שם מגיש: יונתן שירמן

ת.ז: 208870246

מטלה מס' 3 – מבוא ללמידת מכונה

CLUSTERS

במטלה זו אבחן 3 מודלים שונים ליצירת CLUSTERS – Agglomerative, DBscan, Kmeans ואבחן את איכות הקלאסטרים שכל אחד יצר עם פרמטרים שונים כל הרצה, באמצעות מטריקת silhouette. לבסוף ייבחר המודל הטוב ביותר עפ"י אותו מדד silhouette.

אבצע את הניסויים במקביל ולבסוף אשווה בין המודלים. בכל הרצה, יתבצעו 3 ניסויים, 1 לכל מודל ובסוף תתבצע ההשוואה בין התוצאות.

ניסוי מס' 1 – ערכים דיפולטיביים

בניסויים אלו אתבסס על ערכים דיפולטיביים של המודלים.

KMEANS:

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

kmeans= KMeans().fit(X)
#kmeans.fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.7160120743931654
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering().fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.5698462725885549
```

DBSCAN

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

#eps=0.5, min_samples=5
DB = DBSCAN().fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.28865209433880534
```

RESULTS:

```
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: KMeans with a silhouette score of 0.7162749930599402
```

המודל עם התוצאה ביותר עם הערכים הדיפולטיביים הוא KMEANS עם ציון של 0.71

ניסוי מס' 2

KMEANS:

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

kmeans= KMeans(n_clusters=8, algorithm='lloyd').fit(X)
#kmeans.fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.7162749930599402
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering(n_clusters=2, metric='manhattan', compute_full_tree=False, linkage='average').fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.5698462725885549
```

DBSCAN

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


DB = DBSCAN(eps=0.5, min_samples=5, metric='euclidean', algorithm='auto').fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.28865209433880534
```

RESULTS:

```
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: KMeans with a silhouette score of 0.7162749930599402
```

בניסוי מס' 2 התוצאה הטובה ביותר הייתה של מודל KMEANS

ניסוי מס' 3

KMEANS:

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

kmeans= KMeans(n_clusters=5, algorithm='elkan',max_iter=200).fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.643268128150937
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering(n_clusters=5, metric='manhattan', compute_full_tree=False, linkage='average').fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.6923413939127293
```

DBSCAN

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score


DB = DBSCAN(eps=0.6, min_samples=7, metric='manhattan', algorithm='auto').fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.33453336951829593
```

RESULTS:

```python
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: Agglomerative Clustering with a silhouette score of 0.6923413939127293
```

בניסוי זה AGGLOMERATIVE השיג את התוצאה הגבוהה ביותר.

ניסוי מס' 4

KMEANS:

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

kmeans= KMeans(n_clusters=5, algorithm='lloyd',max_iter=200).fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.6976139874193659
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering(n_clusters=5, metric='cosine', compute_full_tree=False, linkage='average').fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.5449758907237462
```

DBSCAN

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


DB = DBSCAN(eps=0.6, min_samples=7, metric='euclidean', algorithm='auto').fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.4247975447433235
```

RESULTS:

```
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: KMeans with a silhouette score of 0.6976139874193659
```

גם כאן מודל KMEANS הציג את התוצאות הטובות ביותר מבין המודלים.

ניסוי מס' 5

KMEANS:

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

kmeans= KMeans(n_clusters=6, algorithm='lloyd',max_iter=400).fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.6780131367812913
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering(n_clusters=7, metric='euclidean',linkage='complete').fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.668466237784836
```

DBSCAN

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score


DB = DBSCAN(eps=1.5, min_samples=7, metric='euclidean', algorithm='kd_tree').fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.6888237813315415
```

RESULTS:

```python
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: DBSCAN with a silhouette score of 0.6888237813315415
```

בניסוי זה מודל DBSCAN יצא עם ידו על העליונה עם התוצאה הטובה ביותר

ניסוי מס' 6

KMEANS:

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

kmeans= KMeans(n_clusters=7, algorithm='lloyd',max_iter=100, n_init=10).fit(X)
kmlabels = kmeans.labels_
KMscore = silhouette_score(X, kmlabels)
centers = kmeans.cluster_centers_
print("Silhouette Score for KMeans:", KMscore)
```

```
Silhouette Score for KMeans: 0.7017611019403646
```

Agglomerative Clustering:

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


Aggl = AgglomerativeClustering(n_clusters=7, metric='manhattan',linkage='average').fit(X)
cluster_labels = Aggl.labels_
Agglscore = silhouette_score(X, cluster_labels)
print(f"Silhouette Score: {Agglscore:}")
```

```
Silhouette Score: 0.6768246922298061
```

DBSCAN

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


DB = DBSCAN(eps=1.5, min_samples=7, metric='manhattan', algorithm='kd_tree').fit(X)
cluster_labels = DB.labels_


if len(set(cluster_labels)) > 1: #cant evaluate clustering quality with only 1 cluster.
    DBscore = silhouette_score(X, cluster_labels)
    print(f"Silhouette Score: {DBscore:}")
else:
    print("Silhouette Score cannot be calculated (only one cluster detected).")
```

```
Silhouette Score: 0.7203869947734542
```

RESULTS:

```python
scores = {
    "KMeans": KMscore,
    "DBSCAN": DBscore,
    "Agglomerative Clustering": Agglscore
}

best_model = max(scores, key=scores.get)
print(f"The best model is: {best_model} with a silhouette score of {scores[best_model]}")
```

```
The best model is: DBSCAN with a silhouette score of 0.7203869947734542
```

גם כאן התקבלה התוצאה הטובה ביותר למודל DBSCAN.