Syracuse University
School of Information Studies
IST 707 – Applied Machine Learning

Building Machine Learning Models for Predicting Water Quality, Using Cape Coral, FL Dataset

Group Members

Maruthamuthu Kanagarathinam Sowmeya

Shriya Rajendra Gawade

Nikhil Sriram Budamaguntala

Besjana Muraku

Table of Contents

Abstract

Water quality has faced a significant decline over the recent years as a result of growing urbanization, high rates of industrialisation, and improper garbage disposal. Predicting water quality is now an important issue in preserving the aquatic environment because it is necessary for a healthy civilization. In this project we will leverage with the support of machine learning, the possibility of predicting water quality in areas where the water is not only used for drinking, but its quality is important for irrigation, and a variety of animal species. Existing methods use algorithms that decrease the accuracy of unbalanced datasets and increase computational complexity, with a primary focus on either water quality. This study explores these existing methods and explores new one for better balanced data, higher accuracy. The machine learning models used for balancing dataset are SMOTE and under sampling where the one with better result is SMOTE. After having a better-balanced dataset, knn, random forest, decision tree models and artificial neural network (ANN) are explored for forecasting water quality. The results indicate that ANN model performs better with an accuracy of 95%.

Introduction

As one of the essential supporters of life, water can also become the harbor of critical diseases. This natural resource is very beneficial for domestic, industrial and agricultural use. Although it’s widely distributed, there are many areas which suffer from this resource as it cannot be used due to its high level of pollution. According to the most recent surveys on national water quality from the U.S. Environmental Protection Agency, nearly half of U.S. rivers and more than one-third of lakes are polluted and unfit for swimming, fishing, and drinking. Water pollution is damaging to living species, and it causes a variety of illnesses. It has caused damage to the aquatic ecosystem, necessitating the search for potential solutions. Several global laws and guidelines have been established to protect water quality. Also, local authorities, industries and individuals have a high impact on water contamination or pollution. To get quality of the water up to the standard needed for public usage, most water supplies must be appropriately managed. To guarantee that tap water is safe to drink, the EPA establishes standards that limit the number of certain contaminants in water supplied by public water systems. The US Food and Drug Administration (FDA) sets standards for pollutants in water that must provide the same level of public health protection. Water quality alters according to the source of impurities, the time of year (season), the geological formation and other factors.

The goal for our project is to predict the quality of water in the Cape Town city, Florida and give better insights on what range of chemical factors contribute on deciding the water quality levels and how much contaminated they are.

Objective

Since we have high levels of pollution in nearly half of US rivers and more than one-third of lakes are polluted and unfit for swimming, fishing, and drinking. Water pollution is damaging to living species, and it causes a variety of illnesses. It’s has caused damage on the aquatic ecosystem, necessitating the search for potential solutions. Several global laws and guidelines have been established to protect water quality.

Use Case of the project

This would also serve as a supporting tool for government and agencies on decision-making for water resources.

In Florida, “CERP dictates a complex funding arrangement whereby the federal and state governments shared construction costs equally, subjecting the plan to the unpredictable nature of the legislative budgeting process at both the federal and state levels”

- Florida water recovery measures by Government

In the following section are describe can be found an analysis of data to understand better their characteristics. Then we try to understand if there is a specific pattern for the missing values, which will then be helpful to define the model that should be used for the data imputation. After imputing the missing data, the next phase concerns balancing. Since one of the reasons for bias in machine learning models is unbalanced data, we try resolve this issue by trying several algorithms. The final step is building the machine learning models that predict whether the quality of water is “poor” or “good”.

Project Workflow

To implement this project, we followed specific steps. Initially we had to define the dataset to be used. The second step consisted in working with the dataset which consist of: selecting the important features, impute missing data and balance the water quality categories. For each of the steps of working with the dataset, several models were tested and the best performing one is chosen. Next, several training models are tested, such as naïve bayes, knn, random forest and finally artificial neural network. The project workflow is shown in figure 1.

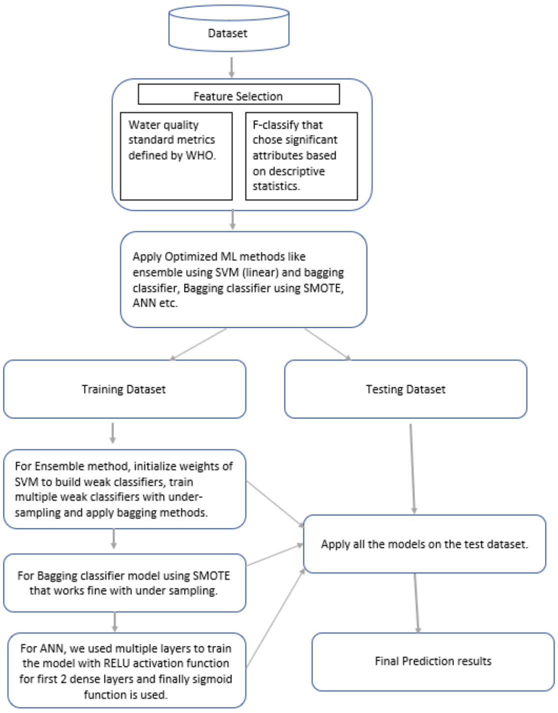

Dataset
Feature Selection
Water quality standard metrics defined by WHO.
F-classify that chose significant attributes based on descriptive statistics.
Apply Optimized ML methods like ensemble using SVM (linear) and bagging classifier, Bagging classifier using SMOTE, ANN etc.
Training Dataset
Testing Dataset
For Ensemble method, initialize weights of SVM to build weak classifiers, train multiple weak classifiers with under-sampling and apply bagging methods.
For Bagging classifier model using SMOTE that works fine with under sampling.
For ANN, we used multiple layers to train the model with RELU activation function for first 2 dense layers and finally sigmoid function is used.
Apply all the models on the test dataset.
Final Prediction results

Figure 1: Project workflow

Dataset Description

We have explored three datasets for testing water quality, they are:

- ArcGIS Hub Water Quality Cape Coral Dataset[1]
- U.S. Fish and Wildlife Service[2]
- NYC Water Quality[3]

[1] https://hub.arcgis.com/datasets/b0579ba7aa1145e090c3a74e295564df/explore?location=26.646897%2C-81.929073%2C11.71&showTable=true

[2] https://data.doi.gov/dataset/water-quality-data

[3] https://redivis.com/datasets/g8je-49bxaag3y

Among these websites we have selected data provided by the city of Cape Coral because it has more significant features and helps to predict our model more accurately. The data set contains 18148 observations with 48 variables, it also has missing values 229 in water temperature, 385 in specific conductance, 299 in dissolved oxygen, 567 in pH, 293 in sampling depth, 303 in salinity, 18148 in dissolved oxygen saturation, 6956 in total dissolved solids, 12073 in Fecal coliforms, 18148 in organic nitrogen, 15858 in volatile suspended solids, 15893 in volatile dissolved solids. Among all the features we need to select more significant and eliminate remaining fields. The data set contains 18148 observations with 48 variables.

Data Pre-processing

Before delving into defining and building the models for predicting the water quality, it is important to initially understand the data and perform necessary data clean up and preparation. The selected dataset contains 18148 observations(records/entities) with 48 variables (attributes). Considering that the more attributes a dataset has it lessen the chances that the predictive model would be stable. The initial filtering of attributes was performed by considering the standards defined by World Health Organization (WHO). The following table shows the list of attributes selected which have a higher degree of importance in measuring water quality, and their range of values indicating a good quality of water.

Attributes	**Description**	**Threshold Values**
Water Temperature	Temp of water collected at certain point of time	30-35 Celsius
Specific Conductance	Collective concentration of ions in water	<=50
Dissolve Oxygen	Amount of oxygen present in water	<6
Ph	Measure of acidity/basicity in water	6.5-8.0
NH3	Amount of combined nitrogen and hydrogen present	<0.2
NO2	Amount of Nitrites present in water	<1
NO3	Amount of Nitrates present in water	<10
Salinity	Salt content dissolved in water	<7
Total Kjeldahl Nitrogen	Total sum of ammonia nitrogen and organic nitrogenous compounds	<1
Total Nitrogen	Amount of Nitrites and Nitrates contaminant present in water	<10
Total Phosphates	Total amount of phosphorus present in water	<10
Bioloxd	Dissolved gas molecules held in water	<5
Alkalinity	Amount of neutralizing power of acids and bases	<200
Fecal Coliforms	Amount of harmful pathogens present in water	<200

Selecting Features on basis of domain Knowledge: There are different chemical components present in the water with multiple combinations that affect the quality of water. So, by intruding further with the domain knowledge specific to chemicals we have taken 12 variables refined out of 48 as those features are considered important as per the bio chemist's knowledge and reference articles [1] related to water quality and its contamination.

Temperature came out to be an important factor in deciding water quality levels as the water that's hot generates lead which is harmful as compared to a normal or cold water. But we have ignored that attribute as all observations is recorded around the mean room temperature of 30. Also, the lab recorded water samples by force fitting the chemicals are eliminated for model training as it might create a bias while training the models. The following heatmap, figure 2, indicates that the selected attributes have no significant correlation with each other.

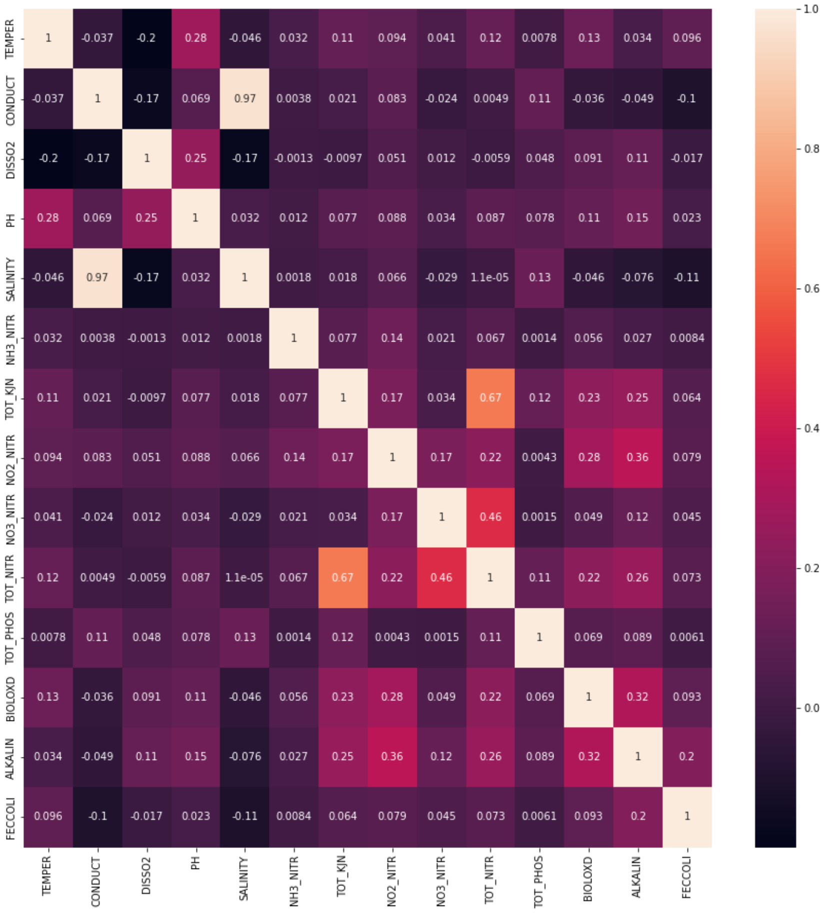


Figure 2: Attribute correlation heatmap

Defining a Target Variable

For defining the target variable, we used the water quality standards defined by the WHO and created a new variable named water quality which has two attributes "good" and "bad". The new variable was computed using:

- Conduct < 50.0
- Dissolved Oxygen < 6
- 6.5 <= PH <= 8.0
- Salinity < 7
- NH3 < 0.2
- Total Kjeldahl Nitrogen < 1.0
- NO2 < 1
- NO3 < 10
- Total Nitrogen < 10

- Total Phosphate <10
- Bioloxd < 5
- Alakinity < 200
- Fecal coliform < 200

Handling Missing Values

As presented earlier in this report, in the selected dataset are detected missing variables in some of the attributes. Following, an information about the amount of missing values for each attribute, taking in consideration that the overall dataset consists of 18, 148 observations.

```
TEMPER        229
CONDUCT       385
DISSO2        299
PH            567
SALINITY      303
NH3_NITR     4890
TOT_KJN      4939
NO2_NITR     5353
NO3_NITR     5403
TOT_NITR     5280
TOT_PHOS     5041
BIOLOXD      5554
ALKALIN      6284
FECCOLI     12073
```

The following bar chart shows the completeness of each attribute. Here it can be identified that the attribute with the most missing data is FECCOLI, where more than 50% of values are missing.

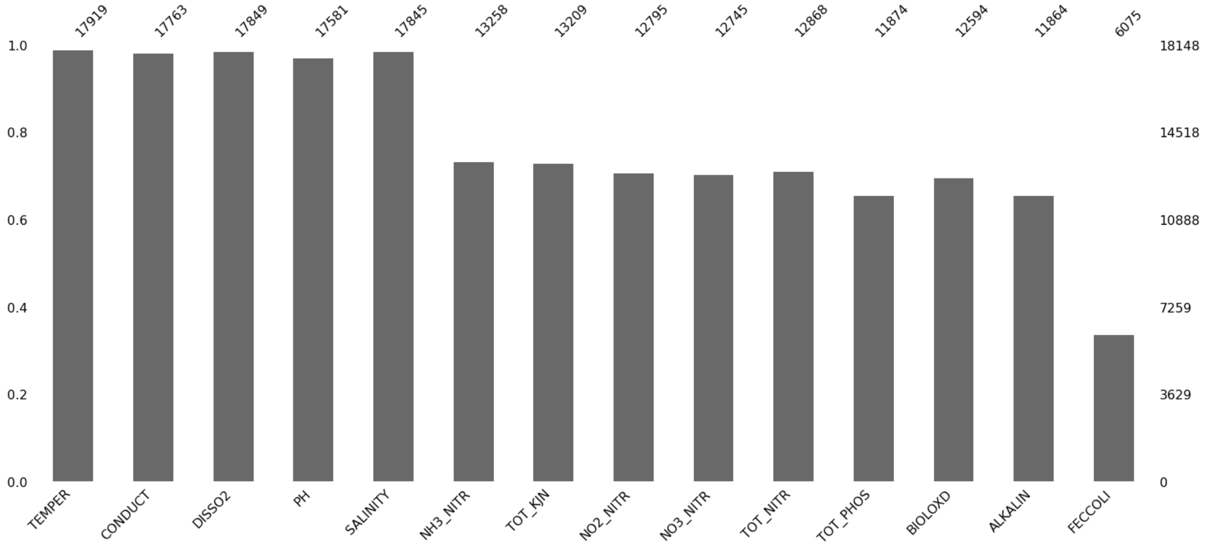


Figure 3: Missing data for each attribute

Next, we try to identify any correlation of missing values between variables. The following heat map serves for this purpose. As it is shown, there are several pairs of attributes that seem to be correlated regarding the missing values, such as TOT_KJN and NH3_NITR, or NO2_NITR and NH3_NITR. We can say that the missing values are not completely at random but still we cannot identify a specific pattern. Therefore, we are assuming that we are dealing with MAR data.

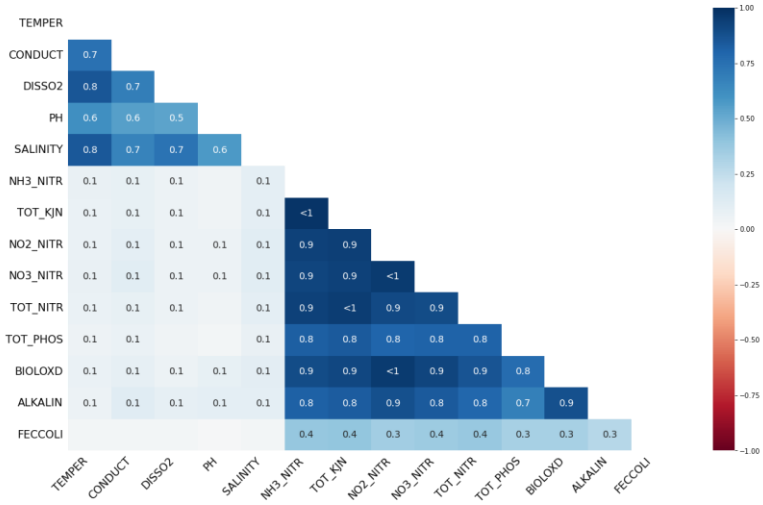


Figure 4: Missing data heatmap

For handling missing values, we tried three methods Denoising Encoder, KNN impute and soft impute. We have used Denoising Encoder that converts meaningless values into a clean data.

Denoising encoder

Denoising encoder is a neural network that trains the data identify patterns in noisy data and map appropriate values onto the meaningful values by passing it to the encoder which takes noisy data and reduce it to a lower dimensional representation which is then passed on to the decoder to get a cleaned version of dataset which is an imputed one.
The Denoising encoder consists of multiple layers starting from encoder given with input dimensions and activation layer – 'rectified linear unit' along with input data. Decoder with the same number of input dimensions and activation layer as linear along with encoded input. Then compiled with Adam's optimizer and loss function is calculated as mean squared error. The model is trained for 100 epochs with a batch size of 16. We split our dataset in 80% training and 20% testing. The resulting loss is loss: 2.23e-15 and the dataset has no missing data.

Once the estimated values are filled, we store them as the imputed data and define dummy variable as "Water Quality" by creating a subset of rules for variables by defining the range of those values.
These rule set resulted in the 17164 observations that recorded poor water and 984 as good water.

KNN Impute

Next, we have implemented KNNImpute by defining the k neighbours as 4 and train the model. The K-Nearest Neighbours algorithm is used by the KNN Impute method to fill in missing numbers in a dataset. The basic idea is to use the remaining features to find the K nearest neighbours of each sample with missing values, and then use their values to fill in the missing values. K is a hyperparameter that controls how many of the closest neighbours are used for estimation. KNN Impute can be used with both numbers and lists. KNN Impute assumes that

the missing values are missing at random (MAR), and the estimated values assume that the missing values are similar to the observed values.
This rule set reported in 16708 observations as poor water and 1440 as good.

SoftImpute

SoftImpute is a method for filling in missing data in matrices. It is used to fill in missing data in datasets. It is based on low-rank matrix completion, which says that the data can be expressed by a low-rank matrix plus some random noise. SoftImpute takes into account the rank of the matrix and tries to reduce the sum of squared errors between the observed data and the low-rank approximation. SoftImpute can deal with missing data in many different ways, such as data that is missing totally at random (MCAR), missing at random (MAR), or missing not at random (MNAR). SoftImpute can also handle datasets with a lot of missing data and can be used to fill in missing values for both continuous and categorical data. SoftImpute is used a lot in areas like biostatistics, bioinformatics, and the social sciences. It is part of the 'fancyimpute' Python package.

Finally, we have executed the Soft Impute for the same above rule set given for KNN and got the counts of 15618 observations as poor water and 2530 as good water.

In the following image is shown the distribution of values for each attribute selected, after the imputation.

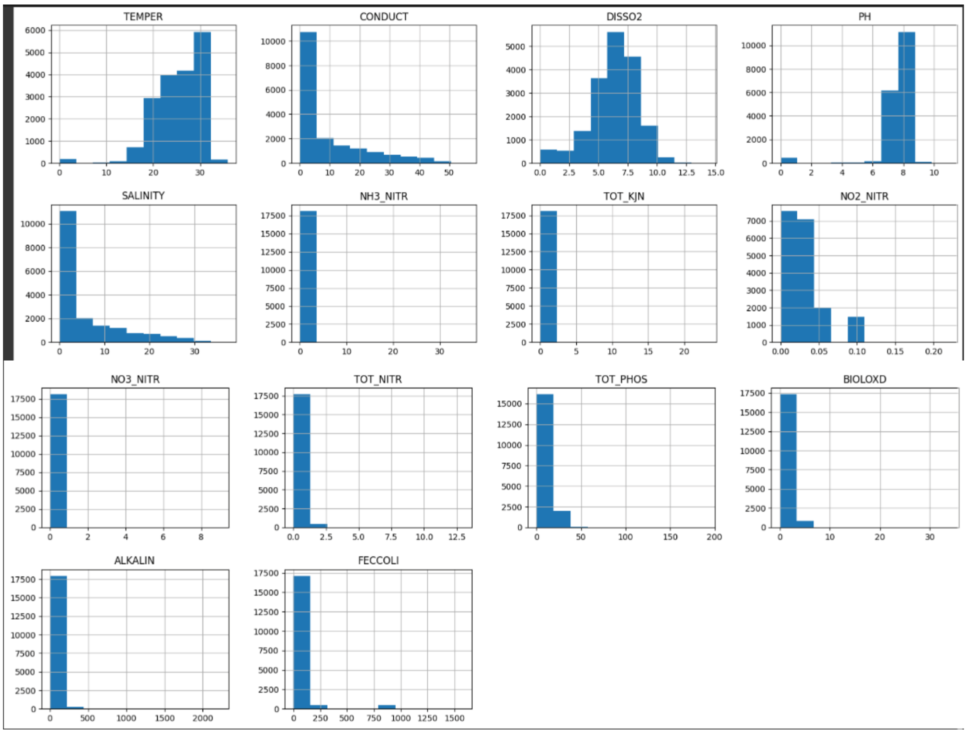


Figure 5: Value distribution for each attribute

Although the data is imbalanced yet, out of the three balancing methods, DAE gave better results in stabilizing both the good & poor water to utmost extent.

Scaling Features

We will investigate transforming the data into its normalized form using MinMaxScalar() function. Defining Train and Test Set: The dataset is split into training and testing dataset with 80% of train data and 20% of test data.

Results

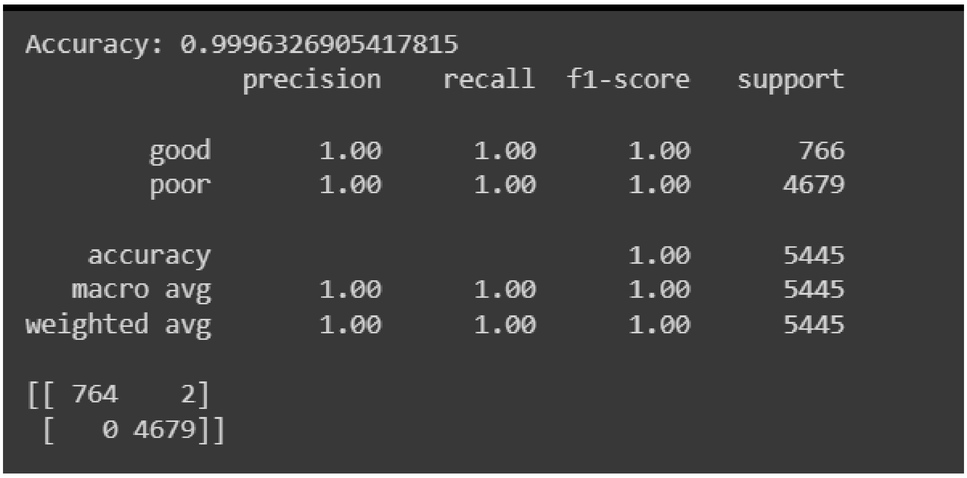
We have highly imbalanced class distribution so there are chances of model being bias and overfit. We decide to approach this issue using SMOTE and Ensemble Methods.

SMOTE

This approach used in machine learning to address imbalanced datasets where one class has considerably fewer samples than the other. It is a data augmentation technique that generates synthetic examples of the minority class by producing new samples that fall between existing minority class samples. This technique aims to create a more balanced class distribution by providing additional representative samples for the minority class. Ultimately, using SMOTE can enhance machine learning models' performance when working with imbalanced datasets.

We used SMOTE to generate new data which provides us with more balanced data and then we used Bagging Classifier to train the model.

We have got an accuracy of 99.9% which is an overfit model in predicting the water quality.



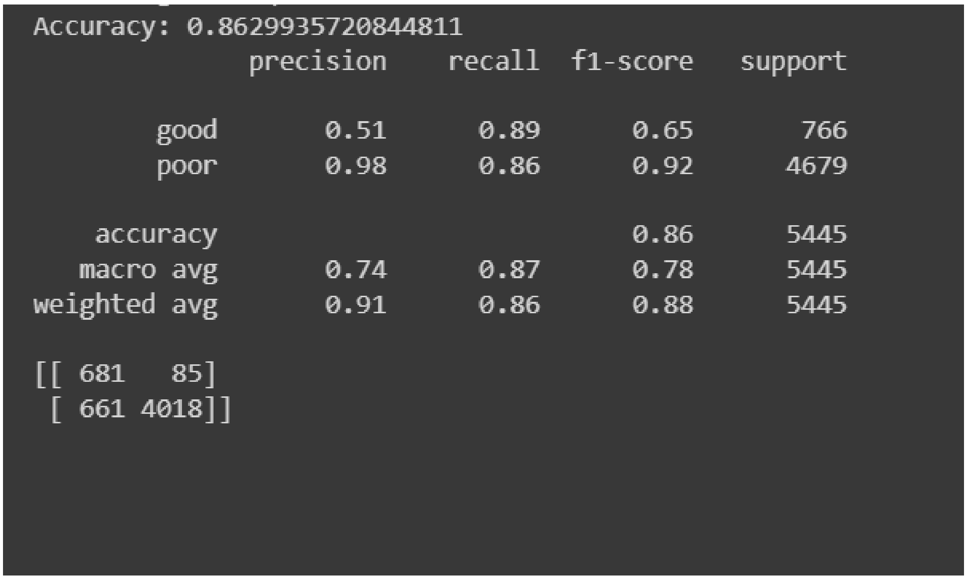
```
Accuracy: 0.9996326905417815
              precision    recall  f1-score   support

        good       1.00      1.00      1.00       766
        poor       1.00      1.00      1.00      4679

    accuracy                           1.00      5445
   macro avg       1.00      1.00      1.00      5445
weighted avg       1.00      1.00      1.00      5445

[[ 764    2]
 [   0 4679]]
```

Figure 6: SMOTE model output

Ensemble Method

We utilized the f_classif score to choose the most important features and assigned weights to two classes. Then, we developed an SVM model to tackle overfitting and enhance the model's stability. Additionally, we created a bagging classifier model and conducted under sampling to balance the class distribution. Lastly, we trained the model.

The Accuracy of Ensemble turned out to be 86.2% which seems to be decent prediction as compared to the above SMOTE method and could be improved by tuning of hyperparameters with different range of values.



```
Accuracy: 0.8629935720844811
              precision    recall  f1-score   support

        good       0.51      0.89      0.65       766
        poor       0.98      0.86      0.92      4679

    accuracy                           0.86      5445
   macro avg       0.74      0.87      0.78      5445
weighted avg       0.91      0.86      0.88      5445

[[ 681   85]
 [ 661 4018]]
```

Figure 7: Ensemble method output

However, we will be working on this data with certain models that gives best results for the imbalanced dataset.

Predicting Models

In the following section we describe the machine learning models built and tested for predicting water quality. Naïve Bayes is used as the base line model

Naïve Bayes

Naive Bayes is a probabilistic algorithm used in machine learning for classification. It is based on Bayes' theorem and the idea that traits don't depend on each other. Naive Bayes figures out how likely it is that a given case belongs to a certain class by multiplying the probabilities of the features that make up that class. It's called "naive" because it thinks that each trait is separate from the others. The idea that features are independent can sometimes be taken too far, which makes the program less accurate than others.

This model is used as a base line comparison for the other models. In this case we used the parameter tuning of var_smoothing (portion of the largest variance of all features that is added to variances for calculation stability) is tested against these values [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]. The result of the execution is as following:

Best parameters: {'var_smoothing': 1e-09}
Accuracy: 0.800
Precision: 0.223
F1 score: 0.364

KNN

KNN, which stands for "K-Nearest Neighbours," is a method for machine learning that can be used for both classification and regression. It is a non-parametric method, which means that it doesn't make any assumptions about how the data are spread out. KNN works by finding the K data points that are closest to the new instance being classified based on some similarity measure, like Euclidean distance. Then, the new instance's projected class or value is based on the majority class or average value of its K closest neighbours. KNN is a method that is easy to use and works well with both linear and nonlinear data.

In our case we have knowledge about the domain and we know that we want our information to be classified in two clusters. By using the balanced dataset as a result of SMOTE, we execute in the KNN model and the results are as following:

Accuracy: 0.939
Precision: 0.488
F1 score: 0.562

Decision Tree

In this model the parameters to be tuned are max_depth where the values to be tested are [3, 5, 7, 10], min_samples_split where the values to be tested are [2,5,10], and min_samples_leaf [1,2,4]. The results after the execution of the model are as following:

Best parameters: {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}
Accuracy: 0.998
Precision: 0.981
F1 score: 0.984

Random Forest

Random forest is a method for machine learning that is used for both classification and regression tasks when it is supervised. It is a type of learning that uses various decision trees to improve the model's ability to predict. The algorithm works by making a forest of decision trees. Each tree is trained on a random subset of the training data and a random subset of the features. During prediction, all of the results from each tree are added together to make the end prediction. Random forest can handle high-dimensional data, noisy data, and missing values. It also has built-in ways to choose which features to use and can figure out how important each one is. But it can overfit on small datasets or datasets with many traits that are similar.

In Random Forest we need to perform parameter tuning. In this case the parameter n_estimators, which indicates number of trees you want to build before getting the most votes or averages of predictions (more trees give you better speed, but they also slow down code) is tested against the values [50,100,150,200]. The results of this execution are as following:

Best parameters: {'n_estimators': 200}
Accuracy: 0.999
Precision: 0.993
F1 score: 0.992

Artificial Neural Network

ANNs are machine learning models based on brain anatomy and function. Neurons accept inputs, compute, and output. The initial layer of an ANN receives input, and successive layers process and transform it before creating the output. Each neuron in a layer is connected to neurons in the previous and/or next layer via weighted connections that influence signal strength. The model's task performance is optimized by adjusting weights during training. Backpropagation estimates the gradient of the model's loss function with respect to the weights and updates the weights to decrease loss. ANNs excel in classification, regression, and clustering on many benchmark datasets. They handle complex, high-dimensional data like pictures, audio, and text well. They are computationally expensive to train and may overfit if the model is too complicated or the dataset is too little.

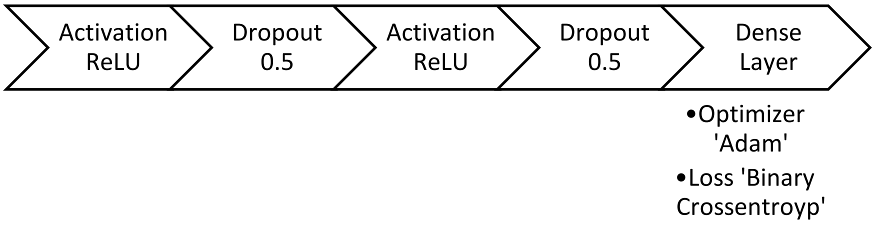


Figure 8: ANN model

Figure 8 depicts the suggested ANN model architecture which was a solution provided by (Rustam et al. 2022). The architecture of this model consists of an initial activation, followed by a dropout, then re-applied another activation with an dropout. The rectified linear unit (ReLU) activation layer linearizes the feature set after the dense layer of 256 neurons calculates input data. To simplify the model, we employed a dropout layer with a 0.5 dropout rate to randomly eliminate 50% of the neurons after the activation layer. After the dropout layer, the second dense layer has 256 neurons, followed by the ReLU activation layer and 0.5 dropout layer. We employed a dense two-neuron for predicting water quality layer by using the Adam optimizer and calculate the loss function using binary crossentropy. The ANN model is trained for 20 epochs and results in an accuracy of 95%.

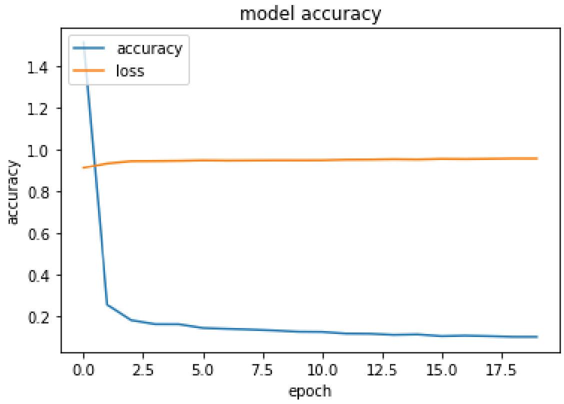


Figure 9: ANN model training

Conclusion

In this project the selected dataset played an important role on defining the predicting model. The amount of missing data and the imbalance of poor and good water were the key steps important to be implemented before building the final classification model. For the imputation three algorithms were tested of which the decoder autoencoder model was selected since it had a better RMSE. Among the two balancing algorithms, the SMOTE one is selected. The resulting dataset got as output of the second step is tested to several classification models. Following a table grouping all the results for each predictive model. The high values of accuracy are due to the fact that still the dataset has more poor records compared to good one.

Model	**NB**	**KNN**	**RF**	**DT**	**ANN**
Accuracy	0.8	0.939	0.999	0.998	0.95
F1 score	0.36	0.562	0.992	0.984	0.98

References

[1]. Ubah, J.I., Orakwe, L.C., Ogbu, K.N. *et al.* Forecasting water quality parameters using artificial neural network for irrigation purposes. *Sci Rep* **11**, 24438 (2021). https://doi.org/10.1038/s41598-021-04062-5

[2]. Amir Hamzeh Haghiabi; Ali Heidar Nasrolahi; Abbas Parsaie. Water quality prediction using machine learning methods. Water Quality Research Journal (2018) 53 (1): 3–13. https://doi.org/10.2166/wqrj.2018.025

[3]. Mengyuan Zhu, Jiawei Wang, Xiao Yang, Yu Zhang, Linyu Zhang, Hongqiang Ren, Bing Wu, Lin Ye,A review of the application of machine learning in water quality evaluation, Eco-Environment & Health,Volume 1, Issue 2,2022, Pages 107-116, ISSN 2772-9850, https://doi.org/10.1016/j.eehl.2022.06.001.

[4]. Joslyn, Kathleen, "Water Quality Factor Prediction Using Supervised Machine Learning" (2018). REU Final Reports. 6, https://pdxscholar.library.pdx.edu/reu_reports/6