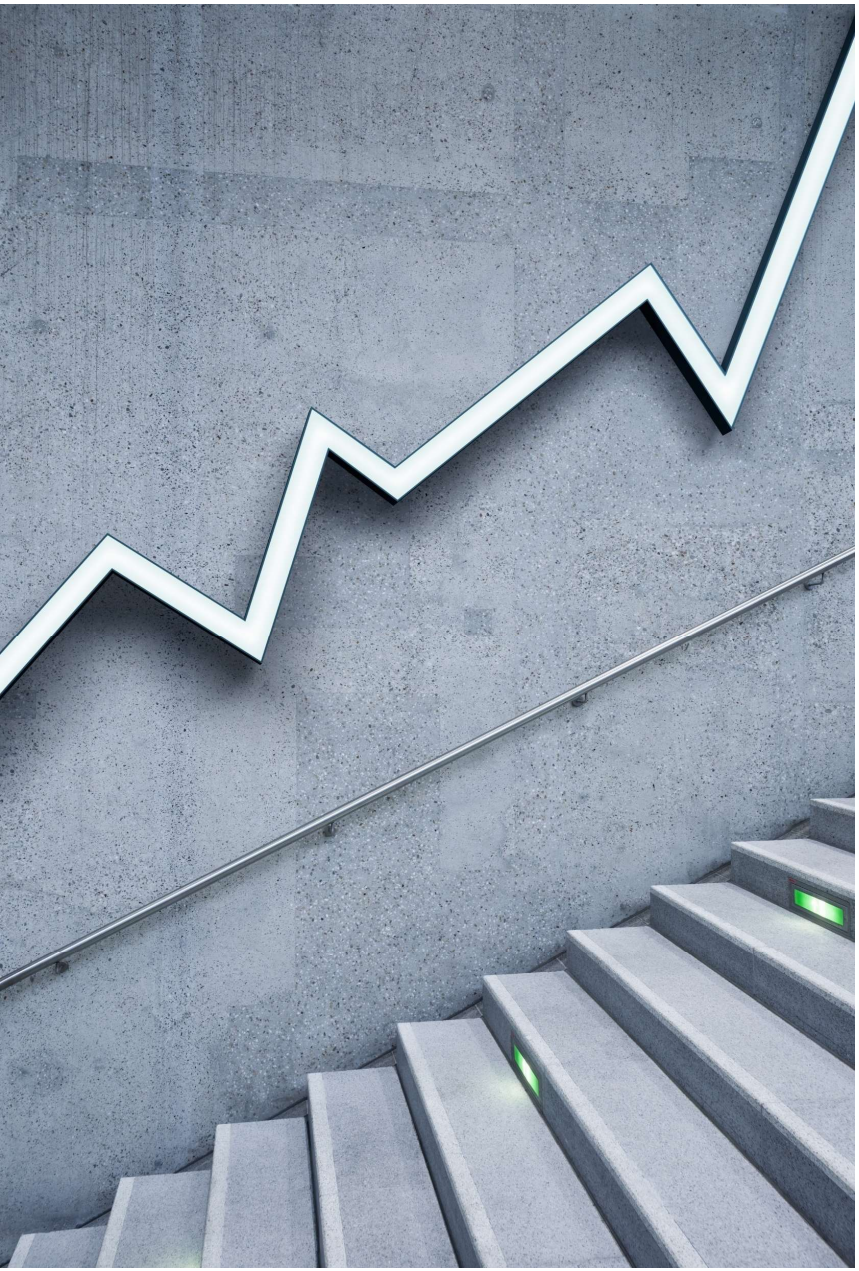


מבוא למדע הנתונים

דר' אריה יעקבי

"The Sexiest Job of the 21st Century"
(Davenport and Patil)

סמסטר ב' תשפ"ב

במצגת זו השתמשתי בין השאר בחומרים של ד"ר אורית גולדמן

חלוקה לאשכולות (צבירים)
Clustering

מה נלמד בשיעור זה

כיצד מבצעים clustering מסוג kMeans

- ❖ הקדמה ל- kMeans
- ❖ תכונות של kMeans
- ❖ יישום באמצעות פייתון:
  - ❖ קריאה של הנתונים
  - ❖ כיצד לקבוע את הערך k שמציין את מספר מקבצים
  - ❖ הצגה ויזואלית של המקבצים

- K-means היא טכניקה המוצאת מספר שצוין על ידי המשתמש של אשכולות (צבירים)
- צבירים אלו שנמצאו מיוצגים בדרך כלל על ידי הצנטרואידים (מרכזים) שלהם
- K-means היא אחת השיטות הקלות ביותר להבנה ויישום
- הטכניקה משתמשת במודל מתמטי של צמצום (מינימיזציה) של המרחק בין הנקודות המשויכות למרכז האשכול. מבחינה מתמטית, האופטימיזציה נראית כך:

$$J = \sum_{i=0}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2$$

כאשר S_i הוא צביר , הוא האלמנט ה-j של צביר i, ו- μ_i הוא מרכז הצביר

מאפיינים של k-means

- ניתן להרחבה: ניתן ליישם אותו ביעילות רבה וניתן להשתמש בו גם עם מערכי נתונים גדולים וממדים גבוהים.
- צורת אשכול: חסרון אחד של k-means הוא שהוא מעדיף אשכולות כדוריים בצורה ובגודל דומה. זהו גם חלוקה לאשכולות קשה (hard) שבו כל נקודה שייכת לאשכול אחד בלבד, ומכאן שלא יכולים להיות לה אשכולות מפוצלים או חופפים.
- יציבות: מכיוון ש- k-means היא שיטת ליצירת אשכול קשה, היא אינה חזקה בפני חריגים ותמיד תקצה אפילו את הנקודות החריגות לאשכולות.
- תוצאות סבירות: בדרך כלל, ל- k-means יש אתחול אקראי של צנטרואידים (מרכזי אשכולות) מה שהופך אותו ללא דטרמיניסטי, ולכן עלול להוביל לתוצאות בלתי צפויות בכל פעם שהאשכול מחושב (נוצר).

ספריות שנשתמש בהן

לא לשכוח להתקין ספריות אלה באמצעות:

Pip install pandas, matplotlib, sklearn

נייבא את הספריות המתאימות ע"י מיקום הפקודות הבאות בראש הקובץ

```
import pandas as pd
import matplotlib as plt
from sklearn.cluster import kMeans
```

נתונים

הקובץ מכיל 5 רשומות על לקוחות:
מ"ז, מין, גיל, הכנסה שנתית (אלפי $),
הוצאות – ציון (1-100).

המטרה שלנו היא לקבץ לאשכולות על
מנת לזהות דפוסים בין הכנסה
להוצאה

```
# loading the dataset
dataset = pd.read_csv('incomeData.csv')
print(dataset.head())
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
X = dataset.iloc[:, [3, 4]].values  # Create a new array with columns 3,4
```

```
[
[39  15 ]
[81  15 ]
[6   16 ]
[77  16 ]
[40  17 ]
...
[83  137]
]
```

ב-iloc הערך הראשון מייצג אילו שורות לבחור, והערך השני מייצג אילו עמודות לבחור. מכיוון שאנו רוצים את עמודות הכנסה והוצאות, אנו בוחרים את העמודה הרביעית והחמישית (זכור שהאינדקס בפיתון מתחיל מ-0)

כיצד לקבוע כמה אשכולות לייצר (k)

נראה כיצד להשתמש בשיטת המרפק כדי למצוא את המספר האופטימלי של אשכולות.

זכרו שיש שיטות אחרות למצוא את הערך האופטימלי של k

בשיטת המרפק, מקבץ מבוצע על ידי שינוי ערכי k בתוך תחום נתון, ואז נמדוד את הביצועים.

אנו משתמשים ב- WCSS (בתוך אשכול סכום ריבועים).

WCSS הוא סכום המרחק בריבוע בין כל נקודה לבין מרכז הצביר (האשכול).

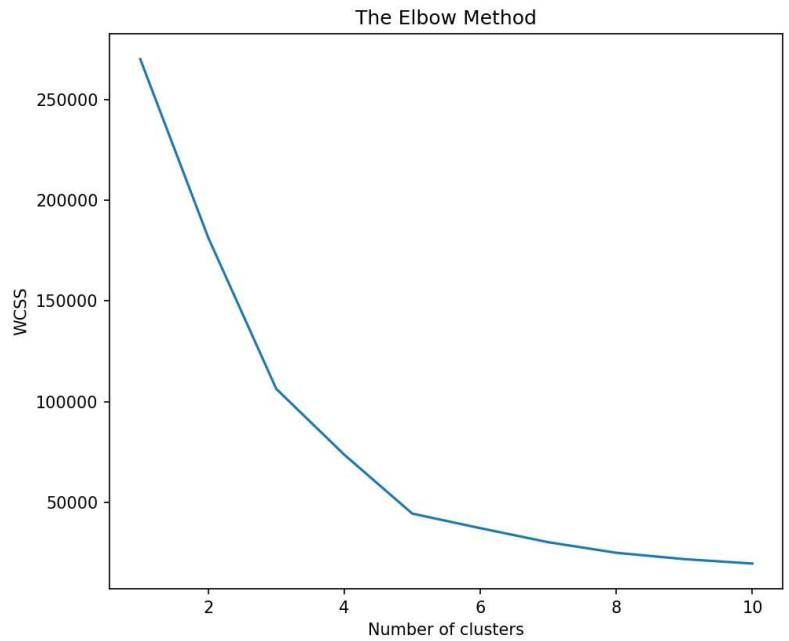
הדיאגרמה של WCSS (ציר y) והערך של k (ציר x) דומה לצורת מרפק.

מגרף זה, אנו מוצאים את הערך האופטימלי של k כאשר העקומה מתחילה להיות כמעט מקבילה לציר ה- x (להיות רוויה – saturated)

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Inertia : float
סכום המרחקים בריבוע של דגימות למרכז האשכול הקרוב ביותר שלהן

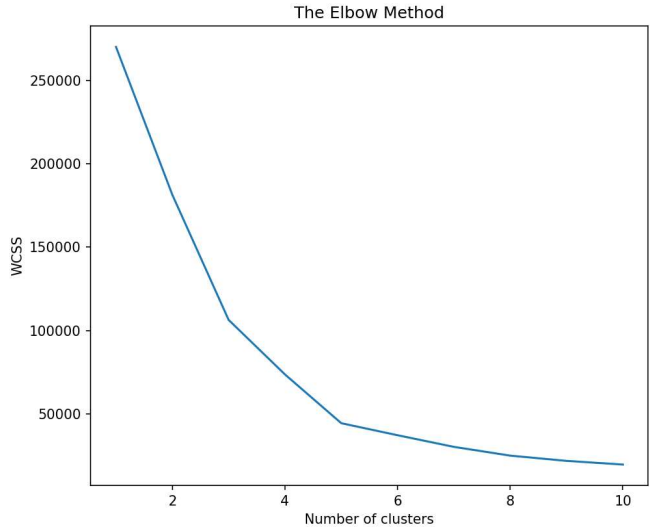


כיצד לקבוע כמה אשכולות לייצר (k)

כעת, לאחר שקבענו את המספר האופטימלי של אשכולות עבור מערך הנתונים שלנו. אנו יכולים לראות מהגרף שהערך האופטימלי של k=5

אנו יכולים לבצע את האשכול הסופי ולהקצות "תוויות פסאודו" לנקודות הנתונים שלנו כדי לבצע הדמיה ויזואלית של האשכולות.

תוויות פסאודו: אלו תוויות שניתנו לנקודות נתונים על ידי מדען הנתונים כדי להקצות לאיזה אשכול נקודה שייכת. בוא נראה את זה בפעולה.




```
# Cluster the data using k = 5
kmeans = KMeans(n_clusters = 5, random_state = 42)

# Assigning pseudo labels to the data points and printing them
y_kmeans = kmeans.fit_predict(X)
print(y_kmeans)
```

kmeans.fit_predict(X) - חישוב מרכזי אשכולות וחיזוי אינדקס אשכולות עבור כל תצפית במדגם

```
[2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
 3 2 3 2 3 2 0 2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 4 1 4 0 4 1 4 1 4 0 4 1 4 1 4 1 4 1 4 0 4 1 4 1 4
 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1
 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4]
```

תוויות פסאודו נעות בין 0-4 עבור 5 אשכולות, כל ערך מתאים לאשכול אחד

ויזואליזציה של האשכולות

הסבר לפקודות

X[y_kmeans == 4, 0]

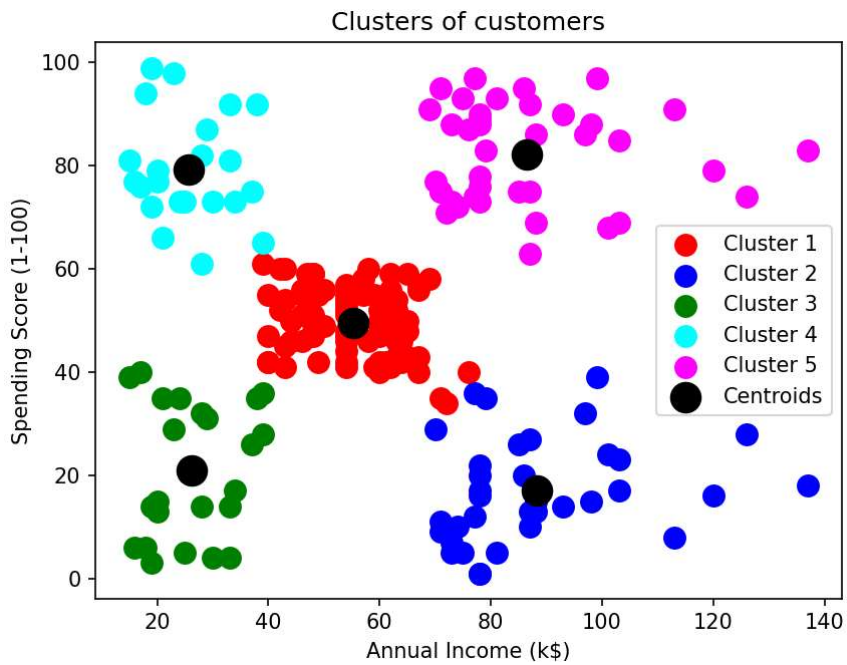
לוקח נקודות נתונים ממערך הנתונים שהוקצו להם פסאודו-תווית 4 ולוקח את ההכנסה השנתית (העמודה ה-0 ב- X)

X[y_kmeans == 4, 1]

הוא לוקח נקודות נתונים ממערך הנתונים שהוקצו להם פסאודו-תווית 4 ולוקח את ציון ההוצאות (העמודה ה-1 ב- X)

```
# s - size, c - color
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s=100, c='red', label='Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s=100, c='blue', label='Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s=100, c='green', label='Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s=100, c='cyan', label='Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s=100, c='magenta', label='Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=200, c='black',
                                                                        label='Centroids')

plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



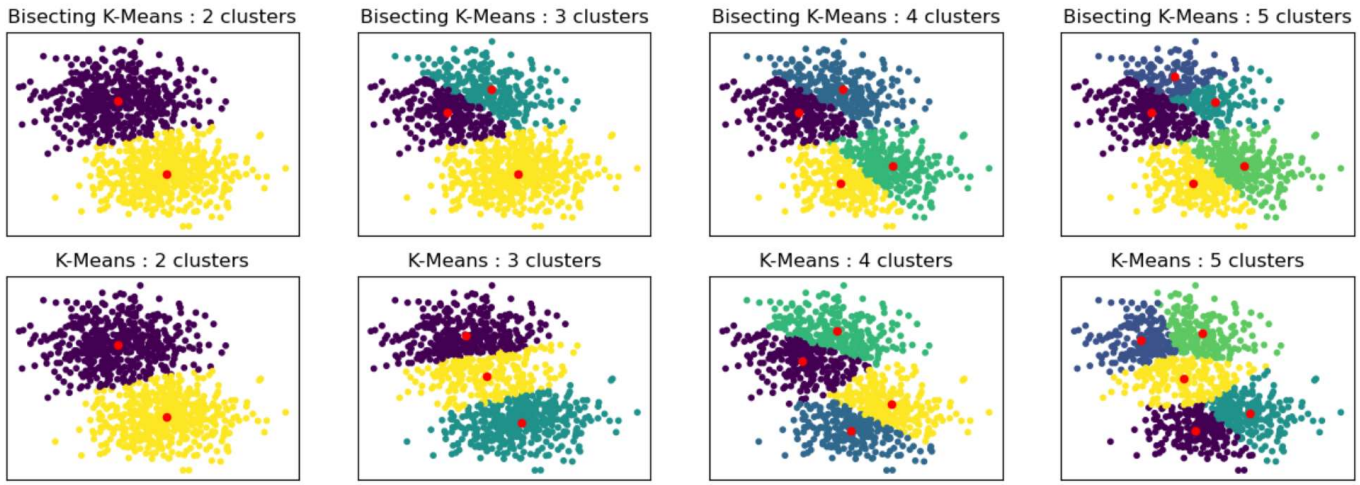
מרכזי הצבירים (אשכולות)

כמו כן, אובייקט המחלקה KMeans מאפשר לראות את מרכזי האשכולות ישירות על ידי קריאה ל- cluster_centers_ ישירות על אובייקט המחלקה.

מרכזי האשכולות שלנו מימין

```
[[55.2962963  49.51851852]
 [88.2        17.11428571]
 [26.30434783 20.91304348]
 [25.72727273 79.36363636]
 [86.53846154 82.12820513]]
```

השוואה בין Bisecting K-Means and Regular K-Means

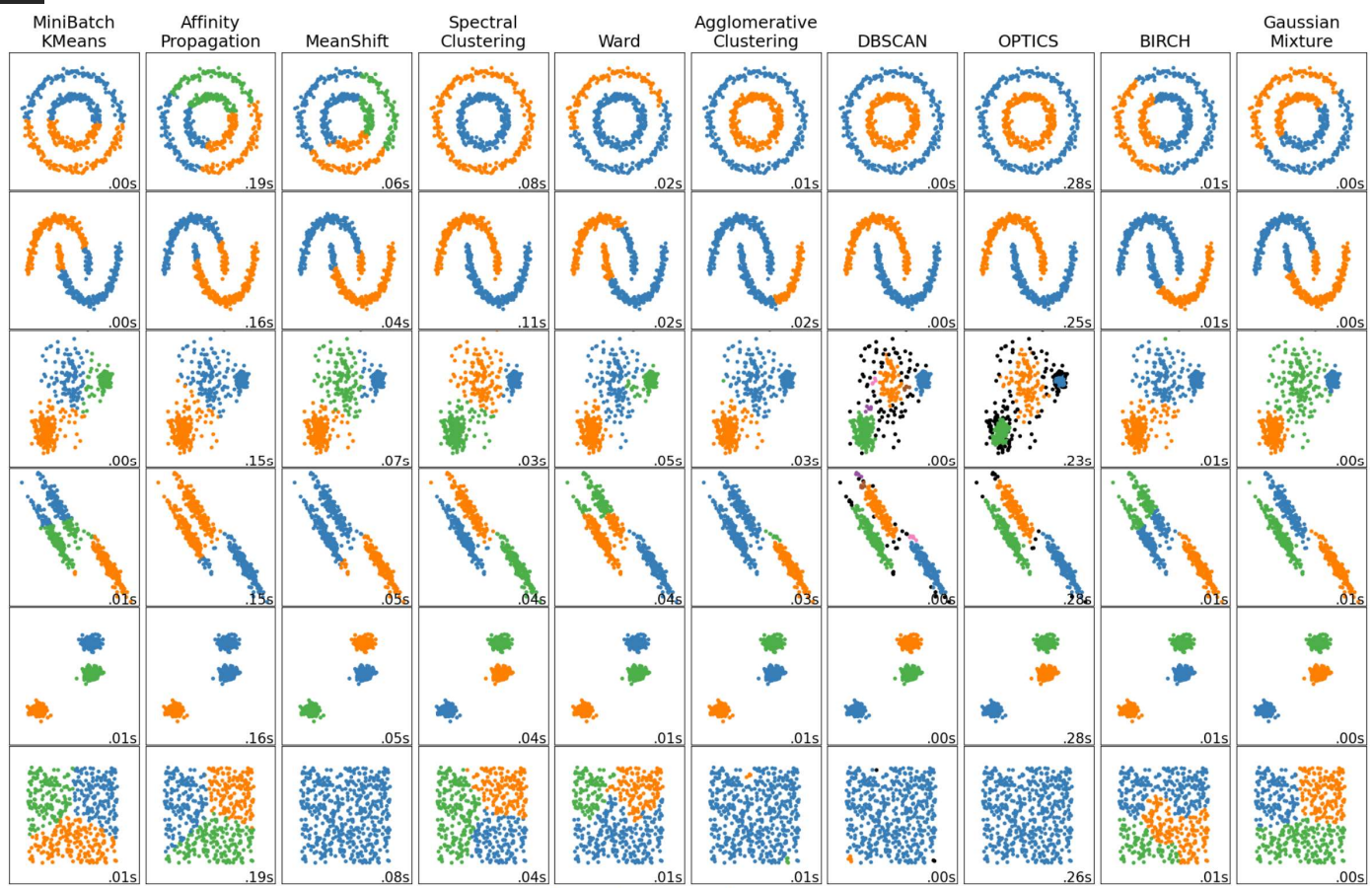


דוגמה זו מציגה הבדלים בין אלגוריתם K-Means רגיל לבין Bisecting K-Means.

בעוד שצבירי K-Means שונים כאשר הם בעלי n_clusters הולכים וגדלים, אשכולות Bisecting K-Means מתבססים על הקודמים.

ניתן לראות הבדל זה חזותית.

השוואה בין שיטות שונות של clustering



A comparison of the clustering algorithms in scikit-learn

השוואה בין שיטות שונות של clustering

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
OPTICS	minimum cluster membership	Very large n_samples, large n_clusters	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large n_clusters and n_samples	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points
Bisecting K-Means	number of clusters	Very large n_samples, medium n_clusters	General-purpose, even cluster size, flat geometry, no empty clusters, inductive, hierarchical	Distances between points

לסיכום

מה למדנו היום?

- הקדמה ל- kMeans
- תכונות של kMeans
- יישום באמצעות פייתון:
  - קריאה של הנתונים
  - כיצד לקבוע את הערך k שמציין את מספר מקבצים
  - הצגה ויזואלית של המקבצים

חשוב!

למדו את המושגים החדשים הם חשובים

להתראות בשבוע הבא!