॥वसुधैव कुटुम्बकम्॥

Title: ML-Enabled Water Potability Classification

Author: Idhika Nikhil Vaidya
Course: MSc Data Science and Spatial Analytics
College: Symbiosis Institute of geoinformatics

Internship Project Report

Internship Organizations:
Technex IIT (BHU) Varanasi
Eisystems Services

Project mentors

Mr. Mayur Dev Sewak
General Manager, Operations
Eisystems Services

Ms. Mallika Srivastava
Trainer, Data Science & Analytics Domain
Eisystems Services

Acknowledgement

I would like to express my sincere gratitude to Technex IIT (BHU) Varanasi and Eisystems Services for their support and collaboration throughout my internship project.

I am grateful to Mr. Mayur Dev Sewak, General Manager, Operations at Ei Systems Services, and Ms. Mallika Srivastava, Trainer, Data Science & Analytics Domain at Eisystems Services, for their guidance, expertise, and valuable insights that have contributed significantly to the success of this project.

I would also like to thank the faculty and staff of Symbiosis Institute of Geoinformatics for their support and the knowledge they imparted during my studies, which laid the foundation for this project.

I extend my heartfelt appreciation to my friends, family, and all individuals who have directly or indirectly contributed to this project. Your support and encouragement have been invaluable.

I am grateful to all those who have contributed and been an integral part of this journey.

Table of Contents

List of Figures

List of Tables

Nomenclature

F-1 Score: Harmonic Mean of Precision and Recall
GaussianNB: Gaussian Naive Bayes
KNN: K-Nearest Neighbors
ML: Machine Learning
pH: Potential of Hydrogen
P: Precision
ROC AUC: Receiver Operating Characteristic Area Under Curve
R: Recall
SMOTE: Synthetic Minority Over-sampling Technique.
SVC: Support Vector Classification
WHO: World Health Organization
NTU: Nephelometric Turbidity Units

Abstract

This project aims to develop a machine learning model for predicting the potability of water samples. Access to safe drinking water is essential for maintaining good health and preventing the transmission of waterborne diseases.

The relevant data has been collected, analyzed and preprocessed. The preprocessed data is used for data analysis and visualizations to gain valuable insights into the features that affect the potability of a water sample.The data is prepared for model building, in which the Random Forest algorithm has the best overall performance (Accuracy: 80, Precision: 80, Recall: 61, F1-Score: 70 ,ROC-AUC: 99) .

The developed model provides a valuable tool for water quality management and public health initiatives by accurately identifying potential health risks associated with consuming contaminated water. The application of Random Forest in water potability classification demonstrates promising potential for using Machine Learning for addressing the critical issue of safe drinking water and empowering communities to ensure their health and well-being.

Project Summary

This project aims to classify water samples as potable or non-potable based on their features and gain insights into how these features affect water potability. The project's objectives, scope, and expected outcomes are summarized as follows:

Objectives

- To classify water samples as potable or not based on their features
- To gain insights on how the features of a water sample affect its potability

Scope

The project uses data about a diverse range of water samples from various sources, including drinking water reservoirs, wells, rivers, and groundwater. A comprehensive set of features related to chemical, physical, and biological properties of the water are there for each sample. The scope of the project encompasses data preprocessing, exploratory data analysis,data preparation, model building and output prediction.

Expected Outcomes

1. We anticipate the creation of a classification model that can accurately predict water potability based on the provided features. This model will enable efficient and automated assessment of water samples, aiding in the identification of potable and non-potable water sources.
2. Through analyzing the relationships between the water sample features and potability, we aim to gain valuable insights into how specific properties and characteristics influence water quality. This knowledge can contribute to improved understanding and decision-making in water resource management and public health initiatives.

In summary, this project endeavors to develop a classification model for determining water potability based on sample features while also uncovering insights into the factors influencing water quality. The outcomes will provide practical applications in the assessment of water samples and contribute to the broader understanding of water resource management and public health considerations.

Introduction

Access to safe drinking water is a fundamental human right, crucial for maintaining good health and preventing the spread of waterborne diseases. However, many regions face challenges in ensuring the availability of clean and potable water. Inadequate water supply and poor sanitation contribute to the transmission of diseases such as diarrhea, dysentery, hepatitis A, typhoid, cholera, and polio. The impact of these diseases on public health and development cannot be understated.

Addressing these issues requires a comprehensive approach at national, regional, and local levels, with a focus on safe drinking water policies and effective management of water resources. The costs associated with providing clean water are far outweighed by the expenses incurred in dealing with the health consequences of contaminated water sources.

One area where machine learning can play a significant role is in predicting the potability of water samples. By developing a model that can accurately classify whether water is safe for consumption or not, we can enhance decision-making processes and prioritize resources to ensure that communities have access to clean drinking water.

This project aims to utilize a machine learning model to assess the potability of water samples. By analyzing various parameters and features of the water, such as pH levels, hardness, solids content, and presence of chemicals, the model will provide a classification indicating whether the water is fit for consumption. Such a model can be instrumental in guiding policymakers, water management authorities, and healthcare professionals in their efforts to safeguard public health and improve water quality.

By leveraging the power of machine learning, we can take significant strides towards ensuring safe drinking water for all, promoting better health outcomes, and minimizing the risks associated with contaminated water sources.

Metadata

Water quality metrics for 3276 different water bodies are contained in the input file . Each water sample has 10 features.The dependent variable is potability and the other variables are independent.

1. pH value: PH is a crucial parameter in assessing the acid-base equilibrium of water and is indicative of its acidic or alkaline state. The World Health Organization (WHO) recommends a pH range of 6.5 to 8.5 as the maximum permissible limit for water. The findings of the recent study revealed pH values between 6.52 to 6.83, which fall within the WHO recommended range.

2. Hardness: Hardness is mainly caused by calcium and magnesium salts. These salts are dissolved from geologic deposits through which water travels. The length of time water is in contact with hardness-producing material helps determine how much hardness there is in raw water. Hardness was originally defined as the capacity of water to precipitate soap caused by Calcium and Magnesium.

3. Solids (Total dissolved solids - TDS): Water has the ability to dissolve a wide range of inorganic and some organic minerals or salts such as potassium, calcium, sodium, bicarbonates, chlorides, magnesium, sulfates, etc. These minerals produced an unwanted taste and diluted color in the appearance of water. This is the important parameter for the use of water. The water with a high TDS value indicates that water is highly mineralized. The desirable limit for TDS is 500 mg/l and the maximum limit is 1000 mg/l which is prescribed for drinking purposes.

4. Chloramines: Chlorine and chloramine are the major disinfectants used in public water systems. Chloramines are most commonly formed when ammonia is added to chlorine to treat drinking water. Chlorine levels up to 4 milligrams per liter (mg/L or 4 parts per million (ppm)) are considered safe in drinking water.

5. Sulfate: Sulfates are naturally occurring substances that are found in minerals, soil, and rocks. They are present in ambient air, groundwater, plants, and food. The principal commercial use of sulfate is in the chemical industry. Sulfate concentration in seawater

is about 2,700 milligrams per liter (mg/L). It ranges from 3 to 30 mg/L in most freshwater supplies, although much higher concentrations (1000 mg/L) are found in some geographic locations.

6. Conductivity: Pure water is not a good conductor of electric current rather it's a good insulator. An increase in ions concentration enhances the electrical conductivity of water. Generally, the amount of dissolved solids in water determines electrical conductivity. Electrical conductivity (EC) actually measures the ionic process of a solution that enables it to transmit current. According to WHO standards, EC value should not exceed 400 μS/cm.

7. Organic_carbon: Total Organic Carbon (TOC) in source waters comes from decaying natural organic matter (NOM) as well as synthetic sources. TOC is a measure of the total amount of carbon in organic compounds in pure water. According to the US EPA < 2 mg/L as TOC in treated / drinking water, and < 4 mg/Lit in source water which is used for treatment.

8. Trihalomethanes: THMs are chemicals that may be found in water treated with chlorine. The concentration of THMs in drinking water varies according to the level of organic material in the water, the amount of chlorine required to treat the water, and the temperature of the water that is being treated. THM levels up to 80 ppm are considered safe in drinking water.

9. Turbidity: The turbidity of water depends on the quantity of solid matter present in the suspended state. It is a measure of the light-emitting properties of water and the test is used to indicate the quality of waste discharge with respect to the colloidal matter. The mean turbidity value obtained for Wondo Genet Campus (0.98 NTU) is lower than the WHO recommended value of 5.00 NTU.

10. Potability: Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable.

Methodology

Importing the necessary libraries:

Pandas, Numpy, Sklearn, Seaborn, Matplotlib
Functions: warnings, train_test_split, StandardScaler,accuracy_score
Classification Algorithms: LogisticRegression, SVC, GaussianNB, KNeighbors, RandomForest

A. Data Collection

1. Loading the dataset: The dataset is loaded into a pandas dataframe. It is collected from a publicly available dataset [1]

B. Data Preprocessing

2. To get more information about the dataset the functions used are:
head(), info(), shape, describe() ,columns

3. Removing null values

- We check for the number of null values in the dataset using isnull().sum(). 3 columns in the dataset contain a lot of null values. These columns are Ph Value, Sulfate and Trihalomethanes

- To decide how to handle the null values in the data we plot histograms which show the distribution for the columns which contain null values.
As all three plots show bell shaped curves , we can conclude their distribution is normal. Thus it is most suitable to fill the null values using the mean.

- We fill the null values in the three columns with the mean of the class label that they belong to (Potability: 1 or 0) . By using class-specific means, we aim to preserve the class-specific information while filling the missing values. This can potentially help in preserving the class-specific characteristics and patterns in the data.

- We check if the null values have been removed. The null values have been removed successfully.

Figure 1: Histogram plot of Ph Value

Figure 2: Histogram plot of Sulfate

Figure 3: Histogram plot of Trihalomethanes

C. EXPLORATORY DATA ANALYSIS

1. Correlation Matrix

   Heatmap is used to visualize the correlation matrix. This is done using the Seaborn library.

   Figure 4: Correlation matrix

2. Pairplot

   It is visualized using the Seaborn library. This visualization tells us how the variables are related to each other and how the values of each class label are positioned.

   Figure 5: Pairplot

D. DATA PREPARATION

1. Train-test split

- We select the dependent and independent features:
    - Independent features: ph, Hardness, Solids, Chloramines, Sulfate , Conductivity , Organic carbon, Trihalomethanes and Turbidity

    - Dependent feature: Potability

- We use the train-test-split method to split the dataset into training data and testing data. We see the shape of the training and testing dataset and ensure whether they have the same number of columns. They have the same number of columns

2. Removing Data Imbalance

- We plot a barplot to check for a data imbalance between class labels in the potability column.Data imbalance does exist, more records of not potable water than potable.
- To fix the data imbalance SMOTE function is used. This function creates synthetic data of the minority class and fixes the data imbalance.

Figure 6: Data Imbalance Barplot

3. Feature Scaling

Standard Scaler is used because features of the given input dataset fluctuate significantly within their ranges. They are recorded in various units of measurement. Thus StandardScaler standardizes the input features so that they can be used in the machine learning model.

4. Feature Selection

Extra Trees Classifier: It prints the feature importances of the trained model. It is an algorithm in the sklearn library.
We can decide which features are relevant and need to be used by the model. We plot the feature importances to compare and evaluate them. A threshold of 0.075 is set for feature importance, features that have more feature importance than the threshold value will be used in the model.
All the features are important thus we will take them all for prediction.

E. Model Building

1. Multiple models fit on the data and their accuracy is tested using various evaluation metrics. These 5 classification machine learning models are:

- SVC, or Support Vector Machine, is a type of supervised learning algorithm that finds the hyperplane that best separates data into different classes for improved classification and regression tasks.

- K-Nearest Neighbors is another supervised learning algorithm that predicts values based on the majority vote of the K nearest neighbors, making it effective for handling complex and nonlinear relationships between features.

- Logistic Regression is a type of supervised learning algorithm that estimates the probability of an event occurring based on input variables for improved classification tasks.

- Gaussian Naive Bayes is another supervised learning algorithm that estimates the probability of an event occurring based on input variables, assuming they are independent and have a Gaussian distribution, making it effective for handling high-dimensional data and datasets with a small number of observations.

- Random Forest is an ensemble learning algorithm that uses decision trees and is employed for classification and regression tasks. The algorithm generates numerous decision trees by randomly selecting data and features, and then combines the predictions from all the trees to produce a final prediction.

2. The evaluation metrics selected are:

- Accuracy: This metric quantifies the percentage of correct predictions made by the model out of the total number of predictions. It is a widely used measure in assessing overall model performance.

- Precision: Precision measures how many of the positive predictions made by the model are actually true positives. It helps determine the model's ability to minimize false positives.

- Recall: Recall measures how many of the actual positive examples in the data were correctly identified by the model. It is an indicator of the model's ability to avoid false negatives.

- F-1 Score: The F-1 Score is a combined measure of precision and recall. It takes the harmonic mean of the two metrics and provides a balanced evaluation of the model's performance, particularly in scenarios where precision and recall are both important.

- ROC AUC: The ROC AUC (Receiver Operating Characteristic Area Under the Curve) metric evaluates how well the model can distinguish between positive and negative examples in the data. It does this by plotting the True Positive Rate against the False Positive Rate, providing an overall measure of the model's classification performance.

These evaluation metrics will be used to judge the overall performance of the model.

3. Model's name and its performance on the evaluation metrics is stored in a dataframe called 'res'.This dataframe is printed and arranged such that the accuracy_score is in a descending order.

4. The res dataframe is visualized as a barplot. We can compare the performance of the algorithms.

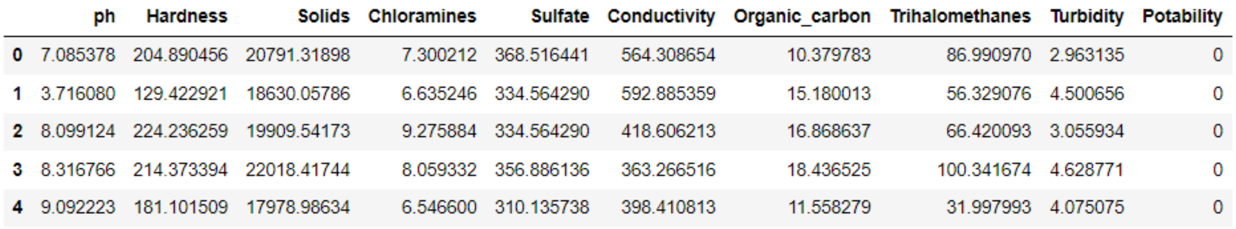
E. Output Prediction

The model is used to predict class labels on the testing data. The predicted output is displayed. The potability values (class labels) for the records in the testing dataset is predicted as 1 or 0

Results

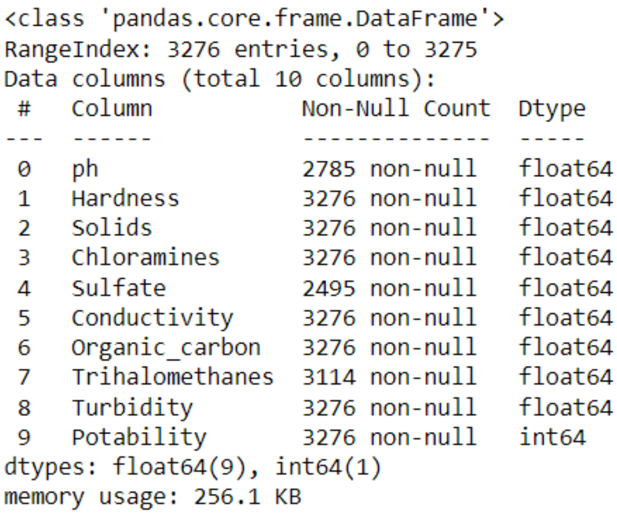
A. Data Preprocessing

Table 1: Sample records



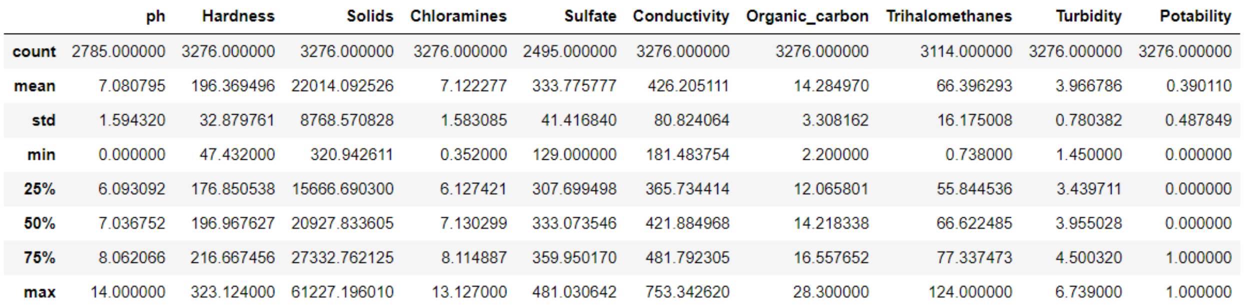
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.085378	204.890456	20791.31898	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.05786	6.635246	334.564290	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.54173	9.275884	334.564290	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.41744	8.059332	356.886136	363.266516	18.436525	100.341674	4.628771	0
4	9.092223	181.101509	17978.98634	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

Table 2: Summary of the dataset



```
<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
```

Table 3:Statistical summary of the dataset



	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
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	15666.690300	6.127421	307.699498	365.734414	12.065801	55.844536	3.439711	0.000000
50%	7.036752	196.967627	20927.833605	7.130299	333.073546	421.884968	14.218338	66.622485	3.955028	0.000000
75%	8.062066	216.667456	27332.762125	8.114887	359.950170	481.792305	16.557652	77.337473	4.500320	1.000000
max	14.000000	323.124000	61227.196010	13.127000	481.030642	753.342620	28.300000	124.000000	6.739000	1.000000

3. Histogram plots

Description:
These plots show the distribution of the data in the columns which have null values.

Insights:
Figure 1, Figure 2 and Figure 3 show a bell shaped curve which implies the data in the columns has a normal distribution. Thus the null values can be filled using the mean.

Fig 1: Histogram plot of Ph Value

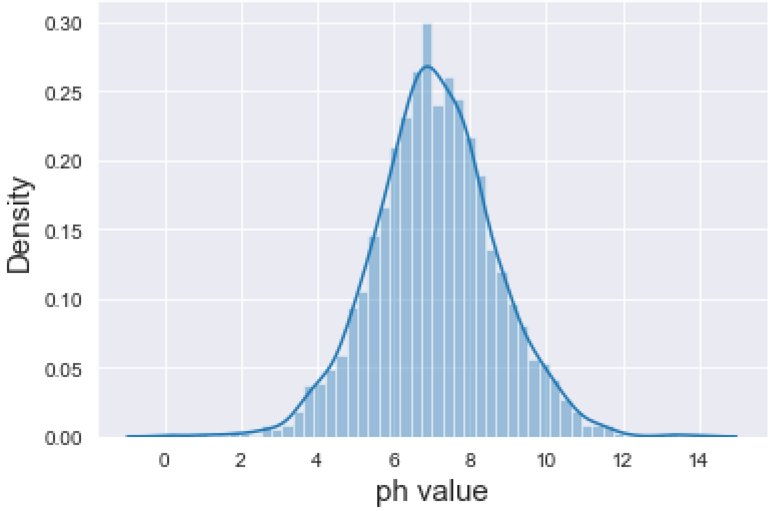
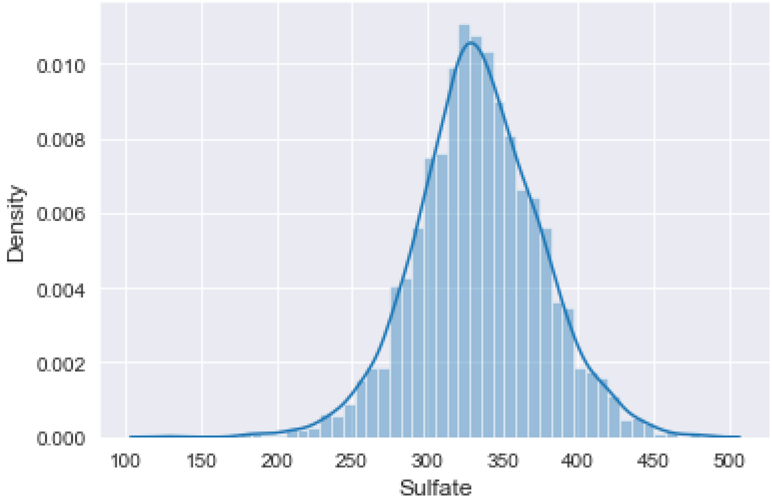


Fig 2: Histogram plot of Sulfate



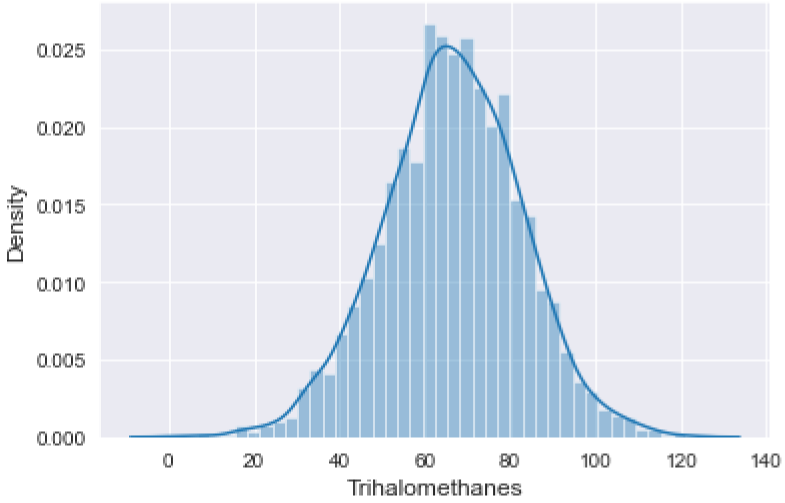


Fig 3: Histogram plot of Trihalomethanes

B. Exploratory Data Analysis

Figure 4: Correlation matrix

Description:
This shows correlation between the columns in the dataset.Heatmap is used to visualize the correlation matrix.

Insights:

Correlation of the features of the water sample with its potability:
All the features are correlated with the potability of water.
Positive correlation: Solids, Chloramines, Trihalomethanes, Turbidity
Negative correlation: ph, Hardness, Sulfate, Conductivity, Organic carbon, Trihalomethanes, Turbidity

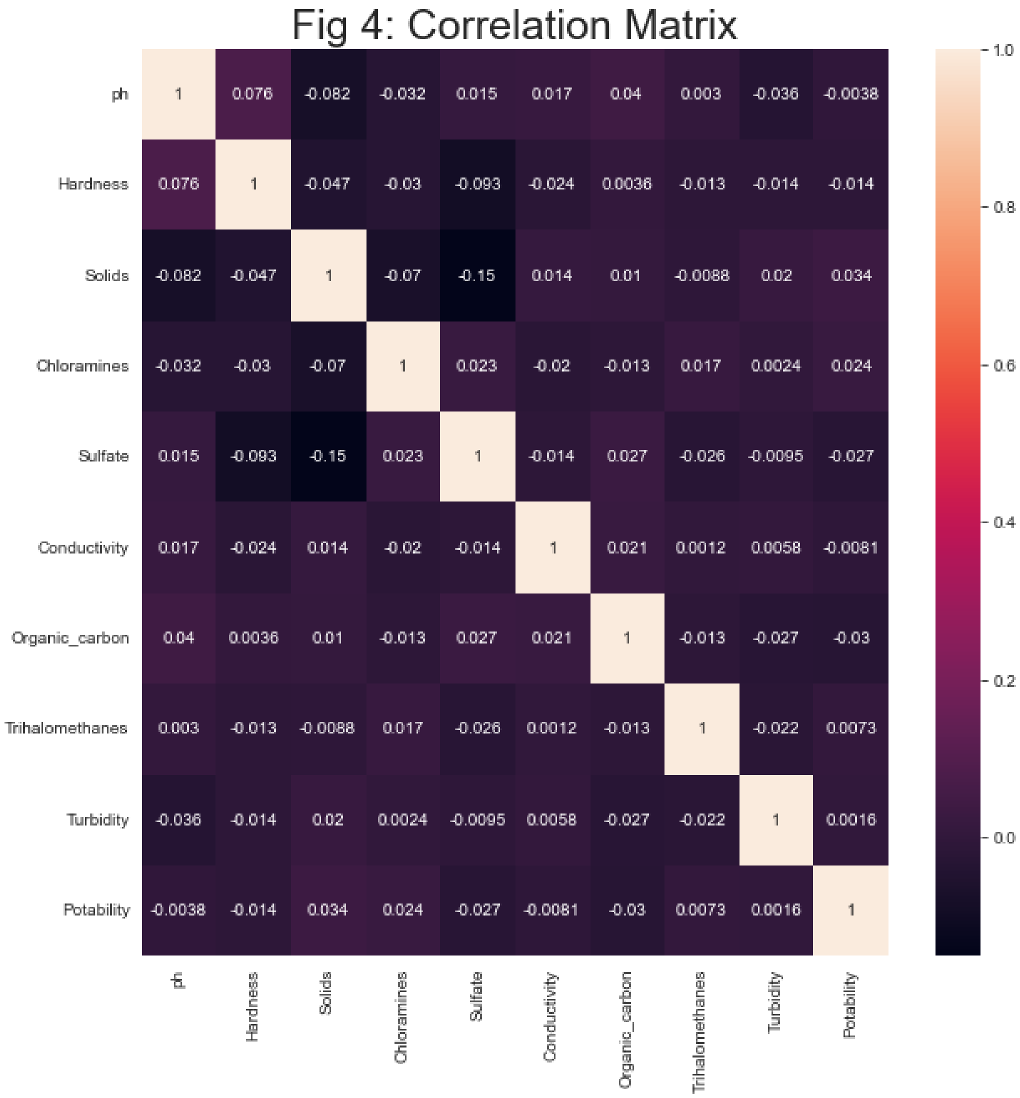
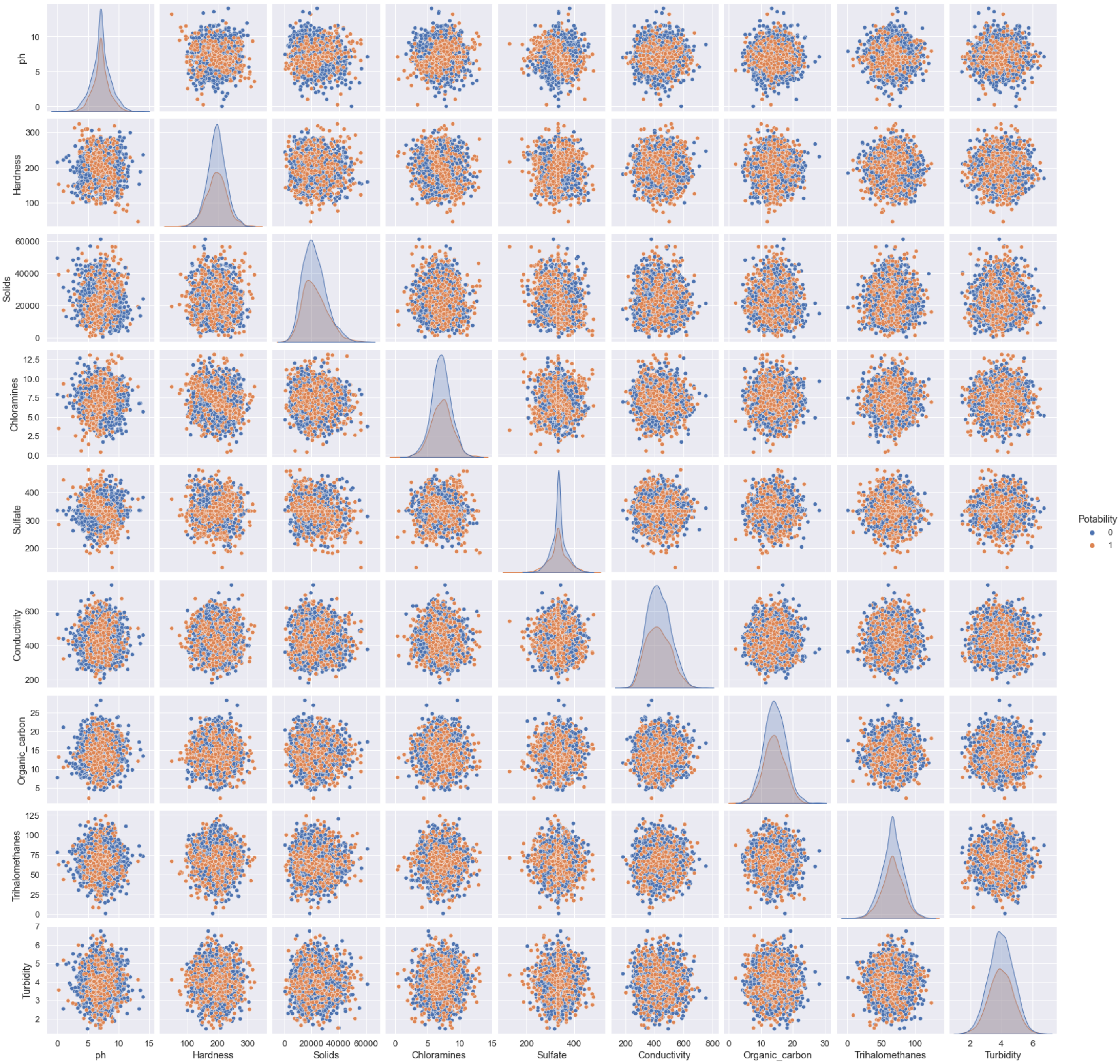


Fig 4: Correlation Matrix

Figure 5: Pairplot

Description:
The data values are color coded using their class labels. This visualization tells us how the variables are related to each other and how the values of each class label are positioned.
Insights:
The data points representing drinkable water are concentrated towards the center of the scatterplot, this implies that certain variables or features tend to have values that fall within a balanced or moderate range for potable water samples.
The data points representing non-drinkable water samples tend to be located away from the center of the scatterplot.

These observations provide further evidence of the correlation between the features of the water samples and their potability.



C. Data Preparation

Figure 6: Data Imbalance Barplot
Description: It shows the value counts of samples in each class label

Insights:
There is a much larger proportion of water samples of non drinkable water in the dataset than drinkable water samples therefore there is a data imbalance. The data imbalance will have to be fixed to increase the model's accuracy.

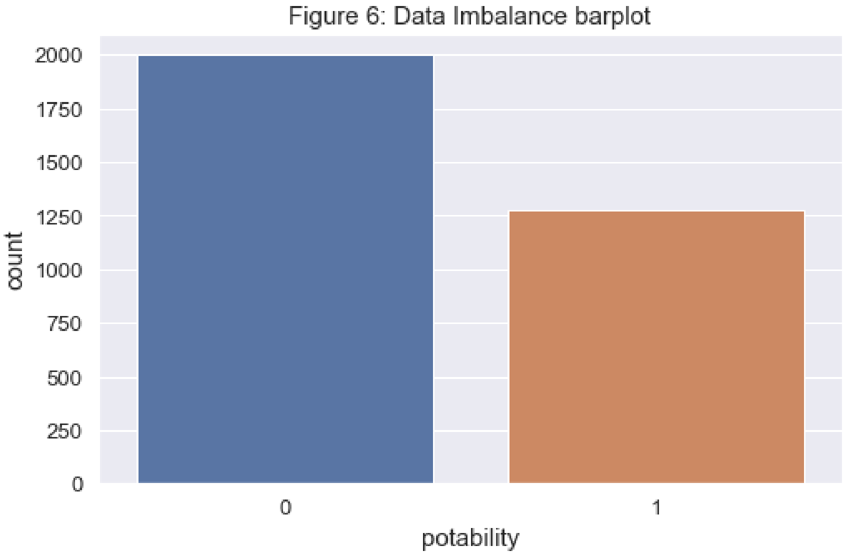
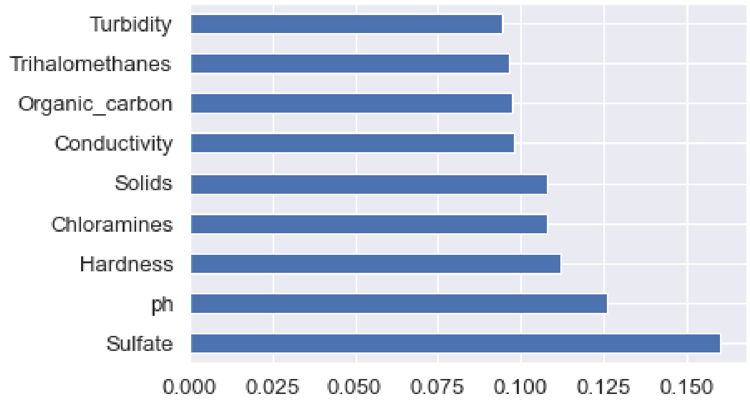


Figure 7: Feature Importance

Description:
The importance of features for predicting the class label of the water sample is plotted using a horizontal bar chart.

Insights:
All the features have a good feature importance and thereby must be considered while training and testing the ML model.
Sulphate has the highest feature importance.

Figure 7: Feature Importance

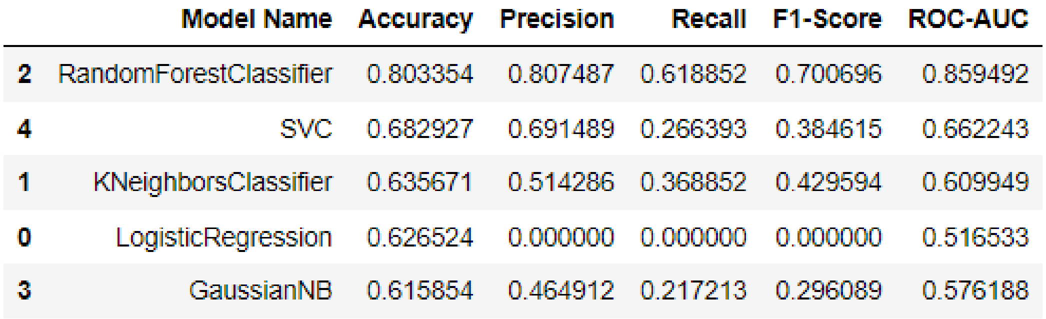


D. Model Building

Table 4: Model Evaluation

Description: The table shows the performance of the 5 algorithms on the evaluation metrics used.

Insights: Random Forest has the best overall performance for water potability classification in all the evaluation metrics. and thus should be used in the ML model.

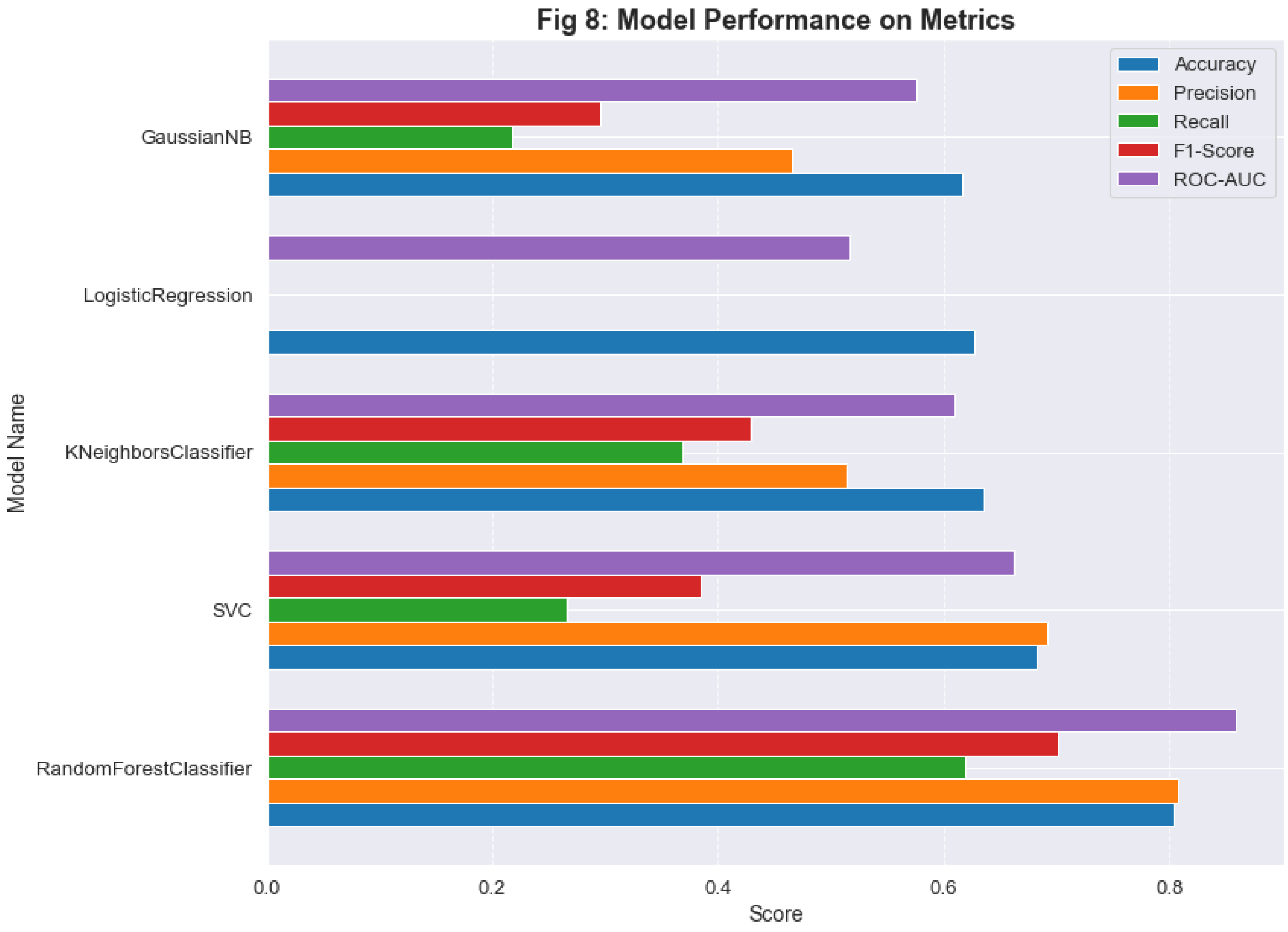


	Model Name	Accuracy	Precision	Recall	F1-Score	ROC-AUC
2	RandomForestClassifier	0.803354	0.807487	0.618852	0.700696	0.859492
4	SVC	0.682927	0.691489	0.266393	0.384615	0.662243
1	KNeighborsClassifier	0.635671	0.514286	0.368852	0.429594	0.609949
0	LogisticRegression	0.626524	0.000000	0.000000	0.000000	0.516533
3	GaussianNB	0.615854	0.464912	0.217213	0.296089	0.576188

Figure 8: Model Performance on Metrics

Description: Horizontal bar chart that shows the model's performance over various evaluation metrics.

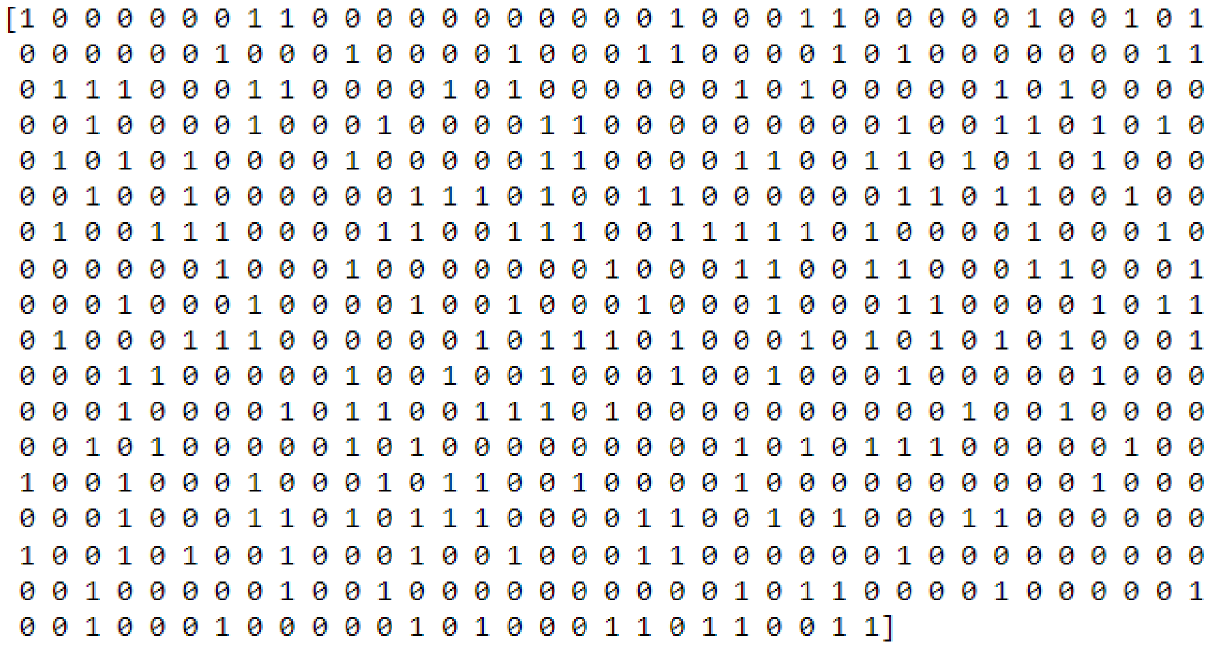
Insights:
Random performance has a very good performance compared to all the other algorithms used. Compared to single decision trees, Random Forest offers various benefits such as better handling of missing data, improved accuracy, and reduced overfitting. It is widely used in applications including fraud detection, customer segmentation, and image classification. Random Forest is a good option for problems with high-dimensional feature spaces, complex input-output relationships, and noisy data, such as Water Potability Classification.



E. Output Prediction

Description: Series of predicted class labels for the records in the testing dataset

Insights: The Random Forest Classifier has successfully predicted the class labels for the testing dataset with an accuracy of 80%.



```
[1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1
 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1
 0 1 1 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0
 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 0 0
 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 0
 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1
 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1
 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0 0 1
 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0
 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0
 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 0 1 0 0
 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
 0 0 0 1 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0
 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1
 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 0 1 1]
```

Conclusion

In conclusion, the developed water potability classification machine learning model demonstrates its value in predicting the safety of water samples for human consumption. By utilizing a dataset comprising various water quality parameters such as pH, hardness, and chemical concentrations, the model can effectively classify water samples as potable or non-potable.

The model's capability to accurately identify potential health risks associated with contaminated water makes it a valuable asset for water quality management and public health initiatives. It can assist decision-makers in prioritizing resources, implementing appropriate interventions, and ensuring access to safe drinking water.

To further enhance the model's accuracy and robustness, future improvements can be made by fine-tuning its hyperparameters or by incorporating more extensive and diverse training data. This would contribute to a more comprehensive understanding of water potability and enable more accurate predictions.

Overall, the water potability classification machine learning model offers promising potential in addressing the critical issue of safe drinking water, empowering communities to safeguard their health and well-being.

System Requirement

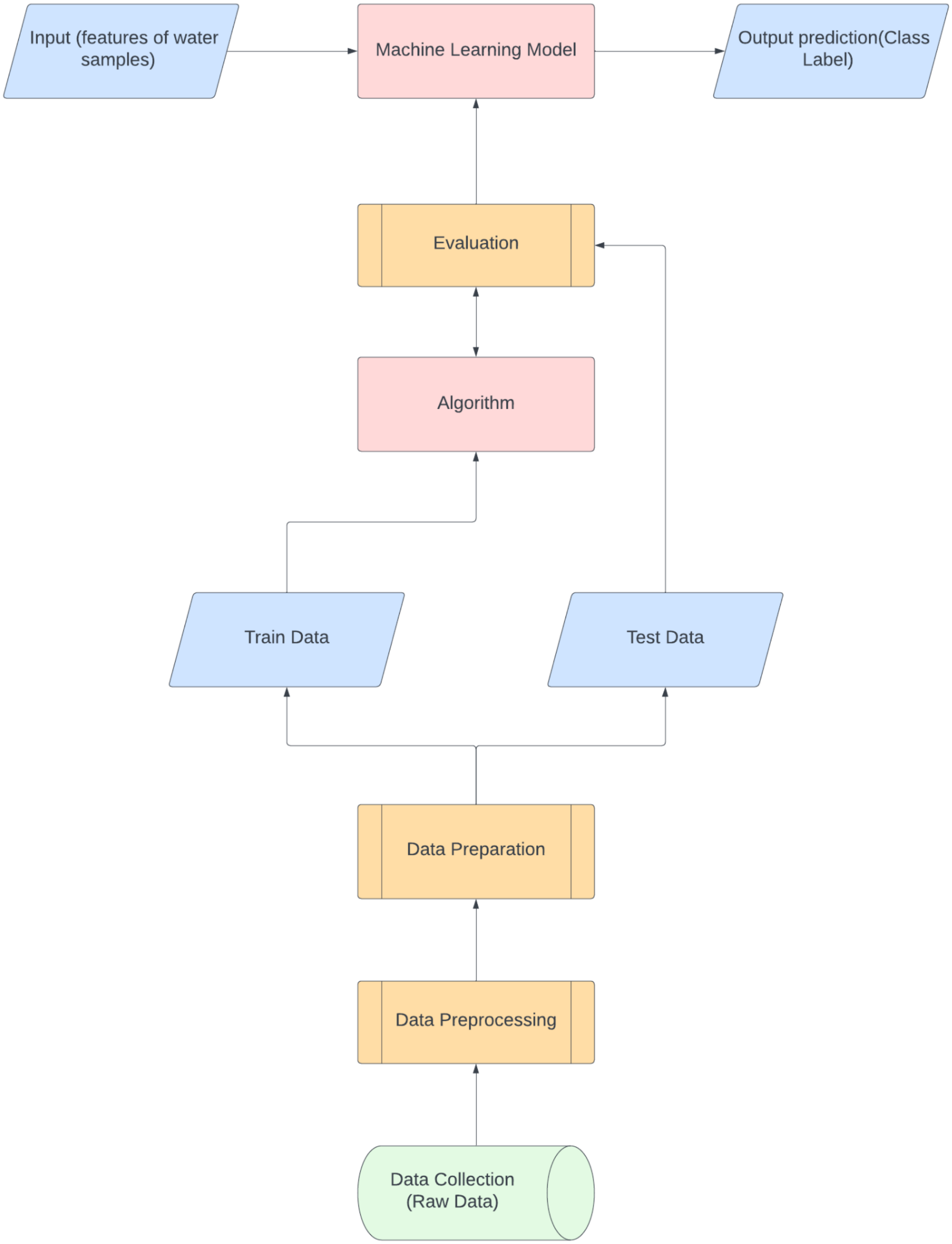
- Anaconda navigator
- Jupyter Notebook

The following python libraries need to be installed:

- Numpy
- Pandas
- Scikit-learn
- Matplotlib and Seaborn

Data Flow Diagram

Figure 9: Data Flow diagram



Code

```
# Water Potability Classification
"""

#import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

#importing functions
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

#importing Classification ML Algorithms
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier

"""Data Collection"""

df =pd.read_csv(r"/content/drinking_water_potability.csv")

#style for the visualizations
sns.set_style('darkgrid')

"""## Data Preprocessing"""
```

```
df.head()

df.info()

df.shape

df.describe()

#Visualising the statistical summary using boxplots

for column in df.columns[:-1]:
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df[column])
    plt.title('Box plot of {}'.format(column), fontsize=20)

df.columns

df.isnull().sum()

#To decide how to fill the null values, we plot the distribution of the columns with null values
sns.distplot(df['ph'])
plt.xlabel('ph value', fontsize=15)
plt.ylabel('Density', fontsize=15)
plt.title('Histogram plot of Ph Value', fontsize=18)

sns.distplot(df['Sulfate'])
plt.xlabel('Sulfate', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.title('Histogram plot of Sulfate', fontsize=18)

sns.distplot(df['Trihalomethanes'])
plt.xlabel('Trihalomethanes', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.title('Histogram plot of Trihalomethanes', fontsize=18)

#the columns with null values have a normal distribution thus we will fill
#the null values using mean
```

```
''' preparing data for model '''
#filling null values depending on the mean of each class label of Potability: to increase the accuracy of the model
ph_mean = df[df['Potability'] == 0]['ph'].mean(skipna=True)
df.loc[(df['Potability'] == 0) & (df['ph'].isna()), 'ph'] = ph_mean

ph_mean_1 = df[df['Potability'] == 1]['ph'].mean(skipna=True)
df.loc[(df['Potability'] == 1) & (df['ph'].isna()), 'ph'] = ph_mean_1

sulf_mean = df[df['Potability'] == 0]['Sulfate'].mean(skipna=True)
df.loc[(df['Potability'] == 0) & (df['Sulfate'].isna()), 'Sulfate'] = sulf_mean

sulf_mean_1 = df[df['Potability'] == 1]['Sulfate'].mean(skipna=True)
df.loc[(df['Potability'] == 1) & (df['Sulfate'].isna()), 'Sulfate'] = sulf_mean_1

traih_mean = df[df['Potability'] == 0]['Trihalomethanes'].mean(skipna=True)
df.loc[(df['Potability'] == 0) & (df['Trihalomethanes'].isna()), 'Trihalomethanes'] = traih_mean

trah_mean_1 = df[df['Potability'] == 1]['Trihalomethanes'].mean(skipna=True)
df.loc[(df['Potability'] == 1) & (df['Trihalomethanes'].isna()), 'Trihalomethanes'] = trah_mean_1

#Check if null values have been removed
df.isnull().sum()

"""## Relational Analysis"""

#correlation matrix
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True)
plt.title('Correlation Matrix',fontsize=25)

#Pairplot
plt.figure(figsize=(10,10))
sns.pairplot(data=df,hue='Potability')
#orange: potable water
#blue: not potable water
```

```
#Visualising the statistical summary

''' box plot is used to represent the statistical summary of all the features in the dataset'''

"""## Feature Selection"""

# lets see feature importance
from sklearn.ensemble import ExtraTreesClassifier
x = df.drop(['Potability'],axis=1) #independent variables
y =df.Potability #dependent variables
Ext = ExtraTreesClassifier()
Ext.fit(x,y)
print(Ext.feature_importances_)

#plotting the importance of features
feature = pd.Series(Ext.feature_importances_,index=x.columns)
feature.sort_values(ascending=True).nlargest(10).plot(kind='barh')

# Define a threshold for feature importance score
threshold = 0.075

# Get the names of features with importance score less than the threshold
to_drop = feature[feature < threshold].index.tolist()

# Drop the less important features from the dataset
X_selected = X.drop(to_drop, axis=1)

# Print the remaining features
print(X_selected.columns)

#All the features are important thus we will take them all for prediction

"""## Model Building"""

''' independent and dependent features '''
X = df.iloc[:, :-1] #independent variable
y = df.iloc[:, -1]  #dependent variable: Potability
```

```
''' train test split '''
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

print("X_train shape: ", X_train.shape)
print("X_test shape: ", X_test.shape)

#features of the given dataset fluctuate significantly within their ranges
#They are recorded in various units of measurement
#We will use Standard Scaler
''' standard scaler '''
sc = StandardScaler()

sc.fit_transform(X_train)
sc.transform(X_test)

#Value counts of potability: to check if the data is imbalanced
pot_lbl = df.Potability.value_counts()

#barplot
plt.figure(figsize=(8,5))
sns.barplot(x=pot_lbl.index, y=pot_lbl)
plt.xlabel('Potability',fontsize=15)
plt.ylabel('count',fontsize=15)

#To fix the data imbalance, the synthetic data is created to increase the samples of the minority
class
#and make them equal to the samples of the majority class
from imblearn.over_sampling import SMOTE

# apply SMOTE
smote = SMOTE(random_state=42)
smote.fit_resample(X_train, y_train)

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_auc_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

# Define the list of models to evaluate
models = [
    LogisticRegression(),
    KNeighborsClassifier(),
    RandomForestClassifier(),
    GaussianNB(),
    SVC()
]

# Define empty lists to store the performance metrics of each model
models_acc = []
models_prec = []
models_recall = []
models_f1 = []
models_roc_auc = []

# Iterate over the models and evaluate their performance
for model in models:
    # 1. Model Training
    model.fit(X_train, y_train)

    # 2. Model Evaluation
    # Compute the performance metrics and append them to the corresponding lists
    models_acc.append(accuracy_score(y_test, model.predict(X_test)))
    models_prec.append(precision_score(y_test, model.predict(X_test)))
    models_recall.append(recall_score(y_test, model.predict(X_test)))
    models_f1.append(f1_score(y_test, model.predict(X_test)))
    models_roc_auc.append(roc_auc_score(y_test, model.predict_proba(X_test)[:, 1] if hasattr(model, "predict_proba") else model.decision_function(X_test)))

''' Creating a dataframe '''
res = pd.DataFrame({
    "Model Name": ['LogisticRegression', 'KNeighborsClassifier', 'RandomForestClassifier', 'GaussianNB', 'SVC'],
    'Accuracy': models_acc,
    'Precision': models_prec,
    'Recall': models_recall,
    'F1-Score': models_f1,
    'ROC-AUC': models_roc_auc
})

print('Table 4: Model Evaluation')
res.sort_values(by=['Accuracy'], ascending=False)

import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(12, 10))

# Define colors for each metric
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd"]

# Sort models by accuracy
res = res.sort_values(by="Accuracy", ascending=False)

# Plot horizontal bar chart with different colors for each metric
res.plot(x="Model Name", y=["Accuracy", "Precision", "Recall", "F1-Score", "ROC-AUC"],
         kind="barh", ax=ax, color=colors, width=0.75)

# Add title and axis labels
ax.set_title("Fig 8: Model Performance on Metrics", fontsize=18, fontweight="bold")
ax.set_xlabel("Score", fontsize=14)
ax.set_ylabel("Model Name", fontsize=14)

# Add grid lines
ax.grid(axis="x", linestyle="dashed", alpha=0.7)

plt.show()

"""## Output prediction"""

# Initialize a random forest classifier with default hyperparameters
rf = RandomForestClassifier()
```

```
# Train the random forest model on the training data
rf.fit(X_train, y_train)

# Predict on the testing data
y_pred = rf.predict(X_test)
print(y_pred)
```

References

https://www.kaggle.com/datasets/artimule/drinking-water-probability