Francisco Javier Ortín Cervera, 204066IAPM, fortin@taltech.ee

ITI8565 Machine learning. First assignment

Ortín Cervera, Francisco Javier

I. PROJECT DESCRIPTION

Water potability is a very important quality of the water, essential for human being. Sometimes is not so easy to say if it is safe to drink it or not. There are many parameters that influence its potability as described in the data set section of this paper.

The goal of this project is to find a model that can predict accurately if the water is potable or not. Part of the goal is to discover what are the parameters of the water that contribute most to the potability of the water and explore if there is an efficient way of safely say the water is potable with the minimum number of parameters. Such investigation requires inquire into different machine learning methods that can help find such an answer better than simple linear regressions and probability studies.

II. DATASET

The file containing the data can be downloaded from the Kaggle web site containing a data set about water potability [1].

It is a csv file containing data about different parameters of the water:

- **ph**: pH of 1. water (0 to 14).
- **Hardness**: Capacity of water to precipitate soap in mg/L.
- **Solids**: Total dissolved solids in ppm.
- **Chloramine**: Amount of Chloramines in ppm.
- **Sulfate**: Amount of Sulfates dissolved in mg/L.
- **Conductivity**: Electrical conductivity of water in μS/cm.
- **Organic**_carbon: Amount of organic carbon in ppm.
- **Trihalomethanes**: Amount of Trihalomethanes in μg/L.
- **Turbidity**: Measure of light emiting property of water in NTU.
- **Potability**: Indicates if water is safe for human consumption. Potable -1 and not potable -0

Seeing the histograms of each feature (figure 1) we realize they are normally distributed.

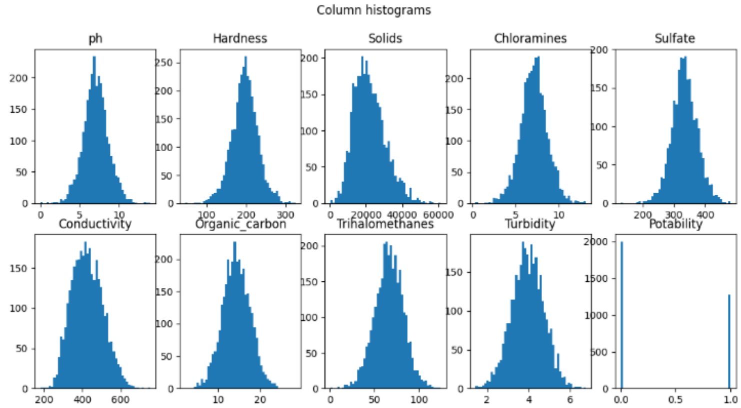

Figure 1. Original data distributions

Also, if we calculate the correlation matrix there is no significant correlation between any of the features as can be observed in figure 2:

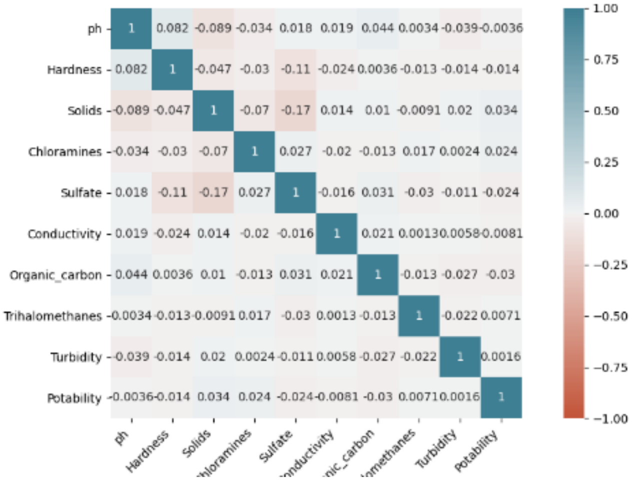

Figure 2. Correlation's plot

Understanding the previous plots, we can eliminate the hypothesis that some 2 parameters can be related and therefore the number of dimensions of the dataset can be reduced. This is not the case; they are all independent.

III. DATA CLEANING AND NORMALIZATION

Doing a first analysis of the data there is an obstacle to be resolved before using the data for developing any estimator. There are several missing entries in the data set. In table 1 can be seen which the feature values are missing.

Table 1. Not-a-number values

Feature	**Nan values**
ph	491
Hardness	0
Solids	0
Chloramines	0
Sulfate	781
Conductivity	0
Organic_carbon	0
Trihalomethanes	162
Turbidity	0

The different Nan values have been filled with the mean of the feature. Other alternative would be to interpolate the values but since the is no relation between the features there are no 2 variables to interpolate. Another option would be to eliminate the rows with Nan values but that means getting rid of 491+781+162=1434 elements (43%). Even though is a big portion of the data set this option has been explored in the implementation of a neural network.

IV. FEATURE SELECTION

To find which are the values that predict that contribute the most to the variance of the data a Principal Component Analysis has been performed. Turns out these are the 3 parameters that contribute the most: *ph*, *hardness*, and *solids*.

These three features alone explain 31.34% + 12.20% + 11.68% = 55.22% of the variance.

But as previously discussed there are no features that can be discarded due to redundancy.

V. CLUSTERING

Clustering is not an indispensable task in this problem but it can give an insight of the goodness of the data the same way the correlations analysis did. For that the SSE of different cluster numbers using k-means have been measure and plot like shown in figure 3.

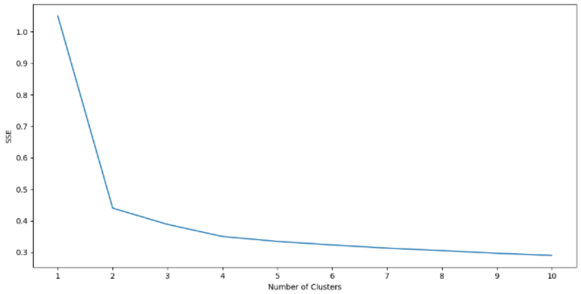

Figure 3. SSE values for clustering with k-means.

A better option is trying to calculate the silhouette coefficient for the different cluster numbers. And it works parallelly to the knee method for this data set as it can be seen in figure 4.

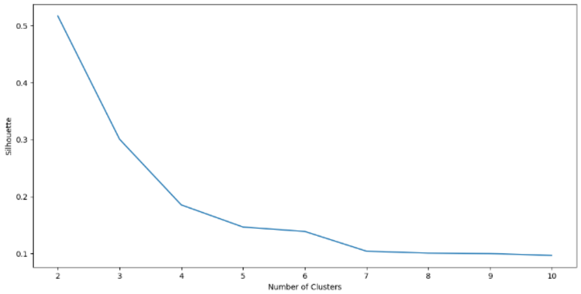

Figure 4. Silhouette coefficients for k-means clustering.

While the knee method gives us cluster number of k=3 but with the silhouette coefficient calculation is perfectly clear that is still 2 (silhouette coefficient 0.968). This clustering just confirms the data is good enough to be classified in 2 classes and not more.

VI. DECISION TREE CLASSIFIER

Decision tree classifier has been selected for supervised learning. It seemed the right classifier for binary classification.

Since it is a relatively small dataset it can be fit quite extensively so that arriving to high depth it gets a 100% accuracy and F1-score.

In figure 5 can be observed the decision tree until depth 4 and 8 leaves.

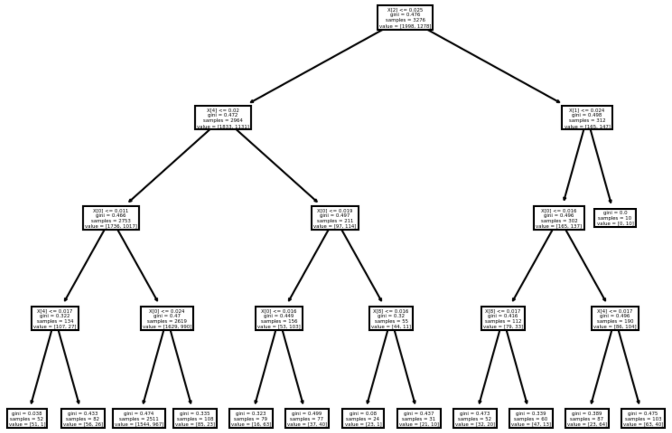

Figure 5. Decision tree until depth 4.

Also, it has been analysed how is the performance growing for the growing maximum depth as it can be seen in figure 6.

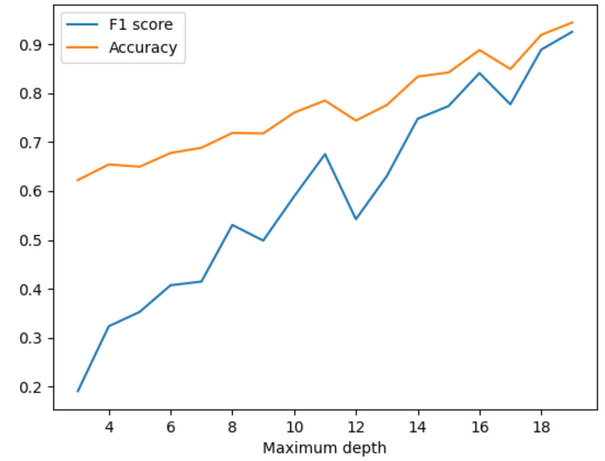

Figure 6. Evolution of decision tree accuracy and F1-score given a maximum depth.

VII. NEURAL NETWORK

As an unsupervised learning method, a neural network has been implemented with the architecture represented in figure 8.

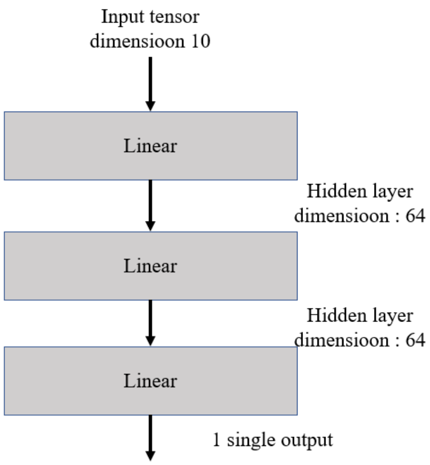

Figure 7. Neural network architecture

For the activation function the linear rectifier unit has been used. With the intention of avoiding overfitting a dropout probability has also been added after the hidden layer. Different probabilities have been tested with a very significant change in the output so finally it is set to 0.2.

In table 2 can be seen the confusion matrix after training the neural network during 300 epochs. Also, table 3 contains a more detailed overview of the performance of the same model.

Table 2.Confusion matrix of the neural network classifier

	Potable	Not Potable
Potable	558	45
Not Potable	331	49

Table 3. Classification report of the neural network classifier

	Precision	Recall	F1-score	Support
0	0.63	0.93	0.75	603
1	0.52	0.13	0.21	380
Accuracy			0.62	983
Macro avg	0.57	0.53	0.48	983
Weighted avg	0.59	0.62	0.54	983

Figure 8 shows the steady growing the model performance during training and posterior evaluation. But as it can be seen it does not grow over 70% accuracy.

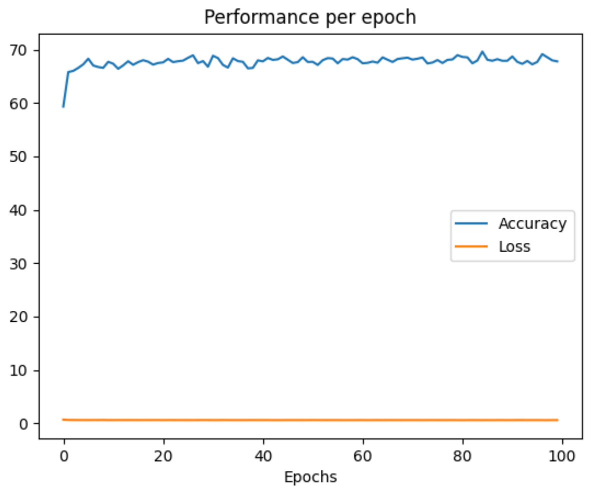

Figure 8. Performance measured in accuracy of the neural network classifier.

VIII. CONCLUSIONS

After implemented different methods we have achieved a 100% accuracy in the supervised learning method though it would not be feasible to use it in a larger data set. As well as it seems that it might be not so accurate outside the test set.

On the other hand, the results are not so promising but the good new are that it fails mostly when recognizing that water is safe to drink, since the recall for the 0 label (not potable) is higher than the recall for the 1 label (potable). That can also be observed in the confusion matrix. It mistook potable water for not potable quite often. Something that has not been mentioned at the beginning and that can influence this result is the fact that the sample is not balance. There are more 0 labels than 1.

CODE

All the code can be found in the git lab repository: https://gitlab.cs.ttu.ee/fortin/iti8565-ml-finalproject

LIBRARIES USED

Most of the algorithms for building the models and transforming the data into something more malleable has been done using several Python libraries.

This are the different libraries that have been used:

From the standard python libraries:

- multiprocessing
- copy
- sys

Libraries that need to be installed apart using pip installer:

- matplotlib
- pandas
- sklearn
- seaborn
- pandas
- numpy
- kneed
- torch

REFERENCES

[1] “Water Quality | Kaggle.” https://www.kaggle.com/adityakadiwal/water-potability (accessed Jun. 01, 2021).

[2] “Feature Selection For Machine Learning in Python.” https://machinelearningmastery.com/feature-selection-machine-learning-python/ (accessed Jun. 06, 2021).

[3] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/ (accessed Jun. 04, 2021).

[4] “Welcome to kneed’s documentation! — kneed 0.6.0 documentation.” https://kneed.readthedocs.io/en/stable/ (accessed Jun. 05, 2021).

[5] “K-Means Clustering in Python: A Practical Guide – Real Python.” https://realpython.com/k-means-clustering-python/#how-to-perform-k-means-clustering-in-python (accessed Jun. 04, 2021).