COVID-19 Infection Analysis and Prediction

ROSINA Maxime

ISEP - Institut Supérieur d'Électronique de Paris
IG.2310 - Introduction to Artificial Intelligence: Project
2020/2021

Summary

I - Analysis of the dataset

After numerous months affected by a sanitary crisis, the epidemiological situation is still evolving. I use this dataset which is daily updated. I downloaded my dataset December, 11, 2020.
The aim of this project is to apply artificial intelligence methods to this dataset in order to make analyzes and predictions to help the society to better understand the spread of the COVID-19 infection.

I.A - Dataset comprehension and missing values

In order to understand all the attributes contained in this dataset, you can take a look at some scientific web papers such as Nature[1]. In this database, each row represents a single individual case.
To begin, take a look at some important information such as the shape of the dataset, the missing values, the outliers, and the correlation between attributes.

The dataset is composed of 33 columns and 2,676,311 rows. There are a lot of missing values.

```
Checking missing values

ID                          0
age                   2098293
sex                   2096154
city                   977681
province               452664
country                   115
latitude                   61
longitude                  61
geo_resolution             61
date_onset_symptoms    2414712
date_admission_hospital    2560100
date_confirmation       108489
symptoms               2674259
lives_in_Wuhan         2671973
travel_history_dates    2673700
travel_history_location    2667089
reported_market_exposure    2675242
additional_information    2630456
chronic_disease_binary          0
chronic_disease         2676096
source                   566964
sequence_available      2676299
outcome                 2368929
date_death_or_discharge    2673163
notes_for_discussion    2675671
location                2662935
admin3                  2595877
admin2                  1850257
admin1                  1418753
country_new               30553
admin_id                     61
data_moderator_initials    933328
travel_history_binary      65579
```

First I dealt with the missing values coming from the column age. I replaced all the missing values from age with the median age. I choose to remplace by the median because: "*If the variable is skewed, the mean is biased by the values at the far end of the*

[1] "Epidemiological data from the COVID-19 outbreak ... - Nature." 24 mars. 2020, https://www.nature.com/articles/s41597-020-0448-0. Date de consultation : 13 déc.. 2020.

distribution. Therefore, the median is a better representation of the majority of the values in the variable."[2]

Then, I dropped all the rows with missing values from the attributes that I wanted to keep (`sex`, `country`, `lives_in_Wuhan`, `chronic_disease_binary`, `travel_history_dates`, `outcome`).

Finally, I had to manage a problem with the outcome column. In fact, data were not formalized, and many descriptions were given (`'Recovered'`, `'recovered'`, `'recover'` etc.). I formalized all the descriptions with three different values: `'died'`, `'hospitalized'` and `'discharged'`.

- ❏ `'died'` is the value for people who die from COVID-19
- ❏ `'hospitalized'` is the value for the people who are currently hospitalized because of COVID-19
- ❏ `'discharged'` is the value for the people that are leaving the hospital (including all recovering people).

	age	sex	country	lives_in_Wuhan	chronic_disease_binary	outcome
504	35	0	23	0	0.0	0
756	35	0	23	1	0.0	0
765	35	0	23	0	0.0	0
797	4	0	13	0	0.0	0
798	39	0	13	0	0.0	1
...	...	...	...	...	...	...
637804	35	0	11	0	1.0	1
657538	35	1	11	0	0.0	0
658592	35	0	11	0	0.0	1
658603	35	1	11	0	0.0	1
672418	30	0	45	0	1.0	1

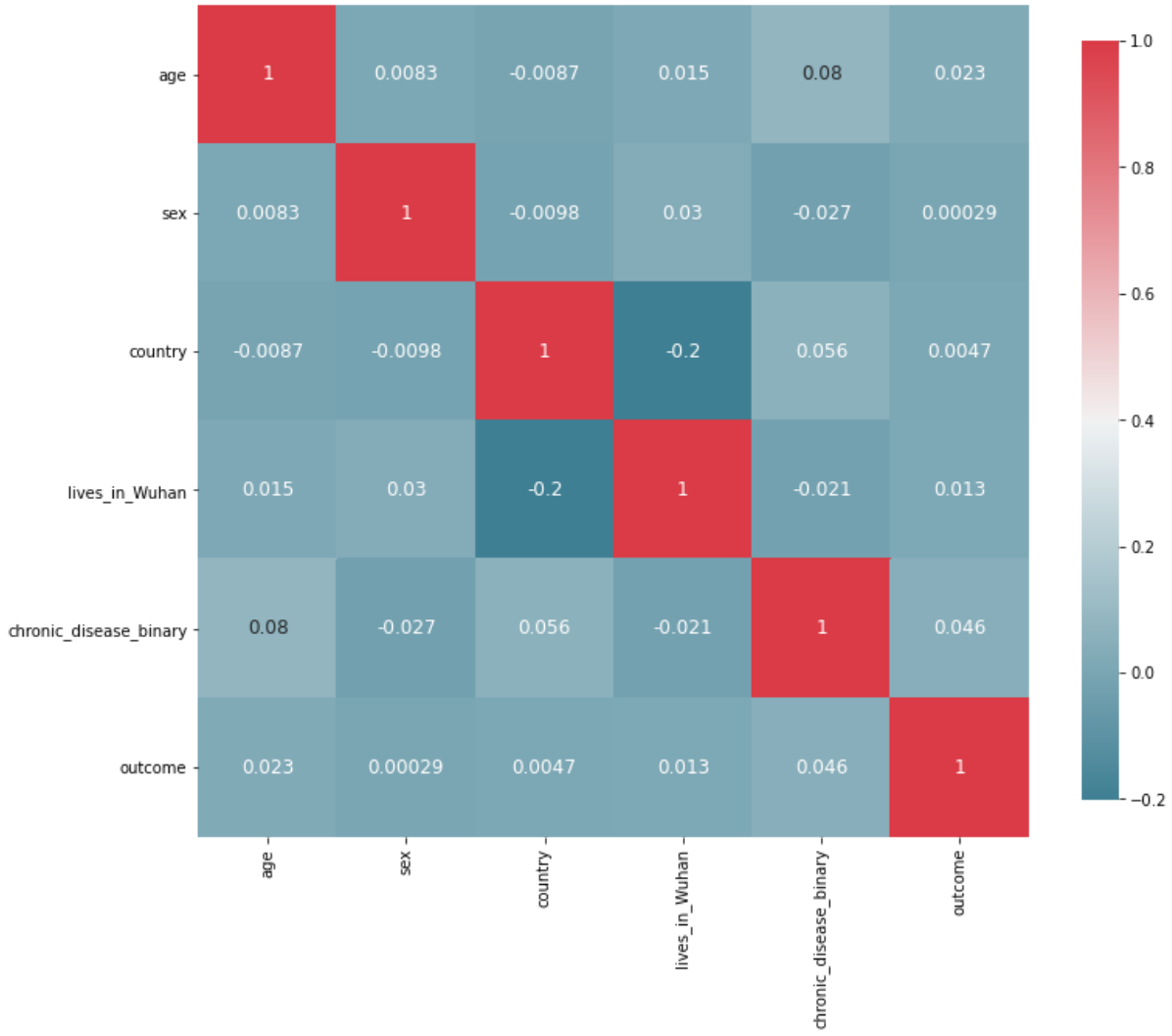
1583 rows × 6 columns

Our dataset is now ready !

[2] "Feature Engineering Part-1 Mean/ Median Imputation. | by ...." 17 août. 2020, https://medium.com/analytics-vidhya/feature-engineering-part-1-mean-median-imputation-761043b95379. Date de consultation : 13 déc.. 2020.

I.B - Correlation

I took a look at the correlation between the different attributes.



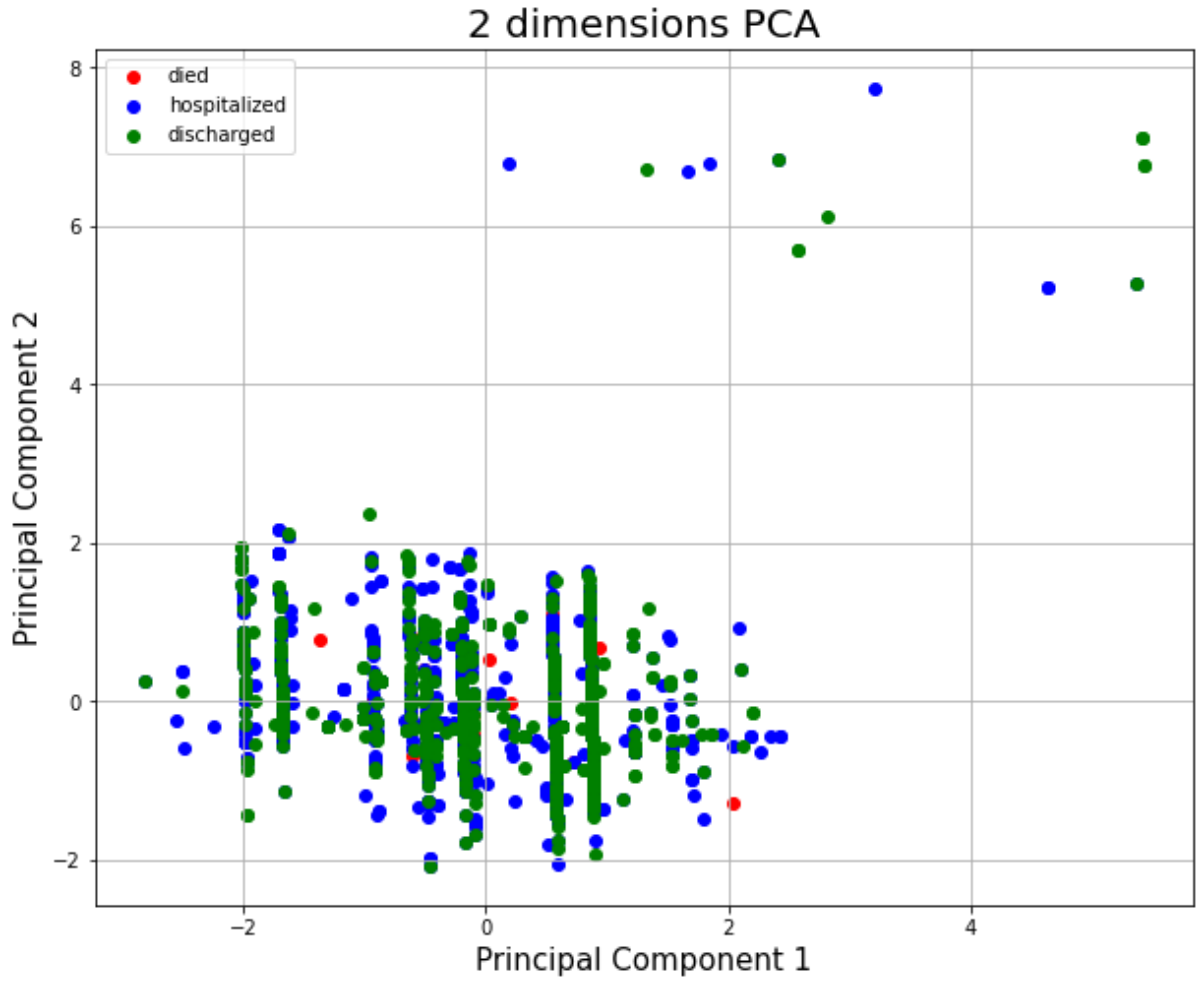
I obtained the correlation matrix above. We can see that the most correlated variables are:

- ❏ lives_in_Wuhan and country with -20%
- ❏ age and chronic_disease_binary with 8%
- ❏ country and chronic_disease_binary with 5.6%
- ❏ outcome and chronic_disease_binary with 4.6%

Any of the correlations above are surprising. The country and living in Wuhan are the most correlated because a lot of patients that were tested were coming from Wuhan. Age and chronic disease are correlated because most chronic illnesses happen when you are old. Outcome and chronic_disease_binary are also correlated because, we know nowadays, that an important part of the disease illness are coming from people who have some chronic illness in addition to the COVID-19.

I.C - PCA

Now we take a closer look with a 2 dimensions PCA.



We can see that the data is very unbalanced between `'died'`, `'hospitalized'` and `'discharged'` values. There only a few `'died'`.

II - Bayes Nets

We now compute some probability according to the data coming from our dataset.

II.A - Probability of having symptoms of Covid-19 after visiting Wuhan

The probability for a person to have symptoms of COVID-19 if this person visited Wuhan is 60.63 %

II.B - Probability of being a true patient if having symptoms

Probability for a person to be a true patient if this person has symptoms of COVID-19 and this person visited Wuhan is 98.42 %.

II.C - Probability to die after visiting Wuhan

Probability for a person to die if this person visited Wuhan is 25.32%.

II.D - Average recovery day for a patient after visiting Wuhan

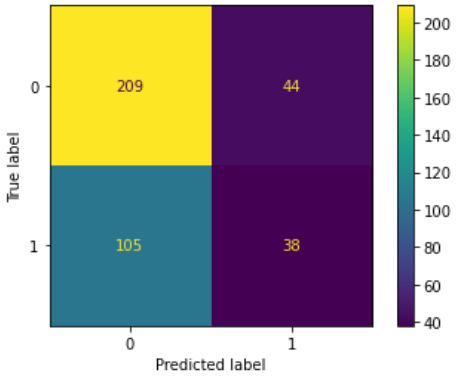
Average recovery interval for a patient if this person visited Wuhan is 18 days.

III - Machine Learning

III.A - K-Nearest Neighbours (K-NN) and Bayes Classification

KNN model and confusion matrix

	precision	recall	f1-score	support
0	0.67	0.83	0.74	253
1	0.46	0.27	0.34	143
accuracy			0.62	396
macro avg	0.56	0.55	0.54	396
weighted avg	0.59	0.62	0.59	396



Naive Bayes model and confusion matrix

To have better results I think that we need to work with more balanced data, and with a larger sample. In fact, we only have 67% of precision in the best case that is not really high.

III.B - Logistic Regression

Now I will use logistic regression in order to predict people age based on sex, country, lives_in_Wuhan, chronic_disease_binary, travel_history_dates, and outcome.
To do so I created some age categories in another column `ageGroup`. The four categories were:

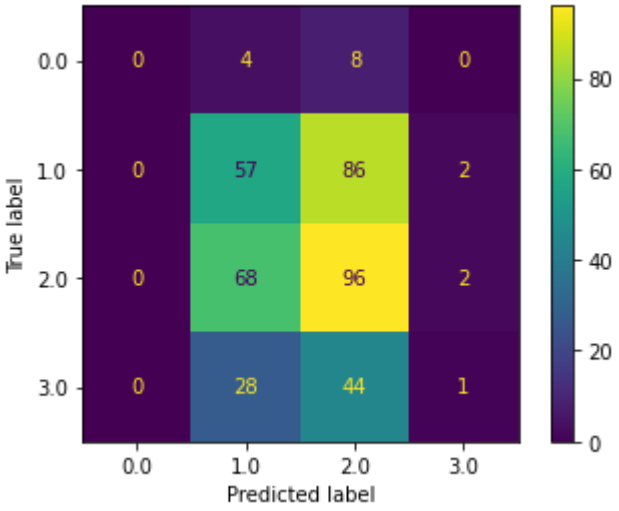
- ❏ 0 for people between 0 and 20 years old
- ❏ 1 for people between 20 and 40 years old

- 2 for people between 40 and 60 years old
- 3 for people between 60 years old and more (up to 150 years old)

I applied Logistic Regression and I got the following results.

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	12
1.0	0.36	0.39	0.38	145
2.0	0.41	0.58	0.48	166
3.0	0.20	0.01	0.03	73
accuracy			0.39	396
macro avg	0.24	0.25	0.22	396
weighted avg	0.34	0.39	0.34	396

MSE : 0.898989898989899



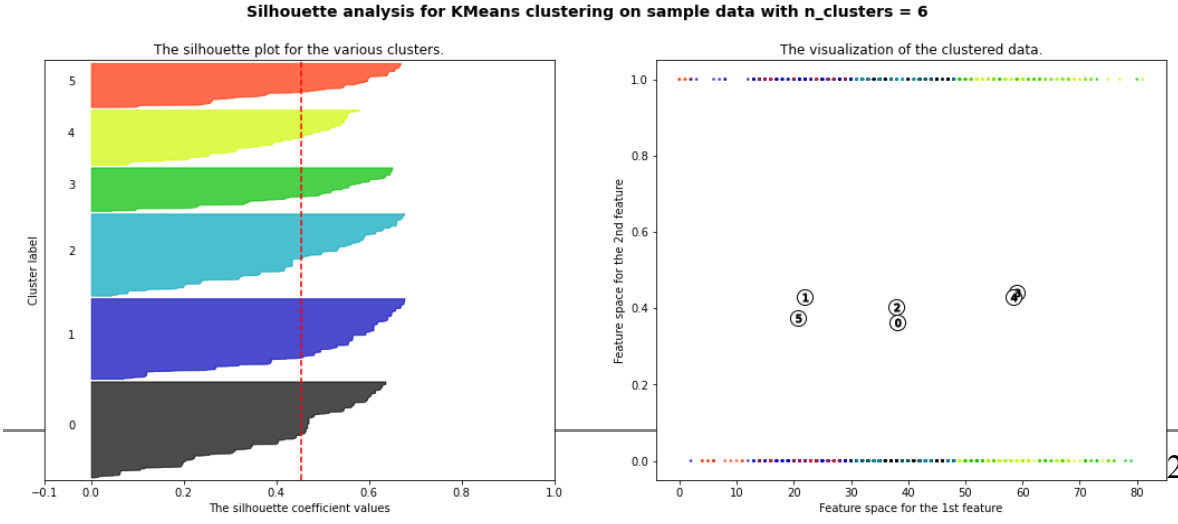
Logistic regression model, MSE and confusion matrix

The following results are quite surprising because we don't have many people in category 0. The data is once again unbalanced. But we have a good MSE (0.989). We can clearly see that the logistic regression model better predicts people in age category 1 and 2 than category 0 and 3. However the precision is very low.

III.C - K-Means

Now, we will apply K-Means to clusterize our data.

From the above silhouette score we can see that the score is higher with 6 clusters.



IV - Improving the results and Theoretical formalism

IV.A - Unbalanced data

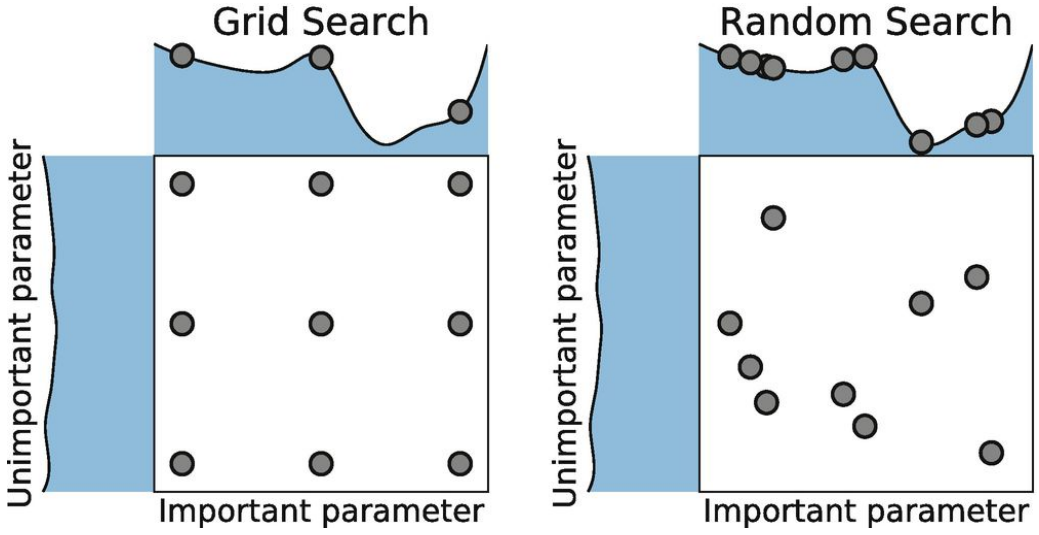
The data is unbalanced, especially for the column outcome. We have:
If we reduce randomly the majority class the balance of the data will increase but it is not enough. In fact, we have only 28 persons who died in comparison to more than 500. The models will certainly increase in precision.

IV.B - Manage missing values

We could manage the missing values in a better way instead of just deleting rows with missing values. For rows with binary values we could analyze and replace the missing values with the one with the highest probability.

IV.C - Grid-search algorithm

The grid-search algorithm is an automation algorithm used for Hyperparameter optimization. Hyperparameter optimization is the problem of choosing an optimal set of optimal parameters to optimize a model.



After applying the grid search algorithm to our different models I obtained the table below:

Model	Naive Bayes classifier	KNN	Logistic regression
Best score	0.6621848739495798	0.6529269334852585	0.4035393818544367
Best(s) parameter(s)	var_smoothing = 1.0	n_neighbors = 18	C = 10.0 penalty = 'l2'

We can say that the best model in our case is the Naive Bayes classifier, with a var_smoothing at 1.

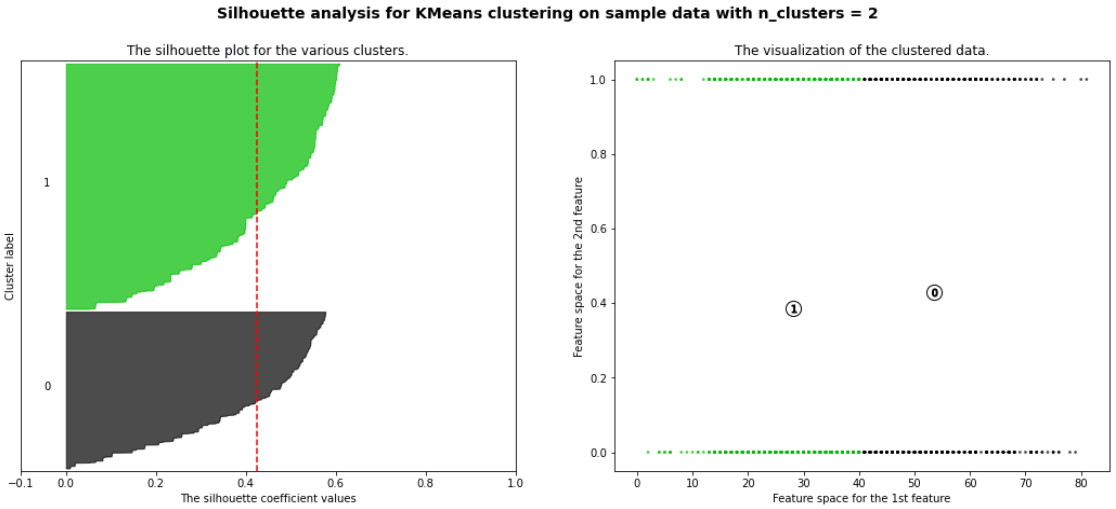
IV.D - Best model

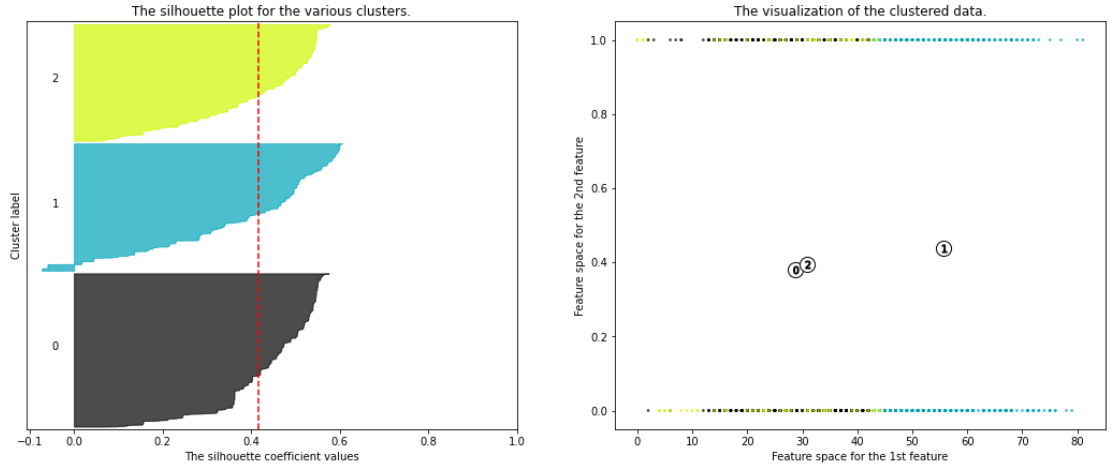
In our case, the best model is the Naive Bayes classifier. The parameter var_smoothing gives the best results when it sets at 1. This parameter is the portion of the largest variance of all features that is added to variances for calculation stability. The second parameter that we have not studied here is the priors corresponding to the prior probabilities of the classes.

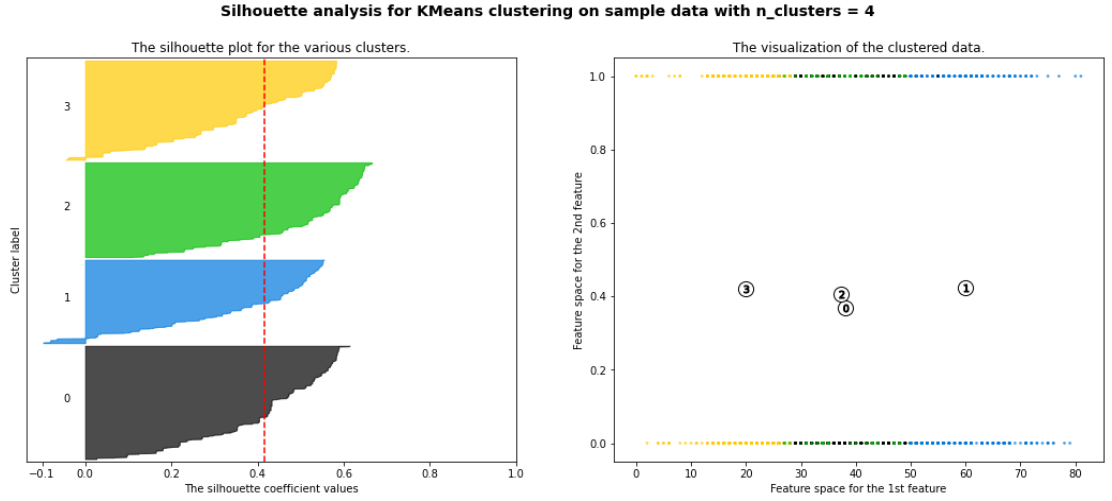
Conclusion

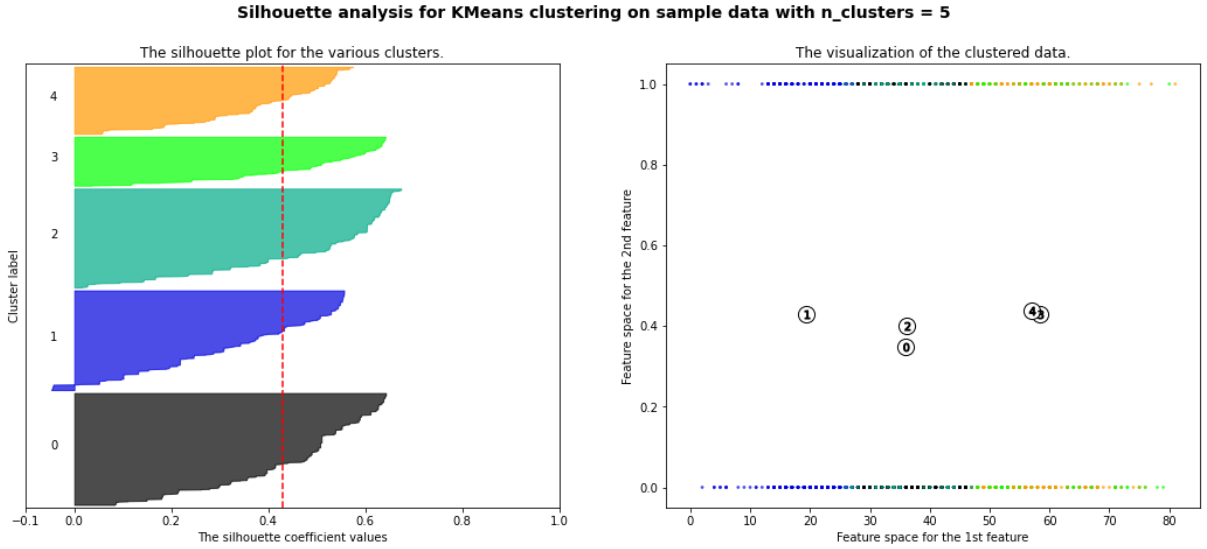
We applied some of the machine learning models to our Covid19 dataset. The results are that our dataset is not optimized to perform high accurate predictions. In order to overcome the problem, we need to better balance our data, to better manage our missing values, and to use the best model with the best parameters, here the Naive Bayes classifier.

Appendix


Silhouette analysis for KMeans clustering on sample data with n_clusters = 2
The silhouette plot for the various clusters.
Cluster label
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 3
The silhouette plot for the various clusters.
Cluster label
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 4
The silhouette plot for the various clusters.
Cluster label
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 5
The silhouette plot for the various clusters.
Cluster label
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature