

**WATER QUALITY PREDICTION USING
CATBOOST CLASSIFIER ALGORITHM**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

J. ANUBAMA (Reg.No – 39120010)
A. YOGESHWARI (Reg.No – 39120140)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

MAY- 2023

WATER QUALITY PREDICTION USING CATBOOST CLASSIFIER ALGORITHM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

**J. ANUBAMA (Reg.No – 39120010)
A. YOGESHWARI (Reg.No – 39120140)**



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

MAY- 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

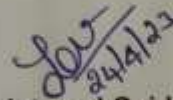
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

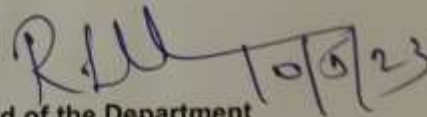
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **J. Anubama** (Reg.No – 39120010) and **A. Yogeshwari** (Reg.No – 39120140) who carried out the Project Phase-2 entitled "**WATER QUALITY PREDICTION USING CATBOOST CLASSIFIER ALGORITHM**" under my supervision from January 2023 to May 2023.


24/4/23

Internal Guide

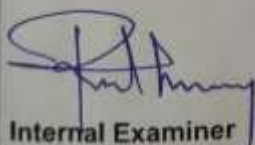
Mrs. J. MERLIN MARY JENITHA, M.Tech.,

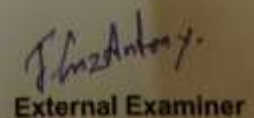

10/5/23

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.

Submitted for Viva voce Examination held on 10/05/2023


Internal Examiner


External Examiner

DECLARATION

I, **J. Anubama (Reg.No- 39120010)**, hereby declare that the Project Phase-2 Report entitled "**WATER QUALITY PREDICTION USING CATBOOST CLASSIFIER ALGORITHM**" done by me under the guidance of **Mrs. J. Merlin Mary Jenitha, M.Tech.**, is submitted in partial fulfillment of the requirements for the award Bachelor of Technology degree in Information Technology.

DATE: 10/05/2023

PLACE: Chennai

J. Anubama

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. R. Subhashini M.E., Ph.D.**, Head of the Department ,Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. J. Merlin Mary Jenitha, M.Tech.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTARCT

Human existence depends on water, so its purity must be assessed for improved consumption. The goal is to use a cat-boost algorithm to forecast the water quality findings as accurately as possible. It is anticipated that improving water quality will lessen health-related issues. By gathering a variety of data, including variable detection and analysis methods like univariate, bivariate, and multivariate analysis, the study of the water quality is carried out. The study compares and discusses how different algorithms function. Data gathering is the first step in the process, during which historical information about the water condition is gathered. With both dependent and independent factors, data analysis is performed. The main objective is to identify the physicochemical characteristics of water, including temperature, pH, EC, hardness, chlorides, alkalinity, phosphate, and sulfate in water samples taken from various monitoring sites. To forecast the trend of data and the quality of the water, the appropriate algorithm is used. The structure has logins for users and government officers, with the latter having access to submit complaints to officials. In order for the police to quickly locate the user, the user is also permitted to provide information about the water purity in addition to their residential data.

TABLE OF CONTENTS

CHAPTER NO	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF TABLES	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
	2.1 Inferences from Literature Survey	2
	2.2 Open Problems in Existing System	4
3	REQUIREMENTS ANALYSIS	5
	3.1 Feasibility Studies/Risk Analysis of the Project	5
	3.2 Software Requirements Specification Document	5
	3.3 System Use case	6
4	DESCRIPTION OF PROPOSED SYSTEM	7
	4.1 Selected Methodology or process model	7
	4.2 Architecture / Overall Design of Proposed System	7
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	8
5	IMPLEMENTATION DETAILS	16
	5.1 Development and Deployment Setup	16
	5.2 Algorithms	24
6	RESULTS AND DISCUSSION	29
7	CONCLUSION	30
	7.1 Conclusion	30
	7.2 Future work	30

REFERENCES	31
APPENDIX	37
A. SOURCE CODE	37
B. SCREENSHOTS	48
C. RESEARCH PAPER	52

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	Use Case Diagram for Water Quality Prediction	6
4.1	Work Flow Diagram for Water Quality Prediction	8
4.2	Class Diagram for Water Quality Prediction	9
4.3	Entity Diagram for water Quality prediction	9
4.4	System Architecture	10
4.5	Anaconda Navigator	12
5.1	Data Analysis	17
5.2	Data Classification	17
5.3	Data configuration	18
5.4	Data Preprocessing Modular Diagram	18
5.5	Portable Drinking water Analysis from Data optimized	19
5.6	Data visualization for Sulfate	20
5.7	Plotted chart analysis for Conductivity	20
5.8	Portable water range	21

5.9	Analysis of Turbidity in water bodies	21
5.10	Confusion Matrix for the data provided	22
5.11	Module Diagram for Data Visualization	24
5.12	Module Diagram for Algorithm Implementation	25
5.13	Implementation of Catboost Classifier Algorithm	26
5.14	Implementation of Logistic Regression	27
5.15	Implementation of Random Forest Algorithm	28
5.16	Accuracy of Naïve Baye's Algorithm	29
6.1	Comparison of Algorithms used for training data set	30

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE.NO
5.1	DATA ANALYSIS	17

CHAPTER 1

INTRODUCTION

Water quality straightforwardly affects general well-being and the climate. Water is utilized for different practices, like drinking, horticulture, and industry. As of late, the improvement of water sports and amusement has incredibly assisted with drawing in vacationers. Among different wellsprings of water supply, because of simple access, streams have been utilized all the more regularly for the advancement of human social orders. Utilizing other water assets, for example, groundwater and seawater in some cases helped with issues. For instance, utilizing groundwater without a reasonable re-energize will prompt land subsidence, and utilizing seawater is normally connected with contamination transmission. Thusly, the utilization of waterways has stood out. A few examinations connected with waterways all over the planet have been directed and a field of designing named stream designing has been proposed. In-stream designing concentrates on morphological changes, silt transport, water quality, and contamination transmission systems are vital. Stream structure, residue transport, and morphology of waterways are explored in the power through the pressure of waterways in waterway design. Assessment of water quality is an essential stage for the improvement of farming undertakings as far as an assurance of trimming design, kind of water system framework, and frameworks of water refinement for businesses.

CHAPTER 2

LITERATURE REVIEW

2.1 Review of Literature Survey

Md. Saikat Islam Khan, et al.[12] approach which estimates the water quality has been one of the significant challenges faced by the world in recent decades. This paper presents a water quality prediction model utilizing the principal component regression technique. 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 for the PCA output. Finally, the Gradient Boosting Classifier is utilized to classify the water quality status. The proposed system is experimentally evaluated on a Gulshan Lake-related dataset. The results demonstrate 95% prediction accuracy for the principal component regression method and 100% classification accuracy for the Gradient Boosting Classifier method, which show credible performance compared with the state-of-artmodels.

Theyazn H. H Aldhyani et al.[19],During last years, water quality has been threatened by various pollutants. Therefore, modeling and predicting water quality have become very important in controlling water pollution. In this work, advanced artificial intelligence (AI) algorithms are developed to predict water quality index (WQI) and water quality classification (WQC). For the WQI prediction, artificial neural network models, namely nonlinear autoregressive neural network (NARNET) and long short-term memory (LSTM) deep learning algorithm, have been developed. In addition, three machine learning algorithms, namely, support vector machine (SVM), K-nearest neighbor (K-NN), and Naive Bayes, have been used for the WQC forecasting. The used dataset has 7 significant parameters, and the developed models were evaluated based on some statistical parameters. The results revealed that the proposed models can accurately predict WQI and classify the water quality according to superior robustness. Prediction results demonstrated that the NARNET model performed slightly better than the LSTM for the prediction of the WQI values and the SVM algorithm has achieved the highest accuracy (97.01%) for the WQC prediction. Furthermore, the NARNET and LSTM models have achieved similar

accuracy for the testing phase with a slight difference in the regression coefficient (RNARNET = 96:17% and RLSTM = 94:21%). This kind of promising research can contribute significantly to water management.

Jian Zhou et al. [9] Water quality prediction has great significance for water environment protection. A water quality prediction method based on the Improved Grey Relational Analysis (IGRA) algorithm and a Long-Short Term Memory (LSTM) neural network is proposed in this paper. Firstly, considering the multivariate correlation of water quality information, IGRA, in terms of similarity and proximity, is proposed to make a feature selection for water quality information. Secondly, considering the time sequence of water quality information, the water quality prediction model based on LSTM, whose inputs are the features obtained by IGRA, is established. Finally, the proposed method is applied to two actual water quality datasets: Tai Lake and Victoria Bay. Experimental results demonstrate that proposed method can take full advantage of the multivariate correlations and time sequence of water quality information to achieve better performance on water quality prediction compared with the single feature or non-sequential prediction methods.

Dao Nguyen Khoi, e tal .[3] is the effective management of water quantity and quality, it is absolutely essential to estimate the pollution level of the existing surface water. This case study aims to evaluate the performance of twelve machine learning (ML) models, including five boosting-based algorithms (adaptive boosting, gradient boosting, histogram-based gradient boosting, light gradient boosting, and extreme gradient boosting), three decision tree-based algorithms (decision tree, extra trees, and random forest), and four ANN-based algorithms (multilayer perceptron, radial basis function, deep feed-forward neural network, and convolutional neural network), in estimating the surface water quality of the La Buong River in Vietnam. Water quality data at four monitoring stations alongside the La Buong River for the period 2010–2017 were utilized to calculate the water quality index (WQI). Prediction performance of the ML models was evaluated by using two efficiency statistics (i.e., R^2 and RMSE). The results indicated that all twelve ML models have good performance in predicting the WQI but that extreme gradient boosting (XGBoost) has the best performance with the highest accuracy ($R^2 = 0.989$ and $RMSE = 0.107$). The findings strengthen the argument that ML models, especially XGBoost, may be employed for WQI prediction with a high level of accuracy, which

will further improve water quality management. Amir Hamzeh Haghiabi, et al. [2] investigates the performance of artificial intelligence techniques including artificial neural network (ANN), group method of data handling (GMDH) and support vector machine (SVM) for predicting water quality components of Tیره River located in the southwest of Iran. To develop the ANN and SVM, different types of transfer and kernel functions were tested, respectively. Reviewing the results of ANN and SVM indicated that both models have suitable performance for predicting water quality components. During the process of development of ANN and SVM, it was found that tansig and RBF as transfer and kernel functions have the best performance among the tested functions. Comparison of outcomes of GMDH model.

2.2 Open Problems in Existing System

The accuracy of water quality prediction is slightly less than ANN and SVM. The evaluation of the accuracy of the applied models according to the error indexes declared that SVM was the most accurate model. Examining the results of the models showed that all of them had some over-estimation properties. By evaluating the results of the models based on the DDR index, it was found that the lowest DDR value was related to the performance of the SVM model. Trees have a profound impact on human life that makes their year-round monitoring with dielectric characteristics (DCs) of wood water as controlled parameters of great importance. This deals with studying some behavior features of DCs of different- type wood water in various phenophases under varying moisture, temperature, and concentrations of dissolved substances. DCs carry information on the phase composition of water present in plant tissue, volume fractions of bound and free water, concentrations, and types of dissolved substances. In order to describe DC, the refractive model accounting phase composition of water in wood was used. The revealed behavior features of n and k for three different categories of wood water allowed us to calculate the DC of living tree branches

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Feasibility Studies/Risk Analysis of the Project

- To collect data set for the training with respective of all parameters to included.
- To design a user friendly frame work for easy interactions among different user.
- To analyse the quality of water or to predict the portable water.
- To train the machine with huge number of dataset and collected data.

3.2 Software Requirements Specification Document

[1] Software Requirements:

- I. Operating System : Windows 10 or later
- II. Tool :
 - Anaconda with Jupyter Notebook
 - Python
 - Html
 - CSS
 - Flask Web Framework

[2] Hardware requirements:

- I. Hard disk : minimum 10 GB
- II. RAM : minimum 4 GB
- III. Preprocessor : inteli3core

3.3 System Use Case

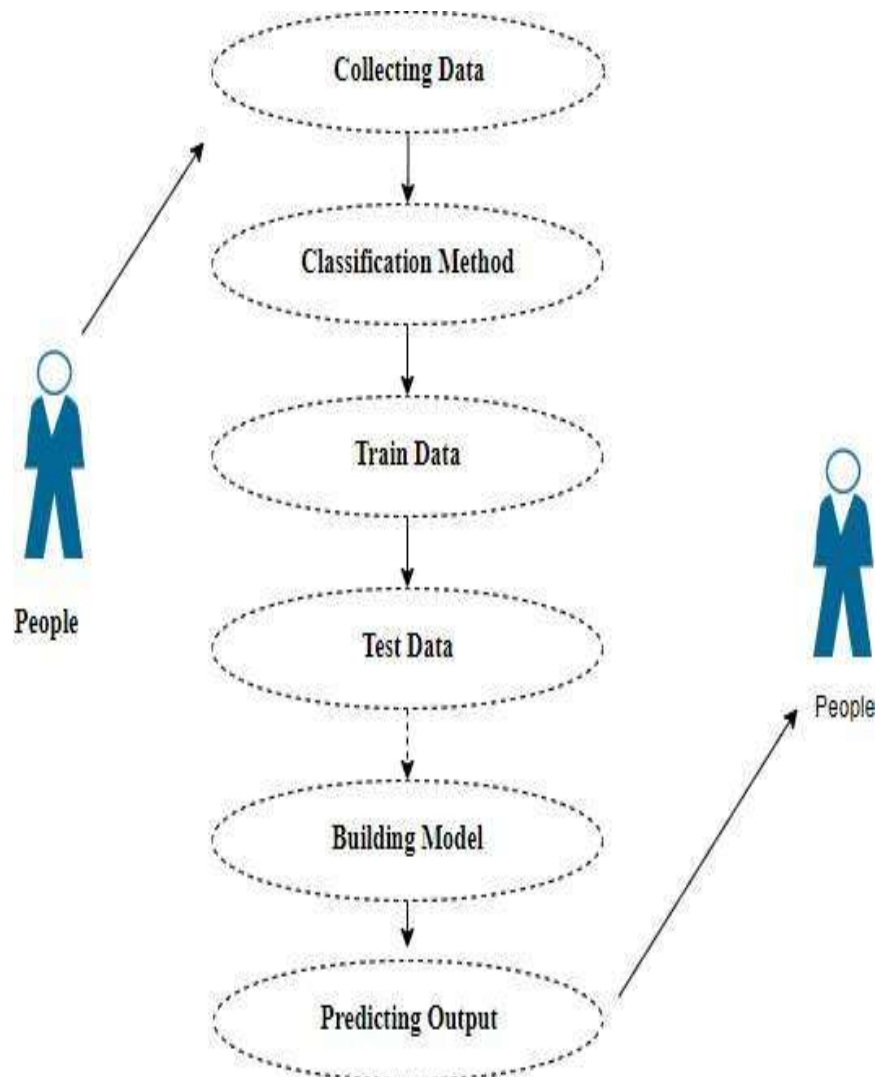


Figure 3.1 - Use Case Diagram for Water Quality Prediction

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

4.1 Selected Methodology or process model

The aim of this study is the prediction of water quality components using Machine Learning techniques. Access to safe drinking water is essential to health, a basic human right, and a component of effective policy for health protection. In this Project, we can find out the prediction of water quality. The goal is to develop a machine-learning model for Water Quality Prediction, to potentially replace the updatable supervised machine-learning classification models by predicting results in the form of best accuracy by comparing supervised algorithms. The scope of the project is that IT investigates the quality of the water with computer based prediction could reduce with errors and improve with prediction find out the quality of the Water Quality prediction.

4.2 Architecture / Overall Design of Proposed System

The proposed method is to build a machine-learning model for the classification of Water Quality. The process carries from data collection where past data related to water quality are collected. Data mining is a commonly used technique for processing enormous data. Water quality is found before, can lower problems arise in human body. Machine learning is now applied where it reduces manual effort and a better model makes error less which leads to preventing problems. The data analysis is done on the dataset, proper variable identification is done that is both the dependent variables and independent variables are found. The proper machine learning algorithms are applied to the dataset where the pattern of data is learned. After applying different algorithms a better algorithm is used for the prediction of the outcome. The steps involved in the process stated as

- Data Pre-processing
- Data Analysis of Visualization
- Implementing Cat Boost Algorithm
- Implementing Logistic Regression Algorithm
- Implementing Random Forest Algorithm
- implementing Naïve Bayer's Algorithm
- Deployment

4.2.2 Workflow diagram

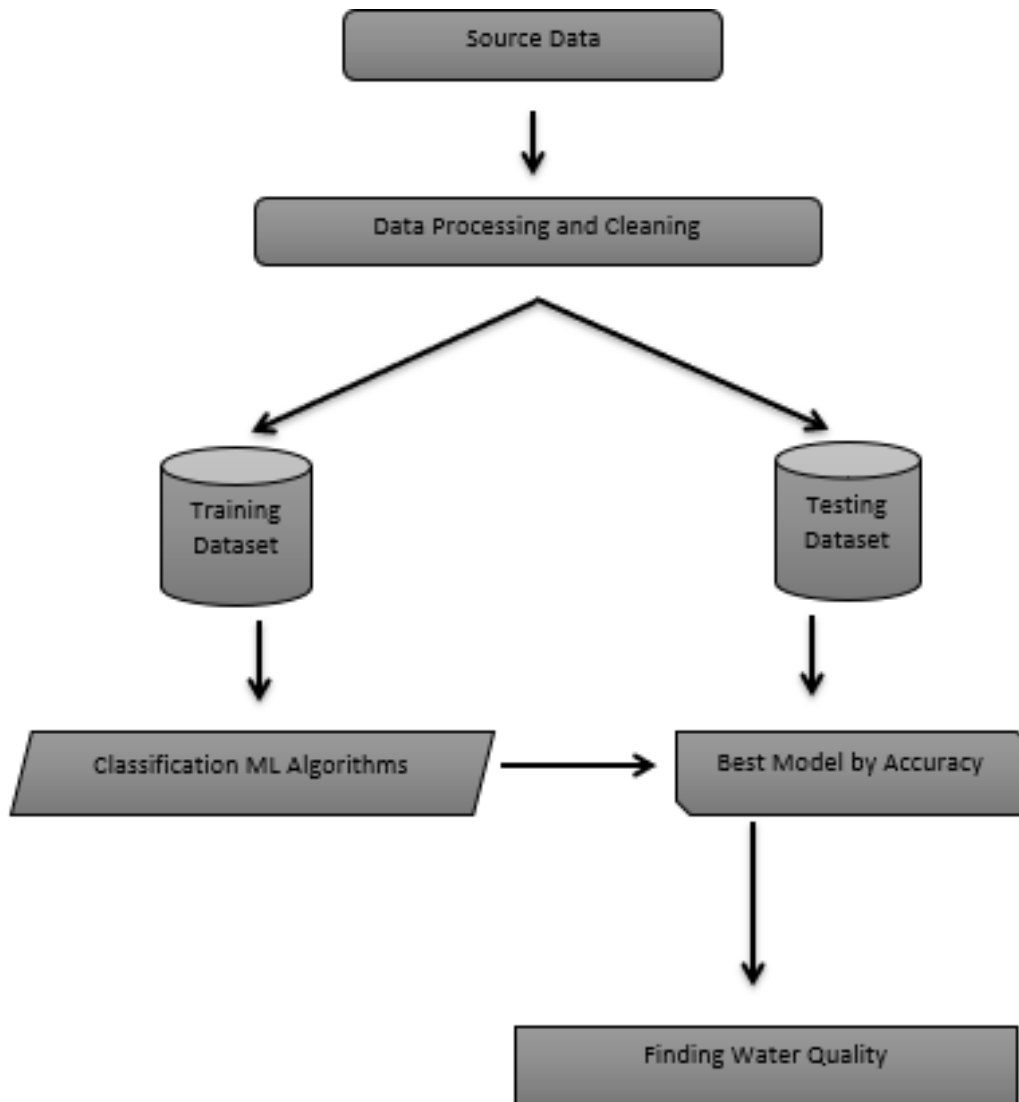


Figure 4.1 - Work Flow Diagram for Water Quality Prediction

4.2.3 Class Diagram

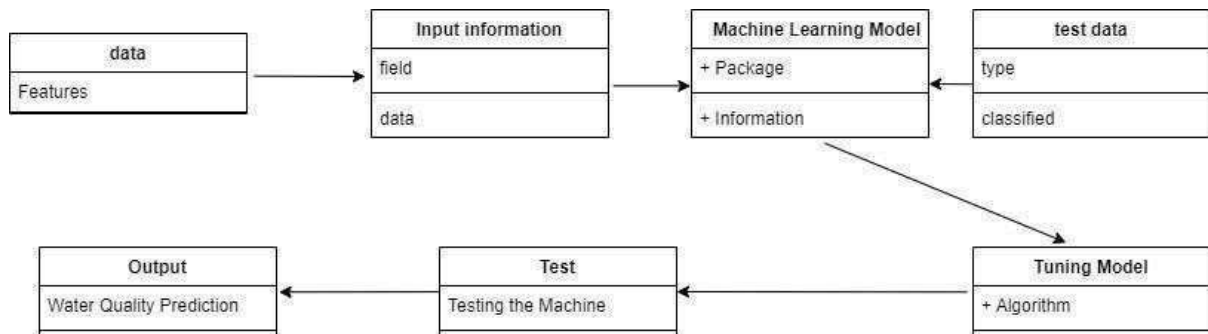


Figure 4.2 - Class Diagram for Water Quality Prediction

4.2.4 Entity Relationship Diagram (ERD)

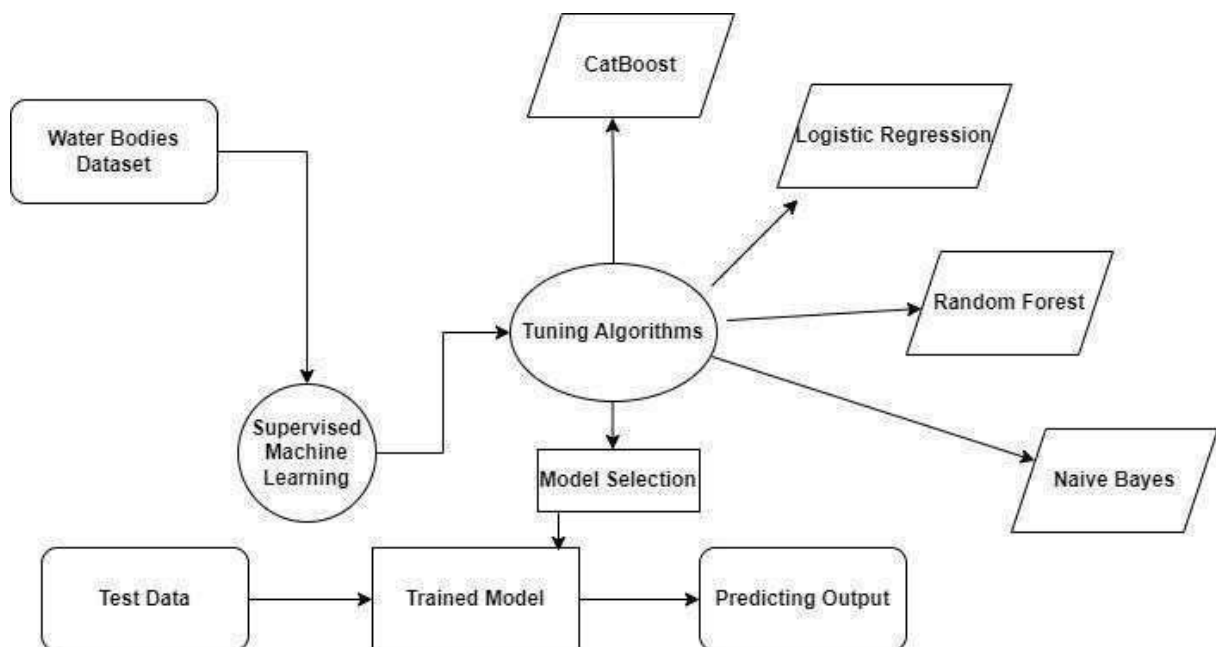


Figure 4.3 - Entity Diagram

4.2.5 System Architecture

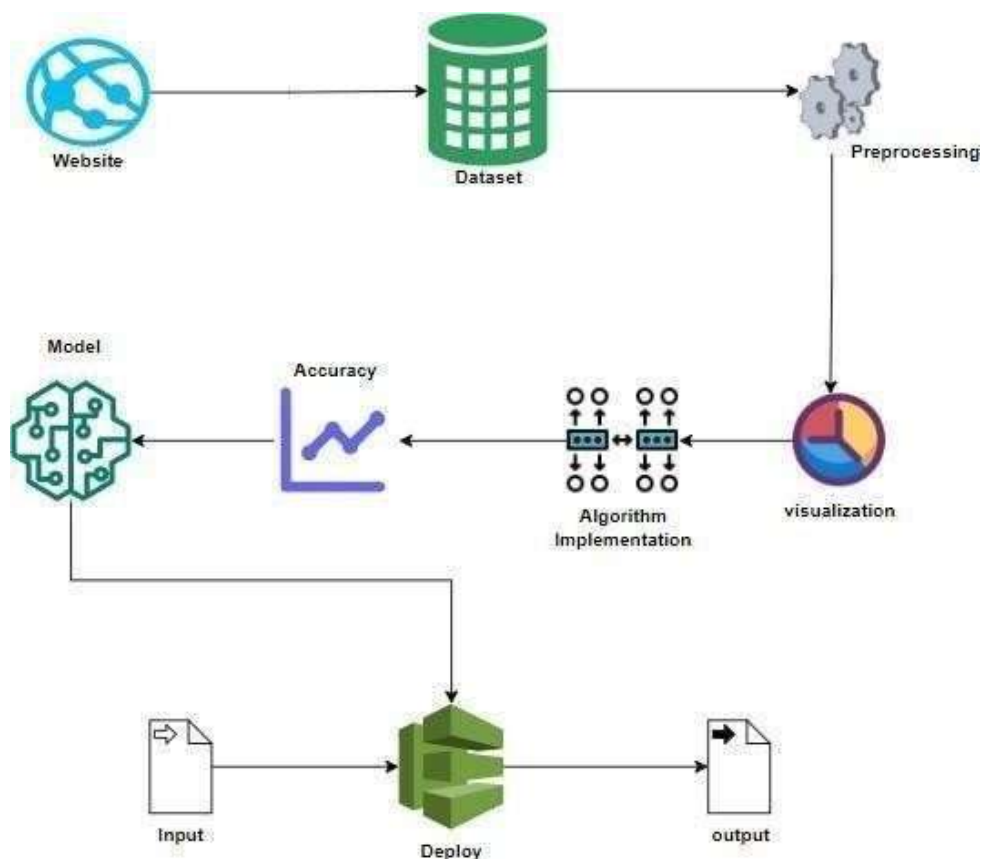


Figure 4.4 – System Architecture

4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

4.3.1 MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data

the corresponding labelling to learn data has to be labelled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

4.3.2 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. Anaconda. Now, if you are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science. Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions. The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly. Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz
- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)

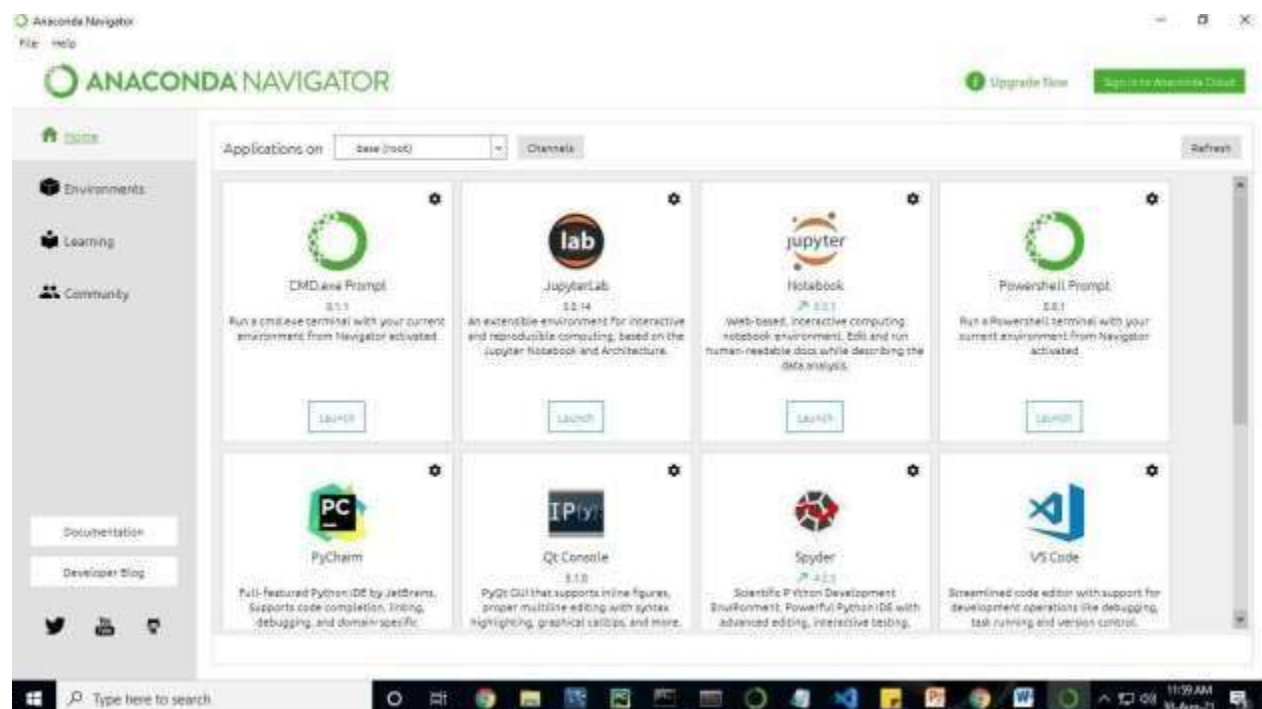
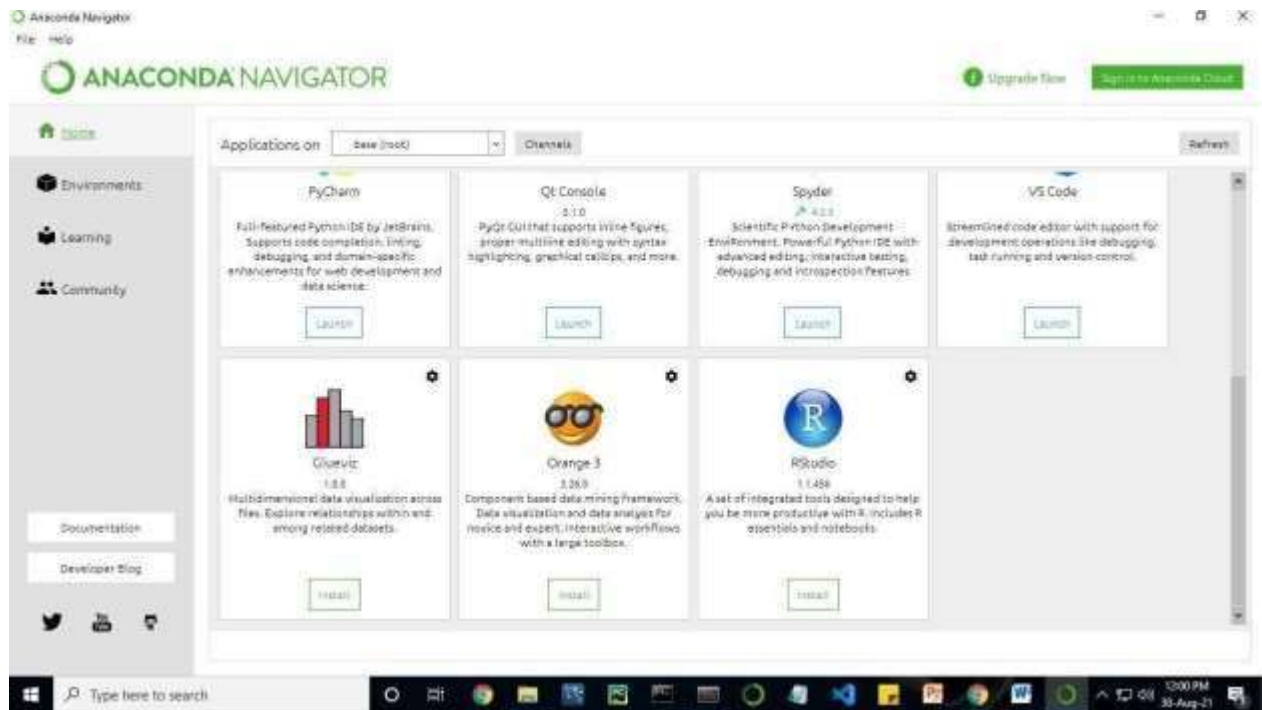


Figure 4.5 - Anaconda Navigator

4.3.3 JUPYTER NOTEBOOK

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started. Project Jupyter is a project and community whose goal is to "develop open-

source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez. Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis. Installation: The easiest way to install the *Jupyter Notebook App* is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called Anaconda

KERNEL: A notebook *kernel* is a “computational engine” that executes the code contained in a Notebook document. The *ipython kernel*, referenced in this guide, executes python code. Kernels for many other languages exist.

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming. The best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird’s eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

4.3.4 PYTHON

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than C or Pascal. Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's

"There" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy

4.3.5 FLASK WEB DEPLOYMENT FRAMEWORK

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework-related tools.

Advantages of Flask:

- Higher compatibility with latest technologies.
- Technical experimentation.
- Easier to use for simple cases.
- Codebase size is relatively smaller.
- High scalability for simple applications.
- Easy to build a quick prototype.
- Routing URL is easy.
- Easy to develop and maintain applications.

4.3.6 HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

4.3.7 CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users. CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Development and Deployment Setup

5.1.1 DATA PREPROCESSING

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the givendataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data

- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

The data set contains water quality metrics for 3276 different water bodies with the values of minerals and others. The quality of water is predicted by the given data values which includes and should be in the range given below

COMPONENT	RANGE
Ph	6.5 to 8.5
Solids	Btw- 50 to 150
Sulfate	500 Mg/l
Organic carbon	Below 0.500 Mg/l TOC
Turbidity	Below 1 TOC
Hardness	120 to 170 Mg
Conductivity	200 to 1000 mu/cm
Potability	300 pp m to 500 ppm
Trihalomethanes	100 ppb
Chloramines	4 Mg/l or 4ppm

Table – 5.1 - Data Analysis

In [4]: 1 data

Out[4]:

	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	10630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0
...	...	...	...	...	...	...	...	...	...	...
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	13.894419	66.687695	4.436821	1
3272	7.808856	193.553212	17329.802160	8.061362	NaN	392.449580	19.903225	NaN	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	NaN	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	NaN	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	NaN	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

Figure – 5.2 - Data Classification

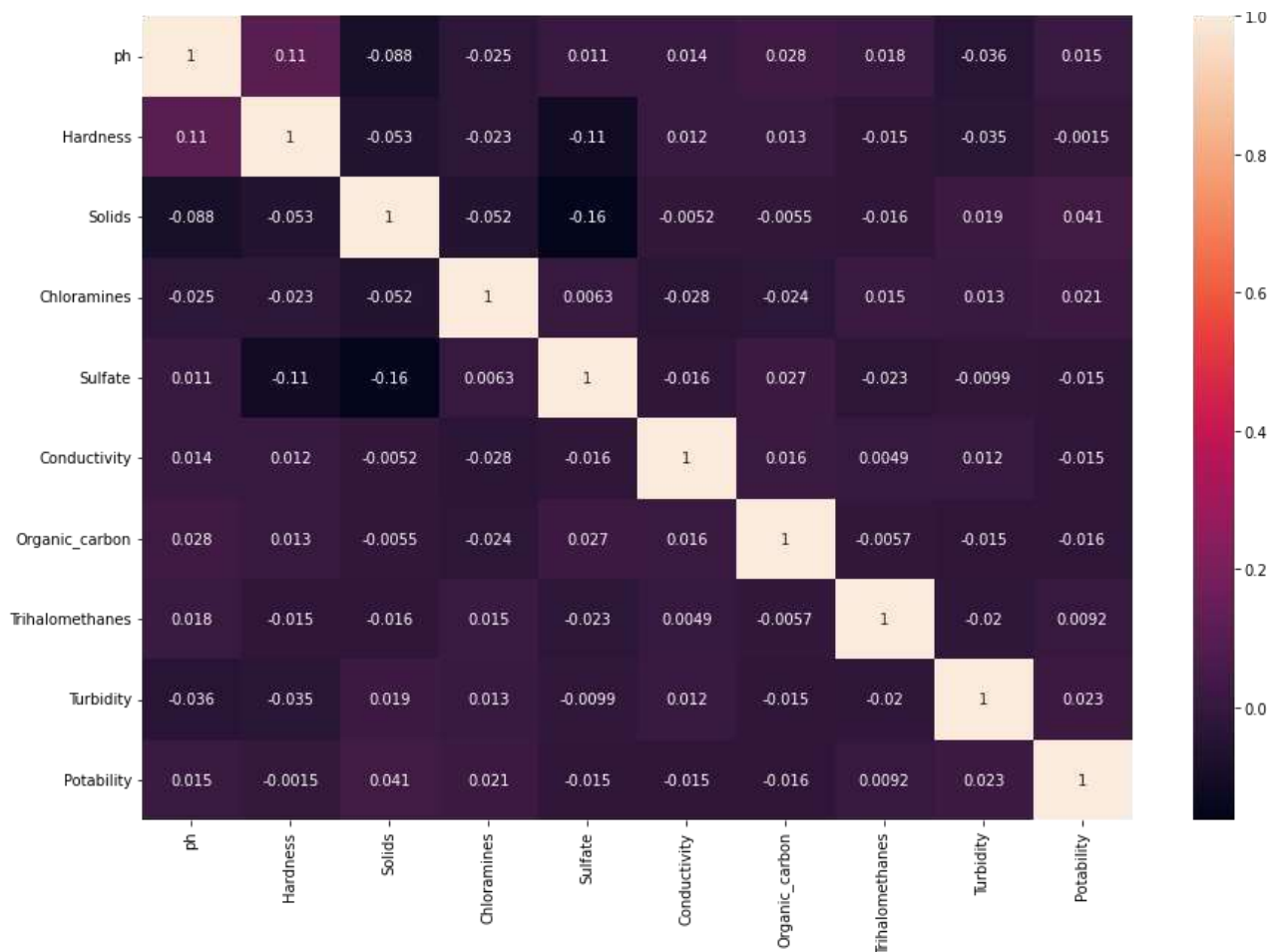


Figure – 5.3 - Data configuration

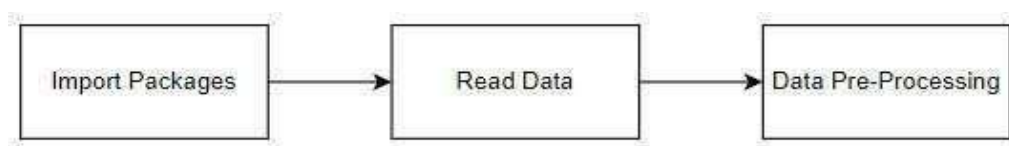


Figure – 5.4 - Data Preprocessing Modular Diagram

5.1.2 DATA VISUALIZATION

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding.

This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end. Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with barcharts.
- How to summarize data distributions with histograms and box plots.

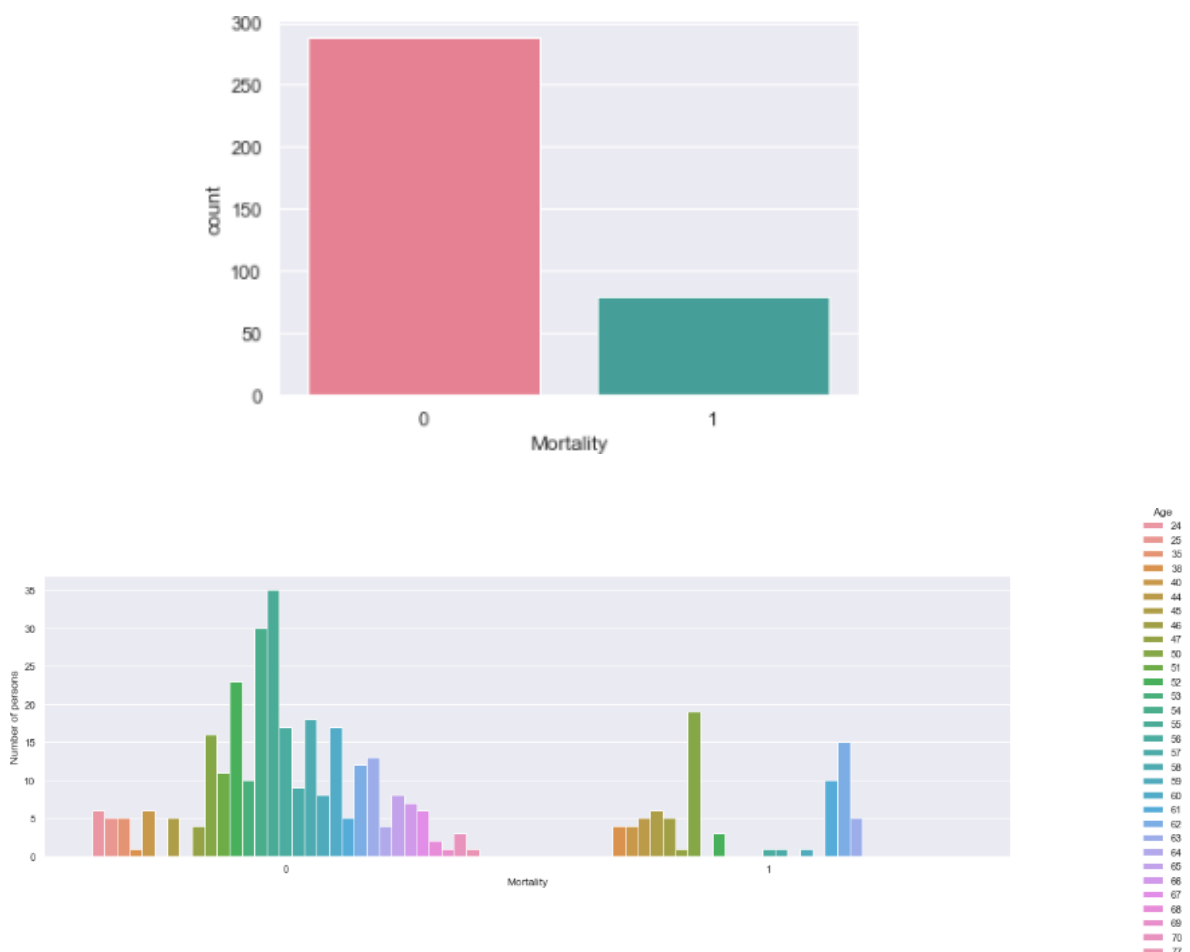


Figure 5.5 - Portable Drinking water Analysis from Data optimized

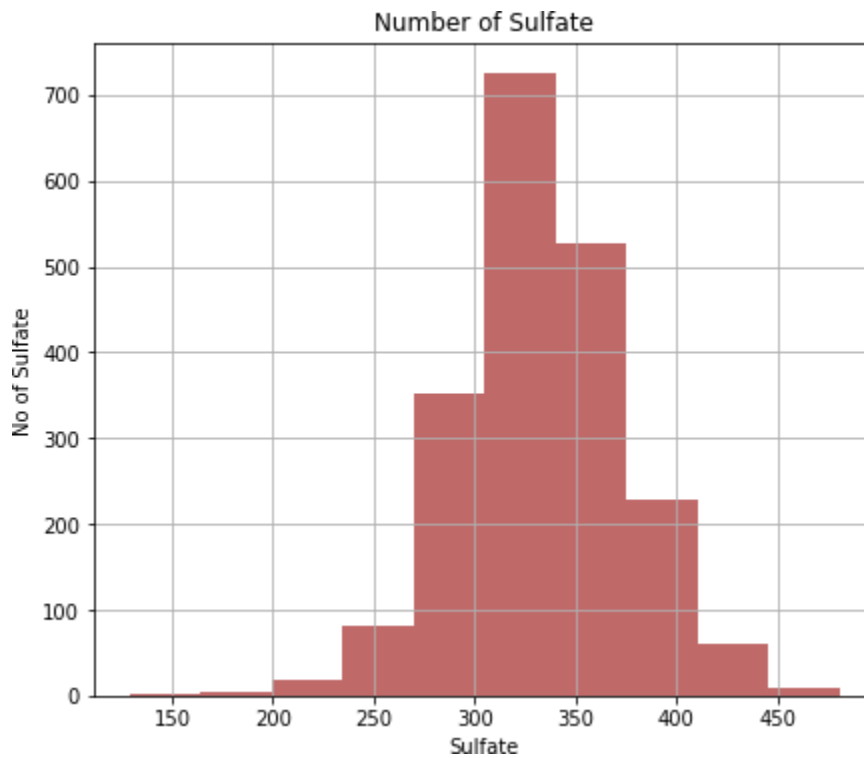


Figure 5.6 – Data visualization for Sulfate

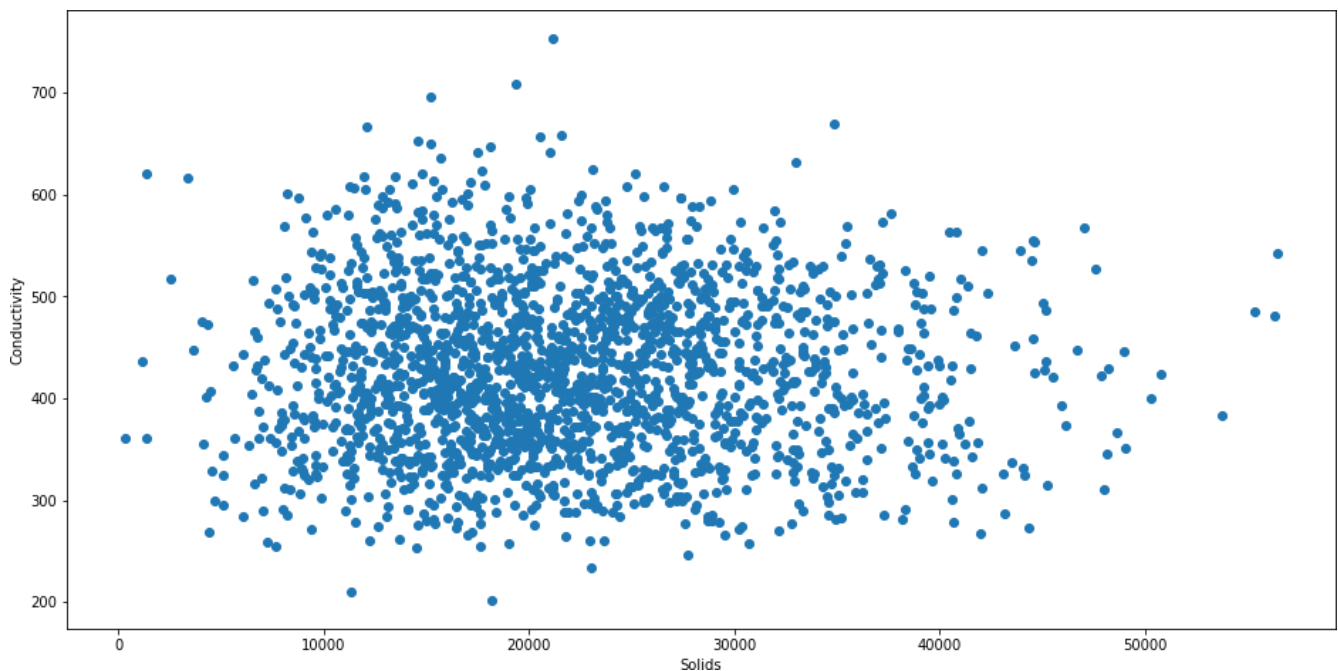


Figure 5.7 – Plotted chart analysis for Conductivity

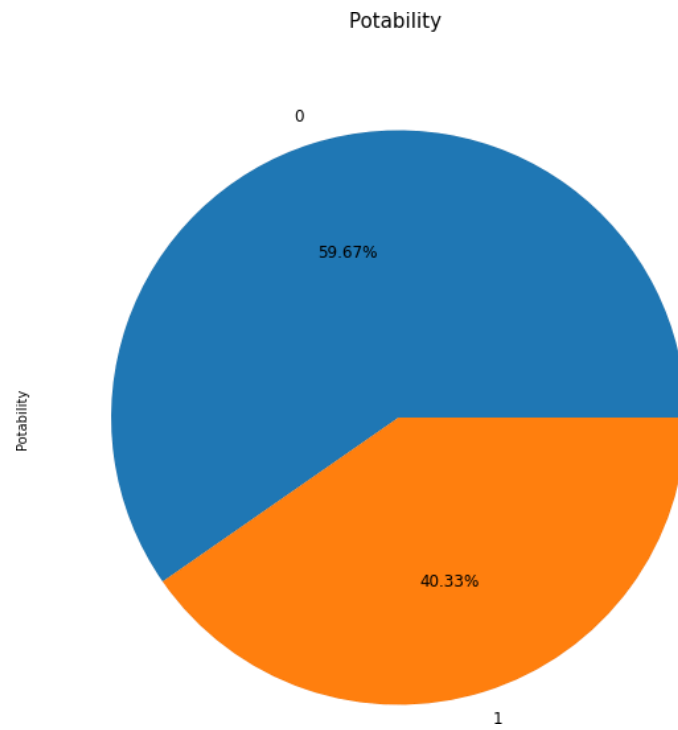


Figure 5.8 – Portable water range

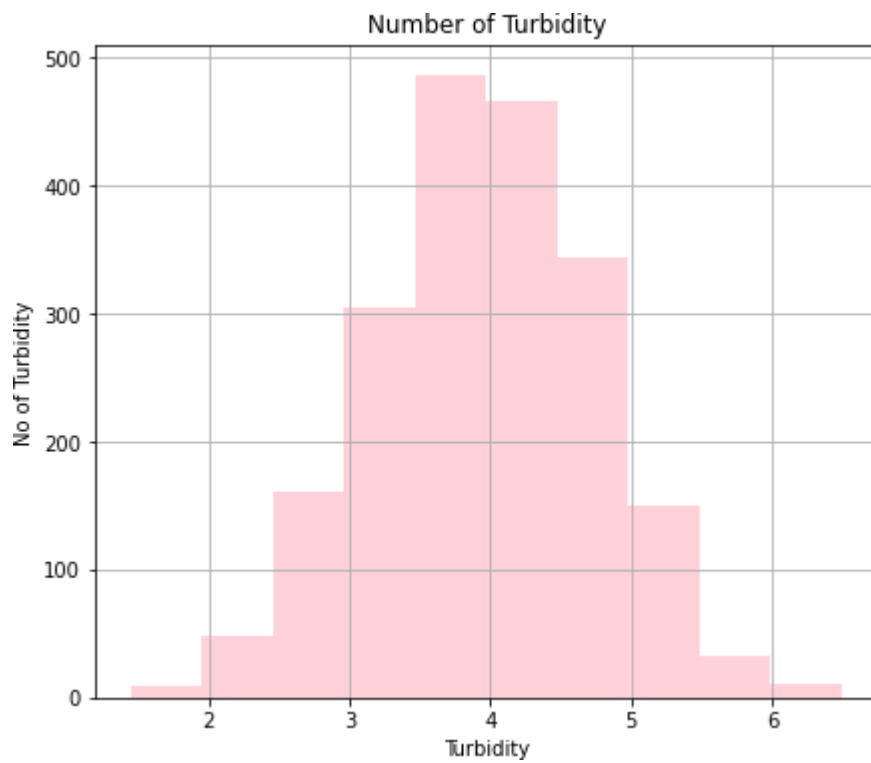


Figure 5.9 – Analysis of Turbidity in water bodies

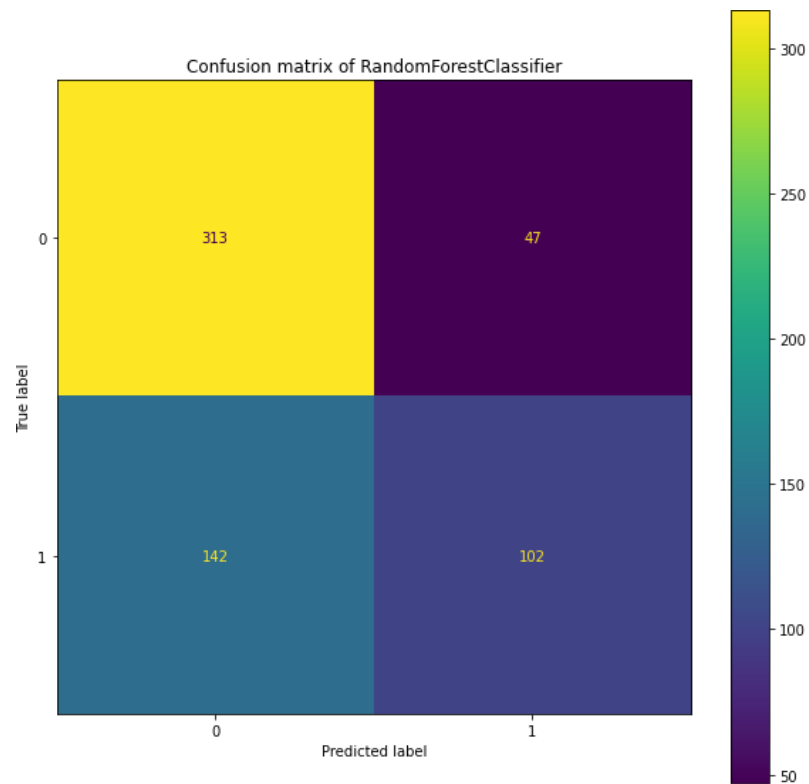


Figure 5.10 – Confusion Matrix for the dataprovided

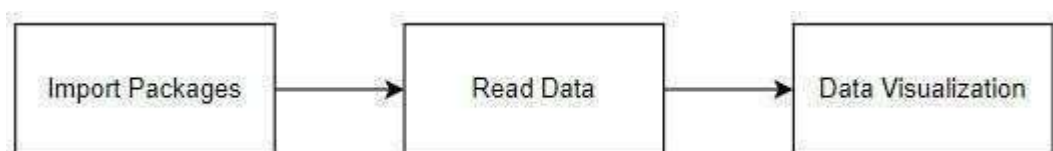


Figure 5.11 - Module Diagram for Data Visualization

5.1.3 Performance Metrics to calculate

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is

yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of the actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

True Positive Rate(TPR) = $TP / (TP + FN)$ False Positive rate(FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. Precision = $TP / (TP + FP)$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labelled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = $TP / (TP + FN)$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

5.2 ALGORITHMS

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. The 4 different algorithms are compared:

- **CatBoost**
- Logistic Regression
- Random Forest Classifier
- Naïve Bayes

MODULE DIAGRAM

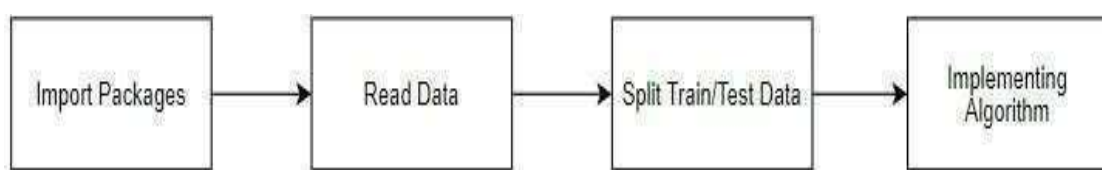


Figure 5.12 – Module Diagram for Algorithm Implementation

GIVEN INPUT EXPECTED OUTPUT

input : data

output: getting accuracy

5.2.1 CATBOOST

CatBoost is an algorithm for gradient boosting on decision trees. It is developed by Yandex researchers and engineers, and is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks at Yandex and in other companies, including CERN, Cloudflare, Careem taxi. The CatBoost algorithm is based on Gradient Descent and is a powerful technique for supervised machine learning tasks. It will be well suited to problems that involve categorical data. It is widely used for regression and classification tasks and it is also one of the most used algorithms in Kaggle.

Finding Accuracy

```
In [11]: 1 accuracy = accuracy_score(y_test,predicted)
        2 print('Accuracy of CatBoostClassifier ',accuracy*100)
```

Accuracy of CatBoostClassifier 69.37086092715232

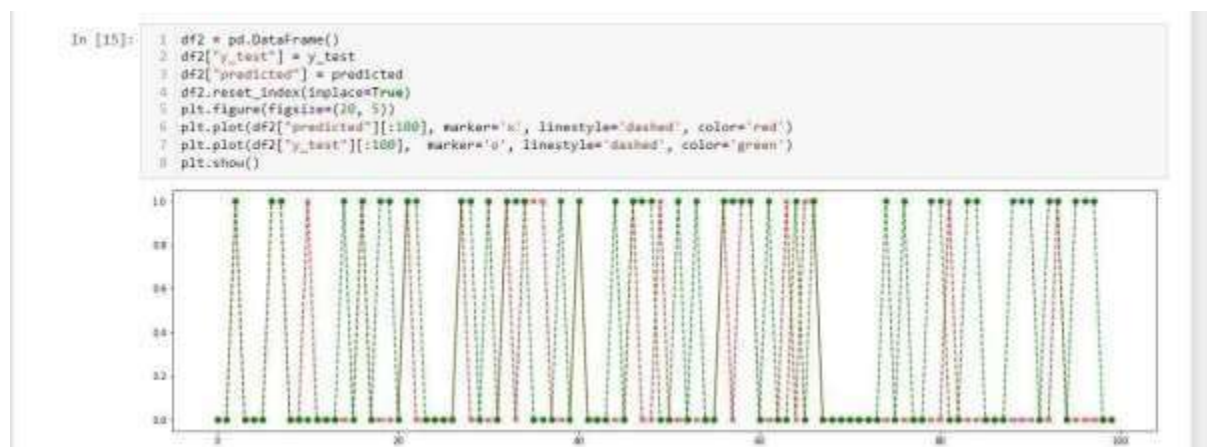


Figure 5.13 – Implementation of Catboost Classifier Algorithm

5.2.2 LOGISTIC REGRESSION

Logistic regression is commonly used for prediction and classification problems. Some of these use cases include: Fraud detection: Logistic regression models can help teams identify data anomalies, which are predictive of fraud. Logistic Regression is one of the basic and

popular algorithms to solve a classification problem. It is named 'Logistic Regression' because its underlying technique is quite the same as Linear Regression. The term “Logistic” is taken from the Logit function that is used in this method of classification. Logistic regression is easier to implement, interpret, and very efficient to train. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting. It makes no assumptions about distributions of classes in feature space.

```
] 1 lr = LogisticRegression()  
2 lr.fit(X_train,y_train)  
3 predicted = lr.predict(X_test)
```

Finding Accuracy

```
] 1 accuracy = accuracy_score(y_test,predicted)  
2 print('Accuracy of LogisticRegression',accuracy*100)
```

Accuracy of LogisticRegression 59.437086092715234

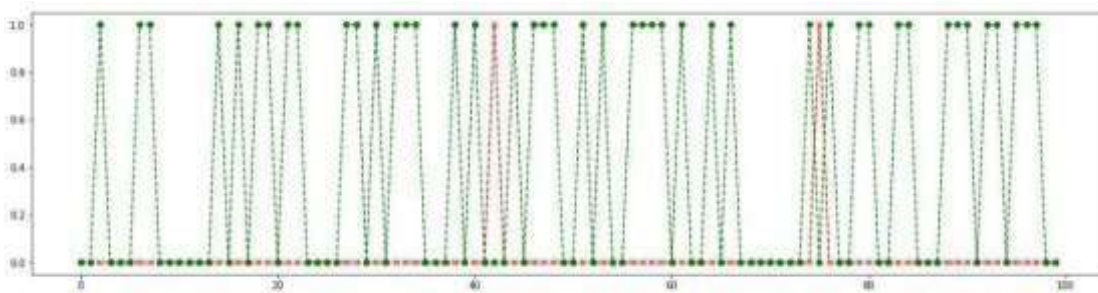


Figure 5.14 – Implementation of Logistic Regression

5.2.3 RANDOM FOREST CLASSIFIER

From there, the random forest classifier can be used to solve for regression or classification problems. The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Random Forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. For multiclass problem you will need to reduce

it into multiple binary classification problems. Random Forest works well with a mixture of numerical and categorical features.



Figure 5.15 – Implementation of Random Forest Algorithm

5.2.4 Naive Bayes

The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality. Therefore they are considered as naive. The Naive Bayes classifier works on the principle of conditional probability, as given by the Bayes theorem. While calculating the math on probability, we usually denote probability as P. Some of the probabilities in this event would be as follows: The probability of getting two heads = $\frac{1}{4}$. Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent.

```
In [10]: 1 nb = GaussianNB()
         2 nb.fit(X_train,y_train)
         3 predicted = nb.predict(X_test)
```

Finding Accuracy

```
In [11]: 1 accuracy = accuracy_score(y_test,predicted)
         2 print('Accuracy of GaussianNB',accuracy*100)
```

Accuracy of GaussianNB 60.43046357615894

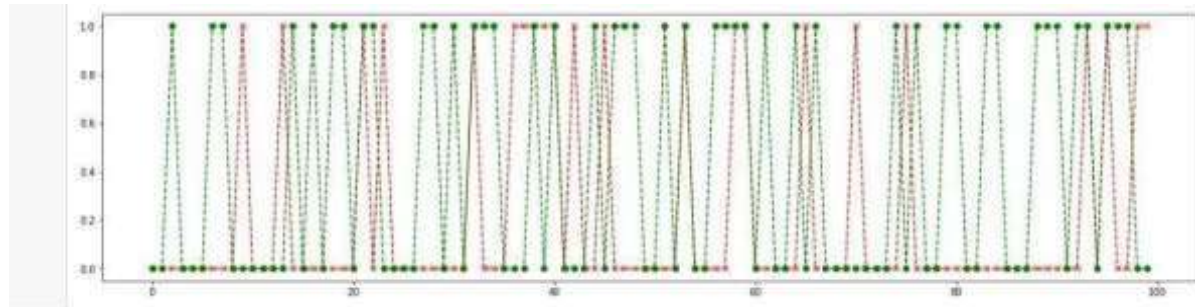
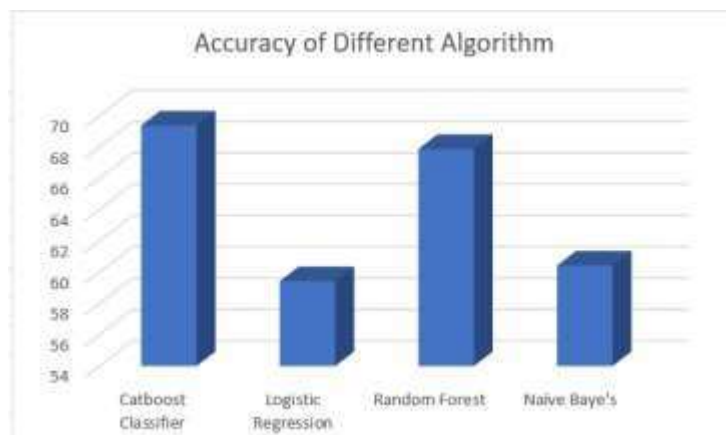


Figure 5.16 – Accuracy of Gaussian's Algorithm

CHAPTER 6

RESULTS AND DISCUSSION

This study concentrated on two primary concepts. The first was the development of a real-time water monitoring network for collecting data on water parameters from bodies of water. Second, the use of machine learning (ML) algorithms to analyse water quality. Data from this monitoring network should ideally be collected so that ML models may be used to assess the water's suitability for drinking or irrigation. Due to a lack of relevant datasets, two acceptable datasets were created in this study and utilised to train and evaluate four ML models investigated, which are Catboost classifier, Random Forest (RF), Logistic Regression (LR), and Naive Bayes. The test results revealed that catboost worked best for drinking water, with the highest classification accuracy and the lowest false positive and negative readings. Finally, a model for determining the most significant water characteristics in terms of ML model classification accuracy. Furthermore, water quality indices were created manually and used to assess water's "fitness for use." Future research should investigate the usage of unsupervised ML models as alternatives to manually derived water quality indices. The analytical procedure begins with data cleaning and processing, followed by missing value analysis, exploratory analysis, and lastly model creation and assessment. The algorithm with the highest accuracy score on the wider test set will be determined. The discovered one is used in an application that can assist in determining the Water Quality Prediction. The task that will be implemented in the Machine Learning environment and to automate this procedure by displaying the prediction result in a web or desktop application.



Comparison of different algorithms used for data training set

CHAPTER 7

CONCLUSION & FUTURE WORKS

CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set of higher accuracy score algorithm will be find out. The founded one is used in the application which can help to find the Water Quality Prediction.

7.1 FUTURE WORKS

- To automate this process by show the prediction result in web application or desktop application.
- To optimize the work to implement in Machine Learning environment.

A. REFERENCES

- [1] Ajayi 1 , Antoine B. Bagula 1 , Hlonipani C. Maluleke 1 , Zaheed Gaffoor 2
Nebo Jovanovic 2 , And Kevin Bellville, Cape Town 7535, South Africa
2Department of Earth Science, South Africa- WaterNet: A Network for
Monitoring and Assessing Water Quality for Drinking and Irrigation
- [2] Amir Hamzeh Haghiabi, Ali Heidar Nasrolahi and Abbas Parsaie .. -Water
Quality Prediction Using Machine Learning Methodss investigates the
performance of artificial intelligence techniques including artificial neural network
(ANN)- 2020
- [3] Dao Nguyen Khoi, Nguyen Trong Quan , Do Quang Linh , Pham Thi Thao Nhi
and Nguyen Thi Diem Thuy in the year 2022 - study using Machine Learning
Models for Predicting the Water Quality Index in the Ia Buong River,Vietnam
- [4] Di Wu;Hao Wang;Hadi Mohammed;Razak Seidu -Quality Risk Analysis for
Sustainable Smart Water Supply Using Data Perception IEEE Transactions on
Sustainable Computing Year: 2020 | Publisher: IEEE
- [5] Di Wu;Hao Wang;Razak Seidu 2019 IEEE Intl Conf on Dependable,
Autonomic and Secure Computing, - Tensor Model for Quality Analysis in
Industrial Drinking Water Supply SystemIntl Conf on Pervasive Intelligence and
Computing, Intl Conf on Cloud and Big Data Computing,
(DASC/PiCom/CBDDCom/CyberSciTech) Year: 2019 | Publisher: IEEE
- [6] Gaganjot Kang;Jerry Zeyu Gao;Gang Xie 2017- Data Driven Water Quality
Analysis and Prediction IEEE Third International Conference on Big Data
Computing Service and Applications (BigDataService)Data- Driven Water
Quality Analysis and Prediction: Year: 2017 | Publisher: IEEE

- [7] Hao Wang;Hadi Mohammed;Razak Seidu- Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception Di Wu EEE Transactions on Sustainable Computing Year: 2020 Publisher: IEEE
- [8] Huimin Jia;Xiaofeng Zhou – Water Quality Prediction Method Based on LSTM- - 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) Year: 2020 | Publisher: IEEE
- [9] Jian Zhou , Yuanyuan Wang , Fu Xiao, Yunyun Wang and Lijuan Sun - Water Quality Prediction Method Based on IGRA and LSTM which is proposed in the year 2018.Water quality prediction has great significance for water environment protection.
- [10] Jitha P Nair;M S Vijaya 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)Year: 2021 - Predictive Models for River Water Quality using Machine Learning and Big Data Techniques - A Survey Publisher: IEEE+
- [11] K. P. Rasheed Abdul Haq;V. P. Harigovindan Year: 2022 - Water Quality Preditcion for Smart Aquaculture Using Hybrid Deep Learning Models
- [12] Md. Saikat Islam Khan, Nazrul Islam, Jia Uddin , Sifatul Islam, Mostofa Kamal Nasir in the year 2021 - A study on Water Quality Predction and Classification based on Principal Component Regression and Gradient boosting Classifier approach
- [13] P.Varalakshmi - Prediction of water quality using Naive Bayesian algorithm Year: 2017 | Publisher: IEEE
- [14] Priskilla Angel Rani;R. Sujeethra;C. Yesubai Rubavathi - Prediction of Water Quality using Artificial Neural Network
- [15] QuanXi Dong;YongZhe Lin;Jing Bi;Haitao Yuan International Conference on Systems, Man and Cybernetics (SMC)Year: 2019 -An Integrated Deep Neural Network Approach for Large-Scale Water Quality Time Series PredictionPublisher: IEEE
- [16] Rongali Yang- Summary of Water Quality Prediction Models Based on Machine Learning - Year: 2021 |Publisher: IEEE

- [17] Sanaz Imen, Ni-Bin Chang, Senior Member, IEEE, Y. Jeffrey Yang, and Arash Golchubian Developing a Model-Based Drinking Water Decision Support System Featuring Remote Sensing and Fast Learning Techniques
- [18] Swati Chopade, Hari Prabhat Gupta , Rahul Mishra , Graduate Student Member, IEEE, Aman Oswal, Preti Kumari , Graduate Student Member, IEEE ,A Sensor Based River Water Quality Assessment System Using Deep Learning Neural Network
- [19] Theyazn H. H Aldhyani , Mohammed Al-Yaari , Hasan Alkahtani, and Mashael Maashi proposed - the Water Qaulity Prediction using Aritificial Intelligence
- [20] Xiao-yi Wang;Ji-ping Xu;Zai-wen Liu;Jun Dai;Shi-ping Zhu- Intelligent Water Quality Remote Monitor and Water Bloom Prediction
- [21] Yanfei Zhao;Zhihong Zou;Shenglong Wang -A Back Propagation Neural Network Model based on kalman filter for water quality prediction Publisher: IEEE Year: 2022 | Publisher: IEEE
- [22] Yinshan Yu;Yan Qu;Hongyun Zhang;Lingjie Jiang;Mingzhen Shao;Dandan Wei;Dongyang Zhang- Indicator analysis of lake water quality based on fuzzy neural network and spectrophotometry 2022 IEEE 10th Joint International Information Technology and ArtificialIntelligence Conference (ITAIC) Year: 2022 | Publisher: IEEE

B. APPENDIX

SOURCE CODE

Data Preprocessing:

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd

data = pd.read_csv('water_potability.csv')

data
Before removing the null data

data.shape

data.isnull().sum()
After removing the null data

df = data.dropna()

df.shape

df.isnull().sum()

df.info()

df.columns

df.duplicated()

df.duplicated().sum()

df.Conductivity.unique()

df.ph.unique()

df.Turbidity.unique()

df.Potability.unique()

pd.crosstab(df.Potability,df.Solids)

pd.crosstab(df.Potability,df.Turbidity)

df.Potability.value_counts()
```

```
df.columns  
Before LabelEncoder
```

```
df.head()
```

```
df.corr()
```

Module 2: Exploration data analysis of visualization and training a model by given attributes

```
import warnings  
warnings.filterwarnings('ignore')
```

```
#import library packages  
import pandas as p  
import matplotlib.pyplot as plt  
import seaborn as s  
import numpy as n
```

```
data = p.read_csv("water_potability.csv")
```

```
df=data.dropna()
```

```
df.columns
```

```
df.columns
```

```
#Histogram Plot  
df['Sulfate'].hist(figsize=(7,6), color='brown', alpha=0.7)  
plt.xlabel('Sulfate')  
plt.ylabel('No of Sulfate')  
plt.title('Number of Sulfate')
```

```
#Histogram Plot  
df['Turbidity'].hist(figsize=(7,6), color='pink', alpha=0.7)  
plt.xlabel('Turbidity')  
plt.ylabel('No of Turbidity')  
plt.title('Number of Turbidity')
```

```
#Propagation by variable  
def PropByVar(df, variable):  
    dataframe_pie = df[variable].value_counts()  
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)  
    ax.set_title(variable + ' \n', fontsize = 15)  
    return n.round(dataframe_pie/df.shape[0]*100,2)  
PropByVar(df, 'Potability')  
  
plt.boxplot(df['Trihalomethanes'])
```

```

plt.show()

plt.boxplot(df['ph'])
plt.show()

# Heatmap plot diagram
fig, ax = plt.subplots(figsize=(15,10))
s.heatmap(df.corr(), annot=True)

df.columns

fig, ax = plt.subplots(figsize=(16,8))
ax.scatter(df['Solids'],df['Conductivity'])
ax.set_xlabel('Solids')
ax.set_ylabel('Conductivity')
plt.show()

df.columns
Splitting Train/Test:

#preprocessing, split test and dataset, split response variable
X = df.drop(labels='Potability', axis=1)
#Response variable
y = df.loc[:, 'Potability']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))

```

IMPLEMENTING CATBOOST CLASSIFIER ALGORITHM

```

#import library packages
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

# Load given dataset
data = pd.read_csv("water_potability.csv")

df = data.dropna()

df.columns

```

```
df.head()
```

```
#preprocessing, split test and dataset, split response variable
```

```
X = df.drop(labels='Potability', axis=1)
```

```
#Response variable
```

```
y = df.loc[:, 'Potability']
```

```
#Splitting for train and test
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42, stratify=y)
```

```
print("Number of training dataset: ", len(X_train))
```

```
print("Number of test dataset: ", len(X_test))
```

```
print("Total number of dataset: ", len(X_train)+len(X_test))
```

```
Implementing CatBoostClassifier Algo
```

```
from catboost import CatBoostClassifier
```

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, plot_confusion_matrix
```

```
Training
```

```
cat = CatBoostClassifier()
```

```
cat.fit(X_train, y_train)
```

```
predicted = cat.predict(X_test)
```

```
Finding Accuracy
```

```
accuracy = accuracy_score(y_test, predicted)
```

```
print('Accuracy of CatBoostClassifier ', accuracy*100)
```

```
Finding Classification Report
```

```
cr = classification_report(y_test, predicted)
```

```
print('Classification report\n\n', cr)
```

```
Finding Confusion matrix
```

```
cm = confusion_matrix(y_test, predicted)
```

```
print('Confusion matrix\n\n', cm)
```

```
import matplotlib.pyplot as plt
```

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

```
plot_confusion_matrix(cat, X_test, y_test, ax=ax)
```

```
plt.title('Confusion matrix of CatBoostClassifier')
```

```
plt.show()
```

```
df2 = pd.DataFrame()
```

```
df2["y_test"] = y_test
```

```
df2["predicted"] = predicted
```

```
df2.reset_index(inplace=True)
```

```
plt.figure(figsize=(20, 5))
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

IMPLEMENTING RANDOM FOREST ALGORITHM

#import library packages

import pandas **as** pd

import warnings

warnings.filterwarnings('ignore')

Load given dataset

data = pd.read_csv("water_potability.csv")

df = data.dropna()

df.columns

df.head()

#preprocessing, split test and dataset, split response variable

X = df.drop(labels='Potability', axis=1)

#Response variable

y = df.loc[:, 'Potability']

#Splitting for train and test

from sklearn.model_selection **import** train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42, stratify=y)

print("Number of training dataset: ", len(X_train))

print("Number of test dataset: ", len(X_test))

print("Total number of dataset: ", len(X_train)+len(X_test))

Implementing RF Algo

from sklearn.ensemble **import** RandomForestClassifier

from sklearn.metrics **import** confusion_matrix, classification_report, accuracy_score, plot_confusion_matrix

Training

rf = RandomForestClassifier()

rf.fit(X_train, y_train)

predicted = rf.predict(X_test)

Finding Accuracy

accuracy = accuracy_score(y_test, predicted) rint('Accuracy

Finding Classification Report

```
cr = classification_report(y_test,predicted)
print('Classification report\n\n',cr)
Finding Confusion matrix
```

```
cm = confusion_matrix(y_test,predicted)
print('Confusion matrix\n\n',cm)
```

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,10))
plot_confusion_matrix(rf, X_test, y_test, ax=ax)
plt.title('Confusion matrix of Random Forest')
plt.show()
```

```
df2 = pd.DataFrame()
df2["y_test"] = y_test
df2["predicted"] = predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

IMPLEMENTING ADABOOST ALGORITHM

```
#import library packages
```

```
import pandas as pd
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
# Load given dataset
```

```
data = pd.read_csv("water_potability.csv")
```

```
df = data.dropna()
```

```
df.columns
```

```
df.head()
```

```
#preprocessing, split test and dataset, split response variable
```

```
X = df.drop(labels='Potability', axis=1)
```

```
#Response variable
```

```
y = df.loc[:, 'Potability']
```

```
#Splitting for train and test
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42,
stratify=y)
```

```
print("Number of training dataset: ", len(X_train))
```

```
print("Number of test dataset: ", len(X_test))
```

```
print("Total number of dataset: ", len(X_train)+len(X_test))
```

Implementing AdaBoost Algo

```
from sklearn.ensemble import AdaBoostClassifier
```

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
```

```
plot_confusion_matrix
```

Training

```
ab = AdaBoostClassifier()
```

```
ab.fit(X_train,y_train)
```

```
predicted = ab.predict(X_test)
```

Finding Accuracy

```
accuracy = accuracy_score(y_test,predicted)
```

```
print('Accuracy of AdaBoost Classifier',accuracy*100)
```

Finding Classification Report

```
cr = classification_report(y_test,predicted)
```

```
print('Classification report\n\n',cr)
```

Finding Confusion matrix

```
cm = confusion_matrix(y_test,predicted)
```

```
print('Confusion matrix\n\n',cm)
```

```
import matplotlib.pyplot as plt
```

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

```
plot_confusion_matrix(ab, X_test, y_test, ax=ax)
```

```
plt.title('Confusion matrix of AdaBoost')
```

```
plt.show()
```

```
import matplotlib.pyplot as plt
```

```
df2 = pd.DataFrame()
```

```
df2["y_test"] = y_test
```

```
df2["predicted"] = predicted
```

```
df2.reset_index(inplace=True)
```

```
plt.figure(figsize=(20, 5))
```

```
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
```

```
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
```

```
plt.show()
```

```
.boxplot(df['ph'])
```

```
plt.show()
```

```
# Heatmap plot diagram
```

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

```
s.heatmap(df.corr(), annot=True)
```

```
fig, ax = plt.subplots(figsize=(16,8))
```

```
ax.scatter(df['Solids'],df['Conductivity'])
```

```
ax.set_xlabel('Solids')
```

```
ax.set_ylabel('Conductivity')
```

```
plt.show()
```

```

preprocessing, split test and dataset, split response variable
X = df.drop(labels='Potability', axis=1)
#Response variable
y = df.loc[:, 'Potability']
#We'll use a test size of 30%. We also stratify the split on the response variable,
which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, plot_confusion_matrix
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
predicted = rf.predict(X_test)
accuracy = accuracy_score(y_test, predicted)
print('Accuracy of RandomForestClassifier', accuracy*100)
cr = classification_report(y_test, predicted)
print('Classification report\n\n', cr)
cm = confusion_matrix(y_test, predicted)
print('Confusion matrix\n\n', cm)
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,10))
plot_confusion_matrix(rf, X_test, y_test, ax=ax)
plt.title('Confusion matrix of RandomForestClassifier')
plt.show()
df2 = pd.DataFrame() df2["y_test"] = y_test df2["predicted"] = predicted
df2.reset_index(inplace)

#import library packages
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

# Load given data

df2 = pd.DataFrame() df2["y_test"] = y_test df2["predicted"] = predicted
df2.reset_index(inplace)

#import library packages
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

# Load given data

```

```

data = pd.read_csv("water_potability.csv")

df = data.dropna()
df.columns

df.head()
#preprocessing, split test and dataset, split response variable
X = df.drop(labels='Potability', axis=1)
#Response variable
y = df.loc[:, 'Potability']

#Splitting for train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42,
stratify=y)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))
Implementing LogisticRegression Algo

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
plot_confusion_matrix
Training

lr = LogisticRegression()
lr.fit(X_train,y_train)
predicted = lr.predict(X_test)
Finding Accuracy

accuracy = accuracy_score(y_test,predicted)
print('Accuracy of LogisticRegression',accuracy*100)
Finding Classification Report

cr = classification_report(y_test,predicted)
print('Classification report\n\n',cr)
Finding Confusion matrix

cm = confusion_matrix(y_test,predicted)
print('Confusion matrix\n\n',cm)

import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,10))
plot_confusion_matrix(lr, X_test, y_test, ax=ax)
plt.title('Confusion matrix of LogisticRegression')
plt.show()
df2 = pd.DataFrame()
df2["y_test"] = y_test
df2["predicted"] = predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20, 5))

```

```
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

IMPLEMENTING NAIVE BAYES ALGORITHM

```
#import library packages
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

# Load given dataset
data = pd.read_csv("water_potability.csv")

df = data.dropna()

df.columns

df.head()

#preprocessing, split test and dataset, split response variable
X = df.drop(labels='Potability', axis=1)
#Response variable
y = df.loc[:, 'Potability']

#Splitting for train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42,
stratify=y)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))
Implementing LogisticRegression Algo

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
plot_confusion_matrix
Training

nb = GaussianNB()
nb.fit(X_train,y_train)
predicted = nb.predict(X_test)
Finding Accuracy

accuracy = accuracy_score(y_test,predicted)
print('Accuracy of GaussianNB',accuracy*100)
Finding Clasiification Report
```

```
cr = classification_report(y_test,predicted)
print('Classification report\n\n',cr)
Finding Confusion matrix
```

```
cm = confusion_matrix(y_test,predicted)
print('Confusion matrix\n\n',cm)
```

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,10))
plot_confusion_matrix(nb, X_test, y_test, ax=ax)
plt.title('Confusion matrix of GaussianNB')
plt.show()
```

```
df2 = pd.DataFrame()
df2["y_test"] = y_test
df2["predicted"] = predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()
```

Deploy

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import joblib
```

```
app = Flask(__name__)
model = joblib.load('RF.pkl')
```

```
@app.route('/')
def home():
    return render_template('index.html')
```

```
@app.route('/predict',methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = [(x) for x in request.form.values()]
    int_features = [int(i) for i in int_features]
    final_features = [np.array(int_features)]
    print(final_features)
    prediction = model.predict(final_features)
```

```

output = prediction[0]
print(output)
if output == 0:
    output = 'Drinkable'

    output = 'Not Drinkable'
return render_template('index.html', prediction_text='Water is : {}'.format(output))
__name__ == "__main__":
app.run(host="localhost", port=8000)

import pandas as pd
data = pd.read_csv('water_potability.csv')
data.isnull().sum()
df = data.dropna()
df.shape
df.isnull().sum()
df.info()
df.columns
df.duplicated()
df.duplicated().sum()
df.Conductivity.unique()
df.ph.unique()
df.Turbidity.unique()
df.Potability.unique()
df.Potability.value_counts()
df.columns
df.head()
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
df['Sulfate'].hist(figsize=(7,6), color='brown', alpha=0.7)
plt.xlabel('Sulfate')
plt.ylabel('No of Sulfate')
plt.title('Number of Sulfate')
#Histogram Plot
df['Turbidity'].hist(figsize=(7,6), color='pink', alpha=0.7)
plt.xlabel('Turbidity')
plt.ylabel('No of Turbidity')
plt.title('Number of Turbidity')
Propagation by variable
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' \n', fontsize = 15)
    return n.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'Potability')
plt.boxplot(df['Trihalomethanes'])

```

```

.boxplot(df['ph'])
plt.show()
# Heatmap plot diagram
fig, ax =
plt.subplots(figsize=(15,10))
s.heatmap(df.corr(),
annot=True) fig, ax =
plt.subplots(figsize=(16,8))
ax.scatter(df['Solids'],df['Conducti
vity']) ax.set_xlabel('Solids')
ax.set_ylabel('Conductivity')
plt.show()
preprocessing, split test and dataset, split response
variable X = df.drop(labels='Potability', axis=1)
#Response
variable y =
df.loc[:, 'Potability
']
#We'll use a test size of 30%. We also stratify the split on the response
variable, which is very important to do because there are so few fraudulent
transactions. from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, plot_confusion_matrix
rf =
RandomForestClassifier(
) rf.fit(X_train,y_train)
predicted =
rf.predict(X_test)
accuracy = accuracy_score(y_test,predicted)
print('Accuracy of
RandomForestClassifier',accuracy*100) cr =
classification_report(y_test,predicted)
print('Classification report\n\n',cr)
cm = confusion_matrix(y_test,predicted)
print('Confusion matrix\n\n',cm)
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,10))
plot_confusion_matrix(rf, X_test, y_test, ax=ax)
plt.title('Confusion matrix of RandomForestClassifier')
plt.show()
df2 = pd.DataFrame()
df2["y_test"] = y_test
df2["predicted"] =
predicted
df2.reset_index(inplace=

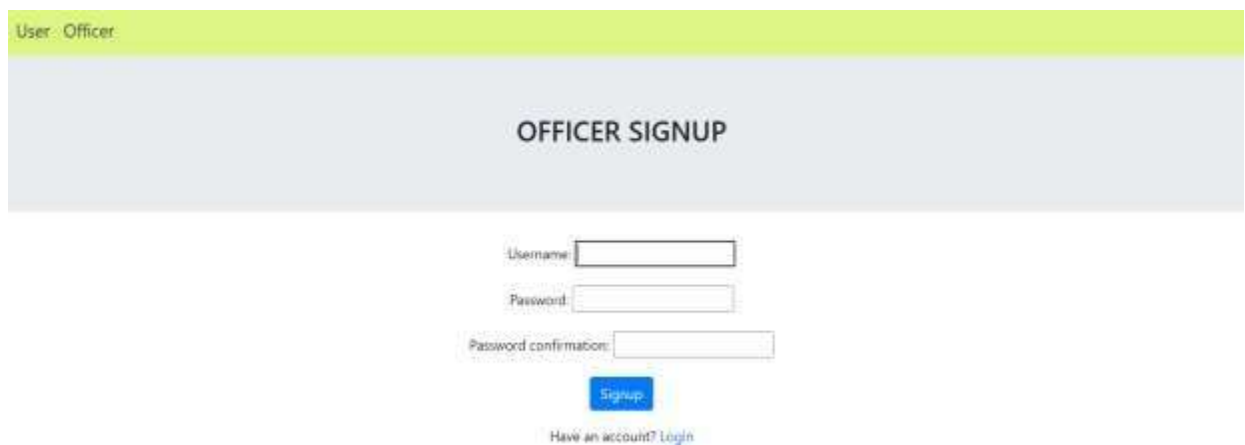
```

C. SCREEN SHOTS



The screenshot shows a web page with a light green header bar containing a "Logout" link on the right. Below the header is a light blue section with the title "USER SIGNUP" in bold. The main content area is white and contains a signup form with three input fields: "Username:", "Password:", and "Password confirmation:". Below these fields is a blue "Signup" button. At the bottom of the form, there is a link that says "Have an account? Login".

Sign up page for User



The screenshot shows a web page with a light green header bar containing the text "User - Officer" on the left. Below the header is a light blue section with the title "OFFICER SIGNUP" in bold. The main content area is white and contains a signup form with three input fields: "Username:", "Password:", and "Password confirmation:". Below these fields is a blue "Signup" button. At the bottom of the form, there is a link that says "Have an account? Login".

Sign up page for Officer

Logout

USER LOGIN

Username:

user

Password:

Login

Dont have an account? [signup](#)

User login page

Officer

OFFICER LOGIN

Username:

office12

Password:

Login

Dont have an account? [signup](#)

Officer login page

Enter Details
Feedback

WATER QUALITY PREDICTION

Name

Address

Email

Phone

Chloramines

Sulfate

Conductivity

Organic_carbon

Trihalomethanes

Home page for framework

pH

Hardness

Solids

Trihalomethanes

Turbidity

The water is Not drinkable

Prediction of Water Quality

Enter Details	Logout
Feedback	

OFFICER FEEDBACK

Will check and take action

Feedback in User Page

All	Last	Logout
-----	------	--------

Name	Address	Email	phone	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic carbon	Trihalomethanes	Turbidity	result
USER	123street	user@gmail.com	1234567890	9	8	9	9	6	8	5	7	5	The water is Not drinkable

Feedback:

Will check and take action

Submit

Feedback Submission in Officer Side

All	Last	Logout
-----	------	--------

Name	Address	Email	phone	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic carbon	Trihalomethanes	Turbidity	result
ABC	No:20, Velankulam, Chennai-35	abc@gmail.com	4525245254	78	54	895	1	56	47	12	45	15	The water is Not drinkable
DEF	No:20, Velankulam, Chennai-35	abc@gmail.com	4525245254	4	4	4	4	4	4	4	4	4	The water is Not drinkable
yogeshwari	45	yogeshwari@gmail.com	098765431	4	78	12	45	75	63	71	42	7	The water is Not drinkable
a	a	abc@gmail.com	4525245254	4	0	0	0	0	0	0	0	0	The water is Not drinkable

user	location	mail@gmail.com	1234567890	7.5	0	0	0	0	0	00	0	0	The water is Not drinkable
Anubama J	1234 Chennai	12345@gmail.com	1234567891	4	5	8	5	3	2	8	4	1	The water is Not drinkable
User	no12_chennai	user@gmail.com	0987654321	5	24	23	23	43	5	8	9	0	The water is Not drinkable

Past User Data Records

Water Quality Prediction Using Catboost Classifier Algorithm

A Yogeshwari¹
Dept. of Information Technology
Sathyabama Institute of Science
and Technology
Chennai 119,
India

J Anubama²
Dept. of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119,
India

Mrs. J. Merlin Mary Jenitha³
Assistant Professor
Dept. of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119,
India

Mrs. C. Geetha⁴
Assistant Professor
Dept. of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119,
India

Mrs. D. Ramalakshmi⁵
Assistant Professor
Dept. of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119,
India

B.Pavithra⁶
Assistant Professor
Department of Computer
Science & Engg.,
Sathyabama Institute of
Science and Technology
Chennai 119,
India

Abstract— Human existence depends on water, so its purity must be assessed for improved consumption. The goal is to use a cat-boost algorithm to forecast the water quality findings as accurately as possible. It is anticipated that improving water quality will lessen health-related issues. By gathering a variety of data, including variable detection and analysis methods like univariate, bivariate, and multivariate analysis, the study of the water quality is carried out. The study compares and discusses how different algorithms function. Data gathering is the first step in the process, during which historical information about the water condition is gathered. With both dependent and independent factors, data analysis is performed. The main objective is to identify the physicochemical characteristics of water, including temperature, pH, EC, hardness, chlorides, alkalinity, phosphate, and sulfate in water samples taken from various monitoring sites. To forecast the trend of data and the quality of the water, the appropriate algorithm is used. The structure has logins for users and government officers, with the latter having access to submit complaints to officials. In order for the police to quickly locate the user, the user is also permitted to provide information about the water purity in addition to their residential data.

Keywords—water quality, Prediction, Accuracy, Data-Preprocessing, pH, Data Visualization, Cat boost Algorithm

I. INTRODUCTION

Water is the basic need of Human life. It has a direct impact to the health of human and to eco system. Water is the primary source in human activities. Water quality assessment is an important step in the development of agricultural endeavors in terms of trimming design, type of water system framework, and water refinement frameworks for business. Water quality is determined by various factors such as pH, temperature, nitrate, dissolved oxygen (DO), electrical conductivity, sediment, and cellular oxygen consumption are all variables that must be considered. (BOD). In many areas of the world, aquatic insects move and spread diseases

like dengue fever, diarrhea, and schistosomiasis which is very harmful to the human health and also for the environment. water are not just affect the human health but also infect the society causing genetical and hereditary problems resulting a great impact in the human culture. In order to prevent these medical impacts and to safeguard human wellbeing the quality of water must be analyzed

II. MATERIALS ANDMETHODS

The proposed strategy entails creating a framework for the categorization of water quality and for a user-friendly report system. The procedure begins with data collection, where previous data connected to water quality

measures are gathered. Water purity is discovered beforehand, so it can reduce issues in human health. Machine learning is now being used to minimise manual error, which can improve the precision in forecasting the quality of water using different measures collected in the dataset. Data analysis is performed on the information, and appropriate variable identification is performed where both dependent and independent variables are discovered. Following that, the Cat boost Algorithm is used to forecast water purity.

III. PROPOSED METHOD

The proposed system has seven modules namely Data Pre-processing, Data Analysis, Data Visualization, Implementing Cat Boost Algorithm, Implementing Logistic Regression Algorithm, Implementing Random Forest Algorithm, Implementing Naïve Bayer's Algorithm and Deployment of the system.

[a] *Data Pre -Processing*: The process of preparing and assessing data for training. This program collects data from various sources and then removes null values, useless values, and incorrect data. This module processes the water in different areas such as pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic- carbon, Trihalomethanes, and Turbidity. The Water Quality for the specified categories is analyzed within a certain limit.

[b] *Table I – water components and its portable range*

WATER COMPONENT	ADEQUATE RANGE
ph	6.5 to 8.5
Hardness	120 to 170 Mg
Solids	50 to 150
Chloramines	4 Mg/l or 4ppm
Sulfate	500 Mg/l
Organic carbon	Below0.500 Mg/l TOC
Trihalomethanes	100 ppb
Turbidity	Below 1 TOC
Conductivity	200 to 1000 mu/cm

[1] *pH* :The pH level of the water indicates whether the water is acidic or basic with

the range of 0 to 14. The pH numbers less than 7 suggest acidity, while pH values greater than 7 indicate baseness. The pH of water is actually a determination of the amount of free hydrogen and hydroxyl molecules. Water that contains more free hydroxyl ions is basic, whereas water that contains more free hydrogen ions is acidic. Each figure represents a 10-fold variation in the acidity or basicity of the water. Water with a pH of five is ten times more corrosive than water with a pH of six.

Table II – pH of different water

water compound	ph level
Osmotic water	5 to 7
Faucet water	7.5
Alkaline water	8 to 9
Sea Water	>8
Acid	5 to 5.5

Figure I- pH level

SULFURIC ACID	GASTRIC ACID	CITRIC ACID	SODIUM CARBONATE	OXALIC ACID	BEVERAGES FROM PLANTATION CROPS	LACTIC ACID	DI-HYDROGEN OXIDE
0	1	2	3	4	5	6	7
PESTICIDES FERTILIZER BATTERIES	STOMACH ACID	LEMON	SODA	TOMATO SPINACH	COFFEE TEA	MILK	WATER
							

ALKALINE WATER	SODIUM HYDROXIDE CARBONATE	WATERBORNE HYDROXIDE	PH SOLUTION	ACIDIC OR POTASSIUM SALTS OF FATTY ACIDS	SODIUM HYPOCHLORITE	SODIUM HYDROXIDE
8	9	10	11	12	13	14
BLOOD SEA WATER	Egg WHITE SHAVING FOAM	GREAT SALT LAKE MILK OF MAGNESIA	AMMONIA SOLUTIONS	SOAPS DISH WASHERS	BLEACH	DRAIN CLEANER
						
ALKALINE						

[2] *Hardness*: The high concentration of calcium and magnesium refers to the hardness of water. The overall hardness (mg/L) is the aggregate of the calcium and magnesium concentrations, both given as calcium carbonate which is used to measure water hardness. The metal ions in hard water precipitate out of solution as insoluble precipitates, making them inaccessible to living creatures .

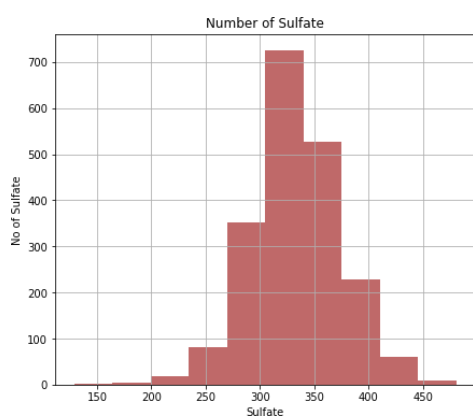
[3] *Solids* :Solids in water refers dissolved particles as well as floating and settleable particles. An organism swells when placed in water with few solids, such as distilled water is

particularly worrisome when pesticides are involved. Pesticide concentrations may increase considerably above those of the original application as irrigation water travels through irrigation channels in regions with a lot of solids. Increased solids ratios can block irrigation systems and cause watered plant roots to lose water instead of receiving it. Total solids are formed as a result of industrial pollution, sewage, fertilizers, fertiliser runoff from roads, and soil erosion.

[4] *Chloramines*: The method of disinfecting and eliminating pathogens from drinking water by adding chloramine. Monochloramine is a form of chloramine used to disinfect drinking water. Chloramine lasts longer in water pipelines and produces fewer disinfection residues. Some water providers are switching to chloramine to comply with EPA regulations that limit product disinfection.

[5] *Sulfate*: Water containing more than 250 mg/L sulphate may have a bitter or medicinal flavour. More corrosion-resistant conduit components, such as polyethylene lines, are commonly used in areas with high sulphate levels. Sulfate elements can be found in dirt and geological layers, and some of them dissolve into groundwater as it travels through. Epsom salt, sodium sulphate, and calcium sulphate are all elements that contain sulphate. (gypsum). Sulfate levels in groundwater in Minnesota are usually low—less than 250 mg/L. The southwest and the state's western border have greater sulphate amounts. Sulfate levels are also high in some wells in the state's northeastern and southeastern regions, though less frequently.

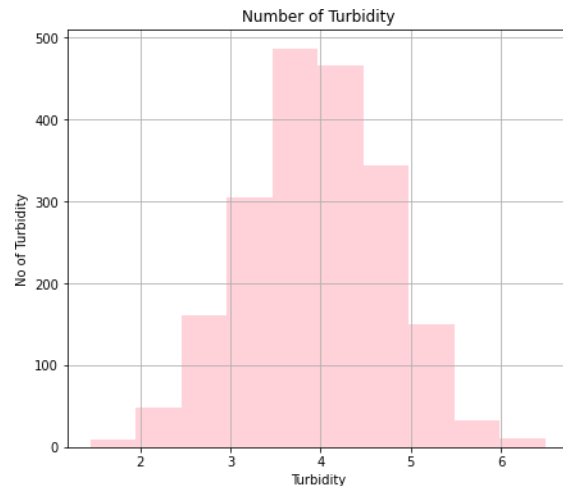
Figure II



[6] *Turbidity*: Turbidity refers to the measure of the transparency of a liquid. When light is transmitted through a water sample, the quantity of light scattered by the components of water is

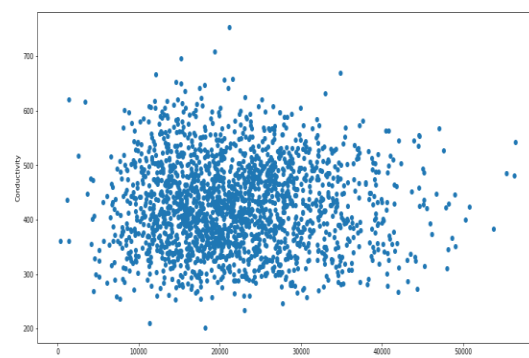
measured. Turbid water looks hazy or murky due to particulates that are dissolved or suspended in it and disperse light. Particulate matter includes inorganic materials, soluble colored organic compounds, cyanobacteria, and other microscopic creatures.

Figure III



[7] *Conductivity*: The conductivity of water is a measurement of its ability to transport an electrical charge. Because electrical current can pass through dissolved salts and other inorganic compounds, conductivity increases with saltiness. Water conductivity is important because it can reveal the amounts of dissolved substances, chemicals, and elements in a fluid. The salt and conductivity of the water can be used to identify what is dissolved in it. A higher conductivity number in the SWMP data indicates that there are more compounds dissolved in the water. transmission is used to quantify electrical transmission in water.

Figure IV



[c] *Data*

Visualization: The visual representation of data is known as data visualization. By utilizing graphic

components such as charts and graphs, data visualization tools make it simple to spot and assess trends, anomalies, and patterns in data.

Figure V

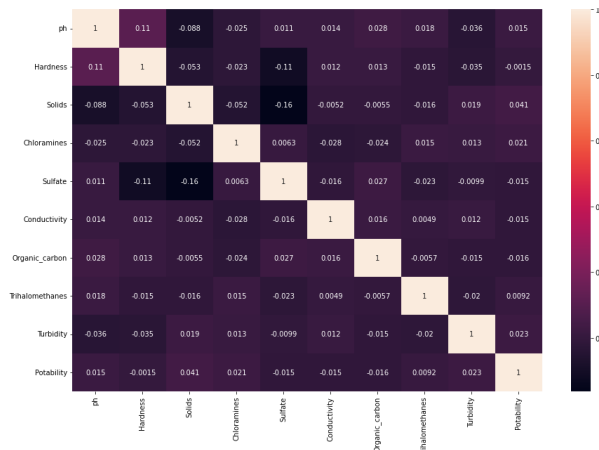
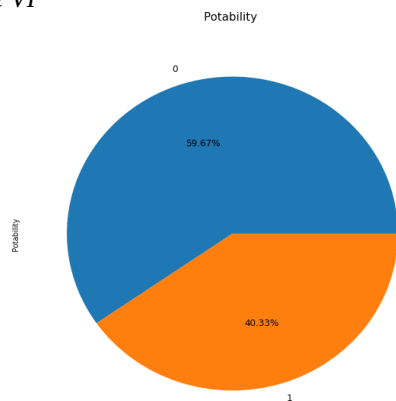


Figure VI



[d] *Random Forest Algorithm*: The Random Forest can do both classification and regression which can manage large databases with many variables to improve the model's precision and prevents overfitting. The Increase in large amount of data to find more accuracy might slow down the performance of the model.

[e] *Naïve Bayes's Algorithm*: The Naive Bayes classifier is a quick, precise, and dependable technique that is used because it has high precision and speed on large datasets but the premise of these distinct predictor parameter, which implies that all of the parameter are independent of one another parameter

[f] *Logistic Rsegression*: Logistic regression

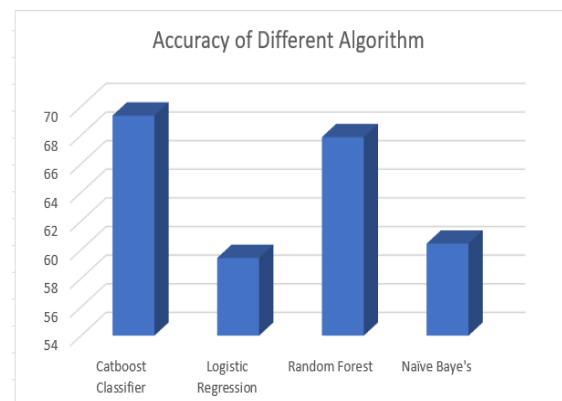
is a technique for predicting the likelihood of an objective variable which has only two potential groups. The dependent variable is binary in character, with data recorded as either 1 (for success/yes) or 0 (for failure/no).

[g] *Cat boost Algorithm*: Cat Boost is a combination of the words "Category" and "Boosting" which is a categorical-based program. This implies the method of converting categorical values to numerical values by using various statistics on categorical feature pairings and categorical and numerical feature combinations.

Benefits of CatBoost:

The main advantage of using cat boost classifier algorithm is because of its efficiency to deal with categorical features. It's also rapid, simple to use, precise and incorporating categorical factors into your feature collection for better accuracy.

IV. RESULTS AND DISCUSSION



Water purity cannot be determined solely by one aspect. Because the catboost classifier deals with data classification and the data set contains numerous data factors such as pH, hardness, solidity, turbidity, chloramines, organic compounds, and sulfate, catboost is evaluated with the highest level of precision when compared to other algorithms.

V. CONCLUSION

The models investigate the feasibility of using Supervised Machine Learning techniques such as the Catboost Algorithm for predicting the quality of water and to deploy a framework for a water complaint system. The Catboost Algorithm is used in the implementation of a system to improve the precision of water purity, resulting in better health and cost reductions. The purity of water is expected

to reduce the number of problems that arise.

VI. REFERENCES

- [1] A. N. Hasan and K. M. Alhammadi, "Quality Monitoring of Abu Dhabi Drinking Water Using Machine Learning Classifiers," 2021 14th International Conference on Developments in eSystems Engineering (DeSE), 2021
- [2] Ajayi 1, Antoine B. Bagula1 ,Hloniphani C. Maluleke 1 , ZaheedGaffoor 2 , Nebo Jovanovic2 , And Kevin Bellville, Cape Town 7535, South Africa 2Department of Earth Science, South Africa- WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation. Conference on Research in Intelligent and Computing in Engineering (RICE), 2018
- [3] Dao Nguyen Khoi, Nguyen Trong Quan, Do
- [4] Di Wu; Hao Wang; Razak Seidu 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, - Tensor Model for Quality Analysis in Industrial Drinking Water Supply System Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, (DASC/PiCom/CBDCom/Cyber SciTech) Year: 2019 | Publisher: IEEE
- [5] G. Sebestyen, A. Hangan and Z. Czako, "Anomaly detection in water supply infrastructures systems," 2021 23rd International Conference on Control Systems and Computer Science (CSCS), 2021
- [6] Gaganjot Kang; Jerry Zeyu Gao; Gang Xie 2017- Data-Driven Water Quality Analysis and Prediction IEEE Third International Conference on Big Data Computing Service Applications (Big Data Service) Data Driven Water Quality Analysis and Prediction: Year: 2017 | Publisher: IEEE
- [7] Hao Wang; Hadi Mohammed; Razak Seidu- Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception Di Wu IEEE Transactions on Sustainable Computing Year: 2020 Publisher: IEEE
- [8] Huimin Jia; Xiaofeng Zhou – Water Quality Prediction Method Based on LSTM-BP - 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) Year: 2020 | Publisher: IEEE IEEE 36
- [9] Jian Zhou, Yuanyuan Wang, Fu Xiao, Yunyun Wang and Lijuan Sun - Water Quality Prediction Method Based on IGRA and LSTM which is proposed in the year 2018. Water quality prediction has great significance for water environment protection.
- [10] Jitha P Nair; M S Vijaya 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) Year: 2021 - Predictive Models for River Water Quality using Machine Learning and Big Data Techniques - A Survey Publisher: IEEE
- [11] K. P. Rasheed Abdul Haq; V. P. Harigovindan Year: 2022- Water Quality Prediction for Smart Aquaculture Using Hybrid Deep Learning Models
- [12] Md. Saikat Islam Khan, Nazrul Islam, Jia Uddin, Sifatul Islam, Mostofa Kamal Nasir in the year 2021 - A study on Water Quality Prediction and Classification based on Principal Component Regression and Gradient boosting Classifier approach
- [13] N. Kumar Koditala and P. Shekar Pandey, "Water Quality Monitoring System Using IoT and Machine Learning," 2018 International
- [14] N. M. Ragi, R. Holla and G. Manju, "Predicting Water Quality Parameters Using Machine Learning," 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2019
- [15] N. Radhakrishnan and A. S. Pillai, "Comparison of Water Quality Classification Models using Machine Learning," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020
- [16] P. Varalakshmi - Prediction of water quality using Naive Bayesian algorithm Year: 2017 | Publisher: IEEE
- [17] Priskilla Angel Rani; R. Sujeethra; C. Yesubai Rubavathi - Prediction of Water Quality using Artificial Neural Network Quang Linh, Pham Thi Thao Nhi, and Nguyen Thi Diem Thuy in the year 2022 – a study using Machine Learning Models for Predicting the Water Quality Index in the La Buong River, Vietnam
- [18] Quan Xi Dong; Yong Zhe Lin; Jing Bi; Haitao Yuan International Conference on Systems, Man and Cybernetics (SMC) Year: 2019 - An Integrated Deep Neural Network Approach for Large-Scale Water Quality Time Series Prediction Publisher: IEEE
- [19] Razak Seidu- Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception IEEE Transactions on Sustainable Computing Year: 2020 | Publisher: IEEE
- [20] S. J. Kale, M. A. Khandekar and S. D. Agashe, "Prediction of Water Filter Bed Backwash Time for Water Treatment Plant using Machine Learning

Algorithm," 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), 2021

- [21] SanazImen, Ni-Bin Chang, Senior Member, IEEE, Y. Jeffrey Yang, and ArashGolchubian Developing a Model-Based Drinking Water Decision Support System Featuring Remote Sensing and Fast LearningTechniques
- [22] Rongali Yang- Summary of Water Quality Prediction Models Based on