For part A, we first decide we are going to predict whether a case is outbreak related or not. Within the program, there are five helper functions. First one is called get_data() which will obtain all data from database and transform it into pandas DataFrame.

After that, for question1, I use function called plot_histogram to plot a histogram. In this function, it takes all data we get from last function and show total number of both true and false for label which is outbreak related.

For handling missing values, since we have handled missing values for age_group, acquisition_goup, and outbreak_related when we import them to databases, we only need to handle missing values in retail_and_recreation, grocery, parks, transit_station, workplaces. I use SimpleImputer from scikit-learn to handle them. And use mean value to replace missing values.

For handling categorical attributes, I use get_dummies from pandas which is one-hot encoding technic to handle them.

For normalization of numeric attributes, I use MinMaxScaler from scikit-learn. For each numeric attribute, I first separate them from original data and reshape it. And after normalization, I combine it with original data.

For feature selection I first drop label from data, and for rest of them, I use SelectKBest from scikit-learn to remove potentially redundant attributes.

Finally, I split train set and test set using StratifiedShuffleSplit. As we can see from the result, the distribution of label is imbalanced. There are 91571 false and only 18374 true for outbreak_related. Therefore, I use near miss algorithm to balance the class distribution. In order to do this, I use NearMiss from scikit-learn.

During the preprocessing part, we can see the class distribution is unbalanced which is why we have to do the under sampling using near miss algorithm. And there are also a lot of missing values in mobility dimension.