

# Pusula Vaka Çalışması Dökümantasyonu

Beytullah Bektaş

Beytullahbektas19@gmail.com

## 1. Proje Tanımı ve Amacı

Bu proje, **fiziksel tıp ve rehabilitasyon** alanına ait **hasta verisi** üzerinde gerçekleştirilmiştir. Veri seti 2235 gözlem ve 13 sütundan oluşmaktadır.

- **Projenin Tanımı:**  
Çalışmada, hasta kayıtlarını içeren veri seti üzerinde **Exploratory Data Analysis (EDA)** yapılmış, veri kalitesi incelenmiş ve verinin makine öğrenmesi modellerine hazır hale getirilmesi için **ön işleme (preprocessing)** adımları uygulanmıştır.
- **Amaç:**  
Bu proje kapsamında modelleme yapılması beklenmemektedir. Asıl hedef, veriyi **temiz, tutarlı ve analiz edilebilir** hale getirmektir. Bunun için:
  - Eksik değerlerin uygun yöntemlerle doldurulması,
  - Tutarsız değerlerin ayıklanması,
  - Kategorik değişkenlerin encode edilmesi,
  - Sayısal değişkenlerin normalize/standartlaştırılması,
  - Bütün bu adımların tekrar edilebilir hale getirilmesi için **pipeline yapısı** kurulması planlanmıştır.
- **Hedef Değişken (Target):**  
Projede hedef değişken **TedaviSuresi (tedavi süresi - seans sayısı)** olarak tanımlanmıştır.

## 2. Veri Setinin Özellikleri ve Hataları

### Genel Özellikler

- **Gözlem Sayısı:** 2235
- **Değişken Sayısı:** 13
- **Benzersiz Hasta Sayısı:** 404
- **Veri Yapısı:** Her hasta için birden fazla satır kayıt bulunmakta (örneğin aynı hastanın farklı tedavi seansları).

### Sütunların Açıklamaları

- **HastaNo:** Anonimleştirilmiş hasta kimliği.
- **Yas:** Hastanın yaşı.
- **Cinsiyet:** Erkek / Kadın.
- **KanGrubu:** Kan grubu bilgisi (ör. 0 Rh+, A Rh-).
- **Uyruk:** Hastanın uyruğu.
- **KronikHastalik:** Kronik rahatsızlıkları (virgülle ayrılmış).
- **Bolum:** Hastanın tedavi gördüğü bölüm(ler).
- **Alerji:** Alerji bilgisi (boş veya virgülle ayrılmış olabilir).
- **Tanilar:** Hastaya konulan tanılar.

- **TedaviAdi:** Uygulanan tedavi adı.
- **TedaviSuresi:** Tedavi süresi (seans sayısı).
- **UygulamaYerleri:** Uygulama yapılan vücut bölgeleri.
- **UygulamaSuresi:** Uygulama süresi (dakika).

#### Hatalar ve Tutarsızlıklar

- **Duplikatlar:** 1379 satır birebir aynı → %61,7 oranında tekrar.
- **Eksik Değerler:**
  - Cinsiyet sütununda 169 eksik kayıt.
  - Kan Grubu sütununda 675 eksik kayıt (~%30).
  - Alerji sütununda yoğun NaN değerler.
- **Tutarsız Değerler:**
  - “Uyruk” sütununda çoğunlukla “Türkiye” bulunurken, 27 adet “Tokelau” değeri var → anormal/veri hatası olma ihtimali yüksek.
  - KronikHastalik, Tanilar ve Bolum sütunlarında birden fazla değer aynı hücrede tutulmuş → preprocessing sırasında ayrıştırma gerekebilir.
- **Veri Tipi Problemleri:**
  - **TedaviSuresi:** “15 Seans” gibi metin formatında → sayısallaştırılmalı.
  - **UygulamaSuresi:** “20 Dakika” formatında → sayısallaştırma

### 3. Veri Yükleme ve Kullanılan Kütüphaneler

Proje kapsamında veri analizi ve ön işleme adımlarını gerçekleştirmek için Python programlama dili ve çeşitli kütüphaneler kullanılmıştır.

#### Kullanılan Kütüphaneler

- **pandas:** Veri okuma, işleme ve tablo yapısında analiz için.
- **numpy:** Sayısal işlemler ve matematiksel hesaplamalar için.
- **matplotlib.pyplot:** Grafik çizimleri için temel kütüphane.
- **seaborn:** İleri düzey istatistiksel görselleştirmeler için.
- **scikit-learn (StandardScaler, OneHotEncoder):** Sayısal verilerin ölçeklenmesi ve kategorik değişkenlerin dönüştürülmesi için.
- **openpyxl:** Excel dosyalarının pandas tarafından okunabilmesi için.
- **warnings:** Gereksiz uyarı mesajlarını bastırmak için.

#### Kod Açıklaması

- Öncelikle gerekli kütüphaneler projeye import edilmiştir.

- **warnings.filterwarnings("ignore")** satırı, gereksiz uyarıların çıktı ekranını doldurmaması için kullanılmıştır.
- **%matplotlib inline** ifadesi, grafiklerin Jupyter Notebook ortamında hücre altında görünmesini sağlar (script çalıştırmada gerekli değildir).
- Veri seti, **pd.read\_excel()** fonksiyonu ile yüklenmiştir.
- **df.head()** ile ilk 5 satır görüntülenerek verinin yapısı hızlıca incelenmiştir.
- **df.info()** fonksiyonu ile sütunların veri tipleri, null (eksik) değer olup olmadığı ve kayıt sayıları hakkında genel bilgi alınmıştır.

### Bu Adımın Amacı

Bu bölümün amacı, veri setini projeye dahil etmek ve ilk aşamada:

- Veri tiplerini görmek,
- Eksik değer olup olmadığını hızlıca kontrol etmek,
- Veri setinin boyutunu anlamak

olmuştur. Böylece, sonraki adımlarda hangi sütunlara temizlik, dönüşüm veya eksik değer doldurma işlemleri yapılacağı belirlenebilecektir.

## 5. Veri Temizleme Adımları

Bu bölümde veri setinde yer alan tutarsızlıklar, eksik değerler ve yanlış biçimlendirilmiş bilgiler düzeltilmiştir. Amaç, veriyi **tutarlı, temiz ve analiz edilebilir** hale getirmektir.

### 5.1 Kronik Hastalıklar (KronikHastalik)

- Tüm değerler **küçük harfe** çevrilip baştaki/sondaki boşluklar silindi.
- **Yanlış yazılmış terimler** (örneğin “hiporitiroidizm” → “hipotiroidizm”, “gripın” → “gripin”) düzeltilerek standart hale getirildi.
- Eksik değerler boş string (“”) ile dolduruldu.
- Değerler **liste formatına** dönüştürüldü, böylece birden fazla hastalık ayrı ayrı etiketlenebilir hale getirildi.

**Amaç:** Analiz sırasında aynı hastalığın farklı yazımlarla ayrı kategoriye düşmesini engellemek

### 5.2 Bölüm (Bolum)

- Bölüm isimleri üzerinde yazım standardizasyonu yapıldı (örneğin “Laboratuar” → “Laboratuvar”).
- Başta ve sonda bulunan gereksiz boşluklar temizlendi.

**Amaç:** Aynı bölümler farklı yazımlarla tekrar etmesin, istatistiksel analizde karışıklık olmasın.

### 5.3 Alerji (Alerji)

- Tüm veriler string tipine dönüştürüldü, küçük harfe çevrildi ve boşluklar temizlendi.
- Yaygın yazım hataları ve varyasyonlar (ör. “yer fstg” → “Yer Fıstığı”, “voltaren” → “Voltaren”) birleştirildi.
- Birden fazla alerji virgülle ayrılarak listelere dönüştürüldü.
- Her kelimenin ilk harfi büyük yazılarak okunabilirlik artırıldı.

**Amaç:** Alerji bilgilerini daha düzenli hale getirip analize uygun forma getirmek.

### 5.4 Tanılar (Tanılar)

- Veriler küçük harfe çevrildi, gereksiz boşluk ve tekrar eden virgüller temizlendi.
- Sık görülen yanlış yazımlar düzeltilerek standartlaştırıldı (ör. “dorsalji” → “Dorsalji”, “lumbosakral bölge” → “Lumbosakral Bölge”).
- Eksik değerler boş string ile dolduruldu.
- Değerler liste formatına dönüştürüldü.

**Amaç:** Çok değerli tanıları ayrı ayrı işleyebilmek ve yanlış yazım kaynaklı tekrarları önlemek.

### 5.5 Tedavi Adı (TedaviAdi)

- Küçük harfe çevrilip boşluklar temizlendi.
- Yazım farklılıkları normalize edildi (ör. “dorsalji 1” → “Dorsalji”, “gonartroz-meniskopati” → “Gonartroz - Meniskopati”).
- Regex kullanılarak benzer tipteki tedaviler tek isim altında birleştirildi (ör. tüm “iv disk bozukluğu” varyasyonları → “İV Disk Bozukluğu”).
- Son olarak baş harfleri büyütülerek okunabilirlik artırıldı.

**Amaç:** Tedavi adlarında tekrarları önlemek ve daha net gruplama sağlamak.

### 5.6 Tedavi Süresi (TedaviSuresi)

- Veriler string formatından temizlenerek sadece sayı kısmı bırakıldı (örn. “15 Seans” → 15).
- Numerik formata dönüştürüldü.

**Amaç:** Tedavi süresini doğrudan sayısal analizlerde kullanabilmek.

### 5.7 Uygulama Yerleri (UygulamaYerleri)

- Veriler küçük harfe çevrildi ve boşluklar temizlendi.
- Farklı yazımlar normalize edildi (ör. “sağ el bilek bölgesi” → “el bileği sağ”).
- Virgülle ayrılan çoklu değerler liste haline getirildi.

**Amaç:** Uygulama bölgelerini daha düzenli hale getirip analize uygun formata dönüştürmek.

## 5.8 Uygulama Süresi (UygulamaSuresi)

- “Dakika” ifadesi silinerek yalnızca sayı kısmı bırakıldı.
- Numerik formata dönüştürüldü.

**Amaç:** Uygulama süresini doğrudan sayısal analizlerde kullanabilmek.

## 5.9 Anlamsız Verilerin Temizlenmesi

- Metinsel sütunlarda “xx”, “deneme”, “test”, “123” gibi anlamsız veriler tespit edilip **NaN** değerine çevrildi.

**Amaç:** Veri setinde analiz sonuçlarını bozabilecek alakasız girdilerin temizlenmesi.

## Bu Adımın Genel Sonucu

Yapılan tüm bu işlemler sayesinde:

- Eksik ve tutarsız veriler minimize edildi,
- String tipinde olup aslında sayısal olan sütunlar dönüştürüldü,
- Çok değerli sütunlar liste haline getirildi,
- Tekrar eden yazım hataları düzeltildi.

Böylece veri seti, hem **EDA görselleştirmeleri** hem de **pipeline tabanlı preprocessing** için uygun hale getirilmiştir.

## 6. EDA (Exploratory Data Analysis) ve Eksik Veri Analizi

### 6.1 Eksik Veri Analizi

Veri setinde eksik değerlerin dağılımı **df.isnull().sum()** fonksiyonu ile incelenmiştir. Çıktıya göre özellikler:

- **KanGrubu, Cinsiyet, KronikHastalik, Alerji, Tanilar, Bolum ve UygulamaYerleri** sütunlarında eksik değerler olduğu tespit edilmiştir.

Eksik değerlerin dağılımı görselleştirmek için **missingno** kütüphanesi kullanılmıştır:

- **Eksik veri matrisi (msno.matrix):** Eksik değerlerin hangi satırlarda yoğunlaştığını görsel olarak ortaya koymuştur.
- **Eksik veri bar grafiği (msno.bar):** Hangi sütunda ne kadar eksik değer bulunduğu bar grafik ile gösterilmiştir.

Bu görselleştirmeler, doldurma stratejilerinin belirlenmesine yardımcı olmuştur.

## 6.2 Eksik Değerlerin Doldurulması

Eksik değerleri doldurmak için farklı stratejiler uygulanmıştır:

- **KanGrubu sütunu:** Veri setinden çıkarılmıştır. Eksik değer oranı yüksek olduğu ve analizde doğrudan kullanılmayacağı düşünülmüştür.
- **Cinsiyet:** En sık görülen kategori (**mode**) ile doldurulmuştur.
- **KronikHastalik:** Eksik değerler “**Yok**” olarak atanmıştır.
- **Alerji:** Eksik değerler “**Yok**” olarak atanmıştır.
- **Tanilar:** Eksik tanılar, aynı **TedaviAdi**'na sahip diğer satırlardan en sık tekrar eden değerle doldurulmuştur. Eğer eşleşme bulunmazsa “**Tanı yok**” olarak işaretlenmiştir.
- **UygulamaYerleri:** Eksik olan değerler, aynı **Tanilar** sütununa sahip diğer satırlardan doldurulmuştur. Eğer uygun eşleşme bulunmazsa “**Bilinmiyor**” değeri atanmıştır.
- **Bolum:** Eksik olan değerler, en sık görülen değer (**mode**) ile doldurulmuştur.
- **TedaviAdi:** Eksik değerler, aynı **Tanilar** değerine sahip diğer satırlardan doldurulmuş, eşleşme yoksa “**Belirtilmemiş**” atanmıştır.

## 6.3 Liste Sütunları İçin Doldurma

- **UygulamaYerleri\_List:**  
Eksik listeler, aynı tanıya sahip satırlardan eşleştirilerek doldurulmuştur. Böylece eksik bölgeler, en olası uygulama yerleriyle tamamlanmıştır.

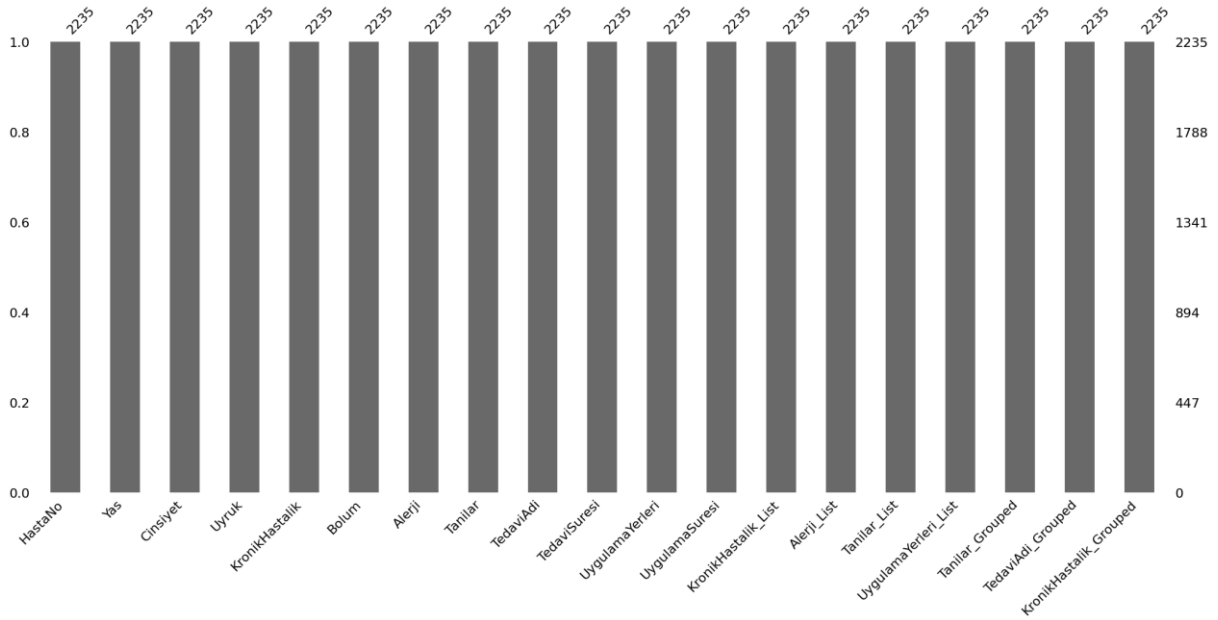
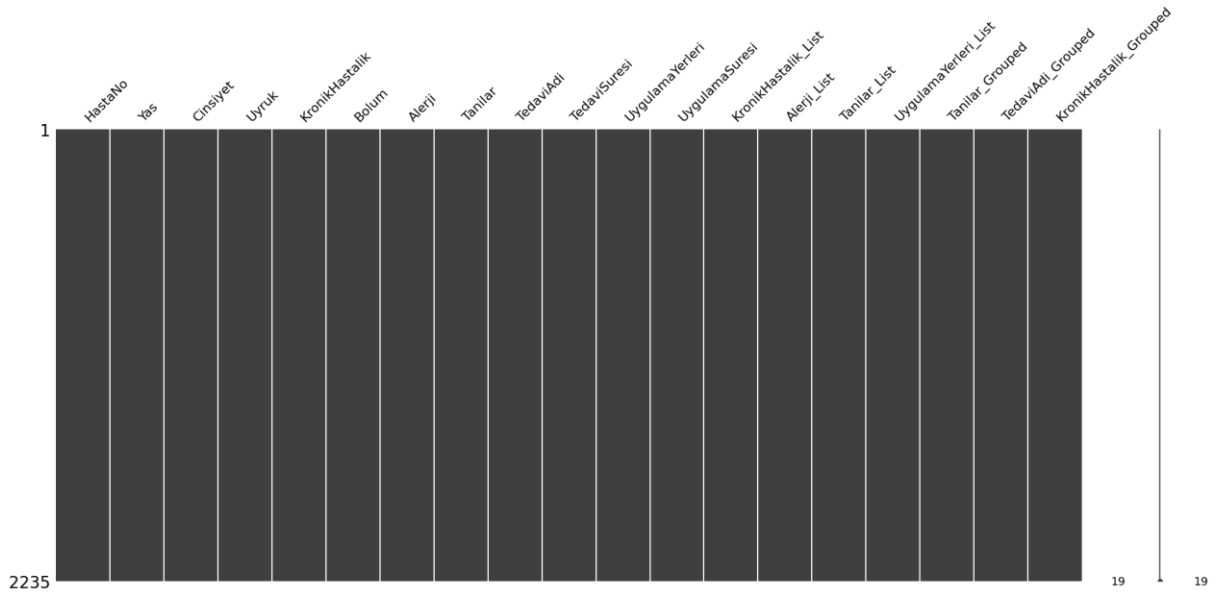
## 6.4 Temizlenmiş Verinin Kaydedilmesi

Tüm bu işlemler sonucunda veri seti temizlenmiş, eksik değerler uygun yöntemlerle doldurulmuş ve işlenebilir hale getirilmiştir.

Temizlenen veri, **pusula\_clean\_ready.xlsx** adıyla kaydedilmiştir.

### Bu aşamanın sonucu:

- Eksik veriler doldurulmuş,
- Tutarsız sütunlar düzeltilmiş,
- Veri seti görselleştirme ve modelleme için hazır hale getirilmiştir.



## 7. EDA Görselleştirmeleri

### 7.1 Yaş Dağılımı

- Histogramda yaşların dağılımı yaklaşık olarak **normal dağılım** göstermektedir.
- En yoğun grup **35–55 yaş aralığıdır**.
- 10 yaş altı çocuklar ve 80 yaş üzeri hastalar düşük oranda temsil edilmektedir.

**Sonuç:** Veri setindeki hastaların büyük çoğunluğu orta yaş grubundandır.



## 7.2 Yaş ve Tedavi Süresi İlişkisi

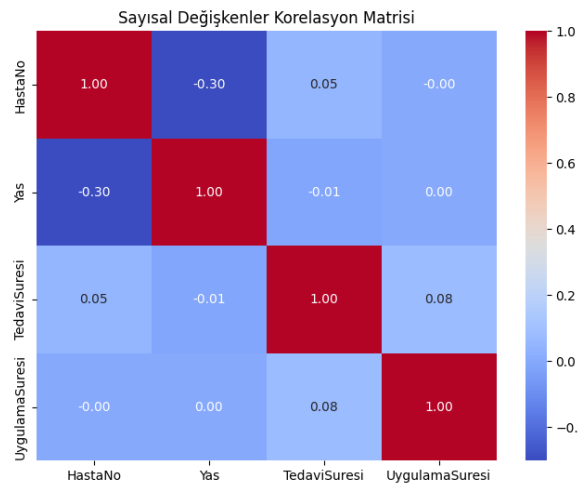
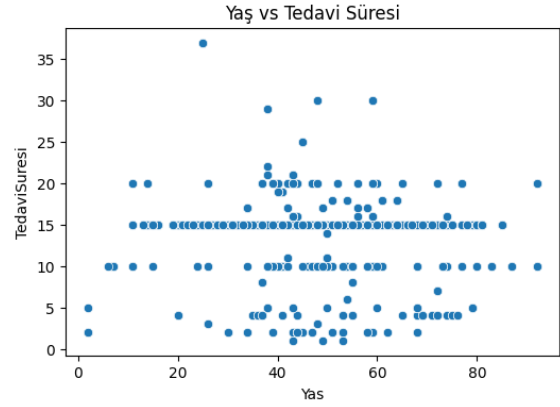
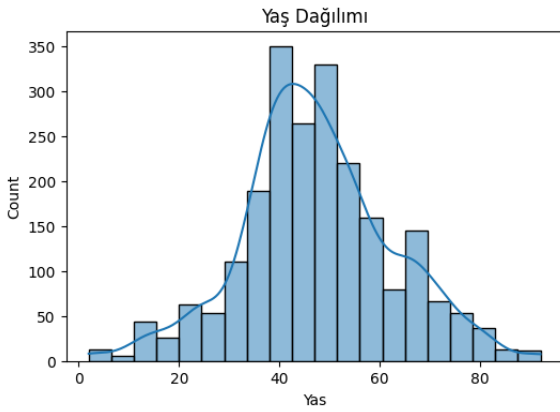
- Scatter plot incelendiğinde, **yaş ile tedavi süresi arasında belirgin bir ilişki görülmemektedir.**
- Tedavi süresi değerleri özellikle **10–20 seans** aralığında yoğunlaşmaktadır.
- 35 seansın üzerindeki tedaviler nadirdir.

**Sonuç:** Tedavi süresi, hastanın yaşından çok **tanı ve tedavi türüne** bağlıdır.

## 7.3 Sayısal Değişkenler Korelasyon Matrisi

- Yaş ↔ Tedavi Süresi:** Korelasyon çok zayıf (-0.01).
- Tedavi Süresi ↔ Uygulama Süresi:** Zayıf pozitif ilişki (0.08).

**Sonuç:** Sayısal değişkenler arasında güçlü bir ilişki bulunmamaktadır. Bu nedenle ileride yapılacak modellemelerde **kategorik sütunlar** (Tanılar, Tedavi Adı, Bölüm vb.) daha belirleyici olacaktır.



## 8. Encoding İşlemleri

Veri setindeki kategorik değişkenler makine öğrenmesi algoritmaları tarafından işlenebilmesi için **OneHotEncoder** yöntemi ile sayısal forma dönüştürülmüştür.

### 8.1 Küçük Kardinaliteli Kategoriler

- **Cinsiyet, Uyruk, Bolum** sütunları az sayıda benzersiz kategoriye sahiptir.
- Bu sütunlar **OneHotEncoder** ile dönüştürülmüş, ilk kategori **drop="first"** parametresiyle atılmıştır. Böylece “dummy variable trap” engellenmiştir.
- Elde edilen sütun örnekleri: Cinsiyet\_Kadın, Uyruk\_Türkiye, Bolum\_Fiziksel Tıp ve Rehabilitasyon vb.

### 8.2 Büyük Kardinaliteli Kategoriler

- **Tanilar, TedaviAdi, KronikHastalik** sütunlarında çok sayıda benzersiz değer bulunmaktadır.
- Bu sütunlar doğrudan encode edilirse çok fazla sütun oluşacaktı.
- Bunun yerine:
  - En sık görülen **ilk 20 kategori** korunmuş,
  - Diğer tüm kategoriler **“Diğer”** etiketi altında toplanmıştır.
- Bu gruplama sonrası OneHotEncoder ile encode edilmiştir.

### 8.3 Encode Edilmiş Verilerin Birleştirilmesi

- Küçük ve büyük kategoriler için üretilen encode edilmiş sütunlar birleştirilmiştir.
- Orijinal kategorik sütunlar (HastaNo, KronikHastalik, Alerji, Tanilar, TedaviAdi, UygulamaYerleri) ve listeler (\*\_List sütunları) çıkarılmıştır.
- Sonuçta **df\_manual\_encoded** isimli, yalnızca numerik sütunlardan oluşan veri seti elde edilmiştir.

### 8.4 Sonuç

- Encode edilmiş veri setinin boyutu: **(2235, 87)**
- Yani 2235 gözlem (satır) ve 87 özellik (sütun) bulunmaktadır.
- Bu adım sayesinde:
  - Veriler modele hazır hale gelmiş,
  - Büyük kategoriler boyut patlamasına yol açmadan işlenmiş,
  - Küçük kategoriler doğrudan modele uygun sayısal formata dönüştürülmüştür.

## 9. Scaling (Ölçeklendirme)

### 9.1 Kullanılan Yöntem

Sayısal değişkenlerin farklı ölçeklerde olması, makine öğrenmesi algoritmalarının performansını olumsuz etkileyebilir. Bu nedenle tüm sayısal sütunlar **StandardScaler** ile ölçeklendirilmiştir.

- **StandardScaler:** Her değişkenin ortalamasını 0, standart sapmasını 1 olacak şekilde dönüştürür.
- Bu işlem sayesinde farklı büyüklüklerdeki değişkenler aynı ölçeğe getirilir.

### 9.2 Ölçeklenen Sütunlar

- **Yas**
- **TedaviSuresi**
- **UygulamaSuresi**

### 9.3 Sonuçlar

- Ölçekleme sonrası veriler standart normal dağılıma uygun hale gelmiştir.
- Örneğin:
  - Yaş sütunu → [-1.27, +0.83] aralığında değerler,
  - Tedavi Süresi sütunu → [-2.56, +0.11] gibi normalize edilmiş değerler,
  - Uygulama Süresi sütunu → [-1.84, +0.54] gibi değerler üretmiştir.
- **Manuel encode + scaling sonrası veri boyutu: (2235, 80)**

### 9.4 Bu Adımın Önemi

- Ölçeklendirme ile tüm sayısal sütunlar aynı seviyeye çekilmiştir.
- Ayrıca model eğitim süresini kısaltır ve daha dengeli ağırlıklandırma yapılmasına imkan tanır.

**Sonuç:** Artık veri seti, hem encoding hem de scaling işlemleri tamamlanmış şekilde, pipeline'a alınmaya ve modellemeye hazır hale getirilmiştir.

## 10. Pipeline Yapısı

Veri ön işleme adımlarının tekrar edilebilir ve otomatik bir şekilde yapılabilmesi için **Pipeline** yapısı oluşturulmuştur. Bu yapı sayesinde manuel adımlarla yapılan işlemler tek bir akış halinde uygulanabilmektedir.

### 10.1 Pipeline Adımları

#### 1. Numeric Features (Sayısal Değişkenler):

- Sütunlar: Yas, TedaviSuresi, UygulamaSuresi
- Eksik değer doldurma: Median stratejisi
- Ölçekleme: StandardScaler

#### 2. Small Categorical Features (Küçük Kategoriler):

- Sütunlar: Cinsiyet, Uyruk, Bolum
- Eksik değer doldurma: Most frequent (en sık görülen değer)
- Encoding: OneHotEncoder (drop='first')

#### 3. Big Categorical Features (Büyük Kategoriler):

- Sütunlar: Tanilar, TedaviAdi, KronikHastalik (ilk 20 kategori + "Diğer")
- Eksik değer doldurma: Most frequent
- Encoding: OneHotEncoder (drop='first')


#### 4. ColumnTransformer:

- Yukarıdaki üç grup birleştirilmiştir.
- Gereksiz sütunlar (HastaNo, metinsel listeler vb.) dışarıda bırakılmıştır.

#### 5. Pipeline:

- Tüm işlem adımları Pipeline içine alınmış ve tek seferde uygulanabilir hale getirilmiştir

### 10.2 Çıktılar

- Pipeline başarıyla uygulanmıştır 
- **Sonuç veri seti boyutu:** (2235, 77)
- İlk 5 gözlem örneği:

### 10.3 Bu Adımın Önemi

- Pipeline sayesinde veri ön işleme adımları **otomatikleştirilmiş** oldu.
- Aynı veri setine veya farklı bir veri setine tekrar uygulandığında, aynı dönüşümler sorunsuz şekilde yapılacaktır.
- Manuel encode + scaling ile pipeline sonucu karşılaştırıldığında aynı sütun sayısı ve aynı çıktı elde edilmiştir. Bu da pipeline'ın doğruluğunu göstermektedir.

## 12. Manuel Encode vs Pipeline Karşılaştırması

Manuel olarak yapılan encode + scaling adımları ile Pipeline yapısının çıktıları karşılaştırılmıştır.

### 12.1 Sütun Sayıları

- **Manuel encode sütun sayısı: 77**
- **Pipeline sütun sayısı: 77**

### 12.2 Sütunlar Arasındaki Farklar

- Manuel encode'de olup Pipeline'da olmayan sütunlar: **Ø (boş küme)**
- Pipeline'da olup Manuel encode'de olmayan sütunlar: **Ø (boş küme)**

### 12.3 Yorum

- Hem manuel yöntem hem de Pipeline, aynı sayıda sütun ve aynı içerikte sonuç üretmiştir.
- Bu durum, Pipeline'ın manuel işlemleri **doğru şekilde tekrar edebildiğini** ve **otomatik bir çözüm** sunduğunu kanıtlamaktadır.
- Pipeline yaklaşımı, manuel yöntemle kıyasla **daha güvenilir, tekrar edilebilir ve ölçeklenebilir** bir çözüm sağlamaktadır.

### 13. Sonuç ve Değerlendirme

Bu proje kapsamında fiziksel tıp ve rehabilitasyon alanına ait hasta verisi üzerinde **EDA (Exploratory Data Analysis)** ve **veri ön işleme (preprocessing)** adımları uygulanmıştır. Projenin temel amacı, veriyi **makine öğrenmesine hazır hale getirmek** olmuştur.

#### 13.1 Öğrenilenler

- **Eksik veri doldurma:** Mode, median ve ilişkili sütunlara göre doldurma stratejileri başarıyla uygulanmıştır.
- **Veri temizleme:** Anlamsız değerler silinmiş, yazım hataları düzeltilmiş, çoklu değerler liste formatına dönüştürülmüştür.
- **Encoding:**
  - Küçük kategoriler (örn. Cinsiyet, Uyruk) doğrudan OneHotEncoder ile dönüştürülmüştür.
  - Büyük kategoriler (örn. Tanılar, Tedavi Adı) önce gruplandırılmış, ardından encode edilmiştir.
- **Scaling:** Tüm sayısal sütunlar StandardScaler ile ölçeklendirilmiş, böylece algoritmalara hazır hale getirilmiştir.
- **Pipeline:** Manuel yapılan işlemler Pipeline yapısı ile otomatik hale getirilmiş ve aynı sonuçlar elde edilmiştir.

#### 13.2 Sonuçlar

- Temizlenmiş, encode edilmiş ve ölçeklendirilmiş veri seti elde edilmiştir.
- Çıktılar başarıyla kaydedilmiştir:
  - pusula\_clean\_ready.xlsx
  - manual\_encoded\_ready.csv
  - pipeline\_ready.xlsx
- Manuel yöntem ile Pipeline karşılaştırılmış, **ikisinin de aynı sonuçları verdiği** görülmüştür (77 sütun, aynı içerik).

#### Genel Değerlendirme:

Proje başarıyla tamamlanmıştır. Elde edilen sonuçlar, preprocessing adımlarının hem manuel hem de pipeline ile doğru şekilde gerçekleştirildiğini göstermektedir. Pipeline yaklaşımı, sürecin tekrarlanabilirliğini ve ölçeklenebilirliğini sağlamış, gelecekte modelleme aşamasına doğrudan geçilebilecek bir veri seti sunmuştur.