Analyzing and Predictive Modeling for Water Quality Analysis

Data Visualization

Histograms:

We utilized data visualization libraries, including Matplotlib and Seaborn, to create histograms for the water quality parameters. These histograms provide a visual representation of the distribution of each parameter within the dataset. By analyzing the histograms, we gained valuable insights into the range, distribution, and central tendencies of the parameters.

Scatter Plots:

In addition to histograms, we created scatter plots to explore the relationships between individual water quality parameters and water potability. Scatter plots help visualize any patterns or trends that might exist between the parameters and the target variable. This allowed us to identify any potential correlations or clusters that can aid in predictive modeling.

Correlation Matrix:

To better understand the interplay between the water quality parameters, we calculated and visualized a correlation matrix. This matrix provides insights into how each parameter correlates with others. A higher correlation suggests a stronger relationship, while a lower correlation indicates a weaker relationship. This information is crucial for feature selection and model development.

Predictive Modeling

Data Splitting:

Before building the predictive model, we divided the dataset into two subsets: a training set and a testing set. The training set is used to train the model, while the testing set is employed to evaluate its performance.

Logistic Regression:

For this project, we decided to implement a Logistic Regression model for the task of determining water potability. Logistic Regression is a suitable choice for binary classification tasks, and we aimed to predict whether water samples are potable or non-potable based on the water quality parameters.

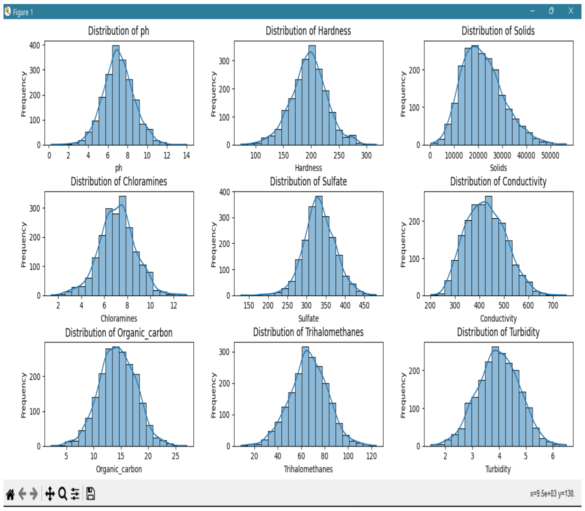
Model Evaluation:

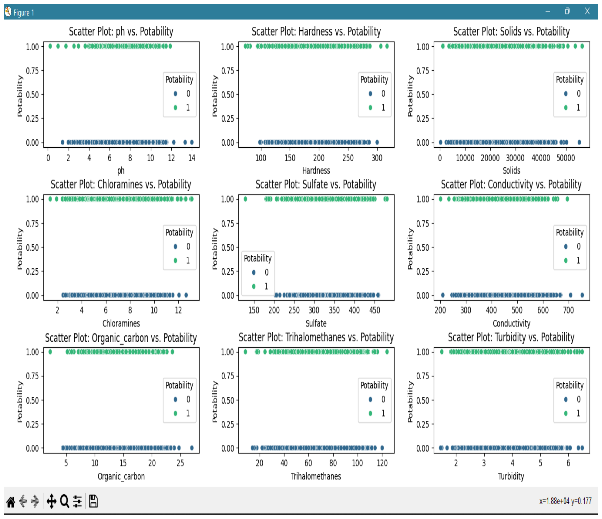
After training the Logistic Regression model, we evaluated its performance using various metrics, including accuracy, precision, recall, F1-score, and a confusion matrix. These metrics provide an overview of how well the model can distinguish between potable and non-potable water samples. Model evaluation is essential for assessing the model's effectiveness and identifying areas for improvement.

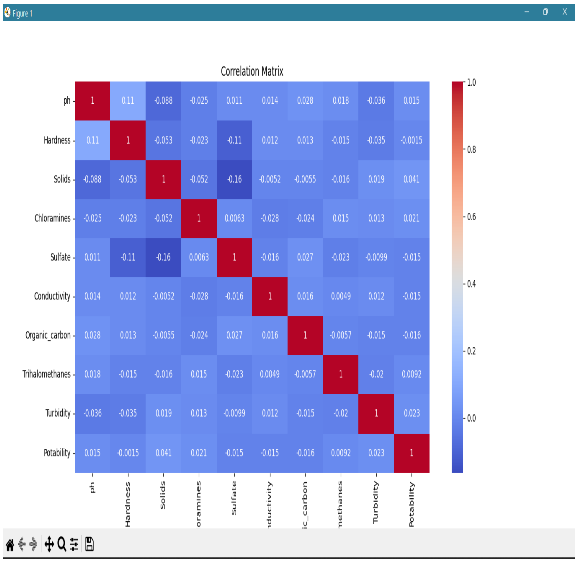
code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
# Load the preprocessed dataset
data = pd.read_csv("Dataloading_preprossed.csv")
# Data Visualization
# Create histograms for selected parameters
plt.figure(figsize=(10, 8))
for i, col in enumerate(data.columns[:-1]):
    plt.subplot(3, 3, i + 1)
    sns.histplot(data[col], bins=20, kde=True)
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
# Create scatter plots for selected parameter pairs
plt.figure(figsize=(10, 8))
for i, col in enumerate(data.columns[:-1]):
    plt.subplot(3, 3, i + 1)
    sns.scatterplot(
        data=data, x=col, y="Potability", hue="Potability",
palette="viridis"
    )
    plt.title(f"Scatter Plot: {col} vs. Potability")
    plt.xlabel(col)
    plt.ylabel("Potability")
plt.tight_layout()
plt.show()
# Calculate the correlation matrix and create a heatmap
correlation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
# Building a Predictive Model
# Data Splitting
X = data.drop("Potability", axis=1)
y = data["Potability"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
# Create a Logistic Regression model
rf_model = LogisticRegression()
rf_model.fit(X_train, y_train)
# Model Evaluation
y_pred = rf_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, zero_division=1)
confusion = confusion_matrix(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", report)
print("Confusion Matrix:\n", confusion)
```

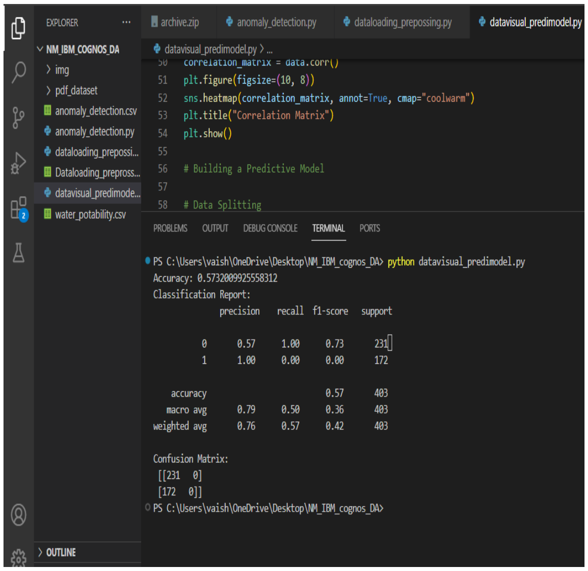
Results:







	ph	Hardness	Solids	oramines	Sulfate	nductivity	ic_carbon	methanes	Turbidity	Potability
ph	1	0.11	-0.088	-0.025	0.011	0.014	0.028	0.018	-0.036	0.015
Hardness	0.11	1	-0.053	-0.023	-0.11	0.012	0.013	-0.015	-0.035	-0.0015
Solids	-0.088	-0.053	1	-0.052	-0.16	-0.0052	-0.0055	-0.016	0.019	0.041
Chloramines	-0.025	-0.023	-0.052	1	0.0063	-0.028	-0.024	0.015	0.013	0.021
Sulfate	0.011	-0.11	-0.16	0.0063	1	-0.016	0.027	-0.023	-0.0099	-0.015
Conductivity	0.014	0.012	-0.0052	-0.028	-0.016	1	0.016	0.0049	0.012	-0.015
Organic_carbon	0.028	0.013	-0.0055	-0.024	0.027	0.016	1	-0.0057	-0.015	-0.016
Trihalomethanes	0.018	-0.015	-0.016	0.015	-0.023	0.0049	-0.0057	1	-0.02	0.0092
Turbidity	-0.036	-0.035	0.019	0.013	-0.0099	0.012	-0.015	-0.02	1	0.023
Potability	0.015	-0.0015	0.041	0.021	-0.015	-0.015	-0.016	0.0092	0.023	1



```python
correlation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()

# Building a Predictive Model

# Data Splitting
```

```
PS C:\Users\vaish\OneDrive\Desktop\NM_IBM_cognos_DA> python datavisual_predimodel.py
Accuracy: 0.5732009925558312
Classification Report:
              precision    recall  f1-score   support

           0       0.57      1.00      0.73       231
           1       1.00      0.00      0.00       172

    accuracy                           0.57       403
   macro avg       0.79      0.50      0.36       403
weighted avg       0.76      0.57      0.42       403

Confusion Matrix:
 [[231   0]
 [172   0]]
PS C:\Users\vaish\OneDrive\Desktop\NM_IBM_cognos_DA>
```

Conclusion:

This phase of the project emphasizes the importance of data visualization and predictive modeling to gain insights into water quality parameters and make informed decisions regarding water potability.