1) ما هو التجمع؟ اشرح بمثال بطريقة هل يمكن أن تكون مفيدة؟

التجزئة (Clustering) هي عملية تنظيم البيانات المتشابهة معًا في مجموعات صغيرة تسمى (clusters)، حيث يتم تجميع البيانات التي تشترك في خصائص معينة معًا داخل نفس التجمع (cluster)، بينما تكون متميزة عن بيانات التجمعات الأخرى. بمعنى آخر، فإنها تسعى إلى فصل البيانات إلى مجموعات تتشابه داخلها وتختلف بينها.

على سبيل المثال، يمكن تجميع المستخدمين الذين يشاهدون ويفضلون مقاطع الفيديو ذات المحتوى التقني في مجموعة واحدة، بينما يمكن تجميع المستخدمين الذين يفضلون مقاطع الفيديو ذات المحتوى المنزلي في مجموعة أخرى.

مثل إذا كانت بيانات الاستخدام تشير إلى أن هناك مجموعة كبيرة من المستخدمين يشاهدون بشكل رئيسي مقاطع الفيديو التقنية والمرتبطة بالتكنولوجيا، فيمكن تجميعهم في تجمع واحد. بينما يمكن تجميع المستخدمين الذين يشاهدون بشكل رئيسي مقاطع الفيديو التي تتعلق بالطبخ والديكور في (cluster) آخر.

مثال اخر: لنفترض أن لدينا مجموعة من المشترين عبر الإنترنت، ونريد تقسيمهم إلى مجموعات استنادًا إلى تفضيلاتهم فيما يتعلق بالمنتجات التقنية. إذا كانت بيانات المشترين تشير إلى أن هناك مجموعة يشترون بشكل رئيسي الهواتف الذكية والأجهزة الإلكترونية، فيمكن تجميعهم في (cluster) واحد. بينما يمكن تجميع المشترين الذين يشترون بشكل رئيسي الأجهزة المنزلية مثل الثلاجات والغسالات في (cluster) آخر.

هذا المثال يظهر كيف يمكن استخدام التجزئة لتحليل سلوك المشترين وتقسيمهم إلى مجموعات استنادًا إلى تفضيلاتهم في المنتجات، مما يمكن للشركة من تحديد استراتيجيات التسويق والعروض الترويجية المستهدفة لكل مجموعة.

فيما يلي بعض الطرق التي يمكن أن تكون فيها التجزئة مفيدة:

1. تصنيف العملاء أو المستهلكين: من خلال تجزئة العملاء أو المستهلكين بناءً على عادات الشراء أو التفضيلات، يمكن للشركات توجيه استراتيجيات التسويق بشكل أفضل وتلبية احتياجات العملاء بشكل أكثر فعالية.
2. تجزئة الصور أو الفيديوهات: يمكن استخدام التجزئة في معالجة الصور والفيديوهات لتنظيمها وتصنيفها بناءً على الخصائص المشتركة مثل الألوان أو الأشكال أو النمط.
3. تحليل البيانات الجغرافية: يمكن استخدام التجزئة لتجميع البيانات الجغرافية مثل العناوين البريدية أو المواقع الجغرافية لتحديد مناطق محددة أو توزيعات جغرافية.
4. تحليل السلوك الاجتماعي عبر وسائل التواصل الاجتماعي: يمكن استخدام التجزئة لتقسيم المستخدمين على وسائل التواصل الاجتماعي إلى مجموعات استنادًا إلى أنماط تفاعلهم ومشاركاتهم، مما يسمح للشركات والمعلنين بفهم أفضل لجمهورهم وتصميم الحملات التسويقية بشكل أكثر فعالية.
5. تقسيم المرضى في المجال الطبي: يمكن استخدام التجزئة لتقسيم المرضى إلى مجموعات استنادًا إلى التشخيصات الطبية والعوامل الصحية المشتركة، مما يمكن الأطباء من تخصيص العلاجات والرعاية بشكل أكثر دقة وفعالية.

2) ما هي أنواع مختلفة من التجمع؟ اشرح مع الأمثلة.

1. التجزئة النمطية:(Partitioning Clustering)

في هذا النوع من التجزئة، يتم تقسيم مجموعة البيانات إلى عدد محدد من التجمعات، وتعتمد هذه التجميعات على معايير محددة مثل المسافات بين النقاط. واحدة من أشهر أنواع التجزئة النمطية هي خوارزمية K-means.

مثال: تقسيم مجموعة من العملاء في متجر إلكتروني إلى مجموعات استنادًا إلى عادات شرائهم، حيث يمكن تقسيمهم إلى مجموعة من يشترون ملابس رياضية ومجموعة أخرى من يشترون ملابس أنيقة.

2. التجزئة بوسائل القياس:(K-Means Clustering)

في هذا النوع من التجزئة، يتم تقسيم مجموعة البيانات إلى عدد محدد من التجمعات، حيث يعتمد تكوين هذه التجمعات على المسافات بين النقاط. واحدة من أشهر أنواع التجزئة بوسائل القياس هي خوارزمية.K-Means
مثال: يمكن تقسيم مجموعة من العملاء في متجر إلكتروني إلى مجموعات استنادًا إلى عادات شرائهم. على سبيل المثال، يمكن تقسيمهم إلى مجموعة من يشترون ملابس رياضية ومجموعة أخرى من يشترون ملابس أنيقة.

3. التجمعات الهرمية:(Hierarchical Clustering)

هذا النوع من التجزئة يقوم بتقسيم المجموعة إلى تجمعات فرعية تدريجيًا بناءً على مستويات مختلفة من التفاعل بين البيانات. يمكن أن يكون التجزئة هرمية متقدمة لتكون تجزئة مفصلة تصل إلى مستويات صغيرة جدًا (وتسمى "التجزئة النمطية").
مثال: تقسيم مجموعة من السكان في منطقة معينة إلى مناطق فرعية بناءً على المواصفات الجغرافية، مثل البلدان أو المدن.

4. التجمعات المركزية:(Centroid Clustering)
في هذا النوع من التجمعات، يتم تقسيم البيانات استنادًا إلى المسافة من نقاط مركزية محددة مسبقًا (مراكز). يتم اختيار هذه المراكز بحيث تمثل مواقع متوسطة لمجموعات البيانات.
مثال: تجميع السكان في المدن بناءً على المسافة من وسط المدينة، يتم تجزئة السكان في المدن استنادًا إلى مسافتهم من وسط المدينة. يتم اختيار وسط المدينة كمركز للتجمع، ومن ثم يتم تقسيم السكان إلى مجموعات استنادًا إلى المسافة من هذا المركز.

5. التجمعات الكثافية:(Density-based Clustering)
في هذا النوع من التجمعات، يتم تحديد التجمعات بناءً على كثافة البيانات في المساحة، حيث تكون المناطق ذات كثافة عالية من المعطيات تمثل تجمعات.
مثال: تجزئة المناطق الحضرية إلى أحياء، يتم تحديد التجمعات بناءً على كثافة السكان في المناطق الحضرية. يتم تحديد التجمعات عن طريق تحديد المناطق التي تحتوي على كثافة عالية من السكان وتعتبر نقاط انطلاق لتجمعات جديدة.

6. التجمعات الطبيعية:(Natural Clustering)
يشير هذا النوع من التجمعات إلى تجمعات تظهر بشكل طبيعي في البيانات دون وجود تدخل خارجي في عملية التجميع، مثل تجمعات المدن في البيانات الجغرافية.
مثال: تجزئة البيانات الجغرافية للمدن والقرى، تظهر تجمعات طبيعية في البيانات الجغرافية للمدن والقرى بناءً على المواقع الجغرافية والعوامل البيئية والاقتصادية المحيطة. على سبيل المثال، يمكن تجزئة البيانات الجغرافية لمنطقة إلى تجمعات طبيعية مثل المدن والقرى والمناطق الريفية.

7. التجمعات المناسبة للغرض:(Task-specific Clustering)
في هذا النوع من التجمعات، يتم تكييف عملية التجميع بشكل خاص لحل مشكلة محددة أو تحقيق هدف محدد، مما يتطلب استخدام معايير خاصة ومتخصصة في عملية التجميع.
مثال: تجزئة العملاء في متجر التجزئة لإعداد حملات تسويقية مستهدفة، يتم تكييف عملية التجميع بشكل خاص لتحليل بيانات العملاء في متجر التجزئة بهدف إعداد حملات تسويقية مستهدفة. يمكن استخدام معايير مثل نمط الشراء والميزانية لتقسيم العملاء إلى مجموعات مستهدفة.

3) وضح مشكلتين على الأقل مع خوارزمية K-Means Clustering واضف الحلول المحتملة ؟

المشكلة الأولى: النقاط الأولية:(Initial Centroids)

- المشكلة: يعتمد أداء خوارزمية K-Means بشكل كبير على اختيار النقاط المركزية الأولية. إذا تم اختيار النقاط الأولية بشكل عشوائي، فقد يؤدي ذلك إلى وصول الخوارزمية إلى حلول محلية غير مثلى، خاصةً في الحالات التي تحتوي على تجمعات مختلفة بأحجام متفاوتة أو شكل غير منتظم.
- الحل: من أجل التغلب على هذه المشكلة، يمكن استخدام تقنية K-means++ لاختيار النقاط المركزية الأولية بشكل ذكي. تقوم هذه التقنية بتحديد النقاط المركزية الأولية بحيث تكون متباعدة بشكل جيد، مما يزيد من فرص الوصول إلى تجزئة أكثر دقة واستقرارًا. بالإضافة إلى ذلك، يمكن أيضًا استخدام أساليب أخرى مثل الاستنتاج من بيانات سابقة أو استخدام تقنيات تحليل البيانات لتحديد النقاط المركزية الأولية بشكل أفضل.

المشكلة الثانية: عدد الـ K في K-Means

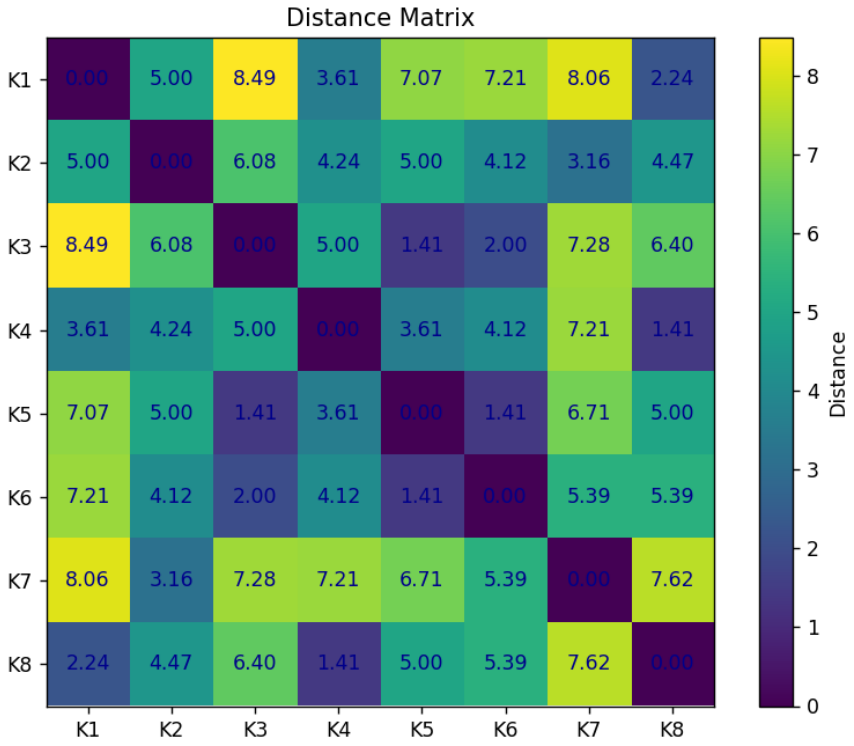
- المشكلة: قد يكون من الصعب تحديد عدد الـ K الأمثل، أي عدد التجمعات التي يجب تقسيم البيانات إليها. اختيار قيمة غير مناسبة لـ K قد يؤدي إلى تجزئة غير دقيقة للبيانات، حيث يمكن أن تكون التجمعات مفرطة في التجزئة (عدد كبير من التجمعات) أو غير مكتملة (عدد قليل من التجمعات).
- الحل: هناك عدة طرق لتحديد عدد الـ K بشكل مناسب، منها:
  - أسلوب الكوع :(Elbow Method) يقوم هذا الأسلوب بتجريب قيم مختلفة لـ K وقياس متوسط مربعات الانحراف (SSE) لكل قيمة من الـ K يتم اختيار قيمة الـ K التي تظهر فيها منحنى SSE انحناءً حادًا، مما يشير إلى أن تجزئة البيانات بهذه القيمة من الـ K هي الأكثر دقة. ويتم توضيحها بالرسم.
  - أسلوب :Silhouette يستخدم هذا الأسلوب قياس الانفصال بين التجمعات لتحديد الـ K الأمثل. يتم اختيار القيمة التي تحقق أعلى قيمة لمقياسSilhouette ، حيث يشير ذلك إلى أن البيانات تكون مجمعة بشكل جيد في التجمعات المحددة.
  - الاستنتاج البصري: قد يكون من المفيد أيضًا فحص البيانات بشكل بصري وتقدير عدد التجمعات بناءً على فهمك للبيانات والهدف من التجزئة.

4) اوجد المسافة ال Euclidean d(K1, K2)؟

أ-

Sqrt((2-2)**2+(10-5)**2)=5

بعد الحل تبين ان المسافة بين (K1, K2) تساوي 5



ب-

في البداية تبدو ان التكرارات هي نفسها ولكن عند التدقيق في الاحداثيات يتبين انها مختلفة قليلا، ومع ذلك فإن التباين بين النقطة الوسطى في التكرارات هو تغير ادنى.

نلاحظ انه من الرغم انه تتشابه بصريا (يصعب ملاحظتها بالعين) الا انها تتغير وتتلاقى في التكرار 2.

K-Means Clustering Iteration 1
Cluster 1
Cluster 2
Cluster 3
Centroids
X
Y

K-Means Clustering Iteration 2
Cluster 1
Cluster 2
Cluster 3
Centroids
X
Y

K-Means Clustering Iteration 3
Cluster 1
Cluster 2
Cluster 3
Centroids
X
Y

Clustering Analysis Report

Introduction:
This report provides an analysis of clustering techniques applied to a dataset using the K-Means and hierarchical clustering algorithms. The dataset contains points in a 2-dimensional space represented by the features X and Y.

Dataset Overview:
The dataset consists of points distributed in a 2-dimensional space. Each point is represented by its coordinates (X, Y). Before clustering, the dataset is visualized to understand its distribution and characteristics.

K-Means Clustering:
K-Means clustering is applied to the dataset with a predetermined number of clusters (`n_clusters = 3`). The algorithm iteratively assigns points to the nearest centroid and updates the centroids until convergence. Three iterations are performed to ensure stability in the results. The final centroids and cluster labels are visualized along with the original dataset.

Hierarchical Clustering:
Hierarchical clustering is applied using three different linkage methods: single, complete, and average. Each method defines the distance between clusters differently, leading to varying cluster structures. The resulting dendrograms are visualized to illustrate the hierarchical clustering process for each method.

Method Explanations:
-Single Linkage:
- Defines the distance between two clusters as the shortest distance between any two points in the clusters.
- Tends to produce elongated clusters sensitive to outliers and noise.

-Complete Linkage:
- Defines the distance between two clusters as the maximum distance between any two points in the clusters.
- Tends to produce compact, spherical clusters less sensitive to outliers.

-Average Linkage:
- Defines the distance between two clusters as the average distance between all pairs of points in the clusters.
- Balances sensitivity to outliers and cluster compactness, producing moderate compactness.

Elbow Method:

The elbow method is employed to determine the optimal number of clusters for K-Means clustering. It calculates the within-cluster sum of squares (inertia) for different numbers of clusters and identifies the point where the inertia starts to decrease at a slower rate, suggesting the optimal number of clusters.

Conclusion:

-K-Means clustering and hierarchical clustering with different linkage methods provide insights into the structure of the dataset.

-Each clustering technique has its advantages and limitations, which should be considered based on the specific characteristics of the dataset and the analysis goals.

-Further analysis and interpretation can be conducted based on the clustering results to extract meaningful patterns or insights from the data.

(6

```
class K-Means:
   تهيئة عدد التجمعات والحد الأقصى لعدد التكرارات//
def __init__(self, n_clusters, max_iters=1000):
        self.n_clusters = n_clusters
        self.max_iters = max_iters

تهيئة المراكز//
def fit(self, data, centroids):
        self.centroids = centroids
       إنشاء مصفوفة فارغة لتخزين تعيينات التجمع لكل نقطة بيانات//
       self.labels = np.zeros(len(data))
        // max_iters الدوران حتى التقارب أو الوصول إلى
        for _ in range(self.max_iters):
        تعيين كل نقطة بيانات إلى المركز الأقرب لها//
           for i in range(len(data)):
                       distances = [self.euclidean_distance(data[i], centroid) for centroid in self.centroids]
                        self.labels[i] = np.argmin(distances)
           تحديث المراكز استنادًا إلى المتوسط الحسابي لنقاط البيانات في كل تجمع//
           for cluster in range(self.n_clusters):
              cluster_points = data[self.labels == cluster]
              if len(cluster_points) > 0:
                 self.centroids[cluster] = np.mean(cluster_points, axis=0)

        return self.labels, self.centroids

   حساب المسافة اليوروكليدية بين نقطتين//
   def euclidean_distance(self, p1, p2):
      return np.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)
```

Assignment6-SulimanAlgaramanli.py

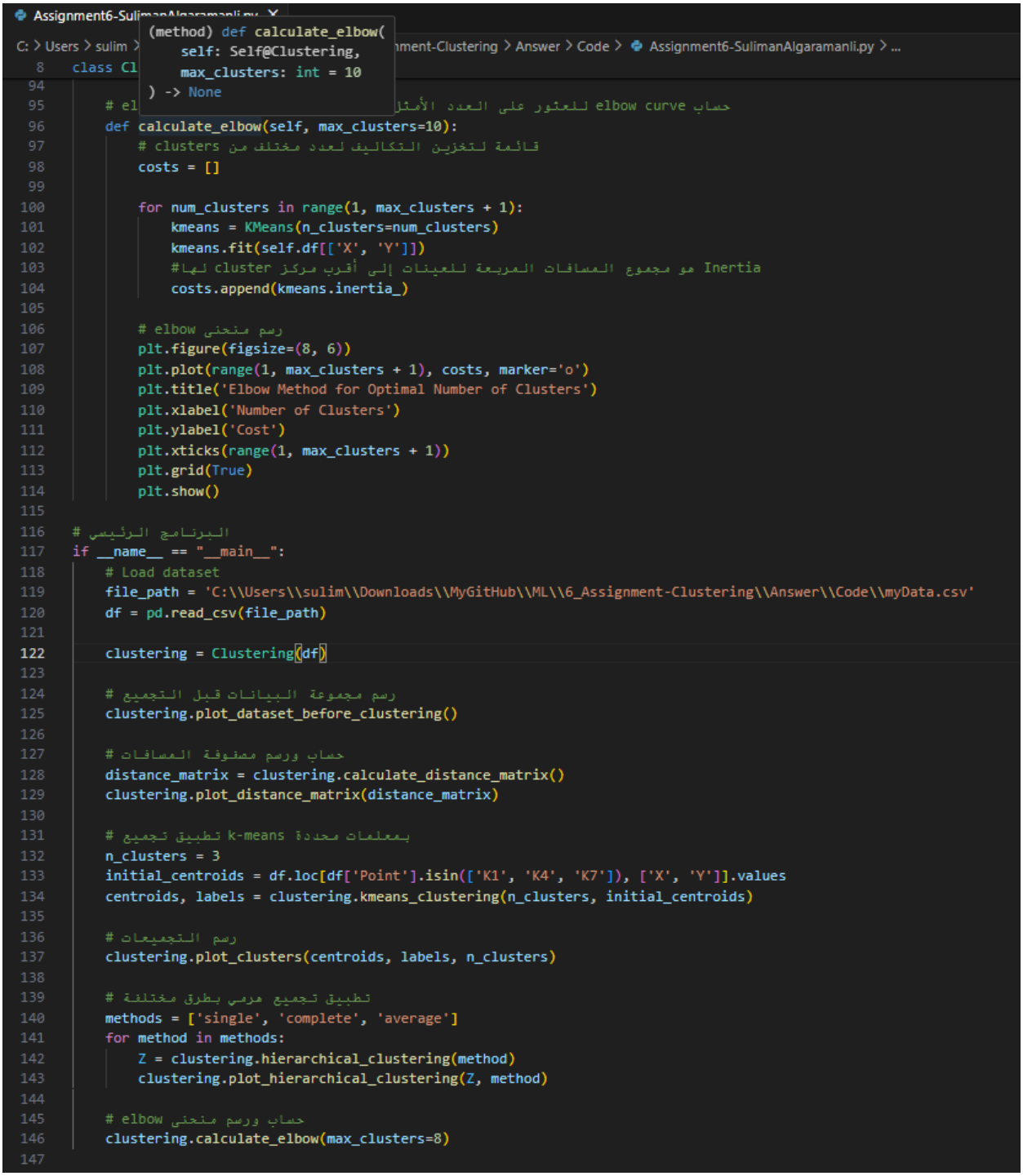
C: > Users > sulim > Downloads > MyGitHub > ML > 6_Assignment-Clustering > Answer > Code > Assignment6-SulimanAlgaramanli.py > ...

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram

#يتضمن وظائف لتنفيذ عمليات التجميع ورسم البيانات والنتائج
class Clustering:

    def __init__(self, df):
        self.df = df

    # رسم مجموعة البيانات قبل التجميع
    def plot_dataset_before_clustering(self):
        plt.figure(figsize=(8, 6))
        plt.scatter(self.df['X'], self.df['Y'], color='red')
        plt.title('Dataset Before Clustering')
        plt.xlabel('X')
        plt.ylabel('Y')
        plt.show()

    # رسم مصفوفة المسافات
    def plot_distance_matrix(self, distance_matrix):
        plt.figure(figsize=(8, 6))
        plt.imshow(distance_matrix, cmap='viridis')
        plt.colorbar(label='Distance')
        plt.title('Distance Matrix')
        plt.xticks(np.arange(len(self.df)), self.df['Point'])
        plt.yticks(np.arange(len(self.df)), self.df['Point'])

        for i in range(len(self.df)):
            for j in range(len(self.df)):
                plt.text(j, i, f'{distance_matrix[i, j]:.2f}', ha='center', va='center', color='darkblue')
        plt.show()

    # رسم التجميعات بعد تطبيق تجميع k-means
    def plot_clusters(self, centroids, labels, n_clusters):
        for iteration in range(3):
            plt.figure(figsize=(8, 6))
            for cluster in range(n_clusters):
                cluster_points = self.df.iloc[labels == cluster]
                plt.scatter(cluster_points['X'], cluster_points['Y'], label=f'Cluster {cluster + 1}')
            plt.scatter(centroids[:, 0], centroids[:, 1], color='black', marker='X', label='Centroids')
            plt.title(f'K-Means Clustering Iteration {iteration + 1}')
            plt.xlabel('X')
            plt.ylabel('Y')
            plt.legend()
            plt.show()

    # رسم شجرة التجميع الهرمي
    def plot_hierarchical_clustering(self, Z, method):
        plt.figure(figsize=(8, 6))
        dendrogram(Z)
        plt.title(f'Hierarchical Clustering ({method.capitalize()} Linkage)')
        plt.xlabel('Sample Index')
```

Assignment6-SulimanAlgaramanli.py

C: > Users > sulim > Downloads > MyGitHub > ML > 6_Assignment-Clustering > Answer > Code > Assignment6-SulimanAlgaramanli.py > ...

```
class Clustering:
    def plot_hierarchical_clustering(self, Z, method):
        plt.figure(figsize=(8, 6))
        dendrogram(Z)
        plt.title(f'Hierarchical Clustering ({method.capitalize()} Linkage)')
        plt.xlabel('Sample Index')
        plt.ylabel('Distance')
        plt.show()

    # حساب مصفوفة المسافات
    def calculate_distance_matrix(self):
        distance_matrix = np.zeros((len(self.df), len(self.df)))

        # حساب المسافة اليوروكليدية بين نقطتين p2 و p1 في الفضاء
        def euclidean_distance(p1, p2):
            return np.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

        for i, row1 in self.df.iterrows():
            for j, row2 in self.df.iterrows():
                distance_matrix[i, j] = euclidean_distance((row1['X'], row1['Y']), (row2['X'], row2['Y']))

        return distance_matrix

    # k-means تطبيق تجميع
    def kmeans_clustering(self, n_clusters, centroids, n_iterations=3, convergence_threshold=1e-4):

        for iteration in range(n_iterations):
            kmeans = KMeans(n_clusters=n_clusters, init=centroids, n_init=1)
            labels = kmeans.fit_predict(self.df[['X', 'Y']])
            new_centroids = kmeans.cluster_centers_

            if np.allclose(centroids, new_centroids, atol=convergence_threshold):
                print(f"Converged after {iteration + 1} iterations.")
                break

            centroids = new_centroids

        return centroids, labels

    # تطبيق تجميع هرمي
    def hierarchical_clustering(self, method):
        X = self.df[['X', 'Y']].values
        Z = linkage(X, method=method)
        return Z

    # elbow حساب elbow curve للعثور على العدد الأمثل للمجموعات باستخدام طريقة
    def calculate_elbow(self, max_clusters=10):
        # clusters قائمة لتخزين التكاليف لعدد مختلف من
        costs = []

        for num_clusters in range(1, max_clusters + 1):
            kmeans = KMeans(n_clusters=num_clusters)
            kmeans.fit(self.df[['X', 'Y']])
            #Inertia هو مجموع المسافات المربعة للعينات إلى أقرب مركز لها cluster
            costs.append(kmeans.inertia_)
```


Assignment6-SulimanAlgaramanli.py
(method) def calculate_elbow(
self: Self@Clustering,
max_clusters: int = 10
) -> None
C: > Users > sulim > nment-Clustering > Answer > Code > Assignment6-SulimanAlgaramanli.py > ...
class Cl
# el حساب elbow curve للعثور على العدد الأمثل
def calculate_elbow(self, max_clusters=10):
# clusters قائمة لتخزين التكاليف لعدد مختلف من
costs = []
for num_clusters in range(1, max_clusters + 1):
kmeans = KMeans(n_clusters=num_clusters)
kmeans.fit(self.df[['X', 'Y']])
#هو مجموع المسافات المربعة للعينات إلى أقرب مركز cluster لها Inertia
costs.append(kmeans.inertia_)
# elbow رسم منحنى
plt.figure(figsize=(8, 6))
plt.plot(range(1, max_clusters + 1), costs, marker='o')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Cost')
plt.xticks(range(1, max_clusters + 1))
plt.grid(True)
plt.show()
# البرنامج الرئيسي
if __name__ == "__main__":
# Load dataset
file_path = 'C:\\Users\\sulim\\Downloads\\MyGitHub\\ML\\6_Assignment-Clustering\\Answer\\Code\\myData.csv'
df = pd.read_csv(file_path)
clustering = Clustering(df)
# رسم مجموعة البيانات قبل التجميع
clustering.plot_dataset_before_clustering()
# حساب ورسم مصفوفة المسافات
distance_matrix = clustering.calculate_distance_matrix()
clustering.plot_distance_matrix(distance_matrix)
# تطبيق تجميع k-means بمعلمات محددة
n_clusters = 3
initial_centroids = df.loc[df['Point'].isin(['K1', 'K4', 'K7']), ['X', 'Y']].values
centroids, labels = clustering.kmeans_clustering(n_clusters, initial_centroids)
# رسم التجميعات
clustering.plot_clusters(centroids, labels, n_clusters)
# تطبيق تجميع هرمي بطرق مختلفة
methods = ['single', 'complete', 'average']
for method in methods:
Z = clustering.hierarchical_clustering(method)
clustering.plot_hierarchical_clustering(Z, method)
# elbow حساب ورسم منحنى
clustering.calculate_elbow(max_clusters=8)