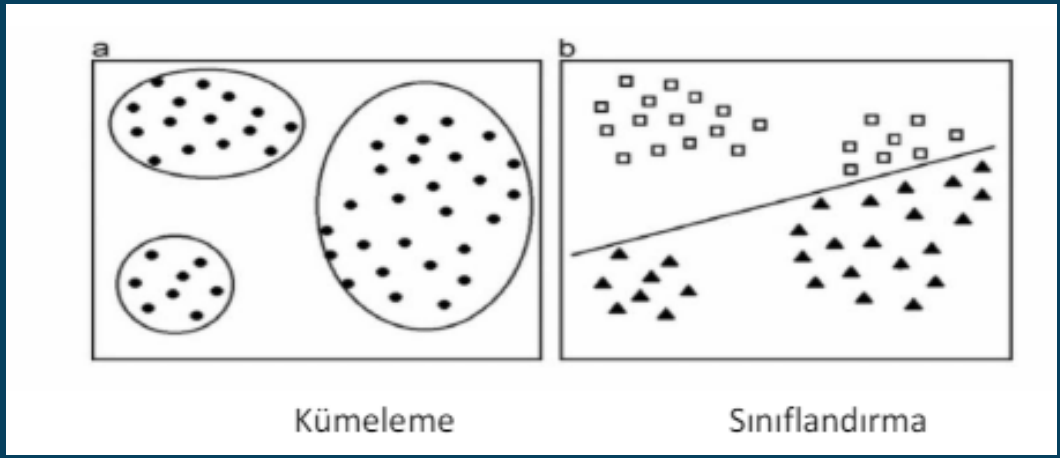
Kümeleme (Clustering)

Kümeleme (Clustering) :

- Kümeleme, veri analizi için sıkça kullanılan bir istatistiksel tekniktir ve denetimsiz öğrenmenin bir yöntemidir. Denetimsiz öğrenme, veri kümesi ile çıktıların olmadığı bir öğrenme metodudur. Veri kümesindeki verileri yorumlayarak ortak noktaları bulmak ve bunları kümeleştirme işlemi yapılarak anlamlı bir veri elde edebilmektir. Denetimsiz öğrenme sürecinde, sadece veriler mevcuttur ve bu verilerden sonuçlar çıkarılmaya çalışılır. Ancak, verilerin doğru olduğu veya sonuçların kesin doğruluğu garantisi yoktur çünkü başlangıçta verilere ilişkin herhangi bir bilgi sağlanmaz. Değişkenler arasındaki ilişkilere dayalı olarak veriyi kümeleyerek çeşitli modeller ve yapılar oluşturulabilir. Örneğin, bir alışveriş sitesinde bir ürün satın alındığında, kullanıcılara diğer alınabilecek ürünler önerilebilir. Benzer şekilde, bir hizmet satın alındığında, bu hizmetle ilişkili diğer hizmetler müşterinin ilgi alanına girebilir.



- Denetimsiz öğrenmede, kümeleme oldukça önemlidir. Kümeleme algoritmaları basitçe, bir veri kümesindeki öğeleri kendi aralarında gruplamaya çalışır. Burada kaç grup olacağı veya en uygun küme sayısını algoritma kendisi belirler. Verilerin benzerlik, yakınlık veya uzaklık gibi ölçütlerle analiz edilerek sınıflara ayrılmasına kümeleme denir. Bir örnekle açıklamak gerekirse, alışveriş yapılan bir markette kasiyerin bir robot olduğunu düşünelim ve tüm ürünlerin birbirine karıştığını hayal edelim. Robot elmayı bulur, tanır ve diğer ürünleri yığın içerisinde toplamaya başlar. Seçme işlemi devam ettikçe robot yetenek kazanır ve performansını artırabilir; hatalı seçimler yaparsa bunları ayıklayabilir. Bu şekilde, ürünler arasındaki benzerlik azalırken ayrım yapma yeteneği artar. Böylece, ürünler sınıflandırılmış olur.

Kümelemenin uygulama alanları:

- Tıp’da elde edilen görüntülemeler üzerindeki farklıları analiz edilerek değişik nitelikler çıkartılabilir. Suç Yerlerinin Belirlenmesi: Bir şehirdeki belirli bölgelerde mevcut olan suçlarla ilgili veriler,
- suç kategorisi, suç alanı ve ikisi arasındaki ilişki, bir şehirdeki ya da bölgedeki suça eğilimli alanlara ilişkin kaliteli bilgiler verebilir. Oyuncu istatistiklerini analiz etmek:
- Oyuncu istatistiklerini analiz etmek, spor dünyasının her zaman kritik bir unsuru olmuştur ve artan rekabetle birlikte, makine öğrenmenin burada oynayacağı kritik bir rol vardır.
- Çağrı Kaydı Detay Analizi: Bir çağrı detay kaydı (CDR), telekom şirketleri tarafından bir müşterinin araması, SMS ve internet etkinliği sırasında elde edilen bilgilerdir. Bu bilgiler, müşteri demografisiyle birlikte kullanıldığında, müşterinin ihtiyaçları hakkında daha fazla bilgi sağlar.

Kümeleme Çeşitleri:

- Hiyerarşik Kümeleme
- Gürültülü Uygulamaların Yoğunluğa Dayalı Konumsal Kümelenmesi (DBSCAN) (DBSCAN)
- K-means Kümeleme
- Ağırlık Ortalama Kaydırma Kümelemesi
- Gauss Karışım Modelleri (GMM) kullanarak Beklenti-Maksimizasyon (EM) Kümeleme

Hiyerarşik Kümeleme:

- Hiyerarşik kümeleme algoritmaları 2 kategoriye ayrılır: yukarıdan aşağıya veya aşağıdan yukarıya. Aşağıdan yukarıya algoritmalar, her veri noktasını başlangıçta tek bir küme olarak ele alır ve ardından tüm kümeler tüm veri noktalarını içeren tek bir kümede birleştirilene kadar küme çiftlerini art arda birleştirir (veya toplar). Bu nedenle aşağıdan yukarıya hiyerarşik kümeleme, hiyerarşik kümelemeli kümeleme (hierarchical agglomerative clustering) veya HAC olarak adlandırılır. Bu küme hiyerarşisi bir ağaç (veya dendrogram) olarak temsil edilir. Ağacın kökü, tüm örnekleri toplayan benzersiz kümedir, yapraklar yalnızca bir örnek içeren kümelerdir.

- Hiyerarşik kümeleme, her veri noktasını tek bir küme olarak başlatarak işe başlar; yani, eğer veri kümemizde X veri noktası varsa, o zaman başlangıçta X adet kümemiz olur. Daha sonra, iki küme arasındaki mesafeyi ölçen bir mesafe ölçüsü seçeriz. Örneğin, iki küme arasındaki mesafeyi, birinci kümedeki veri noktaları ile ikinci kümedeki veri noktaları arasındaki ortalama mesafe olarak tanımlayan ortalama bağlantıyı kullanabiliriz.
- Her yinelemede, en küçük ortalama bağlantıya sahip iki küme birleştirilir. Yani, seçtiğimiz uzaklık ölçütüne göre, birbirleri arasındaki en küçük mesafeye sahip olan iki küme birleştirilir çünkü en benzer olanlardır ve bu nedenle birleştirilmeleri gereklidir.
- Bu adım, ağacın köküne ulaşana kadar tekrarlanır, yani tüm veri noktalarını içeren tek bir küme elde edilir. Bu şekilde, sonunda kaç küme istediğimizi seçebiliriz, yani sadece kümeleme işlemini ne zaman durduracağımızı belirleyerek.
- Hiyerarşik kümeleme, küme sayısını belirlememizi gerektirmez ve oluşturduğumuz ağaç yapısı sayesinde en uygun küme sayısını seçebiliriz. Ayrıca, algoritma mesafe ölçüsü seçimine duyarlı değildir; diğer kümeleme algoritmalarında olduğu gibi uzaklık ölçüsü seçimi kritik değildir. Hiyerarşik kümeleme, verilerin hiyerarşik bir yapıya sahip olduğu ve bu hiyerarşiyi korumak istediğiniz durumlarda özellikle kullanışlıdır; diğer kümeleme algoritmaları bu tür yapıları koruyamaz. Ancak, bu avantajlar, $O(n^3)$ zaman karmaşıklığına sahip olması nedeniyle K-Ortalamaları ve GMM'ye göre daha düşük bir verimlilikle gelir.

K-Means Algoritması:

- K-means algoritmasında kullanılan örneklem, k adet kümeye bölünür. Algoritmanın özü birbirlerine benzerlik gösteren verilerin aynı küme içerisine alınmasına dayanır. Algoritmadaki benzerlik terimi, veriler arasıdaki uzaklığa göre belirlenmektedir. Uzaklığın az olması benzerliğin yüksek, çok olması ise düşük olduğu anlamına gelmektedir. K-means algoritmasının yapısı aşağıdaki gibidir;
- 1) K adet rastgele küme oluştur
- 2) Kare hata oranını hesapla
- 3) Verilerin kümelerin orta noktalarına olan uzaklıklarını bul
- 4) Her veri için en yakın kümeyi, o verinin kümesi olarak belirle
- 5) Yeni yerleşim düzenine göre hata oranını hesapla
- 6) Eğer önceki hata oranı ile şimdiki hata oranı eşit değilse 2,3,4,5 ve 6. adımları tekrarla
- 7) Eğer önceki hata oranı ile şimdiki hata oranı eşitse kümeleme işlemini sonlandır Dirsek yöntemi, kümelere nasıl ihtiyaç duyacağımız konusunda kullanışlı olur. Dirsek noktasının belirlenmesi gerekir.

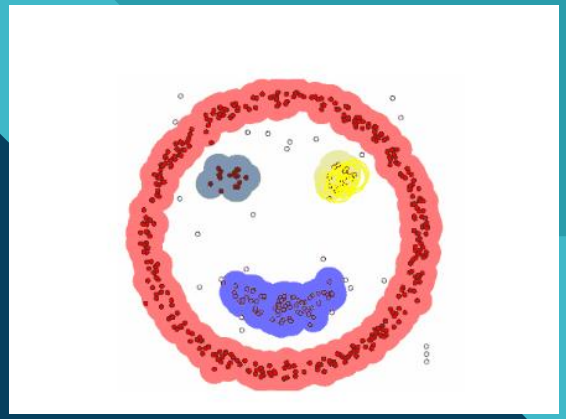
Ağırlık Ortalama Kaydırma Kümelemesi:

- Ortalama kaydırma kümeleme, veri noktalarının yoğun alanlarını bulmaya çalışan kayan pencere tabanlı bir algoritmadır. Centroid tabanlı bir algoritmadır, yani amacın her bir grubun / sınıfın merkez noktalarını bulmaktır, bu da kayan pencere içindeki noktaların ortalaması olacak merkez noktaları için adayları güncelleyerek çalışır. Bu aday pencereler daha sonra, neredeyse kopyaları ortadan kaldırmak için bir işlem sonrası aşamasında filtrelenir ve nihai merkez noktaları ve bunlara karşılık gelen grupları oluşturur. Sürgülü pencerelerin tümü ile uçtan uca tüm sürecin bir örneği aşağıda gösterilmiştir. Her siyah nokta, kayan bir pencerenin merkezini temsil eder ve her gri nokta bir veri noktasıdır.

Ortalama kayma, bir tepe tırmanma algoritması olarak adlandırılan bir yöntemdir ve bir dizi noktanın yoğunluğunu analiz etmek için kullanılır. İki boyutlu bir uzayda çalışırken, bir C noktasında (rastgele seçilmiş bir nokta) ortalanmış ve bir çekirdek olarak r yarıçapına sahip dairesel bir kayan pencere ile başlarız.
Her yinelemede, kayan pencere merkez noktası pencere içindeki noktaların ortalamasına kaydırılarak daha yoğun bir bölgeye doğru hareket ettirilir. Sürgülü pencere içindeki yoğunluk, pencere içindeki noktaların sayısına bağlıdır. Bu şekilde, penceredeki noktaların ortalamasına doğru kayarak, daha yüksek nokta yoğunluğuna sahip alanlara doğru ilerleriz.
Pencerede daha fazla nokta barındırabilecek bir yön olmayana kadar, kayan pencereyi ortalamaya göre kaydırmaya devam ederiz. Bu adım süreci, tüm noktalar bir pencerenin içinde kalana kadar birçok sürgülü pencerede tekrarlanır. Birden çok sürgülü pencere örtüştüğünde, en çok noktayı içeren pencere korunur. Veri noktaları daha sonra bulundukları kayan pencereye göre kümelenir.

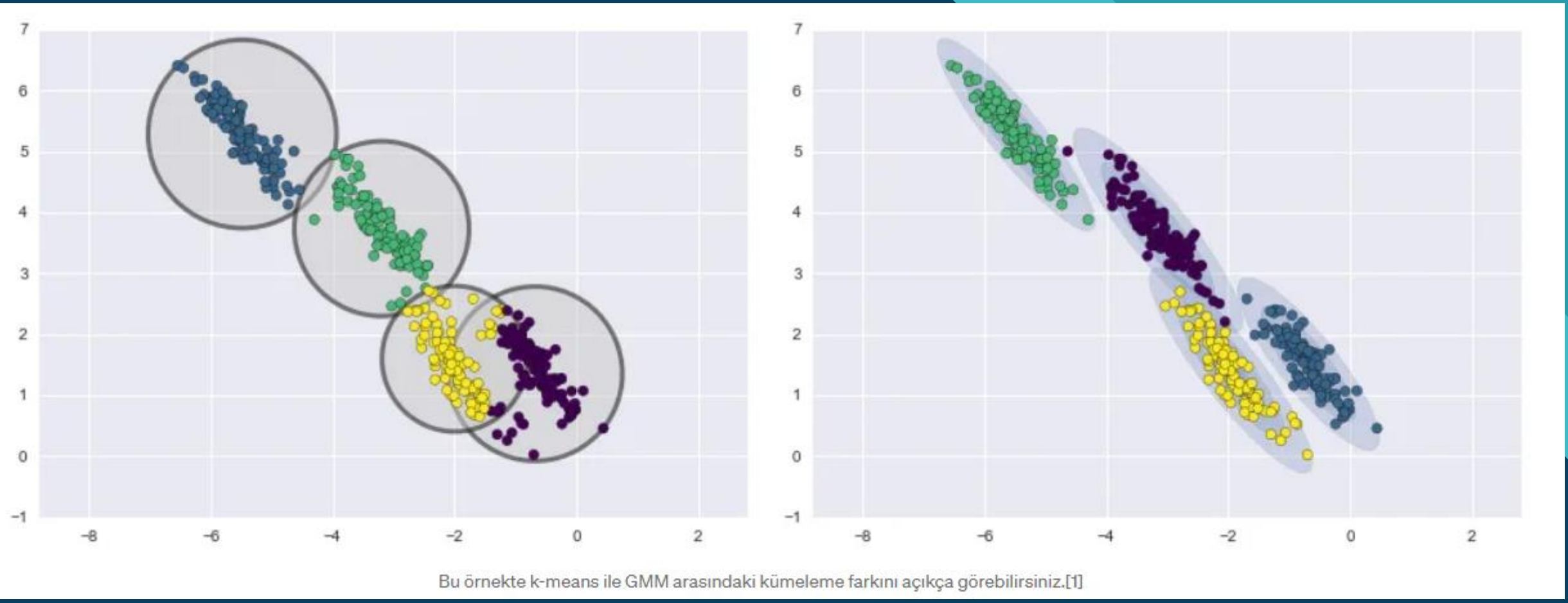
Gürültülü Uygulamaların Yoğunluğa Dayalı Konumsal Kümelenmesi (DBSCAN):

- DBSCAN, ortalama kaymaya ekseninde, benzer ve yoğunluklu bölgeleri kümeleyen bir algoritmadır, ancak birkaç önemli avantajı vardır. Minimum bölge sayısında ve uzaklıkta maksimum yoğunluk bölgesi oluşturulması hedeflenir.
- DBSCAN, keyfi bir başlangıç veri noktasıyla başlar ve bu noktanın komşuluğu bir epsilon ε uzaklığı kullanılarak belirlenir (ε mesafesi içindeki tüm noktalar komşuluk noktalarıdır).
- Eğer bu mahalle içinde yeterli sayıda nokta (minPoints'e göre) varsa, kümeleme işlemi başlar ve mevcut veri noktası yeni kümedeki ilk nokta olur. Aksi takdirde, nokta gürültü olarak etiketlenir (ancak daha sonra bu gürültülü nokta kümenin bir parçası haline gelebilir). Her iki durumda da, bu nokta "ziyaret edildi" olarak işaretlenir.
- Yeni kümedeki bu ilk nokta için, ε mesafesi komşuluğundaki noktalar da aynı kümenin parçası olur. ε mahallesindeki tüm noktaları aynı kümeye ait yapma prosedürü, küme grubuna yeni eklenen tüm yeni noktalar için tekrarlanır.
- Bu süreç, kümedeki tüm noktalar belirlenene kadar, yani kümenin ε mahallesindeki tüm noktalar ziyaret edilip etiketlenene kadar tekrarlanır. Mevcut kümeyle işimiz bittiğinde, yeni bir ziyaret edilmemiş nokta alınır ve işlenir, bu da başka bir küme veya gürültü keşfine yol açar. Bu işlem, tüm noktalar ziyaret edildi olarak işaretlenene kadar tekrarlanır.



Gauss Karışım Modelleri (GMM) kullanarak Beklenti-Maksimizasyon (EM) Kümeleme:

- K-Ortalamaların en büyük dezavantajlarından biri, küme merkezi için ortalama değerin naif kullanımıdır. Sol taraftaki resimde, aynı ortalamaya merkezlenmiş farklı yarıçaplara sahip iki dairesel küme olduğunu görebiliriz. Bu durumda, K-Ortalamaların başarısız olduğunu çünkü küme merkezlerinin birbirine çok yakın olduğunu ve kümelerin dairesel olmadığı durumlarda etkili olamadığını anlayabiliriz.
- Gauss Karışım Modelleri (GMM'ler), K-Ortalamalardan daha fazla esneklik sağlar. GMM'ler, veri noktalarının Gauss olarak dağıtıldığını varsayarlar; bu, ortalamayı kullanarak dairesellikten daha az kısıtlayıcı bir varsayımdır. Bu şekilde, kümelerin şeklini açıklamak için ortalama ve standart sapma gibi iki parametremiz olur. Örneğin, iki boyutlu bir örnekte, her Gauss dağılımı tek bir kümeye atanabilir, böylece her türlü eliptik şekil alabilirler.
- Gauss'un parametrelerini (örneğin, ortalama ve standart sapma) bulmak için Beklenti-Maksimizasyon (EM) adı verilen bir optimizasyon algoritması kullanılır. Bu şekilde, kümelere uydurulan Gauss dağılımları elde edilir. GMM'leri kullanarak Beklenti-Maksimizasyon kümeleme sürecine geçebiliriz.



Bu örnekte k-means ile GMM arasındaki kümeleme farkını açıkça görebilirsiniz.[1]