



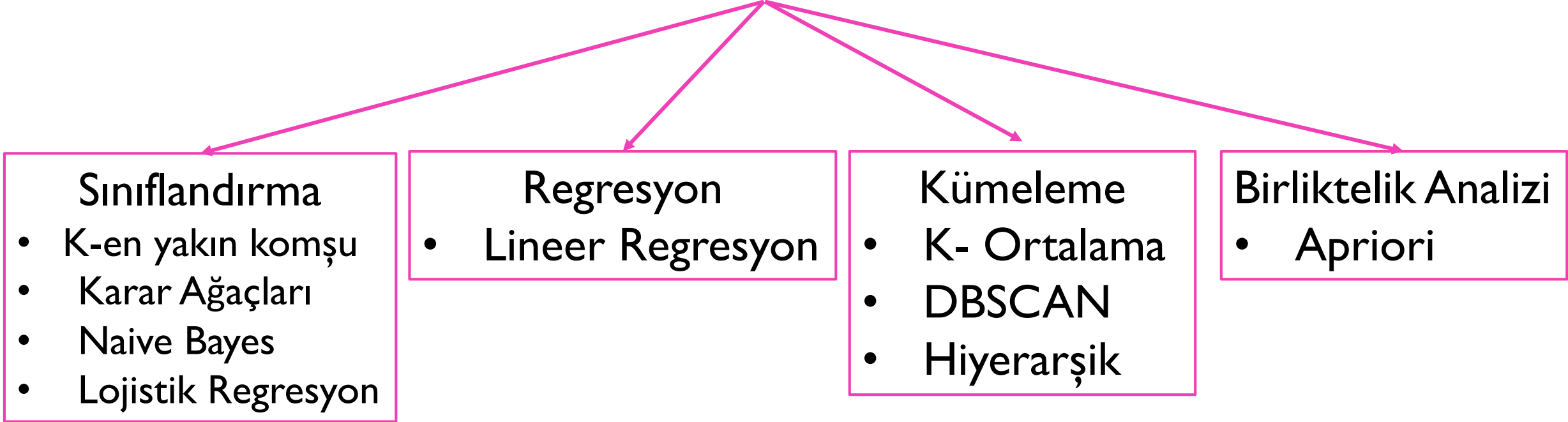
VERİ MADENCİLİĞİ

Fırat İsmailođlu, PhD

Kümeleme (Gruplama)



Veri Madenciliği Görevleri



Etiketlenmemiş (Unlabeled) Veri

Hasta Adı Soyadı	Yaş	Kilo	Boy	Sigara Alışkanlığı	Ailede Kansersiz Kişi Varlığı	Kanser
Hasta 1	45	90	178	1	1		1
Hasta 2	26	56	165	1	0		0
...							
Hasta n	78	68	163	1	0		1

Etiketlenmemiş veride sınıf bilgisi bulunmaz. Fakat yinede biz kümeleme analizi yaparak örnekler arasındaki ilişkiyi keşfedebiliriz. Bu ise veriyi anlamamızı, özetlemememizi sağlar.

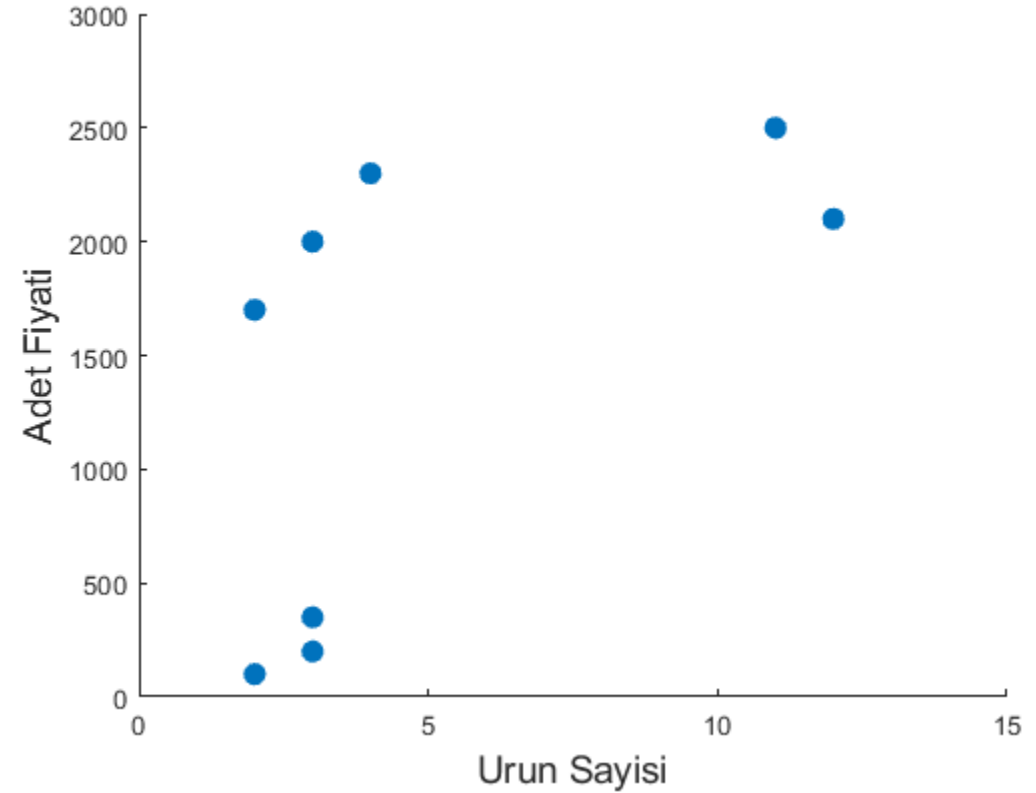
Ayrıca sınıf bilgisi verilmiş olan veri setinde dahi kümeleme analizi yapabiliriz.

Böylece elimizdeki veriyi anlamış oluruz.

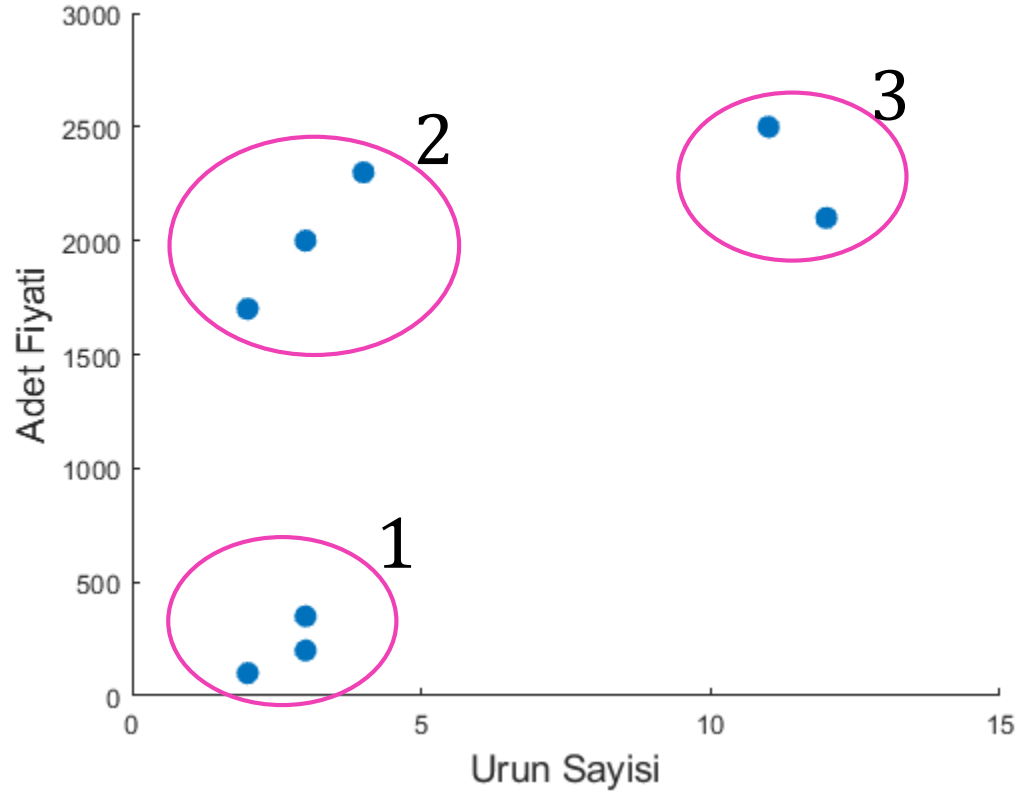


ör. Diyelimki aşağıdaki gibi bir veri setimiz olsun. Burada satırlar kişileri, özellikler (kolonlar) kişilerin satın aldığı ürün sayısını ve bu ürünlerin her birine verdikleri parayı gösterebilir.

Ürün Sayısı	Adet Fiyatı
3	350
11	2500
12	2100
2	1700
4	2300
3	2000
2	100
3	200



Bu veri seti aşağıdaki gibi üç kümeye ayrılabilir.

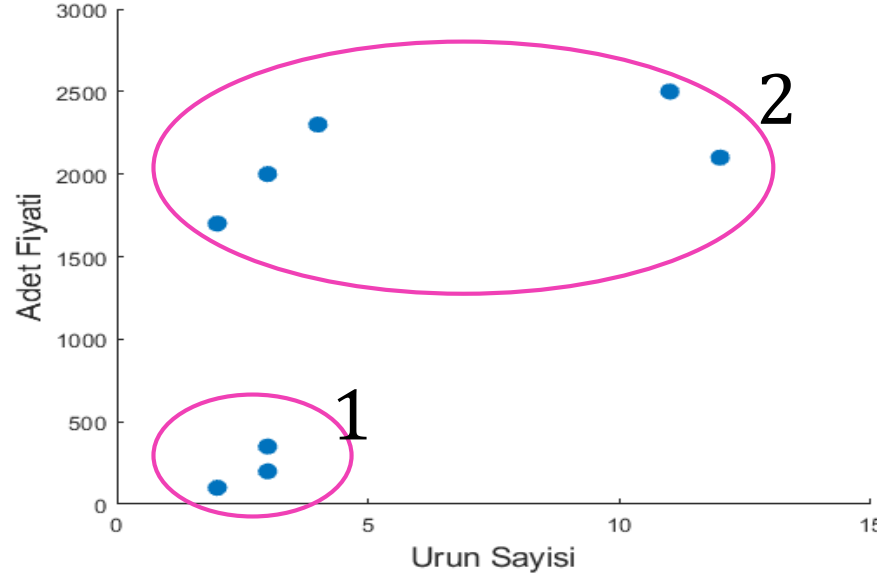


Dikkat çeken noktalar:

- Aynı kümenin içindeki elemanlar birbirine benzerdir. Örneğin 1. kümedeki kişilerin aldığı ürün sayısı azdır; bu ürünlere ödedikleri fiyatlar düşüktür.
- Farklı kümelerdeki elemanlar birbirinden farklıdır. Örneğin 1. kümedeki kişilerin aldığı ürün sayısı az ve bu ürünlere ödedikleri miktar düşük iken; 3. kümedeki kişilerin aldığı ürün sayısı fazla ve ürünlere ödedikleri miktar yüksektir.
- Kümeler farklı sayıda eleman içerebilir.
- Kümeleme analizi otomatik olarak kümeleri bulmamıza yarayan methodların çalışmasıdır.



Kaç küme olacağı sabit değildir. Kesin bir doğru yoktur. Örneğin bir önceki örnekteki veri seti iki küme olacak şekilde de kümelenebilir.



Küme Vs Sınıf

Bir sınıfın üyeleri ortak karakteristiklere sahiptir. Örneğin kanser sınıfının üyeleri belirli bazı özellikler gösterirler.

Kümeler de aynı sınıflar gibidir. Aynı kümeye ait elemanlar benzer özellik gösterirler. Fakat kümeyi sınıftan ayıran temel neden, küme başta verilmemesi, kesin olarak bilinmemeleridir.

Kümeler potansiyel sınıflardır. Bunları ortaya çıkarmak elimizdeki veriyi daha iyi anlamayı sağlar.



Kümeleme Örnekleri

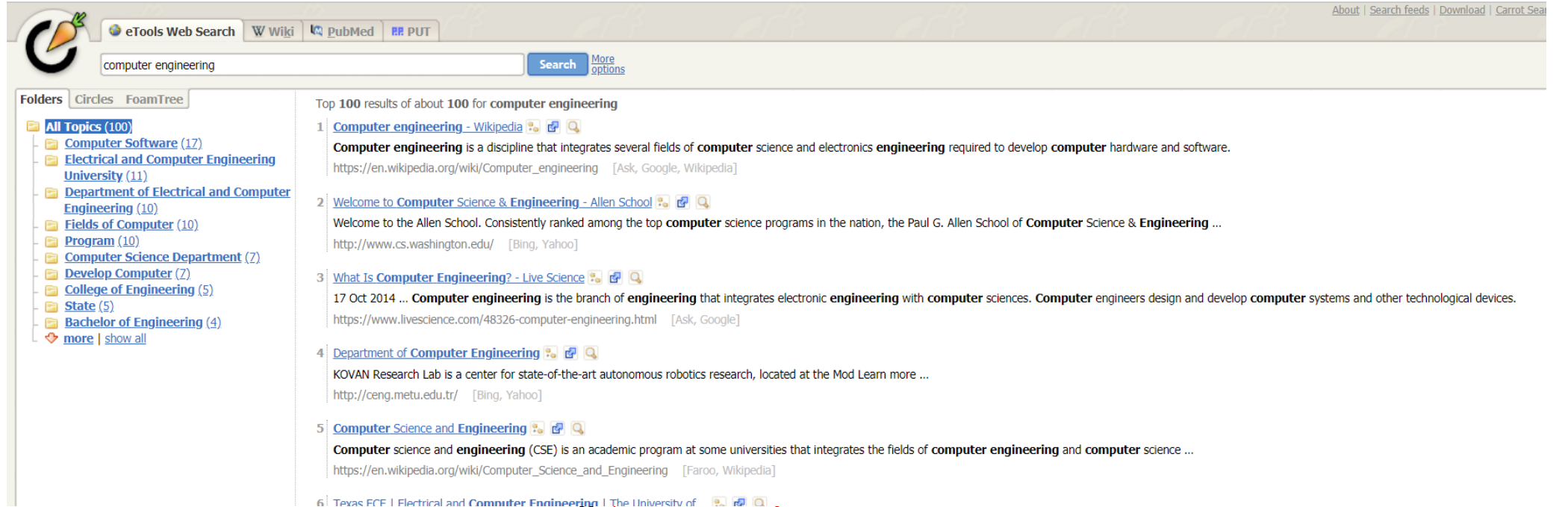
Biyoloji

Kümeleme çalışmasıyla aynı fonksiyona sahip genler tespit edilebilir.

Bilgi Erişim (Information Retrieval)

Web, milyarlarca web sayfasından oluşur ve bir arama motoru kullanarak aradığımız bir kelime binlerce sonuç içerir. Kümeleme bu sonuçların gruplanmasında kullanılabilir, oluşan her bir grup aranan kelimenin bir özelliğini yakalar.

Carrot 2 bu yaklaşımla çalışan bir arama motorudur.



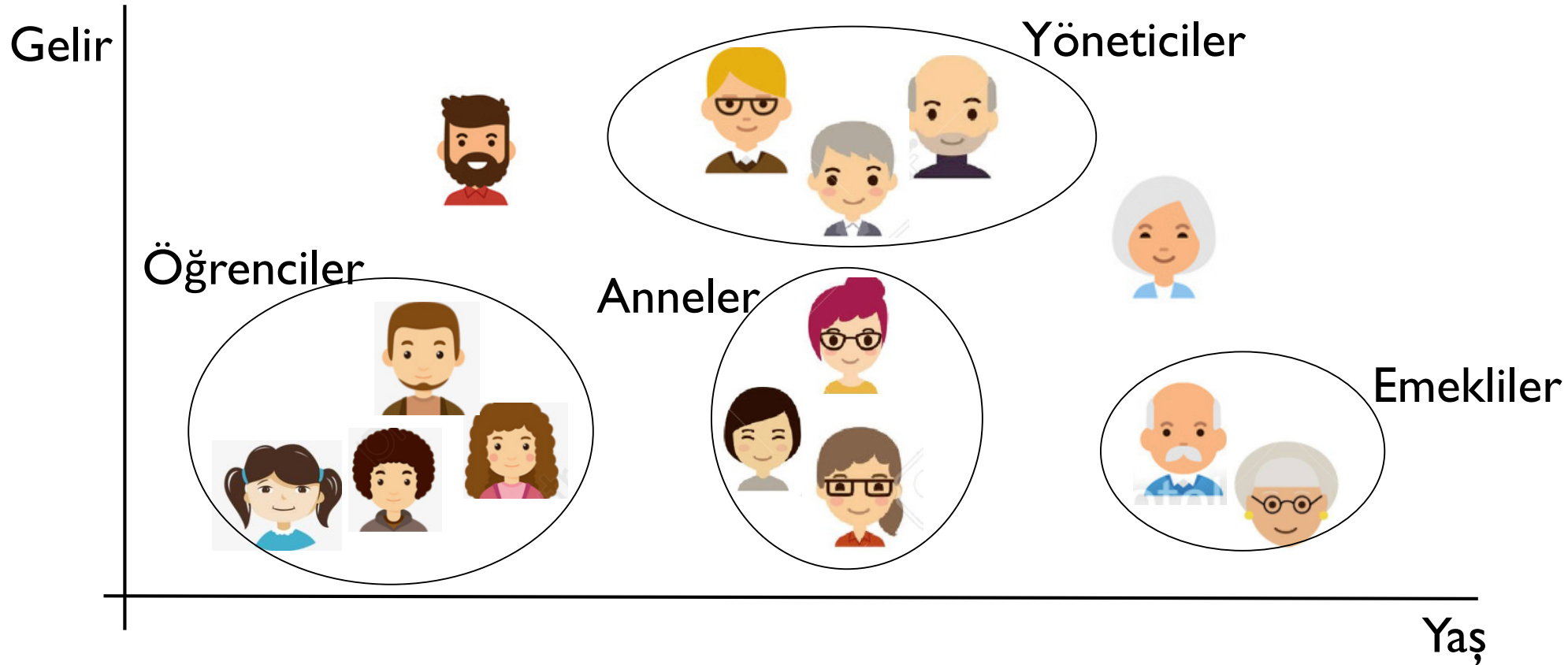
The screenshot shows the Carrot 2 search engine interface. The search bar contains the text 'computer engineering'. The search results are displayed in a list format. The first result is 'Computer engineering - Wikipedia', which is highlighted. The second result is 'Welcome to Computer Science & Engineering - Allen School'. The third result is 'What Is Computer Engineering? - Live Science'. The fourth result is 'Department of Computer Engineering'. The fifth result is 'Computer Science and Engineering'. The sixth result is 'Texas FCE | Electrical and Computer Engineering | The University of ...'. The left sidebar shows a folder structure with 'All Topics (100)' and various sub-folders like 'Computer Software (17)', 'Electrical and Computer Engineering University (11)', etc. A pink bracket on the left side of the image points to the folder structure, with the word 'Gruplar' written next to it.

Gruplar

Tıp : Genellikle bir hastalığın birden çok çeşidi olur. Tıp verileri ile yapılan kümeleme analizi, hastalıkların lat türlerinin bulunmasına yardımcı olabilir.

İşletmeler

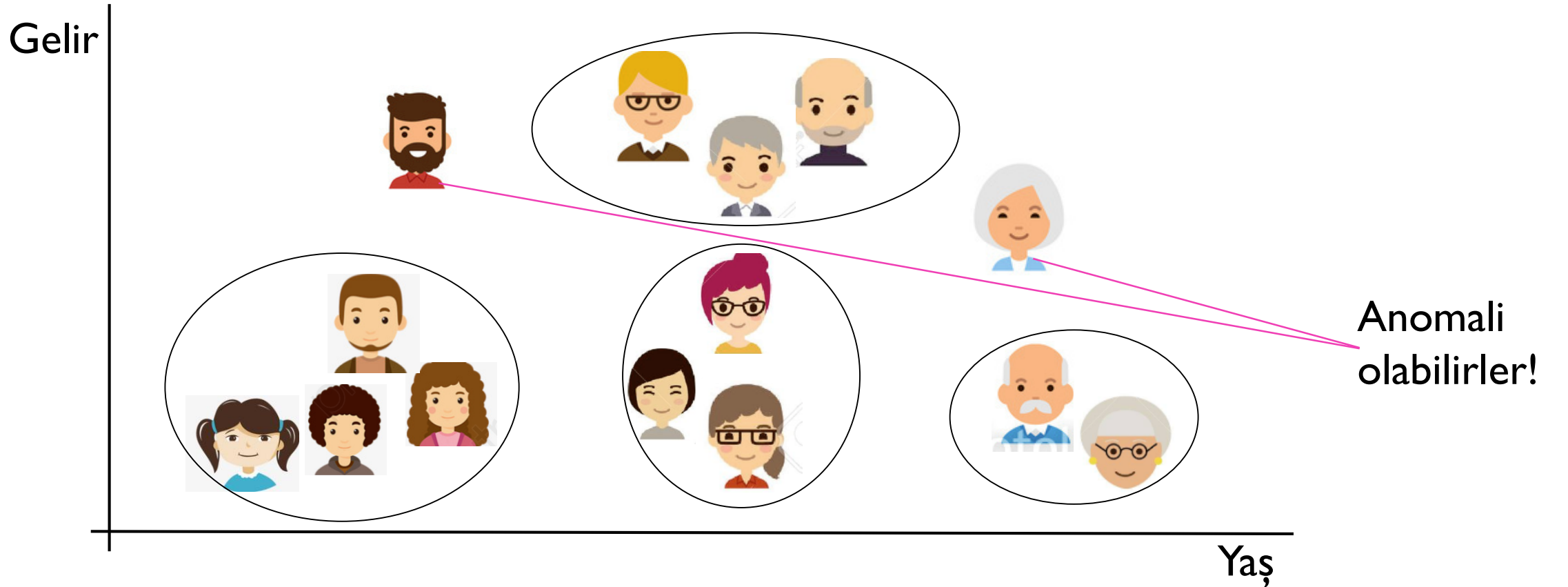
İşletmeler şu anki ve potansiyel müşterilerinden büyük miktarda bilgi toplarlar. Kümeleme, müşterileri bölümlere ayırmada kullanılabilir. Böylece her bir gruptaki müşterilere özel kampanyalar düzenlenebilir.



Kümelemenin Diğer Amaçları

I. Anomali Tespiti (Anomaly Detection)

Kümeleme ile verideki anomalileri tespit edebiliriz. Basitçe, kümeleme analizi tarafından herhangi bir kümeye dahil edilmeyen tekil örneklerin anomali olma riski yüksektir.









Kümelemenin Diğer Amaçları

2. Veri Özetleme (Data Summarization)

Küme prototipleri, dahil olduğu kümenin en tipik (most representative) elemanıdır. Bu eleman genel olarak kümedeki diğer elemanlara olan ortalama uzaklığı minimum olan elemandır.

ör.

				Ortalama Uzaklık
	0	0.4	1.1	0.5
	0.4	0	0.2	0.2
	1.1	0.2	0	0.46

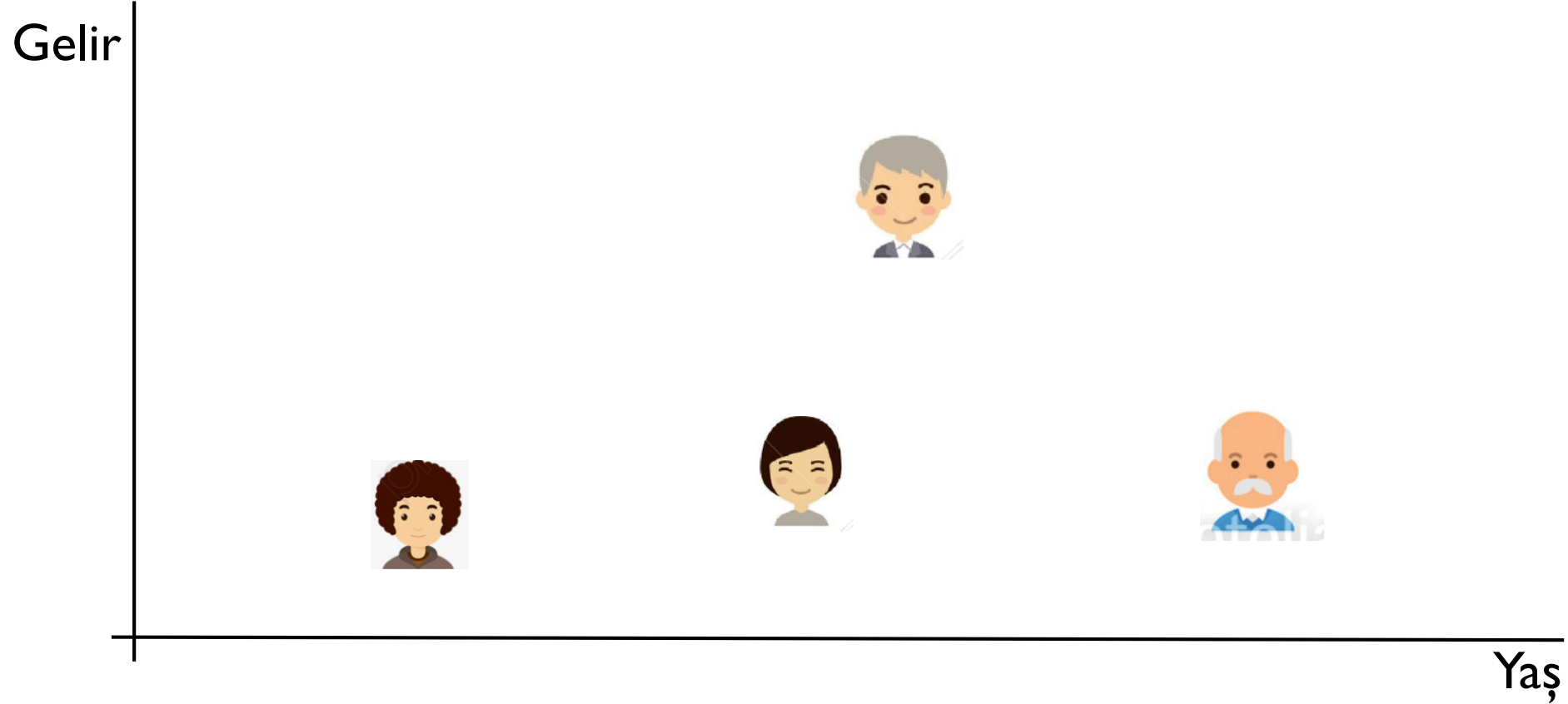
En tipik eleman ←

→ Minimum

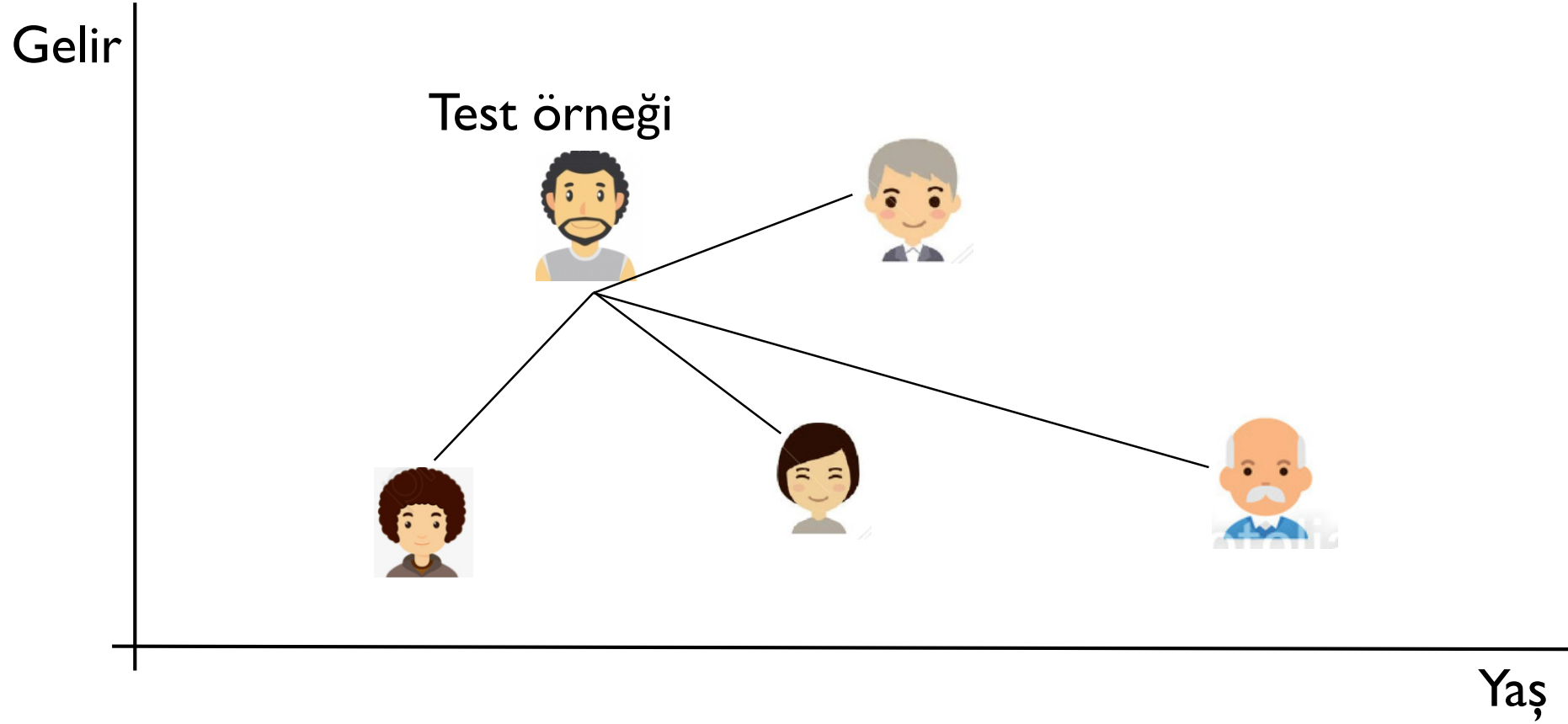


Kümelemenin Diğer Amaçları

Her kümenin yalnızca en tipik elemanını alarak veri setini özetleyebiliriz (sıkıştırabiliriz). Böylece daha az veri kaydetmiş oluruz.



Özetlenmiş veride sınıflandırma algoritmaları çok daha hızlı çalışır. Örneğin k-en yakın komşu algoritmasında bir test örneğini sınıflandırmak için yalnızca küme protipleri ile olan uzaklıklarını hesaplarız.



Kümelemede En Çok Kullanılan Uzaklık Fonksiyonları

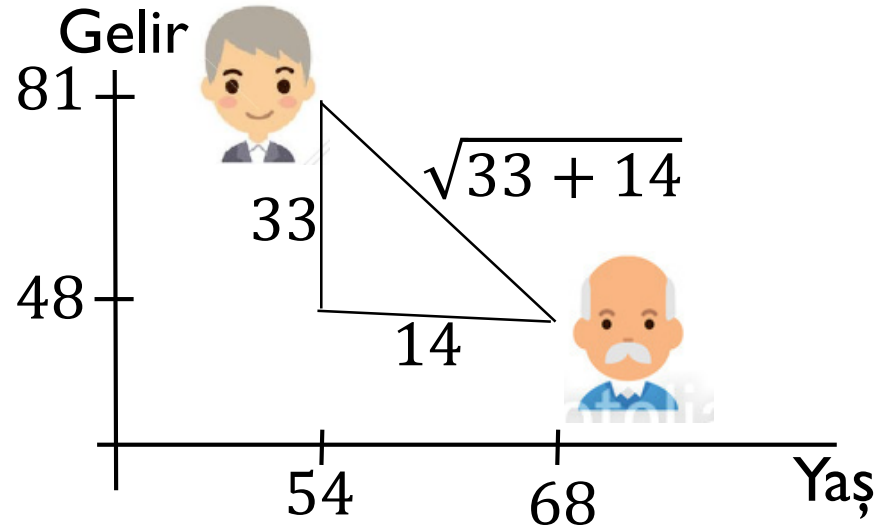
Kümeleme analizi yapabilmemiz için örnekler arası uzaklığın tanımlı olması gerekir; yani bir uzaklık fonksiyonumuzun var olması gerekir. Böylece hangi örnekler birbirine benzer, hangi örnekler birbirinden ayırılır hesaplayabiliriz.

Bu anlamda en çok kullanılan uzaklık fonksiyonu Öklid uzaklığı ve Cosine benzerliğidir.

I. Öklid Uzaklığı

$x^1 = (x_1^1, x_2^1, \dots, x_d^1)$ ile $x^2 = (x_1^2, x_2^2, \dots, x_d^2)$ örnekleri arası Öklid uzaklığı:

$$d_{\text{öklid}}(x^1, x^2) = \sqrt{(x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2 + \dots + (x_d^1 - x_d^2)^2}$$



2. Cosine Benzerliği

Cosine benzerliği daha çok dökümanlar arası uzaklığı hesaplarken kullanılır.

$x^1 = (x_1^1, x_2^1, \dots, x_d^1)$ ile $x^2 = (x_1^2, x_2^2, \dots, x_d^2)$ örneklerinin (vektörlerinin) iç çarpımı:

$$\langle x^1, x^2 \rangle = \underbrace{\|x^1\|}_{x^1 \text{ vektörünün}} \cdot \underbrace{\|x^2\| \cdot \cos \Theta}_{x^1 \text{ ve } x^2 \text{ vektörleri arasındaki açının cosinüsü}}$$

x^1 vektörünün
uzunluğu

x^1 ve x^2 vektörleri arasındaki
açının cosinüsü

Buradan $\cos \Theta$ yalnız bırakılırsa

$$\cos \Theta = \frac{\langle x^1, x^2 \rangle}{\|x^1\| \cdot \|x^2\|}$$

$\cos \Theta$ benzerliğin (yakınlığın) ölçüsüdür. Uzaklık için bu değeri 1'den çıkarırız: $1 - \cos \Theta$.



2. Cosine Benzerliđi

ör. 1. cümle: 'Seni sevmeyen ölsün', 2. cümle: 'Sev seni seveni', 3. cümle: 'Sevmekten kim usanır'
cümlelerinin birbirlerine olan cosine benzerliklerini bulunuz.

	1. cümle	2. cümle	3. cümle
Sen	1	1	0
Sevmek	1	2	1
Ölmek	1	0	0
Kim	0	0	1
Usanmak	0	0	1

1. cümle (1,1,1,0,0); 2. cümle (1,2,0,0,0); 3. cümle (0,1,0,1,1);

$$\cos(1. \text{ cümle}, 2. \text{ cümle}) = \frac{1 \cdot 1 + 1 \cdot 2 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0}{\sqrt{1^2 + 1^2 + 1^2} \cdot \sqrt{1^2 + 2^2}} = 0.77$$



Kümeleme Algoritmaları

I. K- Ortalama Algoritması (K-Means Algorithm)

En çok kullanılan kümeleme algoritmasıdır. İteratif bir algoritmadır.

İteratif bir algoritmadır.

İterasyona başlamadan önce:

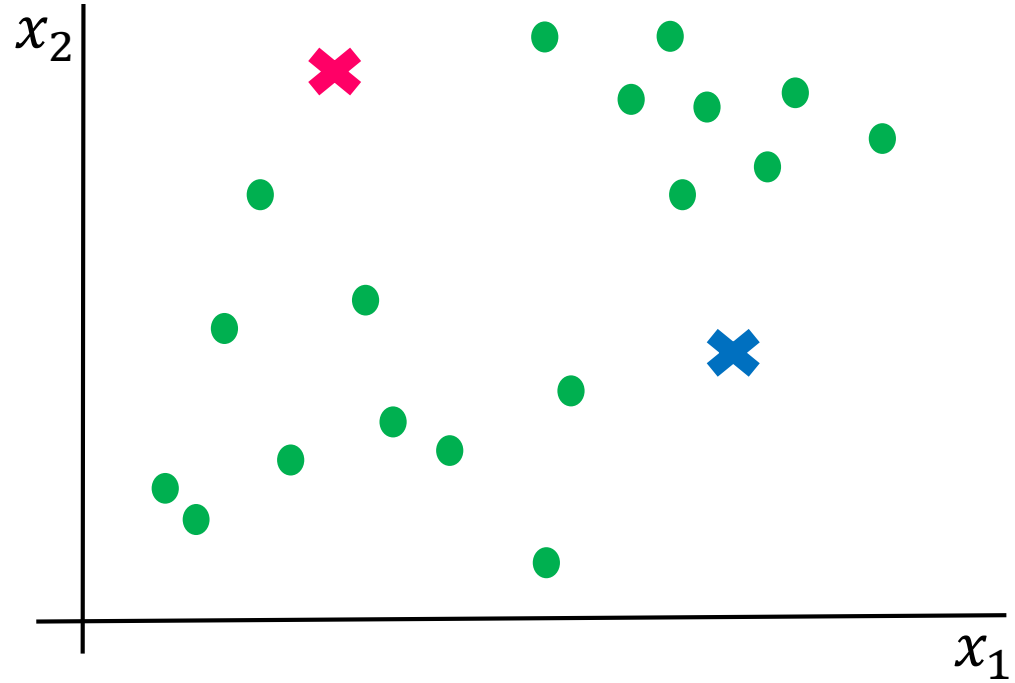
1. Küme sayısına karar ver.
2. Rastgele küme merkezleri belirle.

Her bir iterasyonda:

1. Her bir örneğin küme merkezlerine uzaklıklarını hesapla; küme merkezine en yakın olduğu kümeye dahil et.
2. Küme merkezlerini güncelle.



K- Ortalama Algoritması (K-Means Algorithm)



● Kümelenmemiş veri örnekleri

✕ Rastgele seçilmiş başlangıç küme merkezleri

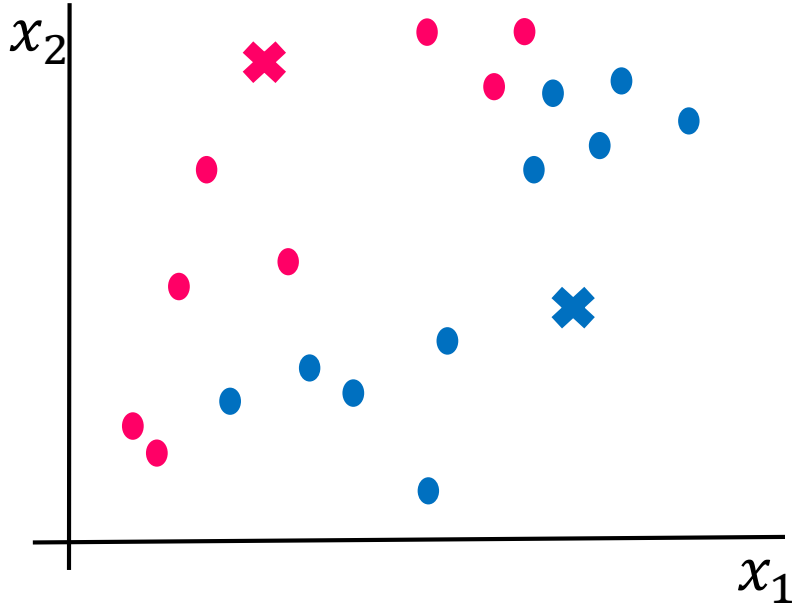
Küme sayısı iki olsun. Bu kümelerin başlangıç merkezlerini ise şekilde görüldüğü gibi rastgele seçelim. İterasyonu başlatalım.



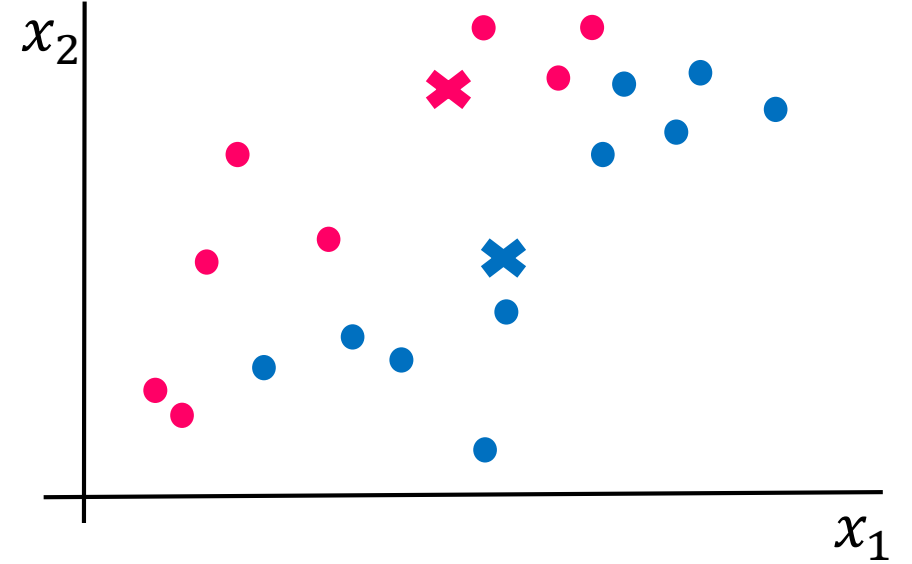
K- Ortalama Algoritması (K-Means Algorithm)

iter 1

Her bir örneğin merkezlere uzakliklarini hesapladik.
En yakin olduđu merkeze göre kümeledik.



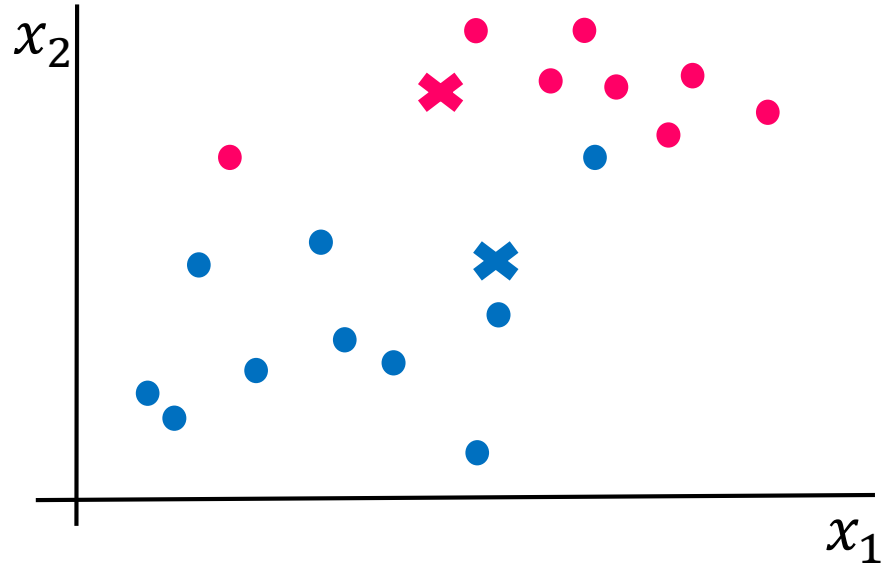
Yeni bulunan kümelere göre küme merkezlerini güncelledik.



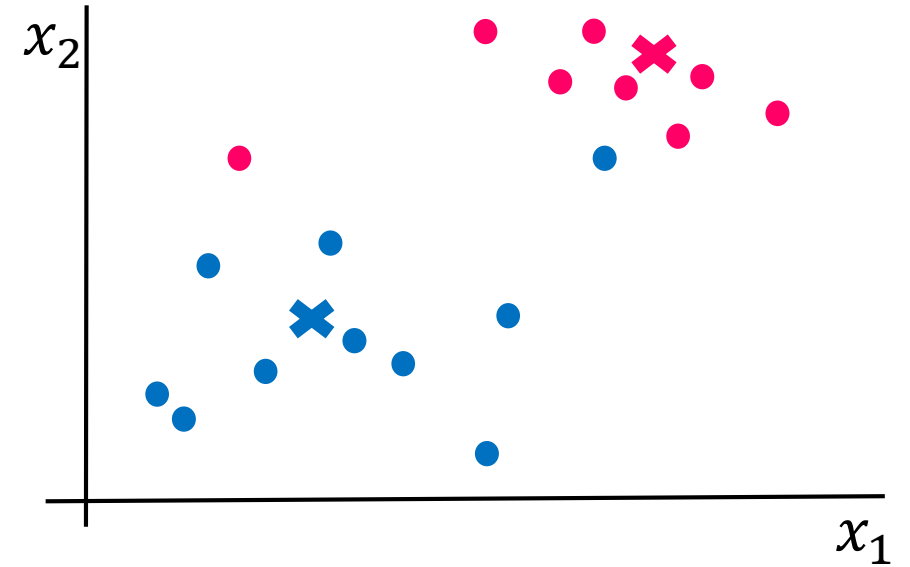
K- Ortalama Algoritması (K-Means Algorithm)

iter 2

Her bir örneğin yeni merkezlere uzakliklarini hesapladik. En yakin olduđu merkeze göre kümeledik.



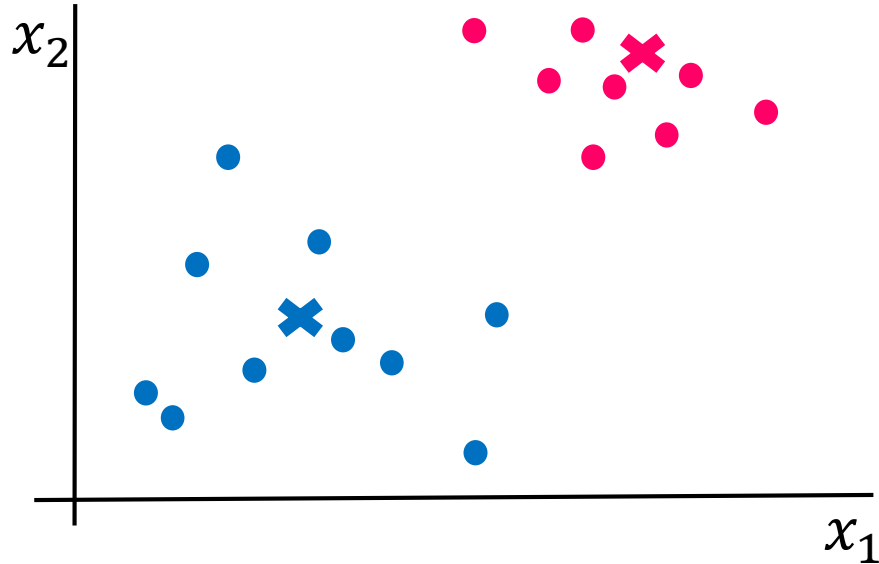
Yeni bulunan kümelere göre küme merkezlerini güncelledik.



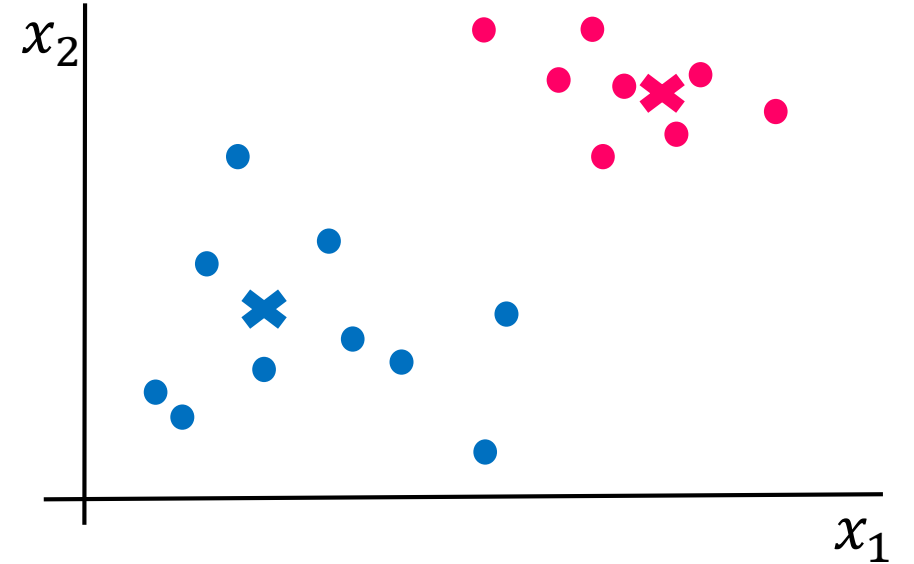
K- Ortalama Algoritması (K-Means Algorithm)

iter 3

Her bir örneğin yeni merkezlere uzakliklarini hesapladik. En yakin olduđu merkeze göre kümeledik.



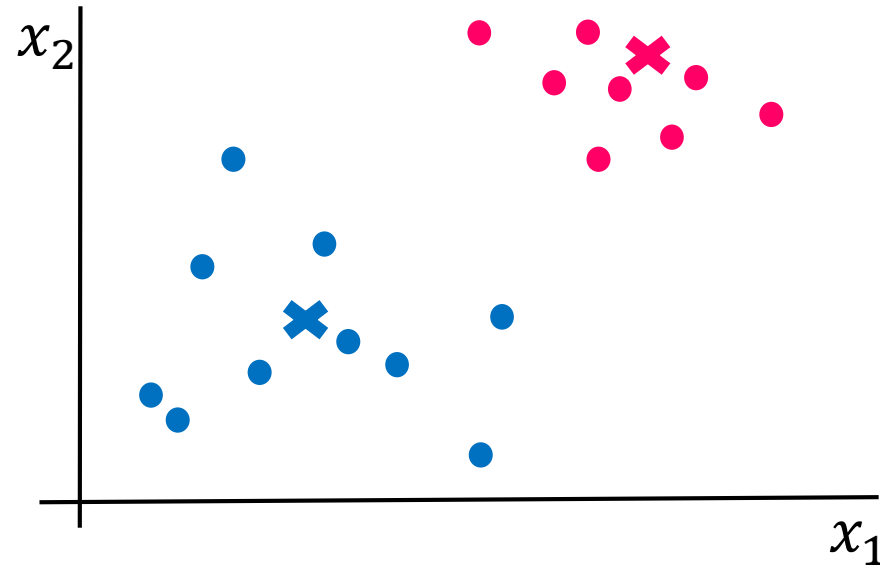
Yeni bulunan kümelere göre küme merkezlerini güncelledik.



K- Ortalama Algoritması (K-Means Algorithm)

iter 4

Her bir örneğin yeni merkezlere uzakliklarini hesapladik. En yakin olduđu merkeze göre kümeledik. Fakat gördükki hiç bir örneğin dahil olduđu küme deđişmedi. Yani kümeler deđişmedi. İterasyonu burda sonlandırabiliriz.

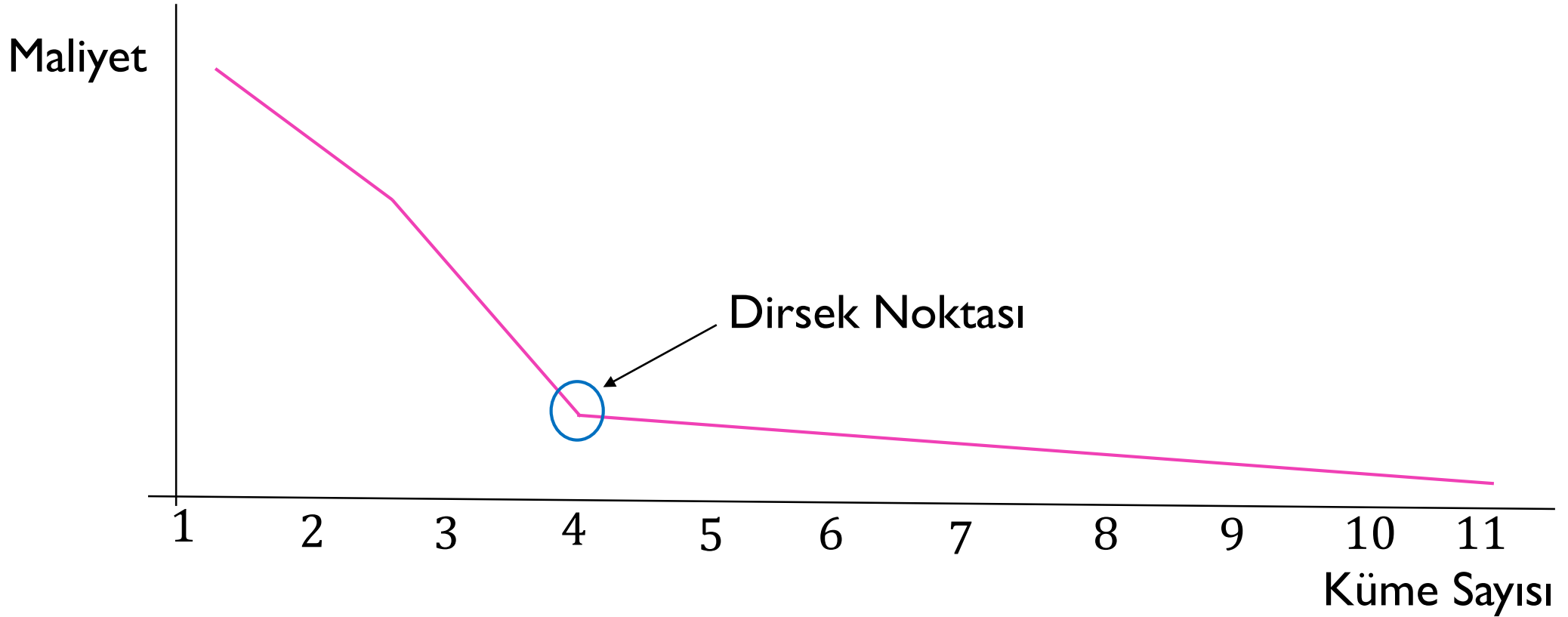


K-Ortalama Algoritmasında Küme Sayısına Nasıl Karar Veririz?

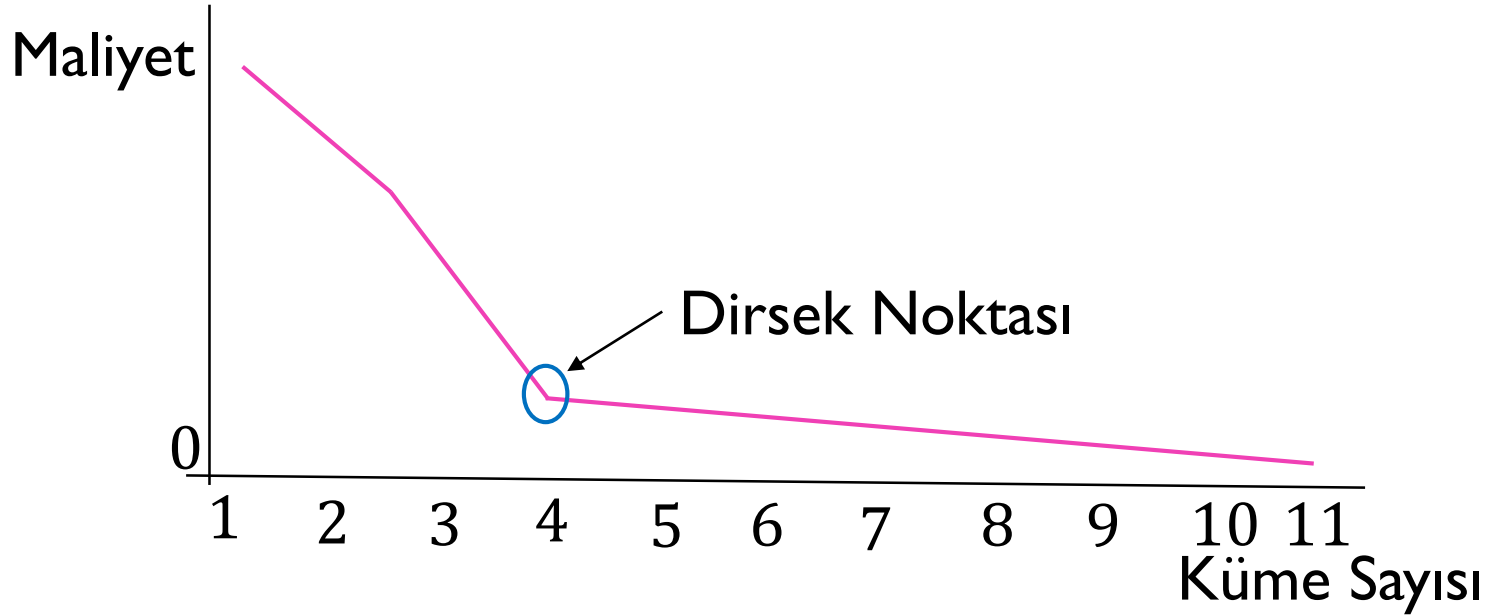
Geçen hafta k-ortalama algoritması ile kümeleme yaparken veri setinde kaç küme olacağını önceden bilemeyiz demiştik. Küme sayısını kullanıcıdan almıştık.

Bu hafta ise küme sayısını dirsek methodu (elbow method) ile tahmin etmeye çalışacağız.

Dirsek Methodu (Elbow Method)



Dirsek Methodu (Elbow Method)



4 kümeden sonra maliyette anlamlı bir azalış gözlemlenmiyor. Bu yüzden bu veri seti için ideal küme sayısı 4'tür.

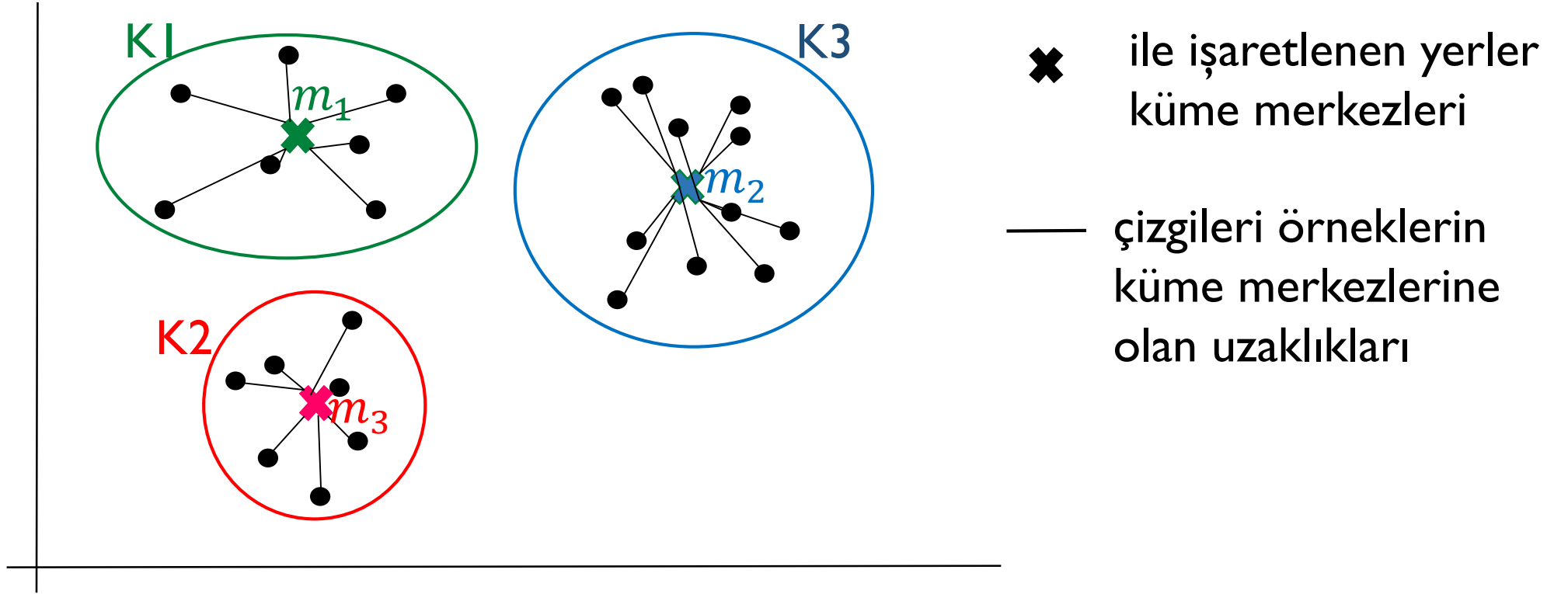
Peki maliyet k-ortalama ile kümeleme yaparken ne anlama geliyor?

Maliyet, genel olarak istemediğimiz durumların toplam değerinin sayısal ifadesidir.

Kümeleme yaparken de istemediğimiz şey her örneğin kendi küme merkezine uzak mesafede olmasıdır. Bu, cezalandırılması gereken bir durumdur.



Kümelemede Maliyet Hesabı



Maliyet yukarıda gösterilen siyah çizgilerin uzunluklarının toplamıdır. Bunun sayısal ifadesi için diyelimki S_1, S_2, \dots, S_k gibi k tane kümemiz ve m_1, m_2, \dots, m_k bu kümelerin merkez noktaları olsun.

Kümelemede Maliyet Hesabı

Maliyet:

$$\sum_{i=1}^k \sum_{x \in S_i} (x - m_i)^2$$

Her bir küme için

Kümemenin her elamanı için

Not: Bu şekilde hesaplanmış maliyete küme içi toplam varyasyon (total within-cluster variation) da denir.

Soru: Maliyet ne zaman 0 olur?

Cevap:

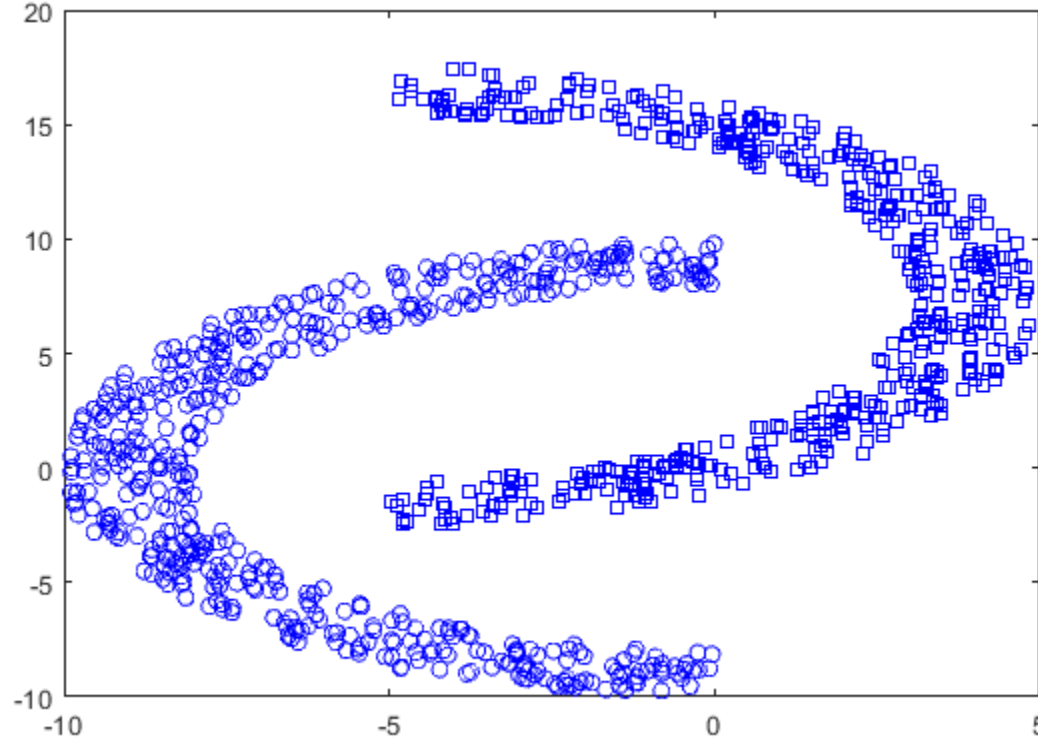


K-Ortalama Algoritmasının Zayıflıkları

K-ortalama algoritması basit fakat güçlü bir algoritmadır. Bir çok durumda verideki kümeleri bulmayı sağlar.

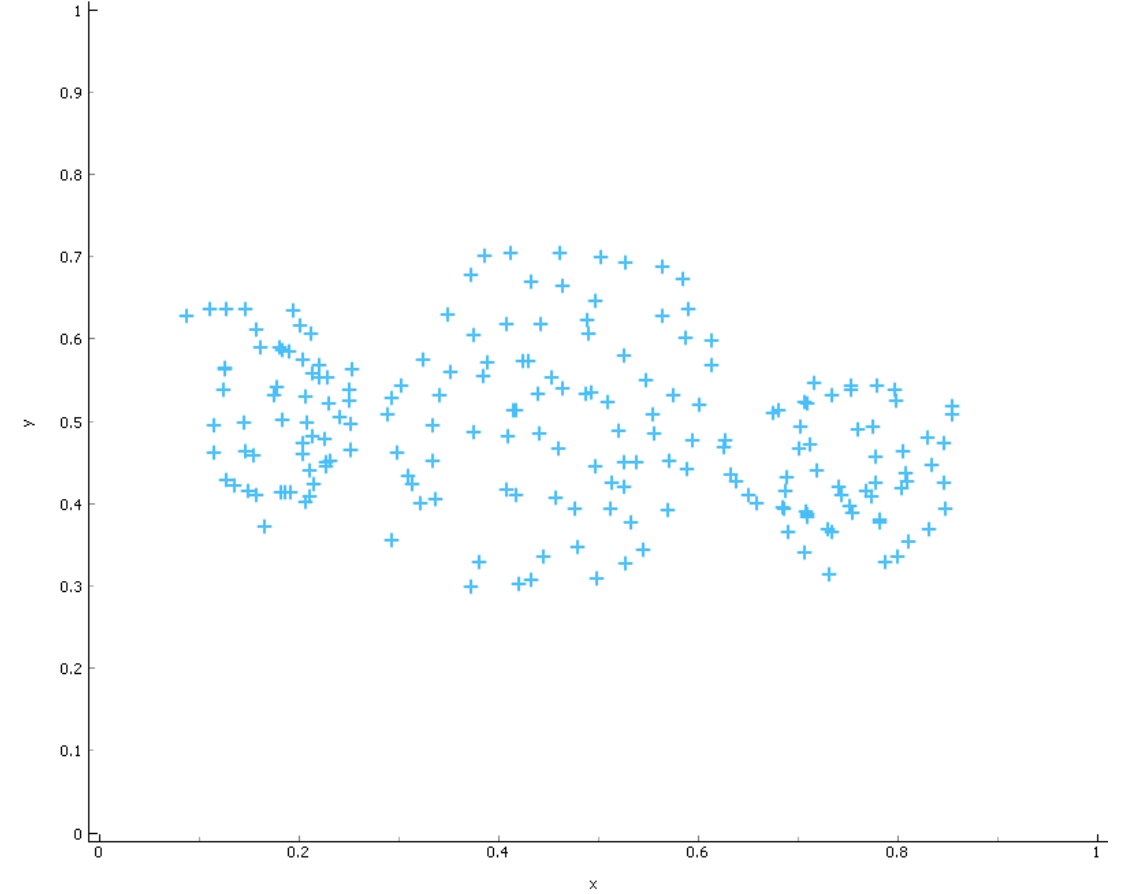
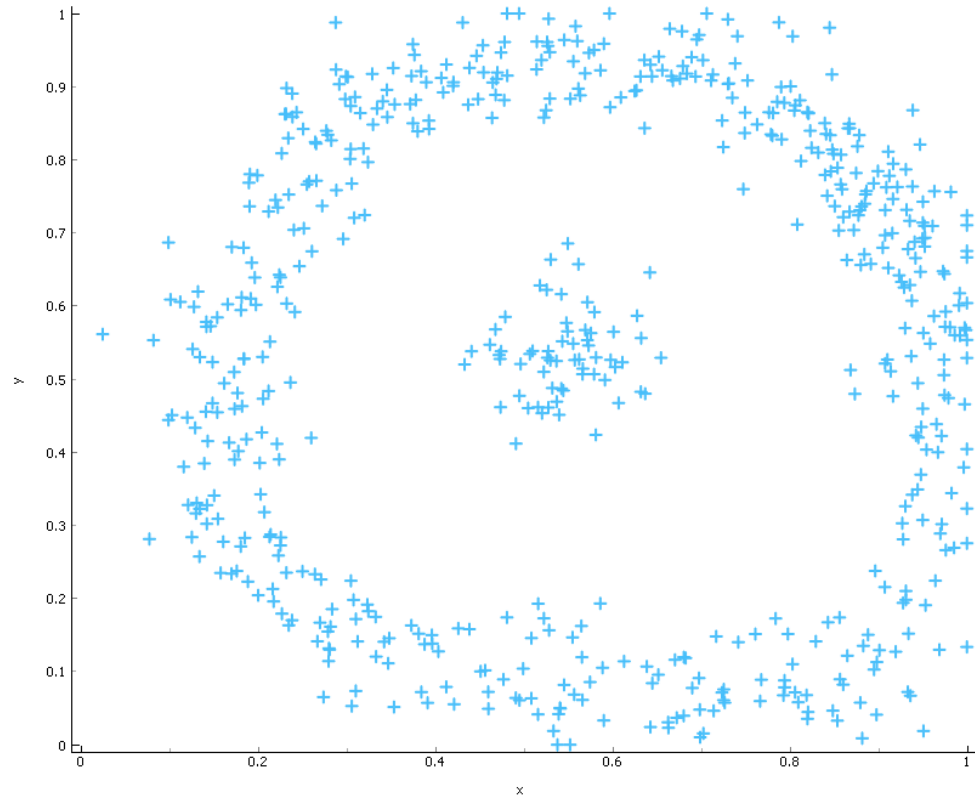
Fakat k-ortalama algoritması bazı durumlarda kümeleri bulamaz.

1. eğer verideki kümeler küresel (spherical) değilse:



K-Ortalama Algoritmasının Zayıflıkları

2. eğer verideki kümeler içiçe geçmişse, yada birbirine çok bitişikse



K-Ortalama Algoritmasının Zayıflıkları

K-ortalama tarafından yanlış bulunan kümeler

