Intensive Data Processing Systems - Assignment 2

Josep Maria Salvia Hornos - 48053698M

1 Introduction

In this assignment, we will be working with a CSV file containing information about hotels. The file includes various columns such as hotel name, location, star rating, price, and reviews. Our task is to clean the data and prepare it for further analysis. We will be using the Python programming language and Pandas library to perform the data cleaning process. The goal is to remove any inconsistencies, errors, or missing values, and create a clean and organized dataset that can be used for statistical analysis or machine learning algorithms. Through this exercise, we will gain experience in data wrangling, which is a crucial step in any data science project.

2 Requirements

1. pandas: pandas is a popular open-source data manipulation library used for data analysis and cleaning tasks. It provides a variety of data structures and functions for handling structured data, including data frames, which are similar to tables in a relational database.
2. scikit-learn: scikit-learn is a Python library used for machine learning tasks such as data preprocessing, feature extraction, model selection, and evaluation. It provides a range of supervised and unsupervised learning algorithms and is widely used in the data science community.
3. matplotlib: matplotlib is a Python plotting library used to create 2D graphics and visualizations. It provides a variety of visualization options, such as scatter plots, histograms, and bar charts, and is widely used for data analysis and communication of results.

3 Cleaning up NaNs

Before diving into the details of the data cleaning process, it is important to note that handling missing data is a critical step in ensuring the accuracy and reliability of data analysis. In this section, we focus on cleaning up NaNs (Not a Number) in the dataset. NaN values can be caused by a variety of factors, such as data entry errors, measurement issues, or missing data. Dealing with NaN values is essential for avoiding biased or erroneous analyses. Therefore, we present a step-by-step process for handling NaN values in our dataset.

3.1 Too many NaNs

In this section, we describe a systematic approach for handling NaN values in the dataset, which involves identifying columns with too many NaN values, dropping unnecessary columns, analyzing the remaining columns, and deciding whether it is reasonable for the data to be missing.

1. We identify columns with too many NaN values (>70%).
2. We drop columns with too many NaN values that are not useful for our analysis.
3. We analyze the remaining columns that have a significant percentage of NaN values.
4. We calculate the percentage of NaN values for each column and decide whether it is reasonable for the data to be missing.

3.2 Manual analysis of NaNs

Having discussed the process of identifying and dropping columns with too many NaN values, we now move onto the manual analysis of NaNs in the dataset. In this section, we focus on specific columns with a significant percentage of NaN values and investigate the reasons for their missing values. We then implement appropriate measures to handle these NaN values in order to ensure the accuracy and reliability of our analysis.

1. Not all hotels have WiFi, but the percentage of missing NaNs is the same for all columns except WiFi
2. After testing the hypothesis that NaNs are in the same rows, the decision was made to drop the rows with NaNs in the columns being analyzed
3. The 'topics' column was split into three columns since all topics were always three
4. The 'rooms' column had only two rows with NaNs, which were dropped
5. The 'listsaved' column had only ten rows with NaNs, which were dropped.

4 Type conversion

In this section, we address the issue of columns with incorrect data types in the original dataset. We identify columns with object data type that should be converted to numeric data types, and we implement appropriate measures to convert them to the desired data types. In addition, we address the issue of non-numeric characters or values in certain columns that need to be removed before converting them to the appropriate data types.

1. The original dataset contains columns with object data type that should be converted to numeric data types.

2. The 'wifi' column contains non-numeric characters, such as dots and commas, that need to be removed to convert the column to a float data type.
3. The 'rooms' column contains non-numeric values such as 'rooms' and 'habitaciones', which also need to be removed.
4. After cleaning the columns, all columns except 'rooms' are converted to float data type, while the 'rooms' column is converted to integer data type.

5 Hotel ID

In this section, we address the issue of duplicated rows with the same hotel ID but different values in certain columns. We implement a merge algorithm to merge these rows and create a dataset with unique hotel IDs.

1. We clean the dataset with hotel information by merging rows with the same hotel ID but different values in certain columns.
2. The merge algorithm involves discarding NaN values, computing the mode of the remaining values, and randomly selecting a value if there is no mode.
3. After applying the merge algorithm, the code drops duplicate rows and sets the hotel ID as the index.
4. The cleaned dataset has no repeated hotel IDs and is ready for further analysis.

6 Filling In missing values

Missing values can arise due to various reasons, such as data collection errors, incomplete data, or data processing issues. In this section, we address the issue of missing values in the numeric columns of the dataset.

1. The numeric columns in the dataset were analyzed, and several errors and weird values were identified.
2. A function called `clean_up_numeric_errors` was defined to clean up the errors.
3. The number of NaNs in each column was checked, and a decision was made to input the missing values with a linear regression model.
4. A linear regression model was created for each column that had missing values, and the missing values were predicted and filled in the original dataframe. Afterward, the `clean_up_numeric_errors` function was applied again to ensure that the predictions were in the correct range.

7 Data visualization

The distribution of numeric columns is plotted, and the issues with the 'reviews' column are discussed. The final dataset is described, and some final considerations are given.

1. The code goes column by column to check if everything is correct in the dataset.
2. Object and Numeric columns are printed separately to analyze their categorical or numerical values in the later step.
3. The distribution of numeric columns is plotted, and the issues with the 'reviews' column are discussed, resulting in dropping the column.
4. The final dataset is described, and some final considerations are given, such as scaling the price column, understanding the meaning of 'position' and 'listsaved' columns, and recognizing some numerical columns as ordinal.

8 Final Data and Considerations

1. Scaling the price column between 0 and 10 may be a beneficial approach to normalize the data and make the analysis more consistent.
2. The definitions of "position" and "listsaved" columns are unclear. However, if "position" refers to a measure of a hotel's positioning, it may be useful to scale it as well.
3. The number of rooms column appears to be slightly skewed, but is generally acceptable.

The final dataset contains several columns, including both categorical and numerical data. The classification column represents the continent, and the category column represents a subcategory of the continent. The remaining columns are:

- rooms: the number of rooms
- staff: staff score between 0 and 10
- comfort: comfort score between 0 and 10
- score: score between 0 and 10
- location: location score between 0 and 10
- stars: number of stars, an integer column (1, 2, 3, 4, 5)
- price: price score between 0 and 5
- services: services score between 0 and 10
- name: the name of the hotel
- country: the country where the hotel is located

- wifi: wifi score between 0 and 10
- value: value score between 0 and 10
- clean: clean score between 0 and 10
- listsaved: an unclear numerical column
- position: an unclear numerical column
- topic1, topic2, topic3: categorical columns representing selected topics from the hotel description.

It is important to note that some of the columns, such as the number of stars, scores, and price, could be considered either categorical or numerical depending on the context. In this case, these columns are best defined as ordinal columns because they have a natural order but are not continuous numbers. The number of stars, for example, has a natural order (1, 2, 3, 4, 5), but it is not possible to have 2.5 stars. Similarly, the scores columns and the price column have a finite number of values that follow a natural order, but are not continuous.

9 Conclusions

We have explored several aspects of data analysis, including data cleaning, data filling, data visualization, and data interpretation. We started by loading a hotel dataset and identifying the data types and data quality issues present in the dataset. We addressed these issues through data cleaning techniques such as removing duplicates, handling missing values, and correcting data formats.

In this case, we had several variables available that could potentially be used to predict the missing values, such as the hotel's location, comfort score, and staff score. Additionally, linear regression allowed us to estimate the values in a way that minimized the overall error between the predicted values and the observed values in the data set.

Next, we explored the dataset through data visualization, where we created several plots to understand the distribution of the variables.

Finally, we concluded our conversation by discussing the final dataset's columns and identifying whether they are numerical or categorical. We also explored the concept of ordinal columns and how they differ from categorical and numerical columns. Overall, we have explored different stages of data analysis, and the tools and techniques used in each stage to extract meaningful insights from data.