Covid-19 Pandemic and Machine Learning

Utsav Kumar
IIIT Allahabad
ism2016001@iiita.ac.in

Raghvendra Kumar
IIIT Allahabad
iim2016004@iiita.ac.in

I. Abstract

This study report is about the usage of a machine learning method, namely Polynomial Regression that has been used here, to predict the progress and regress of COVID-19 pandemic.

II. Introduction

Viral pandemics are a serious threat. COVID-19 is not the first, and it won't be the last. But, like never before, we now are collecting and sharing what we learn about the virus on a huge scale. Hundreds of scientists, researchers, and doctors around the world are combining their efforts to collect data and develop solutions. For our part we have gathered our dataset (from reliable sources) on COVID-19 pandemic and then used polynomial regression upon it to do an analysis on the pandemic's progress and regress. Furthermore, there is also a comparison on how the disease passage is between India and the rest of the world.

III. Polynomial Regression

A. Definition and example

Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted by $\mathrm{E}(y|x)$.

The reason here, to use a polynomial regression was that upon a visual inspection of our variables by doing univariate and bivariate inspections of our data before we began our regression analysis, it showed a simple scatter plot that revealed a curvilinear relationship instead of a linear one, which to a great extend is true to real life examples as hardly any epidemic that has occurred had a linear growth rate(without any intervention to stop it).

The goal of regression analysis is to model the expected value of a dependent variable y in terms of the value of an independent variable (or vector of independent variables) x. In simple linear regression, the model

$y = \lambda_0 + \lambda_1 x + \epsilon,$

is used, where ϵ is an unobserved random error with mean zero conditioned on a scalar variable x. In this model, for each unit increase in the value of x, the conditional expectation of y increases by λ_1 units.

In many settings, such a linear relationship may not hold. For example, if we are modeling the yield of a chemical synthesis in terms of the temperature at which the synthesis takes place, we may find that the yield improves by increasing amounts for each unit increase in temperature. In this case, we might propose a quadratic model of the form

$y = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \epsilon.$

In this model, when the temperature is increased from x to x + 1 units, the expected yield changes by $\lambda_1 + \lambda_2(2x + 1)$. (This can be seen by replacing x in this equation with x+1 and subtracting the equation in x from the equation in x+1.) For infinitesimal changes in x, the effect on y is given by the total derivative with respect to x: $\lambda_1 + 2\lambda_2 x$. The fact that the change in yield depends on x is what makes the relationship between x and y nonlinear even though the model is linear in the parameters to be estimated.

In general, we can model the expected value of y as an *n*th degree polynomial, yielding the general polynomial regression model

$y = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3 + \cdots + \lambda_n x^n + \epsilon.$

Conveniently, these models are all linear from the point of view of estimation, since the regression function is linear in terms of the unknown parameters $\lambda_0, \lambda_1, ...$ Therefore, for least squares analysis, the computational and inferential problems of polynomial regression can be completely addressed using the techniques of multiple regression. This is done by treating x, x^2, ... as being distinct independent variables in a multiple regression model.

B. Matrix form and calculation of estimates

The polynomial regression model

$y_i = \lambda_0 + \lambda_1 x_i + \lambda_2 x_i^2 + \cdots + \lambda_m x_i^m + \epsilon_i \ (i = 1, 2, \ldots, n)$

can be expressed in matrix form in terms of a design matrix $\mathbf{X}$, a response vector $\vec{y}$, a parameter vector $\vec{\lambda}$, and a vector $\vec{\epsilon}$ of random errors. The i-th row of $\mathbf{X}$ and $\vec{y}$ will contain the x and y value for the i-th data sample. Then the model can be written as a system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}, \quad \text{which}$$

when using pure matrix notation is written as

$\vec{y} = \mathbf{X}\vec{\lambda} + \vec{\epsilon}$.

The vector of estimated polynomial regression coefficients (using ordinary least squares estimation) is

$\widehat{\vec{\lambda}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\,\mathbf{X}^{\mathsf{T}}\vec{y}$,

assuming m $<$ n which is required for the matrix to be invertible; then since $\mathbf{X}$ is a Vandermonde matrix, the invertibility condition is guaranteed to hold if all the x_i values are distinct. This is the unique least-squares solution.

C. *Adavntages of Polynomial Regression :*

1) Broad range of function can be fit under it.
2) Polynomial basically fits wide range of curvature.
3) Polynomial provides the best approximation of the relationship between dependent and independent variable.

D. *Disadvantages of Polynomial Regression :*

1) It is too sensitive to the outliers.
2) The presence of one or two outliers in the data can seriously affect the results of a nonlinear analysis.
3) In addition there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.

E. *Uses of Polynomial Regression :*

It is basically used for analysis of non-linear phenomenon such as:

- To study the growth rate of tissues.
- To study the rise of different diseases within any population.
- To study the distribution of carbon isotopes in lake sediments.
- To study about the generation of any synthesis.

IV. DATA ACQUISITION

A. *DATA-SET I*

This is COVID-19 World Dataset from COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University Which have three type of data.

- Time series data of global confirm cases.
- Time series data of global recovered cases.
- Time series data of global death cases.

In this dataset, we have world data (confirm case, recover case, death case) from date 22 January 2020 to 28 April 2020. We have 102 columns as Province/State, Country/Region, Latitude, Longitude and date from 22/01/2020 to 28/04/2020.

B. *DATA-SET II*

We have this taken India dataset from www.covid19india.org which contains time series of COVID-19 in india from 30/01/2020 to 30/04/2020. It has following features:

- Date.
- Daily Confirmed.
- Total Confirmed.
- Daily Recovered.
- Total Recovered.
- Daily Deceased
- Total Deceased

V. DATA PRE-PROCESSING

A. *DATA-SET I*

In this world data-set, data like state, country, latitude and longitude are not useful. We calculate total confirm case, total recover case and total death case by calculating cumulative sum from 22/01/2020 to 28/04/2020. We also calculate active case and closed case by following formula:

- Closed cases = Recovered Cases + Death Cases
- Active Cases = Confirmed Cases - Closed Cases.

We use, data of 22/01/2020 as day 0, 23/01/2020 as day 1 so on till 28/04/2020 as day 98 and plot graph accordingly. We do our analysis on COVID-19 world data and COVID-19 daily change world data.

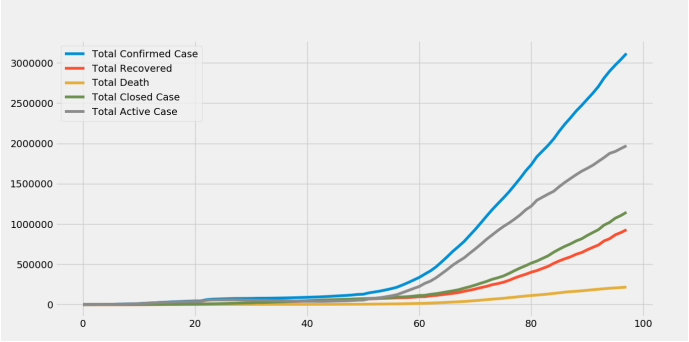

Fig. 1: Total World Data from 22/01/2020

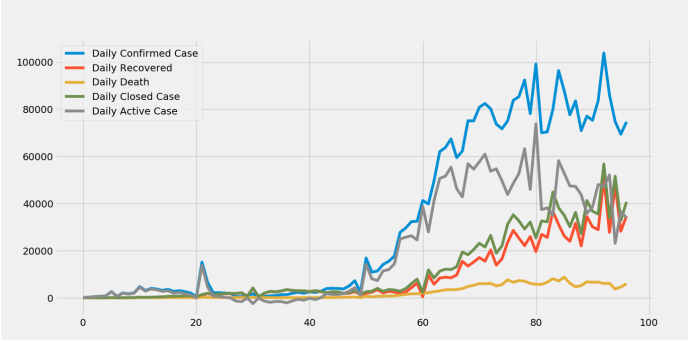

Fig. 2: Daily World Data change from 22/01/2020

We observe that, world data is exponentially increasing on daily basis and that the daily change data curves are not smooth and is not rapidly increasing. Using this we can't get data in which we can create a model to predict peak and the last or the end day of active cases. Thus, we firstly perform log transformation on the data and then plot the graph as follow:

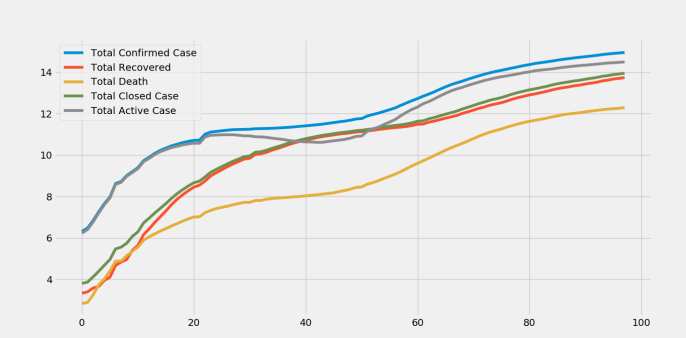

Fig. 3: Log transformation of Total World Data from 22/01/2020

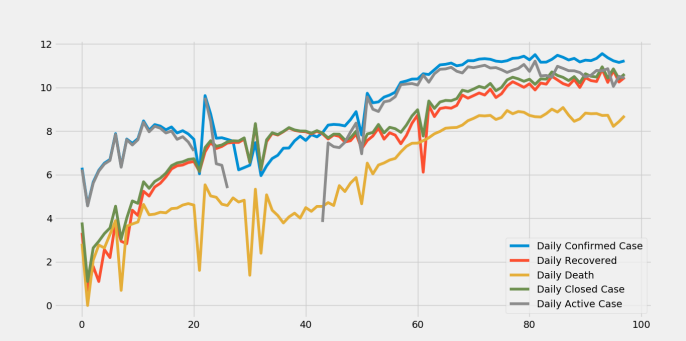

Fig. 4: Log transformation of daily World Data change from 22/01/2020

B. DATA-SET II

In the Indian dataset, We have confirmed cases, recovered cases, death cases and their daily change from 22/01/2020 to 28/04/2020 from which We calculate active case and closed case by following formula:

- Closed cases = Recovered Cases + Death Cases
- Active Cases = Confirmed Cases - Closed Cases.

We use the date 30/01/2020 as day 0, 31/01/2020 as day 1 so on till 30/04/2020 as day 92. Firstly, to visually analyze the data, we plot the graph as following.

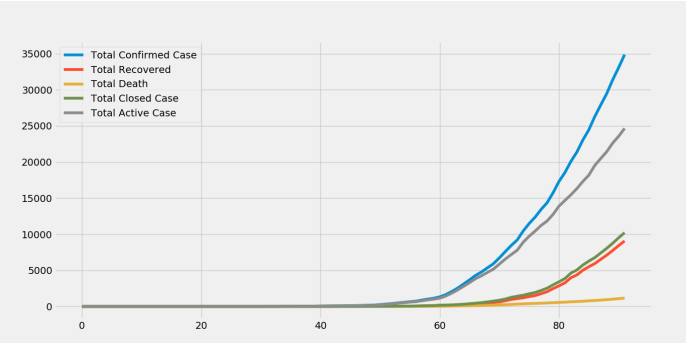

Fig. 5: Total Indian Data from 30/01/2020

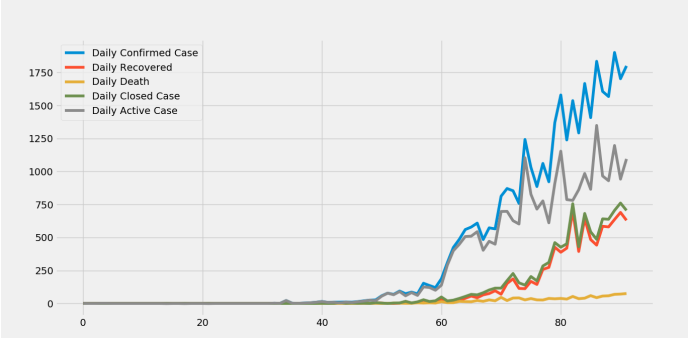

Fig. 6: Daily Indian Data change from 30/01/2020

Again we observe that, we have a similar problem in the Indian dataset as that of the world dataset. Thus, We perform log transformation and plot graphs for better analysis as follow:

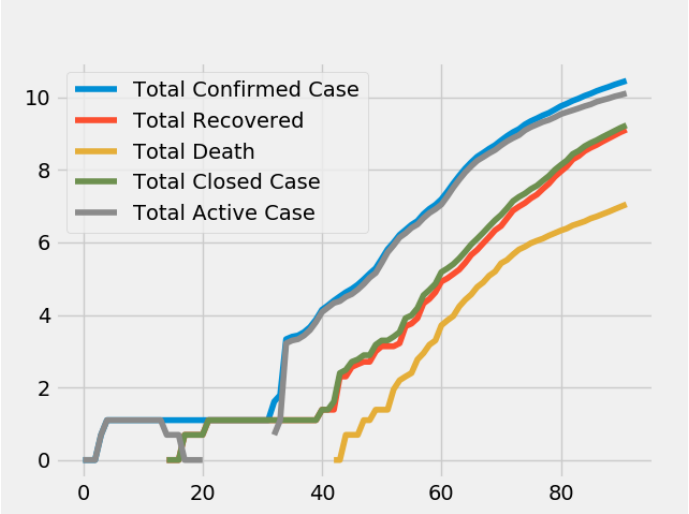

Fig. 7: Log transformation of Total Indian Data from 30/01/2020

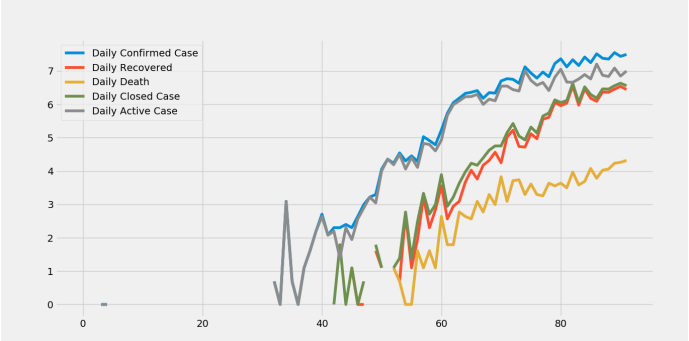

Fig. 8: Log transformation of daily Indian Data change from 30/01/2020

VI. EXPERIMENTAL RESULTS

A. DATA-SET I

To predict from the given dataset, we select the log transformation of daily change data because from this we can consider

acceleration factor of data. We create model on confirm cases and closed cases. After prediction we take the inverse log transformation of predicted data to change it to it's original form. We calculate cumulative sum to know total cases from daily cases. From both these model we calculate active cases. We use the most recent 1 week data as testing data and other remaining data as training data and for error check we use mean absolute error between log transformation of data and predicted data.

```
Confirm Case
Train
mean absolute error = 0.6421389125498826
Test
mean absolute error = 0.4904161491671697
Closed case
Train
mean absolute error = 0.45088981003783535
Test
mean absolute error = 1.7068062294078237
```

Fig. 9: Error computation on World Data

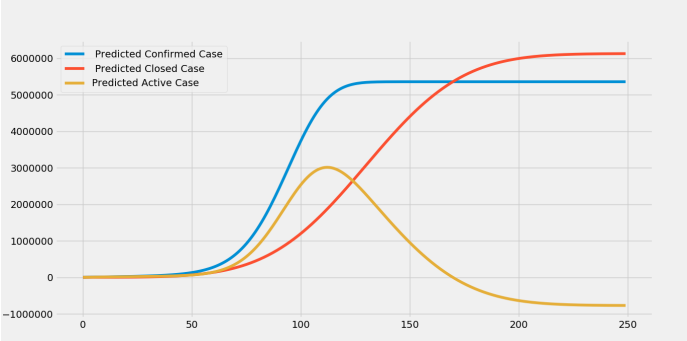

Fig. 10: predictions on world data from 22/01/2020

From the above graph, We can observe that world's last new case will emerge on 157th day, peak of active cases will be appearing on 112th day and active case last till 170th day. If we use DD/MM/YYYY format then our observation will be following:

- Last new case will be appearing on 26/06/2020.
- Peak of active cases will be appearing on 12/04/2020.
- Active case will last till 09/07/2020.
- No. of active cases at its peak will be 3013818.
- Total confirmed cases across the world will be 5358273.

B. DATA-SET II

Similar to previous model of dataset-1, we also use log transformation of daily change data of Indian data to create model. Here, we also create model for confirmed cases and closed cased. In graph of log transformation of daily change data of India there are many value that are missing, so we remove those points. After prediction we take inverse log transformation of predicted data to change it to it's original form. We calculate cumulative sum to know total cases from daily cases. From both these model we calculate active cases. We then use most recent 1 week data as testing data and remaining other as training data. For error check we use mean absolute error between log transformation of data and predicted data.

```
Confirmed Case
Train
mean absolute error = 0.2780968636438859
Test
mean absolute error = 4.3270401852039635
Closed case
Train
mean absolute error = 0.28167966374253095
Test
mean absolute error = 6.3838066044263595
```

Fig. 11: Error computation on Indian Data

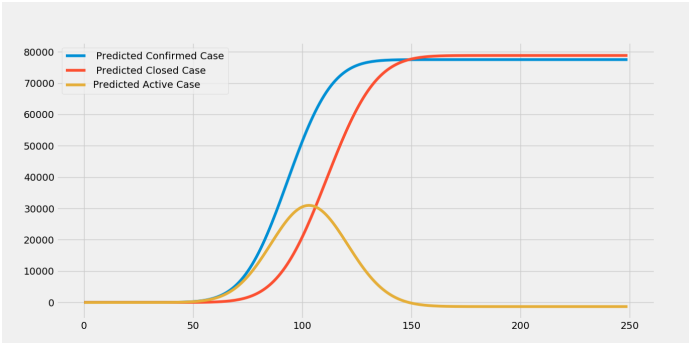

Fig. 12: predictions on Indian data from 30/01/2020

From above graph, We can observe that India's last new case will emerge on 143th day, pick of active case will take place on 103th and active case last till 150th day. After using DD/MM/YYYY format our observation will be following:

- Last new case will be appearing on 20/06/2020 .
- Peak of active cases will be appearing on 11/04/2020.
- Active cases will last till 27/06/2020.
- No. active cases at its peak will be 30987.
- Total confirmed cases across India will be 77520.

VII. CONCLUSION

The aim of this project was to apply a machine learning method to make a model on the COVID-19 dataset and to predict about the progress and regress of this pandemic that has caused worldwide havoc. Also, from our observations, we can also see that in comparison to rest of world, this pandemic is not so far spread in India. Lastly, we can conclude by saying

that machine learning is going to be a very important tool for us to fight off this ongoing pandemic and it would be very helpful in saving lives of common masses, both, for now and in the future.

VIII. References

1) Geeks for geeks: Implementation of Polynomial Regression
2) Wikipedia: Polynomial Regression
3) Kaggle : Corona World dataset
4) Covid19india: Corona India Dataset
5) towardsdatascience: Kernel function