Started by arranging the data in 2 different directories, “dataIN” and "dataOUT." The “dataIN” directory has the "facebook_dataset.csv," "google_dataset.csv,” and "website_dataset.csv." The “dataOUT” directory will have the final 4th dataset.

The next step was observing the data, seeing that all of them were CSV's. I noticed that the Facebook and Google datasets were correctly formatted, with the values being split into different columns.

When it comes to the "website_dataset,” we can notice that we have one big column with the data separated by “;” and other data scattered across random columns. First, I merged the data into one column, so the randomly scattered data and the data in the first column can create complete data. I used the Excel “textjoin()” function. Then I split the information based on the labels provided, such as "root_domain," "domain_suffix," etc.; for that, I used the Excel “Text to Columns” feature and used the “;” as a delimiter. Now we have clear data for further computation.

Trying to load the dataframes into Python using Pandas proved to be a challenge due to the way the data is formatted. Using the argument “on_bad_lines=’warn’” allowed me to load the data and receive warnings for each of the badly formatted data. The null values are also being replaced with “NaN.”.

The next step was the decision on what join to use and what the join should take place on. First, I merged the Facebook and Google datasets. I decided to join based on the “name” value since most of the firm’s names tend to be unique and most firms trademark their name. Next, for the type of join, I went with an outer join since I wanted all firms to be included even if there is no match between the two datasets, to avoid losing important data. If the name of the firm is in both datasets, then the information will be added from both datasets.

After joining, the Python program looks over the Google dataset and fills in the missing values with the ones from the Facebook dataset. If there is already a value in the Google dataset, then that value will be used; if both of them are missing, then NaN will be placed. I also renamed the column “categories” from the Facebook dataset to “category” to match the Google dataset.

After that, I renamed the "root_domain," "main_city," "main_country," "main_region," and “s_category” columns from the website dataset to match the ones from the “merged” dataset. I used an outer join on the “domain” and merged to two datasets to create a final one that has all the three datasets inside.

After that, I only kept the 'category', 'address', 'country_name','region_name', 'phone', and 'name' columns and ordered them alphabetically based on "category.”.

The reason I only kept these columns is because they were the ones with the most relevant information referring to the assignment. Colums such as "raw_phone" from the Google

dataset were more nicely formatted than the "phone" I decided to keep, but the amount of data was smaller.