VISVESVARYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

Project Report on

“Forecasting & Severity Analysis of COVID-19 Using Machine Learning Approach”

submitted in the partial fulfillment of the requirement for the award degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

submitted by

Ms. Ashwini	**1AR18CS004**
Ms. Mamatha U	**1AR18CS018**
Ms. Nagajyothi P	**1AR18CS024**
Ms. Rashmitha R	**1AR18CS035**

Under the guidance of

Mr. Ravi G H
Assistant Professor,
Department of CS & E

AIEMS
BENGALURU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

2021-22

B.V.V. Sangha’s

AMRUTA INSTITUTE OF ENGINEERING & MANAGEMENT SCIENCES

Bidadi Industrial Area, Bidadi, Bengaluru – 562109

BVV Sangha, Bagalkot

AMRUTA INSTITUTE OF ENGINEERING & MANAGEMENT SCIENCES

Approved by AICTE, New Delhi

Recognized by Government of Karnataka & Affiliated to VTU, Belagavi

AIEMS
BENGALURU

Department of Computer Science & Engineering

AIEMS
BENGALURU

CERTIFICATE

This is to certify that the project work entitled **"Forecasting & Severity Analysis of COVID-19 Using Machine Learning Approach"** is a bonafide work carried out by

Ms. Ashwini	1AR18CS004
Ms. Mamatha U	1AR18CS018
Ms. Nagajyothi P	1AR18CS024
Ms. Rashmitha R	1AR18CS035

in partial fulfilment of award of Degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi, during the academic year 2021-2022.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements associated with of Project Work (18CSP83) prescribed for the said degree.

Signature of the guide,	Signature of the HOD,	Signature of the Principal,
Mr. Ravi G H Assistant Professor, Dept. of CSE, AIEMS	**Dr. M S Patel** Professor & Head, Dept. of CSE, AIEMS	**Dr. Santosh M Muranal** Principal, AIEMS

External Viva

Name of the Examiners **Signature with Date**

1.

2.

DECLARATION

We the undersigned students of 8th semester, Department of Computer Science & Engineering, B.V.V Sangha's Amruta Institute of Engineering & Management Sciences, declare that the project work entitled "Forecasting and Severity Analysis of COVID-19 Using Machine Learning Approach" is a bonafide work of us.

We also declare that this project was not entitled for submission to any other university in the past shall remain the only submission made and will not be submitted by us to any other university in the future.

Name	**USN**	**Signature**
Ms. Ashwini	**1AR18CS004**	____________________
Ms. Mamatha U	**1AR18CS018**	____________________
Ms. Nagajyothi P	**1AR18CS024**	____________________
Ms. Rashmitha R	**1AR18CS035**	____________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We are grateful to our institution, **B.V.V Sangha's Amruta Institute of Engineering & Management Sciences (AIEMS)**, with its ideals and inspirations for having provided us with the facilities, which has made this project a success.

We earnestly thank **Dr. Santosh M Muranal, Principal, AIEMS**, for facilitating academic excellence in the college and providing us with the congenial environment to work in, that helped us in completing this project.

We wish to extend our profound thanks to **Dr. M S Patel, Professor & Head, Department of Computer Science & Engineering, AIEMS**, for giving us the consent to carry out this project.

We owe our sincere thanks to our project Co-Ordinator and internal guide **Mr. Ravi G H, Assistant Professor of Computer Science & Engineering, AIEMS**, for his immense help during the project and also for his valuable suggestions on the project report preparations, which helped us in the successful completion of the project, and also for helping us to carry out project as scheduled and for his valuable suggestions.

We would like to thank all the faculties of **Computer Science & Engineering Department**, for their valuable advice and support.

We would like to thank all the teaching and nonteaching staff of Computer Science department for their valuable advice and support. We would like to express our sincere thanks to our parents and friends for their support.

Ms. ASHWINI [1AR18CS004]
Ms. MAMATHA U [1AR18CS018]
Ms. NAGAJYOTHI P [1AR18CS024]
Ms. RASHMITHA R [1AR18CS035]

ABSTRACT

SARS-CoV-2 (n-coronavirus) is a global pandemic that causes the deaths of millions of people worldwide. It can cause Pneumonia and severe acute respiratory syndrome (SARS) and lead to death in severe cases. It is an asymptomatic disease that hardens our life and work conditions. As there is no effective treatment available, many scientists and researchers are trying their best to fight the pandemic. This paper focused on the coronavirus pandemic situation in the global and Bangladesh region and its related effects and future status. We have utilized different information representation and machine learning calculations to recreate the affirmed, recuperated, and passing cases. We believe the research will help scientists, researchers, and ordinary people predict and analyze this pandemic's impact. Finally, the comparison and analysis of different models and algorithms successfully showed our visualization and prediction success.

CONTENTS

CHAPTERS **PAGE NO.**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Covid-19 disease is the one off the disorders. Though the symptoms are benign initially, they become more severe over time. Although for most people COVID-19 causes only mild illness, it can make some people very ill. More rarely, the disease can be fatal. Older people, and those with pre-existing medical conditions (such as high blood pressure, heart problems or diabetes) appear to be more vulnerable. COVID illness (COVID-19) is an irresistible infection brought about by a newfound Corona virus. Corona virus is a group of infections that causes the normal chilly, serious intense respiratory condition (SARS), and the Middle East respiratory syndrome (MERS). It is otherwise called the extreme intense respiratory disorder COVID 2 (SARS-CoV-2).

The novel generation of the coronavirus disease (COVID-19) was reported in late December 2019 in Wuhan, China. After only a few months, the virus became a global outbreak in 2020. On May 2020. The World Health Organisation (WHO) announced the situation as pandemic. The statistics by WHO on 8th October 2020 confirm 36 million infected people and a scary number of 1,056,000 deaths in 200 countries. With the growing trend of patients, there is still no effective cure or available treatment for the virus.

Scientists, healthcare organisations, and researchers are continuously working to produce appropriate medications or vaccines for the deadly virus, no definite success has been reported at the time of this research, and there is no certain treatment or recommendation to prevent or cure this new disease. Therefore, precautions are taken by the whole world to limit the spread of infection. These harsh conditions have forced the global communities to look for alternative ways to reduce the spread of the virus.

For several months, the World Health Organisation believed that COVID-19 was only transmittable via droplets emitted when people sneeze or cough and the virus does not linger in the air. However, on 8 July 2020, the WHO announced: “There is emerging evidence that COVID-19 is an airborne disease that can be spread by tiny particles suspended in the air after people talk or breathe, especially in crowded, closed environments or poorly ventilated settings”. Therefore, social distancing now claims to be even more important than thought before, and one of the best ways to stop the spread of the disease by forecasting the COVID-19 cases.

1.1 MACHINE LEARNING

Machine learning is a discipline that deals with programming the system so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs. To solve this problem, algorithms are developed that build knowledge from a specific data and past experience by applying the principles of statistical science, probability, logic, mathematical optimization, reinforcement learning, and control theory.

1.1.1 TYPES OF MACHINE LEARNING

Machine learning is available all around the world and all are experiencing any one of the occurrences every day. A new method of data is arising from using machine learning. As there are few complexities seen in machine learning it is divided in to two main two main areas called supervised and unsupervised learning Each type has its own advantages and different working process.70% of machine learning is said to be supervised learning and can expect only 10-20% as unsupervised learning.

1.1.1.1 Supervised learning

Supervised learning involves building a machine learning model that is based on labeled samples. For example, if there is a need to build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, first need to create a database and label it. There is a need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new examples. There are many supervised learning algorithms such as Logistic Regression, Neural networks, Support Vector Machines (SVMs).

Common examples of supervised learning include classifying e-mails into spam and ham categories, labeling web pages based on their content, and voice recognition.

1.1.1.2 Unsupervised learning

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called unlabeled data.

1.1.1.3 Reinforcement learning

Like a traditional method of analysis, the algorithm is there that discovers data via trial and process, later which results in greater rewards. Action, agent, environment are the major components of reinforcement learning. Decision-makers are called agents, everything through which agent interact is called environment and all the actions performed by the agents are referred to actions here. When an agent makes a choice of action to increase the expected reward for a specified duration, reinforcement learning happens.

1.1.2 PHASES OF MACHINE LEARNING

Machine Learning has given the computer systems the abilities to automatically learn without being explicitly programmed. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the cycle is to find a solution to the problem or project.

1.1.2.1 Gathering Data

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, the different data sources has to be identified, as data can be collected from various sources as files, databases, internet or mobile devices. It is one of the most important steps of the life cycle. The quantity and quality of the output. The more will be the data, the more accurate will be the prediction.

1.1.2.2 Data preparation

After collecting the data, the data are to be prepare it for further steps. Data preparation and prepare to use in machine learning training.

In this steps, first, all the data were put together, and then randomize the ordering of data. This step can be further divided into two processes.

- Data Exploration

It is used to understand the nature of data that have to work with. The characteristics, format and quality of data are need to be understand. A better understanding of data leads to an effective outcome.

- Data pre-processing

It is an important step in data mining process. The phrase “garbage in, garbage out” is particularly applicable to data mining and machine learning projects. Data gathering methods or often loosely controlled, resulting in out of range values, impossible data combinations, missing values etc.

1.1.2.3 Data Wrangling

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use,

and the most important steps of the complete process. Cleaning of data is required to address the quality issues.

It is not necessary the data that have collected is always of use as some of the data may not be useful. In real world applications, collected data may have various issues, including.

- Missing values
- Duplicate data
- Invalid data
- Noise

1.1.2.4 Train Model

The next step is to train the model, in this step the model will be trained to improve its performance for better outcome of the problem. Various datasets are used to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules and features.

1.1.2.5 Test Model

Once the machine learning model has been trained on a given dataset, then the model is tested. In this step, the accuracy of the model will be checked by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

1.1.2.6 Deployment

The last step of machine learning life cycle is deployment, where the model will be deployed in the real-world system. If the above- prepared model is producing an accurate result as per the requirement with acceptable speed, then the model will be deployed in the real system.

1.1.3 APPLICATIONS OF MACHINE LEARNING ALGORITHMS

The developed machine learning algorithms are used in various applications such as:

- Vision processing
- Language processing
- Forecasting things like stock market trends, weather
- Pattern recognition

- Data mining
- Robotics
- Expert systems

1.2 DEEP LEARNING

Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier.

At its simplest, deep learning can be thought of as a way to automate predictive analytics. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.

1.2.1 DEEP LEARNING METHODS

Various methods can be used to create strong deep learning models. These techniques include learning rate decay, transfer learning, training from scratch and dropout.

1.2.1.1 Learning rate decay

The learning rate is a hyperparameter, a factor that defines the system or set conditions for its operation prior to the learning process that controls how much change the model experiences in response to the estimated error every time the model weights are altered. Learning rates that are too high may result in unstable training processes or the learning of a suboptimal set of weights. Learning rates that are too small may produce a lengthy training process that has the potential to get stuck.The learning rate decay method also called learning rate annealing or adaptive learning rates is the process of adapting the learning rate to increase performance and reduce training time.

The easiest and most common adaptations of learning rate during training include techniques to reduce the learning rate over time.

1.2.1.2 Transfer learning

This process involves perfecting a previously trained model; it requires an interface to the internals of a preexisting network. First, users feed the existing network new data containing previously unknown classifications. Once adjustments are made to the network, new tasks can be performed with more specific categorizing abilities. This method has the advantage of requiring much less data than others, thus reducing computation time to minutes or hours.

1.2.1.3 Training from scratch

This method requires a developer to collect a large labeled data set and conFigureure a network architecture that can learn the features and model. This technique is especially useful for new applications, as well as applications with a large number of output categories. However, overall, it is a less common approach, as it requires inordinate amounts of data, causing training to take days or weeks.

1.2.1.4 Dropout

This method attempts to solve the problem of overfitting in networks with large amounts of parameters by randomly dropping units and their connections from the neural network during training. It has been proven that the dropout method can improve the performance of neural networks on supervised learning tasks in areas such as speech recognition, document classification and computational biology.

1.3 PYTHON

Python is a popular programming language. It was created by Guido van Rossum and released in 1991. It is used for:

- Web development
- Software development
- Mathematics
- System scripting

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database system. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms.
- Python has a simple syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python was imagined in the late 1980s and its usage started in December 30, 1989 by Guido van Rossum at Centrum Wiskunde and Informatica (CWI) in the Netherlands as a successor to the ABC dialect (itself roused by SETL) capable of exemption dealing with and interfacing with the Amoeba working system. Van Rossum remains Python's chief creator. His proceeding with focal part in Python's advancement is reflected in the title given him by the Python people group.

1.3.1 PYTHON FEATURE

- Simple and easy to learn Python as only 33 keywords but JAVA as 83 keywords.
- High level programming language. Python is platform independent.
- Both object oriented and procedure oriented language.
- Interpreted language (It means not going to compile)
- Extensible
- Portability, moving from one platform to another without any change.
- Dynamically typed Programming Language. In python it is not required to declare type in python.

- Free ware (there is no license and cannot pay anything) furthermore, Open source (can able to see source code)

1.3.2 LIMITATIONS OF PYTHON

- Performance wise it is not up to the mark. Because its an interpreted language. Interpreter able to see only one line (JAVA is better performance compare to python in java JIT (just in compiler)).
- Mobile applications it is not up to the mark python is not suitable large scale enterprise applications.

1.3.3 FLAVORS OF PYTHON

- Cpython: It can be standard, it ca be used to c language python.
- Jpython: It is for JAVA application.
- Iron python: to work with Microsoft .net platform.
- Py: Internally JIT (just in time complex) compiler is there so performance wise too good.
- Ruby python: To handle Big data happily go for Anaconda python.
- Stackless (python for concurrency)
- Parallely you execute (like multithreaded) go for stackless.

1.3.4 APPLICATIONS OF PYTHON

1. **GUI-Based Desktop Applications:**

Python has simple syntax., modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using python includes.

- Image Processing and Graphic Design Applications:

Python has been used to make 2D imaging such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use python in variable proportions.

- Scientific and Computational Applications:

The higher speeds, productivity and availability of tools, such as scientific python and numeric python, have resulted in python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as Free CAD and finite element method software, such as Abacus, are coded in python.

- Games:

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using python including Civilization-IV, Disney’s Toontown Online, Vega Strike etc.

2. Web Frameworks and Web Applications

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP,SMTP, XML-RFC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in Aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

3. Enterprise and Business Applications

With features that include special libraries, extensibility, scalability and easily readable syntax, python is a suitable coding language for customizing larger applications. Reddit,which was originally written in Common lips, was rewritten in python in 2005. Python also contributed in a large part to functionality in You Tube.

4. **Operating Systems**

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, Fedora's and Red Hat Enterprise. Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of python for portage, its package management system.

5. **Language Development**

Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to python. Further, syntax of languages like Apple's Swift, Coffee Script, Cobra and OCaml all share similarity with Python.

6. **Prototyping**

Besides being quick and easy to learn. Python also has the open source advantage of being free with the support of a large community. This makes it the preferred choice for prototype development. Further, the agility, extensibility and scalability and ease of refactoring code associated with Python allow faster development from initial prototype. Since its origin in 1989, Python allow faster development from initial prototype. Since its origin in 1989, python has grown to become part of a plethora of web-based, desktop-based, graphic design, scientific and computational applications. With python available for Windows, Mac OS X and Linux/UNIX, it offers ease of development for enterprises. Additionally, the latest release Python 3.4.3 builds on the existing strengths of the language, with drastic improvement in Unicode support, among other new features.

1.3.5 PYTHON IN MACHINE LEARNING

Python is a popular platform used for research and development of production systems. It is a vast language with number of modules, packages and libraries that provides multiple ways of achieving a task. Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are used in data science and data analysis. They are also extensively used for creating scalable machine learning algorithms. Python implements popular machine learning techniques such as Classification, Regression, Recommendation and Clustering.

Python offers ready-made framework for performing data mining tasks on large volumes of data effectively in lesser time. It includes several implementations achieved through algorithms such as linear regression, logistic regression, Naïve Bayes, k-means, K nearest neighbor and Random Forest.

Libraries and Packages

To understand machine learning, there is need to have basic knowledge of python programming. In addition, there are a number of libraries and packages generally used in performing various machine learning tasks as listed below:

- **numpy** – is used for its n-dimensional array objects.
- **pandas** – is a data analysis library that includes data frames.
- **matplotlib** – is 2D plotting library for creating graphs and plots.
- **scikit-learn** – the algorithms used for data analysis and data mining tasks.
- **seaborn** – a data visualization library based on matplotlib.

1.4 OBJECTIVE

This project is a push to dissect the aggregate information of affirmed passings and recouped cases over the long run, which is examined in the information investigation area. The primary center is to investigate the spread pattern of this infection around the country. This project proposing two algorithms, i.e., Auto Regression algorithm & AR Model to measure the daily increase in confirmed, recovered, death cases and growth factor.

1.4.1 SCOPE OF THE PROJECT:

Since this project is associated with the medical related problem, the scope of the project is pretty high and it helps to check whether the vaccine is available for everyone or the people follow the SOPs to avoid the spread of the corona virus. Furthermore, on the other side, the possible solution is to follow the SOPs set by the World Health Organization (WHO) for the prevention of this disease. Forecasting helps in prevention of the disease.

1.4.2 PROBLEM IDENTIFICATION

The analysis suggests that the coronavirus is probably originated from the bats and transmitted to the other animals before going into the humans from the Wuhan (China) wet market in December 2019. Soon after that, it is spread like a fire in a forest and wrapped the whole world. In the initial phases, most countries imposed a lockdown to stop the spread of this deadly disease. But this is not a practical solution as the whole economy of the particular country goes down. Especially it creates a catastrophic situation for underdeveloped countries in terms of economy.

Multiple companies have launched the vaccines in different countries. But, to be fully vaccinated in the world is a time taking process and, roughly it takes approximately five years for the full-fledged immunization to the world's population. So, there are two possible solutions, either the vaccine is available for everyone or the people follow the SOPs to avoid the spread of the coronavirus. Furthermore, on the other side, the possible solution is to follow the SOPs set by the

World Health Organization (WHO) for the prevention of this disease. Forecasting helps in prevention of the disease.

1.4.3 PROBLEM STATEMENT

With the number of coronavirus cases growing exponentially, the nations are facing a shortage of doctors, particularly in rural areas where the quantity of specialists is less compared to urban areas. A doctor takes roughly 6 to 12 years to procure the necessary qualifications.

Thus, the number of doctors can’t be expanded quickly in a short time frame. A Telemedicine framework ought to be energized as far as possible in this difficult time.

1.4.4 EXISTING SYSTEM

We have few systems which uses basic algorithms they attempt to detect the problems through a kind of text data but yielding less efficiency. Existing system uses Naive Baye's algorithms, Support vector Machine, LinearSVC etc, but the disadvantage being they need lot of Data for training which apparently takes more execution time and also yields less efficient outputs.

1.4.5 DISADVANTAGES

- Less Efficient.
- Low Accuracy.

Proposed System:

The primary center is to investigate the spread pattern of this infection around the country. This project proposing two algorithms, i.e., Logistic Regression algorithm & Random Forest Regression forecasting algorithms to measure the daily increase in confirmed, recovered, death cases and growth factor.

1.4.7 ADVANTAGES

- High Performance.
- More effective.
- Fault tolerant.
- Scales well.

1.4.8 APPLICATION

- In Covid-19 Control system.
- In Health Care Domain for other diseases prediction also it may useful.

CHAPTER 2

LITERATURE SURVEY

Literature review on different techniques given by various researchers is being presented.

1. Early Detection of COVID-19 Hotspots Using Spatio-Temporal Data

Recently, the Centers for Disease Control and Prevention (CDC) has worked with other federal agencies to identify counties with increasing coronavirus disease 2019 (COVID-19) incidence (hotspots) and offers support to local health departments to limit the spread of the disease. Understanding the spatio-temporal dynamics of hotspot events is of great importance to support policy decisions and prevent large-scale outbreaks.

This paper presents a spatio-temporal Bayesian framework for early detection of COVID-19 hotspots (at the county level) in the United States. We assume both the observed number of cases and hotspots depend on a class of latent random variables, which encode the underlying spatio-temporal dynamics of the transmission of COVID-19. Such latent variables follow a zeromean Gaussian process, whose covariance is specified by a non-stationary kernel function. The most salient feature of our kernel function is that deep neural networks are introduced to enhance the model’s representative power while still enjoying the interpretability of the kernel.

We derive a sparse model and fit the model using a variational learning strategy to circumvent the computational intractability for large data sets. Our model demonstrates better interpretability and superior hotspot detection performance compared to other baseline methods.

2. Predicting COVID-19 in China Using Hybrid AI Model

The coronavirus disease 2019 (COVID-19) breaking out in late December 2019 is gradually being controlled in China, but it is still spreading rapidly in many other countries and regions worldwide. It is urgent to conduct prediction research on the development and spread of the epidemic. In this article, a hybrid artificial-intelligence (AI) model is proposed for COVID-19 prediction. First, as traditional epidemic models treat all individuals with coronavirus as having the same infection rate, an improved susceptible-infected (ISI) model is proposed to estimate the variety of the infection rates for analyzing the transmission laws and development trend.

Second, considering the effects of prevention and control measures and the increase of the public's prevention awareness, the natural language processing (NLP) module and the long short-term memory (LSTM) network are embedded into the ISI model to build the hybrid AI model for COVID-19 prediction. The experimental results on the epidemic data of several typical provinces and cities in China show that individuals with coronavirus have a higher infection rate within the third to eighth days after they were infected, which is more in line with the actual transmission laws of the epidemic. Moreover, compared with the traditional epidemic models, the proposed hybrid AI model can significantly reduce the errors of the prediction results.

3. A greedy-based oversampling approach to improve the prediction of mortality in MERS patients

Predicting mortality of Middle East respiratory syndrome (MERS) patients with identified outcomes is a core goal for hospitals in deciding whether a new patient should be hospitalized or not in the presence of limited resources of the hospitals. We present an oversampling approach that we call Greedy-Based Oversampling Approach (GBOA). We evaluate our approach and compare it against the standard oversampling approach from a classification perspective on real dataset collected from the Saudi Ministry of Health using two popular supervised classification methods, Random Forests and Support Vector Machines. Our results demonstrate that our approach outperforms the other standard approach from a classification perspective by giving the highest accuracy with statistical significance on the 20 simulations of the real dataset.

4. A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization From Chest CT

Accurate and rapid diagnosis of COVID-19 suspected cases plays a crucial role in timely quarantine and medical treatment. Developing a deep learning-based model for automatic COVID-19 diagnosis on chest CT is helpful to counter the outbreak of SARS-CoV-

2. A weakly-supervised deep learning framework was developed using 3D CT volumes for COVID-19 classification and lesion localization.

For each patient, the lung region was segmented using a pre-trained UNet; then the segmented 3D lung region was fed into a 3D deep neural network to predict the probability of COVID-19 infectious; the COVID-19 lesions are localized by combining the activation regions in the classification network and the unsupervised connected components. 499 CT volumes were used for training and 131 CT volumes were used for testing. Our algorithm obtained 0.959 ROC AUC and 0.976 PR AUC. When using a probability threshold of 0.5 to classify COVID-positive and COVID-negative, the algorithm obtained an accuracy of 0.901, a positive predictive value of 0.840 and a very high negative predictive value of 0.982.

The algorithm took only 1.93 seconds to process a single patient's CT volume using a dedicated GPU. Our weakly-supervised deep learning model can accurately predict the COVID-19 infectious probability and discover lesion regions in chest CT without the need for annotating the lesions for training.

CHAPTER 3

SYSTEM DESIGN AND MODELING

3.1. SYSTEM ARCHITECTURE

An efficient prediction algorithm has built using the ANACONDA3Jupyter Notebook platform, where we utilize the data processing technique. The multi-stage data analysis like table and the graphical view has used to analyze the pandemic of COVID-19. The next week's prediction of confirming cases have done by using the Polynomial Regression algorithm, Arima model, and Facebook prophet time series forecasting. Before that, we classify our region into classification depending on the confirms cases, death rates, recovery rates. After completing the research and applying the algorithms, we find and declaring the best algorithm for the COVID-19 forecasting method by comparing the root mean square error value (RMSE). In Fig. 1 the overall flow chart of the introduced system represented.

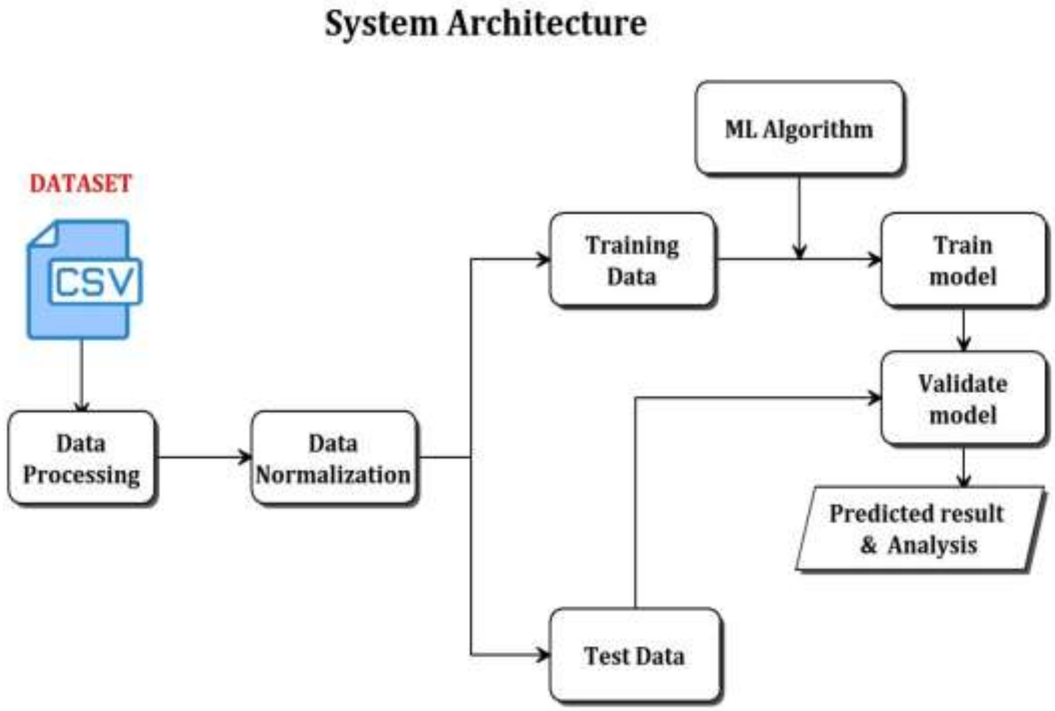


Fig 3.1.1 : System Architecture

3.2 FLOW CHART DIAGRAM

It is important to complete all tasks and meet deadlines. There are many project management tools that are available to help project managers manage their tasks and schedule and one of them is the flowchart.

A flowchart is one of the seven basic quality tools used in project management and it displays the actions that are necessary to meet the goals of a particular task in the most practical sequence. Also called as process maps, this type of tool displays a series of steps with branching possibilities that depict one or more inputs and transforms them to outputs.

The advantage of flowcharts is that they show the activities involved in a project including the decision points, parallel paths, branching loops as well as the overall sequence of processing through mapping the operational details within the horizontal value chain. Moreover, this particular tool is very used in estimating and understanding the cost of quality for a particular process. This is done by using the branching logic of the workflow and estimating the expected monetary returns.

3.3 USE CASE DIAGRAMS:

A use case is a set of scenarios that describing an interaction between a source and a destination. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. shows the use case diagram.

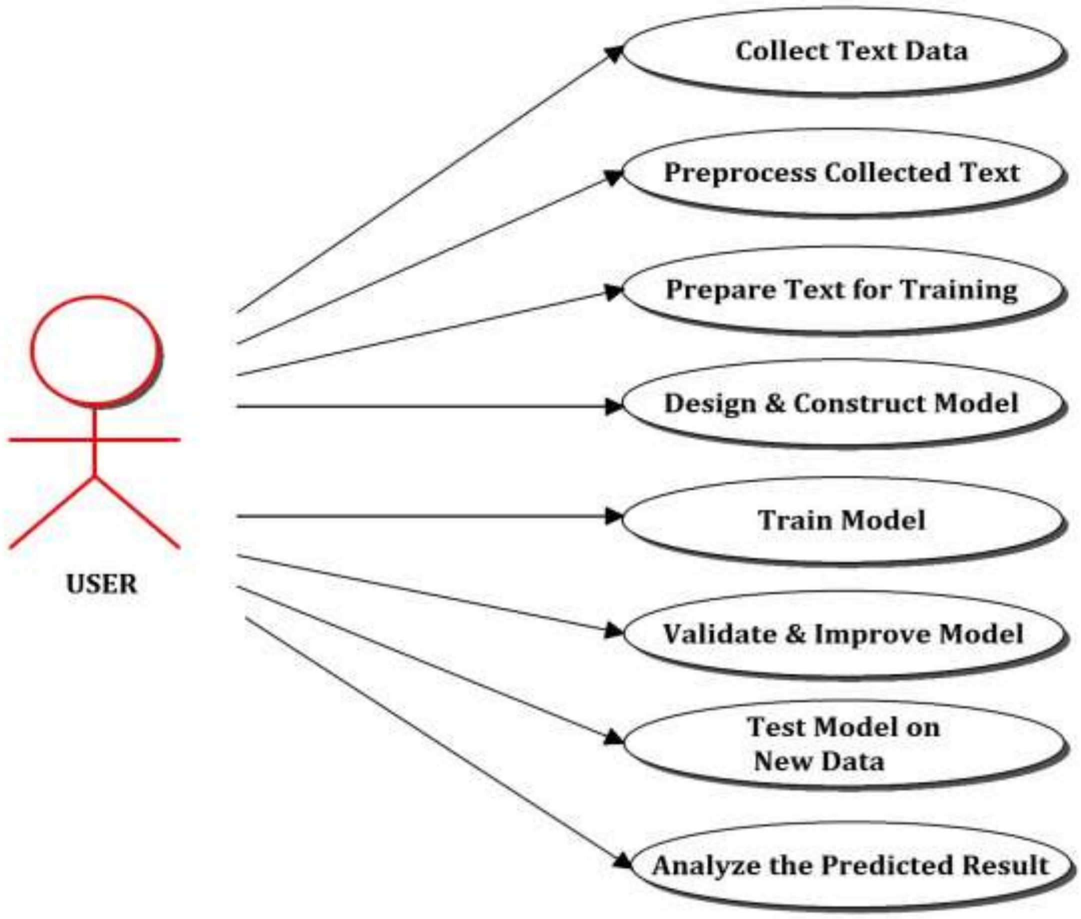


Fig 3.3.1 : Use case diagram user

3.4 DATA FLOW DIAGRAM:

A data flow diagram (DFD) is graphic representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context level DFD first which shows the interaction between the system and outside entities. DFD’s show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:

1. Squares representing external entities, which are sources and destinations of information entering and leaving the system.

2. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly.
4. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.
5. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

3.4.1 LEVEL 0 DATA FLOW DIAGRAM

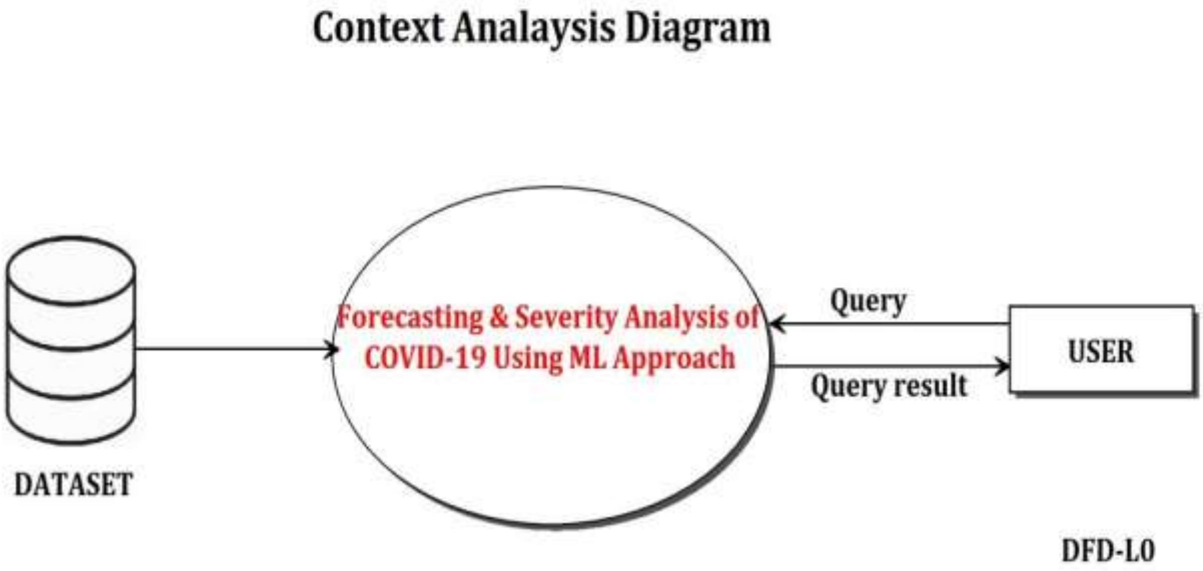

Context Analaysis Diagram
DATASET
Forecasting & Severity Analysis of COVID-19 Using ML Approach
Query
Query result
USER
DFD-L0

Fig 3.4.1.1 : Level 0 data flow diagram

3.4.2 LEVEL 1 DATA FLOW DIAGRAM

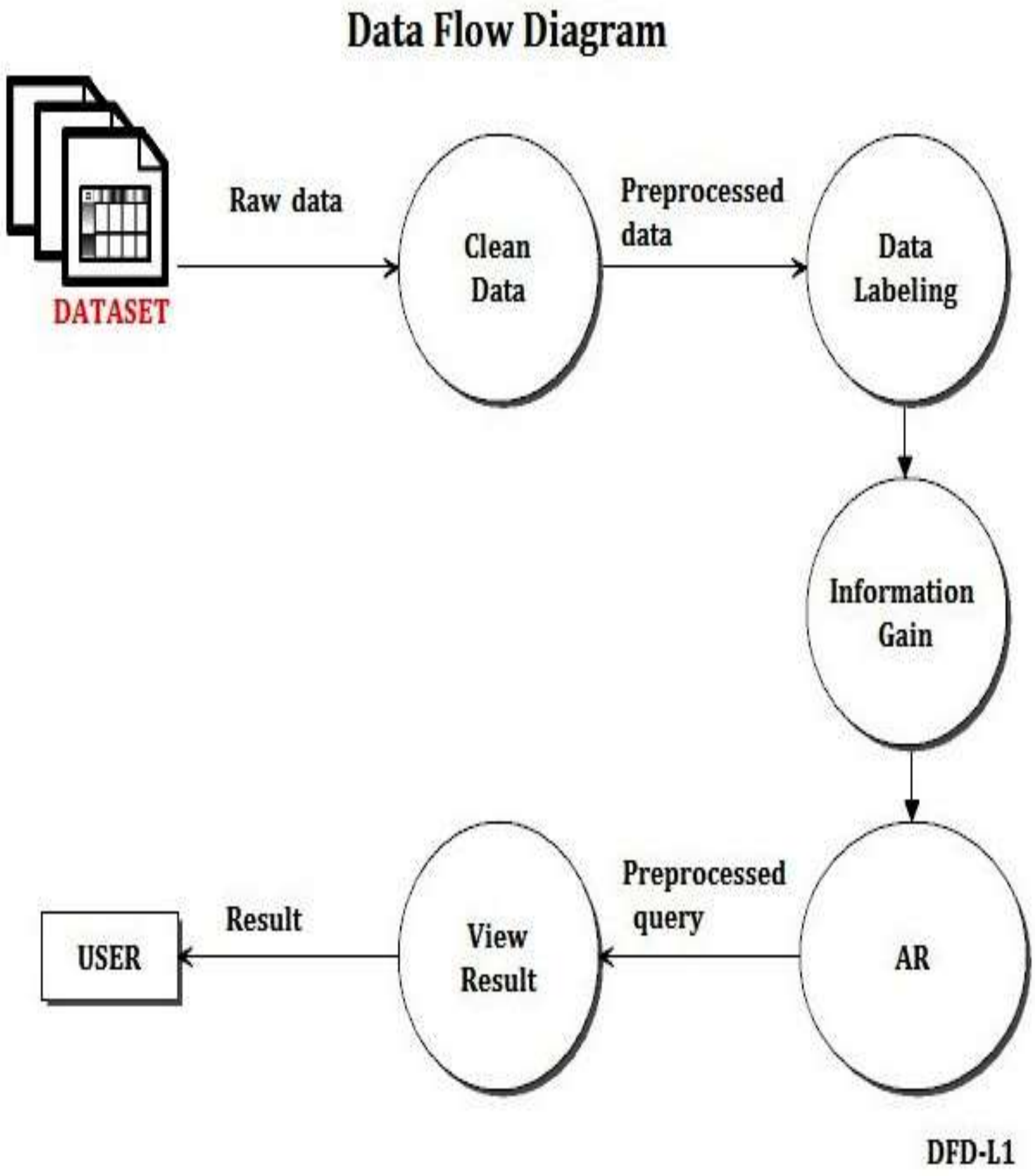


Fig 3.4.2.1 : Level 1 data flow diagram

3.4.3 Level2 DATA FLOW DIAGRAM

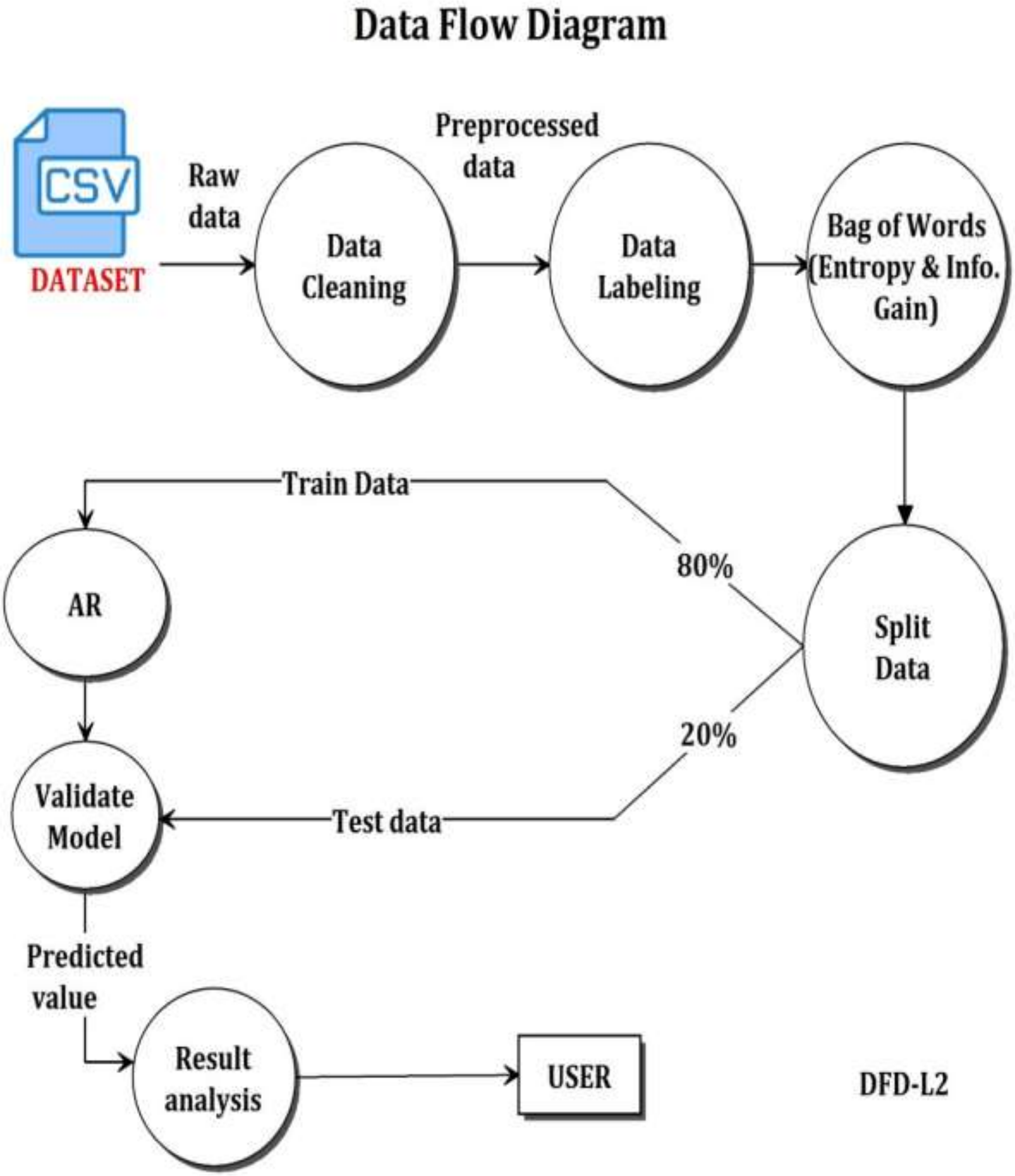


Fig 3.4.3.1 : Level2 data flow diagram

3.5 FLOW CHART DIAGRAM

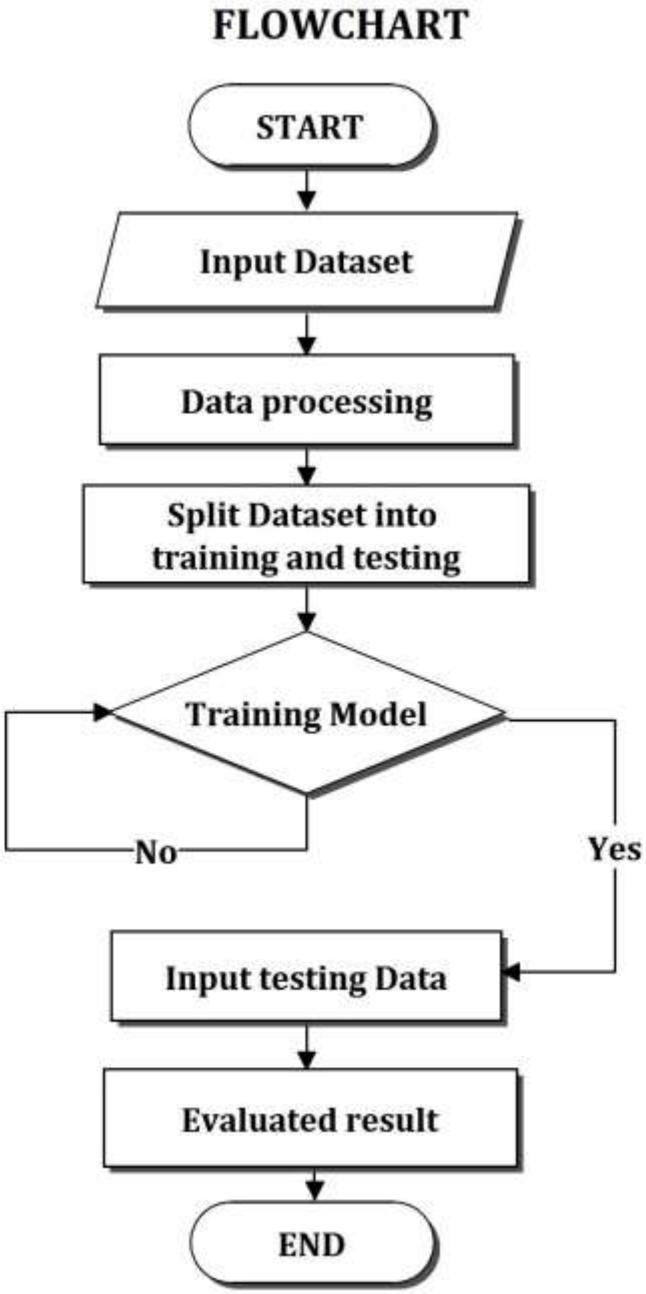
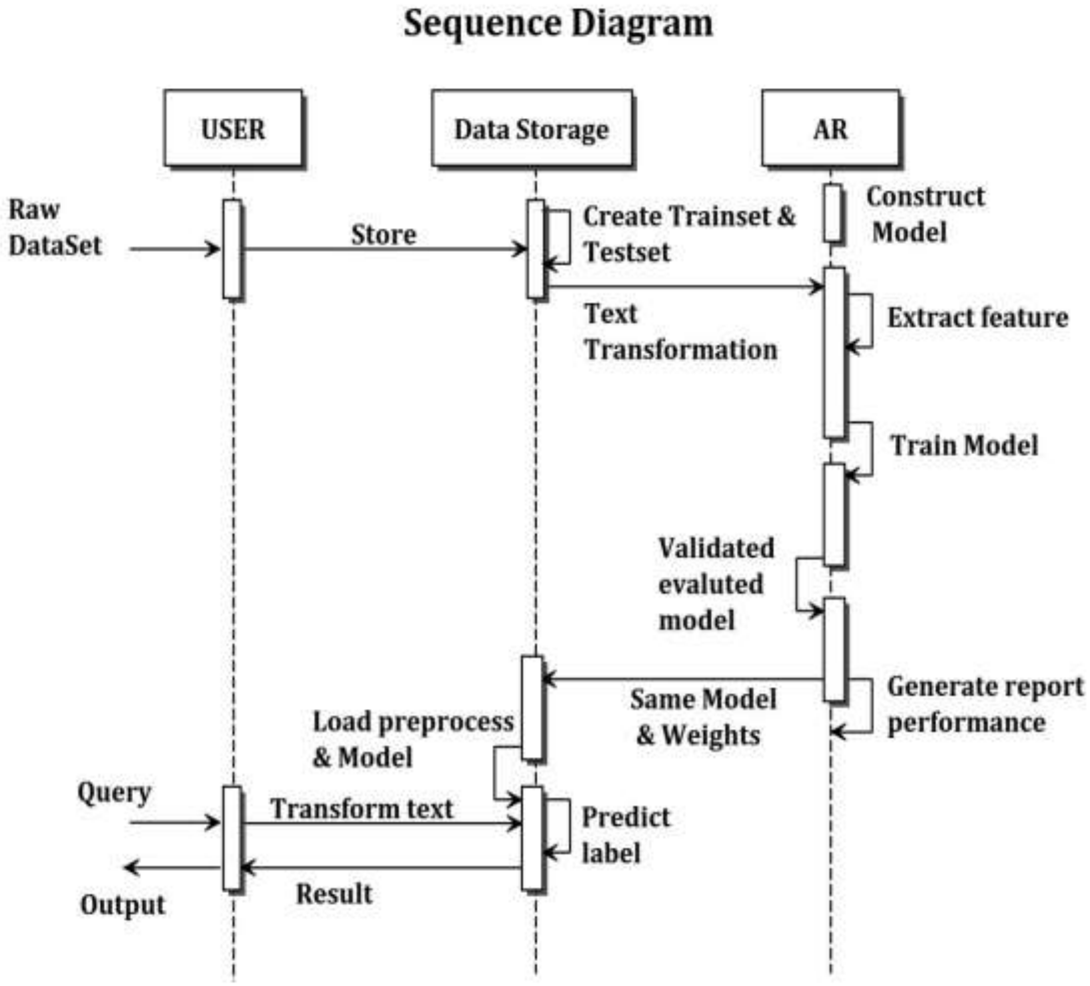


Fig 3.5.1: Flow chart diagram

3.6 SEQUENCE DIAGRAM



3.6.1 : sequence diagram

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

A Software Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a sets of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, the SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software system must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of Requirements are not always obvious and can come from any number of source.

4.1 HARDWARE AND SOFTWARE REQUIREMENTS

4.1.1 Hardware Requirements

- System Processor : i7 / i5 / i3 processor
- Hard Disk : 500 GB.
- Ram : 8 GB / 12 GB.
- Any desktop / Laptop system with above configuration or higher level.

4.1.2 Software Requirements:

• Operating system	:	Windows 8 / 10 (64 bits OS)
• Programming Language	:	Python 3
• Framework	:	Anaconda
• Libraries	:	Keras, TensorFlow
IDE	:	Jupyter Notebook

4.2 ALGORITHMS

Autoregression analysis is a standard technique in signal processing where a linear predictor estimates the value of each sample of a signal by a linear combination of previous values.

What Is an Autoregressive Model? A statistical model is autoregressive if it predicts future values based on past values. For example, an autoregressive model might seek to predict a stock's future prices based on its past performance.

An autoregressive model is when a value from a time series is regressed on previous values from that same time series. for example, on y_{t-1} : $y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$.

A regression model, such as linear regression, models an output value based on a linear combination of input values.

For example: y = b0 + b1*X1

Where y is the prediction, b0 and b1 are coefficients found by optimizing the model on training data, and X is an input value.

This technique can be used on time series where input variables are taken as observations at previous time steps, called lag variables. For example, we can predict the value for the next time step (t+1) given the observations at the last two steps (t-1 and t-2). As a regression model, this would look as follows:

X(t+1) = b0 + b1*X(t-1) + b2*X(t-2)

Because the regression model uses data from the same input variable at previous time steps, it is referred to as an autoregression (regression of self).

For an AR(1) model:

- when b1=0, xt is equivalent to white noise;
- when b1=1 and c=0c=0, xt is equivalent to a random walk;
- when b1=1 and c≠0c≠0, xt is equivalent to a random walk with drift;
- when b1<0, xt tends to oscillate around the mean.

We normally restrict autoregressive models to stationary data, in which case some constraints on the values of the parameters are required.

- For an AR(1) model: −1<b1<1
- For an AR(2) model: −1<b2<1, b1+b2<1, b2−b1<1.

When p≥3p≥3, the restrictions are much more complicated. R takes care of these restrictions when estimating a model.

4.3 MODULES DESCRIPTION

Modules

- Data Acquisition and Preprocessing
- Feature Selection and Data Preparation
- Model Construction and Model Training
- Model Validation and Result Analysis

4.3.1 Data Acquisition and Preprocessing

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

The primary data collected from the online sources remains in the raw form of statements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires. The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Data Preprocessing is necessary because of the presence of unformatted real-world data. Mostly real-world data is composed of -

Inaccurate data (missing data) - There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometrics and much more.

The presence of noisy data (erroneous data and outliers) - The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.

Inconsistent data - The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.

4.3.2 Feature Selection and Data Preparation

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process.

Feature engineering is the most important art in machine learning which creates the huge difference between a good model and a bad model. Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.

The process of organizing data into groups and classes on the basis of certain characteristics is known as the classification of data. Classification helps in making comparisons among the categories of observations. It can be either according to numerical characteristics or according to attributes. So here we need to visualize the prepared data to find whether the training data contains the correct label, which is known as a target or target attribute.

Next, we will slice a single data set into a training set and test set.

- ➢ **Training set—**a subset to train a model.
- ➢ **Test set—**a subset to test the trained model.

Make sure that your test set meets the following two conditions:

- ✓ Is large enough to yield statistically meaningful results.
- ✓ Is representative of the data set as a whole? In other words, don't pick a test set with different characteristics than the training set.

Assuming that your test set meets the preceding two conditions, your goal is to create a model that generalizes well to new data. **Our test set serves as a proxy for new data.**

4.3.3 Model Construction and Model Training

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term ML model refers to the model artifact that is created by the training process. The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

4.3.4 Model Validation and Result Analysis

In testing phase the model is applied to new set of data. The training and test data are two different datasets. The goal in building a machine learning model is to have the model perform well. On the training set, as well as generalize well on new data in the test set. Once the build model is tested then we will pass real time data for the prediction. Once prediction is done then we will analyzes the output to find out the crucial information.

Never train on test data**.** If you are seeing surprisingly good results on your evaluation metrics, it might be a sign that you are accidentally training on the test set. For example, high accuracy might indicate that test data has leaked into the training set. The mean absolute error, root mean square error and coefficient of determination were chosen as parameters to evaluate and score the models.

The average delays in departure and arrival has been predicted using three basic statistical parameters: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and the Coefficient of Determination (CD). The Mean Absolute Error helps to determine how close the predicted outcomes are to the consequent outcomes. It is a more natural measure of average error.The Root Mean Square Error is the square root of the mean of squares of all the errors. As compared to Mean Absolute Error, it helps to expand and liquidate the large errors. It is very commonly used and is an efficient error metric for numerical predictions.

The Coefficient of Determination is a very key attribute of regression analysis. It is denoted by R^2. It is a statistical measure of how close the data is to the fitted regression line and is often used in classical regression analysis.

The observed results obtained while evaluating for arrival delay with the test data. This shows that the features chosen are a good predictor of patterns with high accuracy and small error.

CHAPTER 5

IMPLEMENTATION TECHNOLOGIES

5.1.1 ANACONDA

Anaconda is a free and open-source distribution of the programming languages Python and R (check out these Python online courses and R programming courses). The distribution comes with the Python interpreter and various packages related to machine learning and data science. Basically, the idea behind Anaconda is to make it easy for people interested in those fields to install all (or most) of the packages needed with a single installation.

5.1.2 WHAT IS JUPYTER NOTEBOOK AND ANACONDA?

Jupyter Notebook and Anaconda – Jupyter Notebook is an interactive Python shell that runs in your browser. When installed through Anaconda, it is easy to quickly set up a Python development environment. Since it’s easy to set up and easy to run, it will be easy to learn Python. Jupyter Notebook turns your browser into a Python development environment. The only thing you have to install is Anaconda. In essence, it allows you to enter a few lines of

Python code, press CTRL+Enter, and execute the code. You enter the code in cells and then run the currently selected cell. There are also options to run all the cells in your notebook. This is useful if you are developing a larger program.

What Is Anaconda?

Anaconda is the easiest way to ensure that you don’t spend all day installing Jupyter. Simply download the Anaconda package and run the installer. The Anaconda software package contains everything you need to create a Python development environment. Anaconda comes in two versions—one for Python 2.7 and one for Python 3.x. For the purposes of this guide, install the one for Python 2.7. Anaconda is an open-source data science platform. It contains over 100 packages for use with Python, R, and Scala. You can download and install Anaconda quickly with minimal effort. Once installed, you can update the packages or Python version or create environments for different projects.

Getting Started

1. Download and install Anaconda at https://www.anaconda.com/download.
2. Once you’ve installed Anaconda, you’re ready to create your first notebook. Run the Jupyter Notebook application that was installed as part of Anaconda.
3. Your browser will open to the following address: http://localhost:8888. If you’re running Internet Explorer, close it. Use Firefox or Chrome for the best results. From there, browse to http://localhost:8888.
4. Start a new notebook. On the right-hand side of the browser, click the drop-down button that says “New” and select *Python* or *Python 2*.
5. This will open a new iPython notebook in another browser tab. You can have

many notebooks open in many tabs.

6. Jupyter Notebook contains cells. You can type Python code in each cell. To get started (for Python 2.7), type print "Hello, World!" in the first cell and hit CTRL+Enter. If you're using Python 3.5, then the command is print ("Hello, World!").

5.1.3 HOW TO CREATE ENVIRONMENT IN CONDA AND JUPYTER?

Let's imagine you want to use Jupyter Notebook to install both Tensorflow 2.0 and Tensorflow 1.15. First, decide whether you want to utilise Tensorflow on the GPU or the CPU for this example. Add "-gpu" to TensorFlow to use the GPU version; otherwise, leave it alone.

5.1.4 PYTHON INTRODUCTION:

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted −** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.1.5 FEATURES:

Python's features include – All these.

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases

.

- **GUI Programming −** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable −** Python provides a better structure and support for large programs than shell scripting.

Why python emerging as a leader:

There's battle out there happening in the minds of aspiring data scientists to choose the best data science tool. Though there are quite a number of data science tools that provide the much-needed option, the close combat narrows down between two popular languages – Python and R. Between the two, Python is emerging as the popular language used more in data science applications.

Take the case of the tech giant Google that has created the deep learning framework called tensor flow – Python is the primary language used for creating this framework. Its footprint has continued to increase in the environment promoted by Netflix. Production engineers at Face book and Khan Academy have for long been using it as a prominent language in their environment. Python has other advantages that speed up it's upward swing to the top of data science tools. It integrates well with the most cloud as well as platform-as-a-service providers. In supporting multiprocessing for parallel computing, it brings the distinct advantage of ensuring large-scale performance in data science and machine learning. Python can also be extended with modules written in C/C++.

Where Python becomes the perfect-fit:

There are tailor-made situations where it is the best data science tool for the job. It is perfect when data analysis tasks involve integration with web apps or when there is a need to incorporate statistical code into the production database. The full-fledged programming nature of Python makes it a perfect fit for implementing algorithms. Its packages rooted for specific data science jobs. Packages like Numpy, SciPy, and pandas produce good results for data analysis jobs. While there is a need for graphics, Python's Matplotlib emerges as a good package, and for machine learning tasks, scikit-learn becomes the ideal alternate.

Why is Python preferred over other data science tools?

It is ‘Pythonic’ when the code is written in a fluent and natural style. Apart from that, it is also known for other features that have captured the imaginations of data science community.

Easy to learn:

The most alluring factor of Python is that anyone aspiring to learn this language can learn it easily and quickly. When compared to other data science languages like R, Python promotes a shorter learning curve and scores over others by promoting an easy-to-understand syntax.

Scalability:

When compared to other languages like R, Python has established a lead by emerging as a scalable language, and it is faster than other languages like Matlab and Stata. Python’s scalability lies in the flexibility that it gives to solve problems, as in the case of YouTube that migrated to Python. Python has come good for different usages in different industries and for rapid development of applications of all kinds.

Choice of data science libraries:

The significant factor giving the push for Python is the variety of data science/data analytics libraries made available for the aspirants. Pandas, StatsModels, NumPy, SciPy, and Scikit-Learn, are some of the libraries well known in the data science community. Python does not stop with that as libraries have been growing over time. What you thought was a constraint a year ago would be addressed well by Python with a robust solution addressing problems of specific nature.

Python community:

One of the reasons for the phenomenal rise of Python is attributed to its ecosystem. As Python extends its reach to the data science community, more and more volunteers are creating data science libraries. This, in turn, has led the way for creating the most modern tools and processing in Python. The widespread and involved community promotes easy access for aspirants who want to find solutions to their coding problems. Whatever queries you need, it is a click or a Google search away. Enthusiasts can also find access to professionals on Code mentor and Stack Overflow to find the right answers for their queries.

Graphics and visualization:

Python comes with varied visualization options. Matplotlib provides the solid foundation around which other libraries like Sea born, pandas plotting, and ggplot have been built. The visualization packages help you get a good sense of data, create charts, graphical plot and create web-ready interactive plots.

5.1.6 WHY CHOOSE PYTHON?

If you're going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day. According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the. Popularity of Programming Language Index

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like. Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly. This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step. One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. computationally intensive, like graphics processing.

For some applications that are particularly or intense number crunching, this can be limiting. In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language.

These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language. Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect. Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It’s Not That Simple

For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types. Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Is Python ‘the’ tool for machine learning?

When python comes to data science, machine learning is one of the significant elements used to maximize value from data. With Python as the data science tool, exploring the basics of machine learning becomes easy and effective. In a nutshell, machine learning is more about statistics, mathematical optimization, and probability. It has become the most preferred machine learning tool in the way it allows aspirants to ‘do math’ easily.

Name any math function, and you have a Python package meeting the requirement. There is Numpy for numerical linear algebra, CVXOPT for convex optimization, Scipy for general scientific computing, SymPy for symbolic algebra, PYMC3, and Statsmodel for statistical modeling. With the grip on the basics of machine learning algorithm including logistic regression and linear regression, it makes it easy to implement machine learning systems for predictions by way of its scikit-learn library. It’s easy to customize for neutral networks and deep learning with libraries including Keras, Theano, and TensorFlow. Data science landscape is changing rapidly, and tools used for extracting value from data science have also grown in numbers. The two most popular languages that fight for the top spot are R and Python. Both are revered by enthusiasts, and both come with their strengths and weaknesses. But with the tech giants like Google showing the way to use Python and with the learning curve made short and easy, it inches ahead to become the most popular language in the data science world.

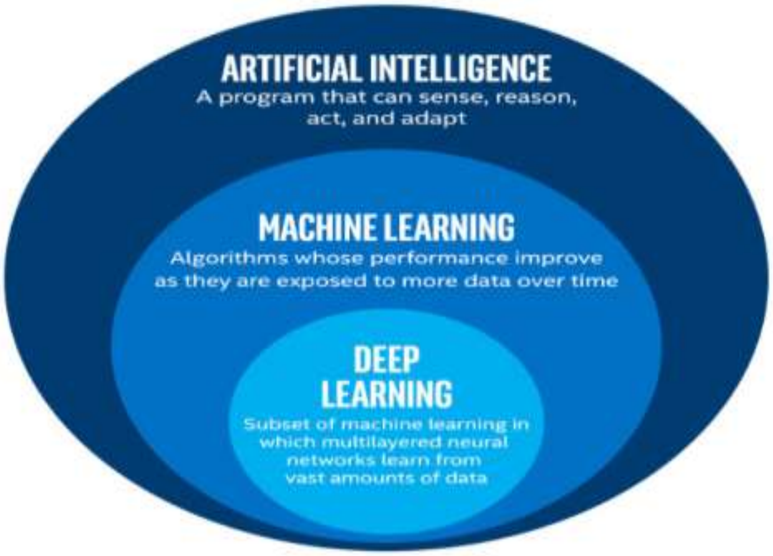

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Fig 5.5.6.1: Machine Learning Model

SDLC (System Development Life Cycle)

SDLC will cover the details explanation of methodology that is being used to make this project complete and working well. Many methodology or findings from this field mainly generated into journal for others to take advantages and improve as upcoming studies. The method is use to achieve the objective of the project that will accomplish a perfect result. In order to evaluate this project, the methodology based on System Development Life Cycle (SDLC), generally three major step, which is planning, implementing and analysis.

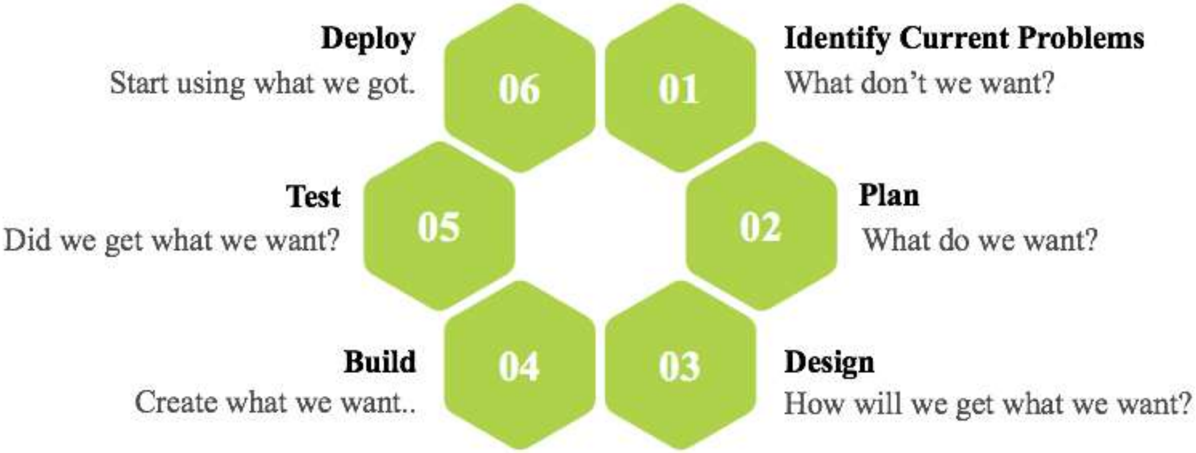


Fig 5.1.2.1 : Software Development Life Cycle

5.2 METHODOLOGY

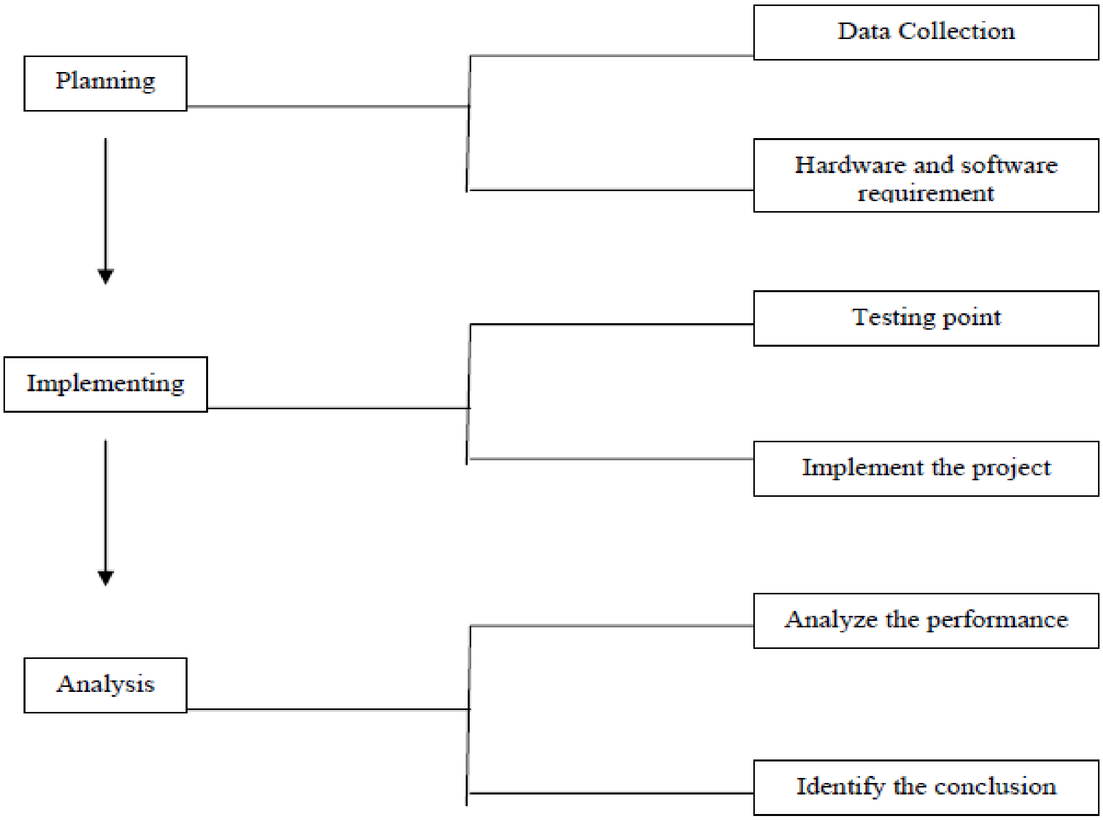


Fig 5.2.1 : Steps of Methodology

1. Planning:

To identify all the information and requirement such as hardware and software, planning must be done in the proper manner. The planning phase has two main elements namely data collection and the requirements of hardware and software.

2. Data collection:

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

The primary data collected from the online sources remains in the raw form of statements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires. The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of Iterative Analysis. The set of steps is known as Data Preprocessing. It includes -

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Data Preprocessing is necessary because of the presence of unformatted real-world data. Mostly real-world data is composed of -

- **Inaccurate data (missing data) -** There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometrics and much more.
- **The presence of noisy data (erroneous data and outliers) -** The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.

- **Inconsistent data -** The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.

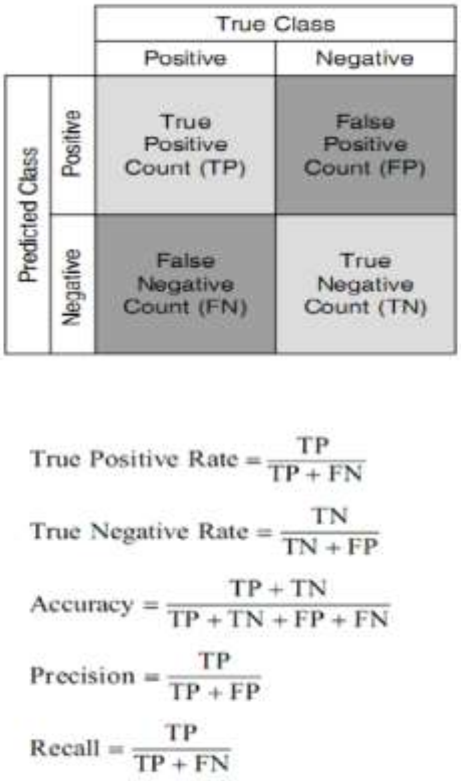
3. Implementing

In this work, a business intelligent model has been developed, to classify Dataset based on a specific business structure deal with Drug recommendation using a suitable machine learning technique. The model was evaluated by a scientific approach to measure accuracy, build our model.

4. Analysis

In this final phase, we will test our classification model on our prepared image dataset and also measure the performance on our dataset. To evaluate the performance of our created classification and make it comparable to current approaches, we use accuracy to measure the effectiveness of classifiers.

After model building, knowing the power of model prediction on a new instance, is very important issue. Once a predictive model is developed using the historical data, one would be curious as to how the model will perform on the data that it has not seen during the model building process. One might even try multiple model types for the same prediction problem, and then, would like to know which model is the one to use for the real-world decision making situation, simply by comparing them on their prediction performance (e.g., accuracy). To measure the performance of a predictor, there are commonly used performance metrics, such as accuracy, recall etc. First, the most commonly used performance metrics will be described, and then some famous estimation methodologies are explained and compared to each other. "Performance Metrics for Predictive Modeling In classification problems, the primary source of performance measurements is a coincidence matrix (**classification matrix or a contingency table**)". Above figure shows a coincidence matrix for a two-class classification problem. The equations of the most commonly used metrics that can be calculated from the coincidence matrix are also given in Fig 2.7.



		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 5.2.2: confusion matrix and formulae

As being seen in above figure, the numbers along the diagonal from upper-left to lower-right represent the correct decisions made, and the numbers outside this diagonal represent the errors. "The true positive rate (also called hit rate or recall) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count. The false positive rate (also called a false alarm rate) of the classifier is estimated by dividing the incorrectly classified negatives (the false negative count) by the total negatives. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

5. Flexibility

Sometimes you just don't want to use what is already there but you want to define something of your own (for example a cost function, a metric, a layer, etc.). Although Keras 2 has been designed in such a way that you can implement almost everything you want but we all know that low-level libraries provides more flexibility. Same is the case with TF. *You can tweak* TF much more as compared to Keras.

6. Functionality

Although Keras provides all the general purpose functionalities for building Deep learning models, it doesn't provide as much as TF. TensorFlow offers more advanced operations as compared to Keras. This comes very handy if you are doing a research or developing some special kind of deep learning models. Some examples regarding high level operations are**:**

7. Threading and Queues

Queues are a powerful mechanism for computing tensors asynchronously in a graph. Similarly, you can execute multiple threads for the same Session for parallel computations and hence speed up your operations.

8. Debugger

Another extra power of TF. With TensorFlow, you get a specialized debugger. It provides visibility into the internal structure and states of running TensorFlow graphs. Insights from debugger can be used to facilitate debugging of various types of bugs during both training and inference.

9. Control

The more control you have over your network, more better understanding you have of what's going on with your network. With TF, you get such a control over your network. You can control whatever you want in your network. Operations on weights or gradients can be done like a charm in TF.

5.3 NUMPY

Numpy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

Numpy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

5.3.1 OPERATIONS USING NUMPY

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

5.3.2 NumPy A REPLACEMENT FOR MATLAB

NumPy is often used along with packages like SciPy (Scientific Python) and Matplotlib (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.

The most important object defined in NumPy is an N-dimensional array type called ndarray. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index. Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called dtype).

5.3.3 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

5.3.4 ALGORITHEMS AND MODELS

5.3.4.1 CLUSTERING ALGORITHM (K-Means):

- K-means clustering is a method of quantization vectors, which is common in data harvesting of cluster analysis. Unpredictable algorithms usually deduct data sets using only input vectors without referring to displayed or specified consequences. The approach presents a hierarchical and fast way to evaluate a given collection of data across a clustering algorithm (implying k clusters) concluded by the centering of data.
- Finally, the aim of using k means clustering for decreasing an outside function acknowledged as the squared error function, which is given:

$$M=\sum j=1k\sum i=1n\|x\hat{}(j)i-yj\|\hat{}2$$

1.Distance Measure:

To figure out the similarities between the alternatives, this distance between the points takes as a standard metric. The basic separation computation utilizes the Euclidean measurement, that characterizes the span among a^i and b^i, where:

$$d=[\Sigma(ai-bi)\hat{}2]\hat{}1/2$$

2. Algorithm Steps:

The following approaches to this algorithm:

- Here K points are objecting state effective internal control, and the initial position is labeled.
- Each group object assigns to the nearest centroid.
- Recalculate the position of the K centroids after assigning all the objects.
- Phases 2 and 3 likely proceed. And why the category should differ from the one that minimizes the metric.

5.3.4.2 LINEAR REGRESSION ALGORITHM:

- There are many machine learning algorithms, and here, the equation is formed by Linear construction on the data with a direct curve correlation. This relationship is created between the objective variable and the free factors. The method is to acquire a regression model from a set of input values (calculated values).
- This process is a very straightforward machine learning approach, where the dependent variable is modeled as a linear order of predictors. More high-level regression methods and models include regression and multivariable regression, which is an acknowledged linear regression algorithm.
- At a simple linear regression method, one feature or single independent variable. So, the formula for linear regression is:
- $y=\theta0+\theta1X1$
- Suppose, there is only one variable, the model represented relationship formula is:
- $Y=\theta0+\theta1X+\theta2X+\theta3X\hat{}2+\theta4X\hat{}2++\theta mX\hat{}n$
- Here, n is the linear equation degree. As n (degree) increases, the feature complexity increases. The stated linear regression assessment used to forecast each number of reported case scenarios. The θ_m is the polynomial achieved by applying a sci-kit-learn Machine Learning algorithm in Python.

5.3.4.3 ARIMA MODEL:

- ARIMA, a model set up by Box and Jenkins, is furthermore called the B-J model, and it’s a mix of the AR (Auto-Regressive) model and the MA (Moving Average) model.

Econometricians normally use ARIMA models applied to figure time arrangement, and the ARIMA model is known as the most unpredictable and progressed time arrangement determining technique in the global. The generalized equation of the ARIMA model is:

- $Xt=\alpha 1Xt-1+\alpha 2Xt-2+\ldots+\alpha pXt-p+\mu t-\beta 1\mu t-1-\beta 2\mu t-2-\ldots-\beta q\mu t-q \,\|\, Xt=\nabla\text{\^{}}dyt$

- Here, Xt= new time series after steps difference of d, $\alpha 1,\alpha 2,\ldots\alpha p$= auto regressive coefficient, $\beta 1,\beta 2\ldots,\beta q$=moving average coefficient, ∇= backward difference operators, μt= random disturbance and yt= time series.

5.3.4.4 FORECASTING MODEL

- It utilized for anticipating time with three fundamental model parts: pattern, irregularity, and occasions. They consolidated in the accompanying condition:

$$Z(a)=P(a)+Q(a)+R(a)+\xi(a)$$

- Here, P(*a*)= piecewise straight or strategic development bend shaping non-intermittent time arrangement shifts, Q(*a*)= periodical variations(e.g. week after week/yearly irregularity), R(*a*)= effects of occasions (client gave) with unpredictable timetables, $\xi(a)$= blunder term considering any uncommon varieties not upheld by the model.

1.Piecewise straight or strategic development:

- The underlying one is called Nonlinear, Saturating Growth. It expressed in the form of the logistic growth model:
- $P(a)=C/(1+e^{(-k(a-m))}a)$
- Here, C= carrying capacity which is the maximum value of the curve, k= growth rate that reflects curve steepness, and m= a variable offset.
- This calculated condition empowers non-straight displaying development with immersion when the development pace of significant worth decreases with its development.

2. Seasonality:

- Seasonal effects q(a) are approximated by the following function:
- P is the period (365.25 for yearly data and 7 for weekly data),
- Parameters [a_1, b_1, ….., a_N, b_N] need to be estimated for a given N to model seasonality.

3. Occasion:

- Occasion incur predictable shocks to a time series.

5.3.5 IMPLIMENTATION

Forecasting and Severity Analysis of COVID-19 Source code.

ImportLib

```
Import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

The above piece of code defines the installation of libraries in which import warnings avoids the unnecessary warnings. Numpy are defined to work with datasets . Pandas are defined to work with arrays. Matplotlib are the low level graphs for visualization. Seaborn is used to choose the different colors.

```
df=pd.read_csv("complete.csv")
df.head()
df.tail()
df.isna().sum() f['NameofState/UT'].unique()
df.shape
```

The above piece of code defines the feature extraction. Here the csv file is defined and read with pandas. df.head() shows the initial values from the dataset. df.tail()shows the last values from the dataset. Isna () is used to detect the missing values.

```
sta=['Karnataka']
df1=df[df['NameofState/UT']==sta[0]]
df1.head()
df1.tail()
df1.shape
df1.reset_index(drop=True,inplace=True)
df1.head()
df1.tail()
df2=df1[['Date','TotalConfirmedcases']]
df2.head()
df2.describe()
```

The above piece of code defines the select the state.
Here df2=df1[['Date','TotalConfirmedcases']] defines the date and total confirmed cases. It outputs the same.

```
plt.figure(figsize=(18,8))
 df2.plot.line(x='Date',y='TotalConfirmedcases',figsize=(18,8))
 plt.show()
df2.plot.hist(x='Date',y='TotalConfirmedcases',figsize=(18,8))
plt.show()
```

The above piece of code defines the plotting the graph. Here, plt.figure(figsize=(18,8))
Is used to plot the graph. Here the figure size is 18 in width and 8 in length. Here the x-axis indicates the date and y-axis indicates the total confirmed cases.

```
from statsmodels.tsa.ar_modelimportAR
 X=df2['TotalConfirmedcases'].values
 print(len(X))
X=np.log(X)
#plt.plot(X)
train,test=X[0:X.shape[0]-30],X[X.shape[0]-30:]
print(train)
print(test)
```

The above piece of code defines the model implementataion. Here stats models are imported. statsmodels is a **Python module** that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. Here AR models are imported. In statistics, econometrics and signal processing, an **autoregressive** (**AR**) **model** is a representation of a type of random process; as such, it is used to describe certain time-varying processes in nature, economics, etc. The autoregressive model specifies that the output variable depends linearly on its own previous values and on a stochastic term (an imperfectly predictable term); thus the model is in the form of a stochastic difference equation. Here we train the dataset with the parameters namely Date, Name of State, Latitude and longitude, Total confirmed cases, Death etc..
Of size 269 kb of 4600 datset values which is 80%. Then test is used to check the validate performances of the training dataset. Then print the same.

```
model=AR(train)
model_fit=model.fit()
```

The above piece of code defines the autoregression model. Here we need to train the autoregression model.

```
predictions=model_fit.predict(start=len(train),end=len(train)+29,dynamic=True)
test predictions
```

The above piece of code defines the prediction making.

```
from sklearn.metricsimportmean_squared_error,r2_score
mse=mean_squared_error(test,predictions)
 print('MSE:%f'% mse)
from mathimportsqrt rmse=sqrt(mse)
print('RMSE:%f'% rmse)
 print(rmse)
print('AccuracyScore',1-rmse)
Y=df2['Date']
years=Y[len(train):]
len(predictions)
```

The above piece of code defines the calculation part. Here the mean squared error is calculated with the components of test result and predicted results. Even the root mean squared error is calculated with the components of mse results. By calculating these two parameter terms, the accuracy score can be predicted.

```
plt.figure(figsize=(18,8))
plt.plot(years,predictions,label='Predicted')
plt.plot(years,test,label='Actual'
```

The above piece of code defines the graphic visualization. Here the predicted results and the actual results are graphically shown in graphs.

CHAPTER 6

TESTING

6.1 SOFTWARE TESTING INTRODUCTION

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software .Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification & validation

6.2 Explanation for SDLC & STLC

SDLC: The software development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

6.3 PHASES OF SOFTWARE DEVELOPMENT

- Requirement Analysis
- Software design
- Development or Coding
- Testing
- Maintenance

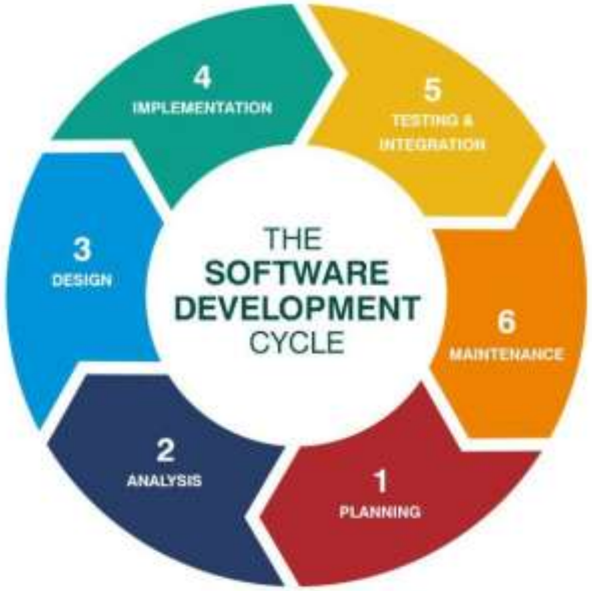


Fig 6.3 Software Deveiopment cycle

6.3.1 REQUIREMENT ANALYSIS

The requirements of a desired software product are extracted. Based the business scenario the SRS (Software Requirement Specification) document is prepared in this phase.

6.3.2 DESIGN

Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues for the software. Design phase is concerned with making sure the software system will meet the requirements of the product.

There are 2 stages in design,

HLD – High Level Design

LLD – Low Level Design

HLD – gives the architecture of the software product to be developed and is done by architects and senior developers.

LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

6.3.3 TESTING

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding defects.

6.3.4 MAINTANANCE

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

6.4 SDLC MODELS

6.4.1 WATER FALL MODEL

It will be executing one by one of the SDLC process. The design Starts after completing the requirements analysis coding begins after design. It is a traditional model It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards (like a waterfall), through the different phases.

6.4.2 PROTOTYPE MODEL

Developed from the sample after getting good feedback from the customer. This is the Valuable mechanism for gaining better understanding of the customer needs

6.4.3 Rapid application development model(RAD):

This mechanism will develop from already existing one. If The New requirement is matching in already existing requirement, will develop from that**.**

6.4.4 SPIRAL MODEL

This mechanism is update the application version by version. All the SDLC process will update version by version**.**

6.4.5 V-MODELV:

V model is a process where the development and testing phases can do parallely. For every development phase there is a testing phase. Development phases are called as verification whereas testing phases are called as validation

6.5 STLC (Software Testing Life Cycle): Testing itself has many phases i.e. is called as STLC. STLC is part of SDLC

- Test Plan
- Test Development
- Test Execution
- Analyze Results
- Defect Tracking
- Summaries Report

6.5.1. TEST PLAN

It is a document which describes the testing environment, purpose, scope, objectives, test strategy, schedules, mile stones, testing tool, roles and responsibilities, risks, training, staffing and who is going to test the application, what type of tests should be performed and how it will track the defects.

6.5.2. TEST DEVELOPMENT

Preparing test cases, test data, Preparing test procedure, Preparing test scenario, Writing test script

6.5.3 TEST EXECUTION

In this phase we execute the documents those are prepared in test development phase

6.5.4 ANALYZE RESULT

Once executed documents will get results either pass or fail. we need to analyze the results during this phase.

6.5.5. DEFECT TRACKING

Whenever we get defect on the application we need to prepare the bug report file and forwards to Test Team Lead and Dev Team. The Dev Team will fix the bug. Again we have to test the application. This cycle repeats till we get the software without defects.

6.6 TYPES OF TESTING:

- White Box Testing
- Black Box Testing
- Grey box testing

6.6.1WHITE BOX TESTING

White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

6.6.2 BLACK BOX TESTING

Its also called as behavioral testing. It focuses on the functional requirements of the software. Testing either functional or non functional without reference to the internal structure of the component or system is called black box testing.

6.6.3 GREY BOX TESTING

Grey box testing is the combination n of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

6.7 LEVEL OF TESTING USED IN PROJECT

6.7.1 Unit testing

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.**

6.7.2 Integration testing

All module which make application are tested . Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirement**.**

6.7.3 System testing

To test the complete system in terms of functionality and non functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

6.7.4 Functional testing

The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

6.7.5 Alpha testing

Alpha testing is final testing before the software is released to the general public. This testing is conducted at the developer site and in a controlled environment by the end user of the software.

6.7.6 Beta testing

The beta test is conducted at one or more customer sites by the end user of the software. The beta test is conducted at one or more customer sites by the end user of the software.

6.8 UNIT TESTING CASES

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.**

List of Test case (SAMPLE TEST CASE)

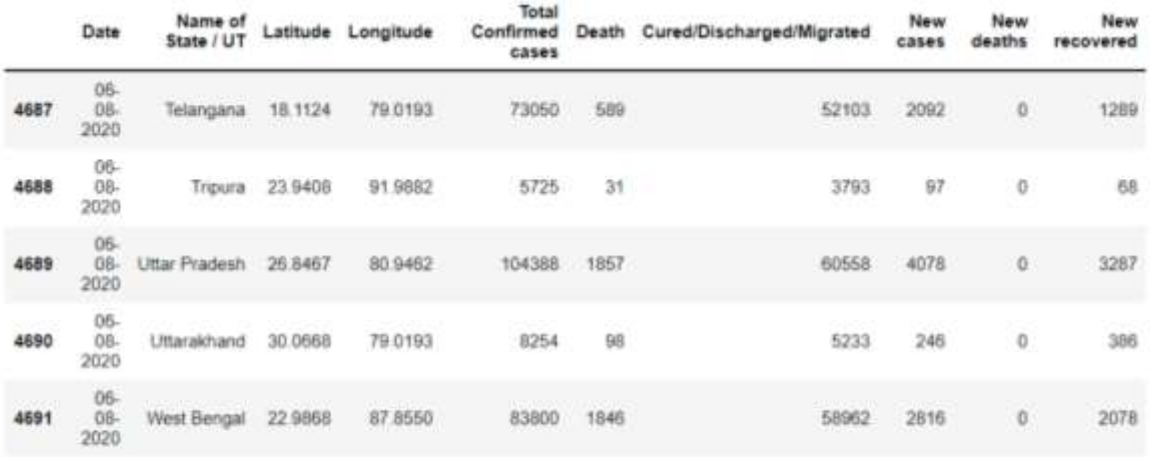
Id	**Test Case Title**	**Test Input**	**Result**	**Remarks**
TC_1	Data Upload Dataset File Path	File uploaded	Successfully	Pass
TC _2	Data Cleaning	Raw Dataset	Cleaned Data	Pass
TC _3	Data Preparation for Training	Dataset and Split-Ratio	Train-set and Test-set created successfully	Pass
TC _4	Model Construction and Training	Training Algorithm And Train-set	Model trained successfully using train-set	Pass
TC _5	Model Validation	Trained Model And Test-set	Display model validation parameters with its values	Pass
TC _6	Display Result	Model Performance Statistics	Classification accuracy and error rate with plot	Pass

Fig 6.8.1 : Sample test cases

CHAPTER 7

RESULTS

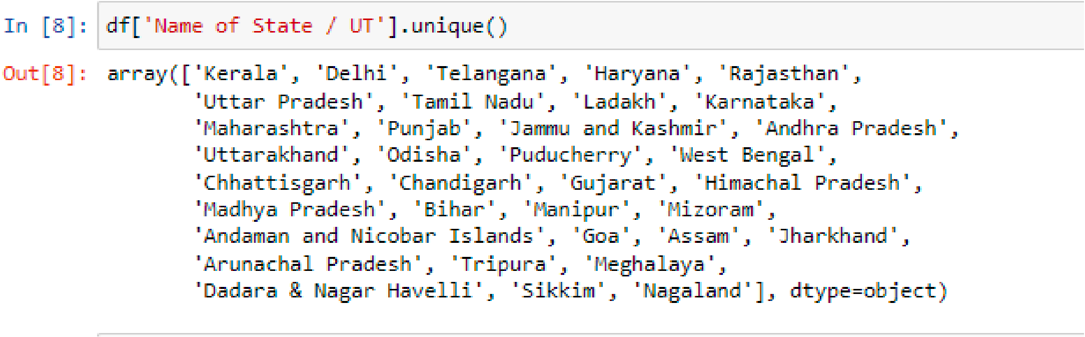
A result is the final consequence of actions or events expressed qualitatively or quantitatively. Performance analysis is an operational analysis, is a set of basic quantitative relationship between the performance quantities.



	Date	Name of State / UT	Latitude	Longitude	Total Confirmed cases	Death	Cured/Discharged/Migrated	New cases	New deaths	New recovered
4687	06-08-2020	Telangana	18.1124	79.0193	73050	589	52103	2092	0	1289
4688	06-08-2020	Tripura	23.9408	91.9882	5725	31	3793	97	0	68
4689	06-08-2020	Uttar Pradesh	26.8467	80.9462	104388	1857	60558	4078	0	3287
4690	06-08-2020	Uttarakhand	30.0668	79.0193	8254	98	5233	246	0	386
4691	06-08-2020	West Bengal	22.9868	87.8550	83800	1846	58962	2816	0	2078

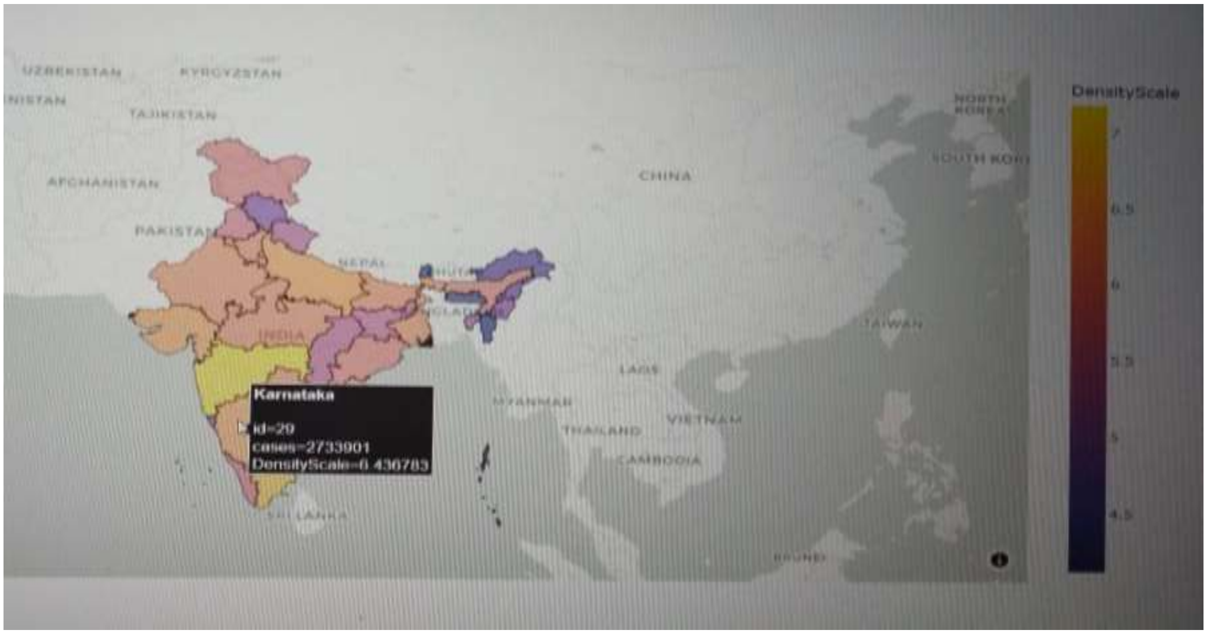
Fig 7.1 dataset used in the project

In Data-set analysis, we apply the clustering algorithm Logistic Regression algorithm to classify the dangerous region in this India. Here, we take the mortal rate, recovery rate, COVID-19 positive confirmation rate as a feature for algorithm. There are 4691 datum in this dataset.



```
In [8]: df['Name of State / UT'].unique()

Out[8]: array(['Kerala', 'Delhi', 'Telangana', 'Haryana', 'Rajasthan',
               'Uttar Pradesh', 'Tamil Nadu', 'Ladakh', 'Karnataka',
               'Maharashtra', 'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh',
               'Uttarakhand', 'Odisha', 'Puducherry', 'West Bengal',
               'Chhattisgarh', 'Chandigarh', 'Gujarat', 'Himachal Pradesh',
               'Madhya Pradesh', 'Bihar', 'Manipur', 'Mizoram',
               'Andaman and Nicobar Islands', 'Goa', 'Assam', 'Jharkhand',
               'Arunachal Pradesh', 'Tripura', 'Meghalaya',
               'Dadara & Nagar Havelli', 'Sikkim', 'Nagaland'], dtype=object)
```

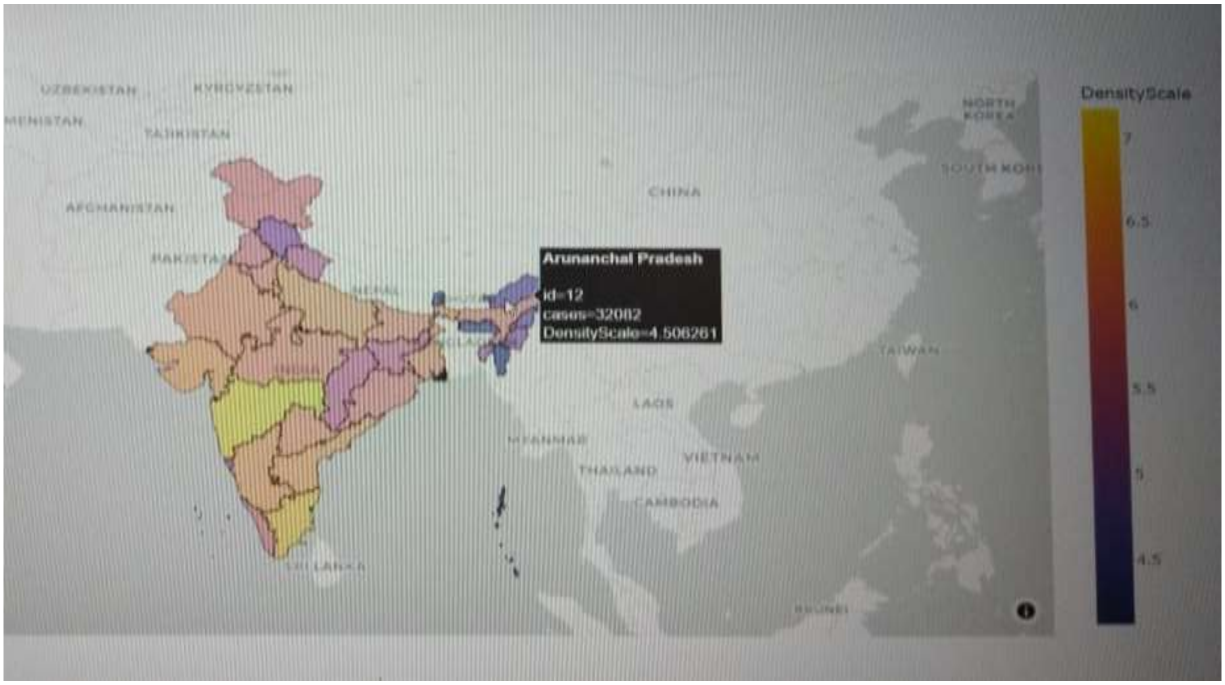
Fig 7.2 Name of State consider in the dataset



Karnataka

id=29
cases=2733901
DensityScale=6.436783

Fig 7.3 Density scale of the Karnataka

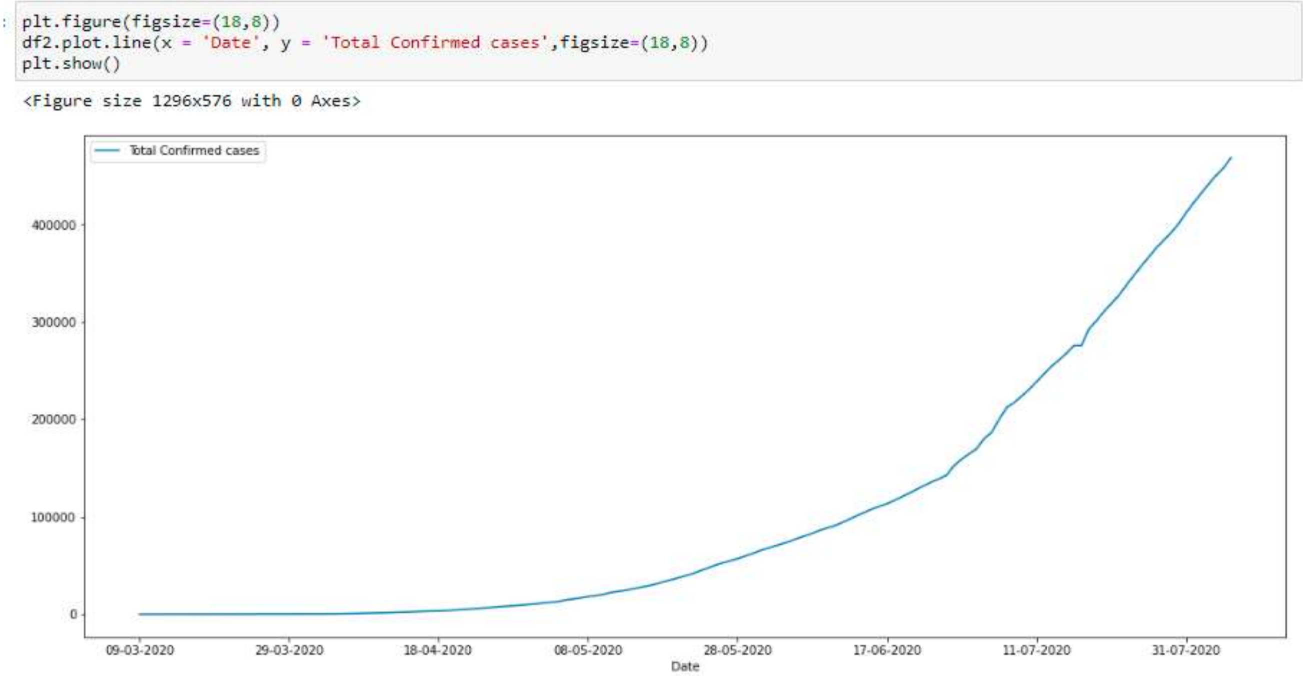


Arunanchal Pradesh

id=12
cases=32082
DensityScale=4.506261

Fig 7.4 Density scale of the Arunanchal Pradesh

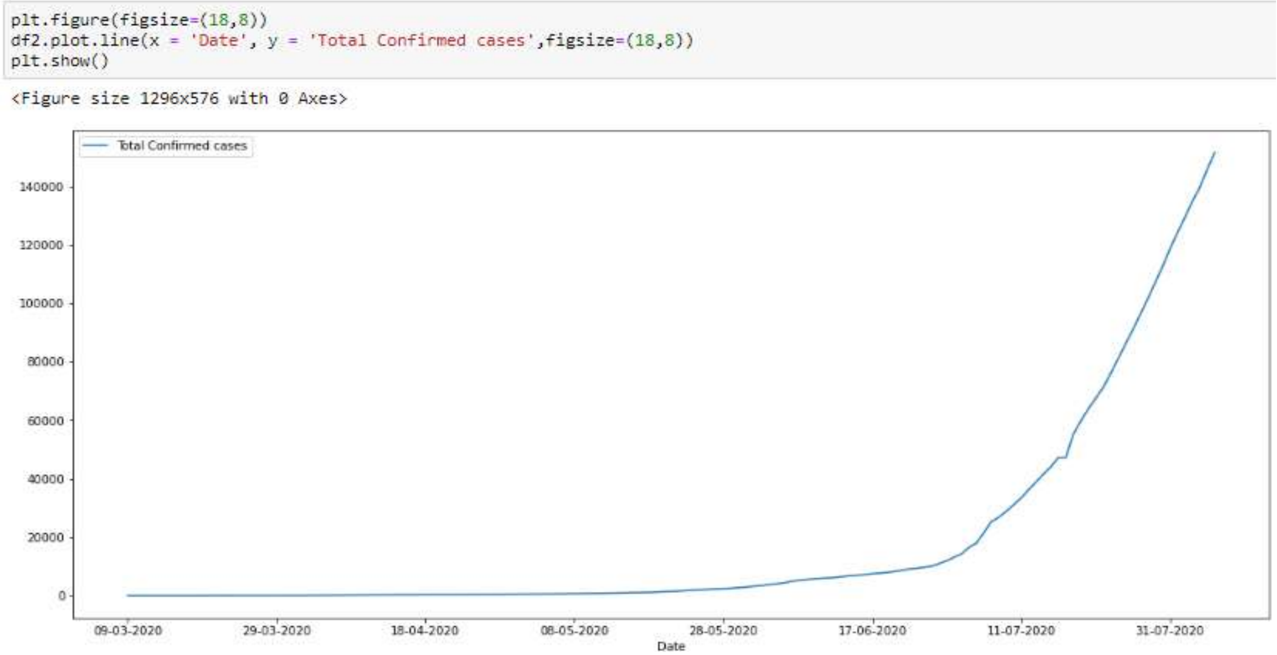
The range begins from 0 which the lowest number of cases to the highest 7, and differentiated by the color as mention in the fig 7.3 and fig 7.4. It hover detail like state name , id , number of cases and the density scale.



```
plt.figure(figsize=(18,8))
df2.plot.line(x = 'Date', y = 'Total Confirmed cases',figsize=(18,8))
plt.show()
```

```
<Figure size 1296x576 with 0 Axes>
```

Fig 7.5 graph of increase cases in Maharashtra



```
plt.figure(figsize=(18,8))
df2.plot.line(x = 'Date', y = 'Total Confirmed cases',figsize=(18,8))
plt.show()
```

```
<Figure size 1296x576 with 0 Axes>
```

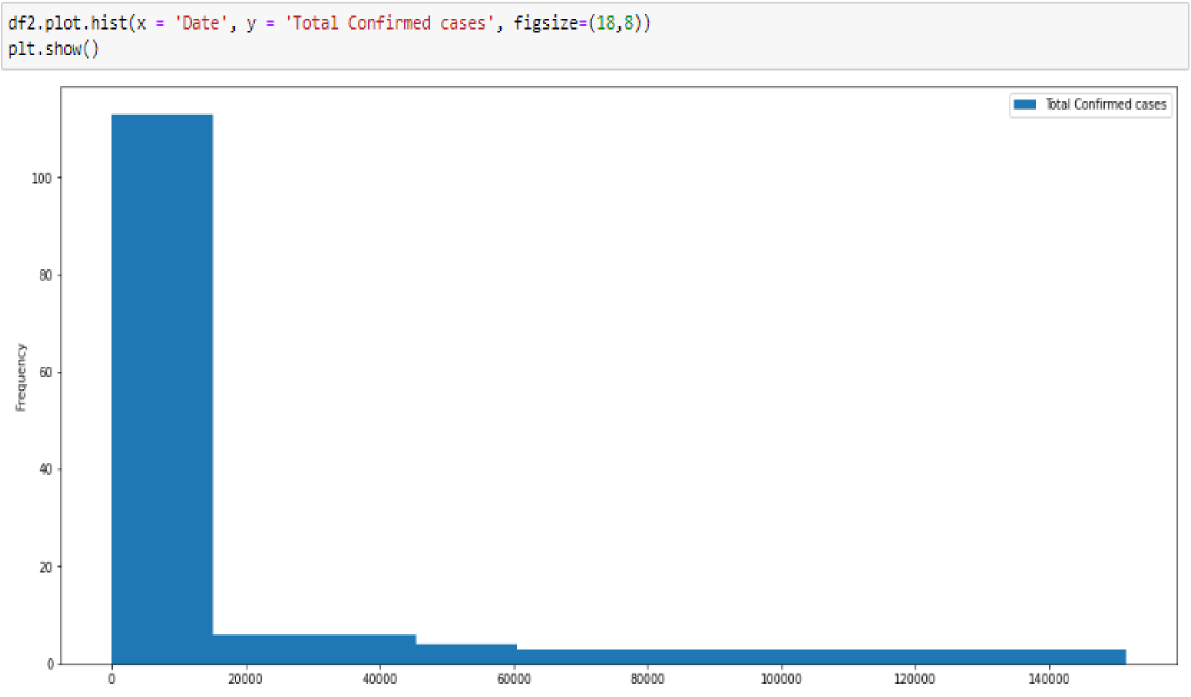
Fig 7.6 graph of increase cases in Karnataka

Fig 7.5 shows Total Confirmed cases in Maharashtra in y axis and Date in x axis. It is clearly seen an increase in the number of cases as the days increase. Which tell u the spread rate of covid in Karnataka .

Fig 7.6 shows Total Confirmed cases in Maharashtra in y axis and Date in x axis. It is clearly seen an increase in the number of cases as the days increase. Which tell u the spread rate of covid in Maharastra .

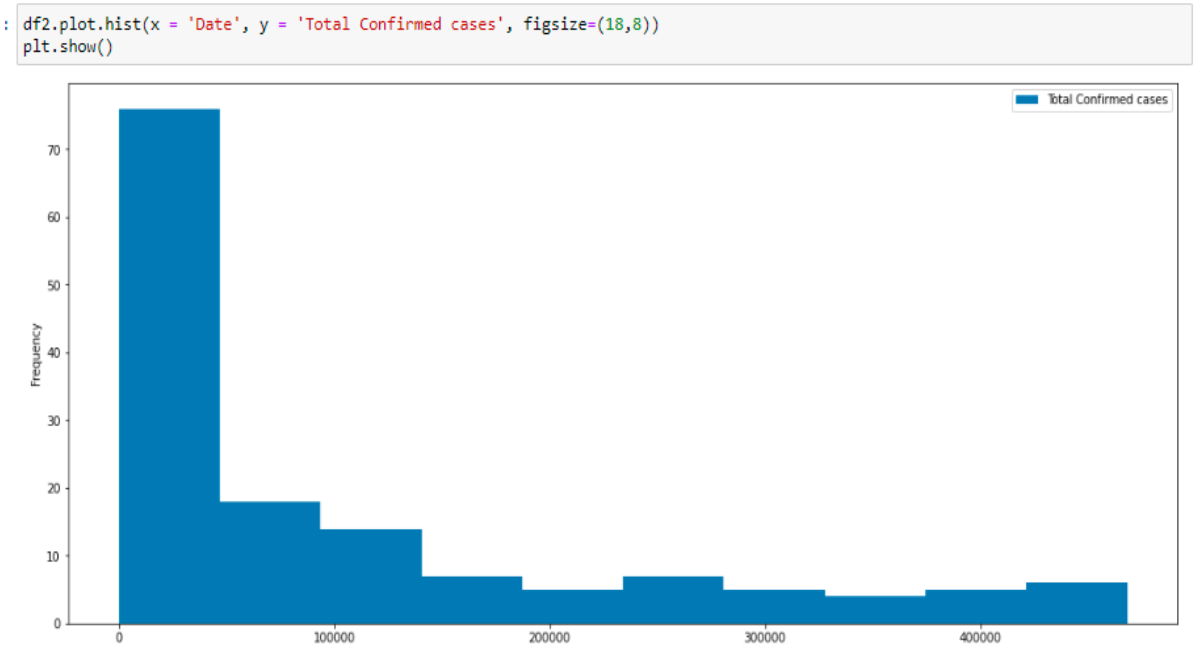
Fig 7.7 and 7.8 Gives a histogram representation, x axis is number of cases and y axis is the frequency in Karnataka and Maharastra

Fig 7.5 and 7.6 shows the number of cases reached in a some duration of days (100000 number of cases has a duration of 15 days)



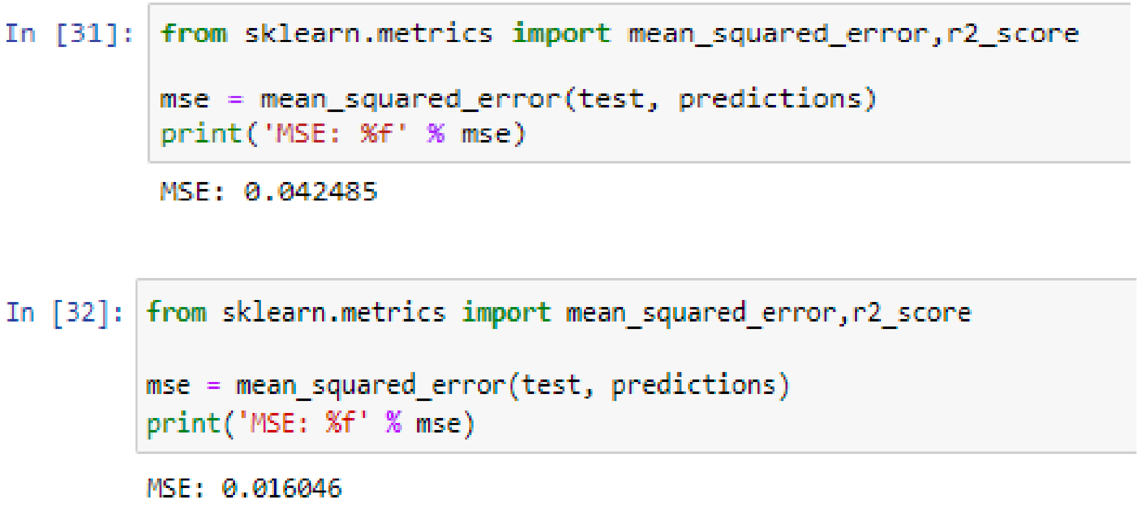
```
df2.plot.hist(x = 'Date', y = 'Total Confirmed cases', figsize=(18,8))
plt.show()
```

Fig 7.7 histogram representation of cases in Karnataka.



```
: df2.plot.hist(x = 'Date', y = 'Total Confirmed cases', figsize=(18,8))
  plt.show()
```

Fig 7.8 histogram representation of cases in Maharashtra

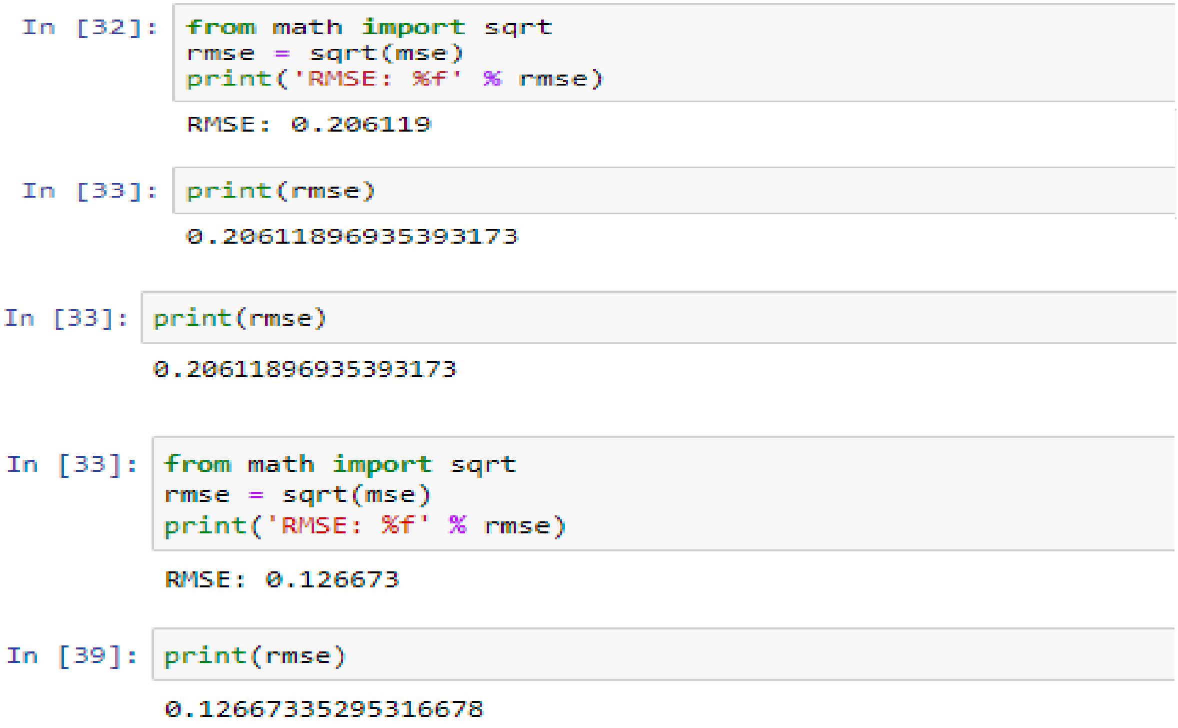


```
In [31]: from sklearn.metrics import mean_squared_error,r2_score

         mse = mean_squared_error(test, predictions)
         print('MSE: %f' % mse)

         MSE: 0.042485
```

```
In [32]: from sklearn.metrics import mean_squared_error,r2_score

         mse = mean_squared_error(test, predictions)
         print('MSE: %f' % mse)

         MSE: 0.016046
```

Fig 7.9 Mean Square Error

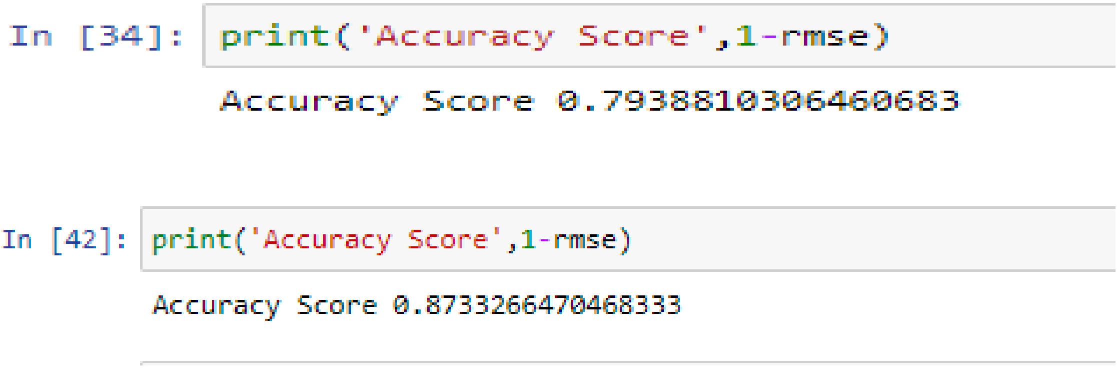
The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It’s called the mean squared error as you’re finding the average of a set of errors. The lower the MSE, the better the forecast.



```
In [32]: from math import sqrt
         rmse = sqrt(mse)
         print('RMSE: %f' % rmse)

         RMSE: 0.206119

In [33]: print(rmse)

         0.20611896935393173

In [33]: print(rmse)

         0.20611896935393173

In [33]: from math import sqrt
         rmse = sqrt(mse)
         print('RMSE: %f' % rmse)

         RMSE: 0.126673

In [39]: print(rmse)

         0.12667335295316678
```

Fig 7.10 Root Mean Square Error

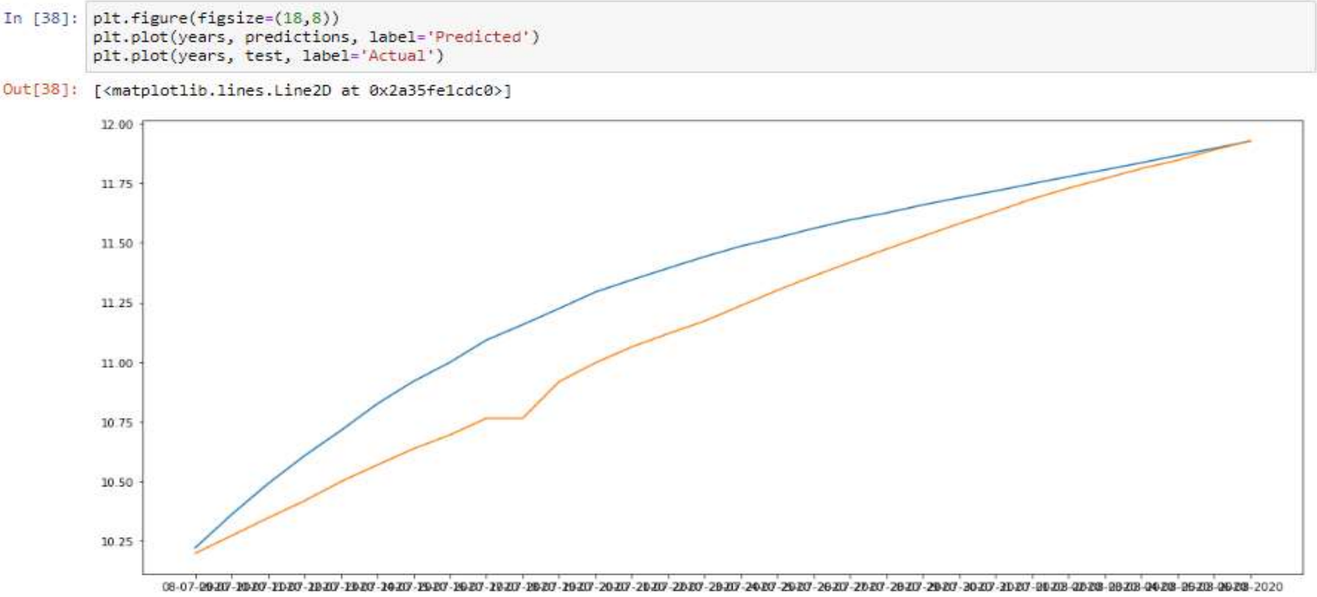
Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit



```
In [34]: print('Accuracy Score',1-rmse)

         Accuracy Score 0.7938810306460683

In [42]: print('Accuracy Score',1-rmse)

         Accuracy Score 0.8733266470468333
```

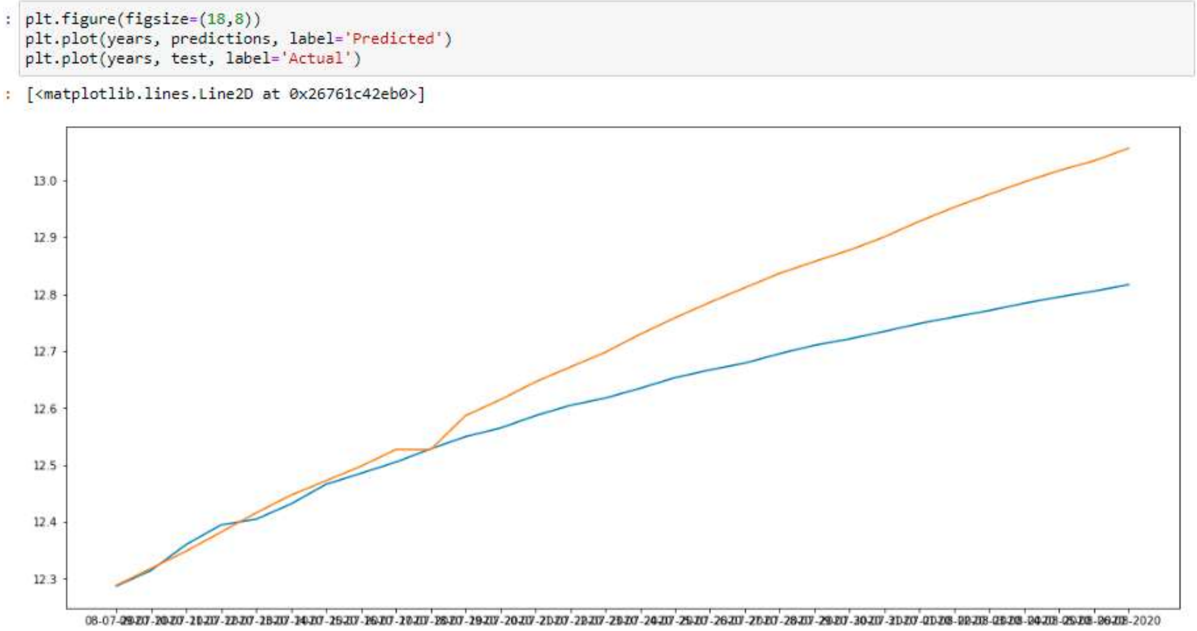
Fig 7.11 Accuracy Score

Fig 7.8 Show Accuracy is the most intuitive performance measure and it is ratio of correctly predicted observation to the total observations.



```
In [38]: plt.figure(figsize=(18,8))
         plt.plot(years, predictions, label='Predicted')
         plt.plot(years, test, label='Actual')

Out[38]: [<matplotlib.lines.Line2D at 0x2a35fe1cdc0>]
```

Fig 7.12 Predicted and Actual line graph



```
: plt.figure(figsize=(18,8))
  plt.plot(years, predictions, label='Predicted')
  plt.plot(years, test, label='Actual')

: [<matplotlib.lines.Line2D at 0x26761c42eb0>]
```

Fig 7.13 Predicted and Actual line graph

It is graph of both predicted and actual dataset which is plotted. The red is predicted and blue is actual.

CHAPTER 8

CONCLUSION AND FUTURE WORKS

In Forecasting and severity analysis of COVID-19 research, we evaluate the state of COVID-19 as a result of its rapid growth in the affected cases and forecast new numbers for the following week with different models. The various forecasting techniques clearly show the increasing rate of the newly confirmed cases. With all the researched methods, we have found Facebook's Prophet model to be the perfect one to show the accurate forecasting with the least RMSE value. With the increasing number of confirmed cases, it is quite impossible to bring everything under control for a country. So it is essential to create public awareness, follow them strictly with the help of laws, and provide necessary medical resources for the general people

REFERENCES

1. X. Wang and others , ”A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization From Chest CT,” in IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2615-2625, Aug. 2020, doi: 10.1109/TMI.2020.2995965.

2. T. Turki and Z. Wei, “A greedy-based oversampling approach to improve the prediction of mortality in MERS patients,” 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, 2016, pp. 1-5.

3. E. Kim, S. Lee, J. H. Kim, Y. T. Byun, H. Lee and T. Lee, “Implementation of novel model based on Genetic Algorithm and TSP for path prediction of pandemic,” 2013 International Conference on Computing, Management and Telecommunications (ComManTel).

4. N. Zheng , “Predicting COVID-19 in China Using Hybrid AI Model,” in IEEE Transactions on Cybernetics.

5.“Home,” Humanitarian Data Exchange. [Online].
Available: https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases. [Accessed: 24-Aug-2020].

6. A. Jain, A. Rajavat and R. Bhartiya, ”Design, Analysis and Implementation of Modified K-Mean Algorithm for Large Data-Set to Increase Scalability and Efficiency,” 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, 2012, pp. 627- 631, doi: 10.1109/CICN.2012.95.

7. Pham, D. & Dimov, Stefan & Nguyen, Cuong. (2005). Selection of K in K -means clustering. Proceedings of The Institution of Mechanical Engineers Part C-journal of Mechanical Engineering Science - PROC INST MECH ENG C-J MECH E. 219.

8. H. Li and S. Yamamoto, ”Polynomial regression based model-free predictive control for nonlinear systems,” 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016, pp. 578-582, doi: 10.1109/SICE.2016.7749264

9. S. Karthika, V. Margaret and K. Balaraman, ”Hybrid short term load forecasting using ARIMASVM,” 2017 Innovations in Power and Advanced Computing Technologies (i-PACT),

Vellore, 2017, pp. 1-7, doi: 10.1109/IPACT.2017.824506010

10. E. Kim, S. Lee, J. H. Kim, Y. T. Byun, H. Lee and T. Lee, “Implementation of novel model based on Genetic Algorithm and TSP for path prediction of pandemic,” 2016 International Conference on Computing, Management and Telecommunications (ComManTel).

11. N. Zheng, “Predicting COVID-19 in China Using Hybrid AI Model,” in IEEE Transactions on Cybernetics.

12. 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, 2012, pp. 627- 631, doi: 10.1109/CICN.2012.95.