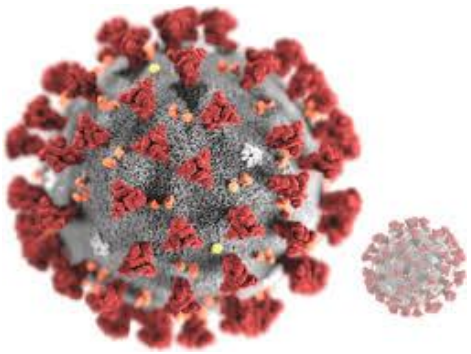
REPORT ON TRACKING COVID-19 SPREAD



Viral pandemics are a serious threat.

COVID-19 is not the first,

and it won't be the last

COVID-19

In December 2019, a novel coronavirus was found in a seafood wholesale market in Wuhan, China. WHO officially named this coronavirus as COVID-19. Since the first patient was hospitalized on December 12, 2019, China has reported a total of 78,824 confirmed CONID-19 cases and 2,788 deaths as of February 28, 2020. Wuhan's cumulative confirmed cases and deaths accounted for 61.1% and 76.5% of the whole China mainland , making it the priority center for epidemic prevention and control. Meanwhile, 51 countries and regions outside China have reported 4,879 confirmed cases and 79 deaths as of February 28, 2020. COVID-19 epidemic does great harm to people's daily life and country's economic development. This paper adopts three kinds of mathematical models, i.e., Logistic model, Bertalanffy model and Gompertz model. The epidemic trends of SARS were first fitted and analyzed in order to prove the validity of the existing mathematical models. The results were then used to fit and analyze the situation of COVID-19. The prediction results of three different mathematical models are different for different parameters and in different regions. In general, the fitting effect of Logistic model may be the best among the three models studied in this paper, while the fitting effect of Gompertz model may be better than Bertalanffy model. According to the current trend, based on the three models, the total number of people expected to be infected is 49852-57447 in Wuhan,12972-13405 in non-Hubei areas and 80261-85140 in China respectively. 2020 in Wuhan and before late-March, 2020 in other areas respectively.

Prediction and analysis of covid-19

ML technology plays an important role in helping the nation in this Panedemic. Such as:

- Identify who is most at risk,
- Diagnose patients,
- Develop drugs faster,
- Finding existing drugs that can help
- Predict the spread of the disease,
- Understand viruses better,
- Map where viruses come from, andS

So, Based on Machine Learning using python, A code is developed which can track the spread of Covid-19. Also, By using this data I applied different regression techniques to predict the future data of any given date by the user.

Early Prediction Model of Infectious Diseases Based on Machine Learning

In short, machine learning is known as to learn more useful information from a large amount of data using its own algorithm model for specific problems. Machine learning spans a variety of fields, such as medicine, computer science, statistics, engineering technology, psychology, etc. For example, neural network, a relatively mature machine learning algorithm, can simulate any high-dimensional non-linear optimal mapping between input and output by imitating the processing function of the biological brain's nervous system. When faced with complex data relations, the traditional statistical method is not such effective, which may not receive accurate results as the neural network. Since most new infectious diseases occurring in human beings are of animal origin (animal infectious diseases), it is an effective prerequisite to predict diseases by determining the common intrinsic characteristics of species and environmental conditions that lead to the overflow of new infections.. However, the highly nonlinear and complex problems to be analyzed in the early prediction model of infectious diseases based on machine learning usually lead to local minima and global minima, leading to some limitations of the machine learning model.

Model Selection

Logistic model:

Logistic model is mainly used in epidemiology. It is commonly to explore the risk factors of a certain disease, and predict the probability of occurrence of a certain disease according to the risk factors. We can roughly predict the development and transmission law of epidemiology through logistic regression analysis

$$Q(t)=\frac{a}{1+e^{b-c(t-t0)}}$$

Qt is the cumulative confirmed cases (deaths);
a is the predicted maximum of confirmed cases (deaths).
b and c are fitting coefficients.
t is the number of days since the first case.
T0 is the time when the first case occurred.

Model Evaluation:

The regression coefficient (R2) is used to evaluate the fitting ability of various methods and can be obtained by the following equation.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

y_i is the actual cumulative confirmed COVID-19 cases; $\hat{y}_i$ is the predicted cumulative confirmed COVID-19 cases; $\bar{y}$ is the average of the actual cumulative confirmed COVID-19 cases. The closer the fitting coefficient is to 1, the more accurate the prediction.

TRACKING COVID-19 SPREAD IN INDIA

Code:

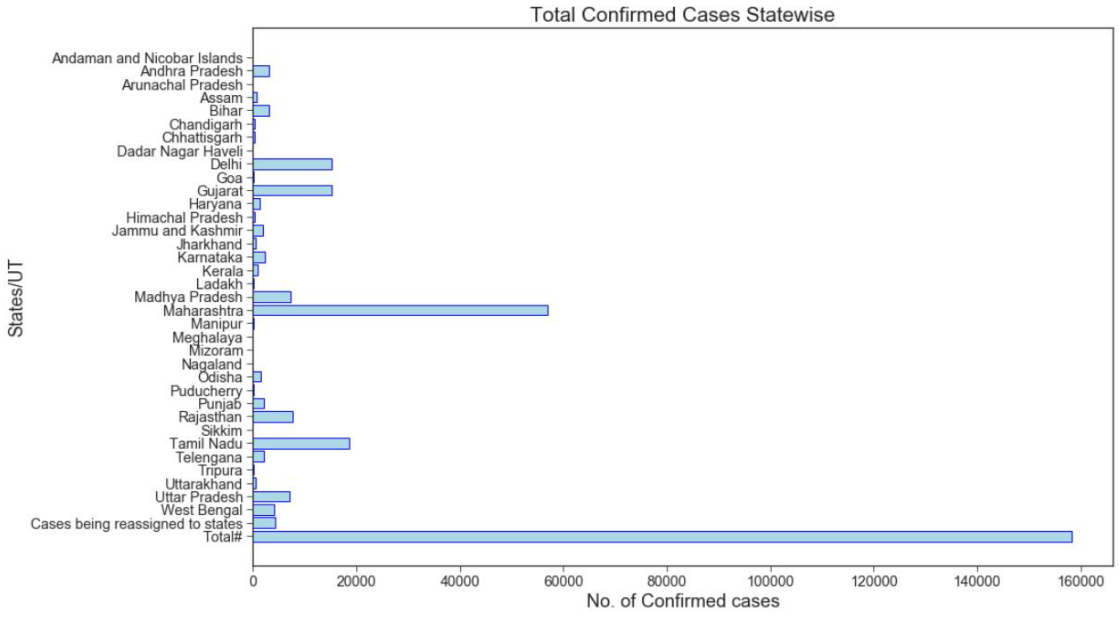
In [2]:

```
import requests
from bs4 import BeautifulSoup
from tabulate import tabulate
import os
import numpy as np
import matplotlib.pyplot as plt

extract_contents = lambda row: [x.text.replace('\n', '') for x in row]
URL = 'https://www.mohfw.gov.in/'

SHORT_HEADERS = ['SNo', 'State','Indian-Confirmed(Including Foreign Confirmed)','Cured','Death']

response = requests.get(URL).content
soup = BeautifulSoup(response, 'html.parser')
header = extract_contents(soup.tr.find_all('th'))

stats = []
all_rows = soup.find_all('tr')

for row in all_rows:
    stat = extract_contents(row.find_all('td'))

    if stat:
        if len(stat) == 4:
            # last row
            stat = ['', *stat]
            stats.append(stat)
        elif len(stat) == 5:
            stats.append(stat)

stats[-1][0] = len(stats)
stats[-1][1] = "Total Cases"

objects = []
for row in stats :
 objects.append(row[1])

y_pos = np.arange(len(objects))
```

```
performance = []
for row in stats[:len(stats)-1] :
 performance.append(int(row[2]))

performance.append(int(stats[-1][2][:len(stats[-1][2])-1]))

table = tabulate(stats, headers=SHORT_HEADERS)
print(table)

plt.barh(y_pos, performance, align='center', alpha=0.5,
color=(234/256.0, 128/256.0, 252/256.0),
edgecolor=(106/256.0, 27/256.0, 154/256.0))

plt.yticks(y_pos, objects)
plt.xlim(1,performance[-1]+1000)
plt.xlabel('Number of Cases')
plt.title('Corona Virus Cases')
plt.show()
```

Output:

SNo	State	Indian-Confirmed(Including Foreign Confirmed)	Cured	Death
1	Andaman and Nicobar Islands	33	33	0
2	Andhra Pradesh	3171	2057	58
3	Arunachal Pradesh	2	1	0
4	Assam	781	87	4
5	Bihar	3061	1083	15
6	Chandigarh	279	187	4
7	Chhattisgarh	369	83	0
8	Dadar Nagar Haveli	2	0	0
9	Delhi	15257	7264	303
10	Goa	68	37	0
11	Gujarat	15195	7549	938
12	Haryana	1381	838	18
13	Himachal Pradesh	273	70	5
14	Jammu and Kashmir	1921	854	26
15	Jharkhand	448	185	4
16	Karnataka	2418	781	47
17	Kerala	1004	552	7
18	Ladakh	53	43	0
19	Madhya Pradesh	7261	3927	313
20	Maharashtra	56948	17918	1897
21	Manipur	44	4	0
22	Meghalaya	20	12	1
23	Mizoram	1	1	0
24	Nagaland	4	0	0
25	Odisha	1593	733	7
26	Puducherry	46	12	0
27	Punjab	2139	1918	40
28	Rajasthan	7703	4457	173
29	Sikkim	1	0	0
30	Tamil Nadu	18545	9909	133
31	Telengana	2098	1284	63
32	Tripura	230	165	0
33	Uttarakhand	469	79	4
34	Uttar Pradesh	6991	3991	182
35	West Bengal	4192	1578	289
	Cases being reassigned to states	4332		
37	Total Cases	158333	67692	4531

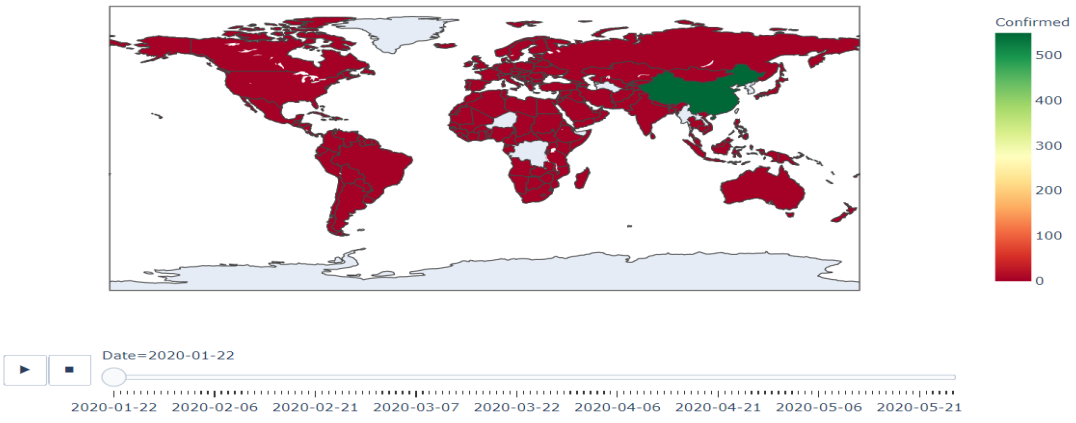


Link : http://localhost:8889/nbconvert/html/tracking%20in%20india%20covid-19%20spread.ipynb?download=false

TRACKING COVID-19 WORLDWIDE

In [1]:
```
import pycountry
import plotly.express as px
import pandas as pd
# ----------- Step 1 ------------
URL_DATASET = r'https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'
df1 = pd.read_csv(URL_DATASET)
# print(df1.head) # Uncomment to see what the dataframe is like
# ----------- Step 2 ------------
list_countries = df1['Country'].unique().tolist()
# print(list_countries) # Uncomment to see list of countries
d_country_code = {}  # To hold the country names and their ISO
for country in list_countries:
    try:
        country_data = pycountry.countries.search_fuzzy(country)
        # country_data is a list of objects of class pycountry.db.Country
        # The first item  ie at index 0 of list is best fit
        # object of class Country have an alpha_3 attribute
        country_code = country_data[0].alpha_3
        d_country_code.update({country: country_code})
    except:
        print('could not add ISO 3 code for ->', country)
        # If could not find country, make ISO code ' '
        d_country_code.update({country: ' '})

# print(d_country_code) # Uncomment to check dictionary

# create a new column iso_alpha in the df
# and fill it with appropriate iso 3 code
for k, v in d_country_code.items():
    df1.loc[(df1.Country == k), 'iso_alpha'] = v

# print(df1.head)  # Uncomment to confirm that ISO codes added
# ----------- Step 3 ------------
fig = px.choropleth(data_frame = df1,
                    locations= "iso_alpha",
                    color= "Confirmed",  # value in column 'Confirmed' determines color
                    hover_name= "Country",
                    color_continuous_scale= 'RdYlGn',  #  color scale red, yellow green
                    animation_frame= "Date")

fig.show()
```

```
could not add ISO 3 code for -> Congo (Brazzaville)
could not add ISO 3 code for -> Congo (Kinshasa)
could not add ISO 3 code for -> Diamond Princess
could not add ISO 3 code for -> Korea, South
could not add ISO 3 code for -> Laos
could not add ISO 3 code for -> MS Zaandam
could not add ISO 3 code for -> Taiwan*
could not add ISO 3 code for -> West Bank and Gaza
```



Link: http://localhost:8889/nbconvert/html/covid%20-19%20spread%20tracker%20worldwide.ipynb?download=false

Note: This link contains the code with the output.

Conclusion:

Machine learning is an important tool in fighting the current pandemic. If we take this opportunity to collect data, pool our knowledge, and combine our skills, we can save many lives – both now and in the future.

References:

[1]. https://app.developer.here.com/coronavirus/

[2]. covid19india.org.

[3]. https://en.wikipedia.org/wiki/COVID-19_apps

Submitted by: Nidhi sharma

Email: nidhi2556jmu@gmail.com