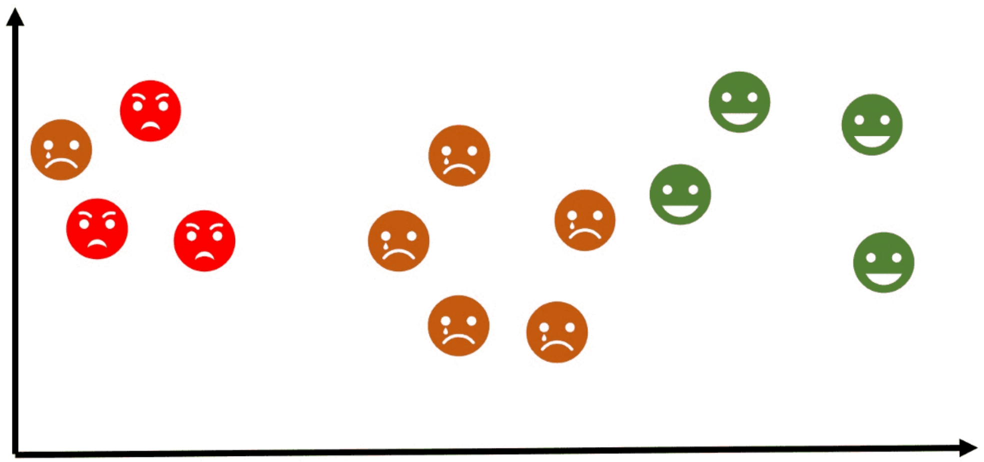
EKO 469-Veri Madenciliği

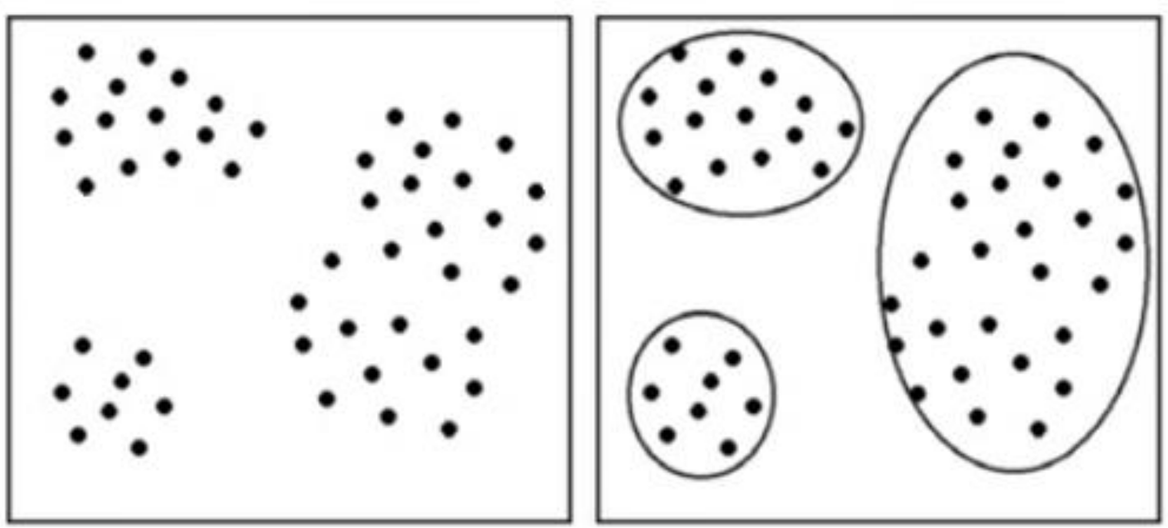
Hafta 13 – Kümeleme

Arş. Gör. Dr. Muhammed Kotan

Kümeleme (Clustering)



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

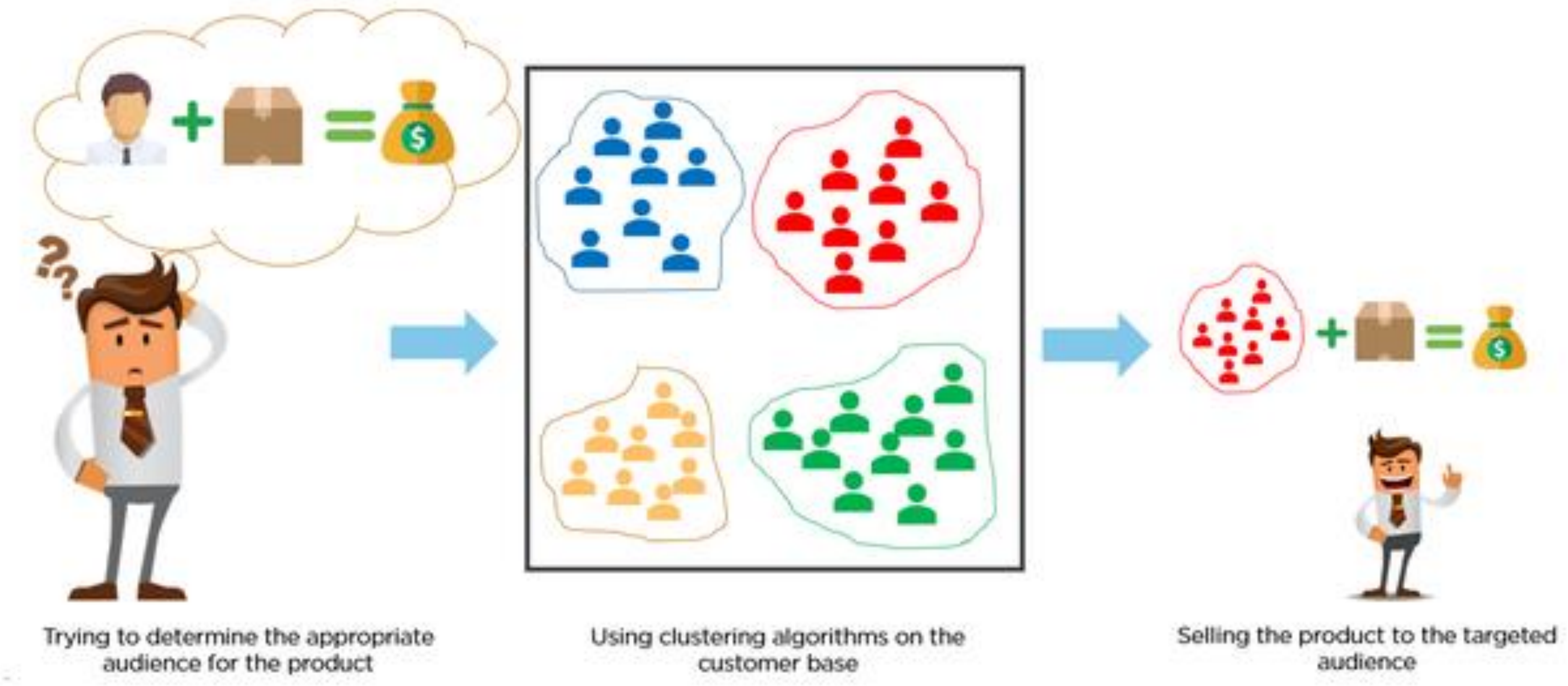


Kümeleme (Clustering)

- Kümeleme analizi; Desen tanımlama, veri analizi, resim işleme, pazar araştırması vb. pratikte birçok aktivitede kullanılır.

- 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 paternlerin ortaya çıkarılmasında da kullanılabilir.

Kümeleme



Kaynak: https://medium.com/@ekrem.hatipoglu/machine-learning-clustering-k%C3%BCmeleme-k-means-algorithm-part-13-be33aeef4fc8

Kümeleme (Clustering)

- Kümelemenin sınıflandırmadan farkı sınıflandırmadaki gibi önceden tanımlı sınıf etiketlerinin olmamasıdır. Bu sebeple kümelemede, sınıflandırmadaki gibi örnekleyerek öğrenme yerine gözlemleyerek öğrenme kavramı geçerlidir.

- Kümeleme yöntemlerinin bir çoğu gözlem değerleri arasındaki uzaklıkların hesaplanması esasına dayanır. Bu nedenle iki nokta arasındaki uzaklığı hesaplayan bağıntılara gereksinim vardır.

Veri Matrisi ve Farklılık Matrisi

- **Veri matrisi:**
  - ➢ n nesne X p öznitelik
  - ➢ Her satır bir nesneye karşılık gelir

- **Farklılık matrisi:**
  - ➢ n nesne X n nesne
  - ➢ d(i,j): i. ve j. nesnelerin farklılığı
  - ➢ d(i,j)=0 bir nesnenin kendisi ile farklılığı
  - ➢ d(i,j)=d(j,i)
  - ➢ d(i,j) >= 0
  - ➢ sim(i,j)=1-d(i,j)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Nümerik Öznitelikler İçin Uzaklık Ölçümü

➢Minkowski mesafesi

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h},$$

➢Manhattan mesafesi

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|.$$

➢Öklid mesafesi

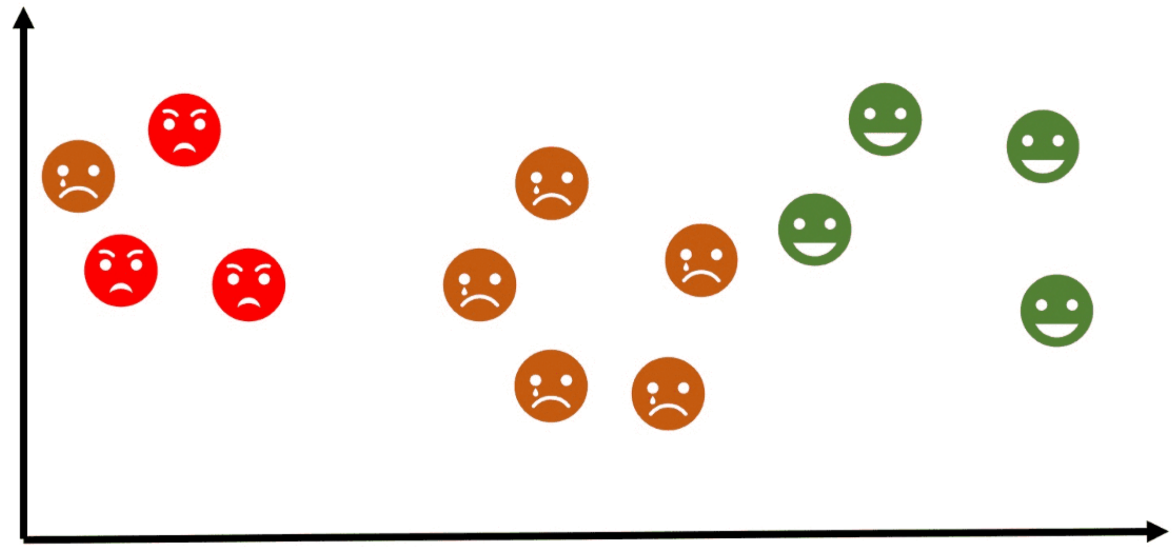
$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}.$$

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

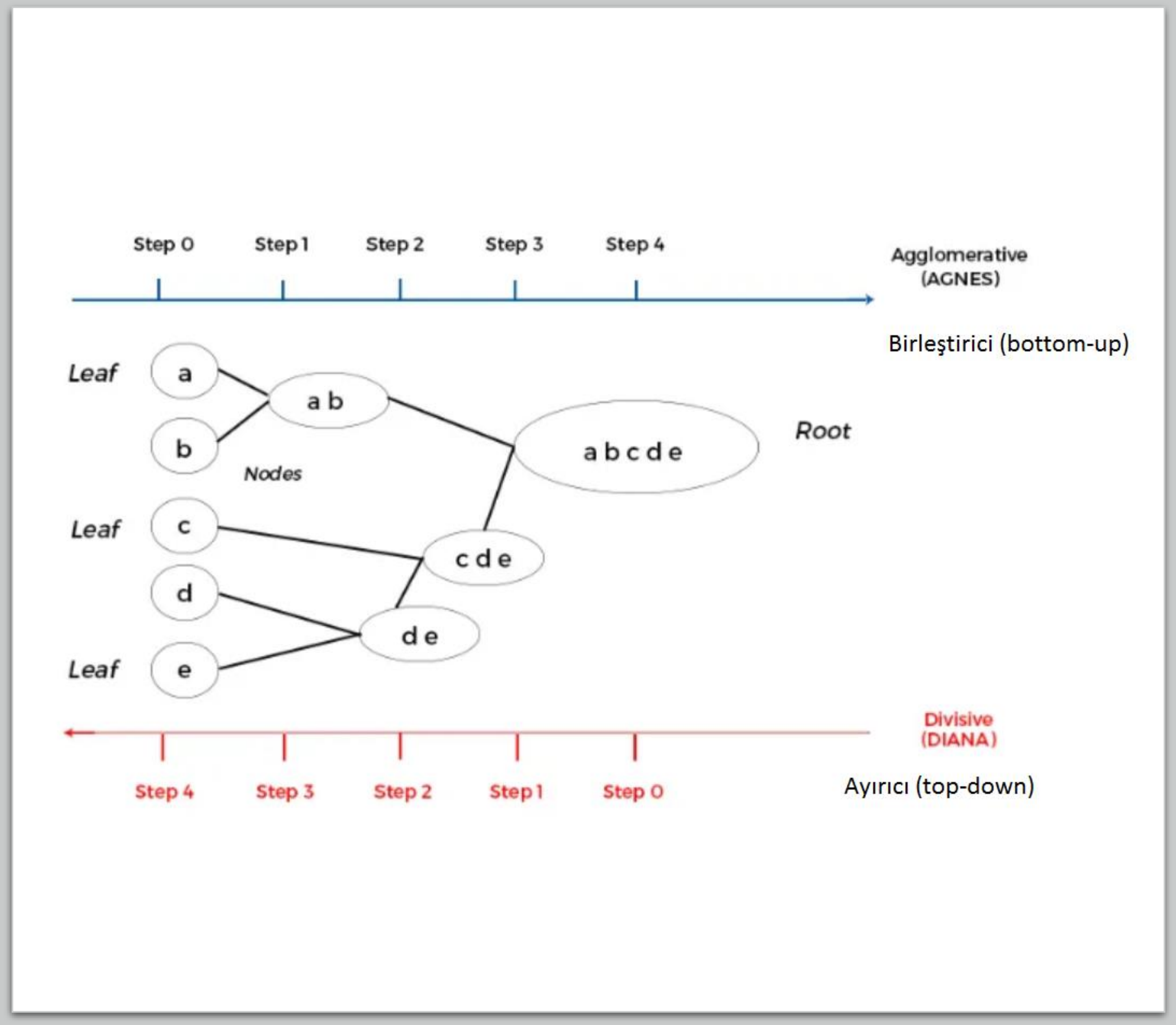
Kümeleme Yöntemleri

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



1. Hiyerarşik Kümeleme

- Hiyerarşik kümeleme yöntemleri, veri setinde kaç grubun bulunduğunun başlangıçta bilinmediği durumlarda kullanılabilir.
- **“Birleştirici/Toplamalı Hiyerarşik Kümeleme Yöntemleri (Agglomerative)”** ve **“Ayırıcı/Bölünmeli Hiyerarşik Kümeleme Yöntemleri (Divisive)”** olmak üzere ikiye ayrılır.
- Hiyerarşik yöntemlerin ağaç diyagramları ile gösterilen sonuçlarına dendogram denir.

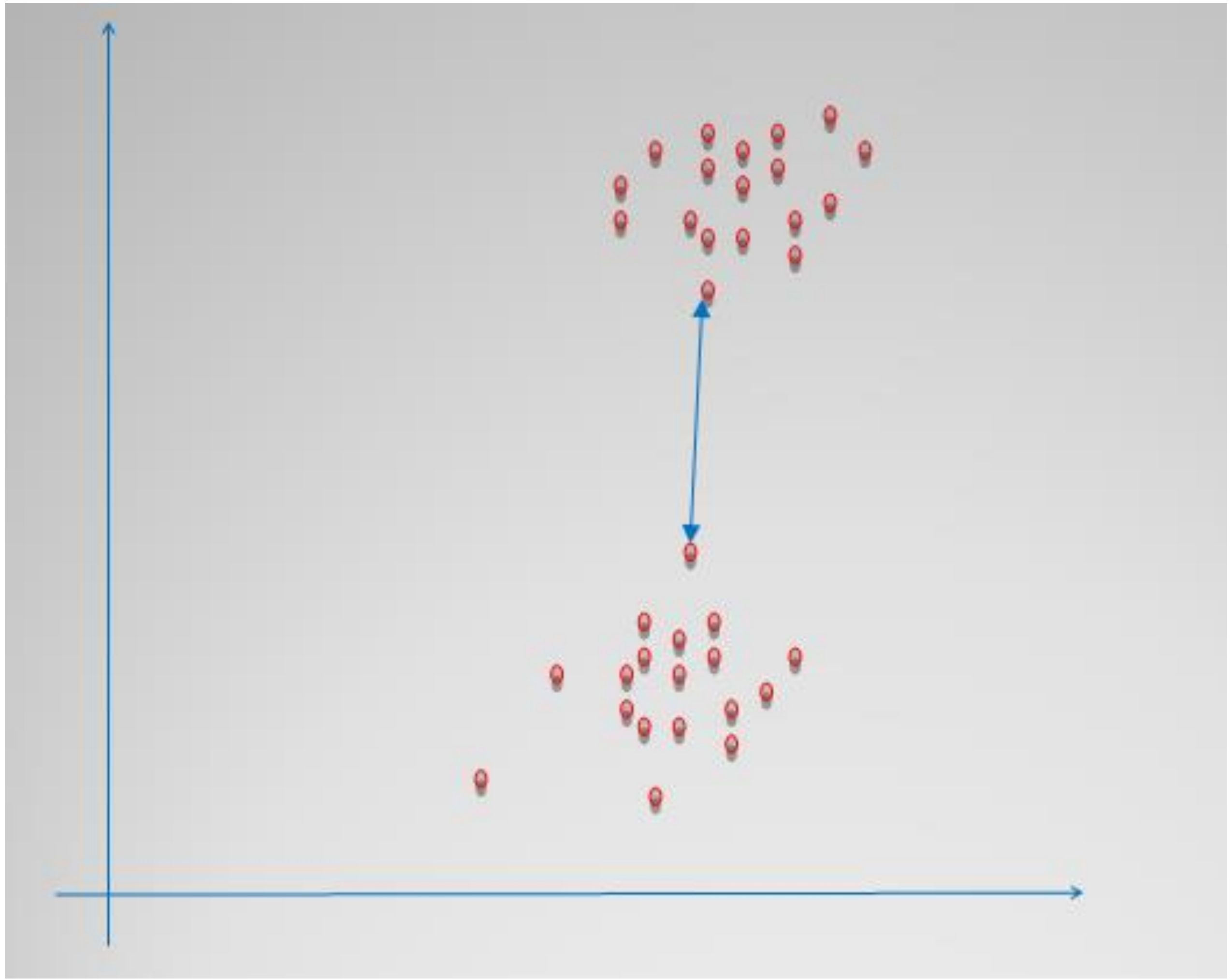


1.1 Birleştirici/Toplamalı Hiyerarşik Yöntemler

- Bu yöntemde başlangıç aşamasında tüm gözlem değerleri birer küme olarak ele alınır ve adım adım bu kümeler birleştirilerek yeni kümeler elde edilir. Tüm gözlemler tek bir kümede toplanınca işlem sonlandırılır.
- **En yakın komşu algoritması (tek bağlantılı kümeleme yöntemi)** ve **en uzak komşu algoritması (tam bağlantılı kümeleme yöntemi)** birleştirici hiyerarşik kümeleme yöntemine örnek olarak verilebilir.
- Gözlem değerleri arasındaki uzaklıkların hesaplanabilmesi için Öklid uzaklık formülünden yararlanılabilir.

1.1.1 En yakın komşu algoritması

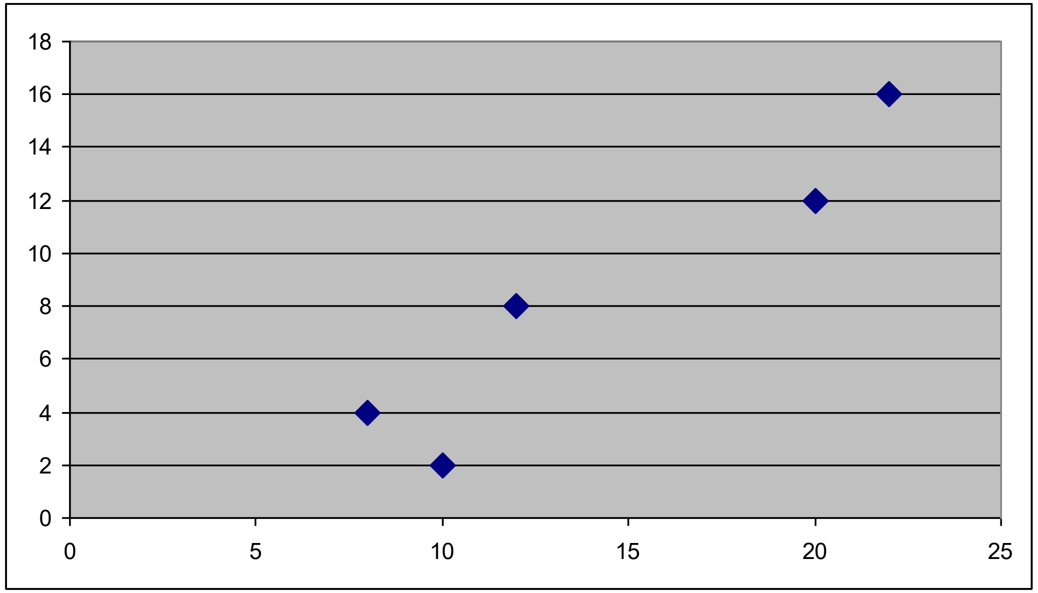
- En yakın komşu algoritmasında gözlemler arasında birbirine en yakın olanların uzaklığı iki kümenin birbirine olan uzaklığı olarak değerlendirilir.
- En düşük uzaklık seçilerek bu uzaklıkla ilgili elemanlar birleştirilip yeni bir küme elde edilir. Daha sonra uzaklıklar yeniden hesaplanır.



1.1.1
En yakın komşu algoritması

- 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



1.1.1 En yakın komşu algoritması

- Uzaklık tablosunu oluşturalım:

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

$$uzak(1,2) = \sqrt{(8-12)^2 + (4-8)^2} = 5.66$$
$$uzak(1,3) = \sqrt{(8-10)^2 + (4-2)^2} = 2.83$$
$$uzak(1,4) = \sqrt{(8-20)^2 + (4-12)^2} = 14.42$$
$$uzak(1,5) = \sqrt{(8-22)^2 + (4-16)^2} = 18.44$$
$$uzak(2,3) = \sqrt{(12-10)^2 + (8-2)^2} = 6.32$$
$$uzak(2,4) = \sqrt{(12-20)^2 + (8-12)^2} = 8.94$$
$$uzak(2,5) = \sqrt{(12-22)^2 + (8-16)^2} = 12.81$$
$$uzak(3,4) = \sqrt{(10-20)^2 + (2-12)^2} = 14.14$$
$$uzak(3,5) = \sqrt{(10-22)^2 + (2-16)^2} = 18.44$$
$$uzak(4,5) = \sqrt{(20-22)^2 + (12-16)^2} = 4.47$$

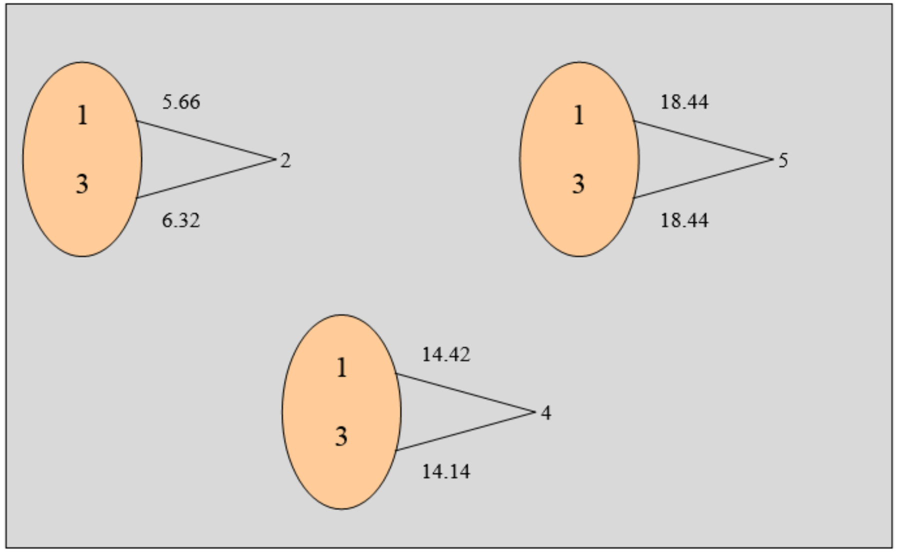
1.1.1
En yakın komşu algoritması

- 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	

1.1.1
En yakın komşu algoritması

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

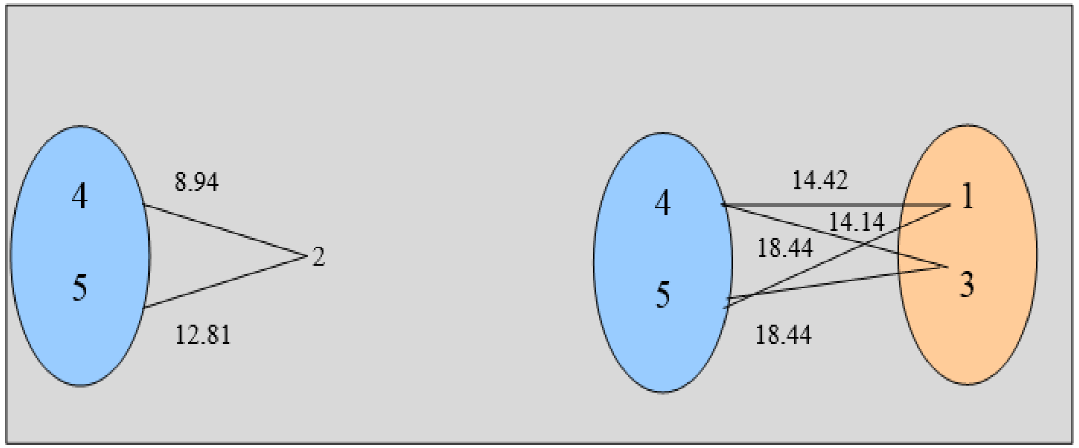


Hasta no	{1,3}	2	4	5
{1,3}				
2	5.66			
4	14.14	8.94		
5	18.44	12.81	4.47	

- En düşük uzaklık 4.47 olup bu değere sahip 4 ve 5 nolu gözlemler birleştirilerek {4,5} kümesini elde ederiz.

1.1.1 En yakın komşu algoritması

- {4,5} ile 2 gözlemi arasındaki mesafe 8.94, {1,3} ile arasındaki mesafe 14.14 tür.

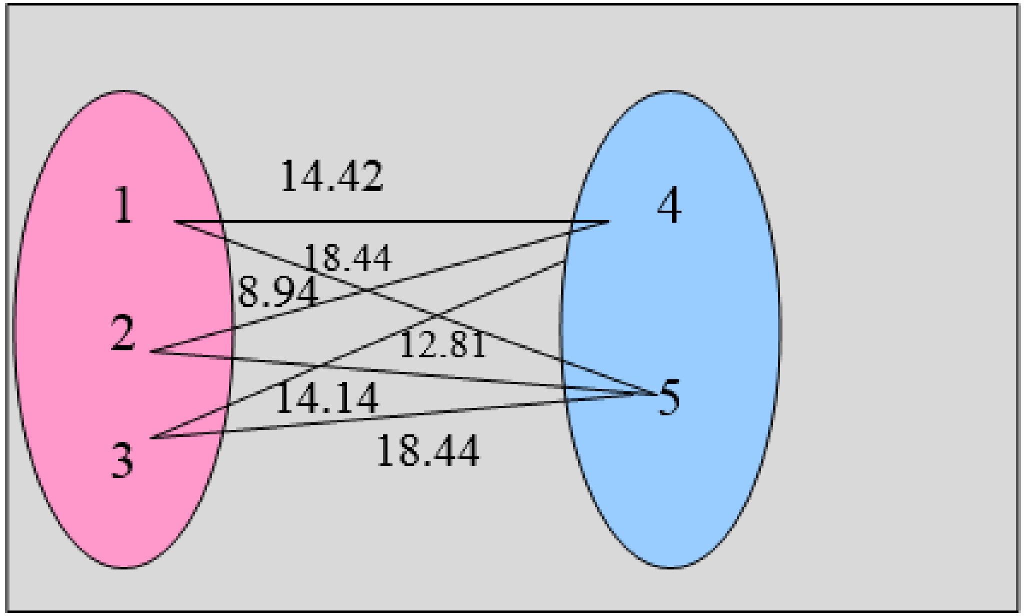


- Yeni uzaklık tablosundaki en düşük uzaklık 5.66 olup bu değere sahip {1,3} ve 2 nolu gözlemler birleştirilerek {1,2,3} kümesini elde ederiz.

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

1.1.1 En yakın komşu algoritması

- {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**	

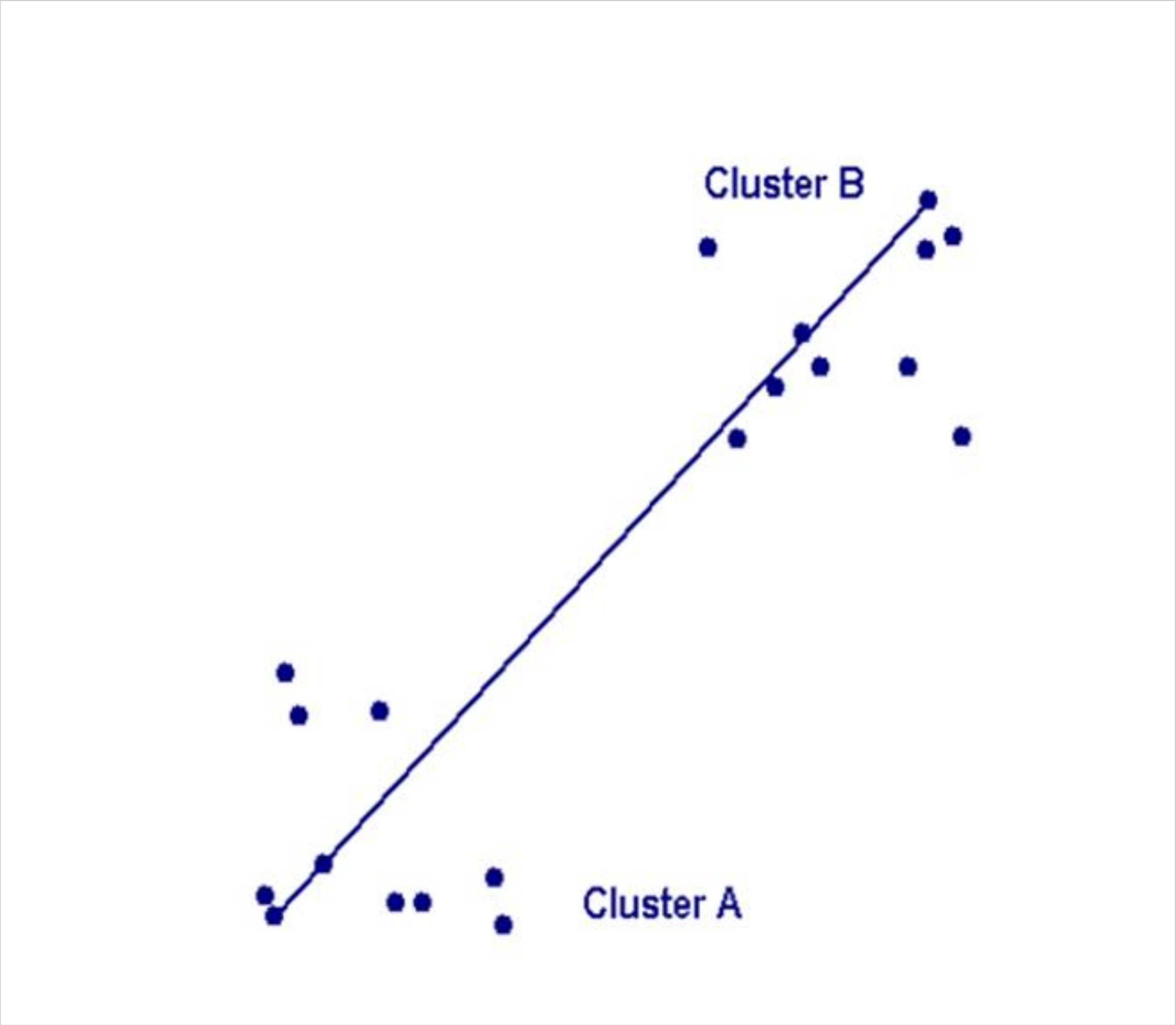
1.1.1 En yakın komşu algoritması

- 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}

1.1.2 En uzak komşu algoritması örnek

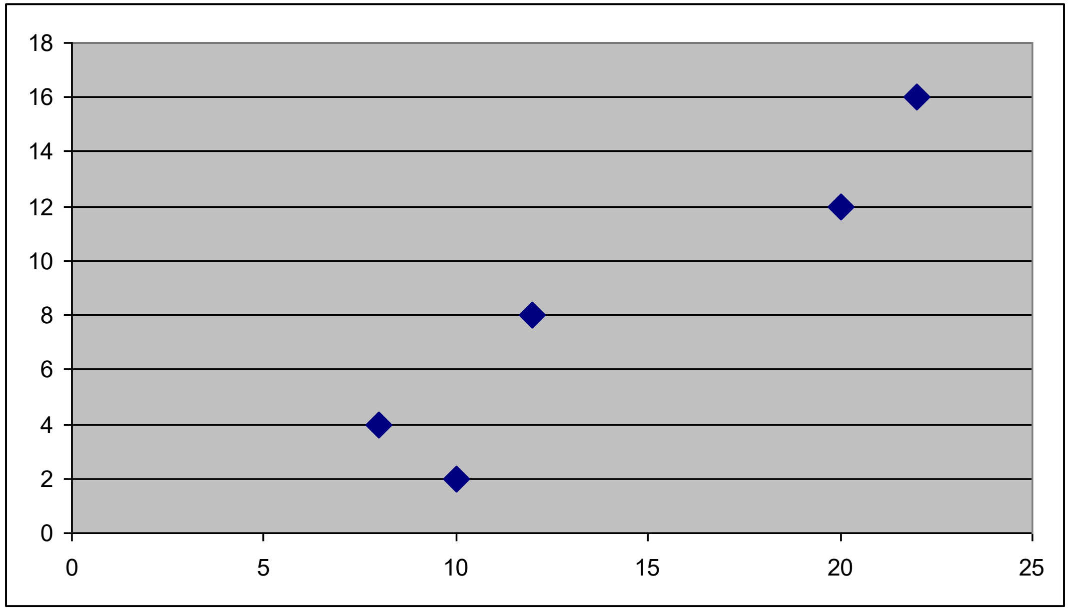
- Bu yaklaşımda iki küme arasındaki en uzak mesafe uzaklık tablosuna iki küme arasındaki mesafe olarak işlenir.
- Birleştirme işlemi yapılırken, en yakın komşu algoritmasında olduğu gibi tablodaki en kısa mesafeye sahip olan gözlemler seçilir.



1.1.2 En uzak komşu algoritması örnek

- Önceki örnekteki verileri kullanarak Tam Bağlantılı Hiyerarşik Kümeleme Yöntemi (en uzak komşu algoritması) ile 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



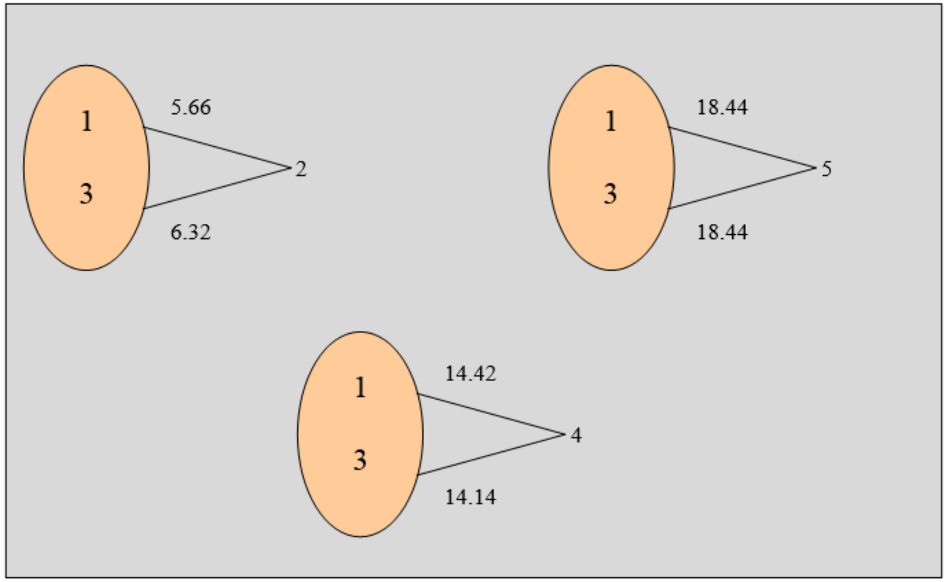
1.1.2
En uzak komşu algoritması örnek

- İlk olarak Öklid bağlantısı kullanılarak uzaklıklar hesaplanır.
- İlk aşamada tek bağlantılı hiyerarşik kümelemede olduğu gibi en küçük mesafeye sahip olan gözlemler birleştirilir.

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	

1.1.2
En uzak komşu algoritması örnek

Daha sonra {1,3} kümesi ile diğer gözlemler arasındaki uzaklıklar hesaplanır. Buradaki en uzak mesafeler seçilerek uzaklık tablosuna kaydedilir.

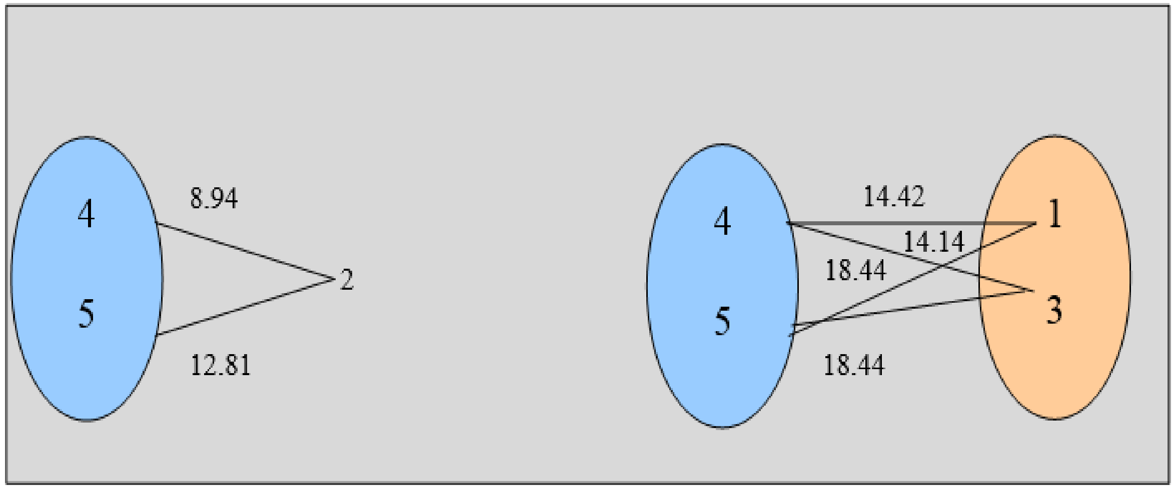


Hasta no	{1,3}	2	4	5
{1,3}				
2	6.32			
4	14.42	8.94		
5	18.44	12.81	4.47	

- En düşük uzaklık 4 ve 5 gözlemlerine ait olduğu için bu gözlemler birleştirilir.

1.1.2 En uzak komşu algoritması

- {4,5} ve diğer gözlemler için yeni mesafeler hesaplanır. En uzak mesafeler uzaklık tablosuna işlenir.

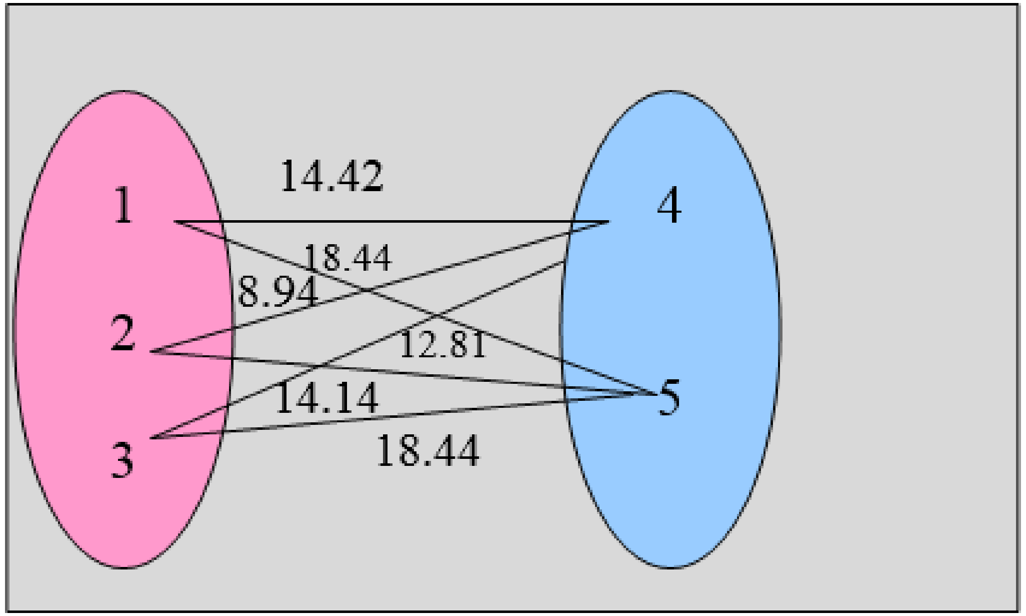


Hasta no	{1,3}	2	{4,5}
{1,3}			
2	5.66		
{4,5}	18.44	12.81	

- Yeni uzaklık tablosunda en küçük mesafe 2 ve {1,3} gözlemlerine ait olduğu için bu gözlemler birleştirilir.

1.1.2
En uzak komşu algoritması

- Yeni uzaklık tablosunda en küçük mesafe 2 ve {1,3} gözlemlerine ait olduğu için bu gözlemler birleştirilerek yeni mesafeler hesaplanır.

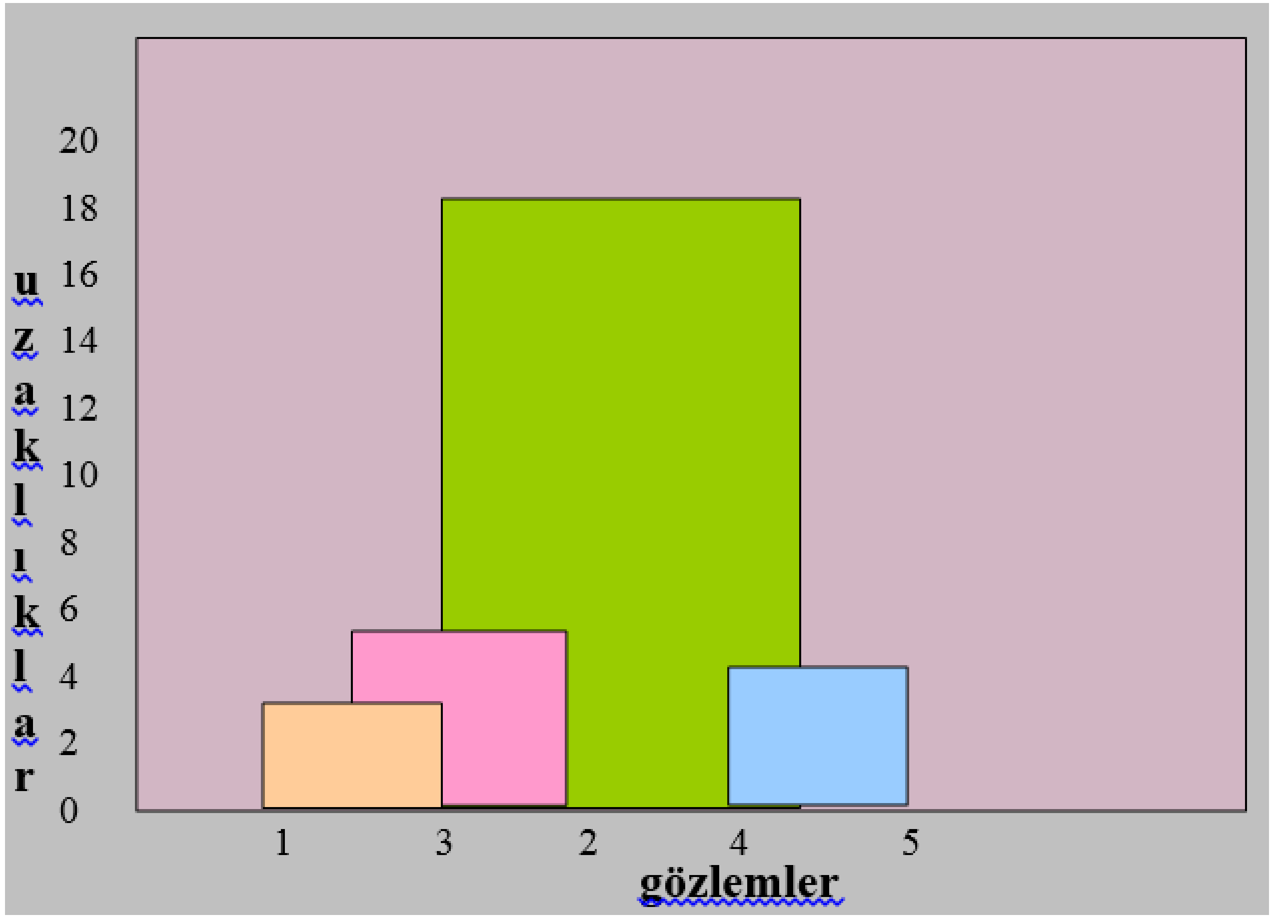


Hasta no	{1,2,3}	{4,5}
{1,2,3}		
{4,5}	18.44	

1.1.2
En uzak komşu algoritması

Sonuç kümenin tanımlanması ve dendrogram

Uzaklıklar	Kümeler
2.83	{1,3}
4.47	{4,5}
5.66	{1,2,3}
18.44	{1,2,3,4,5}



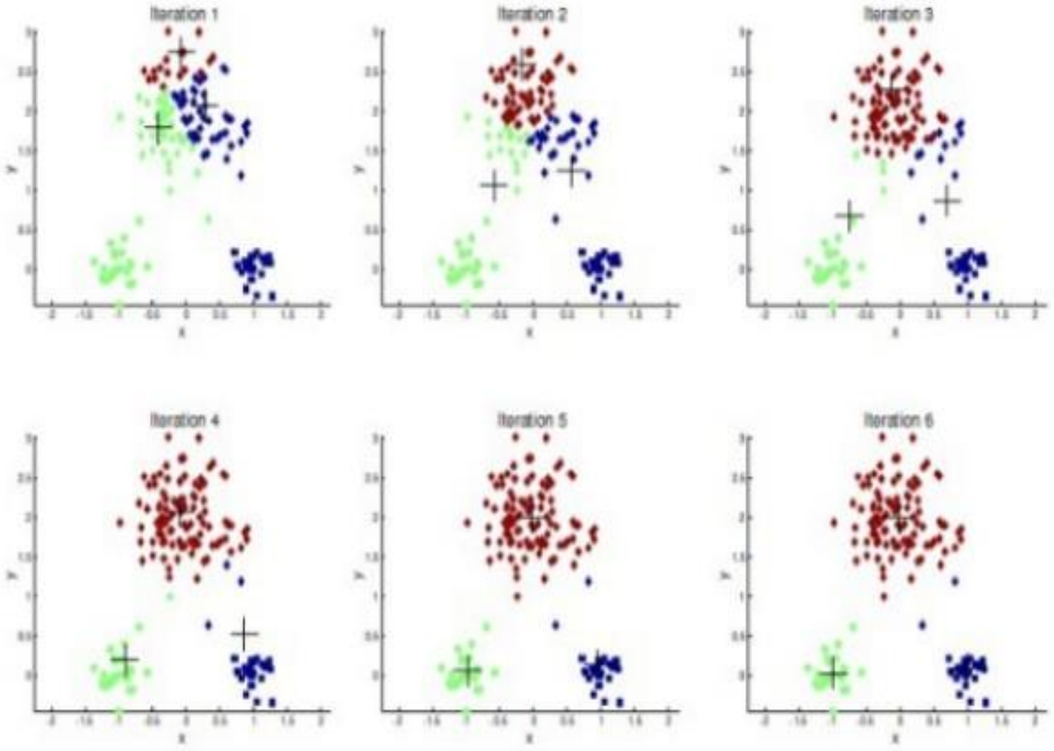
1.2 Ayırıcı/Bölünmeli Hiyerarşik Yöntemler

- Başlangıçta bütün gözlemler tek bir küme olarak değerlendirilir.
- Her aşamada benzer olmayan gözlemler belirlenerek daha küçük kümeler oluşturulur.
- Her gözlem ayrı bir küme olarak gösterilinceye kadar bu süreç devam eder.

2. Bölümlemeli Yöntemler:

2.1 K-means

- Kümeleme algoritmalarının en basitidir.
- Veriyi en iyi ifade edecek K adet vektör bulmaya çalışır.
- K sayısı kullanıcı tarafından verilir.
- Uygun k sayısının belirlenmesi sorun olabilir.
- k sayısını kendi hesaplayan x-means algoritması da kullanılabilir.
- Sadece sayısal veriler ile çalışır.
- Bu metot çok geniş veritabanları üzerinde de uygulanabilir. Çünkü karmaşıklığı oldukça azdır.



2.1 K-Means

- K-Means algoritması, veritabanındaki n tane nesnenin k adet kümeye bölümlenmesini sağlar. Kümeleme sonucu küme içi (intra-cluster) elamanlar arasındaki benzerlikler çok iken, kümeler arası (inter-cluster) elamanların benzerlikleri çok düşüktür.
- K-means kümeleme yönteminin değerlendirilmesinde en yaygın olarak karesel hata kriteri SSE (sum of squared errors) kullanılır. En düşük SSE değerine sahip kümeleme sonucu en iyi sonucu verir.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

K-means Algoritmasının adımları

1. *Kaç tane küme oluşturulacağı **k** sayısı ile belirlenir. K adet nokta küme merkezi olarak atanır.*
2. *Her gözlem en yakın merkez noktanın olduğu kümeye dâhil edilir.*
3. *Her nesnenin ortalaması hesaplanır. Merkez nokta kümedeki nesnelerin ortalaması olacak şekilde güncellenir.*
4. *Nesnelerin kümelemesinde değişiklik olmayana kadar adım 2'ye geri dönülür.*

2.1 K-Means - Örnek

Aşağıdaki 8 nokta için 3 küme elde ediniz

A1(2, 10)

A2(2, 5)

A3(8, 4)

A4(5, 8)

A5(7, 5)

A6(6, 4)

A7(1, 2)

A8(4, 9)

İlk küme merkezleri A1(2, 10), A4(5, 8) ve A7(1, 2) olsun.

İki nokta arasındaki uzaklık değerlerini aşağıdaki formülle hesaplayalım:

$a=(x1, y1)$ ve $b=(x2, y2)$;

$\rho(a, b) = |x2 - x1| + |y2 - y1|$.

2.1 K-Means - Örnek

İlk küme merkezleri A1(2, 10), A4(5, 8) ve A7(1, 2) olsun.

İki nokta arasındaki uzaklık değerlerini aşağıdaki formülle hesaplayalım:

$a=(x1, y1)$ ve $b=(x2, y2)$;

$\rho(a, b) = |x2 - x1| + |y2 - y1|$.

1.İterasyon

		(2, 10)	(5, 8)	(1, 2)	
	Nokta	**1.küme**	**2.küme**	**3.küme**	**Küme**
A1	(2, 10)				
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	**(4, 9)**				

2.1 K-Means - Örnek

- Her gözlem kendine en yakın merkez noktanın olduğu kümeye dahil edilir.

nokta	merkez1
$x1, y1$	$x2, y2$
(2, 10)	(2, 10)

$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$$\begin{aligned}\rho(\text{nokta, merkez1}) &= |x2 - x1| + |y2 - y1| \\ &= |2 - 2| + |10 - 10| \\ &= 0 + 0 \\ &= 0\end{aligned}$$

$x1, y1$	$x2, y2$
(2, 10)	(5, 8)

$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$$\begin{aligned}\rho(\text{nokta, merkez2}) &= |x2 - x1| + |y2 - y1| \\ &= |5 - 2| + |8 - 10| \\ &= 3 + 2 \\ &= 5\end{aligned}$$

$x1, y1$	$x2, y2$
(2, 10)	(1, 2)

$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$$\begin{aligned}\rho(\text{nokta, merkez3}) &= |x2 - x1| + |y2 - y1| \\ &= |1 - 2| + |2 - 10| \\ &= 1 + 8 \\ &= 9\end{aligned}$$

2.1 K-Means - Örnek

- Her gözlem kendine en yakın merkez noktanın olduğu kümeye dahil edilir.
- Elde edilen verileri tabloya yerleştirelim.

1.İterasyon

		(2, 10)	(5, 8)	(1, 2)	
	Nokta	**1.küme**	**2.küme**	**3.küme**	Küme
A1	(2, 10)	0	5	9	1
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	**(4, 9)**				

1.küme	2.küme	3.küme
(2, 10)		

2.1 K-Means - Örnek

- Her gözlem kendine en yakın merkez noktanın olduğu kümeye dahil edilir.
- Elde edilen verileri tabloya yerleştirelim.

1.İterasyon

		(2, 10)	(5, 8)	(1, 2)	
	Nokta	**1.küme**	**2.küme**	**3.küme**	Küme
A1	(2, 10)	0	5	9	**1**
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	**(4, 9)**				

1.küme	2.küme	3.küme
(2, 10)		

2.1 K-Means - Örnek

x1, y1 *x2, y2*
(2, 5) (2, 10)

$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$\rho(nokta, merkez1) = |x2 - x1| + |y2 - y1|$
$= |2 - 2| + |10 - 5|$
$= 0 + 5$
$= 5$

x1, y1 *x2, y2*
(2, 5) (5, 8)

$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$\rho(nokta, merkez2) = |x2 - x1| + |y2 - y1|$
$= |5 - 2| + |8 - 5|$
$= 3 + 3$
$= 6$

x1, y1 *x2, y2*
(2, 5) (1, 2)
$\rho(a, b) = |x2 - x1| + |y2 - y1|$

$\rho(nokta, merkez3) = |x2 - x1| + |y2 - y1|$
$= |1 - 2| + |2 - 5|$
$= 1 + 3$
$= 4$

1.İterasyon

		(2, 10)	(5, 8)	(1, 2)	
	Nokta	1.küme	2.küme	3.küme	Küme
A1	(2, 10)	0	5	9	1
A2	(2, 5)	5	6	4	3
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(4, 9)				

1.küme	2.küme	3.küme
(2, 10)		**(2, 5)**

2.1 K-Means - Örnek

Bu şekilde devam ederek sonuç tabloyu tamamlayalım.

1.İterasyon

		(2, 10)	(5, 8)	(1, 2)	
	Nokta	**1.küme**	**2.küme**	**3.küme**	Küme
A1	(2, 10)	0	5	9	**1**
A2	(2, 5)	5	6	4	**3**
A3	(8, 4)	12	7	9	**2**
A4	(5, 8)	5	0	10	**2**
A5	(7, 5)	10	5	9	**2**
A6	(6, 4)	10	5	7	**2**
A7	(1, 2)	9	10	0	**3**
A8	**(4, 9)**	**3**	**2**	**10**	**2**

1.küme	2.küme	3.küme
(2, 10)	(8, 4)	(2, 5)
	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	
	(4, 9)	

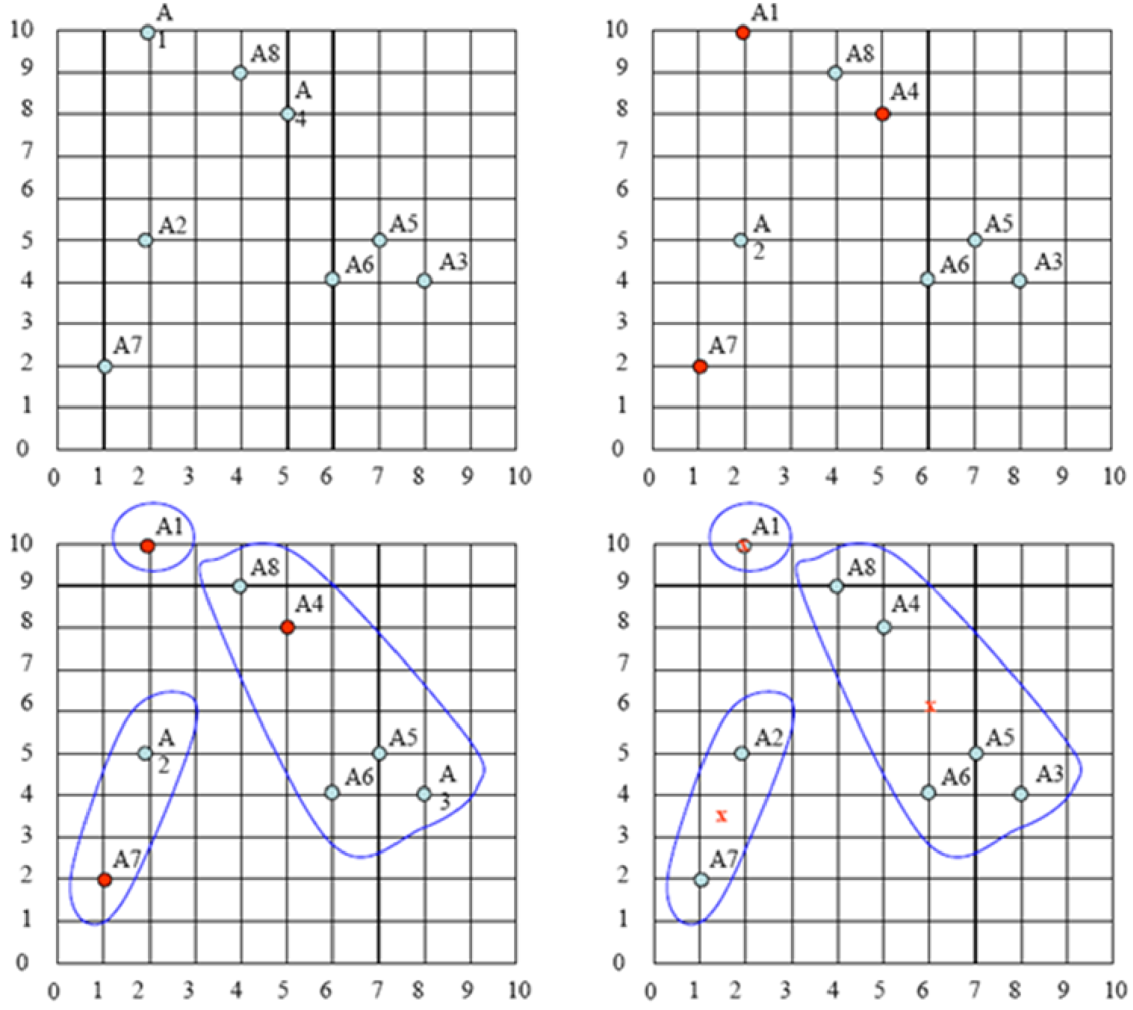
2.1 K-Means - Örnek

Yeni küme merkezlerini hesaplayalım:

1.küme için A1(2, 10)
2.küme için ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6)
3.küme için ((2+1)/2, (5+2)/2) = (1.5, 3.5)

Yeni kümeler:
1:{A1}
2:{A3,A4,A5,A6,A8}
3:{A2,A7}

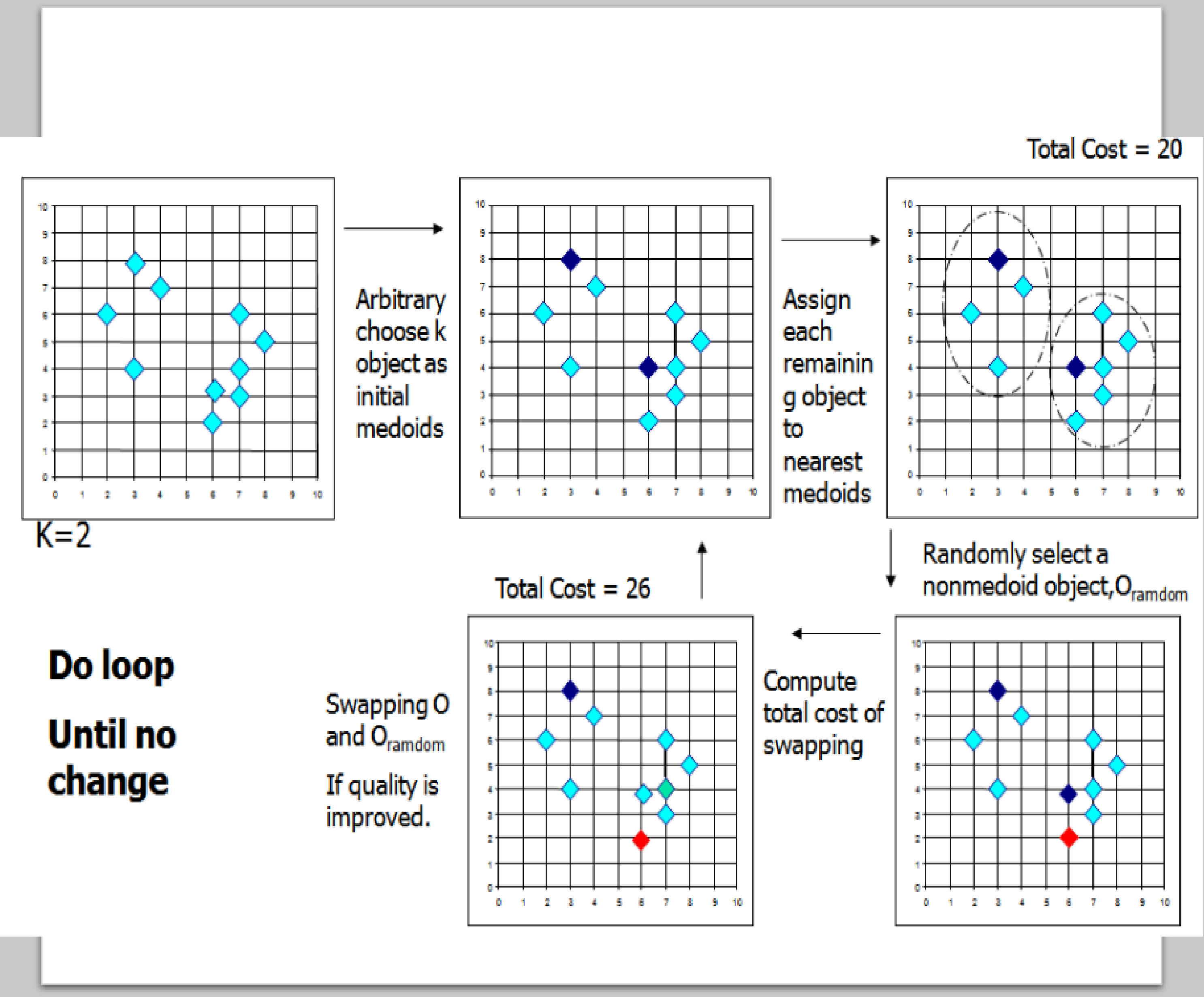
Olarak elde edilmiştir.



2. Bölümlemeli Yöntemler:

2.2 K-medoids

- Verinin yapısal özelliklerini temsil eden k tane temsilci nesneyi bulma esasına dayanır.
- Temsilci nesne medoid olarak adlandırılır ve kümenin merkezine en yakın noktadır.
- Her bir küme için kabaca bir temsilci nesne belirlenir. Kalan her nesne bu medoid ile karşılaştırılır ve benzerliğine göre o nesne kümeye dahil edilir. Bir kümedeki nesneyi alarak, daha yüksek kaliteyi elde edene dek kümeler arasında iteratif olarak yer değiştirme yapılır.
- Merkezi elemanların kümeyi temsil etmesinden dolayı gürültülü veriye karşı duyarlı değildir.



2. Bölümlemeli Yöntemler:
2.3 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 luk bir 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 n^2 oranında daha maliyetli bir yoldur.

Referanslar

- ***Veri madenciliği ders notları, Dr. Burcu Çarklı YAVUZ, SAÜ (Temel İçerik)***
- Veri madenciliği ders notları, Doç. Dr. Nilüfer YURTAY, SAÜ
- www.medium.com
- https://www.congrelate.com/41-machine-learning-clusters-gif/