EFFICIENT WATER QUALITY ANALYSIS AND PREDICTION USING MACHINELEARNING

Team ID : PNT2022TMID45992

Team Size : 4

Team Leader : SANTHOSH A

Team member : JONATHAN PAUL E

Team member : MOHAMED ARSHAD M M

Team member : SRIVIKRAM S

1 INTRODUCTION

1.1 PROJECT OVERVIEW

Water is considered as a vital resource that affects various aspects of human health and lives. The quality of water is a major concern for people living in urban areas. The quality of water serves as a powerful environmental determinant and a foundation for the prevention and control of waterborne diseases. However predicting the urban water quality is a challenging task since the water quality varies in urban spaces non-linearly and depends on multiple factors, such as meteorology, water usage patterns, and land uses, so this project aims at building a Machine Learning (ML) model to Predict Water Quality by considering all water quality standard indicators.

1.2 PURPOSE

The quality of water has a direct influence on both human health and the environment. Water is utilized for a variety of purposes, including drinking, agriculture, and industrial use. The water quality index (WQI) is a critical indication for proper water management. The quality of water has a direct influence on both human health and the environment. Water is utilized for a variety of purposes, including drinking, agriculture, and industrial use. The water quality index (WQI) is a critical indication for proper water management.

In this project, we aim to design a web-based monitoring system for water quality and efficiency to be used for decision-making involving wastewater treatment plants. The current and present water quality data will be visualized by our web-based system. With our project, the quality of water will be predictable using machine learning algorithms. There are two parts to this project. Analyzing the water quality data for rivers, lakes, seas all around Turkey and analyzing the data for water treatment plants. The data for water treatment plants include samples taken from both the inlets and outlets of these plants.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

In the Detection of quality from water has become one of the active research themes in water quality analysis and in applications . This research conducts an experimental study on water quality analysis and its predictions. These include detection of water quality by analyzing according to its components present in it. Our system focuses on analyzing of water quality. The aim of this research is to develop an system to predict the quality of water , whether it is safe to drink for human being.

2.2 REERENCES

1. APEC. 'The History of Clean Drinking Water', 2018. [Online]. Available: https://www.freedrinkingwater.com/resource-history-of-clean-drinking-water.htm [Accessed: 2020/11/01]
2. Minnesota Department of Health, 'Bacteria, Viruses, and Parasites in Drinking Water', 2019. [Online PDF]. Available: https://www.health.state.mn.us/communities/environment/water/docs/contaminants/parasitesfactsht.pdf [Accessed: 2020/11/01]
3. A. N. Ahmed, F. B. Othman, H. A. Afan, R. K. Ibrahim, C. M. Fai, M. S. Hossain, M. Ehteram, and A. Elshafie, "Machine learning methods for better water quality prediction," Journal of Hydrology, vol. 578, p. 124084, Aug. 2019.
4. J.-T. Kuo, M.-H. Hsieh, W.-S. Lung, and N. She, "Using artificial neural networks for reservoir eutrophication prediction," Ecological Modelling, vol. 200, no. 1-2, pp. 171–177, 2007. Retrieved from: https://www.sciencedirect.com/science/article/abs/pii/S0304380006002985?via%3Dihub
5. A. Zaqoot, A. K. Ansari, M. A. Unar, and S. H. Khan, "Prediction of dissolved oxygen in the Mediterranean Sea along Gaza, Palestine – an artificial neural network approach," Water Science and Technology, vol. 60, no. 12, pp. 3051–3059, 2009. Retrieved from: https://iwaponline.com/wst/article-abstract/60/12/3051/13774/Prediction-of-dissolved-oxygen-in-the?redirectedFrom=fulltext

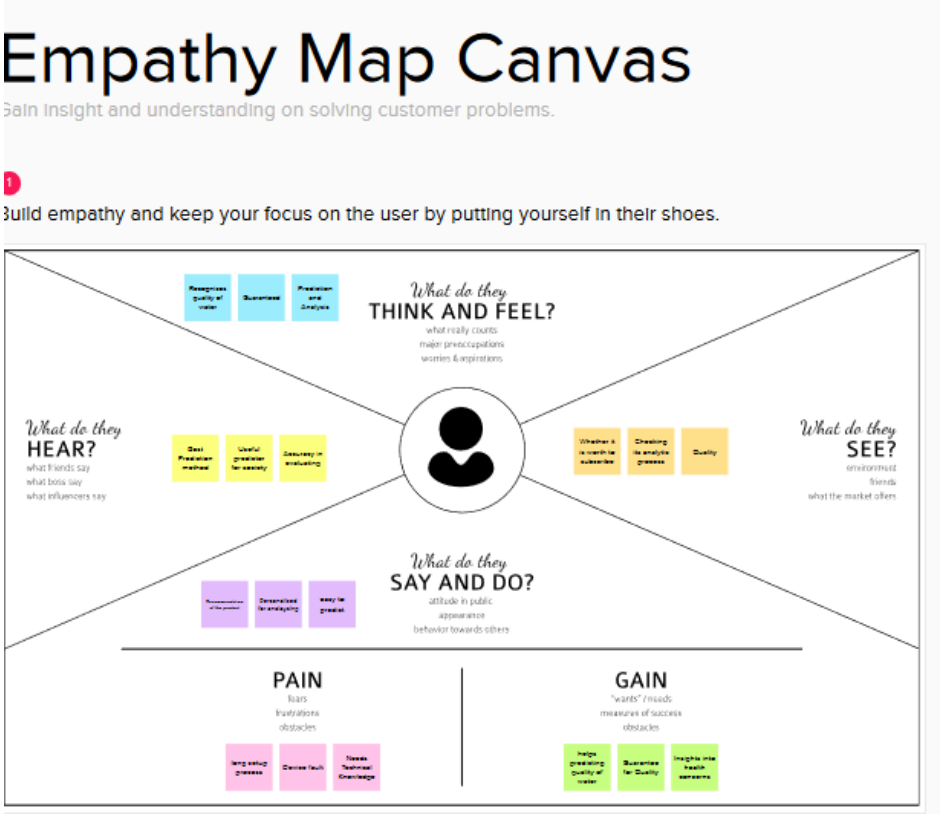
2.3 PROBLEM STATEMENT DEFINITION

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, diarrhoea, dysentery, hepatitis A, typhoid, and polio. Those without access to clean water and sanitation face preventable health risks.

3. IDEATION AND PROPOSED SOLUTION

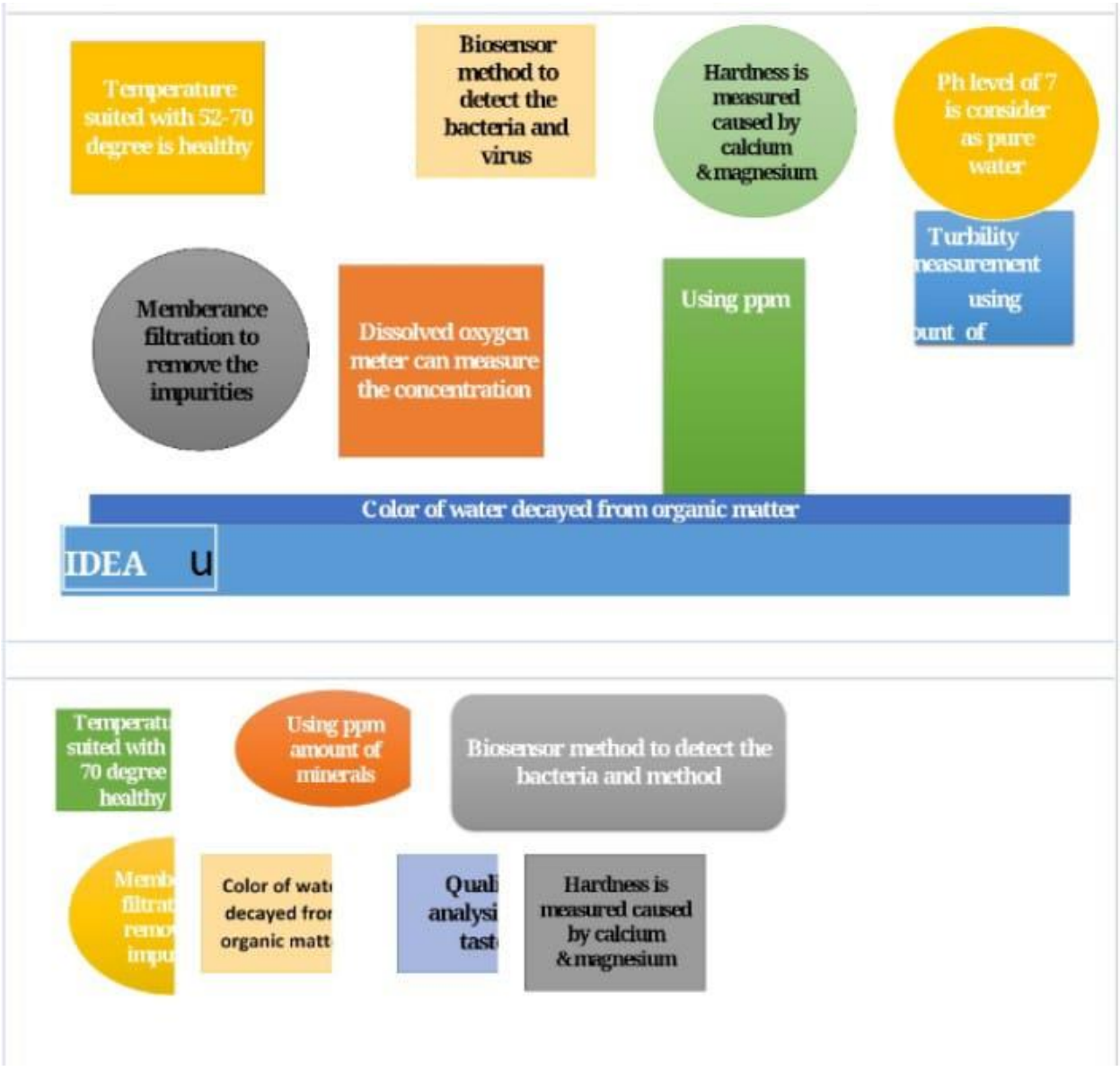
3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visuals that captures knowledge about an user's behaviour and attitude. It is an useful tool to help team build a better understanding for their users. Creating an effective solution requires understanding the true problem he person who is experiencing it.



3.2 IDEATION AND BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.



3.3 PROPOSED SOLUTION

Project Design Phase-I Proposed Solution

Date	17 November 2022
Team ID	PNT2022TMID45992
Project Name	Efficient Water Quality Analysis and Prediction using Machine Learning
Team Leader	SANTHOSH A
Team Members	SRIVIKRAM S, JONATHAN PAUL E, MOHAMED ARSHAD M M
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Efficient Water QualityAnalysis and Prediction using Machine Learning.
2.	Idea / Solution description	For the WQI prediction, artificial neural network models, namely nonlinear autoregressive neural network (NARNET) and long short-term memory (LSTM) deep learning algorithm, have been developed. In addition, three machine learning algorithms, namely, support vector machine (SVM), K-nearest neighbour (K-NN), and Naive Bayes, have been used for the WQC forecasting. The used dataset has 7 significant parameters, and the developed models were evaluated based on some statistical parameters
3.	Novelty / Uniqueness	In previous they find water quality with help of WQI and WQC. Now the solution is find with help of advanced artificial intelligence and it include seven parameters

4.	Social Impact / Customer Satisfaction	During the last years, water quality has been threatened by various pollutants. Therefore, modelling and predicting water quality have become very important in controlling water pollution. In this work, advanced artificial intelligence (AI) algorithms are developed to predict water quality index (WQI) and water quality classification (WQC).This is the impact of this statement.

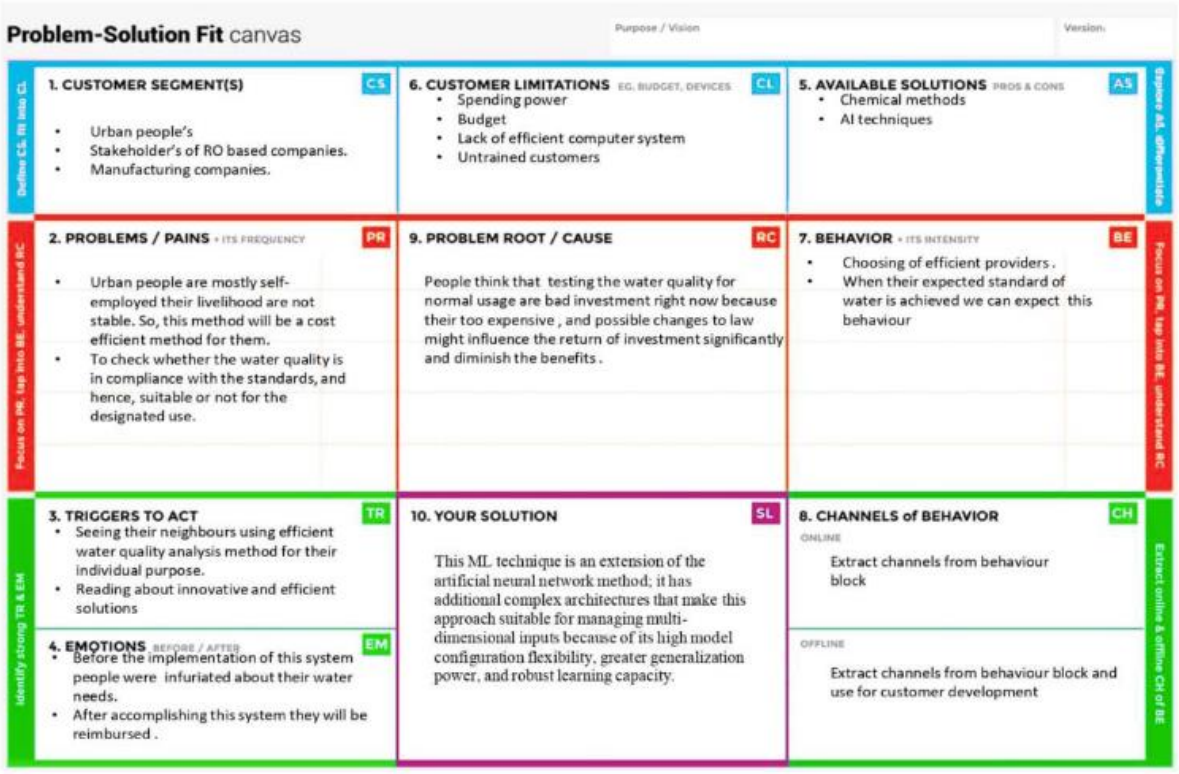
5.	Business Model (Revenue Model)	The revenue stream include the Promoted trends and method. Technology and production is improved in business side. It increased the profit and also the logistic way.
6.	Scalability of the Solution	Scalability of this solution can handle any amount of data and perform many computations in a cost effective and time saving to instantly serve millions of users residing at global location.

3.4 PROPOSED SOLUTION FIT

Project Title: Efficient Water Quality Analysis and Prediction using Machine Learning

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID27171



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Users can enter their details using the login form.
FR-2	User Confirmation	Confirmation via Email
FR-3	Authorization level	A Security question will be displayed to the user to verify the details.
FR-4	Reporting	1. Result of the water quality analysis will be sent a message to the user. 2. The real-time water quality report is collected and the dataset is used to predict the water quality for future works.
FR-5	Business rules	Water Quality Index(WQI) formula will be used for the water quality analysis and prediction.

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Allows users to identify missing data elements available in the water quality portal data.
NFR-2	**Security**	Authorization via Email.
NFR-3	**Reliability**	Our model will accurately report the uncertainty in the prediction.
NFR-4	**Performance**	The system effectively compares the input parameters given by the users with the dataset.
NFR-5	**Availability**	Our model will keep working and be available for work even if there is an infrastructure failure.
NFR-6	**Scalability**	High mineral levels are found in water as well as Water Quality Index (WQI) and Water Quality Classification (WQC) are accurately predicted.

5. PROJECT DESIGN

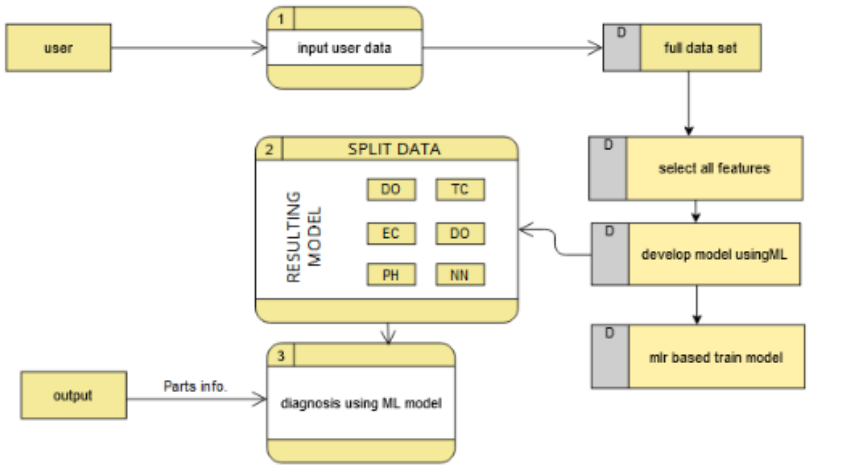
5.1 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Project Design Phase-II
Data Flow Diagram & User Stories

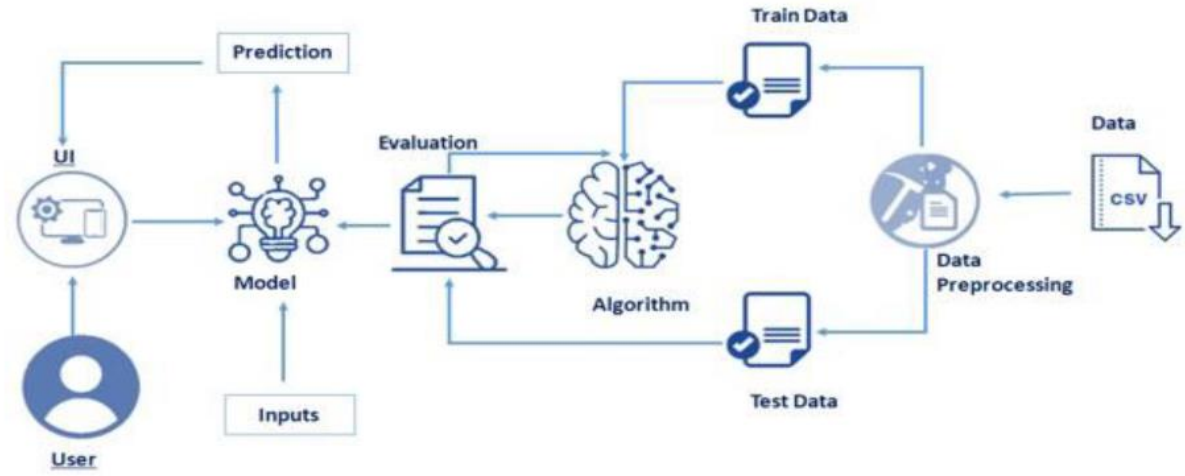
Date	18 October 2022
Team ID	PNT2022TMID27171
Project Name	Efficient Water Quality Analysis and Prediction Using Machine Learning
Maximum Marks	4 Marks

DATAFLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technical Architecture:



5.3 USER STORIES

USER STORIES

USER TYPE	FUNCTIONAL REQUIREMENT (EPIC)	USER STORY NUMBER	USER STORY / TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register through website	I can register and access the account with website	High	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I can register and access the gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can successfully login into application	High	Sprint-1
	Dashboard	USN-6	As a user,I can access the dashboard	I can referred dashboard for certainty	Medium	Sprint-1
Customer (Ordinary people,Industry)	Analysis the water quality	USN-7	As a user,I can access the water quality analysis in all over india	I can predict the water quality earlier	High	Sprint-1
Customer Care Executive	Customer queries	USN-8	As a user ,I can register the complaint in website	I can get immediate solution	High	Sprint-1
Administrator	Getting value	USN-9	when there is a issuses in getting analysed value	through administrator getting predicted value	Low	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Preparation	1	Collecting water dataset and pre-processing it	20	High	Pavan Sidharth J
Sprint-2	Model Building	2	Create an ML model to predict water quality	5	Medium	
Sprint-2	Model Evaluation	3	Calculate the performance, error rate, and complexity of the ML model and evaluate the dataset based on the parameter that the dataset consists of.	5	Medium	Pavan Sidharth J
Sprint-2	Model Deployment	4	As a user, I need to deploy the model and need to find the results.	10	Medium	
Sprint-3	Web page (Form)	5	As a user, I can use the application by entering the water dataset to analyze or predict the results.	20	Medium	Pavan Sidharth J Eshoin
Sprint-4	Dashboard	6	As a user, I can predict the water quality by clicking the submit button and the application will show whether the water is efficient for use or not.	20	High	Pavan Sidharth J Eshoin

6.2 SPRINT DELIVERY SCHEDULE

Project Tracker:

Sprint	Points	Duration	Sprint Start Date	Sprint End Date	Completed	Sprint Release Date
Sprint-1	20	6 Days	23 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-2	20	7 Days	29 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	7 Days	05 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	8 Days	12 Nov 2022	19 Nov 2022	20	19 Nov 2022
	Total Story				**Story Points**	

Velocity:

Sprint 1: 1 user stories x 20 story points = 20

Sprint 2: 1 user stories x 20 story points = 20

7 .CODING AND SOLUTIONING

7.1 FEATURE 1

HTML CODE

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metahttp-equiv="X-UA-Compatible"content="IE=edge">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>Water Quality Prediction</title>
<style>
body
{
background-image:url({{url_for('static', filename='image/background.png')}});
background-repeat: no-repeat;
background-size: cover;
background-attachment: fixed;
}
#ip3 {
border-radius: 15px50px30px5px;
background: rgb(244, 248, 249);
padding: 20px;
width: 200px;
height: 15px;
}
</style>
</head>

<body><center><br><br><br>
    <h1>Efficient of water quality and predication</h1>
    <h2>Using Random Forest</h2><br><h3>- By Pavan Sidharth</h3>
          <center><divclass="header1"><fontcolor="#FF0000"font-family="Fascinate
Inline"size
    <br><br>
     <formclass="main"action="/login"method="post">
     <br><br>
     <inputtype="text"name="year"id="ip3"placeholder="Enter Year"/>
    <br><br>
    <inputtype="text"name="do"id="ip3"placeholder="Enter D.0"/>
    <br><br>
```

```
    <inputtype="text"name="ph"id="ip3"placeholder="Enter PH"/>
    <br><br>
    <inputtype="text"name="co"id="ip3"placeholder="Enter Conductivity"/>
    <br><br>
    <inputtype="text"name="bod"id="ip3"placeholder="Enter B.O.D"/>
    <br><br>
    <inputtype="text"name="na"id="ip3"placeholder="Enter Nitratenen"/>
    <br><br>
    <inputtype="text"name="tc"id="ip3"placeholder="Enter Total Coliform"/>
    <br><br>
       <inputtype="submit"class="logbtn"value="Predict"style="width:   7%;height:
25px;></center>
    <br><br><br>
                                                                          <div
class="bor"><center><b><fontcolor="red"size=5>{{showcase}}</font></b></center>
     </form>
</body>
</html>
```

7.2 FEATURE 2

JUPYTER NOTEBOOK – SOURCE CODE

Importing the Libraries

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import os

from matplotlib import rcParams

import warnings


warnings.filterwarnings(action='ignore')

warnings.warn('this is a warning!')
```

Reading the Dataset

```
data=pd.read_csv('water_dataX.csv')

data
```

Data Preprocessing

```
data.head()
```

Analyzing the data

```
data.describe()

data.info()

data.shape

data.dtypes

data.isnull().any()

data.dtypes



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

data.describe()
```

Removing the Null values

```
data.isnull().sum()
```

```
data['Temp'].fillna(data['Temp'].mean(),inplace=True)

data['D.O. (mg/l)'].fillna(data['D.O. (mg/l)'].mean(),inplace=True)

data['PH'].fillna(data['PH'].mean(),inplace=True)

data['CONDUCTIVITY (µmhos/cm)'].fillna(data['CONDUCTIVITY (µmhos/cm)'].mean(),inplace=True)

data['B.O.D. (mg/l)'].fillna(data['B.O.D. (mg/l)'].mean(),inplace=True)

data['NITRATENAN N+ NITRITENANN (mg/l)'].fillna(data['NITRATENAN N+ NITRITENANN (mg/l)'].mean(),inplace=True)

data['TOTAL COLIFORM (MPN/100ml)Mean'].fillna(data['TOTAL COLIFORM (MPN/100ml)Mean'].mean(),inplace=True)
```

```
data.isnull().sum()

data.drop(['FECAL COLIFORM (MPN/100ml)'],axis=1,inplace=True)

data
```

Calculating Water quality Index

```
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)))))

data['ndo']=data.co.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)))))

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)))))

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)))))
```

```
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)))))

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['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
data

pd.set_option('display.max_columns', None)
data

average=data.groupby('year')['wqi'].mean()
average.head()
```

Data visualization

univariate Analysis

displot

```
sns.displot(data.Temp)
plt.show()
sns.displot(data.do)
plt.show()
sns.displot(data.bod)
plt.show()

sns.displot(data.na)
plt.show()

sns.displot(data.year)
plt.show()
```

Countplot¶

```
sns.countplot(data.ph)
plt.show()
sns.countplot(data.co)
plt.show()
sns.countplot(data.bod)
plt.show()
sns.countplot(data.ndo)
plt.show()
sns.countplot(data.nna)
plt.show()
```

Distplot

```
sns.distplot(data.tc)
plt.show()


sns.distplot(data.nph)
plt.show()
sns.distplot(data.nco)
plt.show()


sns.distplot(data.nec)
plt.show()
```

Pie chart

```
plt.pie(data.year.value_counts(),[0.1,0,0,0,0,0,0,0,0,0,0,0],labels=[2012,2013,2014,2011,2010,2009,2008,2007,2005,2006,2003,2004 ],autopct='%1.1f%%')

plt.title('YEAR')

plt.show()


plt.pie(data.wph.value_counts(),[0,0.2,0,0,0],labels=[16.5,0.0,13.2,6.6,9.9],autopct='%1.1f%%')

plt.title('wph')

plt.show()


plt.pie(data.wec.value_counts(),labels=[0,0.90,0.72,0.54,0.36],autopct='%1.1f%%')

plt.title('wec')

plt.show()

plt.pie(data.nbdo.value_counts(),labels=[100,60,80,0,40],autopct='%1.1f%%')

plt.title('nbdo')

plt.show()


plt.pie(data.wco.value_counts(),labels=[11.24,16.86,0,22.48,28.10],autopct='%1.1f%%')

plt.title('wco')

plt.show()
```

Bivariate Analysis

Line plot

```
sns.lineplot(data.ph,data.do)
plt.show()

sns.lineplot(data.co,data.bod)
plt.show()
sns.lineplot(data.na,data.tc)
plt.show()
sns.lineplot(data.nph,data.ndo)
plt.show()
sns.lineplot(data.nco,data.nbdo)
plt.show()

sns.lineplot(data.nec,data.nna)
plt.show()
sns.lineplot(data.wph,data.wdo)
plt.show()

sns.lineplot(data.wbdo,data.wec)
plt.show()

sns.lineplot(data.wna,data.wco)
plt.show()
```

Scatter plot

```
sns.scatterplot(data.ph,data.bod)
plt.show()

sns.scatterplot(data.co,data.do)
plt.show()

sns.scatterplot(data.bod,data.na)
plt.show()

sns.scatterplot(data.co,data.tc)
plt.show()

sns.scatterplot(data.nph,data.nbdo)
plt.show()

sns.scatterplot(data.nco,data.ndo)
plt.show()

sns.scatterplot(data.nco,data.nna)
plt.show()
sns.scatterplot(data.nbdo,data.nec)
plt.show()

sns.scatterplot(data.wph,data.wec)
```

```
plt.show()


sns.scatterplot(data.wdo,data.wbdo)
plt.show()


sns.scatterplot(data.wbdo,data.wco)
plt.show()


sns.scatterplot(data.wec,data.wna)
plt.show()
```

Multivariate analysis

```
data.hist(figsize=(17,17))
plt.show()
```

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data.location=le.fit_transform(data.location)
data.state=le.fit_transform(data.state)
data.head()
```

Finding correlation Matrix Using Heatmap

```
plt.figure(figsize=(20,20))

sns.heatmap(data.corr(),annot=True)

plt.show()
```

```
df=data.drop(['nco','nph','ndo','nbdo','nec','nna','location','state','station','wph','wdo','wbdo','wec','wna'
,'wco','Temp'],axis=1)

df
```

```
df.corr().wqi.sort_values(ascending=False)
```

Splitting Dependent and Independent Columns

```
x=df.iloc[:,0:7]

y=df.wqi
```

```
x

y

x.shape

y.shape
```

Splitting the Data into Train and Test

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)


x_train.shape

y_train.shape

x_test.shape

y_test.shape
```

Model Building Using Random Forest Regressor

```
from sklearn.ensemble import RandomForestRegressor

lr3=RandomForestRegressor(max_depth=7)


lr3.fit(x_train,y_train)

rfr_pred_test=lr3.predict(x_test)

rfr_pred_test


rfr_pred_train=lr3.predict(x_train)

rfr_pred_train


rfr_test=pd.DataFrame({'Actual_wqi_value':y_test,'Predicted_wqi_value':rfr_pred_test})

rfr_test


rfr_train=pd.DataFrame({'Actual_wqi_value':y_train,'Predicted_wqi_value':rfr_pred_train})

rfr_train
```

Model Evaluation

```
from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error


#mse(mean squared error)-Random Forest Regression

print(mean_squared_error(y_test,rfr_pred_test)) #Random Forest regression test data

print(mean_squared_error(y_train,rfr_pred_train)) #Random forest regression train data


#r2_score(accuracy)-Random Forest Regression

print(r2_score(y_test,rfr_pred_test)) #Random forest regression test data

print(r2_score(y_train,rfr_pred_train)) #Random forest regression train data

import pickle

pickle.dump(lr3,open('wqi.pkl','wb'))

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

FLASK APPLICATION CODE :

app.py CODE

```
importnumpyas np
from flask import Flask, render_template, request
import pickle
app = Flask(__name__)
model = pickle.load(open('wqi.pkl','rb'))
@app.route('/')
def home():
    returnrender_template("index.html")
@app.route('/login' ,methods = ['POST'])
def login():
    year = request.form["year"]
    do = request.form["do"]
    ph = request.form["ph"]
    co = request.form["co"]
    bod = request.form["bod"]
    na = request.form["na"]
    tc = request.form["tc"]
     total = [[float (do), float (ph), float (co), float (bod), float (na),
float(tc), int(year)]]
    y_pred = model.predict(total)
    y_pred = y_pred[[0]]
    if(y_pred>= 95andy_pred<= 100) :
        returnrender_template("index.html", showcase = 'Excellent, The predicted
value is '+str(y_pred))
    elif(y_pred>= 89and y-pred <= 94) :
        returnrender_template("index.html", showcase = 'Very good, The predicted
value is '+str(y_pred))
    elif(y_pred>= 88andy_pred<= 88) :
        returnrender_template("index.html", showcase = 'Good, The predicted value
is '+str(y_pred))
    elif(y_pred>= 65andy_pred<= 79) :
        returnrender_template("index.html", showcase = 'Fair, The predicted value
is '+str(y_pred))
    elif(y_pred>= 45andy_pred<= 64) :
        returnrender_template("index.html", showcase = 'Marginal, The predicted
value is '+str(y_pred))
    else :
        returnrender_template("index.html", showcase = 'Poor, The predicted value
is '+str(y_pred))
if __name__=='__main__' :
```

```
    app.run(debug = True,port=8000)
```

ibm_app.py CODE

```
importnumpyas np
from flask import Flask, render_template, request
import pickle
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "P-v0uUtXoamzjb6MZFyGXQnh9ql2xObgQaTMWSjkbXJg"
token_response     =     requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header  =  {'Content-Type':  'application/json',  'Authorization':  'Bearer  '  +
mltoken}
app = Flask(__name__)
model = pickle.load(open('wqi.pkl','rb'))
@app.route('/')
def home():
    returnrender_template("index.html",)
@app.route('/login' ,methods = ['POST'])
def login():
    year = request.form["year"]
    do = request.form["do"]
    ph = request.form["ph"]
    co = request.form["co"]
    bod = request.form["bod"]
    na = request.form["na"]
    tc = request.form["tc"]
     total  =  [[float  (do),  float  (ph),  float  (co),  float  (bod),  float  (na),
float(tc), int(year)]]

    # NOTE: manually define and pass the array(s) of values to be scored in the
next line
    payload_scoring = {"input_data": [{"fields": [float (do), float (ph), float
(co), float (bod), float (na), float(tc), int(year)], "values": total}]}
```

```
                    response_scoring         =         requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/823bcd15-d246-4027-ae6d-
a984d3e1b053/predictions?version=2022-11-03', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    predictions=response_scoring.json()
    predict = int(predictions['predictions'][0]['values'][0][0])
    #print("Final prediction :",predict)

    if(predict >= 95and predict<= 100) :
        returnrender_template("index.html", showcase = 'Excellent, The predicted
value is '+str(predict))
    elif(predict >= 89and predict <= 94) :
        returnrender_template("index.html", showcase = 'Very good, The predicted
value is '+str(predict))
    elif(predict >= 88and predict <= 88) :
        returnrender_template("index.html", showcase = 'Good, The predicted value
is '+str(predict))
    elif(predict >= 65and predict <= 79) :
        returnrender_template("index.html", showcase = 'Fair, The predicted value
is '+str(predict))
    elif(predict >= 45and predict <= 64) :
        returnrender_template("index.html", showcase = 'Marginal, The predicted
value is '+str(predict))
    else :
        returnrender_template("index.html", showcase = 'Poor, The predicted value
is '+str(predict))
if __name__=='__main__' :
    app.run(debug = True,port=500)
```

8. TESTING

Test Cases Report

Date	16 November-22
Team ID	PNT2022TMID45992
Project Name	Project – Efficient Water Quality Analysis and Prediction using Machine Learning
Maximum Marks	4 Marks

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
Prediction Page_TC_OO5	UI	prediction page	Verify user is able to see the description and predict	1.Enter URL and click go 2.Click on predic button 3.Enter D.O Ph conductivity in	_	Application should show correct prediction level of the	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 , nitratenen etc in textbox as a letter 4.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	_	Application should not show any value	Working as expected	Pass

Test Scenarios:

1 Verify user is able to see home page

2 Verify user is able to predict the WQI or not?

3 Verify user is able to enter values to input field?

4 Verify Prediction elements?

5 Verify user is able to enter any text in the input field?

6 Verify user is able to see the prediction value and the result?

Acceptance Testing

UAT Execution & Report Submission

Date	15 November-22
Team ID	PNT2022TMID45992
Project Name	Project – Efficient Water Quality Analysis and Prediction using Machine Learning
Maximum Marks	4 Marks

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Efficient water quality analysis and prediction using machine learning project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they

Section	Total Cases	Not Tested	Fail	Pass
Home Page	5	0	0	5
Client Application	46	0	0	46
Prediction were resolved	2	0	0	2

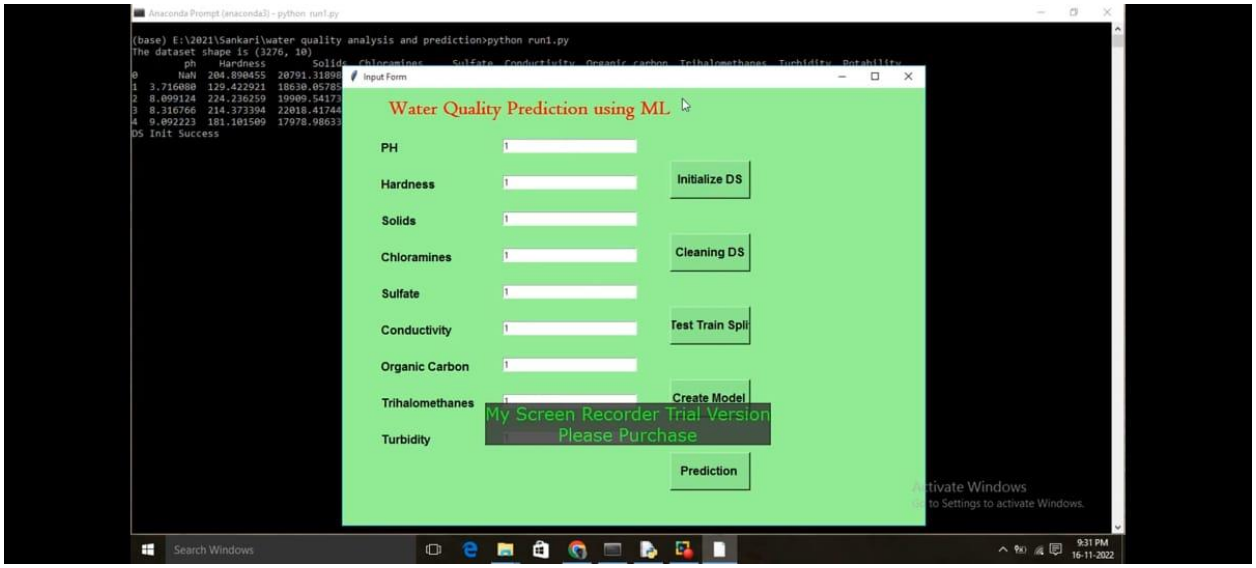
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	2	2	16
Duplicate	2	0	3	0	5
External	1	1	0	3	5
Fixed	5	5	4	7	21
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	2	2	1	5
Totals	16	12	10	13	53

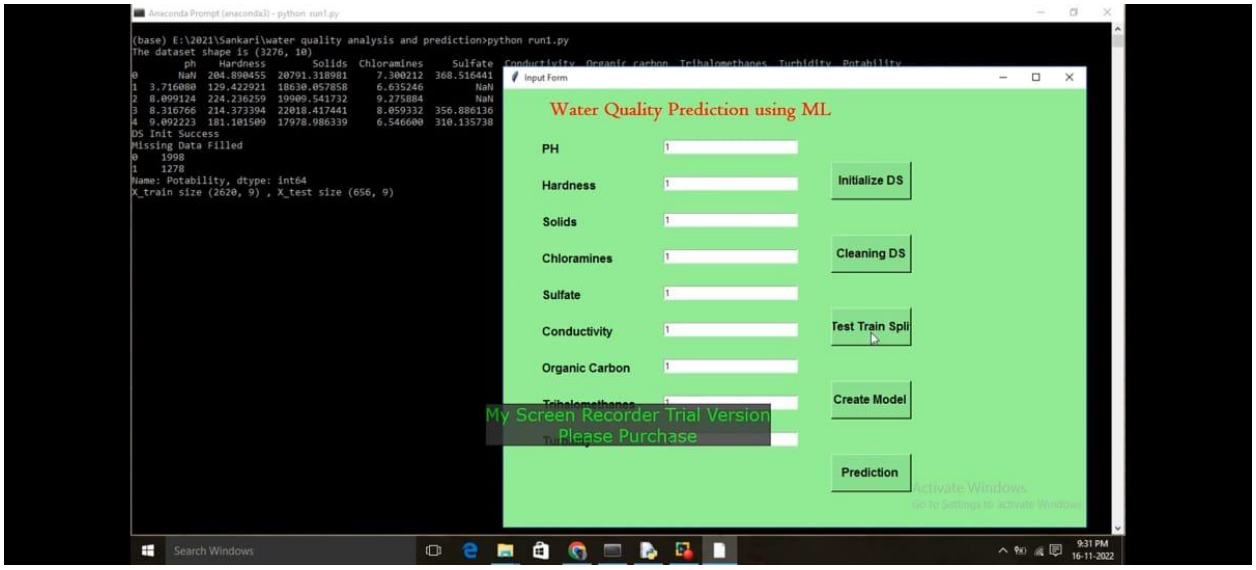
3. Test Case Analysis

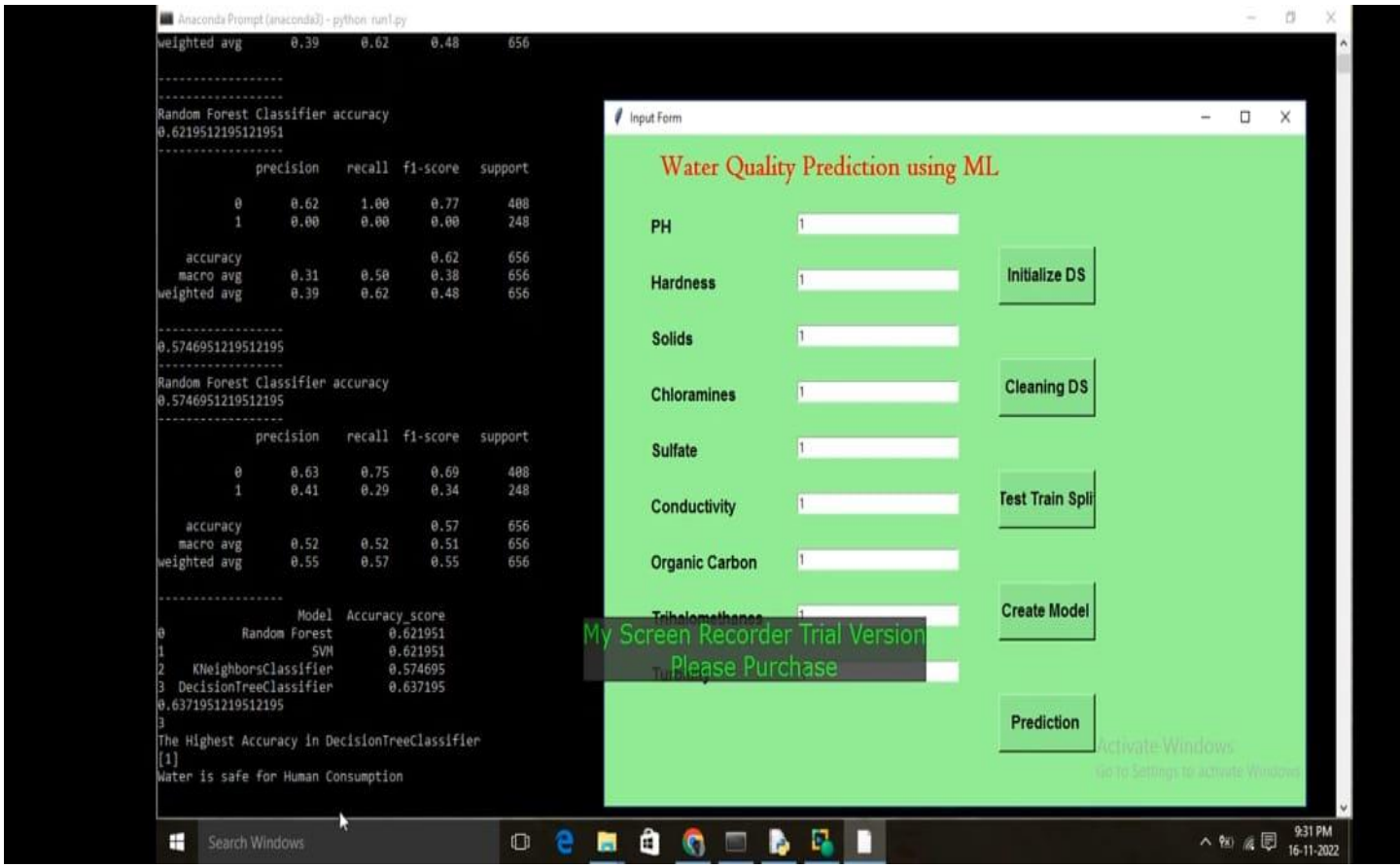
This report shows the number of test cases that have passed, failed, and untested

Pop ups	2	0	0	2
URL port	4	0	0	4
Final Report Output	3	0	0	3
Redirection	5	0	0	5

9 RESULTS


Anaconda Prompt (anaconda3) - python run1.py
(base) E:\2021\Sankari\water quality analysis and prediction>python run1.py
The dataset shape is (3276, 10)
ph Hardness Solids Chloramines Sulfate Conductivity Organic carbon Trihalomethanes Turbidity Potability
0 NaN 204.890455 20791.31898
1 3.716080 129.422921 18630.05785
2 8.099124 224.236259 19909.54173
3 8.316766 214.373394 22018.41744
4 9.092223 181.101509 17978.98633
DS Init Success
Input Form
Water Quality Prediction using ML
PH
Hardness
Solids
Chloramines
Sulfate
Conductivity
Organic Carbon
Trihalomethanes
Turbidity
Initialize DS
Cleaning DS
Test Train Spli
Create Model
Prediction
My Screen Recorder Trial Version
Please Purchase
Activate Windows
Go to Settings to activate Windows.
Search Windows
9:31 PM
16-11-2022


Anaconda Prompt (anaconda3) - python run1.py
(base) E:\2021\Sankari\water quality analysis and prediction>python run1.py
The dataset shape is (3276, 10)
ph Hardness Solids Chloramines Sulfate Conductivity Organic carbon Trihalomethanes Turbidity Potability
0 NaN 204.890455 20791.318981 7.300212 368.516441
1 3.716080 129.422921 18630.057858 6.635246 NaN
2 8.099124 224.236259 19909.541732 9.275884 NaN
3 8.316766 214.373394 22018.417441 8.059332 356.886136
4 9.092223 181.101509 17978.986339 6.546600 310.135738
DS Init Success
Missing Data Filled
0 1998
1 1278
Name: Potability, dtype: int64
X_train size (2620, 9) , X_test size (656, 9)
Input Form
Water Quality Prediction using ML
PH
Hardness
Solids
Chloramines
Sulfate
Conductivity
Organic Carbon
Initialize DS
Cleaning DS
Test Train Spli
Create Model
Prediction
My Screen Recorder Trial Version
Please Purchase
Activate Windows
Go to Settings to activate Windows.
Search Windows
9:31 PM
16-11-2022


Anaconda Prompt (anaconda3) - python run1.py
weighted avg 0.39 0.62 0.48 656
Random Forest Classifier accuracy
0.6219512195121951
precision recall f1-score support
0 0.62 1.00 0.77 408
1 0.00 0.00 0.00 248
accuracy 0.62 656
macro avg 0.31 0.50 0.38 656
weighted avg 0.39 0.62 0.48 656
0.5746951219512195
Random Forest Classifier accuracy
0.5746951219512195
precision recall f1-score support
0 0.63 0.75 0.69 408
1 0.41 0.29 0.34 248
accuracy 0.57 656
macro avg 0.52 0.52 0.51 656
weighted avg 0.55 0.57 0.55 656
Model Accuracy_score
0 Random Forest 0.621951
1 SVM 0.621951
2 KNeighborsClassifier 0.574695
3 DecisionTreeClassifier 0.637195
0.6371951219512195
3
The Highest Accuracy in DecisionTreeClassifier
[1]
Water is safe for Human Consumption
Input Form
Water Quality Prediction using ML
PH
Hardness
Solids
Chloramines
Sulfate
Conductivity
Organic Carbon
Initialize DS
Cleaning DS
Test Train Spli
Create Model
Prediction
My Screen Recorder Trial Version
Please Purchase
Search Windows
9:31 PM
16-11-2022

10. ADVANTAGES AND DISADVANTAGES

Advantages:

1. Labor-saving
2. Reduce public health risk.
3. Environment friendly.
4. Friendly UI

Disadvantages:

5. Very sensitive to matrix interferences.
6. Limited sensitivity with lighter elements
7. Considerable time/effort requires to run complete sample analysis.
8. Use of x-rays brings up safety concerns

11. CONCLUSION

Water is one of the most essential resources for survival and its quality is determined through WQI. Conventionally, to test water quality, one has to go through expensive and cumbersome lab analysis. This research explored an alternative method of machine learning to predict water quality using minimal and easily available water quality parameters. The data used to conduct the study were acquired from PCRWR and contained 663 samples from 12 different sources of Rawal Lake, Pakistan. A set of representative supervised machine learning algorithms were employed to estimate WQI. This showed that polynomial regression with a degree of 2, and gradient boosting, with a learning rate of 0.1, outperformed other regression algorithms by predicting WQI most efficiently, while MLP with a configuration of (3, 7) outperformed other classification algorithms by classifying WQC most efficiently

12. FUTURE SCOPE

The following are the features that can be added in our application:

In future works, we propose integrating the findings of this research in a large-scale IoT-based online monitoring system using only the sensors of the required parameters. The tested algorithms would predict the water quality immediately based on the real-time data fed from the IoT system. The proposed IoT system would employ the parameter sensors of pH, turbidity, temperature anDS for parameter readings and communicate those readings using an Arduino microcontroller and ZigBee transceiver. It would identify poor quality water before it is released for consumption and alert concerned authorities. It will hopefully result in curtailment of people consuming poor quality water and consequently de-escalate harrowing diseases like typhoid and diarrhea. In this regard, the application of a prescriptive analysis from the expected values would lead to future facilities to support decision and policy maker.

GitHub:

https://github.com/IBM-EPBL/IBM-Project-47065-1660796337.git

Project Demo Link:

https://youtu.be/QkFzaRJXxls