Predicting Water Quality for Agricultural Use - Introduction to Machine Learning Analysis

Fabrice Faustin
CSCI 4371

1 Introduction

Computer technology has come a long way ever since its inception. From merely being able to perform basic calculations to now being able to predict the coming week's weather, our rapid and explosive advancements in the field just cannot seem to come to a halt. And among the various innovations that we have brought to the limelight, machine learning is perhaps the most important element of our modern and current technological society.

Machine Learning involves having computer systems learn and recognize patterns of data to be able to produce desired output without developers having to tell it exactly what to do. Essentially, it is the concept of having computer systems provide results based on experience. For instance, a machine learning system may predict the weather forecast based on the patterns that it has recognized from an exhaustive list of past weather information. Likewise, it can involve a computer software being able to distinguish cats from dogs based on the extensive library of cats and dogs images it was trained on.

Many techniques are employed to guide how we use our machine learning algorithms, among which two popular ones stand out: supervised and unsupervised learning. Supervised learning is a more guided approach to learning for the machine, and involves having the computer be trained on a dataset, and then testing what it has learned with another dataset that it has not seen to determine if its learning strategy was effective in producing a desired output. We can then continuously train the computer if that is not the case until the output is in correspondence with the unseen data. This approach is akin to giving a quiz as a teacher to a student. Whenever the student does poorly, we go back to teach them with a different approach and test said student again on the quiz. Once the student gets it right, we can be confident that they have learned the right way and should be able to provide the same results for other similar types of quizzes. Hence the name supervised learning.

On the other hand, unsupervised learning has the computer understand and learn anything about provided data on its own and give feedback on what it has learned. This approach is used to

perhaps identify underlying trends or patterns of data that were hidden to the human eyes, and to be able to regroup and reconsider said data thanks to that newfound vision.

We usually employ supervised or unsupervised learning to solve two kinds of problems: regression or classification. Regression tasks involve predicting continuous values, such as house prices or stock prices, based on input features. In contrast, classification tasks entail assigning labels or categories to new observations, such as identifying objects in images, or predicting if a patient has diabetes or not for instance.

In this report, I will provide feedback on my specific application of supervised learning on a classification task I had worked on. I will explore the different models I employed as the classification solutions for my particular dataset along with the different tuning I administered to each of them to control the output. I will also compare and contrast the different models I have used, and analyze which one was better for my particular classification task and why. Finally, I will provide insight on what this project of mine has taught me regarding machine learning as a whole and how I can use this experience moving forward.

2 Data

The dataset that I used for this project can be found on the Kraggle website. The person who shared this dataset there actually obtained from the Indian t for the years 2018, 2019, and 2020. This data was gathered in order to help assess the quality of water based on several factors, which would then help make more informed agricultural work decisions in the region. Each observation in the different datasets are comprised of 26 dimensions, among which the 24 features along with their data types are listed below:

- A serial number or sno, identifying each unique observation. (Type : Numeric)
- The district in which the water was gathered. (Type : Categorical Text Data).
- Mandal (Type : Categorical Text Data),
- Village (Type : Categorical Text Data),
- Latitude (Type : Numeric),
- Longitude (Type : Numeric),
- ground water level or gwl (Type : Numeric),
- Season (Type : Categorical Text Data),
- pH (Type : Numeric),
- Electrical conductivity or E.C (Type : Numeric),
- Total dissolved solids or T.D.S (Type : Numeric),
- Various chemical compositions (CO3, HCO3, Cl, F, NO3, SO4, Na, K, Ca, Mg) (Type : Numeric),
- Total hardness (Type : Numeric),

- residual sodium carbonate or RSC (Type : Numeric),
- Sodium adsorption ratio or SAR (Type : Numeric).

The target variables for this dataset were titled 'Classification' and 'Classification1', both categorically defining the quality of groundwater samples.

The 'Classification' target variable delineates nine distinct classes from C1S1 through C4S4. Each class signifies a unique combination of salinity and sodium levels as well as crop compatibility. For instance, C1S1 classifies groundwater with low salinity and sodium levels which is ideal for irrigation across various soil types. C4S4 defines groundwater with very high salinity and sodium levels, which is therefore unsuitable for irrigation in most scenarios, necessitating specialized soil management techniques.

In the examined 2018 dataset which has 379 observations, there is a dominant proportion of 63% of observations classified as C3S1. 24% of observations fall under the C2S1 classification, which indicates medium salinity and low sodium levels, or in other words relatively favorable conditions for irrigation across diverse soil types. The remaining observations account for the remaining 13%.

3 Methods

- **Data Preprocessing:**

The features from the original dataset in Kraggle was divided into two primary data types: categorical (text and numeric) and numeric. Categorical data represent variables that can take on a limited, predefined set of values, such as district names or seasons. On the other hand, numeric data consist of continuous numerical values, such as pH levels or total dissolved solids (TDS). With the objective of simplifying the analysis and focusing on the most informative features, I made the decision to retain only the numeric features while excluding categorical text data. The 'Classification' target variable, denoting groundwater quality classes, was also chosen as the sole target variable due to its inherent relevance to the study's objectives. Additionally, the RSC feature was omitted to streamline the design process. I focused on the 2018 dataset. In this particular dataset, some values for some features were null. I fixed this by changing these null values to zero to homogenize everything.

- **Train - Test Split**

In order to effectively utilize machine learning models in a supervised learning setting, we first need to train the model on a dataset, which essentially means have the model analyze the patterns of said dataset and determine a prediction strategy for similar looking data based on its algorithm. Afterwards, the goal is to then test this prediction strategy on a new set of similar data

that the model has never seen and verify if our model will correctly predict the classes of the features of the new dataset. In that respect, it is beneficial to split your dataset into training and testing subsets. Generally, we want to have a reasonably large training dataset so that the model has more data to work with and analyze patterns from. With that logic, I decided to split my dataset after the preprocessing into 80% training subset and 20% testing subset.

- **Classification Models:**

Two distinct classification models were implemented for my predictive analysis: K-nearest neighbors (KNN), and decision tree. K-Nearest Neighbors is a learning model based on the principle of similarity. It classifies a data point by examining the class of its nearest neighbors. It is particularly useful in classifying multiple points into classes.

Decision trees are hierarchical structures that recursively partition the feature space based on feature values. At each node of the tree, a decision is made based on a feature's value, splitting the data into subsets. The process continues recursively until a stopping criterion is met, such as reaching a maximum tree depth or purity criteria for leaf nodes.

- **Hyperparameter Tuning:**

Hyperparameters are parameters that are not directly learned from the data during training but are set before training begins and can influence the behavior of the learning algorithm of a model and thus influence the model's performance. In the context of the experiments conducted, here are the hyperparameters that were tuned:

Decision Tree: Max Tree Depth

- The max tree depth determines the maximum number of levels in the decision tree.
- Increasing the max tree depth allows the model to capture more intricate patterns in the data but may lead to overfitting if set too high.

K-nearest Neighbors: Number of Neighbors

- The number of neighbors (K) in KNN determines the number of nearest neighbors to consider when making predictions.
- A smaller K value results in a more flexible model that may be sensitive to noise, while a larger K value provides a smoother decision boundary but may overlook local patterns.

- **Cross Validation:**

I also employed 5-fold cross-validation on each iteration of each experiment to ensure robustness in my experiment. Cross-validation is essentially to test one's model without having to use our testing portion of the dataset. The way it works is that it splits the training dataset into k different equal subsets or folds. Then, each fold is chosen as the testing subdataset of the training data and will be used to assess the performance of the model based on its training on the rest of the training data. After testing is complete for one particular fold, the process repeats until all folds have been used as a test subdataset and the remaining training dataset has been evaluated for performance against it.

Cross validation provides a helpful way to make the most of a limited dataset. By further dividing training data into test and training data, we get to keep more unseen data for our model to be tested on after the cross-validation phase, which is beneficial because we always need a good amount to test our model against.

As mentioned previously, I used 5-fold cross validation in this project, which means that each tuned model will be divided into 5 subsets that will all in turn be used as a subtest dataset against the rest of the training data for the model. The cross-validation score was also computed for each fold, which helped give insight into the performance of our particular model.

- **Performance Metrics:**

To measure the performance of my experiments and of my models as a whole, I decided to compute and include the mean accuracy for each model for every tuning of their specific hyperparameters. The mean accuracy was computed by averaging the 5 cross validation scores resulting from the 5-fold cross validation I used for each instance of my models.

In the context of machine learning model performance, accuracy refers to the ratio of the amount of predictions that the model got right over the total number of predictions made by the model. Accuracy is a very standard performance metric used in machine learning to assess simple performance evaluation. However, its simplistic nature also means that it may not communicate enough detailed information about the performance we are evaluating.

My second performance metric was the precision score of the instances of the two models. In machine learning, precision is more specific than accuracy and is especially useful when it comes to classification problems because it focuses on the accuracy of the correct prediction of the model when it predicted that targets were of a particular class.

4 Results

Comparing the different instances of the decision tree model in terms of accuracy, where I tuned the max tree depth for each instance, my result for all instances were as follows: At max tree depth set to 3, the accuracy was evaluated at 0.90972 or 91%. The somewhat same accuracy was recorded when the max tree depth was set to 5 (0.909774 or a slightly higher 91%). Finally, when the max tree depth was 7, the accuracy dropped to 0.88634 or 88.6%) Overall, my decision tree model recorded the highest mean accuracy when the max tree depth was set to 5.

When it comes to the precision, the decision model with the maximum tree depth number set to 5 was the highest with a score of 0.91305 or 91.4%. At max depth equal 3, the score was 0.88239 or 88.2%. At max depth set to 7, it was 0.89971 or 90%. In this comparison, the model performed best when the max depth was set to 5.

Overall, the decision tree model performed better in terms of both accuracy and precision when the max tree depth was set to 5.

Comparing the different instances of the K nearest neighbor model in terms of accuracy, where I tuned the k hyperparameter for each instance, my result for all instances were as follows: At k set to 5, the accuracy was evaluated at 0.899774 or 90%. The accuracy recorded when the k hyperparameter was set to 10 was 0.9064407 or 91%. Finally, when the k hyperparameter was 15, the accuracy dropped to 0.8963841 or 89.6%) Overall, my k nearest neighbor model recorded the highest mean accuracy when k was set to 10.

Moving on to precision, the k nearest neighbor model with the number of neighbors set to 5 was the highest with a score of 0.851825 or 85%. At k equal 10, the score was 0.8351587 or 84%. At k set to 15, it was 0.8122351 or 81%. As such, in this comparison, the model performed best when the number of neighbors k was set to 5.

Overall, the K nearest neighbor model performed better in terms of accuracy when k was set to 10 and computed better precision when the number of neighbors was set to 5.

Finally, when comparing the best versions of both models, the decision tree model was better in terms of both accuracy and precision.

5 Discussion

The results from the comparison between the decision tree and k-nearest neighbors (KNN) models indicated that the decision tree model outperformed the KNN model in terms of both accuracy and precision. Across different values of the maximum tree depth, the decision tree model exhibited stable performance, suggesting robustness to changes in hyperparameters.

More specifically, the decision tree model achieved its highest mean accuracy and precision when the maximum tree depth was set to 5, indicating that a moderate level of model complexity is maybe optimal for this particular dataset. On the other hand, the KNN model's performance varied more significantly with changes in the number of neighbors, potentially indicating that that the dataset might not be well-suited for the KNN algorithm.

Furthermore, I believe that the results from this analysis effectively proves that the numerical features of the dataset are sufficient in providing an effective prediction of water quality, with both accuracy and precision scores being at about 90%. This is a great insight and perhaps grounds for more research to figure out whether we can omit other features in the dataset that I have used and still produce relatively great performance. Are some chemicals more essential for determining the quality of water than others? Perhaps we can test out this hypothesis in a new study.

6 Conclusion

In conclusion, this project was a good starting point into the grounds of machine learning for me. It familiarized me with the various steps involved in performing such studies, with some steps that I would have not not considered beforehand. Indeed, the choice of a dataset, and how you are going to read that particular dataset are crucial decisions that can impact how you are going to fit your models and even open new avenues when it comes to hypothesis testing. I am now moderately interested in machine learning, and can see myself learning one or more concepts and tricks and teaching computers how to effectively learn.

References

https://www.kaggle.com/datasets/sivapriyagarladinne/telangana-post-monsoon-ground-water-quality-data?resource=download&select=ground_water_quality_2020_post.csv