Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026

Hafta 3
Veri Temizleme ve Önişleme Teknikleri

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

01

Eksik verilerin doldurulması

02

Aykırı değerlerin belirlenmesi ve işlenmesi

03

Normalizasyon, standardizasyon

04

Uygulama

Eksik Veri Problemi ve Önemi

Veri analizi sürecinde, eksik veriler kaçınılmazdır. Ölçüm hataları, sensör arızaları, katılımcıların anket sorularını boş bırakması veya teknik aksaklıklar gibi nedenlerle bazı veri noktaları kaydedilemeyebilir. Eksik veri, bir veri kümesinde bulunması gereken bilginin mevcut olmaması durumudur. Eksik veriler doğru şekilde ele alınmazsa, analiz sonuçlarını yanıltabilir, önyargılı yorumlara yol açabilir ve makine öğrenmesi modellerinin doğruluğunu düşürebilir. Bu nedenle eksik veri analizi ve uygun doldurma yöntemleri, veri ön işleme sürecinin kritik adımlarından biridir.

Eksik Verileri Doldurma Yöntemleri

Silme Yöntemleri

- **Kayıt (Satır) Silme:** Eksik veri içeren satırlar tamamen çıkarılır.
- **Değişken (Sütun) Silme:** Eksik veri oranı çok yüksek olan sütunlar tamamen kaldırılır.
- **Dezavantaj:** Çok veri kaybına yol açabilir, örneklem küçülür

Zaman Serilerinde Eksik Veri Doldurma

- **Önceki Değerle Doldurma (Forward Fill):** Eksik noktaya en yakın önceki değer atanır.
- **Sonraki Değerle Doldurma (Backward Fill):** Eksik noktaya en yakın sonraki değer atanır.
- **Interpolasyon:** Eksik noktalar, çevresindeki değerler arasında matematiksel tahminle doldurulur.

Basit Doldurma Yöntemleri

- **Sabit Değer ile Doldurma:** Eksik hücrelere belirlenen sabit bir değer (ör. 0, “Bilinmiyor”) yazılır.
- **Ortalamayla Doldurma:** Sayısal değişkenlerde, eksik hücreler o sütunun ortalaması ile doldurulur.
- **Ortanca (Median) ile Doldurma:** Özellikle uç değerlerin çok olduğu durumlarda tercih edilir.
- **Mod ile Doldurma:** Kategorik değişkenlerde en sık görülen değer kullanılır.

Gelişmiş Doldurma Yöntemleri

- **K-En Yakın Komşu (KNN) İmputasyonu:** Eksik değerler, benzer diğer gözlemlerden tahmin edilir.
- **Regresyon ile Doldurma:** Eksik değer, diğer değişkenler kullanılarak tahmin edilir.
- **Çoklu İmputasyon (Multiple Imputation):** Birden fazla tahmin modeli kullanılarak birden fazla tamamlanmış veri seti üretilir, ardından sonuçlar birleştirilir.

Aykırı Değer Nedir?

Aykırı değer, veri kümesindeki genel eğilimden veya dağılımdan belirgin şekilde farklı olan gözlemlerdir.
Yani, veri setindeki çoğu değer belirli bir aralıkta toplanırken, aykırı değer bu aralığın çok dışında yer alır. Örneğin Bir sınıfta öğrencilerin boyları genelde 160–180 cm arasında ise, 120 cm veya 210 cm boyundaki biri aykırı değer sayılır.
Aykırı değerler, istatistiksel yöntemler (Z-skoru, IQR), görselleştirme teknikleri (boxplot) veya algoritmalar (Isolation Forest, DBSCAN) ile tespit edilebilir..(Genellikle büyük veri ve makine öğrenmesinde Algoritmik yöntemler kullanılır.)

Aykırı Değerler Nasıl Belirlenir?

İstatistiksel Yöntemler

a) Z-Skoru (Standart Sapma Yöntemi)

- Verinin ortalamadan kaç standart sapma uzakta olduğunu ölçer.

$$z = \frac{x-\mu}{\sigma}$$

- X → veri değeri
  μ → Ortalama
  σ → Standart sapma

- |Z|>3 olduğunda o veri aykırı veri kabul edilir.

b) IQR (Çeyrekler Arası Aralık) Yöntemi

- Veriyi sıralayıp, ortadaki %50'lik kısmın yayılımına göre uç değerleri bulmak.

$$IQR = Q3 - Q1$$

- Q1 → Verinin %25'lik kısmının alt değeri (1. çeyrek)
- Q3 → Verinin %75'lik kısmının üst değeri (3. çeyrek)

- **Alt sınır =** Q1 – 1.5 × IQR
- **Üst sınır =** Q3 + 1.5 × IQR

bu sınırların dışında kalan değerler aykırı değer olarak kabul edilir.

Aykırı Değerler Nasıl Belirlenir?

Görsel Yöntemler

a) Boxplot (Kutu Grafiği)

- Verinin medyanını, çeyreklerini (Q1 ve Q3) ve aykırı değerleri tek bir grafikte gösterir.
- Ortadaki kutu → Q1 ile Q3 arasındaki değerler
  Kutunun içindeki çizgi → medyan
  “Whisker” (çizgi kolları) → uç değer sınırları
- Özellikle veri setinde uç noktaların nerede olduğunu görmek için kullanılır.
- Hızlı ve anlaşılır, tek değişken için ideal bir yöntemdir.

b) Scatter Plot (Saçılım Grafiği)

- İki değişken arasındaki ilişkiyi noktalarla gösterir. Aykırı değerler genelde bu noktaların genel kümesinden uzak olarak görünür.
- İlişkiyi ve aykırı değerleri aynı anda görme imkânı veren bir yöntemdir.
- Özellikle regresyon analizi veya iki boyutlu verilerde kullanılır.

Aykırı Değerler Nasıl Belirlenir?

Algoritmik Yöntemler

a) Isolation Forest

- Bu yöntem, veriyi rastgele bölerek aykırı değerleri “izole” etmeye çalışır.
- Aykırı değerler, diğerlerinden daha kolay ayrılabildiği için daha kısa bölünme yollarına sahiptir.
- Hem sayısal hem kategorik verilerde kullanılabilir.
- Sızıntı tespiti, dolandırıcılık tespiti, anomali analizinde kullanılır.

b)DBSCAN

- Bir kümeleme algoritmasıdır. Yoğun bölgelerdeki noktaları küme olarak gruplar, düşük yoğunluklu bölgelerde kalan noktaları “gürültü” yani aykırı değer olarak işaretler
- Konum verisi, sensör verisi, uzamsal veri analizlerinde kullanılır

c)LOF

- Her veri noktasının komşularına olan uzaklık yoğunluğunu hesaplar.
- Bir nokta, komşularına göre çok daha seyrek bölgede yer alıyorsa aykırı değer kabul edilir.
- Özellikle çok boyutlu anomali tespitlerinde kullanılır.

Kesikli Hale Getirme (Discretization)

Sürekli verilerin kesikli hale dönüştürülmesi işlemidir. Bu yöntem, verinin anlamlı kategorilere bölünmesini sağlar ve bazen model başarısını artırır. Eşit genişlik (equal-width), eşit frekans (equal-frequency), k-means temelli discretization gibi yöntemler kullanılır. Birçok sınıflandırma algoritması kesikli veriler ile daha iyi ve doğru sonuçlar verirler.

İkili Hale Getirme (Binarization)

Sürekli yada kategorik değişkenin ikili (1-0) sisteme dönüştürülmesidir. Genellikle birliktelik kuralları için gerçekleştirilmektedir.Sürekli veriler önce kategorik verilere daha sonra ikili değerlere çevrilir.Bu yöntem özellikle lojistik regresyon, karar ağaçları ve sinir ağlarında kullanılır. Ayrıca kategorik veriler one-hot encoding ile ikili vektörlere dönüştürülür.

Veri Ön İşlemede Ölçeklendirme Teknikleri

Normalizasyon (Min-Max Scaling):

- Verileri belirli bir minimum ve maksimum değere (genellikle 0 ile 1 arasında) dönüştürür.
- Her sütunun en yüksek değerinden en düşük değeri çıkarın ve aralığa bölünerek bulunur. Her yeni sütunun en düşük değeri 0, en yüksek değeri ise 1'dir.
- Sezonsal etki gibi istenmeyen özellikler ortadan kalkar.
- Özellikle veriler farklı ölçeklerdeyse, onları aynı aralığa getirerek modellerin performansını artırırmakta önemli rol oynar.

$$z = \frac{x \ - \min(x)}{\max(x) - \min(x)}$$

Standardizasyon (Z-skoru Standardizasyonu):

- Verileri ortalaması 0, standart sapması 1 olacak şekilde dönüştürür.
- Aykırı değerlere karşı normalizasyondan daha dayanıklıdır.

$$z = \frac{x_i - \mu}{\sigma}$$

- μ –> ortalama
  σ → standart sapma

PCA

PCA, yani Başlıca Bileşen Analizi, yüksek boyutlu verileri daha az sayıda bileşene indirgemek için kullanılan bir yöntemdir. Bu sayede veri setindeki önemli bilgiyi koruyarak, verinin boyutunu azaltır ve görselleştirmeyi kolaylaştırır. Ayrıca, gürültüyü azaltıp modellerin performansını artırmaya yardımcı olur. Hem hesaplama maliyetini azaltır hem de veriyi görselleştirmeyi kolaylaştırır.

Nasıl Kullanılır?

- Verinin varyansını maksimize eden doğrultuları (bileşenleri) bulur.
- İlk bileşen, verideki en yüksek varyansı taşır, ikinci bileşen ise birinciden bağımsız en yüksek ikinci varyansı taşır ve böyle devam eder.
- Bu bileşenler doğrusal birleşimlerdir ve orijinal değişkenlerden türetilir.

Boyut indirgeme, Gürültü azaltma, Veri görselleştirme, Ön işleme olarak modelin performansını artırmada kullanılır.

UYGULAMA

Normalizasyon

Kod Parçası

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.decomposition import PCA
import numpy as np

# Örnek veri (5 örnek, 3 özellik)
X = np.array([
    [10, 200, 0.5],
    [15, 180, 0.7],
    [7, 210, 0.2],
    [20, 190, 0.9],
    [13, 195, 0.4]
])

# MinMaxScaler ile normalizasyon
min_max_scaler = MinMaxScaler()
X_minmax = min_max_scaler.fit_transform(X)
print("MinMaxScaler ile normalizasyon:\n", X_minmax)
```

Kod Çıktısı

```
MinMaxScaler ile normalizasyon:
 [[0.23076923 0.66666667 0.42857143]
 [0.61538462 0.         0.71428571]
 [0.         1.         0.        ]
 [1.         0.33333333 1.        ]
 [0.46153846 0.5        0.28571429]]
```

UYGULAMA

Standardizasyon

Kod Parçası

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.decomposition import PCA
import numpy as np

# Örnek veri (5 örnek, 3 özellik)
X = np.array([
    [10, 200, 0.5],
    [15, 180, 0.7],
    [7, 210, 0.2],
    [20, 190, 0.9],
    [13, 195, 0.4]
])
# StandardScaler ile standardizasyon
standard_scaler = StandardScaler()
X_standard = standard_scaler.fit_transform(X)
print("\nStandardScaler ile standardizasyon:\n", X_standard)
```

Kod Çıktısı

```
StandardScaler ile standardizasyon:
 [[-0.67763093  0.5        -0.16552118]
 [ 0.45175395 -1.5         0.66208471]
 [-1.35526185  1.5        -1.40693001]
 [ 1.58113883 -0.5         1.4896906 ]
 [ 0.          0.        -0.57932412]]
```

UYGULAMA

Standardizasyon

Kod Parçası

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.decomposition import PCA
import numpy as np

# Örnek veri (5 örnek, 3 özellik)
X = np.array([
    [10, 200, 0.5],
    [15, 180, 0.7],
    [7, 210, 0.2],
    [20, 190, 0.9],
    [13, 195, 0.4]
])

standard_scaler = StandardScaler()
X_standard = standard_scaler.fit_transform(X)

# PCA ile boyut indirgeme (2 bileşene indirgeme)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_standard)
print("\nPCA ile boyut indirgeme sonucu:\n", X_pca)

```

Kod Çıktısı

```
PCA ile boyut indirgeme sonucu:
 [[-0.7721068   0.06922206]
 [ 1.48482142 -0.8229622 ]
 [-2.45665813  0.17488424]
 [ 2.08673263  0.78346902]
 [-0.34278912 -0.20461312]]
```

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026