

Kümeleme algoritmaları

K-means clustering

Gerçek Dünya Senaryolarında Kümeleme Uygulamaları

- Kümeleme endüstride yaygın olarak kullanılan bir tekniktir. Bankacılık ve [öneri motorlarından](#) belge kümeleme ve görüntü segmentasyonuna kadar hemen hemen her alanda kullanılıyor .

Müşteri segmentasyonu

- Bunu daha önce ele almıştık; kümelemenin en yaygın uygulamalarından biri müşteri segmentasyonudur. Üstelik bu sadece bankacılıkla sınırlı değil. Bu strateji telekom, e-ticaret, spor, reklamcılık, satış vb. gibi işlevler genelinde geçerlidir.

Belge Kümeleme

- Bu, kümelemenin başka bir yaygın uygulamasıdır. Diyelim ki birden fazla belgeniz var ve benzer belgeleri bir arada kümelemeniz gerekiyor. Kümeleme, bu belgeleri benzer belgeler aynı kümelerde olacak şekilde gruplandırmamıza yardımcı olur.



Document Clustering

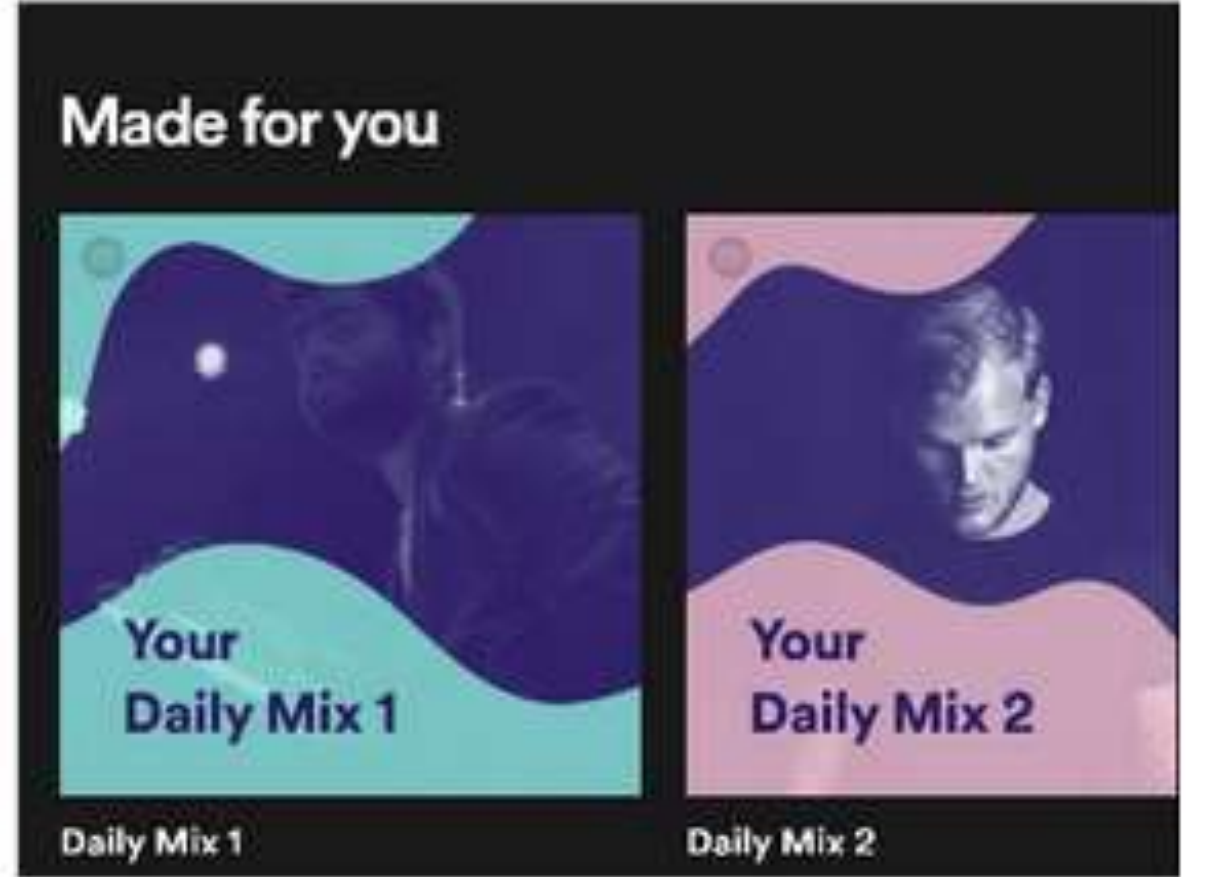
Resim parçalama

- Görüntü segmentasyonunu gerçekleştirmek için kümelemeyi de kullanabiliriz. Burada görüntüdeki benzer pikselleri bir araya getirmeye çalışıyoruz. Aynı grupta benzer piksellere sahip kümeler oluşturmak için kümeleme uygulayabiliriz.



Öneri Motorları

- [Kümeleme aynı zamanda öneri motorlarında](#) da kullanılabilir . Arkadaşlarınıza şarkı önermek istediğinizi varsayalım. Bu kişinin beğendiği şarkılara bakabilir, ardından benzer şarkıları bulmak için kümelemeyi kullanabilir ve son olarak en benzer şarkıları önerebilirsiniz.



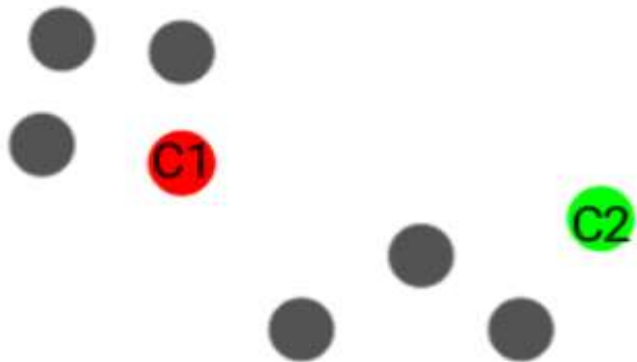
K- means algoritmasının yapım mantığı nedir?

1. Küme sayısını seçin k

K-ortalamlarda ilk adım küme sayısını (k) seçmektir.

2. Verilerden k rastgele noktayı merkez olarak seçin

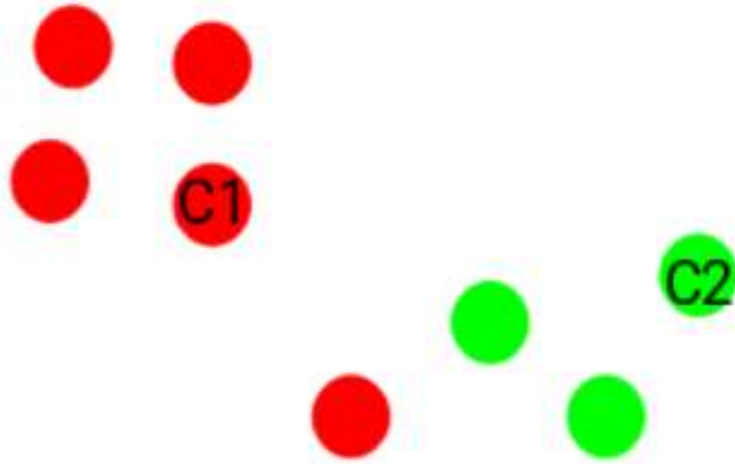
Daha sonra her kümenin merkezini rastgele seçiyoruz. Diyelim ki 2 kümeye sahip olmak istiyoruz, yani k burada 2'ye eşit. Daha sonra merkezini rastgele seçiyoruz:



Burada kırmızı ve yeşil daireler bu kümelerin ağırlık merkezini temsil ediyor.

3. Tüm noktaları en yakın küme merkezine atayın

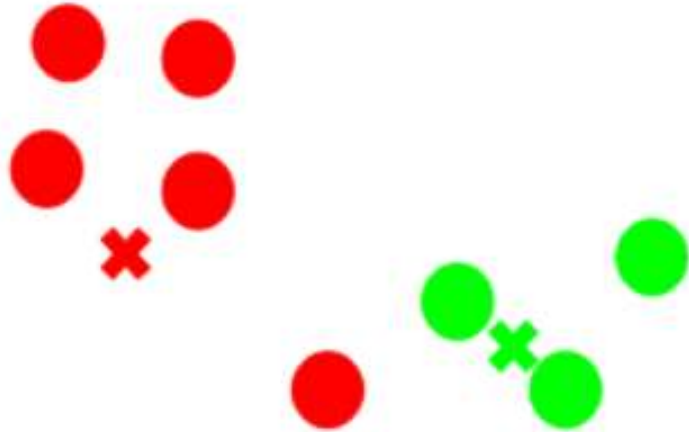
Merkezleri başlattığımızda, her noktayı en yakın küme merkezine atarız:



Burada kırmızı noktaya daha yakın noktaların kırmızı kümeye, yeşil noktaya daha yakın noktaların ise yeşil kümeye atandığını görebilirsiniz. .

4. **Yeni oluşturulan kümelerin ağırlık merkezlerini yeniden hesaplayın**

Şimdi, tüm noktaları her iki kümeye de atadıktan sonra, bir sonraki adım yeni oluşturulan kümelerin ağırlık merkezlerini hesaplamak olacaktır:



Burada kırmızı ve yeşil çarpılar yeni merkez noktalarıdır.

5. **3. ve 4. adımları tekrarlayın**

Daha sonra 3. ve 4. adımları tekrarlıyoruz:



Ağırlık merkezini hesaplama ve tüm noktaları ağırlık merkezine olan mesafelerine göre kümeye atama adımı tek bir yinelemedir . Ama durun – bu süreci ne zaman durdurmalıyız? Sonsuza kadar koşamaz, değil mi?

K-Means Kümelenmesine İlişkin Durdurma Kriterleri

K-means algoritmasını durdurmak için benimsenebilecek esasen üç durdurma kriteri vardır:

- Yeni oluşan kümelerin ağırlık merkezlerinin değişmemesi
- Puanlar aynı kümede kalması
- Maksimum yinleme sayısına ulaşılması

Yeni oluşan kümelerin ağırlık merkezlerinin değişmemesi

- Yeni oluşan kümelerin ağırlık merkezleri değişmiyorsa algoritmayı durdurabiliriz. Birden fazla yinelemeden sonra bile tüm kümeler için aynı ağırlık merkezlerini alıyorsak, algoritmanın yeni bir model öğrenmediğini ve bunun eğitimin durdurulması için bir işaret olduğunu söyleyebiliriz.

Puanların aynı kümede kalması

- Algoritmayı birden fazla yineleme için eğittikten sonra bile noktaların aynı kümede kalması, eğitim sürecini durdurmamız gerektiğinin bir başka açık işaretidir.

Maksimum yineleme sayısına ulaşılması

- Son olarak maksimum iterasyon sayısına ulaşıldığında eğitimi durdurabiliriz. Yineleme sayısını 100 olarak ayarladığımızı varsayalım. Süreç durmadan önce 100 yineleme boyunca tekrarlanacaktır.

Phyton kodunun çalıştırılması

The image shows a Python IDE interface with a dark theme. The main editor window displays a Python script for K-means clustering. The script imports necessary libraries, generates random data points, fits a K-means model, and visualizes the results. The script is titled 'k-means.py' and is being executed. The output is a scatter plot titled 'K-means Kümeleme' showing four clusters of data points in different colors (green, yellow, blue, and purple) with their respective centroids marked by red 'x's. The plot is displayed in a window titled 'Figure 1'. The IDE interface includes a sidebar with 'VARIABLES', 'WATCH', 'CALL STACK', and 'BREAKPOINTS' sections. The bottom status bar shows the current file is 'Python File (python)' and the Python version is '3.9.12 (base: conda)'.

```
1 import matplotlib.pyplot as plt
2 from sklearn.datasets import make_blobs
3 from sklearn.cluster import KMeans
4
5 # Adım 1: Veri kümesi oluşturun
6 # Rastgele bir veri kümesi oluşturmak için make_blobs kullanıyoruz
7 X, y = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)
8
9 # Adım 2: K-means modelini oluşturun ve veriye fit edin
10 # KMeans modelini oluşturuyoruz ve belirli bir k (küme sayısı) belirliyoruz
11 kmeans = KMeans(n_clusters=4)
12 kmeans.fit(X)
13
14 # Adım 3: Sonuçları görselleştirin
15 # Veri noktalarını ve merkezlerini (centroid) çiziyoruz
16 plt.scatter(X[:, 0], X[:, 1], s=50, c=kmeans.labels_, cmap='viridis')
17
18 # Küme merkezlerini çiziyoruz
19 centers = kmeans.cluster_centers_
20 plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, marker='x')
21 plt.title('K-means Kümeleme')
22 plt.xlabel('Özellik 1')
23 plt.ylabel('Özellik 2')
24 plt.show()
```

PS D:\Masaüstü\Halil\python> & 'c:\Users\eozen\anaconda3\python.exe' 'c:\Users\eozen\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '52984' '-' 'D:\Masaüstü\Halil\python\k-means.py'

Ln 25, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.12 (base: conda)