H3- Veri Madenciliği

Kümeleme Algoritmaları

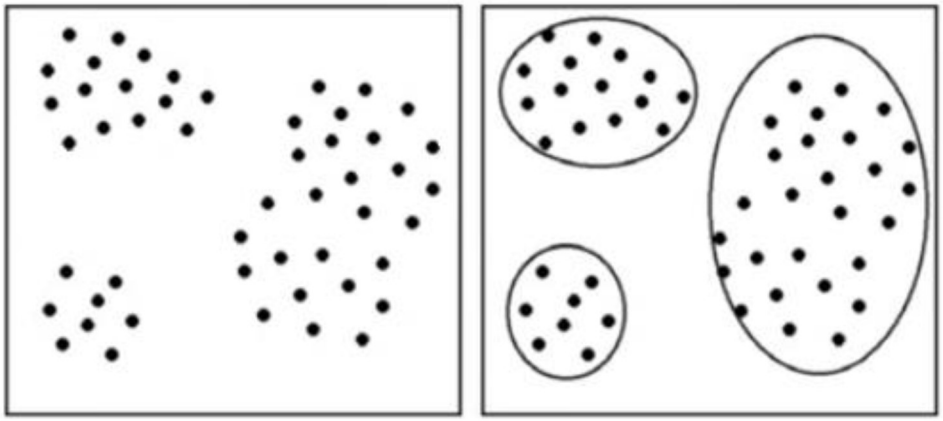
Dr. Öğr. Üyesi Alper Talha KARADENİZ

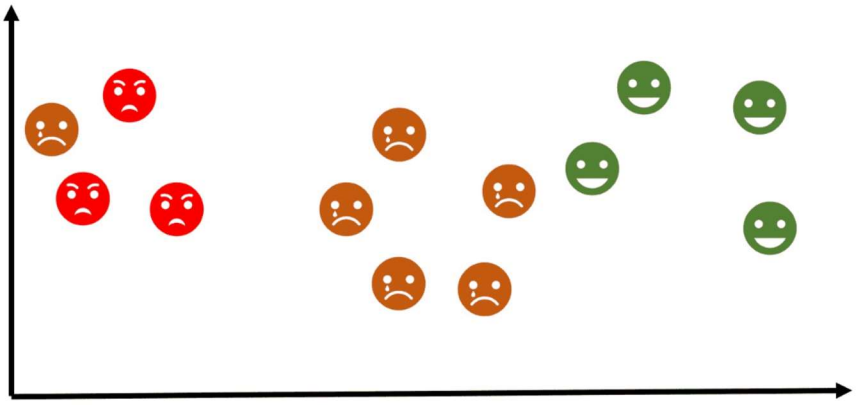
Samsun Üniversitesi

Yazılım Mühendisliği

Kümeleme Algoritmaları

- Küme, benzer nesnelerin oluşturduğu bir gruptur.
- Kümeleme, birbirine benzeyen nesnelerin aynı grupta toplanmasıdır.
- Aynı küme içinde benzerlikler fazla, kümeler arası benzerlikler ise azdır.





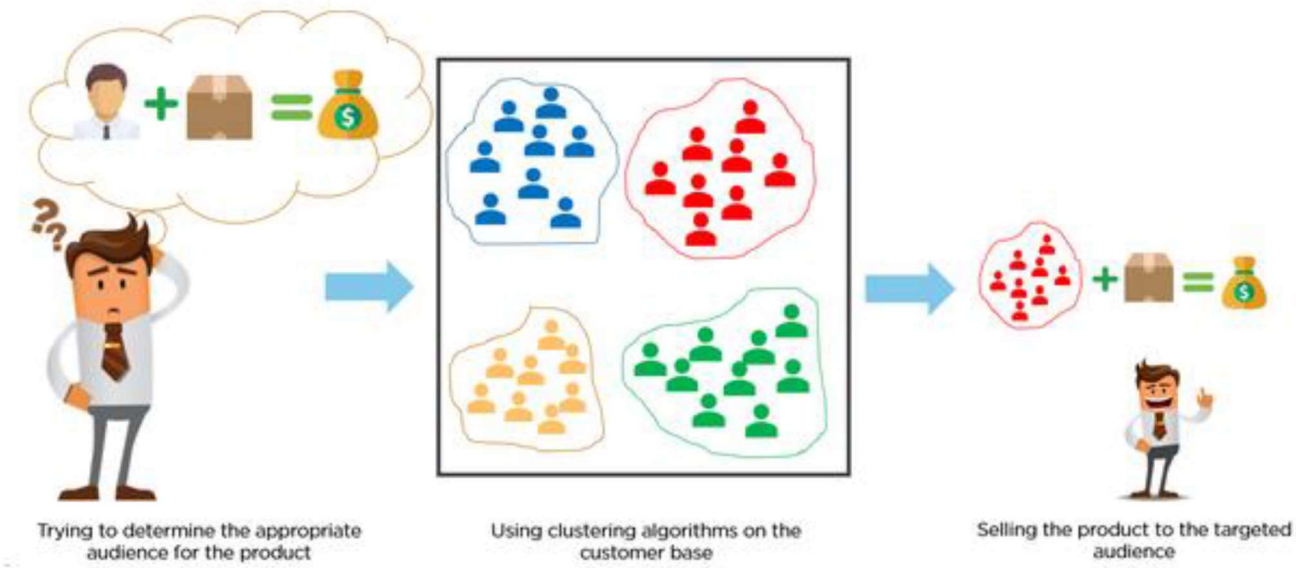
Bir küme temsil ettiği nesneleri en iyi şekilde ifade edecek biçimde düzenlenir. Kümeleme işleminin uygulandığı veri setindeki her bir veriye nesne adı verilir. Bu nesneler iki boyutlu düzlem üzerinde noktalarla gösterilir.

Kümeleme analizi, veri indirgeme veya nesnelerin doğal sınıflarını bulma gibi çeşitli amaçlarla kullanılmaktadır.

Bu alanlar;

- örüntü tanıma
- veri analizi
- resim tanıma
- pazarlama
- metin madenciliği
- doküman toplama
- istatistik araştırmaları
- makine öğrenimi
- şehir planlama
- coğrafik analizler (deprem, meteoroloji, yerleşim alanları)
- uzaysal veri tabanı uygulamaları
- Web uygulamaları
- müşteri ilişkileri yönetimi
- sağlık ve biyoloji alanında yapılan araştırmalardır.

- Kümeleyerek, datalar arasındaki ilginç desenler yakalanabilir.
- Pazarlamacıların kendi müşterileri arasındaki farklı grupları karakterize etmesi sağlanabilir.
- Biyolojide bitki ve hayvan taksonomilerini genlere göre sınıflandırmada kullanılır.
- Yeryüzü incelemelerinde belli toprak parçalarını tanımlamak için kullanılır.
- Web dokümanlarını sınıflamakta kullanılır.
- Bir hastalık veya sağlık durumu sık sık çeşitli varyasyonlar gösterir ve kümeleme analizi bu değişik çeşitlilikleri ortaya çıkarmada kullanılabilir.
- Örnek olarak kümeleme depresyonun değişik türlerinin belirlenmesinde kullanılmıştır.
- Kümeleme analizi aynı zamanda hastalıkların zaman ve mekanda dağılımı ile ilgili paternlerinortaya çıkarılmasında da kullanılabilir.



- Kümelenme, bir “denetimsiz öğrenme” problemi olarak düşünülebilir; etiketlenmemiş verilerden oluşan bir koleksiyonda bir yapı bulmakla ilgilenir.
- Kümelenme, “birbirine benzer üyeleri olan grupları, kümeler halinde düzenleme süreci” olarak tanımlanabilir. Bu nedenle bir küme, aralarında “benzerlik” bulunan ve diğer kümelere ait nesnelere “benzemeyen” bir nesne koleksiyonudur.
- Bir başka deyişle nesneler sadece benzerlik ölçümüne göre değil niteleyici kavrama uyuyorsa birlikte gruplanır.
- Kümelenmenin amacı, bir grup etiketlenmemiş veride içsel gruplamayı belirlemektir.
- Burada soru: neyin iyi bir kümelenme oluşturduğuna nasıl karar verileceğidir. Bir ölçüt belirlemek güçtür. Kriteri sağlaması gereken kullanıcıdır; kümelenin sonucu kullanıcının gereksinimine uymalıdır .

- Örneğin, iyi bir kümeleme homojen gruplar için temsilciler bularak sağlanabilir(veri azaltma/data reduction) ya da “doğal kümeler” bulma yoluna gidilebilir ve bilinmeyen özellikleri tanımlanabilir (“natural” data types), kullanışlı ve uygun gruplar bulma (“useful” data classes) veya olağandışı/istisnai veri nesneleri bulma (outlier detection) da olabilir.

Kümeleme algoritmaları birçok alanda kullanılabilmektedir.

- Pazarlamada: Müşterilerin özelliklerini ve geçmiş satın alma kayıtlarını içeren büyük bir veri tabanında benzer davranışa sahip müşterileri bulma.
- Biyoloji: Bitkilerin ve hayvanların özelliklerine göre sınıflanması.
- Kütüphanelerde: Kitap siparişinde.
- Sigortacılıkta ortalama hasar maliyeti yüksek olan sigorta sahiplerinin belirlenmesi; dolandırıcılıkların belirlenmesi.

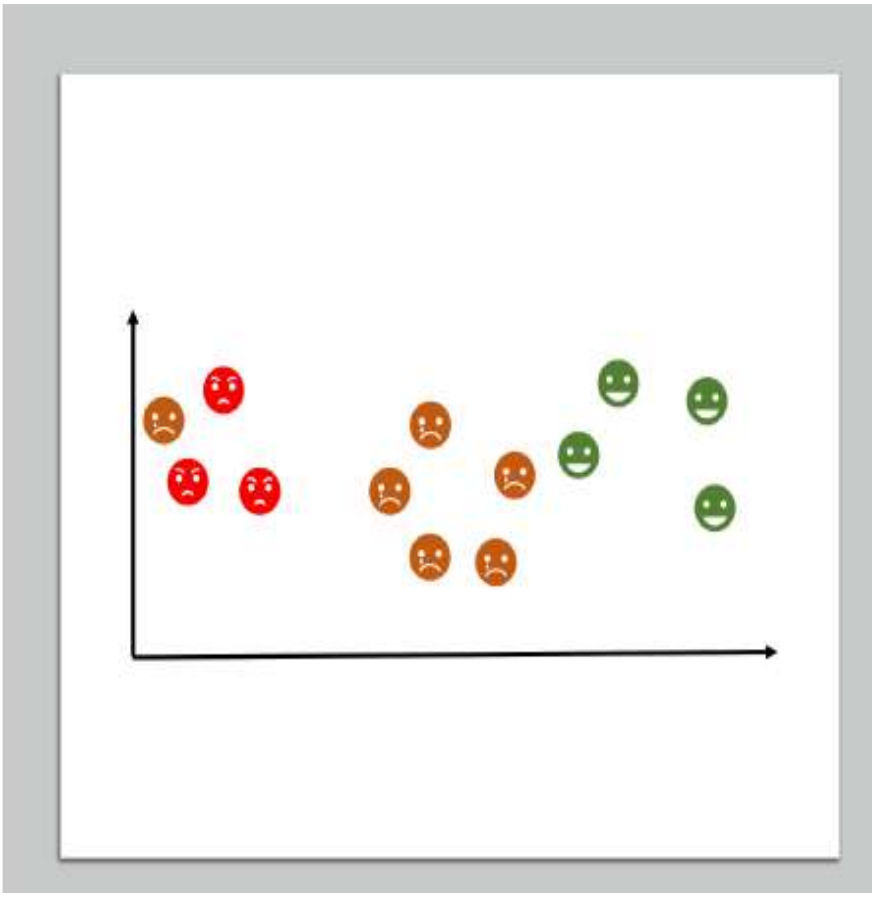
- Şehir Planlamacılığında: konut gruplarının konut türlerine, değerlerine ve coğrafi konumlarına göre belirlenmesi.
- Deprem çalışmalarında: tehlikeli bölgeleri tespit etmek için deprem merkez üslerinin gözlemlenmesi
- WWW'de doküman sınıflamak için:
- Blogların kümelenmesi için ve benzer erişim örüntüsü sergileyen grupları keşfetmek için.

Gereklilikler: Bir kümeleme algoritmasının olmazsa olmazları:

- Ölçeklenebilirlik
- Farklı niteliklerle/özelliklerle uğraşmak
- Rasgele şekilli kümeleri keşfetmek
- Girdi parametreleri belirlemek için bir alanın bilgisinin minimum gereklilikleri
- Gürültü ve aykırı değerlerle başa çıkma yeteneği
- Girdi kayıtlarının sırasına duyarsızlık
- Yüksek çok boyutluluk
- Yorumlanabilirlik ve kullanılabilirlik.

Kümeleme Yöntemleri

- Hiyerarşik Yöntemler
  - Birleştirici/Toplamalı Yöntemler
  - Ayırıcı/Bölünmeli Yöntemler
- Bölümlemeli Yöntemler
  - K-means
  - K-medoids
  - CLARA
- Yoğunluk Bazlı Yöntemler
- Grid Bazlı Yöntemler
- Model Bazlı Yöntemler



Hiyerarşik Yöntemler

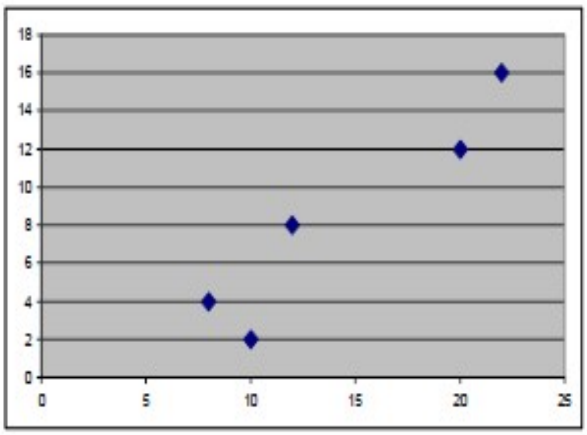
En yakın komşu algoritması

•En yakın komşu algoritmasında gözlemler arasında birbirine en yakın olanların uzaklığı iki kümenin birbirine olan uzaklığı olarak değerlendirilir.

•En düşük uzaklık seçilerek bu uzaklıkla ilgili elemanlar birleştirilip yeni bir küme elde edilir. Daha sonra uzaklıklar yeniden hesaplanır.

Tablo değerlerinden hareketle Tek Bağlantılı Hiyerarşik Kümeleme Yöntemi (en yakın komşu algoritması) kullanarak kümeleme işlemi yapalım.

hasta no	ilk ay için migren atak sayısı	atak süresi
1	8	4
2	12	8
3	10	2
4	20	12
5	22	16



Uzaklık tablosunu oluşturalım:

$$uzak(i,j) = \sqrt{\sum_{k=1}^{p} (x_{ij} - x_{jk})^2}$$

$$uzak(1,2) = \sqrt{(8-12)^2 + (4-8)^2} = 5.66$$

$$uzak(1,3) = \sqrt{(8-10)^2 + (4-2)^2} = 2.83$$

$$uzak(1,4) = \sqrt{(8-20)^2 + (4-12)^2} = 14.42$$

$$uzak(1,5) = \sqrt{(8-22)^2 + (4-16)^2} = 18.44$$

$$uzak(2,3) = \sqrt{(12-10)^2 + (8-2)^2} = 6.32$$

$$uzak(2,4) = \sqrt{(12-20)^2 + (8-12)^2} = 8.94$$

$$uzak(2,5) = \sqrt{(12-22)^2 + (8-16)^2} = 12.81$$

$$uzak(3,4) = \sqrt{(10-20)^2 + (2-12)^2} = 14.14$$

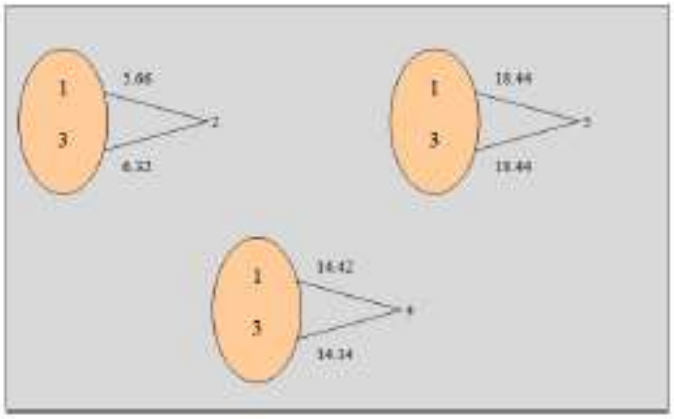
$$uzak(3,5) = \sqrt{(10-22)^2 + (2-16)^2} = 18.44$$

$$uzak(4,5) = \sqrt{(20-22)^2 + (12-16)^2} = 4.47$$

Elde edilen uzaklık matrisinde en düşük uzaklık 2.83 olup bu değere sahip 1 ve 3 nolu gözlemler birleştirilerek{1,3} kümesini elde ederiz.

Hasta no	1	2	3	4	5
1					
2	5.66				
3	2.83	6.32			
4	14.42	8.94	14.14		
5	18.44	12.81	18.44	4.47	

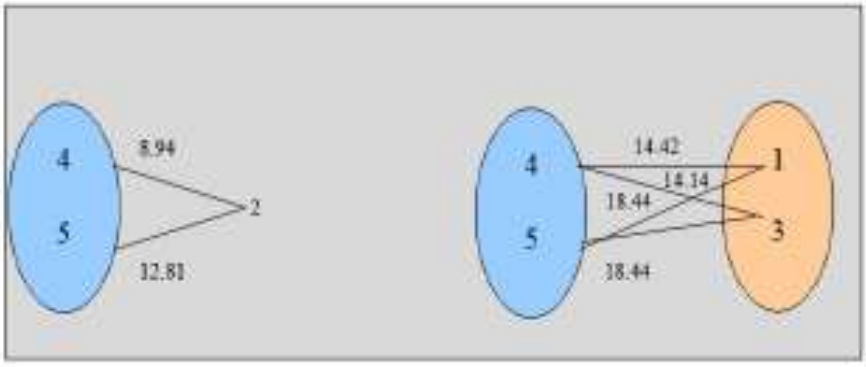
Daha sonra {1,3} kümesi ile diğer gözlemler arasındaki uzaklıklar hesaplanır.



Hasta no	{1,3}	2	4	5
{1,3}				
2	5.66			
4	14.14	8.94		
5	18.44	12.81	4.47	

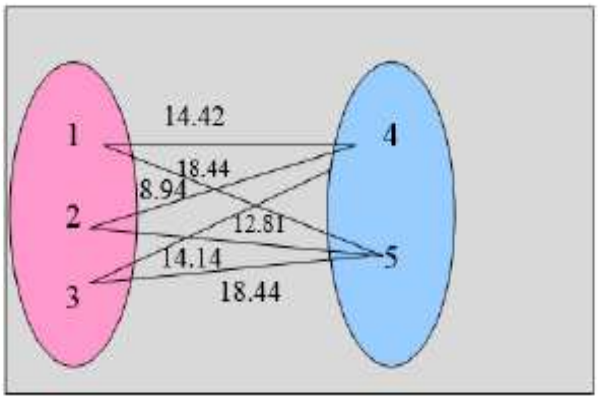
- En düşük uzaklık 4.47 olup bu değere sahip 4 ve 5 nolu gözlemler birleştirilerek {4,5} kümesini elde ederiz.
- {4,5} ile 2 gözlemi arasındaki mesafe 8.94, {1,3} ile arasındaki mesafe 14.14 tür.

•Yeni uzaklık tablosundaki en düşük uzaklık 5.66 olup bu değere sahip {1,3} ve 2 nolu gözlemler birleştirilerek {1,2,3} kümesini elde ederiz.



Hasta no	{1,3}	2	{4,5}
{1,3}			
2	5.66		
{4,5}	14.14	8.94	

{1,2,3} kümesi ile diğer uzaklıkların hesaplanması ve yeni uzaklık tablosu:

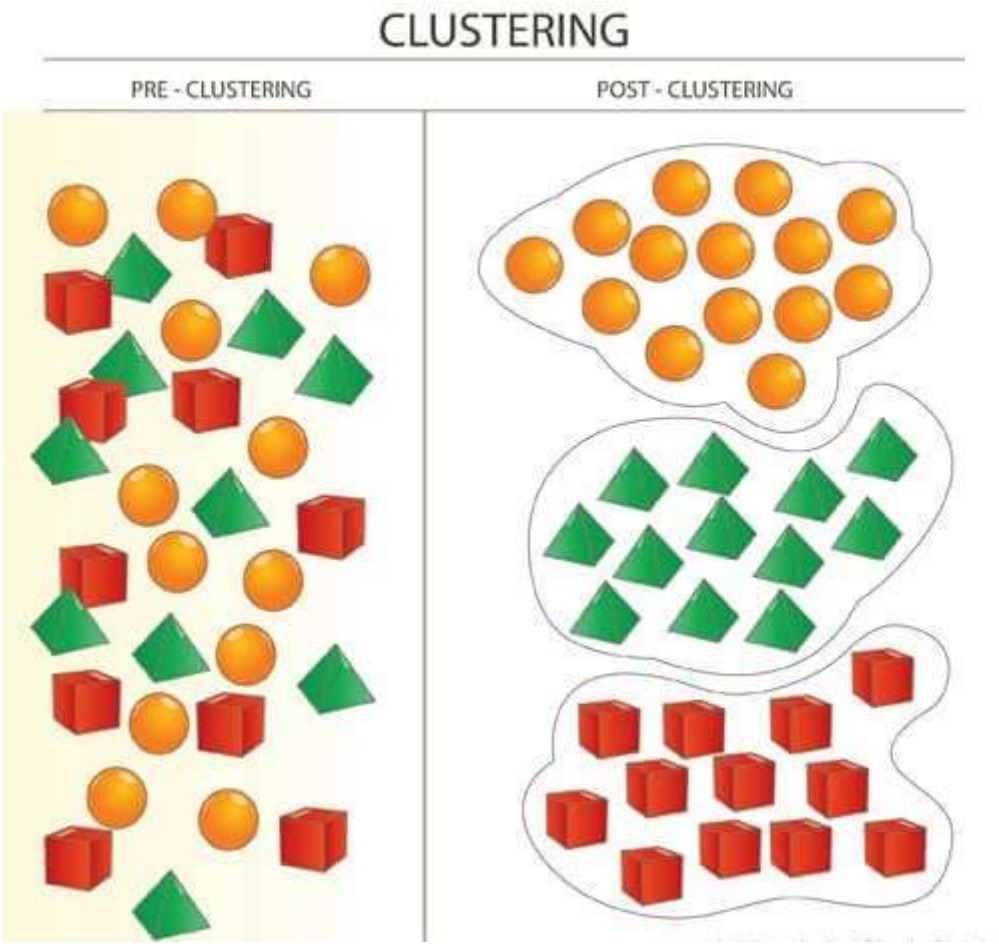


Hasta no	{1,2,3}	{4,5}
{1,2,3}		
{4,5}	8.94	

Elde edilen iki küme birleştirilerek sonuç küme bulunur. Bu küme {1,2,3,4,5} gözlemlerinden oluşan kümedir. Uzaklık düzeyi göz önüne alındığında kümeler şu şekilde belirlenmiştir:

Uzaklıklar	Kümeler
2.83	{1,3}
4.47	{4,5}
5.66	{1,2,3}
8.94	{1,2,3,4,5}

K-Means



Kümeleme, verilerin yapısı hakkında bir sezgiyi elde etmek için kullanılan en yaygın keşifsel veri analizi tekniklerinden biridir. Farklı kümelerdeki veri noktaları çok farklıyken, aynı alt gruptaki (küme) veri noktalarının çok benzer olması nedeniyle verilerdeki alt grupların belirlenmesi görevi olarak tanımlanabilir.

K-means kümeleme, denetimsiz öğrenme için kullanılan bir makine öğrenimi algoritmasıdır. Veri noktalarının

benzerliğine dayalı olarak verileri önceden tanımlanmış sayıda kümeye (*k*) kümeleme yöntemidir.

K-Means algoritması birçok alanda kullanılabilmektedir.

- ***Belge Sınıflandırması***
  Belgeleri etiketlere, konulara ve belgenin içeriğine göre birden fazla kategoride kümeleyin. Bu çok standart bir sınıflandırma problemidir ve k-means aracı bu amaç için oldukça uygun bir algoritmadır.

- **Suç Yerlerinin Belirlenmesi**
  Bir şehirdeki belirli bölgelerde mevcut olan suçlarla ilgili veriler, suç kategorisi, suç alanı ve ikisi arasındaki ilişki, bir şehirdeki ya da bölgedeki suça eğilimli alanlara ilişkin kaliteli bilgiler verebilir.
- **Müşteri Segmentasyonu**
  Kümeleme, pazarlamacıların müşteri tabanını geliştirmelerine, hedef alanlarda çalışmasına ve müşterileri satın alma geçmişine, ilgi alanlarına veya etkinlik izlemeye göre segmentlere ayırmasına yardımcı olur. Sınıflandırma, şirketin belirli

kampanyalar için belirli müşteri kümelerini hedeflemesine yardımcı olur.

- **Oyuncu Analizi**
  Oyuncu istatistiklerini analiz etmek, spor dünyasının her zaman kritik bir unsuru olmuştur ve artan rekabetle birlikte, makine öğrenmenin burada oynayacağı kritik bir rol vardır.
- **Dolandırıcılık Tespiti**
  Makine öğrenimi sahtekarlık tespitinde önemli bir rol oynar ve otomobil, sağlık ve sigorta sahtekarlığı tespitinde sayısız uygulamaya sahiptir. Sahte iddialarla ilgili geçmiş verileri kullanarak, yeni iddiaları , sahte kalıpları belirten kümelere yakınlığına dayanarak izole etmek mümkündür.
- **Çağrı Kaydı Detay Analizi**
  Bir çağrı detay kaydı (CDR), telekom şirketleri tarafından bir müşterinin araması, SMS ve internet etkinliği sırasında elde edilen bilgilerdir. Bu bilgiler, müşteri demografisiyle birlikte kullanıldığında, müşterinin ihtiyaçları hakkında daha fazla bilgi sağlar. Müşteri segmentlerini saatlerce kullanımlarına göre anlamak için kullanılır.

- **BT Uyarılarının Otomatik Kümelenmesi**
  Ağ, depolama veya veritabanı gibi büyük kurumsal BT altyapı teknolojisi bileşenleri büyük hacimli uyarı mesajları üretir. Uyarı mesajları potansiyel olarak operasyonel sorunlara işaret ettiğinden, sonraki işlemler için önceliklendirme için manuel olarak taranmaları gerekir. Verilerin kümelenmesi, uyarı kategorileri hakkında bilgi verebilir ve ortalama onarım süresi ve arıza tahminlerinde yardımcı olabilir.

K-Means Algoritması temel olarak 5 aşamadan oluşmaktadır:

1. **Küme Sayısının (*K*) Belirlenmesi**: Çalışmaya başlamadan önce belirlenen küme sayısı (*K*) seçilir. Bu, verilerin kaç küme içinde gruplanacağını belirler.
2. **Uzaklıkların Hesaplanması**: Merkez dışındaki verilerin k merkezlere mesafesi ölçülür.
3. **Kümeleme**: Her bir gözlemin mesafelerine göre kümeleme işlemi yapılır.
4. **Merkez Hesaplama**: Veriler en yakın k merkeze atandıktan sonra oluşan kümeler için yeniden merkez hesaplamaları yapılır.
5. **Küme Sayısı Seçilir**: 4. aşamada belirlenen iterasyon sayısı kadar tekrar yapılan işlemler sonucunda, küme içi hata kareler toplamlarının toplamının minimum olduğu durumdaki gözlemlerin kümelenme yapısı, nihai kümeleme olarak seçilir.

Örnek:

Elimizde, iki boyutlu düzlemde 6 adet nokta olsun:

$$P_1 = (1,1), \quad P_2 = (2,1), \quad P_3 = (4,3), \quad P_4 = (5,4), \quad P_5 = (8,7), \quad P_6 = (9,7)$$

Bu veri noktalarını $k = 2$ olacak şekilde kümelemek istediğimizi varsayalım.

Küme Sayısı Seçimi

$k = 2$. (İki küme: C_1 ve C_2.)

K-means'te merkezler (centroid) genelde rastgele seçilir. Kolaylık olsun diye:

- $\mu_1^{(0)} = P_1 = (1,1)$
- $\mu_2^{(0)} = P_5 = (8,7)$

> **Burada $\mu_j^{(0)}$ ifadesi, j. kümenin 0. iterasyondaki (yani başlangıçtaki) merkezini gösterir.**

Her nokta için, mevcut merkezlere **öklid uzaklığı** (d) hesaplanır ve nokta, en kısa mesafe hangi merkeze ise o kümeye atanır.

Uzaklık Formülü

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- $\mu_1^{(0)} = (1,1)$
- $\mu_2^{(0)} = (8,7)$

$P_1 = (1,1)$

- $d(P_1, \mu_1^{(0)}) = \sqrt{(1-1)^2 + (1-1)^2} = 0$
- $d(P_1, \mu_2^{(0)}) = \sqrt{(1-8)^2 + (1-7)^2} = \sqrt{(-7)^2 + (-6)^2} = \sqrt{49+36} = \sqrt{85} \approx 9.22$

En yakın merkez: μ_1. $P_1 \in C_1$.

$P_2 = (2,1)$

- $d(P_2, \mu_1^{(0)}) = \sqrt{(2-1)^2 + (1-1)^2} = \sqrt{1+0} = 1$
- $d(P_2, \mu_2^{(0)}) = \sqrt{(2-8)^2 + (1-7)^2} = \sqrt{(-6)^2 + (-6)^2} = \sqrt{36+36} = \sqrt{72} \approx 8.49$

En yakın merkez: μ_1. $P_2 \in C_1$.

$P_3 = (4,3)$

- $d(P_3, \mu_1^{(0)}) = \sqrt{(4-1)^2 + (3-1)^2} = \sqrt{3^2 + 2^2} = \sqrt{9+4} = \sqrt{13} \approx 3.61$
- $d(P_3, \mu_2^{(0)}) = \sqrt{(4-8)^2 + (3-7)^2} = \sqrt{(-4)^2 + (-4)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$

En yakın merkez: μ_1. $P_3 \in C_1$.

$P_4 = (5,4)$

- $d(P_4, \mu_1^{(0)}) = \sqrt{(5-1)^2 + (4-1)^2} = \sqrt{4^2 + 3^2} = \sqrt{16+9} = \sqrt{25} = 5$
- $d(P_4, \mu_2^{(0)}) = \sqrt{(5-8)^2 + (4-7)^2} = \sqrt{(-3)^2 + (-3)^2} = \sqrt{9+9} = \sqrt{18} \approx 4.24$

En yakın merkez: μ_2. $P_4 \in C_2$.

$P_5 = (8,7)$

- $d(P_5, \mu_1^{(0)}) = \sqrt{(8-1)^2 + (7-1)^2} = \sqrt{7^2 + 6^2} = \sqrt{49+36} = \sqrt{85} \approx 9.22$
- $d(P_5, \mu_2^{(0)}) = \sqrt{(8-8)^2 + (7-7)^2} = \sqrt{0+0} = 0$

En yakın merkez: μ_2. $P_5 \in C_2$.

(Zaten başlangıç olarak da bu noktayı seçmiştik.)

$P_6 = (9, 7)$

- $d(P_6, \mu_1^{(0)}) = \sqrt{(9-1)^2 + (7-1)^2} = \sqrt{8^2 + 6^2} = \sqrt{64 + 36} = \sqrt{100} = 10$
- $d(P_6, \mu_2^{(0)}) = \sqrt{(9-8)^2 + (7-7)^2} = \sqrt{1 + 0} = 1$

En yakın merkez: μ_2. $P_6 \in C_2$.

İlk iterasyon sonucu :

- $C_1 = \{P_1, P_2, P_3\} = \{(1,1), (2,1), (4,3)\}$
- $C_2 = \{P_4, P_5, P_6\} = \{(5,4), (8,7), (9,7)\}$

Merkezleri Güncelleme:

Yeni merkezler, her kümenin **ortalama noktası** (mean) olarak güncellenir.

- $\mu_1^{(1)}$:

$$\mu_1^{(1)} = \left(\frac{1+2+4}{3}, \frac{1+1+3}{3} \right) = \left(\frac{7}{3}, \frac{5}{3} \right) = (2.\overline{3}, 1.\overline{6})$$

- $\mu_2^{(1)}$:

$$\mu_2^{(1)} = \left(\frac{5+8+9}{3}, \frac{4+7+7}{3} \right) = \left(\frac{22}{3}, \frac{18}{3} \right) = (7.\overline{3}, 6)$$

Yeni merkezler:

$$\mu_1^{(1)} = (2.33\ldots, 1.66\ldots), \quad \mu_2^{(1)} = (7.33\ldots, 6)$$

Her noktayı tekrar ölçelim (burada kısa geçeceğiz, sadece sonuç odaklı gösterelim):

- P_1 ve P_2 önceki konumlarına yakın olduğu için $\mu_1^{(1)}$'e daha yakın kalmaya devam eder.
- P_3 (4,3), $\mu_1^{(1)}$'e mi yakın yoksa $\mu_2^{(1)}$'e mi yakın?
- P_4 (5,4), tekrar kontrol edip hangi merkeze daha yakınsa ona atanır.
- P_5 (8,7) ve P_6 (9,7) muhtemelen $\mu_2^{(1)}$ etrafında kalacaktır.

Hızlı hesaplama yaptığımızda, kümelerde bir değişiklik olmayacağını öngörebiliriz. (İsterseniz tek tek uzaklıkları yine elle veya kısa hesapla teyit edebilirsiniz.)

Kümeler değişmezse, merkezler aynı kalır. Dol ıyla **algoritma durağan hâle gelmiştir** (konverjans).

Sonuç olarak:

- $C_1 = \{(1,1), (2,1), (4,3)\}$
- $C_2 = \{(5,4), (8,7), (9,7)\}$

ve merkezler:

- $\mu_1 = (2.33\ldots, 1.66\ldots)$
- $\mu_2 = (7.33\ldots, 6)$

Bu örnek uygulamayı Python ile gerçekleyelim.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
# Veri noktalarını numpy dizisi haline getirelim
X = np.array([
    [1, 1],  # P1
    [2, 1],  # P2
    [4, 3],  # P3
    [5, 4],  # P4
    [8, 7],  # P5
    [9, 7]   # P6
])
# KMeans modelini oluştur (k=2), rastgelelik için random_state=42 ekleyelim
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(X)
# Her noktanın hangi kümede olduğunu alalım
labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

```
print("Küme Etiketleri:", labels)
print("Merkezler (centroids):\n", centroids)
# Veri noktalarını çizdirme
plt.figure(figsize=(6, 5))
colors = ["red", "blue"]
for i in range(len(X)):
    plt.scatter(X[i, 0], X[i, 1],
                color=colors[labels[i]],
                s=100, edgecolors="black")
# Küme merkezlerini çizdirme
plt.scatter(centroids[:, 0], centroids[:, 1],
            color="green", marker="X", s=200,
            label="Merkezler")
plt.title("K-Means (k=2) Örneği")
plt.xlabel("X koordinatı")
plt.ylabel("Y koordinatı")
plt.legend()
plt.grid(True)
plt.show()
```

CLARA

•Küçük ölçekli veritabanlarında kullanılan k-medoid yerine büyük veritabanlarında CLARA kullanılır.

•Temel fikir, tüm veriyi değerlendirmek yerine, tüm veriyi temsil eden ufak bir kesit alınarak analiz yapılmasıdır. Bu kesit rasgele bir şekilde bulunur.

Örneğin 1.000.000 lukbir kayıt dizisinde 100. , 1000. , 1300., 150000. kayıtlar.

•CLARA metodunun etkisi ve kalitesi, boyuta ve rasgele seçilen verilerin ne kadar iyi seçildiğine bağlıdır.

•CLARA metodu, alınan örnek verilere fazla bağlı olduğu için CLARANS adlı bir metot geliştirilmiştir. CLARANS da örnek bir nesne alınır ve algoritma bir kez geliştirilir, algoritma tekrarlanırken nesne de değiştirilir. CLARANS metodu ile daha kaliteli bir sonuç elde edilir ancak n2 oranında daha maliyetli bir yoldur.