COVID-19 Pandemic Prediction

Submitted in partial fulfilment of the requirements

for the award of the degree of Bachelor of Computer Applications

Guide:
Er. Raveendra Nath Pathak

Submitted by:
Ranjeet Kumar Shrivastav

RollNo. 170110330035

Submitted to

Department of Computer science

STUDY CENTRE

S.S. Jeena Campus, Almora

ACKNOWLEDGEMENTS

In the “Acknowledgements” page, the writer recognizes his indebtedness for guidance and assistance of the thesis adviser and other members of the faculty. Courtesy demands that he also recognizes specific contributions by other persons or institutions such as libraries and research foundations. Acknowledgements should be expressed simply, tastefully, and tactfully.

CERTIFICATE

This is to certify that this project entitled “COVID-19 Pandemic Prediction” submitted in partial fulfillment of the degree of Bachelor of Computer Applications to the “Ranjeet Kumar Shrivastav” through S. S. Jeena Campus, Almora, done by Mr. Ranjeet Kumar Shrivastav, Roll No. 170110330035 is an authentic work carried out by him under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Signature of the student

Signature of the Guide

SELF CERTIFICATE

This is to certify that the dissertation/project report entitled "COVID-19 Pandemic Prediction" is done by me is an authentic work carried out for the partial fulfillment of the requirements for the award of the degree of Bachelor of Computer Applications under the guidance of Er. Raveendra Nath Pathak. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

(signature of the student)
Ranjeet Kumar Shrivastav
Roll No.170110330035
S. S. Jeena campus, ALMORA

TABLE OF CONTENTS

- Haryana
- Jammu & Kashmir
- Karnataka
- Kerala
- Maharashtra
- Madhya Pradesh
- Punjab
- Rajasthan
- Tamil Nadu
- Telangana
- Uttar Pradesh
- West Bengal

4.1.2. USA

➢ USA prediction by specific state

- Alabama
- Arizona
- California
- Colorado
- Florida
- Georgia
- Illinois
- New Jersey

- New York
- Washington

5. Coding

6. Testing

6.1. Introduction to Testing

6.1.1. White Box Testing

6.1.2. Black Box Testing

6.2 Testing Approach

7. Conclusion

1. Introduction

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

The best way to prevent and slow down transmission is be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using and alcohol based rub frequently and not touching your face.

The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it's important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow).

At this time, there are no specific vaccines or treatments for COVID-19. However, there are many ongoing clinical trials evaluating potential treatments. WHO will continue to provide updated information as soon as clinical findings become available.

1.1 Overview of Project

The project entitled "COVID-19 Pandemic Prediction" id designed and developed using jupyter notebook and python as a programming language. The project showcases how this virus looks like in the coming months. This prediction depends on the collected data from different sources and changes accordingly.

1.2 Objectives

To study what numbers, look like in the coming future as this virus spread more and more or how the number change when safety measures taken by the government of each country.

2. Exploratory data analysis (EDA)

First of all, let's take a look on the data structure:

	Id	Province_State	Country_Region	Date	ConfirmedCases	Fatalities
0	1	None	Afghanistan	2020-01-22	0.0	0.0
1	2	None	Afghanistan	2020-01-23	0.0	0.0
2	3	None	Afghanistan	2020-01-24	0.0	0.0
3	4	None	Afghanistan	2020-01-25	0.0	0.0
4	5	None	Afghanistan	2020-01-26	0.0	0.0

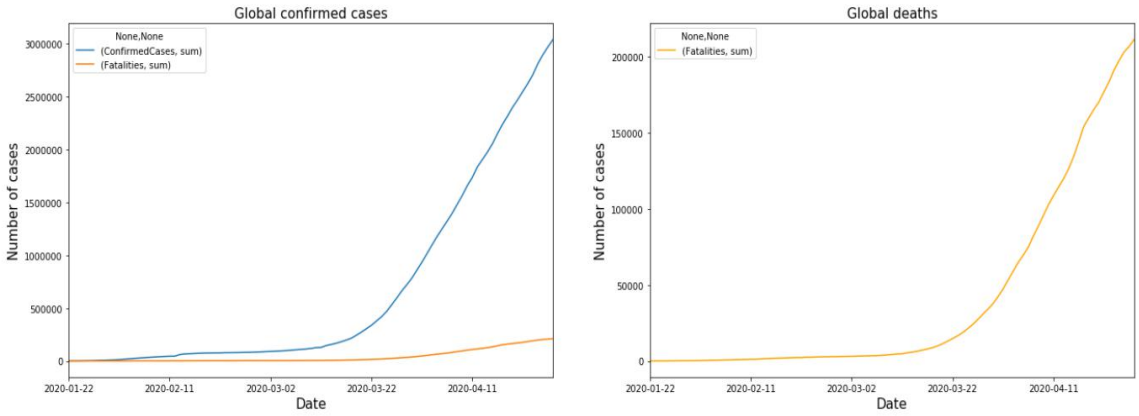
	Id	ConfirmedCases	Fatalities
count	26292.000000	26292.000000	26292.000000
mean	17826.500000	1186.183896	62.265594
std	10300.678012	8549.128727	695.049077
min	1.000000	0.000000	0.000000
25%	8913.750000	0.000000	0.000000
50%	17826.500000	1.000000	0.000000
75%	26739.250000	117.000000	1.000000
max	35652.000000	203020.000000	21067.000000

```
number of country_region:  184
dates go from day 2020-01-22 to day 2020-04-20 ,a total of  90 days
countries with state informed:  ['Australia' 'Canada' 'China' 'Denmark' 'France' 'Netherlands' 'US'
 'United Kingdom']
```

The dataset covers 163 countries and almost 3 full months from 22 jan,2020,

Which is enough data to get some clues about the pandemic.

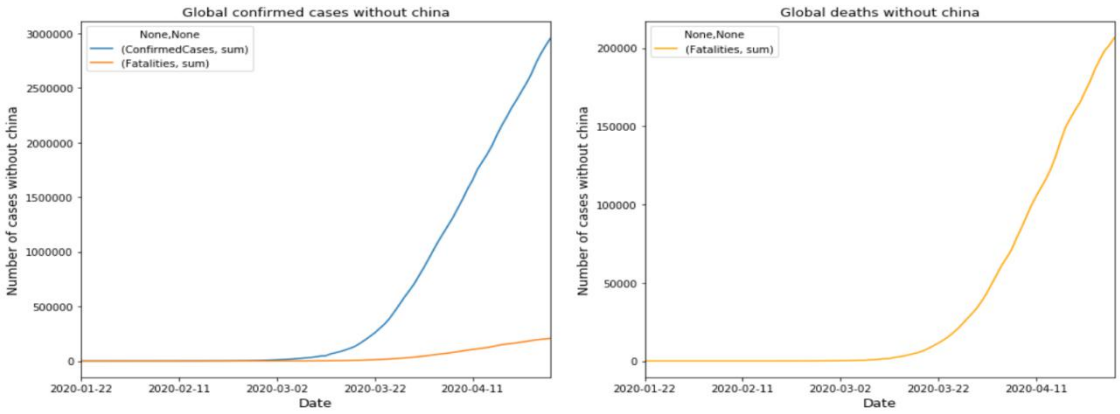
Let’s see a few plots of the worldwide tendency to see if we can extract some insights:



The global curve shows a rich fine structure, but these numbers are strongly affected by the vector zero country, China. Given that COVID-19 started there, during the initial expansion of the virus there was no reliable information about the real infected cases. In fact, the criteria to consider infection cases was modified around 02-01-2020, which strongly perturbed the curve as you can see from the figure.

2.1 COVID-19 global tendency excluding China:

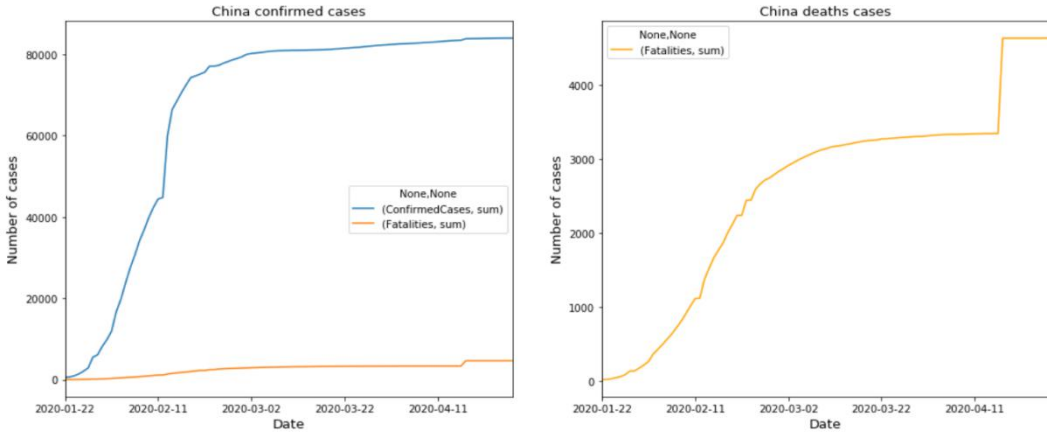
Since details of the initial breakthrough strongly interfere with results, it's recommended to analyze China independently. Let's first see the results Without China:



In this case the general behavior looks cleaner, and in fact the curve resembles a typical epidemiology model like SIR. SIR models present a large increasing in the number of infections that, once it reaches the maximum of the contagion, decreases with a lower slope. The SIR models the flows of people between three states: Susceptible (S), Infected (I), and Resistant (R). The SIR model is used where individuals infect each other directly. The rate that people become infected is proportional to the number of people who are infected, and the number of people who are susceptible. If there are lots of people infected, the chances of a susceptible coming into contact with someone who is infected is high. Likewise, if there are very few people who are susceptible, the chances of a susceptible coming into contact with an infected is lower. And I think that's why WHO and government keep focusing on social distancing.

2.2. COVID-19 tendency in China

Since China was the initial infected country, the COVID-19 behavior is different from the rest of the world. The medical system was not prepared for the pandemic, in fact no one was aware of the virus until several cases were reported. Moreover, China government tool strong contention measures in a considerable short period of time and, while the virus is widely spread, they have been able to control the increasing of the infections.



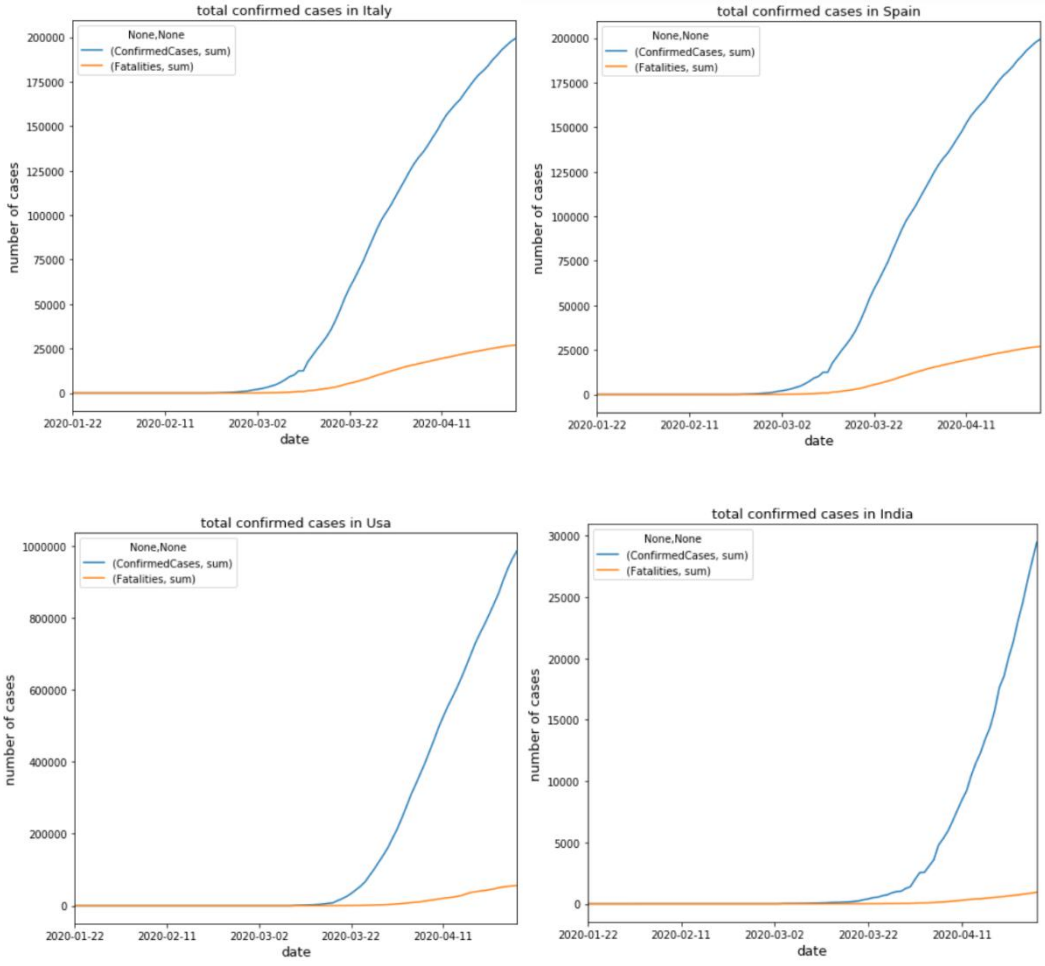
Both plots are less smooth than the curve from the rest of the world cumulative, The moment in which the criteria to consider an infected case was changed is directly spotted.

There are some irregularities, but the reasons may be that both the resources spent to monitor the pandemic and the security measures to stop or have been changing over time.

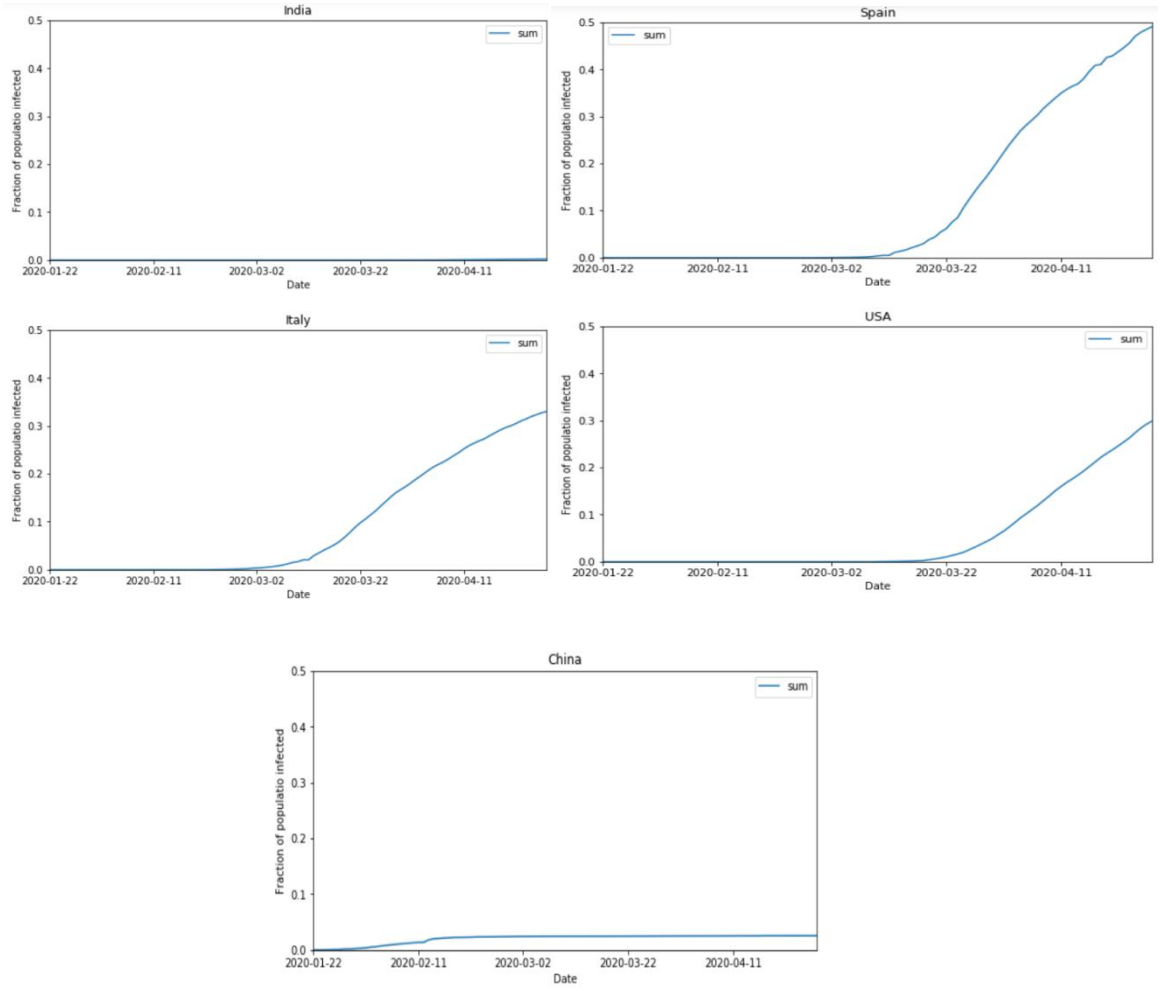
It looks like the curve has reached a plateau, which would imply that China is on their maximum of contagion.

2.3. India, Italy, Spain and USA

Both Italy and Spain are experiencing the larger increase in COVID-19 positives in Europe. The third country we will study in this section is USA, as of today USA has become the next hotspot nation after china, and the fourth country is India, as the 2nd largest population country, India's action against COVID-19 may impact dramatically on global slop.

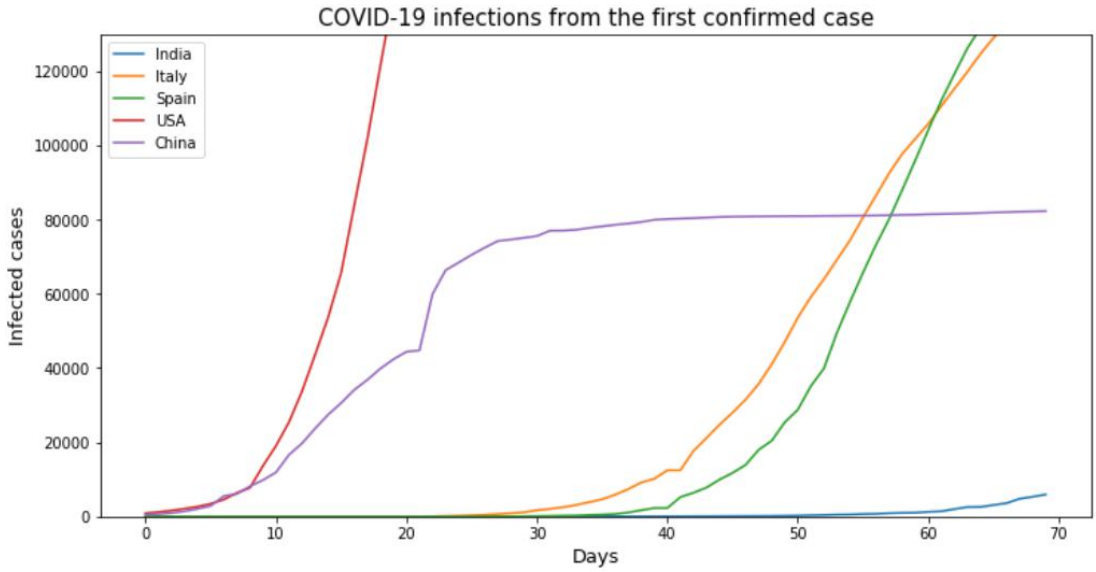


As a fraction of the total population of each country:



in the case of the world's highest population countries India and China, the cases might look lower in comparison to other countries, both Italy and Spain took major damage due to this pandemic and the USA has faced a serious increase in cases.

In order to compare the 5 countries, it’s also interesting to see the evolution of the infections from the first confirmed case:



Observations:

- **India.** India shows less cases compare to other countries. This may be due to nation-wide lockdown but the cases keep increasing day by day. The number of cases is around 26200 on 2-05-2020, this is a small fraction of its population.
- **Italy.** With almost 205,463 confirmed cases, Italy shows one of the most alarming scenarios of COVID-19. The infections curve is very steep, and more than 0.3% of population has been infected.
- **Spain.** Spain has slightly increase in number of cumulative infected cases than Italy, near 239,639. However, Spain's total population is lower (around 46 millions) and hence the percentage of population has been infected rises up to 0.5%.
- **USA.** With 1M confirmed cases, USA crosses Spain on total confirmed cases, and shows one of the most alarming scenarios of COVID-19, and the number keep rising day by day. USA face a sudden spike on its COVID-19 cases and its keep increasing.

Design

3.1 Requirement Analysis

Application Requirements

For this, I used jupyter notebook. Its written in python programming language. It supports windows, Linux.

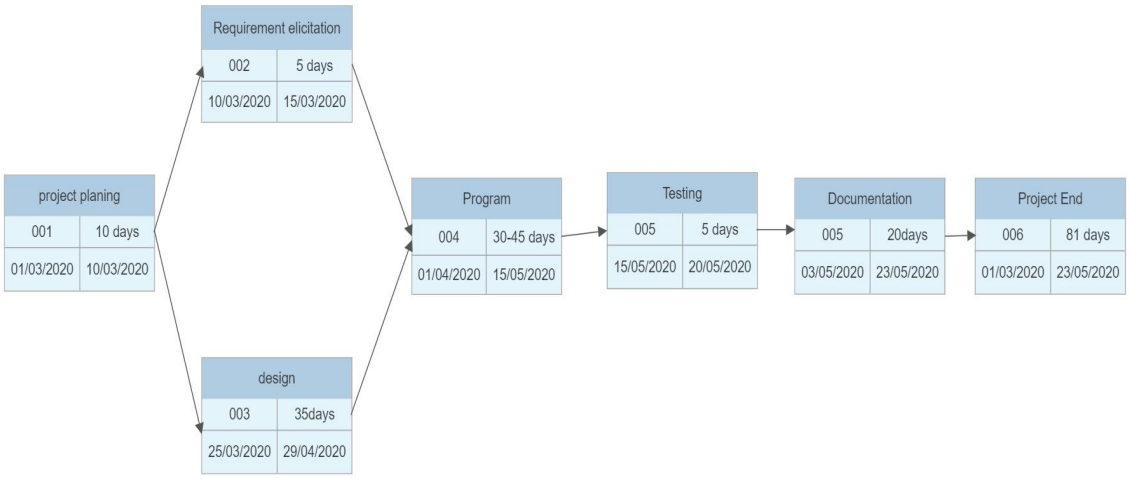
Hardware Requirements

RAM	512MB or higher
Hard Disk	10GB or higher
Computer Processor	any intel or ryzen processors

3.2 Pert Chart

A PERT chart is a project management tool that provides a graphical representation of a project's timeline. The program Evaluation Review Technique (PERT) breaks down the individual tasks of a project for analysis. PERT charts are considered preferable to Gantt charts because they identify task dependencies, but they're often more difficult to interpret.

PERT chart were first created by the U.S. navy's special projects office in 1957 to guide the polaris nuclear submarine project. A pert chart uses circles or rectangles called nodes to represent project events or milestones. These nodes are linked by vectors, or lines, that represent various tasks. A pert chart allows managers to evaluate the time and resources necessary to manage a project.

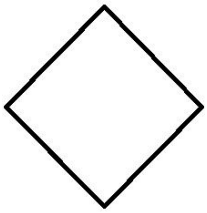


3.3 Data Flow Diagram

The dataflow diagram (DFD) is the one of the most important modeling tools. It shows the use of the data pictorially. DFD represents the flow of the data between different transformations and process in the system. The dataflow shows logical flow of the data. Different notations used in DFD are;

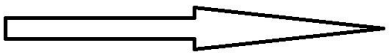
Functional Processing

It is represented by an diamond. This notation specifies the processing or main transactions.



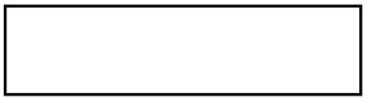
Data Flow

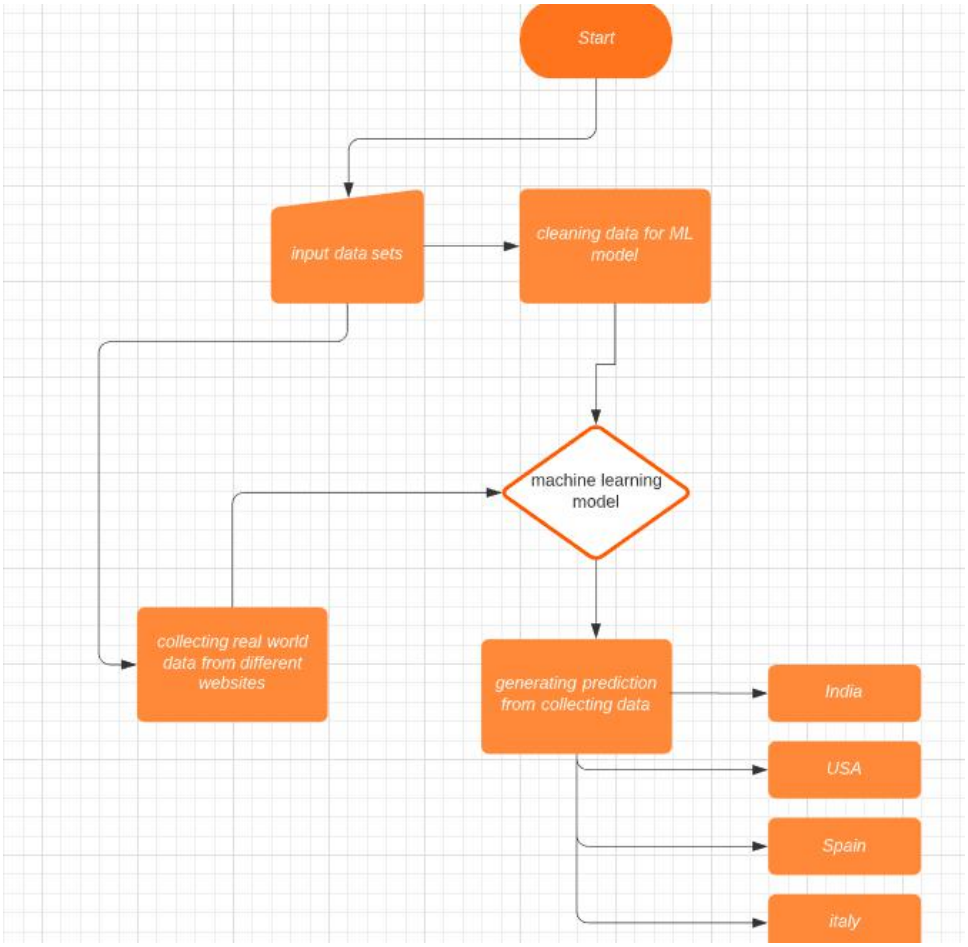
An arrow line represents it and name of the data is specified by the side of the line as label. This arrow line is used to indicate data movement.



Source or Sink

It is represented by a rectangle. It is used for specifying from where data is coming and where it will reach.





Data Flow Diagram

3. Model

I will use a model from a marketing paper by Emmanuelle Le Nagard and Alexandre Steyer, that attempts to reflect the social structure of a diffusion process. Their application was the diffusion of innovations, not epidemics. However, there are commonalities in both domains, as the number of contacts each infected person / innovation adopter has seems relevant. It also has the added benefit to allow fitting parameters to the beginning of a time series.

The model is also sensitive to when we define the origin of time for the epidemic process. Here, I just look the first point of the time series available, but adding a lag parameter could be attempted.

We need to explore the 3d parameter space to find a minimum, using gradient descent. There are a number of algorithms to do that in scipy.optimize, I stopped at the first one that seemed to work.

3.1. Diffusion Model

The model is from a marketing paper by Emmanuelle Le Nagard and Alexandre Steyer, that attempts to reflect the social structure of a diffusion process.

The essential idea here is the diffusion formula for product innovation and applying that to viral infection. We can build two models using the same expression: one for confirmed cases, one for fatalities(deaths). Here is the formula.

$$N(1-e^{-a(t-t_0)})^{\alpha}$$

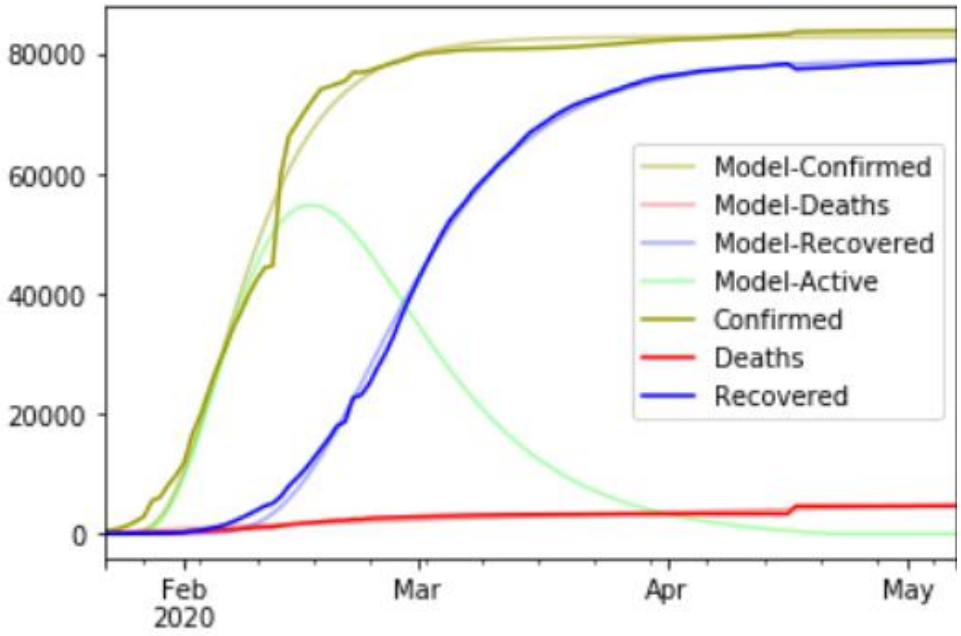
The main difference in our implementation is that we are setting N = 1. here is the justification.

- When t = 0 the value of this expression is 0, i.e. zero infections or deaths.
- When t is extremely large this expression converges to a stable value that is N if α = 1, or a stable value that is a fraction of N if α > 1. Note that α cannot be less than 1.
- N is the population. In our case the population of world or different countries.

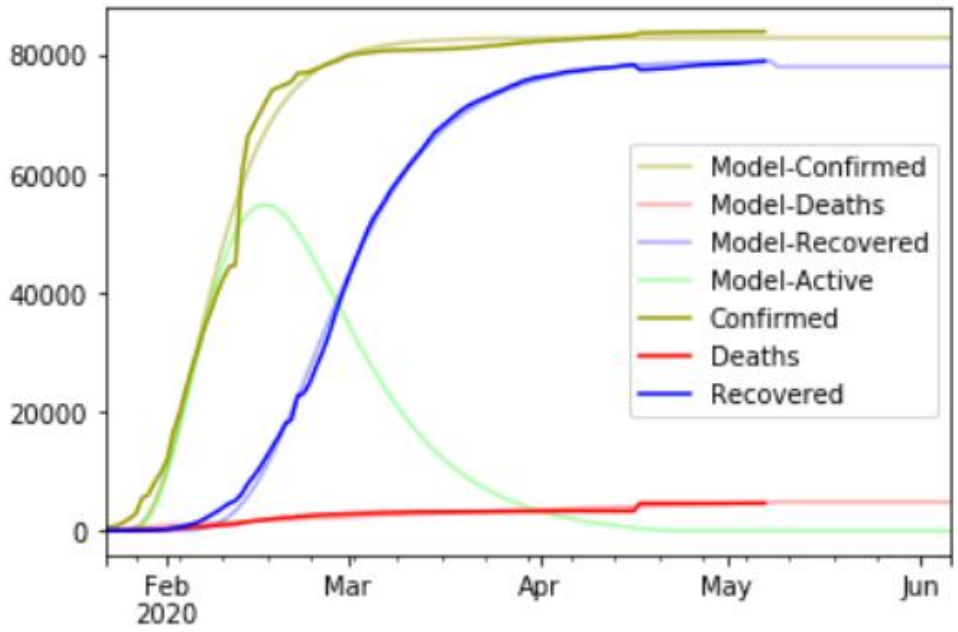
3.2. Model testing

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system, (in this case, Model) is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements.

In this case, we need to test our model to check if it generates desire results or not:



Curve looks perfect, Let's extend the prediction curve using china data:



Looks like no problem, let’s build the model.

4. Predictions

The spread of COVID-19 in the world has put the humanity at risk. The resources of some of the largest economies are stressed out due to the large infectivity and transmissibility of this disease. Due to the growing magnitude of number of cases and its subsequent stress on the administration and health professionals, some prediction methods would be required to predict the number of cases in future.

4.1 predict by specify country

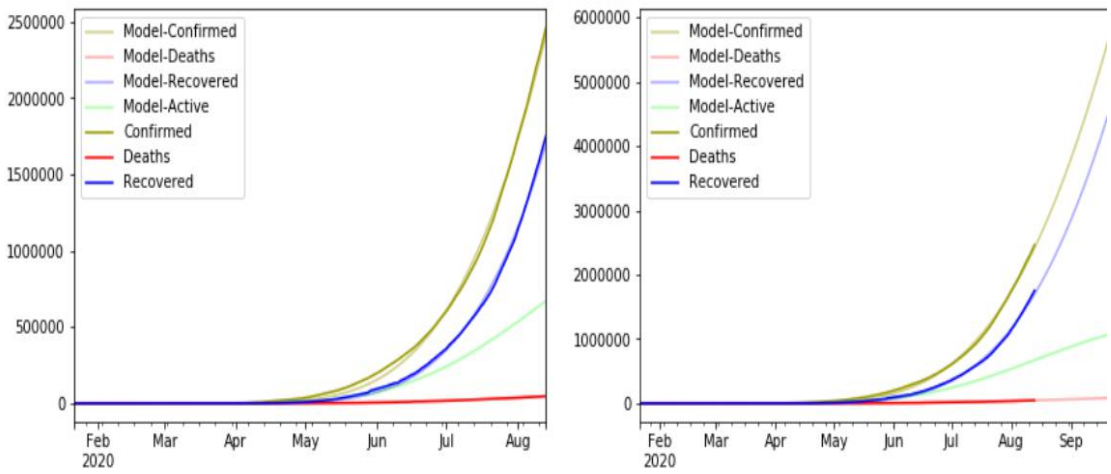
In this, we predict COVID-19 growth in most suffer countries. we analyze how numbers look like in a few months. so, let's get started

4.1.1. India

The first case of COVID-19 was reported in India on 30th January 2020 with origin from china. It spreads to the maximum of districts of the country. As on 9th April 2020 the total cases reported in India are 5734 with 472 recoveries and 199 deaths. However, the rate of infection is lower as compared to other countries.

There is a lot of stress on the part of administration and health officials for accommodating patients with possible symptoms of COVID-19. So, for that some prediction tools must be used to know about the number of cases in coming months.

India COVID-19 Prediction



The data has been used from an online site that keeps updating their data as the new cases emerge. The first graph showing the current number of confirmed cases, deaths, and recovered cases. The second graph showing model-confirmed, model-deaths and model-recovered that represent data in coming months. In this figure, the official data (darker color) indicates the official data available and forecasted data (lighter color) indicates the forecast of a total number of confirmed cases, deaths and recovered cases. From this graph, it is observed that the forecasted number of total confirmed positive cases closely matches with the available official data. The recovery rate of confirmed cases is also high in case of COVID-19, however, the time taken for the patients to recover is also large. With many patients, the stress on the medical resource increases, so estimation/prediction of time taken for recovery is also required for proper arrangement and utilization of available resources.

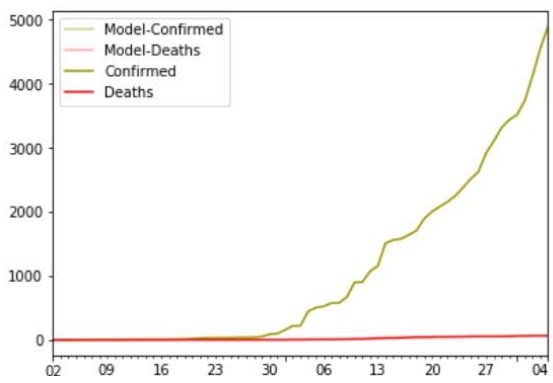
With the outbreak of the COVID-19 pandemic, various measures are has been adopted by the Govt. of India to prevent its spread. One of the measures is social isolation and lockdown. Social isolation is the complete lack of contact between an individual and society, while, lockdown is an emergency protocol that usually prevents people from

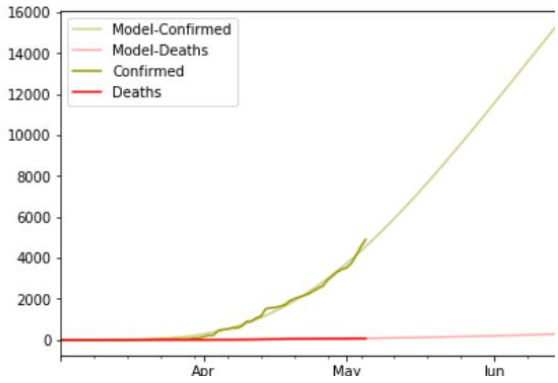
leaving an area. These two measures prevent the spread of COVID-19 from effected person to healthy individual to a great extent.

India's prediction by specific state

- Delhi

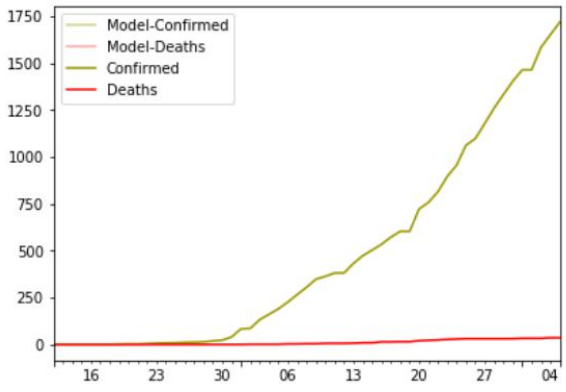
The first case of the COVID-19 pandemic in the Indian capital Delhi was reported on 2 march 2020. Delhi has the third highest number of confirmed cases of COVID-19 pandemic in India after Maharashtra and Tamil Nadu. The first case of corona virus at Delhi was confirmed on 2 march when a 45 years old person from east Delhi with a history of travel from Italy has been tested to positive for COVID-19. In May, the total number of confirmed cases cross 4500 mark, and as the model suggested it keeps increasing day by day.

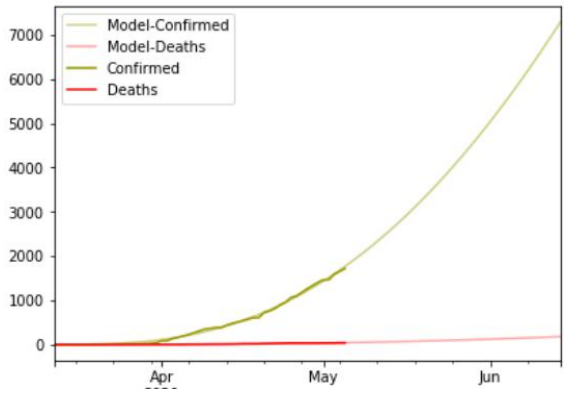




- Andhra Pradesh

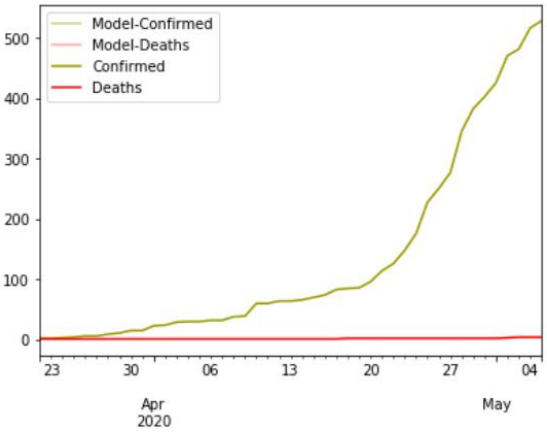
The first case of the COVID-19 pandemic in the Indian state of Andhra Pradesh was reported in Nellore on 12 march 2020. A 24-year-old who was confirmed positive for coronavirus, was also its first victim. He had travel history to Italy. In May, the total number of confirmed cases cross 1600 mark, and as the model suggested it keeps increasing day by day.

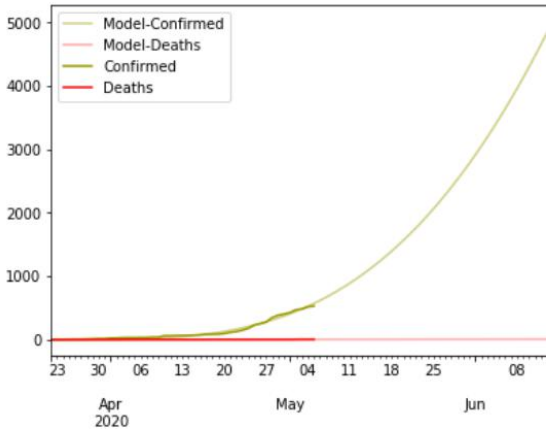




- Bihar

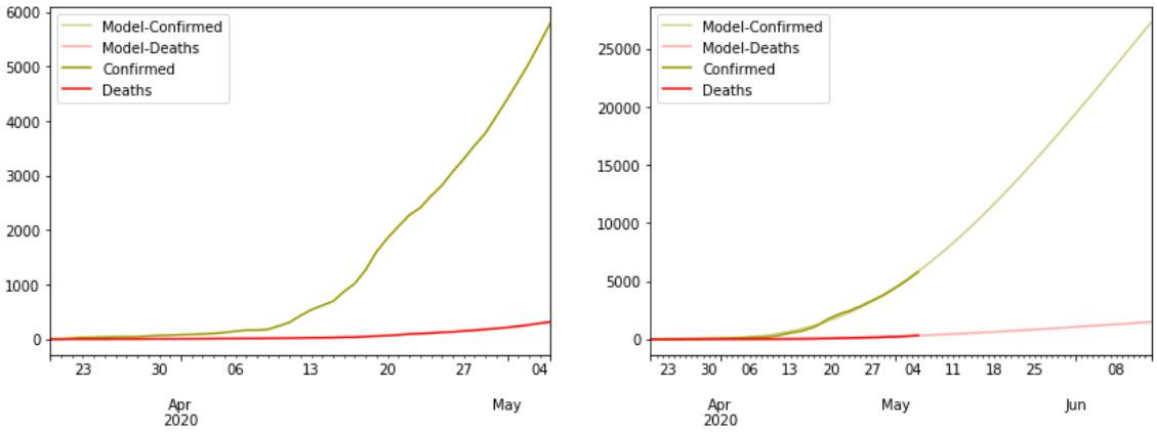
The first case of the COVID-19 pandemic in the Indian state of Bihar was reported in Munger on 22 March 2020. A 38-year-old who was confirmed positive for Corona virus, was also its first victim. He had travel history to Qatar. In May, the total number of confirmed cases cross 550 mark, and as the model suggested it keeps increasing day by day.





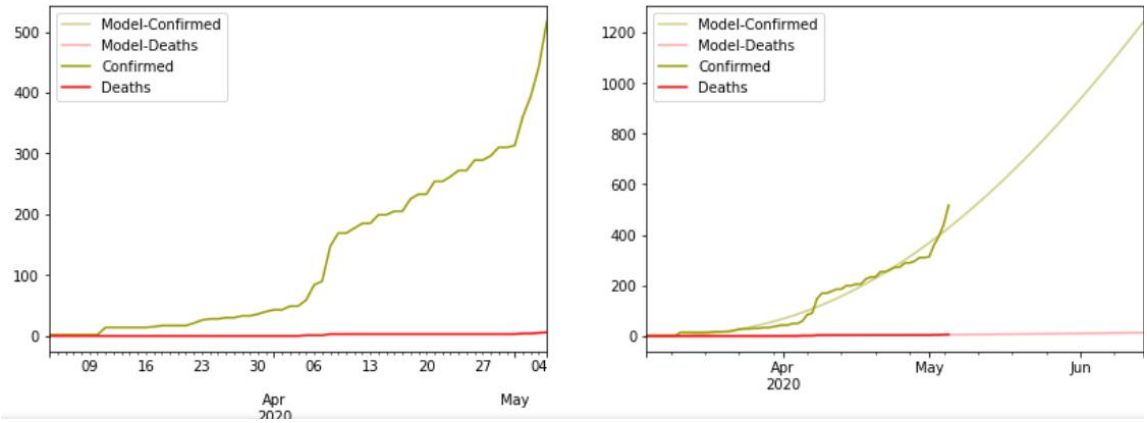
- Gujarat

In Gujarat, the first two cases of the COVID-19 were confirmed on 19 March 2020 in Rajkot and Surat. Gujarat reports first two cases of COVID-19. A 32-year=old man from Rajkot, who returned from Saudi Arabia, and a 21-year-old woman from Surat, who returned from UK, were tested positive. In May, the total number of confirmed cases cross 5500 mark, and as the model suggested it keeps increasing day by day.



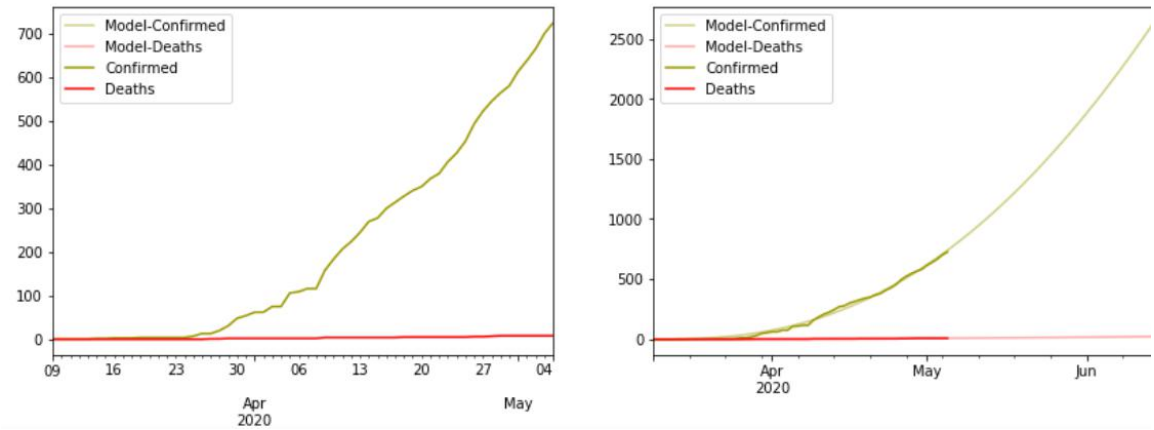
- Haryana

The first case of the COVID-19 pandemic in the Indian state of Haryana was reported on March 2020. In May, the total number of confirmed cases cross 500 mark, and as the model suggested it keeps increasing day by day.



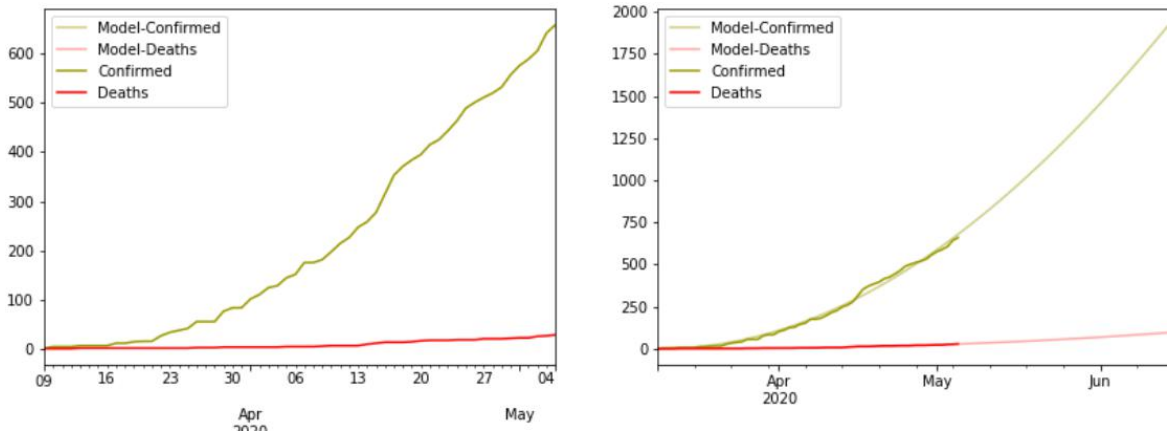
- Jammu & Kashmir

The first case of the COVID-19 pandemic in India was reported on 30 January 2020, originating from china. Slowly, the pandemic spread to various states and union territories including the union territory of Jammu & Kashmir. 02 suspected cases with high virus load were detected and isolated on 4 march in Government medical college, Jammu. One of them became the first confirmed positive case on 9 march 2020. Both individuals had a travel history to Iran. In May, the total number of confirmed cases cross 700 mark, and as the model suggested it keeps increasing day by day.



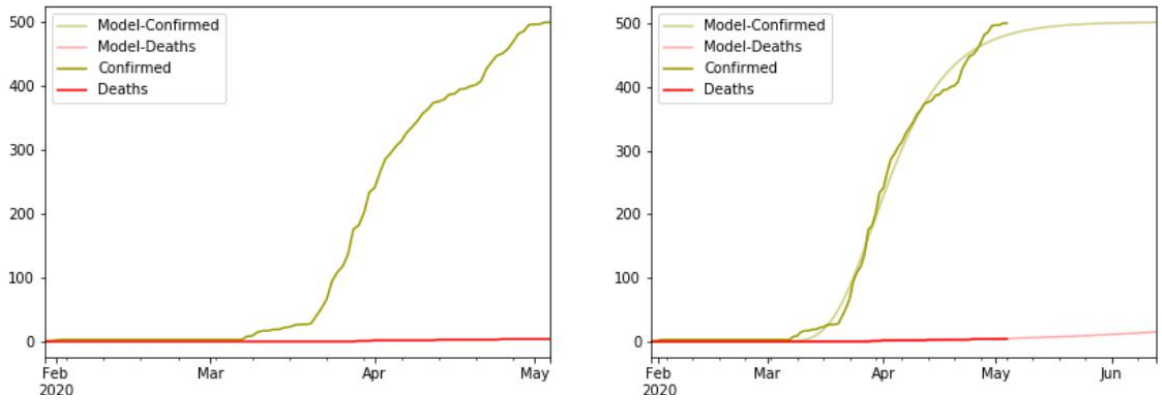
- Karnataka

The first case of the COVID-19 pandemic in the Indian state of Karnataka was confirmed on 9 march 2020. Two days later, the state became the first in India to invoke the provisions of the Epidemic diseases Act,1897, which are set to last for a year, to curb the spread of the disease till date. In May, the total number of confirmed cases cross 750 mark, and as the model suggested it keeps increasing day by day.



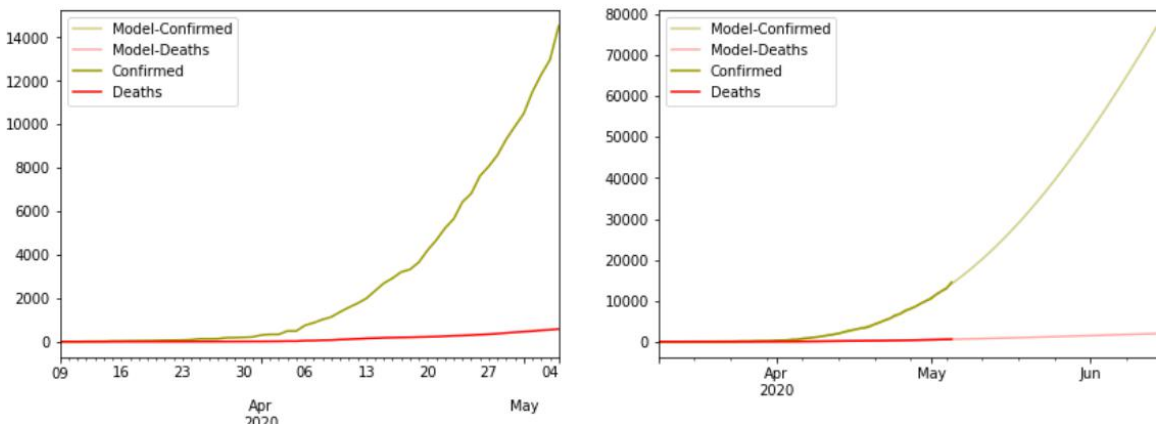
- Kerala

The first case of the COVID-19 pandemic in Kerala (which was also the first in all of India) was confirmed in Thrissur on 30 January 2020. Kerala has one of the lowest mortality rates in India (0.4%) compared to the national average of 1.63%. Kerala's success in containing COVID-19 has been widely praised both nationally and internationally. Despite the high number of cases in March, Kerala had, by April 30, reduced the rate of increase of new cases to less than 0.25% per day. The first positive cases of coronavirus in India were reported from three students of Kerala origin, travelling from the Wuhan province of China, which was the point of origin of the disease. They belonged to Thrissur, Alappuzha and Kasaragod districts of Kerala and two of them were medical students at a university in Wuhan. In May, the total number of confirmed cases cross 500 mark, and as the model suggested it keeps increasing day by day.



- Maharashtra

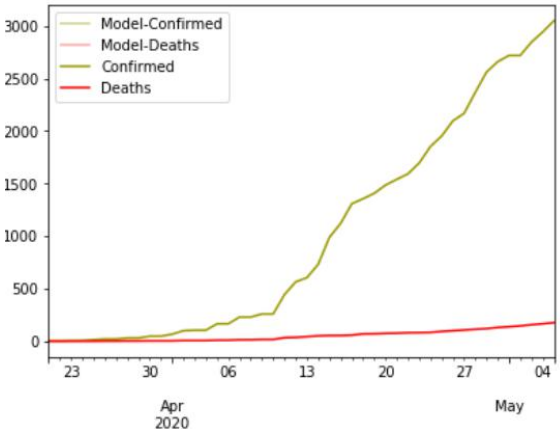
The first case of the COVID-19 pandemic in the Indian state of Maharashtra was confirmed on 9 March 2020. Maharashtra is a hotspot that accounts for nearly one-third of the total cases in India as well as about 40% of all deaths. In May, the total number of confirmed cases cross 14,500 mark, and as the model suggested it keeps increasing day by day.

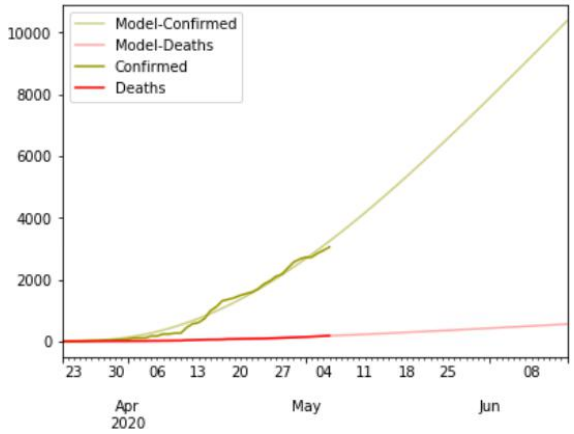


- Madhya Pradesh

The first cases of the COVID-19 pandemic in Madhya Pradesh were confirmed on 20 March 2020. its first cases of coronavirus with four persons testing positive in the state's Jabalpur city. Three members of a family who had returned from Dubai and another

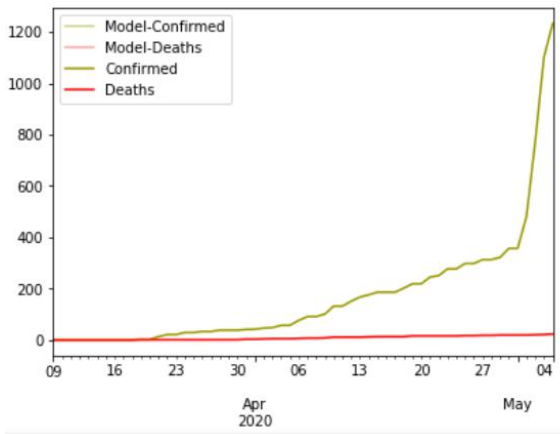
person who returned from Germany were found to have contracted the infection. In May, the total number of confirmed cases cross 3,000 mark, and as the model suggested it keeps increasing day by day.

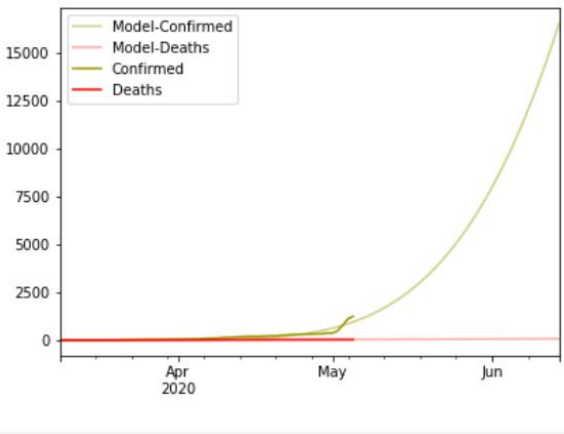




- Punjab

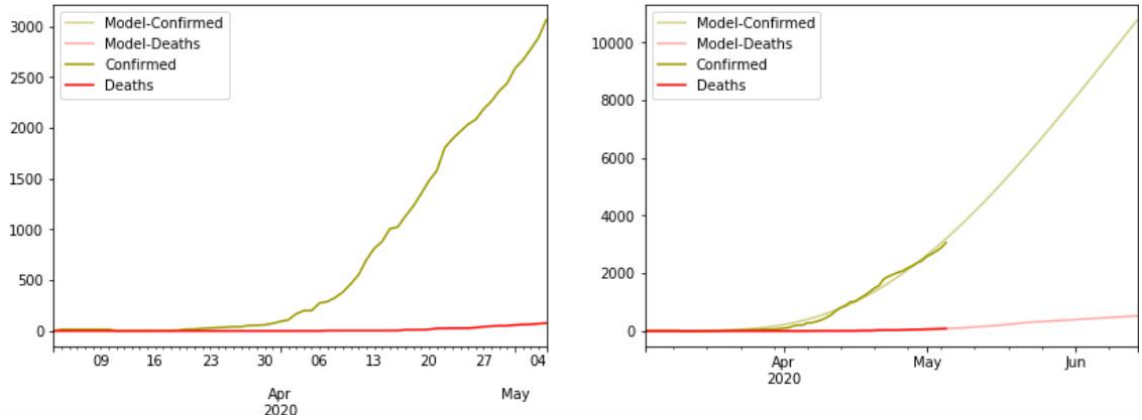
The COVID-19 pandemic was confirmed to have spread to the Indian state Punjab on 9 March 2020, when an Indian man returning from Italy was tested positive. First case was confirmed in Punjab at Amritsar, a person returned from Italy. A 72-year-old man in Punjab who had returned from Germany via Italy became the fourth victim of the virus in the country and first in Punjab. In May, the total number of confirmed cases cross 1200 mark, and as the model suggested it keeps increasing day by day.





- Rajasthan

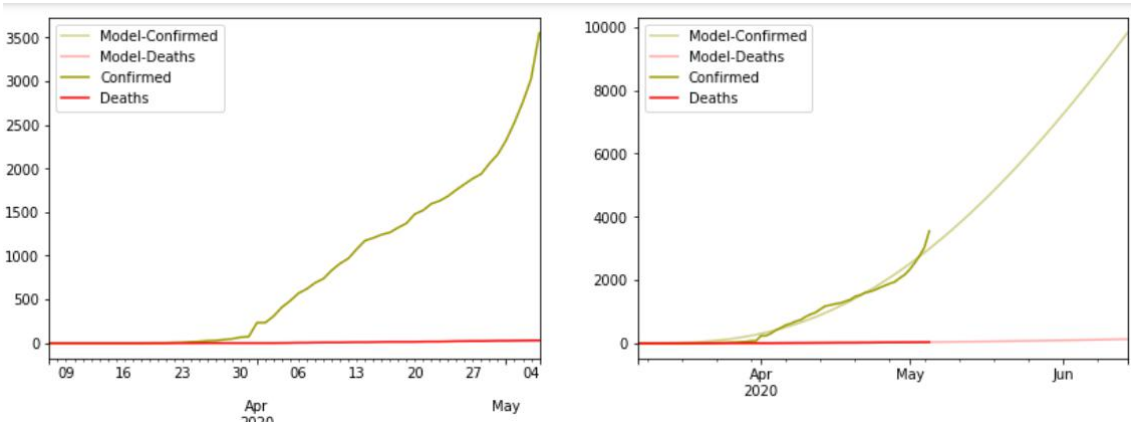
The first case of the COVID-19 pandemic in the Indian state of Rajasthan was reported on 2 march 2020 in Jaipur. On 2 March 2020, a 69-year-old Italian tourist who was part of a group of 23 tourists from Italy tested positive for COVID-19. His initial test returned negative, but after deterioration of his condition, a second test was made which returned positive. Two days later, his wife also tested positive which worried the health authorities since the group had travelled to numerous tourist destinations. On 3 March, the union Health Ministry held a meeting with officials of the Rajasthan Health Department over the outbreak. The Chief Minister of Rajasthan ordered the hotels where the Italian tourists resided to be sanitized. On 11 March, an 85-year-old man in Jaipur tested positive who had travel history to Dubai. In May, the total number of confirmed cases reach 3000 mark, and as the model suggested it keeps increasing day by day.



- Tamil Nadu

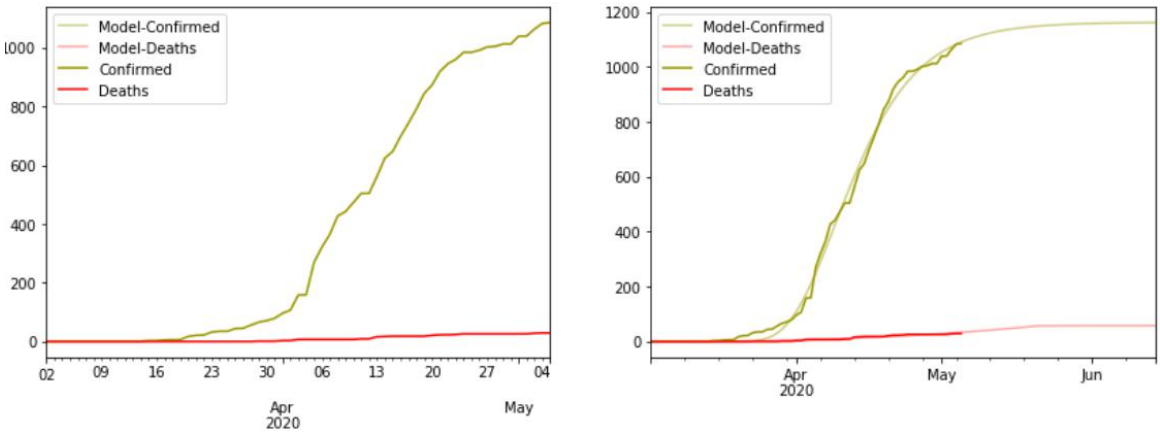
The first case of the COVID-19 pandemic in the Indian state of Tamil Nadu was reported on 7 march 2020. Tamil Nadu has the second highest number of confirmed cases in India after Maharashtra. All 37 districts of the state are affected by the pandemic with capital

district Chennai being the worst affected. More than half of the confirmed cases are from Chennai, which is also the most populous district of the state. The first case of coronavirus was confirmed on 7 March in a resident from Kanchipuram in Chennai. He had returned from Oman and stated developing symptoms including fever and cough. He was isolated in Rajiv Gandhi Government General Hospital. Later, on 10 March, he recovered and tested negative. In May, the total number of confirmed cases reach 3500 mark, and as the model suggested it keeps increasing day by day.



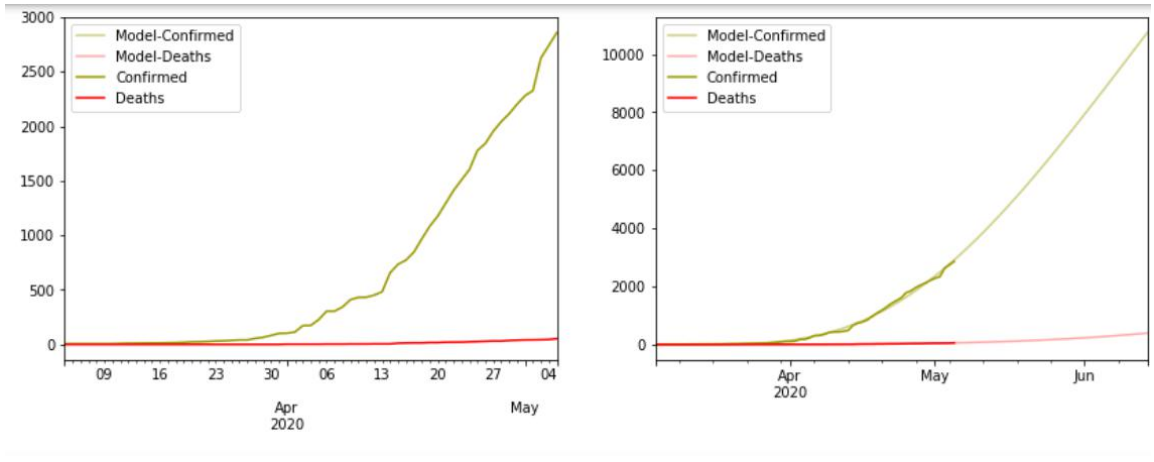
- Telangana

The first case of the COVID-19 pandemic in India was reported on 30 January 2020, originating from China. Slowly, the pandemic spread to various states and union territories including the state of Telangana. The first case was recorded in this region on 2 March from a man who had travel history with the UAE. In May, the total number of confirmed cases reach 1000 mark, and as the model suggested it keeps increasing day by day.



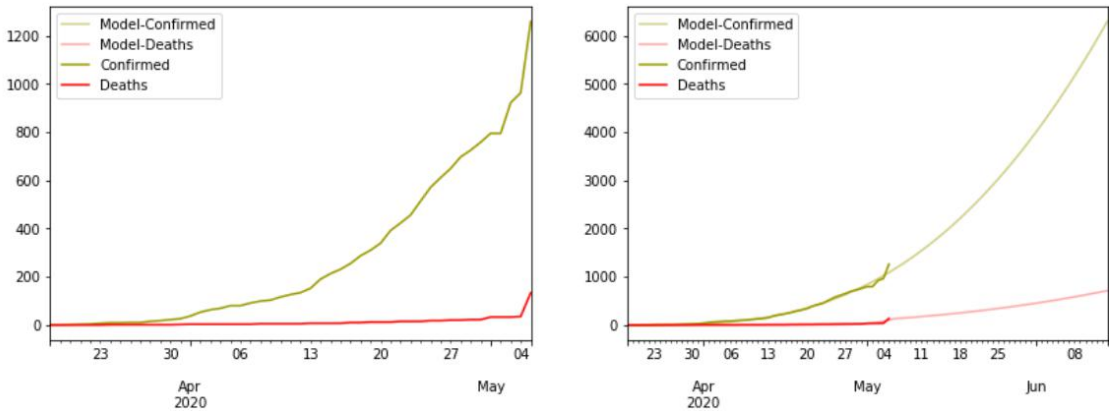
- Uttar Pradesh

The COVID-19 pandemic in Uttar Pradesh, India was first confirmed on 4 March 2020, with the first positive case in Ghaziabad. A middle-aged man in Ghaziabad who had travel history to Iran tested positive. In May, the total number of confirmed cases cross 2500 mark, and as the model suggested it keeps increasing day by day.



- West Bengal

The COVID-19 pandemic was first confirmed in the Indian state of West Bengal on 17 march 2020 in Kolkata. They first case, aged 18 years who had returned from the UK on 15 March, was tested positive. In May, the total number of confirmed cases reach 1200 mark, and as the model suggested it keeps increasing day by day.



4.1.2 USA

The COVID-19 pandemic in the United states is part of the worldwide pandemic of corona virus disease 2019(COVID-19). On December 31,2019, China announced the discovery of a cluster of pneumonia cases in Wuhan. The first American case was reported on January 20, and the U.S. outbreak was officially declared a public health emergency on January 31. some restrictions were placed on flights arriving from China, but the initial U.S. response to the pandemic was otherwise slow, in-terms of preparing the health care system, stopping other travel, and testing for the virus. Meanwhile, President Donald trump downplayed the threat posed by the virus and claimed the outbreak was under control.

The first known American deaths occurred in February, but were not known to be caused by COVID-19 until April. By the end of March, cases had been confirmed in all fifty U.S. states, the District of Columbia, and all inhabited U.S. territories except American Samoa.

On March 13, President Trump declared a national emergency. The Trump administration largely waited until mid-March to start purchasing large quantities of medical equipment. In late March, the administration started to use the Defense Production Act to direct industries to produce medical equipment. By April 17, the federal government approved disaster

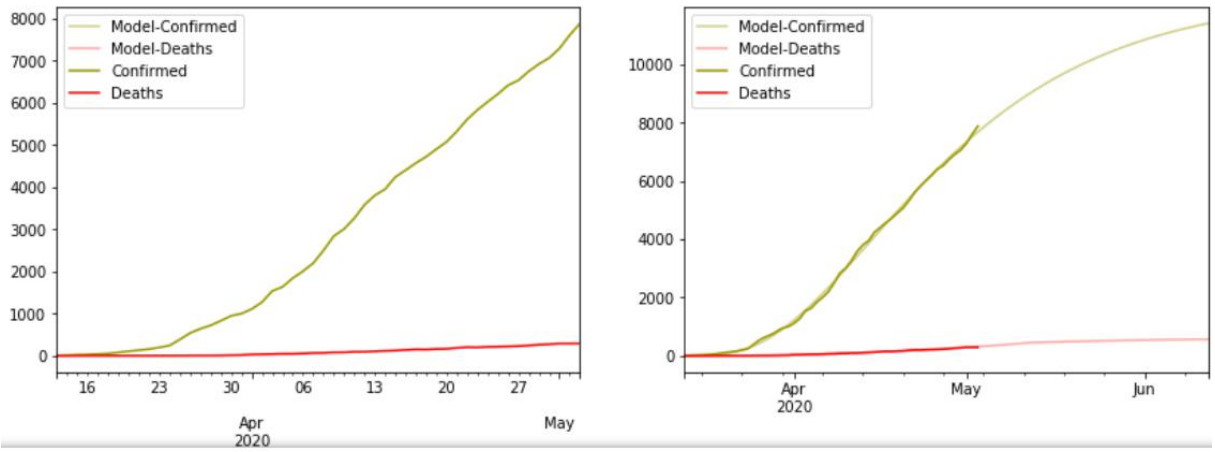
declarations for all states and territories. A second rise in infections began in June 2020, following relaxed restrictions in several states. State and local responses to the outbreak have included prohibitions and cancellation of large-scale gatherings (including festivals and sporting events), stay at home orders, and the closure of schools. Disproportionate numbers of cases have been observed among Black and Latino populations, and there were reported incidents of xenophobia and racism against Asian Americans. Clusters of infections and deaths have occurred in many areas.

On May 5, media misreported COVID-19 'infections' discovered in meat-packing plants in Iowa as 'cases'. The CDC prepared detailed guidelines for the reopening of businesses, public transit, restaurants, religious organizations, schools, and other public places. The Trump administration shelved the guidelines, but an unauthorized copy was published by the Associated Press on May 7. Six flow charts were ultimately published on May 15, and a sixty-page set of guidelines was released without comment on May 20, weeks after many states had already emerged from lockdown. By May 27, less than four months after the pandemic reached the U.S., 100,000 Americans had died with COVID-19. State economic reopenings and lack of widespread mask orders resulted in a sharp rise in cases across most of the continental U.S. outside of the Northeast. By June 11, the number of cases in the U.S. had passed two million.

USA prediction by specific state

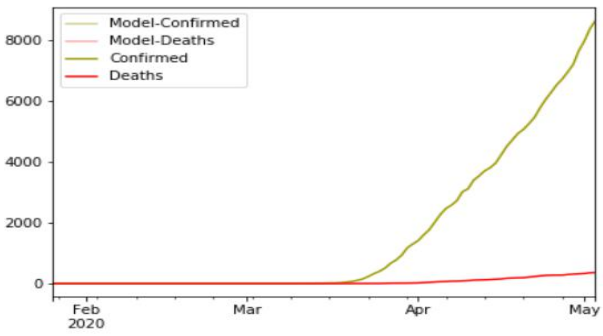
- Alabama

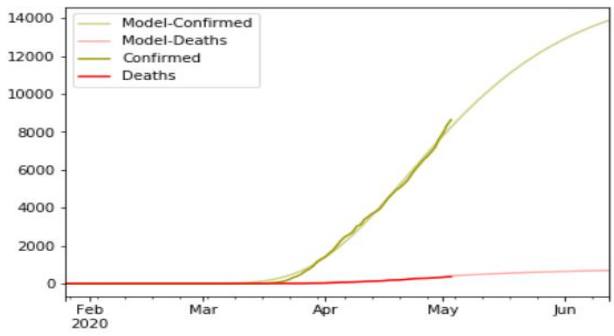
The COVID-19 pandemic was confirmed to have reached the U.S. state of Alabama in March 2020. On January 22, the Alabama department of public health asked healthcare providers to conduct screening of patients seeking care for influenza-like illnesses with travel to Wuhan,China. On February 4, the Alabama Department of Public Health asked travelers to mainland China who returned to the United States on or after January 22, 2020 to contact the Infectious Diseases and Outbreaks Division as soon as they arrive in Alabama. On March 2, the Alabama Department of Public Health advised individuals to "wash your hands frequently, avoid touching your face, cover coughs and sneezes, stay home when you are ill, and practice social distancing strategies". ADPH also asked universities and colleges to implement plans to mitigate the spread of disease on their campuses. On April 3, Government issued a statewide stay-at-home order until April 30. In May, the total number of confirmed cases reach 8000 mark, and as the model suggested it keeps increasing day by day.



- Arizona

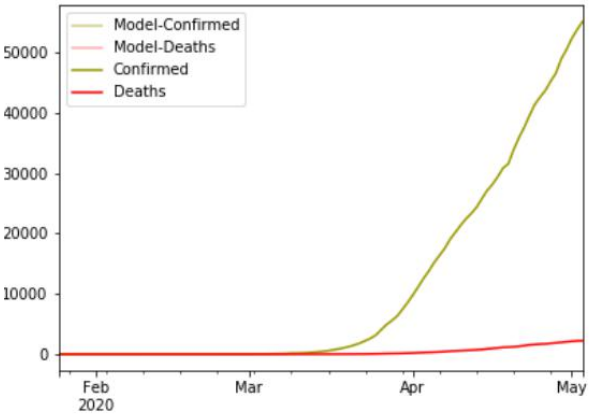
The COVID-19 pandemic was confirmed to have reached the U.S. state of Arizona in January 2020. The first confirmed case of COVID-19 in Arizona was announced by the Arizona department of health services(ADHS) on January 26,2020. A 20-year-old male student of Arizona state university(ASU), who had traveled to Wuhan,China, the point of origin of the outbreak, was diagnosed with COVID-19 and placed in isolation, and after repeated negative tests, the student was released from isolation and has since made a full recovery. On March 6, a woman from Pinal county was diagnosed with COVID-19. The woman, in her 40s, is a healthcare worker and was hospitalized at a Phoenix-area hospital, according to the Pinal County Public Health Department. This case was the first confirmed instance in Arizona of community spread, or where the source of the infection is unknown. On March 30, Government issued a statewide stay at home order to stop the spread of new coronavirus, barring Arizonans from leaving their residences except for food, medicine, and other essentials. The order took effect at the close of business March 31. On April 29, Government announced a partial reopening to begin on May 4 with details describing how some non-essential businesses can operate. The stay at home order was extended until May 15. In May, the total number of confirmed cases reach 8000 mark, and as the model suggested it keeps increasing day by day.

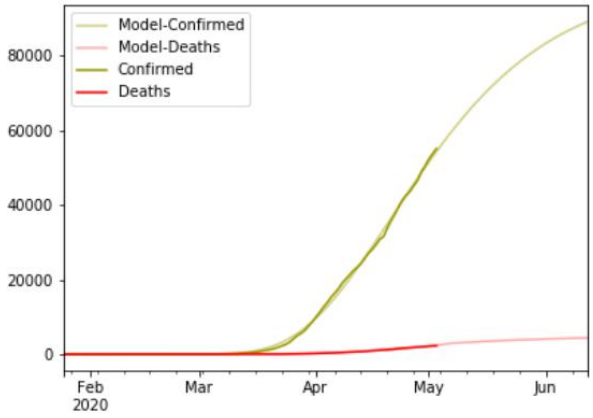




- California

The first case relating to the COVID-19 pandemic in the U.S. state of California was confirmed on January 26, 2020. On January 26, the Centers for disease Control and Prevention (CDC) confirmed the first case in California, the third case in the U.S. The person, a man in his 50s, who had returned from travel to Wuhan, China, was released from the hospital in Orange county on February 1 in good condition to in-home isolation. On January 31, the CDC confirmed the seventh case in the U.S., a man in Santa Clara county, who had recently traveled to Wuhan. The man recovered at home and was released from in-home isolation on February 20. A state of emergency has been in place in the state since March 4, 2020. A mandatory statewide stay-at-home order was issued on March 19. In May, the total number of confirmed cases cross 50000 mark, and as the model suggested it keeps increasing day by day.

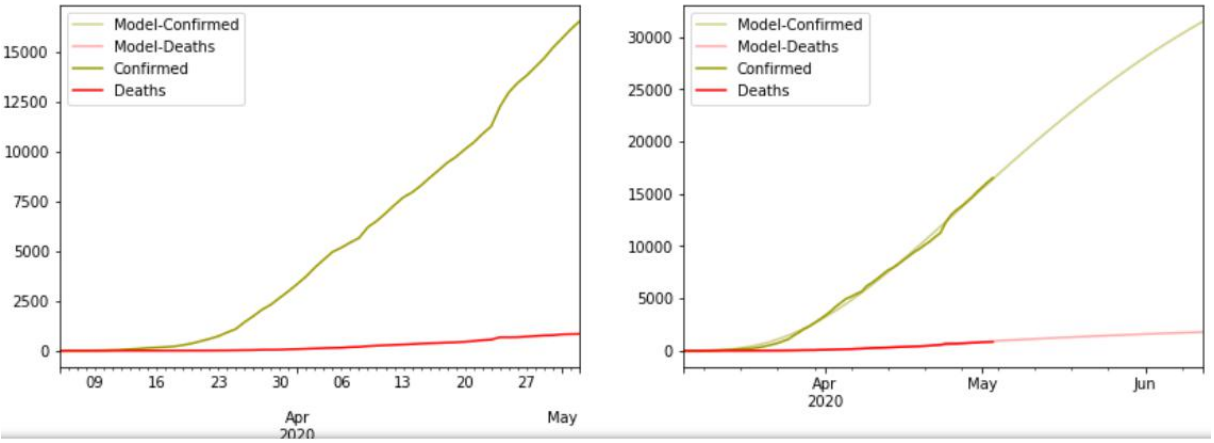




- Colorado

The COVID-19 pandemic reached Colorado on March 5,2020, When the state's first two cases were confirmed. On March 5, public health officials reported the first two cases of coronavirus in the state. The first known case was a man in his 30s visiting Summit county who had traveled to Italy in February with a companion who later tested positive for

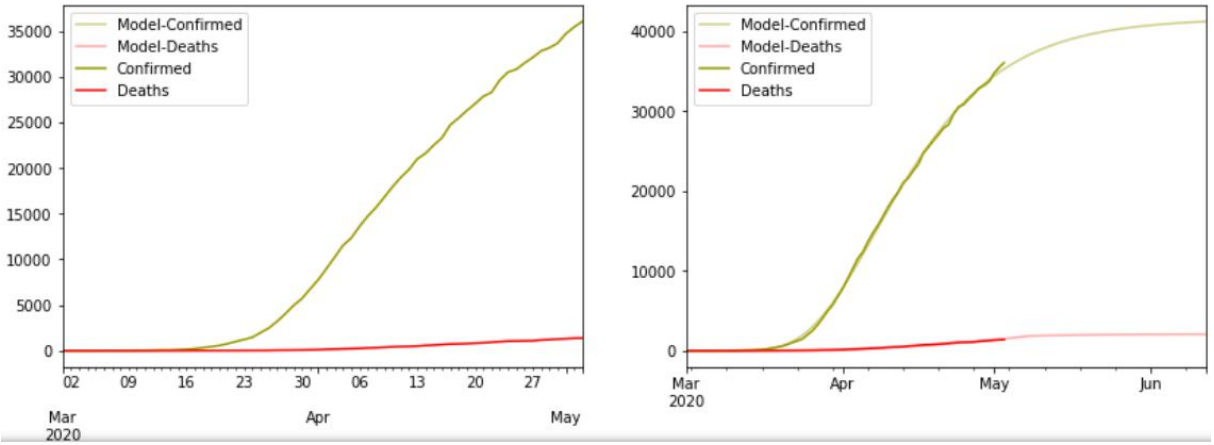
the virus. The second case was an elderly woman in Douglas county who had traveled on an international cruise. Both cases were considered presumptive positives; they had been tested by the state but had not been sent to the centers for Disease Control and Prevention(CDC) for verification. On April 1, Guidelines for patient prioritizing are made, with 3,342 cases, and 620 people hospitalized with COVID-19, Doctors could be forced to decide which patients to treat. In May, the total number of confirmed cases cross 15000 mark, and as the model suggested it keeps increasing day by day.



- Florida

On March 1, 2020, the U.S.state of Florida officially reported its first two COVID-19 cases, in Manatee and Hillsborough counties. In response, Government then declared a public health emergency. There is evidence, however, that community spread of COVID-19 began in Florida much earlier, perhaps as early as the first week of January, with as many as 171 people in Florida who had shown symptoms now identified with COVID-19, prior to receiving confirmation from the CDC. By March 11, the CDC saw evidence to conclude that community spread of the virus had occurred within the state. On April 1, Government issued an executive order to restrict activities within the state to those deemed as essential services.

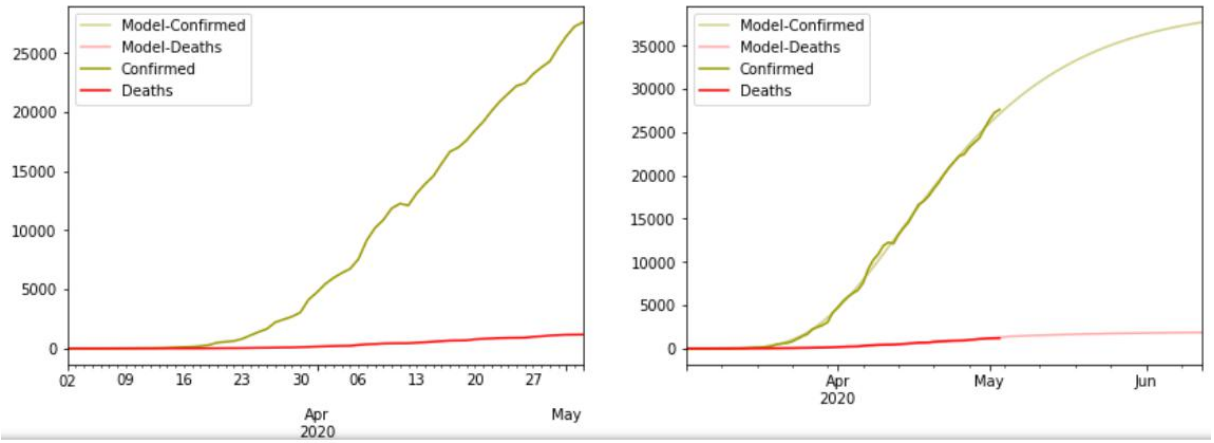
In May, the total number of confirmed cases cross 35000 mark, and as the model suggested it keeps increasing day by day.



- Georgia

The COVID-19 Pandemic was confirmed to have spread to Georgia when its first case was confirmed in Tbilisi on 26 February 2020. All flights from China and Wuhan to Tbilisi International Airport were cancelled until 27 January. The Health Ministry announced that all arriving passengers from China would be screened. Georgia also temporarily shut down all flights to Iran. On 26 February, Georgia confirmed its first COVID-19 case. A 50-year-old man, who returned to Georgia from Iran, was admitted to Infectious Diseases Hospital in Tbilisi. He came back to the Georgian border via Azerbaijan by taxi. As of 15 March, 33 cases were confirmed, 637 were held under quarantine and 54 were under hospital supervision. On 16 March, the spokesperson of the Government of Georgia Irakli Chikovani announced special measures and recommendations. The government of Georgia banned entrance to Georgia for any foreign nationals for the next two weeks. The Coordination Council recommended all elderly citizens of Georgia to avoid mass gatherings and isolate themselves. The Government also recommends cafes, restaurants and bars to offer customers the take-away service. 33 cases of coronavirus were confirmed in Georgia, 637 persons

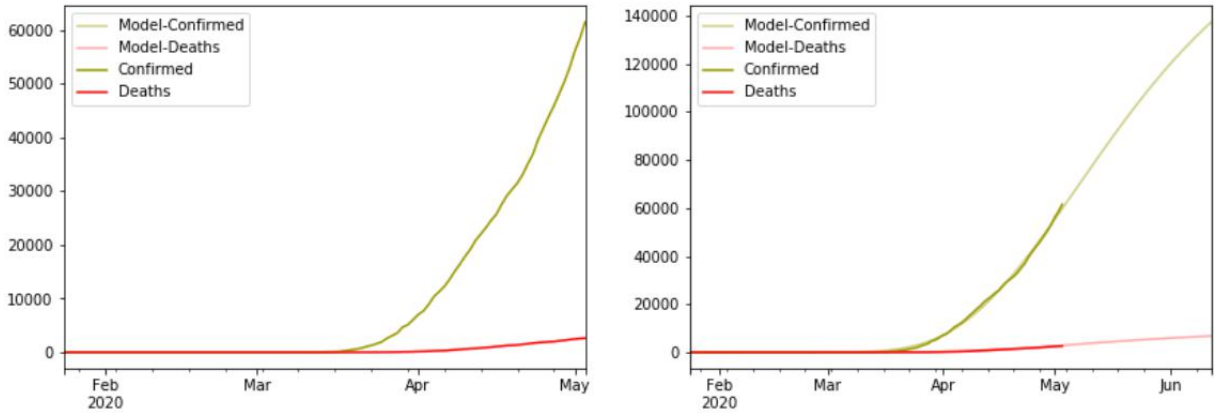
remain in quarantine and 54 persons are under direct medical supervision in hospitals for 16 March. The government disseminated a special SMS to all phones in Georgia informing population about measures and recommendations. In May, the total number of confirmed cases reach 30000 mark, and as the model suggested it keeps increasing day by day.



- Illinois

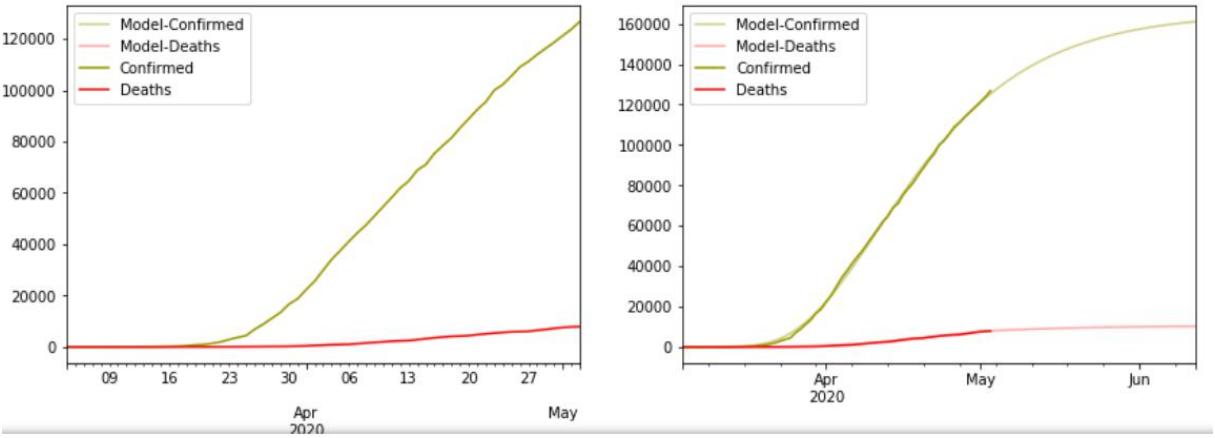
The ongoing COVID-19 pandemic began in the U.S. state of Illinois on January 24, 2020, when a woman in Chicago, who had just returned from the pandemic's place of origin in Wuhan, Hubei, China, tested positive for the virus. This was the second case of COVID-19 in the United States during the pandemic. The woman's husband was diagnosed with the disease a few days later, the first known case of human-to-human transmission in the united states. Community transmission was not suspected until March 8, when a case with no connection to other cases or recent travel was confirmed. In mid-March, as the number of known cases rose into the double digits, Governor J.B. Pritzker issued a disaster proclamation, the state's equivalent of a state of emergency, to respond to the crisis. The state took measures to halt the spread of the disease by closing all schools and colleges, ordering a stop to eviction enforcements, ordering all bars and restaurants closed to sit-in diners, and otherwise restricting large gatherings of people. As the virus spread further, the state enacted an even

stronger shelter in place order, affecting schools and businesses across the state. At first declared between the dates of March 21 and April 7, the order was later extended until April 30, then May 29. In May, the total number of confirmed cases reach 60000 mark, and as the model suggested it keeps increasing day by day.



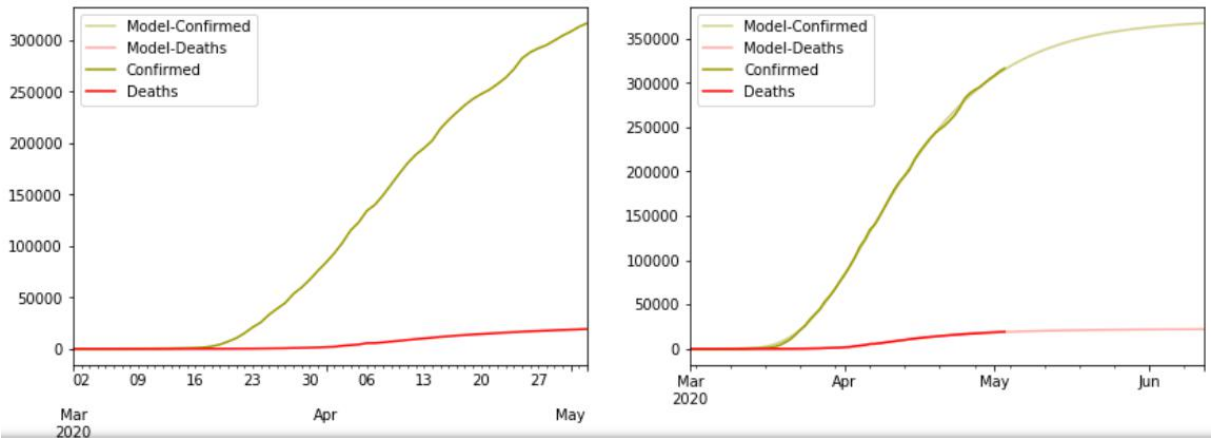
- New jersey

The COVID-19 pandemic reached the U.S. state of New Jersey with the first confirmed case arriving in Bergen county on March 2, 2020 and testing positive on March 4. On March 9, Governor Phil Murphy declared a state of emergency. A day later, schools and universities began closing and switching classes to online instruction. Also on March 10, the first person in the state died from the disease. A statewide curfew began on March 16, and all casinos, gyms, and movie theaters were closed; restaurants and bars were only allowed to remain open for delivery and takeout. On March 21, as the number of COVID-19 cases in the state surpassed 1,000, Governor Murphy announced a statewide stay-at-home order, requiring that all non-essential businesses be closed indefinitely by 9 p.m. that day. In May, the total number of confirmed cases reach 120000 mark, and as the model suggested it keeps increasing day by day.



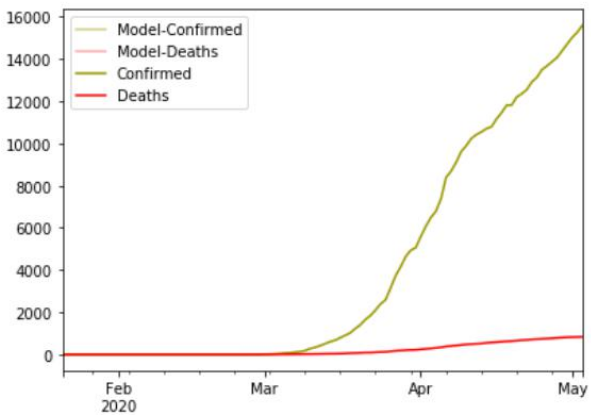
- New York

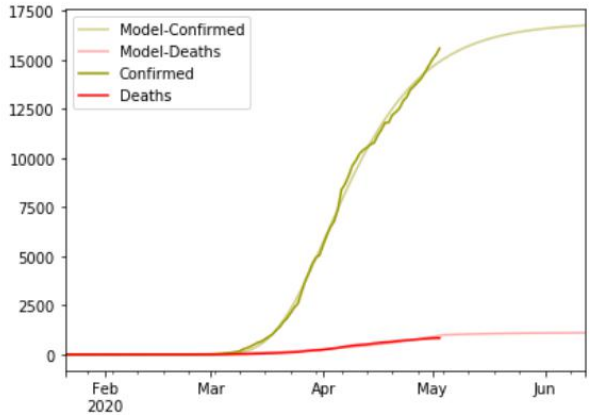
The first case of COVID-19 in the U.S. state of New York during the pandemic was confirmed on March 1, 2020, and the state quickly became an epicenter of the pandemic, with a record 12,274 new cases reported on April 4. By April 10, New York had more confirmed cases than any other country besides its own, but since then the outbreak has been mostly controlled in the state. In May, the total number of confirmed cases reach 300000 mark, and as the model suggested it keeps increasing day by day. Despite the high number of cases in March and April, by May 7, New York had reduced the rate of increase of new cases to less than 1 percent per day.



- Washington

The first confirmed case relating to the COVID-19 pandemic in the United States was announced by the state of Washington on January 21, 2020. Washington made the first announcement of a death from the disease in the U.S. on February 29 and later announced that two deaths there on February 26 were also due to COVID-19. Until mid-March, Washington had the highest absolute number of confirmed cases and the highest number per capita of any state in the country, until it was surpassed by New York state on April 10, 2020. Many of the deceased were residents of a nursing home in Kirkland, an Eastside suburb of Seattle in King county. In May, the total number of confirmed cases reach 16000 mark, and as the model suggested it keeps increasing day by day.





5.Coding

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import preprocessing

import time

from datetime import datetime

from scipy import integrate, optimize

import warnings

warnings.filterwarnings('ignore')
```

#ml libraries

```
import lightgbm as lgb

import xgboost as xgb

from xgboost import plot_importance, plot_tree

from sklearn.model_selection import RandomizedSearchCV ,GridSearchCV

from sklearn import linear_model

from sklearn.metrics import mean_squared_error
```

#data till 15th may 2020

```
submission=pd.read_csv(r'C:\Users\Ranjeet shrivastav\Downloads\covid-19\submission.csv')

test=pd.read_csv(r'C:\Users\Ranjeet shrivastav\Downloads\covid-19\test.csv')

train=pd.read_csv(r'C:\Users\Ranjeet shrivastav\Downloads\covid-19\train.csv')

display(train.head(80))

display(train.describe())

print("number of country_region: ",train['Country_Region'].nunique())

print("dates go from day",min(train['Date']),"to day",max(train['Date']),",a total of ",train['Date'].nunique(),"days")

print("countries with state informed: ",train[train['Province_State'].isna()==False]['Country_Region'].unique())

confirmed_total_date = train.groupby(['Date']).agg({'ConfirmedCases':['sum']})

fatalities_total_date = train.groupby(['Date']).agg({'Fatalities':['sum']})

total_date = confirmed_total_date.join(fatalities_total_date)


fig, (ax1,ax2)=plt.subplots(1,2,figsize=(22,7))

total_date.plot(ax=ax1)

ax1.set_title("Global confirmed cases", size=15)

ax1.set_ylabel("Number of cases", size=15)

ax1.set_xlabel("Date",size=15)

fatalities_total_date.plot(ax=ax2,color='orange')
```

```
ax2.set_title("Global deaths",size=15)

ax2.set_ylabel("Number of cases",size=15)

ax2.set_xlabel("Date",size=15)
```

#china excluded

```
confirmed_total_date_noChina                                                =
train[train['Country_Region']!='China'].groupby(['Date']).agg({'Confirme
dCases':['sum']})

fatalities_total_date_noChina                                               =
train[train['Country_Region']!='China'].groupby(['Date']).agg({'Fataliti
es':['sum']})

total_date_noChina                                                          =
confirmed_total_date_noChina.join(fatalities_total_date_noChina)



fig, (ax1,ax2)=plt.subplots(1,2,figsize=(17,7))

total_date_noChina.plot(ax=ax1)

ax1.set_title("Global confirmed cases without china", size=13)

ax1.set_ylabel("Number of cases without china", size=13)

ax1.set_xlabel("Date",size=13)

fatalities_total_date_noChina.plot(ax=ax2,color='orange')

ax2.set_title("Global deaths without china",size=13)

ax2.set_ylabel("Number of cases without china",size=13)

ax2.set_xlabel("Date",size=13)
```

#covid-19 tendency in china

```
confirmed_total_date_china=train[train['Country_Region']=='China'].group
by(['Date']).agg({'ConfirmedCases':['sum']})
```

```
fatalities_total_date_china=train[train['Country_Region']=='China'].grou
pby(['Date']).agg({'Fatalities':['sum']})
total_date_china=confirmed_total_date_china.join(fatalities_total_date_c
hina)


fig,(ax1,ax2)=plt.subplots(1,2,figsize=(17,7))


total_date_china.plot(ax=ax1)
ax1.set_title("China confirmed cases",size=13)
ax1.set_ylabel("Number of cases",size=13)
ax1.set_xlabel("Date",size=13)


fatalities_total_date_china.plot(ax=ax2,color='orange')
ax2.set_title("China deaths cases",size=13)
ax2.set_ylabel("Number of cases",size=13)
ax2.set_xlabel("Date",size=13)
confirmed_total_date_india=
train[train['Country_Region']=='India'].groupby('Date').agg({'ConfirmedC
ases':['sum']})
fatalities_total_date_india=train[train['Country_Region']=='India'].grou
pby('Date').agg({'Fatalities':['sum']})
total_date_india=
confirmed_total_date_india.join(fatalities_total_date_india)


fig, (ax1,ax2)=plt.subplots(1,2,figsize=(17,7))
```

```
total_date_india.plot(ax=ax1)

ax1.set_title('total confirmed cases in India',size=13)

ax1.set_ylabel('number of cases',size=13)

ax1.set_xlabel('date',size=13)


fatalities_total_date_india.plot(ax=ax2)

ax2.set_title('total deaths cases in India',size=13)

ax2.set_ylabel('number of cases',size=13)

ax2.set_xlabel('date',size=13)

confirmed_total_date_italy=train[train['Country_Region']=='Italy'].group
by('Date').agg({'ConfirmedCases':['sum']})

fatalities_total_date_italy=train[train['Country_Region']=='Italy'].grou
pby('Date').agg({'Fatalities':['sum']})

total_date_italy=confirmed_total_date_italy.join(fatalities_total_date_i
taly)


fig,(ax1,ax2)=plt.subplots(1,2,figsize=(17,7))

total_date_italy.plot(ax=ax1)

ax1.set_title('total confirmed cases in Italy',size=13)

ax1.set_ylabel('number of cases',size=13)

ax1.set_xlabel('date',size=13)


fatalities_total_date_italy.plot(ax=ax2)

ax2.set_title('total deaths in Italy',size=13)
```

```
ax2.set_ylabel('number of cases',size=13)

ax2.set_xlabel('date',size=13)

confirmed_total_date_spain=train[train['Country_Region']=='Spain'].group
by('Date').agg({'ConfirmedCases':['sum']})

fatalities_total_date_spain=train[train['Country_Region']=='Spain'].grou
pby('Date').agg({'Fatalities':['sum']})

total_date_spain=confirmed_total_date_spain.join(fatalities_total_date_s
pain)


fig,(ax1,ax2)=plt.subplots(1,2,figsize=(17,7))

total_date_italy.plot(ax=ax1)

ax1.set_title('total confirmed cases in Spain',size=13)

ax1.set_ylabel('number of cases',size=13)

ax1.set_xlabel('date',size=13)


fatalities_total_date_italy.plot(ax=ax2)

ax2.set_title('total deaths in Spain',size=13)

ax2.set_ylabel('number of cases',size=13)

ax2.set_xlabel('date',size=13)

confirmed_total_date_usa=train[train['Country_Region']=='US'].groupby('D
ate').agg({'ConfirmedCases':['sum']})

fatalities_total_date_usa=train[train['Country_Region']=='US'].groupby('
Date').agg({'Fatalities':['sum']})

total_date_usa=confirmed_total_date_usa.join(fatalities_total_date_usa)
```

```
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(17,7))
total_date_usa.plot(ax=ax1)
ax1.set_title('total confirmed cases in Usa',size=13)
ax1.set_ylabel('number of cases',size=13)
ax1.set_xlabel('date',size=13)


fatalities_total_date_usa.plot(ax=ax2)
ax2.set_title('total deaths in Usa',size=13)
ax2.set_ylabel('number of cases',size=13)
ax2.set_xlabel('date',size=13)
```

#providing a real time data from a website

```
import requests
from bs4 import BeautifulSoup
def main(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.content, 'html.parser')
    main=      soup.find("div",      class_="col-md-8      country-pop-
description").find_all_next("strong")[1]
    return (main.text)


```

#for india

```
india=main("https://www.worldometers.info/world-population/india-
population/")
pop_india=india.translate({ord(i): None for i in ','})
```

```
pop_india=int(pop_india)


total_date_india.ConfirmedCases=
total_date_india.ConfirmedCases/pop_india*100
plt.figure(figsize=(20,10))
plt.subplot(2,2,1)
total_date_india.ConfirmedCases.plot(ax=plt.gca(),title='India')
plt.ylabel("Fraction of populatio infected")
plt.ylim(0,0.5)
```

#for italy

```
italy=main("https://www.worldometers.info/world-population/italy-
population/")
pop_italy=italy.translate({ord(i): None for i in ','})
pop_italy=int(pop_italy)


total_date_italy.ConfirmedCases=
total_date_italy.ConfirmedCases/pop_italy*100
plt.figure(figsize=(20,10))
plt.subplot(2,2,2)
total_date_italy.ConfirmedCases.plot(ax=plt.gca(),title='Italy')
plt.ylabel("Fraction of populatio infected")
plt.ylim(0,0.5)
```

#for spain

```
spain=main("https://www.worldometers.info/world-population/spain-
population/")
pop_spain=spain.translate({ord(i): None for i in ','})
pop_spain=int(pop_spain)


total_date_spain.ConfirmedCases=
total_date_spain.ConfirmedCases/pop_spain*100
plt.figure(figsize=(20,10))
plt.subplot(2,2,3)
total_date_spain.ConfirmedCases.plot(ax=plt.gca(),title='Spain')
plt.ylabel("Fraction of populatio infected")
plt.ylim(0,0.5)
```

#for usa

```
usa=main("https://www.worldometers.info/world-population/us-population/")
pop_usa=usa.translate({ord(i): None for i in ','})
pop_usa=int(pop_usa)


total_date_usa.ConfirmedCases= total_date_usa.ConfirmedCases/pop_usa*100
plt.figure(figsize=(20,10))
plt.subplot(2,2,4)
```

```
total_date_usa.ConfirmedCases.plot(ax=plt.gca(),title='USA')

plt.ylabel("Fraction of populatio infected")

plt.ylim(0,0.5)
```

#for china

```
china=main("https://www.worldometers.info/world-population/china-
population/")

pop_china=usa.translate({ord(i): None for i in ','})

pop_china=int(pop_china)


total_date_china.ConfirmedCases=
total_date_china.ConfirmedCases/pop_china*100

plt.figure(figsize=(20,10))

plt.subplot(2,2,4)

total_date_china.ConfirmedCases.plot(ax=plt.gca(),title='China')

plt.ylabel("Fraction of populatio infected")

plt.ylim(0,0.5)

confirmed_total_date_india=train  [(train['Country_Region']=='India')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'ConfirmedCases':['su
m']})

fatalities_total_date_india=  train[(train['Country_Region']=='India')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'Fatalities':['sum']})

total_date_india=
confirmed_total_date_india.join(fatalities_total_date_india)
```

```
confirmed_total_date_italy=train  [(train['Country_Region']=='Italy')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'ConfirmedCases':['su
m']})

fatalities_total_date_italy=  train[(train['Country_Region']=='Italy')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'Fatalities':['sum']})

total_date_italy=
confirmed_total_date_italy.join(fatalities_total_date_italy)


confirmed_total_date_spain=train  [(train['Country_Region']=='Spain')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'ConfirmedCases':['su
m']})

fatalities_total_date_spain=  train[(train['Country_Region']=='Spain')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'Fatalities':['sum']})

total_date_spain=
confirmed_total_date_spain.join(fatalities_total_date_spain)


confirmed_total_date_usa=train    [(train['Country_Region']=='US')    &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'ConfirmedCases':['su
m']})

fatalities_total_date_usa=    train[(train['Country_Region']=='US')    &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'Fatalities':['sum']})

total_date_usa= confirmed_total_date_usa.join(fatalities_total_date_usa)
```

```
confirmed_total_date_china=train  [(train['Country_Region']=='China')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'ConfirmedCases':['su
m']})
fatalities_total_date_china=  train[(train['Country_Region']=='China')  &
train['ConfirmedCases']!=0].groupby(['Date']).agg({'Fatalities':['sum']})
total_date_china=
confirmed_total_date_china.join(fatalities_total_date_china)



india= [i for i in total_date_india.ConfirmedCases['sum'].values]
india_10= india[0:70]

italy= [i for i in total_date_italy.ConfirmedCases['sum'].values]
italy_10= italy[0:70]

spain= [i for i in total_date_spain.ConfirmedCases['sum'].values]
spain_10= spain[0:70]

usa= [i for i in total_date_usa.ConfirmedCases['sum'].values]
usa_10= usa[0:70]

china= [i for i in total_date_china.ConfirmedCases['sum'].values]
china_10= china[0:70]
```

```
plt.figure(figsize=(12,6))
plt.plot(india_10)
plt.plot(italy_10)
plt.plot(spain_10)
plt.plot(usa_10)
plt.plot(china_10)


plt.legend(["India","Italy","Spain","USA","China"], loc='upper left')
plt.title("COVID-19 infections from the first confirmed case",size=15)
plt.xlabel("Days",size=13)
plt.ylabel("Infected cases",size=13)
plt.ylim(0,130000)
plt.show()
import datetime
confirmed_df =
pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid
19_confirmed_global.csv')
death_df =
pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid
19_deaths_global.csv')
recovered_df =
pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
```

```
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid
19_recovered_global.csv')


confirmed_table      =      confirmed_df.melt(id_vars=["Province/State",
"Country/Region",        "Lat",        "Long"],        var_name="Date",
value_name="Confirmed").fillna('').drop(['Lat', 'Long'], axis=1)

death_table = death_df.melt(id_vars=["Province/State", "Country/Region",
"Lat",                     "Long"],                     var_name="Date",
value_name="Deaths").fillna('').drop(['Lat', 'Long'], axis=1)

recovered_table      =      recovered_df.melt(id_vars=["Province/State",
"Country/Region",        "Lat",        "Long"],        var_name="Date",
value_name="Recovered").fillna('').drop(['Lat', 'Long'], axis=1)


full_table = confirmed_table.merge(death_table).merge(recovered_table)


full_table['Date'] = pd.to_datetime(full_table['Date'])

full_table
```

#cleaning data

```
def get_time_series(country):
```

for some countries, data is spread over several Provinces

```
    if             full_table[full_table['Country/Region']             ==
country]['Province/State'].nunique() > 1:

        country_table    =    full_table[full_table['Country/Region']    ==
country]

        country_df = pd.DataFrame(pd.pivot_table(country_table, values =
['Confirmed', 'Deaths', 'Recovered'],
```

```
                                index='Date', aggfunc=sum).to_records())
        return    country_df.set_index('Date')[['Confirmed',    'Deaths',
'Recovered']]
    df = full_table[(full_table['Country/Region'] == country)
                & (full_table['Province/State'].isin(['', country]))]
    return df.set_index('Date')[['Confirmed', 'Deaths', 'Recovered']]



def get_time_series_province(province):
    # for some countries, data is spread over several Provinces
    df = full_table[(full_table['Province/State'] == province)]
    return df.set_index('Date')[['Confirmed', 'Deaths', 'Recovered']]


country='China'
df= get_time_series(country)
if len(df)>1 and df.iloc[-2,0]>=df.iloc[-1,0]:
    df.drop(df.tail(1).index,inplace=True)
df.tail(10)
```

#model

```
import math
def model_with_lag(N,a,alpha,lag,t):
    lag=min(max(lag,-100),100)
    return max(N,0)*(1-math.e**(min(-a,0)*(t-lag)))**max(alpha,0)
```

```
def model(N,a,alpha,t):
    return max(N,0)*(1-math.e**(min(-a,0)*t))**max(alpha,0)
model_index=0
def model_loss(params):
    N,a,alpha=params
    model_x=[]
    r=0
    for t in range(len(df)):
        r+=(model(N,a,alpha,t)-df.iloc[t,model_index])**2
    return math.sqrt(r)
from scipy.optimize import minimize
use_lag_model=False
if use_lag_model:

opt=minimize(model_loss,x0=np.array([200000,0.05,15,0]),method='Nelder-
Mead',tol=1e-5).x
else:
    model_index=0

opt_confirmed=minimize(model_loss,x0=np.array([200000,0.05,15]),method='
Nelder-Mead',tol=1e-5).x
    model_index=1

opt_deaths=minimize(model_loss,x0=np.array([200000,0.05,15]),method='Nel
der-Mead',tol=1e-5).x
    model_index=2
```

```
opt_recovered=minimize(model_loss,x0=np.array([200000,0.05,15]),method='
Nelder-Mead',tol=1e-5).x
import matplotlib
%matplotlib inline


model_x=[]
for t in range(len(df)):
    model_x.append([df.index[t],model(*opt_confirmed,t),
                    model(*opt_deaths,t),model(*opt_recovered,t)])
model_sim= pd.DataFrame(model_x,dtype=int)
model_sim.set_index(0,inplace=True)
model_sim.columns=['Model-Confirmed','Model-Deaths','Model-Recovered']


model_sim['Model-Active']=model_sim['Model-Confirmed']-model_sim['Model-
Deaths']-model_sim['Model-Recovered']
model_sim.loc[model_sim['Model-Active']<0,'Model-Active']=0
plot_color=['#99990077','#FF000055','#0000FF55','#00FF0055','#999900FF',
'#FF0000FF','#0000FFFF','#00FF00FF']


pd.concat([model_sim,df],axis=1).plot(color=plot_color)
plt.show()
import datetime
start_date=df.index[0]
n_days=len(df)+30
```

```
extended_model_x=[]
last_row=[]

isValid=True
last_death_rate=0

for t in range(n_days):
    extended_model_x.append([start_date+datetime.timedelta(days=t),
                             model(*opt_confirmed,t),
                             model(*opt_deaths,t),
                             model(*opt_recovered,t)])

    if (t>len(df)):
        last_row=extended_model_x[-1]
        if    (last_row[2]+last_row[3]>last_row[1])    or    (last_row[2]>
last_row[1]*0.12):
            if (isValid):
                last_row2 =extended_model_x[-2]
                last_death_rate =last_row2[2]/last_row2[1]
                isValid= False

        if (last_row[2]>last_row[1]*0.05):
            last_row[2] = last_row[1]*last_death_rate
```

```
        if (last_row[2]+last_row[3]>last_row[1]):
            last_row[2] = last_row[1]*last_death_rate
            last_row[3] = last_row[1]*(1-last_death_rate)

extended_model_sim = pd.DataFrame(extended_model_x, dtype=int)
extended_model_sim.set_index(0, inplace=True)

extended_model_sim.columns = ['Model-Confirmed', 'Model-Deaths', 'Model-
Recovered']
extended_model_sim['Model-Active'] = extended_model_sim['Model-
Confirmed'] - extended_model_sim['Model-Deaths'] -
extended_model_sim['Model-Recovered']
extended_model_sim.loc[extended_model_sim['Model-Active']<0,'Model-
Active'] = 0

plot_color = ['#99990077', '#FF000055', '#0000FF55', '#00FF0055',
'#999900FF', '#FF0000FF', '#0000FFFF', '#00FF00FF']

pd.concat([extended_model_sim, df], axis=1).plot(color = plot_color)
print('China COVID-19 Prediction')
plt.show()
df.tail()
pd.options.display.float_format='{:20,.0f}'.format
concat_df=pd.concat([df,extended_model_sim],axis=1)
concat_df[concat_df.index.day%3==0]
```

```
def display_fit(df,opt_confirmed,opt_deaths,opt_recovered,ax):
    model_x=[]

    isValid=True
    last_death_rate=0

    for t in range (len(df)):

model_x.append([df.index[t],model(*opt_confirmed,t),model(*opt_deaths,t),
                        model(*opt_recovered,t)])

        #if deaths + recovered > confirmed or deaths rate > 5%, maybe not valid
        if(t > len(df)):
            last_row = model_x[-1]
            if (last_row[2] + last_row[3] > last_row[1]) or (last_row[2]
> last_row[1]*0.05):
                if(isValid):
                    last_row2 = model_x[-2]
                    last_death_rate = last_row2[2]/last_row2[1]
                    isValid = False

            if (last_row[2] > last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate
```

```
            if (last_row[2] + last_row[3] > last_row[1]):
                last_row[2] = last_row[1]*last_death_rate
                last_row[3] = last_row[1]*(1-last_death_rate)
    model_sim = pd.DataFrame(model_x,dtype=int)
    model_sim.set_index(0,inplace=True)
    model_sim.columns = ['Model-Confirmed','Model-Deaths','Model-
Recovered']

    model_sim['Model-Active'] = model_sim['Model-Confirmed']-
model_sim['Model-Deaths']-model_sim['Model-Recovered']
    model_sim.loc[model_sim['Model-Active']<0,'Model-Active']=0
    plot_color= ['#99990077', '#FF000055', '#0000FF55', '#00FF0055',
'#999900FF', '#FF0000FF', '#0000FFFF', '#00FF00FF']

    return pd.concat([model_sim,df],axis=1).plot(ax=ax,
figsize=(14,10),color = plot_color)

def display_extended_curve(df, opt_confirmed, opt_deaths,
opt_recovered,ax):
    start_date=df.index[0]
    n_days=len(df)+40
    extended_model_x=[]

    isValid=True
    last_death_rate=0
```

```
    for t in range(n_days):
        extended_model_x.append([start_date + datetime.timedelta(days=t),
model(*opt_confirmed,t),model(*opt_deaths,t),
                                 model(*opt_recovered,t)])
```

#if deaths + recoverd >confirmed or deaths rate>5%, maybe not valid

```
        if (t>len(df)):
            last_row=extended_model_x[-1]
            if          (last_row[2]+last_row[3]>last_row[1])          or
(last_row[2]>last_row[1]*0.05):
                if (isValid):
                    last_row2=extended_model_x[-2]
                    last_death_rate = last_row[2]/last_row2[1]

                    isValid=False

            if (last_row[2]>last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate

            if(last_row[2]+last_row[3]>last_row[1]):
                last_row[2]=last_row[1]*last_death_rate
                last_row[3] = last_row[1]*(1-last_death_rate)
```

```
    extended_model_sim=pd.DataFrame(extended_model_x,dtype=int)
    extended_model_sim.set_index(0,inplace=True)
    extended_model_sim.columns=                ['Model-Confirmed','Model-
Deaths','Model-Recovered']
    extended_model_sim['Model-Active']=extended_model_sim['Model-
Confirmed']-extended_model_sim['Model-Deaths']-
extended_model_sim['Model-Recovered']
    extended_model_sim.loc[extended_model_sim['Model-Active']<0,'Model-
Active']=0
    plot_color  =  ['#99990077',  '#FF000055',  '#0000FF55',  '#00FF0055',
'#999900FF', '#FF0000FF', '#0000FFFF', '#00FF00FF']

    return
pd.concat([extended_model_sim,df],axis=1).plot(ax=ax,figsize=(14,10),col
or=plot_color)



def opt_display_model(df,stats):
    #if the last data point repeats the previous one,or is lower, drop
it

    if len(df)>1 and df.iloc[-2,0]>=df.iloc[-1,0]:
        df.drop(df.tail(1).index,inplace=True)
```

```
    global model_index
    model_index=0
    opt_confirmed                                                    =
minimize(model_loss,x0=np.array([200000,0.05,15]),method='Nelder-
Mead',tol=1e-5).x
    model_index=1
    opt_deaths                                                       =
minimize(model_loss,x0=np.array([200000,0.05,15]),method='Nelder-
Mead',tol=1e-5).x
    model_index=2
    opt_recovered=
minimize(model_loss,x0=np.array([200000,0.05,15]),method='Nelder-
Mead',tol=1e-5).x


    if min(opt_confirmed)>0:
        stats.append([country,*opt_confirmed,*opt_deaths,*opt_recovered])
        n_plot=len(stats)
        plt.figure(1)
        ax1=plt.subplot(221)
        display_fit(df,opt_confirmed,opt_deaths,opt_recovered,ax1)
        ax2=plt.subplot(222)

display_extended_curve(df,opt_confirmed,opt_deaths,opt_recovered,ax2)
        plt.show()
```

World COVID-19 Prediction
#predict world with china data

```
stats=[]

df=full_table[['Province/State','Country/Region','Date','Confirmed','Deaths
',
              'Recovered']].groupby('Date').sum()

print('World COVID-19 Prediction (With China Data)')

opt_display_model(df,stats)
```

Predict World (Without China Data)

```
stats = []


df = full_table[full_table['Country/Region'] !=
'China'][['Province/State','Country/Region', 'Date', 'Confirmed', 'Deaths',
'Recovered']].groupby('Date').sum()

print('World COVID-19 Prediction(Without China Data)')

opt_display_model(df, stats)
```

Predict by Specify Country

#country specify

```
stats = []

for country in ['India','US','Italy','Spain']:

    df = get_time_series(country)

    print('{} COVID-19 Prediction'.format(country))

    opt_display_model(df, stats)
```

Predict all country greater than 10000

```
stats = []

for country in sorted(full_table['Country/Region'].unique()):

    df = get_time_series(country)

    if len(df)==0 or (max(df['Confirmed']) < 10000):

        continue

    print('{} COVID-19 Prediction'.format(country))

    opt_display_model(df, stats)

stats_df = pd.DataFrame(stats)

stats_df.columns = ['country','Confirmed-N','Confirmed-a','Confirmed-alpha',

                    'Deaths-N','Deaths-a','Deaths-alpha','Recovered-
N','Recovered-a',

                    'Recovered-alpha']

stats_df
```

```
pd.set_option('display.max_rows',500)

pd.options.display.float_format = '{:20,.4f}'.format

stats_df.astype({'Confirmed-N':'int'}).sort_values(by='Confirmed-
N',ascending=False)


ax = stats_df.plot.scatter(x='Confirmed-alpha', y='Confirmed-a')

plt.show()
```

#predict all province greater than 1000

```
stats = []

for Province in sorted(full_table['Province/State'].unique()):

    if (Province == ''):

        continue

    df = get_time_series_province(Province)

    if len(df) == 0 or (max(df['Confirmed'])<1000):

        continue

    print('{} COVID-19 Prediction'.format(Province))

    opt_display_model(df, stats)
```

#predict US by province/state

```
fulltable_us = pd.read_csv(r'C:\Users\Ranjeet shrivastav\Downloads\covid-
19\us-counties.csv')

fulltable_us = fulltable_us.drop(['fips'],
axis=1).groupby(['date','state']).sum().reset_index()

fulltable_us.columns = ['Date','Province/State','Confirmed','Deaths']

fulltable_us['Date'] = pd.to_datetime(fulltable_us['Date'])

fulltable_us
```

#only has confirmed and deaths two column, so we build a new model

```
def display_fit_us(df, opt_confirmed, opt_deaths, ax):

    model_x = []


    isValid = True

    last_death_rate = 0


    for t in range(len(df)):

        model_x.append([df.index[t], model(*opt_confirmed, t),
model(*opt_deaths, t)])
```

#if deaths + recovered > confirmed or deaths rate > 5%, maybe not valid

```
        if (t>len(df)):
            last_row = model_x[-1]
            if(last_row[2]> last_row[1]*0.05):
                if (isValid):
                    last_row2 = model_x[-2]
                    last_death_rate = last_row2[2]/last_row2[1]
                    isValid=False
            if (last_row[2]> last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate

        model_sim = pd.DataFrame(model_x, dtype=int)
        model_sim.set_index(0, inplace=True)
        model_sim.columns = ['Model-Confirmed', 'Model-Deaths']
        plot_color = ['#99990077', '#FF000055', '#999900FF', '#FF0000FF']
        return pd.concat([model_sim,
df],axis=1).plot(ax=ax,figsize=(14,10),color = plot_color)



def display_extended_curve_us(df, opt_confirmed, opt_deaths, ax):

    start_date = df.index[0]
    n_days = len(df) + 40
    extended_model_x = []

    isValid = True
    last_death_rate = 0

    for t in range(n_days):
        extended_model_x.append([start_date + datetime.timedelta(days=t),
model(*opt_confirmed, t), model(*opt_deaths, t)])
```

#if deaths + recovered > confirmed or deaths rate > 5%, maybe not valid

```
        if (t > len(df)):
            last_row = extended_model_x[-1]
            if (last_row[2] > last_row[1]*0.05):
                if (isValid):
                    last_row2 = extended_model_x[-2]
                    last_death_rate = last_row2[2]/last_row2[1]
                    isValid = False

            if (last_row[2] > last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate




    extended_model_sim = pd.DataFrame(extended_model_x, dtype=int)
    extended_model_sim.set_index(0, inplace=True)
    extended_model_sim.columns = ['Model-Confirmed', 'Model-Deaths']


    plot_color = ['#99990077', '#FF000055', '#999900FF', '#FF0000FF']


    return pd.concat([extended_model_sim, df], axis=1).plot(ax=ax,
figsize=(14, 10), color = plot_color)


def opt_display_model_us(df, stats):
    # if the last data point repeats the previous one, or is lower, drop it
    if len(df) > 1 and df.iloc[-2,0] >= df.iloc[-1,0]:
        df.drop(df.tail(1).index,inplace=True)
    global model_index
    model_index = 0
    opt_confirmed = minimize(model_loss, x0=np.array([200000, 0.05, 15]),
method='Nelder-Mead', tol=1e-5).x
    model_index = 1
    opt_deaths = minimize(model_loss, x0=np.array([200000, 0.05, 15]),
method='Nelder-Mead', tol=1e-5).x
    if min(opt_confirmed) > 0:
        stats.append([country, *opt_confirmed, *opt_deaths])
```

```
        n_plot = len(stats)
        plt.figure(1)
        ax1 = plt.subplot(221)
        display_fit_us(df, opt_confirmed, opt_deaths, ax1)
        ax2 = plt.subplot(222)
        display_extended_curve_us(df, opt_confirmed, opt_deaths, ax2)
        plt.show()


def get_time_series_province_us(province):
    # for some countries, data is spread over several Provinces
    global fulltable_us
    df = fulltable_us[(fulltable_us['Province/State'] == province)]
    return df.set_index('Date')[['Confirmed', 'Deaths']]
#test washington prediction
df = get_time_series_province_us('Washington')
df
stats = []
print('{} of US COVID-19 Prediction'.format('Washington'))
opt_display_model_us(df, stats)
#all provinces of United States which confirm cases greater than 500
stats= []
for Province in sorted(fulltable_us['Province/State'].unique()):
    if(Province==''):
        continue
    df = get_time_series_province_us(Province)
    if len(df)==0 or (max(df['Confirmed']) < 500):
        continue
    print('{} of US COVID-19 Prediction'.format(Province))
    opt_display_model_us(df,stats)
#predict india by state
fulltable_india = pd.read_csv(r'C:\Users\Ranjeet
shrivastav\Downloads\covid-19\covid_19_india.csv')
fulltable_india = fulltable_india.drop(['Sno'],
axis=1).groupby(['Date','State/UnionTerritory']).sum().reset_index()
fulltable_india.columns = ['Date','State/Union','Deaths','Confirmed']
```

```
fulltable_india['Date'] = pd.to_datetime(fulltable_india['Date'])
fulltable_india
```

#for india

```
def display_fit_india(df, opt_confirmed, opt_deaths, ax):
    model_x = []

    isValid = True
    last_death_rate = 0

    for t in range(len(df)):
        model_x.append([df.index[t], model(*opt_confirmed, t),
model(*opt_deaths, t)])
```

#if deaths + recovered > confirmed or deaths rate > 5%, maybe not valid

```
        if (t>len(df)):
            last_row = model_x[-1]
            if(last_row[2]> last_row[1]*0.05):
                if (isValid):
                    last_row2 = model_x[-2]
                    last_death_rate = last_row2[2]/last_row2[1]
                    isValid=False
            if (last_row[2]> last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate

        model_sim = pd.DataFrame(model_x, dtype=int)
        model_sim.set_index(0, inplace=True)
        model_sim.columns = ['Model-Confirmed', 'Model-Deaths']
        plot_color = ['#99990077', '#FF000055', '#999900FF', '#FF0000FF']
        return pd.concat([model_sim,
df],axis=1).plot(ax=ax,figsize=(14,10),color = plot_color)



def display_extended_curve_india(df, opt_confirmed, opt_deaths, ax):
```

```
    start_date = df.index[0]
    n_days = len(df) + 40
    extended_model_x = []

    isValid = True
    last_death_rate = 0

    for t in range(n_days):
        extended_model_x.append([start_date + datetime.timedelta(days=t),
model(*opt_confirmed, t), model(*opt_deaths, t)])
```

#if deaths + recovered > confirmed or deaths rate > 5%, maybe not valid

```
        if (t > len(df)):
            last_row = extended_model_x[-1]
            if (last_row[2] > last_row[1]*0.05):
                if (isValid):
                    last_row2 = extended_model_x[-2]
                    last_death_rate = last_row2[2]/last_row2[1]
                    isValid = False

            if (last_row[2] > last_row[1]*0.05):
                last_row[2] = last_row[1]*last_death_rate




    extended_model_sim = pd.DataFrame(extended_model_x, dtype=int)
    extended_model_sim.set_index(0, inplace=True)
    extended_model_sim.columns = ['Model-Confirmed', 'Model-Deaths']

    plot_color = ['#99990077', '#FF000055', '#999900FF', '#FF0000FF']

    return pd.concat([extended_model_sim, df], axis=1).plot(ax=ax,
figsize=(14, 10), color = plot_color)

def opt_display_model_india(df, stats):
```

```
    # if the last data point repeats the previous one, or is lower, drop it
    if len(df) > 1 and df.iloc[-2,0] >= df.iloc[-1,0]:
        df.drop(df.tail(1).index,inplace=True)
    global model_index
    model_index = 0
    opt_confirmed = minimize(model_loss, x0=np.array([200000, 0.05, 15]),
method='Nelder-Mead', tol=1e-5).x
    model_index = 1
    opt_deaths = minimize(model_loss, x0=np.array([200000, 0.05, 15]),
method='Nelder-Mead', tol=1e-5).x
    if min(opt_confirmed) > 0:
        stats.append([country, *opt_confirmed, *opt_deaths])
        n_plot = len(stats)
        plt.figure(1)
        ax1 = plt.subplot(221)
        display_fit_us(df, opt_confirmed, opt_deaths, ax1)
        ax2 = plt.subplot(222)
        display_extended_curve_us(df, opt_confirmed, opt_deaths, ax2)
        plt.show()



def get_time_series_province_india(province):
    # for some countries, data is spread over several Provinces
    global fulltable_india
    df = fulltable_india[(fulltable_india['State/Union'] == province)]
    return df.set_index('Date')[['Confirmed', 'Deaths']]
```

#test delhi prediction

```
df = get_time_series_province_india('Delhi')
df
stats = []
print('{} of India COVID-19 Prediction'.format('Delhi'))
opt_display_model_india(df, stats)
```

#all provinces of india which confirm cases greater than 500

```
stats= []
for Province in sorted(fulltable_india['State/Union'].unique()):
```

```
if(Province==''):
    continue
df = get_time_series_province_india(Province)
if len(df)==0 or (max(df['Confirmed']) < 500):
    continue
print('{} of INDIA COVID-19 Prediction'.format(Province))
opt_display_model_india(df,stats)
```

6.Testing

6.1. Introduction to Testing

Testing goes side by side with the implementation that is aimed at ensuring that the program works accurately and efficiently before the live operation is performed. The common view of testing held by the user is process of executing a program with explicit intention of handling errors. The program which has been developed has to be tested to prove its validity.

Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases, and makes it shine. This program was tested using the following two techniques of application testing.

6.1.1. White Box Testing

a. By using this technique, it was tested that all the individual logical paths were executed at least once and every statement in the program was executed once during testing
b. All the logical decisions where tested on both their true and false sides
c. All the loops were tested with data in between the range and especially at the boundary values.

6.1.2. Black Box Testing

a. By the use of this technique, the missing functions were identified and placed in their position.

b. The errors in the program were identified and corrected.

c. The errors in the datasets were identified.

d. This technique was along used to identify the initialization and termination errors and correct them.

6.2 Testing approach

The answers lie in the data set. In order to test a machine learning algorithm, tester defines three different datasets viz. Training dataset, validation dataset and a test dataset (a subset of training dataset).Here, below is the basic approach a tester can follow in order to test the developed learning algorithm:

1. Tester first defines three datasets, training dataset(65%), validation dataset(20%) and test dataset(15%). Please randomize the dataset before splitting and do not use the validation/test dataset in your training dataset.

2. Tester once defines the data set, Will begin to train the models with the training dataset. Once this training model is done, the tester then performs to evaluate the models with the validation dataset. This is iterative and can embrace any tweaks/changes needed for a model based on results that can be done and re-evaluated. This ensures that the test dataset remains unused and can be used to test an evaluated model.

3. Once the evaluation of all the models is done, the best model that the team feels confident about based on the least error rate and high approximate prediction will be picked and tested with a test dataset to ensure the model still performs well and matches with validation dataset results. If you find the model accuracy is high then you must ensure that test/validation sets are not leaked into your training dataset.

7. Conclusion

I have investigated the problem of COVID-19 spread on global scale and also in various countries and their states/province (specially India and USA). A mathematical model has been established, which follows the actual data trend of COVID-19 spread. I collect data from various websites. Model prediction rate totally depend on provided data-sets.

Over the past two months, COVID-19 has emerged as a public health threat around the world. It adds to the list of previous epidemic infections disease outbreaks, including Bovine Spongiform Encephalitis in 1986, the Avian flu in 1997, the SARS in 2002, the Swine flu in 2009, and the Ebola in 2014. all these outbreaks remind us that we live in a habitat where it is necessary to respect the relationship between animal, social life, and the environment to survive and thrive. The global experience is teaching that containment measures and aggressive contract tracing are mandatory to keep the infection under control until an approved treatment or a vaccine is available to the global community. They should also minimize the economic burden of disease, and improve understanding of disease mechanisms, health problems, disease emergence, and reemergence to respond in a proportionate and timely manner. This will help in detecting, preventing, and combating future pandemics based on our experience from COVID-19 outbreaks.

This report shows how COVID-19 cases looks like in near future, in the absence of specific, effective treatment and given a lack of resources in managing active COVID-19 patients, prevention and early containment of the disease appear to be the most feasible option for countries. This report also shows that the confirmed cases keep increasing in world's two

largest democracies, India and USA. In USA, the numbers didn't look good and in case of India the numbers slowly rising.

The impact of COVID-19 has been also affecting sectors like entertainment, tourism, restaurants, and the travel industry, with a tremendous escalation of job losses. Disrupted supply chain and declining stock markets are the final consequences of these social changes, thus hitting the global economy.

In India, the first case of COVID-19 was reported on January 30th, 2020, followed by two similar cases on February 2nd and 3rd. All three had a travel history to Wuhan, China. A month later, on March 2nd, two new cases were reported – one each from New Delhi and Hyderabad. A sharp increase in numbers then followed. To contain the spread, the Ministry of Health and Family Welfare (MoHFW) immediately took action and issued a travel advisory, as travel restrictions had previously demonstrated efficacious on outbreaks of SARS, Ebola, and bubonic plague. All international travelers entering the country were asked to self-quarantine for 14 days. All travel visas to other countries were canceled until April 15th, 2020. All the states were asked to invoke the Epidemic Disease Act, which allowed officials to quarantine suspected cases and close down public places. An intensive campaign was rolled out and guidelines were developed for personal hygiene, surveillance, contact tracing, quarantine, diagnosis, laboratory tests, and management. People were advised not to visit farms, live animal markets or places where animals are slaughtered and to avoid mass gatherings. All the health care facilities were asked to stop regular out-patient and in-patient services and to continue with solely emergency services. Doctors were encouraged to use telemedicine services. Arogya Setu app was also launched to connect essential health services with people of India to fight against COVID-19. This app will reach out and inform the users of the risk, best practice and relevant advisories pertaining to containment of COVID-19. Amenities like hotels, colleges, railway train coaches, etc., were converted into quarantine facilities and large

public places as stadiums were converted into isolation wards to handle an anticipated increased number of cases. Some of the states converted existing hospitals to exclusively handle COVID-19 patients. On March 22nd, Prime Minister Narendra Modi initiated the lockdown process with a 14-hour 'Janta Curfew', followed by lockdown in 75 COVID-19 affected districts and a nationwide lockdown for the 3 weeks. A containment plan involving the State and twenty ministries was set up. A round-the-clock control room was set up at the headquarters of the General Director of Health Service (DGHS) to address the virus-related queries. The countries of the South Asian Association for Regional Cooperation (SAARC) were invited to fight jointly against this pandemic and 10 million US dollars were allocated for SAARC countries. A huge evacuation program of many Indian nationals was done from the COVID-19 affected areas.

In the United States (U.S.), there was a lack of coordination in the national response, with an unclear message from the U.S. President with often a variance with information from the U.S. National Institutes of Health (NIH) and the Centers for Disease Control (CDC). Early and severe outbreaks in Washington State and New York State prompted the six counties in the San Francisco Bay Area to impose stay-at-home orders on March 17th, 2020 in response to initial cases. Two days later, the entire state of California was placed under these orders to "flatten the curve" of new severe cases. The 50 U.S. states and many localities within them individually imposed measures ranging from tight provisions as in California to more lenient approaches such as bans on gatherings and curfews. There was no consistent national policy. This chaotic pattern included a wide variation in travel restrictions and even quarantines for travelers from one state to another and travel restrictions. Even in the absence of enforced restrictions, social distancing was emphasized and those feeling ill were urged to stay home. Early in the epidemic, policy on investigating contacts and quarantining them varied from city to city.

In conclusion, India and the world have a long legacy of successful efforts to prevent or cure widespread infections. Recalling the successful smallpox eradication campaign of the 1970s, we are reminded of the pivotal role of leadership and sound management to stop the killer disease. Hence, India has the opportunity to reverse disease predictions with stringent containment measures, social distancing, increasing case detection, isolation and quarantining the contacts. Efforts should be made to enlist community support and ownership so containment measures do not depend exclusively on administrative measures; critical situations like a mass exodus of laborers from the major cities to the rural areas as well as mass gatherings like Nizamuddin Markaz event which happened recently should be avoided.

Looking into the near future, containing the COVID-19 epidemic is likely to take several months; public health interventions will be directed towards social distancing and improving hygienic practices. These interventions will be effective in delaying the onset of wide community transmission, reducing peak incidence and its impact on public services. Testing, contact tracing, isolation of infected, and precautionary self-isolation of contacts is critical in reducing the number of new cases. An exceptionally high degree of understanding in the population and acceptance of these measures is also critical. These interventions have to be balanced with getting back to normal life and everyday activities to the best extent possible until a reversing the trajectory of the pandemic is traced.