Ensemble Models for Surface Water Quality Predictions

Kennly Weerasinghe, Hubert Majewski, & Enoch Kim

March 31, 2023

1 Introduction

This project consists of data obtained from the National Science Foundation's (NSF) National Ecological Observatory Network (NEON) database. The objective of this project was to obtain data sets that quantified variables that are related to surface water quality, identify suitable predictors and a target response to construct ensemble based models. Within the NEON database we found five such data sets that we were able to work with and the proceeding text will describe our data engineering and machine learning model construction methodology and rationale.

2 Data

Of the available data sets the five that were selected include: Dissolved gases in surface water (NEON, 2023a), Temperature in surface water (NEON, 2023d), Nitrate in surface water (NEON, 2023b), Water quality (NEON, 2023e), and Surface water microbe cell count (NEON, 2023c). The data is segmented by site and by month. Of the sites ultimately chosen the following criteria were used: (1) sites that had sufficient entries between the years 2018-2020, (2) data availability across all five data sets for the sites chosen. The sites total to 24 and include the following: ARIK, BIGC, BLDE, BLUE, CARI, COMO, CUPE, GUIL, HOPB, KING, LECO, LEWI, MART, MAYF, MCDI, MCRA, OKSR, POSE, PRIN, REDB, SYCA, TECR, WALK, and WLOU. Since the data was segmented by month and we obtained data for 36 months this resulted in 36 .csv files per site times 24 sites for a total of 864 .csv files. NEON provides a package for R called neonUtilities that stacks the files, which simplified the process of working with the data. 864 files were stacked by the data set to reduce to 5 csv files. Due to the size of the 5 stacked files working in Mi-

crosoft Excel was not feasible. To work around this problem we imported them into PostgreSQL in order to analyze the data in depth, which will be detailed further in the next section.

3 Data Wrangling

Once the 5 stacked files were imported into PostgreSQL we observed that many of the columns not only included data we would find useful for modeling, but also metadata that ultimately would not serve a purpose in our modeling efforts. Once we identified the potentially useful columns of data we then created views of each data set. Of the 5 data sets, 3 (nitrate, temperature, and water quality) contained daily data measurements and 2 (dissolved gases and microbe cell count) had measurements taken weekly or monthly. The nitrate, temperate, and water quality data were combined into a view named 'nwt' using a full outer join in order to preserve any missing days worth of data. Similarly, the dissolved gases and microbe cell count data was also combined into one view named 'cdc' using a full outer join. Both tables were then exported to a jupyter notebook file for further data engineering using pandas in python.

A connection was established using sqlalchemy and the 'nwt' and 'cdc' tables were imported. The date, domain, and siteid were used throughout the process as indices. For both tables, the mean values per column feature were calculated based on siteid and the date to make sure that there was one unique value for nitrate mean, specific conductance, dissolved oxygen, sea level oxygen, pH, chlorophyll, turbidity, fdom, spectrum count, surface water mean temperature, CH_4 concentration, CO_2 concentration, N_2O concentration, and microbial abundance per ml. Both tables were then merged using a left join.

Spectrum count and sea level dissolved oxygen were dropped because spectrum count is a

measure of fDOM and chlorophyll, which already exists, and sea level dissolved oxygen is a transformation of dissolved oxygen. Leaving these features would add unnecessary dimensionality to the model and dropping them addresses potential issues of collinearity. Values for mean temperature, pH, chlorophyll, turbidity, dissolved oxygen, nitrate mean, fdom and specific conductance that were illegal based on the natural system were either coerced to a lower bound or converted to NaN for interpolation in the next step. The final table with the NaN values per feature were then interpolated using the time method and the remaining NaN values were replaced using the backfill method. The full table was analyzed using the ProfileReport from ydata-profiling package.

Based on the profile report, a number of features exhibited a log-normal distribution including: nitrate mean, specific conductance, dissolved oxygen, chlorophyll, turbidity, fDOM, CH4, CO2, N2O, and microbial abundance per ml. Based on this analysis we decided when building the models that it would be worthwhile to create additional columns with log transformations of the features that exhibited the log-normal distribution in order to evaluate level-level, log-level, and log-log models.

4 Models

The ensemble tree-based models produced made use of three algorithms: XGBoost, LightGBM, and Random Forest. All three of these algorithms are capable of deployment for classification or regression tasks but for the purposes of this work they were used for regression. XGBoost and LightGBM makes use of gradient boosting, whereas Random Forest makes use of bagging i.e. bootstrap aggregation; boosting and bagging both have their advantages and disadvantages. In brief, boosting can reduce both variance and bias but has a higher risk of overfitting, while bagging

can reduce variance without being as prone to overfitting but may not be as effective in handling imbalanced data sets. The biggest difference between XGBoost and LightGBM is the speed of training, with LightGBM making using of gradient based one-side sampling to reduce the time needed to train. Three separate models were built on each algorithm, a level-level model, log-level model, and a log-log model. The level-level model had the features and response data in its raw form post interpolation. The log-level model made use of a log transformation of the response variable (microbial cell count). The log-log model had both the response variable and a subset of the features log transformed for a total of 9 models.

The error metrics used to assess the models were RMSE and R^2 give by the following formulas:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{n}} \tag{1}$$

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

The code and the results can be seen in more detail within the notebook files.

5 Out of Sample Results

LightGBM Models

The level-level LightGBM model had RMSE of 2950713.773 and R^2 of 0.826. The log-level LightGBM model had RMSE of 0.655 and R^2 of 0.855. The log-log LightGBM model had RMSE of 0.668 and R^2 of 0.849.

XGBoost Models

The level-level XGBoost model had RMSE of 2842615.358 and R^2 of 0.839. The log-level XGBoost model had RMSE of 0.655 and R^2 of 0.855. The log-log XGBoost model had RMSE of 0.667 and R^2 of 0.849.

Random Forest Models

The level-level Random Forest model had RMSE of 2778201.722 and R^2 of 0.846. The log-level Random Forest model had RMSE of 0.653 and R^2 of 0.856. The log-log Random Forest model had RMSE of 0.653 and R^2 of 0.856.

Note: The log-level and log-log RMSE values are in the log transformed units.

6 Model Evaluation

Based on our results, all 9 models performed similarly when evaluated against the R^2 values. If we look at the RMSE values of the level-level models they are high, but they fall within the the range of recorded values in the data set, with a minimum of 4548.889 to a maximum of 62606310. The models perform better than selecting the base prediction which would be $\bar{y}$. Our models exhibit low variance in error with respect to the training error metrics but there is a high degree of bias (see the notebook for training error). One method to address the high bias would be to increase the model size. For XGBoost and LightGBM, this would involve expanding the space of values for the hyper parameters such as the learning_rate, max_depth, and num_leaves; and for Random Forest, n_estimators and max_depth could be expanded to increase model size. Increasing the model size would result in increased time training the models and potentially increased risk of overfitting, which could be mitigated by incorporating L1 and L2 regularization. Other methods

of addressing the error include: in depth error analysis of the predictions, evaluation of the current available features, and or consulting with a domain expert to see if the model as constructed may be missing information that would improve the models ability to capture the phenomena.

7 Interpretability

Of the XGBoost models, the log-log model was selected to construct a SHAP analysis to visualize and enumerate the feature interactions to understand the impact of the feature contributions towards the predictions. The order of impact on the predictions, from greatest to least, was as follows: CH_4, CO_2, N_2O, dissolved oxygen, nitrate mean, fDOM, specific conductance, chlorophyll, pH, mean temperature, and turbidity. The predictions when analyzed by site exhibited substantial variability when analyzing for feature interactions (see XGBoost notebook).

References

NEON (2023a). *Dissolved gases in surface water (DP1.20097.001)*. DOI: 10.48443/PF8R-WZ28. URL: https://data.neonscience.org/data-products/DP1.20097.001/RELEASE-2023.

— (2023b). *Nitrate in surface water (DP1.20033.001)*. DOI: 10.48443/MM5S-DF23. URL: https://data.neonscience.org/data-products/DP1.20033.001/RELEASE-2023.

— (2023c). *Surface water microbe cell count (DP1.20138.001)*. DOI: 10.48443/8AH9-H316. URL: https://data.neonscience.org/data-products/DP1.20138.001/RELEASE-2023.

— (2023d). *Temperature (PRT) in surface water (DP1.20053.001)*. DOI: 10.48443/33ZJ-CQ24. URL: https://data.neonscience.org/data-products/DP1.20053.001/RELEASE-2023.

— (2023e). *Water quality (DP1.20288.001)*. DOI: 10.48443/MPW3-3Q06. URL: https://data.neonscience.org/data-products/DP1.20288.001/RELEASE-2023.