A project report on

WATER QUALITY PREDICTION

Submitted in partial fulfilment for the award of the degree of

Master of Technology (Integrated 5 Year)

Software Engineering

by

K.CHANDANA (18MIS7060)

AMARAVATI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

28-08-2021

GUIDED

BY

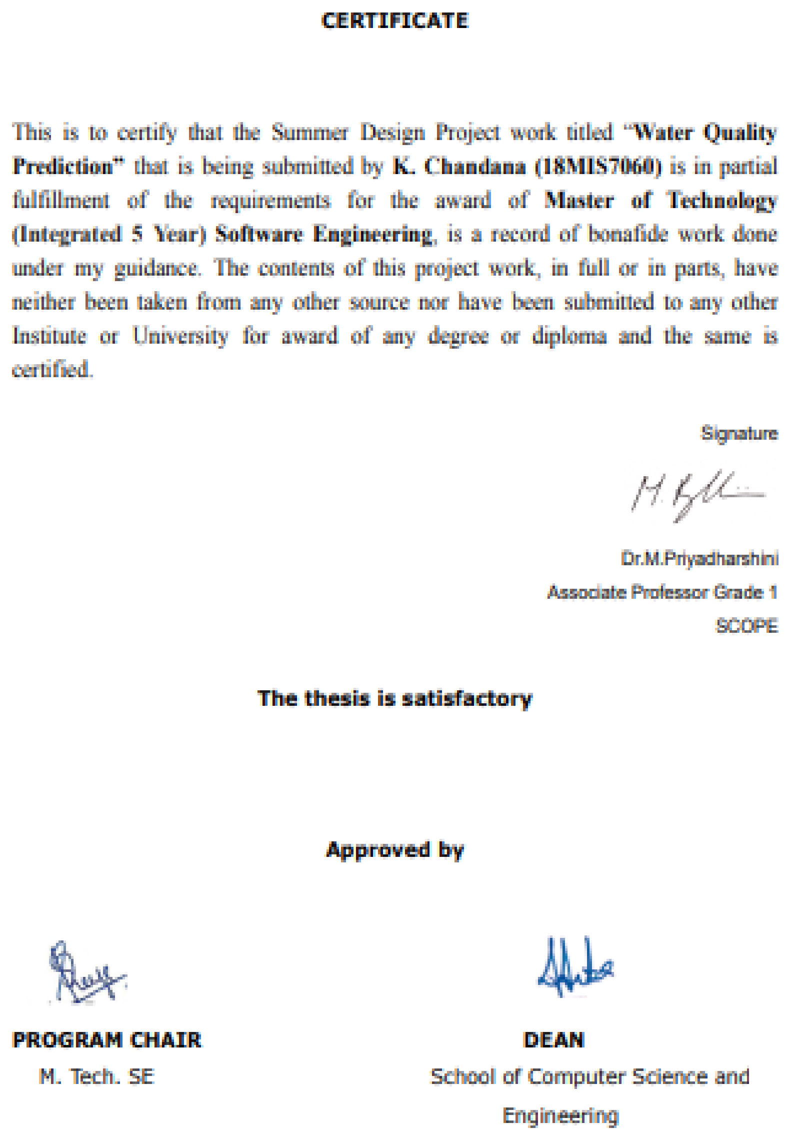
Dr.PRIYADHARSHINI

Associate Professor Grade 1

SCOPE OF COMPUTER SCIENCE ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY -AMARAVATHI.

CERTIFICATE



CERTIFICATE

This is to certify that the Summer Design Project work titled **"Water Quality Prediction"** that is being submitted by **K. Chandana (18MIS7060)** is in partial fulfillment of the requirements for the award of **Master of Technology (Integrated 5 Year) Software Engineering**, is a record of bonafide work done under my guidance. The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Signature

Dr.M.Priyadharshini
Associate Professor Grade 1
SCOPE

The thesis is satisfactory

Approved by

PROGRAM CHAIR
M. Tech. SE

DEAN
School of Computer Science and Engineering

ABSTRACT

Water Quality Prediction remains a crucial task for government officials, especially for countries such as India, owing to the emergence of water related health issues and their causal effects. The analysis of water quality at real time would certainly be helpful to human beings as it would create awareness about the quality of the water during climatic changes like rain conditions. This paper proposes a real time water quality prediction combining the real-time monitoring and prediction mechanisms. Water quality prediction approach monitors water quality parameters at predefined water quality monitoring sites; it predicts the water quality parameter values using prediction algorithms such as Random Forest (RF) or Linear Regression (LR) when monitoring sites are not accessible; and, it utilizes Google Maps for specifying the quality of water at real-time to the user of Water Quality predictor.

i

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude towards Dr.M.Priyadharshini, Associate Professor Grade 1, Scope, VIT-AP, for her constant guidance, continual encouragement, and understanding; more than all, she taught me patience in my endeavour. My association with her is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Software Training.

I would like to express my gratitude to Chancellor Dr.G.Viswanathan, Vice President Dr. Sankar Vishwathan, Vice Chancellor Dr.Kota Reddy, and Dean Academics Dr.Jagadish Mudiganti, for providing an environment to work in and for his inspiration during the tenure of the course.

In a jubilant mood I express ingeniously my whole-hearted thanks to Programme Chair Dr Reeja, all teaching staff and members working as members of our university for their not-self-centred enthusiasm coupled with timely encouragement showered on me with zeal, which prompted the acquisition of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who helped me to persuade and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Amaravathi

Date: 28-08-2021 **K.Chandana**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

There are no living species on earth who can survive without drinking water. The flow of water from rivers and lakes contributes explicitly or implicitly to both human well-being and the fisheries industry. However, water is frequently polluted because the industry has been growing every year on the back of spiralling demand, and hazardous waste is discharged into the rivers and lakes by those industries. Millions of people die every year, untold losses of income, and agricultural land deteriorates due to water pollution. In recent years, there are several studies have shown that the quality of groundwater has declined significantly in most countries. Therefore, surveillance of water quality is mandatory. Though water quality can be tested using traditional techniques such as collecting the water specimens manually and then analysing it in a laboratory. But it can be considered time consuming and expensive. Sensors can also be regarded as another conventional approach. However, using sensors is costly to test all the water quality variables and often show low precision. Another solution for monitoring water quality is predictive modelling using machine learning and deep learning approaches. Compared to other conventional methods, it has several advantages: lower costs, efficiency in terms of time required for travel and collection, enables prediction under various phases of a system, and predicts desirable values when accessing a site is inconvenient. The researchers had extensively used prediction models for their studies of water quality management systems over the last few years including artificial neural network.

Traditionally, water quality Prediction was manually carried out once in a while, mostly once in six months, by some selective water quality chemists across the country. In this approach, water samples were collected and analysed at laboratories using complex analytical or measurement procedures/methods. Although the manual approach provides several measurement options, the water quality predictions were not carried out at real time or the measurements were not precise owing to the chemical reactions of water samples along with the transporting vessels. A substantial reduction in the price of sensors, including water quality measurement sensors, has shown a drastic improvement in the water quality prediction procedures in recent years.

In recent decades, these methods have had a tremendous impact on the aquatic climate and the animals. But most of the models, including artificial neural network, wavelet neural network, recurrent neural network, and decision tree, required lots of input parameters and computational power, which are considered expensive to construct such models.

With this motivation, this paper used a water quality Prediction (WQP), a combination of different water quality metrics that demonstrates the water quality condition of a particular region, and applied both prediction and classification models to predict water quality prediction and classify the water quality status. Random Forest Algorithm (RFA) is used to predict WQP in this analysis, combining both supervised and unsupervised techniques.

1.1 PURPOSE

The purpose of this project is to predict the quality of water by analysing the Ph value, Capacity of water ,Total dissolved solids ,Amount of Chloramines in ppm ,Electrical conductivity of water ,Amount of organic carbon , Amount ofTrihalomethanes ,Measure of light emitting property of water in NTU (Nephelometric Turbidity Units) , Indicates if water is safe for human consumption.

1.2. INTENDED AUDIENCE

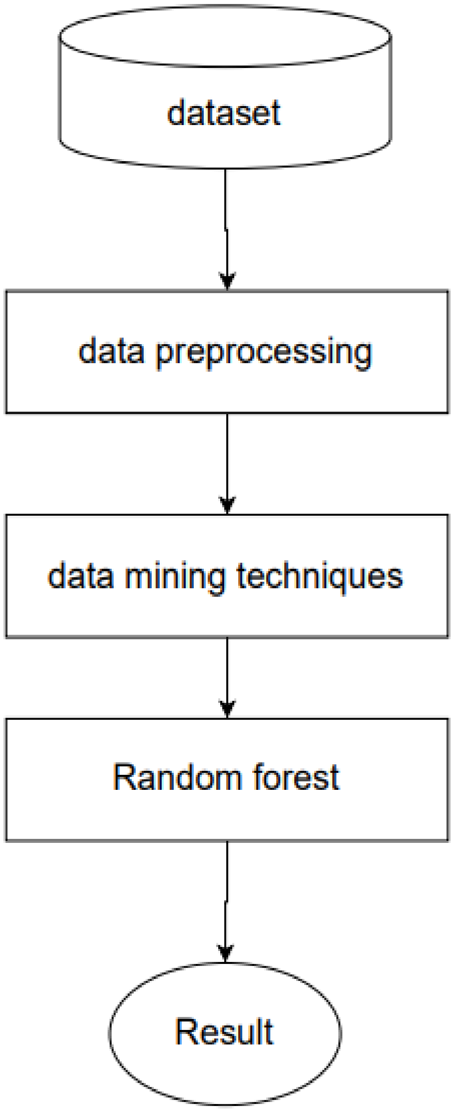
This document is to be read by the development team, the project managers, marketing, staff, testers and document writers. Our stakeholders ,company manufacturing associated hardware, company providing embedded operating systems, shareholders, and distributors who market the finished product, may review the document to learn about the project and to understand the requirement . The report has been organised approximately in order to increase specificity .The developers and project managers need to become intimately familiar with the reports of the project.

1.3. PROJECT SCOPE

Access to safe drinking-water is essential to human health, a basic human right and a component of effective policy for health protection. This is important as a health and development issue at a national, regional and local level issue. In some regions, it has been shown that investments in the water supply and sanitation of the canals can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the intervention.

2. SYSTEM DESIGN AND ARCHITECTURE

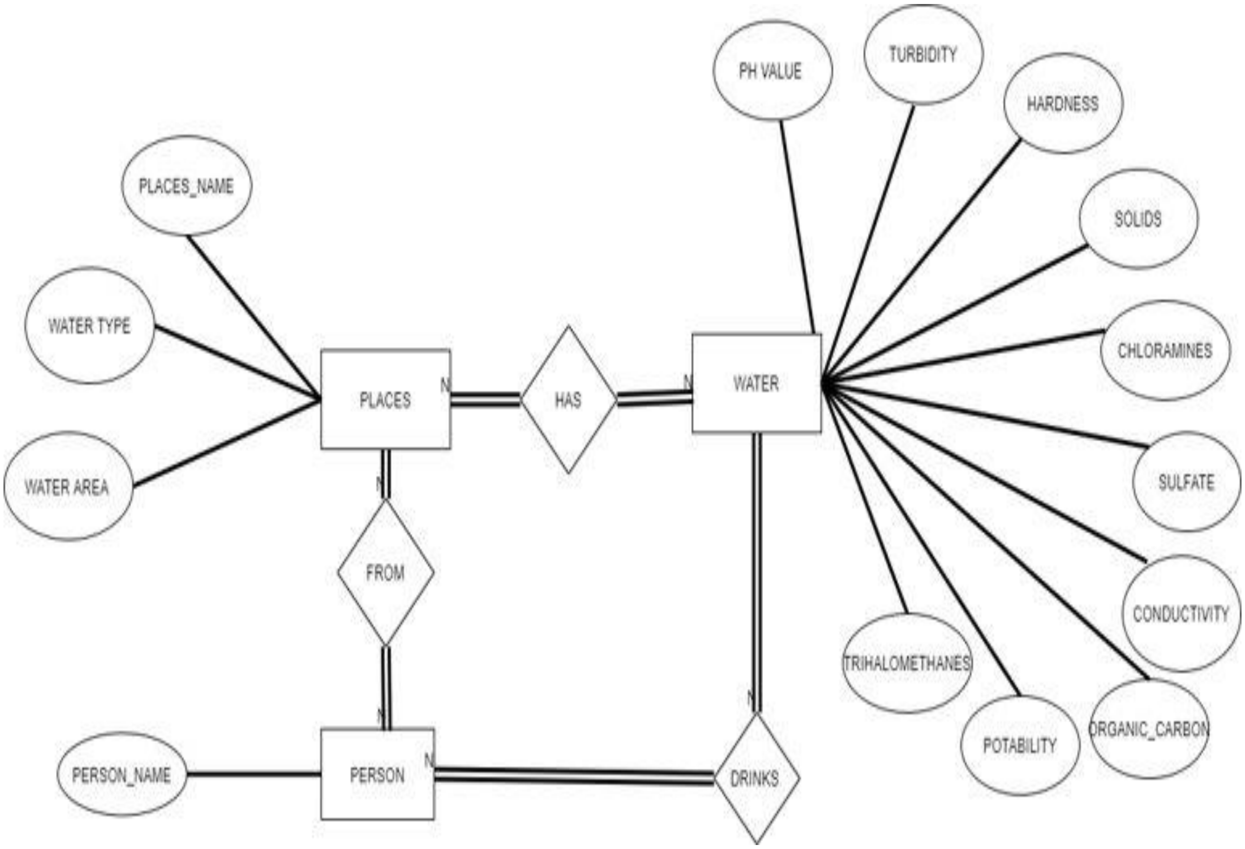
2.1. IMPLEMENTATION



1 Implementation of the project with steps

In figure 1.1 we can see the project implementation in step by step processes.

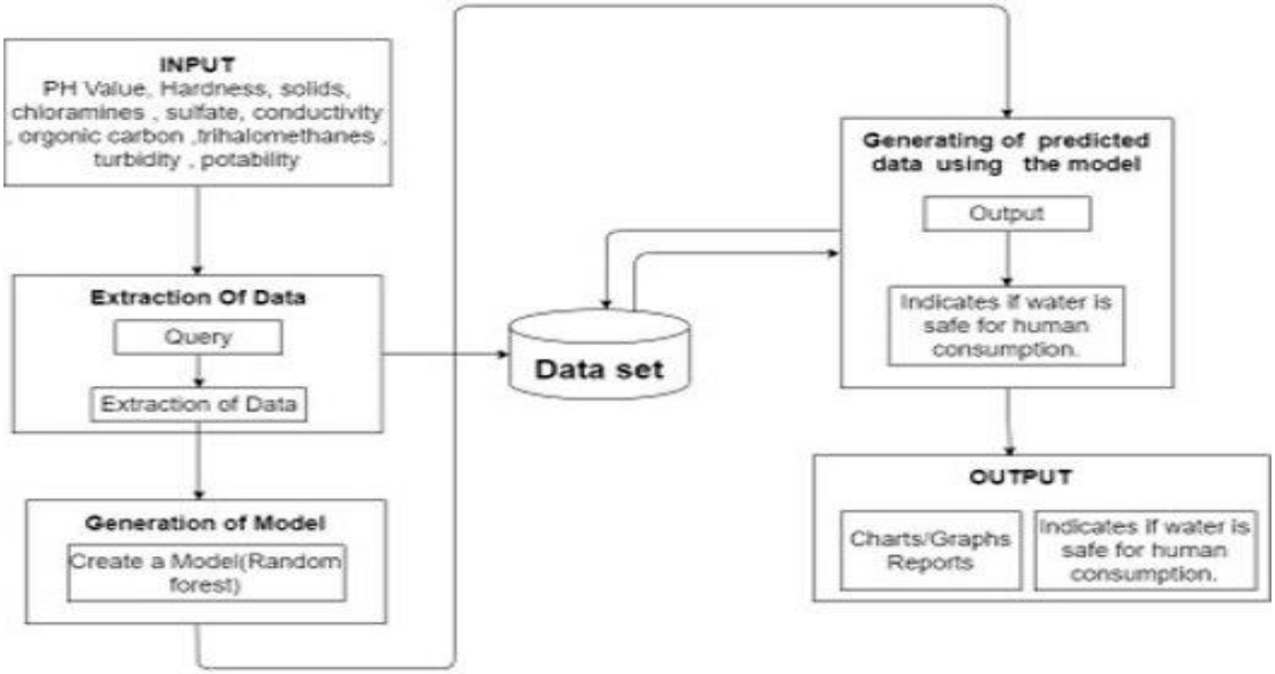
2.2 ER Diagram



2 Entity relationship Diagram

In figure 2 we have shown the relationship between the water and places.

2.3. ARCHITECTURE DIAGRAM


INPUT
PH Value, Hardness, solids, chloramines , sulfate, conductivity , orgonic carbon ,trihalomethanes , turbidity , potability
Extraction Of Data
Query
Extraction of Data
Generation of Model
Create a Model(Random forest)
Data set
Generating of predicted data using the model
Output
Indicates if water is safe for human consumption.
OUTPUT
Charts/Graphs Reports
Indicates if water is safe for human consumption.

3 Architecture Diagram

In the above diagram (3) we can see the process and extrication of data .First we took the dataset then by using the algorithms we find the desired output.

2.4. OPERATING ENVIRONMENT

Programming language	Python
IDE	Google colab
Operating System	Windows XP
Processor	Windows 10,Core i3

1.1 Operating System used for the project

3. SYSTEM INTERFACES

3.1. **Hardware Interfaces:**

Hardware	-	Minimum System Requirements
Processor	-	1.2 GHZ Processor Speed
Memory	-	128 MB RAM (256 preferable)
Disk space	-	60 GIGABYTE
Display	-	800X600 Colours (1024x768 High Colour 16 bit)

1.2 Hardware requirements for the project

3.2Software Interfaces

Software	-	Minimum System Requirements
Operating System	-	Windows
Runtime Server	-	Google colab

1.3 Software requirements for the project

3.3. SYSTEM FEATURES

The groundwater plays a prime role in a country like India. In this paper, we proposed classification algorithms like Random forest with data analytics tool google colab to generate an effective predictive model which predicts whether water is “High” of “Low” for drinking purposes based on water quality parameters. Random forest will produce better results with accuracy and classification error. In future we intend to use more classification algorithms with extended dataset to analyse the ground water quality hence proper water treatment is required in terms of community health.

3.4. MATERIALS AND METHODS

3.4.1 Essentials

1. PH value:

PH is an important parameter in evaluating the acid–base balance of water. It also indicates acidic or alkaline condition of water status. WHO has recommended the maximum permissible limit of pH from 6.5 to 8.5. The current investigation ranges were 6.52–6.83 which are in the range of WHO standards.

2. Hardness:

Hardness is mainly caused by calcium and magnesium salts. These salts are dissolved from geologic deposits through which water travels. The length of time water is in contact with hardness producing material helps determine how much hardness there is in raw water. It is originally defined as the capacity of water to precipitate soap caused by Calcium and Magnesium.

3. Solids (Total dissolved solids - TDS):

Water has the ability to dissolve a wide range of inorganic and some organic minerals such as potassium, calcium, sodium, bicarbonates, chlorides, magnesium, sulphates etc. These minerals produced an unwanted taste and diluted colour in the appearance of water. This is the important parameter for the use of water. The water with high TDS value indicates that water is highly mineralized. The Desired limit for TDS is 500 mg/l and the maximum limit is 1000 mg/l which is prescribed for drinking purpose.

4. Chloramines:

Chlorine and chloramine are the major disinfectants used in public water systems. Chloramines are most commonly formed when ammonia is added to chlorine to treat drinking water. Chlorine levels up to 4 milligrams per litre (mg/L or 4 parts per million (ppm)) are considered safe in drinking water.

5. Sulphate:

Sulphates are naturally occurring substances that are found in minerals, soil, and rocks. They are present in ambient air, groundwater, plants, and food. The principal commercial use of sulphate is in the chemical industry. Sulphate concentration in seawater is about 2,700 milligrams per litre (mg/L). It ranges from 3 to 30 mg/L in most freshwater supplies, although much higher concentrations (1000 mg/L) are found in some geographic locations.

6. Conductivity:

Pure water is not a good conductor of electric current rather it's a good insulator. Increase in ions concentration enhances the electrical conductivity of water. Generally, the amount of dissolved solids in water determines the electrical conductivity. Electrical conductivity (EC) actually measures the ionic process of a solution that enables it to transmit current. According to WHO standards, EC value should not exceed 400 μS/cm.

7. Organic carbon:

Total Organic Carbon (TOC) in source waters comes from decaying natural organic matter (NOM) as well as synthetic sources. TOC is measured by the total amount of carbon in organic compounds in pure water. According to the US EPA < 2 mg/L as TOC in treated / drinking water, and < 4 mg/Lit in source water which is used for treatment.

8. Trihalomethanes:

THMs are chemicals which may be found in water treated with chlorine. The concentration of THMs in drinking water varies according to the level of organic material in the water, the amount of chlorine required to treat the water, and the temperature of the water that is being treated. THM levels up to 80 ppm is considered safe in drinking water.

9. Turbidity:

The turbidity of water depends on the quantity of solid matter present in the suspended state. It is a measure of light emitting properties of water and the test is used to indicate the quality of waste discharge with respect to colloidal matter. The mean turbidity value obtained for Wando Genet Campus (0.98 NTU) is lower than the WHO recommended value of 5.00 NTU.

10. Portability:

Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable.

3.4.2 Algorithms

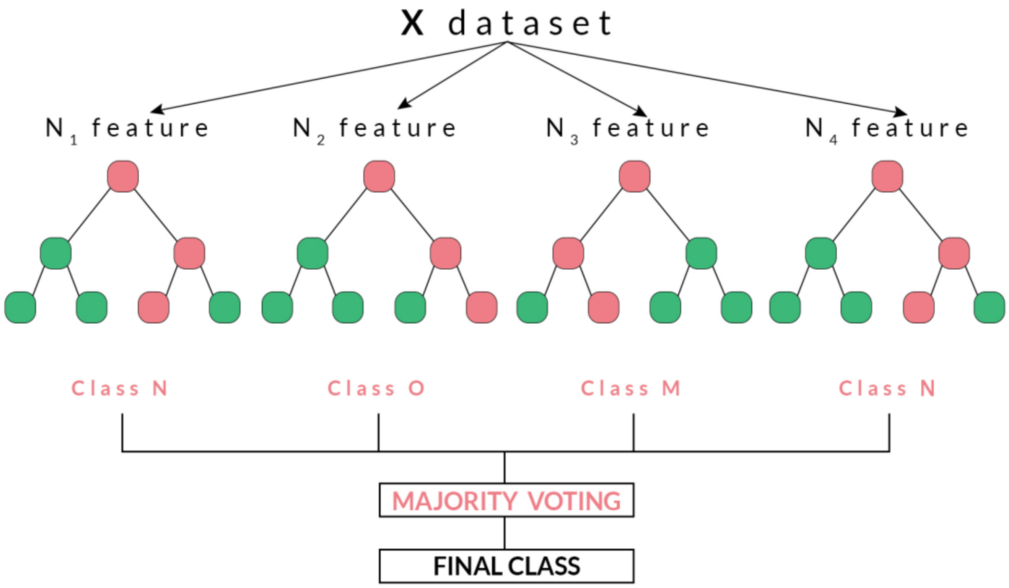
RANDOM FOREST ALGORITHM

Random forest is a supervised learning algorithm used for both classification as well as regression purposes. But it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, a random forest algorithm creates decision trees on data samples and then gets the prediction from each of the columns and finally selects the best solution by means of voting. It is a group of items ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps −

- Step 1 − First, we start with the selection of random samples from a dataset.
- Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- 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.

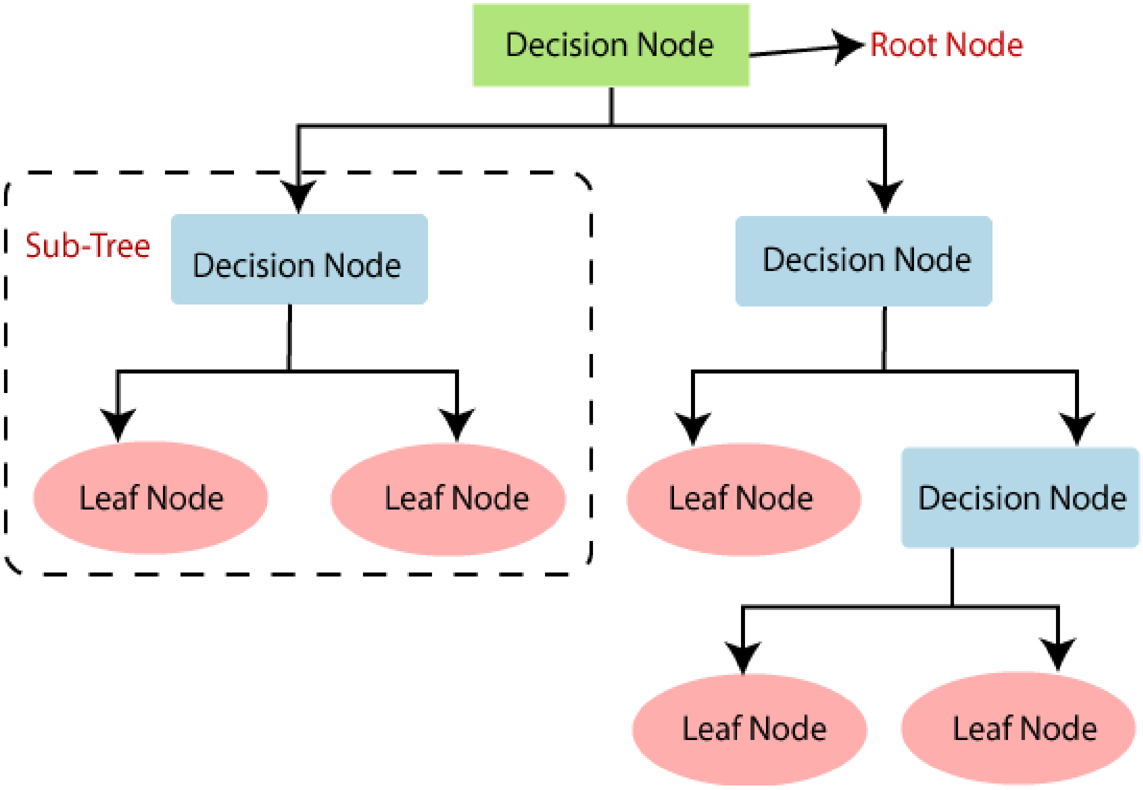


4 Working principle of random forest

DECISION TREE ALGORITHM

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- In a Decision tree there are two nodes, which are Decision Node and Leaf Node. The Decision nodes are used to make any decision and they have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.



5 Working processes of Decision tree

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: start the tree with the root node, says S, which contains the complete dataset of the project.
- Step-2: Next, Find the best attribute in the dataset by using Attribute Selection Measure (ASM).
- Step-3: Divide the root node into subsets that contain possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3, then continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

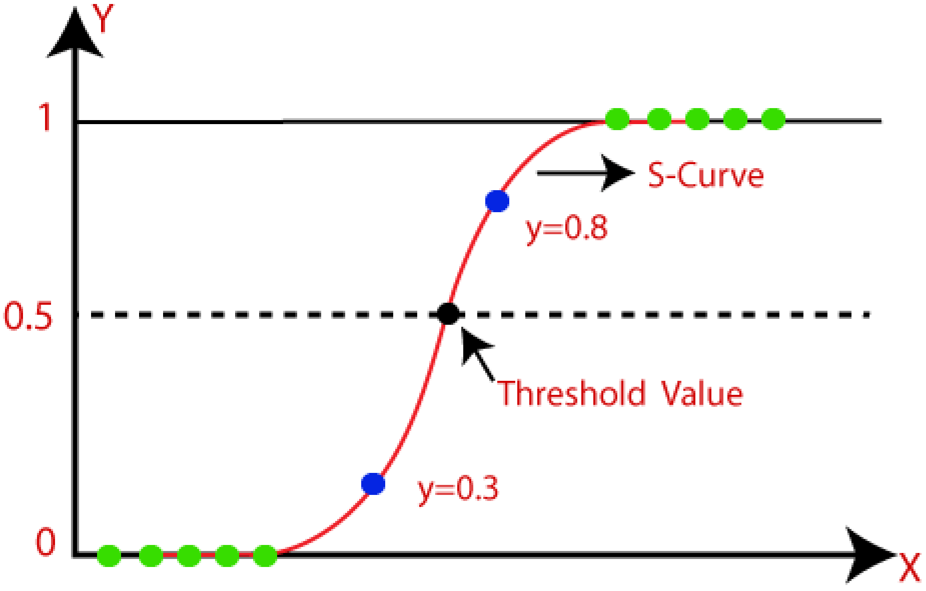
Logistic Regression Algorithm

A logistic regression algorithm is a machine learning regression algorithm which measures the ways in which a set of data conforms to two particular variables. The algorithm dictates the variables, the relationship, and the ways in which the variables interact. The most common form of a logistic regression algorithm is a binomial algorithm. This type of the algorithm has two particular outputs which can result from the function. The algorithm places the data set into one of these areas and then maps changes in the data set over time and plots it. This map represents a curve that displays the relationships inherent in the data set. There are also more complicated forms of logistic regression that display multiple variables.

Logistic regression is different from linear regression in that it represents a curve with a changing slope. Linear regression is more fixed and unchanging. It is more focused on drawing a line that fits the means of a data set than drawing a curve which reflects the relationship between variables. This process of logistic regression is not only applicable to an existing data set. It may also be used to predict future behaviour of the dataset.

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps of the algorithm:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.



6 Logistic Regression

4. CODE AND OUTPUT

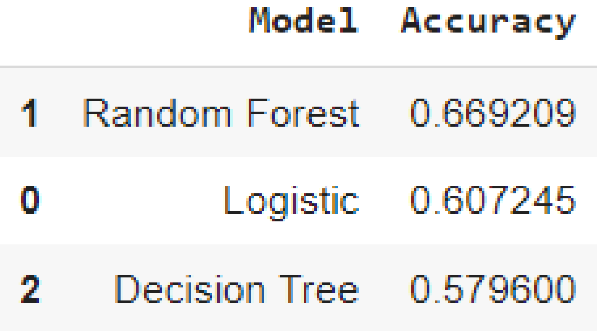
4.1PREDICTION MODEL

Data-Pre-processing: The dataset included some null values. For handling such null values, the mean method is used in this analysis. Furthermore, Min–Max scalar is used to scale the data, which makes the computation easier.

Data-Split: The collected data are then divided into two sets on the dataset: training and testing set with a proportion of 80 and 20 percent.

Algorithm: For predicting the WQP, machine learning algorithms can be used. In this analysis, several machine learning algorithms are used, including Linear Regression, Decision Tree, Random Forest Regression, and Support Vector Regression. Finally, the best prediction model is selected by comparing the performances of such models. Table 3 presents the experimental parameters used for those models.

Table: 1.4 Result



	Model	Accuracy
1	Random Forest	0.669209
0	Logistic	0.607245
2	Decision Tree	0.579600

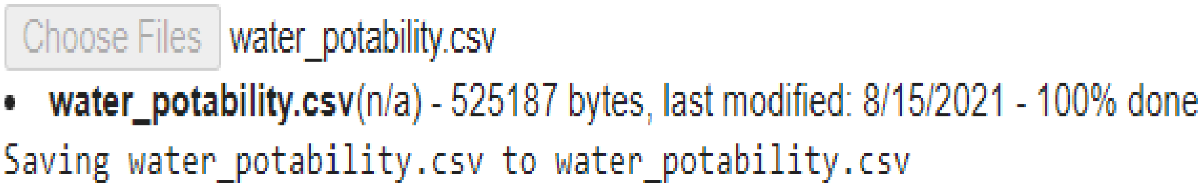
4.2. CODE AND OUTPUT

Here I am importing all the libraries that are used to predict the water

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

Here I upload the dataset

```
from google.colab import files
uploaded=files.upload()
```



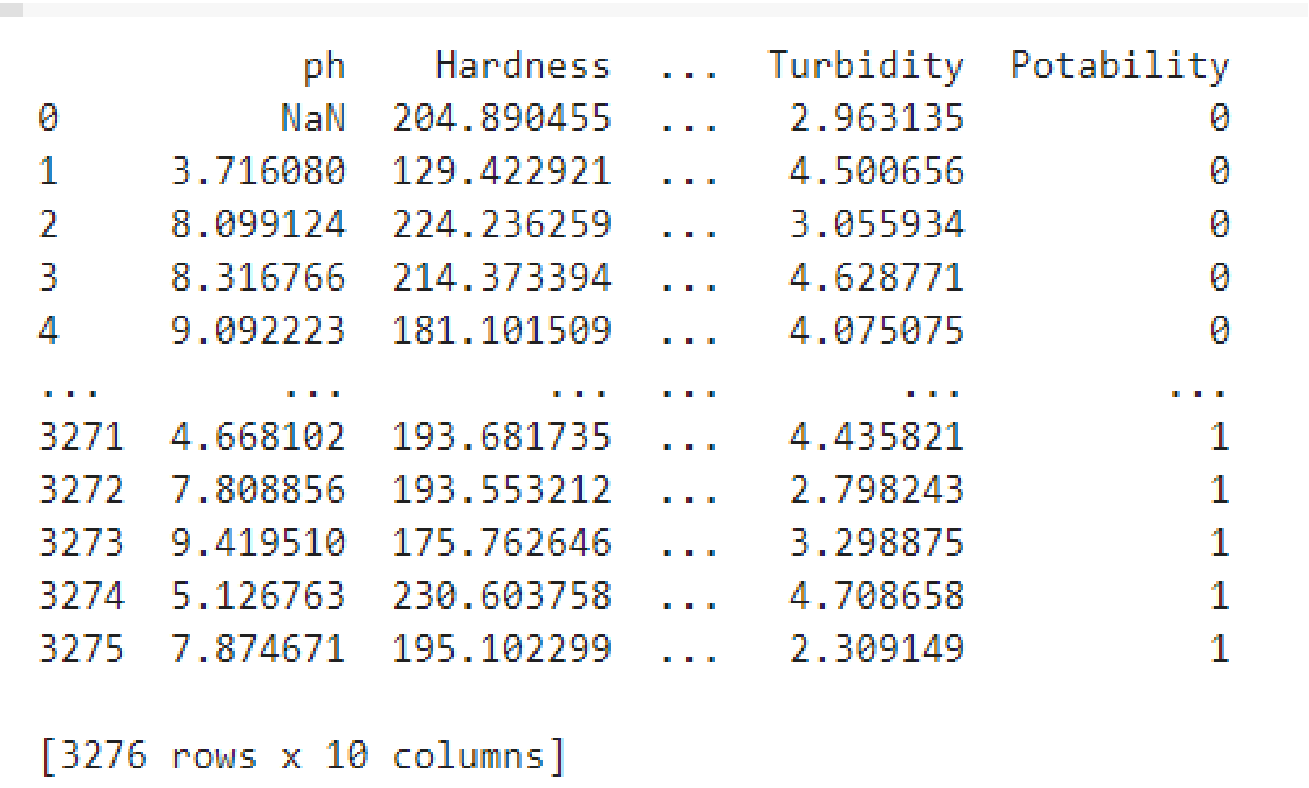
```
water_potability.csv
• water_potability.csv(n/a) - 525187 bytes, last modified: 8/15/2021 - 100% done
Saving water_potability.csv to water_potability.csv
```

Here I print the data set and the data set has 10 rows and 3276 columns

```
import io
```

df = pd.read_csv(io.BytesIO(uploaded['water_potability.csv']))

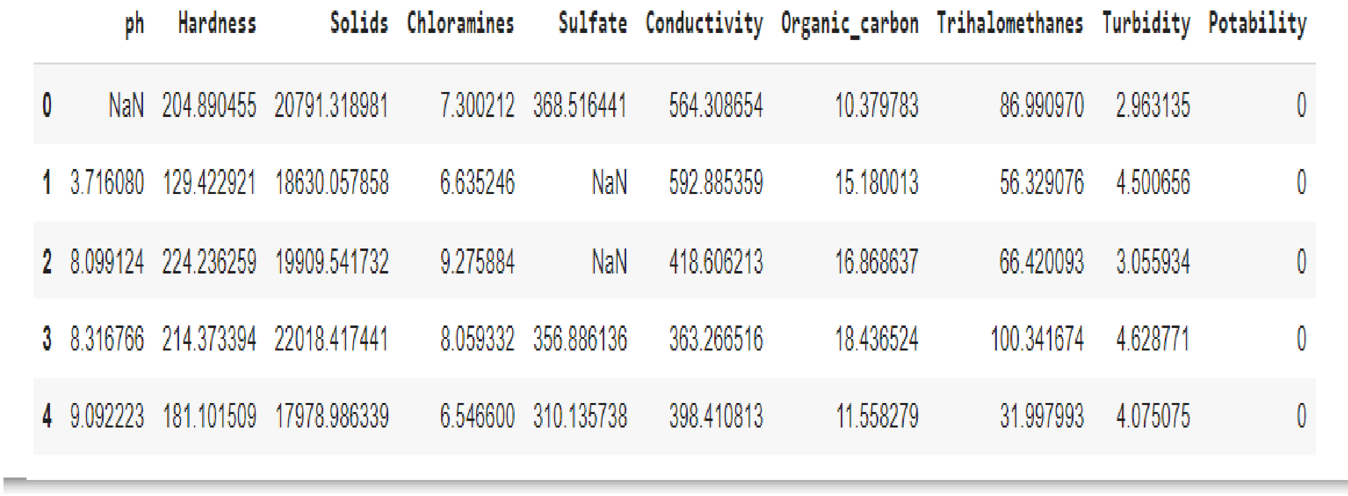
print(df)



```
             ph    Hardness  ...  Turbidity  Potability
0           NaN  204.890455  ...   2.963135           0
1      3.716080  129.422921  ...   4.500656           0
2      8.099124  224.236259  ...   3.055934           0
3      8.316766  214.373394  ...   4.628771           0
4      9.092223  181.101509  ...   4.075075           0
...         ...         ...  ...        ...         ...
3271   4.668102  193.681735  ...   4.435821           1
3272   7.808856  193.553212  ...   2.798243           1
3273   9.419510  175.762646  ...   3.298875           1
3274   5.126763  230.603758  ...   4.708658           1
3275   7.874671  195.102299  ...   2.309149           1

[3276 rows x 10 columns]
```

Here I am taking only head part to make my prediction easy

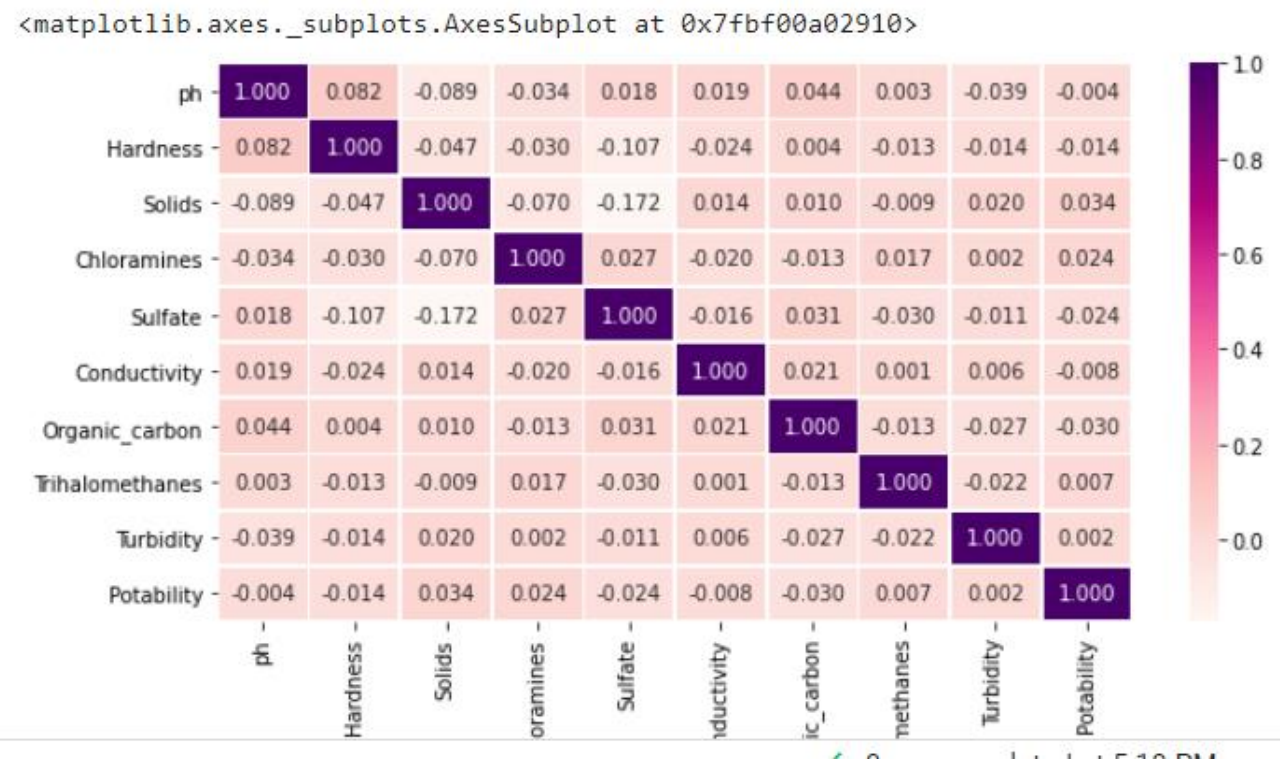
df.head()



	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

Here I plot the graph using heat map to represent the data in 2 dimension form.

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True, cmap="RdPu",fmt='.3f',linewidths=.8)
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf00a02910>
```

	ph	Hardness	Solids	oramines	Sulfate	nductivity	ic_carbon	methanes	Turbidity	Potability
ph	1.000	0.082	-0.089	-0.034	0.018	0.019	0.044	0.003	-0.039	-0.004
Hardness	0.082	1.000	-0.047	-0.030	-0.107	-0.024	0.004	-0.013	-0.014	-0.014
Solids	-0.089	-0.047	1.000	-0.070	-0.172	0.014	0.010	-0.009	0.020	0.034
Chloramines	-0.034	-0.030	-0.070	1.000	0.027	-0.020	-0.013	0.017	0.002	0.024
Sulfate	0.018	-0.107	-0.172	0.027	1.000	-0.016	0.031	-0.030	-0.011	-0.024
Conductivity	0.019	-0.024	0.014	-0.020	-0.016	1.000	0.021	0.001	0.006	-0.008
Organic_carbon	0.044	0.004	0.010	-0.013	0.031	0.021	1.000	-0.013	-0.027	-0.030
Trihalomethanes	0.003	-0.013	-0.009	0.017	-0.030	0.001	-0.013	1.000	-0.022	0.007
Turbidity	-0.039	-0.014	0.020	0.002	-0.011	0.006	-0.027	-0.022	1.000	0.002
Potability	-0.004	-0.014	0.034	0.024	-0.024	-0.008	-0.030	0.007	0.002	1.000

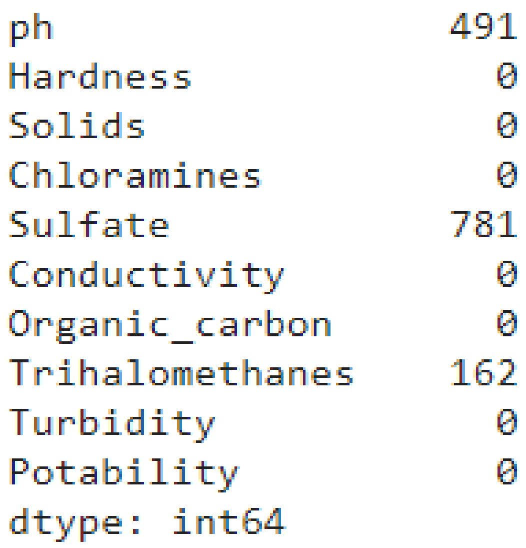
In this step I found if there are any duplicate values

```
dup = df.duplicated().sum()
print('Any Duplicate Value:',dup)
```

```
Any Duplicate Value: 0
```

Here I want to found how many null values are present in each column

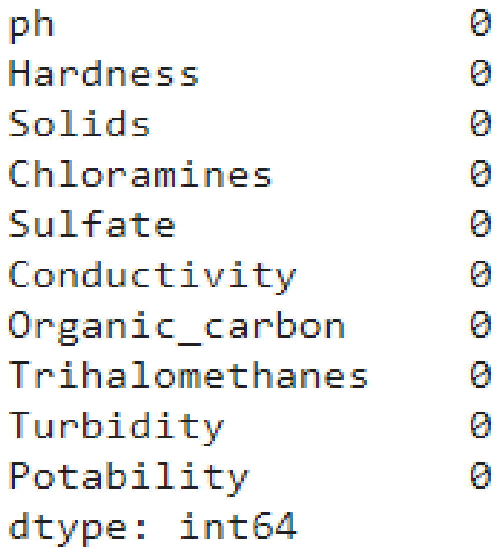
```
df.isnull().sum()
```



```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

In this step I filled the missing values by using finding mean method

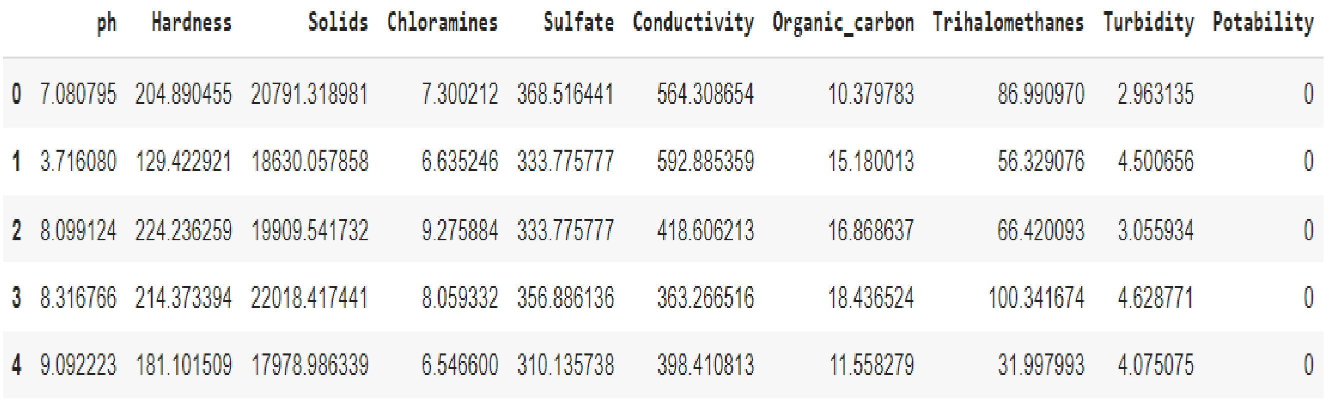
```
df["ph"].fillna(value = df["ph"].mean(), inplace = True)
df["Sulfate"].fillna(value = df["Sulfate"].mean(), inplace = True)
df["Trihalomethanes"].fillna(value = df["Trihalomethanes"].mean(), inplace = True)
df.isnull().sum()
```



```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

Here I print the head of the data after filling the missing values using mean

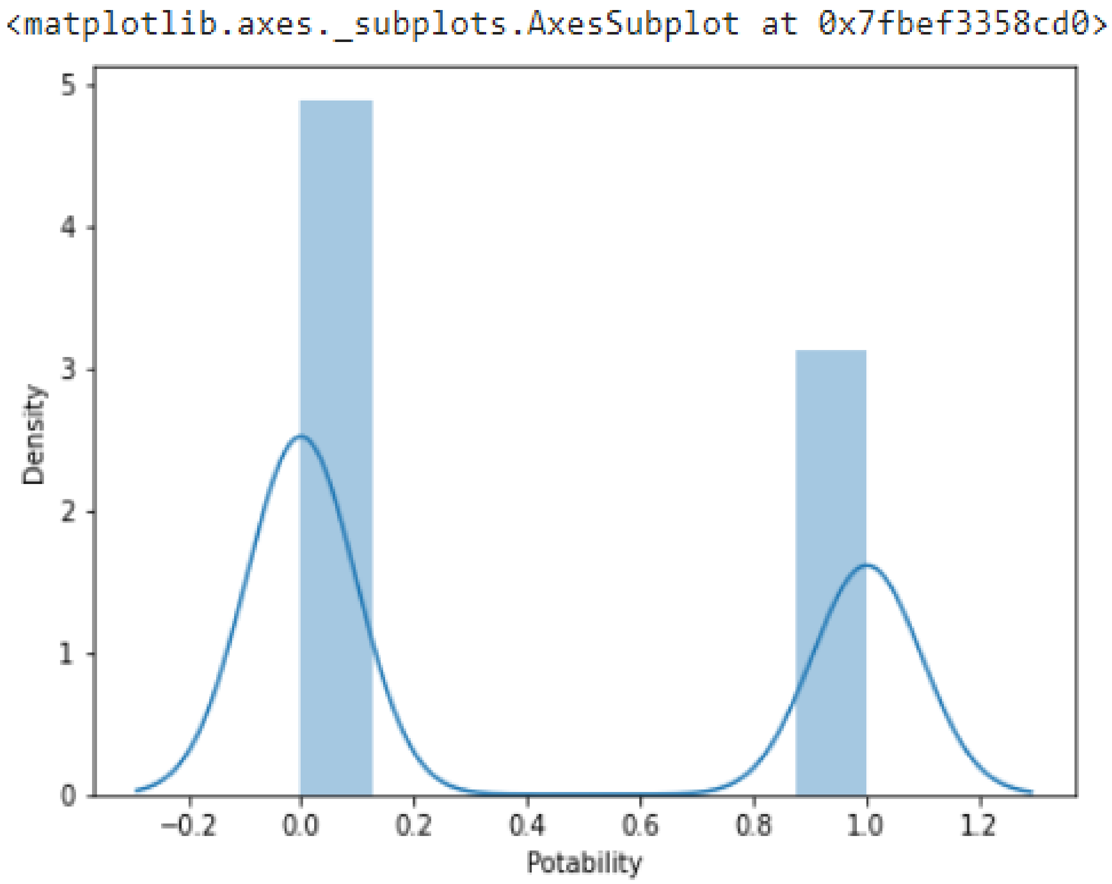
df.head()



	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.080795	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	333.775777	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	333.775777	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

Here I just plot the graph by taking density on y-axis and potability on x-axis

plt.rcParams['figure.figsize'] = [7,5]
sns.distplot(df['Potability'])

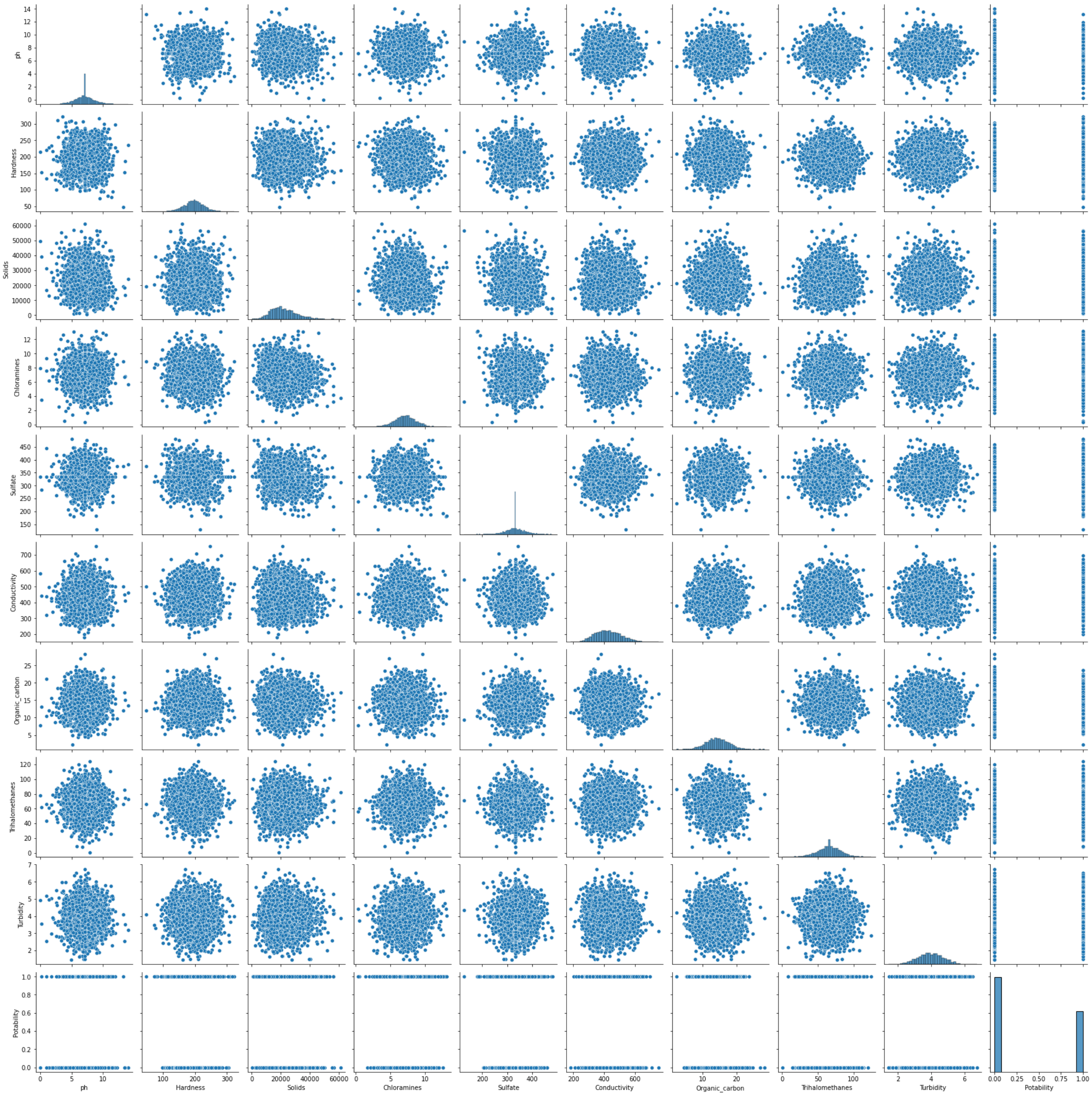


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbef3358cd0>
```

Density
0
1
2
3
4
5
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Potability

Here I plot a pair plot

sns.pairplot(data = df)

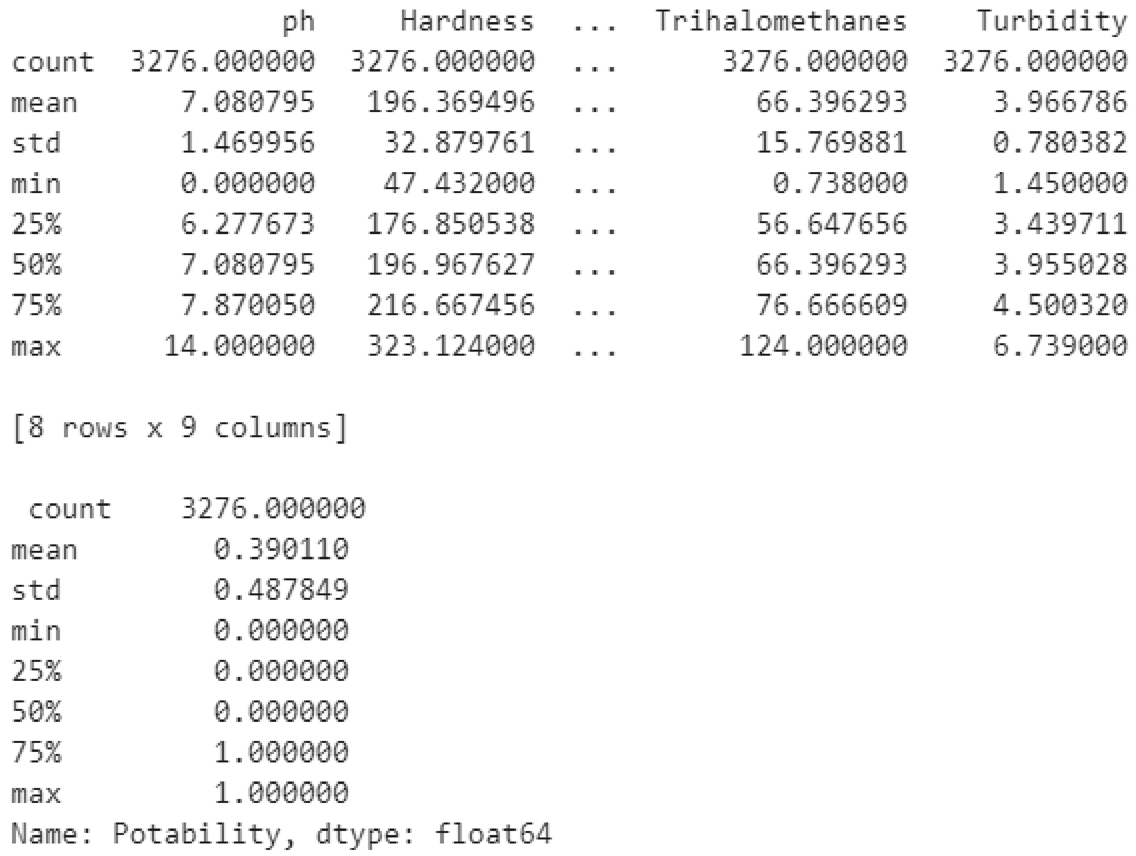


After the pairplot I divided the dataset into train and test data

```
x = df.drop(['Potability'],axis=True)
y = df['Potability']
x_train,x_test,y_train,y_test = tts(x,y,test_size=0.32, random_state = 50)
print(x.describe(),"\n","\n", y.describe())
```

After that I used logistic regression to find the accuracy

```
logi = LogisticRegression(ma
```

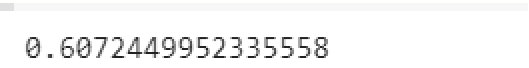


```
                ph     Hardness  ...  Trihalomethanes   Turbidity
count  3276.000000  3276.000000  ...      3276.000000  3276.000000
mean      7.080795   196.369496  ...        66.396293     3.966786
std       1.469956    32.879761  ...        15.769881     0.780382
min       0.000000    47.432000  ...         0.738000     1.450000
25%       6.277673   176.850538  ...        56.647656     3.439711
50%       7.080795   196.967627  ...        66.396293     3.955028
75%       7.870050   216.667456  ...        76.666609     4.500320
max      14.000000   323.124000  ...       124.000000     6.739000

[8 rows x 9 columns]

 count    3276.000000
mean        0.390110
std         0.487849
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
Name: Potability, dtype: float64
```

Here I print the accuracy gained by the Logistic regression

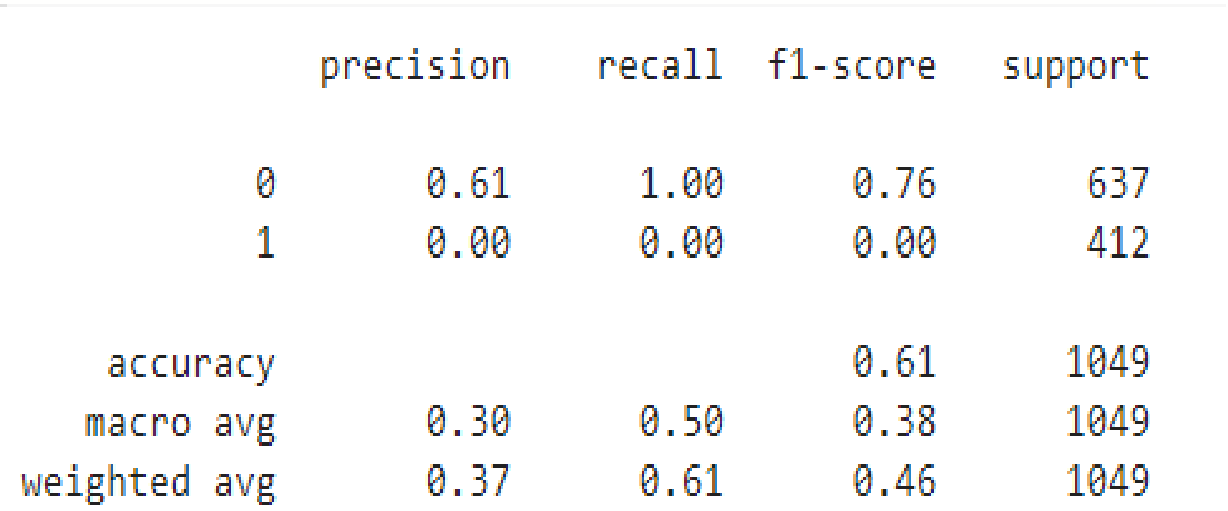
```
x_iter = 120, random_state=0,n_jobs=20)
logi.fit(x_train,y_train)
```

```
predlogi_y = logi.predict(x_test)
Acc= accuracy_score(predlogi_y,y_test)
print( Acc)
```



```
0.6072449952335558
```

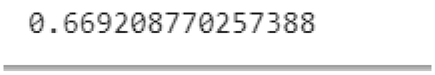
```
print(classification_report(y_test,predlogi_y))
```



```
              precision    recall  f1-score   support

           0       0.61      1.00      0.76       637
           1       0.00      0.00      0.00       412

    accuracy                           0.61      1049
   macro avg       0.30      0.50      0.38      1049
weighted avg       0.37      0.61      0.46      1049
```

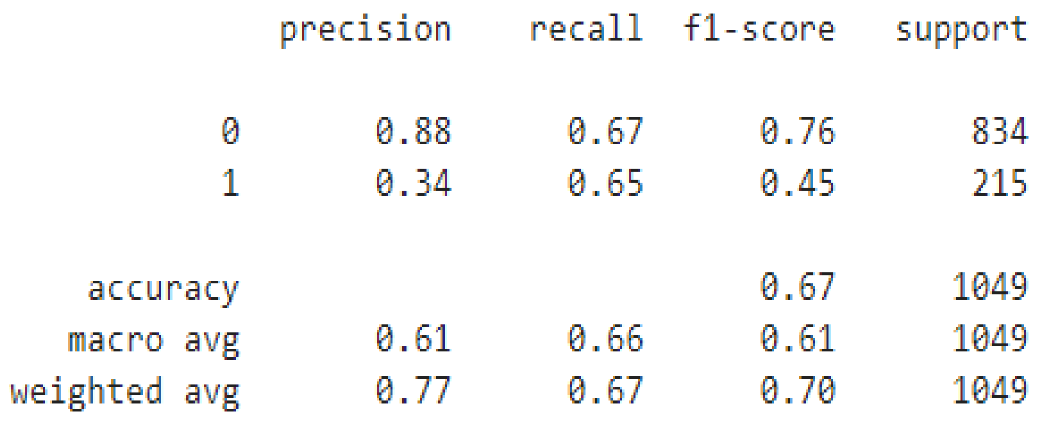
Here I used the Random Forest Classifier to know the accuracy

```
RFC = RandomForestClassifier()
RFC.fit(x_train,y_train)
y_RFC = RFC.predict(x_test)
Acc_rfc= accuracy_score(y_RFC,y_test)
print( Acc_rfc)
```



```
0.669208770257388
```

```
print(classification_report(y_RFC,y_test))
```

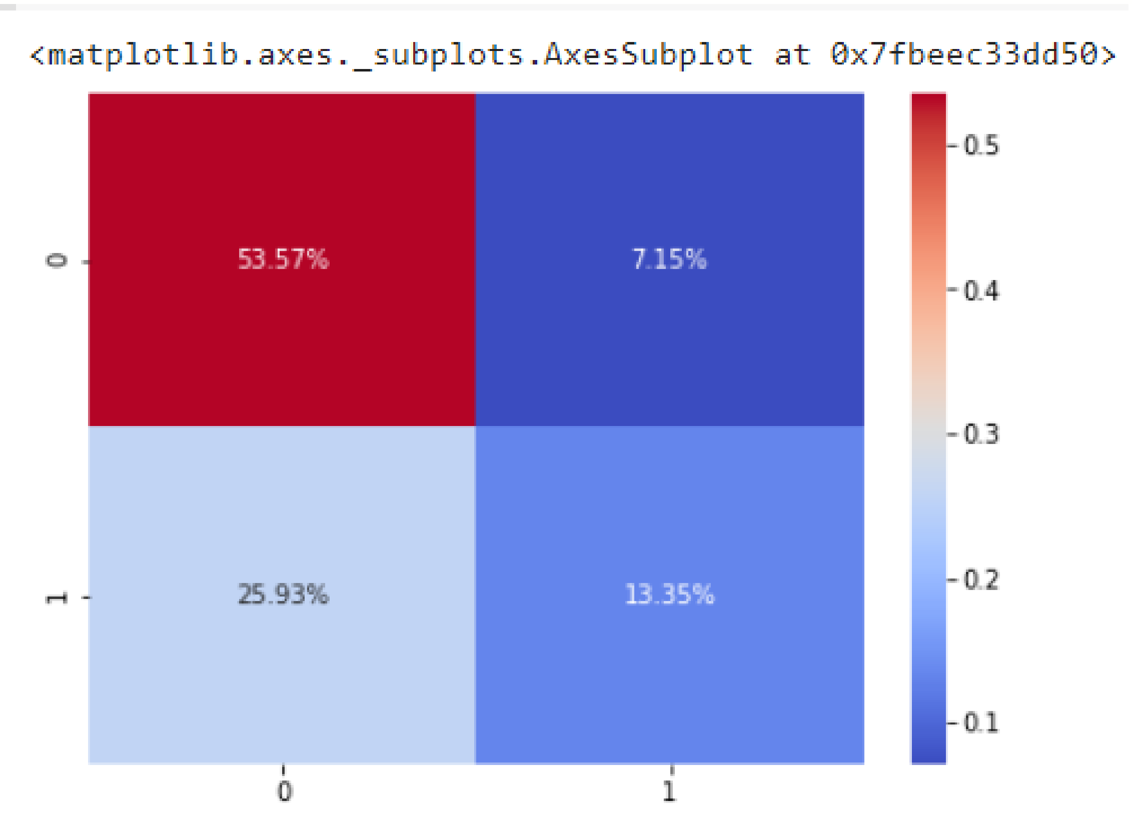


```
              precision    recall  f1-score   support

           0       0.88      0.67      0.76       834
           1       0.34      0.65      0.45       215

    accuracy                           0.67      1049
   macro avg       0.61      0.66      0.61      1049
weighted avg       0.77      0.67      0.70      1049
```

Here I want to draw a confusion matrix by using heatmap

cmr= confusion_matrix(y_test,y_RFC)

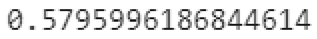
sns.heatmap(cmr/np.sum(cmr), annot= True, fmt= '0.2%', cmap= 'coolwarm')



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbeec33dd50>
```

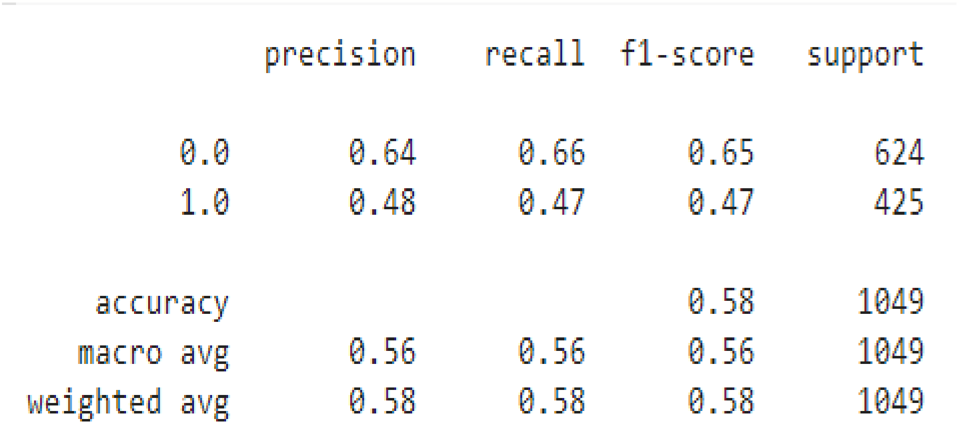
Here I found the accuracy by using Decision Tree Regression

```
DTR = DecisionTreeRegressor()
DTR.fit(x_train,y_train)
y_pred = DTR.predict(x_test)
Acc_dt= accuracy_score(y_pred,y_test)
print( Acc_dt)
```



```
0.5795996186844614
```

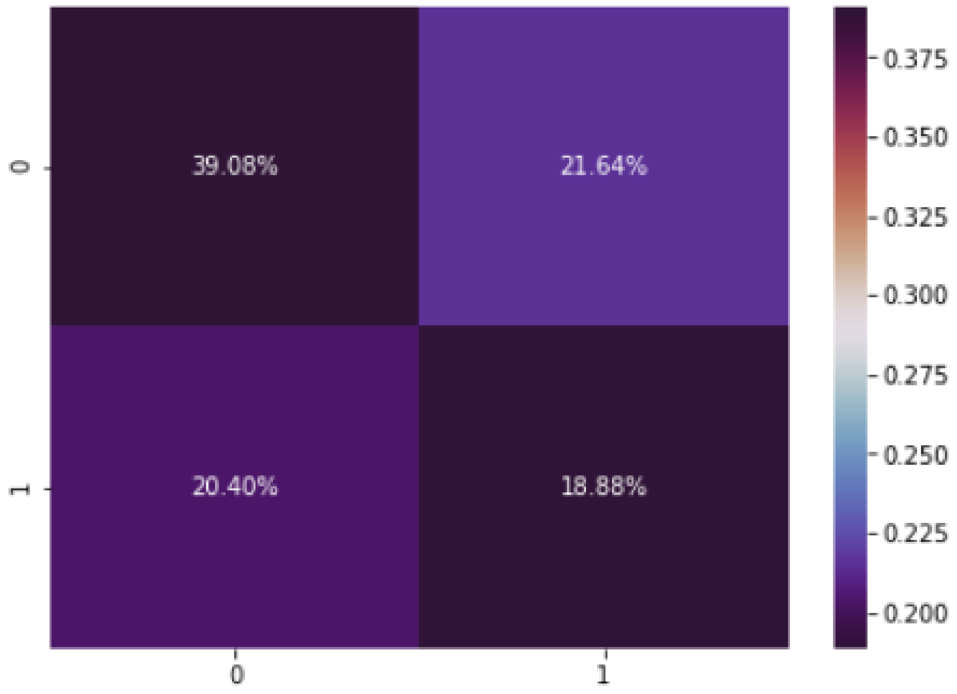
```
print(classification_report(y_pred,y_test))
```



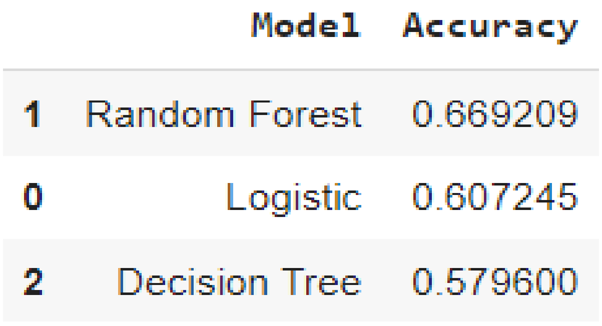
```
              precision    recall  f1-score   support

         0.0       0.64      0.66      0.65       624
         1.0       0.48      0.47      0.47       425

    accuracy                           0.58      1049
   macro avg       0.56      0.56      0.56      1049
weighted avg       0.58      0.58      0.58      1049
```

```
cmd= confusion_matrix(y_test,y_pred)
sns.heatmap(cmd/np.sum(cmd), annot= True, fmt= '0.2%', cmap=
'twilight_shifted')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf00a27950>
```



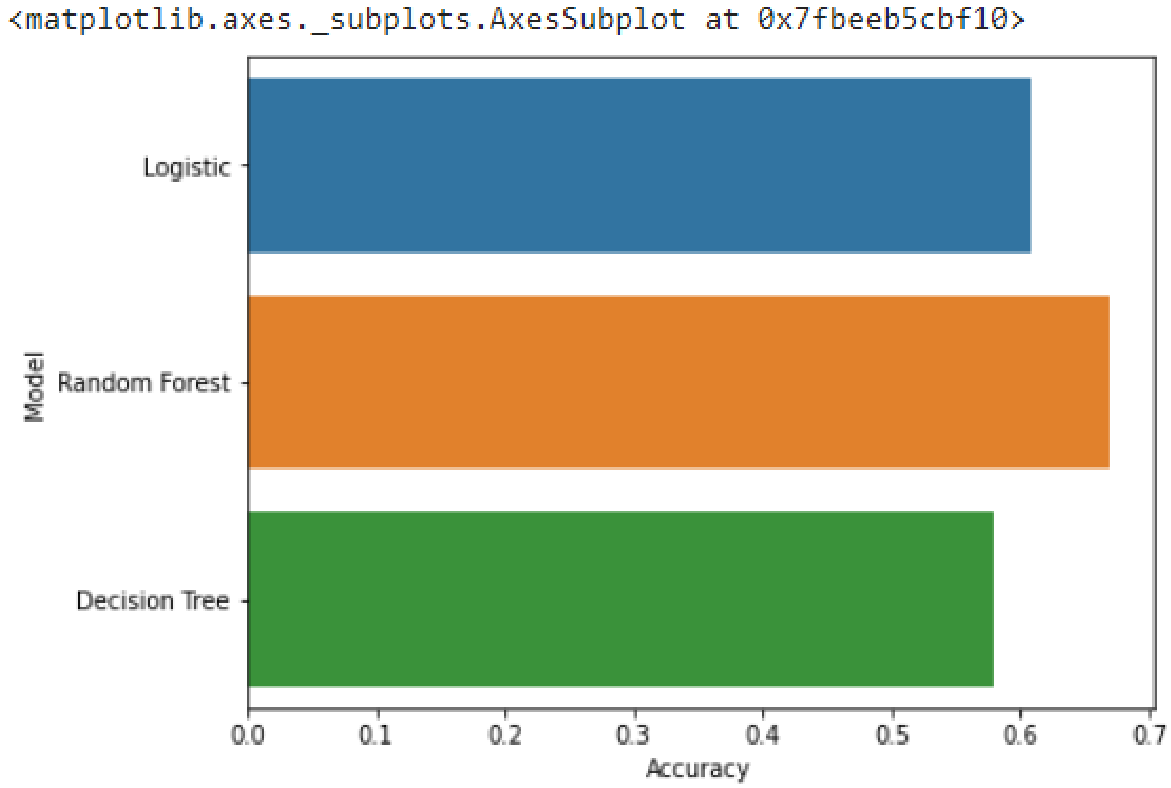
```
models = pd.DataFrame({"Model":['Logistic','Random Forest','Decision Tree'],
          "Accuracy":[Acc,Acc_rfc,Acc_dt]})#Creat Data.....
models.sort_values(by='Accuracy', ascending=False)
```



	Model	Accuracy
1	Random Forest	0.669209
0	Logistic	0.607245
2	Decision Tree	0.579600

Here I visualise the accuracy of the logistic regression,Random forest and decision tree

sns.barplot(x= 'Accuracy', y= 'Model', data= models)



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbeeb5cbf10>
```

5. CONCLUSION & FUTURE WORK

This paper demonstrated a method for predicting and classifying the water quality using machine learning algorithms. The water metrics, including PH, DO, SS, EC, Turbidity, Chloride, COD, TDS, and Alkalinity, were used in this study. For data pre-processing, the median technique used to handle the null values and min–max scalar to scale the data. For the prediction purpose, we applied the Random forest Algorithm (RFA), Linear Regression methods. After analysing the performance of multiple models, Random Forest with Support Vector Regression seems to be more effective with an accuracy of 66%. However, if the number of components is reduced, then RFA with the Multiple Linear Regression model proved to be more effective. Besides, to check the performance of the model, the proposed model is compared with several state-of-art classifiers, including Decision Tree Classifier, Support Vector Classifier, and Random Forest Classifier. Experimental results showed that the Random Forest Classifier classified water quality status more efficiently. Despite the achievements outlined in this paper, some improvements are still possible, including we can collect more training samples to make the model more stable and more progress is possible on the prediction model. Those issues will be overcome in future research, perhaps by proper tuning of the RFA model.

In future we intend to use more classification algorithms with extended data sets to analyse the ground water quality.

References

1. Water quality prediction using machine learning methods | Water Quality Research Journal | IWA Publishing (iwaponline.com)

2. Water Quality Prediction (7 model) | Kaggle

3. Prediction of estuarine water quality using interpretable machine learning approach - ScienceDirect