Ashwin Rathie

CS 4641 – Machine Learning

Professor Byron Boots

September 23, 2018

Analysis of Supervised Learning Models

In this analysis, 5 different Supervised Learning models are examined based upon their performance on two classification problems. The models in question are 1) Decision Trees, 2) Neural Networks, 3) Boosting (specifically ADA boosting), 4) Support Vector Machines (SVMs), and 5) k-Nearest Neighbors.

Datasets

The two datasets I have chosen interest different aspects of my psyche. The first dataset activates the optimistic, idealistic part of me that believes technology has the ability to help a countless number of people dramatically; it has to do with analyzing breast cancer and its symptoms/physical signifiers in the hopes that a predictive machine learning model can help detect breast cancer in its early stages. The features of this dataset are standard measurements in medical examination of those being checked for breast cancer; these metrics include smoothness, points of concavity, perimeter, symmetry, and fractal dimension (among others) of the patients' breasts. The target output of the dataset is whether or not the potential cancer is malignant (M) or benign (B), making this a binary classification problem. This dataset contained a relatively low 570 examples; However, its 30 different features could prove to make up for the deficiency in quantity of entries by increasing the quality. If a supervised learning algorithm could yield a low error rate for predicting whether a tumor is malignant or benign, it could be utilized by medical professionals and potentially save millions of lives.

The second dataset ignites a personal, familial side of me; my mother works as a real estate agent so naturally a dataset containing features of houses in King County, USA, and their sale prices jumped out at me. The features include square footage of the house, number of bedrooms and bathrooms, lot size, year built, size of the basement, and condition of the house (among others). The target output for this problem is the price of the house. Because price is a continuous, numeric metric, this dataset initially seems like it would lend itself towards a regression problem rather than a classification problem. However, it can be transformed into a classification problem by segmenting the possible values of outputs into categories of price. This was achieved through assigning each data entry to one of 10 price categories. These categories are the percentiles the houses' prices fit into in relation to the rest of the houses in the vast dataset (0-10 percentile, 10-20 percentile, 20-30 percentile, etc.). The result is a classification problem with 10 different possible classifications. In contrast to the breast cancer dataset, the housing prices dataset contains over 21,000 data entries, roughly 40 times that of the former data set. This discrepancy could prove to be a factor in the performance of the supervised learning models. A successful application of supervised machine learning models to this problem could produce a tool that allows real-estate buyers to shop more intelligently and seek fair prices.

Testing the Models

In testing the models, I conducted two tests per supervised learning algorithm per dataset (a grand total of 20 tests).

The first test will be referred to as the learning test. This test demonstrates the performance of the models with "Error" on the y-axis and "Percentage of Training Data" used to train the model on the x-axis and displays the error rates of the training set and the testing set. Error is defined in this analysis as the number of times the model's output was inaccurate divided by the total number of data examples applied to it. The training data was set to 67% (2/3) of the total data and the testing data was set to 16.5% (1/6) of the total data. The final 16.5% (1/6) was allocated to a validation set that will be used in the next test. This test could prove useful for exposing overfitting and determining how much data is ideal for training the various models.

The second test will be referred to as the validation test. In this test, the models are run on the validation data (not the test data from the learning tests) and the training data for comparison. The error is still examined as the y-axis metric, but not the x-axis metric is a hyper-parameter depending on the model in question (decision trees: max depth, neural network:

hidden layers size, ADA boost: learning rate, SVM: gamma, KNN: number of neighbors). The models' performances are measured as the hyper-parameters are varied. This test could prove useful in tuning the hyper-parameters to determine in which conditions each model works best. Furthermore, this test incorporates k-fold cross validation, for which there are 10 'folds' of the training data for all of the models except for the SVMs. Because SVMs require a greater amount of training data to yield meaningful results, less 'folds' were necessary as too great a number would divide the data into subsets that are too small, so the cross validation 'folds' were set to 5 for this type of model.

Decision Trees

Decision trees are an example of a greedy search algorithm that utilize the heuristic of information gain. In implementing the decision tree model for the two classification problems at hand, it was important to take into account the principle of Occam's Razor, the idea that the simplest, most consistent explanation is often best. In the context of decision trees, this means that a smaller decision tree is generally better than a larger, overly complex one. For this reason, I decided to prune the decision trees by limiting their max depths to 5 for the initial learning tests; in the validation tests, the max depths will be varied.

Another factor that was taken into account is that too little data and too many irrelevant features could result in overfitting; with just 570 entries and 30 different features, the breast cancer dataset yield a very poor the decision tree performance. To mitigate this, the breast cancer features were limited to five most relevant attributes: texture, perimeter, smoothness, compactness, and symmetry (this change in data was kept consistent throughout all models' tests).

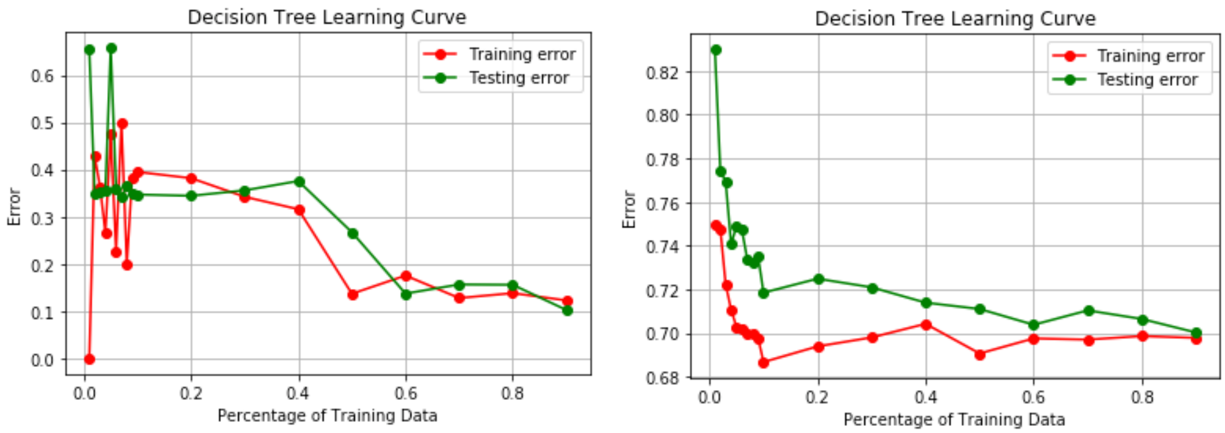

***Figure 1**: Breast Cancer (left) and Housing Prices (right) Tree Learning Curves*

When looking at the graphs produced from the learn tests, it can be seen that the training error and testing error seem to trend towards lower error at similar rates to each other in both classification problems. Additionally, both classification problems share the property that the errors of both seem to level out after approximately 50% of the training data is used, with little meaningful improvement in performance from 50% of the training data to 100%. This is likely the result of the pruning condition of max depth set at 5; more data used would normally result in more decision tree branches formed, but the limit at 5 means that more data does not necessarily result in more branches in the tree.

A dramatic difference in the results of the two problems is that the breast cancer model yielded must more accurate (lower error) scores than did the housing prices model. With just the knowledge of these tests isolated from later tests, this could be due to the specific implementation of this model, the fact that the breast cancer problem is binary rather than the housing prices' 10 categories, or the breast cancer problem may simply have a greater correlation between its features and output. Another difference is that the errors of the breast cancer analysis seem to have dropped at a quicker rate initially than did the housing prices analysis; the former's testing error dropped by approximately .30 in the first 10% of training data used while the latter's dropped by only .12 in the same training data percentage frame. This could be because the breast cancer data has far less data examples than the housing prices data with roughly the same number of features (after the limiting of breast cancer features) so the breast cancer decision tree's splits differentiate between the fewer number of training example more quickly.

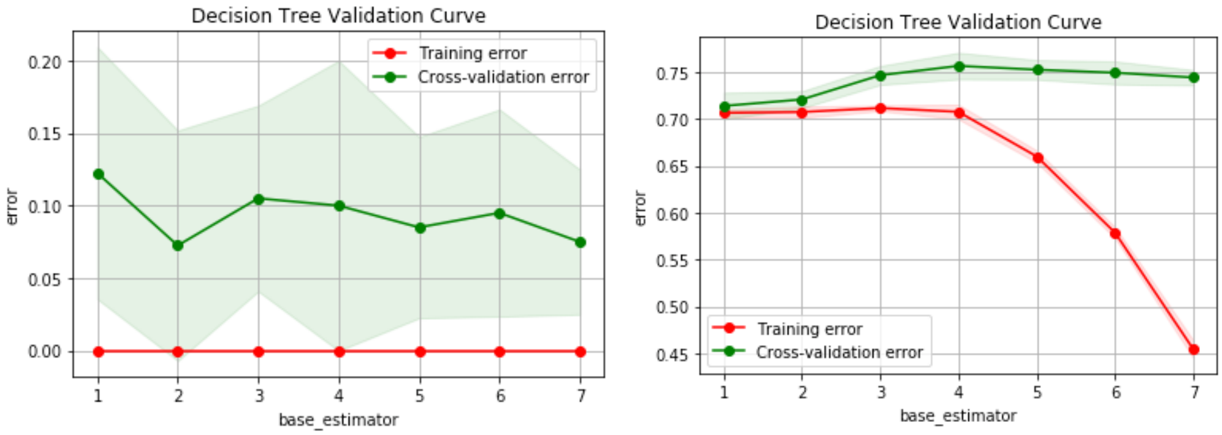

***Figure 2**: Breast Cancer (left) and House Prices (right) Tree Validation Curves*

Error is measured as the max depth (x-axis) is varied.

Note: The shaded area represents the cross-validation standard deviation

When examining the validation curves in Figure 2, we notice that the training error in the breast cancer tree is flat-lined at 0. This may be due to an implementation error or the fact that the classifier had already been trained on the training data set. Another observation of the graphs is that the cross validation standard deviation (represented by the colored area) is much greater in the breast cancer results than it is in the housing prices results; the lesser amount of data in each CV fold of the breast cancer dataset than in each fold of the housing prices dataset is likely the cause of the wider spread.

As the housing prices model passes a max-depth of 4, the training error quickly dips while the CV error remains relatively constant. This is an example of overfitting, as ever-increasingly complex tree is matching perfectly to the data it is being trained on, causing training error to plummet, while performance on external data is not improved.

Neural Networks

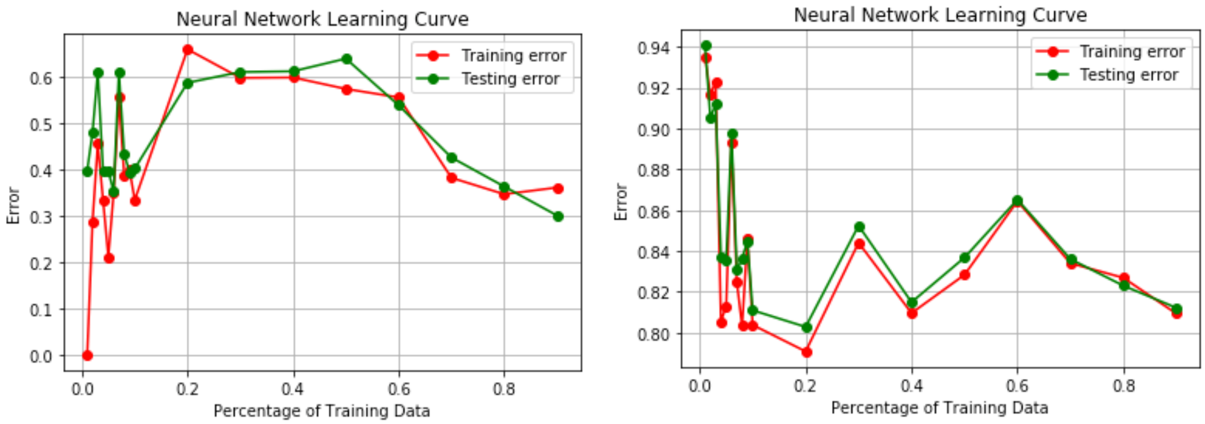

Figure 3*: Breast Cancer (Left) and Neural Network (right) NN Learning Curves*

Neural networks appear to have performed the most poorly among the 5 supervised learning models, as they generally have the highest error rate. This could be due to the training limitations; neural networks are extremely computationally heavy and therefore take a relatively long time to train. These models took longer to train than all models except SVMs and were trained without GPUs, which are exceptional at mathematical computation, which make them ideal for neural network training. Although the overall accuracy was not very good, the training error and testing error were consistently very close to one another, indicating that overfitting was minimal to none.

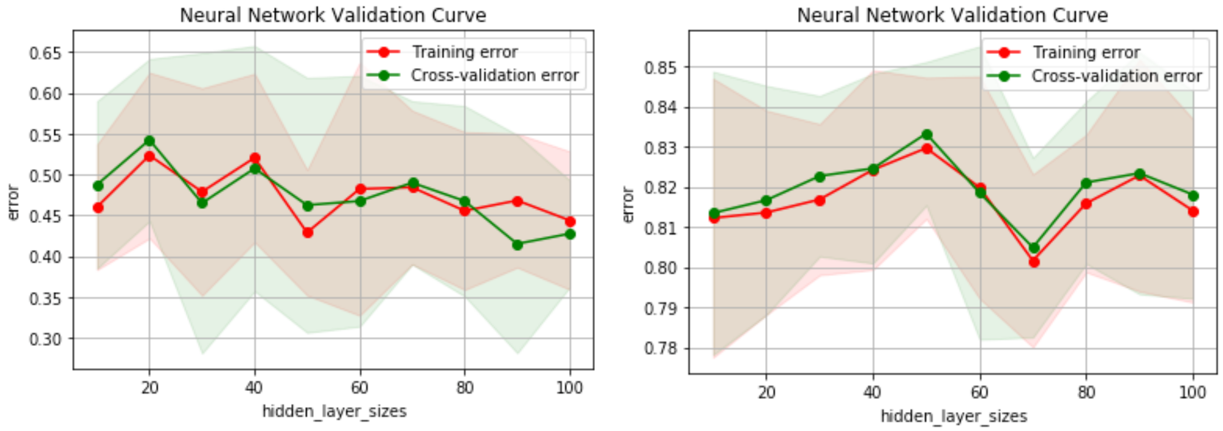

Figure 4*: Breast Cancer (left) and House Prices (right) NN Validation Curves*

Error is measured as hidden layer size (x-axis) is varied.

Note: The shaded area represents the cross-validation standard deviation

The validation curve graphs reinforce the conclusion that neural networks performed relatively poorly on both classification problems; with a score of about .45 error in the breast cancer data, a binary classification problem, it barely outperforms sheer guessing. In comparison, the decision tree scored about .10 error rate on the same data. It can also be seen that the cross validation standard deviation is very high in both curves, indicating that neural networks do not perform consistently with lesser amounts of training data. Although the performance does improve slightly as the hidden layers size is increased, it doesn't improve by a significant amount, showing that sheer number of perceptions does not guarantee meaningfully better performance.

ADA Boosting

ADA boosting is very intriguing because it uses a weak classifier (in this analysis, a shallow decision tree), but applies in a recursive manner to yield in generally strong results.

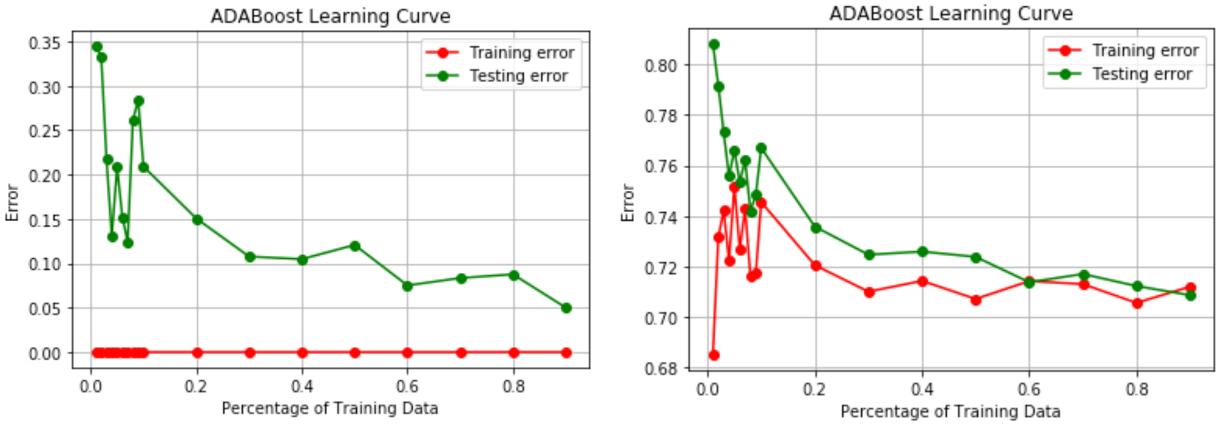

Figure 5*: Breast Cancer (left) and Housing Prices (right) Boost Learning Curves*

The training error in the Breast Cancer is flat-lined, possibly because the classifier has already been trained on the same data and quickly reached a training error of 0. The training error of 0 exposes a peculiar property of the ADA boost model. The testing error continues to decrease even after the training error had already bottomed out at 0. This is a very counter-intuitive and unique feature of ADA boosting among these supervised learning models. When the training error hasn't hit 0, it appears to remain closely correlated with the testing error, as displayed on the housing prices learning curve.

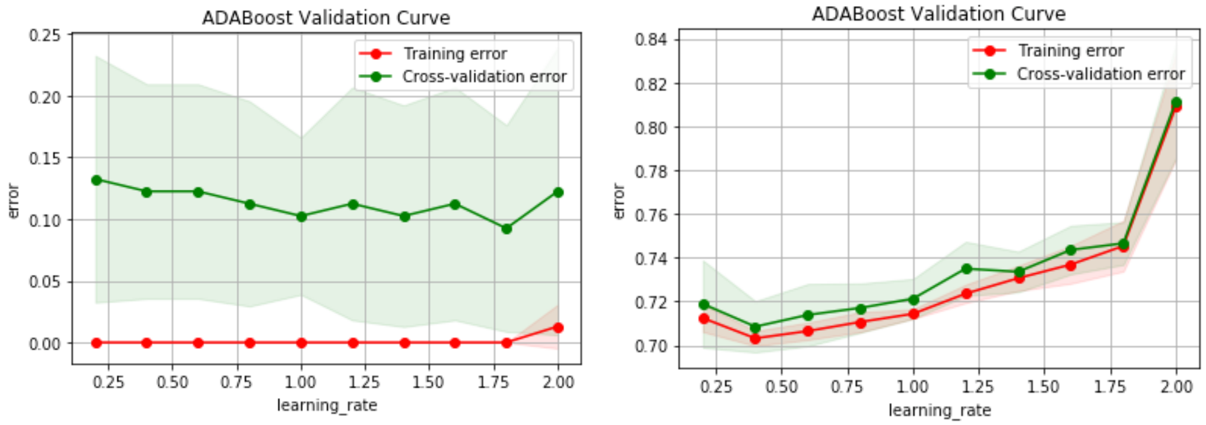

Figure 6*: Breast Cancer (left) and House Prices (right) Boost Validation Curves*

Error is measured as learning rate (x-axis) is varied.

Note: The shaded area represents the cross-validation standard deviation

The right graph shows that an overly aggressive learning rate in ADA boosting implementation could result in a higher error; this is while applying an excessively high learning rate can get to the ideal weights more quickly, it will cause the weights to miss the ideal by overshooting the target. The overall performance exhibited in Figure 6 is quite good in terms of having a low error rate; the error rate is comparable to those produced by the decision tree model, the KNN model, and SVMs. However, ADA boosting did take more time and greater computational effort to train when compared to the decision tree and KNN. While the ADA boost model does use a less complex decision tree than the deeper decision tree model, it recursively applies the decision tree many time whereas the more complex decision tree only has to be created once.

Support Vector Machine

Like neural networks, SVMs are very computationally heavy, which makes training them time-consuming. However, they depart from neural networks in their performance on the two classification problems in question.

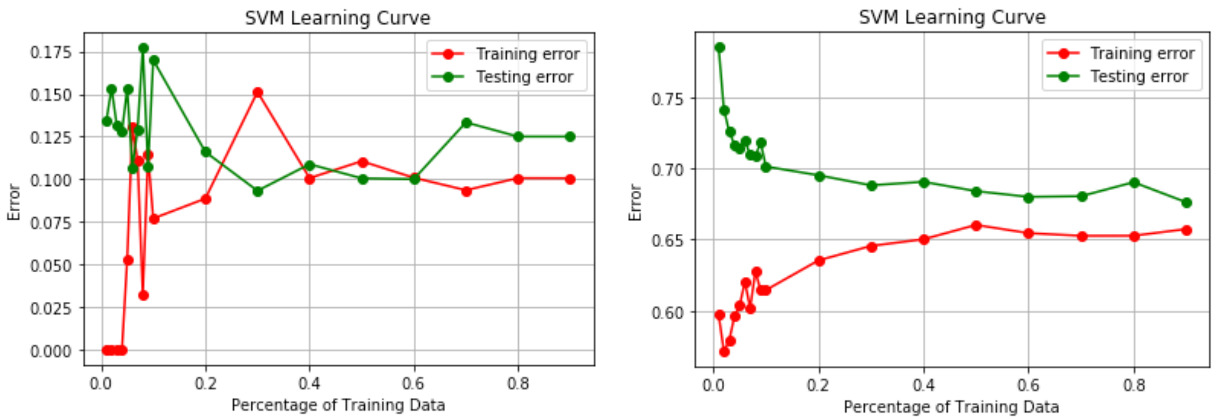

***Figure 7**: Breast Cancer (left) and Housing Prices (right) SVM Learning Curves*

An examination of the general error rates seen in Figure 5 show it vastly outperforms neural networks in terms of achieving lower error rates. The error rates indicated in the learning curve graphs are comparable or better than those produced by all the other models with the exception of neural networks. However, the validation tests in Figure 8 below point to some deficiencies in SVMs.

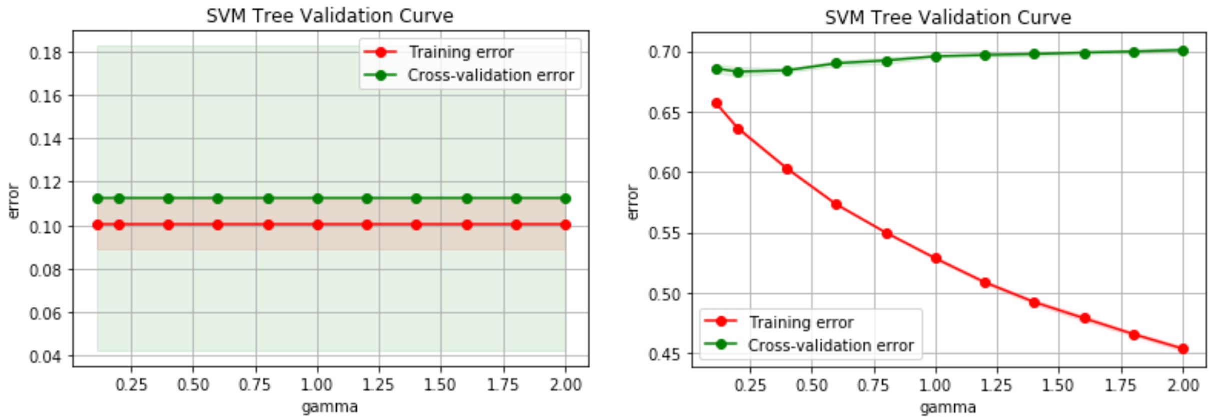

***Figure 8**: Breast Cancer (left) and House Prices (right) SVM Validation Curves*

Error is measured as gamma (x-axis) is varied.

Note: The shaded area represents the cross-validation standard deviation

Both curves in Figure 8 expose a potential weakness of SVMs. The breast cancer validation curve shows a breakdown of the SVM implementation. The spread of the cross-validation standard deviation across the entire graph indicates that the SVM simply did not have enough data. SVMs require more data than other models, and the k-folded breast cancer data set proved to be too small, despite only having 5 folds than the 10 folds of the models. The house prices curve shows that SVMs are susceptible to overfitting if the gamma in increased too much. As the gamma hyper-parameter is increased, the training error steadily drops, but the cross-validation error increases, suggesting that the model is overfitting the training data. This occurs because the gamma essentially dictates how much influence each one of the training examples has on the model; a higher gamma would manipulate the model to fit the training data more closely and more quickly.

K-Nearest Neighbors

In a subjective sense, KNN is the most conceptually simple of these 5 supervised learning models. As seen in figure 9 below, the error rates produced by KNN are competitive with those produced by other models.

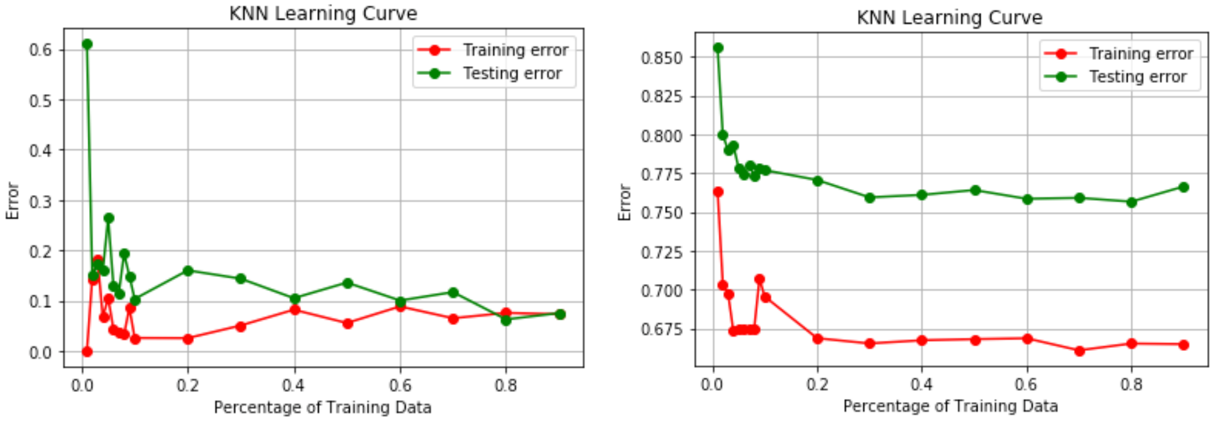

***Figure 9**: Breast Cancer (left) and Housing Prices (right) KNN Learning Curves*

While the housing prices error rate doesn't outperform SVMs or ADA Boosting, it remains close. The breast cancer error rate is among the lowest and can only be matched by ADA Boosting. However, as seen in both of Figure 9's learning curves, the low error rates are consistent throughout the percentages of training data used, showing that KNN does not require large amounts of data to excel like SVMs.

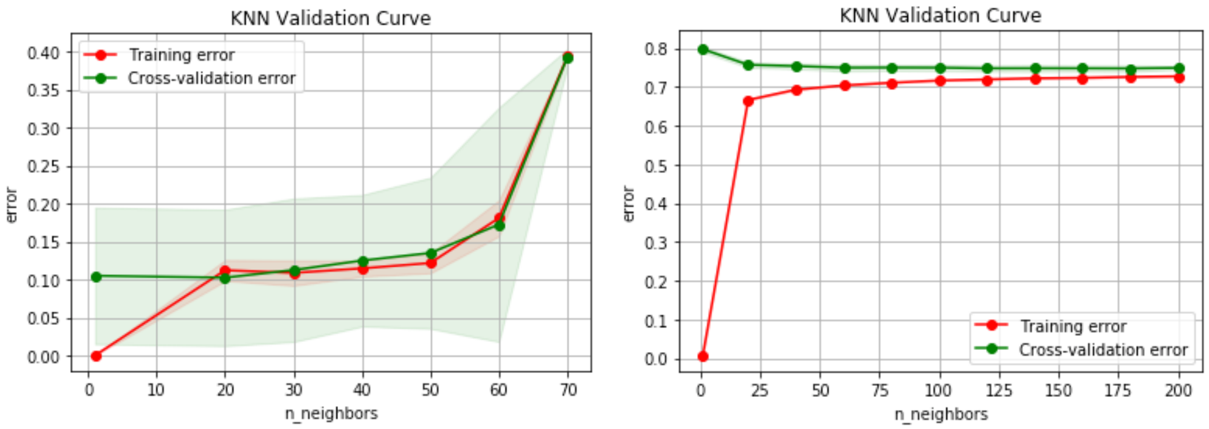

***Figure 10**: Breast Cancer (left) and House Prices (right) KNN Validation Curves*

Error is measured as # of neighbors (x-axis) is varied.

Note: The shaded area represents the cross-validation standard deviation

The breast cancer validation curve in figure 10 shows that as the number of neighbors becomes a significant portion of the total data, the error rate will increase. This is intuitive, because it would cause data to normalize to just one output value, which is not going to be accurate for most balanced datasets. It must be noted that the KNN models were trained in a lesser time that all of the other models included in this analysis, and in a significantly lesser time than all but the decision trees model.

Conclusion

In this analysis, two intriguing, real-world classification problems were leveraged to analyze the performance and behavior of 5 supervised learning models as the percentage of training data and various hyper-parameters were manipulated. A breast cancer dataset, including attributes looked for in breast cancer patients and the resulting diagnosis, was used to form a binary classification problem. A house prices dataset, including details about homes and their sale prices, was used to create a 10-category classification problem.

After applying the 5 models to both of these problems, it became clear that these supervised learning algorithms were able to address to the breast cancer problem with much greater accuracy than the house prices problem. This is most likely not due to the specific implementation of the models, because this conclusion was consistent across all models.

However, the nature of binary problems compared to 10-category problems and a greater correlation in the breast cancer dataset were likely the causes of this conclusion.

With the exception of neural networks, which underperformed relative to the other 4 models, all of the models performed comparably in terms of accuracy. However, when determining which model is "best", lowest error rate is not the only factor. I believe susceptibility to overfitting and time taken to train the model are also significant factor. In these regards, K-Nearest Neighbors is the best supervised learning model in my opinion. Despite being the most conceptually simply (subjectively), it consistently maintains a low error rate and is does not generally overfit. Because it is the most conceptually simply, it doesn't not require as much computation as some of the other models and, therefore, trains in a little amount of time. The combination of these advantages and lack of deficiencies makes it, in my estimation, the best supervised learning model.

Citations:

Code for Implementation of models provided by Uday Patil and Sohan Choudhury

House Prices dataset provided by "harlfoxem" at
https://www.kaggle.com/harlfoxem/housesalesprediction

Breast Cancer dataset provided by "UCI Machine Learning" at
https://www.kaggle.com/uciml/breast-cancer-wisconsin-data