A

Project Report

on

TIME SERIES FORECASTING ON COVID-19 DATA WITH PARAMETRIC CURVE FITTING

Submitted for partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

By

SUSHMA SALUNKE	**(2451-17-733-010)**
KUNAPAREDDY JYOTHIRMAYEE	**(2451-17-733-050)**
KARNATI CHAITANYA	**(2451-17-733-051)**

Under the guidance of

Mrs.K.V. SRILAKSHMI
Assistant Professor
Department of CSE

M.V.S.R. ENGINEERING COLLEGE
Department of Computer Science and Engineering
(Affiliated to Osmania University & Recognized by AICTE)
Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510

Academic Year: 2020-21

CERTIFICATE

This is to certify that the project work entitled "Time Series Forecasting on Covid-19 Data with Parametric Curve Fitting" is a bonafide work carried out by **Sushma Salunke(2451-17-733-010), Kunapareddy Jyothirmayee (2451-17-733-050), Karnati Chaitanya (2451-17-733-051)** in partial fulfilment of the requirements for the award of degree of **Bachelor of Engineering** in **Computer Science And Engineering** from **Maturi Venkata Subba Rao (MVSR) Engineering College,** affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2020-21 under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide
Mrs.K.V.Srilakshmi
Assistant Professor
Department of CSE
MVSREC.

Head of the Department
Mr.J.Prasanna Kumar
Professor & Head
Department of CSE
MVSREC.

DECLARATION

This is to certify that the work reported in the present project entitled "Time Series Forecasting on Covid-19 Data with Parametric Curve Fitting" is a record of bonafide work done by us in the Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Osmania University during the Academic Year 2020-2021. The reports are based on the project work done entirely by us and not copied from any other source. The results embodied in this project report have not been submitted to any other University or Institution for the award of any degree or diploma.

Sushma Salunke	Kunapareddy Jyothirmayee	Karnati Chaitanya
(2451-17-733-010)	(2451-17-733-050)	(2451-17-733-051)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project guide **K.V.Srilakshmi,** Assistant Professor for her valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **Dr. G. Kanaka Durga** and **J. Prasanna Kumar,** Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (CSE).

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

Sushma Salunke (2451-17-733-010)
Kunapareddy Jyothirmayee (2451-17-733-050)
Karnati Chaitanya (2451-17-733-051)

VISION

- To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

MISSION

- To make learning process exciting, stimulating and interesting.
- To impart adequate fundamental knowledge and soft skills to students.
- To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students.
- To develop economically feasible and socially acceptable software.

PEOs:

PEO-1: Demonstrate technical competence to successfully execute industry related software projects as a team member, leader or entrepreneur to meet customer business objectives.

PEO-2: Engage in life-long learning process by pursuing professional certifications, higher education or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Advance in their professional careers by understanding the impact of computing on society or environment to make technical contributions using a multidisciplinary and ethical approach.

PROGRAM OUTCOMES(POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principle and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Lifelong learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

13. (PSO-1) Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.

14. (PSO-2) Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship.

COURSE OBJECTIVES AND OUTCOMES

Course Objectives:

- Understand the significance of survey in necessary domains for problem identification.
- Understand to map requirements into software specification.
- Learn to apply concepts of software engineering for design of the identified real world problem
- Improve the coding capabilities by implementing the various modules of project
- Comprehend the suitable documentation procedure for a technical project.
- Exhibit effective communication and presentation skills.
- Learn the process of planning the complete lifecycle of a project.

Course Outcomes:

CO1: Review recent advancements to formulate a precise problem statement with applications towards society.

CO2: Identify system requirements, explore design alternatives to design software based solution within the scope of project.

CO3: Implement, test and build the solution using contemporary technologies and tools.

CO4: Exhibit effective communication to present ideas clearly and produce well-structured technical report.

CO5: Demonstrate qualities necessary to work in a team and execute project as per plan.

ABSTRACT

COVID-19, a novel coronavirus, is currently a major worldwide threat. It has infected more than a million people globally leading to hundred-thousands of deaths. In such grave circumstances, it is very important to predict the future infected cases to support prevention of the disease and aid in the healthcare service preparation. During the current COVID-19 pandemic, there have been various efforts to forecast and analyze the infection cases, deaths and recoveries. Prediction or forecasting are the terms used to state about what will happen in the future. Sometimes these predictions enable and empower the user with capacities to control the situation. This is very true considering the current ―COVID-19 situation. Following that notion, we thought to design a model and then employed it for forecasting future COVID-19 cases.

Time series forecasting can be challenging as there are many different methods you could use i.e Auto Regression(AR), Moving Average(MA), Autoregressive Moving Average(ARMA), Autoregressive Integrated Moving Average(ARIMA) etc. Here, we have used FbProphet forecasting model on covid19 data for predicting the future cases where mainly we have focused on confirmed cases. Compared to the time series forecasting techniques, FbProphet is a powerful time series analysis package released by Core Data Science Team at Facebook. Prophet is an open source library published by Facebook that is based on decomposable (trend+seasonality+holidays) models. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters. It is simple and easy to go package for performing time series analytics and forecasting at scale. So, this is how one can use the Fbprophet library to easily predict future time series data without wasting much time on tuning the model. There is also a provision to perform cross-validation with the help of the Prophet library which helps in increasing the accuracy of predictions.

Sushma Salunke (2451-17-733-010)
Kunapareddy Jyothirmayee (2451-17-733-050)
Karnati Chaitanya (2451-17-733-051)

Index

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The concept of time series analysis has come a long way right since its inception. Plenty of research has been done in time series analysis to accomplish multiple objectives. Even in today's machine learning and deep learning era, time series forecasting plays a crucial role to make important business decisions. Since the time series data are in real time, it can be used in organizations such as power management, stock market, health care, business, marketing, weather forecasting and many more. Time series prediction is essentially a part of temporal data mining and statistics. It is the process of careful collection and rigorous study of data that has been collected over a continuous period and development of a proper model that describes the inherent trend of the series. Time series forecasting is a process to predict the future with the help of history data. The assumption of time series forecasting is that the information will repeat itself in near future.

1.1 PROBLEM STATEMENT

All of us are aware of the 2019-nCoV (Corona virus) outbreak and how deadly it has turned out to be. The World Health Organization is currently monitoring the situation closely and working towards developing a Global Strategic Preparedness. Every day a situation report is published to track the number of Confirmed Cases, Deaths, Recoveries, New Cases and so on. Think of a scenario where an estimate of the outbreak in the future can help countries implement strategic measures today. Strategic measures involve setting up health centers, stocking medical necessities, spreading awareness and so on.

1.2 OBJECTIVE

- The main objective of this project is to provide an early forecast model for the spread of novel coronavirus, also known as SARS-CoV-2.
- The aim of this system is the future forecasting of COVID-19 spread focusing on the number of new positive cases, and the number of recoveries.

- Forecasting of the COVID-19 outbreak aims to predict the number of confirmed cases and fatalities that would occur in the next 365 days by the spread of the Corona virus using FbProphet model. This aims to make the public aware of the consequences of not following the health safety measures.
- These prediction systems can be very helpful in decision making to handle the present scenario to guide early interventions to manage these diseases very effectively.

1.3 MOTIVATION

Time Series Forecasting has always been a very important area of research in many domains because many different types of data are stored as time series. For example we can find a lot of time series data in medicine, weather forecasting, biology, supply chain management and stock prices forecasting, etc.

Modeling and forecasting the daily spread behavior of the virus can assist the health systems to be ready to accommodate the upcoming number of patients. The main motive of this system is to prepare ourselves for the future by knowing the present scenario.

1.4 SCOPE

The cases of COVID-19 will rise in the coming days and the situation may turn alarming if proper measures are not followed. Since the economic activities have started in the country, people need to be more careful while going out. And explosion of the pandemic in the whole country can cause a serious damage to human lives, healthcare system as well as the economy of the country. This system forecasts thus can also be of great help for the authorities to take timely actions and make decisions to contain the COVID-19 crisis. The future modifications to further improve the predictive accuracy of the models will include the creation of ensembles of the presented models that would combine the best of many worlds in order to reduce the overall error as well as the adoption of multivariate time series modeling that take into account other factors that are either directly or indirectly related to the spread of the pandemic. Real-time live forecasting will be one of the primary focuses in our future work.

1.5 SOFTWARE AND HARDWARE REQUIREMENTS

1.5.1 Software Requirements

1.5.1.1 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning etc. It is a client-server application. The application starts the server on local machine and opens the notebook interface in web browser where it can be edited and run from. The notebook is saved as ipynb file and can be exported as html, pdf and LaTeX files.

1.5.1.2 Python

Python is an interpreted, high-level and general-purpose popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for web development(server-side), software development, mathematics, system scripting. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms including structured(particularly, procedural), object-oriented and functional programming. Python is simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace.

Libraries –

- Pickle - Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling.
- Prophet - Prophet is open source software released by Facebook's Core Data Science team. It is available for download on CRAN and PyPI. Prophet is a procedure for forecasting time series data based on an additive model where

non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

- Flask - Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
- CORS - It stands for Cross-Origin Resource Sharing. It allows you to make requests from one website to another website in the browser, which is normally prohibited by another browser policy called the Same-Origin Policy (SOP).
- Sklearn - Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. The sklearn.metrics module implements functions assessing prediction error for specific purposes.
- Scipy - Curve fitting involves finding the optimal parameters to a function that maps examples of inputs to outputs. The SciPy Python library provides an API to fit a curve to a dataset.
- Lmfit - Lmfit provides a high-level interface to non-linear optimization and curve fitting problems for Python. It builds on and extends many of the optimization methods of scipy.optimize.

1.5.1.3 Postman

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs faster. Postman is a popular API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result is more efficient and less tedious work.

1.5.2 Hardware Requirements

- Processor Intel i3 and above
- RAM:4GB and higher

CHAPTER II

LITERATURE SURVEY

2.1 SURVEY OF MAJOR AREA RELEVANT TO PROJECT

Millions of people have been infected and lakhs of people have lost their lives due to the worldwide ongoing novel Coronavirus (COVID-19) pandemic. It is of utmost importance to identify the future infected cases and the virus spread rate for advance preparation in the healthcare services to avoid deaths. Accurately forecasting the spread of COVID-19 is an analytical and challenging real-world problem to the research community. World health organization (WHO) declared on 11th March, 2020, the outbreak of COVID-19 as a pandemic.

The spread of COVID-19 can be classified under three major stages- 1. Local outbreak: at this stage, spreading chain of the virus among the people can be tracked, and the source of infection can be found out. The cases in this stage mostly relate to within family or friends, or the local exposure. 2. Community transmission: at this stage, source of the chain of infected people cannot be found out. The infected cases grow through cluster transmission in the communities. 3. Large scale transmission: at this stage, the virus spreads rapidly to other regions of a country due to uncontrolled mobility of people at large scale. The virus has spread at large scale, and much information is available for the analysis. Predictive analysis of COVID-19 has become a research area to support health services and governments to plan and contain the spread of the infectious disease.

Modeling and forecasting the daily spread behavior of the virus can assist the health systems to be ready to accommodate the upcoming number of patients. Accurate forecasting of the disease is a matter of concern because it may impact governments policy, containment rules, health system, and social life.

The objective of this research is to provide evaluative study of prediction models using COVID-19 cases, and forecasting the impact of the virus in the affected countries, and worldwide. We present trend analysis of COVID-19 cases, and compared the performance of the models using the metrics such as the mean absolute error (MAE), root mean square error (RMSE), root relative squared error (RRSE), and mean absolute percentage error (MAPE). We generate forecasting results for COVID-

19 confirmed, active, recovered, and death cases. The results show that ARIMA outperformed the Prophet model.

Time series forecasting models are used to predict the futuristic outcomes based on historical information. They have adopted ARIMA and Facebook Prophet (FbProphet) model in our evaluative and forecasting study.

According to their study they have suggested that this forecasting can be further improved by using ensemble of multiple prediction models. The obtained forecasting results further can be improved by taking various variables into account like population density, weather, health system, patient history etc. using deep learning techniques, and artificial intelligence.

According to recent study, Artificial Intelligence, Machine Learning and Deep Learning techniques to diagnose COVID19 disease. These techniques and methods help to detect the COVID-19 symptoms from the crowdsourced respiratory sound data. They have detect the COVID-19 positive case symptoms with cough sound, screening of a patient breath with speech results, and an Artificial Intelligence (AI) machine can sense COVID-19 symptoms from continuous speech and mental health situation. Researchers have performed PCA and SVM classifier on COVID-19 dataset to detect COVID-19 disease and achieved 80%, 82%, 80% of accuracies for 3 tasks (COVID Positive/Non-COVID, COVID Positive with Cough/Non-COVID with Cough, COVID Positive with cough/Non-COVID asthma cough) with three modalities (Cough + Breath, Cough, Breath). They have also performed an SVM classifier to the diagnosis of COVID-19 disease on Corona voice detected data app and analyzed conditions of COVID patient concerning sleep, fatigue, and anxiety is recorded 57%, 50%, 50% accuracy. They have performed PDS, down-sampling, low pass filter, and TPE methods to produce labeled data on the COUGHVID dataset; and finally, it produced 632 labeled COVID-19 cough records the accuracy of audible dyspnea (93%), wheezing (90.5%), stridor (98.7%), choking (99.1%), nasal congestion (99.2), and 86.2% of accuracy for labeled as mild.

These techniques help researchers and clinical scientists to move towards their research directions.

2.2 TECHNIQUES AND ALGORITHMS RELEVANT WITH RESPECT TO PROJECT

Time-series is simply a series of data points ordered in time. In a time-series, time is often the independent variable, and the goal is usually to make a forecast for the future. It is a forecasting area that focuses on analyzing past observations of random variables to develop a model that captures the underlying relationship and its pattern. This approach is usually beneficial when there is no knowledge or little knowledge of the data generating process. Over past decades, a lot of research has produced a significant development and improvement of time series forecasting models.

We have used FbProphet model to predict the confirmed cases of this COVID-19.Other than ARIMA,SARIMA,AR,MA etc. Prophet is an open-source tool from Facebook initially developed to forecast time series data related to businesses' time series problems. It is based on a decomposed additive model where non-linear trends are fit with seasonality; it also takes into account the effects of holidays. The trend shows the tendency of the data to increase or decrease over a long period, and it filters out the seasonal variations. Seasonality is the variation that occurs over a short period, and is not prominent enough to be called a "trend".

Based on the dataset the future impacts of COVID-19 will be forecasted considering the total confirmed cases worldwide. To use this model ,we have imported Prophet library from FbProphet where 'ds' and 'y' are two dataframe columns given as input to Prophet. To make predictions, a dataframe need to be created with the future dates. In context of FbProphet these are identified as periods. The periods are parameters indicating how many days to create, the forecasting consists of 365 days as the period. Prophet includes functionality for time series cross validation to measure forecast error using historical data. This is done by selecting cutoff points in the history, and for each of them fitting the model using data only up to that cutoff point. The output of cross validation is a dataframe with the true values y and the out-of-sample forecast values yhat, at each simulated forecast date and for each cutoff date. The exact prediction is shown as yhat and yhat_lower, yhat_upper represents the minimum prediction and maximum prediction.

2.3 APPLICATIONS

- Stock Market Prediction

  Stock prices are not randomly generated values instead they can be treated as a discrete-time series model which is based on a set of well-defined numerical data items collected at successive points at regular intervals of time. Since it is essential to identify a model to analyze trends of stock prices with adequate information for decision making, it recommends that transforming the time series using ARIMA is a better algorithmic approach than forecasting directly, as it gives more authentic and reliable results.

  Autoregressive Integrated Moving Average (ARIMA) Model converts non-stationary data to stationary data before working on it. It is one of the most popular models to predict linear time series data.

  ARIMA model has been used extensively in the field of finance and economics as it is known to be robust, efficient and has a strong potential for short-term share market prediction.

- Electricity requirement prediction using time series forecasting

  The main objective of this application is to forecast the electricity requirement of a particular household or an office or any building. Forecasting is done using the PROPHET model which gives better results compared to other models like ARIMA and so on. Dataset considered here is a publicly available dataset called 'Appliance' dataset with what are all the appliances that are there in the particular household and number of appliances that are running on a day at every 10 minutes interval and so on. From the entire dataset, only two attributes are selected, and Log transformation is applied to the selected attributes. Finally, the PROPHET model is applied and the forecasting is done. The findings of the proposed models are: (i) Forecasting is done for the next 30 days based on different components like daily component, weekly component and trend component (ii) Wednesday is the lowest power utilization day and, power utilization increases till Saturday and Saturday is the highest power utilization day (iii) PROPHET model makes predictions

very accurate based on the future data and is easy to make predictions compared to ARIMA.

- Air Passengers Occupancy Prediction

  Predicting a number of passengers per trip is an important topic in airline business and travel economy which has spurred the interest of airline companies to develop better predictive models to accommodate in their business model. This paper presents extensive process of predicting the occupancy of the airline seats using the ARIMA model. Published airline passenger data is obtained from RPubs is used with predictive model developed. Results achieved convey that the autoregressive integrated moving average model has a strong potential for prediction and can compete with existing models for this business projects.

- Time Series Analysis of Climate Variables

  Climatic changes are predominantly governed by the changes in temperature and precipitation. Studies focusing on the assessment of climatic changes are often based upon the analysis of time series of temperature and precipitation. Analysis of time series invariably involves the evaluation of trends and seasonality in the data. Trends are the long-term increase or decrease in the time series, whereas seasonality refers to the variations in the data at regular short intervals such as weekly, monthly, biyearly, quarterly, etc.

CHAPTER III

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

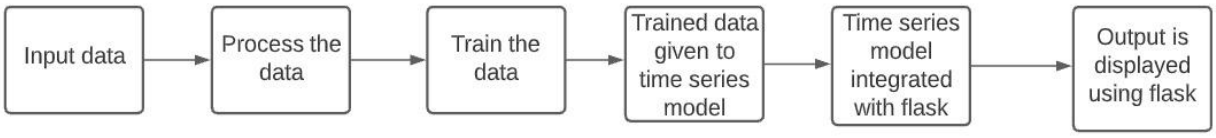

Fig – 3.1 System Architecture

This represents the System Architecture where we give the input data as the covid dataset then processing the data takes place next we train the data and this trained data is given to Time Series model where we fit the dataset into the model then predictions are given by the model, we integrate this model into flask. At the end we call REST API through Postman where prediction horizon parameter is passed to the API and we get back JSON response with forecast data.

3.2 MODULES

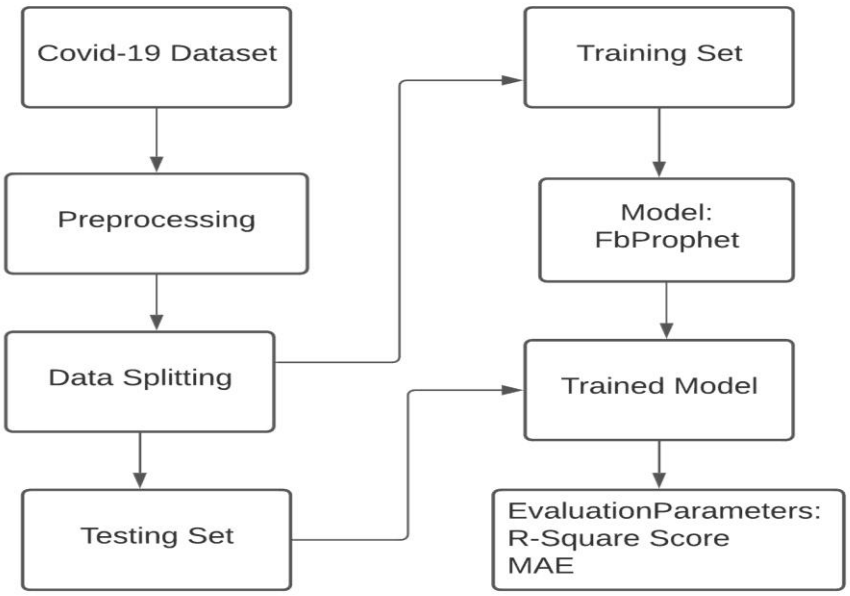

Fig – 3.2.1 Training and Testing the model

In this module, First we give the COVID-19 Dataset as input then we perform Data Pre-processing where we Clean the data like removing null values, removing the missing values etc. The next step is Data Splitting where we split the dataset into Training set and Testing set we give the training set to the Time Series model then we get the output as trained model. To this trained model testing set is given and then we perform evaluation metrics i.e. R-Square Score, Mean Absolute Error etc.

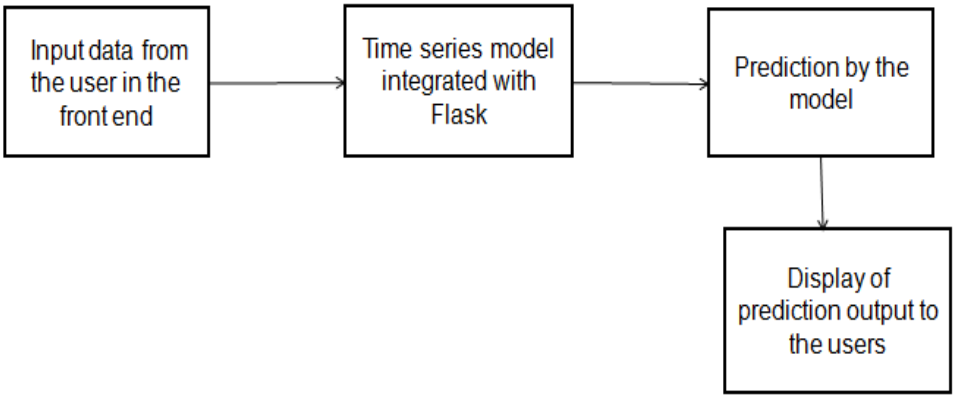

Fig – 3.2.2 Model Integration with flask

In the second module i.e. model integration with flask we give the Input as the COVID-19 Dataset in the front end then the Time Series model (Fbprophet) is integrated with Flask. Flask works perfectly to expose Prophet model to the external world through REST API .Then in the Postman we give the input as Horizon and the Predicted outputs will be displayed to the user.

CHAPTER IV
IMPLEMENTATION

4.1 ENVIRONMENTAL SETUP

To develop our project, we have made use of the Jupyter Notebook of Anaconda Navigator which is known for executing python projects effectively.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux. To get Navigator, install Anaconda.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter notebooks provide an environment where you can freely combine human-readable narrative with computer-readable code.

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. In other words: it's a single document where you can run code, display the output, and also add explanations, formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster. It's Free and Easy to Start, Wide support for all APIs and Schemas and Extensible. Postman is a scalable API testing tool that quickly integrates into CI/CD

pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development.

API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.

4.2 IMPLEMENTATION OF EACH MODULE

In the training and testing module, we have taken input as COVID-19 dataset where the columns present are SNO, Observation Date, Province/State, Country/Region, Last Update, Confirmed, Deaths, Recovered then in Data Pre-processing where we done the cleaning part we have removed the columns ‘SNO’, ‘Last Update’ renaming of columns is done i.e. ‘Observation Date’ to ‘Date’, Country/Region as Country and in 'Country/Region' column Mainland China is renamed as 'China'. In the third step i.e. Data Splitting where we split the data into Training and Testing set. In our case we split the training and testing into 70:30.Then, in the training set after training the data to that data FbProphet model(Forecasting model) is given then the result is the trained model. To that trained model we give the testing set and then we apply evaluation metrics to that model. Other than machine learning in time series forecasting we have different evaluation metrics like RMSE, MAE, MSE, MDAPE etc. We have considered RMSE, MSE, MAE as our evaluation metrics.

The last part of training and testing module is Parametric Curve Fitting as we are mainly focusing on prediction of confirmed cases in the future. In mathematics, Parametric curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints. In parametric curve fitting we have considered Time series data of confirmed cases where we have imported curve_fit by using the module optimize from scipy. Then, we have considered to forecast the cases of India and we have created total and new cases as the new columns where new cases is calculated as new cases(t) = total(t) - total(t-1).

The objective of curve fitting is to find the optimal combination of parameters that minimize the error. For the total cases time series, we need to find the best function to model the data and then fit to get the optimal parameters. Therefore, logistic function to model the total cases and gaussian function to model the new cases. Now that we

have the models fitted, we can finally use them to forecast. We have plotted the parametric curve using the logistic and gaussian models taking into consideration both the total and new cases.

In the next module i.e. time series model integration with flask where flask is having some functionalities using that we integrate then the time series model is saved using pickle model and the pickle module is loaded. After loading the loaded pickle module then we give the input to the model from the front end then by using postman we predict the output that will be displayed to the users.

In the Postman which is a collaboration platform for API development. We have done REST API call through Postman where the request is sent with the method Post to the localhost that is Horizon parameter is passed to the API and we get back JSON response with forecast data that is ds, yhat, yhat_lower, yhat_upper will be given as output. A Flask extension for handling Cross Origin Resource Sharing (CORS), making cross-origin AJAX possible.

4.3 INTEGRATION AND DEPLOYMENT

- As we are using time series forecasting if the dataset used was seasonal then we could get the best results but as the dataset taken was covid19 which was not a seasonal dataset.
- Deployment of FB prophet with flask is also one of the issue.

CHAPTER V
EVALUATION

5.1 DATASET

In our Project the dataset used is employed from Kaggle which consists of 236018 rows and 8 columns of different countries all over the world.

SNO, Observation Date, Province/State, Country/Region, Last update, Confirmed, Deaths, Recovered are the columns present in this dataset.

(dataset:https://drive.google.com/drive/folders/1sZoQDaO_GUDHKXLAhSOVWOa1jZTF3YKA?usp=sharing)

	A	B	C	D	E	F	G	H
	SNo	Observati	Province/	Country/R	Last Upda	Confirmed	Deaths	Recovered
1	1	01/22/202	Anhui	Mainland	1/22/2020	1	0	0
2	2	01/22/202	Beijing	Mainland	1/22/2020	14	0	0
3	3	01/22/202	Chongqin	Mainland	1/22/2020	6	0	0
4	4	01/22/202	Fujian	Mainland	1/22/2020	1	0	0
5	5	01/22/202	Gansu	Mainland	1/22/2020	0	0	0
6	6	01/22/202	Guangdor	Mainland	1/22/2020	26	0	0
7	7	01/22/202	Guangxi	Mainland	1/22/2020	2	0	0
8	8	01/22/202	Guizhou	Mainland	1/22/2020	1	0	0
9	9	01/22/202	Hainan	Mainland	1/22/2020	4	0	0
10	10	01/22/202	Hebei	Mainland	1/22/2020	1	0	0
11	11	01/22/202	Heilongjia	Mainland	1/22/2020	0	0	0
12	12	01/22/202	Henan	Mainland	1/22/2020	5	0	0
13	13	01/22/202	Hong Kon	Hong Kon	1/22/2020	0	0	0
14	14	01/22/202	Hubei	Mainland	1/22/2020	444	17	28
15	15	01/22/202	Hunan	Mainland	1/22/2020	4	0	0
16	16	01/22/202	Inner Mor	Mainland	1/22/2020	0	0	0
17	17	01/22/202	Jiangsu	Mainland	1/22/2020	1	0	0
18	18	01/22/202	Jiangxi	Mainland	1/22/2020	2	0	0
19	19	01/22/202	Jilin	Mainland	1/22/2020	0	0	0
20	20	01/22/202	Liaoning	Mainland	1/22/2020	2	0	0
21	21	01/22/202	Macau	Macau	1/22/2020	1	0	0
22	22	01/22/202	Ningxia	Mainland	1/22/2020	1	0	0

Fig- 5.1 Dataset

5.2 EVALUATION PROCEDURE

- Root Mean Squared Error (RMSE): Represents the sample standard deviation of the differences between predicted values and observed values. This is a great measurement to use when comparing models as it shows how many deviations from the mean the forecasted values fall.
- Mean Absolute Error (MAE) : Takes the sum of the absolute difference from actual to forecast and averages them. It is less sensitive to the occasional very large error because it does not square the errors in the calculation.
- Mean Squared Error (MSE): It is calculated as the average of the squared forecast error values. Squaring the forecast error values forces them to be positive; it also has the effect of putting more weight on large errors.
- Median Absolute Percentage Error (MDAPE):It calculates the error between the forecast and the eventual outcomes.

5.3 TEST CASES

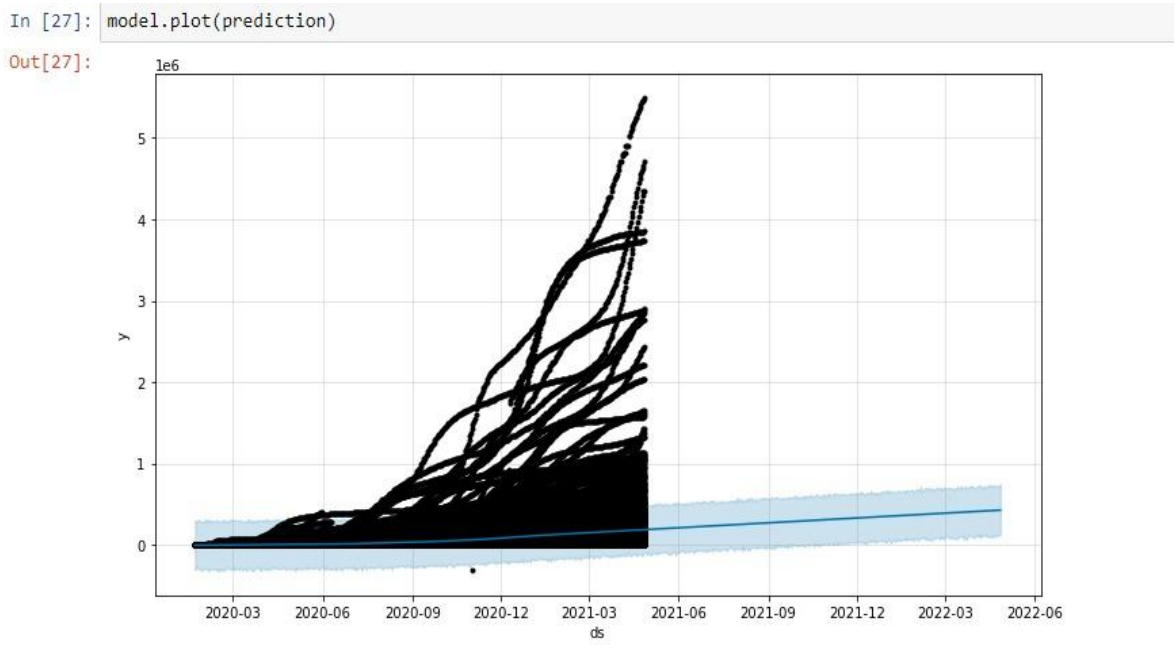

Fig - 5.3.1 Predictions plotted by FbProphet model

```
In [28]: model.plot_components(prediction)
```

Out[28]:

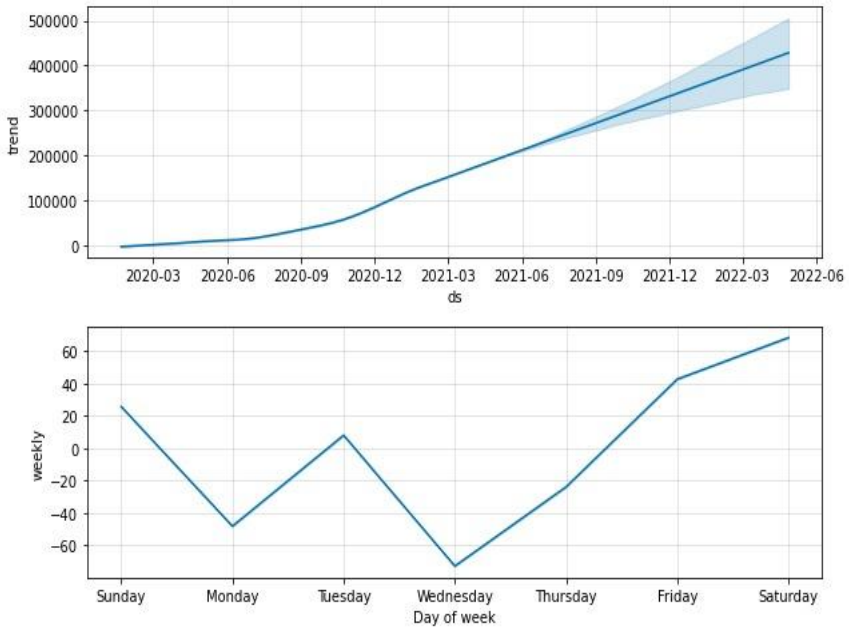

Fig - 5.3.2 Time Series Components Predictions by FbProphet model

```
In [57]: from fbprophet.diagnostics import performance_metrics
         df_p = performance_metrics(df_cv)
         df_p.head()

         INFO:fbprophet:Skipping MAPE because y close to 0
```

Out[57]:

	horizon	mse	rmse	mae	mdape	coverage
0	2 days	5.452238e+10	233500.277901	93277.958390	4.360665	0.904876
1	3 days	5.511596e+10	234767.887082	93798.521085	4.294970	0.904374
2	4 days	5.562212e+10	235843.424425	94159.943383	4.278091	0.903941
3	5 days	5.618304e+10	237029.627620	94637.041455	4.239989	0.902742
4	6 days	5.653301e+10	237766.708640	94678.733180	4.282997	0.902983

Fig - 5.3.3 Performance Metrics for FbProphet model

5.4 RESULTS

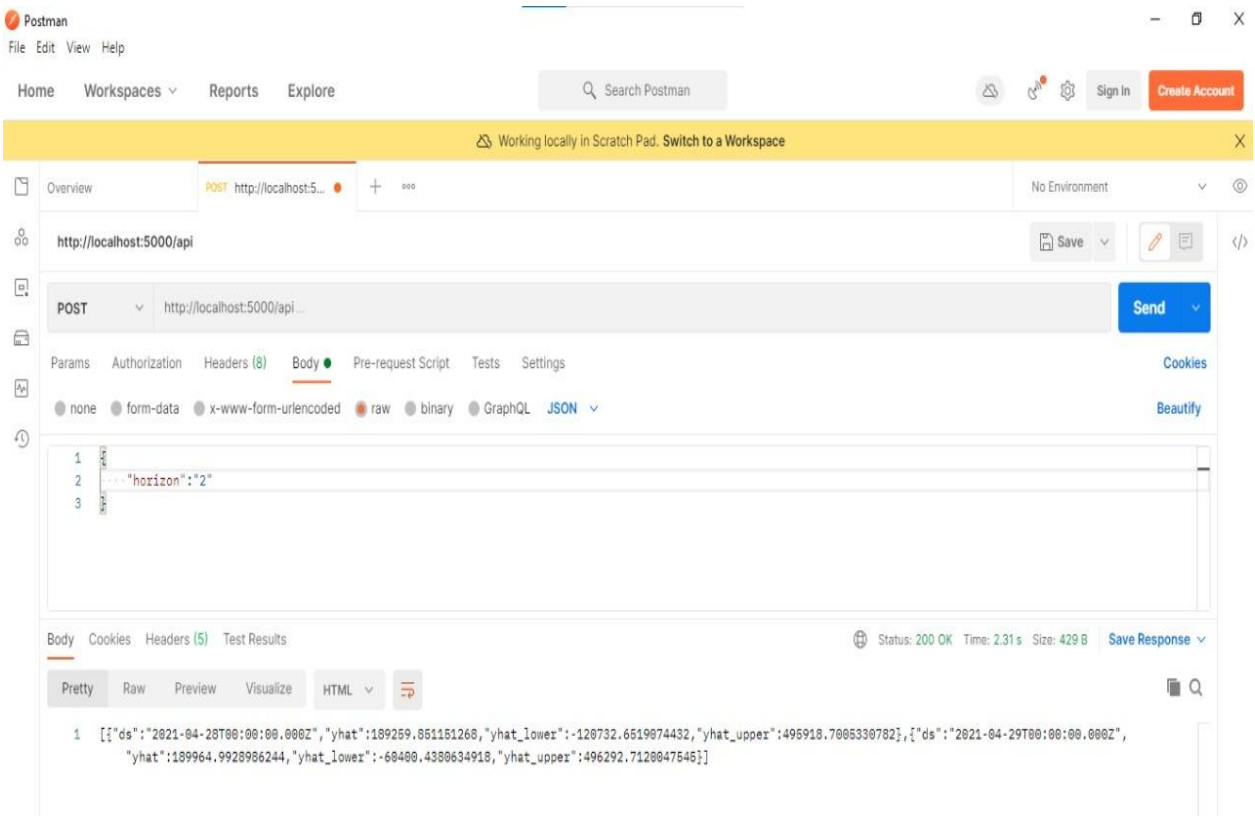

Fig - 5.4.1 Rest API call through Postman for predicting horizon parameter where value is 2

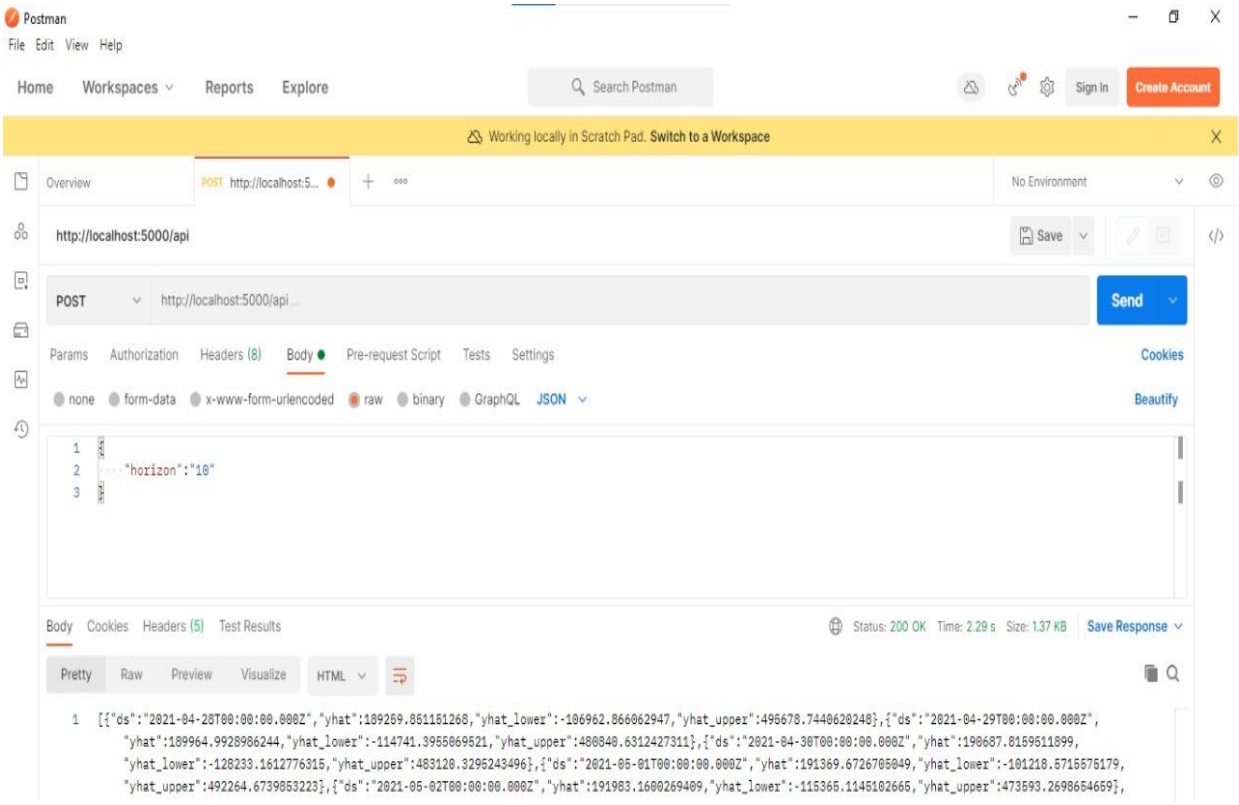

Fig - 5.4.2 Rest API call through Postman for predicting horizon parameter where value is 10

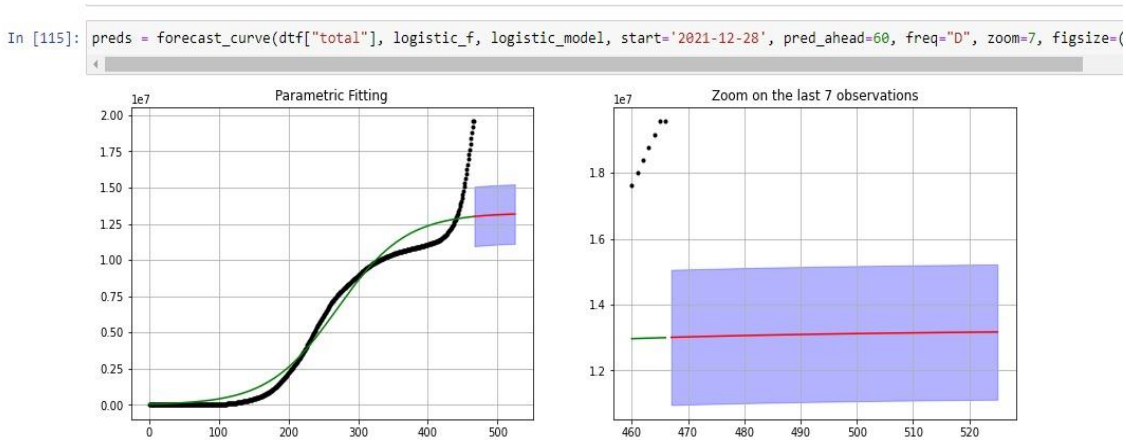

Fig – 5.4.3 Parametric Curve Fitting for Prediction of total covid-19 cases

```
In [119]: preds = forecast_curve(dtf["new"], gaussian_f, gaussian_model, start='2021-12-28', pred_ahead=60, freq="D", zoom=7, figsize=(15,5
```

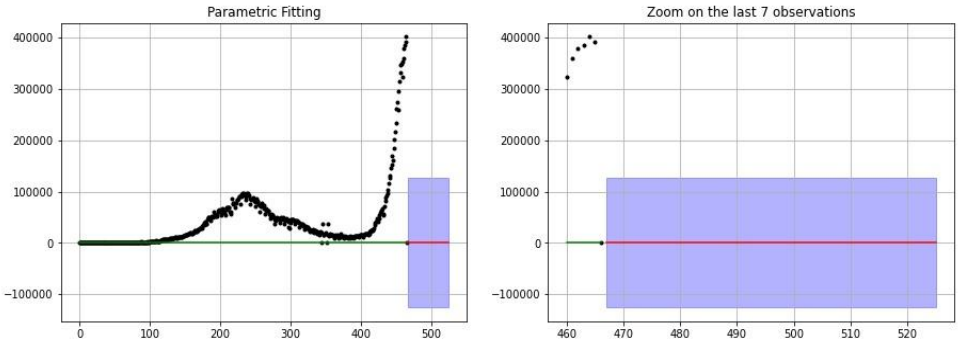

Fig – 5.4.4 Parametric Curve Fitting for Prediction of new covid-19 cases

CHAPTER VI
CONCLUSION & FUTURE ENHANCEMENTS

In this project, we have proposed a Time Series model using FbProphet on covid19 dataset to predict number of confirmed cases in future and even we have shown a parametric curve on the time series dataset. At present we have deployed our model using flask through REST API in Postman where prediction horizon parameter is passed to the API and we get back JSON response with forecast data. Further, we can deploy the model in the GUI Application so that it will be useful for the government to check for the increase in number of cases and take necessary action.

The future scope of this project is to forecast for a minute and hourly forecasts also using PROPHET, which are the components not available also it should be ensuring to create computerized calculations to provide information within a standard range and naturally predict the number of cases daily and every week. According to these principles, the government and emergency clinics can also keep a clear responsibility and provide flexible clinical help/services for new patients.

REFERENCES

[1]https://medium.com/analytics-vidhya/time-series-forecasting-of-covid19-data-with-fbprophet-3a73dba59106.

[2]https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ecf2c0e3a.

[3] D. Acemoglu, V. Chernozhukov, I. Werning, M.D. Whinston, A multi-risk sir model with optimally targeted lockdown. Tech. rep., National Bureau of Economic Research (2020)

[4] G.E. Box, D.A. Pierce, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J. Am. Stat. Assoc. 65(332), 1509–1526 (1970)

[5] S. B. Lajevardi and B. Minaei-Bidgoli, "Combination of Time Series Decision Tree and Clustering : A Case Study in Aerology Event Prediction", International Conference on Computer and Electrical Engineering, pp. 111-115, 2008.

[6] S. Mehrmolaei and M. R. Keyvanpour, "A Brief Survey on Event Prediction Methods in Time Series" in Artificial Intelligence Perspectives and Applications, Springer International Publishing, pp. 235-246, 2015.

[7] Lu, H.; Stratton, C.W.; Tang, Y.W. Outbreak of Pneumonia of Unknown Etiology in Wuhan China: The Mystery and the Miracle. *J. Med Virol.* **2020**, *92*, 401–402.

[8] Rasheed Omobolaji Alabi, Akpojoto Siemuri, and Mohammed Elmus-rati. Covid-19: Easing the coronavirus lockdowns with caution. medRxiv,2020

[9] WHO. Laboratory Testing for Coronavirus Disease 2019 (COVID-19) in Suspected Human Cases: Interim Guidance, 2 March 2020; Technical report. WHO: Geneva, Switzerland, 2020.

[10] H. Chen and L. Wu, “A new measure of forecast accuracy,” the 2nd IEEE International Conference on Information and Financial Engineering, Chongqing, 2010. pp. 710-712.

APPENDIX

Source Code

Model.ipynb

```
import pandas as pd
import numpy as np
from fbprophet import Prophet
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
import seaborn as sns
sns.set(color_codes=True)
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
df=pd.read_csv('covid_19_data.csv')
df.head()
df.tail()
df.drop('SNo',axis=1,inplace=True)
df.head()
df.drop('Last Update',axis=1,inplace=True)
df.rename(columns={'Country/Region':'Country','ObservationDate':'Date'},inplace=Tr
ue)
df.head()
df['Country']=df['Country'].replace(['Mainland China'],'China')
df.plot()
country_grouped=df.groupby(['Country','Date']).agg({"Confirmed":'sum',"Recovered"
:'sum',"Deaths":'sum'})
country_grouped["Death_Percent"] = country_grouped["Deaths"] /
country_grouped["Confirmed"] * 100
country_grouped["Recovered_Percent"] = country_grouped["Recovered"] /
country_grouped["Confirmed"] * 100
# get total sum of each country
total_sum_country = country_grouped.groupby(['Country']).tail(1)
total_sum_country.tail(20)
```

```
total_sum_country = total_sum_country.reset_index()
top_10_confirmed_country =
total_sum_country.sort_values(by=['Confirmed'],ascending=False).head(10)
    fig1, ax1 = plt.subplots()
    ax1.pie(top_10_confirmed_country['Confirmed'],
    labels=top_10_confirmed_country['Country'], autopct='%1.1f%%',
          shadow=True, startangle=90)
    ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
     plt.show()
    global_cases = df.reset_index()
    confirmed_df = global_cases[['Date', 'Confirmed']]
    recovered_df = global_cases[['Date', 'Recovered']]
    deaths_df = global_cases[['Date', 'Deaths']]
  def rename_func(df):
      cols = df.columns
      df.rename(columns= {cols[0]: 'ds', cols[1]:'y'}, inplace= True)
   return df
    confirmed_df = rename_func(confirmed_df)
    recovered_df = rename_func(recovered_df)
    deaths_df = rename_func(deaths_df)

    def train_test_split(df, ratio):
      """
      Parameters:
      dataframe- Dataframe required to be split
      ratio: Percentage of training split in the scale of 0-100
      """
      divisor = round((ratio/100)*df.shape[0])
      print(divisor)
      train = df.iloc[:divisor]
      test = df.iloc[divisor:]
   return train, test

 confirmed_train, confirmed_test = train_test_split(confirmed_df, 70)
```

```
confirmed_train.shape, confirmed_test.shape
dataset = df[['Date','Confirmed']]
 dataset.head()
 dataset.columns = ['ds','y']
 dataset.head()

model = Prophet()
model.fit(dataset)
future_dates=model.make_future_dataframe(periods=365)
future_dates.tail()
prediction=model.predict(future_dates)
prediction.head()

model.plot(prediction)
model.plot_components(prediction)

from fbprophet.diagnostics import cross_validation
df_cv = cross_validation(model, initial='30 days', period='15 days', horizon = '10
days')
 df_cv.head()
 df_cv.tail()

from fbprophet.diagnostics import performance_metrics
df_p = performance_metrics(df_cv)
 df_p.head()
 df_p.head()

from fbprophet.plot import plot_cross_validation_metric
fig = plot_cross_validation_metric(df_cv, metric='rmse')
from fbprophet.plot import plot_cross_validation_metric
fig = plot_cross_validation_metric(df_cv, metric='mse')

r2_score(df_cv.y, df_cv.yhat)
mean_squared_error(df_cv.y, df_cv.yhat)
```

```
mean_absolute_error(df_cv.y, df_cv.yhat)

import pickle
with open('forecast_model.pckl', 'wb') as fout:
 pickle.dump(model, fout)
with open('forecast_model.pckl', 'rb') as fin:
    m2 = pickle.load(fin)

from flask import Flask, jsonify, request
 from flask_cors import CORS, cross_origin
app = Flask(__name__)
CORS(app)
@app.route('/api',methods=['POST'])
def predict():
      horizon = int(request.json['horizon'])
      print(request)

      future2 = m2.make_future_dataframe(periods=horizon)
      forecast2 = m2.predict(future2)

      data = forecast2[['ds', 'yhat', 'yhat_lower', 'yhat_upper']][-horizon:]

       ret = data.to_json(orient='records', date_format='iso')

        return ret


    if __name__ == "__main__":
    app.run(debug=False)
```

Parametric Curve Fitting.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
## For parametric fitting
from scipy.optimize import curve_fit

 dtf = pd.read_csv("time_series_covid_19_confirmed.csv", sep=",")
 dtf.head()

## group by country, sum and transpose
dtf = dtf.drop(['Province/State','Lat','Long'],
axis=1).groupby("Country/Region").sum().T

## convert index to datetime
dtf.index = pd.to_datetime(dtf.index, infer_datetime_format=True)

## create total cases column
dtf = pd.DataFrame(index=dtf.index, data=dtf["India"].values, columns=["total"])
## create new cases column
dtf["new"] = dtf["total"] - dtf["total"].shift(1)
dtf["new"] = dtf["new"].fillna(method='bfill')
 dtf.head()
 dtf.tail()
    '''
     Linear function: f(x) = a + b*x
     '''
     def f(x):
        return 10 + 25000*x
     y_linear = f(x=np.arange(len(dtf)))
     '''
     Exponential function: f(x) = a + b^x
     '''
```

```
def f(x):
    return 10 + 1.18**x

y_exponential = f(x=np.arange(len(dtf)))
'''
Logistic function: f(x) = a / (1 + e^(-b*(x-c)))
'''
def f(x):
    return 90000 / (1 + np.exp(-0.5*(x-20)))

y_logistic = f(x=np.arange(len(dtf)))

fig, ax = plt.subplots(figsize=(13,5))
ax.scatter(dtf["total"].index, dtf["total"].values, color="black")
ax.plot(dtf["total"].index, y_linear, label="linear", color="red")
ax.plot(dtf["total"].index, y_exponential, label="exponential", color="green")
ax.plot(dtf["total"].index, y_logistic, label="logistic", color="blue")
ax.legend()
plt.show()

def f(x):
    return 6000 * np.exp(-0.5 * ((x-60)/6)**2)

y_gaussian = f(x=np.arange(len(dtf)))
fig, ax = plt.subplots(figsize=(13,5))
ax.bar(dtf["new"].index, dtf["new"].values, color="black")
ax.plot(dtf["new"].index, y_gaussian, color="red")
plt.show()

'''
Logistic function: f(x) = capacity / (1 + e^-k*(x - midpoint) )
'''
def logistic_f(X, c, k, m):
    y = c / (1 + np.exp(-k*(X-m)))
```

```
    return y
## optimize from scipy
X_l=np.arange(len(dtf["total"]))
y_l=dtf["total"].values
max_fev=500
p0_l=[np.max(dtf["total"]), 1, 1]

logistic_model, cov_l = curve_fit(logistic_f, X_l, y_l, p0_l)

'''
Gaussian function: f(x) = a * e^(-0.5 * ((x-μ)/σ)**2)
'''
def gaussian_f(X, a, b, c):
    y = a * np.exp(-0.5 * ((X-b)/c)**2)
    return y
## optimize from scipy

X=np.arange(len(dtf["new"]))
y=dtf["new"].values
maxfev=500
p0=[1, np.mean(dtf["new"]), 1]
## print the parameters
gaussian_model, cov = curve_fit(gaussian_f, X, y, p0)

'''
Plot parametric fitting.
'''
def utils_plot_parametric(dtf, zoom=30, figsize=(15,5)):
    ## interval
    dtf["residuals"] = dtf["ts"] - dtf["model"]
    dtf["conf_int_low"] = dtf["forecast"] - 1.96*dtf["residuals"].std()
    dtf["conf_int_up"] = dtf["forecast"] + 1.96*dtf["residuals"].std()
    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=figsize)
```

```
  ## entire series
dtf["ts"].plot(marker=".", linestyle='None', ax=ax[0], title="Parametric Fitting",
color="black")
dtf["model"].plot(ax=ax[0], color="green")
dtf["forecast"].plot(ax=ax[0], grid=True, color="red")
ax[0].fill_between(x=dtf.index, y1=dtf['conf_int_low'], y2=dtf['conf_int_up'],
color='b', alpha=0.3)

  ## focus on last
  first_idx = dtf[pd.notnull(dtf["forecast"])].index[0]
  first_loc = dtf.index.tolist().index(first_idx)
  zoom_idx = dtf.index[first_loc-zoom]
 dtf.loc[zoom_idx:]["ts"].plot(marker=".", linestyle='None', ax=ax[1],
color="black", title="Zoom on the last "+str(zoom)+" observations")
 dtf.loc[zoom_idx:]["model"].plot(ax=ax[1], color="green")
 dtf.loc[zoom_idx:]["forecast"].plot(ax=ax[1], grid=True, color="red")
  ax[1].fill_between(x=dtf.loc[zoom_idx:].index,
y1=dtf.loc[zoom_idx:]['conf_int_low'],
 y2=dtf.loc[zoom_idx:]['conf_int_up'], color='b', alpha=0.3)
  plt.show()
return dtf[["ts","model","residuals","conf_int_low","forecast","conf_int_up"]]

'''
Forecast unknown future.
:parameter
    :param ts: pandas series
    :param f: function
    :param model: list of optim params
    :param pred_ahead: number of observations to forecast (ex. pred_ahead=30)
    :param freq: None or str - 'B' business day, 'D' daily, 'W' weekly, 'M' monthly,
'A' annual, 'Q' quarterly
    :param zoom: for plotting
'''
```

```
def forecast_curve(ts, f, model, start,  pred_ahead=60, freq="D", zoom=30,
figsize=(15,5)):
    ## fit
    X = np.arange(len(ts))
    fitted = f(X, model[0], model[1], model[2])
    dtf = ts.to_frame(name="ts")
    dtf["model"] = fitted

    ## index
    index = pd.date_range(start=start,periods=pred_ahead,freq=freq)
    index = index[1:]

    ## forecast
    Xnew = np.arange(len(ts)+1, len(ts)+1+len(index))
    preds = f(Xnew, model[0], model[1], model[2])
    dtf = dtf.append(pd.DataFrame(data=preds, index=index,
columns=["forecast"]))
    dtf.reset_index(level = 0, inplace = True)

   ## plot
   utils_plot_parametric(dtf, zoom=zoom)
  return dtf

from lmfit import model
preds = forecast_curve(dtf["total"], logistic_f, logistic_model, start='2021-12-28',
pred_ahead=60, freq="D", zoom=30, figsize=(15,5))
preds = forecast_curve(dtf["new"], gaussian_f, gaussian_model, start='2021-12-28',
pred_ahead=60, freq="D", zoom=45, figsize=(15,5))
```