informatics

MDPI

Article

Prompt Design through ChatGPT's Zero-Shot Learning Prompts: A Case of Cost-Sensitive Learning on a Water Potability Dataset

Kokisa Phorah , Malusi Sibiya * and Mbuyu Sumbwanyambe

Florida Campus, University of South Africa, Johannesburg 1709, South Africa; phorake@unisa.ac.za (K.P.); sumbwm@unisa.ac.za (M.S.)

* Correspondence: sibiym@unisa.ac.za

Abstract: Datasets used in AI applications for human health require careful selection. In healthcare, machine learning (ML) models are fine-tuned to reduce errors, and our study focuses on minimizing errors by generating code snippets for cost-sensitive learning using water potability datasets. Water potability ensures safe drinking water through various scientific methods, with our approach using ML algorithms for prediction. We preprocess data with ChatGPT-generated code snippets and aim to demonstrate how zero-shot learning prompts in ChatGPT can produce reliable code snippets that cater to cost-sensitive learning. Our dataset is sourced from Kaggle. We compare model performance metrics of logistic regressors and gradient boosting classifiers without additional code fine-tuning to check the accuracy. Other classifier performance metrics are compared with results of the top 5 code authors on the Kaggle scoreboard. Cost-sensitive learning is crucial in domains like healthcare to prevent misclassifications with serious consequences, such as type II errors in water potability assessment.

Keywords: ChatGPT; water potability; water quality; machine learning; zero shot-learning; cost-sensitive learning

check for updates

Citation: Phorah, K.; Sibiya, M.; Sumbwanyambe, M. Prompt Design through ChatGPT's Zero-Shot Learning Prompts: A Case of Cost-Sensitive Learning on a Water Potability Dataset. *Informatics* **2024**, *11*, 27. https://doi.org/10.3390/informatics11020027

Academic Editor: Zhiwen Yu

Received: 27 March 2024
Revised: 19 April 2024
Accepted: 23 April 2024
Published: 28 April 2024

1. Introduction

The word "potability" refers to the quality or state of being suitable for drinking or consumption. In the context of water, potability refers to whether water is safe and clean enough for humans to drink without causing harm or illness. Assessing the potability of water involves various scientific and regulatory considerations to ensure that it meets certain standards and does not pose health risks to consumers. Recent advancements in water quality detection methods have been observed in the research conducted by Yaroshenko et al. [1]. Their study offers a comprehensive review of the suitability of various technologies for real-time water quality monitoring, particularly those tested in practical settings. The performance of sensors based on molecularly imprinted polymers is extensively evaluated, shedding light on their operational principles, stability in real-world applications, and potential for mass production [1]. The deployment of sensors beyond laboratory settings for water quality detection has led to the emergence of dataset repositories. In this study, we focus on how generative AI can ensure the utilization of high-quality datasets for predicting water quality with ML algorithms. Through the ChatGPT platform, we were able to generate Python code snippets by parsing prompts, aimed at cleaning the dataset through cost-sensitive learning techniques. The methodology section provides a detailed explanation of the prompt design aimed at achieving enhanced results.

The impact of this approach is demonstrated by comparing the performance metrics of the chosen classifier without fine tuning the original generated code snippets.

2. Literature Review

A significant amount of research has been dedicated to water quality assessment and monitoring. The evidence in the literature suggests that traditional scientific methods of water quality assessment have been supplanted by artificial methods, involving the use of sensors and ML algorithms in the monitoring processes. However, it is important to note the distinction between water quality and water potability. To distinguish between water quality and water potability, this section reviews the approaches employed by researchers in these two disciplines. The efficacy of machine learning (ML) techniques for water quality prediction was investigated in a recent study by Yaroshenko et al. A machine learning classifier model was constructed using real-world data, with all measured characteristics utilized as significant features. The dataset was partitioned into training and testing subsets, and various ML algorithms were employed, with support vector machine and k-nearest neighbor demonstrating superior performance in terms of F1-score and ROC AUC values. Conversely, the LASSO-LARS and stochastic gradient descent methods exhibited higher recall values [1]. The Root Zone Water Quality Model (RZWQM), developed by USDA–ARS scientists, integrated physical, chemical, and biological processes to simulate water and agrochemical movement in agricultural fields. Ahuja et al. evaluated the model's performance using field data, demonstrating reasonable simulation of soil water movement and pesticide persistence [2]. Shrestha and Kazama applied multivariate statistical techniques to assess temporal and spatial variations in water quality within the Fuji River basin. Cluster analysis categorized sampling sites into pollution gradient clusters, while factor analysis revealed key factors driving water quality variations across different pollution levels. Discriminant analysis effectively reduced data dimensionality and identified indicator parameters for water quality assessment, pollution source identification, and river water quality management [3]. The utilization of multivariate statistical techniques for the evaluation and interpretation of water quality datasets was highlighted in the study by researchers focusing on the Gomti river in India [4]. Through cluster analysis, distinct groups within the river's catchment regions were identified, while factor analysis/principal component analysis revealed key factors responsible for variations in water quality across different catchment areas. Discriminant analysis facilitated data reduction and pattern recognition, aiding in the identification of indicator parameters for effective water quality management. Additionally, receptor modeling techniques provided insight into pollution sources/factors contributing to river contamination.

The Athens Water Supply and Sewerage Company (EYDAP SA) played a crucial role in supplying potable water to millions of inhabitants in Attica, Greece [5]. Stringent quality control measures were enforced, with thorough analysis conducted to ensure compliance with established guidelines. Statistical tools were employed to enhance quality control processes, with particular attention given to parameters such as turbidity, residual chlorine, and aluminum levels. Statistical process control techniques were utilized to evaluate control limits and improve process quality. In addressing the imperative need for accurate detection and identification of contaminants in drinking water, a real-time event adaptive detection, identification, and warning (READiw) methodology was explored [6]. Through pilot-scale pipe flow experiments, various chemical and biological contaminants were examined, with adaptive transformation techniques enhancing sensor detection capabilities. Kinetic and chemical differences among contaminants allowed for their distinguishability, providing a reliable method for contamination event identification. The optimization and artificial intelligence (AI) techniques applied in the simulation and operation of the Barra Bonita reservoir in Brazil were elucidated in the methodology proposed by Chaves et al. [7]. A fuzzy stochastic dynamic programming model was developed to calculate optimal operation procedures, considering multiple fuzzy objectives. The Markov chain technique handled the stochastic nature of river flow, while the water quality analysis employed artificial neural network models to predict organic matter and nutrient loads based on river discharge. The proposed methodology demonstrated efficacy in reservoir operation, providing a valuable tool for water resource management. The necessity for comprehensive information on water quality, especially concerning sediment, phosphorus, and nitrogen exports from catchments, is emphasized by catchment managers and stakeholder groups [8]. Due to the limited availability of intensive spatial and temporal data on nutrient concentrations or loads, there's a demand for nutrient export models capable of providing valuable insights with sparse data inputs. This paper evaluates four such models and various direct estimation methods for their efficacy in predicting loads in Australian catchment scenarios. The discussion underscores the significance of coordinated data collection over extended periods and fine temporal scales to improve load prediction accuracy.

Artificial neural network procedures were employed to define and predict diatom assemblage structures in Luxembourg streams using environmental data [9]. Self-organizing maps (SOM) classified samples based on their diatom composition, while a multilayer perceptron with a backpropagation learning algorithm (BPN) predicted these assemblages. Classical methods were then utilized to identify relationships between diatom assemblages and SOM cell numbers. The study demonstrated a high predictability of diatom assemblages using physical and chemical parameters within a limited geographical area. In planning sampling regimes, minimizing estimation error or sampling effort for a desired accuracy is essential [10]. This paper compares classical and geostatistical approaches for matching sampling effort to accuracy using airborne thematic mapper images of British lakes. It illustrates that the systematic scheme outperforms the random scheme, especially with increasing sample size and spatial dependence. The study underscores the necessity of calibrating sampling regimes to the spatial dynamics of the lake and suggests remote sensing as an ideal means for determining such dynamics. A pilot study was conducted to assess the hormonal activity of freshwaters in Victoria using recombinant receptor-reporter gene bioassays [11]. Water samples from the Yarra River were analyzed for toxicity, genotoxicity, and receptor assay activity. The results indicated weak to moderate toxicity with no significant location-based trends along the river. Estrogenic, thyroid, and retinoic acid receptor activity was negligible, while AhR activity increased downstream, possibly influenced by bush fires. Approximately 24% of total AhR activity was associated with suspended solids. The preceding reviews focused on evaluating various aspects of water quality. In the following reviews, the focus shifts towards assessing water potability. These reviews examine the suitability of water for human consumption, considering factors such as chemical composition, microbial contamination, and adherence to regulatory standards. Nyende-Byakika et al. provide insights into the raw water quality of Bospoort dam in South Africa [12]. Through a comprehensive time-series analysis, various parameters were monitored, revealing that while most parameters remained within recommended threshold levels for the majority of the study period, conductivity, hardness, and high coliform counts exceeded acceptable limits. The water exhibited excessive hardness and high conductivity, surpassing alarm levels for a considerable portion of the study duration despite dissolved solids being below their alarm thresholds. Notably, total coliform and E. coli counts were found to be significantly elevated, indicating potential microbial contamination concerns. Pehlivan and Emre investigate the environmental and hydrological processes in the Sarma Stream basin, located southwest of Akcakoca in the Duzce Province of Turkey [13]. Samples from various sources, including rocks, soil, stream water, rain, snowmelt, and bed and suspended sediment, were collected and analyzed. The study reveals that sandstone and soil samples contribute to the stream's muddy flow during the rainy season, with chlorite-type minerals prevalent in the bed and suspended sediments. The water chemistry indicates a calcium bicarbonate-rich composition, influenced by acid rain and containing elevated levels of certain heavy metals and elements, necessitating treatment of water in the Sariyayla Reservoir.

Comparatively, a study by an undisclosed author assesses potable water filtration methods commonly used in rural Ghanaian communities [14]. Physico-chemical and microbiological analyses were conducted on water samples from ponds, dams, and rivers, revealing elevated levels of total suspended solids, turbidity, total coliforms, and bacterial counts. However, filtration methods, including ceramic filters and household sand filters, effectively reduced these parameters to acceptable standards. The study suggests that a combination of filtration methods, including the use of alum and activated carbon, could further improve water quality, recommending follow-up research in this area. Elizabeth and Rajpramikh monitored the microbiological and physico-chemical parameters of drinking water samples from two villages in the Vizianagaram District. Borewell water from Somi Naiduvalasa village exhibited elevated coliform levels and exceeded permissible limits for various physic-chemical parameters, rendering it non-potable [15]. In contrast, the historical shift in perceptions of water quality assessment was discussed, highlighting the transition from sensory-based evaluations to reliance on standardized analytical methods. This shift, driven by institutional and regulatory practices, marginalized consumer sensory knowledge as merely aesthetic, focusing instead on objective analytical data. However, the exclusion of sensory information from water quality assessment overlooks the subjective experiences of consumers, calling for new practices that engage consumers as valuable participants in ensuring water quality [16]. Furthermore, an evaluation of the water potability of various regions in Ludhiana, Punjab, revealed suboptimal potability levels despite acceptable hardness and pH values. Physicochemical and bacteriological analysis conducted across six areas of Ludhiana city showed low levels of potability, highlighting the need for interventions by local water authorities to ensure the supply of safe drinking water to the population [17]. The integration of sensor materials into new-generation transducers and the use of household electronic devices for signal registration offer potential for the development of economical, portable detectors operating in real-time mode [18].

In another study, physico-chemical and microbial parameters of water quality in hand-dug wells in Bolgatanga, Ghana, were assessed. The study revealed elevated coliform levels in dry seasons and increased concentrations of various parameters during the rainy season, suggesting infiltration from stormwater. The proximity to pollution sources also influenced coliform counts, indicating the need for the disinfection of well water before use [19]. Similarly, research on the potability of packaged sachet water within the Federal University of Agriculture, Abeokuta campus, Nigeria, found that while physicochemical parameters met WHO and Nigerian standards, bacteriological analysis revealed total bacteria count in all samples and contamination with total coliforms in two brands. The study underscores the importance of routine water quality examination and regulatory oversight to ensure safe drinking water supply [20]. The assessment of water quality in a village involved analyzing various physicochemical parameters and calculating a water quality index. While most parameters met Indian standards, coliform levels exceeded permissible limits, indicating contamination and leading to waterborne diseases. Although some water sources were classified as excellent, the disinfection of coliform before use was recommended [21]. Furthermore, groundwater samples from different areas in Ariyalur District, Tamil Nadu, were analyzed for various physicochemical parameters. The majority of samples were found unsuitable for drinking purposes, highlighting the need for comprehensive water quality management [22].

Thus far, researchers have found ChatGPT to be beneficial in several areas, leading to its application in various research endeavors [23–28]. Our proposed approach is centered on ChatGPT's zero-shot learning prompt design for generating code snippets to preprocess the data, aiming to improve cost-sensitive learning metrics when training a classifier model. ChatGPT is a chatbot developed by OpenAI that uses a large language model to engage in conversation, answer questions, and provide information on a wide range of topics [29]

3. Materials and Methods

Misclassification by ML classifiers results in either type I or type II errors. In ML applications where the health of people comes first, it is important to design ML classifiers for cost-sensitive learning. Cost-sensitive learning, also known as imbalanced learning or asymmetric learning, is a technique in machine learning where the classifier's training process is adjusted to account for the unequal costs associated with different types of errors, including type I and type II errors. The birth of ChatGPT has bridged the gap between academics who lack coding skills but can only solve problems through mathematical presentations, for instance. Similarly, this study aims to introduce prompt design on ChatGPT for the generation of code snippets based on cost-sensitive learning to bridge the gap between the lack of coding skills and theoretical knowledge. In this study, cost-sensitive learning is demonstrated on a water potability dataset. Through a set of prompts on ChatGPT, various Python code snippets were generated. The aim of this study is not to update the generated code snippets for better performance of the chosen classifier but to provide insight of designing zero-shot learning prompts for cost-sensitive learning. Figure 1 shows zero-shot learning prompts on ChatGPT. In the context of prompt design, the two famous types of prompts are zero-shot learning and few-shot learning. Zero-shot learning prompts are designed to enable a model (ChatGPT in this context) to generate responses or make predictions for classes or categories that it has never been explicitly trained on. Few-shot learning prompts, on the other hand, are designed to help a model (ChatGPT in this context) adapt quickly to new tasks or classes with only a small number of examples or shots per class. It can be seen in Figure 1 that each prompt from prompt 1 to prompt 8 of the design prompts generates a Python code snippet. The Python code is later updated by the user to add the water potability dataset.

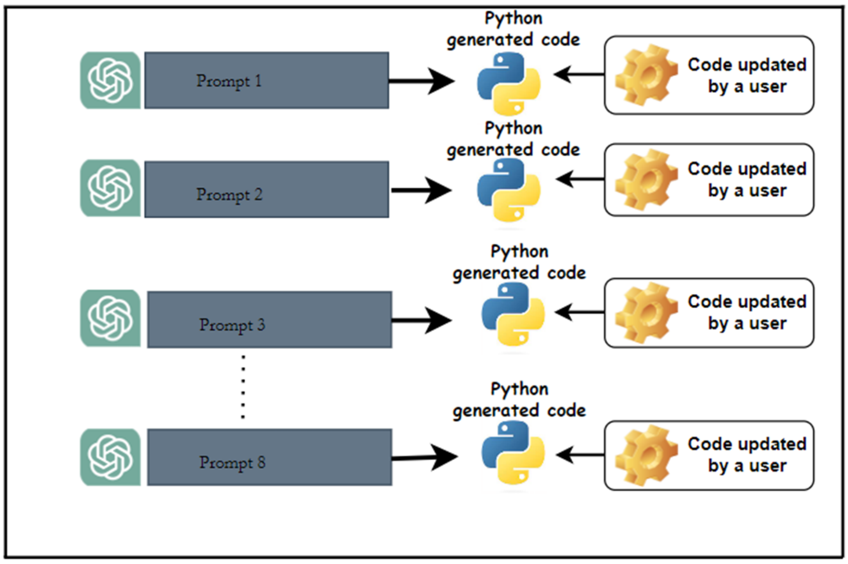

Figure 1. Zero-shot learning prompts on ChatGPT.

In this study, zero-shot learning prompts were developed to generate code snippets for cost-sensitive learning on a water potability dataset. The generated code snippets were tested on Jupyter IDE (Appendix A). Although the aim was not to tune the classifiers, the logistic regressor and gradient boosting classifiers were used to demonstrate the zero-shot learning prompts. To incorporate the water potability dataset into the generated code snippets, the Pandas library was used to read the CSV file and convert it to the Pandas DataFrame, which was further utilized to train the classifiers. Below are the eight carefully designed prompts recommended for generating code snippets when considering cost-sensitive learning (Appendix A).

1. Feature Engineering:

Prompt: "Generate Python code snippet to perform feature engineering on a [Classifier name]. Include techniques such as adding interaction terms, creating polynomial features, and transforming variables. Show all the classification metrics of the model".

2. Handling Imbalanced Classes:

Prompt: "Generate Python code snippet to handle class imbalance in a [Classifier name]. Include techniques such as oversampling, under-sampling, or using weighted loss functions to address imbalanced class distributions. Show all the classification metrics of the model".

3. Regularization:

Prompt: "Write Python code to implement regularization techniques on a [Classifier name]. Include options for controlling tree complexity and learning rate to prevent overfitting. Show all the classification metrics of the model".

4. Hyperparameter Tuning:

Prompt: "Generate Python code snippet for hyperparameter tuning of a [Classifier name]. Include techniques like grid search or random search to optimize hyperparameters such as tree depth, learning rate, and the number of estimators. Show all the classification metrics of the model".

5. Ensemble Methods:

Prompt: "Write Python code to implement ensemble methods for improving a [Classifier name] performance. Include techniques such as bagging, boosting, or stacking to combine multiple models. Show all the classification metrics of the model".

6. Cross-Validation:

Prompt: "Generate Python code snippet to perform k-fold cross-validation on a [Classifier name]. Ensure that the code evaluates model performance accurately and reliably. Show all the classification metrics of the model".

7. Feature Selection:

Prompt: "Write Python code to select relevant features for a [Classifier name]. Include techniques such as recursive feature elimination or feature importance ranking to improve model simplicity and performance. Show all the classification metrics of the model".

8. Optimizing Decision Threshold:

Prompt: "Write Python code to optimize the decision threshold of a [Classifier name]. Include techniques to adjust the threshold to balance between sensitivity and specificity based on specific requirements. Show all the classification metrics of the model".

After comparing the accuracies of the authors' code without parameter optimization with the code generated by ChatGPT, we proceeded to compare additional classifier metrics, particularly focusing on the logistic regressor, with those of the top 5 Kaggle authors listed on the Kaggle scoreboard (https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability/code) (accessed on 2 February 2024).

While zero-shot learning on ChatGPT is fascinating, delving into the history of transformers provides valuable insights into the technological advancements that paved the way for the existence and capabilities of ChatGPT. Transformers have revolutionized natural language processing (NLP) and artificial intelligence (AI) tasks, offering unprecedented capabilities in understanding, and generating human-like text. Among the forefront pioneers in leveraging transformers is OpenAI's ChatGPT. ChatGPT is built upon the transformer architecture, a paradigm-shifting model introduced by Vaswani et al. [30]. Unlike traditional recurrent neural networks (RNNs) or convolutional neural networks (CNNs), transformers employ self-attention mechanisms to capture global dependencies within input sequences, enabling efficient parallelization and long-range context modeling. Figure 2 shows the transformer model architecture [30].

In conclusion, transformers have propelled ChatGPT to the forefront of AI-powered natural language processing. With its transformative capabilities, ChatGPT represents a significant milestone in the evolution of conversational AI and holds immense potential for diverse applications. As depicted in Figure 2, the transformer relies on attention, a mechanism that enables each word in a sentence to consider all other words' relevance, surpassing the limitations of sequential models like RNNs or LSTMs. Through self-attention, words compute their attention scores with respect to all others, facilitating an understanding of their contextual relationships. Multi-head attention allows the model to capture diverse aspects of these relationships simultaneously. To address sequence order, positional encodings are incorporated into input embeddings. Following self-attention, a feedforward neural network further processes the output, reducing dimensionality. Residual connections around each sub-layer and layer normalization aid in training deeper models. In tasks like translation, transformers employ an encoder–decoder architecture where the encoder processes the input sequence, and the decoder generates the output sequence.

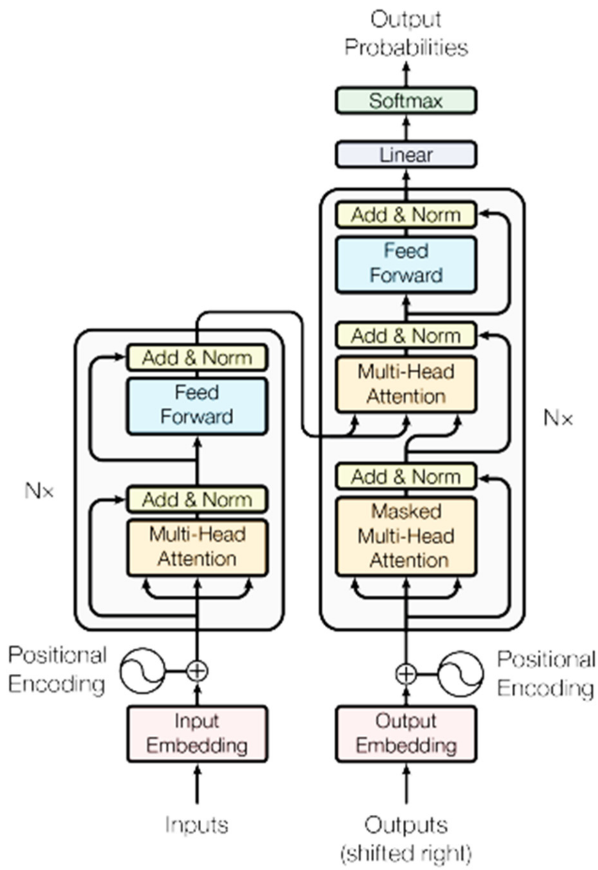

Figure 2. The transformer model architecture [30].

In the Results section, we discuss the classifier performance results and the shortcomings observed in the code snippets generated by zero-shot learning prompts when incorporated with the water potability dataset.

4. Results and Discussion

The water potability dataset consisted of 10 columns of which 9 were inputs and 1 a target. Figure 3 illustrates the distribution of classes in the target column of the water potability dataset, and the code to view this distribution was generated by the authors to provide insight into the data handled by ChatGPT for code generation. The prompts on ChatGPT did not encompass the classification distribution information, as the focus was on investigating the capability to generate code with performance metrics. It can be seen in Figure 3 that there is an imbalance of classes, and the predictions may not be accurate. In cases where classes are imbalanced, a few approaches can be used to mitigate class imbalance. One approach commonly used to address imbalanced classes in a classification problem is resampling. This can involve either oversampling the minority class (creating more instances of the minority class) or under sampling the majority class (removing instances of the majority class). Another approach is to use different evaluation metrics such as F1-score, precision, and recall, which can provide a better understanding of model performance when dealing with imbalanced classes. Additionally, ensemble methods like boosting algorithms (e.g., AdaBoost, XGBoost) and bagging algorithms (e.g., Random Forest) can also help mitigate class imbalance by adjusting the weights of samples or aggregating predictions from multiple classifiers.

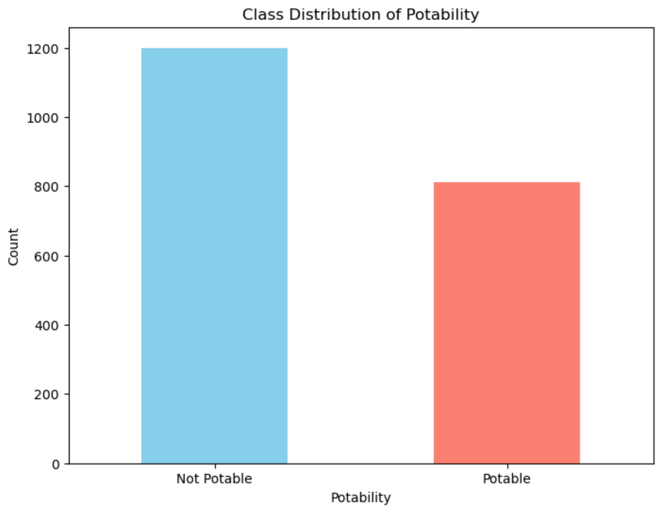

Figure 3. Target class distribution in the water potability dataset.

As previously stated, the primary focus of this study was not on enhancing the performance metrics of the classifiers utilized, but rather on presenting a set of zero-shot learning prompts tailored for ChatGPT to generate code snippets conducive to cost-sensitive learning. To illustrate this, the logistic regressor and gradient boosting classifiers were employed. The performance metrics of these models were not subjected to further refinement for enhancement, as the primary objective of the study was to formulate a series of zero-shot learning prompts for generating code snippets pertinent to cost-sensitive learning. These generated code snippets were evaluated using a water potability dataset, chosen due to its relevance for models designed for cost-sensitive learning. A comparative analysis of these models, along with their associated prompts, is presented in tabular form, and the results are subsequently discussed. Figure 4 provides a simplified visual representation of the previously described process.

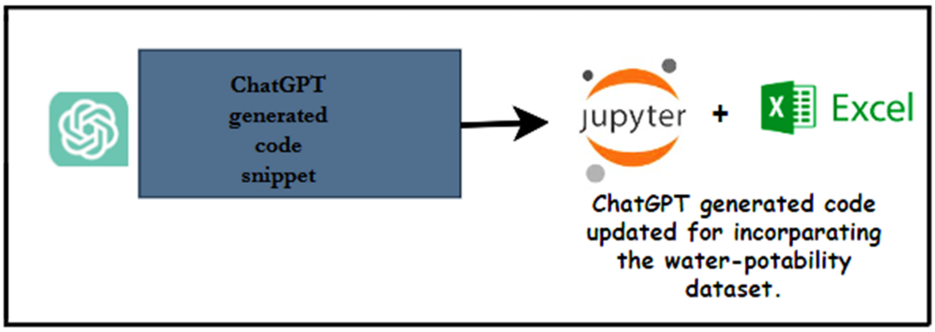

Figure 4. A pictorial depiction of the code generation process.

Overview of Classifier Metrics for Different Generated Code Snippets

Below is a summary of the classifiers that were used in the study to demonstrate the performance of the code snippets that were generated through zero-shot learning prompts on ChatGPT.

Logistic regression tackles binary classification problems by modeling the relationship between features (X) and the probability of belonging to the positive class (y = 1) using a linear model. This model's output is transformed by the sigmoid function to ensure probabilities between 0 and 1. The model is then optimized using maximum likelihood estimation to find the best coefficients for the linear model. Finally, a threshold is used on the predicted probabilities to classify new data points.

Gradient boosting classifiers (GBCs) tackle classification by combining multiple weak learners, often decision trees. These learners are added sequentially, with each focusing on correcting the errors of the previous ones. GBC uses loss functions to quantify errors and pseudo-residuals to pinpoint areas for improvement. Each weak learner is trained on these pseudo-residuals, and their predictions are combined to make final classifications. This ensemble approach offers flexibility, interpretability, and improved accuracy compared to single models.

Table 1 presents a summary of the classifier accuracies obtained from the generated code snippets using zero-shot learning prompts on ChatGPT.

Table 1. Comparison between classifier accuracy based on prompts generated by humans and by ChatGPT.

Cost-Sensitive Learning Approach	Zero-Shot Learning Prompts on ChatGPT	Metric Results of the Human-Written Code Using Logistic Regressor as a Base Classifier NB: Cost-Sensitive Learning Not Catered for.	Metric Results of the ChatGPT-Generated Code Snippets [Both Logistic Regressor and Gradient Boosting Classifiers] (Appendix A) NB: Cost-Sensitive Learning Is Catered for.
1. Feature Engineering	Prompt 1	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Accuracy \rightarrow 66\%$ **Gradient boosting:** $Accuracy \rightarrow 65\%$
2. Handling Imbalanced Classes	Prompt 2	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Accuracy \rightarrow 43\%$ **Gradient boosting:** $Case1: Accuracy \rightarrow 60.7\%$ $Case2: Accuracy \rightarrow 62\%$ $Case3: Accuracy \rightarrow 58.5\%$
3. Regularization	Prompt 3	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Accuracy \rightarrow 57\% (For\ both\ L1\ and\ L2)$ **Gradient boosting:** $Accuracy \rightarrow 64\%$
4. Hyperparameter Tuning	Prompt 4	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Accuracy \rightarrow 57\%$ **Gradient boosting:** $Accuracy \rightarrow 66\%$
5. Ensemble Methods	Prompt 5	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Bagging\ Accuracy: 57\%$ $AdaBoost\ Accuracy: 57\%$ $Stacking\ Accuracy \rightarrow 59.8\%$ **Gradient boosting:** $Case1: Base\ Classifier\ Accuracy \rightarrow 64\%$ $Case1: Bagging\ Ensemble\ Accuracy \rightarrow 65\%$ $Case2: Base\ Classifier\ Accuracy \rightarrow 64\%$ $Case2: Boosting\ Ensemble\ Accuracy \rightarrow 59\%$ $Case3: Stacking\ Ensemble\ Accuracy \rightarrow 65\%$
6. Cross-Validation	Prompt 6	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Mean\ accuracy \rightarrow 60.5\%$ $Standard\ deviation \rightarrow 0.012$ **Gradient boosting:** $Mean\ accuracy: 64\%$ $Standard\ deviation: 0.0157$
7. Feature Selection	Prompt 7	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Model\ Accuracy\ with\ Selected\ Features \rightarrow 57\%$ $Model\ Accuracy\ with\ Top\ Features \rightarrow 57\%$ **Gradient boosting:** $Accuracy\ with\ selected\ features \rightarrow 67\%$
8. Optimizing Decision Threshold	Prompt 8	Logistic regressor is a base model: $Accuracy \rightarrow 57\%$	**Logistic regressor:** $Accuracy\ with\ optimal\ threshold \rightarrow 57.5\%$ **Gradient boosting:** $ROC\ AUC \rightarrow 66\%$

Figure 5 is a histogram that offers a visual representation of the variability in accuracy values for both models, highlighting differences in performance and distribution characteristics.

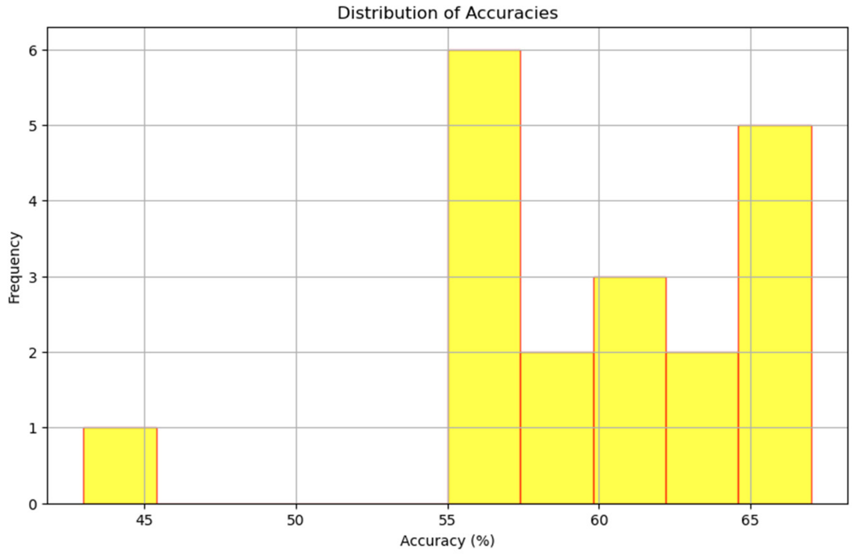

Figure 5. Distribution of accuracy values for the logistic regressor and gradient boosting classifiers (distribution by both ChatGPT models mixed).

Logistic regressor classifier: In the provided histogram, the accuracy distribution for the logistic regressor model spans from approximately 43% to 66%. Notably, there is a concentration of accuracy values around 57%, with a few instances deviating from this central value, ranging from as low as 43% to as high as 66%. The distribution exhibits a somewhat skewed pattern, with more occurrences towards the lower end of the accuracy range.

Gradient boosting classifier: For the gradient boosting model, accuracy values range from approximately 58.5% to 67%. Unlike the logistic regressor, the distribution is relatively more consistent, with most accuracy values clustered between 60% and 67%. Fewer extreme values are observed, indicating a more stable performance. The distribution appears to be slightly right-skewed, with more occurrences towards the higher end of the accuracy range.

Thus far, the explained results are founded on accuracy. Nevertheless, it is crucial to acknowledge that other performance metrics need to be taken into account in case-sensitive learning. The subsequent results are showcased for alternative performance metrics, excluding accuracy, which was detailed in Table 1. Unlike the accuracy results in Table 1, which were compared against the model performance of the code written by the authors, the following performance metrics generated by ChatGPT's code were compared against those achieved by Kaggle code authors who ranked at the top of the scoreboard for water potability predictions using the same dataset. The results of the top 5 code authors on the Kaggle scoreboard were selected to compare their model outcomes with ChatGPT's performance results (https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability/code) (accessed on 2 February 2024).

1. Additional experiment results based on Prompts 1–4

To demonstrate additional metrics beyond accuracy, as shown in Table 1, we specifically examined the results of the logistic regression code generated by ChatGPT. The Precision, Recall, F1-score, and ROC AUC results were consistent across logistic regression models generated by ChatGPT prompts 1–4, as depicted in Table 1. Based on the performance of models generated by prompts 1–4, it appears that the models failed to discern meaningful patterns from the data and are essentially making random predictions. This could stem from various factors such as insufficient data, inappropriate model selection, or a lack of feature relevance.

At the time of writing this article, the authors of "Neural Network from scratch using PyTorch", ranked second on the Kaggle scoreboard, and achieved the performance metrics depicted in Figure 6 (https://www.kaggle.com/code/sidhaarth2110035/neural-network-from-scratch-using-pytorch) (accessed on 2 February 2024).

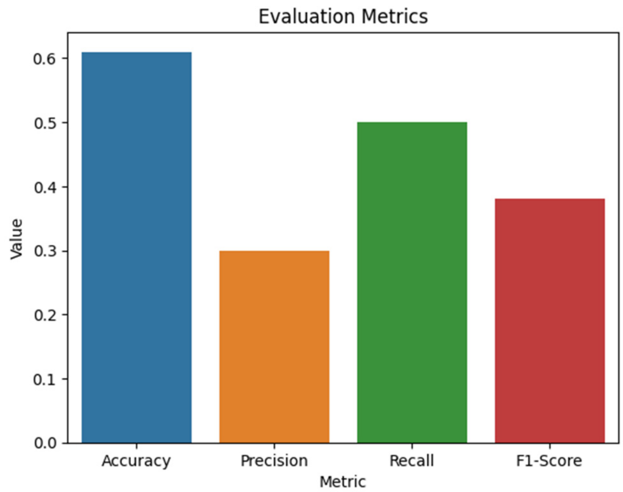

Figure 6. Performance metrics of the model by Kaggle authors of "Neural Network from scratch using PyTorch".

The model developed by the aforementioned Kaggle code authors prove to outperform the models generated by invoking prompts 1–4.

2. *Additional experiment results based on Prompts 5–8*

Invoking prompt 6 on ChatGPT resulted in the generation of a code snippet with model performance results similar to those displayed in Table 2. However, Table 3 showcases the results of the models developed using the code generated by invoking prompt 5. Based on the results provided, the "Stacking model" demonstrated significant Precision, Recall, F1-Score, and ROC AUC values compared to the Bagging and AdaBoost models. Referring to both Figure 6 and Table 1, it is evident that starting from the accuracy of the "Stacking model" presented in Table 1 to Precision, Recall, and F1-Score displayed in Table 3, the ChatGPT code outperforms the Kaggle authors in Precision, Recall, and F1-Score. The Kaggle authors' model resulted in an accuracy approximately equal to that of ChatGPT's "Stacking model" (≈62%).

Table 2. Additional performance metrics by ChatGPT logistic regression model.

Metrics	Result	Description
Precision	0%	A precision of 0% indicates that none of the positive predictions made by the model were correct.
Recall	0%	A recall of 0% indicates that the model failed to correctly identify any of the actual positive instances.
F1-Score	0%	An F1-score of 0% indicates that both precision and recall are extremely low.
ROC AUC	49%	ROC AUC is close to 0.5, indicating that the model's ability to distinguish between classes is almost equivalent to random chance.

Table 3. Performance of models generated by the ChatGPT code on invoking prompt 5.

Model	Precision	Recall	F1-Score	ROC AUC
Bagging	0%	0%	0%	50%
Adaboost	0%	0%	0%	50%
Stacking	56%	49%	52%	60%

The performance of the "Stacking model" in Table 3 is visually represented by Figure 7.

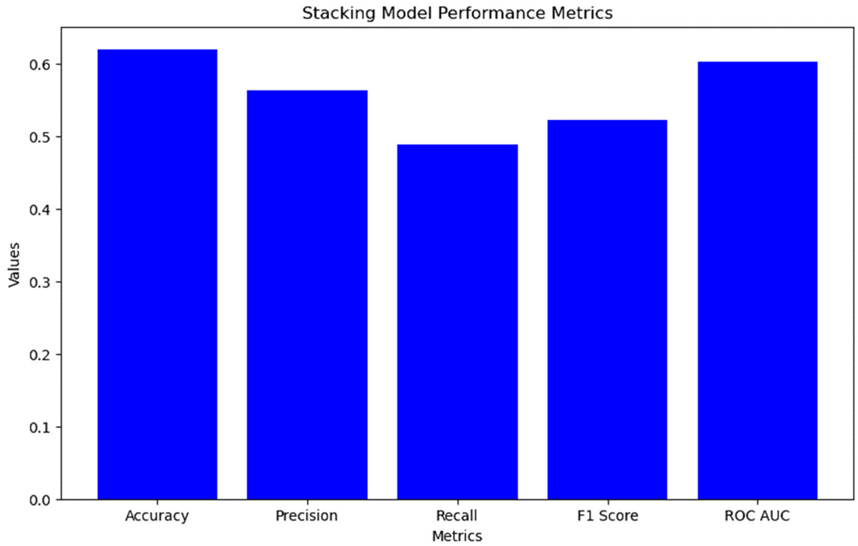

Figure 7. "Stacking model" performance as displayed in Table 3.

The results obtained by invoking prompt 7 on ChatGPT are presented in Table 4. A logistic regression model was developed in two versions: one with selected features and another with top features. However, both models showed poor performance across all evaluation metrics. Although the 'Top Features' model slightly outperformed the other in terms of precision, it still demonstrated very low recall and overall effectiveness, indicating room for improvement. When compared to the results of the Kaggle authors depicted in Figure 6, the models displayed in Table 4 outperform them across all metrics except for the precision of the "Top Features" model.

Table 4. Performance of models generated by the ChatGPT code on invoking prompt 7.

Model	Precision	Recall	F1-Score	ROC AUC
Selected Features	0%	0%	0%	50%
Top Features	100%	0.5%	1.1%	50%

Figure 8 depicts the performance of the "Top Features model" displayed in Table 4.

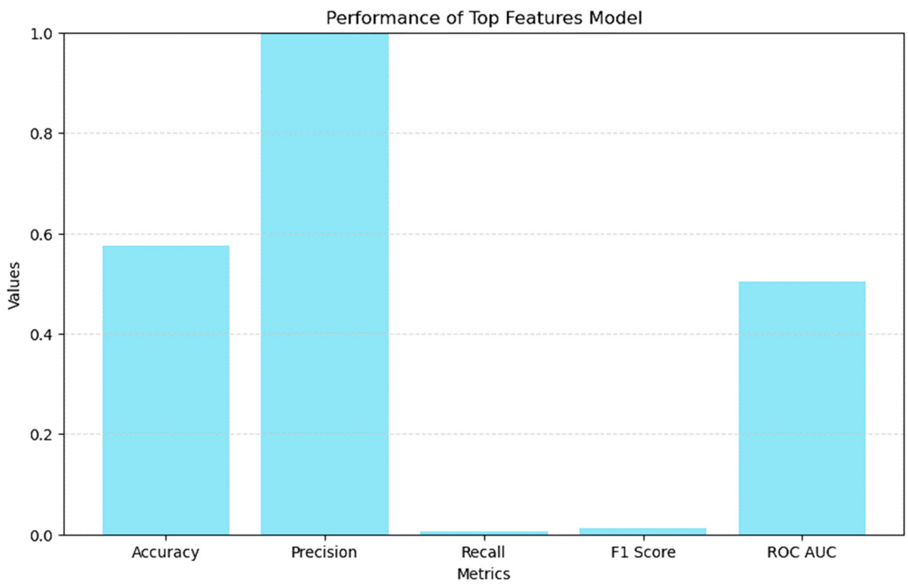

Figure 8. "Top Features model" performance as displayed in Table 4.

The results derived from invoking prompt 8 on ChatGPT closely resemble those of the "Top Features model" depicted in Table 4. Appendix A provides information on accessing the code snippets for the experiments conducted in this study.

The ChatGPT prompts we developed aim to generate code snippets that can optimize the classifier models for cost-sensitive learning. It is important to understand that the models with the best performance metrics are the ones to consider for further improvement. In this study, models with metrics resulting in 0% performance are clearly not suitable for further optimization. The prompts that led to better performance metrics, like those used for the "Stacking" and "Top Features" models in this study, are the ones to adopt and refine their generated code snippets for even better performance. It is also important to realize that the way data are structured can affect how well a model performs. This is particularly significant due to the curse of dimensionality, where high-dimensional data can pose challenges for machine learning models. As a result, prompts showing 0% metrics might have performed better in different datasets due to variations in data structure and dimensionality.

5. Discussions

Furthermore, the variability in accuracy values for logistic regression and gradient boosting classifiers was explored, shedding light on differences in distribution characteristics. The study primarily aimed to introduce zero-shot learning prompts tailored for ChatGPT to generate code snippets suitable for cost-sensitive learning. Logistic regression and gradient boosting classifiers were employed without extensive refinement, highlighting the emphasis on code generation over model optimization. Comparative analyses were conducted between models generated by ChatGPT prompts and human-written code from Kaggle champions, revealing competitive performance levels in terms of precision, recall, and F1-score.

Specific experiments were conducted to showcase the utility of zero-shot learning prompts in generating code snippets for various aspects of model development, including feature engineering, handling imbalanced classes, regularization, hyperparameter tuning, ensemble methods, cross-validation, feature selection, and optimizing decision thresholds. The results demonstrated varying degrees of model performance across different prompts, underscoring the versatility of ChatGPT in generating tailored code snippets to meet specific requirements.

The imbalance of classes in the water potability dataset was also addressed, with potential approaches discussed to mitigate this issue, such as resampling techniques and alternative evaluation metrics. The findings highlight the effectiveness of zero-shot learning prompts in tackling real-world challenges and streamlining model development for cost-sensitive learning tasks.

6. Conclusions

The study showcases the effectiveness of zero-shot learning prompts tailored for ChatGPT in generating code snippets suitable for cost-sensitive learning tasks. By employing these prompts, the study demonstrated the versatility of ChatGPT in automating various aspects of machine learning model development. Comparative analyses between models generated by ChatGPT prompts and human-written code from Kaggle champions revealed competitive performance in terms of precision, recall, and F1-score. This suggests that ChatGPT-generated code snippets can achieve comparable results to manually written code.

Specific experiments conducted to explore different aspects of model development, such as feature engineering, handling imbalanced classes, and hyperparameter tuning, underscored the versatility of ChatGPT in generating tailored code snippets to address specific requirements. This highlights ChatGPT's potential in streamlining the model development process.

The study addressed the imbalance of classes in the water potability dataset and discussed potential approaches, such as resampling techniques and alternative evaluation metrics, to mitigate this issue. This demonstrates the effectiveness of zero-shot learning prompts in tackling real-world challenges and enhancing model development for cost-sensitive learning tasks.

In conclusion, the study contributes to advancements in the field of machine learning by showcasing the potential of ChatGPT in automating aspects of model development. By facilitating code generation for various machine learning tasks, ChatGPT can accelerate the pace of research and innovation in the field, ultimately leading to more efficient and effective machine learning models. The Supplementary Materials provide the codes used in this study (https://zenodo.org/records/10884304) (accessed on 27 March 2024).

Supplementary Materials: The code for the experiments conducted in this study is available at Zenodo repository (https://zenodo.org/records/10884304).

Author Contributions: Conceptualization, M.S. (Malusi Sibiya) and K.P.; methodology, M.S. (Malusi Sibiya); software, K.P.; validation, M.S. (Mbuyu Sumbwanyambe) and M.S. (Malusi Sibiya); formal analysis, K.P.; investigation, M.S. (Malusi Sibiya); resources, K.P.; data curation, K.P.; writing—original draft preparation, K.P.; writing—review and editing, M.S. (Malusi Sibiya); visualization, M.S. (Mbuyu Sumbwayambe); supervision, M.S. (Mbuyu Sumbwanyambe). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This research was conducted under the ethical research certificate issued to the authors by the University of South Africa.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Data are available at Zenodo repository (https://zenodo.org/records/10884304) (accessed on 27 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The data and code used in this research was submitted with this paper. The code and dataset are published in Zenodo repository (https://zenodo.org/records/10884304).

References

1. Yaroshenko, I.; Kirsanov, D.; Marjanovic, M.; Lieberzeit, P.A.; Korostynska, O.; Mason, A.; Frau, I.; Legin, A. Real-time water quality monitoring with chemical sensors. *Sensors* **2020**, *20*, 3432. [CrossRef]
2. Ahuja, L.R.; Ma, Q.L.; Rojas, K.W.; Boesten, J.J.; Farahani, H.J. A field test of root zone water quality model—Pesticide and bromide behavior. *Pestic. Sci.* **1996**, *48*, 101–108. [CrossRef]
3. Shrestha, S.; Kazama, F. Assessment of surface water quality using multivariate statistical techniques: A case study of the Fuji river basin, Japan. *Environ. Model. Softw.* **2007**, *22*, 464–475. [CrossRef]
4. Singh, K.P.; Malik, A.; Sinha, S. Water quality assessment and apportionment of pollution sources of Gomti river (India) using multivariate statistical techniques—A case study. *Anal. Chim. Acta* **2005**, *538*, 355–374. [CrossRef]
5. Smeti, E.M.; Thanasoulias, N.C.; Kousouris, L.P.; Tzoumerkas, P.C. An approach for the application of statistical process control techniques for quality improvement of treated water. *Desalination* **2007**, *213*, 273–281. [CrossRef]
6. Yang, Y.J.; Haught, R.C.; Goodrich, J.A. Real-time contaminant detection and classification in a drinking water pipe using conventional water quality sensors: Techniques and experimental results. *J. Environ. Manag.* **2009**, *90*, 2494–2506. [CrossRef]
7. Chaves, P.; Tsukatani, T.; Kojiri, T. Operation of storage reservoir for water quality by using optimization and artificial intelligence techniques. *Math. Comput. Simul.* **2004**, *67*, 419–432. [CrossRef]
8. Gevrey, M.; Rimet, F.; Park, Y.S.; Giraudel, J.L.; Ector, L.; Lek, S. Water quality assessment using diatom assemblages and advanced modelling techniques. *Freshw. Biol.* **2004**, *49*, 208–220. [CrossRef]
9. Letcher, R.A.; Jakeman, A.J.; Calfas, M.; Linforth, S.; Baginska, B.; Lawrence, I. A comparison of catchment water quality models and direct estimation techniques. *Environ. Model. Softw.* **2002**, *17*, 77–85. [CrossRef]
10. Hedger, R.D.; Atkinson, P.M.; Malthus, T.J. Optimizing sampling strategies for estimating mean water quality in lakes using geostatistical techniques with remote sensing. *Lakes Reserv. Res. Manag.* **2001**, *6*, 279–288. [CrossRef]
11. Allinson, M.; Shiraishi, F.; Kamata, R.; Kageyama, S.; Nakajima, D.; Goto, S.; Allinson, G. A pilot study of the water quality of the Yarra River, Victoria, Australia, using in vitro techniques. *Bull. Environ. Contam. Toxicol.* **2011**, *87*, 591–596. [CrossRef]
12. Nyende-Byakika, S.; Ndambuki, J.M.; Onyango, M.S.; Morake, L. Potability analysis of raw water from Bospoort dam, South Africa. *Water Pract. Technol.* **2016**, *11*, 634–643. [CrossRef]

13. Pehlivan, R.; Emre, H. Potability and hydrogeochemisty of the Sarma Stream water, Duzce, Turkey. *Water Resour.* **2017**, *44*, 315–330. [CrossRef]
14. Achio, S.; Kutsanedzie, F.; Ameko, E. Comparative analysis on the effectiveness of various filtration methods on the potability of water. *Water Qual. Res. J. Can.* **2016**, *51*, 42–46. [CrossRef]
15. Elizabeth, K.M.; Rajpramikh, K.E. Potability of Water among the Tribals of Vizianagaram Sub-plan Area, Andhra Pradesh: Microbiological and Physico-Chemical Analysis. *Anthropologist* **2000**, *2*, 181–184. [CrossRef]
16. Spackman, C.; Burlingame, G.A. Sensory politics: The tug-of-war between potability and palatability in municipal water production. *Soc. Stud. Sci.* **2018**, *48*, 350–371. [CrossRef]
17. Mahajan, M.; Bhardwaj, K. Potability analysis of drinking water in various regions of Ludhiana District, Punjab, India. *Int. Res. J. Pharm.* **2017**, *8*, 87–90. [CrossRef]
18. Lvova, L.; Di Natale, C.; Paolesse, R. Chemical sensors for water potability assessment. *Bottled Packag. Water* **2019**, *4*, 177–208.
19. Abanyie, S.K.; Boateng, A.; Ampofo, S. Investigating the potability of water from dug wells: A case study of the Bolgatanga Township, Ghana. *Afr. J. Environ. Sci. Technol.* **2016**, *10*, 307–315.
20. Opafola, O.T.; Oladepo, K.T.; Ajibade, F.O.; David, A.O. Potability assessment of packaged sachet water sold within a tertiary institution in southwestern Nigeria. *J. King Saud Univ. Sci.* **2020**, *32*, 1999–2004. [CrossRef]
21. Chauhan, J.S.; Badwal, T.; Badola, N. Assessment of potability of spring water and its health implication in a hilly village of Uttarakhand, India. *Appl. Water Sci.* **2020**, *10*, 201. [CrossRef]
22. Arulnangai, R.; Sihabudeen, M.M.; Vivekanand, P.A.; Kamaraj, P. Influence of physico chemical parameters on potability of ground water in ariyalur area of Tamil Nadu, India. *Mater. Today Proc.* **2021**, *36*, 923–928. [CrossRef]
23. An, H.; Li, X.; Huang, Y.; Wang, W.; Wu, Y.; Liu, L.; Ling, W.; Li, W.; Zhao, H.; Lu, D.; et al. A new ChatGPT-empowered, easy-to-use machine learning paradigm for environmental science. *Eco-Environ. Health* **2024**, *3*, 131–136. [CrossRef]
24. Barberio, A. *Large Language Models in Data Preparation: Opportunities and Challenges*; Scuola di Ingegneria Industriale e dell'Informazione: Milan, Italy, 2022.
25. Hassani, H.; Silva, E.S. The role of ChatGPT in data science: How ai-assisted conversational interfaces are revolutionizing the field. *Big Data Cogn. Comput.* **2023**, *7*, 62. [CrossRef]
26. Roumeliotis, K.I.; Tselikas, N.D. ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet* **2023**, *15*, 192. [CrossRef]
27. Mujahid, M.; Rustam, F.; Shafique, R.; Chunduri, V.; Villar, M.G.; Ballester, J.B.; Diez, I.d.l.T.; Ashraf, I. Analyzing sentiments regarding ChatGPT using novel BERT: A machine learning approach. *Information* **2023**, *14*, 474. [CrossRef]
28. Lubiana, T. Ten Quick Tips for Harnessing the Power of ChatGPT. GPT-4 in Computational Biology. *PLOS Comput. Biol.* **2023**, *19*, e1011319. [CrossRef]
29. OpenAI. ChatGPT [3.5]. 2024. Available online: https://chat.openai.com/c/53c0468f-e40d-439c-a90b-e224d64afdc8 (accessed on 2 February 2024).
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need, Carlifornia. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.