CMPT 459 Milestone 1 Report

Hazem Hisham, Harry Preet Singh, Jiongyu Zhu

1.1 For the process of cleaning messy outcome labels, we will be taking a look at the column "outcome" from our csv file, and create a new column "outcome_group". Four types of outcome groups are created, and each entry in the csv file gets assigned an "outcome group" value based on the boolean value of if they belong to a certain "outcome group" based on their "outcome" column. After accessing every entry, we drop the "outcome" column as it is no longer needed. Number of cases in each outcome group: Hospitalized: 135726, Nonhospitalized: 779, Deceased: 4031, Recovered: 65310

1.2 The prediction of the outcome_group labels in the cases_2021_train.csv and cases_2021_test.csv datasets is best described as the task of classification. We have a classifier model that determines the class of an object based on its attribute (outcome), and our goal is to assign a new attribute(outcome group) as accurately as possible.

1.3 In this section we really dove into the three datasets to visualize them and really see what they look like. All three datasets were visualized using a mix of line plots, histograms, barplots, and maps. The maps were a nice touch for us to see them geographically. It seems to be that India and the Philippines account for the majority of the dataset. We plotted distributions of our outcome group and also visualized the class hospitalized and non hospitalized against many features. An interesting observation we noticed is that India may have the most cases, but actually it has little to no non-hospitalized cases, meaning all the cases are either hospitalized, recovered or deceased.

1.4 Here we focused on cleaning the data, handling all the nans and re-formatting certain attributes. We changed some countries back to proper name (Taiwan* -> Taiwan) and dropped the null values we had in age simply because there was a massive number of them. As for latitude and longitude we ensured that their data types were float. As for the missing values in source and additional information, it was replaced with "None" and in a similar case for those values with missing province, we subbed in the country name for it as some countries do not have provinces (ex: Yemen). The numerical attributes were imputed using the mean for the most part. For provinces that had NA values in location data, we took the coordinates of that entry, found the nearest province in the train and test data to that entry, and replace the province in the location data with the province of the test and train data using haversine formula.

1.5 For detecting and removing the outliers for cases_2021_train.csv, cases_2021_test.csv, location_2021.csv. We check every single quantitative column to detect if some values are more out of line compared to others. Since our data is not evenly distributed but rather skewed, we can not use Z-score, we ended up using the interquartile range (IQR).For cases_2021_train.csv and cases_2021_test.csv, we decided to detect the outliers on 3 attributes: age, latitude and longitude. If we put latitude and longitude as one of the outlier columns, we would further reduce our data by 25%, which is far more than the target within 5% we wanted. Thus in the end, we decided to not use latitude and longitude to detect outliers in cases_2021_train.csv and cases_2021_test.csv, only using age which reduced 2% of the outliers for us.For location_2021.csv, we had a lot more quantitative columns to work with. After testing: Confirmed, Active, Latitude, Longitude, Incident rate and Fatality rate. We decided to use both incident_rate and case_fatality_rate, as these attributes give us a ~5% reduction for outliers.

1.6 CSV can be merged using pandas method pd.merge() with the how parameter set to inner. Specifying a column or a list to merge on, and which type of merge to use.In our project, we chose to merge cases_2021_train.csv and cases_2021_test.csv on location_2021.csv with the list [‘country’, ‘province’], and return new data frame containing all rows from cases_2021_train.csv and cases_2021_test.csv including those rows also who do not have values in the right dataframe and set location_2021.csv column value to NAN. The reason behind this decision is because we think the location information serves as an enhancement to the specific cases, instead of the other way around. This is because we have attributes like Active, deaths, etc.. that were now added. Report the number of rows (cases/samples): Cases_train: 19,544 , Cases_test: 9,665

1.7 To do feature selection we decided to compute mutual information scores. We found a way online to compute the mutual information scores by encoding the categorical features. In our case, since this is a classification task, our target vector outcome_group is also categorical. So we had to encode both the categorical data from the cases_train merging with location (called merged_train in code) and the outcome group vector from merged_train. By encoding it we were able to obtain its MI scores. Now we also did the same for the numerical/continuous features like age, latitude, longitude, etc. However in this case we didn’t have to encode anything because these features are numerical from the get go.We produced plots of the MI scores and found that we should be dropping age, sex and chronic_binary. Everything else will be selected. Since this algorithm is very stochastic, we ran it a few times and averaged the scores out.