PROJECT REPORT

EFFICIENT WATER QUALITY ANALYSIS AND PREDICTION USING MACHINE LEARNING

submitted by

PNT2022TMID52735

Rithika V A	-	CIT1907035
Shobika P	-	CIT1907046
Snekha C	-	CIT1907047
Divya Dharshini M	-	CIT2007203

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

a. PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

Water quality analysis is to predict that the water is safe to drink or not using some parameters like PH value, conductivity, hardness, etc. Various models such as XG Boost, Logistic regression, Support vector classifier, Random forest Classifier are used for training to improvise the efficiency.

b. PURPOSE

Water quality monitoring can help researchers predict and learnfrom natural processesin the environment and determine human impacts on an ecosystem. These measurement efforts can also assist in restoration projects or ensure environmental standards are being met. Requires to address the problem that is currently involved with the water quality. It will also ensure that the waterquality is protected from every potential cause of contamination and an appropriate approach is involved with the treatment system.

CHAPTER 2

LITERATURE SURVEY

a. EXISTING PROBLEM

Poor water quality can pose a health risk for people. Poor water quality can also pose a health risk for ecosystems. If drinking water contains unsafe levels of contaminants, it can cause health effects, such as gastrointestinal illnesses, nervous system or reproductive effects, and chronic diseases such as cancer. Assessment of water quality using conventional methods causes losses in economic value,which in turn affects the decision-making capacity for water quality management programs.

b. REFERENCES

Title: WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking andIrrigation Purposes(2022)

Author: antoine b. bagula, hloniphani c. maluleke, kevin c. pietersen

This study proposes a network architecture to collect data on water parameters in real-time and use Machine Learning (ML) tools to automatically determine suitability of water samples for drinking and irrigation purposes. The developed monitoring network is based on LoRa and takes the land topology into consideration. Results of simulations done in Radio Mobile revealed a partial mesh network topology as the most adequate. The primary requirement was to have data that could be used to train (and test) our ML models to classify water samples. After aggregation we ended up with two datasets containing approximately.

Title: Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception (2020)

Author: Di Wu, Hao Wang, Hadi Mohammed,Razak Seidu

The eventual aim of this work is to predict water quality risk. In order to find the risk model, we have investigated with researchers from water quality control. The models need to be evolved with both domain knowledge data set growing. The scalability of our method can serve as a very powerful tool for practical water quality early warning. The data we collected for this application is from several industrial drinking water supply systems in Norway. Some of the physical and chemical indicators from A lesund were only recorded 25 times for 11 years; alkalinities all equal to zero; values for Ecoli are over 95 percent zero.

Title: Empirical Comparison of Approaches for Mitigating Effects of Class Imbalances in Water Quality Anomaly Detection.

Author: Eustace M. Dogo, Nnamdii. Nwulu, Bhekisipho twala, Clinton ohis aigbavboa.

We conduct an exploratory study to compare the performance of selectMV, data-level ICD and ensemble approaches previously proposed in the literature on different classifiers. The empirical evidence based on the argument that a combination of missing value and resampling methods can improve them performance of classifiers. The dataset used in all our experiments is obtained from GECCO 2018 industrial challenge project, sourcedfrom public water utility company located in Germany. The dataset is a time series based and made up of ten independent variables, and one dependant variable. The goal of this dataset is a classification problem intended for drinking-water quality anomalydetection, to predict if there is an event or not.

Title: Connected Sensors, Innovative Sensor Deployment, and Intelligent Data Analysis for Online Water QualityMonitoring (2019)

Author: Libu Manjakkal, Srinjoy Mitra , Yvan R. Petillot, Jamie Shutler.

This paper presentsa comprehensive review of the sensors, deployment, and analysis technologies for WQM. A network of networked water bodies could enhance the global data

inter comparability and enable WQM at a global scaleto address globalchallenges related to food, drinking water, and health. A large number of PCB parameters that need to be monitored to ascertain the WQ are summarized. As a result, robust strategieshave been sought from time to time to bridge the knowledge gaps and to generate reliableestimates to developappropriate mitigation measures. In this regard, the different methods for the deployment of autonomous sensors have been explored along with development of suitable interface electronics for real-time data transmission and communication. There are many purposes labeled under tasks of operation and surveillance, including monitoring to report on status.

Title: Monitoring Water Quality Parametersof Taihu Lake Based on Remote SensingImages and LSTM-RNN(2019)

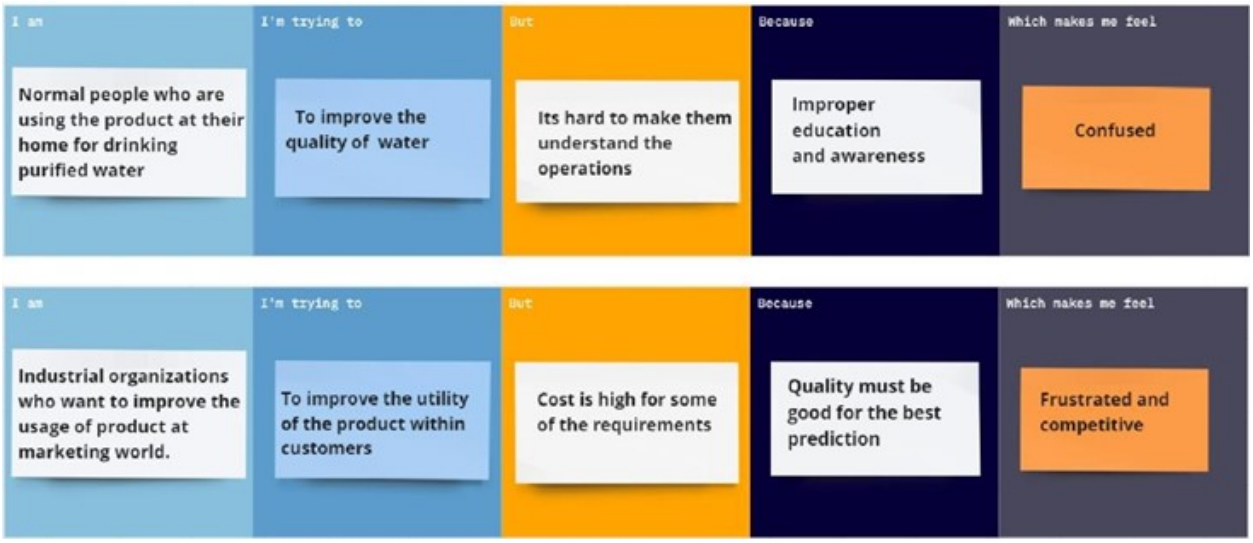
Author: CHUHAN QI, SHUOHUANG, AND XIAOFEIWANG

Long-term dynamic monitoring of water quality has critical and far-reaching significant for human social life, industrial production and agricultural irrigation. In the research of water quality monitoring, the combination of satellite remote sensing images and machine learning has become afocus of attention. The LSTM networkmodel is chosen to retrievewater quality parameters and the results are highly accurate, the model is robust and performs well, and has the abilityto meet the basic requirements of actual water qualitymonitoring.

C. PROBLEM STATEMENT DEFINITION

Safe and readily available water is important for public health, whether it is used for drinking, domestic use, food production or recreational purposes. Better water supplies and sanitation, as well as better management of water resources, can contribute greatly to poverty reduction and economic growth. It is known that contaminated water and inadequate sanitation facilitate the transmission of diseases such as cholera, diarehoea, dysentery, hepatitis A, typhoid, and polio. Those without access to clean water and sanitation face preventable ones leads to health risks. The proposed model will predict that the water is safe to drink or not using some parameters like PH value, conductivity, hardness, etc. Various models such as XG Boost, Logistic regression,

Support vector classifier, Random forest Classifier are used for trainingto improvise the efficiency.

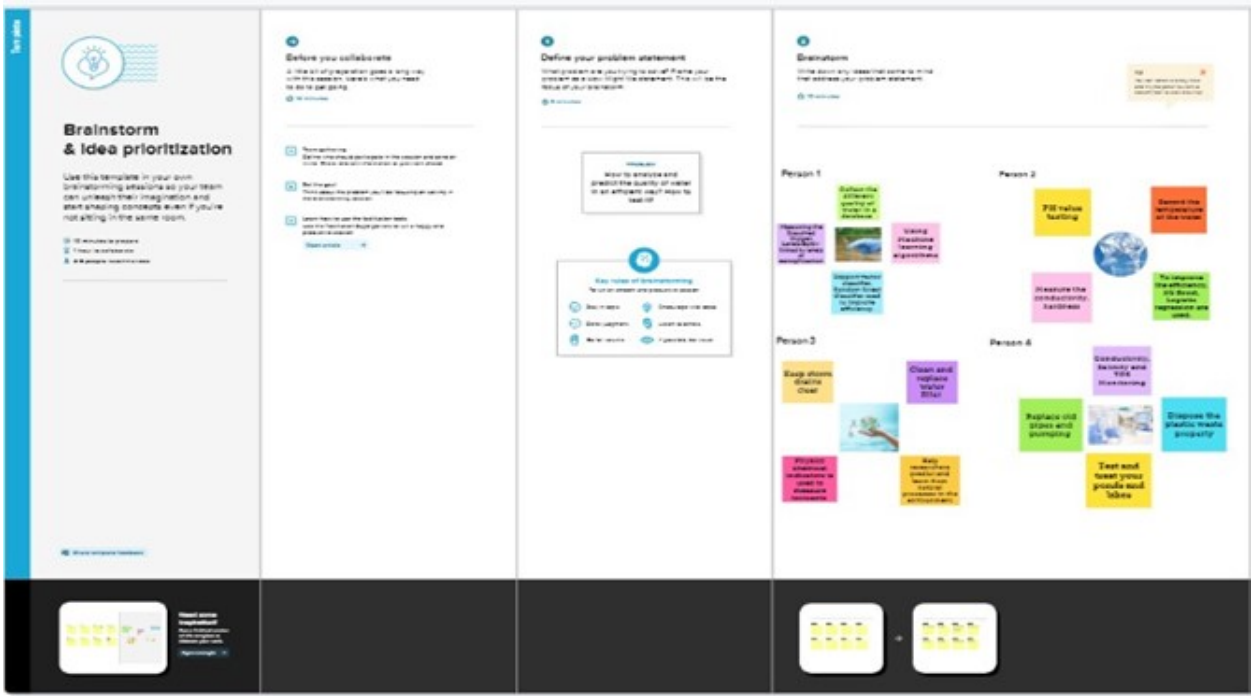


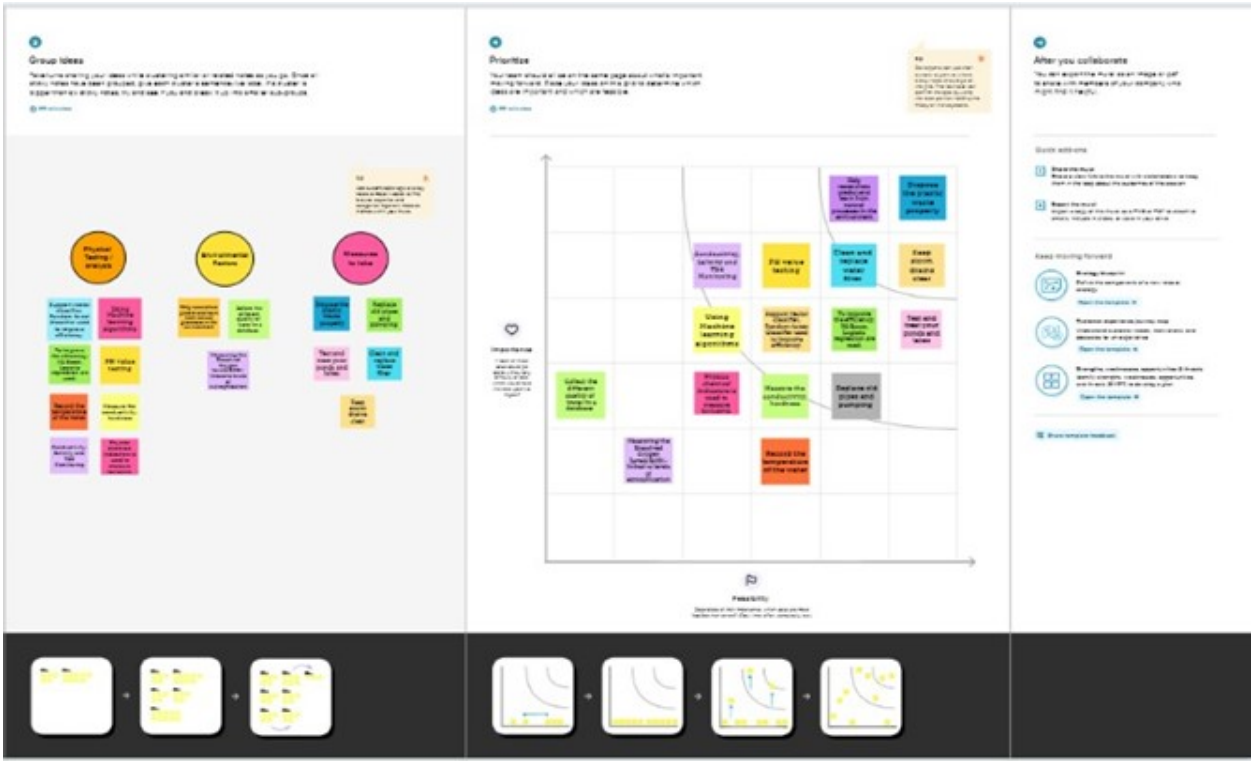
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

a. EMPATHY MAP CANVAS

b. IDEATION &BRAINSTORMING



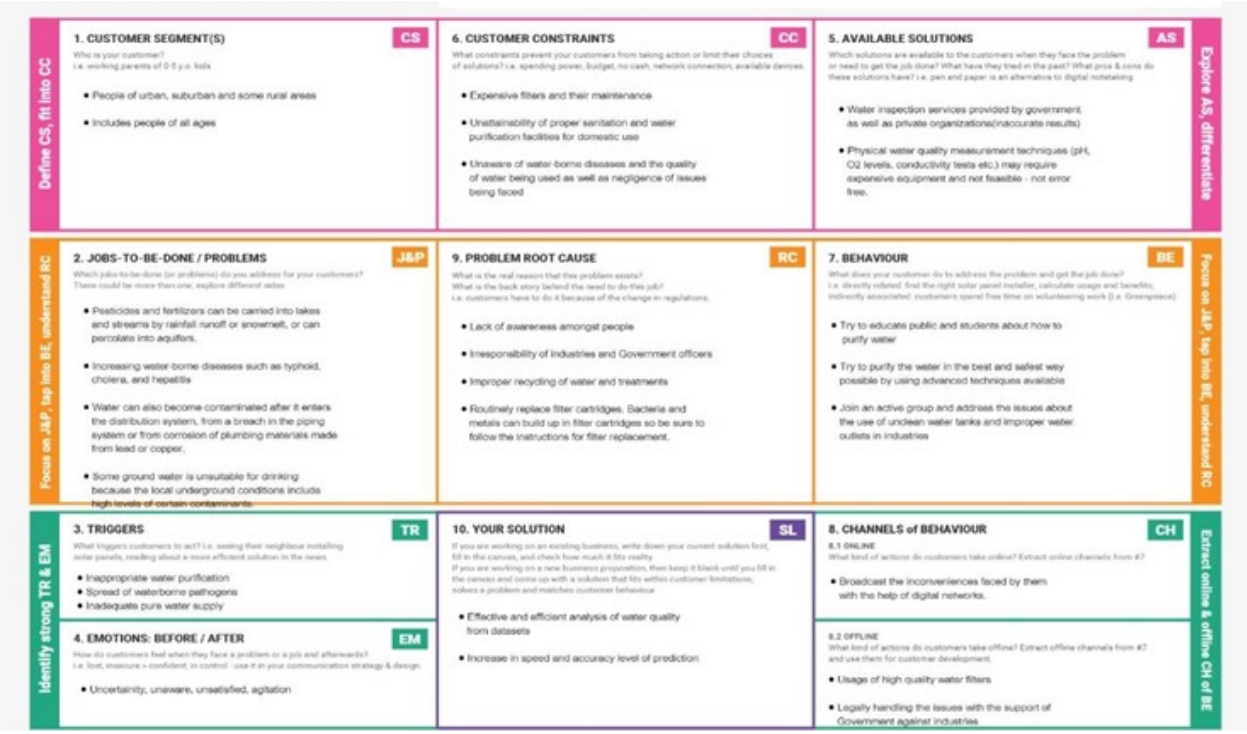


c. PROPOSED SOLUTION

S. No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Safe and readily available water is important for public health, whether it is used for drinking, domestic use, food production or recreational purposes. Better water supplies and sanitation, as well as bettermanagement of waterresources, can contribute greatly to poverty reduction and economic growth. It is known that contaminated water and inadequate sanitation facilitate the transmission of diseases such as cholera, dysentery, hepatitis A, typhoid, and polio. Thosewithout access to clean water and sanitation face preventable onesleads to health risks.
2	Idea / Solution description	The proposed model will predict that the water is safe to drink or not using some parameters like PH value,conductivity, hardness, etc. Various modelssuch as XG Boost, Logistic regression, Support vector classifier, Random forest Classifier are used for training to improvise the efficiency.
3	Novelty / Uniqueness	To understand what constitutes safe, potable waterand distinguish between potable and non-potable water by applying machine learning techniques.
4	Social Impact / Customer Satisfaction	Water quality monitoring can help researchers predict and learn from natural processes in the environment and determine human impacts on an ecosystem. Poor water quality can pose a health risk for people. Poor water quality can also pose a health risk for ecosystems. If drinking watercontains unsafe levelsof contaminants, it can cause health effects, such as gastrointestinal illnesses, nervous system or reproductive effects, and chronic diseases such as cancer.Thus Water qualitytesting is an important part of environmental monitoring that helps us to lead a healthy life.

5	Business Model (Revenue Model)	Assessment of water quality using conventional methods causes losses in economic value, which in turn affects the decision- making capacity for water qualitymanagement programs. Therefore, to tackle theseissues, it is essential to adopt a potential and cost-efficient approach for quick and accurate assessment of water quality. In our project, the application of machine learning (ML) techniques can be an effective and reliable approach for the evaluation of water quality.
6	Scalability of theSolution	Objective weighting system-based approaches are more reliable because they consider local variations in a dataset during the computation process. This ML technique is anextension of the artificial neuralnetwork method; it has additional complex architectures that make this approach suitable for managing multi-dimensional inputs because of its high model configuration flexibility, greater generalization power, and robust learning capacity.

d. PROBLEM SOLUTION FIT



CHAPTER 4
REQUIREMENT ANALYSIS

a. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration throughGmailRegistration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Executive administration	Regulation of monitoring the water environment status and regulatory compliance like pollution event emergency management and it includes two different functions: earlywarning/forecast monitoring.
FR-4	Data handling	File contains water quality metrics for differentwater bodies.
FR-5	Quality analysis	Analyze with the acquired information of the water across various water quality indicator like (PH, Turbidity, TDS, Temperature) usingdifferent models
FR-6	Model prediction	Confirming based on water qualityindex and showsthe machine learning prediction (Good, Partially Good, Poor) with the percentage of presence of various parameter

FR-7	Remote Visualization	Visualization through chartsbased on presentand past values of all the parameter for future forecast.
FR-8	Notification services	Confirming through notification of water status prediction with parameter presence along with timestamp.

b. NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system provides a natural interaction with the users. Accurate water quality prediction with short timeanalysis and provide prediction safe to drink or not usingsome parameters and provide a great significance for water environment protection.
NFR-2	Security	The model enableswith the high security system as the user's data will not be shared to the other sources. The system is protected with the user name and password throughout the process.
NFR-3	Reliability	The system is very reliable as it can last for long periodof time when it is well maintained. The model can be extended in large scale by increasing the datasets.

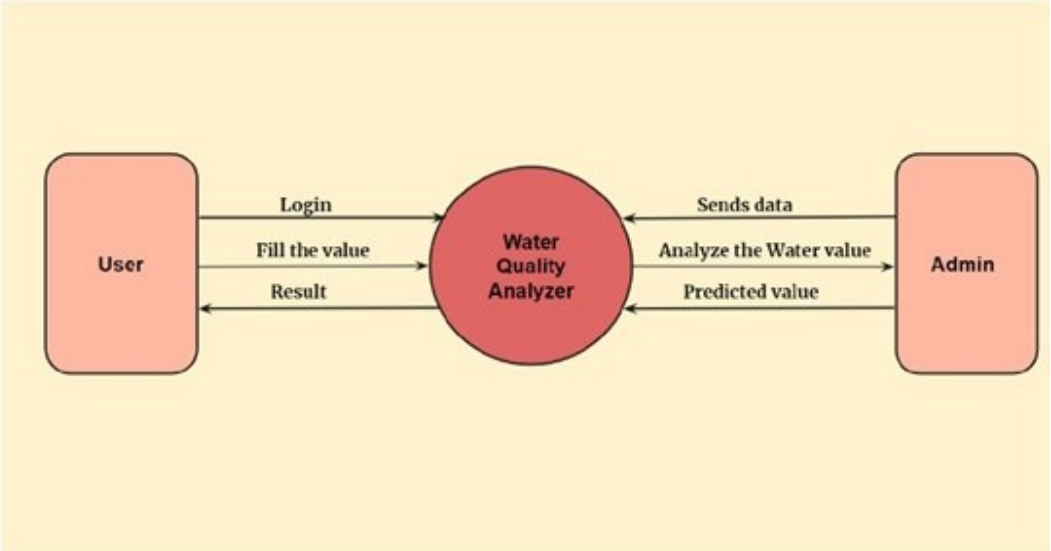
NFR-4	Performance	Our system should run on 32 bit (x86) or 64 bit (x64) Dual -core 2.66 -GHZ or faster processor. It should not exceed 2 GBRAM.
NFR-5	Availability	The system should be available for the duration of the user access the system untilthe user terminate the access. The system response to request of the user in less time and the recovery is doneis less time.
NFR-6	Scalability	It provides an efficient outcome and has the ability to increase or decrease the performance of the system based on the datasets.

CHAPTER 5

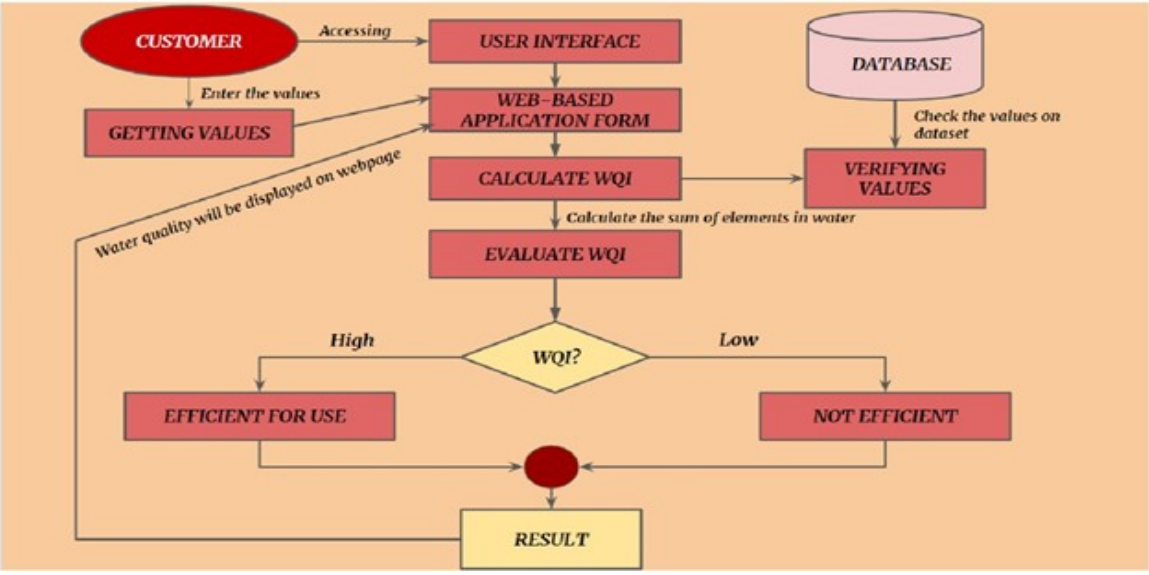
PROJECT DESIGN

a. DATA FLOW DIAGRAM

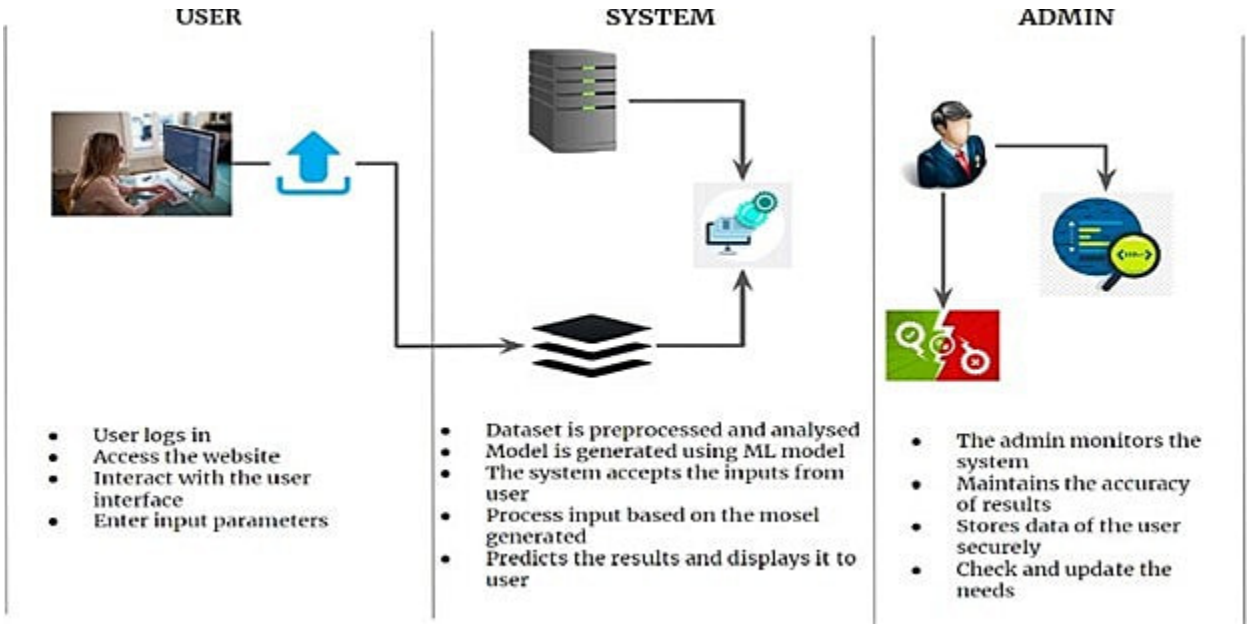
DFD LEVEL 0

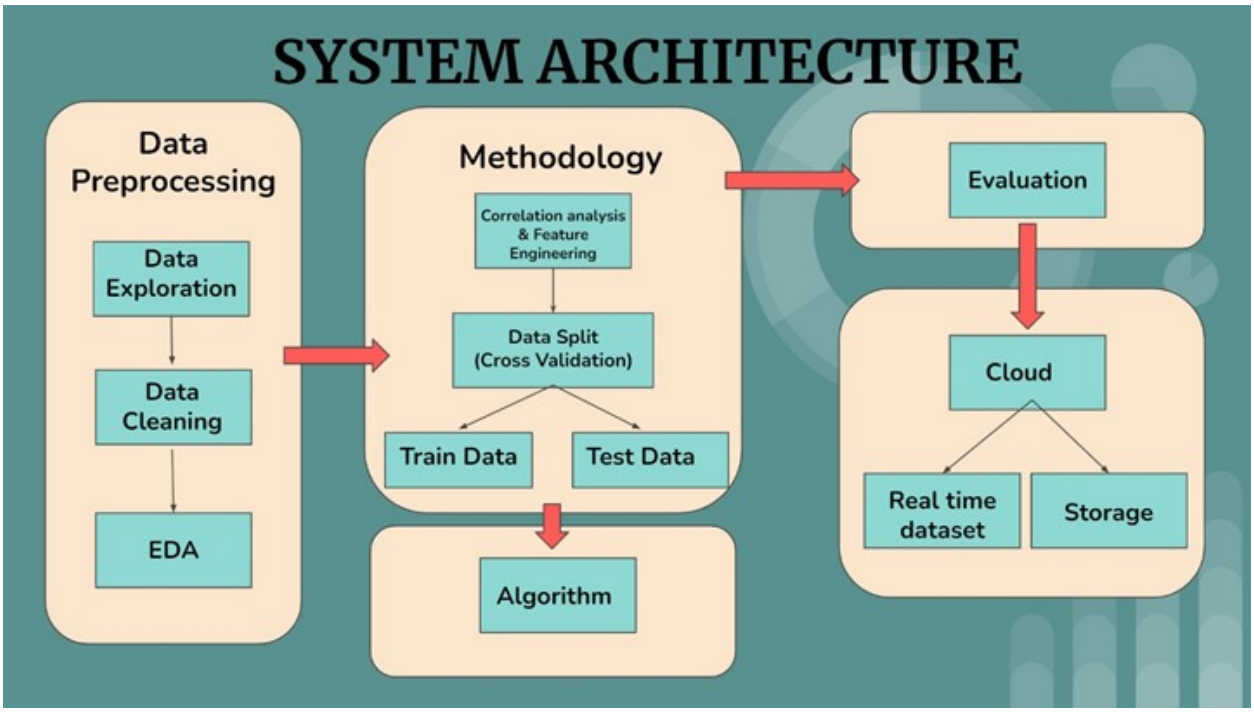


Data Flow Diagrams: DFD LEVEL 1



b. SOLUTION & TECHNICAL ARCHITECTURE





c. USER STORIES

User Type	Functional Requirements	User Story Number	User Story / Task	Acceptace Criteria	Priority	Release
Customer (web user)	Access Webpage	USN-1	As a user, anyone can access the web page to check the waterquality .	I can access my webpage through online at any time	High	Sprint-1
Customer	Usage of water	USN-2	As per the usage of user, the quality of water should be predicted in easy way	Prediction can be done in easy way	High	Sprint-2
Customer	Accuracy of water	USN-3	By using the prediction model the user will know the quality of water ona daily basis	The qualityanalysis of waterwill be accurate	High	Sprint-3
Administrator	Manage the web page	USN-4	As an admin, he/she can manage user details andupdate parameters essential forprediction	Make changeson User Interface (UI)	High	Sprint-3
Administrator	Calculation ofWQI	USN-5	As an admin,he/she can update the calculations forwater quality index calculation	Improves the accuracy ofthe calculation	High	Sprint-3

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

a. SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1,2	Collecting/ downloading dataset for preprocessing.	12	High	Rithika VA DivyaDharshini M
Sprint-1	Data Pre processing	USN-2	Formats the data and handles the missing data in the dataset.	8	Medium	Snekha C Shobika P
Sprint-2	Model Building	USN-1,2	Calculate the Water Quality Index (WQI) using specified formula for every parameter.	10	High	Divya DharshiniM, Snekha C
Sprint-2	Accessing datasets	USN-1,2	Splitting the data into training and testing dataset from the entire dataset	10	High	RithikaVA, Shobika P

Sprint-3	Training andTesting	USN-1,2	Training the model using Random Forest Regression algorithm and testing the performance of the model (accuracy rate)	20	High	Snekha C, Shobika P
Sprint-4	Implementation of Web page and user login	USN-1,2	Implementing the web page for collecting the data from user	12	High	Rithika VA, Divya Dharshini M

b. SPRINT DELIVERY SCHEDULE

Sprint	**Total StoryPoints**	**Duration**	**SprintStart Date**	**Sprint End Date(Planned)**	**Story PointsCompleted (as on Planned End Date)**	**Sprint ReleaseDate (Actual)**
Sprint-1	20	6 ays	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

Importing the libraries

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

Reading Dataset

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

Analysis of the data

Using head() we find the top 5 records of the dataset to understand about the parameters

In [3]:
```
data.head()
```

Out[3]:

	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

Descriptive Statistics Of The Dataset

In [4]:
```
data.describe()
```

Out[4]:

	year
count	1991.000000
mean	2010.038172
std	3.057333
min	2003.000000
25%	2008.000000
50%	2011.000000
75%	2013.000000
max	2014.000000

The Above Description shows only the Descriptive Statistics of the year column beacuse it is the only column which has the datatype 'int64'

info() method is used to get the columns in the dataset along with their properties

In [5]:
```
data.info()
```

```
RangeIndex: 1991 entries, 0 to 1990
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   STATION CODE                    1991 non-null   object
 1   LOCATIONS                       1991 non-null   object
 2   STATE                           1991 non-null   object
 3   Temp                            1991 non-null   object
 4   D.O. (mg/l)                     1991 non-null   object
 5   PH                              1991 non-null   object
 6   CONDUCTIVITY (µmhos/cm)         1991 non-null   object
 7   B.O.D. (mg/l)                   1991 non-null   object
 8   NITRATENAN N+ NITRITENANN (mg/l) 1991 non-null   object
 9   FECAL COLIFORM (MPN/100ml)      1991 non-null   object
 10  TOTAL COLIFORM (MPN/100ml)Mean  1991 non-null   object
 11  year                            1991 non-null   int64
dtypes: int64(1), object(11)
memory usage: 186.8+ KB
```

In [6]:
```
data.shape
```

Out[6]:
```
(1991, 12)
```

Exploratory Data Analysis

Handling Missing Values

In [7]:
```
data.isnull().any()
```

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

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
STATION CODE                        0
LOCATIONS                           0
STATE                               0
Temp                                0
D.O. (mg/l)                         0
PH                                  0
CONDUCTIVITY (µmhos/cm)             0
B.O.D. (mg/l)                       0
NITRATENAN N+ NITRITENANN (mg/l)    0
FECAL COLIFORM (MPN/100ml)          0
TOTAL COLIFORM (MPN/100ml)Mean      0
year                                0
dtype: int64
```

There is no null values present in any of the columns of the used dataset

In [9]:
```
data.dtypes
```

Out[9]:
```
STATION CODE                        object
LOCATIONS                           object
STATE                               object
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
```

Most of the numeric columns has the data type of object. So let's convert those columns into float datatype.

In [10]:
```
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[10]:
```
STATION CODE                         object
LOCATIONS                            object
STATE                                object
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
```

Check For The Missing Values Again.

Water Quality Index Calculation

Calculation of pH (in range of 1 to 100)

In [16]:
```
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)))))
```

Calculation of Dissolved Oxygen(in range of 1 to 100)

In [17]:
```
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)))))
```

Calculation of Total Coliform(in range of 1 to 100)

In [18]:
```
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)))))
```

Calculation of B.O.D - Biological Oxygen Demand (in range of 1 to 100)

In [19]:
```
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)))))
```

Calculation of Electrical Conductivity (in range of 1 to 100)

In [20]:
```
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 [21]:
```
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)))))
```

In [22]:
```
data.head()
```

Out[22]:

	station	location	state	Temp	do	ph	co	bod	na	tc	year	npH	ndo	nco	nbdo	nec	nna
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049	0.1	27.0	2014	100	100	80	60	60	100
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000	0.2	8391.0	2014	100	80	40	100	60	100
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179.0	1.700000	0.1	5330.0	2014	80	100	40	100	60	100
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64.0	3.800000	0.5	8443.0	2014	80	80	40	80	100	100
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83.0	1.900000	0.4	5500.0	2014	100	80	40	100	80	100

In [23]:
```
data.dtypes
```

Out[23]:
```
station      object
location     object
state        object
Temp        float64
do          float64
ph          float64
co          float64
bod         float64
na          float64
tc          float64
year          int64
npH           int64
ndo           int64
nco           int64
nbdo          int64
nec           int64
nna           int64
dtype: object
```

Calculation of Water Quality Index Value

In [24]:
```
data['wph']=data.npH * 0.165
data['wdo']=data.ndo * 0.281
data['wbdo']=data.nbdo * 0.234
data['wec']=data.nec* 0.009
data['wna']=data.nna * 0.028
data['wco']=data.nco * 0.281
data['wqi']=data.wph+data.wdo+data.wbdo+data.wec+data.wna+data.wco
```

In [25]:
```
data
```

Out[25]:

	station	location	state	Temp	do	ph	co	bod	na	tc	...	nbdo	nec	nna	wph	wdo	wbdo	wec	wna	wco	wqi
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.600000	6.7	7.5	203.0	6.940049	0.100000	27.0	...	60	60	100	16.5	28.10	14.04	0.54	2.8	22.48	84.46
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.800000	5.7	7.2	189.0	2.000000	0.200000	8391.0	...	100	60	100	16.5	22.48	23.40	0.54	2.8	11.24	76.96
2	1475	ZUARI AT PANCHAWADI	GOA	29.500000	6.3	6.9	179.0	1.700000	0.100000	5330.0	...	100	60	100	13.2	28.10	23.40	0.54	2.8	11.24	79.28
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.700000	5.8	6.9	64.0	3.800000	0.500000	8443.0	...	80	100	100	13.2	22.48	18.72	0.90	2.8	11.24	69.34
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.500000	5.8	7.3	83.0	1.900000	0.400000	5500.0	...	100	80	100	16.5	22.48	23.40	0.72	2.8	11.24	77.14
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1986	1330	TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU	NAN	26.209814	7.9	738.0	7.2	2.700000	0.518000	202.0	...	100	100	100	0.0	28.10	23.40	0.90	2.8	16.86	72.06
1987	1450	PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, T...	NAN	29.000000	7.5	585.0	6.3	2.600000	0.155000	315.0	...	100	100	100	0.0	28.10	23.40	0.90	2.8	16.86	72.06
1988	1403	GUMTI AT U/S SOUTH TRIPURA,TRIPURA	NAN	28.000000	7.6	98.0	6.2	1.200000	1.623079	570.0	...	100	100	100	0.0	28.10	23.40	0.90	2.8	11.24	66.44
1989	1404	GUMTI AT D/S SOUTH TRIPURA, TRIPURA	NAN	28.000000	7.7	91.0	6.5	1.300000	1.623079	562.0	...	100	100	100	0.0	28.10	23.40	0.90	2.8	11.24	66.44
1990	1726	CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA	NAN	29.000000	7.6	110.0	5.7	1.100000	1.623079	546.0	...	100	100	100	0.0	28.10	23.40	0.90	2.8	11.24	66.44

1991 rows × 24 columns

Calculating Overall Water Quality Index for Each Year

In [27]:
```
average=data.groupby('year')['wqi'].mean()
```

In [28]:
```
average
```

Out[28]:
```
year
2003    66.239545
2004    61.290000
2005    73.762689
2006    72.909714
2007    74.233000
2008    73.506289
2009    74.456685
2010    75.454787
2011    76.666580
2012    79.496712
2013    76.494789
2014    77.839592
Name: wqi, dtype: float64
```

In [29]:
```
datel=average.reset_index(level=0,inplace=False)
datel
```

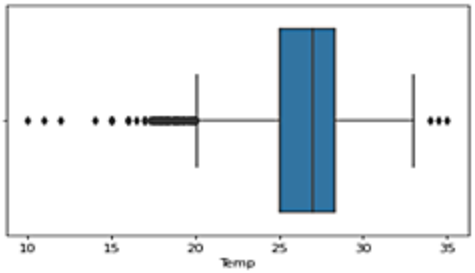
Out[29]:

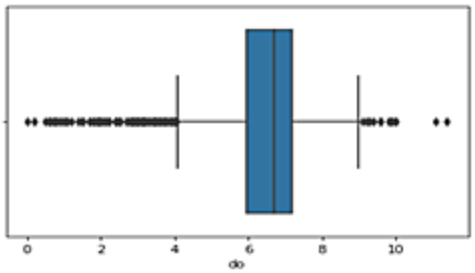
	year	wqi
0	2003	66.239545
1	2004	61.290000
2	2005	73.762689
3	2006	72.909714
4	2007	74.233000
5	2008	73.506289
6	2009	74.456685

Data Visualizations

1. Univariate Analysis

In [30]:
```
for col in data.columns:
    if data.dtypes[col]=='float64'or data.dtypes[col]=='int64':
        sns.boxplot(x=data[col]).set(xlabel=col)
        plt.show()
```

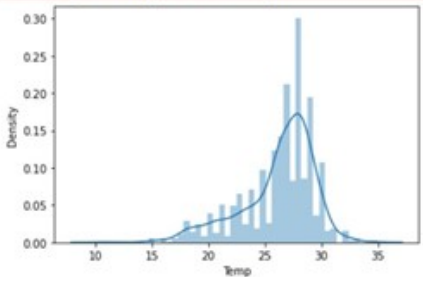




In [31]:

```
for col in data.columns:
    if data.dtypes[col]=='float64':
        sns.distplot(x=data[col]).set(xlabel=col)
        plt.show()
```

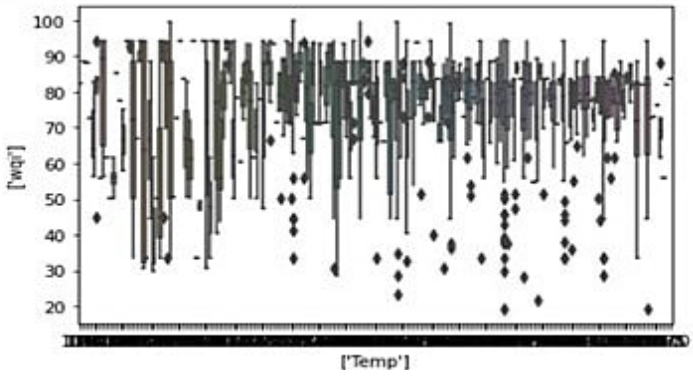
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```



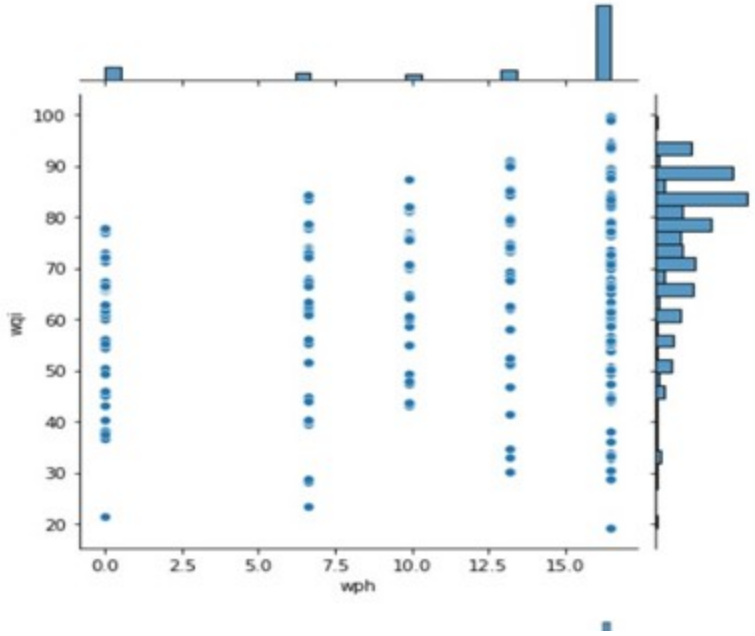
2. Bivariate Analysis

[32]:

```
for col1 in data.columns:
    if data.dtypes[col1]=='float64':
        for col2 in ['wqi']:
            if data.dtypes[col2]=='float64':
                sns.boxplot(x=data[col1],y=data[col2]).set(xlabel=[col1],ylabel=[col2])
                plt.show()
```



In [33]:
```
for col1 in ['wph','wdo','wbdo','wec','wna','wco']:
    for col2 in ['wqi']:
        sns.jointplot(x=data[col1],y=data[col2])
        plt.show()
```



In [35]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[35]:

Removing Unnecessary Columns in the Dataset which is used to calculate the Water Quality Index (WQI)

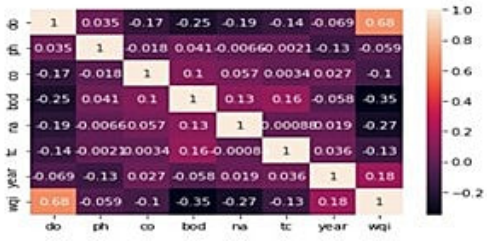
In [36]:
```
data.drop(['Temp','station','location','state','nbdo',"nec","nna","wph","wdo","wbdo","wec","wna","wco","npH","ndo","nco"],axis = 1,inplace=True)
data.head()
```

Out[36]:

	do	ph	co	bod	na	tc	year	wqi
0	6.7	7.5	203.0	6.940049	0.1	27.0	2014	84.46
1	5.7	7.2	189.0	2.000000	0.2	8391.0	2014	76.96
2	6.3	6.9	179.0	1.700000	0.1	5330.0	2014	79.28
3	5.8	6.9	64.0	3.800000	0.5	8443.0	2014	69.34
4	5.8	7.3	83.0	1.900000	0.4	5500.0	2014	77.14

In [38]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[38]:



Splitting Dataset into Dependent and Independent Columns

In [39]:
```
x=data.iloc[:,0:7].values
y=data.iloc[:,7:].values
```

In [40]:
```
x.shape
```

Out[40]: (1991, 7)

In [41]:
```
y.shape
```

Out[41]: (1991, 1)

In [42]:
```
x
```

Out[42]:
```
array([[6.70000000e+00, 7.50000000e+00, 2.03000000e+02, ...,
        1.00000000e-01, 2.70000000e+01, 2.01400000e+03],
       [5.70000000e+00, 7.20000000e+00, 1.89000000e+02, ...,
        2.00000000e-01, 8.39100000e+03, 2.01400000e+03],
       [6.30000000e+00, 6.90000000e+00, 1.79000000e+02, ...,
        1.00000000e-01, 5.33000000e+03, 2.01400000e+03],
       ...,
       [7.60000000e+00, 9.80000000e+01, 6.20000000e+00, ...,
        1.62307871e+00, 5.70000000e+02, 2.00300000e+03],
       [7.70000000e+00, 9.10000000e+01, 6.50000000e+00, ...,
        1.62307871e+00, 5.62000000e+02, 2.00300000e+03],
       [7.60000000e+00, 1.10000000e+02, 5.70000000e+00, ...,
        1.62307871e+00, 5.46000000e+02, 2.00300000e+03]])
```

In [43]:
```
y
```

Out[43]:
```
array([[84.46],
       [76.96],
       [79.28],
       ...,
       [66.44],
       [66.44],
       [66.44]])
```

Splitting Dataset into Training and Testing

In [44]:
```
from sklearn.model_selection import train_test_split
```

In [45]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=10)
```

In [46]:
```
x_train.shape
```

Out[46]: (1592, 7)

In [47]:
```
x_test.shape
```

Out[47]: (399, 7)

In [48]:
```
y_train.shape
```

Out[48]: (1592, 1)

In [49]:
```
y_test.shape
```

Out[49]: (399, 1)

Usage of Random forest Regression algorithm

In [50]:
```
#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [51]:
```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Pleas
e change the shape of y to (n_samples,), for example using ravel().
  This is separate from the ipykernel package so we can avoid doing imports until
```

Model Evaluation

In [52]:
```
from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(y_test,y_pred))
print('MSE:',metrics.mean_squared_error(y_test,y_pred))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
MAE: 1.013774436090232
MSE: 6.240685845864675
RMSE: 2.498136472370248
```

In [53]:
```
#accuracy of the model
metrics.r2_score(y_test, y_pred)
```

Out[53]: 0.9659820315121997

CHAPTER 8

TESTING

a. TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status
Home Page_TC_OO1	Functional	Home Page	Verify user is able to see the dashboard of the webpage	1.to ensure that user can able to see information about water quality by clicking info 2.Verify the prediction button to analyse the quality a.Info button b.Predict button	_	Home and the buttons should be displayed	Working as expected	Pass
info page_TC_OO2	Functional	Info Page	Verify user is able to see the information of the webpage	1.to ensure that user can able to see information about water quality by clicking info 2.Verify Register/Signup page accepts only unique email a.Info button b.Predict button	_	Information should be displayed to the user	Working as expected	Pass

Prediction Page_TC_OO5	UI	prediction page	Verify user is able to see the description and predict button	1.Enter URL and click go 2.Click on predic button 3.Enter D.O,Ph,conductivity in the text box 4.Enter nitratenen and give the total coliform to predict the water level 5.Click on predict button	_	Application should show correct prediction level of the water and user should ensure the quality by referring the given parameter table.	Working as expected	Pass
Prediction Page_TC_OO5	Functional	prediction page	Verify user is able to predict by giving letters	1.Enter URL and click go 2.Click on predic button 3.Enter D.O,Ph,conductivity in the text box 4.Enter nitratenen as a letter 5.Click on predict button	_	Application should not show any value because user entered letter	Working as expected	Pass
Prediction Page_TC_OO5	Functional	prediction page	Verify the parameter table is listed below	1.parameters should be displayed by the result of water quality 2.displayed range should be shown in table	_	Application should show range of the result predicted by the webpage	Working as expected	Pass
Prediction Page_TC_OO5	Functional	prediction page	Verify user can leave any field unfilled	1.Enter URL and click go 2.Click on predic button 3.Enter D.O,Ph,conductivity in the text box 4.Leave nitratenen field as empty 5.Click on predict button	_	Application should not show any value because user left a field as empty	Working as expected	Pass

b. USER ACCEPTANCE TESTING

i. DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	8	2	4	10	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	5	2	1	8
Totals	19	24	17	16	58

ii. TEST CASE ANALYSIS

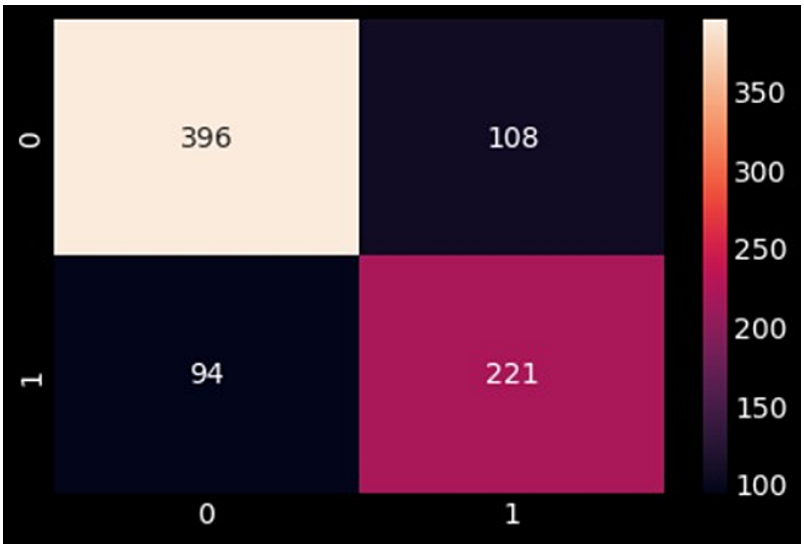
Section	Total Cases	Not Tested	Fail	Pass
Home Page	7	0	0	7
Client Application	51	0	0	51
Prediction	2	0	0	2
Pop ups	3	0	0	3
URL port	9	0	0	9
Final Report Output	4	0	0	4
Redirection	2	0	0	2

CHAPTER 9
RESULTS

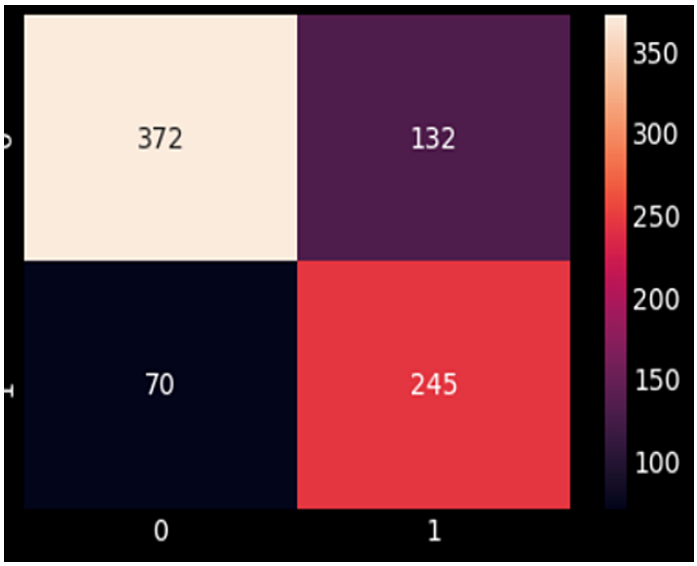
a. PERFORMANCE METRICS

i. Confusion Matrix:

RANDOM FOREST CLASSIFIER:



XGB CLASSIFIER:



ii. Precision, Recall Specificity,F1 Score :

RANDOM FOREST CLASSIFIER:

```
print('Random Forest Classifier\n')
Rfc = RandomForestClassifier()
Rfc.fit(X_train, y_train)

y_Rfc = Rfc.predict(X_test)
print(metrics.classification_report(y_test, y_Rfc))
print(modelAccuracy.append(metrics.accuracy_score(y_test, y_Rfc)))

sns.heatmap(confusion_matrix(y_test, y_Rfc), annot=True, fmt='d')
plt.show()
```

```
Random Forest Classifier

              precision    recall  f1-score   support

           0       0.81      0.79      0.80       504
           1       0.67      0.70      0.69       315

    accuracy                           0.75       819
   macro avg       0.74      0.74      0.74       819
weighted avg       0.76      0.75      0.75       819
```

XGB CLASSIFIER:

```
print('XGB Classifier\n')
xgb = XGBClassifier()
xgb.fit(X_train, y_train)

y_xgb = xgb.predict(X_test)
print(metrics.classification_report(y_test, y_xgb))
print(modelAccuracy.append(metrics.accuracy_score(y_test, y_xgb))

sns.heatmap(confusion_matrix(y_test, y_xgb), annot=True, fmt='d'
plt.show()
```

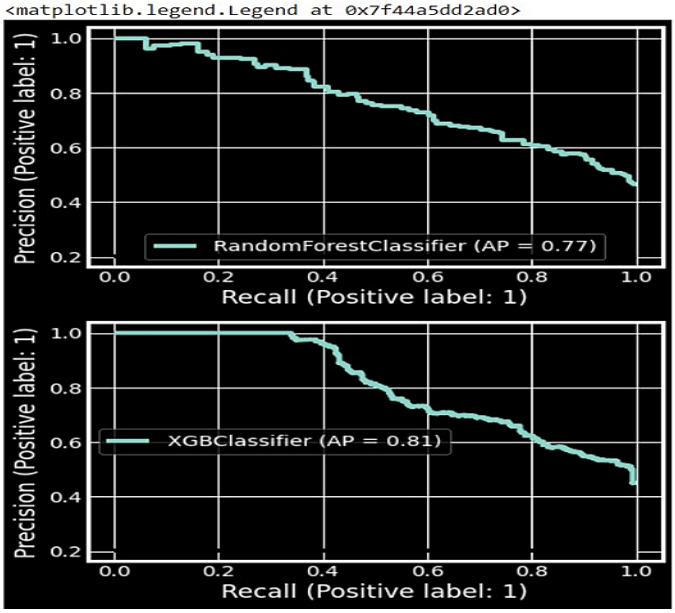
```
XGB Classifier

              precision    recall  f1-score   support

           0       0.84      0.74      0.79       504
           1       0.65      0.78      0.71       315

    accuracy                           0.75       819
   macro avg       0.75      0.76      0.75       819
weighted avg       0.77      0.75      0.76       819
```

iii. PRECISION – RECALL OR PRCURVE:

PRECISION RECALL CURVE - RANDOM FOREST,XGB Classifier

```
[95] from scikitplot.metrics import plot_roc_curve
     from sklearn.metrics import plot_precision_recall_curve
     plot_precision_recall_curve(Rfc,X_test,y_test)
     plt.plot([0,1], [0.2035,0.2035], c='k')
     plt.legend(loc='best')
     plot_precision_recall_curve(xgb,X_test,y_test)
     plt.plot([0,1], [0.2035,0.2035], c='k')
     plt.legend(loc='best')
```

```
<matplotlib.legend.Legend at 0x7f44a5dd2ad0>
```

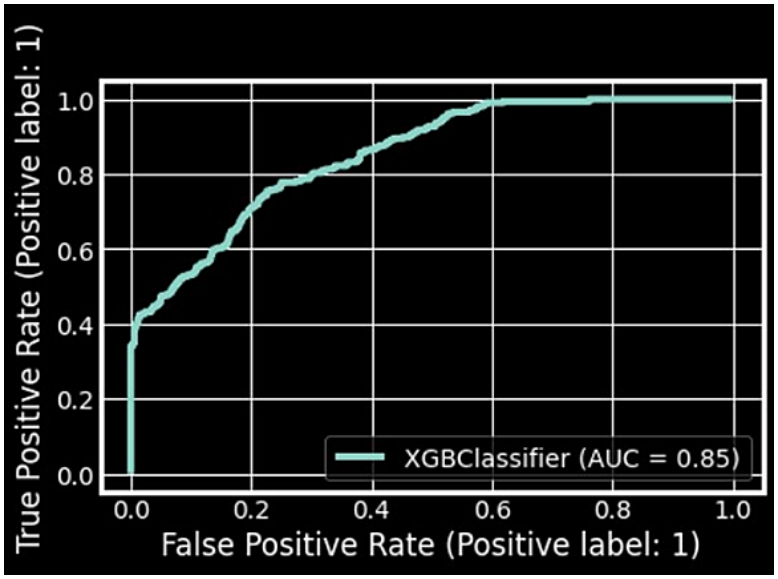


iv ROC(Receiver OperatingCharacteristic)

ROC CURVE - RANDOM FOREST ,XGB CLASSIFIER

```
plot_roc_curve(Rfc,X_test,y_test)
plot_roc_curve(xgb,X_test,y_test)
```

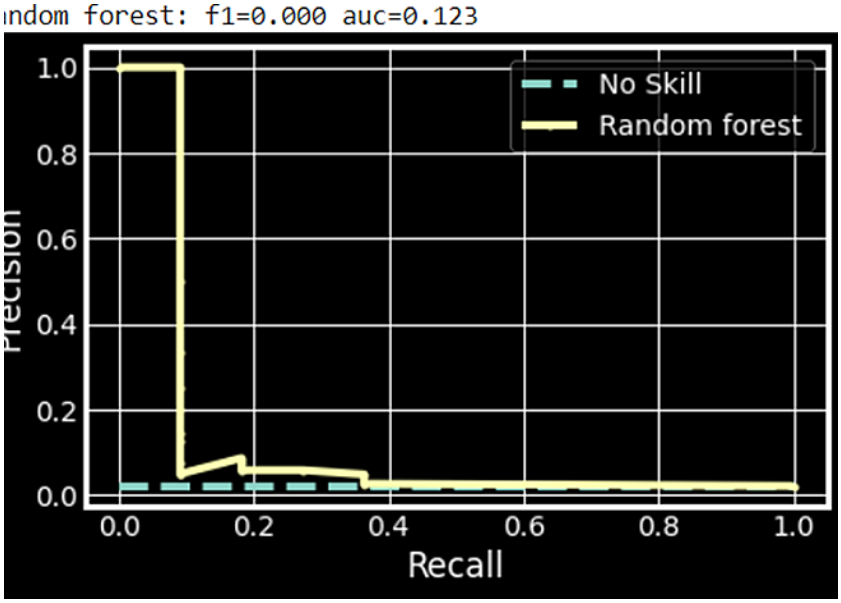
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning:

Function plot_roc_curve is deprecated; Function :func:`plot_roc_curve` is deprecated in 1.0 and will be r

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning:

Function plot_roc_curve is deprecated; Function :func:`plot_roc_curve` is deprecated in 1.0 and will be r

<sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7f44a5ffa550>
```



iv. PR VS ROC CURVE:

```
from sklearn.datasets import make_classification from sklearn.linear_model importLogisticRegression
from sklearn.model_selection import train_test_split from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import aucfrom matplotlib import pyplot
# generate 2 class dataset
X, y = make_classification(n_samples=1000, n_classes=2, weights=[0.99,0.01], random_state=1)# split
intotrain/test sets
trainX, testX, trainy, testy = train_test_split(X, y, test_size=0.5, random_state=2)# fit a model
model = RandomForestClassifier()model.fit(trainX, trainy)
#predict probabilities
lr_probs = model.predict_proba(testX)
# keep probabilities for the positiveoutcome onlylr_probs= lr_probs[:, 1]
# predict class values
yhat = model.predict(testX)
# calculate precision and recall for each threshold
lr_precision, lr_recall, _ = precision_recall_curve(testy, lr_probs)# calculate scores
lr_f1, lr_auc= f1_score(testy, yhat),auc(lr_recall, lr_precision)# summarize scores
print('Random forest:f1=%.3f auc=%.3f' % (lr_f1, lr_auc))# plot the precision-recall curves


no_skill = len(testy[testy==1]) / len(testy)
pyplot.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill') pyplot.plot(lr_recall, lr_precision,
marker='.', label='Random forest')# axis labels
```

pyplot.xlabel('Recall') pyplot.ylabel('Precision')# show the legend pyplot.legend()
# show the plotpyplot.show()



CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Provides better understanding of interaction betweenconstituents.
- Reduces manual work.
- Help the researchers to predict and handle the lotof data.
- The user can access the website anywhere from anydevice.
- Analyse data continually and instantly alert users to changes in the system and reducing the needfor unreliable and expensive sampling.

DISADVANTAGES:

- Requires a high performance server for faster predictions.
- Cannot handle complex data.

CHAPTER 11

CONCLUSION

This project demonstrated a web application that uses machinelearning to recognizehandwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. Water quality levels for drinking purposes were evaluated via the water quality index (WQI) method. We predict the water Quality based on the water metrics, including PH, DO, SS, EC, Turbidity, Chloride, COD, TDS, and Alkalinity, were used in this study. The computed WQI values were found to range between 0 and 100. Water quality analysis is to predict that the water is safe to drink or not using some parameters like PH value, conductivity, hardness, etc. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such aspredicting the qualityof water in an efficient way.

CHAPTER 12
FUTURE SCOPE

The two algorithms are the most widely used evolutionary algorithms. In the future, more evolution algorithms, like differential evolution and ant colony optimization, will be explored to forecast the water quality. On the other hand, ideas from other data processing methods, like boost learning and weighted timing analysis will be investigated. In the future, more efforts will be made to find more datasets to build a more reliable water quality prediction model. This project has endless potential and can always be enhanced to become better.Implementing this conceptin the real world will benefits for people health.

APPENDIX

SOURCE CODE

About.html:

```
  DOCTYPE html>
<html>
 <head>
   <title>Home</title>
   <style>
     body {
       background-image: linear-gradient(to left,#00348d, #8fd1f0);
       background-size: cover;
     }
     .pd {
       padding-bottom: 100%;
     }
     .navbar {
       margin: 0px;
       padding: 20px;
       background-color: white;
       opacity: 0.6;
       color: black;
       font-family: "Roboto", sans-serif;
       font-style: italic;
```

```
      border-radius: 20px;
      font-size: 25px;
    }
    a {
      color: grey;
      float: right;
      text-decoration: none;
      font-style: normal;
      padding-right: 20px;
    }
    p {
      color: turqouise;
      font-style: italic;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <div class="navbar">
    <a href="/predict">Predict</a>
    <a href="/info">Info</a>
    <a href="/about">Home</a>
    <br />
  </div>
```

```
  <br />
  <center>
    <div style="color: white; margin-top: 100px">
      <font
        size="15"
        font-family="Comic Sans MS"
        style="color: white; margin-top: 100px"
        >Efficient Water Quality Analysis & Prediction using Machine Learning
      </font>
    </div>
  </center>
  <div style="color: white; margin-top: 100px">
    <br />
    <center>
      <p>
        Water is very important to the human body. Every one of your cells,
        organs and tissues use water to help with temperature regulation,
        keeping hydrated and maintaining bodily functions. In addition, water
        acts as a lubricant and cushions your joints. Drinking water is great
        for your overall health.
      </p>
    </center>
  </div>
</body>
```

```
</html>
```

Base.html:

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Predict</title>
    <link
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
      rel="stylesheet"
    />

    <style>
      .bar {
        margin: 0px;
        padding: 20px;
        background-color: white;
        opacity: 0.6;
        color: black;
        font-family: "Roboto", sans-serif;
        font-style: italic;
        border-radius: 20px;
        font-size: 25px;
      }
      a {
        color: grey;
```

```
      float: right;
      text-decoration: none;
      font-style: normal;
      padding-right: 20px;
    }
    body {
      background-image: linear-gradient(to bottom,#013ad6 0%, #34dae6 100%);
      background-size: cover;
    }
  </style>
</head>

<body>
  <div class="bar">
    <a href="/predict">Predict</a>
    <a href="/info">Info</a>
    <a href="/about">Home</a>
    <br />
  </div>
  <nav class="navbar navbar-dark bg-dark">
    <div class="container"></div>
  </nav>
  <div class="container">
    <div id="content" style="margin-top: 2em">
      {% block content %}{% endblock %}
    </div>
  </div>
```

```
  </body>
</html>
```

Info.html:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .navbar {
        margin: 0px;
        padding: 20px;
        background-color: white;
        opacity: 0.6;
        color: black;
        font-family: "Roboto", sans-serif;
        font-style: italic;
        border-radius: 20px;
        font-size: 25px;
      }
      a {
        color: grey;
        float: right;
        text-decoration: none;
        font-style: normal;
        padding-right: 20px;
      }
      img {
        width: 550px;
        height: 400px;
```

```
  padding: 10px;
  margin-top: 0px;
}
img:hover {
  border-radius: 100px;
  border-color: grey;
}
body {
  background-image: linear-gradient(to left, rgb(56, 56, 57),rgb(190, 136, 232));
  background-size: cover;
}
h1 {
  font-size: 60px;
  text-align: center;
  color: white;
  font-style: italic;
  font-weight: bolder;
}
div {
  margin-left: 50px;
}
img {
  width: 500px;
  height: 400px;
  padding: 10px;
  margin-top: 0px;
}
img:hover {
  border-radius: 100px;
```

```
      border-color: grey;
    }
  </style>
  <title>Info</title>
 </head>
 <body>
  <div class="navbar">
    <a href="/predict">Predict</a>
    <a href="/info">Info</a>
    <a href="/about">Home</a>
    <br />
  </div>
  <div>
    <h1
      style="
        text-decoration: wavy;
        text-transform: uppercase;
        font-weight: bold;
      "
    >
      WATER IS DRIVING FORCE OF ALL NATURE
    </h1>
  </div>
  <h1> <font color="lightskyblue" size="7" font-family="Comic Sans MS" >pH Value And Its Impact on Water Quality</font></h1>
<p>
<font color="white" size="6" font-family="Comic Sans MS" >pH is a measure of how acidic or basic a water sample is.</font><br/>
<font color="white" size="6" font-family="Comic Sans MS" >The range goes from 0 to 14.</font><br/>
<ul>
```

pH level with less than 7 is acidic.

pH level with greater than 7 is basic.

pH value equal to 7 is neutral.

<h1> Nitrate Value And Its Impact on Water Quality</h1>

<p>

Nitrate occurs naturally and at safe and healthy levels in some foods.

Other sources of nitrate includes Discharge from Sewage systems and animal wastes,etc.

Water Level with less than 6 mg/L can be used for drinking.

Health concern occurs with Nitrate levels over 7 mg/L.

<h1> Dissolved Oxygen Value And Its Impact on Water Quality</h1>

<p>

Dissolved Oxygen (DO) is essential for the survival of fish and other aquatic organisms.

Oxygen is also introduced as a byproduct of aquatic plant photosynthesis.

The Colder water is, the more oxygen it can hold.

The Warmer water is, the less oxygen can be dissolved in it.

When oxygen levels are whiteuced

there are chances of increase in bacteria or algae in water which causes adverse health effects.

<h1> Coliform Value And Its Impact on Water Quality</h1>

<p>

Coliform Bacteria in water indicates the disease causing organisms.

Types of Coliforms are Total Coliform, Fecal Coliform and E.coli

More the amount of Coliform, more the potential contamination sources in the water sample.

Less the amount of Coliform, more purer the water sample is.

<h1> Conductivity Value And Its Impact on Water Quality</h1>

<p>

Conducitvity measures the water's ability to conduct electricity due to presence or absence of certain ions.

Pure Water conducts electricity poorly and can be used for drinking.

Water that has certain chemicals or elements in it, and at varying amount including sodium, magnesium, calcium and chloride is a better conductor of electricity.

</body>

</html>

Predic.html

```
{% extends "base.html" %} {% block content %}

<h1 style="margin-bottom: 20px; color: rgb(220, 183, 89);">
  <center>Water Quality Predictor</center>
</h1>
<div>
  <form id="predict" method="post" action="/predict">
    <center>
      <input
        type="text"
        name="do"
        id=""
        placeholder="Enter D.O"
        style="display: block"
      />
      <br />

      <input
        type="text"
        name="ph"
        id=""
        placeholder="Enter PH"
        style="display: block"
      />
      <br />

      <input
```

```
  type="text"
  name="co"
  id=""
  placeholder="Enter Conductivity"
  style="display: block"
/>
<br />

<input
  type="text"
  name="bod"
  id=""
  placeholder="Enter B.O.D"
  style="display: block"
/>
<br />

<input
  type="text"
  name="na"
  id=""
  placeholder="Enter Nitratenen"
  style="display: block"
/>
<br />

<input
  type="text"
  name="tc"
```

```
        id=""
        placeholder="Enter Total Coliform"
        style="display: block"
      />
      <br />
      <br />
      <input type="submit" value="Submit" />
    </center>
  </form>
  <script />
</div>

{% endblock %}
```

Result.html

```
{% extends "base.html" %} {% block content %}

<h1 style="margin-bottom: 20px; color: rgb(224, 189, 100);">
  <center>Water Quality Predictor</center>
</h1>
<div>
  <div class="loader" style="display: none"></div>
  <h2 style="color: white" id="result">
    <center><span>Result : WQI - {{prediction}}</span></center>
  </h2>
  <style>
    table, th, td {
      border:3px solid rgb(255, 254, 247);
    }
    </style>
```

```
 <h2 style="color:white; font-size: larger;">
  <center>Water Quality Index Analysis</center>
 </h2>
 <br>
 <br>
  <table style="width:100%">
    <tr>
      <th style="color:bisque"><center>Range of Quality of Water</center></th>
      <th style="color:bisque"><center>Analysis Report</center></th>
    </tr>
    <tr>
      <td><b>Value between 95 and 100</b></td>
      <td><b>No purification or treatment of water is needed.It can be used fr drinking purposes as the water is pure.</b></td>
    </tr>
    <tr>
      <td><b>Value between 89 and 94</b></td>
      <td><b>Minor purification or treatment of water is needed.It can be used for drinking or cooking purposes.</b></td>
    </tr>
    <tr>
      <td><b>Value between 80 and 88</b></td>
      <td><b>Conventional purification or treatment of water is needed.It can be used only for cooking purposes.</b></td>
    </tr>
    <tr>
      <td><b>Value between 65 and 79</b></td>
      <td><b>Extensive purification or treatment of water is needed.It can be used for drinking or cooking purposes only if the various impurities are removed.</b></td>
    </tr>
    <tr>
      <td><b>Value between 45 and 64</b></td>
      <td><b>Doubtful in purifying and treating the water so as to get pure water.It can be used for irrigation purposes.</b></td>
    </tr>
    <tr>
      <td><b>Value less than 44</b></td>
```

```
      <td><b>The water is not fit for to be used for driking.It cannot be used for drinking and household purposes and can be used for gardening and irrigational purposes.</b></td>
     </tr>
   </table>
   <br>
   <br>
   <h1><b>Your sample is in the range of 45 and 64 and it is in marginal range.</b></h1>
</div>

{% endblock %}
```

GITHUB:

https://github.com/IBM-EPBL/IBM-Project-13442-1659518556

PROJECT DEMO:

https://drive.google.com/file/d/1AOezG1595MCxuelClAcA_0OGLDpj_ahD/view?usp=share_link