Drug Side Effects Analysis Project

1. Introduction

This project aims to analyze the side effects of drugs and prepare the dataset for predictive modeling. The process includes performing Exploratory Data Analysis (EDA) and data preprocessing to clean and handle missing data using appropriate methods.

2. Exploratory Data Analysis (EDA)

In this phase, the dataset was explored to understand its structure, identify missing values, and detect any anomalies. Visualizations were created to uncover patterns and relationships within the dataset.

2.1 Dataset Overview

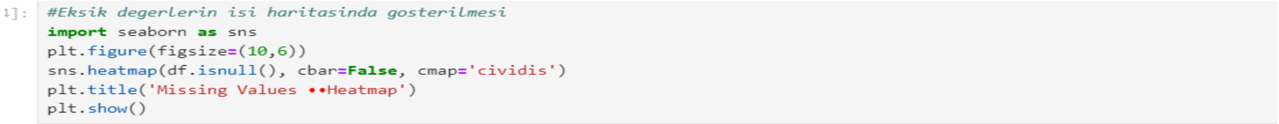
The general structure of the dataset was examined:

- **The data types of each column and the presence of missing values were analyzed.**
- **Commands used: df.info(), df.describe(), df.isnull().sum()**

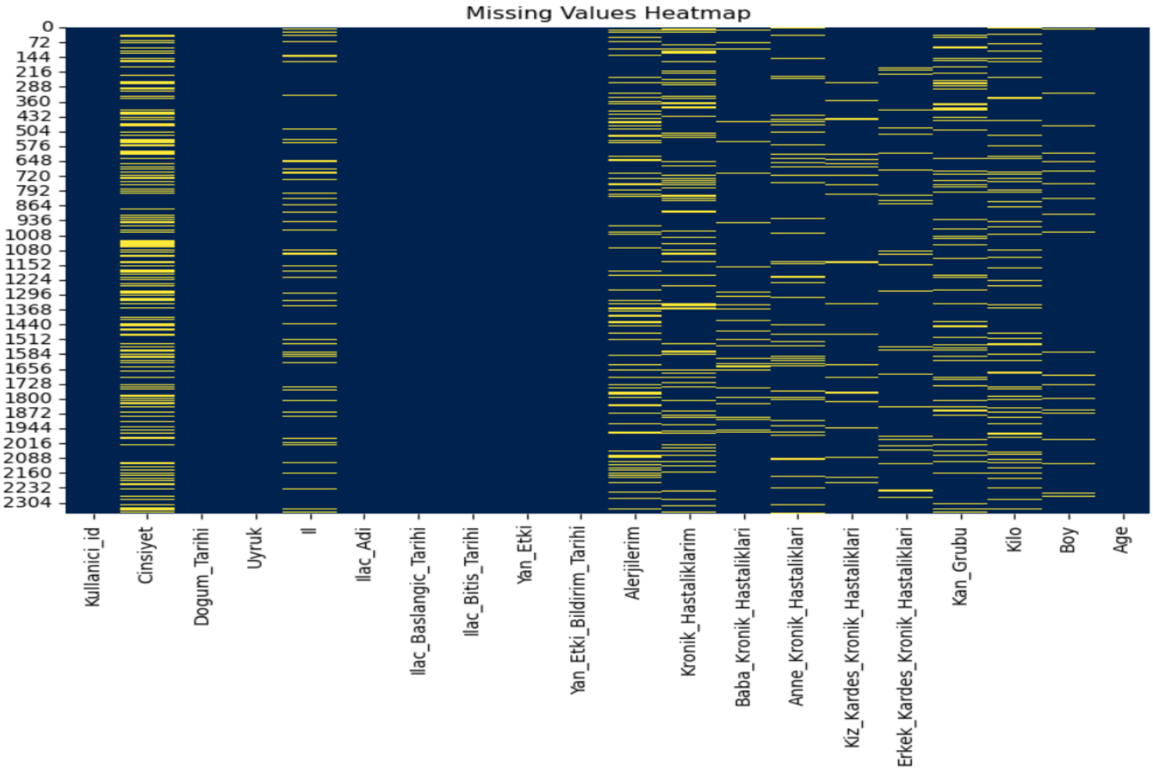
2.2 Missing Data Analysis

Missing values were identified and visualized using a heatmap.

Code:



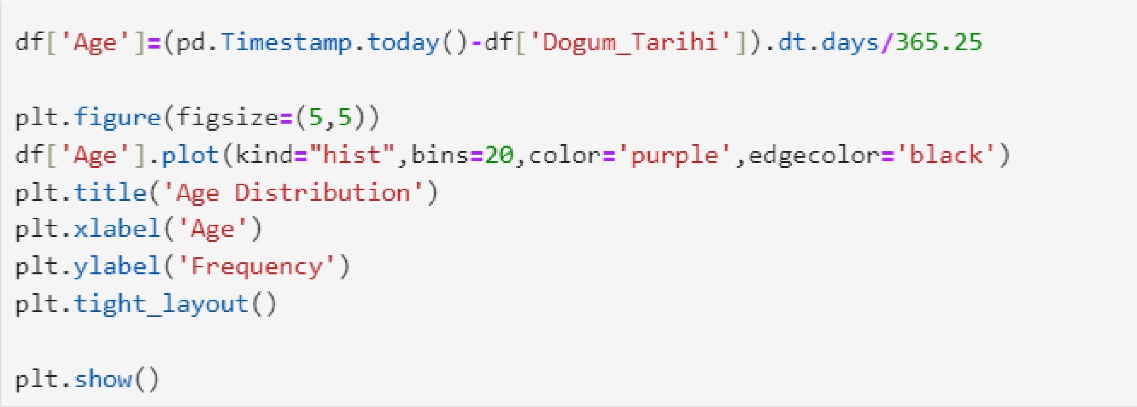
```
#Eksik degerlerin isi haritasinda gosterilmesi
import seaborn as sns
plt.figure(figsize=(10,6))
sns.heatmap(df.isnull(), cbar=False, cmap='cividis')
plt.title('Missing Values ••Heatmap')
plt.show()
```



2.3 Data Distribution

The distribution of numerical and categorical variables was analyzed using histograms and box plots.

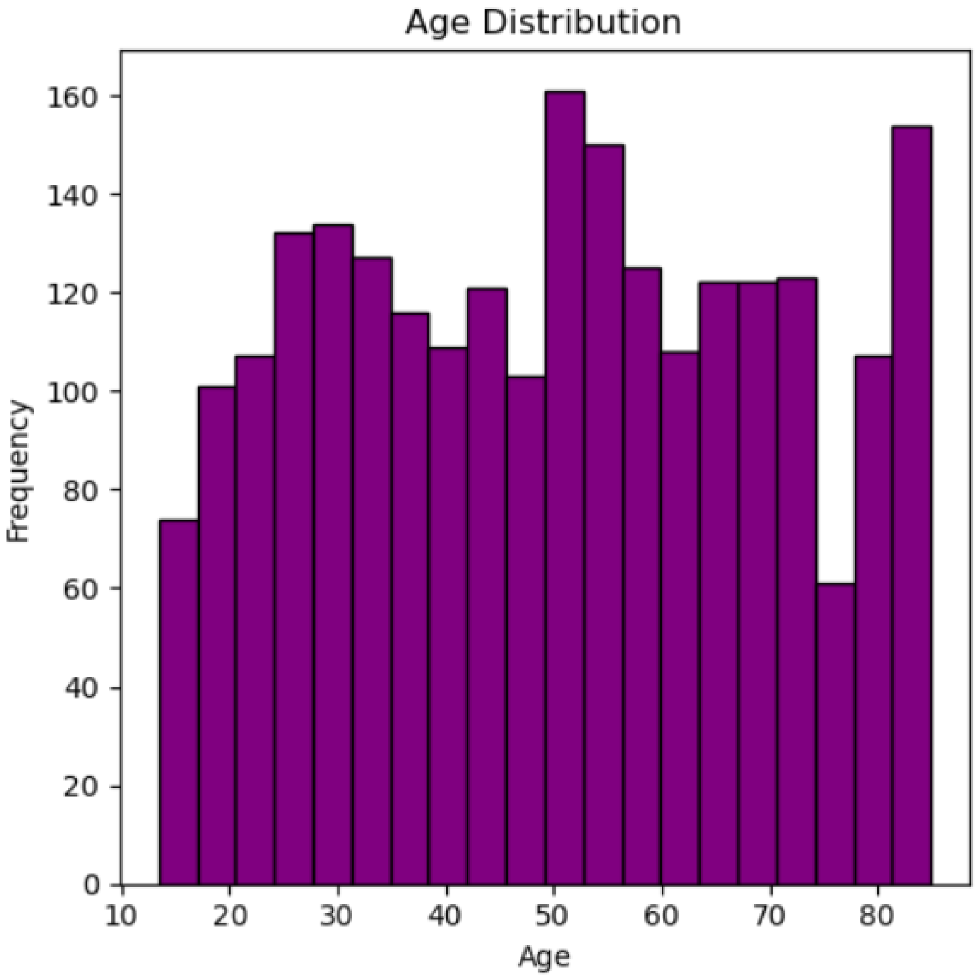
Example Histogram:



```
df['Age']=(pd.Timestamp.today()-df['Dogum_Tarihi']).dt.days/365.25

plt.figure(figsize=(5,5))
df['Age'].plot(kind="hist",bins=20,color='purple',edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.tight_layout()

plt.show()
```



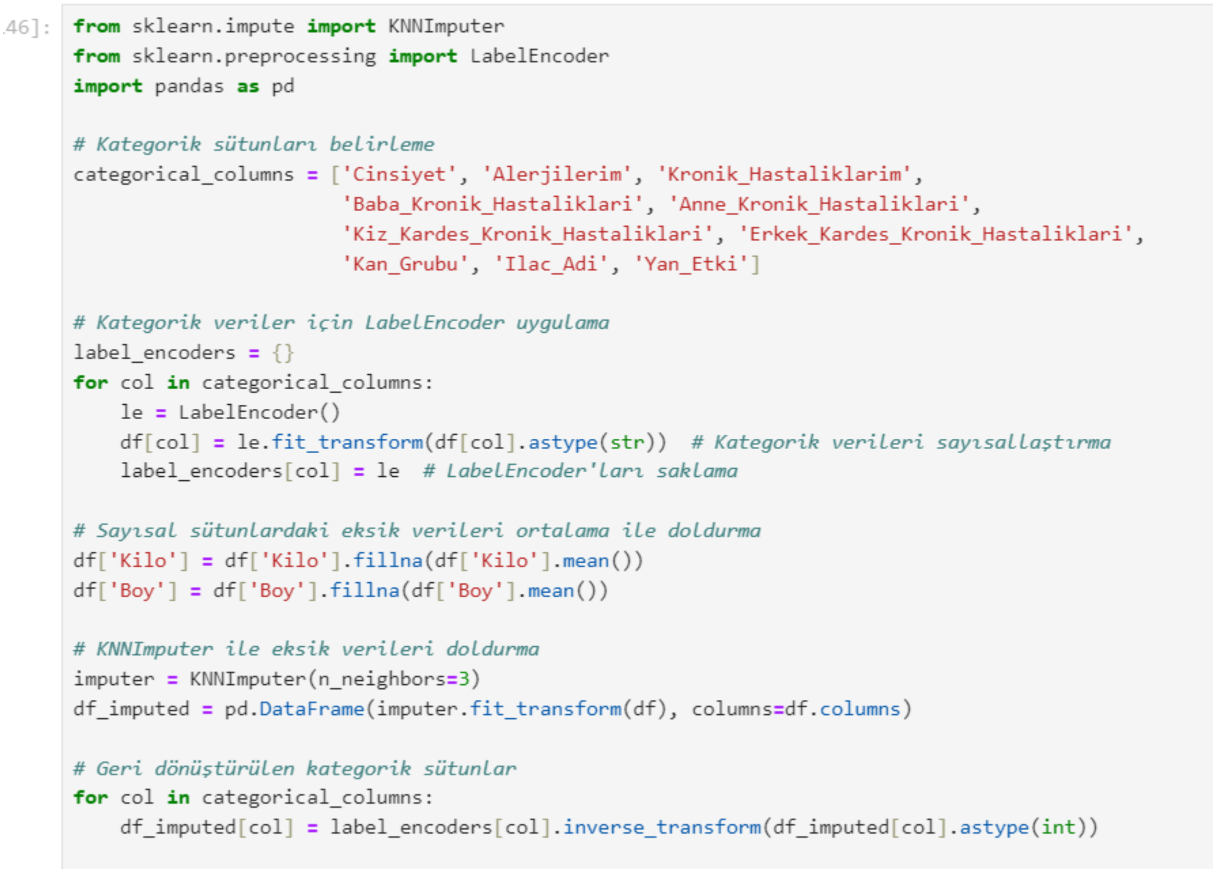
3. Data Preprocessing

Based on the findings from EDA, data preprocessing steps were carried out to handle missing data and prepare the dataset for further analysis. KNNImputer was used to fill in missing values for both categorical and numerical features.

3.1 Handling Missing Values with KNNImputer

Missing values were handled using KNNImputer. Before applying KNN, categorical features were encoded using LabelEncoder, and after imputation, they were transformed back to their original form.

Code:



```python
from sklearn.impute import KNNImputer
from sklearn.preprocessing import LabelEncoder
import pandas as pd

# Kategorik sütunları belirleme
categorical_columns = ['Cinsiyet', 'Alerjilerim', 'Kronik_Hastaliklarim',
                       'Baba_Kronik_Hastaliklari', 'Anne_Kronik_Hastaliklari',
                       'Kiz_Kardes_Kronik_Hastaliklari', 'Erkek_Kardes_Kronik_Hastaliklari',
                       'Kan_Grubu', 'Ilac_Adi', 'Yan_Etki']

# Kategorik veriler için LabelEncoder uygulama
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))  # Kategorik verileri sayısallaştırma
    label_encoders[col] = le  # LabelEncoder'ları saklama

# Sayısal sütunlardaki eksik verileri ortalama ile doldurma
df['Kilo'] = df['Kilo'].fillna(df['Kilo'].mean())
df['Boy'] = df['Boy'].fillna(df['Boy'].mean())

# KNNImputer ile eksik verileri doldurma
imputer = KNNImputer(n_neighbors=3)
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)

# Geri dönüştürülen kategorik sütunlar
for col in categorical_columns:
    df_imputed[col] = label_encoders[col].inverse_transform(df_imputed[col].astype(int))
```

4. Results

By following the EDA and preprocessing steps, the dataset was cleaned and prepared for further analysis and modeling. Missing values were successfully filled, and categorical features were encoded and restored to their original form after imputation.

5. Contact Information

- **Name: Mehmet Emin Güvercin**
- **Email: m.guvercin34@gmail.com**