VERİ MADENCİLİĞİNDE KÜMELEME

HAZIRLAYAN: Melike Akalan

İçerik

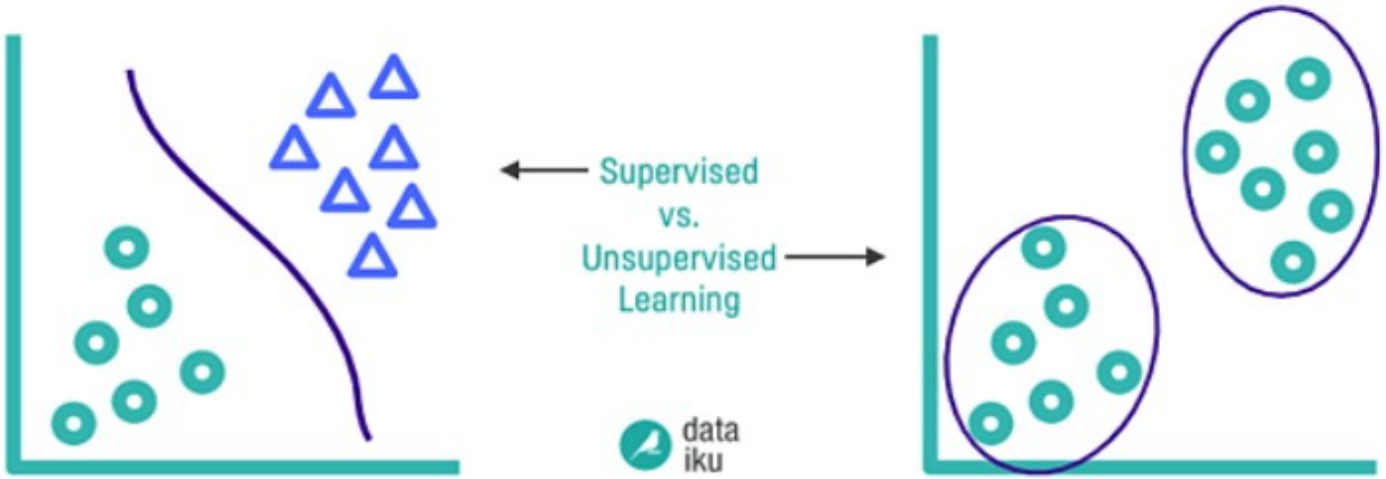
- Kümeleme Nedir?
- Uygulama Alanları
- Yöntemleri
- Kullanılan Algoritmalar
- Algoritmaların Avantajları/Dezavantajları
- Performans Ölçütleri
- Sonuç ve Yorum

Kümeleme

- Veri madenciliğinde kullanılan bir modelleme yöntemidir.
- Büyük veri içerisindeki benzer özellikli verileri gruplara(kümelere) ayırma işlemidir.
- Veri setindeki etiketlenmemiş veriler arasındaki gizli ilişkileri ortaya çıkarır.
- Üzerinde işlem yapılacak veriler işlenmemiş, ham verilerdir (raw data).

Kümeleme

- Denetimsiz öğrenme (unsupervised learning) yöntemidir.
- Yani veri hakkında önceden bilgi sahibi değilizdir.

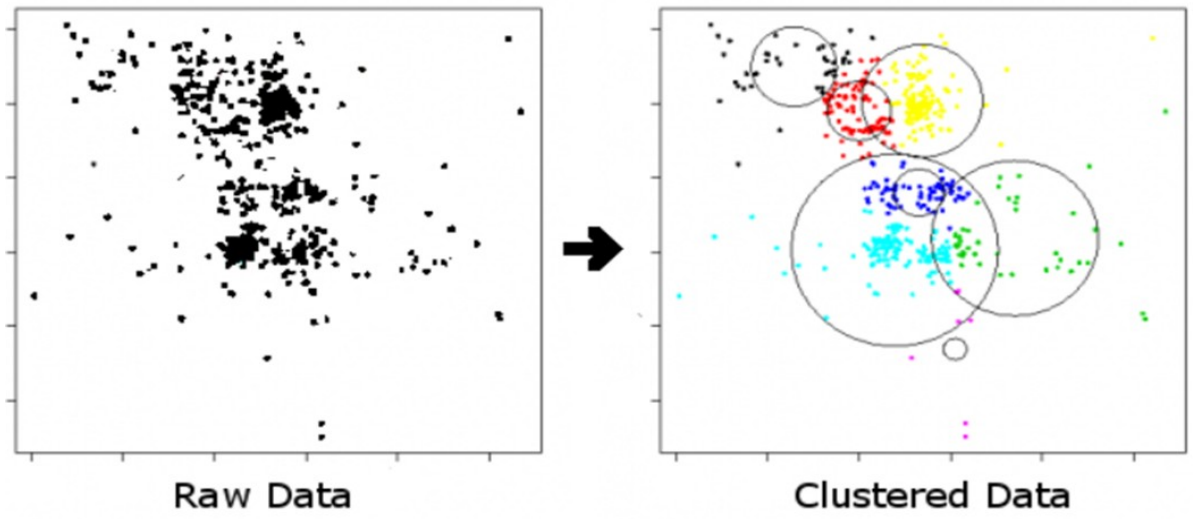


Uygulama Alanları

- **Akan veri analizi** çalışmalarında,
- **Sağlık,**
- **Genetik,**
- **Pazarlama, sigortacılık,**
- **Tavsiye sistemleri,**
- **Sahtekarlık tespiti** gibi pek çok farklı alanda kümeleme yaklaşımı kullanılır.

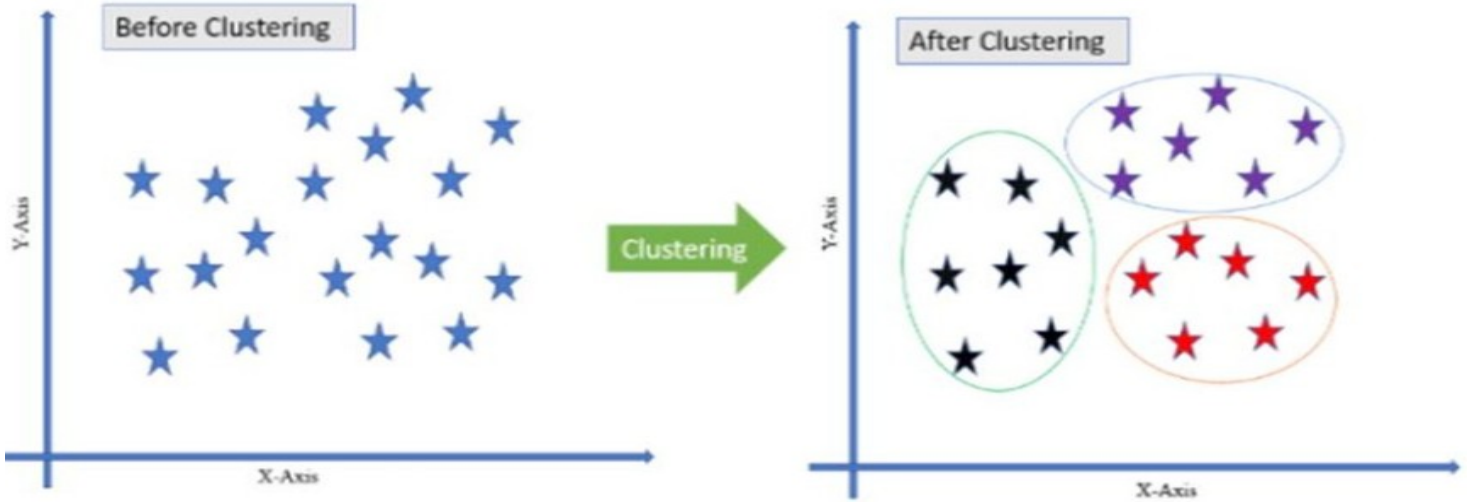
Kümeleme

- Kümeleme(clustering), verilerin uzaklık ölçütüne göre ilgili kümelere atanması işlemidir.
- Her veri nokta vb. şekil ile ifade edilir.
- Aynı küme içerisindeki veriler birbirlerine daha yakındır.



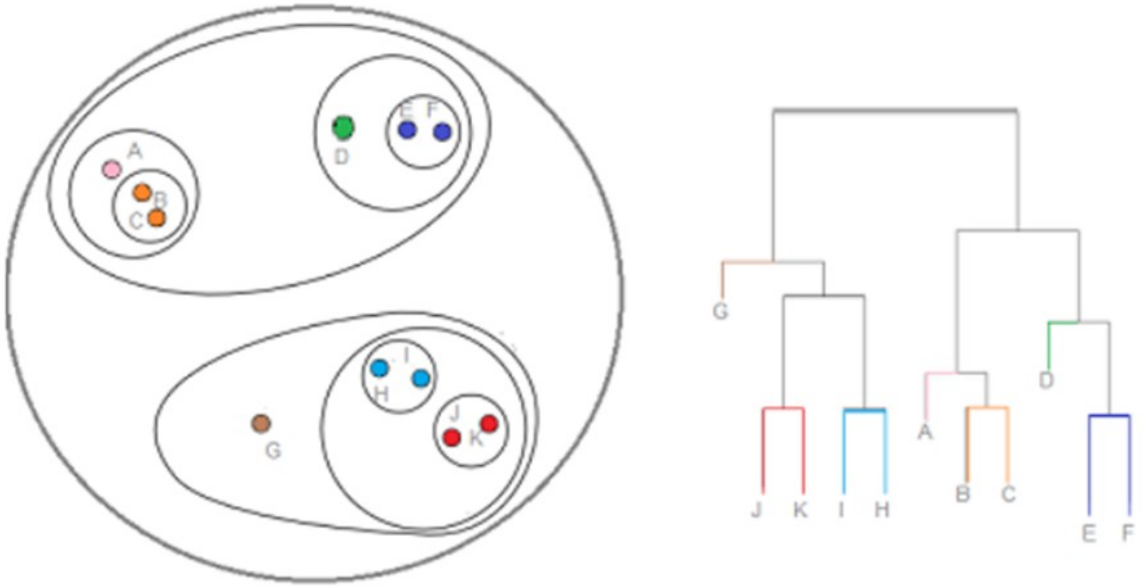
Kümeleme Yöntemleri

- **Ayrıştırmalı(Bölütleme) Kümeleme**
  Verilerin alt kümelere ayrılmasıdır. Her bir veri alt kümelerin yalnızca birinde bulunabilir.



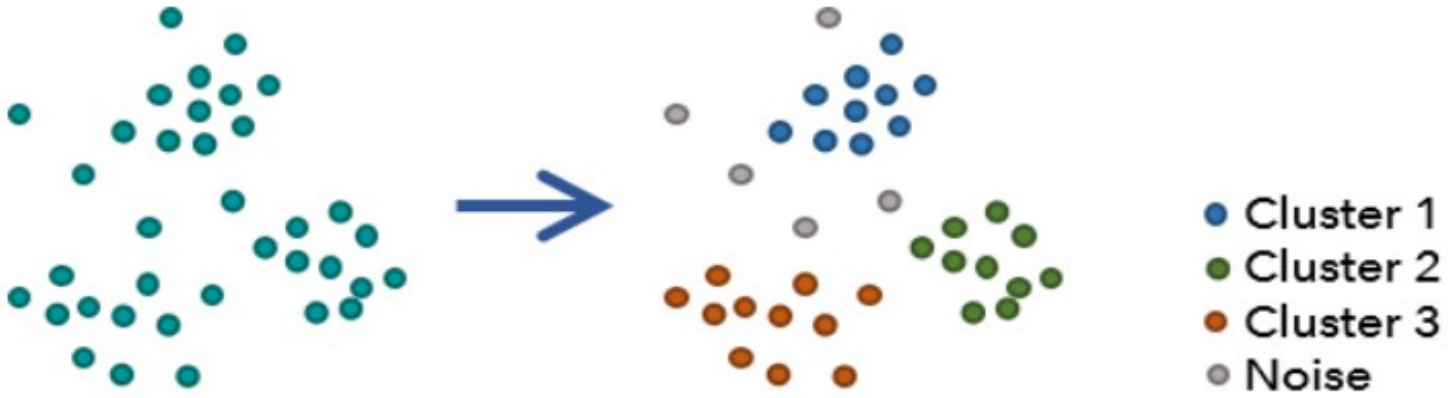
Kümeleme Yöntemleri

- **Hiyerarşik Kümeleme**
  Bir hiyerarşik ağaç gibi kümelerin iç içe dizilmesidir.



Kümeleme Yöntemleri

- **Yoğunluk Tabanlı Kümeleme**
  Veriler yoğun olduğu bölgelere göre kümelendirilir. Yoğunluğu az olanlar ise Gürültü(noise) verileri ya da küme sınırını oluşturan verilerdir.



Algoritmalar

- Ayrıştırmalı Kümeleme Algoritmaları:
  K-Means, K-Medoids
- Hiyerarşik Kümeleme Algoritmaları:
  Agnes, Diana
- Yoğunluk Tabanlı Kümeleme Algoritmaları:
  *Dbscan(*Density-based spatial clustering of applications with noise), *Optics, Clique, Denclue*

Küme Sayısının (k değeri) Belirlenmesi

- Hiyerarşik yöntem verilerin birbirlerinin yakınlık durumuna göre küme oluşturur.
- Yoğunluk tabanlı yöntem de verilerin yoğunluğuna göre küme oluşturur.
- Dolayısıyla bu iki yöntemde k değerinin önceden bilinmesine gerek yoktur.
- K-Means algoritmasında ise kullanılacak verinin bölüneceği küme sayısını, kullanıcı kendisi belirler.

K-Means

- K değerinin belirlenmesi için çoğunlukla elbow methodu kullanılır.
- Elbow methodu kümeler arasındaki en uzak mesafeleri hesaplayarak bunu grafikle gösterir.
- Grafikteki kırılma noktası k değeri olarak seçilir.

K-Means

1. Küme merkezlerini (c) rastgele seç.
2. Her veri için küme merkezleri arasındaki uzaklığı hesapla.
3. Seçilen veriyi hangi küme merkezine daha yakınsa o kümeye ata.
4. Verinin (x) atanmış olduğu küme merkezini (v) formüle göre güncelle.

$$v_i = (1 / c_i) \sum_{j=1}^{c_i} x_i$$

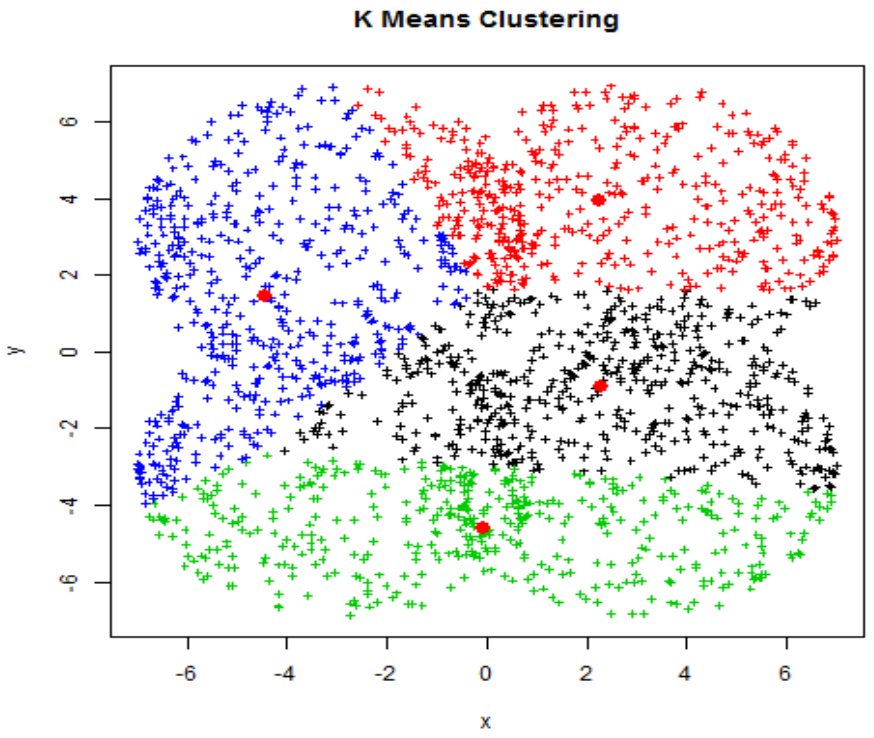
K-Means

5. Her veri için küme merkezleri arasındaki uzaklığı tekrar hesapla.
6. Eğer hiçbir veri herhangi bir kümeye atanmadıysa dur, 3. adımdan başlayarak devam et.

Kümeler arasında eleman değişimi veya merkez noktalarda değişim varsa algoritma sonlandırılır.

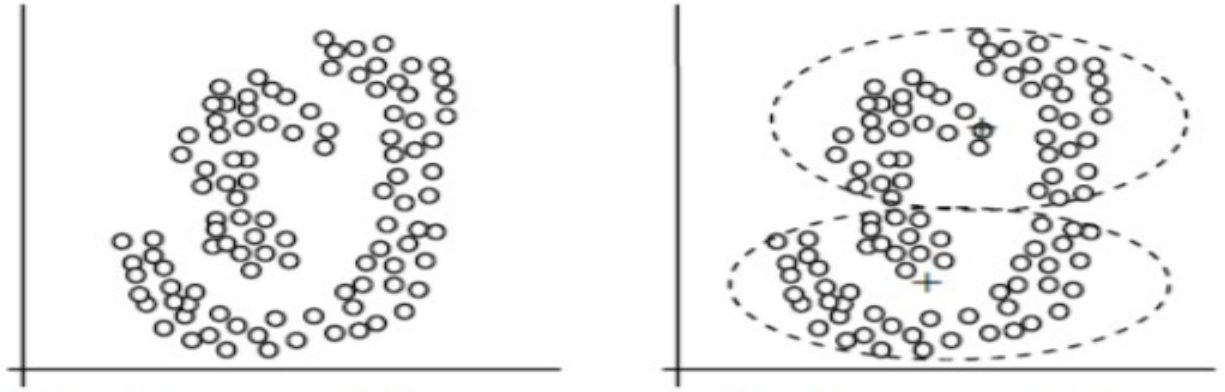
K-Means

- K-Means her çalıştığında verileri atadığı kümeleri merkezlerine göre günceller.



K-Means

- Veri setindeki kümeler bazen kendiliğinden oluşabilir. Bu gibi durumlarda yoğunluk tabanlı ya da hiyerarşik yöntemler kullanılması daha mantıklıdır.

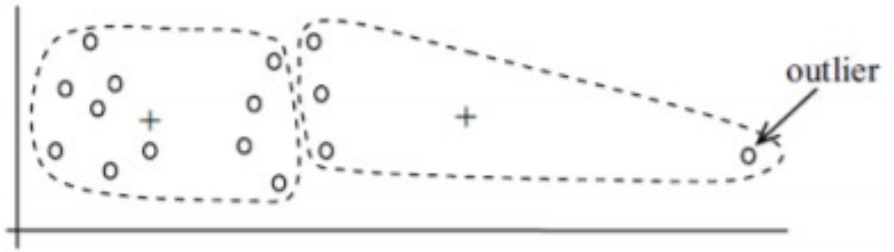


(A): Two natural clusters

(B): *k*-means clusters

K-Means

- K-Means gürültü verilerini de küme içerisine dahil etmektedir. Küme içerisinde aykırı(outlier) veri olması performansı düşürür.



(A): Undesirable clusters

(B): Ideal clusters

K-Means Avantaj/Dezavantaj

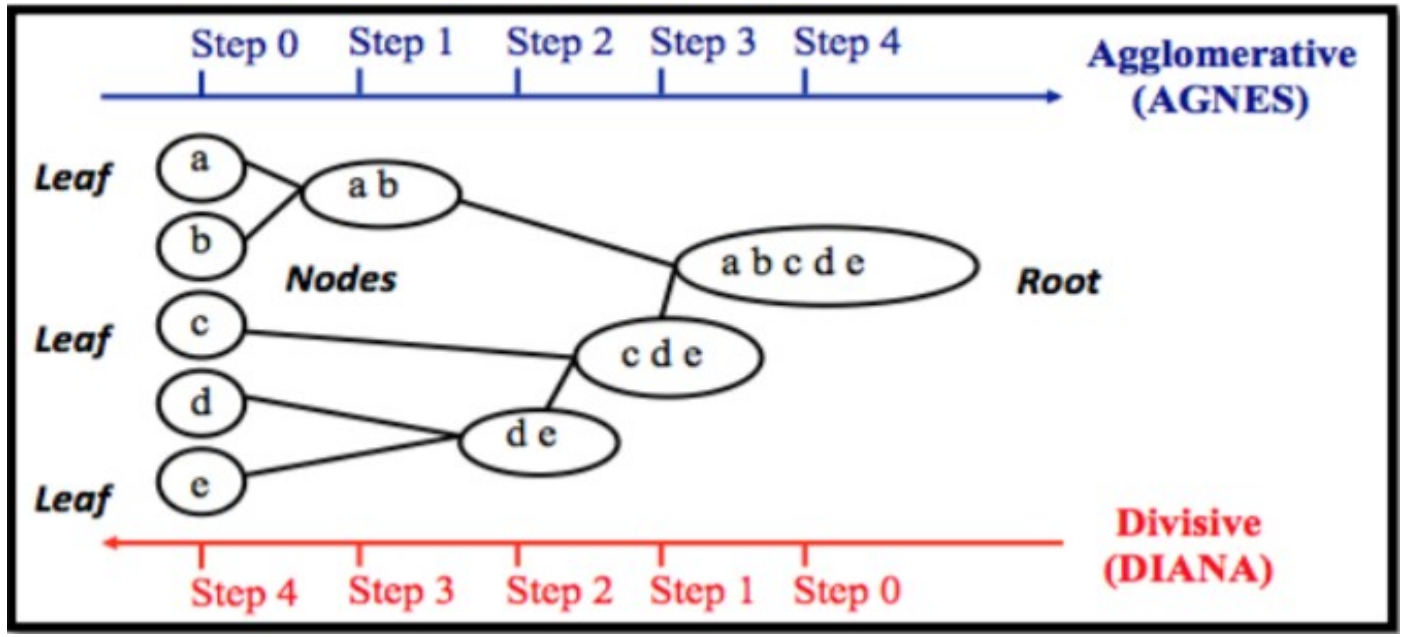
- Algoritmanın işleyişi anlaşılır ve basittir.
- Diğer kümeleme yöntemlerine göre daha az karmaşıktır.
- Verilerin oluşturduğu grupların boyutu ya da yoğunluğu farklı, içerisinde aykırılıklar (gürültüler) olduğunda, şekli dairesel olmadığında, algoritma bu vb. durumlarda başarılı olmayabilir.

Agnes(Agglomerative Nesting)

- Aşağıdan yukarıya (yapraktan köke) doğru kümeleme yapmaktadır.
- Başlangıç aşamasında verilerin her birinin farklı küme olduğu varsayılır.
- Aralarında en az uzaklık olan kümeler ikişer ikişer yeni kümeler oluşturur.
- Köke ulaşıncaya kadar algoritma bu şekilde çalışmaya devam eder.

Agnes

- Kümelenecek veri kalmadığında algoritma sonlandırılır.



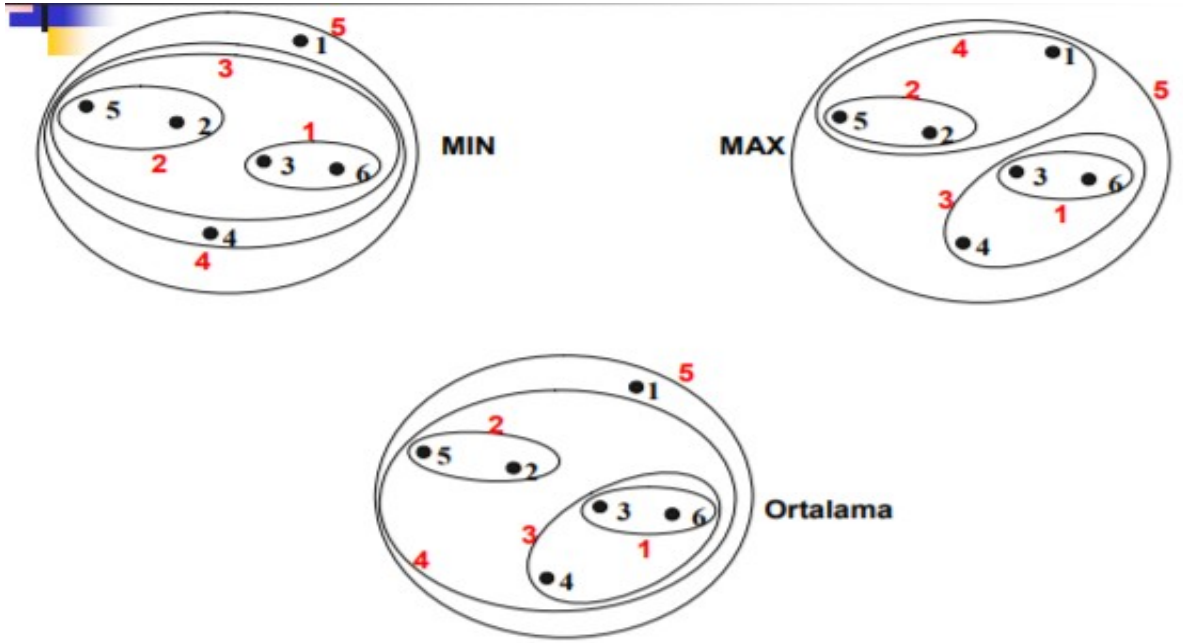
Agnes

- Kümeler arasındaki uzaklık(benzerlik) genelde şu üç yöntemle belirlenir:

1. **Min:** Her iki küme içerisindeki birbirine en yakın olan verilerin uzaklığıdır.
2. **Max:** Her iki küme içerisindeki en uzak iki verinin uzaklığıdır.
3. **Ortalama:** Her iki küme içerisindeki verilerin birbirlerine olan uzaklıkların ortalamasıdır.

Agnes

- Değişen uzaklık yöntemlerine göre kümeleme sonuçları da değişmektedir.



Agnes Avantaj/Dezavantaj

- Kullanıcının küme sayısını belirlemesine gerek yoktur.
- Anlamlı taksonomiler oluşturabilir. (hiyerarşik olarak gruplanan verileri kategorize etmek daha kolaydır.)
- Dezavantajları:
  Gürültü verilerini kümeye ekleyebilir.
  Büyük kümeleri parçalayarak bozabilir.
  Farklı boyuttaki kümeleri oluşturmakta zorlanabilir.

Dbscan

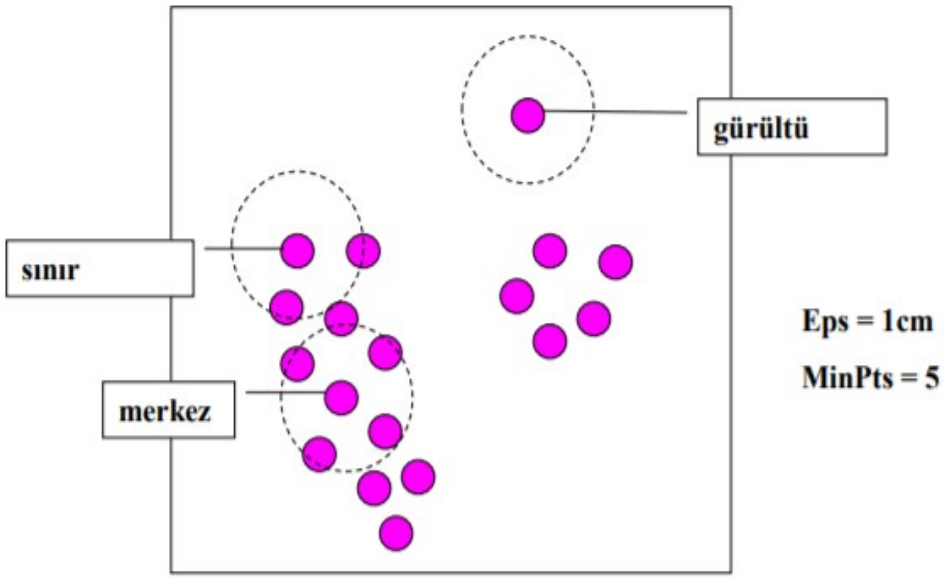
- Kümeler veri setindeki yoğunluğu fazla olan alanlarda oluşturulur.
- Küme yoğunluklarının seyrek olduğu kısımlarda ise gürültü verileri ya da küme sınırını oluşturan veriler bulunur.
- Algoritma verinin sınır noktası ya da gürültü olduğunu tespit etmek için «Eps» ve «MinPts» parametrelerini kullanır.

Dbscan

- Eps: En büyük komşuluk yarıçapıdır.
  MinPts: Eps yarıçaplı komşuluk bölgesinde bulunan minumum veri sayısıdır.
- **Yoğunluk:** Verilen yarıçap (Eps) içerisindeki veri (nokta) sayısıdır.
- **Merkez noktası:** Eps yarıçapında, Minpts'den daha fazla verisi olan noktadır.
- **Sınır noktası:** Eps yarıçapında, Minpts'den daha az verisi olan noktadır.

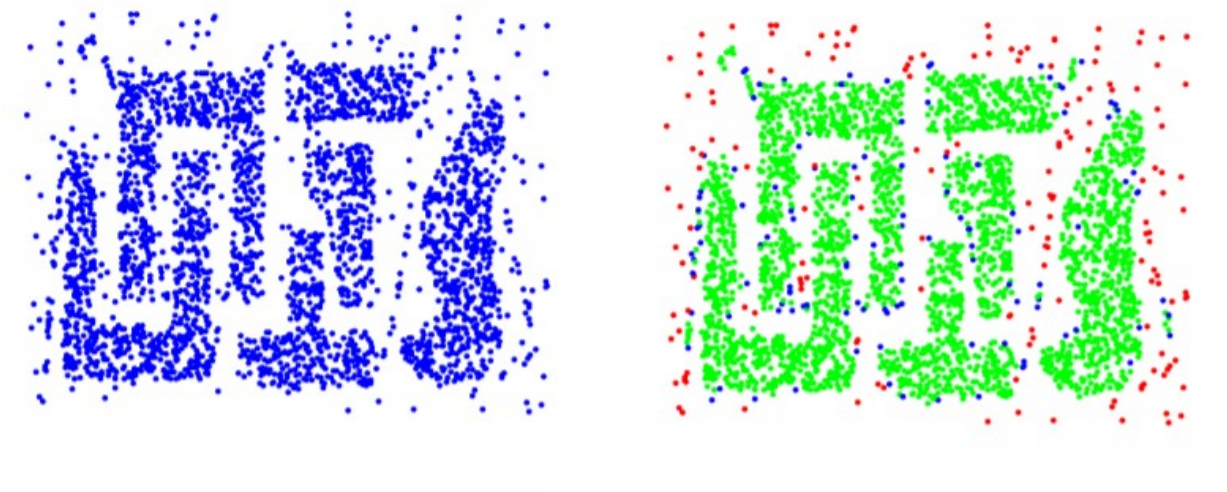
Dbscan

- **Gürültü noktası:** Merkez veya sınır noktası olmayan noktadır.



Dbscan

- Zor ve karmaşık verilerde kolaylıkla kümeleme yapabilir.



Orjinal Noktalar

Nokta tipleri: merkez, sınır ve gürültü

Eps = 10, MinPts = 4

Dbscan Avantaj/Dezavantaj

- Kullanıcının küme sayısını belirlemesine gerek yoktur.
- Farklı şekil ve boyutlarda küme oluşturabilir.
- Büyük veri tabanları olan ve gürültü verisi fazla olan çalışmalarda kullanılabilir.
- Dezavantajı çalışma mantığı detaylıdır ve çalışması performans gerektirir.

Performans Ölçütleri

- **<u>Uzaklık / Benzerlik</u>**
  Aynı küme içerisindeki veriler ne kadar benzerse(yakınsa); farklı küme içerisindeki veriler de ne kadar uzak(az benzerse) kümeleme işlemi o kadar başarılır.
- Benzerlik için: s(i,j)
  Uzaklık için: d(i,j) = 1 – s(i,j) hesaplanır.

- **Entropy**
- Her küme için entropi hesaplanır. Kümedeki farklı etiketlerin olasılıkları alınır.

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j) \log_2 \Pr_i(c_j)$$

- Burada, Di i. küme, Pri (cj) j. sınıf etiketinin olasılığıdır.
- Tüm kümeler için entropi hesaplanır.

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i)$$

- |Di| i. kümedeki eleman sayısıdır. |D| toplam eleman sayısıdır

- **Purity(Saflık)**
- Her küme için purity hesaplanır.

$$purity(D_i) = \max_j(\Pr_i(c_j))$$

- Burada, Di i. küme, Pri (cj) j. küme etiketinin olasılığıdır.
- Tüm kümeler için purity hesaplanır.

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i)$$

- |Di| i. kümedeki eleman sayısıdır. |D| toplam eleman sayısıdır.

- **Ground truth**
- Verilerin küme sayısı belirli olduğu durumlarda elde edilen sonuç bu küme sayısına göre değerlendirilir.
- Her küme içerisindeki doğru atanmış elemanlara (referans verilere) göre algoritma sonucu değerlendirilir.

Sonuç ve Yorum

- K-Means algoritmasının önceliği uzaklıklar olduğu için her veri tipinde doğru sonuç veremeyebilir; fakat bu zamanda en çok bilinen ve de kullanılan algoritma K-Means olmuştur. (hem kolaylığı hem de anlaşılabilirliğinden ötürü.)
- Günümüzde ise veri yoğun olarak artmaktadır; artan verilerin hem yoğunluğu açısından hem de doğru sonucu vermesi için **«Dbscan»** algoritmasının geliştirilmesi ilgili çalışmalarda üzerinde daha çok gidilmesi gerekmektedir.

Kaynakça

- https://dergipark.org.tr/tr/download/article-file/467823
- https://dergipark.org.tr/tr/download/article-file/193944
- https://dergipark.org.tr/tr/download/article-file/751785
- https://dergipark.org.tr/tr/download/article-file/282717
- https://dergipark.org.tr/en/download/article-file/384439

- Teşekkürler.