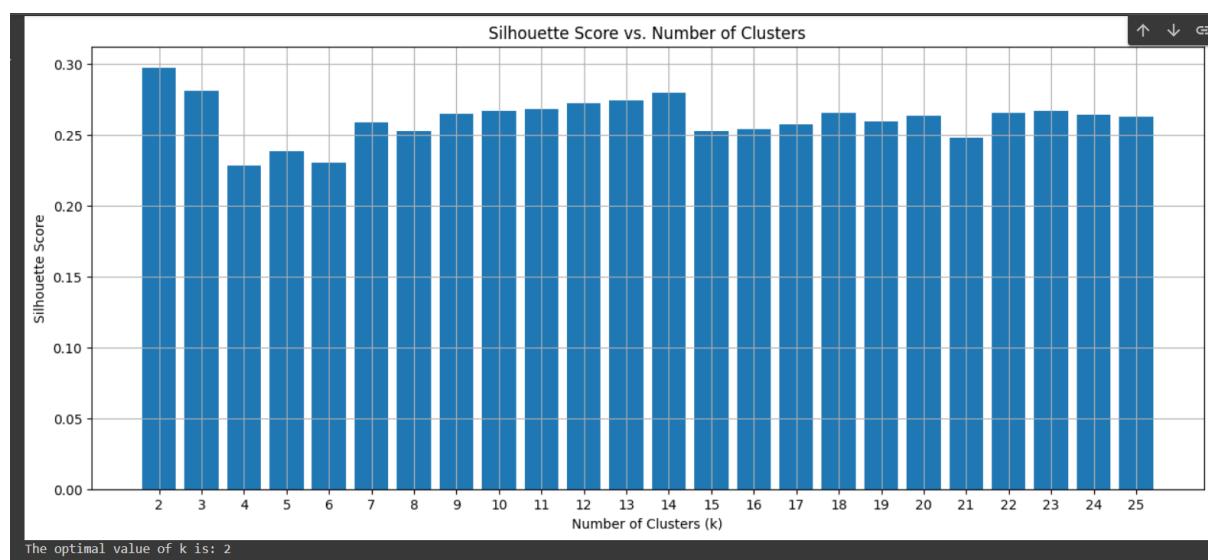
[43] k_values = range(2, 26)

[43] silhouette_scores = []

[43] for k in k_values:
[43]     kmeans = KMeans(n_clusters=k, random_state=42)
[43]     cluster_labels = kmeans.fit_predict(data_scaled)
[43]     silhouette_avg = silhouette_score(data_scaled, cluster_labels)
[43]     silhouette_scores.append(silhouette_avg)

[43] plt.figure(figsize=(15, 6))
[43] plt.bar(k_values, silhouette_scores)
[43] plt.title('Silhouette Score vs. Number of Clusters')
[43] plt.xlabel('Number of Clusters (k)')
[43] plt.ylabel('silhouette Score')
[43] plt.xticks(range(2, 26))
[43] plt.grid(True)
[43] plt.show()

[43] best_k2 = k_values[silhouette_scores.index(max(silhouette_scores))]
[43] print(f"The optimal value of k is: {best_k2}")
```



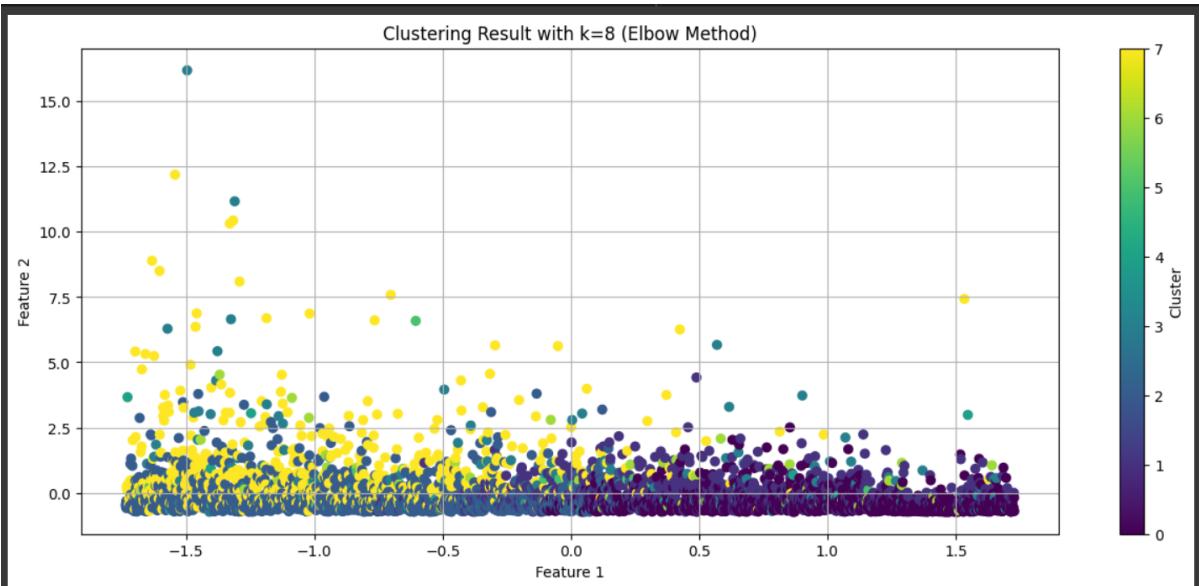
K = 2

▼ 5-8

```
2s [●] best_k = 8 # or 7

kmeans = KMeans(n_clusters=best_k, random_state=42)
cluster_labels = kmeans.fit_predict(data_scaled)

plt.figure(figsize=(15, 6))
plt.scatter(data_scaled[:, 0], data_scaled[:, 1], c=cluster_labels, cmap='viridis')
plt.title(f'Clustering Result with k={best_k} (Elbow Method)')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar(label='Cluster')
plt.grid(True)
plt.show()
```



```
best_k2 = 2

kmeans = KMeans(n_clusters=best_k2, random_state=42)
cluster_labels = kmeans.fit_predict(data_scaled)

plt.figure(figsize=(15, 6))
plt.scatter(data_scaled[:, 0], data_scaled[:, 1], c=cluster_labels, cmap='viridis')
plt.title(f'Clustering Result with k={best_k2} (Silhouette Score)')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar(label='Cluster')
plt.grid(True)
plt.show()
```

