The dataset I am using for my capstone project is from Kaggle and therefore relatively clean already. However, there was still quite a bit of data wrangling and processing to be done. The overarching goal for the wrangling and processing is to filter out the features that are going to be unhelpful in the future analysis.

An info call on the dataframe reveals that there are 65188 total records with a total of 46 columns (variables) measured. In a data science sense, there are 45 feature or predictor variables and 1 target variable- the CLASS variable. Looking into the dataframe, I immediately see that there are several mismatched non-null entries between the feature variables. This will pose to be the biggest challenge of the data wrangling and processing.

As in any data science endeavor, the most comprehensive analysis includes every feature variable and every entry. Unfortunately this result is not always achievable. For instance, there are several feature variables that have less than 90% or even 80% of the maximum non-null entries, and more that have even less. Inspecting these variables that lack all of the entries, I find that there is no immediately obvious way to intelligently fill all or even most of these missing entries. There are simply too many unique values to safely fill missing entries. This brings about another issue to deal with that will be explained shortly.

Re-broadening the scope of the wrangling- the most obvious variables that I need to cut are those with several null entries. I extend this to define a threshold of non-null entries that a feature variable must have in order to remain in the analysis process. This threshold must filter out enough of the unimportant variables, while simultaneously retaining most of the dataframe information. However, I cannot keep (for example) all variables with > 50% of maximum number of non-null entries. Due to the fact that I will have to drop *all* observations (rows) that contain *one* null entry, I need to be careful about setting the filter threshold too low. This would result in a large cut of data after the null-entry dropping on the remaining dataframe.

I set a threshold of > 50000 non-null entries. This is around 76% of the maximum observations, but keep around 70% of the original number of feature variables. This is after dropping 8 variables that were roughly 90% null-entries. At this point my dataframe contains 30 feature variables and, after dropping all observations that contain any null-entry, 44000 observations. Overall, this is a great retention of the bulk of the data and I shed a lot of the unhelpful weight of the data.

This concludes the wrangling process of my analysis. However, I still must process the data so that I can focus on solely those features that have the greatest impact on the binary variable CLASS.

As previously and briefly mentioned, there were feature variables that contain several unique values. Not all of these feature variables were cut during the wrangling process, and will be counterproductive in the machine learning (ML) portion of the analysis for the following reason: if there are proportionally too many values in a given variable many ML models can *overfit* this data, which can lead to large computation times or a poorly-predicting model. Therefore, the unique values need a threshold filter as well. Initially I choose a number of unique values per feature to be that of approximately 10% (4000) of the number of current observations (44000). I say initially, because this number may need to be lowered or raised based upon future analysis. Applying this unique filter I get a dataframe with 21 predictor variables (and again the 1 target variable CLASS).

The data wrangling and processing cut about half of the data feature variables by applying 2 filtration thresholds. Thus in the end I have a dataframe with 22 variables, each with 44572 observations. This proves to still be quite a lot of data to work with as I move forward in the analysis.