TABLE OF CONTENTS

1.INTRODUCTION

Water is the most important of sources, vital for sustaining all kinds of life; however, it is in constant threat of pollution by life itself. Water is one of the most communicable mediums with a far reach. Rapid industrialization has consequently led to deterioration of quality at an alarming rate. Poor water quality results have been known to be one of the major factors of escalation of harrowing diseases. As reported, in developing countries, 80% of the diseases are water borne diseases, which have led to 5 million deaths and 2.5 billion illnesses. The most common of these diseases in Pakistan are diarrhea, typhoid, gastroenteritis, cryptosporidium infections, some forms of hepatitis and giardiasis intestinal worms. In Pakistan, water borne diseases, cause a GDP loss of 0.6–1.44% every year.

This makes it a pressing problem, particularly in a developing country like Pakistan. Water quality is currently estimated through expensive and time-consuming lab and statistical analyses, which require sample collection, transport to labs, and a considerable amount of time and calculation, which is quite ineffective given water is quite a communicable medium and time is of the essence if water is polluted with disease-inducing waste. The horrific consequences of water pollution necessitate a quicker and cheaper alternative.

In this regard, the main motivation in this study is to propose and evaluate an alternative method based on supervised machine learning for the efficient prediction of water quality in real-time.

2. LITERATURE REVIEW

This research explores the methodologies that have been employed to help solve problems related to water quality. Typically, conventional lab analysis and statistical analysis are used in research to aid in determining water quality, while some analyses employ machine learning methodologies to assist in finding an optimized solution for the water quality problem. Local research employing lab analysis helped us gain a greater insight into the water quality problem in Pakistan. In one such research study, Daud et al. gathered water samples from different areas of Pakistan and tested them against different parameters using a manual lab analysis and found a high presence of E. coli and fecal coliform due to industrial and sewerage waste. Alamgir et al.Tested 46 different samples from Orangi town, Karachi, using manual lab analysis and found them to be high in sulphates and total fecal coliform count.

After getting familiar with the water quality research concerning Pakistan, I explored research employing machine learning methodologies in the realm of water quality. When it comes to estimating water quality using machine learning, Shafi et al.Estimated water quality using classical machine learning algorithms namely, Support Vector Machines (SVM), Neural Networks (NN), Deep Neural Networks (Deep NN) and k Nearest Neighbors (kNN), with the highest accuracy of 93% with Deep NN. The estimated water quality in their work is based on only three parameters: turbidity, temperature and pH, which are tested according to World Health Organization (WHO) standards (Available online at URL https://www.who.int/airpollution/guidelines/en/). Using only three parameters and comparing them to standardized values is quite a limitation when predicting water quality. Ahmad et al. Employed single feed forward neural networks and a combination of multiple neural networks to estimate the WQI. They used 25 water quality parameters as the input. Using a combination of backward elimination and forward selection selective combination methods, they achieved an R2 and MSE of 0.9270, 0.9390 and 0.1200, 0.1158, respectively. The use of 25 parameters makes their solution a little immoderate in terms of an inexpensive real time system, given the price of the parameter sensors. Sakizadeh predicted the WQI using 16 water quality parameters and ANN with Bayesian regularization. His study yielded correlation coefficients between the observed and predicted values of 0.94 and 0.77, respectively. Abyaneh predicted the chemical oxygen demand (COD) and the biochemical oxygen demand (BOD) using two conventional machine learning methodologies namely, ANN and multivariate linear regression.

They used four parameters, namely pH, temperature, total suspended solids (TSS) and total suspended (TS) to predict the COD and BOD. Ali and Qamar used the unsupervised technique of the average linkage (within groups) method of hierarchical clustering to classify samples into water quality classes. However, they ignored the major parameters associated with WQI during the learning process and they did not use any standardized water quality index to evaluate their predictions. Gazzaz et al. Used ANN to predict the WQI with a model explaining almost 99.5% of variation in the data. They used 23 parameters to predict the WQI, which turns out to be quite expensive if one is to use it for an IoT system, given the prices of the sensors. Rankovic et al. Predicted the dissolved oxygen (DO) using a feedforward neural network (FNN). They used 10 parameters to predict the DO, which again defeats the purpose if it has to be used for a real-time WQI estimation with an IoT system.

Most of the research either employed manual lab analysis, not estimating the water quality index standard, or used too many parameters to be efficient enough.

2. SYSTEM REQUIREMENTS

2.1 Hardware and Software requirements

Software Configuration:
Operating System : Windows 10
Front End : FLASK Back end : MongoDB Server : FLASK
Language : HTML, CSS, Python
Browser : Google Chrome

Hardware Configuration:
Processor : Pentium III or higher
RAM : 128 MB or More
Hard Disk : 20 GB or More

2.2 Tools

FLASK: Fla3sk is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

SCIKIT-LEARN: Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib

3. THEORETICAL ANALYSIS

3.1 Data Preprocessing

The data used for this research was obtained from water_dataX.csv and it was cleaned by performing a box plot analysis, discussed in this section. After the data were cleaned, they were normalized using q-value normalization to convert them to the range of 0–100 to calculate the WQI using six available parameters. The complete procedure is detailed next.

3.2 Q-Value Normalization

Q-value normalization was used to normalize the parameters, particularly the water quality parameters to fit them in the range of 0 to 100 for easier index calculation

We used them to convert five of these parameters within the range of 0 to 100 [14,15]. For the sixth parameter, namely nitrites, due to unavailability of its q-value ranges, we used the WHO standards to distinctly convert them to the 0–100 range by means of a set of thresholds as follows: assigning 100 if its below 1, 80 if its below 2, 50 if its below 3 and 0 if its greater than 3, reflecting strict penalization. Once the values were q-normalized and were in the range of 0–100, they were used for calculating the WQI of the dataset using

3.3 Water Qualiity Index (WQI)

Water quality index (WQI) is the singular measure that indicates the quality of water and it is calculated using various parameters that are truly reflective of the water's quality. To conventionally calculate the WQI, nine water quality parameters are used, but if we did not have all of them, we could still estimate the water quality index with at least six defined parameters. We had five parameters, namely fecal coliform, pH, temperature, turbidity and total dissolved solids in our dataset. We also considered nitrites as the sixth parameter as the weight and relative importance of nitrites in the WQI calculation is stated to be equal to that of nitrates in multiple WQI studies. Using these parameters and their assigned weightages, we calculated the WQI of each sample, where qvalue reflects the value of a parameter in the range of 0–100 and w_ f actor represents the weight of a particular parameter. 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

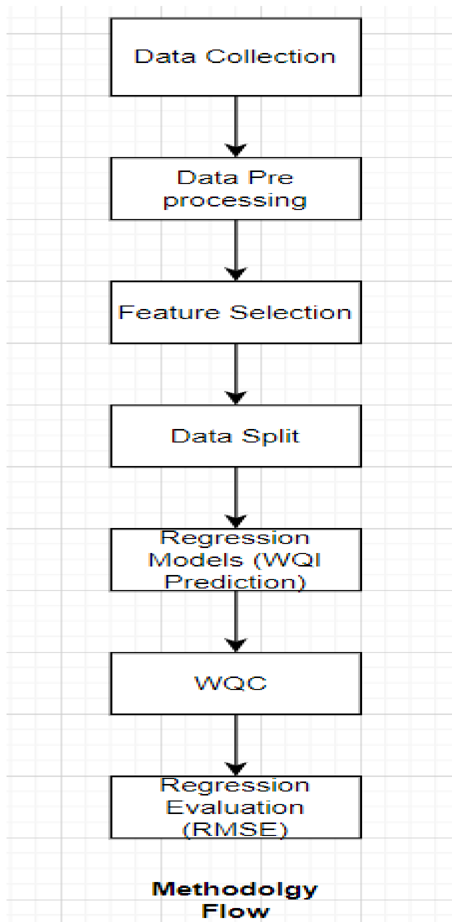
WQI = (∑qvalue × w_ f actor) / (∑w_ f actor)

3.4. Water Qulaity Class (WQC)

Once we had estimated the WQI, we defined the water quality class (WQC) of each sample using the WQI in classification algorithms.

Water Quality Index Range	Class
0–25	Very bad
25–50	Bad
50–70	Medium
70–90	Good
90–100	Excellent

4. FLOW CHART



5. RESULTS

While employing the regression algorithms, we found Linear Regression, gradient descent, having a RMSE of 1.1987, to be the most efficient algorithm.
WQC is predicted based on Water Quality Index Range Class.

6. DISCUSSION

Water Quality is conventionally calculated using water quality parameters, which are acquired through time consuming lab analysis. We explored alternative methods of machine learning to estimate it and found several studies employing them. These studies used more than 10 water quality parameters to predict WQI. Ahmad et al. used 25 input parameters, Sakizadeh used 16 parameters, Gazzaz et al. used 23 input parameters in their methodology, and Rankovic et al. used 10 input parameters, which is unsuitable for inexpensive real time systems. Whereas, our methodology employs only four water quality parameters to predict WQI, with a MAE of 1.96, and to predict water quality class with an accuracy of 85%. Our results make a base for an inexpensive real time water quality detection system, while other studies, although they use machine learning, use too many parameters to be incorporated in real time systems.

7. CONCLUSIONS AND FUTURE WORK

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.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. 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 and TDS 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 makers.

9. APPENDIX

9.1 Source Code

app.py

```
from flask import Flask, request, render_template, redirect, url_for
app = Flask(__name__)

@app.route('/success/<name>')
def wqi(name):
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    filename="E://Internship//water_dataX.csv"
    data = pd.read_csv(filename, encoding= 'unicode_escape')
    print(data.head())

    #conversions
    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')
    print(data.dtypes)

    #initialization to remove column  FECAL COLIFORM (MPN/100ml)  and RENAME the
features with simpler names
    start=2
    end=1779
    station=data.iloc [start:end ,0]
    location=data.iloc [start:end ,1]
```

```
state=data.iloc [start:end ,2]
do= data.iloc [start:end ,4].astype(np.float64)
value=0
ph = data.iloc[ start:end,5]
co = data.iloc [start:end ,6].astype(np.float64)

year=data.iloc[start:end,11]
tc=data.iloc [2:end ,10].astype(np.float64)


bod = data.iloc [start:end ,7].astype(np.float64)
na= data.iloc [start:end ,8].astype(np.float64)
print(na.dtype)

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

#adding new Parameters according to their “WHO” standard limits from range(0-100)
for WQI calculations(q_value).
#Q VALUE NORMALIZATION


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

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

```
data['nco']=data.tc.apply(lambda x:(100 if (5>=x>=0)
                         else(80 if  (50>=x>=5)
                             else(60 if (500>=x>=50)
                                 else(40 if (10000>=x>=500)
                                     else 0)))))
#calc of B.O.D
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
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)))))
#Calulation of nitrate
data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)
                         else(80 if  (50>=x>=20)
                             else(60 if (100>=x>=50)
                                 else(40 if (200>=x>=100)
                                     else 0)))))

data.head()
print(data.dtypes)


#qvalue reflects the value of a parameter in the range of 0–100 and w_ f actor
represents
#the weight of a particular parameter as listed in Table 2. 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
```

```
#we add new columns now by multiplying values to weights
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
print(data.head())

#calculation overall wqi for each year
ag=data.groupby('year')['wqi'].mean()

data=ag.reset_index(level=0,inplace=False)
print(data)

data = data[np.isfinite(data['wqi'])]
print(data.head())

#scatter plot of data points
cols =['year']
y = data['wqi']
x=data[cols]

plt.scatter(x,y)
#plt.show()

import matplotlib.pyplot as plt
data=data.set_index('year')
data.plot(figsize=(15,6))
#plt.show()

from sklearn import neighbors,datasets
data=data.reset_index(level=0,inplace=False)
data
```

```
#using linear regression to predict
from sklearn import linear_model
from sklearn.model_selection import train_test_split

cols =['year']
y = data['wqi']
x=data[cols]
reg=linear_model.LinearRegression()
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=4)
reg.fit(x_train,y_train)

a=reg.predict(x_test)
print(a)

print(y_test)

from sklearn.metrics import mean_squared_error
print('mse:%.2f'%mean_squared_error(y_test,a))

dt = pd.DataFrame({'Actual': y_test, 'Predicted': a})


print(x_test)

x = (x - x.mean()) / x.std()
x = np.c_[np.ones(x.shape[0]), x]
print(x)

alpha = 0.1 #Step size
iterations = 3000 #No. of iterations
m = y.size #No. of data points
np.random.seed(4) #Setting the seed
theta = np.random.rand(2) #Picking some random values to start with

def gradient_descent(x, y, theta, iterations, alpha):
    past_costs = []
    past_thetas = [theta]
```

```
    for i in range(iterations):
        prediction = np.dot(x, theta)
        error = prediction - y
        cost = 1/(2*m) * np.dot(error.T, error)
        past_costs.append(cost)
        theta = theta - (alpha * (1/m) * np.dot(x.T, error))
        past_thetas.append(theta)

    return past_thetas, past_costs

past_thetas, past_costs = gradient_descent(x, y, theta, iterations, alpha)
theta = past_thetas[-1]

#Print the results...
print("Gradient Descent: {:.2f}, {:.2f}".format(theta[0], theta[1]))

print(x_test)
print(y_test)

#prediction of january(2013-2015) across india
import numpy as np
newB=[74.76, 2.13]

def rmse(y,y_pred):
    rmse= np.sqrt(sum(y-y_pred))
    return rmse


y_pred=x.dot(newB)

dt = pd.DataFrame({'Actual': y, 'Predicted': y_pred})
dt=pd.concat([data, dt], axis=1)
print(dt)

#testing the accuracy of the model

from sklearn import metrics
```

```
    print(np.sqrt(metrics.mean_squared_error(y,y_pred)))

    #plotting the actual and predicted results
    x_axis=dt.year
    y_axis=dt.Actual
    y1_axis=dt.Predicted
    plt.scatter(x_axis,y_axis)
    #plt.plot(x_axis,y1_axis,color='r')
    plt.title("linear regression")


    print(name)
    name=float(name)
    xx=[[name]]
    res=reg.predict(xx)
    res

    waterquality=""
    if(res<=100 and res>=90):
      waterquality=".....EXCELLENT, fit to drink"
    elif(res<=89 and res>=85):
      waterquality=".....GOOD, fit to drink"
    elif(res<=84 and res>=80):
      waterquality=".....ACCEPTABLE, drink at your own risk"
    elif(res<=79 and res>=60):
      waterquality=".....BAD, Not fit to drink"
    elif(res<=59 and res>=19):
      waterquality=".....POOR, Not fit to drink, chances of you dying is high"



    #res=res+"  "+waterquality;
    return '<h2 style="color:steelblue">The Predicted WQI for the input year is %s' % res +
waterquality


@app.route('/',methods = ['POST', 'GET'])
```

```
def login():
    if request.method == 'POST':
        user = request.form['year']
        return redirect(url_for('wqi',name = user))
    else:
        return render_template('proj_water_quality.html')

if __name__ == '__main__':
    app.run(debug = True)
```

proj_water_quality.html

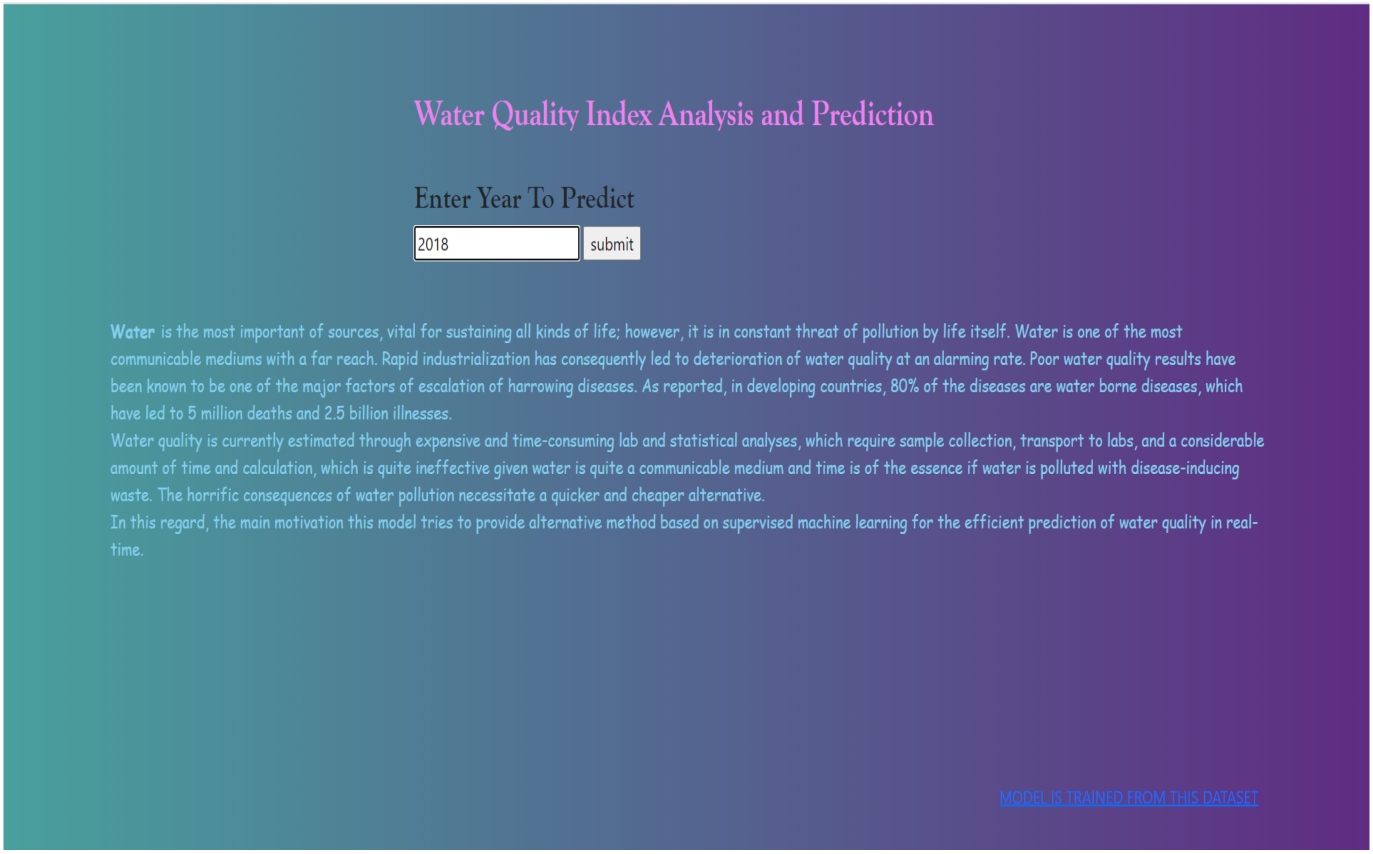
```
<html>
<head>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpb
zKgwra6" crossorigin="anonymous">

<style>
body {
        background: #5f2c82;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #49a09d, #5f2c82);  /* Chrome 10-25,
Safari 5.1-6 */
background: linear-gradient(to right, #49a09d, #5f2c82); /* W3C, IE 10+/ Edge, Firefox
16+, Chrome 26+, Opera 12+, Safari 7+ */
}

.zero {
margin-top:15%;
margin-left:77%
}
h2 {
font-family:"Goudy Old Style";
font-weight:bold;
padding-left:30%;
```

```
padding-top:5%;
color:violet
}

h3{
font-family:"Goudy Old Style";
padding-left:30%;
padding-top:2%;
font-weight:bold;

}
.one {
margin-left:30%;
}

.info {
font-family:"Comic Sans MS";
color:skyblue;
margin-top:50px;
}
</style>


</head>
<body>
	<form action="http://127.0.0.1:5000" method="post">
<h2> Water Quality Index Analysis and Prediction </h2>
<h3> Enter Year To Predict</h3>

<div class="one">
<input type="text" name="year" placeholder="YEAR" required="required" />
<input type="submit" value="submit">
</div>
</form>

<div class="container">
<p class="info">
```

```
<strong>Water </strong>is the most important of sources, vital for sustaining all kinds
of life; however, it is in
constant threat of pollution by life itself. Water is one of the most communicable
mediums with a far
reach. Rapid industrialization has consequently led to deterioration of water quality at
an alarming
rate. Poor water quality results have been known to be one of the major factors of
escalation of
harrowing diseases. As reported, in developing countries, 80% of the diseases are water
borne diseases,
which have led to 5 million deaths and 2.5 billion illnesses.
<br>
Water quality is currently estimated through expensive and time-consuming lab and
statistical
analyses, which require sample collection, transport to labs, and a considerable amount
of time and calculation, which is quite ineffective given water is quite a communicable
medium and time is of
the essence if water is polluted with disease-inducing waste. The horrific consequences
of water
pollution necessitate a quicker and cheaper alternative.
<br>
In this regard, the main motivation this model tries to provide alternative method
based on supervised machine learning for the efficient prediction of water quality in
real-time.
</p>
<p class="zero">
<a href="https://www.kaggle.com/anbarivan/indian-water-quality-data"> MODEL IS
TRAINED FROM THIS DATASET </a>  </p>
</div>
</body>
</html>
```

9.2 UI OUTPUT AND SCREENSHOT

Home Page



Output Screen

The Predicted WQI for the input year is [76.81176123].....BAD, Not fit to drink

10. REFERENCES

1. PCRWR. National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006); Pakistan Council of Research in Water Resources Islamabad: Islamabad, Pakistan, 2007. Available online: http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/Water%20Quality%20Monitoring%20Report%202005-06.pdf (accessed on 23 August 2019).
2. Mehmood, S.; Ahmad, A.; Ahmed, A.; Khalid, N.; Javed, T. Drinking Water Quality in Capital City of Pakistan. Open Access Sci. Rep. 2013, 2. [CrossRef]
3. PCRWR. Water Quality of Filtration Plants, Monitoring Report; PCRWR: Islamabad, Pakistan, 2010. Available online: http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/FILTRTAION%20PLANTS% 20REPOT-CDA.pdf (accessed on 23 August 2019).
4. Gazzaz, N.M.; Yusoff, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F. Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors. Mar. Pollut. Bull. 2012, 64, 2409–2420. [CrossRef]
5. Daud, M.K.; Nafees, M.; Ali, S.; Rizwan, M.; Bajwa, R.A.; Shakoor, M.B.; Arshad, M.U.; Chatha, S.A.S.; Deeba, F.; Murad, W.; et al. Drinking water quality status and contamination in Pakistan. BioMed Res. Int. 2017, 2017, 7908183. [CrossRef]
6. Alamgir, A.; Khan, M.N.A.; Hany; Shaukat, S.S.; Mehmood, K.; Ahmed, A.; Ali, S.J.; Ahmed, S. Public health quality of drinking water supply in Orangi town, Karachi, Pakistan. Bull. Environ. Pharmacol. Life Sci. 2015, 4, 88–94.
7. Shafi, U.; Mumtaz, R.; Anwar, H.; Qamar, A.M.; Khurshid, H. Surface Water Pollution Detection using Internet of Things. In Proceedings of the 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 8–10 October 2018; pp. 92–96.
8. Ahmad, Z.; Rahim, N.; Bahadori, A.; Zhang, J. Improving water quality index prediction in Perak River basin Malaysia through a combination of multiple neural networks. Int. J. River Basin Manag. 2017, 15, 79–87. [CrossRef]
9. Sakizadeh, M. Artificial intelligence for the prediction of water quality index in groundwater systems. Model. Earth Syst. Environ. 2016, 2, 8. [CrossRef]
10. Abyaneh, H.Z. Evaluation of multivariate linear regression and artificial neural networks in prediction of water quality parameters. J. Environ. Health Sci. Eng. 2014, 12, 40. [CrossRef].