PROJECT DOCUMENT – PHASE 4

WATER QUALITY ANALYSIS

Madras Institute of Technology, Anna University

Harishma Sabu Ranjana S Ranjini S Roshni N Vithya S

About Dataset

Context

Access to safe drinking-water is essential to 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. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions.

Content

The water_potability.csv file contains water quality metrics for 3276 different water bodies.

1. pH value:

PH is an important parameter in evaluating the acid–base balance of water. It is also the indicator of acidic or alkaline condition of water status. WHO has recommended 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. Hardness was 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 or salts such as potassium, calcium, sodium, bicarbonates, chlorides, magnesium, sulfates etc. These minerals produced un-wanted taste and diluted color in 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. Desirable limit for TDS is 500 mg/l and maximum limit is 1000 mg/l which 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 liter (mg/L or 4 parts per million (ppm)) are considered safe in drinking water.

5. Sulfate:

Sulfates 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 sulfate is in the chemical industry. Sulfate concentration in seawater is about 2,700 milligrams per liter (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'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 exceeded 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 a measure of the total amount of carbon in organic compounds in pure water. According to US EPA < 2 mg/L as TOC in treated / drinking water, and < 4 mg/Lit in source water which is use 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:

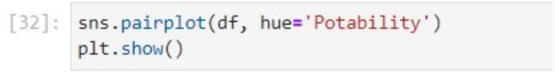
The turbidity of water depends on the quantity of solid matter present in the suspended state. It is a measure of light emitting properties of water and the test is used to indicate the quality of waste discharge with respect to colloidal matter. The mean turbidity value obtained for Wondo Genet Campus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

10. Potability:

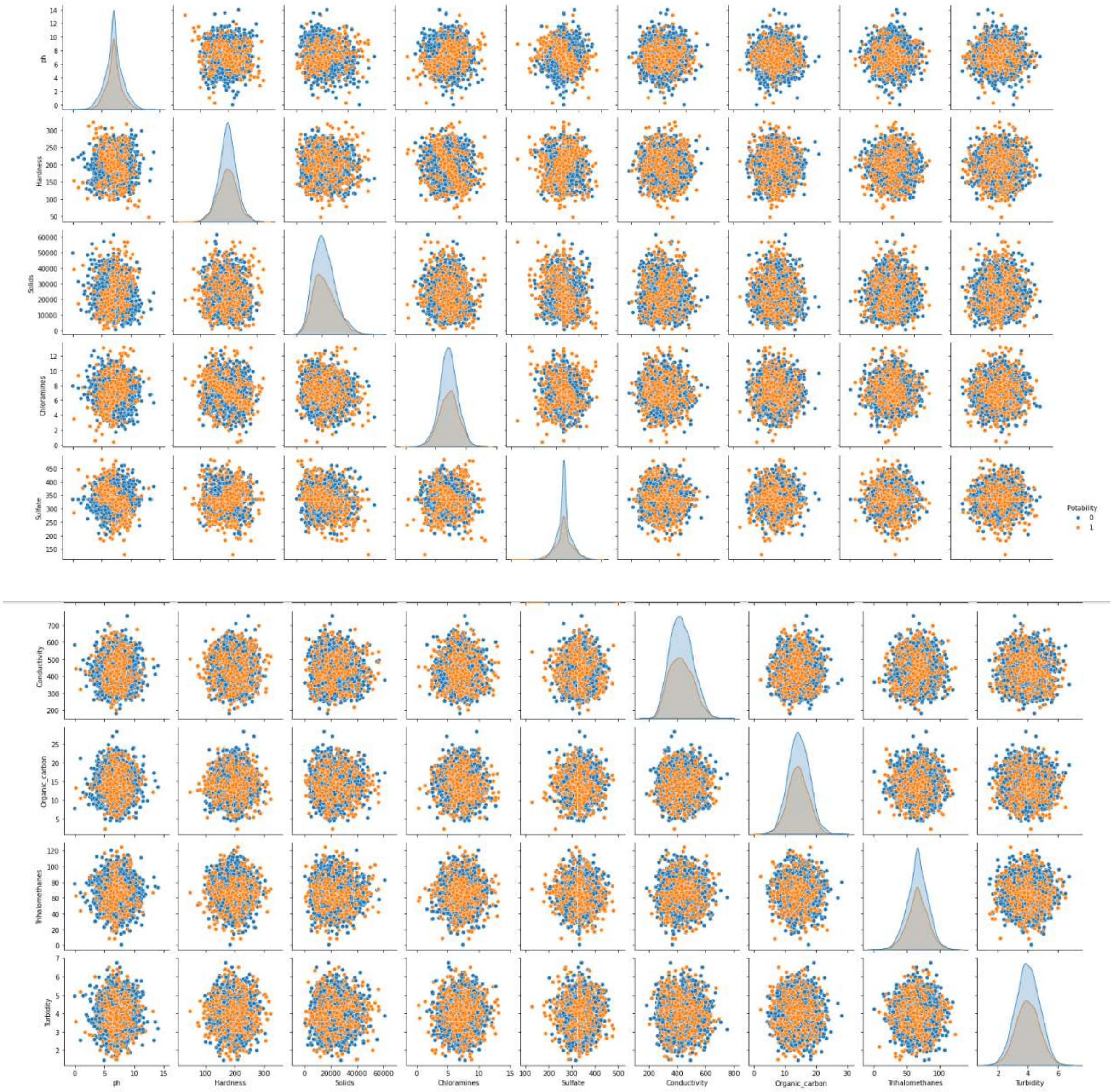
Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable. (0) Water is not safe to drink and (1) Water is safe to drink

DATA EXPLORATION AND VISUALIZATION

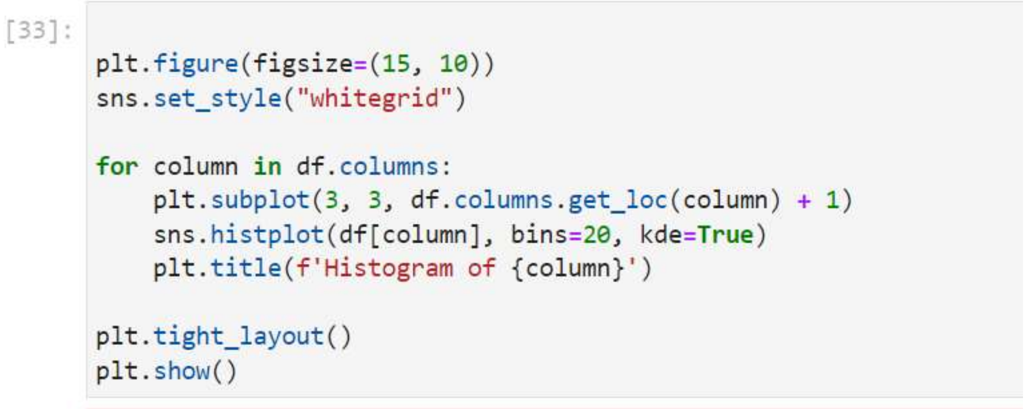
SCATTER PLOTS



```
[32]: sns.pairplot(df, hue='Potability')
      plt.show()
```

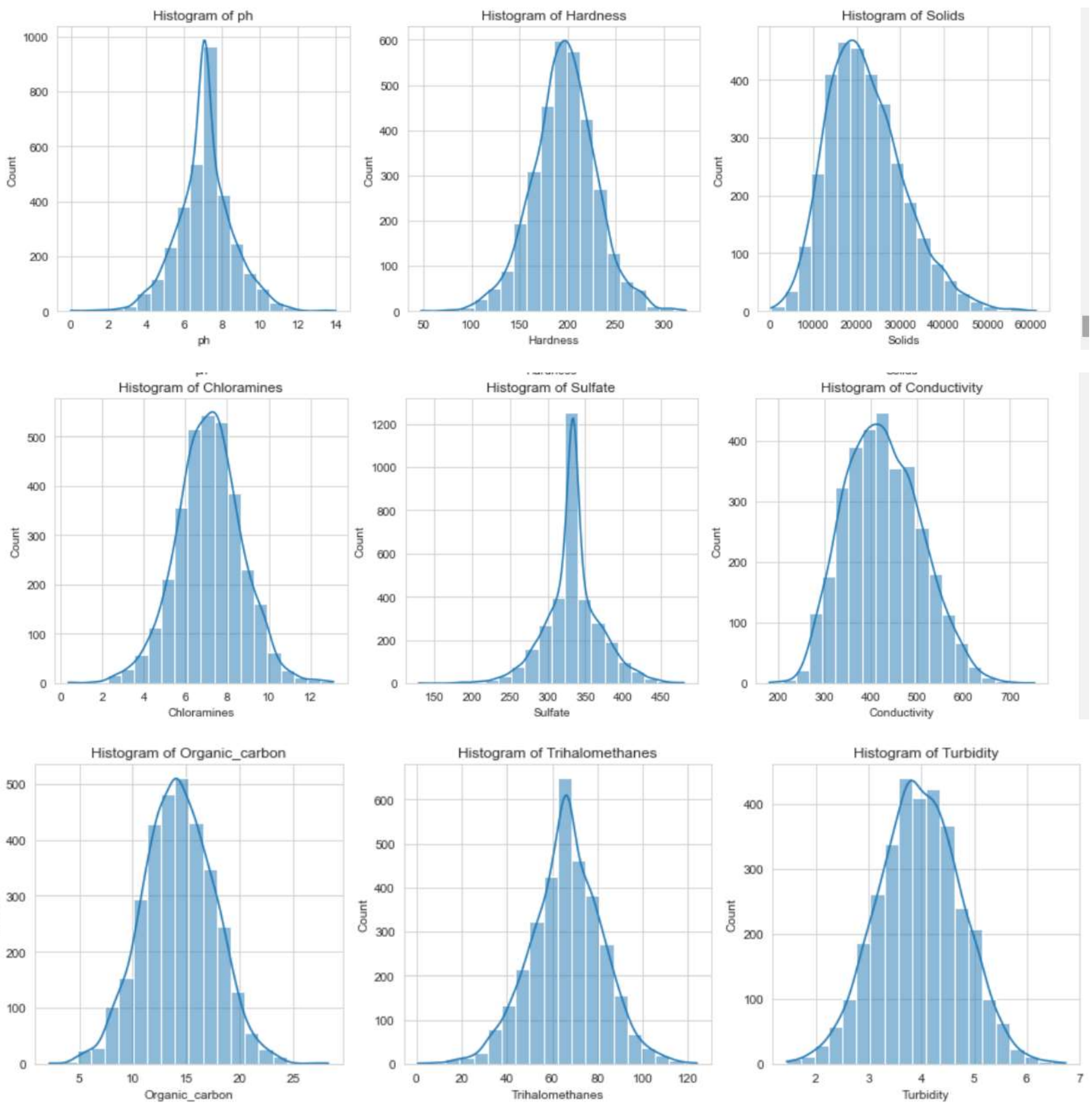


HISTOGRAMS

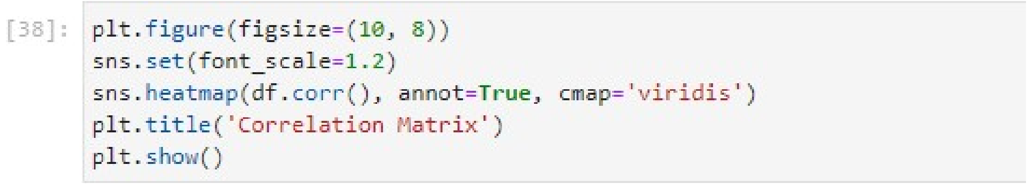


[33]:

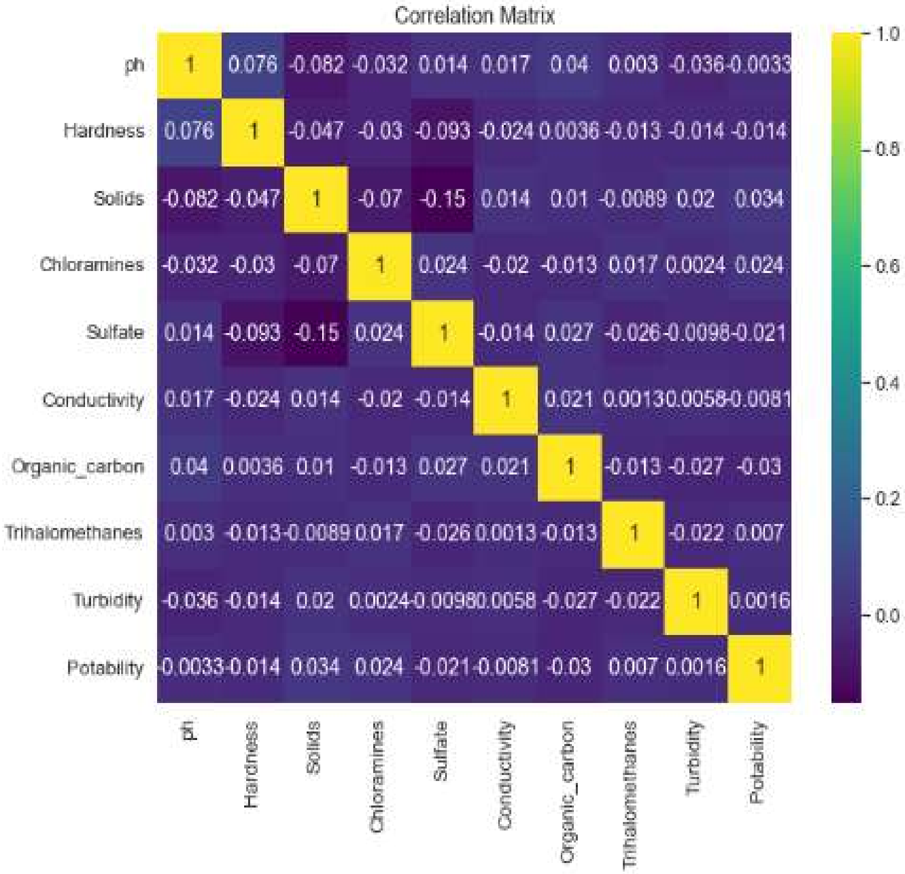
```
plt.figure(figsize=(15, 10))
sns.set_style("whitegrid")

for column in df.columns:
    plt.subplot(3, 3, df.columns.get_loc(column) + 1)
    sns.histplot(df[column], bins=20, kde=True)
    plt.title(f'Histogram of {column}')

plt.tight_layout()
plt.show()
```



CORRELATION MATRIX



```
[38]: plt.figure(figsize=(10, 8))
      sns.set(font_scale=1.2)
      sns.heatmap(df.corr(), annot=True, cmap='viridis')
      plt.title('Correlation Matrix')
      plt.show()
```

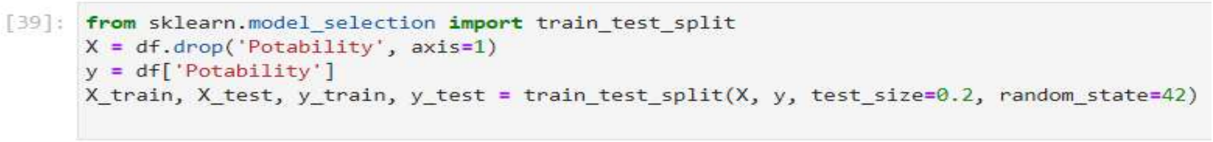


Correlation Matrix

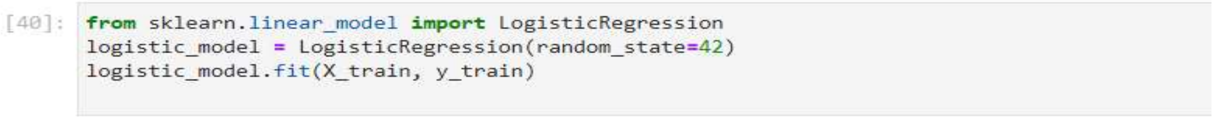
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1	0.076	-0.082	-0.032	0.014	0.017	0.04	0.003	-0.036	-0.0033
Hardness	0.076	1	-0.047	-0.03	-0.093	-0.024	0.0036	-0.013	-0.014	-0.014
Solids	-0.082	-0.047	1	-0.07	-0.15	0.014	0.01	-0.0089	0.02	0.034
Chloramines	-0.032	-0.03	-0.07	1	0.024	-0.02	-0.013	0.017	0.0024	0.024
Sulfate	0.014	-0.093	-0.15	0.024	1	-0.014	0.027	-0.026	-0.0098	-0.021
Conductivity	0.017	-0.024	0.014	-0.02	-0.014	1	0.021	0.0013	0.0058	-0.0081
Organic_carbon	0.04	0.0036	0.01	-0.013	0.027	0.021	1	-0.013	-0.027	-0.03
Trihalomethanes	0.003	-0.013	-0.0089	0.017	-0.026	0.0013	-0.013	1	-0.022	0.007
Turbidity	-0.036	-0.014	0.02	0.0024	-0.0098	0.0058	-0.027	-0.022	1	0.0016
Potability	-0.0033	-0.014	0.034	0.024	-0.021	-0.0081	-0.03	0.007	0.0016	1

PREDICTIVE MODEL

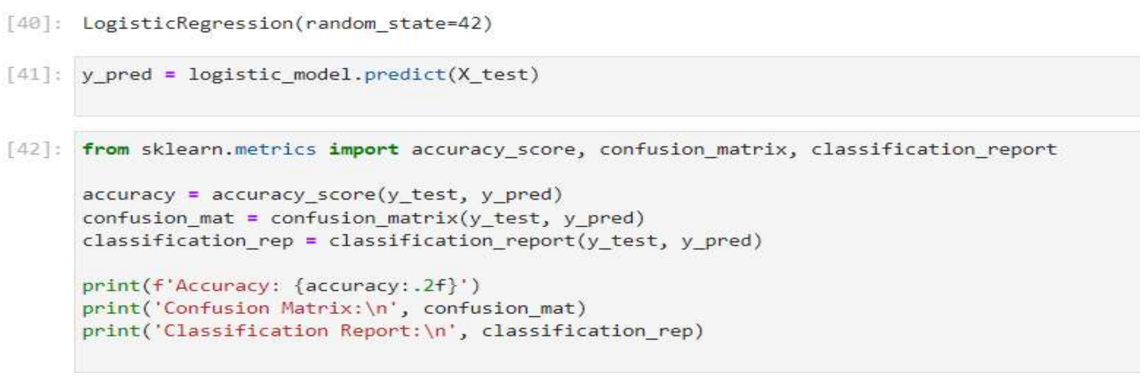
1.Using Logistic Regression



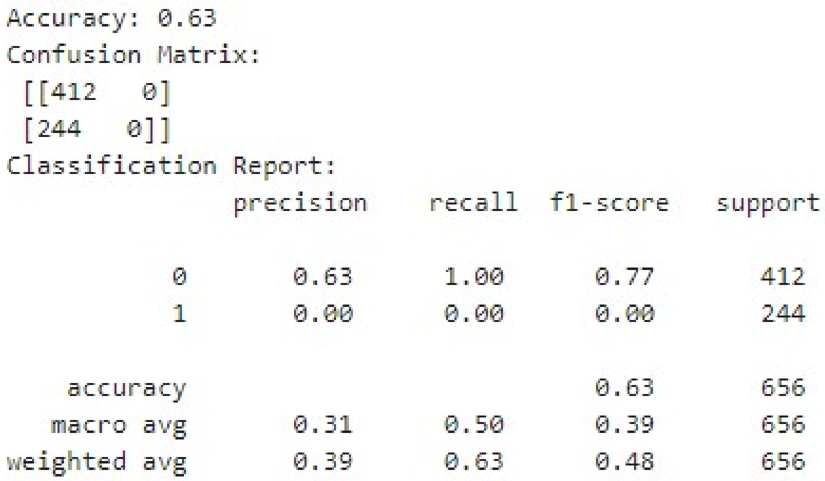
```
[39]: from sklearn.model_selection import train_test_split
      X = df.drop('Potability', axis=1)
      y = df['Potability']
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```
[40]: from sklearn.linear_model import LogisticRegression
      logistic_model = LogisticRegression(random_state=42)
      logistic_model.fit(X_train, y_train)
```

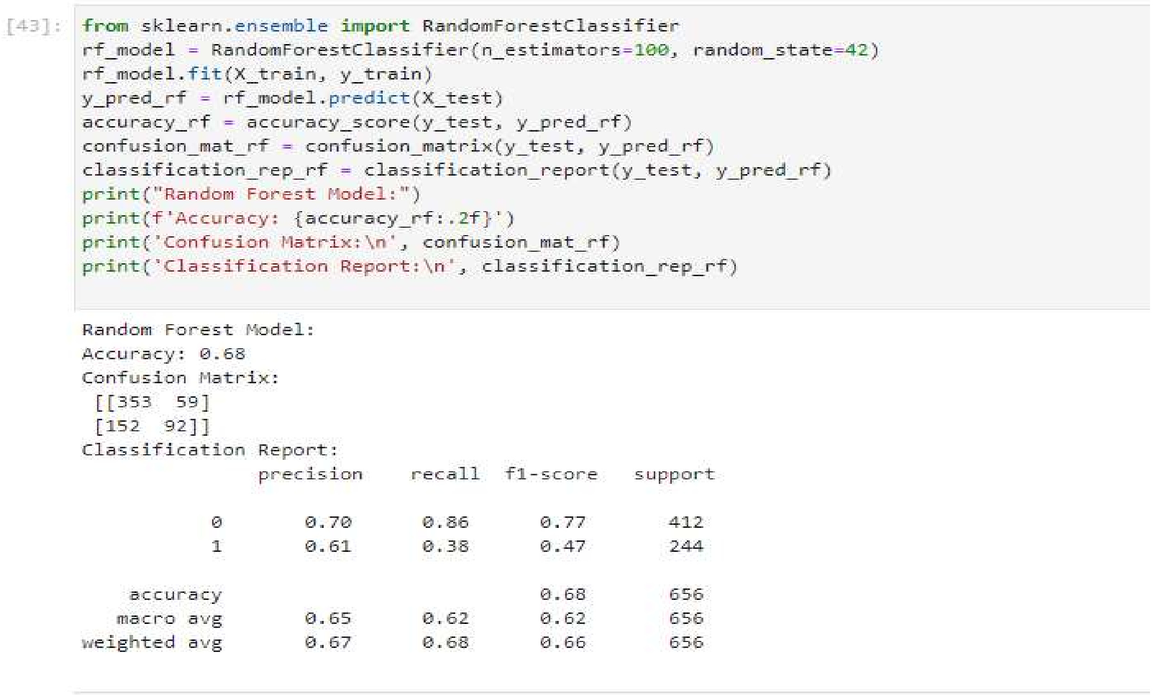


```
[40]:  LogisticRegression(random_state=42)

[41]:  y_pred = logistic_model.predict(X_test)

[42]:  from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

       accuracy = accuracy_score(y_test, y_pred)
       confusion_mat = confusion_matrix(y_test, y_pred)
       classification_rep = classification_report(y_test, y_pred)

       print(f'Accuracy: {accuracy:.2f}')
       print('Confusion Matrix:\n', confusion_mat)
       print('Classification Report:\n', classification_rep)
```



```
Accuracy: 0.63
Confusion Matrix:
 [[412   0]
 [244   0]]
Classification Report:
               precision    recall  f1-score   support

           0       0.63      1.00      0.77       412
           1       0.00      0.00      0.00       244

    accuracy                           0.63       656
   macro avg       0.31      0.50      0.39       656
weighted avg       0.39      0.63      0.48       656
```

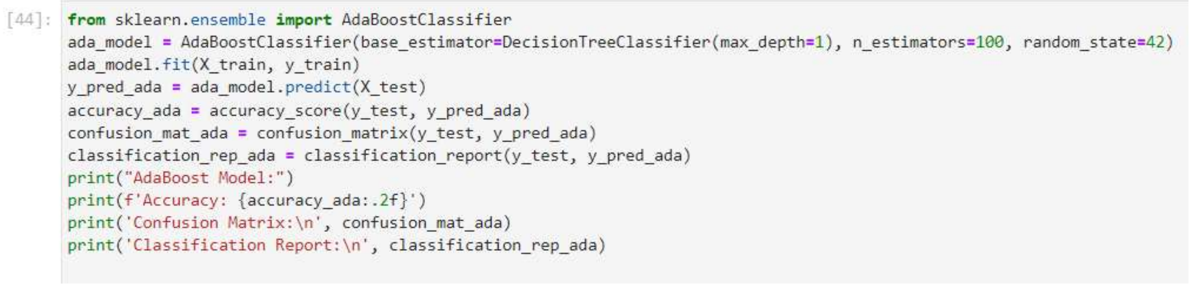
2. Using Random forest



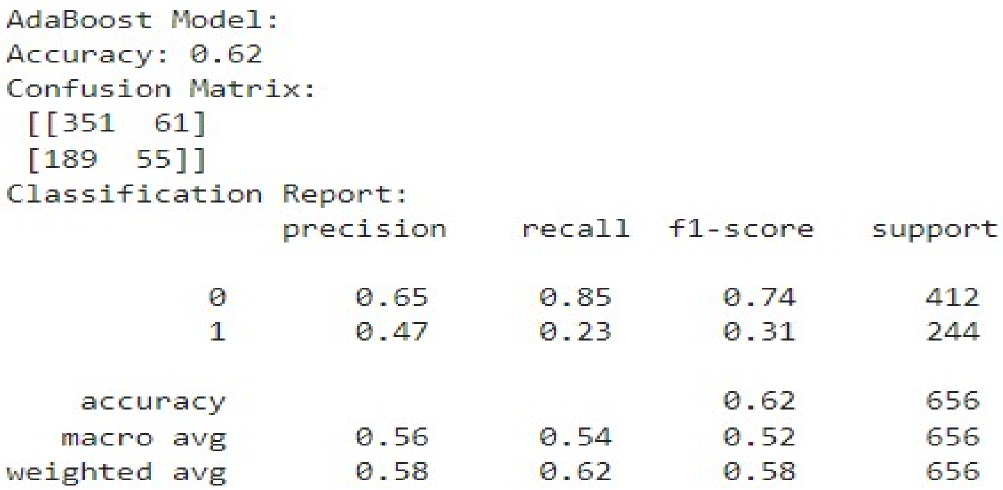
```
[43]:  from sklearn.ensemble import RandomForestClassifier
       rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
       rf_model.fit(X_train, y_train)
       y_pred_rf = rf_model.predict(X_test)
       accuracy_rf = accuracy_score(y_test, y_pred_rf)
       confusion_mat_rf = confusion_matrix(y_test, y_pred_rf)
       classification_rep_rf = classification_report(y_test, y_pred_rf)
       print("Random Forest Model:")
       print(f'Accuracy: {accuracy_rf:.2f}')
       print('Confusion Matrix:\n', confusion_mat_rf)
       print('Classification Report:\n', classification_rep_rf)
```

```
Random Forest Model:
Accuracy: 0.68
Confusion Matrix:
 [[353  59]
 [152  92]]
Classification Report:
               precision    recall  f1-score   support

           0       0.70      0.86      0.77       412
           1       0.61      0.38      0.47       244

    accuracy                           0.68       656
   macro avg       0.65      0.62      0.62       656
weighted avg       0.67      0.68      0.66       656
```

3.UsingAdaBoost Model

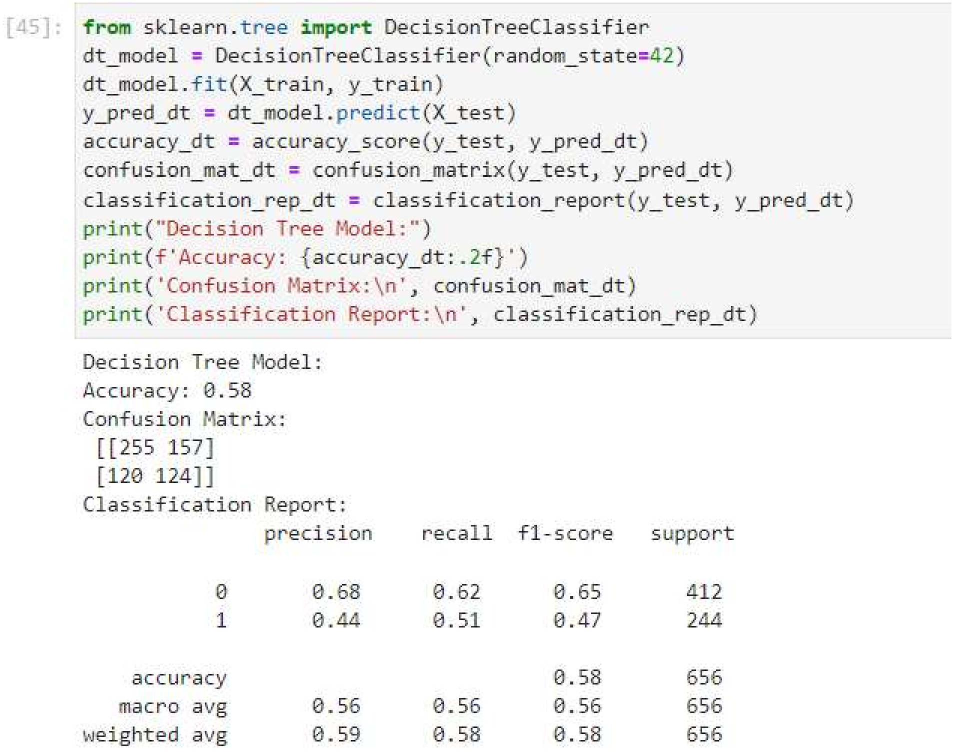


```
[44]: from sklearn.ensemble import AdaBoostClassifier
      ada_model = AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=1), n_estimators=100, random_state=42)
      ada_model.fit(X_train, y_train)
      y_pred_ada = ada_model.predict(X_test)
      accuracy_ada = accuracy_score(y_test, y_pred_ada)
      confusion_mat_ada = confusion_matrix(y_test, y_pred_ada)
      classification_rep_ada = classification_report(y_test, y_pred_ada)
      print("AdaBoost Model:")
      print(f'Accuracy: {accuracy_ada:.2f}')
      print('Confusion Matrix:\n', confusion_mat_ada)
      print('Classification Report:\n', classification_rep_ada)
```



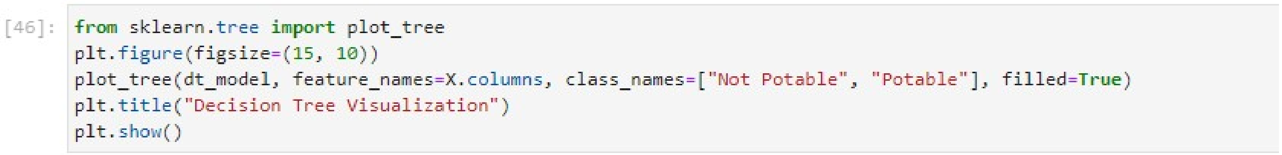
```
AdaBoost Model:
Accuracy: 0.62
Confusion Matrix:
 [[351  61]
 [189  55]]
Classification Report:
               precision    recall  f1-score   support

           0       0.65      0.85      0.74       412
           1       0.47      0.23      0.31       244

    accuracy                           0.62       656
   macro avg       0.56      0.54      0.52       656
weighted avg       0.58      0.62      0.58       656
```

4.Using Decision Tree

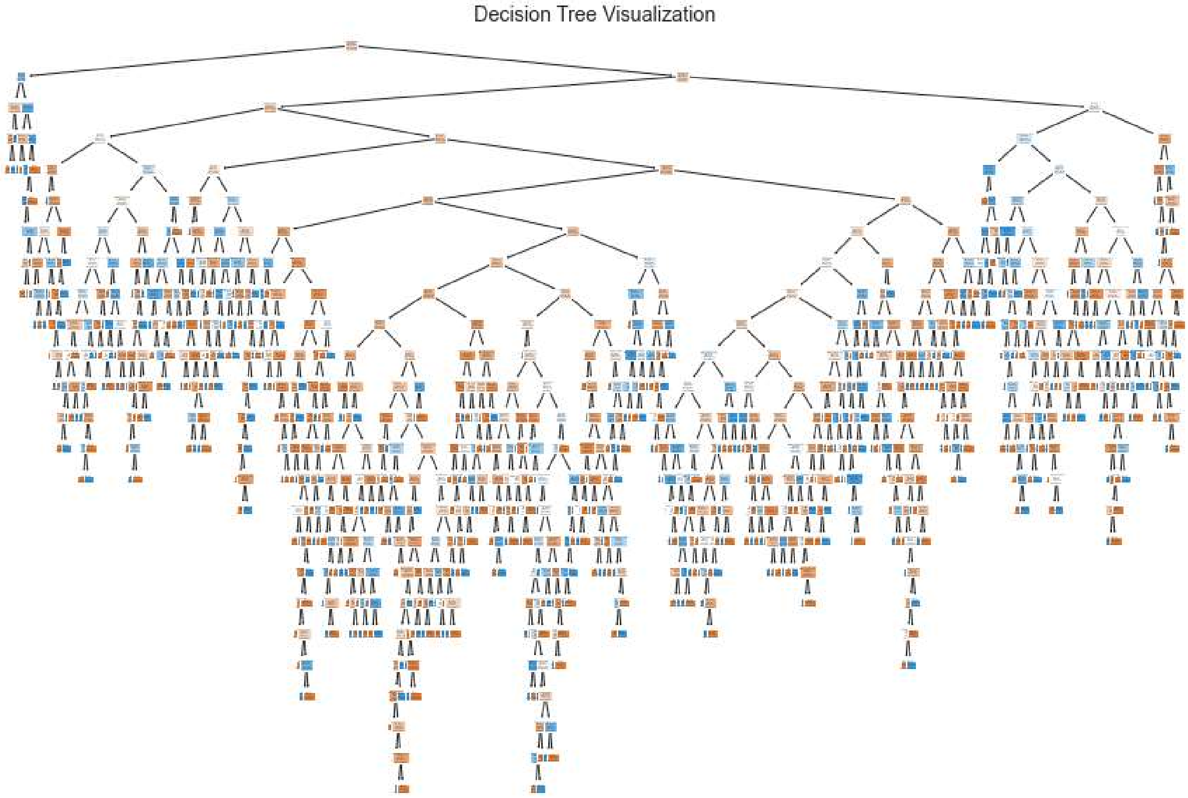


```
[45]: from sklearn.tree import DecisionTreeClassifier
      dt_model = DecisionTreeClassifier(random_state=42)
      dt_model.fit(X_train, y_train)
      y_pred_dt = dt_model.predict(X_test)
      accuracy_dt = accuracy_score(y_test, y_pred_dt)
      confusion_mat_dt = confusion_matrix(y_test, y_pred_dt)
      classification_rep_dt = classification_report(y_test, y_pred_dt)
      print("Decision Tree Model:")
      print(f'Accuracy: {accuracy_dt:.2f}')
      print('Confusion Matrix:\n', confusion_mat_dt)
      print('Classification Report:\n', classification_rep_dt)
```

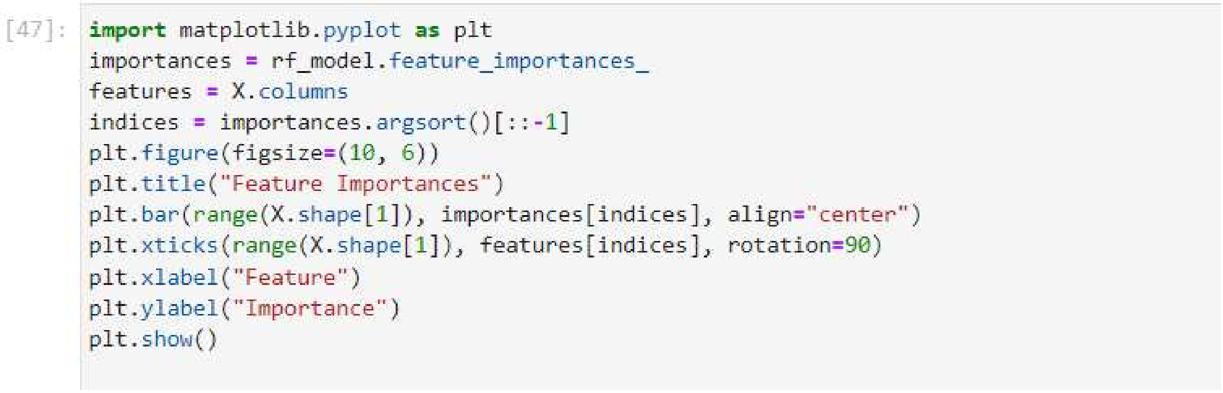
```
Decision Tree Model:
Accuracy: 0.58
Confusion Matrix:
 [[255 157]
 [120 124]]
Classification Report:
               precision    recall  f1-score   support

           0       0.68      0.62      0.65       412
           1       0.44      0.51      0.47       244

    accuracy                           0.58       656
   macro avg       0.56      0.56      0.56       656
weighted avg       0.59      0.58      0.58       656
```



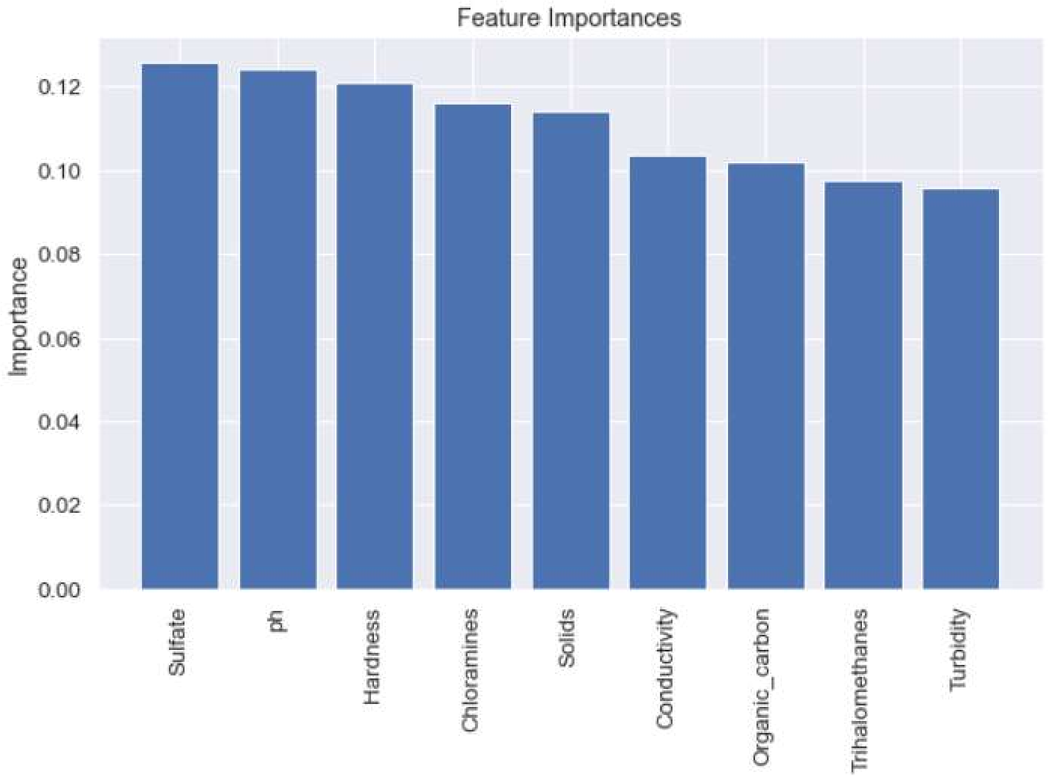
```
[46]: from sklearn.tree import plot_tree
      plt.figure(figsize=(15, 10))
      plot_tree(dt_model, feature_names=X.columns, class_names=["Not Potable", "Potable"], filled=True)
      plt.title("Decision Tree Visualization")
      plt.show()
```



VISUALIZING FEATURE IMPORTANCE



```
[47]: import matplotlib.pyplot as plt
      importances = rf_model.feature_importances_
      features = X.columns
      indices = importances.argsort()[::-1]
      plt.figure(figsize=(10, 6))
      plt.title("Feature Importances")
      plt.bar(range(X.shape[1]), importances[indices], align="center")
      plt.xticks(range(X.shape[1]), features[indices], rotation=90)
      plt.xlabel("Feature")
      plt.ylabel("Importance")
      plt.show()
```


Feature Importances
Importance
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Sulfate
ph
Hardness
Chloramines
Solids
Conductivity
Organic_carbon
Trihalomethanes
Turbidity