Supervised Learning Analysis

Travis Latzke

1 Introduction

For many machine learning classification problems, the goal of the problem is to build a model that achieves as high of an accuracy score as possible. For example, models that classify handwritten digits are often evaluated by how accurately they can classify a certain number of digits, if the test set has an even distribution of digit labels. If the labels in the test set are not evenly distributed, then accuracy might not be the best metric to evaluate different models for the problem. For instance, consider two models, A and B that are trained on a test set C. Assume that model A achieves an accuracy rate of 99% on the test set and model B achieves an accuracy rate of 98%. However, also assume that model A never classified the digit '0' correctly, but model B classified the digit '0' correctly 98% percent of the time. This scenario could happen if there is a small number of zeros in the test set compared to the number of other digits in the test set. In this case, the accuracy metric might indicate that model A is better than model B, even though model B is an all-around better model than model A.

Fortunately, different metrics can be used to account for data imbalances like the case above. The AUC_ROC metric can be used be in binary classification to measure how good a model can distinguish one class from another class. AUC_ROC can be extended to multiclass problems by generating an ROC curve for each label and then by calculating the area under each curve. Then the values can be averaged, so the overall model can be evaluated. If this type of evaluation was done on model A and model B as explained above, then the AUC_ROC metric would have likely deemed model B to be a better model because AUC_ROC is better at capturing a model's all-around performance than accuracy is. The AUC_ROC metric also works well in this case because the penalty for incorrectly classifying a digit is the same for all the possible labels. This is not the case for many other types of problems, particularly in finance or health care.

2 Classification Problems

In finance, one important classification problem involves classifying a loan as "good" or "bad." A good loan is defined to be a sum of cash that is loaned to a constituent over a period of time and the constituent pays the full amount back to the lender, plus any interest accrued over that period of time. A loan is defined to be "bad" when the constituent fails to pay the borrowed amount back. The problem is important because financial companies could save millions of dollars if they're able to better predict if a given loan will be good or bad. Therefore, it is important to minimize the number of times a model predicts that a constituent can repay a loan, when in reality the constituent is incapable of doing so. This case is often referred to as being a "false negative" example because the model falsely labeled the case as negative, when the model should have labeled the case as a positive (positive = bad loan, negative = good loan).

One way to minimize the number of 'false negatives' is to teach the model to always classify a case to be positive. However, if financial companies always labeled a loan as "bad" then they would never make money because they would never issue any loans. Therefore, the goal of the model should be to minimize the number of "false negatives" while maintaining a decent ability in classifying an accurate number of "true negative" examples. This phenomenon is generally referred to as a precision-recall trade-off and the trade-off threshold is usually determined by an expert in the domain field, which is finance in this case.

Classification problems also occur in other fields, such as health care. One problem involves building models to identify whether a patient has cardiovascular disease. This is an important problem because it could help patients receive the necessary medications and medical advice to persevere through the disease. Similar the financial loan problem, it is important to minimize the number of "False negative" predictions made by the model (negative == no disease, positive == disease), so that people who need treatment are not misdiagnosed. Again, we can't just teach the model to always classify an example to be a positive case because the hospitals reputation would be at stake, and some patients would be given medications that they don't need. Therefore, this problem should be evaluated by generating precision-recall curves and then letting a domain expert set the optimal threshold point for the trade-off between false negative examples and false positive examples.

3 Data Collection and Processing

Choosing and collecting data is an important step for every type of machine learning problem. In the case of supervised learning, it is crucial to label the data correctly and collect as many useful features as possible, so that the model can be properly trained. Fortunately, there are several public data sets available that contain financial and health care data that can be used to solve the problems mentioned in the previous section.

The financial loan data set was used to train various classification problems to predict whether a loan will be "good" or "bad." The data set consists of 2,260,668 example loan cases, with each case specifying whether the loan was good or bad, along with other key features, like the amount of the loan, duration of the loan, income of the individual who the loan was issued to, etc. However, many cases were missing values for important features, so those cases were filtered out of the data set through a pre-processing step. After filtering the examples with missing data, there were 17,254 cases left in the data set. 14,602 of the cases were negative cases and 2,652 were positive cases, so the dataset was significantly imbalanced.

The cardiovascular data set contained 70,000 records with each record containing medical information about a patient and a binary field indicating whether the patient has cardiovascular disease. In this dataset, there were no missing values for any of the features, so no filtering was done. The dataset was also fairly balanced, with 34,979 positive cases and 35,021 negative cases. Both datasets were normalized, so that the features in each dataset took on values ranging from 0 to 1.

Both datasets were also split into separate training and testing datasets. Each test set contained 20% of the examples from the original dataset, while each training set contained 80%. The training sets were used to train several models with different underlying learning algorithms. The models were then evaluated against one another to see which model performed the best. All models were trained under 3-fold cross validation, and scored with the F1 metric. The F1 metric was used because it's the harmonic mean of precision and recall, so a higher F1 score usually indicates a better precision-recall trade-off, which is the useful plot for domain experts to use in choosing an optimal threshold.

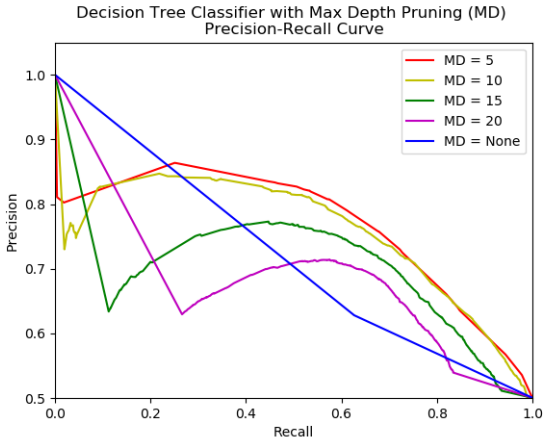

Fig. 1: *Precision-Recall curve for multiple decision tree classifiers trained on the cardiovascular dataset with different max depth pruning parameter.*

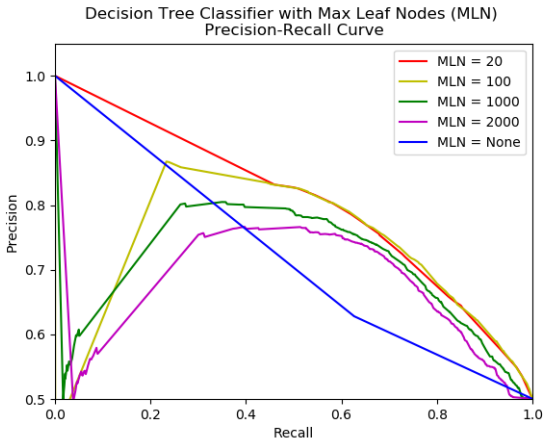

Fig. 2: *Precision-Recall curve for multiple decision tree classifiers trained on the cardiovascular dataset with different max leaf node pruning parameter.*

4 Results

4.1 Decision Tree Results

4.1.1 *Cardiovascular Problem*

Figures 1-2 show how different pruning parameters affect the precision-recall curve for each decision tree model that was trained to solve the cardiovascular problem. Two types of pruning parameters were used to study how the performance of each model changed. The first parameter "max depth" is used to limit the height of a decision tree during the training process. The other pruning parameter that is studied is the "max leaf node" parameter. As the name suggests, the parameter limits the number of leaf nodes that the model can have. The pruning parameters are studied because they generally help cut out areas of the decision tree that have little importance and they also make the models less susceptible to over-fitting.

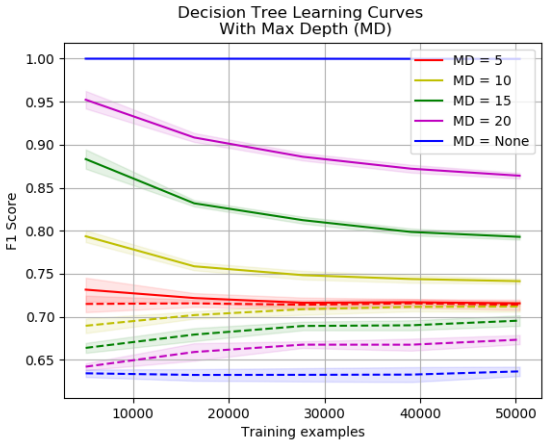

Fig. 3: *The F1 learning curve as a function of training examples for each model trained with max depth pruning.*

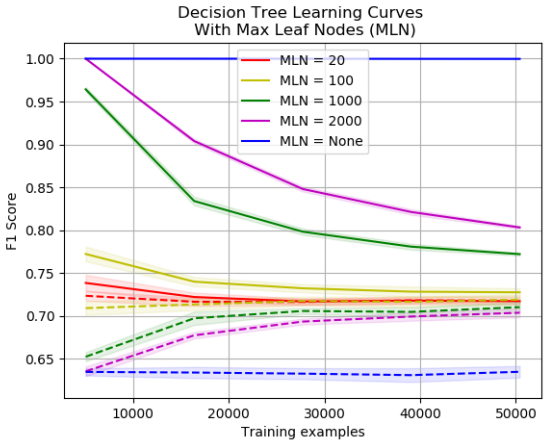

Fig. 4: *The F1 learning curve as a function of training examples for each model trained with max leaf node pruning.*

Figures 3-4 show the learning curves for the decision tree models trained with the associated pruning parameters. The solid line indicates the F1 score on the training data and the dashed line indicates the F1 score on the cross-validation data. Ideally, both F1 scores should converge as the number of training examples increase. This is evident for all models except for the model with no pruning parameters.

4.1.2 Financial Loan Problem

Similar to Figures 1-2, figures 5-6 show how different pruning parameters affect the precision-recall curve for decision tree models trained to solve the financial loan problem. Different parameter values were chosen for this problem because the underlying dataset is significantly different than the cardiovascular problem. Both plots indicate that the best precision-recall curves are formed when the parameters chosen hit a sweet spot (not too high and not too low) as seen by the yellow curve.

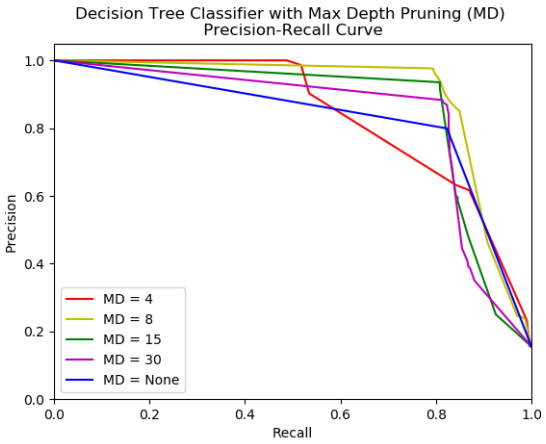

Fig. 5: *Precision-Recall curve for multiple decision tree classifiers trained on the financial loan dataset with different max depth pruning parameter.*

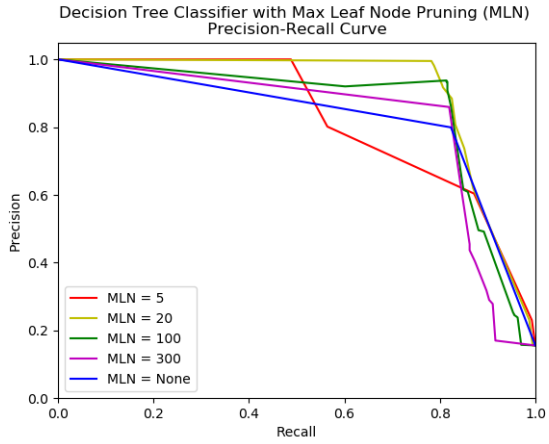

Fig. 6: *Precision-Recall curve for multiple decision tree classifiers trained on the financial loan dataset with different max leaf node pruning parameter.*

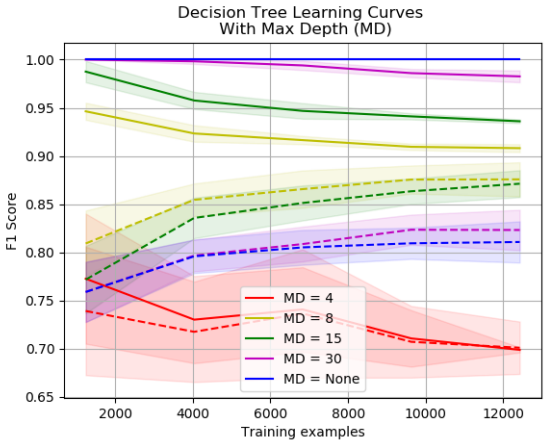

Fig. 7: *The F1 learning curve as a function of training examples for each model trained with max depth pruning.*

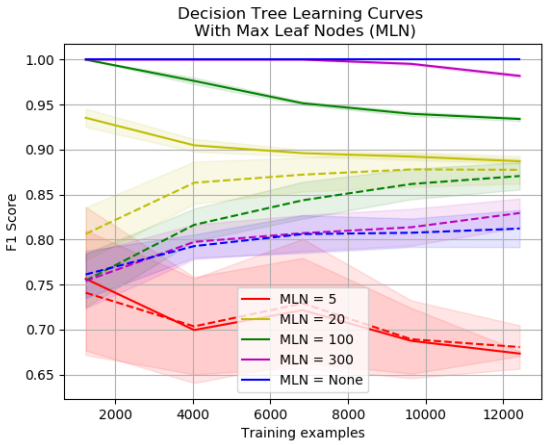

Fig. 8: *The F1 learning curve as a function of training examples for each model trained with max leaf node pruning.*

Figures 7-8 show the learning curves associated to the models in figures 5-6. Again, the solid lines indicate the F1 score from the training data and dashed lines indicate the F1 score from the validation data. Both plots show a nice convergence pattern for the model in yellow. The same

model also yielded the highest F1 score, thus showing that optimal number of max leaf nodes is around twenty nodes.

4.1.3 Decision Tree Improvements

The decision tree models above are analyzed under separate pruning parameters. The analysis could be taken further by training models with both 'max depth' and 'max leaf node' parameters and analyze if the combination of parameters could further improve the best decision tree model. However, the number of combinations to test for is quite large, so the analysis above only compared one parameter at a time.

4.2 Boosted Decision Tree Results

One common method to improve the performance of a model is to train the model with an underlying boosting algorithm. Boosting can improve a models performance by placing higher weights on the training examples that are more difficult to learn. However, the algorithm can also lead to over-fitting on certain problems and datasets, so it is important to gather experimental evidence showing how boosting affects certain models. Figure 9 shows how boosting affects the precision-recall curves for previous decision tree models with max depth pruning.

4.2.1 Cardiovascular Problem

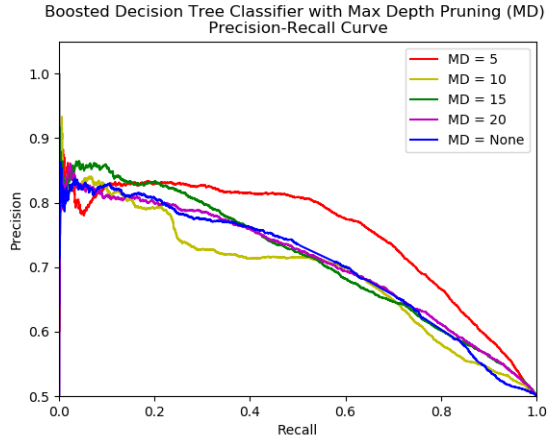

Fig. 9: *Precision-Recall curve for multiple decision tree classifiers trained on the financial loan dataset with different max depth pruning parameter.*

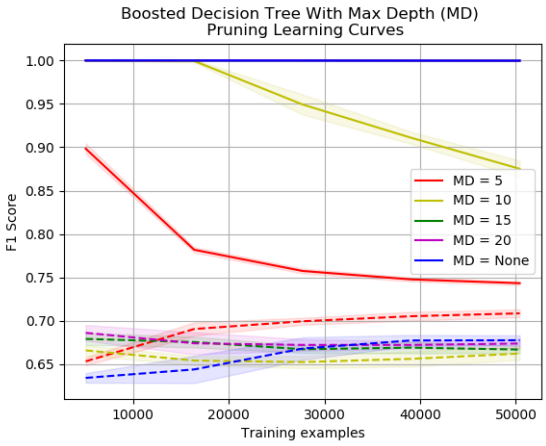

Fig. 10: *The F1 learning curve as a function of training examples for each model trained with boosting and max depth pruning.*

Figure 10 shows how boosting affects the learning curve for each decision tree model with max depth pruning described in the previous cardiovascular section. Figure 10 shows that boosting actually helped the model with no pruning parameters increase it's F1 score as the model was trained with more training examples. Figure 10 also shows that several models obtained a perfect score on the training data and the associated cross-validation score for those models was lower than the models trained without boosting. This shows strong evidence of over-fitting for the models trained with boosting.

4.2.2 Financial Loan Problem

The same boosting algorithm was also applied to the decision tree models that were trained to solve the financial loan problem. Figure 11 shows the precision-recall curve for the boosted decision tree models, which shows very identical curves for the models with pruning parameters.

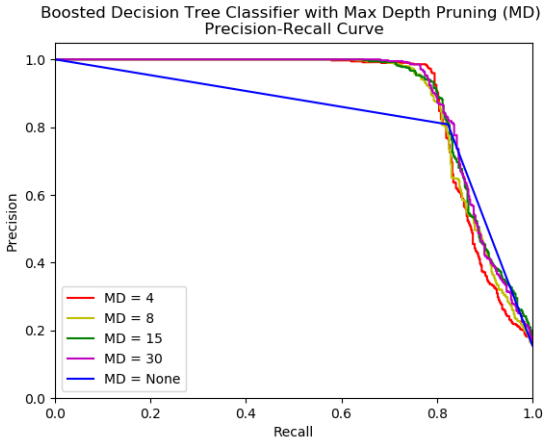

Fig. 11: *Precision-Recall curve for multiple decision tree classifiers trained on the financial loan dataset with different max depth pruning parameter.*

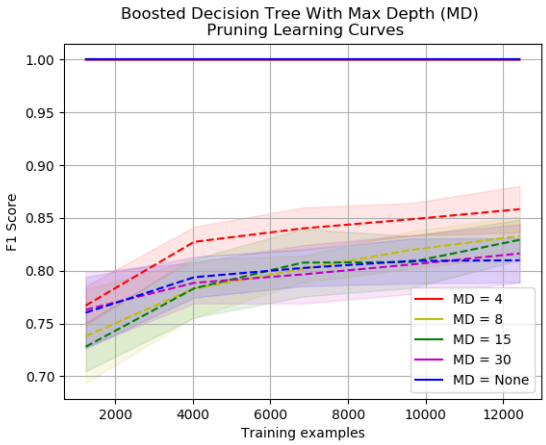

Fig. 12: *The F1 learning curve as a function of training examples for each model trained with boosting max depth pruning.*

Similar to figure 10, the learning curves in figure 12 show strong evidence of over-fitting. In this case, every model

obtained a perfect score on the training data, while the cross-validation score for each model was significantly less. Figure 12 also shows a general pattern for the cross validation error, as the models with a higher depth pruning parameter display more cross-validation error.

4.2.3 Boosted Decision Tree Improvements

Over-fitting seemed to be a problem for the boosted decision tree models in both cardiovascular and financial loan problems. The learning curves for both problems indicate that over-fitting was less severe when the pruning parameters forced the tree to be smaller and more concise. Therefore, it could be possible to increase the F1 score by forcing the depth of the tree to be smaller than the smallest depth values included in the analysis above.

4.3 Neural Network Results

4.3.1 Cardiovascular Problem

Figure 13 shows the precision-recall curve for various neural networks with varying hidden layer sizes. The plot shows that the size and number of hidden layers in the neural network has little influence in regard to improving the models performance. Figure 14 actually shows that each model's F1 training score and cross validation score converge roughly to the same value. The only significant difference between the models is the rate at which each model improves its F1 score. Figure 14 also shows evidence indicating that neural networks with more nodes tend to learn more quickly.

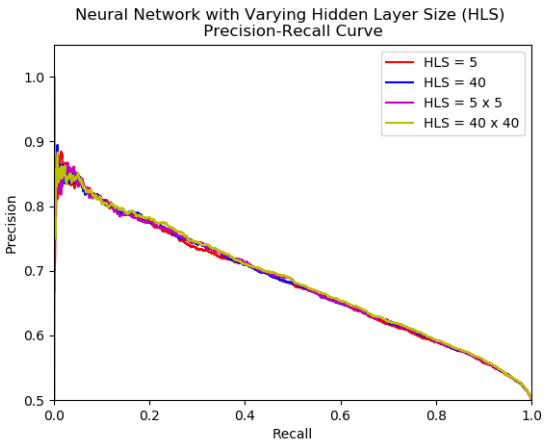

Fig. 13: *Precision-Recall curve for neural network classifiers trained on the cardiovascular dataset with hidden layer sizes.*

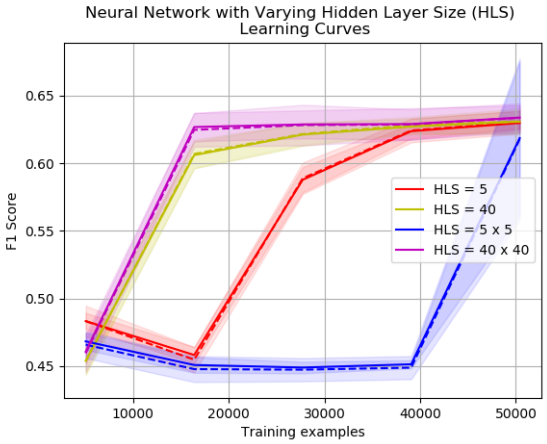

Fig. 14: *The F1 learning curve as a function of training examples for each neural network trained with varying hidden layer sizes.*

4.3.2 Financial Loan Problem

Figure 15 shows strong evidence in regard to how neural networks with different hidden layer sizes affect the precision-recall curve for the financial loan problem. As the number of nodes increase in the first hidden layer, precision-recall curve seems to improve. Figure 16 shows that each model generally improves with more training examples, so its also possible that more training data could further improve the neural network models.

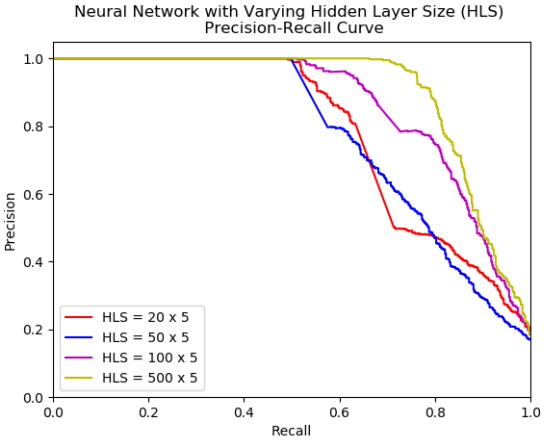

Fig. 15: *Precision-Recall curve for neural network classifiers trained on the financial loan dataset with hidden layer sizes.*

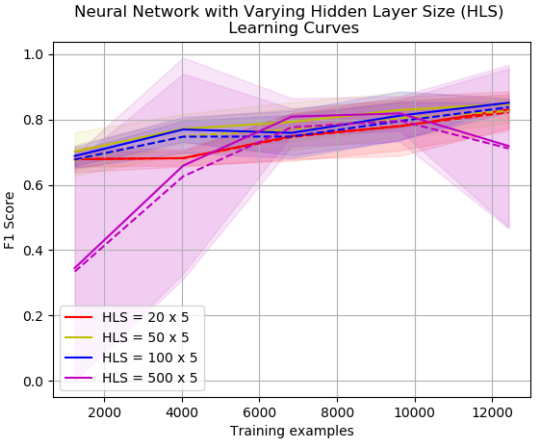

Fig. 16: *The F1 learning curve as a function of training examples for each neural network trained with varying hidden layer sizes.*

4.3.3 Neural Network Improvements

The neural networks with a larger first hidden layer tended to have better precision-recall curves than the neural networks with a smaller first hidden layer for the financial loan problem. However, it is difficult to speculate how much more the model could improve if the number of nodes in its first layer continues to grow. Training large neural networks dramatically increases the analysis runtime, which made analyzing larger neural networks difficult. The neural networks trained on the cardiovascular dataset were far less interesting than the ones trained on the financial loan dataset because the cardiovascular models did not change in performance much when their parameters were changed. One reason for this could be due to the number of features in each dataset. The cardiovascular dataset contained only 10 features, while the financial loan dataset contained 57 features. Therefore, its possible that the features in the cardiovascular dataset play a large role in the models consistent performance.

4.4 k Nearest Neighbors Results

Figure 17 shows the precision-recall curves for k nearest neighbor models that were trained on the cardiovascular dataset. The plots closely

4.4.1 Cardiovascular Problem

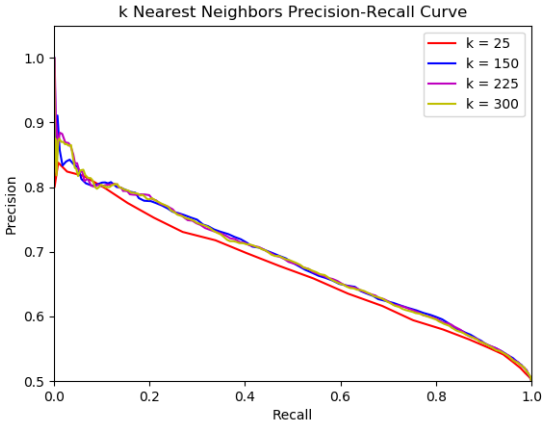

Fig. 17: *Precision-Recall curve for neural network classifiers trained on the financial loan dataset with hidden layer sizes.*

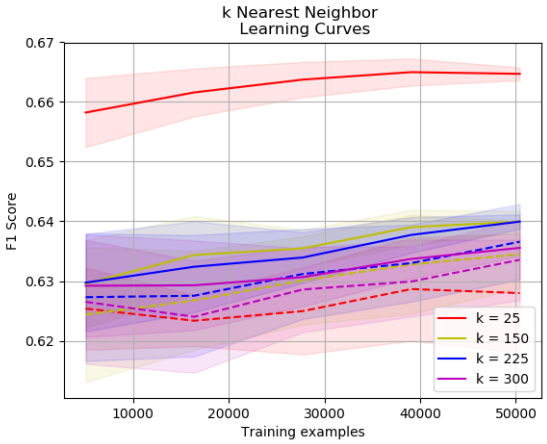

Fig. 18: *The F1 learning curve as a function of training examples for each neural network trained with varying hidden layer sizes.*

4.4.2 Financial Loan Problem

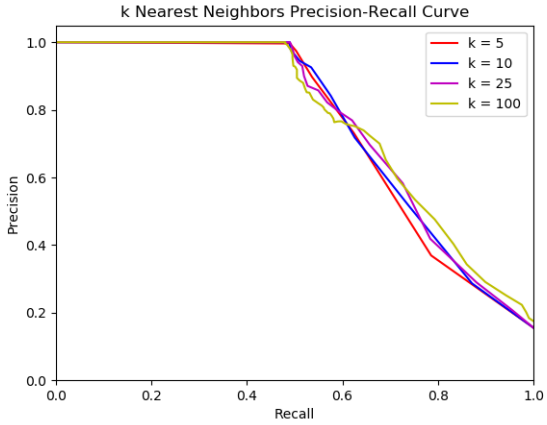

Fig. 19: *Precision-Recall curve for neural network classifiers trained on the financial loan dataset with hidden layer sizes.*

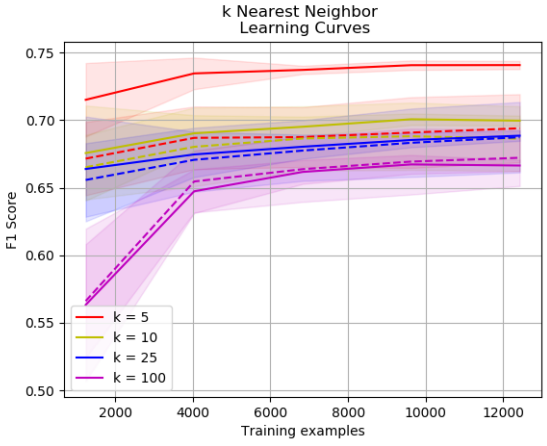

Fig. 20: *The F1 learning curve as a function of training examples for each neural network trained with varying hidden layer sizes.*

4.4.3 k Nearest Neighbor Improvements

One aspect of the k nearest neighbor algorithm that was not included in the analysis above is the distance function that is used in calculating the k nearest neighbors for a

particular training example. The default distance function is the euclidean distance function, which weights all features the same when calculating the distance between all of the features. It is possible to swap distance functions to improve the algorithms performance. However, the complexity of the cardiovascular and financial loan problems makes it difficult to try new distance functions unless one is a domain expert in one of those fields. Domain experts could help aid in weighting different features and different cases to build a more optimal distance function. Therefore, the analysis above only included different variations of k.

4.5 Support Vector Machine Results

Figures 21-22 show the precision-recall curves for support vector machines (SVM's) that were trained on the cardiovascular and financial loan datasets accordingly.

4.5.1 *Cardiovascular Problem*

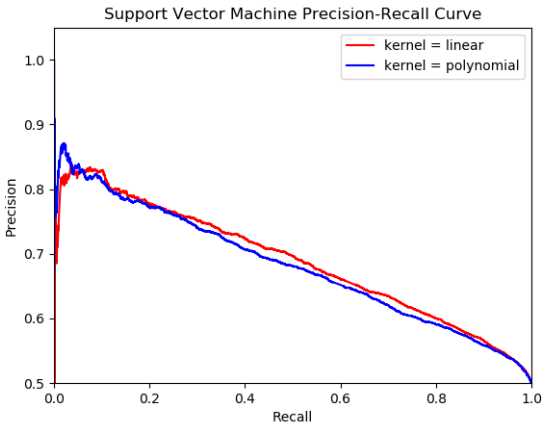

Fig. 21: *Precision-Recall curve for SVM classifiers trained on the cardiovascular dataset with different kernels.*

4.5.2 *Financial Loan Problem*

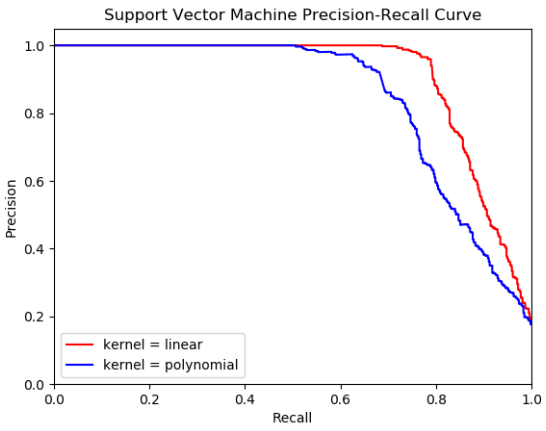

Fig. 22: *Precision-Recall curve for SVM classifiers trained on the financial loan dataset with different kernels.*

4.5.3 *Support Vector Machine Improvements*

Unfortunately, the size of both datasets made it complicated to generate learning curve plots as a function of training examples. Tables 1 and 2 show how their associated prediction and training times complicate generating the plots. This also made it more complicated to test if different kernel performed better than the linear and polynomial kernels. One possible solution to test out new kernels more efficiently would be reducing the training examples and possibly even some less important training features, so that some quick prototypes could be built. Removing unimportant features would hopefully still provide a model that is somewhat representative of the entire dataset.

5 Results Comparison

Figure 23 shows the best precision-recall curve for each learning algorithm that was trained on the cardiovascular dataset. The best model was chosen by the model that attained the highest F1 score on the test set. Table 1 shows the F1 score for each model that aims to solve the cardiovascular problem. The F1 score in bold indicates the highest score for each particular model type. The best overall values are highlighted in green and the poorest overall values are highlighted in red.

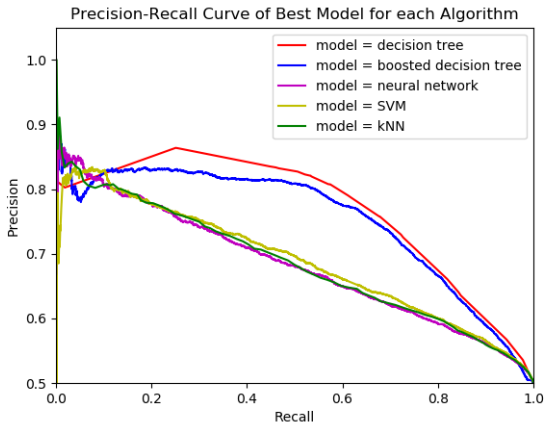

Fig. 23: *Best precision-recall curves for each model type trained on the cardiovascular dataset.*

Figure 24 shows the best precision-recall curve for each learning algorithm that was trained on the financial loan dataset. The best model was chosen by the model that attained the highest F1 score on the test set. Table 2 shows the F1 score for each model that aims to solve the financial loan problem. The F1 score in bold indicates the highest score for each particular model type.

TABLE 1: Cardiovascular Model Comparison

Model	Training Time (s)	Prediction Time (s)	F1 Score
DTC (MD = 5)	0.067	0.001	**0.715**
DTC (MD = 10)	0.123	0.001	0.712
DTC (MD = 15)	0.182	0.002	0.693
DTC (MD = 20)	0.212	0.003	0.678
DTC (None)	0.285	0.003	0.627
DTC (MLN = 20)	0.067	0.001	0.715
DTC (MLN = 100)	0.095	0.001	0.717
DTC (MLN = 1000)	0.159	0.002	0.708
DTC (MLN = 2000)	0.165	0.003	0.704
Boosted DTC (MD = 5)	3.59	0.058	**0.705**
Boosted DTC (MD = 10)	6.81	0.078	0.661
Boosted DTC (MD = 15)	9.87	0.111	0.659
Boosted DTC (MD = 20)	10.7	0.123	0.671
Boosted DTC (None)	6.01	0.104	0.661
NN (HLS = 5)	6.92	0.001	0.629
NN (HLS = 5 x 5)	6.6	0.005	0.623
NN (HLS = 40)	7.35	0.001	**0.641**
NN (HLS = 40 x 40)	9.98	0.005	0.637
kNN (k = 25)	2.75	3.6	0.625
kNN (k = 150)	2.8	4.7	0.631
kNN (k = 225)	2.8	5.1	**0.634**
kNN (k = 300)	2.9	5.9	0.629
SVM (linear)	91.2	6.2	**0.629**
SVM (polynomial)	92.8	7.8	0.592

TABLE 2: Financial Loan Model Comparison

Model	Training Time (s)	Prediction Time (s)	F1 Score
DTC (MD = 4)	0.201	0.001	0.679
DTC (MD = 8)	0.375	0.001	**0.873**
DTC (MD = 15)	0.549	0.001	0.866
DTC (MD = 30)	0.903	0.001	0.845
DTC (None)	1.1	0.001	0.811
DTC (MLN = 5)	0.249	0.001	0.656
DTC (MLN = 20)	0.289	0.001	0.876
DTC (MLN = 100)	0.447	0.001	0.871
DTC (MLN = 300)	0.898	0.001	0.838
Boosted DTC (MD = 4)	10.8	0.027	**0.848**
Boosted DTC (MD = 8)	21.6	0.032	0.836
Boosted DTC (MD = 15)	30.1	0.036	0.806
Boosted DTC (MD = 30)	38.5	0.039	0.802
Boosted DTC (None)	0.963	0.002	0.815
NN (HLS = 20 x 5)	2.98	0.001	0.658
NN (HLS = 50 x 5)	0.991	0.002	0.658
NN (HLS = 100 x 5)	3.53	0.002	0.704
NN (HLS = 500 x 5)	10.3	0.02	**0.824**
kNN (k = 5)	0.386	10.1	0.671
kNN (k = 10)	0.386	10.1	0.662
kNN (k = 25)	0.367	10.1	**0.659**
kNN (k = 100)	0.372	10.4	0.648
SVM (linear)	1000+	42.4	**0.793**
SVM (polynomial)	1000+	50.1	0.717

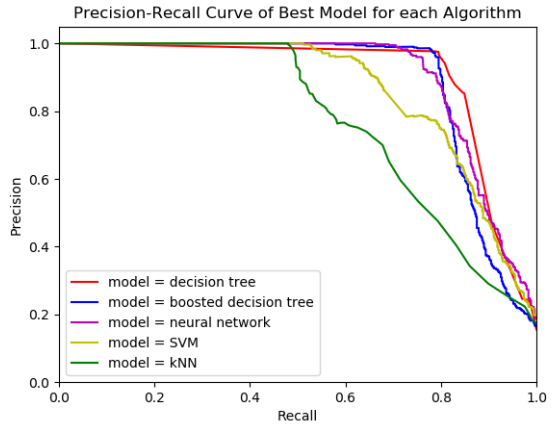

Fig. 24: *Best precision-recall curve for each model type trained on the financial loan dataset.*

The precision-recall curves shown in figures 23-24 can be an important tool for domain experts that wish to find an optimal trade-off between the number of false positive and false negative outcomes for a given model. The decision tree classifiers have shown to provide better precision-recall curves and F1 scores than all other models for both the cardiovascular and financial loan problem. Tables 1 and 2 also show that decision trees could predict and be trained faster than all of the other models. Boosted decision trees had similar F1 scores to regular decision trees, but took roughly ten times as long to train. Neural networks compared well to decision trees in their speed in making predictions, but took longer to train and generally yielded lower F1 scores for both problems. The neural networks in the cardiovascular problem took much longer to train than the neural networks of roughly the same size in the financial loan problem. That is likely a result of the cardiovascular dataset being roughly three times as large as the financial loan dataset. However, the financial loan dataset had 47 more features than the cardiovascular dataset, so SVM took significantly longer to train for the financial loan problem. This also resulted in the kNN models trained on the financial loan dataset to be slower to be slower in making predictions than the kNN models that were trained on the cardiovascular dataset.