COMP 4331 – Group Project Report

Task 1

Before doing part (a), we need to first delete the duplicated rows from the countries_data. After deletion, duplicate rows are found to be 3757 and the data size reduces to 107 rows.

a. Data Preprocessing

Missing data
Before filling in the null data, we first observe the number of null data in each column. Only three of the columns contains null data.

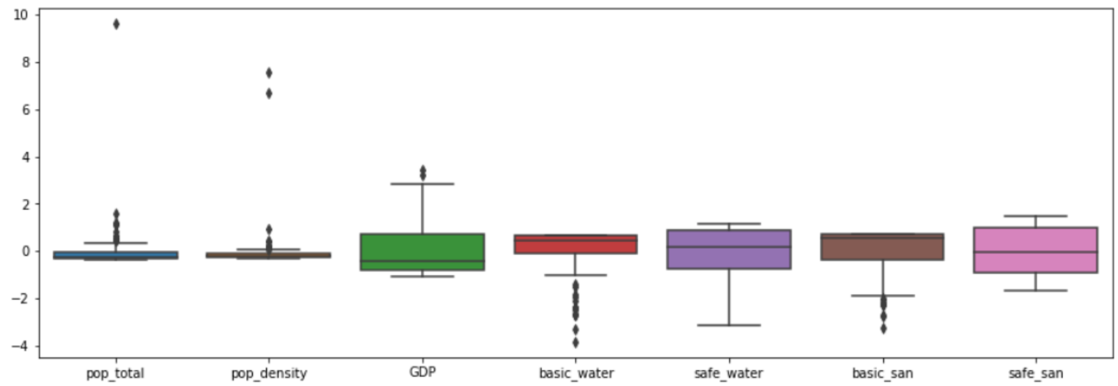
| attribute | Number of null data |
|---|---|
| pop_density | 1 |
| safe_water | 36 |
| safe_san | 38 |

Then we fit the entire countries data (except first two string columns – country names and country codes) into the imputer using IteractiveImputer() and transform the dataset to fill in the missing value.

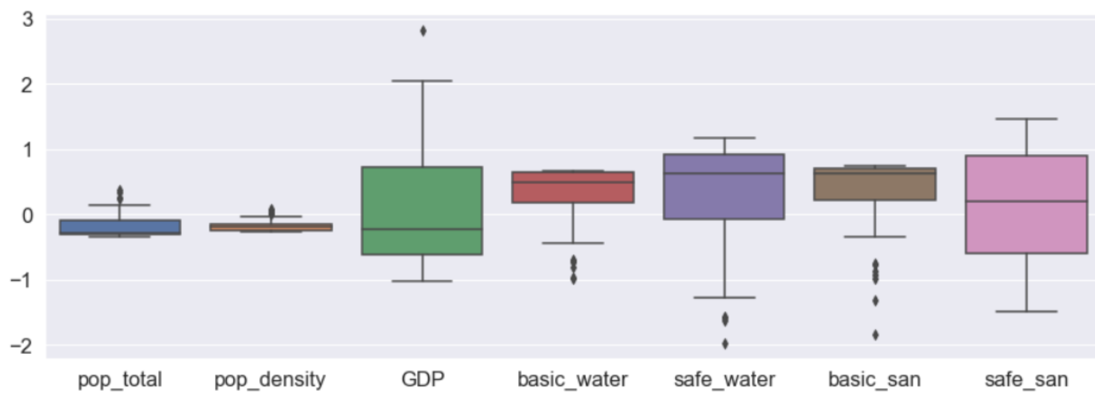
Standardization
After that fit this dataset (except columns with string data) to StandardScaler() function to standardize the numeric data. A summary table for all numeric columns is output to ensure the precision of standardization as mean and std value for each attribute are 0 and 1 respectively.

| | pop_total | pop_density | GDP | basic_water | safe_water | basic_san | safe_san |
|---|---|---|---|---|---|---|---|
| **mean** | -1.971424e-17 | -2.075183e-18 | -1.411125e-16 | -5.686002e-16 | 1.867665e-16 | -4.150366e-16 | -2.593979e-16 |
| **std** | 1.004706e+00 | 1.004706e+00 | 1.004706e+00 | 1.004706e+00 | 1.004706e+00 | 1.004706e+00 | 1.004706e+00 |

Outliers Deletion
Records with data value larger or less than 1.5*boundary are considered as outliers. We calculate the boundary as [lower quantile – 1.5 * inter-quartile range, higher quantile + 1.5 * inter-quartile range]. Some people may choose the coefficient to be 1.75 or 1, but 1.5 should be the best value to indicate the outlier values mathematically.
There are 32 records detected as outliers, so the dataset is further reduced to 75 rows, meaning 75 countries left.
For additional visualization, here are the boxplots showing the records before and after deletion of outliers:
(Before)

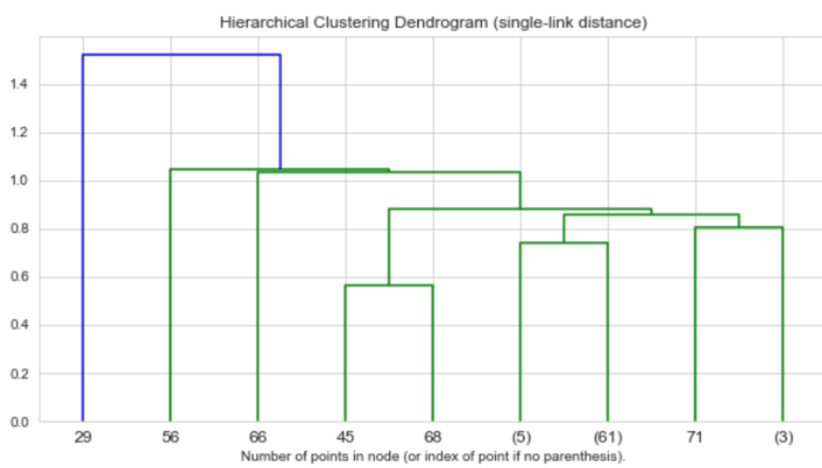


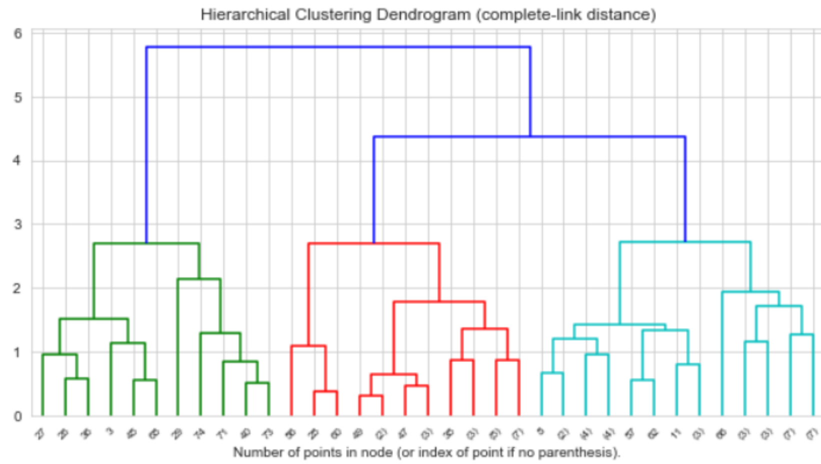

(After)

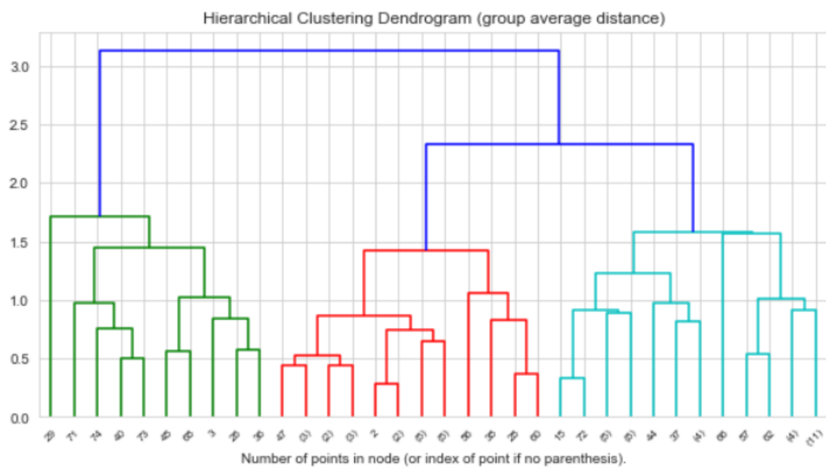



b. Hierarchical Clustering

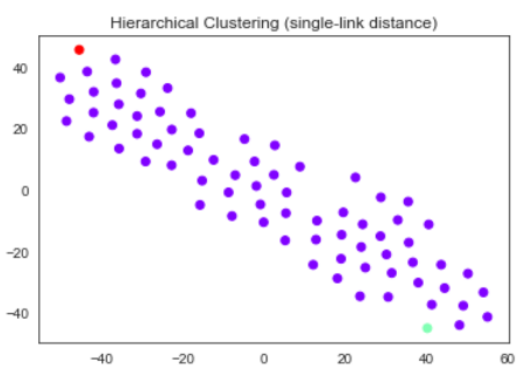
AgglomerativeCluster() function is applied for hierarchical clustering and plot_dendrogram() provided in the tutorial notes are used for dendrogram visualization. The three measurements of Euclidean distance are the inputs of the argument "linkage" in the AgglomerativeCluster() function. The dendrogram plotted with three different distance measurement are shown below:

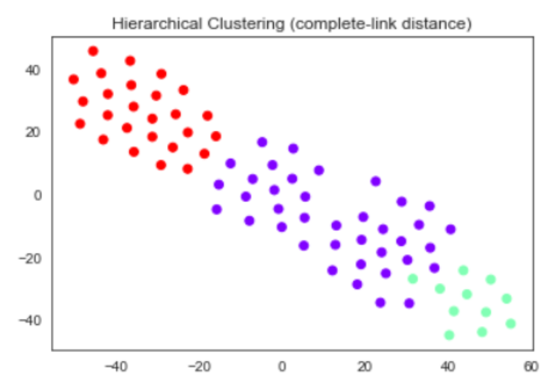


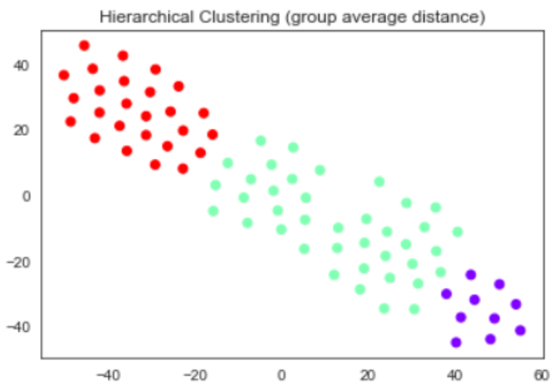



Then, dataset is embedded by the function of TSNE() and reduced to 2-dimension dataset. AgglomerativeClustering() is used again to undergo clustering analysis and the result is visualized as followed:






c. Clustering Validity Measure

Davies-Bouldin and Silhouette score are computed as following table:

| Euclidean Distance | Davies-Bouldin Score | Silhouette Score |
|---|---|---|
| Single-link | 0.5137742665128471 | 0.10201944334301297 |
| Complete-link | 0.819269481106398 | 0.41386704795264717 |
| Group average | 0.7733178860340705 | 0.42968049502721345 |

Davies-Bouldin score measures the average similarity of each cluster with its most similar cluster, so the lower value indicates a better clustering analysis. Whilst, Silhouette Score measures the degree of overlapping between clusters. It ranges between -1 to 1. Higher and positive value indicates a sample is assigned to the right cluster.
Comparing the performances by these three measurements, single-link distance would be first considered as the worst since its silhouette score is a lot lower than the other two, provided that they have similar value of Davies-Bouldin score. Eventually, group average distance would be the best given the fact that it has both lower Davies-Bouldin score and higher silhouette score than complete-link distance.

d. Visualization
Using the clustering result in part (b), here are the names of the countries and summary statistics for each cluster group (avoiding the messiness of the report, boxplot for each attribute can be seen in the jupyter notebook):

Countries in cluster 0: ['Bolivia', 'Guatemala', "Lao People's Democratic Republic", 'Peru', 'Cambodia', 'Myanmar', 'Nicaragua', 'Nepal', 'Senegal', 'Mali']

| | pop_total | pop_density | GDP | basic_water | safe_water | basic_san | safe_san |
|---|---|---|---|---|---|---|---|
| **count** | 1.000000e+01 | 10.000000 | 10.000000 | 10.000000 | 10.000000 | 10.000000 | 10.000000 |
| **mean** | 2.094376e+07 | 74.055834 | 6468.767884 | 84.978949 | 38.923178 | 62.543983 | 32.059497 |
| **std** | 1.428360e+07 | 61.137676 | 3379.900334 | 6.105321 | 14.191634 | 11.095004 | 12.594602 |
| **min** | 6.545502e+06 | 10.480146 | 2423.828765 | 78.260830 | 16.081866 | 39.335420 | 18.709404 |
| **25%** | 1.270892e+07 | 26.392651 | 3811.473443 | 80.889581 | 27.040556 | 59.599459 | 22.184779 |
| **50%** | 1.654528e+07 | 67.982832 | 5493.235881 | 81.917973 | 41.579167 | 63.193140 | 29.038677 |
| **75%** | 2.637104e+07 | 89.629179 | 8784.341511 | 90.548966 | 51.288046 | 72.023001 | 38.969355 |
| **max** | 5.404542e+07 | 195.939107 | 13380.364420 | 94.190581 | 55.990782 | 74.459410 | 58.053590 |

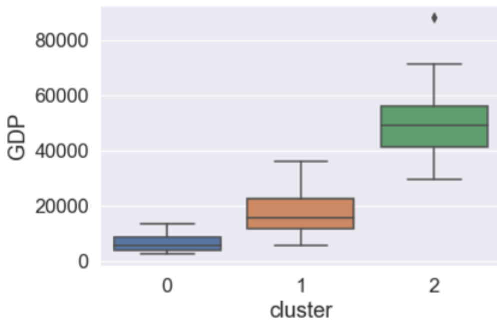
Countries in cluster 1: ['Kyrgyzstan', 'Ukraine', 'Libyan Arab Jamahiriya', 'Dominican Republic', 'Egypt', 'Oman', 'Costa Rica', 'Colombia', 'Uzbekistan', 'Morocco', 'South Africa', 'El Salvador', 'Vietnam', 'Azerbaijan', 'Algeria', 'Moldova, Republic of', 'Paraguay', 'Thailand', 'Bosnia and Herzegovina', 'Ecuador', 'Jordan', 'Venezuela', 'Sri Lanka', 'Serbia', 'Albania', 'Tunisia', 'Croatia', 'Turkey', 'Honduras', 'Panama', 'Uruguay', 'Kazakhstan', 'Romania', 'Bulgaria', 'Chile', 'Puerto Rico', 'Argentina', 'Belarus', 'Iraq']

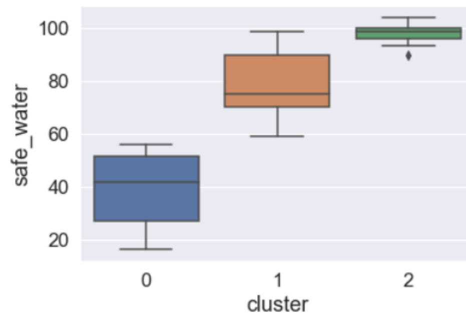
| | pop_total | pop_density | GDP | basic_water | safe_water | basic_san | safe_san |
|---|---|---|---|---|---|---|---|
| **count** | 3.900000e+01 | 39.000000 | 39.000000 | 39.000000 | 39.000000 | 39.000000 | 39.000000 |
| **mean** | 2.464838e+07 | 95.322751 | 17489.428637 | 95.572848 | 78.475592 | 92.347198 | 52.087110 |
| **std** | 2.666615e+07 | 91.084068 | 8003.124202 | 3.982459 | 11.437728 | 6.657334 | 17.906308 |
| **min** | 2.657637e+06 | 3.795632 | 5470.811536 | 85.522116 | 58.833327 | 75.747098 | 16.986489 |
| **25%** | 6.455226e+06 | 38.857146 | 11833.434120 | 93.673354 | 70.144795 | 87.787643 | 41.539256 |
| **50%** | 1.073896e+07 | 77.029671 | 15643.731450 | 96.483971 | 75.068012 | 94.258505 | 52.535786 |
| **75%** | 3.789078e+07 | 101.742866 | 22700.898870 | 99.007799 | 89.735831 | 97.453477 | 64.635585 |
| **max** | 1.003881e+08 | 360.017362 | 35948.191960 | 100.000000 | 98.639170 | 100.000001 | 80.554925 |

Countries in cluster 2: ['Italy', 'Canada', 'Austria', 'Czech Republic', 'Malaysia', 'New Zealand', 'United Kingdom', 'Slovakia', 'Kuwait', 'Poland', 'Switzerland', 'Greece', 'Finland', 'Portugal', 'Sweden', 'Norway', 'Germany', 'Hungary', 'Slovenia', 'Saudi Arabia', 'Australia', 'Ireland', 'France', 'United Arab Emirates', 'Spain', 'Denmark']

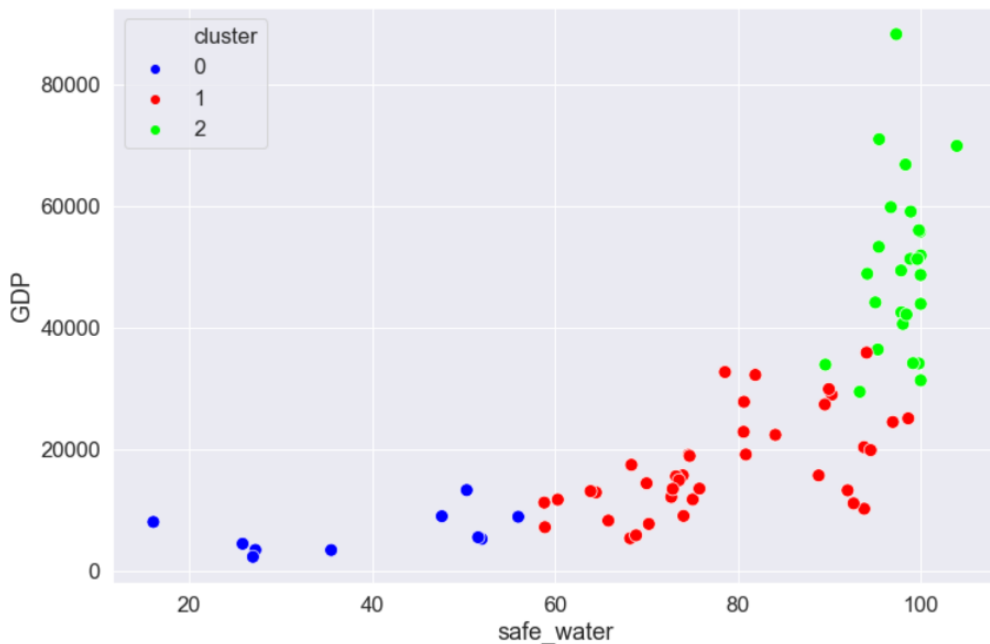
| | pop_total | pop_density | GDP | basic_water | safe_water | basic_san | safe_san |
|---|---|---|---|---|---|---|---|
| **count** | 2.600000e+01 | 26.000000 | 26.000000 | 26.000000 | 26.000000 | 26.000000 | 26.000000 |
| **mean** | 2.341429e+07 | 108.033904 | 49811.061718 | 99.605637 | 97.805082 | 98.929310 | 90.460826 |
| **std** | 2.378461e+07 | 77.052477 | 13900.266243 | 0.859682 | 2.866048 | 1.691049 | 7.606282 |
| **min** | 2.087946e+06 | 3.247871 | 29525.577360 | 96.695939 | 89.572762 | 91.245181 | 75.639872 |
| **25%** | 5.594874e+06 | 36.399485 | 41045.950625 | 99.740445 | 95.775224 | 98.778477 | 83.457854 |
| **50%** | 1.027744e+07 | 107.554727 | 49171.831160 | 99.999998 | 98.392232 | 99.255525 | 93.315917 |
| **75%** | 3.675908e+07 | 137.145556 | 55992.944715 | 100.000000 | 99.796934 | 99.820011 | 96.535138 |
| **max** | 8.313280e+07 | 274.708982 | 88240.901030 | 100.000005 | 103.952502 | 100.000000 | 100.000000 |

To observe the similarity of each attributes between the groups, we can look at attributes contain distant mean values between groups. “GDP”, “safe_water”, and “safe_san” are found to be the best attributes to illustrate differences between the groups. As the instruction only require specifying two attributes only, we can further observe their boxplots. “GDP” and “safe_water” would be the best as they have least overlapping areas.




Using these two attributes, a scatter plot showing different clusters is shown below:




Task 2

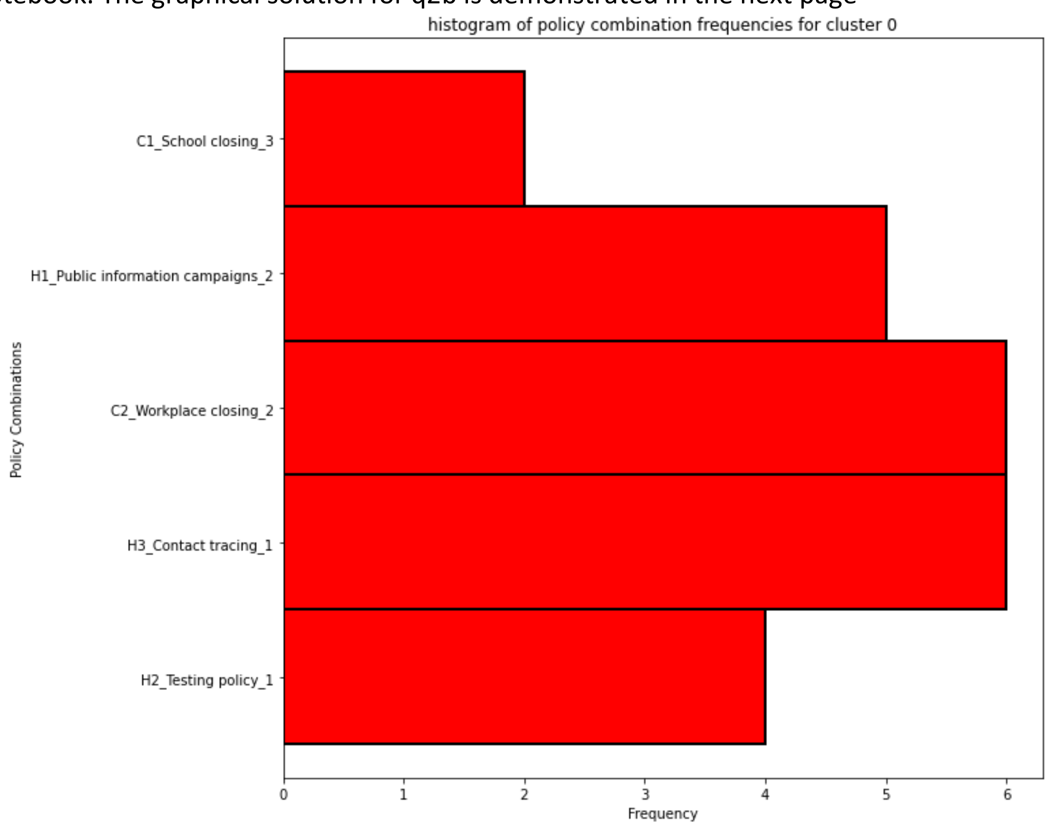
(a) We used apriori algorithm to find the target data.

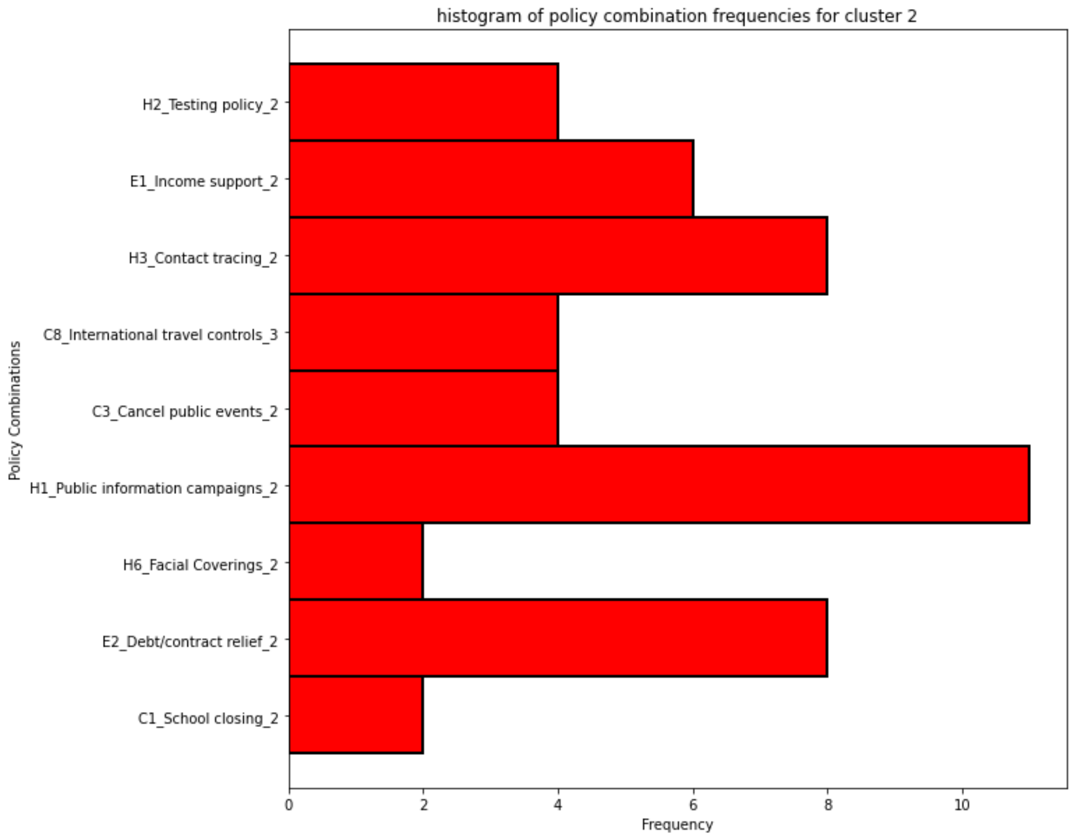
For data preprocessing, we first extracted all policy data columns along with new_cases_percentages column. For new_cases_percentages column, we renamed it to ‘class_1_or_0’ with value 1 if the original data is 0/1 and value 0 if the original data is 2/3. We also added a column to identify the cluster of the record. The data of cluster column is extracted from results in q1d. Now, the data columns should look like [policy colmns, class_1_or_0, cluster] with value 1(0) indicating true(false) in [policy colmns, class_1_or_0] and value 0,1,2 in [cluster] indicating the cluster of a record.

Then, we performed apriori algorithm for each cluster. We did that with a for loop. For each cluster, we filtered the data with corresponding value in the cluster column. Then we dropped the cluster column (as now all records in the dataframe belong to one cluster) and the class_0_or_1 column (we only want frequent policy combinations for now). Now the dataframe should only contain values 0 and 1 which is favorable for apriori function. We then used the apriori function from mlxtend with minimum support of 0.3 to extract policy combinations that exist in at least 30% of the data for each cluster.

For each frequent policy combinations, we filtered the original data for each cluster (data with policy combinations and class_0_or_1) with the policy combinations found above (keep the data with value 1 in the frequent policy columns). Now the data should represent records with a frequent policy combination along with class_0_or_1. We then just simply counted the number of records with class_0_or_1 = 1 in that dataset and divide it by the number of the corresponding frequent policy combination dataset. This should show what portion of the data in that policy combination has number of new cases <= 0.00221%. We will keep the policy combinations with portion >= 0.6.

(b) To plot the histogram we will count use Counter to count the frequency for each policy in the effective policy combinations found above.
The result shows cluster 1 has no policy combinations found. For cluster 0 we found 10 combinations and for cluster 2 we found 22 combinations. Details are shown in the python notebook. The graphical solution for q2b is demonstrated in the next page






Task 3

**Data Preprocessing**

First, create a DataFrame ("Symptoms") for all the symptoms. Then to output "0" if the record's original value is less than the attribute's median value, and set it to 1 otherwise, we use "qcut" to output this solution.

| | pct_fever_weighted | pct_cough_weighted | pct_difficulty_breathing_weighted | pct_fatigue_weighted | pct_stuffy_runny_nose_weighted | pct_aches_muscle_ |
|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 | 1 | |
| **1** | 0 | 0 | 0 | 0 | 0 | |
| **2** | 0 | 0 | 0 | 0 | 1 | |
| **3** | 1 | 1 | 1 | 1 | 1 | |
| **4** | 0 | 1 | 1 | 1 | 0 | |
| **...** | ... | ... | ... | ... | ... | |
| **3859** | 1 | 1 | 1 | 1 | 1 | |
| **3860** | 0 | 0 | 0 | 0 | 0 | |
| **3861** | 0 | 0 | 0 | 0 | 0 | |
| **3862** | 0 | 0 | 0 | 0 | 0 | |
| **3863** | 1 | 0 | 1 | 1 | 0 | |

3864 rows × 13 columns

Second, create another DataFrame ("Case") for defining the total number of cases in percentage of population as: total cases/pop total $\times$ 100.

Then, I also use "qcut" to output high ("1") if it is larger than the corresponding median, and set it to low ("0") otherwise.

| | pop_total | total_cases | case |
|---|---|---|---|
| **0** | 6456900 | 3151 | 0.048801 |
| **1** | 60297396 | 239706 | 0.397540 |
| **2** | 37589262 | 131495 | 0.349821 |
| **3** | 11513100 | 45565 | 0.395767 |
| **4** | 8877067 | 23875 | 0.268951 |
| **...** | ... | ... | ... |
| **3859** | 144373535 | 975576 | 0.675730 |
| **3860** | 69625582 | 3216 | 0.004619 |
| **3861** | 126264931 | 27029 | 0.021407 |
| **3862** | 50339443 | 26688 | 0.053016 |
| **3863** | 36471769 | 65453 | 0.179462 |

3864 rows × 3 columns

| | case |
|---|---|
| **0** | 0 |
| **1** | 1 |
| **2** | 1 |
| **3** | 1 |
| **4** | 1 |
| **...** | ... |
| **3859** | 1 |
| **3860** | 0 |
| **3861** | 0 |
| **3862** | 0 |
| **3863** | 1 |

Third, combine DataFrame ("Symptoms") and DataFrame("Case") into one DataFrame("Final")

| | pct_fever_weighted | pct_cough_weighted | pct_difficulty_breathing_weighted | pct_fatigue_weighted | pct_stuffy_runny_nose_weighted | pct_aches_muscle_ |
|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 | 1 | |
| **1** | 0 | 0 | 0 | 0 | 0 | |
| **2** | 0 | 0 | 0 | 0 | 1 | |
| **3** | 1 | 1 | 1 | 1 | 1 | |
| **4** | 0 | 1 | 1 | 1 | 0 | |
| **...** | ... | ... | ... | ... | ... | |
| **3859** | 1 | 1 | 1 | 1 | 1 | |
| **3860** | 0 | 0 | 0 | 0 | 0 | |
| **3861** | 0 | 0 | 0 | 0 | 0 | |
| **3862** | 0 | 0 | 0 | 0 | 0 | |
| **3863** | 1 | 0 | 1 | 1 | 0 | |

3864 rows × 14 columns

a)

I use apriori function with "min_support = 0.2" to extract all symptom combinations that appear in at least 20% of all the records from the DataFrame("Symptoms")

| | support | itemsets |
|---|---|---|
| **0** | 0.499224 | (pct_fever_weighted) |
| **1** | 0.5 | (pct_cough_weighted) |
| **2** | 0.499224 | (pct_difficulty_breathing_weighted) |
| **3** | 0.499482 | (pct_fatigue_weighted) |
| **4** | 0.499741 | (pct_stuffy_runny_nose_weighted) |
| **...** | ... | ... |
| **2300** | 0.216615 | (pct_headache_weighted, pct_anosmia_ageusia_we... |
| **2301** | 0.200052 | (pct_headache_weighted, pct_anosmia_ageusia_we... |
| **2302** | 0.202381 | (pct_difficulty_breathing_weighted, pct_anosmi... |
| **2303** | 0.204193 | (pct_headache_weighted, pct_difficulty_breathi... |
| **2304** | 0.205745 | (pct_headache_weighted, pct_anosmia_ageusia_we... |

2305 rows × 2 columns

b)

To find those that appear in at least 60% of all records and have high total number of cases in percentage of population among all the symptom combinations obtained in part (a) above, I use apriori function again with “min_support = 0.2” to extract all symptom combinations including the attribute “case” that appear in at least 20% of all the records from the DataFrame(“Final”). Then, use “association rule” to find confidence with “min_threshold = 0.6” and “consequents = case” to output the symptom combinations in antecedents.

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | convict |
|---|---|---|---|---|---|---|---|---|---|
| 60 | (pct_difficulty_breathing_weighted) | (case) | 0.499224 | 0.5 | 0.302536 | 0.606013 | 1.212027 | 0.052924 | 1.269 |
| 75 | (pct_stuffy_runny_nose_weighted) | (case) | 0.499741 | 0.5 | 0.327899 | 0.656137 | 1.312273 | 0.078028 | 1.454 |
| 435 | (pct_difficulty_breathing_weighted, pct_cough_... | (case) | 0.361801 | 0.5 | 0.240683 | 0.665236 | 1.330472 | 0.059783 | 1.493 |
| 469 | (pct_cough_weighted, pct_stuffy_runny_nose_wei... | (case) | 0.386646 | 0.5 | 0.263975 | 0.682731 | 1.365462 | 0.070652 | 1.575 |
| 496 | (pct_cough_weighted, pct_aches_muscle_pain_wei... | (case) | 0.361284 | 0.5 | 0.219720 | 0.608166 | 1.216332 | 0.039079 | 1.276 |
| 517 | (pct_cough_weighted, pct_sore_throat_weighted) | (case) | 0.360507 | 0.5 | 0.226190 | 0.627423 | 1.254846 | 0.045937 | 1.342 |
| 626 | (pct_difficulty_breathing_weighted, pct_stuffy... | (case) | 0.355331 | 0.5 | 0.247930 | 0.697742 | 1.395484 | 0.070264 | 1.654 |
| 650 | (pct_difficulty_breathing_weighted, pct_aches_... | (case) | 0.352226 | 0.5 | 0.223085 | 0.633358 | 1.266716 | 0.046972 | 1.363 |
| 682 | (pct_difficulty_breathing_weighted, pct_sore_t... | (case) | 0.359990 | 0.5 | 0.220756 | 0.613228 | 1.226456 | 0.040761 | 1.292 |
| 719 | (pct_difficulty_breathing_weighted, pct_nausea... | (case) | 0.352743 | 0.5 | 0.216097 | 0.612619 | 1.225238 | 0.039726 | 1.290 |
| 745 | (pct_difficulty_breathing_weighted, pct_chills... | (case) | 0.332298 | 0.5 | 0.200569 | 0.603583 | 1.207165 | 0.034420 | 1.261 |
| 818 | (pct_stuffy_runny_nose_weighted, pct_aches_mus... | (case) | 0.336698 | 0.5 | 0.231625 | 0.687932 | 1.375865 | 0.063276 | 1.602 |
| 839 | (pct_sore_throat_weighted, pct_stuffy_runny_no... | (case) | 0.336439 | 0.5 | 0.221532 | 0.658462 | 1.316923 | 0.053313 | 1.463 |
| 868 | (pct_nausea_weighted, pct_stuffy_runny_nose_we... | (case) | 0.323758 | 0.5 | 0.213251 | 0.658673 | 1.317346 | 0.051372 | 1.464 |
| 2922 | (pct_difficulty_breathing_weighted, pct_cough_... | (case) | 0.307195 | 0.5 | 0.220238 | 0.716933 | 1.433867 | 0.066641 | 1.766 |
| 3271 | (pct_cough_weighted, pct_stuffy_runny_nose_wei... | (case) | 0.289337 | 0.5 | 0.201605 | 0.696780 | 1.393560 | 0.056936 | 1.648 |
| 3336 | (pct_cough_weighted, pct_sore_throat_weighted,... | (case) | 0.297101 | 0.5 | 0.205745 | 0.692509 | 1.385017 | 0.057195 | 1.626 |

Extraction of the rules below:

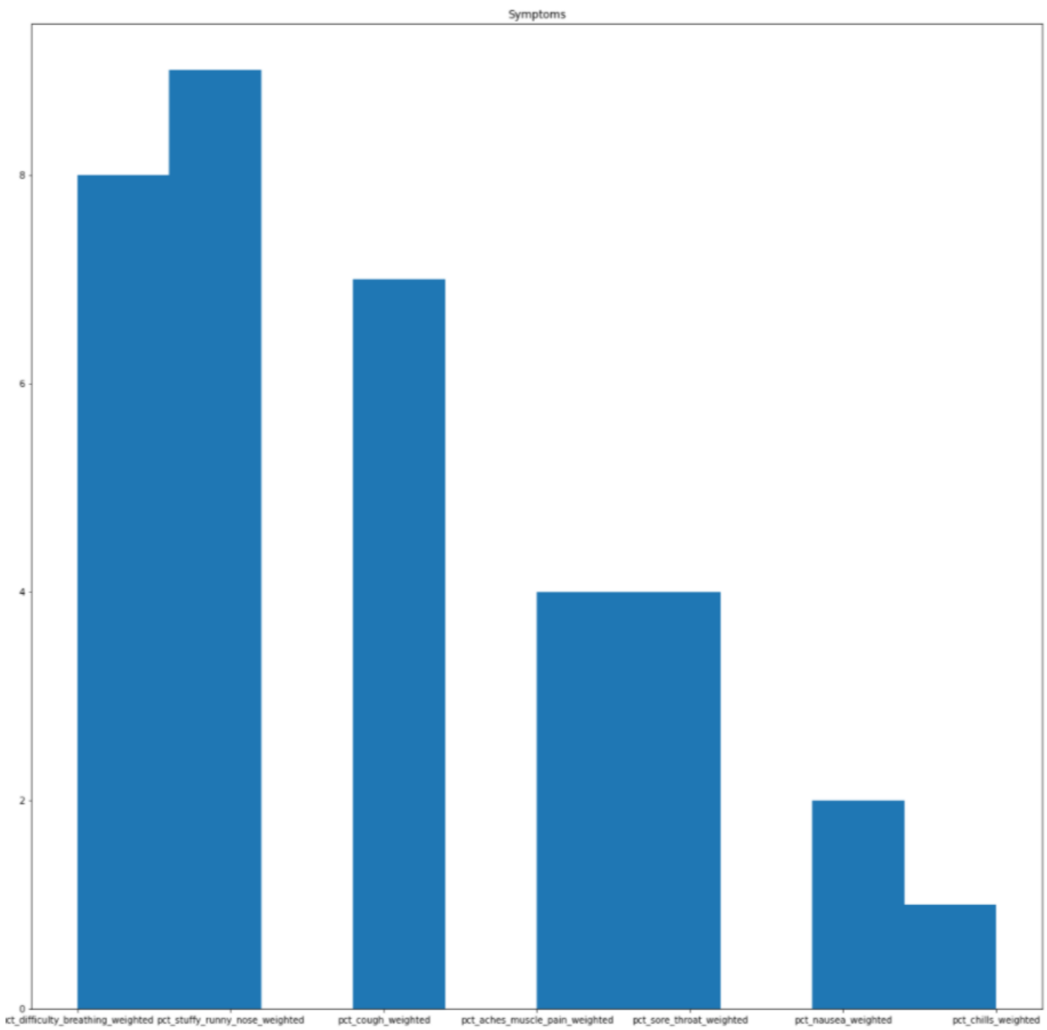
```
60                   (pct_difficulty_breathing_weighted)
75                      (pct_stuffy_runny_nose_weighted)
435     (pct_difficulty_breathing_weighted, pct_cough_...
469     (pct_cough_weighted, pct_stuffy_runny_nose_wei...
496     (pct_cough_weighted, pct_aches_muscle_pain_wei...
517        (pct_cough_weighted, pct_sore_throat_weighted)
626     (pct_difficulty_breathing_weighted, pct_stuffy...
650     (pct_difficulty_breathing_weighted, pct_aches_...
682     (pct_difficulty_breathing_weighted, pct_sore_t...
719     (pct_difficulty_breathing_weighted, pct_nausea...
745     (pct_difficulty_breathing_weighted, pct_chills...
818     (pct_stuffy_runny_nose_weighted, pct_aches_mus...
839     (pct_sore_throat_weighted, pct_stuffy_runny_no...
868     (pct_nausea_weighted, pct_stuffy_runny_nose_we...
2922    (pct_difficulty_breathing_weighted, pct_cough_...
3271    (pct_cough_weighted, pct_stuffy_runny_nose_wei...
3336    (pct_cough_weighted, pct_sore_throat_weighted,...
Name: antecedents, dtype: object
```

c)

First, list out all the symptoms in part 3b) symptom combinations as a list.

```
['pct_difficulty_breathing_weighted', 'pct_stuffy_runny_nose_weighted', 'pct_difficulty_breathing_weighted', 'pct_cough_weig
d', 'pct_cough_weighted', 'pct_stuffy_runny_nose_weighted', 'pct_cough_weighted', 'pct_aches_muscle_pain_weighted', 'pct_cou
weighted', 'pct_sore_throat_weighted', 'pct_difficulty_breathing_weighted', 'pct_stuffy_runny_nose_weighted', 'pct_difficult
reathing_weighted', 'pct_aches_muscle_pain_weighted', 'pct_difficulty_breathing_weighted', 'pct_sore_throat_weighted', 'pct_
ficulty_breathing_weighted', 'pct_nausea_weighted', 'pct_difficulty_breathing_weighted', 'pct_chills_weighted', 'pct_stuffy_
ny_nose_weighted', 'pct_aches_muscle_pain_weighted', 'pct_sore_throat_weighted', 'pct_stuffy_runny_nose_weighted', 'pct_naus
weighted', 'pct_stuffy_runny_nose_weighted', 'pct_difficulty_breathing_weighted', 'pct_cough_weighted', 'pct_stuffy_runny_no
weighted', 'pct_cough_weighted', 'pct_stuffy_runny_nose_weighted', 'pct_aches_muscle_pain_weighted', 'pct_cough_weighted', '
_sore_throat_weighted', 'pct_stuffy_runny_nose_weighted']
```

Then, plot the histogram for these 7 symptoms.




Task 4

(a) For data preprocessing we first filtered the mobility data, pop_density and new_cases_percentages columns out. Then we added a new column log_density with value log(pop_density). Then we plotted the scatter plot of each mobility data column against log_density. Different colors were used for different new_cases_percentages class. Details of the graph are shown in the python notebook.

(b) To divide mobility data into five equal sizes, we used quantile function for pop_density. we divided the data according to the 0-20%,20-40%,40-60%,60-80% and 80-100% quantiles. Then we calculated a correlation matrix using corr() function for each of the five portions of the data and sorted the result to find the mobility data columns with highest correlation to pop_denisty. Details of the answer are shown in the python notebook.

Task 5

1st Additional Analysis

For the 3 clusters country-group we found in task 1d) (denote as "Cluster0", "Cluster1" and "Cluster2")

Cluster 0 country list:

```
Cluster 0 has a data size of  (10, 10)
Countries in cluster 0 :
['Bolivia', 'Guatemala', "Lao People's Democratic Republic", 'Peru', 'Cambodia', 'Myanmar', 'Nicaragua', 'Nepal', 'Senegal',
ali']
```

Extract their values of the 7-symptoms found in task 3c)

| | pct_difficulty_breathing_weighted | pct_stuffy_runny_nose_weighted | pct_cough_weighted | pct_aches_muscle_pain_weighted | pct_sore_throat_weighted | pc |
|---|---|---|---|---|---|---|
| **3** | 1 | 1 | 1 | 1 | 1 | |
| **40** | 0 | 0 | 0 | 0 | 1 | |
| **48** | 0 | 0 | 0 | 1 | 0 | |
| **60** | 1 | 1 | 1 | 1 | 1 | |
| **69** | 1 | 0 | 1 | 0 | 1 | |
| **75** | 0 | 0 | 0 | 0 | 0 | |
| **255** | 1 | 1 | 1 | 1 | 1 | |
| **311** | 0 | 0 | 0 | 1 | 0 | |
| **683** | 0 | 0 | 1 | 1 | 0 | |
| **767** | 0 | 0 | 0 | 0 | 0 | |

Compute their frequency for each 7-symptoms

```
pct_difficulty_breathing_weighted     4
pct_stuffy_runny_nose_weighted        3
pct_cough_weighted                    5
pct_aches_muscle_pain_weighted        6
pct_sore_throat_weighted              5
pct_nausea_weighted                   4
pct_chills_weighted                   6
Name: 1, dtype: int64
```

Average symptoms per country in Cluster 0

```
3.3
```

Cluster 1 country list:

```
Cluster 1 has a data size of  (39, 10)
Countries in cluster 1 :
['Kyrgyzstan', 'Ukraine', 'Libyan Arab Jamahiriya', 'Dominican Republic', 'Egypt', 'Oman', 'Costa Rica', 'Colombia', 'Uzbeki
n', 'Morocco', 'South Africa', 'El Salvador', 'Vietnam', 'Azerbaijan', 'Algeria', 'Moldova, Republic of', 'Paraguay', 'Thail
d', 'Bosnia and Herzegovina', 'Ecuador', 'Jordan', 'Venezuela', 'Sri Lanka', 'Serbia', 'Albania', 'Tunisia', 'Croatia', 'Tur
y', 'Honduras', 'Panama', 'Uruguay', 'Kazakhstan', 'Romania', 'Bulgaria', 'Chile', 'Puerto Rico', 'Argentina', 'Belarus', 'I
q']
```

Extract their values of the 7-symptoms found in task 3c)

| | pct_difficulty_breathing_weighted | pct_stuffy_runny_nose_weighted | pct_cough_weighted | pct_aches_muscle_pain_weighted | pct_sore_throat_weighted | pct_na |
|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 0 | 1 | 1 | |
| **8** | 0 | 1 | 1 | 0 | 0 | |
| **9** | 1 | 0 | 0 | 1 | 1 | |
| **10** | 1 | 1 | 1 | 1 | 1 | |
| **11** | 1 | 1 | 1 | 1 | 1 | |
| **12** | 0 | 0 | 0 | 1 | 1 | |
| **13** | 1 | 1 | 1 | 1 | 1 | |
| **14** | 0 | 0 | 0 | 0 | 1 | |
| **16** | 0 | 0 | 0 | 0 | 0 | |
| **17** | 1 | 0 | 0 | 0 | 0 | |
| **21** | 0 | 1 | 1 | 1 | 1 | |
| **28** | 0 | 0 | 0 | 0 | 1 | |
| **30** | 0 | 0 | 0 | 0 | 0 | |
| **31** | 1 | 0 | 0 | 0 | 0 | |
| **32** | 1 | 0 | 1 | 1 | 1 | |
| **41** | 0 | 0 | 0 | 0 | 0 | |
| **56** | 1 | 1 | 1 | 1 | 1 | |
| **59** | 0 | 0 | 0 | 0 | 0 | |
| **62** | 1 | 0 | 1 | 0 | 1 | |
| **64** | 1 | 1 | 1 | 1 | 1 | |
| **70** | 1 | 0 | 0 | 1 | 1 | |
| **72** | 1 | 1 | 1 | 0 | 1 | |
| **73** | 0 | 0 | 0 | 0 | 0 | |
| **74** | 1 | 1 | 0 | 1 | 0 | |
| **79** | 1 | 0 | 0 | 0 | 1 | |
| **101** | 1 | 0 | 1 | 1 | 1 | |
| **104** | 1 | 1 | 1 | 0 | 0 | |
| **105** | 1 | 0 | 0 | 1 | 0 | |
| **113** | 0 | 0 | 0 | 0 | 1 | |
| **128** | 0 | 1 | 0 | 0 | 1 | |
| **147** | 1 | 1 | 1 | 0 | 1 | |
| **175** | 0 | 0 | 0 | 0 | 0 | |
| **182** | 0 | 0 | 0 | 0 | 0 | |
| **223** | 0 | 0 | 0 | 0 | 0 | |
| **226** | 1 | 1 | 1 | 1 | 1 | |
| **231** | 0 | 1 | 0 | 1 | 1 | |
| **247** | 0 | 1 | 0 | 0 | 0 | |
| **273** | 1 | 1 | 1 | 1 | 1 | |
| **322** | 1 | 0 | 0 | 0 | 0 | |

Compute their frequency for each 7-symptoms

```
pct_difficulty_breathing_weighted    21
pct_stuffy_runny_nose_weighted       17
pct_cough_weighted                   15
pct_aches_muscle_pain_weighted       17
pct_sore_throat_weighted             23
pct_nausea_weighted                  21
pct_chills_weighted                  21
Name: 1, dtype: int64
```

Average symptoms per country in Cluster 1

```
3.4615384615384617
```

Cluster 2 country list:

```
Cluster 2 has a data size of  (26, 10)
Countries in cluster 2 :
['Italy', 'Canada', 'Austria', 'Czech Republic', 'Malaysia', 'New Zealand', 'United Kingdom', 'Slovakia', 'Kuwait', 'Polan
'Switzerland', 'Greece', 'Finland', 'Portugal', 'Sweden', 'Norway', 'Germany', 'Hungary', 'Slovenia', 'Saudi Arabia', 'Aus
a', 'Ireland', 'France', 'United Arab Emirates', 'Spain', 'Denmark']
```

Extract their values of the 7-symptoms found in task 3c)

| | pct_difficulty_breathing_weighted | pct_stuffy_runny_nose_weighted | pct_cough_weighted | pct_aches_muscle_pain_weighted | pct_sore_throat_weighted |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 | 0 |
| 5 | 1 | 1 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 |
| 22 | 1 | 1 | 0 | 1 | 0 |
| 35 | 0 | 1 | 0 | 0 | 0 |
| 37 | 0 | 0 | 0 | 1 | 0 |
| 39 | 1 | 1 | 0 | 0 | 0 |
| 44 | 0 | 1 | 0 | 1 | 0 |
| 51 | 0 | 0 | 0 | 0 | 0 |
| 53 | 1 | 1 | 1 | 1 | 0 |
| 55 | 0 | 0 | 0 | 0 | 0 |
| 57 | 1 | 1 | 1 | 1 | 1 |
| 63 | 0 | 1 | 1 | 1 | 0 |
| 77 | 1 | 1 | 1 | 1 | 0 |
| 78 | 0 | 1 | 1 | 0 | 0 |
| 80 | 1 | 1 | 0 | 0 | 0 |
| 115 | 0 | 0 | 0 | 1 | 0 |
| 122 | 1 | 1 | 1 | 1 | 0 |
| 123 | 0 | 1 | 0 | 1 | 0 |
| 138 | 1 | 1 | 1 | 0 | 1 |
| 153 | 0 | 0 | 0 | 0 | 0 |
| 194 | 1 | 1 | 1 | 0 | 1 |
| 297 | 1 | 1 | 1 | 1 | 1 |

Compute their frequency for each 7-symptoms

```
pct_difficulty_breathing_weighted    13
pct_stuffy_runny_nose_weighted       18
pct_cough_weighted                   11
pct_aches_muscle_pain_weighted       13
pct_sore_throat_weighted              5
pct_nausea_weighted                   5
pct_chills_weighted                   7
Name: 1, dtype: int64
```

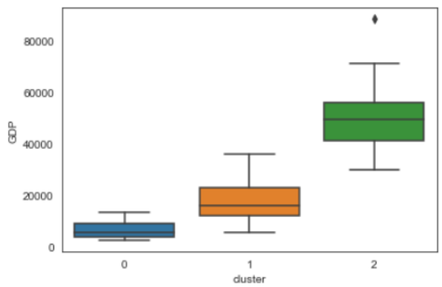
Average symptoms per country in Cluster 2

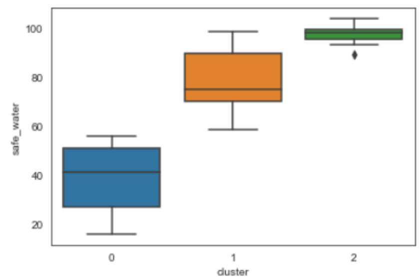
```
2.769230769230769
```

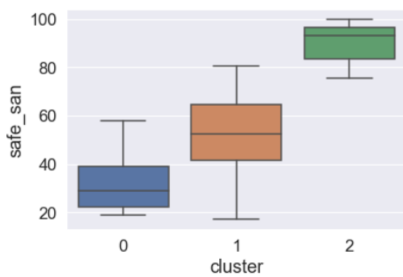
From the above additional analysis, we can see that Cluster 0 and Cluster 1 have high and similar average number of symptoms that lead to Covid-19 per country.

Interpretation:
This means Cluster 2 has more mild illness for Covid-19 comparing to Cluster 0 and Cluster 1.
And also there will be less spread of Covid-19 among the countries in Cluster 2 as less symptoms which lead to high Covid-19 cases is found.
Citizens in Cluster 2 have stronger immune system as they have less symptoms.

Why Cluster 2 is having better situation?
box-plot visualization from task 1d)




From above, we can assume that Cluster 0 is 3rd world country groups, Cluster 1 is developing country groups while Cluster 2 is developed country group

Since Cluster 0 and Cluster 1 are poorer (much less GDP per capita than Cluster 2), they may not have the resource to improve more hygienic water services and sanitation service. With less hygienic water and environment, citizens in Cluster 0 and Cluster 1 may have weaker immune systems, leading to more symptoms when Covid-19 appear.

Also, poorer government may lead to less medical resources and education on Covid-19. There may be less medical support to cure patients with Covid-19 and patients may not have sufficient money to visit the doctors or do not have related knowledge on the symptoms of Covid-19, leading to delay in treatment so more symptoms are found (more serious) than Cluster 2 which have the most resources with most hygienic and developed water and sanitation services. So this vicious cycle, lead to more spread.

Suggestion:

1. Cluster 0 and Cluster 1 country group could improve their policies related to resources and education

Below are 3 kinds of policy we extracted which are most related: Government provide direct cash payment to people who lose their jobs or cannot work, Government freeze financial obligations for households and Public info campaigns for Covid-19.

Direct cash payments and freezing financial obligations could relief medical expense or no income if isolated in hospital when having Covid-19, so could motivate patients to visit the doctors as soon as possible to prevent more spread and more numbers of symptoms.

Cluster 0 Cluster 1 Cluster 2

```
E1_Income support_1                 6.0
E1_Income support_2                 NaN
E2_Debt/contract relief_1           2.0
E2_Debt/contract relief_2           4.0
H1_Public information campaigns_1   NaN
H1_Public information campaigns_2  10.0
Name: 1, dtype: float64
```

```
E1_Income support_1                 22
E1_Income support_2                  9
E2_Debt/contract relief_1           11
E2_Debt/contract relief_2           20
H1_Public information campaigns_1    3
H1_Public information campaigns_2   36
Name: 1, dtype: int64
```

```
E1_Income support_1                 7.0
E1_Income support_2                18.0
E2_Debt/contract relief_1           9.0
E2_Debt/contract relief_2          15.0
H1_Public information campaigns_1   NaN
H1_Public information campaigns_2  26.0
Name: 1, dtype: float64
```

From the above, we can see that:

For Cluster 0: An average 2.2 policies per country and on average 1.4 stricter policies

For Cluster 1: An average 2.6 policies per country and on average 1.7 stricter policies

For Cluster 2: An average 2.9 policies per country and on average 2.3 stricter policies

So, we suggest Cluster 0 and Cluster 1 to take Cluster 2 as an example and implement these policies if haven't and stricter if implemented.

2. However, as above mentioned, Cluster 0 and Cluster 1 do not have enough resources, leading to less ability to implement more financial or education related policies. So, we also suggest Cluster 2 country group to subsidize cluster 0 and cluster 1. From moral perspective, Cluster 2 have stronger ability and sufficient resource on responding Covid-19 situation, they could donate additional resources to Cluster 0 and 1 like technology or knowledge transfer related to Covid-19. From prevention perspective, since now it is a globalization century, even though Cluster 2 countries could much effectively response to Covid-19, worse situations in Cluster 0 and 1 still have high probability to spread covid-19 to cluster 2 (developed countries)

As below we can see that it is true Cluster 0 and Cluster 1 covid-19 may spread to Cluster 2.

Cluster 0 Cluster 1 Cluster 2

```
C8_International travel controls_1    NaN
C8_International travel controls_2    3.0
C8_International travel controls_3    1.0
C8_International travel controls_4    5.0
```

```
C8_International travel controls_1     3
C8_International travel controls_2     2
C8_International travel controls_3     7
C8_International travel controls_4    25
```

```
C8_International travel controls_1    NaN
C8_International travel controls_2    5.0
C8_International travel controls_3   14.0
C8_International travel controls_4    7.0
```

For the policy of international travel controls (top 2 strict policy: control_3 and control_4),

Cluster 0: on average 0.60 top 2 strict policies per country and on average 0.5 most strict policy per country

Cluster 1: on average 0.82 top 2 strict policies per country and on average 0.64 most strict policy per country

Cluster 2: on average 0.81 top 2 strict policies per country and on average 0.27 most strict policy per country

We can see that Cluster 2 don't have stricter control on international travel controls so spread of Covid-19 from other country groups is still possible, helping others can still benefit their own too, especially currently there are variant virus coming up, reducing all threats globally can reduce the probability of more variant virus which will start another outbreak again to everywhere.

## 2nd Additional Analysis

**2. The 7 symptoms which lead to high Covid-19 cases**

According to task 3b) the 7 symptoms which lead to high cases are the following

```
pct_difficulty_breathing_weighted
pct_stuffy_runny_nose_weighted
pct_cough_weighted
pct_aches_muscle_pain_weighted
pct_sore_throat_weighted
pct_nausea_weighted
pct_chills_weighted
Name: 1, dtype: int64
```

Suggestion:

1. All countries and cities especially for Cluster 0 and Cluster 1 which have high amount of these symptoms on average, could focus more on testing these 7 symptoms for their citizens as they are more certain to be the symptoms of Covid-19, governments could implement compulsory Covid-19 testing and immediate isolation for patients who are found to have any of these symptoms. Countries and cities should also do public campaign on keeping citizens aware of these symptoms so citizens could visit the hospital as soon as possible when they found to feel chills or difficulty in breathing etc. to prevent further spread in society.

2. And for one of the symptoms "cough", countries and cities should also implement strict policy on wearing facial masks like compulsory facial masks wearing in public areas. This prevents droplet infections of Covid-19 as one of the most effective spread is through mist saliva for Covid-19.

Below are the facial covering policy for the three clusters:

Cluster 0 Cluster 1 Cluster 2

| Cluster 0 | | Cluster 1 | | Cluster 2 | |
|---|---|---|---|---|---|
| H6_Facial Coverings_1 | 1.0 | H6_Facial Coverings_1 | 1 | H6_Facial Coverings_1 | 1 |
| H6_Facial Coverings_2 | NaN | H6_Facial Coverings_2 | 9 | H6_Facial Coverings_2 | 10 |
| H6_Facial Coverings_3 | 4.0 | H6_Facial Coverings_3 | 16 | H6_Facial Coverings_3 | 5 |
| H6_Facial Coverings_4 | 2.0 | H6_Facial Coverings_4 | 6 | H6_Facial Coverings_4 | 2 |

For the policy of facial coverings (most strict mask policy = H6_Facial Coverings_4),

Cluster 0: on average implement 0.7 mask policy per country and on average 0.2 most strict mask policy per country

Cluster 1: on average implement 0.82 mask policy per country and on average 0.15 most strict mask policy per country

Cluster 2: on average implement 0.69 mask policy per country and on average 0.08 most strict mask policy per country

We can see all three clusters are not doing well for the strictest mask wearing policy: "Compulsory facial mask wearing outside home at all times regardless of location or presence of other people", They should all implement this policy to prevent further outbreak of Covid-19.

## 3rd Additional Analysis

Effective policy for the Cluster 0 and Cluster 2 are found in task 2b, are they implementing these policies to prevent new cases percentage of Covid? (implement the or stricter effective policies)

Cluster 0:
Effective policy from task 2b:
['H2_Testing policy_1', 'H3_Contact tracing_1', 'C2_Workplace closing_2', 'H1_Public information campaigns_2', 'C1_School closing_3']
Frequency of the policies above including the stricter policies (for example there is “workplace closing_3” which is a stricter policy than “workplace closing_2”, so if “workplace closing_2” is an effective policy, we should also look at “workplace closing_3”)

```
C1_School closing_3                  6.0
C2_Workplace closing_2               4.0
C2_Workplace closing_3               2.0
H1_Public information campaigns_2   10.0
H2_Testing policy_1                  7.0
H2_Testing policy_2                  2.0
H2_Testing policy_3                  NaN
H3_Contact tracing_1                 3.0
H3_Contact tracing_2                 5.0
Name: 1, dtype: float64
```

So, we can say that on average the countries in Cluster 0 implemented the above policies (there are 10 countries in cluster 0):
School closing: 6/10 = 0.6
Workplace closing: (4+2)/10 = 0.6
Public Information Campaign: 10/10 = 1
Testing policy: (7+2+0)/10 = 0.9
Contact tracing: (3+5)/10 = 0.8
Average effective and stricter policies implemented per country in Cluster 0: 3.9

Cluster 2:
Effective policy from task 2b:
['C1_School closing_2', 'E2_Debt/contract relief_2', 'H6_Facial Coverings_2', 'H1_Public information campaigns_2', 'C3_Cancel public events_2', 'C8_International travel controls_3', 'H3_Contact tracing_2', 'E1_Income support_2', 'H2_Testing policy_2']
Frequency of the policies above including the stricter policies

```
C1_School closing_2                   6.0
C1_School closing_3                  12.0
C3_Cancel public events_2            16.0
C8_International travel controls_3   14.0
C8_International travel controls_4    7.0
E1_Income support_2                  18.0
E2_Debt/contract relief_2            15.0
H1_Public information campaigns_2    26.0
H2_Testing policy_2                  10.0
H2_Testing policy_3                  11.0
H3_Contact tracing_2                 20.0
H6_Facial Coverings_2                10.0
H6_Facial Coverings_3                 5.0
H6_Facial Coverings_4                 2.0
Name: 1, dtype: float64
```

So, we can say that on average the countries in Cluster 1 implemented the above policies (there are 26 countries in cluster 2):
School closing: (6+12)/26 = 0.69

Cancel public event: 16/26 = 0.62
International travel controls: (14+7)/26 = 0.81
Income Support: 18/26 = 0.69
Debt/contract relief: 15/26 = 0.58
Public Information Campaign: 26/26 = 1
Testing policy: (10+11)/26 = 0.81
Contact tracing: 20/26 = 0.77
Facial coverings: (10+5+2)/26 = 0.65

Average effective and stricter policies implemented per country in Cluster 2: `6.615384615384615`

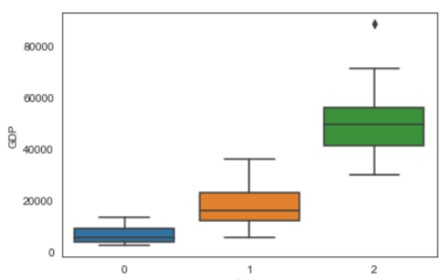
Suggestion:

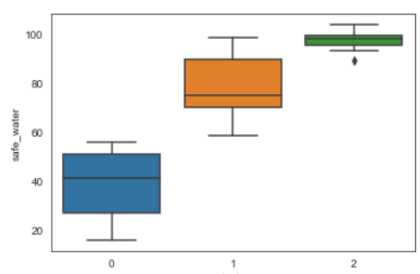
1. From above since school and workplace closing are the two lowest value for cluster 0, this country group should implement “require closing all levels” for schools and university. While for workplace, they should at least require closing or work from home for some sectors or categories of workers, if implemented already, they could also level up to the stricter policy “require closing or work for home for all-but-essential workplaces (eg grocery stores, hospitals)”

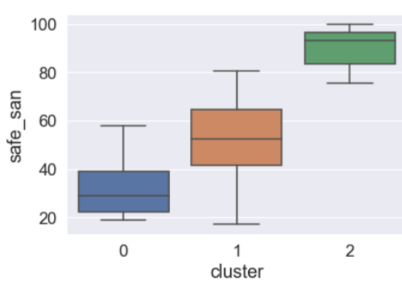
2. From above since Debt/contract relief and Cancel public event are the lowest value for cluster 2, this country group should implement “broad debt/contract relief” like freezing financial obligations for households. While for cancel public event, government should cancel all public events like concerts or exhibitions which may have high density in these events leading to quicker and easier spread of Covid-19 during these events.

4th Additional Analysis

box-plot visualization from task 1d)




Hong Kong’s value of GDP, safe_water and safe_san

```
GDP: $[62375.11937000001]
Safe_water: [100.0]
Safe_San: [91.77344928]
```

We can see that Hong Kong belongs to cluster 2 since all values for GDP, safe_water and safe_san are all within the highest boxplot as shown above which belongs to Cluster 2.
Hong Kong's situation on whether implemented the effective policies for cluster 2 found in part 2b:

| | C1_School closing_2 | C3_Cancel public events_2 | C8_International travel controls_3 | E1_Income support_2 | E2_Debt/contract relief_2 | H1_Public information campaigns_2 | H2_Testing policy_2 | H3_Contact tracing_2 | H6_Fa Covering |
|---|---|---|---|---|---|---|---|---|---|
| 29 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | |

No policy in school closing, international travel control and facial coverings. However, maybe Hong Kong have stricter policy for these missing policies, so extract them below:

| | C1_School closing_3 | C8_International travel controls_4 | H2_Testing policy_3 | H6_Facial Coverings_3 | H6_Facial Coverings_4 |
|---|---|---|---|---|---|
| 29 | 1 | 1 | 0 | 0 | 0 |

Suggestion:
From the above data we can see that the only policy which must be improved is facial coverings as Hong Kong haven't reach the effective policy for this category.
While test policy has reach the standard of effective policy, however HK can still improve and try to implement strictest test policy if possible, so more invisible cases could be found out quickly and cut down the spreading chain which create a better protective screen.

5th Additional Analysis

The equal-sized percentiles of population density computed in task 4b).

```
quantiles of population density:
0.0       3.247871
0.2      30.595784
0.4      77.470851
0.6     107.981487
0.8     235.253514
1.0    7952.998418
Name: pop_density, dtype: float64
```

Extract the Hong Kong's population density value.

| country_name | country_code | pop_total | pop_density |
|---|---|---|---|
| Hong Kong | HKG | 7507400 | 7096.190476 |

We can see that Hong Kong's population density falls within the highest percentiles (5th quantile).
Extract the top 5 attributes that are most correlated to "new_cases_percentages" for the 5th quantile (not absolute value).

```
Computing top 5 attributes that are most correlated to new_cases_percentages for 5th quantile
residential_percent_change_from_baseline              0.292798
driving                                               0.022303
parks_percent_change_from_baseline                   -0.020180
walking                                              -0.021731
grocery_and_pharmacy_percent_change_from_baseline    -0.156423
Name: new_cases_percentages, dtype: float64
```

Interpretation:
Hong Kong, which belongs to the 5th quantile, have higher risk of Covid-19 infection at residential area and commute through driving.

1. We predict that since Hong Kong is considered among the high dense locations, residential areas may be relatively denser, too. With more people in a small residential area, it is more likely to spread Covid-19, like through the lift and corridor. Professor Yuen Kwok-yung also mentioned that contact of the front door passcode button and lift button in the residential building may lead to more spread especially when it is touched by many different people (residents/food deliver/maintenance worker). Also, with less space in residential corridor, there is more probability to have more contact with others comparing to places like USA where people mostly resides in their own houses instead of buildings (less contact with other people).

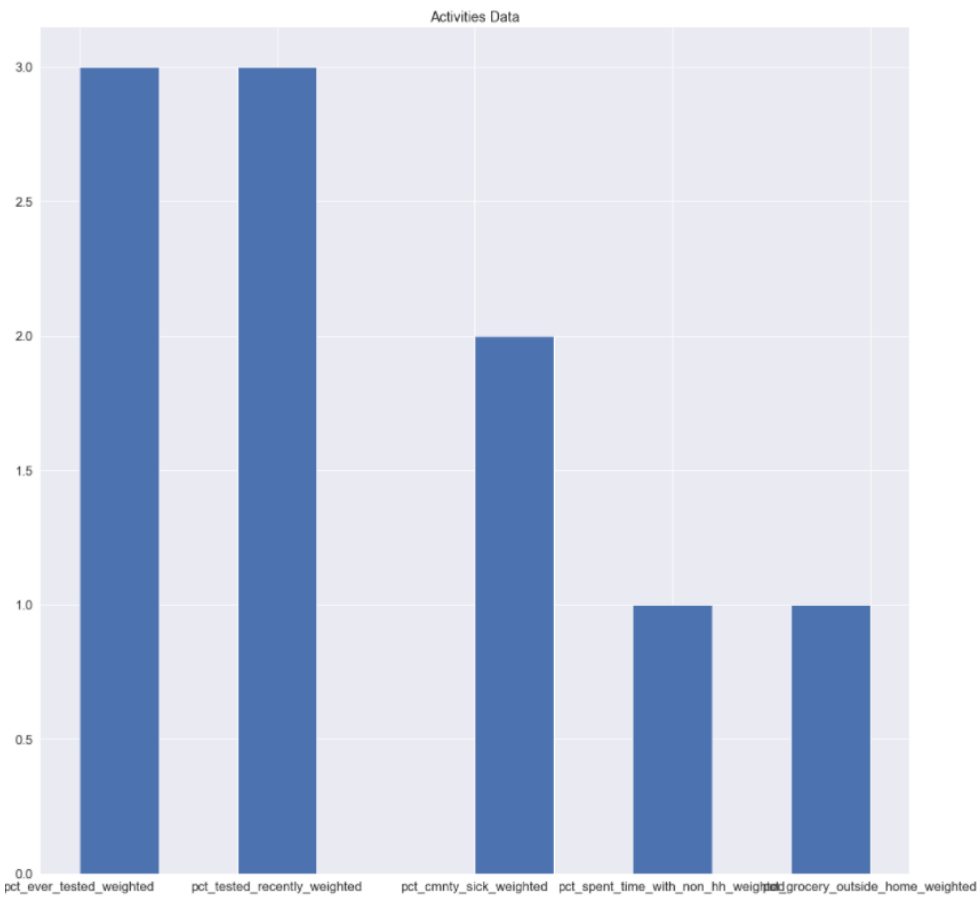
2. From the recent Hong Kong news, we can see that commute through taxi is usually one of the riskier travelling methods. Since taxi drivers are in higher and closer contact with different passengers, there will be higher cross-infections between passengers and taxi drivers.

Suggestions:
For residential areas, janitors should sterilize the lift and front door buttons every hour and do frequent cleaning for the whole building like the corridor to reduce the Covid-19 virus density. HK citizens should also wash their hands or use alcohol to sterilize their hands after touching the buttons.
For Driving, HK government should do frequent and compulsory Covid-tesing for taxi and Uber drivers as they have the highest risk of getting infected and spreading the virus to passengers. HK citizens should also prevent taking taxi and Uber if not necessary.

Extracting the related policy from the dataset (Staying at home prevent button contact and human contact; Restricting internal movement can reduce taxi-taking; Contact tracing can find which passengers taxi drivers served and find the high-contact people quickly),

| | C6_Stay at home requirements_1 | C6_Stay at home requirements_2 | C6_Stay at home requirements_3 | C7_Restrictions on internal movement_1 | C7_Restrictions on internal movement_2 | H3_Contact tracing_1 | H3_Cor tracir |
|---|---|---|---|---|---|---|---|
| 29 | 1 | 0 | 0 | 0 | 0 | 0 | |

HK do not have strict policy on staying at home and restricting internal movement, so HK government can require citizens not leaving house with exceptions for daily exercise, grocery shopping and essential trips.
Government can also recommend not to travel between cities if not necessary or even with some restrictions for internal movements after 10pm.
6th Additional Analysis




Human activities that lead to high Covid-19 cases

Cluster 0 Cluster 1 Cluster 2

```
pct_ever_tested_weighted               5
pct_tested_recently_weighted           6
pct_cmnty_sick_weighted                8
pct_spent_time_with_non_hh_weighted    2
pct_grocery_outside_home_weighted      3
Name: 1, dtype: int64
```

```
df.sum()/10
```

```
2.4
```

```
pct_ever_tested_weighted              11
pct_tested_recently_weighted          10
pct_cmnty_sick_weighted                5
pct_spent_time_with_non_hh_weighted   19
pct_grocery_outside_home_weighted     17
Name: 1, dtype: int64
```

```
df2.sum()/26
```

```
2.3846153846153846
```

```
pct_ever_tested_weighted              13
pct_tested_recently_weighted          14
pct_cmnty_sick_weighted               15
pct_spent_time_with_non_hh_weighted   15
pct_grocery_outside_home_weighted     21
Name: 1, dtype: int64
```

```
df1.sum()/39
```

```
2.0
```

Interpretation:

Why testing lead to high cases? Testing are suppose to discover more Covid-19 cases and also with more people infected, more people are motivated to take the testings, so cases will be high.

Excluding ever_tested and tested_recently, the average frequency of each cluster is:
Cluster 0: (8+2+3)/10 = 1.30
Cluster 1: (12+12+21)/39 = 1.31
Cluster 2: (5+19+17)/26 = 1.46

We can see that Cluster 2 have the highest average human activities followed by Cluster 1 then Cluster 0.

Suggestion:
Citizens in Cluster 2 should prevent high contact with people other than their family members during the hard time of Covid-19. They should also prevent frequent visit to grocery stores if not necessary, and keep at least 1.5 meter between other customers to prevent high contact with each other. They should also sterilize their clothes and belongings and wash their hands after meeting other people and going back from grocery stores.

For Hong Kong's value:

| | pct_ever_tested_weighted | pct_tested_recently_weighted | pct_cmnty_sick_weighted | pct_spent_time_with_non_hh_weighted | pct_grocery_outside_home_weighted |
|---|---|---|---|---|---|
| 29 | 0 | 0 | 0 | 0 | 1 |

Suggestion:
Hong Kong should do more testing for covid-19 to discover patients as soon as possible to prevent continue spreading of the virus.
While same as Cluster 2's suggestions, HK citizens should also prevent frequent visit to grocery stores if not necessary, and keep at least 1.5 meter between other customers to prevent high contact with each other. They should also sterilize their clothes and belongings and wash their hands after going back home from grocery stores.

Additional prediction on surprising findings
Why Cluster 1 which have higher GDP and more developed water and sanitation service still have slightly higher average number of symptoms than Cluster 0 (milder illness)?
We have three explaination:
1. facial coverings

| Cluster 0 | | Cluster 1 | | Cluster 2 | |
|---|---|---|---|---|---|
| H6_Facial Coverings_1 | 1.0 | H6_Facial Coverings_1 | 1 | H6_Facial Coverings_1 | 1 |
| H6_Facial Coverings_2 | NaN | H6_Facial Coverings_2 | 9 | H6_Facial Coverings_2 | 10 |
| H6_Facial Coverings_3 | 4.0 | H6_Facial Coverings_3 | 16 | H6_Facial Coverings_3 | 5 |
| H6_Facial Coverings_4 | 2.0 | H6_Facial Coverings_4 | 6 | H6_Facial Coverings_4 | 2 |

For the policy of facial coverings (most strict mask policy = H6_Facial Coverings_4),
Cluster 0: on average implement 0.7 mask policy per country and on average 0.2 most strict mask policy per country
Cluster 1: on average implement 0.82 mask policy per country and on average 0.15 most strict mask policy per country
Cluster 2: on average implement 0.69 mask policy per country and on average 0.08 most strict mask policy per country

As above, we can see that Cluster 0 have highest value on the strictest mask policy

2. human activities

Human activities that lead to high Covid-19 cases

Cluster 0

```
pct_ever_tested_weighted                5
pct_tested_recently_weighted            6
pct_cmnty_sick_weighted                 8
pct_spent_time_with_non_hh_weighted     2
pct_grocery_outside_home_weighted       3
Name: 1, dtype: int64
```

```
df.sum()/10
```

```
2.4
```

Cluster 1

```
pct_ever_tested_weighted               13
pct_tested_recently_weighted           14
pct_cmnty_sick_weighted                15
pct_spent_time_with_non_hh_weighted    15
pct_grocery_outside_home_weighted      21
Name: 1, dtype: int64
```

```
df1.sum()/39
```

```
2.0
```

Cluster 2

```
pct_ever_tested_weighted               11
pct_tested_recently_weighted           10
pct_cmnty_sick_weighted                 5
pct_spent_time_with_non_hh_weighted    19
pct_grocery_outside_home_weighted      17
Name: 1, dtype: int64
```

```
df2.sum()/26
```

```
2.3846153846153846
```

Interpretation:

Why testing lead to high cases? Testing are suppose to discover more Covid-19 cases and also with more people infected, more people are motivated to take the testings, so cases will be high.

Excluding ever_tested and tested_recently, the average frequency of each cluster is:

Cluster 0: 1.30

Cluster 1: 1.31

Cluster 2: 1.46

We can see that for human activities without COVID testing, Cluster 0 have the lowest movement in these three areas (cmnty_sick, spent_time and grocery).

And also focusing on the two testing value (ever_tested and tested recently)

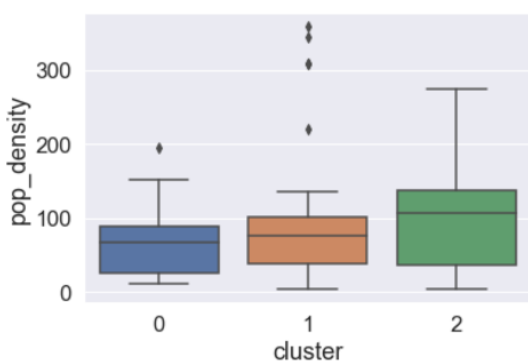
Cluster 0 (10 countries): (10+11)/10 = 1.1

Cluster 1 (39 countries): (13+14)/39 = 0.69

Cluster 2 (26 countries): (10+11)/26 = 0.81

We can see that Cluster 0 have highest estimated percentage of people taking the Covid-19 test, this can conclude that more testing could find Covid-19 patients more and quicker before the patient become more serious and start to appear more related symptoms, faster testing could cure Covid-19 when you only have mild symptoms and illness.

3. Population Density




We can see that the population density for cluster 1 is a bit higher than cluster 0. Since as mentioned, one of the most effective way of spreading for Covid-19 is through contact which include saliva mist or infecting through eyes, nose and mouth through hand contact (also number and seriousness of the symptoms are correlated to the amount of virus bearing), in denser areas, there might be higher probability to contact more amount of virus bearing and lead to higher infection possibility of Covid-19. So Cluster 1 has slightly higher symptoms than cluster 0 (a bit more serious illness).

Even though Cluster 2 have highest population density, however more sufficient and effective policies and resources are implemented to prevent widespread of covid, so the situation is still better.

Task 6

This task has chosen three types of models that are covered in the lectures – decision tree, support vector machine, and neural network. As neural network may have a different data preprocessing, so we separate this task into two parts to study which models would obtain the best accuracy score.

Part 1 – decision tree, random forest, and support vector machines

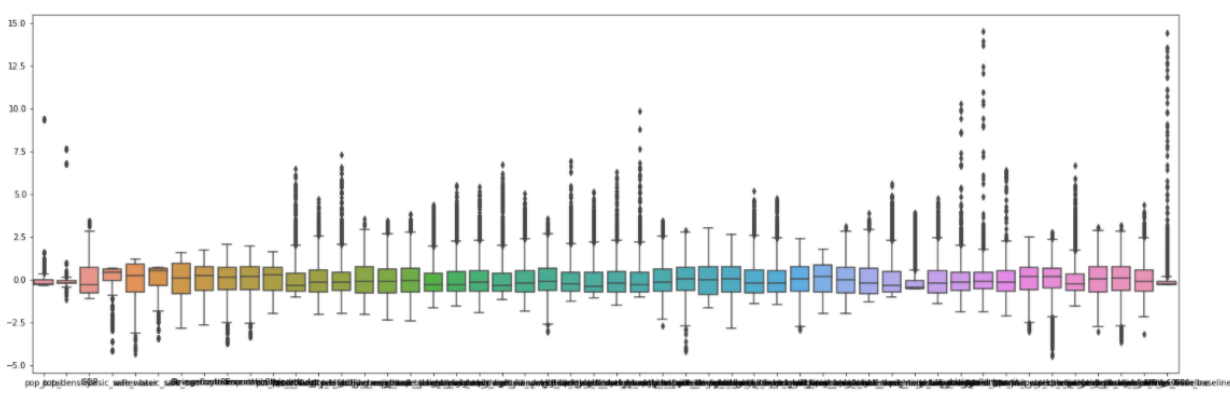
a. Data preprocessing

Missing data and standardization

After confirming that there is no duplicated rows, we can then move on to deal with missing values. We apply the same approach as task 1a to fill in missing value and standardization. All numeric columns, including columns with binary values, are fit into IteractiveImputer() function and transform the dataset to fill in missing value. We include binary columns to train the imputer, as we suggest binary columns act as an additional information which may enhance the Imputer to build a better function to fill in the missing data. Then we extract the numeric columns, excluding binary columns this time, and use StandardScalar() function to standardize them. Here is a photo showing part of the data frame containing the numeric columns after filling in missing data and standardizing:

| | pop_total | pop_density | GDP | basic_water | safe_water | basic_san | safe_san | StringencyIndex | GovernmentResponseIndex |
|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.321534 | -0.243881 | -0.931539 | -0.380915 | -0.133931 | 0.562727 | -0.494859 | 0.464881 | 0.193299 |
| 1 | 0.061303 | -0.072700 | 0.815299 | 0.593151 | 0.826126 | 0.664564 | 1.217384 | -0.473848 | 0.070585 |
| 2 | -0.100165 | -0.272559 | 1.137595 | 0.592624 | 0.962932 | 0.687651 | 0.760089 | -0.039028 | 0.499854 |
| 3 | -0.285582 | -0.266201 | -0.768463 | 0.057314 | -1.119590 | -1.046179 | -1.194567 | 1.220415 | 0.622797 |
| 4 | -0.304326 | -0.170266 | 1.488031 | 0.638436 | 0.964555 | 0.718530 | 1.235118 | -1.665026 | -0.889605 |

Outlier detection



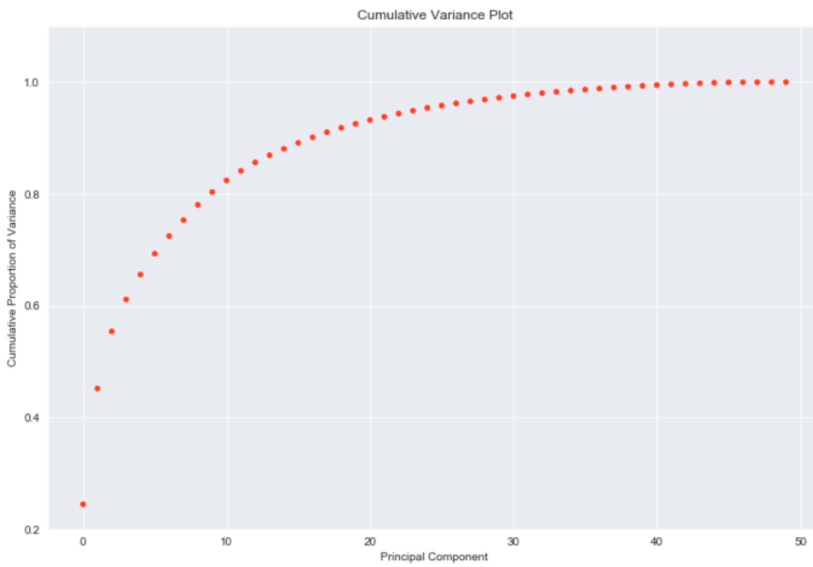
As we plot a boxplot and study the outlier distribution of each attribute, most of the attributes are found to contain numerous outliers. Thus, we decide to study the records and see if there are records considered as outliers.

```
Number of deleted data:   2601
data remained:  1263
```

Using the same approach (same boundary) as task 1, more than two-third of the dataset is considered as outliers. As the matter of it, it is favourable to not reduce the dataset to such as small size. If we delete this kind of "outliers" and train the model, the test set is very likely to contain these "outliers" and, thus, certainly result into a low accuracy.

Detection of Multicollinearity

We also would like to further study the correlations between each attribute. PCA is applied to measure the cumulative variance of the numeric columns (excluding binary columns).




Unlike the mini-project, the numeric columns are distinctive, and the cumulative variance does not rise quickly that more than one-third of the numeric columns cover 90% of the entire data information. It may still be acceptable to reduce around half of the numeric columns for model creation. However, considering binary columns as well, binary columns, in fact, constitutes large portion of the dataset. Such reduction of numeric data's dimensionality may not bring to a significant change in overall dimensionality. To avoid complicating the overall model creation process, it is better to retain the original data dimensionality.

Data Frame Setup for Model Creation

Excluding the need of outlier deletion and dimensionality reduction, we choose to set up data frame for model creation. The first two columns, country names and country codes, are removed and all numeric and binary columns are retained to form a data frame variable called "df". "df" containing the data after all missing data are filled in. The second data frame, named "df2", is similar as the first one, but with the numeric columns to be standardized. Thus, we wish to study whether standardizing the data would allow a higher accuracy of model performance.

Grid Search Setting

Now, we firstly split the dataset into train and test data with the ratio of "8:2". Normally, there are three ways of weighting of train to test data – "9:1", "8:2", and "7:3". For decision tree model creation, we think "9:1" is suitable for dataset with large number of rows. For our dataset we have 2048 features, but just 2060 rows. So, "9:1" will move too many data to train the model and there will be too less for testing the model. Vice versa for adopting "7:3".

For the following section, we will use GridSearchCV() function to train the models, It allows high simplicity of the model creation process as setting hyperparameter tuning, cross validation, and scoring selection can all be done within the function. Our scoring measurement is set to be "f1_weighted" ("f1" is fine too as the label classes are evenly distributed), instead of "accuracy", as we believe it is a better measurement of accuracy. Besides, we choose Stratified K-Fold Cross Validation for model creation as it can certainly reduce the overfitting problem the most.

As the dataset is split into a ratio of "8:2", number of splits for cross validation would set to be 5, so only 1 out 5 splits is used for test set. The accuracy scoring used for grid search would be "weighted f1 score", referring to the weighted average taking all split test set into account.

b. Decision tree model

For parameters, there are few inputs for decision tree modelling that we think should be suitable and important to consider – criterion for classification, maximum depth, minimum samples split, and ccp_alpha.

df1:

For first round of grid search, the parameter is set, and the best parameter set is computed as followed:

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Gini |
| Maximum depth | 3 to 10 | 10 |
| Minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 2 |
| Ccp_alpha | 0,0.005,0.01,0.015,0.02 | 0 |

The highest average accuracy: 72.21% (correct to 2 decimal places)

Observing the result in first round, the parameter is adjusted a bit in the second round to further investigate the performance in a larger range of parameter values (ccp_alpha is removed as it does not show a significant role in first round):

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Gini |

| | | |
|---|---|---|
| Maximum depth | 11 to 15 | 10 |
| Minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 2 |

The highest average accuracy: 73.39% (correct to 2 decimal places)

df2:
same approach is applied in here.
(first round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Entropy |
| Maximum depth | 3 to 10 | 10 |
| Minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 8 |
| Ccp_alpha | 0,0.005,0.01,0.015,0.02 | 0 |

(second round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Gini |
| Maximum depth | 11 to 15 | 14 |
| Minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 2 |

The highest average accuracy (among two rounds): 73.08% (correct to 2 decimal places)

Combining the model performances in both dafa frames, as both data frames show a similar accuracy score, it indicates standardization would not really enhance the model performance. The best performance of decision tree model would obtain around 70% - 75%.

c. Random Forest model
After training decision tree model, we would also suggest an advanced tree classification model – random forest, and see if it would allow a better performance. For parameters, we keep using the same inputs mentioned above – criterion for classification, maximum depth, minimum samples split, and ccp_alpha. Plus, we would add one more crucial parameter, n_estimators, which is unique to random forest parameter tuning. Same approach is applied to the model training mprocess.

df1:
(first round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Entropy |
| maximum depth | 5 to 9 | 9 |
| n_estimators | 100, 500, 1000 | 1000 |
| minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 2 |
| ccp_alpha | 0,0.005,0.01 | 0 |

(second round – ccp_alpha is also removed due to insignificant effect shown in first round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Entropy |
| maximum depth | 10 to 20 | 16 |
| n_estimators | 500, 1000, 1500 | 500 |
| minimum samples split | 2, 5, 8 | 5 |

The overall highest average accuracy: 81.70% (correct to 2 decimal places)

df2:
(first round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Entropy |
| maximum depth | 5 to 9 | 9 |
| n_estimators | 100, 500, 1000 | 1000 |
| minimum samples split | 2, 4, 6, 8, 10, 12, 14 | 2 |
| ccp_alpha | 0,0.005,0.01 | 0 |

(second round)

| Parameter | Range | Best input |
|---|---|---|
| criterion | Gini and entropy | Entropy |
| maximum depth | 10 to 20 | 16 |
| n_estimators | 500, 1000, 1500 | 1000 |
| minimum samples split | 2, 5, 8 | 5 |

The overall highest average accuracy: 81.66% (correct to 2 decimal places)

Combining the model performances in both data frames, random forest model also shows similar findings as decision tree, but with a better performance – achieving nearly 82%. Standardization also does not enhance the model performance. It is reasonable as the fact that standardization will not change the nature of numeric data, but rather suits for outlier detection and dimensionality reduction analysis.

d. Support vector machine model

In addition to decision tree and random forest model, SVM is also a powerful model taught in lectures. However, it is only effective in standardized samples, so "df" won't not used in this modelling.

There are several kernel choices for decision functions. We would first undergo a simple model training and observe which one would lead to outstanding performance, so we could focus on one kernel for grid search to avoid time-consuming if possible.

```
Method: Polynomial kernel
              precision    recall  f1-score   support

         0.0       0.76      0.15      0.25       208
         1.0       0.28      0.93      0.42       188
         2.0       0.76      0.16      0.26       198
         3.0       0.89      0.27      0.42       179

    accuracy                           0.37       773
   macro avg       0.67      0.38      0.34       773
weighted avg       0.67      0.37      0.33       773

Method: RBF kernel
              precision    recall  f1-score   support

         0.0       0.85      0.76      0.80       196
         1.0       0.58      0.73      0.64       176
         2.0       0.68      0.62      0.65       186
         3.0       0.86      0.83      0.85       215

    accuracy                           0.74       773
   macro avg       0.74      0.73      0.73       773
weighted avg       0.75      0.74      0.74       773

Method: Sigmoid kernel
              precision    recall  f1-score   support

         0.0       0.59      0.64      0.61       187
         1.0       0.46      0.41      0.43       190
         2.0       0.32      0.30      0.31       193
         3.0       0.61      0.66      0.63       203

    accuracy                           0.50       773
   macro avg       0.50      0.50      0.50       773
weighted avg       0.50      0.50      0.50       773

Method: Linear kernel
              precision    recall  f1-score   support

         0.0       0.75      0.82      0.78       195
         1.0       0.63      0.63      0.63       212
         2.0       0.62      0.54      0.58       182
         3.0       0.82      0.85      0.83       184

    accuracy                           0.71       773
   macro avg       0.71      0.71      0.71       773
weighted avg       0.70      0.71      0.70       773
```

RBF kernel obtains the highest accuracy, so we choose to focus on this kernel in grid search.

For parameter tuning, we choose three main inputs – C value, gamma value, and decision function shape.
(first round)

| Parameter | Range | Best input |
|---|---|---|
| C value | 0.1, 1, 10, 100 | 1 |
| Gamma value | 0.001, 0.01, 0.1, 1, 'auto' | 0.1 |
| Decision function shape | 'ovo', 'ovr' | 'ovo' |

As the input ranges for C and gamma value are broad, they are further adjusted in second round.
(second round)

| Parameter | Range | Best input |
|---|---|---|
| C value | 0.5, 1, 1.5 | 1.5 |
| Gamma value | 0.05, 0.1, 015, 0.2 | 0.1 |
| Decision function shape | 'ovo', 'ovr' | 'ovo' |

The overall highest average accuracy: 80.97% (correct to 2 decimal places)

e. First part reflection

| Model | Highest possible performance class (in 5% interval) |
|---|---|
| Decision tree | 70 – 75% |
| Random forest | 80 – 85% |
| Support vector machine | 80 – 85% |

SVM and random forest model both obtain a high accuracy, roughly 81%~82%. Personally, SVM would be considered as the best model among these three types of models. Although its highest accuracy score is 1% slightly lower than random forest. Training random forest model uses several hours and is, thus, time-consuming.

Besides with time-consuming of random forest, it would be better to undergo standardization and PCA in the grid search as well. However, as there are binary columns, we must first extract the numeric columns for standardization and PCA and combine those back to the binary columns. Such preprocessing would definitely increase the complexity of grid search and result into even longer processing time. In fact, the same concept is also applied to SVM too.

Overall, the model performance is still optimistic with over 80% of accuracy. However, it would be more confident if we could achieve an accuracy of over 90% since it would likely to get lower accuracy score in the final test set. Thus, we would like to move on a more powerful model – neural network, in the second part.

Part 2 – Neural Network Model

1. Data Preprocessing
   We first filled the columns with null value with IterativeImputer from sklearn.impute
   The we dropped total_cases column.
   The we did standardization (to a scale of [-1,1]) on data columns except policy data columns and new_cases_percentages. The reason for this exception is policy data is categorical attributes and do not require standardization.
2. Global Variable declaration
   We declared some global variables for training including Epoch, number of target classes, device to run pytorch, etc.
   For hyperparameter tuning, it is laborious to perform k-fold cross validation. So we chose to use cross validation for training and testing during hyperparameter tuning.
   There are approximately 3800 records available and we want to use as many data as possible to train the model. Therefore, we used a cross validation threshold of 0.1. Namely there will be around 3420 training data and around 380 validation data.
3. Hyperparameter Tuning
   We used a module named optuna to perform hyperparameter tuning. Optuna is a hyperparameter tuning tool for different machine learning models. For pytorch, it can tune

and prune model structure and optimizer. It tunes the parameter by looping over different parameters and selecting the set of parameters with best score.

In our model, we tuned model structure, optimize and batchsize. We chose to tune these parameters as they are the most significant features of pytorch neural network model. For loss function, we used the cross entropy loss. For simplicity we did not prune it.

For model structure, we tuned the number of layers, number of output features in each layer and dropout for each layer. We looped through one to four layers with output features 4 to 88 (4 for final classes, 88 for initial input) and dropout of 0.3 to 0.6. The reason for the range of dropout is that we wanted to set an average dropout range and we consider 0.3 to 0.6 to be appropriate. Dropout values within this range is not too high nor too low.

For optimizer, we tuned the type, learning rate and momentum of the optimizer. For type, we chose to tune between SGD and RMSprop as these two functions have learning rate and momentum as parameters. We just need one set of code to tune these parameters after choosing the type of optimizer (if we chose to add other methods like Adam we need one more set of code to tune because Adam do not have momentum as parameter). For learning rate, chose to tune value between 0.001 to 0.2 as common learning rates is around this range. For momentum we chose to tune between 0.01 to 0.9 and that is pretty much the range of momentum.

For batch size we set 5 different batch sizes from [10,50,100,150,200] and ran the optuna tuning with each batch size.

For the optuna tuning process, a ‘study’ will be created. Then for each study we have 1000 trials. For each trial we select different parameters and build and train models using pytorch neural network technique. Note that the training and validation data were declared as global variables and the results of each trial are therefore comparable.

4. Testing and validating the model

   After hyperparameter tuning, we should have the best model parameters. However, we still need to validate it with different data splits.

   As we were unable to find k-cross validation function for neural network, we did 10-fold cross validation with our own function.

   First, we built an untrained new model with best parameters. Then we split the dataset, trained the new model with 90% of the data and tested it with the remaining records. The whole process is repeated 10 times and we took the average as the final validation score.
5. Final training and saving the model

   The final step is to train the model with all the available data and prepare for the testing dataset. We also saved the model as model.pt.

From the result we can see validation score for neural network is above 90% and is the highest among all the models we built. Therefore, we chose the neural network model as the final model for this question.