ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

UNIVERSITY COLLEGE OF ENGINEERING

MACHINE LEARNING

Water quality prediction using Machine Learning

On the completion of internship

Submitted by

POLIREDDY SUSHMA

Reg.no: 208297601042

Under the esteemed guidance of

Mrs. JAYANTHI HARINI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

2020-2024

ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

UNIVERSITY COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this internship report entitled, "**WATER QUALITY PREDICTION USING MACHINE LEARNING**" is a bonafide work of **Polireddy Sushma**, Reg.No:208297601042, submitted in the completion of internship, during the period 2020-2024. This work is carried out by him under my supervision and guidance and submitted to Department of Computer Science and Engineering, Adikavi Nannaya University.

INTERNSHIP/COURSE GUIDE **HEAD OF THE DEPARTMENT**

DATE:

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

CIN: U80902TN2021PTC141464
www.pantechelearning.com

PEHYD00034/INTERNSHIP/ML/2023-2024

COMPLETION CERTIFICATION

This is to certify that **Mr./Ms. POLIREDDY SUSHMA,** Roll Number **(208297601042),** who is pursuing **(COMPUTER SCIENCE AND ENGINEERING)** Department **ADIKAVI NANNAYA UNIVERSITY** has successfully completed his/her Internship at**, Pantech e Learning Pvt. Ltd** on **("Water Quality Prediction using Machine Learning")** under the guidance of **Mr./MS. P.SRUTHI REDDY**in our organization.

During the Internship period, the candidate has shown keen interest and commitment towards learning and his/her performance was good.

Duration of Internship From: 24/04/2023 to 17/06/2023

Yours truly,

Pantech e Learning Pvt.Ltd,

Pantech E Learning Pvt. Ltd. Hyd-18.

Srinivasan.N
(Branch Manager)

Pantech e Learning Pvt Ltd
4th Floor, Delta Chambers,
Behind Chennai Shopping Mall, Ameerpet, Hyderabad, Telangana – 500 016
Phone:91 040-40077960. | hr@pantechmail.com

DECLARATION

I certify that the work contained in this report is original and has been done by me under the guidance of my supervisor. The work has not been submitted to any other Institute for any degree or diploma. I have followed the guidelines provided by the university in preparing the report. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the student

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deep gratitude to the members who assisted us directly and indirectly for the completion of this project work. I feel fortunate to pursue my Bachelor Degree from the campus of Adikavi Nannaya University. It provided all the facilities in the area of Computer Science and Engineering. I profusely thank, **Dr.P. Venkateswara Rao** Associate Professor & Principal, University College of Engineering for all the encouragement and support. I also thank, Head of the Department, **Dr.B. Kezia Rani** Associate Professor of Computer Science and Engineering for the Guidance throughout the academic. I profusely thank **Mrs.CH.Harini** Assistant Professor, University College of Engineering for the guidance right from the beginning of the internship and her valuable suggestion for me to succeed in completing this project. I wish to express my deep sense of gratitude to the management of Pantech E-Learning Pvt.ltd, for giving me an opportunity to complete my "MACHINE LEARNING" course for the completion of internship. A great deal of thanks goes to review committee members and the entire faculty members for their support throughout the project.

P.SUSHMA(208297601042)

ABSTRACT

One of the key functions of global water resource management authorities is river water quality (WQ) assessment. A water quality index (WQI) is developed for water assessments considering numerous quality-related variables. WQI assessments typically take a long time and are prone to errors during sub-indices generation. This can be tackled through the latest machine learning (ML) techniques renowned for superior accuracy. In this study, water samples were taken from the wells in the study area (North Pakistan) to develop WQI prediction models. Few standalone algorithms, i.e., random trees (RT), random forest (RF), Decision tree, Support vector machine(SVM) and other classification algorithms are used. Water quality prediction involve classification techniques to predict the water potability. Classification is the most used technique that is being used for the prediction of potability of water which involve Logistic Regression and Support vector classifier techniques in this project. Different performance criteria are being employed in order to boost the performance of already existing ways.

INDEX

LIST OF FIGURES

INTRODUCTION

1.1 Problem Definition :

Water pollution is one of the critical challenges of the modern world. The global water crisis is the serious threat the human race faces these days. Water quality is need to be checked in certain regions whether the water is suitable for drinking or not i.e. water potability. Water quality is predicted by analyzing the soluble contents in the water. Classification techniques are used to predict the water quality

1.2 Scope of the problem:

A supervised machine learning predictive algorithm is consumed with the predefined collection of training data. The algorithm then gains expertise from the training dataset and produces rules for predicting the class label for a new data set.

Learning phases consists to use mathematical algorithms to generate and strengthen the predictor function. Training data used in this process has an attribute input value and its defined output value. The expected ML algorithm quality is compared with the often known output. This is repeated in much iteration of training data until the optimal prediction accuracy is reached or the upper limit number of loops is finished. In the field of unsupervised learning algorithms, the class label output value is not known in data. Alternatively, a cluster of data loads the software, and the algorithm identifies a pattern and relationships within it.

1.3 Objective of the project :

Water quality can be known by predicting the potability. Quality prediction is the method of designing models that are utilized in the initial stages of the process to predict systems such as units or classes. This can be achieved by classifying the modules as potable or not potable. Different methods are used to identify the classification module, the most common of which is support vector classifier (SVC) and Logistic Regression. The model performance is evaluated for every technique applied.

MACHINE LEARNING

2.1 What is Machine Learning?

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modelling. The machine receives data as input, use an algorithm to formulate answers. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

2.2 How does it work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if it's feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**.

Learning:

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

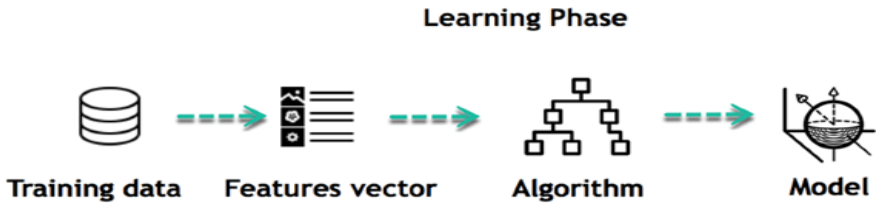


FIG 1: Learning Phase

Inference:

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

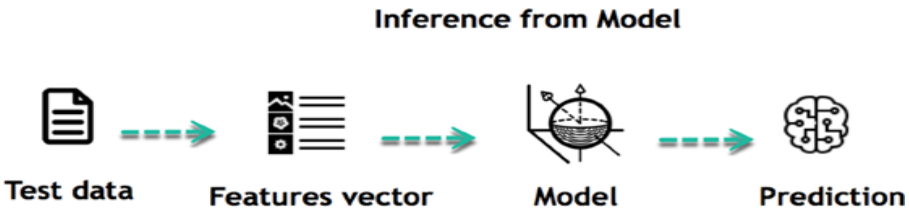


FIG 2: Inference from Model

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

2.3 Algorithms in Machine Learning:

Machine Learning algorithms are the programs that can learn the hidden patterns from the data, predict the output, and improve the performance from experiences on their own.

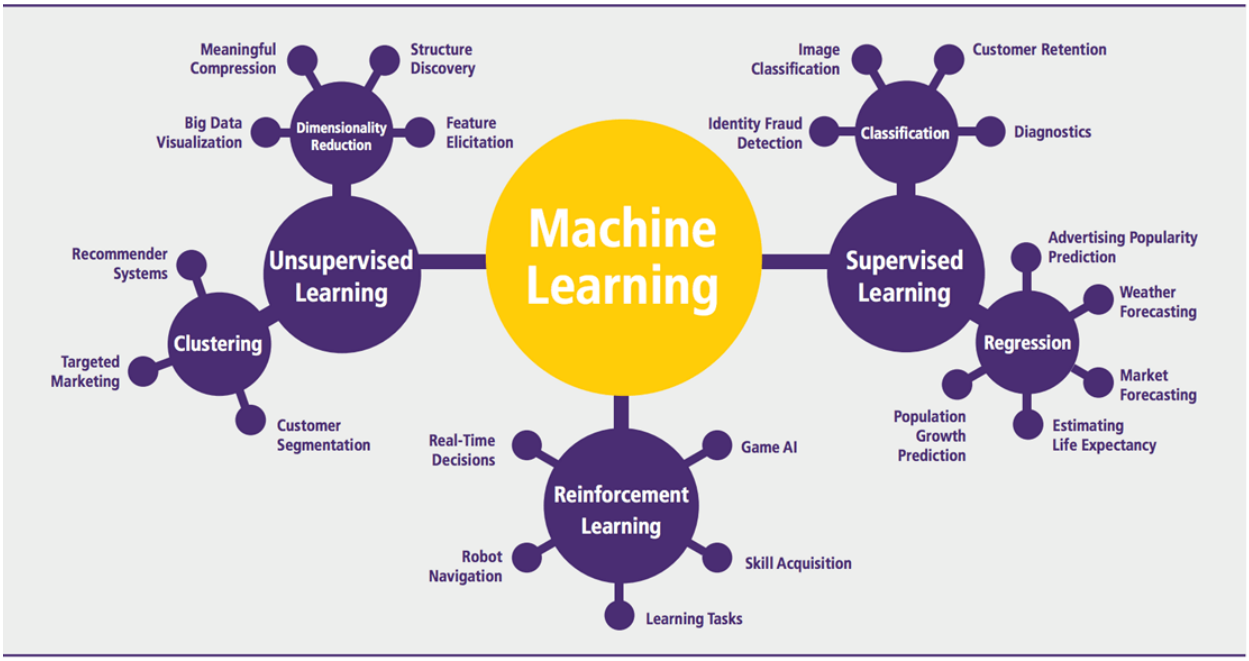


FIG 3: Machine Learning

Machine Learning Algorithm can be broadly classified into three types:

1. Supervised Learning Algorithms
2. Unsupervised Learning Algorithms
3. Reinforcement Learning algorithm

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification

Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a

class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering

PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

2.4 APPLICATIONS OF MACHINE LEARNING

Augmentation:

- Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation:

- Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

- Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

- The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

- Healthcare was one of the first industry to use machine learning with image detection.

Marketing

- Broad usage of AI is done in marketing.

PYTHON

3.1 What is Python?

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

1. Python is Interpreted
2. Python is Interactive
3. Python is Object-Oriented
4. Python is a Beginner's Language

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

3.2 Python Features:

1. **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2. **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
3. **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
4. **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
5. **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6. **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7. **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
8. **Databases:** Python provides interfaces to all major commercial databases.
9. **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of UNIX.

10. **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

3.3 Libraries that are used in this project:

1. Numpy : numerical calculations, array creations
2. Pandas : read the data (in any format excel ,csv,Json)
3. Matplotlib : graphical representation in 2D
    1. 1.legend()
    2. plt show()
4. Seaborn : data visualisation in 3D
5. Sklearn : implementation

    - we import the sci-kit
    - label encoder

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 Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

What applications can access using Anaconda Navigator:

1. Jupyter Lab
2. Jupyter Notebook
3. VS Code
4. Spyder
5. RS Studio etc.

SOFTWARE SPECIFICATIONS

4.1 System Requirements :

1. Operating System**:** Windows 7 Ultimate 32 bit / Windows XP

4.2 Hardware Requirements :

1. 4 GB RAM
2. 80 GB Hard Disk
3. Intel Processor
4. LAN

4.3 Software Requirements :

1. Windows OS
2. Python GUI or Anaconda Navigator

PROPOSED APPROACH STEPS

1. First, we take software dataset.

2. Filter dataset according to requirements and create a new dataset which has attribute according to analysis to be done

3. Perform Pre-Processing on the dataset

4. Split the data into training and testing

5. Train the model with training data then analyse testing dataset over classification algorithm

6. Finally you will get results as accuracy metrics

LEVEL 0

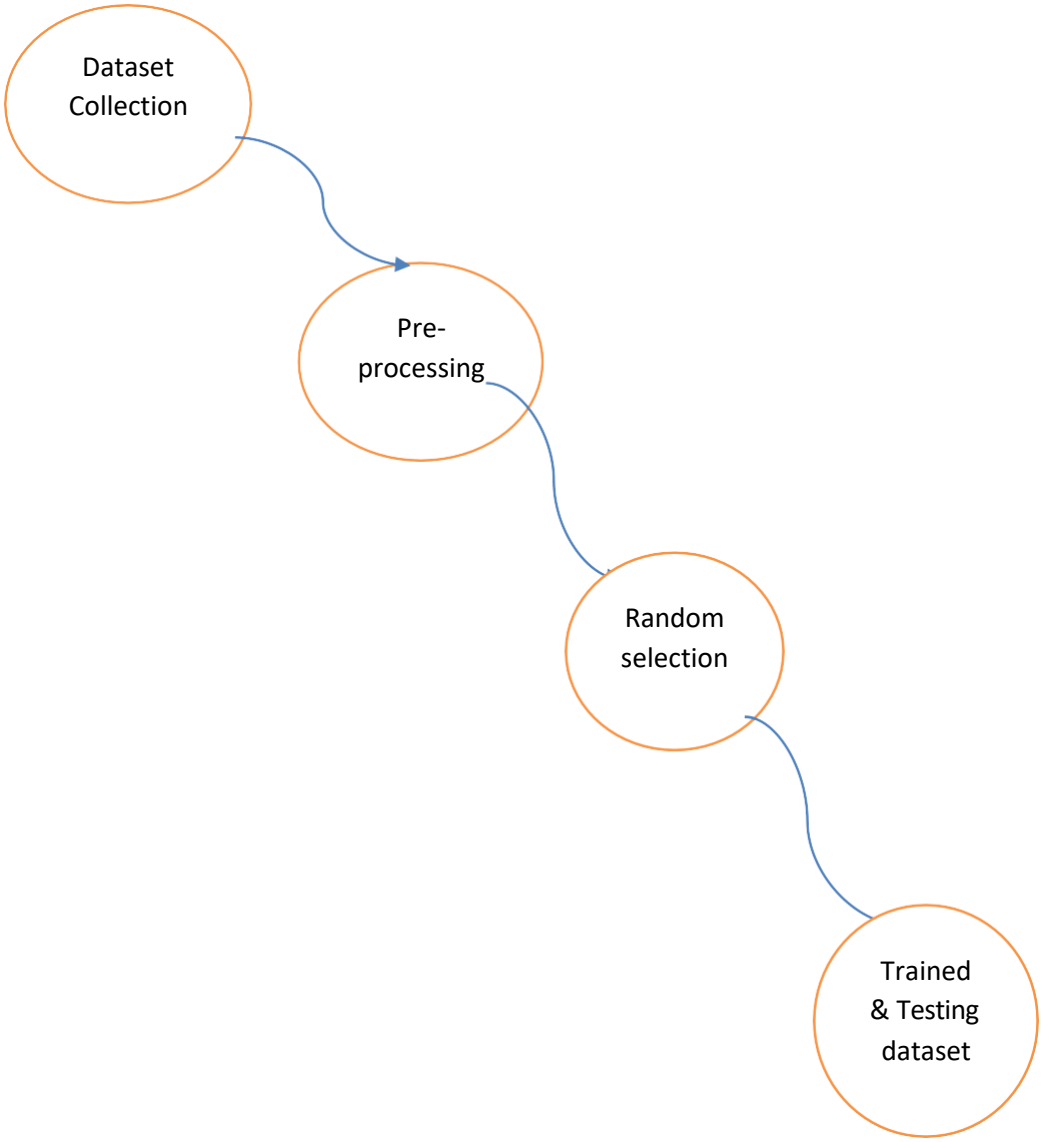


FIG 4: Level 0 in steps

LEVEL 1

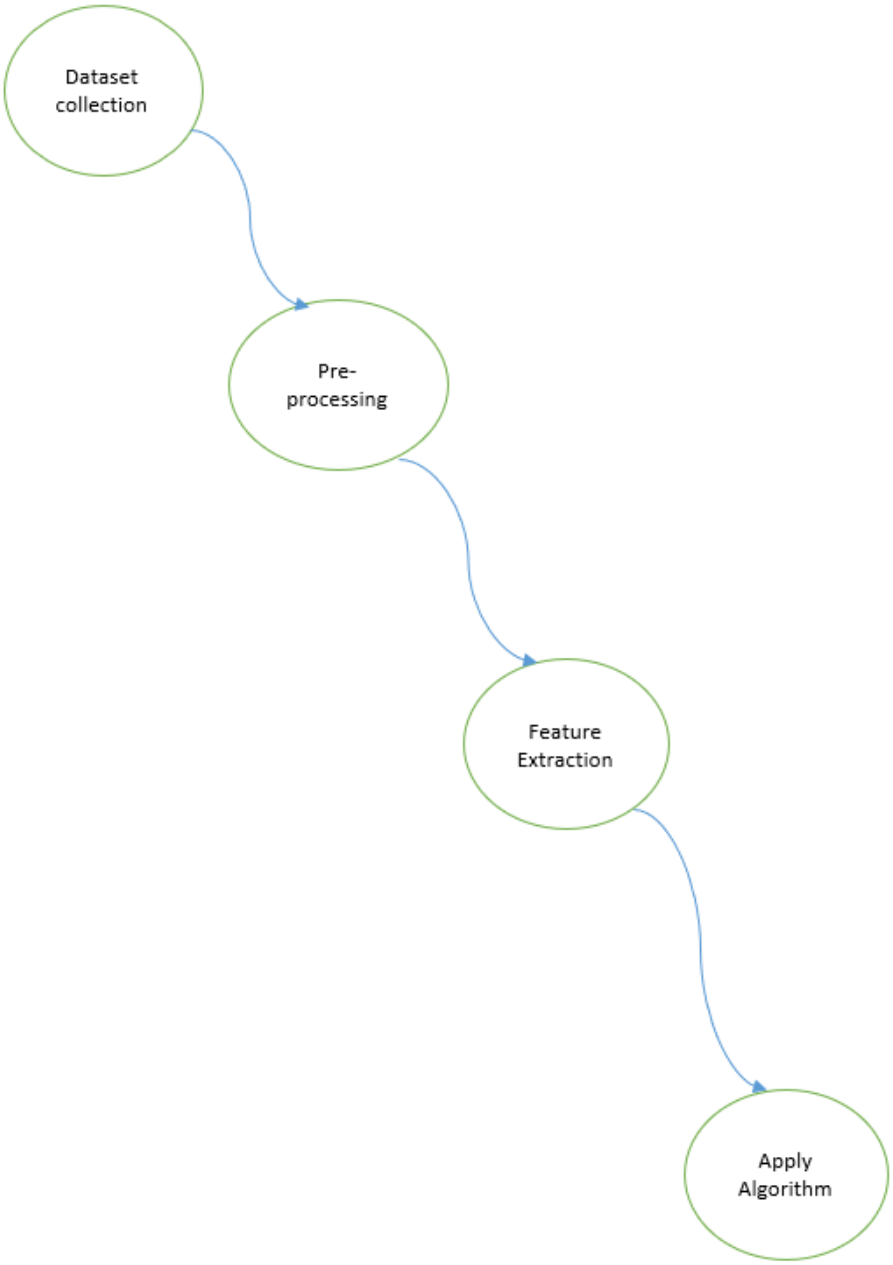


FIG 5: Level1 in steps

LEVEL 2

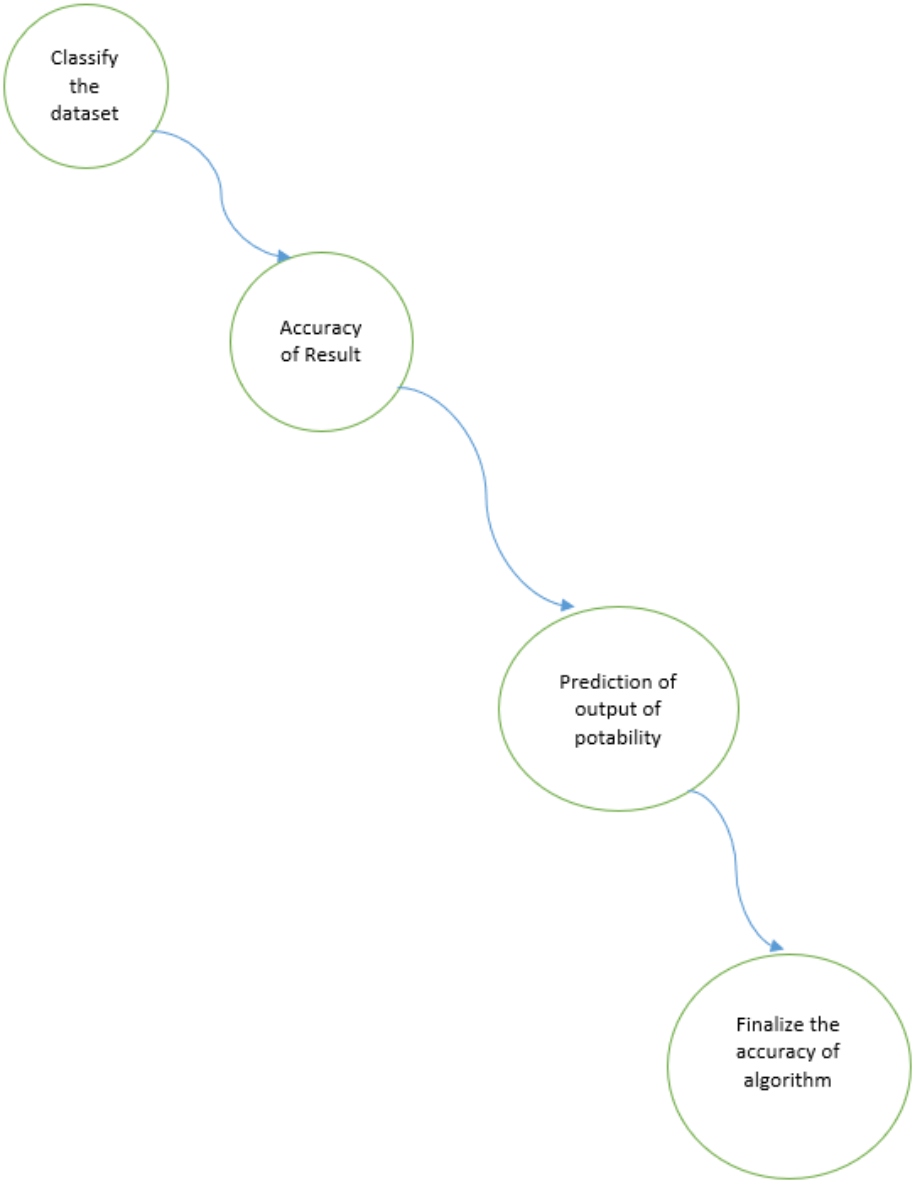


FIG 6: Level2 in steps

SYSTEM ARCHITECTURE

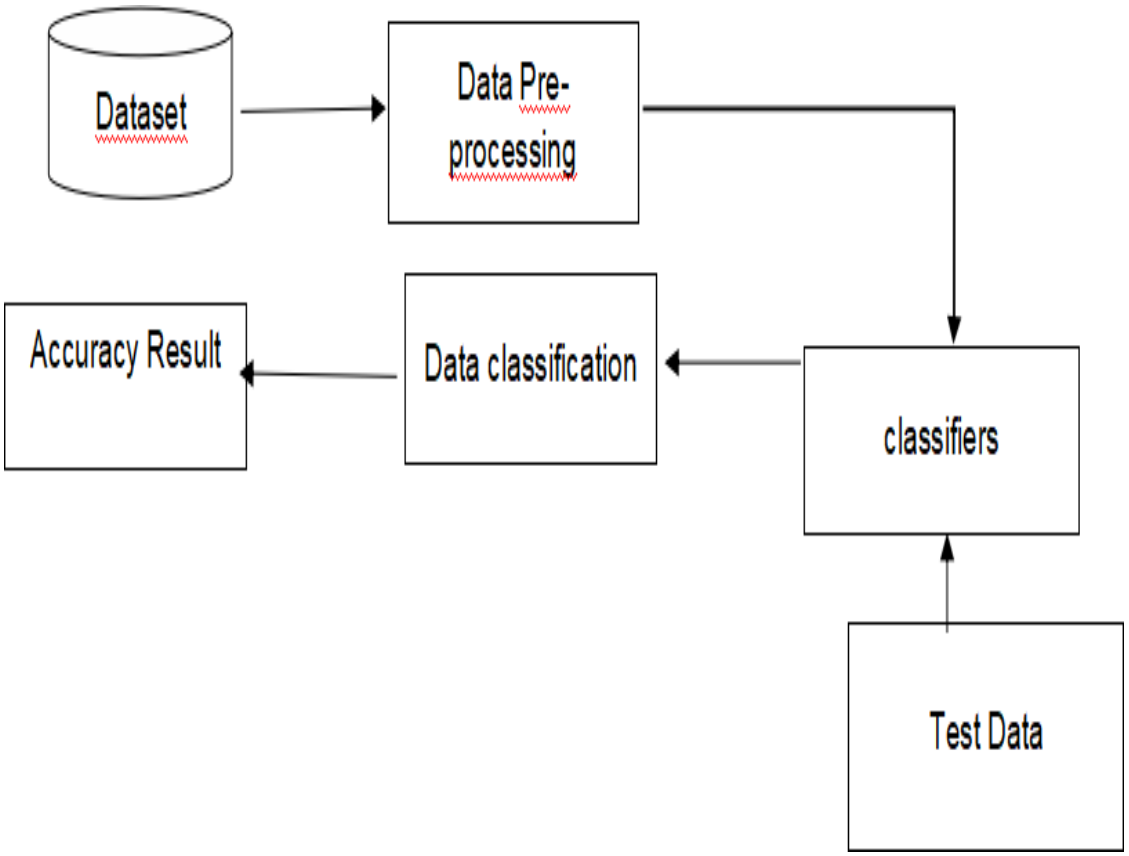


FIG 7: System Architecture

MODULES IN MACHINE LEARNING

1. DATA COLLECTION
2. DATA PRE-PROCESSING
3. FEATURE EXTRATION
4. EVALUATION MODEL

DATA COLLECTION:

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into

machine learning models.

Data used in this paper is a set of data with features . This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

It involves several stages, each contributing to a project's overall success and reliability. Here is a detailed explanation of each stage:

- **Defining the objective:** The first step is to define the objective well. It entails identifying the specific information required and the purpose for which it will be used. A well-defined objective helps maintain focus and ensures the collected data will be relevant and meaningful.
- **Planning the activity:** Once the objective is established, the next step is to plan how data will be collected. It involves determining the data sources, **sample size**, collection methods, and tools or instruments. The available resources, time constraints, and potential limitations are considered while planning.
- **Determining data sources:** Researchers identify the sources for data gathering. Depending on the objective, these sources can include surveys, interviews, observations, existing databases, or sensor data. Each source has advantages and limitations, so selecting the most appropriate ones for the study is important.
- **Selecting data collection methods:** Various methods are available for gathering data, and the choice depends on the nature of the information needed and the resources available. Common methods are surveys/questionnaires, interviews, observations, experiments, document analysis, and online data collection—method selection is based on its suitability.
- **Developing data collection instruments:** If surveys, questionnaires, or interview guides are used, researchers develop instruments and data collection tools that can be used in the data collection process. These instruments comprise relevant questions or prompts that elicit the required information. Pilot testing with a small group can help identify and address issues before full-scale data collection begins.
- **Sampling:** It involves selecting a subset of individuals, cases, or entities from a larger population from which data can be gathered. Sampling techniques depend on the research design and objectives. Common techniques are random sampling, **stratified sampling**, convenience sampling, and purposive sampling.

- **Training data collectors:** If multiple individuals are assigned to collect data, they must be trained on using collection methods, instruments, and protocols for consistency and accuracy. Training helps reduce bias, standardize procedures, and maintain data quality.
- **Collecting data:** The data collection process involves conducting surveys or interviews, noting observations, or gathering information from specific sources. Adhering to guidelines is key.

DATA PRE-PROCESSING:

Organize your selected data by formatting, cleaning and sampling from it.
Three common data pre-processing steps are:

- Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
- Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
- Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset

Three common data pre-processing steps are:

1. **Formatting**: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
2. **Cleaning**: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymzed or

removed from the data entirely.

3. **Sampling**: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

FEATURE EXTRATION:

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

EVALUATION MODEL:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models.

Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

SYSTEM ANALYSIS & DESCRIPTION

UML Diagrams:

In the field of software engineering, the Unified Modelling Language (UML) is a standardized visual specification language for object modelling. UML is a general-purpose modelling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model.

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- actors
- business processes
- (logical) components
- activities
- programming language statements
- database schemas, and
- Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

Importance of UML in Modelling:

A modelling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system. A modelling language such as UML is thus a standard

language for software blueprints.

1. Use case Diagram:

A use case diagram shows a set of use cases and actors and their relationships. We apply the use case diagram to illustrate the static use case view of a system. Use case diagram are especially important in organizing and modelling the behaviors of a system.

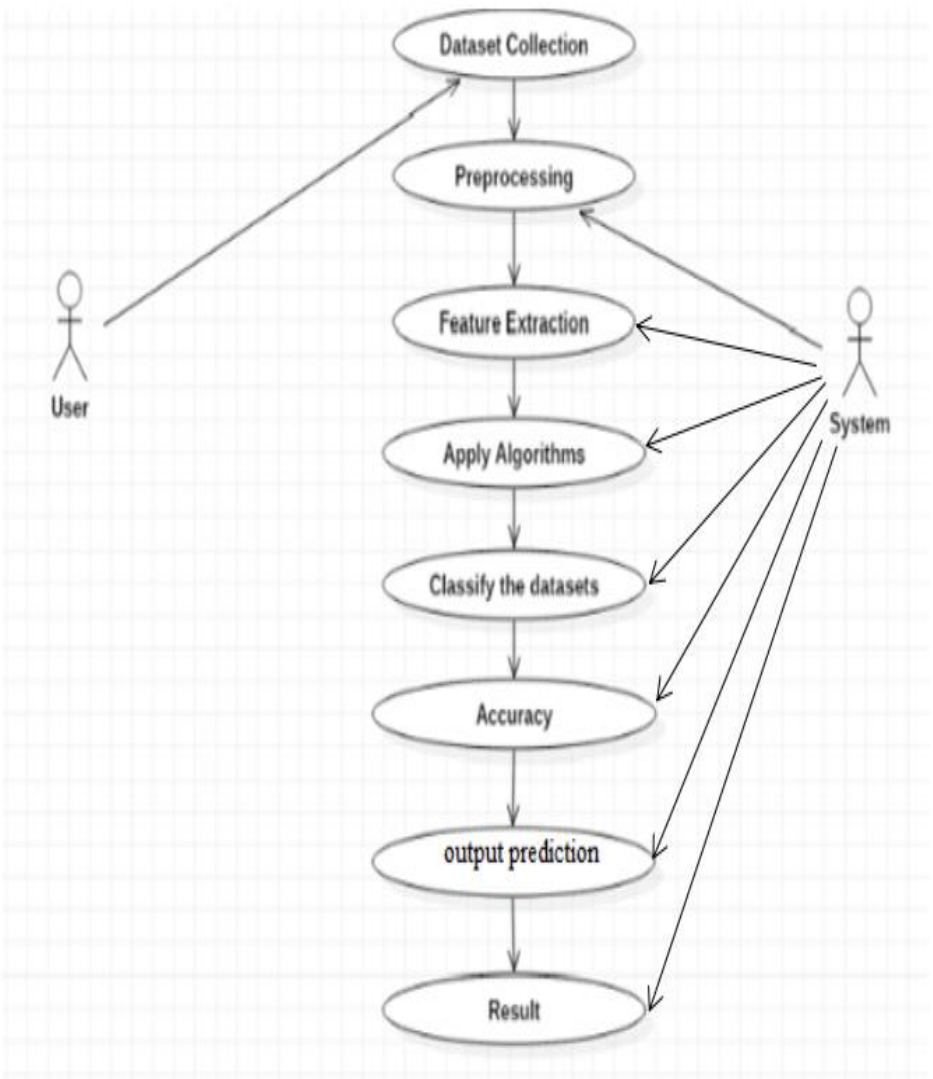


FIG 8: Use Case Diagram

2. **Class Diagram**:

Classes are the most important building block of any object oriented system. A class is a description of set of objects that share the same attributes, operations, relationships and semantics. A class implements one or more interfaces. It is graphically rendered as a rectangle.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

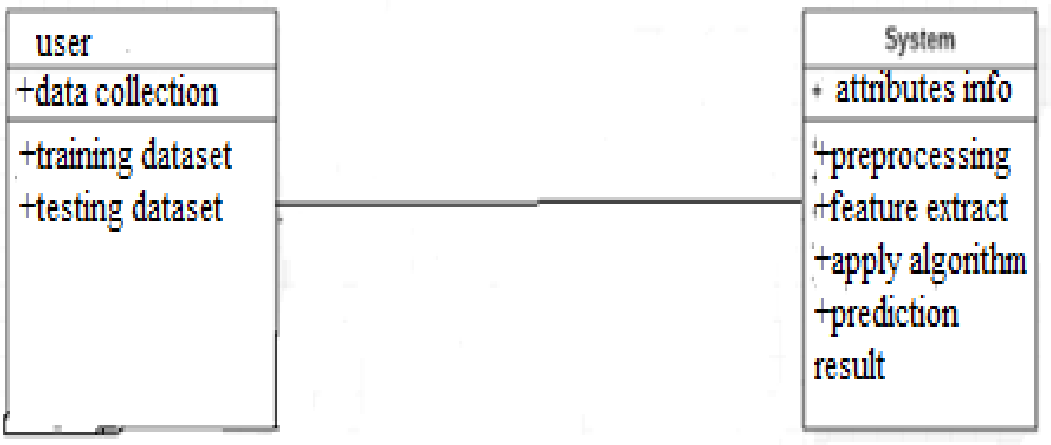


FIG 9: Class Diagram

3. **Sequence Diagram**:

A sequence diagram emphasizes the time ordering of messages. These are used to model the dynamic aspects of the system. A sequence diagram shows a set of objects and messages that are dispatched between those objects based on time-ordering.

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment.

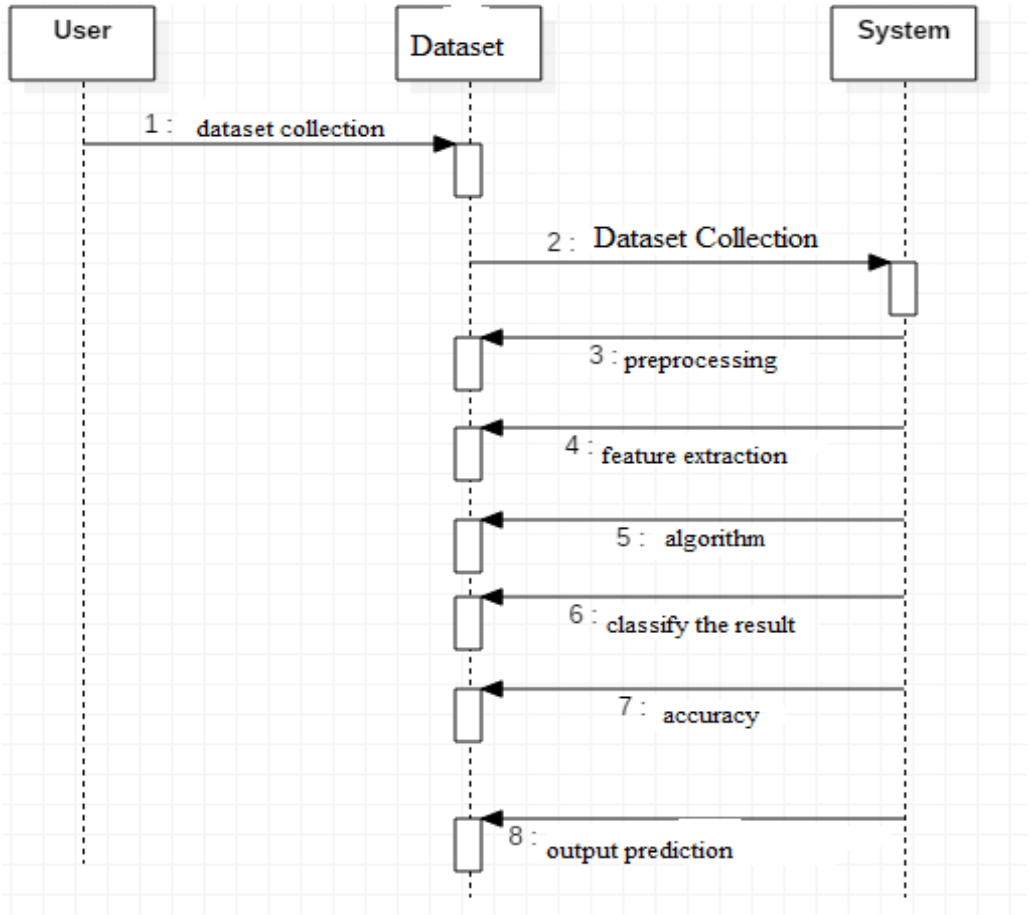


FIG 10: Sequence Diagram

3. **Activity Diagram:**

The purpose of an activity diagram is to provide a view of flows and what is going on inside a use case or among several classes. An activity diagram is just to explain the internal operations performed and also the transitions that are triggered by the completion of the particular operations. At the abstract level it explains the sequence of the activities.

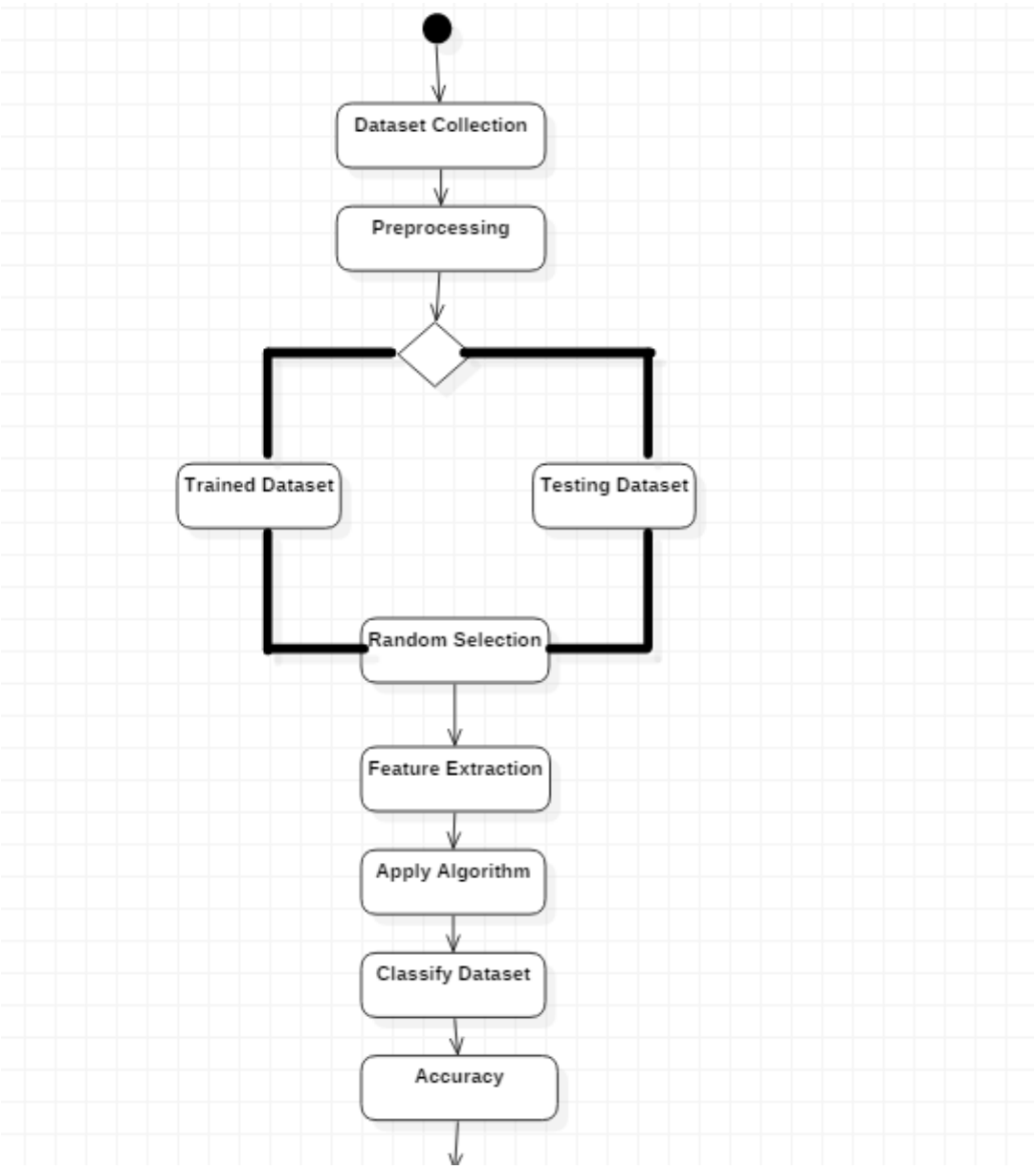


FIG 11: Activity Diagram

DESIGN METHODOLOGY

Proposed model of this project:

1. Logistic Regression

2. Support Vector Classifier

Support Vector Machine:

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for either classification or regression tasks. It is generally utilized in characterization issues. In the SVM calculation, we plot every datum thing as a point in n-dimensional space (where n is number of highlights you have) with the estimation of each element being the estimation of a specific arrange. At that point, we perform order by finding the hyper-plane that separates the two classes quite well. Bolster Vectors are essentially the co-ordinates of individual perception. The SVM classifier is a wilderness which best isolates the two classes (hyper-plane/line)

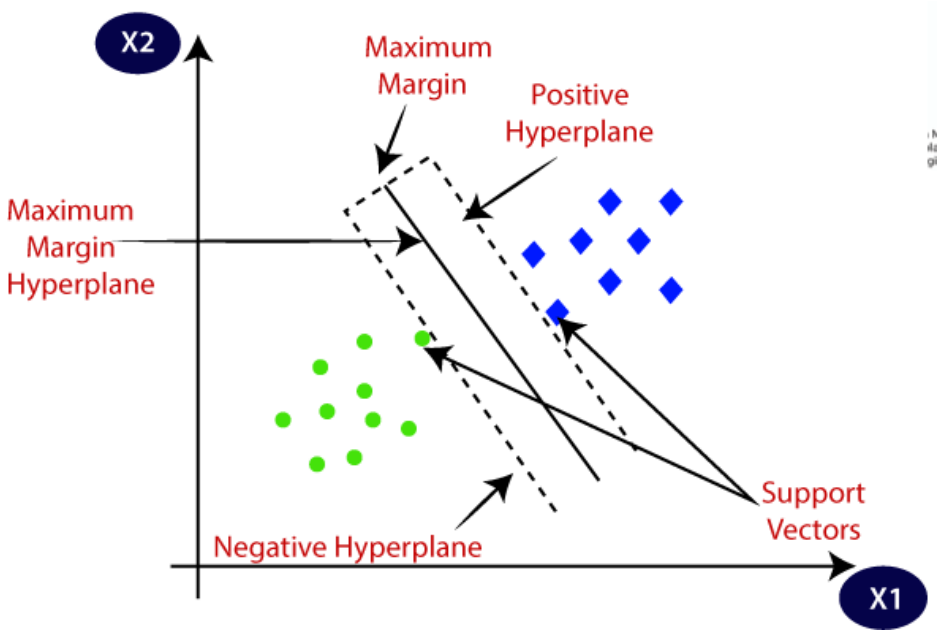


FIG 12: Support Vector Machine

The main objective of SVM is to find a hyper plane that best separates the data points into different classes while maximizing the margin between the classes.

HOW SVM WORKS FOR BINARY CLSSIFICATION:

1. Data Representation:

SVM requires labelled data for training, where each data point is represented as feature vector in an n-dimensional space, where n is the number features. For binary classification, each data point should be associated with one of two classes: positive class (1) or negative class (-1).

2. Finding the Optimal Hyper plane:

The optimal hyper plane is the one that maximizes the margin between the two classes, i.e., the distance between the hyper plane and the closest data points from each class, known as support vectors. The hyper plane can be represented as: w * x + b = 0, where w is the weight vector (perpendicular to the hyper plane) and b is the bias term.

3. Maximizing Margin:

SVM aims to maximize the margin, which is the width between the parallel hyper planes that are closest to the support vectors of each class. These parallel hyper planes can be represented as: w * x + b = 1 (for the positive class) and w * x + b = -1 (for the negative class).

4. Optimization Problem:

The optimization problem in SVM involves finding the best values for the weight vector w and bias term b that maximizes the margin while satisfying the constraint that all data points are correctly classified. This optimization problem is typically formulated as a convex quadratic programming problem.

5. Kernel Trick (Optional):

In cases where the data is not linearly separable, SVM can use the kernel trick to transform the data into a higher-dimensional feature space where it becomes linearly separable. The most common kernel functions are the linear kernel, polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel.

6. Regularization Parameter (C):

SVM has a regularization parameter C, which controls the trade-off between maximizing the margin and minimizing the classification error on the training data. A smaller C value allows

for a wider margin but may lead to misclassifications, while a larger C value results in a narrower margin but fewer misclassifications.

7. Prediction:

To classify new data points, SVM computes the sign of the decision function (w * x + b). If the result is positive, the data point belongs to the positive class, and if it's negative, the data point belongs to the negative class.

8. Handling Multi-Class Problems:

SVM is inherently a binary classifier, but it can be extended to handle multi-class problems using techniques like one-vs-one or one-vs-rest.

SVM is widely used for a variety of tasks, especially when the number of features is relatively small compared to the number of data points, as it tends to perform well in such scenarios.

Support vector classifier

A Support Vector Classifier (SVC) is a type of supervised machine learning algorithm used for classification tasks. It belongs to a broader class of algorithms known as Support Vector Machines (SVM).

The SVC algorithm works by finding the optimal hyperplane that best separates the different classes in the input data. The hyperplane is a decision boundary that maximizes the margin between the classes. The data points closest to the hyperplane are called support vectors, and they play a crucial role in determining the position and orientation of the hyperplane.

The SVC algorithm is particularly effective when dealing with high-dimensional datasets and complex decision boundaries. It can handle both linearly separable and non-linearly separable data by using different mathematical techniques, such as the kernel trick, which maps the original data into a higher-dimensional space.

During the training phase, the SVC algorithm learns the parameters of the hyperplane by solving an optimization problem. The goal is to minimize the classification error while maximizing the margin between the classes. The algorithm can handle both binary and multiclass classification problems.

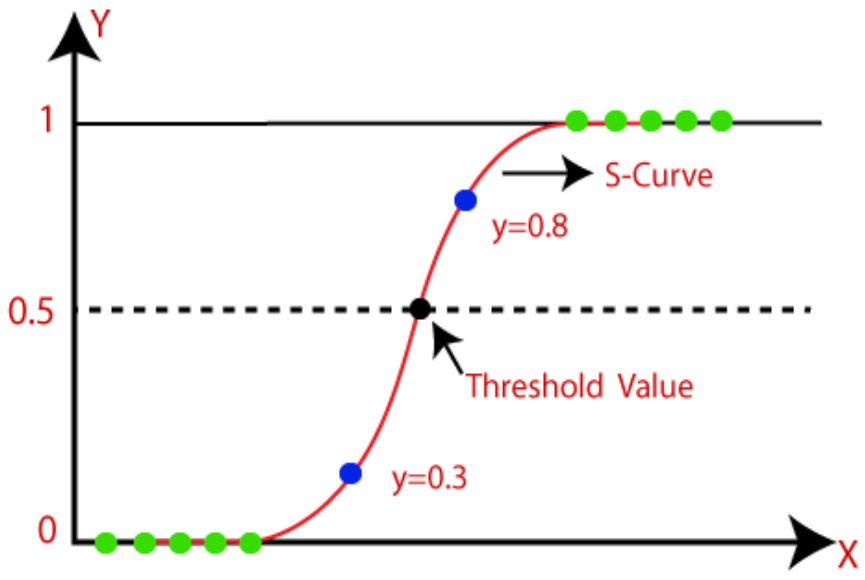
Once the SVC model is trained, it can be used to make predictions on new, unseen data by assigning the data points to one of the predefined classes based on their position relative to the learned hyperplane.

Overall, the SVC algorithm is a powerful tool for classification tasks, providing a flexible and effective approach to separating different classes in the input data.

Logistic Regression:

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

FIG 13: Logistic Regression



Steps in Logistic Regression: To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

RESULTS

Data mining is a process to extract knowledge from existing data. It is used as a tool in banking and finance, in general, to discover useful information from the operational and historical data to enable better decision-making. It is an interdisciplinary field, the confluence of Statistics, Database technology, Information science, Machine learning, and Visualization. It involves steps that include data selection, data integration, data transformation, data mining, pattern evaluation, knowledge presentation. Banks use data mining in various application areas like marketing, fraud detection, risk management, money laundering detection and investment banking.

DATA

```
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('new_water_potability.csv')
```

```
df
```

	s.no	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	0	7.156857	204.890456	20791.31898	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	1	3.716080	129.422921	18630.05786	6.635246	336.094350	592.885359	15.180013	56.329076	4.500656	0
2	2	8.099124	224.236259	19909.54173	9.275884	330.449166	418.606213	16.868637	66.420093	3.055934	0
3	3	8.316766	214.373394	22018.41744	8.059332	356.886136	363.266516	18.436525	100.341674	4.628771	0
4	4	9.092223	181.101509	17978.98634	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0
...	...	...	...	...	...	...	...	...	...	...	...
3271	3271	4.668102	193.681736	47580.99160	7.166639	359.948574	526.424171	13.894419	66.687695	4.435821	1
3272	3272	7.808856	193.553212	17329.80216	8.061362	364.091541	392.449580	19.903225	64.327280	2.798243	1
3273	3273	9.419510	175.762646	33155.57822	7.350233	327.357588	432.044783	11.039070	69.845400	3.298875	1
3274	3274	5.126763	230.603758	11983.86938	6.303357	325.952434	402.883113	11.168946	77.488213	4.708658	1
3275	3275	7.874671	195.102299	17404.17706	7.509306	345.728296	327.459761	16.140368	78.698446	2.309149	1

3276 rows × 11 columns

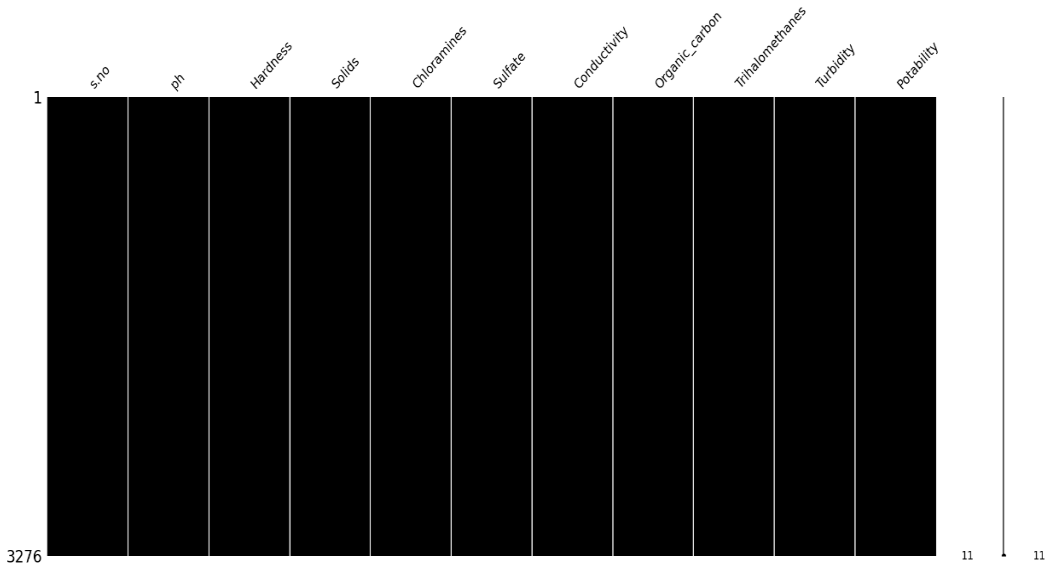
FIG 14: Data collection

Exploratory Data Analysis:

Handling missing values and duplicates

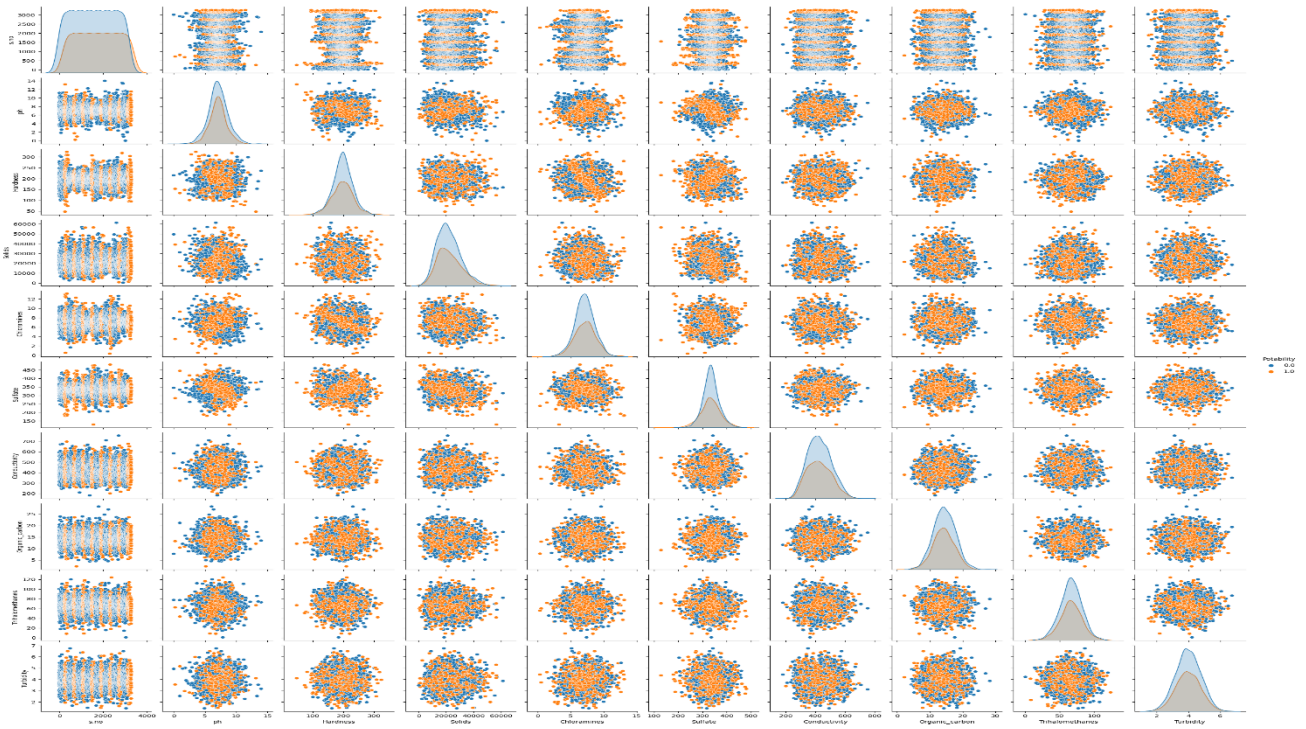
FIG 15: Handling missing and null values

```
In [15]:  1 msno.matrix(new, color=(0, 0, 0))
Out[15]: <Axes: >
```

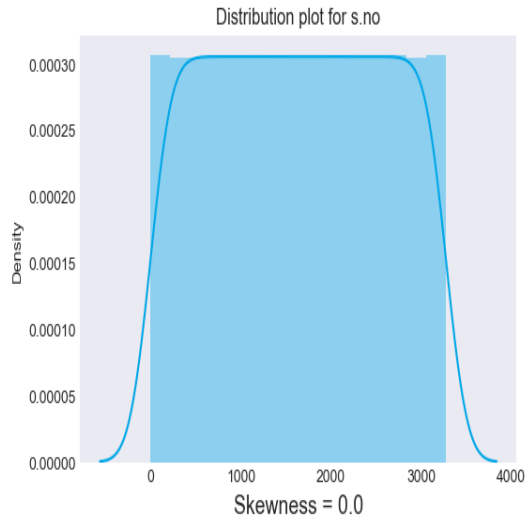


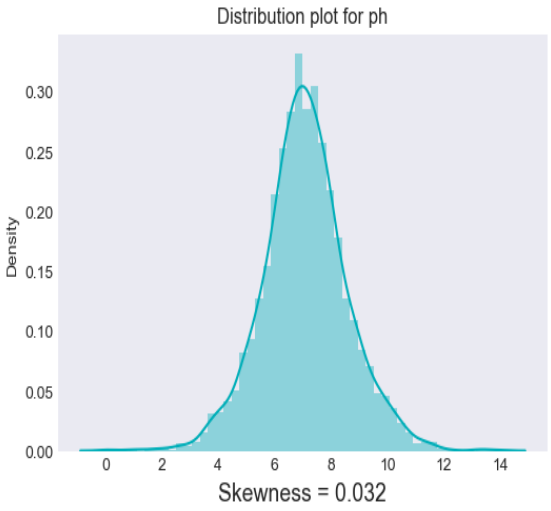
- There Are No Null Values.

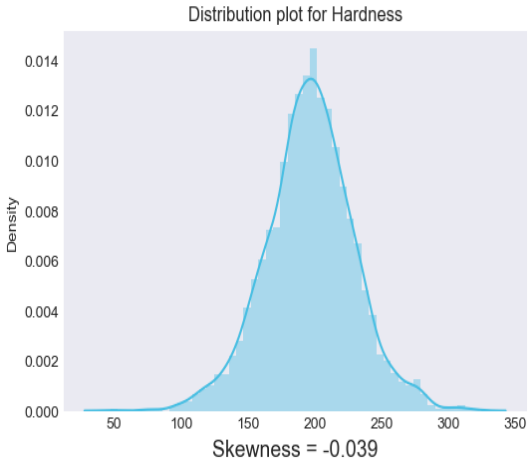
Visualization : FIG16:Pairplot

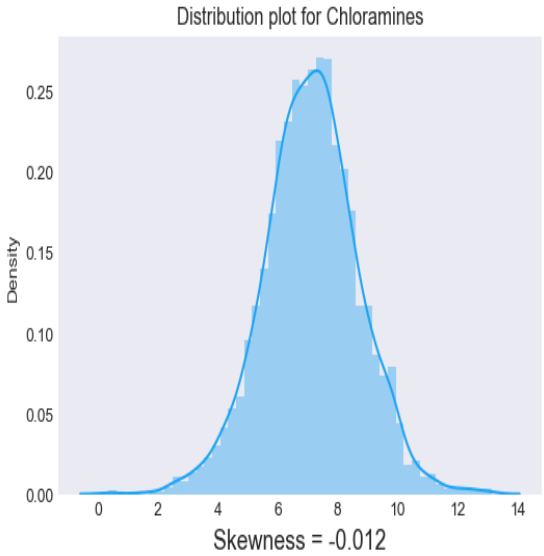


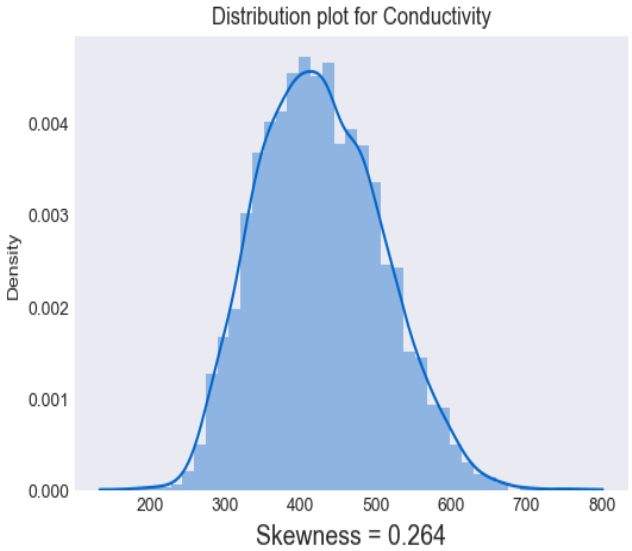
Skewness: FIG17:Skewness

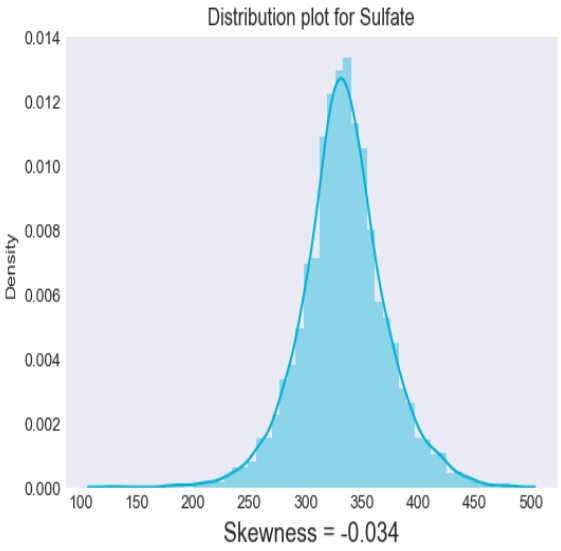


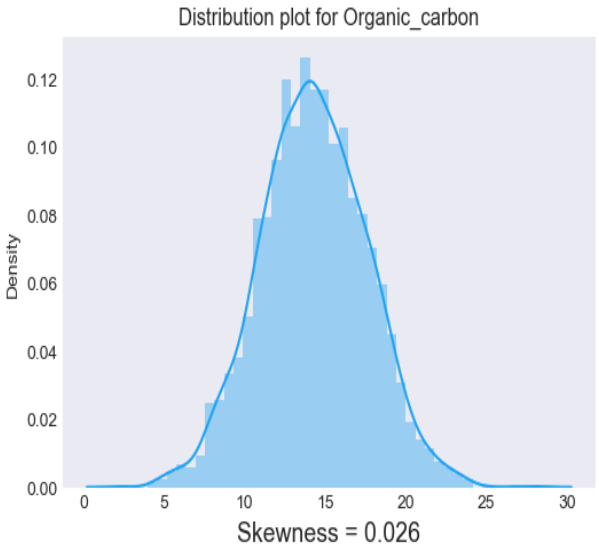


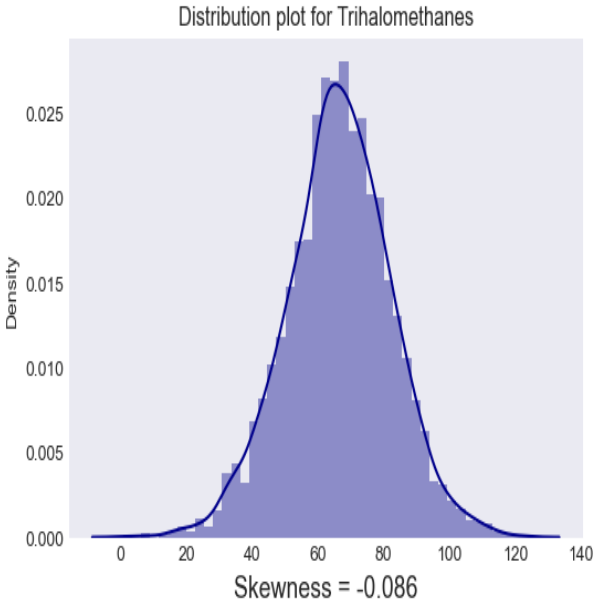


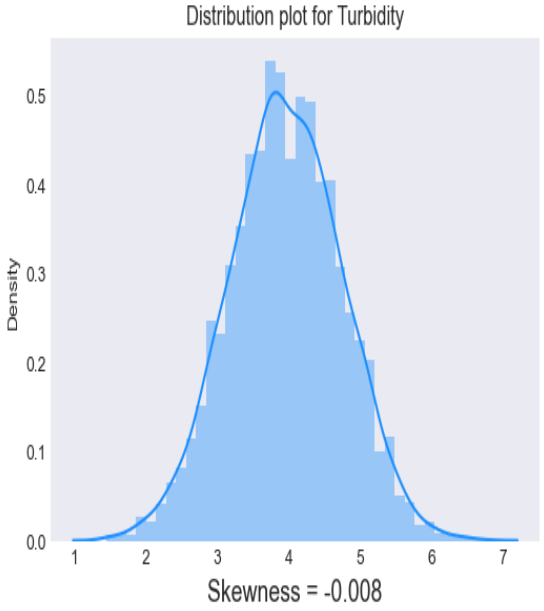


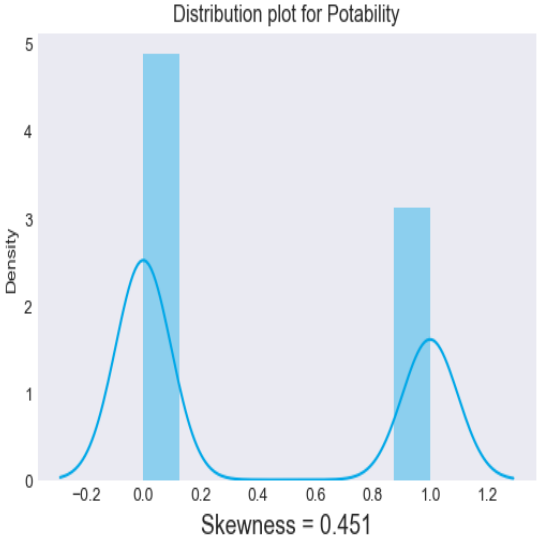



Distribution plot for Sulfate
Density
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
100 150 200 250 300 350 400 450 500
Skewness = -0.034


Distribution plot for Organic_carbon
Density
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0 5 10 15 20 25 30
Skewness = 0.026

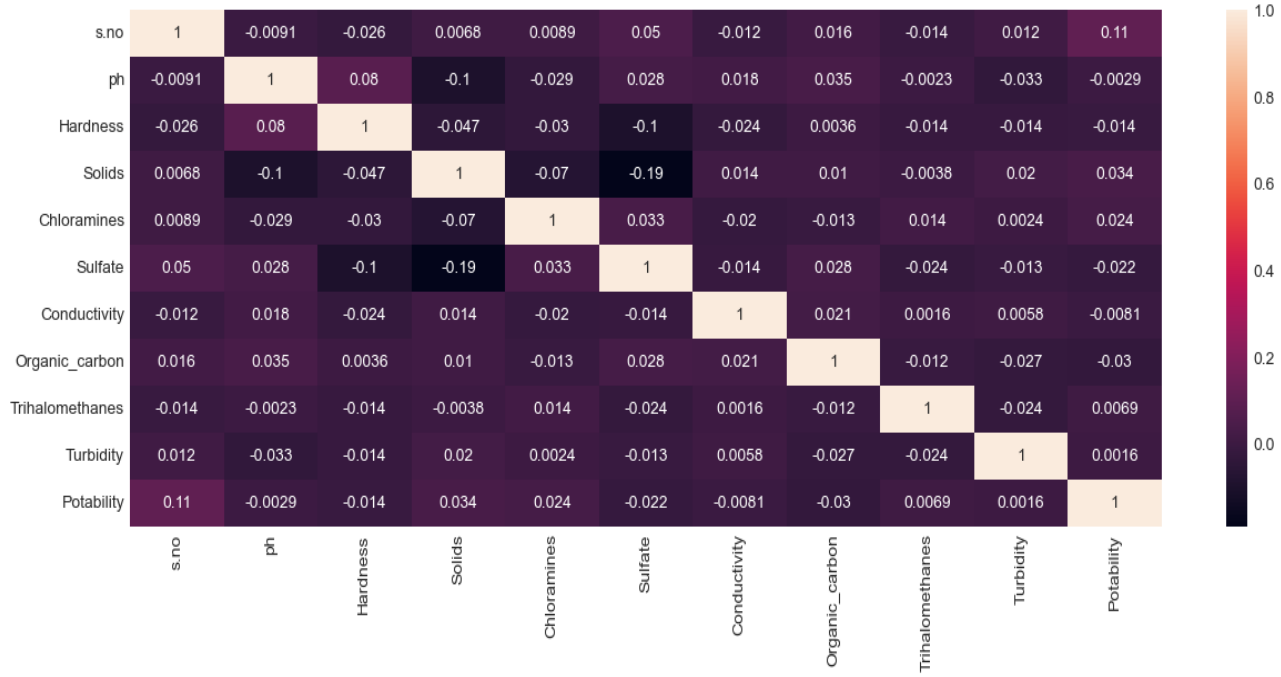

Distribution plot for Trihalomethanes
Density
0.025
0.020
0.015
0.010
0.005
0.000
0 20 40 60 80 100 120 140
Skewness = -0.086


Distribution plot for Turbidity
Density
0.5
0.4
0.3
0.2
0.1
0.0
1 2 3 4 5 6 7
Skewness = -0.008


Distribution plot for Potability
Density
5
4
3
2
1
0
−0.2 0.0 0.2 0.4 0.6 0.8 1.0 1.2
Skewness = 0.451

Correlation:

FIG18:Heatmap



Outliers :

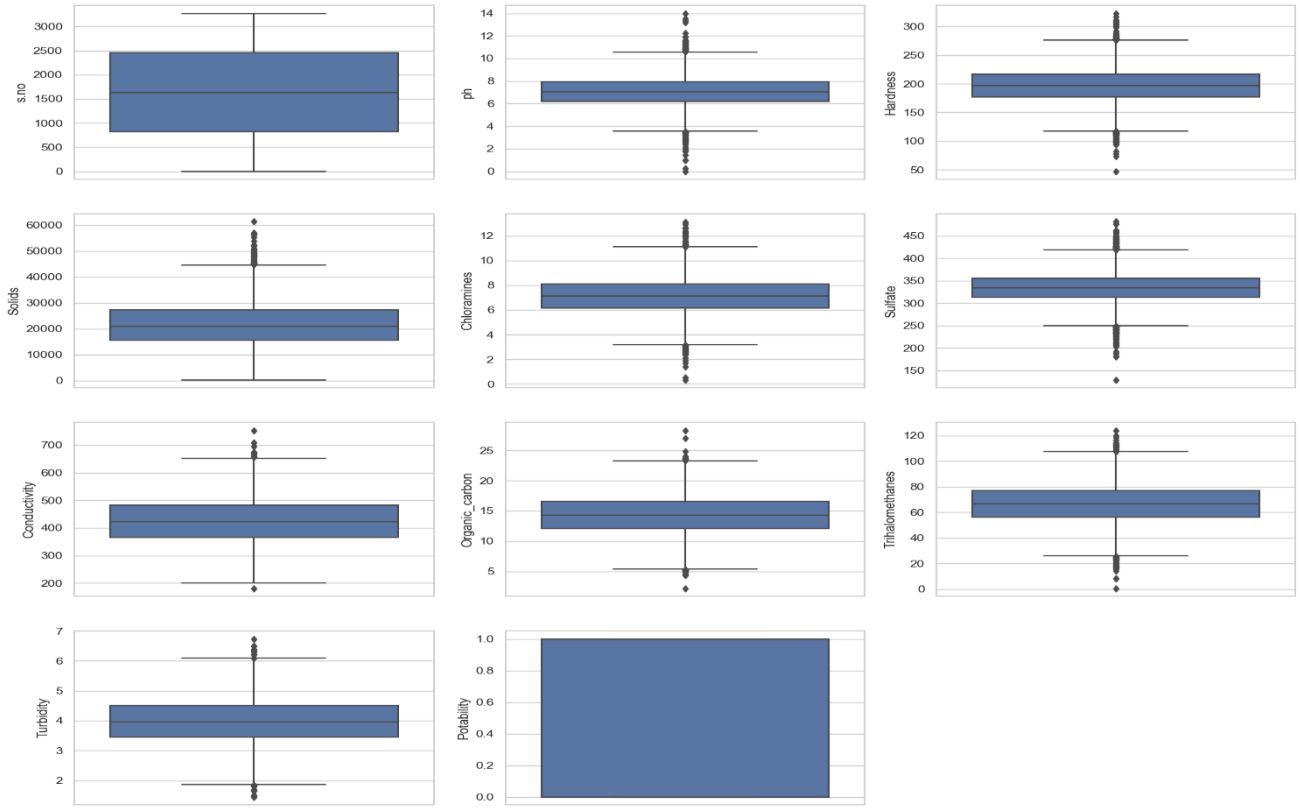
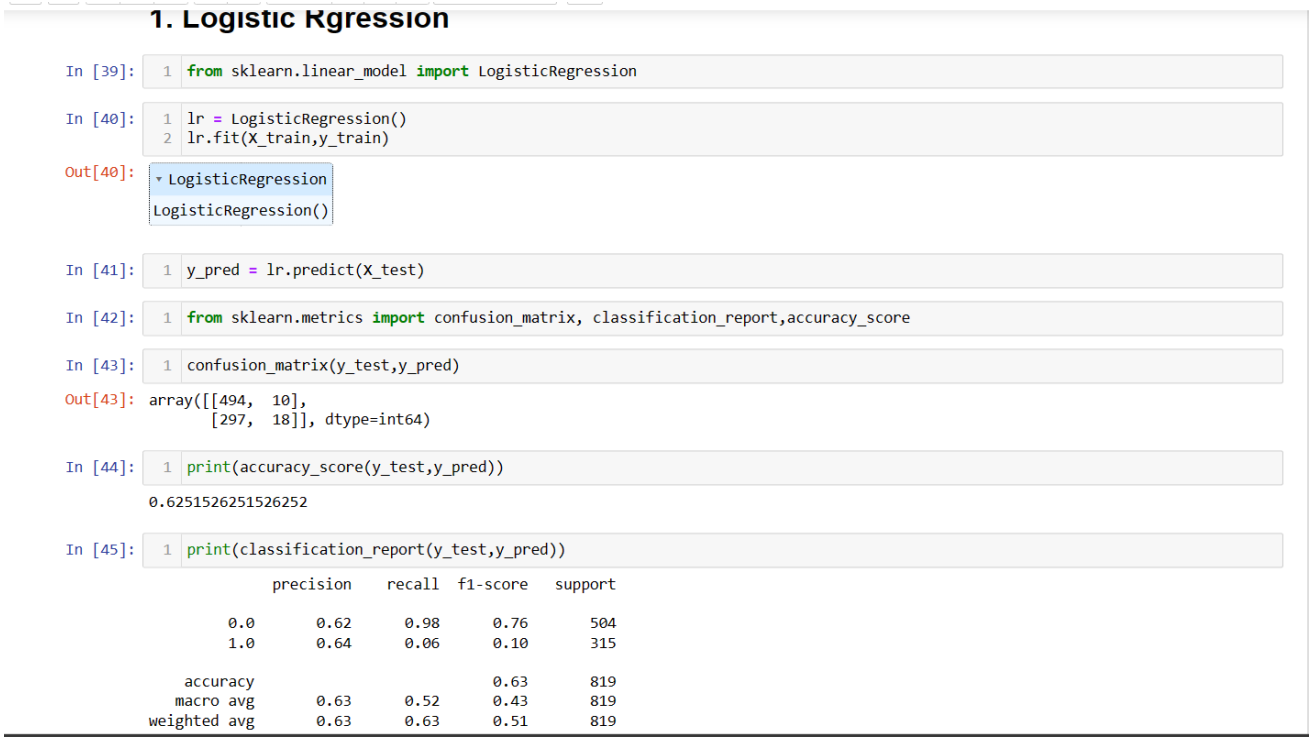


Fig19:Boxplot

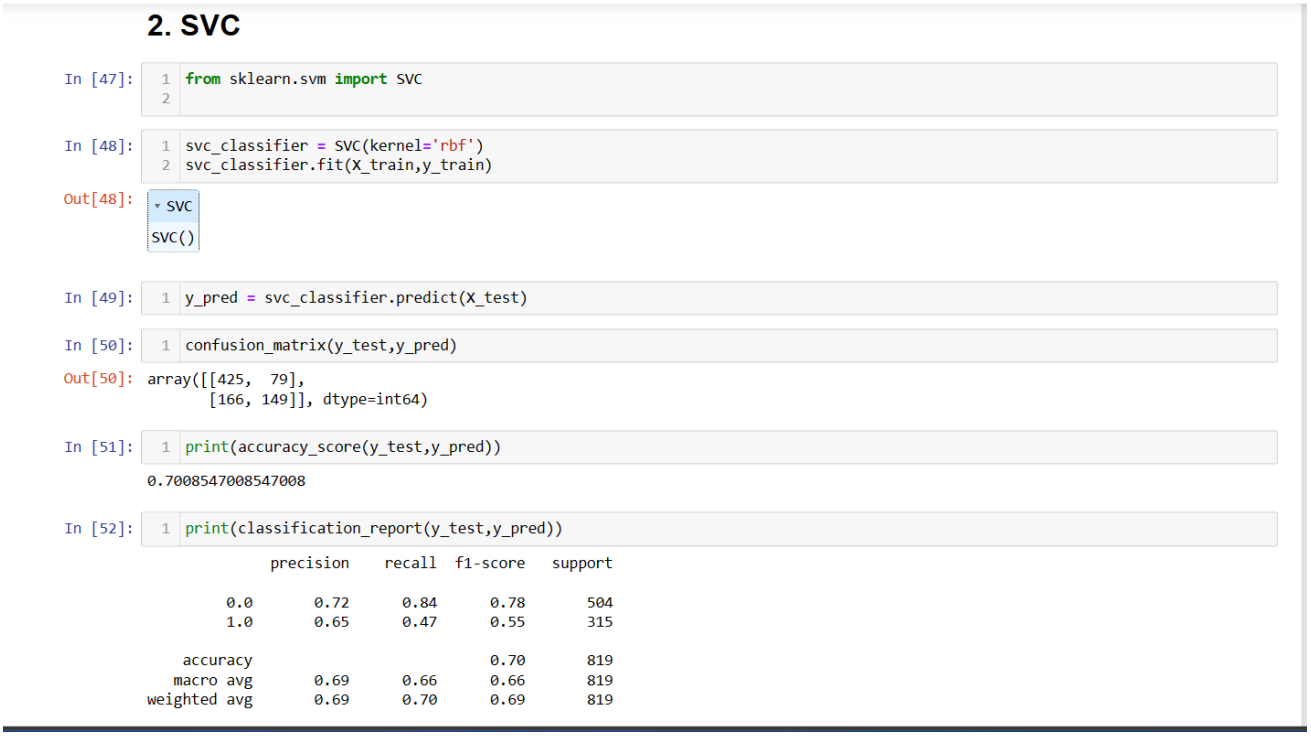
FIG 20: Logistic Regression



1. Logistic Rgression

```
In [39]:
from sklearn.linear_model import LogisticRegression

In [40]:
lr = LogisticRegression()
lr.fit(X_train,y_train)

Out[40]:
▾ LogisticRegression
LogisticRegression()

In [41]:
y_pred = lr.predict(X_test)

In [42]:
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score

In [43]:
confusion_matrix(y_test,y_pred)

Out[43]:
array([[494,  10],
       [297,  18]], dtype=int64)

In [44]:
print(accuracy_score(y_test,y_pred))

0.6251526251526252

In [45]:
print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

         0.0       0.62      0.98      0.76       504
         1.0       0.64      0.06      0.10       315

    accuracy                           0.63       819
   macro avg       0.63      0.52      0.43       819
weighted avg       0.63      0.63      0.51       819
```

FIG 21: Support vector classifier



2. SVC

```
In [47]:
from sklearn.svm import SVC

In [48]:
svc_classifier = SVC(kernel='rbf')
svc_classifier.fit(X_train,y_train)

Out[48]:
▾ SVC
SVC()

In [49]:
y_pred = svc_classifier.predict(X_test)

In [50]:
confusion_matrix(y_test,y_pred)

Out[50]:
array([[425,  79],
       [166, 149]], dtype=int64)

In [51]:
print(accuracy_score(y_test,y_pred))

0.7008547008547008

In [52]:
print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

         0.0       0.72      0.84      0.78       504
         1.0       0.65      0.47      0.55       315

    accuracy                           0.70       819
   macro avg       0.69      0.66      0.66       819
weighted avg       0.69      0.70      0.69       819
```

OUTPUT:

```
input_data = (312,5.783956,161.8265,29299.12,7.028797,350.4309,375.7807,19.76258,86.69846,3.497577)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = svc_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
  print('The water is not potable')
else:
  print('The water is potable')
```

```
[1.]
The water is potable
```

CONCLUSION

This research primarily seeks to use information-mining techniques to predict water quality. Moreover, this domain is now a significant research field whereby numerous strategies have been explored to somehow enhance the efficiency of predicting the water quality and its potability. Throughout this study, by designing a new hybrid model using classification, dealt with the issue of classification accuracy for massive datasets.

These findings can be more improved with the use of several datasets. An increase in the number of datasets can enhance the findings. It is also possible to compare further techniques.

REFERENCES

1. https://www.javatpoint.com

2. https://www.ibm.com

3. https://www.geeksforgeeks.org

4. Jayanthi, R. and Florence, L., 2019. Software defect prediction techniques using metrics based on neural network classifiers. Cluster Computing, 22(1), pp.77-88.

5. Felix, E.A. and Lee, S.P., 2017. Integrated approach to software defect prediction. IEEE Access, 5, pp.21524-21547.

6. Wang, T., Zhang, Z., Jing, X., Zhang, L.: Multiple kernel ensemble learning for software defect prediction. Autom. Softw. Eng. 23, 569–590 (2015).