VERİ BİLİMİ DERSİ

Kümeleme, Zaman Serileri ve Öneri Sistemleri

Hafta 8 · Modül 8

Dr. Murat Altun

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6
Saat

3
Notebook

3
Yöntem

İçindekiler

Denetimsiz Öğrenme Nedir?

BÖLÜM 1

Denetimli vs Denetimsiz Öğrenme

Özellik	Denetimli	Denetimsiz
Etiket	Var (y değeri)	Yok
Amaç	Tahmin / Sınıflandırma	Yapı Keşfi / Gruplama
Örnek	Spam tespiti, fiyat tahmini	Müşteri segmentasyonu
Algoritma	Lojistik Reg., RF, XGB	K-Means, DBSCAN, PCA
Değerlendirme	Accuracy, RMSE	Silhouette, Inertia

Kullanım Alanları

- Müşteri segmentasyonu
- Anomali tespiti (fraud)
- Boyut indirgeme (PCA)
- Market sepet analizi

%60
Verinin Etiketsiz

3+
Kümeleme Algoritması

Gerçek dünyada verilerin büyük çoğunluğu etiketsizdir. Denetimsiz öğrenme, verinin doğal yapısını keşfetmek için en güçlü yaklaşımdır.

K-Means Kümeleme Algoritması

BÖLÜM 1

Algoritma Adımları

1. K adet rastgele merkez (centroid) seç
2. Her veri noktasını en yakın centroid'e ata
3. Her kümenin yeni centroid'ini hesapla (ortalama)
4. Centroid'ler değişmeyene kadar 2–3'ü tekrarla
5. Yakınsama → Kümeler oluştu!

Anahtar Kavramlar

Inertia: Küme içi toplam mesafe (düşük → sıkı kümeler)

K: Küme sayısı (kullanıcı belirler)

Öklid mesafesi: Varsayılan uzaklık ölçüsü

Avantajlar

- Hızlı ve ölçeklenebilir
- Anlaşılması kolay
- Büyük veri setleri için uygun

Dezavantajlar

- K önceden bilinmeli
- Küresel kümeler varsayar
- Outlier'lara duyarlı
- Başlangıç noktası etkisi

Elbow Method — Optimal K Seçimi

Dirsek Yöntemi (Elbow Method)

- Her K değeri için inertia (WCSS) hesaplanır
- Grafik K vs Inertia olarak çizilir
- Eğrinin "dirsek" yaptığı nokta optimal K'dır
- Dirsek sonrası iyileşme azalır → diminishing returns
- Gerçek dünyada net dirsek her zaman oluşmaz

```
from sklearn.cluster import KMeans

inertias = []
for k in range(2, 11):
    km = KMeans(n_clusters=k, random_state=42)
    km.fit(X_scaled)
    inertias.append(km.inertia_)

plt.plot(range(2,11), inertias, 'o-')
```

K=?

Optimal
Küme Sayısı

WCSS

Within-Cluster
Sum of Squares

Örnek Inertia Değerleri

K	Inertia	Değişim
2	1500	—
3	800	-700 ↓↓
4	500	**-300 ↓ (dirsek)**
5	420	-80 ↓
6	400	-20

Silhouette Score — Kümeleme Kalitesi

Silhouette Katsayısı

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i))$$

a(i) = Aynı küme içi ortalama uzaklık (cohesion)
b(i) = En yakın komşu kümeye ortalama uzaklık (separation)
Değer aralığı: -1 ile +1 arası

Yorumlama Rehberi

0.71 – 1.00	Mükemmel kümeleme
0.51 – 0.70	İyi kümeleme
0.26 – 0.50	Orta düzey
≤ 0.25	Zayıf / yapay kümeler

```
from sklearn.metrics import silhouette_score

km = KMeans(n_clusters=4, random_state=42)
labels = km.fit_predict(X_scaled)
score = silhouette_score(X_scaled, labels)
print(f'Silhouette Score: {score:.3f}')
```

Müşteri Segmentasyonu — RFM Analizi

R

Recency

Son alışverişten bu yana geçen gün sayısı

F

Frequency

Toplam alışveriş sayısı (sipariş adedi)

M

Monetary

Toplam harcama tutarı (TL)

K-Means ile Oluşturulan Tipik Segmentler

Segment	Recency	Frequency	Monetary	Aksiyon
VIP Müşteri	Düşük ↓	Yüksek ↑	Yüksek ↑	Sadakat programı
Risk Altında	Yüksek ↑	Orta	Orta	Geri kazanım kampanyası
Yeni Müşteri	Düşük ↓	Düşük ↓	Düşük ↓	Hoş geldin teklifi
Pasif Müşteri	Yüksek ↑	Düşük ↓	Düşük ↓	Re-engagement e-mail

RFM + K-Means Kod Uygulaması


```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# RFM metriklerini hesapla
snapshot = df['InvoiceDate'].max() + pd.Timedelta(days=1)
rfm = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (snapshot - x.max()).days,  # Recency
    'InvoiceNo':   'nunique',                            # Frequency
    'TotalPrice':  'sum'                                 # Monetary
})
rfm.columns = ['Recency', 'Frequency', 'Monetary']

# Ölçekleme + Kümeleme
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm)
km = KMeans(n_clusters=4, random_state=42)
rfm['Segment'] = km.fit_predict(rfm_scaled)
rfm['Segment'].value_counts()
```

Zaman Serileri Nedir?

Tanım ve Temel Kavramlar

Zaman serisi: Belirli zaman aralıklarında toplanan sıralı veri noktaları. Geçmiş verilerdeki örüntülerden geleceği tahmin etmeyi amaçlar. Hisse senedi fiyatları, hava sıcaklığı, satış verileri gibi alanlarda yaygın kullanılır.

24/7
Sürekli
Veri Akışı

∞
Sektör
Uygulaması

Trend

Uzun vadeli yükseliş veya düşüş. Satışların yıldan yıla artması gibi.

Mevsimsellik

Düzenli tekrar eden kalıplar. Kış aylarında enerji tüketimi artışı gibi.

Durağanlık

İstatistiksel özelliklerin zamana göre sabit kalması. Model için önemli.

Facebook Prophet — Otomatik Zaman Serisi

BÖLÜM 2

Prophet Nedir?

Meta (Facebook) tarafından geliştirilen açık kaynaklı zaman serisi tahmin aracı. Trend + mevsimsellik + tatil etkilerini otomatik ayrıştırır. Eksik veri ve outlier'lara dayanıklıdır. Minimum ayarlama ile güçlü sonuçlar verir.

Neden Prophet?

1. Otomatik trend algılama
2. Haftalık / yıllık mevsimsellik
3. Tatil etkisi ekleme desteği
4. Belirsizlik aralıkları (CI)

Trend Decomposition: $y(t) = g(t) + s(t) + h(t) + \varepsilon(t)$

$g(t)$

Trend

Doğrusal veya lojistik büyüme modeli

$s(t)$

Mevsimsellik

Fourier serileri ile periyodik değişimler

$h(t)$

Tatil Etkisi

Bayram, kampanya gibi özel günler

$\varepsilon(t)$

Hata

Modelin açıklayamadığı gürültü

Prophet — Kod Uygulaması


```
from prophet import Prophet
import pandas as pd

# Veriyi Prophet formatına dönüştür (ds, y)
df_prophet = df[['Date', 'Sales']].rename(
    columns={'Date': 'ds', 'Sales': 'y'}
)

# Model oluştur ve eğit
model = Prophet(
    yearly_seasonality=True,
    weekly_seasonality=True,
    changepoint_prior_scale=0.05
)
model.fit(df_prophet)

# 90 günlük tahmin
future = model.make_future_dataframe(periods=90)
forecast = model.predict(future)

# Görselleştirme
model.plot(forecast)
model.plot_components(forecast)
```

Uygulama: Adidas US Sales Satış Tahmini

Veri Seti: Adidas US Sales

- Kaynak: Kaggle Adidas US Sales Dataset
- Dönem: 2020–2021 (2 yıllık satış verisi)
- Özellikler: Tarih, ürün kategorisi, eyalet, satış tutarı, birim fiyat, kâr marjı
- Amaç: Prophet ile 2022 Q1 satış tahmini

Beklenen Çıktılar

- Tahmin grafiği (gerçek + forecast + CI)
- Trend bileşeni: Genel yükseliş eğilimi
- Haftalık mevsimsellik: Haftasonu ↑
- Yıllık mevsimsellik: Q4 (tatil sezonu) ↑
- MAPE < %15 hedefi

9.6K
Satış Kaydı

90
Günlük Tahmin

Analiz İş Akışı

1. Veri yükleme ve tarih dönüşümü
2. Günlük satış toplamı → zaman serisi
3. Prophet modeli eğitimi
4. 90 günlük gelecek tahmini
5. Trend ve mevsimsellik analizi

Öneri Sistemleri Nedir?

Tanım

Kullanıcıların ilgisini çekebilecek ürünleri, içerikleri veya hizmetleri otomatik olarak tahmin eden ve öneren sistemlerdir. Kişiselleştirilmiş deneyim sunarak kullanıcı memnuniyetini ve geliri artırır.

%35
Amazon Geliri

%80
Netflix İzlenmesi

Netflix

İzleme geçmişi + puanlama → film/dizi önerisi

300M+
Abone

Spotify

Dinleme alışkanlıkları → haftalık keşif listesi

600M+
Kullanıcı

Amazon

Satın alma + göz atma → ürün önerisi

350M+
Ürün

YouTube

İzleme süresi + etkileşim → video önerisi

2B+
Kullanıcı

İçerik Tabanlı vs İşbirlikçi Filtreleme

İçerik Tabanlı (Content-based)

Öğenin özellikleri ile kullanıcı profilini eşleştirir.

- Ürün/içerik özelliklerini analiz eder
- Kullanıcının geçmiş tercihlerini kullanır
- Yeni kullanıcı sorunu (cold start) az
- TF-IDF, cosine similarity kullanır

Örnek:

"Aksiyon filmi sevdin → benzer aksiyon filmleri öner"
Filmin türü, yönetmeni, oyuncuları analiz edilir.

İşbirlikçi Filtreleme (Collaborative)

Benzer kullanıcıların tercihlerini kullanır.

- Kullanıcı-öğe etkileşim matrisini analiz eder
- Öğe özelliklerine ihtiyaç duymaz
- Sürpriz öneriler yapabilir (serendipity)
- SVD, ALS, NMF gibi matris ayrıştırma

Örnek:

"Sana benzeyen kullanıcılar şunu beğendi → sana da öner"
Kullanıcı puanlamaları ve davranışları analiz edilir.

Collaborative Filtering — Matris Ayrıştırma

Kullanıcı-Ürün Puanlama Matrisi

	Film A	Film B	Film C	Film D
Ali	5	3	?	1
Ayşe	4	?	5	2
Mehmet	?	4	4	?
Zeynep	3	5	?	3

Cosine Similarity

$$\cos(A, B) = (A \cdot B) / (||A|| \times ||B||)$$

- Vektörler arası açıyı ölçer
- 0 = ilişkisiz, 1 = özdeş tercihler
- Yüksek benzerlik → güçlü öneri
- Seyrek matris sorununa çözüm: SVD

SVD (Singular Value Decomposition) — Matris Ayrıştırma

$R \approx U \times \Sigma \times V^T$

Rating matrisi, kullanıcı ve öğe gizli faktörlerine ayrıştırılır

Boyut İndirgeme

Binlerce boyutlu matris k boyuta indirilir (genellikle k=50-200)

Boşluk Doldurma

"?" hücreleri tahmin edilir → öneri yapılır

Film Öneri Sistemi — Surprise SVD


```python
from surprise import Dataset, Reader, SVD
from surprise.model_selection import cross_validate

# MovieLens veri seti
data = Dataset.load_builtin('ml-100k')

# SVD modeli oluştur ve değerlendir
model = SVD(n_factors=100, n_epochs=20, lr_all=0.005)
results = cross_validate(
    model, data, measures=['RMSE', 'MAE'], cv=5
)
print(f"RMSE: {results['test_rmse'].mean():.4f}")

# Tüm veri ile eğit + tahmin
trainset = data.build_full_trainset()
model.fit(trainset)

# Kullanıcı 196 için Film 302 tahmin puanı
pred = model.predict(uid='196', iid='302')
print(f"Tahmini puan: {pred.est:.2f}")
```

3 Yöntem Karşılaştırma Tablosu

ÖZET

Özellik	Kümeleme	Zaman Serisi	Öneri Sistemleri
Amaç	Gruplama / Segmentasyon	Gelecek değer tahmini	Kişiselleştirilmiş öneri
Veri Tipi	Özellik vektörleri	Zamana bağlı sıralı veri	Kullanıcı-öğe etkileşimi
Algoritma	K-Means, DBSCAN	Prophet, ARIMA, LSTM	SVD, ALS, Content-based
Çıktı	Küme etiketi (0, 1, 2..)	Sayısal tahmin + CI	Top-N öneri listesi
Metrik	Silhouette, Inertia	MAPE, RMSE	RMSE, Precision@K
İş Uygulaması	Müşteri segmentasyonu	Satış/stok tahmini	Ürün/içerik önerme
Python Kütüphanesi	scikit-learn	prophet, statsmodels	surprise, implicit
Veri Boyutu	Orta-Büyük	Zaman serisi uzunluğu	Seyrek matris (sparse)

Haftalık Notebook'lar

UYGULAMA

1 Notebook 1: K-Means Segmentasyon

`kmeans_segmentasyon.ipynb`

Online Retail veri seti ile RFM analizi ve müşteri segmentasyonu. Elbow Method ve Silhouette Score ile optimal K belirleme.

K-Means | RFM | Elbow | Silhouette

2 Notebook 2: Zaman Serisi Prophet

`zaman_serisi_prophet.ipynb`

Adidas US Sales verisi ile satış tahmini. Prophet modeli, trend decomposition, mevsimsellik analizi ve 90 günlük forecast.

Prophet | Forecast | Trend | Mevsimsellik

3 Notebook 3: Film Öneri Sistemi

`film_oneri.ipynb`

MovieLens 100K veri seti ile collaborative filtering. Surprise SVD modeli, cross-validation ve kişiselleştirilmiş öneriler.

SVD | MovieLens | Surprise | Collaborative

Ödev ve Kaynaklar

UYGULAMA

Bu Hafta Ödevleri

Ödev 1: Mall Customer Segmentasyon

Kaggle Mall Customers veri seti ile K-Means kümeleme. Müşterileri gelir ve harcama skoruna göre segmentlere ayırın. Elbow + Silhouette ile K seçin.

Ödev 2: MovieLens Öneri Sistemi

MovieLens 100K veri seti ile SVD tabanlı film öneri sistemi kurun. RMSE'yi minimize edin ve kendinize 5 film önerisi oluşturun.

Faydalı Kaynaklar

1. scikit-learn Clustering Dokümantasyonu
   scikit-learn.org/stable/modules/clustering
2. Prophet Dokümantasyonu
   facebook.github.io/prophet/
3. Surprise Kütüphanesi
   surpriselib.com
4. Kaggle: Mall Customers Dataset
   kaggle.com/datasets/vjchoudhary7/...
5. Kaggle: MovieLens 100K
   kaggle.com/datasets/prajitdatta/...

Hafta 8 — Öne Çıkan Noktalar

1. Denetimsiz öğrenme, etiketsiz veriden anlamlı yapılar keşfeder — K-Means en yaygın yöntemdir.
2. RFM analizi ile müşteri segmentasyonu, pazarlama stratejisinin temelini oluşturur.
3. Facebook Prophet, minimum kod ile güçlü zaman serisi tahminleri yapar.
4. Öneri sistemleri, kişiselleştirilmiş deneyim sunarak geliri %35'e kadar artırır.
5. 3 yöntem birlikte kullanıldığında veri biliminin tam gücü ortaya çıkar.

“Verinin dilini anlamak, geleceği şekillendirmenin ilk adımıdır.”

Dr. Murat Altun · Veri Bilimi ve Yapay Zekâ Eğitimi · 2026