SUMMER TRAINING REPORT

On

Machine Learning

Submitted to Guru Gobind Singh Indraprastha University, Delhi (India)
in partial fulfillment of the requirement for the award of the degree of

B.TECH

in

INFORMATION TECHNOLOGY

Submitted By

ANMOL ASHRI

ROLL No. 40715003117

DEPT. OF INFORMATION TECHNOLOGY

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY,
NEW DELHI-110058

NOVEMBER

ACKNOWLEDGEMENT

A project work owes its success from commencement to completion, to the people in love with developers at various stages. Let me in this page express my gratitude to all those who helped us in various stage of this project. First, I would like to express my sincere gratitude indebtedness to **Dr. Anupama Kaushik** (HOD, Department of Information Technology, Maharaja Surajmal Institute of Technology New Delhi) for allowing me to undergo the summer training of 8 weeks at **Coursera.**

I am grateful to our guide **Ms Jyoti Arora**, for the help provided in completion of the project, which was assigned to me. Without her friendly help and guidance it was difficult to develop this project.

I am also thankful to **Mr Surender Singh** for helping me to prepare the final report and presentation.

Last but not least, I pay my sincere thanks and gratitude to all the mates for their support and for making our training valuable and fruitful.

Submitted By:

ANMOL ASHRI

(40715003117)

CERTIFICATE

Stanford | ONLINE

COURSE CERTIFICATE

Jul 31 2020

Anmol Ashri

has successfully completed

Machine Learning

an online non-credit course authorized by Stanford University and offered through Coursera

EDUCATION FOR EVERYONE

coursera

COURSE CERTIFICATE

Associate Professor Andrew Ng
Computer Science Department
Stanford University

SOME ONLINE COURSES MAY DRAW ON MATERIAL FROM COURSES TAUGHT ON-CAMPUS BUT THEY ARE NOT EQUIVALENT TO ON-CAMPUS COURSES. THIS STATEMENT DOES NOT AFFIRM THAT THIS PARTICIPANT WAS ENROLLED AS A STUDENT AT STANFORD UNIVERSITY IN ANY WAY. IT DOES NOT CONFER A STANFORD UNIVERSITY GRADE, COURSE CREDIT OR DEGREE, AND IT DOES NOT VERIFY THE IDENTITY OF THE PARTICIPANT.

Verify at coursera.org/verify/L2RQM7NHZ3N9

Coursera has confirmed the identity of this individual and their participation in the course.

CANDIDATE’S DECLARATION

I, **ANMOL ASHRI**, Roll No - **40715003117**, B.Tech (Semester- 7^{th}) of the Maharaja Surajmal Institute of Technology, New Delhi hereby declare that the Training Report entitled **“Machine Learning”** is an original work and data provided in the study is authentic to the best of my knowledge. This report has not been submitted to any other Institute for the award of any other degree.

ANMOL ASHRI

(40715003117)

Place: **New Delhi**

Date: **9 November’2020**

Coursera

Coursera was founded by Daphne Koller and Andrew Ng with a vision of providing life-transforming learning experiences to anyone, anywhere. It is now a leading online learning platform for higher education, where 73 million learners from around the world come to learn skills of the future. More than 200 of the world’s top universities and industry educators partner with Coursera to offer courses, Specializations, certificates, and degree programs. Thousands of companies trust the company’s enterprise platform Coursera for Business to transform their talent. Coursera for government equips government employees and citizens with in-demand skills to build a competitive workforce. Coursera for Campus empowers any university to offer high-quality, job-relevant online education to students, alumni, faculty, and staff. Coursera is backed by leading investors that include Kleiner Perkins, New Enterprise Associates, Learn Capital, and SEEK Group.

Coursera works with universities and other organizations to offer online courses, certifications, and degrees in a variety of subjects, such as engineering, data science, machine learning, mathematics, business, financing, computer science, digital marketing, humanities, medicine, biology, social sciences, 4500 plus variety of courses giving students a very broad range of information & experience in different fields.

LIST OF FIGURES

CONTENT

1. INTRODUCTION TO MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning.In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.

1.1 How to Approach Machine Learning ?

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

- Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches have been developed which don't fit neatly into this three-fold categorisation, and sometimes more than one is used by the same machine learning system. For example topic modeling, dimensionality reduction or meta learning.

As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.

1.2 Explanation of Approaches

Types of learning algorithms :The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

1. Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs.The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

2 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated, for example, by *internal compactness*, or the similarity between members of the same cluster, and *separation*, the difference between clusters. Other methods are based on *estimated density* and *graph connectivity*.

3 Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

2. <u>MACHINE LEARNING ALGORITHMS</u>

1. Linear Regression

To understand the working functionality of this algorithm, imagine how you would arrange random logs of wood in increasing order of their weight. There is a catch; however – you cannot weigh each log. You have to guess its weight just by looking at the height and girth of the log (visual analysis) and arrange them using a combination of these visible parameters. This is what linear regression is like.

In this process, a relationship is established between independent and dependent variables by fitting them to a line. This line is known as the regression line and represented by a linear equation Y= a *X + b.

In this equation:

- Y – Dependent Variable
- a – Slope
- X – Independent variable
- b – Intercept

The coefficients a & b are derived by minimizing the sum of the squared difference of distance between data points and the regression line.

2. Logistic Regression

Logistic Regression is used to estimate discrete values (usually binary values like 0/1) from a set of independent variables. It helps predict the probability of an event by fitting data to a logit function. It is also called logit regression.

These methods listed below are often used to help improve logistic regression models:

- include interaction terms
- eliminate features
- regularize techniques
- use a non-linear model

3. Decision Tree

It is one of the most popular machine learning algorithms in use today; this is a supervised learning algorithm that is used for classifying problems. It works well classifying for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets based on the most significant attributes/ independent variables.

4. SVM (Support Vector Machine)

SVM is a method of classification in which you plot raw data as points in an n-dimensional space (where n is the number of features you have). The value of each feature is then tied to a particular coordinate, making it easy to classify the data. Lines called classifiers can be used to split the data and plot them on a graph.

5. Naive Bayes

A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Even if these features are related to each other, a Naive Bayes classifier would consider all of these properties independently when calculating the probability of a particular outcome.

A Naive Bayesian model is easy to build and useful for massive datasets. It's simple and is known to outperform even highly sophisticated classification methods become a Machine Learning Engine

6. KNN (K- Nearest Neighbors)

This algorithm can be applied to both classification and regression problems. Apparently, within the Data Science industry, it's more widely used to solve classification problems. It's a simple algorithm that stores all available cases and classifies any new cases by taking a majority vote of its k neighbors. The case is then assigned to the class with which it has the most in common. A distance function performs this measurement.

KNN can be easily understood by comparing it to real life. For example, if you want information about a person, it makes sense to talk to his or her friends and colleagues!

Things to consider before selecting KNN:

- KNN is computationally expensive
- Variables should be normalized, or else higher range variables can bias the algorithm
- Data still needs to be pre-processed.

7. K-Means

It is an unsupervised algorithm that solves clustering problems. Data sets are classified into a particular number of clusters (let's call that number K) in such a way that all the data points within a cluster are homogenous and heterogeneous from the data in other clusters.

How K-means forms clusters:

- The K-means algorithm picks k number of points, called centroids, for each cluster.
- Each data point forms a cluster with the closest centroids, i.e., K clusters.
- It now creates new centroids based on the existing cluster members.
- With these new centroids, the closest distance for each data point is determined. This process is repeated until the centroids do not change.

8. Random Forest

A collective of decision trees is called a Random Forest. To classify a new object based on its attributes, each tree is classified, and the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is planted & grown as follows:

- If the number of cases in the training set is N, then a sample of N cases is taken at random. This sample will be the training set for growing the tree.
- If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M, and the best split on this m is used to split the node. The value of m is held constant during this process.
- Each tree is grown to the most substantial extent possible. There is no pruning.

9. Dimensionality Reduction Algorithms

In today's world, vast amounts of data are being stored and analyzed by corporates, government agencies, and research organizations. As a data scientist, you know that this raw data contains a lot of information - the challenge is in identifying significant patterns and variables.

Dimensionality reduction algorithms like Decision Tree, Factor Analysis, Missing Value Ratio, and Random Forest can help you find relevant details.

10. Gradient Boosting & AdaBoost

These are boosting algorithms used when massive loads of data have to be handled to make predictions with high accuracy. Boosting is an ensemble learning algorithm that combines the predictive power of several base estimators to improve robustness.

In short, it combines multiple weak or average predictors to build a strong predictor. These boosting algorithms always work well in data science competitions like Kaggle, AV Hackathon, CrowdAnalytix. These are the most preferred machine learning algorithms today. Use them, along with Python and R Codes, to achieve accurate outcomes.

3 <u>ABSTRACT</u>

Covid-19 has been declared as the pandemic and is spreading at an alarming rate. The current study describes the situation outbreak in India. This study mainly focuses on the minimum number of hospital beds required for covid-19 patients along with prediction of total number confirmed covid-19 cases and total predicted deaths due to covid-19 till 30 September 2020 using multi time series forecasting models : Fb Prophet and ARIMA. The performance of the two models are compared on the basis of error metrics :root mean square error(RMSE),mean absolute error(MAE),mean absolute percentage error(MAPE) .The study also uses exploratory data analysis to report the current situation of covid-19 in India .The data used in the study is

taken from datasets available on Kaggle.com and covers up time period till 20 July 2020. All the data visualization, analysis and prediction are made using Python 3 in Jupyter notebook .The outcome of the study will be useful to government and healthcare communities to initiate appropriate measures to control this outbreak in India.

4 INTRODUCTION OF PROJECT

With the initial outbreak of the coronavirus originated from Wuhan in China around December 2019, the virus has spread throughout the world within the next few weeks [1]. The exponential growth of the infection among the world's different countries has presented an unprecedented challenge. The World Health Organisation (WHO) has declared this contagious disease as pandemic [2]. The world is moving towards a very stressful stage with the spread of novel coronavirus. Different countries in the world are suffering from this pandemic and causing disaster to the public health system and bringing enormous disruption to the economy and society [3]. With the prevailing circumstances, it has become significant for the healthcare professionals and policymakers to plan and adequately equipped for the conditions that may arise due to the rapid spread of the COVID-19.

Countries are using various Information, Communication, and Technological (ICT) aspects to control the situation. Different ICT tools, applications, and services are used to predict future conditions and plan accordingly [4]. Artificial Intelligence (AI) can play a significant role in fighting against the pandemic situation. Different Time-series AI models can be used for estimating and predicting the spread rate. A Laplacian based model has been developed to study the transmission of Pine Wilt disease [5]. A mathematical model for analyzing and understanding human liver behaviour and the spread of dengue was predicted [6]. A model was designed using SIRS [7] to resist the spread of the syncytial virus in infants.

Every infectious disease epidemic reveals specific patterns and such patterns needed to be studied based on transmission dynamics of such outbreaks. Paramount methods to eliminate such contagious infections rely on the processes to be followed to predict the future of such diseases. India is a vast country with 1.38 billion populations. Each state of this country is at a different stage of the epidemic. There is massive pressure on the administration and health care professionals for accommodating different patients with the symptoms of coronavirus. It is the essential requirement to understand the growth pattern of the virus; as such information will help to handle the upcoming situation. This information will play a crucial role in planning in advance to overcome the situation of panic. So, in order to facilitate such information timely, time-series data-driven prediction models must be used to predict the number of cases for the aid of future health planning [3].

The data is growing and changing dynamically, so using statistical and epidemiological methods will result in uncertainty [8,9]. The researchers have used various deep learning and machine learning-based models to predict the number of cases of COVID-19 infections [10] in order to facilitate in advance with the requirements such as extensive testing, number of beds in the hospital and others to control the spread and reduce the mortality rate.

COVID-19 is a time-series data set which contains both non-linear and non-stationary patterns, and it is highly recommended to use the model which can extract the design using sequential networks. In this paper, we have used multi time series models such as Fb Prophet Model, ARIMA Model for prediction of the COVID-19 total number of cases in India in up to 30th September 2020. These models will allow dealing with the data with different veracity as making a decision based on an individual model would be critical. With the help of these models we have tried to predict a total number of COVID-19 cases in India, the total number of mortality due to the significant COVID-19 outbreak in India and minimum whole numbers of hospital beds would be needed for COVID-19 patients in India in order to control the further outbreak of this epidemic in India. In the absence of a permanent cure of COVID-19, these approximations will provide an intuition into the resource allocation to keep this epidemic under control. Besides focusing on the dynamics of the COVID-19 outbreak, the practical purpose of this analysis will provide government officials with realistic approximations which can be helpful in deriving policy making to control the magnitude of this epidemic. Such assessment of mortality due to COVID-19 and number of required hospital beds for COVID-19 patients will help to anticipate the extent of this epidemic in India. They will provide some critical information for planning the health care system in India to control this epidemic.

5 PREDICTION MODELS

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 analysing past observations of random variables to develop a model that captures the underlying relationship and its pattern. This approach is usually beneficial in the case when there is no knowledge or little knowledge on the data generating process. Over past decades, a lot of research has produced a great development and improvement of time series forecasting models.

In this section, the time-series models that have been used for predictions are detailed as:-

5.1 Auto-Regressive Integrated Moving Average Model (ARIMA)

One of the most well-known and widely used families of time series models include the Auto Regressive Integrated Moving Average (ARIMA) model. This model is a well known implementation of Box-Jenkin methodology [11]. It is a model for time-series data analysis, based on a three-class component to manage stationary and non-stationary time datasets.

The model is generally denoted by ARIMA (p, d, q), where the parameter p and q are the order of the Auto-Regressive (AR) and Moving Average (MA) models and d is the level of differencing. The first component is AR(p) of time series model deals with the dependent observation for predicting future events. The Second component MA(q) handles different white noise annotations representing past forecast errors for the estimation of the outcomes depending on the future value. This combination allows the ARIMA model to deal with the time-series non-stationary values. Both combinations will make an ARMA model. But as in the mean of the observed data, the variance is not constant, so the third component is used as Integrating (I(d)) to convert the observations using differencing series [12].

Further, the model can be mathematically expressed as:-

$$y_t^{'} = \mu + \delta_1 y_{t-1}^{'} + \delta_2 y_{t-2}^{'} + \dots\dots\dots + \delta_p y_{t-p}^{'} + \vartheta_1 y_{t-1} + \vartheta_2 y_{t-2} + \dots\dots\dots + \vartheta_q y_{t-q} + \varepsilon_t \qquad (1)$$

where $y_t^{'}$ denotes the actual value of the variable under consideration at time t , y_{t-1} is the random error at time t . The δ_i and ϑ_j are coefficients of the ARIMA model. Building an ARIMA model involves generally three iterative steps : Achieving stationarity (A model that shows stationarity is one that shows there is constancy to the data over time) ;Autocorrelation function (ACF) and the partial autocorrelation function (PACF) plots are used for parameters (p and q) estimation and model diagnostics are checked to find the 'best' fitted forecasting model using Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC)[6].

5.2 Facebook Prophet Model

Prophet is an open-source tool from Facebook initially developed for forecasting time series data which is 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 of time and it filters out the seasonal variations. Seasonality is the variation that occurs over a short period of time and is not prominent enough to be called a "trend".

Mathematically, Prophet model is represented as:

$$y(t) = g(t) + s(t) + h(t) + e(t) \qquad (2)$$

where:

- $g(t)$ refers to trend (changes over a long period of time).
- $s(t)$ refers to seasonality (periodic or short term changes).
- $h(t)$ refers to effects of holidays to the forecast.
- $e(t)$ refers to the unconditional changes that is specific to a business or a person or a circumstance. It is also called the error term.
- $y(t)$ is the forecast.

Fundamentally it is a regression model with explicable parameters that often work well with their default values. It also allows user to choose the components concerning their forecasting and give better results based on necessary adjustments [14]. To forecast trend prophet use two models : Logistic growth model and other one is piecewise linear model .In the case in which data to be forecasted has saturating and non-linear data (grows non-linearly) and after reaching the saturation point shows little to growth or shrink and only exhibits some seasonal changes then logistic growth model is best option. If the data shows linear properties and growth or shrink trends in the past then a piecewise linear model is a better choice.

The logistic growth model is fit under following statistical equation:-

$$g(t) = C \big/ \left(1 + e^{-k(t-m)}\right) \tag{3}$$

where

- C is the carry capacity
- k is the growth rate
- m is an offset parameter

Piecewise linear model is fit using the following statistical equations :-

$y = \beta + \beta_1 * x, x \leq c;$
(4)

$$y = \beta - \beta_2 * c + (\beta_1 + \beta_2) * x, x > c$$
(5)

where c is the trend change point and β_1 and β_2 are dummy variables in (4) and (5).To find seasonality it depends on Fourier series to provide a flexible model of periodic effect [15] and to account holiday a predefined data of past and future holidays events is required, holiday effects are assumed to be independent so assimilating them to the model becomes insignificant[14] .

6 Dataset

In this paper, the prediction is done for calculating the total number of confirmed cases, death cases and minimum number of hospital beds required by the patients to fight against COVID-19. All these datasets are collected from the starting date of the epidemic in India to 20/07/2020. We have explored two datasets for the prediction and visualization of the COVID-19 outbreak.

6.1 Novel Coronavirus 2019 dataset

Due to the long duration of COVID-19 pandemic, the data is continuously changing and collected periodically by researchers at John Hopkins University and made publically available on [16]. Dataset contains the data from date - 20/1/2020. This includes the number of confirmed cases, number of cured cases and morality rate which is maintained on a time-series basis. The data is provided with the country name, code, city/province, sex, age, reporting date of a confirmed COVID-19 patients around the world.

6.2 COVID-19 in India Dataset

The dataset is extracted and made from the data provided by covid19india.org for the individual level details and testing details in India and rest of data is provided by Indian Ministry of Health & Family Welfare [17]. Dataset contains the data from date 30/1/2020 on this date the first confirmed case of COVID-19 was reported in India from Kerala. The dataset contains time-series data including count of confirmed cases, count for cured cases and death rate of India as well as individual states. This dataset also provides us individual testing details such as reported date, age, gender, detected city and detected state. It also provides the number of COVID-19 daily tests done by each state in India and number of running COVID-19 labs headed by ICMR (Indian Council of Medical Research). Dataset also provides data on the number of hospital beds in each state and data on medical infrastructure of each state. Additionally it also provides data on population at a state level.

We have used this dataset for our COVID-19 visualization, forecasting and prediction of total number of COVID-19 cases in India with the prediction of total number of deaths and predicted number of hospital beds required in India to tackle the COVID-19.

7 Visualization and Experimental Results

7.1 Visualization of COVID-19

To better understand the current position of the COVID-19 in India, this section explores the visualization of the pandemic in India from the data taken in consideration. Data is visualised on the basis of different criteria such as number of total cases, number of total recovered cases and number of total deaths occurred from COVID-19 spread in India. The section covers different aspects of pandemic to understand the spread in a better way.We have used Python 3.0 programming language to show all visualization with the help of Jupyter Notebook which is an open source web application used by us to run all Python code.

7.1.1 Rate of spread of COVID-19 in India

First case of COVID-19 in India was reported on 30/1/2020 in Kerala and from that day the number of COVID-19 cases is increasing gradually in India. We have used the COVID-19 India data till 20/7/2020. Number of total confirmed COVID-19 cases in India till 20/7/2020 is 11,55,338 ; total number of recovered COVID-19 cases in India till 20/7/2020 is 7,24,578 and total number of deaths due to COVID-19 outbreak in India till 20/7/2020 is 28,082 as shown in figure 1.

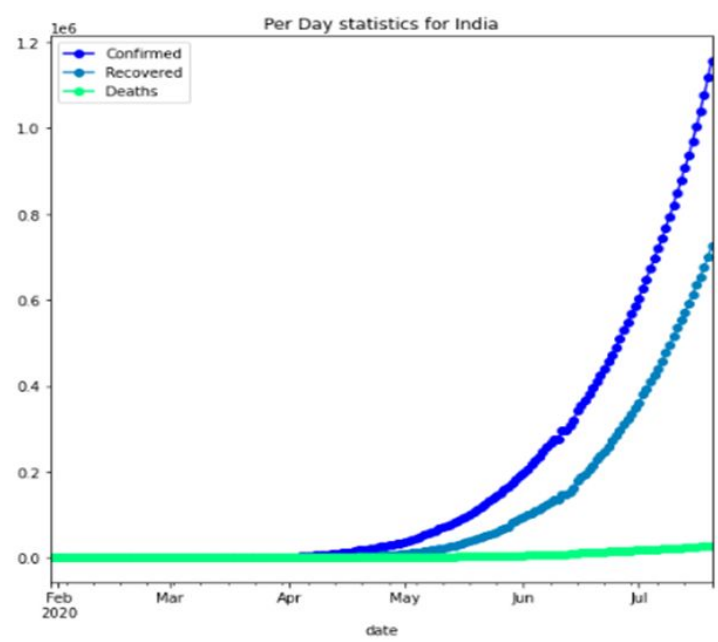
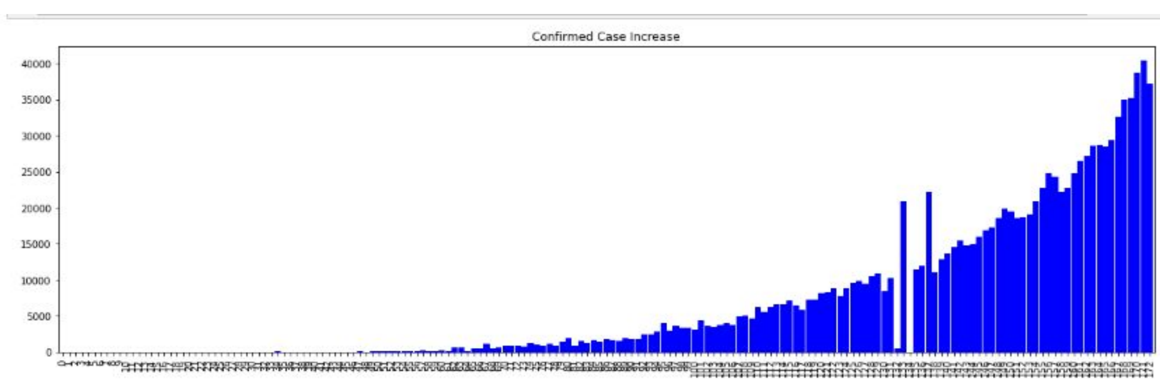


Figure 1. Rate of spread of COVID-19 in India

The effect of pandemic in different States of India is represented in Figure 4. It can be evaluated that Maharashtra leads in the total number of confirmed cases (3,18,695) among all the states in India. It also leads in the total number of deaths(12,030) due to the current outbreak. Tamil Nadu and Delhi are the states after Maharashtra in number of confirmed cases. Delhi leads in the recovery rate(total number of recovered COVID-19 cases /total number of COVID-19 cases in the state) among all the states in India with a recovery rate of 0.85 followed by Ladakh which has a recovery rate of 0.84. Maharashtra, Gujarat, Telangana have an equal death rate(total number of total deaths due to COVID-19 in state/total number of confirmed COVID-19 cases in the state) of 0.04 which is highest among other states and greater than India's death rate of 0.02.The numbers of confirmed cases are increasing rapidly. Figure 2 shows the rate of growth of number of confirmed cases day wise. Figure 3 shows the number of deaths occurring due to the pandemic each day. From the visualization, it could be analyzed that there is a rapid rise in daily number of confirmed cases in the month of July as on 19/7/2020 India reported 40425 cases which are highest till 20/7/2020 in India. Highest COVID-19 single day deaths was reported on day 50 which is equal to 2000 as shown in above bar plot Figure 3.Y-axis of bar charts shown in figure 2 and figure 3 shows numerical values of cases and deaths; x-axis shows each day(time period) in both the bar charts.

Figure 2.Number of Confirmed COVID-19 per day in India till 20/7/2020



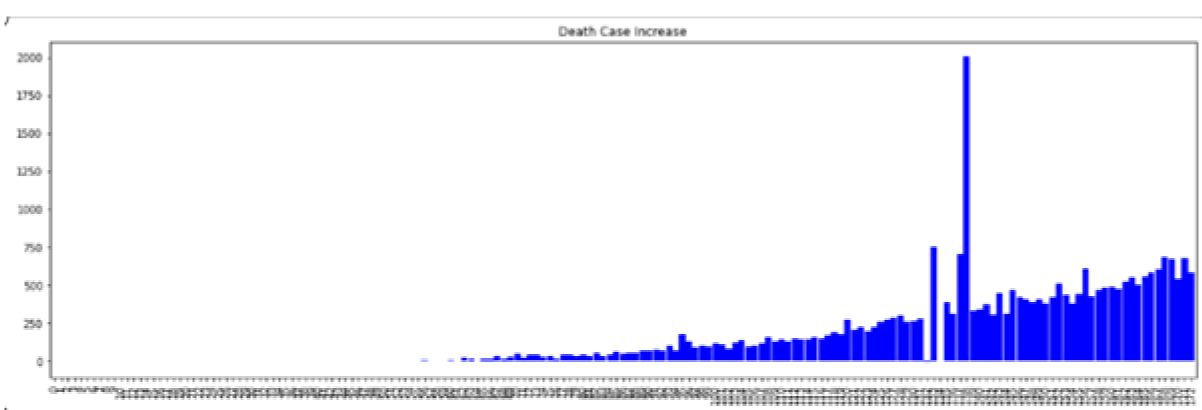


Figure 3.Number of new deaths due to COVID-19 per day in India till 20/7/2020

State	Confirmed	Deaths	Recovered	Recovery Rate	Death Rate
Maharashtra	318695	12030	175029	0.55	0.04
Tamil Nadu	175678	2551	121776	0.69	0.01
Delhi	123747	3663	104918	0.85	0.03
Karnataka	67420	1403	23795	0.35	0.02
Andhra Pradesh	53724	696	24228	0.45	0.01
Uttar Pradesh	51160	1192	30831	0.6	0.02
Gujarat	49353	2162	35678	0.72	0.04
Telangana	46274	422	34323	0.74	0.01
West Bengal	44769	1147	26418	0.59	0.03
Rajasthan	30390	568	22195	0.73	0.02
Bihar	27646	217	17433	0.63	0.01
Haryana	26858	355	20226	0.75	0.01
Assam	25382	58	17095	0.67	0
Madhya Pradesh	23310	738	15684	0.67	0.03
Odisha	18110	97	12910	0.71	0.01
Jammu and Kashmir	14650	254	8274	0.56	0.02
Kerala	13274	43	5616	0.42	0
Punjab	10510	262	7118	0.68	0.02
Jharkhand	5756	53	2810	0.49	0.01
Chhattisgarh	5561	25	3944	0.71	0
Uttarakhand	4642	55	3212	0.69	0.01
Telengana	4111	156	1817	0.44	0.04
Goa	3853	23	2361	0.61	0.01
Tripura	3079	7	1845	0.6	0
Puducherry	2092	29	1265	0.6	0.01
Manipur	1925	0	1307	0.68	0
Himachal Pradesh	1631	11	1067	0.65	0.01
Ladakh	1195	2	1007	0.84	0
Nagaland	1021	0	484	0.47	0
Arunachal Pradesh	790	3	265	0.36	0
Chandigarh	737	12	518	0.7	0.02
Dadra and Nagar Haveli and Daman and Diu	684	2	448	0.65	0
Meghalaya	466	4	66	0.14	0.01
Sikkim	305	0	92	0.3	0
Mizoram	297	0	168	0.57	0
Andaman and Nicobar Islands	207	0	152	0.73	0
Dadar Nagar Haveli	26	0	2	0.08	0
Daman & Diu	2	0	0	0	0

Figure 4. State Wise Covid-19 Trend in India

7.1.2 Healthcare facilities in India

With the outbreak of pandemic in every state of the country crossing different stages, it has become essential for the healthcare policy makers to conduct proper test among the population of different States. It will help to control the epidemic timely and helps to insure the real scenario. Till 20/7/2020 India has done about 1,43,81,303 COVID-19 tests and in terms of testing India stands at fourth position[18] among all countries affected. Figure 5 shows, COVID-19 testing done by each state.

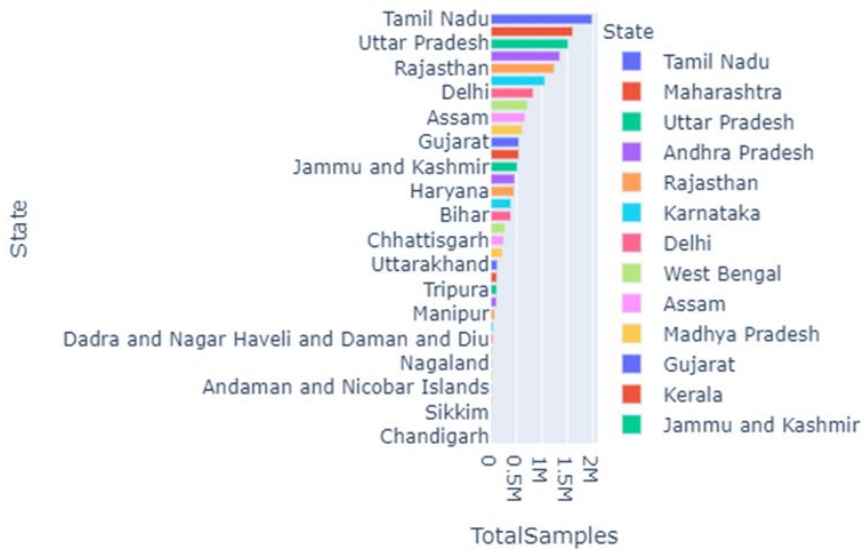


Figure 5 . Distribution of Tests conducted in each state

Figure 5 shows that Tamil Nadu leads among all the states in terms of COVID-19 testing with 19,84,579 followed by Maharashtra (16,03,802) and Uttar Pradesh being at third position with 15,13,827 number of tests.India has a diverse healthcare system with both public and private health sectors playing essential roles. Accommodation of inundation of COVID-19 patients, will require rapid augmentation of current magnitude or modifications in the health care system of India. India has 37,725 [19] number of hospital facilities in the public sector and 7,39,024 [19] number of hospital beds in the public sector.Figure 6 and Figure 7 represent the number of hospital beds in terms of top 5 states in health facilities and top 5 states in terms of number of hospital beds. The data has been shown for the number of beds of top-5 states in Figure 7 on the categorization of health facilities in rural areas, urban areas and public areas.Representation of hospital beds is divided in Rural beds, Urban beds and Public Beds. Tamil Nadu leads among the quantity of rural beds: 40179 followed by Uttar Pradesh:39104.West Bengal leads in number of Urban Beds : 58882 beds followed by karnataka : 49093 and Tamil Nadu again leads in number of public beds among all states, with 72616 number of public beds followed by Maharashtra with 68998 number of public beds.

Figure 6. Distribution of Medical Facilities

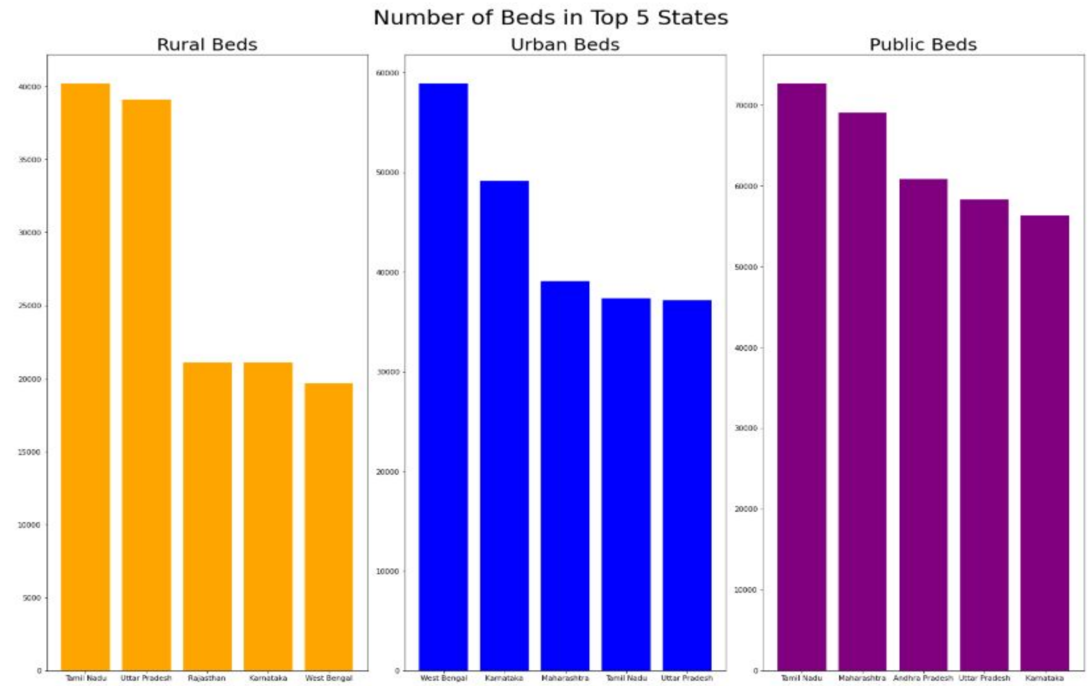


Figure 7. Number of Beds in Top-5 states

7.2 Experimental Results and Analysis

In this section, prediction of COVID-19 for total number of confirmed cases, minimum number of hospital beds required for the COVID-19 patients in India and number of mortality due to COVID-19 outbreak in India till 30/9/2020. The experiments are performed on the data [16] from 30/1/2020 to 20/7/2020 using ARIMA and Fb Prophet model.
The accuracy of the models is calculated using root mean square error (RMSE)[21], mean absolute error (MAE)[21] and mean absolute percentage error (MAPE)[22].

7.2.1 Prediction for total number of Confirmed cases in India

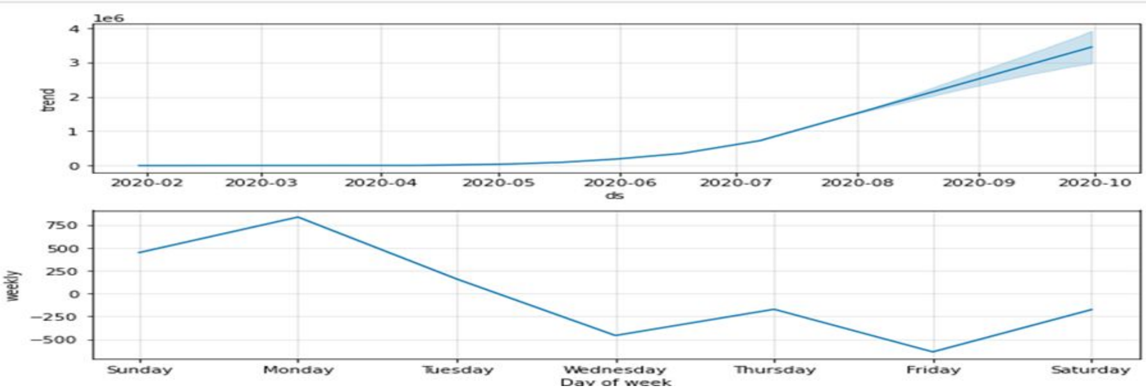
Firstly, prediction is done using the Fb Prophet model. Table 1 shows the results of the prediction carried out for 21/9/2020 to 30/9/2020 as date timestamp. We have also added change points (These are important points to identify the period where the data set trend changes).

Date Timestamp	**Actual Prediction**	**Lower Approximation**	**Upper Approximation**
2020-09-21	3156649	2725163	3563744
2020-09-22	3187985	2741739	3605418

2020-09-23	3219360	2757522	3648972
2020-09-24	3251650	2774195	3693603
2020-09-25	3283187	2796353	3726838
2020-09-26	3315652	2815485	3771656
2020-09-27	3348278	2837339	3814664
2020-09-28	3380669	2867271	3862418
2020-09-29	3411995	2887571	3909353
2020-09-30	3443381	2905718	3956548

Table 1. Predictions from 21/9/2020 to 30//9/2020 using Fb Prophet model

The actual prediction shows that the number of confirmed COVID-19 cases in India based on the trend calculated by the model by 30/9/2020 would be 34,43,381 as exact predictions while 29,05,718 lowest value of prediction and 39,56,548 maximum value of prediction.The Figure 9 indicates the relationship of the total confirmed COVID-19 cases in India with time where the x-axis shows time and y-axis shows the numerical value of confirmed COVID-19 cases. The black dotted line indicates original values and the blue solid line indicate the predicted values of



the COVID-19 confirmed cases of India.

Figure 8 .Representation of Trend and weekly analysis

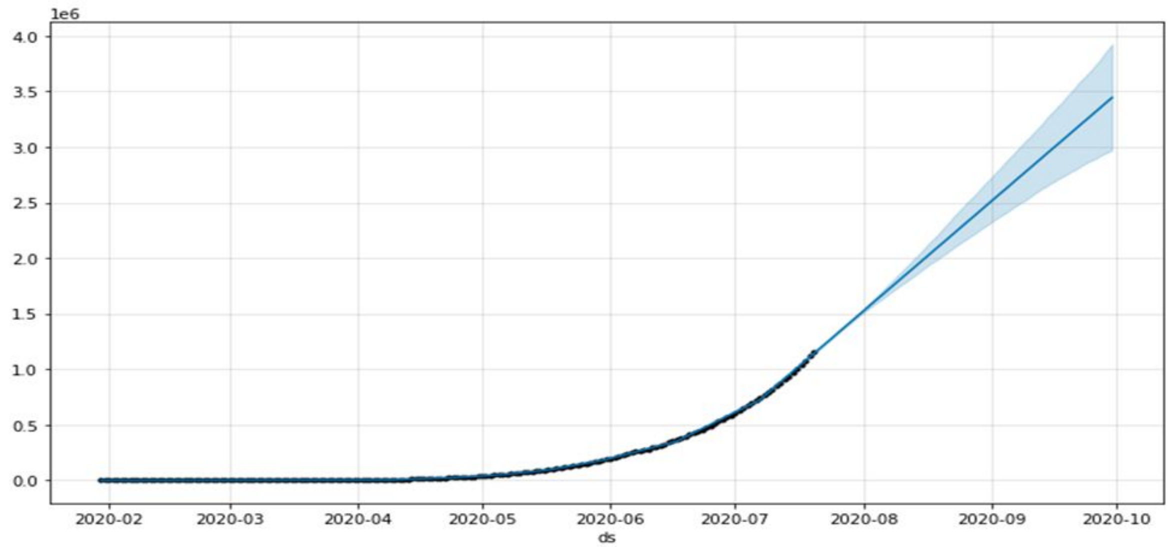


Figure 9 .Prediction of total numbers of confirmed cases in India using FB prophet

Figure 8 above indicates the trend of confirmed cases and weekly analysis of COVID-19 cases till 30/9/2020. The weekly analysis represents the number of cases that are maximum on Monday and minimum on Friday.Secondly, prediction was done on the basis of the ARIMA time series model.Autocorrelation function (ACF) graph and partial autocorrelation (PACF) as shown in figure 10(b) were used to estimate the parameters of ARIMA model. Through experimentation, the best fitted forecasting model using Akaike Information Criterion (AIC) and the Bayesian Information Criterion(BIC) , the values for the parameters that ARIMA(2,2,4) is best fitted for which both AIC and BIC are lowest(AIC=2977.789 ,BIC=3002.685) and it's model summary shown in table 2. Using this we have predicted the total number of confirmed COVID-19 cases in India upto 30/9/2020 shown in figure 10 which includes graphical representation with 95 % CI(Confidence Interval). For better visualization we have shown prediction using a python library called matplotlib, as shown in figure 11.

(a) (b)

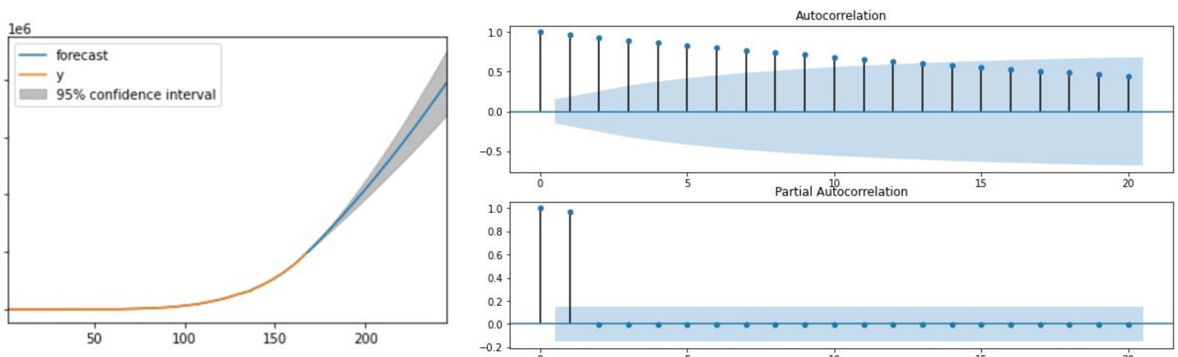
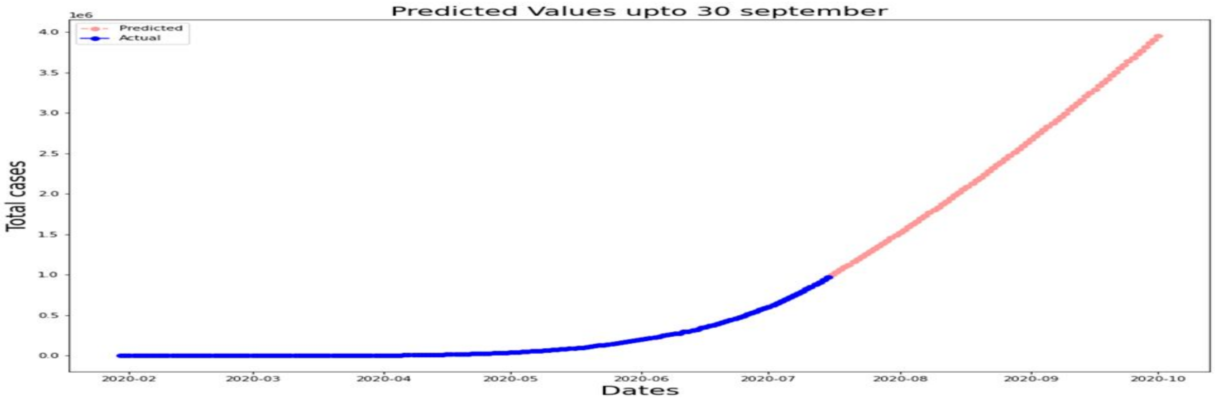


Figure 10. (a) Prediction of COVID-19 confirmed cases using ARIMA model (b) ACF



and PACF plots.\

Figure 11. Prediction of COVID-19 confirmed cases using ARIMA till 30/9/2020

From figure 11, it can be predicted that the exact number of COVID-19 cases based on the data used are 39,53,681 with 33,91,791 lower limit value and 45,15,571 higher limit value.To evaluate the performance of the models, we have split our data into train set which contain the data till 15/7/2020 and a test set which contains data from 16/7/2020 to 20/7/2020 is used to evaluate error metrics of our model. We have calculated error metrics from 16/7/2020 to 20/7/2020 from which we have found lowest RMSE, MAE, MAPE on 16/7/2020 shown in table 3.

ARIMA(2,2,4)	**Std error**	**z**	**P>|z|**
AR(1)	0.356	-3.064	0.002
AR(2)	0.301	-1.433	0.152
MA(1)	0.346	0.427	0.669
MA(2)	0.109	-1.795	0.073
MA(3)	0.183	-1.508	0.131
MA(4)	0.090	3.413	0.001

Table 2. Model Summary ARIMA(2,2,4)

Serial Number	**MODEL**	**RMSE**	**MAE**	**MAPE**
1	ARIMA	4885	4885	0.00487
2	FB Prophet	4388.174590	4388.174590	0.004371

Table 3. Evaluation of the Models on the basis of total number of confirmed cases in India

7.2.2 Prediction for total number of hospital beds required

COVID-19 has put humongous pressure on the medical infrastructure of India. As the cases are increasing, demand for hospital beds are also increasing with it. According to WHO not all COVID-19 patients need to be hospitalized, only 20 percent of them need special treatment or hospitalized so based on this we have predicted the minimum number of beds required for COVID-19 patients to fight against the on-going pandemic [20]. Nevertheless more investigation as factors like individual patients health report ,age distribution of patients are not taken into consideration in this prediction.We have calculated this by calculating total active COVID-19 cases in India and it is assumed 20 percent from active cases need hospital support according to WHO [1]. At first we showed predictions using the FB prophet Model . Below table which shows the prediction of minimum number of hospital beds in India from 21/9/2020 to 30/9/2020. Table 4 shows that the number of confirmed COVID-19 cases in India based on the trend calculated by the model by 30/9/2020 would be 2,29,523 as exact predictions while 1,91,673 lowest value of prediction and 2,67,016 maximum value of prediction.

Date Timestamp	**Actual Prediction**	**Lower Approximation**	**Upper Approximation**
2020-09-21	210766	179735	240597
2020-09-22	212750	181101	243559
2020-09-23	214815	181770	245975
2020-09-24	216903	183403	248237
2020-09-25	218988	184311	251111
2020-09-26	221133	185778	254708
2020-09-27	223360	187705	257161
2020-09-28	225474	189501	260878
2020-09-29	227458	190530	263879
2020-09-30	229523	191673	267016

Table 4. Predictions from 21/9/2020 to 30//9/2020 using Fb Prophet model

The figure 12 indicates the relationship of the minimum number of hospital beds with time where the x-axis shows time and y-axis shows the numerical value of minimum number of hospital beds. The black dotted line indicates original values and the blue solid line indicates the predicted values of minimum number of hospital beds. Almost the predicted and the original values are going in the same way as shown in the figure.

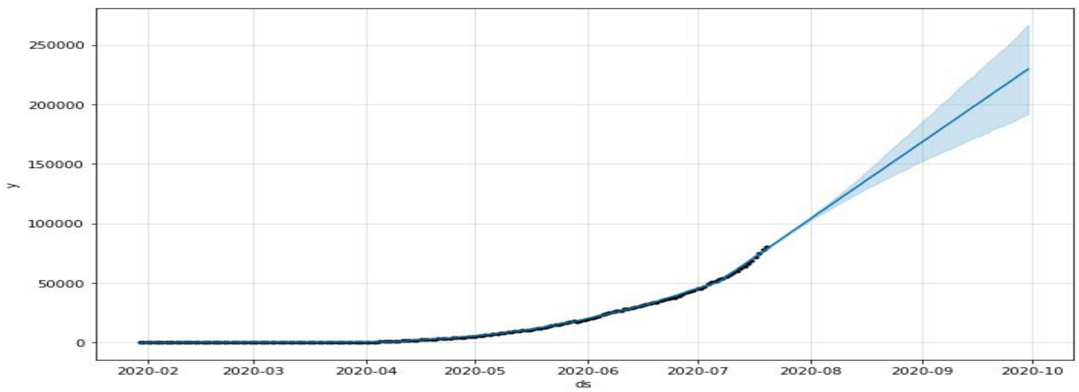


Figure 12. Prediction of minimum number of hospital beds using Fb Prophet model

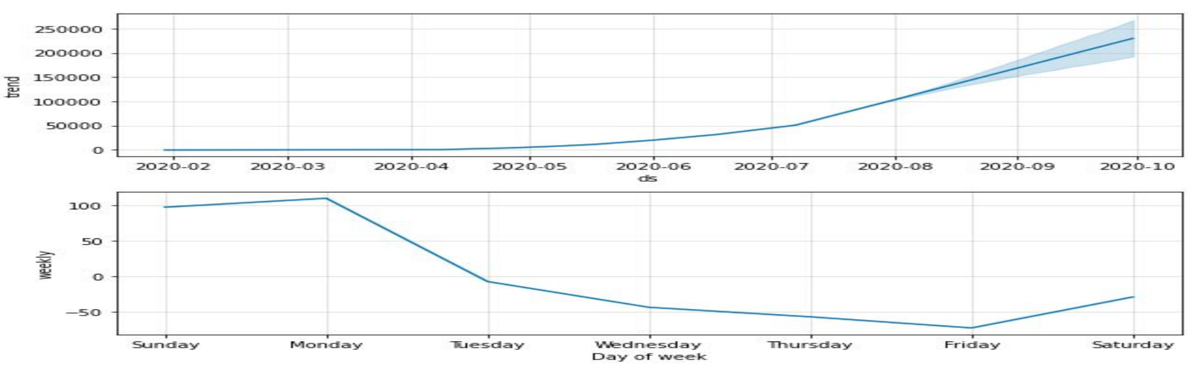


Figure 13. Representation of trend and weekly analysis

Figure 13 indicates the trend of our prediction for the number of hospital beds and weekly analysis of the number of hospital beds till 30/9/2020. The weekly analysis was maximum on Monday, then the weekly analysis decreased till Friday then it increased again.Secondly, we have predicted the number of hospital beds, using the ARIMA time series model. Autocorrelation function (ACF) graph and partial autocorrelation (PACF) as shown in figure 14(b) were used to estimate the parameters of ARIMA model.Through experimentation,the best fitted forecasting model using Akaike Information Criterion (AIC) and the Bayesian Information Criterion(BIC) , we have found by the experimentation that ARIMA(3,2,0) is best fitted for which both AIC and BIC are lowest(AIC=2331.87 ,BIC=2347.43) and it's model summary shown in table 5.Using this we have predicted minimum number of hospital beds required for

COVID-19 patients in India upto 30/9/2020 shown in figure 14(a) which includes graphical representation with 95 % CI(Confidence Interval).For better visualization we have shown prediction using python library called matplotlib shown in figure 15.

(a) (b)

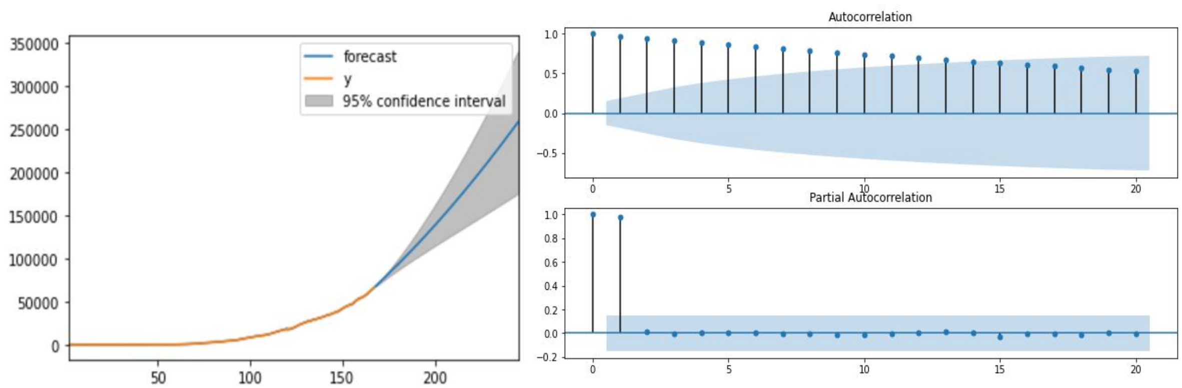


Figure 14. (a) Prediction of minimum number of hospital beds left side in which y shows actual number of required hospital beds for COVID-19 patients in India (b) ACF and PACF plots.

We can evaluate from figure 15 that the exact number of predicted cases based on the data used are 2,58,175 with 1,75,344 lower limit value and 3,41,006 higher limit value. To validate the accuracy of our models we have split our data into train set which contain the data till 15/7/2020 and test set which which contain data from 16/7/2020 to 20/7/2020 are used to evaluate error metrics of our model. We have calculated error metrics from 16/7/2020 to 20/7/2020 from which we have found lowest rmse, mae, mape on 16/7/2020 shown in table 6.

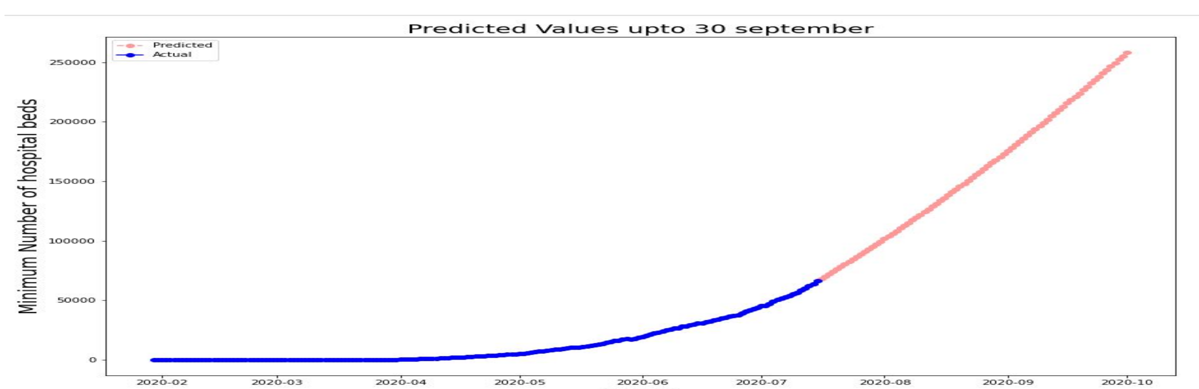


Figure 15. Prediction of minimum number of covid-19 beds using ARIMA till 30/9/2020

ARIMA(3,2,0)	**Std error**	**z**	**P>|z|**
AR(1)	0.070	-11.492	0.000
AR(2)	0.095	-4.745	0.000
AR(3)	0.075	-3.055	0.002

Table 5 Model Summary of ARIMA(3,2,0)

Serial Number	**MODEL**	**RMSE**	**MAE**	**MAPE**
1	ARIMA	366	366	0.00536
2	FB Prophet	883.949967	883.949967	0.012905

Table 6 Evaluation of the Models on the basis of minimum number of hospital beds for covid-19 patienrs in India

7.2.3 Total number of mortality due to COVID-19 outbreak in India

Prediction of mortality of any epidemic is the most important process as in the absence of permanent cure of an epidemic estimating mortality becomes one of the crucial steps. Same lies with COVID-19 as till now there is no permanent cure of COVID-19 in the world so prediction of mortality due to COVID-19 will help the government to control it. We have predicted mortality due to COVID-19 till 30/9/2020 in India with the help of time series model. At first we showed predictions using the FB prophet Model . Below table which shows the prediction of mortality due to COVID-19 outbreak in India from 21/9/2020 to 30/9/2020 ds indicates DateTime stamp, yhat shows the exact prediction, yhat_lower shows lowest value of prediction while yhat_upper shows maximum value of prediction. Table 7 shows that the number of confirmed COVID-19 cases in India based on the trend calculated by the model by 30/9/2020 would be 68116 as exact predictions while 60,405 lowest value of prediction and 76,0234 maximum value of prediction.

Date Timestamp	**Actual Prediction**	**Lower Approximation**	**Upper Approximation**
2020-09-21	63058	56904	69375
2020-09-22	63670	57378	70245
2020-09-23	63670	57616	70837
2020-09-24	64771	58129	71565
2020-09-25	65309	58478	72263
2020-09-26	65863	58867	73066

2020-09-27	66416	59419	74518
2020-09-28	66971	59483	74518
2020-09-29	67584	59943	75307
2020-09-30	68116	60406	76024

Table 7. Predictions from 21/9/2020 to 30//9/2020 using Fb Prophet model

The figure 16 indicates the relationship of the mortality due to COVID-19 outbreak in India with time where the x-axis shows time and y-axis shows the numerical value of mortality due to COVID-19 .The black dotted line indicates original values and the blue solid line indicate the predicted values of mortality due to COVID-19 in India.Almost the predicted and the original values are going in the same way as we have seen in figure 16.

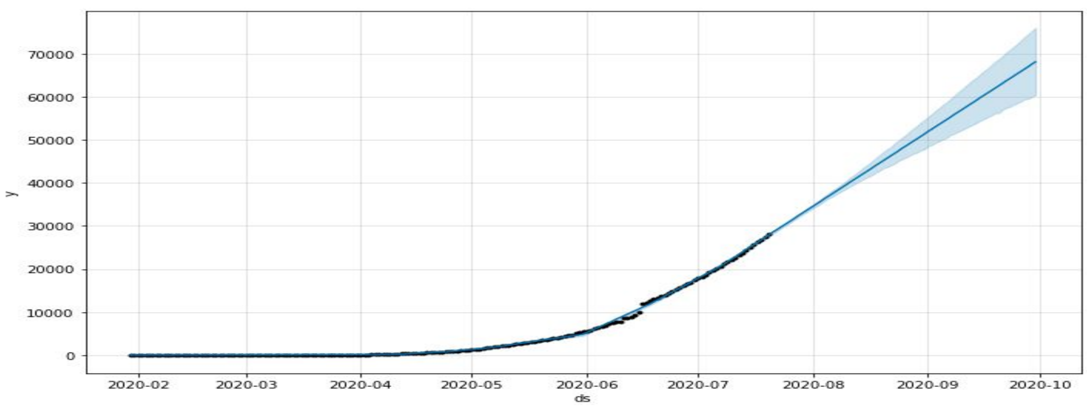


Figure 16. Prediction of total number of mortality due to COVID-19 in India using FB Prophet Model till 30/9/2020

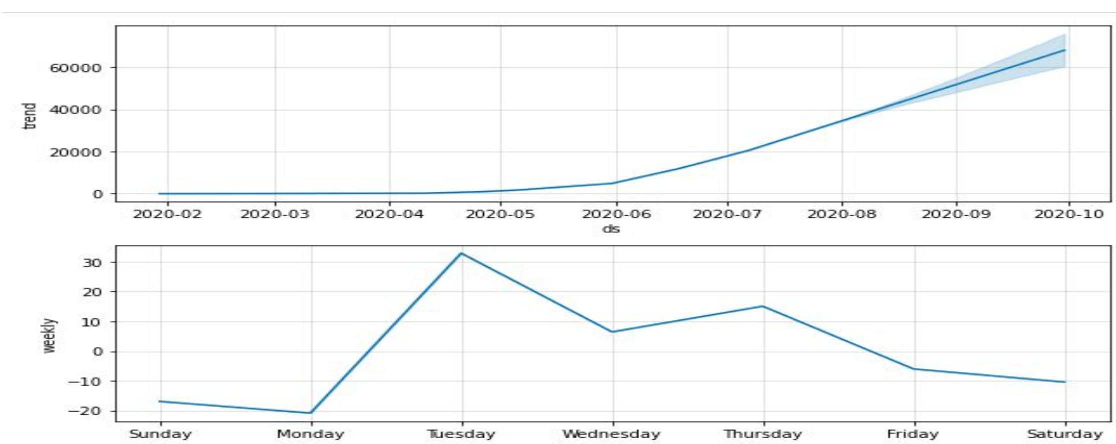


Figure 17. Representation of trend and weekly analysis

Figure 17 indicates the trend of our prediction of the mortality due to COVID-19 outbreak in India and weekly analysis of mortality due to COVID-19 in India till 30/9/2020. The weekly analysis was maximum on Tuesday then the weekly analysis was decreasing(except on Thursday) and reached minimum on Monday as we have seen in the figure 17.Next we have shown prediction using the ARIMA time series model. Autocorrelation function (ACF) graph and partial autocorrelation (PACF) as shown in figure 18(b) were used to estimate the parameters of ARIMA model.Through experimentation, the best fitted forecasting model using Akaike Information Criterion (AIC) and the Bayesian Information Criterion(BIC) , we found that ARIMA(3,2,1) is best fitted for which both AIC and BIC are lowest(AIC=2145.003 ,BIC=2163.675) and it's model summary shown in table 8.Using this we have predicted total number of mortality due to COVID-19 in India upto 30/9/2020 shown in figure 18(a) which includes graphical representation with 95 % CI(Confidence Interval).For better visualization we have shown prediction using python library called matplotlib shown in figure 19.

(a) (b)

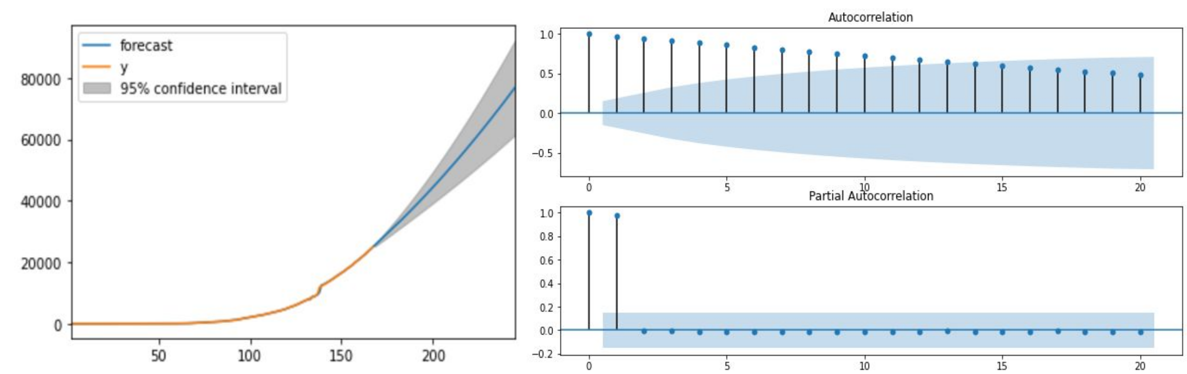


Figure 18. (a) Prediction of mortality due to COVID-19 left side in which y shows actual number of mortality due to COVID-19 in India (b) ACF and PACF plots.

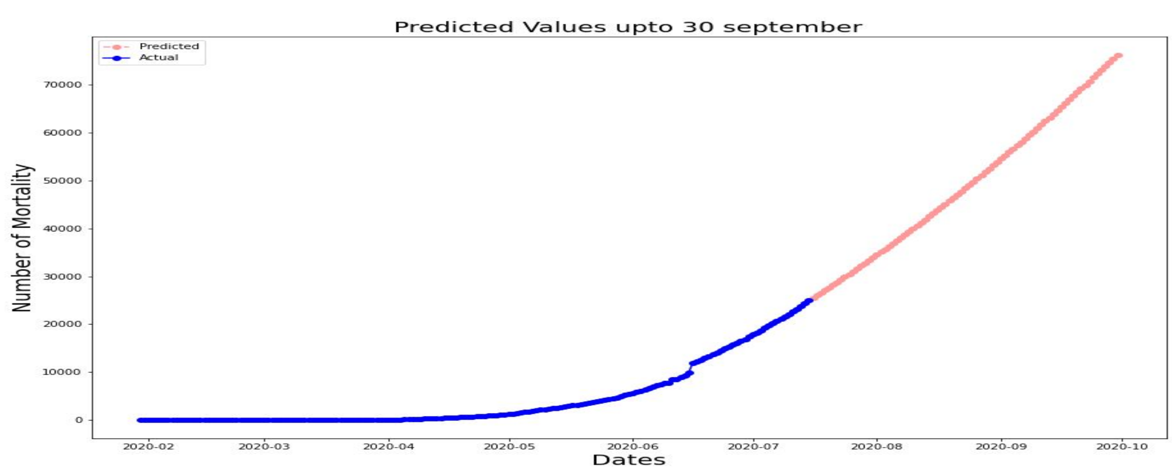


Figure 19. Prediction of mortality due to COVID-19 in India ARIMA till 30/9/2020

We can evaluate from figure 19 that the exact number of predicted cases based on the data used are 76,190 with 61,540 lower limit value and 92,435 higher limit value.To validate the accuracy of our models we have split our data into train set which contains the data till 15/7/2020 and test set which which contain data from 16/7/2020 to 20/7/2020 is used to evaluate error metrics of our models.We have calculated error metrics from 16/7/2020 to 20/7/2020 from which we have found lowest rmse,mae,mape on 16/7/2020 shown in table 9.

ARIMA(3,2,1)	**Std error**	**z**	**P>|z|**
AR(1)	0.088	0.731	0.465
AR(2)	0.084	0.096	0.923
AR(3)	0.083	-0.201	0.840
MA(1)	0.042	-21.154	0.000

Table 8 Model Summary ARIMA(3,2,1)

Serial Number	**MODEL**	**RMSE**	**MAE**	**MAPE**
1	ARIMA	146	146	0.00571
2	FB Prophet	250.259984	203.126350	0.008234

Table 9 Evaluation of the Models on the basis of mortality due to COVID-19 in India

8 CONCLUSION

In this project, we focused on presenting visualization of the current breakdown of the pandemic in India. Time-series prediction models such as FB Prophet model and ARIMA model are used to predict and analyze the total number of COVID-19 cases in India till 30/9/2020. The predicted results are used to estimate the minimum number of hospital beds required for the patients. The results of the two models are compared and through our investigation based on the experimentation, we identified that ARIMA model performs better than FB Prophet model except in prediction of total number of COVID-19 cases. These predictions clearly show that COVID-19 in India is going to increase at an alarming rate in the months of August and September. This study will allow timely decision by the health authorities and make the government to efficiently control this pandemic in India.

9 BIBLIOGRAPHY AND REFERENCES

[1] Wang, Li-sheng, Wang, Yi-ru, Ye, Da-wei, Liu, Qing-quan, 2020b. A review of the 2019 Novel Coronavirus (COVID-19) based on current evidence. Int. J. Antimicrob. Agents https://doi.org/10.1016/j.ijantimicag.2020.105948 (in press)

[2] WHO, 2020. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public. Worldometers, 2020.

[3] Tomar A, Gupta N. Prediction for the spread of COVID-19 in India and effectiveness of preventive measures. Science of The Total Environment. 2020 Apr 20:138762.

[4] Zhao, S. and Chen, H., 2020. Modeling the epidemic dynamics and control of COVID-19 outbreak in China. *Quantitative Biology*, pp.1-9.

[5] Shah K, Alqudah MA, Jarad F, Abdeljawad T. Semi-analytical study of pine wilt disease model with convex rate under caputo–febrizio fractional order derivative. Chaos Solitons Fractals 2020;135:109754.

[6] Jajarmi A, Arshad S, Baleanu D. A new fractional modelling and control strategy for the outbreak of dengue fever. Physica A 2019;535:122524.

[7] Jajarmi A, Yusuf A, Baleanu D, Inc M. A new fractional hrsv model and its optimal control: a non-singular operator approach. Physica A 2019:123860.

[8] Liestøl K, Andersen PK. Updating of covariates and choice of time origin in survival analysis: problems with vaguely defined disease states. Stat Med 2002;21(23):3701–14.

[9] Krätschmer V. Strong consistency of least-squares estimation in linear regression models with vague concepts. J Multivar Anal 2006;97(3):633–54.

[10] Bandyopadhyay, S. K., Dutta, S., Machine learning approach for confirmation of Covid-19 cases: positive, negative, death and release. medRxiv 2020.

[11] Makridakis S, Hibon M. ARMA models and the Box–Jenkins methodology. Journal of Forecasting. 1997 May;16(3):147-63.

[12] Solo V. The order of differencing in ARIMA models. Journal of the American Statistical Association. 1984 Dec 1; 79(388):916-21.

[13] Hyndman RJ, Athanasopoulos G. Forecasting: principles and practice. OTexts; 2018 May 8.

[14] Taylor SJ, Letham B. Forecasting at scale. The American Statistician. 2018 Jan 2;72(1):37-45.

[15] Harvey AC, Shephard NG. Estimation and testing of stochastic variance models. Suntory and Toyota International Centres for Economics and Related Disciplines, LSE; 1993.

[16] https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset .

[17] https://www.kaggle.com/sudalairajkumar/covid19-in-india .

[18] worldometer : https://www.worldometers.info/coronavirus/

[19] Hospital in the country :Ministry of Health and Family Welfare

[20] WHO Media Statement : https://www.who.int/indonesia/news/detail/08-03-2020-knowing-the-risk-for-COVID-19#:~:text=Most%20people%20(about%2080,are%20at%20greater%20risk.

[21] T. Chai , R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature.

[22] Ummul Khair, Hasanul Fahmi, Sarudin Al Hakim,Robbi Rahim. Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error