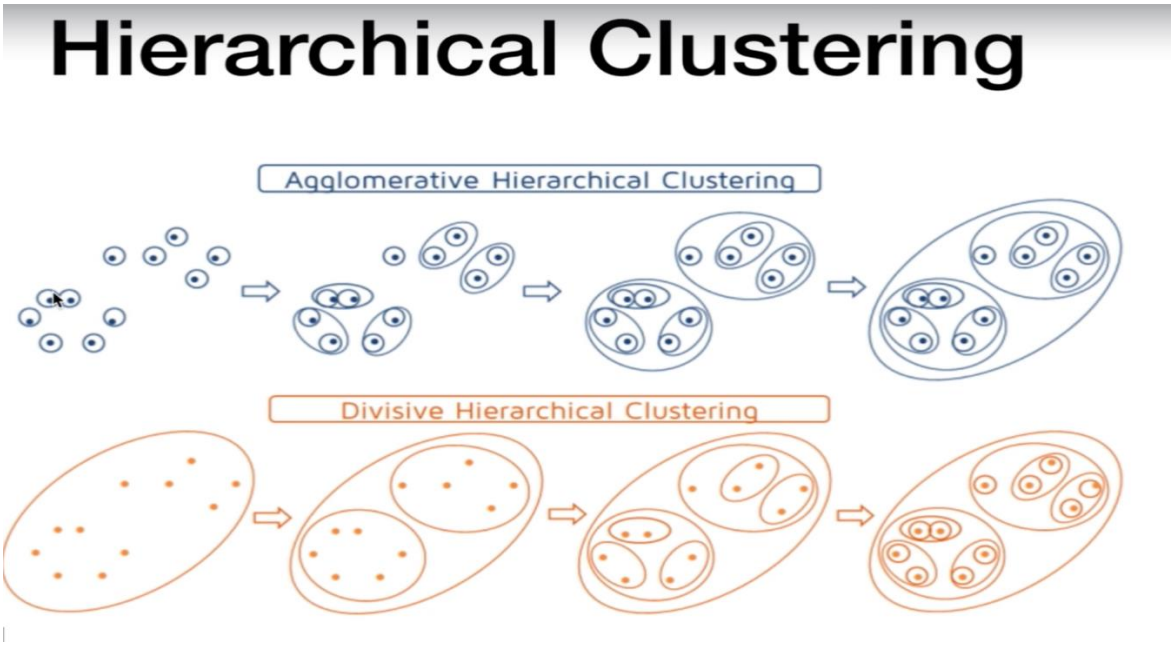
Hiyerarşik Kümeleme

- İki farklı çeşidi vardır: **Agglomerative(**Aşağıdan yukarıya**)** ve **Divisive(**Yukarıdan aşağıya**)**
- **Agglomerative Çalışması:**
  - Başlangıçta her veri tek bir küme olarak alınır.
  - En yakın ikişer komşuyu alıp ikişerli kümeler oluşturulur.
  - Aynı işlem tekrarlanarak en yakın iki küme ile yeni kümeler oluşturulur. Taa ki tüm datalar tek bir kümede birleşene kadar.

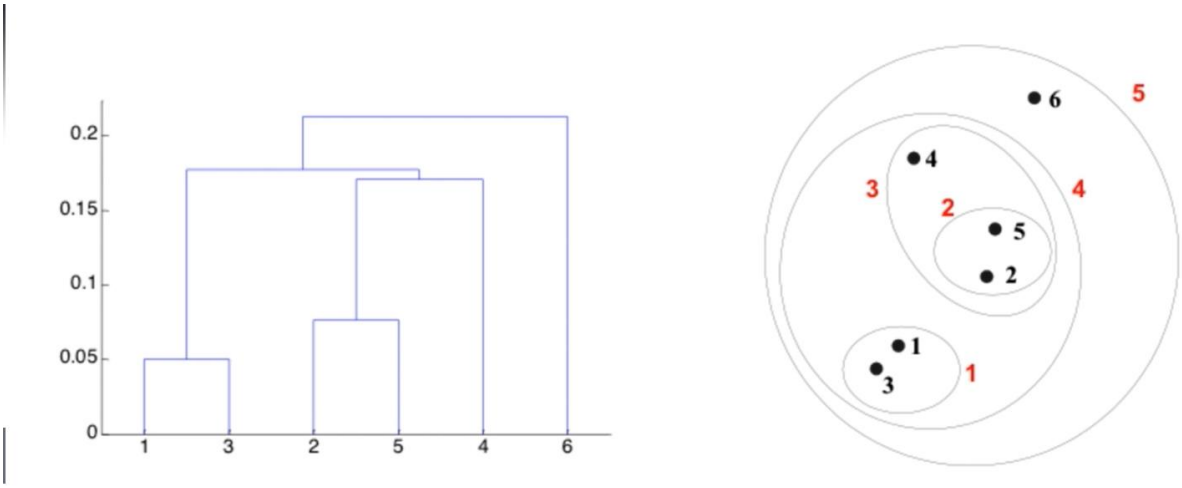


- Biz ders boyunca agglomerative yaklaşımında bahsediyor olacağız.
- Şimdi burada bazı sorunlar var mesela en yakın olan kümeleri gruplandırdığımızı söylüyoruz ama bu mesafeyi nasıl ölçeceğiz? Kümeler tek elemandan oluşuyorken bu sıkıntı değil ama bir kümede çok data olunca centeroidlerden mi ölçeceğiz en uzak datalarından mı en yakınlarından mı yoksa tüm elemanlar arası uzaklığın ortalamasını mı alalım? Bir çok seçenek ortaya çıkar.
- Ki tek data bile olduğunda farklı ölçüm metrikleri kullanılabiliyordu K-NN de bahsetmiştik.Genelde öklit kullanıyoruz.

- Bu gruplandırmaları yaptıktan sonra, K sayısına göre biz istediğimiz versiyonu seçiyoruz. K = 1 ise en büyük kümeyi alıyoruz, K=2 ise küçük olan 2'li yi gibi...

Dendogram ve Hiyerarşik Kümeleme

➔ Mesela aşağıdaki gibi siyah noktalardan oluşan bir data set olsun:

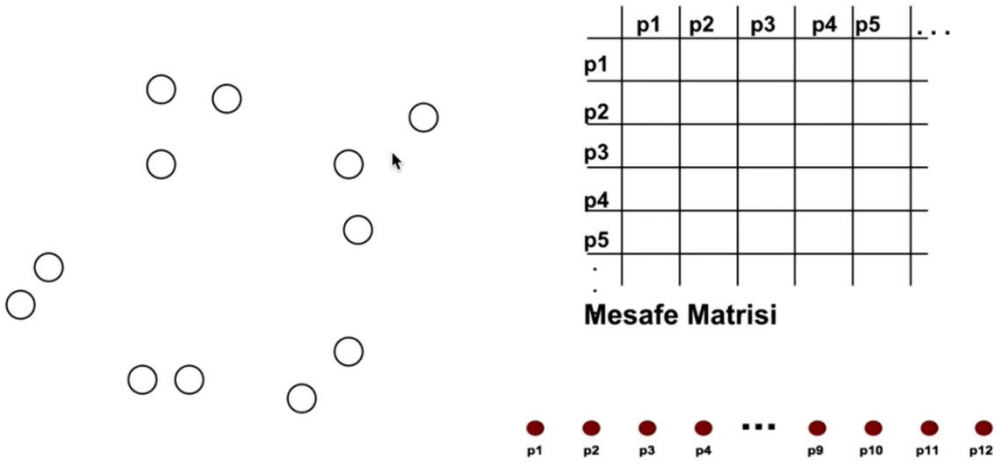


➔ Önce en yakın 2 komşu birleştirilir, bu bizim için 1 ve 3 noktaları. Dendogramda da bu birleşim görülebilir. Dendogramın y eksenini birleşimler arasındaki mesafe gibi düşünebiliriz. Görüyoruz ki en yakın birleşim 1 ve 3 arasında gerçekleşmiş.

➔ Daha sonra 2 ile 5 birleşmiş bunlar arası mesafe 0.05'ten büyük neredeyse 0.1.

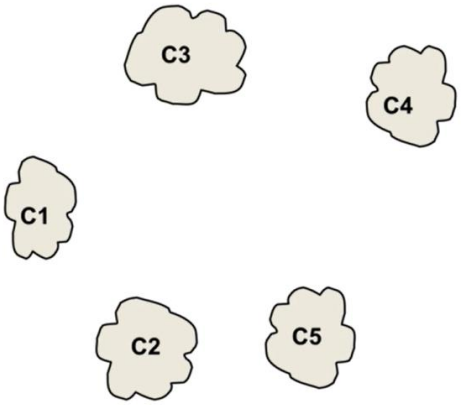
➔ Sonuç olarak soldaki dendogram bu birleşimleri gösteriyor.

➔ Tabi biz burada 1 ve 3 ün en yakın olduğunu gözle gördük ancak bilgisayar bunu direkt anlayamıyor tabi tek tek tüm noktalar arası mesafe ölçümlerini yapmalı:



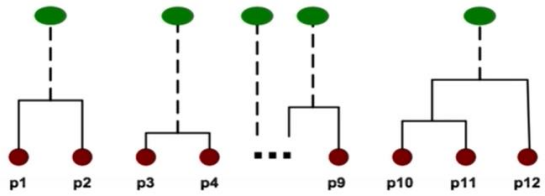
➔ Bunun için böyle bir mesafe matrisi ortaya çıkıyor. Ve buna göre clustering işlemi yapılıyor.

➔ Diyelim ki clustering yapıldı:

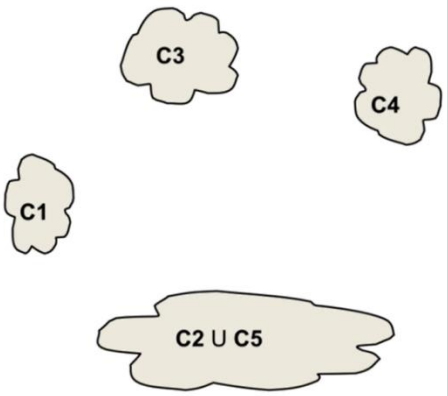


	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

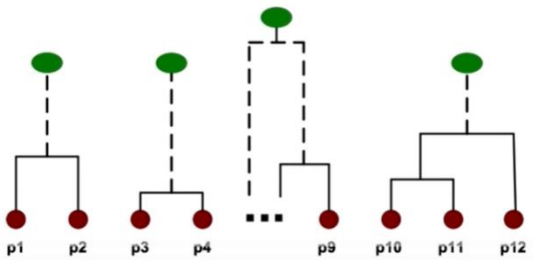
Mesafe Matrisi



➔ Şimdi artık tek noktalar arası clusteringden çıktık clusterlar arası clustering adımına girdik. Böyle olunca mesafe matrisinde iki sütun birleşip yeni bir sütun oluşturacak. Mesela C2 ve C5 birleşti diyelim artık ikisi için ayrı mesafe tutmayacağız yeni C2C5 için tek bir sütunda mesafe tutarız.

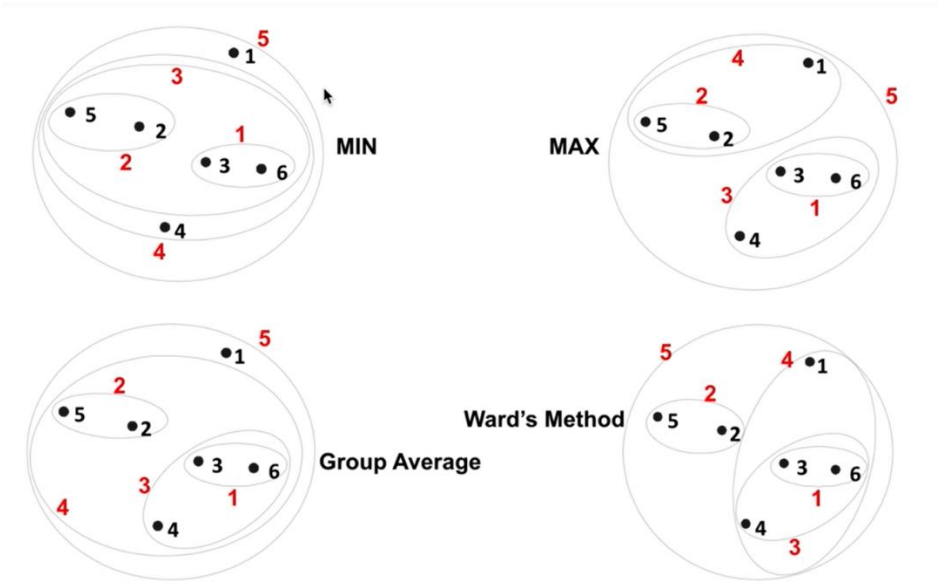


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		



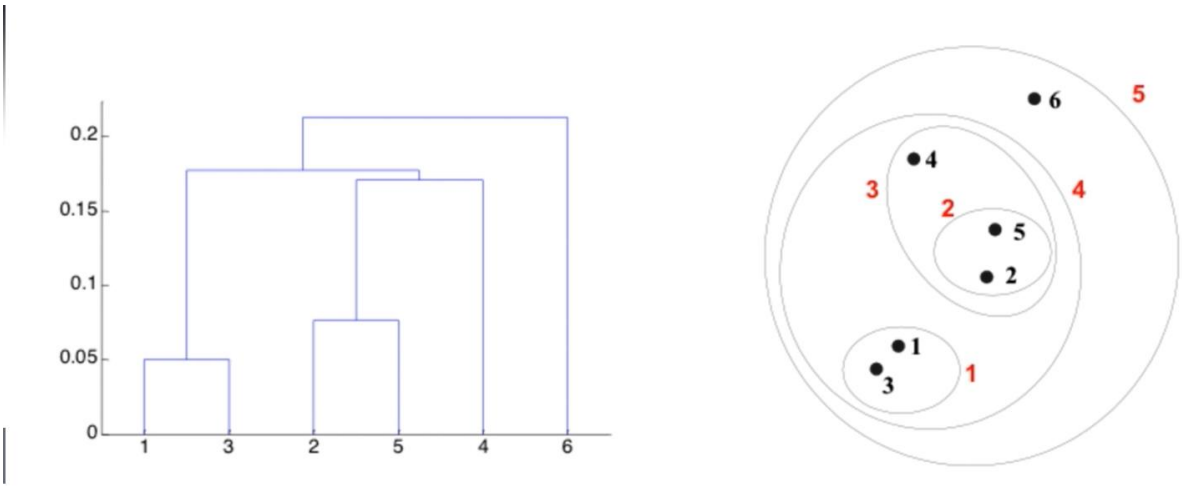
➔ Daha önce de bahsettiğimiz gibi clusterlar arası mesafeleri ölçmek için farklı metrics var. Ortalama mesafe, min mesafe, max mesafe, centroid mesafesi vb. ve her bir method farklı bir clustering ile sonuçlanır! Bizim de kullanacağımız bir hesaplama yöntemi ise Ward distance yöntemi. Bu aslında andrew'in K-mean clustering için anlattığı J() methodunu içeriyor. Yani mesela C1 ve C3 arası mesafeye bakıyoruz: C1'in elemanlarının C1 merkezine uzaklıklarının toplamı + C2 elemanlarının C2 merkezine uzaklıkları toplamı + C1 ve C3 tek grup olsaydı bu grubun elemanlarının

bu gurubun centroidine olan uzaklıkları toplamı. Bu da bir distance hesaplama yöntemi!



➔ Hangi gruplandırmanın daha iyi olduğu duruma göre değiştir.

➔ Dendogram oluştuktan sonra istenilen sayıda gruba ayırma işlemi de dendogram yardımıyla yapılabilir:



➔ Mesela en 0.2 seviyesinden kesersek 2 class elde edeceğimizi görüyoruz. Bunu dendogram üzerinden anlayabiliriz. Bize kaç class lazımsa ona göre dendogram üzerinden nerede keseceğimizi bulabiliriz.

➔ Peki K kaç seçilmeli, elbow'a benzer bir method var mı? Var. Yine dendogram üzerinden bakılır max distance sağlayan yani en uzun bağlantılardan kesilmesi daha iyi olabilir. Bu örnek için 0.1 civarından kesersek elimizde 1-3, 2-5, 4, 6 şeklinde 4 cluster kalacağı görülür.