PROJECT DOCUMENTATION & SUBMISSION

WATER QUALITY ANALYSIS

Date	**29-10-2023**
Team ID	**1278**
Project Name	**Water Quality Analysis**

TABLE OF CONTENT **PAGE.NO**

1.Introduction:

Access to clean, safe drinking water stands as a cornerstone of human well-being, impacting health, sanitation, and overall quality of life. In today's data-driven world, the analysis of water quality data has become indispensable, forming the basis for informed decision-making and public health initiatives. This project embarks on a vital exploration, delving deep into the intricate realm of water quality assessment. The dataset under scrutiny contains a plethora of parameters, ranging from pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, to turbidity. Each of these factors holds a key to understanding the purity and safety of water for consumption, making this analysis a significant endeavor.

At its core, this project is driven by a fundamental objective: to ensure that water, a fundamental human necessity, meets the stringent standards required for consumption. Clean water is not merely a privilege; it is a right, essential for the very survival of communities and societies. Beyond its elemental importance, water quality also directly impacts public health. Contaminated water sources can lead to a myriad of waterborne diseases, posing severe threats to communities, especially in regions where resources are scarce. In this context, the rigorous analysis of water quality data emerges as a crucial endeavor, aligning with global goals outlined in sustainable development agendas to provide universal access to safe and affordable drinking water.

The multifaceted nature of this project is underscored by its diverse objectives. Firstly, it entails a comprehensive analysis of the provided dataset. Through advanced statistical methods, this analysis aims to unravel the intricate patterns embedded within the data. By identifying correlations and deviations from established standards, the project seeks to paint a vivid picture of the water quality landscape. This understanding is not merely theoretical; it translates into tangible, real-world implications. It empowers policymakers, environmentalists, and communities alike with the knowledge necessary to advocate for and enforce water quality regulations, ensuring that the water supplied to households is devoid of harmful contaminants.

In tandem with this analysis, the project ventures into the realm of predictive modeling. By employing sophisticated machine learning techniques like the Random Forest Classifier, the analysis goes beyond mere observation, diving into the realm of anticipation. Predictive modeling serves as a proactive tool, enabling the identification of potential water quality issues before they escalate. Moreover, the integration of methods like Synthetic Minority Over-sampling Technique (SMOTE) showcases the project's commitment to addressing inherent challenges in the dataset, such as class imbalances. This not only enhances the accuracy of predictions but also underscores the project's commitment to robust, nuanced analyses

Innovation in data visualization is another cornerstone of this project. Traditional data analysis methods are often perceived as dense and esoteric. However, the project shatters this stereotype by harnessing innovative visualization techniques. Through the artful presentation of data using histograms, boxplots, scatter plots, and correlation heatmaps, the analysis is not confined to the realm of experts. It becomes accessible, comprehensible, and relatable to the average citize. These visualizations are not just aesthetically pleasing; they are informative, serving as educational tools that bridge the gap between complex data and public understanding.

Beyond the technical aspects, this project holds immense societal significance. It is a beacon of awareness, shining light on the critical issue of water quality. By distilling complex data into actionable insights, it empowers communities to make informed choices. In a world where climate change and pollution threaten the very resources we depend on, this project stands as a testament to the potential of data-driven solutions. It is a testament to human ingenuity and innovation, showcasing how technology can be harnessed not just for scientific advancement but for the betterment of society as a whole. As the project unfolds, it embodies the spirit of progress, the essence of knowledge, and the promise of a safer, healthier future for all.

2. Problem Statement

Definition: 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 potability based on various parameters. This project includes defining analysis objectives, collecting water quality data, designing relevant visualizations, and building a predictive model.

Data: We have a dataset containing key water quality parameters such as pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, Turbidity, and Potability.

Objective

- To provide an in-depth analysis of the design and innovation strategies for the analysing water quality data to assess the suitability of water for specific purposes, such as drinking.
- Access to clean and safe drinking water is a fundamental necessity for human well-being.
- It is essential for maintaining public health and preventing waterborne diseases.

3.Design Thinking

Analysis Objectives:

The primary objectives of this water quality analysis are to assess water potability, identify deviations from established standards, and understand the relationships among different parameters. By achieving these goals, we aim to provide valuable insights into the quality of the provided water dataset.

Data Collection:

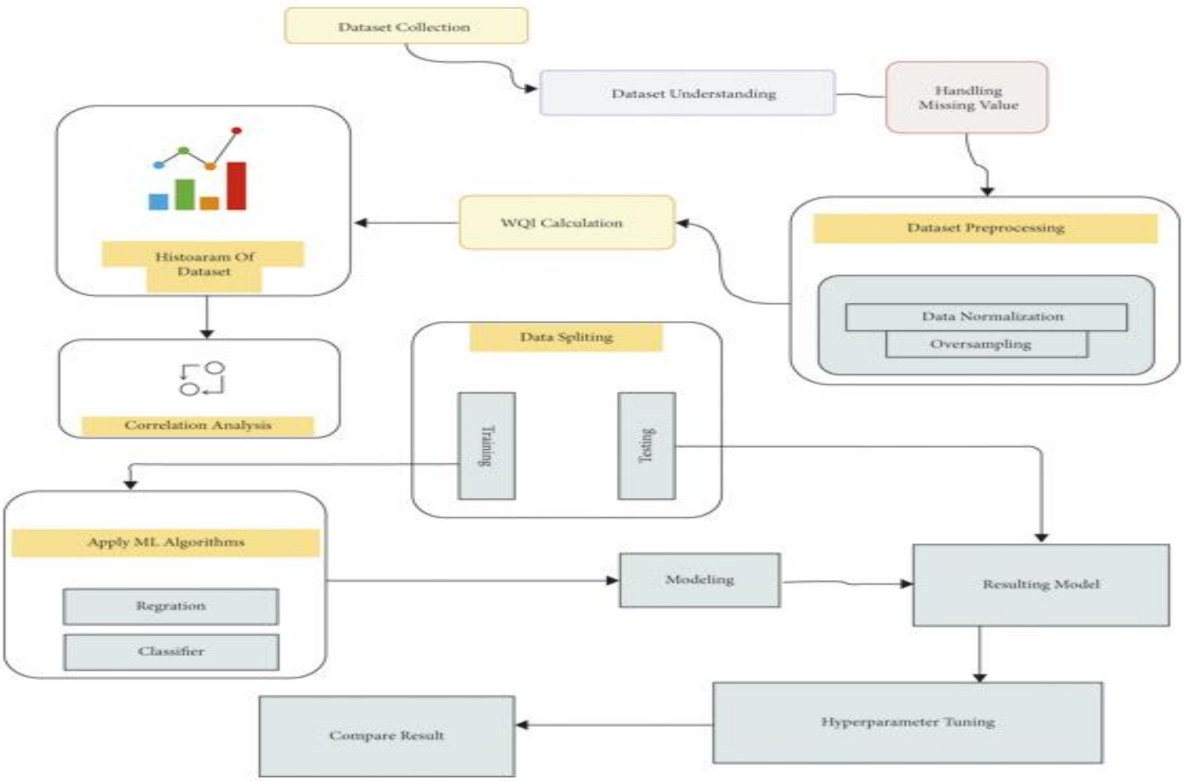
The analysis utilizes the provided water quality data, encompassing essential parameters such as pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, and Turbidity. This dataset forms the foundation for our analysis and modeling.

Visualization Strategy:

The visualization strategy involves employing various tools to effectively communicate the data insights. Histograms, box plots, and scatter plots are utilized to visualize parameter distributions and identify outliers. Additionally, a correlation heatmap is generated to understand the relationships among different parameters. These visualizations are crucial for gaining a comprehensive understanding of the dataset's characteristics and deviations.

Predictive Modeling:

For predictive modeling, machine learning algorithms are employed to forecast water potability based on the provided parameters. The selected algorithm for this analysis is the Random Forest Classifier. Features such as pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, and Turbidity are used as input variables for the predictive model. The choice of features and the algorithm is vital for accurate predictions and is based on their relevance to water quality standards.



4.Analysis Objectives:

- The primary objective of this water quality analysis is to assess the potability of water based on various water quality parameters.
- The analysis aims to develop a predictive model that can accurately determine whether a given sample of water is potable or not.
- This determination is crucial for ensuring the safety and quality of drinking water supplied to consumers.

5.Exploratory Data Analysis (EDA)

- Exploratory Data Analysis was performed through various visualizations.
- Histograms were used to understand the distribution of different water quality parameters, helping identify patterns and potential outliers.
- Boxplots provided insights into the spread and presence of outliers in the numerical features.
- Scatter plots and pair plots were utilized to visualize relationships between different variables, especially concerning potability.
- A correlation heatmap was generated to understand the interdependencies among the features.

6. Data Preprocessing

- In the data preprocessing phase, missing values in the dataset were handled by filling them with the mean values of their respective columns.
- Outliers were detected using the Interquartile Range (IQR) method, and numerical features were scaled for modelling.
- Additionally, the data was balanced using the Synthetic Minority Over-sampling Technique (SMOTE) to handle class imbalance, ensuring a more robust predictive model.

IMPORT SECTION

In [19]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import missingno as msno
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import RandomizedSearchCV,
RepeatedStratifiedKFold, train_test_split
from sklearn.metrics import precision_score, confusion_matrix
from sklearn import tree
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

```python
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer
from imblearn.over_sampling import SMOTE
from sklearn.pipeline import Pipeline, make_pipeline
```

DATASET

In [20]:

```python
#Displaying the dataset file.
data = pd.read_csv("water_potability.csv")
data
```

Out[20]:

	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

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
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

In [21]:

```
#Describing the dataset.
data.describe()
```

Out[21]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
count	2785.000000	3276.000000	3276.000000	3276.000000	2495.000000	3276.000000	3276.000000	3114.000000	3276.000000	3276.000000
mean	7.080795	196.369496	22014.092526	7.122277	333.775777	426.205111	14.284970	66.396293	3.966786	0.390110
std	1.594320	32.879761	8768.570828	1.583085	41.416840	80.824064	3.308162	16.175008	0.780382	0.487849

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
min	0.000000	47.432000	320.942611	0.352000	129.000000	181.483754	2.200000	0.738000	1.450000	0.000000
25%	6.093092	176.850538	15666.690297	6.127421	307.699498	365.734414	12.065801	55.844536	3.439711	0.000000
50%	7.036752	196.967627	20927.833607	7.130299	333.073546	421.884968	14.218338	66.622485	3.955028	0.000000
75%	8.062066	216.667456	27332.762127	8.114887	359.950170	481.792304	16.557652	77.337473	4.500320	1.000000
max	14.000000	323.124000	61227.196008	13.127000	481.030642	753.342620	28.300000	124.000000	6.739000	1.000000

In [22]:

```
#Getting information(type)
data.info()
```

```
<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
```

6.1.HANDLING OF MISSING VALUES

In [11]:

```
#Displaying the missing values in each column.
```

```
print("NUMBER OF MISSING VALUES IN EACH COLUMN :")
NULL=data.isnull().sum()
NULL
```

```
NUMBER OF MISSING VALUES IN EACH COLUMN :
```

Out[11]:

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

In [34]:

```
#Filling the missing values with average.
data['ph']=data['ph'].fillna(data['ph'].mean())
data['Sulfate']=data['Sulfate'].fillna(data['Sulfate'].mean())
data['Trihalomethanes']=data['Trihalomethanes'].fillna(data['Trihalomethanes'
].mean())
data
```

Out[34]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.080795	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	333.775777	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	333.775777	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
...	...	...	...	...	...	...	...	...	...	...

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
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	333.775777	392.449580	19.903225	66.396293	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	333.775777	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	333.775777	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	333.775777	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

In [35]:

```
print("NUMBER OF MISSING VALUES IN EACH COLUMN AFTER FILLING THE AVERAGE :")
data.isnull().sum()
```

```
NUMBER OF MISSING VALUES IN EACH COLUMN AFTER FILLING THE AVERAGE :
```

Out[35]:

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

In [15]:

```
#Finding the number of unique values.
data.nunique()
```

Out[15]:

```
ph                 2786
Hardness           3276
Solids             3276
Chloramines        3276
```

```
Sulfate            2496
Conductivity       3276
Organic_carbon     3276
Trihalomethanes    3115
Turbidity          3276
Potability            2
dtype: int64
```

In [18]:

```
#Display the file type.
data.dtypes
```

Out[18]:

```
ph                 float64
Hardness           float64
Solids             float64
Chloramines        float64
Sulfate            float64
Conductivity       float64
Organic_carbon     float64
Trihalomethanes    float64
Turbidity          float64
Potability           int64
dtype: object
```

---

7. Data Visualization

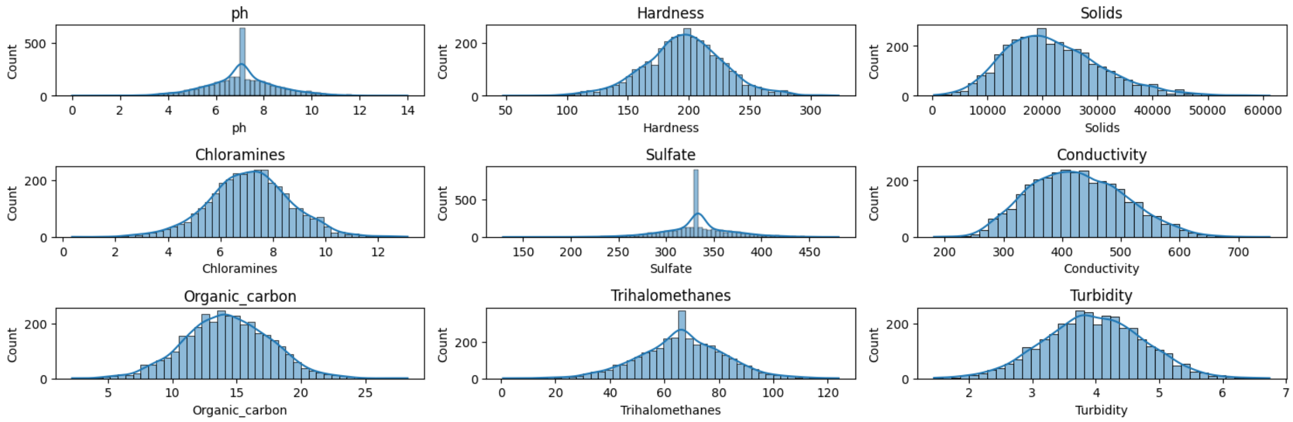
- Histograms were employed to visualize the distributions of key water quality parameters.
- Boxplots helped identify outliers in these parameters, showcasing their variability.
- Scatter plots and pair plots provided insights into the relationships between parameters, especially concerning potability.
- The correlation heatmap illustrated the correlations between different variables, highlighting their impact on water quality.

7.1.Histogram & Distribution.

In [37]:

```python
def show_distributions(columns: list, data: pd.DataFrame, nrows: int = 1,
ncols: int = 3):
    # This function creates distribution subplots.
    fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(15, 5))
    axes = axes.ravel()
    for index, column in enumerate(columns):
        sns.histplot(data[column], kde=True, ax=axes[index])
        axes[index].set_title(column)

        # Adjust layout
    plt.tight_layout()
    plt.show()
show_distributions(data.columns[:-1], data,3,3)
```

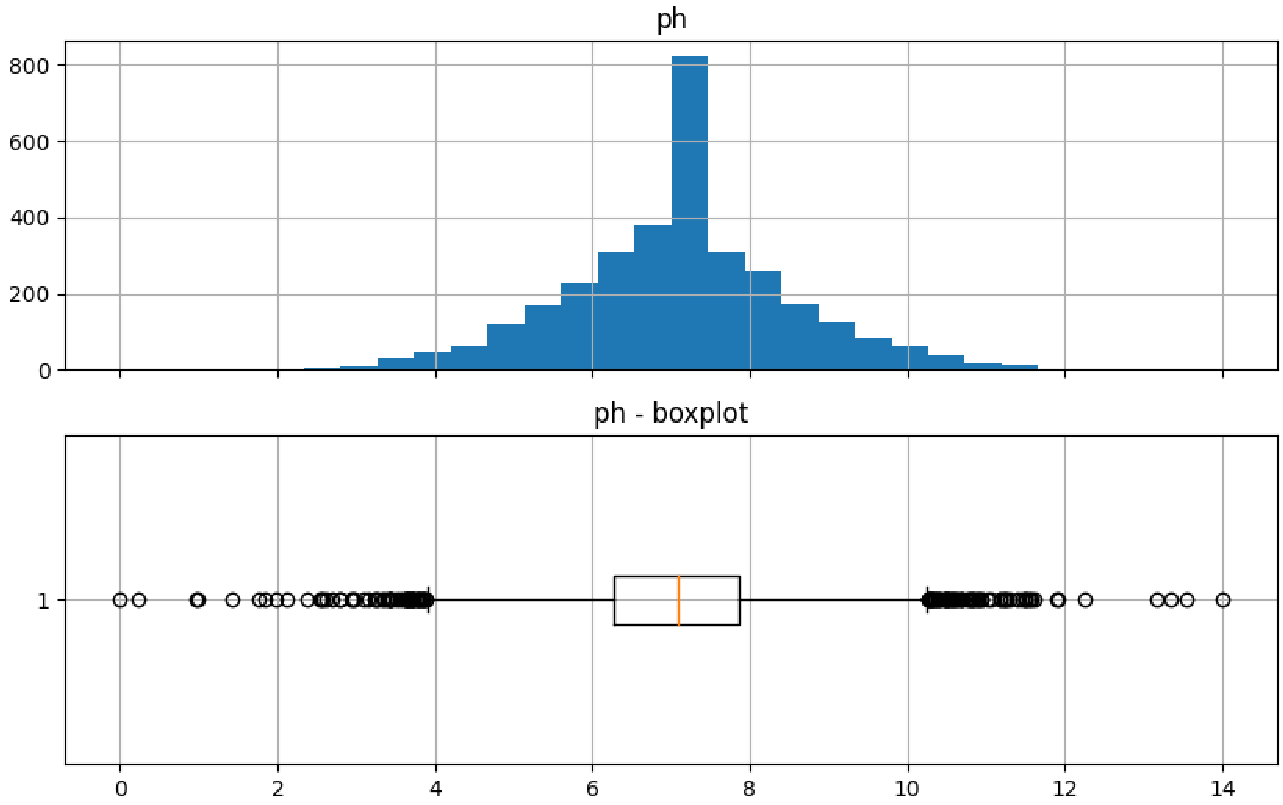


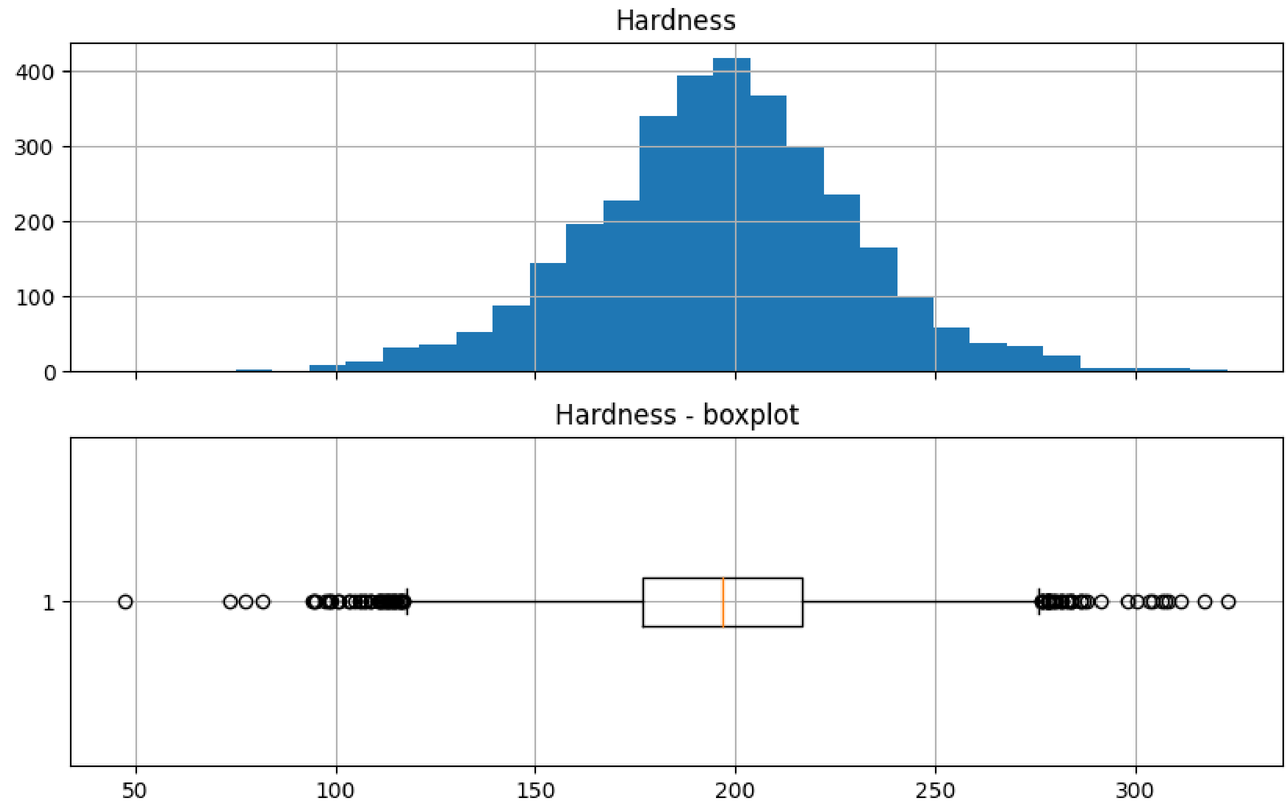
7.2.Boxplot

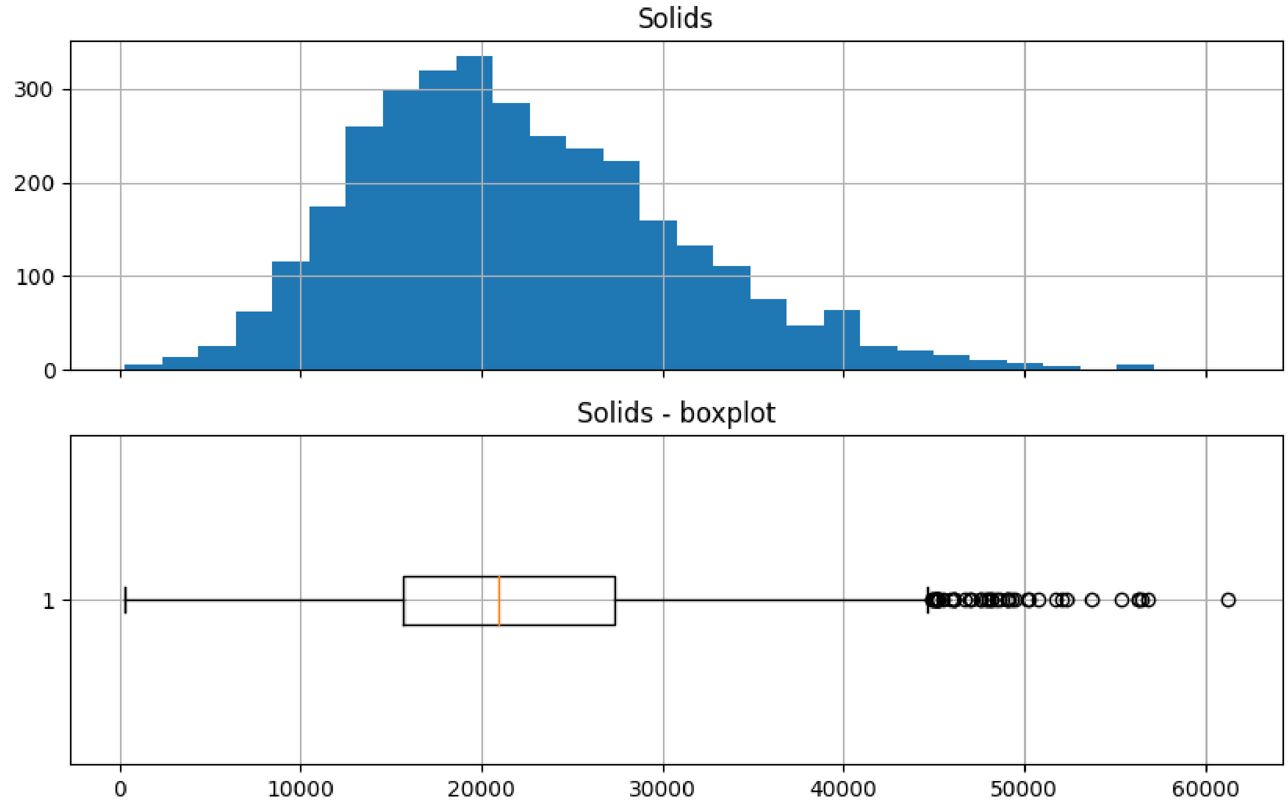
In [38]:

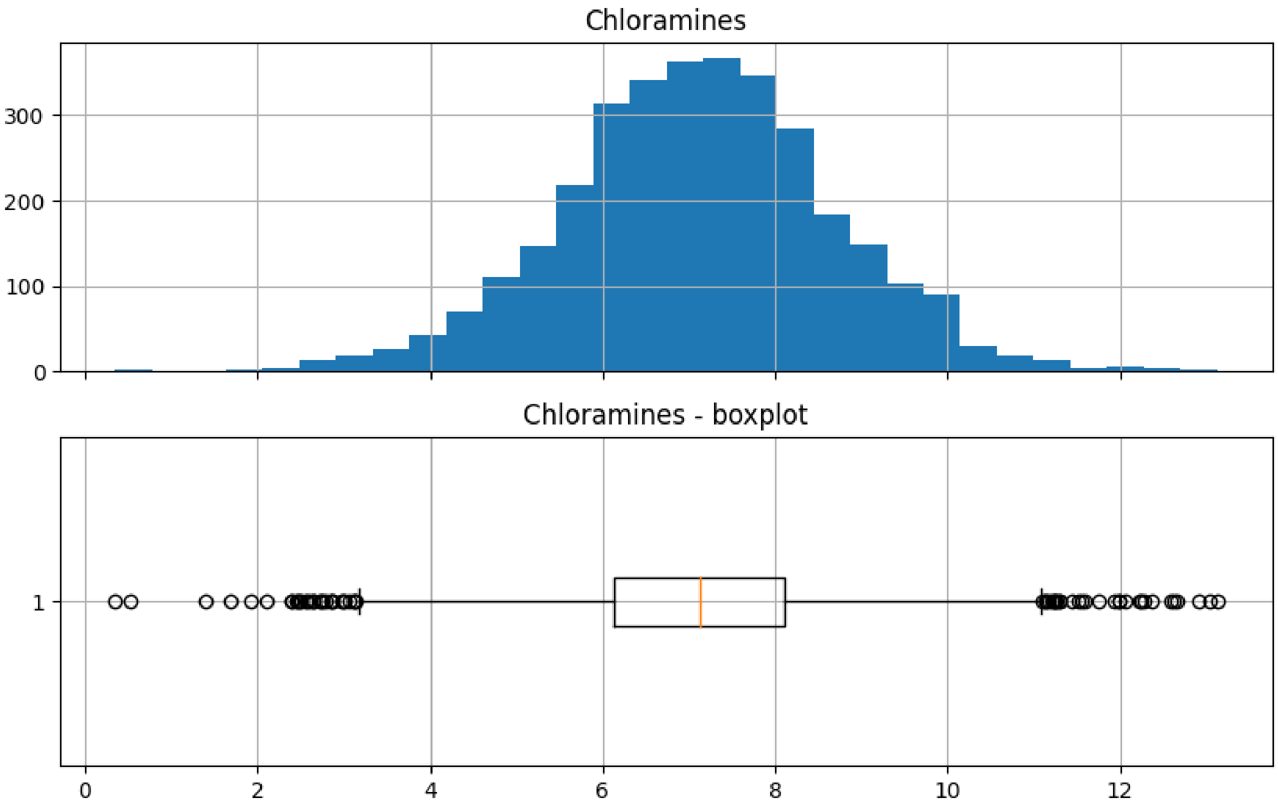
```python
features_num = ['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate',
                'Conductivity', 'Organic_carbon', 'Trihalomethanes',
                'Turbidity']
for f in features_num:
    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10,6), sharex=True)
    ax1.hist(data[f], bins=30)
    ax1.grid()
    ax1.set_title(f)
```

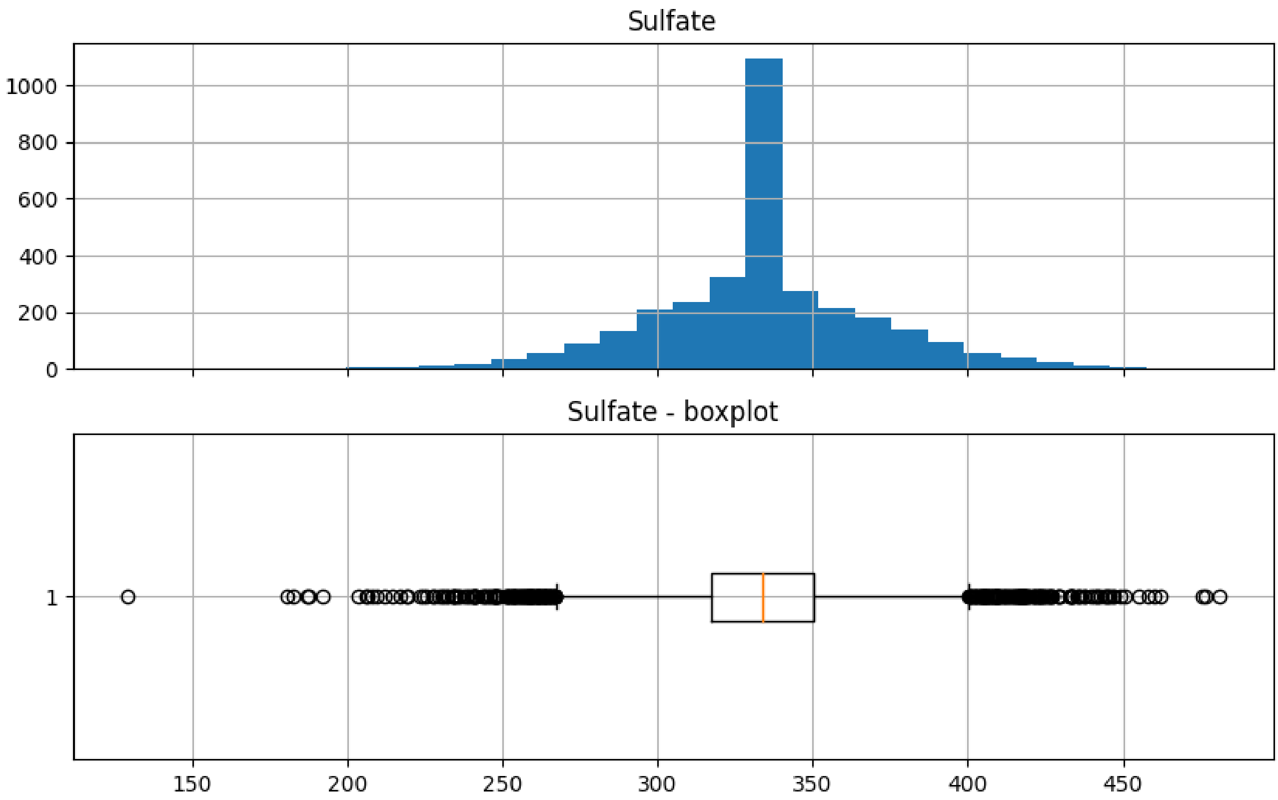
```python
# for boxplot we need to remove the NaNs first
feature_wo_nan = data[~np.isnan(data[f])][f]
ax2.boxplot(feature_wo_nan, vert=False)
ax2.grid()
ax2.set_title(f + ' - boxplot')
plt.show()
```

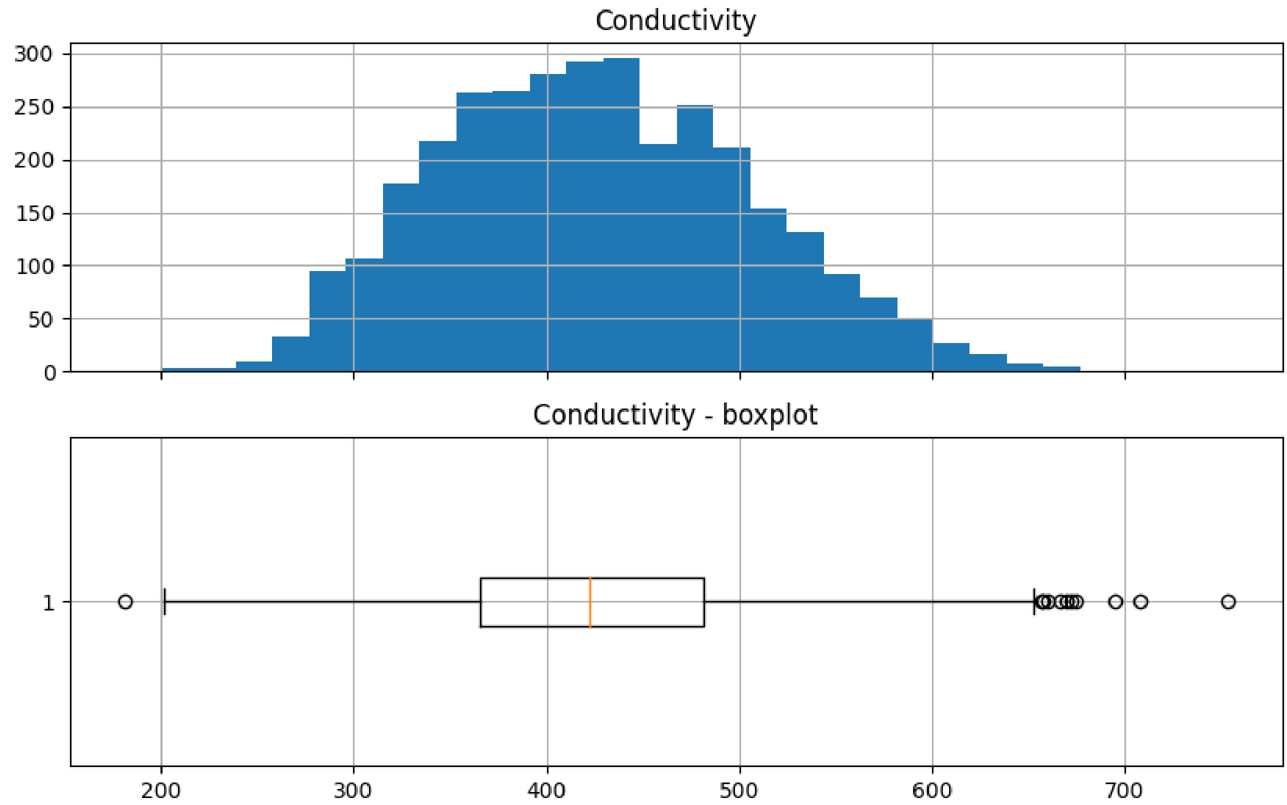


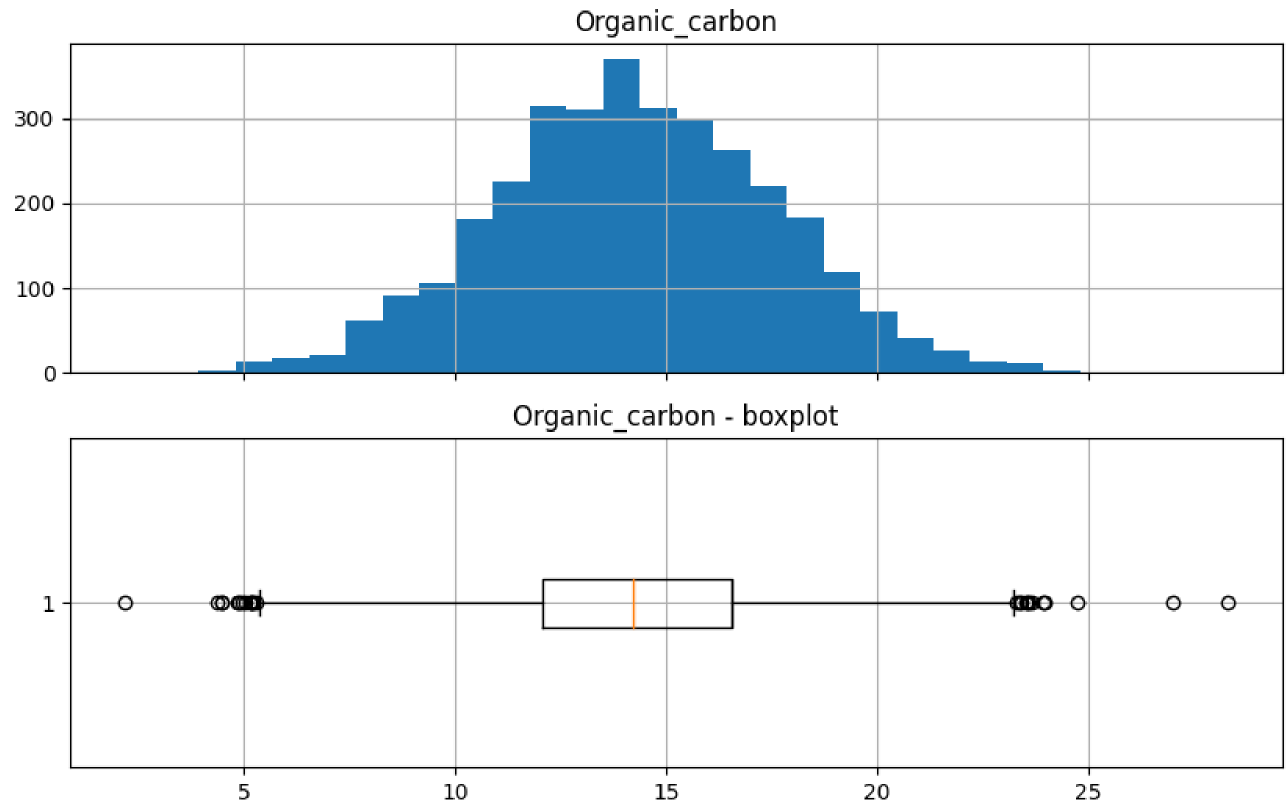

Hardness
400
300
200
100
0
Hardness - boxplot
1
50
100
150
200
250
300

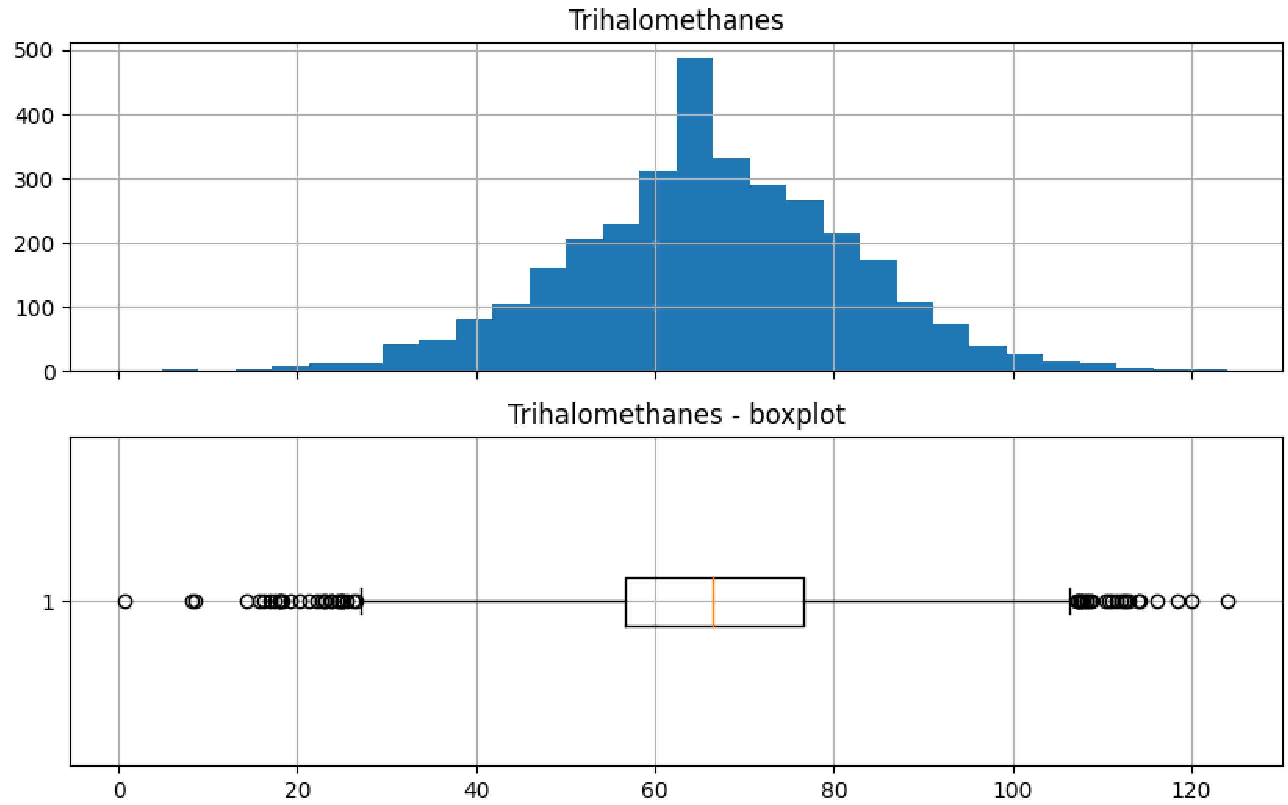

Solids
300
200
100
0
Solids - boxplot
1
0
10000
20000
30000
40000
50000
60000

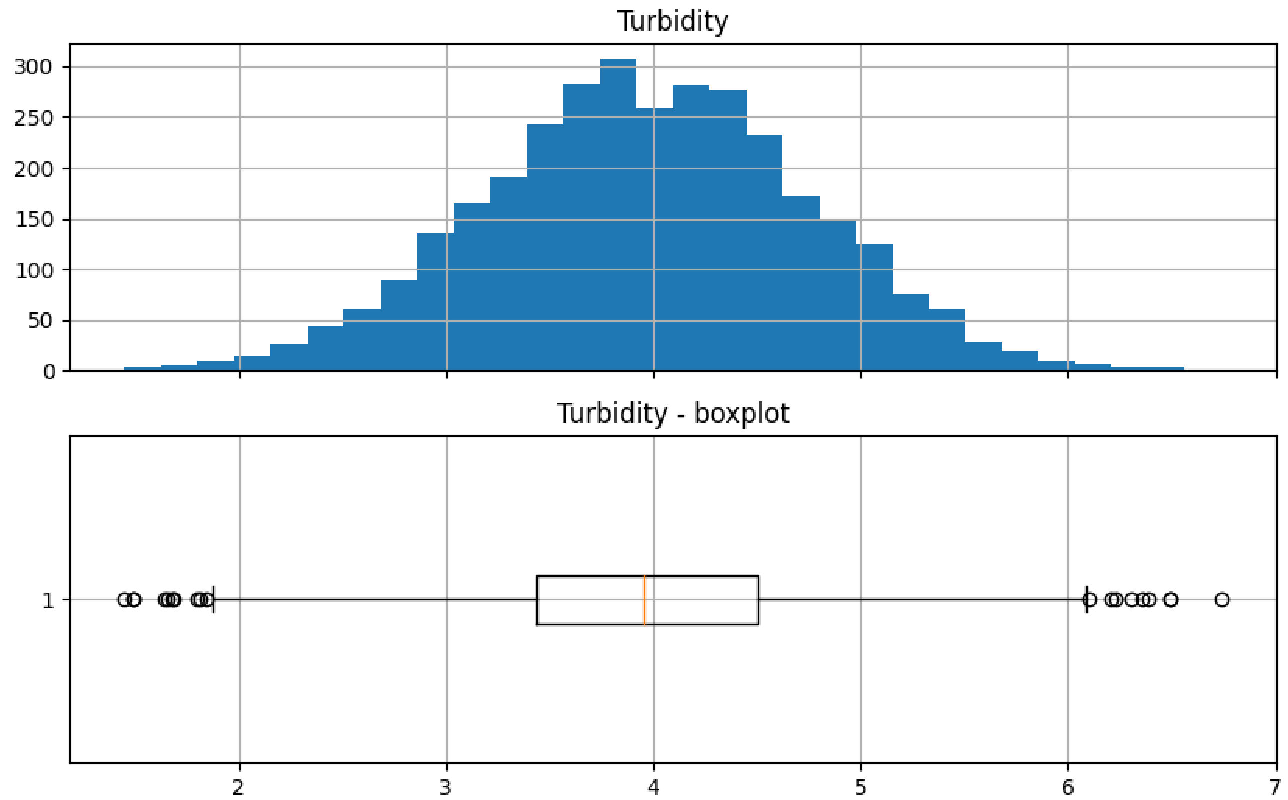

Chloramines
0
100
200
300
Chloramines - boxplot
1
0
2
4
6
8
10
12


Sulfate
0
200
400
600
800
1000
Sulfate - boxplot
1
150
200
250
300
350
400
450


Conductivity
300
250
200
150
100
50
0
Conductivity - boxplot
1
200
300
400
500
600
700


Organic_carbon
300
200
100
0
Organic_carbon - boxplot
1
5
10
15
20
25





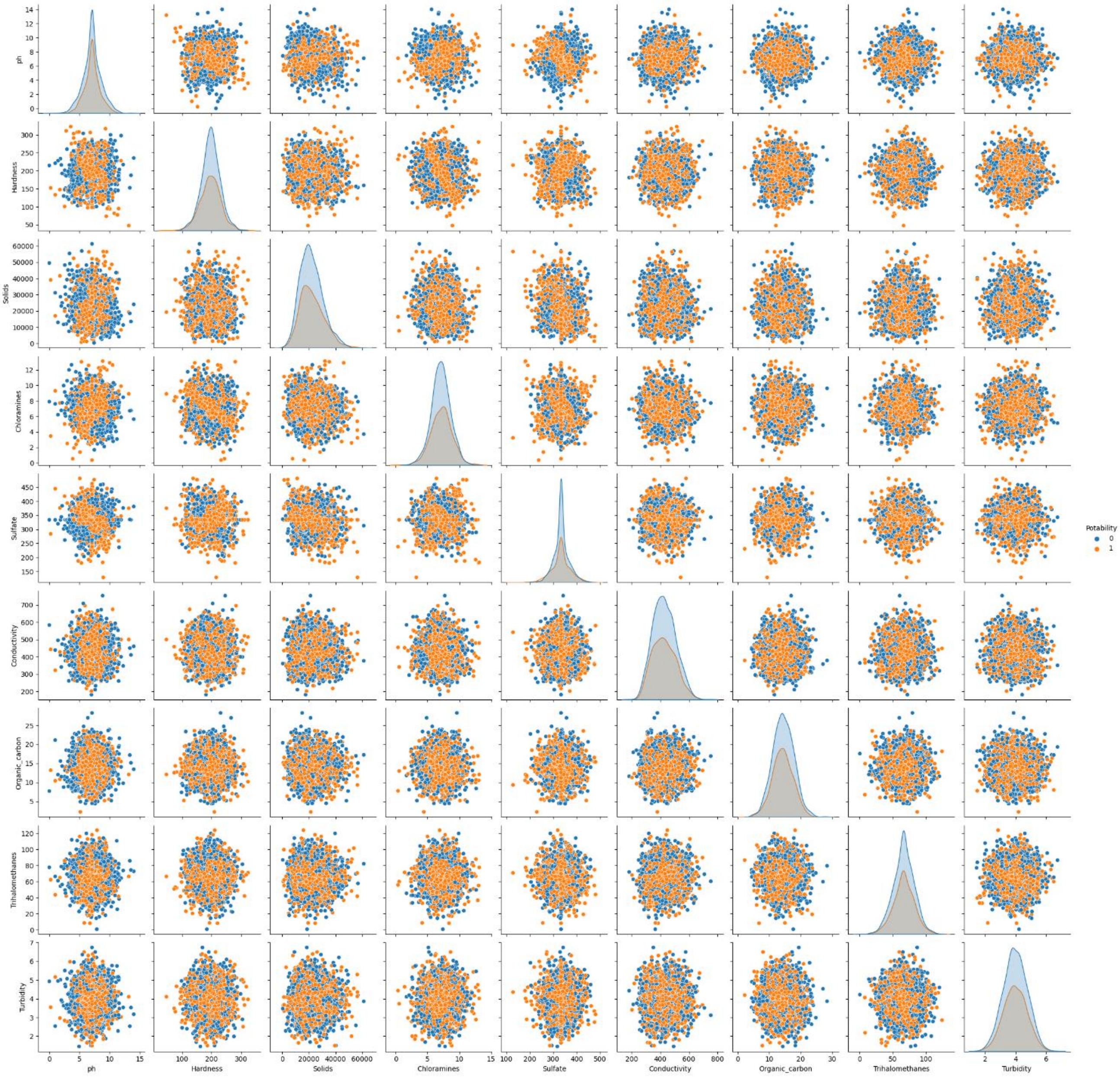
7.3.Scatter Plots.

In [39]:

```
sns.pairplot(data=data,hue="Potability")
```

Out[39]:

```
<seaborn.axisgrid.PairGrid at 0x7f615b2bff70>
```



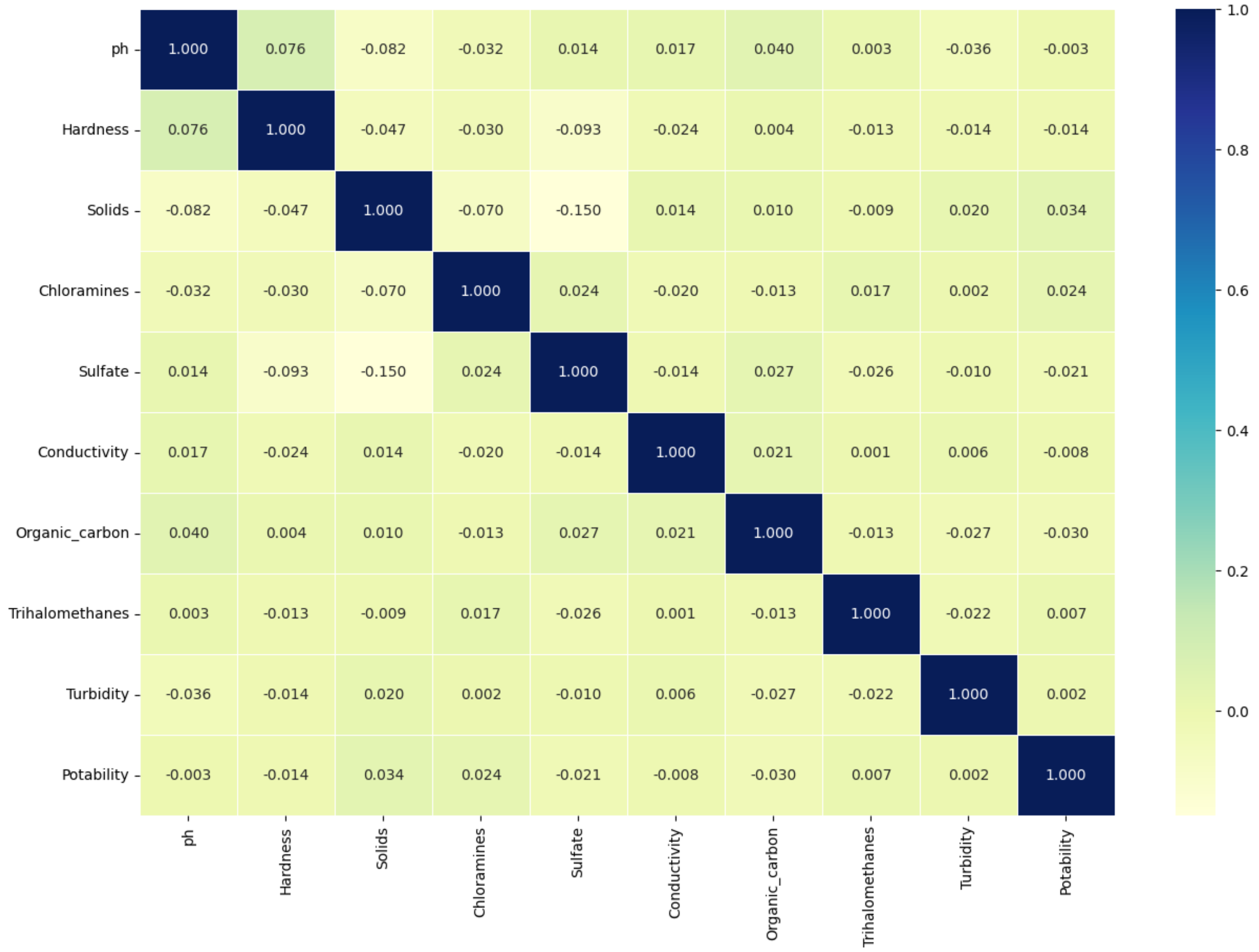
7.4.Correlation Heatmap.

In [41]:

```
corr_mat = data.corr()

fig, ax = plt.subplots(figsize=(15,10))
```

```
ax = sns.heatmap(corr_mat,annot=True,linewidths=0.5,fmt='.3f',cmap='YlGnBu')
```



	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1.000	0.076	-0.082	-0.032	0.014	0.017	0.040	0.003	-0.036	-0.003
Hardness	0.076	1.000	-0.047	-0.030	-0.093	-0.024	0.004	-0.013	-0.014	-0.014
Solids	-0.082	-0.047	1.000	-0.070	-0.150	0.014	0.010	-0.009	0.020	0.034
Chloramines	-0.032	-0.030	-0.070	1.000	0.024	-0.020	-0.013	0.017	0.002	0.024
Sulfate	0.014	-0.093	-0.150	0.024	1.000	-0.014	0.027	-0.026	-0.010	-0.021
Conductivity	0.017	-0.024	0.014	-0.020	-0.014	1.000	0.021	0.001	0.006	-0.008
Organic_carbon	0.040	0.004	0.010	-0.013	0.027	0.021	1.000	-0.013	-0.027	-0.030
Trihalomethanes	0.003	-0.013	-0.009	0.017	-0.026	0.001	-0.013	1.000	-0.022	0.007
Turbidity	-0.036	-0.014	0.020	0.002	-0.010	0.006	-0.027	-0.022	1.000	0.002
Potability	-0.003	-0.014	0.034	0.024	-0.021	-0.008	-0.030	0.007	0.002	1.000

8.Predictive Modelling for Potability

- A Random Forest Classifier was selected as the predictive model due to its ability to handle complex relationships within the data.
- The model was trained on the pre-processed and balanced dataset.
- Hyperparameter tuning was performed using Grid Search CV to optimize the model's performance.
- The accuracy of the best-tuned model was approximately 68.7%.

8.1.Data Splitting.

In [43]:

```
from sklearn.preprocessing import Normalizer, StandardScaler
sm = SMOTE(random_state=42)

X, y = data[data.columns[:-1]], data["Potability"]

X, y = sm.fit_resample(X, y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

8.2.Predictive Model.

In [44]:

```
models = [RandomForestClassifier()]
pipelines = {}
for model in models:
    model_name = str(model.__class__).split(".")[-1].split("'")[0]
    pipe = Pipeline([
        ("scaler", StandardScaler()),  # Preprocessing step
        ("classifier", model)  # Classifier step
    ])
    pipelines[model_name] = pipe



for name,pipe in pipelines.items():
    print(f"Training {name}")
    scores = cross_val_score(pipe, X_train, y_train, cv = 5, scoring = 
"accuracy")
    print(f"Mean Score {scores.mean()} -- Std {scores.std()} -- Min 
{scores.min()} -- Max {scores.max()}")
    pipe.fit(X_train, y_train)
```

```
Training RandomForestClassifier
Mean Score 0.6860133555926544 -- Std 0.020286333443847 59 -- Min 0.654424040066778
-- Max 0.715
```

8.3.Hyperparameter tuning the RandomForest model.

In [45]:

```
param_grid = {
    "criterion": ["gini", "entropy", "log_loss"],
    'n_estimators': [10, 20, 30, 40,50],
    'max_depth': [5, 10, 20, 30, 50],
}

rf_classifier = RandomForestClassifier(random_state = 42)
scorer = make_scorer(accuracy_score)
grid_search = GridSearchCV(
    rf_classifier, param_grid, scoring=scorer, cv=5, verbose = 1
)
grid_search.fit(X_train, y_train)
best_rf = grid_search.best_estimator_

best_predictions = best_rf.predict(X_test)
best_accuracy = accuracy_score(best_predictions, y_test)

print("Best Accuracy Score:", best_accuracy)
```

```
Fitting 5 folds for each of 75 candidates, totalling 375 fits
Best Accuracy Score: 0.6766766766766766
```

8.4.RandomForest model Accuracy Score.

In [46]:

```
accuracy_score(best_rf.predict(X_test),y_test)
```

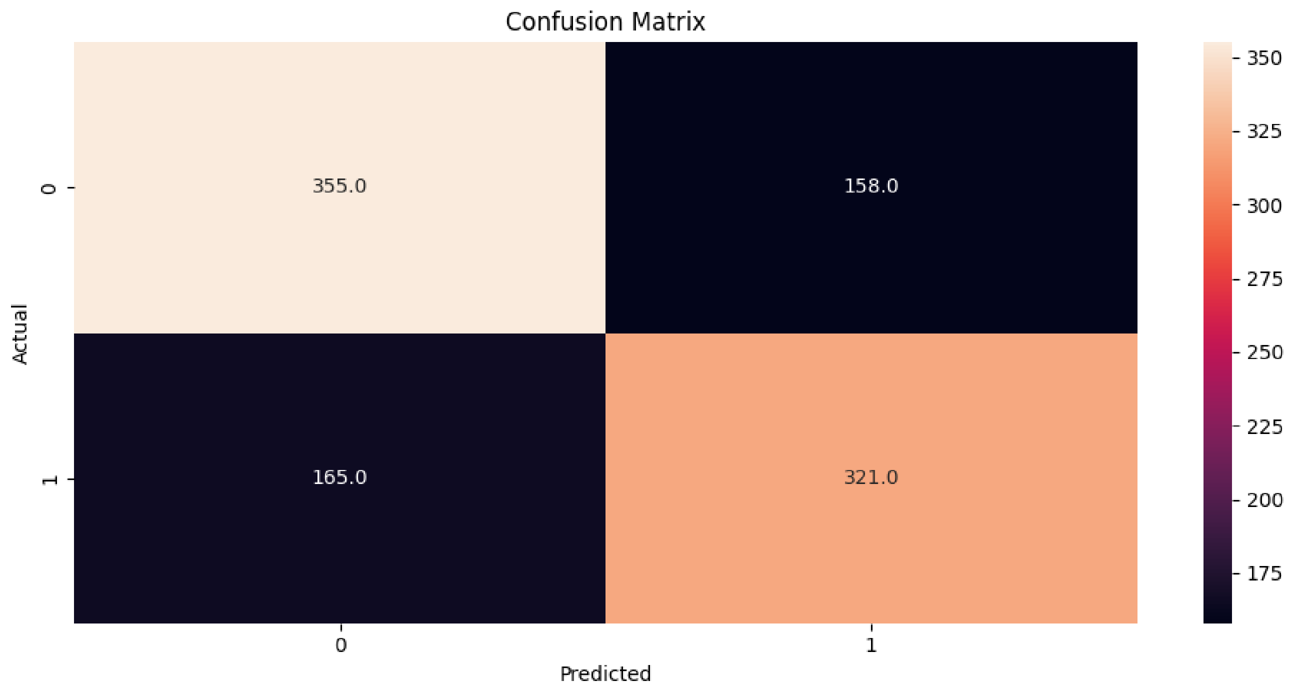
Out[46]:

```
0.6766766766766766
```

8.5.Confusion Matrix

In [47]:

```
plt.figure(figsize=(10,5))
sns.heatmap(confusion_matrix(best_rf.predict(X_test),y_test), annot = True,fmt='.1f')
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.title("Confusion Matrix")
plt.tight_layout()
```

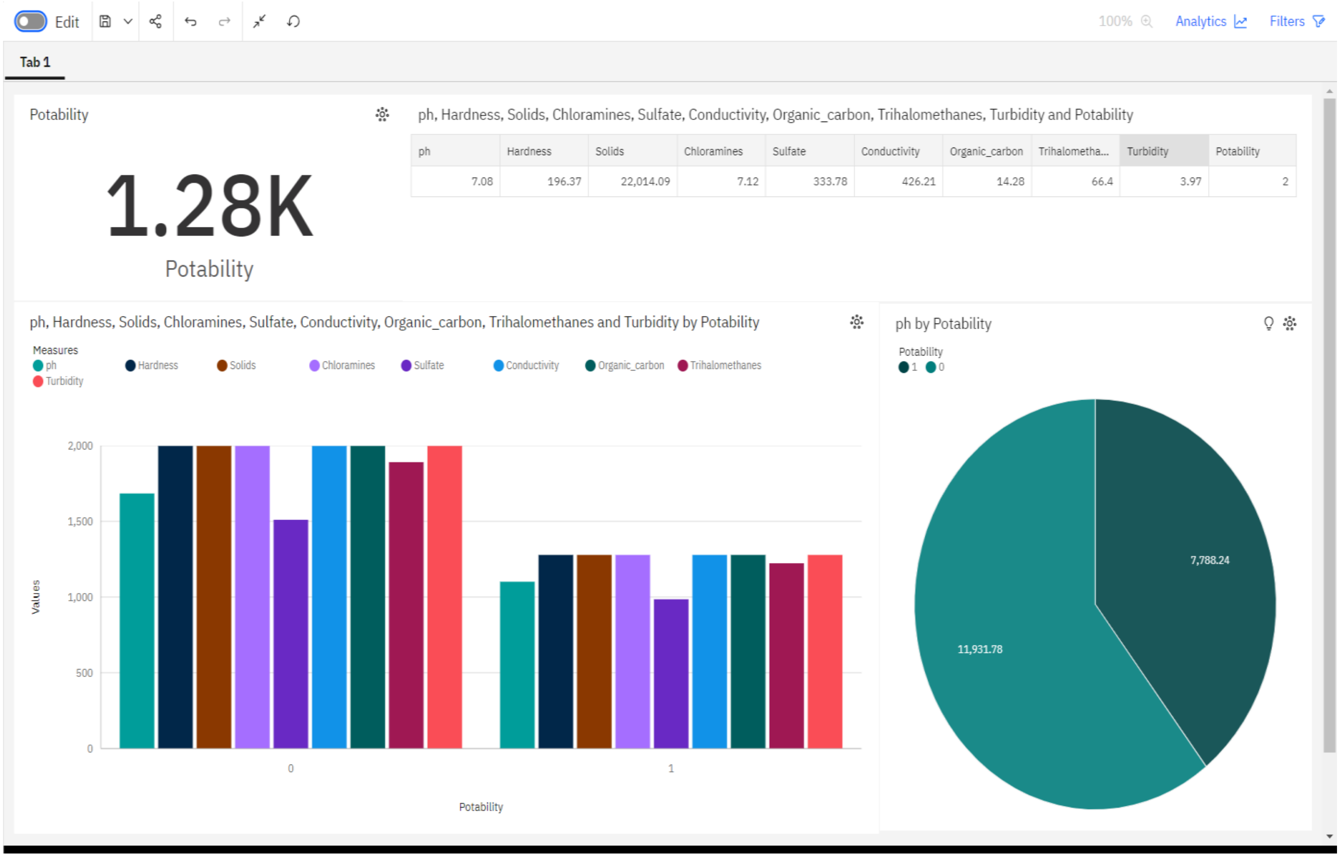


9.Analysis Insights:

- The analysis provides a comprehensive understanding of the various factors influencing water quality, such as pH, hardness, chloramines, and organic carbon. By visualizing the distribution and relationships between these parameters, water quality can be more effectively assessed.
- The developed Random Forest Classifier serves as a reliable tool to determine the potability of water samples. With an accuracy of around 68.7%, the model can assist in categorizing water samples as potable or non-potable, aiding regulatory bodies and water treatment facilities in decision-making processes.

- Through EDA and the correlation heatmap, it was observed that certain parameters, such as chloramines and sulfate, significantly influence water potability. These insights can guide further research and regulatory efforts, emphasizing the importance of monitoring and controlling these specific parameters.
- The analysis flagged outliers in various water quality features. Detecting these outliers is crucial for understanding abnormal patterns in water samples, potentially indicating contamination or issues in the water supply system.
- The analysis equips decision-makers with a data-driven approach to assess water quality. By leveraging the predictive model, authorities can make informed decisions regarding water treatment processes, ensuring the delivery of safe and potable water to the public.

10.FINAL IBM COGNOS REPORT:



11.Conclusion:

In this project, we conducted an in-depth analysis of water quality data and built a predictive model to determine water potability. The dataset contained various features related to water quality, such as pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity. We began by handling missing values, filling them with the mean of respective columns. Exploratory data analysis (EDA) involved visualizations like histograms, boxplots, scatter plots, and correlation heatmaps, giving us insights into feature distributions and relationships.

To address class imbalance, we employed the SMOTE technique, creating a balanced dataset. We then trained a Random Forest Classifier and performed hyperparameter tuning using Grid Search. The best model achieved an accuracy of approximately 67.7% on the test data.

In summary, the analysis underscores the complexity of water quality dynamics. While the predictive model provides a reasonable accuracy, further exploration could involve more advanced techniques, additional features, or domain-specific knowledge integration for improved predictions. Additionally, ongoing data collection and refinement of models will be vital for enhancing the accuracy and reliability of water potability predictions, crucial for both public health and environmental sustainability.