KÜMELEME NEDİR?

Kümeleme, benzer özelliklere sahip veri noktalarını gruplara ayıran bir makine öğrenimi tekniğidir. Veri setindeki yapıyı keşfetmek ve veri noktalarını anlamlı gruplara ayırmak için kullanılır. Temel amacı, aynı kümedeki veri noktalarının birbirine benzer olması ve farklı kümeler arasındaki veri noktalarının birbirinden farklı olmasıdır.

Yaygın olarak kullanılan bazı kümeleme algoritmaları

K-Means Kümeleme

Hiyerarşik Kümeleme

DBSCAN (Yoğunluk Bazlı Kümeleme)

Gaussian Karışım Modelleri (GMM)

Spectral Kümeleme

Kümeleme algoritmaları, diğer makine öğrenimi algoritmalarından farklı bir yaklaşıma sahiptir ve belirli avantajlara sahiptir:

1. **Denetimsiz Öğrenme**: Kümeleme, denetimsiz öğrenme türüdür, yani etiketlenmemiş veri setlerinde çalışabilir. Bu, veri setindeki yapıyı keşfetmek ve anlamak için kullanılabilir, çünkü veri noktalarının etiketlenmesine gerek yoktur.

2. **Yapıyı Anlama**: Kümeleme, veri setindeki doğal yapıyı anlamak için kullanılır. Benzer özelliklere sahip veri noktalarını gruplara ayırarak, veri setindeki farklı grupları belirler.

3. **Keşif ve Segmentasyon**: Kümeleme, veri setindeki önemli grupları belirlemek ve segmentasyon yapmak için kullanılır. Bu, pazarlama analizi, müşteri segmentasyonu, kullanıcı profillerinin belirlenmesi gibi birçok alanda kullanılabilir.

4. **Ön İşleme ve Veri Hazırlığı**: Kümeleme, veri setinin ön işleme aşamasında ve veri hazırlığı sürecinde kullanılabilir. Özelliklerin ölçeklendirilmesi, eksik veri noktalarının doldurulması ve gereksiz özelliklerin çıkarılması gibi adımlarda kullanılabilir.

5. **Gürültüye Dayanıklılık**: Bazı kümeleme algoritmaları, gürültülü veriye dayanıklıdır ve gürültüyü genellikle küme dışında bırakır. Bu, veri setindeki anlamlı desenleri belirlemede ve gürültüyü dikkate almamak istediğiniz durumlarda faydalı olabilir.

Özetlemek gerekirse, kümeleme algoritmaları, veri setindeki yapıyı anlamak, önemli grupları belirlemek ve veri setini segmente etmek için kullanılır. Denetimsiz öğrenme türünde olmaları ve etiketlenmemiş veri setlerinde çalışabilmeleri, kümeleme algoritmalarını diğer algoritmalarla ayıran önemli avantajlardan biridir.

KÜMELEME ALGORİTMASI ÖRNEK

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# Örnek veri setini oluşturma
X = np.array([[1, 2], [5, 8], [1.5, 1.8], [8, 8], [1, 0.6], [9, 11]])

# K-Means algoritmasını uygulama
kmeans = KMeans(n_clusters=2) # Küme sayısını belirtme
kmeans.fit(X)

# Küme merkezlerini ve etiketleri alınması
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

# Kümeleme sonuçlarını görselleştirme
colors = ["g.", "r."] # Kümeleme etiketlerine göre renkler
for i in range(len(X)):
plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize=10)

# Küme merkezlerini görselleştirme
 plt.scatter(centroids[:, 0], centroids[:, 1], marker="x", s=150, linewidths=5, zorder=10)
plt.show()
```

Bu örnek, K-Means kümeleme algoritmasını kullanarak bir veri setini iki kümeye ayırır. Öncelikle, küme sayısı belirlenir (bu örnekte 2). Daha sonra, K-Means algoritması veri setine uygulanır ve veri noktaları kümelerine atanır. Son olarak, küme merkezleri ve etiketler alınır ve kümeleme sonuçları görselleştirilir.

Görselleştirme, veri noktalarını ve küme merkezlerini grafik üzerinde gösterir. Her veri noktası, etiketine göre yeşil veya kırmızı renkle gösterilir ve küme merkezleri çapraz işaretlerle işaretlenir. Bu şekilde, K-Means algoritmasının veri setini nasıl kümeler halinde ayırdığını görebilirsiniz.

ÖRNEK 2

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# Veri setini oluşturma
X = np.array([1, 2, 3, 10, 11, 12])

# Veri setini yeniden şekillendirme (K-Means algoritması için gerekli)
X = X.reshape(-1, 1)

# K-Means algoritmasını uygulama
kmeans = KMeans(n_clusters=2, random_state=42)  # Küme sayısını belirtme
kmeans.fit(X)

# Küme merkezlerini ve etiketleri alınması
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

# Kümeleme sonuçlarını görselleştirme
plt.scatter(X, [0]*len(X), c=labels, cmap='viridis', s=50, alpha=0.5)
plt.scatter(centroids, [0]*len(centroids), c='red', s=200, marker='x')
plt.show()
```

Bu örnekte, tek boyutlu bir veri seti oluşturuyoruz ve K-Means algoritmasıyla bu veri setini iki farklı küme halinde böleceğiz. Görselleştirme, veri noktalarını ve küme merkezlerini bir çizgi üzerinde gösterir. Kümeleme sonuçları, farklı renklerdeki veri noktaları ve küme merkezlerinin çapraz işaretlerle gösterilmesiyle görülebilir. Bu şekilde, K-Means algoritmasının tek boyutlu bir veri setini nasıl iki farklı küme halinde ayırdığını görebilirsiniz.