Using RF classifier and Logistic Regression algorithms for binary classification of water potability

Abstract

Water availability ranges amongst the most crucial needs for humankind, and since the very beginning civilization has been sprawling. It is utterly necessary that population have access to potable water, wherever they are, on a daily basis. This short report explores several way of implementing Machine Learning algorithms so as to determine their accuracy on predicting water potability given a set of parameters influencing water.

1. Introduction

Water bears great significance for most organisms and living beings, ranging from Human to the most petite arthropods. As it composes no more than three quarters of our planet's surface, it is of great importance for consumption, whether it be directly from the faucet or to wash our food, just to cite a few. The reason why freshwater is so important is because it constitutes a very small part of the total water on the globe: amongst the less than 2% of available freshwater on Earth, barely 0.3% are actually accessible to us for consumption purposes. This is to say the issues one would be confronted to if they were to be deprived of the *blue gold*.

This is actually an issue that has been addressed since studies have been conducted on water quality assessment, notably with the rise of *ecotoxicology* from the early 1970s onward linked to the necessity to maintain our water resources pristine and the least polluted possible, after several health issues had been made public and certain animal populations had been subject to polluted water (Lecomte, 2012).

Henceforth, ecotoxicologists as well as environmentalists have been working on sanitizing and depolluting waters originating from WWTPs or spilled into the environment. That way, the waters can be available to their consumers and pose no threat to the population. It has become increasingly important to be able to estimate water pollutants concentrations in critical situations.

In case we are able to collect data from different waterbodies that might be subject to pollution, an interesting approach is to try to predict their potability based on a set of different parameters that can come into play when it comes to determining pollution.

2. Aim

The aim of this practical work is to determine if a model can be trained through a Machine Learning algorithmic approach using 2 different binary classification ML algorithms, and to estimate its accuracy on classifying potability thanks to the algorithms chosen.

3. Dataset

To meet this report's expectations, we have decided to focus on a well-known, commonly-used and often referred to dataset of water potability simply named 'water potability', available and directly downloadable as a .csv file from the online database *kaggle* using the following link: `https://www.kaggle.com/`

"Welcome to kaggle, there's a lot of bad data" (Uarov (2020), on kaggle discussion forum) quote informs us upon the dataset's origins, which tells us this is a fake dataset. Indeed, the data resemble too closely to normalised data - difficult to obtain for real data - but this dataset is ideally suited to be applied to real situations.

The dataset was designed to be handled using different ML algorithms as well as statistical analysis tools. It consists of a 3276-row by 10-column dataframe that represents 9 different water quality metrics ("parameters") that have been sampled on 3276 different water bodies (Kadiwal (2020), (`https://www.kaggle.com/datasets/adityakadiwal/water-potability`). The various metrics are:

- pH
- Hardness
- Solids (total dissolved solids TDS)
- Chloramines
- Sulfate
- Conductivity
- Organic Carbon

- Trihalomethanes
- Turbidity

And a sole column that classifies the water into potable (1) or non-potable (0): Potability: 0 or 1.

4. Methodology

The methodological approach involves using mainly 2 different ML classification algorithms adapted to binary classification. The first one we have used is Random Forest Classifier (RF), the second one is Logistic Regression (Log Reg), and the idea is to compare their accuracy scoring in terms of predicting water potability.

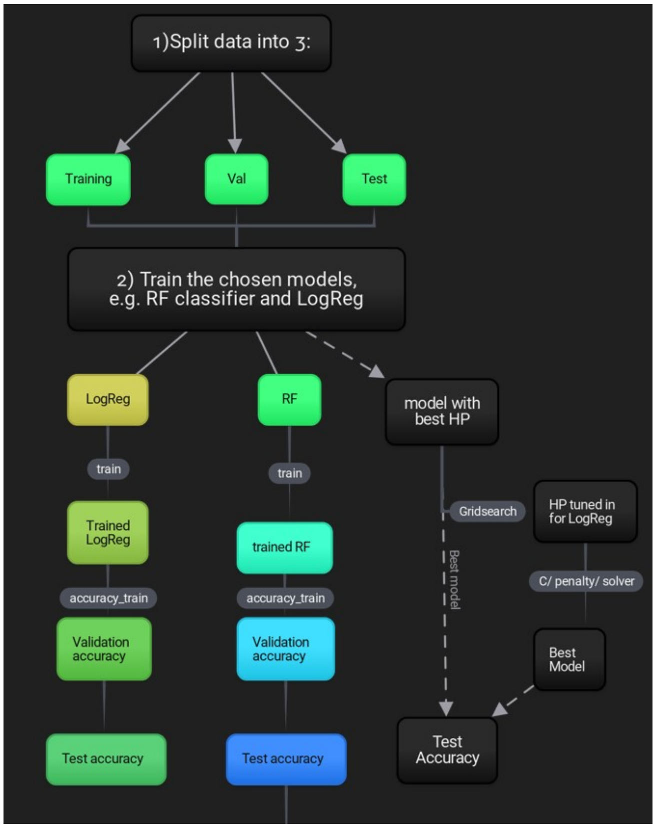

Figure 1. Schematic used for methodology

5. Results

The Python code on which the results were generated is available (hopefully) at this link: https://github.com/Etienne-98/2022_ML_EES/blob/main/Personal_Project_Etienne_Delaloye.ipynb.

5.1. Logistic Regression

5.1.1. LogReg() instantiation

The Logistic Regression classifier first instantiated without hyperparameters was fitted and predicted using the common hands-on Python commands. Its score on the test, validation and the train test were computed, as well as the overall accuracy score, which did unfortunately not exceed 57.57%.

5.1.2. LorgReg() Hyperparameters Search

When applying the GridSearchCV hyperparameter tuning method, it provides us with the best HP in order for the model to perform the best. However, we have to handle its results delicatly, as this may not always provide perfect Hyperparameters for the task at hand. In our case, GridSearchCV did provide us the following hyperparameters tuning for the best accuracy:

- Tuned Hyperparameters are : 'C': 10.0, 'penalty': 'l2', 'solver': 'liblinear'
- Accuracy is : 61.35295542635658

where:

- C:'10', the inverse of regularization
- penalty:'l2', the regularization
- solver: 'liblinear', the solver

The resulting score is not quite bad, after all! we can see that it is over 60%. For better visualization, we can take a peek at the classification report created using the command classification_report() (figure 2):

Table 1. Classification accuracies for Logistic Regression classifier on test set

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.58	0.99	0.73	231
1	0.67	0.02	0.04	172
ACCURACY			0.58	403
MACRO AVG	0.62	0.51	0.31	403
WEIGHTED AVG	0.62	0.58	0.44	403

5.2. Random Forest

5.2.1. RandomForestClassifier() instantiation

After getting the results from our Logistic Regression classifier, we did the same with the RF classifier, which scored a bit less at first. We obtained the following results:

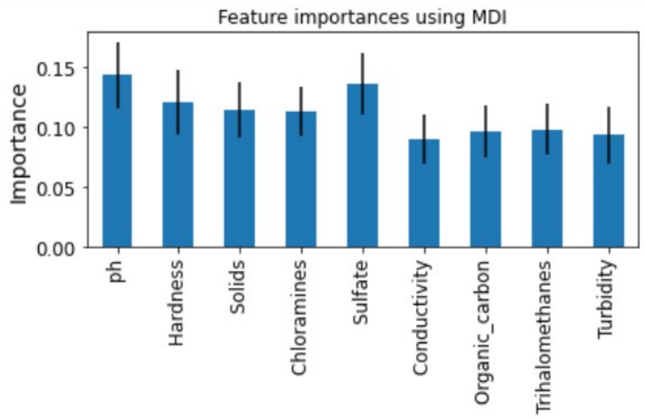

Figure 2. Random Forest Feature importance

- The RF classifier Model Accuracy on Validation set is: 58.39 %
- The RF classifier Model Accuracy on Test set is: 67.0 %

The classification report is visualised in table 2 as follows:

Table 2. Classification accuracies for RF classifier

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.67	0.85	0.75	231
1	0.68	0.43	0.53	172
ACCURACY			0.67	403
MACRO AVG	0.67	0.64	0.64	403
WEIGHTED AVG	0.67	0.67	0.65	403

5.2.2. RandomForestClassifier() Hyperparameters Search

As for RF classifier, we have decided to create a soft- and hard-voting classifying forest full of randomly voting trees. Soft-voting (SVC) is going to vote according to the probabilities made by various classifiers on predictions whereas hard-voting (HVC) is going to vote according to the majority voting (on the mode of predictions, that is they take the weighted majority of pred)(Kumar, 2020). The results are quite satisfying:

- VC soft accuracy score is:0.6476426799007444 *100 = 64.76426799007444%
- VC hard accuracy score is: 0.674937965260546 *100 = 67.4937965260546%

6. Discussion

Next are the following points summing up the results' characteristics in terms of differences and accuracy relevance.

6.1. Logistic Regression

The logistic regression classifier did not perform that well overall when instantiating it with no Hyperparameters. It barely aced it with a 57%. Perhaps because the dataset was not big enough, and we did split it into a 80% training size and a 20% test size. But actually the partitioning was quite good overall.

6.1.1. LogReg() instantiation

Results from the section 5.1.2 show us a 61% accuracy, which is quite good. Without choosing any particular hyperparameters, we have managed to obtain satisfying results. But let's see what a more accurate instantiation gives us.

6.1.2. LorgReg() Hyperparameters Search

Table 1 shows the overall accuracy score of the Logistic regression classifier on the test set. It does have a better precision on the predicted value 1 than on 0, with a 67% of false-positive detection averaged over prediction 1, and a weighted average of 0.62. By looking at the F1- score - which kind of makes the compromise between precision and recall - we can have a glimpse of our model's precision. Indeed, it provides an analytical result of the ratio between the positively-predicted values (that is, the true positives TP) and the erroneous ones (that is, False Positive FP and False Negative FN). The recall (number of correctly attributed features to a given class) is excellent for class 0 though as opposed to class 1. The F1-score though is of 0.44 on average, so that is not so good, as the model is likely to make close to twice a prediction of TP as well as FP. It will detect roughly 1 out of 2 times a TP like a FP.

6.2. Random Forest

The RF classifier did pretty well on hard and soft voting (see section 5.2.2), as well as Table 2 shows,where there were quite good at predicting the ”1” and ”0” with roughly the same precision (0.64 and 0.65, respectively), e.g. the potability and non-potability. However they were better at predicting the non-potable characteristic of the water, which yields a more satisfying performance.

6.2.1. RandomForestClassifier() instantiation

The somewhat different results of 58.39 % on Validation and 67.0 % on Test indicate that the accuracy is higher as they are for LogReg. However, let's see if we can enhance the results ...

6.2.2. RandomForestClassifier() Hyperparameters Search

Hyperparameters fine-tuning has helped us come up with Table 2. As opposed to what Table 1 provided us, precision is on average higher for RF, as is the F1-score, reaching a 0.62 weighted average. Plus, what is relevant to notice, is the recall, of 0.85 for 0 and of 0.43 for 1, which means that RF is overall and on average better at correctly attributing features to a given class (recall) for both potable (1) and non-potable(0).

In addition to that, the 0.65 weighted average F1-score precision tells us we can trust RF more than Logistic Regression.

6.2.3. Feature importance

When looking at the feature importances (figure 2), we can see that pH and Sulfate concentration seem to play a role in predicting the water potability. It could be of significance to conduct for instance a regression of the predicted variable (0 and 1) according to the most significant ones, e.g. pH and Sulfate for instance, that could weigh more in the determination.

7. Conclusion

In light of the aforementioned points presented and discussed above, we can state that, in the frame of our project, the binary classification algorithm RF classifier did perform on average better than what the Logistic Regression classifier did. By looking at their F1-scores, RF did a 0.65 on weighted average, against 0.44 for LogReg. What is more, the RF's voting trees parametrization has helped us reach almost 68% accuracy on test set.

8. References

- Kadiwal, A. (2020). Water Quality. https://www.kaggle.com/datasets/adityakadiwal/water-potability
- Kumar, A. (2020, septembre 7). Hard vs Soft Voting Classifier Python Example. Data Analytics. https://vitalflux.com/hard-vs-soft-voting-classifier-python-example/

Author: Etienne Delaloye

Machine Learning For Earth and Environmental Sciences Master 2022, Autumn, UNIL DATE: December 23rd 2022