TEAM ID: PNT2022TMID29328

PROJECT TITLE: Efficient Water Quality Analysis & Prediction using Machine Learning

Project Report

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.

1.2 Purpose

This project aims at building a Machine Learning (ML) model to Predict Water Quality by considering all water quality standard indicators.Using ML techniques (Regression models) to predict the quality of water instead of using physical measurements or sensors to obtain the quality of water. ML techniques improves the accuracy of measurement over existing chemical and physical techniques as it is infeasible to obtain all the required features to predict the water quality.

2. LITERATURE SURVEY

2.1 Existing problem

The proposed system is intended to determine portability. It is divided into two phases, one for training and the other for testing. The following procedures are carried out in both sections. The data set was chosen as follows: The collection of essential parameters that affect water quality, identification of the number of data samples, and definition of the class labels for each data sample present in the data are all factors that go into selecting the water quality data set, which is a prerequisite to model construction. Ten indicator parameters make up the data sets used in this study. pH value and hardness are examples of these factors. The proposed approach, however, is not constrained by the number of parameters or the selection of parameters. A k-fold cross-validation technique is employed to set the learning and testing framework in this study, corresponding to each data sample in the data set. Using this technique, the dataset is separated into k-disjoint sets of equal size, each with roughly the same class distribution. In turn, this division's subsets are utilized as the test set, with the remaining subsets serving as the training set. These are the Decision Tree (DT) and K-Nearest Neighbour (KNN) methods. Each strategy takes a different approach in terms of the underlying relational structure between the indicator parameters and the class label. As a result, each technique's performance for the same data set is likely to differ. Validating the performance of different classifiers on an unknown data set: Data mining provides several metrics for validating the performance of different classifiers on an unknown data set. A repeated cross-validation procedure in the Matlab caret package created the learning and testing environment. The following procedure was used to apply the classification algorithm:

1. The data set was split into training (80%) and testing (20%). (20 percent).
2. The training set was subjected to repeated cross-validation, with the number ofiterations fixed to Classifiers being trained in this manner.
3. The model's optimal parameter configuration was selected, resulting in maximumaccuracy.
4. The model was scrutinized.

2.2 References

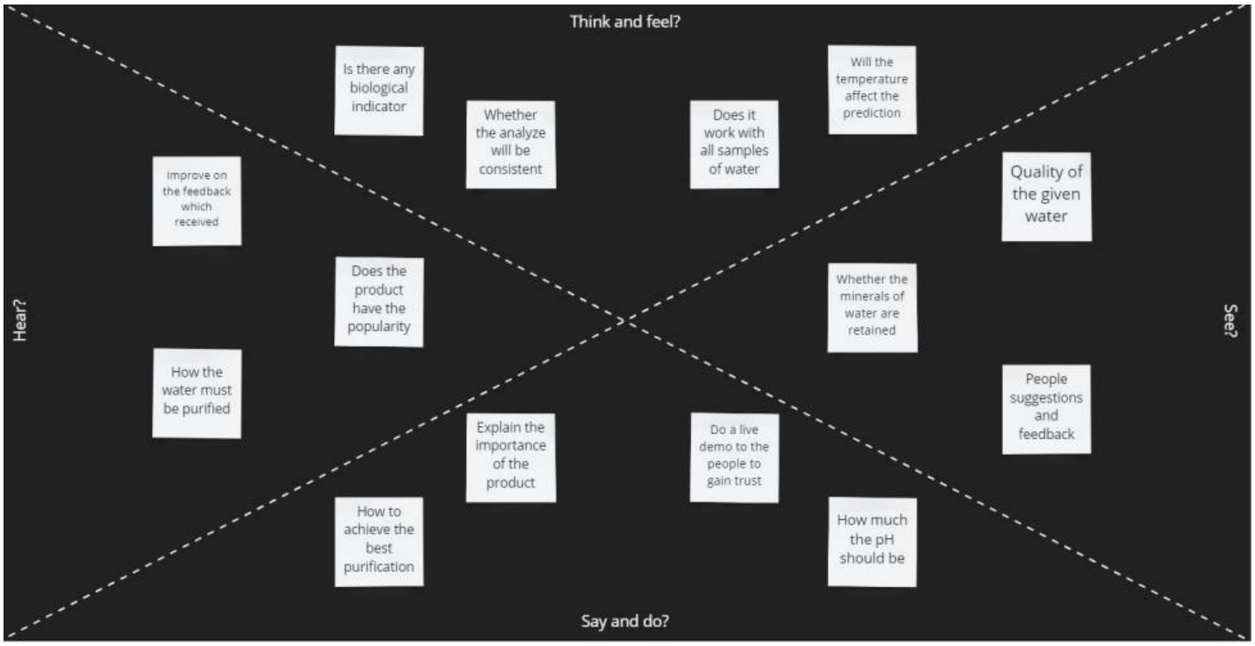
- PCRWR. National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006); Pakistan Council of Research in Water Resources Islamabad: Islamabad, Pakistan, 2007.
- Ling, J.K.B. Water Quality Study and Its Relationship with High Tide and Low Tide at Kuantan River. Bachelor’s Thesis, Universiti Malaysia Pahang, Gambang, Malaysia, 2010.

2.3 Problem Statement Definition

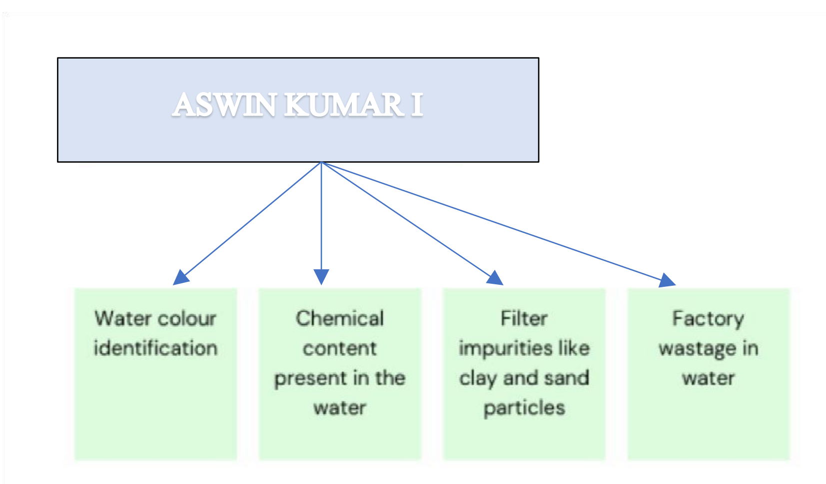
The main aim of the project is to predict the quality of the water. We are building a web app to predict the quality of the water. Project aims at building a Machine Learning (ML) model to Predict Water Quality by considering all water quality standard indicators. WQI is fundamentally calculated by initially multiplying the q value of each parameter by its corresponding weight, adding them all up and then dividing the result by the sum of weights of the employed parameters

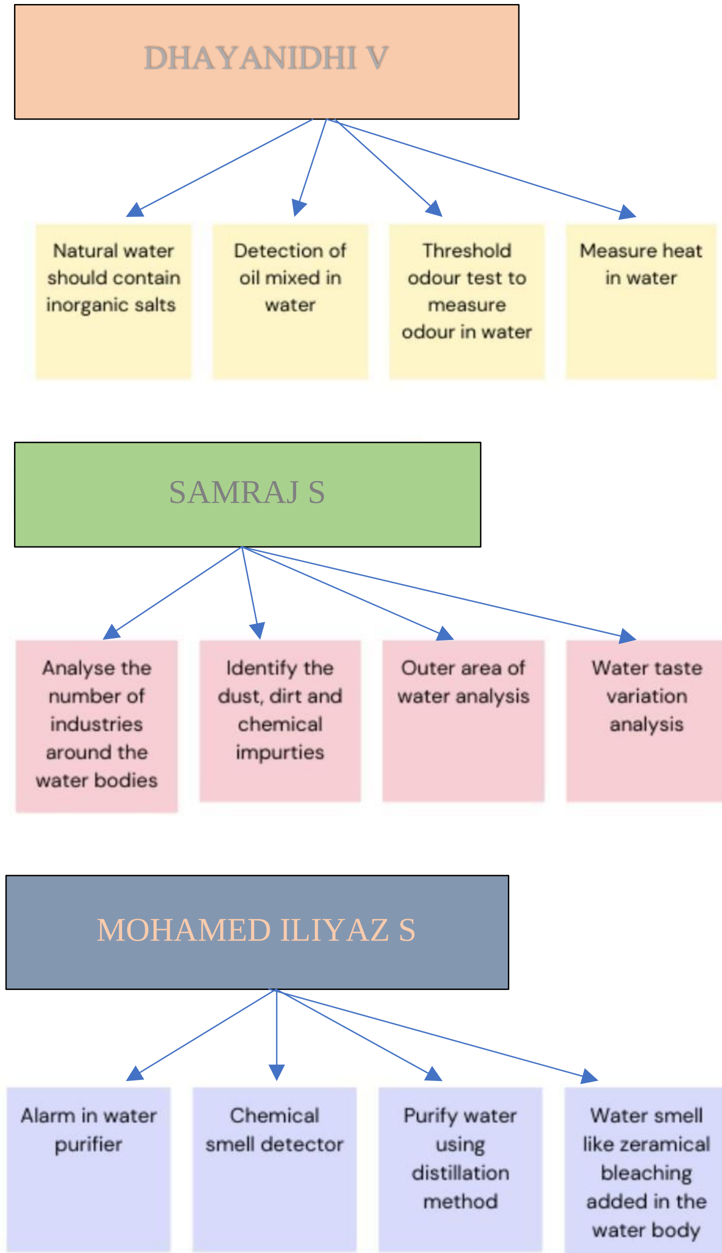
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming





3.3 Proposed Solution

S.No	Parameter	Description

1.	Problem Statement	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.
2.	Idea / Solution description	The solution is derived from the data sets by comparing the accuracy rate with the previous data set and the current data set.
3.	Novelty / Uniqueness	Using ML techniques (Regression models) to predict the quality of water instead of using physical measurements or sensors to obtain the quality of water. ML techniques improves the accuracy of measurement over existing chemical and physical techniques as it is infeasible to obtain all the required features to predict the water quality. Physical and chemical measurements may lead to the usage of expensive instruments and also take a lot of time. ML techniques make the process easier, feasible and faster.
4.	Social Impact /Customer Satisfaction	Our intended audience consists of people who are concerned about the quality of water they drink. Water's health is more important which should be considered as many water-borne diseases are more widely known. The proposed solution will help in identifying water pollution and helps the customer to drink healthy water.

5.	Business Model (Revenue Model)	Industries that provide sanitation facilities and products (like water purifiers, quality testers etc.) can deploy this solution to provide more waste water treatment plants, better insights in health concerns and there may also be an increase in awareness and demand for better water quality testing and
		availability. People will start looking for treatments related to water-borne diseases as the awareness increases
6.	Scalability of the Solution	The solution proposed will be deployed as a web application. So, it is easily accessible by anyone who has internet services and has no specific software and hardware specifications

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns.

Purpose:

- Customer needs to know about water's parameters such as pH, nitrate content so that it can be given to the ML model to predict the quality of water.
- User uses various experimental techniques like analyzing the quantity of chemical present and also analyzes physical properties of the water.
- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration 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: early warning/forecast monitoring.
FR-4	Data handling	File contains water quality metrics for different water Bodies.
FR-5	Quality analysis	Analyze with the acquired information of the water across various water quality indicator like (PH, Turbidity TDS Temperature) using different model.
FR-6	Model Prediction	Confirming based on water quality index and shows the machine learning prediction (Good, Partially Good, Poor) with the percentage of presence of various parameter.
FR-7	Remote Visualization	Visualization through charts based on present and 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.

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system provides a natural interaction with the users. Accurate water quality prediction with short time analysis and provide prediction safe to drink or not using some parameters and provide a great significance for water environment protection.
NFR-2	**Security**	The model enables with 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 period of 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 GB RAM.
NFR-5	**Availability**	The system should be available for the duration of the user access the system until the user terminate the access. The system response to request of the user in less time and the recovery is done is 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.

5. PROJECT DESIGN

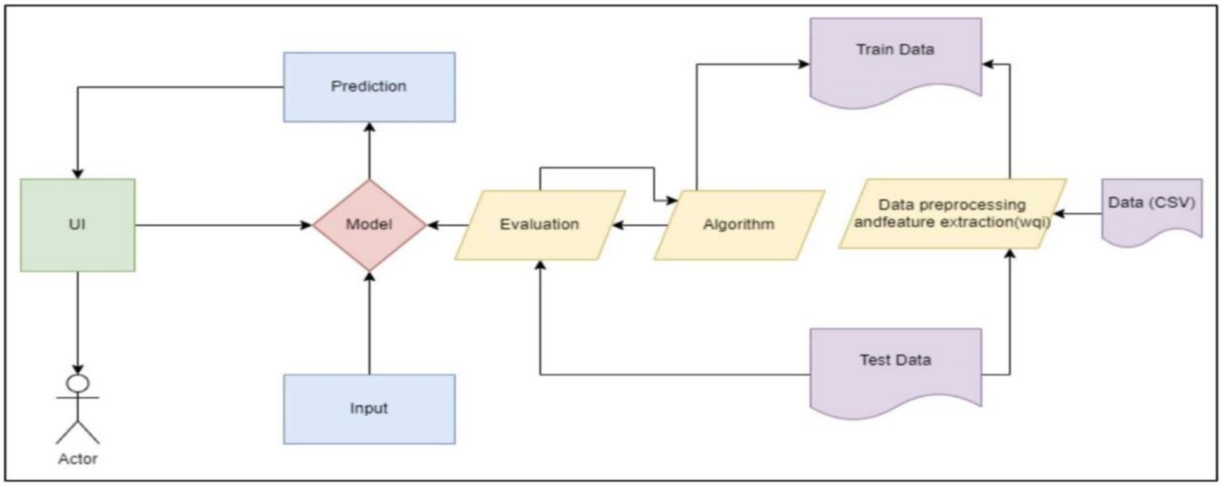
5.1 Data Flow Diagrams

5. PROJECT DESIGN

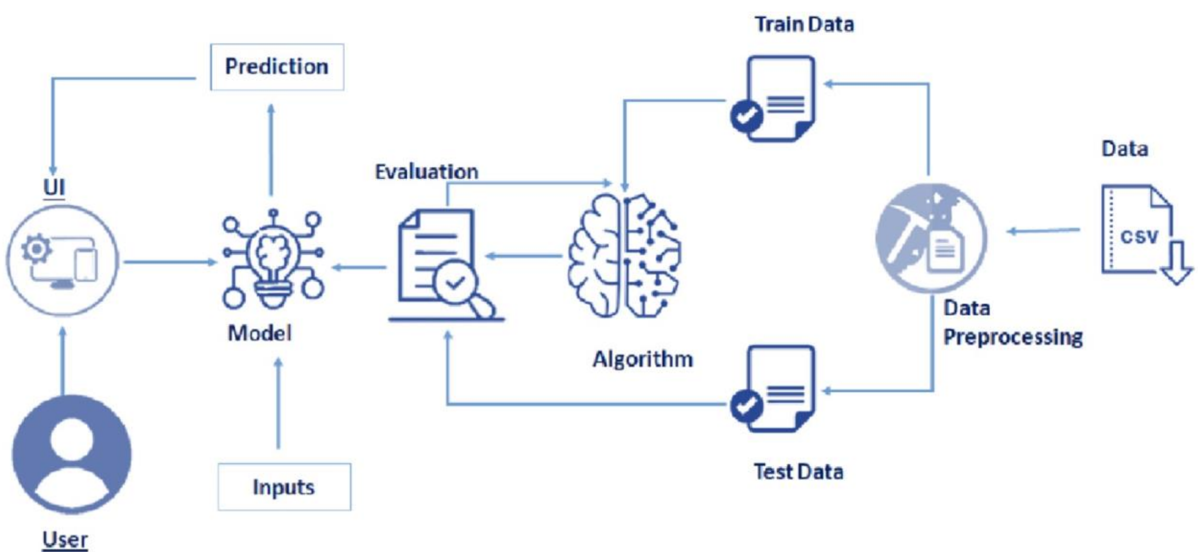
5.1 Data Flow Diagrams

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-I	Data Preparation	USN-I	Collecting water dataset and pre-processing it	20	High	Dhayanidhi V Aswin Kumar I
Sprint-2	Model Building	USN-2	Create an ML model to predict water quality	5	Medium	Aswin Kumar I Dhayanidhi V Samraj S Mohamed Iliyaz S
Sprint-2	Model Evaluation	USN-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	
Sprint-2	Model Deployment	USN-4	As a user, I need to deploy the model and need to find the results.	10	Medium	
Sprint-3	Web page (Form)	USN-5	Asa user, I can use the application by entering the water dataset to analyze or predict the results.	20	Medium	Dhayanidhi V
Sprint-4	Dashboard	USN-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	Aswin Kumar I Samraj S Mohamed Iliyaz S

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-I	Data Preparation	USN-I	Collecting water dataset and pre-processing it	20	High	Dhayanidhi V Aswin Kumar I
Sprint-2	Model Building	USN-2	Create an ML model to predict water quality	5	Medium	Aswin Kumar I Dhayanidhi V Samraj S Mohamed Iliyaz S
Sprint-2	Model Evaluation	USN-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	
Sprint-2	Model Deployment	USN-4	As a user, I need to deploy the model and need to find the results.	10	Medium	
Sprint-3	Web page (Form)	USN-5	Asa user, I can use the application by entering the water dataset to analyze or predict the results.	20	Medium	Dhayanidhi V
Sprint-4	Dashboard	USN-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	Aswin Kumar I Samraj S Mohamed Iliyaz S

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Completed Points	Sprint Release Date
Sprint-1	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	7 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-3	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022
Sprint-4	20	8 Days	21 Nov 2022	25 Nov 2022	20	25 Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1

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

Reading Dataset

In [ ]:

In [2]:
```python
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='XASQkNEL212fp8ybFcqZgV8bG9ErwvLqzJFEzwDfuFa3',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'datascience-donotdelete-pr-otpznaf0icrijh'
object_key = 'water_data1.txt'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

df=pd.read_csv(streaming_body_1)
df
```

Out[2]:

	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

Analyse the data

In [3]:
```python
df.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

In [4]:
```python
df.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

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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
```

```
 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
```

n [6]:
```
df.shape
```

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

Handling Missing Values

n [7]:
```
df.isnull().any()
```

ut[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
```

n [8]:
```
df.isnull().sum()
```

ut[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
```

In [9]:
```
df.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
```

In [10]:
```
df['Temp']=pd.to_numeric(df['Temp'],errors='coerce')
df['D.O. (mg/l)']=pd.to_numeric(df['D.O. (mg/l)'],errors='coerce')
df['PH']=pd.to_numeric(df['PH'],errors='coerce')
df['B.O.D. (mg/l)']=pd.to_numeric(df['B.O.D. (mg/l)'],errors='coerce')
df['CONDUCTIVITY (µmhos/cm)']=pd.to_numeric(df['CONDUCTIVITY (µmhos/cm)'],errors='coerce')
df['NITRATENAN N+ NITRITENANN (mg/l)']=pd.to_numeric(df['NITRATENAN N+ NITRITENANN (mg/l)'],errors='coerce')
df['TOTAL COLIFORM (MPN/100ml)Mean']=pd.to_numeric(df['TOTAL COLIFORM (MPN/100ml)Mean'],errors='coerce')
df.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
```

```
In [11]: df.isnull().sum()
```

```
Out[11]: STATION CODE                         0
LOCATIONS                            0
STATE                                0
Temp                                92
D.O. (mg/l)                         31
PH                                   8
CONDUCTIVITY (µmhos/cm)             25
B.O.D. (mg/l)                       43
NITRATENAN N+ NITRITENANN (mg/l)   225
FECAL COLIFORM (MPN/100ml)           0
TOTAL COLIFORM (MPN/100ml)Mean     132
year                                 0
dtype: int64
```

```
In [12]: df['Temp'].fillna(df['Temp'].mean(),inplace=True)
df['D.O. (mg/l)'].fillna(df['D.O. (mg/l)'].mean(),inplace=True)
df['PH'].fillna(df['PH'].mean(),inplace=True)
df['CONDUCTIVITY (µmhos/cm)'].fillna(df['CONDUCTIVITY (µmhos/cm)'].mean(),inplace=True)
df['B.O.D. (mg/l)'].fillna(df['B.O.D. (mg/l)'].mean(),inplace=True)
df['NITRATENAN N+ NITRITENANN (mg/l)'].fillna(df['NITRATENAN N+ NITRITENANN (mg/l)'].mean(),inplace=True)
df['TOTAL COLIFORM (MPN/100ml)Mean'].fillna(df['TOTAL COLIFORM (MPN/100ml)Mean'].mean(),inplace=True)
```

```
In [13]: df.drop(["FECAL COLIFORM (MPN/100ml)"],axis=1,inplace=True)
```

```
In [14]: df=df.rename(columns = {'D.O. (mg/l)': 'do'})
df=df.rename(columns = {'CONDUCTIVITY (µmhos/cm)': 'co'})
df=df.rename(columns = {'B.O.D. (mg/l)': 'bod'})
df=df.rename(columns = {'NITRATENAN N+ NITRITENANN (mg/l)': 'na'})
df=df.rename(columns = {'TOTAL COLIFORM (MPN/100ml)Mean': 'tc'})
df=df.rename(columns = {'STATION CODE': 'station'})
df=df.rename(columns = {'LOCATIONS': 'location'})
df=df.rename(columns = {'STATE': 'state'})
df=df.rename(columns = {'PH': 'ph'})
```

Water Quality Index (WQI) Calculation

```
In [15]: #calculation of pH
df['npH']=df.ph.apply(lambda x: (100 if(8.5>=x>=7)
                                 else(80 if(8.6>=x>=8.5) or (6.9>=x>=6.8)
                                   else (60 if(8.8>=x>=8.6) or (6.8>=x>=6.7)
                                       else(40 if(9>=x>=8.8) or (6.7>=x>=6.5)
                                           else 0)))))
```

```
In [16]: #calculation of dissolved oxygen
df['ndo']=df.do.apply(lambda x: (100 if(x>=6)
                                 else(80 if(6>=x>=5.1)
                                   else (60 if(5>=x>=4.1)
                                       else(40 if(4>=x>=3)
                                           else 0)))))
```

```
In [17]: #calculation of total coliform
df['nco']=df.tc.apply(lambda x: (100 if(5>=x>=0)
                                 else(80 if(50>=x>=5)
                                   else (60 if(500>=x>=50)
                                       else(40 if(10000>=x>=500)
                                           else 0)))))
```

```
In [18]: #calculation of B.D.O
df['nbdo']=df.bod.apply(lambda x:(100 if(3>=x>=0)
                                 else(80 if(6>=x>=3)
                                   else (60 if(80>=x>=6)
                                       else(40 if(125>=x>=80)
                                           else 0)))))
```

```
In [19]: #calculation of electric conductivity
df['nec']=df.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 [20]: #calculation of nitrate
df['nna']=df.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 [20]: #calculation of nitrate
df['nna']=df.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)))))
```

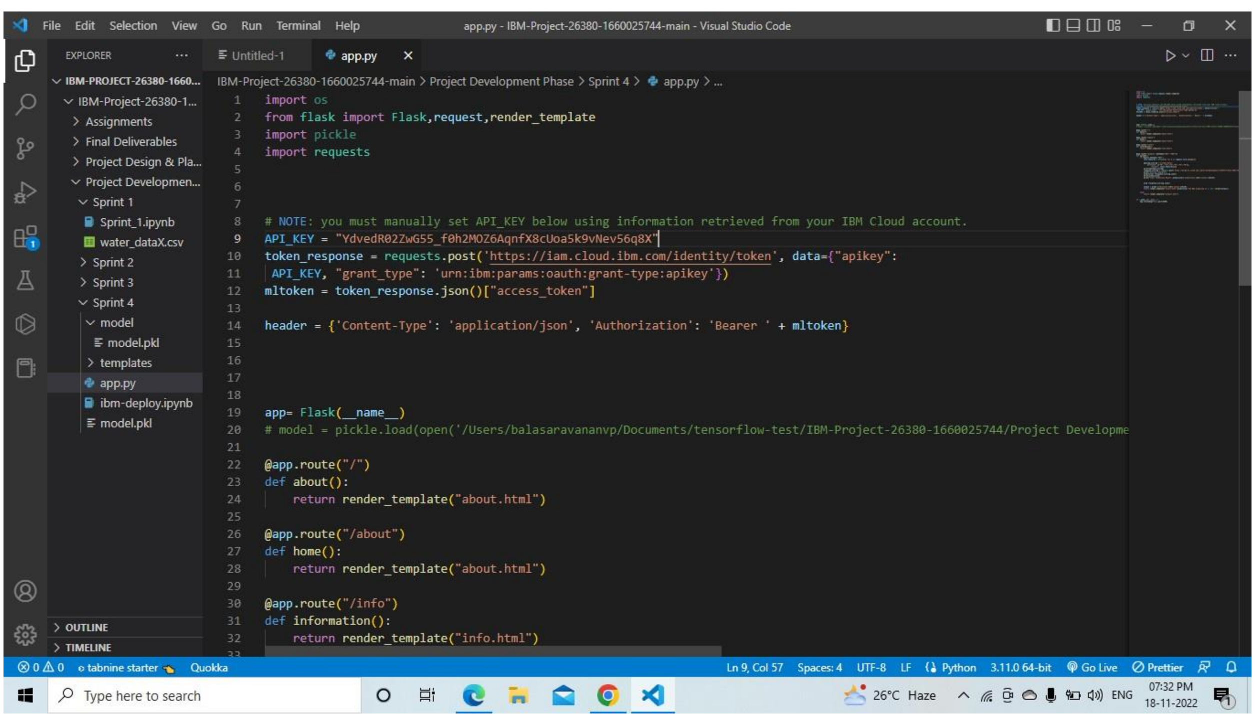
```
In [21]: #Calculation of Water Quality Index WQI
df['wph']=df.npH*0.165
df['wdo']=df.ndo*0.281
df['wbdo']=df.nbdo*0.234
df['wec']=df.nec*0.009
df['wna']=df.nna*0.028
df['wco']=df.nco*0.281
df['wqi']=df.wph+df.wdo+df.wbdo+df.wec+df.wna+df.wco
df
```

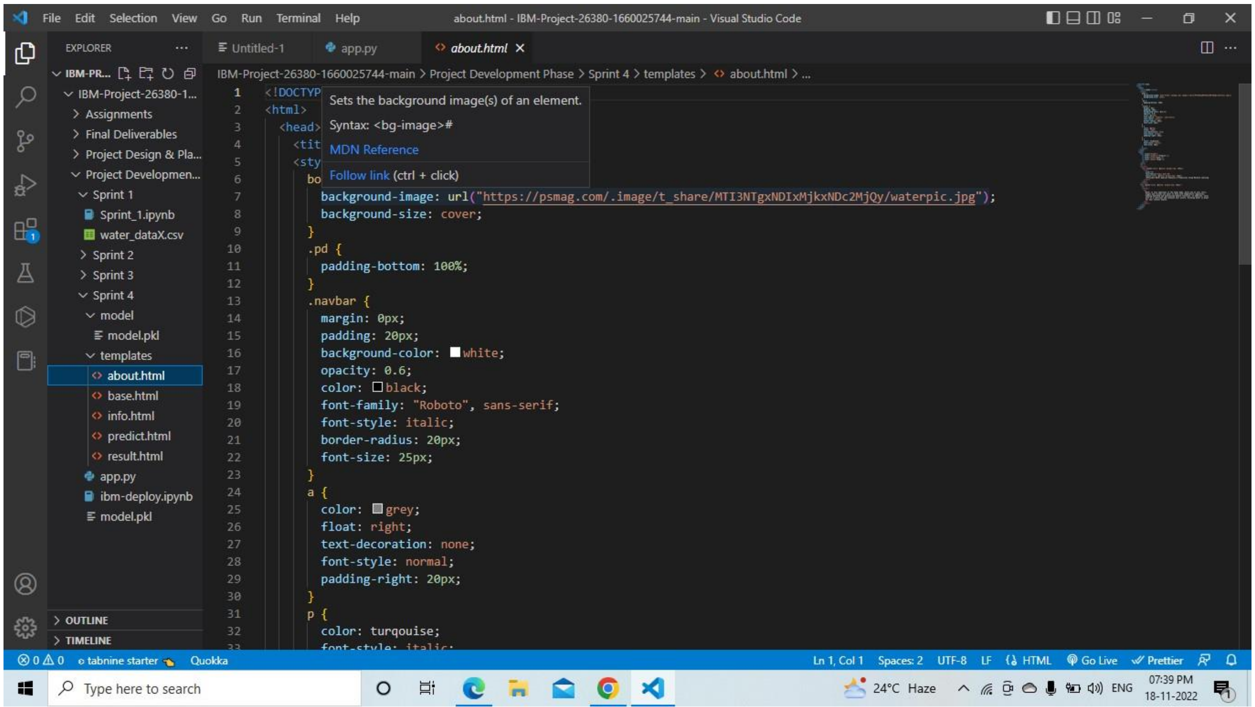
Out[21]:

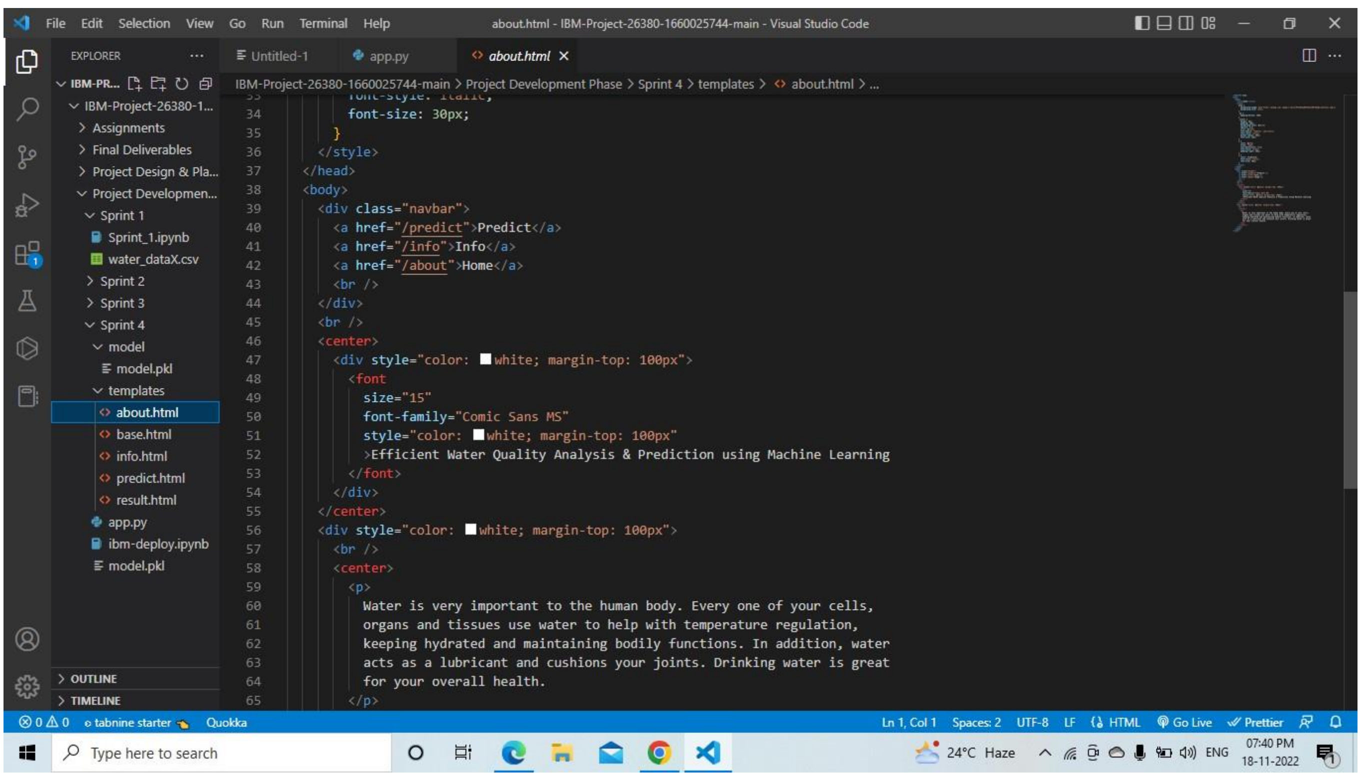
	station	location	state	Temp	do	ph	co	bod	na	tc	...	nbdo	nec	nna	wph	wdo	wbdo	wec	wna	wco	wqi
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.600000	6.7	7.5	203.0	6.940049	0.100000	27.0	...	60	60	100	16.5	28.10	14.04	0.54	2.8	22.48	84.46
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	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

7.2 Feature 2




File Edit Selection View Go Run Terminal Help
about.html - IBM-Project-26380-1660025744-main - Visual Studio Code
EXPLORER
Untitled-1
app.py
about.html
IBM-Project-26380-1660025744-main > Project Development Phase > Sprint 4 > templates > about.html > ...
IBM-Project-26380-1...
Assignments
Final Deliverables
Project Design & Pla...
Project Developmen...
Sprint 1
Sprint_1.ipynb
water_dataX.csv
Sprint 2
Sprint 3
Sprint 4
model
model.pkl
templates
about.html
base.html
info.html
predict.html
result.html
app.py
ibm-deploy.ipynb
model.pkl
OUTLINE
TIMELINE
Sets the background image(s) of an element.
Syntax: <bg-image>#
MDN Reference
Follow link (ctrl + click)
<!DOCTYP
<html>
<head>
<tit
<sty
bo
background-image: url("https://psmag.com/.image/t_share/MTI3NTgxNDIxMjkxNDc2MjQy/waterpic.jpg");
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: turquoise;
Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML Go Live Prettier
tabnine starter Quokka
Type here to search
24°C Haze ENG 07:39 PM 18-11-2022

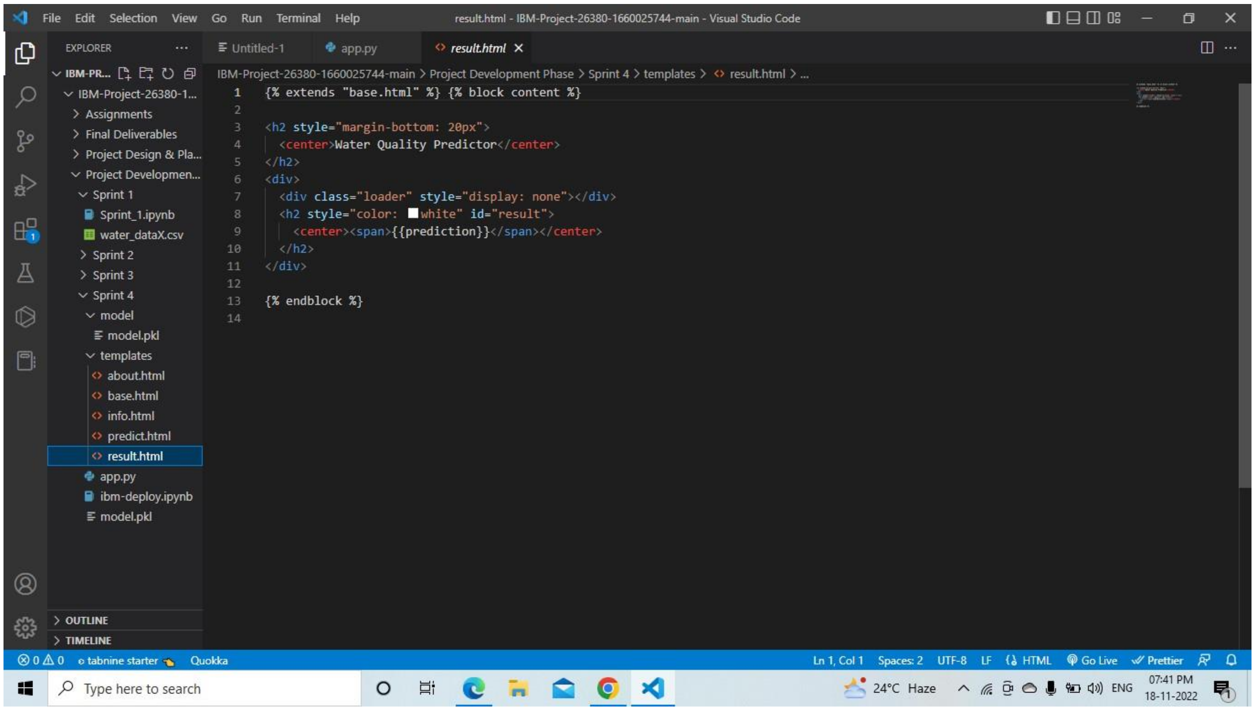

File Edit Selection View Go Run Terminal Help
about.html - IBM-Project-26380-1660025744-main - Visual Studio Code
EXPLORER
Untitled-1
app.py
about.html
IBM-Project-26380-1660025744-main > Project Development Phase > Sprint 4 > templates > about.html > ...
font-size: 30px;
}
</style>
</head>
<body>
<div class="navbar">
Predict
Info
Home

</div>

<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

</div>
</center>
<div style="color: white; margin-top: 100px">

<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>
Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML Go Live Prettier
tabnine starter Quokka
Type here to search
24°C Haze ENG 07:40 PM 18-11-2022



8. TESTING

8.1 Test Cases

Test case ID	Feature	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	comments	TC for Automation	BU ID	Executed By
IndexPage_ TC 001	UI	Index Page	Verify the UI elements in Index	I.Enter the localhost url and click go.	127.0.0.1 .500	Application should show below UI elements. 1.Title of the project. 2.Description Of the project.	working expected	PASS	Successful	Y		Dhayanidhi V Aswin Kumar I Mohamed Iliyaz S

IndexPage _TC 002	UI	Index Page	Verify the user able to navigate into the predict page	the localhost url and click	0.0.1 .500 0	User should navigate to predict page	expected	PASS	Successful	Y		Dhayanidhi V Aswin Kumar I Samraj S

PredictPage TC 003	UI	Predict Page	Verify the UI elements in Predict Page	1. .Enter the localhost url and click go. 2. Click on Want to predict	127. 0.0.1 soo	Application should show below UI elements: I Enter the data input 2.Check the predict	expected	PASS	Successful	Y		Aswin Kumar I Dhayanidhi V
PredictPage TC 004	Functional	Predict page	Verify user is able to give input in the form	1. Enter the localhost url and click go. 2.C1ick predict 3. Enter the values	127. 0.0. I 0	User should able to give input textbox	Working as expected	PASS	Successful	Y		Mohamed Iliyaz S Samraj S
PredictPage TC 005	UI	Predict Page	Verify users are able to see the result text When clicking on the predict button.	I.Enter the localhost url and click go. 2.Click predict button 3.Enter input data 4. click on the predict button. 4.Click on the predict button.	127. 0.0.1 .500	Users should be able to predict the quality predicted value is XX WQI text.	Working as expected	PASS	Successful	Y		Dhayanidhi V Aswin Kumar I Samraj S Mohamed Iliyaz s

8.2 User Acceptance Testing

Purpose of User Acceptance Testing

The purpose of this document is to briefly explain the test coverage and open issues of the [Efficient Water Quality Analysis & Prediction using Machine Learning] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Severity 5	Subtotal
By Design	1	1	1	0	0	3
Duplicate	1	0	1	0	0	2
External	1	0	0	1	0	2
Fixed	2	1	0	0	0	3
Not Reproduced	0	0	0	0	0	0
Skipped	0	0	0	0	0	0

Test Case Analysis

Shows the number of test cases that have passed, failed, and untested

Section	Total cases	Not Tested	Fail	Pass
Index Page	2	0	0	2
Predict Page	8	0	0	8

9. RESULTS

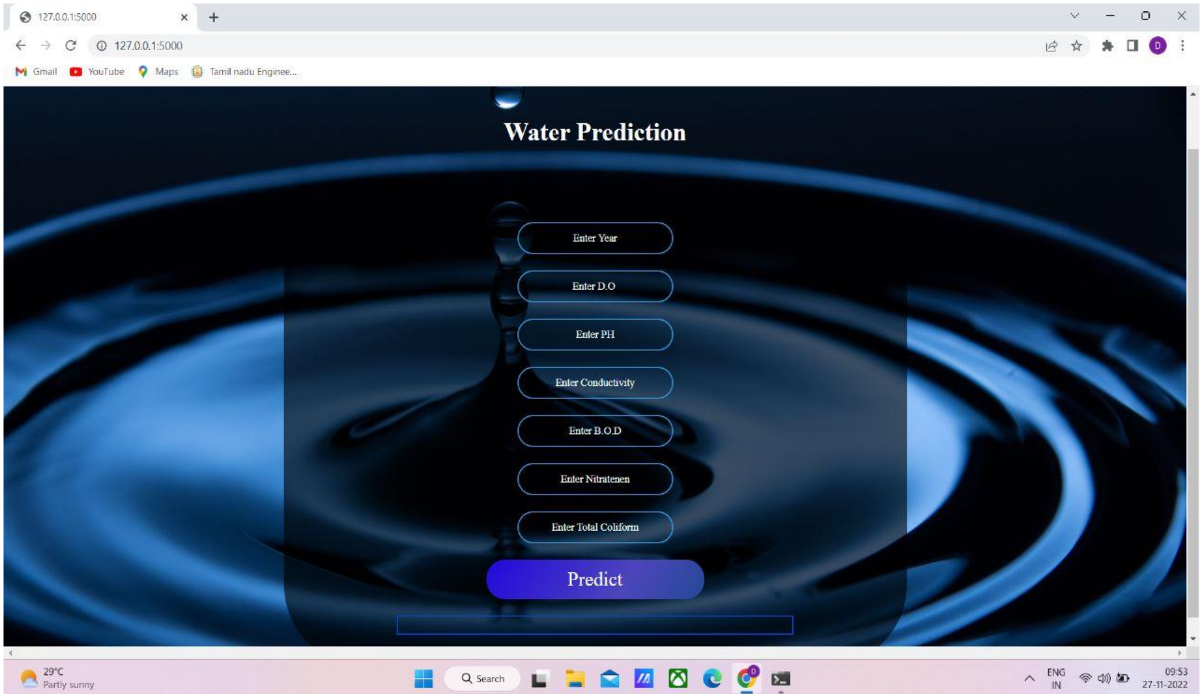
9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot

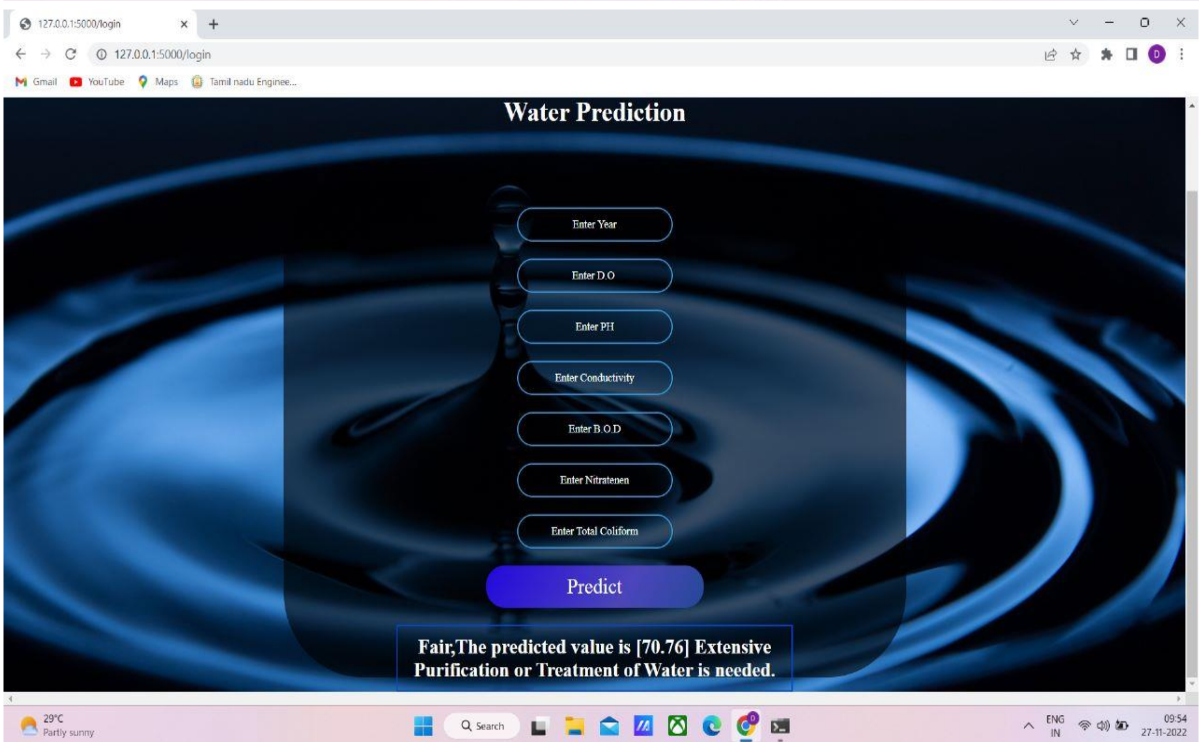
1.	Metrics	**Regression Model:**
		MAE : 0.987
		
		MSE : 5.55
		
		RMSE : 2.35
		
		R2 score: 0.96
			`In [47]: from sklearn import metrics`
			`print('MAE:',metrics.mean_absolute_error(y_test,y_pred))`
			`print('MSE:',metrics.mean_squared_error(y_test,y_pred))`
			`print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred)))`
			`MAE: 0.9872080200501312`
			`MSE: 5.555095879699248`
			`RMSE: 2.3569250899634566`
			`In [48]: metrics.r2_score(y_test, y_pred)`
			`Out[48]: 0.96971918125809`

2.	Tune the Model	Hyperparameter Tuning –
		n_estimators = 10,
			`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)`

9.2 Output



127.0.0.1:5000
Water Prediction
Enter Year
Enter D.O
Enter PH
Enter Conductivity
Enter B.O.D
Enter Nitratenen
Enter Total Coliform
Predict



127.0.0.1:5000/login
Water Prediction
Enter Year
Enter D.O
Enter PH
Enter Conductivity
Enter B.O.D
Enter Nitratenen
Enter Total Coliform
Predict
Fair,The predicted value is [70.76] Extensive Purification or Treatment of Water is needed.

GITHUB: https://github.com/IBM-EPBL/IBM-Project-35789-1660288729

DEMO LINK:
https://drive.google.com/file/d/1f7nvlgc8UP1Fteuq2ef4xxzAnGcygEm/view?usp=drivesdk