

OUR TEAM

TEAM LEADER :

RICHARD NICHOLAS M

TEAM MEMBERS :

SANTHOSH KUMAR P
SATHYAMOORTHY A
VINOOTH S
YOKESWARAN K
VASUDEVAN K



WATER QUALITY ANALYSIS

ANOMALY DETECTION TECHNIQUES

Introduction

Water is one of our most vital resources, and ensuring its quality is paramount for human health, environmental sustainability, and industrial processes. The analysis of water quality parameters is crucial to identify unusual patterns or anomalies that could signal contamination, environmental changes, or infrastructure issues.

This presentation explores the use of anomaly detection techniques in the field of water quality analysis. Anomaly detection is a powerful tool for uncovering unusual or unexpected patterns in data, which can often signify issues or changes that require immediate attention.



Phase 2 : Innovation

Consider Exploring anomaly detection techniques to identify unusual patterns in Water Quality Parameters



Data Collection

Water Quality Analysis is done by using the Dataset of “Water_Potability” provided by the dataset site www.Kaggle.com

Dataset Link :

<https://www.kaggle.com/datasets/adityakadiwal/water-potability>



Dataset Observation

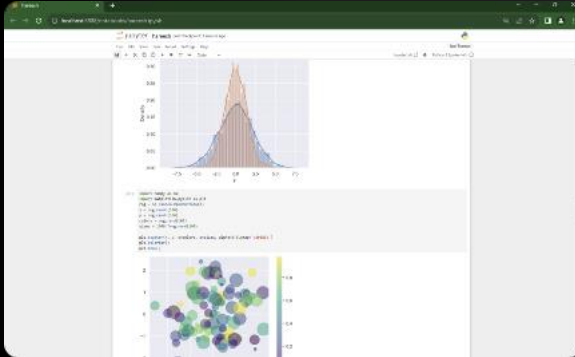
The **water_potability.csv** file contains water quality metrics for **3276** different water bodies.

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.

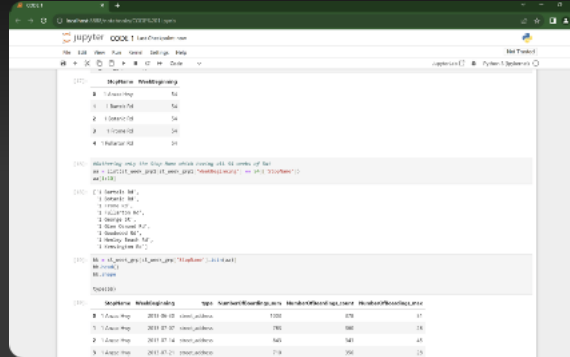


Why Jupyter Notebook ?



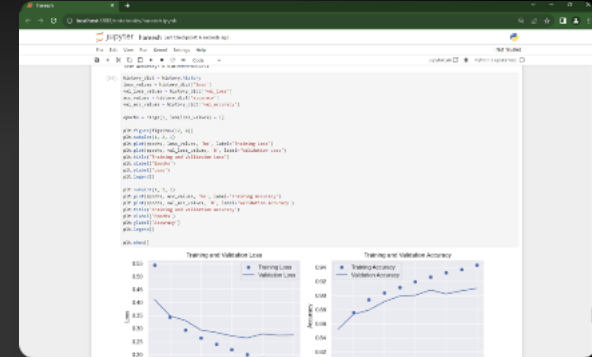
Visualizing Data in Jupyter Notebook

Demonstrating the power of data visualization using Jupyter Notebook's interactive capabilities.



Executing Code in Jupyter Notebook

Witness the step-by-step execution of code snippets within the Jupyter Notebook environment.

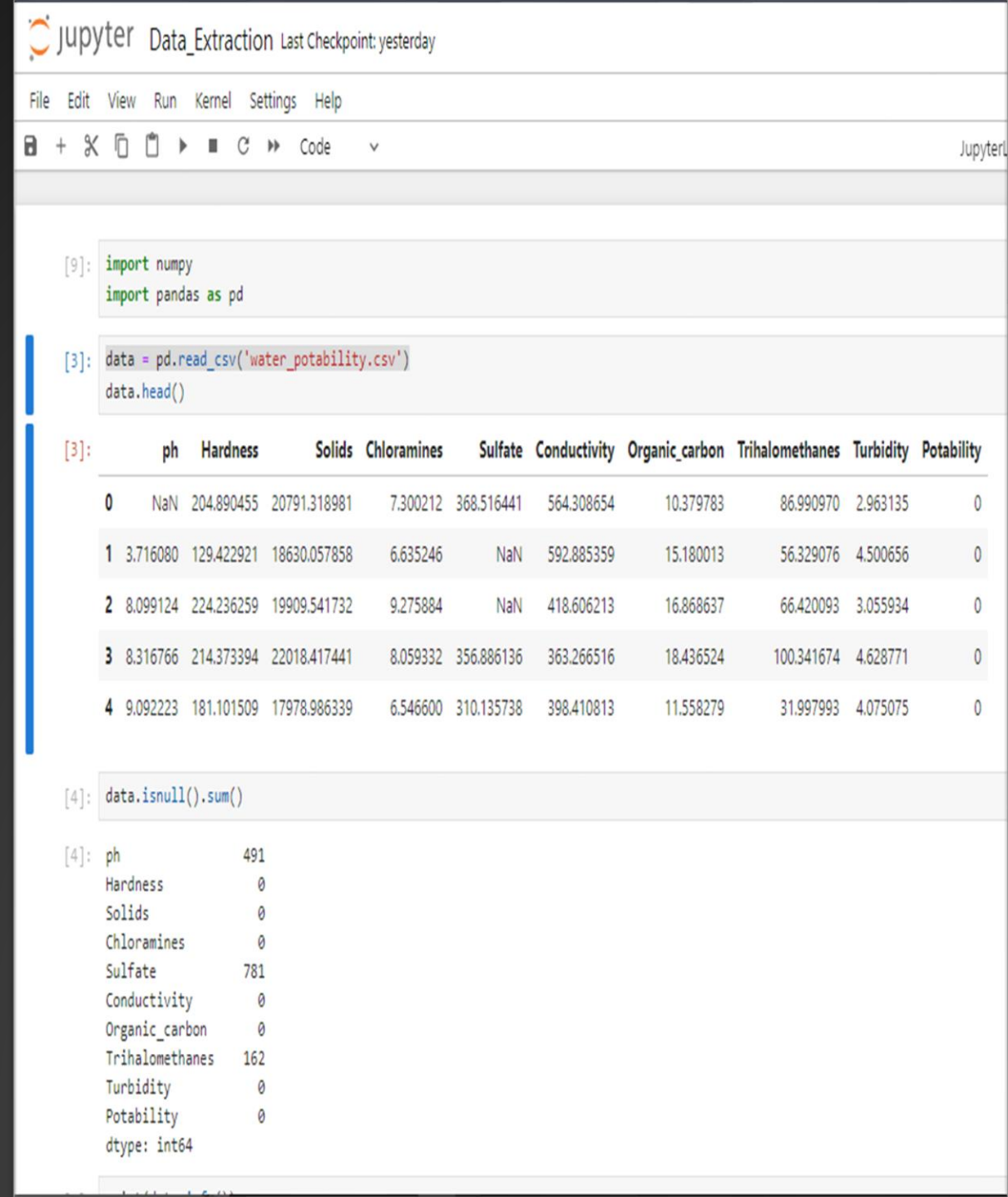


Machine Learning Algorithms in Jupyter Notebook

Showcasing the implementation of KNN, CNN, and Gradient Descent algorithms in Jupyter Notebook.

Data Extraction With Jupyter Notebook

Jupyter Notebook is widely used to extract data and to work with datasets due to its high flexibility and user-friendly interface. For this project, we utilized Jupyter Notebook to extract data from CSV files. Specifically, we extracted data from dataset provided by the Kaggle **'water_potability.csv'**. By leveraging Pandas, a popular library of Python, we were able to efficiently extract and manipulate the data.



The screenshot shows a Jupyter Notebook window titled "Data_Extraction" with a "Last Checkpoint: yesterday" status. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for saving, opening, and running code. The notebook contains four code cells:

- Cell [9]:

```
import numpy
import pandas as pd
```
- Cell [3]:

```
data = pd.read_csv('water_potability.csv')
data.head()
```
- Cell [3]: Displays the first five rows of the 'water_potability.csv' dataset as a table.
- Cell [4]:

```
data.isnull().sum()
```
- Cell [4]: Displays the count of missing values for each column.

	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

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

Visualization with NumPy, Pandas and Matplotlib.

In this section, we will demonstrate how NumPy and Pandas Data Frame were used for effective data visualization. The visualization techniques used include graph plotting, histograms, and scatter plots.



Introduction



NumPy

NumPy, which stands for "Numerical Python," provides support for working with large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is a core library in the Python data.



Pandas DataFrame

A pandas DataFrame is a two-dimensional, labelled data structure commonly used in data analysis and manipulation within the Python programming language. It is one of the fundamental data structures provided by the pandas library.



Matplotlib

Matplotlib is a widely-used Python library for creating static and interactive visualizations and plots. It is a powerful tool for data visualization and is particularly popular in the fields of data analysis, scientific computing, and machine learning.

```

Jupyter Notebook
localhost:8889/notebooks/Downloads/numpy.ipynb

jupyter numpy Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help

In [18]: import numpy as np
a = np.array(42)
b = np.array([1,2,3,4,5])
c = np.array([[1,2,3],[4,5,6]])
d = np.array([[[1,2,3],[4,5,6],[1,2,3],[4,5,6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)

0
1
2
3

In [20]: import numpy as np
arr = np.array([1,2,3,4], ndmin=5)

print(arr)
print('number of dimensions :',arr.ndim)

[[[[[1 2 3 4]]]])
number of dimensions : 5

In [22]: import numpy as np
arr = np.array([1,2,3,4])
print(arr[0])

1

In [23]: import numpy as np
arr = np.array([1,2,3,4])
print(arr[1])

2

```

NumPy

- **Data Loading:** NumPy aids in reading and handling datasets from various file formats.
- **Data Preparation:** It assists in cleaning, normalizing, and converting data types.
- **Data Exploration:** NumPy calculates statistics, revealing insights into transportation data.
- **Numerical Operations:** Efficient element-wise operations, aggregations, and calculations are easily performed.
- The above operations are processed in the adjacent screenshot.

Pandas

- **Data Import:** Pandas reads transportation datasets from various formats such as CSV, Excel, SQL databases, or web sources, into DataFrame structures.
- **Data Exploration:** Pandas provides tools for exploring and summarizing data, including methods for checking basic statistics, identifying missing values, and understanding data distributions.
- **Data Manipulation:** Pandas filters, selects, merges, and reshapes data efficiently.
- **Time Series Analysis:** Supports analysis of time-related trends in transportation data.

jupyter Data_Extraction Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

+ ✂ 📄 📌 ▶ ■ 🔁 ⏪ Code ▾ Jupyter

```
[9]: import numpy
import pandas as pd

[3]: data = pd.read_csv('water_potability.csv')
data.head()
```

	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

```
[10]: data.shape

[10]: (3276, 10)

[4]: data.isnull().sum()
```

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

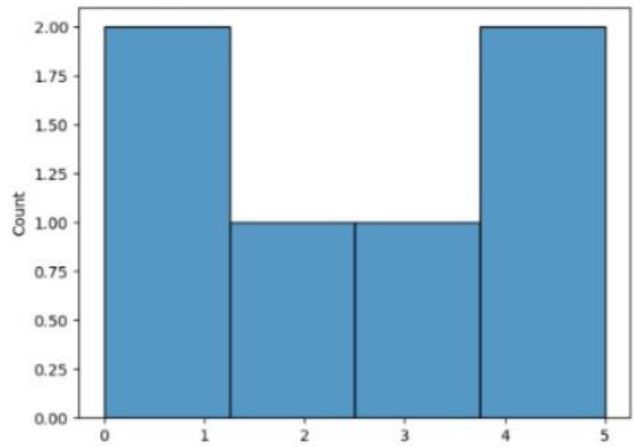
```
[5]: print(data.info())
```



```
py.ipynb
ter numpy Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code
```

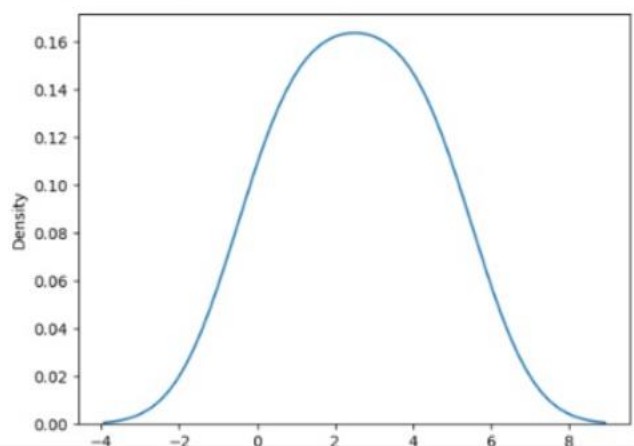
```
36]: import matplotlib.pyplot as plt
import seaborn as sns
sns.histplot([0,1,2,3,4,5])
plt.show()

36]: <Axes: ylabel='Count'>
```



```
37]: import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot([0,1,2,3,4,5], hist=False)
plt.show()

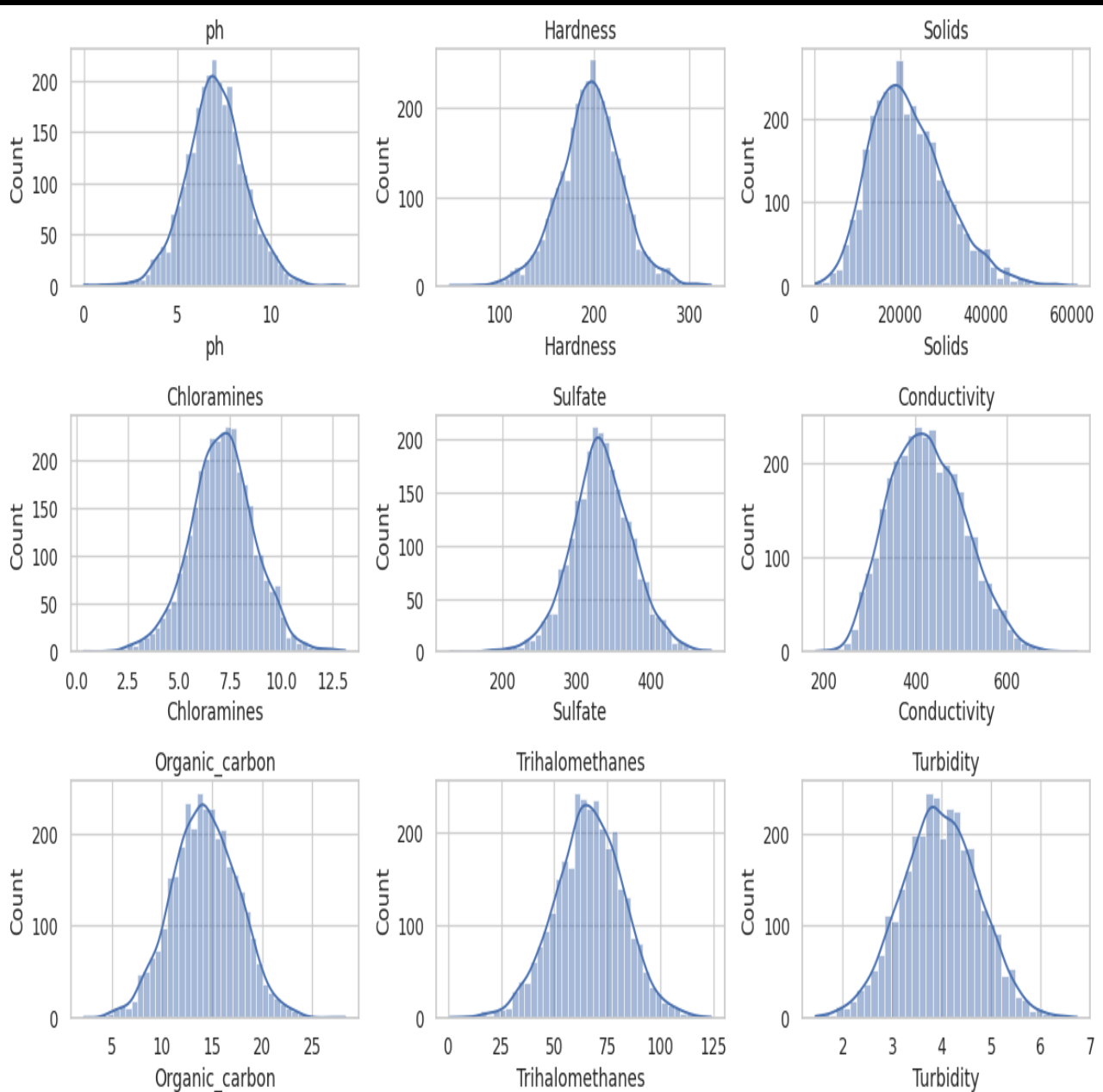
37]: <Axes: ylabel='Density'>
```



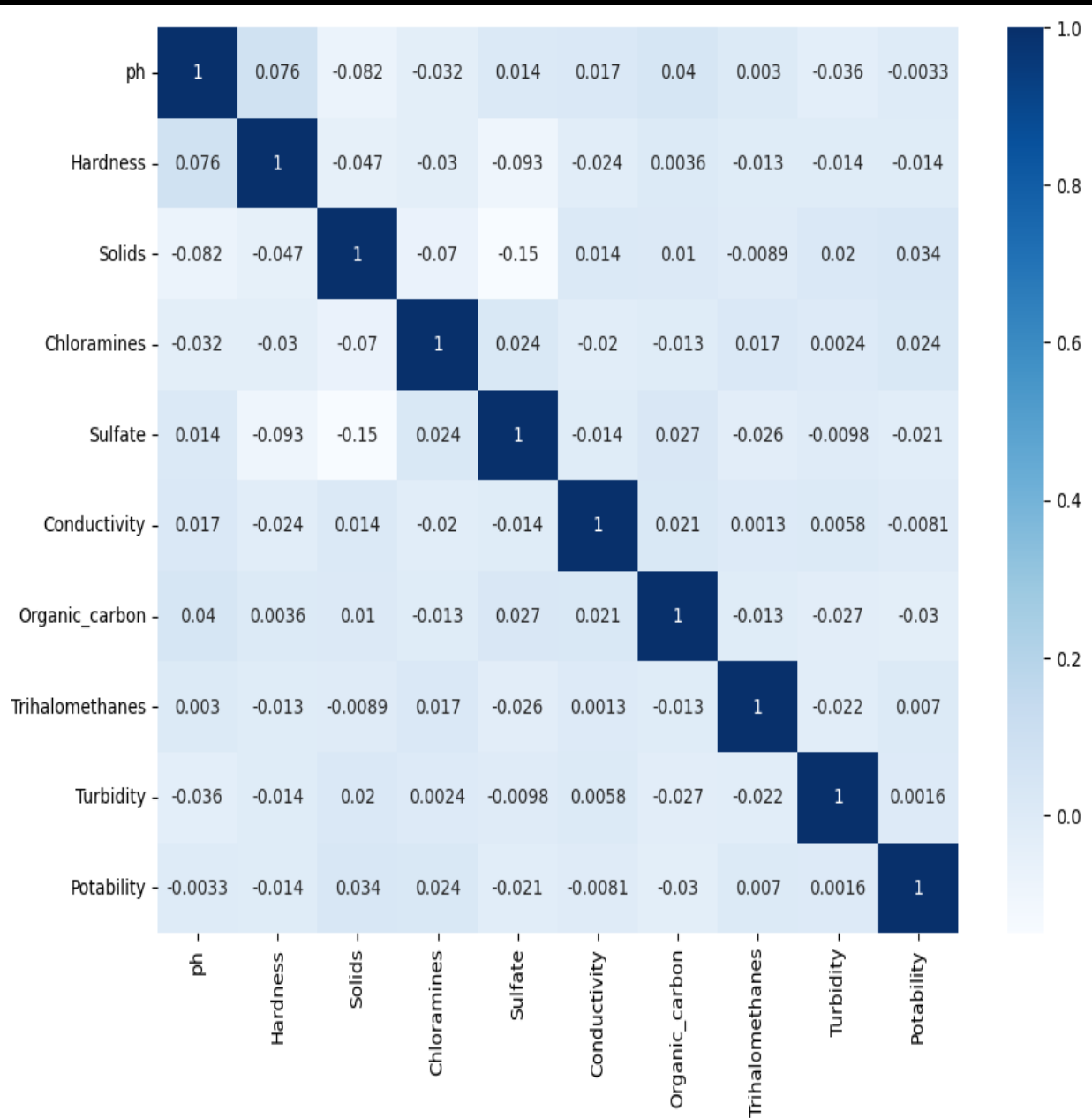
Matplotlib

- **Data Visualization:** Matplotlib creates diverse transportation data visualizations.
- **Exploratory Data Analysis(EDA):** Helps explore data, find trends, and detect outliers.
- **Performance Metrics:** Visualizes traffic patterns and efficiency metrics.
- **Customization and Interactivity:** Offers flexibility and interactivity for effective data representation.

White Grid :



Heatmap :



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.

A One-Class Support Vector Machine (One-Class SVM) is a machine learning algorithm used for novelty detection and anomaly detection. One-Class SVM is a powerful algorithm for identifying anomalies or outliers in datasets where one class dominates, making it valuable for various anomaly detection tasks.

The results of this algorithm can lead to more efficient for Anomaly detection to identify unusual patterns in Water Quality Parameters.

One Class – Support Vector Machine

A One-Class Support Vector Machine (One-Class SVM) is a machine learning algorithm used for novelty detection and anomaly detection. It is particularly useful when you have a dataset with only one class of examples (i.e., mostly normal data) and you want to identify rare anomalies or outliers within that class. One-Class SVM is a powerful algorithm for identifying anomalies or outliers in datasets where one class dominates, making it valuable for various anomaly detection tasks.

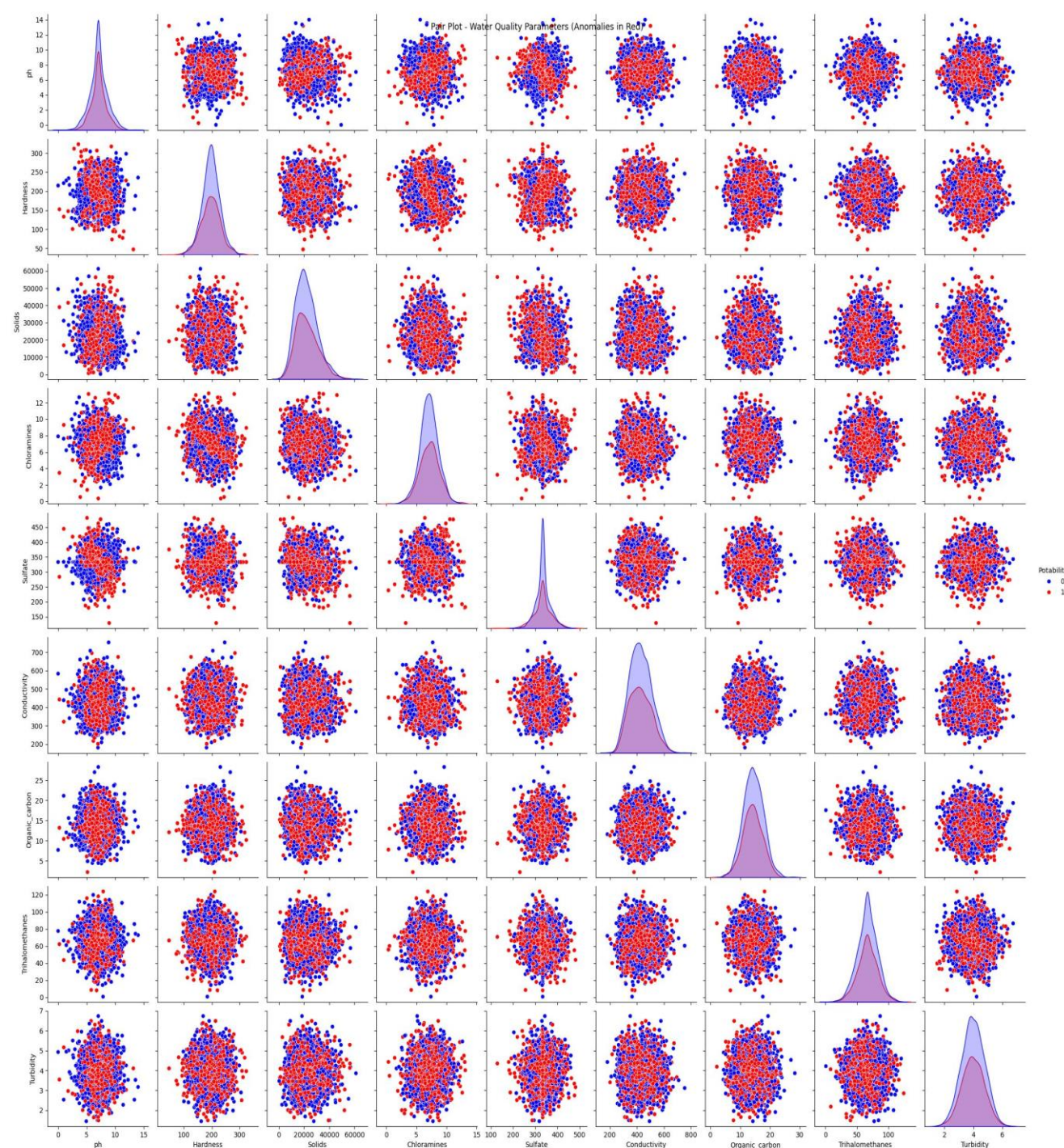
```
jupyter Anomalies Detection - SVM Last Checkpoint: 8 hours ago
File Edit View Run Kernel Settings Help
+ ✂ 📄 📌 ▶ ■ ↺ ⏮ Code ▼

[4]: import pandas as pd
import numpy as np
data = pd.read_csv('water_potability.csv')
for feature in data.columns:
    data[feature].fillna(data[feature].median(), inplace = True)

•[13]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import OneClassSVM
from sklearn.preprocessing import StandardScaler
import seaborn as sns

features = ['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate', 'Conductivity',
           'Organic_carbon', 'Trihalomethanes', 'Turbidity']
scaler = StandardScaler()
X = scaler.fit_transform(data[features])

nu = 0.05
model = OneClassSVM(kernel='rbf', nu=nu)
model.fit(X)
anomaly_scores = model.decision_function(X)
threshold = np.percentile(anomaly_scores, 5)
anomalies = data[anomaly_scores < threshold]
g = sns.pairplot(data, hue='Potability', palette={0: 'red', 1: 'blue'})
for ax in g.axes.flat:
    if ax.get_legend():
        legends = ax.get_legend().legendHandles
        legends[1].set_marker('o') # Marker for category 0
        legends[0].set_marker('x') # Marker for category 1
g.fig.suptitle('Pair Plot - Water Quality Parameters (Anomalies in Red)')
plt.show()
```



Output

In anomaly detection, we would normally identify anomalies (unusual patterns) within the "Non-Potable" or "0" class, as these are the cases where water quality issues or problems are typically found. "Potable" or "1" class represents the majority of data points and is considered normal.

Interpretation:

- **0 (Red):** This class typically represents the class of data points that are labeled as "Non-Potable" or "Unsafe." These are water quality samples that are considered unsafe or not suitable for consumption or use.

- **1 (Blue):** This class typically represents the class of data points that are labeled as "Potable" or "Safe for Consumption." These are water quality samples that are considered safe and suitable for consumption or use.

The Code Overview

The provided code performs anomaly detection on water quality data using a One-Class Support Vector Machine (SVM) and then visualizes the anomalies in a pair plot. Here's an overview of the code:

Data Loading and Preprocessing:

- The code starts by loading a water quality dataset from a CSV file and extracting all relevant parameters (features) that describe the water quality, such as pH, hardness, solids, chloramines, etc.
- The features are then standardized using the StandardScaler to ensure that they have a mean of 0 and a standard deviation of 1.

One-Class SVM for Anomaly Detection:

- A One-Class SVM model is created and fitted to the standardized features. The One-Class SVM is a machine learning algorithm that learns to identify anomalies in data.
- The nu parameter is set to control the sensitivity of the model. You can adjust it based on your dataset and the desired level of sensitivity to anomalies.

Anomaly Detection and Thresholding:

- Anomaly scores are computed for each data point using the decision function of the One-Class SVM model.
- A threshold is set to classify data points as anomalies. In the code, the threshold is set to the 5th percentile of the anomaly scores. This can be adjusted based on your requirements.

Identifying Anomalies:

- Data points with anomaly scores below the threshold are considered anomalies and are extracted from the dataset.

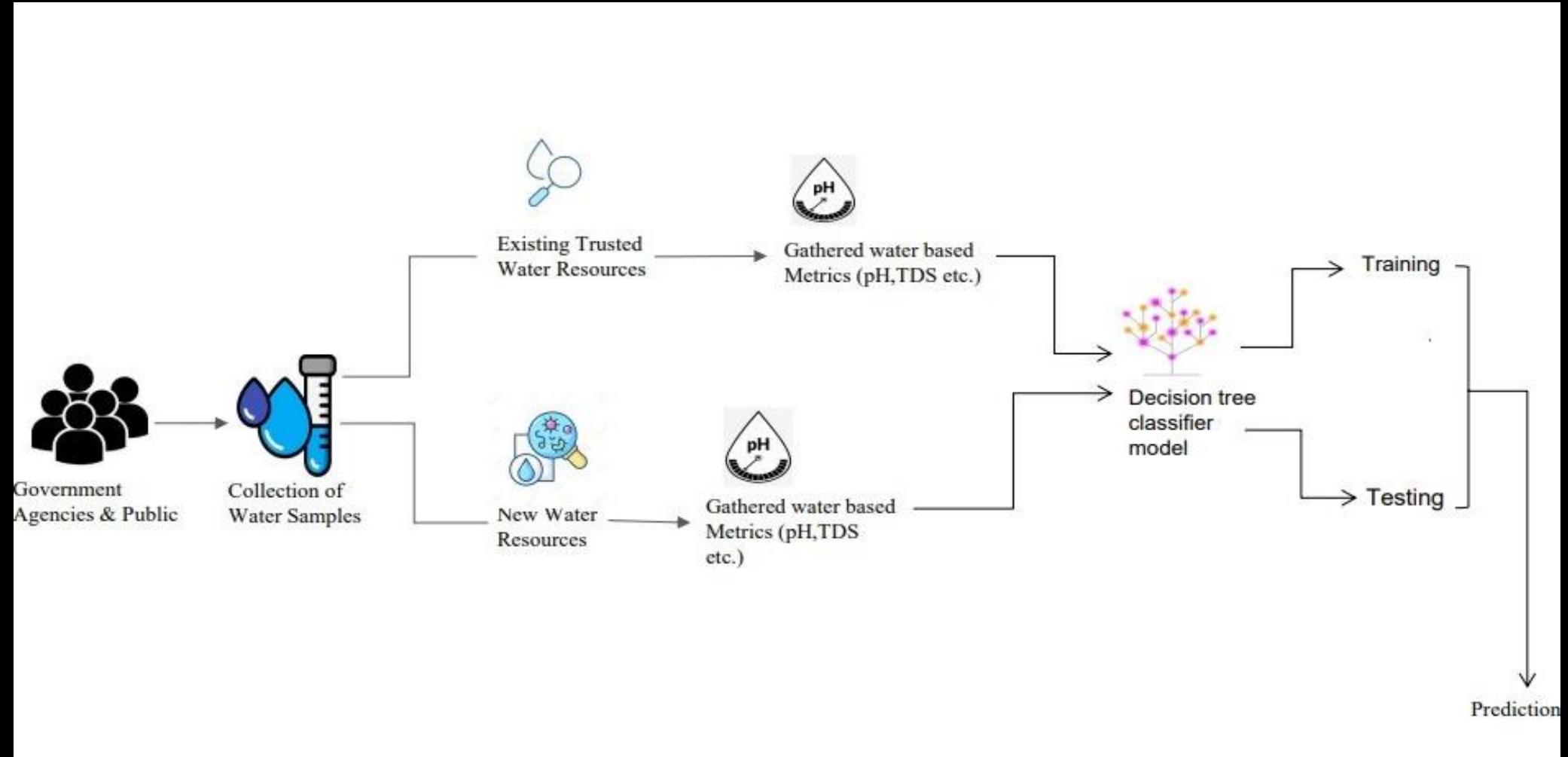
Visualization:

- A pair plot is created using Seaborn to visualize the relationships between all pairs of parameters in the water quality dataset.
- Data points are color-coded based on their 'Potability' status (assuming it's a binary classification variable in the dataset), with anomalies shown in red and normal data points in blue.

Proposed Solution

PARAMETER	DESCRIPTION
Problem Statement (Problem to be solved)	<ul style="list-style-type: none">Water quality prediction using machine learning techniques. Our model predicts the drinkability of the water based parameters such as pH, Conductivity, and hardness of the water.
Idea / Solution Description	<ul style="list-style-type: none">Water quality prediction model using the principal component analysis followed by decision tree classification.Firstly, the water quality index (WQI) is calculated using the weighted arithmetic index method.Secondly, the principal component analysis (PCA) is applied to the dataset, and the most dominant WQI parameters have been extracted.Thirdly, to predict the WQI, different regression algorithms are used to the PCA output.Finally, the decision tree classifier model is utilized to classify the Water quality status.

Flow Chart



Cognos Analytics in Action



Data Exploration

Through Cognos Analytics, we were able to explore the data and gain valuable insights into the public transportation efficiency, enabling us to make data-driven decisions.



Interactive Dashboard

With the interactive dashboards in Cognos Analytics, we created visually appealing representations of the analysed data, making it easier to understand and interpret for stakeholders.



Reporting and Sharing

Cognos Analytics provided us with the ability to generate comprehensive reports and share them with relevant stakeholders, ensuring effective communication of the analysis results.



Conclusion

In conclusion, advanced techniques for anomaly detection in water quality parameters can help unveil hidden patterns and improve the quality of water. Machine learning is a powerful tool for detecting anomalies and predicting their causes. Best practices and evaluation metrics can help ensure the accuracy and reliability of anomaly detection. Future directions can shape the field and open up new opportunities for research and innovation.