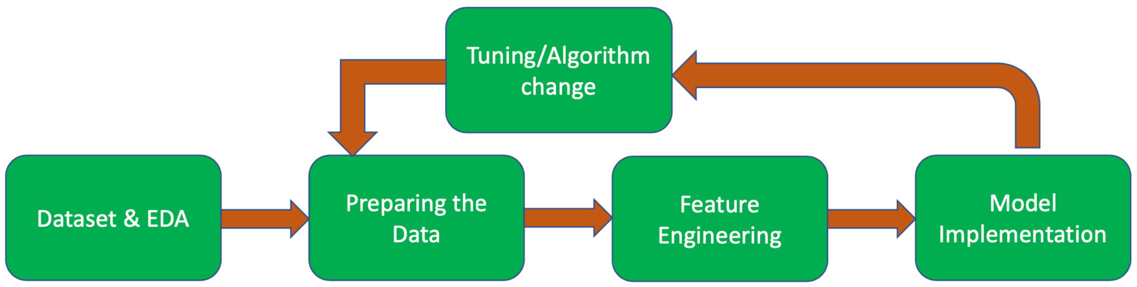
Covid - 19
Exploratory Data Analysis and Machine Learning

Tejaswee Katanguri

Summary:

Best model is time series regressor with accuracy close to ~80%(This score may vary based on dataset split).

*Important:
Here I am training the models on Covid-19 data on different countries and will try to predict how the new cases will rise in the new countries that the model has not seen before.



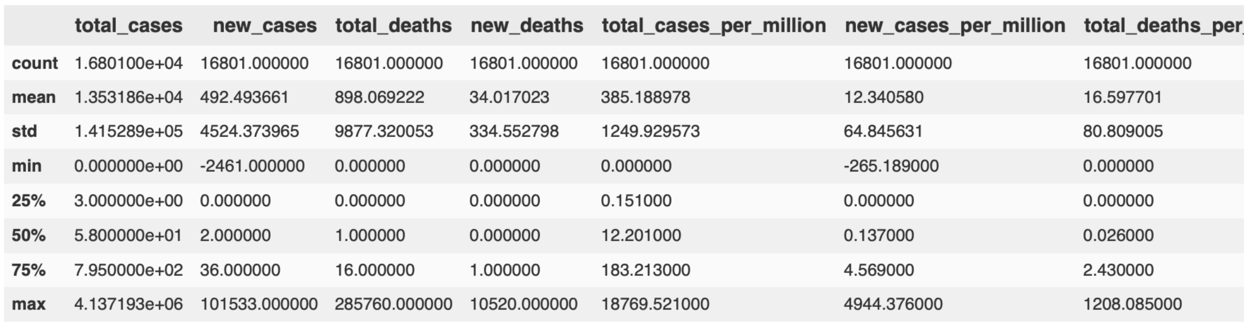
Dataset:

I used "Our world in data" Covid-19 dataset (https://github.com/owid/covid-19-data/tree/master/public/data). Dataset has universal information on new cases on per-country basis, in total >190 countries information is included. Along with Covid information, dataset comes along with social and economic information of the region providing interesting opportunities in modeling.

Output of df.head() and df.describe() in notebook will provide more information.

Total samples: 16801
Total number of columns/features: 27

Below is just sample of what describe output looks like.



	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million	total_deaths_per
count	1.680100e+04	16801.000000	16801.000000	16801.000000	16801.000000	16801.000000	16801.000000
mean	1.353186e+04	492.493661	898.069222	34.017023	385.188978	12.340580	16.597701
std	1.415289e+05	4524.373965	9877.320053	334.552798	1249.929573	64.845631	80.809005
min	0.000000e+00	-2461.000000	0.000000	0.000000	0.000000	-265.189000	0.000000
25%	3.000000e+00	0.000000	0.000000	0.000000	0.151000	0.000000	0.000000
50%	5.800000e+01	2.000000	1.000000	0.000000	12.201000	0.137000	0.026000
75%	7.950000e+02	36.000000	16.000000	1.000000	183.213000	4.569000	2.430000
max	4.137193e+06	101533.000000	285760.000000	10520.000000	18769.521000	4944.376000	1208.085000

EDA:

Before analyzing the dataset, I had to select the feature which were either integers or floats and not strings.

Univariate Analysis:

I did a histogram analysis to see the range in which each of the features exist. Except for handful of features, most of the features has a high range around 0 causing them to mask the histograms of the other samples. This was due to the fact that data was collected from Dec 2019 when lot of countries did not begin tracking the pandemic and hence lot of samples ended up being 0, we will talk about this during feature engineering phase.

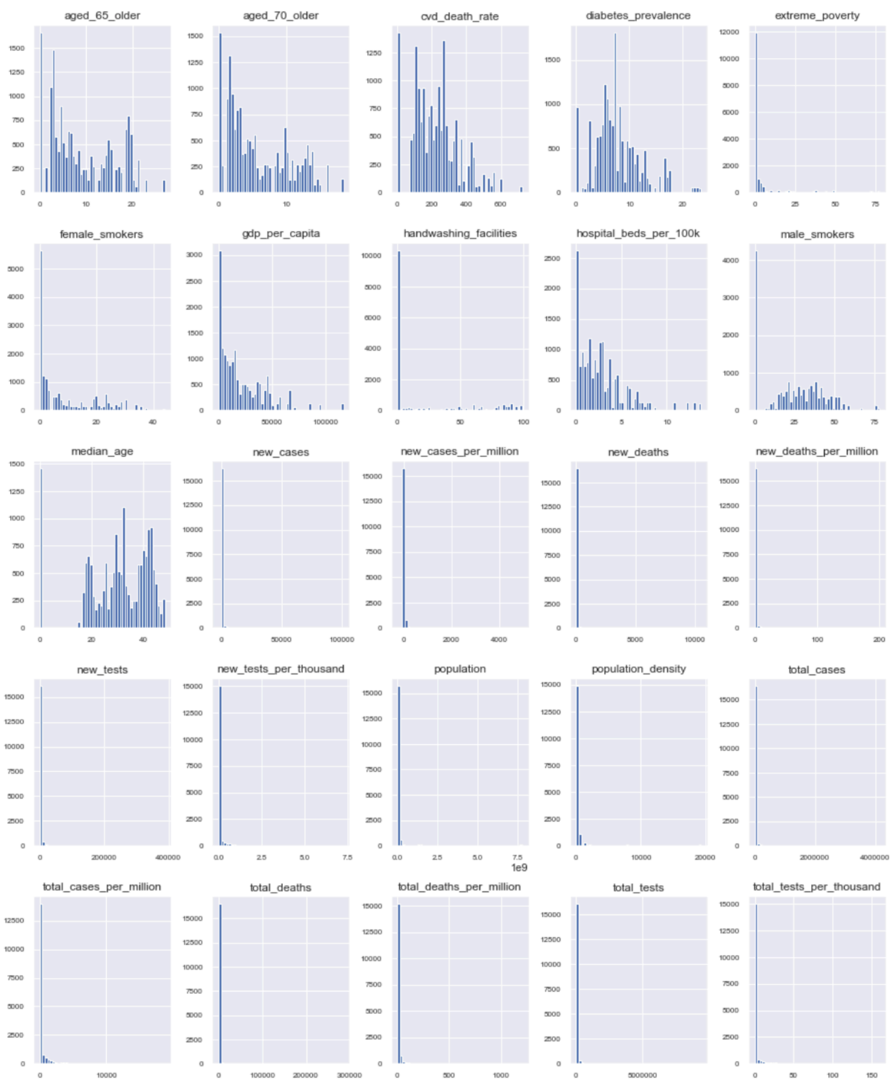


Figure 2

Tests vs Deaths:

Given the lot of media coverage, I wanted to see how increasing testing is allowing health care workers to identify more cases. I wanted to check the correlation between tests and deaths. The resulting graph and data is a perfect linear increase.

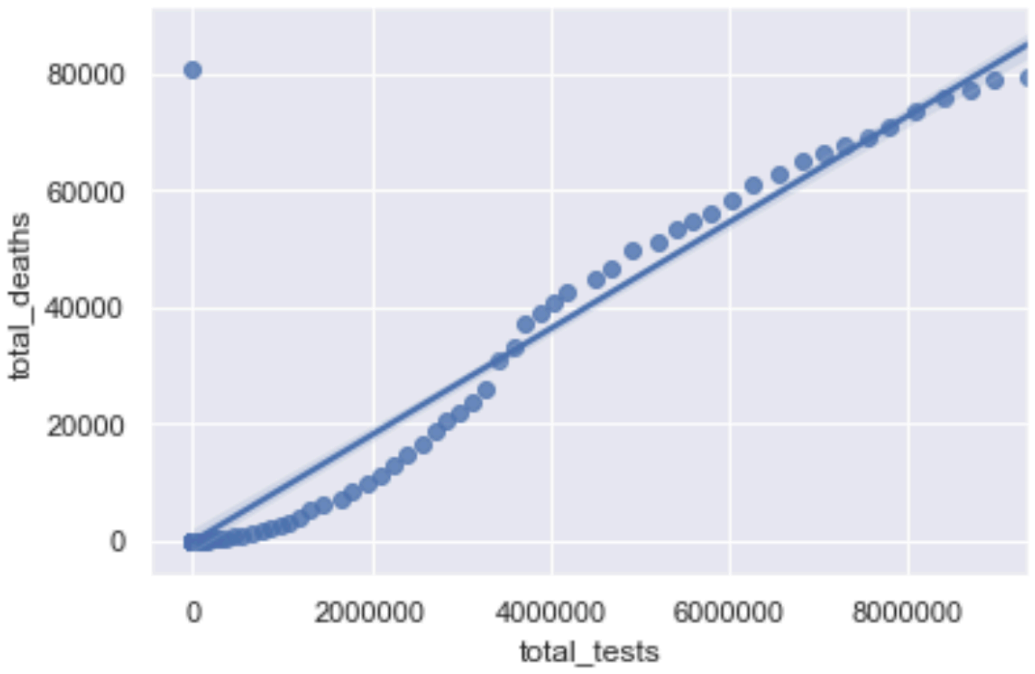


Figure 3

Before we move to trivariate analysis, I added a new feature as part of feature engineering "Days since the first case", code for this is in the notebook. This will track the days since the first case was seen and deleted the days before the first case cleaning up the dataset further. Further more this feature will also help when we implement time series modeling.

Trivariate Analysis: Figure 4 shows interesting relation between new_cases which is going to be our predictor variable and rest of the features. We can see a clear pattern of diabetes, smoking correlation with the cases, in similar fashion total tests, new tests and population also track the cases nicely. Finally "Days since the first case" show the relation of linear increase with respect to new cases making it a very important feature.

Correlation matrix below figure 4 also shows the similar information.

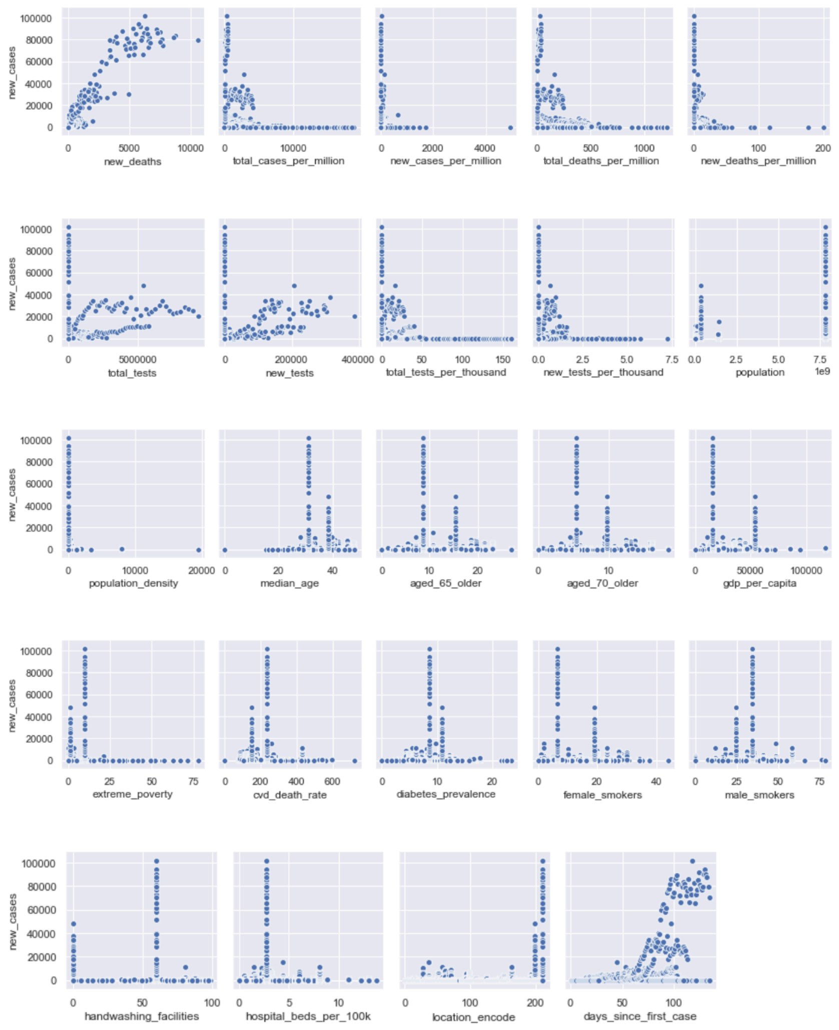
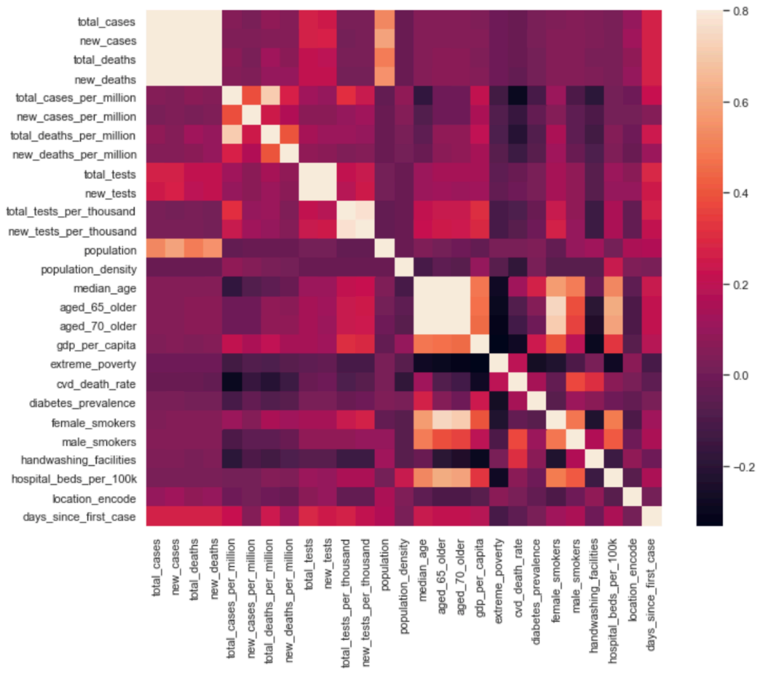


Figure 4



Modeling:

I took the dataset and split it 70-30 train test split and randomized the samples and thus completely losing any time series data benefit, but this was done to set the baseline for our best models.

As the dataset is randomized. Input will be per day event information with the features and output is the new cases that day. To be fair, I have removed total cases and case related information from the feature set.

Metric used to measure models:

Root Mean Squared Log Error, where RMSE is applied to the log of the target and the prediction output. It works as the approximation to the percentage error between our forecasting model and the target which is a nice way to understand the errors in our model. Closer to 0 the model the better it is.

Comparing baseline models:

I selected K Nearest Neighbors as our baseline model and as expected it did not do well with the log error of 1.21, basically predicting at random.

As a next step instead of using all the features, we only selected the feature that were important based on our analysis and used the below models.

Type of Algorithm	Log error
Linear Regression Model	2.39
Lasso Model	2.38

The above two models also failed to make any meaningful prediction when the model is trained on the shuffled data.

With these results it is clear that the model will have a best chance with the time series implementation.

Time series Models:

Before starting of the time series models I noticed couple of things that needed feature engineering.

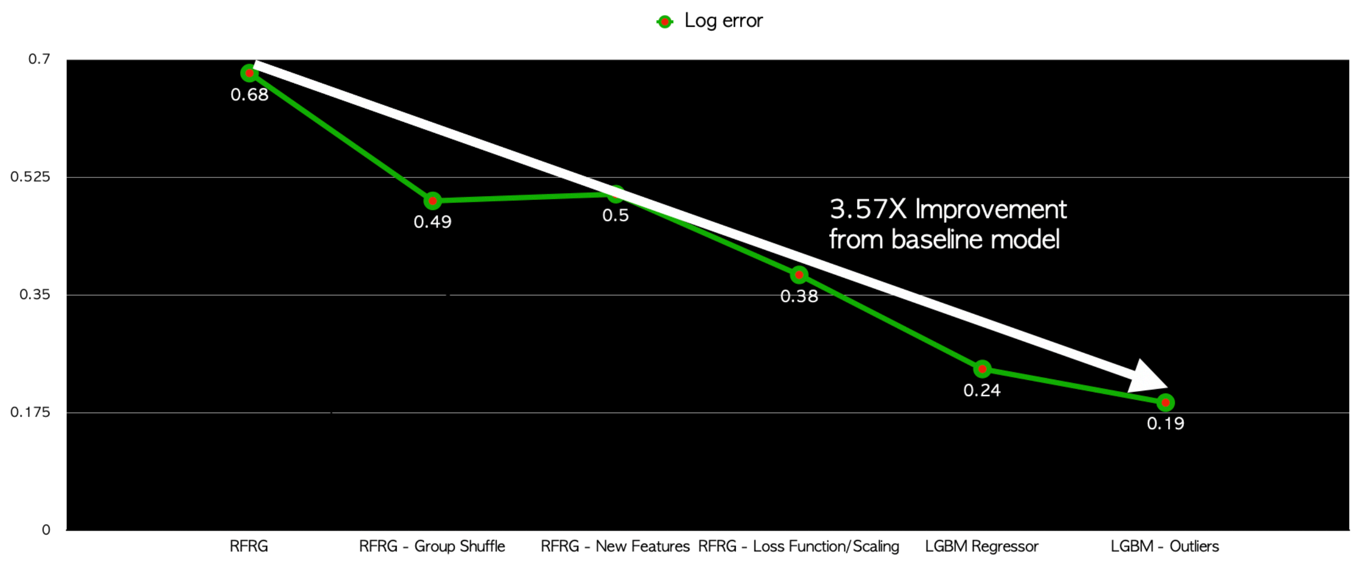
1. To encode the country names to a integer.
2. Drop the country names and aggregated stats with the label “World”

3. Add new feature to the dataset that can make use of last days information.

But first lets start with baseline time series model Random forest regressor.

I again split the dataset into 70-30% train-test but did not randomize to preserve the time information. Instead I randomized the data by groups of countries so the time information within the country will still be available. After that I realized I needed more features, so I did more feature engineering and added new features “Last_day_deaths”, “Last_day_cases” etc, which further improved the performance of the model.

After the above changes, I tuned the number of estimators for the best performance, next I changed the loss function of the above model by taking the log of the target to make the distribution closer. Finally I tried the new LGBM regressor which showed improvement in performance and final model was 75% accurate.



Above chart shows how our error comes decreasing with the changes we make. Overall we improved the accuracy of the model by 3.57x from our baseline. Finally when looking at the LGBM regressors, I noticed for the new cases lower that 8000 for the day the model was super accurate close to 92% but for new cases more that 8000 model fell short. Upon further analysis I noticed the dataset is not balanced when it comes to new cases and out of 16000 samples only 72 samples had cases greater than 8000 and none were in the training set unfortunately.

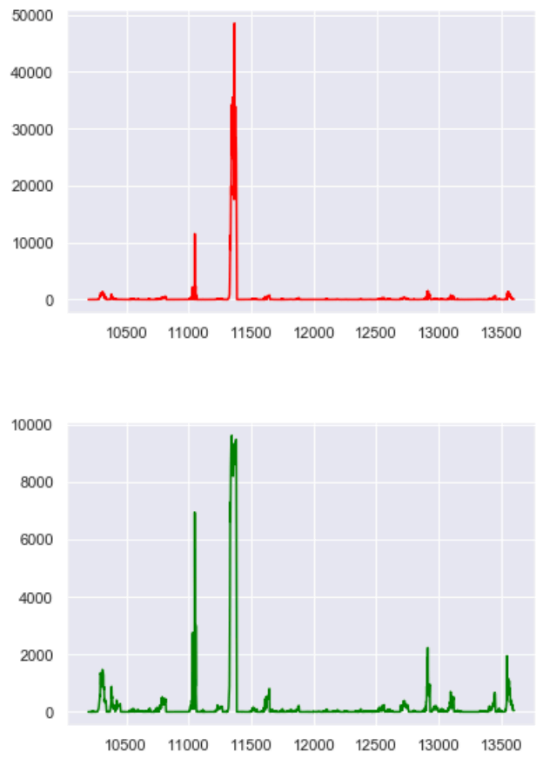


Figure 6

Red is the true data and green is the predicted data, as we can see model predicts the trend currently but when the new cases spike to 50000 the model can't follow the rise similarly mainly because the training data lacks such behavior. I removed some of these outliers and model performance further increase to 82% which is shown in the graph above. I am sure eliminating the outliers is not the right approach and next steps will be looking at how to augment the data without removing the outliers.