ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Data Modelling and Analysis of COVID-19 Cases & Epidemic In Raipur,INDIA

Hemlata Sinha[1], Sumit Kumar Roy[2], Arvind Singh Solanki[3], Aakarsh Shrivastva[4] , Mahak Khatwani[5]

Associate Professor [1] , Assistant Professor[2] ,UG Student[3,4,5]

[1,2,3,4]Department of Electronics & Telecommunication Engineering, SSIPMT,Raipur

[1,2,3,4]Shri Shankaracharya Institute Of Professional Management And Technology Raipur CSVTU,Bhilai

(Corresponding author: Dr. Hemlata Sinha)

Abstract - This paper attempts to provide an approximate estimation of new COVID-19 cases. It uses predictive data model using machine learning. The effect of lockdown was also considered in the estimation of possible new cases. this paper mainly focuses on data collection, modelling of data, analysis of parameters, predictive modelling, data screening and final predictive model. Preparation and representation of data were the focal points of the discussion in attempt to tie them all inside engineering and how they affects the overall environment or society.

Keywords - COVID19, Lockdown, python, NumPy, pandas, matplotlib, flask, pickle, folium, sklearn, Regression, Polynomial regression.

I. INTRODUCTION

Coronavirus disease is a new and contagious disease caused by a new virus, known as the novel coronavirus. The disease affects the lungs and causes a respiratory illness with symptoms like the flu such as cold, throat infection, cough, fever, and in critical cases, difficulty in breathing. The active period of the novel coronavirus is of fourteen days. It is suggested by medical authorities that one can protect themself by washing hands frequently, avoiding touching the nose, ears, and face, and maintaining social distancing (for 3 feet or 4 steps between them) with other people. World Health Organization (WHO) declares that COVID-19 is a pandemic and releases guidelines to help countries maintain essential health services during the COVID-19 pandemic on 11 March 2020.[1]

This disease originated in China in December 2019 and has already caused havoc around the world, including India. In India, the first case was detected on 27 January 2020 and the number of cases from 02 February 2020 to 01 March 2020 remained three but after from 02 March onward cases are being regularly reported. There was a rapid increase in the number of cases on 04 March 2020 due to various reasons, one of those being changes in testing policy.

This outbreak has caused a global pandemic with more than 2,40,40.851 positive cases and 2,62,239 deaths (Data as per 14/05/2021 in India) [3]. India has witnessed its first positive case on 27 January 2020, in Kerala, IN. Respectively, in Chhattisgarh, Raipur, IN the first positive case found on 19th March 2020. A 24-year-old woman has been tested positive for covid19 in here, making her the first confirmed case [2] with a travel history from London, UK via Mumbai to Raipur airport.

Many proactive steps are taken by the government to control the spread of disease, like after getting the first case of covid19 in Raipur. CM Bhupesh Baghel imposes the Raipur shutdown and The Chhattisgarh government on Thursday i.e., 19/05/2020, imposed Section-144, a law that prohibits gatherings, across the state, and ordered the closure of malls, supermarkets, restaurants, clubs, and street vendors.[4] Other officials including lockdown, social awareness, and identification of clusters of cases. Complete lockdown of the nation for 21 days, containment zones/areas, and immediate isolation of infected cases are the proactive steps taken by the authorities. The virus spread in a chain which makes it unstable to get control easily.

Modeling the Covid-19 pandemic spread is challenging. But there are data that can be used to project resource demands. Estimates of the reproductive number (R) of SARS-CoV-2 show that at the beginning of the

epidemic, each infected person spreads the virus to at least two others, on average (Emanuel et al. in N Engl J Med. 2020, Livingston and Bucher in JAMA 323(14):1335, 2020). A conservatively low estimate is that 5% of the population could become infected within 3months. Preliminary data from China and Italy regarding the distribution of case severity and fatality vary widely (Wu and McGoogan in JAMA 323(13):1239–42, 2020). A recent large-scale analysis from China suggests that 80% of those infected either are asymptomatic or have mild symptoms; a finding that implies that demand for advanced medical services might apply to only 20% of the total infected. Of patients infected with Covid-19, about 15% have severe illness and 5% have a critical illness (Emanuel et al. in N Engl J Med. 2020). Overall, mortality ranges from 0.25% to as high as 3.0% (Emanuel et al. in N Engl J Med. 2020, Wilson et al. in Emerg Infect Dis 26(6):1339, 2020). Case fatality rates are much higher for vulnerable populations, such as persons over the age of 80 years (>14%) and those with coexisting conditions (10% for those with cardiovascular disease and 7% for those with diabetes) (Emanuel et al. in N Engl J Med. 2020). Overall, Covid-19 is substantially deadlier than seasonal influenzas, which has a mortality of roughly 0.1%.

Public health depends heavily on predicting how diseases such as those caused by Covid-19 spread across the world. During the first days of a replacement outbreak, when reliable data are still scarce, researchers address mathematical models which will predict where people that might be infected are going and the way likely they're to bring the disease with them. These computational methods use known statistical equations that calculate the probability of people transmitting the illness. Modern computational power allows these models to quickly incorporate multiple inputs, like a given disease's ability to pass from person to person and therefore the movement patterns of probably infected people traveling by air and land. This process sometimes involves making assumptions about unknown factors, like an individual's exact travel pattern. By plugging in several possible versions of every input. However, researchers can update the models as new information becomes available and compare their results to observed patterns for the illness.

Covid19 is an extremely contagious virus. In addition, many infected individuals show mild or no symptoms of infection. The epidemic has spread unwittingly via these individuals, who are called asymptomatic carriers. In a study, Li et al. [3] segregated the documented (reported) cases and undocumented cases (asymptomatic carriers) of 375 cities of China, and showed that "undocumented infections often experience mild, limited or no symptoms and hence go unrecognized, and, counting on their contagiousness and numbers, can expose a far greater portion of the population to the virus than would otherwise occur."

Epidemiologists have made various models for understanding and forecast epidemics. Kermack and McKendrick [4] constructed one of the first models, called the SIR model, where, the variables S and I describe respectively the numbers of susceptible and infected individuals. The variable R represents the removed individuals who have either recovered or died., the SIR model has been generalized to the SEIR model that includes exposed individuals, E, who are infected but not yet infectious [5, 6].

More complex models of epidemiology include features of quarantine, lockdowns, stochasticity, interactions among population pockets, etc. Note that quarantines and lockdowns help in suppressing the maximum number of the infected individuals; such steps are critical for the epidemic management with limited public health resources. The saturation or flattening of the curve in China is attributed to strong lockdowns.

For COVID-19 epidemic, some of the new models have managed to provide good forecasts that appears to match with the data. Peng et al. [7] constructed a seven-variable model (including quarantined and death variables) for epidemic spread in China and predicted that the daily count of exposed and infectious individuals will be negligible by 30 March 2020. Their predictions are in good agreement with the present data. Wang et al. employed another model and studied the effects of epidemic on various age groups. Using different models, Labadin and Hong [9] and Shim et al. [10] studied the COVID-19 epidemic in Malaysia and South Korea respectively.

Kucharski et al. [11] and Roosa et al. [12] employed epidemic models for creating short-term predictions in China. Chinazzi et al. [13] studied the consequences of travel restrictions on the spread of COVID-19 in China and within the world. Hellewell et al. [14] performed feasibility studies of controlling the covid19 epidemic by isolation. Mandal et al. [15] constructed the India-specific model for devising intervention strategies; they focussed on four metros - Delhi, Mumbai, Kolkata, and Bengaluru—along with intercity connectivity. To account for spatiotemporal behaviour, Min et al. [16] simulated how a disease could spread within a network with different mixing styles and showed that the typical epidemic size and speed depend critically on network parameters. Meyer and Held [17] studied the effect of power-law movements of humans on the disease spread. In addition, there are many epidemic

models that are inspired by increase models [6].

The mathematical modelling of this evolving pandemic in India has been attempted by a good range of researchers from the very beginning of cases in India. An initial analysis of those models regarding India revealed large variations in scope, assumptions, prediction on numbers, the course of the pandemic in India, effect of varied interventions, effect on health care services, and so on.

The illustration, which is created at the Centres for Disease Control and Prevention (CDC), reveals ultrastructural morphology exhibited by coronaviruses [19]. a completely unique coronavirus, named Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV-2), was identified because the reason for a plague of respiratory illness first detected in Wuhan, China in 2019. The illness caused by this virus has been named coronavirus disease 2019 (COVID-19).

In this paper, we describe a model of covid19 spread by using innovative big data analytics techniques and tools. The study points out the key factors that affect the spread of the virus, such as the number of cases with dates, virus and daily infection, recovered and decreased in a number of cases. Relevant models and data analysis can provide some basis and guidance for the related location about epidemic prevention and control.

Amidst such an important ongoing public health crisis that also has severe economic repercussions, we reverted to mathematical modeling that can shed light on essential epidemiologic parameters that determine the fate of the epidemic. Here, we present the results of the analysis of time series of epidemiological data available in the public domain (WHO, CDC, ECDC, NHC, and DXY) from March 18, 2020, to June 14, 2021, and attempt an n-days (for e.g., n=2) forecast of the spreading dynamics of the emerged total cases of coronavirus epidemic in Raipur, IN with data from date March 18, 2020, to July 28, 2020, for the July 31, 2020 day. The other predictive model is a probability of a person affected by COVID19 using parameters of symptoms and present in a manner of user-friendly hand. For our analysis, we employed the real-time data available at worldometer [8]. A similar data set is available at the Corona Resource Center of John Hopkins University,[20]. This model may be used for other countries as well. The proposed model can be used to predict the stage of covid19 also by matching the available data with analytical results. Further, the other statistics like the number of deaths or the number of recovered cases can also be predicted with sufficient accuracy.

Hence, modeling and forecast this epidemic are of critical importance. Here, we analyse the publicly available data set of the epidemic. This feature can be used as an important diagnostic for flattening the curve.

II. PYTHON

Python is an interpreted, high-level and general-purpose programing language. Python's design philosophy emphasizes code readability with its notable use of serious indentation. Its language constructs and object-oriented approach aim to assist programmers to write clear, logical code for little and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming.

Python has usually described as a "batteries included" language thanks to its comprehensive standard library. Python was created within the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, like list comprehensions, and a garbage pickup system with reference counting, and was discontinued version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. With Python 2's end-of-life (and pip having dropped support in 2021), only Python 3.6.x and later are supported, with older versions still supporting e.g., Windows 7 (and old installers not restricted to 64-bit Windows).7 (and old installers not restricted to 64-bit Windows).

Python interpreters are supported for mainstream operating systems and available for a couple of more (and within the past supported many more). A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. As of January 2021, Python ranks third in TIOBE’s index of hottest programming languages, behind C and Java having previously gained second place and their award for the most popularity gain for 2020.

Python's large standard library commonly cited together of its greatest strengths provides tools suited to several tasks. For Internet-facing applications, many standard formats and protocols like MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and

unit testing.8 Some parts of the standard library are covered by specifications (for example, the online Server Gateway Interface (WSGI) implementation follows PEP 333), but most modules aren't. They are specified by their code, internal documentation, and test suites. However, because most of the quality library is cross-platform Python code, only a couple of modules need altering or rewriting for variant implementations.

Python libraries used here are; *NumPy, pandas, matplotlib, flask, pickle, folium, sklearn*

[1] NUMPY: NumPy may be a Python library used for working with arrays. It also has functions for working within the domain of algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you'll use it freely. In Python, we've lists that serve the aim of arrays, but they're slow to process. NumPy aims to supply an array object that's up to 50x faster than traditional Python lists. The array object in NumPy is called an array, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently utilized in data science, where speed and resources are vital. NumPy stands for Numerical Python.

[2] PANDAS: Pandas could also be a Python library used for working with data sets. it's functions for analysing, cleaning, exploring, and manipulating data. The name "Pandas" features a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allow us to research big data and make conclusions supported statistical theories. Pandas can clan messy data sets and make them readable and relevant, Relevant data is extremely important in Data Science.

[3] PLOTLY: The Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a good range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

[4] FOLIUM: Folium may be a Python library used for visualizing geospatial data. It is easy to use and yet a strong library. Folium may be a Python wrapper for Leaflet.js which may be a leading open-source JavaScript library for plotting interactive maps. Folium builds on the info wrangling strengths of the Python ecosystem and therefore the mapping strengths of the Leaflet.js library. Manipulate the data in Python, then visualize it in on a Leaflet map via Folium.

Folium makes it easy to visualize data that's been manipulated in Python on an interactive Leaflet map. The location gets selected by the use of latitude and longitude. (lat. And long of Raipur; 21.2514, 81.6296)

[5] FLASK: Flask may be a micro web framework written in Python. It is classified as a micro-framework because it doesn't require particular tools or libraries. It has no database abstraction layer, form validation, or other components where pre-existing third-party libraries provide common functions.

[6] PICKLE: Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on a disk. What pickle does is that it "serializes" the thing first before writing it to file. Pickling may be a thanks to convert a python object (list, dict, etc.) into a personality stream. The idea is that this character stream contains all the knowledge necessary to reconstruct the thing in another python script.

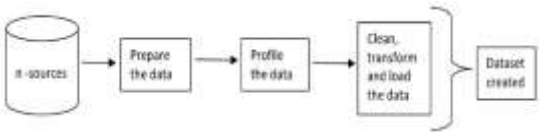
[7] SKLEARN: Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib

Open source, commercially usable - BSD license.

III. METHODOLOGY

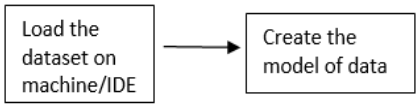
i. Collection of data / preparation of dataset



Firstly, research on the data(information) is done. Like what kind of data is need with varies in parameters, after that various source are been search this search consists of websites, data warehouses, and various data-set libraries. After selecting the data from various sources, all data are collected at one place then profiling is done, naming and nomenclature after that the data is been read like information present inside there are the correct, is there any void space, duplicated data, mismatched name/data. Now, our data is understandable and now it is called a dataset. Basically, the dataset is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question.

ii. Model the data

- Design a data model
- Deploy a data model with parameters
- To optimise the data

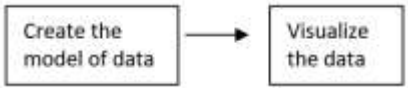


Code to load the dataset to machine;

```
data = pd.read_csv('March-April.2020.csv')
data
```

Creating the model of data from the dataset is the first step towards solving the problem. Basically, to understand the data more fairly and can further be optimised with various parameters to find the result and analyse. (IDE, we use here are Jupiter notebook, VS studio and for coding python language is used).iii. Visualization of data

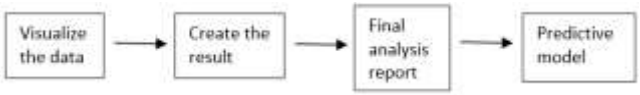
- Create reports
- Create the dashboard
- Model view, configure, n-factors



Visualization of data can be done by using inbuilt python libraries which are; *pandas*, *NumPy*, *matplotlib, plotly.express and folium* for visualization over map and creation of dashboard.

iv. Analyse the data and Predictive model

- Slicing
- Filter
- Statistical summary



IV. PARAMETERS [2][7] [20]

- Lockdown 1: 22/07/2020 - 06/08/2020 (16+6 days)
- Lockdown 2: 21/09/2020 - 28/09/2020 (7 days)
- Lockdown 3: 27/12/2020 - 02/01/2021(7 days)
- Lockdown 4: 04/04/2021 - 31/05/2021 (27 days)
- Quarantine zoning and Section- 144: 19/05/2020 (still)
- Face mask and social distancing made compulsory from, 01/06/2020.
- Movement stops from one district to another: 05/04/2021 to 19/04/2021 (13 days)

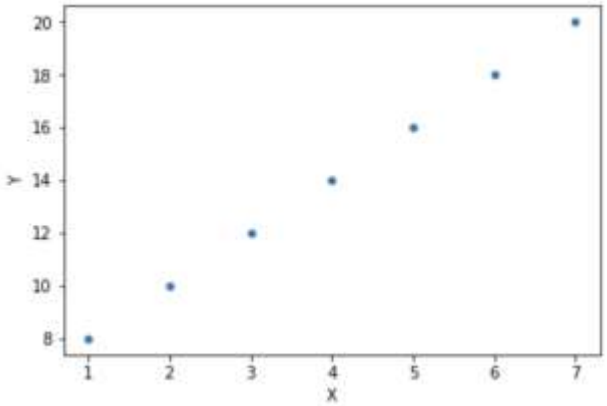
V. PREDICTIVE MODEL

Predictive modelling is the process of taking known results and developing a model that can predict values for new occurrences. It uses historical data to predict future events. There are many different types of predictive modelling techniques including *ANOVA, linear regression (ordinary least squares), logistic regression, ridge regression, time series, decision trees, neural networks, and many more*. Selecting the correct predictive modelling technique at the start of your project can save a lot of time. Choosing the incorrect modelling technique can result in inaccurate predictions and residual plots that experience non-constant variance and/or mean.[12][13]

i. Regression Analysis

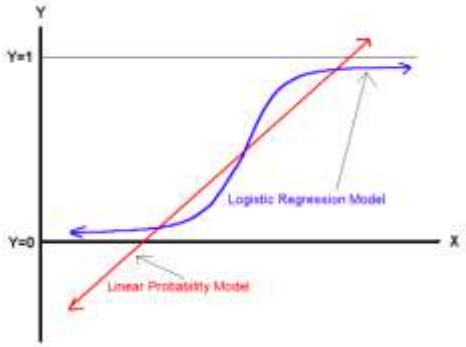
Regression analysis is employed to predict endless target variable from one or multiple independent variables. Typically, multivariate analysis is employed with naturally-occurring variables, instead of variables that are manipulated through experimentation.

ii. Linear Regression

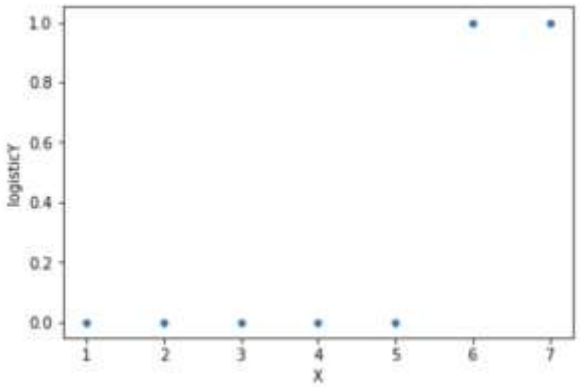


Linear regression is to be used when the target variable is continuous and therefore the dependent variable(s) is continuous or a mix of continuous and categorical, and therefore the relationship between the independent variable and dependent variables are linear. All the predictor variables should be normally distributed with constant variance and should demonstrate little to no multicollinearity nor autocorrelation with one another.

iii Logistic Regression



Logistic regression doesn't require a linear relationship between the target and therefore the dependent variable(s). The target variable is binary (assumes a worth of either 0 or 1) or dichotomous. The errors of a logistic regression need not be normally distributed and the variance of the residuals does not need to be constant. However, the dependent variables are binary, the observations must be independent of every other, there must be little to no multicollinearity nor autocorrelation within the data, and therefore the sample size should be large. Lastly, while this analysis doesn't require the independent and dependent variable(s) to be linearly related, the independent variables must be linearly associated with the log odds.



If the scatter plot between the independent variable(s) and the dependent variable looks like the plot above, a logistic model might be the best model to represent that data.

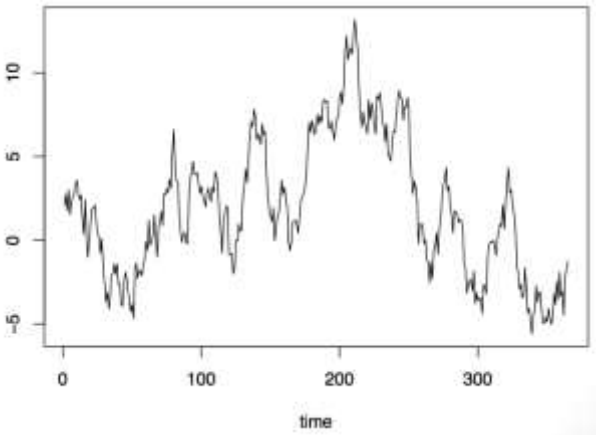
iv. Ridge Regression

For variables that have high multicollinearity, like X1 and X2 during this case, a ridge regression could also be the simplest choice so as to normalize the variance of the residuals with an error term.

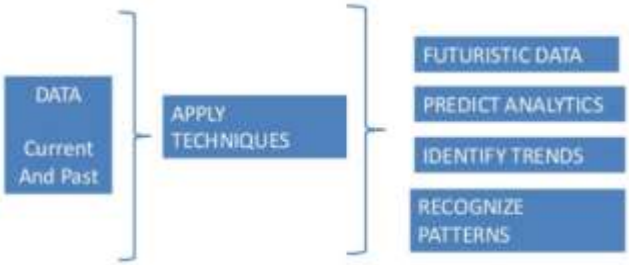
Ridge regression is a technique for analysing multiple regression variables that experience multicollinearity. It takes the ordinary least squares approach, and honors that the residuals experience high variances by adding a degree of bias to the regression estimates to reduce the standard errors. The assumptions follow those of multiple correlations, the scatter plots must be linear, there must be constant variance with no outliers, and therefore the dependent variables must exhibit independence.

v. Time Series

Time-series regression analysis is a method for predicting future responses based on response history. The data for a time series should be a set of observations on the values that a variable takes at different points in time. The data is bivariate and the independent variable is time.



A. Prediction of Covid19 cases for n numbers of days using polynomial regression.



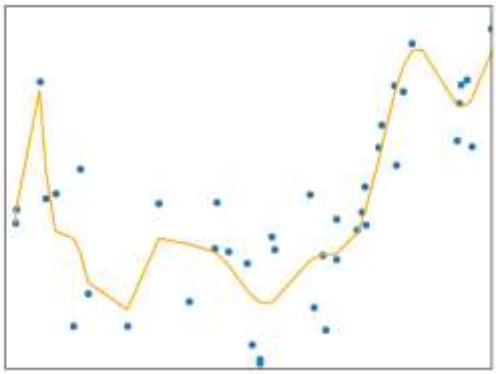
i. Regression

In statistical modeling, multivariate analysis may be a set of statistical processes for estimating the relationships between a variable and one or more independent variables. for instance, if parents were very tall the youngsters attended be tall but shorter than their parents. If parents were very short the youngsters attended be short but taller than their parents were. This discovery he called "regression to the mean",

ii. Polynomial regression

Linear regression requires the relation between the variable and therefore the experimental variable to be linear. If the distribution of the info was more complex as

shown within the below figure? Can linear models be wont to fit non-linear data? How can we generate a curve that best captures the info as shown below? Well, we will answer these questions here.

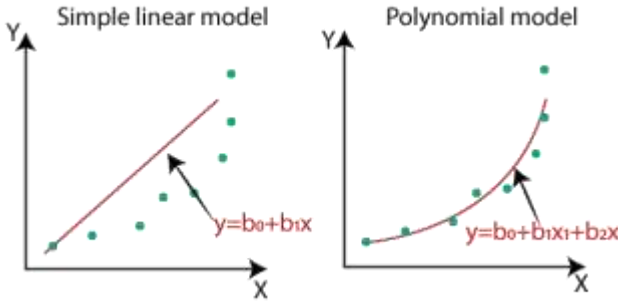


Polynomial Regression may be a regression algorithm that models the connection between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+..... b_nx_1^n$$

It is also called the special case of multiple rectilinear regression in ML, because we add some polynomial terms to the Multiple rectilinear regression equation to convert it into Polynomial Regression. it's a linear model with some modification so on extend the accuracy. The dataset utilized in Polynomial regression for training is of non-linear nature. It makes use of a linear regression model to suit the complicated and non-linear functions and datasets.[16][17]

In the below image, we've taken a dataset which is arranged non-linearly. So, if we attempt to cover it with a linear model, then we will clearly see that it hardly covers any datum. On the opposite hand, a curve is suitable to hide most of the info points, which is of the Polynomial model.



Hence, if the datasets are arranged during a non-linear fashion, then we should always use the Polynomial Regression model rather than Simple rectilinear regression.

If we apply a linear model on a linear dataset, then it provides us an honest result as we've seen in Simple rectilinear regression, but if we apply the same model with none modification on a non-linear dataset, then it'll produce a drastic output. Due to which loss function will increase, the error rate is going to be high, and accuracy are going to be decreased.

So, for such cases, where data points are arranged during a non-linear fashion, we'd like the Polynomial Regression model. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset with the concept of polynomial regression.

I train the machine with X_test, X_train with the first 30 datasets, and last 30 datasets, and data train and a predictive model get created. Now, as the exponential factor is the parameter so I defined n as the days. Likewise, if I want to predict the total_case after for example 2 days/ 10 days it will predict the graph and data and show the accuracy of the predictive model.

Furthermore, on the actual day both can be differential on data and by the graph. This kind of model helps society to well-prepared about the bad days or can be known about the graph and also can be predictive as the graph is going up cases are going to increase but if the difference is getting weaker which means soon the cases are going to be decreased.[13][14][15][16]

VI. DATA SCREENING

Data collection from sources

Here, we have the dataset of countries in time-series format. From where first we will fetch out our country, 'IN', then required district - Raipur, IN.

Fetching data from different source

```
In [1]: import pandas as pd
```

```
In [6]: source_2 = pd.read_csv('countries.csv')
        source_2
```

Out[6]:

	country	year	population
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460
...	...	...	...
1699	Zimbabwe	1987	9216418
1700	Zimbabwe	1992	10704340
1701	Zimbabwe	1997	11404948
1702	Zimbabwe	2002	11926563
1703	Zimbabwe	2007	12311143

1704 rows × 3 columns

```
In [11]: IN = source_2[source_2.country == 'India']
         IN
```

Out[11]:

	country	year	population
696	India	1952	372000000
697	India	1957	409000000
698	India	1962	454000000
699	India	1967	506000000
700	India	1972	567000000
701	India	1977	634000000
702	India	1982	708000000
703	India	1987	788000000
704	India	1992	872000000
705	India	1997	959000000
706	India	2002	1034172547
707	India	2007	1110396331

Now, that we have find out India, now from India we fetch out all data from there and save it as ‘districts.csv’ file.

Fetching data from different source

```
In [1]: import pandas as pd
```

```
In [3]: source_3 = pd.read_csv('districts.csv')
        source_3
```

Out[3]:

	Date	State	District	Confirmed	Recovered	Deceased	Other	Tested
0	26-04-2020	Andaman and Nicobar Islands	Unknown	33	11	0	0	2679.0
1	26-04-2020	Andhra Pradesh	Anantapur	53	14	4	0	NaN
2	26-04-2020	Andhra Pradesh	Chittoor	73	11	0	0	NaN
3	26-04-2020	Andhra Pradesh	East Godavari	30	12	0	0	NaN
4	26-04-2020	Andhra Pradesh	Guntur	214	29	8	0	NaN
...	...	...	...	...	...	...	...	...
241554	07-05-2021	West Bengal	Purba Bardhaman	23150	18586	714	0	NaN
241555	07-05-2021	West Bengal	Purba Medinipur	21643	20690	382	0	NaN
241556	07-05-2021	West Bengal	Purulia	15395	12188	85	0	NaN
241557	07-05-2021	West Bengal	South 24 Parganas	19196	16370	641	0	NaN
241558	07-05-2021	West Bengal	Uttar Dinajpur	11601	9797	109	0	NaN

241559 rows × 8 columns

2) *Cleaning the data*

As we have data from all the districts, in that we go for ‘Raipur’.

```
In [20]: Raipur = source_3[source_3.District == 'Raipur']
         Raipur
```

Out[20]:

	Date	State	District	Confirmed	Recovered	Deceased	Other	Tested
41	26-04-2020	Chhattisgarh	Raipur	6	3	0	0	NaN
452	27-04-2020	Chhattisgarh	Raipur	6	3	0	0	NaN
878	28-04-2020	Chhattisgarh	Raipur	6	3	0	0	NaN
1306	29-04-2020	Chhattisgarh	Raipur	6	3	0	0	NaN
1740	30-04-2020	Chhattisgarh	Raipur	6	3	0	0	NaN
...	...	...	...	...	...	...	...	...
238390	03-05-2021	Chhattisgarh	Raipur	144387	131489	2544	0	12704.0
239044	04-05-2021	Chhattisgarh	Raipur	145395	132029	2583	0	12704.0
239698	05-05-2021	Chhattisgarh	Raipur	146311	132562	2647	0	12704.0
240353	06-05-2021	Chhattisgarh	Raipur	147298	133356	2692	0	12704.0
241009	07-05-2021	Chhattisgarh	Raipur	148119	134014	2732	0	12704.0

377 rows × 8 columns

First 5 data of dataset

```
data.head()
```

	DATE	DISTRICT	STATE	TOTAL_CASES	TODAYS_CASES	ACTIVE_CASES	RECOVERED_CASES	DECRESED_CASES
0	1/19/2020	RAIPUR	CHHATTISGARH	1	1	1	0	0
1	1/20/2020	RAIPUR	CHHATTISGARH	1	0	1	0	0
2	1/21/2020	RAIPUR	CHHATTISGARH	1	0	1	0	0
3	1/22/2020	RAIPUR	CHHATTISGARH	1	0	1	0	0
4	1/23/2020	RAIPUR	CHHATTISGARH	1	0	1	0	0

From last, 5 dataset

```
data.tail()
```

	DATE	DISTRICT	STATE	TOTAL_CASES	TODAYS_CASES	ACTIVE_CASES	RECOVERED_CASES	DECRESED_CASES
38	4/26/2020	RAIPUR	CHHATTISGARH	9	0	12	0	0
39	4/27/2020	RAIPUR	CHHATTISGARH	11	0	13	0	0
40	4/28/2020	RAIPUR	CHHATTISGARH	17	0	18	0	0
41	4/29/2020	RAIPUR	CHHATTISGARH	23	0	19	0	0
42	4/30/2020	RAIPUR	CHHATTISGARH	29	0	20	0	0

Reading columns of dataset

```
data.columns
```

```
Index(['DATE', 'DISTRICT', 'STATE', 'TOTAL_CASES', 'TODAYS_CASES',
       'ACTIVE_CASES', 'RECOVERED_CASES', 'DECRESED_CASES'],
      dtype='object')
```

Data information

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43 entries, 0 to 42
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   DATE             43 non-null     object
 1   DISTRICT         43 non-null     object
 2   STATE            43 non-null     object
 3   TOTAL_CASES      43 non-null     int64 
 4   TODAYS_CASES     43 non-null     int64 
 5   ACTIVE_CASES     43 non-null     int64 
 6   RECOVERED_CASES  43 non-null     int64 
 7   DECRESED_CASES   43 non-null     int64 
dtypes: int64(5), object(3)
memory usage: 2.2+ KB
```

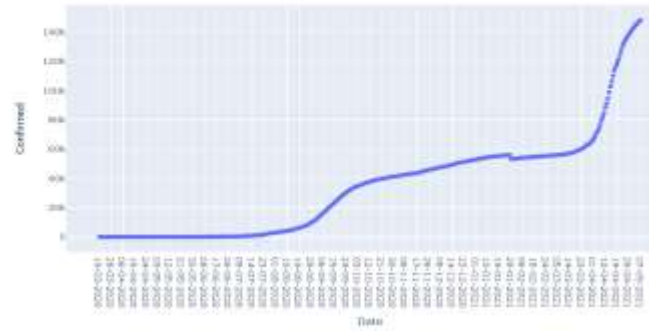
Describe dataset

```
data.describe()
```

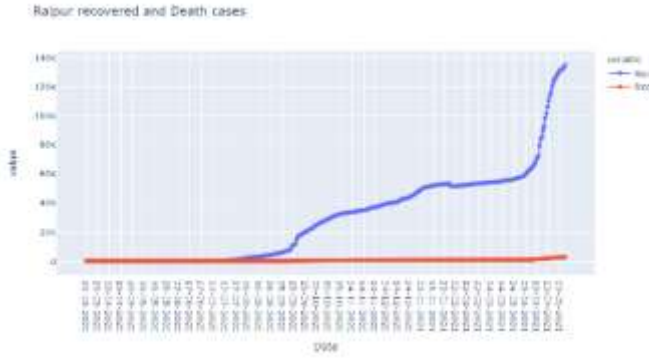
	TOTAL_CASES	TODAYS_CASES	ACTIVE_CASES	RECOVERED_CASES	DECRESED_CASES
count	43.000000	43.000000	43.000000	43.000000	43.0
mean	4.883721	0.953488	5.744186	0.581395	0.0
std	5.447626	1.951233	4.088853	1.621765	0.0
min	1.000000	0.000000	1.000000	0.000000	0.0
25%	3.000000	0.000000	3.000000	0.000000	0.0
50%	4.000000	0.000000	4.000000	0.000000	0.0
75%	4.000000	1.000000	7.000000	0.000000	0.0
max	29.000000	6.000000	16.000000	5.000000	0.0

3) *Visualize the data*
Total confirmed cases in Raipur, IN. with normal graph (from 19/03/2020 to 07/05/2021)

```
source = pd.read_csv('all_data.csv')
Raipur = source[source.District == 'Raipur']
fig = px.scatter(Raipur, x = 'Date', y = 'Confirmed')
fig.show()
```

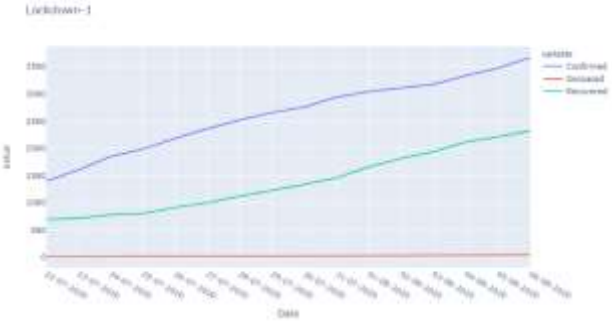


Total recovered and decreased cases in Raipur, IN.

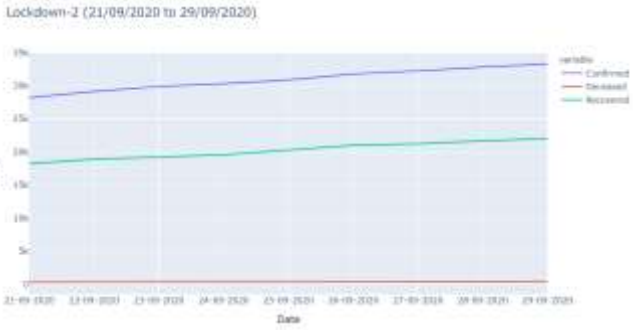


Comparison between total, recovered and decreased cases.

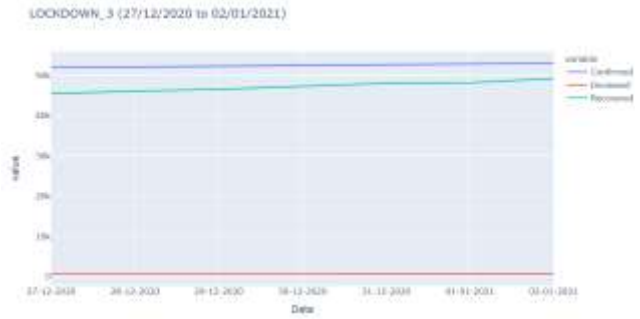
Lockdown period followed by, section 144, compulsory face-mask and maintain social distancing,



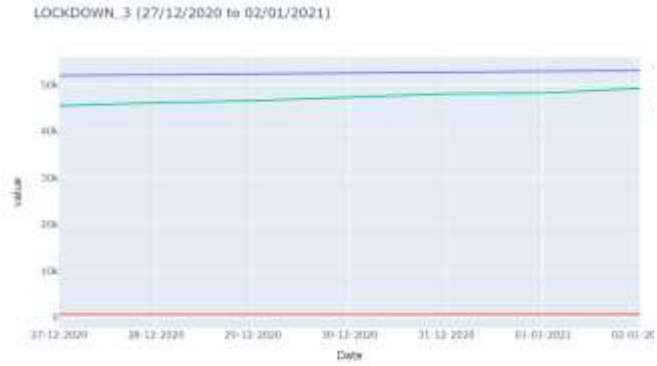
```
Fatality rate during Lockdown-1 = 1.0 %
Recovered rate during Lockdown-1 = 63.0 %
Active rate during Lockdown-1 = 36.0 %
```




```
Fatality rate during Lockdown-2 = 1.3 %
Recovered rate during Lockdown-2 = 66.0 %
Active rate during Lockdown-2 = 33.0 %
```



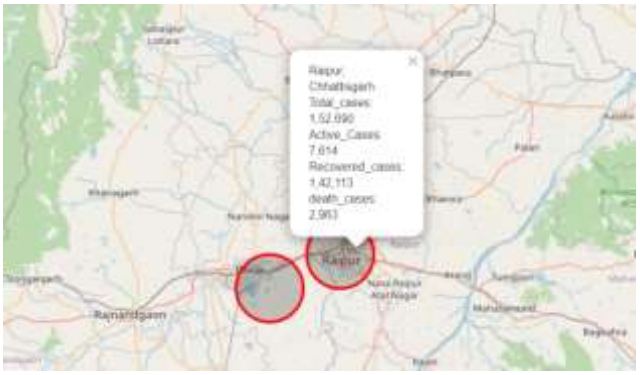
Fatality rate during Lockdown-2 = 1.3 %
Recovered rate during Lockdown-2 = 66.0 %
Active rate during Lockdown-2 = 33.0 %



Fatality rate during Lockdown-3 = 1.4 %
Recovered rate during Lockdown-3 = 93.0 %
Active rate during Lockdown-3 = 5.8 %

Fatality rate during Lockdown-4 = 1.8 %
Recovered rate during Lockdown-4 = 91.0 %
Active rate during Lockdown-4 = 7.1 %

Formula used for calculation,
Fatality rate = (Decreased cases/ confirmed cases) x 100
Recovery rate = (Recovered cases/confirmed cases) x 100
Active rate = [confirmed - (Recovered cases+ decreased cases)] x 100

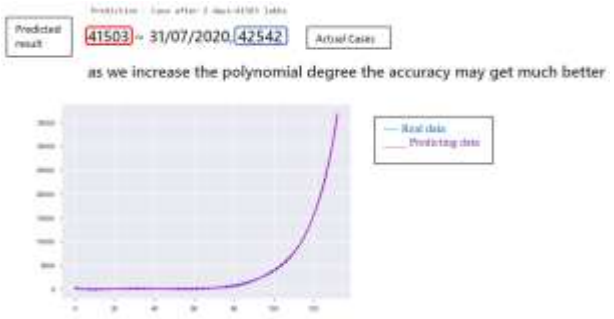


Above figure shows, Cases Visualisation on geographic-map**.**

4) **Dashboard** for visual representation of confirmed, recovered and death cases for all around the world.

VII. PREDICTIVE MODEL

A. Prediction of Covid19 cases for '*n*' numbers of days using polynomial regression.



VIII. CONCLUSION

Using the COVID-19 real-time data we report that the count of infected individuals, in Raipur, Chhattisgarh, India. And exhibit flattening of the curve. We conjecture that a rapid increase in cases may be due to the epidemic transmission by asymptomatic carriers traveling long distances, and due to community spread.

The prediction model, which will predict the cases for n number of days, here I had considered 2 days for case forecast, which is having an accuracy of 99.94%. where the data is been considered from the first day of the case in Raipur to the next 133 days and predicting the 135th day.

From data visualization and calculation, we can understand that the fatality rate of Raipur is very adequate, in all the seasons but there is an uncertain peak in few weeks. As there is no proper medical treatment available and vaccination is just started. Maintaining the social distancing, wearing masks and using hand sanitizer is preferable to decrease the count of cases and followed by a lockdown period in order for human welfare.

Thus, the COVID-19 epidemic data contains valuable insights which will help in forecasting the epidemic spread.

ACKNOWLEDGMENT

First and foremost, we would like to thank our college SSIPMT for constant support and our Head of Institute Dr. Alok Kumar Jain, his indirect support is also a blessing.

we would also like to extend my gratitude towards, family, friends, acquaintances and all other people who directly or indirectly aided the efforts during the completion of this project. Any help of any kind is greatly appreciated, without them, this would have been a rather difficult endeavour.

REFERENCES

[1] World Health Organization, Coronavirus disease (COVID-2019) situation reports; https://www.who.int/emergencies/diseases/novel-coronavirus2019/situation-reports/

[2]https://www.hindustantimes.com/indianews/coronavirus-first-case-of-covid-19-infection-reported-in - Chhattisgarh/story- TZX1YjdbyeuUHi3OQqfL.html

[3] R. Li, S. Pei, B. Chen, Y. Song, T. Zhang, W. Yang, and J. Shaman, Science 6, eabb3221 (2020)

[4] W. O. Kermack and A. G. McKendrick, Proceedings of the Royal Society A 115, 700 (1927).

[5] O. N. Bjørnstad, Epidemics: Models and Data using R (Springer, 2018).

[6] D. J. Daley and J. Gani, Epidemic Modelling: An Introduction (Cambridge University Press, 2001

[7]https://indianexpress.com/article/india/chhattisgarh-orders-shutdown-after-first-coronavirus-case-6322967/

[8] WorldOMeter, URL https://www.worldometers.info/coronavirus/.

[7] L. Peng, W. Yang, D. Zhang, C. Zhuge, and L. Hong, arXiv.org (2020), 2002.06563v1.

[8] C. Wang, L. Liu, X. Hao, H. Guo, Q. Wang, J. Huang, N. He, H. Yu, X. Lin, A. Pan, et al., medrxiv.org (doi.org/10.1101/2020.03.03.20030593) (2020).

[9] J. Labadin and B. H. Hong, medrxiv.org (doi.org/10.1101/2020.02.07.20021188) (2020).

[10] E. Shim, A. Tariq, W. Choi, Y. Lee, and G. Chowell, International Journal of Infectious Diseases (preprint) (2020)

[11] A. J. Kucharski, T. W. Russell, C. Diamond, L. Yang, J. Edmunds, S. Funk, and R. M. Eggo, The Lancet Infectious Diseases (preprint) (2020).

[12] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. M. Hyman, P. Yan, and G. Chowell, Infectious Disease Modelling 5, 256 (2020).

[13] M. Chinazzi, J. T. Davis, M. Ajelli, C. Gioannini, M. Litvinova, S. Merler, A. Pastore y Piontti, K. Mu, L. Rossi, K. Sun, et al., Science (preprint) (2020).

[14] J. Hellewell, S. Abbott, A. Gimma, N. I. Bosse, C. I. Jarvis, T. W. Russell, J. D. Munday, A. J. Kucharski, and J. Edmunds, The Lancet Global Health 8, e488 (2020).

[15] S. Mandal, T. Bhatnagar, N. Arinaminpathy, A. Agarwal, A. Chowdhury, M. Murhekar, R. Gangakhedkar, and S. Sarkar, Indian Journal of Medical Research (preprint) (2020).
[16] Y. Min, X. Jin, Y. Ge, and J. Chang, PLoS ONE 8, e57100 (2013).
[17] S. Meyer and L. Held, The Annals of Applied Statistics 8, 1612 (2014).
[18] K. Wu, D. Darcet, Q. Wang, and D. Sornette, arXiv.org (2020), arXiv2003.05681.
[19] Coronavirus image https://unsplash.com/photos/w9KEokhajKw
[20] Dataset URL https://api.covid19india.org/documentation

Code: https://github.com/Arvind-tech98/Arvind_Solanki