**WATER QUALITY ANALYSIS**

**PHASE : DEVELOPMENT PART 2**

**INTRODUCTION**

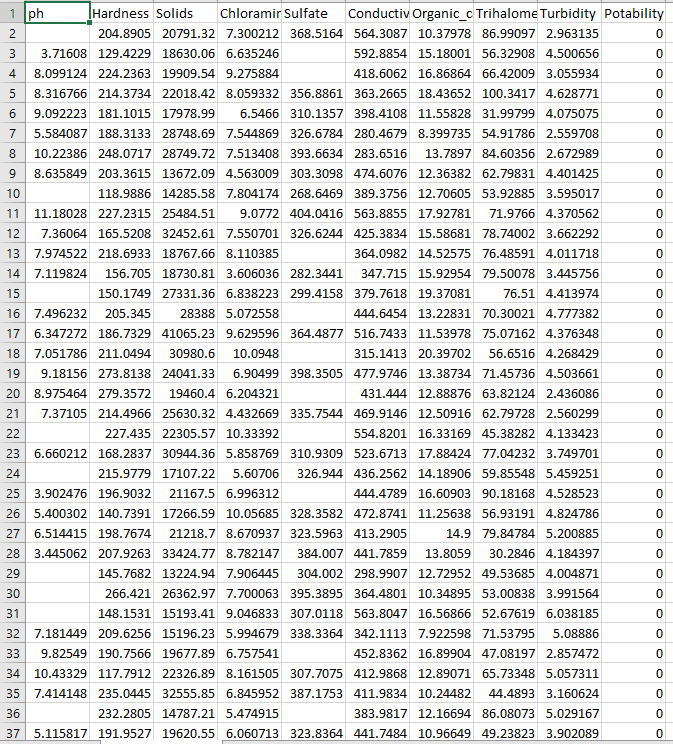
Water quality analysis is a crucial process that plays a fundamental role in ensuring the safety and sustainability of our planet's most precious resource - water. It involves the systematic examination of various physical, chemical, and biological characteristics of water to determine its suitability for various purposes, including drinking, recreation, agriculture, and industrial use. As the world's population continues to grow and environmental pressures increase, understanding and monitoring water quality has become more important than ever.

Water quality analysis enables us to assess the presence of contaminants, pollutants, and pathogens in water bodies, which can have a significant impact on both human health and the ecosystem. It helps us identify potential threats to water sources and informs decisions regarding water treatment and conservation strategies. By providing valuable data, water quality analysis empowers policymakers, scientists, and environmentalists to make informed choices and develop effective solutions to address water-related challenges.

This introduction sets the stage for a deeper exploration of the methods, tools, and significance of water quality analysis in safeguarding our environment, health, and the sustainability of one of our most vital resources.

Dataset Used:

<https://www.kaggle.com/datasets/adityakadiwal/water-potability/data>



In this phase we have to building the analysis by creating visualizations and building a predictive model.

* Use visualization libraries (e.g., Matplotlib, Seaborn) to create histograms, scatter plots, and correlation matrices.
* Build a predictive model (e.g., Logistic Regression, Random Forest) to determine water potability based on water quality parameters.

Histogram:

A **Histogram** is a graphical representation of the distribution of data. The histogram is represented by a set of rectangles, adjacent to each other, where each bar represent a kind of data.

Scatter Plot:

Scatter plots are the graphs that present the relationship between two variables in a data-set. It represents data points on a two-dimensional plane or on a Cartesian system

Correlation Matrices :

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

Logistic Regression:

Logistic regression is a [supervised machine learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) algorithm mainly used for [classification](https://www.geeksforgeeks.org/getting-started-with-classification/) tasks where the goal is to predict the probability that an instance of belonging to a given class

Random Forest:

The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample

# Import Libraries

import pandas as pd *# data processing, CSV file I/O*

import numpy as np *# linear algebra*

import matplotlib.pyplot as plt

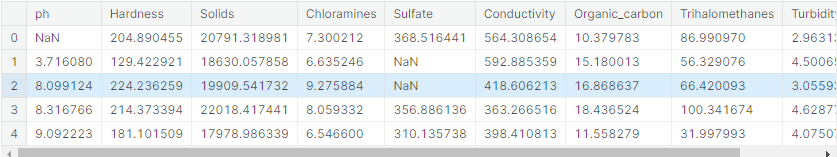
import seaborn as sns

import plotly.express as px

from sklearn.preprocessing import StandardScaler

data=pd.read\_csv('/kaggle/input/water-quality-and-potability/water\_potability.csv')

data.head()



**Data Cleaning:**

we have null values so for the right solution we have to clean this values

data.info()

o/p

RangeIndex: 3276 entries, 0 to 3275

Data columns (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 ph 2785 non-null float64

1 Hardness 3276 non-null float64

2 Solids 3276 non-null float64

3 Chloramines 3276 non-null float64

4 Sulfate 2495 non-null float64

5 Conductivity 3276 non-null float64

6 Organic\_carbon 3276 non-null float64

7 Trihalomethanes 3114 non-null float64

8 Turbidity 3276 non-null float64

9 Potability 3276 non-null int64

We have 3276 rows and there is a 781 row missing value surface data this is highest we already have small data and there is almost 1000 rows is null value . we remove the null values permanently

data.isnull().sum()

o/p

ph 491

Hardness 0

Solids 0

Chloramines 0

Sulfate 781

Conductivity 0

Organic\_carbon 0

Trihalomethanes 162

Turbidity 0

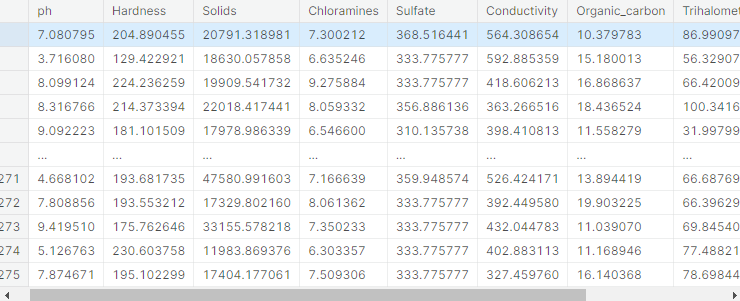
Potability 0

dtype: int64

so where ever ph=null it will get replaced with 7.080795, sulfate=null get replaced with 333.775777, Trihalomethanes=null get replaced with 66.396293

data.fillna(data.mean(),inplace=True)

Data # to print data without null

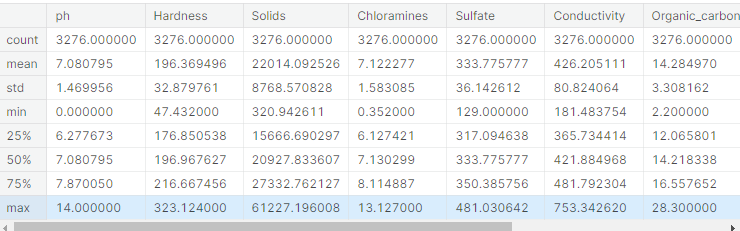


# **Exploratoy Data Analysis**

Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.

data.describe()

o/p:



**Checking if we need to do Dimensonility Reduction**

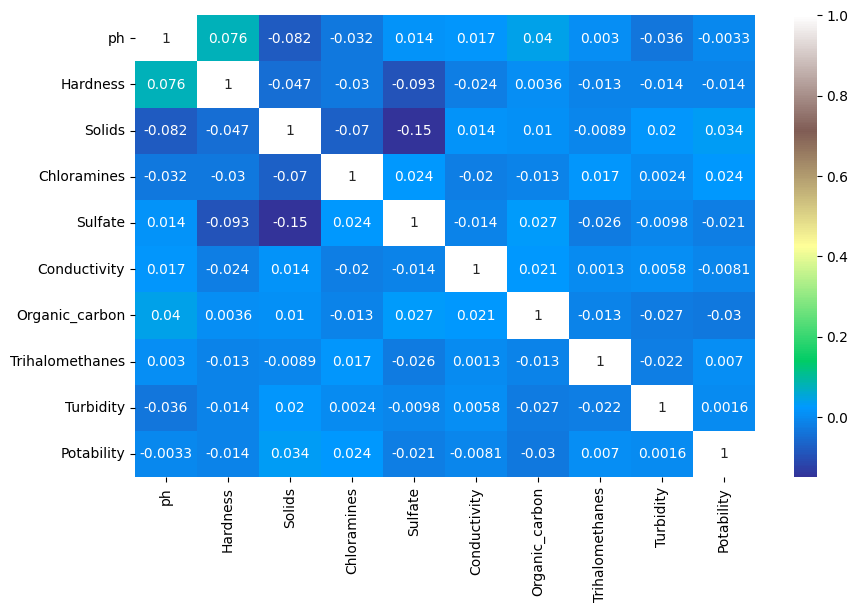
*we are trying the reduce the dimension to which are correlating for that we are looking for the similarity of the features with this chart* *because less feature is making easy to the predict but we have so small similarity of the features and we cant use the remove feature*

*sns.heatmap(data.corr(),annot= True,cmap='terrain')*

*fig= plt.gcf()*

*fig.set\_size\_inches(10,6)*

*plt.show()*

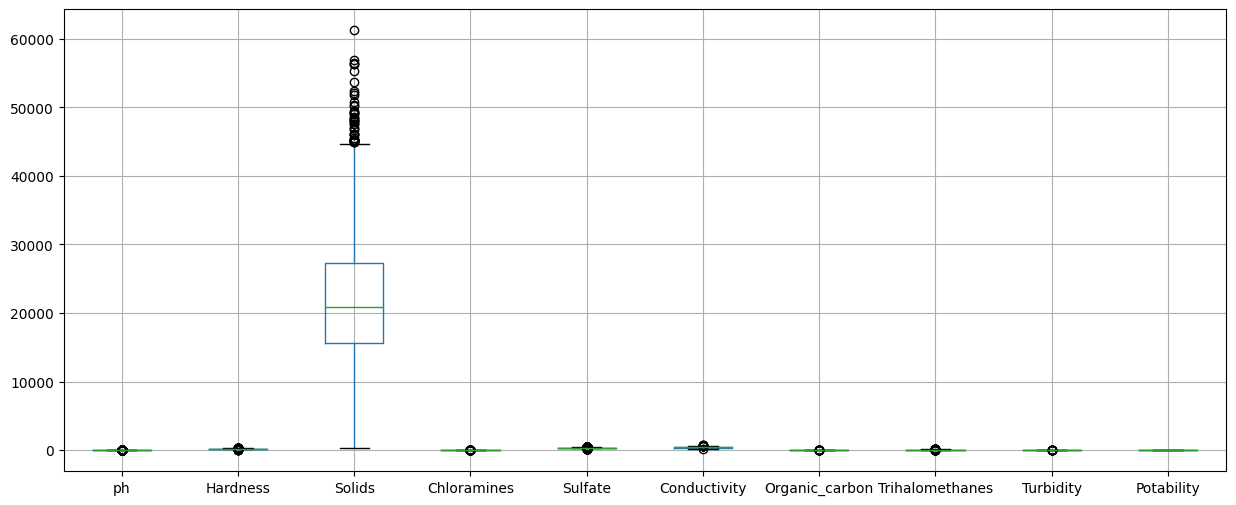
**

# Next Step Is To Check The Outlier Using Box Plot:

*You can see outliers but if we remove this outliers we cant have good predict the result will be closer to good water*

*data.boxplot(figsize=(15,6))*

*plt.show()*

*o/p*

data['Solids'].describe()

o/p:

count 3276.000000

mean 22014.092526

std 8768.570828

min 320.942611

25% 15666.690297

50% 20927.833607

75% 27332.762127

max 61227.196008

# **visualization (Histogram , scatter plot)**

data['Potability'].value\_counts()

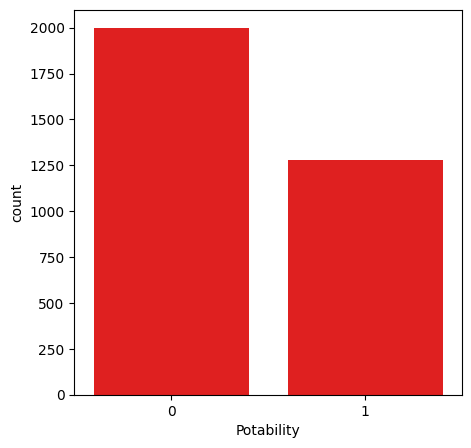
o/p Potability

0 1998

1 1278

plt.figure(figsize=(5,5))

sns.countplot( x=data["Potability"], color="red")

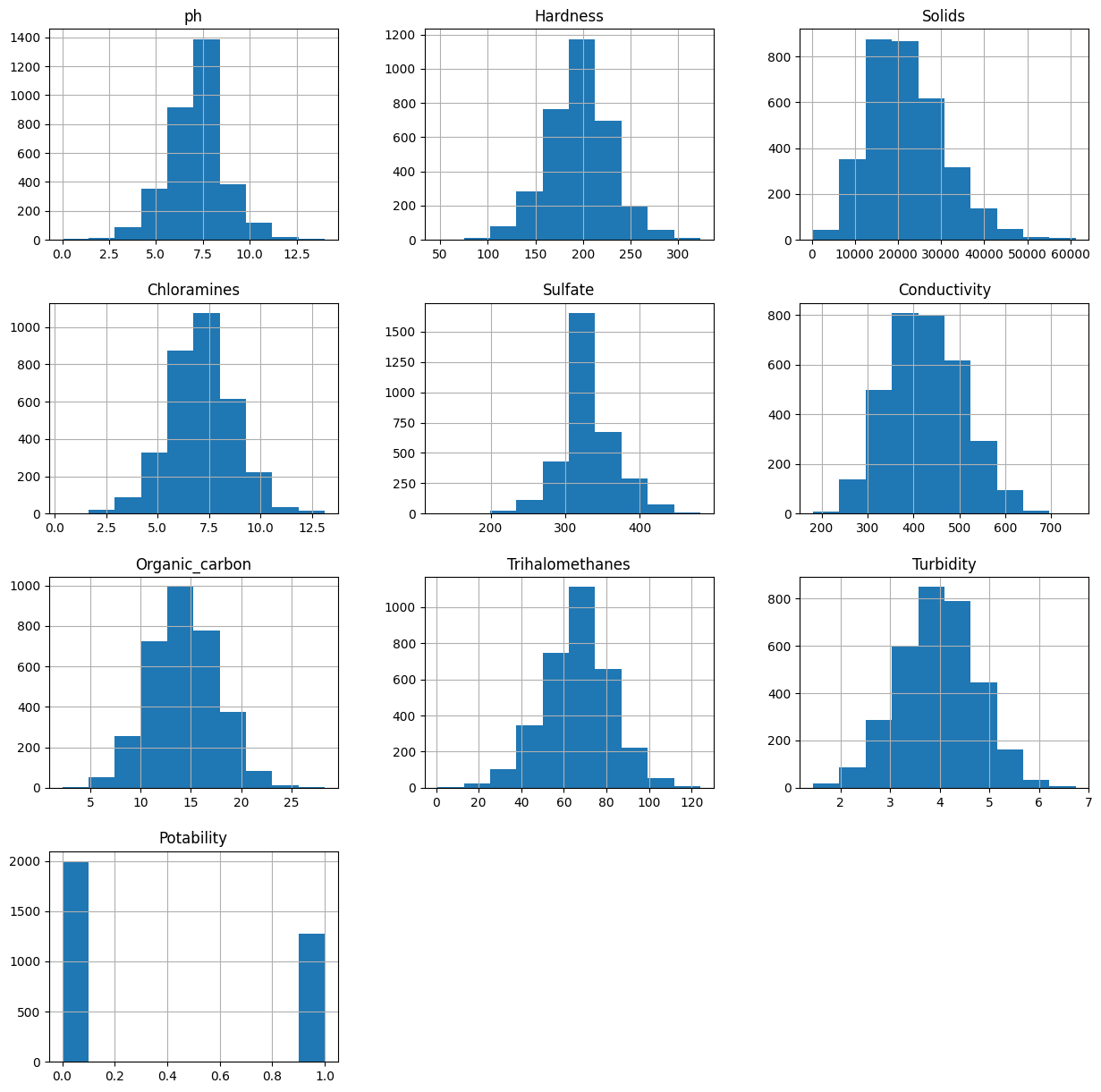
plt.show()

o/p

# Histogram

data.hist(figsize=(15,15))

plt.show()



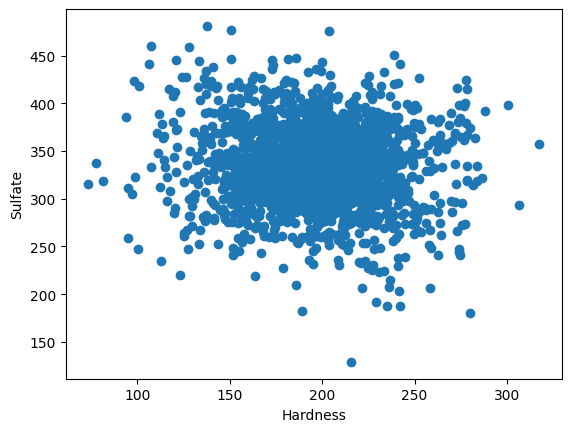
# Scatter Plot

gp = plt.scatter(ks['Hardness'],ks['Sulfate'])

plt.xlabel('Hardness')

plt.ylabel('Sulfate')

plt.show(gp)



sns.scatterplot(x=data['ph'], y=data['Potability'])

plt.show()

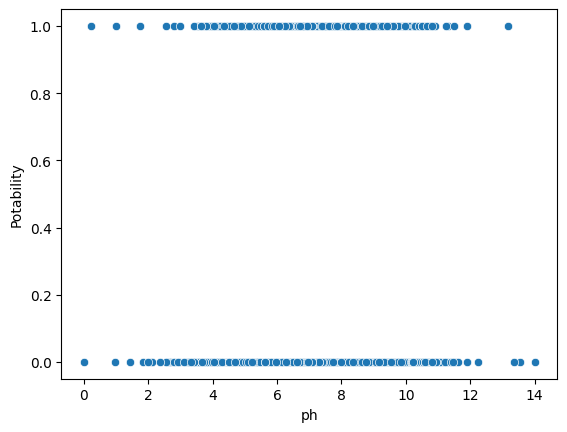


fig = px.scatter(data,x ="ph",y="Hardness",color= "Potability",template="plotly\_dark")

fig.show()

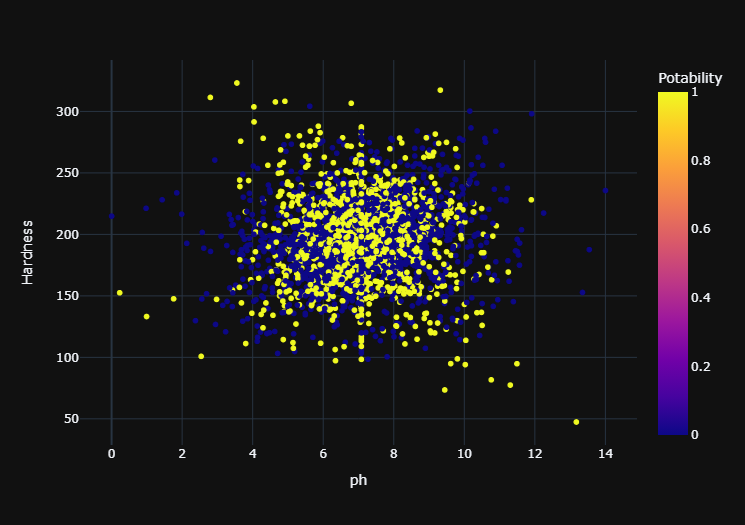
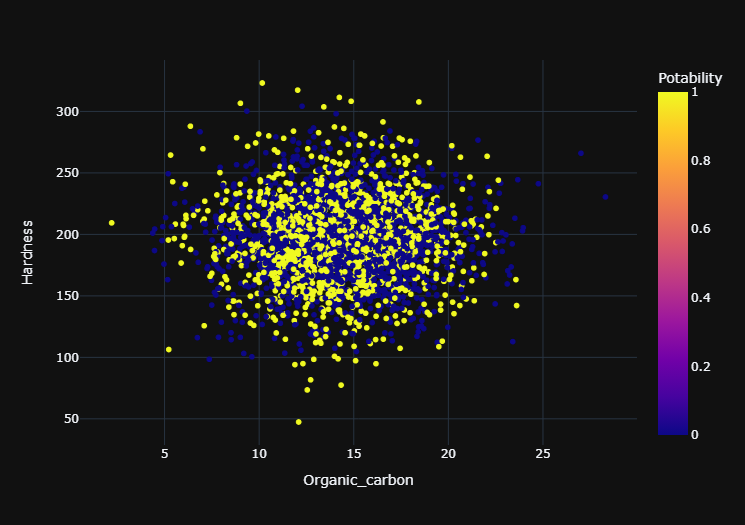


fig = px.scatter(data,x ="Organic\_carbon",y="Hardness",color= "Potability",template="plotly\_dark")

fig.show()



# **Predictive Model Training:**

Predictive modeling is the process of using known results to create a statistical model that can be used for predictive analysis

## Y= data['Potability'] # target variable is potability

from sklearn.model\_selection import train\_test\_split

X\_train , X\_test , Y\_train , Y\_test = train\_test\_split(X,Y,test\_size=0.2, shuffle=True,random\_state=101)

### we splited the data to make a prediction on that train data

X\_train

## X\_test

## Y\_train

o/p

748 1

2279 0

1960 1

1491 1

2991 0

..

599 0

1599 1

1361 0

1547 1

863 0

Y\_test

o/p

2541 0

2605 0

330 1

515 0

400 1

..

482 0

2970 0

50 0

839 0

374 1

Random Forest:

The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(criterion= 'entropy', min\_samples\_split= 3,)

dt.fit(X\_train,Y\_train)

Y\_test

o/p

2541 0

2605 0

330 1

515 0

400 1

..

482 0

2970 0

50 0

839 0

374 1

Y\_prediction=dt.predict(X\_test)

from sklearn.metrics import accuracy\_score, confusion\_matrix

accuracy\_score(Y\_prediction,Y\_test)\*100

o/p:59.756097560975604

confusion\_matrix(Y\_prediction,Y\_test)

o/p array([[262, 124],

[140, 130]])

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import RepeatedStratifiedKFold

dt= DecisionTreeClassifier()

criterion = ["gini","entropy"]

splitter = ['best','random ']

min\_samples\_split=range (1,10)

parameters = dict(criterion=criterion,splitter= splitter, min\_samples\_split= min\_samples\_split)

cv= RepeatedStratifiedKFold(n\_splits = 5,random\_state=101)

grid\_search\_cv\_dt= GridSearchCV(estimator=dt, param\_grid=parameters,scoring='accuracy',cv=cv)

grid\_search\_cv\_dt.fit(X\_train,Y\_train)

o/p:

One or more of the test scores are non-finite:

[ nan nan 0.57664122 nan 0.5798855 nan

0.58038168 nan 0.57916031 nan 0.58041985 nan

0.58118321 nan 0.58053435 nan 0.58049618 nan

nan nan 0.58519084 nan 0.58431298 nan

0.58473282 nan 0.58461832 nan 0.58278626 nan

0.58534351 nan 0.58305344 nan 0.58793893 nan]

print(grid\_search\_cv\_dt.best\_params\_)

prediction\_grid=grid\_search\_cv\_dt.predict(X\_test)

accuracy\_score(Y\_test,prediction\_grid)\*100

o/p 59.756097560975604

# Conclusion:

this project has been a comprehensive exploration of water quality analysis, combining the power of data visualization and predictive modeling to enhance our understanding of water potability. By leveraging visualization libraries such as Matplotlib and Seaborn, we have created informative histograms, scatter plots, and correlation matrices, which have illuminated the relationships between various water quality parameters. These visualizations have provided valuable insights into the data, helping us identify patterns and trends that might not be immediately apparent from raw numbers alone.

Moreover, the development of a predictive model, utilizing techniques like Logistic Regression and Random Forest, has taken our analysis a step further. We have not only assessed the current water quality but also sought to predict water potability based on the collected data. This predictive model has the potential to be a valuable tool for assessing the safety of water sources in real-time and making informed decisions about water treatment and distribution.

## 

## 