WATER QUALITY ANALYSIS

- The project involves analyzing water quality data to assess the suitability of water for specific purposes, such as drinking. The objective is to identify potential issues or deviations from regulatory standards and determine water portability based on various parameters. This project includes defining analysis objectives, collecting water quality data, designing relevant visualizations, and building a predictive model

ABSTRACT:

- Water quality analysis is to measure the required parameters of water, following standard methods, to check whether they are in accordance with the standard.
- Colorimeters and Photometers are used to analyze samples of water, suspended sediment, and bottom material for their content of inorganic and organic constituents.
- Commonly used methods include chelating ion-exchange and, for trace organic analysis, solvent extraction, carbon adsorption, and resin adsorption using nonionic macroeticular resins. Minor variations in microbiological analyses can cause significant changes in results.

OBJECTIVES:

- Water quality analysts monitor and analyze the chemical, physical, and biological components of water to ensure that it meets safety standards for human consumption
- Water has a neutral pH of 7, which indicates that it is neither acidic or basic. The scale ranges from 0 (very acidic) to 14 (very basic). It is normal for water to have a range of between 6.5 and 8.5 on the scale. PH in water may fluctuate with differing environmental factors.
- 0 to 60 mg/L (milligrams per liter) as calcium carbonate is classified as soft; 61 to 120 mg/L as moderately hard; 121 to 180 mg/L as hard; and more than 180 mg/L as very hard.
- The objective of the water quality analysis project is to assess and improve the quality of water in a specific area, such as a river, lake, or reservoir. This project aims to provide accurate and timely data on water quality parameters, identify potential sources of contamination, and develop strategies to maintain or enhance water quality.
- improving water quality for safe drinking water, protecting aquatic ecosystems, or supporting recreational activities.
- Develop a prototype of the water quality monitoring system, which may include sensors, data collection methods, and a data management platform.

ANALYSIS APPROACH:

1. Sample Collection: Collect water samples from various sources, ensuring they are representative of the area of interest. Properly label and store the samples to prevent contamination.

2. Physical Parameters: Measure physical characteristics like temperature, turbidity, color, and odor. These can provide initial insights into water quality.

3. Chemical Analysis: Conduct chemical tests to determine the levels of key constituents such as pH, dissolved oxygen (DO), nutrients (nitrogen and phosphorus), heavy metals, and organic contaminants. This helps assess pollution levels and potential health risks.

4. Biological Assessment: Examine the presence and diversity of aquatic organisms, such as macro invertebrates and algae. These can indicate the ecological health of the water.

5. Microbiological Testing: Test for the presence of bacteria, viruses, and other microorganisms. High levels of certain pathogens can pose health risks.

6. Suspended Solids and Sediment Analysis: Determine the amount of suspended solids and sediment in the water, which can affect water clarity and aquatic habitats.

7. Taste and Odor Analysis: Evaluate the taste and odor of the water, which can be indicative of contamination or the presence of certain chemicals.

8. Statistical Analysis: Analyze the data using statistical methods to identify trends, correlations, and anomalies. This helps in understanding the overall water quality.

9. Regulatory Compliance: Compare the results with local, state, and national water quality standards and regulations to assess compliance.

10. Geographic Information Systems (GIS): Use GIS to map and visualize water quality data spatially, aiding in identifying pollution sources and patterns.

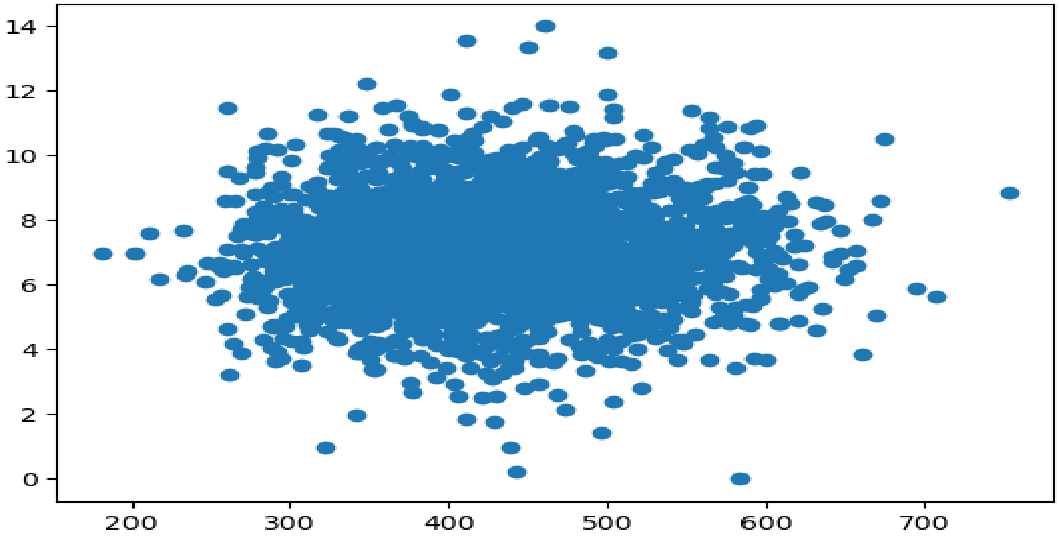
11. Long-Term Monitoring: Establish regular monitoring programs to track changes in water quality over time, which can be critical for detecting trends and addressing issues.

12. Interpretation and Reporting: Interpret the findings and prepare reports that communicate the water quality status, potential risks, and recommended actions to relevant stakeholders.

13. Mitigation and Remediation: Implement measures to address water quality issues, which may include treatment, pollution control, and habitat restoration.

14. Public Awareness: Educate the public about water quality concerns and promote responsible water use and conservation.

VISUALIZATION SELECTION:



STEPS:

1.Data Collection**:**

- Collect data on the water quality parameters of interest. This data should include both normal and unusual values for the parameters.
- It includes pH value, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes.

2.Data Preparation**:**

- Prepare the data for analysis by cleaning and organizing it. This may involve removing outliers, filling in missing values, and transforming the data if necessary.

3.Model Selection:

- Choose an appropriate linear regression model to fit the data. This may involve selecting a simple linear regression model or a more complex model that includes multiple variables.

4.Model fitting**:**

- Fit the selected model to the data using a statistical software package such as R or Python.

5.Model Evaluation:

- Evaluate the performance of the model by examining its goodness of fit and statistical significance.

6.Prediction:

- Once the model has been fitted and evaluated, it can be used to predict unusual values in water quality parameters.

7.Deployment:

- If the model is satisfied and meets your requirements you can deploy it in a production environment in real time.

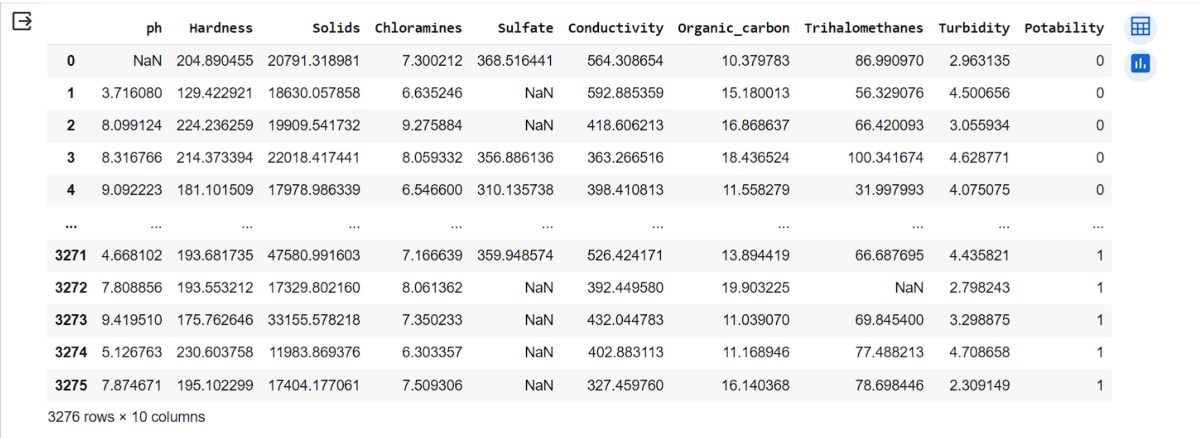
CHECKING THE MISSING VALUES

import pandas as pd

#This library used to read and write a data frame ,

df = pd.read_csv('water_potability.csv')
#upload the dataset using the pandas library

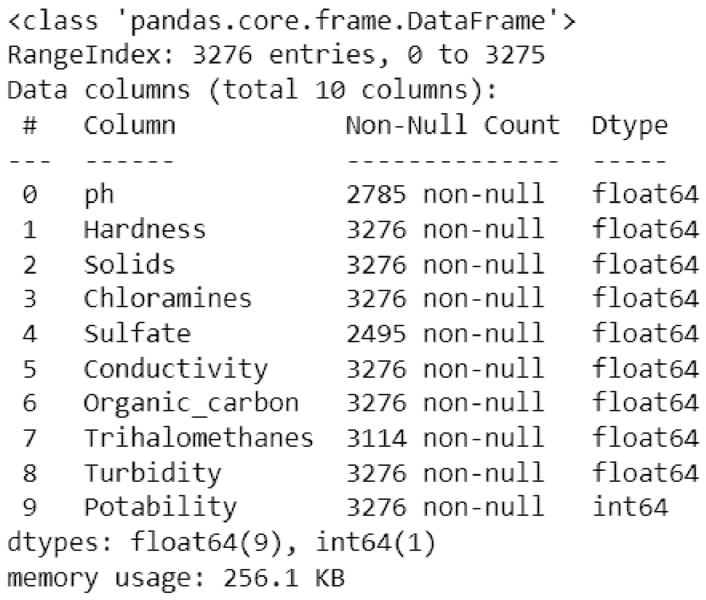
df
#printing the data set
OUTPUT:



```
            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
 ...       ...         ...           ...          ...         ...           ...             ...              ...        ...         ...
3271  4.668102  193.681735  47580.991603     7.166639  359.948574    526.424171       13.894419        66.687695   4.435821           1
3272  7.808856  193.553212  17329.802160     8.061362         NaN    392.449580       19.903225              NaN   2.798243           1
3273  9.419510  175.762646  33155.578218     7.350233         NaN    432.044783       11.039070        69.845400   3.298875           1
3274  5.126763  230.603758  11983.869376     6.303357         NaN    402.883113       11.168946        77.488213   4.708658           1
3275  7.874671  195.102299  17404.177061     7.509306         NaN    327.459760       16.140368        78.698446   2.309149           1
3276 rows × 10 columns
```

df.info()

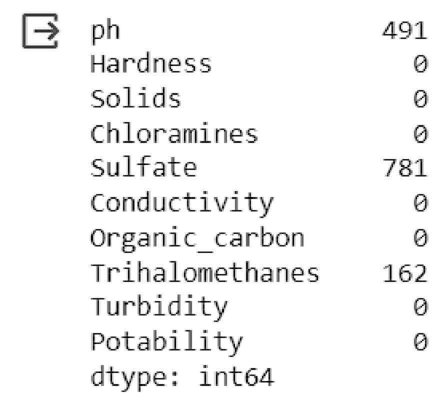
#checking the datatype of the dataset
OUTPUT:



```
<class 'pandas.core.frame.DataFrame'>
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  
dtypes: float64(9), int64(1)
memory usage: 256.1 KB
```

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

#checking the missing values in the dataset
OUTPUT:



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

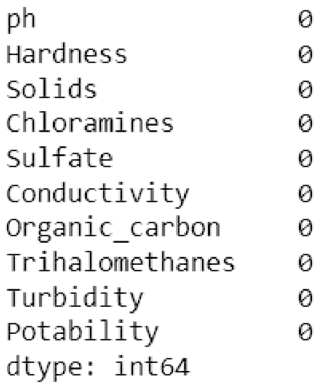
```
df.fillna(9.5,inplace=True)
```

#Replace the missing values using fillna() command with the common value 9.5

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

#again checking for the missing values

OUTPUT:

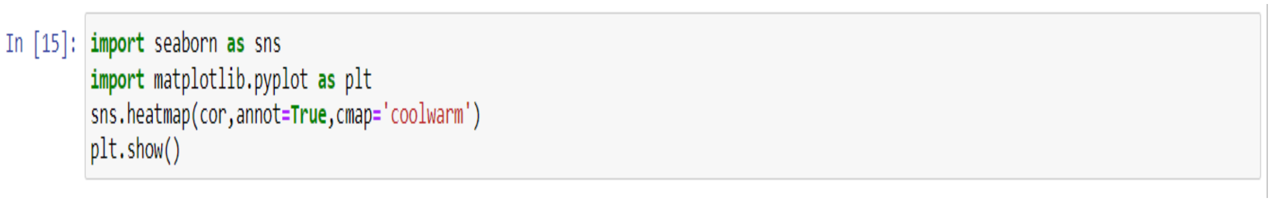


```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

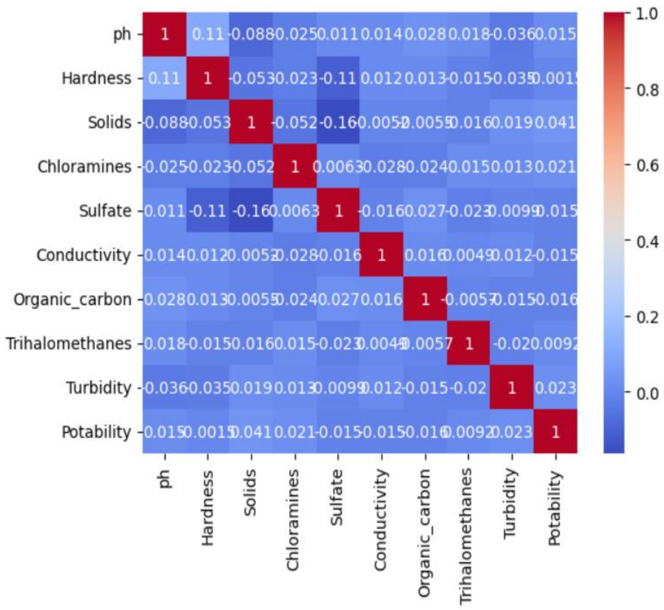
To replicate a water quality analysis project in Python, gather data from reliable sources and clean it. Perform EDA to understand the dataset and create visualizations. Visualize the data and extract relevant features. Split the data into training and testing sets. Build a predictive model and evaluate its performance. Visualize the model's results and deploy it for production. Monitor the model's performance and update it with new data as needed.

Exploratory Analysis:

#Importing the libraries

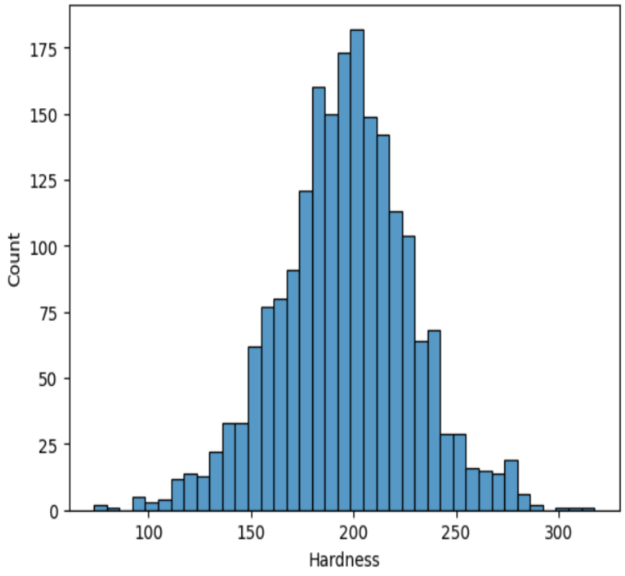


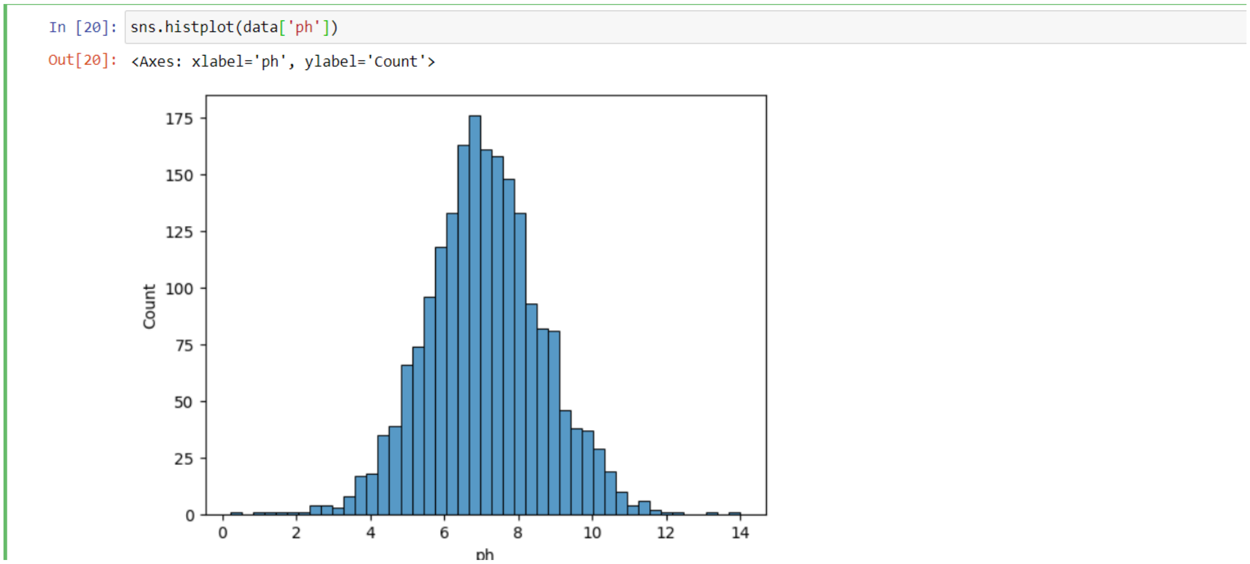
```
In [15]: import seaborn as sns
         import matplotlib.pyplot as plt
         sns.heatmap(cor,annot=True,cmap='coolwarm')
         plt.show()
```



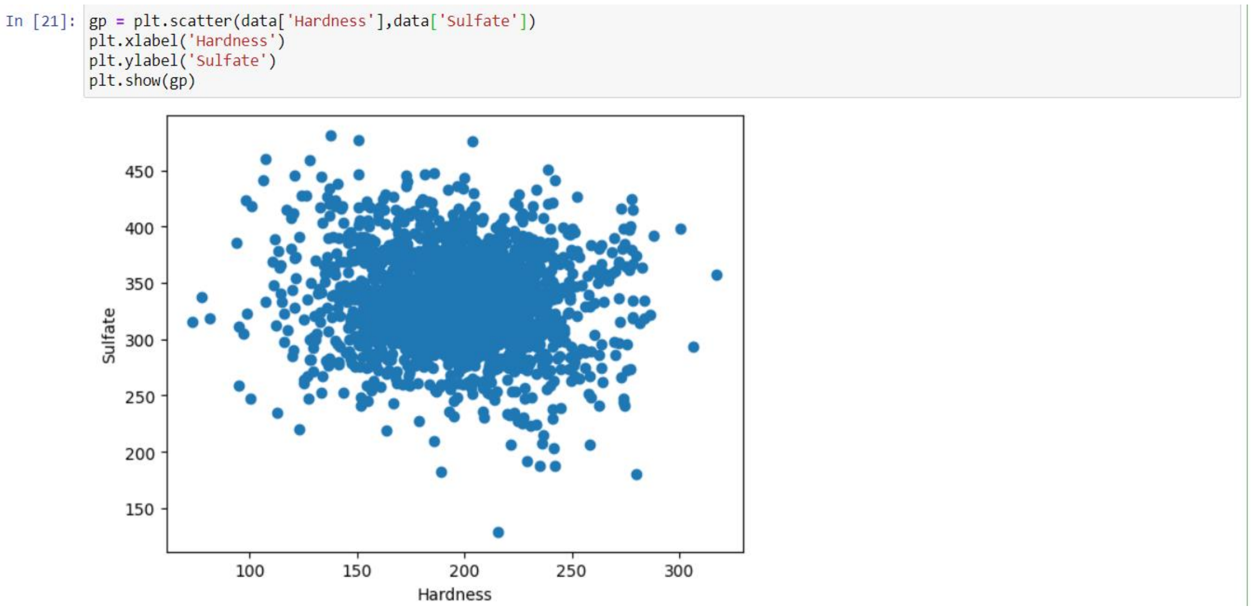
```
In [19]: sns.histplot(data['Hardness'])
```

```
Out[19]: <Axes: xlabel='Hardness', ylabel='Count'>
```



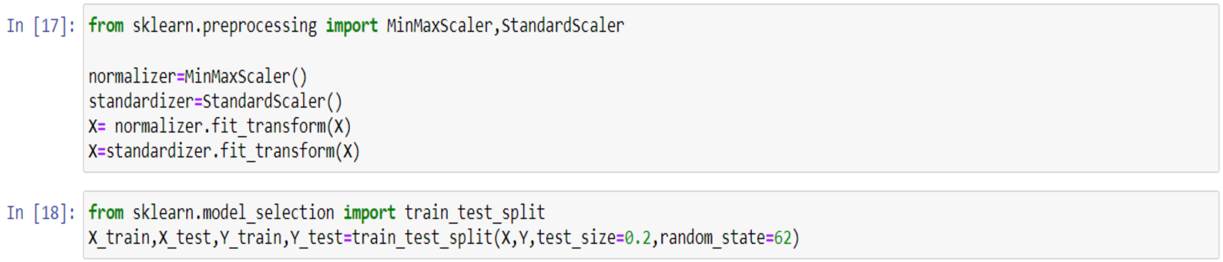


```
In [20]: sns.histplot(data['ph'])
Out[20]: <Axes: xlabel='ph', ylabel='Count'>
```



```
In [21]: gp = plt.scatter(data['Hardness'],data['Sulfate'])
         plt.xlabel('Hardness')
         plt.ylabel('Sulfate')
         plt.show(gp)
```

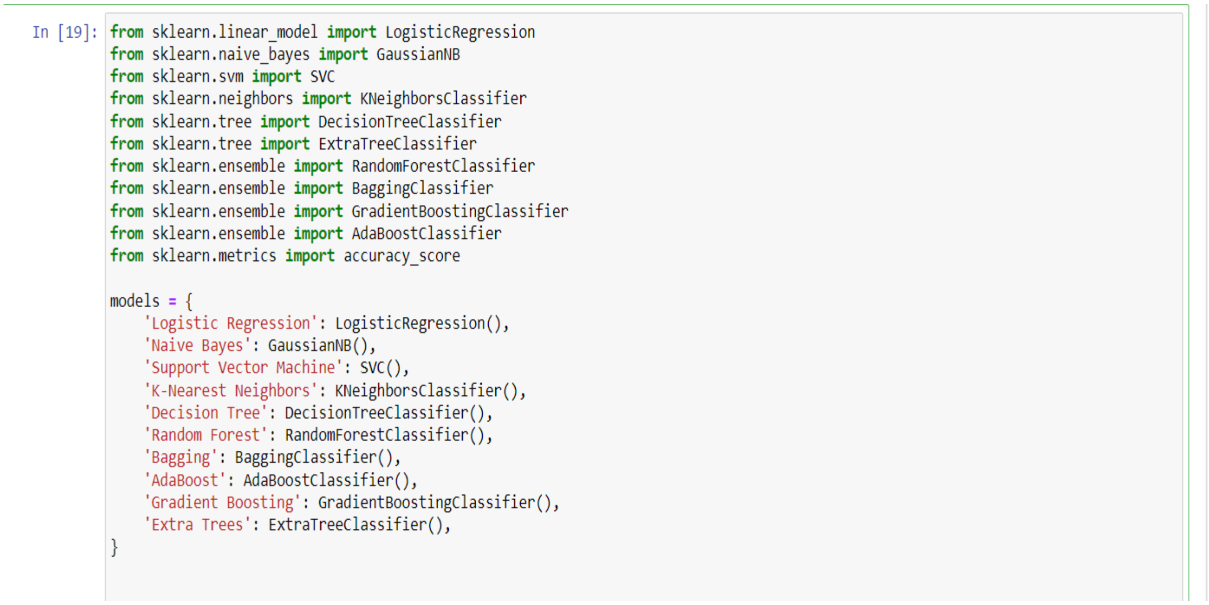
DATA NORMALIZATION AND STANDARDIZATION



```
In [17]: from sklearn.preprocessing import MinMaxScaler,StandardScaler

         normalizer=MinMaxScaler()
         standardizer=StandardScaler()
         X= normalizer.fit_transform(X)
         X=standardizer.fit_transform(X)
```

```
In [18]: 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,random_state=62)
```

MODEL BUILDING



```
In [19]: from sklearn.linear_model import LogisticRegression
         from sklearn.naive_bayes import GaussianNB
         from sklearn.svm import SVC
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.tree import ExtraTreeClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.ensemble import BaggingClassifier
         from sklearn.ensemble import GradientBoostingClassifier
         from sklearn.ensemble import AdaBoostClassifier
         from sklearn.metrics import accuracy_score

         models = {
             'Logistic Regression': LogisticRegression(),
             'Naive Bayes': GaussianNB(),
             'Support Vector Machine': SVC(),
             'K-Nearest Neighbors': KNeighborsClassifier(),
             'Decision Tree': DecisionTreeClassifier(),
             'Random Forest': RandomForestClassifier(),
             'Bagging': BaggingClassifier(),
             'AdaBoost': AdaBoostClassifier(),
             'Gradient Boosting': GradientBoostingClassifier(),
             'Extra Trees': ExtraTreeClassifier(),
         }
```

Virtualization:

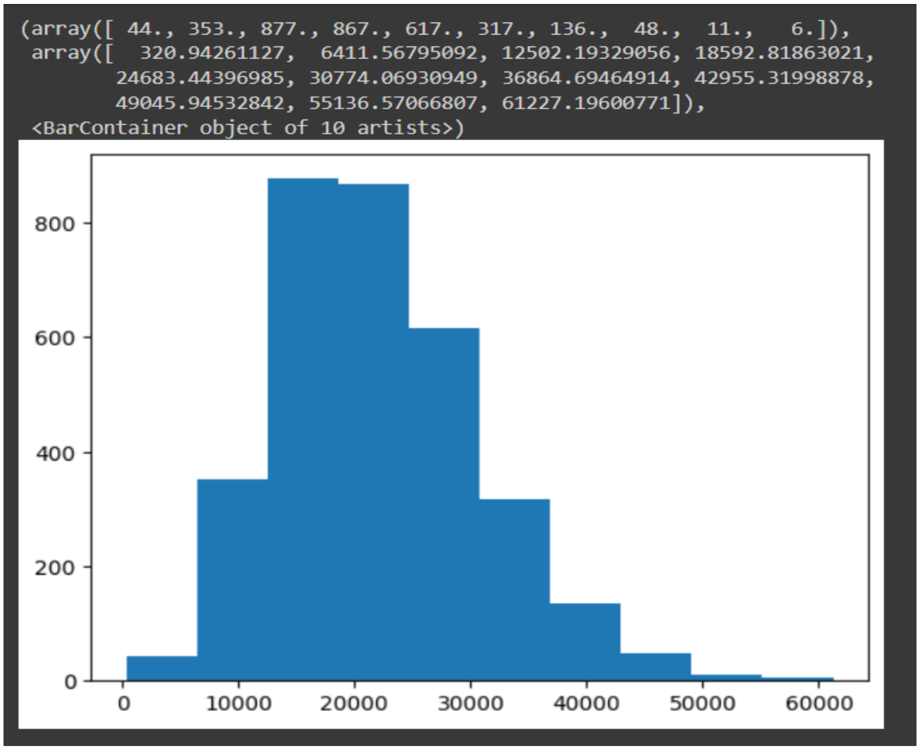
Import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

plt.hist(df["Solids"])

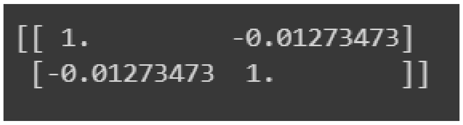
Output:



```
(array([ 44., 353., 877., 867., 617., 317., 136.,  48.,  11.,   6.]),
 array([  320.94261127,  6411.56795092, 12502.19329056, 18592.81863021,
        24683.44396985, 30774.06930949, 36864.69464914, 42955.31998878,
        49045.94532842, 55136.57066807, 61227.19600771]),
 <BarContainer object of 10 artists>)
```

```
x=df["Sulfate"]

y=df["Trihalomethanes"]

matrix = np.corrcoef(x, y)

print(matrix)
```

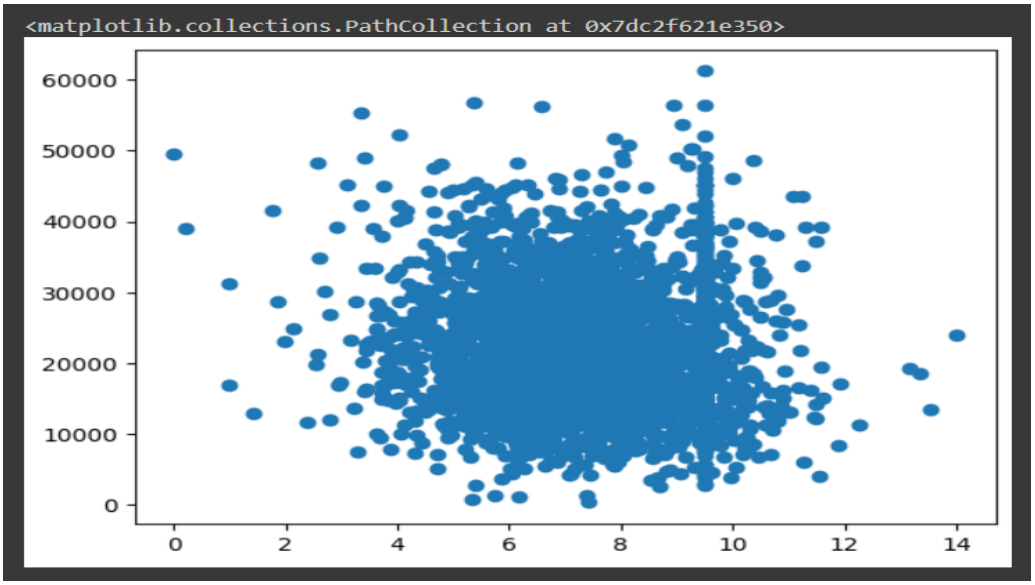
Output:



```
[[ 1.         -0.01273473]
 [-0.01273473  1.        ]]
```

```
plt.scatter(df["ph"],df["Solids"])
```

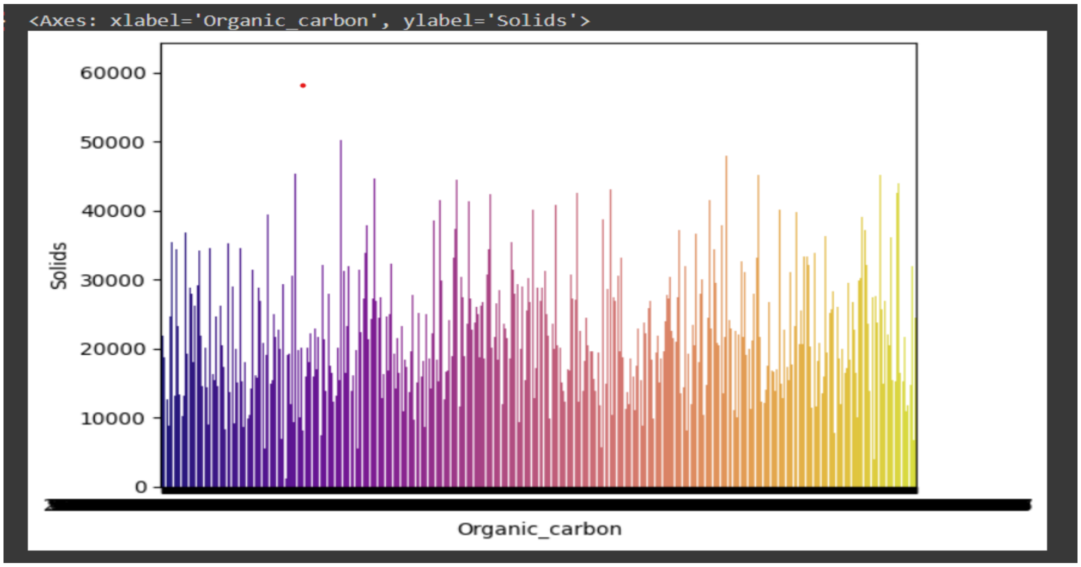
Output:



```
<matplotlib.collections.PathCollection at 0x7dc2f621e350>
```

sns.barplot(x=df["Organic_carbon"],y=df["Solids"],data=df,palette="plasma")

Output:



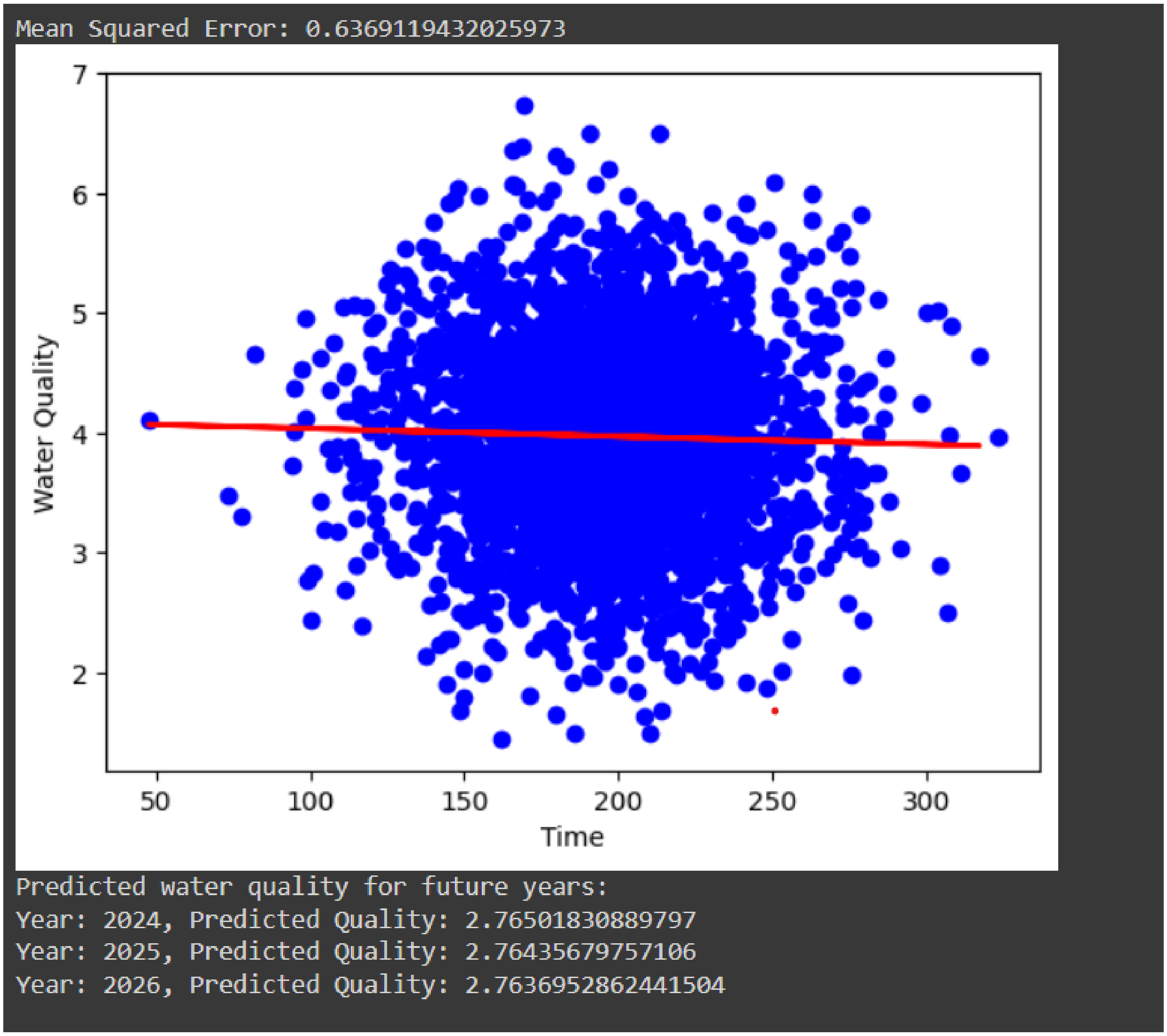
```
<Axes: xlabel='Organic_carbon', ylabel='Solids'>
```

Predict :

From sklearn.linear_mode limport LinearRegression

From sklearn.metrics import mean_squared_error

```
X = data['Hardness'].values.reshape(-1,1)

y = data['Turbidity'].values

X_train,X_test,y_train,y_test = train_test_split(X, y,test_size=0.2,random_state=0)

model = LinearRegression()

model.fit(X_train,y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test,y_pred)

print(f"Mean Squared Error: {mse}")

plt.scatter(X, y, color='blue')

plt.plot(X_test,y_pred, color='red', linewidth=2)

plt.xlabel('Time')

plt.ylabel('Water Quality')

plt.show()

future_time = np.array([2024,2025,2026]).reshape(-1,1)

future_quality = model.predict(future_time)

print("Predicted water quality for future years:")

for year, quality inzip([2024,2025,2026],future_quality):

 print(f"Year: {year}, Predicted Quality: {quality}")
```



```
Mean Squared Error: 0.6369119432025973
```

```
Predicted water quality for future years:
Year: 2024, Predicted Quality: 2.76501830889797
Year: 2025, Predicted Quality: 2.76435679757106
Year: 2026, Predicted Quality: 2.7636952862441504
```

CONCLUSION:

- Insights from water quality analysis play a critical role in assessing water quality and determining its portability .

Key Parameters:

- Water quality analysis typically involves measuring various parameters such as pH, turbidity, dissolved oxygen, temperature, and concentrations of specific contaminants (e.g., bacteria, heavy metals, nitrates).Safety for Human Consumption: Parameters like pH, turbidity, and dissolved oxygen are directly related to water's safety for human consumption. Low pH levels or high turbidity can indicate water quality issues .Detection of Contaminants: Water quality analysis can identify the presence of harmful contaminants, such as bacteria (e.g., E. coli), heavy metals (e.g., lead, arsenic), or chemicals (e.g., pesticides).Monitoring for Seasonal Variations :Insights from ongoing water quality analysis can reveal seasonal variations and trends. Early Warning for Pollution Events :Real-time monitoring systems can detect sudden pollution events, such as industrial spills or runoff from agricultural areas. Rapid response to such events can prevent contamination and safeguard portability. Continuous monitoring and adjustments are made to ensure that water leaving treatment plants is portable.