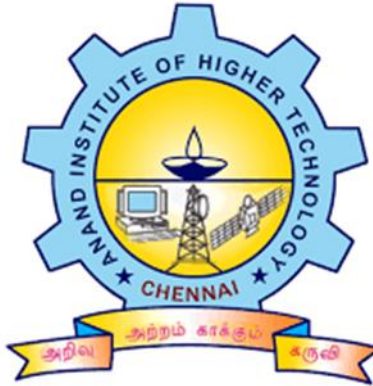


**ANAND INSTITUTE OF HIGHER TECHNOLOGY
OLD MAHABALIPURAM ROAD,
KALASALINGAM NAGAR,
KAZHIPATTUR.**



**WATER QUALITY ANALYSIS
By
DATA ANALYTICS WITH COGNOS**

PHASE – 4

PROJECT : WATER QUALITY ANALYSIS

PROJECT ID : PROJ_229797_TEAM_2

**TEAM MEMBERS: RANJITH KUMAR V
SHARATH P**

TEAM LEADER: SURENTHAR S

WATER QUALITY ANALYSIS

INTRODUCTION :-

Water quality analysis is the process of assessing and evaluating the physical, chemical, and biological characteristics of water to determine its suitability for various purposes, such as drinking, industrial use, agriculture, and environmental conservation. This analysis is crucial in ensuring the safety and sustainability of water resources. Here's an introduction to the key aspects of water quality analysis:

- ❖ Parameters
- ❖ Sources of Water
- ❖ Importance
 - Health and Safety
 - Environmental Conservation
 - Industrial and Agricultural Use



DATA VISUALISATION:-

1. **Line Chart:** Show trends in water quality parameters over time, such as pH levels, turbidity, or dissolved oxygen.
2. **Bar Chart:** Compare different water quality parameters at a specific location or across multiple locations.
3. **Heatmap:** Visualize spatial variations in water quality across a map, indicating areas with high or low values for specific parameters.
4. **Scatter Plot:** Display relationships between two variables, like temperature and dissolved oxygen, to identify correlations.
5. **Histogram:** Illustrate the distribution of a particular parameter, like pollutant concentration, within your dataset.
6. **Box Plot:** Show the spread and skewness of water quality data, revealing outliers and quartile ranges.
7. **Pie Chart:** Highlight the composition of different pollutants in a water sample.
8. **Radar Chart:** Compare water quality across multiple parameters in a single plot, useful for multi-parameter analysis.
9. **Stacked Area Chart:** Demonstrate the cumulative impact of different parameters on water quality over time.
10. **3D Surface Plot:** Depict water quality in a 3D space, useful for visualizing complex datasets. Remember to label axes, provide legends, and choose appropriate color schemes to make your data visualization clear and understandable. The choice of visualization depends on your specific dataset and the insights you want to convey.

IMPORTANCE OF VISUALIZING THE DATASET :-

The importance of visualizing datasets cannot be overstated in the realm of data analysis. Visualization serves as a powerful bridge between raw data and human comprehension. It enhances our ability to understand, interpret, and extract meaningful insights from complex datasets.

By transforming numbers and statistics into charts, graphs, and interactive displays, visualization offers several key advantages. Firstly, it enables us to detect patterns, trends, and outliers that might remain hidden in tabular data, facilitating more accurate and timely decision-making. Moreover, it supports data exploration by allowing users to interact with the data, making it easier to uncover

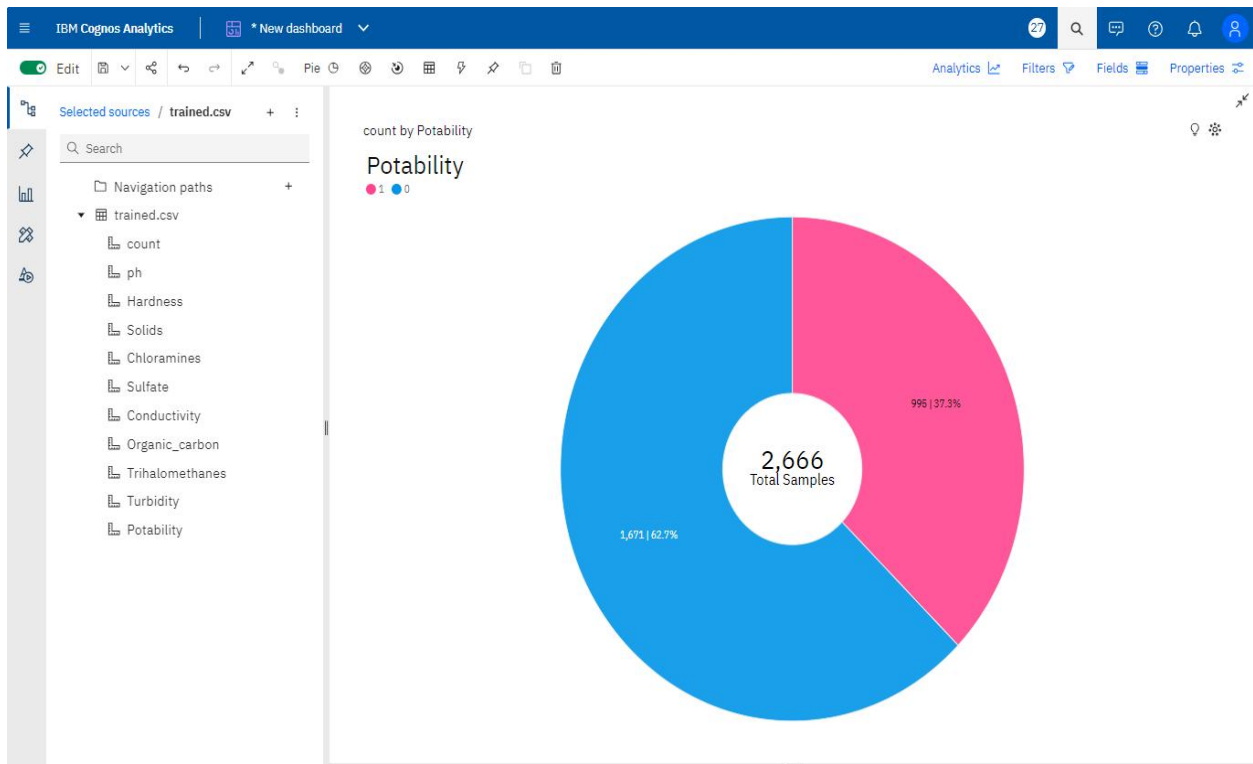


specific details and refine analysis.

This feature is particularly valuable in the age of big data, where sifting through vast datasets can be a formidable challenge. It is a

time-saving tool that provides a rapid overview of data, streamlining the analysis process.

VISUALIZATION OF THE DATASET USING COGNOS

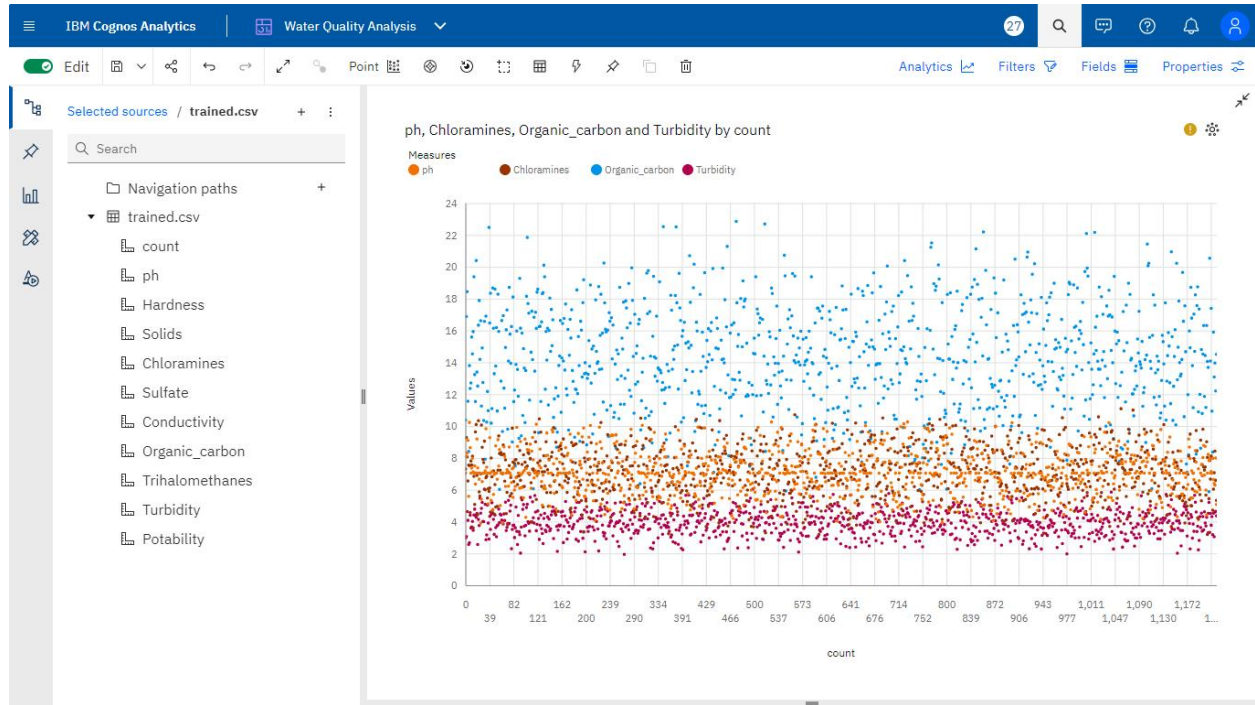


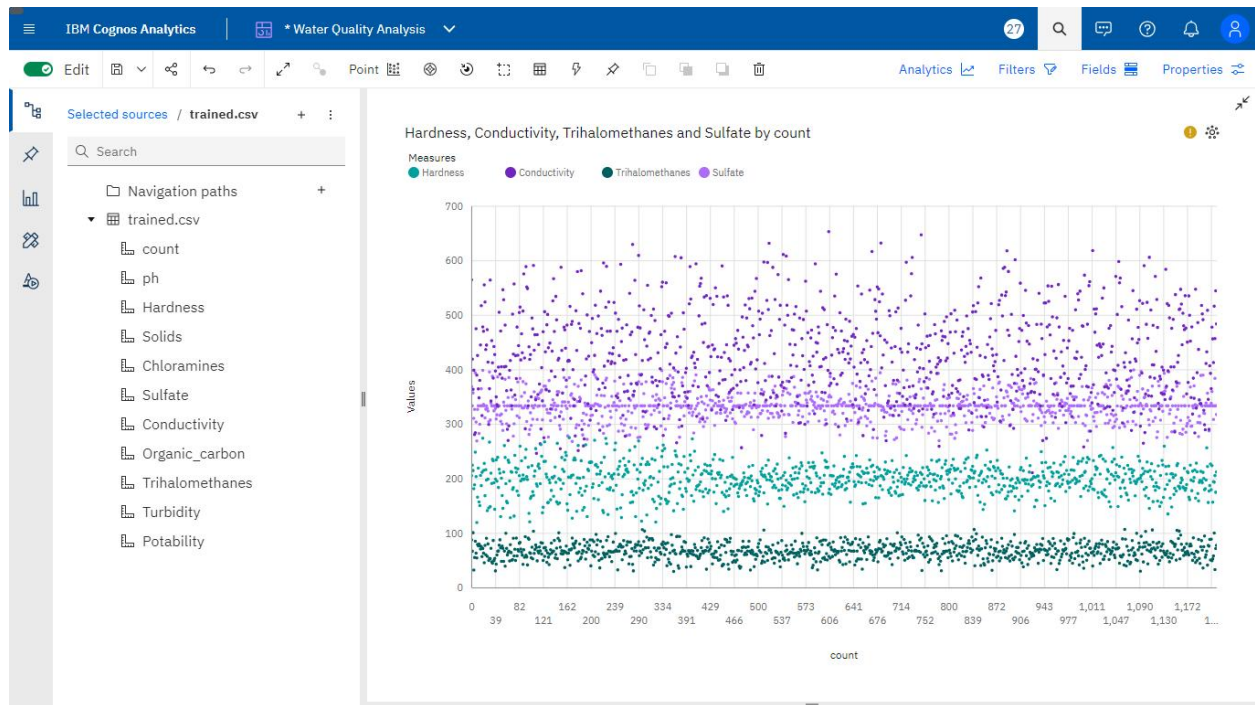
Visualization using Pie Chart :

Visualizing the Potability data using Pie Chart provided in the Cognos.

Here the insights of the Pie chart is the dataset contains 62.7% Not Potable and 37.3% Potable datas among the 2666 Water Samples.

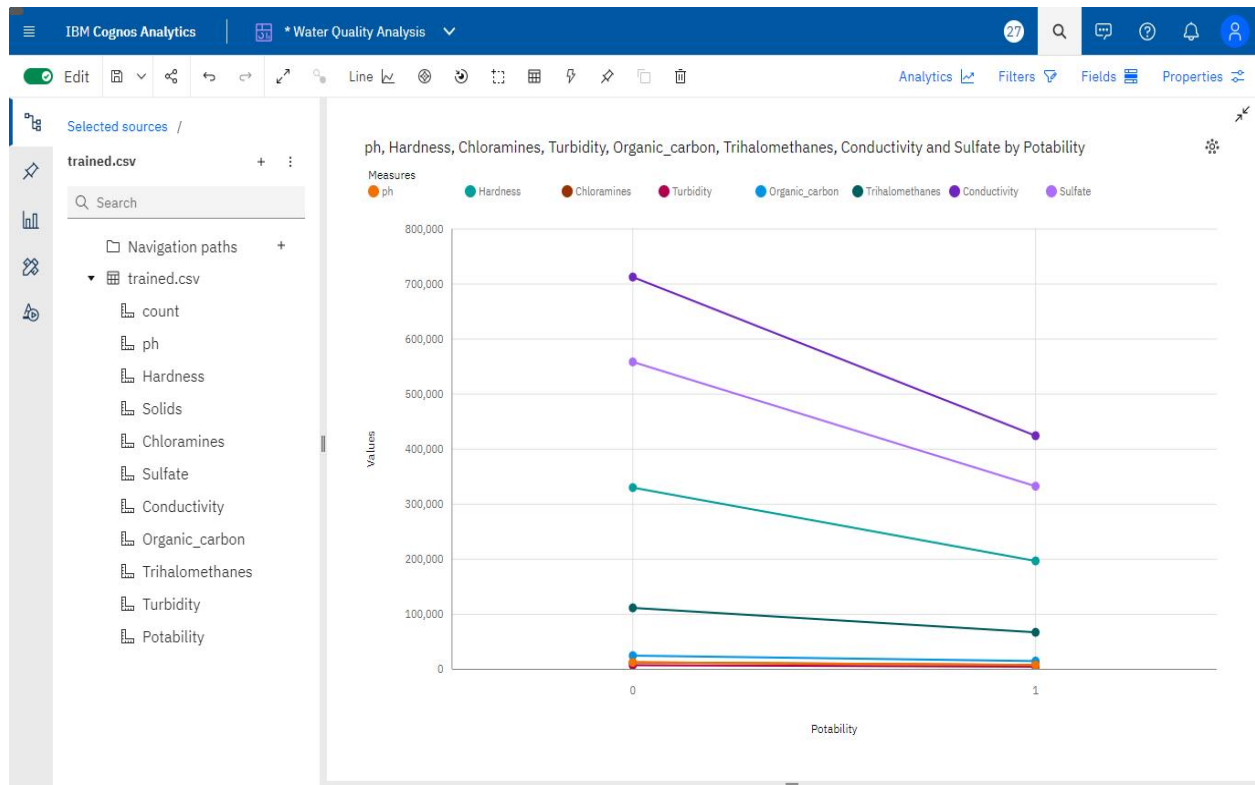
VISUALIZING USING THE POINTS:





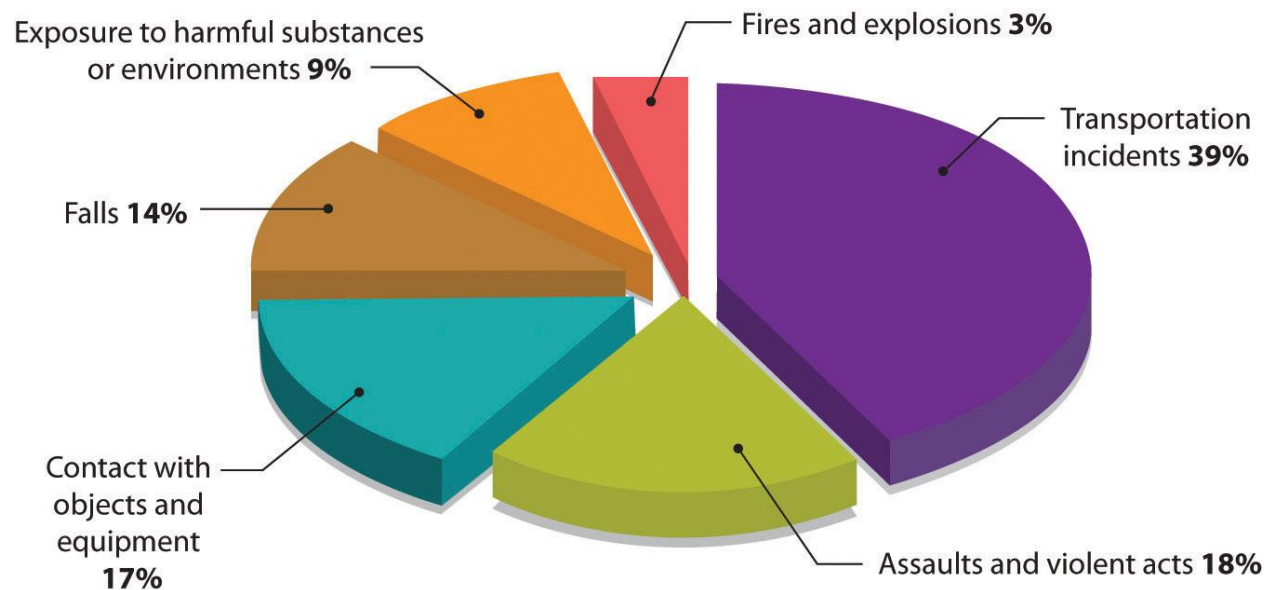
Visualizing the Distribution of each Water Quality Parameters provided in the Dataset using Point Chart provided in the Cognos.

VISUALIZING USING LINE CHART :-



Visualizing the Potability of each water quality parameters using Line Chart provided in the Cognos.

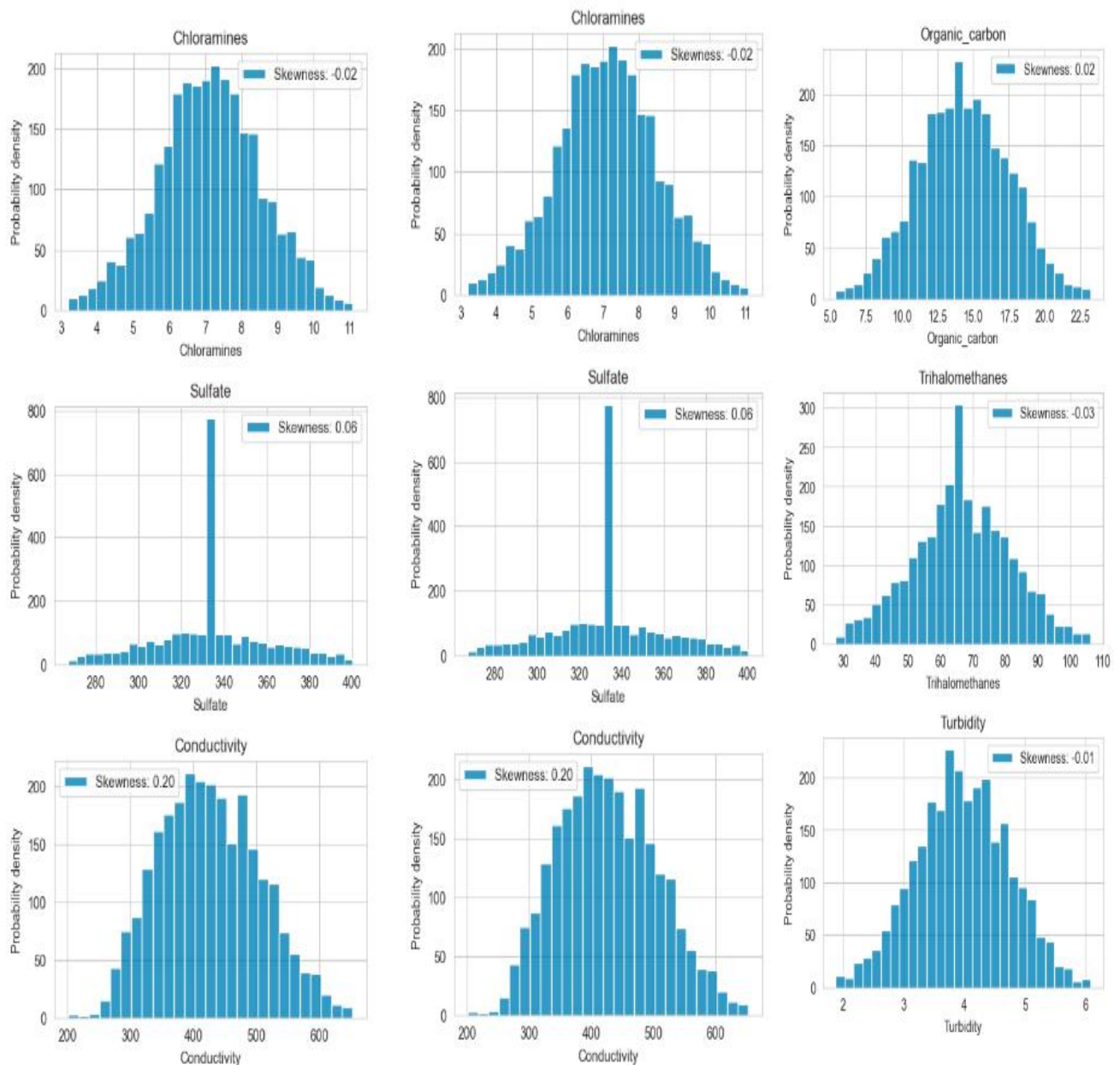
VISUALIZATION USING THE PIE CHART :-



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

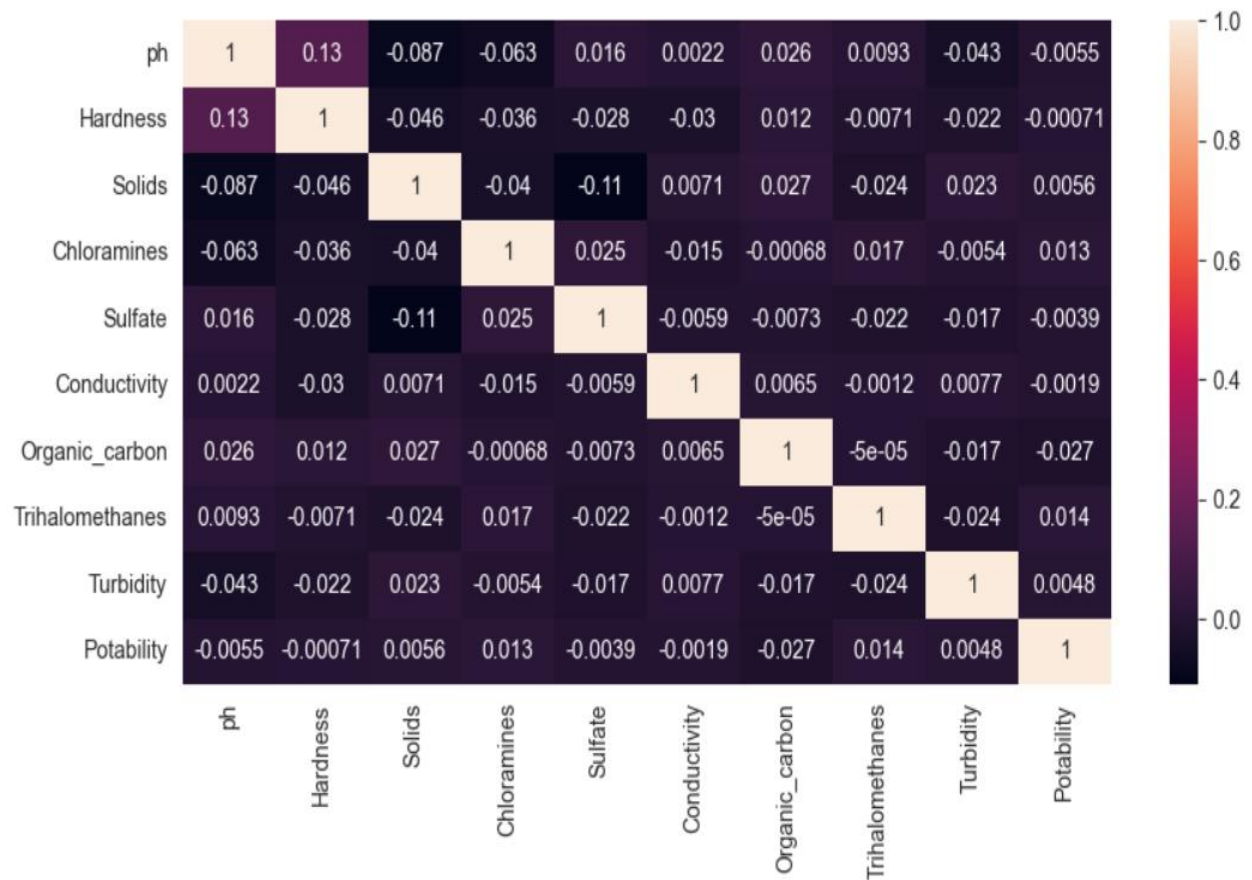
Visualizing the Potability of each water quality parameters using Pie Chart provided in the.

VISUALIZATION USING THE HISTPLOT :-



Visualizing the Potability Density of each water quality parameters using Histplot provided in the Cognos.

VISUALIZING CORREALTION :-



MACHINE LEARNING ALGORITHMS :-

- Machine learning algorithms play a crucial role in data analysis by enabling automated data modeling, pattern recognition, and predictive analytics. Machine learning algorithms enhance data analysis by automating complex tasks, uncovering hidden patterns, and providing data-driven insights.

LIBRARY :-

```
from sklearn.metrics import classification_report, accuracy_score
```

LOGISTIC REGRESSION :-

Logistic Regression is a commonly used algorithm for binary and multiclass classification problems. It is a fundamental class algorithm that models the probability of class membership using the sigmoid function. It estimates parameters to create a decision boundary that separates data points into different classes. Accuracy is used to evaluate the model's performance by comparing its predictions to actual class

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression()
log_reg.fit(X_train_final, y_train)
y_pred = log_reg.predict(X_test_final)
log_acc = accuracy_score(y_test, y_pred)
print(classification_report(y_test, y_pred))
print("Test Set Accuracy : ", log_acc)
```

	precision	recall	f1-score	support
0	0.62	1.00	0.77	497
1	0.00	0.00	0.00	303
accuracy			0.62	800
macro avg	0.31	0.50	0.38	800
weighted avg	0.39	0.62	0.48	800

Test Set Accuracy : 0.62125

labels.

The Test Accuracy of Logistic Regression is 62%

K-NEAREST NEIGHBOR CLASSIFIER :-

KNN is a simple yet effective supervised machine learning algorithm used for both classification and regression tasks. It operates on the principle of similarity and is based on the idea that data points with similar features are more likely to belong to the same class or have similar target values. KNN is a straightforward algorithm that relies on the concept of similarity to classify or predict data points. It is non-parametric and lazy (as it doesn't build a model during training),

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=9)
knn.fit(X_train_final, y_train)
y_pred = knn.predict(X_test_final)
knn_acc = accuracy_score(y_test, y_pred)

print(classification_report(y_test, y_pred))
print("Test Set Accuracy : ", knn_acc)
```

	precision	recall	f1-score	support
0	0.65	0.86	0.74	497
1	0.51	0.24	0.33	303
accuracy			0.62	800
macro avg	0.58	0.55	0.53	800
weighted avg	0.60	0.62	0.58	800

```
Test Set Accuracy : 0.62375
```

making it suitable for various tasks.

The Test Accuracy of K-Nearest Neighbor is 62%

SUPPORT VECTOR CLASSIFIER :-

SVM is a powerful algorithm for classification and regression tasks that aims to find an optimal hyperplane to separate different classes while maximizing the margin between them. It can handle both linear and nonlinear data, and its performance is evaluated using accuracy

```
from sklearn.svm import SVC

svc_classifier = SVC(class_weight = "balanced" , C=100, gamma=0.01)
svc_classifier.fit(X_train_final, y_train)
y_pred_scv = svc_classifier.predict(X_test_final)
svm_acc = accuracy_score(y_test, y_pred_scv)

print(classification_report(y_test, y_pred))
print("The Test Accuracy is : ",svm_acc)
```

	precision	recall	f1-score	support
0	0.66	0.87	0.75	497
1	0.55	0.26	0.35	303
accuracy			0.64	800
macro avg	0.60	0.56	0.55	800
weighted avg	0.62	0.64	0.60	800

The Test Accuracy is : 0.6325

metrics on training and test datasets.

The Test Accuracy of Support Vector Classifier is 63%

DECISION TREE CLASSIFIER :-

A Decision Tree is a machine learning algorithm used for both classification and regression tasks. It builds a tree-like structure of decisions based on feature values to classify data. It makes use of impurity measures like entropy to determine the best feature splits. By controlling the tree's depth and evaluating its accuracy, one can create a model that balances between fitting the training data well

```
from sklearn.tree import DecisionTreeClassifier

dtc = DecisionTreeClassifier(criterion='entropy',max_depth=5)
dtc.fit(X_train_final, y_train)
y_pred = dtc.predict(X_test_final)
dtc_acc= accuracy_score(y_test,dtc.predict(X_test_final))

print(classification_report(y_test, y_pred))
print("Test Set Accuracy : ", dtc_acc)
```

	precision	recall	f1-score	support
0	0.63	0.92	0.75	497
1	0.46	0.11	0.17	303
accuracy			0.61	800
macro avg	0.54	0.51	0.46	800
weighted avg	0.56	0.61	0.53	800

```
Test Set Accuracy : 0.61375
```

and generalizing to new data.

The Test Accuracy of Decision Tree Classifier is 61%

ALGORITHM ANALYSIS :-

This code facilitates the comparison of different machine learning models by recording and presenting their training and test accuracy

```
models = pd.DataFrame({  
    'Model': ['Logistic', 'KNN', 'SVM', 'Decision Tree Classifier'],  
    'Test': [log_acc, knn_acc, svm_acc, dtc_acc]  
})  
  
models.sort_values(by = 'Test', ascending = False)
```

	Model	Test
2	SVM	0.63250
1	KNN	0.62375
0	Logistic	0.62125
3	Decision Tree Classifier	0.61375

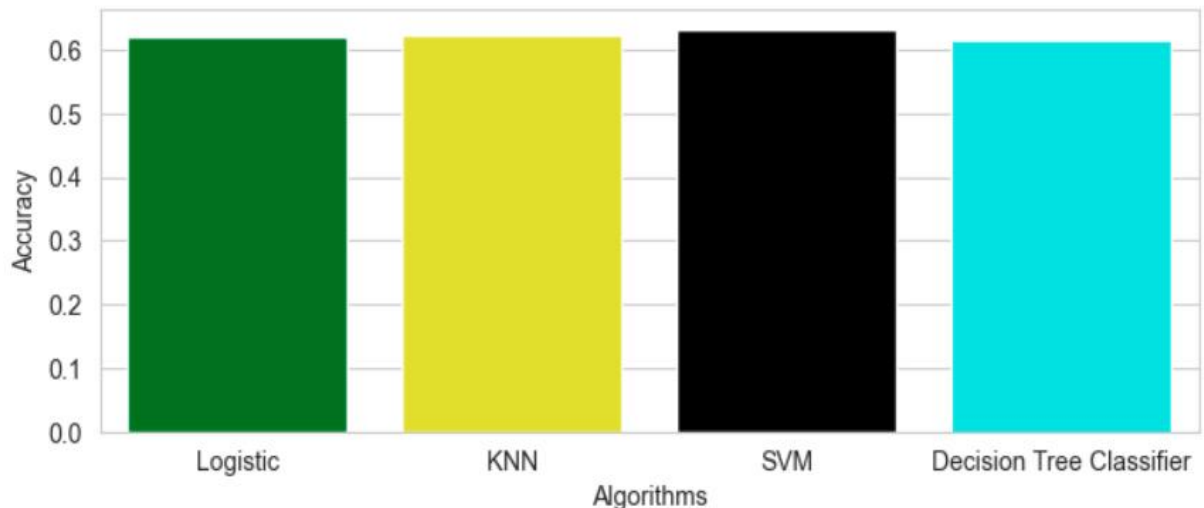
in a structured table, sorted by test accuracy for easy assessment.

This code organizes the performance metrics of different machine learning models in a structured tabular format, sorts the models based on their test accuracy, and provides a clear comparison of how well each model performed on the test dataset. This is a common practice to help data analysts and machine learning practitioners choose the best model for a given task.

ALGORITHM VISUALIZATION :-

This code generates a bar plot using Seaborn to visually compare and present the test accuracy of different machine learning algorithms. The choice of colors, style, and figure size enhances the readability

```
colors = ["green", "yellow", "black", "cyan"]
sns.set_style("whitegrid")
plt.figure(figsize=(8,3))
sns.barplot(x=models['Model'], y=models['Test'], palette = colors )
plt.ylabel("Accuracy")
plt.xlabel("Algorithms")
plt.show()
```



and presentation of the plot.

The accuracy score for support vector machine is 63%. As compare with other models the accuracy score is much higher in support vector machine.

DATA MODEL (MACHINE LEARNING ALGORITHM)

In machine learning, a "data model" is not a standard term. However, I can provide information on two key components: data and machine learning algorithms.

1. **Data:** Data is essential for training machine learning models. It consists of input features and corresponding target labels. Data can be structured (tabular data), unstructured (text, images), or semi-structured. High-quality and relevant data is crucial for the success of machine learning algorithms.

2. **Machine Learning Algorithm:** Machine learning algorithms are computational methods that learn patterns and make predictions from data. Common algorithms include linear regression, decision trees, random forests, support vector machines, neural networks, and more. The choice of algorithm depends on the problem you are trying to solve and the nature of your data.

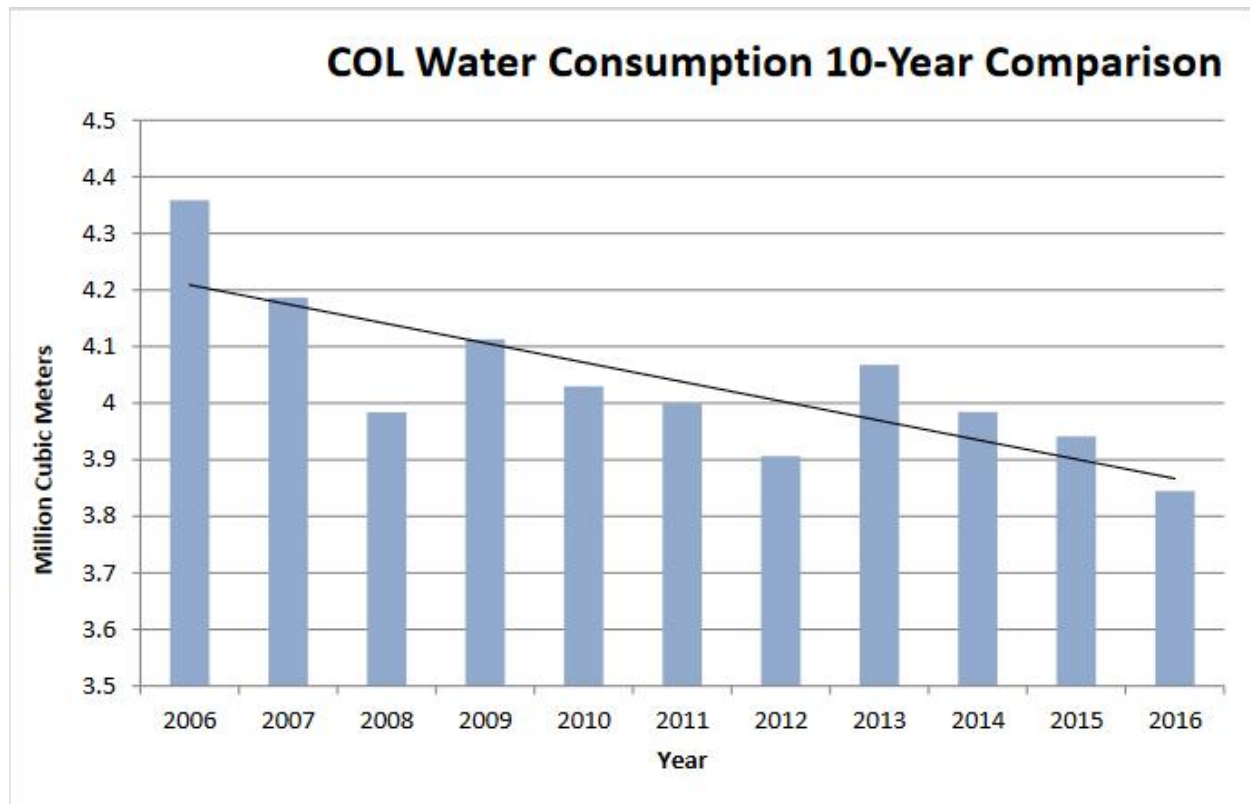
To create a machine learning model, you collect and preprocess your data, select an appropriate algorithm, train the model on the data, and then evaluate its performance. The model can then be used for tasks like classification, regression, clustering, or recommendation, depending on the problem you're addressing.

DATA OBSERVATION :-

The water_potability.csv file contains water quality metrics for 3276 different water bodies. It contains Water Quality Parameters such as

pH, Hardness, Solids, Chloramines, Sulphate, Conductivity, Organic_Carbon, Trihalomethanes, Turbidity. The Potability value defines the water quality based on the parameters given.

If Potability value is 0 then the water is Potable or the value is 1 then the water is Not Potable.



PREDICTIVE MODEL :-

A predictive model for water quality analysis is a valuable tool that uses historical water quality data to make predictions about future water quality conditions. These models can help in assessing and managing water resources, ensuring the safety of drinking water, and protecting aquatic ecosystems. Here's an overview of how to build a predictive model for water quality analysis:

POINTS:-

1. **Data Collection:** Gather historical water quality data from various sources, including measurements of physical, chemical, and biological parameters such as temperature, pH, dissolved oxygen, turbidity, nutrient levels, and pollutant concentrations. This data should cover a range of conditions and locations.
2. **Data Preprocessing:** Clean and preprocess the data, handling missing values, outliers, and inconsistencies. Transform the data as needed, such as encoding categorical variables or scaling numeric features.
3. **Feature Engineering:** Create relevant features or variables that can aid in prediction. For example, you may calculate monthly averages or seasonal patterns to capture trends in the data.
4. **Model Selection:** Choose an appropriate machine learning or statistical model for water quality prediction. Common choices include time series models (e.g., ARIMA or LSTM), regression models, decision trees, or ensemble methods.

5. **Split Data:** Divide the dataset into training, validation, and testing sets to train, validate, and evaluate the model's performance.

6. **Model Training :**Train the predictive model on the training dataset. The model should learn to identify patterns and relationships in the water quality data.

7. **Validation:** Use the validation dataset to fine-tune model hyperparameters and assess its performance. You may use metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or others to measure the model's accuracy.

8. **Testing :**Evaluate the model's performance on the testing dataset to ensure it can generalize to unseen data effectively.

9. **Deployment:** Once the predictive model performs well, it can be deployed to make real-time or future water quality predictions. For example, it can predict pollutant levels or the likelihood of harmful algal blooms in a water body.

10. **Continuous Monitoring:** Maintain the predictive model by retraining it regularly with new data. This helps the model adapt to changing water quality conditions over time.

11. **Interpretation and Action:**Interpret the model's predictions to make informed decisions and take appropriate actions. For instance, if the model predicts deteriorating water quality, it can trigger water treatment measures or environmental management actions.

Remember that building an effective predictive model for water quality analysis requires a good understanding of the domain, access to quality data, and ongoing monitoring and maintenance of the model to ensure its accuracy and relevance.

CODE:-

```
import tkinter as tk

from tkinter import Entry, Button, Label, Frame

import pickle

import pandas as pd

import numpy as np

import joblib

scaler = joblib.load("final_scaler.save")

model = pickle.load(open('final_model.pkl', 'rb'))

# Function to make a prediction

def predict():

    input_features = [float(entry.get()) for entry in entry_widgets]

    features_value = [np.array(input_features)]

    feature_names = ["ph", "Hardness", "Solids", "Chloramines", "Sulfate",

"Conductivity", "Organic_carbon","Trihalomethanes", "Turbidity"]

    df = pd.DataFrame(features_value, columns=feature_names)

    df = scaler.transform(df)

    output = model.predict(df)

    if output[0] == 1:

        prediction = "safe"

        result_label.config(text="Water is Safe for Human Consumption",

fg="#68FF00")

    else:

        prediction = "not safe"
```

```

        result_label.config(text="Water is Not Safe for Human Consumption",
fg="red")

# Create a Tkinter application window
app = tk.Tk()
app.title("Water Quality Prediction")

# Set the window geometry and background color
app.geometry("470x480")
app.configure(bg='#ABFFF1')

# Create a frame for the input fields and labels
input_frame = Frame(app, bg='#ABFFF1')
input_frame.pack(pady=10)

# Create input entry fields with labels
entry_labels = ["pH:", "Hardness:", "Solids:", "Chloramines:", "Sulfate:",
                "Conductivity:", "Organic Carbon:", "Trihalomethanes:",
                "Turbidity"]
entry_widgets = []
for label_text in entry_labels:
    label = Label(input_frame, text=label_text, bg='#ABFFF1', font=("copperplate
gothic bold", 14,"bold"), fg='black')
    label.grid(row=entry_labels.index(label_text), column=0, sticky='w', padx=10,
pady=5)
    entry = Entry(input_frame, font=("Arial", 12))
    entry.grid(row=entry_labels.index(label_text), column=1, padx=10, pady=5)
    entry_widgets.append(entry)

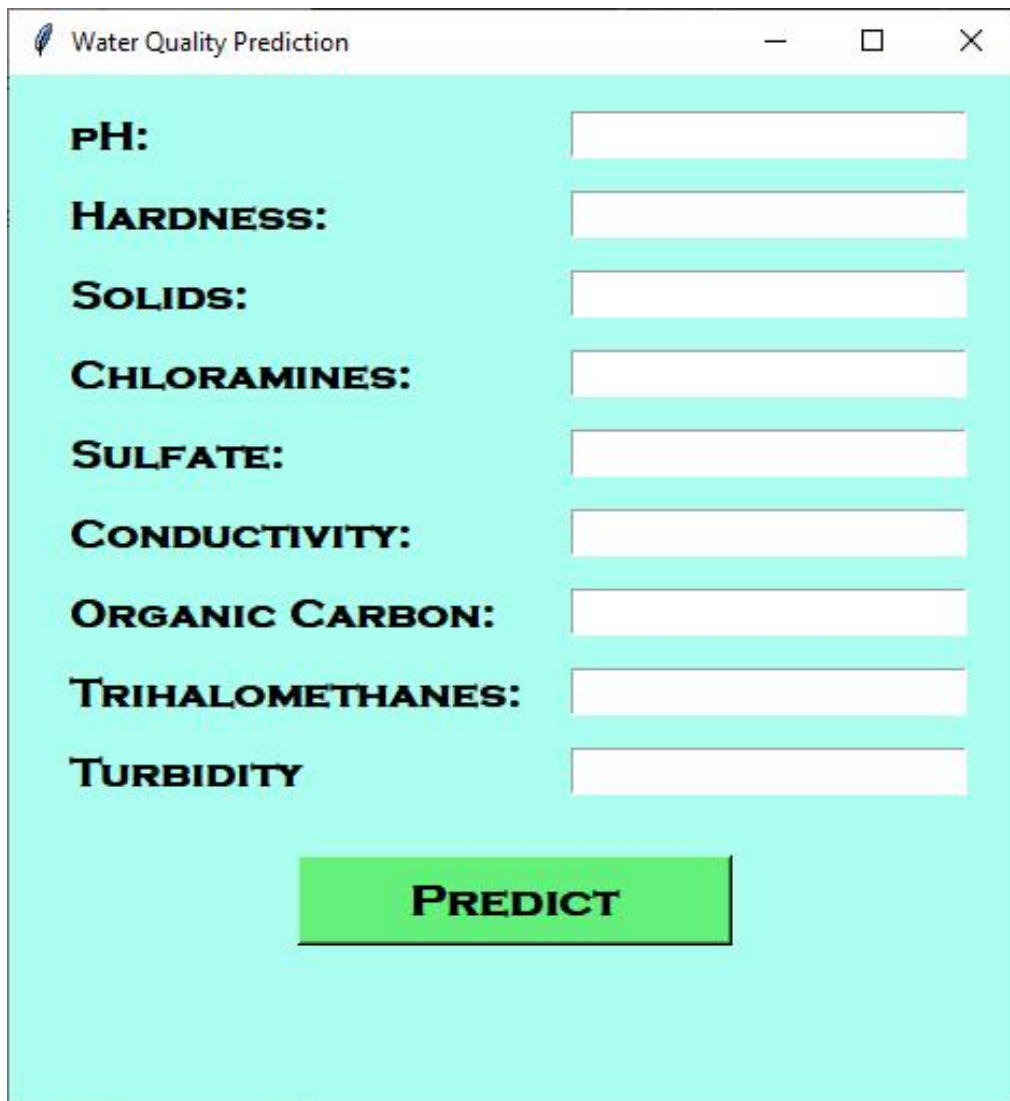
# Create a prediction button with custom style
predict_button = Button(app, text="Predict", command=predict, font=("copperplate
gothic bold", 16,"bold"), bg='#63F07B', fg='black',width = 12)
predict_button.pack(pady=10)

```

```
# Create a label to display the prediction with increased font size and custom
style
result_label = Label(app, text="", font=("copperplate gothic bold", 14),
bg="#ABFFF1")
result_label.pack()
# Start the Tkinter main loop
app.mainloop()
```

OUTPUT :-

HOME INTERFACE :-



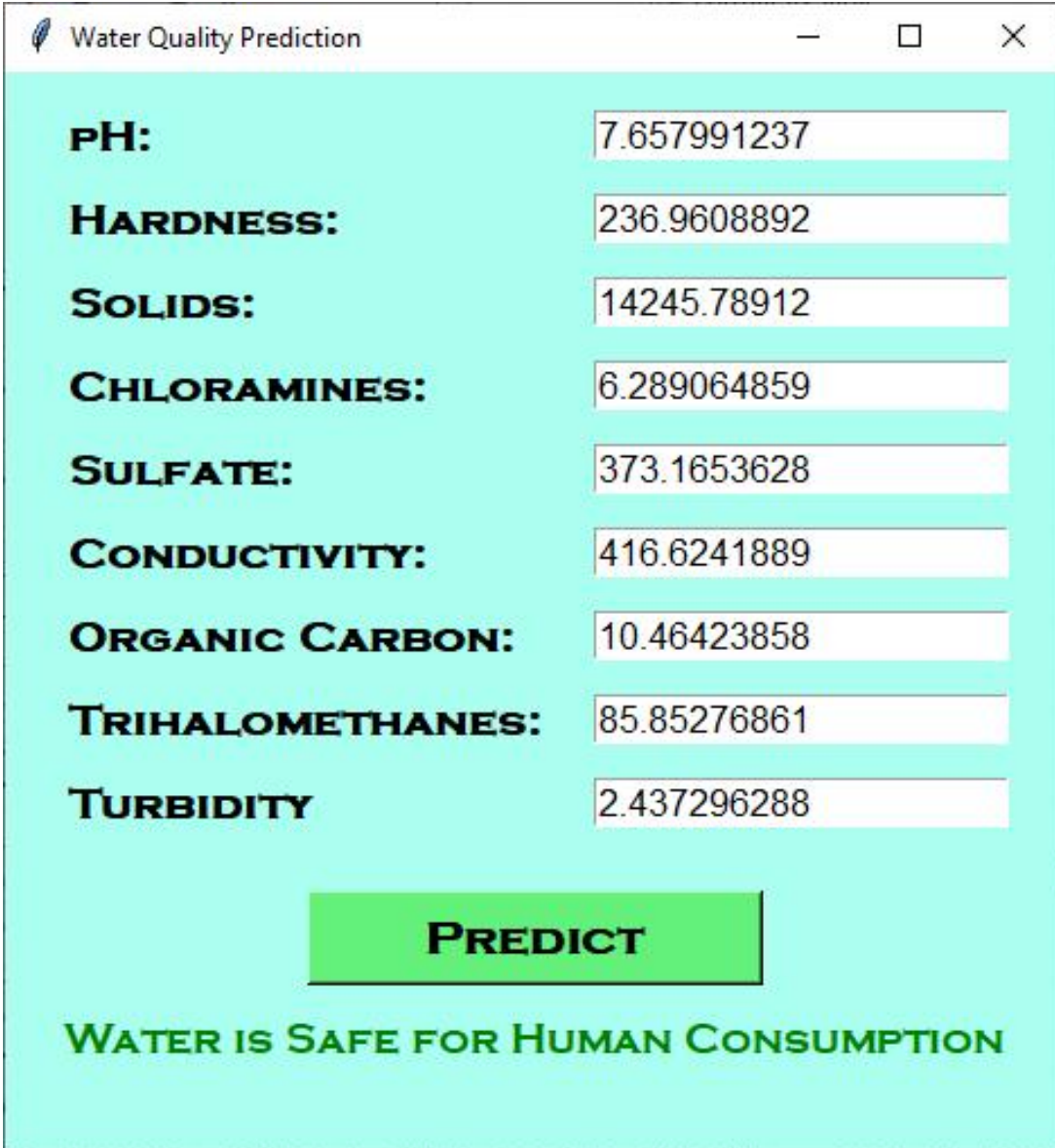
A screenshot of a web application window titled "Water Quality Prediction". The window has a light blue background and a white header bar with the title and standard window controls (minimize, maximize, close). The main content area contains a list of water quality parameters, each followed by a white input field. The parameters are: PH, HARDNESS, SOLIDS, CHLORAMINES, SULFATE, CONDUCTIVITY, ORGANIC CARBON, TRIHALOMETHANES, and TURBIDITY. Below the input fields is a large green button with the text "PREDICT" in white capital letters.

Parameter	Input Field
PH:	<input type="text"/>
HARDNESS:	<input type="text"/>
SOLIDS:	<input type="text"/>
CHLORAMINES:	<input type="text"/>
SULFATE:	<input type="text"/>
CONDUCTIVITY:	<input type="text"/>
ORGANIC CARBON:	<input type="text"/>
TRIHALOMETHANES:	<input type="text"/>
TURBIDITY	<input type="text"/>

PREDICT

Let's test the predictive model using the water parameters provided in the dataset.

TESTING WITH POTABLE VALUE :-



The image shows a software window titled "Water Quality Prediction". It contains a list of water quality parameters with corresponding numerical values entered in text boxes. At the bottom, there is a green "PREDICT" button and a green text label indicating the prediction result.

Parameter	Value
PH:	7.657991237
HARDNESS:	236.9608892
SOLIDS:	14245.78912
CHLORAMINES:	6.289064859
SULFATE:	373.1653628
CONDUCTIVITY:	416.6241889
ORGANIC CARBON:	10.46423858
TRIHALOMETHANES:	85.85276861
TURBIDITY	2.437296288

PREDICT

WATER IS SAFE FOR HUMAN CONSUMPTION

Here, we tested the Potable values provided in the dataset. So the Predictive model shows the result i.e “Water is Safe for Human Consumption”.

Water Quality Prediction

PH:	7.682872498
HARDNESS:	180.7013755
SOLIDS:	12105.72193
CHLORAMINES:	5.396716118
SULFATE:	296.2388769
CONDUCTIVITY:	469.8356255
ORGANIC CARBON:	15.83176344
TRIHALOMETHANES:	61.8020951
TURBIDITY	3.778606788

PREDICT

WATER IS NOT SAFE FOR HUMAN CONSUMPTION

TESTING WITH POTABLE VALUE :-

Here, we tested the Not Potable values provided in the dataset. So the Predictive model shows the result i.e “Water is Not Safe for Human Consumption”.

CONCLUSION :-

Conducting a comprehensive water quality analysis is essential to understanding and managing the state of water in a specific environment. The conclusions drawn from such an analysis can vary depending on the specific objectives, location, and parameters studied. However, some common conclusions that may emerge from water quality analysis include:

- **Assessment of Drinking Water Safety:** The analysis helps determine whether water sources meet regulatory standards for safe drinking, identifying potential contaminants such as bacteria, heavy metals, and chemicals.
- **Environmental Impact:** It provides insights into the impact of human activities on aquatic ecosystems, assessing the health of rivers, lakes, and oceans and identifying sources of pollution.
- **Agricultural and Industrial Suitability:** Water quality analysis can determine if water is suitable for irrigation, industrial processes, or recreational activities based on factors like salinity, pH, and nutrient levels.
- **Eutrophication and Algal Blooms:** Monitoring nutrient levels can help identify areas at risk of eutrophication and harmful algal blooms, which can have ecological and health implications.
- **Regulatory Compliance:** Conclusions may relate to adherence to local and national water quality regulations, enabling authorities to take corrective actions if necessary.
- **Impact on Biodiversity:** Understanding water quality is vital for assessing its impact on aquatic life, including fish populations, invertebrates, and the overall biodiversity of a given ecosystem.
- **Seasonal Variations:** Water quality may vary with seasons, so conclusions often highlight these fluctuations and their implications.

- Long-Term Trends: Water quality analysis can reveal long-term trends, such as gradual improvement or deterioration, helping to guide conservation and management efforts.
- In all cases, water quality analysis is a critical tool for ensuring the responsible management of water resources and safeguarding both human health and the environment.