SEMINAR REPORT

ON

ANALYSIS AND PREDICTIONS OF SPREAD, RECOVERY AND DEATH CAUSED BY COVID-19

BY

MS. HARSHADA SHRIKANT MORJE

MC21955

GUIDE – DR. GANESH MAGAR

POST GRADUATE DEPARTMENT OF COMPUTER SCIENCE,
S.N.D.T. WOMEN'S UNIVERSITY

MASTER OF COMPUTER APPLICATION
SEM VI - (2020 – 2021)

CERTIFICATE

This is to certify that MS. HARSHADA SHRIKANT MORJE has completed the seminar on ANALYSIS AND PREDICTIONS OF SPREAD, RECOVERY AND DEATH CAUSED BY COVID-19 satisfactorily as a partial fulfillment of the Post Graduate Degree of Master Of Computer Application (MCA).

Internal Examiner :	**External Examiner :**
Name :	Name :
Date :	Date :

Head of Department :

Name :

Date :

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my mentor DR. GANESH MAGAR for his valuable guidance and support in completing my research paper.

I would also like to express my gratitude towards him for allowing me to do a study on research paper. This study helped me learn many new things. Without the support and suggestions from my mentor, this study would not have been completed.

I take this opportunity to thank SNDT Woman`s University for giving me chance to do this project.

Harshada S. Morje.

Place:

Date:

ABSTRACT

The novel coronavirus outbreak was first reported in late December 2019 in Wuhan, China and more than 19 Crores people were infected with this disease and over 4.2 million worldwide lost their lives. The first case in India was reported on 30 January 2020 in Kerala state and has affected 3.16 Crores as of 30th July 2021 across the nation. More than 4.2 lakhs people has lost their lives.

This paper presents a detailed study of recently developed forecasting models and predicts the number of confirmed, recovered, and death cases in India caused by COVID-19. The dataset is collected from an internet platform known as Kaggle.com. Exploitation the dataset, the Anaconda tool and Python programming language are used for data cleaning, analysis, and representation. The machine learning algorithm used in this project is linear regression. The polynomial regression applied for prediction have been used to improve the accuracy. The finding suggests that lockdown and social distancing are two important factors that can help to suppress the increasing spread rate of COVID-19.

List of Figures and Tables

INDEX

CHAPTER 1 : INTRODUCTION

The coronavirus disease spreads through getting in touch with an infected person, touching a thing or object that has the virus on its surface and then touching their mouth, eyes, ears, or nose. The first case of COVID19 was detected in the last week of January 2020 in India and only 3 cases were diagnosed in next month. As of 8 August 2021, the total number of confirmed cases in India was 3.2 Crores with 3.1 crores recovered cases and 4.28L deaths. There is a need for the current situation to predict possible infected and death cases by using a computational model to arrange the necessary resources.

The virus of COVID-19 shows great resemblance with the Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS) coronavirus as investigated by pathologists. It is the seventh member of the coronavirus family that can spread among humans and easily transmit human-to-human through droplets of coughs or sneezing by an infected person. Major symptoms of COVID-19 are fever, cough, shortness of breath, and some patients show symptoms of diarrhea. The major problem in the case of this disease is that its symptoms generally appear after 2 to 14 days if an individual gets infected. This period is known as the incubation period and the mean incubation period is approximately 5 days. An infected person may infect the number of healthy persons during the incubation period. These patients are asymptomatic and major challenge is to identify them. The rate at which one infected person transmits this disease into others is termed as transmission rate (Ro). Recent studies by leading research organizations estimated that *Ro* is between 1.5 and 3.5[3-5]. This rate of transmission is very high in comparison to SARS (2.0) and common flu (1.3), and thus it is very dangerous. In the early stage, the Case Fatality Rate (CFR) for coronavirus was estimated at 2%. While the fatality rate for SARS and MERS was approximately 10% and 30% respectivelyAlmost 5% of the total infected persons lost their lives around the world, while in India CFR was near to 2.8%.

What is Covid ?

Syndrome (SARS) and some types of common cold.

How does the coronavirus spread?

As of now, researchers know that the coronavirus is spread through droplets and virus particles released into the air when an infected person breathes, talks, laughs, sings, coughs or sneezes. Larger droplets may fall to the ground in a few seconds, but tiny infectious particles can linger in the air and accumulate in indoor places, especially where many people are gathered and there is poor ventilation. This is why mask-wearing, hand hygiene and physical distancing are essential to preventing COVID-19.

What is the incubation period for COVID-19?

Symptoms show up in people within two to 14 days of exposure to the virus. A person infected with the coronavirus is contagious to others for up to two days before symptoms appear, and they remain contagious to others for 10 to 20 days, depending upon their immune system and the severity of their illness.

What are the symptoms of coronavirus?

Symptoms can include fever, cough and shortness of breath. In more severe cases, infection can cause pneumonia or breathing difficulties. More rarely, the disease can be fatal.

These symptoms are similar to the flu (influenza) or the common cold, which are a lot more common than COVID-19. This is why testing is required to confirm if someone has COVID-19. It’s important to remember that key prevention measures are the same – frequent hand washing, and respiratory hygiene (cover your cough or sneeze with a flexed elbow or tissue, then throw away the tissue into a closed bin).

How can I avoid the risk of infection?

Here are four precautions you and your family can take to avoid infection:

Wash your hands frequently using soap and water or an alcohol-based hand rub.

Fig 1.1

Cover mouth and nose with flexed elbow or tissue when coughing or sneezing. Dispose used tissue immediately

Fig 1.2

Avoid close contact with anyone who has cold or flu-like symptoms.

Fig 1.3

Seek medical care early if you or your child has a fever, cough or difficulty breathing.

Fig 1.4

<u>What should I do if as person has symptoms of COVID-19 or has been in contact with someone who has tested positive?</u>

- Seek medical attention. If advised by the doctor, then get your child tested for COVID-19. If advised by a doctor isolate/stay home.
- Record temperature and oxygen saturation with a pulse oximeter if available at home, every 6 hours. Measure their temperature frequently. If it is more than 100-degree F,

then you can do tepid sponging with tap water and give them syrup or tablet paracetamol. If fever is >100°F, give paracetamol 10–15 mg/kg/dose.

- Continue to follow good hand and respiratory hygiene practices like regular and washing with soap so that your child is protected against other viruses and bacteria causing diseases.

- Continue to follow personal protective measures for yourself and your child. Your child should wear a surgical mask anytime they are around people. Change the mask after eight hours of continuous wear. Caregivers interacting with the child should wear gloves and a mask.

- Feed your child home-cooked food and keep them well hydrated. Give plenty of liquids and give a soft and light diet. One may give vitamin C, zinc to boost overall health and immunity

- Other nutritional supplements like syrup multivitamin, Vitamin C, vitamin D, calcium can be given as per their doctor's advice.

<u>The major contributions of this research are as follows:</u>

• This paper performs a descriptive analysis of the COVID-19 outbreak in different states of India.

• We propose a model for predicting the number of confirmed, recovered, and death cases due to COVID-19.

• The proposed model deploys linear regression to predict the cases.

The detailed analysis is carried out from the considered dataset of COVID-19 for India. 4 introduces the proposed prediction model to estimate the count of infection, recovery, and death due to COVID-19. Section 5 concludes this study.

CHAPTER 2 : METHODOLOGY

2.1 System Architecture

There are few steps can be performed for gathering data, analysis and modelling to get best predictions.

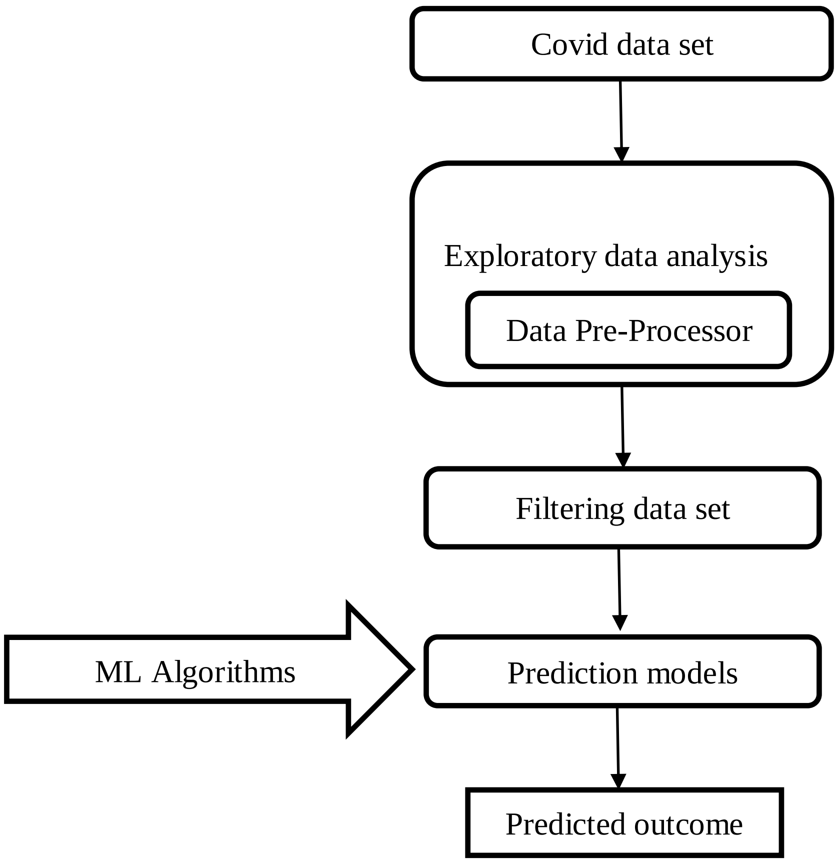


Fig.2.1. System architecture

Mining

- Descriptive Analytics will help an organization to know what has happened in the past, it would give you the past analytics using the data that are stored.
- For a company, it is necessary to know the past events that help them to make decisions based on the statistics using historical data.
- For example, you might want to know how much money you lost due to fraud and many more.

2.2 . Experiment

- This project/study uses a covid dataset of India. The Anaconda is used to run an experiment. Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
- Anaconda is a Python distribution focused on data driven projects.
- Anaconda is the best tool in processing a large amount of data for the required purpose

2.3 DATA PREPARATION AND ANALYSIS PROCEDURE:

1. Statistically exploring the data- a check list:
   - Check data dimensions
   - Rows, columns and column names
   - Data types and unique values per column.

2. Cleaning the data:
   - Look for any missing data.
   - Identify and convert categorical values to numerical representation or convert numerical to categorical representation using dummy variables if suitable for modelling and check for distinct values in categorical columns.

3. Statistically overview of data**:**
   - Check head, tails of data to see complete required data loaded. Understand the relationship of columns and how they are affecting each other.
   - Check the curves on graph and accuracy obtained

4. Graphical representation of data:
   - Perform visualization on dataset attributes.

2.4. METHODOLOGY

A. Machine Learning Algorithm

The ML algorithm is a logic that grasp one step ahead when exposed to more information/data. When ML is exposed to training data it produces model. To build a Model, the Machine Learning Algorithm used here Linear Regression (Supervised Learning). It predicts the output values based on the input data fed. This algorithm builds a model based on the training data produced and predicts the new data.

B. Dataset

The Anaconda is used to import the dataset. Dataset can be in excel or in CSV format. The dataset is reviewed and normalized. Normalization is changing the value of numeric columns of the dataset to common values and fit into a specific range.

This is a covid data including the confirmed, active, recovered and death cases of India. This data was downloaded from https://www.kaggle.com/sudalairajkumar/covid19-in-india

This dataset contains :

- Date
- State/UnionTerritory
- Cured
- Deaths
- Confirmed
- Active
- Total Doses
- Male
- Female
- Total Covaxin
- Total CoviShield
- Total_Sputnik

C. Analytical study of covid data across the country

We study the given data set of covid to find relations between different factors that affect the cases count. Our main objectives are :

- Finding which state has highest confirmed cases
- Finding which state has highest active cases
- Finding which state has highest cured/recovered cases
- Finding which state has highest death cases
- Finding the count differences between Top 3 states wrt to Highest cases
- Finding which regression is more applicable to this covid data

2.5 Data cleaning

At this stage, by using Anaconda we import dataset and remove redundant, missing, duplicate, and unnecessary data for further processing. This stage is the most time-consuming stage in Data Science because to prevent wrongful prediction and get rid of the inconsistencies of data.

Data loading processing

1. Loading the database
2. Check for appropriate columns
3. Check for null values
4. Check for title names
5. Generate different csv file

CHAPTER 3 : DATA EXPLORATION AND ANALYSIS

3.1 LINE PLOT

A line chart or line plot or line graph or curve chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. It is a basic type of chart common in many fields. It is similar to a scatter plot except that the measurement points are ordered (typically by their x-axis value) and joined with straight line segments. A line chart is often used to visualize a trend in data over intervals of time – a time series – thus the line is often drawn chronologically. In these cases they are known as run charts.

Syntax :

plotly.express.line(data_frame=None, x=None, y=None, line_group=None, color=None, line_dash=None, hover_name=None, hover_data=None, custom_data=None, text=None, facet_row=None, facet_col=None, facet_col_wrap=0, facet_row_spacing=None, facet_col_spacing=None, error_x=None, error_x_minus=None, error_y=None, error_y_minus=None, animation_frame=None, animation_group=None, category_orders=None, labels=None, orientation=None, color_discrete_sequence=None, color_discrete_map=None, line_dash_sequence=None, line_dash_map=None, log_x=False, log_y=False, range_x=None, range_y=None, line_shape=None, render_mode='auto', title=None, template=None, width=None, height=None)

In a 2D line plot, each row of data_frame is represented as vertex of a polyline mark in 2D space.

Parameters

- data_frame (DataFrame or array-like or dict) – This argument needs to be passed for column names (and not keyword names) to be used. Array-like and dict are tranformed internally to a pandas DataFrame. Optional: if missing, a DataFrame gets constructed under the hood using the other arguments.
- x (str or int or Series or array-like) – Either a name of a column in data_frame, or a pandas Series or array_like object. Values from this column or array_like are used to position marks along the x axis in cartesian coordinates. Either x or y can optionally

be a list of column references or array_likes, in which case the data will be treated as if it were ‘wide’ rather than ‘long’.

- y (str or int or Series or array-like) – Either a name of a column in data_frame, or a pandas Series or array_like object. Values from this column or array_like are used to position marks along the y axis in cartesian coordinates. Either x or y can optionally be a list of column references or array_likes, in which case the data will be treated as if it were ‘wide’ rather than ‘long’.
- line_group (str or int or Series or array-like) – Either a name of a column in data_frame, or a pandas Series or array_like object. Values from this column or array_like are used to group rows of data_frame into lines.
- color (str or int or Series or array-like) – Either a name of a column in data_frame, or a pandas Series or array_like object. Values from this column or array_like are used to assign color to marks.
- text (str or int or Series or array-like) – Either a name of a column in data_frame, or a pandas Series or array_like object. Values from this column or array_like appear in the figure as text labels.
- labels (dict with str keys and str values (default {})) – By default, column names are used in the figure for axis titles, legend entries and hovers. This parameter allows this to be overridden. The keys of this dict should correspond to column names, and the values should correspond to the desired label to be displayed.
- line_shape (str (default 'linear')) – One of 'linear' or 'spline'.
- title (str) – The figure title.
- template (str or dict or plotly.graph_objects.layout.Template instance) – The figure template name (must be a key in plotly.io.templates) or definition.
- width (int (default None)) – The figure width in pixels.
- height (int (default None)) – The figure height in pixels.

Code:

```
#Import the packages matplotlib first to get the proper plotting:
import matplotlib.pyplot as plt
#plotting graph
fig=px.line(overall,x='State/UnionTerritory',y=['Confirmed', 'Active', 'Cured', 'Deaths'], title='Covid cases')
fig.show()
```

3.2 BAR PLOT :

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

Syntax:

seaborn.barplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, e stimator=<function_mean_at0x7fecadf1cee0>, ci=95, n_boot=1000, units=None, seed=Non e, orient=None, color=None, palette=None, saturation=0.75, errcolor='.26', errwidth=None, capsize=None, dodge=True, ax=None)

Parameters :

- x, y, huenames of variables in data or vector data, optional
  Inputs for plotting long-form data. See examples for interpretation.
- dataDataFrame, array, or list of arrays, optional
  Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.
- estimatorcallable that maps vector -> scalar, optional
  Statistical function to estimate within each categorical bin.
- Orientation of the plot (vertical or horizontal). This is usually inferred based on the type of the input variables, but it can be used to resolve ambiguitiy when both x and y are numeric or when plotting wide-form data.
- colormatplotlib color, optional
  Color for all of the elements, or seed for a gradient palette.
- matplotlib colors.
  errcolormatplotlib color
  Color for the lines that represent the confidence interval.

3.3 KDE PLOT :

- A kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset, 21nalogous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions.

- The approach is explained further in the user guide.

- Relative to a histogram, KDE can produce a plot that is less cluttered and more interpretable, especially when drawing multiple distributions. But it has the potential to introduce distortions if the underlying distribution is bounded or not smooth. Like a histogram, the quality of the representation also depends on the selection of good smoothing parameters.

Syntax :

seaborn.kdeplot(x=None, *, y=None, shade=None, vertical=False, kernel=None, bw=None, gridsize=200, cut=3, clip=None, legend=True, cumulative=False, shade_lowest=None, cba r=False, cbar_ax=None, cbar_kws=None, ax=None, weights=None, hue=None, palette=No ne, hue_order=None, hue_norm=None, multiple='layer', common_norm=True, common_g rid=False, levels=10, thresh=0.05, bw_method='scott', bw_adjust=1, log_scale=None, colo r=None, fill=None)

Parameters :

x, yvectors or keys in data

Variables that specify positions on the x and y axes.

Shadebool

Alias for fill. Using fill is recommended.

Verticalbool

Orientation parameter.

Deprecated since version 0.11.0: specify orientation by assigning the x or y variables.

Kernelstr

Function that defines the kernel.

Deprecated since version 0.11.0: support for non-Gaussian kernels has been removed.

Bwstr, number, or callable

Smoothing parameter.

Deprecated since version 0.11.0: see bw_method and bw_adjust.

Gridsizeint

Number of points on each dimension of the evaluation grid.

Cbarbool

If True, add a colorbar to annotate the color mapping in a bivariate plot. Note: Does not currently support plots with a hue variable well.

Cbar_axmatplotlib.axes.Axes

Pre-existing axes for the colorbar.

Cbar_kwsdict

Additional parameters passed to **matplotlib.figure.Figure.colorbar()**.

Axmatplotlib.axes.Axes

Pre-existing axes for the plot. Otherwise, call **matplotlib.pyplot.gca()** internally.

Weightsvector or key in data

If provided, weight the kernel density estimation using these values.

Huevector or key in data

Semantic variable that is mapped to determine the color of plot elements.

Palettestring, list, dict, or matplotlib.colors.Colormap

Method for choosing the colors to use when mapping the hue semantic. String values are passed to **color_palette()**. List or dict values imply categorical mapping, while a colormap object implies numeric mapping.

Colormatplotlib color

Single color specification for when hue mapping is not used. Otherwise, the plot will try to hook into the matplotlib property cycle.

Fillbool or None

If True, fill in the area under univariate density curves or between bivariate contours. If None, the default depends on multiple.

Datapandas.DataFrame, numpy.ndarray, mapping, or sequence

Input data structure. Either a long-form collection of vectors that can be assigned to named variables or a wide-form dataset that will be internally reshaped.

CHAPTER 4 : DATA VISUALIZATION

1. Line Plotting

i. Plotting Confirmed, Active, Cured and Death cases of all States

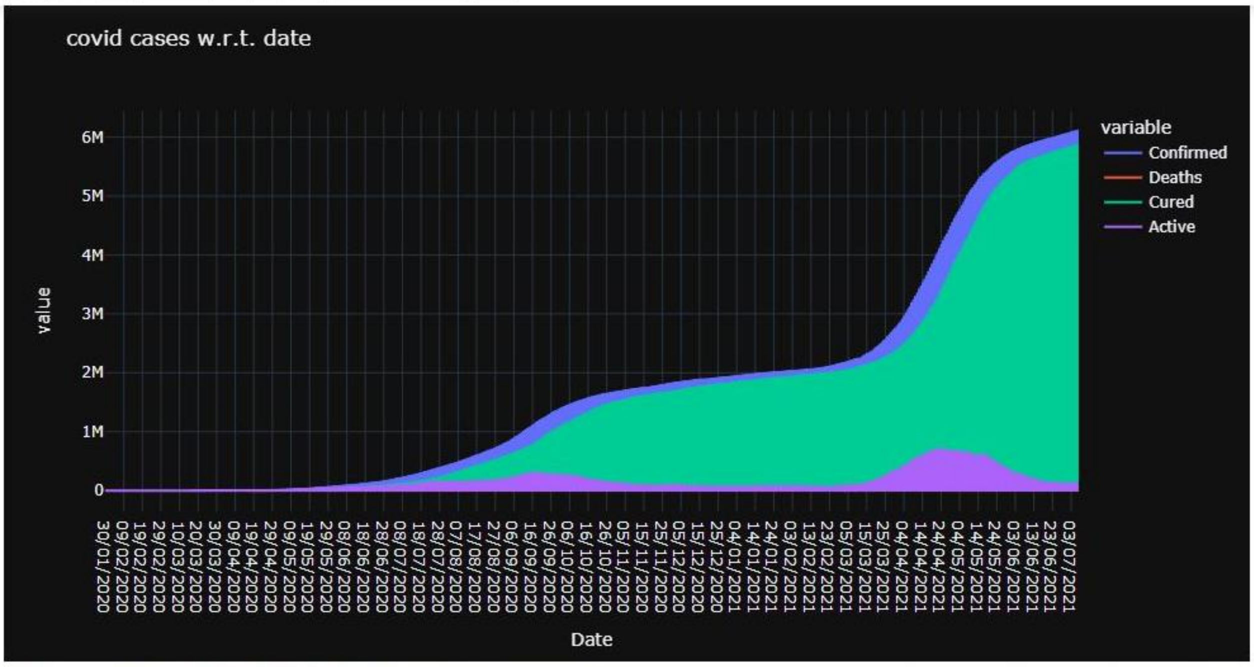


Fig 4.1

ii. Total doses administered wrt Gender

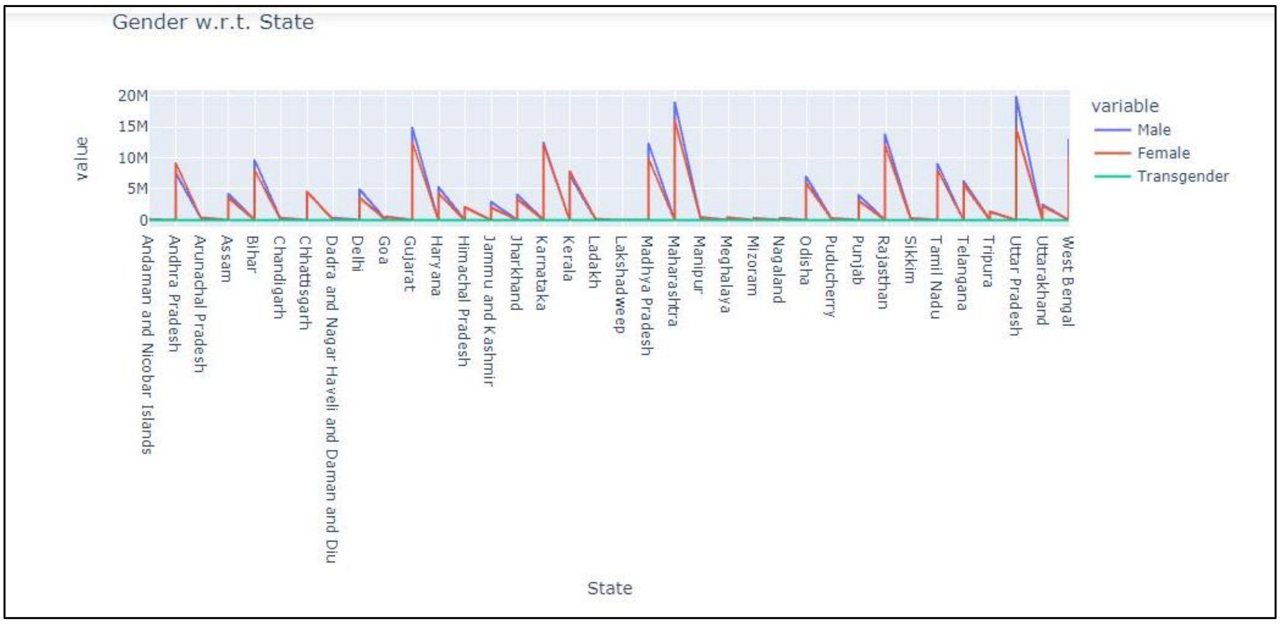


Fig 4.2

iii. Total doses administered wrt First/Second dose

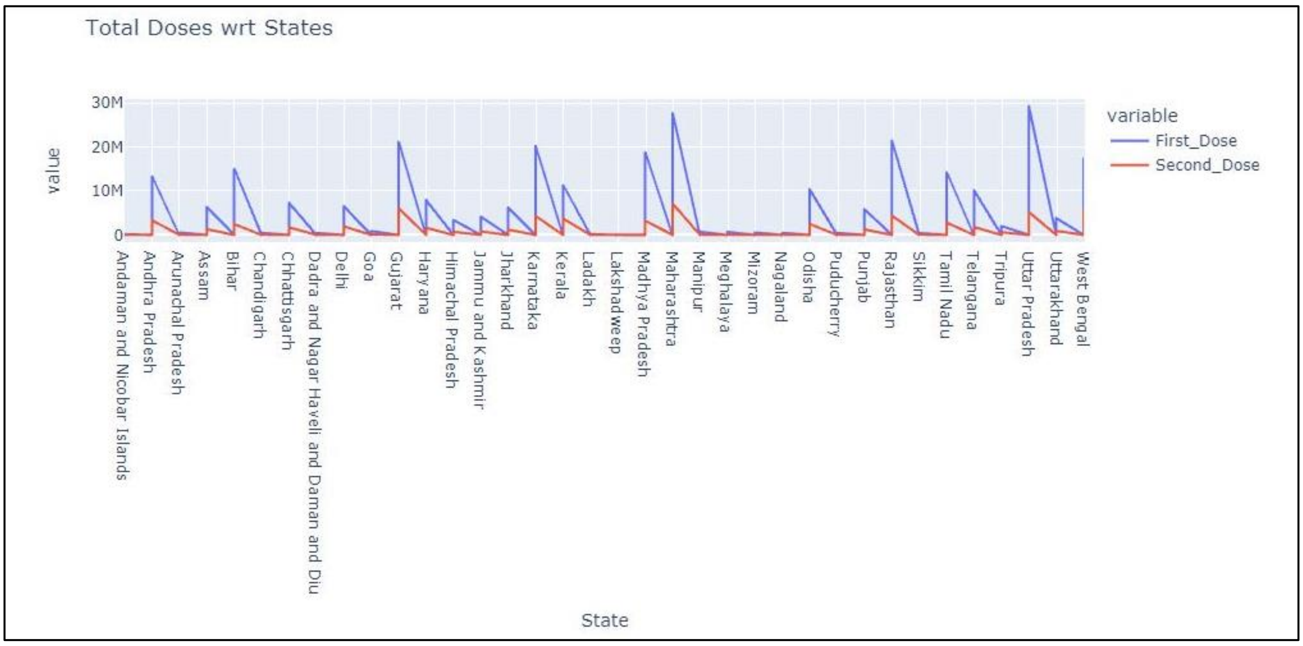


Fig 4.3

iv. Total doses administered wrt Types of doses

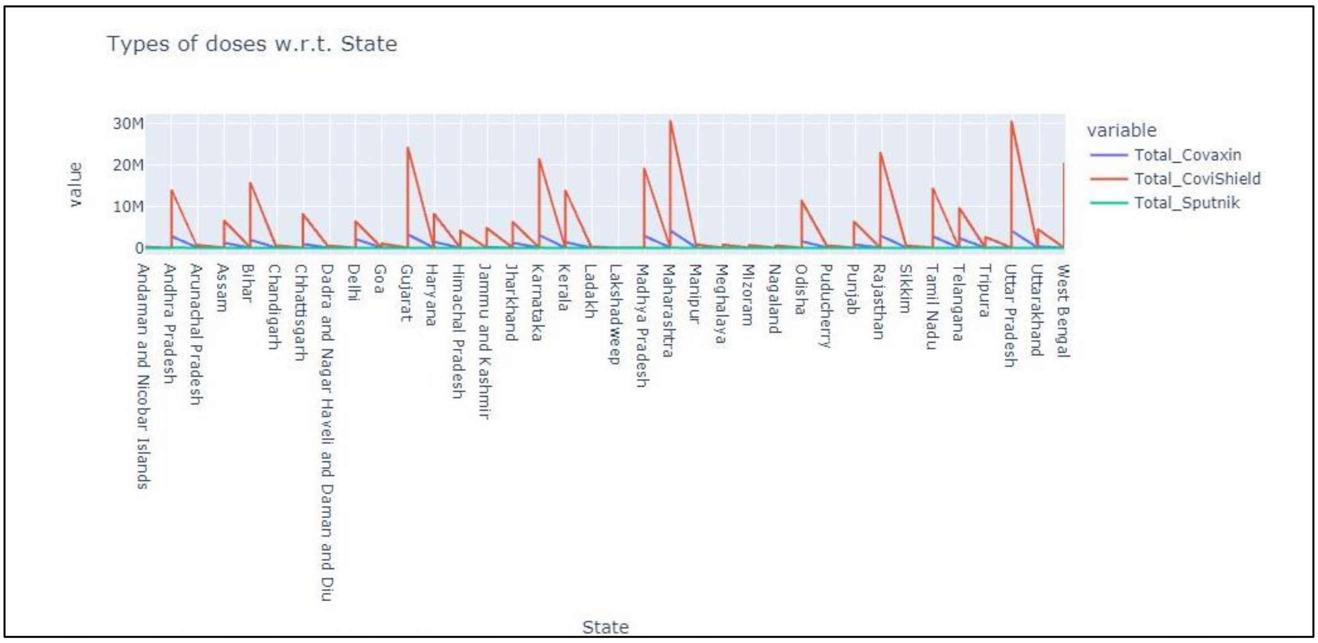


Fig 4.4

v. **Total doses administered wrt Age group**

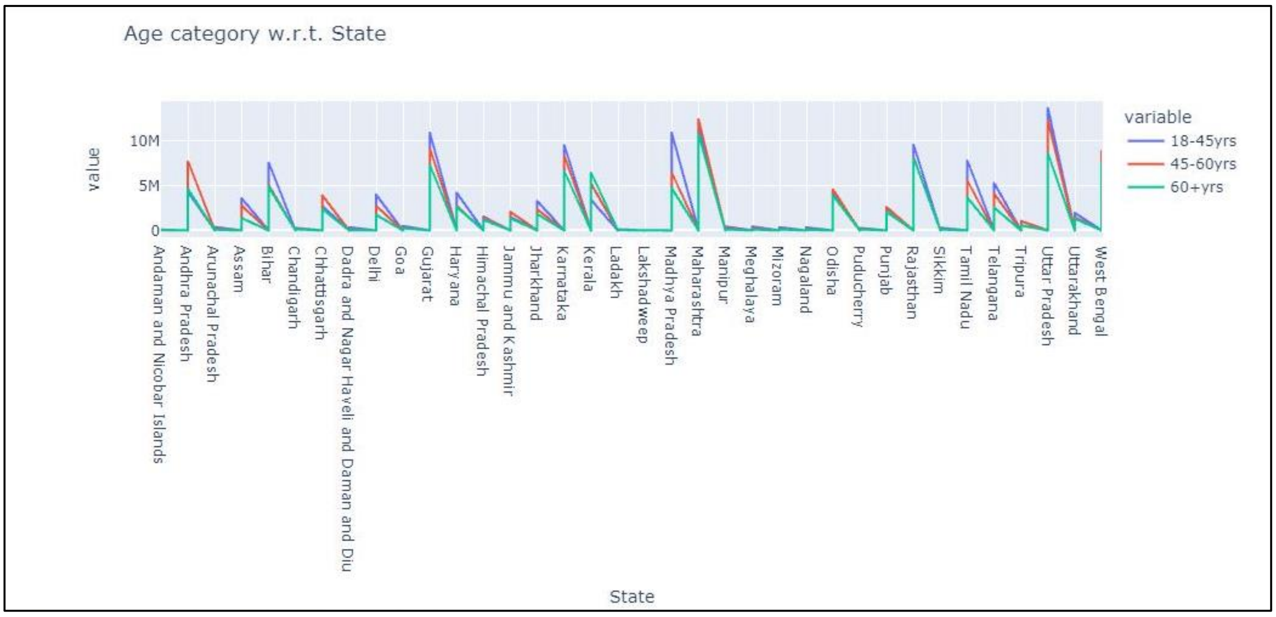


Fig 4.5

2. **Bar Plotting**

i. **Top 10 Highly impacted states – Confirmed**

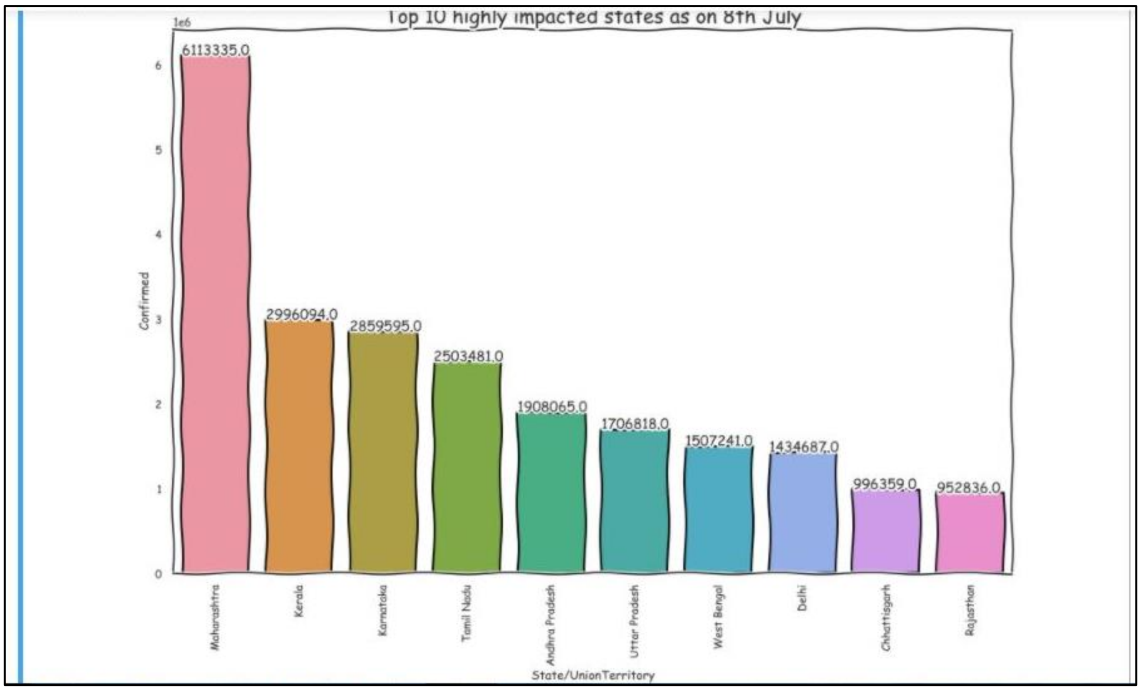


Fig 4.6

ii. Top 10 Highly impacted states – Active

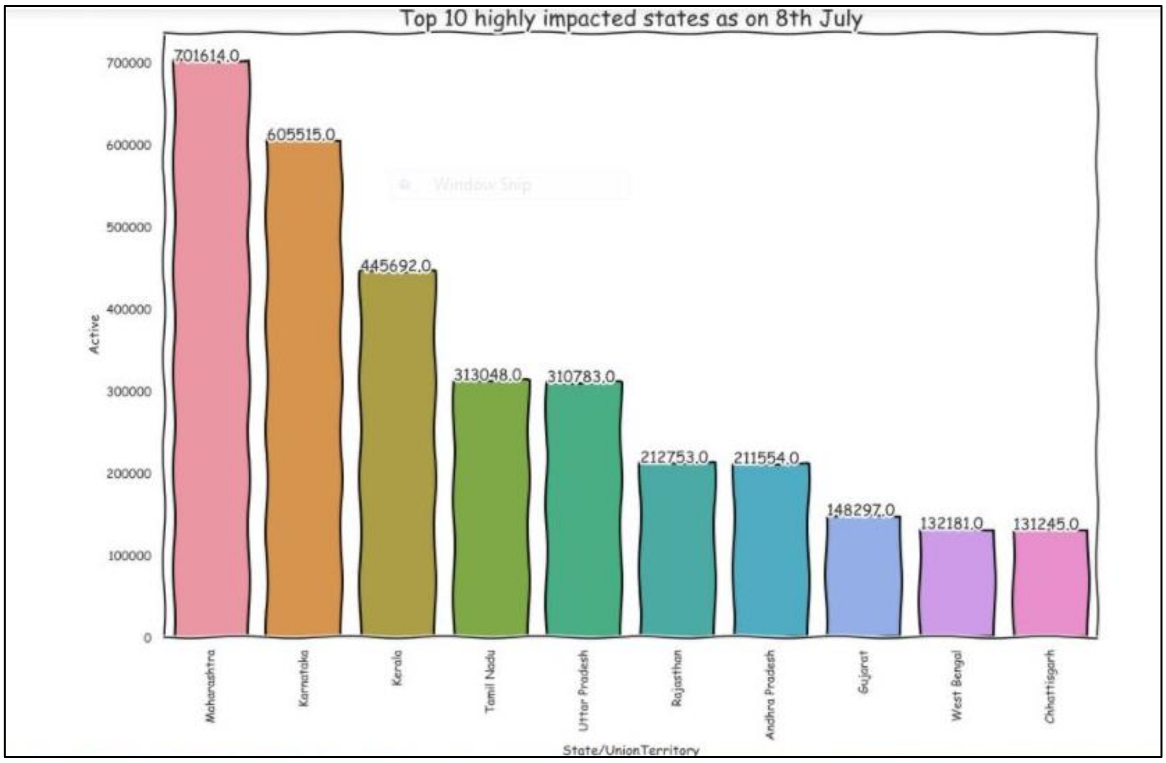


Fig 4.7

iii. Top 10 Highly impacted states - Cured

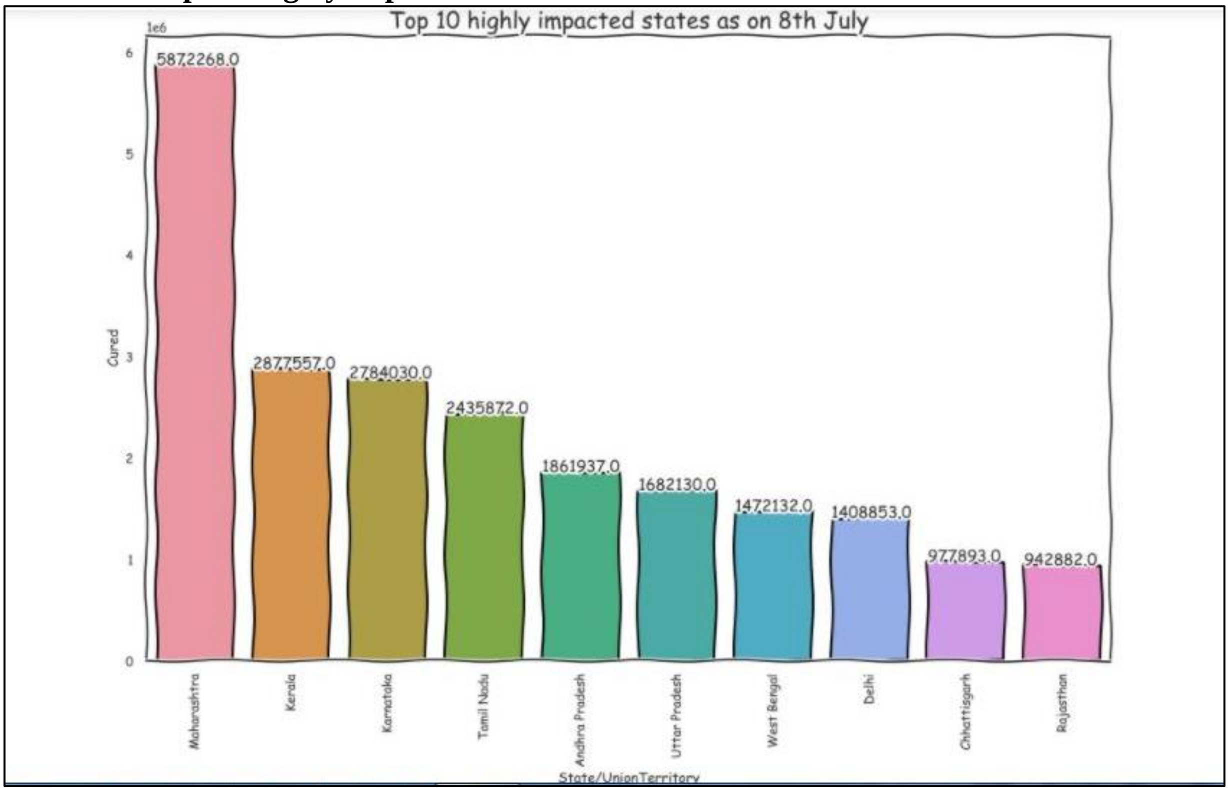


Fig 4.8

iv. Top 10 Highly impacted states - Death

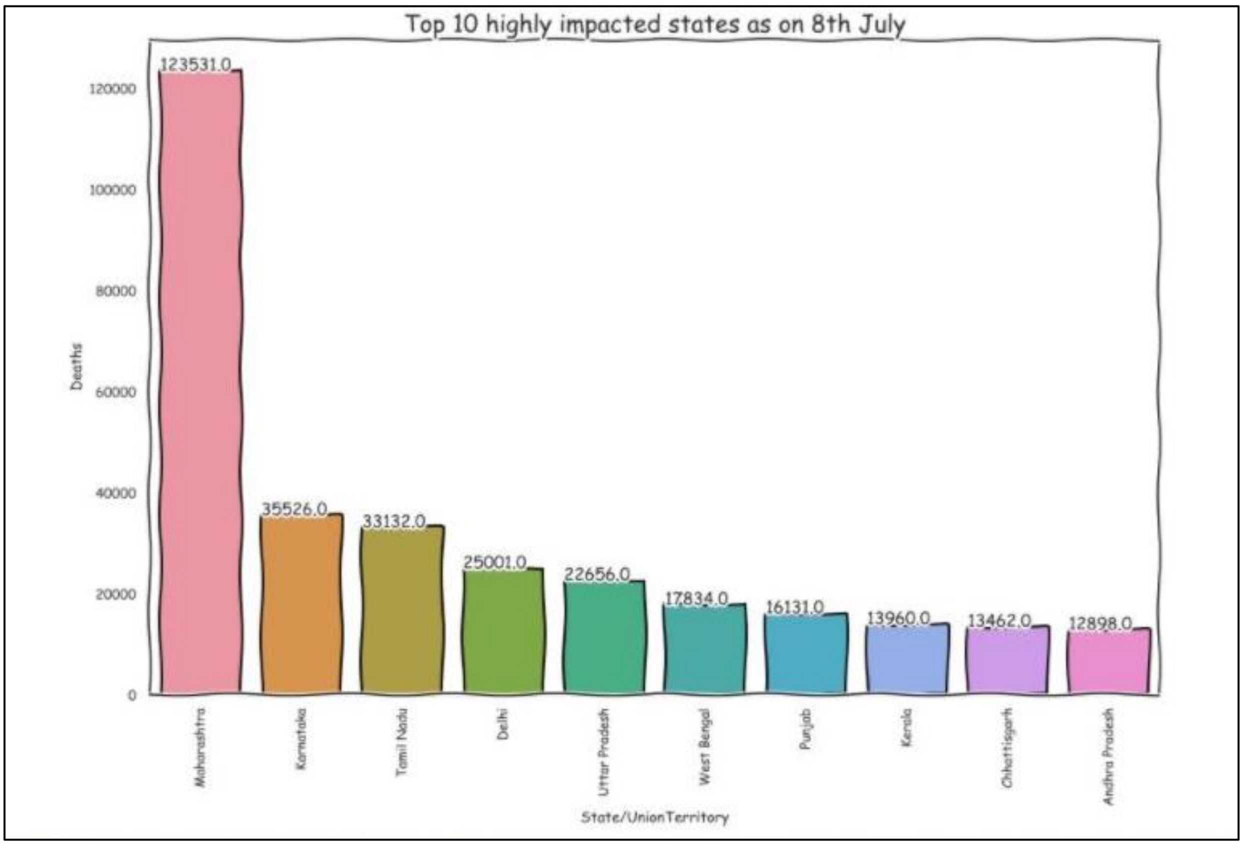


Fig 4.9

3. KDE Plotting

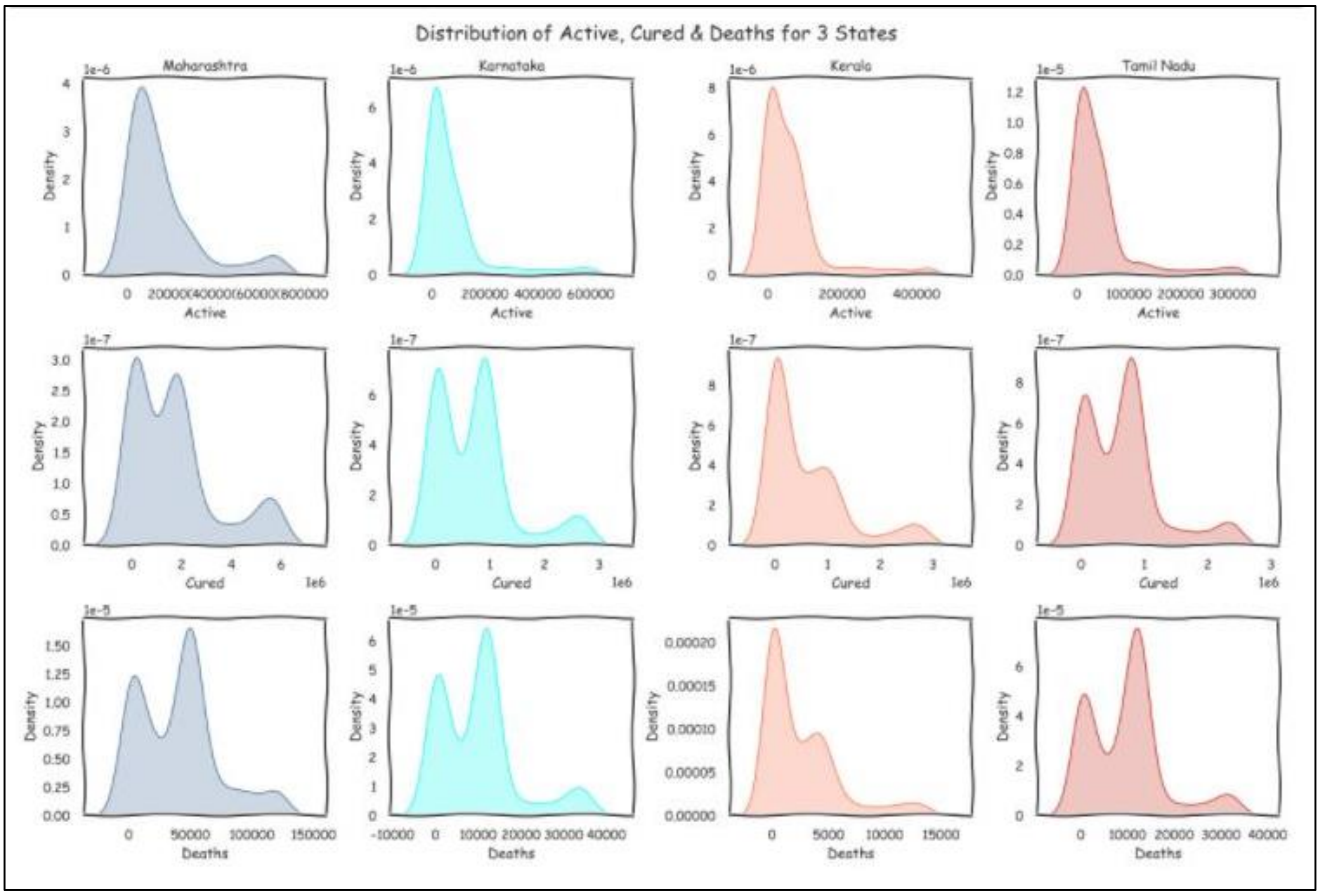


Fig 4.10

CHAPTER 5 : PREDICTIVE ANALYSIS

Support vector regression:

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

Support vector regression uses svm classification algorithm to forecast a continuous variable. But other regression models are used to minimize the error between predicted value and actual value. SVR tries to fit best line among the predefined error value. Svr have few important key words such as kernel, hyper plane, boundary line, support vector. Support vector regressions have two types .

Linear SVR :- $\sum (\alpha i - \alpha i *).(xi , x) + b \; N \; i=1$
Non linear SVR :- $\sum (\alpha i - \alpha i *).K(xi , x) + b \; N \; i=1$

Prediction : Support Vector Machines

Algorithms :

Step 1: Importing the libraries

Step 2: Reading the dataset

Step 3: Feature Scaling

Step 4: Fitting SVR to the dataset

Step 5. Checking accuracy

Step 6. Predicting a new result

Step 7. Visualizing the SVR results (for higher resolution and smoother curve)

Code:

Step 1: Importing the libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
```

Step 2: Reading the dataset

```
data = pd.read_csv("D:\SEM VI\Seminar ppt\Data\Cleaned\Overall_India.csv")
```

Step 3: Feature Scaling

```
sc_X = StandardScaler()
sc_Y = StandardScaler()
Sx = sc_X.fit_transform(x)
Sy = sc_Y.fit_transform(y)
```

Step 4: Fitting SVR to the dataset

```
reg = SVR(kernel='rbf')
reg.fit(Sx,Sy.ravel())
```

Step 5. Checking accuracy

```
reg.score(Sx,Sy)*100
```

Step 6. Visualizing the SVR results (for higher resolution and smoother curve)

```
plt.scatter(x,y)
plt.plot(x,reg.predict(x),color='k',linewidth=3)
plt.xlabel("Number of Days")
plt.ylabel("Confirmed cases of Maharashtra")
plt.title("Support Vector Regression",fontsize=15)
plt.show()
```

Output:

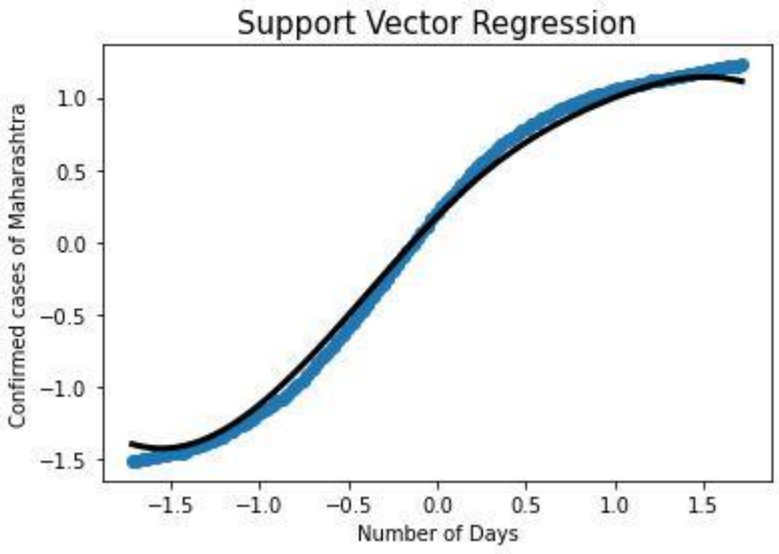


Fig.5.1. SVR - Confirmed cases of Maharashtra

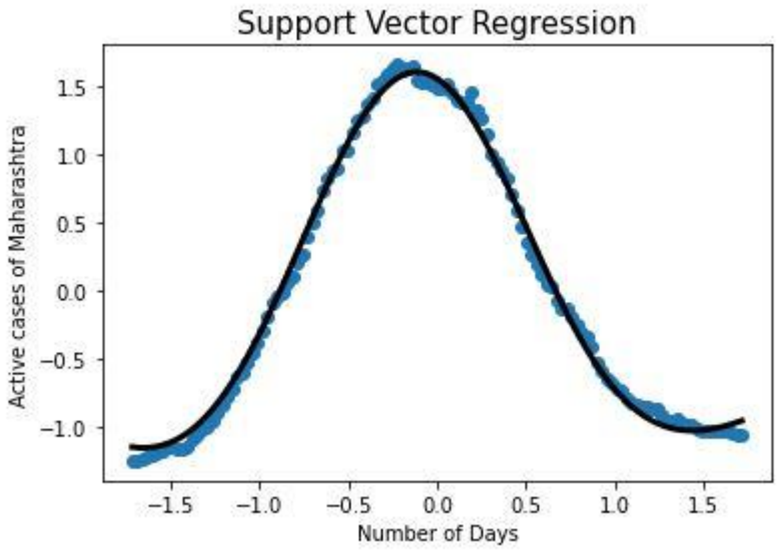


Fig.5.2. SVR - Active cases of Maharashtra

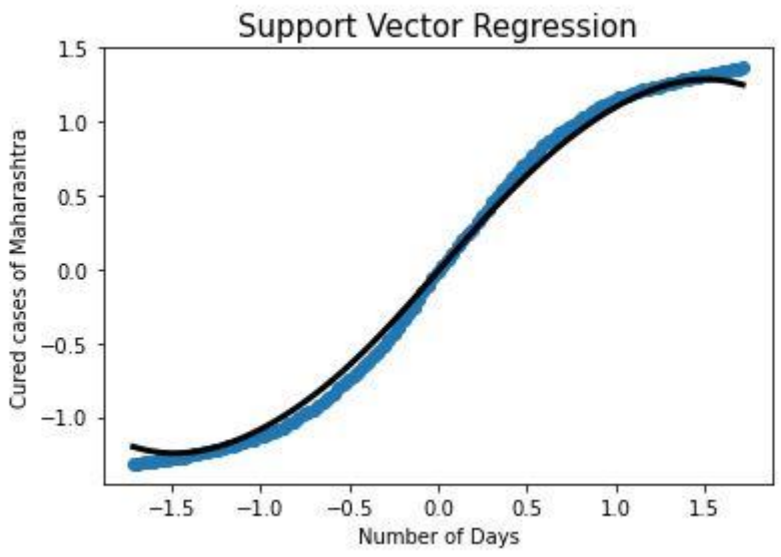


Fig.5.3. SVR - Cured of Maharashtra

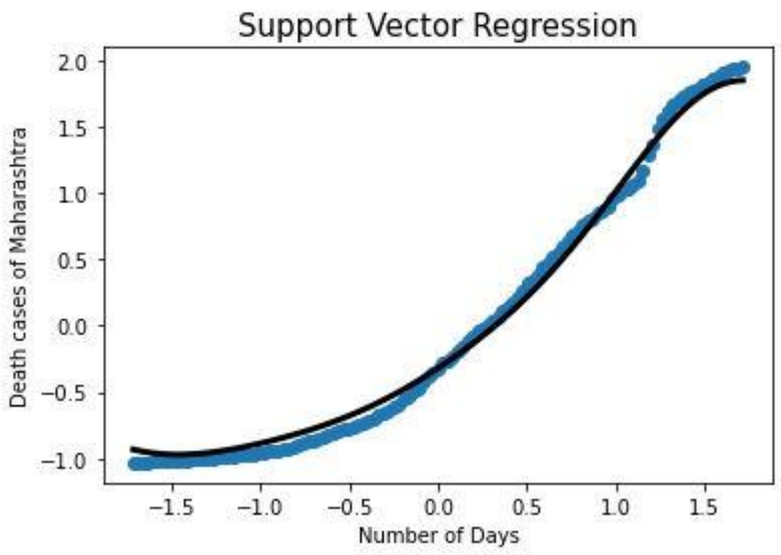


Fig.5.4. SVR - Death cases of Maharashtra

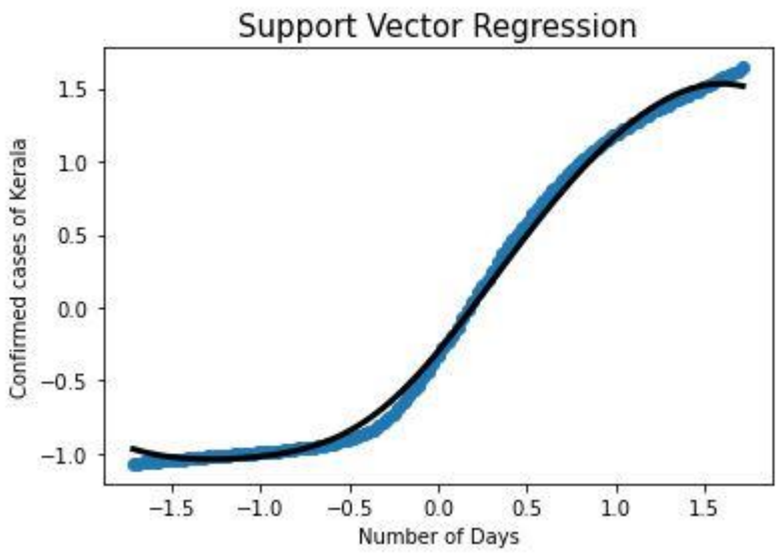


Fig.5.5. SVR - Confirmed cases of Kerala

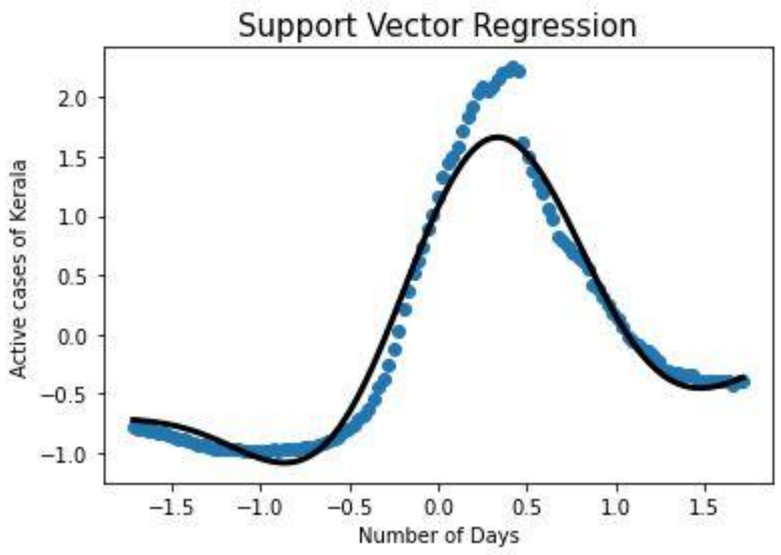


Fig.5.6. SVR - Active cases of Kerala

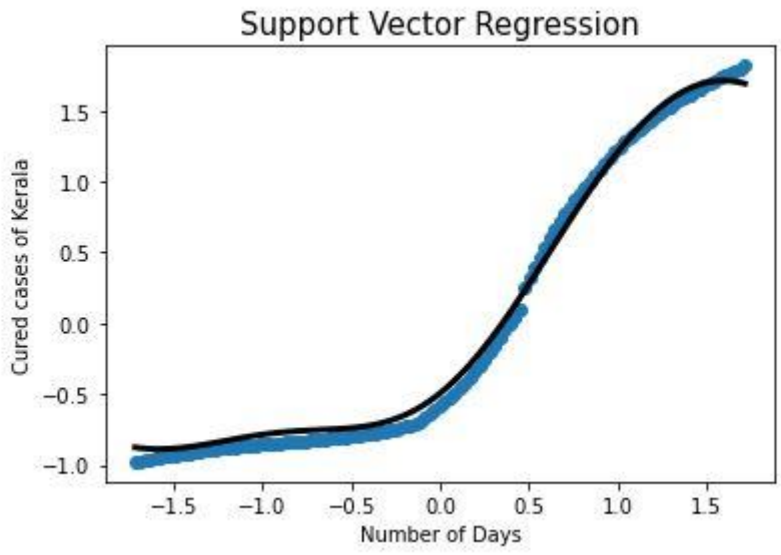


Fig.5.7. SVR - Cured of Kerala

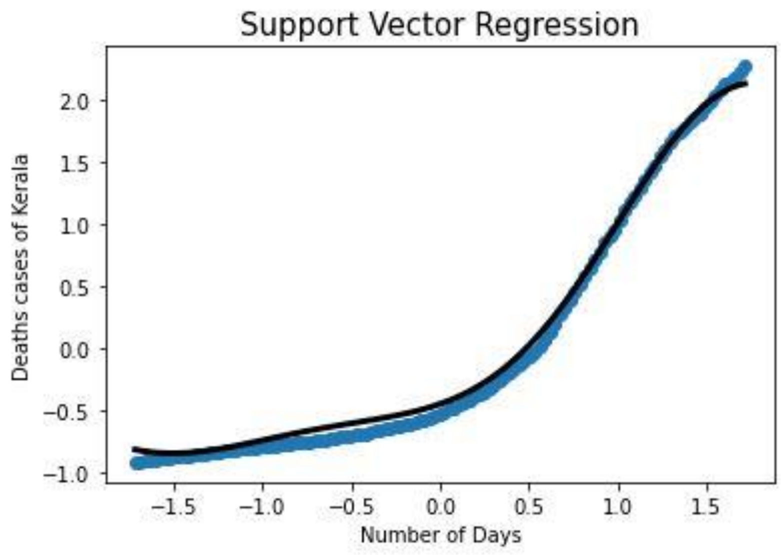


Fig.5.8. SVR - Death cases of Kerala

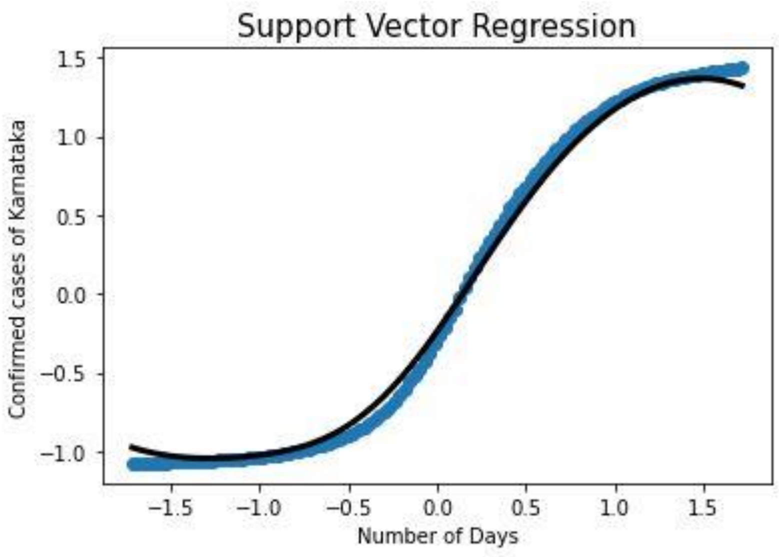


Fig.5.9. SVR - Confirmed cases of Karnataka

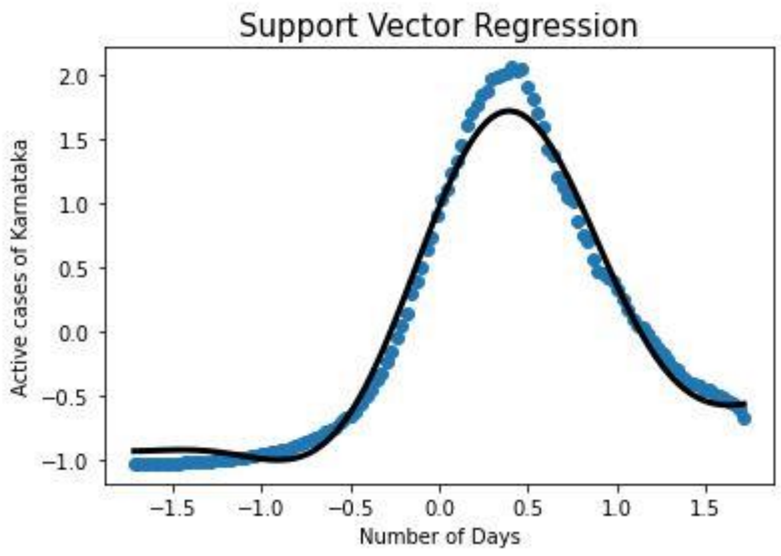


Fig.5.10. SVR - Active cases of Karnataka

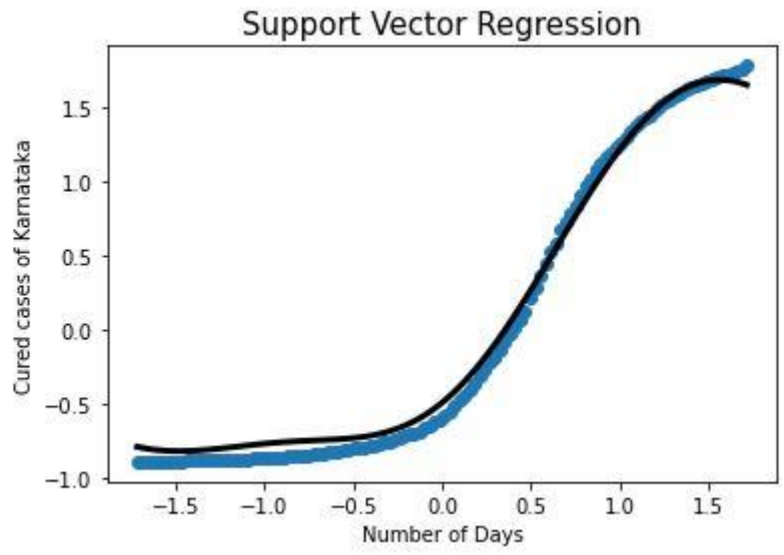


Fig.5.11. SVR - Cured of Karnataka

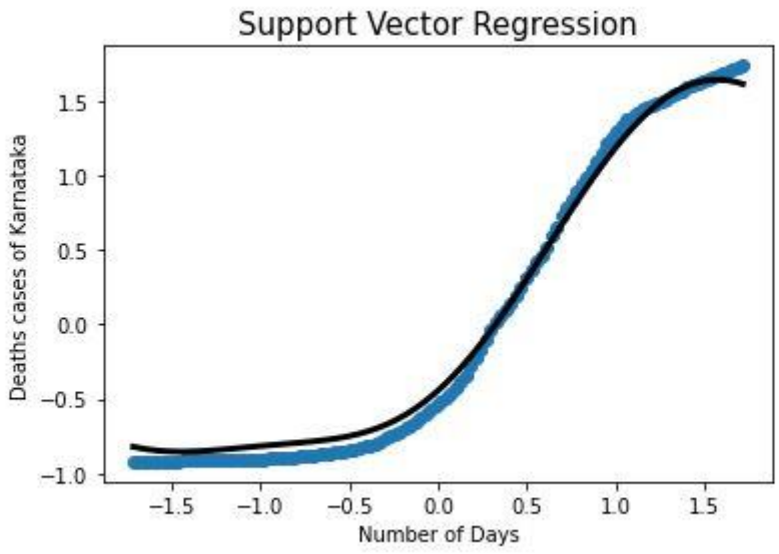


Fig.5.12. SVR - Death cases of Karnataka

Polynomial Regression :

Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.

In a curvilinear relationship, the value of the target variable changes in a non-uniform manner with respect to the predictor (s).

This linear equation can be used to represent a linear relationship. But, in polynomial regression, we have a polynomial equation of degree *n* represented as:

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

Here, The number of higher-order terms increases with the increasing value of *n*, and hence the equation becomes more complicated.

Advantages of using Polynomial Regression:

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

Disadvantages of using Polynomial Regression

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

Algorithm :

Step 1: Import libraries and dataset

Step 2: Dividing the dataset into 2 components

Step 3: Fitting Linear Regression to the dataset

Step 4: Fitting Polynomial Regression to the dataset

Step 5: Visualising the Polynomial Regression results using scatter plot.

Step 6: Predicting new result with Polynomial Regression.

Code :

Step 1: Import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
```

Step 2: Import dataset and divide the dataset into 2 components

```
data = pd.read_csv("D:\SEM VI\Seminar ppt\Data\Cleaned\Overall_India.csv")
x = np.arange(len(Day))
y = Day.values
```

Step 3: Fitting Linear Regression to the dataset

```
reg = LinearRegression()
reg.fit(X,y)
```

Step 4: Fitting Polynomial Regression to the dataset

```
Poly = PolynomialFeatures(degree=6)
X = Poly.fit_transform(x)
```

Step 5: Visualising the Polynomial Regression results using scatter plot.

```
plt.scatter(x,y)
plt.plot(x,Yp,color='red',linewidth=2)
plt.xlabel("Number of Days")
plt.ylabel("Confirmed cases of Maharashtra")
plt.title("Polynomial Regression",fontsize=15)
plt.show()
```

Step 6: Predicting new result with Polynomial Regression

```
reg.predict(Poly.transform([[123]]))
```

Output:

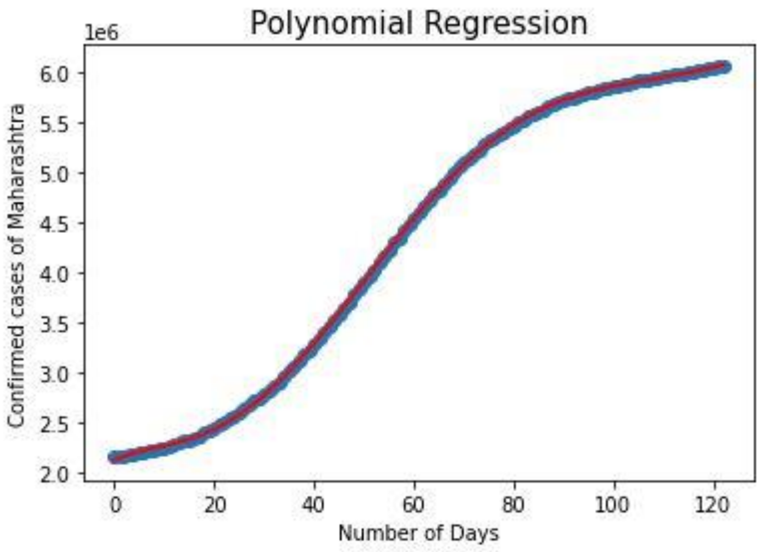


Fig.5.13. PR - Confirmed cases of Maharashtra

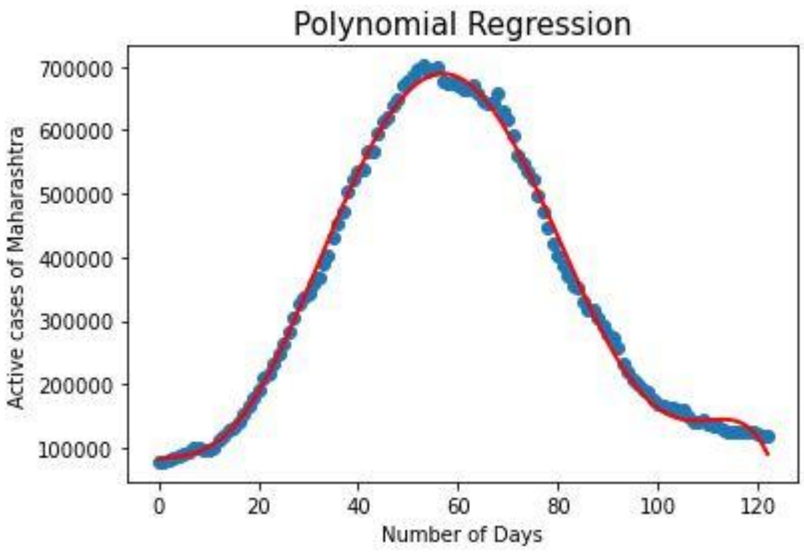


Fig.5.14. PR - Active cases of Maharashtra

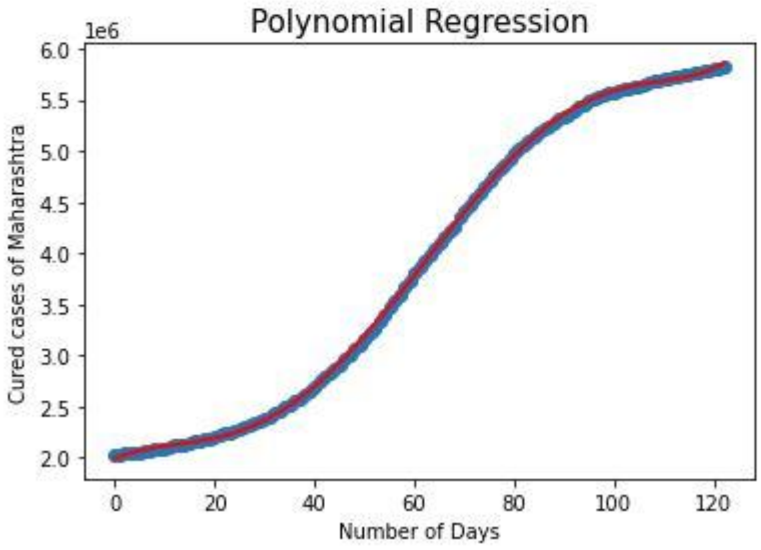


Fig.5.15. PR - Cured of Maharashtra

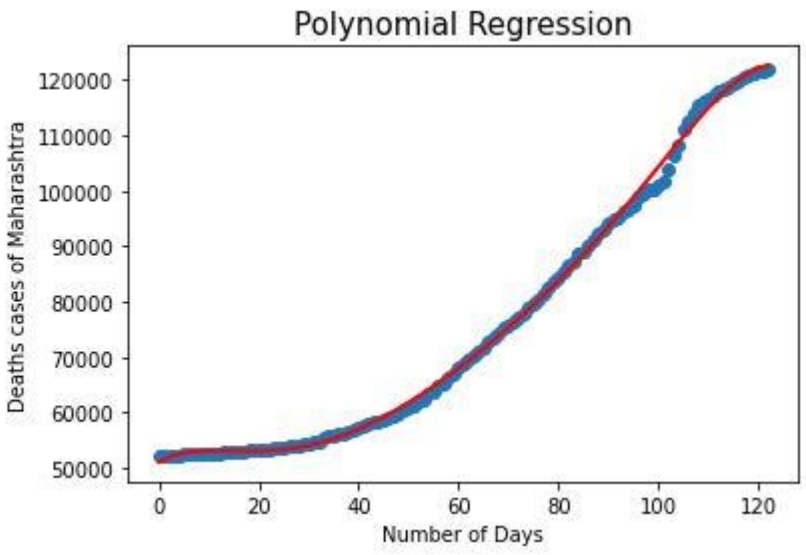


Fig.5.16. PR - Death cases of Maharashtra

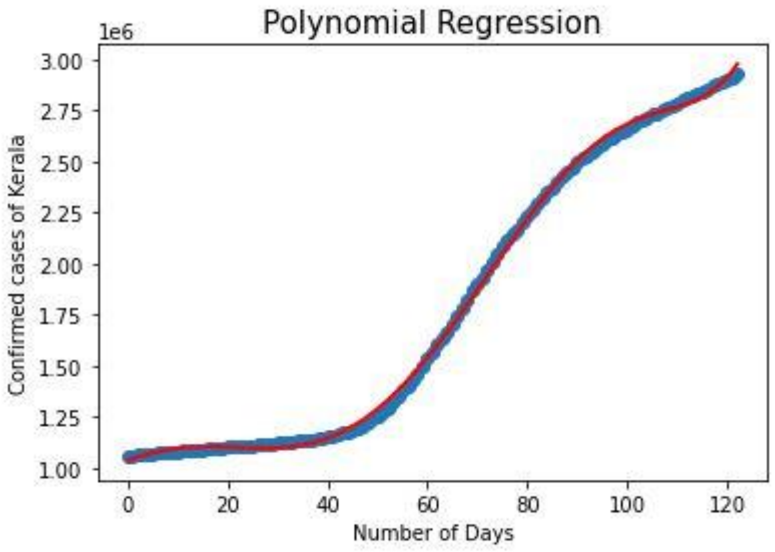


Fig.5.17. PR - Confirmed cases of Kerala

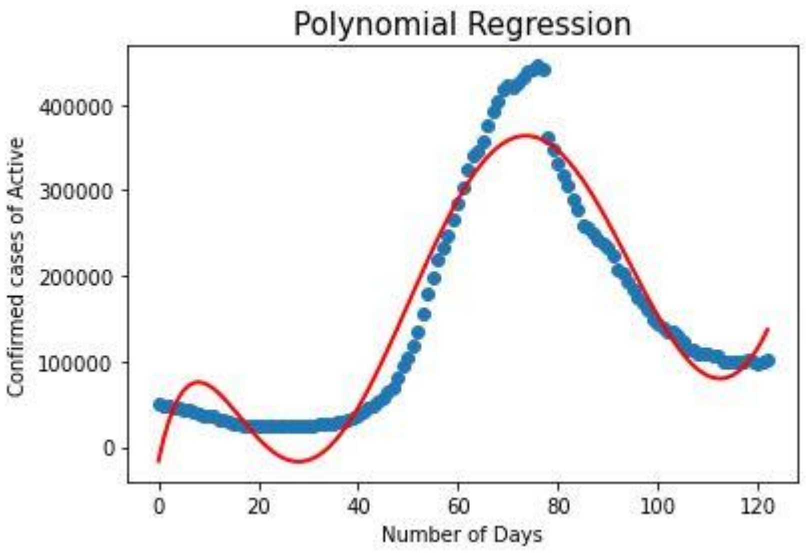


Fig.5.18. PR - Active cases of Kerala

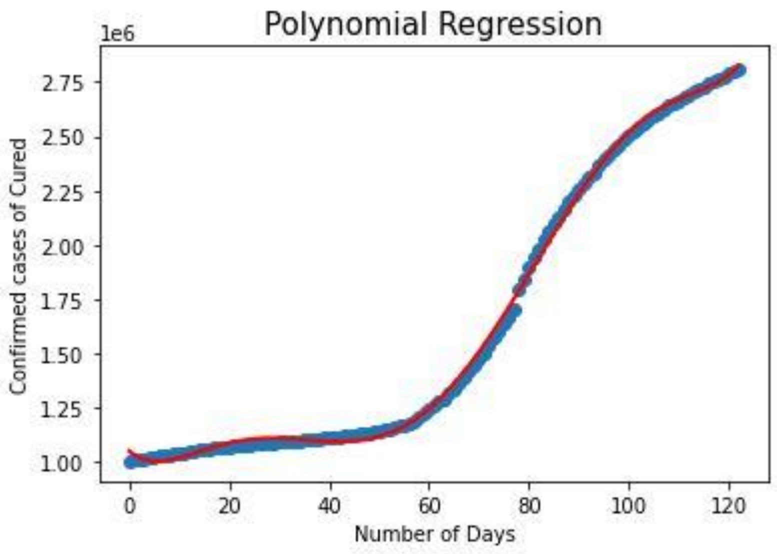


Fig.5.19. PR - Cured of Kerala

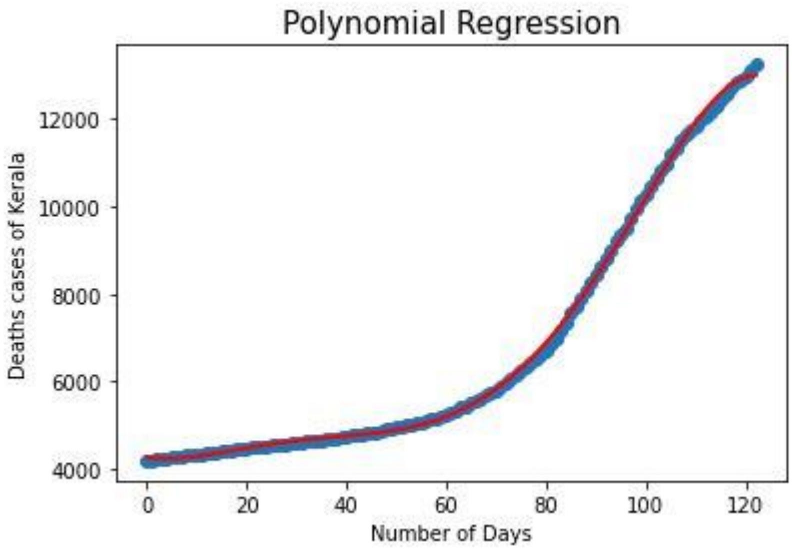


Fig.5.20. PR - Death cases of Kerala

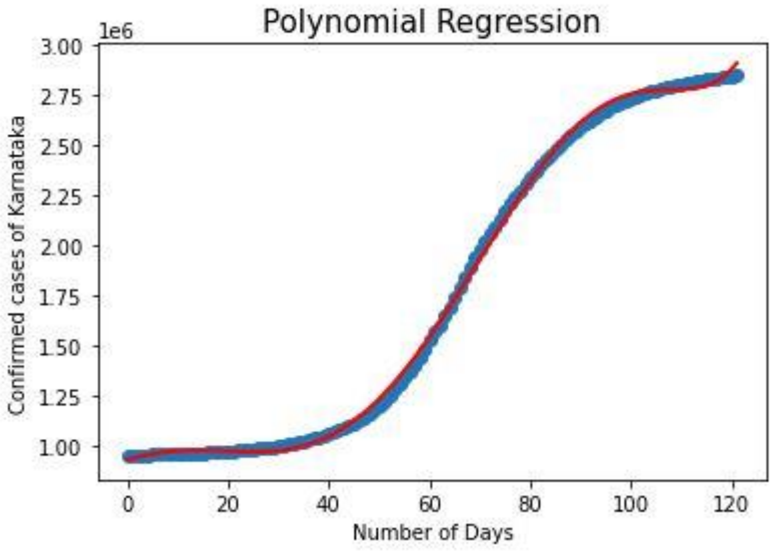


Fig.5.21. PR - Confirmed cases of Karnataka

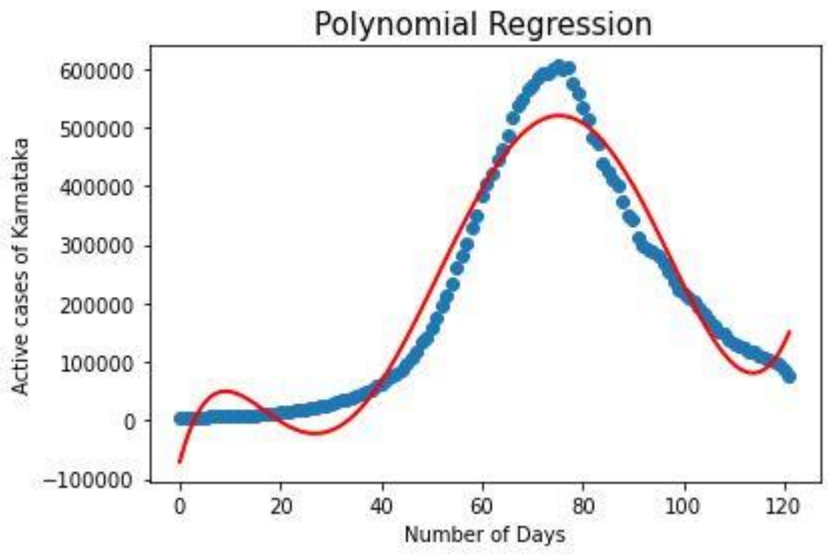


Fig.5.22. PR - Active cases of Karnataka

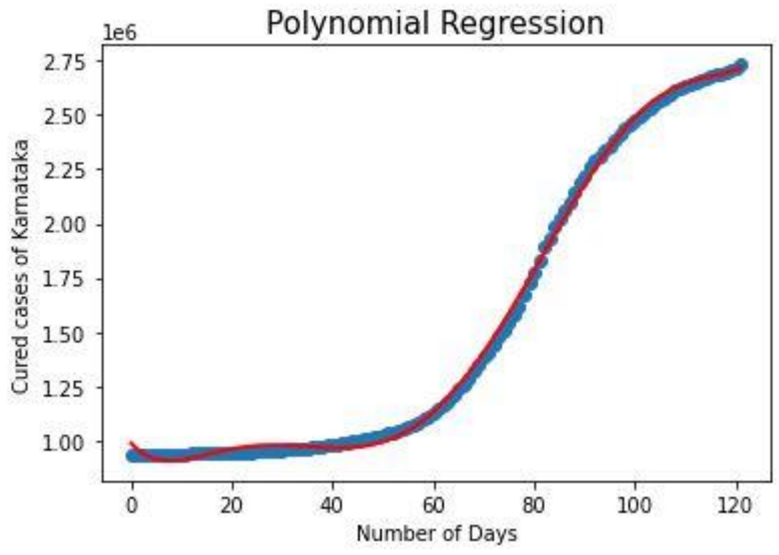


Fig.5.23. PR - Cured of Karnataka

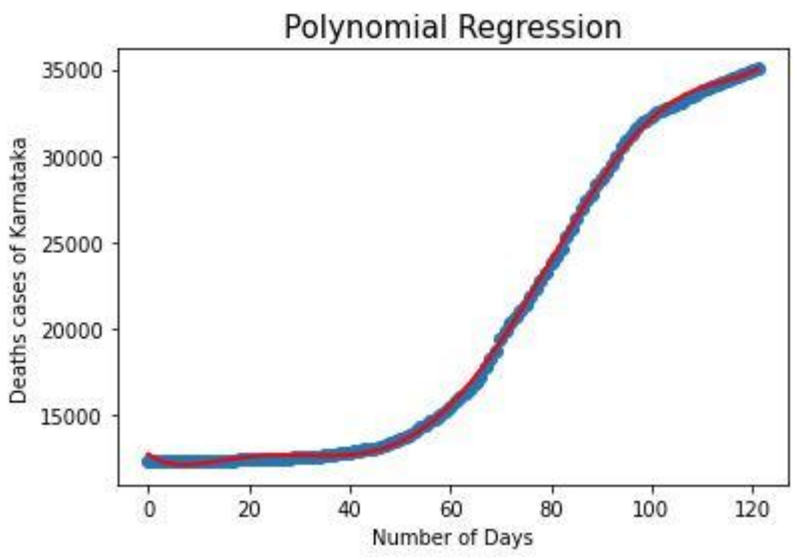
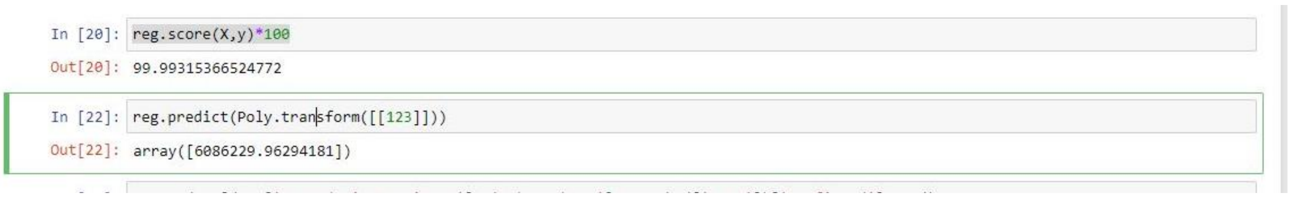


Fig.5.24. PR - Death cases of Karnataka

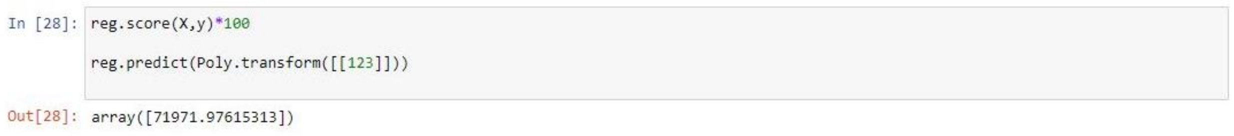
Prediction:



```
In [20]: reg.score(X,y)*100
Out[20]: 99.99315366524772

In [22]: reg.predict(Poly.transform([[123]]))
Out[22]: array([6086229.96294181])
```

Fig.5.25. PR - Prediction of Confirmed cases of Maharashtra

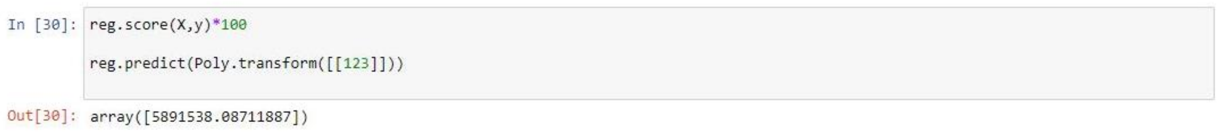


```
In [28]: reg.score(X,y)*100

         reg.predict(Poly.transform([[123]]))

Out[28]: array([71971.97615313])
```

Fig.5.26. PR - Prediction of Active cases of Maharashtra

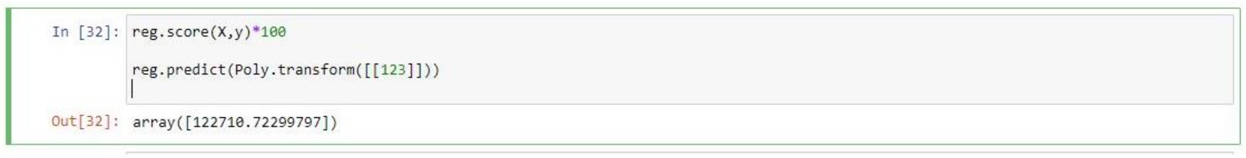


```
In [30]: reg.score(X,y)*100

         reg.predict(Poly.transform([[123]]))

Out[30]: array([5891538.08711887])
```

Fig.5.27. PR - Prediction of Cured of Maharashtra



```
In [32]: reg.score(X,y)*100

         reg.predict(Poly.transform([[123]]))

Out[32]: array([122710.72299797])
```

Fig.5.28. PR - Prediction of Death cases of Maharashtra

Random Forest:

Random Forest is a supervised machine learning algorithm creates randomly a forest with several trees . Why we use random forest instead of decision tree, decision trees are easy to implement and work efficiently with training data, but it gives less accuracy this happens due to over fitting . Over fitting occurs when a model trains the data to such an extent that is negatively impacts the performance of the model on new data. For this reason random forest comes into way. Function: train (formula, dataset, method=”rf”,trControl=trcontrol()) [where “rf” is random forest method].

The Decision Tree is an easily understood and interpreted algorithm and hence a single tree may not be enough for the model to learn the features from it. On the other hand, Random Forest is also a “Tree”-based algorithm that uses the qualities features of multiple Decision Trees for making decisions.

Therefore, it can be referred to as a *‘Forest’* of trees and hence the name “Random Forest”. The term ‘*Random*’ is due to the fact that this algorithm is a forest of ***‘Randomly created Decision Trees’***.

Algorithms:

Step 1 : Import the required libraries.
Step 2 : Import and print the dataset
Step 3 : Select all rows and column 1 from dataset to x and all rows and column 2 as y
Step 4 : Fit Random forest regressor to the dataset
Step 5 : Predicting a new result
Step 6 : Visualising the result

Code:

Step 1 : Import the required libraries.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
```

Step 2 : Import and print the dataset

```
data = pd.read_csv("D:\SEM VI\Seminar ppt\Data\Cleaned\Overall_India.csv")
data.head()
```

Step 3 : Select all rows and column 1 from dataset to x and all rows and column 2 as y

```
x = np.arange(len(Day))
y = Day.values
```

Step 4 : Fit Random forest regressor to the dataset

```
reg = RandomForestRegressor(n_estimators=300)
reg.fit(x,y)
```

Step 5 : Predicting a new result

```
reg.predict([[123]])
```

Step 6 : Visualising the result

```
plt.scatter(x,y)
plt.plot(x,reg.predict(x),color='red',linewidth=2)
plt.xlabel("Number of Days")
plt.ylabel("Confirmed cases of Maharashtra")
plt.title("Random Forest Regression",fontsize=15)
plt.show()
```

Output :

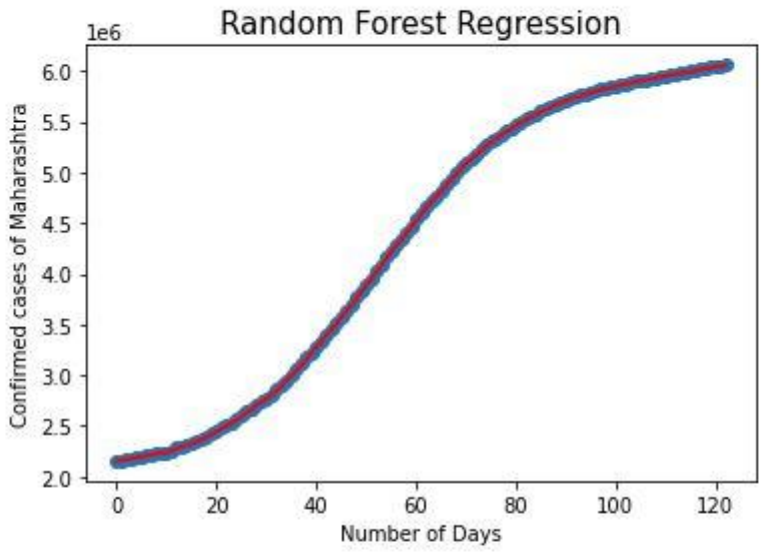


Fig.5.29. RF - Confirmed cases of Maharashtra

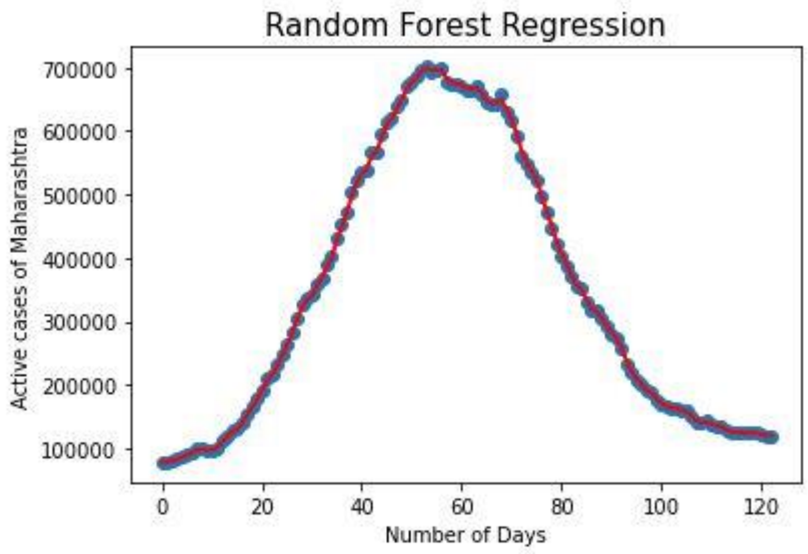


Fig.5.30. RF - Active cases of Maharashtra

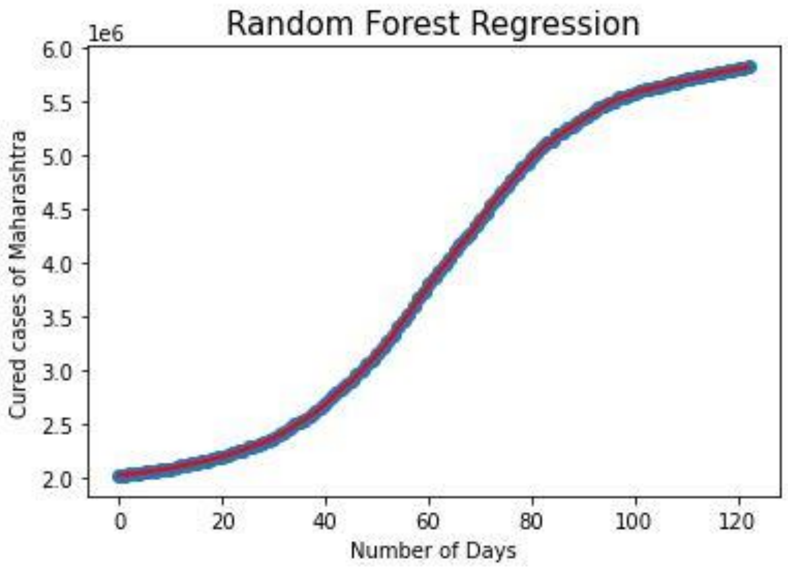


Fig.5.31. RF - Cured of Maharashtra

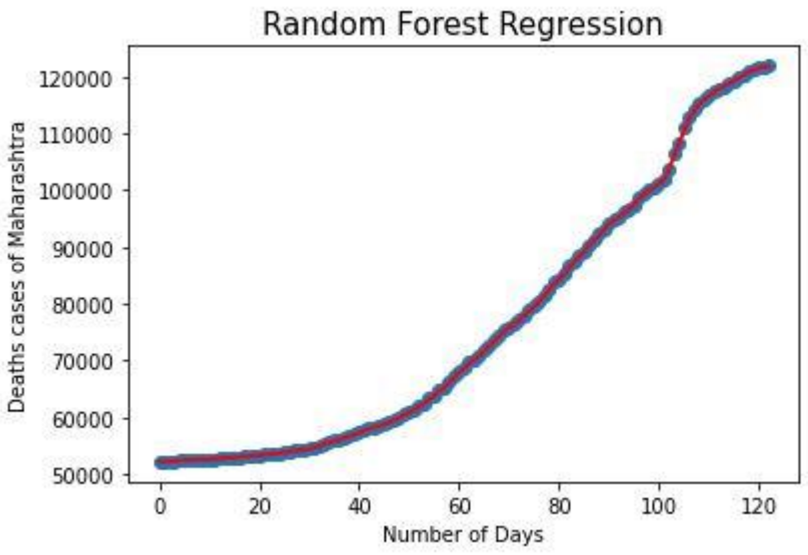


Fig.5.32. RF - Death cases of Maharashtra

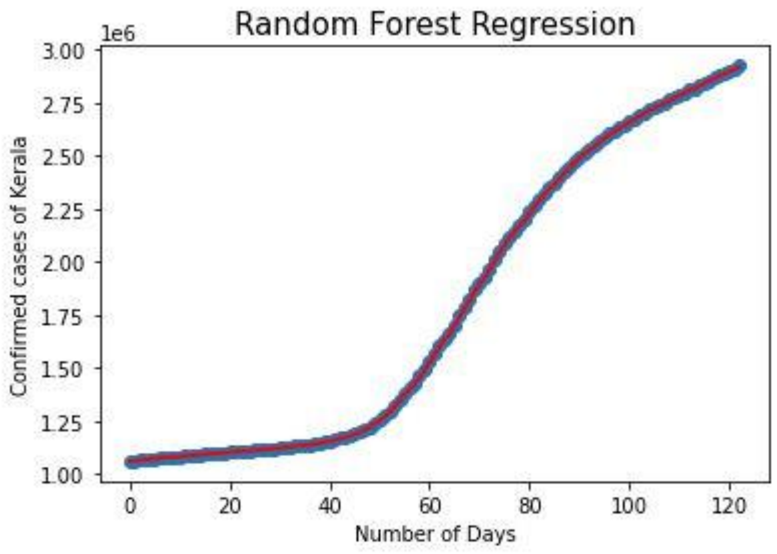


Fig.5.33. RF - Confirmed cases of Kerala

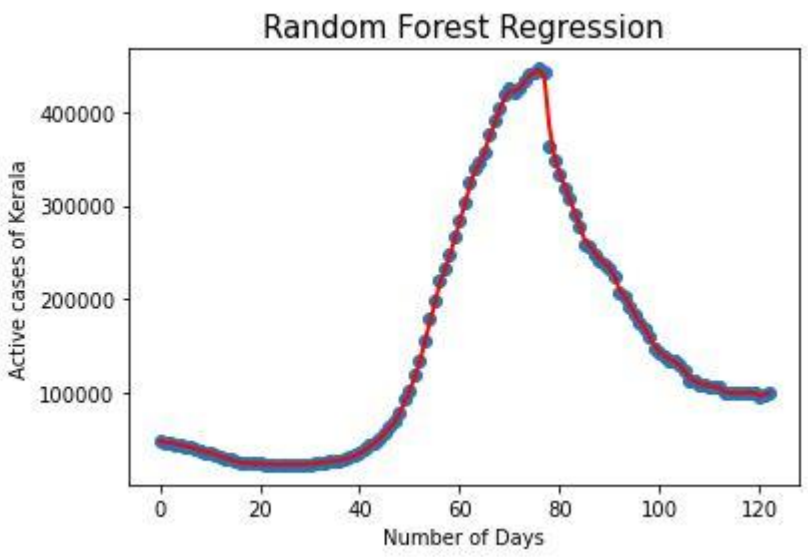


Fig.5.34. RF - Active cases of Kerala

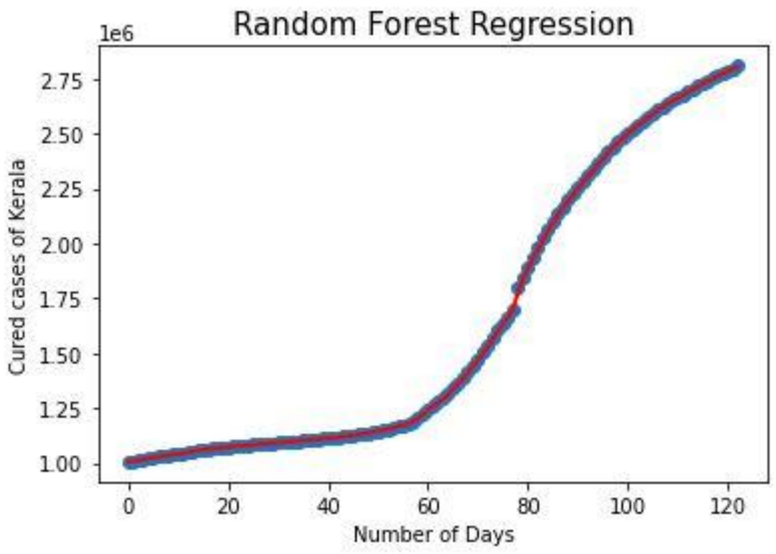


Fig.5.35. RF - Cured of Kerala

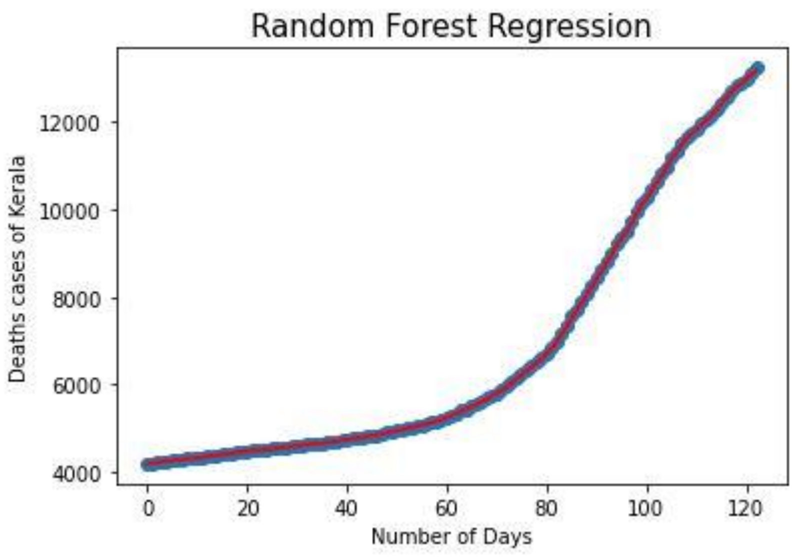


Fig.5.36. RF - Death cases of Kerala

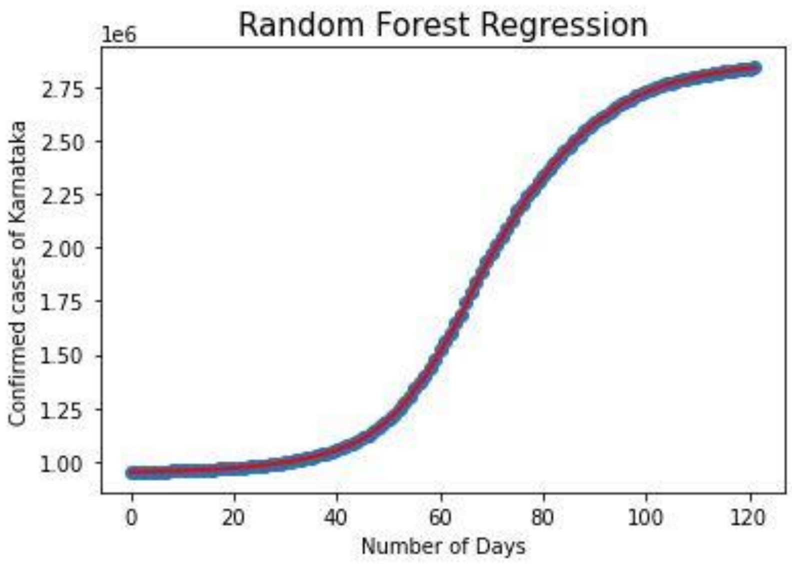


Fig.5.37. RF - Confirmed cases of Karnataka

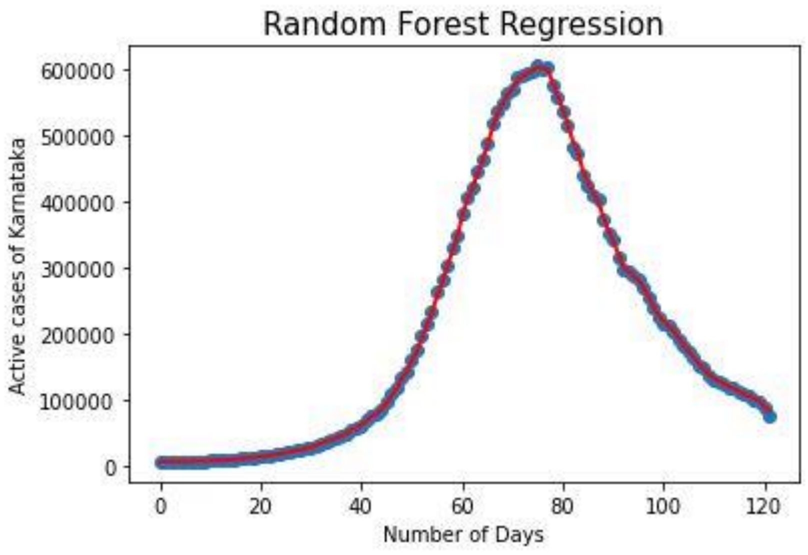


Fig.5.38. RF - Active cases of Karnataka

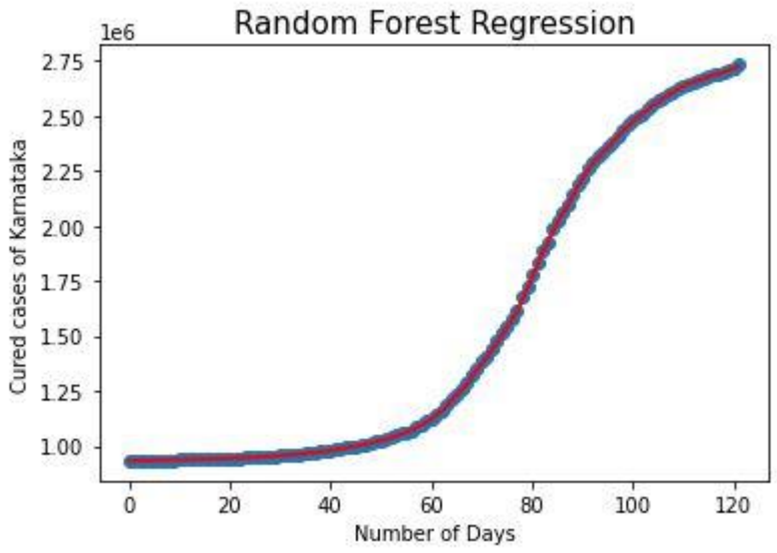


Fig.5.39. RF - Cured of Karnataka

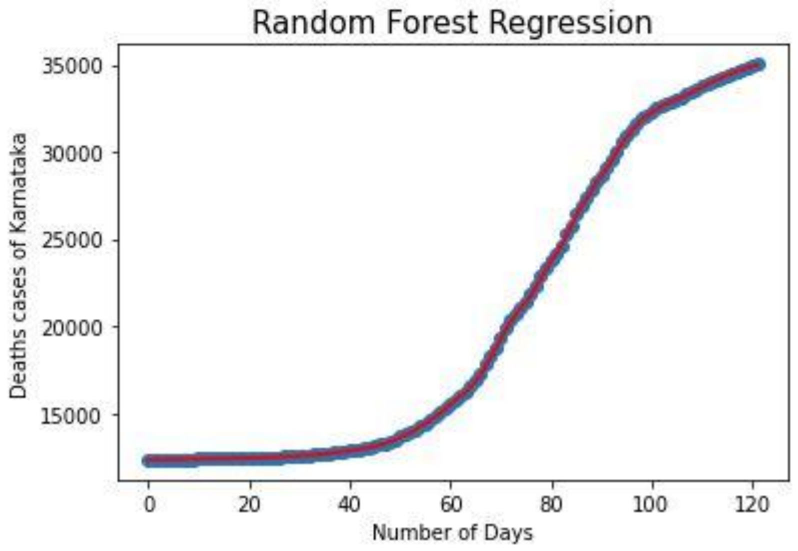


Fig.5.40. RF - Death cases of Karnataka

CHAPTER 6 : RESULT

Predicting mining

Prediction outcome will gives predicted values of a dataset after applying machine learning algorithms. Among all the algorithms the best algorithm with accuracy can be determined. In order to keep with the results random forest technique offers the best result among alternative algorithms.

MODEL COMPARISON:

In this section, we contrast the predictive effects of linear regression, support vector regression, random forest and decision tree models. We choose linear regression as the baseline model and introduce its prediction result into the result comparison.

The baseline model (linear regression)'s accuracy result, which is 80%, is not ideal. Each of the models we select performs better than the baseline model. Support vector regression, Polynomial regression and Random forest are 98%, 99%, 95% respectively. It is observed that Polynomial regression performs the best among them, with the result of 99%.

Compared with the other prediction models, Polynomial regression model plays a prominent role in predicting the cases with higher accuracy.

CHAPTER 7: CONCLUSION

.

Covid cases prediction is a crucial part of the strategic planning process. It allows to forecast how the cases may raise in the future.

Finally we conclude that prediction of covid cases on Indian data has been done and we have observed which prediction model gives accurate results. For predicting the confirmed, active, cured and death cases, we applied several machine learning algorithms (Support vector regression, Polynomial regression, Random Forest, Decision tree,). Among all these algorithms, Polynomial regression (with degree 6) gave us the best accurate result with minimum error rate

Chapter 8: LIMITATIONS

We had some limitations in our study. One limitation is that some of our data were missing. We did not consider the data of 2020 as it will give inappropriate result. However, we still believe that what the model revealed was accurate, even allowing for the faulty recoding. We cannot say what the effects of vaccination would be, but this may have contributed to the relatively low, albeit highly significant, value.

CHAPTER 9: Future Scope

The main objective of data visualization is the overall idea about the data mining model .In data mining most of the times we are retrieving the data from the repositories which are in the hidden form. This is the difficult task for a user. So this visualization on of the data mining model helps us to provide utmost levels of understanding and trust. The data mining models are of two types: Predictive and Descriptive.

The descriptive model identifies the patterns or relationships in data and explores the properties of the data examined. Ex. Clustering, Summarization, Association rule, Sequence discovery etc. Many of the data mining applications are aimed to predict the future state of the data.

Prediction is the process of analysing the current and past states of the attribute and prediction of its future state. The regression involves the learning of function that map data item to real valued prediction variable. In the time series analysis the value of an attribute is examined as it varies over time.

The clusters are defined by studying the behaviour of the data by the domain experts. The term segmentation is used in very specific context; it is a process of partitioning of database in to disjoint grouping of similar tuples. Summarization is the technique of presenting the summarize information from the data. The association rule finds the association between the different attributes. Association rule mining is a two-step process:
Finding all frequent item sets, Generating strong association rules from the frequent item sets. Sequence discovery is a process of finding the sequence patterns in data. This sequence can be used to understand the trend

CHAPTER 10 : REFERENCES

- **R. Kumari et al., "Analysis and predictions of spread, recovery, and death caused by COVID-19 in India," in Big Data Mining and Analytics, vol. 4, no. 2, pp. 65-75, June 2021, doi: 10.26599/BDMA.2020.9020013.**
- **https://www.kaggle.com/sudalairajkumar/covid19-in-india**
- **https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus**
- **https://seaborn.pydata.org/**
- **https://numpy.org/doc/stable/user/whatisnumpy.html**