COVID-19 Pandemic in the Philippines

About COVID-19

The COVID19 Pandemic is caused by severe acute respiratory syndrome coronavirus 2. Th e outbreak was first identified in China last December 2019. This coronavirus disease quickl y spread across 213 countries infecting millions and killing over 492k people worldwide. The outbreak of this disease has caused a major international crisis so far, and influence import ant aspects of daily life.

Main Objectives

I need a strong predictive model that will tell how the virus could spreadacross the country. The main objective of this task is to build a model using Python and Prophet that predicts th e spread of the virus in the next 180 days.

Use Case

- Develop a Model that will forecast COVID-19 cases in the next 180 days.

Tasks to be Performed

- Analyze and Explore the data (Data Wrangling was performed back-end).
- Forecast the COVID-19 cases using Prophet.

Dataset

- Dateset for building a model which is Prophet.
- https://en.wikipedia.org/wiki/COVID-19_pandemic_in_the_Philippines

Data Exploration

Statistics [edit]

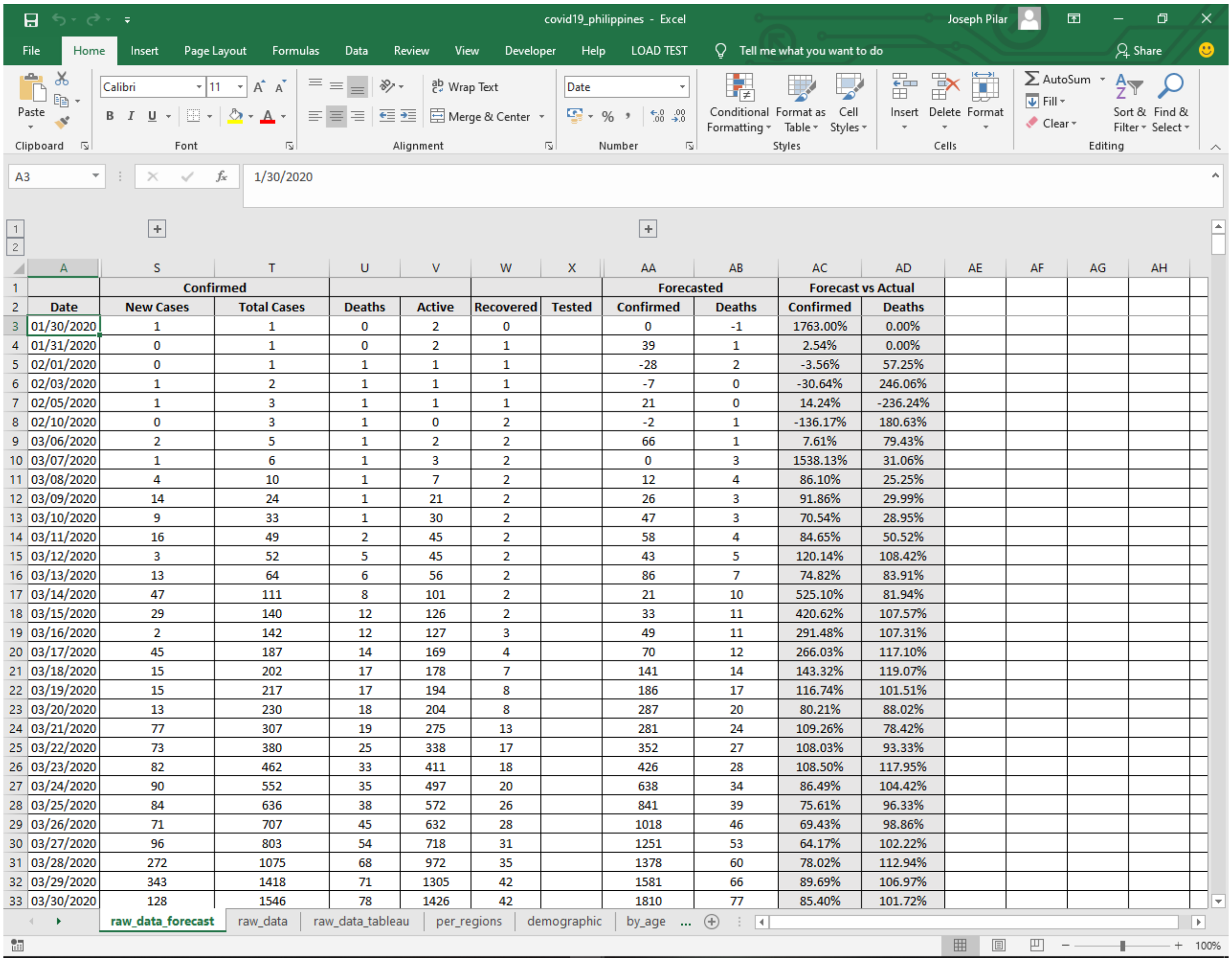
By region [edit]

Confirmed COVID-19 cases in the Philippines by region of residence (v · t · e)

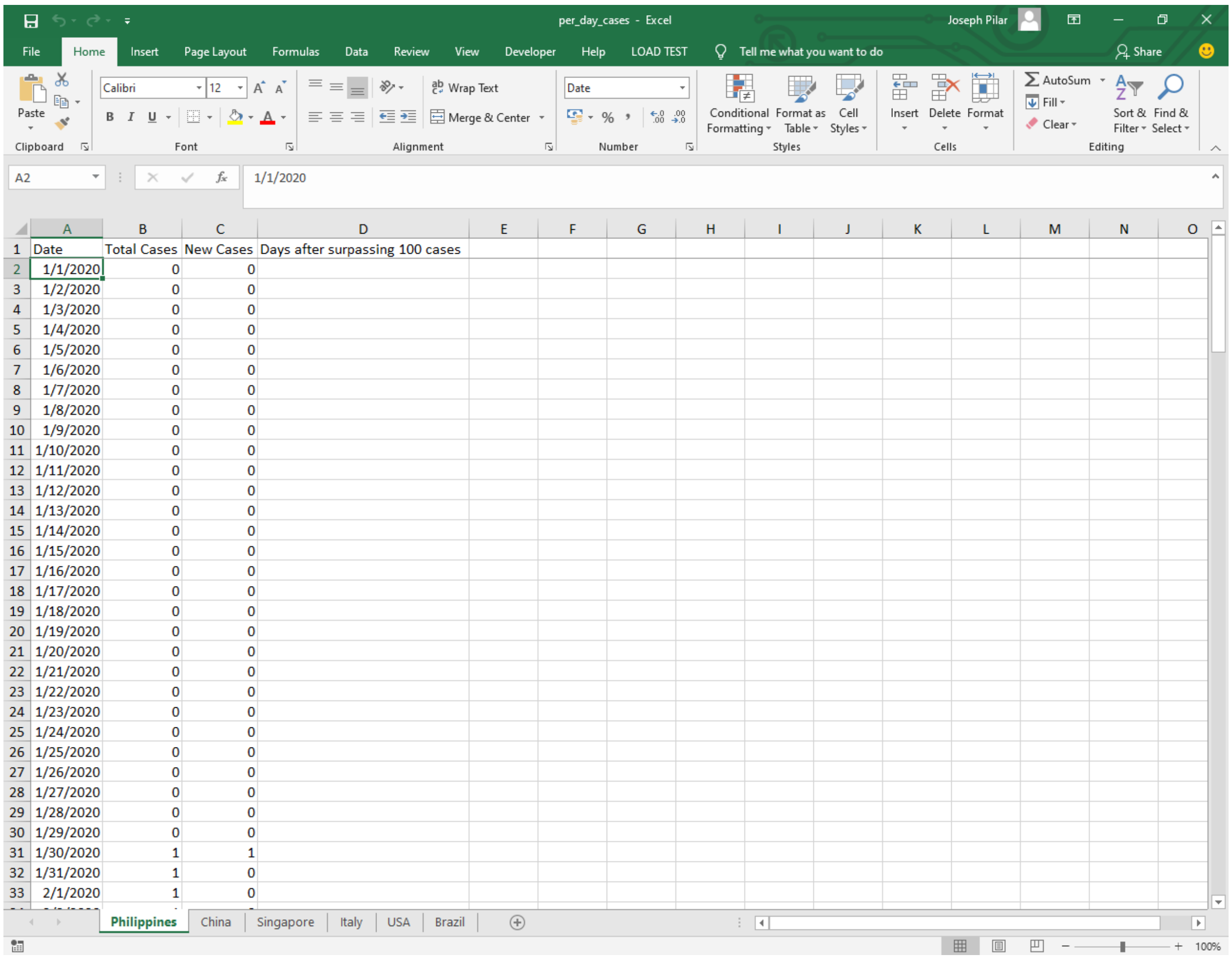
Region	Cases	Deaths		Active		Recov.		Tested	
		#	%	#	%	#	%	#	% pos.
Metro Manila	69,434	1,077	1.55	33,234	47.86	35,123	50.58	1,064,315	11.48
Cordillera	392	4	1.02	235	59.95	153	39.03	69,519	0.91
Ilocos Region	589	17	2.89	366	62.14	206	34.97	26,856	1.05
Cagayan Valley	460	2	0.43	265	57.61	193	41.96	7,329	3.64
Central Luzon	3,589	64	1.78	1,934	53.89	1,591	44.33	93,437	4.39
Calabarzon	15,451	175	1.13	9,899	64.07	5,377	34.80	57,569	8.96
Mimaropa	380	6	1.58	196	51.58	178	46.84	2,859	5.7
Bicol Region	752	7	0.93	452	60.11	293	38.96	13,918	4.31
Western Visayas	1,855	23	1.24	1,003	54.07	829	44.69	76,858	2.76
Central Visayas	17,405	720	4.14	4,207	24.17	12,478	71.69	116,090	15.55
Eastern Visayas	1,200	2	0.17	544	45.33	654	54.50	23,006	5.36
Zamboanga Peninsula	948	21	2.22	374	39.45	553	58.33	15,448	6.92
Northern Mindanao	801	12	1.50	377	47.07	412	51.44	9,580	11.35
Davao Region	1,469	51	3.47	512	34.85	906	61.67	40,476	5.21
Soccsksargen	407	3	0.74	223	54.79	181	44.47	1,681	11.36
Caraga	294	2	0.68	188	63.95	104	35.37	–	–
Bangsamoro	484	5	1.03	162	33.47	317	65.50	4,375	9.97
Repatriates	5,886	14	0.24	1,817	30.87	4,055	68.89	–	–
For validation	5,089	4	0.08	1,571	30.87	3,514	69.05	–	–
Philippines	**126,885**	**2,209**	**1.74**	**57,559**	**45.36**	**67,117**	**52.90**	**1,623,316**	**9.84**

Daily COVID-19 cases in the Philippines by region of residence (v · t · e)

Date	Regions																	Confirmed		Deaths		Active	Recov.
	NCR	CAR	I	II	III	IV-A	IV-B	V	VI	VII	VIII	IX	X	XI	XII	XIII	BAR	New	Total	New	Total	Total	Total
January 30, 2020										1								1	1	–	–	2	–
January 31, 2020																		0	1	–	–	2	1
February 1, 2020																		0	1	1	1	1	1
February 3, 2020										1								1	2	0	1	1	1
February 5, 2020										1								1	3	0	1	1	1
February 10, 2020																		0	3	0	1	0	2
March 6, 2020	1					1												2	5	0	1	2	2
March 7, 2020						1												1	6	0	1	3	2
March 8, 2020	4																	4	10	0	1	7	2
March 9, 2020	12				1	1												14	24	0	1	21	2
March 10, 2020	7				1					1								9	33	0	1	30	2
March 11, 2020	13					2				1								16	49	1	2	45	2
March 12, 2020	2					1												3	52	3	5	45	2
March 13, 2020	9				2	1												13	64	1	6	56	2
March 14, 2020	36	1			3	7												47	111	2	8	101	2
March 15, 2020	22				2	4								1				29	140	4	12	126	2
March 16, 2020	2																	2	142	0	12	127	3
March 17, 2020	38				1	5									1			45	187	2	14	169	4
March 18, 2020	11					2				1							1	15	202	3	17	178	7
March 19, 2020	13					2												15	217	0	17	194	8
March 20, 2020	13																	13	230	1	18	204	8
March 21, 2020	57		2	1	5	7	1		2					2				77	307	1	19	275	13
March 22, 2020	57				1	15												73	380	6	25	338	17
March 23, 2020	65	1			5	8								2				82	462	8	33	411	18
March 24, 2020	71				3	12					1		1				2	90	552	2	35	497	20
March 25, 2020	63				2	12	1		2			1	1		2		1	84	636	3	38	572	26
March 26, 2020	46	1	3	2	7	11												71	707	7	45	632	28
March 27, 2020	77		4	2	1	9		3	3			1		1				96	803	9	54	718	31
March 28, 2020	191		1	6	15	44	2		6					2				272	1075	14	68	972	35



	A	S	T	U	V	W	X	AA	AB	AC	AD
1		Confirmed						Forecasted		Forecast vs Actual	
2	Date	New Cases	Total Cases	Deaths	Active	Recovered	Tested	Confirmed	Deaths	Confirmed	Deaths
3	01/30/2020	1	1	0	2	0		0	-1	1763.00%	0.00%
4	01/31/2020	0	1	0	2	1		39	1	2.54%	0.00%
5	02/01/2020	0	1	1	1	1		-28	2	-3.56%	57.25%
6	02/03/2020	1	2	1	1	1		-7	0	-30.64%	246.06%
7	02/05/2020	1	3	1	1	1		21	0	14.24%	-236.24%
8	02/10/2020	0	3	1	0	2		-2	1	-136.17%	180.63%
9	03/06/2020	2	5	1	2	2		66	1	7.61%	79.43%
10	03/07/2020	1	6	1	3	2		0	3	1538.13%	31.06%
11	03/08/2020	4	10	1	7	2		12	4	86.10%	25.25%
12	03/09/2020	14	24	1	21	2		26	3	91.86%	29.99%
13	03/10/2020	9	33	1	30	2		47	3	70.54%	28.95%
14	03/11/2020	16	49	2	45	2		58	4	84.65%	50.52%
15	03/12/2020	3	52	5	45	2		43	5	120.14%	108.42%
16	03/13/2020	13	64	6	56	2		86	7	74.82%	83.91%
17	03/14/2020	47	111	8	101	2		21	10	525.10%	81.94%
18	03/15/2020	29	140	12	126	2		33	11	420.62%	107.57%
19	03/16/2020	2	142	12	127	3		49	11	291.48%	107.31%
20	03/17/2020	45	187	14	169	4		70	12	266.03%	117.10%
21	03/18/2020	15	202	17	178	7		141	14	143.32%	119.07%
22	03/19/2020	15	217	17	194	8		186	17	116.74%	101.51%
23	03/20/2020	13	230	18	204	8		287	20	80.21%	88.02%
24	03/21/2020	77	307	19	275	13		281	24	109.26%	78.42%
25	03/22/2020	73	380	25	338	17		352	27	108.03%	93.33%
26	03/23/2020	82	462	33	411	18		426	28	108.50%	117.95%
27	03/24/2020	90	552	35	497	20		638	34	86.49%	104.42%
28	03/25/2020	84	636	38	572	26		841	39	75.61%	96.33%
29	03/26/2020	71	707	45	632	28		1018	46	69.43%	98.86%
30	03/27/2020	96	803	54	718	31		1251	53	64.17%	102.22%
31	03/28/2020	272	1075	68	972	35		1378	60	78.02%	112.94%
32	03/29/2020	343	1418	71	1305	42		1581	66	89.69%	106.97%
33	03/30/2020	128	1546	78	1426	42		1810	77	85.40%	101.72%



Date	Total Cases	New Cases	Days after surpassing 100 cases
1/1/2020	0	0	
1/2/2020	0	0	
1/3/2020	0	0	
1/4/2020	0	0	
1/5/2020	0	0	
1/6/2020	0	0	
1/7/2020	0	0	
1/8/2020	0	0	
1/9/2020	0	0	
1/10/2020	0	0	
1/11/2020	0	0	
1/12/2020	0	0	
1/13/2020	0	0	
1/14/2020	0	0	
1/15/2020	0	0	
1/16/2020	0	0	
1/17/2020	0	0	
1/18/2020	0	0	
1/19/2020	0	0	
1/20/2020	0	0	
1/21/2020	0	0	
1/22/2020	0	0	
1/23/2020	0	0	
1/24/2020	0	0	
1/25/2020	0	0	
1/26/2020	0	0	
1/27/2020	0	0	
1/28/2020	0	0	
1/29/2020	0	0	
1/30/2020	1	1	
1/31/2020	1	0	
2/1/2020	1	0	

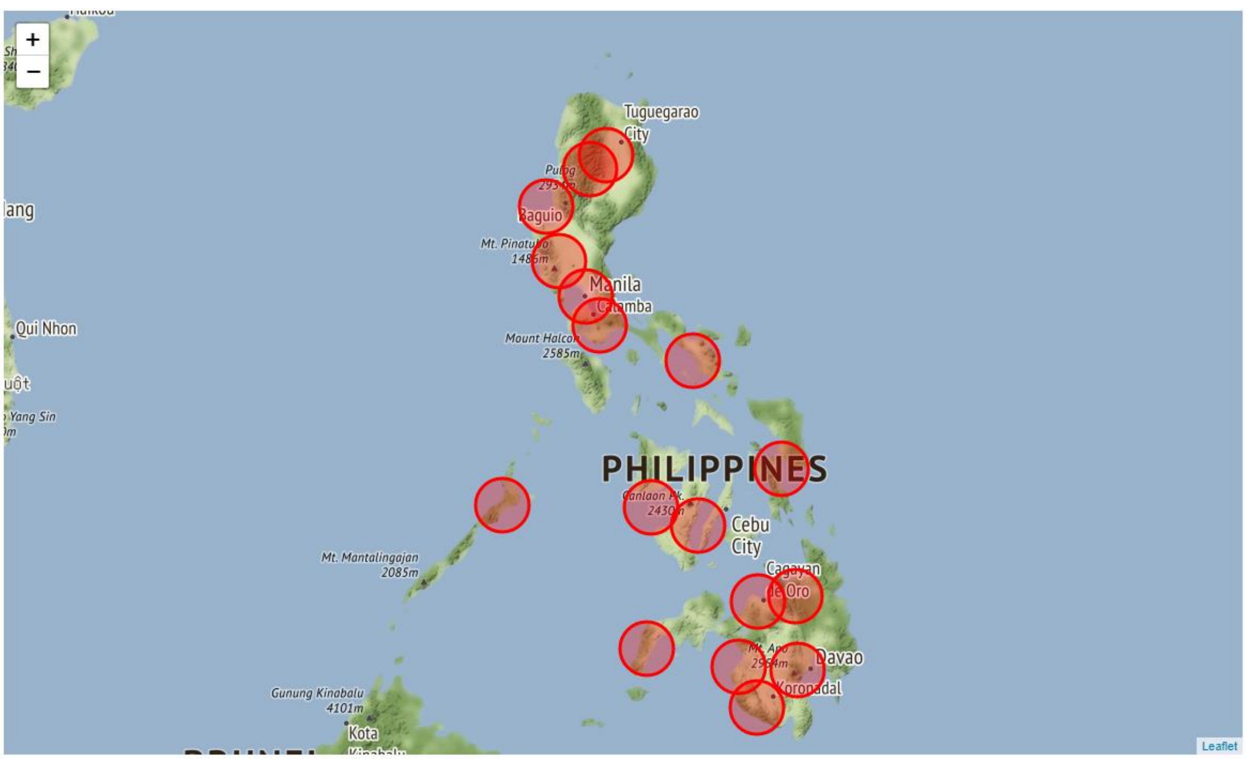
Data Cleaning

```
ds_raw = pd.read_excel('covid19_philippines.xlsx', 'raw_data', skiprows=[0])
ds_regions = pd.read_excel('covid19_philippines.xlsx', 'per_regions')
ds_demographics = pd.read_excel('covid19_philippines.xlsx', 'demographic')
ds_age = pd.read_excel('covid19_philippines.xlsx', 'by_age')
#ds_raw.drop(df_raw.iloc[:, 1:18], inplace = True, axis = 1)

ds_regions.rename(columns={'Location':'Region'}, inplace=True)
ds_raw['Date'] = ds_raw['Date'].dt.date
ds_raw.tail()
```

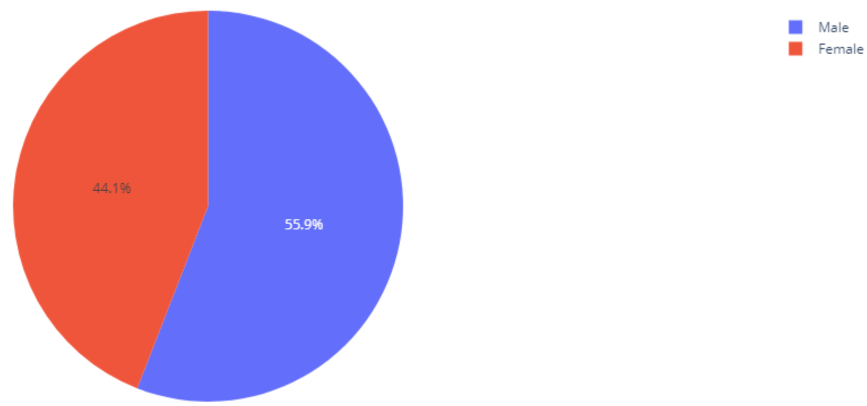
	Date	New Cases	Total Cases	Deaths	Active	Recovered	Tested
182	2020-07-01	999	38511	1270	26803.0	10438	704549.0
183	2020-07-02	294	38805	1274	26858.0	10673	720918.0
184	2020-07-03	1531	40336	1280	NaN	11073	738502.0
185	2020-07-04	1494	41830	1290	NaN	11453	NaN
186	2020-07-05	2434	44254	1297	NaN	11942	NaN

```
[3] ds_Philippines = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name='Philippines')
ds_Italy = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name="Italy")
ds_USA = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name='USA')
ds_Singapore = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name="Singapore")
ds_Brazil = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name="Brazil")
ds_China = pd.read_excel('per_day_cases.xlsx',parse_dates=True, sheet_name="China")

print(ds_regions.columns)
```

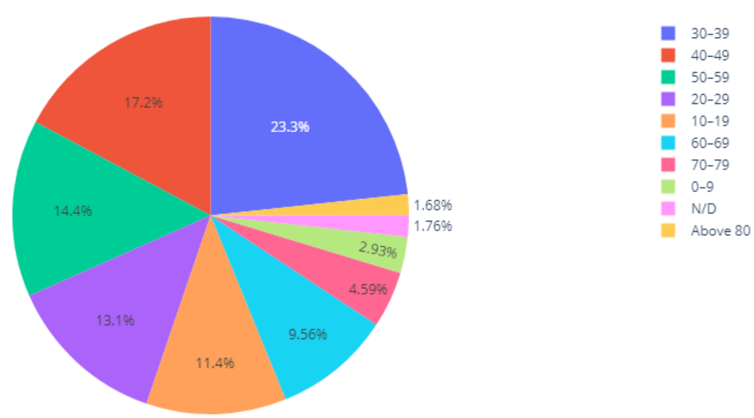
Data Visualization



Confirmed COVID-19 cases in the Philippines by Gender



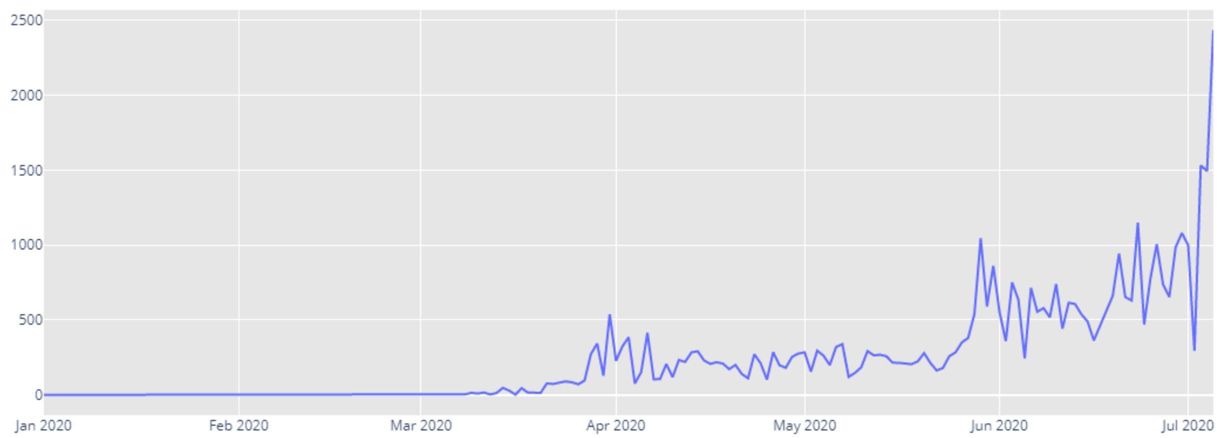
Confirmed COVID-19 cases in the Philippines by Age



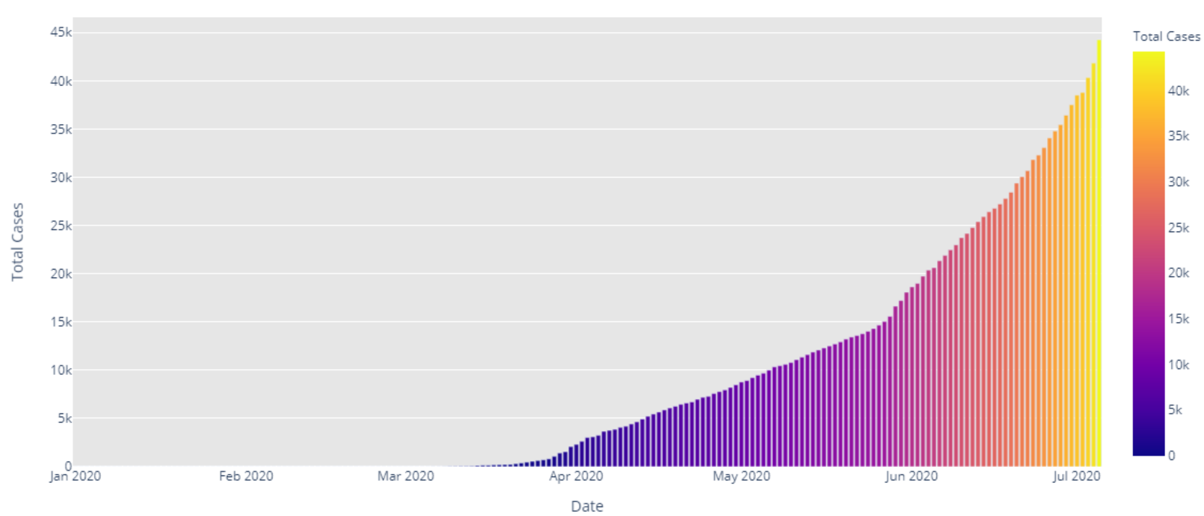
Visualization Inference

- NCR, Region VII (Central Visayas), and Region IV-A (Calabarzon) are currently the Top 3 regions with highest number of confirmed cases.
- The highest new cases listed were on July 03, 2020, with 1,531 got positive with the virus. Since Enhanced Community Quarantine was implemented in the country, The lowest new cases in a day were on March 16, 2020.
- 57.1% of total cases were from National Capital Region and 55.9% of infected patients were Male.
- 23.3% of infected patients are within the age of 30-39 years old.

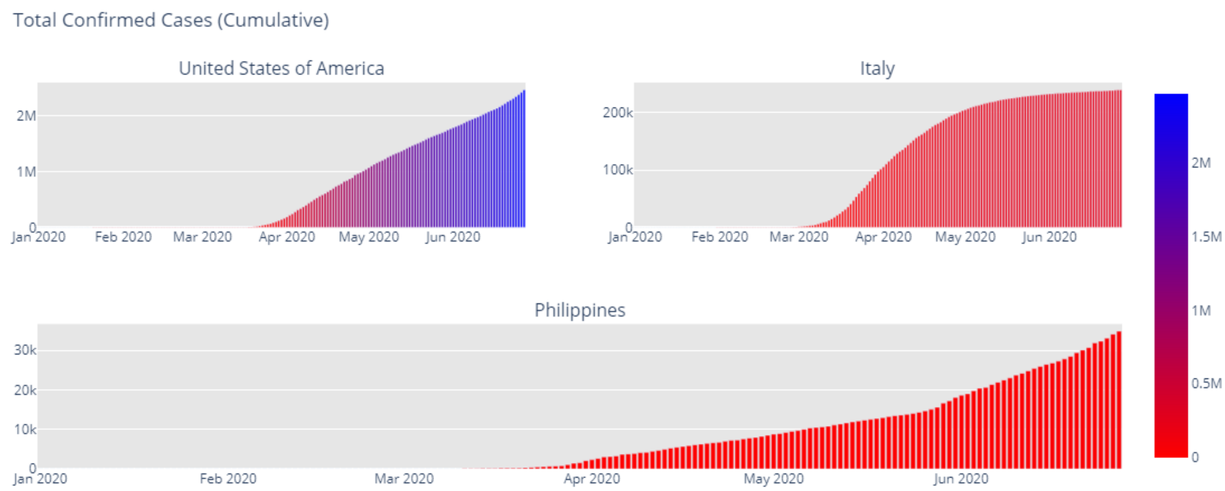
Trend of COVID-19 New Cases in the Philippines



COVID-19 Cases in the Philippines (Cumulative Cases)



Comparison between Countries



Model Definition

Prophet is an open source library published by Facebook that is based on **decomposable (trend+seasonality+holidays) models**. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters and has support for including impact of custom seasonality and holidays!

Introduction to Prophet

Understanding time based patterns is critical for any business. Questions like how much inventory to maintain, how much footfall do you expect in your store to how many people will travel by an airline – all of these are important time series problems to solve.

This is why time series forecasting is one of the must-know techniques for any data scientist. From predicting the weather to the sales of a product, it is integrated into the data science ecosystem and that makes it a mandatory addition to a data scientist's skillset.

If you are a beginner, time series also provides a good way to start working on real life projects. You can relate to time series very easily and they help you enter the larger world of machine learning.

What’s new in Prophet?

When a forecasting model doesn’t run as planned, we want to be able to tune the parameters of the method with regards to the specific problem at hand. Tuning these methods requires a thorough understanding of how the underlying time series models work. The first input parameters to automated ARIMA, for instance, are the maximum orders of the differencing, the auto-regressive components, and the moving average components. A typical analyst will not know how to adjust these orders to avoid the behaviour and this is the type of expertise that is hard to acquire and scale.

The Prophet package provides intuitive parameters which are easy to tune. Even someone who lacks expertise in forecasting models can use this to make meaningful predictions for a variety of problems in a business scenario.

The Prophet Forecasting Model

We use a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- **g(t)**: piecewise linear or logistic growth curve for modelling non-periodic changes in time series
- **s(t)**: periodic changes (e.g. weekly/yearly seasonality)
- **h(t)**: effects of holidays (user provided) with irregular schedules
- **ε_t**: error term accounts for any unusual changes not accommodated by the model

Using time as a regressor, Prophet is trying to fit several linear and non linear functions of time as components. Modeling seasonality as an additive component is the same approach taken by exponential smoothing in [Holt-Winters technique] . We are, in effect, framing the forecasting problem as a curve-fitting exercise rather than looking explicitly at the time based dependence of each observation within a time series.

Trend

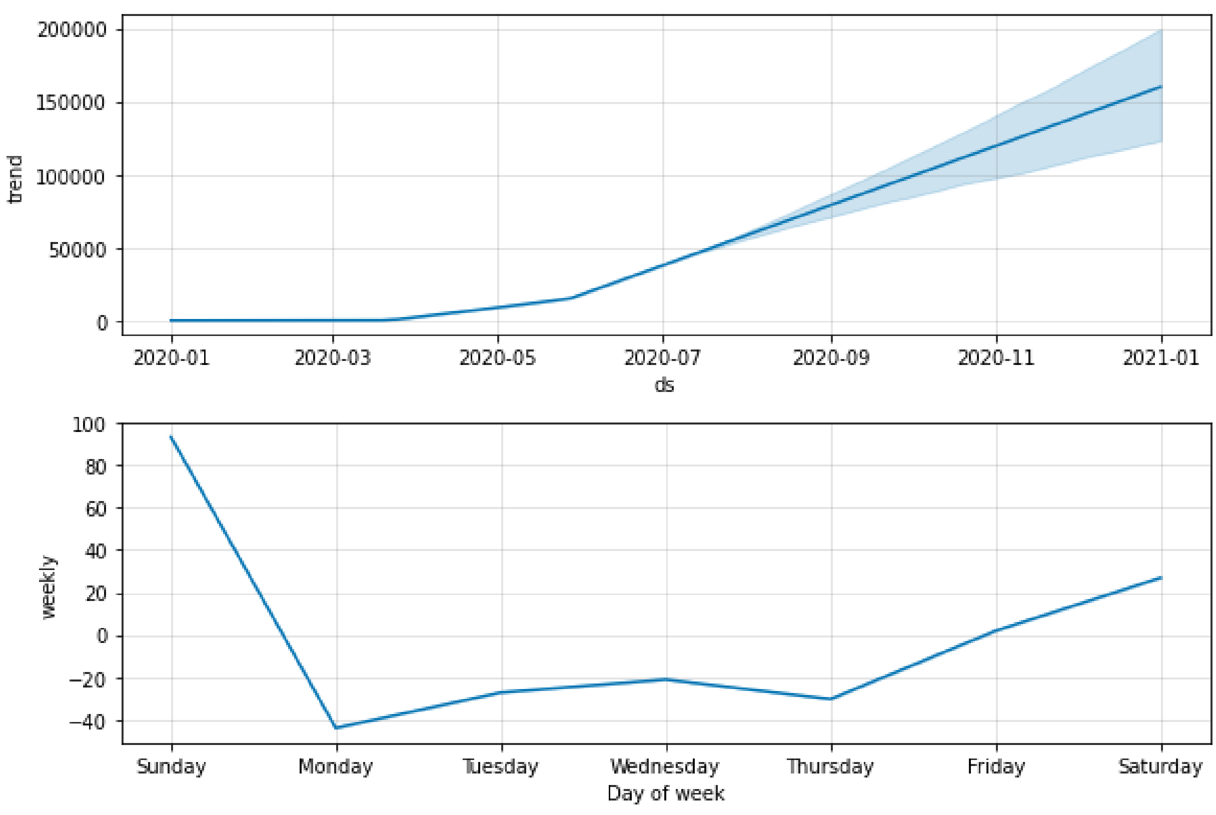
Trend is modelled by fitting a piece wise linear curve over the trend or the non-periodic part of the time series. The linear fitting exercise ensures that it is least affected by spikes/missing data.

Saturating growth

An important question to ask here is – Do we expect the target to keep growing/falling for the entire forecast interval?

More often than not, there are cases with non-linear growth with a running maximum capacity. I will illustrate this with an example below.

Let’s say we are trying to forecast number of downloads of an app in a region for the next 12 months. The maximum downloads is always capped by the total number of smartphone users in the region. The number of smartphone users will also, however, increase with time.



Changepoints

Another question to answer is whether my time series encounters any underlying changes in the phenomena e.g. a new product launch, unforeseen calamity etc. At such points, the growth rate is allowed to change. These changepoints are automatically selected. However, a user can also feed the changepoints manually if it is required. In the below plot, the dotted lines represent the changepoints for the given time series.

As the number of changepoints allowed is increased the fit becomes more flexible. There are basically 2 problems an analyst might face while working with the trend component:

- Overfitting
- Underfitting

A parameter called changepoint_prior_scale could be used to adjust the trend flexibility and tackle the above 2 problems. Higher value will fit a more flexible curve to the time series.

Seasonality

To fit and forecast the effects of seasonality, prophet relies on fourier series to provide a flexible model. Seasonal effects s(t) are approximated by the following function:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

P is the period (365.25 for yearly data and 7 for weekly data)

Parameters [a_1, b_1, ….., a_N, b_N] need to be estimated for a given N to model seasonality.

The fourier order N that defines whether high frequency changes are allowed to be modelled is an important parameter to set here. For a time series, if the user believes the high frequency components are just noise and should not be considered for modelling, he/she could set the values of N from to a lower value. If not, N can be tuned to a higher value and set using the forecast accuracy.

Holidays and events

Holidays and events incur predictable shocks to a time series. For instance, Diwali in India occurs on a different day each year and a large portion of the population buy a lot of new items during this period.

Prophet allows the analyst to provide a custom list of past and future events. A window around such days are considered separately and additional parameters are fitted to model the effect of holidays and events.

Prophet in action (using Python)

Currently implementations of Prophet are available in both Python and R. They have exactly the same features.

Prophet() function is used do define a Prophet forecasting model in Python. Let us look at the most important parameters:

3.1 Trend parameters

Parameter	**Description**
growth	linear' or 'logistic' to specify a linear or logistic trend
changepoints	List of dates at which to include potential changepoints (automatic if not specified)
n_changepoints	If changepoints in not supplied, you may provide the number of changepoints to be automatically included
changepoint_prior_scale	Parameter for changing flexibility of automatic changepoint selection

3.2 Seasonality & Holiday Parameters

Parameter	Description
yearly_seasonality	Fit yearly seasonality
weekly_seasonality	Fit weekly seasonality
daily_seasonality	Fit daily seasonality
holidays	Feed dataframe containing holiday name and date
seasonality_prior_scale	Parameter for changing strength of seasonality model
holiday_prior_scale	Parameter for changing strength of holiday model

Model Training

Import necessary packages and reading dataset:

```
[27] from fbprophet import Prophet
```

```
confirmed = ds_raw.groupby('Date').sum()['Total Cases'].reset_index()
new = ds_raw.groupby('Date').sum()['New Cases'].reset_index()
deaths = ds_raw.groupby('Date').sum()['Deaths'].reset_index()
recovered = ds_raw.groupby('Date').sum()['Recovered'].reset_index()
```

```
[ ] confirmed.columns = ['ds','y']
    confirmed['ds'] = pd.to_datetime(confirmed['ds'])
    confirmed.tail()
```

	ds	y
182	2020-07-01	38511
183	2020-07-02	38805
184	2020-07-03	40336
185	2020-07-04	41830
186	2020-07-05	44254

```
[ ] model_total_cases = Prophet(interval_width=0.95)
    model_total_cases.fit(confirmed)
    future_total_cases = model_total_cases.make_future_dataframe(periods=180)
    future_total_cases.tail()
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

	ds
362	2020-12-28
363	2020-12-29
364	2020-12-30
365	2020-12-31
366	2021-01-01

Prophet requires the variable names in the time series to be:

- y – Target
- ds – Datetime

So, the next step is to convert the dataframe according to the above specifications

Fitting the prophet model:

```
model_total_cases = Prophet(interval_width=0.95)
model_total_cases.fit(confirmed)
future_total_cases = model_total_cases.make_future_dataframe(periods=180)
future_total_cases.tail()
```

We can look at the various components using the following command:

```
forecast_total_cases = model_total_cases.predict(future_total_cases)
forecast_total_cases.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	wee
362	2020-12-28	157350.663148	121331.476022	195784.629715	121627.493060	195661.307631	-43.899821	-43.899821	-43.899821	-43.899
363	2020-12-29	158014.942002	121908.802145	196753.947873	121998.456750	196682.626898	-27.150007	-27.150007	-27.150007	-27.150
364	2020-12-30	158679.220856	122101.473005	197720.103875	122386.336712	197685.956205	-21.037690	-21.037690	-21.037690	-21.037
365	2020-12-31	159343.499710	122935.141070	197919.636094	122846.194526	198637.432088	-30.240056	-30.240056	-30.240056	-30.240
366	2021-01-01	160007.778565	123183.299945	199135.280965	123260.266262	199588.907972	1.932663	1.932663	1.932663	1.932

Model Evaluation

Prophet certainly is a good choice for producing quick accurate forecasts. It has intuitive parameters that can be tweaked by someone who has good domain knowledge but lacks technical skills in forecasting models. Readers can also try and fit Prophet directly over the hourly data and discuss in the comments if they are able to get a better result.