ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
**UZAKTAN EĞİTİM UYGULAMA VE ARAŞTIRMA MERKEZİ**

# YAPAY ZEKÂ TEKNOLOJİLERİ VE YÖNETİMİ

www.hacibayram.edu.tr/uzem

# 6. Denetimsiz Öğrenme

## 6.1. Denetimsiz Öğrenme Kavramı

Denetimsiz öğrenme, makine öğrenmesi alanında, etiketlenmemiş verilerle öğrenme sürecidir. Bu yöntemde, veriler üzerinde herhangi bir hedef değişken veya sınıf etiketi bulunmaz; bu nedenle model, veri setindeki yapıları, örüntüleri ve ilişkileri keşfetmeye çalışır. Denetimsiz öğrenmenin temel amacı, veriler arasındaki benzerlikleri veya farklılıkları belirlemek ve bu bilgileri kullanarak veri setini organize etmektir. Örneğin, denetimsiz öğrenme algoritmaları kullanılarak müşteri segmentasyonu yapılabilir; bu durumda, müşteriler benzer özelliklerine göre gruplara ayrılır.

Denetimsiz öğrenme teknikleri arasında kümeleme (clustering) ve boyut indirgeme (dimensionality reduction) gibi yöntemler bulunur.

## 6.2. Kümeleme

Kümeleme (clustering), denetimsiz öğrenme tekniklerinden biri olup, verileri benzerliklerine göre gruplara ayırma sürecidir. Kümeleme, etiketlenmemiş veriler üzerinde çalışarak, veri seti içindeki doğal grupları veya kümeleri belirlemeye çalışır. Her küme, içindeki veri noktalarının birbirine daha yakın olduğu, ancak diğer kümelerdeki veri noktalarından belirgin şekilde farklı olduğu bir grup olarak tanımlanır. Bu teknik, müşteri segmentasyonu, pazarlama stratejileri geliştirme, biyoinformatik analizleri ve sosyal ağ analizleri gibi birçok alanda yaygın olarak kullanılır. En yaygın kümeleme algoritmalarından bazıları K-means, Hiyerarşik Kümeleme ve DBSCAN'dir.

### 6.2.1 K Means

K-means, en popüler kümeleme algoritmalarından biri olup, verileri belirli sayıda kümeye ayırarak her kümede benzer özelliklere sahip veri noktalarını bir araya getirmeyi amaçlar. Bu algoritma, her bir kümenin merkezi olarak tanımlanan "merkez" noktaları etrafında kümeler oluşturur. K-means algoritmasının temel amacı, her bir veri noktasını en yakın merkez noktaya atayarak, kümeler arasındaki içsel farklılıkları minimize etmek ve her bir kümenin içindeki veri noktalarının birbirine olabildiğince benzer olmasını sağlamaktır.

K-means algoritması birkaç adımdan oluşur. İlk olarak, kullanıcı tarafından belirlenen K sayısı kadar merkez noktası rastgele seçilir. Daha sonra, her veri noktası en yakın merkez noktasına atanır ve bu şekilde geçici kümeler oluşturulur. Bu adım

tamamlandıktan sonra, her kümenin merkezi, o kümeye ait veri noktalarının ortalaması alınarak yeniden hesaplanır. Bu süreç, kümelerin merkezleri stabil hale gelene kadar veya belirli bir iterasyon sayısına ulaşılana kadar tekrarlanır. Sonuç olarak, K-means algoritması, veri setini K adet kümeye ayırarak her bir kümenin merkezi etrafında yoğunlaşan veri noktalarını belirler.

K-means algoritmasının en büyük avantajlarından biri, basitliği ve hızlı bir şekilde sonuç vermesidir. Ancak, algoritmanın performansı ve sonuçları, başta belirlenen K değeri ve başlangıçta seçilen merkez noktalarına oldukça bağımlıdır. Yanlış K değeri veya kötü başlangıç noktaları, zayıf kümeleme sonuçlarına yol açabilir. Ayrıca, K-means, küme şekillerinin dairesel olması gerektiğini varsayar; bu nedenle daha karmaşık şekillerdeki kümeleri iyi tanımlayamayabilir. Bu sınırlamaların üstesinden gelmek için genellikle K-means++ gibi varyantlar veya diğer kümeleme algoritmaları tercih edilebilir.

K-means, geniş bir uygulama yelpazesinde kullanılmaktadır. Örneğin, müşteri segmentasyonu, pazarlama stratejileri geliştirme, görüntü sıkıştırma ve biyolojik veri analizi gibi alanlarda yaygın olarak tercih edilir. Bu esneklik, K-means algoritmasının veriler arasındaki gizli yapıları keşfetmede etkili bir araç olmasını sağlar. Ancak, her veri seti ve uygulama için doğru K değerini seçmek ve algoritmanın sınırlamalarını göz önünde bulundurmak, başarılı sonuçlar elde etmek için kritiktir

### 6.2.2 Hiyerarşik Kümeleme

Hiyerarşik kümeleme, verileri hiyerarşik bir yapı içerisinde gruplayan ve bu yapıyı bir ağaç veya dendrogram olarak görselleştiren bir kümeleme yöntemidir. Hiyerarşik kümeleme, iki ana yaklaşımdan biri olan yığmacı (agglomerative) veya bölümsel (divisive) stratejilerle gerçekleştirilebilir. Yığmacı yöntem, başlangıçta her bir veri noktasını ayrı bir küme olarak kabul eder ve benzer kümeleri iteratif olarak birleştirir. Bu süreç, tüm veri noktaları tek bir küme oluşturana kadar devam eder. Bölümsel yöntem ise tam tersine, tüm veri noktalarını tek bir küme olarak başlatır ve iteratif olarak en benzer olmayan kümeleri ayırır.

Hiyerarşik kümelemenin en büyük avantajlarından biri, küme sayısını önceden belirlemeye gerek olmamasıdır. Dendrogram adı verilen hiyerarşik ağaç yapısı, veri seti içerisindeki kümeleme yapısını farklı seviyelerde gösterir ve kullanıcının istediği seviyede kesme yaparak uygun sayıda kümeyi seçmesine olanak tanır. Bu, verilerdeki doğal yapıyı ve kümeleme hiyerarşisini daha iyi anlamayı sağlar. Hiyerarşik kümeleme, özellikle farklı

boyutlardaki veya şekillerdeki kümelerin var olduğu veri setlerinde, daha esnek ve anlamlı sonuçlar verebilir.

Ancak, hiyerarşik kümelemenin bazı dezavantajları da vardır. Özellikle büyük veri setlerinde hesaplama açısından maliyetli olabilir, çünkü algoritma her bir adımda en yakın çiftleri bulmak için veriyi sürekli karşılaştırır. Ayrıca, birleştirme veya bölme işlemleri geri alınamaz, bu da bazen yanlış birleştirmelere veya bölmelere yol açabilir. Yine de, hiyerarşik kümeleme, karmaşık veri setlerinde doğal grupların belirlenmesinde ve veri içindeki hiyerarşik yapıları keşfetmede güçlü bir araç olarak kabul edilir.

### 6.2.3 Gauss Karışım Modeli

Gauss Karışım Modeli (GMM), bir veri setinin farklı normal (Gauss) dağılımlarının karışımı olarak modellendiği istatistiksel bir yöntemdir. GMM, verilerin birden fazla alt gruba ait olduğunu ve bu alt grupların her birinin Gauss dağılımı (normal dağılım) ile temsil edilebileceğini varsayar. Bu model, her bir alt grubun (veya bileşenin) kendi ortalama ve varyansına sahip olduğunu ve veri setindeki tüm veri noktalarının bu bileşenlerin bir kombinasyonu olduğunu kabul eder. GMM, özellikle verilerin kümeleme yapısını belirlemek ve kümeleme analizi yapmak için yaygın olarak kullanılır.

GMM, beklenti-maksimizasyon (Expectation-Maximization, EM) algoritması ile optimize edilir. Bu algoritma, veri noktalarını her bir Gauss bileşeni için bir olasılıkla ilişkilendirir ve ardından bu bileşenlerin parametrelerini (ortalama, varyans ve bileşen ağırlıkları) optimize eder. Beklenti adımında, her bir veri noktasının her bileşene ait olma olasılığı hesaplanır. Maksimizasyon adımında ise bu olasılıklara dayanarak model parametreleri güncellenir. Bu süreç, modelin parametreleri stabil hale gelene kadar tekrar eder. GMM, K-means gibi sert sınıflandırma yapmaz; bunun yerine, her veri noktasının her bir bileşene ait olma olasılığını verir, bu da yumuşak kümeleme olarak bilinir.

GMM'nin en büyük avantajı, verilerin daha karmaşık dağılımlarla daha esnek bir şekilde modellenmesine olanak tanımasıdır. Özellikle verilerin farklı şekil ve boyutlardaki kümelere ait olduğu durumlarda, GMM, K-means gibi sert kümeleme yöntemlerinden daha iyi performans gösterebilir. Ancak, GMM de bazı dezavantajlara sahiptir. Özellikle, bileşen sayısının önceden belirlenmesi gerekliliği ve bu sayı yanlış seçildiğinde modelin kötü performans gösterme riski vardır. Ayrıca, GMM'nin hesaplama karmaşıklığı, büyük veri setlerinde performans

sorunlarına yol açabilir. Bu nedenle, GMM'nin kullanımı dikkatli bir model seçimi ve değerlendirme gerektirir.

### 6.2.4 Yoğunluk Bazlı Kümeleme

Yoğunluk Bazlı Kümeleme, verilerin yoğunluklarına göre kümeler oluşturmayı amaçlayan bir kümeleme yöntemidir. Bu tür algoritmalar, yoğun bölgelerdeki veri noktalarını kümeler halinde gruplarken, düşük yoğunluklu bölgelerdeki noktaları gürültü olarak kabul eder. Bu, özellikle kümelerin karmaşık şekillerde olduğu veya verilerin içinde gürültü ya da aykırı değerlerin bulunduğu durumlarda etkili bir yöntemdir. Yoğunluk bazlı kümelemenin en bilinen algoritması DBSCAN (Density-Based Spatial Clustering of Applications with Noise)'dir.

DBSCAN, iki ana parametre kullanarak kümeler oluşturur: Epsilon (ε) ve MinPts. Epsilon, bir noktanın çevresinde bir komşuluk yarıçapı tanımlar ve bu yarıçap içinde en az MinPts sayıda nokta bulunması durumunda o nokta bir "çekirdek nokta" olarak kabul edilir. Çekirdek noktalardan başlayarak, birbirine yakın olan diğer çekirdek noktalarla birleşen kümeler oluşturulur. Bu süreç, birbirine bağlı tüm yoğun alanlar tespit edilene kadar devam eder. Kümelerin dışında kalan veri noktaları ise gürültü veya aykırı değer olarak kabul edilir.

Yoğunluk bazlı kümelemenin en büyük avantajlarından biri, küme sayısını önceden belirlemeye gerek olmamasıdır. Ayrıca, küme şekillerine karşı duyarlı olmadığı için dairesel veya karmaşık şekilli kümeleri doğru bir şekilde tespit edebilir. Ancak, DBSCAN gibi algoritmaların performansı, Epsilon ve MinPts parametrelerinin doğru bir şekilde seçilmesine bağlıdır. Bu parametreler yanlış seçildiğinde, algoritma çok fazla gürültü tespit edebilir veya tüm veriyi tek bir küme olarak değerlendirebilir. Bu nedenle, doğru parametre seçimi ve veri özelliklerine göre ayarlamalar, yoğunluk bazlı kümelemenin başarılı bir şekilde uygulanması için kritik öneme sahiptir.

## 6.3. Sınıflandırma vs Kümeleme

Sınıflandırma ve kümeleme, makine öğrenmesinde iki temel veri analiz yöntemidir, ancak birbirinden farklı amaçlara ve yaklaşımlara sahiptirler. Sınıflandırma, verileri önceden tanımlanmış etiketlere göre kategorize etmeyi amaçlayan denetimli bir öğrenme sürecidir. Bu süreçte, model, eğitim verilerinden öğrenerek, yeni ve görülmemiş verileri doğru bir şekilde sınıflandırmaya çalışır. Örneğin, bir e-posta sisteminde sınıflandırma kullanılarak gelen mesajların "spam" veya "spam değil" olarak etiketlenmesi sağlanır. Sınıflandırma algoritmaları,

belirli bir hedef değişkene bağlı olarak çalışır ve modelin doğruluğu, tahminlerin doğru olup olmadığına göre ölçülür.

Öte yandan, kümeleme, verilerin herhangi bir ön bilgi veya etiket olmaksızın doğal olarak gruplara ayrıldığı denetimsiz bir öğrenme yöntemidir. Kümeleme algoritmaları, veriler arasındaki benzerlikleri belirleyerek, bu benzerliklere dayalı olarak verileri kümeler halinde organize eder. Kümeleme, özellikle veri setlerinde gizli desenleri veya grupları ortaya çıkarmak için kullanılır. Örneğin, müşteri segmentasyonu yapmak amacıyla kümeleme algoritmaları kullanılarak, benzer alışveriş davranışlarına sahip müşteriler gruplandırılabilir. Kümeleme sonuçları genellikle daha keşif odaklıdır ve grupların doğru olup olmadığını belirlemek için doğrudan bir metrik bulunmaz.

Bu iki yöntem arasındaki temel fark, sınıflandırmanın belirli ve bilinen kategorilere yönelik bir tahmin yapması, kümelemenin ise verilerdeki gizli yapıları keşfetmeye çalışmasıdır. Sınıflandırma, etiketlenmiş veriler gerektirir ve daha çok tahmin odaklıdır; kümeleme ise etiketlenmemiş verilerle çalışır ve daha çok veri keşfi ve segmentasyon amacı taşır. İkisinin de farklı uygulama alanları ve güçlü yönleri vardır, bu nedenle hangi yöntemin kullanılacağı, çözülmesi gereken probleme bağlıdır.

## 6.4. Takviyeli Öğrenme

Takviyeli öğrenme (Reinforcement Learning, RL), makine öğrenmesi alanında bir ajan ile çevresi arasındaki etkileşimler yoluyla öğrenme sürecini modelleyen bir tekniktir. Bu yöntemde, bir ajan, belirli bir çevrede hareket eder ve eylemlerinin sonuçlarına göre ödüller veya cezalar alır. Ajanın amacı, uzun vadede en yüksek toplam ödülü elde edecek stratejiyi (politika) öğrenmektir. Bu süreçte ajan, belirli durumlarda hangi eylemleri yapması gerektiğini öğrenir ve her eylem, gelecekteki ödülleri maksimize etmek için bir strateji oluşturmasına katkıda bulunur. Takviyeli öğrenme, denetimli ve denetimsiz öğrenmeden farklıdır çünkü model, doğru cevapları bilmez; bunun yerine, çevresel geri bildirimlerden öğrenir.

Takviyeli öğrenmenin temel unsurları durumlar (states), eylemler (actions) ve ödül (reward) şeklindedir. Ajan, belirli bir durumda, çevresine uygun bir eylemde bulunur ve bu eylem sonrasında çevreden bir ödül alır. Ajan, bu ödülleri göz önünde bulundurarak gelecekteki eylemlerini optimize etmeye çalışır. Örneğin, bir robotun bir labirentte çıkışı bulması senaryosunda, robot her adımda çevresine göre bir eylem seçer ve doğru yolda ilerledikçe ödül alır. Zamanla, robot, en kısa sürede çıkışı bulmasını sağlayacak stratejiyi öğrenir.

# KAYNAKÇA


- Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Pearson.
- Burkov, A. (2019). The hundred-page machine learning book (Vol. 1, p. 32).
- Quebec City, QC, Canada: Andriy Burkov.
- Smola, A. (2008). Introduction to machine learning. Link: https://alex.smola.org/drafts/thebook.pdf

## BÖLÜM SORULARI

1) Denetimsiz öğrenme nedir ve denetimli öğrenmeden nasıl farklıdır? Bu tür bir öğrenmede model, verilerdeki hangi bilgileri ortaya çıkarmaya çalışır?

2) Kümeleme nedir ve denetimsiz öğrenmede nasıl bir rol oynar? Kümeleme algoritmalarının genel çalışma prensiplerini açıklayınız.

3) K-Means algoritmasının nasıl çalıştığını adım adım anlatınız ve bu algoritmanın hangi tür problemlerde kullanıldığını açıklayınız.

4) Takviyeli öğrenmenin ne olduğunu ve nasıl çalıştığını açıklayınız. Ajanın ödül veya ceza sistemi ile nasıl öğrenme sağladığını tartışınız.