Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026

Hafta 5

Uzaklık Ölçütleri ve Kümeleme Yöntemleri

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

01

Euclidean, Manhattan

02

Jaccard, Cosine, Hamming

03

Sürekli, ikili ve ordinal veriler için örneklerle hesaplama

04

K-Means, DBSCAN

05

Hiyerarşik Kümeleme

06

Silhouette Skoru ile model değerlendirme

07

Uygulama

Metrik Olma Şartları

Negatif olmama (Non-negativity):

- İki nokta arasındaki mesafe negatif olamaz.
- Fiziksel olarak mesafe ölçümü her zaman sıfır veya pozitiftir.
- **$d(x,y) \geq 0$**

Özdeşlik (Identity of Indiscernibles):

- Sadece aynı nokta için mesafe sıfırdır.
- Eğer mesafe sıfırsa, bu iki nokta tamamen aynıdır.
- **$d(x,y)=0 \Leftrightarrow x=y$**

Simetri (Symmetry):

- Bir noktadan diğerine olan mesafe, ters yöndeki mesafe ile aynıdır.
- **$d(x,y) = d(y,x)$**

Üçgen Eşitsizliği (Triangle Inequality):

- Doğrudan x ile z arasındaki mesafe, arada bir başka y noktasına uğrayarak gidilen toplam mesafeden daha kısa veya eşit olmalıdır.
- **$d(x,z) \leq d(x,y) + d(y,z)$**

Euclidean Mesafesi (Öklid Mesafesi):

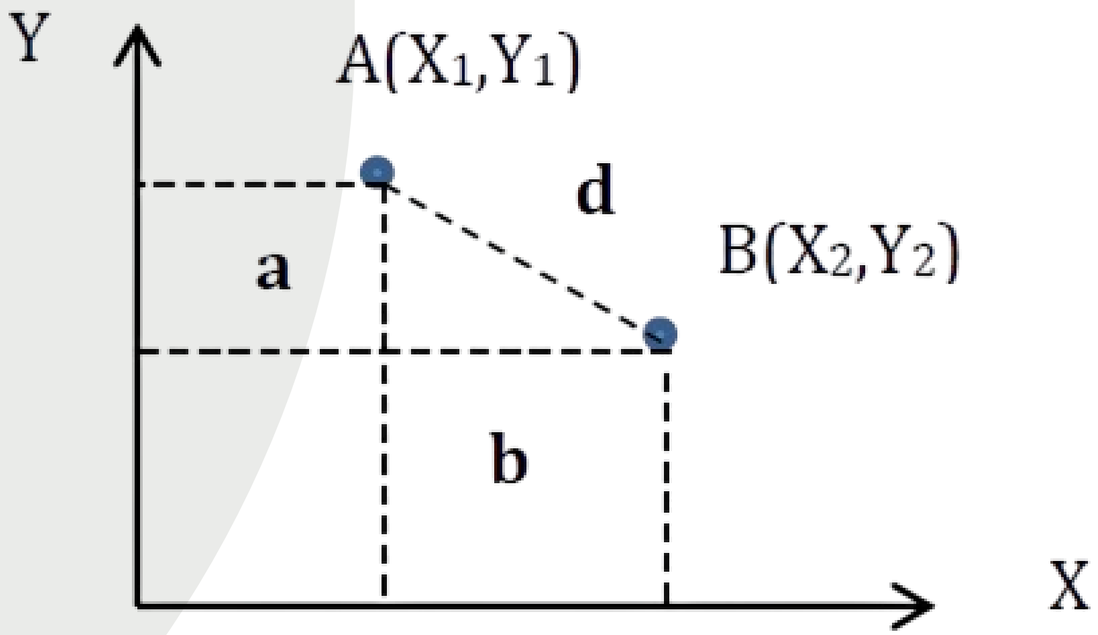
İki nokta arasındaki "düz çizgi" mesafesidir. Yani, klasik geometri derslerinde öğrendiğimiz Pisagor teoremi ile hesaplanan mesafe

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2)}$$

Öklid mesafesi

Özellikleri:

- Fiziksel dünyada gerçek mesafe gibi düşünülebilir.
- Daha çok sürekli değişkenler ve yoğun veri kümelerinde tercih edilir.
- Aykırı değerlere (outlier) karşı duyarlıdır.
- Metriktir.



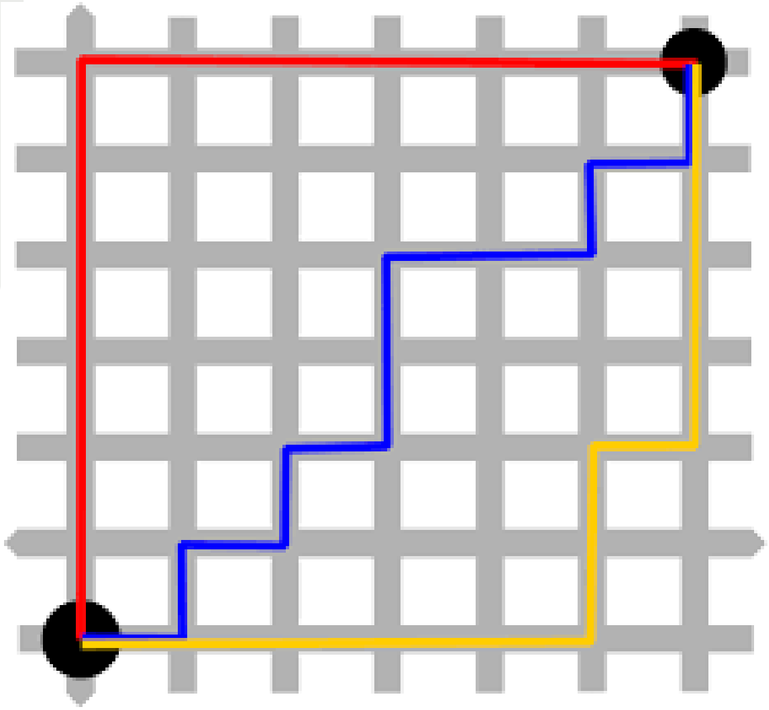
Manhattan Mesafesi (L1 Normu, Taksi Mesafesi):

İki nokta arasındaki mesafeyi yalnızca dikey ve yatay yollar üzerinden ölçer. Bir şehir haritasında taksinin sadece caddeler ve sokaklardan gitmesi gibi düşünebiliriz.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Özellikleri:

- Özellikle ızgara (grid) yapısına sahip verilerde kullanılır.
- Aykırı değerlere Euclidean'a göre daha az duyarlıdır.
- Yüksek boyutlu veri kümelerinde (high dimensional data) daha stabil çalışır.
- Metriktir.



Jaccard Benzerlik Katsayısı:

İki küme arasındaki benzerliği ölçmek için kullanılan bir metriktir. İki kümenin ortak eleman sayısının, bu kümelerin birleşimindeki toplam eleman sayısına oranı ile hesaplanır.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

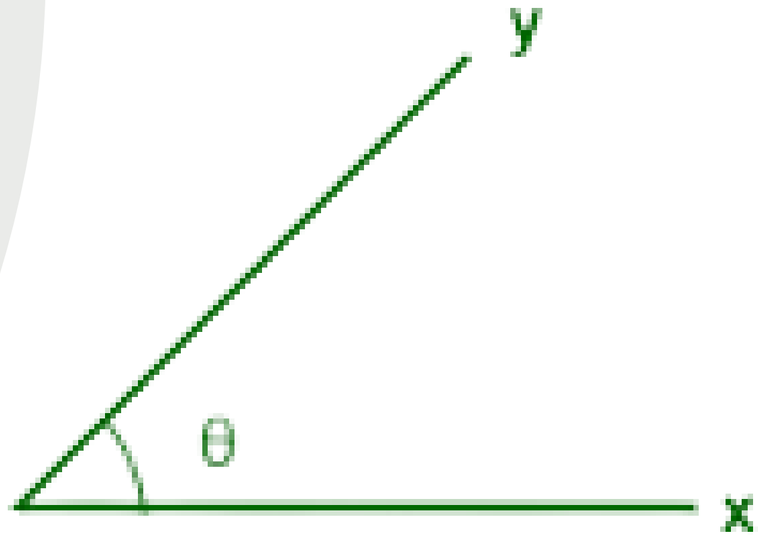
- **Değer aralığı:** 0 ile 1 arasında değişir.
  - 1: Kümeler tamamen aynı.
  - 0: Kümelerde ortak eleman yok.
- Metriktir.
- Metin madenciliği, küme karşılaştırmaları, öneri sistemleri gibi alanlarda kullanılır.

Cosine Benzerliği:

İki vektör arasındaki yön benzerliğini ölçer. Vektörlerin büyüklüklerinden ziyade, birbirlerine olan açısal yakınlıklarını değerlendirir.

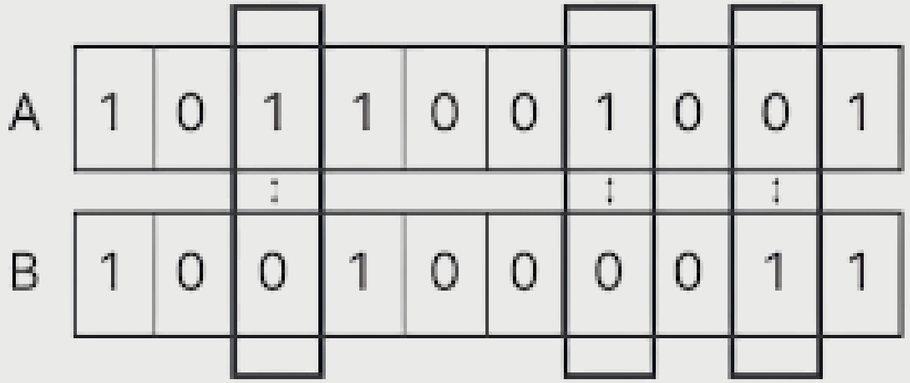
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- **Değer aralığı:** -1 ile 1 arasında değişir.
  1: Vektörler aynı yönde.
  0: Vektörler dik (ilişkisiz).
  -1: Vektörler zıt yönde.
- Doğal dil işleme (NLP), belge benzerliği ölçümü, metin analizi gibi alnlarda kullanılır.
- Diğeryöntemlerle uygun şekilde tanımlandığında metrik olabilir, ancak hepsi her zaman bu 4 koşulu sağlamayabilir.



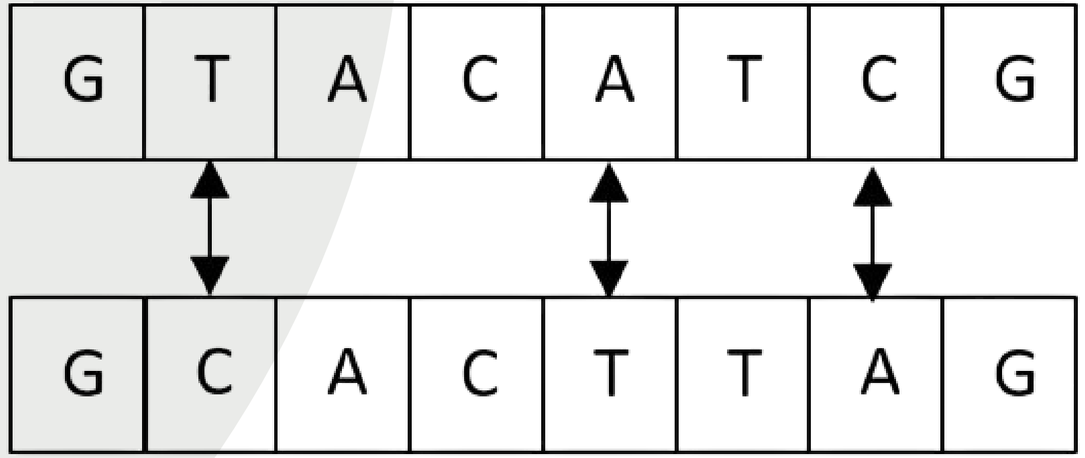
Hamming Mesafesi (Hamming Distance):

İki eşit uzunluktaki dizi/kelime arasında farklı olan pozisyon sayısını ölçer.



- **Değer Aralığı:** 0 ve pozitif tam sayılar.
  0 → Tamamen aynı
  n → Tüm karakterler farklı

- Hata tespit/düzeltme kodları, DNA dizilimi karşılaştırma gibi alanlarda kullanılır.
- Metriktir



Sürekli Veriler İçin Örneklerle Açıklama

Örnek1: A = (170 cm, 65 kg)
B = (180 cm, 75 kg)

Euclidean Distance:

$d^2 = (170\text{-}180)^2 + (65\text{-}75)^2 = 100+100 = 200$

$d \approx 14.14$

Cosine Similarity:

$$\cos(\theta) = \frac{(170 * 180 + 65 * 75)^2}{(170^2 + 65^2) * (180^2 + 75^2)} \approx 0.999$$

İkili (Binary) Veriler İçin Örneklerle Açıklama

Örnek2: A = (1, 1, 0, 1)
B = (1, 0, 0, 1)

Hamming Distance:

(1 vs 1) → aynı
(1 vs 0) → farklı (1)
(0 vs 0) → aynı
(1 vs 1) → aynı
Fark sayısı = 1 → 0.25

Jaccard Similarity:

$$\frac{\text{Ortak 1`ler}}{\text{Toplam 1`ler}} = \frac{2}{3} = 0.667$$

Ordinal Veriler İçin Örneklerle Açıklama

Örnek3: A = (5, 3, 4)
B = (4, 2, 5)

Manhattan Distance:

$d = |5-4| + |3-2| + |4-5| = 1+1+1 = 3$

Euclidean Distance:

$d^2 = (5-4)^2 + (3-2)^2 + (4-5)^2 = 1+1+1 = 3$

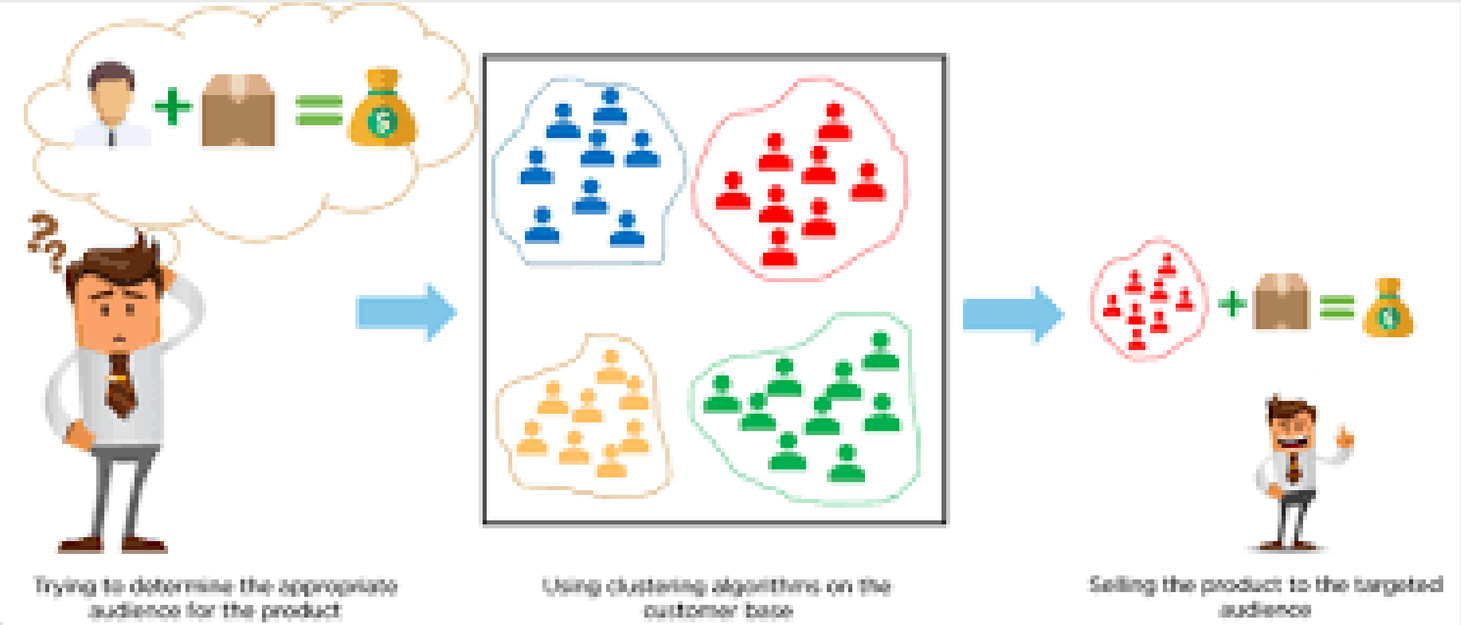
$d \approx 1.732$

Kümeleme Algoritmaları

- Küme, benzer nesnelerin oluşturduğu bir gruptur.
- Kümeleme, birbirine benzeyen nesnelerin aynı grupta toplanmasıdır.
- Aynı küme içerisinde benzerlikler fazla, kümeler arası benzerlikler ise azdır
- Kümeleme işleminde amaç, her kümeyi içindeki nesneleri en iyi temsil edecek şekilde düzenlemektir. Uygulanan veri setinde yer alan her bir veri birimine nesne adı verilir. Bu nesneler, iki boyutlu bir düzlemde genellikle noktalar şeklinde görselleştirilir.
- Kümeleme analizi, veri indirgeme, nesnelerin doğal sınıflarını bulma, örüntü keşfi ve veri özetleme gibi çeşitli amaçlarla kullanılan güçlü bir veri madenciliği yöntemidir. Bu teknik, benzer özellikler taşıyan nesneleri aynı küme içerisinde toplayarak veri setinin daha anlaşılır, düzenli ve yorumlanabilir bir yapıya kavuşmasını sağlar. Böylece hem veri içindeki gizli ilişkiler ortaya çıkarılır hem de daha verimli analizler için temel bilgi sağlanır.

Kümeleme Algoritmaları

- Kümeleme, bir "denetimsiz öğrenme" problemi olarak düşünülebilir; Etiketlenmemiş verilerden oluşan bir koleksiyonda bir yapı bulmakla ilgilenir.
- Kümeleme, "birbirine benzer üyeleri olan grupları, kümeler halinde düzenleme süreci" olarak tanımlanabilir. Bu nedenle bir küme, aralarında benzerlik bulunan ve diğer kümelere ait nesnelere benzemeyen bir nesne koleksiyonudur.
- Burada soru: neyin iyi bir kümeleme oluşturduğuna nasıl karar verileceğidir. Bir ölçüt belirlemek güçtür. Kriterleri sağlaması gereken kullanıcıdır; kullanıcının gereksinimlerine göre uyarlanmalıdır.



Kümeleme Algoritmaları Uygulama Alanları

- **Müşteri segmentasyonu:** Pazarlama stratejilerinde farklı müşteri gruplarını belirleme
- **Görüntü işleme:** Benzer pikselleri veya nesneleri gruplayarak görsel veriyi analiz etme
- **Biyoinformatik:** Gen ve protein benzerliklerini sınıflandırma
- **Anomali tespiti:** Normal dışı veri noktalarının belirlenmesi
- **Coğrafi analiz:** Benzer özelliklere sahip bölgelerin belirlenmesi
- **Arama motoru optimizasyonu:** Benzer içeriklerin gruplanması
- **Tavsiye sistemleri:** Kullanıcıların benzer ilgi alanlarına göre öneri üretme

Kümeleme Algoritmaları Gereksinimleri

- **Benzerlik veya uzaklık ölçütü** → Nesneler arasındaki benzerliği veya farklılığı belirlemek için uygun bir metrik (örn. Euclidean, Manhattan, Cosine, Jaccard).
- **Veri temsil biçimi** → Verilerin sayısal, ikili, kategorik veya karma tiplerde temsil edilmesi.
- **Küme sayısının belirlenmesi** → Bazı algoritmalarda (örn. K-Means) önceden küme sayısının bilinmesi gerekir.
- **Başlangıç koşulları** → Algoritmanın başlaması için başlangıç merkezleri veya parametrelerin belirlenmesi.
- **Küme içi homojenlik, kümeler arası heterojenlik** → Bir küme içindeki nesnelerin birbirine benzer, farklı kümelerdeki nesnelerin ise farklı olması.
- **Algoritma durdurma kriteri** → Maksimum iterasyon sayısı veya küme merkezlerinin artık değişmemesi gibi durma koşulları.
- **Veri ön işleme** → Ölçekleme, normalizasyon, eksik verilerin işlenmesi gibi ön hazırlıkların yapılması.
- **Gürültü ve aykırı değerlerle başa çıkma** → Anormal verilerin kümelenme sürecini bozmasını önleyecek mekanizmalar.
- **Ölçeklenebilirlik** → Büyük veri setlerinde makul sürede çalışabilecek şekilde tasarlanmış olması.
- **Yorumlanabilirlik** → Elde edilen kümelerin anlamlı ve analiz edilebilir olması.

Kümeleme Yöntemleri

Bölümlemeli Yöntemler ⟶
- K-Means
- K-Medoids
- CLARA

Hiyerarşik Yöntemler ⟶
- Birleştirici / Toplamalı
- Ayırıcı / Bölünmeli

Yoğunluk Bazlı Yöntemler

Grid Bazlı Yöntemler

Model Bazlı Yöntemler

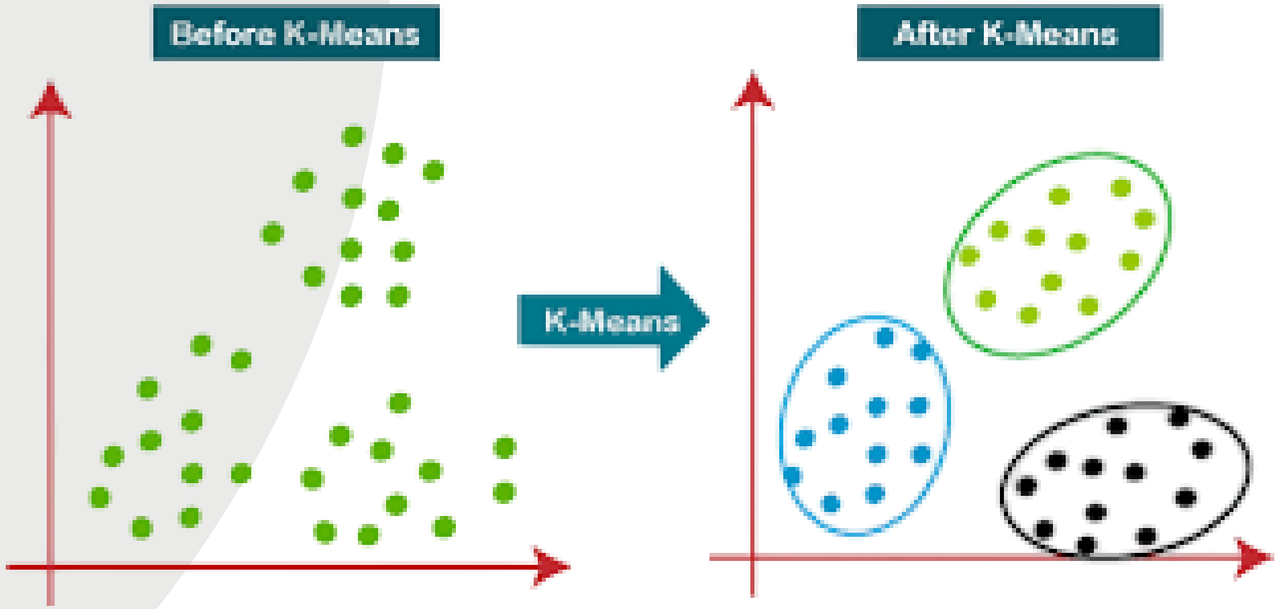
Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

K-Means

K-Means, gözetimsiz öğrenmede kullanılan, veriyi K adet kümeye bölen bir bölmeli (partitioning) kümeleme algoritmasıdır. Amaç, kümelerin “merkezleri” (centroid) etrafında küme içi kareler toplamını (SSE) en aza indirmektir.

K Nasıl Seçilir?

- **Dirsek (Elbow):** KKK arttıkça inertia/SSE’deki azalışın yavaşladığı “dirsek” noktasını seç.
- **Silhouette skoru:** [–1,1][-1,1][–1,1]. 1’e yakınsa kümeler net ayrılmıştır.
- **Gap istatistiği / BIC (GMM için):** Daha istatistiksel yöntemler.



K-Means Algoritmasının Kullanıldığı Alanlar

Müşteri Segmentasyonu:

- Pazarlamada müşterileri satın alma davranışları, demografik özellikleri veya ilgi alanlarına göre gruplamak için kullanılır.
- Örneğin, e-ticaret siteleri müşterileri “yüksek harcama yapanlar”, “fırsat kovalayanlar” gibi segmentlere ayırabilir.

Görüntü Sıkıştırma:

- Bir görüntüdeki piksel renklerini daha az sayıda renk grubuna (küme) indirerek dosya boyutunu küçültmede kullanılır.
- Özellikle JPEG sıkıştırma yöntemlerinde benzer renk tonları gruplanarak veri kaybı minimize edilir.

Anomali Tespiti:

- Veri kümesindeki “normal” davranışların dışında kalan aykırı (outlier) verileri tespit etmek için uygulanır.
- Bankacılıkta sahte işlem tespiti veya siber güvenlikte anormal ağ trafiği analizi buna örnektir.

Belge Kümeleme:

- Metin madenciliğinde benzer içeriklere sahip dokümanları gruplamak için kullanılır.
- Haber siteleri benzer haberleri bir aya getirerek konu başlıkları oluşturabilir.

K-Means Algoritmasının Kullanıldığı Alanlar

Coğrafi Veri Analizi:

- Konum verilerini gruplamak için kullanılır.
- Örneğin, bir zincir mağaza yeni şube açacağı yerleri mevcut müşteri yoğunluğu kümelerine göre belirleyebilir.

Sosyal Medya Analizi:

- Kullanıcıların beğeni, paylaşım ve takip davranışlarına göre gruplara ayrılması sağlanır.
- Bu sayede hedefli reklamcılık yapılabilir.

Tıbbi Veri Analizi:

- Hasta verilerini benzer semptomlara veya tahlil sonuçlarına göre gruplamak için kullanılır.
- Hastalık türlerinin sınıflandırılması veya tedavi planlarının kişiselleştirilmesi gibi alanlarda kullanılır.

Ürün ve Pazar Araştırmaları:

- Ürünleri benzer özelliklerine göre gruplamak veya pazar segmentlerini belirlemek için uygulanır.

K-Means Algoritmasının Temel Adımları

Küme Sayısını Belirleme (K seçimi) →

- Algoritmanın çalışabilmesi için kaç küme oluşturulacağını baştan belirlemek gerekir.
- K sayısı, veri yapısına ve analizin amacına göre seçilir.

Küme Başlangıç Merkezlerinin Seçimi →

- Her küme için başlangıç merkezleri (centroid) rastgele veya k-means++ yöntemiyle seçilir.
- Bu merkezler, kümelerin temsilcisi olacak şekilde başlatılır ve algoritmanın doğruluğunu etkiler.

Uzaklık Hesaplanması →

- Her veri noktası, her küme merkezi ile arasındaki mesafeye göre değerlendirilir.
- En yaygın kullanılan uzaklık ölçütü Euclidean distance'tır.
- Nokta en yakın merkeze atanır ve kümeler geçici olarak oluşur.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

K-Means Algoritmasının Temel Adımları

Kümeleme / Merkez Güncelleme →

- Atama adımından sonra her kümenin merkezi, o kümedeki noktaların aritmetik ortalaması alınarak güncellenir.
- Merkezler, artık kendi kümesindeki noktaları daha iyi temsil edecek şekilde hareket eder.

Küme Sayısı ve Durdurma Krite →

- Küme sayısı baştan belirlidir, fakat algoritma iterasyonlar boyunca durma koşulunu kontrol eder:Küme atamaları değişmiyorsa, Merkezlerdeki değişim çok küçükse, Maksimum iterasyon sayısına ulaşıldıysa
- Bu noktada algoritma durur ve nihai kümeler elde edilir.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Varsayımlar, Güçlü ve Zayıf Yönler

Varsayımlar / en iyi çalıştığı durumlar:

- Kümeler yaklaşık küresel/konveks ve benzer ölçekte.
- Özellikler benzer ölçeklerde (ölçeklenmiş/veri standardize).

Güçlü yönler:

- Basit, çok hızlı ve ölçeklenebilir.
- Büyük veride mini-batch ile çok pratik.

Zayıf yönler & tuzaklar:

- K sayısını bilmen gerekir.
- Aykırı değerler ve ölçek farklarına duyarlı.
- Küresel olmayan (örn. iki-ay) yapılarda zayıf.
- Başlangıca duyarlı; birden çok başlatma (n_init) önerilir.
- Boş küme oluşabilir (büyük merkez kaymaları sonrası).

Zayıflıklara çözümler

- **Ölçekle**: StandardScaler / MinMaxScaler.
- **Aykırılara:** robust ölçekleme, aykırı temizleme veya K-Medoids (PAM).
- **K'tan kaçınma:** Hiyerarşik yöntemler, DBSCAN/HDBSCAN (yoğunluk temelli).
- **Küresel olmayan yapılar:** Spektral Kümeleme, GMM, DBSCAN.

DBSCAN

DBSCAN Nedir?

- DBSCAN, veriyi yoğunluk temelli kümelere bölen bir gözetimsiz kümeleme algoritmasıdır.
- K-Means gibi küresel veya önceden belirlenmiş K kümesi gerekmez.
- Veri noktalarının yoğunluklarına göre kümeler oluşturur ve aykırı değerleri (outlier) otomatik olarak belirler.
- Yüksek yoğunluklu noktalar bir küme oluşturur.Düşük yoğunluklu veya izole noktalar gürültü (noise) olarak kabul edilir.

DBSCAN'in Temel Parametreleri

1. **eps (epsilon):**
   - Bir noktanın komşu olarak sayılacağı maksimum mesafe.
   - Nokta ile diğer noktalar arasındaki mesafe $\leqslant$ eps ise komşudur.
2. **minPts (minimum points):**
   - Bir noktanın "çekirdek noktası (core point)" sayılabilmesi için gerekli minimum komşu sayısı.

DBSCAN

DBSCAN'de Nokta Türleri

- **Core point (Çekirdek Nokta):**
  eps yarıçapında en az minPts kadar komşusu olan nokta.
- **Border point (Sınır Noktası):**
  Çekirdek noktaların eps yarıçapında yer alır ama kendi başına minPts'i sağlamaz.
- **Noise (Gürültü / Outlier):**
  Ne çekirdek ne de sınır noktasına dahil olan nokta.

Algoritmanın Adımları

- Rastgele bir nokta seç.
- Noktanın eps yarıçapındaki komşularını say.
- Eğer $\geqslant$ minPts → çekirdek noktadır → yeni küme başlat veya mevcut kümeye ekle.
- Eğer < minPts → geçici olarak gürültü olabilir.
- Çekirdek noktaların komşularını kontrol et → sınır ve yeni çekirdek noktaları kümeye ekle.
- Tüm noktalar işlenene kadar devam et.

Hiyerarşik Kümeleme Nedir?

Hiyerarşik kümeleme, verileri kümeler halinde gruplarken bu kümeler arasındaki ilişkileri hiyerarşik (ağaç şeklinde) bir yapıda gösteren kümeleme yöntemidir.

- Sonuç genellikle dendrogram adı verilen ağaç diyagramı ile görselleştirilir.
- Küme sayısını önceden belirlemek gerekmez; dendrograma bakarak karar verilir.

Birleştirici (Agglomerative) – Aşağıdan Yukarıya →

- Her veri noktası tek başına bir küme olarak başlar.
- En yakın iki küme birleştirilir.
- Tek bir küme kalana kadar devam eder.

Ayırıcı (Divisive) – Yukarıdan Aşağıya →

- Tüm veri tek bir küme olarak başlar.
- Kümeler adım adım ayrılır.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Hiyerarşik Kümeleme

Küme Mesafesi (Linkage) Ölçüm Yöntemleri

- **Single Linkage:** İki küme arasındaki en yakın noktalar arasındaki mesafe.
- **Complete Linkage:** İki küme arasındaki en uzak noktalar arasındaki mesafe.
- **Average Linkage:** İki küme arasındaki ortalama mesafe.
- **Ward's Method:** Kümeler birleştiğinde toplam varyansı en az artıracak birleşmeyi seçer.

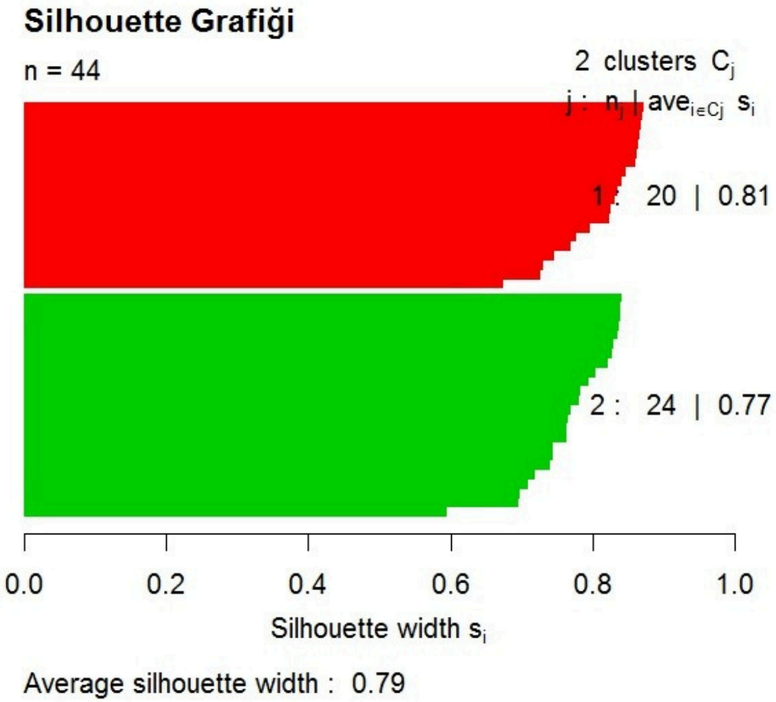
Algoritmanın Adımları

- Her veri noktası başlangıçta tek başına bir kümedir.
- Tüm kümeler arası mesafeler hesaplanır.
- En yakın iki küme birleştirilir.
- Mesafe matrisi güncellenir.
- Tek bir küme kalana kadar devam edilir.
- Dendrogram çizilir ve uygun küme sayısı belirlenir.

Silhouette Skoru Nedir?

Silhouette skoru, kümeleme algoritmasının ne kadar iyi çalıştığını değerlendiren bir ölçüttür.

- Her veri noktasının kendi kümesindeki noktalara ne kadar yakın
- ve diğer kümelerdeki noktalara ne kadar uzak olduğunu ölçer.
- Değer aralığı: -1 ile 1
  - +1 → Veri noktası çok iyi küme içine oturmuş.
  - 0 → İki küme sınırında.
  - -1 → Yanlış kümeye atanmış.
- Kümeleme sonucunu etkili bir şekilde değerlendirmeye yarar.
- Küme sayısı belirlemede yardımcı olur (en yüksek ortalama skor seçilir)



Silhouette Skoru

Hesaplama Mantığı

- Bir veri noktası iii için:

  a(i): Aynı kümedeki diğer noktalarla olan ortalama mesafe. (küme içi uzaklık)

  b(i): Diğer kümelerden en yakın olan küme ile ortalama mesafe. (küme dışı uzaklık)

Yorumlama

- 0.71 – 1.00 → Mükemmel kümeleme
- 0.51 – 0.70 → İyi
- 0.26 – 0.50 → Kabul edilebilir
- 0.00 – 0.25 → Zayıf kümeleme
- < 0 → Yanlış kümeleme

UYGULAMA

Sürekli Veri

Kod Parçası

```python
import numpy as np
import pandas as pd
from sklearn.metrics import pairwise_distances
from sklearn.metrics.pairwise import cosine_similarity

# Sürekli veri
continuous = np.array([
    [5.1, 3.5, 1.4, 0.2],
    [4.9, 3.0, 1.4, 0.2],
    [6.7, 3.1, 4.7, 1.5]
])
# Sürekli - Manhattan
manhattan_df = pd.DataFrame(pairwise_distances(continuous, metric="manhattan"),
                            columns=["A", "B", "C"], index=["A", "B", "C"])
print("\nSürekli - Manhattan Mesafe Matrisi:\n", manhattan_df)
```

Kod Çıktısı

```
Sürekli - Manhattan Mesafe Matrisi:
     A    B    C
A  0.0  0.7  6.6
B  0.7  0.0  6.5
C  6.6  6.5  0.0
```

UYGULAMA

İkili Veri

Kod Parçası

```python
import numpy as np
import pandas as pd
from sklearn.metrics import pairwise_distances
from sklearn.metrics.pairwise import cosine_similarity

# İkili veri
binary = np.array([
    [1, 0, 1, 0],
    [1, 1, 0, 0],
    [0, 0, 1, 1]
])
# İkili - Hamming
hamming_df = pd.DataFrame(pairwise_distances(binary, metric="hamming"),
                          columns=["A", "B", "C"], index=["A", "B", "C"])
print("\nİkili - Hamming Mesafe Matrisi:\n", hamming_df)

```

Kod Çıktısı

```
İkili - Hamming Mesafe Matrisi:
     A    B    C
A  0.0  0.5  0.5
B  0.5  0.0  1.0
C  0.5  1.0  0.0
```

UYGULAMA

Ordinal Veri

Kod Parçası

```python
import numpy as np
import pandas as pd
from sklearn.metrics import pairwise_distances
from sklearn.metrics.pairwise import cosine_similarity

# Ordinal veri
ordinal = np.array([
    [5, 3, 4, 2],
    [4, 2, 3, 1],
    [5, 4, 4, 3]
])

# Ordinal - Cosine Similarity
cosine_df = pd.DataFrame(cosine_similarity(ordinal),
                         columns=["A", "B", "C"], index=["A", "B", "C"])
print("\nOrdinal - Cosine Benzerlik Matrisi:\n", cosine_df)
```

Kod Çıktısı

```
Ordinal - Cosine Benzerlik Matrisi:
          A         B         C
A  1.000000  0.993808  0.988287
B  0.993808  1.000000  0.966353
C  0.988287  0.966353  1.000000
```

UYGULAMA

Müşteri Segmentasyonu

```
CustomerID,Gender,Age,Annual Income (k$),Spending Score (1-100)
1,Male,19,15,39
2,Male,21,15,81
3,Female,20,16,6
4,Female,23,16,77
5,Female,31,17,40
6,Female,22,17,76
7,Female,35,18,6
8,Female,23,18,94
9,Male,64,19,3
10,Female,30,19,72
11,Male,67,19,14
12,Female,35,19,99
13,Female,58,20,15
14,Female,24,20,77
15,Male,37,20,13
16,Male,22,20,79
17,Female,35,21,35
18,Male,20,21,66
19,Male,52,23,29
20,Female,35,23,98
```

CSV Dosyası Oluşturma ve Kaydetme:

Bu çalışmada kullanılan müşteri verileri, Mall_Customers.csv adlı bir dosyada saklanmaktadır. CSV (Comma Separated Values) dosyaları, verileri sütunlar halinde, sütunlar arasında ise virgül (,) ile ayırarak saklayan düz metin dosyalarıdır.

Dosyayı oluşturmak için:

1. Excel veya Google Sheets gibi bir tablo programı açılır.
2. İlgili başlıklar ve veri satırları tabloya girilir. Örneğin:
   - CustomerID, Gender, Age, Annual Income (k$), Spending Score (1-100)
3. Tüm veriler girildikten sonra Farklı Kaydet seçeneği ile CSV (Comma delimited) (*.csv) formatında kaydedilir.
4. Kaydedilen dosya, Python kodunun bulunduğu dizine yerleştirilir.

UYGULAMA

Müşteri Segmentasyonu

Kod Parçası

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Veri setini yükle
df = pd.read_csv("Mall_Customers.csv")
X = df[["Annual Income (k$)", "Spending Score (1-100)"]]

# K-Means modeli (k=5)
kmeans = KMeans(n_clusters=5, random_state=42)
labels = kmeans.fit_predict(X)

# Silhouette skoru
print("Silhouette Skoru:", round(silhouette_score(X, labels), 3))
```

Kod Çıktısı

```
Silhouette Skoru: 0.7
```

UYGULAMA

Müşteri Segmentasyonu

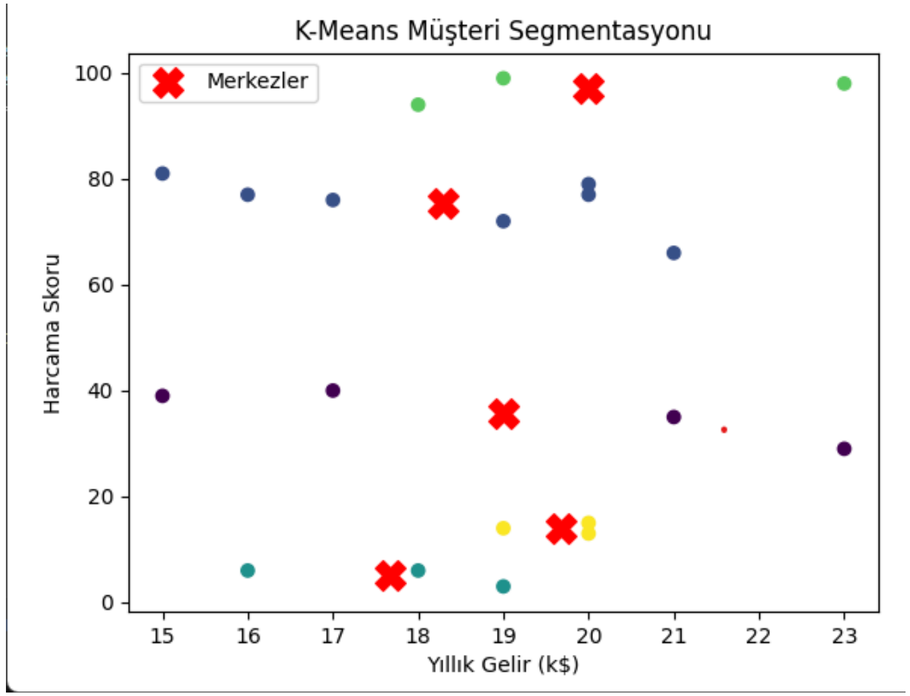
Kod Parçası

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Veri setini yükle
df = pd.read_csv("Mall_Customers.csv")
X = df[["Annual Income (k$)", "Spending Score (1-100)"]]

# K-Means modeli (k=5)
kmeans = KMeans(n_clusters=5, random_state=42)
labels = kmeans.fit_predict(X)
# Grafikte göster
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
            c='red', marker='X', s=200, label="Merkezler")
plt.xlabel("Yıllık Gelir (k$)")
plt.ylabel("Harcama Skoru")
plt.title("K-Means Müşteri Segmentasyonu")
plt.legend()
plt.show()
```

Kod Çıktısı



Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026