1. INTRODUCTION

1.1 Introduction

Machine Learning is a branch of Artificial Intelligence that aims at solving real-life engineering problems. This technique requires no programming, whereas it depends on only data learning where the machine learns from pre-existing data and predicts the result accordingly. Machine Learning methods have the benefit of using decision trees, heuristic learning, knowledge acquisition, and mathematical models. It thus provides controllability, observability, stability, quality, and effectiveness.

Water is utilized for drinking, household usage, food production, or leisure, safe and readily available water is critical for public health. Improving supplies of water, and also improved management of water resources, might help countries thrive and reduce poverty. There are many reasons why water is deteriorating because India there are many industrial areas so the release of pollutants in rivers is the main reason for water deteriorating. There are many other reasons for water deterioration like people’s garbage (plastics), the unwanted things in rivers, their nearest ponds, lakes, and also in the sea, and due to plastic and unwanted garbage, there are toxic occurrences. So, for all these reasons, water is deteriorating nowadays.

Contaminated water and inadequate sanitation have been related to diseases such as typhoid, dysentery, polio, cholera, hepatitis, and diarrhea. People are exposed to preventable health dangers due to a lack of, inadequate, or poorly managed water and sanitation facilities. It is especially the case in health facilities, at which water shortage, hygiene, and cleanliness assistance expose staff† and patients to viruses and bacteria.

The methodology is as follows: prepare the dataset, followed by data pre-processing such as dealing with missing values and categorical values. Feature selection will be performed by using a variety of tools. Lastly, the classifier performance before and after feature selection will be evaluated further.

1.2 Existing System

The Existing system is carrying validation of two classifiers, Random Forest Classifier, and Extra Trees Classifier, which have improved accuracy. We will now analyze the final model using these two classifiers. In order to evaluate the performance of the different classifier confusion matrices and classification report for different classifiers is generated. Here Usage of the confusion matrix for Decision Tree, XGBoost, AdaBoost, and SVC.

Disadvantages:

1. Doesn’t generate accurate and efficient results.
2. Computation time is very high.
3. Lacking accuracy may result in a lack of efficient further treatment.

1.3 Proposed System

The standards used to assess the sustainability of water resources are constantly evaluated as new factors are found. Standards and guidelines for contamination levels in drinking water are being developed by regulatory agencies. In response to the changing criteria, the water supply sector is creating new and hence by showing the importance of Machine Learning for predicting the water quality. All elements that affect water quality, as well as the public health relevance of components and available treatment technology, must be considered when developing drinking water quality guidelines.

Advantages:

- Generates accurate and efficient results
- Computation time is greatly reduced
- Reduces manual work
- Automated prediction

1.4 System Requirements

1.4.1 Hardware Requirements:

•	System Type	:	Intel Core i3 or above
•	Cache Memory	:	4MB (Megabyte)
•	RAM	:	8 gigabytes (GB)

1.4.2 Software Requirements:

•	Operating System	:	Windows 10 Home, 64-bit Operating System.
•	Coding Language	:	Python, Html & CSS
•	IDE	:	Visual Studio Code
•	Browser	:	Any latest browser like chrome

2. LITERATURE SURVEY

2.1 Literature Survey

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

This study looks into the approaches that were used to help solve water quality challenges. In most studies, traditional analyses in the laboratory and data analysis are two types of analysis utilized to help determine the quality of water, but other studies apply machine learning approaches to help find an optimal solution to the water quality problem.

Potable water quality is typically impacted by the source water's quality, how it is handled before being delivered, how it is distributed, how it is maintained, and how effectively it is filtered at residence. Furthermore, in rural areas and small municipalities, drinking water is frequently drawn straight from wells or retrieved unfiltered from rivers, lakes, and reservoirs. As a result, the purity of the source water is a significant factor affecting the quality of the drinking water. Many developing nations have achieved waterborne disease reduction and the development of safe water supplies is a significant public health aim in recent years, and the situation has improved slightly. Many water quality evaluation approaches have been proposed since Horton produced the first Water Quality Index (WQI) in the 1960s. The two indices for determining the general state of drinking source water quality are straightforward, adaptable, and stable, with little sensitivity to input data.

2.2 Some Machine Learning Methods

- **Supervised machine learning algorithms:**

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- **Unsupervised machine learning algorithms:**

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

- **Reinforcement machine learning algorithms:**

Reinforcement machine learning algorithms are a learning method that interacts with its environment by producing actions and discovering errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of Machine Learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

2.4 Advantages of Machine Learning

Machine learning algorithms automate analyzing and interpretation of data and can be used to build predictive models. It eliminates manual data analysis and allows organizations to make data-driven decisions quickly and accurately.

Machine learning algorithms employ pattern recognition techniques to analyze and to extract the meaningful insights from data, subsequently utilizing these insights to make more accurate predictions. It can be beneficial when dealing with large datasets or constantly changing data.

Machine learning algorithms can automate specific processes, reducing the time required to process and analyze data. It can improve overall efficiency and allow organizations to make more informed decisions.

2.5 Libraries in Machine Learning

- NumPy
- Scikit learn
- Pandas
- Matplotlib
- Seaborn
- Flask

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow use NumPy internally for the manipulation of Tensors.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit learns can also be used for data mining and data analysis, which makes it a great tool who is starting out with Machine Learning.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and a wide variety of tools for data analysis. It provides many inbuilt methods for groping, combining, and filtering data.

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named plot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar charts, etc.

Seaborn in python issued to create graphics which is easy to manage. Seaborn is a library provided by python, which basically helps to visualize the data and make it more and more undertakable by the user. With the help of the library, we can plot our data and make a graphical representation of it. Internally this library uses matplotlib; in short, it is based on matplotlib only. This also makes it efficient to create attractive and more informative graphical representations of our data. This library is integrated with the panda's data structure.

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on the WSGI toolkit the and jinja2 template engine. Flask is considered a micro-framework. Flask is a web framework, it's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, and template engine. It is a WSGI web app framework.

3. SYSTEM ANALYSIS

3.1 Importance of Machine Learning in Python

Machine Learning and Data Science are one of the fastest-growing technological fields. This field results in amazing changes in the medical field, production, robotics, etc. The main reason for the advancement in this field is the increase in the computational power and availability of large amounts of data. In Data Science, this data is analyzed and made suitable for creating machine learning models and products.

In today’s article, we are going to discuss the Water Quality Prediction. Based on some past observations, we’re predicting the pH and other parameters of the water. Through this project, you will get familiar with the exploratory data analysis and feature engineering techniques that need to be applied to process data.

3.2 Implementation of Machine Learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum.It is used for:

1. web development (server-side),
2. Software development,
3. Mathematics,
4. system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse, Flask, and Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as

opposed to other programming languages which often use semicolons or parentheses.
Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions, and classes. Other programming languages often use curly brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formulas. This made the processing time-consuming, tedious, and inefficient. But in the modern days, it is become very much easy and more efficient compared to the olden days by various Python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries.

3.3 Scope of the Project

Water Quality analysis is all about analyzing the data that is present in the data set and predicted from the attributes like pH, probability, etc. The algorithm used by them provided an accuracy of over 69.81% from the algorithm Extra Tree Classifier. We collected datasets from Kaggle.

3.4 Data Set Analysis

It contains 10 attributes in the water potability file which are used to predict the water quality. The Water Quality dataset is

1. pH value
2. Hardness
3. Solids
4. Chloramines
5. Sulfate
6. Conductivity
7. Organic carbon
8. Trihalomethanes
9. Turbidity
10. Potability

pH value:

The pH value of the water is a crucial metric fordeciding its acid-base stability.

Hardness:

Calcium and magnesium mariners are major contributors to rigidity. These mariners are dissolved from the environmental strata that thewater passes through.

Solids:

Water may break down into a variety of animate andinanimate minerals like potassium, calcium,sodium, chlorides, magnesium, sulfates, and so on.

Chloramines:

Chloramines In public water systems, the most common detergents are chlorine and chloramine.

Sulfate:

Sulfates are naturally occurring substances that are found in minerals, soil, and rocks. They are present in ambient air, groundwater, plants, and food.

Conductivity:

Pure water is not a good conductor of electric current rather it is a good insulator. An increase in ion concentration enhances the electrical conductivity of water.

Organic carbon:

Total Organic Carbon in source waters is obtained from naturally degraded the organic matter.

Trihalomethanes:

THMs are compounds that can form in the chlorine-treated resource.

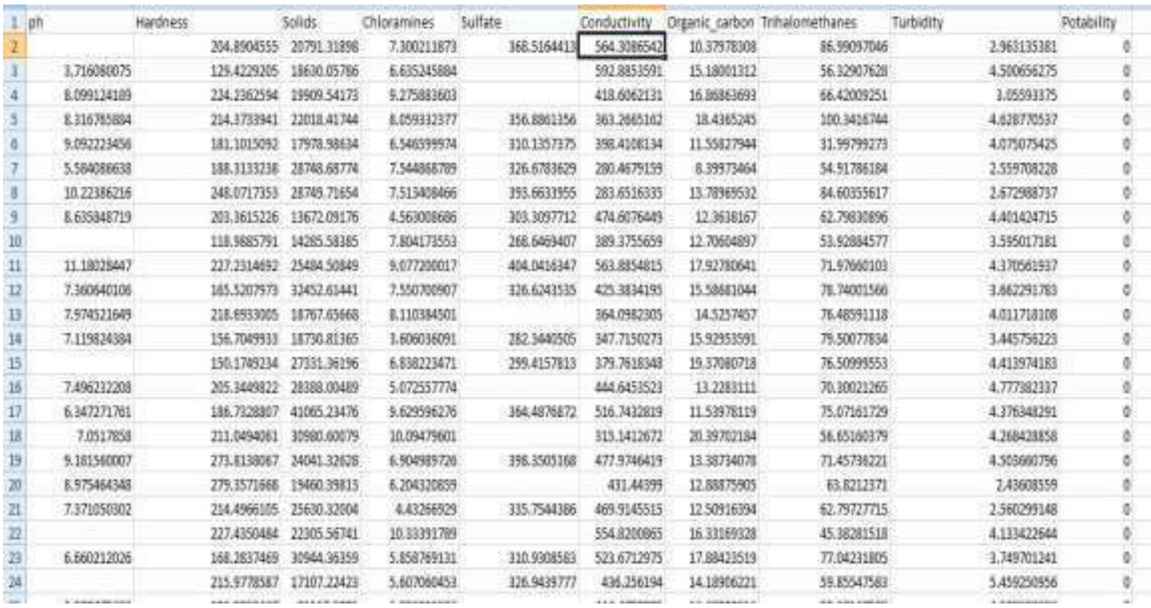
Turbidity:

The turbidity of the water is determined by theamount of suspended hard particles.

Potability:

Specifies if the water is safe for human utilization, with 1 indicating drinkable water and 0 indicating thatit is not for drinkable water.

Data Set:



1	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
2		204.8904555	20791.31898	7.300211873	368.5164413	564.3086542	10.37978308	86.99097046	2.963135381	0
3	3.716080075	129.4229205	18630.05786	6.635245884		592.8853591	15.18001312	56.32907628	4.500656275	0
4	8.099124189	224.2362594	19909.54173	9.275883603		418.6062131	16.86863693	66.42009251	3.05593375	0
5	8.316765884	214.3733941	22018.41744	8.059332377	356.8861356	363.2665162	18.4365245	100.3416744	4.628770537	0
6	9.092223456	181.1015092	17978.98634	6.546599974	310.1357375	398.4108134	11.55827944	31.99799273	4.075075425	0
7	5.584086638	188.3133238	28748.68774	7.544868789	326.6783629	280.4679159	8.39973464	54.91786184	2.559708228	0
8	10.22386216	248.0717353	28749.71654	7.513408466	393.6633955	283.6516335	13.78969532	84.60355617	2.672988737	0
9	8.635848719	203.3615226	13672.09176	4.563008686	303.3097712	474.6076449	12.3638167	62.79830896	4.401424715	0
10		118.9885791	14285.58385	7.804173553	268.6469407	389.3755659	12.70604897	53.92884577	3.595017181	0
11	11.18028447	227.2314692	25484.50849	9.077200017	404.0416347	563.8854815	17.92780641	71.97660103	4.370561937	0
12	7.360640106	165.5207973	32452.61441	7.550700907	326.6243535	425.3834195	15.58681044	78.74001566	3.662291783	0
13	7.974521649	218.6933005	18767.65668	8.110384501		364.0982305	14.5257457	76.48591118	4.011718108	0
14	7.119824384	156.7049933	18730.81365	3.606036091	282.3440505	347.7150273	15.92953591	79.50077834	3.445756223	0
15		150.1749234	27331.36196	6.838223471	299.4157813	379.7618348	19.37080718	76.50999553	4.413974183	0
16	7.496232208	205.3449822	28388.00489	5.072557774		444.6453523	13.2283111	70.30021265	4.777382337	0
17	6.347271761	186.7328807	41065.23476	9.629596276	364.4876872	516.7432819	11.53978119	75.07161729	4.376348291	0
18	7.0517858	211.0494061	30980.60079	10.09479601		315.1412672	20.39702184	56.65160379	4.268428858	0
19	9.181560007	273.8138067	24041.32628	6.904989726	398.3505168	477.9746419	13.38734078	71.45738221	4.503660796	0
20	8.975464348	279.3571666	19460.39813	6.204320859		431.44399	12.88875905	63.8212371	2.43608559	0
21	7.371050302	214.4966105	25630.32004	4.43266929	335.7544386	469.9145515	12.50916394	62.79727715	2.560299148	0
22		227.4350484	22305.56741	10.33391789		554.8200865	16.33169328	45.38281518	4.133422644	0
23	6.660212026	168.2837469	30944.36359	5.858769131	310.9308583	523.6712975	17.88423519	77.04231805	3.749701241	0
24		215.9778587	17107.22423	5.607060453	326.9439777	436.256194	14.18906221	59.85547583	5.459250956	0

Fig: 3.1 Dataset

Fig: 3.1 is the data set of Water Quality Prediction using machine learning which contains pH value, Hardness, Solids, Chloramines, Conductivity, Organic carbon, Trihalomethanes, Turbidity, and Potability.

3.5 Methodology

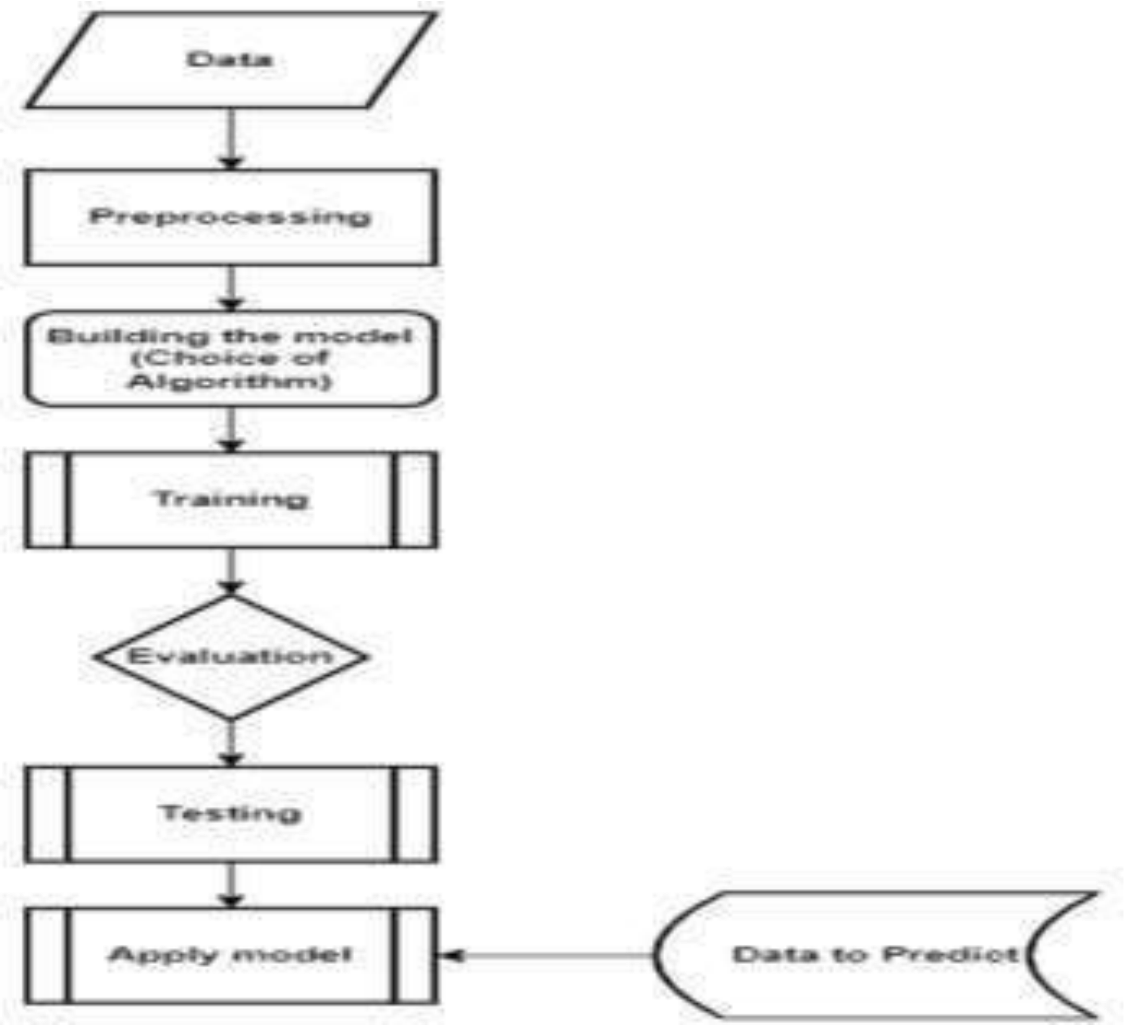
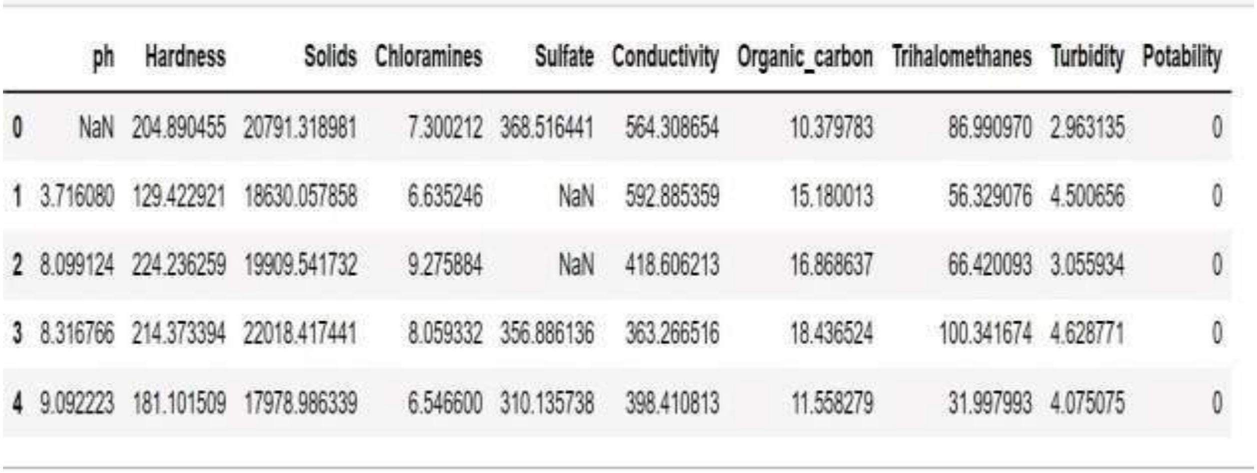


Fig: 3.2 Work-flow diagram of the proposal.

Fig: 2 depicts the proposed framework for water quality prediction. Firstly, we pre-process the datasets. In the pre-processing stage, correlation between attributes of the datasets is analyzed for finding useful features in detecting diabetes. After that, the data is divided into two sets: training and testing. The training set is utilized to develop predictive ML models using a variety of machine learning algorithms. Next, we assess the proposal's performance with respect to different metrics. Finally, the best ML model is deployed in a web application using flask. Following this, we describe the workflow of each part briefly:

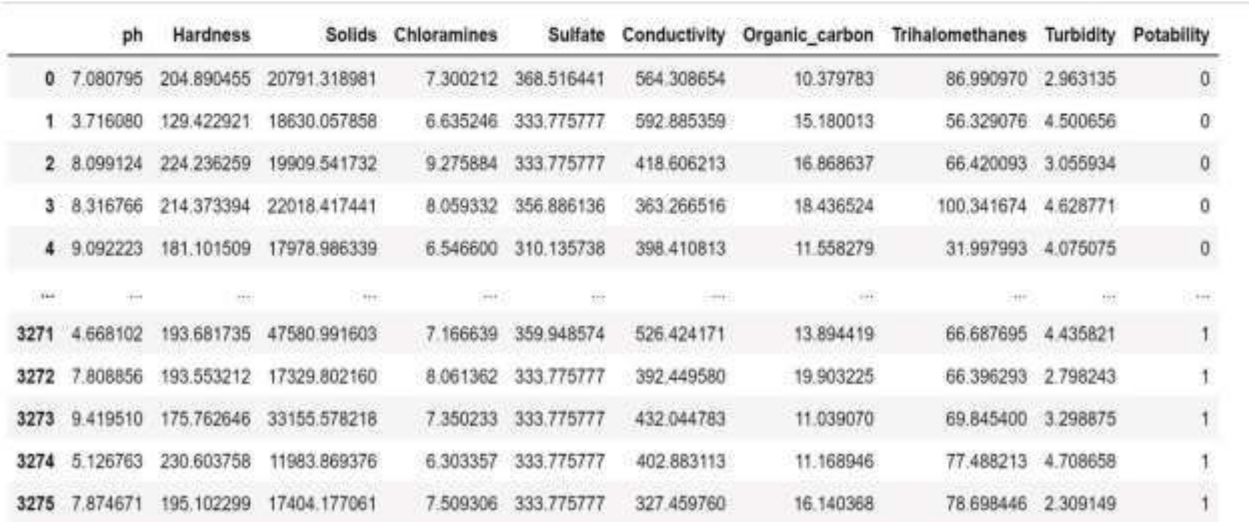
3.5.1 Missing Values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset are represented as ‘Nan’ but it is a non-standard missing value and it has to be converted into a standard missing value Nan. So that pandas can detect the missing values. Fig: 3.3 below is a heat map representing the missing values. We have filled the missing values with 0. Fig: 3.4 below is the heat map representing after filling in missing values.



	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

Fig: 3.3 Before missing data



	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
...	...	...	...	...	...	...	...	...	...	...
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

Fig: 3.4 After filling missing data

3.5.2 Correlation Coefficient Method

We can find dependency between two attributes p and q using the Correlation coefficient method using the formula.

rp, q= ∑(pi-p)(qi-q)/nσpσq

=∑(pi qi)-np q/ nσpσq

n is the total number of patterns, pi and qi are respective values of p and q attributes in patterns i, p and q are respective mean values of p and q attributes, σp , σq are respective standard deviations values of p and q attributes. Generally, -1≤ rp,q ≤ +1. If rp,q < 0, then p and q are negatively correlated. If rp,q =0, then p and q are independent attributes and there is no correlation between them. If rp,q > 0, then p and q are positively correlated. We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute. Fig:3.8.2 is the correlation heat map. After applying correlation the attributes are PR interval , QRS duration , QT interval , QTc interval, P wave , T wave , QRS wave and problem . The attribute Vent_rate got dropped.

To find the dependent variables and to predict hard-to-estimate variables through easily attainable parameters, we performed correlation analysis to extract the possible relationships between the parameters. We used the most commonly used and effective correlation method, known as the Pearson correlation. We applied the Pearson correlation on the raw values of the parameters listed in Table 4 and applied it after normalizing the values through q-value normalization as explained in the subsequent section. As the correlation chart in Table 4 indicates:

- Alkalinity (Alk) is highly correlated with hardness (CaCO3) and calcium (Ca). • Hardness is highly correlated with alkalinity and calcium, and loosely correlated with pH.
- Conductance is highly correlated with total dissolved solids, chlorides and fecal coliform count, and loosely correlated with calcium and temperature.
- Chlorides are highly correlated with conductance and TDS, and loosely correlated with temperature, calcium and fecal coliform.

Now that we have listed the correlation analysis observations, we find that our predicting parameter WQI is correlated with seven parameters, namely temperature, turbidity, pH, hardness as CaCO3, conductance, total dissolved solids and fecal coliform count. We have to choose the minimal number of parameters to predict the WQI, in order to lower the cost of the system. The three parameters whose sensors are easily available, cost the lowest and contribute distinctly to the WQI are temperature, turbidity and pH, which deems them naturally selected. The other convenient parameter is total dissolved solids, whose sensor is also easily available and is correlated with conductance and fecal coliform count, which means selecting TDS would allow us to discard the other two parameters. We leave the remaining inconvenient parameter, hardness as CaCO3, out because it is not highly correlated comparatively and is not easy to acquire.

To conclude the correlation analysis, we selected four parameters for the prediction of WQI, namely, temperature, turbidity, pH and total dissolved solids. We initially just considered the first three parameters, given their low cost, and if needed, TDS will be included later to analyse its contribution to the accuracy.

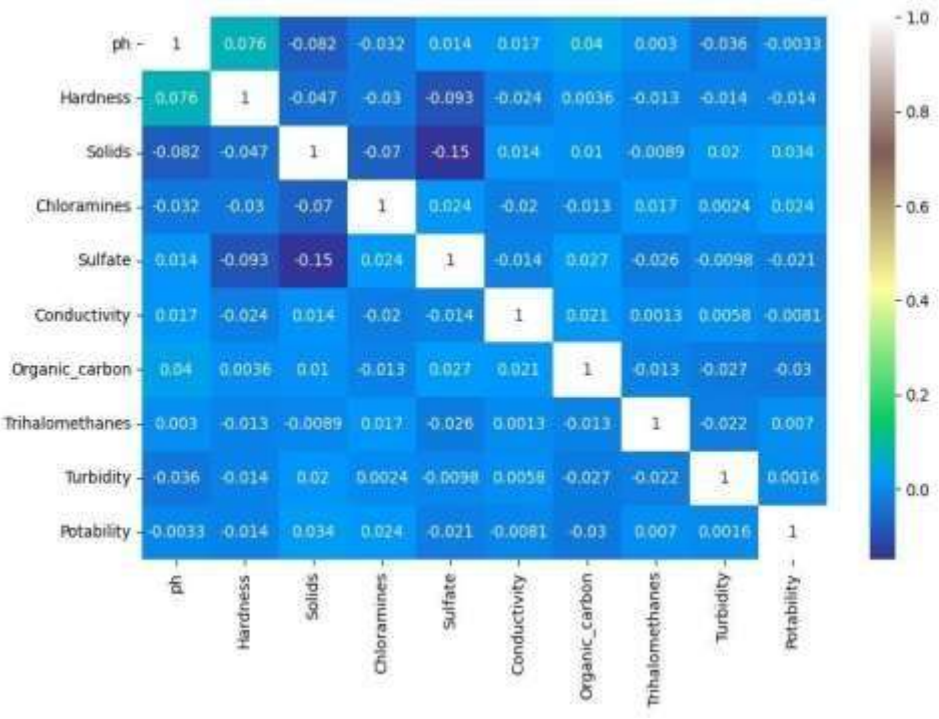


Fig: 3.5 Correlation

3.5.3 Cross Validation:

Cross-validation is a technique in which we train our model using the subset of the data- set and then evaluate using the complementary subset of the data set. The three steps involved in cross-validation are as follows:

- Reserve some portion of the sample data set.
- Using the rest data-set train the model.
- Test the model using the reserve portion of the data set.

The last step prior to applying the machine learning model is splitting the provided data in order to train the model, test it with a certain part of the data and compute the accuracy measures to establish the model's performance. This research explores the cross-validation data-splitting technique. Cross-validation splits the data into k subsets and iterates over all the subsets, considering k-1 subsets as the training dataset and 1 subset as the testing dataset. This ensures an efficient split and use of proper and definitive data for training and testing. This is generally computationally expensive, given the iterations, but our research uses a small dataset, which is mostly the case with water quality datasets, making cross-validation more suited for this problem.

3.6 Classification

It is a process of categorizing data into given classes. Its primary goal is to identify the class of our new data.

3.6.1 Machine Learning Algorithms for Classification

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Random Forest, Logistic Regression, Support vector machine, etc.

1. Random Forest:

A random forest is a machine-learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The random forest algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or means of the output from various trees. Increasing the number of trees increases the precision of the outcome.

2. Extra Trees Classifier (ETC):

It is a type of ensemble learning technique that aggregates the results of multiple decorrelated decision trees collected in a "forest" to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature set from which each decision tree must select the best feature to split the data based on some mathematical criteria. This random sample of features leads to the creation of multiple de-correlated decision trees.

3. Support vector machine:

Support vector machine is a linear model for classification and regression problems. It is a supervised machine learning algorithm. It can solve linear and non-linear problems and work well many practical problems. The idea of support vector machine is simple: The algorithm creates a line or hyperplane which separates the data into classes. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high dimensional feature spaces.

4. Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

3.7 Implementation Code

```
   import pandas as pd
   import matplotlib.pyplot as plt
   import seaborn as sns
   import pickle
   data = pd.read_csv(r'D:\Datasets\water_potability.csv) data
   data.shape
   data.head()
   data.info()
   data.isnull().sum()
   data.describe()
   data.fillna(data.mean(),inplace=True)
               data
   data.isnull().sum()

   sns.heatmap(data.corr(),annot=True,cmap='terrain')
   fig=plt.gcf()
    fig.set_size_inches(10,6)
   plt.show()

   data.boxplot(figsize=(16,6))
   plt.show()

   data['Solids'].describe()
   data['Potability'].value_counts()
   data.Potability.value_counts().plot(kind="bar", color=["brown", "salmon"])
   plt.show()

   data.hist(figsize=(14,12))
   plt.show()

   sns.barplot(x=data['ph'],y=data['Hardness'],hue=data['Potability'])
    plt.show()

   sns.scatterplot(x=data['ph'],y=data['Potability'])
   plt.show()
X = data.drop('Potability', axis=1)
Y = data['Potability']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2, shuffle=True,
random_state=0)
X_train
Y_train

from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(criterion= 'entropy', min_samples_split= 9, splitter='best')
dt.fit(X_train,Y_train)
X_test
Y_test
Y_prediction = dt.predict(X_test)

from sklearn.metrics import accuracy_score, confusion_matrix
accuracy_score(Y_prediction,Y_test) * 100
confusion_matrix(Y_prediction, Y_test)

res =
dt.predict([[7.080795,210.732854,13671.416030,8.546187,418.470551,352.252328,1
0.353659,45.304007,3.364891]])[0]
res
Y_test.shape

from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RepeatedStratifiedKFold
dt = DecisionTreeClassifier() criterion = ["gini","entropy"] splitter  = ["best","random"]
min_samples_split=range(1,10)
parameters = dict(criterion=criterion, splitter= splitter,
min_samples_split=min_samples_split)
cv= RepeatedStratifiedKFold(n_splits=5, random_state=101) grid_search_cv_dt
= GridSearchCV(estimator=dt, param_grid = parameters,scoring='accuracy',
cv=cv)

grid_search_cv_dt.fit(X_train,Y_train)
print(grid_search_cv_dt.best_params_)
prediction_grid = grid_search_cv_dt.predict(X_test)
accuracy_score(prediction_grid,Y_test) *100
confusion_matrix(Y_test,prediction_grid)

from sklearn.ensemble import RandomForestRegressor
```

```
regressor = RandomForestRegressor(n_estimators=100, random_state=42)
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)
from sklearn import metrics
metrics.r2_score(Y_test, Y_pred) * 100
regressor.predict([[7.080795,210.732854,13671.416030,8.546187,418.470551,352.252328,10.353659,45.304007,3.364891]])

from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(metric='manhattan',n_neighbors=22) knn.fit(X_train,Y_train)
prediction_knn=knn.predict(X_test)
accuracy_knn=accuracy_score(Y_test,prediction_knn)*100
print('accuracy_score score : ',accuracy_score(Y_test,prediction_knn)*100,'%')

from numpy import loadtxt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# fit model no training data
model = XGBClassifier()
model.fit(X_train, Y_train)
print(model)
# make predictions for test data
y_pre = model.predict(X_test)
predictions = [round(value) for value in y_pre]
# evaluate predictions
accuracy = accuracy_score(Y_test, predictions)
print("Accuracy: %.2f%%" % (accuracy * 100.0))

from sklearn.ensemble import AdaBoostClassifier
abc=AdaBoostClassifier(n_estimators=100,learning_rate=1)
model=abc.fit(X_train,Y_train)
prediction_abc=model.predict(X_test)
print("Accuracy:",metrics.accuracy_score(Y_test, prediction_abc)* 100, '%')
```

```
from sklearn.ensemble import ExtraTreesClassifier
etc=ExtraTreesClassifier(n_estimators=900,random_state=1)
model_etc=etc.fit(X_train,Y_train)
prediction_etc=model_etc.predict(X_test)
print("Accuracy:",metrics.accuracy_score(Y_test, prediction_etc)* 100, '%')

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=1000, random_state=42)
rf.fit(X_train, Y_train)
prediction_rf = rf.predict(X_test)


from sklearn.metrics import classification_report rand_score
= rf.score(X_test, Y_test)
classification_report_rf = classification_report(Y_test, prediction_rf)
print("Accuracy score:", rand_score * 100)

from sklearn import svm, datasets
sm = svm.SVC(kernel='linear', C=1).fit(X_train, Y_train)
sm_predict = sm.predict(X_test)
print(accuracy_score(Y_test, sm_predict)* 100)

from sklearn.ensemble import ExtraTreesClassifier
etc=ExtraTreesClassifier(n_estimators=900,random_state=1)
model_etc=etc.fit(X_train,Y_train)
prediction_etc=model_etc.predict(X_test)
print ("Accuracy:",metrics.accuracy_score(Y_test, prediction_etc)* 100, '%')
```

Predict.html

```
<!DOCTYPE html>
<html>
<head>
    <title>water quality prediction</title>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-
awesome.min.css">
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="style.css">


</head>
    <body>
        <div class="background-image">
            <h1 style="text-align:center; font-size:60px; color: rgb(43, 92, 226);
textdecoration:underline; font-family: 'Josefin Sans', sans-serif;">Water Quality
Prediction</h1>
    <form action="/login" class="loginbox"
method="post"><br><br><br><br><br><br><br><br>
        <center>
            <input type="text" name="ph" placeholder="Enter pH value"
required="required" /> &nbsp; &nbsp; &nbsp; &nbsp;
            <input type="text" name="Hardness" placeholder="Enter Hardness"
required="required" /><br><br>
            <input type="text" name="Solids" placeholder="Enter Solids"
required="required" />&nbsp; &nbsp; &nbsp; &nbsp;
            <input type="text" name="Chloramines" placeholder="Enter Chloramines"
required="required" /><br><br>
            <input type="text" name="Sulfate" placeholder="Enter Sulfate"
required="required" />&nbsp; &nbsp; &nbsp; &nbsp;
            <input type="text" name="Conductivity" placeholder="Enter Conductivity"
required="required" /><br><br>
```

```
            <input type="text" name="Organic_carbon" placeholder="Enter
Organic_carbon" required="required" />&nbsp; &nbsp; &nbsp; &nbsp;
            <input type="text" name="Trihalomethanes" placeholder="Enter
Trihalomethanes" required="required" /><br><br>
            <input type="text" name="Turbidity" placeholder="Enter Turbidity"
required="required" /><br><br>
            <button type="submit" class="btn btn-light btn-outline-success"
id="sendMessageButton">Predict</button>

        </center>
    </form>
    <h1 style="color: rgb(51, 0, 255); font-family: 'Josefin Sans',
sansserif;">{{showcase}}</h1>
        </div>
    </body>

</html>
```

Style.css

```
      * {
    margin: 0;
    padding: 0;
  }
  .background-image {
        background-image: url('./testing-water-quality.jpg');
        background-size: cover;
        background-repeat: no-repeat;
        height: 100vh;
    }
```

App.py

```
from flask import Flask, request, render_template
import pickle
import pandas as pd
import numpy as np
app = Flask(_name_)
classifier=pickle.load(open('model.pkl','rb'))
@app.route("/")
def home():
    return render_template("front.html")

@app.route("/predict", methods = ['POST','GET'])
def predict():
    print(request.form)
    int_features=[int(x) for x in request.form.values()]
    final=[np.array(int_features)]
    print(int_features)
    prediction=classifier.predict_proba(final)
    output = '{0:.{1}f}'.format(prediction[0][1], 2)
    if output>str(0.5):
        return render_template('front.html',pred='Water is safe to drink'.format(output))
    else:
        return render_template('front.html',pred='water is unsafe to drink'.format(output))
if __name___== "_main_":
    app.run(debug=True)
```

Screen Shots

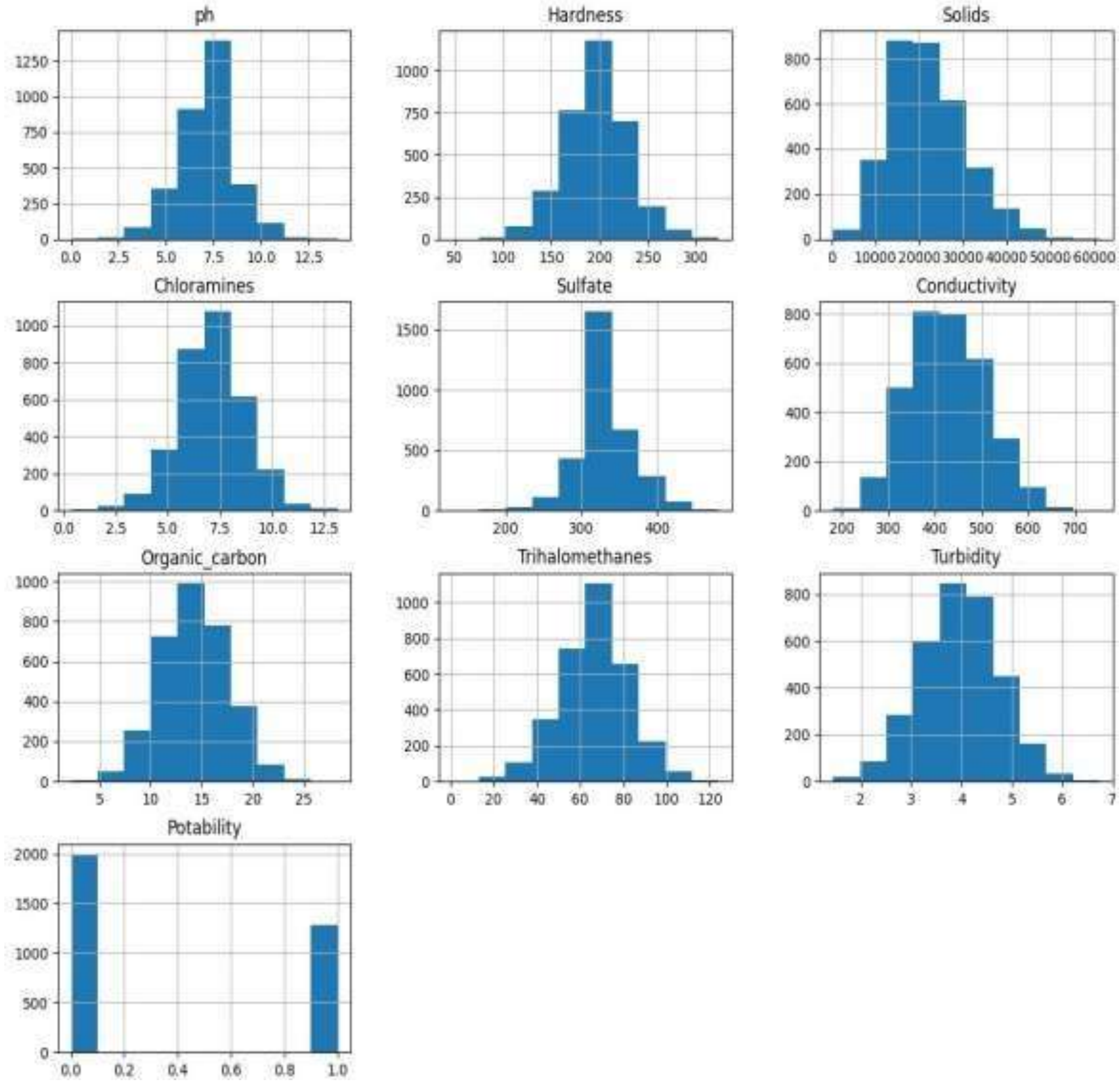


Fig: 3.6 Histogram

Visualize all the features of the data set in the above Fig: 3.6. Visualize the features' correlation using a seaborn heat map function. but you can see in the below heatmap that there is no correlation between any feature; it means that we can’t reduce the dimension.

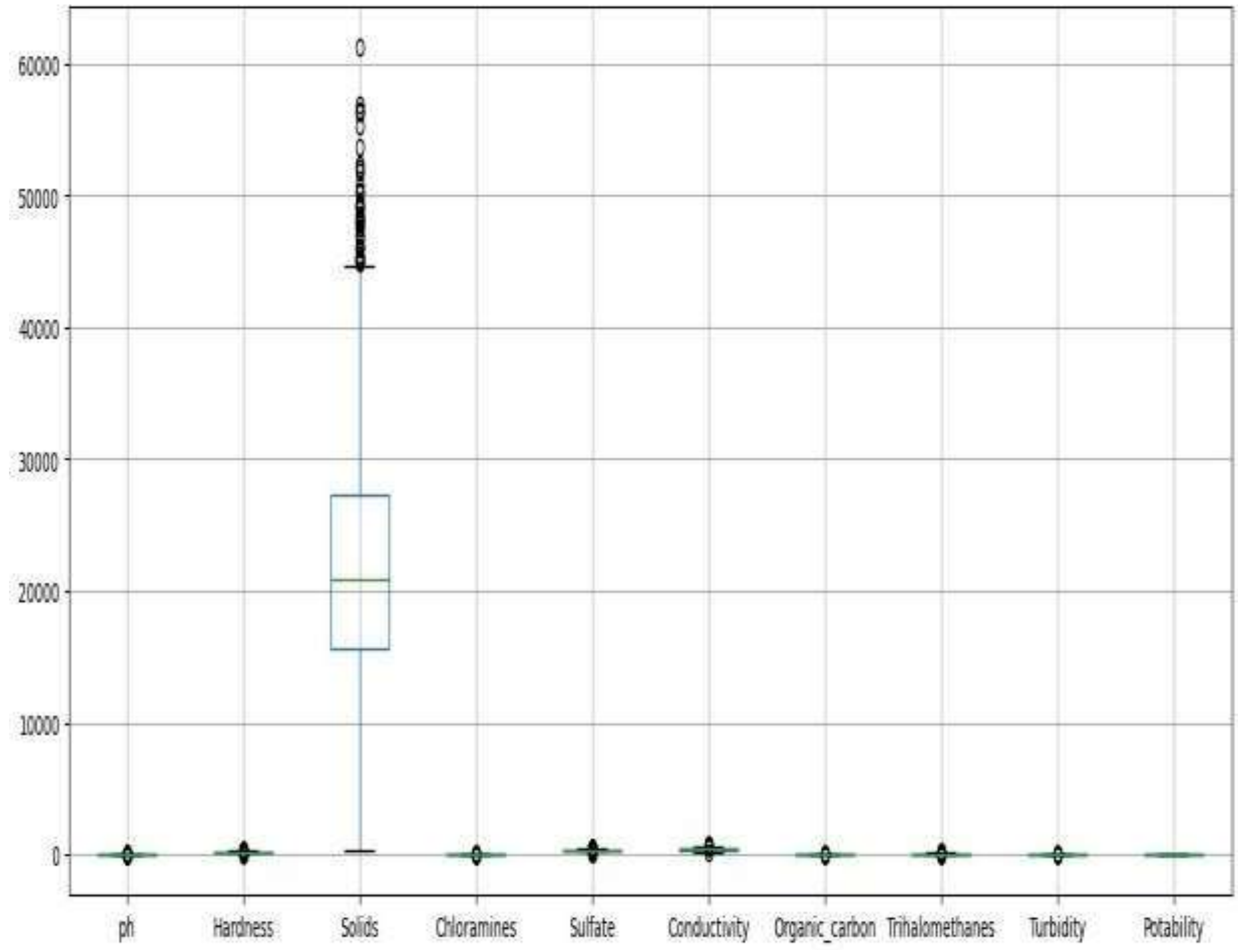


Fig: 3.7 Boxplot

The above figure tells about the removal of outliers using the boxplot function.

Now see the outlier using a boxplot function. So, you can see that the Solid feature contains outliers but we can't remove the outliers from it because if we remove the outliers from the Solid feature. So, water will be safe to drink every time. It contains an outlier to make the water impure which means it will tell us whether water is safe or not. Solid may be high to make the water unsafe to drink.

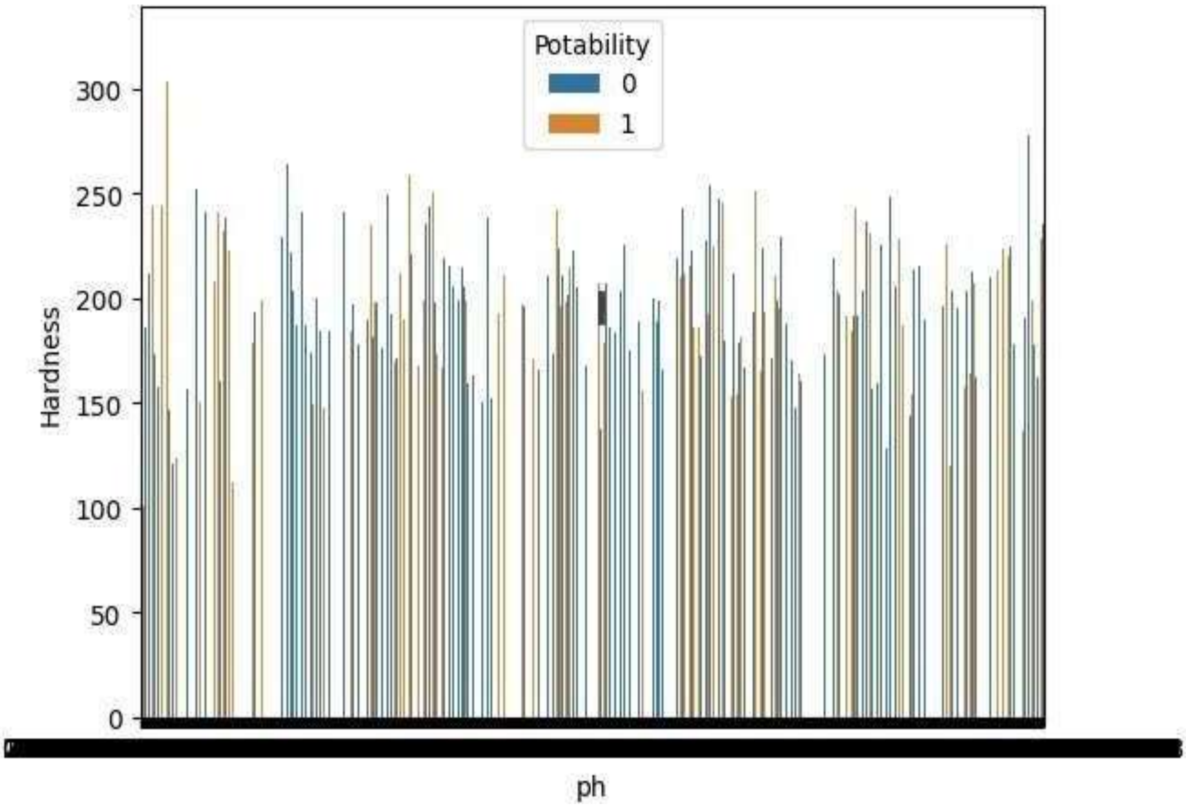


Fig: 3.8 Bar plot

The above Fig: 3.8 shows the bar plot between the two attributes named pH value and hardness values. Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable. (0) Water is not safe to drink and (1) Water is safe to drink.

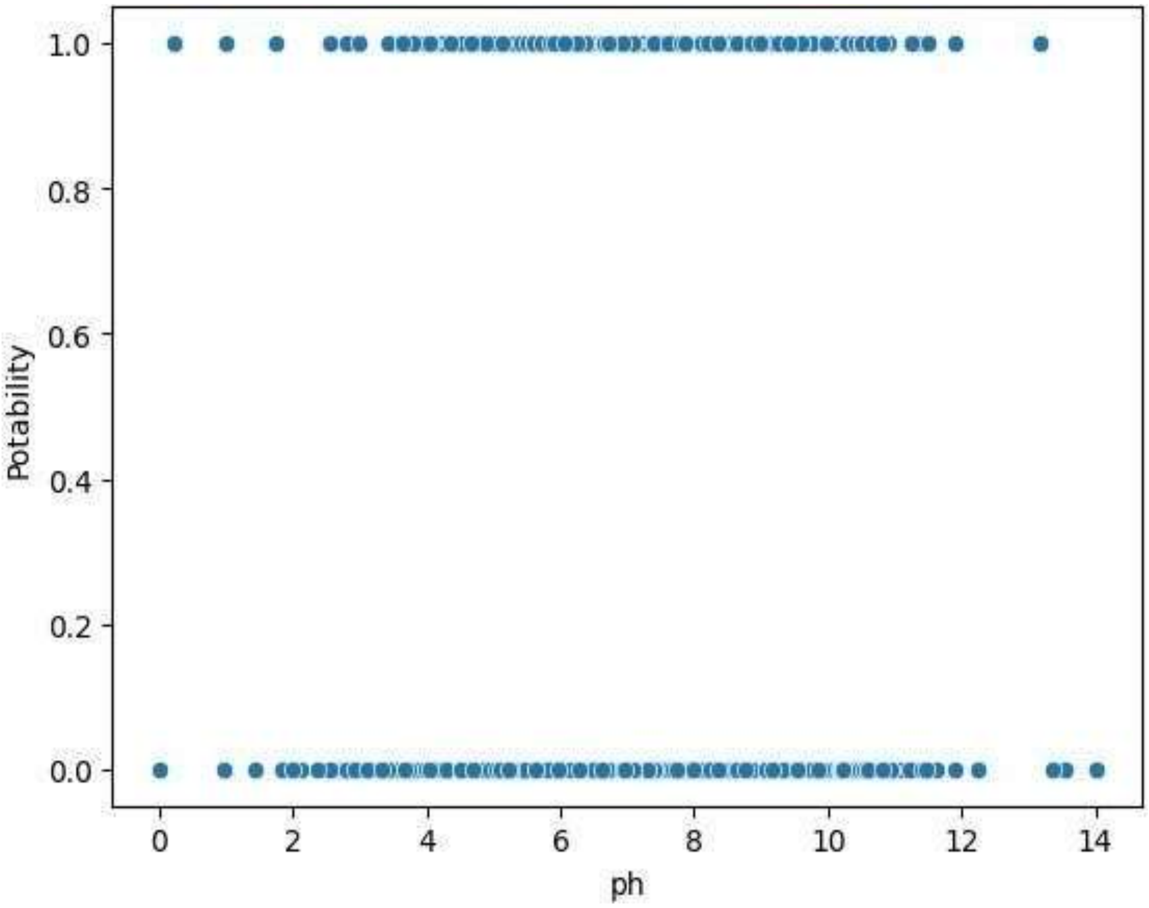


Fig: 3.9 Scatter plot

Now we find the scatter plot by using two attributes named pH and potability. Now see the above Fig: 3.9 explains about the pH value is in the range of 0 to 13 which means its potability value is 1, which means the water is used to drink. If the pH is greater than 13 its potability value in which means the water is unsafe to drink.

3.8 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. A **true positive** (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class. A **false positive** (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

4. OUTPUT SCREENS

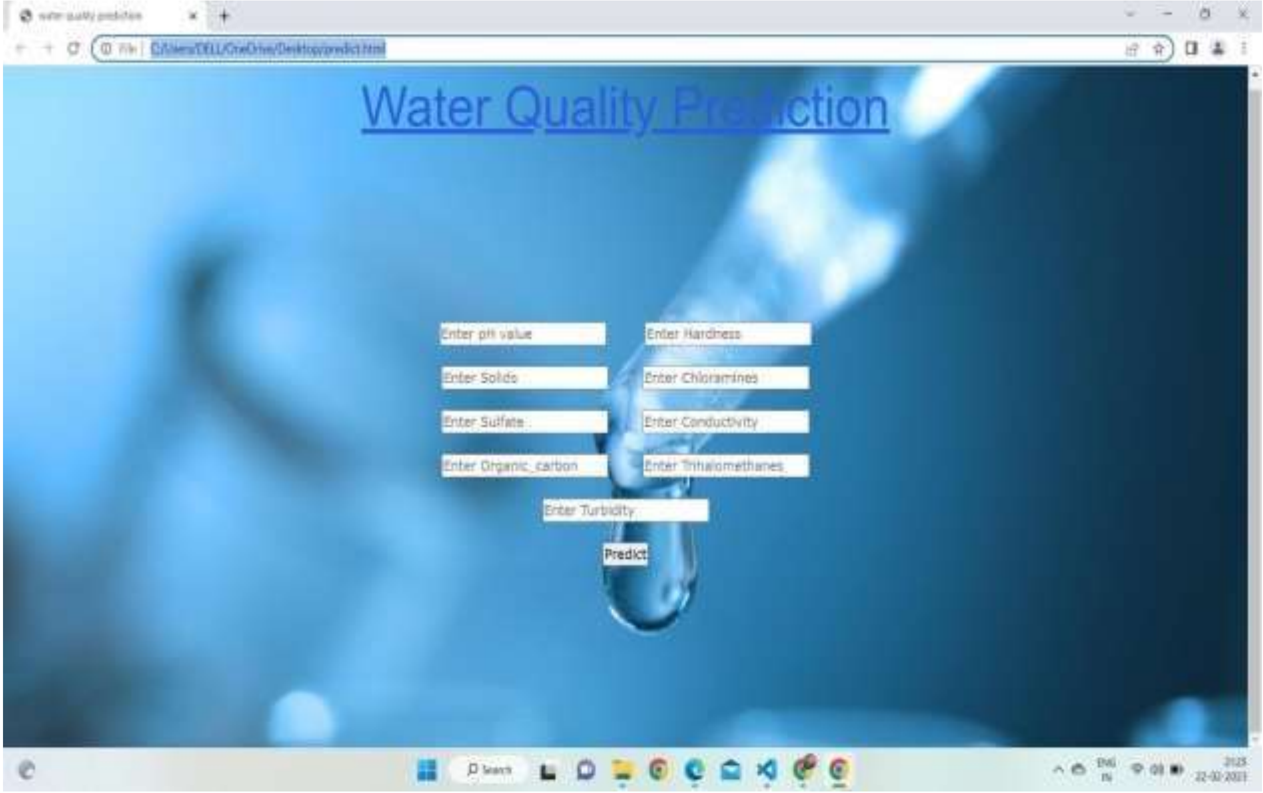
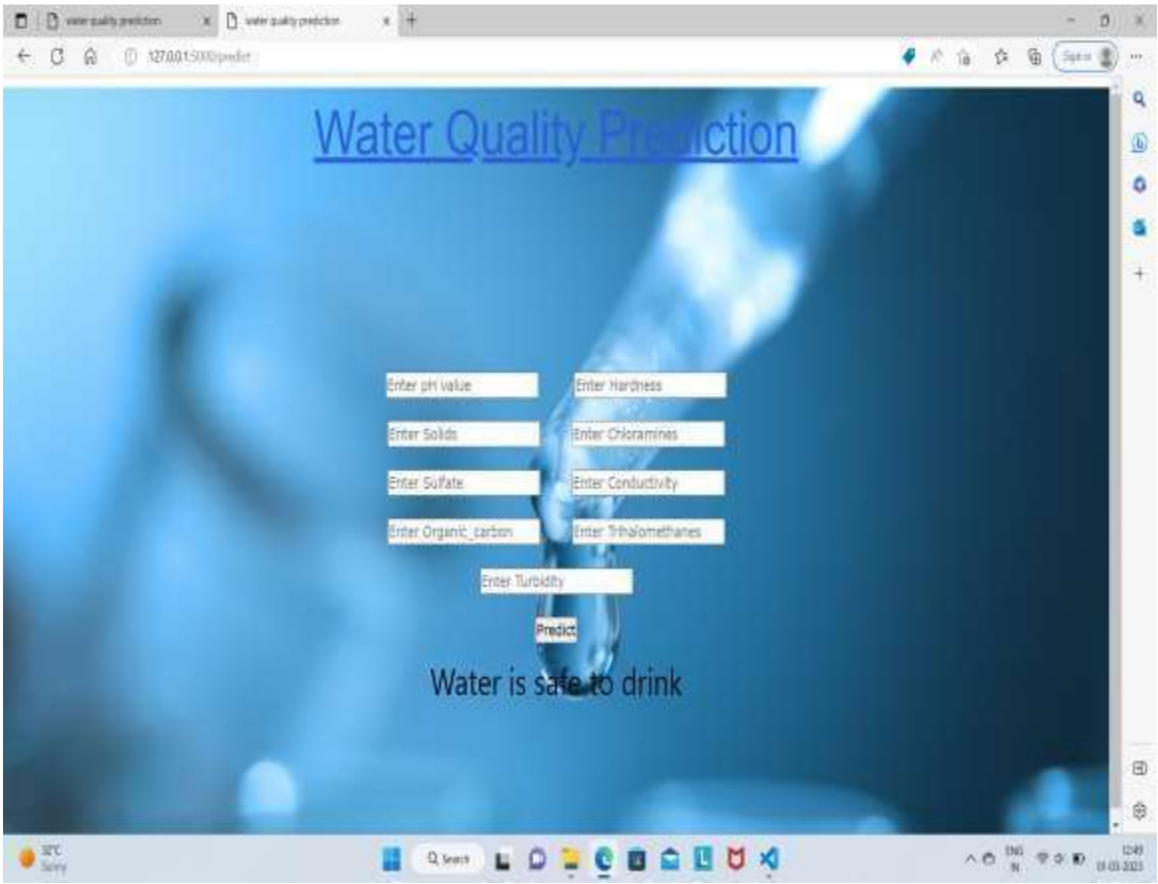


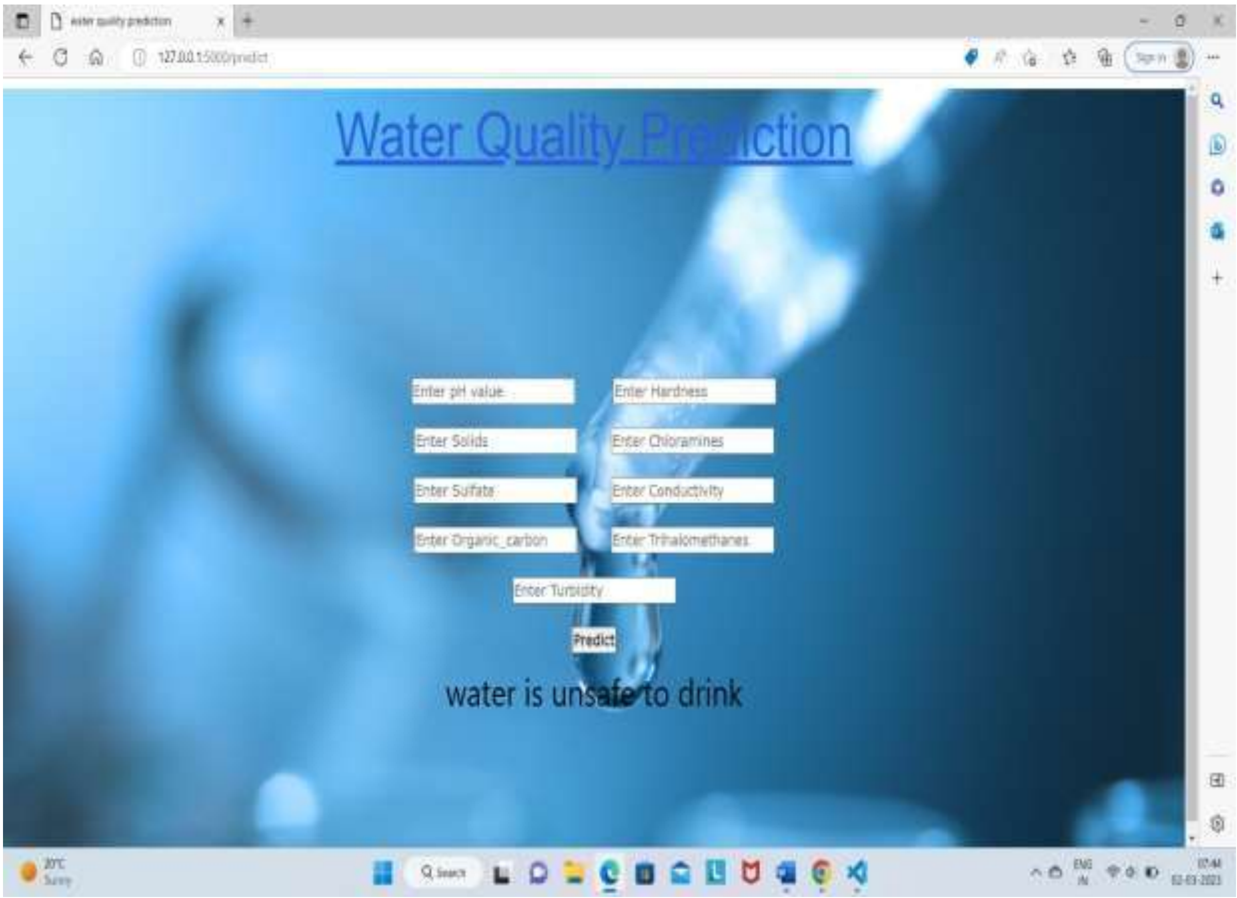
Fig: 4.1 Water Quality Prediction



Water Quality Prediction

Water is safe to drink

Fig: 4.2 Water is safe



water is unsafe to drink

Fig: 4.3 Water is unsafe

5. CONCLUSION

Water is one of the most essential resources for survival and its quality is determined through WQI. Conventionally, to test water quality, one has to go through expensive and cumbersome lab analysis. This research explored an alternative method of machine learning to predict water quality using minimal and easily available water quality parameters. The data used to conduct the study were acquired from PCRWR and contained 663 samples from 12 different sources in Rawal Lake, Pakistan. A set of representative supervised machine learning algorithms was employed to estimate WQI.

We used 4 algorithms Random Forest, Extra Tree classifier, Support Vector Machine, Decision Tree in order to predict whether water is used to drink or not. The accuracy varies for different algorithms. The accuracy for Random Forest algorithm is 69.66%. The accuracy of Decision Tree algorithm is 62.04 % and Support Vector Machine is 62.80%. The highest accuracy for Extra Tree Classifier using is 69.81%.

6. FUTURE SCOPE

Water quality prediction using machine learning is a rapidly developing area of research that has great potential for addressing the challenges of ensuring clean water supplies. With the advancement in machine learning techniques, it is now possible to make accurate predictions of water quality parameters such as pH, dissolved oxygen, total dissolved solids, and other pollutants.

Improved accuracy: Machine learning algorithms can be trained with large amounts of data to improve the accuracy of water quality predictions. As more data becomes available, the accuracy of these predictions is likely to increase. Machine learning algorithms can be used to develop predictive models that can be used to detect changes in water quality in real time. Early warning systems: Machine learning algorithms can be used to develop early warning systems for water quality issues. These systems can detect changes in water quality parameters before they become critical, allowing water management authorities to take proactive measures to address the issue.

Overall, the future scope for water quality prediction using machine learning is vast, and it has the potential to revolutionize the way we manage our water resources.

7. REFERENCES

1. PCRWR. National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006); Pakistan Council of Research in Water Resources Islamabad: Islamabad, Pakistan, 2007.
2. PCRWR. Water Quality of Filtration Plants, Monitoring Report; PCRWR: Islamabad, Pakistan, 2010. Available online: http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/FILTRTAION%20PLANTS% 20REPOT-CDA.pdf (accessed on 23 August 2019)
3. Gazzaz, N.M.; Yusoff, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F. Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors. Mar. Pollut. Bull. 2012, 64, 2409–2420.
4. Daud, M.K.; Nafees, M.; Ali, S.; Rizwan, M.; Bajwa, R.A.; Shakoor, M.B.; Arshad, M.U.; Chatha, S.A.S.; Deeba, F.; Murad, W.; et al. Drinking water quality status and contamination in Pakistan. BioMed Res. Int. 2017, 2017, 7908183.
5. Alamgir, A.; Khan, M.N.A.; Hany; Shaukat, S.S.; Mehmood, K.; Ahmed, A.; Ali, S.J.;
6. Ahmed, S. Public health quality of drinking water supply in Orangi town, Karachi, Pakistan. Bull. Environ. Pharmacol. Life Sci. 2015, 4, 88–94.
7. Shafi, U.; Mumtaz, R.; Anwar, H.; Qamar, A.M.; Khurshid, H. Surface Water Pollution Detection using the Internet of Things. In Proceedings of the 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 8–10 October 2018; pp. 92–96.