Abstract

Our chosen problem is to predict water potability based on certain measurable metrics. The dataset came from Kaggle and seems to be a comprehensive set of ten metrics regarding water quality, ex: ph, hardness, solids, etc. The dataset is said to be comprised of water samples from 3276 different bodies of water. The methods we used were logistic regression, support vector machine, and neural networks. Results for the first two aforementioned models were poor at around 60% and the result for neural networks was bad at 40%. Logistic regression and support vector machine tend to provide false negatives (potable predicted as non potable), while neural nets are the opposite with false positives (non potable predicted as potable). We concluded that if this dataset needed to be used logistic regression or support vector machines were the models to use.

Intro

The problem is to accurately predict potable water from a water sample, “Access to safe drinking-water is essential to health, a basic human right and a component of effective policy for health protection”(Kaggle). The dataset is said to be comprised of water samples from 3276 different bodies of water, both drinkable and non drinkable.

The dataset is appropriate because it is sufficiently large and completely numeric, with plenty of features and only two classes. Our approaches were to use lists of different parameters and try them all to return the highest precision score, which should also have a good accuracy. This approach was very feasible for logistic regression and support vector machine but arguably terrible for neural nets, as the brute force method took 20+ minutes, even with multiple processes.

Stats

Potables

Max

ph	11.898078
Hardness	317.338124
Solids	56488.672413
Chloramines	13.127000
Sulfate	481.030642
Conductivity	695.369528
Organic_carbon	23.604298
Trihalomethanes	124.000000
Turbidity	6.494249
Potability	1.000000

Min

ph	0.227499
Hardness	73.492234
Solids	1198.943699
Chloramines	1.390871
Sulfate	129.000000
Conductivity	201.619737
Organic_carbon	2.200000
Trihalomethanes	8.577013
Turbidity	1.492207
Potability	1.000000

Mean

ph	7.113791
Hardness	195.908341
Solids	22344.922883
Chloramines	7.174395
Sulfate	332.457832
Conductivity	425.005423
Organic_carbon	14.294764
Trihalomethanes	66.581596
Turbidity	3.991254
Potability	1.000000

Median

ph	7.046549
Hardness	197.617494
Solids	21217.158596
Chloramines	7.212254
Sulfate	331.087177
Conductivity	421.099917
Organic_carbon	14.252684
Trihalomethanes	66.612984
Turbidity	4.007347
Potability	1.000000

Mode

ph	Hardness	Solids	Chloramines	Sulfate	Conductivity

Organic_carbon	Trihalomethanes	Turbidity	Potability
15.0	69.0	4.0	1.0

(Mode pt.2)

Standard Deviation

ph	1.437623
Hardness	35.301146
Solids	8891.547966
Chloramines	1.732796
Sulfate	47.446190
Conductivity	81.950982
Organic_carbon	3.257917
Trihalomethanes	16.297713
Turbidity	0.776408
Potability	0.000000

Nonpotables

```
Max
ph                 14.000000
Hardness          300.292476
Solids          55334.702799
Chloramines        12.653362
Sulfate           460.107069
Conductivity      753.342620
Organic_carbon     27.006707
Trihalomethanes   120.030077
Turbidity           6.494749
Potability          0.000000
```

```
Min
ph                  1.431782
Hardness           98.452931
Solids            320.942611
Chloramines         2.456014
Sulfate           203.444521
Conductivity      210.319182
Organic_carbon      4.371899
Trihalomethanes    14.343161
Turbidity           1.450000
Potability          0.000000
```

```
Mean
ph                  7.067201
Hardness          196.008440
Solids          21628.535122
Chloramines         7.107267
Sulfate           333.742928
Conductivity      427.554342
Organic_carbon     14.400250
Trihalomethanes    66.278712
Turbidity           3.955181
Potability          0.000000
```

```
Median
ph                  6.992004
Hardness          196.799368
Solids          20507.399647
Chloramines         7.103718
Sulfate           332.615625
Conductivity      424.479471
Organic_carbon     14.351828
Trihalomethanes    66.206116
Turbidity           3.944085
Potability          0.000000
```

```
Mode
    ph  Hardness   Solids  Chloramines  Sulfate  Conductivity  Organic_carbon
0  7.0     192.0  15825.0          7.0    334.0         414.0            14.0

   Trihalomethanes  Turbidity  Potability
0             64.0        4.0           0
```

(Ignore the two leading zeros under Mode)

```
Standard Deviation
ph                  1.659106
Hardness           30.717642
Solids           8461.108693
Chloramines         1.476577
Sulfate            36.398403
Conductivity       79.882677
Organic_carbon      3.370196
Trihalomethanes    15.931953
Turbidity           0.782984
Potability          0.000000
```

Stats Summary

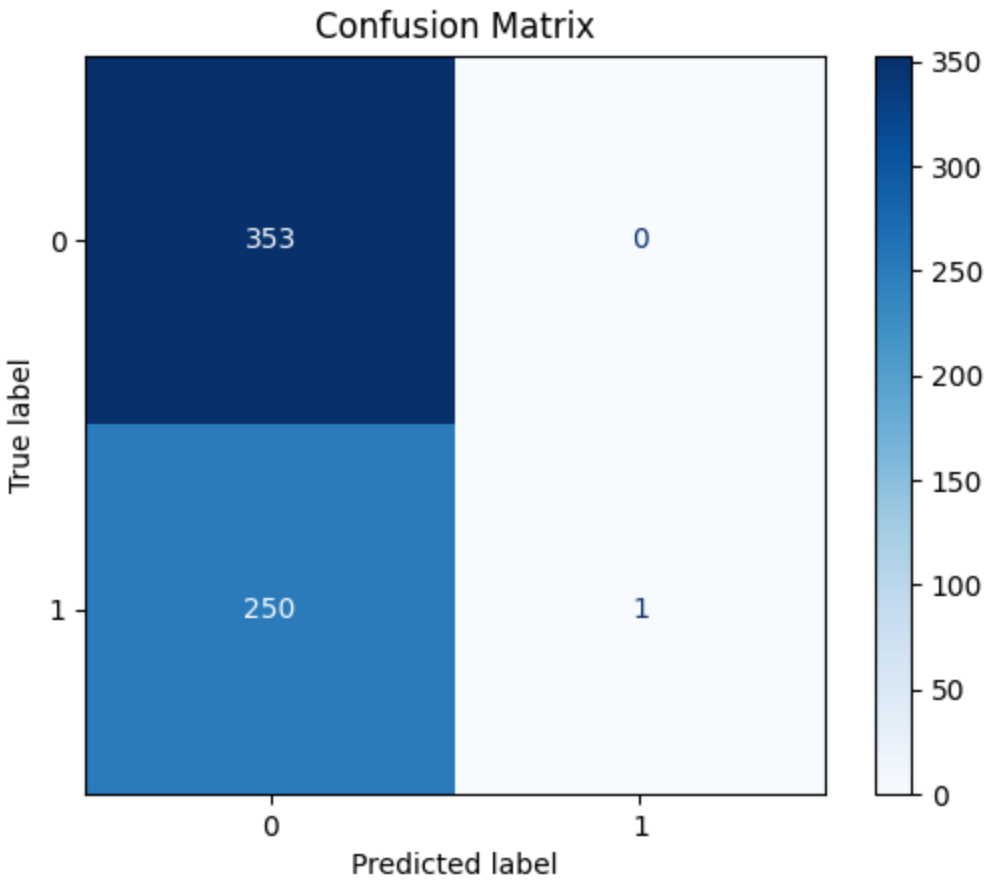
The stats show how the water potability data varies widely, and how some features might not be very useful. One of the numbers that caught my eye specifically was the standard deviation of “Solids” for both classes, which is extremely large.

Methods/Results

Logistic regression

Logistic regression is a model that generates an optimized logistic function based on the training data and the amount of restriction applied to the regression. The only parameter is the amount of restriction(regularization), but the optimization algorithms make an impact as well.

The result of logistic regression was poor, the avg acc was just 60% with the highest precision score having an accuracy of 58.6%. The model is always weighted towards predicting unsafe water correctly, but not being able to predict safe.

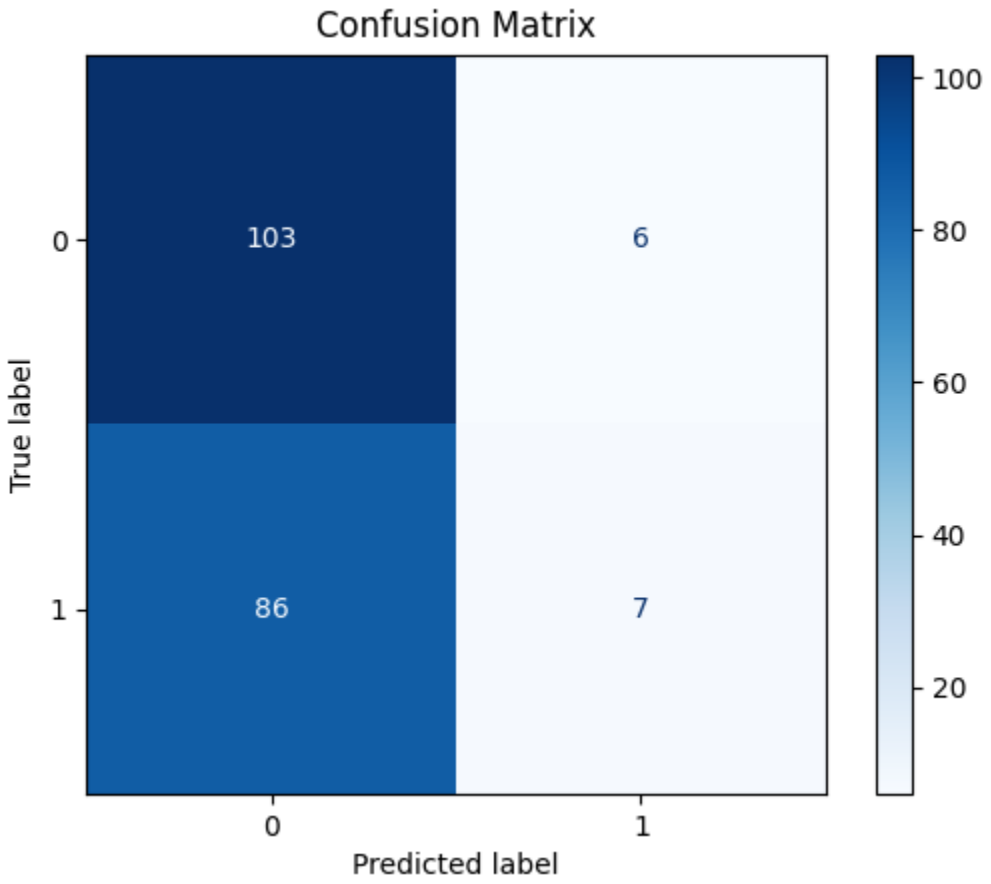


[0.7, 89, 'l2', 'saga', 0.5860 (parameters in order of training size, c-value, regression, and optimization algorithm.

SVM

Support vector machine is a model that generates an optimized function based on the training data to slice data into two sections/classes using a line/plane/hyperplane (depending on number of features). The only parameter is the amount of restriction(regularization).

The result was poor, the avg acc was just 60% with the highest precision score having an accuracy of 54%. The model is always weighted towards predicting unsafe water correctly, but not being able to predict safe water, just like logistic regression.

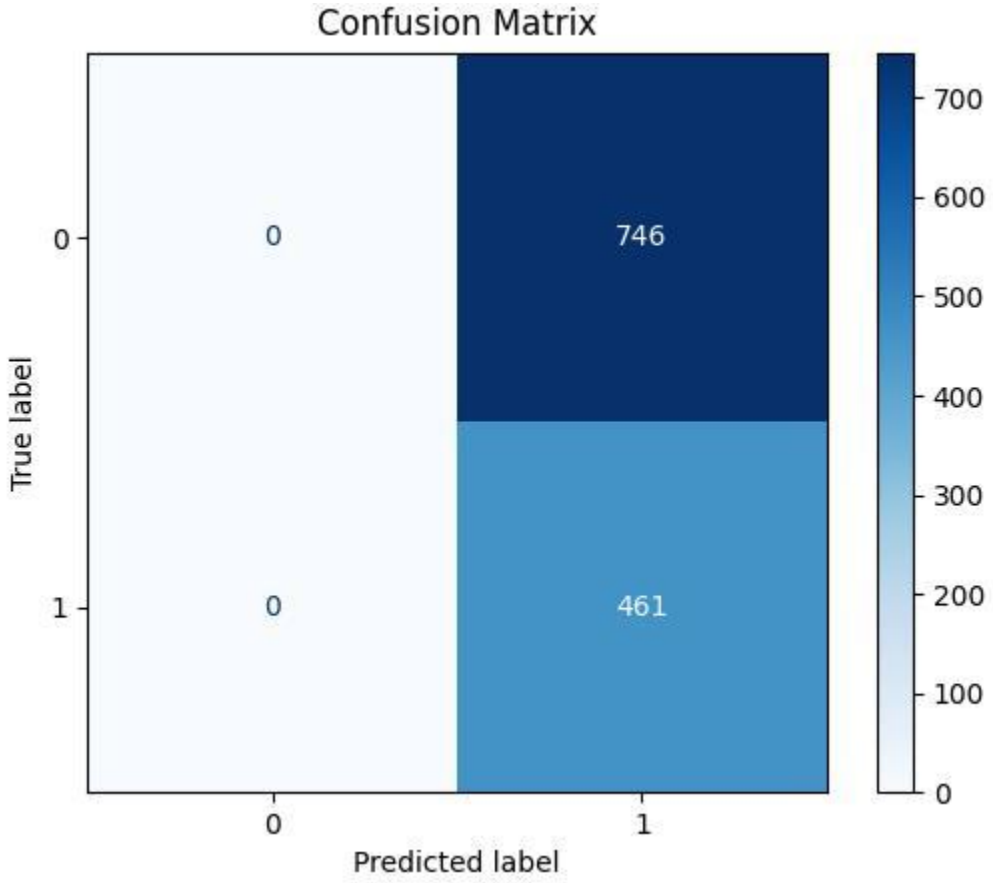


[0.9, 89, 89, 0.544 (parameters in order of training size, c-value, and gamma.

Neural Nets

Neural networks are models that generate an optimized network of weight and biases in order to accurately predict a class. Usually they have many input nodes(features) to a number of hidden layers and then to a single output layer which decides the classification. The weights are optimized from end to beginning and the initial weights are pretty much guessed/random.

The result was very poor, with an average accuracy for the best precision score being 40%. The model is always weighted towards predicting safe water correctly, but not being able to predict unsafe water, the exact opposite of the other two models.




```
[0.4, [8, 8, 8], 'logistic', '
sgd', 34, 0.38193869096934546]
```
(parameters in order of training size, hidden-layers, activation function, solver algorithm, c-value, and the resulting accuracy.

Discussion

Logistic regression and support vector machine performed sub-par from what I expected. Kaggle gives this dataset a high usability number, and on initial thought I believed that meant that the dataset was good for machine learning, but with further inspection that number only indicates how well documented the dataset is and has nothing to do with usability. The two aforementioned models had very similar results and always leaned towards predicting all/most samples as non potable.

Starting on neural nets I had hopes that it would turn out better than the other models but it under-performed even them by far. On top of being worse the model leaned towards predicting all water as potable which would be the absolutely worse outcome. All of the models were run with a combination of many parameters to attempt to find the best set. However, almost every model output different parameters every time they were run.

I believe that neural nets had a worse outcome because I could not try an extensive list of the number of hidden layers and size of hidden layers as the execution time of the program became extremely long, even with multiprocessing.

Conclusion

This problem has remained unsolved. My group concluded on extensive analysis of the features that this dataset is either very bad for machine learning or even fake, as many of the features for each class overlap and thus make the accuracies terrible.

Despite the dataset, logistic regression and support vector machine were fairly accurate in solving the problem and gave much more true negatives (potable water predicted non potable) than false positives (non potable water predicted potable). Even so, I would personally not trust whatever result a water potability tester trained from this dataset gave.

References

Water potability dataset

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html?highlight=mlpclassifier#sklearn.neural_network.MLPClassifier