Preliminary Information

For our project, we decided to compare different Machine Learning methods' performance. We will take a simple data set and apply the various ML techniques and see which yields the best results. Given a simple data set, our hypothesis is that a simpler technique, SVC or Decision Tree, will outperform more complex methods (such as a NN). Simple solutions work well with simple problems.

Method

We will perform 40 training sessions using three different ML techniques, Decision Tree, SVC, and a Neural Network. After the 40 sessions we will compare average train time, highest test scores, and overall ease to find strong parameters. Finally, we will decide which method was most effective for our specific problem.

Data

Our data is a math class student performance set (a link to this data can be found here). We will only evaluate the cumulative term grades, not the individual semesters. We slightly modify this data by changing the resulting grade from a 0-20 score to a pass/fail boolean. We assume a standard American grading curve (0.6 is the passing mark), 0-11 is a failure, 12+ is passing.

Decision Tree

When we started training decision tree models, we noticed a few things: there was an overfitting issue, training was very fast (~1 second), and F1 scores weren't too terrible with how little we changed the hyperparameters. Also, many of the hyperparameters for the decision tree model are continuous, meaning we could try basically any number as long as it fits the type (int, float, etc.). These properties of the decision tree model led us on a long string of trial and error to find just one model that could score more than the others. Fortunately, the fast training time allowed us to rapidly try new configurations. At training session #40, we got a model that had a training F1 of 0.696 and testing F1 of 0.746, beating the best neural network and SVM. We were definitely surprised that the decision tree model was able to perform better than a neural network, but with data as simple as this, it seems that the decision tree is right at home.

Neural Network

Unsurprisingly, configuring a well-scoring neural network was extremely difficult. Between network structure, parameters, and activation functions, it was quite tedious to get any viable results. On occasion, the changing of a parameter would result in an excess of 15 minutes worth of training. While the typical session took <5 seconds, the occasional time heavy attempt added up to a very grueling process. For these reasons, neural networks have very poor ease of use. Obviously this is coming from a machine learning novice, but even with simple data the network was difficult to work with. As far as scoring is concerned, the neural network was able to achieve a respectable training F1 of 0.8, the testing F1 fell short at only 0.69. This is significantly better than random, but not quite consistent enough to warrant praise. Not to mention the average time to train is significantly higher than both the SVC and Decision Tree at 26.85 seconds.

SVC

The first thing that immediately stuck out to us about the support vector machine was that it didn’t have nearly as many possible configurations as a decision tree or neural network. Some hyperparameters had a few discrete choices, and other hyperparameters would only affect that model depending on which choice was made. This led us to stopping training before 40 sessions; there simply weren’t any more options available to continue training new models. Even with trying almost all possible configurations, we were not able to beat the decision tree in terms of F1 score. The training time was very similar, just a couple seconds, but it had a much more volatile result. In fact, one configuration was so bad that it got a training score of 0.015, but another one got a training score of 0.983. By the time we stopped training SVMs, the best score we found was on the second session, with a training F1 of 0.715 and testing F1 of 0.637, and we continued to get this score multiple times throughout the process. We’ve seen that SVC has the potential to get high scores (based on training F1), but perhaps the limit for this simple dataset has been reached.

Conclusion

Our hypothesis was partially correct. A decision tree turned out to be the most effective technique for the given problem, though the SVC fell below the Neural Network. Does this mean that Decision Trees are the best for all situations? Of course not. More complex classification would likely require the power that neural networks offer. Similarly, mid-range complexity problems would likely benefit greatly from an SVC. Certain ML techniques are better suited for certain situations. While a decision tree worked well for us, it doesn’t mean that it is the ideal solution to every problem.

Classification methods on Simple Data

Dylan Sharp & Ian Shakespeare

Goal

In class, we discussed three methods of classification: Decision Tree, Support Vector Machines, and Neural Networks. Each of which are very different ways of classifying data.

Does one have a clear advantage over the others?

Our goal is to identify what is the optimal classification method when working with simple data.

Setup

- We define a method as “optimal” based on three categories:
  - Ease of Use i.e. how hard was it to find ideal parameters
  - Training time, the total amount of time spent training
  - Scoring, both in training and testing
- Method of Comparison:
  - We gave each method 40 training sessions
  - In each, we recording train time and scores
  - Our end results will be based on max score and average train time
  - Ease of Use will be assessed from a predominantly subjective point of view, though number of trains to find ideal parameters will factor in
- Now, for the data…

Data

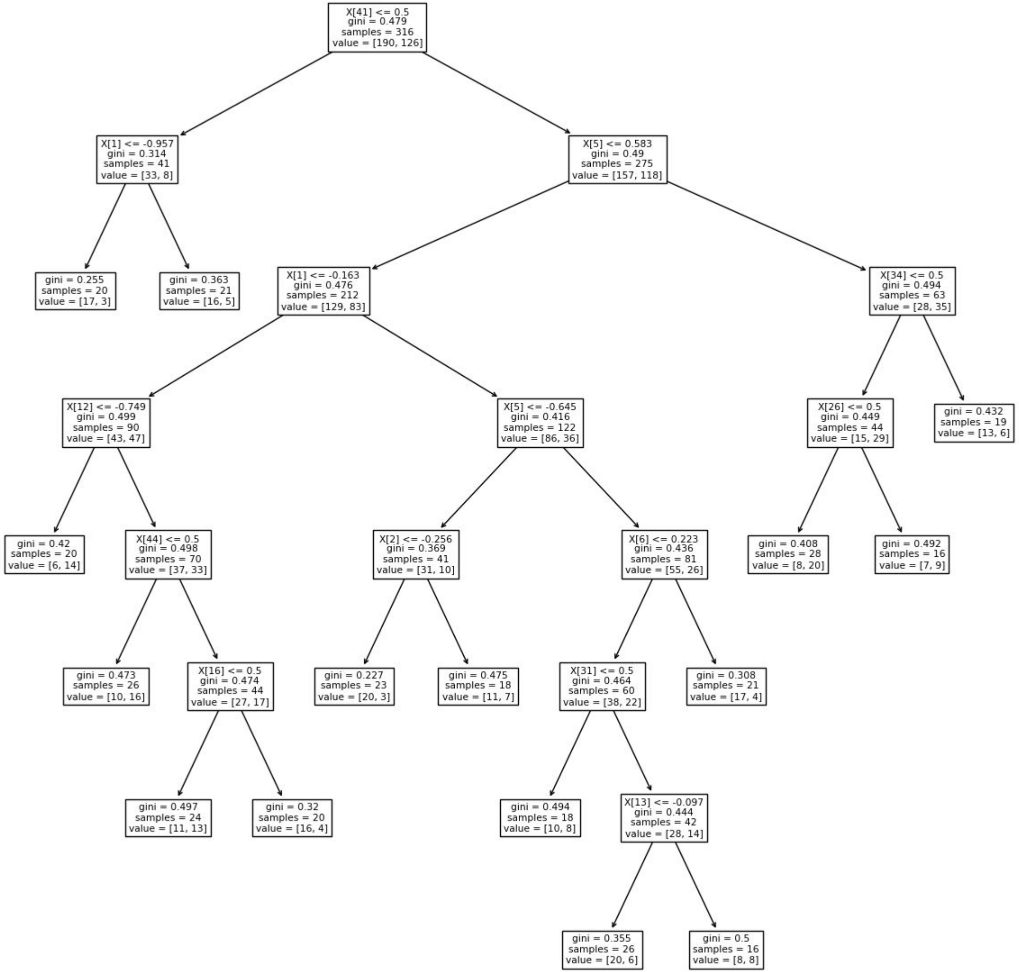
Our simple data set will be the Student Performance Data Set (collected by Cortez, https://archive.ics.uci.edu/ml/datasets/student+performance). This data contains 30 features, ranging from student age to family relationship. The label columns are an end grade between 0-20. For our project we considered the standard grade scheme and evaluated all grades above 12 as passing and everything below as failing, meaning our agent will classify into either Pass or Fail. We will only evaluate end-of-year grades.

Decision Tree

- Average Training Time: ~1 second
- Ease of Use: 6/10
  - Many potential hyperparameter settings
  - Trial and Error
  - Runs fast
- Highest Training F1 Score: 1.0
- Highest Testing F1 Score: 0.746

Decision Tree

- Depth: 8
- Leaf Nodes: 16


X[41] <= 0.5 gini = 0.479 samples = 316 value = [190, 126]
X[1] <= -0.957 gini = 0.314 samples = 41 value = [33, 8]
X[5] <= 0.583 gini = 0.49 samples = 275 value = [157, 118]
gini = 0.255 samples = 20 value = [17, 3]
gini = 0.363 samples = 21 value = [16, 5]
X[1] <= -0.163 gini = 0.476 samples = 212 value = [129, 83]
X[34] <= 0.5 gini = 0.494 samples = 63 value = [28, 35]
X[12] <= -0.749 gini = 0.499 samples = 90 value = [43, 47]
X[5] <= -0.645 gini = 0.416 samples = 122 value = [86, 36]
X[26] <= 0.5 gini = 0.449 samples = 44 value = [15, 29]
gini = 0.432 samples = 19 value = [13, 6]
gini = 0.42 samples = 20 value = [6, 14]
X[44] <= 0.5 gini = 0.498 samples = 70 value = [37, 33]
X[2] <= -0.256 gini = 0.369 samples = 41 value = [31, 10]
X[6] <= 0.223 gini = 0.436 samples = 81 value = [55, 26]
gini = 0.408 samples = 28 value = [8, 20]
gini = 0.492 samples = 16 value = [7, 9]
gini = 0.473 samples = 26 value = [10, 16]
X[16] <= 0.5 gini = 0.474 samples = 44 value = [27, 17]
gini = 0.227 samples = 23 value = [20, 3]
gini = 0.475 samples = 18 value = [11, 7]
X[31] <= 0.5 gini = 0.464 samples = 60 value = [38, 22]
gini = 0.308 samples = 21 value = [17, 4]
gini = 0.497 samples = 24 value = [11, 13]
gini = 0.32 samples = 20 value = [16, 4]
gini = 0.494 samples = 18 value = [10, 8]
X[13] <= -0.097 gini = 0.444 samples = 42 value = [28, 14]
gini = 0.355 samples = 26 value = [20, 6]
gini = 0.5 samples = 16 value = [8, 8]

Support Vector

- Average Training Time: ~2 seconds
- Ease of Use: 8/10
  - Not many hyperparameter options
  - Less Trial and Error
  - Runs fast
- Highest Training F1 Score: 0.983
- Highest Testing F1 Score: 0.637
- 33 training sessions total. Highest was found on the 2nd session.

Neural Network

How does the neural network shape up?

Average Train Time: 26.85 seconds

Highest Training F1: 0.88

Highest Scoring F1: 0.69

Ease of Use: 2/10

It should be no surprise that making an optimal Neural Network can be rather difficult. The Neural Network's F1s were lower than the Decision Tree, but better than random!

Conclusion

Undoubtedly, Decision Tree is the best learning method for our data set. It’s ease of use was middle of the pack, but the average training time and test score is significantly higher than the other two methods.

Does this mean the Decision Trees are the end-all-be-all machine learning technique? No, probably not. The first immediate issue with the Decision Tree model is is frequently overtfits to the training data. Also, Decision Trees are best when predicting discrete values, and if you want to predict continuous values you have to define discrete categories, thus losing data. If this is the case, use a neural network or SVM instead, but if there is enough time and computational power, a neural network should have a higher ceiling.