Computer Science Department

Final Project: Predicting Water Drinkability

Course: Introduction to AI (CS156)

Course Instructor: Yulia Newton, Ph.D.

Rohan Athalye

Introduction/Background

According to UNICEF and the World Health Organization (WHO), approximately 1 in 3 individuals lack access to safe drinking water. Having access to safe drinking water is a fundamental human right and essential to human health. Thus, the problem I will be solving is predicting whether water is safe to drink based on various water quality metrics. To me, this is a particularly interesting problem as I am not too familiar with the factors that go into assessing water quality, so it would be helpful to know the appropriate amounts for each factor in order to have safe drinking water. According to the Centers for Disease Control and Prevention (CDC), drinking contaminated water can cause adverse health effects such as diarrhea, cholera, and typhoid fever, all of which may even lead to death. Thus, this is an extremely important problem that needs to be solved as soon as possible.

Given that access to safe drinking water is a pressing worldwide issue, many have attempted to come up with a solution for it. 1 notable solution is Clean Water AI, a device that detects harmful bacteria and particles in water and allows users to see their drinking water at a microscopic level with real-time detection. Furthermore, cities can install these devices across various water sources to monitor their water quality in real time. Behind the scenes, Clean Water AI uses a deep learning neural network model to recognize harmful bacteria and dangerous particles in water. I agree with their approach and trust that sufficient training, testing, and parameter/hyperparameter tuning of the model has been performed to provide the most accurate results.

Dataset description

The dataset I used to predict whether water is safe to drink is the "Water Quality" dataset from Kaggle datasets. This dataset contains various water quality metrics for 3276 different water bodies.

I used 9 numeric independent variables, each representing a water quality metric, to train my model. The independent variables are: pH (pH value of the water); hardness (capacity of water to precipitate soap in mg/L); solids (total dissolved solids in ppm); chloramines (amount of chloramines in ppm); sulfate (amount of sulfates dissolved in mg/L); conductivity (electrical conductivity of water in µS/cm); organic carbon (amount of organic carbon in ppm); trihalomethanes (amount of trihalomethanes in µg/L); and turbidity (measure of light emitting property of water in Nephelometric Turbidity Units).

The dependent variable is a numeric variable that returns 1 if the water is safe for human consumption and 0 if the water is not safe for human consumption. The dependent variable is a value that my model aims to predict and suggests that the problem is a classification problem with 2 classes: 1 and 0.

I split my dataset into training and test sets at an 80/20 ratio. Since I do not have a set aside validation dataset, I used a 5-fold cross-validation approach. I also stratified my split to ensure that each fold has similar class distribution/representation to that of the overall dataset.

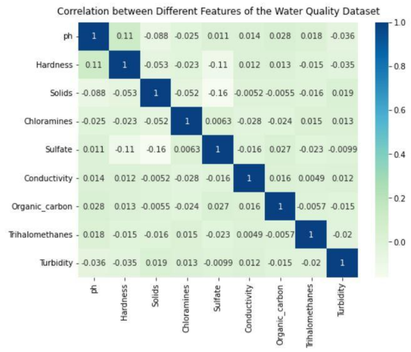

Figure 1: *Correlation Matrix of the Water Quality dataset*

I plotted a correlation matrix of the dataset above to see the correlation between different features of the dataset. Surprisingly, there weren't any strong positive or negative correlations between any of the features.

Methodology

Algorithm

For this project, I used a Support Vector Machine (SVM). An SVM works by finding a hyperplane in an N-dimensional space that distinctly classifies some data points. The goal is to find that hyperplane with the maximum margin, meaning the maximum distance between the data points of 2 classes. Doing so would make it easier to classify future data points.

I decided to use an SVM because SVMs perform well on high dimensional data. Since my dataset has 9 independent variables, meaning it is in 9 dimensions, I thought an SVM would be a good choice for this project. In addition, if the data is not linearly separable, SVMs can use a kernel function to transform the data into a new space in which it is separable.

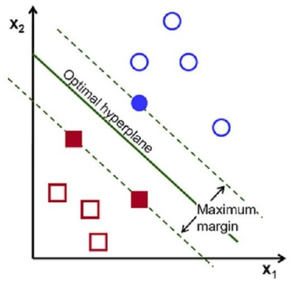

Figure 2: *Example of an SVM*

Application to the project

I rescaled the training data using StandardScaler which standardizes every independent variable in the dataset. I performed this rescaling because SVM models are very sensitive to the scale of the data, so they often perform best when different independent variables are rescaled to the same space. I then fitted my SVM model to the training data using a Support Vector Classifier (SVC) with the default RBF kernel function.

Analysis and Results

Validation/cross-validation results

After performing 5-fold cross-validation, my SVM model produced an average cross-validation accuracy of ~67.47%. The individual cross-validation accuracies were ~71.43%, ~66.15%, ~66.46%, ~65.73%, and 67.6% respectively.

Training results

My trained SVM model produced a ~75.50% accuracy on the training dataset.

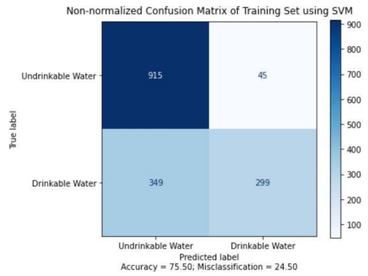

Figure 3*: Confusion Matrix of Training Dataset*

According to the confusion matrix above, we can see there were several misclassifications made on the training set: water bodies that have drinkable water were classified as having undrinkable water and vice versa.

Test results

My trained SVM model produced a ~69.48% accuracy on the test dataset.

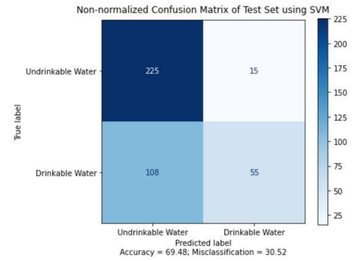

Figure 4*: Confusion Matrix of Test Dataset*

According to the confusion matrix above, we can see there were several misclassifications made on the test set: water bodies that have drinkable water were classified as having undrinkable water and vice versa.

Summary/Conclusions

Overall, my SVM model did not perform well, producing a low average cross-validation accuracy, training accuracy, and test accuracy. To improve, I could have performed hyperparameter tuning using GridSearchCV or RandomizedSearchCV to find the best combination of hyperparameters. For example, I could have experimented with different kernel functions, kernel coefficients, regularization values, etc.

Given more time, I would have liked to perform in-depth feature selection and/or feature extraction with the goal of building a simpler model that produces improved accuracies.

Key References

- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://www.kaggle.com/datasets/adityakadiwal/water-potability
- https://www.who.int/news/item/18-06-2019-1-in-3-people-globally-do-not-have-access-to-safe-drinking-water-unicef-who
- https://www.cdc.gov/healthywater/global/assessing.html
- https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/
- https://medium.datadriveninvestor.com/support-vector-machines-important-questions-a47224692495

Acknowledgements

I would like to thank Dr. Newton for introducing so many different machine learning models and providing examples for each of them. This greatly helped shape the way I approached this project, and I hope to apply the skills I have gained here in the future.

www.PosterPresentations.com