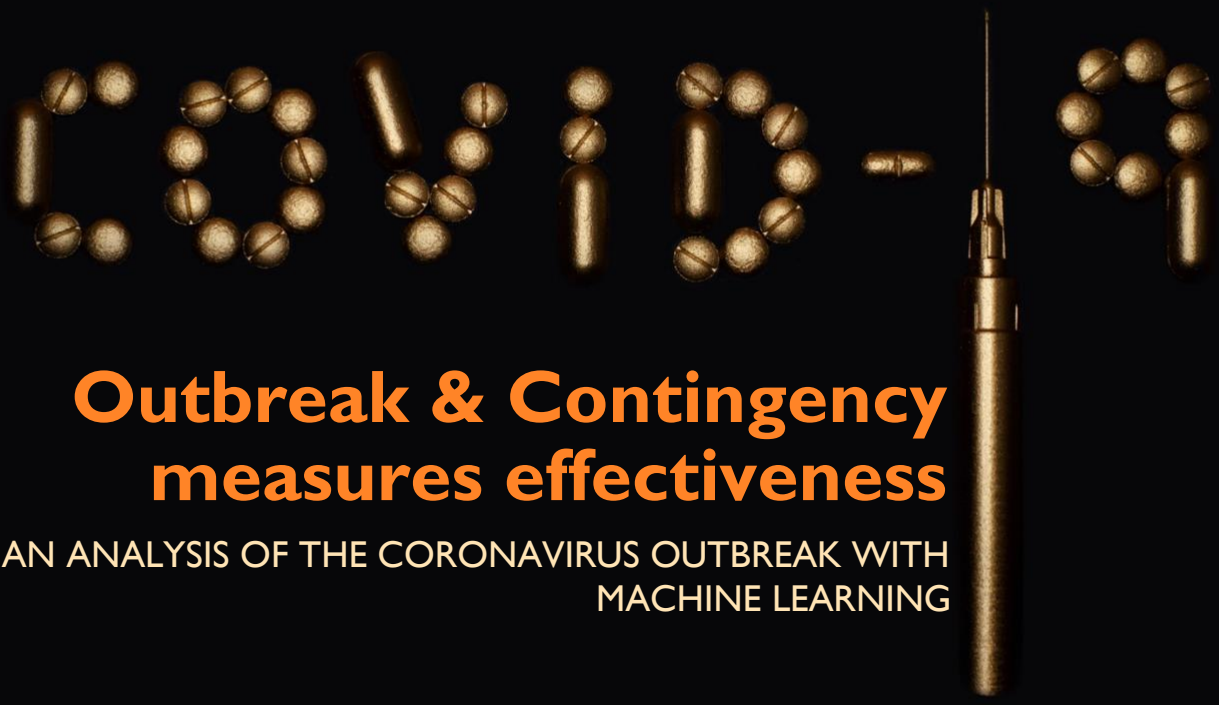


Outbreak & Contingency measures effectiveness

AN ANALYSIS OF THE CORONAVIRUS OUTBREAK WITH MACHINE LEARNING

Chandana Udupa, Mireia Riviere & Tianer Qi

Table of Contents

Introduction

All the coding and the visualizations can be found on the GitHub Directory:

- GitHub Link: https://github.com/chandanaUdupa/COVID-19-Data-Analysis

Problem Proposal

Recently, in the late months of 2019, a new virus named SARS-CoV-2 (which causes the illness known as covid-19) appeared in the city of Wuhan (China). This virus has rapidly spread across all the globe infecting hundreds of thousands of individuals, causing tens of thousands of deaths and making many countries all over the world to implement quarantine measures in order to prevent further spreading.

For these reasons, models able to predict the evolution of this pandemic are necessary, being the main focus of this assignment. The application of **machine learning** techniques to predict the virus' spread and in order to better understand the spreading of the virus.

Target Class: Predict a model that predicts the Covid-19's spread in the most affected countries in Europe for the next 5 days.

Objectives

The aim of this report is to provide a better understanding of the new virus that affected in a short time frame all the world. To provide a general visualization and comprehension on Covid-19 this project will have different objectives that define its development:

Analyze its spread around the globe Worldwide, Continent and Country wise.

Evaluate its main metrics such as Mortality and Recovery rates.

Create a regression model that evaluates more in depth and predicts accurately Covid-19's evolution and increase.

Besides the general introduction and evaluation of the Dataset in Worldwide terms, the analysis will get geographical screened, until focusing on **Europe's countries** due to the information transparency policies in these countries, turning the information on the Covid-19 spread provided more reliable and hence more accurate in making a prediction model.

Dataset

The dataset is stored in a single file (covid_19_data.csv) containing daily **cumulative** information on the number of affected cases, deaths and recovery of the virus by region/country from the 22nd of January to the 22nd of March (61 days)

Has a Total of 8 attributes: Sno (the serial Number), ObservationDate (the Date of the observation), Province/State (categorical), Country/region (categorical), LastUpdate, Confirmed, Deaths Recovered (The last 3 are the cumulative number of cases till that date).

To complete the information provided by this dataset, two other datasets have been merged with the original one:

- The first one, for visualization purposes, provides longitude and latitude for every country in the dataset.
- The second merged dataset includes more information that may be useful to test correlations and provide a different perspective to the results we obtain. The columns added are information regarding the Population number, the Area in square meters, the Population Density, Net migration and GDP($) per capita by country.
- The third is a preinstalled Orange Data Mining software which included the country specific information of 'Physicians (per 10,000 people) 2001-2014', 'Public health expenditure (% of GDP) 2014', 'Population Median age (years) 2015', 'Gender Development Index value', 'Life expectancy', and 'Total Population (millions) 2015'. This data set was mostly used for the synergies and results final interpretation.

Both first Datasets integration with the original one have done by merging the original one with the interested columns of the other on a left join on the countries' columns. Before the countries columns had to go thought a process of standardization and harmonization with the country_converter library in python.

```
standard_names = coco.convert(names=countr, to='name_short')
```

All datasets are linked in the appendix, named respectively: Dataset I, Dataset II and Dataset III.

Original Data Frame shape: (7926, 8)

I. Preprocessing

Data Set(s) Pre-processing

- There are no duplicates
- Null Values: 3433 in "Province/State" column. Nan values where substituted by "Unknown" in order to not loose any of the observations.

Country Names standardization:

- Convert "UK", "Channel Islands", and "North Ireland" into "United Kingdom".
- Convert "Republic of Ireland" to "Ireland"
- Eliminate rows containing "Others" in countries column (total of: 45 observations)
- Standardication of Country names with python library country_converter.

New Data Frame shape: (7881, 8)

Continent: A categorical column was added also with countr_converter function and a dictionary creation that relates the standardized country names with its continent.

New Data Frame shape: (7881, 9)

Observation Date: conversion to the **Index** of the Data frame

Latitude & Longitude: Two new rows including Country wise information regarding it's geographical position where included with the merge with the *Dataset I* in the Appendix. The information on the countries of St. Barths, Saint-Martin and Curacao had to be introduced later as there was no information regarding these (which we're in the original dataset, and else there was missing values)

Delete Sno, Last Update, and Province/State due to the scope of our research.

New Data Frame shape: (7881, 8)

Active Cases: The active cases where computed from the already existing rows:

```
df['Active Cases'] = df['Confirmed'] - df['Deaths'] -
df['Recovered']
```

Country specific information: Population, Area(sq.m), Pop Density (per sq.m.) Net migration, GDP($ per capita) where included through the merging with Dataset II in the appendix.

New Data Frame shape: (7881, 13)

Final Csv's

Furthermore, three final datasets have been created to be tested: the **Completed,** the Completed **– China's** observations, the Completed for only the countries belonging to **Europe**.

- Final Csv with no changes
- Final Csv without China's observations

```
corona_pred_china = corona_pred.loc[corona_pred['Country'] !=
"China"]
```

- Final Csv only with European Countries observations

```
europe_pred = corona_pred[corona_pred['Country'].isin(europe)]
```

The **final Csv** has **as columns**: Country Name, Continent Name, Confirmed, Deaths Recovered, and Active Cases. Other country level information: Population, Area, Population Density, Net Migration, GDP, Longitude and Latitude. And the Observation Date as the index.

Algorithm ready Pre-processing

Target Variables for the prediction: Confirmed Cases

As shown in the image in the appendix: Overview of the Workflow of ML. The Data set needs to be further analyzed and prepared. Through this Preprocessing the three Datasets mentioned above have been tested.

First steps:

- ObservationDate column as datetime
- Population density and Net Migration columns as a float
- Sorting the Datased: based on the Country names and ObservationDate.

In order to reduce the computational effort by the model, analyzing the correlation of the targeted feature with from all the 12 other features in order to select the ones to which is more dependent on.

	Confirmed	Deaths	Recovered	Active Cases
Confirmed	1	0.89	0.76	0.99
Deaths	0.89	1	0.8	0.84
Recovered	0.76	0.8	1	0.68
Active Cases	0.99	0.84	0.68	1

As it can be seen in the Graph **Confirmed**, **Deaths**, **Recovered** and **Active Cases** are **highly correlated between them**. What means that using all of them in the algorithm training will lead to noise creation. This made us add **new features** to the dataset:

a) ***days_since_first:*** Since datetime columns should be converted to numerical, we added days_since_first column to the dataset which represents the days since the first occurance of COVID-19 for each country.

```
for i in df.index[1:]:
    if df['Country'][i]==df['Country'][i-1]:
        df.loc[i,'days_since_first']=(df.at[i-
1,'days_since_first']+1)
        df.loc[i,'previous_Confirmed']=df.at[i-1,'Confirmed']
    else:
        df.loc[i,'days_since_first'] = 0
        df.loc[i,'previous_Confirmed'] = 0
```

b) ***previous_"Target_variable":*** This is one of the important feature which would help the model to learn and predict the Target values by taking into account the previous value.

c) ***increase_rate***: has been added to represent the rate at which "Target_variable" cases are increasing. Below is the logic used

```
covid_df['increase_rate']=((previous_Target_variable -
Target_variable) /previous_Target_variable)
```

	Confirmed	Population	Area (sq. mi.)	Pop. Density (per sq. mi.)	Net migration	days_since_first	previous_Confirmed	increase_rate
Confirmed	1	0.029	0.013	-0.033	0.011	0.28	1	-0.017
Population	0.029	1	0.3	-0.069	-0.055	0.15	0.025	-0.024
Area (sq. mi.)	0.013	0.3	1	-0.1	0.077	0.16	0.0073	-0.029
Pop. Density (per sq. mi.)	-0.033	-0.069	-0.1	1	0.29	0.14	-0.032	-0.046
Net migration	0.011	-0.055	0.077	0.29	1	0.13	0.0087	-0.0056
days_since_first	0.28	0.15	0.16	0.14	0.13	1	0.27	-0.076
previous_Confirmed	1	0.025	0.0073	-0.032	0.0087	0.27	1	-0.021
increase_rate	-0.017	-0.024	-0.029	-0.046	-0.0056	-0.076	-0.021	1

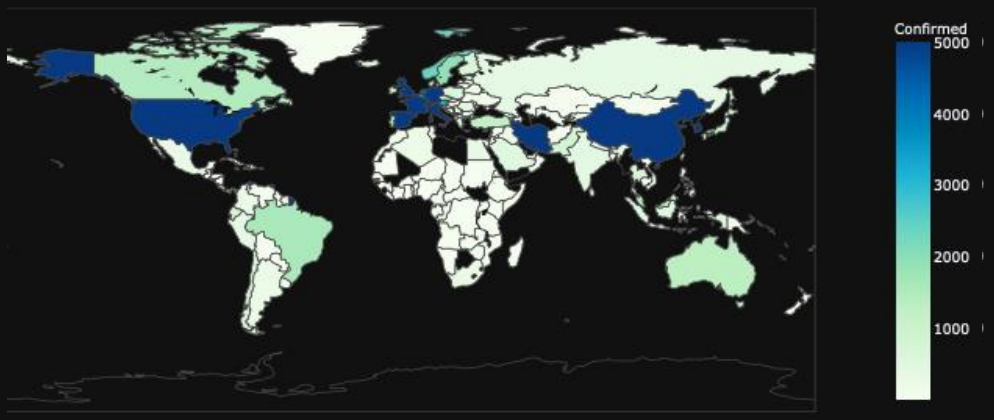
Moreover, with the inclusion of these variables one of the main drawbacks of the regression models for **timeseries forecasting** has been solved by including time variables.

From this new correlation graph, it has been decided to train & test the algorithm based on previous_Target_feature column and drop the other mentioned above.

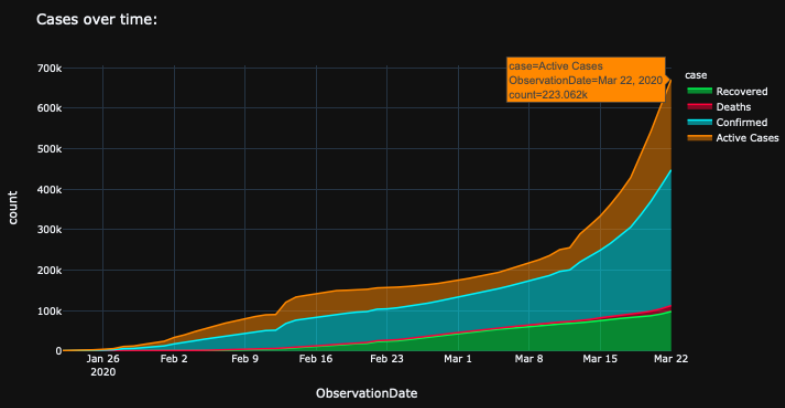
```
Y = df["Target_variable"]

X = df["Population", "Area", "Pop.Density", "Net migration",
"days_since_first", "previous_Target_variable"]
```

II. Worldwide schema

General Overview

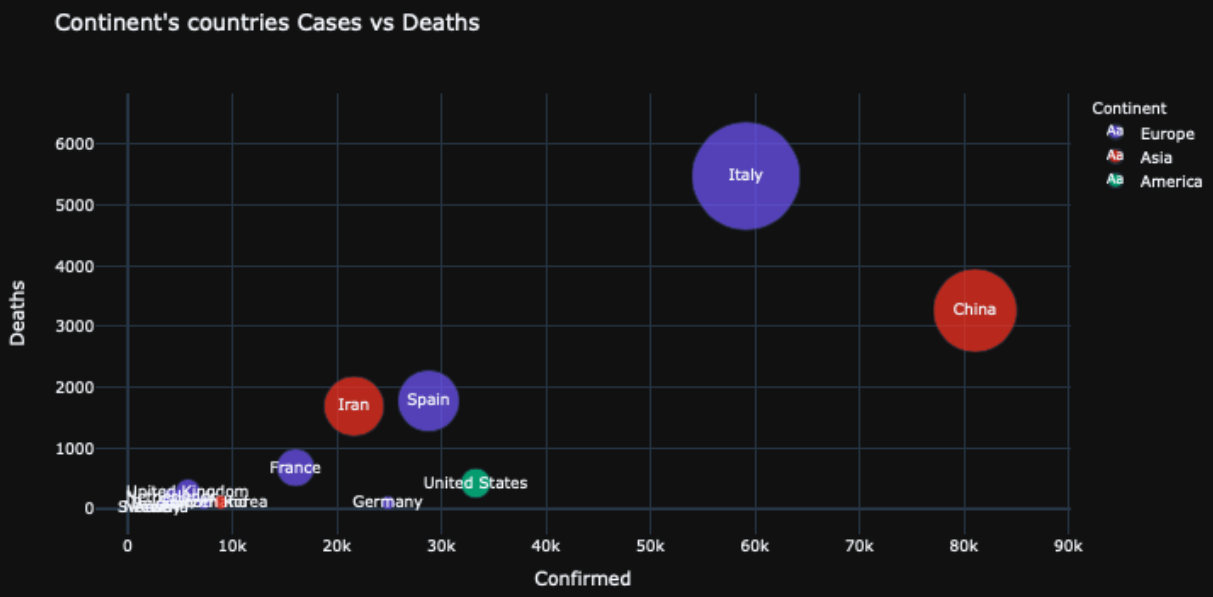


By the 22nd of March of 2020 the sum of the Confirmed cases of Covid-19 where of 335.245 people, affecting Europe, Asia and North America the most. Of those 223.062 where still considered as Active Cases. Furthermore, by that date 14.642 people where reported as dead due to the virus infection.



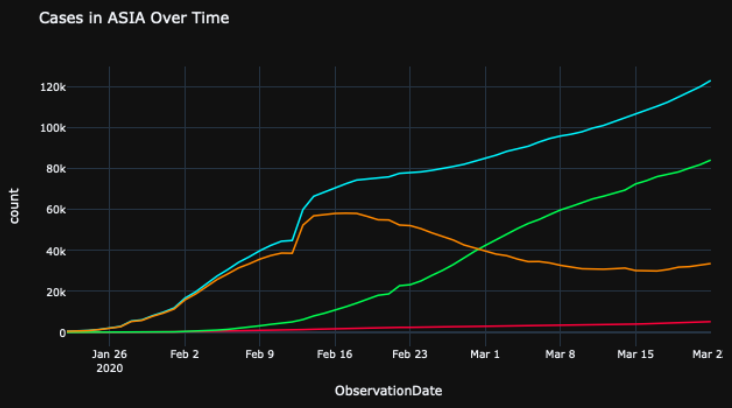
Analysis by Continent

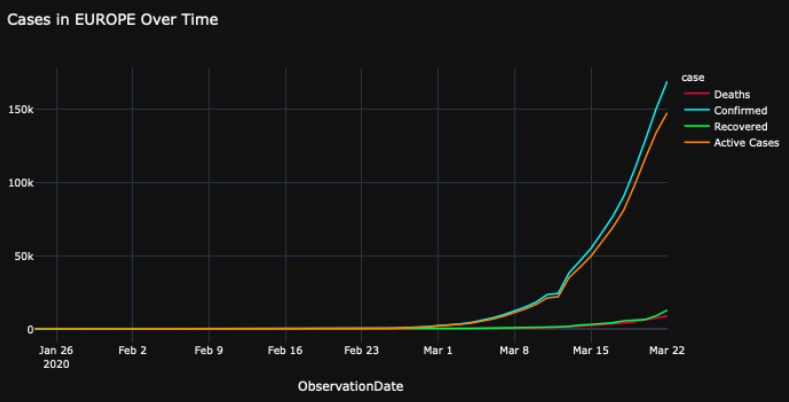
The country with the highest Confirmed, Deaths and Active cases is Europe with 169.185, 8.807 and 127.89 respectively, followed by Asia with 123.234 Confirmed Cases. How is it possible if the first case of Covid-19 was registered in China?



Comparing, on the 22nd of March of 2020, the Confirmed Cases and Deaths reported by the most affected countries world wide, a growth trend can be observed, for which China may seem an outlier (with much less deaths theoretically reported). This information provides us some mistrust. Therefore one of the Datasets that is analyzed in in which China is taken as an outlier and is no longer used to predict the future behavior of the spread of Covid-19.

When analyzing the trends for the 4 targeted variables separately for each continent, supporting the conclusions made before, Asia's behavior to the others':



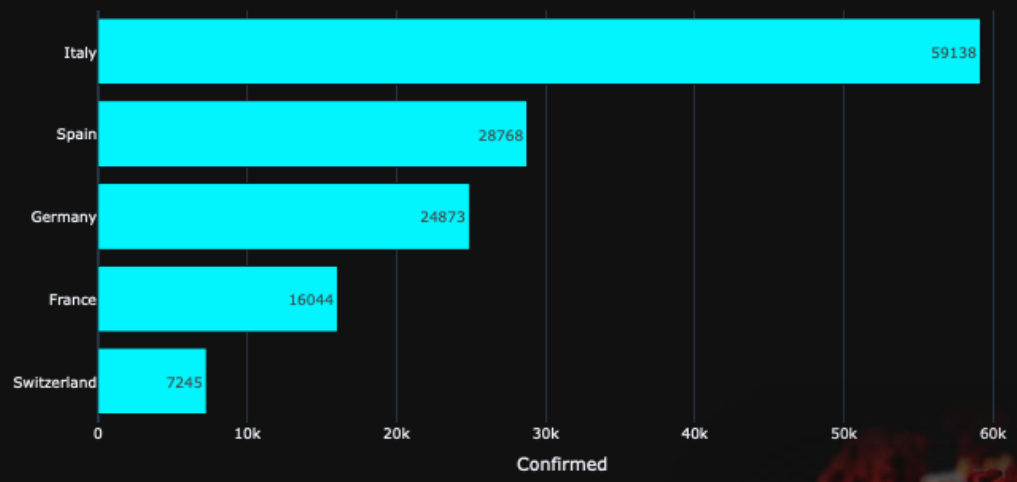


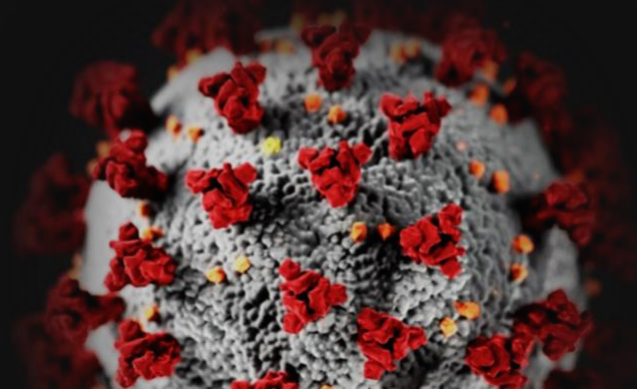
III. Europe Outbreak

Contextualization

Europe has been chosen to be the target of the further analysis due to two main reasons: information transparency and being the second continent where Covid-19 spread the soonest.

Therefore the Target of prediction of the prediction algorithm will be based on the 5 European countries with more cases:





IV. Models

Mentioning again the picture named as *Overview of the Workflow of ML* in the Appendix, once the Data has been pre-processed (by **solving correlation synergies between the variables and time features have been included to solve Regressions model's time series forecasting limitations**) and X and Y define it's time for the next: Apply an algorithm to train and test subsets in order to train the algorithm. But which algorithm? We have tested three different: SVR, Random Forest and XGB. For which the results will be shown at the end of this section of the Report.

Train and Test Split

```
Y = df["Target_variable"]

X = df["Population", "Area", "Pop.Density", "Net migration",
"days_since_first", "previous_Target_variable"]
```

The break that worked the better when testing the data set was splitting the data by **15% test** and **85% train.**

Models Definition

Why these algorithms? Because although time variables have been included into the analysis many of them had to be through a first test and evaluated in terms of performance with its Mean Squared Error, in order to understand which adapts the best to the objectives.

Process of selection

Models tested: SVR, Neural Network, Decision Tree, KNN, Random Forest and XGBoost Regressors with python's **Scikit-learn** software. **Random Forest** and **XGBoost** outperformed the others and the clearly worst performing models where Decision Tree ad KNN. In order to compare the best performing results and reassure it's validity, they where again tested with **Orange**. With Orange the model tested where: SVM, Random Forest, Neural Network, and Linear Regression (with 4 different flavors: no regularization, Lasso, Ridge, and Elastic Net).

Comparing Orange's and Scikit-learn results the models that where further developed where Random Forest and XGBoost. Why?

- **XGBoost**, as shown later in the report, had performed remarkably well on python.
- **Random Forest** even though RMSE when compared to the other models is high in Orange, when comparing it's results with the real occurrence Random Forest is clearly better predicting reality.

Linear Regression	SVM	Random Forest	Neural Network	Confirmed
11.9	-1.0	4.7	1.8	4.0
-28.0	-36.9	4.1	-86.7	4.0
-23.6	-32.9	5.1	-14.2	5.0
-19.5	-29.2	6.5	26.3	7.0
-22.4	-31.6	7.0	-71.6	7.0
-21.6	-31.0	7.0	-66.6	7.0

Model	MSE	RMSE	MAE	R2
SVM	47817.500	218.672	49.370	0.993
Random Forest	191215.385	437.282	43.576	0.974
Neural Network	3406226.741	1845.597	495.437	0.532
Linear Regression	46660.309	216.010	51.746	0.994
Linear Regression	46660.308	216.010	51.746	0.994
Linear Regression	46653.172	215.993	51.447	0.994
Linear Regression	46648.625	215.983	51.576	0.994

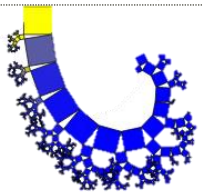
Best Parameters selection

To find the **best parameters** that define the algorithm the hyper-parameters where passed as arguments in GridSearchCV .

Random Forest Regressor

```
RandomForestRegressor('criterion': 'mse', 'min_samples_leaf': 1,
'n_estimators': 400, 'random_state': 1, cv=3)
```

```
MAE train:  12.55647376739714
MSE train:  12739.658144235762
RMSE train:  112.8700941092713
r2:  0.9992367123517645
MAE test:  14.575479161227033
MSE test:  8219.474125974813
RMSE test:  90.6613154877802
r2:  0.99751466039578
```



XGBoost Regressor

```
xgb_reg = xgb.XGBRegressor(n_estimators=100,  learning_rate=0.08,
gamma=0, subsample=0.8,colsample_bytree=1, max_depth=10)
```

```
MAE train:  2.3677058369516675
MSE train:  1361.1393502775222
RMSE train:  36.893622081296414
r2:  0.9999124978941143
MAE test:  11.275512963116586
MSE test:  6874.03235006801
RMSE test:  82.90978440490609
r2:  0.9979219081186896
```

Selection

Dataset Selected

As the number of opinions regarding the low reliability in China's reported numbers the Dataset which included this country's information was early excluded. And when testing in the algorithm the Dataset with only observations occurred within the European countries the model was clearly **underfitted**. As it can be seen in the next image, due to the low number of observations the model becomes under-trained causing the high-value of MSE. *(these two images show the results of the metrics obtained for XGBoost and Random Forest regressor respectively)*

```
MAE train:  40.457025266904175
MSE train:  116785.41423783088
RMSE train:  341.7388099672481
r2:  0.9970799518856828
MAE test:  31.771247458205654
MSE test:  50475.860949497495
RMSE test:  224.66833544026068
r2:  0.9970475167235154
```

```
MAE train:  29.492991213683233
MSE train:  32636.043096886955
RMSE train:  180.65448540483837
r2:  0.9990919404634143
MAE test:  22.009361189846295
MSE test:  12570.100385273874
RMSE test:  112.11645902932305
r2:  0.9992655531505863
```

The rest of the report proceeds taking into account all the observations but China's. So the named before: **Final Csv with China as an outlier**.

Algorithms testing (XGB & Random Forest)

The results shown in the Model's section are good? But will it work when predicting correctly the next 5 days evolution? This could be tested by taking information of the Kaggle's dataset (appendix Dataset IV). From which a **Data frame** with the **predicted infections** and **actual infections** that

occurred for the **22nd till the 26th of March 2020** was predicted for the countries of Italy, Spain, Germany, France and Switzerland.

The process is fully recorded in the GitHub link reported at the beginning of the introduction and in the appendix.

V. Predictions

Interpretations

As it was mentioned at the beginning of the report, one of the main drawbacks of forecasting one a time series with regression machine learning models is the date-time inclusion. Which was already solved. But there is another one: we are predicting an epidemic spreading for which many more complex models which take into account it's normal evolution and other factors such as: affect to its evolution. A pandemic is very susceptible to external containment measures. So as to reflect the influence of government containment measures and understand its influence to this report's predictions, there are some specific measures taken that should be taken into account:

Italy:

March 4	Schools closed	3089
March 9	Lockdown	9172
March 22	Only essential stores	59138

Spain

Entire March	Schools closed	
March 14	Lockdown	6391
March 29	Only essential stores	80000

Germany

March 13	Schools closed	3062
March 14	Lockdown	6391
March 29	Borders were closed to 5 neighbour countries	80000

France

March 16	Schools closed & no events of more than 100 persons	6633
March 18	Lockdown	9134

Switzerland

March 13	Schools & events of more than 100 persons	1139
March 16	Shops closed only essentials stores	2353
March 20	No more than 5 persons in public spaces	4840

Results

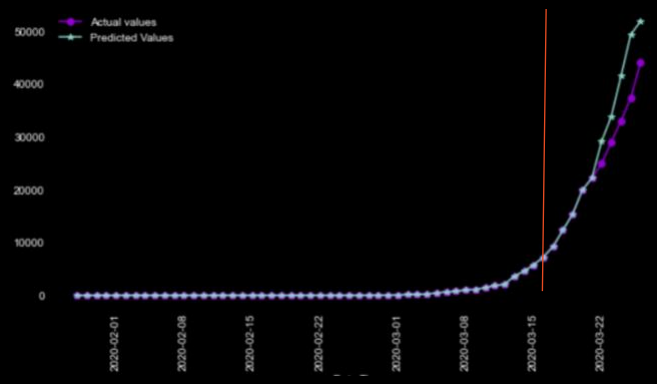
Spread Forecasting: As mentioned in the objectives of this report, besides providing some insights regarding Covid-19's evolution, the aim was to predict the next 5 days in the increase of people infected by the virus (from the 22nd till the 26th of March of 2020).

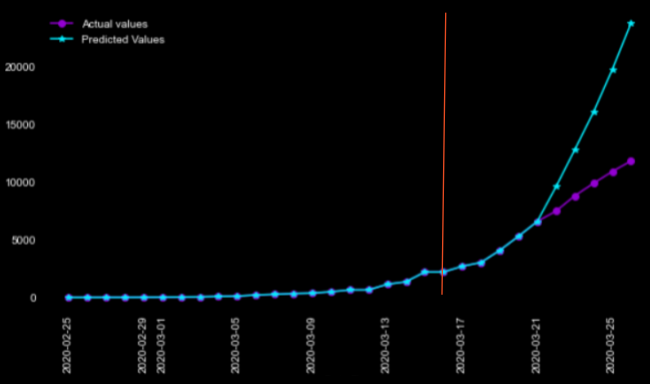
The results shown are organized by country and the plot on the left represents Random Forest and the plot on the right XGBoost results. On the X-axis are the Observed Dates from the 22nd of January till the 26th of March, and on the Y-axis the number of Confirmed Cases.

Predictions of Confirmed cases for the period 22/03/2020 until 25/03/2020

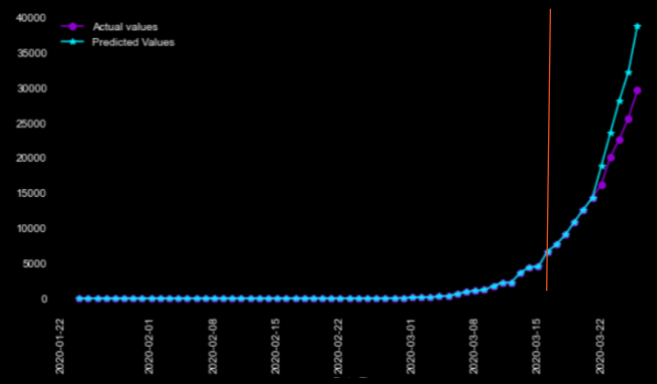
** In read you can find the day in which every government applied a lockdown or similar policies.*

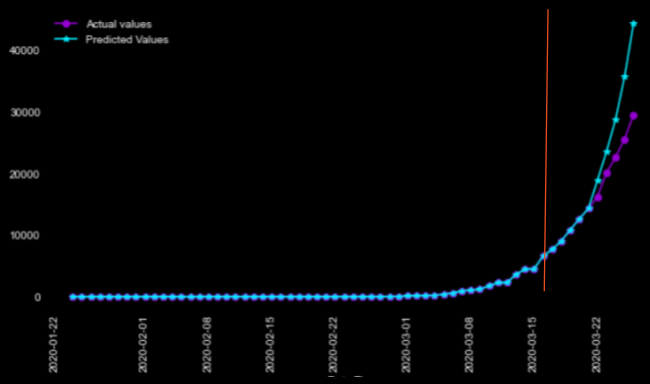
Switzerland



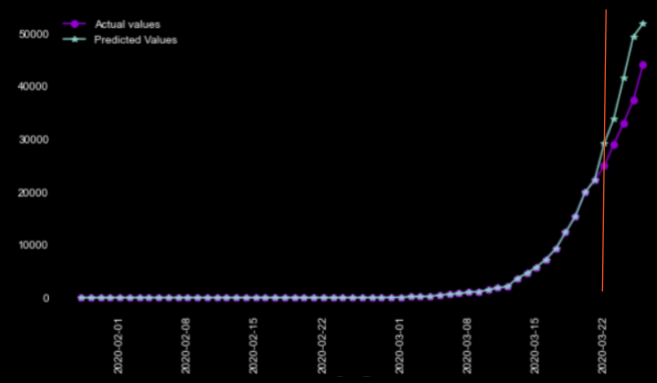


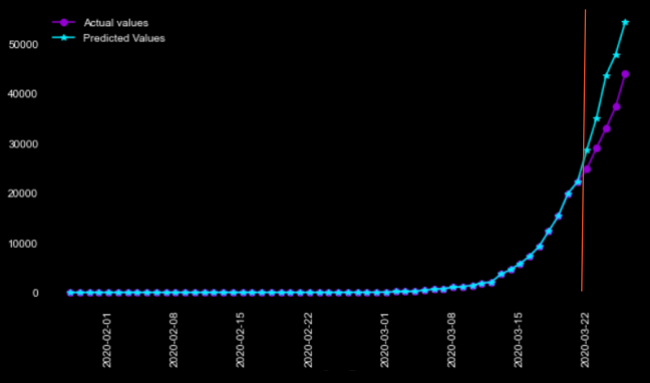
France



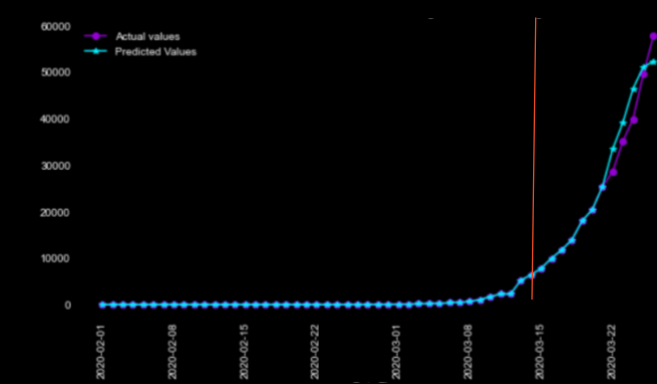


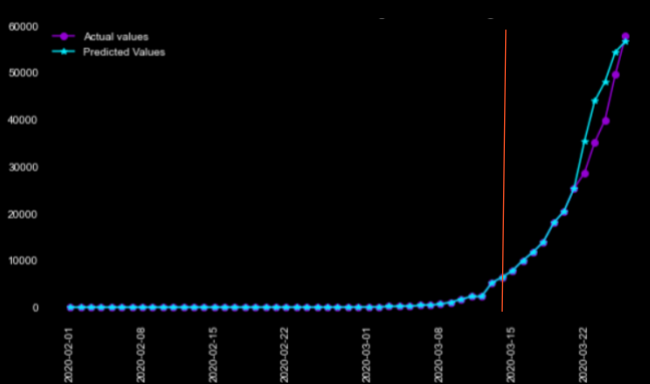
Germany



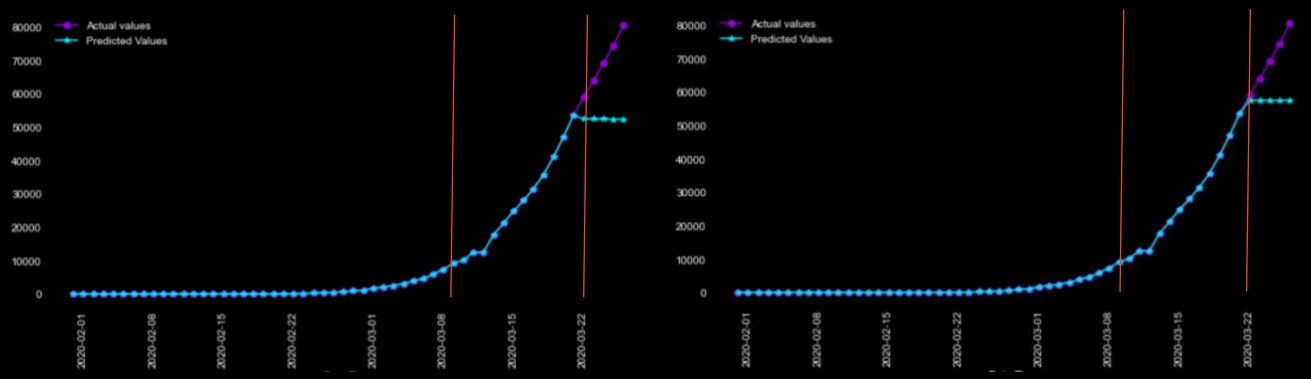


Spain





Italy

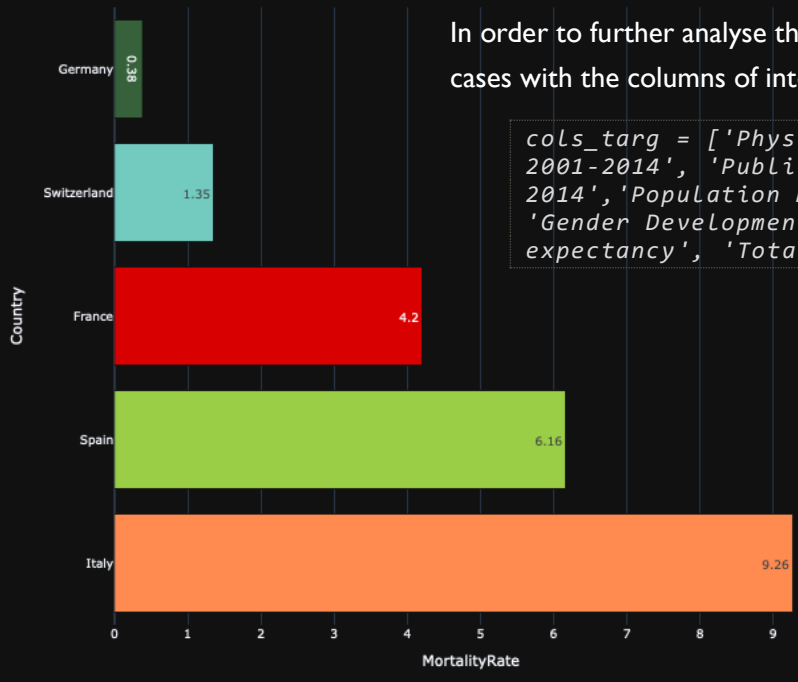


It's interesting to observe how some governments reaction, like Switzerland's, were less restrictive but applied in an earlier stage of the virus' spread. And how it's real spread is lower than the one predicted by both algorithms. Instead, for Spain which started the lockdown on the 14th of March with 6391 positive cases reported the real spread is higher than the one predicted. Here it should be important to mention that all the information provided by the governments, although European countries have been chosen due to its transparency, due to lack of material to perform test, there is a possible bias between the know new positive cases with the real ones.

Besides the direct contingency measures by every government and the fact that there is some bias in the data every government obtains there are other factors that affect its spread such as the population mean age (Dataset III).

Country Metrics

As it could be observed in the beginning of the report, Confirmed cases are highly correlated with Deaths and Recoveries, therefore also with mortality rate. This graph classify the 5 countries by it's Mortality Rates. In terms of less dead's per confirmed cases Germany and Switzerland performed the best, But what may explain that? And how does this information affect the algorithms adjustment performance?

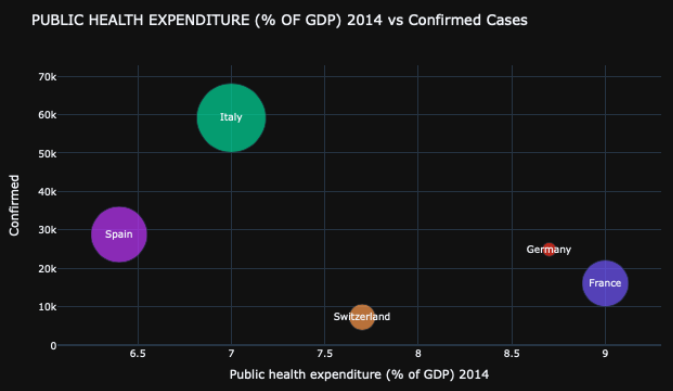


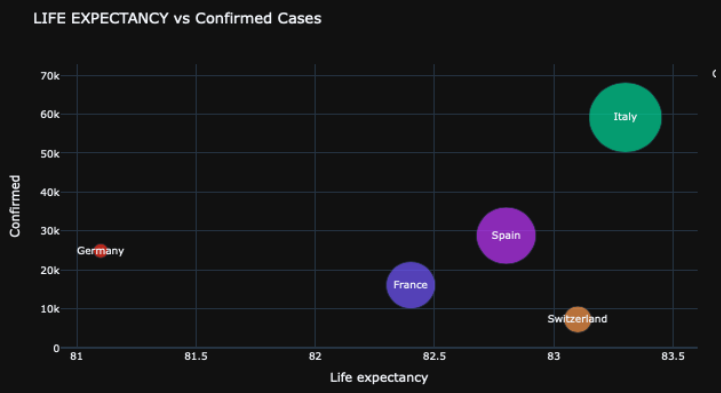
In order to further analyse this information synergies related to Confirmed cases with the columns of interest of the Dataset III:

```
cols_targ = ['Physicians  (per 10,000 people) 2001-2014', 'Public health expenditure (% of GDP) 2014','Population Median age (years) 2015', 'Gender Development Index value', 'Life expectancy', 'Total Population (millions) 2015']
```

Confirmed Cases vs the Target Variable vs Death Rate

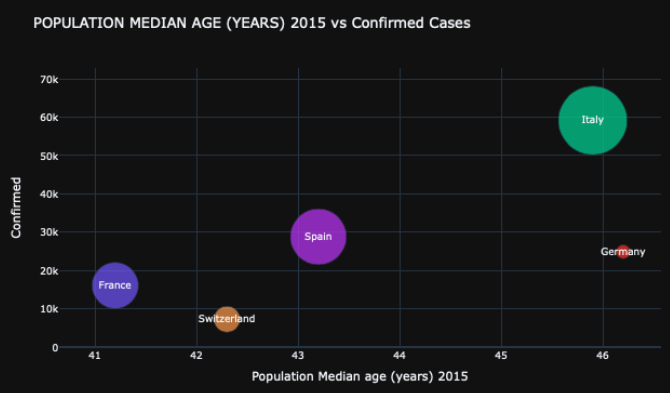
These Scatter plots analyze on the Y-axis the Confirmed cases, X-axis the Target Variable, with a size equivalent to every country's Death Rate.

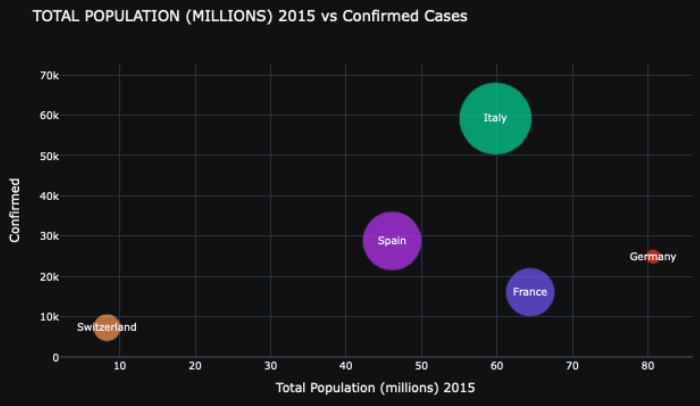




As we can see in the graph, Germany and France have invested the highest % of GDP to health sector, we can assume that lower the investment in Health sector, higher the Confirmed cases. Italy & Spain being the lowest investors in Health sector have high number of Confirmed cases.

Italy's life expectancy is comparably the best in Europe with the age>.83 years. Since, the confirmed cases are highest in Italy, older people are more vulnerable to becoming severely ill with





Italy has more population of people with age group > 45 years. Since, the number of confirmed cases are more in Italy, it can be assumed easily from the above graph that the people of age group more than 45 are infected more.

Here there is no specific correlation between Confirmed cases and Population.

As "Life Expectancy" graph says that Italy has highest life expectancy. Since life expectancy leads to an increase in population size. It is third densely populated country in Europe with highest number of confirmed cases.

Model Limitations:

Correlation doesn't mean causation, but by analyzing these plots the fact that exist country specific, and probably region specific, synergies it is more than obvious. What implies that to properly predict Covid-19's pandemic progression needs a complex model that takes into account the total population and it's different states (susceptible, infected and recovered like the SIR model), the incubation period or the virus, the measures taken against its spread, geological specific information such as pollution and population, and the capacity and resources of each area to fight against it (medical facilities) between others. Furthermore, for Covid-19 the model should also take into account the probability of the infected to die and the probability of the recovered to get infected again, and when can they transmit the virus to others.

There are many variables to take into account and mixed with the fact that there is no information previous to December 2019 about SARS-CoV-2, it's note even know which variables should be taken into account to predict it's spread and it's lethality. Also, again due to the fact that it's worldwide expansion started not long ago, and that the data in which the model was tested was of only 60 days, the model would need more to do a more accurate prediction. Also, low coordination not only among different states but also in different regions don't help to establish general trends.

Besides all these limitations we hope that the model and the datasets evaluations helped provided more insights about Covid-19's infections and how different contingency measures and metrics affect its expansion.

Appendix

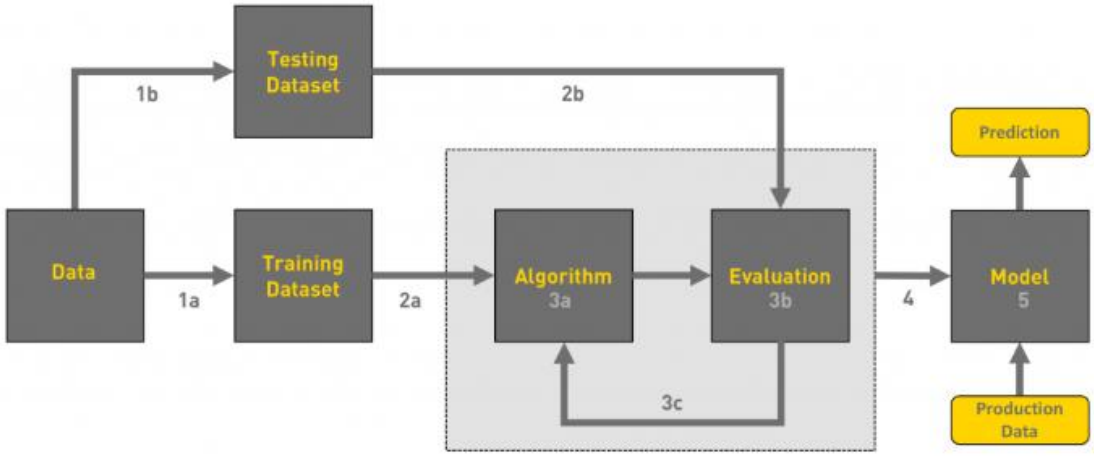
Dataset I: "world_country_and_usa_states_latitude_and_longitude_values.csv" (https://www.kaggle.com/paultimothymooney/latitude-and-longitude-for-every-country-and-state/version/1#world_country_and_usa_states_latitude_and_longitude_values.csv)

Dataset II: “countries of the world.csv” (https://www.kaggle.com/fernandol/countries-of-the-world#countries%20of%20the%20world.csv)

Dataset III: ”HDI” (Provided by Dataset feature of Orange)

Dataset IV: “covid_19_data_Kaggle.csv” (https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset#covid_19_data.csv)

Image 1: Overview of the Workflow of ML (https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94)



: