

# **Analysis & Prediction of covid-19 using Supervised Machine learning**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

by

**VELAGALA TEJA REDDY (Reg. No. 37110816)**

**THOTAKURA KONDALA RAO NAIDU (Reg.No. 37110783)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING  
SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)  
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**March – 2021**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with "A" grade by NAAC**

**12B Status by UGC | Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119**

**[www.sathyabama.ac.in](http://www.sathyabama.ac.in)**



---

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## **BONAFIDE CERTIFICATE**

**This is to certify that this Project Report is the bonafide work of  
VELAGALA TEJA REDDY (Reg. No. 37110816) and THOTAKURA KONDALA  
RAO NAIDU (Reg.No. 37110783) who carried out the project entitled"  
Analysis & Prediction of covid-19 using Supervised Machine learning"  
under my supervision from November 2020 to March 2021.**

**Internal Guide**

**Mr. MURARI. D. KAMALESH M.E., (Ph.D.,)**

**Head of the Department**

**Dr.S.Vigneshwari M.E., Ph.D.,**

**Dr.L.Lakshmanan M.E., Ph.D.,**

---

Submitted for Viva voce Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## **DECLARATION**

I VELAGALA TEJA REDDY (Reg. No. 37110816) and THOTAKURA KONDALA RAO NAIDU (Reg.No. 37110783) hereby declare that theProject Report entitled" Analysis & Prediction of covid-19 using Supervised Machine learning" done by me under the guidance of Mr. MURARI.D.KAMALESH M.E., (Ph.D.,) Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D .**, Dean, School of Computing and **Dr.L.Lakshmanan M.E., Ph.D.**, and **Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. MURARI.D.KAMALESH M.E., (Ph.D.)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Several outbreak prediction models for COVID-19 are being used by officials around the world to make informed decisions and enforce relevant control measures. Among the standard models for COVID-19 global pandemic prediction, simple epidemiological and statistical models have received more attention by authorities, and these models are popular in the media. Due to a high level of uncertainty and lack of essential data, standard models have shown low accuracy for long-term prediction. Although the literature includes several attempts to address this issue, the essential generalization and robustness abilities of existing models need to be improved. This project presents a comparative analysis of machine learning and soft computing models to predict the COVID-19 outbreak as an alternative to susceptible–infected–recovered (SIR) and susceptible-exposed-infectious-removed (SEIR) models. Among a wide range of machine learning models investigated, two models showed promising results. Based on the results reported here, and due to the highly complex nature of the COVID-19 outbreak and variation in its behaviour across nations, this study suggests machine learning as an effective tool to model the outbreak. This project provides an initial bench marking to demonstrate the potential of machine learning for future research. This project further suggests that a genuine novelty in outbreak prediction can be realized by integrating machine learning and SEIR models.

## TABLE OF CONTENTS

| CHAPTER NO. | TITLE                                      | PAGE NO. |
|-------------|--------------------------------------------|----------|
|             | <b>ABSTRACT</b>                            | v        |
|             | <b>LIST OF FIGURES</b>                     | viii     |
|             | <b>LIST OF TABLES</b>                      | ix       |
|             | <b>LIST OF ABBREVIATIONS</b>               | ix       |
| 1           | <b>INTRODUCTION</b>                        | 1        |
|             | 1.1 Introduction                           | 1        |
|             | 1.2 Outline of the Project                 | 2        |
|             | 1.3 Goals of the Project                   | 3        |
|             | 1.4 Motivation for the Project             | 3        |
| 2           | <b>LITERATURE SURVEY</b>                   | 4        |
| 3           | <b>AIM AND SCOPE</b>                       | 5        |
|             | 3.1 Aim and Objective                      | 5        |
|             | 3.2 Scope of the Project                   | 5        |
|             | 3.3 Background                             | 5        |
|             | 3.4 Algorithms                             | 8        |
|             | 3.5 Software Environment                   | 11       |
|             | 3.6 Dataset                                | 11       |
|             | 3.7 Data Collection                        | 15       |
|             | 3.8 Data Pre-Processing                    | 16       |
|             | 3.9 Implementation                         | 20       |
|             | 3.10 Performance Metrics                   | 20       |
| 4           | <b>COVID ANALYSIS AND PREDICTION</b>       | 21       |
|             | 4.1 How Covid Analysis and Prediction work | 21       |
|             | 4.2 Architecture of the Project            | 22       |
|             | 4.3 Analysis and Visualization             | 23       |
| 5           | <b>RESULTS AND DISCUSSIONS</b>             | 27       |
|             | 5.1 Experiment Results                     | 27       |
|             | 5.2 Support Vector Machine                 | 28       |

|   |                                              |    |
|---|----------------------------------------------|----|
|   | 5.3 Polynomial Regression algorithm results  | 29 |
|   | 5.4 Bayesian ridge network algorithm results | 31 |
|   | 5.5 Results Comparison                       | 32 |
| 6 | <b>CONCLUSION AND FUTURE WORK</b>            | 36 |
|   | <b>REFERENCES</b>                            | 37 |
|   | <b>APPENDIX</b>                              |    |
|   | A) Plagiarism Report                         | 39 |
|   | B) Journal Paper                             | 40 |
|   | C) Source code                               | 45 |

## LIST OF FIGURES

| FIGURE NO. | FIGURE NAME                                              | PAGE NO. |
|------------|----------------------------------------------------------|----------|
| 3.1        | Support Vector Machine                                   | 08       |
| 3.2        | Polynomial Regression                                    | 10       |
| 3.3        | Bayesian Network                                         | 10       |
| 4.1        | Architecture Diagram                                     | 22       |
| 4.2        | Tabular Representation of Covid cases and Mortality rate | 25       |
| 4.3        | Animated Graphical Representation                        | 26       |
| 5.1        | Prediction using Support Vector Machine                  | 29       |
| 5.2        | Prediction using Polynomial Regression                   | 30       |
| 5.3        | Prediction using Bayesian Network                        | 31       |
| 5.4        | Algorithm accuracy comparision                           | 32       |
| 5.5        | Total Positive Cases as of Jan 2nd 2021                  | 33       |
| 5.6        | Total Death Cases as of Jan 2nd 2021                     | 34       |
| 5.7        | Total Recovered Cases as of Jan 2nd 2021                 | 34       |

## LIST OF TABLES

| TABLE NO. | TABLE NAME       | PAGE NO. |
|-----------|------------------|----------|
| 5.1       | Accuracy Results | 33       |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION               |
|--------------|-------------------------|
| COVID        | Coronavirus Disease     |
| SVM          | Support Vector Machine  |
| PR           | Polynomial Regression   |
| AI           | Artificial Intelligence |
| CSV          | Comma-Separated Value   |

# **CHAPTER 1**

## **INTRODUCTION**

### **N**

#### **1.1 INTRODUCTION**

Corona viruses are a large family of viruses that are known to cause illness ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). SARS was first seen in 2002 in China and MERS was first seen in 2012 in Saudi Arabia. The latest virus seen in Wuhan, China is called SARS-COV-2 and it causes corona virus. A pneumonia of unknown cause detected in Wuhan, China was first reported to the World Health Organisation (WHO) Country Office in China on 31 December, 2019. Since, then the number of cases of corona virus are increasing along with high death toll. Corona virus spread from one city to whole country in just 30days On Feb 11, it was named as COVID-19 by World Health Organisation (WHO). As this COVID-19 is spread from person to person, Artificial intelligence based electronic devices can play a pivotal role in preventing the spread of this virus. As the role of healthcare epidemiologists has expanded, the pervasiveness of electronic health data has expanded too. We all have been affected by the current COVID-19 pandemic. However, the impact of the pandemic and its consequences are felt differently depending on our status as individuals and as members of society. While some try to adapt to working online, home-schooling their children and ordering food via Online cart, others have no choice but to be exposed to the virus while keeping society functioning. Our different social identities and the social groups we belong to determine our inclusion within society and, by extension, our vulnerability to epidemics. Employing a human rights lens in this analysis helps us translate needs and social problems into rights, focusing our attention on the broader socio-political structural context as the cause of the social problems. Human rights highlight the inherent dignity and worth of all people, who are the primary rights-holders.

Governments (and other social actors, such as corporations) are the duty-bearers, and as such have the obligation to respect, protect, and fulfil human rights. Human rights cannot be separated from the societal contexts in which they are recognized, claimed, enforced, and fulfilled. Specifically, social rights, which include the right to health, can

become important tools for advancing people's citizenship and enhancing their ability to participate as active members of society. Such an understanding of social rights calls our attention to the concept of equality, which requires that we place a greater emphasis on "solidarity" and the "collective." Furthermore, in order to generate equality, solidarity, and social integration, the fulfilment of social rights is not optional. In order to fulfil social integration, social policies need to reflect a commitment to respect and protect the most vulnerable individuals and to create the conditions for the fulfilment of economic and social rights for all. Considering how Humans are overrepresented among those getting infected and dying from COVID-19, experts recommend that more testing be done in minority communities and that more medical services be provided. Although the law requires insurers to cover testing for patients who go to their doctor's office or who visit urgent care or emergency rooms, patients are fearful of ending up with a bill if their visit does not result in a COVID test. Furthermore, minority patients who lack insurance or are underinsured are less likely to be tested for COVID-19, even when experiencing alarming symptoms. On average it will take 5–6 days from when someone is infected with the virus to get symptoms, however it would take up to 14 days to recover. Currently we got the medicine to overcome the virus but the more over yet the trails are going on to find the medicine to make available to common man in reasonable Prices. Till today COVID-19 has been infected the humans of more than 177 countries leads to patients out of which 112 million people are infected with virus among them 2,485,386 lost their lives and 87,798,712 people have been recovered till now.

## **1.2 OUTLINE OF THE PROJECT**

The present data i.e. the number of Active, Death and Recovered Cases can be used for training machine learning algorithms to improve its decision-making in terms of predicting diseases. As of DEC 04, 2020, totally 67+ Million cases of COVID-19 have been registered and total number of deaths are 123000. COVID-19 has spread across the globe with around 213 countries and territories affected. As the rise in number of

cases of infected corona virus quickly outnumbered the available medical resources in hospitals, resulted a substantial burden on the health care systems. Due to the limited availability of resources at hospitals and the time delay for the results of the

medical tests, it is a typical situation for health workers to give proper medical treatment to the patients. As the number of cases to test for corona virus is increasing rapidly day by day, it is not possible to test due to the time and cost factors. In our Project, we would like to use machine learning techniques to predict the infection of corona virus in patients.

### **1.3 GOALS OF THE PROJECT**

Considering the Rate of growth in corona cases using various machine learning algorithms to be trained and used to predict the Future. This prediction of corona cases can help the Government to arrange the requirements for the public which includes their basic needs, food for daily wage workers and medication for the effected people. Due to the insufficiency of COVID test kits, ventilators, oxygen cylinders, hospital beds, and unavailability of proper treatment, it is very essential to analyse the growth rates of positive cases, number of recoveries. So, then some proper arrangements can be made to overcome losses of lives. For example, based on the analysis of these data the government can have the proper information to the number of cases till a certain day, and before that day they can arrange all the necessary medical equipment to prevent losses of lives. Currently machine learning methods are playing crucial role and being widely used in healthcare sector and for having more efficient prediction of COVID-19.

### **1.4 MOTIVATION FOR THE PROJECT**

The main advantage of doing the task would be that this will give the idea about the level of spread, and in according to that, the government and the public can make proper plans to handle this pandemic situation by taking measures to reduce the virus spread by various mitigation and other necessary actions. The actions that are taken till date how efficiently they would result. So, by the prediction people will be able to get awareness of this pandemic situation. And Government can have the proper

information to the number of cases till a certain day, and before that day they can arrange all the necessary medical equipment to prevent losses of lives.

## CHAPTER 2

### LITERATURE SURVEY

| AUTHOR                                             | YEAR OF PUBLICATION | DESCRIPTIONS                                                                                                    | PROS                                   | CONS                                      |
|----------------------------------------------------|---------------------|-----------------------------------------------------------------------------------------------------------------|----------------------------------------|-------------------------------------------|
| J. Homolak                                         | 2020                | Corona Viruses basic test cases have taken and analysis on various aspects have be done to represent the cases. | Analysis of infected cases             | Prediction using various algorithm models |
| Zhai, F.,<br>Zhai, Y.,<br>Cong, C.,<br>Song, T., X | 2020                | Research progress of coronavirus based on bibliometric analysis                                                 | Graphical presentation of covid cases. | Future forcasting of infection            |

## **CHAPTER 3**

### **AIM AND SCOPE**

#### **3.1 AIM AND OBJECTIVE**

The aim of this project is to predict COVID-19 how Positive cases, Death cases, Recovery cases will be in future using machine learning techniques. The prediction is performed using the datasets. The goal is to identify the Upcoming cases if the rate of growth in cases continues. The main objective is to identify the most suitable machine learning technique for prediction, to perform on analysing the datasets. Preparing a machine learning model that could make accurate predictions of COVID-19.

#### **3.2 SCOPE OF THE PROJECT**

This includes the study on how the analysis and prediction process is being done. Selecting the best Algorithm to use which helps for accurate prediction. Machine Learning Algorithm have been used for the Analysis and Prediction. Common people as well as the government will be able to use the prediction effectively to take necessary action so, then some proper arrangements can be made to overcome losses of lives. For example, based on the analysis of these data the government can have the proper information to the number of cases till a certain day, and before that day they can arrange all the necessary medical equipment to prevent losses of lives.

#### **3.3 BACKGROUND**

Machine Learning is a subset of Artificial Intelligence (AI) and was evolved from pattern recognition where the data can be structured for the understanding of the users. Recently, many applications have been developed using Machine Learning in various fields such as healthcare, banking, military equipment, space etc. Currently,

Machine Learning is a rapidly evolving and continuously developing field. It programs computers using data to optimize their performance. It learns the parameters to optimize the computer programs using the training data or its past experiences. Using

the data, it can also predict the future. Machine Learning also helps us in building a mathematical model using the statistics of the data. The main objective of Machine Learning is that it learns from the feed data without any interference of humans that is, it automatically learns from given data (experience) and gives us the desired output where it searches the trends/patterns in the data. It is broadly classified into four types:

- Supervised Machine Learning.
- Unsupervised Machine Learning.
- Semi-Supervised Machine Learning.
- Reinforcement Machine Learning.

### ***3.3.1 Supervised Machine Learning***

Supervised Learning is a Machine Learning model that is built to give out predictions. This algorithm is performed by taking a labelled set of data as input and also known responses as output to learn the regression/classification model. It develops predictive models from classification algorithms and regression techniques.

**Classification:** predicts discrete responses. Here, the algorithm labels by choosing two or more classes for each example. If it is done between two classes then it is called binary classification and if it is done between two or more classes then it is called multi- class classification. Applications of classification includes hand writing recognition, medical imaging etc.

**Regression:** predicts continuous responses. Here, the algorithms returns a statistical value. For example, a set of data is collected such that the people are happy when considered the amount of sleep. Here, sleep and happy are both variables. Now, the analysis is done by making predictions. The types of popular regression techniques are:

- Linear regression.
- Logical regression.

### ***3.3.2 Unsupervised Machine Learning***

Unlike the supervised learning, there is no supervisor here and we only have input data. Here, the basic aim is to find certain patterns in the data that occur more than others. According to the statistics, it is called density estimation. One of the methods for the density estimation is called clustering. Here, the input data is formed into clusters or groupings. Here, the assumptions are made such that the clusters are discovered which will match reasonably well with a classification. This is a data driven approach that works better when provided with sufficient data. For example, the movies in Netflix.com are suggested based on the principal of clustering of movies where several similar movies are grouped based on customer's recently watched movie list. It mostly discovers the unknown patterns in the data but most of the time these approximations are weak when compared with the supervised learning.

### ***3.3.3 Semi-Supervised Machine Learning***

The name "semi-supervised learning" comes from the fact that the data used is between supervised and unsupervised learning [57]. Semi-supervised algorithm has the tendency to learn both from labelled and unlabelled data. Semi-supervised machine learning gives high accuracy with a minimum annotation work. Semi-supervised machine learning uses mostly unlabelled data together combined with labelled data to give better classifiers. As less annotation work is enough to give good accuracy, humans have less work to do here.

### ***3.3.4 Reinforcement Machine Learning***

Reinforcement learning learns its behaviour from a trial-and-error method in a dynamic environment. Here, the problem is solved by taking an appropriate action in a certain situation to maximize the output and to obtain the acquired results. In

Reinforcement Learning, there is presentation of the input or output data. Instead, when the desired action is chosen, the agent is immediately told the reward and the next state are not considering the long terms actions. For the agent to act optimally it should have the

knowledge about states, rewards, transitions and actions actively. Formally, the model consists of:

- A discrete set of environment states,  $S$ ;
- A discrete set of agent actions,  $A$ ;
- A set of scalar reinforcement signals; typically  $\{0;1\}$  or the real numbers.

### **3.4 ALGORITHMS**

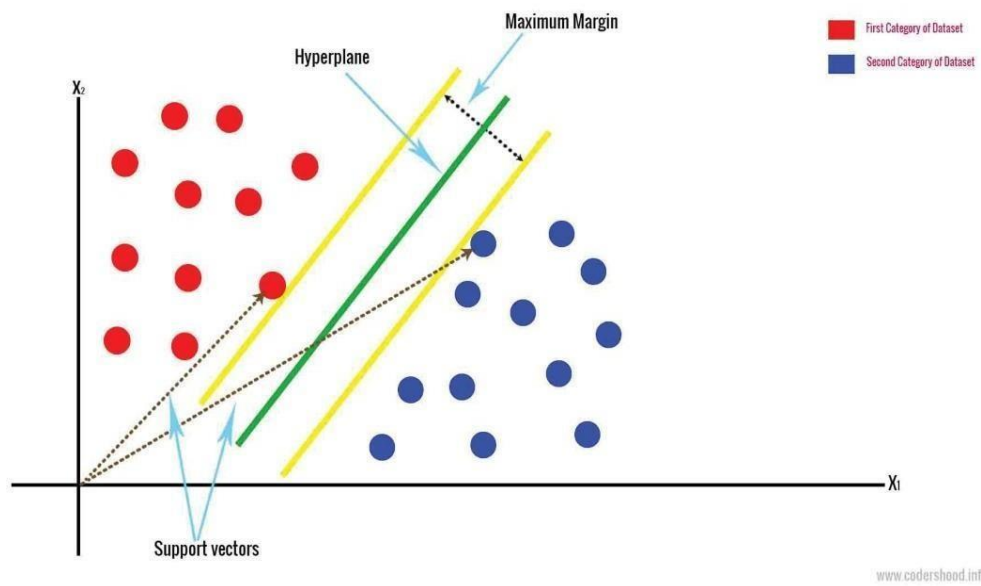
During our research, we have investigated three algorithms through which we have performed supervised classification.

#### ***3.4.1 Support Vector Machine (Svm)***

Support Vector Machines performs classification by constructing N-dimensional hyper plane that separates the data into two categories. In SVM, the predictor variable is called an attribute and the transformed attribute is called a feature. Selecting the most suitable representative data is called feature selection. A set of features describing one case is called a vector. The ultimate goal of SVM modelling is to find the optimal hyper plane that separates the clusters where on one side of the plane there is target variable and on the other side of the plane other category. The vectors

which are near

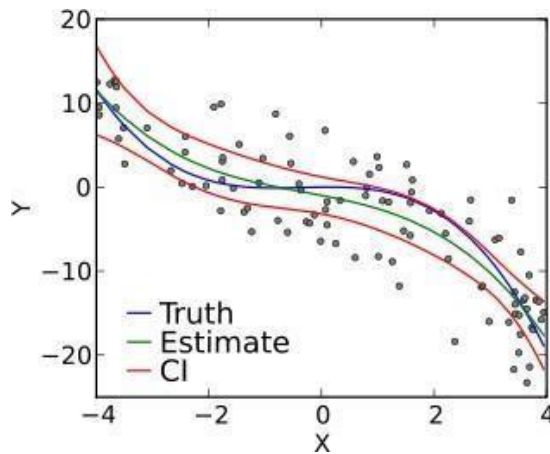
*Fig 3.1: Support Vector Machine*



the hyper plane are the support vectors. In Figure , a typical example of support vector machine is depicted.

### **3.4.2 Polynomial Regression (Pr)**

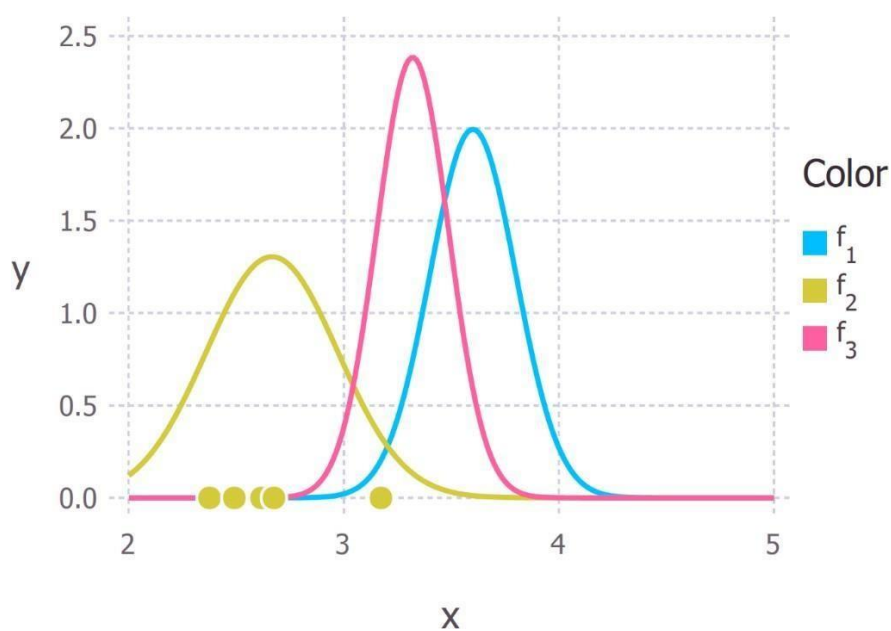
In statistics, polynomial regression is a form of regression analysis in which the relationship between the independent variable  $x$  and the dependent variable  $y$  is modelled as an  $n$ th degree polynomial in  $x$ . Polynomial regression fits a nonlinear relationship between the value of  $x$  and the corresponding conditional mean of  $y$ , denoted  $E(y | x)$ . Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem it is linear, in the sense that the regression function  $E(y | x)$  is linear in the unknown parameters that are estimated from the data. For this reason, polynomial regression is considered to be a special case of multiple linear regression. The explanatory (independent) variables resulting from the polynomial expansion of the "baseline" variables are known as higher-degree terms. Such variables are also used in classification settings. 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 this model, for each unit increase in the value of  $x$ , the conditional expectation of  $y$  increases by  $\beta_1$  units. In many settings, such a linear relationship may not hold. For example, if we are modelling 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. In this model, when the temperature is increased from  $x$  to  $x + 1$  units, the expected yield changes by (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  $x$ :



3.2: Polynomial Regression

### 3.4.3 Bayesian Network

A Bayesian model is a statistical model where you use probability to represent all uncertainty within the model, both the uncertainty regarding the output but also the uncertainty regarding the input to the model. The Bayesian framework for machine learning states that you start out by enumerating all reasonable models of the data and assigning your prior belief  $P(M)$  to each of these models. Then, upon observing the data  $D$ , you evaluate how probable the data was under each of these models to compute  $P(D|M)$ . Bayesian learning has many advantages over other learning programs: Interpolation Bayesian learning methods interpolate all the way to pure engineering. When faced with any learning problem, there is a choice of how much time and effort a human vs. a computer puts in.



*Fig 3.3: Bayesian Network*

### 3.5 SOFTWARE

#### ENVIRONMENT PYTHON

Python is a high level and effective general use programming language. It supports multi-paradigms. Python has a large standard library which provide tools suited to perform various tasks. Python is a simple, less-clustered language with extensive features and libraries. Different programming abilities are utilized for performing the experiment in our work. In this thesis, the following python libraries were used.

- **Pandas** - It is a python package that provides expressive data structures designed to work with both relational and labelled data. It is an open source python library that allows reading and writing data between data structures.
- **Numpy** - It is an open source python package for scientific computing. Numpy also adds fast array processing capacities to python.
- **Matplotlib** - It is an open source python package used for making plots and 2D representations. It integrates with python to give effective and interactive plots for visualization.
- **Tensor flow** - It is a mathematical open source python library designed by Google Brain Team for Machine intelligence.
- **Sklearn** - It is an open source python machine learning library designed to work alongside Numpy. It features various machine learning algorithms for classification, clustering and regression

### 3.6 DATASET

A Dataset is the basic data container in PyMVPA. It serves as the primary form of data storage, but also as a common container for results returned by most algorithms. The dataset assumes that the first axis of the data is to be used to define individual samples, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists

values for each of the variables, such as height and weight of an object.

### 3.6.1 Importing Data In Python

1. Import csv with open ("E:\\customers.csv" ;r) as custfile: rows=csv. reader(custfile, delimiter=',';) for r in rows: print(r)
2. Import pandas as pd df = pd. ExcelFile("E:\\customers.xlsx") data=df.  
...
3. Import pyodbc sql\_conn = pyodbc

#### Python Pandas read\_csv – Load Data

##### From CSV Files

CSV (comma-separated value) files are a common file format for transferring and storing data. The ability to read, manipulate, and write data to and from CSV files using Python is a key skill to master for any data scientist or business analysis. In this post, we'll go over what CSV files are, how to read CSV files into Pandas Data Frames, and how to write Data Frames back to CSV files post analysis. Pandas is the most popular data manipulation package in Python, and Data Frames are the Pandas data type for storing tabular 2D data.

##### Load CSV files to Python Pandas

The basic process of loading data from a CSV file into a Pandas

Data Frame (with all going well) is achieved using the "read\_csv" function in

Pandas:

```
# Load the Pandas libraries with alias ;pd;
```

```
Import pandas as pd
```

```
# Read data from file ;filename.csv;
```

```
# (in the same directory that your python process is
```

```
based) # Control delimiters, rows, column names with
```

```
read_csv data = pd.read_csv("&quot;filename.csv&quot;")
```

```
# Preview the first 5 lines of the loaded data
```

```
data.head()
```

While this code seems simple, an understanding of three fundamental concepts is required to fully grasp and debug the operation of the data loading procedure if you run into issues:

1. Understanding file extensions and file types – what do the letters CSV actually mean? What's the difference between a .csv file and a .txt file?
2. Understanding how data is represented inside CSV files – if you open a CSV file, what does the data actually look like?
3. Understanding the Python path and how to reference a file – what is the absolute and relative path to the file you are loading? What directory are you working in?
4. CSV data formats and errors – common errors with the function.

Each of these topics is discussed below, and we finish this tutorial by looking at some more advanced CSV loading mechanisms and giving some broad advantages and disadvantages of the CSV format.

## 1. File Extensions and File Types

The first step to working with comma-separated-value (CSV) files is understanding the concept of file types and file extensions.

1. Data is stored on your computer in individual "files", or containers, each with a different name.
2. Each file contains data of different types – the internals of a Word document is quite different from the internals of an image.
3. Computers determine how to read files using the "file extension", that is the code that follows the dot (".") in the filename.
4. So, a filename is typically in the form "<random name>.<file extension>". Examples:

- project1.DOCX – a Microsoft Word file called Project1.
- shanes\_file.TXT – a simple text file called shanes\_file
- IMG\_5673.JPG – An image file called IMG\_5673.

- Other well known file types and extensions include: XLSX: Excel, PDF: Portable Document Format, PNG – images, ZIP – compressed file format, GIF – animation, MPEG – video, MP3 – music etc. See a complete list of extensions [here](#).

5. A CSV file is a file with a “.csv” file extension, e.g. “data.csv”, “super\_information.csv”. The “CSV” in this case lets the computer know that the data contained in the file is in “comma separated value” format, which we’ll discuss below. File extensions are hidden by default on a lot of operating systems. The first step that any self-respecting engineer, software engineer, or data scientist will do on a new computer is to ensure that file extensions are shown in their Explorer (Windows) or Finder (Mac) windows. Folder with file extensions showing. Before working with CSV files, ensure that you can see your file extensions in your operating system. Different file contents are denoted by the file extension, or letters after the dot, of the file name. e.g. TXT is text, DOCX is Microsoft Word, PNG are images, CSV is comma-separated value data. To check if file extensions are showing in your system, create a new text document with Notepad (Windows) or Text Edit (Mac) and save it to a folder of your choice. If you can’t see the “.txt” extension in your folder when you view it, you will have to change your settings.

- In Microsoft Windows: Open Control Panel > Appearance and Personalization. Now, click on Folder Options or File Explorer Option, as it is now called > View tab. In this tab, under Advance Settings, you will see the option Hide extensions for known file types. Uncheck this option and click on Apply and OK.
- In Mac OS: Open Finder > in menu, click Finder > Preferences, Click Advanced, Select the checkbox for “Show all filename extensions”.

2. Data Representation in CSV files A “CSV” file, that is, a file with a “csv” file type, is a basic text file. Any text editor such as Notepad on windows or Text Edit on Mac, can open a CSV file and show the contents. Sublime Text is a wonderful and multi-functional text editor option for any platform. CSV is a standard for storing tabular

data in text format, where commas are used to separate the different columns, and

newlines (carriage return / press enter) used to separate rows. Typically, the first row in a CSV file contains the names of the columns for the data.

### **3.7 DATA COLLECTION**

A data set is a collection of data In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. Preparing Your Dataset for Machine Learning: 8 Basic Techniques That Make Your Data Better

1. Articulate the problem early.
2. Establish data collection mechanisms.
3. Format data to make it consistent.
4. Reduce data.
5. Complete data cleaning.
6. Decompose data.
7. Rescale data.
8. Discretize data

Data collection was an essential and protracted process. Regardless the field of research, accuracy of the data collection is essential to maintain cohesion. As the clinical information of patients was not publicly available, it was an inflexible and tedious process to collect the data.so we gathered the data from the kaggle where the tremendous of can be available as per requirement we can search and get the data from these website.

- 1) Kaggle: A data science site that contains a variety of externally-contributed interesting datasets.
- 2) UCI Machine Learning Repository: One of the oldest sources of datasets on the

web, and a great first stop when looking for interesting datasets.

### 3.8 DATA PRE-PROCESSING:

When we talk about data, we usually think of some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc.

Machines don't understand free text, image or video data as it is, they understand 1s and 0s. So it probably won't be good enough if we put on a slideshow of all our images and expect our machine learning model to get trained just by that. In any Machine Learning process, Data Pre-processing is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

#### Features in Machine Learning

- A dataset can be viewed as a collection of data objects, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities. Data objects are described by a number of features that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc. Features are often called as variables, characteristics, fields, attributes, or dimensions.
- A feature is an individual measurable property or characteristic of a phenomenon being observed
- For instance, colour, mileage and power can be considered as features of a car. There are different types of features that we can come across when we deal with data. Now that we have gone over the basics, let us begin with the steps of Data Pre-processing. Remember, not all the steps are applicable for each problem, it is highly dependent on the data we are working with, so maybe only a few steps might be required with your dataset. Generally they are:
  - Data Quality Assessment

- Feature Aggregation
- Feature Sampling
- Dimensionality Reduction
- Feature Encoding

### **3.8.1 Data Quality Assessment**

Because data is often taken from multiple sources which are normally not too reliable and that too in different formats, more than half our time is consumed in dealing with data quality issues when working on a machine learning problem. It is simply unrealistic to expect that the data will be perfect. There may be problems due to human error, limitations of measuring devices, or flaws in the data collection process. Let's go over a few of them and methods to deal with them:

#### **Missing Values:**

It is very much usual to have missing values in your dataset. It may have happened during data collection, or maybe due to some data validation rule, but regardless missing values must be taken into consideration.

#### **Eliminate Rows with Missing Data:**

Simple and sometimes effective strategy. Fails if many objects have missing values. If a feature has mostly missing values, then that feature itself can also be eliminated.

#### **Estimate Missing Values:**

If only a reasonable percentage of values are missing, then we can also run simple interpolation methods to fill in those values. However, most common method of dealing with missing values is by filling them in with the mean, median or mode value of the respective feature.

#### **Inconsistent Values:**

We know that data can contain inconsistent values. Most probably we have already faced this issue at some point. For instance, the 'Address' field contains the 'Phone number'. It may be due to human error or maybe the information was misread while being scanned from a handwritten form. It is therefore always advised to perform data assessment like knowing what the data type of the features should be and whether it is the same for all the data objects.

#### **Duplicate Values:**

A dataset may include data objects which are duplicates of one another. It may happen when say the same person submits a form more than once. The term deduplication is

often used to refer to the process of dealing with duplicates. In most cases, the duplicates are removed so as to not give that particular data object an advantage or bias, when running machine learning algorithms.

### **3.8.2 Feature Aggregation**

Feature Aggregations are performed so as to take the aggregated values in order to put the data in a better perspective. Think of transactional data, suppose we have day- to-day transactions of a product from recording the daily sales of that product in various store locations over the year. Aggregating the transactions to single store-wide monthly or yearly transactions will help us reducing hundreds or potentially thousands of transactions that occur daily at a specific store, thereby reducing the number of data objects.

- This results in reduction of memory consumption and processing time
- Aggregations provide us with a high-level view of the data as the behaviour of groups or aggregates is more stable than individual data objects Sampling is a very common method for selecting a subset of the dataset that we are analysing.

In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more expensive, machine learning algorithm.

The key principle here is that the sampling should be done in such a manner that the sample generated should have approximately the same properties as the original dataset, meaning that the sample is representative. This involves choosing the correct sample size and sampling strategy. Simple Random Sampling dictates that there is an equal probability of selecting any particular entity. It has two main variations as well:

- Sampling without Replacement: As each item is selected, it is removed from the set of all the objects that form the total dataset.

- Sampling with Replacement: Items are not removed from the total dataset after getting selected. This means they can get selected more than once.

Although Simple Random Sampling provides two great sampling techniques, it can fail to output a representative sample when the dataset includes object types which vary drastically in ratio. This can cause problems when the sample needs to have a proper representation of all object types

Train / Validation / Test Split: After feature encoding is done, our dataset is ready for the exciting machine learning algorithms! But before we start deciding the algorithm which should be used, it is always advised to split the dataset into 2 or sometimes 3 parts. Machine Learning algorithms, or any algorithm for that matter, has to be first trained on the data distribution available and then validated and tested, before it can be deployed to deal with real-world data.

Training data: This is the part on which your machine learning algorithms are actually trained to build a model. The model tries to learn the dataset and its various characteristics and intricacies, which also raises the issue of Overfitting v/s under fitting.

Validation data: This is the part of the dataset which is used to validate our various model fits. In simpler words, we use validation data to choose and improve our model hyper parameters. The model does not learn the validation set but uses it to get to a better state of hyper parameters.

Test data: This part of the dataset is used to test our model hypothesis. It is left untouched and unseen until the model and hyper parameters are decided, and only after that the model is applied on the test data to get an accurate measure of how it would perform when deployed on real-world data.

Split Ratio: Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training purposes (usually the case) For instance, training on textual data, image data, or video data usually involves thousands of features! If the model has a lot of hyper parameters that can be tuned, then keeping a higher percentage of data for the validation set is

advisable. Models with less number of hyper parameters are easy to tune and update, and so we can keep a smaller validation set. Like many other things in Machine Learning, the split ratio is highly dependent on the problem we are trying to solve and

must be decided after taking into account all the various details about the model and the dataset in hand.

There are seven significant steps in data pre-processing in Machine Learning:

1. Acquire the dataset.
2. Import all the crucial libraries.
3. Import the dataset.
4. Identifying and handling the missing values.
5. Encoding the categorical data.
6. Splitting the dataset.
7. Feature scaling.

### **3.9 IMPLEMENTATION**

The experiment was conducted in the jupyter notebook, which is a default integrated development and learning environment for python. The experiment was conducted in various phases that are mentioned below:

- I) providing the datasets of confirmed, Deaths, Recovery cases.
- II) Analysing the transmission rate of the virus.
- III) Train the Machine with the datasets.
- IV) Selection of accurate algorithm for prediction.
- V) Predicting the spread of coronavirus.

### **3.10 PERFORMANCE METRICS**

It is an essential task to measure the performance of a machine learning model. As our model requires classification, we have used accuracy as the performance metric. Accuracy is the metric used in this thesis for evaluation of the algorithms. It is the most used performance metric to evaluate classification techniques. This measure

allows us to understand which model is best at identifying patterns in training set to give better predictions in the unknown test data-set.

## CHAPTER 4

### COVID ANALYSIS AND PREDICTION

#### 4.1 HOW COVID ANALYSIS AND PREDICTION WORK

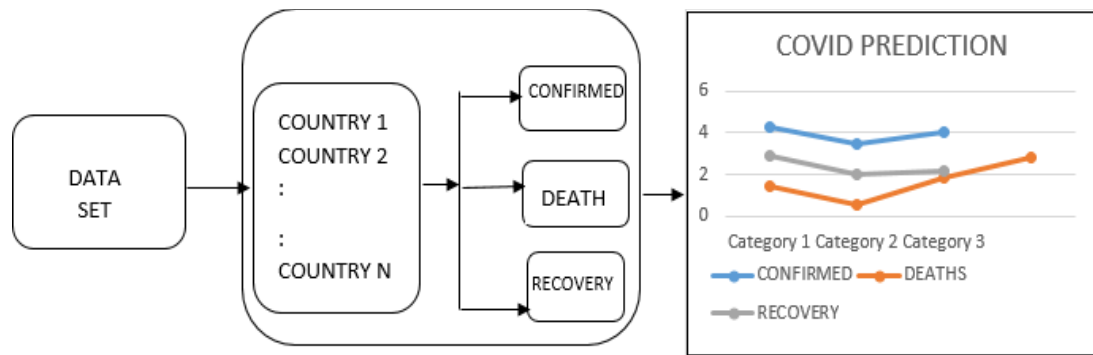
The parameters of several simple mathematical models (i.e., logistic, linear, logarithmic, quadratic, cubic, compound, power and exponential) were fitted using GA, PSO, and GWO. The logistic model outperformed other methods and showed promising results on training for 30 days. Extrapolation of the prediction beyond the original observation range of 30-days should not be expected to be realistic considering the new statistics. The fitted models generally showed low accuracy and also weak generalization ability for the five countries. Although the prediction for China was promising, the model was insufficient for extrapolation, as expected. In turn, the logistic GWO outperformed the PSO and GA and the computational cost for GWO was reported as satisfactory. Consequently, for further assessment of the ML models, the logistic model fitted with GWO was used for comparative analysis. In the next step, for introducing the machine learning methods for time-series prediction, two scenarios were proposed. Scenario 1 considered four data samples from the progress of the infection from previous days, as reported in table 3. The sampling for data processing was done weekly for scenario 1. However, scenario 2 was devoted to daily sampling for all previous consecutive days. Providing these two scenarios expanded the scope of this study. Training and test results for the two machine learning models (MLP and ANFIS) were considered for the two scenarios. A detailed investigation was also carried out to explore the most suitable number of neurons. For the MLP, the performances of using 8, 12 and 16 neurons were analysed throughout the study. For the ANFIS, the membership function (MF) types of Tri, Trap, and Gauss were analysed throughout the study. The five countries of Italy, China, Iran, Germany, and USA were considered. The performance of both ML models for these countries varied amongst the two different scenarios. Given the observed results, it is not possible to select the most suitable scenario. Therefore, both daily and weekly

sampling can be used in machine learning modelling. Comparison between analytical and machine learning models using the deviation from the target value indicated that the MLP in both scenarios delivered the most accurate results. Extrapolation for long-term prediction of

up to 150 days using the ML models was tested. The actual prediction of MLP and ANFIS for the five countries was reported which showed the progression of the outbreak. 3.2 Study Design and Data Source. We used the current daily data on number of COVID-19 infected cases, recoveries, and deaths in India to predict the number of infected and removed cases at any given time (L. Wang et al., 2020). We obtained the data (up to April 14) from the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University Centre for Systems Science and Engineering (JHUCSSE) and from covid19india.org (covid19india, 2020; Donegal. 2020; Johns Hopkins University Centre for Systems Science and Engineering, n.d.). Some of our testing data came from <https://ourworldindata.org/coronavirus>. Statistical Model for Predictions. The ESIR model works by assuming that the true underlying probabilities of the three compartments follow a latent Markov transition process, and that we observe only the daily proportions of infected cases and removed cases.

#### **4.2 ARCHITECTURE OF THE PROJECT:**

For this project we are going to collect the data from the various platforms which was the real time data set we will prepare the datasets as we require in separated way of confirmed cases, death cases & recovered cases. The datasets which we are preparing will be with the dates as follows the longitude & latitude and the country name followed by the number of cases on that particular date. In the same format we are going to prepare the datasets for all three confirmed, death, recovered cases. Here below one the architecture for our project.



*Fig 4.1 : Architecture Diagram*

The Datasets contains the around 199 countries data of the people who are infected with corona virus and Number of deaths and recovery cases will be mentioned as follows .Here the data sets are initially provided later the data gets pre-processed by removing the data cells which contains no cases and the complete data will be initialized they will be analysed in accordingly as we required and here we performed with the best visualization techniques like bar graph representation, pie chart and bar chart race was included to present the comparison among the different provisions in a well stated way.

### **4.3 ANALYSIS AND VISUALIZATION:**

We are using the gradient function to represent the COVID cases of the countries/ provinces in a better way by colour variation representation. With the rapid spread in the novel corona-virus across countries, the World Health Organisation (WHO) and several countries have published latest results on the impact of COVID-19 over the past few months.I have been going through many sources and articles to understand the fatality trend and I was excited to come across this data source and decided to see some visualization on the same. The aim here is to understand how visualization helps to derive informative insights from data sources. For the visualization part, I am using Plotly is a visualization tool available in python which supports a number of interactive, high-quality graphs and is a great tool for data science beginners. The csv file contains information on the affected countries which helps to identify the virus spread, information on infected cases, number of deaths and recoveries across countries. The country co-ordinates are also provided for analysis. A time series data which contains the counts on infected cases, deaths and recoveries across countries is also given. The time series data has individual files for each case and needs to be processed before visualization. The country co-ordinates are also provided for time series visualization on geo plots such as Choropleth Maps. It is crucial to show how rapidly the virus has spread in different countries in a short span of time. Time line analysis would require some pre-processing on the original data to be visualized in

Plotly which is provided in the notebook. Here I have only shown the trend for infected cases. A similar trend can also be observed for deaths and recoveries across countries. The most intriguing part is to compare the impact of COVID-19 to Severe Acute Respiratory

Syndrome (SARS) which is similar in nature. The dataset for SARS was obtained from kaggle data source. The data only contained overall infected and mortality cases and thus visualizing the same here. I have analysed the impact of both viruses in a window frame starting from its discovery to its impact in the next three months. While our initial charts support the comparison of different regions, the tool required the user to drive the process of choosing the regions to compare. From the beginning, we felt the need to have an automatic tool that could, given a region of interest, return the closest regions given a similarity function. Looking at the disease's spread in distinct places, but with similar growth patterns, can be useful to predict behaviours. Thus, we developed a search engine to support queries using different similarity functions. For example, we show the results of searching for regions similar to Italy concerning the number of deaths. The results are listed in a ranking, with pairwise comparisons that detail attributes of the different locations and evolution charts, aligned by the day of the first death. We observe the similarities in the evolution in the number of deaths and cases for Italy, France, Spain, and the United Kingdom. While using the tool, we also saw similarities among cities from Brazil and the United States, both countries with large COVID-19 numbers. The examples so far give a glimpse of how data visualization can help in the understanding of COVID-19. Figure 3 illustrates other applications where data visualization can help. The first example shows how multidimensional projections and network visualizations can help the *literature exploration* of papers that describe novel coronavirus research. As new research about COVID-19 is published, there is a great need to review up-to-date literature and treatments conveniently. Contact tracing is another application that relies on graph visualization to trace the network of people who may have been in contact with a COVID-19 patient, an activity essential to control the dissemination of the disease and essential for directing social distancing regulations. Graph visualization is also important in for social media spread and fact checking. With many people at home, social networks are playing a significant role in people's lives these days. Unfortunately, the dissemination of fake news and automated posting from robots is also rising. Fact-checking over the propagation network can help identify misleading

information and patterns of dissemination. The fourth and final example highlights the importance of data visualization in the analysis of scientific simulations. It shows the visualization of simulating the transport and spread of novel coronavirus in closed

spaces, which shows how an infected person can disseminate the virus indoors. Many other examples include data visualization in the analysis of COVID-19 data, and many more is surfacing every day. We hope this summary highlights interesting

|    | Country Name   | Number of Confirmed Cases | Number of Deaths | Number of Recoveries | Number of Active Cases | Mortality Rate |
|----|----------------|---------------------------|------------------|----------------------|------------------------|----------------|
| 0  | US             | 14139577                  | 276325           | 5404018              | 8459234                | 0.019543       |
| 1  | India          | 9571559                   | 139188           | 9016289              | 416082                 | 0.014542       |
| 2  | Brazil         | 6487084                   | 175270           | 5792539              | 519275                 | 0.027018       |
| 3  | Russia         | 2354934                   | 41173            | 1843526              | 470235                 | 0.017484       |
| 4  | France         | 2310271                   | 54231            | 172574               | 2083466                | 0.023474       |
| 5  | United Kingdom | 1678418                   | 60210            | 3623                 | 1614585                | 0.035873       |
| 6  | Spain          | 1675902                   | 46038            | 150376               | 1479488                | 0.027471       |
| 7  | Italy          | 1664829                   | 58038            | 846809               | 759982                 | 0.034861       |
| 8  | Argentina      | 1447732                   | 39305            | 1274675              | 133752                 | 0.027149       |
| 9  | Colombia       | 1343322                   | 37305            | 1233115              | 72902                  | 0.027771       |
| 10 | Mexico         | 1144643                   | 108173           | 843231               | 193239                 | 0.094504       |

examples,

give pointers to other references, and motivate people to pursue other applications.

*Fig 4.2: Tabular Representation of COVID Cases and Mortality Rate*

For visualization we used various functions which are imported from the various libraries as follows:

**Seaborn** is a python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. It was originally developed at Stanford University and is widely used for plotting and visualizing data. There are several advantages:

1. It possesses built-in themes for better visualizations
2. It has tools built in statistical functions which reveal hidden patterns in the data set.
3. It has functions to visualize matrices of data which become very important when visualizing large data sets.
4. It possesses built-in themes for better visualizations.

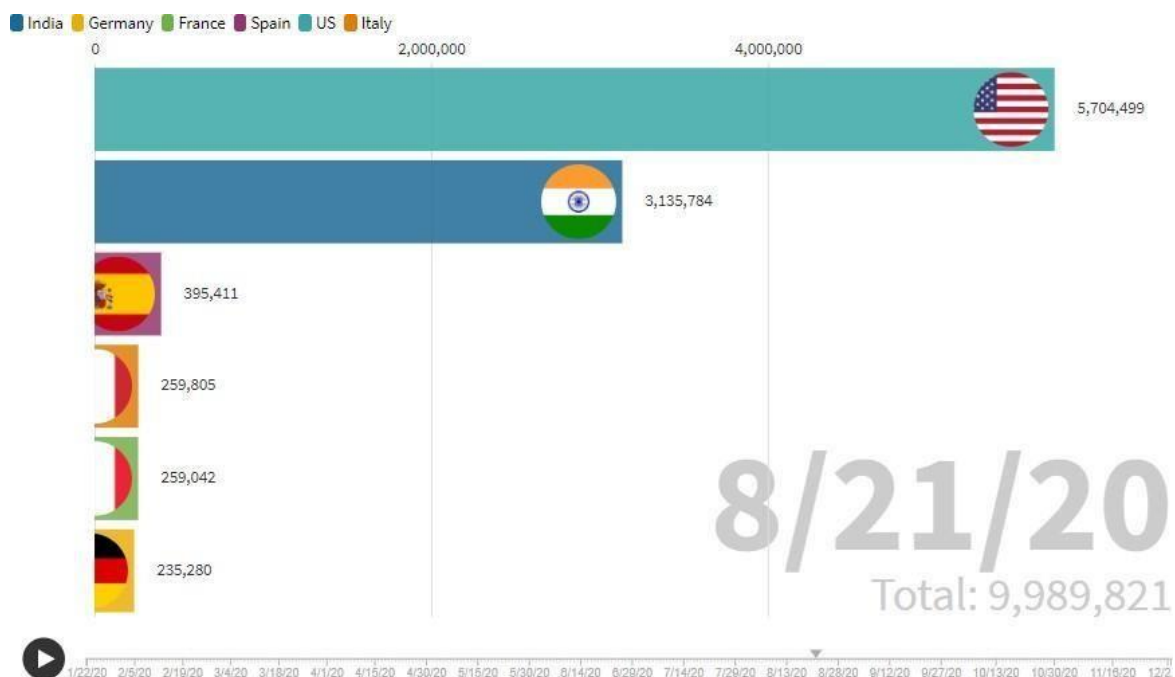
**Pie charts** are typically used to show percentage or proportional data. Usually, the percentage represented by each category is provided next to the corresponding slice

of the pie. matplotlib provides the pie method to make pie charts. It has several advantages:

1. It summarizes a large data set in visual form.

2. It displays the relative proportions of multiple classes of data.
3. The size of the circle is made proportional to the total quantity.

**Bar Chart Race** Create fascinating animated trend charts, with bars that race to the top based on ranks. The animated bar chart race helps you visualize the change in trends over time, these type of charts are very popular on social media as they provide a holistic data story/insight in a concise and easy to understand chart.



*Fig 4.3: Animated Graphical Representation*

To compare the cases to understand in a better view we used one of the best technique of bar chart race in which we provide data and the graph will be played as per the data provided in the dataset. Here we provide the data for comparing the providence as which one having the more or less cases can be sorted out here we compared among the six countries. Create fascinating animated trend charts, with bars that race to the top based on ranks. The animated bar chart race helps you visualize the change in trends over time, these type of charts are very popular on social media as they provide a holistic data story/insight in a concise and easy to understand chart.

## **CHAPTER 5**

### **RESULTS AND DISCUSSIONS**

#### **5.1 EXPERIMENT RESULTS:**

This chapter presents the results that are obtained from the experiment. The performance metric is utilized to evaluate the performance of the algorithms that were selected after the Literature Review. Three algorithms that were identified as the most suitable for the classification task to predict COVID-19 are:

- SVM (Support Vector Machine).
- Polynomial regression.
- Bayesian Network.

Each of the above stated algorithms were trained with the data-set that was collected and results were interpreted. Performance of each algorithm was evaluated at different stages of training set. Each algorithm was trained with the three different records sets containing 199+ country dataset records respectively. This experiment is performed to obtain which algorithm would be the most suitable for prediction of COVID-19. We could also assess which algorithm would perform better with prediction values. We also cross-checked with the original data in the Worldometer website which was giving us updated data of COVID cases. Worldometer is run by an international team of developers, researchers, and volunteers with the goal of making world statistics available in a thought-provoking and time relevant format to a wide audience around the world. It is published by a small and independent digital media company based in the United States. We have no political, governmental, or corporate affiliation. Furthermore, we have no investors, donors, grants, or backers of any type. We are completely independent and self-financed through automated programmatic advertising sold in real time on multiple ad exchanges. They collect and process data around the clock, 24 hours a day, 7 days a week. Multiple updates per minute are performed on average by our team of analysts and researchers who validate the data from an ever-growing list of over 5,000 sources under the constant

solicitation of users who alert us as soon as an official announcement is made anywhere around the world. There sources include Official Websites of Ministries of Health or other Government Institutions and Government authorities' social media accounts.

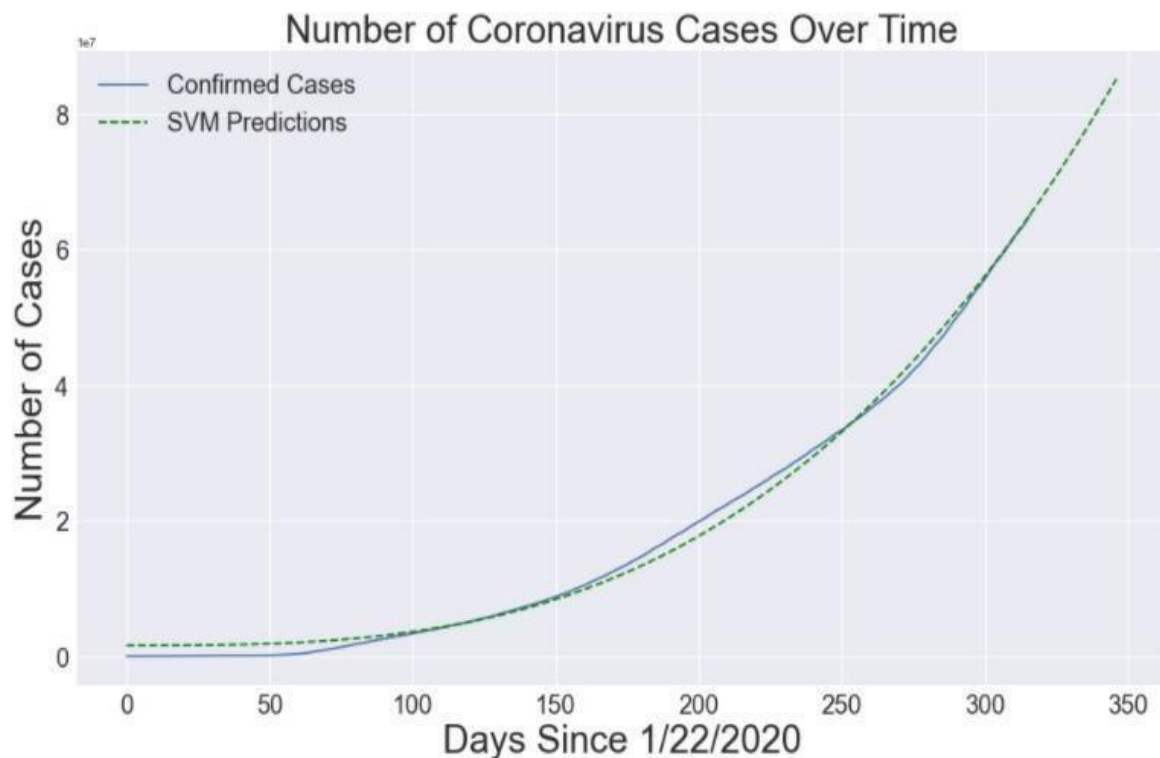
Because national aggregates often lag behind the regional and local health departments' data, part of work consists in monitoring thousands of daily reports released by local authorities. Their multilingual team also monitors press briefings' live streams throughout the day. Occasionally, they use a selection of leading and trusted news wires with a proven history of accuracy in communicating the data reported by Governments in live press conferences before it is published on the Official Websites. In 2020, the website gained popularity during the COVID-19 pandemic. Worldometer COVID-19 figures have also been cited by *Financial Times*, *The New York Times*, *The Washington Post*, *Fox News*, *CNN*, and *Rede Globo*. In 2020, the website attained greater popularity due to hosting statistics relating to the COVID-19 pandemic. We found the best accuracy of prediction of our proposed methods in which we found the SVM of 98%, polynomial of 89%, Bayesian Network of 75% accuracy we found from their predictions.

## **5.2 SUPPORT VECTOR MACHINE**

Support vector machine is the classifier that maximizes the margin. The goal of a classifier in our example below is to find a line or  $(n-1)$  dimension hyper-plane that separates the two classes present in the  $n$ -dimensional space. The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane. Supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labelled training data for each category, they're able to categorize new text. So you're working on a text classification problem. The support vector uses a mathematical function, often called a kernel function which is a math function that matches the new data to the best image from the training data in order to predict the unknown picture's label. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH). Support Vectors – Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with

the help of these data points. We trained the algorithm by providing the 320+ days data in order to get the best accurate values in prediction it was well trained by using the data provided in the three different

data sets and we found the accuracy from this approach of 98% accuracy from this model. And the below graph shows the accuracy of this method.



*Fig 5.1: Prediction using Support Vector Machine*

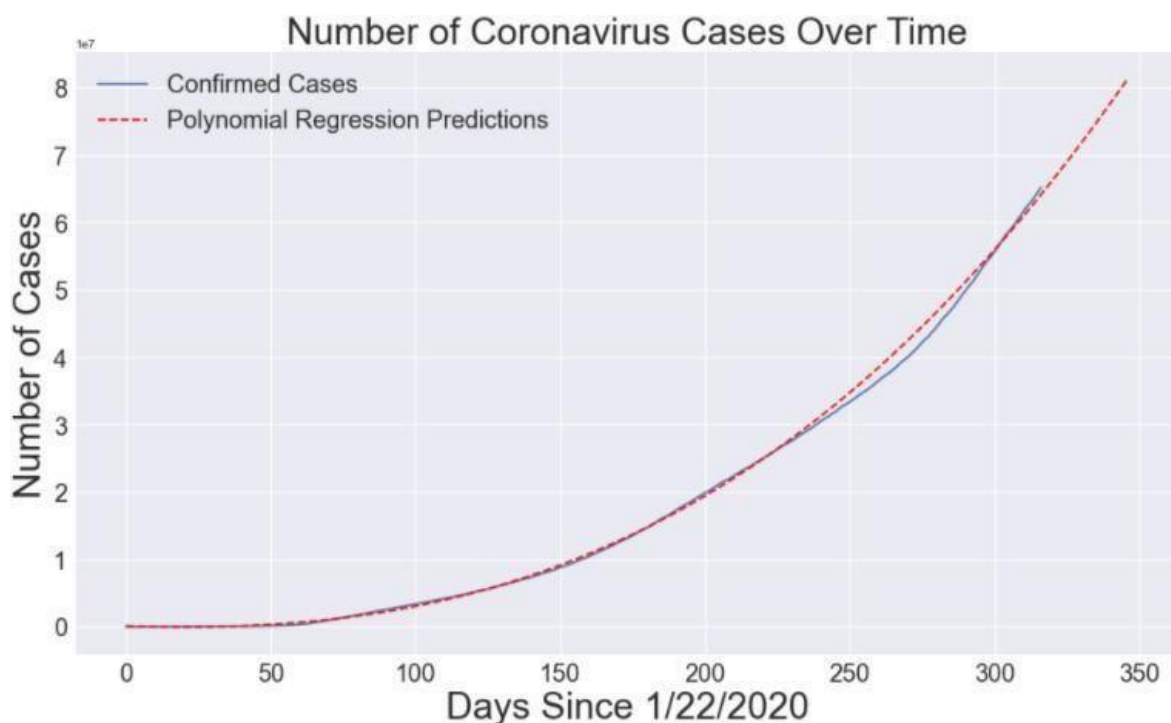
As we see in the above graph about the positive cases and the SVM predicted positive cases blue line represents the count of positive cases which we are given in the data set and the green line was SVM Predicted cases we can see the accuracy of our model by this graph. In the same way we also perform death cases and recovery cases using this approach.

### **5.3 POLYNOMIAL REGRESSION ALGORITHM RESULTS:**

In statistics, polynomial regression is a form of regression analysis in which the relationship between the independent variable  $x$  and the dependent variable  $y$  is modelled as an  $n$ th degree polynomial in  $x$ . The equation has the form  $Y = a + bX$ ,

where  $Y$  is the dependent variable (that's the variable that goes on the  $Y$  axis),  $X$  is the independent variable (i.e. it is plotted on the  $X$  axis),  $b$  is the slope of the line and

is the y-intercept. The Regression Equation is the algebraic expression of the regression lines. It is used to predict the values of the dependent variable from the given values of independent variables. The following algebraic equations can be solved simultaneously to obtain the values of parameter 'a' and 'b'. It is used in many experimental procedures to produce the outcome using this equation. It provides a great defined relationship between the independent and dependent variables. It is used to study the isotopes of the sediments. Polynomial provides the best approximation of the relationship between the dependent and independent variable. A Broad range of function can be fit under it. Polynomial basically fits a wide range of curvature. Regression analysis uses data, specifically two or more variables, to provide some idea of where future data points will be. The benefit of regression analysis is that this type of statistical calculation gives best way to see into the future cases. We trained the algorithm by providing the 320+ days data in order to get the best accurate values in prediction it was well trained by using the data provided in the three different data sets and we found the accuracy from this approach of 89% accuracy from this model. And the below graph shows the accuracy of this method.

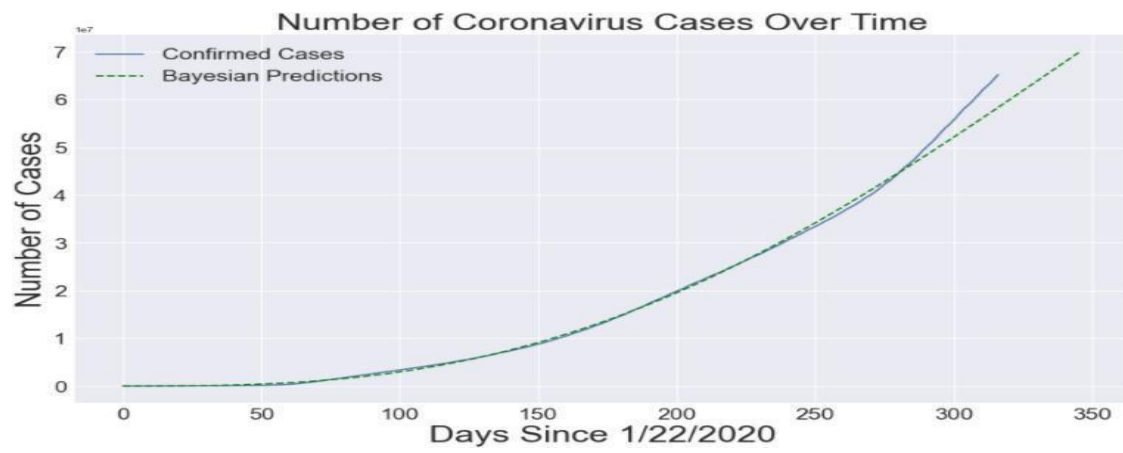


*Fig 5.2: Prediction using Polynomial Regression*

As we see in the above graph about the positive cases and the Polynomial predicted positive cases blue line represents the count of positive cases which we are given in the data set and the red line was Polynomial Predicted cases we can see the accuracy of our model by this graph. In the same way we also perform death cases and recovery cases using this approach.

#### **5.4 BAYESIAN RIDGE NETWORK ALGORITHM RESULTS:**

Bayesian regression allows a natural mechanism to survive insufficient data or poorly distributed data by formulating linear regression using probability distributors rather than point estimates. The output or response 'y' is assumed to drawn from a probability distribution rather than estimated as a single value. Bayesian analysis, a method of statistical inference that allows one to combine prior information about a population parameter with evidence from information contained in a sample to guide the statistical inference process. Bayesian statistics is a particular approach to applying probability to statistical problems. It provides us with mathematical tools to update our beliefs about random events in light of seeing new data or evidence about those events. In Bayesian analysis, a parameter is summarized by an entire distribution of values instead of one fixed value as in classical frequentist analysis. A posterior distribution comprises a prior distribution about a parameter and a likelihood model providing information about the parameter based on observed data. We trained the algorithm by providing the 320+ days data in order to get the best accurate values in prediction it was well trained by using the data provided in the three different data sets



*Fig 5.3: Prediction using Bayesian Network*

and we found the accuracy from this approach of 89% accuracy from this model. And the below graph shows the accuracy of this method.

As we see in the above graph about the positive cases and the Bayesian predicted positive cases blue line represents the count of positive cases which we are given in the data set and the green line was Bayesian Predicted cases we can see the accuracy of our model by this graph. In the same way we also perform death cases and recovery cases using this approach.

## 5.5 RESULTS COMPARISON:

Based on the experiments conducted, the overall accuracy results are tabulated for comparison in Table 5.5. A pictorial representation of performance of each algorithm at different datasets.

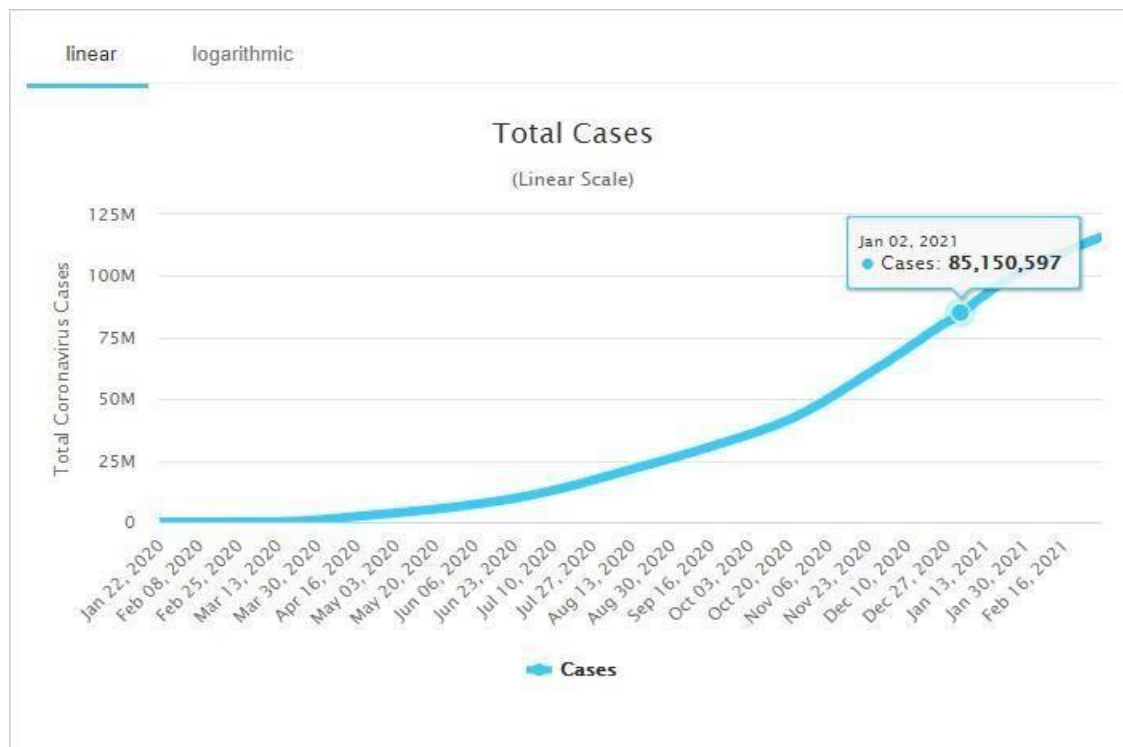
| Date Bayesian Ridge Predicted of Confirmed Cases Worldwide |            | Date PR Predicted number of Confirmed Cases Worldwide |            | Date SVM Predicted of Confirmed Cases Worldwide |            |
|------------------------------------------------------------|------------|-------------------------------------------------------|------------|-------------------------------------------------|------------|
| 0                                                          | 12-04-2020 | 0                                                     | 12-04-2020 | 0                                               | 12-04-2020 |
| 1                                                          | 12-05-2020 | 1                                                     | 12-05-2020 | 1                                               | 12-05-2020 |
| 2                                                          | 12-06-2020 | 2                                                     | 12-06-2020 | 2                                               | 12-06-2020 |
| 3                                                          | 12-07-2020 | 3                                                     | 12-07-2020 | 3                                               | 12-07-2020 |
| 4                                                          | 12-08-2020 | 4                                                     | 12-08-2020 | 4                                               | 12-08-2020 |
| 5                                                          | 12-09-2020 | 5                                                     | 12-09-2020 | 5                                               | 12-09-2020 |
| 6                                                          | 12-10-2020 | 6                                                     | 12-10-2020 | 6                                               | 12-10-2020 |
| 7                                                          | 12-11-2020 | 7                                                     | 12-11-2020 | 7                                               | 12-11-2020 |
| 8                                                          | 12-12-2020 | 8                                                     | 12-12-2020 | 8                                               | 12-12-2020 |
| 9                                                          | 12-13-2020 | 9                                                     | 12-13-2020 | 9                                               | 12-13-2020 |
| 10                                                         | 12-14-2020 | 10                                                    | 12-14-2020 | 10                                              | 12-14-2020 |
| 11                                                         | 12-15-2020 | 11                                                    | 12-15-2020 | 11                                              | 12-15-2020 |
| 12                                                         | 12-16-2020 | 12                                                    | 12-16-2020 | 12                                              | 12-16-2020 |
| 13                                                         | 12-17-2020 | 13                                                    | 12-17-2020 | 13                                              | 12-17-2020 |
| 14                                                         | 12-18-2020 | 14                                                    | 12-18-2020 | 14                                              | 12-18-2020 |
| 15                                                         | 12-19-2020 | 15                                                    | 12-19-2020 | 15                                              | 12-19-2020 |
| 16                                                         | 12-20-2020 | 16                                                    | 12-20-2020 | 16                                              | 12-20-2020 |
| 17                                                         | 12-21-2020 | 17                                                    | 12-21-2020 | 17                                              | 12-21-2020 |
| 18                                                         | 12-22-2020 | 18                                                    | 12-22-2020 | 18                                              | 12-22-2020 |
| 19                                                         | 12-23-2020 | 19                                                    | 12-23-2020 | 19                                              | 12-23-2020 |
| 20                                                         | 12-24-2020 | 20                                                    | 12-24-2020 | 20                                              | 12-24-2020 |
| 21                                                         | 12-25-2020 | 21                                                    | 12-25-2020 | 21                                              | 12-25-2020 |
| 22                                                         | 12-26-2020 | 22                                                    | 12-26-2020 | 22                                              | 12-26-2020 |
| 23                                                         | 12-27-2020 | 23                                                    | 12-27-2020 | 23                                              | 12-27-2020 |
| 24                                                         | 12-28-2020 | 24                                                    | 12-28-2020 | 24                                              | 12-28-2020 |
| 25                                                         | 12-29-2020 | 25                                                    | 12-29-2020 | 25                                              | 12-29-2020 |
| 26                                                         | 12-30-2020 | 26                                                    | 12-30-2020 | 26                                              | 12-30-2020 |
| 27                                                         | 12-31-2020 | 27                                                    | 12-31-2020 | 27                                              | 12-31-2020 |
| 28                                                         | 01-01-2021 | 28                                                    | 01-01-2021 | 28                                              | 01-01-2021 |
| 29                                                         | 01-02-2021 | 29                                                    | 01-02-2021 | 29                                              | 01-02-2021 |

Fig 5.4: Algorithm accuracy comparison

| S.No | Name of the algorithm               | Number of Positive cases predicted | Number of death cases predicted | Number of Recovery cases predicted |
|------|-------------------------------------|------------------------------------|---------------------------------|------------------------------------|
| 1.   | Support vector Machine algorithm    | 85,193,470                         | 1,878,861                       | 63,055,247                         |
| 2.   | Polynomial Regression algorithm     | 81,350,116                         | 1,565,460                       | 57,892,164                         |
| 3.   | Bayesian Ridge Regression algorithm | 70,171,668                         | 1,065,871                       | 55,981,782                         |

*Table 5.1: Accuracy Results*

So we can see here all the confirmed, deaths & recovery cases in separate columns are the predicted cases till January 2, 2021. Here we compared the data with the current obtained cases in the world on all three aspects from the worldometer. We compared them from that website and we found the data we got from this three models are the best approach models. We can see the below figures of confirmed cases, recovered cases, Death cases from worldometer.



*Fig 5.5: Total Positive Cases as of Jan 2<sup>nd</sup> 2021*

*Courtesy: Worldometer*

## Total Deaths

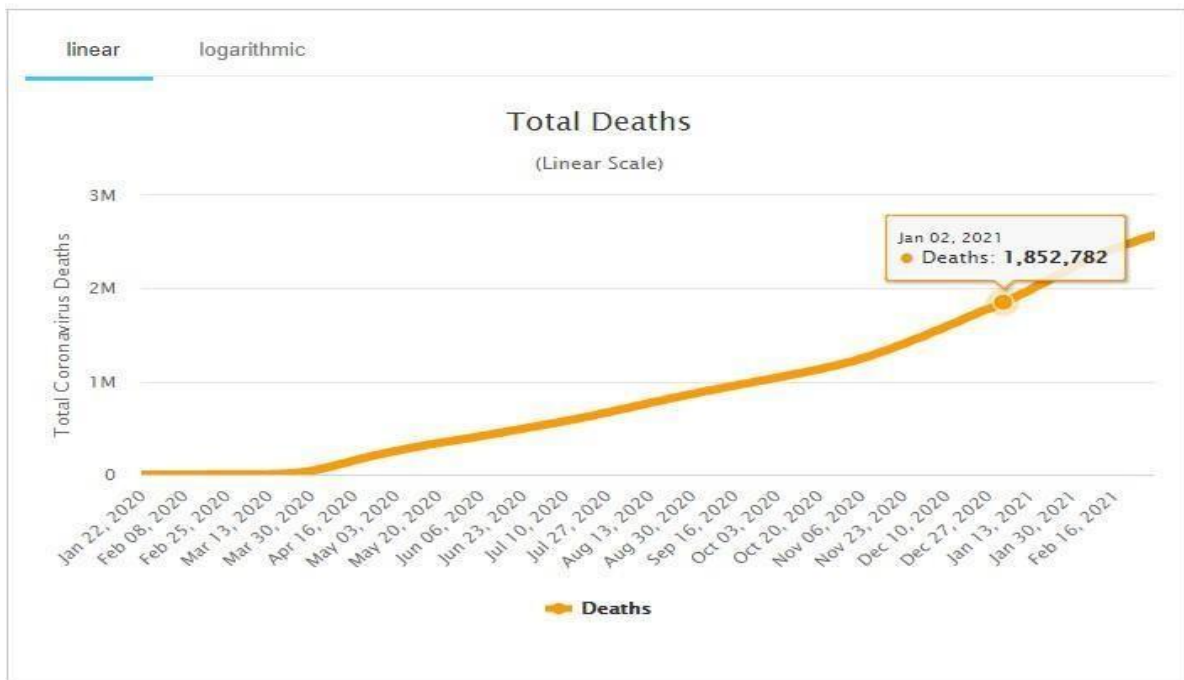


Fig 5.6: Total Death Cases as of Jan 2<sup>nd</sup> 2021

Courtesy: Worldometer

## Recovered and Discharged



Fig 5.7: Total Recovered Cases as of Jan 2<sup>nd</sup> 2021

Courtesy: Worldometer

Above we can check the prediction of our approaches with the real time data so we can find how much efficiency we are getting can be easily traced we took the data from JAN 22, 2020 to DEC 3, 2020. And We predicted 30 days so that means up to JAN 2, 2021. And we got the data from the worldometer which shows our approaches gave the accurate prediction in all confirmed, Death, recovery cases as well. This work presents a model for prediction of COVID-19 mortality that was developed using a hybrid approach, by calibrating the predictions from a baseline model designed to rank the population according to the risk for severe respiratory infections or sepsis to report COVID-19 mortality rates. The resulting model, developed rapidly under conditions of uncertainty, was used to evaluate the risk for COVID-19 mortality for the entire population.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE**

#### **WORK**

In this Project, a systematic literature review has been conducted to identify the suitable algorithm for prediction of COVID-19 in patients. There was no pure evidence found to summarize one algorithm as the suitable technique for prediction. Hence, a set of algorithms which include Support Vector Machine (SVM), Polynomial Regression, Bayesian Ridge were chosen. The selected algorithms were trained with the datasets.

To evaluate the accuracy of machine learning models, each algorithm is trained with record sets of varying number of cases. Using accuracy performance metric, the trained algorithms were assessed. After result analysis, Support Vector Machine (SVM) showed better prediction accuracy in comparison with both Polynomial Regression, Bayesian Ridge. The trained algorithms were also assessed to find the features that affect the prediction of COVID-19 in patients. There is a lot of scope for Machine Learning in Healthcare.

For Future work, it is recommended to work on calibrated and ensemble methods that could resolve quirky problems faster with better outcomes than the existing algorithms. Also an AI-based application can be developed using various sensors and features to identify and help diagnose diseases.

As healthcare prediction is an essential field for future, a prediction system that could find the possibility of outbreak of novel diseases that could harm mankind through socio-economic and cultural factor consideration can be developed.

## REFERENCES

- [1] <https://respiratoryresearch.biomedcentral.com/articles/10.1186/s12931-02001479-w>
- [2] <https://www.scientificamerican.com/article/how-coronaviruses-cause-infection-from-colds-to-deadly-pneumonia1/>
- [3] <https://www.who.int/vietnam/news/detail/14-07-2020-q-a-how-is-covid-19-transmitted>
- [4] <https://www.worldometers.info/coronavirus/#countries>
- [5] <https://www.javatpoint.com/machinelearning-polynomial-regression>
- [6] <https://www.analyticsvidhya.com/blog/2017/09/>
- [7] Tan W., Zhao X., Ma X., Wang W., Niu P., Xu W., et al. A novel coronavirus genome identified in a cluster of pneumonia cases—Wuhan, China 2019–2020. *China CDC Weekly* 2020; 2(4):61-62.
- [8] Awad M., Khanna R.. Support vector regression. *Efficient learning machines*. 2015
- [9] WHO EMRO | Questions and answers | COVID-19 | Health topics.
- [10] Support Vector Machine Machine learning algorithm with example and code, January 2019. Library Catalog: [www.codershood.info](http://www.codershood.info) Section: Machine learning.
- [11] Ali Al-Hazmi. Challenges presented by MERS corona virus, and SARS corona virus to global health. *Saudi journal of biological sciences*, 23(4):507–511, 2016. Publisher: Elsevier.
- [12] Sina F Ardabili, Amir Mosavi, Pedram Ghamisi, Filip Ferdinand, Annamaria R Varkonyi-Koczy, Uwe Reuter, Timon Rabczuk, and Peter M Atkinson. Covid-19 outbreak prediction with machine learning. Available at SSRN 3580188, 2020.
- [13] David W Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, 33(7):1123–1131, 2014.
- [14] Hetal Bhavsar and Amit Ganatra. A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(4):2231–2307, 2012.
- [15] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of

supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning, pages 161–168, 2006.

- [16] Nanshan Chen, Min Zhou, Xuan Dong, Jieming Qu, Fengyun Gong, Yang Han, Yang Qiu, Jingli Wang, Ying Liu, Yuan Wei, et al. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study. *The Lancet*, 395(10223):507–513, 2020.
- [17] Dursun Delen. Analysis of cancer data: a data mining approach. *Expert Systems*, 26(1):100–112, 2009.
- [18] Manoj Durairaj and Veera Ranjani. Data mining applications in healthcare sector: a study. *International journal of scientific & technology research*, 2(10):29– 35, 2013.
- [19] Ashok Kumar Dwivedi. Performance evaluation of different machine learning techniques for prediction of heart disease. *Neural Computing and Applications*, 29(10):685–693, 2018.
- [20] Arihito Endo, Takeo Shibata, and Hiroshi Tanaka. Comparison of seven algorithms to predict breast cancer survival (< special issue> contribution to 21 century intelligent technologies and bioinformatics). *International Journal of Biomedical Soft Computing and Human Sciences: the official journal of the Biomedical Fuzzy Systems Association*, 13(2):11–16, 2008.
- [21] Ya-Han Hu, Yi-Lien Lee, Ming-Feng Kang, and Pei-Ju Lee. Constructing inpatient pressure injury prediction models using machine learning techniques. *Computers, Informatics, Nursing: CIN*, 2020.
- [22] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [23] G. Kesavaraj and S. Sukumaran. A study on classification techniques in data mining. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–7, 2013.
- [24] Barbara Kitchenham. Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004):1–26, 2004.
- [25] Halgurd S. Maghdid, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, Kevin Curran, and Khaled Rabie. A novel ai-enabled framework to diagnose coronavirus covid 19 using smartphone embedded sensors: Design study. *arXiv preprint arXiv:2003.07434*, 2020.

# PLAGIARISM REPORT

Paper Presentation.doc

## ORIGINALITY REPORT

13%

SIMILARITY INDEX

1%

INTERNET SOURCES

10%

PUBLICATIONS

2%

STUDENT PAPERS

## PRIMARY SOURCES

1

Milind Yadav, Murukessan Perumal, M Srinivas.  
"Analysis on novel coronavirus (COVID-19)  
using machine learning methods", Chaos,  
Solitons & Fractals, 2020

Publication

9%

2

Submitted to A'Sharqiyah University, Oman

Student Paper

2%

3

[www.hindawi.com](http://www.hindawi.com)

Internet Source

1%

4

[globalizationandhealth.biomedcentral.com](http://globalizationandhealth.biomedcentral.com)

Internet Source

1%

Exclude quotes On

Exclude bibliography On

Exclude matches Off

# Analysis & Prediction of Covid-19 using Supervised Machine Learning

**Velagala TejaReddy<sup>1</sup>, T. Kondala Rao Naidu<sup>2</sup>, Mr. Murari Kamalesh<sup>3</sup>;**

Students<sup>1,2</sup>, Sathyabama Institute of Science and Technology, Chennai, India

Assistant Professor<sup>3</sup>, Department of Computer Science & Engineering, SIST, Chennai,

India

**ABSTRACT---** we are going to perform prediction of novel corona Virus. It is a spreadable disease caused by a newly discovered virus (corona), It is infested mainly to people through direct, Indirect, or through infested secretions such as spit. This virus causes infection in human beings and has killed many individuals in across the world. However, virus is stated as contagion by the (WHO) - World Health Organization and all nation-states are fighting to prevent this with lock-down everywhere. Here, main intent of this work are four altered tasks such as 1) providing the datasets of confirmed, Death, Recovery cases to the machine. 2) Virus transmission rate analysis .3) Train the Machine with the datasets provided. 4) Forecasting the spread of coronavirus. The benefit of implementation these tasks to reduce the virus range by various alleviation. In this paper, we propose supervised machine learning methods to execute four different tasks associated to novel corona virus. This approach calculation is then equated among three Supervised Machine Learning algorithms on standard available datasets. The results will be signify its excellence in both effectiveness and precision.

## **KEYWORDS---**

COVID-19, Machine Learning, Prediction, Supervised machine Learning, Polynomial Regression, Support Vector Machine.

## **1.INTRODUCTION**

Corona virus is a massive structural viruses which causes sickness differing from the gentle virus to more extreme infections like Middle East Respiratory Syndrome and Severe Acute Respiratory Syndrome. Conferring to Present data says that spread of virus occurs easily between individuals through immediate, backhanded, or close contact with infection pervaded individuals through contaminated discharges like salivation and respiratory emissions, or through their respiratory drops, which are shot out when a tainted individual wheezes, sniffles, communicate. The tale Covid can support on surface like hardened steel and plastic a few days and on cardboard, and copper not many hours the infection will sustain. Corona virus distresses various individuals alteredly. For the most part contaminated individuals would extend gentle to direct infection and gets recuperated without hospitalization. The Most regularly seen indications are fever, dry hack, also drowsiness and some commonly seen symptoms are niggles furthermore, torments, sore throat, loose bowels, conjunctivitis, cerebral pain, loss of taste or smell. The Severe side effects are trouble breathing, chest agony or pressing factor loss of discourse or movement. Its better Seek immediate medical consideration if you have any of

the beyond given symptoms. Individuals with light manifestations who are usually solid ought to deal with their side effects at house. On normal it will take

5–6 days from when somebody is swarmed with the infection to get side effects, anyway it would require as long as 14 days to get better. Currently we got the medicine to overcome the virus but the more over yet the trails are going on to find the medicine to make available to common man in reasonable Prices. Till today COVID-19 has been contaminated the people of in excess of 177 nations prompts patients out of which 112 million people are infected with virus among them 2,485,386 lost their lives and 87,798,712 people have been recovered till now.

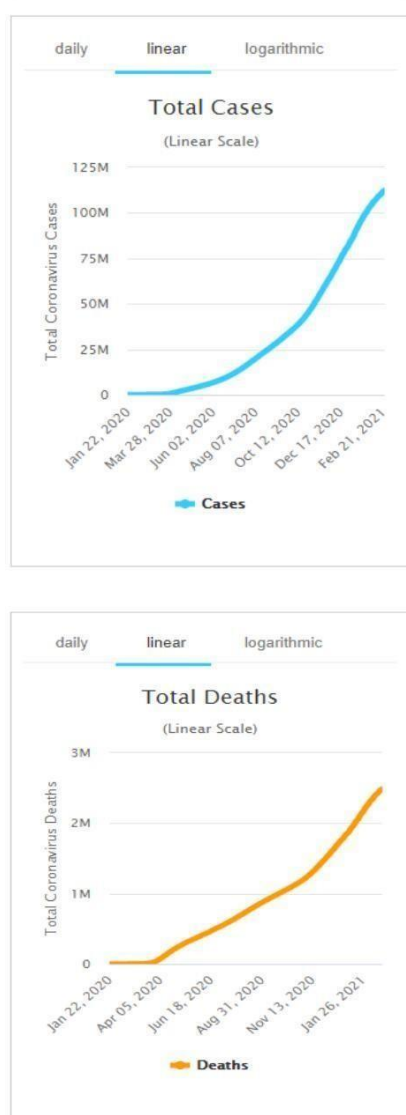


Fig.1 Covid cases in linear scale presentation published by worldometer.

Both the above figures represents the COVID confirmed cases and death cases across the world. Mainly Due to the insufficiency of COVID test packs, ventilators, oxygen chambers, medical clinic beds, and inaccessibility of legitimate therapy, it is fundamental to break down the development paces of positive cases, number of recuperations. So then some proper preparations can be prepared to overcome victims of lives. For instance, in light of the examination of these information the public authority can have the legitimate data to the quantity of cases till a specific day, and before that day they can organize all the fundamental clinical gear to forestall misfortunes of lives. At present AI strategies are playing crucial role and being generally utilized in medical services area and for having more proficient expectation of COVID-19. In this work, the Polynomial Regression model is utilized to tackle the four different errands of COVID-19 related issues. This strategy will be fitted into the datasets comprises of the absolute number of COVID19 affirmed cases, passings cases and the quantity of recuperations for some, nations like India, US, china, UK, Canada. And by using this proposed method we can forecast the future number of absolute cases. These undertaking can help a country state to assess the spread of the infection, alert individuals, and start upgrades. It'll likewise assist that country with being ready for the future to overcome the pandemic situation, which may help to save lives. On proposed technique results with other understand relapse models, for example, support vector machine algorithm, Polynomial Regression, Bayesian Network. Comparing those three machine learning algorithms and Prediction can be done. Along with the prediction also performed with the visualisation in it. So that we can

identify the total positive cases along with deaths, recoveries and active cases can be easily traced out through the visualization technique.

**2. APPLIED METHODS---** This section will be describes about the motive and the complete overview of the tasks and methods that are being approached. So to overcome the effects of COVID, a few undertakings can be performed. These errands can help with approximating and concentrate information from COVID19 information, which can help a country state to perceive the banquet of the infection, to make wakefulness in people. It would definitely help that nation to get prepared for future pandemic circumstance that may help in saving lives. In this paper, significantly four different undertakings that were performed and they are as per the following 1)Giving the datasets of affirmed, passings, recuperation cases. 2)Analyzing the gala pace of the infection. 3)Train the Machine with the datasets. 4)Predicting the spread of coronavirus. In our proposed work Support Vector Regression model, Polynomial regression and Bayesian network are used to work on those four tasks.

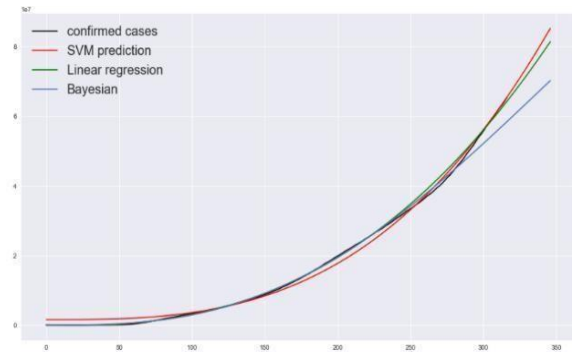
**Polynomial Regression Algorithm ---** Polynomial Regression On account of we add some polynomial focuses to the Multiple Linear relapse estimation to adjust it into Polynomial Regression. It is a direct model with not many changes to rise the exactness. The datasets use in Polynomial relapse for preparing are non-direct nature. It upholds straight relapse ideal to fit the problematical and non-direct capacities and datasets. The need of Polynomial Regression in Machine Learning is for following reasons .If we put on a direct model on a direct dataset, at that point it passes on us with great outcome as we have found in Simple Linear Regression, however on the off chance that we relate the model for what it's worth with no modification on a non- direct dataset, may create divert yield. Because of which misfortune capacities will expand, the blunder rate will turn out to be high, and precision will be diminished.

For such cases information focuses ought to be coordinated in a non-direct style, we require the Polynomial Regression model.

**Support Vector Machine Algorithm (SVM)** SVM is a supervised Machine Learning algorithm which can be used for classification and regression models. Though, mostly we use this to resolve classification problems. Then, we perform organization finding the hyper-plane that differs the two modules very well.

**Results for Experiment ---** We have looked at the exhibition aftereffects of our proposed technique with the extra notable algorithms Polynomial Regression & Bayesian network. The dataset utilized for the four assignments comprises of the all out number of positive cases, recuperations, passings for above 325 days ( from 22/01/2020 to 04/12/2020) in various nations and locales, Country/Region name, and date. Then again, for each of the four assignment dates, scope, longitude area name and absolute positive cases information was utilized. Providing the datasets. Data have to be collected and stored in the excel sheet by classifying with date, country name along with that latitude and longitude have to be provided and In the same format we have to collect the all three files on confirmed cases, death cases, recovery cases have to be declared properly. Analyzing the transmission rate of the virus. Proper analysis have to be done on the datasets and visualization will done as shown below.

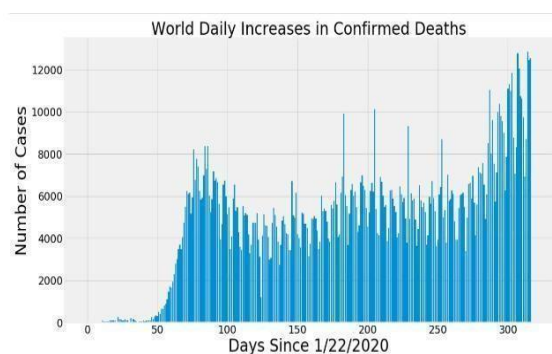
Train the Machine with the datasets  
providing the datasets to train the machine



to function in accordingly visualization can be done .To classify in various forms as to present that so one can find the number of cases in their region so easily. As shown in the given figure visualization can be done. To foresee the result to configuration model to anticipate. In the event that we utilize directed learning or, in all likelihood some cross breed that has incorporates that approach, information are going to enrich with data labelling or annotation. Predicting the spread of coronavirus—we used three algorithms SVM, Bayesian Network and polynomial techniques to predict future upcoming COVID cases in the world.



To represent the COVID cases we used the best visualization technique bar chart race to show the public in detail of cases among the countries in comparison. And the following below is the predicted positive cases using support vector machine algorithm.



|    | Date       | SVM Predicted of Confirmed Cases Worldwide |
|----|------------|--------------------------------------------|
| 0  | 12-04-2020 | 65883463.0                                 |
| 1  | 12-05-2020 | 66493882.0                                 |
| 2  | 12-06-2020 | 67108153.0                                 |
| 3  | 12-07-2020 | 67726267.0                                 |
| 4  | 12-08-2020 | 68348296.0                                 |
| 5  | 12-09-2020 | 68974193.0                                 |
| 6  | 12-10-2020 | 69603969.0                                 |
| 7  | 12-11-2020 | 70237698.0                                 |
| 8  | 12-12-2020 | 70875330.0                                 |
| 9  | 12-13-2020 | 71516898.0                                 |
| 10 | 12-14-2020 | 72162414.0                                 |
| 11 | 12-15-2020 | 72811891.0                                 |
| 12 | 12-16-2020 | 73465340.0                                 |
| 13 | 12-17-2020 | 74122773.0                                 |
| 14 | 12-18-2020 | 74784203.0                                 |
| 15 | 12-19-2020 | 75449641.0                                 |
| 16 | 12-20-2020 | 76119100.0                                 |
| 17 | 12-21-2020 | 76792592.0                                 |
| 18 | 12-22-2020 | 77470129.0                                 |
| 19 | 12-23-2020 | 78151724.0                                 |
| 20 | 12-24-2020 | 78837367.0                                 |
| 21 | 12-25-2020 | 79527132.0                                 |
| 22 | 12-26-2020 | 80220970.0                                 |
| 23 | 12-27-2020 | 80918913.0                                 |
| 24 | 12-28-2020 | 81620975.0                                 |
| 25 | 12-29-2020 | 82327166.0                                 |
| 26 | 12-30-2020 | 83037499.0                                 |
| 27 | 12-31-2020 | 83751966.0                                 |
| 28 | 01-01-2021 | 84470639.0                                 |
| 29 | 01-02-2021 | 85193470.0                                 |

**Conclusion**—COVID-19 effects human beings and throws to sick-ness and also severe harm will be created in the lungs. COVID-19 gave a huge loss to many people in the entire world. A huge number of people have lost their lives. We recommend the Support Vector Machine procedure founded novel virus analysis on four altered tasks. We also used regression methods & Bayesian network to get better classification accuracy. The benefit of the errand is that, it gives a thought on pace of level of spread, and in as per that, the public authority and people in general can make fitting arrangements to switch this pandemic situation by taking measures to diminish the infection feast by different

alleviation furthermore, other vital activities. The actions that are taken till date how professionally they would consequence. So, by the prediction people will be able to get alertness of this pandemic situation. And Proper data which provides number of cases till a certain date can be used by the Government can organize all the essential medical utensils to avoid losses of lives in advance.

## References

[1] <https://respiratoryresearch.biomedcentral.com/articles/10.1186/s12931-020-01479-w>

[2] <https://www.scientificamerican.com/article/how-coronaviruses-cause-infection-from-colds-to-deadly-pneumonia1/>

[3] <https://www.who.int/vietnam/news/detail/14-07-2020-q-a-how-is-covid-19-transmitted>

[4] <https://www.worldometers.info/coronavirus/#countries>

[5] <https://www.javatpoint.com/machine-learning-polynomial-regression>

[6] <https://www.analyticsvidhya.com/blog/2017/09/>

[7] Tan W., Ma X., Zhao X., Wang W., Xu W., Niu P., et al. A novel coronavirus genome identified in a cluster of pneumonia cases—Wuhan, China 2019–2020. *China CDC Weekly* 2020; 2(4):61-62.

[8] Awad M., Khanna R.. Support vector regression. *Efficient learning machines*. 2015

[9] WHO EMRO | Questions and answers | COVID-19 | Health topics.

[10] Support Vector Machine Machine learning algorithm with example and code,

January 2019. Library Catalog: [www.codershood.info](http://www.codershood.info) Section: Machine learning.

Ali Al-Hazmi. Challenges presented by MERS corona virus, and SARS corona virus to global health. *Saudi journal of biological sciences*, 23(4):507–511, 2016. Publisher: Elsevier.

## SOURCE CODE

```
#importing Libraries
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import pandas as pd
import random
import math
import time
import xgboost

from sklearn.linear_model import LinearRegression, BayesianRidge
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
from IPython.core.display import HTML

import datetime
import operator
plt.style.use('seaborn')
%matplotlib inline

#Reading Confirmed cases file
confirmed_cases = pd.read_csv('time_series_covid19_confirmed_global.csv')
confirmed_cases.head()

#Reading Death cases file
deaths_reported = pd.read_csv('time_series_covid19_deaths_global.csv')
deaths_reported.head()

#Reading recovered cases file
recovered_cases = pd.read_csv('time_series_covid19_recovered_global.csv')
recovered_cases.head()

latest_data = pd.read_csv('csse_daily.csv')
latest_data.head()

# Fetching all the columns from confirmed dataset
cols = confirmed_cases.keys()
```

```
cols
# Extracting the date columns
confirmed = confirmed_cases.loc[:, cols[4]:cols[-1]]
deaths = deaths_reported.loc[:, cols[4]:cols[-1]]
recoveries = recovered_cases.loc[:, cols[4]:cols[-1]]

confirmed
dates = confirmed.keys()

world_cases = []
total_deaths = []
mortality_rate = []
recovery_rate = []
total_recovered = []
total_active = []

china_cases = []
italy_cases = []
us_cases = []
spain_cases = []
france_cases = []
germany_cases = []
uk_cases = []
russia_cases = []
india_cases = []

china_deaths = []
italy_deaths = []
us_deaths = []
spain_deaths = []
france_deaths = []
germany_deaths = []
uk_deaths = []
russia_deaths = []
india_deaths = []
```

```

china_recoveries = []
italy_recoveries = []
us_recoveries = []
spain_recoveries = []
france_recoveries = []
germany_recoveries = []
uk_recoveries = []
russia_recoveries = []
india_recoveries = []

dates
for i in dates:
    confirmed_sum = confirmed[i].sum()
    death_sum = deaths[i].sum()
    recovered_sum = recoveries[i].sum()

    world_cases.append(confirmed_sum)
    total_deaths.append(death_sum)
    total_recovered.append(recovered_sum)
    total_active.append(confirmed_sum-death_sum-
        recovered_sum)

    mortality_rate.append(death_sum/confirmed_sum)
    recovery_rate.append(recovered_sum/confirmed_sum)

china_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='China'][i].sum())

italy_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Italy'][i].sum())

us_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='US'][i].sum())

spain_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Spain'][i].sum())

france_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='France'][i].sum())

```

```
germany_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Germany'][i].sum())
```

```
uk_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='United Kingdom'][i].sum())
```

```
russia_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Russia'][i].sum())
```

```
india_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='India'][i].sum())
```

```
china_deaths.append(deaths_reported[deaths_reported['Country/Region']=='China'][i].sum())
```

```
italy_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Italy'][i].sum())
```

```
us_deaths.append(deaths_reported[deaths_reported['Country/Region']=='US'][i].sum())
```

```
spain_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Spain'][i].sum())
```

```
france_deaths.append(deaths_reported[deaths_reported['Country/Region']=='France'][i].sum())
```

```
germany_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Germany'][i].sum())
```

```
uk_deaths.append(deaths_reported[deaths_reported['Country/Region']=='United Kingdom'][i].sum())
```

```
russia_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Russia'][i].sum())
```

```
india_deaths.append(deaths_reported[deaths_reported['Country/Region']=='India'][i].sum())
```

```
china_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='China'][i].sum())
```

```

italy_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='
'Italy'][i].sum())

us_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='US
'][i].sum())

spain_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='
'Spain'][i].sum())

france_recoveries.append(recovered_cases[recovered_cases['Country/Region']=
'France'][i].sum())

germany_recoveries.append(recovered_cases[recovered_cases['Country/Region']
=='Germany'][i].sum())

uk_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='Un
ited Kingdom'][i].sum())

russia_recoveries.append(recovered_cases[recovered_cases['Country/Region']=
'Russia'][i].sum())

india_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='
'India'][i].sum())

world_cases
total_deaths
confirmed_sum
death_sum
recovered_sum
india_cases
italy_recoveries
def daily_increase(data):
    d = []
    for i in range(len(data)):
        if i == 0:
            d.append(data[0])
        else:
            d.append(data[i]-data[i-1])
    return d
# confirmed cases
world_daily_increase = daily_increase(world_cases)
china_daily_increase = daily_increase(china_cases)

```

```

italy_daily_increase = daily_increase(italy_cases)
us_daily_increase = daily_increase(us_cases)
spain_daily_increase = daily_increase(spain_cases)
france_daily_increase = daily_increase(france_cases)
germany_daily_increase = daily_increase(germany_cases)
uk_daily_increase = daily_increase(uk_cases)
india_daily_increase = daily_increase(india_cases)
india_daily_increase

# deaths
world_daily_death = daily_increase(total_deaths)
china_daily_death = daily_increase(china_deaths)
italy_daily_death = daily_increase(italy_deaths)
us_daily_death = daily_increase(us_deaths)
spain_daily_death = daily_increase(spain_deaths)
france_daily_death = daily_increase(france_deaths)
germany_daily_death = daily_increase(germany_deaths)
uk_daily_death = daily_increase(uk_deaths)
india_daily_death = daily_increase(india_deaths)
india_daily_death

# recoveries
world_daily_recovery = daily_increase(total_recovered)
china_daily_recovery = daily_increase(china_recoveries)
italy_daily_recovery = daily_increase(italy_recoveries)
us_daily_recovery = daily_increase(us_recoveries)
spain_daily_recovery = daily_increase(spain_recoveries)
france_daily_recovery = daily_increase(france_recoveries)
germany_daily_recovery = daily_increase(germany_recoveries)
uk_daily_recovery = daily_increase(uk_recoveries)
india_daily_recovery = daily_increase(india_recoveries)
india_daily_recovery
world_daily_recovery

unique_countries = list(latest_data['Country_Region'].unique())
unique_countries

country_confirmed_cases = []
country_death_cases = []

```

```

country_active_cases = []
country_recovery_cases = []
country_mortality_rate = []

no_cases = []
for i in unique_countries:
    cases =
latest_data[latest_data['Country_Region']==i]['Confirmed'].sum()
    if cases > 0:
        country_confirmed_cases.append(cases)
    else:
        no_cases.append(i)

for i in no_cases:
    unique_countries.remove(i)

# sort countries by the number of confirmed cases
unique_countries = [k for k, v in
sorted(zip(unique_countries,
country_confirmed_cases), key=operator.itemgetter(1), reverse=True)]

for i in range(len(unique_countries)):
    country_confirmed_cases[i] =
latest_data[latest_data['Country_Region']==unique_countries[i]]['Confirmed'
].sum()

country_death_cases.append(latest_data[latest_data['Country_Region']==unique
_countries[i]]['Deaths'].sum())

country_recovery_cases.append(latest_data[latest_data['Country_Region']==un
ique_countries[i]]['Recovered'].sum())

    country_active_cases.append(country_confirmed_cases[i] -
country_death_cases[i] - country_recovery_cases[i])

country_mortality_rate.append(country_death_cases[i]/country_confirmed_case
s[i])

country_df = pd.DataFrame({'Country Name': unique_countries, 'Number of
Confirmed Cases': country_confirmed_cases,
                            'Number of Deaths': country_death_cases, 'Number
of Recoveries' : country_recovery_cases,
                            'Number of Active Cases' : country_active_cases,
                            'Mortality Rate': country_mortality_rate})

# number of cases per country/region

```

```

country_df.style.background_gradient(cmap='Blues')

unique_provinces = list(latest_data['Province_State'].unique())

province_confirmed_cases = []
province_country = []
province_death_cases = []
province_recovery_cases = []
province_mortality_rate = []

no_cases = []
for i in unique_provinces:
    cases =
latest_data[latest_data['Province_State']==i]['Confirmed'].sum()
    if cases > 0:
        province_confirmed_cases.append(cases)
    else:
        no_cases.append(i)

# remove areas with no confirmed cases
for i in no_cases:
    unique_provinces.remove(i)

unique_provinces = [k for k, v in sorted(zip(unique_provinces,
province_confirmed_cases), key=operator.itemgetter(1), reverse=True)]
for i in range(len(unique_provinces)):
    province_confirmed_cases[i] =
latest_data[latest_data['Province_State']==unique_provinces[i]]['Confirmed'
].sum()

province_country.append(latest_data[latest_data['Province_State']==unique_p
rovinces[i]]['Country_Region'].unique()[0])

province_death_cases.append(latest_data[latest_data['Province_State']==uniqu
e_provinces[i]]['Deaths'].sum())

province_recovery_cases.append(latest_data[latest_data['Province_State']==u
nique_provinces[i]]['Recovered'].sum())

province_mortality_rate.append(province_death_cases[i]/province_confirmed_c
ases[i])

```

```

# number of cases per province/state/city
province_df = pd.DataFrame({'Province/State Name': unique_provinces,
'Country': province_country, 'Number of Confirmed Cases':
province_confirmed_cases,
                                'Number of Deaths': province_death_cases, 'Number
of Recoveries' : province_recovery_cases,
                                'Mortality Rate': province_mortality_rate})

# number of cases per country/region

province_df.style.background_gradient(cmap='Reds')

# Dealing with missing values
nan_indices = []

# handle nan if there is any, it is usually a float: float('nan')

for i in range(len(unique_provinces)):
    if type(unique_provinces[i]) == float:
        nan_indices.append(i)

unique_provinces = list(unique_provinces)
province_confirmed_cases = list(province_confirmed_cases)

for i in nan_indices:
    unique_provinces.pop(i)
    province_confirmed_cases.pop(i)

India_confirmed =
latest_data[latest_data['Country_Region']=='India']['Confirmed'].sum()
outside_India_confirmed = np.sum(country_confirmed_cases) - India_confirmed
plt.figure(figsize=(16, 9))
plt.barh('India', India_confirmed)
plt.barh('Outside India', outside_India_confirmed)
plt.title('Number of Coronavirus Confirmed Cases', size=20)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

print('Outside India {} cases:'.format(outside_India_confirmed))
print('India: {} cases'.format(India_confirmed))

```

```

print('Total: {} cases'.format(India_confirmed+outside_India_confirmed))

# Only show 10 countries with the most confirmed cases, the rest are
grouped into the other category
visual_unique_countries = []
visual_confirmed_cases = []
others = np.sum(country_confirmed_cases[10:])

for i in range(len(country_confirmed_cases[:10])):
    visual_unique_countries.append(unique_countries[i])
    visual_confirmed_cases.append(country_confirmed_cases[i])

visual_unique_countries.append('Others')
) visual_confirmed_cases.append(others)
def plot_bar_graphs(x, y, title):
    plt.figure(figsize=(16, 9))
    plt.barh(x, y)
    plt.title(title, size=20)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

plot_bar_graphs(visual_unique_countries, visual_confirmed_cases, 'Number
of Covid-19 Confirmed Cases in Countries/Regions [1 unit=50lakh_cases]')
def plot_pie_charts(x, y, title):
    c = random.choices(list(mcolors.CSS4_COLORS.values()), k =
len(unique_countries))
    plt.figure(figsize=(20,15))
    plt.title(title, size=20)
    plt.pie(y, colors=c)
    plt.legend(x, loc='best', fontsize=15)
    plt.show()

plot_pie_charts(visual_unique_countries, visual_confirmed_cases, 'Covid-19
Confirmed Cases per Country')

# Only show 10 provinces with the most confirmed cases, the rest are
grouped into the others category
visual_unique_provinces = []
visual_confirmed_cases2 = []
others = np.sum(province_confirmed_cases[10:])

```

```

for i in range(len(province_confirmed_cases[:10])):
    visual_unique_provinces.append(unique_provinces[i])
    visual_confirmed_cases2.append(province_confirmed_cases[i])

visual_unique_provinces.append('Others')
visual_confirmed_cases2.append(others)

plot_bar_graphs(visual_unique_provinces, visual_confirmed_cases2, 'Number
of Coronavirus Confirmed Cases in Provinces/States[1 unit=50 lakh
cases]')
def plot_pie_country_with_regions(country_name, title):
    regions =
list(latest_data[latest_data['Country_Region']==country_name]['Province_Sta
te'].unique())

    confirmed_cases = []
    no_cases = []

    for i in regions:
        cases =
latest_data[latest_data['Province_State']==i]['Confirmed'].sum()

        if cases > 0:
            confirmed_cases.append(cases)
        else:
            no_cases.append(i)

    # remove areas with no confirmed cases
    for i in no_cases:
        regions.remove(i)

    # only show the top 10 states
    regions = [k for k, v in sorted(zip(regions, confirmed_cases),
key=operator.itemgetter(1), reverse=True)]

    for i in range(len(regions)):
        confirmed_cases[i] =
latest_data[latest_data['Province_State']==regions[i]]['Confirmed'].sum()

    # additional province/state will be considered "others"

    if(len(regions)>10):
        regions_10 = regions[:10]

```

```

        regions_10.append('Others')
        confirmed_cases_10 = confirmed_cases[:10]
        confirmed_cases_10.append(np.sum(confirmed_cases[10:]))
        plot_pie_charts(regions_10, confirmed_cases_10, title)
    else:
        plot_pie_charts(regions, confirmed_cases, title)
plot_pie_country_with_regions('India', 'COVID-19 Confirmed Cases in India')
days_since_1_22 = np.array([i for i in range(len(dates))]).reshape(-1, 1)
world_cases = np.array(world_cases).reshape(-1, 1)
total_deaths = np.array(total_deaths).reshape(-1, 1)
total_recovered = np.array(total_recovered).reshape(-1, 1)
days_since_1_22
days_in_future = 30
future_forecast = np.array([i for i in
range(len(dates)+days_in_future)]).reshape(-1, 1)
adjusted_dates = future_forecast[:-30]
future_forecast
start = '1-22-2020'
start_date = datetime.datetime.strptime(start, '%m-%d-%Y')
future_forecast_dates = []
for i in range(len(future_forecast)):
    future_forecast_dates.append((start_date +
datetime.timedelta(days=i)).strftime('%m-%d-%Y'))
X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed =
train_test_split(days_since_1_22, world_cases, test_size=0.25,
shuffle=False)
# transform our data for polynomial regression
poly = PolynomialFeatures(degree=3)
poly_X_train_confirmed = poly.fit_transform(X_train_confirmed)
poly_X_test_confirmed = poly.fit_transform(X_test_confirmed)
poly_future_forecast = poly.fit_transform(future_forecast)
# transform our data for bayesian polynomial regression
bayesian_poly = PolynomialFeatures(degree=4)
bayesian_poly_X_train_confirmed =
bayesian_poly.fit_transform(X_train_confirmed)
bayesian_poly_X_test_confirmed =
bayesian_poly.fit_transform(X_test_confirmed)

```

```

bayesian_poly_future_forecast =
bayesian_poly.fit_transform(future_forecast)

# polynomial regression
linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_confirmed, y_train_confirmed)
test_linear_pred = linear_model.predict(poly_X_test_confirmed)
linear_pred = linear_model.predict(poly_future_forecast)
print('MAE:', mean_absolute_error(test_linear_pred,
y_test_confirmed)) print('MSE:',mean_squared_error(test_linear_pred,
y_test_confirmed)) # polynomial regression
linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_confirmed, y_train_confirmed)
test_linear_pred = linear_model.predict(poly_X_test_confirmed)
linear_pred = linear_model.predict(poly_future_forecast) print('MAE:',
mean_absolute_error(test_linear_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_linear_pred, y_test_confirmed))
bayesian_search.best_params_
bayesian_confirmed = bayesian_search.best_estimator_
test_bayesian_pred =
bayesian_confirmed.predict(bayesian_poly_X_test_confirmed)
bayesian_pred = bayesian_confirmed.predict(bayesian_poly_future_forecast)
print('MAE:', mean_absolute_error(test_bayesian_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_bayesian_pred, y_test_confirmed))
plt.plot(y_test_confirmed)
plt.plot(test_linear_pred)
plt.legend(['Test Data', 'Polynomial Regression Predictions'])
plt.plot(y_test_confirmed)
plt.plot(test_bayesian_pred)
plt.legend(['Test Data', 'Bayesian Ridge Polynomial Predictions'])
# svm_confirmed = svm_search.best_estimator_
svm_confirmed = SVR(shrinking=True, kernel='poly',gamma=0.01,
epsilon=1,degree=3.8, C=0.002946)
svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
svm_pred = svm_confirmed.predict(future_forecast)
svm_test_pred = svm_confirmed.predict(X_test_confirmed)
plt.plot(y_test_confirmed)
plt.plot(svm_test_pred)

```

```

plt.legend(['Test Data', 'SVM Predictions'])
print('MAE:', mean_absolute_error(svm_test_pred, y_test_confirmed))
print('MSE:', mean_squared_error(svm_test_pred, y_test_confirmed))
def plot_predictions(x, y, pred, algo_name, color):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y)
    plt.plot(future_forecast, pred, linestyle='dashed', color=color)
    plt.title('Number of Coronavirus Cases Over Time', size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.legend(['Confirmed Cases', algo_name], prop={'size': 20})
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
plot_predictions(adjusted_dates, world_cases, linear_pred, 'Polynomial Regression Predictions', 'red')
plot_predictions(adjusted_dates, world_cases, svm_pred, 'SVM Predictions', 'green')
plot_predictions(adjusted_dates, world_cases, bayesian_pred, 'Bayesian Predictions', 'green')
def plot_predictions(x, y, pred, algo_name, color):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y)
    plt.plot(future_forecast, pred, linestyle='dashed', color=color)
    plt.title('# of Coronavirus Cases Over Time', size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('# of Cases', size=30)
    plt.legend(['Confirmed Cases', algo_name], prop={'size': 20})
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
plt.figure(figsize=(16, 9))
plt.plot(world_cases, 'black')
plt.plot(svm_pred, 'red')
plt.plot(linear_pred, 'green')
plt.plot(bayesian_pred)

```

```

plt.legend(['confirmed cases', 'SVM prediction', 'Linear
regression', 'Bayesian'], prop={'size': 20})

# Future predictions using polynomial regression
linear_pred = linear_pred.reshape(1,-1)[0]

poly_df = pd.DataFrame({'Date': future_forecast_dates[-30:], 'PR Predicted
number of Confirmed Cases Worldwide': np.round(linear_pred[-30:])})

poly_df

# Future predictions using Bayesian Ridge
bayesian_pred_df = pd.DataFrame({'Date': future_forecast_dates[-30:],
'Bayesian Ridge Predicted of Confirmed Cases Worldwide':
np.round(bayesian_pred[-30:])})

bayesian_pred_df

# Future predictions using SVM
svm_df = pd.DataFrame({'Date': future_forecast_dates[-30:], 'SVM Predicted
of Confirmed Cases Worldwide': np.round(svm_pred[-30:])})

svm_df

X_train_death, X_test_death, y_train_death, y_test_death =
train_test_split(days_since_1_22, total_deaths, test_size=0.25,
shuffle=False)

# svm_Death = svm_search.best_estimator_
svm_death = SVR(shrinking=True, kernel='poly', gamma=0.01,
epsilon=1, degree=2.5, C=0.05)

svm_death.fit(X_train_death, y_train_death)

svm_predi = svm_death.predict(future_forecast)

svm_dfi = pd.DataFrame({'Date': future_forecast_dates[-30:], 'SVM Predicted
of death Cases Worldwide': np.round(svm_predi[-30:])})

svm_dfi

X_train_death, X_test_death, y_train_death, y_test_death =
train_test_split(days_since_1_22, total_deaths, test_size=0.25,
shuffle=False)

#Poly_Death = Poly.best_estimator_
poly = PolynomialFeatures(degree=3)

poly_X_train_death = poly.fit_transform(X_train_death)
poly_X_test_death = poly.fit_transform(X_test_death)
poly_future_forecast = poly.fit_transform(future_forecast)

#Poly_Death = Poly.best_estimator_
linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_death, y_train_death)
test_linear_pred = linear_model.predict(poly_X_test_death)
linear_pred = linear_model.predict(poly_future_forecast)

```

```

print('MAE:', mean_absolute_error(test_linear_pred, y_test_death))
print('MSE:', mean_squared_error(test_linear_pred, y_test_death))
linear_pred = linear_pred.reshape(1, -1)[0]

poly_dfi = pd.DataFrame({'Date': future_forecast_dates[-30:], 'Poly_Reg
Predicted number of death Cases Worldwide': np.round(linear_pred[-30:])})

poly_dfi

X_train_recovery, X_test_recovery, y_train_recovery, y_test_recovery =
train_test_split(days_since_1_22, total_recovered, test_size=0.25,
shuffle=False)

# svm_recovery = svm_search.best_estimator_

svm_recovery = SVR(shrinking=True, kernel='poly', gamma=0.01,
epsilon=1, degree=3.2, C=0.0102)

svm_recovery.fit(X_train_recovery, y_train_recovery)

svm_prede = svm_recovery.predict(future_forecast)

svm_dfe = pd.DataFrame({'Date': future_forecast_dates[-30:], 'SVM Predicted
of recovery Cases Worldwide': np.round(svm_prede[-30:])})

svm_dfe

train_recovery, X_test_recovery, y_train_recovery, y_test_recovery =
train_test_split(days_since_1_22, total_recovered, test_size=0.25,
shuffle=False)

#Poly_recovery = Poly.best_estimator_

poly = PolynomialFeatures(degree=4)

poly_X_train_recovery = poly.fit_transform(X_train_recovery)
poly_X_test_recovery = poly.fit_transform(X_test_recovery)
poly_future_forecast = poly.fit_transform(future_forecast)

linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_recovery, y_train_recovery)

test_linear_pred = linear_model.predict(poly_X_test_recovery)
linear_pred = linear_model.predict(poly_future_forecast)

print('MAE:', mean_absolute_error(test_linear_pred,
y_test_recovery)) print('MSE:', mean_squared_error(test_linear_pred,
y_test_recovery)) linear_pred = linear_pred.reshape(1, -1)[0]

poly_dfe = pd.DataFrame({'Date': future_forecast_dates[-
30:], 'polynomialRegression Predicted number of recovery Cases
Worldwide': np.round(linear_pred[-30:])})

poly_dfe

adjusted_dates = adjusted_dates.reshape(1, -1)[0]

plt.figure(figsize=(16, 9))

plt.plot(adjusted_dates, world_cases)

```

```

plt.title('Number of Coronavirus Cases Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, total_deaths)
plt.title('Number of Coronavirus Deaths Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, total_recovered)
plt.title('Number of Coronavirus Recoveries Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases',
size=30) plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, total_active)
plt.title('Number of Coronavirus Active Cases Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Numner of Active Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_increase)
plt.title('World Daily Increases in Confirmed Cases',
size=30) plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)

```

```

plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_death)
plt.title('World Daily Increases in Confirmed Deaths', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases',
size=30) plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_recovery)
plt.title('World Daily Increases in Confirmed Recoveries', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases',
size=30) plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
#changing dates into days
v=1
day_date=[]
for i in
    range(len(dates)):
        v=i*1
        day_date.append(v)
day_date
#plotting the cases (confirmed, recover and deaths) with time

plt.figure(figsize=(20,12))
plt.plot(day_date,world_cases)
plt.plot(day_date,total_recovered,color='green')
plt.plot(day_date,total_deaths,color='r')
plt.title("Coronavirus Cases, deaths and recoveries with time", size=30)
plt.xlabel("Days Since 1/22/2020",size=30)
plt.ylabel("Number of Cases",size=30)

```

```

plt.legend(['Confirmed cases', 'Revovered cases', 'Death cases'],
prop={'size': 20})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

plt.figure(figsize=(16, 9))

plt.plot(total_recovered, total_deaths)

plt.title('Number of Coronavirus Deaths vs. Number of Coronavirus
Recoveries', size=30)

plt.xlabel('Number of Coronavirus Recoveries', size=30)

plt.ylabel('Number of Coronavirus Deaths', size=30)

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

def country_plot(x, y1, y2, y3, y4, country):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y1)
    plt.title('{} Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 9))
    plt.bar(x, y2)
    plt.title('{} Daily Increases in Confirmed Cases'.format(country),
size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 9))
    plt.bar(x, y3)
    plt.title('{} Daily Increases in Deaths'.format(country), size=30)

```

```

plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.bar(x, y4)
plt.title('{} Daily Increases in Recoveries'.format(country), size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases',
size=30) plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

country_plot(adjusted_dates, india_cases, india_daily_increase,
india_daily_death, india_daily_recovery, 'India')

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_cases)
plt.plot(adjusted_dates, italy_cases)
plt.plot(adjusted_dates, us_cases)
plt.plot(adjusted_dates, spain_cases)
plt.plot(adjusted_dates, france_cases)
plt.plot(adjusted_dates, germany_cases)
plt.plot(adjusted_dates, india_cases)
plt.title('Number of Coronavirus Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)

plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'],
prop={'size': 20})

plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_deaths)
plt.plot(adjusted_dates, italy_deaths)
plt.plot(adjusted_dates, us_deaths)

```

```

plt.plot(adjusted_dates, spain_deaths)
plt.plot(adjusted_dates, france_deaths)
plt.plot(adjusted_dates, germany_deaths)
plt.plot(adjusted_dates, india_deaths)
plt.title('Number of Coronavirus Deaths', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'],
prop={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_recoveries)
plt.plot(adjusted_dates, italy_recoveries)
plt.plot(adjusted_dates, us_recoveries)
plt.plot(adjusted_dates, spain_recoveries)
plt.plot(adjusted_dates, france_recoveries)
plt.plot(adjusted_dates, germany_recoveries)
plt.plot(adjusted_dates, india_recoveries)
plt.title('Number of Coronavirus Recoveries', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'],
prop={'size': 20})
plt.xticks(size=20
)
plt.yticks(size=20
) plt.show()

mean_mortality_rate = np.mean(mortality_rate)
plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, mortality_rate, color='orange')
plt.axhline(y = mean_mortality_rate, linestyle='--', color='black')
plt.title('Mortality Rate of Coronavirus Over Time', size=30)
plt.legend(['mortality rate', 'y='+str(mean_mortality_rate)], prop={'size':
20})
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Mortality Rate', size=30)

```

```
plt.xticks(size=20
)
plt.yticks(size=20
) plt.show()

HTML('''<div class="flourish-embed flourish-bar-chart-race" data-
src="visualisation/5434224" data-
url="https://flo.uri.sh/visualisation/5434224/embed"><script
src="https://public.flourish.studio/resources/embed.js"></script></div>''')
```