

מטלה 3

מגיש: אור שני
ת.ז. 206812034

:K-MEANS

ניסוי 1:

```
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
kmeans.fit(X)
optimal_score = silhouette_score(X, kmeans.labels_)

# final result
print(f"Optimal K={optimal_k}, Silhouette Score: {optimal_score}")
```

⇒ Optimal K=8, Silhouette Score: 0.716259254869025

ניסוי 2:

```
kmeans = KMeans(n_clusters=optimal_k, random_state=10000)
kmeans.fit(X)
optimal_score = silhouette_score(X, kmeans.labels_)

# final result
print(f"Optimal K={optimal_k}, Silhouette Score: {optimal_score}")
```

⇒ Optimal K=8, Silhouette Score: 0.7162749930599402

ניסוי 3:

```
#optimal number of clusters
optimal_k = k_values[np.argmax(silhouette_scores)]

kmeans = KMeans(n_clusters=optimal_k,init='random', random_state=1,algorithm='elkan')
kmeans.fit(X)
optimal_score = silhouette_score(X, kmeans.labels_)

# final result
print(f"Optimal K={optimal_k}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=8, Silhouette Score: 0.7162749930599402

ניסוי 4:

```
#optimal number of clusters
optimal_k = k_values[np.argmax(silhouette_scores)]

kmeans = KMeans(n_clusters=optimal_k,copy_x=False,algorithm='elkan',random_state=300)
kmeans.fit(X)
optimal_score = silhouette_score(X, kmeans.labels_)

# final result
print(f"Optimal K={optimal_k}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=8, Silhouette Score: 0.716259254869025

ניסוי 5:

```
kmeans = KMeans(n_clusters=optimal_k,copy_x=False,random_state=200)
kmeans.fit(X)
optimal_score = silhouette_score(X, kmeans.labels_)

# final result
print(f"Optimal K={optimal_k}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=8, Silhouette Score: 0.7160621457169183

:Agglomerative Clustering

ניסוי 1:

```
agg_clustering = AgglomerativeClustering(n_clusters=optimal_clusters)
agg_labels = agg_clustering.fit_predict(X)
optimal_score = silhouette_score(X, agg_labels)

print(f"Optimal K={optimal_clusters}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=6, Silhouette Score: 0.7256900714288006

ניסוי 2:

```
agg_clustering = AgglomerativeClustering(n_clusters=optimal_clusters,compute_full_tree=False)
agg_labels = agg_clustering.fit_predict(X)
optimal_score = silhouette_score(X, agg_labels)

print(f"Optimal K={optimal_clusters}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=6, Silhouette Score: 0.7256900714288006

ניסוי 3:

```
agg_clustering = AgglomerativeClustering(n_clusters=optimal_clusters,linkage='average')
agg_labels = agg_clustering.fit_predict(X)
optimal_score = silhouette_score(X, agg_labels)

print(f"Optimal K={optimal_clusters}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=6, Silhouette Score: 0.6680458764033119

ניסוי 4:

```
optimal_clusters = cluster_range[np.argmax(silhouette_scores)]

agg_clustering = AgglomerativeClustering(n_clusters=optimal_clusters,compute_distances=True,compute_full_tree=True)
agg_labels = agg_clustering.fit_predict(X)
optimal_score = silhouette_score(X, agg_labels)

print(f"Optimal K={optimal_clusters}, Silhouette Score: {optimal_score}")
```

➡ Optimal K=6, Silhouette Score: 0.7256900714288006

ניסוי 5:

```
agg_clustering = AgglomerativeClustering(n_clusters=optimal_clusters,linkage="complete",metric="chebyshev")
agg_labels = agg_clustering.fit_predict(X)
optimal_score = silhouette_score(X, agg_labels)

print(f"Optimal K={optimal_clusters}, Silhouette Score: {optimal_score}")
```

⇒ Optimal K=6, Silhouette Score: 0.6643503632218198

DBSCAN

ניסוי 1:

```
dbscan = DBSCAN(eps=eps_optimal, min_samples=10)
dbscan_labels = dbscan.fit_predict(X)

if len(set(dbscan_labels)) > 1:
    score = silhouette_score(X, dbscan_labels)
    print(f"DBSCAN - eps={eps_optimal}, Silhouette Score: {score}")
```

⇒ DBSCAN - eps=1.5, Silhouette Score: 0.7222593187874933

ניסוי 2:

```
dbscan = DBSCAN(eps=eps_optimal, min_samples=10,metric='manhattan')
dbscan_labels = dbscan.fit_predict(X)

if len(set(dbscan_labels)) > 1:
    score = silhouette_score(X, dbscan_labels)
    print(f"DBSCAN - eps={eps_optimal}, Silhouette Score: {score}")
```

⇒ DBSCAN - eps=1.5, Silhouette Score: 0.7070662292310961

ניסוי 3:

```
dbscan = DBSCAN(eps=eps_optimal, min_samples=5,metric='manhattan')
dbscan_labels = dbscan.fit_predict(X)

if len(set(dbscan_labels)) > 1:
    score = silhouette_score(X, dbscan_labels)
    print(f"DBSCAN - eps={eps_optimal}, Silhouette Score: {score}")
```

⇒ DBSCAN - eps=1.5, Silhouette Score: 0.7203869947734542

ניסוי 4:

```

dbscan = DBSCAN(eps=eps_optimal, min_samples=10,algorithm='kd_tree')
dbscan_labels = dbscan.fit_predict(X)

if len(set(dbscan_labels)) > 1:
    score = silhouette_score(X, dbscan_labels)
    print(f"DBSCAN - eps={eps_optimal}, Silhouette Score: {score}")

```

DBSCAN - eps=1.5, Silhouette Score: 0.7222593187874933

ניסוי 5:

```

dbscan = DBSCAN(eps=eps_optimal, min_samples=10,algorithm='ball_tree',metric="l1")
dbscan_labels = dbscan.fit_predict(X)

if len(set(dbscan_labels)) > 1:
    score = silhouette_score(X, dbscan_labels)
    print(f"DBSCAN - eps={eps_optimal}, Silhouette Score: {score}")

```

DBSCAN - eps=1.5, Silhouette Score: 0.7070662292310961