PROJECT:
PREDICTION MODEL FOR COVID-19 NEW DEATHS

STUDENT: ISTERI KLAJDI
ACADEMIC YEAR: 2020/2021

Big Data Course

Introduction

The main purpose of this project is the analysis of the "Our World in Data" Covid-19 Dataset and the creation of a ML model for the prediction of new deaths related to the coronavirus.

The analysis is generalized for all the locations contained in the dataset but in this report, we will focus mainly on Italy.

We can summarize the project in:

1. Environment Setup
2. Data Exploration & Visualization
3. Creation of a Machine Learning Model
4. Predictions before Vaccinations
5. Predictions after Vaccinations
6. Execution Time

Environment Setup

ANACONDA:

A python distribution and package manager with GUI.

Gives us a large variety of libraries and packages for data analysis and visualization, it also includes Jupyter Notebook.

JUPYTER:

It offers a web-based environment for working with notebooks.

Notebooks are also used for transformation, statistical modeling, data visualization, machine learning and they also integrate leverage Big Data tools such as Apache Spark.

APACHE SPARK:

It is a unified analytics engine for large-scale data processing.

Spark is a cluster computing platform designed to be fast, it extends MapReduce model to efficiently support more types of computations, such as interactive queries.

It is also designed to be highly accessible thanks to simple APIs in Python, Java, Scala

Data Exploration

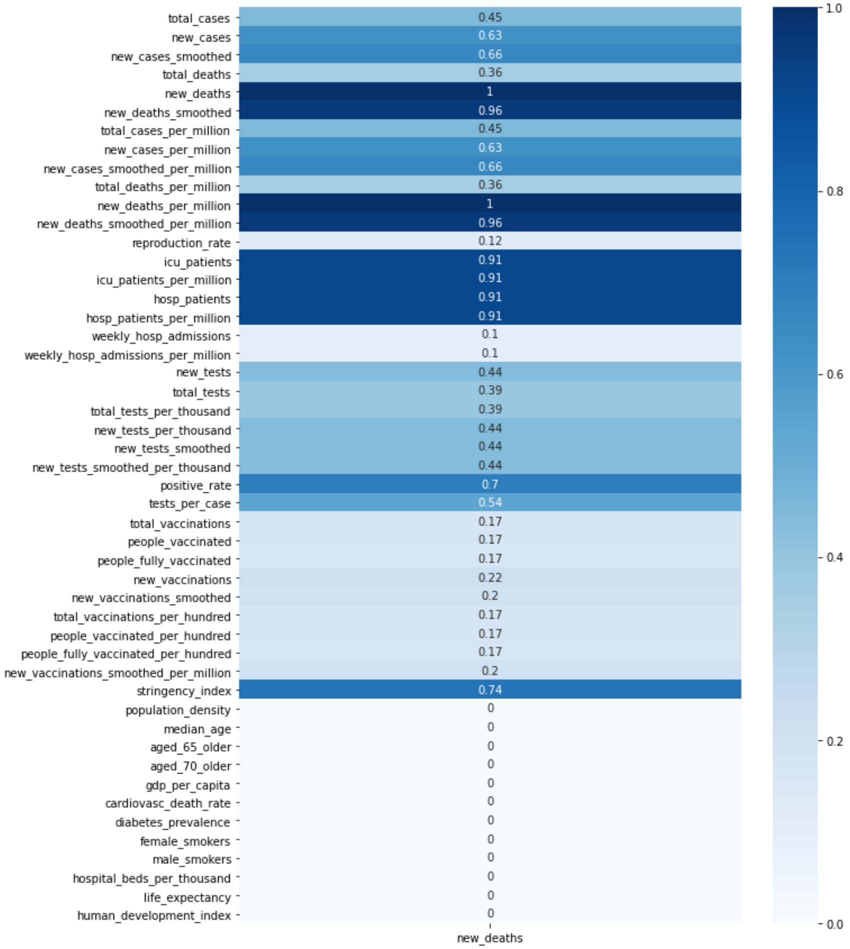
The "Our World in Data" Covid-19 Dataset is composed of the following columns, here we can see some statistical information about the data.

summary	count	mean	stddev	min	max
iso_code	85580	None	None	ABW	ZWE
continent	81451	None	None	Africa	South America
location	85580	None	None	Afghanistan	Zimbabwe
date	85580	None	None	2020-01-01	2021-05-03
total_cases	83470	832778.9957349946	5757477.472047493	1.0	1.52870507E8
new_cases	83468	5835.210092490535	36508.86533731523	-74347.0	905992.0
new_cases_smoothed	82467	5814.830038548904	35814.991549206636	-6223.0	826374.286
total_deaths	73790	23163.885973709173	137024.56622028106	1.0	3202523.0
new_deaths	73948	139.27223183858928	760.1312758706464	-1918.0	17906.0
new_deaths_smoothed	82467	123.41811175379846	695.9858349571732	-232.143	14435.143
total_cases_per_million	83019	10162.936744359886	19463.958179571233	0.001	171901.896
new_cases_per_million	83017	74.39997649878887	175.60479416028457	-2153.437	8652.658
new_cases_smoothed_per_million	82021	74.5016326550511	149.09617539623656	-276.825	2648.773
total_deaths_per_million	73352	226.7381250409071	397.8053083164839	0.001	2877.95
new_deaths_per_million	73510	1.5092371786151573	3.977495993528583	-76.445	218.329
new_deaths_smoothed_per_million	82021	1.338711878665185	2.9391449349779664	-10.921	63.14
reproduction_rate	69306	1.0182404120855324	0.35633672563155433	-0.01	5.77
icu_patients	8691	1087.9978138303993	3033.991358320877	0.0	29990.0
icu_patients_per_million	8691	26.403532965136367	27.860613702241253	0.0	192.642
hosp_patients	10821	4832.752795490251	12433.561423445204	0.0	129637.0
hosp_patients_per_million	10821	173.69404269475964	216.330380008436	0.0	1532.573
weekly_icu_admissions	790	280.36216708860724	588.1153416060671	0.0	4037.019
weekly_icu_admissions_per_million	790	21.10003544303796	37.10012126341996	0.0	279.13
weekly_hosp_admissions	1298	3994.2353412943	11634.79089101694	0.0	116232.0
weekly_hosp_admissions_per_million	1298	115.32590986132524	230.24720305086265	0.0	2656.911
new_tests	38945	44019.99345230453	228082.89074859317	-239172.0	3.2022805E7
total_tests	38652	5963484.145788058	2.702657004024681E7	0.0	4.13502739E8
total_tests_per_thousand	38652	227.4760523388175	495.3537686855608	0.0	6233.953
new_tests_per_thousand	38945	1.9372400564899257	15.289038589037073	-23.01	2827.217
new_tests_smoothed	44625	41416.30742857143	148450.0041091272	0.0	4594014.0
new_tests_smoothed_per_thousand	44625	1.7755527619047682	4.810456516565619	0.0	405.595
positive_rate	42904	0.08897072534029671	0.09759395562535529	0.0	0.742
tests_per_case	42311	159.4553425823062	864.9287640966207	1.3	44258.7
tests_units	46079	None	None	people tested	units unclear
total_vaccinations	9551	1.4957485974452937E7	6.869741373731461E7	0.0	1.162067962E9
people_vaccinated	8910	9225179.1661055	3.9027792779540956E7	0.0	6.04516098E8
people_fully_vaccinated	6576	4772619.979622871	1.8868453614355136E7	1.0	2.75959041E8
new_vaccinations	8110	424912.7676942047	1698118.626609131	0.0	2.4728855E7
new_vaccinations_smoothed	15322	226123.2664143062	1159589.1827649393	0.0	2.0323434E7
total_vaccinations_per_hundred	9551	13.58571144382781	21.893459977692924	0.0	211.08
people_vaccinated_per_hundred	8910	9.51923007856343	13.923543885901049	0.0	111.32
people_fully_vaccinated_per_hundred	6576	5.181970802919693	9.721152822921756	0.0	99.76
new_vaccinations_smoothed_per_million	15322	2784.0497324109124	4620.116653779478	0.0	118759.0
stringency_index	72673	58.71260275480682	21.648871289508666	0.0	100.0
population	85029	1.283663044966776E8	6.902714046105045E8	809.0	7.794798729E9
population_density	79659	349.79385832101934	1703.2164207822327	0.137	20546.766
median_age	77078	30.521583331174902	9.115053092093216	15.1	48.2
aged_65_older	76198	8.772912753614143	6.223711873362256	1.144	27.049
aged_70_older	76646	5.556767946141541	4.248685782933569	0.526	18.493
gdp_per_capita	77418	19139.031322351304	19826.54402373803	661.24	116935.6
extreme_poverty	52700	13.35125616698124	19.943899746923773	0.1	77.6
cardiovasc_death_rate	78005	257.8088030126566	118.77751200084232	79.37	724.417
diabetes_prevalence	79158	7.821495237373721	3.9780571783350993	0.99	30.53
female_smokers	61115	10.519806561401039	10.402752297957854	0.1	44.0
male_smokers	60214	32.657161872646405	13.475414080788601	7.7	78.1
handwashing_facilities	39197	50.91309169068946	31.763061733684147	1.188	98.999
hospital_beds_per_thousand	71182	3.0294416846951417	2.4634261515052818	0.1	13.8
life_expectancy	81224	73.16502240718631	7.5497456561661656	53.28	86.75
human_development_index	77890	0.7271036590064779	0.15005860466163015	0.394	0.957

Target Value to predict

Data Visualization (1)

We will focus our prediction on the new Covid-19 deaths in Italy and to do that we need to understand which are the features to include for the creation of the ML model. To make this choice we will analyze the correlation between the new deaths and all the other columns of the dataset using a correlation heatmap.



Where ρ = correlation and -1 < ρ < 1

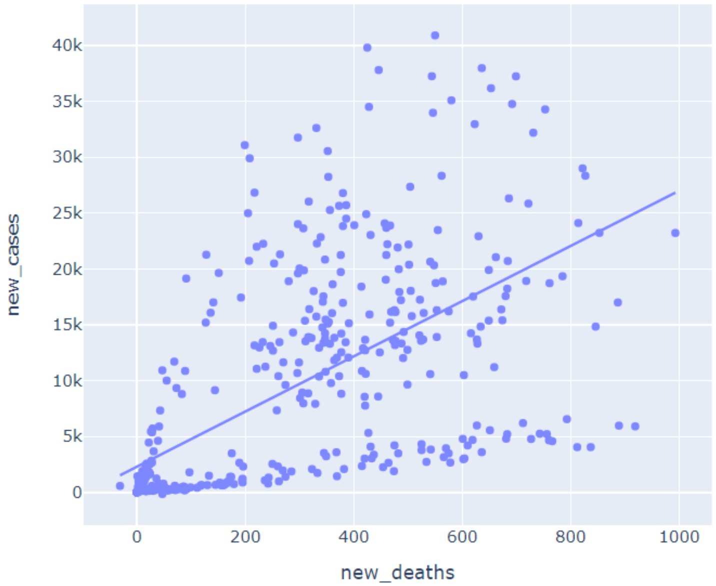
- ρ = 1 ➔ Strong Direct Correlation
- ρ = 0 ➔ No Correlation
- ρ = -1 ➔ Strong Inverse Correlation

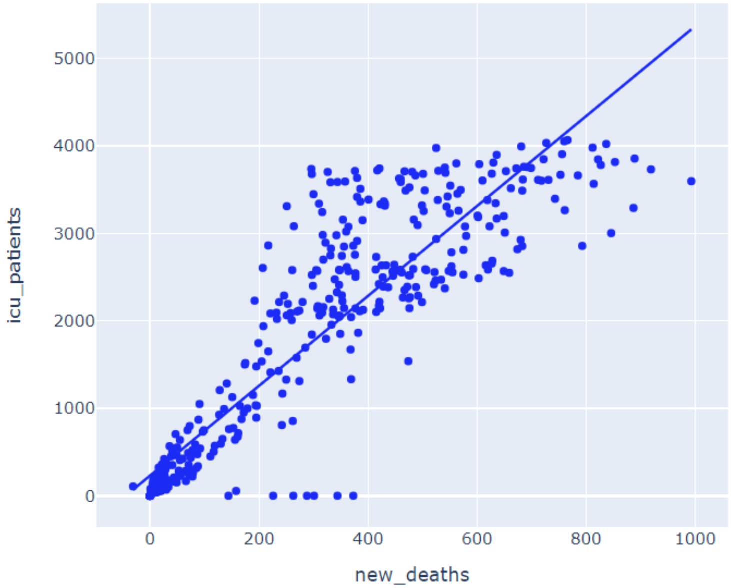
Here we can see that we have a strong correlation between:

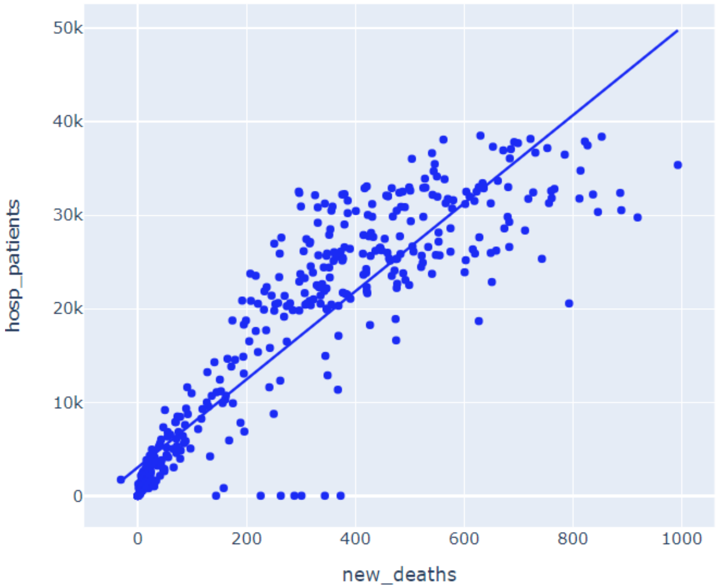
- new_deaths & new_cases → ρ = 0.633
- new_deaths & icu_patients → ρ = 0.912
- new_deaths & hosp_patients → ρ = 0.913

Data Visualization (2)

In the following scatterplots we can see a trendline calculated with the OLS method, the trendline's slope show us the grade of correlation between the two variables, due to these grades we will choose these variables for the prediction.





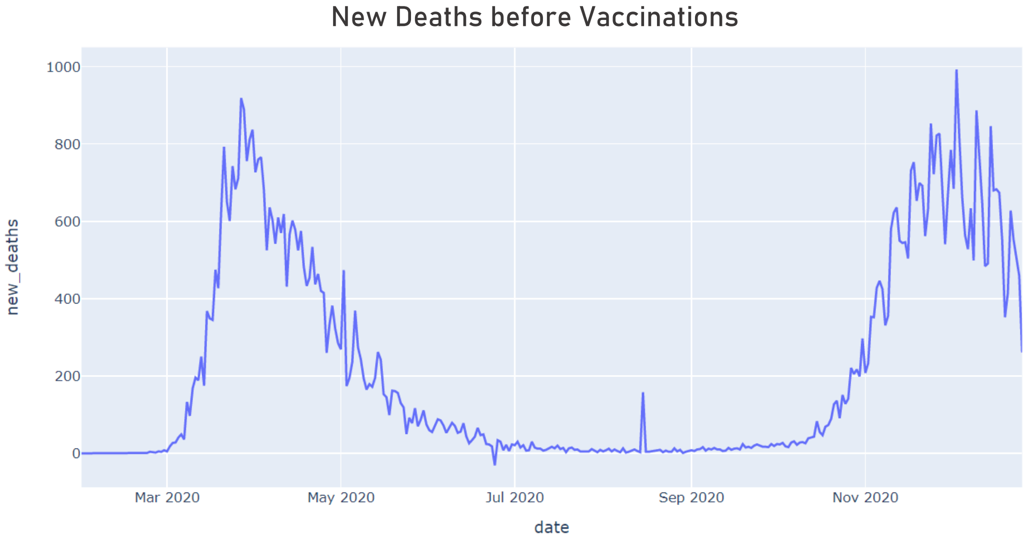


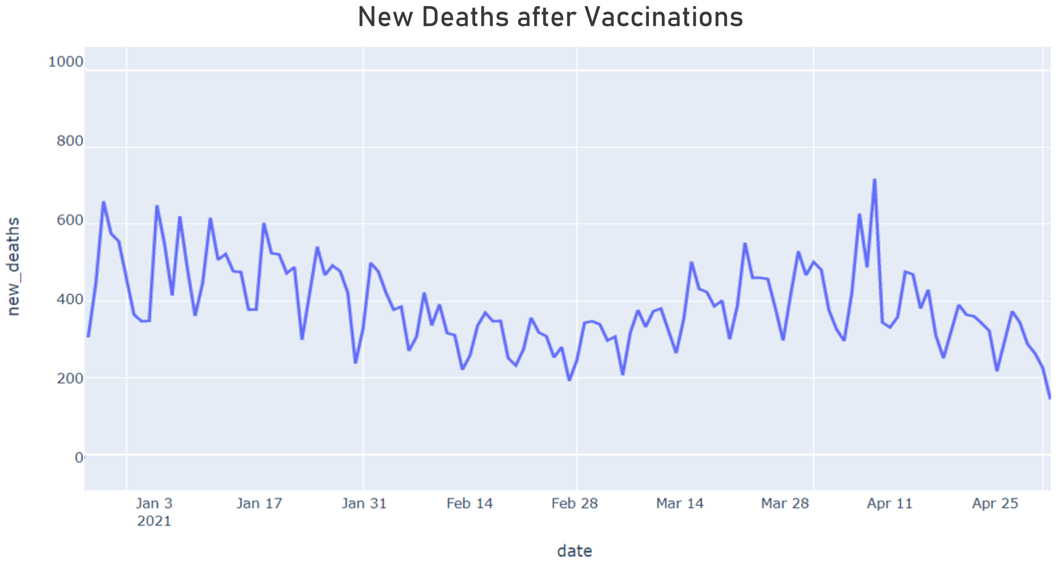
Data Visualization (3)

We will split the dataframe in two parts:

1. Dataframe with the data before Vaccinations;
2. Dataframe with data after Vaccinations.

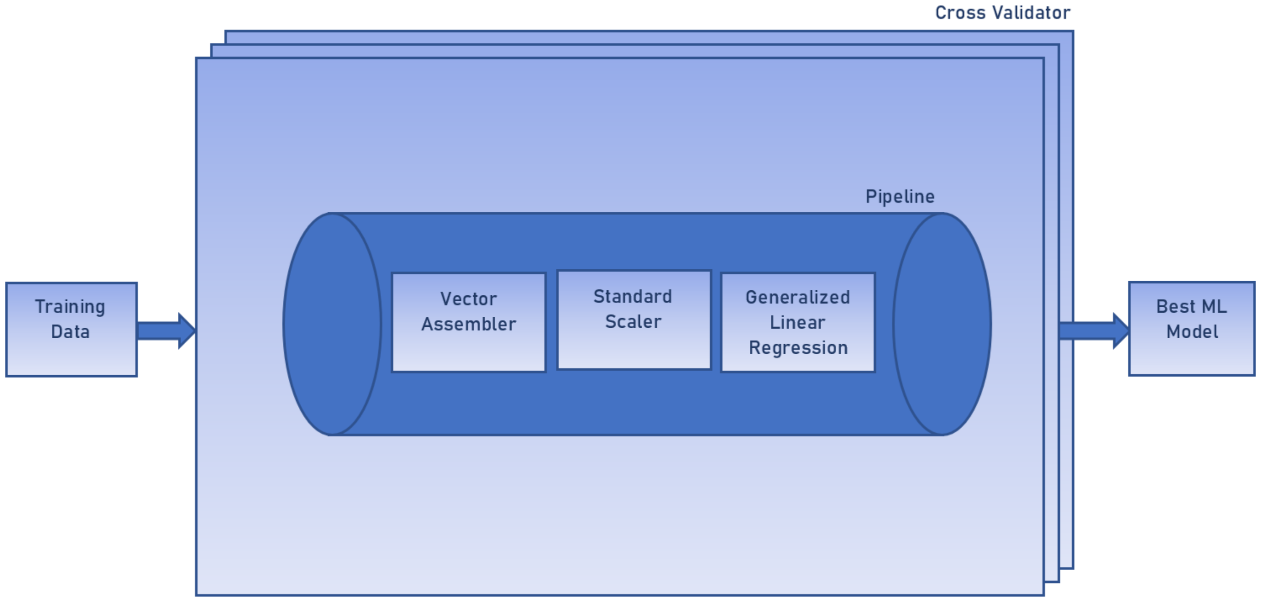
The objective is to split the first dataframe "before vaccinations" in two parts where 70% will be used to train the model and 30% to test the model. After that we will apply the ML model to the second dataframe "after vaccinations" to see if the model can give a quality prediction of it.





Machine Learning Model

Now we have everything set for the creation of the ML model, to do so we will take advantage of the Apache Spark pipeline.



Basically, a pipeline is a set of ML stages composed of transformers and estimators, in our case composed of:

- Vector Assembler: used to create a feature vector;
- Standard Scaler: a transformer that normalizes each feature of the vector assembler to have unit standard deviation, used to have a lower dispersion distribution;
- Generalized Linear Regression: an estimator which takes features and creates a model which is a transformer used to predict futures values of a label

Then we have a Cross-Validator, which is basically a hyperparameter tuner, its main function is to automatically split the dataframe into a set of folds which are used for training/test of the model, for each fold the validator analyzes results of prediction, then compares all the results and choses the best parameters to assign to the model, parameters are chosen inside a Param Grid, then the validator gives us the best model found.

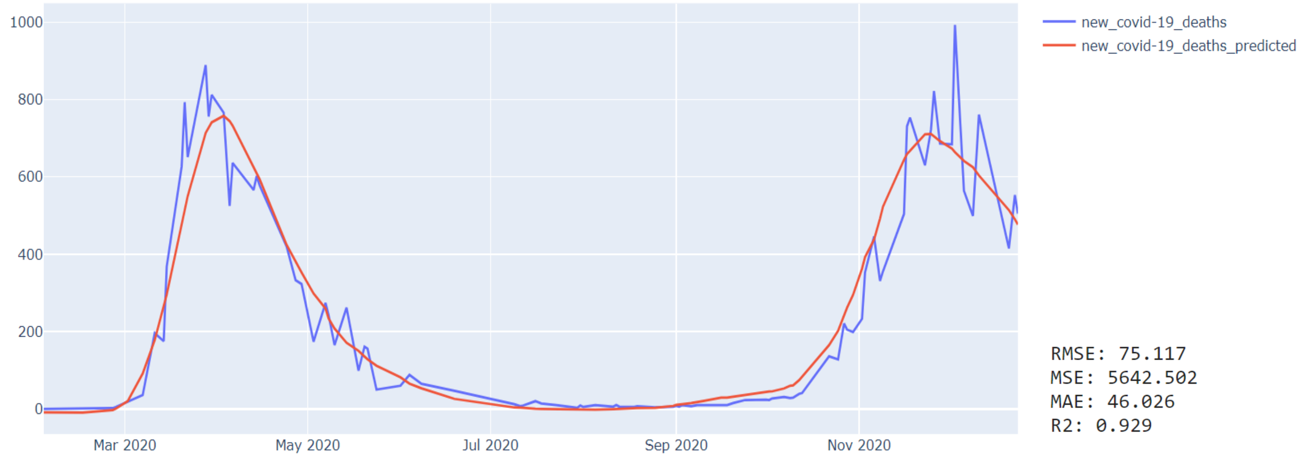
Predictions before Vaccinations

After training and testing the model, we applied it to the dataframe with data before vaccinations to see the results and the precision in the predictions.

label	new_cases	icu_patients	hosp_patients	index	unscaled_features	features	prediction
0.0	0.0	0.0	0.0	3	(4,[3],[3.0])	(4,[3],[0.0315751...	-9.104579437450319
0.0	0.0	0.0	0.0	6	(4,[3],[6.0])	(4,[3],[0.0631503...	-9.1319248065238
0.0	0.0	0.0	0.0	12	(4,[3],[12.0])	(4,[3],[0.1263006...	-9.186615544670763
0.0	0.0	0.0	0.0	13	(4,[3],[13.0])	(4,[3],[0.1368256...	-9.195730667695257
0.0	0.0	0.0	0.0	14	(4,[3],[14.0])	(4,[3],[0.1473507...	-9.20484579071975
0.0	0.0	0.0	0.0	16	(4,[3],[16.0])	(4,[3],[0.1684008...	-9.223076036768738
2.0	131.0	36.0	164.0	26	[131.0,36.0,164.0...	[0.01346131007941...	-3.114129154261537
5.0	202.0	56.0	304.0	27	[202.0,56.0,304.0...	[0.02075713462627...	0.50461799780002
18.0	342.0	166.0	908.0	31	[342.0,166.0,908....	[0.03514326753557...	20.152549040516014
36.0	1247.0	567.0	3218.0	36	[1247.0,567.0,321...	[0.12813934098496...	91.72153314313375
196.0	2313.0	1028.0	6866.0	40	[2313.0,1028.0,68...	[0.23767946728004...	177.65195587290464
175.0	3497.0	1518.0	9890.0	43	[3497.0,1518.0,98...	[0.35934504845581...	265.77085230441935
368.0	3590.0	1672.0	11335.0	44	[3590.0,1672.0,11...	[0.36890155103128...	295.6799787889233
627.0	5986.0	2655.0	18675.0	49	[5986.0,2655.0,18...	[0.61510993996469...	477.1504384412424
793.0	6557.0	2857.0	20565.0	50	[6557.0,2857.0,20...	[0.67378481061618...	515.7456205332933
651.0	5560.0	3009.0	22855.0	51	[5560.0,3009.0,22...	[0.57133499268354...	549.9639294414459
889.0	5974.0	3856.0	30532.0	57	[5974.0,3856.0,30...	[0.61387684285818...	713.5930381028747
756.0	5217.0	3906.0	31292.0	58	[5217.0,3906.0,31...	[0.53608896705576...	725.4594001811868
812.0	4050.0	3981.0	31776.0	59	[4050.0,3981.0,31...	[0.41617027344754...	740.8794296036061
766.0	4585.0	4068.0	32809.0	63	[4585.0,4068.0,32...	[0.47114585277950...	757.881319629163

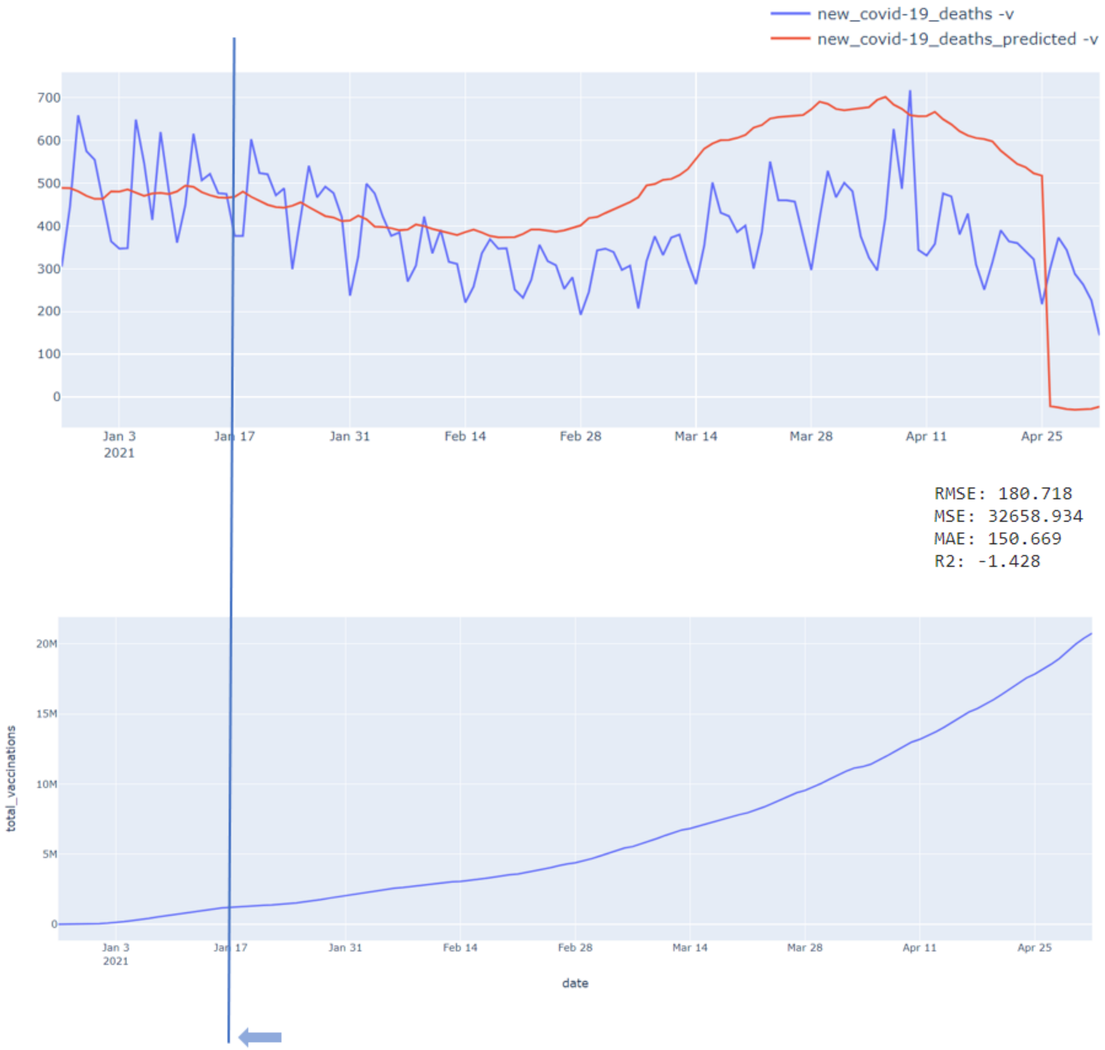
only showing top 20 rows

Total Rows -before vaccinations- : 104.000



RMSE: 75.117
MSE: 5642.502
MAE: 46.026
R2: 0.929

Predictions after Vaccinations



Here we also plotted the vaccinations graph to see an interesting trend, when the vaccinations' function starts to increase significantly, we have a loss of precision in the data predicted by the model due to the features we had chosen to build the model.

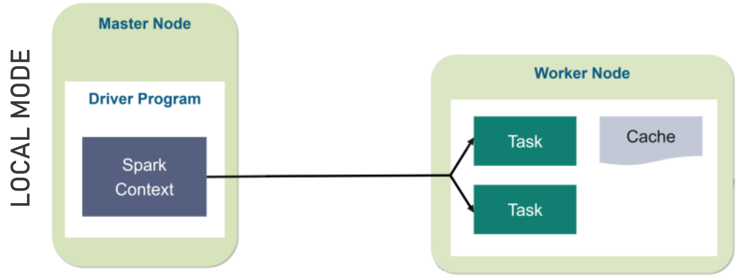
In this case we have a negative coefficient of determination (R2), so data are fitted very poorly, that's because we choose to not put total_vaccinations in the features for creating the model and this factor affects the predictions by decreasing the number of new deaths.

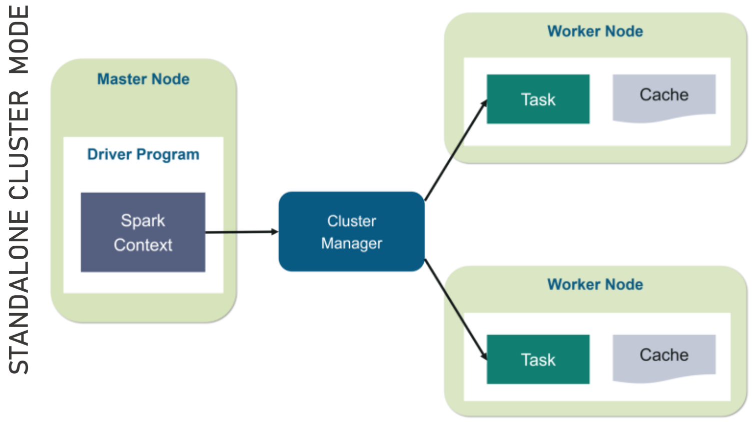
We can conclude that the model cannot process these new predictions, but we can also deduce that new vaccinations are lowering the number of new Covid-19 deaths swiftly, so we can at least predict that the pandemic will end very soon.

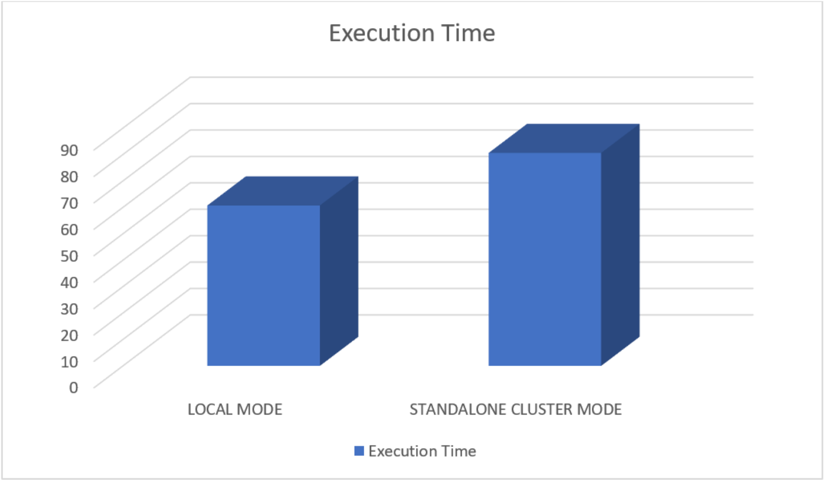
Execution Time

Apache Spark allows users to run its jobs in Local Mode and in Cluster Mode using its Standalone Cluster.

We run the code in Local Mode and Spark Standalone Cluster Mode to see the difference in execution times.







*Execution Time in Seconds

THANK YOU FOR YOUR ATTENTION