RUKİYE KARAATLI

[rukiyekaraatli@gmail.com](mailto:rukiyekaraatli@gmail.com)

**Kütüphanelerin İçe Aktarılması**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

from sklearn.preprocessing import StandardScaler

**Veri Setini Yükleme ve İnceleme**

Kod:

# Verisetini yükleyelim.

sample\_data = pd.read\_excel("side\_effect\_data 1.xlsx")

# İlk satırlara göz atmak için:

print(sample\_data.head())

# Veri yapısını incelemek için:

print(sample\_data.info())

# Eksik veri kontrolü sağalamak için:

print(sample\_data.isnull().sum())

Çıktı:

Kullanici\_id Cinsiyet Dogum\_Tarihi Uyruk Il \

0 107 Male 1960-03-01 Turkiye Canakkale

1 140 Male 1939-10-12 Turkiye Trabzon

2 2 Female 1976-12-17 Turkiye Canakkale

3 83 Male 1977-06-17 Turkiye Adana

4 7 Female 1976-09-03 Turkiye Izmir

Ilac\_Adi Ilac\_Baslangic\_Tarihi Ilac\_Bitis\_Tarihi \

0 trifluoperazine 2022-01-09 2022-03-04

1 fluphenazine hcl 2022-01-09 2022-03-08

2 warfarin sodium 2022-01-11 2022-03-12

3 valproic acid 2022-01-04 2022-03-12

4 carbamazepine extended release 2022-01-13 2022-03-06

Yan\_Etki Yan\_Etki\_Bildirim\_Tarihi Alerjilerim \

0 Kabizlik 2022-02-19 18:28:43 Ceviz

1 Yorgunluk 2022-02-03 20:48:17 Toz

2 Carpinti 2022-02-04 05:29:20 Muz

3 Sinirlilik 2022-02-08 01:01:21 Pancar

4 Agizda Farkli Bir Tat 2022-02-12 05:33:06 NaN

Kronik Hastaliklarim Baba Kronik Hastaliklari \

0 Hipertansiyon, Kan Hastaliklari Guatr, Hipertansiyon

1 NaN Guatr, Diger

2 Kalp Hastaliklari, Diyabet Diyabet, KOAH

3 Diyabet, Diger Kalp Hastaliklari, Diger

4 Diyabet, Kalp Hastaliklari Alzheimer, Hipertansiyon

Anne Kronik Hastaliklari Kiz Kardes Kronik Hastaliklari \

0 KOAH Kemik Erimesi, Kalp Hastaliklari

1 Hipertansiyon, Kalp Hastaliklari

2 Kemik Erimesi, Diyabet Diyabet, Kemik Erimesi

3 NaN Astim

4 Kan Hastaliklari, Kemik Erimesi Diyabet, Diger

Erkek Kardes Kronik Hastaliklari Kan Grubu Kilo Boy

0 Kemik Erimesi, Guatr B RH- 103.0 191.0

1 KOAH, Diyabet NaN 81.0 181.0

2 Diger B RH- 93.0 158.0

3 Kalp Hastaliklari, Kanser AB RH- NaN 165.0

4 Alzheimer, Hipertansiyon AB RH- 99.0 172.0

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2357 entries, 0 to 2356

Data columns (total 19 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Kullanici\_id 2357 non-null int64

1 Cinsiyet 1579 non-null object

2 Dogum\_Tarihi 2357 non-null datetime64[ns]

3 Uyruk 2357 non-null object

4 Il 2130 non-null object

5 Ilac\_Adi 2357 non-null object

6 Ilac\_Baslangic\_Tarihi 2357 non-null datetime64[ns]

7 Ilac\_Bitis\_Tarihi 2357 non-null datetime64[ns]

8 Yan\_Etki 2357 non-null object

9 Yan\_Etki\_Bildirim\_Tarihi 2357 non-null datetime64[ns]

10 Alerjilerim 1873 non-null object

11 Kronik Hastaliklarim 1965 non-null object

12 Baba Kronik Hastaliklari 2201 non-null object

13 Anne Kronik Hastaliklari 2140 non-null object

14 Kiz Kardes Kronik Hastaliklari 2260 non-null object

15 Erkek Kardes Kronik Hastaliklari 2236 non-null object

16 Kan Grubu 2010 non-null object

17 Kilo 2064 non-null float64

18 Boy 2243 non-null float64

dtypes: datetime64[ns](4), float64(2), int64(1), object(12)

memory usage: 350.0+ KB

None

Kullanici\_id 0

Cinsiyet 778

Dogum\_Tarihi 0

Uyruk 0

Il 227

Ilac\_Adi 0

Ilac\_Baslangic\_Tarihi 0

Ilac\_Bitis\_Tarihi 0

Yan\_Etki 0

Yan\_Etki\_Bildirim\_Tarihi 0

Alerjilerim 484

Kronik Hastaliklarim 392

Baba Kronik Hastaliklari 156

Anne Kronik Hastaliklari 217

Kiz Kardes Kronik Hastaliklari 97

Erkek Kardes Kronik Hastaliklari 121

Kan Grubu 347

Kilo 293

Boy 114

dtype: int64

Yorum:

Bu veri setinde, özellikle kronik hastalıklar, cinsiyet, kan grubu, boy ve kilo gibi bazı kritik bilgilerin eksik olduğunu görüyorum. Eksik verilerin bu kadar yüksek oranda olduğu durumlarda, eksik değerleri doldurma (imputation), eksik verileri çıkarma veya farklı tekniklerle veri kalitesini artırma gibi adımlar uygulamak gerekir. Özellikle makine öğrenmesi modelleri için eksik değerlerle başa çıkmak önemli olacaktır.

**Veri Görselleştirme**

Kod:

# Sayısal değişkenlerin dağılımını görmek için histogramları kullanalım.

sample\_data.hist(bins=50, figsize=(20,15))

plt.show()

Çıktı: metin, diyagram, çizgi, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yorum:

Genel olarak sayısal değişkenlerde dengeli bir dağılım gözlemledim.Ama genel bir bakış atmak gerekirse :

100 den fazla kişi 1970-1980 tarihleri arasında doğmuş,

En yoğun yan etki bildirim tarihi 02-19 olarak gözüküyor.

80-85 kg kilo ya sahip kişi sayısı fazla ve 180-185 cm arası boya sahip insan yoğunlukta görülüyor.

Kod:

# Kategorik değişkenler için bar grafikleri kullanalım.

categorical\_columns = ['Cinsiyet', 'Il', 'Kan Grubu']

for col in categorical\_columns:

    plt.figure(figsize=(8, 4))

    sns.countplot(data=sample\_data, x=col)

    plt.title(f'{col} Dağılımı')

    plt.show()

Çıktı: metin, ekran görüntüsü, dikdörtgen, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yorum:

Kadın sayısı erkeğe göre fazla, 200 den fazla kişi Adana yaşıyor ve en yoğun kan grubu AB RH- olarak görülüyor.

Kod:

# Isı haritası ile korelasyon analizi yapalım.

# Önce Sayısal sütunları seçelim

numerical\_data = sample\_data.select\_dtypes(include=['float64', 'int64'])

# Korelasyon matrisi oluşturma ve ısı haritasını çizme

plt.figure(figsize=(10,8))

sns.heatmap(numerical\_data.corr(), annot=True, cmap='coolwarm')

plt.title("Korelasyon Matrisi")

plt.show()

Çıktı: metin, ekran görüntüsü, dikdörtgen, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yorum:

**Eksik Verilerin İşlenmesi**

Kod:

# Kategorik değişkenlerde eksik verileri en sık görülen değer ile dolduralım

imputer\_cat = SimpleImputer(strategy='most\_frequent')

sample\_data[categorical\_columns] = imputer\_cat.fit\_transform(sample\_data[categorical\_columns])

# Sayısal değişkenlerde eksik verileri ortalama ile dolduralım

numerical\_columns = ['Kilo', 'Boy']

imputer\_num = SimpleImputer(strategy='mean')

sample\_data[numerical\_columns] = imputer\_num.fit\_transform(sample\_data[numerical\_columns])

Yorum:

Kategorik Değişkenler için Eksik Verilerin Doldurulması:

SimpleImputer sınıfı, eksik verileri doldurmak için kullanılan bir araçtır. Burada, strategy='most\_frequent' parametresi ile kategorik değişkenlerde eksik veriler en sık görülen değer ile doldurulmaktadır. sample\_data[categorical\_columns], kategorik değişkenlerin bulunduğu DataFrame'in bir dilimidir. fit\_transform metodu, bu dilimdeki eksik verileri doldurmak için en sık görülen değeri hesaplar ve eksik verileri bu değer ile değiştirir.

Sayısal Değişkenler için Eksik Verilerin Doldurulması:

numerical\_columns listesi, sayısal değişkenlerin adlarını içermektedir. Burada, 'Kilo' ve 'Boy' değişkenleri yer almaktadır. SimpleImputer yine kullanılarak, bu sayısal değişkenlerde eksik veriler ortalama (strategy='mean') ile doldurulmaktadır. sample\_data[numerical\_columns] kısmı, sayısal değişkenlerin bulunduğu DataFrame'in bir dilimidir ve fit\_transform metodu ile eksik veriler ortalama ile doldurulmaktadır. Bu işlem, veri setindeki eksik verileri temizlemek ve daha sonrasında yapılacak analizler veya modelleme için verilerin hazır olmasını sağlamak amacıyla yapılmaktadır.

**Kategorik Değişkenlerin Kodlanması**

Kod: # Cinsiyet değişkenini LabelEncoder ile kodlama yapalım

le = LabelEncoder()

sample\_data['Cinsiyet'] = le.fit\_transform(sample\_data['Cinsiyet'])

# Mevcut sütun adlarını kontrol edelim

print(sample\_data.columns)

# Kan Grubu ve diğer çok sınıflı kategorik değişkenler için OneHotEncoder kullanımını gerçekleştirelim

ohe\_columns = ['Kan Grubu', 'Ilac\_Adi', 'Yan\_Etki']

# Hatalı sütunları kontrol edip doğru isimlerle güncelleyelim

if all(col in sample\_data.columns for col in ohe\_columns):

    ohe = OneHotEncoder(sparse\_output=False, drop='first')  # sparse parametresi güncellendi

    ohe\_encoded = pd.DataFrame(ohe.fit\_transform(sample\_data[ohe\_columns]), columns=ohe.get\_feature\_names\_out(ohe\_columns))

    # Orijinal sütunları düşürüp kodlanmış yeni sütunları ekleyelim

    sample\_data = sample\_data.drop(columns=ohe\_columns).reset\_index(drop=True)

    sample\_data = pd.concat([sample\_data, ohe\_encoded], axis=1)

else:

    print("Eksik sütunlar: ", [col for col in ohe\_columns if col not in sample\_data.columns])

Çıktı: Index(['Kullanici\_id', 'Cinsiyet', 'Dogum\_Tarihi', 'Uyruk', 'Il',

'Ilac\_Baslangic\_Tarihi', 'Ilac\_Bitis\_Tarihi',

'Yan\_Etki\_Bildirim\_Tarihi', 'Alerjilerim', 'Kronik Hastaliklarim',

...

'Yan\_Etki\_Kabizlik', 'Yan\_Etki\_Karin Agrisi', 'Yan\_Etki\_Kas Agrisi',

'Yan\_Etki\_Mide Bulantisi', 'Yan\_Etki\_Sinirlilik',

'Yan\_Etki\_Tansiyon Dusuklugu', 'Yan\_Etki\_Tansiyon Yukselme',

'Yan\_Etki\_Terleme', 'Yan\_Etki\_Uykululuk Hali', 'Yan\_Etki\_Yorgunluk'],

dtype='object', length=194)

Eksik sütunlar: ['Kan Grubu', 'Ilac\_Adi', 'Yan\_Etki']

**Sayısal Verilerin Normalizasyonu/Standardizasyonu**

Kod:

# Sayısal verileri standardizasyon (z-skoru) işlemini gerçekleştirelim

scaler = StandardScaler()

sample\_data[numerical\_columns] = scaler.fit\_transform(sample\_data[numerical\_columns])

# Son veri setine göz atalım

print(sample\_data.head())

Çıktı.

Kullanici\_id Cinsiyet Dogum\_Tarihi Uyruk Il \

0 107 1 1960-03-01 Turkiye Canakkale

1 140 1 1939-10-12 Turkiye Trabzon

2 2 0 1976-12-17 Turkiye Canakkale

3 83 1 1977-06-17 Turkiye Adana

4 7 0 1976-09-03 Turkiye Izmir

Ilac\_Baslangic\_Tarihi Ilac\_Bitis\_Tarihi Yan\_Etki\_Bildirim\_Tarihi \

0 2022-01-09 2022-03-04 2022-02-19 18:28:43

1 2022-01-09 2022-03-08 2022-02-03 20:48:17

2 2022-01-11 2022-03-12 2022-02-04 05:29:20

3 2022-01-04 2022-03-12 2022-02-08 01:01:21

4 2022-01-13 2022-03-06 2022-02-12 05:33:06

Alerjilerim Kronik Hastaliklarim ... Yan\_Etki\_Kabizlik \

0 Ceviz Hipertansiyon, Kan Hastaliklari ... 1.0

1 Toz NaN ... 0.0

2 Muz Kalp Hastaliklari, Diyabet ... 0.0

3 Pancar Diyabet, Diger ... 0.0

4 NaN Diyabet, Kalp Hastaliklari ... 0.0

Yan\_Etki\_Karin Agrisi Yan\_Etki\_Kas Agrisi Yan\_Etki\_Mide Bulantisi \

0 0.0 0.0 0.0

1 0.0 0.0 0.0

2 0.0 0.0 0.0

3 0.0 0.0 0.0

4 0.0 0.0 0.0

Yan\_Etki\_Sinirlilik Yan\_Etki\_Tansiyon Dusuklugu \

0 0.0 0.0

1 0.0 0.0

2 0.0 0.0

3 1.0 0.0

4 0.0 0.0

Yan\_Etki\_Tansiyon Yukselme Yan\_Etki\_Terleme Yan\_Etki\_Uykululuk Hali \

0 0.0 0.0 0.0

1 0.0 0.0 0.0

2 0.0 0.0 0.0

3 0.0 0.0 0.0

4 0.0 0.0 0.0

Yan\_Etki\_Yorgunluk

0 0.0

1 1.0

2 0.0

3 0.0

4 0.0

[5 rows x 194 columns]

YORUM:

Bu veri seti, kullanıcıların ilaç kullanımı ile ilgili çeşitli analizler yapmak için uygundur.Örnek olarak;Yan etkilerin analizi,Cinsiyet veya diğer demografik özelliklere göre yan etki deneyimlerinin karşılaştırılması,Kullanıcıların sağlık geçmişleri ile yan etkiler arasındaki ilişkiler incelenebilir.

MAKİNE ÖĞRENMESİ PROJESİ GELİŞTİRECEK OLSAM ;

**Yan Etki Tahmini Modeli**

* **Amaç:** Kullanıcıların belirli ilaçları kullanırken yaşayabilecekleri yan etkileri tahmin etmek.
* **Aşamalar:** Veri ön işleme, öznitelik seçimi, sınıflandırma algoritmaları ile model geliştirme ve değerlendirme.

**Kullanıcı Profiline Dayalı Öneri Sistemi**

* **Amaç:** Kullanıcıların geçmiş sağlık verilerine dayanarak uygun ilaçları önermek.
* **Aşamalar:** Veri ön işleme, öneri modelleme ve kullanıcı geri bildirimleri toplama.

Bu modelleri kullanabilirdim.