**Association Mining**

**Description/Conceptual**

We were curious about is what locations, lifestyle factors, and groups of people are most strongly associated with Alzheimer’s disease. Association mining is one approach to discovering the strongest correlations to Alzheimer’s disease.

By using the Apriori algorithm on the data, we generated the strongest factors associated with Alzheimer’s disease, generated a CSV file and bar graph of the associations. We were then prepared to conduct an analysis on our CSV file and bar graph.

**Data Preprocessing**

During the data preprocessing phase, we realized that there were three columns that were relevant to our question. The column called LocationDesc which contains the data that corresponds to where the patient lives, the column called Topic which is the reason why the patient is visiting their healthcare provider, and the column called Stratification2 which is the demographic information of the patient.

To use the Apriori algorithm on only the relevant data, we created a method called def **get\_matrix\_of\_location\_diagnosis\_and\_demographics():** to generate a multidimensional array that only contained the patient’s location, demographics, and topic discussed with their healthcare provider.

To generate the multidimensional array, we traversed the CSV file of the Alzheimer’s data line by line and extracted only the locations, demographics, and topic discussed with their healthcare provider. To traverse the CSV file of the Alzheimer’s data, we had to split the contents of the CSV file by comma.

Inside of the CSV file, there were strings that took on the form “"White, non-Hispanic"”, “"Disability status, including sensory or mobility limitations"”, and several other variations of this. This was a problem when traversing the CSV file because it was preventing us from splitting the data by comma. For example, “"White, non-Hispanic"” is supposed to be inside a single column, but since there was a comma included in the string, it was treating “"White”” and ““non-Hispanic”” to be in two different columns.

Since there were commas contained directly inside quotation marks of some of the data inside the CSV file, we needed to go inside Microsoft excel and press ctrl f. After we clicked ctrl f, we had to click replace “,” with “”.

After using excel to remove the excess commas, the strings that took on the form “"White, non-Hispanic"” among several other variants became “"White non-Hispanic"”. This was very helpful, since we could now properly use commas as our delimiter when traversing the CSV file, but there were still quotation marks around the strings that took this form.

When preparing the final output, we want to output White non-Hispanic and not “White non-Hispanic”. Likewise, there were several thousand values for demographics, location, and topic discussed with their healthcare provider that was missing from the data.

As a remedy for this, we created a method called **def remove\_unneeded\_characters(inner\_list):** to remove the missing data values and remove the quotation marks from any of the values inside the columns that were relevant to us.

**Data Processing**

The libraries that we imported for the data processing step were mlxtend, pandas, and matplotlib. We also imported TransactionEncoder from mlxtend.preprocessing and apriori from mlxtend.frequent\_patterns.

Mlxtend is an open-source library that contains an implementation of the apriori algorithm. Mlxtend’s TransactionEncoder class also allowed us to transform our data into a usable form for the apriori algorithm.

We processed the data in the def **associations\_location\_diagnosis\_and\_demographics():** method. Once we generated a clean multi-dimensional array that contained only the locations, topics discussed with the healthcare provider, and demographic information, we were not able to do anything with it yet.

The TransactionEncoder class created a pandas DataFrame that was a Boolean equivalent of our two-dimensional matrix. For each value in the Boolean DataFrame, a value of true would be contained if a patient had a certain demographic, location, or topic discussed. Likewise, for each value in the Boolean DataFrame, a value of false would be contained if the patient did not have a certain demographic, location, or topic discussed.

Once we had the Boolean equivalent of our two-dimensional matrix, we applied the apriori algorithm to create a third DataFrame that contained all the frequent patterns that have a minimum support of .02.

To prepare our findings for output, we converted the values of the DataFrame that contained the factors that are most strongly associated with Alzheimer’s disease to strings.

We then called the **plot\_alzheimers\_associations\_as\_bar\_chart(alzheimers\_associations):** method to use matplotlib to create a bar graph of the lifestyle factors, demographics, and geographic locations associated with Alzheimer's disease by using our DataFrame. We used support as our value for the x axis and the value associated with Alzheimer’s disease as our y value.

Before terminating the program, we exported a CSV file of the factors most strongly associated with Alzheimer’s disease from the DataFrame for further research.