URBAN WATER QUALITY PREDICTION

Using Machine Learning Algorthms

Developed by: Anjireddy Gopireddy, Ravali Gollapally, Anusha Indoori, Saikumar Angiravikala, Venkateshwarababu Pitta

Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION

Urban water is a vital resource that affects various aspects of human, health and urban lives. People living in major cities are increasingly concerned about the urban water quality, calling for technology that can monitor and predict the water quality in real time throughout the city.There are two most important goals for prediction and description. Prediction involves using some variables in data set to predict unknown values of other variables and Description concentrates on finding patterns describing the data that can be interpreted by human. The derived knowledge must be new, not obvious, relevant and can be applied in the field where this knowledge has been obtained. It is also the process of extracting useful information from raw data.

There are huge number of phases in the prediction based on Machine Learning, and this prediction problem used most of them, Data collection is the first phase, by this phase data should be collected not usually a less data set, it should be huge data set according to the requirements one should collect or create the data for the prediction. Data Pre-processing is the second phase and this contain a lot of sub-phases for the processing of the data, it includes importing libraries, Data Visualization, Data Transformation, Feature Scaling, Splitting and Label Encoding. Data Splitting, in this phase the data is to be split into two as train data and test data for the training of the model. Then the Fourth phase is Model Training, Supervised learning allows for processing data with target attributes or labelled data. These attributes are mapped in historical data before the training begins. And the last phase is Model evaluation and Testing and it is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That’s the optimization of model parameters to achieve an algorithm’s best performance. Machine learning techniques aid to we developed a deep learning model to predict the water quality.

1.1 Overview

With the rapid development of economy and accelerated urbanization, water pollution has become more and more serious. Urban water quality is of great importance to our daily lives. Prediction of urban water quality help control water pollution and protect human health.

Data Mining is one of the most motivating and vital area of research with the aim of extracting information from tremendous amount of accumulated data sets. Here we developed a deep learning model to predict the water quality by using Machine Learning concepts. Algorithms have been used to build the proposed model: Random Forest, Linear Regression, Decision Tree. By using the algorithm a Flask model has been implemented and tested. The results have been discussed and a full comparison between algorithms was conducted. Decision tree algorithm was selected as best algorithm based on accuracy.

1.2 Purpose

Our aim from the project is to make use of pandas, matplotlib, & seaborn libraries from python to extract the libraries for machine learning for the water quality prediction.

Secondly, to learn how hyper tune the parameters using grid search cross validation for the Decision tree machine learning algorithm.

And in the end, to predict water quality percentage of the specific year, we are using machine learning algorithms and withdrawing the conclusions.

2. LITERATURE SURVEY

- 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, Neural Networks, Deep Neural Networks and k Nearest Neighbors, 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. Using only three parameters and comparing them to standardized values is quite a limitation when predicting water quality.

- 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.

2.1 Existing Problem

Water pollution is a critical issue that can affects humans' health and the entire ecosystem thus inducing economical and social concerns. In this paper, we focus on water quality prediction system. With the rapid development of economy and accelerated urbanization, water pollution has become more and more serious. Urban water quality is of great importance to our daily lives. Prediction of urban water quality help control water pollution and protect human health.

2.2 Proposed Solution

Machine Learning (Decision Tree):

Decision tree 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.

3. 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 Decision tree algorithm, that's how the prediction work great with the Decision tree Algorithm.

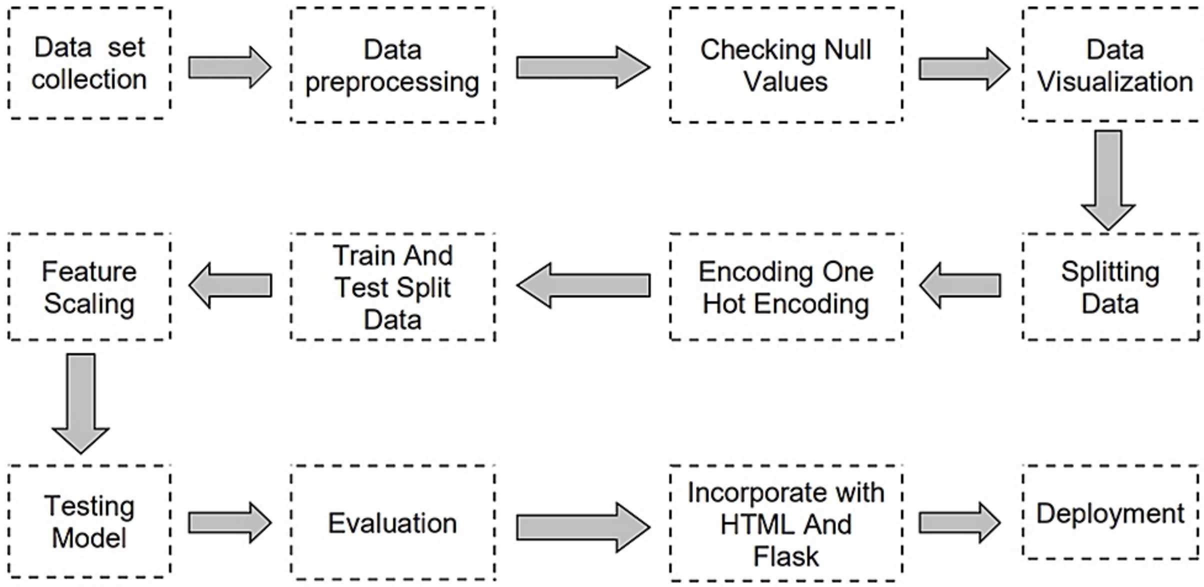
The peculiarity of this problem is collecting the urban water details in real time and working with the prediction at the same time, so we developed an user interface for the people who'll be accessing for the water quality status prediction. Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

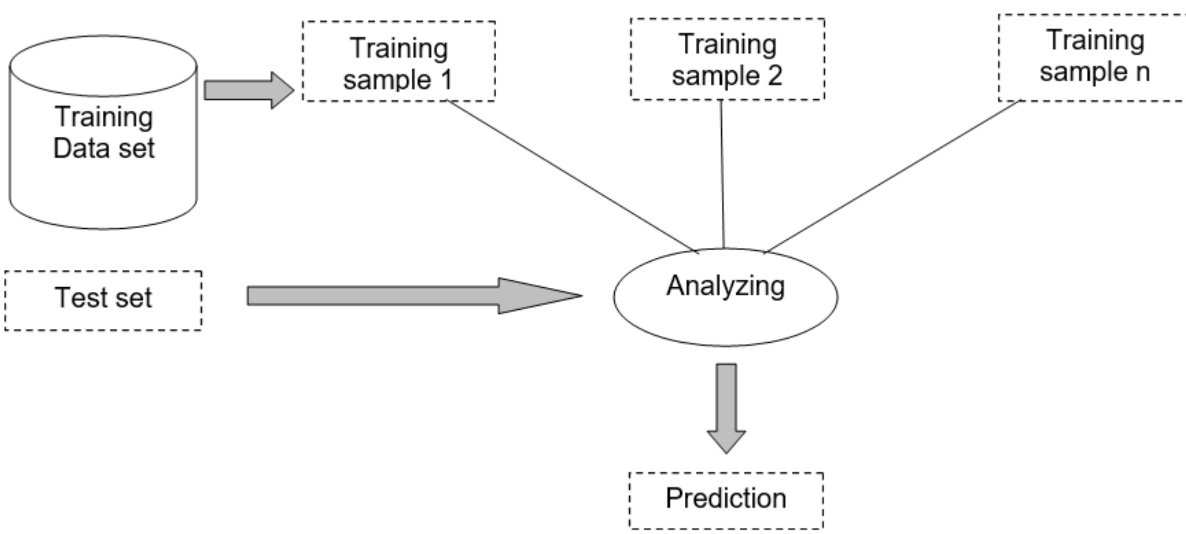
Accuracy=TP+TN/TP+TN+FT+FN

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 Decision tree regression. And developed it to use as a real time prediction problem for the water quality prediction.
In statistics, a receiver operating characteristic, is a two dimensional graphical plot that illustrates the performance of a binary classifier system. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. ROC curve can intuitively represent the performance of classifier.

FPR=FP/FP+TN TPR=TP/TP+F

3.1 Block Diagram





3.2 Software Designing

- Jupyter Notebook Environment
- Spyder Ide
- Machine Learning Algorithms
- Python (pandas, numpy, matplotlib, seaborn,sklearn)
- HTML
- Flask

We developed this water quality status prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language.

For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro framework 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, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

4. EXPERIMENTAL INVESTIGATION

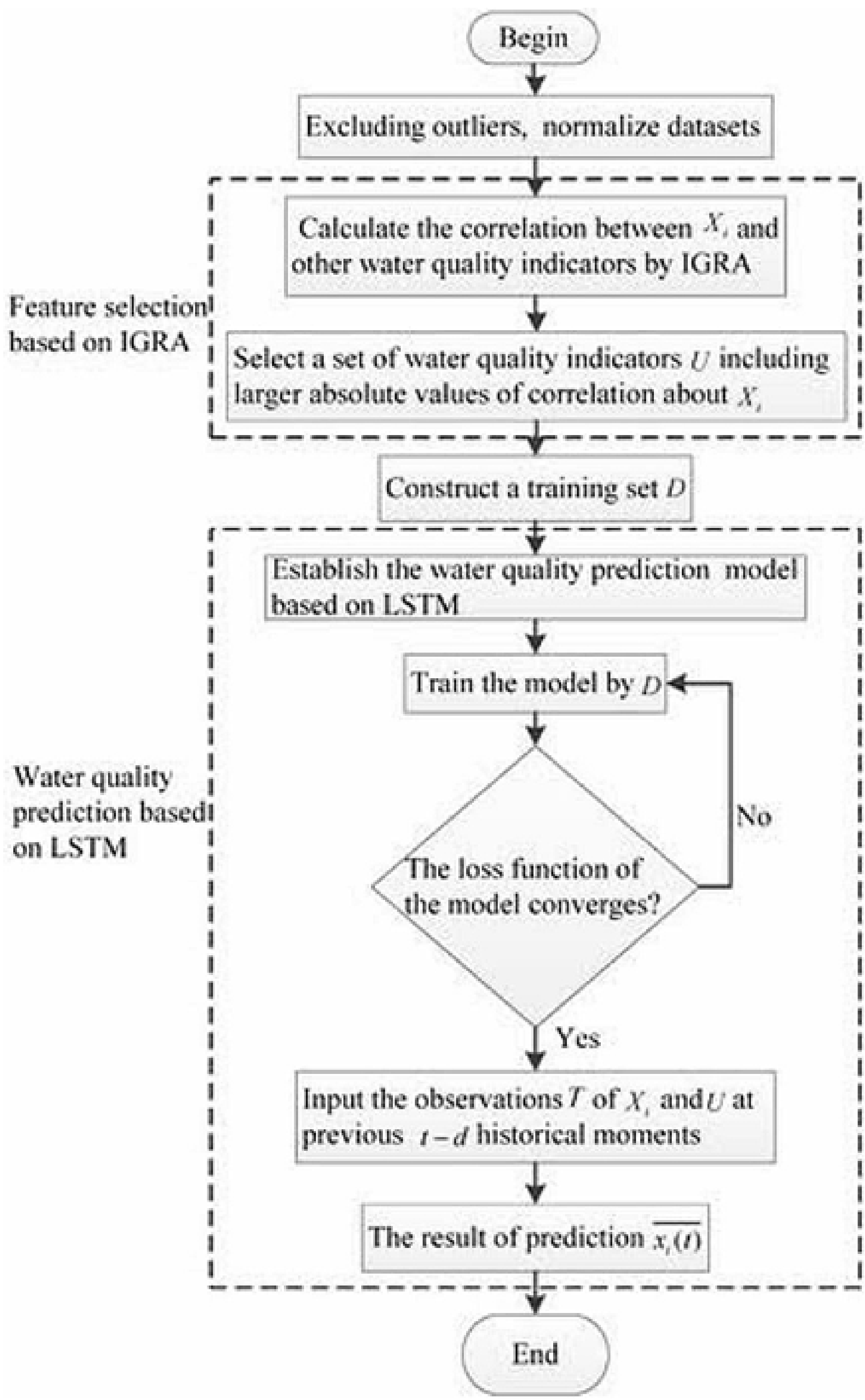
In this paper, the dataset we used is derived from https://www.kaggle.com/anbarivan/indian-water-quality-data It contains 1992 original data of water quality with 12 attributes. After that, the missing values are filled in by means of mode interpolation, and the duplicate or meaningless attributes are deleted, finally we have retained to 9 attributes. Those attributes were shown below in the screenshot of the data set we used.

year	station	do	ph	co	bod	na	tc	wqi
2014	1393	6.7	7.5	203	1.8965	0.1	27	93.82
2014	1399	5.7	7.2	189	2	0.2	8391	76.96
2014	1475	6.3	6.9	179	1.7	0.1	5330	79.28
2014	3181	5.8	6.9	64	3.8	0.5	8443	69.34
2014	3182	5.8	7.3	83	1.9	0.4	5500	77.14
2014	1400	5.5	7.4	81	1.5	0.1	4049	77.14
2014	1476	6.1	6.7	308	1.4	0.3	5672	75.44
2014	3185	6.4	6.7	414	1	0.2	9423	75.44
2014	3186	6.4	7.6	305	2.2	0.1	4990	82.04
2014	3187	6.3	7.6	77	2.3	0.1	4301	82.76
2014	1543	7.1	7.1	176	1.2	0.1	7817	82.58
2014	1548	6.7	6.4	93	1.4	0.1	3433	66.26
2014	2276	7.4	6.8	121	1.7	0.4	18125	68.22
2014	2275	6.9	7	620	1.1	0.1	6300	82.04
2014	3189	6	7.5	72	1.6	0.2	9517	82.94
2014	1546	7.3	7	247	1.5	0.2	2453	82.4
2014	2270	7.3	7	188	1	0.1	3048	82.58
2014	2272	7	6.9	224	1.2	0.3	6742	79.28
2014	1545	7.3	6.7	144	1.5	0.1	3052	76.16
2014	2274	5.3	6.8	319	1.8	0.3	10250	61.88
2014	2271	6.3	6.4	79	1.6	1.4	12842	55.02
2014	2273	5.4	7.6	39	1.4	0.1	6367	77.32
2014	3183	2.2	6.5	322	4.7	1.2	14920	28.12
2014	3184	5.2	7.1	192	2.6	0.3	8925	76.96
2014	3190	5.6	7.5	282	1.8	0.1	5082	76.78
2014	3191	5.5	7.4	275	1.5	0.1	8625	76.78
2014	1547	7.3	6.7	55	1.4	0.1	4003	76.34
2014	3188	6.5	7.5	415	2	0.1	1538	82.04
2014	1544	7.2	6.3	100	1.5	0.1	13575	55.02
2014	2651	6.6	7.8	95	4.9	0.2	36	89.32
2014	1461	6.9	7.9	99	5	0.4	34	89.32

water (1)

Ready

5. FLOWCHART

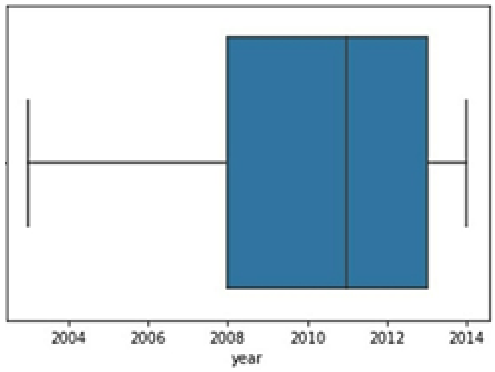


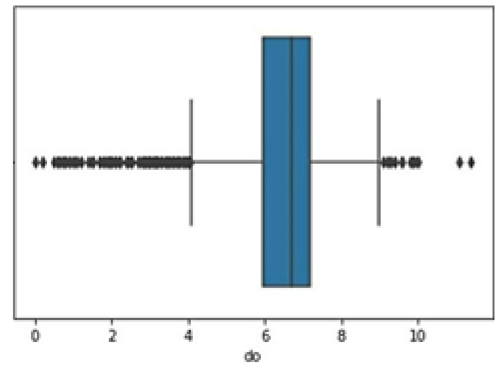
Begin
Excluding outliers, normalize datasets
Calculate the correlation between X_i and other water quality indicators by IGRA
Feature selection based on IGRA
Select a set of water quality indicators U including larger absolute values of correlation about X_i
Construct a training set D
Establish the water quality prediction model based on LSTM
Train the model by D
No
Water quality prediction based on LSTM
The loss function of the model converges?
Yes
Input the observations T of X_i and U at previous $t-d$ historical moments
The result of prediction $\overline{x_i(t)}$
End

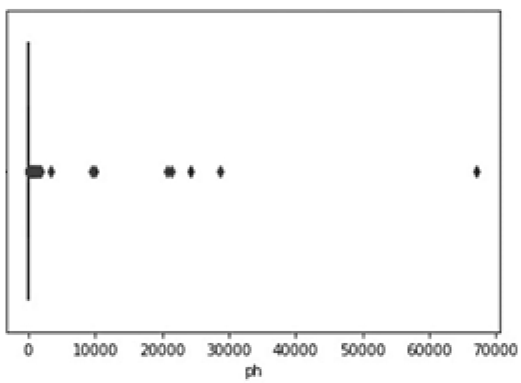
6. RESULT

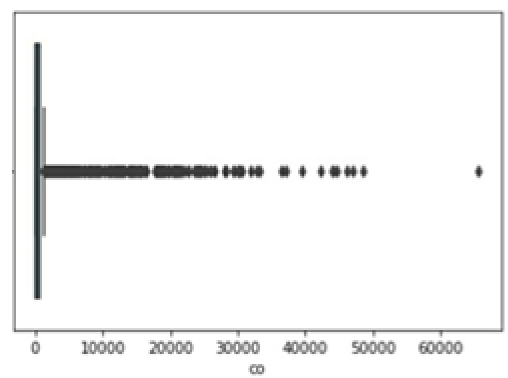
In this paper, the decision tree algorithm is used to predict its performance, and compared with another two machine learning methods namely the linear regression and the Random Forest. The obtained results are displayed in Table below. The results show that, the performance of Decision tree have comparable performance than that of random forest and linear regression, but the still performs the best, with an accuracy of 96%, higher than the linear regression with an accuracy of 59%.

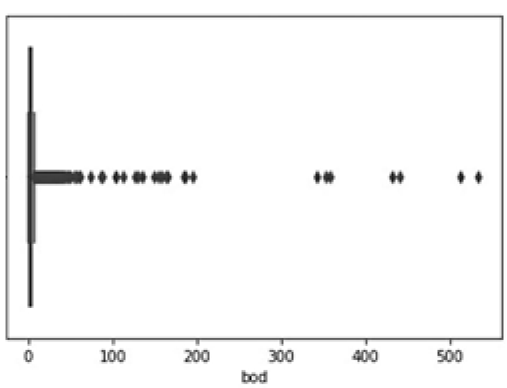
The given are the heatmap of the dataset represents the correlation between attributes and the boxplot of each attribute.

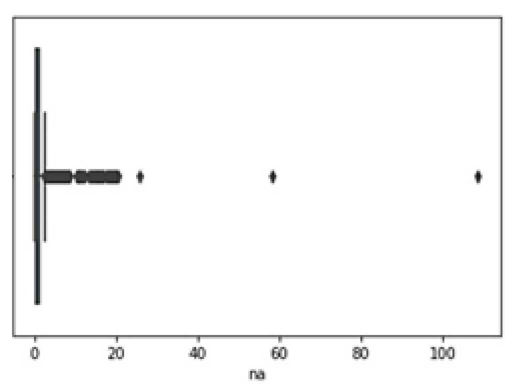

2004
2006
2008
2010
2012
2014
year

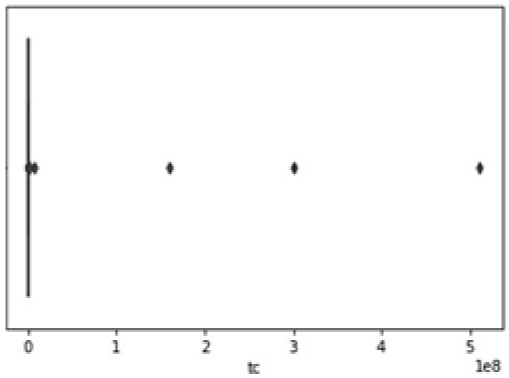

0
2
4
6
8
10
do

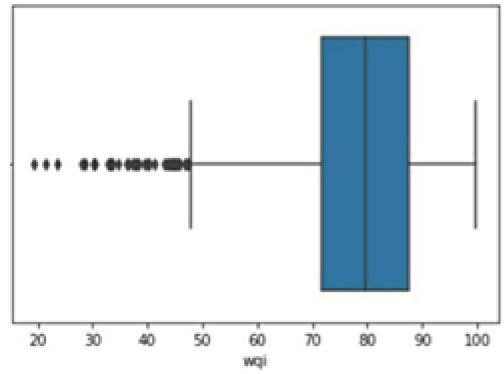

0
10000
20000
30000
40000
50000
60000
70000
ph

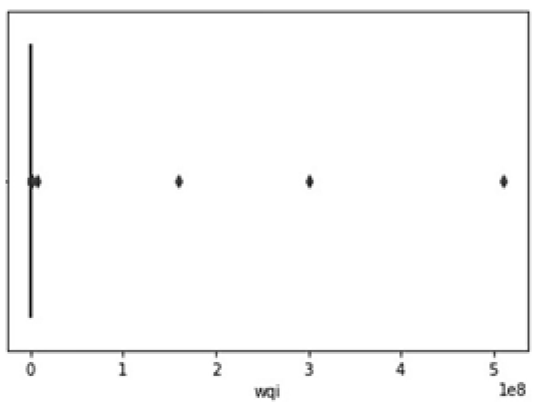

0
10000
20000
30000
40000
50000
60000
co

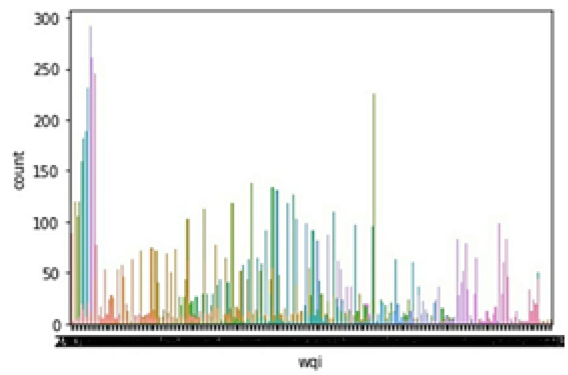

0
100
200
300
400
500
bod


0
20
40
60
80
100
na









S No:	Algorithms Used	Accuracy
1	Random Forest	0.96%
2	Decision Tree	0.96%
3	Linear Regression	0.59%

7. 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.
- Decision tree give 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.

8. 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 analyzed.
- So we use Machine Learning Algorithms to predict the water quality of the urban areas.
- Meeting the increased demand for drinking water.

9. CONCLUSION

In this paper, the Machine learning algorithm is adopted to build a UI model for predicting water quality and the results are compared with other algorithms of linear regression, random forest, decision tree and support vector machine. The experiment shows that the Decision tree 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.

10. FUTURESCOPE

In future the decision tree 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.

11. 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

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://images.pexels.com/photos/1100946/pexels-photo-1100946.jpeg?auto=compress
&cs=tinysrgb&dpr=1&w=500');
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"],.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="../static/css/logo.png" width="100"
height="100"></div></center>
<center><div class="header1"><font color="#FF0000" 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>
```

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)
    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)
```