**URBAN WATER QUALITY PREDICTION**

Using Machine Learning (Naive Bayes Classifier)

**Smart Bridge-Remote Summer Internship Program**

**Developed by : Sunkari Shruthi, Prathyusha Burra, Sriram Sowmya, Venna Aravind**

1. INTRODUCTION

Since the early beginning of the development of natural sciences, collecting and assay of huge amounts of data was one of the leading analytical tools. The same goes for environmental sciences and environmental engineering, which produce higher demand for efficient and productive approaches to work with continuously increasing sizes of the collected data from a huge variety of research fields every day. (Kendall and Costello 2006)

Nowadays, machine learning algorithms have proven themselves as a universal tool for different types of tasks, giving advanced possibilities for dealing with analysed data, including such types of tasks as *data imputation*, *unsupervised clusterization*, *classification* and *regression*. They are commonly used in many research areas; however, they are yet less common among environmental engineering workers, though such tools may provide an extremely efficient alternative to the traditional analytical approaches. (Wilcox, Woon and Aung 2013)

The purpose of the research behind this was in presenting of examples of how such advanced tools may be used on a particular data set meant for increasing water quality in European region. In the following chapters one will go through the presentation of the machine learning, it’s origins and possibilities in general, explanation of the data and models used during the research, results of the application of algorithms, discussion (covering obstacles one can face while working with this kind of models) and conclusion, which will cover the presented material, give advices for engineers and scientists who would like to use this models for their environmental tasks and finally and give some words about the possible future of the development of these tools in environmental field.

1. Overview

Nowadays, There are many risks occurring due to urban water, for those who are getting the problems due to urban water. The analysis of risk in urban water needed to understand what is the meaning of risk. How do you describe the water quality and what are the different water quality parameters and what is the best water quality and how do you monitor water quality and how do you analyse water quality and what are 2 causes of increased turbidity. 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 a new model for regression urban water quality prediction by using Machine Learning concepts.

The model has been built using data form water quality sector to predict the status of water. Six algorithms have been used to build the proposed model: Random Forest, Logistic Regression, Decision Tree, SVM, KNN Naive Bayes. By using the algorithm a Flask model has been implemented and tested. The results has been discussed and a full comparison between algorithms was conducted. Naive Bayes was selected as best algorithm based on accuracy.

1. 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 to hyper tune the parameters using grid search cross validation for the Naive Bayes machine learning algorithm.

And in the end, to predict whether the water quality is good or not using techniques of combining the predictions from multiple machine learning algorithms and withdrawing the conclusions.

1. LITERATURE SURVEY

Data mining is the process of analysing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data as depicted in figure-1. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to classification technique. This approach frequently employs Decision tree based classification Algorithm. In classification, a training set is used to build the model as the classifier which can classify the data items into its appropriate classes.

A test set is used to validate the model.

1. Existing Problem

With the rapid development of economy and accelerated urbanisation, 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. To overcome this kind of problem statement, we developed a Deep Learning model to predict the water quality.

1. Proposed Solution

**Machine Learning (Naive Bayes):**

Naive Bayes algorithm in machine learning methods which efficiently performs both classification and regression tasks. Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. And the mot likely class will be the output predicted for the water quality.

And also we have created an UI using the Flask for the urban water quality prediction, this UI will allow the users to predict the loan 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.

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.

1. THEORETICAL ANALYSIS

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is Naive Bayes Classifier, it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

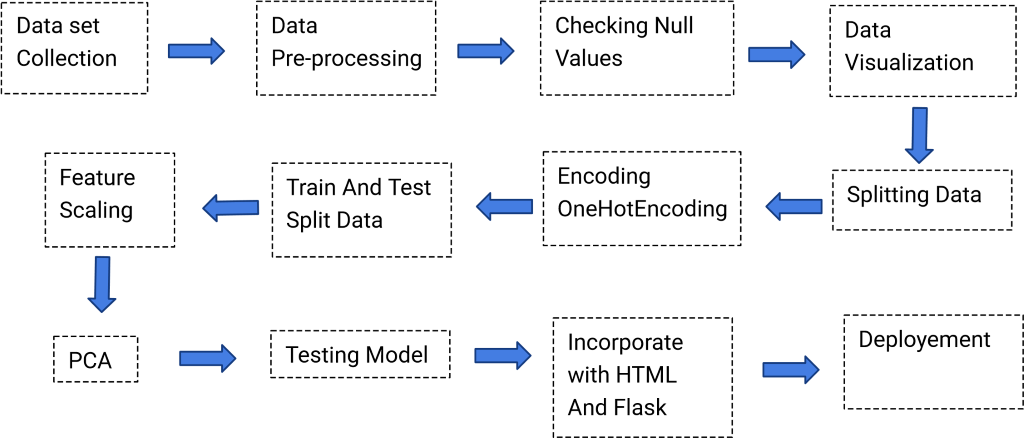
That’s how the prediction work great with the Naive Bayes Algorithm.

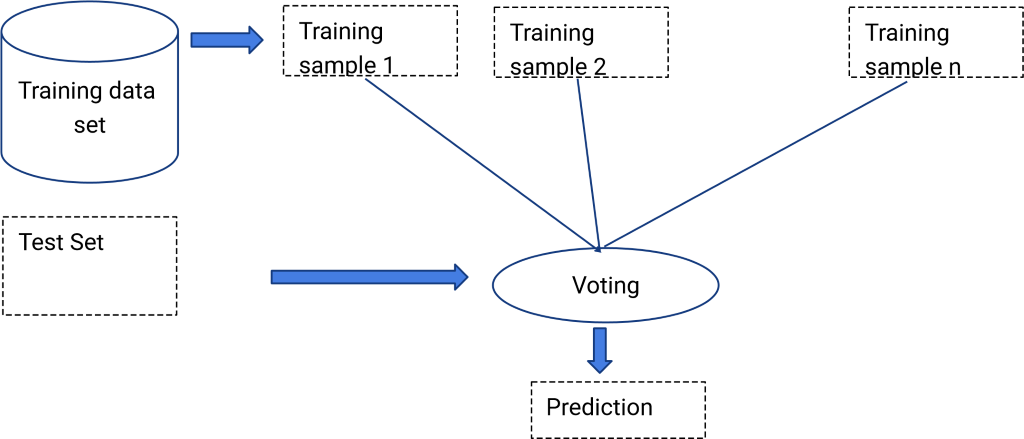
Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

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 Naive Bayes Classifier. And developed it to use as a real time prediction problem for the water quality prediction.

1. Block Diagram





1. Software Designing

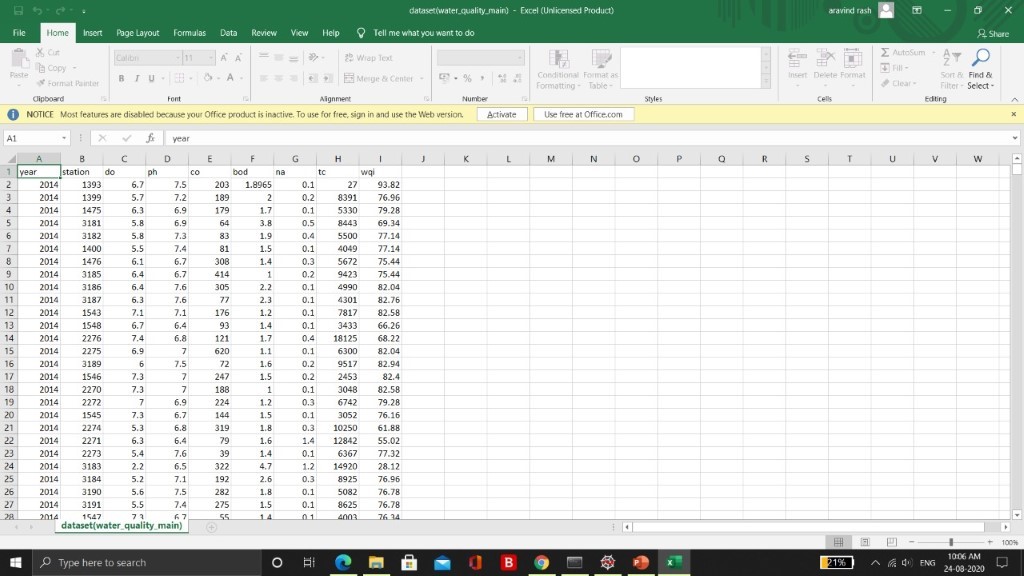
* Jupyter Notebook Environment
* Spyder Ide
* Machine Learning Algorithms
* Python (pandas, numpy, matplotlib, seaborn, sklearn)
* HTML
* Flask

We developed this urban water quality 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.

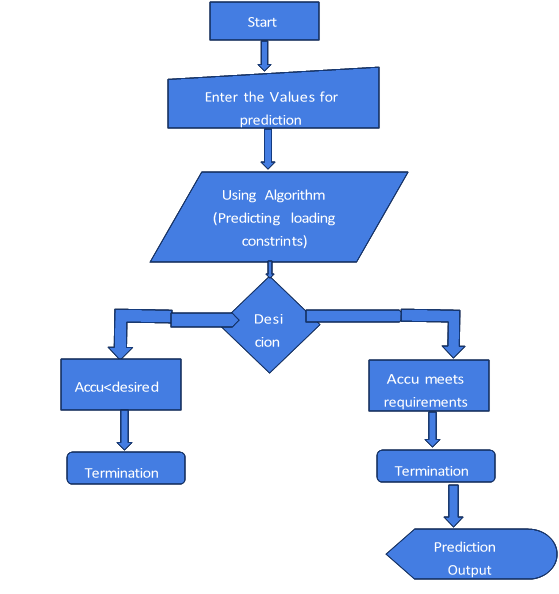
For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing 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.

1. EXPERIMENTAL INVESTIGATION

In this paper, the dataset we used is derived from dataset(water\_quality\_main). In this we are using multiple attributes to predict and get high level of water quality. Those attributes were shown below in the screenshot of the data set we used.

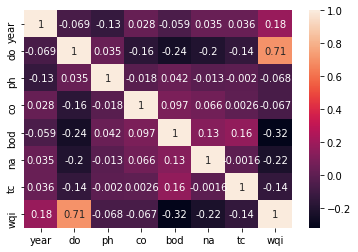


1. FLOWCHART



1. RESULT

In this paper, the Naive Bayes algorithm is used to predict its performance, and compared with another six machine learning methods namely the decision tree, the logistic regression, Random Forest and the SVM. The obtained results are displayed in Table below. But, the Random Forest Regression still perform the best with an accuracy of 94% higher than the Logistic Regression with an accuracy of 61%.



|  |  |  |
| --- | --- | --- |
| **S No:** | **Algorithms Used** | **Accuracy** |
| 1 | Random Forest | 94% |
| 2 | Decision Tree | 94% |
| 3 | Logistic Regression | 61% |

1. ADVANTAGES AND DISADVANTAGES

**Advantages:**

* Easy and simple User Interface for the people who is going to evaluate the water quality.
* Naive Bayes give the accurate result of the prediction up to 94% which is the algorithm we used for prediction.
* It is widely used for managing risks in the water quality industry.
* 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:**

* Gives only 94% accuracy for the water quality.
* 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.

1. APPLICATIONS

* Machine learning is a powerful tool which can be very useful when analysing environmental data, including water quality, and can form a backbone for competent AI systems which help manage and monitor water.
* More accurate predictions will lead to more insightful data, empowering our customers to make better business decisions.
* It is one of the most widely used areas of data mining in the water quality industry.
* Due to tremendous growth in data the water quality industry deals with, analysis and transformation of the data into useful knowledge has become a task beyond human ability.
* So we use Machine Learning Algorithms to analyse the quality of the water.

1. CONCLUSION

In this paper, the Naive Bayes algorithm is adopted to build a UI model for predicting water quality default in and the results are compared with other six algorithms of logistic regression, KNN, random forest, decision tree and support vector machine. The experiment shows that the Naive Bayes algorithm performs outstanding than the other six algorithms in the prediction of water quality 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.

1. FUTURE SCOPE

In future the Naive Bayes 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 six can also be done in future to investigate the power of machine learning algorithms for urban water quality 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.

1. BIBLIOGRAPHY

* Batista, G. E. A. P. A., and M. C. Monard. “A Study of K-Nearest Neighbour as an Imputation Method .” *Soft Computing Systems: Design, Management and Applications* , 2002: pp. 251–260 .
* Behar, Sharon. *Testing the Waters: Chemical and Physical Vital Signs of a River.* River Watch Network, 1996.
* Biau, Gerard. “Analysis of a Random Forests Model.” *The Journal of Machine Learning Research* 13, no. 1 (2012): 1063-1095.
* Blackwell, Matthew, James Honaker, and Gary King. “A Unified Approach to Measurement Error and Missing Data: Overview and Applications.” *Sociological Methods & Research*, 2015: 1-39.
* Breiman, Leo. “Random Forests.” *Machine Learning* 45, no. 1 (2001): 5–32.
* Brooks, Randy. *Forest Soils versus Agricultural Soils.* UI Extension Forestry Information Series, University of Idaho, University of Idaho, 2005.
* Browne, Michael W. “Cross-Validation Methods.” *Journal of Mathematical Psychology* 44, no. 1 (2000): pp 108–132.
* Chapman, Pete, et al. *CRISP-DM 1.0 .* Step-by-step data mining guide, SPSS, 2000. Cheng, Bing, and D. M. Titterington. “Neural Networks: A Review from a Statistical Perspective.” *Statistical Science* 9, no. 1 (1994): 2-30.
* Chenier, Martin R., Danielle Beaumier, Real Roy, Brian T. Driscoll, John R. Lawrence, and Charles W. Greer. “Impact of Seasonal Variations and Nutrient Inputs on Nitrogen Cycling and Degradation of Hexadecane by Replicated River Biofilms.” *Applied and Environmental Microbiology* 69, no. 9 (2003): pp 5170-5177.
* Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” *Machine Learning* 20, no. 3 (1995): pp 273-297.
* Domingos, Pedro. “A few useful things to know about machine learning.” *Communications of the ACM* 55, no. 10 (2012): pp 78-87.
* Frew, James E., and Jeff Dozier. “Environmental Informatics .” *Annual Review of Environment and Resources* 37 (2012): pp. 449-472Hagan, Martin T., Howard B. Demuth, Mark Hudson Beale, and Orlando De Jesus. *Neural Network Design.* University of Colorado, 2002.
* Lebanon, Guy. *Bias, Variance, and MSE of Estimators.* Technical Notes, Georgia: Georgia Institute of Technology, 2010.
* Li, Ray. *History of data mining.* 26 April 2015. http://rayli.net/blog/data/history-of-datamining/ (accessed December 25, 2015).
* Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. “The global k-means clustering algorithm.” *Pattern Recognition* (Elsevier) 36, no. 2 (February 2003): 451–461.
* Lu, Wei, Yanyan Shen, Su Chen, and Beng Chin Ooi. “Efficient Processing of k Nearest Neighbor Joins using MapReduce.” *Proceedings of the VLDB Endowment* 5, no. 10 (2012):
* 1016-1027.
* Mitchell, T. *Machine Learning.* McGraw Hill, 1998.
* Refaeilzadeh, Payam, Lei Tang, and Huan Liu. “Cross-Validation.” *Encyclopedia of Database Systems*, 2009: pp 532-538.
* Shkurin, Aleksei, and Alfredo Vellido. “Random Forests for quality control in G-Protein Coupled Receptor databases .” 2016.
* Simon, Phil. *Too Big to Ignore: The Business Case for Big Data.* New Jersey: Wiley, 2013.
* Starkweather, Jon. *Cross Validation techniques in R: A brief overview of some methods, packages, and functions for assessing prediction models. .* Review, University of North Texas, 2011.
* Teknomo, Kardi. “Kardi Teknomo – K Mean Clustering Tutorial .” *croce.ggf.br.* July 2007. http://croce.ggf.br/dados/K%20mean%20Clustering1.pdf (accessed December 27, 2015). Van Kessel, Chris, Tim Clough, and Jan Willem Van Groenigen. “Dissolved Organic Nitrogen: An Overlooked Pathway of Nitrogen Loss from Agricultural Systems?” *Journal of Environmental Quality* 38, no. 2 (2009): pp 393-401.
* Vellido, A., E. Marti, J. Comas, I. Rodriguez-Roda, and F. Sabater. “Exploring the ecological status of human altered streams through Generative Topographic Mapping .” *Environmental Modelling & Software* (Elsevier), no. 22 (2007 ): 1053e1065.
* Wagstaff, Kiri, Claire Cardie, Seth Rogers, and Stefan Schrödl. “Constrained K-means
* Clustering with Background Knowledge.” *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001: 577-584.
* Warren, McCulloch, and Walter Pitts. “A Logical Calculus of Ideas Immanent in Nervous Activity.” *Bulletin of Mathematical Biophysics* 5, no. 4 (1943): pp 115–133. Weston, Jason. “Support Vector Machine (and Statistical Learning Theory).” *cs.columbia.edu.* 2004.
* http://www.cs.columbia.edu/~kathy/cs4701/documents/jason\_svm\_tutorial.pdf (accessed January 15, 2016).

APPENDIX

**HTML:**

<!DOCTYPE html>

<html>

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>ML API</title>

<link href='https://https://miro.medium.com/max/1000/1\*rSy2sTO\_g0WRbE2BLECiQA.png' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="../static/css/style.css">

<style>

.login{

top:10%;

}

</style>

</head>

<body>

<h> <h1>WATER QUALITY PREDICTION</h1>

<div class="login">

<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url\_for('y\_pred')}}" method="post">

<label for="year">Year</label>

<input type="text" name="year" placeholder="Enter year " id="year" required="required" />

<label for="do">Do</label>

<input type="text" name="do" placeholder="Enter do value" id="do" required="required" />

<label for="ph">Ph</label>

<input type="text" name="ph" placeholder="Enter ph value" id="ph" required="required" />

<label for="co">Co</label>

<input type="text" name="co" placeholder="Enter co value" id="co" required="required" />

<label for="bod">Bod</label>

<input type="text" name="bod" placeholder="Enter bod value" id="bod" required="required" min="0" max="5" />

<label for="na">Na</label>

<input type="text" name="na" placeholder="Enter na value" id="na" required="required" />

<label for="tc">Tc</label>

<input type="text" name="tc" placeholder=" Enter tc value" id="tc" required="required" /></h>

<button type="reset" class="btn btn-primary btn-large">Clear</button>

<button type="submit" class="btn btn-primary btn-large">Predict</button>

<br>

</br>Water Quality Prediction is : {{ prediction\_text }}

</form>

</div>

</body>

</html>

**CSS:**

<html>

<style>

\*{

margin:0;

padding:0;

text-decoration:none;

font-family:montserrat;

}

.pd{

padding-bottom:100%;}

body

{

background-image:url('https://miro.medium.com/max/1000/1\*rSy2sTO\_g0WRbE2BLECiQA.png') no-repeat center fixed;

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:500px;

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;

}

.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);

background-size:200%;

color:#fff;

outline:none;

cursor:pointer;

transition:.5s;

}

.logbtn:hover{

background-position:right;

}

.bottom-text{

margin-top:60px;

text-align:center;

font-size:13px;

}

p.inset {

border-style: inset;

border-color:#6699FF;

}

**APP.PY:**

from flask import Flask, request , render\_template

import pickle

import numpy as np

app = Flask(\_\_name\_\_)

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

@app.route('/')

def home():

return render\_template('page.html')

@app.route('/y\_pred',methods=['POST'])

def y\_pred():

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']

data=[[int(year),float(do),float(ph),float(co),float(bod),float(na),float(tc)]]

prediction = model.predict(data)

print(prediction)

output=prediction[0]

if(output>=95 and output<=100):

return render\_template('page.html', prediction\_text='Excellent The predicted WQI is:'+str(output))

if(output>=85 and output<=94):

return render\_template('page.html', prediction\_text='Good The predicted WQI is:'+str(output))

if(output>=75 and output<=84):

return render\_template('page.html', prediction\_text='Fair The predicted WQI is:'+str(output))

if(output>=65 and output<=74):

return render\_template('page.html', prediction\_text='Mariginal The predicted WQI is:'+str(output))

if(output>=45 and output<=64):

return render\_template('page.html', prediction\_text='Poor The predicted WQI is:'+str(output))

else:

return render\_template('page.html', prediction\_text='Not Fit for drinking The predicted WQI is:'+str(output))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)