**Performance Assessment: NUM3 — NUM3 Task 1: Data Cleaning**

Bader Ale

Department of Information Technology, Western Governors University

D206 Data Cleaning

Professor Eric Straw

March 31, 2023

**Table of Contents**

[Part I: Research Question and Variables 3](#_Toc130848858)

[A: Research Question: 3](#_Toc130848859)

[B: Variable List and Datatypes: 3](#_Toc130848860)

[Part II: Data-Cleaning Plan (Detection) 5](#_Toc130848861)

[C1. Detection of duplicates, missing values and outliers 5](#_Toc130848862)

[C2. Detection of duplicates, missing values and outliers 6](#_Toc130848863)

[Table of Figures 7](#_Toc130848864)

[Works Cited 8](#_Toc130848865)

# Part I: Research Question and Variables

## A: Research Question:

For this performance assessment, my research question is:Is there a relation between the number of times the primary physician visited the patient during their hospital stay and the occurrence of readmission within 30 days following the patient's discharge from the facility?

## B: Variable List and Datatypes:

The following table contains all variables contained in the original dataset along with their respective datatypes and an example.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Datatype** | **Description** | **Example** |
| Unnamed: 0 | int64 | An integer used as an index in original dataframe | 0, 1, 2 |
| CaseOrder | int64 | Variable used to define order of cases | 0, 1, 2 |
| Customer\_id | object | An ID that defines a specific patient | C412403 |
| Interaction | object | Internal ID for patient and corresponding procedures | 8cd49b13-f45a-4b47-a2bd-173ffa932c2f |
| UID | object | Internal ID for patient and corresponding procedures | 3a83ddb66e2ae73798bdf1d705dc0932 |
| City | object | Patient's city of residence | Eva |
| State | object | Patient's state of residence in two letter code | AL (Alabama) |
| County | object | Patient's county of residence | Morgan |
| Zip | int64 | Patient's zipcode | 35621 |
| Lat | float64 | Latitude of patient's billing address | 34.3496 |
| Lng | float64 | Longitude of patient's billing address | -86.72508 |
| Population | int64 | Population within 1 mile radius of patient | 2951 |
| Area | object | Rural, Suburban or Urban | Suburban |
| Timezone | object | Timezone of patient residence | America/Chicago |
| Job | object | Job of patient or insurance holder | Psychologist, sport and exercise |
| Children | float64 | Number of children in patient's househol | 1.0 |
| Age | float64 | Age of patient | 53.0 |
| Education | object | Highest degree earned by patient | Some College, Less than 1 Year |
| Employment | object | Employment status of patient | Full Time |
| Income | float64 | Annual income of patient | 86575.93 |
| Marital | object | Marital Status of patient | Divorced |
| Gender | object | Patient Self-Identification (Male, Female, Non-Binary | Male |
| ReAdmis | object | If patient was readmitted within 30 days of discharge | No |
| VitD\_levels | float64 | Patient Vitamin D levels (ng/mL) | 17.80233 |
| Doc\_visits | int64 | Number of times patient was visited in hospital by doctor | 6 |
| Full\_meals\_eat | int64 | Number of full meals eaten by patient in hospital (0=partial meals) | 0 |
| VitD\_supp | int64 | Number of times Vitamin D supplements were given to patient | 0 |
| Soft\_drink | object | Whether patient drinks 3 or more soft drinks in a day | Yes |
| Initial\_admin | object | How the patient was admitted to hospital (emergency, elective, observation) | Emergency Admission |
| HighBlood | object | Whether patient has high blood pressure | Yes |
| Stroke | object | Whether patient has had a stroke | No |
| Complication\_r | object | Level of patient's acuity assessed by doctor | Medium |
| Overweight | float64 | Whether patient is considered overweight (Yes=1, No=0) | 0.0 |
| Arthritis | object | Whether patient has arthritis | Yes |
| Diabetes | object | Whether patient has diabetes | Yes |
| Hyperlipidemia | object | Whether patient has hyperlipidemia | No |
| BackPain | object | Whether patient has chronic back pain | Yes |
| Anxiety | float64 | Whether patient has chronic anxiety (Yes=1, No=0) | 1.0 |
| Allergic\_rhini | object | Whether patient has allergic rhinitis | Yes |
| Reflux\_esophag | object | Whether patient has acid reflux | No |
| Asthma | object | Whether patient has asthma | Yes |
| Services | object | What service patient received while hospitalized | Blood Work |
| Initial\_days | float64 | Length of Stay of patient | 10.58577 |
| TotalCharge | float64 | Average daily amount billed to patient | 3191.04877 |
| Additional\_cha | float64 | Average daily amount billed to patient for additional procedures | 17939.40342 |
| Item1 | int64 | Survey Answer: Timely Admission (1 = most important,8 = least important) | 3 |
| Item2 | int64 | Survey Answer: Timely Treatment(1 = most important, 8 = least important) | 3 |
| Item3 | int64 | Survey Answer: Timely Visit (1 = most important, 8 = least important) | 2 |
| Item4 | int64 | Survey Answer: Reliability (1 = most important, 8 = least important) | 2 |
| Item5 | int64 | Survey Answer: Options (1 = most important, 8 = least important) | 4 |
| Item6 | int64 | Survey Answer: Hours of Treatment (1 = most important, 8 = least important) | 3 |
| Item7 | int64 | Survey Answer: Courteous Staff (1 = most important, 8 = least important) | 3 |
| Item8 | int64 | Survey Answer: Evidence of active listening from doctor (1 = most important, 8 = least important) | 4 |

# Part II: Data-Cleaning Plan (Detection)

## C1. Detection of duplicates, missing values and outliers

In this section, we will discuss the methods (functions) were used to detect for duplicates, missing values and outliers. We started by importing the pandas python library into our notebook. The method that was used to detect duplicates was the .*duplicated*  and *.value\_counts()* method; the *.duplicated* method allows for the detection of duplicated values in a dataframe while *.value\_counts* allowed for a counts of these duplicated values, if any (NumFOCUS, Inc, 2023). As shown in the code snippet below, no duplicates were found. This was done by comparing the shape of the dataframe by running the *.shape* command with the output of *.value\_counts.* Moreover, the *.duplicated* method itself would return a Boolean value if any duplicated values were to exist.

Text

Description automatically generated with medium confidence

Figure 1: Shape of Original Dataframe

Graphical user interface, text, application

Description automatically generated

Figure 2: Duplicated Method

In order to detect missing values in the data frame, the .*isnull().sum()* command was used. This command involved detecting if any missing values existed and if so, total them by variable.

A picture containing logo

Description automatically generated

Figure 3: Code Snippet for Detecting Missing Values

From this, the variables **Children**, **Age**, **Income**, **Soft\_drink**, **Overweight**, **Anxiety** and **Initial\_days** were found to have missing values. Children, Age, Income and Initial days were quantitative variables (there were numerical in nature) while Overweight, Soft\_drink and Anxiety were qualitative as they were of the Yes/No kind although Soft\_drink was not re-expressed meaning the records were either yes or no. To solve this, ordinal encoding was used in order to re-express “Yes” as 1 and “No” as 2 (Middleton, 2022). Lastly, outliers were detected by way of visual inspection using the Seaborn boxplot function for all quantitative variables described. This involved importing the seaborn package into the Jupyter notebook and using the appropriate command to plot the graph (Waskom, Ph.D, n.d.).

## C2.

The functions and methodology used for the detections of duplicate values, missing values and outliers was selected by inspecting the pandas library documentation for any appropriate command that would achieve the desired outcome (NumFOCUS, Inc, 2023). Moreover, the PowerPoint slides provided in the course dashboard were used as a reference point for achieving the goal. The detection of duplicates and missing values was accomplished by incorporating the Python code specific to the *pandas* library, the boxplots were achieved using the specific *seaborn* package and the PCA analysis was achieved via the *SciKit* package.

## C3.

The programming language used to clean the data was Python and its various libraries. This decision was based on prior experience in working with Python and the Anaconda package manager. Moreover, the IDE used was Visual Studio Code.

# Table of Figures

[Figure 1: Shape of Original Dataframe 5](#_Toc130848820)

[Figure 2: Duplicated Method 5](#_Toc130848821)

[Figure 3: Code Snippet for Detecting Missing Values 6](#_Toc130848822)

# Works Cited

Middleton, D. (2022). Webinar 3: Getting Started with Reexpression of Categorical Variables. Retrieved March 2023, from https://westerngovernorsuniversity.sharepoint.com/:b:/r/sites/D206DataCleaning/Shared%20Documents/Webinar%203%20Getting%20Started%20with%20Re-expression%20of%20Categorical%20Variables.pdf?csf=1&web=1&e=r3QoNq

NumFOCUS, Inc. (2023). *User Guide to pandas*, 1.53. Retrieved March 2023, from pandas: https://pandas.pydata.org/docs/user\_guide/index.html#user-guide

Waskom, Ph.D, M. (n.d.). *seaborn.boxplot*. Retrieved March 2023, from seaborn: statistical data visualization: https://seaborn.pydata.org/generated/seaborn.boxplot.html?highlight=boxplots