Machine Learning Engineer Nanodegree

Capstone Project

Ashraf Hussain 22 June, 2020

Machine Learning Engineer Nanodegree: Forecasting COVID-19 Cases – A Time Series Forecasting Model

I. Definition

Project Overview

On 31 December, 2019, the World Health Organization (WHO) was informed of an outbreak of "pneumonia of unknown cause" detected in Wuhan City, Hubei Province, China. Initially identified as coronavirus disease 2019, it quickly came to be known widely as COVID-19 and has resulted in an ongoing global pandemic. As of 20 June, 2020, more than 8.74 million cases have been reported across 188 countries and territories, resulting in more than 462,000 deaths. More than 4.31 million people have recovered.[^1]

In response to this ongoing public health emergency, Johns Hopkins University (JHU), a private research university in Maryland, USA, developed an interactive web-based dashboard hosted by their Center for Systems Science and Engineering (CSSE). The dashboard visualizes and tracks reported cases in real-time, illustrating the location and number of confirmed COVID-19 cases, deaths and recoveries for all affected countries. It is used by researchers, public health authorities, news agencies and the general public. All the data collected and displayed is made freely available in a GitHub repository.

Problem Statement

This project seeks to forecast number of people infected and number of caused by COVID-19 for a time duration of 14-days based on historical data from JHU. I will be using Amazon SageMaker DeepAR forecasting algorithm, a supervised learning algorithm for forecasting scalar (one-dimensional) time series using recurrent neural networks (RNN) to produce both point and probabilistic forecasts[^4]. DeepAR is an underutilized approach in this area.[^5] The dataset contains hundreds of related time series, and DeepAR outperforms classical forecasting methods including but not limited to autoregressive integrated moving average (ARIMA), exponential smoothing (ETS), Time Series Forecasting with Linear Learner for this type of applications.

At first, I was going to use DeepAR by AWS and compare it to AR (autoregressive model). However after exploring the data, I discovered that this dataset cannot be used for any Time Series Forecasting Model(TSFMs). Epidemic curves (epi curve) do not follow a standard time series requirement however they do follow Logistic & Gaussian functions that are defined as follows:

Epi curve of total number of cases follows Logistic Function is defined by:

> f(x) = capacity / (1 + e^-k*(x - midpoint))[^2]

Epi curve of new of cases follows Gaussian Function is defined by:

> f(x) = a * e^(-0.5 * ((x-μ)/σ)**2)[^3]

For this reason, in the dataset model I propose total cases will fit to to Logistic function and new cases to Gaussian functions.

Metrics

The error represents random variations in the data that follow a specific probability distribution (usually Gaussian). The objective of curve fitting is to find the optimal combination of parameters that minimize the error. Here we are dealing with time series, therefore the independent variable is time. In mathematical terms[^6]

> f(error) = f(time) + error

II. Analysis

Data Exploration

The datasets are accessed from files provided by the JHU GitHub repository time_series_covid19_confirmed_US.csv

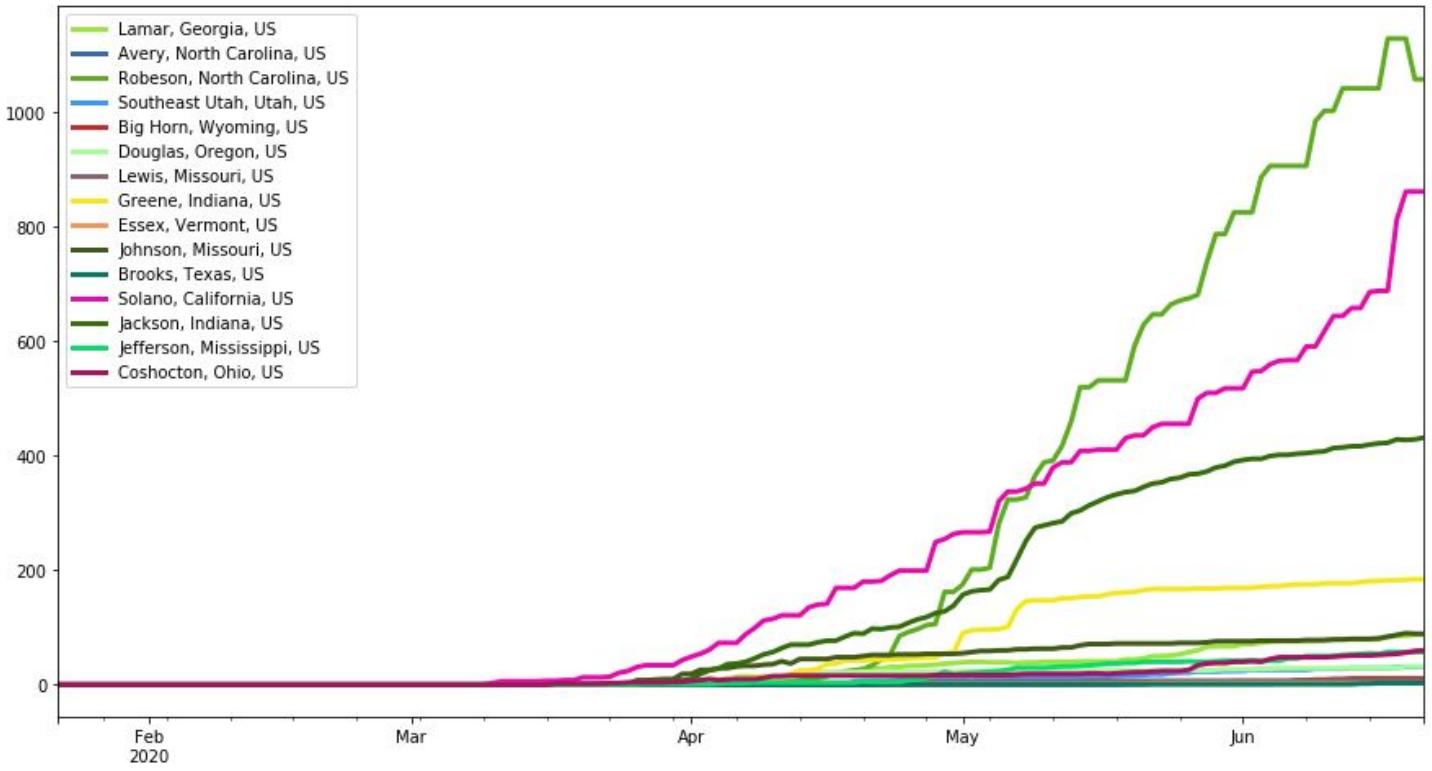
The file have the same columns:

- UID - UID = 840 (country code3) + 000XX (state FIPS code). Ranging from 8400001 to 84000056.
- iso2- Officially assigned country code identifiers 2 Chr (US, CA, ...)
- iso3 - Officially assigned country code identifiers 3 Chr.(USA, CAN, ...)
- code3- country code USA = 840
- FIPS -Federal Information Processing Standards code that uniquely identifies counties within the USA.
- admin2 - County name. US only.
- Province_State - The name of the State within the USA.
- Country_Region - The name of the Country (US).
- Combined_Key - Province_State + Country_Region
- Population - Population
- Number of cases are is columns where each column is a day

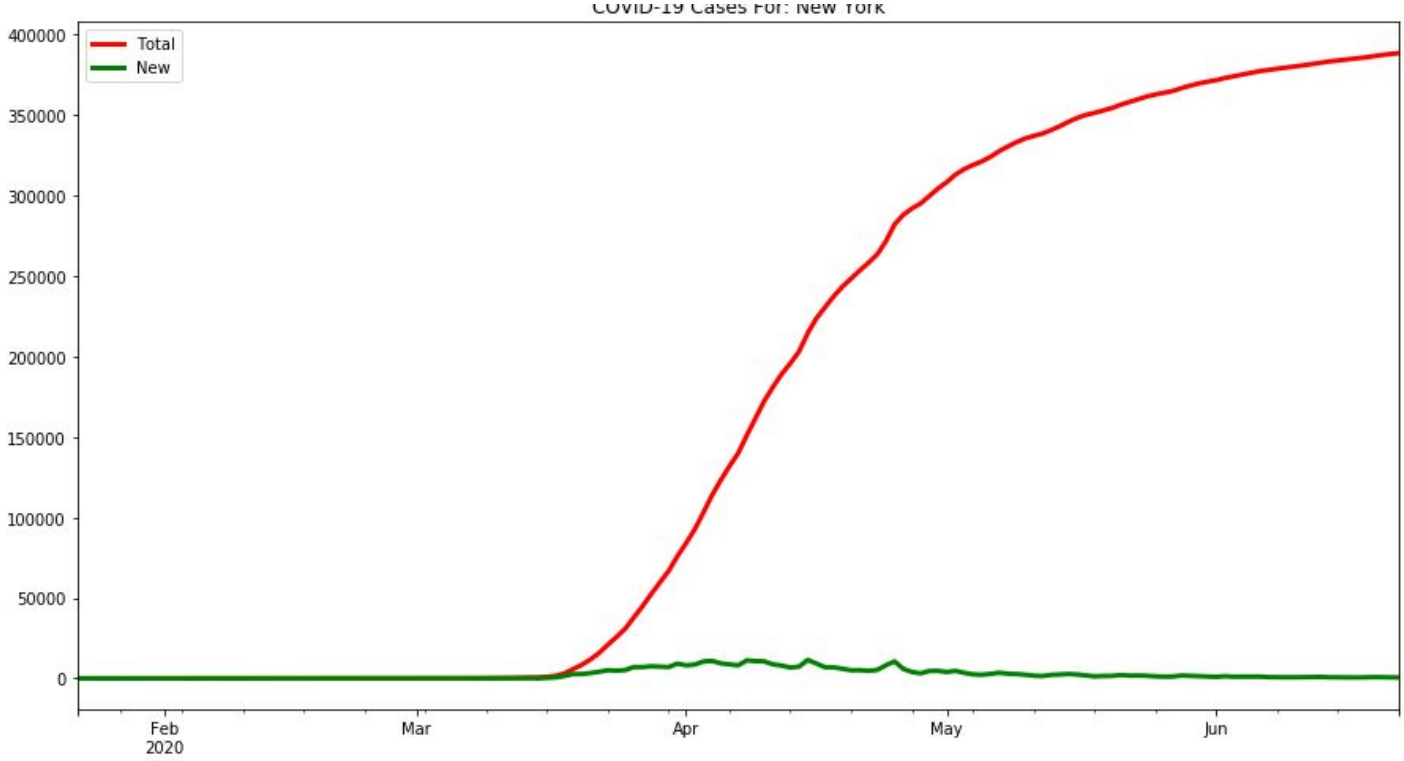
Exploratory Visualization

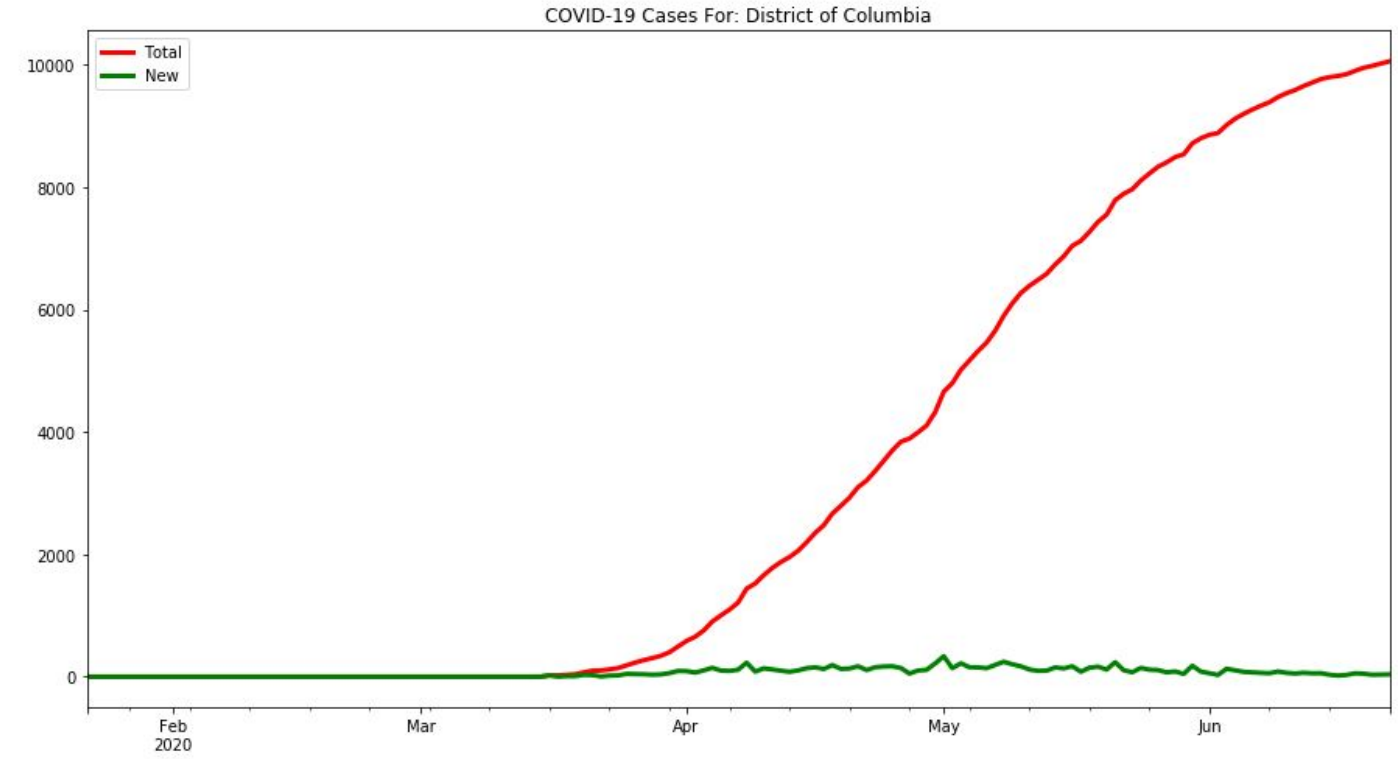
The plot below shows how the COVID-19 cases increase by city. When looking at the graph of 15 cities I noticed the first problem. I cannot use DeepAR on this data because DeepAR needs at least 300 observations available across all training time series. The current dataset has observations for the last 151 days only.

> We recommend training a DeepAR model on as many time series as are available. Although a DeepAR model trained on a single time series might work well, standard forecasting algorithms, such as ARIMA or ETS, might provide more accurate results. The DeepAR algorithm starts to outperform the standard methods when your dataset contains hundreds of related time series. Currently, DeepAR requires that the total number of observations available across all training time series is at least 300.
> Source: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html



Lets look at new vs total number of cases for some cities:





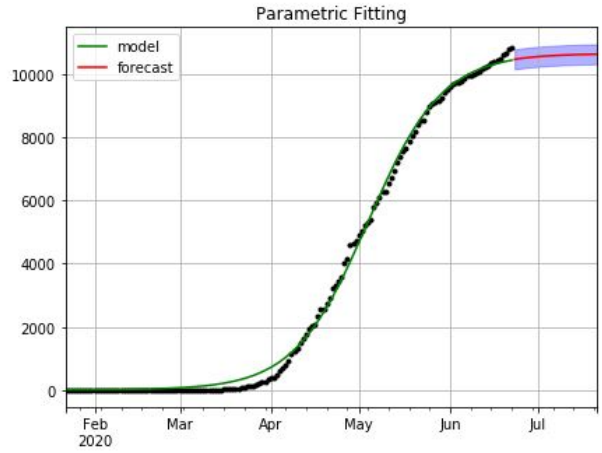
Algorithms and Techniques

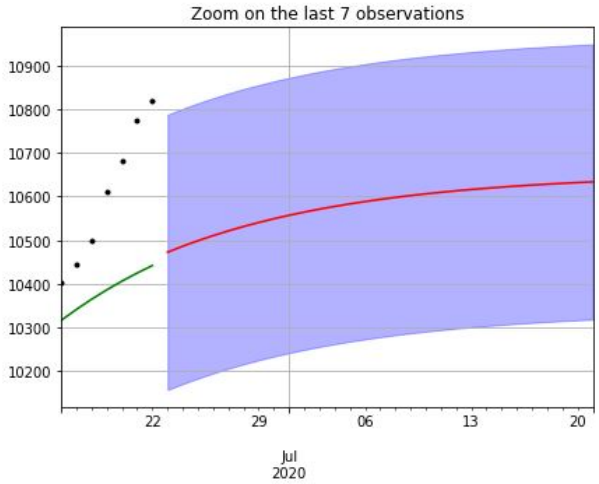
I will be using `scipy.optimize.curve_fit` , which is a part of SciPy package. This will be fitting a pre-defined Gaussian/Logistic Function which is very commonly used in epidemiology. [^7]

Benchmark

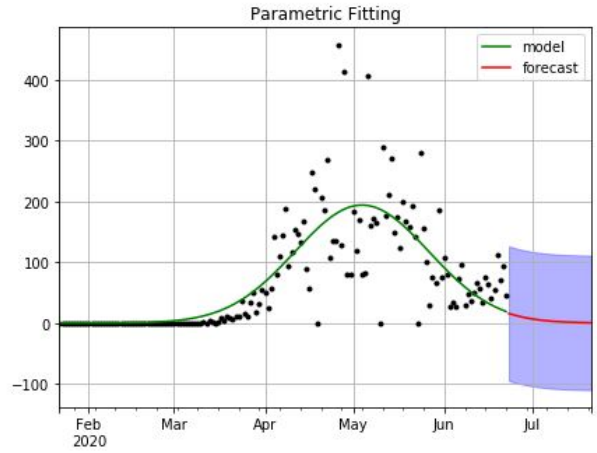
We are still in a crucial stage of the epidemic; there is no real benchmark. The model was able to predict future cases for some states very well. However, for others it was unable to predict at all. I used a random generator to pick sample cities to plot the following graphs:**Delaware**

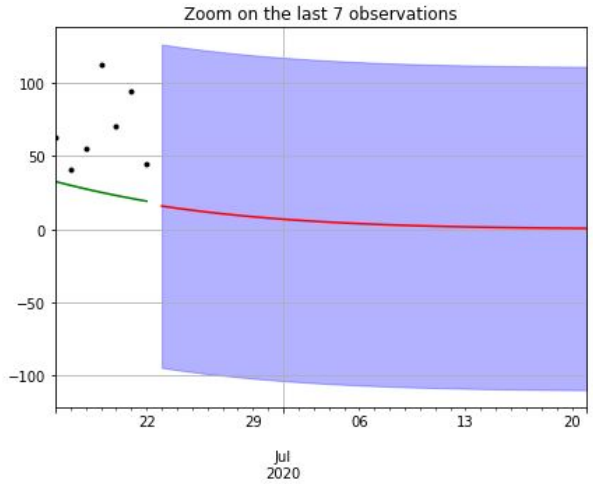
```
###Total Cases###
Making model for Delaware
forecast Delaware
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```






```
###New Cases###
Making model for Delaware
forecast Delaware
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```

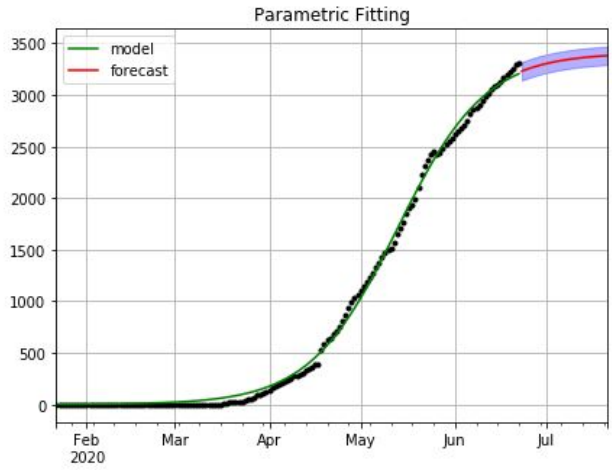


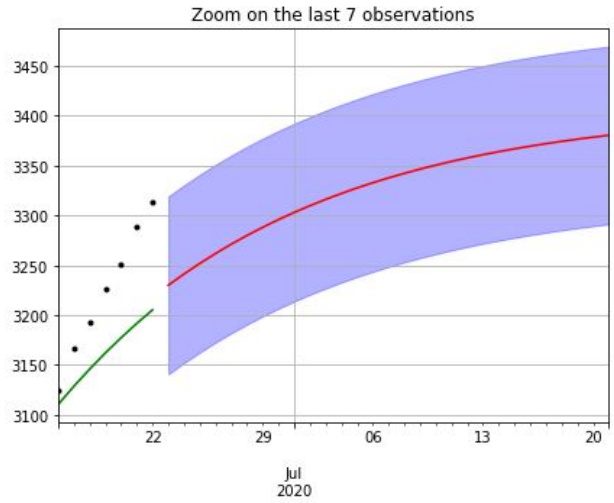


In Delaware, the number of new cases seem to be following an epi curve very well. This indicates that the social distancing measures have been working well to reduce the number of new cases. There was a spike in the number of cases in the week of Memorial Day (May 25), however, they seemed to have recovered quickly falling into a standard epi curve pattern.

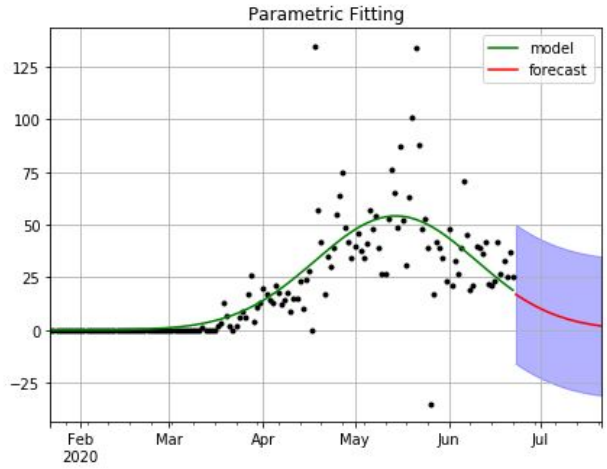
North Dakota

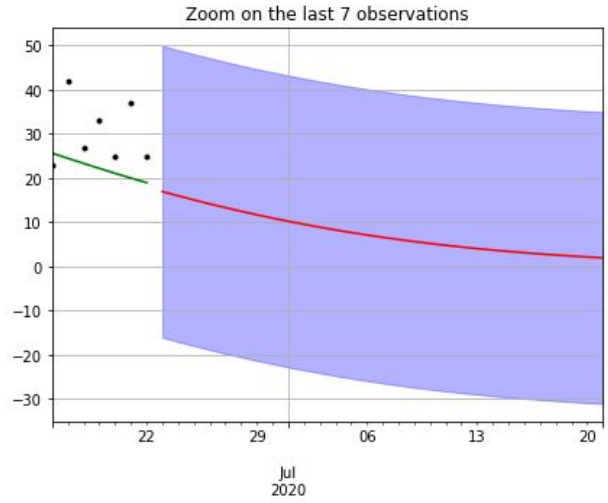
```
###Total Cases###
Making model for North Dakota
forecast North Dakota
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```






```
###New Cases###
Making model for North Dakota
forecast North Dakota
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```

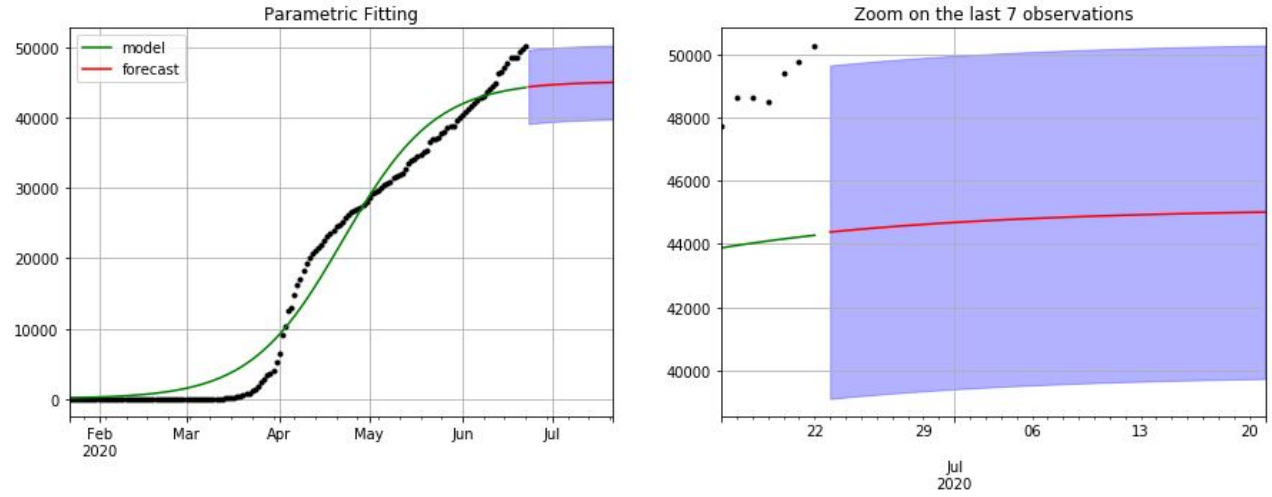




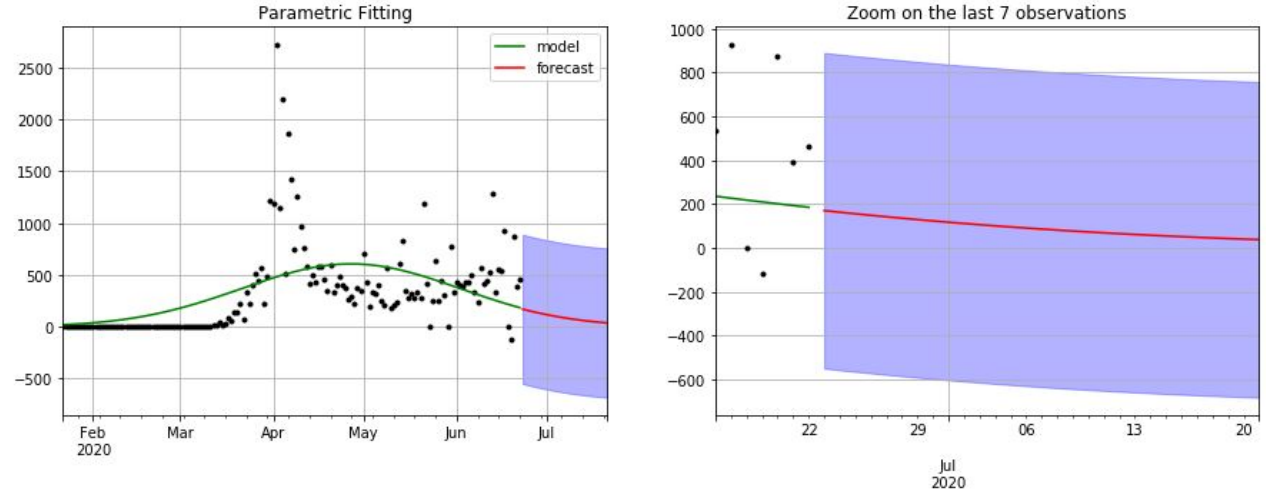
New covid-19 cases in North Dakota seem to be following the same pattern as that of Delaware indicating that social distancing measures are proving to be effective. Their cases peaked around Memorial Day, similar to Delaware, hwoever they seem to have take slightly longer to come back to a normal epi curve.

Maryland

```
###Total Cases###
Making model for Louisiana
forecast Louisiana
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```




```
###New Cases###
Making model for Louisiana
forecast Louisiana
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



Contrary to Delaware and North Dakota, new cases in Maryland peaked during the long weekend in April and seemed to stay within the mean of an epi curve. This irregular peaking may be attributed to people traveling between states for the long weekend and for other reasons.

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

The time_series_covid19_confirmed_US.csv

UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Regio
16	AS	ASM	16	60	NaN	American Samoa	US

UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Regio
316	GU	GUM	316	66	NaN	Guam	US
580	MP	MNP	580	69	NaN	Northern Mariana Islands	US
630	PR	PRI	630	72	NaN	Puerto Rico	US
850	VI	VIR	850	78	NaN	Virgin Islands	US

The data set was imported into a pandas Dataframe.

Data needed minimal data preprocessing because each Date was in one column and City and State were in other columns.

```
csv_file = 'time_series_covid19_confirmed_US.csv'
covid_df = pd.read_csv(csv_file)
```

Then the data was modified to remove the following columns and each state was sum

```
covid_df = covid_df.drop(['UID',
                          'iso2',
                          'iso3',
                          'code3',
                          'FIPS',
                          'Admin2',
                          'Country_Region',
                          'Lat',
                          'Long_',
                          'Combined_Key'], axis=1).groupby("Province_State").sum().T
```

Province_State	Alabama	Alaska	American Samoa	Arizona	Arkansas	Californi
1/22/20	0	0	0	0	0	0
1/23/20	0	0	0	0	0	0
1/24/20	0	0	0	0	0	0
1/25/20	0	0	0	0	0	0
1/26/20	0	0	0	1	0	2

Province_State	Alabama	Alaska	American Samoa	Arizona	Arkansas	Californi

5 rows × 58 columns

The dataframe Date was converted to datetime Index

```
## convert index to datetime
covid_df.index = pd.to_datetime(covid_df.index, infer_datetime_format=True)
```

Now we have a clean data set which will have Date as Index, and sum of cases for each State. From here onwards, we can use the following function to get cumulative number of cases (total) and new cases (new) for a given State in the dataframe.

```
def getCases(df, aState):
    # create total cases column
    error = 0
    try:
        df = pd.DataFrame(index=df.index, data=df[aState].values, columns=["total"])
        #print(dtf.head())
        # create daily changes column
        df["new"] = df["total"] - df["total"].shift(1)
        # Handling Missing Values
        df["new"] = df["new"].fillna(method='bfill')
    except:
        print("No State " + aState + " found")
        error = 1
        df = pd.DataFrame()
    return [df, error]
```

.	total	new
count	153	153
mean	168552.3987	2539.137255
std	162319.6416	3209.162717
min	0	0
25%	0	0
50%	139875	1075
75%	345813	4073
max	388488	11434

Implementation

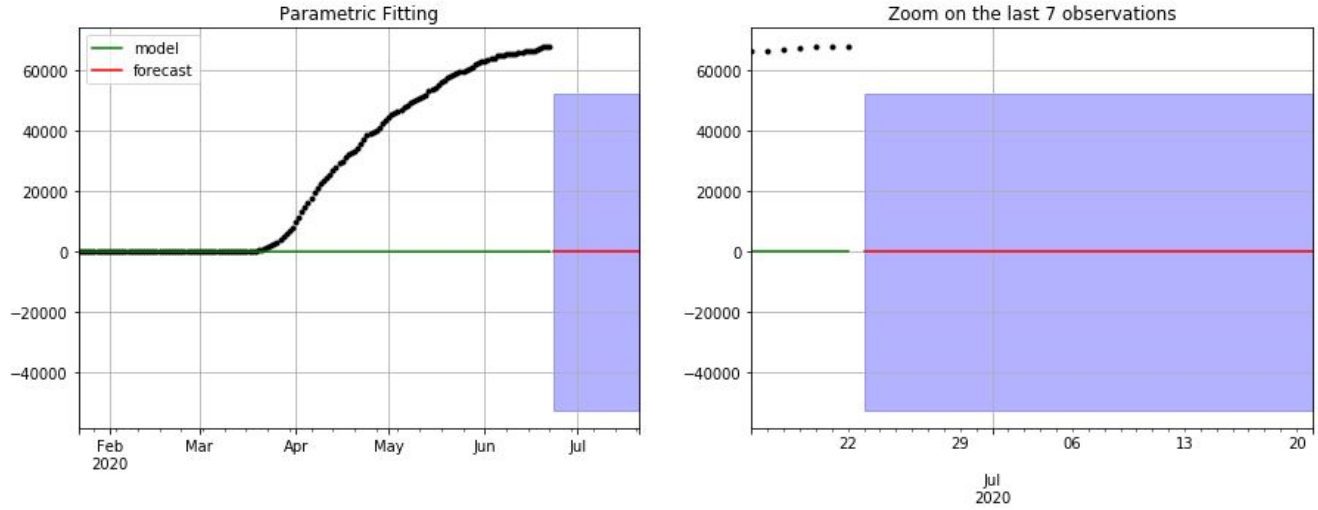
Model: scipy.optimize.curve_fit model from scipy provides a non-linear least squares to fit a function, f (such as Logistic or Gaussian functions) to a given dataframe. Forecast: The forecasting function was provided by `ts_utils.py` that apply the two models (total cases and daily increase) to a new independent variable: the time steps from today till N. It forecast 30 days ahead from today[^8]

Refinement

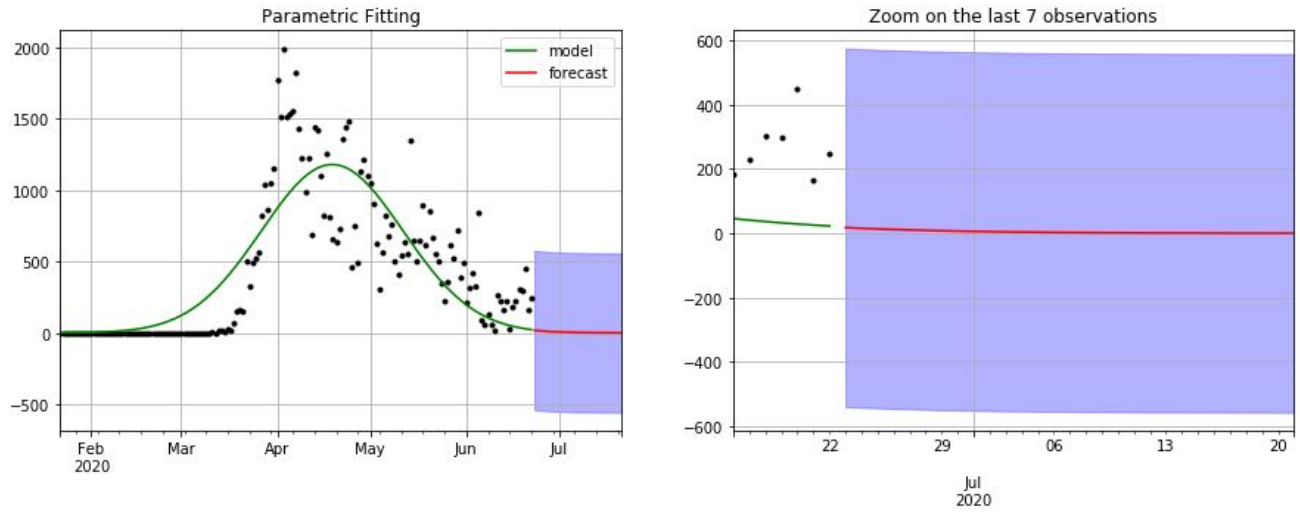
The model and the forecast work well when applied to certain States. However, there are a few States that do not work because they are on a different epi curve, for example

Michigan

```
###Total Cases###
Making model for Michigan
forecast Michigan
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



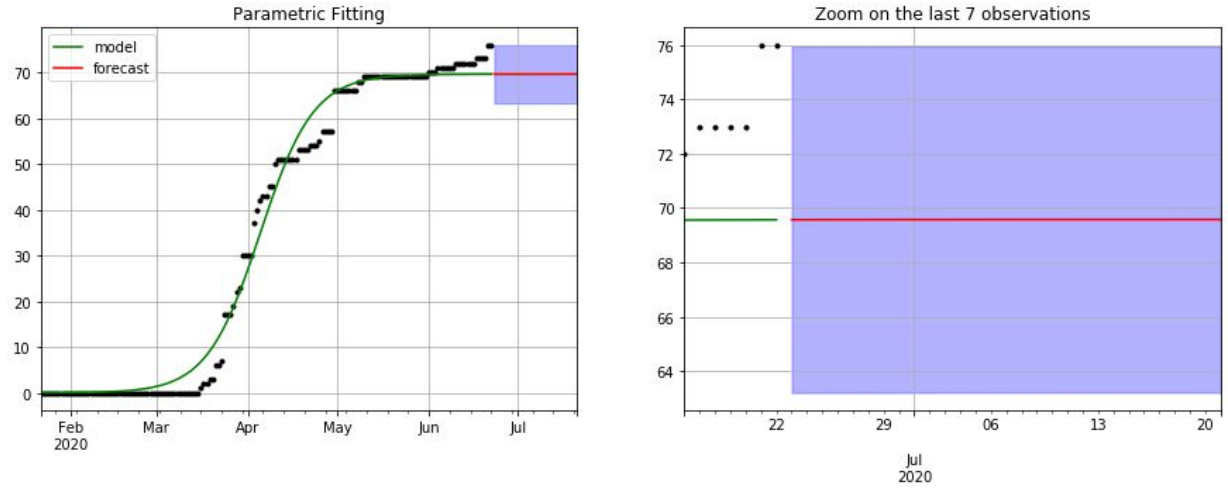

```
###New Cases###
Making model for Michigan
forecast Michigan
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



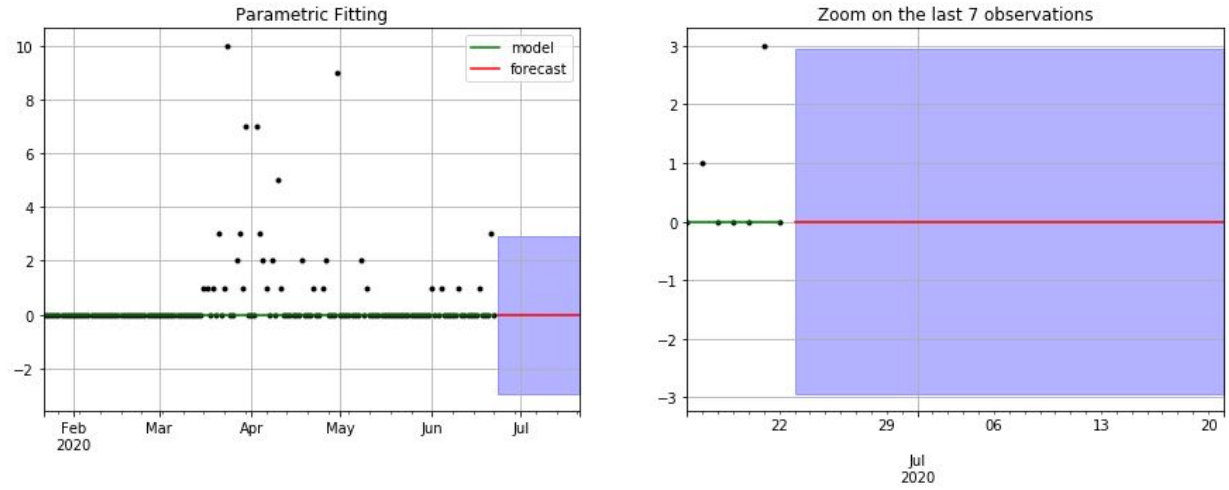
The model could not fit total cases to a Logistic Function

Virgin Islands

```
###Total Cases###
Making model for Virgin Islands
forecast Virgin Islands
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```




```
###New Cases###
Making model for Virgin Islands
forecast Virgin Islands
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



The model could not fit new cases to a Gaussian Function

A better package such as `earlyR` and `EpiEstim` that are part of R work better when applied to an epidemic dataframe. There are two well-researched articles written by Tim Churches that talk about using R to predict COVID-19 cases.

> Tim Churches is a Senior Research Fellow at the UNSW Medicine South Western Sydney Clinical School at Liverpool Hospital, and a health data scientist at the Ingham Institute for Applied Medical Research, also located at Liverpool, Sydney. His background is in general medicine, general practice medicine, occupational health, public health practice, particularly population health surveillance, and clinical epidemiology.

COVID-19 epidemiology with R by Tim Churches

Analysing COVID-19 (2019-nCoV) outbreak data with R - part 1

IV. Results

Model Evaluation and Validation

Given my limited understanding and knowledge in the field of epidemiology, this was the simplest model I was able to work with.

Justification

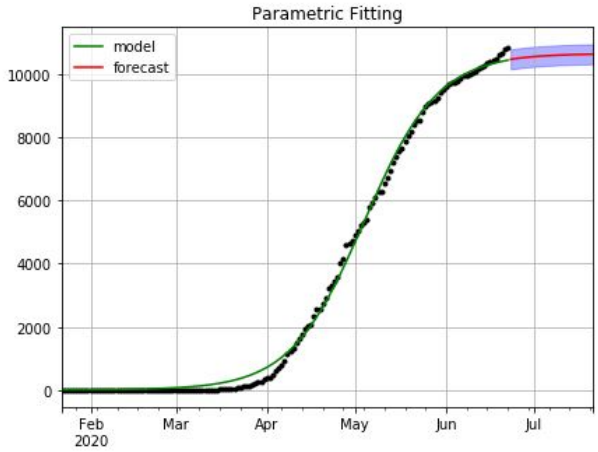
Research is ongoing and hence there is no current benchmark for the COVID-19 epidemic. The closest data comparison is with the Spanish Flu of 1918 that infected 500 million people worldwide and killed more then 50 million people.[^9] We could possibly take its epi curve and try to fit it to the COVID-19 epidemic. Again, given my limited understanding of how epidemics work, it is challenging to calculate the epi curve for the Spanish Flu. For this reason, it is not possible to predict how how any model for COVID-19 epidemic would turn out.

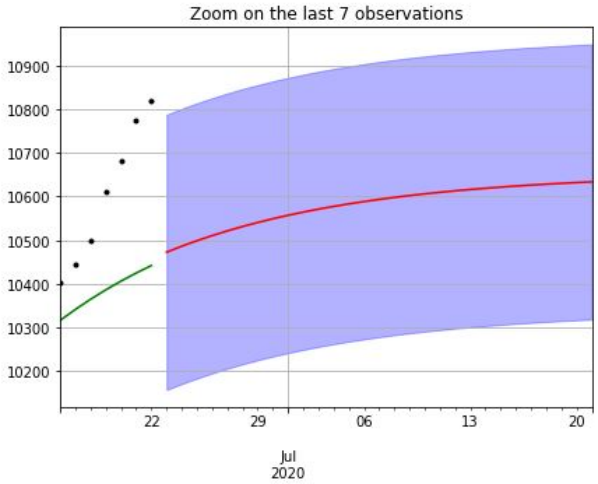
V. Conclusion

Reflection

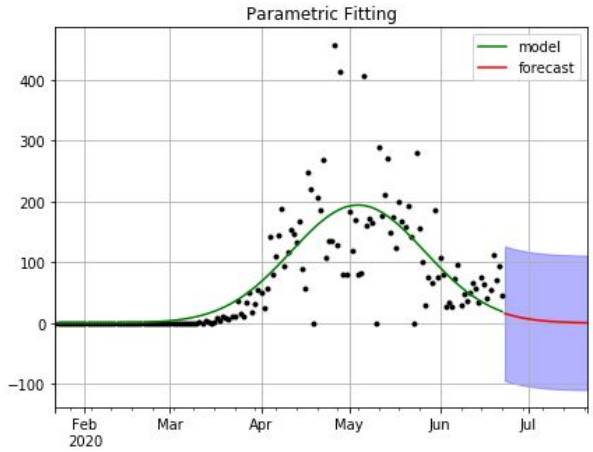
Based on my observations, the dataframes for **North Dakota** and **Delaware** fit the epidemic curve well.

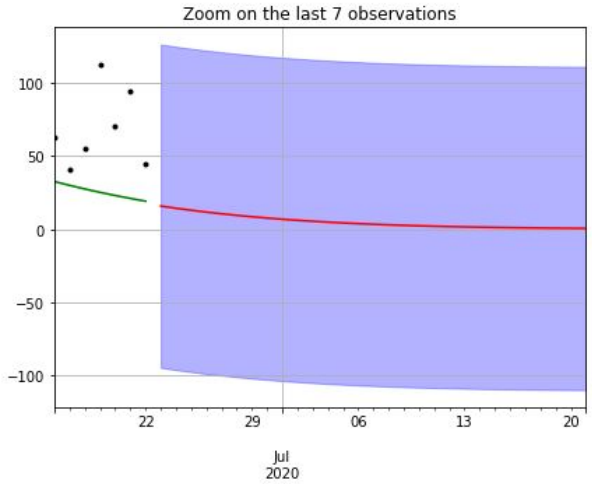
```
###Total Cases###
Making model for Delaware
forecast Delaware
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



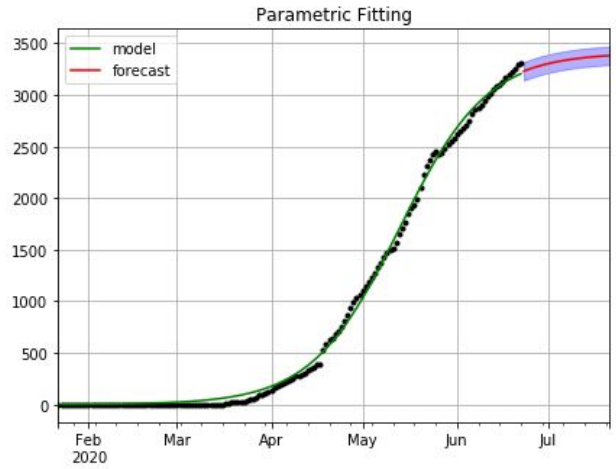


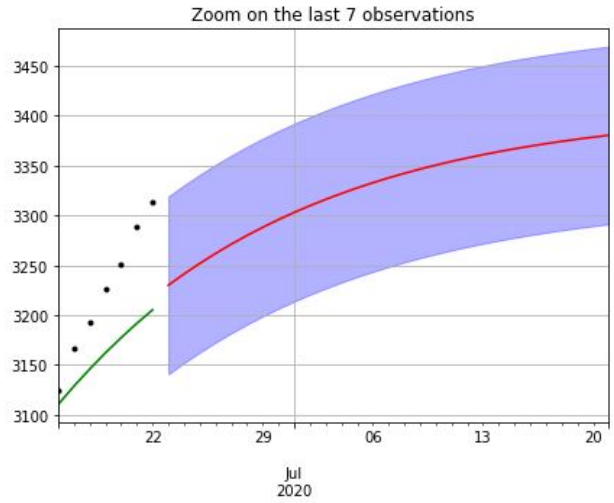

```
###New Cases###
Making model for Delaware
forecast Delaware
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```



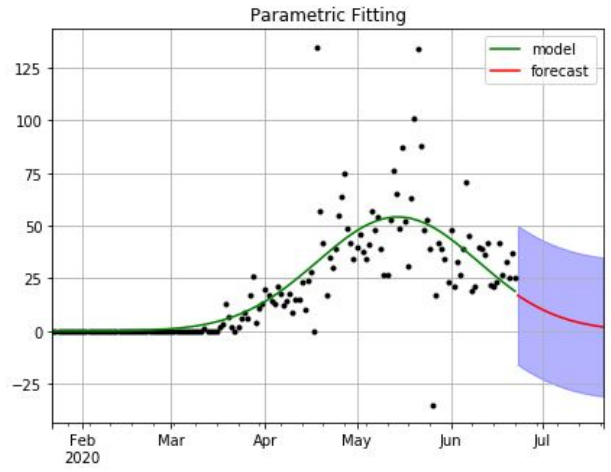


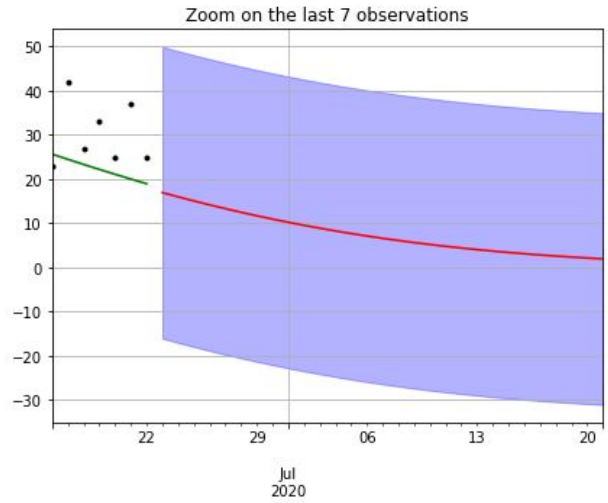
```
###Total Cases###
Making model for North Dakota
forecast North Dakota
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```






```
###New Cases###
Making model for North Dakota
forecast North Dakota
--- generating index date --> start: 2020-06-23 00:00:00 | end: 2020-07-21 00:00:00 | len: 29 ---
```





With respect to North Dakota, they have 3, 313 total cases, 2, 952 recovered cases with only 77 deaths. If they continue on this path, they are predicted to reach equilibrium (zero new cases) by the end of July or August 2020.

Delaware is another State that is following the epi curve closely. Though they had some peaks around Memorial Day week, they seem to have recovered well. If they continue on this trend, they are predicted to reach minimum amount of cases by the end of July or August 2020.

The only challenge with these predictions is that it does not take into account the human factors. Human factors can be defined as human interactions in relation to their environment, such as not following social distance measures; not wearing masks when going out in public areas; not following proper safety and sanitization rules; non-essential travel from one city to another to visit family, friends or to go on a vacation; and more. This will increase the rate of transmission leading to peaks in new cases as was witnessed around the Memorial Day week.

Improvement

1. Using `earlyR` and `EpiEstim` packages which is part of R.
2. Partnering up with an epidemiologist to a better understanding of the COVID-19 epidemic.

Final Thoughts

This project has taught me a lot about machine learning and how epidemiologists are using machine learning. There were several lessons to be learnt but the top three that stood out for me are:

1. Work with the right expert: My understanding of epidemiology is limited. I would have appreciated an opportunity to work with an epidemiologist to understand how an epidemic works in order to apply the nuances of the data to machine learning. It is really critical to understand the story behind the data to be able to build a good machine model.

2. Generalize the problem: My proposal was confined to a narrow dataframe and solution that did not leave any room for improvisation. This led to a tunnel vision when trying to build the data model. An alternative would have been generalizing the problem in the proposal which would have allowed me to manoeuvre in different ways to come up with innovative solutions.

3. Time management: There is a ton of published papers and research available that provide top-notch information. Going forward, I will allow myself sufficient time to go through available research before embarking on the solution. Some good sources of information are Google Scholar, Medium, GitHub repositories, and other open-source packages & libraries. Hindsight is indeed 2020. Armed with this knowledge, I am confident that I can continue to apply myself in the field of machine learning to find novel solutions to human challenges.

To conclude, I would like to thank the following people without whom I would not have been able to complete my project and get an understanding of how COVID-19 is being used in machine learning:

- Dr. Tim Churches.
- Dr. Jason Brownlee
- Mauro Di Pietro
- Subhasree Chatterjee

And last but not least, my wife Shamsia Quraishi for supporting me during my training and being there for me.

Once again thanks and be safe.

Endnotes

[^1]:COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)". ArcGIS. Johns Hopkins University. Retrieved 20 June 2020.

[^2]:Logistic Growth Model for COVID-19

[^3]:Mathematical prediction of the time evolution of the COVID-19 pandemic in Italy by a Gauss error function and Monte Carlo simulations

[^4]:DeepAR Forecasting Algorithm. Amazon Web Services. Retrieved 20 June, 2020

[^5]:Time series prediction. Telesens. Retrieved 20 June, 2020.

[^6]:Time Series Forecasting with Parametric Curve Fitting

[^7]:# Logistic growth modelling of COVID-19 proliferation in China and its international implications

Covid-19 predictions using a Gauss model, based on data from April 2

[^8]:Time Series Forecasting with Parametric Curve Fitting

[^9]:Compare: 1918 Spanish Influenza Pandemic Versus COVID-19