Sümeyye Bakırdal
sumeyyebakirdal@gmail.com

Keşifsel Veri Analizi ve Özellik Mühendisliği

Bu belge veri setini analiz etme ve modelleme için hazırlama sürecini özetlemektedir. Python ve çeşitli kütüphaneler kullanılarak gerçekleştirilen keşifsel veri analizi (EDA) ve veri ön işleme adımları aşağıda açıklanmıştır.

Gerekli Kütüphanelerin İçe Aktarılması ve Veri Setinin Yüklenmesi

Aşağıdaki kod, veri analizi için gerekli kütüphanelerin içe aktarılmasını ve veri setinin yüklenmesini sağlamaktadır.

- **Kütüphaneler:** Numpy, Pandas, Matplotlib, Seaborn, ve Scikit-learn gibi kütüphaneler veri analizi ve görselleştirme için kullanılır.
- **Uyarı Mesajları:** Uyarı mesajları devre dışı bırakılarak daha temiz bir çıktı sağlanır.
- **Pandas Ayarları:** Görselleştirme için veri çerçevesinin görünümü ayarlanır.

Keşifçi Veri Analizi

Aşağıdaki fonksiyon, veri çerçevesinin temel analizini gerçekleştirir.

```
> def check_df(dataframe, head=5): ···
```

Şekil: (2357, 19)

Değişken Türleri: (int64, float64, object,datetime4[ns])

İlk 5 Satır:

```
##################### Head #####################
   Kullanici_id Cinsiyet Dogum_Tarihi    Uyruk        Il                  Ilac_Adi Ilac_Baslangic_Tarihi Ilac_Bitis_Tarihi             Yan_Etki Yan_Etki_Bildirim_Tarihi Alerjilerim  \
0         107     Male   1960-03-01  Turkiye  Canakkale             trifluoperazine           2022-01-09       2022-03-04              Kabizlik      2022-02-19 18:28:43       Ceviz
1         140     Male   1939-10-12  Turkiye    Trabzon               fluphenazine hcl           2022-01-09       2022-03-08             Yorgunluk      2022-02-03 20:48:17         Toz
2           2   Female   1976-12-17  Turkiye  Canakkale               warfarin sodium           2022-01-11       2022-03-12              Carpinti      2022-02-04 05:29:20         Muz
3          83     Male   1977-06-17  Turkiye      Adana                 valproic acid           2022-01-04       2022-03-12            Sinirlilik      2022-02-08 01:01:21      Pancar
4           7   Female   1976-09-03  Turkiye      Izmir  carbamazepine extended release           2022-01-13       2022-03-06  Agizda Farkli Bir Tat      2022-02-12 05:33:06         NaN

            Kronik Hastaliklarim  Baba Kronik Hastaliklari         Anne Kronik Hastaliklari    Kiz Kardes Kronik Hastaliklari Erkek Kardes Kronik Hastaliklari Kan Grubu    Kilo     Boy
0  Hipertansiyon, Kan Hastaliklari      Guatr, Hipertansiyon                             KOAH  Kemik Erimesi, Kalp Hastaliklari            Kemik Erimesi, Guatr     B RH-  103.000 191.000
1                               NaN             Guatr, Diger  Hipertansiyon, Kalp Hastaliklari                                                KOAH, Diyabet      NaN   81.000 181.000
2     Kalp Hastaliklari, Diyabet            Diyabet, KOAH          Kemik Erimesi, Diyabet          Diyabet, Kemik Erimesi                                  Diger     B RH-   93.000 158.000
3                 Diyabet, Diger  Kalp Hastaliklari, Diger                             NaN                                 Astim       Kalp Hastaliklari, Kanser    AB RH-      NaN 165.000
4     Diyabet, Kalp Hastaliklari  Alzheimer, Hipertansiyon  Kan Hastaliklari, Kemik Erimesi                 Diyabet, Diger        Alzheimer, Hipertansiyon    AB RH-   99.000 172.000
```

Son 5 Satır:

```
##################### Tail #####################
      Kullanici_id Cinsiyet Dogum_Tarihi    Uyruk        Il                        Ilac_Adi Ilac_Baslangic_Tarihi Ilac_Bitis_Tarihi             Yan_Etki Yan_Etki_Bildirim_Tarihi    Alerjilerim  \
2352             9      NaN   1957-01-04  Turkiye       NaN  desoximetasone spray, non-aerosol           2022-01-13       2022-03-04               Ishal      2022-02-12 19:13:43  Ispanakgiller
2353           101   Female   2004-11-09  Turkiye    Mersin                olanzapine-fluoxetine           2022-01-02       2022-03-05  Agizda Farkli Bir Tat      2022-02-19 17:39:48       Istiridye
2354           127   Female   1951-11-29  Turkiye    Mersin                           trazodone           2022-01-02       2022-03-12             Yorgunluk      2022-02-03 20:48:17  Deniz Urunler
2355           178     Male   1980-01-30  Turkiye   Kayseri          duloxetine hydrochloride           2022-01-02       2022-03-08              Carpinti      2022-02-04 05:29:20  Sari Kantaron
2356           174   Female   1986-11-07  Turkiye  Istanbul                     valproic acid           2022-01-06       2022-03-06          Istah Artisi      2022-02-17 07:08:01            NaN

     Kronik Hastaliklarim Baba Kronik Hastaliklari         Anne Kronik Hastaliklari Kiz Kardes Kronik Hastaliklari Erkek Kardes Kronik Hastaliklari Kan Grubu    Kilo     Boy
2352                  NaN                                        Astim, Guatr                                                KOAH, Kan Hastaliklari      NaN  50.000 152.000
2353                  NaN     Hipertansiyon, Astim          Astim, Kemik Erimesi        KOAH, Kan Hastaliklari                                Astim, Diyabet    AB RH-     NaN 178.000
2354         Guatr, KOAH         Alzheimer, Diger                                                 Astim, Diyabet        Kalp Hastaliklari, Diger     B RH+  90.000 203.000
2355     Alzheimer, Diger                      NaN        Kalp Hastaliklari, Diger                 Kanser, KOAH                                Astim, KOAH      NaN  90.000 184.000
2356     Alzheimer, Diger           Kanser, Diger  Hipertansiyon, Kan Hastaliklari                                                Hipertansiyon, Diger    AB RH+  79.000 175.000
```

Eksik değerler:

```
##################### NA #####################
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
```

Quantiles:

```
##################### Quantiles #####################
       Kullanici_id                  Dogum_Tarihi         Ilac_Baslangic_Tarihi            Ilac_Bitis_Tarihi      Yan_Etki_Bildirim_Tarihi     Kilo     Boy
count      2357.000                          2357                          2357                         2357                          2357 2064.000 2243.000
mean         97.217  1974-11-25 04:06:12.677131936  2022-01-07 10:47:36.173101312  2022-03-10 16:25:27.365294848  2022-02-10 17:09:30.742044928   80.864  174.638
min           1.000           1939-10-12 00:00:00           2022-01-01 00:00:00           2022-03-02 00:00:00           2022-02-01 04:34:33   50.000  145.000
25%          47.000           1959-02-05 00:00:00           2022-01-04 00:00:00           2022-03-06 00:00:00           2022-02-04 05:29:20   65.000  160.000
50%          97.000           1973-09-09 00:00:00           2022-01-07 00:00:00           2022-03-11 00:00:00           2022-02-09 20:53:54   83.000  176.000
75%         146.000           1992-03-24 00:00:00           2022-01-11 00:00:00           2022-03-15 00:00:00           2022-02-17 07:08:01   96.000  187.000
max         196.000           2011-04-25 00:00:00           2022-01-14 00:00:00           2022-03-19 00:00:00           2022-02-19 21:47:39  110.000  203.000
std          57.017                           NaN                           NaN                          NaN                           NaN   18.635   16.517
```

Değişkenlerin Kategorik ve Sayısal Olarak Ayrılması

Aşağıdaki fonksiyon, veri setindeki değişkenleri kategorik, sayısal ve kardinal kategorik olarak ayırmak için kullanılır. Fonksiyon, her bir değişkenin tipini kontrol ederek ilgili listeleri döndürür.

```
> def grab_col_names(dataframe, cat_th=10, car_th=20):···
```

- `dataframe`: İncelenen veri seti.
- `cat_th`: Kategorik gibi görünen ama sayısal olan değişkenler için eşik değeri.
- `car_th`: Kategorik ama kardinal değişkenler için eşik değeri.

Fonksiyon çalıştırıldığında veri setindeki gözlem ve değişken sayısını, ayrıca her bir kategori için değişken sayısını terminalde görüntüler.

```
Observations: 2357
Variables: 19
cat_cols: 4
num_cols: 7
cat_but_car: 8
num_but_cat: 0
```

Kategorik Değişkenlerin Analizi

Aşağıdaki kod, kategorik değişkenlerin analizini ve görselleştirilmesini sağlar.

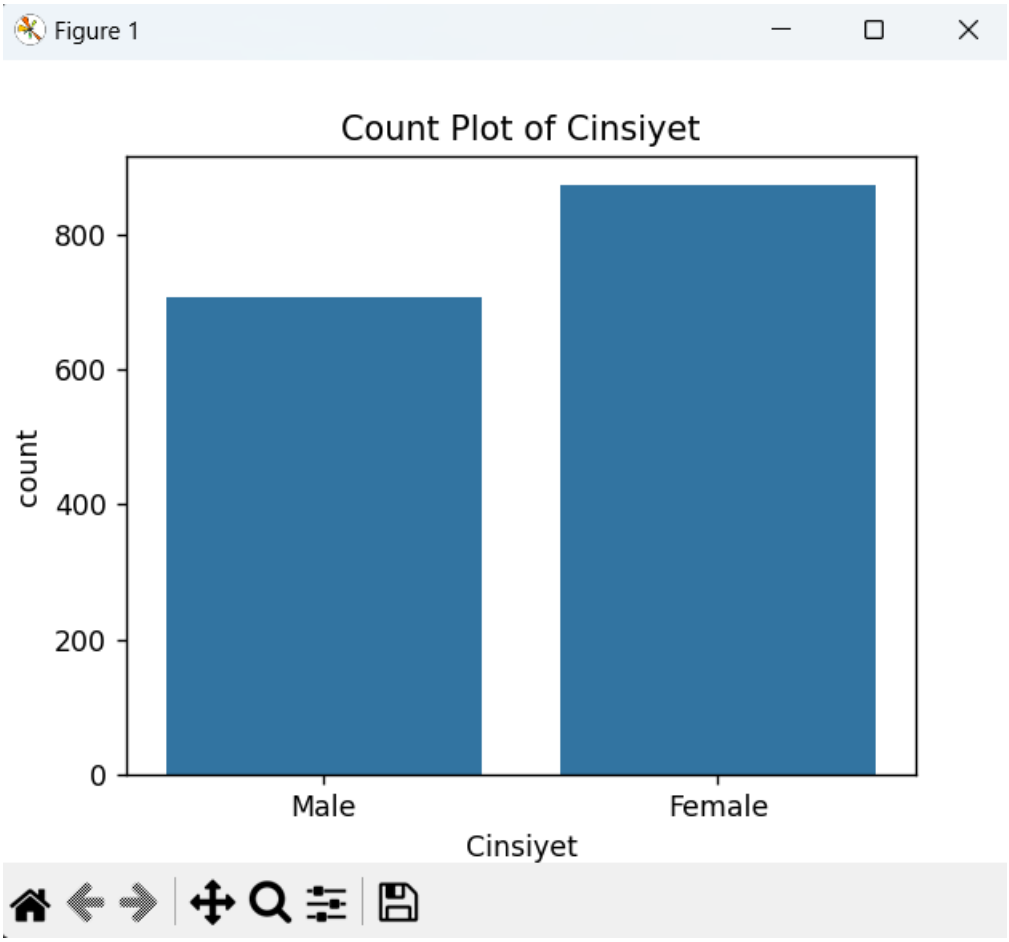
```
> def cat_summary(dataframe, col_name, plot=False):···

> def categorical_summary(dataframe, plot=False):···

  categorical_summary(df, plot=True)
```

- `cat_summary`: Belirli bir kategorik değişkenin değer sayımlarını ve oranlarını gösterir. İsteğe bağlı olarak, değişkenin dağılımını görselleştirir.
- `categorical_summary`: Veri çerçevesindeki tüm kategorik değişkenleri analiz eder.

```
Column: Cinsiyet
          Cinsiyet  Ratio
Cinsiyet
Female         872 36.996
Male           707 29.996
########################################
```



Tarih Zaman Analizi

Veri çerçevesindeki tarih/zaman değişkenlerinin analizini yapar.

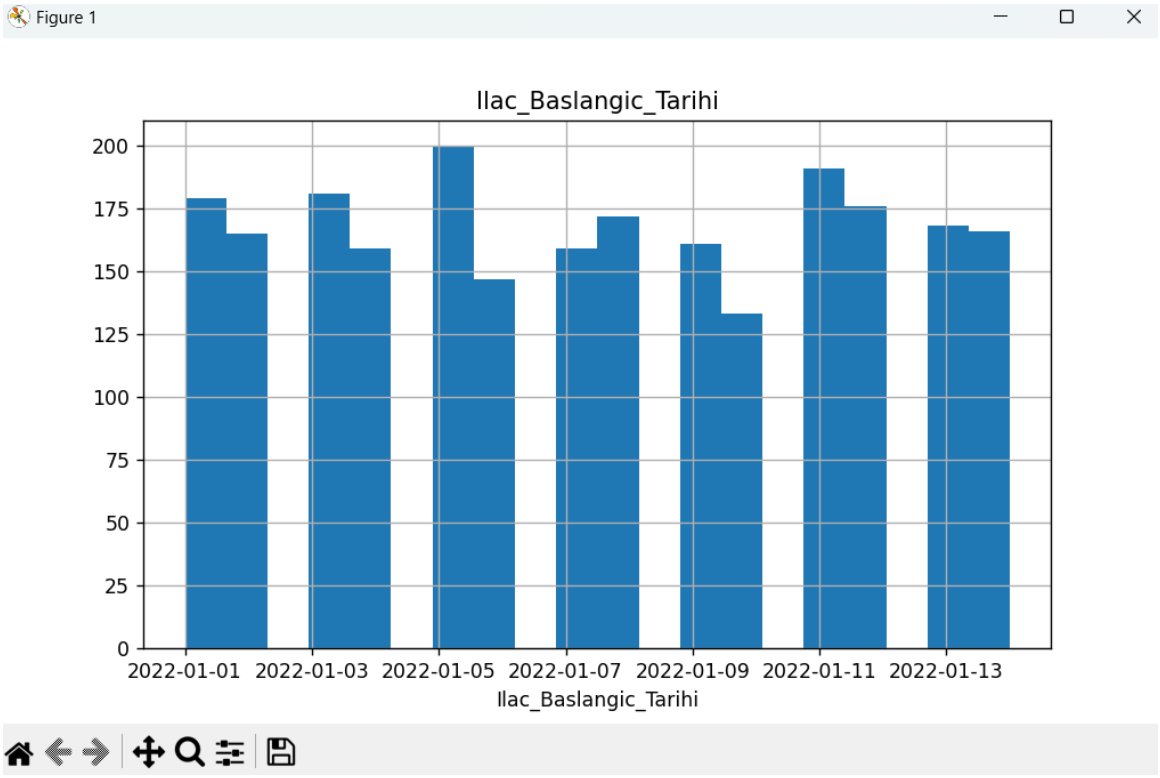
```
> def datetime_analysis(dataframe, datetime_cols):···

  datetime_cols = ['Dogum_Tarihi', 'Ilac_Baslangic_Tarihi', 'Ilac_Bitis_Tarihi', 'Yan_Etki_Bildirim_Tarihi']
  datetime_analysis(df, datetime_cols)
```

Açıklamalar:

- Bu fonksiyon, belirli tarih/zaman değişkenlerinin minimum, maksimum, ortalama, eksik değer sayısı ve benzersiz değer sayısını döndürür.
- Ayrıca, her tarih/zaman değişkeninin histogramını görselleştirir.

```
### Ilac_Baslangic_Tarihi ###
Min: 2022-01-01 00:00:00
Max: 2022-01-14 00:00:00
Mean: 2022-01-07 10:47:36.173101312
Missing Values: 0
Unique Values: 14
```



Sayısal Değişkenlerin Analizi

Sayısal değişkenlerin analizini yapar ve görselleştirir.

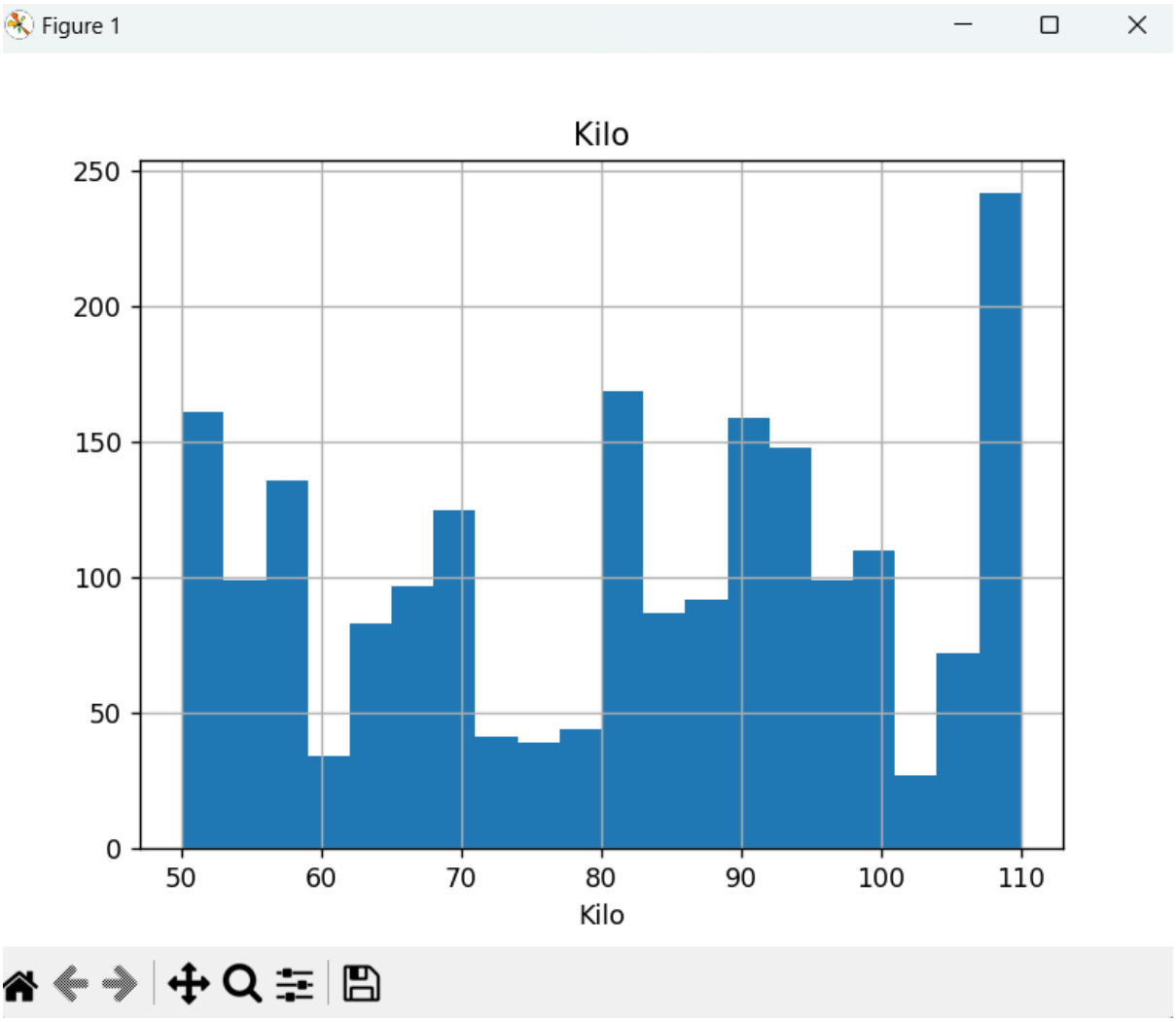
```
def num_summary(dataframe, numerical_col, plot=False):···

for col in num_cols:
    num_summary(df, col, plot=True)
```

Açıklamalar:

- `num_summary`: Sayısal değişkenin temel istatistiklerini hesaplar ve isteğe bağlı olarak dağılımını histogram ile gösterir.
- Sayısal değişkenlerin hedef değişkenle ilişkisini incelemek için `target_summary_with_num` fonksiyonu kullanılır.

```
Name: Kilo, dtype: float64
count   2243.000
mean     174.638
std       16.517
min      145.000
5%       147.000
10%      150.000
20%      158.000
30%      163.000
40%      169.000
50%      176.000
60%      181.000
70%      185.000
80%      189.000
90%      197.000
95%      201.000
99%      203.000
max      203.000
```



Sayısal Değişkenlerin Hedefe Göre Analizi

Sayısal değişkenlerin, belirli bir hedef değişkenle ("Yan_Etki") ilişkisini incelemek için aşağıdaki kod kullanılmıştır.

```
> def target_summary_with_num(dataframe, target, numerical_col):···

  for col in num_cols:
      target_summary_with_num(df, "Yan_Etki", col)
```

Açıklamalar:

- `target_summary_with_num`: Belirli bir sayısal değişkenin ortalamasını hedef değişkene göre gruplandırarak gösterir. Bu, değişkenin hedefle nasıl bir ilişki içinde olduğunu anlamak için faydalıdır.

```
                          Ilac_Baslangic_Tarihi
Yan_Etki
Agizda Farkli Bir Tat 2022-01-07 07:26:27.772925696
Az Uyuma              2022-01-07 16:40:51.063829760
Bas Agrisi            2022-01-08 00:20:16.901408512
Bulanti               2022-01-08 16:07:30.000000000
Carpinti              2022-01-06 20:20:00.000000000
Deride Morarma        2022-01-07 18:19:38.181818112
Gec Bosalma           2022-01-06 10:40:00.000000000
Gormede Bulaniklik    2022-01-07 09:10:35.294117632
Gucsuzluk             2022-01-07 10:40:00.000000000
Huzursuzluk           2022-01-07 10:40:00.000000000
Ishal                 2022-01-07 10:45:31.034482688
Istah Artisi          2022-01-07 15:12:40.563380224
Kabizlik              2022-01-06 19:41:32.307692288
Karin Agrisi          2022-01-07 12:31:18.260869632
Kas Agrisi            2022-01-07 21:06:12.413793024
Mide Bulantisi        2022-01-07 15:00:00.000000000
Sinirlilik            2022-01-06 18:04:26.666666752
Tansiyon Dusuklugu    2022-01-08 12:58:22.702702592
Tansiyon Yukselme     2022-01-07 15:57:53.127753216
Terleme               2022-01-07 10:23:30.309278464
Uykululuk Hali        2022-01-07 20:07:03.529411840
Yorgunluk             2022-01-07 04:00:00.000000000
```

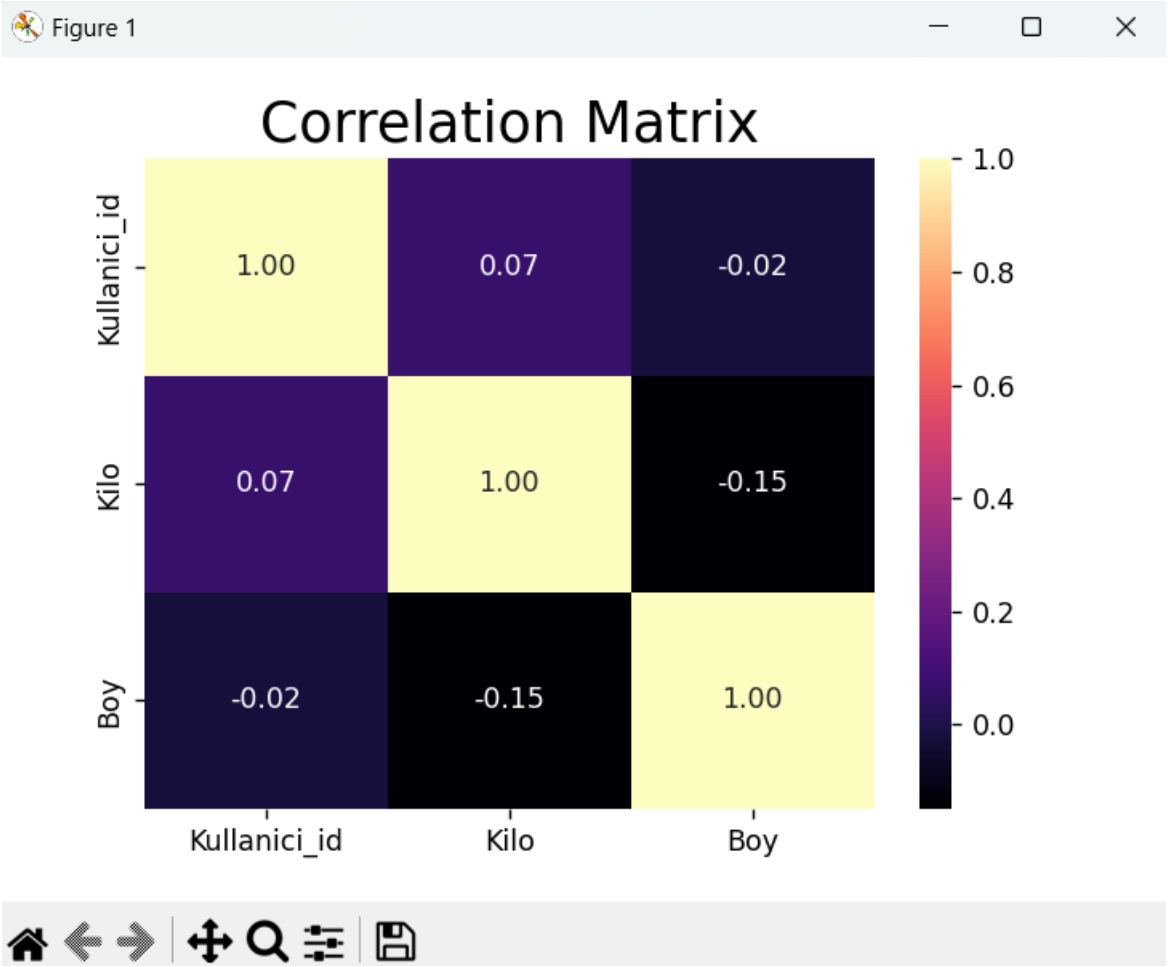
Korelasyon Matrisi

Sayısal değişkenler arasındaki ilişkileri görselleştirmek için korelasyon matrisi kullanılmıştır.

```
# Korelasyon matrisi
numeric_cols = df.select_dtypes(include=["float64", "int64"]).columns
plt.figure(figsize=[18, 13])
sns.heatmap(df[numeric_cols].corr(), annot=True, fmt=".2f", cmap="magma")
plt.title("Correlation Matrix", fontsize=20)
plt.show()
```

Açıklamalar:

- Korelasyon matrisi, sayısal değişkenler arasındaki ilişkileri gösterir. Değişkenler arasındaki korelasyon katsayıları, +1 (pozitif ilişki) ve -1 (negatif ilişki) arasında değişir.



Feature Engineering

Eksik Değer Analizi

Veri setindeki eksik değerlerin analizini yapmak için aşağıdaki fonksiyon kullanılmıştır.

```
def missing_values_table(dataframe):
    na_columns = [col for col in dataframe.columns if dataframe[col].isnull().sum() > 0]
    n_miss = dataframe[na_columns].isnull().sum().sort_values(ascending=False)
    ratio = (dataframe[na_columns].isnull().sum() / dataframe.shape[0] * 100).sort_values(ascending=False)
    missing_df = pd.concat([n_miss, np.round(ratio, 2)], axis=1, keys=['n_miss', 'ratio'])
    print("Eksik değer tablosu:\n", missing_df)

missing_values_table(df)
```

Açıklamalar:

- `missing_values_table`: Veri çerçevesindeki eksik değerleri kontrol eder ve eksik değerlerin sayısını ile yüzdesini gösteren bir tablo oluşturur. Bu, hangi değişkenlerde eksik verinin olduğunu anlamak için önemlidir.

```
Eksik değer tablosu:
                                n_miss  ratio
Cinsiyet                           778 33.010
Alerjilerim                        484 20.530
Kronik Hastaliklarim               392 16.630
Kan Grubu                          347 14.720
Kilo                               293 12.430
Il                                 227  9.630
Anne Kronik Hastaliklari           217  9.210
Baba Kronik Hastaliklari           156  6.620
Erkek Kardes Kronik Hastaliklari   121  5.130
Boy                                114  4.840
Kiz Kardes Kronik Hastaliklari      97  4.120
```

Eksik Değerlerin Doldurulması

Eksik değerlerin doldurulması için kullanılan fonksiyon:

```python
def fill_missing_values(df):
    numerical_cols = df.select_dtypes(include=[np.number]).columns
    imputer_num = SimpleImputer(strategy='median')
    df[numerical_cols] = imputer_num.fit_transform(df[numerical_cols])

    categorical_cols = df.select_dtypes(include=[object]).columns
    imputer_cat = SimpleImputer(strategy='most_frequent')
    df[categorical_cols] = imputer_cat.fit_transform(df[categorical_cols])

    return df

df = fill_missing_values(df)
```

Açıklamalar:

- `fill_missing_values`: Sayısal değişkenlerdeki eksik değerleri medyan ile, kategorik değişkenlerdeki eksik değerleri ise en sık görülen değer ile doldurur. Bu, veri setinin daha tutarlı ve eksiksiz hale gelmesini sağlar.

Aykırı Değer Analizi

Aykırı Değer Tespiti

Veri setindeki aykırı değerleri tespit etmek ve yönetmek için aşağıdaki fonksiyonlar kullanılmıştır.

```python
def outlier_thresholds(dataframe, col_name, q1=0.05, q3=0.95):
    if dataframe[col_name].dtype in ['int64', 'float64']:
        quartile1 = dataframe[col_name].quantile(q1)
        quartile3 = dataframe[col_name].quantile(q3)
        interquantile_range = quartile3 - quartile1
        up_limit = quartile3 + 1.5 * interquantile_range
        low_limit = quartile1 - 1.5 * interquantile_range
        return low_limit, up_limit
    return None, None

def replace_with_thresholds(dataframe, variable):
    low_limit, up_limit = outlier_thresholds(dataframe, variable)
    if low_limit is not None and up_limit is not None:
        dataframe.loc[dataframe[variable] < low_limit, variable] = low_limit
        dataframe.loc[dataframe[variable] > up_limit, variable] = up_limit

for col in num_cols:
    replace_with_thresholds(df, col)
```

Açıklamalar:

- `outlier_thresholds`: Belirli bir değişken için alt ve üst aykırı değer sınırlarını hesaplar. Bu sınırlar, 1.5 katı interquartil aralığı kullanılarak belirlenir.
- `replace_with_thresholds`: Aykırı değerleri tespit ettikten sonra, bu değerleri belirlenen sınırlar ile değiştirir. Böylece veri setinin daha temiz ve analiz edilebilir hale gelmesini sağlar.

Özellik Çıkarımı

Bu bölümde, veri setine yeni değişkenler eklenmiştir. Aşağıda gerçekleştirilen işlemler özetlenmiştir:

Yaş Değişkeninin Oluşturulması: YAS: Doğum tarihinden güncel tarihe kadar geçen gün sayısı hesaplanarak yaş bilgisi güncellenmiştir.

```
df['YAS'] = (pd.to_datetime('today') - df['Dogum_Tarihi']).dt.days // 365
```

Yaş Kategorileri:

- NEW_AGE_CAT: Yaş gruplarına göre (20-49 yaş arası "mature", 50 yaş ve üzeri "senior") kategoriler oluşturulmuştur.

```
df.loc[(df["YAS"] >= 21) & (df["YAS"] < 50), "NEW_AGE_CAT"] = "mature"
df.loc[(df["YAS"] >= 50), "NEW_AGE_CAT"] = "senior"
```

BMI Hesaplama:

- BMI: Vücut kitle indeksi, kilo ve boy bilgileri kullanılarak hesaplanmıştır.

```
df['BMI'] = df['Kilo'] / (df['Boy'] / 100) ** 2
```

BMI Kategorileri:

- NEW_BMI: BMI değerlerine göre (Aşırı Zayıf, Sağlıklı, Aşırı Kilolu, Obez) kategoriler oluşturulmuştur.

```
df['NEW_BMI'] = pd.cut(x=df['BMI'], bins=[0, 18.5, 24.9, 29.9, 100], labels=["Underweight", "Healthy", "Overweight", "Obese"])
```

Yaş ve BMI Kombinasyonu:

- NEW_AGE_BMI_NOM: Yaş ve BMI değerlerine göre detaylı kategoriler oluşturulmuştur (örn. "underweightmature").

```
df.loc[(df["BMI"] < 18.5) & ((df["YAS"] >= 21) & (df["YAS"] < 50)), "NEW_AGE_BMI_NOM"] = "underweightmature"
df.loc[(df["BMI"] < 18.5) & (df["YAS"] >= 50), "NEW_AGE_BMI_NOM"] = "underweightsenior"
df.loc[((df["BMI"] >= 18.5) & (df["BMI"] < 25)) & ((df["YAS"] >= 21) & (df["YAS"] < 50)), "NEW_AGE_BMI_NOM"] = "healthymature"
df.loc[((df["BMI"] >= 18.5) & (df["BMI"] < 25)) & (df["YAS"] >= 50), "NEW_AGE_BMI_NOM"] = "healthysenior"
df.loc[((df["BMI"] >= 25) & (df["BMI"] < 30)) & ((df["YAS"] >= 21) & (df["YAS"] < 50)), "NEW_AGE_BMI_NOM"] = "overweightmature"
df.loc[((df["BMI"] >= 25) & (df["BMI"] < 30)) & (df["YAS"] >= 50), "NEW_AGE_BMI_NOM"] = "overweightsenior"
df.loc[(df["BMI"] >= 30) & ((df["YAS"] >= 21) & (df["YAS"] < 50)), "NEW_AGE_BMI_NOM"] = "obesemature"
df.loc[(df["BMI"] >= 30) & (df["YAS"] >= 50), "NEW_AGE_BMI_NOM"] = "obesesenior"
```

Yan Etki Durumu:

- YENI_YAN_ETKI: Yan etki bildirim tarihinin mevcut olup olmadığına göre 0 veya 1 değeri atanmıştır.

```
df['YENI_YAN_ETKI'] = df['Yan_Etki_Bildirim_Tarihi'].notnull().astype(int)
```

Kilo ve Boy Kombinasyonu ile BMI Kategorileri:

- NEW_AGE_BMI_CAT: Yaş ve BMI değerlerine göre yeni kategoriler oluşturulmuştur (örn. "normalmature").

```
df.loc[(df["YAS"] >= 21) & (df["YAS"] < 50) & (df["BMI"] < 18.5), "NEW_AGE_BMI_CAT"] = "lowmature"
df.loc[(df["YAS"] >= 50) & (df["BMI"] < 18.5), "NEW_AGE_BMI_CAT"] = "lowsenior"
df.loc[(df["YAS"] >= 21) & (df["YAS"] < 50) & (df["BMI"] >= 18.5) & (df["BMI"] < 25), "NEW_AGE_BMI_CAT"] = "normalmature"
df.loc[(df["YAS"] >= 50) & (df["BMI"] >= 18.5) & (df["BMI"] < 25), "NEW_AGE_BMI_CAT"] = "normalsenior"
```

Label Encoding ve One-Hot Encoding

Bu bölümde, kategorik değişkenlerin kodlanması işlemleri yapılmıştır.

1. **Label Encoding:**
   - İkili (binary) kategorik değişkenler için LabelEncoder kullanıldı.
   - ```
     def label_encoder(dataframe, binary_col):
         labelencoder = LabelEncoder()
         dataframe[binary_col] = labelencoder.fit_transform(dataframe[binary_col])
         return dataframe

     binary_cols = [col for col in df.columns if df[col].dtypes == "O" and df[col].nunique() == 2]
     print("Binary Columns:", binary_cols)

     for col in binary_cols:
         df = label_encoder(df, col)
     ```
   - ```
     Binary Columns: ['Cinsiyet']
     ```

One-Hot Encoding:

- İkili değişkenler dışındaki kategorik değişkenler için `pd.get_dummies` kullanıldı.

```
cat_cols = [col for col in cat_cols if col not in binary_cols and col not in ["Yan_Etki"]]
print("Updated Categorical Columns:", cat_cols)

# Kategorik değişkenleri ayarlama
for col in cat_cols:
    df[col] = df[col].astype('category')

def one_hot_encoder(dataframe, categorical_cols, drop_first=False):
    dataframe = pd.get_dummies(dataframe, columns=categorical_cols, drop_first=drop_first,dtype=int)
    return dataframe
df = one_hot_encoder(df, cat_cols, drop_first=True)

# One-hot encoding sonrası veri türlerini kontrol et
print("One-Hot Encoding Sonrası Veri Türleri:")
print(df.dtypes)

# Değişiklikleri kontrol et
print("Güncellenmiş Veri Türleri:")
print(df.dtypes)

# Sonuçları görüntüle
print("İlk 5 Satır:")
print(df.head(5))
print("Veri Setinin Boyutu:", df.shape)
```

Veri Türlerinin Kontrolü:

- One-hot encoding sonrası veri türleri kontrol edildi.

Standartlaştırma

Bu bölümde, sayısal değişkenlerin standartlaştırılması işlemi gerçekleştirilmiştir.

2. **Sayısal Sütunların Belirlenmesi:**
    - Sadece sayısal veri türüne sahip sütunlar filtrelenmiştir.

```
num_cols = [col for col in num_cols if df[col].dtype in ['int64', 'float64']]
print("Sayısal Sütunlar:", num_cols)
```

```
Sayısal Sütunlar: ['Kullanici_id', 'Kilo', 'Boy']
```

Standartlaştırma İşlemi:

- `StandardScaler` kullanılarak sayısal sütunlar standartlaştırılmıştır.

```
scaler = StandardScaler()
df[num_cols] = scaler.fit_transform(df[num_cols])
```

Sümeyyye Bakırdal
sumeyyebakirdal@gmail.com