

## Introduction

About 12 million tons of glass are produced in the United States each year, and according to the [Crime Museum](#), glass should always be collected at the scene of a crime. By doing so, the investigative team can pair glass samples, identify unknown glass, estimate original size and quantities or even the manner in which the glass broke. Our analysis will help such teams be able to make better predictions on what type of glass is relevant. A decision tree could be used even as a visual aid to be a highly interpretable way to classify a piece of glass. Aside from time savings and error mitigation, this would allow the teams to solve more crimes.

[Presentation](#) and [GitHub](#)

## Dataset

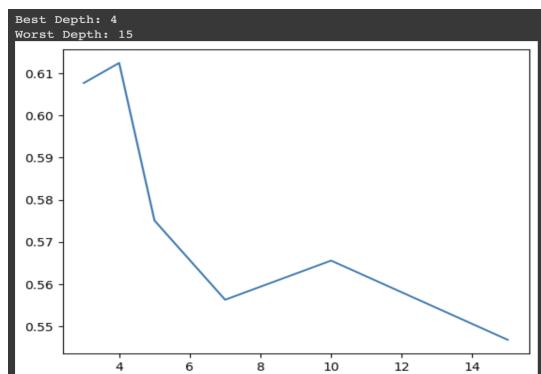
The data set is a collection of data on elemental compounds found in glass. There are over 200 samples which each have ten attributes, with the labeling attribute being what type of glass it is. Eight of the ten attributes represent chemical makeup of the glass sample.

## Analysis Technique

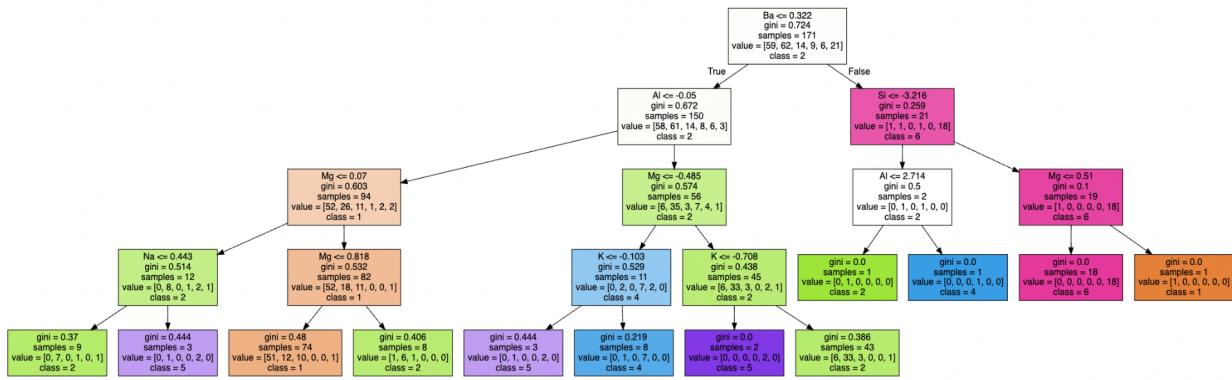
We used two methods to analyze the data. The first is a decision tree, which was used due to its high level of interpretability. We could print out the resulting decision tree on a poster for lab technicians to reference easily. They wouldn't have to know anything about the data or its structure, or even glass for that matter. The second type that we used was a neural network, which can be thought of as a multidimensional decision tree where boundaries and where to split decisions are discovered by the algorithm in higher dimensions rather than by the data themselves.

## Results

We found that both the decision tree and the neural net performed very well at predicting what type of glass the sample was. For our decision tree, we found that having it be 4 decisions deep let the model predict with 74% accuracy and 75% precision. This means that a forensic analyst could use our tree and without any subject matter experience or technical data know-how, have just 26% false classifications. We tested decision trees with different depths and found the one with depth 4 to perform the best and the one with depth 15 to perform the worst. The accuracy vs depth plot is given below.

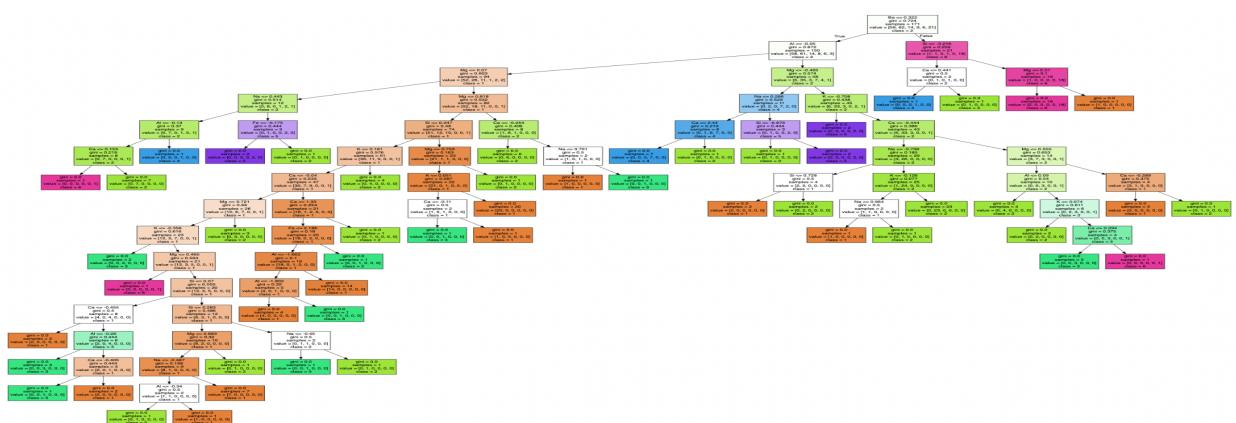


## Depth 4 decision tree :



	precision	recall	f1-score	support
1	0.62	0.91	0.74	11
2	0.83	0.71	0.77	14
3	0.00	0.00	0.00	3
5	1.00	0.25	0.40	4
6	0.50	1.00	0.67	3
7	1.00	1.00	1.00	8
accuracy				0.74
macro avg	0.66	0.65	0.60	43
weighted avg	0.75	0.74	0.71	43

## Depth 15 decision tree:

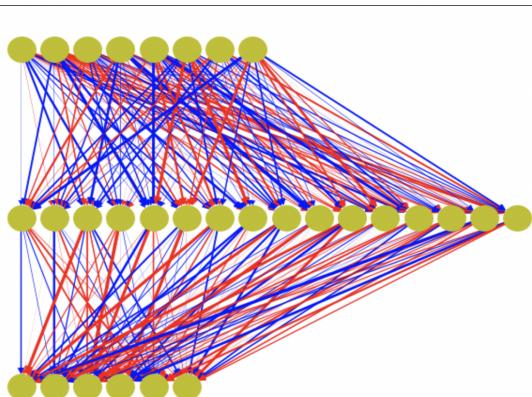


	precision	recall	f1-score	support
1	0.57	0.73	0.64	11
2	0.73	0.57	0.64	14
3	0.50	0.67	0.57	3
5	1.00	1.00	1.00	4
6	1.00	0.67	0.80	3
7	1.00	1.00	1.00	8
accuracy			0.74	43
macro avg	0.80	0.77	0.78	43
weighted avg	0.77	0.74	0.75	43

For the Neural network we built three models and the corresponding results are given below.

Model1 : 1 hidden layer, hidden nodes = [16]

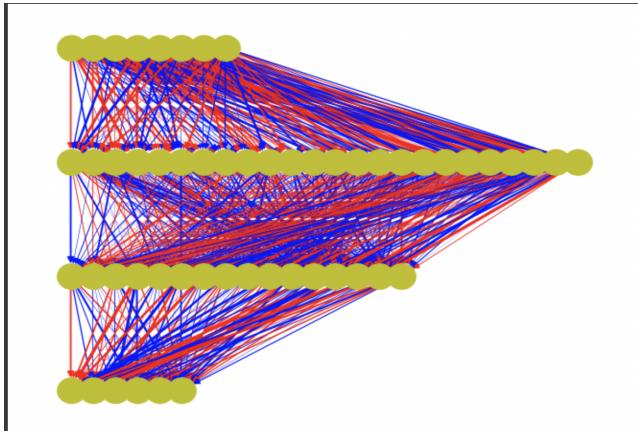
Accuracy : 0.72



	precision	recall	f1-score	support
1	0.62	0.91	0.74	11
2	0.77	0.71	0.74	14
3	0.00	0.00	0.00	3
5	0.75	0.75	0.75	4
6	0.50	0.33	0.40	3
7	0.88	0.88	0.88	8
accuracy			0.72	43
macro avg	0.59	0.60	0.58	43
weighted avg	0.68	0.72	0.69	43

Model2 : 2 hidden layers, hidden nodes = [24,16]

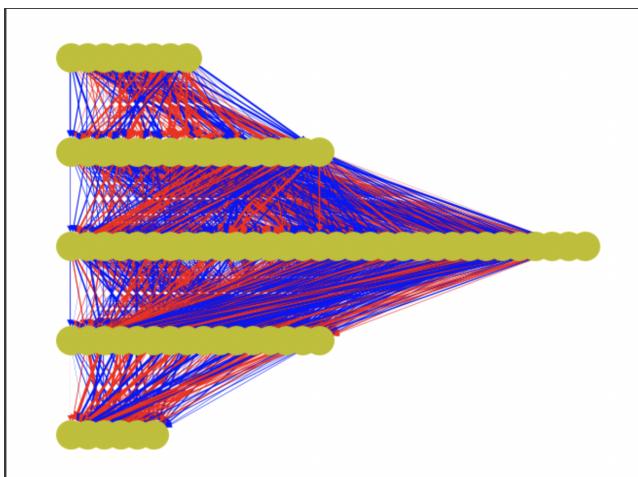
Accuracy : 0.67



	precision	recall	f1-score	support
1	0.54	0.64	0.58	11
2	0.70	0.50	0.58	14
3	0.40	0.67	0.50	3
5	1.00	0.75	0.86	4
6	1.00	0.67	0.80	3
7	0.80	1.00	0.89	8
accuracy			0.67	43
macro avg	0.74	0.70	0.70	43
weighted avg	0.71	0.67	0.67	43

Model3 :3 hidden layers, hidden nodes = [16,32,16]

Accuracy : 0.72



	precision	recall	f1-score	support
1	0.75	0.55	0.63	11
2	0.73	0.79	0.76	14
3	0.40	0.67	0.50	3
5	0.60	0.75	0.67	4
6	1.00	0.33	0.50	3
7	0.89	1.00	0.94	8
accuracy			0.72	43
macro avg	0.73	0.68	0.67	43
weighted avg	0.75	0.72	0.72	43

## Technical

For our data, we had little cleaning to do. The primary change that we made was to limit the number of feature columns. That means we excluded the refractive index data which was highly correlated with Calcium. This had the added benefit of limiting the scope of the lab work to strictly material testing. While commercial refractometers are cheap (around \$100), industrial or lab grade ones are thousands of dollars. Most labs already have a mass spectrometer, thus making our results even more applicable. The elements were reported as a composition percent, but with decision trees we don't need to account for differences in scale, so we didn't standardize the data.

We then made eight different decision trees, and analyzed the ones which performed the worst. We also kept the worst performing one as a comparison tool. The models differed in the depth of the decision tree and in short, we found the following: too shallow of a tree didn't filter enough of the samples fast enough, whereas too deep of a tree made distinctions in the data that weren't

present in the real world. Similarly, we built three different neural nets, varying the number of layers as well as the number of nodes per layer. Important features have more average weights assigned to them. These attributes are also used for splitting the data in the decision tree algorithm. Finally, we used the packages provided by `sklearn.metrics` to analyze the performance of each model. For the decision tree, we used a receiver operating characteristic curve (ROC curve) to illustrate the differences in how the trees performed for each type of glass.