The data visualization portion of my analysis, much like a lot of any data science projects, can be considered an extension of data processing. Examining trends in the data that prove to be both helpful and unhelpful, and exploring the helpful insights in more depth.

The overarching goal of this capstone project is to find trends in the predictor variables of the data as they relate to the target variable CLASS. The data wrangling and processing methods described previously just sought to trim the unhelpful feature variables and data from the dataframe; there was not much attention given to the binary variable. The data visualization aspect of this capstone project will proceed with specific attention given to the target variable.

Specifically, my analysis will proceed by looking at how each of the remaining 21 feature variables relate to the target variable. To this end, I had a distinct way in which I wanted to initially visualize these potential trends.

I set out to visualize the count of the unique values of each feature- how many times did a specific value occur in a given feature variable. I decided that the best way to accomplish this visualization was with a contingency table of a predictor variable and the target variable casted as a seaborn heatmap. A contingency table can be easily configured through Panda's cross tabulate function.

I constructed a function containing an algorithm that took my cleaned and processed dataframe and cross tabulated a predictor variable and the CLASS target variable. Once the data was in this cross tabulated form, the seaborn heatmap could be produced.

I iterated through the entirety of the dataframe's feature variables to cross tabulate them with the target variable and then plot a heatmap of the table to visualize. The way my contingency table was constructed had values corresponding to the number of occurrences of the *top 5* most frequently occurring values of the feature variable. The heatmap was split into rows (0 and 1) of the binary classification target variable to view the counts in each classification. This singular visualization proved to not be as effective as I had previously hoped. While I did see some values dominating certain feature variables, the counts were not as enlightening as say a *percentage* might be.

My next visualization tactic had this in mind. Instead of looking at pure counts of values in each feature variable as they relate to the target variable, I would instead look at percentages of values. I wanted to look at the values in terms of two percentages: the percentage of the value count in regards to the *total* number of observations in the feature variable (i.e. this value occurred 14% of the time in this feature variable) and the percentage of value count in regards to how the values split between the target binary variable (i.e. this value appeared 60% in agreement classification and 40% in disagreement classification). Again, these visualizations

would both be heatmap and thus the algorithmic intensity and computation to achieve them did not increase too much.

Viewing these 3 heatmaps per feature variable proved to be very useful. Each heatmap provided more insight into how the values of each feature variable broke down with the feature variable and how they related to the binary classification. As stated beforehand, the visualization process served to be more data processing in preparation for applying ML alogirthms. Below will be a description of the processing resulting from the visualization.

After inspecting all 21 feature variables, I was able to cut even more variables that I decided were going to be unhelpful in the ML process. My first observation was in the heatmaps corresponding to the feature variables BIOTYPE and Feature_Type. The visualization showed me that these feature variables only had *one* unique value each and therefore would not provide any help in a binary classification prediction. These two feature variables were dropped from the dataframe. Furthermore, the feature variables ORIGIN and CLNVC displayed only two unique values each, with one unique value constituting most of the total percentage breakdown (similar phenomenon to the previous 2 feature variables). Again, these two variables were dropped in addition. On the opposite side of the filtration spectrum, there were 5 more feature variables with far too many unique values. I was able to see this in the percentage-of-total heatmap were even the darkest areas of the map only represented approximately .05% of the values. As mentioned in the data wrangling and processing phase, too many unique values could lead a model to overfit the data and perform poorly. Therefore, these feature variables were also dropped.

By the end of the visualization, I was able to drop 9 more feature variables (current count of 12) from the dataframe and keep those that displayed the strongest correlations in their values to the target variable. Of these feature variables, I will perform a final statistical analysis on them to determine from a quantitative stance if their correlations to the target variable are in fact relevant and should be pursued in the ML algorithm.