Predicting And Analyzing Urban Water Quality With Machine Learning.

Developed by: **Madhuri, Niharika, Vishnu vardhan, Sathwik.**

Abstract:

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. Data analysis is one of the key engines of progress in most of areas of the research in natural sciences, including environmental sciences. Continuous development and technological progress provide us with universal and advanced tools for data analysis, such as machine learning algorithms. The main objective is to explore what kind of data is provided, determine the most important factors to check whether the water is portable or not, and select the most accurate model to suitable for prediction.

Introduction:

Our bodies use water in all the cells, organs, and tissues, to help regulate body temperature and maintain other bodily functions. Because our bodies lose water through breathing, sweating, and digestion, it's crucial to rehydrate and replace water by drinking fluids and eating foods that contain water.

Top 5 Benefits of Drinking Water

- Increases Energy & Relieves Fatigue. Since your brain is mostly water, drinking it helps you think, focus and concentrate better and be more alert. ...
- Promotes Weight Loss. ...
- Flushes Out Toxins. ...
- Improves Skin Complexion. ...
- Maintains Regularity.

Methodology:

- **Problem Understanding:** Initially first we have to spend some time on what are the problems or concerns students having during their pre admission period and we should set the solutions to those problems as objectives of this research.
- **Data Understanding:** Data should be collected from multiple sources and also consider all the factors including which will play a tiny role in student admission process.
- **Data Preparation:** Data should be cleaned that is removing the noise in the data and filling the missing values or extreme values and finalising the attributes/factors which will have crucial importance in student admission process.
- **Building Models:** several ML models have to be developed using various machine learning algorithms for admission to a particular university and the user interface has to be developed to access those models.
- **Evaluation:** Developed models are evaluated according to their accuracy scores. Once the model is finalised that model will be merged with node red for final deployment.

Purpose:

The main objective is to explore what kind of data is provided, determine the most important factors to check whether the water is portable or not, and select the most accurate model to suitable for prediction.

Predicting and Analyzing Urban Water Quality dataset consists of:

1992 rows with 12 columns

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

```
In [3]: data
```

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
...	...	...	...	...	...	...	...	...	...	...	...	...
1986	1330	TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU	NAN	NAN	7.9	738	7.2	2.7	0.518	0.518	202	2003
1987	1450	PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, T...	NAN	29	7.5	585	6.3	2.6	0.155	0.155	315	2003
1988	1403	GUMTI AT U/S SOUTH TRIPURA,TRIPURA	NAN	28	7.6	98	6.2	1.2	NAN	NAN	570	2003
1989	1404	GUMTI AT D/S SOUTH TRIPURA, TRIPURA	NAN	28	7.7	91	6.5	1.3	NAN	NAN	562	2003
1990	1726	CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA	NAN	29	7.6	110	5.7	1.1	NAN	NAN	546	2003

1991 rows × 12 columns

Literature Survey:

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

Example:

1.Linear Regression

2.Logistic Regression

3. Random Forest Regression / Classification.

4. Decision Tree Regression / Classification.

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

On our Dataset , we have applied Random Forest to predict the Accuracy.

Proposed Solution:

Machine Learning (**Random Forest)**:

Random Forest :

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

Working of Random Forest Algorithm

- Step 1 − First, start with the selection of random samples from a given dataset.
- Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result...
- Step 3 − In this step, voting will be performed for every predicted result.
- Step 4 − At last, select the most voted prediction result as the final prediction result.

Random Forest Algorithm in machine learning methods which efficiently performs regression tasks. It predicts the best accuracy. And the most likely class will be the output predicted for the quality estimation. And also we have created an UI using the Flask for the water quality status prediction, this UI will allow the users to predict the water quality status very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value like if it is beneficial to predict the quality of the urban water. Therefore, understanding the problems and trends of water pollution is of great significance for the prevention and control of water pollution. We have

proposed a system that uses Machine learning algorithms to predict the water quality in Urban & to forecast the predictions.

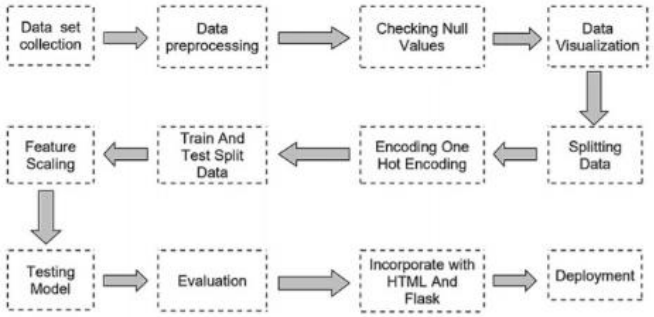
Theoretical Analysis:

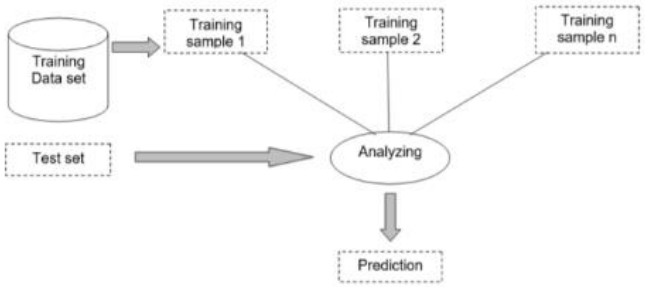
While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results accurate and from them we selected only one algorithm for the prediction problem that is Random Forest algorithm, that's how the prediction work great with the Random Forest algorithm. The peculiarity of this problem is collecting the urban water details in real time and working with the prediction at the same time, so we developed an user interface for the people who'll be accessing for the water quality status prediction. Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

Accuracy=TP+TN/TP+TN+FT+FN

At first we got like lot of worst accuracies because we tried lot of algorithms for the best accurate algorithm , finally after all of that we tried the best suitable algorithm which gives the prediction accurately is Random Forest Algorithm. And developed it to use as a real time prediction problem for the water quality prediction.

Block diagram:



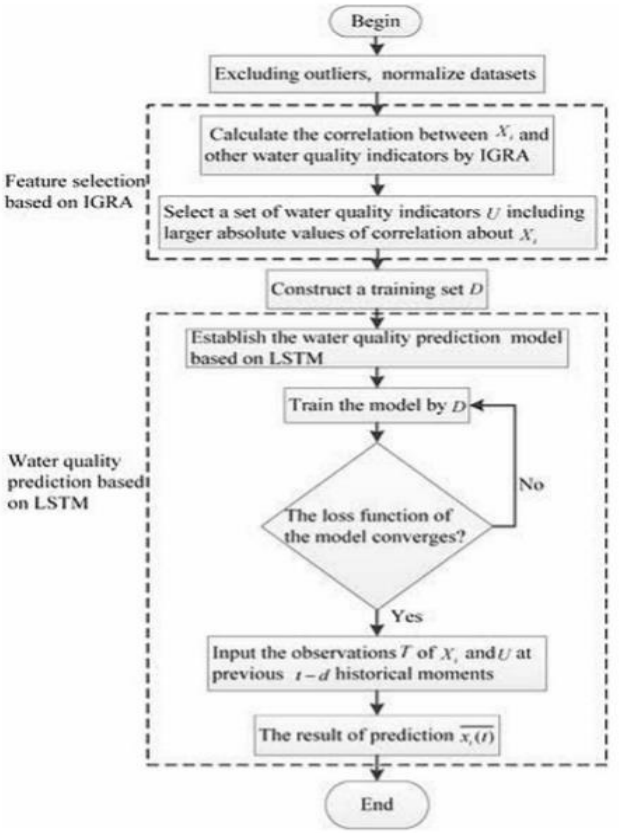


Hardware / Software designing:

- Google colab
- Anaconda navigator

- Jupyter notebook
- Machine learning tools: pandas,
  numpy,
  matplotlib,
  scikitlearn, seaborn.

Flowchart:



Result:

Results of regressions implemented:

Multilinear Regression:

```
In [64]: from sklearn.linear_model import LinearRegression
         mlr = LinearRegression()
         mlr.fit(X_train,y_train)

Out[64]: LinearRegression()

In [66]: ypred = mlr.predict(X_test)

In [67]: from sklearn.metrics import r2_score
         accuracy = r2_score(ypred,y_test)

Out[67]: 66.92470149966083
```

Decision Tree Regression:

```
In [63]: from sklearn.tree import DecisionTreeRegressor
         dec_tree = DecisionTreeRegressor(random_state=0, max_depth=6)
         dec_tree.fit(X_train, y_train)
         y_predict = dec_tree.predict(X_test)
         dec_tree_score = (dec_tree.score(X_test, y_test))
         dec_tree_score*100
Out[63]: 97.59816479570488
```

Random Forest Regression:

```
In [49]: from sklearn.ensemble import RandomForestRegressor
         reg_rf = RandomForestRegressor(n_estimators = 10 ,random_state = 0)
         reg_rf.fit(X_train, y_train)
         y_pred = reg_rf.predict(X_test)

         <ipython-input-49-890c0a298b59>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please cha
         nge the shape of y to (n_samples,), for example using ravel().
           reg_rf.fit(X_train, y_train)

In [57]: 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: 0.6857593984962461
         MSE: 2.7151162807017557
         RMSE: 1.6477609901626376

In [51]: reg_rf.score(X_train, y_train)
Out[51]: 0.997941720945234

In [58]: metrics.r2_score(y_test, y_pred)
Out[58]: 0.9915058686061382
```

Compared to Multilinear, Decision Tree Regression and Random Forest Regression,

Random Forest Regression has Highest Accuracy of 99.15058686

Results after Implementing the Flask:(Python code and Html Code):

s

Applications:

Application of predictive control strategies to the management of complex networks in the urban water.

● It can work in real time and predict as soon as the necessary details for prediction are given to the model.

● It is one of the most widely used areas of data. The water behaviour with reference to pH, do, co and bod can be analysed.

● So we use Machine Learning Algorithms to predict the water quality of the urban areas.

● Meeting the increased demand for drinking water.

ADVANTAGES AND DISADVANTAGES

Advantages:

● Effective predictive model which predicts whether water is “High ” or “Low ” for drinking purpose based on water quality parameters.

● Easy and simple User Interface for the people who is going to evaluate the urban water quality status.

● Random Forest Regression gives the accurate result of the prediction upto 96% which is the algorithm we used for prediction.

● It is composed using the HTML and Python for the web usage in real time.

● It can work in real time and predict as soon as the necessary details for prediction are given to the model.

Disadvantages:

● It could not work anywhere like an web-application, if one is using other should be quiet.

● Needs more than a single value for the prediction.

Conclusion:

In this project , the Machine learning algorithm is adopted to build a UI model for predicting water quality and the results are compared with other algorithms of Random Forest algorithm, linear regression, logistic regression. The experiment shows that the Random Forest Algorithm performs outstanding than the other algorithms in the prediction of quality default and has strong ability of generalization. There is no definitive guide of which algorithms to use given any situation. What may work on some data sets may not necessarily work on others. Therefore, always evaluate methods using cross validation to get a reliable estimates.

Future Scope:

In future the random forest algorithm can be applied on other data sets available for water quality to further investigate its accuracy. A rigorous analysis of other machine learning algorithms other than these can also be done in future to investigate the power of machine learning algorithms for urban water quality status prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state of art performance of the model and a great UI support system making it complete web application model.

Bibliography:

● Mishra, D.R.; D'Sa, E.J.; Mishra, S. Preface: Remote sensing of water resources.

Remote Sens. 2018, 10, 115.

● Jason Brownlee. Stacked Long Short-Term Memory Networks Develop Sequence

Prediction Models in Keras. 18 August 2017. Available online:

https://machinelearningmastery.com/stacked-long-short-term-memorynetworks/

(accessed on 19 January 2019).

● Storey, M.V.; van der Gaag, B.; Burns, B.P. Advances in on-line drinking water

quality monitoring and early warning systems. Water Res. 2011, 45, 741–747.

● Clark, R.; Hakim, S.; Ostfeld, A. Handbook of Water and Wastewater Systems

Protection (Protecting CriticalInfrastructure); Springer: New, York, NY, USA, 2011.

● 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.

Appendix :

Homepage: web.html

```
<html>


<style>
div.header{
  top: 0;
  position: fixed;
  padding-left: 400px;}
div.header1{
  top:20;
  position: fixed;
  padding-left: 490px;
}


*{
      margin:0;
          padding:0;
          border:0;
          outline:0;
          text-decoration:none;
          font-family:montserrat;
}


body
{
background-image:url('https://www.infinitemediacorp.com/wp-
content/uploads/2016/01/waterdrop-1.gif');
```

```
background-position: center;
font-family:sans-serif;
background-size:cover;
margin-top:40px;
}

.main{
        background-color:rgb(0,0,0,0.6);

        width:800px;
        height:590px;
        margin:auto;
        position:center;
        border-top-left-radius:100px;
        border-bottom-right-radius:100px;

}
.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"]{
        border:0;
        background:none;
        display:block;
        margin:20px auto;
        text-align:center;
        border:2px solid #3498db;
        padding:10px 3px;
        width:200px;
```

```
        outline:none;
        color:white;
        border-radius:24px;
        transition:0.25s;

}
.bor{
border:0;
        background:none;
        display:block;
        margin:20px auto;
        text-align:center;
        border:2px solid #8e44ad;
        padding:10px 3px;
        width:500px;
        outline:none;
        color:white;
        transition:0.25s;}
.main input[type="text"]:focus,.main input[type="text"]:focus,.main
input[type="text"]:focus,.main input[type="text"]:focus,.main
input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus{
        width:280px;
        border-color:#8e44ad;
}
.logbtn{
        display:block;
        width:35%;
        height:50px;
        border:none;
```

```
        border-radius:24px;
        background:linear-gradient(120deg,#3498db,#8e44ad,#3498db,#8e44ad);
        background-size:200%;
        color:#fff;
        outline:none;
        cursor:pointer;
        transition:.5s;
        font-size:25;
}
.logbtn:hover{
        background-position:right;
}

input::placeholder{
        color:#F5FFFA;
}
.bottom-text{
        margin-top:60px;
        text-align:center;
        font-size:13px;

}



</style>
<body>
        <center><div class="header"><img
src="https://freepngimg.com/download/water/76727-vector-of-drop-water-drop-water-
free-download-image.png" width="100" height="100"></div></center>
```

```
        <center><div class="header1"><font color="" font-family="Fascinate Inline" size=7
><b>Urban Water Quality Prediction</b></font></div></center>



<br><br><br><br><br>
        <form class="main" action="/login" method="post">
                <br>
                <center><input type="text" name="year" placeholder="Enter Year"/>
                <input type="text" name="do" placeholder="Enter D.O "/>
                <input type="text" name="ph" placeholder="Enter PH"/>
                <input type="text" name="co" placeholder="Enter Conductivity"/>
                <input type="text" name="bod" placeholder="Enter B.O.D"/>
                <input type="text" name="na" placeholder="Enter Nitratenen"/>
                <input type="text" name="tc" placeholder="Enter Total Coliform"/>
                <input type="submit" class="logbtn" value="Predict"></center>
                <div class="bor"><center><b><font color="white"
size=5>{{showcase}}</font></b></center></div>
        </form>



</body>


</html>
```

Python Code: app.py

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

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


@app.route('/')
def home() :
    return render_template("web.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 = [[int(year),float(do),float(ph),float(co),float(bod),float(na),float(tc)]]
    y_pred = model.predict(total)
    print(y_pred)
    y_pred =y_pred[[0]]
    if(y_pred >= 95 and y_pred <= 100) :
        return render_template("web.html",showcase = 'Excellent,The predicted value is '+
str(y_pred))
    elif(y_pred >= 89 and y_pred <= 94) :
        return render_template("web.html",showcase = 'Very good,The predicted value is
'+str(y_pred))
    elif(y_pred >= 80 and y_pred <= 88) :
        return render_template("web.html",showcase = 'Good,The predicted value
is'+str(y_pred))
```

```
    elif(y_pred >= 65 and y_pred <= 79) :
        return render_template("web.html",showcase = 'Fair,The predicted value is
'+str(y_pred))
    elif(y_pred >= 45 and y_pred <= 64) :
        return render_template("web.html",showcase = 'Marginal,The predicted value is
'+str(y_pred))
    else :
        return render_template("web.html",showcase = 'Poor,The predicted value is
'+str(y_pred))



if __name__ == '__main__' :
    app.run(debug = True,port=5000)
```