Unsupervised time series analysis of country wise COVID data

Swiss Data Science Center
Machine learning project 2

Fall 2020

Supervisor: Ekaterina Krymova

Faustine Vigneau, Maxence Courtet, Marc Vandelle

December 17, 2020

Abstract

In this context of worldwide epidemic we tried to use machine learning technics, more precisely clustering, to obtain from countries wise number of Covid-19 cases daily report models that can at the same time provide a good analysis and visualization of the data and some more or less precise prediction on the evolution of the epidemic.

1 Introduction

The year 2020 was deeply touched by the Covid-19 epidemic, therefore we wanted to use this project to try to develop tools that could be useful in the management and analysis of the current crisis. The goal of our project is dual. First, we want to make it possible to visualize the evolution of the epidemic in different countries and thus to offer the possibility of a comparison over time of the number of Covid cases between each of these countries. The second objective is to try to develop models capable of making forecast on the number of expected cases of Covid-19 in the coming days for each country. Our project revolves around the pipeline presented in Figure 1 and consists of 4 main parts. First of all, it will be question of explaining how we preprocess the data provided by the Swiss data science center and then of introducing the different methods that we used to form clusters from this data. The last two parts will finally allow us to present our results both for data visualization but also for the prediction model and its efficiency.

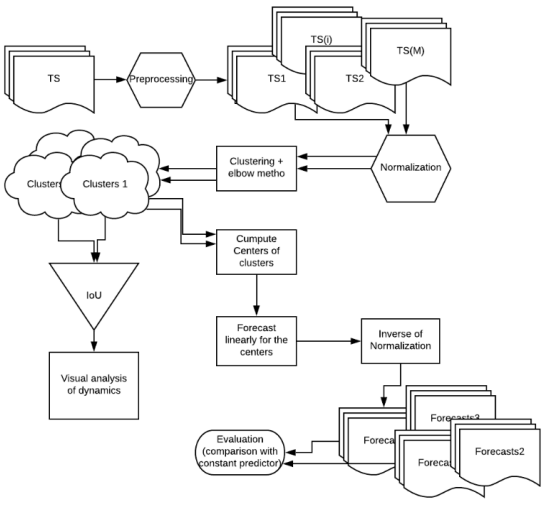

Figure 1: Pipeline of the project.

2 Data presentation

The data we use for this project is a csv file which contains daily reports about the COVID situation for 198 different countries. These daily reports gives us the total population, the number of Covid-19 cases, the cumulative number of Covid-19 cases for 14 days over 10000 people and the number of deaths caused by the virus.

3 Data preprocessing

3.1 Data cleaning

First, we observed that for some countries, the daily report showed negative number of cases. It is probably due to the fact that diagnosis may change over time and to correct their data countries subtract a certain number of cases on one random day which can lead to these negative number of cases. Therefore, we started by replacing these values by 0. After that, since the daily report doesn't start at the same date for all countries, we chose to only keep data from countries which have a history of observations since the first April of 2020. We set it as the starting date. We also observed that for some countries the number of cases could stay low for a long period of time. That's why we added a feature that groups these countries to form a special cluster. We put in this cluster time series for which a proportion superior of a given threshold of the values was under a given value ϵ.

3.2 Time series construction

From this cleaned data, we only kept the number of cases for each day and constructed 3 different types of time series, which are precisely described in the next point. Then, for each type, we defined smallest time series by considering sliding windows over all the period of time the data covers (see an example for Switzerland in Figure 3). Finally, we normalized each sub time series by its maximum number of cases.

Average number of cases per week To obtain the first type of time series, we computed the average number of cases over each week for each country. In this case, we formed 9 sub time series of 16 weeks each by considering a sliding window of length 16 weeks and step 2 weeks. In particular, this means that the first sub time series goes from the first week to the 16th week, the second one from the 3th week to the 18th week, etc... You can find a more precise

scheme in Figure 2. It will help us later tuning our forecast.

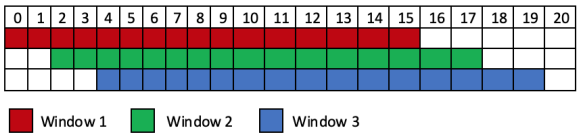

Figure 2: Table of distribution of 3 sliding windows over 21 weeks with size of 16 weeks and step of 2 weeks.

Number of cases per day The second type of time series is just daily cases as in the original data. Here, we considered a sliding window of size 60 days and step of 30 days.

Cumulative number of cases per day The last type of time series we used is the daily cumulative number of cases computed from the daily cases of the data (we do not use the cumulative field from the data since it was cumulative over the last 14 days and not over all the data as we wanted). Here, we also considered a sliding window of size 60 days, step of 30 days.

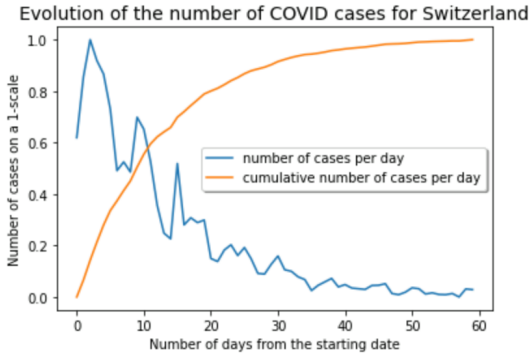

Figure 3: Evolution of COVID cases for Switzerland for the 60 first days from the starting date

4 Clusters and Barycenters

Now that we have correctly preprocessed the data, our goal is to identify and grouping similar time series. Therefore, we use clustering methods. Following the advice of our supervisor, we try to cluster the data using two algorithms known to be efficient and accurate clustering of Time Series : Time Series K-Means and KShape from the ts learn library.

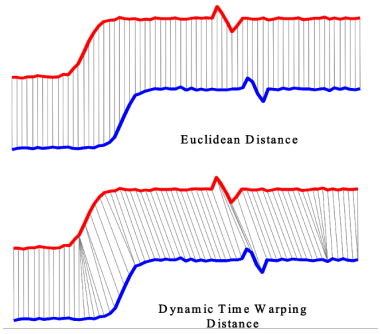

Figure 4: Difference between Euclidean distance and DTW

4.1 Clustering the data and computing the barycenters

For any clustering method, similarity between data points is measured with a distance metric. The most commonly used is the Euclidean distance. In our case, we chose to use the Dynamic Time Warping Distance (DTW) metric as it suits best time series. It is due to the fact that the nonlinear alignment of the DTW gives a more intuitive distance even if two similar time series are not aligned in the time axis (as shown in Figure 4). Moreover, we calculated the center of each cluster for both DTW and soft DTW metrics which turned out to be more efficent for the forecast. Indeed, soft DTW is a smoothed formulation of DTW. The figure 5 shows an example of one of the clusters and its soft dtw barycenter.

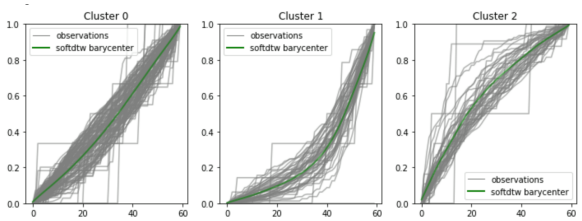

Figure 5: Exemple of 3 clusters

4.2 Finding the optimal number of clusters

We tried to determine the optimal number of clusters to use in order to fit our data. We used the elbow method for inertias (see an example in Figure 6). Indeed, the elbow method is used in determining the number of clusters in a data set. We repeated this method for the three types of time series we have constructed and for both kmeans and kshape clustering algorithms, our results for kmeans are presented

in the Table below and you can find our results for kshape in the jupyter notebooks. Important fact is that for kmeans the number of clusters stay the same for all sliding windows, when instead for kshape the number of clusters can vary depending of the sliding window.

Type of data	Weekly	Daily	Cumul. daily
nbr of cluster	3	5	3

Table 1 : Result of the elbow method for kmeans

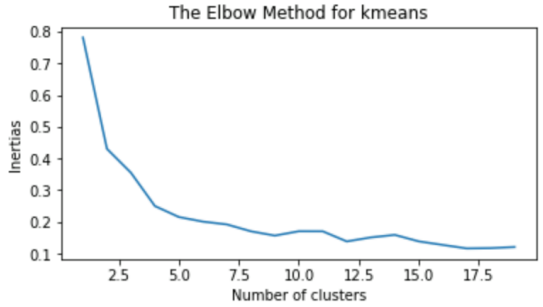

Figure 6

5 Time series visualization

Like it is presented in the introduction, one of our goal during this project was to be able to observe the evolution over time of the Covid-19 epidemic for different countries and to highlight the similarity and difference between them. Thus, we used 2 ways of visualizing the data: Dendrograms and Sankey diagrams.

5.1 Dendrogram

Forst, we wanted to compare the evolution of Covid-19 cases between different countries. For this mean, we used the library scipy.cluster.hierarchy to draw dendrograms with our data. A dendrogram is a tree that connects the most similar countries in term of evolution of number of Covid-19 cases between them using hierarchical clustering (a method of cluster analysis that seek to build a hierarchy between clusters). We measure the dissimilarity between clusters of objects using average dissimilarity (average linkage), you can see the result for one sliding window in Figure 8.

5.2 Sankey diagrams

After looking at the similarity between cluster for each sliding window, we wanted to observe their evolution in time. We used a type of flow diagram called Sankey diagram with the help of the holoviews library. The Sankey diagram is composed of a set of line that represent the flow of country going from on cluster to one another over all the sliding windows we constructed before. Each line has a thickness proportional to the flow rate it represents. We add a feature that allow the user to show the composition of each flow (a list of countries) by pointing them with the mouse. You can use this to get a better understanding of the Sankey diagram in the notebook (or html notebook version).

6 Forecast

The second goal of our project was to allow the forecasting of the number of Covid-19 cases. For this purpose, we use two different forecasts : a naive forecast and an auto regression forecast. The naive one is used to determine the quality of the auto regression forecast. After forecasting the predicated values, we denormalized them to allow the comparison with real values.

6.1 Naive forecast

The naive forecast is pretty simple, it consists of repeating one value during a certain amount of days/weeks. We will detail this for each type of data.

Weekly data naive forecast For the weekly data, we just use the last value of the time series and repeat it to forecast the next 4 weeks.

Daily data naive forecast To forecast the next 7 days in the case of the daily data, we repeat the average of the last 7 values of the time series.

Daily cumulative data naive forecast The naive forecast of the daily cumulative data is exactly the same as the daily data forecast except that we make a cumulative sum of the number of cases per day at the end.

6.2 Autoregression forecast

Auto regression is a method that use previous value of a time series as input to a regression equation to predict future value for this particular time series. To this end, we used the statsmodels library. In the next parts we will present you the details of our computation for each type of time series.

Weekly autoregression forecast We use the auto regression presented above with the last 2 values of the soft dtw barycenter of the cluster corresponding to the time series we want to forecast. Like in the naive forecast, we use this to predict 4 weeks ahead.

Daily autoregression forecast To predict the 7 days ahead in the case of daily data we once again use the last 2 values of the soft dtw barycenter of the cluster corresponding to the time series we are interested in.

Daily cumulative autoregression forecast Just like the naive approach, we use autoregression in the same as than for the daily data and add the obtained values between them to form a cumulative result.

6.3 Quality of forecast

We determine the quality of our forecast, done with both kmeans and kshape clustering, in two steps.

Absolute error First, we compute the absolute error of the naive forecast and the auto regression forecast compared to the real observed values. We also compute the normalized version of this absolute error. We use the formulas below for the weekly data, i being the ith week prediction and K being the number of time series :

$$AE_{f_i} = \frac{1}{K}\sum_{k=1}^{K} abs(weekly_{ik} - forecasted_{ik})$$

$$normAE_{f_i} = \frac{1}{K}\sum_{k=1}^{K} \frac{abs(weekly_{i,k} - forecasted_{ik})}{weekly_{ik}}$$

Relative improvement We compute the relative improvement between the naive forecast and the auto regression forecast. If this score is positive, it means that our auto regression forecast has beaten the naive forecast. We use the formula below for the weekly data, i being the ith week prediction :

$$\frac{AE_{bench_i} - AE_{f_i}}{AE_{bench_i}}$$

Results We present our results in the following table, representing the percentage of time where the AR forecast beats the naive forecast, and this for each week prediction and each clustering method :

Forecast week	1	2	3	4
Kmeans	0.586	0.586	646	0.717
Kshape	0.606	0.369	0.015	0.005

Table 2 : Result of the relative improvement on weekly data

We observe that for the first predictions of weekly data, the AR forecast using kshape clustering is more efficent than the one using kmeans, but its quality decays rapidly for the 2nd, 3rd and 4th week prediction whereas the kmeans keeps a more stable quality.

The Figure 7 represents the relative improvement between the AR forecast and the naive one for five different countries on weekly data for Kmeans clustering.

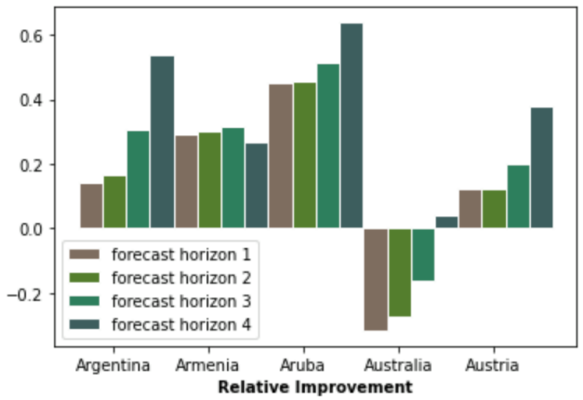

Figure 7: Relative improvement for weekly data

7 Conclusion

In this project, we first preprocessed the data to keep only the meaningful parts. We then clustered the data using two different methods, namely k-means and k-shape. We optimized the number of clusters using the elbow method. To assess the quality of our clustering, we tried different visualization techniques on the clusters such as Sankey diagrams and dendrograms. Once the clustering done, we predicted the future number of COVID-19 cases all around the world. We assessed the quality of this prediction by computing the relative improvement, which compares our forecast to a naive one. We end up beating the naive predictor most of the time when using k-means as clustering method. However, COVID-19 cases records being really noisy data depending on many external aspects such as lockdown or other restrictions, sometimes our model unfortunately doesn't beat a naive approach.

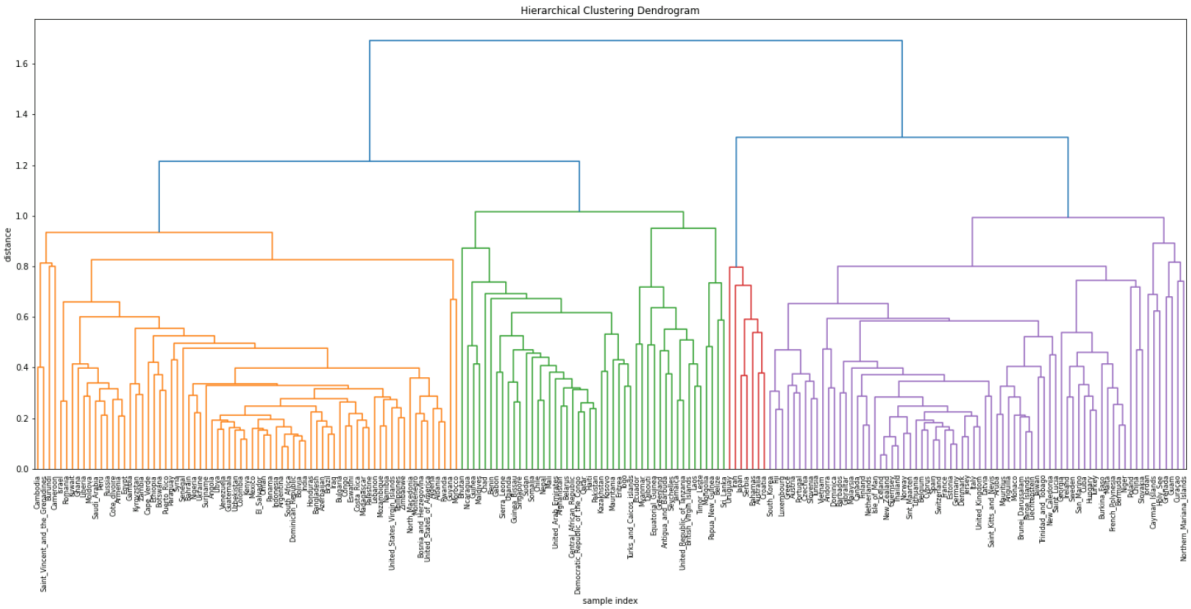

Figure 8: Dendrogram for weekly data