CMPT 459

Milestone 1: Report

Martin Ester

Mohammad Raad Sarar 301326232

Kazi Zubayer Quader Dhrubo 301325924

Ahmed Irteza Haque 301325980

1.1 Exploratory Data Analysis

All the attributes in *cases_train.csv* and *location.csv* were read in by *eda.ipynb* and each attribute was analyzed independently of each other (except for latitude and longitude). Each attribute contains number of missing values. Additionally, for each numerical attribute, count, mean, standard deviation, min, max and the interquartile ranges were printed along with histograms to understand distribution, and box and whisker diagrams to understand outliers. For latitude and longitude values, scatterplots were printed and overlayed over an image of the world map to make sure latitude and longitude values are accurate. For each categorical attribute, histograms were created to understand the distribution of the unique values.
Columns in location.csv (case fatality ratio, recovered, active, incidence rate, deaths) were very skewed and could contain outliers. The active column also contained negative numbers. Rows with negative numbers were removed first and then statistics and visualizations were created.

1.2 Data cleaning and Imputing missing values

We removed only 2 rows from the *cases_train.csv* where latitude and longitude values were not present. Columns *additional_information* and *source* were dropped as we believe the text data would not be of much help during the classification phase as a lot of data was missing. Cleaning and imputing methods are described below:

Age: The values in the form of (*x-y*) were found by python regular expression and the average of x and y was assigned to the age. The values like *z-months* were found by python regular expression and replaced by *z/12*. The values like *a+* or *a-* were also found similarly and replaced by *a*. To **fill in missing values,** a normal distribution was created of the values that were not missing and age was assigned according to the probability determined by the mean and standard deviation of the normal distribution *(np.random.normal(age_mean, age_std, 1))*.

Sex: Before imputing the sex values, we figured out how many male and female values that were not missing, we created a probability of male and female according to the counts and imputed the values according to that probability.

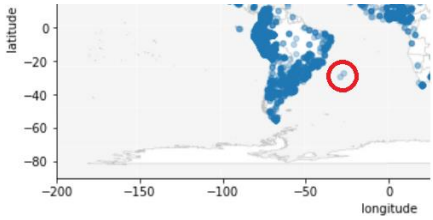
Country: We found that only a few rows (18) with attribute country was missing. So, we printed them all. They all had province as “Taiwan”. So, we assigned “Taiwan” as the country (not sure if Taiwan is the country or China, Google was a little unclear) of the rows that had country missing but had Taiwan as a province.

Province: For rows that did not have a province, we checked the country (as after the previous step all rows have a country value), and found the province with the highest occurrence (*mode*) of that country and assigned it as province. Some countries did not have any provinces listed, in that case finding the mode was not possible. So instead of leaving the province field empty or removing the entire row, we decided to replace the Null value with *“unknown”*. eg. (province: unknown, country: Togo).

Date Confirmation: We used a similar approach to the *province attribute* as we assumed countries have release their date of confirmation in a large batch. So, for missing date confirmation, we found the country first and then found the mode of the date confirmed of that country and imputed with that value.

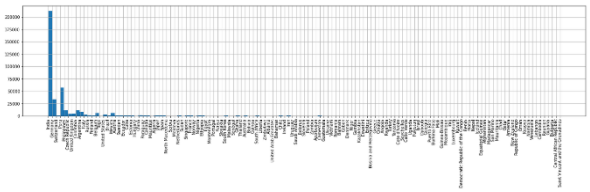
1.3 Dealing with outliers

Latitude & longitude: The scatterplot that we generated of the latitude and longitude showed some points located in the sea (image to the right). We filtered those points (using latitude and longitude thresholds) to see what country and provinces they belonged to. We found that the points belonged to Rio Grande do Sul and Santa Catarina, Brazil. So, we found all the data entries with to province = Rio Grande do Sul and

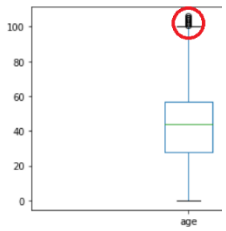


country = Brazil and took the average latitude and longitude values and replaced them with the average values.

Provinces & Countries:



The data for the countries and provinces and countries were vastly skewed with India having over 200,000 records (over 50% of the dataset). Some countries only had one data entry (row). We could have considered them to be outliers, but we decided not to. This is because we realize that even if we have one row for a country, it significantly increases the chance of classifying a row in the test file if that country shows up.



Age: In our box and whisker plot we saw some age values over 100 marked as outliers (plot on the right). But we decided not to change them as it would give us a more diverse dataset while training the model. It is also feasible to believe that people do live over 100 years and this is an age group, whose data cannot be not easily available. So, we decided to keep this group of data without changing the values.

Location.csv: We noticed a lot of attributes with outliers in location.csv. Case fatality ratio, Recovered, Active, Incidence rate, and Deaths were all really skewed where a few locations had most of the data while the other locations had barely any data. The active column also contained negative numbers. Rows with negative numbers were removed first and then statistics and visualizations were created But since we were not asked to deal with outliers in locations.csv, we did not change anything yet, but kept a mental note that it might need to be changed later on.

1.4 Transformation

To transform the location dataset, a new dataframe, US, is created where Country_Region == US. The US dataset is then grouped by Province_State and each column was aggregated in the following way:

Latitude & longitude, Incidence_Rate: mean of aggregation

Confirmed, Deaths, Recovered, Active: sum of aggregation

Case-Fatality_Ratio: aggregated Deaths/aggregated Confirmed*100

All the data with Country_Region == US is removed from the original location dataset and the newly created dataset, US, is appended to the original dataset and then sorted by Country_Region to get a transformed location dataset.

1.5 Joining the cases and location dataset

Initially we join cases and location using composite keys (province, country) for each dataset. We got a dataset which had a lot of NaNs. This is because the cases dataset was complete but the location dataset had a lot missing provinces. We separated the rows with nan values and calculated the minimum distance using the latitude and longitude of both tables. We used the minimum distance data from the locations dataset to impute into the missing cases dataset

1.6 Outcome labels

The different outcomes and their meanings are:

Recovered: Contracted covid-19 but has tested negative afterwards.

Hospitalized: Contracted covid-19 and has been admitted to a hospital but has neither died nor recovered yet.

Nonhospitalized: Contracted covid-19 but did not require medical attention. Neither recovered or deceased.

Deceased: Individual passed away (may they rest in peace) due to covid-19.

The data mining task that predicts outcome labels is **classification**