<u>עבור בדיקת הפרמטרים הכי טובים עבור K-Means</u> –

*בדיקת הפרמטרים מסומנת בצהוב

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

best_kmeans_score = -1
best_kmeans_params = {}
kmeans_results = []

for n_clusters in range(2, 11):
  for init_method in ['k-means++', 'random']:
    kmeans = KMeans(n_clusters=n_clusters, init=init_method, random_state=42)
    kmeans.fit(X)
    score = silhouette_score(X, kmeans.labels_)
    kmeans_results.append((n_clusters, init_method, score))
    if score > best_kmeans_score:
      best_kmeans_score = score

print('All KMeans Results:')
for result in kmeans_results:
  print(f'n_clusters: {result[0]}, init: {result[1]}, silhouette score: {result[2]:}')
```

n_clusters: מספר הקבוצות (נבדקו ערכים בין 2 ל-10).

init: שיטת האתחול (נבדקו ערכים 'k-means++' ו-'random').

התוצאות-

```
All KMeans Results:
n_clusters: 2, init: k-means++, silhouette score: 
0.5672585793914866
n_clusters: 2, init: random, silhouette score: 0.5672585793914866
n_clusters: 3, init: k-means++, silhouette score: 
0.5857904307157648
n_clusters: 3, init: random, silhouette score: 0.5857904307157648
n_clusters: 4, init: k-means++, silhouette score: 
0.6813192037113843
n_clusters: 4, init: random, silhouette score: 0.6813192037113843
n_clusters: 5, init: k-means++, silhouette score: 
0.6432758294253328
n_clusters: 5, init: random, silhouette score: 0.6976139874193659
n_clusters: 6, init: k-means++, silhouette score: 
0.6605985626907828
n_clusters: 6, init: random, silhouette score: 0.7251181630059241
n_clusters: 7, init: k-means++, silhouette score: 
0.7026041646015059
n_clusters: 7, init: random, silhouette score: 0.640026695635202
n_clusters: 8, init: k-means++, silhouette score: 
0.716259254869025
n_clusters: 8, init: random, silhouette score: 0.6178145880678158
n_clusters: 9, init: k-means++, silhouette score: 
0.7117502750033986
n_clusters: 9, init: random, silhouette score: 0.6267846061018335
```

```
n_clusters: 10, init: k-means++, silhouette score: 
0.7105122316647566
 n_clusters: 10, init: random, silhouette score: 0.559886632470254
```

<u>עבור בדיקת הפרמטרים הכי טובים עבור Agglomerative Clustering</u>–

*בדיקת הפרמטרים מסומנת בצהוב

```
from sklearn.cluster import AgglomerativeClustering

best_agg_score = -1
best_agg_params = {}
agg_results = []

for n_clusters in range(2, 11):
    for linkage in ['ward', 'complete', 'average', 'single']:
        agg = AgglomerativeClustering(n_clusters=n_clusters, linkage=linkage)
        agg.fit(X)
        score = silhouette_score(X, agg.labels_)
        agg_results.append((n_clusters, linkage, score))
        if score > best_agg_score:
            best_agg_score = score

print('All Agglomerative Clustering Results:')
for result in agg_results:
    print(f'n_clusters: {result[0]}, linkage: {result[1]}, silhouette score: {result[2]:}')
```

n_clusters: מספר הקבוצות (נבדקו ערכים בין 2 ל-10).

linkage: שיטת הקישור (נבדקו ערכים 'ward', 'complete', 'average', 'single').

התוצאות-

```
All Agglomerative Clustering Results:
n_clusters: 2, linkage: ward, silhouette score: 
0.5698462725885549
n_clusters: 2, linkage: complete, silhouette score: 
0.5090246272851084
n_clusters: 2, linkage: average, silhouette score: 
0.5698462725885549
n_clusters: 2, linkage: single, silhouette score: 
0.3587824456286678
n_clusters: 3, linkage: ward, silhouette score: 
0.5806226231752926
n_clusters: 3, linkage: complete, silhouette score: 
0.47295414538916447
n_clusters: 3, linkage: average, silhouette score: 
0.5806226231752926
n_clusters: 3, linkage: single, silhouette score: 
0.5866220981494794
n_clusters: 4, linkage: ward, silhouette score: 
0.6812638781424561
```

```
n_clusters: 4, linkage: complete, silhouette score:
0.6812638781424561
n_clusters: 4, linkage: average, silhouette score:
0.6812638781424561
n_clusters: 4, linkage: single, silhouette score:
0.5180398014280129
n_clusters: 5, linkage: ward, silhouette score: 0.69778623369389
n_clusters: 5, linkage: complete, silhouette score:
0.6923413939127293
n_clusters: 5, linkage: average, silhouette score:
0.6923413939127293
n_clusters: 5, linkage: single, silhouette score:
0.6923413939127293
n_clusters: 6, linkage: ward, silhouette score:
0.7256900714288006
n_clusters: 6, linkage: complete, silhouette score:
0.6594689980695821
n_clusters: 6, linkage: average, silhouette score:
0.6680458764033119
n_clusters: 6, linkage: single, silhouette score:
0.613327782899459
n_clusters: 7, linkage: ward, silhouette score:
0.7014960182059224
n_clusters: 7, linkage: complete, silhouette score:
0.668466237784836
n_clusters: 7, linkage: average, silhouette score:
0.6765904629884553
n_clusters: 7, linkage: single, silhouette score:
0.604522825115808
n_clusters: 8, linkage: ward, silhouette score:
0.7085758473975964
n_clusters: 8, linkage: complete, silhouette score:
0.7018771832746021
n_clusters: 8, linkage: average, silhouette score:
0.7099391405045268
n_clusters: 8, linkage: single, silhouette score:
0.44573935523939057
n_clusters: 9, linkage: ward, silhouette score:
0.7088190473478537
n_clusters: 9, linkage: complete, silhouette score:
0.7046018505552566
n_clusters: 9, linkage: average, silhouette score:
0.7093447499958512
n_clusters: 9, linkage: single, silhouette score:
0.35175198194067037
n_clusters: 10, linkage: ward, silhouette score:
0.7088165601086022
n_clusters: 10, linkage: complete, silhouette score:
0.7078385268925678
n_clusters: 10, linkage: average, silhouette score:
0.7073095644071425
n_clusters: 10, linkage: single, silhouette score:
0.25184685533953083
```

<u>עבור בדיקת הפרמטרים הכי טובים עבור DBSCAN</u>–

*בדיקת הפרמטרים מסומנת בצהוב

```
from sklearn.cluster import DBSCAN

best_dbscan_score = -1
best_dbscan_params = {}
dbscan_results = []

eps_values = [0.5, 1.0, 1.5, 2.0, 2.5]
min_samples_values = [5, 10, 15, 20]

for eps in eps_values:
   for min_samples in min_samples_values:
      dbscan = DBSCAN(eps=eps, min_samples=min_samples)
      dbscan.fit(X)
      if len(set(dbscan.labels_)) > 1:  # Ensure there is more than one cluster
         score = silhouette_score(X, dbscan.labels_)
         dbscan_results.append((eps, min_samples, score))
         if score > best_dbscan_score:
            best_dbscan_score = score

print('All DBSCAN Results:')
for result in dbscan_results:
   print(f'eps: {result[0]}, min_samples: {result[1]}, silhouette score: {result[2]:}')
```

eps: רדיוס השכנות (נבדקו ערכים 0.5, 1.0, 1.5, 2.0, 2.5).

min_samples: מספר הדוגמאות המינימלי בתוך קבוצת הליבה (נבדקו ערכים 5, 10, 15, 20).

התוצאות-

```
All DBSCAN Results:
eps: 0.5, min_samples: 5, silhouette score: 0.28865209433880534
eps: 0.5, min_samples: 10, silhouette score: 0.44913770731330993
eps: 0.5, min_samples: 15, silhouette score: 0.30549274312867614
eps: 0.5, min_samples: 20, silhouette score: 0.17396195043369547
eps: 1.0, min_samples: 5, silhouette score: 0.7070123298589694
eps: 1.0, min_samples: 10, silhouette score: 0.6190125761523078
eps: 1.0, min_samples: 15, silhouette score: 0.5102933416053745
eps: 1.0, min_samples: 20, silhouette score: 0.6612622555185036
eps: 1.5, min_samples: 5, silhouette score: 0.689401642082808 4
eps: 1.5, min_samples: 10, silhouette score: 0.7222593187874933
eps: 1.5, min_samples: 15, silhouette score: 0.7135727033307426
eps: 1.5, min_samples: 20, silhouette score: 0.6884017793929554
eps: 2.0, min_samples: 5, silhouette score: 0.613327782899459
eps: 2.0, min_samples: 10, silhouette score: 0.613327782899459
eps: 2.0, min_samples: 15, silhouette score: 0.6809842207458477
eps: 2.0, min_samples: 20, silhouette score: 0.6809842207458477
eps: 2.5, min_samples: 5, silhouette score: 0.6923413939127293
eps: 2.5, min_samples: 10, silhouette score: 0.6923413939127293
eps: 2.5, min_samples: 15, silhouette score: 0.6923413939127293
```

```
eps: 2.5, min_samples: 20, silhouette score: 0.6923413939127293
```