Data Wrangling and Cleaning Steps Utilized to Create the df_final.csv Dataset

The data analysis project originally was looking to create a supervised model to determine predictive factors that significantly influence certain types of negative events that occur with medical devices. However, it was decided instead to create an unsupervised prediction model using NLP to predict whether a medical device will be issued a recall, a safety alert, a field safety notice, or a combination of two or more of them as well as the event risk class associated with them. For the purpose of data exploration and of the main analysis of this project, the df_final.csv file was merged and cleaned from three CSV files (devices-1562662526.csv, events-1562662544.csv, and manufacturers-1562662522.csv). Below describes the data wrangling and cleaning steps and methods used to create the final df_final.csv file.

Using IPython Notebook, devices-1562662526.csv, events-1562662544.csv, and manufacturers-1562662522.csv were read into dataframes called devices, events, and manufacturers respectfully. The devices dataframe had 15 columns and 104,066 rows, the manufacturers dataframe had 10 columns and 26,013 rows, and the events dataframe had 30 columns and 109,574 rows. The devices and manufacturers dataframes were merged (in an outer join fashion on manufacturer_id) to create the df dataframe which had 24 columns and 104,137 rows. Then, the df dataframe was merged (in an outer join fashion on device_id) with the events dataframe to create the df_final dataframe which had 53 columns and 109,645 rows.

Columns in the df_final dataframe were removed based on 60% missing percentage criteria as well as whether or not the columns would contribute to the data exploration and the main purpose of this project. By a boolean array, columns with a missing (null) percentage of 60% or more were removed. Then, it was decided that device_id (ID of the device), device_name (name of device), device_country (country where device was created), event_id (ID of event), action_classification (event risk class), event_country (country where event took place), reason (textual reasons device is under investigation or reported), and type (event type) would be kept in the df_final dataframe while the others were dropped. This left the df_final dataframe with 8 columns and 109,645 rows.

The columns were then checked and reorganized based on inconsistent and missing values. The values of the action_classification (event risk class) were reorganized to make its string values more consistent. For example, “I,” “Class I,” and “Class 1” all denoted the category, Class 1, under the action_classification column, and thus “I” and “Class I” were changed into the string value of “Class 1” to indicate they were all of the same categorical value. Additionally, there were outlier values of “Unclassified Correction” and “Voluntary recall.” As there were only four of them, the rows containing “Unclassified Correction” and “Voluntary recall” under the action_classification column were removed from the df_final dataframe since their removal won’t affect the analysis results of this project. Finally, the rows containing the null values of the action_classification, type, and reason columns were removed from the df_final dataframe. This leaves the df_final dataframe with 8 columns and 52,267 rows.

Text cleaning was done on the reason column to prepare the df_final dataframe for the NLP analysis of the project. First, a function called, clean_text(), was created to keep only alphabetical words, remove whitespaces, and convert the text to lowercase before it was applied on the reason column with the new column called, clean_reason, displaying the cleaned text.

Next, a function called, stem_text(), was written to help to stem the words of the text under the clean_reason column. Finally, the remove_stopwords() function was created to remove the stop words of the text under the clean_reason column. The head() method was used on the df_final dataframe to check how the merged and cleaned dataset came out by seeing the top five rows of the dataframe.

The final df_final dataframe was exported as a CSV file named, df_final.csv.