Eren Erdoğan – erenerdogan12323@gmail.com

PUSULA TALENT ACADEMY – DATA SCIENCE CASE STUDY

Overview: This document includes the strategies I followed, the data analysis and the decisions I made as a result of the analysis while completing the task given for Pusula Talent Academy.

I was expected to work on a physical medicine & rehabilitation dataset consisting of 2235 observations and 13 features. The goal is to conduct in-depth EDA and make the data ready for potential predictive modeling.

Strategy: At first, I form a plan to handle the task, the plan has two phases:

- **First Phase: EDA**
  - **Step 1.1: Setup and Initial Data Inspection**
    - Environment Setup
    - Loading Data
    - First Look
    - Check for Duplicates
  - **Step 1.2: Univeriate Analysis**
    - Numerical Features
    - Categorical Features
    - High-Cardinality (Complex) Features
  - **Step 1.3: Bivariate Analysis**
    - Target Value vs. Numerical Features
    - Target Value vs. Categorical Features
- **Second Phase: Data Preprocessing**
  - **Step 2.1: Data Cleaning**
    - Handle Missing Values
    - Address Outliers
    - Correct Data Types
  - **Step 2.2: Feature Engineering**
    - Parse Comma-Seperated Columns
    - Create Bins/Groups
  - **Step 2.3: Feature Transformation and Encoding**
    - Categorical Variable Encoding
    - Numerical Feature Scaling
- **Final Review**

First Phase: EDA

Understanding data characteristics, finding patterns, identifying anomalies.

Step 1.1: Setup and Initial Data Inspection

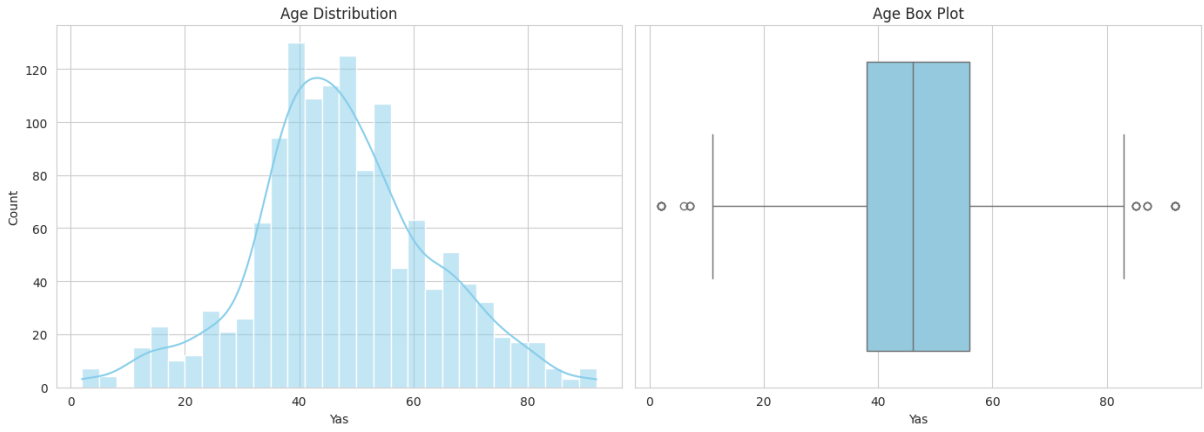
To control and manipulate the dataset, I decided to use “pandas” and “numpy” libraries. Also, for data visualization, i decided to use “matplotlib” and “seaborn”. Because they are some of the most widely used libraries. I also used Google Colab for it’s easy use and share the project easily.

After including libraries, i load the xlsx file and printed a sample of the data set, data frame info and shape of the data set for initial inspection. I checked the sample of the data set to see if it loaded correctly, data frame info to understand which column holds which type of data.

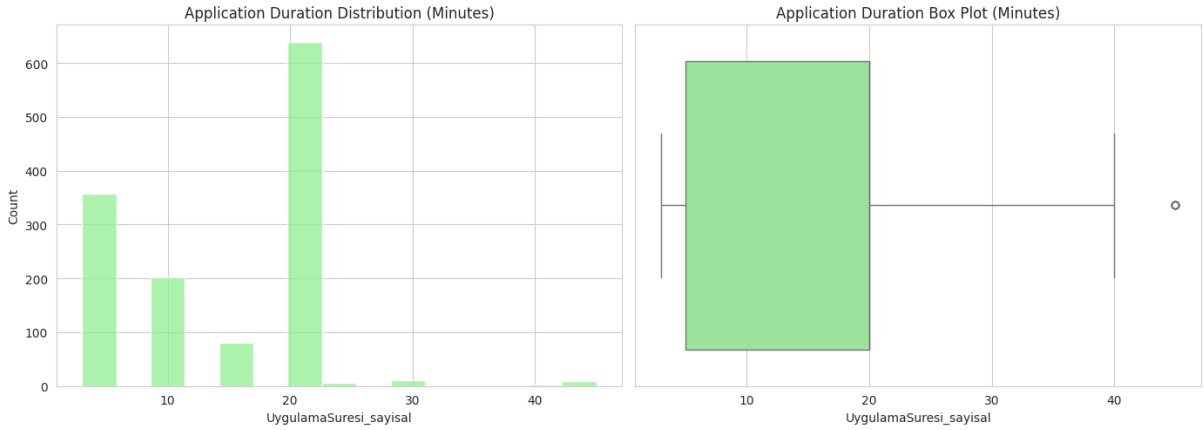
Then, I checked for the duplicated rows. I found there are 928 duplicated rows. Since there aren’t any columns that can distinguish the duplicates, like date or treatment number, I decided to remove the duplicated rows, of course holding one row for each duplication. Which reduced the dataset shape from (2235, 13) to (1307, 13). Duplication process are handled in a single cell of code, so it can be passed if needed.

Step 1.2: Univariate Analysis

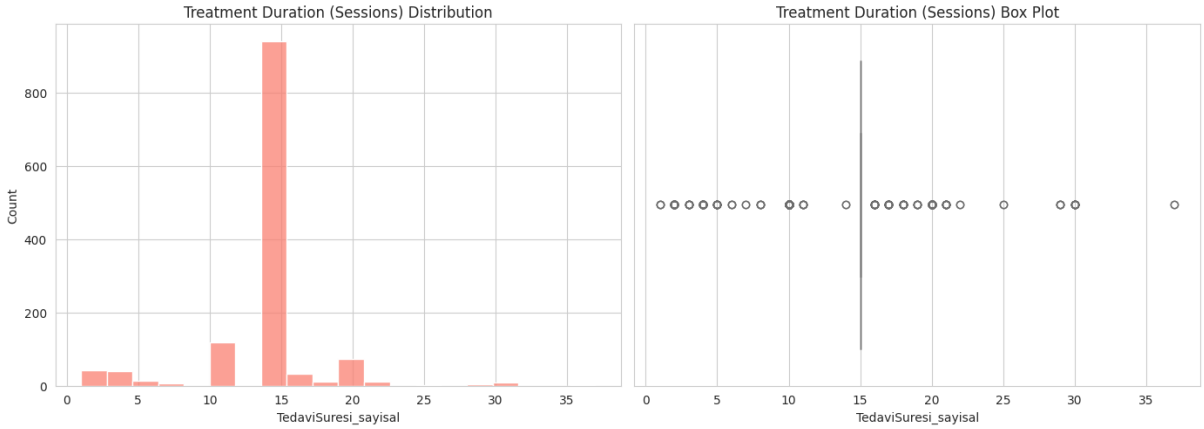
Numerical Features: Columns “Yas”, “UygulamaSuresi” and “TedaviSuresi” are numerical features. So I make the necessary arrangements to “UygulamaSuresi” and “TedaviSuresi” to change the data types to integers.



The distribution of “Yas” is close to normal with majority of patients between 40 and 60 years old.

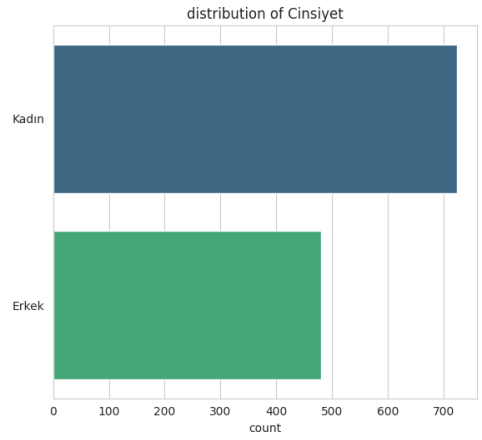


“UygulamaSuresi” is not continous. Major peak is at 20. This can show a standardized session timing.

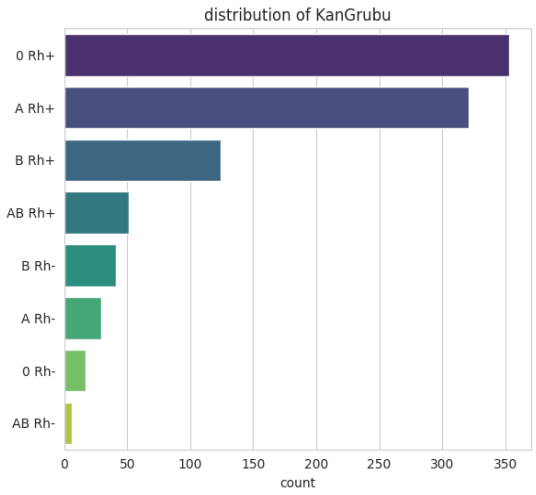


“TedaviSuresi”, which is the target variable, is not continous either. Even there are at least eighteen unique values, there is a major peak at 15. Which direct me to suggest applying classification rather than linear regression. Because when linear regression is applied, the predictions will mostly be around 15 (values like 14.8 or 15.3). So, to apply classification, i must create target classes.

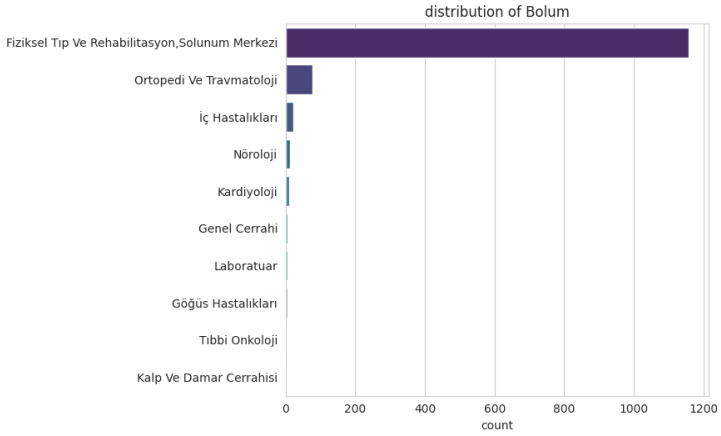
Categorical Features: Columns “Cinsiyet”, “KanGrubu”, “Uyruk” and “Bolum” are simple categorical features, which have no column-seperated information.



There are 723 female and 480 male patients. This gives total 1203 non-missing entries from 1307 rows, which means there are 104 rows missing gender information.



0 Rh+ and A Rh+ are the most prevalent blood types. This column also has a significant number of missing entries.



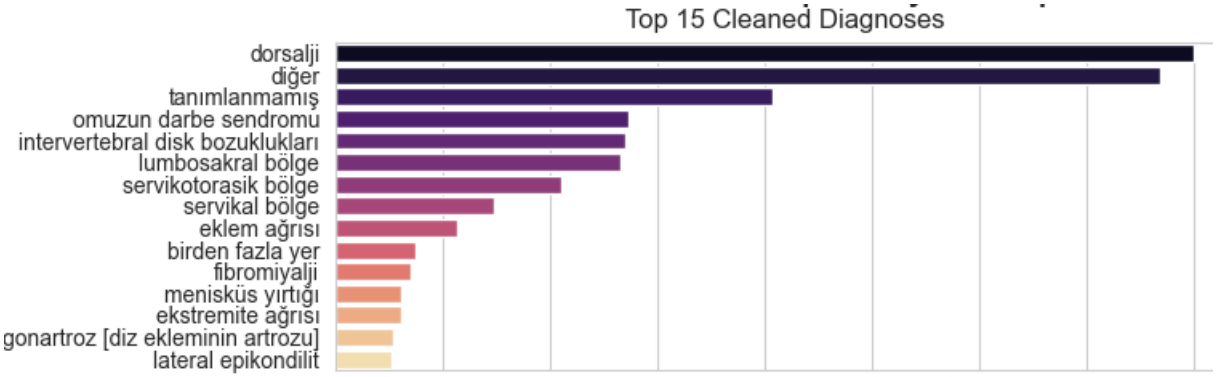
Majority of the patients are in “Fiziksel Tıp Ve Rehabilitasyon,Solunum Merkezi”. Variation is good enough to later exploration of if treatement durations differ across the departments.

High-Cardinality (Complex) Features: Columns “KronikHastaliklar”, “Alerji”, “Tanilar”, “TedaviAdi” and “UygulamaYerleri” are complex categorical features, which have column-seperated information.

When checked, it’s clear that there are some typos/synonyms in some of these features. So, i made some cleaning:

- For “KronikHastaliklar”: “hiportiroidizm” and “hipotirodizm” changed to “hipotiroidizm”.
- For “Alerji”: There are several upper/lower case differences, so all the entries changed to lower case to prevent the seperation of them (For example: “SUCUK” and “Sucuk”, “POLEN” and “Polen”, “TOZ” and “Toz”). Also, the typo mistake “Volteren” is changed to “voltaren”.
- For “Tanilar”: Again, all entries changed to lower case. Also there are two entries as a blank entry and “şimdiki”. These entries removed becuase they don’t identify any diagnoses, so they are unnecessary.

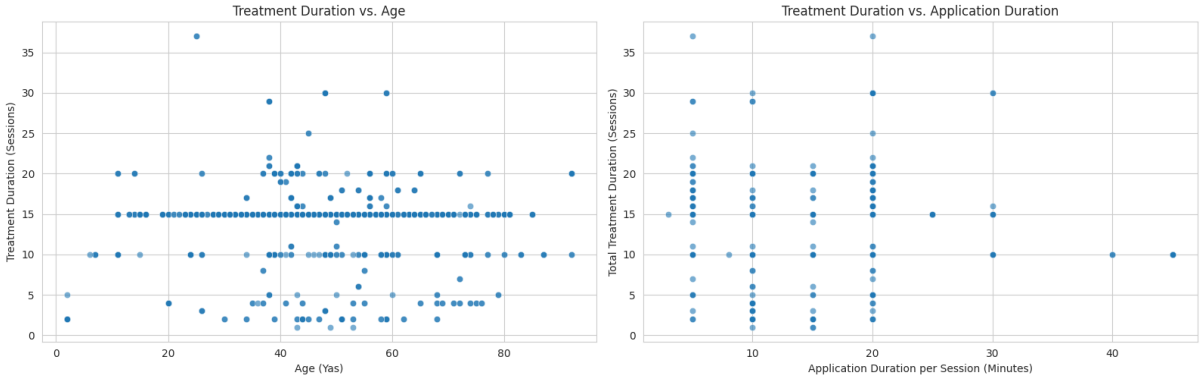
After this cleaning;



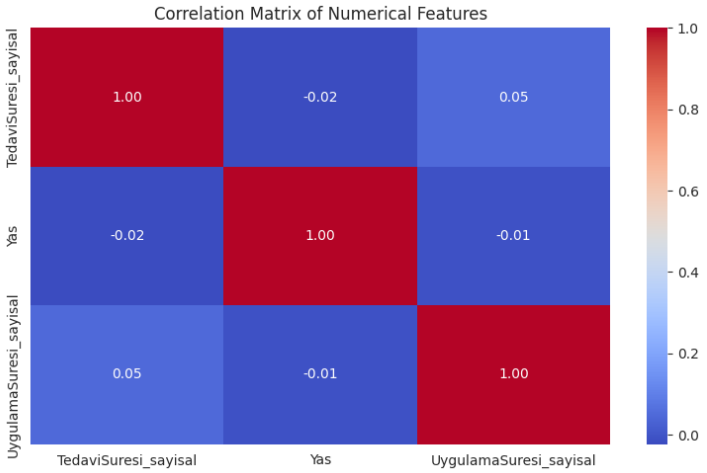
There were 285 unique diagnoses. So, I took the 15 most repeated diagnoses. The reason behind the taking most repeated ones is to reduce overwhelming complexity and find meaningful patterns. A model trained on so many features might learn the noise (random variations) from the rare diagnoses instead of the true underlying patterns, causing it to perform poorly on new data.

Step 1.3: Bivariate Analysis

Target Variable vs. Numerical Features:



The main takeaway is that there is no significant linear relationship between the total treatment duration and patient's age or the application duration.

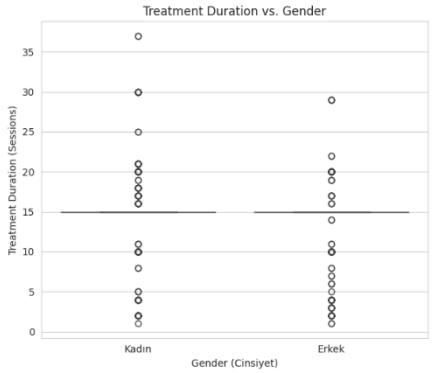


The correlation matrix confirms that the linear relationship between target value and numerical features is low (closer to 0).

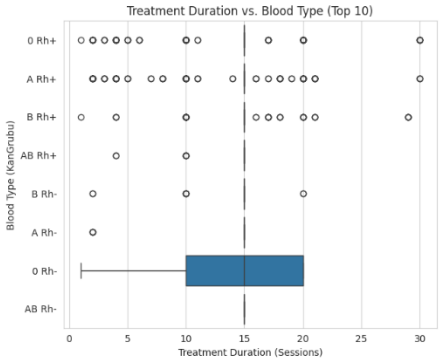
So, the analysis will need to rely more heavily on the categorical features.

Target Value vs. Categorical Features:

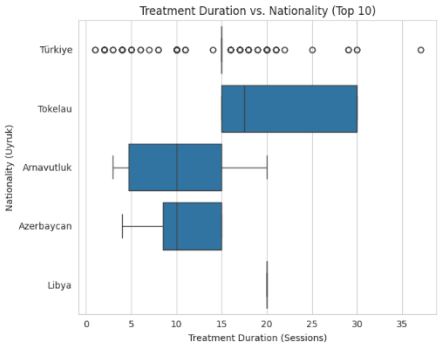
For simple categorical features,



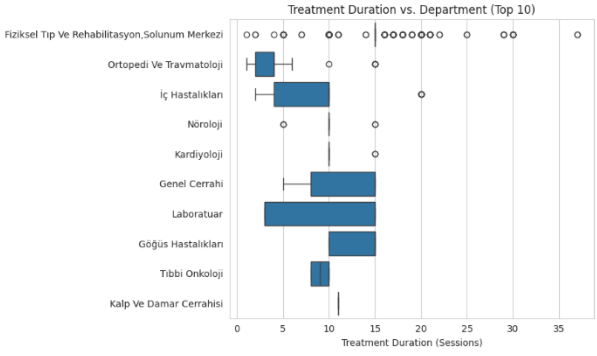
Plots seems quite similar for both genders. Medians are 15. Also, spread of the data is similar. Gender doesn't seem to bee factor in determining.



When comparing across the different blood types, all the box plots look very similar. The median for nearly every blood type is 15 sessions. There are no noticeable differences in the distributions.

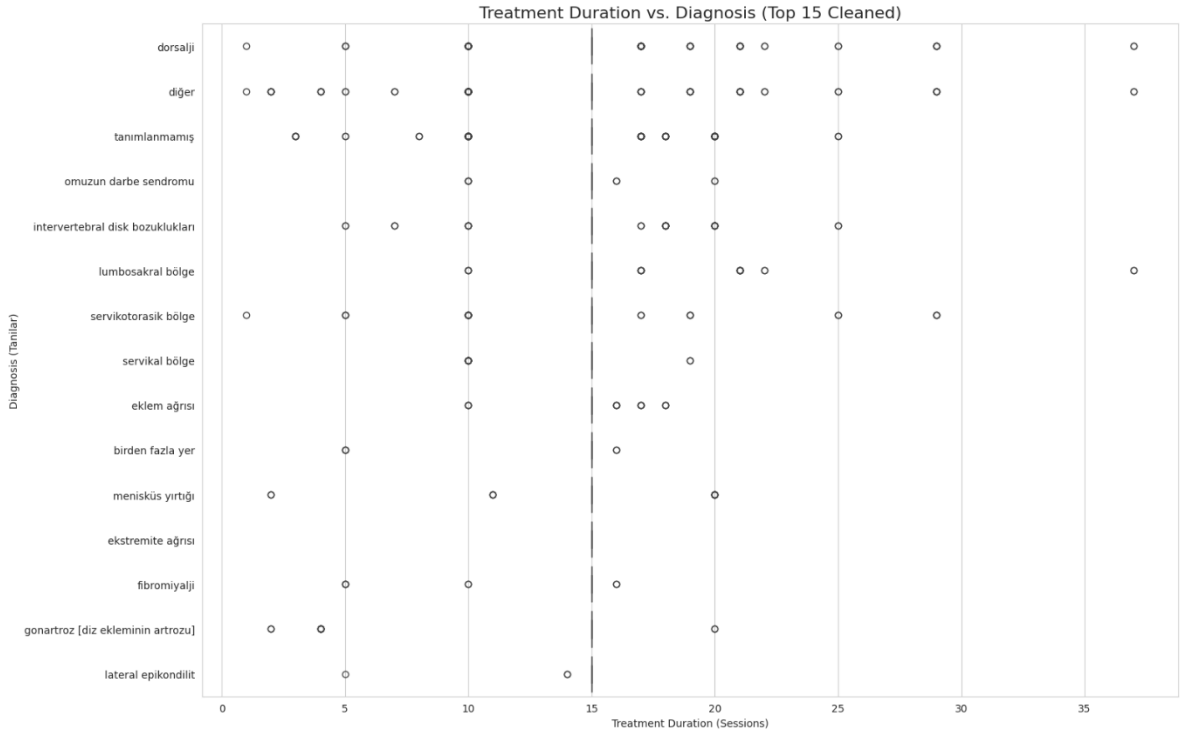


The majority of patients are from Türkiye and their treatment duration distribution mirrors the overall dataset. For the other nationalities, the number of patients is very small. While their box plots might look different, we cannot draw reliable conclusions from such small sample sizes.

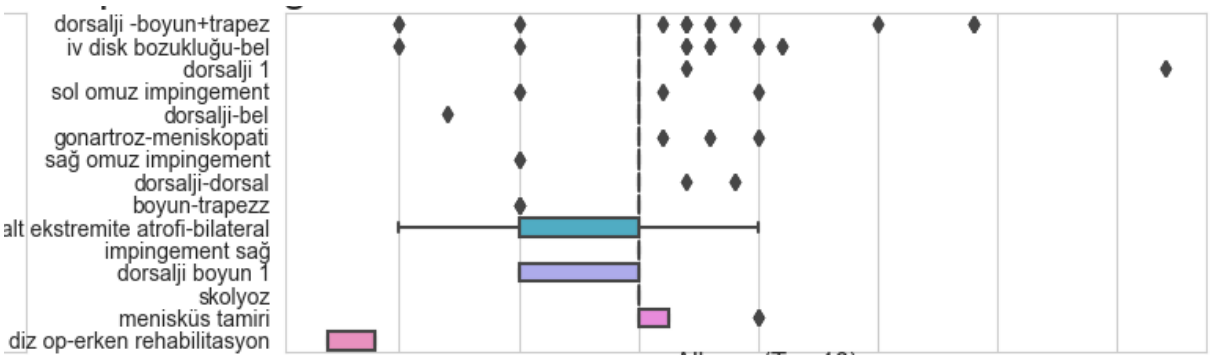


The main department, "Fiziksel Tıp Ve Rehabilitasyon,Solunum Merkezi", has a strong median of 15 sessions, which sets the baseline for the entire dataset. Other departments, like "Ortopedi Ve Travmatoloji" and "Nöroloji", also show a median of 15 sessions. However, some departments like "İç Hastalıkları" and "Kardiyoloji" appear to have slightly different distributions, with some showing a wider range of prescribed sessions. This suggests that the department a patient is in could be a moderately useful feature for predicting their treatment duration.

And, for the complex categorical features:



Certain diagnoses exhibit a much wider interquartile range. This means that while the median might be 15, the actual treatment plan is much more variable. Some patients might get 10 sessions, while others get 20 or 25 for the same diagnosis. This is a very strong signal for a predictive model.



Treatments directly corresponding to the diagnoses with high variability also show high variability here. This confirms that the specific treatment protocol, which is based on the diagnosis, is a key factor.

Second Phase: Data Preprocessing

Aim is to clean the data and transform it into a suitable format for machine learning models based on the insights from first phase.

Step 2.1: Data Cleaning

Handle Missing Values: I already changed some of the data and labeled them with "_cleaned". So, instead of filling original columns, i will work on the cleaned columns.

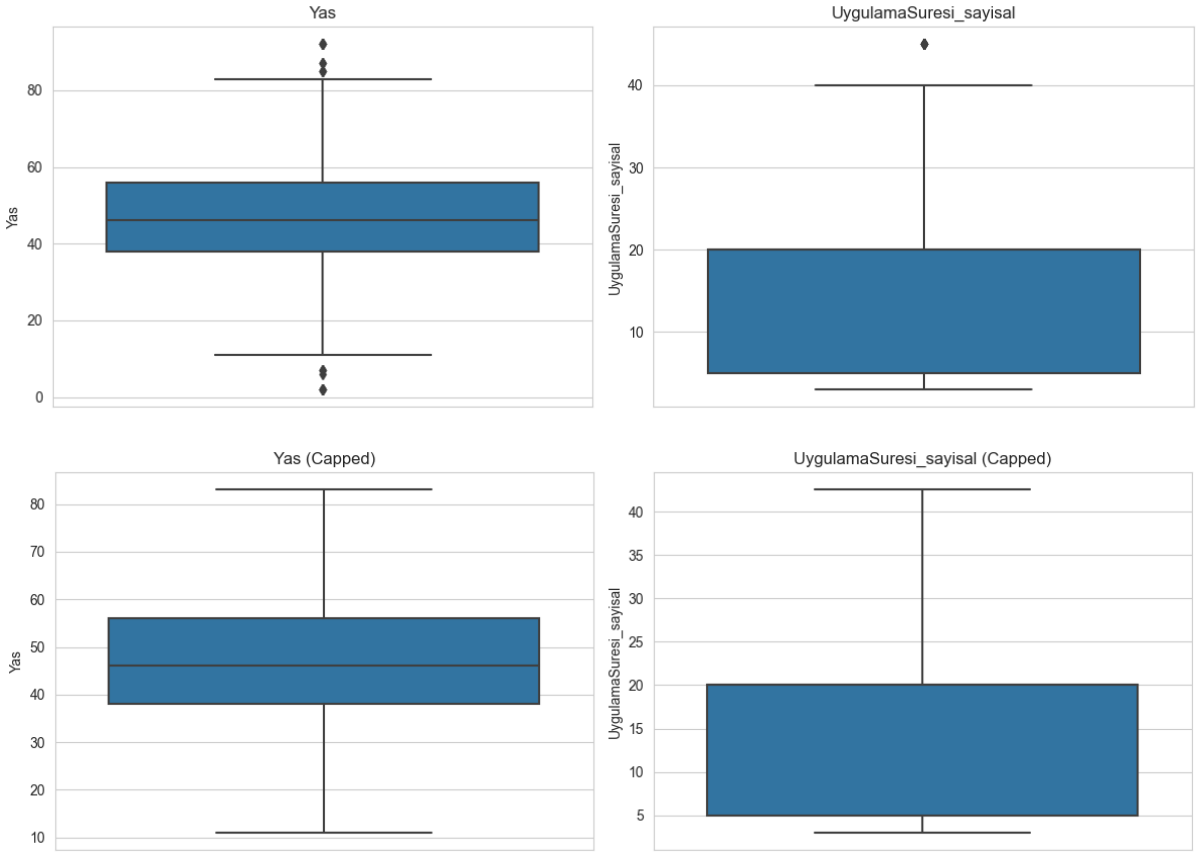
For "Tanilar_cleaned" and "UygulamaYerleri_cleaned", these are important features. So, i'll fill them with their most frequent value (mode) to preserve the overall distributions.

For "KronikHastalik" and "Alerji", blank entry in these columns most likely means the patient has no chronic conditions or allergies. So, i will fill them with "Yok".

For "KanGrubu", I filled the missing entries with "Bilinmiyor".

For "Cinsiyet", "Uyruk" and "Bolum", again i used the mode to fill the missing entries. Since "Uyruk" and "Bolum" have a dominant category.

Address Outliers: I used Interquartile Range (IQR) method and cap them at a reasonable upper limit. This means any value that is unusually high will be replaced by this upper limit, reducing its extreme effect.



Correct Data Types:

First, i checked to see current state of the data frame. During the analysis phase, new numerical versions are created for some of the columns as “_sayisal” tag. So, the original text columns are dropped. Same goes fort he original high-cardinality columns, since “_cleaned” versions are created for them.

Most of the text-based columns are stored as the generic “object” type. I changed them to a more memory-efficient “category” type.

Step 2.2 Feature Engineering

Parse Comma-Seperated Columns: My analysis showed that the specific diagnosis and treatment name had the biggest impact on target value. For these columns’ top 15, i created new “has_” columns with One-Hot Encoding. Also, i created a count feature for diagnosis(“Num_”). For columns with weaker relations, to prevent more noise signal, I only created count features.

Create Bins/Groups: For “Yas”, previous analysis showed that there is no linear relationship with the target value. However, that doesn’t mean there’s no relationship at all. It might be non-linear. So, I created new categorical columns for “Yas” divided to four different groups: “Adolescent”, “Young Adult”, “Adult”, “Senior”. The intervals are 0, 18, 36, 56, 100 respectively. And in the data set, I only hold three (“Young Adult”, “Adult” and “Senior”) columns. So if all of them are “false” for a row, then row’s age category must be “Adolescent”. So there aren’t any missing data.

Also, created a class for “TedaviSuresi”, since the analysis from EDA phase showed that applying a classification rather than linear regression will be more efficient. The groups fort his are “10_seans”, “15_seans”, “30_seans” and “Other”.

Step 2.3 Feature Transformation and Encoding

Categorical Variable Encoding: I applied encoding to all remaining non-numerical columns, which are “Cinsiyet”, “KanGrubu”, “Uyruk”, “Bolum”, “Age_Group”.

I used the “get_dummies()” method in pandas to perform one-hot encoding. After creating the new numerical columns, i dropped the original columns.

Numerical Feature Scaling: I applied StandardScaler in “sklearn” library to “UygulamaSuresi_sayisal”, “Num_Uygulama_Yerleri”, “Num_Kronik_Hastalik”, “Num_Alerji”, “Num_Diagnoses”.

Also used LabelEncoder to encode the “TedaviSuresi_Class” to change it’s values to numerical as well. Then dropped the class column.

Finally, i exported the “model_ready_data.xlsx”.

Final Review

After examination of the final output, i also prepared a refactored version of the code to handle pre-processing. Purpose of this is to create a more readable and organised code. Then i wrote a script that compares two xlsx files' to see that if they hold the same data regarding the column order. Both of the files are containing the same data.