Forecasting daily COVID-19 spread in regions around the world.

Frank Lawrence Nii Adoquaye Acquaye*
Faculty of Computer Science
National Research University Higher School of Economics
fakvey@edu.hse.ru

Bense Valerie Caroline
Faculty of Computer Science
National Research University Higher School of Economics
vbense@edu.hse.ru

Abstract

1 Problem statement:

The year 2020 will forever be remembered as the year the earth stood still. This is primarily due to the spread of COVID-19. As Data Scientists we seek to provide solutions to problems facing humanity and the world at large. In this regard we seek to develop a forecasting model that will predict the daily spread of COVID-19 in regions around the world. Our model predicts the number of daily new cases in regions around the world in order to help policy makers plan and manage the COVID-19 pandemic.

2 Dataset summary and EDA:

2.1 Background of dataset:

The White House Office of Science and Technology Policy (OSTP) pulled together a coalition of research groups and companies (including Kaggle) to prepare the COVID-19 Open Research Dataset (CORD-19) to attempt to address key open scientific questions on COVID-19. Those questions are drawn from National Academies of Sciences, Engineering, and Medicine's (NASEM) and the World Health Organization (WHO).

2.2 Data sources:

The sources of data used in this project can be obtained from Kaggle Dataset. Furthermore we choose to enrich the dataset with data on the population in the different countries. Data such as population density, percentage of urban population and median age. This data was obtained from Kaggle aswell.

2.3 Actual data:

Since the accuracy of such a model is dependent on the freshness of the data, the most up to date data can be found here

*http://acquayefrank.github.io

	Id	County	Province_State	Country_Region	Population	Weight	Date	Target	TargetValue
0	1	NaN	NaN	Afghanistan	27657145	0.058359	2020-01-23	ConfirmedCases	0.0
1	2	NaN	NaN	Afghanistan	27657145	0.583587	2020-01-23	Fatalities	0.0
2	3	NaN	NaN	Afghanistan	27657145	0.058359	2020-01-24	ConfirmedCases	0.0
3	4	NaN	NaN	Afghanistan	27657145	0.583587	2020-01-24	Fatalities	0.0
4	5	NaN	NaN	Afghanistan	27657145	0.058359	2020-01-25	ConfirmedCases	0.0

Figure 1: Preview of first five line items in training dataset.

	ForecastId	County	Province_State	Country_Region	Population	Weight	Date	Target
0	1	NaN	NaN	Afghanistan	27657145	0.058359	2020-04-27	ConfirmedCases
1	2	NaN	NaN	Afghanistan	27657145	0.583587	2020-04-27	Fatalities
2	3	NaN	NaN	Afghanistan	27657145	0.058359	2020-04-28	ConfirmedCases
3	4	NaN	NaN	Afghanistan	27657145	0.583587	2020-04-28	Fatalities
4	5	NaN	NaN	Afghanistan	27657145	0.058359	2020-04-29	ConfirmedCases

Figure 2: Preview of first five line items in test dataset.

2.4 Actual data used in project:

In this project we use frozen dataset i.e dataset that has been frozen in time and this dataset can be found here

2.5 Basic exploratory data analysis:

The dataset for training consists of 8 primary variables with a total of 914232 line items. 1.6% of the line items contained missing data. *Figure 1* shows the a preview of the first five line items in training dataset. Upon further investigations we realise that most instances had $Country = NaN$, also $Province\_State = Nan$ except in cases where the $Country\_Region = U.S.A$. For this reason we exempted these two variables or attributes. In the exploratory stage, there is no clear description of the weight parameter, but we may experiment with it, to see how it impacts predictions but most likely we may drop it.

Figure 2 shows the first five rows of the data for testing. One can easily realise that the target value was not supplied, for this reason we will drop the test dataset and split our training dataset in a manner that allows us to test our models.

We split our training data into *confirmed cases* and *fatalities*. This can be confirmed from *Figure 3*. A simple summary statistic of confirmed cases and fatalities is shown is shown in *Figure 4* and *Figure 5*.
We tried to have a fair understanding of the growth rate of confirmed cases globally by week and it seems exponential. This can be seen if *Figure 6*

In order to have a better understanding of the data, kindly follow the links below to view a detailed report of our EDA. To preview the profile of [train.csv] follow this link. To preview the profile of [test.csv] follow this link. To view a plot of fatalities vs first infections kindly follow this link

3 Methodology:

3.1 Data Cleaning:

- We discovered negative values in the TargetValue field, for this reason we computed the absolute value for all TargetValues

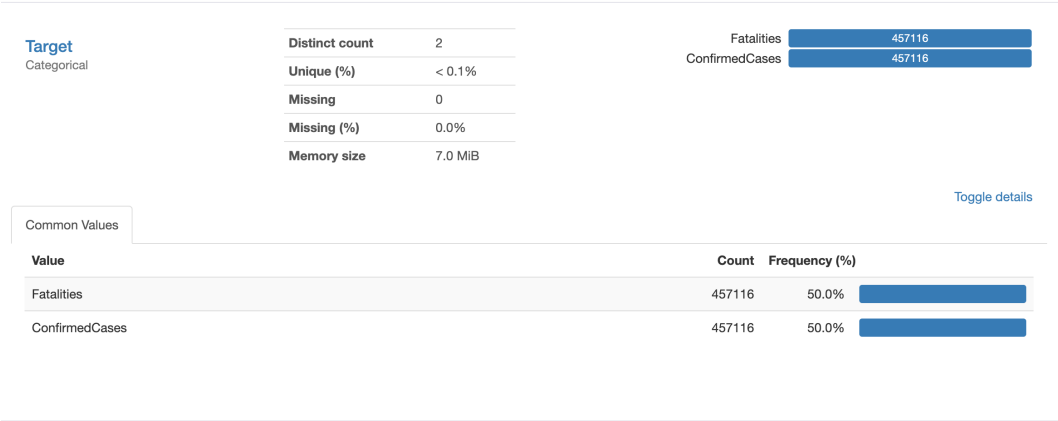

Figure 3: Instances of confirmed cases and instances of fatalities.

	Population	Weight	TargetValue
count	4.571160e+05	457116.000000	457116.000000
mean	2.720127e+06	0.965218	1.319908
std	3.477773e+07	0.175551	28.767670
min	8.600000e+01	0.474908	-1918.000000
25%	1.213300e+04	0.864488	0.000000
50%	3.053100e+04	0.968379	0.000000
75%	1.056120e+05	1.063404	0.000000
max	1.395773e+09	2.239186	4591.000000

Figure 4: Summary statistics of fatalities.

	Population	Weight	TargetValue
count	4.571160e+05	457116.000000	457116.000000
mean	2.720127e+06	0.096522	22.328864
std	3.477773e+07	0.017555	407.011027
min	8.600000e+01	0.047491	-10034.000000
25%	1.213300e+04	0.086449	0.000000
50%	3.053100e+04	0.096838	0.000000
75%	1.056120e+05	0.106340	0.000000
max	1.395773e+09	0.223919	36163.000000

Figure 5: Summary statistics of confirmed cases.

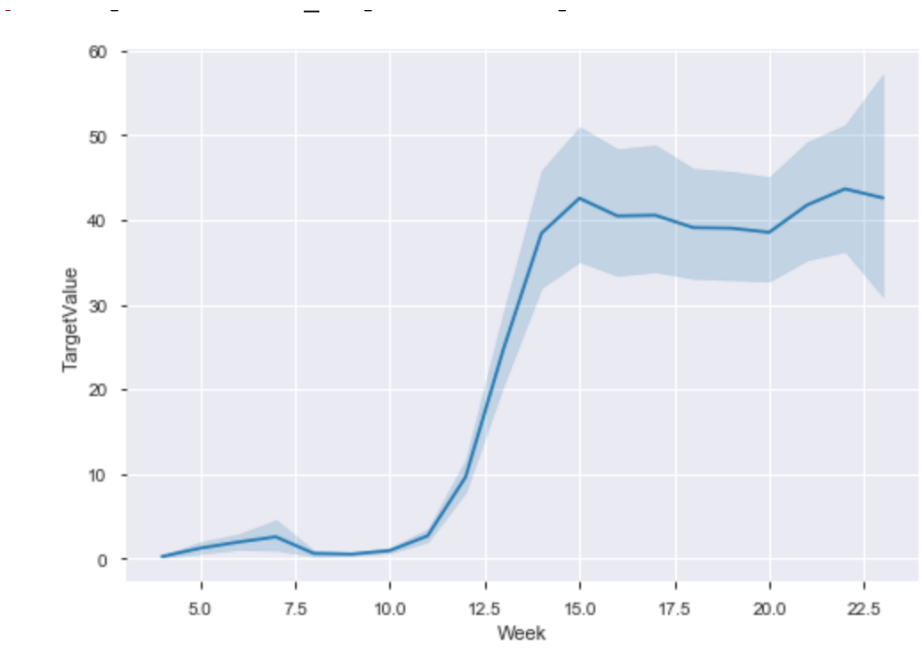

Figure 6: Number of confirmed cases by week.

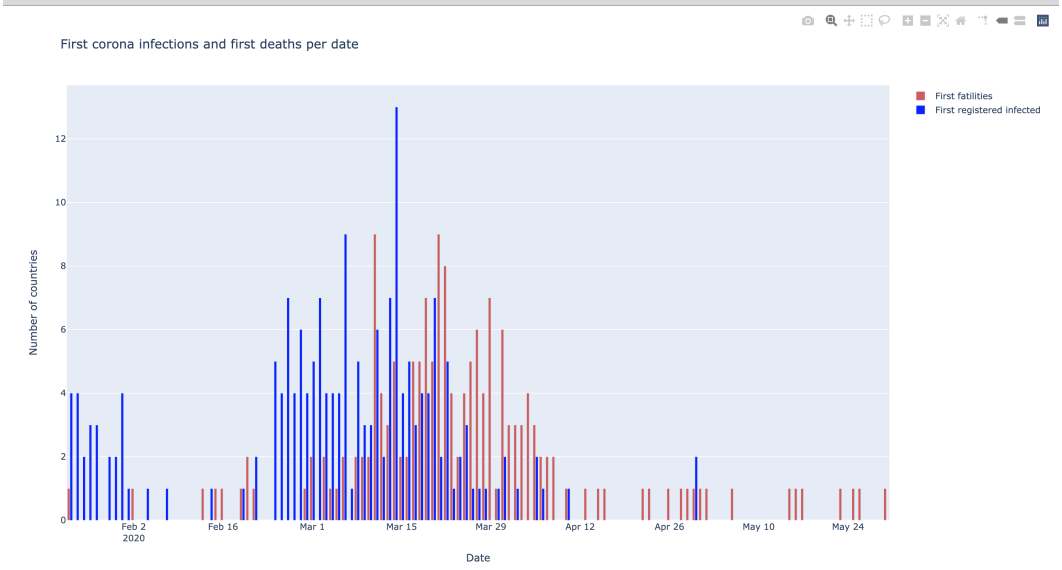

Figure 7: A distribution of first infections vs first deaths.

- We dropped all rows that had their Province_State filled out. This was primarily due to the fact that this field was mostly set for a limited number of countries, such as US, UK and $China$. This made the data skewed more towards American, United Kingdom and Chinese regions.

3.2 Data Enrichment:

We obtained population by country data and selected some fields to enrich our data. We added the:

- population density
- population median age
- urban population percentage

3.3 Feature Engineering:

To create a sufficient model it is of course important to give the model the best features to work with. As stated in a previous section, data on $Populationdensity, UrbanpopulationandMedianage$ was added. Also it was chosen to treat the time-series data $Date$ as more numerical values. From the date the number of the week, the number of the day, the day in the week and the number of days since the first confirmed case(s) were derived. From Permutation importance and impurity based feature importance, stemming from $RandomForestRegressor$, it was found that these alterations indeed resulted in important features. In *figure 11* and *figure 12* we can find that the Permutation difference for the different datasets are fairly similar, although the feature for Population Density seems to be slightly more important in the fatalities dataset than in the confirmed cases dataset. The feature importances that were derived from the same $RandomForestRegressor$ look similar, however give less of an importance to the Country being the US. The plots for the feature importances can be found in *figure 9* and *figure 10*.

- We extracted the week from the date
- We extracted the day from the date
- We extracted the weekday from the date
- We computed the date since the first infection was recorded in a country
- We split our data into recorded fatalities and number of confirmed cases by day

Country	Population	Weight	ConfirmedCases	Fatalities
US	648286272	6.765592	78422.000000	130185.000000
Brazil	206135893	0.052236	33274.000000	1262.000000
Russia	146599183	0.053182	11656.000000	232.000000
India	1295210000	0.047660	8821.000000	269.000000
Peru	31488700	0.057920	8805.000000	195.000000
United Kingdom	65600525	1.048416	6152.000000	7623.000000
Canada	75700840	0.947336	5516.000000	2795.000000
Chile	18191900	0.059821	5470.000000	75.000000
Pakistan	194125062	0.052400	3938.000000	88.000000
Mexico	122273473	0.053701	3891.000000	501.000000
France	69403837	0.924681	3833.000000	6369.000000
Iran	79369900	0.054976	3117.000000	81.000000
Spain	46438422	0.056646	3086.000000	688.000000
Bangladesh	161006790	0.052919	2911.000000	40.000000
Saudi Arabia	32248200	0.057840	2840.000000	24.000000
Qatar	2587564	0.067722	2355.000000	3.000000
Ecuador	16545799	0.060163	2343.000000	410.000000
Turkey	78741053	0.055000	2253.000000	78.000000
Colombia	48759958	0.056489	2165.000000	51.000000
Italy	60665551	0.055801	1900.000000	474.000000

Figure 8: Most inpacted countries.

- We then did a one hot encoding for all countries listed in our data
- We dropped the Id, County and Province_State columns since we realised it negatively impacted predictions as depicted in *Figure 9*. The feature Unnamed is the Id field

4 Experiment setup and results; error analysis:

Our initial approach was to experiment with trees and obtain feature importance, then create regression models with these features. But during our experiments we realised trees performed better. More importantly ensemble learning methods performed better.

Since the official test dataset provided by Kaggle had no labels, we discarded that dataset and split our training data into test and training. We split the data based on date. All dates after '2020-05-20' were placed in our test data and dates after '2020-05-20' were placed in training data.

As can be seen from the table, we have experimented with several models, to find that Tree-based ensemble methods perform best on this dataset. Therefore it was chosen to optimize those models.

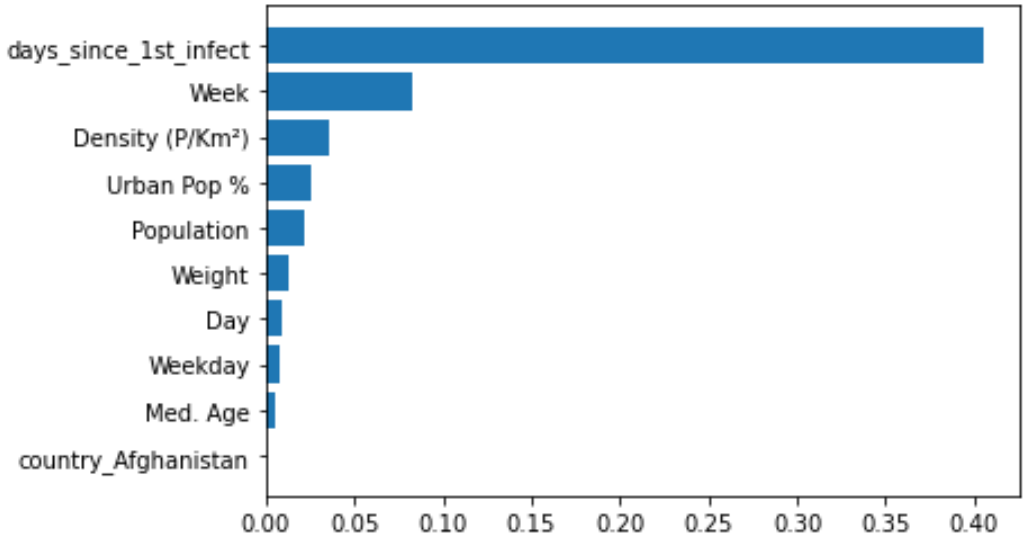

Figure 9: Feature Importance for confirmed cases dataset after some experimental models were created (Random Forest).

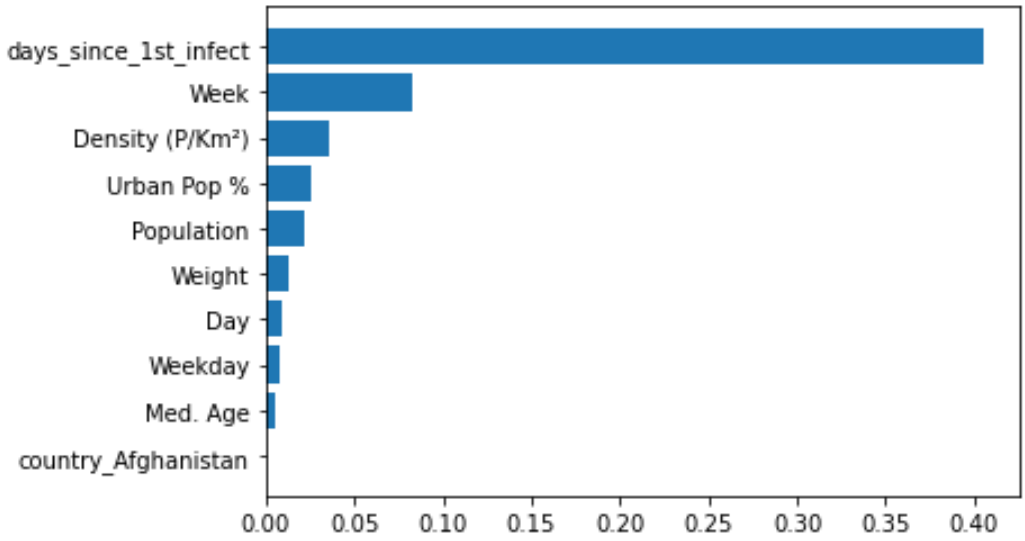

Figure 10: Feature Importance for fatalities dataset after some experimental models were created (Random Forest).

Weight	Feature
1.4787 ± 0.1773	country_United States
0.8355 ± 0.1090	days_since_1st_infect
0.1960 ± 0.0152	Week
0.1714 ± 0.0221	Density (P/Km²)
0.1017 ± 0.0091	Weight
0.0661 ± 0.0132	Urban Pop %
0.0575 ± 0.0060	Population
0.0144 ± 0.0038	Day
0.0127 ± 0.0031	Weekday
0.0107 ± 0.0015	country_Spain
0.0092 ± 0.0014	country_Peru
0.0075 ± 0.0015	Med. Age
0.0042 ± 0.0003	country_United Kingdom
0.0035 ± 0.0004	country_China
0.0034 ± 0.0003	country_India
0.0029 ± 0.0007	country_Ecuador
0.0026 ± 0.0005	country_Saudi Arabia
0.0024 ± 0.0004	country_Canada
0.0022 ± 0.0006	country_Chile
0.0016 ± 0.0001	country_Turkey
… 174 more …	

Figure 11: Permutation importance for confirmed cases dataset.

Weight	Feature
1.5088 ± 0.1577	country_United States
0.9922 ± 0.1476	days_since_1st_infect
0.1323 ± 0.0200	Density (P/Km²)
0.1092 ± 0.0099	Week
0.0887 ± 0.0072	Weight
0.0676 ± 0.0166	Urban Pop %
0.0650 ± 0.0038	Population
0.0189 ± 0.0055	Day
0.0185 ± 0.0027	country_Spain
0.0150 ± 0.0033	Weekday
0.0090 ± 0.0014	country_Peru
0.0039 ± 0.0013	country_Ecuador
0.0037 ± 0.0002	country_United Kingdom
0.0029 ± 0.0005	Med. Age
0.0029 ± 0.0003	country_Turkey
0.0028 ± 0.0008	country_Saudi Arabia
0.0023 ± 0.0002	country_Canada
0.0021 ± 0.0004	country_Chile
0.0021 ± 0.0002	country_India
0.0017 ± 0.0002	country_Italy
… 174 more …	

Figure 12: Permutation importance for fatalities dataset.

Model Analysis		
Model	R2 Score for Confirmed Cases	Comments
Ridge Regression	-404026.412	Performs badly
ElasticNet	0.079	Relatively better than Ridge Regression
ExtraTreesClassifier	0.217	Relatively better than Ridge Regression
Linear Regression	0.475	Relatively better than Ridge Regression
SGDRegressor	0.518	Relatively better than Linear Regression
GradientBoostingRegressor	0.7979	One of our top 4 models
HistGradientBoostingRegressor	0.862	One of our top 4 models
RandomForestRegressor	0.892	One of our top 4 models
DecisionTreeRegressor	0.899	One of our top 4 models

5 Discussion:

In this section the different models and their results, before and after hyperparameter optimization will be discussed. The models we have chosen to optimize were the models that in our initial test performed best. Interestingly, the four best performing models are all, but one, Tree-based Ensemble

models. The Decision Tree Regressor is not an ensemble model, but it is a tree-based model. The ensemble technique combines predictions from several machine learning algorithms, in order to make more accurate predictions. The first two ensemble models we will discuss are Boosting models, which entails that each model that is build will learn which features to focus on from the previous model.

5.1 Gradient Boosting Regressor:

Gradient Boosting is a method in which a prediction model is constructed as an ensemble of prediction models, often decision trees. In the sklearn implementation of this model a regression tree is fitted on a negative gradient, and aims to minimize the given loss function. This model performed decently on the default hyperparameters, achieving a $r2score$ of 0.7979 for both datasets. To optimize this model $RandomSearch$ was used, from different trials it came forth that best results come when leaving a large part of the hyperparameters on their default setting, but tuning the $learningrate$, $maxdepth$ and the $minimumsamplesperleaf$. The model achieved a $r2score$ on the confirmed cases dataset of 0.8469 and on the fatality dataset of 0.8429.

5.2 Hist Gradient Boosting Regressor:

The Hist Gradient Boosting Regressor is quite similar in its workings to the previously discussed model, it is however optimized for big datasets, where the number of datapoints exceed 10.000. It outperforms the normal gradient boosting regressor by 0.862 $r2score$. This might be explained by the way this estimator splits and handles the data (as histograms instead of ordered features). After optimization of the hyperparameters the $r2score$ is ...

5.3 Random Forest Regressor:

The Random forest is a technique where multiple decision trees are build, the predicted score (for regression) is often the mean prediction of all the individual trees. This ensemble model employs bagging technique, in which there is no interaction between the decision trees whilst building them. Random Forests are often praised for their robustness in comparison to a decision tree. On our datasets The Random Forest Regressor performs rather well, achieving a $r2score$ of 0.892 on the confirmed cases dataset, with default parameters. After optimization of the following parameters we achieved an $r2score$ of ...

5.4 Decision Tree Regressor:

Decision Trees are build in a tree structure, it breaks down a dataset into smaller subsets, where at each node a decision must be made. In our case a decision could look like: $Week <= 12$, after this decision the tree splits in subtrees. The more explanatory power a decision has, the higher up the decision will be in the tree. The most important decision node is call the root node. The mentioned explanatory power depends on the metric used, from the optimization we derive the MSE is the best metric for this dataset. The Sklearn implementation of Decision Tree Regressor with default hyperparameters already gives rather good results with an $R2score$ of 0.899. After hyperparameter optimization with Random Search the highest $R2score$ achieved was ...

6 Conclusion:

We realized esemble learning regressors performed better with the key feature being days since first infection was recorded in a country/region.