# WATER QUALITY PREDICTION USING MACHINE LEARNING

**A PROJECT REPORT**

**OF PROJECT-2 (IT 892)**

**BACHELOR OF TECHNOLOGY**
In
Information Technology

(From **Maulana Abul Kalam Azad University of Technology, West Bengal**)

SUBMITTED BY


Saumojit Bhattacharjee (13000216049)

Sohini Majumdar (13000216029)

Suchismita Ghosh (13000216023)

Rajdeep Swarnakar (13000216067)

Under the Guidance of

**Prof. Piyal Sarkar**

**Department of Information Technology**
**Techno Main Salt Lake**
**Kolkata -700091**

*Department of Information Technology*
*Techno Main Salt Lake*
*EM-4/1, Salt Lake, Kolkata-700091*

# BONAFIDE CERTIFICATE

Certified that this report for the project titled "WATER QUALITY PREDICTION USING MACHINE LEARNING" is a part of the final year project work being carried out by " SAUMOJIT BHATTACHARJEE , SOHINI MAJUMDAR , SUCHISMITA GHOSH AND RAJDEEP SWARNAKAR " as partial fulfillment for the degree of Bachelor of Technology in Information Technology, Maulana Abul Kalam University of Technology, West Bengal, under my supervision.

Full Signature of the Candidates (with date)

1. Saumojit Bhattacharjee 11.6.20
2. Sohini Majumdar. 11.06.2020.
3. Suchismita Ghosh. 11.06.2020.
4. Rajdeep Swarnakar. 11/06/2020

Priyal Sinhar

(Signature of the Supervisor)

Dr. Subhamita Mukherjee
HOD, Information Technology

# ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our respected and esteemed person Prof. Piyal Sarkar of the Dept. of Information Technology, Techno Main, Salt Lake, for his/ her valuable guidance, encouragement and helping us to pursue with our project work. His/ Her useful suggestions for this work and co-operative behavior are sincerely acknowledged.

We would like to express our sincere thanks to the teaching fraternity of the Department of Information Technology, for giving us this opportunity to undertake this project and also supporting us whole heartedly.

We also wish to express our gratitude to the HOD and all our teachers of the Department of Information Technology for their kind hearted support, guidance and utmost endeavor to groom and develop our academic skills.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during the effort in shaping this concept till now.

Full Signature of the Candidates (with date)

1. Saumojit Bhattacharyya 11.6.20
2. Sohini Majumdar. 11.06.2020.
3. Suchismita Ghosh. 11.06.2020.
4. Rajdeep Swarnakar. 11/06/2020

# ABSTRACT

WATER IS LIFE!

Water quality has a direct impact on public health and the environment and eco-system of the Earth. Water is used for various practices, such as drinking, agriculture, and industry. India is facing a serious problem of natural resource scarcity, especially that of water in view of population growth and economic development. Our project is all about exploring if there are any associations between the water quality data and the other developmental data. If there is, then what is the extent and predicting if Any kind of water quality issues in India, based on the developmental and water quality data that is available right now. A Web-Application with User-Friendly UI is built which contains all the necessary things to detect and monitor and visualize the Water Quality in India. So That the Government can take appropriate measures to prevent the problem.

As We believe that Prevention is much better than Cure!

# TABLE OF CONTENTS


| TITLE | PAGE NO. |
|---|---|
| TITLE PAGE | |
| CERTIFICATE | |
| ACKNOWLEDGEMENT | i |
| ABSTRACT | ii |
| TABLE OF CONTENTS | iii |
| LIST OF TABLES | iv |
| LIST OF FIGURES | v |
| LIST OF SYMBOLS | vi |
| **CHAPTERS** | |
| **1. INTRODUCTION** | 1 |
| 1.1. INSPIRATION | 3 |
| **2. LITERATURE REVIEW** | 4 |
| 2.1 WATER QUALITY: GENERAL ASPECTS | 5 |
| 2.2. RIVER WATER | 6 |
| 2.3. HUMAN IMPACT ON RIVER WATER | 6 |
| 2.4. INDIAN RIVERS | 6 |
| 2.4.1. WATER RESOURCES OF INDIA | 7 |
| **3. DEFINITION OF THE PROBLEM WITH MODULES AND FUNCTIONALITIES** | 8 |
| 3.1. DEFINITON OF THE PROBLEM | 9 |
| 3.2. MODULES AND FUNCTIONALITIES | 10 |
| **4.SOFTWARE DESIGN** | 11 |
| 4.1. METHODOLOGY | 12 |
| 4.2. DFD(Level-0,1,2) | 13 |
| 4.3. UML-USE CASE DIAGRAM | 15 |
| **5.SOFTWARE AND HARDWARE REQUIREMENTS** | 16 |
| **6.CODE TEMPLATES** | 18 |
| 6.1. CREATION OF THE MODEL USING PYTHON | 19 |
| 6.2. CREATION OF MICRO-WEB FRAMEWORK USING FLASK | 28 |
| 6.3. CREATION OF UI | 30 |
| 6.4. LAUNCHING IT IN THE LOCAL WEB SERVER | 37 |
| 6.5. LAUNCHING IT INTO AWS EC2 INSTANCE | 39 |
| **7.TESTING** | 49 |
| **8.OUTPUT SCREENS** | 52 |
| **9.CONCLUSION** | 55 |
| **10.REFERENCES** | 57 |

# LIST OF TABLES


| TABLE NO. | PARTICULARS | PAGE |
|---|---|---|
| 1 | LOADING THE DATASET | 19 |
| 2 | AFTER STANDARDIZATION | 22 |

# LIST OF FIGURES


| FIGURE NO. | PARTICULARS | PAGE NO. |
|---|---|---|
| 1 | FLOW DIAGRAM | 12 |
| 2.1,2.2,2.3 | DFD: DATA FLOW DIAGRAM (LEVEL 0,1,2) | 13 |
| 3 | UML: USE CASE DIAGRAM | 15 |
| 4 | BEFORE PRE-PROCESSING | 20 |
| 5 | AFTER PRE-PROCESSING | 20 |
| 6 | STANDARDIZATION OF DATA | 21 |
| 7 | HEAT MAP | 22 |
| 8 | HISTOGRAM | 23 |
| 9 | BOXPLOT | 24 |
| 10 | BARCHART | 24 |
| 11,12 | RANDOM FOREST METHODOLOGY | 25 |
| 13,14 | VALIDATION | 28 |
| 15 | ROUTING | 29 |
| 16 | WEB UI FOR HOME PAGE | 32 |
| 17 | WEB UI FOR DOCUMENTATION PAGE | 33 |
| 18 | WEB UI FOR SLIDER.CSS | 37 |
| 19 | RUNNING THE WEB-APP | 37 |
| 20 | WEB UI FOR PREDICT PAGE | 38 |
| 21 | ACTIVITY IN ANACONDA PROMPT | 38 |
| 22 | AMI | 39 |
| 23 | CHOOSING INSTANCE TYPE | 40 |
| 24 | KEY PAIR | 41 |
| 25 | INSTANCE DASHBOARD | 41 |
| 26 | PUTTYGEN | 42 |
| 27 | WINSCP | 42,43 |
| 28,29,30 | PUTTY CONFIGURATION | 44 |
| 31 | INSTALLING PIP3 AND PYTHON LIBRARIES | 45 |
| 32,33,34 | SECURITY GROUP | 46 |
| 35,36 | NETWORK INTERFACE | 47 |
| 37 | EXECUTION FROM THE EC2 INSTANCE | 47 |
| 38 | MONITORING | 48 |
| 39 | EXECUTION | 48 |
| 40.A, 40. B | 1ST TEST CASE | 50 |
| 41.A, 41. B | 2ND TEST CASE | 51 |
| 42 | HOME/PREDICT PAGE | 53 |
| 43 | DOCUMENTATION PAGE | 53 |
| 44 | GRAPHSLIDER PAGE | 54 |
| 45 | RANDOMFOREST REGRESSOR FIT AND PREDICT | 54 |

# LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

| SYMBOLS, ABBREVIATIONS AND NOMENCLATURE | MEANING |
|---|---|
| D.O. | DISSOLVED OXYGEN |
| PH | POTENTIAL OF HYDROGEN |
| B.O.D. | BIOCHEMICAL OXYGEN DEMAND |
| E.C | ELECTRICAL CONDUCTIVITY |
| AMI | AMAZON MACHINE INSTANCE |
| DFD | DATA FLOW DIAGRAM |
| UML | UNIFIED MODELLING LANGUAGE |

# CHAPTER:1

# INTRODUCTION

# INTRODUCTION

Water is LIFE! Water quality has a direct impact on public health and the environment and eco-system of the Earth. Water is used for various practices, such as drinking, agriculture, and industry. India is facing a serious problem of natural resource scarcity, especially that of water in view of population growth and economic development. Most of fresh water bodies all over the world are getting polluted, thus decreasing the portability of water. All life is dependent on water and exists in nature in many forms like ocean, river, lake, clouds, rain, snow and fog etc. However, strictly speaking chemically pure water does not exist for any appreciable length of time in nature.

A lake is a large body of water surrounded by land, inhabited by various aquatic life forms, for all practical purpose, pure water is considered to that which has low dissolved or suspended solids and obnoxious gases as well low in biological life. Such high quality of water may be required only for drinking purposes while for other uses like agriculture and industry, the quality of water can be quite flexible and water polluted up to certain extent in general sense can be regarded as pure. The health of lakes and their biological diversity are directly related to health of almost every component of the ecosystem. Lakes are also subjected to various natural processes taking place in the environment like the hydrologic cycle, with unprecedented development activities; human beings are responsible for choking several lakes to death. Storm water runoff and discharge of sewage into the lakes are few of the common causes where various nutrients enter the aquatic ecosystems resulting in their death.

Of all the water quality issues facing lakes everywhere, eutrophication is of great concern. Eutrophication is a term used to describe the aging of a lake, resulting due to the accumulation of nutrients, sediments, silt and organic matter in the lake from the surrounding watershed. The role of vegetation and sediments as sources and sink of nutrients has been demonstrated. It describes the biological reaction of aquatic systems to nutrient enrichment, the eventual consequence of which is the development of primary production to nuisance proportions. The main cause is excessively adding of phosphorus and nitrogen resulting in high algal biomass, dominance by cyanobacteria and loss of macrophytes.

There has been an increased interest among the public about the Environment and living conditions in India. Especially, after since many manufacturing units are being planned, people are

worried about how it will affect the underground water quality and the environment. Government of India, under the Ministry of Drinking Water and Sanitation has released the Water Quality Affected Data for 2009, 2010, 2011 and 2012. The objective here is to analyse this data to figure out whether there is any connection between the development effort and the quality of water getting affected. This effort will identify such associations and create awareness such that people and the Government can act in time to avoid further deterioration of the water resources.

Let's explore if there are any associations between the water quality data and the other developmental data. If there is, then what is the extent (visualisation) and how can we address it (prescriptive).

Let's predict if there are GOING TO BE water quality issues in areas that are not affected right now, based on the developmental and water quality data that is available right now. Prevention is better than cure!

## 1.1. INSPIRATION:

- Let's explore if there are any repetitive patterns of water quality degradation in the same area for multiple years.
- As a whole, which areas in India has a lot of water quality degradation issues over the years.
- As a whole, for the country, is the water quality degrading or upgrading (number of instances reported of water quality getting affected)?
- Let's explore if there are any associations between the water quality data and the other developmental data. If there is, then what is the extent (visualisation) and how can we address it (prescriptive).
- Let's predict if there are going to be water quality issues in areas that are not affected right now based on the developmental and water quality data that is available right now. Prevention is better than cure!

It would be great if we could have water quality and industrial/development experts in this analysis, so that they can contribute their valuable inputs!

# CHAPTER :2

# LITERATURE REVIEW

# LITERATURE REVIEW

## 2.1 WATER QUALITY: GENERAL ASPECTS:

Water is an invaluable resource and the benefits to mankind from proper management of this resource as well as the disastrous consequences of its mismanagement are very well known. The public awareness about water quality is at its zenith now but at the same time, the danger signals have not shown any abatement. Study of water quality has remained an important pre-occupation with the environmentalist both from the practical and the academic viewpoints.

The concept of quality, however, raises a number of questions which are often controversial because of wide differences in technological and individual perceptions. The economic and the aesthetic considerations describing water quality also come to blows. Technologically, water quality can be catalogued in terms of appropriate physical, chemical and bacteriological parameters which must be accurate, 2 unambiguous, quantitative and reproducible. Many authors prefer to describe water quality in terms of the processes occurring in a catchment ecosystem. This consists of a source of water such as a river with its associated drainage areas, and is a complex combination of biotic and abiotic components and the interactions amongst them.

A simplistic picture of the different hydrological 3 processes occurring in a catchment ecosystem is shown in Fig. 2.1. Water entering a catchment is intercepted by the vegetation and subsequently evaporated, or reaches the ground as through fall and stem-flow. When there is no vegetation, the precipitation comes to the ground directly, although there may be some interception by surface litter. From the ground water infiltrates into the soil or is retained as surface storage which may move downward as surface runoff or is slowly evaporated. Soil moisture may be lost through evaporation and plant transpiration, or percolates downward to form the mass of groundwater. The groundwater is subject to similar losses through plant transpiration and evaporation, or through upward flow by capillary action into the unsaturated soil or through deep transfer, or through base-flow to the river channel. The river also receives contributions from surface runoff, through-flow and inter-flow. The quality of the river water is much different from that of the precipitation water. The q major mechanisms influencing water quality are shown in Fig. 2.2. It clearly indicates (hat the river water quality evolve through a complex series of interactions with soil, rock, and biota of the catchment ecosystem. These interactions are always in a delicate balance and a slight modification to the catchment such as alterations in the land use pattern is likely to generate significant changes in water quality.

Natural water is chemically not pure, it contains a large number of foreign constituents like (I) minerals containing various metal ions, (ii) soluble organics like alcohols, acids and sugars, (iii) trace amounts of organics and inorganics which are normally insoluble, (iv) suspended particulates, (v) living organisms and products of their metabolism and (vi) floating matter such as Oil. All these come either through solution of aerosol particles, which act as the nuclei for condensation of raindrops, or through solution of atmospheric particles by the falling raindrops below the cloud level on impaction. Terrestrial aerosol particles are rich in Ca2+, NH+, NO, HCO and SO2 while aerosols derived from the sea surface are high in Na +, Cl-, Mg2+ and K+.

Water from surface runoff incorporates the soluble materials and entrains sediment particles, making the water turbid. The suspended sediments are likely to adsorb ions and other matter. Water infiltrating into the soil will have its quality modified appreciably due to soil-water interactions. During dry periods, evaporative losses will be maximum and the constituents of the sub-surface water will 27 be enriched. The exchange reactions and the chemical equilibria, involving soil and water, have been known to play an important role in determining water quality. Water also behaves as a chemical weathering agent and this aspect' is -f reflected in water quality through dissolved silica, Na, Ca2+. Mg2+ and HCO-2. The quality of water in contact with soil is also governed by complex biotic

regulation which is reflected in NH and NO. The quality of groundwater, on the other hand, depends on the type of rock entrapping the water and also on depth and movement of aquifers.

## 2.2 RIVER WATER

Surface runoff and groundwater flows together constitute the river water. The contributions to river water include:

(a) Precipitation

(b) Materials added on land erosion.

(c) Solution of chemicals during travel over and through the soil.

(d) Removal of chemicals by plants or by reaction with soil constituents.

(e) Addition through human activities, from both point and non-point sources,

## 2.3 HUMAN IMPACT ON RIVER WATER

Human impact on rivers is large-scale process that leads to diverse negative con-sequences.

There are following ways of such impact: 1) river flow redistribution in time; 2) river flow redistribution in space; 3) river flow withdrawal; 4) physical disturbance of river-beds; 5) pollution; 6) water clogging; 7) thermal pollution.

First way mainly occurs in case of reservoir creation, it is characteristic for the USA, Russia, Canada, Brazil, and China. Run-off redistribution in space used for water supply, navigation, hydropower generation, irrigation, etc. The most large-scale water transfers are typical for Canada, USA, Turkmenistan, and India. Irretrievable water consumption currently constitutes approx. 150 km 3 /year, which equals 1% of normal run-off of fresh water. Agriculture uses 70.1% of fresh water, industries take 20%, and municipal sector – 9.9%. Under physical disturbance of river-beds we mean any man-made changes of water level (cut-offs, changes in depth of the river by excavation or covering of ground, etc.). Open pits in river-beds for extraction of building materials and excavation works for navigation purposes are the most frequent examples of such impact. Water contamination is tremendous and ever-increasing challenge. By its origin, three chief water pollutant groups may be distinguished: 1) municipal waste; 2) industrial waste; 3) agricultural waste. Under water clogging we mean accumulation of foreign substances, mainly insoluble. Thermal pollution connects mainly with cooling water released from electricity generating stations. Further research of human impact on rivers is necessary in order to minimize negative con-sequences of such impact in the future.

## 2.4 INDIAN RIVERS

Abstract India is rich in water resources, being endowed with a network of rivers and blessed with snow cover in the Himalayan range that can meet a variety of water requirements of the country. However, with the rapid increase in the population of the country and the need to meet the increasing demands of irrigation, human and industrial consumption, the available water resources in many parts of the country are getting depleted and the water quality has deteriorated. Indian rivers are polluted due to the discharge of untreated sewage and industrial effluents. The Central Pollution Control Board (CPCB) has established a network of monitoring stations on rivers across the country. The present network is comprising of 870 stations in 26 States and 5 Union Territories spread over the country.

The monitoring is done on monthly or quarterly basis in surface waters and on half yearly basis in case of ground water. The monitoring network covers 189 Rivers, 53 Lakes, 4 Tanks, 2 Ponds, 3 Creeks, 3 Canals, 9 Drains and 218 Wells. Among the 870 stations, 567 are on rivers, 55 on lakes, 9 on drains,12 on canals, 4 on tank, 3 on and creeks,2 on pond and 218 are groundwater stations. Presently the inland water quality monitoring network is operated under a three-tier programme i.e. GEMS, Monitoring of Indian National Aquatic Resources System and Yamuna Action Plan. Water samples are being analysed for 28 parameters consisting of physio-chemical and bacteriological parameters for ambient water samples apart from the field observations. Biomonitoring is also carried out on specific locations. In view of limited resources, limited numbers of organic pollution related parameters are chosen for frequent monitoring i.e. monthly or quarterly and major cations, anions, other inorganic ions and micro pollutants (Toxic Metals & POP's) are analysed once in a year to keep a track of water quality over large period of time. The water quality monitoring of major rivers indicates that organic pollution is predominant and almost all the surface water sources are contaminated to some extent by Coliform Group of Bacteria that make them unfit for human consumption unless disinfected. The grossly polluted rivers on specific stretches are Sabarmati, Godavari, Satluj, Yamuna, Cauvery, Ganga, Krishna, Tapi, Mahanadi and Brahmani whereas relatively clean rivers are Mahi, Narmada, Brahmaputra and Beas with respect to organic and bacterial pollution.

### 2.4.1 WATER RESOURCES OF INDIA

The geographical area of India is 3,287,590 sq-km. The length of its Coastline is about 7500 km. The climate of India varies from tropical monsoon in south to temperate in north. Its terrain has upland plain (Deccan Plateau) in south, flat to rolling plain along the Ganges, deserts in west, Himalayas in north. India is enviably endowed in respect of water resources. The country is literally criss-crossed with rivers and blessed with high precipitation mainly due to the southwest monsoon, which accounts for 75% of the annual rainfall. There are thirteen major river basins (area more than 20,000 square kilometre) in the country, which occupy 82.4% of total drainage basins, contribute eighty five percent of total surface flow and house eighty percent of the country's population. Major river basins are Brahmaputra, Ganga (including Yamuna Sub Basin), Indus (including Satluj and Beas Sub Basin), Godavari, Krishna, Mahanadi, Narmada, Cauvery, Brahmini (including Baitarni Sub Basin), Tapi, Mahi, Pennar and Sabarmati. The classification of river basin based on catchment area is given in Table 1. There are few desert rivers, which flow for some distance and get lost in deserts. There are complete arid areas where evaporation equals rainfall and hence no surface-flow. The medium and minor river basins are mainly in coastal area. On the east coast and part of Kerala State, the width of land between mountain and sea is about 100 km, and hence the riverine length is also about 100 km. whereas, the rivers in the west coast are much shorter as the width of the land between sea and mountains is less than 10 to 40 km. Yet, in spite of the nature's bounty, paucity of water is an issue of national concern resulting in deterioration of water quality in aquatic resources.

# CHAPTER: 3

## DEFINITION OF THE PROBLEM WITH MODULES AND FUNCTIONALITIES

## 3.1. DEFINITION OF THE PROBLEM:

Water Quality prediction using Machine Learning throughout a range of years.

- Indian Water Quality Prediction using ML (Government Data related and will be regulated by government so it is a kind of E-Governance Project)
- The objective here is to analyse this data alongside with Forest, Industries, Habitation, and development projects data in the same area (panchayat) to figure out whether there is any connection between the development effort and the quality of water getting affected. This effort will identify such associations and create awareness such that people and the Govt. can act in time to avoid further deterioration of the water resource. trained sets will be previous years data test sets will be current year data.
- Prediction of a region where there will be water some quality issues in areas that are not affected right now based on the development and water quality data that is available right now. Moral: prevention is better than cure.

Basically, Water Quality Parameters are co-related to each other. Thus, any kind of abnormal change in the molecular or sub-atomic or percentage of Water Ingredients may have adverse effects to the Balanced Eco System of Water.

To solve This year, we have studied and analysed some datasets and invoked the Dataset into our Machine Learning Model.

**THE PARAMETERS OF THE DATASETS:**

a) States of India
b) Year
c) pH
d) B.O.D (mg/l)
e) D.O (mg/l)
f) Conductivity (Electrical Conductivity (E.C)) (S/m)
g) NitrateNan+ (mg/l)
h) Quantity of Fecal Cloroform

**PH OF WATER:**

pH means the potential of Hydrogen. In chemistry, pH is a scale used to specify how acidic or basic a water-based solution is. Acidic solutions have a lower pH, while basic solutions have a higher pH. At room temperature, pure water is neither acidic nor basic and has a pH of 7.It has no unit system.

**B.O.D:**

Biochemical oxygen demand (BOD), the amount of dissolved oxygen used by microorganisms in the biological process of metabolizing organic matter in water. The more organic matter there is (e.g., in sewage and polluted bodies of water), the greater the BOD; and the greater the BOD, the lower the amount of dissolved oxygen available for higher animals such as fishes. The BOD is therefore a reliable gauge of the organic pollution of a body of water.

**D.O:**

Dissolved oxygen (DO) is a measure of how much oxygen is dissolved in the water - the amount of oxygen available to living aquatic organisms. The amount of dissolved oxygen in a stream or lake can tell us a lot about its water quality.

**ELECTRICAL CONDUCTIVITY (E.C):**

Electrical Conductivity of water is its ability to conduct an electric current. Salts or other chemicals that dissolve in water can break down into positively and negatively charged ions.Salinity and total dissolved solids (TDS) are used to calculate the EC of water, which helps to indicate the water's purity. It is measured in Siemens/Meter (S/m).

**NITRATENAN:**

Basically, any excess nitrate in the water is a source of fertilizer for aquatic plants and algae. In many cases, the amount of nitrate in the water is what limits how much plants and algae can grow. If there is an excess level of nitrates, plants and algae will grow excessively.

**FECAL CLOROFORM:**

The presence of faecal coliform bacteria in aquatic environments indicates that the water has been contaminated with the faecal material of man or other animals. ... Significance: Faecal Coliform bacteria indicate the presence of sewage contamination of a waterway and the possible presence of other pathogenic organisms.

Now based on the attributes Our Machine Learning Will Model will predict The Water Quality Index which indicate whether the water of specified region is affected or degraded or not. If Degradation occurs, it will tell the percentage of degradation of Water in that specified region or circle. Based on that report the process of Purification of water is to be water.

Though there is no limitation in the year. But Still we are limiting the Year Attribute from 2007 to

2030.

## 3.2. MODULES AND FUNCTIONALITIES:

- A Web Application is created.
- It will contain A page of several pages which will be linked to Our Trained Machine Learning Model.
- All the Values (Inputs) will be provided to that and as soon as you click on PREDICT Button, it will show you the Water Quality Index (WQI).
- It has a Graph-Slider which will showcase the Graphs and Co-relation between attributes and gives a good visualization.
- It runs on local server using Anaconda Prompt in its virtual environment.
- To make our web-Application device-independent, we used Amazon EC2 (Elastic Compute Cloud) provided by Amazon Web Services to create an EC2 Instance (Virtual Server) and run our Web-Application in that virtual server. So that anyone can access our website from anywhere and from any devices (PC, Smartphone).

# CHAPTER: 4

# SOFTWARE DESIGN

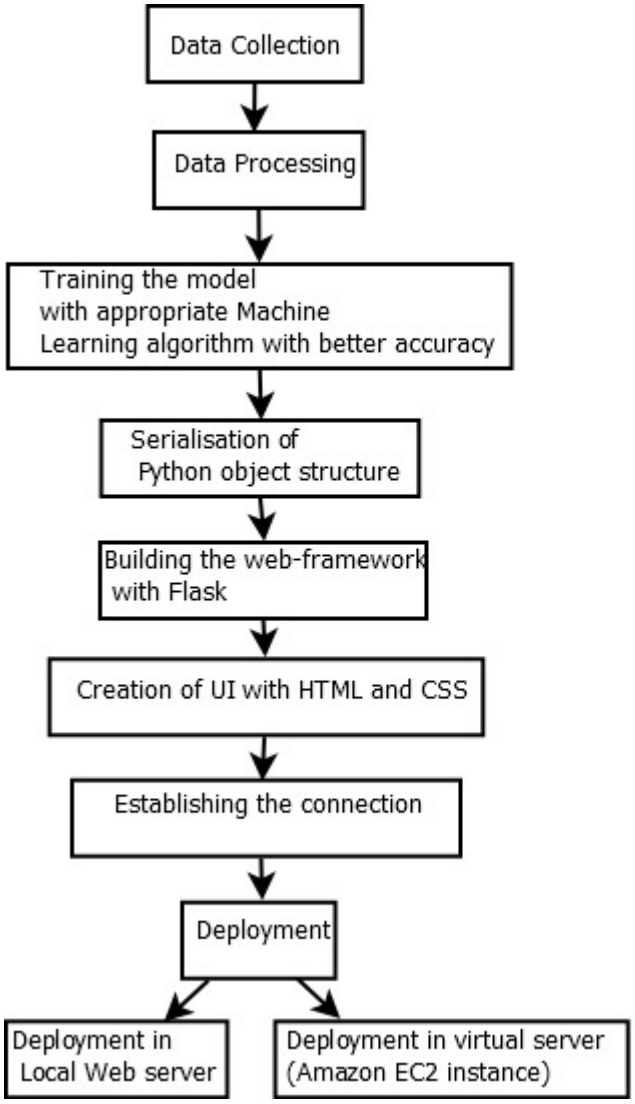
# SOFTWARE DESIGN

## 4.1. METHODOLOGY:

To achieve our objective of the project, we will implement the model through Machine Learning which is part of AI. To do this Python will be used as backend which will supply all the necessary modules required to analyse and predict the water quality of particular state and a micro-web framework will be needed to route the pages to their appropriate destination. There are also some algorithms in python which will make the prediction much better. It will give the results based on the trained Machine Learning Model. Methodologies adopted to get predictions for water quality has been discussed in this section.

Figure: -1

**FLOW-DIAGRAM:**



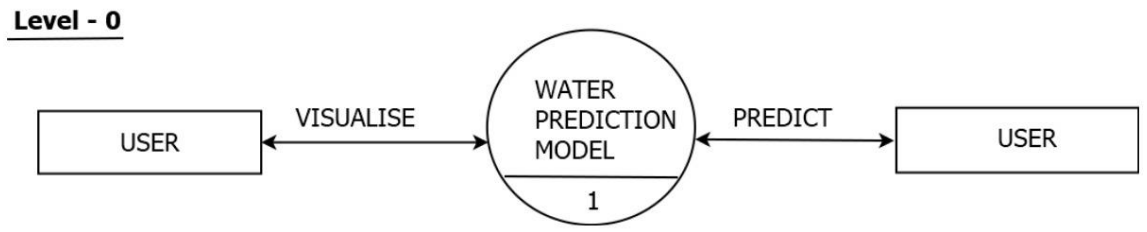
**THE STEPS ARE:**

- **Data Collection:** Data collection is the process of collecting data and discovering the patterns and behaviour of the data. The data includes the parameters like temperature, pH, dissolved oxygen, B.O.D, Electrical Conductivity, Quantity of Nitrate etc.
- **Data Pre-processing:** Data pre-processing helps to improve the quality and efficiency of the data. As raw data is inconsistent and noisy which affects the quality of the data. Data pre-processing is a step which deals with data preparation and transformation.
- **Training the Model:** This includes the process of developing a predictive model, training, testing and validating it. Machine Learning based techniques were used in order to develop the model for prediction. Supervised Machine Learning Algorithms were used as we had data marked in labels.
- **Serialisation of a Python object structure:** Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.
- **Building the Web-Framework with Flask:** Establishing Connection between the UI and the Model is done with the help of Flask which is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.
- **Creation of UI:** A convenient User Interface is created using HTML and CSS.
- **Connection**: A micro web-framework Flask is used to connect with the UI and the serialised python module to maintain the transport of data between them.
- **Deployment: Running it on Local Server:** To do this one must have anaconda prompt installed on his computer. As soon as we launch the server from that. Our website would be going well.
- **Deployment: Running it on Virtual Server:** We also made it possible that anyone can use the website from any website without any hassle. Amazon EC2 Instance is used to do this.

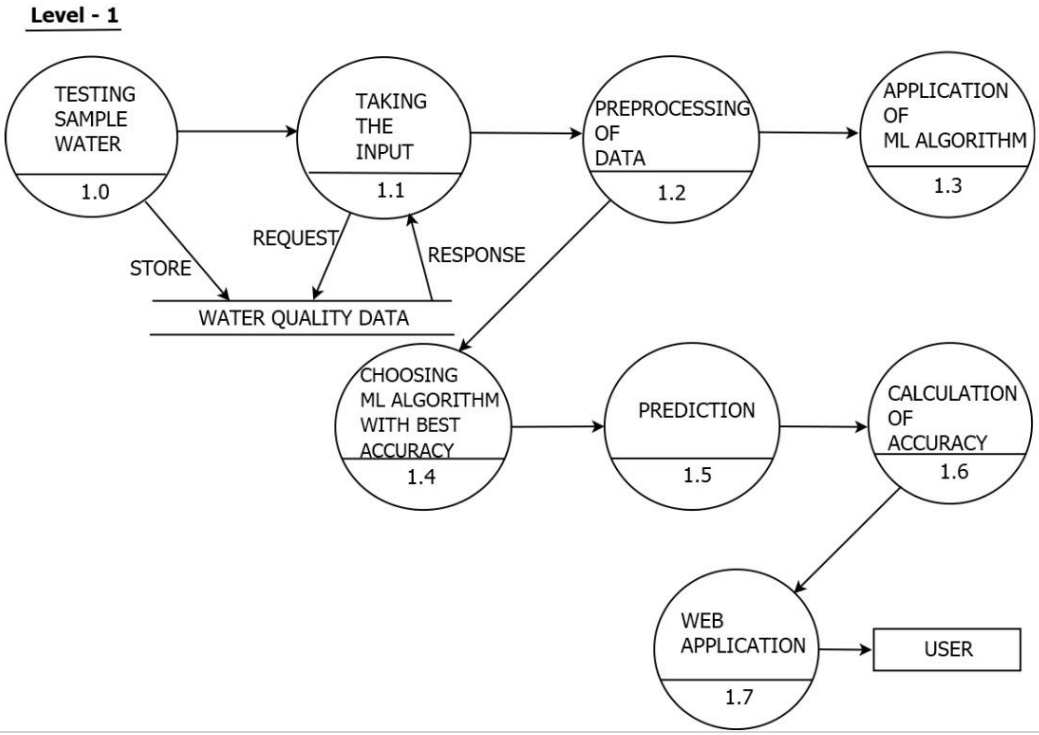
To further illustrate our point of view we have demonstrated it through the software design techniques which includes DFDs, UML and E-R diagram.

## 4.2.DFD:

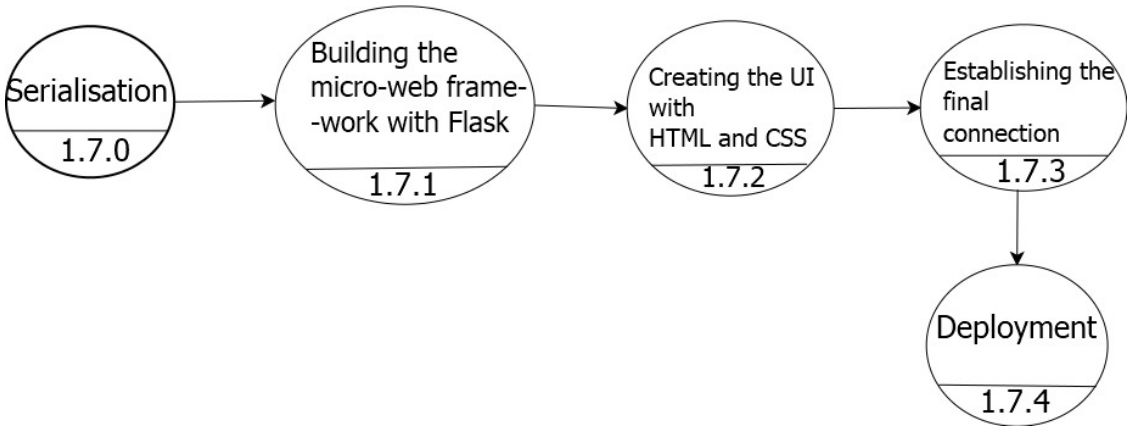
**Level-0 (figure-2.1)**



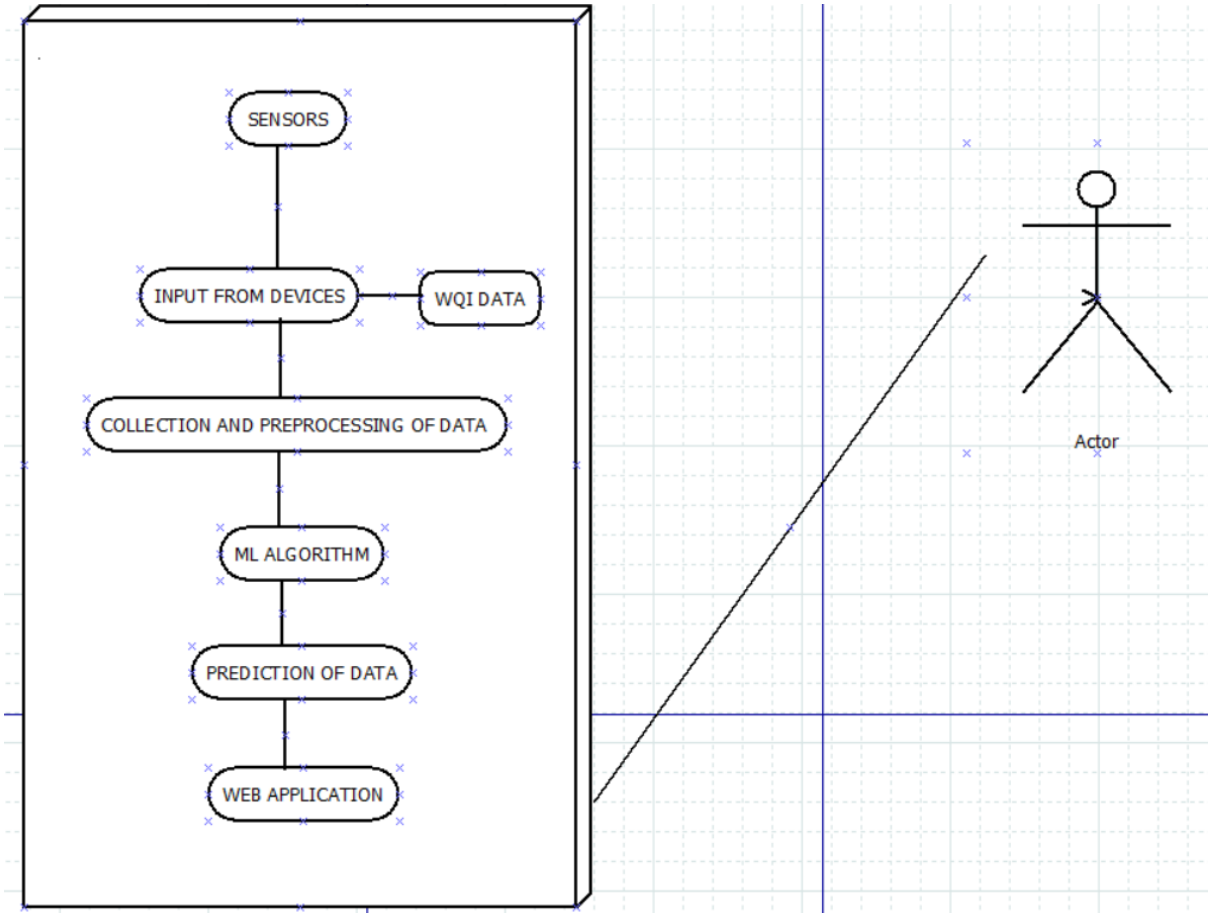
**Level-1(figure-2.2)**




**Level-2(figure-2.3)**



## 4.3. UML: Use Case:




(Figure-3)

# CHAPTER:5

# SOFTWARE AND HARDWARE REQUIREMENTS

# SOFTWARE AND HARDWARE REQUIREMENTS

## SOFTWARE USED:

- Visual Studio Code
- Anaconda Navigator and Anaconda Prompt
- Spyder
- Jupyter Notebook
- WinSCP
- Putty
- PuttyGen
- Python 3.8
- Amazon Web Services (AWS) EC2 INSTANCE

## LANGUAGES USED:

- Python
- HTML
- CSS

## TOOLS USED IN PYTHON:

- Flask
- Numpy
- Pandas
- Matplotlib
- Seaborn
- Sklearn → Sklearn.ensemble →RandomForestRegressor

## TOOLS USED UNDER AWS EC2 INSTANCE:

- SECURITY GROUPS
- NETWORK INTERFACE

## HARDWARE REQUIREMENTS:

- Intel I5 processor (Recommended)

# CHAPTER: 6

# CODE TEMPLATES

# CODE TEMPLATES

**The Code Template is divided into 5 parts.**

- Creation of The Model Using Python
- Creation of Micro-web Framework using Flask
- Creation of UI
- Launching it in the Local Web Server
- Launching it into AWS EC2 Instance

## 6.1. CREATION OF THE MODEL USING PYTHON

**1.Loading the Dataset:** We have collected the dataset from Kaggle and few government websites and merged the dataset to form a fully qualified dataset which will ease the pre-processing & prediction process and uploaded the dataset through pandas. (Table-1)

In this project since State is a String Attribute, Error occurred after the compilation of the Model, hence the State Attribute is converted from its String representation to Integer representation. Every value is mapped to its corresponding Integer No. The Mapping is shown below.

```
GOA=1
MAHARASHTRA=2
KERALA=3
ANDHRA PRADESH=4
KARNATAKA=5
ODISHA=6
PONDICHERRY=7
TAMILNADU=8
HARYANA=9
PUNJAB=10
RAJASTHAN=11
HIMACHAL PRADESH=12
MEGHALAYA=13
MIZORAM=14
TRIPURA=15
DAMAN & DIU=16
GUJARAT=17
ORISSA=18
MADHYA PRADESH=19
MANIPUR=20
```

```
In [1]: import numpy as np
        import pandas as pd
```

```
In [2]: data=pd.read_csv("water_dataX.csv",encoding="ISO-8859-1")
```

Table-1

| | STATION CODE | LOCATIONS | STATE | Temp | D.O. (mg/l) | PH | CONDUCTIVITY (µmhos/cm) | B.O.D. (mg/l) | NITRATENAN N+ NITRITENANN (mg/l) | FECAL COLIFORM (MPN/100ml) | TOTAL COLIFORM (MPN/100ml)Mean | year |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1393 | DAMANGANGA AT D/S OF MADHUBAN, DAMAN | DAMAN & DIU | 30.6 | 6.7 | 7.5 | 203 | NAN | 0.1 | 11 | 27 | 2014 |
| **1** | 1399 | ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI... | GOA | 29.8 | 5.7 | 7.2 | 189 | 2 | 0.2 | 4953 | 8391 | 2014 |
| **2** | 1475 | ZUARI AT PANCHAWADI | GOA | 29.5 | 6.3 | 6.9 | 179 | 1.7 | 0.1 | 3243 | 5330 | 2014 |
| **3** | 3181 | RIVER ZUARI AT BORIM BRIDGE | GOA | 29.7 | 5.8 | 6.9 | 64 | 3.8 | 0.5 | 5382 | 8443 | 2014 |
| **4** | 3182 | RIVER ZUARI AT MARCAIM JETTY | GOA | 29.5 | 5.8 | 7.3 | 83 | 1.9 | 0.4 | 3428 | 5500 | 2014 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1986** | 1330 | TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU | NAN | NAN | 7.9 | 738 | 7.2 | 2.7 | 0.518 | 0.518 | 202 | 2003 |
| **1987** | 1450 | PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, T... | NAN | 29 | 7.5 | 585 | 6.3 | 2.6 | 0.155 | 0.155 | 315 | 2003 |
| **1988** | 1403 | GUMTI AT U/S SOUTH TRIPURA,TRIPURA | NAN | 28 | 7.6 | 98 | 6.2 | 1.2 | NAN | NAN | 570 | 2003 |
| **1989** | 1404 | GUMTI AT D/S SOUTH TRIPURA, TRIPURA | NAN | 28 | 7.7 | 91 | 6.5 | 1.3 | NAN | NAN | 562 | 2003 |
| **1990** | 1726 | CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA | NAN | 29 | 7.6 | 110 | 5.7 | 1.1 | NAN | NAN | 546 | 2003 |

1991 rows × 12 columns

**2. Pre-Processing of Data:** Some basic pre-processing is done so that the model does not drop any kind of runtime errors such as NaN values or the presence of string along with integer values. In essence we have converted the data items from object format to number (int, float) format. (Table-2) Here Replacing NaN Values with Zero.

```
In [4]: data.fillna(0,inplace=True) #it fills NaN with 0's
```

Data-Types Before Pre-Processing (Figure-4)

```
In [6]: data.dtypes

Out[6]: STATION CODE                          int64
        LOCATIONS                            object
        STATE                                 int64
        Temp                                 object
        D.O. (mg/l)                          object
        PH                                   object
        CONDUCTIVITY (µmhos/cm)              object
        B.O.D. (mg/l)                        object
        NITRATENAN N+ NITRITENANN (mg/l)     object
        FECAL COLIFORM (MPN/100ml)           object
        TOTAL COLIFORM (MPN/100ml)Mean       object
        year                                  int64
        dtype: object
```

Data-Types After Pre-Processing (Figure-5)

```
In [7]: #conversion
        data['Temp']=pd.to_numeric(data['Temp'],errors='coerce')
        data['D.O. (mg/l)']=pd.to_numeric(data['D.O. (mg/l)'],errors='coerce')
        data['PH']=pd.to_numeric(data['PH'],errors='coerce')
        data['B.O.D. (mg/l)']=pd.to_numeric(data['B.O.D. (mg/l)'],errors='coerce')
        data['CONDUCTIVITY (µmhos/cm)']=pd.to_numeric(data['CONDUCTIVITY (µmhos/cm)'],errors='coerce')
        data['NITRATENAN N+ NITRITENANN (mg/l)']=pd.to_numeric(data['NITRATENAN N+ NITRITENANN (mg/l)'],errors='coerce')
        data['TOTAL COLIFORM (MPN/100ml)Mean']=pd.to_numeric(data['TOTAL COLIFORM (MPN/100ml)Mean'],errors='coerce')
        data.dtypes

Out[7]: STATION CODE                          int64
        LOCATIONS                            object
        STATE                                 int64
        Temp                                float64
        D.O. (mg/l)                         float64
        PH                                  float64
        CONDUCTIVITY (µmhos/cm)             float64
        B.O.D. (mg/l)                       float64
        NITRATENAN N+ NITRITENANN (mg/l)    float64
        FECAL COLIFORM (MPN/100ml)           object
        TOTAL COLIFORM (MPN/100ml)Mean      float64
        year                                  int64
        dtype: object
```

** errors="coerce" to try-catch the error so that it does not hamper normal execution of the program.

**3. Standardization of Data:** At the 3rd stage standardizing is done. It is meant that every attribute is measured in terms of 100 to find the WQI (Water Quality Index) more comfortably. Figure-6

```
In [11]: data=pd.concat([station,location,state,do,ph,co,bod,na,tc,year],axis=1)
         data. columns = ['station','location','state','do','ph','co','bod','na','tc','year']
```

```
In [12]: #calulation of Ph
         data['npH']=data.ph.apply(lambda x: (100 if (8.5>=x>=7)
                                        else(80 if  (8.6>=x>=8.5) or (6.9>=x>=6.8)
                                             else(60 if (8.8>=x>=8.6) or (6.8>=x>=6.7)
                                                 else(40 if (9>=x>=8.8) or (6.7>=x>=6.5)
                                                     else 0)))))
```

```
In [13]: #calculation of dissolved oxygen
         data['ndo']=data.do.apply(lambda x:(100 if (x>=6)
                                        else(80 if  (6>=x>=5.1)
                                             else(60 if (5>=x>=4.1)
                                                 else(40 if (4>=x>=3)
                                                     else 0)))))
```

```
In [14]: #calculation of total coliform
         data['nco']=data.tc.apply(lambda x:(100 if (5>=x>=0)
                                        else(80 if  (50>=x>=5)
                                             else(60 if (500>=x>=50)
                                                 else(40 if (10000>=x>=500)
                                                     else 0)))))
```

```
In [15]: #calc of B.D.O
         data['nbdo']=data.bod.apply(lambda x:(100 if (3>=x>=0)
                                        else(80 if  (6>=x>=3)
                                             else(60 if (80>=x>=6)
                                                 else(40 if (125>=x>=80)
                                                     else 0)))))
```

```
In [16]: #calculation of electrical conductivity
         data['nec']=data.co.apply(lambda x:(100 if (75>=x>=0)
                                        else(80 if  (150>=x>=75)
                                             else(60 if (225>=x>=150)
                                                 else(40 if (300>=x>=225)
                                                     else 0)))))
```

```
In [17]: #Calulation of nitrate
         data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)
                                        else(80 if  (50>=x>=20)
                                             else(60 if (100>=x>=50)
                                                 else(40 if (200>=x>=100)
                                                     else 0)))))

         data.head()
         data.dtypes
```

```
In [16]: #Calulation of nitrate
         data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)
                                        else(80 if  (50>=x>=20)
                                             else(60 if (100>=x>=50)
                                                 else(40 if (200>=x>=100)
                                                     else 0)))))

         data.head()
         data.dtypes
```

Output: Table-2

```
In [17]: data
```

Out[17]:

| | station | location | state | do | ph | co | bod | na | tc | year | npH | ndo | nco | nec | nbdo | nna |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2** | 1475 | ZUARI AT PANCHAWADI | 1 | 6.3 | 6.9 | 179.0 | 1.7 | 0.10 | 5330.0 | 2014 | 80 | 100 | 40 | 60 | 100 | 100 |
| 3 | 3181 | RIVER ZUARI AT BORIM BRIDGE | 1 | 5.8 | 6.9 | 64.0 | 3.8 | 0.50 | 8443.0 | 2014 | 80 | 80 | 40 | 100 | 80 | 100 |
| **4** | 3182 | RIVER ZUARI AT MARCAIM JETTY | 1 | 5.8 | 7.3 | 83.0 | 1.9 | 0.40 | 5500.0 | 2014 | 100 | 80 | 40 | 80 | 100 | 100 |
| 5 | 1400 | MANDOVI AT NEGHBOURHOOD OF PANAJI, 1 | 1 | 5.5 | 7.4 | 81.0 | 1.5 | 0.10 | 4049.0 | 2014 | 100 | 80 | 40 | 80 | 100 | 100 |
| 6 | 1476 | MANDOVI AT TONCA, MARCELA, 1 | 1 | 6.1 | 6.7 | 308.0 | 1.4 | 0.30 | 5672.0 | 2014 | 60 | 100 | 40 | 0 | 100 | 100 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1037** | 1862 | RIVER KAVERI ON BRIDGE AT BILLIMORANANVALSAD ROAD | 17 | 6.8 | 8.2 | 26600.0 | 3.1 | 0.62 | 23000.0 | 2010 | 100 | 100 | 0 | 0 | 80 | 100 |
| **1038** | 1865 | RIVER DHADAR AT KOTHADA | 17 | 8.1 | 7.6 | 938.0 | 8.6 | 0.40 | 108.0 | 2010 | 100 | 100 | 60 | 0 | 60 | 100 |
| **1039** | 1148 | AMBIKA AT BILIMORA, | 17 | 6.2 | 7.8 | 7778.0 | 2.7 | 0.26 | 6581.0 | 2010 | 100 | 100 | 40 | 0 | 100 | 100 |
| **1040** | 1434 | AMLAKHADI AFTER CONFL. OF W. WATER FROM ANKLESH, | 17 | NaN | 7.2 | 1440.0 | 46.0 | 0.20 | 330.0 | 2010 | 100 | 0 | 60 | 0 | 60 | 100 |
| **1041** | 1438 | MINDHOLA AT STATE HIGHWAY BRIDGE SACHIN, | 17 | 3.2 | 7.7 | 885.0 | 6.3 | 0.51 | 9575.0 | 2010 | 100 | 40 | 40 | 0 | 60 | 100 |

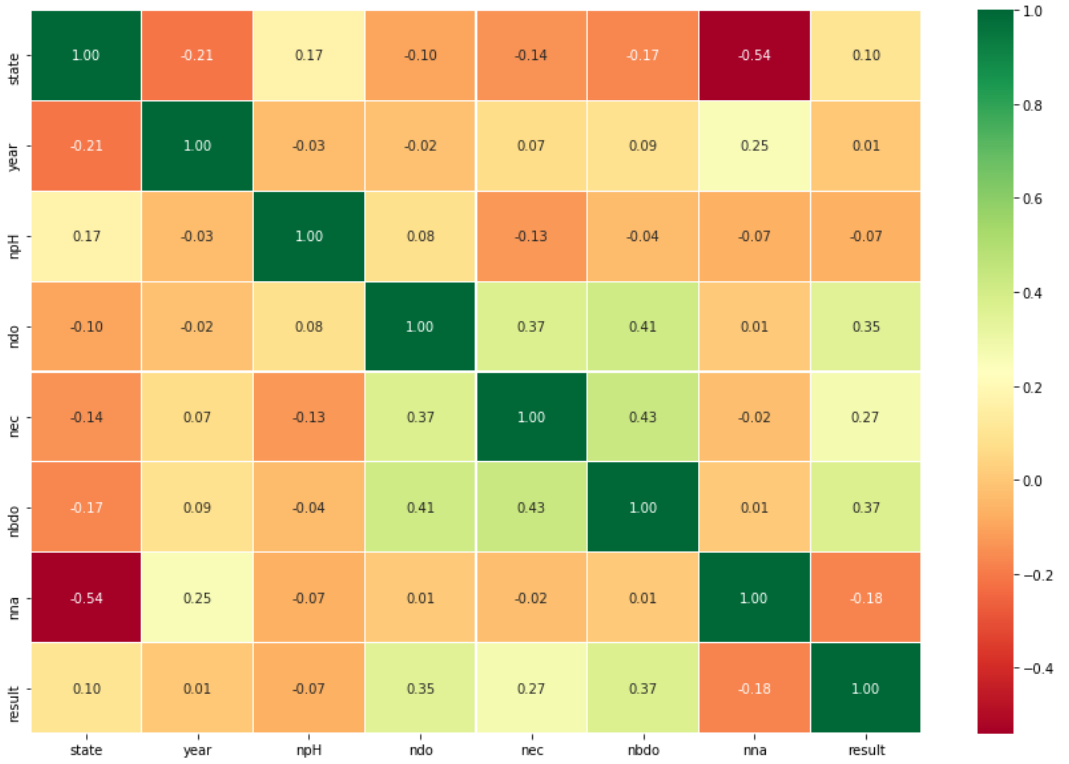
1040 rows × 16 columns

## 4.Visualization:

- **Heatmap**: A Heat-map is a data visualization technique that shows magnitude of a phenomenon as colour in two dimensions. The variation in colour may be by hue or intensity, giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space.

Here It shows the correlation between the attributes. (Figure-7)

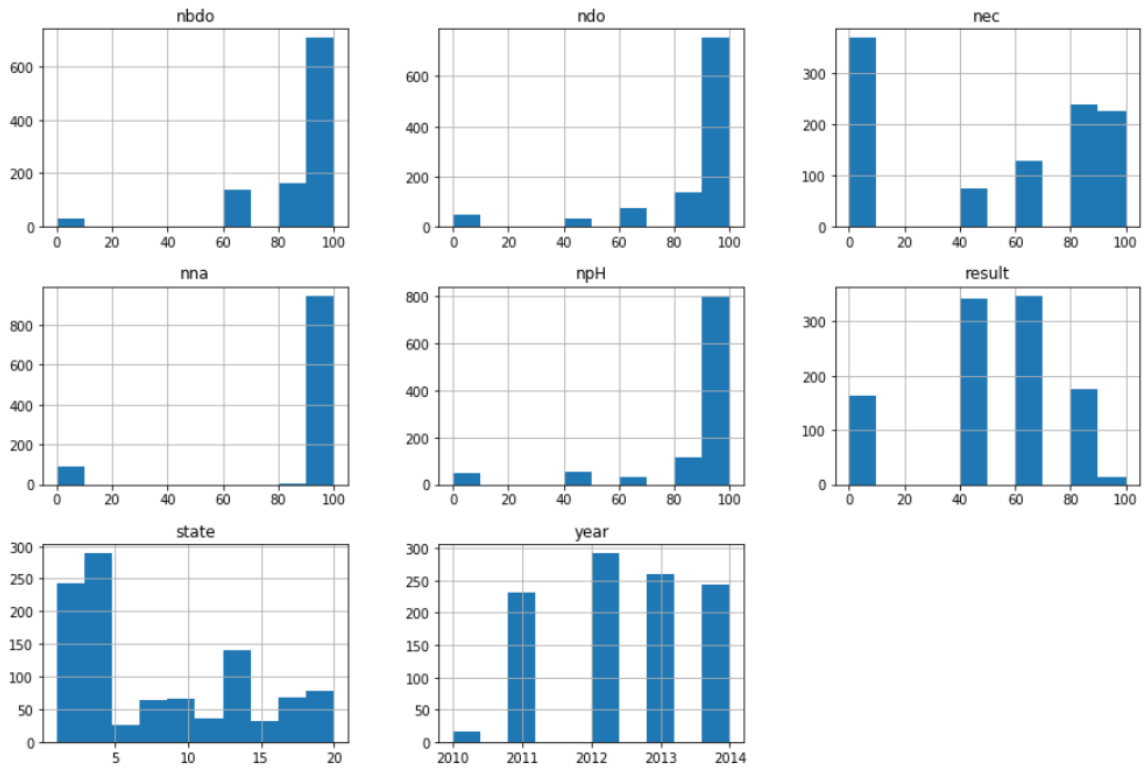
```
In [49]: #co-relation
data['result']=yt
import matplotlib.pyplot as plt
%matplotlib inline
fig,ax=plt.subplots(figsize=(15,10))
sns.heatmap(data.corr(),ax=ax,annot=True,linewidths=0.05,fmt='.2f',cmap="RdYlGn")
plt.show()
```



- **Histogram:** A Histogram is a type of graph that has wide applications in statistics. Histograms provide a visual interpretation of numerical data by indicating the number of data points that lie within a range of values. ... The higher that the bar is, the greater the frequency of data values in that bin.

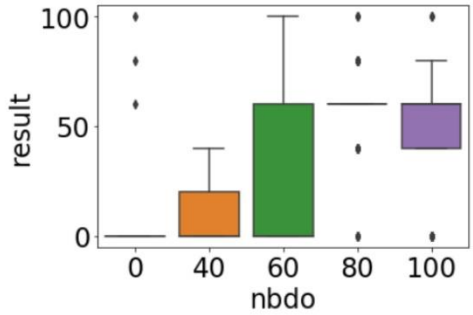
```
In [50]: figg=plt.figure(figsize=(15,10))
         ax=figg.gca()
         data.hist(ax=ax)

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarning: To output multiple subplots, the figure containin
g the passed axes is being cleared
  This is separate from the ipykernel package so we can avoid doing imports until

Out[50]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EBEBEF88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EBFF0AC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC029A88>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC062B88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC096748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC0D0888>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC1099C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC141AC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D9EC14C6C8>]],
      dtype=object)
```

Output: (Figure-8)

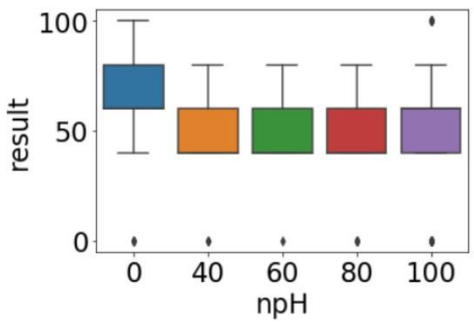


**Boxplot:** A Boxplot is a standardized way of displaying the distribution of data based on a five-number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It can tell you about your outliers and what their values are. (Figure-9)

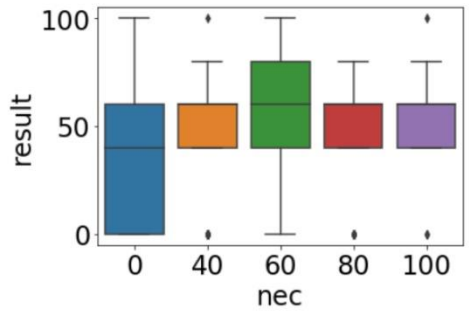
```
In [62]: sns.boxplot(x="nbdo",y="result",data=data)
Out[62]: <matplotlib.axes._subplots.AxesSubplot at 0x1d9edc05a08>
```



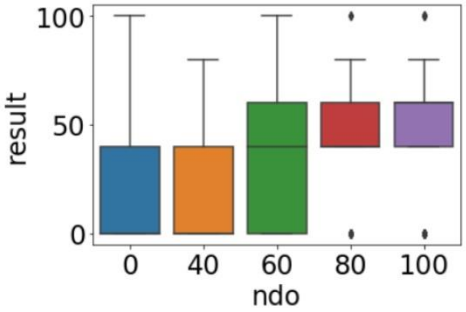

```
In [63]: sns.boxplot(x="npH",y="result",data=data)
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x1d9edc97c08>
```




```
In [64]: sns.boxplot(x="nec",y="result",data=data)
Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x1d9edd4ee88>
```

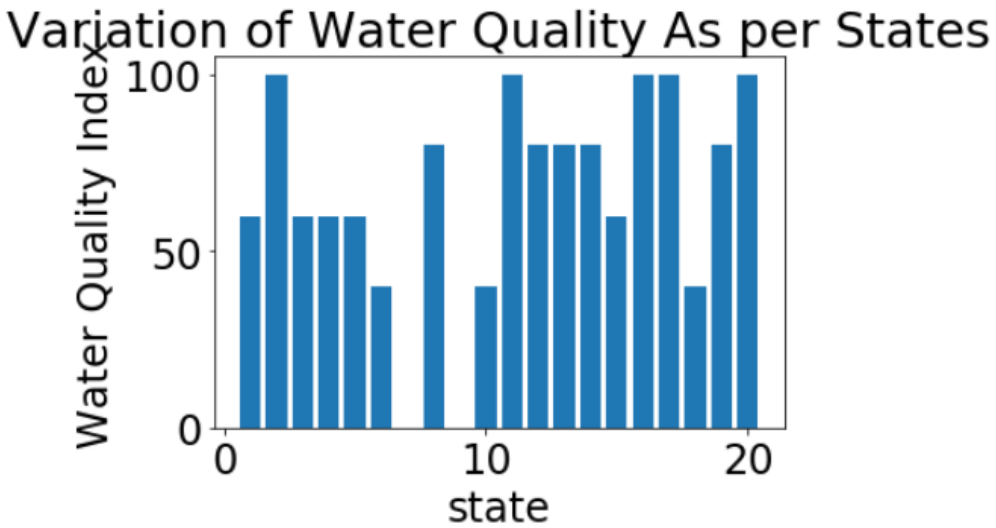



```
In [65]: sns.boxplot(x="ndo",y="result",data=data)
Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x1d9ede1ac88>
```




- **Bar Chart:** A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. (fig.10)

```
In [68]: st=data['state']
         res=data['result']
         plt.bar(st,res)
         plt.xlabel("state")
         plt.ylabel("Water Quality Index")
         plt.title("Variation of Water Quality As per States")
         plt.show()
```



**5.Creating the Model:** We used two Machine Learning Algorithm A) Linear Regression B) Random Forest Regression.

**Linear Regression:** Linear regression is a statistical approach for modelling relationship between a dependent variable with a given set of independent variables. It tries to find the best fit line and give the result.

**Random Forest Regression:** Random forest is a supervised learning algorithm which is used for both classification as well as regression. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. A random forest, in contrast, is an ensemble bagging or averaging method that aims to reduce the variance of individual trees by randomly selecting (and thus de-correlating) many trees from the dataset, and averaging them.

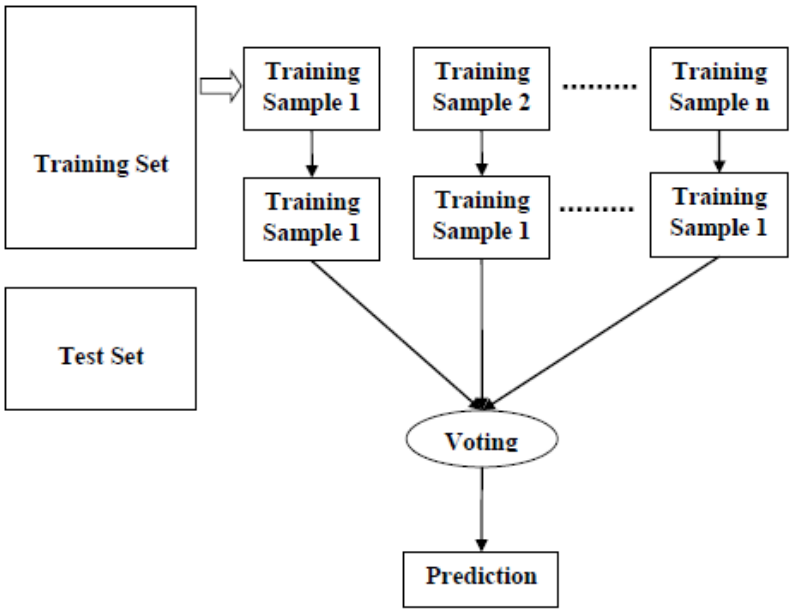
Now Among these two, Random Forest Regression Stand Much better with an accuracy rate of 0.71 while Linear Regression Model shows only 38.2. It happened because the correspondence between attributes is non-linear. Linear Regression failed due to being unable to achieve the best fit line.

Hence, We preferred using Random Forest Regression Algorithm.

**Random Forest Regression: Methodology:**

- Step1: First, start with the selection of random samples from a given dataset.
- Step2: Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- Step 3 − In this step, voting will be performed for every predicted result.
- Step 4 − At last, select the most voted prediction result as the final prediction result.

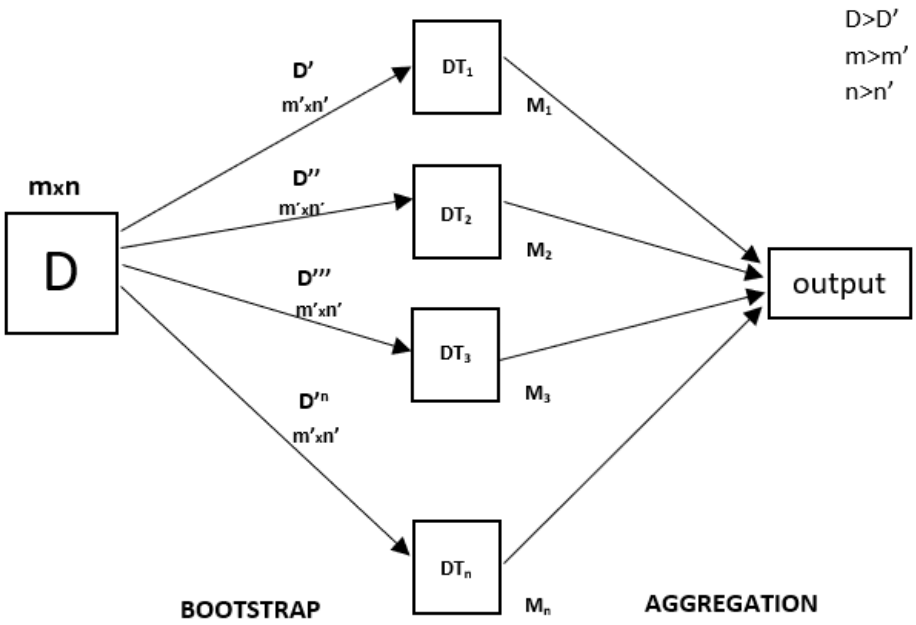
The following diagram will illustrate its working – (figure-11)




**RANDOM FOREST REGRESSION:**

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and

hence the output doesn't depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation. (fig.12)




A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Codes and Implementation of RandomForestRegressor: 1000 Decision Trees are created with Randomness.

```
In [35]: #randomforest

In [36]: from sklearn.ensemble import RandomForestRegressor

In [37]: rfr=RandomForestRegressor(n_estimators=1000,random_state=42)

In [38]: rfr.fit(x_t,y_t)

Out[38]: RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                               max_depth=None, max_features='auto', max_leaf_nodes=None,
                               max_samples=None, min_impurity_decrease=0.0,
                               min_impurity_split=None, min_samples_leaf=1,
                               min_samples_split=2, min_weight_fraction_leaf=0.0,
                               n_estimators=1000, n_jobs=None, oob_score=False,
                               random_state=42, verbose=0, warm_start=False)

In [39]: yrfr=rfr.predict(x_tt)

In [40]: from sklearn.metrics import mean_squared_error
         print('mse:%.2f'%mean_squared_error(y_tt,yrfr))

         mse:180.78

In [41]: y_tt

Out[41]: 800     0
         814    40
         622    40
         676    60
         934    40
                ..
         663     0
         627    40
         477    80
         739    80
         410    40
         Name: nco, Length: 208, dtype: int64

In [42]: yrfr

Out[42]: array([1.08083810e+01, 7.33556429e+01, 4.39350515e+01, 4.81200000e+01,
                5.07751667e+01, 4.46349048e+01, 5.53687705e+01, 4.08560000e+01,
                4.34650000e+01, 3.98646667e+01, 6.10709524e+01, 4.42294349e+01,
                6.00000000e+01, 2.00103377e+01, 5.02538818e+01, 5.88350000e+01,
                8.48966275e+01, 4.39350515e+01, 7.74526844e+01, 5.13976329e+01,
                4.01183333e+01, 6.00000000e+01, 4.31572381e+01, 7.40329918e+01,
                6.70338571e+01, 4.90063571e+01, 3.04880952e+00, 4.66536699e+01,
                7.17728424e+01, 4.00378571e+01, 5.51124252e+01, 2.92187619e+01,
                3.70293333e+01, 4.39915687e+01, 4.39350515e+01, 4.66536699e+01,
                6.34400000e+01, 5.99333333e+01, 4.48861895e+01, 4.54790750e+01,
                6.31570117e+01, 5.53687705e+01, 5.33572922e+01, 6.96693674e+01,
                4.42294349e+01, 4.51021980e+01, 4.25893810e+01, 6.00000000e+01,
                5.33572922e+01, 2.75000000e-01, 4.20918571e+01, 7.33277967e+01,
                1.60000000e-01, 5.02716667e+01, 3.83516972e+01, 4.39350515e+01,
                6.00000000e+01, 5.27918968e+01, 5.98800000e+01, 4.09670476e+01,
                7.33277967e+01, 6.00000000e+01, 6.33043026e+01, 9.44223283e+01,
                4.04845556e+01, 5.33572922e+01, 4.78001905e+01, 5.99533333e+01,
                5.47744444e+01, 4.26666667e-02, 5.48200000e+01, 5.33572922e+01,
                3.98446667e+01, 5.54990476e+00, 5.99533333e+01, 6.31570117e+01,
                3.71777120e+01, 5.69055615e+01, 5.99533333e+01, 0.00000000e+00,
                3.42004762e+01, 5.33572922e+01, 2.00103377e+01, 4.80000000e-01,
                6.00667619e+01, 7.22825653e+01, 2.91750000e+01, 1.89856825e+01,
                4.66536699e+01, 1.80171905e+01, 7.03638732e+01, 6.00000000e+01,
                5.88581905e+01, 4.05072381e+01, 6.00000000e+01, 5.73705714e+01,
                7.57671204e+01, 1.43600000e+01, 7.26381905e+01, 5.92670317e+01,
                9.44223283e+01. 5.98511111e+01. 3.23400000e+01. 5.05974453e+01.
```

## 6.Validation:

**Below the Actual and Predicted Output is shown w.r.t the results of previous years. (Figure-13)

```
In [43]: dtrfr = pd.DataFrame({'Actual': y_tt, 'Predicted': a2}

In [44]: dtrfr.head(20)
```

Out[44]:

| | Actual | Predicted |
|---|---|---|
| **800** | 0 | 27.290000 |
| **814** | 40 | 38.771779 |
| **622** | 40 | 51.013495 |
| **676** | 60 | 50.175768 |
| **934** | 40 | 50.413583 |
| **126** | 60 | 58.725531 |
| **356** | 60 | 53.927114 |
| **795** | 40 | 51.011639 |
| **342** | 40 | 26.712041 |
| **287** | 40 | 56.503799 |
| **422** | 60 | 58.817262 |
| **359** | 40 | 52.042317 |
| **306** | 60 | 45.050543 |
| **694** | 0 | 35.050069 |
| **917** | 40 | 51.869470 |
| **933** | 60 | 48.885582 |
| **60** | 80 | 45.315364 |
| **644** | 40 | 51.013495 |
| **760** | 80 | 56.978047 |
| **793** | 60 | 65.955664 |

After That Accuracy score is determined using r2_score from sklearn-metrics-

It showed 0.71 i.e. 71% accuracy results. (Figure-14)

```
In [45]: from sklearn.metrics import r2_score

In [46]: r2_score(y_tt,yrfr)

Out[46]: 0.7108141613466047
```

## 6.2. CREATION OF MICRO-WEB FRAMEWORK USING FLASK:

**1.SERIALISATION: Python Pickle:** Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. After executing the model.py, we will get the get the pickle file (model.pkl). At first Python pickle serialize the object and then converts the object into a character stream so that this character stream contains all the information necessary to reconstruct the object in another python script.

```
rfr.fit(x_t,y_t)
pickle.dump(rfr,open('model.pkl','wb'))



# In[42]:

model = pickle.load(open('model.pkl','rb'))
yrfr=rfr.predict(x_tt)
```

## Python Pickle Dump

To do the serialisation pickle.dump() function is used to store the object data to the file. pickle.dump() function takes 3 arguments. The first argument is the object that you want to store. The second argument is the file object you get by opening the desired file in write-binary (wb) mode. And the third argument is the key-value argument. This argument defines the protocol. There are two type of protocol – pickle.HIGHEST_PROTOCOL and pickle.DEFAULT_PROTOCOL. See the sample code to know how to dump data using pickle.

## Python Pickle Load

To retrieve pickled data, the steps are quite simple. You have to use pickle.load() function to do that. The primary argument of pickle load function is the file object that you get by opening the file in read-binary (rb) mode.

Here rb and wb is the mode.'rb' means read-binary mode and 'wb' means write-binary mode.

**2.Routing:** Figure-15

app.py > ...

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)

model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='WATER QUALITY PARAMETER VALUE = {}'.format(output))

@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

@app.route('/slider')
def slider():
    return render_template('slider.html')

@app.route('/documentation')
def documentation():
    return render_template('docx.html')
if __name__ == "__main__":
    app.run(debug=True)
```

Here,

- ✓ app variable is the instance of the Flask Web-Framework.

- ✓ @app.route(/predict) – acts as decorator and it leads to the destination route.app.route('/') refers to the parent page
- ✓ render_template - it is used to render HTML which will display in the user's browser. All the HTML PAGES must be kept in the 'templates' directory. Otherwise you will get the error – "templates not found".
- ✓ You will have to write separate function for each and every web page with decorator route and must return render_template('demo.html'). Here demo.html is the example.
- ✓ name variable will be always equal to __main__ because we are executing app.py directly.
- ✓ app.run - it invokes the web-server on.
- ✓ debug=True - It set debug-mode active so that it helps in debugging. Every time something is changed, that Re-Execution of the Deployment will occur automatically, all you have to do is to refresh the webpage – Any sort of change will be reflected to the web application.
- ✓ To Access the Web Application Go to the destination folder and then type python app.py and then go to : -http://127.0.0.1:5000/ or http://localhost:5000/ .

## 6.3. CREATION OF USER INTERFACE(UI):

## HOME/PREDICT PAGE:

## HTML CODE:

```
templates > index.html > html > body > div.login
1  <html lang="en" dir="ltr">
2      <head>
3          <meta charset="utf-8">
4          <title>ML API</title>
5          <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
6      </head>
7  <header>
8          <div class="menu-toggle" id="hamburger">
9              <i class="fas fa-bars"></i>
10         </div>
11         <div class="overlay"></div>
12         <div class="container">
13             <nav>
14                 <h1 class="brand"><a href="index.html">CHAIN<span>CODERS</span></a></h1>
15                 <ul>
16                     <li><a href="#">Home</a></li>
17                     <li><a href="{{ url_for('documentation')}}">Documentaion</a></li>
18                     <li><a href="#">Predict</a></li>
19                     <li><a href="{{ url_for('slider')}}">GraphSlides</a></li>
20                 </ul>
21             </nav>
22         </div>
23 </header>
24 <body>
25     <div class="login">
26         <h1><b>PREDICTION OF WATER QUALITY</b></h1>
27         <br>
28         <!-- Main Input For Receiving Query to our ML -->
29         <form action="{{ url_for('predict')}}"method="post">
30             <input type="text" name="state" placeholder="STATE" required="required">
31             <input type="text" name="year" placeholder="YEAR" required="required"/>
32             <input type="text" name="npH" placeholder="PH" required="required" />
33             <input type="text" name="ndo" placeholder="D.O" required="required" />
34             <input type="text" name="nec" placeholder="EC" required="required" />
35             <input type="text" name="nbdo" placeholder="B.D.O" required="required" />
36             <input type="text" name="nna" placeholder="NITRATE QUATITY" required="required" />
37             <button type="submit" class="btn btn3"><b>P r e d i c t</b></button>
38         </form>
```

```html
    <br><br>
        <h1 style="color: rgb(40, 218, 194);font-size: small;">  {{ prediction_text }}  </h1>
    </div>
</body>
</html>
```

## DOCUMENTATION PAGE

## HTML CODE

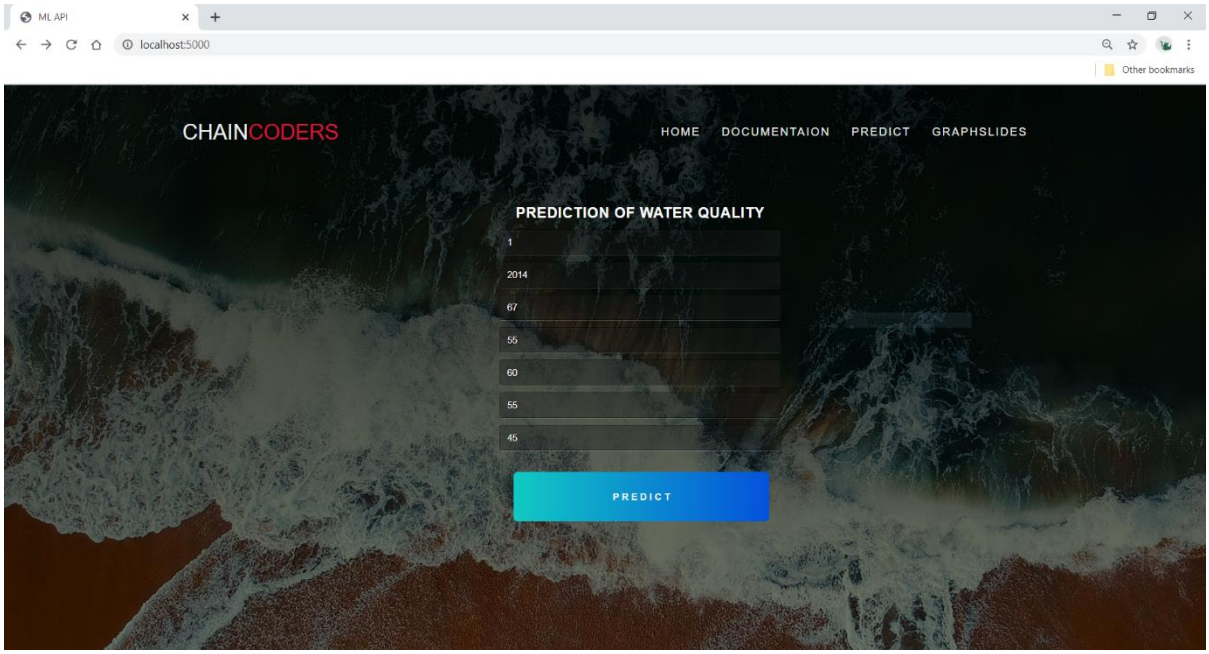
```html
templates > docx.html > html > body
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <link rel="stylesheet"  href="{{ url_for('static', filename='css/style.css') }}">
    </head>
<header>
        <div class="menu-toggle" id="hamburger">
            <i class="fas fa-bars"></i>
        </div>
        <div class="overlay"></div>
        <div class="container">
            <nav>
                <h1 class="brand"><a href="index.html">CHAIN<span>CODERS</span></a></h1>
                <ul>
                    <li><a href="{{ url_for('home')}}">Home</a></li>
                    <li><a href="#">Documentaion</a></li>
                    <li><a href="{{ url_for('home')}}">Predict</a></li>
                    <li><a href="{{ url_for('slider')}}">GraphSlides</a></li>
                </ul>
            </nav>
        </div>
</header>
<body>
    <div class="login">
        <h1><b>DOCUMENTATION OF THE PROJECT</b></h1>
        <br>
        <br>
        <br>
        <a href="https://drive.google.com/file/d/1ivYv51vFFH0gDdQdXkkGEB2QwbV1b3wy/view?usp=sharing">
            <button type="submit" class="btn btn3"><b>C L I C K</b></button></a>
    </div>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <img class="image_d" src="{{ url_for('static', filename='css/statecodes.jpg') }}" alt="IMAGE IS BROKEN"
     style="width:30%;height:40%;display: block;
    margin-left: 38%;
    margin-right: 50%;border:2px solid crimson;">
</body>
</html>
```

## CSS CODE FOR BOTH INDEX.HTML (HOME PAGE) AND DOCX.HTML (DOCUMENTATION PAGE) STYLE.CSS

```css
static > css > # style.css > input:focus
/* CSS FOR BUTTON */
.btn{
    border: none;
    margin: 20px;
    padding: 24px;
    width: 220px;
    font-family: "montserrat",sans-serif;
    text-transform: uppercase;
    border-radius: 6px;
    cursor: pointer;
    color: #fff;
    background-size: 200%;
    transition: 0.6s;
    width: 363px;
    height: 70px;
  }
  .btn3{
    background-image: linear-gradient(to left, #12CBC4, #0652DD, #12CBC4);
  }
  .btn:hover{
    background-position: right;
  }
```

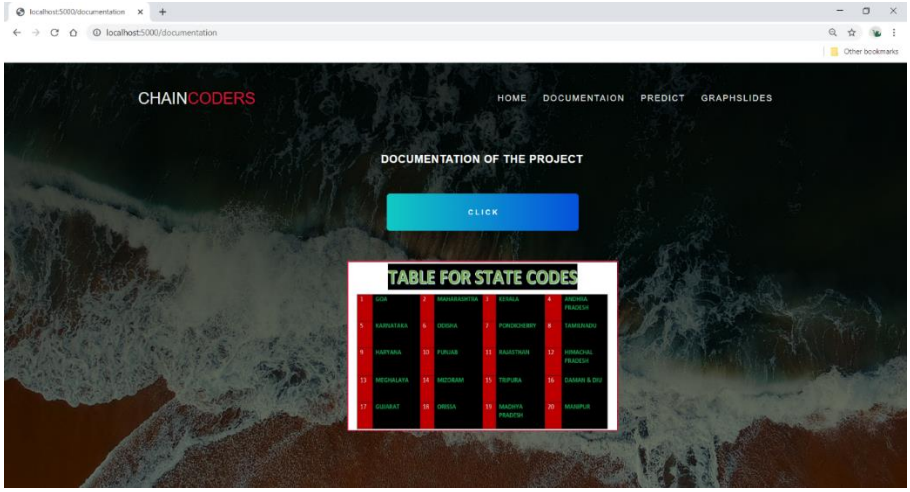
```css
.login {
    position: absolute;
    top: 40%;
    left: 50%;
    margin: -150px 0 0 -150px;
    width:400px;
    height:400px;
}

.login h1 { color: #fff;
    text-shadow: 0 0 10px rgba(0,0,0,0.3);
    letter-spacing:1px;
    text-align:center;
    }

input {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(0,0,0,0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: #fff;
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }
```

## WEB UI FOR HOME PAGE




(Figure-16)

## WEB UI FOR DOCUMENTATION PAGE




(figure-17)

## SLIDER.HTML

```
templates > slider.html > html > body
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <link rel="stylesheet"  href="{{ url_for('static', filename='css/slider.css') }}">
    </head>
<header>
        <div class="menu-toggle" id="hamburger">
            <i class="fas fa-bars"></i>
        </div>
        <div class="overlay"></div>
        <div class="container">
            <nav>
                <h1 class="brand"><a href="index.html">CHAIN<span>CODERS</span></a></h1>
                <ul>
                    <li><a href="{{ url_for('home')}}">Home</a></li>
                    <li><a href="{{ url_for('documentation')}}">Documentaion</a></li>
                    <li><a href="{{ url_for('home')}}">Predict</a></li>
                    <li><a href="#">GraphSlides</a></li>
                </ul>
            </nav>
        </div>
</header>
<body>
    <body>
        <div class="slidershow middle">
            <div class="slides">
            <input type="radio" name="r" id="r1" checked>
            <input type="radio" name="r" id="r2">
            <input type="radio" name="r" id="r3">
            <input type="radio" name="r" id="r4">
            <input type="radio" name="r" id="r5">
            <div class="slide s1">
              <img src="{{ url_for('static', filename='css/1.jpg') }}" alt="64545">
            </div>
            <div class="slide">
              <img src="{{ url_for('static', filename='css/2.jpg') }}" alt="">
            </div>
            <div class="slide">
              <img src="{{ url_for('static', filename='css/3.jpg') }}" alt="">
            </div>
            <div class="slide">
              <img src="{{ url_for('static', filename='css/4.jpg') }}" alt="">
            </div>
            <div class="slide">
              <img src="{{ url_for('static', filename='css/5.jpg') }}" alt="">
            </div>
            </div>
```

```html
            <div class="navigation">
              <label for="r1" class="bar"></label>
              <label for="r2" class="bar"></label>
              <label for="r3" class="bar"></label>
              <label for="r4" class="bar"></label>
              <label for="r5" class="bar"></label>
            </div>
          </div>
        </body>
    </body>
    </html>
```

**SLIDER.CSS (**It also contains style for the navigation bar also and its transition)

static > css > # slider.css > .container

```css
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
html{
    font-size: 10px;
    font-family: "Roboto Cn", sans-serif;
}
a{
    text-decoration: none;
    color: #eee;
}
html{
    width: 100%;
    height: 100vh;
    background: linear-gradient(to bottom, rgba(0,0,0,.8), rgba(0,0,0,.5)), url("support_img.jpg") center no-repeat;
    background-size: cover;
    position: relative;
  overflow: hidden;
}

/* NAV BAR */
.container{
    max-width: 120rem;
    width: 90%;
    margin: 0 auto;
}
.menu-toggle{
    position: fixed;
    top: 2.5rem;
    right: 2.5rem;
    color: #eeeeee;
    font-size: 3rem;
    cursor: pointer;
    z-index: 1000;
    display: none;
}
nav{
    padding-top: 5rem;
    display: flex;
    justify-content: space-between;
    align-items: center;
    text-transform: uppercase;
    font-size: 1.6rem;
}
```

## SLIDER.CSS(CONTINUED)

```
.brand{
    font-size: 3rem;
    font-weight: 300;
    transform: translateX(-100rem);
    animation: slideIn .5s forwards;
}
.brand span{
    color: crimson;
}
nav ul{
    display: flex;
}
nav ul li{
    list-style: none;
    transform: translateX(100rem);
    animation: slideIn .5s forwards;
}
nav ul li:nth-child(1){
    animation-delay: 0s;
}
nav ul li:nth-child(2){
    animation-delay: .5s;
}
nav ul li:nth-child(3){
    animation-delay: 1s;
}

nav ul li:nth-child(4){
    animation-delay: 1.5s;
}

nav ul li a{
    padding: 1rem 0;
    margin: 0 3rem;
    position: relative;
    letter-spacing: 2px;
}

nav ul li a:last-child{
    margin-right: 0;
}
nav ul li a::before,
nav ul li a::after{
    content: '';
    position: absolute;
    width: 100%;
```

static > css > # slider.css > nav ul li a::before

```
    height: 2px;
    background-color: crimson;
    left: 0;
    transform: scaleX(0);
    transition: all .5s;
}

nav ul li a::before{
    top: 0;
    transform-origin: left;
}

nav ul li a::after{
    bottom: 0;
    transform-origin: right;
}

.overlay{
    background-color: rgba(0,0,0,.95);
    position: fixed;
    right: 0;
    left: 0;
    top: 0;
    bottom: 0;
    transition: opacity 650ms;
    transform: scale(0);
    opacity: 0;
   display: none;
}

nav ul li a:hover::before,
nav ul li a:hover::after{
    transform: scaleX(1);
}

@keyframes slideIn {
    from{

    }
    to{
        transform: translateX(0);
    }
}
```

Sp Adobe Spark

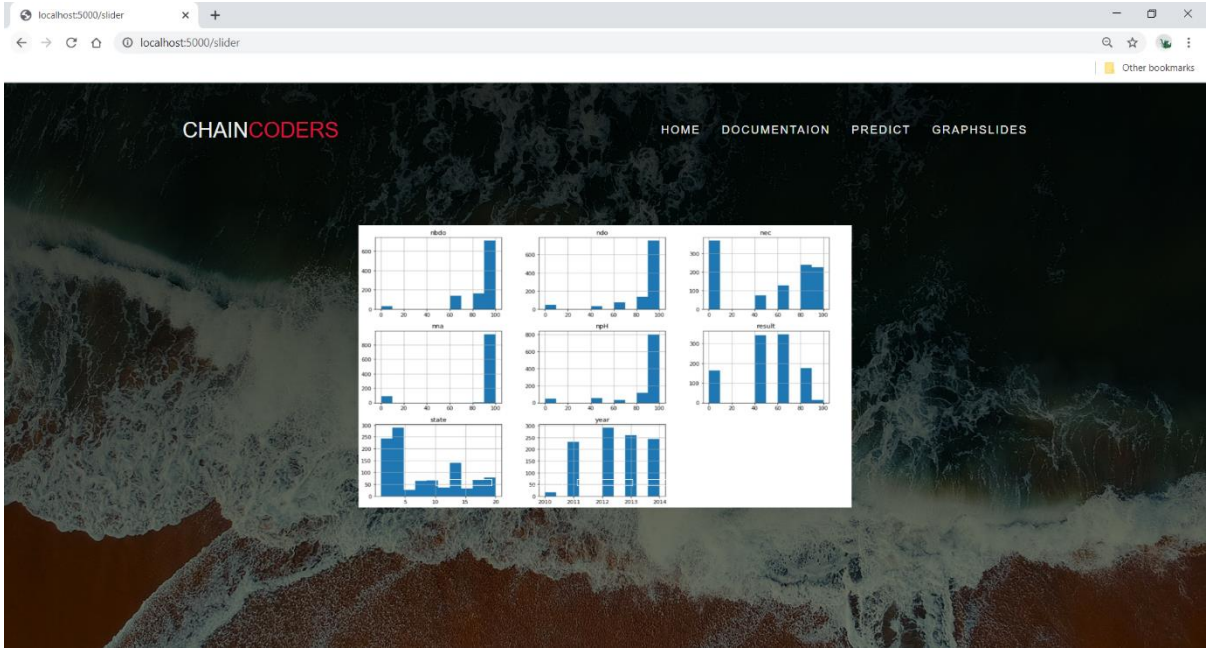
## SLIDER.CSS(CONTINUED)

```
            display: block;
        }

        nav{
            padding-top: 0;
            display: none;
            flex-direction: column;
            justify-content: space-evenly;
            align-items: center;
            height: 100vh;
            text-align: center;
        }

        nav ul{
            flex-direction: column;
        }

        nav ul li{
            margin-top: 5rem;
        }

        nav ul li a{
            margin: 0;
            font-size: 2.5rem;
        }

        .brand{
            font-size: 5rem;
        }

    .overlay.menu-open,
    nav.menu-open{
        display: flex;
        transform: scale(1);
        opacity: 1;
    }

}


/* SLIDER CODE */

.slidershow{
    width: 700px;
    height: 400px;
    overflow: hidden;
}
.middle{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%);
}
.navigation{
    position: absolute;
    bottom: 20px;
    left: 50%;
    transform: translateX(-50%);
    display: flex;
}
.bar{
    width: 80px;
    height: 10px;
    border: 0.5px solid rgb(255, 255, 255);
    margin: 10px;
    cursor: pointer;
    transition: 0.4s;
}
.bar:hover{
    background: crimson;
}

input[name="r"]{
    position: absolute;
    visibility: hidden;
}

.slides{
    width: 500%;
    height: 100%;
    display: flex;
}

.slide{
    width: 20%;
    transition: 0.6s;
}
.slide img{
    width: 100%;
    height: 100%;
```

Sp Adobe Spark

```

#r1:checked ~ .s1{
  margin-left: 0;
}
#r2:checked ~ .s1{
  margin-left: -20%;
}
#r3:checked ~ .s1{
  margin-left: -40%;
}
#r4:checked ~ .s1{
  margin-left: -60%;
}
#r5:checked ~ .s1{
  margin-left: -80%;
}

```

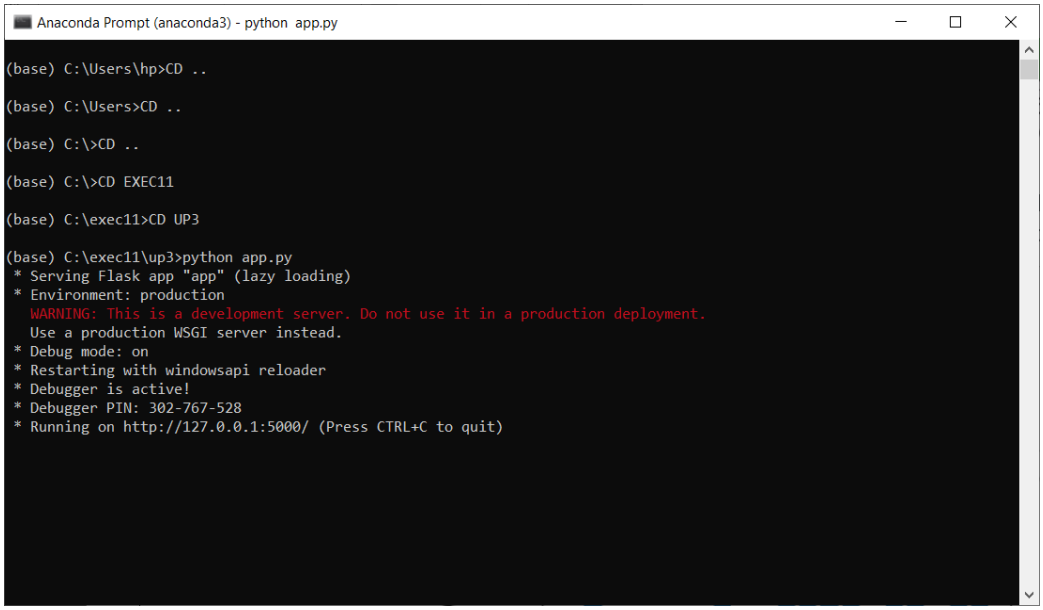
## WEB UI FOR SLIDER.HTML



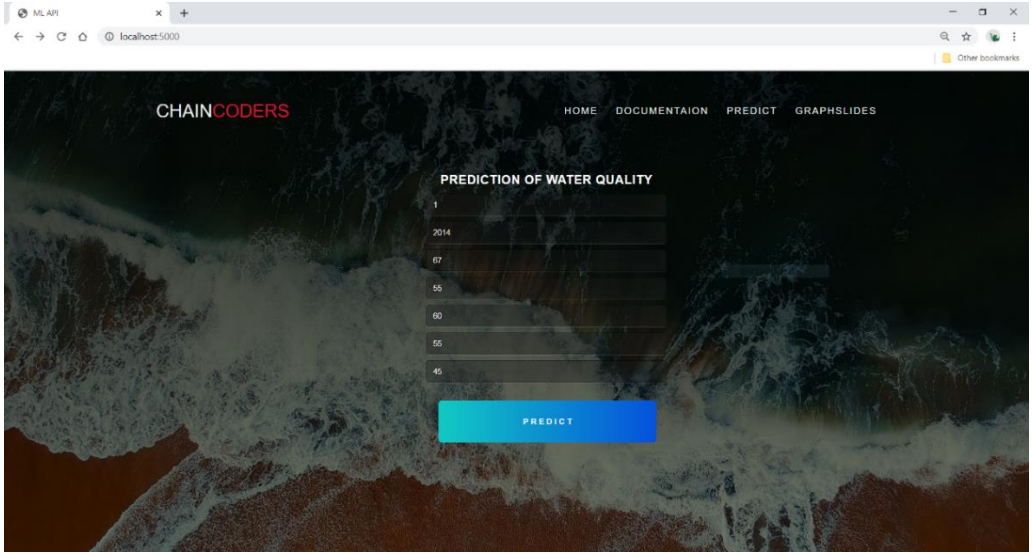

(Figure-18)

## 6.4. LAUNCHING IT IN THE LOCAL WEB SERVER

- Open Anaconda Prompt
- Anaconda Prompt comes with pre-installed flask on its virtual Conda Environment. Hence You don't need to install Flask, Pickle and other things explicitly.
- Go to the destination folder where your app.py (Flask-Code) is present.
- Run it by the command – "python app.py".
- You will see that Loading of Your Flask Application has been initiated.
- It will tell you to go to http://127.0.0.1:5000/ where 127.0.0.1 is nothing but Localhost.
- Work is Done. You will see your web running smoothly.
- After Work Is done. Go back to the Anaconda Prompt and Press CTRL+C to stop the web-server.

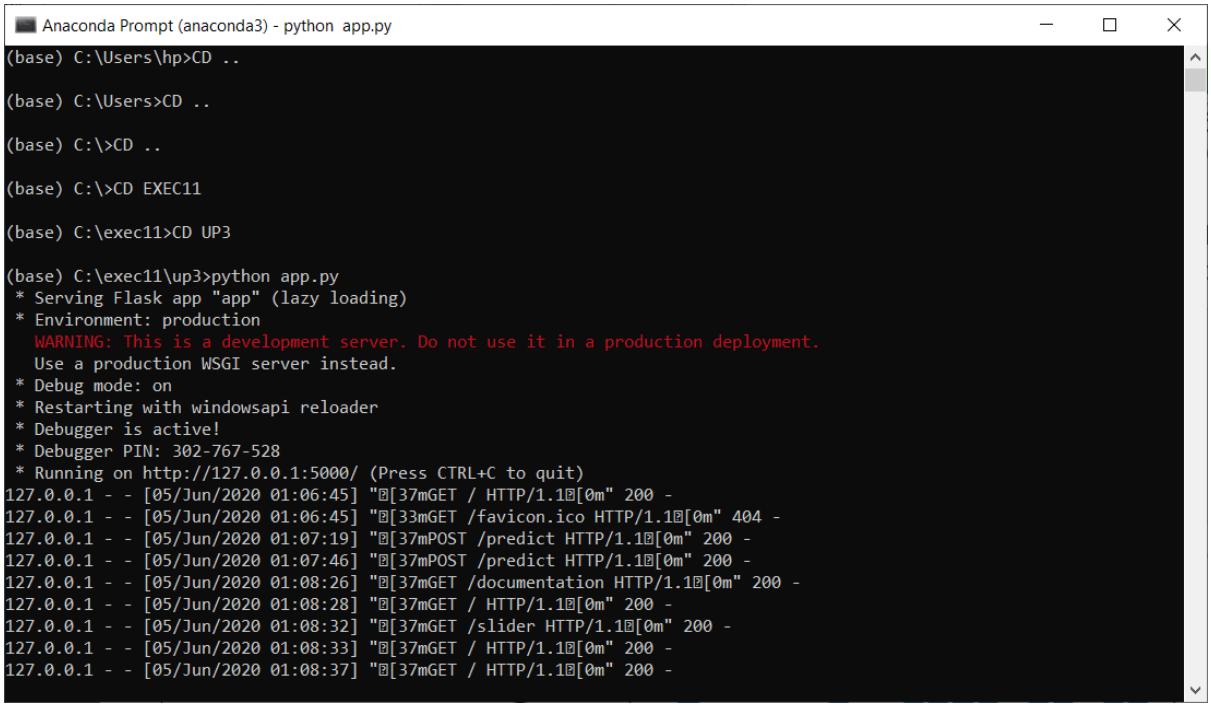



(figure-19)




(Figure-20)

You will see Every Activity occurred to the Web Application from The Command Prompt.



```
Anaconda Prompt (anaconda3) - python  app.py
(base) C:\Users\hp>CD ..

(base) C:\Users>CD ..

(base) C:\>CD ..

(base) C:\>CD EXEC11

(base) C:\exec11>CD UP3

(base) C:\exec11\up3>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 302-767-528
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Jun/2020 01:06:45] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:06:45] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -
127.0.0.1 - - [05/Jun/2020 01:07:19] "[37mPOST /predict HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:07:46] "[37mPOST /predict HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:08:26] "[37mGET /documentation HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:08:28] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:08:32] "[37mGET /slider HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:08:33] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [05/Jun/2020 01:08:37] "[37mGET / HTTP/1.1[0m" 200 -
```

(Figure-21)

## 6.5. LAUNCHING IT IN THE AWS EC2 INSTANCE:

Before Launching One Small Work needs to be done. Do this to your app.py file.

```
if __name__ == "__main__":
    app.run(host='0.0.0.0',port=8080)
```

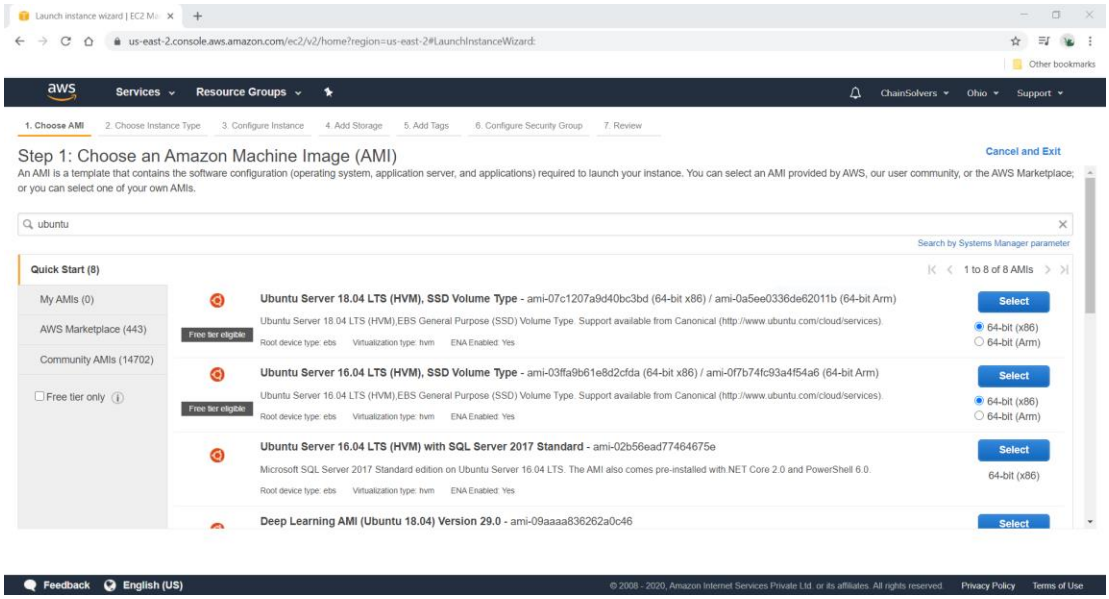
## AMAZON AWS EC2 (ELASTIC COMPUTE CLOUD):

An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. AWS is a comprehensive, evolving cloud computing platform; EC2 is a service that allows business subscribers to run application programs in the computing environment.

### Steps To Launch:

- 1.CREATE AN AWS ACCOUNT AND LOG IN AS ROOT USER
- 2.SEARCH EC2 AND GO TO THAT.
- 3.CHOOSE AN AMAZON MACHINE IMAGE (AMI)

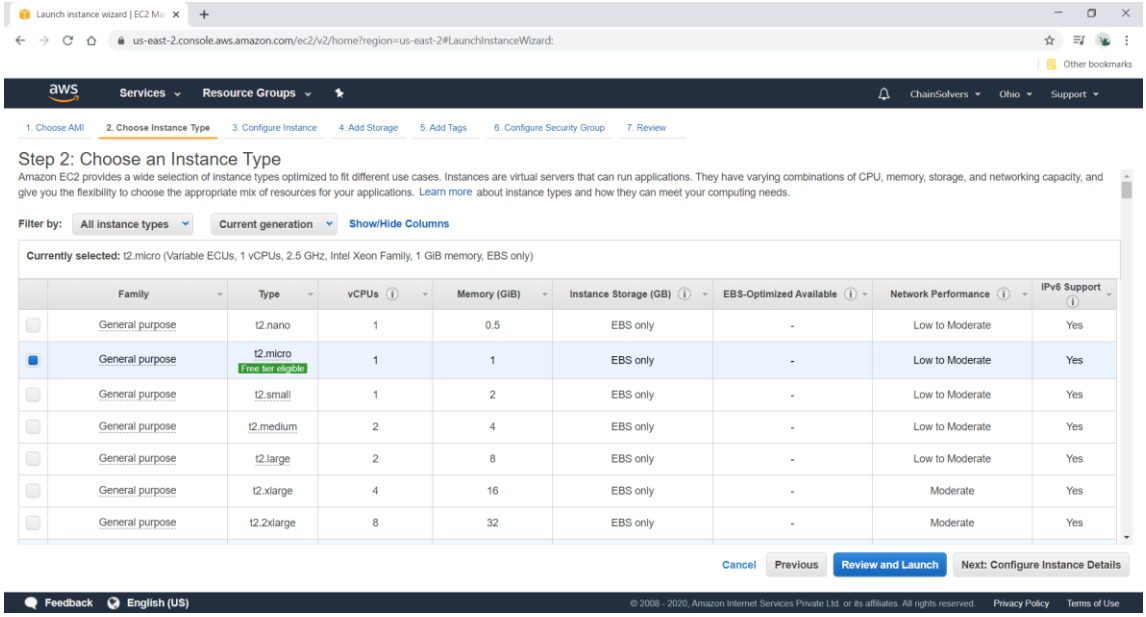
An AMI is a template that contains the software configuration (Operating System, Application Server and applications) required to launch your Instance. You can select an AMI provided by AWS, AWS marketplace, AWS Community or you can select one of your own AMIs. Figure-22.



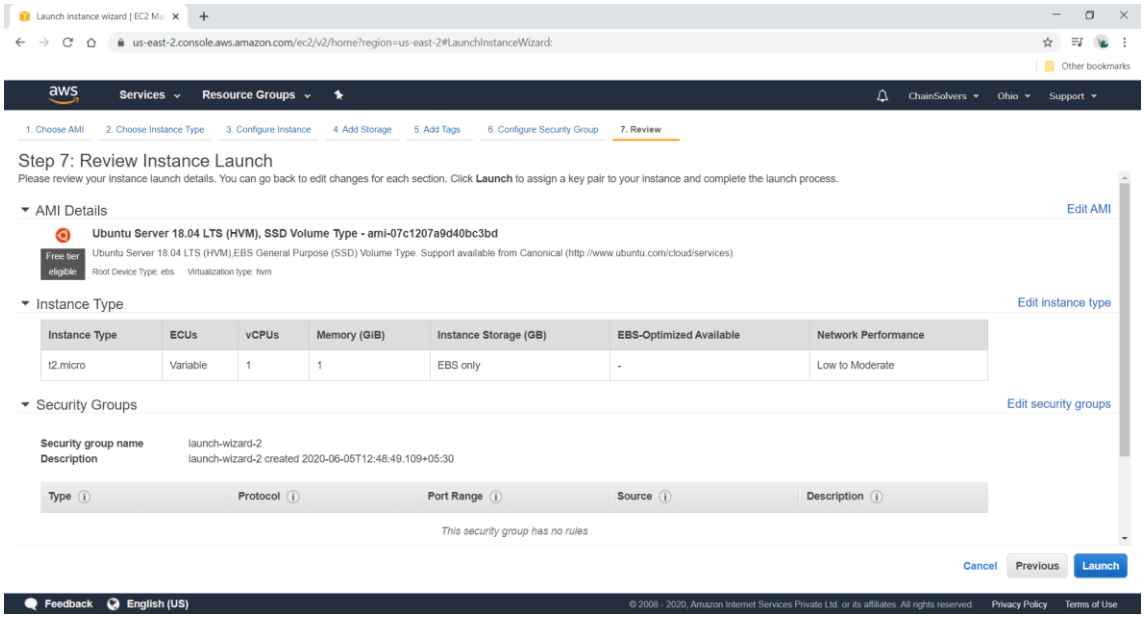

To do this, Ubuntu Server 18.04 LTS of 64-bit was selected.

- 4.CHOOSE AN INSTANCE TYPE: Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combination of CPU, memory, storage and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.

For this project, General purpose "t2micro" Instance was used and other configurations of this Instance are in the figure below. Figure-23




➢ 5.Click on Review and Launch and You will all the details of your selected Instance type.

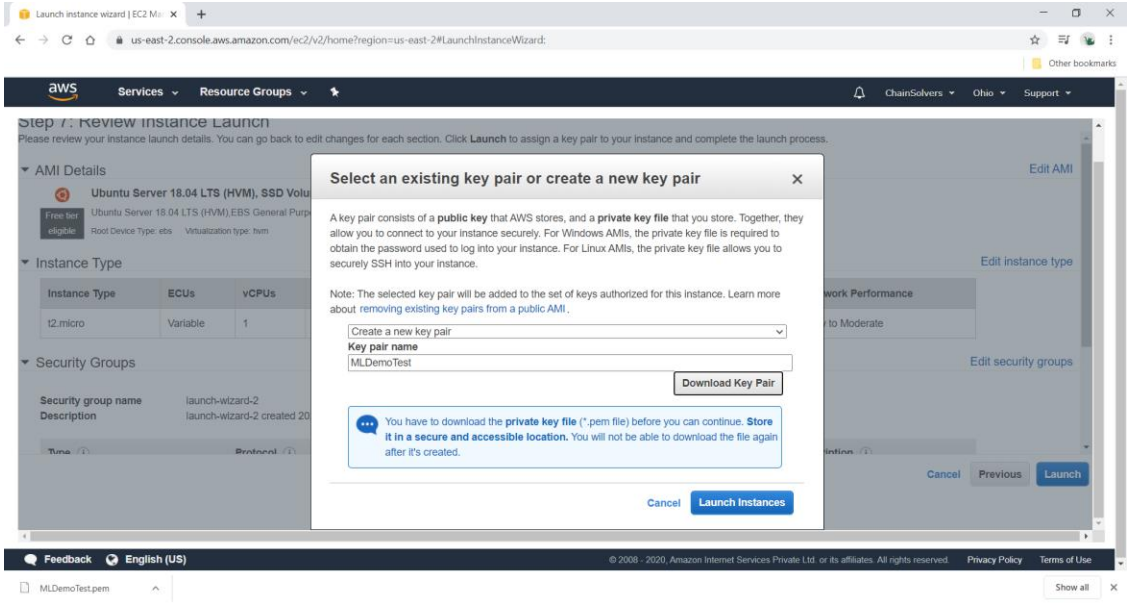



➢ 6.Click on Launch. Now this time it will ask to create a new key pair.

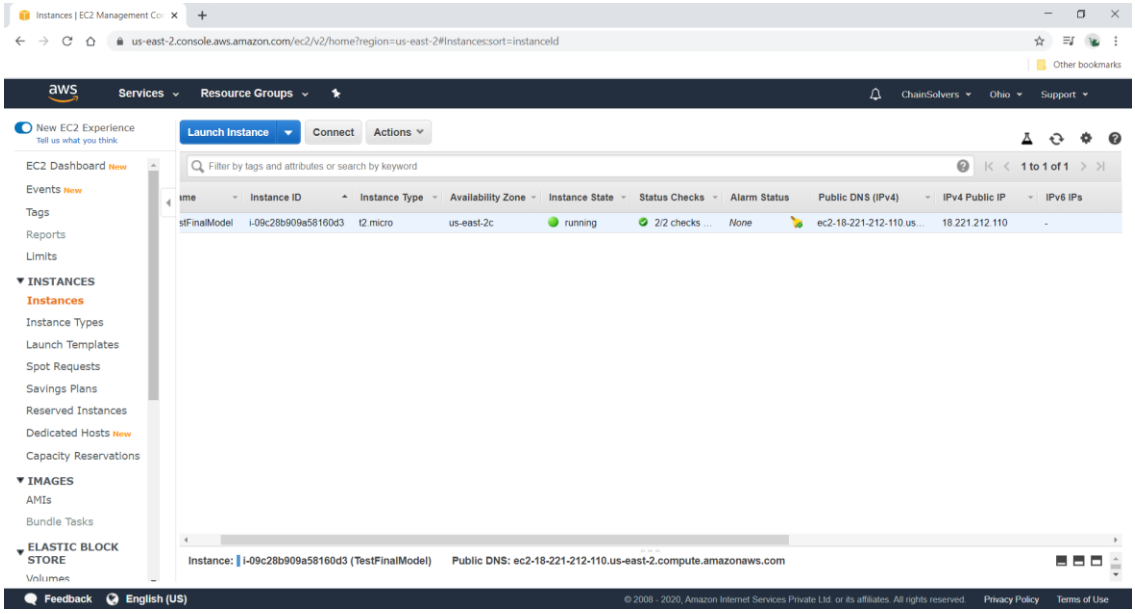
**<u>Key Pair:</u>** A key pair consists of a public key that AWS stores and a private key file that you store. Together they allow you to connect to your instance securely. For Linux AMI, the private key file allows you to securely SSH into Your instance.

After creating one must download the key pair or the private key file (*.pem file) to the Project Directory before launching your instance otherwise he won't be able to access the instance.

Now Click on "<u>Launch Instances</u>". Figure-24

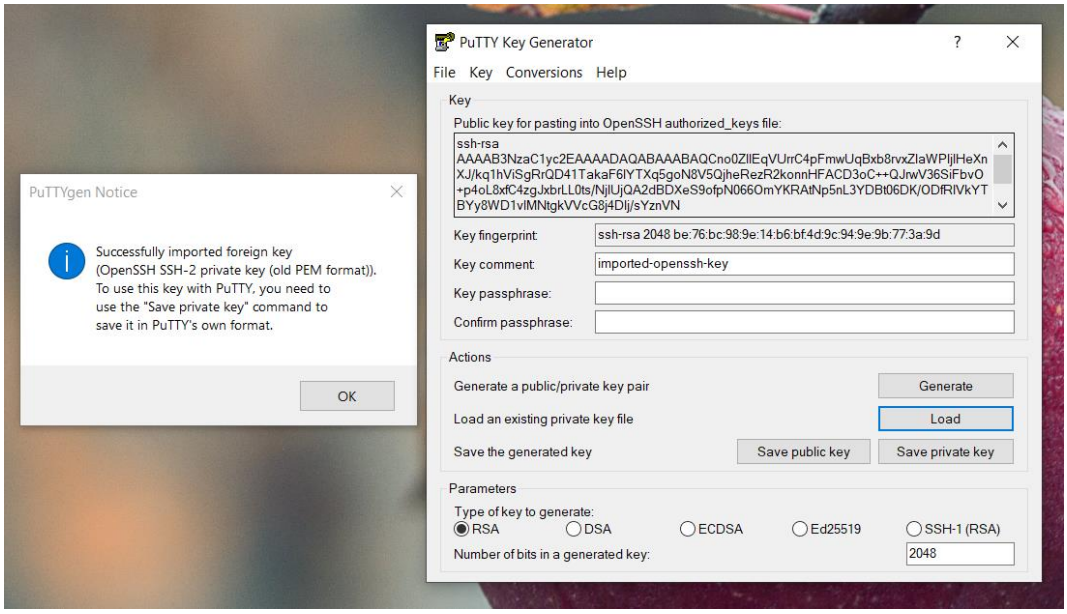


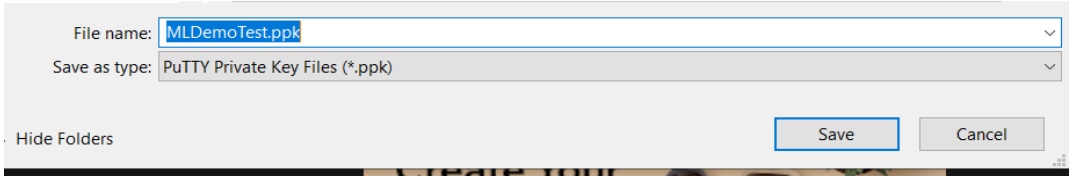

- 7.After Launching the Instance will be running within few minutes. You can check from EC2 Dashboard > Instances. Figure-25




- 8.OPEN "PUTTYGEN" AND LOAD THE MLDemoTest.pem file and click on save "Save Private Key" and then save the file as MLDemoTest.ppk to The Project Directory. Figure-26

**PUTTYGEN:** PuTTYgen is a key generator tool for creating SSH keys for PuTTY. It is analogous to the ssh-keygen tool used in some other SSH implementations. The basic function is to create public and private key pairs.

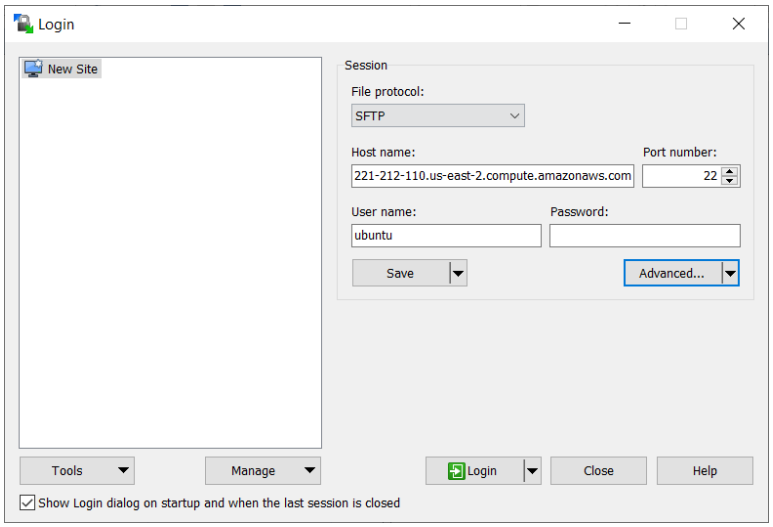



➢ 9.Now Open "WINSCP" Application.

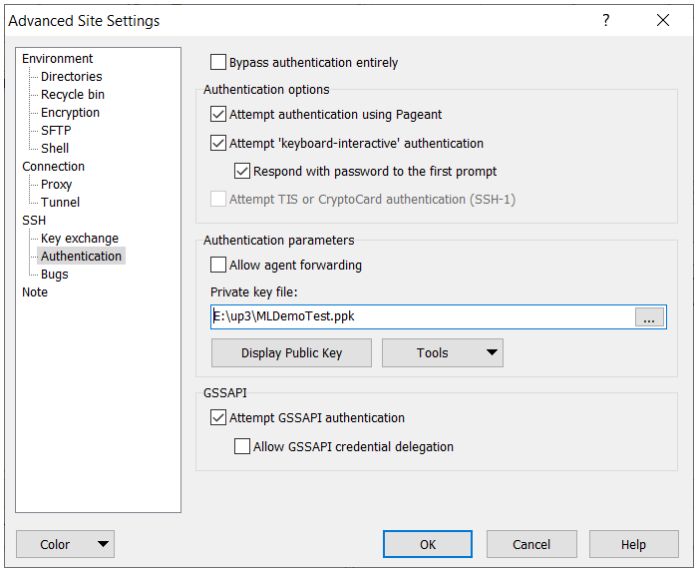
**WINSCP:** WinSCP is a free and open-source SFTP, FTP, WebDAV, Amazon S3 and SCP client for Microsoft Windows. Its main function is secure file transfer between a local and a remote computer.

- There are some details that need to be filled up.
- Host Name: Go to The Instances Dashboard. Then Go to Connect and Copy the DNS and paste it in the Host Name In WINSCP.
- Create username such as '<u>ubuntu</u>'.

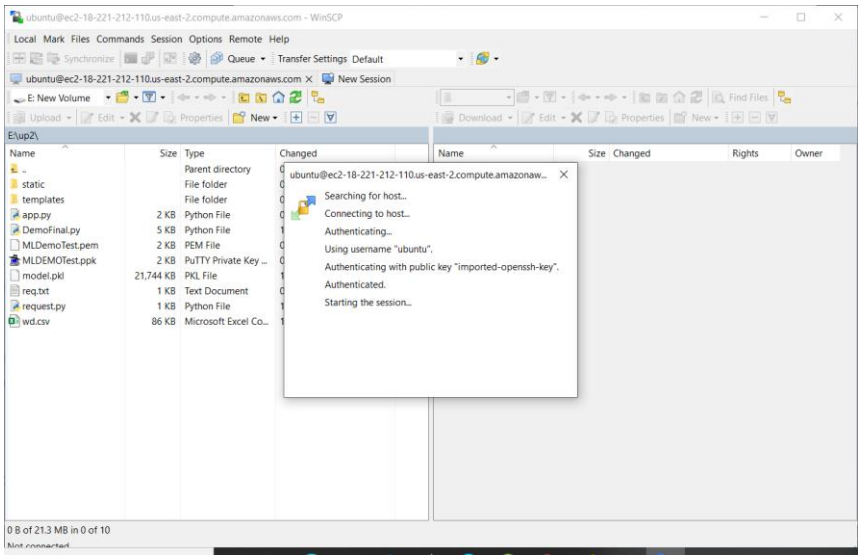



(Figure-27)

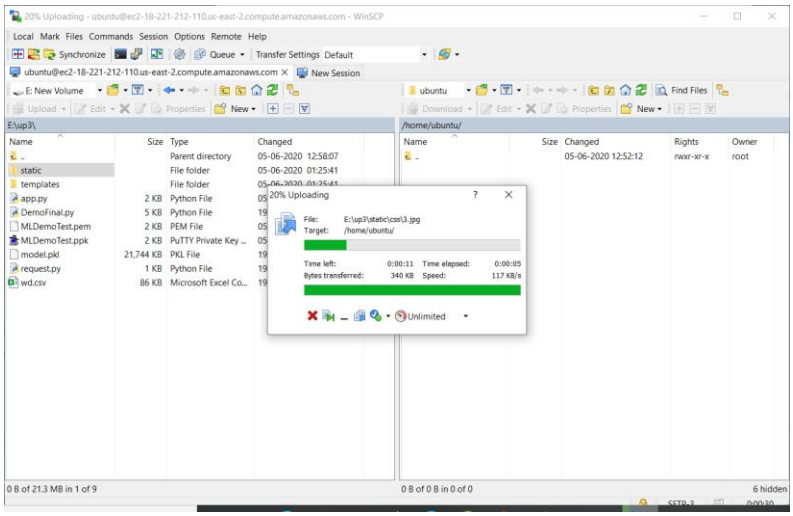
- Go to The Advanced option and Go to SSH then Authentication and upload the private key file (MLDemoTest.ppk) and Click on "OK".



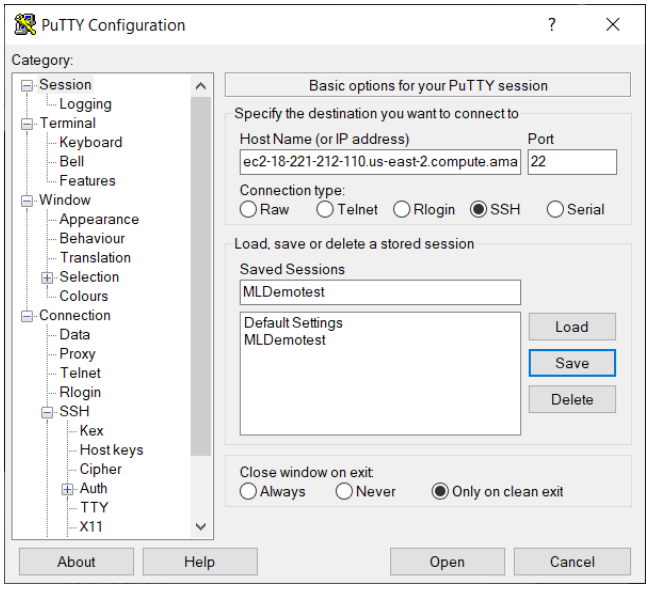

- Then Click on “Login”.

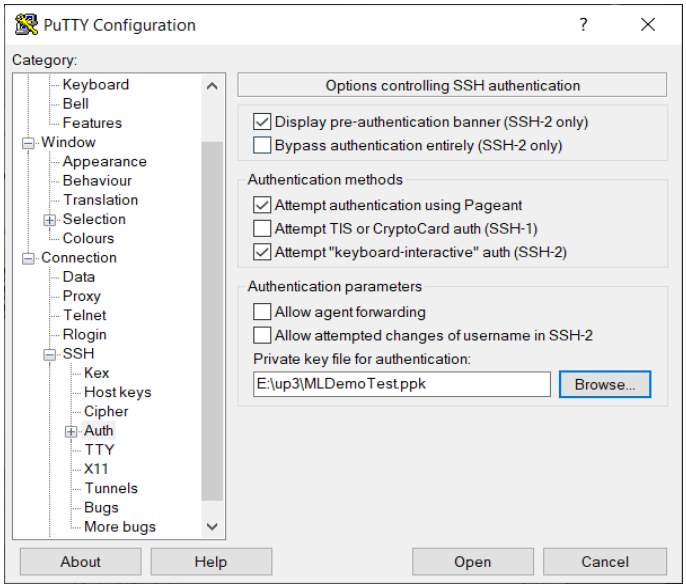



- After few seconds you can transfer the necessary files from your system to EC2 Instance. Just simply Drag and Drop across the barricade between the two. The files will be placed in the location “/home/ubuntu” in the EC2 Instance.

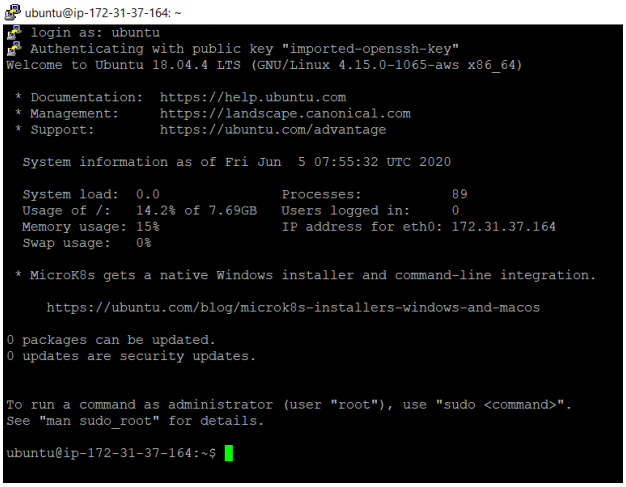


- 10.To Access the command prompt of the Instance, Open PUTTY and Give a session name and then Go to SSH and then Auth and again upload the private key file (MLDemoTest.ppk) and Go Back To The Session and Load and Save it and Then Click On Open. Figure-28,29,30




- 11.It will ask you a login id and type ubuntu as id which has been earlier to access the Instance.



- 12.Now You have to install all the necessary modules such as pip3, Flask and other python modules.

To Do this:

- Type this - > sudo apt-get update && sudo apt-get install python3-pip

PIP3 will be installed supporting the EC2 instance.

- Then Create a text file and List every python module you will need to execute your project.

Type this - > pip3 install -r req.txt

```
ubuntu@ip-172-31-37-164: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Jun  5 07:55:32 UTC 2020

  System load:  0.0               Processes:           89
  Usage of /:   14.2% of 7.69GB   Users logged in:     0
  Memory usage: 15%               IP address for eth0: 172.31.37.164
  Swap usage:   0%

 * MicroK8s gets a native Windows installer and command-line integration.

     https://ubuntu.com/blog/microk8s-installers-windows-and-macos

0 packages can be updated.
0 updates are security updates.


To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-37-164:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-37-164:~$ ls
app.py  model.pkl  req.txt  static  templates
ubuntu@ip-172-31-37-164:~$ sudo apt-get update && sudo apt-get install python3-pip
```

(figure-31)

- 13.Now Create Security Group:

**Security Group:** A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign the instance to up to five security groups. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic.

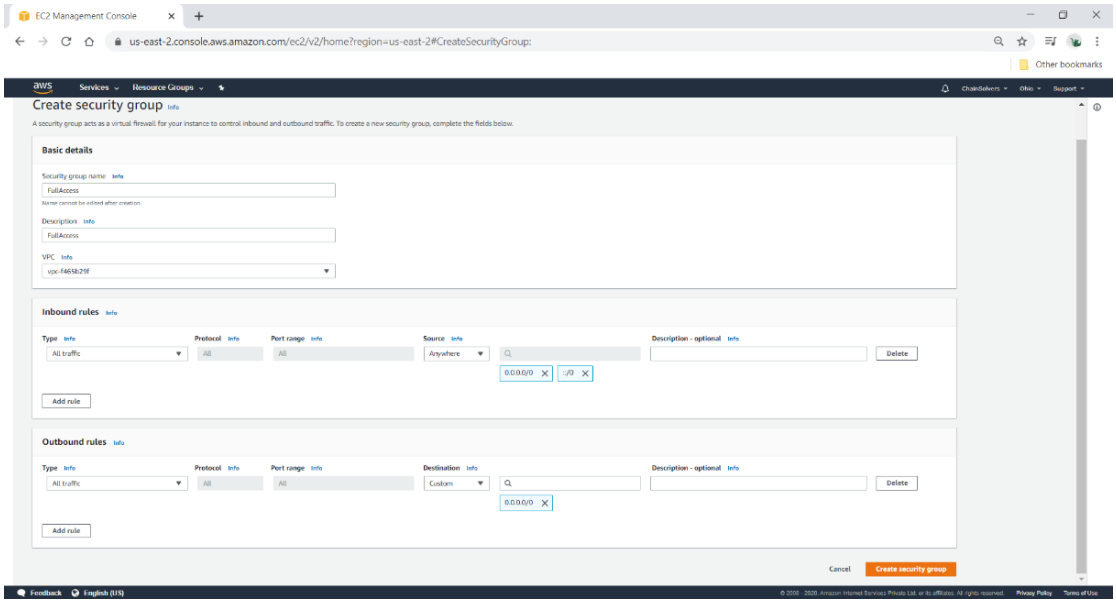
**Default Security Group for Your VPC**

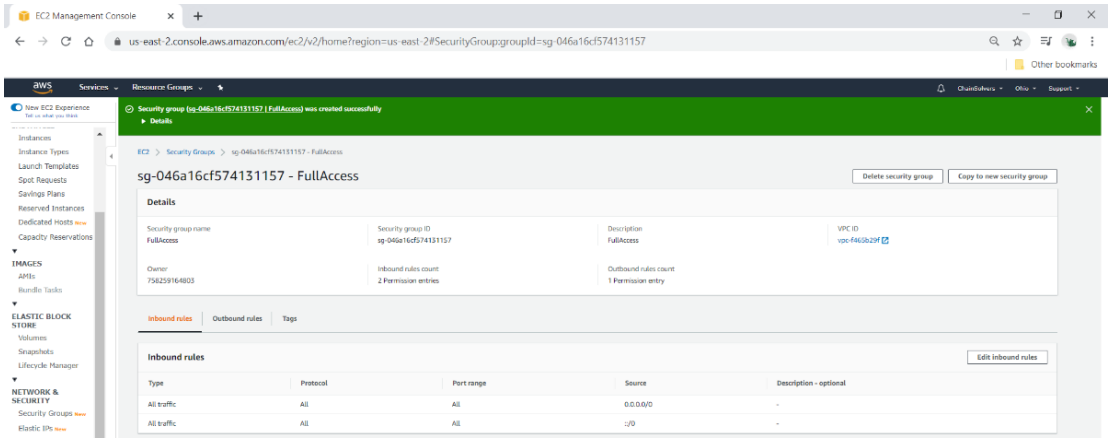
Your VPC automatically comes with a default security group. Each EC2 instance that you launch in your VPC is automatically associated with the default security group if you don't specify a different security group when you launch the instance.

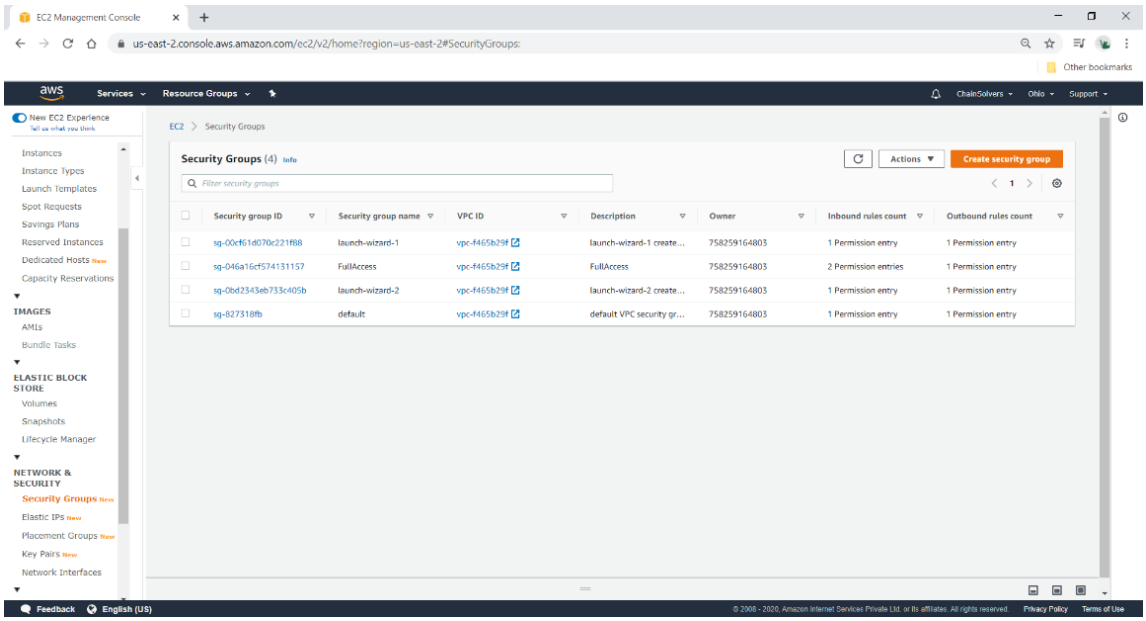
The following table describes the default rules for a default security group.

- ✓ Fill the Details As shown below. And then Click On “Create Security Group”.
- ✓ To Give Full Access you must select “All Traffic” as Type and “Anywhere” as source.
- ✓ It will take as “0.0.0.0/0” and “: :/0” by default.

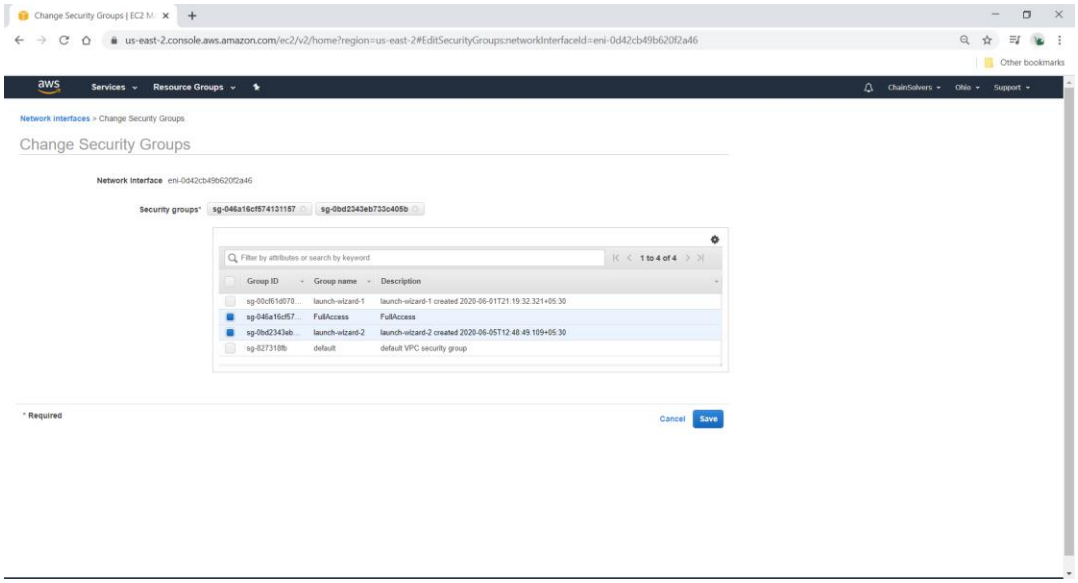
**0.0.0.0/0 and ::/0** - Means source can be any IP address, means from any system request is accepted, 0.0.0.0/0 represents IPv4 and ::/0 represents IPv6. (Figure-32,33,34)

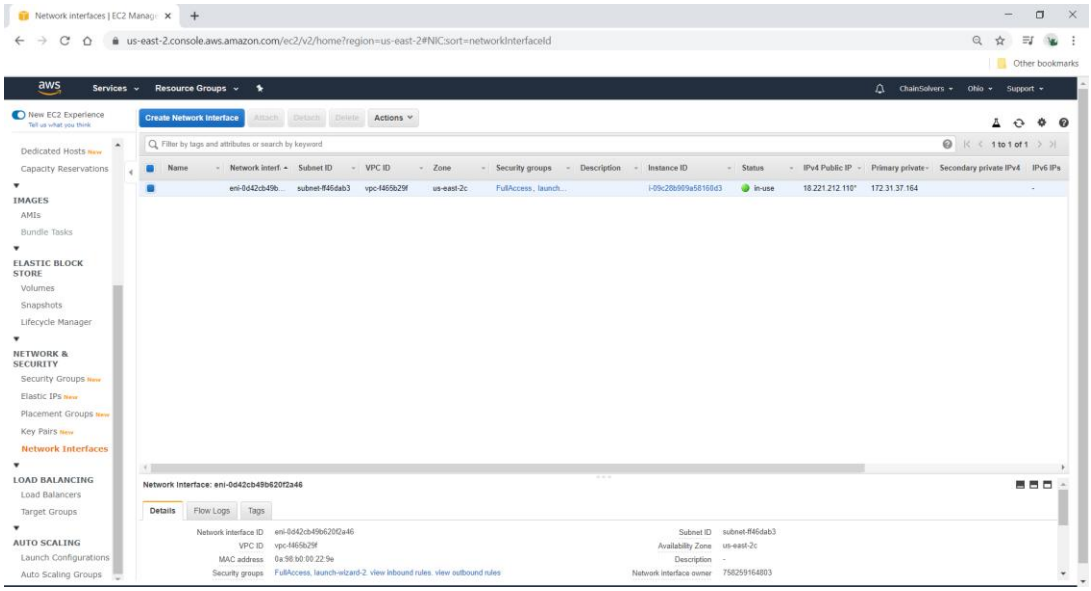


- 14.Now Go to the Network Interface and then Go to “Change Security Groups” and follow the image below and save it. (figure-35,36)



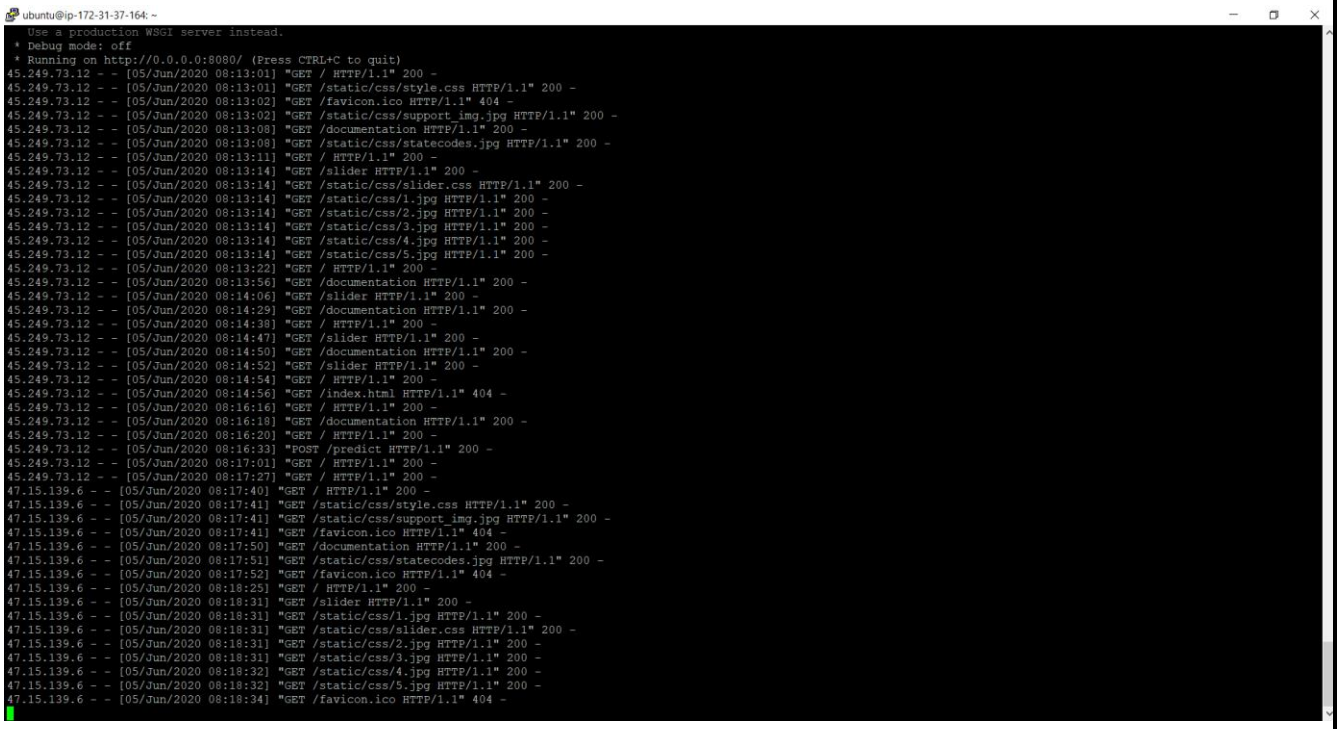



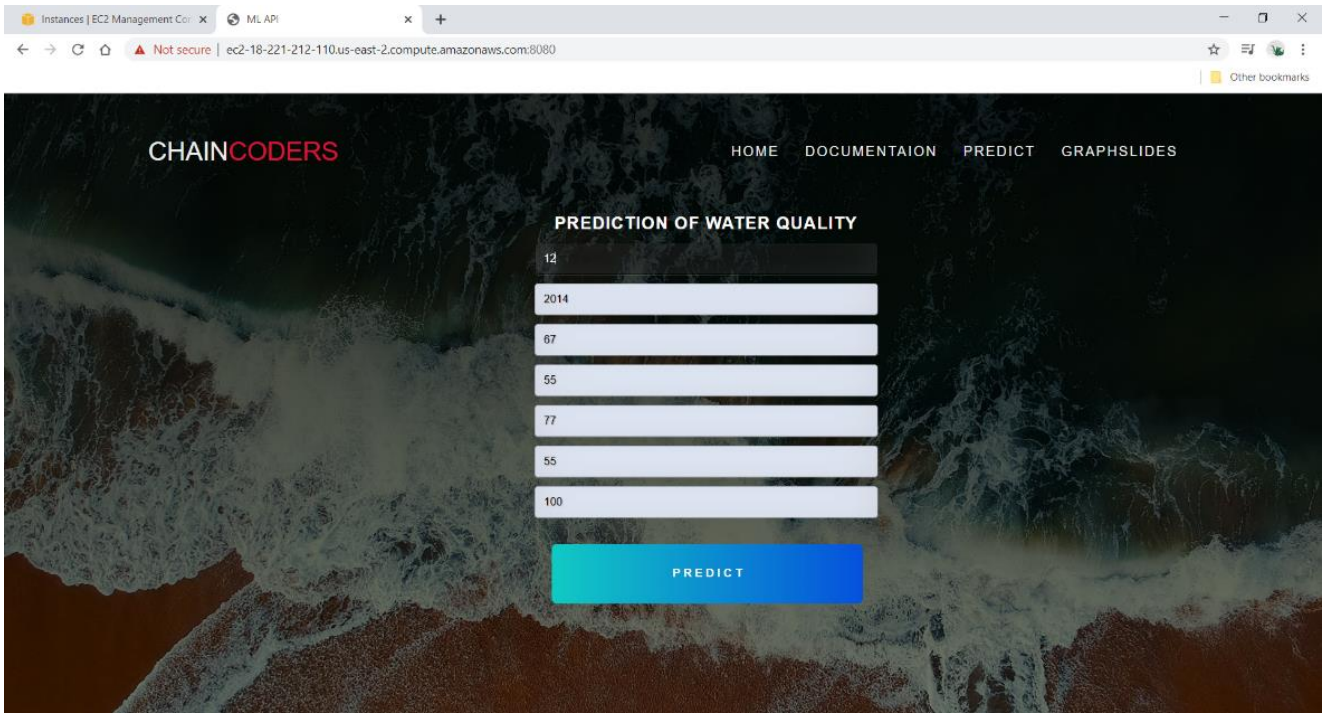

- 15.Now Go Back to the Instance Prompt (Ubuntu) and execute app.py by “python3 app.py”.

```
ubuntu@ip-172-31-37-164:~$ python3 app.py
/home/ubuntu/.local/lib/python3.6/site-packages/sklearn/base.py:334: UserWarning: Tryi
t lead to breaking code or invalid results. Use at your own risk.
  UserWarning)
/home/ubuntu/.local/lib/python3.6/site-packages/sklearn/base.py:334: UserWarning: Tryi
t lead to breaking code or invalid results. Use at your own risk.
  UserWarning)
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Your Project will be executed in your ubuntu Instance. (Figure-37)

- 16.Now Go to the “hostname:8080” i.e. “ec2-18-221-212-110.us-east-2.compute.amazonaws.com:8080” and the Web-Application will run on your browser seamlessly. Figure-38



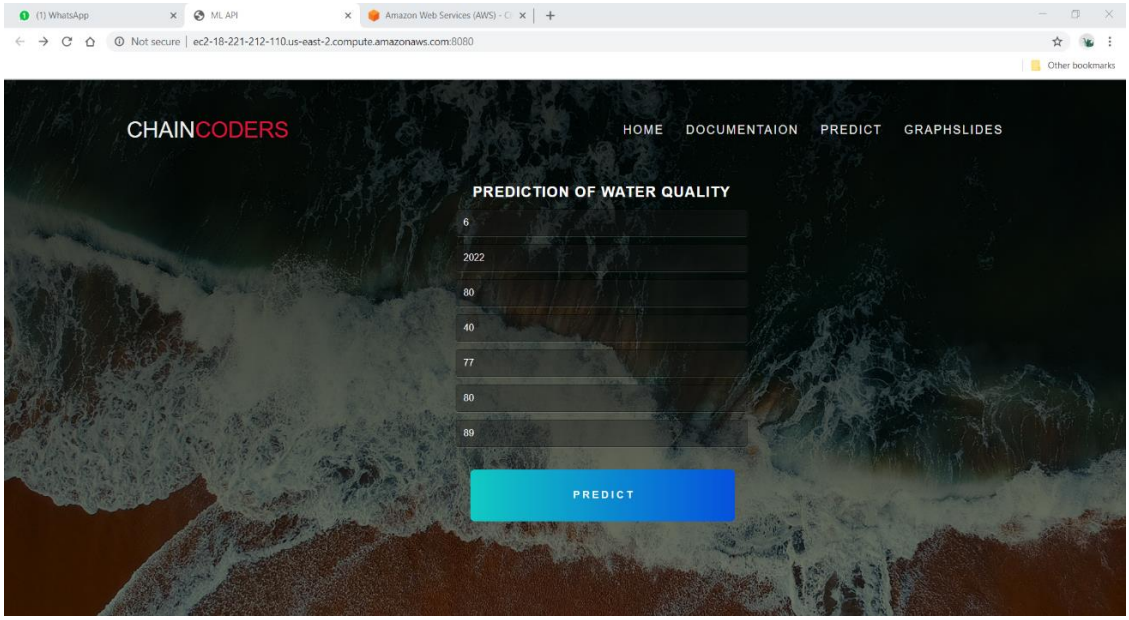

(figure-39)

# CHAPTER: 7

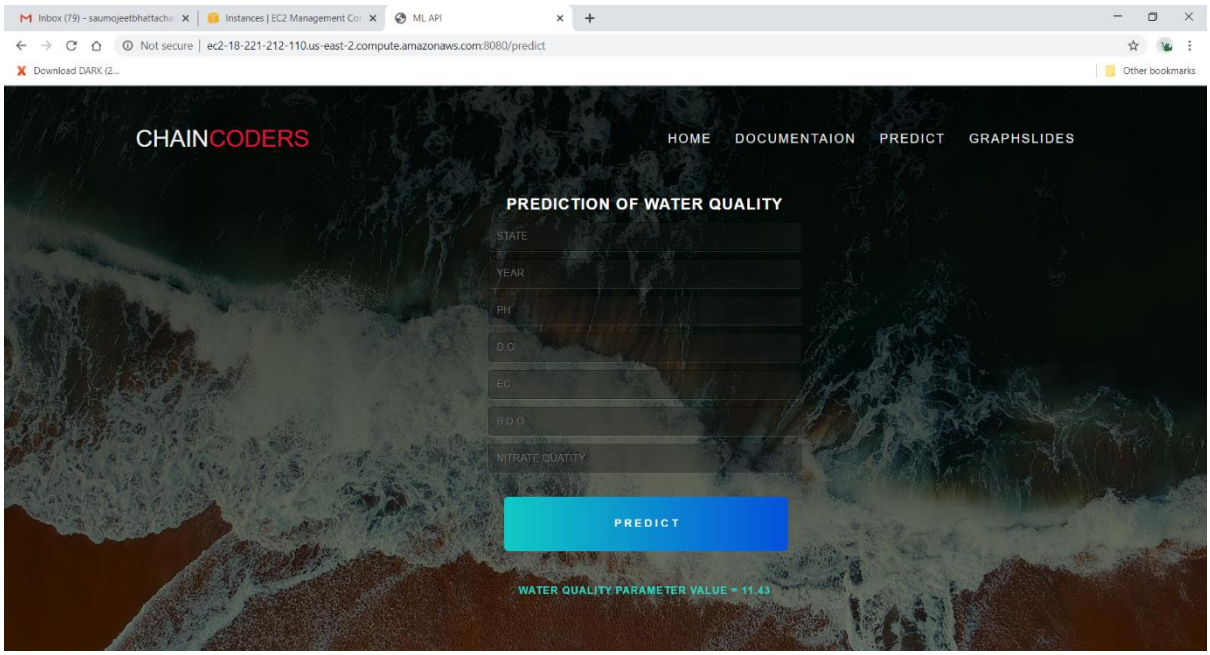
# TESTING

# TESTING

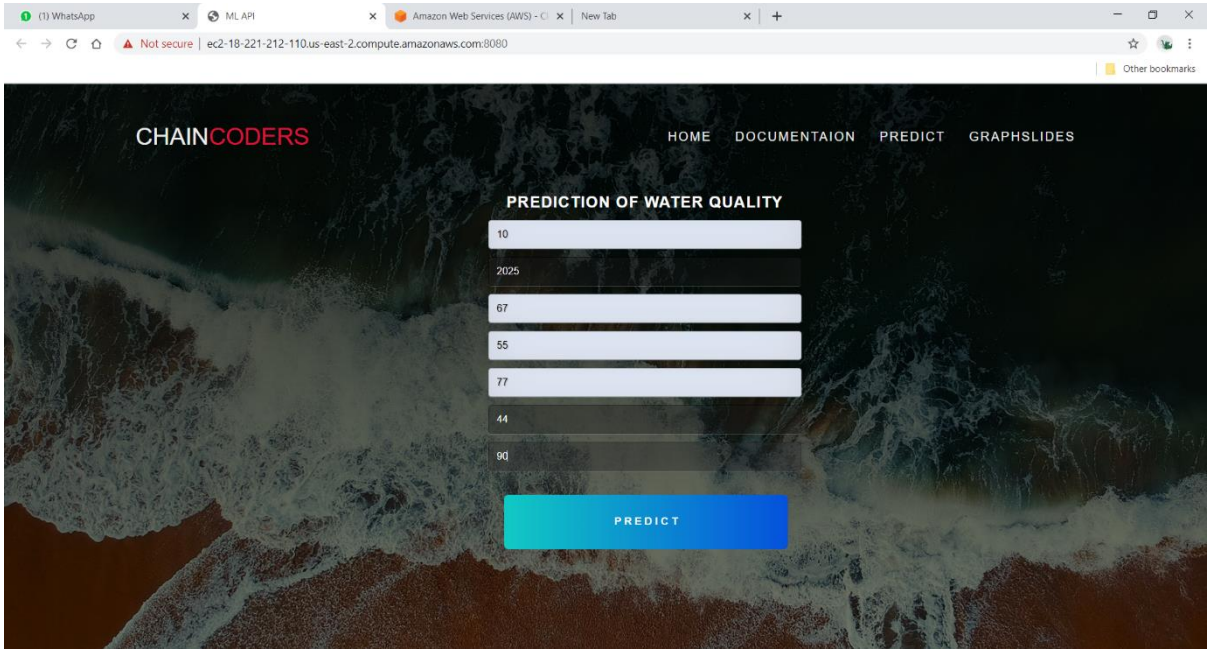
## 1st Test Case: (figure-40.a)



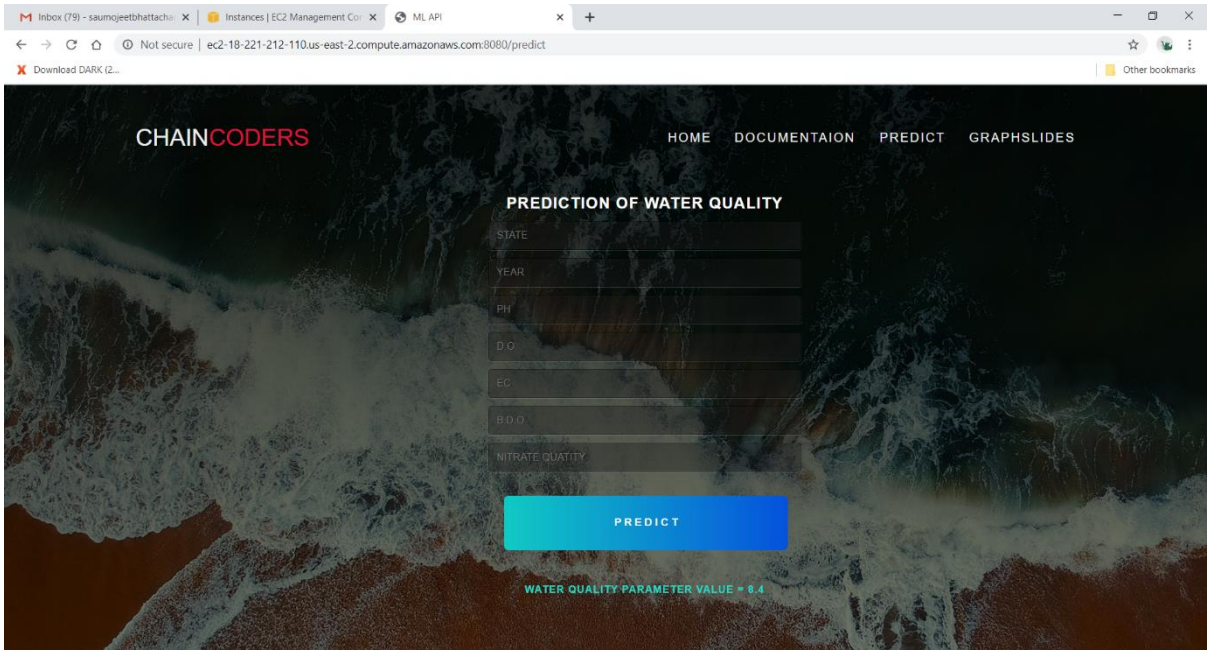

(Figure -40.b)



## 2nd Test Case: (Figure-41.a)




(Figure:-41.b)

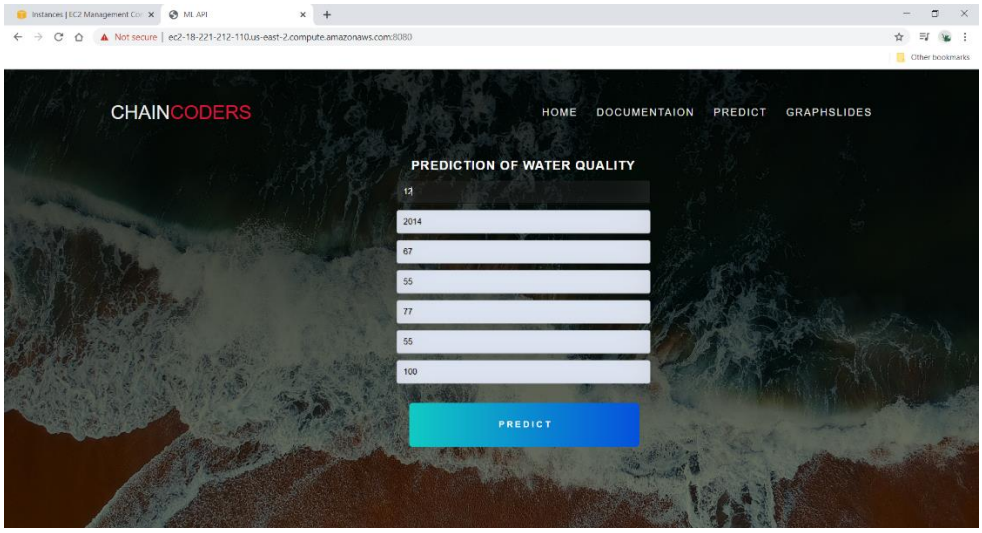


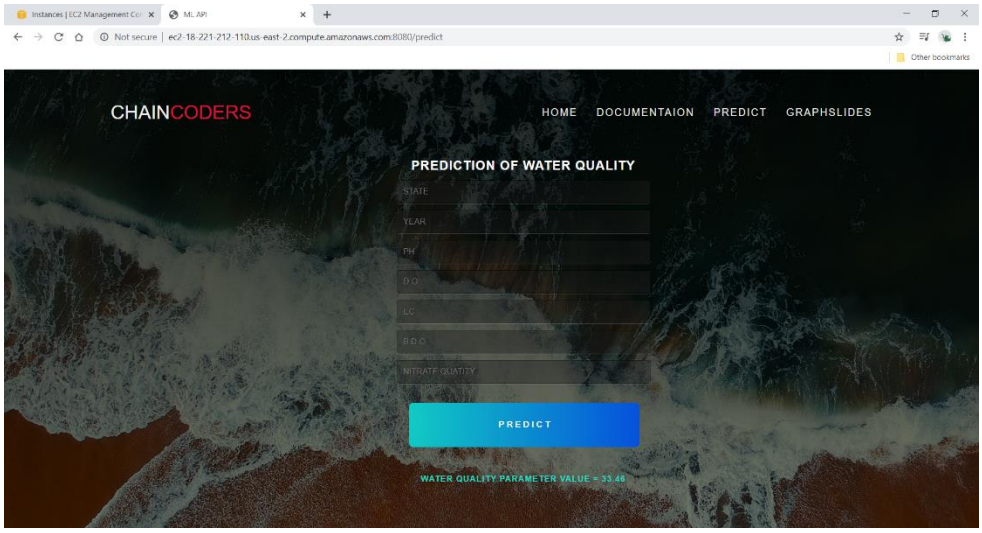
# CHAPTER: 8

# OUTPUT SCREENS

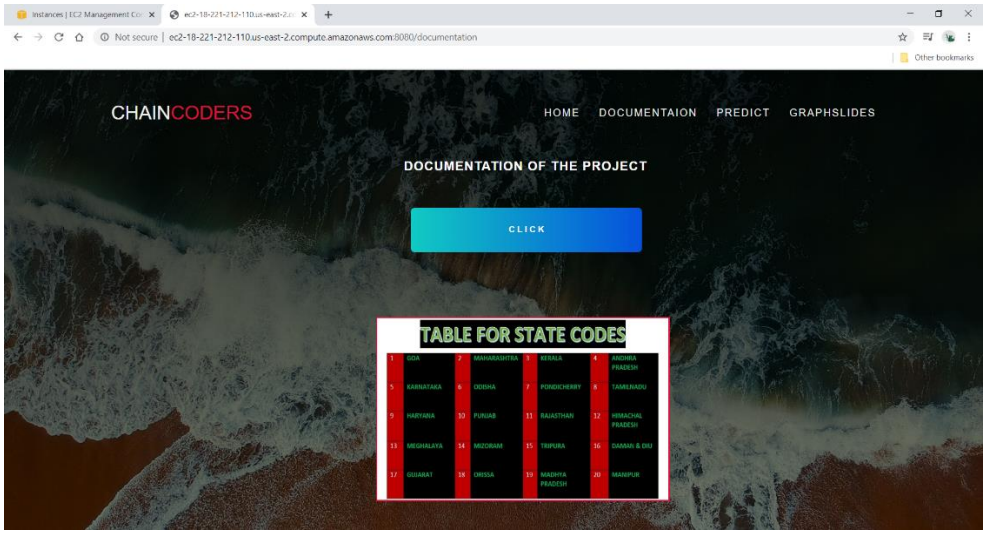
# OUTPUT SCREENS

## HOME/PREDICT PAGE:

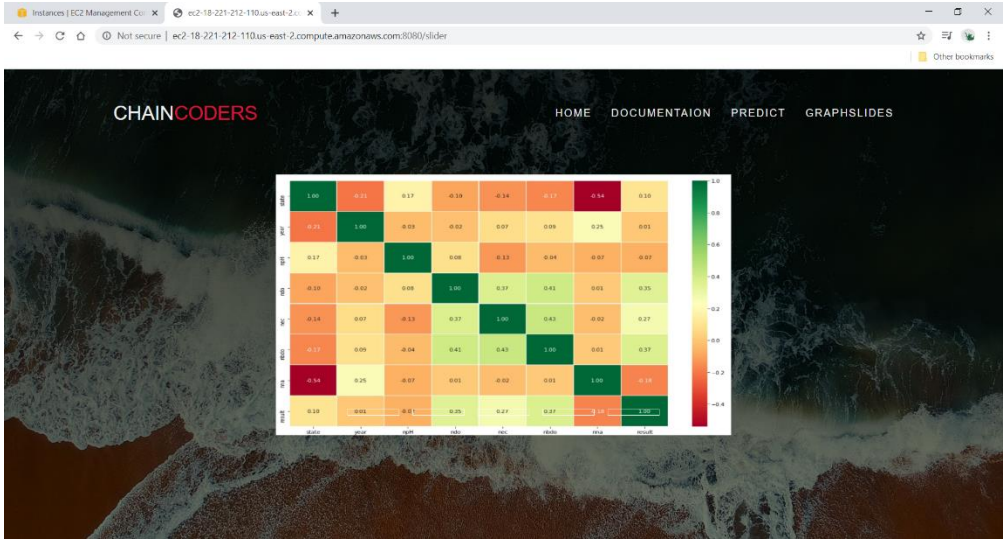



## DOCUMENTATION PAGE:



## GRAPHSLIDER PAGE:




## RANDOMFOREST REGRESSOR FIT AND PREDICT:

```
In [35]: #randomforest

In [36]: from sklearn.ensemble import RandomForestRegressor

In [37]: rfr=RandomForestRegressor(n_estimators=1000,random_state=42)

In [38]: rfr.fit(x_t,y_t)

Out[38]: RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                               max_depth=None, max_features='auto', max_leaf_nodes=None,
                               max_samples=None, min_impurity_decrease=0.0,
                               min_impurity_split=None, min_samples_leaf=1,
                               min_samples_split=2, min_weight_fraction_leaf=0.0,
                               n_estimators=1000, n_jobs=None, oob_score=False,
                               random_state=42, verbose=0, warm_start=False)

In [39]: yrfr=rfr.predict(x_tt)

In [40]: from sklearn.metrics import mean_squared_error
         print('mse:%.2f'%mean_squared_error(y_tt,yrfr))

         mse:180.78

In [41]: y_tt

Out[41]: 800     0
         814    40
         622    40
         676    60
         934    40
                ..
         663     0
         627    40
         477    80
         739    80
         410    40
         Name: nco, Length: 208, dtype: int64
```

# CHAPTER: 9

# CONCLUSION

# CONCLUSION

Accurate detection of water quality changes is a crucial task of water companies. Water supply companies must provide safe drinking water. Nowadays in different areas, we find sensible sensors which monitor data during the time. Normally the data registered by the sensors contain a meaning, such as there can be any event. Sometimes the data are ill-understood and stating if there is an event which is difficult. This work represents the description of several approaches to identifying changes or anomalies occurring on water quality time series data.

The performance of artificial intelligence techniques including Random Forest Regression evaluated to predict the water quality components of Rivers across India. To this end most dataset related well-known components, such as pH, SO4, Na, Ca, Cl, Mg, HCO3 etc., were collected. Results indicated that the applied models have suitable performance for predicting water quality components, however, the best performance was related to the Linear Regression as it showed the linearity. Among the basis of the parameters of water we are able to find out the relationships among them and to calculate and predict Water Quality Index (WQI).

This can help the Government bringing significant changes to India as water is wasted in many regions, states of India. The current example is Chennai where people hesitate drinking tap water and hence buy seal-packed water just to keep them sec

# CHAPTER: 10

# REFERENCES

# REFERENCES


- https://www.researchgate.net/publication/228474581_Effect_of_Human_Activity_on_Rivers
- https://unstats.un.org/unsd/environment/envpdf/pap_wasess5a2india.pdf
- https://www.kaggle.com/anbarivan/indian-water-quality-data
- https://www.youtube.com/channel/UCNU_lfiiWBdtULKOw6X0Dig
- Datasets of Previous Year: Kaggle: https://www.kaggle.com/venkatramakrishnan/india-water-quality-data
- Datasets : https://www.science.gov/topicpages/w/water+quality+datasets
- Datasets: http://data.europa.eu/euodp/data/dataset/data_waterbase-water-quality
- Datasets: https://catalog.data.gov/dataset?res_format=CSV&tags=water+quality
- https://www.c-sharpcorner.com/UploadFile/f55fd9/prediction-of-water-quality-in-river/
- https://www.researchgate.net/figure/Description-of-the-Water-Quality-Dataset_tbl2_287359516
- https://www.tandfonline.com/doi/full/10.1080/24751839.2019.1565653
- https://ir.unimas.my/id/eprint/12651/1/Predicting%20and%20Analyzing%20Water%20Quality%20using%20Mchine%20Learning%20(abstract).pdf
- https://www.researchgate.net/figure/Description-of-the-Water-Quality-Dataset_tbl2_287359516
- https://www.ijcaonline.org/archives/volume125/number9/22461-2015905874
- Datasets : https://www.kaggle.com/anbarivan/indian-water-quality-analysis-and-prediction/data
- https://data.gov.in/dataset-group-name/water-quality