Water Quality Analysis using statistical methods,machine learning



**Introduction:**

Water quality analysis is a critical aspect of environmental monitoring and management. In recent years, advancements in data analytics have revolutionised the way we assess and manage water quality. This project aims to leverage data analytics techniques to gain deeper insights into water quality parameters, enabling more efficient decision-making and proactive environmental protection.

**Objective:**

**The objective of water quality monitoring is to obtain quantitative information on the physical, chemical, and biological characteristics of water via statistical sampling . The type of information sought depends on the objectives of the monitoring programme.**

**1.Statistical methods:**

Statistical methods primarily describe and interpret data patterns. Common techniques used in water quality analysis include:

Descriptive statistics: Provide a summary of the main aspects of the data, such as mean, median, variance, etc.

Correlation analysis: To identify the relationships between different water quality parameters.

Regression analysis: Predicts one variable based on another. For example, predicting pollutant levels based on rainfall amount.

**2**. **Machine learning methods:**

Machine learning methods can predict outcomes based on large datasets. Some commonly used techniques include:

Decision Trees and Random Forests: They work by splitting the data based on feature values and are especially useful for classification or regression tasks related to water quality.

Neural Networks: Especially deep learning networks, can be used to model complex relationships in the data and predict water quality.

Support Vector Machines (SVM): Used for classification and regression of water quality parameters.

K-Nearest Neighbors (KNN): A non-parametric method used for classification and regression.

Clustering (like K-Means): Can help in identifying patterns or groups in water quality datasets.

2. \*\*Data Preprocessing\*\*:

```python

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

# Load the data

data = pd.read\_csv('water\_data.csv')

# Split the data into train and test sets

X = data.drop('Quality', axis=1)

y = data['Quality']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# Scale the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

```

3. \*\*Statistical Analysis\*\*:

You can use Python's `statsmodels` library to get various statistics:

```python

import statsmodels.api as sm

# Using an OLS model for a statistical summary

X\_with\_const = sm.add\_constant(X\_train)

model = sm.OLS(y\_train, X\_with\_const).fit()

print(model.summary())

```

4. \*\*Machine Learning\*\*:

```python

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Using Random Forest for classification

clf = RandomForestClassifier()

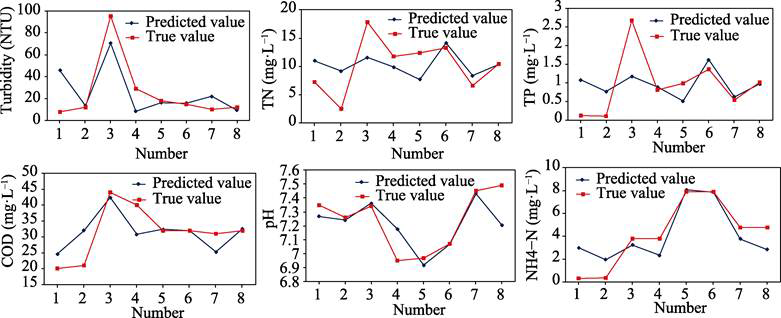
clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

```

**Output:**



**Link for the dataset:**

**DOWNLOAD DATASET:**

* Load dataset into DataFrame using pandas library.

Sample ID pH Temperature (Â°C) Turbidity (NTU) Dissolved Oxygen (mg/L) Conductivity (ÂµS/cm)   
1 7.25 23.1 4.5 7.8 342   
2 7.11 22.3 5.1 6.2 335   
3 7.03 21.5 3.9 8.3 356   
4 7.38 22.9 3.2 9.5 327   
5 7.45 20.7 3.8 8.1 352   
6 6.89 23.6 4.6 7.2 320   
7 7.19 21.2 4.2 8.8 350   
8 6.98 22.1 3.7 6.9 325   
9 7.31 20.4 4.1 8.4 360   
10 7.02 22.7 4.8 7.5 330   
11 7.24 22.4 4.3 8.6 347   
12 7.17 21.6 3.6 7.1 328   
13 6.95 22.3 4.1 6.4 341   
14 7.06 23.5 3.7 9.2 355   
15 7.48 20.8 3.4 7.9 329   
16 6.92 21.4 4.9 6.8 362   
17 7.11 22 4.4 8.1 336   
18 7.3 23.2 3.5 9.6 351   
19 7.13 21.1 4 7.5 319   
20 7.01 23 4.7 8.9 330   
21 6.83 22.5 3.3 6.1 348   
22 7.34 20.3 4.2 8 365   
23 7.16 23.4 4.5 7.7 326   
24 7.25 22.6 3.9 9.1 355   
25 7.39 21.9 4.1 7.4 317   
26 7.02 22.2 4.6 6.6 339   
27 7.27 21.8 3.7 8.7 354   
28 7.09 23.3 5 7 324   
29 7.15 20.6 4.4 8.5 358   
30 7.07 22.8 3.8 6.9 332   
31 7.22 22.5 4.3 8.9 345   
32 6.92 21.7 4.7 6.3 363   
33 7.13 23.1 3.6 8.2 347   
34 7.31 20.9 4 7.6 316   
35 7.03 22.6 4.9 7.9 331   
36 7.21 21.8 3.8 8.5 346   
37 7.13 22.5 4.2 7.7 321   
38 7.09 23.1 4.4 7.2 335   
39 7.35 21.6 3.9 9.2 357   
40 7.02 22 3.2 8.8 318   
41 7.28 23.5 3.7 9.5 353   
42 7.14 20.9 4.3 8.1 330   
43 6.96 22.9 4.8 6.5 344   
44 7.24 22.7 4.1 8 327   
45 7.37 21.5 4.5 9.3 361   
46 7.08 23.3 3.6 7.8 338   
47 7.16 21.1 4.4 7.3 352   
48 7.03 22.4 4 8.3 319   
49 7.32 23.2 4.5 8.9 346   
50 7.19 20.8 3.9 9.1 331   
51 7.12 22.1 4.6 6.8 340   
52 7.26 22.2 3.3 9.4 356   
53 7.04 21.6 3.8 8.4 322   
54 7.41 23 4.2 7.1 364   
55 7.08 21.7 4.1 8.6 337   
56 7.25 22.8 3.5 9.7 349   
57 7.17 20.6 4.7 7.6 328   
58 7.01 23.4 3.6 6.9 333   
59 6.9 22.5 4.2 6 357   
60 7.29 21.9 4.8 7.9 320   
61 7.19 23.1 3.9 9 342   
62 7.01 21.3 4.3 7.4 326   
63 7.15 22.6 4.5 8.2 355   
64 7.12 22 3.7 8.7 334   
65 6.94 23.2 4 6.6 348   
66 7.36 22.4 4.5 9.4 362   
67 7.02 20.7 3.8 8.2 323   
68 7.26 23.5 4.6 7.5 350   
69 7.08 21.4 3.3 9.1 330   
70 7.14 22.9 4.4 7.1 341   
71 7.31 21.2 3.6 8.8 329   
72 7.03 23 4.8 6.7 345   
73 7.18 22.5 3.5 9.3 358   
74 7.24 20.9 4.6 7.9 332   
75 7.14 22.8 4.2 8.5 347   
76 7.06 22.1 3.9 8 321   
77 7.29 23.3 3.4 9.5 351   
78 7.12 21.7 4.7 7.4 330   
79 7.01 22.6 3.2 8.2 339   
80 7.35 22.1 4.5 9.2 359   
81 7.04 23.1 4 6.4 324   
82 7.16 21.3 3.8 7.7 346   
83 7.22 22.3 4.3 8.9 329   
84 7.13 20.7 4.1 8.4 337   
85 7.25 23 3.9 9.6 354   
86 7.09 22.2 4.4 7 333   
87 7.19 21.1 3.3 9 348   
88 7.08 22.7 4.6 7.8 331   
89 7.37 22.4 3.7 9.1 362   
90 7.12 21.8 4.5 8.6 340   
91 7.27 23.4 4.2 7.2 327   
92 7.04 21.5 4.8 6.8 342   
93 7.15 23.1 4.4 8.1 355   
94 7.21 22.5 3.6 8.7 338   
95 6.98 21.9 4 8.9 328   
96 7.34 23.2 4.6 8.3 365   
97 7.02 21 3.9 7.5 332   
98 7.26 22.8 3.2 9.2 358   
99 7.09 23 4.3 7.3 327   
100 7.13 21.6 4.7 7.8 350   
101 7.16 22.2 3.3 8.9 331   
102 7.21 21.4 4.2 7.4 347   
103 7.32 23.3 4.1 9.5 366   
104 7.05 20.9 3.7 8.1 339   
105 7.13 22.5 4.3 7.9 326   
106 7.25 21.9 3.5 9.3 350   
107 7.08 23.1 4.5 6.5 319   
108 7.19 22 4.7 7.8 343   
109 7.01 20.8 3.6 8.4 330   
110 7.33 22.4 4.8 9 360   
111 7.07 23.2 3.5 7.7 327

**DATA EXPLORATION:**

* It is important to understand your data before preprocessing. You can explore your dataset using various pandas functions.
* This initial exploration will help you understand the structure of your data, identify missing values, and identify columns that need preprocessing.

**DATA EXPORT:**

* If you want to save your preprocessed dataset for future

use, you can export it to a CSV file.

**PROGRAM:**

# Import the necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset into a DataFrame

data = pd.read\_csv('/content/Wateranalysis\_Quality\_Dataset.csv')

# Display the first few rows of the dataset to inspect the data

print(data.head())

# Rename the columns as per your specified column names

data.columns = ["Sample ID","pH","Temperature ( °C)","Turbidity (NTU)", "Dissolved Oxygen (mg/L)", '

"Conductivity ( µS/cm)"]

# Display the updated DataFrame with column names

print(data.head())

# Perform any further data preprocessing or analysis as needed

# For example, you can clean missing data, handle data types,

etc.

# Save the preprocessed dataset if needed

# data.to\_csv('preprocessed\_wateranalysis\_quality\_data.csv', index=False)

# Data Visualization Example

# Plot a histogram of the Phlevels

plt.figure(figsize=(8, 6))

sns.histplot(data['SO2'], bins=20, kde=True)

plt.xlabel('turbidity Levels')

plt.ylabel('Dissolved oxygen')

plt.title('Distribution of hardenes in water')

plt.show()

# You can add more data visualization code here to explore

and understand your data.

**DATA PREPROCESSING:**

* Renaming Columns: If the names of the columns in your dataset do not match your needs, rename them.
* Missing Data Management: Review missing values and decide how to handle them. Options include deleting rows with missing values or entering missing values.
* Data Type Conversion: Make sure the column data type is appropriate for your analysis. Convert the column to the desired data type using .astype().
* Remove Duplicates: Check duplicate rows and remove them if any.
* Other preprocessing steps: Depending on your specific analysis, you may need to perform additional preprocessing such as scaling, encoding categorical variables, or creating new features.

**Application in Water Quality Analysis:**

* Prediction: Predict future water quality using historical data. This can be crucial for early warnings and preventive actions.
* Classification: Classify water samples into various quality categories, e.g., potable, non-potable, requires treatment, etc.
* Anomaly Detection: Detect unusual patterns that do not conform to expected behavior. Useful for spotting pollution events or equipment malfunction.
* Feature Importance: Understand which parameters (e.g., pH, turbidity, chemical concentration) have the most influence on water quality.

**Conclusion:**

In conclusion, while statistical methods provide a foundational approach to understanding and predicting water quality, machine learning techniques offer advanced and more accurate ways to model and predict complex water quality relationships. Combining both approaches can offer comprehensive insights into water quality trends and predictions.