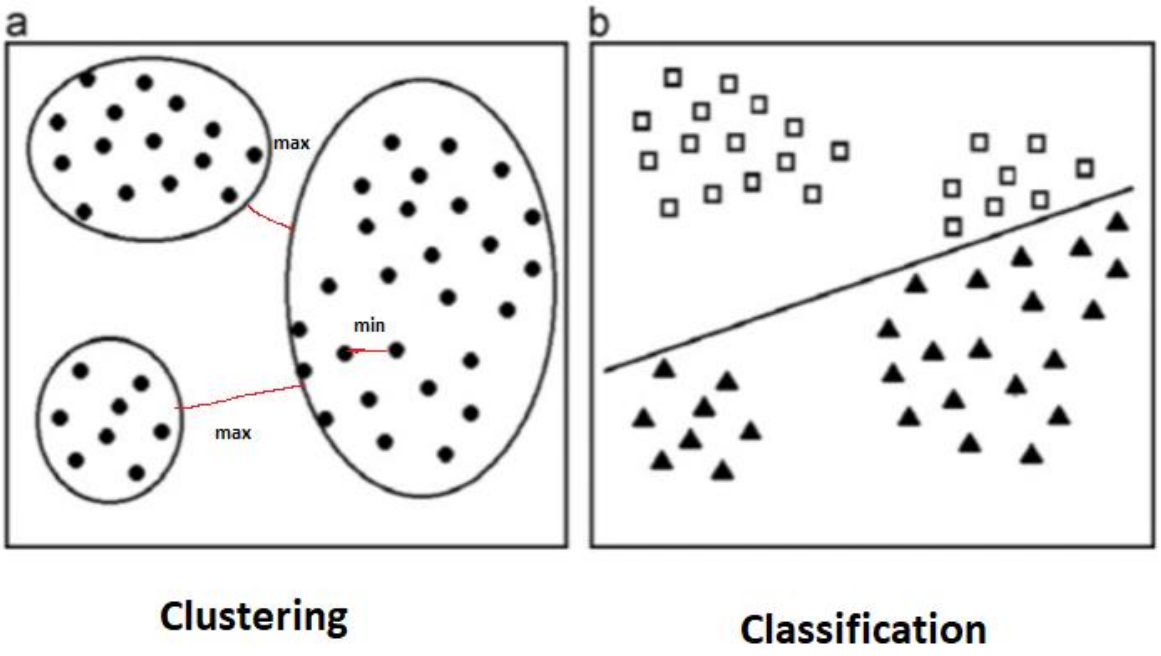
Clustering (Bölütleme / Kümeleme)

Logistic regerssion = sınıfları sayıya donusturerek tahmin

Gözetimsiz Öğrenmeye Giriş (Unsupervised Learning)



Classification: Ön bir sınıf tanımı var , supervise(gözetimli), kendi gözlemimizi makineye öğretiyoruz.

Clustering: Makine tamamen özgür, ön bilgi yok, makine kendi dunyasında kendi algıladığı sekilde problemi tanımlıyor, kendi sınıflarını veriye bakarak oluşturuyor. unsupervise(gözetimsiz)

Kullanılan alanlar;

Müşteri segmentasyonu(müşteri verilerini al kendin bu verilerden bölütler çıkar)

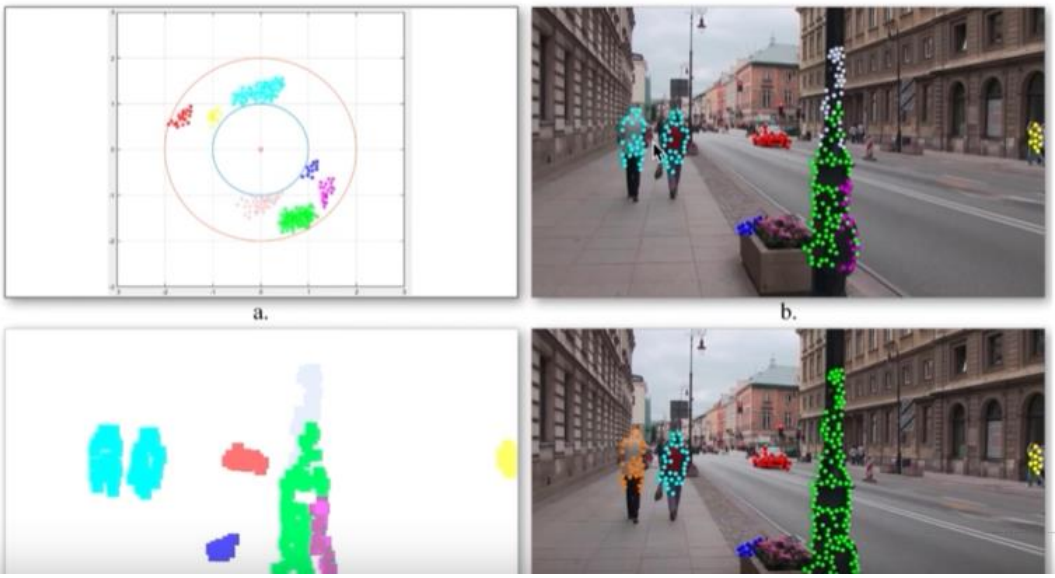
- Kampanyaların özelleştirilmesi
-

Pazar segmentasyonu

Sağlık ve görüntü işleme

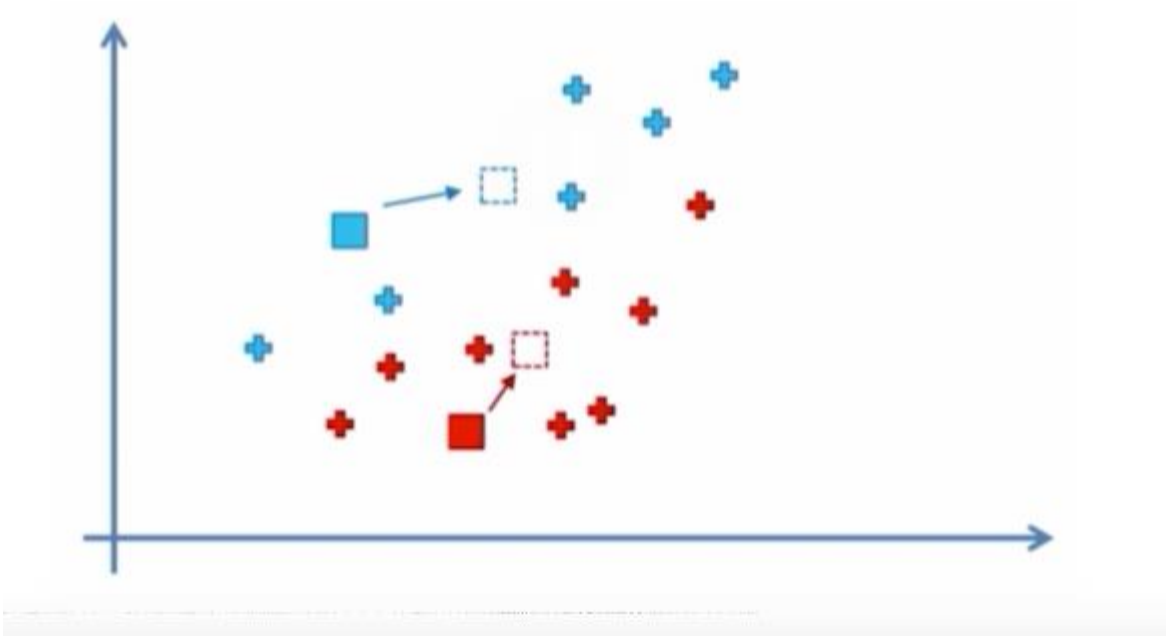
Verinin alt kümesi üzerinde yapılan bütün işlemler

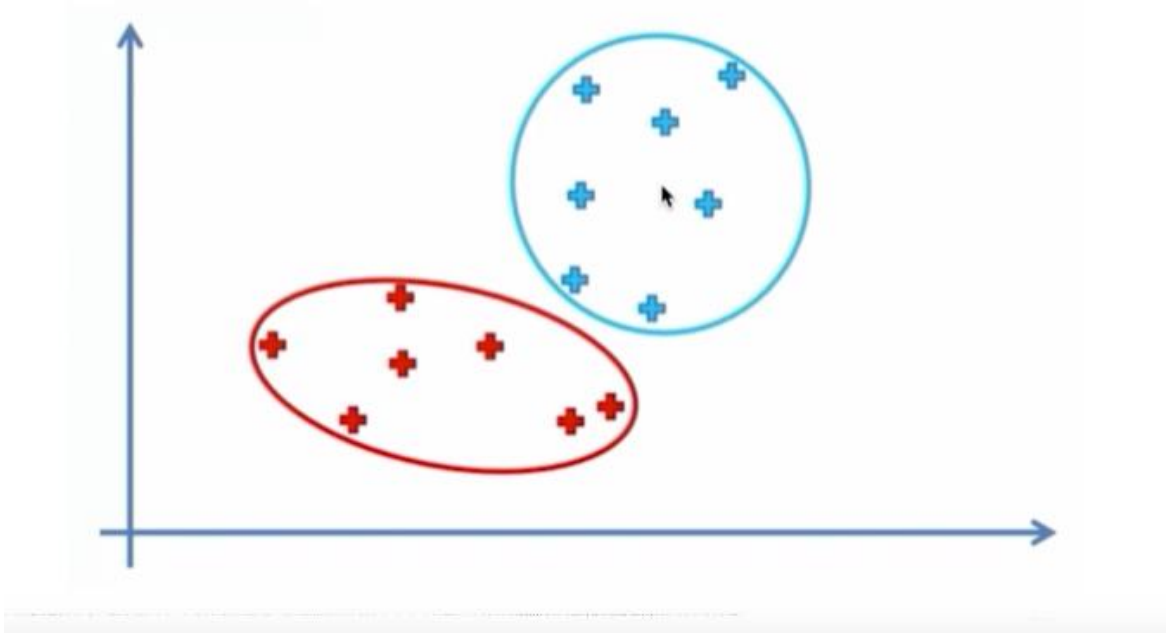
Bilgisayar ile Görü / Kümeleme



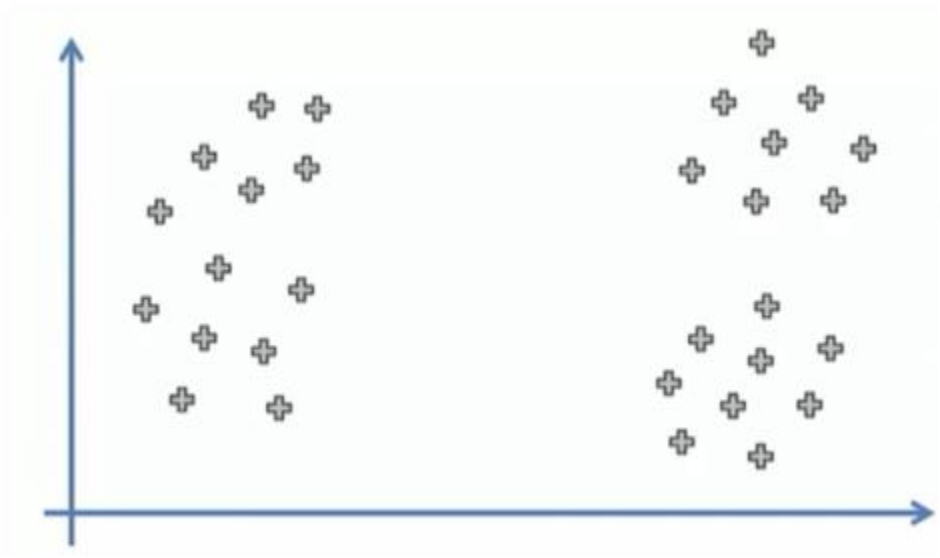
K-Means
K-Ortalama

- Nasıl Çalışır?
  - Kaç Küme olucağı kullanıcıdan parametre olarak seçilir
  - Rasgele olarak k merkez noktası seçilir
  - Her veri örneği en yakın merkez noktasına göre ilgili kümeye atanır.
  - Her küme için yeni merkez noktaları hesaplanarak merkez noktaları kaydırılır
  - Yeni merkez noktalarına göre

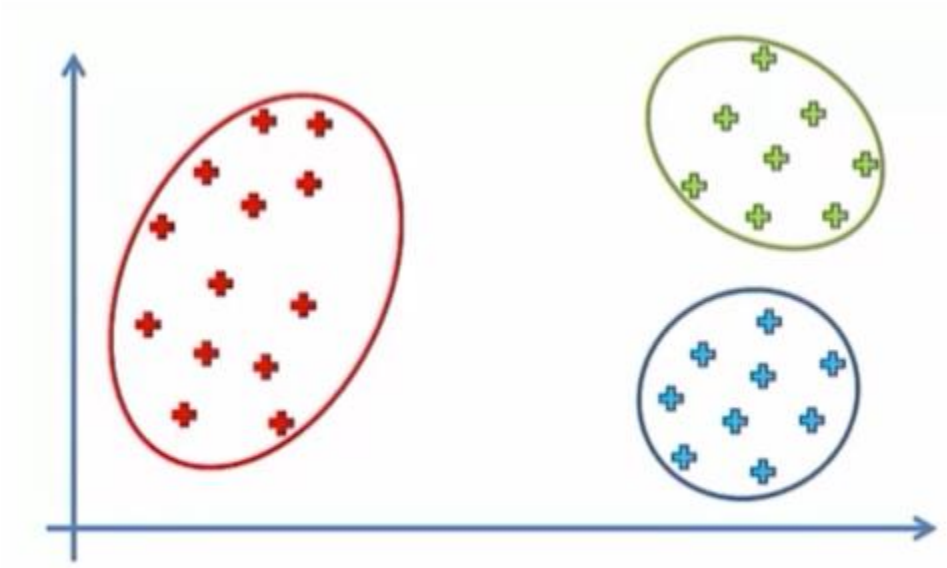




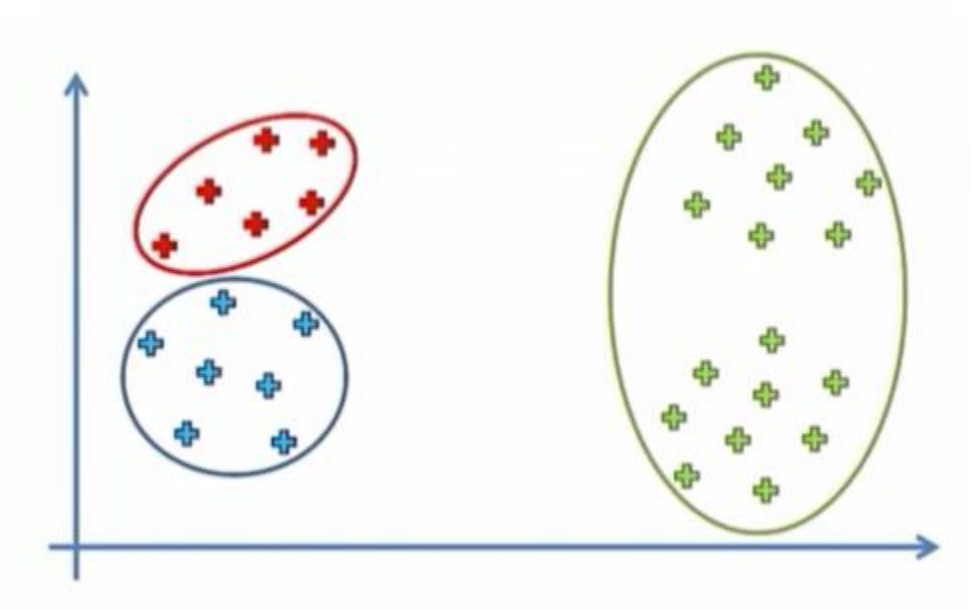
K-MEANS BASLANGIÇ NOKTASI TUZAĞI (RASSAL BAŞLANGIÇ TUZAĞI)



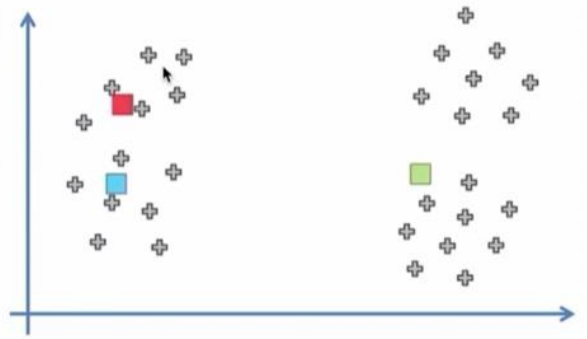
Bu örnek için beklenen;



Eğer uzayda rastgele noktalar seçilerek başlanırsa aşağıdaki gibi bir sonuc alinabilir;



Bunu iyileştirmek için;



K-Means ++
1. Rasgele seçilen noktalardan en yakınına her noktadan uzaklığı hesapla (buna D(x) diyelim)
2. Yeni noktaları mesafenin karesini olasılık alarak ($D(x)^2$ ile bul)

Bu durumda mavinin sağa atlama ihtimali daha yüksek

K Değerinin kaç olacağını karar vermek de ciddi problemlerden birisi.

X-means algoritması x yerine belirledigin aralıktaki sayıların tek tek denenemsi 2'den 50'ye -> 2 kümeden 50 kümeye kadar böl)

K-means ve K-means++ algoritmaları, kümeleme problemlerini çözmek için kullanılan popüler yöntemlerdir. İkisi arasındaki ana fark, küme merkezi başlangıçlarını seçme yöntemleridir.

K-means: Bu algoritma, başlangıçta rastgele seçilen k merkez noktası etrafında kümeleme yapar. Bu rastgele seçim, algoritmanın küresel minimumu bulma garantisi olmadığı için performansı ve sonuçların kalitesini etkileyebilir. Başlangıçta rastgele seçilen merkez noktalar, algoritmanın yerel minimuma takılmasına ve istenmeyen sonuçlara yol açabilir.

K-means++: Bu algoritma, merkez noktalarını daha dikkatli bir şekilde seçerek K-means algoritmasının bu dezavantajını gidermeyi amaçlar. Başlangıçta, ilk merkez noktası rastgele seçilir, ardından diğer merkez noktaları, daha önce seçilen merkezlere uzaklıkları dikkate alınarak seçilir. Bu, merkez noktalarının daha homojen ve veriye daha uygun bir şekilde seçilmesini sağlar. K-means++ algoritması, genellikle K-means'a göre daha hızlı ve daha iyi sonuçlar verir.

K-means++, K-means algoritmasının iyileştirilmiş bir versiyonudur ve genellikle daha istikrarlı ve daha hızlı kümeleme sonuçları sağlar. Bu nedenle, veri kümesinin yapısını daha iyi yansıtan ve daha düşük bir hata ile sonuçlanan K-means++ tercih edilir.

Clustering’in başarısının ölcülmesi icin (x-means veya k-means);

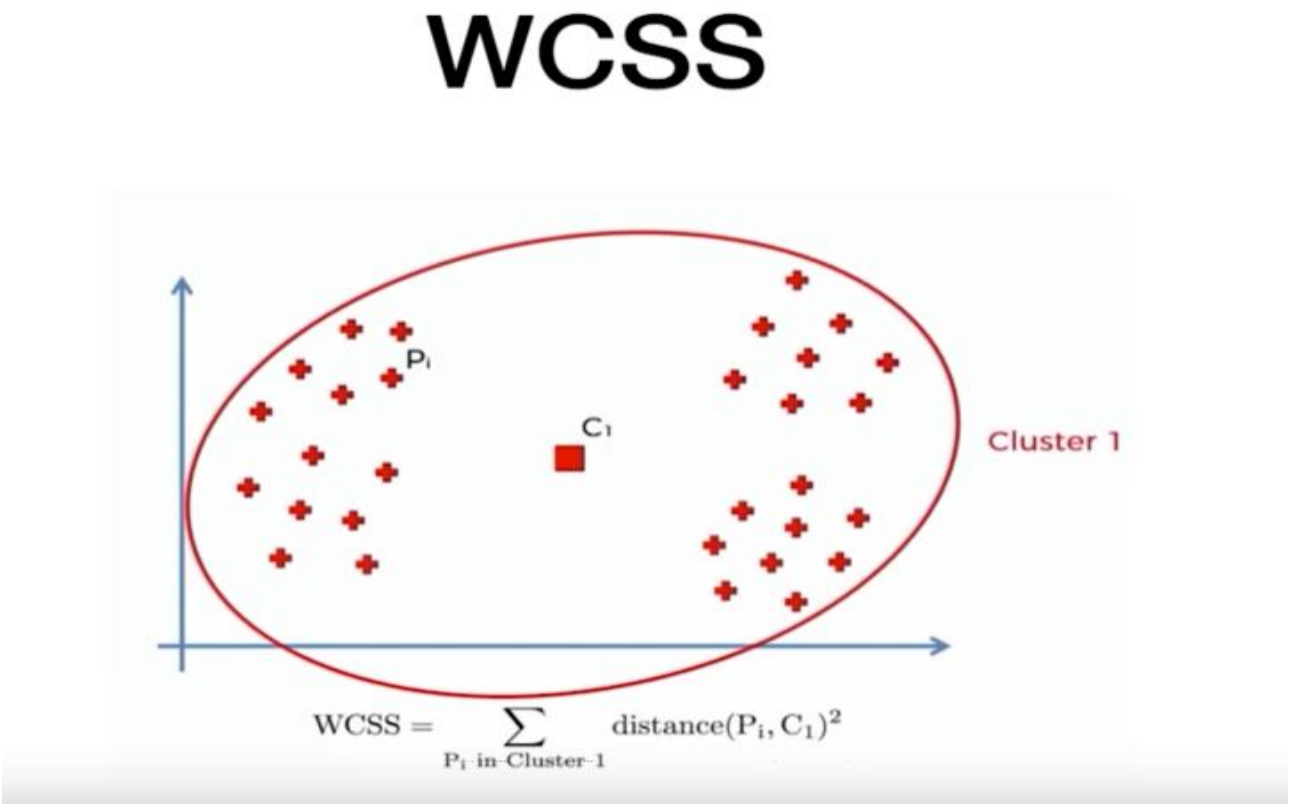
WCSS

within-cluster sums of squares

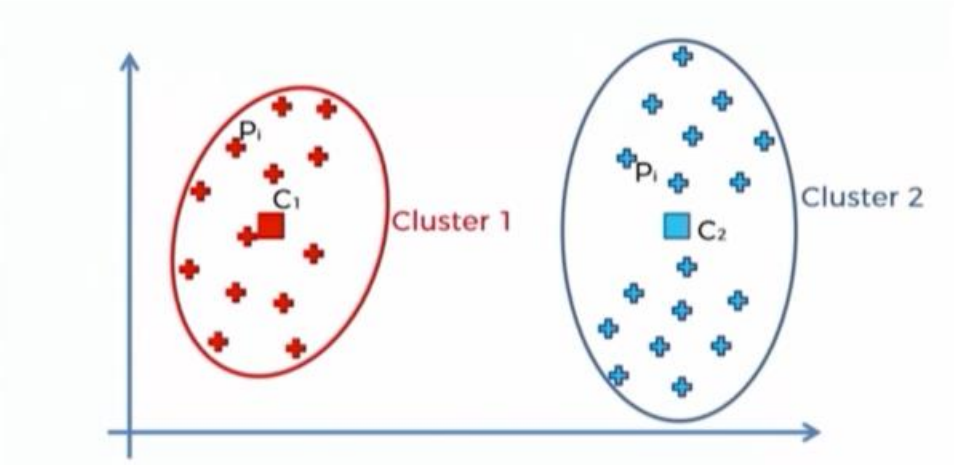
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

within-cluster sum of squares -> küme içi kareler toplamı

Her bir cluster için o cluster içindeki elemanların merkeze olan mesafelerinin kareleri.

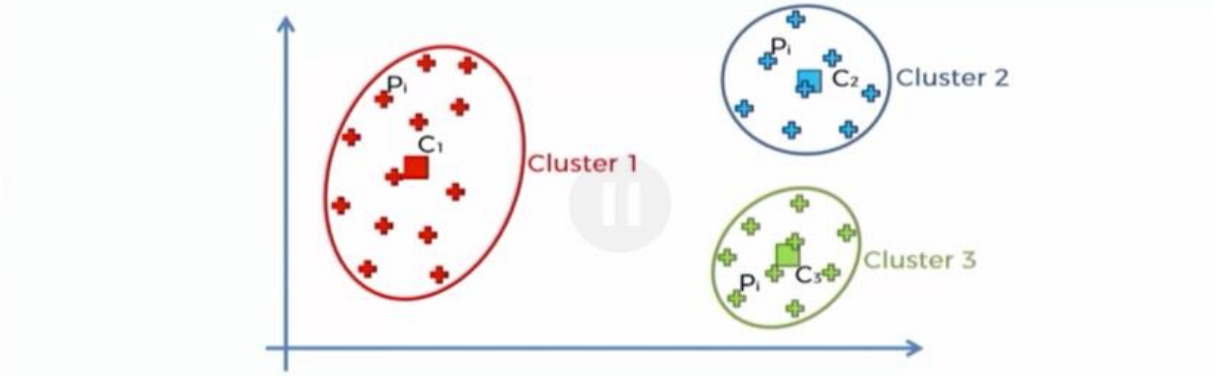


WCSS

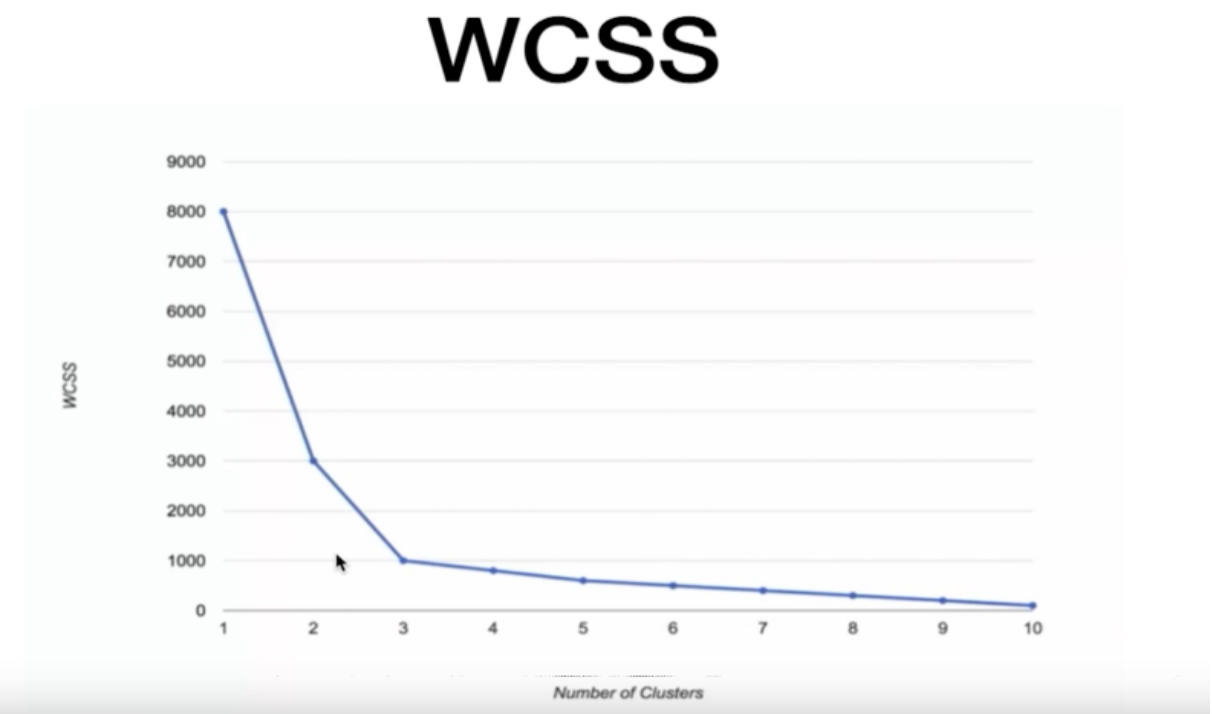


$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2$$

WCSS



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$



[cluster sayısı arttıkca wcss değer düşüyor}

Elimizde 10 veri var ise 10 cluster oluşması wcss değerini 0 yapar . bu demek değildir ki 10 tane cluster oluşturmalıyız. Bu durum overfitting (aşırı öğrenme,ezberleme) örneğidir.

Genelde böyle durumlarda elbow poin(dirsek noktası,kırılma noktası) seçilir .

Yukaridaki örnekte 3 cluster seçilmesi

Bu durum K sayısı(cluster sayısı) belirlenemediği durumda kullanılmalı

Veri dağılımına göre clustering metodu seçmek için;

2.3. Clustering — scikit-learn 1.4.1 documentation

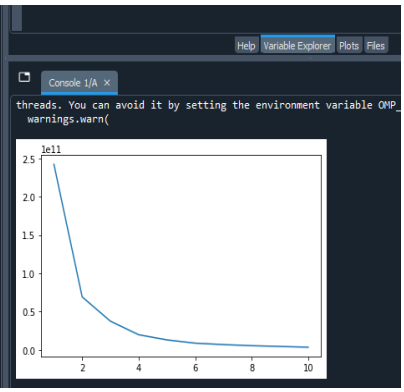
```python
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3,init='k-means++')
kmeans.fit(x)

print(kmeans.cluster_centers_)

sonuclar = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i , init='k-means++', random_state=40)
    kmeans.fit(x)
    sonuclar.append(kmeans.inertia_) # kmeans.inertia_  = wcss

plt.plot(range(1,11),sonuclar)
```



Hierarchical Clustering

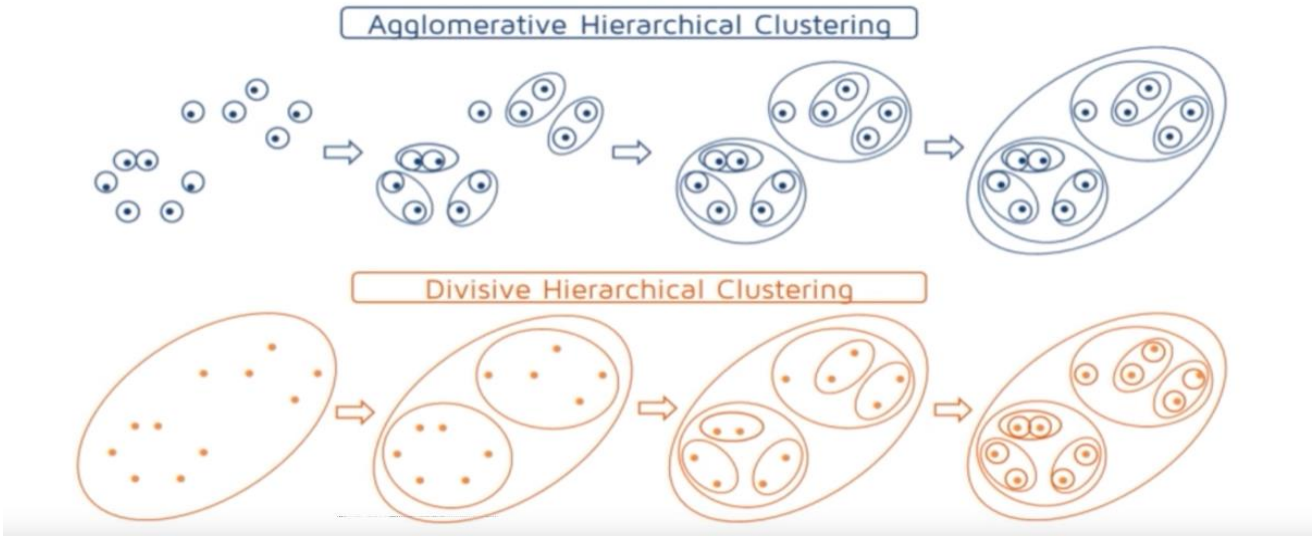
HİYERARŞİK BÖLÜTLEME/KÜMELEME

- Agglomerative **bottom-up agglomerative clustering**
- Divisive **top-to-bottom**

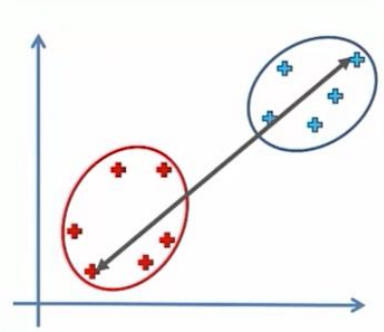
- Algoritma Adımları (Agglomerative)
  - Her veri tek bir küme / bölüt ile başlar
  - En yakın ikişer komşuyu alıp ikişerli küme/bölüt oluşturulur
  - En yakın iki kümeyi alıp yeni bir bölüt oluşturulur
  - Bir önceki adım, tek bir bölüt/küme olana kadar devam eder

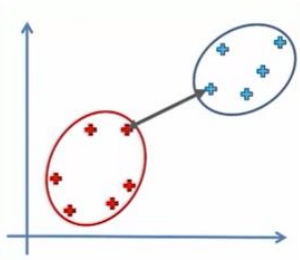
Ihtimal havuzu n/2 ile (n-1) arasında

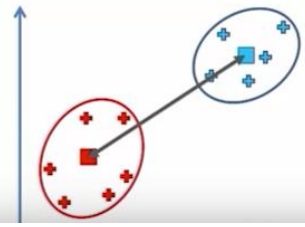
Hierarchical Clustering



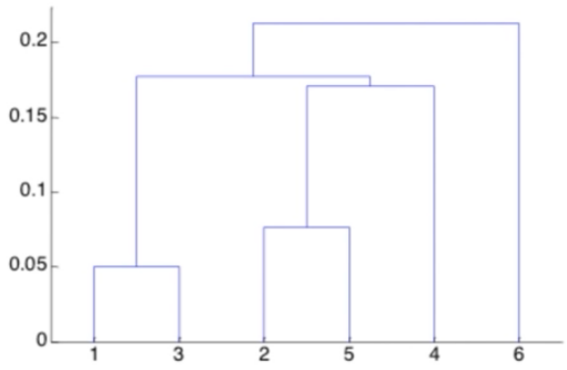
- Mesafe Ölçümü?
  - 1. Metrik Problemi :
    - Öklit Mesafesi
  - 2. Referanslar
    - En Yakın noktlar
    - En Uzak Noktalar
    - Ortalama
    - Merkezler arası mesafe

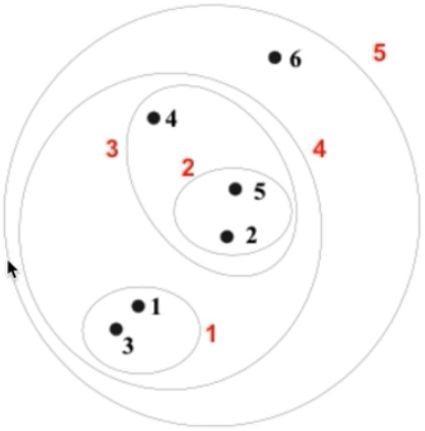


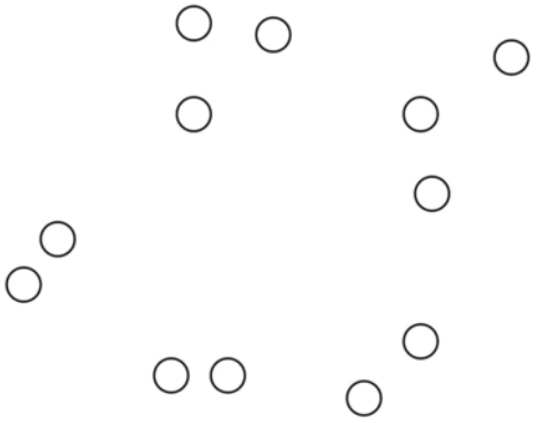




Dendogram

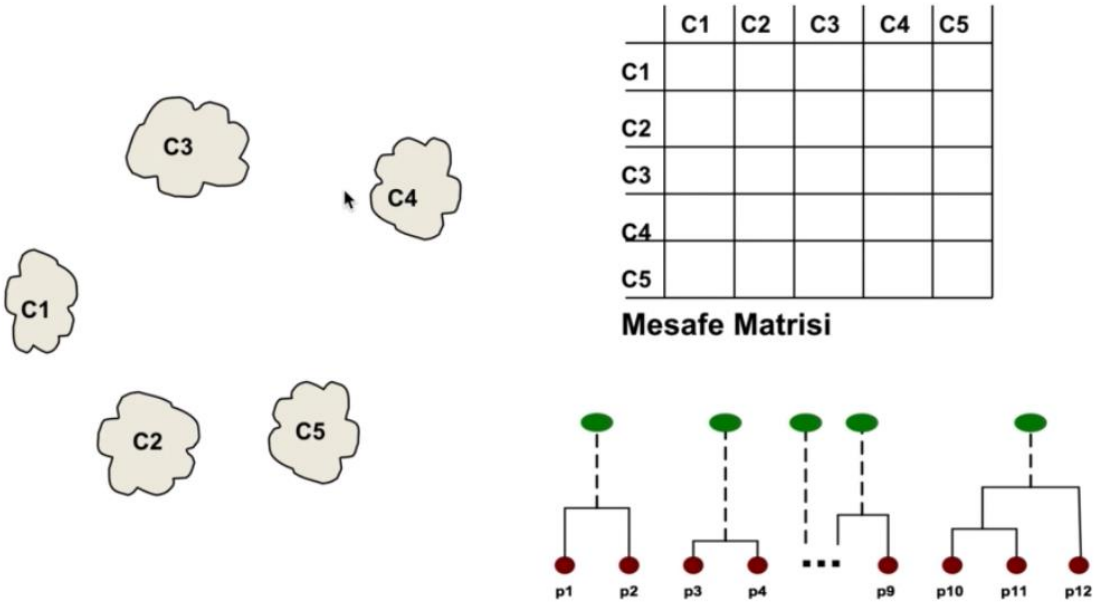




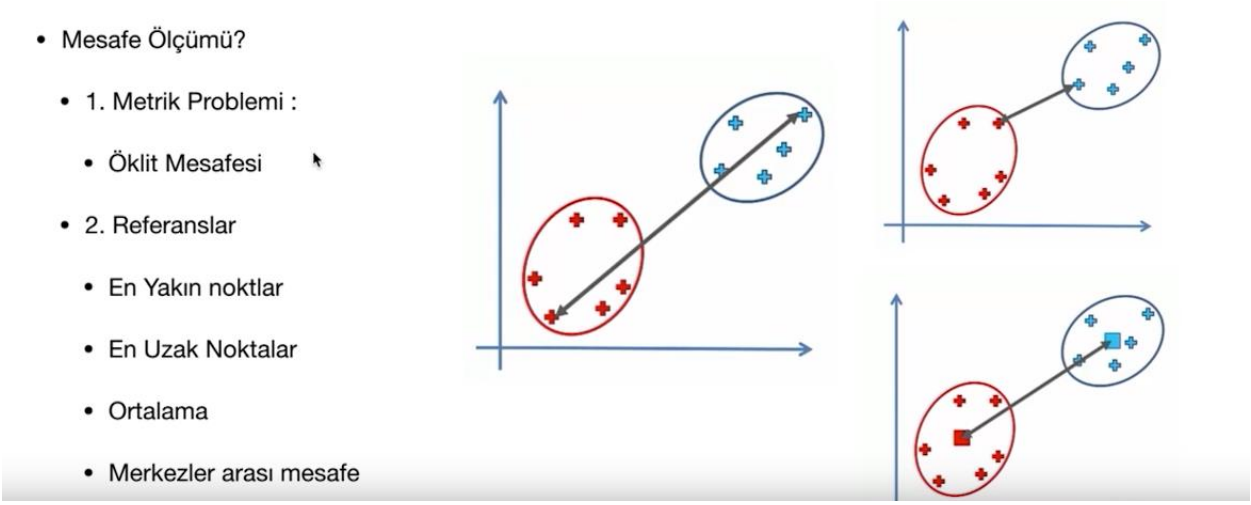


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

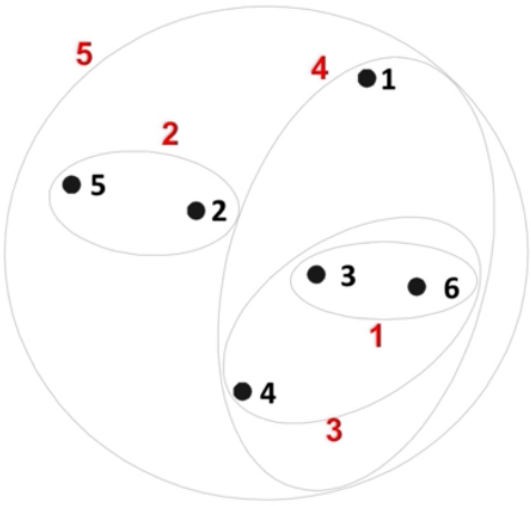
Mesafe Matrisi

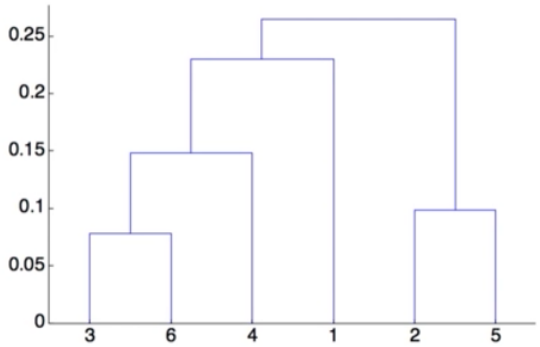


Yukarıdaki şekilde olduğu gibi birden fazla cluster yapısı oluşursa tekrar clusterleri birleştirmek için 4 farklı yöntem kullanılabilir[data pointler cluster yapılırken oklit ile bulunabilir]



Ortalama Mesafe

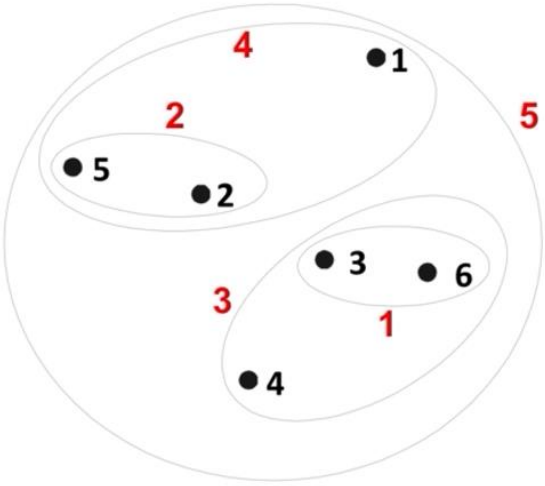


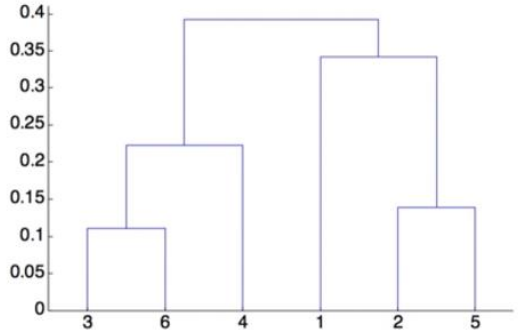


Nested Clusters

Dendrogram

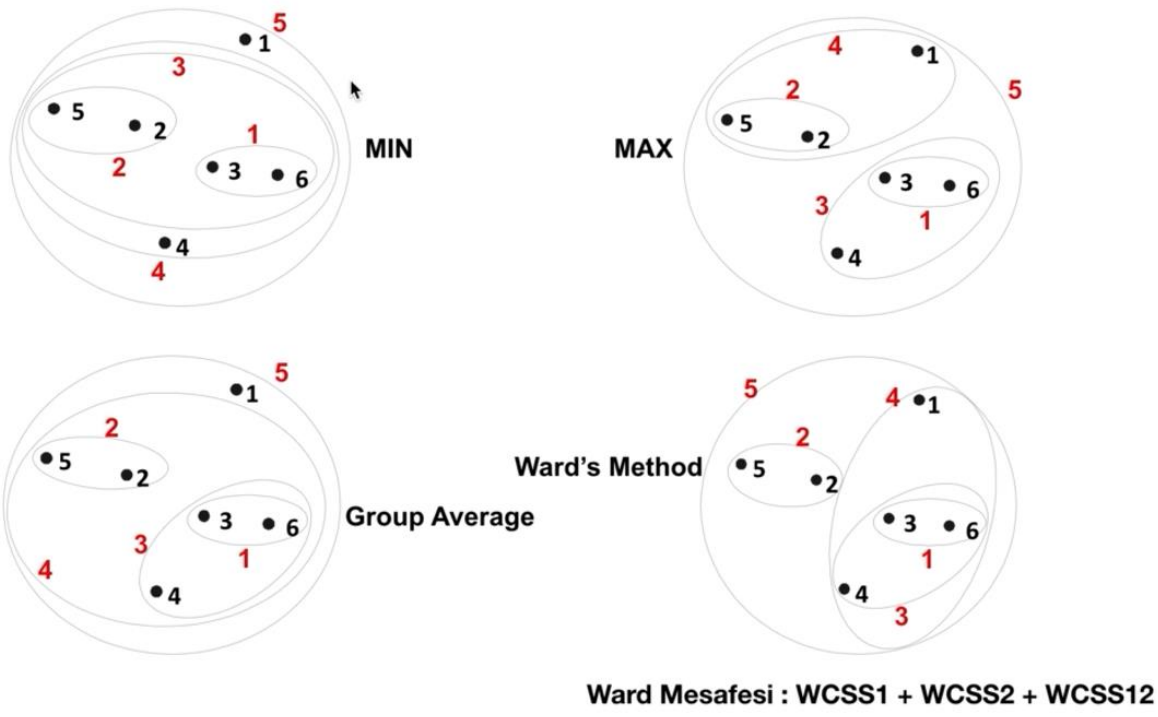
En Uzak Elemanların Mesafesi





Nested Clusters

Dendrogram



Ward's Method: clusterler arasındaki mesefaye göre işlem yapar

Ward mesafesi: Cluster1 ve Cluster2 arasında hesaplanmak istensin;
öncelikle cluster1 wcss değeri hesaplanır daha sonra cluster2 wcss değeri en sonunda bu iki clusterin birleşmiş halininn wcss değeri hesaplanır ve bu üç wcss değeri toplanır.

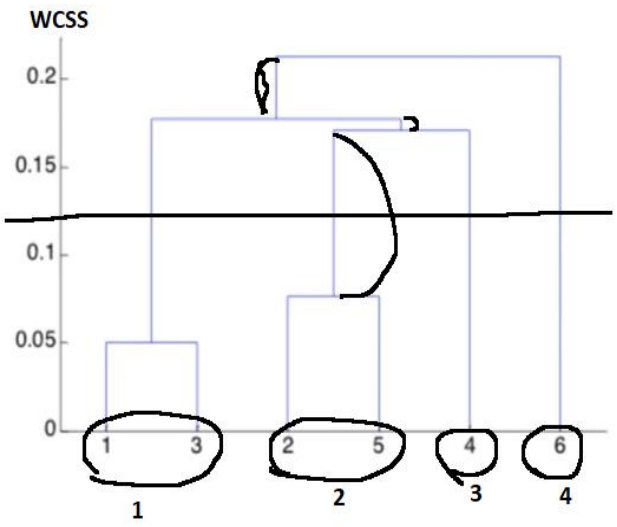
Bu iki cluster arasındaki mesafedir.

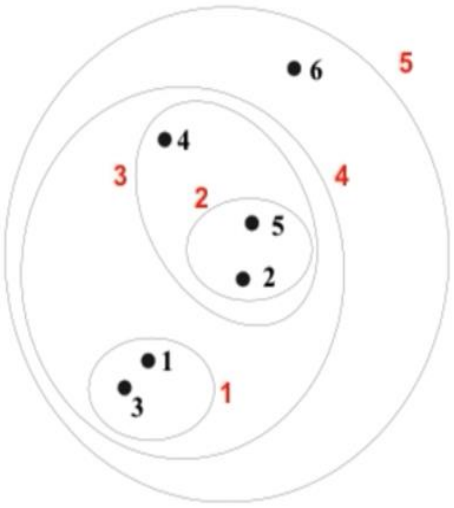
Ward Mesafesi : WCSS1 + WCSS2 + WCSS12

Hiyerarşik bölütlemede de k-means algoritmasında olduğu gibi k değerinin kaç olacağını

K-Wcss grafiğinde kırılan noktaya [dirsek..] bakarak bulabiliriz.

Dendograma Bakarak K sayısına cevap vermek;





Oluşturulan clusterların wcss değerleri arasındaki en büyük farktan bir çizgi çekilerek kac cluster'da[k değeri] en iyi sonucu vereceği anlaşilabilir