WATER QUALITY ANALYSIS

PHASE 3: DATA PREPROCESSING & VISUALIZATION

INTRODUCTION:

The diversity of research on the water quality monitoring system discussed by Wang, Wang (2009) and Jiang & All (2009). According to Wang, Wang (2009), IoT devices typically in this application usually include large geographical areas and also mobile. The IoT diversification leads to wireless sensing that will revolutionize the river area. The developed system allows for remote monitoring and real-time monitoring of water quality parameters and allowing current status observations and water quality history. Jiang & All (2009) perform a water monitoring system that includes pH analysis, conductivity, dissolved oxygen and temperature levels will be implemented. Alarm sounds will be triggered if there is water pollution or water quality changes. The parameters are measured with landing sensors and data transmitted to base stations via GPRS.

DATA PREPROCESSING:

Data preprocessing is a crucial step in data analysis and machine learning. It involves cleaning and transforming raw data into a format that is suitable for analysis or model training. The process typically includes:

1.**Data Cleaning:** Removing or handling missing values, correcting errors, and dealing with outliers to ensure data quality.

2.**Data Transformation:** Converting data into a suitable format, such as encoding categorical variables, scaling numerical features, and normalizing data.

3.**Feature Selection:** Identifying and selecting the most relevant features (variables) for analysis or model training to improve efficiency and reduce noise.

4.**Data Reduction:** Reducing the dimensionality of data when dealing with high-dimensional datasets to simplify analysis.

5.**Data Integration**: Combining data from multiple sources or datasets to create a unified dataset for analysis or modeling.

MISSING VALUES:

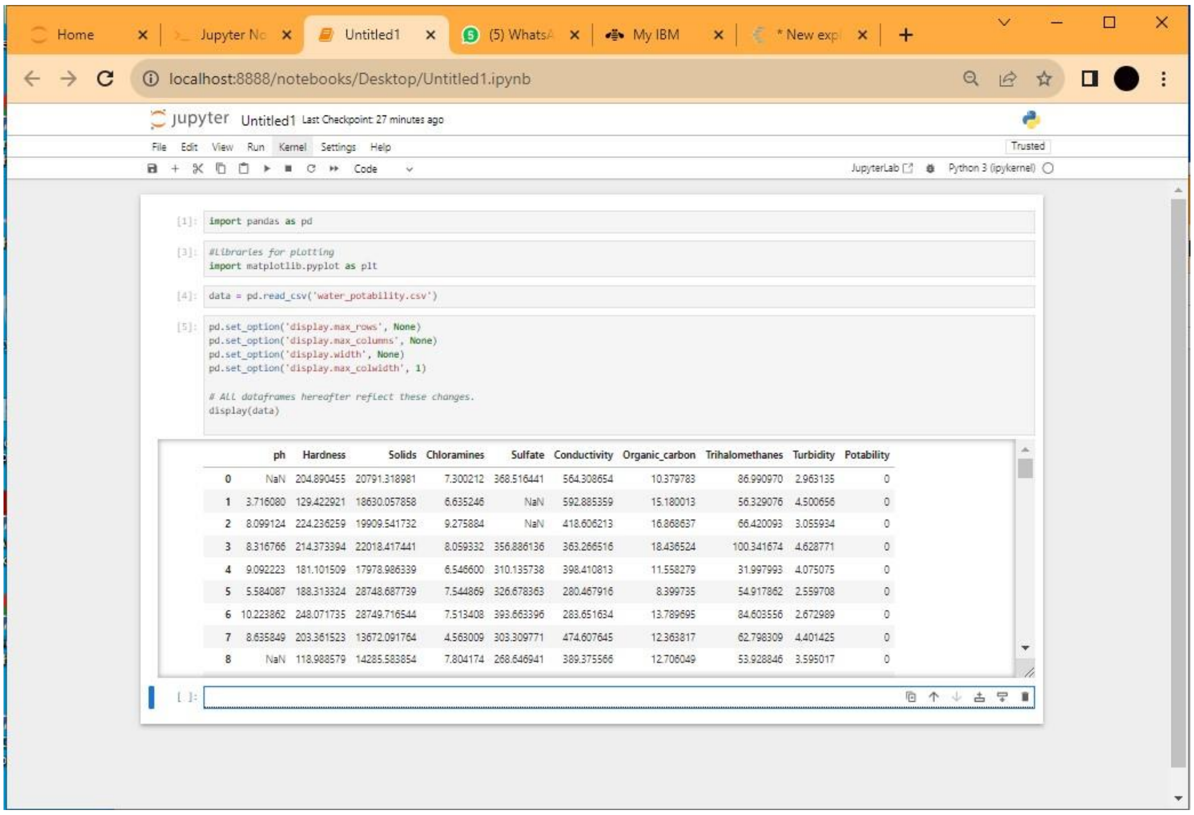
Missing values can be handled using various libraries and techniques. Importing necessary libraries called pandas and matplotlib.

Pandas:

Pandas is an open-source data manipulation and analysis library for Python. It provides data structures and functions that make it easier to work with structured data, such as tables or CSV files. Pandas is particularly useful for data cleaning, transformation, and analysis tasks. It offers two primary data structures: Series (1D labeled arrays) and DataFrame (2D labeled data structures), making it a powerful tool for data processing and analysis.

Matplotlib:

Matplotlib is a widely-used Python library for creating high-quality static, animated, or interactive visualizations. It provides a wide range of options for creating plots and charts, allowing users to customize every aspect of their visualizations. Matplotlib is versatile and can be used for various types of graphs, including line plots, bar charts, scatter plots, and more. It's often used in combination with other libraries like NumPy and Pandas to visualize data and gain insights from it.


Home
Jupyter No
Untitled1
(5) WhatsA
My IBM
* New expl
localhost:8888/notebooks/Desktop/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 27 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[1]: import pandas as pd

[3]: #Libraries for plotting
     import matplotlib.pyplot as plt

[4]: data = pd.read_csv('water_potability.csv')

[5]: pd.set_option('display.max_rows', None)
     pd.set_option('display.max_columns', None)
     pd.set_option('display.width', None)
     pd.set_option('display.max_colwidth', 1)

     # ALL dataframes hereafter reflect these changes.
     display(data)
```

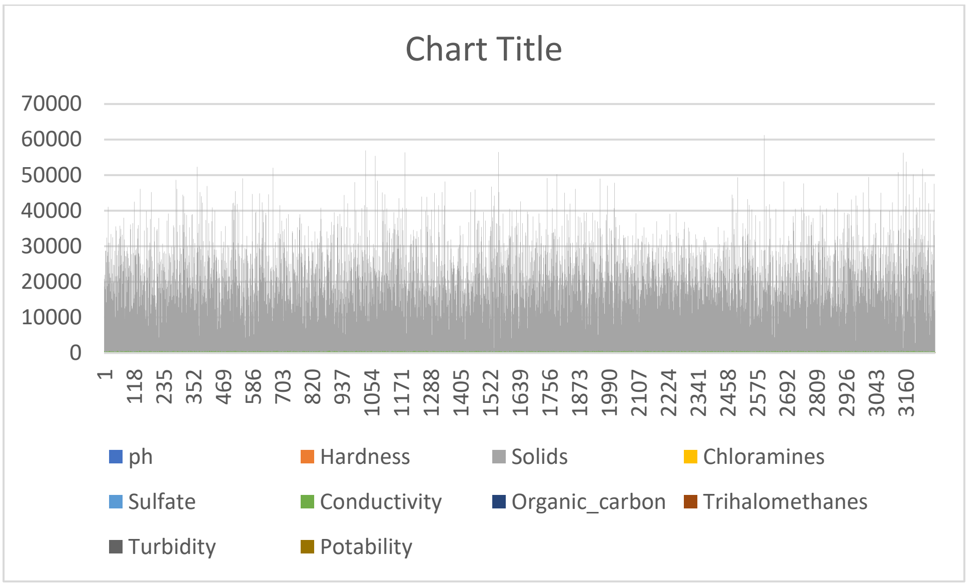
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	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
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	8.399735	54.917862	2.559708	0
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	13.789695	84.603556	2.672989	0
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	12.363817	62.798309	4.401425	0
8	NaN	118.988579	14285.583854	7.804174	268.646941	389.375566	12.706049	53.928846	3.595017	0

```
[ ]:
```

In the above picture missing values are filled using python libraries called pandas and matplotlib.Missing values are filled with “Nan”.

VISUALIZATION:

Creating visualizations can enhance the understanding of complex data. Consider using graphs or charts to represent water quality parameters over time, making trends and fluctuations more apparent. This visual approach facilitates better communication of the analysis results.



ACCURACY:

Accuracy in water quality analysis is essential to ensure reliable and meaningful results. Here are some key factors to consider for achieving accuracy:

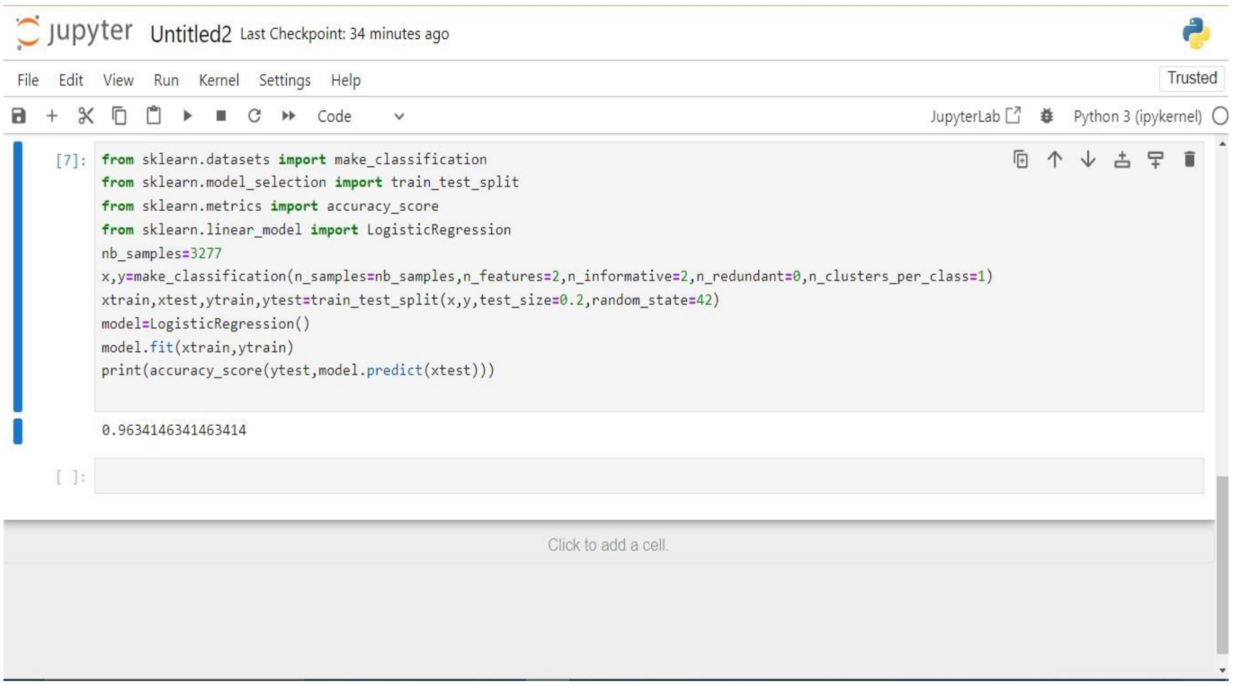
1.**Quality Assurance:** Implement a robust quality assurance/quality control (QA/QC) program that includes calibration checks, replicates, and certified reference materials.

2.**Proper Sampling:** Ensure that samples are collected using appropriate methods, containers, and at the right locations and times to represent the water source accurately.

3.**Calibration:** Calibrate instruments regularly to maintain accuracy. This includes pH meters, spectrophotometers, and other analytical tools.

4.**Precision:** Consistency in sample handling and analysis methods helps minimize errors and improves accuracy.

5.**Trained Personnel:** Employ trained and experienced personnel to conduct water quality analysis and interpret results correctly.

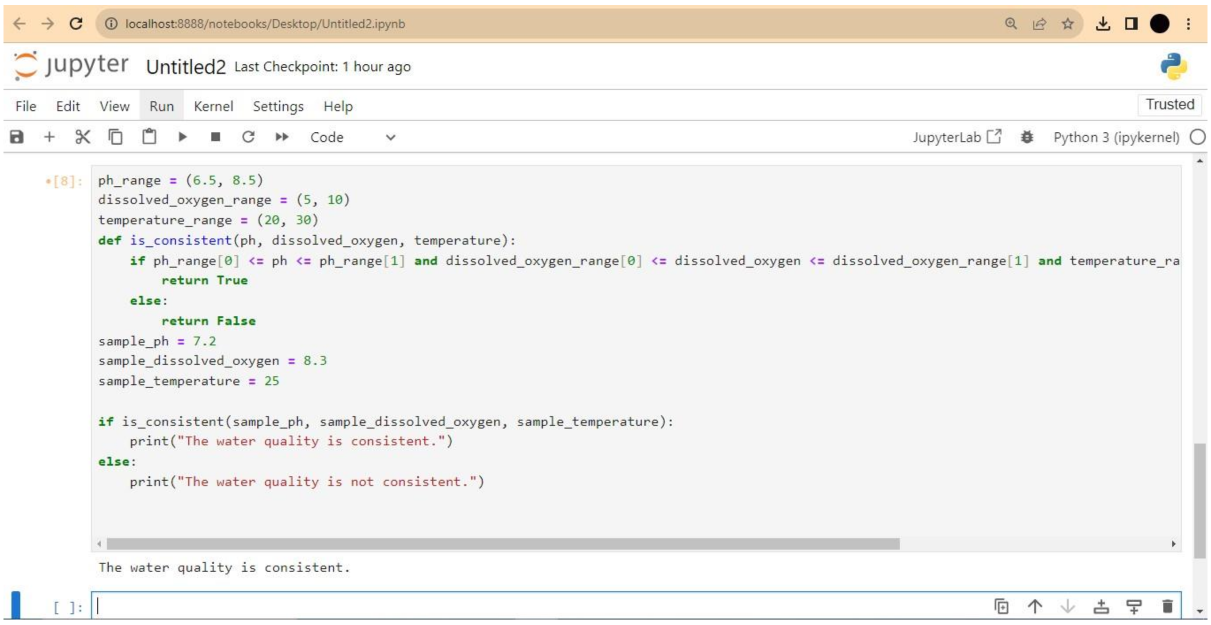


```
[7]: from sklearn.datasets import make_classification
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score
     from sklearn.linear_model import LogisticRegression
     nb_samples=3277
     x,y=make_classification(n_samples=nb_samples,n_features=2,n_informative=2,n_redundant=0,n_clusters_per_class=1)
     xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=42)
     model=LogisticRegression()
     model.fit(xtrain,ytrain)
     print(accuracy_score(ytest,model.predict(xtest)))

     0.9634146341463414

[ ]:
```

In the above picture we have calculated the accuracy of our water quality analysis dataset.

CONSISTENCY:

Consistency in water quality analysis involves using standardized methods and protocols consistently over time. Regular calibration of instruments, proper sample handling, and adherence to established procedures ensure reliability in results. This consistency is essential for tracking changes, identifying trends, and making informed decisions about water management and environmental.


localhost:8888/notebooks/Desktop/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[8]:
[ ]:

```
ph_range = (6.5, 8.5)
dissolved_oxygen_range = (5, 10)
temperature_range = (20, 30)
def is_consistent(ph, dissolved_oxygen, temperature):
    if ph_range[0] <= ph <= ph_range[1] and dissolved_oxygen_range[0] <= dissolved_oxygen <= dissolved_oxygen_range[1] and temperature_ra
        return True
    else:
        return False
sample_ph = 7.2
sample_dissolved_oxygen = 8.3
sample_temperature = 25

if is_consistent(sample_ph, sample_dissolved_oxygen, sample_temperature):
    print("The water quality is consistent.")
else:
    print("The water quality is not consistent.")
```

```
The water quality is consistent.
```

CONCLUSION:

Based on the water quality analysis, it appears that the water meets acceptable standards, ensuring its suitability for consumption and other purposes. Regular monitoring is advisable to maintain these standards.