**Outlier Detection**

**Overview**

We were curious about the possible outliers inside of the dataset. The data file of Alzheimer’s Disease patients contains a description of the topic the patient discussed with their healthcare provider, their demographic information, and location.

One approach to outlier detection is to treat the description of the topic the patient discussed with their healthcare provider, their demographic information, and location as a representation of that person. In other words, we wanted to know what combinations of demographic information, location, and topic discussed are the least common in the dataset.

It is important to note that several different people can have the same description. Several different people can live in the same area, have the same demographic information, and see their healthcare provider for the same reason.

To detect outliers with non-numeric data, we took a frequency-based approach and created a plot of all the descriptions of the patients and the number of times a patient inside the dataset matched that description.

By looking at the chart, we did detect outliers at the bottom right side of the chart. We also generated a CSV file of the data for further research.

**Data preprocessing**

When we call the **def detect\_outliers():** method, the first thing it does is call the **get\_patient\_frequency\_dictionary():** to begin to preprocess the data.

We only care about the patient’s location, demographics, and the topic they discussed with their healthcare provider. Likewise, we also want to efficiently keep track of the counts that the combinations of location, demographics, and topic discussed occurred inside of the dataset. So, we decided on using a dictionary as our data structure to simultaneously keep track of a generalization of the description of a patient and the frequency that it occurred.

Just like when we mined the association rules, we removed the commas inside the quotation marks inside of excel before we ran the program. This way we can successfully split the strings from the Alzheimer’s CSV file by comma.

Once we did that, we looped through the CSV file of the Alzheimer’s patients. During each iteration of the loop, we created a list of the relevant data to be passed to the **def get\_patient\_type(helper\_list):** method.

The purpose of the def get\_patient\_type(helper\_list): method is to generate the strings that represent a generalization of a type of patient, while also preventing quotation marks or empty strings being added to the strings.

After generating a string that represents a generalization of a type of patient, we added the representation of the description to the dictionary or updated the count inside of the dictionary to keep track of the amount of people that matched the description of a person already inside the dictionary. We repeated this process until we reached the end of the Alzheimer’s CSV file.

Once the get\_patient\_frequency\_dictionary(): method generated the dictionary with the relevant information, it was sent to the detect\_outliers(): method.

Once the dictionary reached the detect\_outliers(): method, we sorted it in descending order of frequency so that we can more easily detect the outliers and so that it is more organized once we export the patient type frequency dictionary as a CSV file.

**Data processing**

To process the data, we imported the pandas and matplotlib libraries.

We converted the patient type frequency dictionary to a DataFrame so that we can call the def **create\_frequency\_bar\_graph(patient\_type\_frequency\_data\_frame):** method so that we can create a bar graph to easily visualize the outliers inside of the Alzheimer’s data.

Since there were around 17586 strings that represented a generalization of the types of patients inside of the dataframe, we used an integer index to represent each of the 17586 strings.

We then use matplotlib and the DataFrame to create a bar graph of the number of times the generalization occurred inside of the data set by the generalizations of the patients.

The bar graph illustrated the outliers that existed in the data on the bottom right side of the graph. We then exported the patient type frequency data from the dataframe as a CSV file so we can further analyze the outliers.