CS361: Ground Water Quality Assessment using Machine Learning Algorithms

Ashish Bharti [1] **Mukka Koushik** [2] **Pratyush R** [3] **Badekai Vijesh Ramachandra Bhat** [4]

Abstract

Water holds paramount importance not only for human consumption but also plays a critical role in sustaining crops, livestock and poultry. Both plants and animals rely on direct access to groundwater for their hydration needs. The quality of this groundwater is crucial, as any deviation from acceptable standards can have detrimental effects on living creatures.
This project hence discusses an endeavor studying the purity of groundwater sources present in the state of Telangana with the help of classical Machine Learning Algorithms.

1 Introduction

1.1 Motivation

Telangana being an arid to semi-arid region with erratic rainfall patterns, relies heavily on ground water as crucial source of its water needs for agricultural irrigation, domestic water supply and livestock. In most of the rural areas, ground water serves as primary source of drinking water for households. Hence, measuring the water quality is of utmost importance.
Conducting laboratory tests for water quality can be time-consuming and expensive. This is where Machine learning comes in and provides a cost-effective and efficient way to predict water quality based on readily available data, reducing the requirement for extensive laboratory testing. Water quality assessment often involves analysis of multiple parameters.
The current dataset also has multiple parameters involved, to calculate various water quality measures, such as pH, Total Hardness, Residual Sodium Carbonate and other elemental metals. These parameters often have complex correlations and machine learning models are often used to identify such intricate relationships.

1.2 Target Problem

Developing various machine learning models for multi-class classification problem where the goal is to predict the ground water quality for various purposes such as drinking, livestock and poultry, crop cultivation based on provided features. The provided features are districts, mandals, villages, latitude , longitude and various chemical concentrations. The problem also intends to compare various machine learning models performance amongst each other.

1.3 Outline of proposed direction

The overall plan of the project we intend to implement for the groundwater analysis encompasses several steps which are mentioned below.

1.3.1 Exploratory data analysis

Checking for missing values and outliers should be done. Visualizing class imbalances. Visualizing variation of ground water quality across districts, mandals over time.

1.3.2 Data Preprocessing

As the data is present in separate files in temporal order, their integration is required. Handle missing values to ensure the models robustness. Normalization of the data must be done. Splitting the dataset into test and train sets will be done here.

1.3.3 Machine learning models building

Choosing various machine learning algorithms for multi-class classification. Training the machine learning models. Implementing hyperparameter tuning to improve model performance.

1.3.4 Results and Interpretation on various performance metrics

Models performance comparison on different metrics. Analyzing parameter/feature importance/contribution. Examining the confusion matrix and ROC curve to assess overall performance.

1.4 Major Challenges

The following major challenges could be faced while data preprocessing, building various machine learning models and evaluating the models. Missing data can hinder the

model performance and training. Handling such cases in the dataset is crucial. Ground water can vary spatially and temporally, so to capture these variations robust feature engineering should be used. The model's performance may be sensitive to the selection and quality of input parameters. Identifying the most relevant features is thus important. As, given dataset has 26 parameters, model overfitting may be observed. Underfitting may be observed if the model is not able to capture the complex interrelationships between parameters. Some classes may have significantly less samples compared to others, this could lead to class imbalance.

2 Methods

The methods we intend to implement are mentioned below.

2.1 Imputation

We plan to use linear regression/median analysis to fill in the missing data values.

2.2 Machine Learning Algorithms

The main purpose of the project is to compare the performance of the machine learning algorithms written independently by the authors with that of the external libraries. We intend to build the logic of the following algorithms independently:

2.2.1 Softmax Classification

Softmax is an extension of the regular logistic regression (binary classifier) and is widely used for multiclass classification.

Algorithm 1 Softmax Classification

Input: Training dataset $\{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\}, \eta$
For $k \leftarrow 1$ **to** c **do**
  Initialize $w_k = (w_1^k, \ldots, w_n^k)$ $\quad //w_1^k = b_k$
repeat
  Randomly select $(x, y) \in D$
  For $k \leftarrow 1$ **to** c **do**
    $h_k = e^{w_k^T x} / \sum_{l \in Y} e^{w_l^T x}$
  For $k \leftarrow 1$ **to** c **do**
    For $i \leftarrow 1$ **to** n **do**
      $w_i^k = w_i^k + \eta(\delta_{ky} - h_k)x_i$
  Adjust learning rate η
until termination
return $\theta = (w_1, \ldots, w_c)$

Algorithm 2 Decision Tree

Input: Training dataset $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, max depth d
Output: Trained decision tree T
Define `SPLITENTROPY`(D, f, t) to calculate split entropy based on feature f and threshold t
Define `FINDBESTSPLIT`(D) to find feature and threshold with lowest split entropy
Initialize decision tree T
function `BUILDTREE`(D, depth):
  If $depth > d$ or D is pure, create leaf node with majority class
  Else, find best split feature f and threshold t
  Split D into subsets D_{left} and D_{right}
  Create internal node with split rule (f, t) and attach left and right child nodes
$T \leftarrow$ BUILDTREE(training data, 0)

2.2.2 Decision Trees

Decision Trees is a robust machine learning algorithm for both classification and regression tasks. They help in capturing essential non-linear relationships in the data and identifying feature splits.

2.2.3 Random Forests

Theses build multiple decision trees during training and merge their predictions to improve the overall performance.

2.2.4 Naive Bayes

Naive Bayes is a simple and fast algorithm. Used mostly when features are conditionally independent given the class label.

2.2.5 K Nearest Neighbours

It makes minimal assumptions about the form of data distribution. It can handle complex decision boundaries and non-linear relationships.

2.3 Model evaluation and Performance metrics

Accuracy, Precision, Recall, F1-Score, Area under receiver operating characteristic curve(AU-ROC), Confusion matrix would be used to evaluate the models. Cross-Entropy would be used as loss function.

Algorithm 3 K-Nearest Neighbours

Input:
D: Training dataset $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$
k: the number of nearest neighbors
$d(x, y)$: a distance metric
x: a test sample
For each training sample $(x^{(i)}, y^{(i)}) \in D$
Compute $d(x, x^{(i)})$, the distance between x and $x^{(i)}$
Let $N \subseteq D$ be the set of training samples with the k smallest distances $d(x, x^{(i)})$
return the majority label of the samples in N

3 Intended Experiments

3.1 Data Analysis

As pre-monsoon and post monsoon data is available, we plan to compare the variations among these. More Exploratory data analysis, to consider spatial variations in ground water quality in various districts is also planned.

3.2 Other Problem Explorations

We plan to measure EWQI (Entropy based water quality index) and classify the given dataset based on it. WQI (Water quality Index) is generally dependent on the geographical location and requires expert domain knowledge to assign weights to features/parameters. Moreover, WQI calculation is time taking process. EWQI was found to not require domain knowledge (reference) and it adds weights according to the entropy values calculated for each of the parameters.
Experimenting with feature engineering techniques such as combining certain features to capture non-linear relationships.
Experimenting with ensemble methods such as XGBoost , GBM to compare the accuracy obtained by our models. We will use external libraries for XGBoost.
For avoiding overfitting, dimensionality reduction techniques may be used.

3.3 Model training and deployment

For the various models used like KNN classifier different hyper parameters are required. So, we plan to explore hyperparameter tuning techniques like Bayesian optimization.
The models could be deployed as web application, which take parameters such as chemical compositions etc, to predict the water quality, so as to increase the accessibility of the project. The application could also show the predictions according to each model.

4 References

1. Telangana Open Data Portal. *Telangana Ground Water Department **Pre**-Monsoon Water Quality Data*.Dataset retrieved from: https://data.telangana.gov.in/dataset/telangana-ground-water-department-pre-monsoon-water-quality-data

2. Telangana Open Data Portal. *Telangana Ground Water Department **Post**-Monsoon Water Quality Data*.Dataset retrieved from: https://data.telangana.gov.in/dataset/telangana-ground-water-department-post-monsoon-water-quality-data

3. Sudhakar Singha, Srinivas Pasupuleti, Soumya S Singha, Rambabu Singh , Suresh Kumar *Prediction of groundwater quality using efficient machine learning technique* https://pubmed.ncbi.nlm.nih.gov/34088106/

4. Mahmoud Y. Shams, Ahmed M. Elshewey, El-Sayed M. El-kenawy, Abdelhameed Ibrahim, Fatma M. Talaat , Zahraa Tarek *Water quality prediction using machine learning models based on grid search method* https://link.springer.com/article/10.1007/s11042-023-16737-4

5. Brijnesh Jain *Warped softmax regression for time series classification* https://link.springer.com/article/10.1007/s10115-020-01533-5