Survey of Machine Learning Methods on Water Potability Data

Kyler Sood, Abha Gejji, Nirmal Raj, Wangyang Ge
University of Minnesota, Twin Cities

CSCI 5525 Project — Spring 2022

1 Introduction

For our project, we surveyed various machine learning methods to see how they perform on a water potability dataset. We compared classical and deep learning methods and evaluated their performance.

1.1 Dataset

The data is contained in a .csv file and includes nine continuous attributes which measure water quality along with a binary variable, potability, which indicates whether the water is safe for human consumption. There are a significant number of missing values in the pH, Sulfate, and Trihalomethanes columns, which means that the data needs to be preprocessed. The dataset is found on Kaggle [1].

1.2 Methods for preprocessing

1.2.1 Handling Missing Values and Normalization

The missing values were interpolated using a linear regression model. The observations of the dataset without missing feature values were used to train the linear regression model. This model was then used to predict the values of missing rows. Once the missing values were handled, the features were checked for correlations between them to remove highly correlated features. Since all the features had low correlation with each other, we did not have to remove any. Finally, the feature values were translated and scaled to have zero mean and unit variance.

$$x_{scaled} = \frac{x - \mu_{feature}}{\sigma_{feature}}$$

In order to convert the continuous feature values to discrete feature values (for decision trees), the feature values were split into two using unique values of the feature and the accuracy of the split was plotted against the split values for choosing the value that gives the best accuracy.

1.2.2 Principal Component Analysis (PCA)

Principal component analysis(PCA) is a common technique used for dimensionality reduction. It takes into account the variance of the features without considering the effects the features have on the target value. Following PCA the explained variance ratio is the percent of variance explained by each of the principal components. The explained variance ratio for the water potability dataset is shown in Table 1. It shows that the variance of each feature is similar and elimination based on PCA will therefore be ineffective.

1.2.3 SMOTE: Synthetic Minority Over-sampling Technique

SMOTE is a method for sampling an unbalanced dataset. The data is unbalanced where 40% of data points are potable water and 60% of the data points are non-potable water. By oversampling the minority class we hope to achieve better performance. SMOTE only modifies the training data

feature	explained variance ratio
ph	0.13876337
Hardness	0.12886566
Solids	0.11734319
Chloramines	0.1137457
Sulfate	0.11043887
Conductivity	0.10763871
Organic carbon	0.10551548
Trihalomethanes	0.09613842
Turbidity	0.0815506

Table 1: PCA feature variance

which is then used for the rest of the classification algorithms. It is important that the test dataset is untouched throughout the whole process.

Chawla et al. discuss an approach on how to augment data to generate new data points for the minority class [2]. This new data is the interpolation of data points in the feature using the k-nearest neighbors algorithm. A synthetic data-point is crested in an arbitrarily selected region of the feature space. The performance of SMOTE can be limited since it relies heavily on linear interpolation while the dataset is not necessarily linearly equivalent. Synthetic samples are produced without considering the majority class, possibly resulting in ambiguously-labeled samples, as the regions of both classes may not be linearly separable. Another aspect is that the water potability dataset is slightly imbalanced.

Sample weights in the loss function and weighted sampling while loading the data are methods for handling the class imbalance which can be further explored.

1.3 Literature Survey

In our research, we came across various prior works on applying machine learning to water quality research. Two of these papers used support vector machines to model the relationship between water variables and water quality. [9] [10] Another one used Long Short-Term Memory Networks [11], which motivated us to try something similar in using gated recurrent units in our neural network.

2 Methods and results

2.1 Autoencoder

Unlike for image data, where pretraining on unrelated image data has shown to improve results drastically, transfer learning is uncommon for tabular datasets unless the datasets are relatively similar. So, introducing neural network architecture proves to be challenging.

Neural networks used to encode data for feature selection will require higher dimensionality. When the data is tabular and with only 3072 instances and 9 features neural networks were determined to overfit and weakly generalize.

Even a simple 2-layer model with 9 input nodes, 5 hidden nodes, and 1 output node will have 50 weights to be fitted. Our results indicate this model will always fail to generalize given the limited data samples and features available.

2.2 Linear Discriminant Analysis

Using the Fischer formula, we can reduce the dimensionality of the data. For binary classification, the resulting dimension cannot be higher than 1. After projecting the data on a 1-dimensional axis, the classifier divides this 1-dimensional space into potable and not potable class regions.

2.3 Decision Trees

Decision trees are a machine learning algorithm often used for classification tasks. They resemble a tree structure and employ simple decision-making to classify features into targets. Each node of a

tree corresponds to a feature. Depending on the value of the feature for the current data point, the data point is assigned to one of the child nodes, either another decision or a target value (leaf node). Decision trees are often prone to over-fitting, hence they are deployed with other methods such as boosting, bagging, and pruning.

2.4 Random Forest

Random forest is another common classical classification algorithm. It is less prone to overfitting compared to decision trees. It is a technique where multiple decision trees compose an ensemble to produce a prediction. It requires more computational power when compared to decision trees and is more difficult to interpret, but have been shown to be less noisy [3]. CV grid search was used to find the optimum parameters of random forest. The random forest method obtained the best results out of our tested models.

2.5 KNN

K-nearest neighbor classification (KNN) is another straightforward algorithm where the data points surrounding a test data point influence the prediction. This algorithm has only 1 parameter, k, which is the number of neighbors to be considered, that is, how many data points vote on the prediction. The value of k will dictate the complexity of the model. If k is selected to be too low, the model becomes too specific and overfits. If k is set to be too big, the model becomes too generalized and has poor performance.

2.6 ADABOOST

AdaBoost (Adaptive Boosting) is an algorithm for building a strong learner from a collection of weak learners. They are often used along with other machine learning algorithms in order to improve the performance of the model. AdaBoost makes use of progressive learning where each learner is trained to accurately learn the data points that were wrongly classified by the previous learner. Since each learner is trained to classify the wrong classifications of the previous learner, an ensemble of these weak classifiers provides a very good performance on the data.

2.7 XGBOOST

XGBoost is an optimized distributed gradient boosting library that provides parallel tree boosting. Chen et al. have described a scalable end-to-end system which was state of the art at the time of publication [4]. It is also called an Extreme Gradient boost and is highly efficient and can handle an enormous amount of data. It is a very widespread algorithm used in Kaggle competitions for its multitudes of applications. Gradient boosting is a technique where new models are created based on the errors of previous models. The ensembling of initial models and models built to fine-tune the errors in those models has shown improved performance. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Method	f1-score for 0	f1-score for 1	Accuracy
Decision Tree	0.76	0.16	0.62
Decision Tree + AdaBoost	0.76	0.18	0.62
SMOTE + Linear Discriminant Classifier	0.59	0.44	0.53
SMOTE + Gaussian Naive Bayes	0.66	0.46	0.59
SMOTE + K Nearest Neighbours Classifier	0.67	0.50	0.60
SMOTE + XGBoost	0.69	0.53	0.62
SMOTE + Random Forest	0.74	0.53	0.66

Table 2: Results for Various Classical Techniques

2.8 Artificial Neural Network

Artificial neural networks (ANN) provide a framework for classification and prediction as a broad class of models. ANNs consist of layers of interconnected neurons which perform transformations and operations with optimizable parameters. Convex optimization algorithms such as gradient descent and adaptive moment estimation enable robust performance for ANNs. We employ a nonlinear neural network with layers of: Linear, Softmax, Linear, LeakyReLU, Linear and use Xavier initialization [5]. For optimization, we proceed with ADAM (adaptive moment estimation), a variant of stochastic gradient descent.

2.9 Elman Recurrent Neural Network and Gated Recurrent Units

Recurrent neural networks (RNN) are a class of ANNs in which nodes are connected by edges to form a graph. In applications, RNNs can be used to model temporal data [6]. The graph nature enables for relationships between nodes to produce decisions, useful for classification problems. We continue to use ADAM for convex optimization. RNNs contain learnable hidden states which sequentially depend on each other, which perform transformations and serve as intermediates in the graph between the input and output. Gated recurrent units (GRU) are layers for RNNs which can be used as a variant of the long short-term memory (LSTM) units. GRUs act as neural networks which help prevent vanishing gradients (a potential problem with RNNs) with the final state acting as activation function for the next layer. We optimize the networks using ADAM.
We implement two networks, one using an RNN and one using GRUs, with the following structures:
1) RNN with 4 hidden layers, Leaky ReLU, Linear. The RNN uses tanh for its nonlinearity and we use Xavier initialization for the linear layer.
2) GRU cell, GRU cell, ReLU, Linear, Linear using the Xavier initialization for the linear layers.

2.10 Ensemble Networks using Bagging

Ensembles are a method for combining several machine learning models to produce a more robust and accurate model, alongside advantages with reduced amounts of data. We employ bagging, or bootstrap aggregating, where the predictions of each model are normalized and averaged to produce one aggregate prediction for the ensemble model. Ensemble networks are implied to have superior performance based on the bias-variance decomposition and yield greater diversity in classification problems [7].

Our ensemble model consists of each of the three previous networks (ANN, RNN, GRU RNN) weighted equally and normalized with the Euclidean 2-norm, each trained on equally-sized subsets of the water potability data and optimized using ADAM.

Method	f1-score for 0	f1-score for 1	Accuracy
Nonlinear ANN	0.67	0.81	0.68
Ordinary RNN	0.67	0.78	0.63
GRU RNN	0.67	0.80	0.66
Ensemble	0.67	0.83	0.70

Table 3: Results for Neural Networks

3 Conclusion

Here we surveyed various classical and deep learning approaches to binary classification for a water potability dataset. We first preprocessed the data by linearly interpolating missing values and normalizing, then trained several classifiers on the dataset. We observed the best f1 scores and accuracy among classical methods using SMOTE + Random Forest, which obtained 66% accuracy. Among deep learning methods, the ensemble network using bagging produced a tangible increase in accuracy and f1 scores, with 70% cross-validation accuracy. Based on our broad survey of classical and deep learning techniques, further work and accuracy improvements would require a larger dataset with more features and observations.

We further considered the KNN algorithm with and without employing SMOTE for processing the data.

Method	f1-score for 0	f1-score for 1	Accuracy
K Nearest Neighbours Classifier with SMOTE	0.67	0.50	0.60
K Nearest Neighbours Classifier without SMOTE	0.72	0.43	0.62

We can see that the two results are fairly close, but applying SMOTE gave a marginal improvement to the difference in the two f1 scores. This may be because the 60-40 split between the 1 and 0 data points is not skewed enough for SMOTE to have a significant impact. In empirical applications, the imbalance may be very large which may see more improvement from SMOTE [8].

References

[1] Kadiwal, Aditya. "Water Quality: Drinking Water Potability." Kaggle, 2021. https://www.kaggle.com/datasets/adityakadiwal/water-potability/metadata

[2] Chawla, N. V., et al. "SMOTE: Synthetic Minority Over-Sampling Technique." The Journal of Artificial Intelligence Research, vol. 16, AI Access Foundation, 2002, pp. 321–57, doi: 10.1613/jair.953.

[3] Breiman, Leo. "Random Forests." Machine Learning, vol. 45, no. 1, Springer, 2001, pp. 5–32, doi: 10.1023/A:1010933404324.

[4] Chen, Tianqi, and Carlos Guestrin. "XGBoost." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–94, doi: 10.1145/2939672.2939785.

[5] Glorot, Xavier and Yoshua Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks." PMLR, vol. 9, 2010, pp. 249-256.

[6] Tealab, Ahmed. "Time Series Forecasting using Artificial Neural Networks Methodologies: A Systematic Review." Future Computing and Informatics Journal, vol. 3, 2018, pp. 334-40.

[7] Ganaie, M.A., Hu Minghui, and A. K. Malik. "Ensemble Deep Learning: A Review." arXiv, 2021, doi: 10.48550/arXiv.2104.02395.

[8] Oinar, C. "Introduction to Synthetic Minority Over-sampling Technique and its Implementation from Scratch." Towards Data Science, 2021, https://towardsdatascience.com/introduction-to-synthetic-minority-over-sampling-technique-and-its-implementation-from-scratch-77593647c10d

[9] K. Blix, "Machine Learning Classification, Feature Ranking and Regression for Water Quality Parameters Retrieval in Various Optical Water Types from Hyper-Spectral Observations," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 5608-5611, doi: 10.1109/IGARSS39084.2020.9324717.

[10] X. Wang, L. Ma and X. Wang, "Apply semi-supervised support vector regression for remote sensing water quality retrieving," 2010 IEEE International Geoscience and Remote Sensing Symposium, 2010, pp. 2757-2760, doi: 10.1109/IGARSS.2010.5653832.

[11] J. P. Nair and M. S. Vijaya, "Predictive Models for River Water Quality using Machine Learning and Big Data Techniques - A Survey," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 1747-1753, doi: 10.1109/ICAIS50930.2021.9395832.