CMPT 459 - Martin Ester - Spring 2021
Saqib Aziz Dhuka - 301338669
Cole Stankov - 301295209
Richard Dong - 301317247

Milestone 1 report

Section 1.1

For the train data, the file contained individual covid-19 cases for each row of the dataset. We displayed the information for countries and the outcomes as a bar graph and we also used a scatter plot to plot latitude and longitude to produce a type of heat map for cases. For date, age, sex, we plotted them against frequency (number of cases) using a bar graph. Lastly, we plotted country-province against frequency, however, it did not contain any useful information. Upon reviewing the plots we found that the important columns were “age”, “sex”, “province”, “country”, “latitude”, “longitude”, “date_confirmation”, and “outcomes”. There were a lot of missing data for a lot of the features. For the “Age” column we noticed that it didn’t have a consistent format on how they input the age (Ex. 35, 0.666, 20-30, etc.). This formatting issue was also similar in the “date_confirmation” column where ranges were used in some cases, however, it was not as severe.

As for location data, the file contained country-province covid-19 cases for each row in the dataset. We displayed the latitude and longitude similar to train data and for “Confirmed”, “Deaths”, “Recovered”, “Active”, “Incidence_Rate” and “Case-Fatality_Ratio” columns we displayed the frequency of each for the corresponding country-province. Upon reviewing the plots and data we found the important columns for “Province_State”, “Country_Region”, “Lat”, “Long_”, “Confirmed”, “Deaths”, “Recovered”, “Active”, “Combined_Key”, “Incidence_Rate” and “Case-Fatality_ratio”. We noticed that a lot of the provinces were NaN and the combined key was not following the same format for all of the rows (some contained Unknown, Country where others didn’t have a province). We also noticed there were negative active cases which did not make sense indicating that there may be some outliers.

As for the missing values we found the following for train: “age” - 209265, “sex” - 207084, “province” - 4106, “country” - 18, “latitude” and “longitude” - 2, “date_confirmation” - 288, “additional_information” - 344912 and “source” - 128478. As for the location dataset we found the following missing data: “Province-State” - 176, “Lat” and “Long_” 80, “Active” - 2, “Incidence_Rate” - 80 and Case-Fatality_Ratio” - 48.

Section 1.2

In the ‘age’ column, we parsed the string and changed all ages in range format to a single float number by taking the mean of the range. Ages with characters such as “80+” were stripped of the character and only set as “80”. Ages that were NaN, were set to the mean age of the country that they were in. Any ages that were still NaN were set to the average age of the entire dataset.

The column sex, rows with NaN in the sex column were replaced with unknown. This is done because we believe there was no merit in setting NaN values in sex to be the average sex of the country. After all, this will severely skew the data since there was a large portion of sex with NaN values.

In the province and country column, we found that in training data, there were only 16 countries that were of NaN values and 0 in testing data. In training data, all NaN countries had Taiwan as their province. To have no NaN value, we decided to set the country to China. In the provinces column, NaN provinces were imputed based on matching latitude and longitude of other rows that had no NaN province. Provinces that were still NaN were set to unknown.

There were no missing longitude or longitude values in testing data, and only 2 were missing in training data. These 2 rows were handled as an outlier. Date confirmation had their format standardized, and ranges were set to the middle data of the range. Additional information, source, and outcome were set to unknown because we believe there is no value in inputting false information into these columns. The outcome column had no missing NaN values in either dataset.

Section 1.3

For this part of the assignment, we used a z-score to find the outliers for train data. We separated the data type into numerical data and categorical data. For categorical data, we used frequency to get z-scored and

find the outliers. We used those outliers to analyze and decide on whether or not we wanted to remove the rows or take no action since they were still important.

For ‘age’ we got 91895 outliers, but we decided to take no action for age because it was within a valid age and we figured that we age would be useful for training and predicting an outcome. Furthermore, for ‘latitude’ and ‘longitude’, we got 84142 and 295 outliers respectively, but since latitude and longitude were quite important to determine the location and since they will also be useful for training and testing, we decided to take no action, but we did notice that the train data had two rows where the country, province, latitude, and longitude were missing and we decided to remove those rows since these rows were not of any use.

For categorical data, we used frequency on the unique values of the columns to find outliers. Since the column, ‘additional_information’ and ‘source’ were expected to have outliers and since they were not that important, we decided to not take any actions for the outliers in those columns. The columns ‘outcome’ and ‘sex’ had no outliers. Furthermore, the columns ‘country’ and ‘province’ had 24 and 148 outliers respectively, but since they were an important part of the dataset, we decided to take no action for those outliers. Lastly, ‘date_confirmation’ only had 16 outliers which were negligible and it also did not make sense to exclude those rows.

Section 1.4

For transformation, we first changed the numerical data types to floats and then filled the missing provinces with unknown and corrected the combined keys for each row. Train data had “United States”, location data had “US”, so we changed the location data to match the train data. For each unique combined key’s “Lat” and “Long_” the mean was saved. “Confirmed”, “Deaths” and “Recovered”, the sums were saved. We then dropped the duplicate rows and replaced the values for each unique row with the mean for that “Combined_Key”: latitude and longitude or the sum for that “Combined_Key”: “Confirmed”, “Deaths” and “Recovered”. After that we recalculated the active cases with the formula (“Active” = “Confirmed” - “Deaths” - “Recovered”), the incidence rate with the formula (“Incidence_Rate” = “Confirmed” / 100,000) and “Case-Fatality_Ratio” with the formula (“Case-Fatality_Ratio” (%) = “Confirmed” / “Deaths”). This was our final transformed location dataset.

Section 1.5

Cases and location datasets were joined based on the province and country. This decision was that longitude and latitude values were too unique to be used as the key features. We did not use a range either because, with country borders, it is not guaranteed that a line between any two points of the country will be within the border. We believe that this will invalidate province and country legitimacy so we chose province and country (‘Combined_Key’). The merging method we chose is left join because cases_train contains the labeling and this is how classification models are trained. An inner join was not done because we lost too much case data, which is vital for training. The strategy was to have a new column that contains the province and country that we called “combined key” and joined it under this column. For the missing values after joining the dataset, we decided to replace the values with ‘-1.0’ to indicate a default value. We were not sure if this was the best way to go since we have not decided on a training strategy yet and it would not be fair to remove or perform preprocessing steps on these missing values.

Section 1.6

The unique values in train data outcomes were 'deceased', 'hospitalized', 'nonhospitalized', 'recovered'. The outcomes showed the state of each patient which can be one of the four possibilities. This label relates to prediction tasks because we believe it is to predict what state the patient is in, which is different from classification.