

**Data Science Intern Case Study:**

**Side Effect Data: Exploratory Data Analysis and Preprocessing**

**Deniz Ağar**

[deniz.agar20@gmail.com](mailto:deniz.agar20@gmail.com)

**Introduction**

This document presents an exploratory data analysis (EDA) and data preprocessing steps for the provided **Side Effect Data**. The dataset contains various attributes related to users, including demographic information, medication history, and reported side effects. The primary objective of this analysis is to gain an in-depth understanding of the dataset, uncover potential patterns, detect anomalies, and prepare the data for further modeling or analysis.

In the **Exploratory Data Analysis** (EDA) phase, key descriptive statistics are calculated, and visualizations such as histograms and scatter plots are used to explore the distributions and relationships between different variables. Additionally, missing data and categorical variables are examined, and potential correlations between variables are identified through correlation matrices and heatmaps.

In the **Data Preprocessing** phase, the dataset is cleaned to ensure that it is suitable for further analysis. This includes handling missing values, encoding categorical variables, and normalizing or standardizing numerical features. These steps help to improve the quality of the data and make it ready for any potential modeling or machine learning tasks.

The following sections will detail the findings from the EDA, as well as the preprocessing techniques applied to transform the raw data into a usable format.

**TASKS**

1. **Exploratory Data Analysis (EDA)**

**Step 1.1: Importing Libraries**

As a first step, I created a new Python file and imported the necessary libraries such as Pandas (for data manipulation), Matplotlib and Seaborn (for data visualization), and Missingno (for visualizing missing data). These libraries will support the analysis and presentation of the dataset.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Step 1.2: Loading Data**

I will load the dataset using the Pandas library, I use the pd.read\_excel() function to load the Excel file containing the dataset. The file, named 'side\_effect\_data 1.xlsx', is read into a DataFrame object (df), which allows for easy manipulation and analysis of the data in subsequent steps.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Step 1.3:** **Examining the General Structure of the Data Set**

In this step, I explore the overall structure of the dataset to understand its composition and characteristics. By examining the first few rows, inspecting the data types of each column, and checking for missing values, I can gain initial insights into the content of the dataset. This information is crucial for planning the data cleaning and preprocessing steps, as it reveals potential issues such as incomplete or incorrectly formatted data that need to be addressed.

*Displaying the First 5 Rows:*

print(df.head())

This command shows the first five rows of the dataset. It helps me quickly understand the structure and content of the data, including the types of variables and initial observations.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Displaying Data Types and Column Information:*

print(df.info())

The df.info() function provides a concise summary of the dataset, including the data types of each column and the number of non-null entries. This helps me identify missing values and understand the general composition of the data.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, menü, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

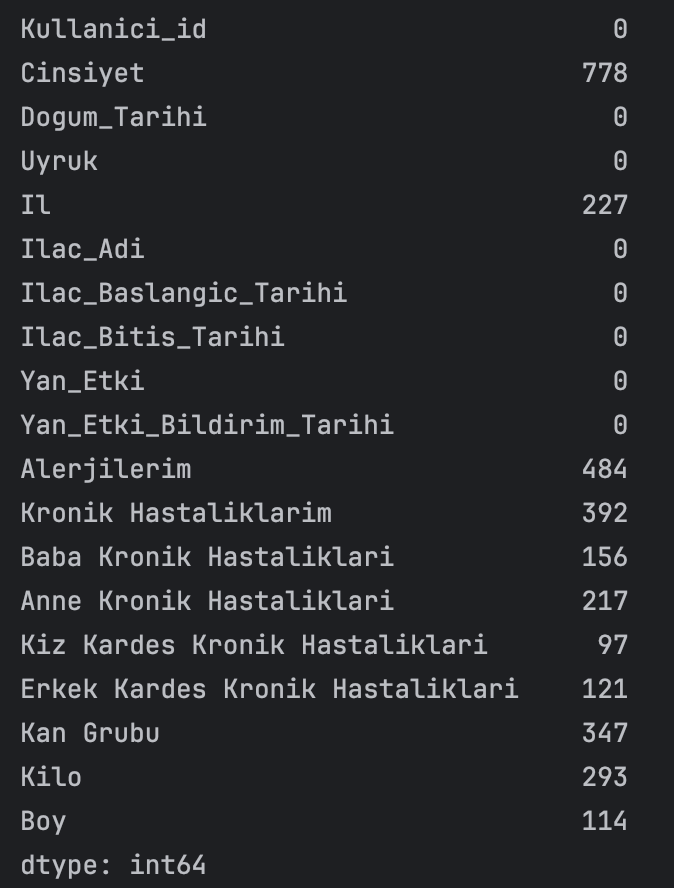
*Checking for Missing Values:*

print(df.isnull().sum())

This command calculates the number of missing values in each column. It gives me an overview of how much data is missing, which is essential for deciding how to handle incomplete data during preprocessing.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



Cinsiyet column has **778** missing values.

İl column has **227** missing values.

Alerjilerim column has **484** missing values.

Kilo column has **293** missing values.

Boy column has **114** missing values.

Columns without missing values are represented with a **0** (e.g., Dogum\_Tarihi, Uyruk, İlac\_Adi).

**Step 1.4: Visualization Steps & Outputs**

1. **Visualizing The Missing Data Matrix**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

This code uses the Missingno library to visualize missing data in the dataset. In the first step, a graphic figure is created with matplotlib and its dimensions are set. Then the msno.matrix() function visualizes the missing data in matrix format. Missing data is shown with white areas, and filled data with black areas. The column names on the x-axis are tilted by 45 degrees to make it more readable and the font sizes are reduced. The left y-axis is hidden because it is not meaningful for missing data analysis. The margins are adjusted with plt.subplots\_adjust() so that the graph fits properly. Finally, the window title is changed to be compatible since I am using MacOS and the graph is displayed. (If you are using Windows or Linux, this code may not work properly.)

**ekran görüntüsü, metin, çizgi, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu**

This visualization allows for quick analysis of missing data in the dataset. We create a missing data matrix using the Missingno library. Missing cells are shown in white, and filled cells are shown in black. This makes it easier to visualize which columns in the dataset have missing data and identify missing data strategies.

1. **Visualizing The Percentage Of Missing Data**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

This code creates a bar chart to visualize the percentage of missing data in the dataset. The percentage of missing data in each column is calculated and visualized. The column names are shown on the X-axis, and the percentage of missing data is shown on the Y-axis. The chart dimensions are adjusted, labels are skewed, and the font size is reduced to increase readability. Finally, the window title is determined and the chart is displayed.

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

Açıklama otomatik olarak oluşturuldu

This chart shows the percentage of missing data in the dataset by column. The X-axis shows the columns in the dataset, while the Y-axis shows the percentage of missing data in each column. As can be seen in the chart, columns such as “Cinsiyet” and “Alerjilerim” have a significant amount of missing data, while some other columns have a much lower rate of missing data. This analysis helps us understand which columns are missing data and make decisions about how to handle missing data.

1. **Visualizing The Age Distribution**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This code is used to calculate the age of individuals based on the birthdate information in the dataset and visualize the age distribution. First, the data in the birthdate column is converted to datetime format and the age of each individual is calculated by subtracting the birthyear from the current year. Then, the calculated age data is visualized in a histogram. This graph shows the age range in which individuals in the dataset are concentrated. Examining the age distribution is an important step to analyze the impact of side effects on specific age groups.

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

Açıklama otomatik olarak oluşturuldu

This graph shows the age distribution as a histogram. The age column is calculated from the date of birth information. The histogram shows the number of people in different age ranges as a frequency. For example, we can see that the most dense age ranges in the dataset are approximately 48-52 and 70-75. This type of visualization is important to examine potential side effects or demographic differences for certain age groups in the dataset to be analyzed.

1. **Visualizing The Gender Distribution**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This code visualizes the gender distribution in the dataset with a bar chart. First, a new figure is created and its size is adjusted, and then the number of data in the `Gender` column is calculated and displayed on the chart. The X-axis shows gender (female and male), and the Y-axis shows frequency values. The title of the chart is set to "Gender Distribution", and finally, the `tight\_layout()` function is used to prevent labels from overlapping.

ekran görüntüsü, ekran, görüntüleme, dikdörtgen, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

This graph visualizes the gender distribution in the dataset. The frequencies of users in the 'Female' and 'Male' categories are shown as bars. As can be seen in the graph, the number of female users is higher than the number of male users. This type of analysis is useful to understand the ratio of female and male users in the dataset and is important to examine if there are different side effects by gender in later analyses.

1. **Visualizing The Correlation Matrix**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

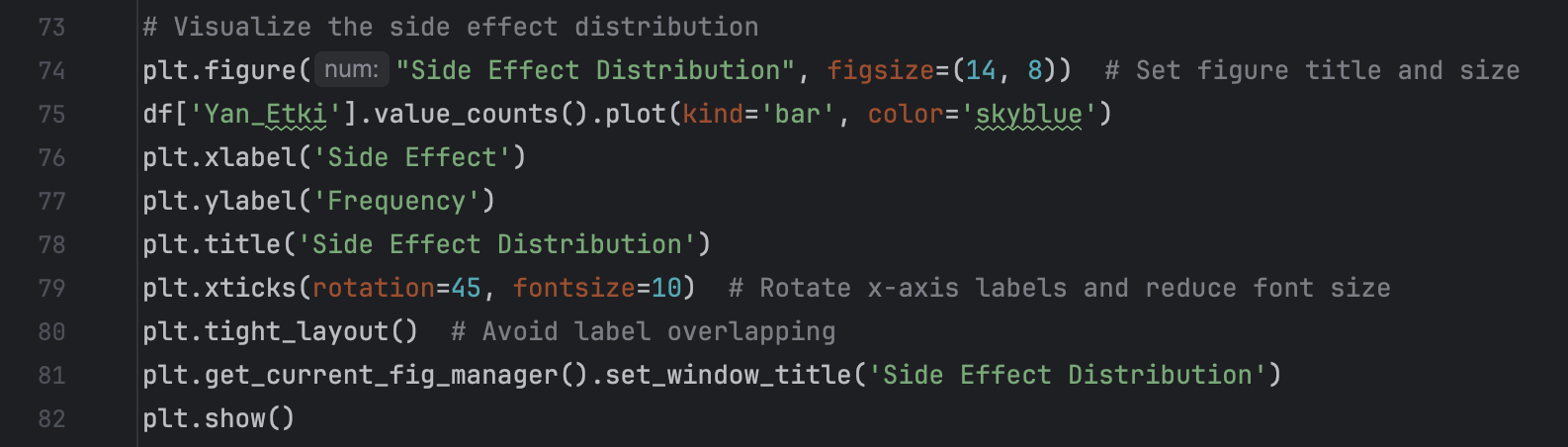
This code creates a correlation matrix (heatmap) to visualize the correlation between numeric columns in the dataset. First, columns containing only numeric data (`float64`, `int64`, `datetime64`) are selected and the correlations of these columns with each other are calculated. These correlations are visualized as a heatmap using the `sns.heatmap` function. While the correlation values ​​are expressed with colors in the heatmap, the numeric correlation value is also shown in each cell. The title of the graph is set to "Correlation Matrix", `tight\_layout()` is used to prevent labels from overlapping, and the window title is set for MacOS.

ekran görüntüsü, kare, dikdörtgen, renklilik içeren bir resim

Açıklama otomatik olarak oluşturuldu

This heatmap allows us to analyze the correlations between the numerical columns in our dataset. The correlation coefficient represents the strength of the linear relationship between two variables, ranging between -1 and 1. A value of 1 indicates a perfect positive correlation (i.e., both variables increase together), while -1 indicates a perfect negative correlation (i.e., as one variable increases, the other decreases). A value of 0 suggests no correlation between the two variables.

1. **Visualizing The Side Effect Distribution**



This block of code is used to visualize the distribution of side effects reported in the dataset. It creates a bar chart to display how frequently each side effect appears in the data. The `plt.figure()` function sets up the chart, specifying its size and title. The `df['Yan\_Etki'].value\_counts().plot(kind='bar')` line creates a bar plot based on the frequency of different side effects in the 'Yan\_Etki' column. The x-axis is labeled 'Side Effect,' and the y-axis shows the frequency, with the title 'Side Effect Distribution.' The x-axis labels are rotated 45 degrees for better readability, and the layout is adjusted to prevent overlapping of elements. Finally, the `plt.show()` function displays the chart. The window title is also customized using `plt.get\_current\_fig\_manager().set\_window\_title()` to ensure consistency across operating systems.

ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, dikdörtgen, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

This bar chart visualizes the distribution of side effects reported in the dataset. The x-axis represents various side effects, and the y-axis shows the frequency with which each side effect was reported. The chart highlights that the most frequently reported side effects are "Ağızda farklı bir tat’ and "Tansiyon Yükselme," both with over 200 occurrences, followed by "Yorgunluk" and "Görmede Bulanıklık." Other side effects are reported less frequently, with decreasing frequency as we move from left to right. The distribution shows that a few side effects are highly prevalent in the dataset, while others are much less common.

1. **Visualizing The Drug Distribution**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

I visualized the distribution of the shortened drug names using a bar plot. The x-axis represents the shortened drug names, while the y-axis displays their frequency. The chart uses a green color palette for easy differentiation. Additionally, the x-axis labels are rotated to avoid overlapping and improve readability. This modification ensures a cleaner and more interpretable visualization of the drug distribution.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

This bar chart visualizes the frequency distribution of drug names, where the names have been shortened to display only the first two words. The x-axis lists the drugs, and the y-axis shows the frequency with which each drug appears in the dataset. By shortening the names, I reduced overcrowding, making the graph easier to read and interpret.

In this visualization, drugs like "fluphenazine hcl" and "divaiproex sodium" appear most frequently, indicating they are more commonly used or reported in the dataset. The frequencies then gradually taper off for less common drugs. The green bars ensure clear distinction between the values while making the overall chart visually appealing.

1. **Visualizing The Relationship Between Gender and Age**

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

This code generates a boxplot to visualize the relationship between gender and age in the dataset. It uses Seaborn’s `boxplot` function to compare the distribution of ages between the two gender categories. The x-axis represents the 'Cinsiyet' (Gender) column, and the y-axis represents the 'age' column. The plot helps identify any differences in the age range and distribution between male and female participants.

The boxplot shows the median, interquartile range (IQR), and potential outliers within the gender groups. It is particularly useful for understanding how age varies across genders and spotting any significant age differences or patterns. The chart is customized to avoid overlapping labels and ensure clear visualization by using `tight\_layout`.

dikdörtgen, ekran görüntüsü, kare, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

This boxplot compares the age distribution across male and female participants. On the y-axis, we have the age, and the x-axis represents gender ('Male' and 'Female'). The box shows the interquartile range (IQR) for both genders, with the horizontal line inside the box representing the median age.

- The median age for both genders appears to be around 50.

- The age range is wider for males, with the upper quartile extending higher than for females.

- There are no significant outliers in this boxplot, and the data shows that both male and female participants span a similar age range, although males tend to have a slightly broader age distribution.

**All Visualizations**

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

Açıklama otomatik olarak oluşturuldu

1. **Data Pre-Processing**

In Task 2, I focused on cleaning and preprocessing the data to prepare it for further modeling based on the findings from the exploratory data analysis (EDA). Specifically, I addressed missing data and transformed categorical variables into a numerical format suitable for machine learning models. To handle missing values, I used **SimpleImputer**—for numerical columns such as 'Kilo' and 'Boy,' the imputer applied strategies like 'mean' for 'Kilo' and 'median' for 'Boy' to fill in the gaps. For categorical columns such as 'Cinsiyet,' 'Il,' and 'Kan Grubu,' I used the **SimpleImputer** with the 'most frequent' strategy to fill missing values with the most commonly occurring category. After that, I applied **LabelEncoder** to convert these categorical columns into numerical labels, ensuring that the data was in a format suitable for analysis or machine learning tasks. This combination of imputing missing values and encoding categorical data ensured the dataset was more robust and ready for predictive modeling.

* 1. **Handling Missing Values with SimpleImputer for Numerical Columns**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

The purpose of this section is to handle missing numerical data by utilizing the SimpleImputer to fill in the gaps. For the Kilo column, missing values are filled with the mean of the existing values, ensuring that the average is maintained. In contrast, for the Boy column, missing values are filled using the median, which is less sensitive to outliers and provides a more robust estimate. This approach ensures that the dataset is complete, preventing any disruption during subsequent modeling or analysis due to missing numerical data.

* 1. **Handling Missing Values for Categorical Columns Using SimpleImputer**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

This section addresses missing values in categorical columns like 'Cinsiyet', 'Il', and 'Kan Grubu'. The SimpleImputer uses the most frequent (mode) strategy to fill in these missing values.

* 1. **Encoding Categorical Variables with LabelEncoder**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

This part of the code converts categorical variables into numerical representations using LabelEncoder. This is done for the 'Cinsiyet', 'Il', and 'Kan Grubu' columns. Many machine learning models cannot process categorical data directly. Converting them to numerical values ensures compatibility with these models. LabelEncoder is particularly useful for categorical data with a defined order or non-repeating categories.

*Outputs*

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

**metin, ekran görüntüsü, menü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Data Analysis and Insights**

*Missing Data Analysis*

Upon analyzing the missing data, it is evident that certain columns, particularly those related to personal health information, such as "Alerjilerim" (My Allergies), "Kronik Hastalıklarım" (My Chronic Diseases), and "Baba Kronik Hastalıkları" (Father's Chronic Diseases), have significant missing values. These columns contain sensitive personal information, and the missing data filling process was deliberately not applied. This decision was made to avoid introducing bias or inaccuracies into the dataset, as randomly filling such data could lead to erroneous conclusions.

ekran görüntüsü, dikdörtgen, çizgi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Age Distribution*

The age distribution shows that the majority of the users are around 50 years old, although the overall age range is relatively well-balanced. Age is a critical factor when analyzing drug side effects, as older individuals may experience different reactions compared to younger users.

Middle-aged and older individuals may be more susceptible to certain drug side effects. Further analysis could explore the correlation between age groups and the prevalence of specific side effects to provide deeper insights.

diyagram, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Gender Distribution*

The gender distribution data reveals that women are more represented in the dataset compared to men. This imbalance in gender representation should be considered when comparing side effects across genders.

The higher number of women in the dataset could lead to a perception that side effects are more common among women. However, this imbalance must be accounted for when analyzing gender differences in drug side effects. Further gender-specific analysis is recommended to avoid skewed conclusions.

ekran görüntüsü, ekran, görüntüleme, dikdörtgen, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Weight and Height Correlation*

The correlation matrix shows a weak negative correlation between weight and height, indicating that taller individuals may tend to weigh less, although the relationship is not very strong (-0.13). Additionally, the correlation between gender and both weight and height is also weak.

While weight and height are important physical characteristics for drug dosage adjustments, there is no strong correlation between these variables and the occurrence of side effects in the dataset. Further analysis could investigate the potential relationship between physical attributes and drug side effects in more detail.

ekran görüntüsü, kare, dikdörtgen, renklilik içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Side Effect Distribution*

The side effect distribution shows that the most frequently reported side effects are a different taste in the mouth, high blood pressure, fatigue, and nausea. These are the most commonly experienced side effects across the dataset.

These common side effects could be linked to specific drugs, age groups, or gender. For instance, high blood pressure could be more prevalent among older individuals. Further analysis can explore whether certain side effects are more common in specific demographic groups or associated with particular drugs.

ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, dikdörtgen, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Drug Distribution*

The drug distribution analysis indicates that certain drugs are more frequently used than others. This may result in more side effects being reported for these drugs, not necessarily because they cause more side effects, but due to their higher usage rate.

The higher frequency of side effects reported for certain drugs could be a reflection of their widespread use. This does not imply that these drugs are inherently more dangerous but rather that more data is available for them. Further analysis can explore the relationship between the frequency of drug use and the prevalence of side effects to draw more accurate conclusions.

ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**CONCLUSION**

The findings from this analysis highlight the importance of considering missing data, demographic distributions, and drug usage when analyzing drug side effects. The intentional exclusion of filling missing values in sensitive health-related columns ensures the accuracy of the analysis. Demographic factors such as gender and age play a significant role in the occurrence of side effects and should be carefully considered in further analyses. Overall, the dataset provides valuable insights into the relationship between physical attributes, drug usage, and the side effects experienced by different demographic groups, paving the way for more informed decisions regarding drug administration.