Environmental Research 202 (2021) 111720

Contents lists available at ScienceDirect

Environmental Research

journal homepage: www.elsevier.com/locate/envres

Automating water quality analysis using ML and auto ML techniques

D. Venkata Vara Prasad [a,c], P. Senthil Kumar [b,c,*], Lokeswari Y. Venkataramana [a,c], G. Prasannamedha [b,c], S. Harshana [a], S. Jahnavi Srividya [a], K. Harrinei [a], Sravya Indraganti [b,c]

[a] Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, 603110, India
[b] Sri Sivasubramaniya Nadar College of Engineering, Department of Chemical Engineering, Chennai, 603110, India
[c] Centre of Excellence in Water Research (CEWAR), Sri Sivasubramaniya Nadar College of Engineering, Chennai, 603110, India

ARTICLE INFO

Keywords:
Machine learning
AutoML
Water quality
SMOTE
TPOT
Water quality index

ABSTRACT

Generation of unprocessed effluents, municipal refuse, factory wastes, junking of compostable and non-compostable effluents has hugely contaminated nature-provided water bodies like rivers, lakes and ponds. Therefore, there is a necessity to look into the water standards before the usage. This is a problem that can greatly benefit from Artificial Intelligence (AI). Traditional methods require human inspection and is time consuming. Automatic Machine Learning (AutoML) facilities supply machine learning with push of a button, or, on a minimum level, ensure to retain algorithm execution, data pipelines, and code, generally, are kept from sight and are anticipated to be the stepping stone for normalising AI. However, it is still a field under research. This work aims to recognize the areas where an AutoML system falls short or outperforms a traditional expert system built by data scientists. Keeping this as the motive, this work dives into the Machine Learning (ML) algorithms for comparing AutoML and an expert architecture built by the authors for Water Quality Assessment to evaluate the Water Quality Index, which gives the general water quality, and the Water Quality Class, a term classified on the basis of the Water Quality Index. The results prove that the accuracy of AutoML and TPOT was 1.4 % higher than conventional ML techniques for binary class water data. For Multi class water data, AutoML was 0.5 % higher and TPOT was 0.6% higher than conventional ML techniques.

1. Introduction

With our natural water bodies being contaminated, the general health of the ecosystem and the human community is in jeopardy. Lakes, rivers and ponds have shaped our personal and industrial routines-from usage as drinking water to incorporation as coolants. A polluted body of water can spur a series of imbalances whose consequences could be as wide as an infectious epidemic. While it can be hard to probe into current practices and evaluate the methodologies of individual sources of pollutants, the quality of a water body to deem the purity of it, laboratory practices, being labour-intensive and time-consuming need an automated technological alternative can be evaluated. With this motivation in hand, water sampling was done from the source, Korattur Lake (in Chennai). Various specifications like pH, turbidity, Total Dissolved Salts, nitrate, phosphate, Chemical Oxygen Demand (COD), iron, sodium and chloride were observed and noted. Basing the architecture on the said parameters, this work analyses and gives a comparison between Machine Learning and Deep Learning computations, to build a constructive model that predicts the quality with maximum accuracy. An identical process is processed in parallel by an Automatic Machine Learning (AutoML) model to observe its results. The results of the models are compared and analysed to obtain a consolidated overview of the Water Quality Index of the water body under study. These results help classify the water body into standard degrees of contamination and provides insights about the pollutants present and future usability.

Automated Machine Learning, in short AutoML, is the automation of executing machine learning procedures to realistic issues. It was introduced to fix the issues using artificial intelligence, to the ever-growing challenge of applying machine learning (Thornton et al., 2012). It encloses the whole pipeline from dataset preparation to deployable machine learning models. It aims at analysis and enhancement of repetitive tasks of machine learning processes like model evaluation, iterative modelling, hyperparameter tuning of models and Algorithm selection.

The principle innovation that the AutoML uses is to look for hyperparameters, used for initial processing of components and choosing the model type, and to optimize their hyperparameters. It employs

* Corresponding author. Sri Sivasubramaniya Nadar College of Engineering, Department of Chemical Engineering, Chennai, 603110, India.
E-mail address: senthilkumarp@ssn.edu.in (P. Senthil Kumar).

https://doi.org/10.1016/j.envres.2021.111720
Received 3 June 2021; Received in revised form 2 July 2021; Accepted 9 July 2021
Available online 20 July 2021

optimization algorithms like Bayesian optimization, a continuous design plan considering global optimization of the hyper-parameters (Mockus, 2012), Neural architecture search (NAS), is an optimization utilized for the automated design of the neural networks (Elsken et al., 2019) and Genetic algorithm frameworks that uses its vital procedures like Mutation, Crossover, Genome coding and Selection to search for the relevant hyperparameters and neural network designs automatically. (Caruana et al., 2004).

AutoML democratises AI and eliminates the need for redundant human labour, increasing efficiency both in terms of time and results.

Due to the serious reduction in computing costs over the past few years, it is now cheaper to have very powerful computers to process calculations, this technology allows us to handle different kinds of datasets: numeric, text and image which considerably expands the range of possibilities and can be used everywhere.

However, AutoML fails to pinpoint the problems for optimising the data used and hence adopting AutoML in areas where the data handled is structured and refined is a much easier task.

One such area would be that of Water Quality Analysis, where the data collected is organised and clean. Since the data is numerical it is the most rudimentary form of Machine Learning and can be streamlined and trusted. The need for feature selection and feature engineering is absent, making human intervention in pre-processing the data unnecessary.

Ahmed et al. (2019) proposed MLP to classify the sample into its WQC (Water Quality Class) with an estimated 85 % accuracy. The work introduced supervised machine learning algorithms for the evaluation of the water quality index (WQI). It was proved that polynomial regression and gradient boosting helped increase the accuracy. All the results observed in the works of Amir Hamzeh Haghiabi et al. (2018) whose work looks into the methodologies of artificial intelligence and their performance including support vector machine (SVM) and artificial neural network (ANN) to anticipate water quality constituents observed in Tireh River situated in southwest Iran. Study of error indexes concluded that the most precise model was, the SVM (Khan and See, 2016). Checks on the correlation between parameters, based on the obtained data from the United States Geological Survey (USGS). The observations were on the basis of Root Mean-Squared Error (RMSE), Mean-Squared Error (MSE) and Regression Analysis and adjusting the related learning rate or restructuring layers. According to the research in (Ahmad et al., 2017), a real-time prediction of WQI was proposed using multiple neural networks that significantly improved the performance as compared to a single neural network. In (Solanki et al., 2015), Solanki et al. used time-series data from USGS National Water Information System (NWIS) and it was modelled with ANN-NAR and maximum accuracy was achieved. According to the research in (Randrianiaina Jerry et al., 2019), deep learning was utilized to do pH, conductivity, dissolved oxygen and turbidity modelling for Itasy Lake. As per the research of Barzegar et al. (2020), water quality was predicted for the Small Prespa Lake situated in Greece and performance of the hybrid CNN-LSTM model surpassed the performance of standalone models namely CNN and LSTM. But it was observed that some of these works had pitfalls such as taking insufficient parameters into account (with respect to water quality), inaccuracy, not efficient in handling the multi-dimensional and unbalanced datasets to name a few. While Li Yang et al. (Yang and Shami, 2020) identifies proper hyperparameters configurations effectively.

With respect to AutoML, Hugo Jair Escalante et al. (Escalante, 2020) delves into the historic progression of Auto ML. The signature findings are summarised and reviewed to gain insights. Xin He et al. (2019) review AutoML methodologies with respect to the pipeline, including data preparation, feature engineering, hyperparameter optimization, and neural architecture search (NAS). This paper provides insights on the process and loopholes encountered through it. Marc- Andre Zoller et al. (Zöller and Huber, 2019) apply selected AutoML frameworks on 137 data sets from established AutoML benchmark suits. They review mathematical formulations covering the pipelines and the open-source options available. Radwa et al. (Shawi et al., 2019) covers potential areas for improvement in AutoML by tackling the CASH (Combined Algorithm Selection and Hyper Parameter Tuning) difficulty. Quanming et al. (Yao et al., 2018) divides and reviews the existing AutoML works in 2 categories-problem set up and corresponding technique employed. This paper gives us insights into the successful applications.

The proposed work banks on two contexts of motivation. The foremost being that Water Quality Analysis holds crucial value as one of the sustaining factors of public health and sustenance. There have been numerous researches conducted in this premise, all evaluating upon selective parameters that promise to determine water quality (Rajamohan et al., 2020; Sujatha et al., 2021). While previous works deal with individualistic algorithms and optimal parameters input, there does not exist a comparisional algorithmic approach towards water quality analysis. Its stark importance stands out when held against the limited data available for this public essentiality. When the available data is meagre and uneven, feature engineering techniques like scaling and Synthetic Minority Oversampling TEchnique (SMOTE) analysis can be done to elevate the quality so as to improve the final results of the model. Hence these are avenues of motivation that this work will explore to add efficiency, which is likely to improve the social weightage of the work.

Secondly, Automated Machine Learning is a relatively new field of research. Launched in 2018 by Google, explored for water Quality Analysis. This is the novel idea which was first used in this research work. While Azure AI and H_2O produce efficient results, the avenue is barely three years old and is still in incubation. While AutoML automates seamlessly the complete pipeline for the user, it's relevance and efficiency to real-life problems like water quality analysis is still under research. To sight niches for improvement and spot the dips in performance is vital for progression of AutoML and this precisely shall be our motivation. This work looks to achieve that by adopting a multi-algorithm comparison module. For every tweaked and perfectly built model made via traditional ML systems, a corresponding AutoML model is created with the same baseline data. Minor test cases that involve usage of advanced libraries of AutoML are also evaluated for the effect they may or may not have upon the system efficiency. Hence, the novelty of AutoML to examine its relevance to our avenue, water quality analysis is explored through this work.

This research aims to identify the water quality of a lake to ensure that water that is available to drink is clean. Conventional methods include performing analysis in labs and could be inaccurate and time-consuming. Hence, this research adopts machine learning/deep learning techniques to solve the problem. In the work proposed by Venkata Vara Prasad D et al. (VenkataVara Prasad et al., 2020), water sampling was done from the source, Korattur Lake (in Tamil Nadu) and was tested using different machine learning/deep learning network models such as Long Short Term Memory (LSTM), Artificial Neural Network (ANN) and Recurrent Neural Networks (RNN). LSTM was observed as the most apt model amongst these three deep learning algorithms, with maximum accuracy of 94 %.

Traditional manual water quality measurement is tedious and time consuming task. So, the scope of this research work is to reduce the time using Automatic Machine Learning (AutoML) and produce more accurate water quality results.

2. Analysis

2.1. Dataset collection

The dataset under study is taken from the Korattur Lake, situated in Chennai, a metropolitan city in South India (https://github.com/Jahnav). Known to be the largest, it spreads over an area of 990 acres, providing potable water to the public for over eighteen years. The dataset consists of observations of over a ten-year period, starting from 2009 until 2019. Under 9 parameters, around 5000 records are existent. The 9 parameters specified are Total Dissolved Solids (TDS), Turbidity,

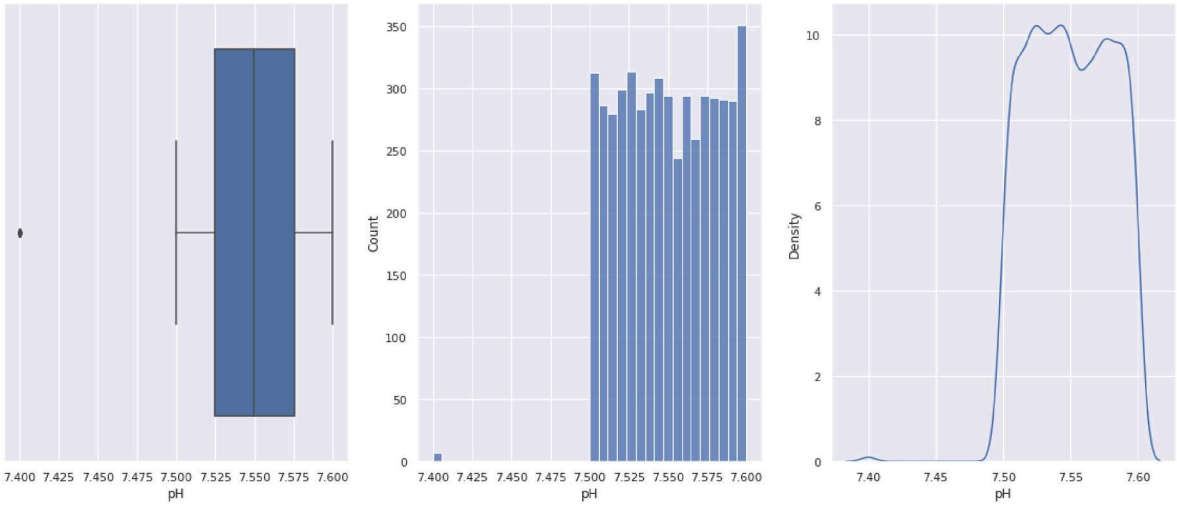

Fig. 1. pH interpretation (a) Box chart to show the outliers of the feature pH (b) Bar chart to show range of the feature pH (c) pH Distribution Plot.

pH, Chemical Oxygen Demand (COD), Iron, Phosphate, Sodium, Chloride and Nitrate.

2.2. Dataset description

The 5000 records majorly are classified into two types, that is, the multi-class and the binary. The parameters considered and the suitable range of parameters required by the drinking water are listed in our previous study (Prasad et al., 2021).

3. Data splitting

As a pre-requisite for priming in deep learning model, it is important to split the data as training set and testing set. After dividing the data into these sets, the deep learning model is trained and later tested on particular parts of the data for the calculation of accuracy in the performance of the model. The data was divided in the ratio of 4:1 for training and testing respectively. Thus, out of all the 5000 samples, 4000 samples were utilized for training and the rest 1000 samples for testing (Prasad et al., 2021).

3.1. Water quality index

The water quality index is evaluated with nine parameters as basis, namely TDS, pH, Phosphate, Turbidity, Iron, Turbidity, Chloride, Sodium and COD which provide an indication for water quality. Each and every parameter, is given weights, on the basis of largest difference between maximum value and minimum value of the respective parameter. The quality rating scale, given by the formula below, is found after each parameter is assigned with weights.

$$Qi = \frac{Ci}{Si} * 100 \tag{1}$$

where, Ci is the concentration of each parameter, Si stands for the desirable or permissible range.

Then, the water quality index is calculated using the following formula

$$WQI = \frac{\Sigma\,(Wi * Qi)}{n} \tag{2}$$

where, Wi stands for the weight assigned to each parameter. On the basis of WQI, the quality of the water is classified (Prasad et al., 2021).

3.2. Exploratory data analysis (EDA) observations

As the first step in the traditional ML pipeline, the data was analysed for the kind of distribution and for further pre-processing, if required.

3.2.1. Binary-class data

Features:

pH.

As observed in Fig. 1a, pH does not have any outliers and from Fig. 1b and c, it can be interpreted that pH follows almost normal distribution. It is a continuous-valued feature.

TDS.

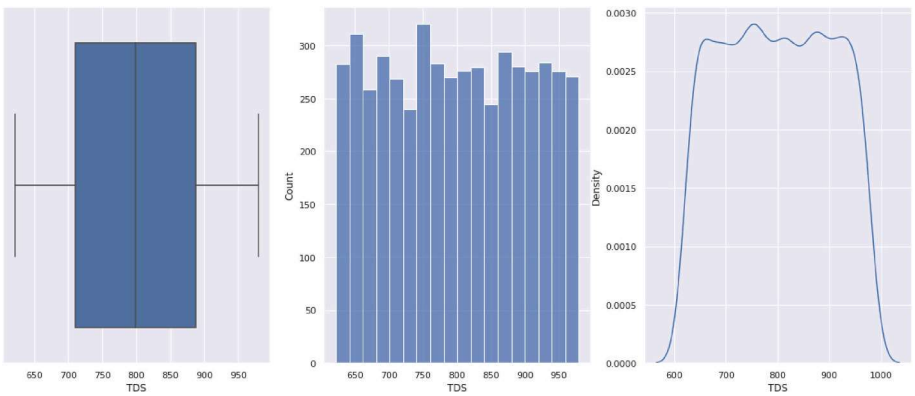

Fig. 2. TDS Interpretation (a) Box chart to show the outliers of the feature TDS (b) Bar chart to show range of the feature TDS (c) TDS Distribution Plot.

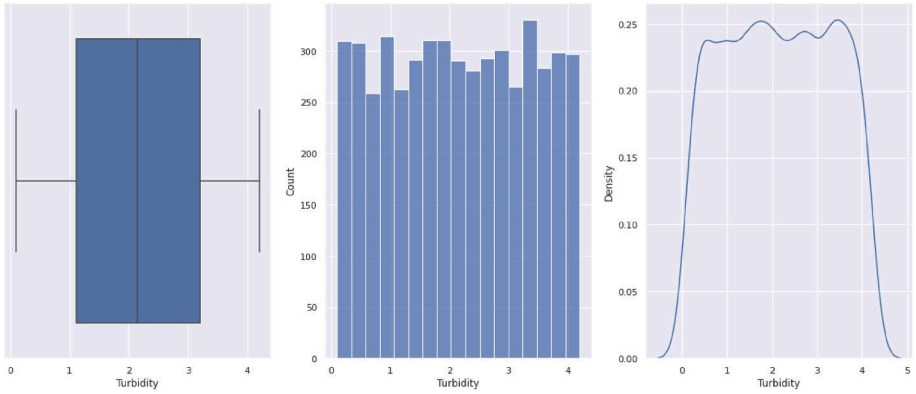

Fig. 3. Turbidity Interpretation (a) Box chart to show the outliers of the feature Turbidity (b) Bar chart to show range of the feature Turbidity (c) Turbidity Distribution Plot.

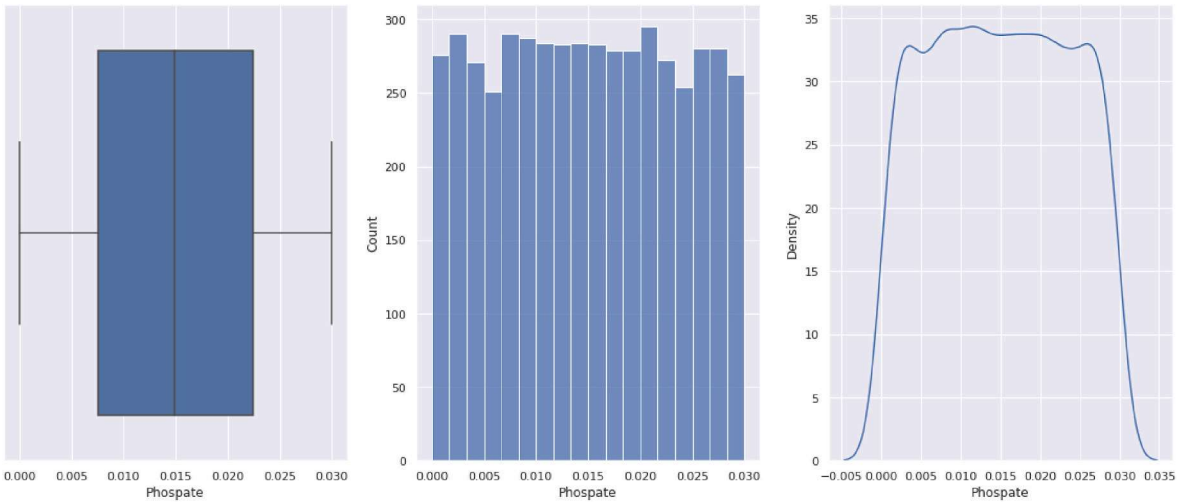

Fig. 4. Phosphate Interpretation (a) Box chart to show the outliers of the feature Phosphate (b) Bar chart to show range of the feature Phosphate (c) Phosphate Distribution Plot.

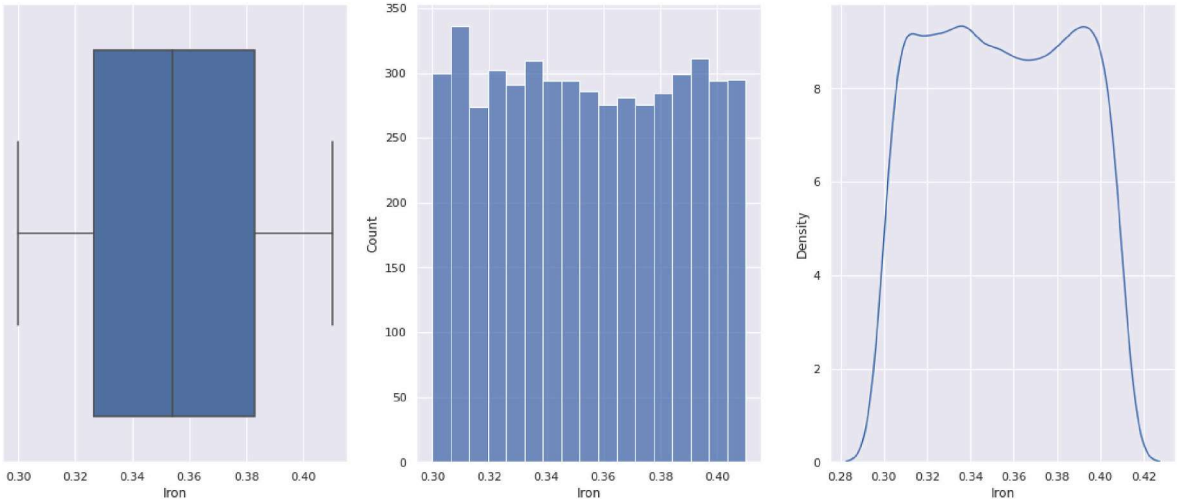

Fig. 5. Iron Interpretation (a) Box chart to show the outliers of the feature Iron (b) Bar chart to show range of the feature Iron (c) Iron Distribution Plot.

As observed in Fig. 2a, TDS does not have any outliers and from Fig. 2b and c, it can be interpreted that TDS follows almost normal distribution. It is also a continuous-valued feature.

Turbidity.

From Fig. 3a, it can be observed that Turbidity does not have any outliers and from Fig. 3b and c, it can be interpreted that it also follows almost normal distribution. It is a continuous-valued feature.

Phosphate.

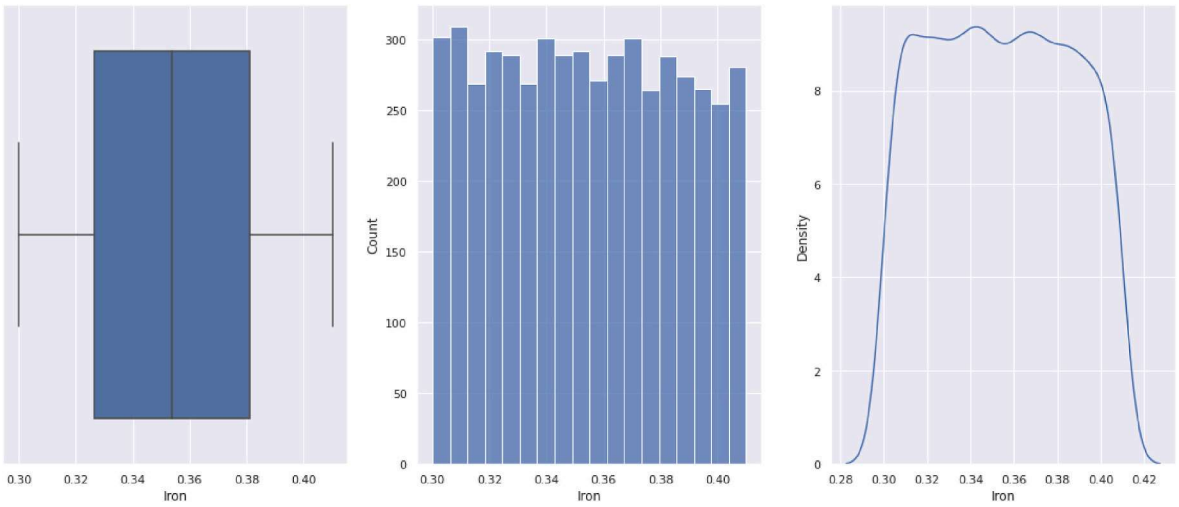

Fig. 6. Density Interpretation (a) Box chart to show the outliers of the feature Density (b) Bar chart to show range of the feature Density(c) Density Distribution Plot.

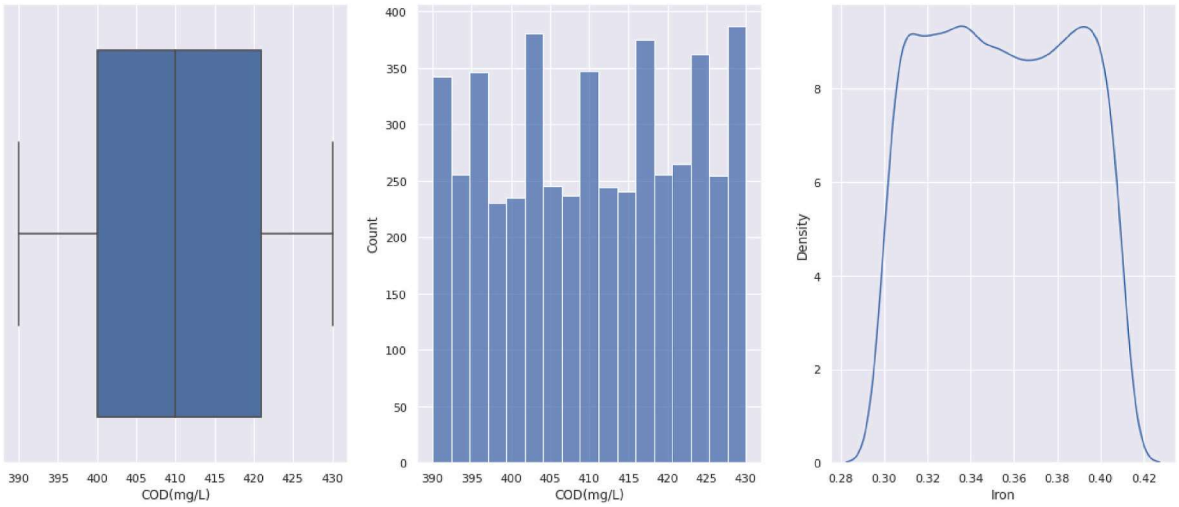

Fig. 7. COD Interpretation (a) Box chart to show the outliers of the feature COD (b) Bar chart to show range of the feature COD (c) COD Distribution Plot.

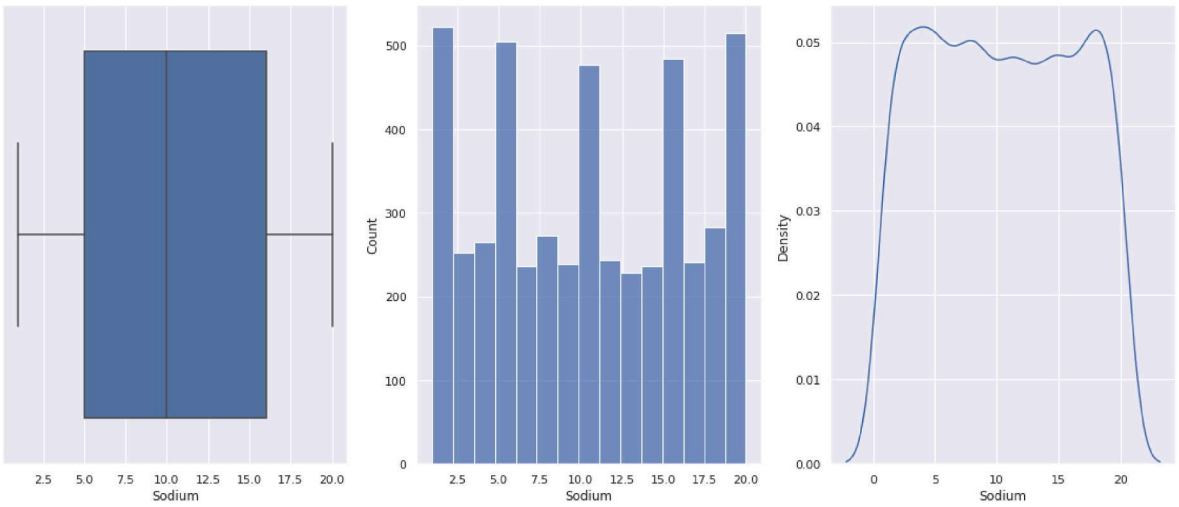

Fig. 8. Sodium Interpretation (a) Box chart to show the outliers of the feature Sodium (b) Bar chart to show range of the feature Sodium (c) Sodium Distribution Plot.

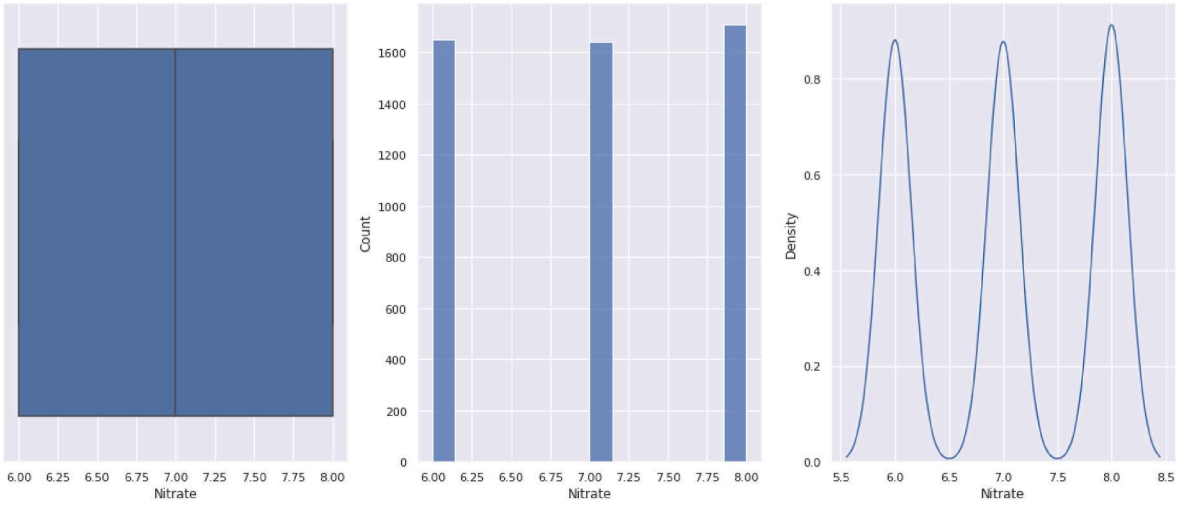

Fig. 9. Nitrate Interpretation (a) Box chart to show the outliers of the feature Nitrate (b) Bar chart to show range of the feature Nitrate (c) Nitrate Distribution Plot.

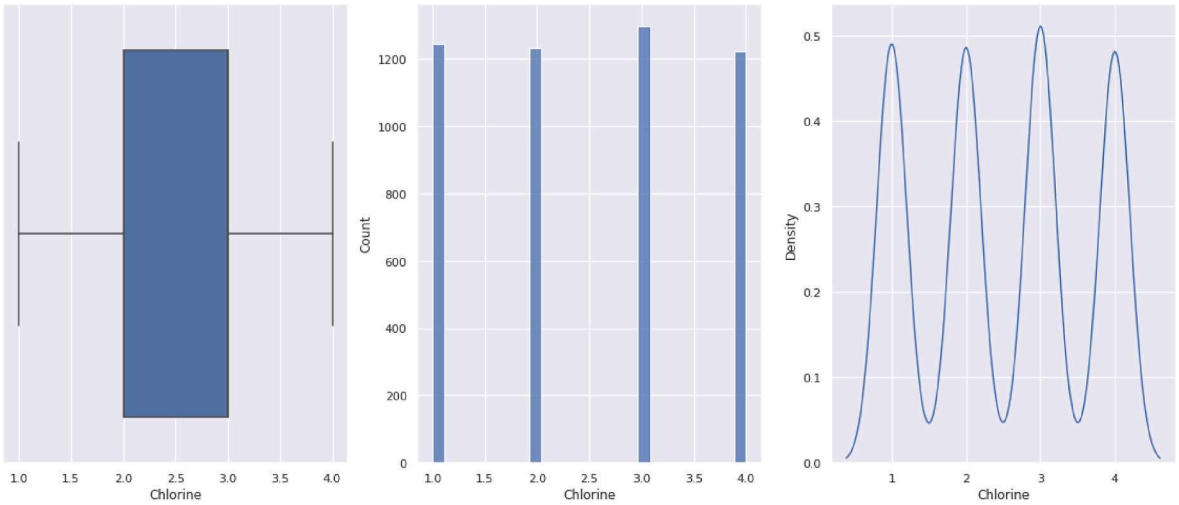

Fig. 10. Chlorine Interpretation (a) Box chart to show the outliers of the feature Chlorine (b) Bar chart to show range of the feature Chlorine (c) Chlorine Distribution Plot.

From Fig. 4a, it can be observed that Phosphate does not have any outliers and from Fig. 4b and c, it can be interpreted that it also follows almost normal distribution. It is a continuous-valued feature.

Iron.

From Fig. 5a, it can be observed that Iron does not have any outliers and from Fig. 5b and c, it can be interpreted that it also follows almost normal distribution. It is also a continuous-valued feature.

Density.

As observed in Fig. 6a, Density does not have any outliers and from Fig. 6b and c, it can be interpreted that TDS follows almost normal distribution. It is also a continuous-valued feature.

COD.

As observed in Fig. 7a, COD does not have any outliers and from Fig. 7b and c, it can be interpreted that it follows almost normal distribution. It is a continuous-valued feature.

Sodium.

From Fig. 8a, it can be observed that Sodium does not have any outliers and from Fig. 8b and c, it can be interpreted that it also follows almost normal distribution. It is also a continuous-valued feature.

Nitrate.

As observed in Fig. 9a, Nitrate does not have any outliers, from Fig. 9b and c, it can be interpreted Nitrate takes only 3 values.

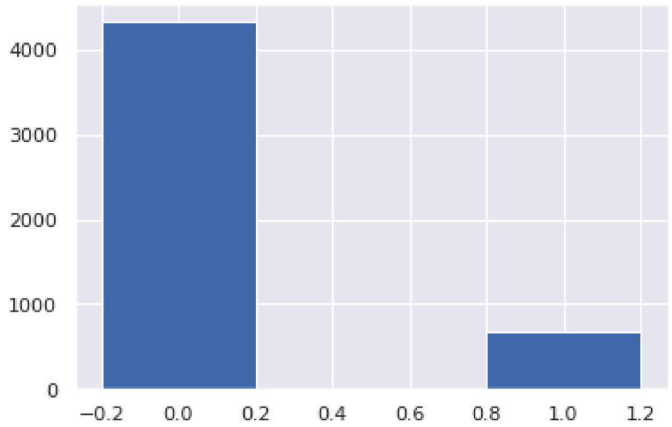

Fig. 11. Class distribution.

Chlorine.

From Fig. 10a, it can be observed that Chlorine does not have any outliers, from Fig. 10b and c, it can be interpreted that Chlorine takes only 4 values.

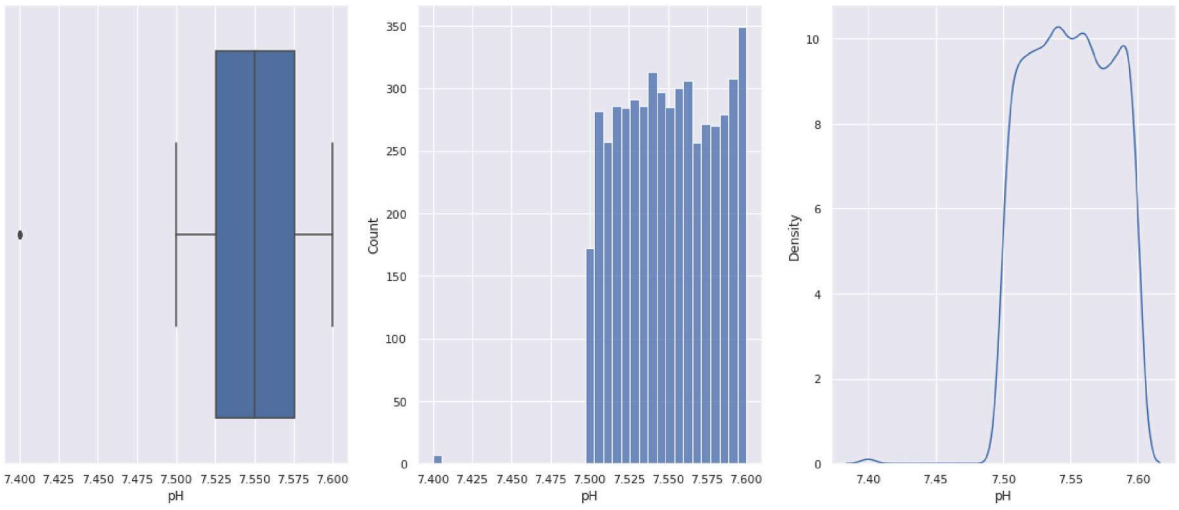

Fig. 12. pH interpretation (a) Box chart to show the outliers of the feature pH (b) Bar chart to show range of the feature pH (c) pH Distribution Plot.

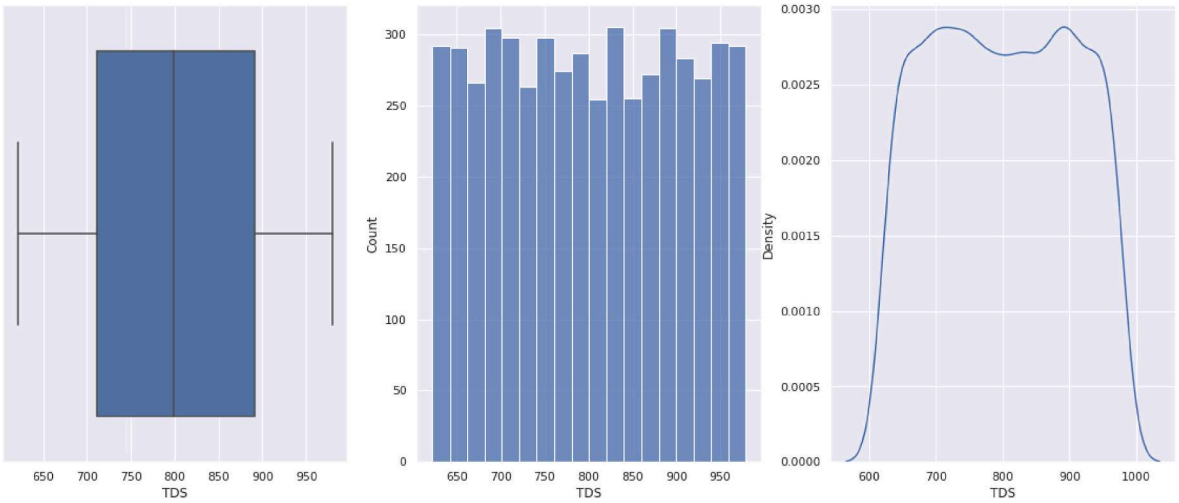

Fig. 13. TDS Interpretation (a) Box chart to show the outliers of the feature TDS (b) Bar chart to show range of the feature TDS (c) TDS Distribution Plot.

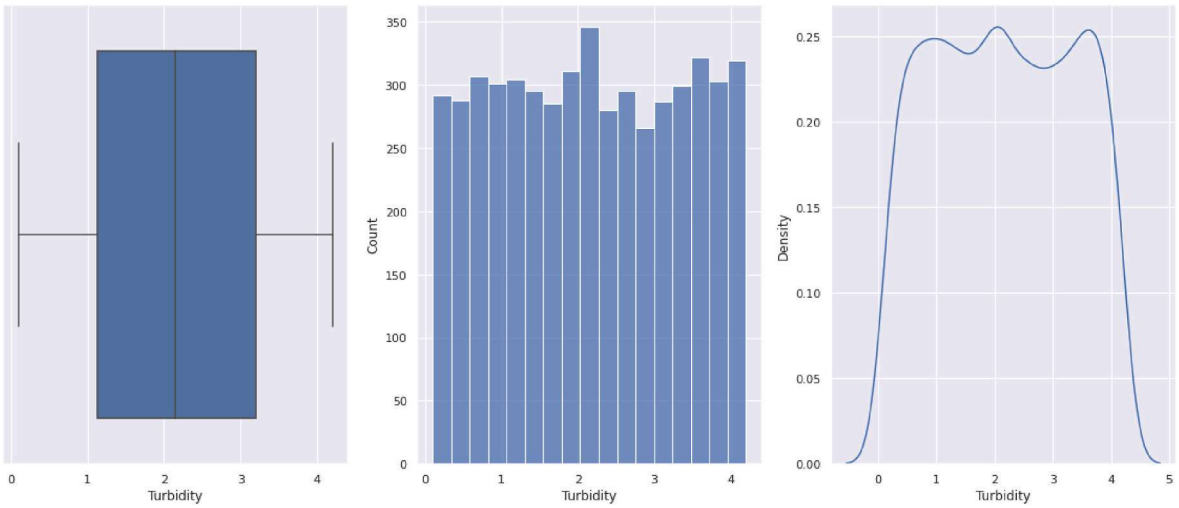

Fig. 14. Turbidity Interpretation (a) Box chart to show the outliers of the feature Turbidity (b) Bar chart to show range of the feature Turbidity (c) Turbidity Distribution Plot.

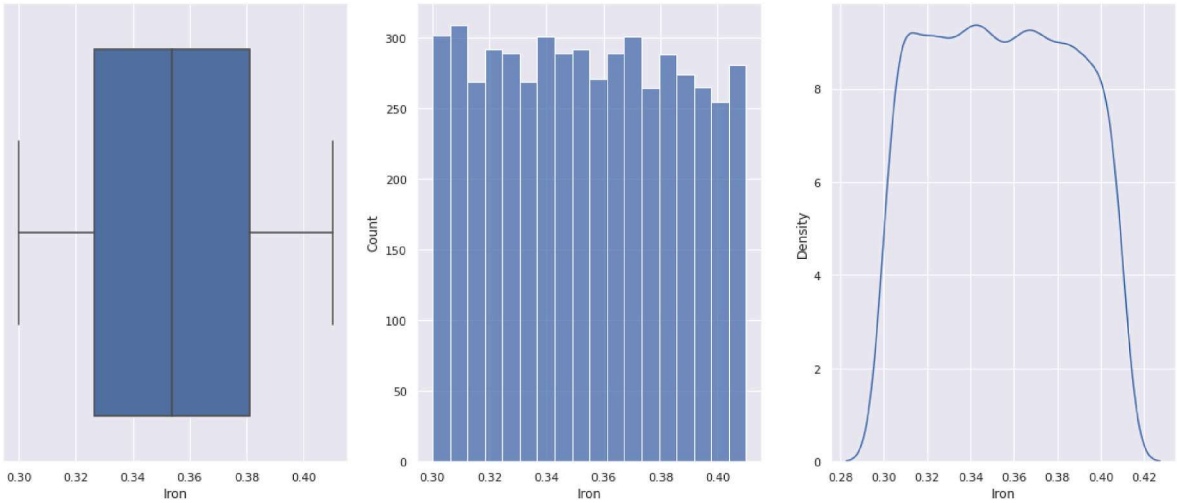

Fig. 15. Phosphate Interpretation (a) Box chart to show the outliers of the feature Phosphate (b) Bar chart to show range of the feature Phosphate (c) Phosphate Distribution Plot.

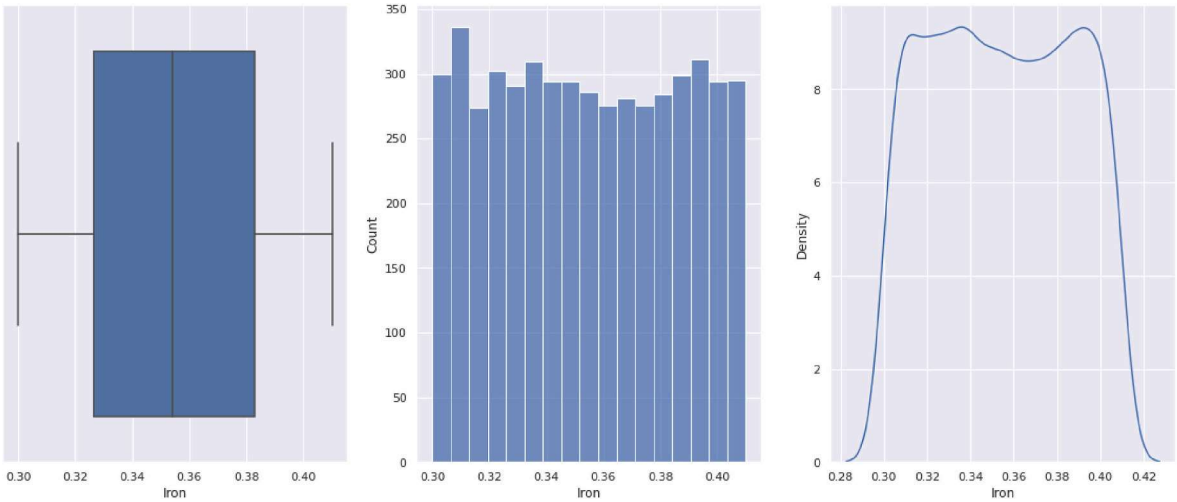

Fig. 16. Iron Interpretation (a) Box chart to show the outliers of the feature Iron (b) Bar chart to show range of the feature Iron (c) Iron Distribution Plot.

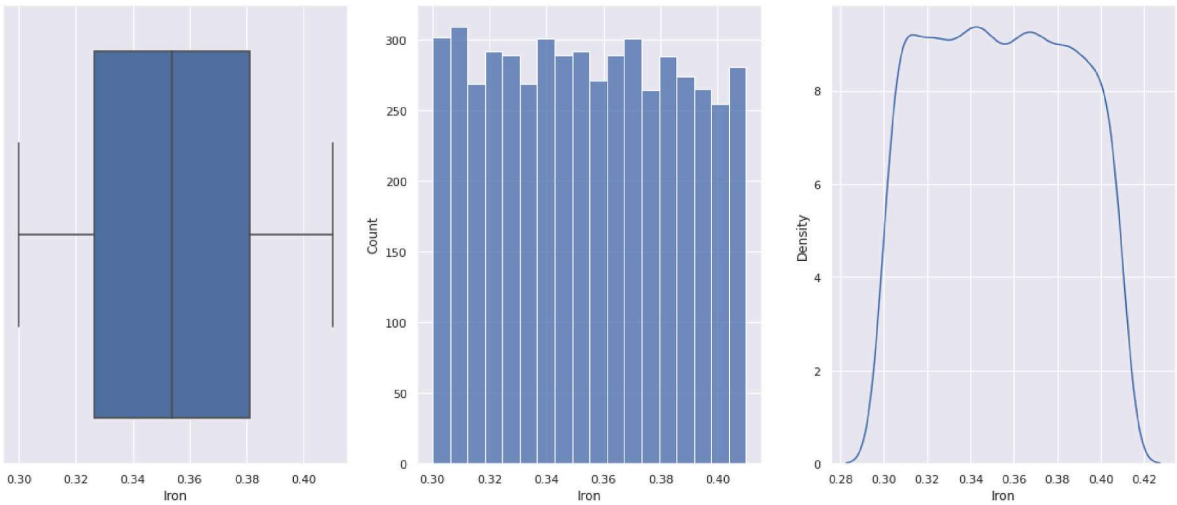

Fig. 17. Density Interpretation (a) Box chart to show the outliers of the feature Density (b) Bar chart to show range of the feature Density(c) Density Distribution Plot.

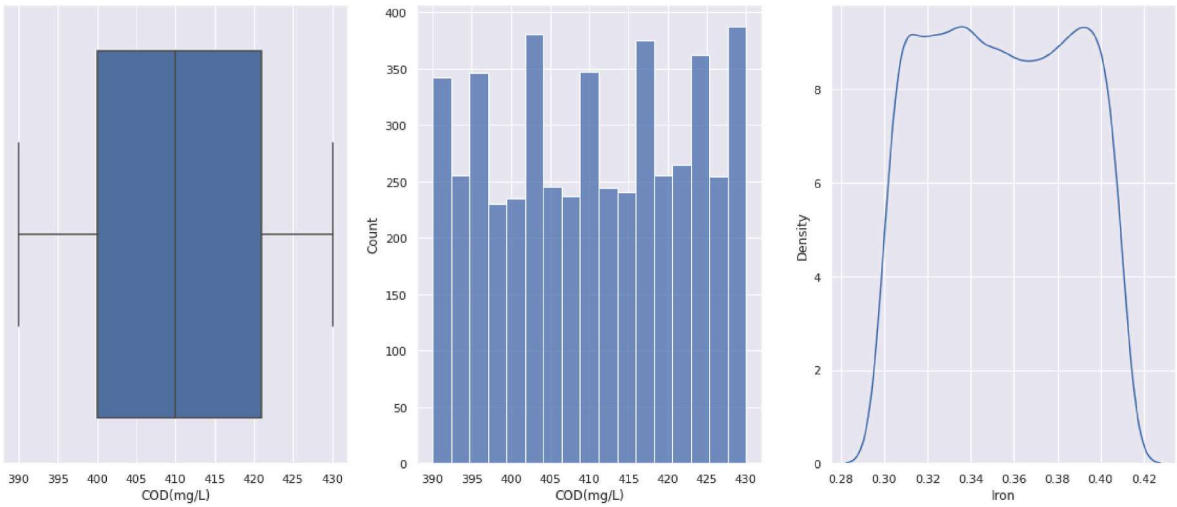

Fig. 18. COD Interpretation (a) Box chart to show the outliers of the feature COD (b) Bar chart to show range of the feature COD (c) COD Distribution Plot.

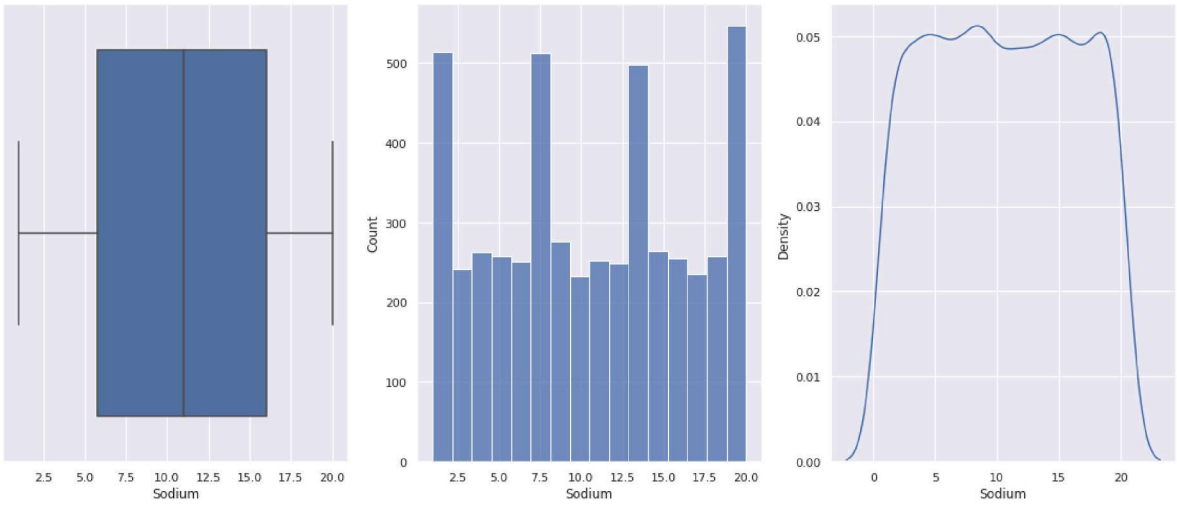

Fig. 19. Sodium Interpretation (a) Box chart to show the outliers of the feature Sodium **(b)** Bar chart to show range of the feature Sodium **(c)** Sodium Distribution Plot.

Class Label. As observed in Fig. 11, the distribution of records in the 2 classes is

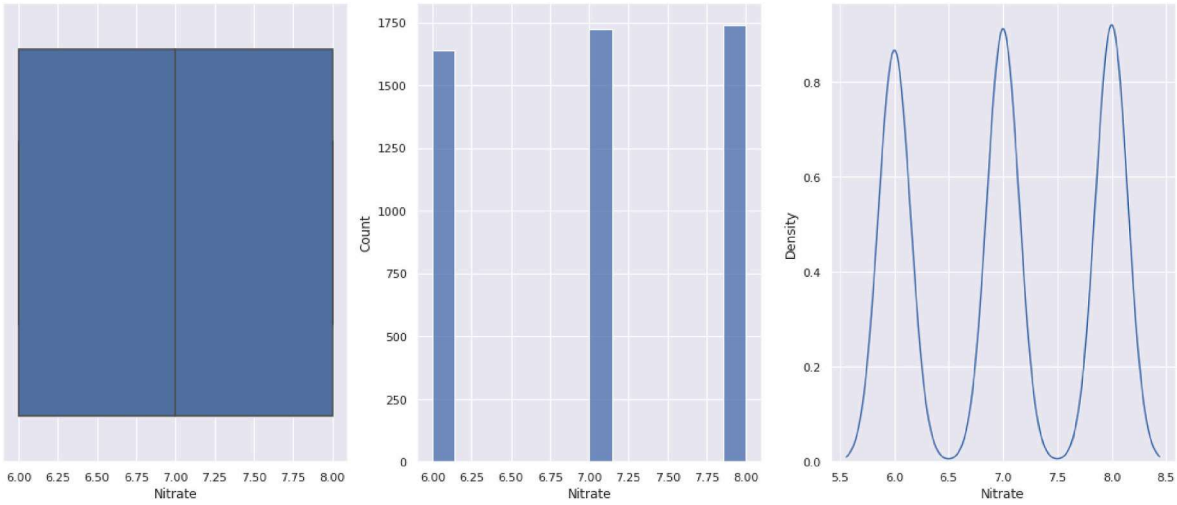

Fig. 20. Nitrate Interpretation (a) Box chart to show the outliers of the feature Nitrate (b) Bar chart to show range of the feature Nitrate (c) Nitrate Distribution Plot.

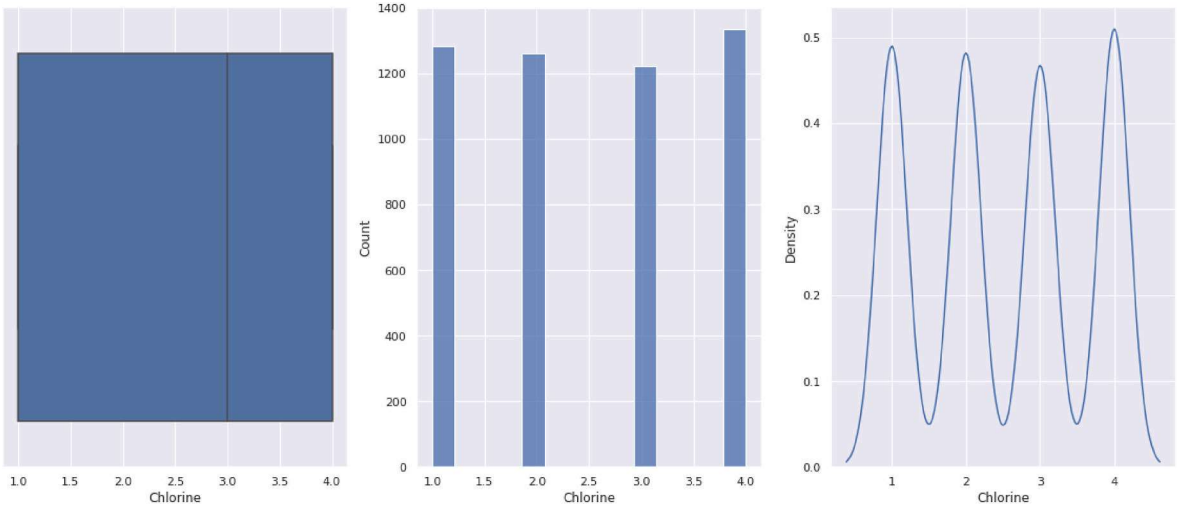

Fig. 21. Chlorine Interpretation (a) Box chart to show the outliers of the feature Chlorine (b) Bar chart to show range of the feature Chlorine (c) Chlorine Distribution Plot.

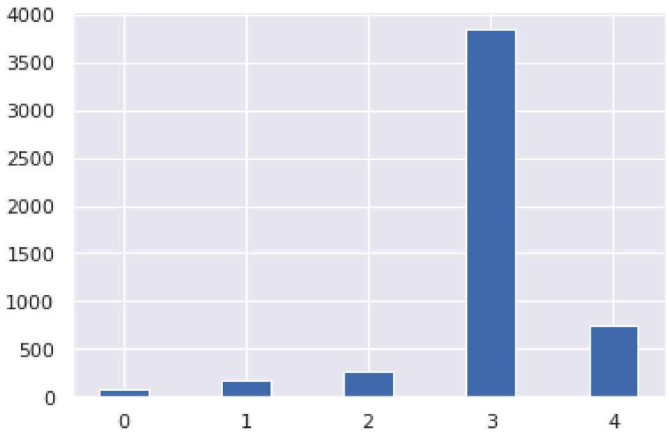

Fig. 22. Class distribution.

uneven and it is majorly concentrated in Class 1.

3.2.2. *Multi-class data*

Features-

pH.

As observed in Fig. 12a, pH does not have any outliers and from Fig. 12b and c, it can be interpreted that pH follows almost normal distribution. It is a continuous-valued feature.

TDS.

As observed in Fig. 13a, TDS does not have any outliers and from Fig. 13b and c, it can be interpreted that TDS follows almost normal distribution. It is also a continuous-valued feature.

Turbidity.

From Fig. 14a, it can be observed that Turbidity does not have any outliers and from Fig. 14b and c, it can be interpreted that it also follows almost normal distribution. It is a continuous-valued feature.

Phosphate.

From Fig. 15a, it can be observed that Phosphate does not have any outliers and from Fig. 15b and c, it can be interpreted that it also follows almost normal distribution. It is a continuous-valued feature.

Iron.

From Fig. 16a, it can be observed that Iron does not have any outliers and from Fig. 16b and c, it can be interpreted that it also follows almost normal distribution. It is also a continuous-valued feature.

Density.

As observed in Fig. 17a, Density does not have any outliers and from Fig. 17b and c, it can be interpreted that TDS follows almost normal distribution. It is also a continuous-valued feature.

COD.

As observed in Fig. 18a, COD does not have any outliers and from Fig. 18b and c, it can be interpreted that it follows almost normal distribution. It is a continuous-valued feature.

Sodium.

From Fig. 19a, it can be observed that Sodium does not have any outliers and from Fig. 19b and c, it can be interpreted that it also follows almost normal distribution. It is also a continuous-valued feature.

Nitrate.

As observed in Fig. 20a, Nitrate does not have any outliers and from Fig. 20b and c, it can be interpreted that there are only 3 values that the feature can take.

Chlorine.

From Fig. 21a, it can be observed that Chlorine does not have any outliers and from Fig. 21b and c, it can be interpreted that there are only 4 values this feature can take.

Class Label.

As observed in Fig. 22, the distribution of records in the 2 classes is uneven and it is majorly concentrated in Class 3.

3.3. *Data preprocessing*

3.3.1. *Data cleaning*

The data did not have any missing values, hence missing values were not imputed. The data had almost no or very few outliers, therefore it was considered as noise-free. It did not require any outlier removal techniques.

3.3.2. *Scaling data*

The data was scaled using Standard scaling for all the features. Standard scaling is a scaling technique where the values are centred around the mean having unit standard deviation. Provide a reference paper or book for Standard Scaling and put it in the reference list with the number cited here. The formula for standard scaling shown in equation (3):

$$z = \frac{(x - \mu)}{\sigma} \tag{3}$$

where $\mu = mean$ and. $\sigma = standard\ deviation$

3.3.3. Balancing classes

The distribution of records in the different classes is uneven and concentrated in Class 0 (Binary Data) and Class 3 (Multi-class Data). It was balanced by using an oversampling technique, namely Synthetic Minority Oversampling Technique (SMOTE).

3.3.3.1. SMOTE. SMOTE creates samples of synthetic minority classes. It is an oversampling technique where it is imperative to resample the data as imbalanced data will degrade the performance of the classifier. It happens because predictions will occur for majority data and minority data will be treated as noise and will be ignored.

For each example, *xk* ε *A* (k = 1, 2, 3 … N), the given formula is used to formulate a new example:

$$x' = x + rand(0,1) * |x - x \tag{4}$$

where and (0, 1) represents a random number between 0 and 1.

Here, N represents the sampling rate and A1 is the set of N examples randomly selected from its k nearest neighbors.

3.4. Methods

The following algorithms were explored that would be suitable for our data and the classification task at hand.

3.4.1. Naive Bayes

Naive Bayes classifiers are on the basis of application of Bayes' theorem with strong independent assumptions in between the features. They are probabilistic classifiers. The probability is calculated with Bayes theorem as given in equation (5) and the highest value is considered.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \tag{5}$$

where X and y are classes.

That feature makes an independent and equal contribution to the outcome is the basic assumption. The complexity of the algorithm in training and testing are as given below:

Training Time Complexity = O (n*d)

Run-time Complexity = O (c*d)

3.4.2. Logistic regression

In the case of logistic regression, the response variable gives the probability of the out-come being positive case. For suppose, the response variable is seen as to equal or higher than certain discrimination threshold, it is concluded to be a positive class; if not, it is concluded as negative class. The complexity of the algorithm in training and testing are as given below:

Training Complexity: n (O(d)) = O (nd)

Testing Complexity: O(d)

n = number of training examples

d = number of dimensions of the data.

3.4.3. Support vector machine

A Support Vector Machine (SVM) is a supervised machine learning model which creates boundaries so any space can be segregated into a set of classes. A new input data point can be classified correctly. It separates data in the best optimal way with a separating hyperplane. The complexity of the algorithm in training and testing are as given below:

Training Time Complexity = O (n2)

Run-time Complexity = O (k*d)

K = number of Support Vectors

d = dimensionality of the data.

3.5. Decision trees

Decision Trees (DTs) are a supervised learning method that are non-parametric. These are used for regression and classification. The prime attribute is considered as the root and the other attributes are compared one by one. It involves division of the given data into subsets. The complexity of the algorithm in training and testing are as given below:

Training Time Complexity = O (n*log(n)*d)

Run-time Complexity = O (maximum depth of the tree)

n = number of points in the Training set

d = dimensionality of the data.

3.6. Random Forest

Random forest consists of individual decision trees that operate as a unit. It gives the weighted total, which is in turn, given as the input for an activation function, finally gives the output. Each individual tree outputs a class prediction. The results are aggregated as seen in Fig. 5. Moreover, the accuracy is proportional to the number of trees.

The complexity of the algorithm in training and testing are as given below:

Training Time Complexity = O (n*log(n)*d*k)

Run-time Complexity = O (depth of tree* k)

Space Complexity = O (depth of tree *k)

k = number of Decision Trees.

3.7. AutoML libraries used

3.7.1. AutoML mljar-supervised

It selected the best out of the following algorithms: Baseline, Linear, Random Forest, Extra Trees, Nearest Neighbors, CatBoost, LightGBM, Xgboost, and Neural Networks. This AutoML library can perform feature pre-processing such as imputation of the values missing and transforming categoricals. It regulates the hyperparameters using not-so-random-search algorithm (random-search over specified values set) and hill climbing to tune up the final models. It also has the capacity to compute Ensemble on the basis of greedy algorithm 18].

3.7.2. TPOT

The Tree-Based Pipeline Optimization Tool (TPOT) is open-source software packages developed for the data science community that aims to automate ML pipelines. It combines stochastic optimization algorithms for search works, like genetic programming, with expression tree representation of different pipelines. Scikit-learn library, that has Python as its base, is utilized by TPOT for its ML menu.

3.8. Metrics used

Accuracy is one of the metrics that calculates model's efficiency and performance. It is defined as the ratio of number of right predictions divided by the total number of predictions. The expression for accuracy is as follows:

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} \tag{6}$$

However, Accuracy is used when the number of positive and negative classes are almost equal. Since it can be a misleading technique of evaluation where a model gets a high overall accuracy due to higher accuracy for the dominant class in the dataset, it doesn't mean the model is good if it fails on all negative instances. A problem arises when the cost of a misclassification is very high.

Hence, precision is a helpful evaluation of the success of prediction for classes that are unbalanced. Precision is defined as the ratio of the rate of the number of samples rightly anticipated as drinkable (Class 1) divided by all the samples segregated as drinkable (Class 1) given by model. The expression for precision is as follows:

Table 1
Results of Binary Classification via traditional ML Pipeline. (a) Without SMOTE (b) With SMOTE.

Algorithm/Type of ML	Without SMOTE			With SMOTE		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy
Naive Bayes	0.945	0.946	0.946	0.945	0.918	0.918
Logistic Regression	0.918	0.92	0.92	0.92	0.88	0.88
SVM	0.918	0.921	0.921	0.918	0.876	0.876
Decision Trees	**0.999**	**0.999**	**0.999**	0.929	0.927	0.927
Random Forest	0.998	0.998	0.998	0.929	0.927	0.927

Table 2
Results of Multi-class Classification via traditional ML Pipeline. (a) Without SMOTE (b) With SMOTE.

Algorithm/Type of ML	Without SMOTE			With SMOTE		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy
Naive Bayes	0.975	0.973	0.973	0.96	0.948	0.948
Logistic Regression	0.958	0.958	0.958	0.962	0.952	0.952
SVM	0.969	0.969	0.969	0.973	0.967	0.967
Decision Trees	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Random Forest	1.0	1.0	1.0	0.998	0.998	0.998

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

Another metric used is recall.

Recall refers to the number of samples rightly anticipated as drinkable (Class 1) by the model divided by all the samples that are actually drinkable (Class 1). It is given as:

$$Recall = \frac{TP}{(TP + FN)} \tag{8}$$

Weighted averages of Precision and Recall used in this review, further helps in the judgement of the accuracy of the model, provided, the dataset is subjected to class imbalance.

The weighted average of any scoring metric is calculated as below:

$$Score_{weighted-avg} = \%Class\ 0 * Score_{class0} + \%Class1 * Score_{class1} \tag{9}$$

Footnotes: TP-True Positives, TN-True Negatives, FP-False Positives, FN-False Negatives.

4. Results

4.1. Traditional model training and results

4.1.1. Binary class data

The binary-class data was trained with the above-mentioned algorithms in two different ways, with SMOTE and without SMOTE. As it can be seen in Table 1, Naive Bayes achieved a precision of 94.5 % and recall of 94.6 % without SMOTE while a precision of 94.5 % and 91.8 % recall with SMOTE. Logistic Regression gives a precision-recall of 91.8%–92 % and 92%–88 % without SMOTE and with SMOTE respectively. SVM performs similarly with a precision-recall of 91.8%–92 % and 92%–88 % without SMOTE and with SMOTE respectively. Decision trees achieved a precision and recall of 99.9 % without SMOTE and a precision of 92.9 % and a recall of 92.7 % with SMOTE. Random Forests performed well with a precision and recall of 99.8 % without SMOTE and a precision of 92.9 % and 92.7 %. Overall, the performance of data without SMOTE is better than the model performance with SMOTE. Decision tree algorithm is the most optimal with an accuracy of 99 % without SMOTE and 92.7 percent with SMOTE for binary class data.

4.1.2. Multiclass data

Similarly, the multi-class data was trained with the above-mentioned algorithms in two different ways, with SMOTE and without SMOTE. From Table 2, it can be Naive Bayes achieved a precision of 97.5 % and recall of 97.3 % without SMOTE while a precision of 96 % and 94.8 % recall with SMOTE. Logistic Regression performed well with a precision and recall of 95.8 % without SMOTE and a precision of 96.2 % and 95.2 %. SVM performs similarly with a precision and recall of 96.9 % and 97.3%–96.7 % without SMOTE and with SMOTE respectively. Decision trees achieved a precision and recall of 100 % with and without SMOTE. Random Forests performed similarly with a precision and recall of 100 % without SMOTE and a precision and recall of 99.8 %. Overall, the performance of decision tree algorithm is the most optimal with an accuracy of 100 % with and without SMOTE for multi-class data.

Table 3
Results of Binary Classification via Auto ML models.

Algorithm	Precision	Recall	Accuracy
Naive Bayes	Not Applicable	Not Applicable	Not Applicable
Logistic Regression	1.0	0.702703	0.914667
SVM	Not Applicable	Not Applicable	Not Applicable
Decision Trees	0.915152	1.0	0.987556
Random Forest	**1.0**	**1.0**	**1.0**

Table 4
Results of Multi-class Classification via Auto ML models.

Algorithm	Precision	Recall	Accuracy
Naive Bayes	Not Applicable	Not Applicable	Not Applicable
Logistic Regression	0.901381	0.938153	0.938153
SVM	Not Applicable	Not Applicable	Not Applicable
Decision Trees	0.974956	0.986934	0.986934
Random Forest	**0.999183**	**0.999129**	**0.999129**

4.2. Automl model training and results

4.2.1. Binary class data

As seen from Table 3, Random Forests performed well with a precision of 91.5 % and recall of 100 %.

4.2.2. Multiclass data

As observed in Table 4, Random Forests performed well with a precision of 99.9 % and recall of 100 %.

5. Discussion

Traditional and AutoML libraries were compared on two types of

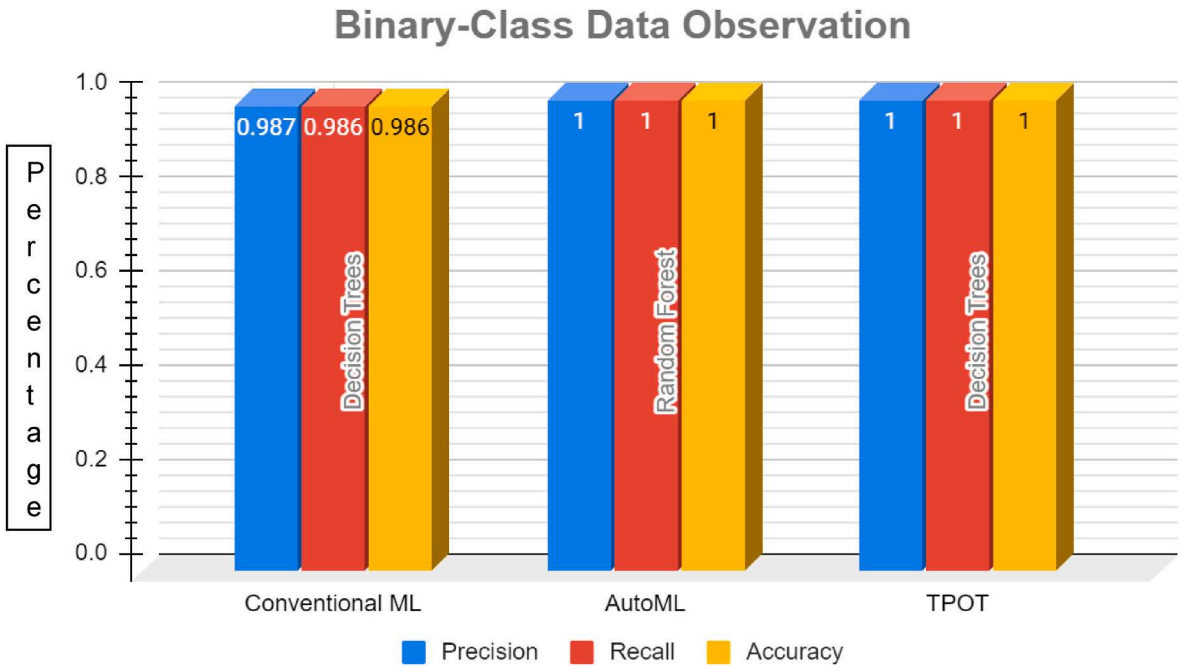

Fig. 23. Bar chart comparison of ML/AutoML techniques for Binary-class data.

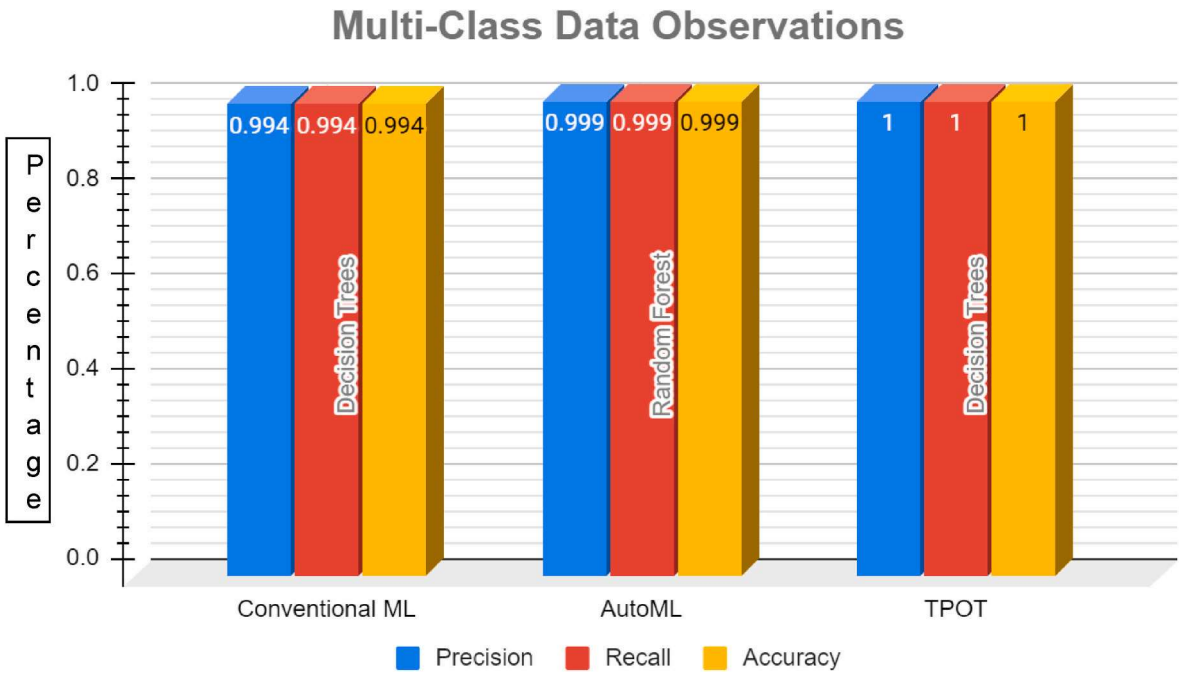

Fig. 24. Bar chart comparison of ML/AutoML techniques for Multi-class data.

data namely binary class and multiclass, the results were plotted with respect to precision, recall and accuracy. The observations obtained are described.

5.1. Comparison of conventional ML and Auto ML libraries on binary-class data

It is observed that from Fig. 23, Decision Trees performs best with a 98.6 % accuracy in the traditional ML pipeline, while Random Forest performs best with a 100 % accuracy in AutoML mljar-supervised and Decision Trees performs best with a 100 % accuracy in TPOT. One can state that with respect to binary class data AutoML libraries had an edge over the traditional ML pipeline. The results prove that the accuracy of AutoML and TPOT was 1.4 % higher than conventional ML techniques for binary class water data.

5.2. Comparison of conventional ML and AutoML libraries on multi-class data

It can also be observed that from Fig. 24 Decision Trees perform best with a 99.4 % accuracy in the traditional ML pipeline while Random Forest performs best with a 100 % accuracy in AutoML *mljar-supervised library* and Decision Trees performs best with a 100 % accuracy in TPOT. One can similarly state that AutoML has an edge over the

traditional ML pipeline. The results prove that the accuracy of AutoML was 0.5% higher and TPOT was 0.6% higher than conventional ML techniques for multi class water data. While comparing the results of AutoML and TPOT with the conventional ML technique in the previous work of (Prasad et al., 2021) there was 7 % increase in accuracy. Conventional ML (Decision Tree) yielded 93 % while AutoML and TPOT methods yielded 100 % accuracy. Conventional ML (Random Forest) yielded 96 % while AutoML and TPOT methods yielded 100 % accuracy. These results applies for both binary and multi class water data.

6. Conclusion

Evaluating traditional and AutoML techniques within the avenue of water quality analysis has yielded insights that may be used for further advancements in the field, considering AutoML is a relatively new addition. Based upon the algorithms tested between traditional and AutoML systems, it can safely be concluded that AutoML is as competent as a conventional model. In fact, the results obtained show a remarkable increase in the performance of algorithms. Moreover, it was proven that every system has a certain scope for customisation and additions within context. For example, from the preliminary stages, data proved to have a profound impact upon the both models. Use of SMOTE increased accuracy, reinforcing the fact that AutoML, efficient as it might be, provides better results when data is cleaned, handled and moulded to suit the purpose. Lastly, all these points are to be viewed in light of the consideration that factors such as time taken, academic experience required are all extremely less in the case of AutoML. One can say that AutoML, in overall terms, can be termed a better tool. The limitation of AutoML is that it will work well for static environment than in the dynamic environment.

Credit author statement

D. Venkata Vara Prasad: Investigation, Methodology and Writing – original draft, P. Senthil Kumar: Conceptualization, Validation and Supervision, Lokeswari Y Venkataramana: Investigation, Resources and Formal analysis, G. Prasannamedha: Investigation, Resources and Formal analysis, S. Harshana: Investigation, Resources and Formal analysis, S Jahnavi Srividya: Investigation, Resources and Formal analysis, K. Harrinei: Investigation, Resources and Formal analysis, Sravya Indraganti: Investigation, Resources and Formal analysis

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahmad, Z., Rahim, N.A., Bahadori, Alireza, Zhang, Jie, 2017. Improving water quality index prediction in Perak River basin Malaysia through a combination of multiple neural networks. Int. J. River Basin Manag. 15 (1), 79–87. https://doi.org/10.1080/15715124.2016.1256297.

Ahmed, U., Mumtaz, R., Anwar, H., Shah, A.A., Irfan, R., García-Nieto, J., 2019. Efficient water quality prediction using supervised machine learning. Water 11 (11), 2210. https://doi.org/10.3390/w11112210.

Barzegar, R., Aalami, M.T., Adamowski, J., 2020. Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. Stoch. Environ. Res. Risk Assess. 34, 415–433. https://doi.org/10.1007/s00477-020-01776-2.

Caruana, Rich, Niculescu-Mizil, Alexandru, Crew, Geoff, Ksikes, Alex, 2004. Ensemble Selection from Libraries of Models. https://doi.org/10.1145/1015330.1015432.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural Architecture Search: A Survey. ArXiv, abs/1808.05377.

Escalante, H., 2020. Automated Machine Learning - a Brief Review at the End of the Early Years. ArXiv, abs/2008.08516.

Haghiabi, Amir, Nasrolahi, Ali, Parsaie, Abbas, 2018. Water quality prediction using machine learning methods. Water Q. Res. J. 53 wqrjc2018025. 10.2166/wqrj.2018.025.

He, Xin, Zhao, Kaiyong, Chu, Xiaowen, 2019. AutoML: A Survey of the State-Of-The-Art.

https://github.com/JahnaviSrividya/Korattur-Lake-Water-Quality-Dataset.

Khan, Y., See, C.S., 2016. Predicting and analyzing water quality using Machine Learning: a comprehensive model. In: 2016 IEEE Long Island Systems, Applications and Technology Conference. LISAT), pp. 1–6. https://doi.org/10.1109/LISAT.2016.7494106.

Mockus, Jonas, 2012. Bayesian Approach to Global Optimization: Theory and Applications.

Prasad, D., Vara, V., Venkataramana, L.Y., Kumar, P.S., Prasannamedha, G., Soumya, K., Poornema, A.J., 2021. Prediction on water quality of a lake in Chennai, India using machine learning algorithms. Desalination Water Treat. 218, 44–51. https://doi.org/10.5004/dwt.2021.26970.

Rajamohan, N., Kumar, P.S., Al Qasmi, F., Rajasimman, M., 2020. Separation of manganese from water using hybrid nanocomposite to control water pollution: kinetic and equilibrium modelling. Int. J. Environ. Anal. Chem. 1–16.

Randrianiaina Jerry, J.C.F., Rakotonirina Rija, I., Ratiarimanana Jean, R., Fils, Lahatra Razafindramisa, 2019. Modelling of lake water quality parameters by deep learning using remote sensing data. Am. J. Geogr. Inf. Syst. 8 (6), 221–227. https://doi.org/10.5923/j.ajgis.20190806.01.

Shawi, R.E., Maher, M., Sakr, S., 2019. Automated Machine Learning: State-Of-The-Art and Open Challenges. ArXiv, abs/1906.02287.

Solanki, Archana, Agrawal, Himanshu, Khare, Kanchan, 2015. Predictive analysis of water quality parameters using deep learning. Int. J. Comput. Appl. 125, 29–34. https://doi.org/10.5120/ijca2015905874.

Sujatha, S., Rajamohan, N., Anbazhagan, S., Vanithasri, M., Rajasimman, M., 2021. Extraction of nickel using a green emulsion liquid membrane–Process intensification, parameter optimization and artificial neural network modeling. Chem. Eng. Processing Proc. Intensification 165, 108444.

Thornton, Chris, Hutter, Frank, Hoos, Holger, Leyton-Brown, Kevin, 2012. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. KDD. https://doi.org/10.1145/2487575.2487629.

Venkata Vara Prasad, D., Venkataramana, Lokeswari Y., Senthil Kumar, P., Prasannamedha, G., Soumya, K., Poornema, A.J., 2020. Water quality analysis in a lake using deep learning methodology: prediction and validation. Int. J. Environ. Anal. Chem. https://doi.org/10.1080/03067319.2020.1801665.

Yang, Li, Shami, Abdallah, 2020. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice.

Yao, Q., Wang, M., Escalante, H., Guyon, I., Hu, Y., Li, Y., Tu, W., Yang, Q., Yu, Y., 2018. Taking Human Out of Learning Applications: A Survey on Automated Machine Learning. ArXiv, abs/1810.13306.

Zöller, M., Huber, M., 2019. Survey on Automated Machine Learning. ArXiv, abs/1904.12054.