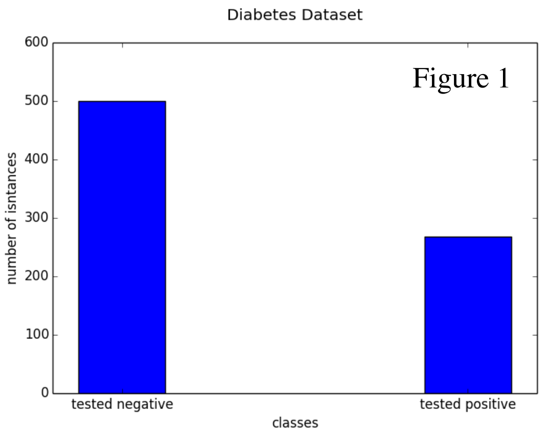
Yamini Nambiar
CS 4641
Project 1
5 February 2017

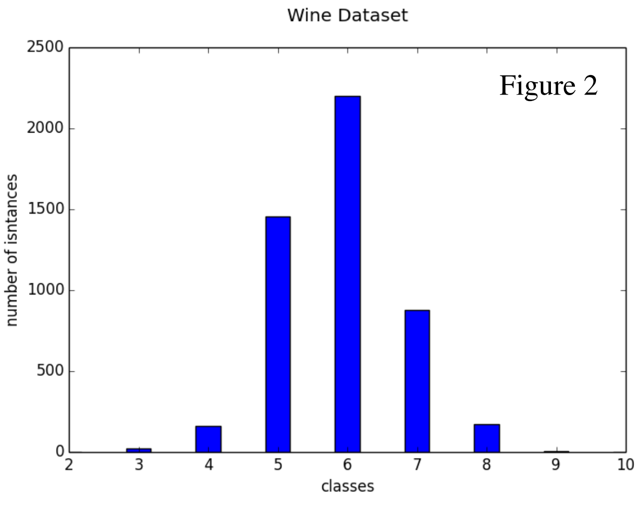
Project 1: Supervised Learning

DATA



Diabetes. The diabetes data set contains 768 instances and 8 attributes plus class. The dataset was donated by the National Institute of Diabetes and Digestive and Kidney Diseases and is based on a population from Phoenix, AZ. The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria. The data is multivariate and each attribute is numeric-valued. Each instance is either classified as 0 or 1, where 0 maps to tested negative for diabetes and 1 maps to tested positive for diabetes. The baseline accuracy for the diabetes dataset is 64.8%. Baseline error was 35.2%. The training set contains 530 instances and the test set contains 238 instances based on a 70-30 split.

Wine. The wine data set contains 4,899 instances + and 11 attributes plus class. The dataset is a result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The classification determines the quantities of 11 constituents found in each of the three types of wines. That data is multivariate and has numeric attribute values. Each instance is classified on a scale of 0 to 10, where each number maps to a classification of wine quality. The baseline accuracy for the wine dataset is 45.1%. Baseline error was 54.9%. The training set contains 3,430 instances and the test set contains 1,469 instances based on a 70-30 split.



Defining Best Model. In the following analysis, models are assessed based on their accuracy rates. A "best model" is the model with the highest accuracy. The accuracy rate represents the percent of instances the model correctly classifies. An accuracy rate greater than 50% means that the model correctly classified instances more frequently than it did not.

Cross-validation. Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. This method reduces variance by averaging over k different partitions, such that the performance estimate, which in this case is the percent of correctly classified instances, is less sensitive to the partitioning of the data. In this analysis, accuracy was measured using 10-fold cross-validation.

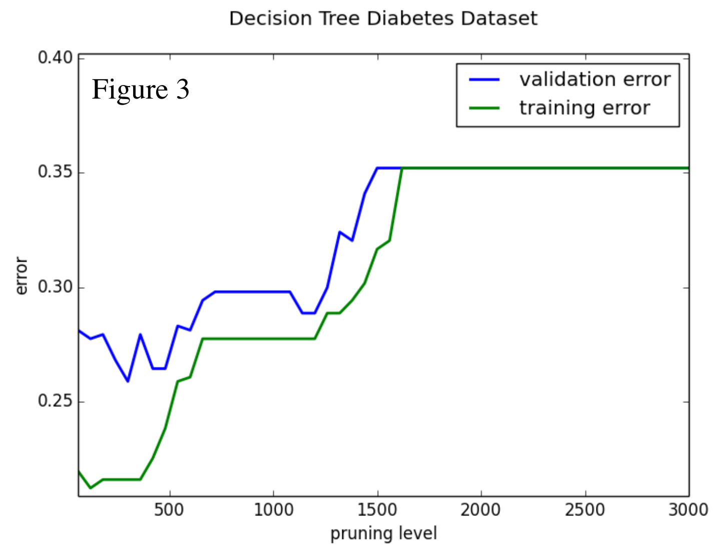
Model Complexity Graphs. Most of the model complexity graphs plot a complexity parameter against error rate. Error rate is calculated as (1-accuracy) through the entire analysis. Most model complexity graphs show the the validation error rate and training error rate as they vary over the range of the complexity parameter. The validation error rate represents how well the model classifies the validation set as part of the k-fold cross-validation method. The training error rate represents how the model classifies the training set it was trained on and should therefore have a lower error rate, or higher accuracy, than the validation error rate.

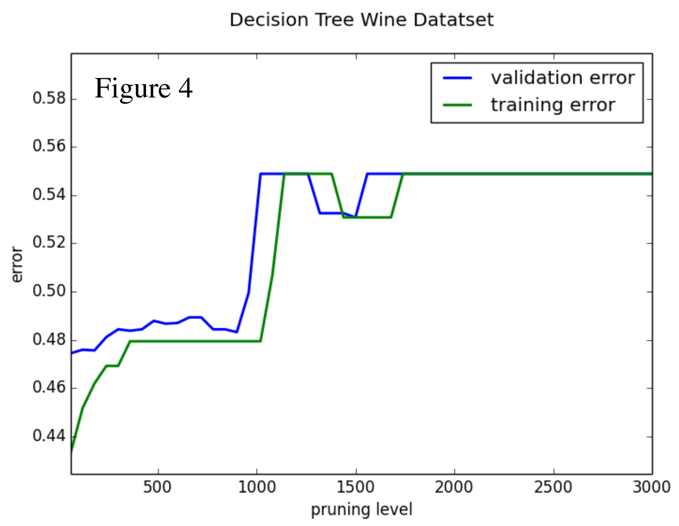
Learning Curves. The learning curves at the end of the analysis plot the prediction accuracy against training set size. The prediction accuracy is derived from the best model identified by the following analysis. The learning curve graphs are meant to represent how, or if, the model gets better at predicting the target classifier as the number of instances used to train the model increases.

DECISION TREES

Description. For both datasets, a complete tree was created using the GINI index split criterion. The algorithm used to model the tree was J48. The model complexity parameter for this experiment was pruning level. Pruning level represents the minimum number of objects required under a node.

Diabetes. For the diabetes dataset, a complete tree with 21 nodes was created. The results of pruning the tree are shown in Figure 3. Without pruning, the tree had an accuracy of 72.8%. The highest accuracy was achieved with a pruning level of 4. Accuracy ranged from 64.8% to 74.4%, making the pruning level a somewhat significant factor for this algorithm's model. 74.4% is a decent model accuracy rate but it can be improved.





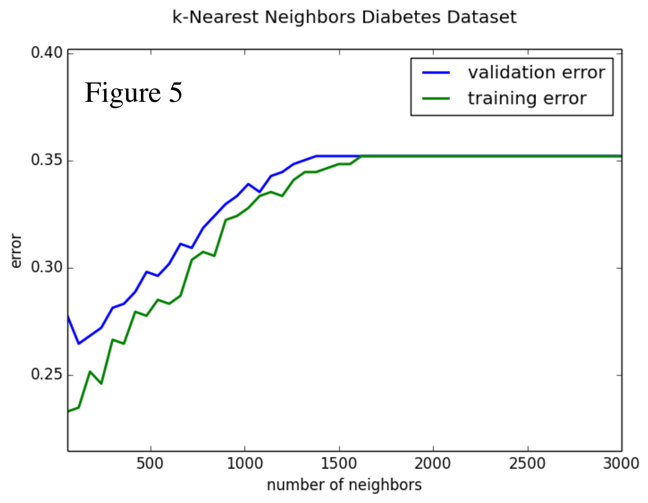
Wine. For the wine dataset, a complete tree with 1039 number of nodes was created. The results of the pruning tree are shown below in Figure 4. Without pruning, the tree had an accuracy of 52.6%. This was also the highest accuracy level among all iterations of varying the minimum number of objects required. The lowest accuracy for this dataset was 45.1% and it occurred at parameters greater than 1750. The baseline for the wine dataset was 45.1% so this model somewhat improved according to the baseline, but 52.6% is still just a little better than a random guess so there is still room for improvement.

Analysis. Overall, decision trees fared okay in terms of correctly classifying instances. For the diabetes dataset, the model showed an increase of 9.6% from the baseline model. For the wine dataset, the model showed an increase of 7.5% from the baseline model. The parameter under scrutiny was pruning level, or minimum number of instances per leaf that any node is guaranteed to have. Higher pruning levels lead to a greater amount of data separation per branching. Generally, higher pruning levels lead to a poorer classification model. As seen in Figure 3 and Figure 4, this hypothesis is proven by the large jumps in error as pruning level increased to the hundreds and then thousands. Furthermore, the best models were generated with pruning levels less than 5. This further proves the hypothesis that lower pruning levels generate more accurate models for these sized datasets. After this analysis, it appears that decision trees with low pruning levels generate more accurate models than those with high pruning levels.

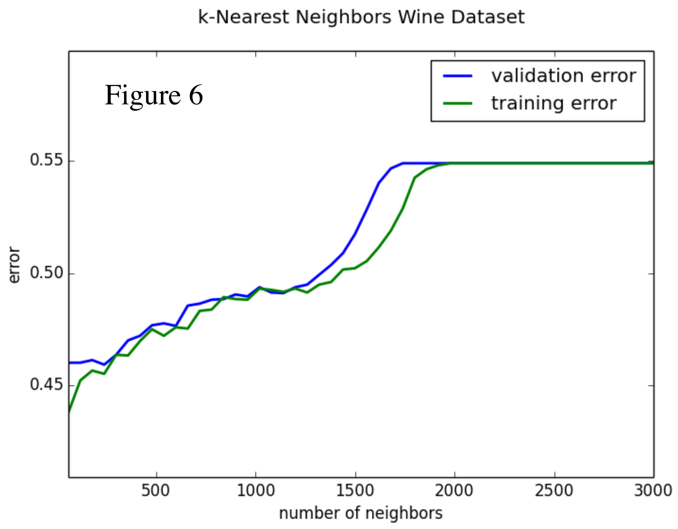
K-NEAREST NEIGHBORS

Description. For both datasets, the constant distance weight was measured for a varying number of neighbors. The algorithm used to model k-nearest neighbors was IBk. The model complexity parameter was the number of nearest neighbors considered in the classification.

Diabetes. For the diabetes dataset, the k-nearest neighbors algorithm performed well. With only 1 neighbor, the model had an accuracy of 70.6%. The highest accuracy was achieved with 14 neighbors on 530 instances. Accuracy ranged from 64.8% to 75.1%.



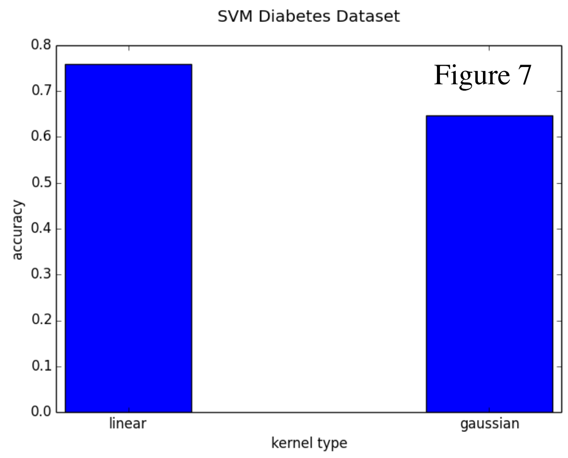
Wine. The wine dataset performed close to average for k-nearest neighbors. The highest accuracy was 54.1% and was found at including 3 nearest neighbors in the classification. With only 1 neighbor, the algorithm performed a little over half decent at 53.9%. Accuracy ranged from a baseline of 45.1% to 54.1%.



Analysis. The k-nearest neighbors algorithm did decently at classifying the diabetes and wine datasets. The algorithm produced a 10.3% increase in accuracy for the diabetes dataset and a 9.0% increase in accuracy for the wine dataset. The algorithm itself uses rote learning or "instance-based learning" and is based on a local average calculation. The parameter varied for this algorithm was the number of neighbors considered in the prediction. If the number of neighbors in the neighborhood is too small, the algorithm is susceptible to noise amidst the data. If the number of neighbors of data is too large, the model is too vague and covers too great an area of instance space to make an accurate prediction. Figure 5 and Figure 6 supports the hypothesis that k-nearest neighbors works well for large datasets and small neighborhoods. In this case, large datasets can be defined as those with more than 500 instances and small neighborhoods can be defined as those with less than 20 nearest neighbors. As seen in the graphs, as the number of neighbors increases, the error rate steadily increases. Overall, the data does show evidence that the more neighbors included in the prediction, the less accurate the model.

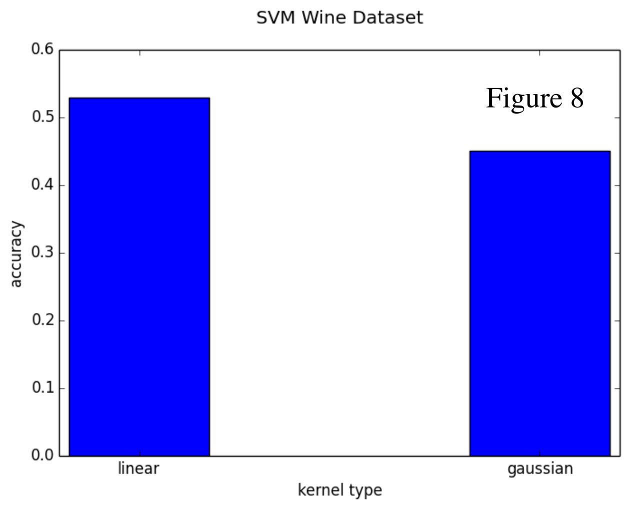
<u>SVM</u>

Description. Two kernel types were used for support vector machine analysis of the diabetes set: a linear kernel and a Gaussian kernel, also known as a radial basis function kernel. The SVM uses pairwise classification in which there is one binary SVM for each pair of classes to separate members of one class from members of the other.



Diabetes. The results of the diabetes dataset SVM is in Figure 7. The linear kernel performed at 75.8% accuracy. The Gaussian kernel performed at 64.8% accuracy. Because the diabetes dataset is so simple, it make sense that the simplest SVM kernel has the highest accuracy rate. If the SVM algorithm were to be run with more complex kernels, it is likely that the accuracy would not increase.

Wine. The results of the wine dataset SVM is in Figure 8. The linear kernel performed at 52.9% accuracy. The RBF kernel, or Gaussian kernel, performed worse at 45.1% accuracy which was the baseline. Because the wine dataset is not very complex, it make sense that the simpler SVM kernel has the highest accuracy rate similar to the diabetes dataset. If the SVM algorithm were to be run with more complex kernels, it is likely that the accuracy would not increase.

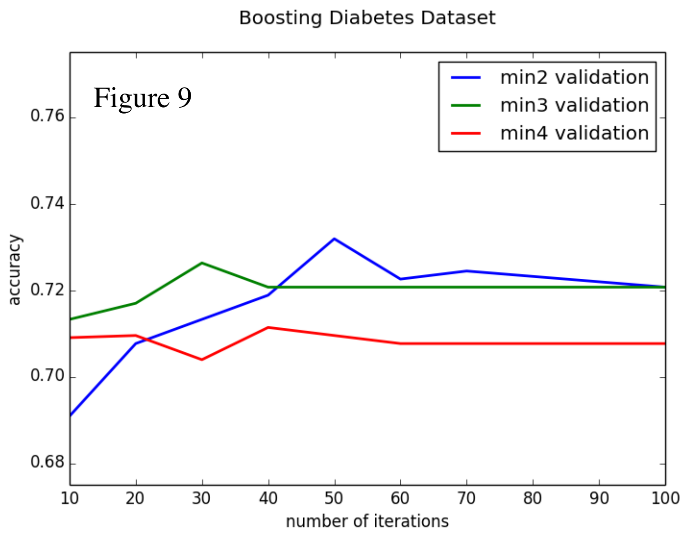


Analysis. For the SVM analysis, the model generated by the linear kernel predicted more accurately than the Gaussian analysis. For the diabetes dataset, the difference between linear and Gaussian kernels was 11.0%. For the wine dataset, the difference was 7.8%. For both datasets, the accuracy rate of the Gaussian model equaled the model's baseline accuracy rate. This is an interesting observation. It implies that the more complex model did a worse job at predicting classifications. This might be because the datasets used for this analysis were fairly straightforward and not complex: there were fewer than 12 attributes, less than 5000 instances, and less than 11 classification classes. Because the datasets were not complex, the less complex SVM model was actually more successful than the complex model at accurately predicting instance classification. Figure 7 and Figure 8 support this hypothesis, as seen in the linear kernel model's higher accuracy rates compared to those of the Gaussian kernel model.

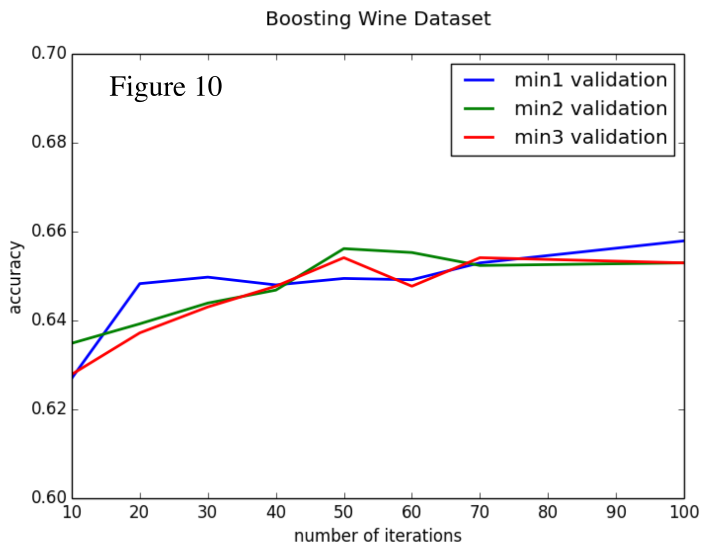
BOOSTING

Description. The base algorithm used for boosting both datasets was the J48 decision tree algorithm. The boosting class was AdaBoostM1 which classifies nominal classes. Overall boosting improved the accuracy of the decision tree algorithm. The parameter modified within the J48 algorithm was the minimum number of instances a leaf is required to have. This parameter was varied over 1, 2, 3, and 4 instances because those were the values that performed the best. To implement boosting, the parameter that was varied was the number of iterations.

Diabetes. The boosting algorithm did mediocre with the diabetes dataset. The worst model was 4.2% above baseline accuracy while the best model was 6.1% above baseline accuracy.



Compared to the other algorithms analyzed, this is lackluster data. The accuracies ranged from 69.1% to 70.7%. As shown in Figure 9, it is interesting to see the variance amongst the min2, min3, and min4 algorithms. The min2 algorithm restricted the J48 base classifier to require 2 instances per leaf, the min3 algorithm required 3 instances per leaf, and so on. According to the decision tree analysis, min4 had the highest accuracy and lowest error rate. However, after application of the boosting algorithm, Figure 9 shows otherwise: after boosting, the min4 algorithm actually had consistently lower accuracies than min2 or min3.



Wine. For the wine data set, boosting did surprisingly well. The best model generated by the boosting algorithm was an entire 20.6% above the baseline accuracy and the worst, or least accurate, model had an accuracy rate 17.6% above the baseline accuracy! The accuracies ranged from 62.7% to 65.7%. It is interesting the note the lack of variance in error among the min1, min2, and min3 algorithms. Although min1 performed better based on the decision tree analysis, this trend is not displayed in the boosting in Figure 10.

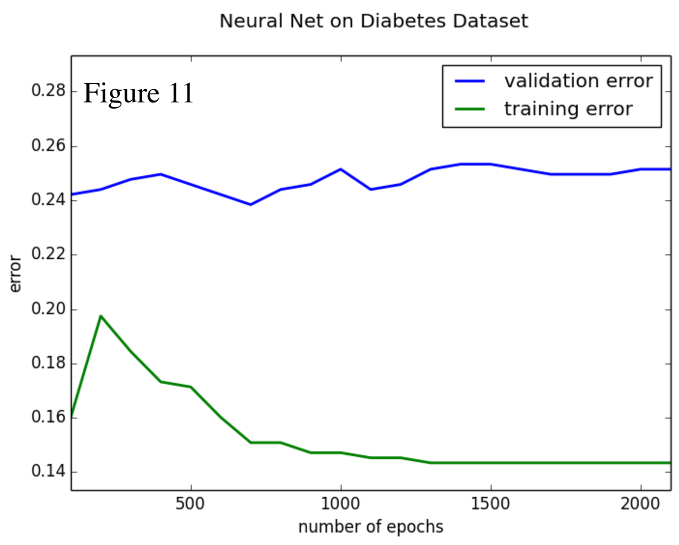
Analysis. Boosting the J48 base class had some very interesting results. For the wine dataset, boosting greatly improved the classifier accuracy, leading to a more than 150% increase in accuracy from 45.1% to 70.7%. This improvement is almost too good to be true and may be due to overfitting and high bias within the model. One way to prevent this in the future is to make the model learn more slowly by setting a low learning rate low so that the complexity of each tree is selected on basis of whether interactions are allowed. This would allow for more complex interactions to be represented more accurately, rather than the model defaulting to compensating for bias through overfitting.

In regard to the diabetes dataset, the variance in accuracy is really interesting, especially when compared to the results of the decision tree. Boosting appears to have actually worsened the accuracy of the min4 algorithm! This may be due to the fact that the diabetes dataset has a smaller number of instances and therefore does not have a complex model to represent. Because of this, the added layer of complexity derived boosting actually confuses the data and decreases the overall accuracy of the model. This result can be seen in Figure 9 and is the opposite of the intended effect of boosting.

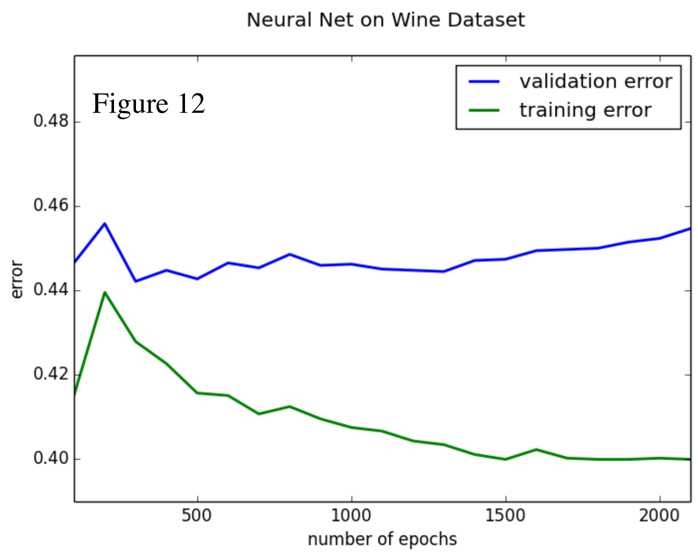
NEURAL NETWORKS

Description. The base algorithm used for both datasets was the MultilayerPerceptron. For both datasets, the signifiant model complexity parameter being varied was the number of epochs. The

MultilayerPerceptron algorithm uses back propagation to classify instances. The number of epochs varied from 0 to over 2000. Of all the algorithms, neural nets generated the most accurate model in comparison to baseline accuracy.



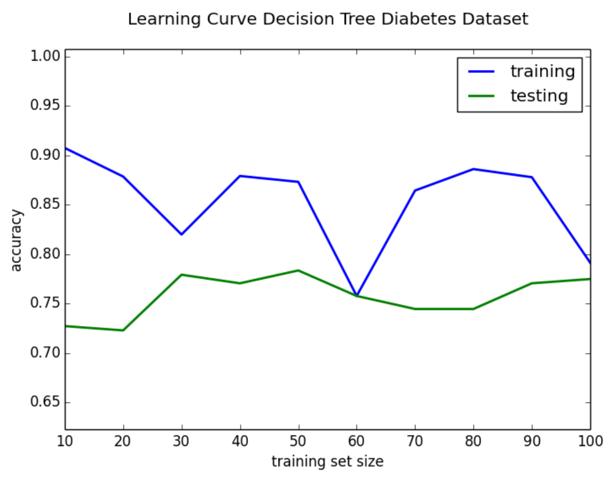
Diabetes. The neural net algorithm performed well on the diabetes set. The lowest error of all the algorithms was achieved with neural nets and had a model accuracy of 77.2%. The epoch number that generated this high accuracy model was 6. The range of accuracy ranged from 75.7% to 77.2%. This is interesting because it means that even the worst neural net model performed better than the baseline for this dataset which was 64.8%.

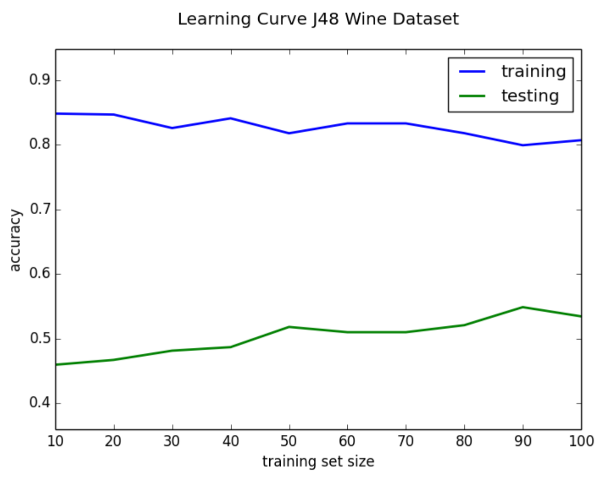


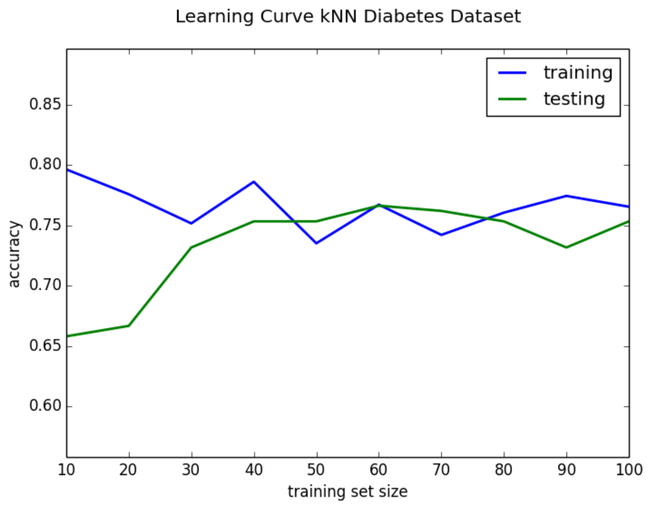
Wine. The neural net algorithm performed much better than the baseline accuracy for the wine dataset. Baseline accuracy was 45.1%, and the highest accuracy model for the wine dataset was 56.8% at number of epochs equaling 2. The range of accuracies did not vary too much and went from 54.4% to 56.8%. Both of these values show that the neural net had a greater than 10% increase in accuracy compared to the baseline. Although a 56.8% accuracy rate still describes a fairly poor classification model, this rate is a significant improvement compared to the baseline accuracy rate.

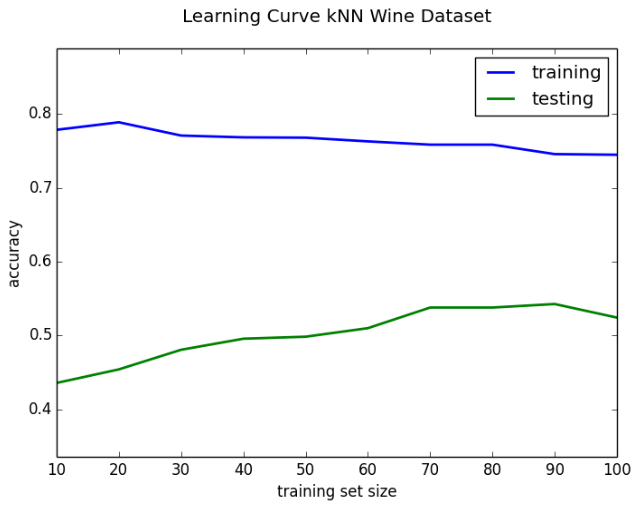
Analysis. The neural net algorithm performed the best of all the learning algorithms by showing the highest accuracy rate for both datasets. For the diabetes dataset, the neural net generated a whopping increase of 12.8% accuracy compared to the baseline. For the wine dataset, the neural data displayed a similarly high increase in accuracy of 11.7%. Neural net algorithms are typically trained using an iterative optimization method. In this case, that method was gradient descent which employs back propagation. At the beginning of the analysis, multiple parameters were tested to see which would affect the model accuracy the most. A few that were played with include learning rate, momentum, number of iterations, and number of epochs. Ultimately, the parameter varied for the final experiment was number of epochs. An epoch is a single pass through the entire training set, followed by testing of the verification set. The results of these experiments can been seen in Figure 11 and Figure 12. As seen in these graphs, as the number of epochs increase, the more accurate the generated models become. There are hints of overfitting among the lower values of number of epochs as seen in the random spikes in error rate. This might be due to noise in the data and can be fixed by regularizing the models.

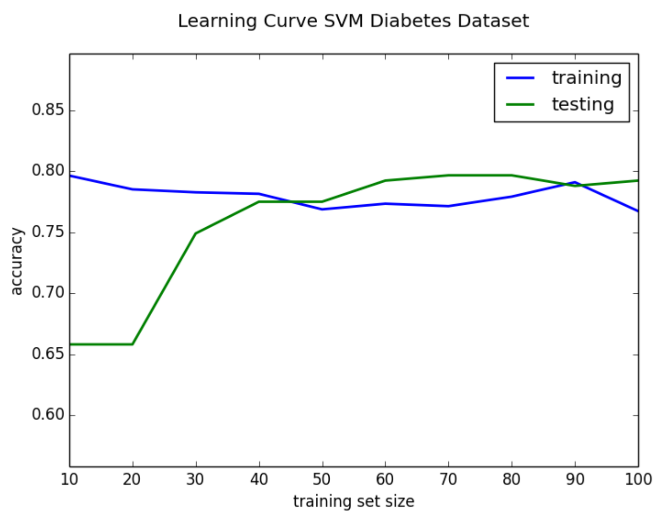
Learning Curves

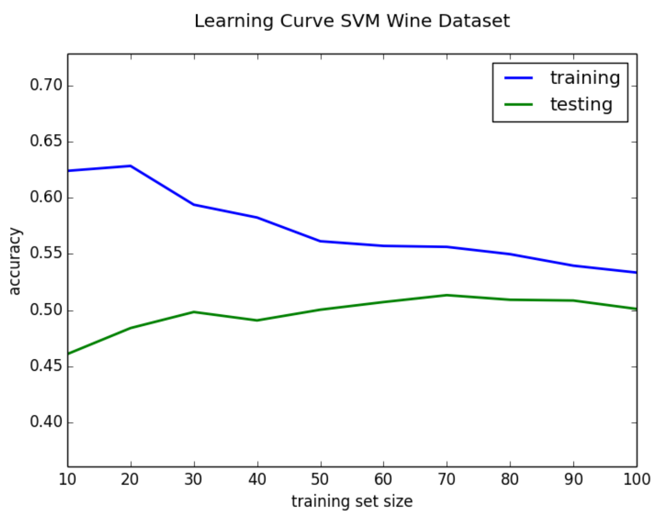


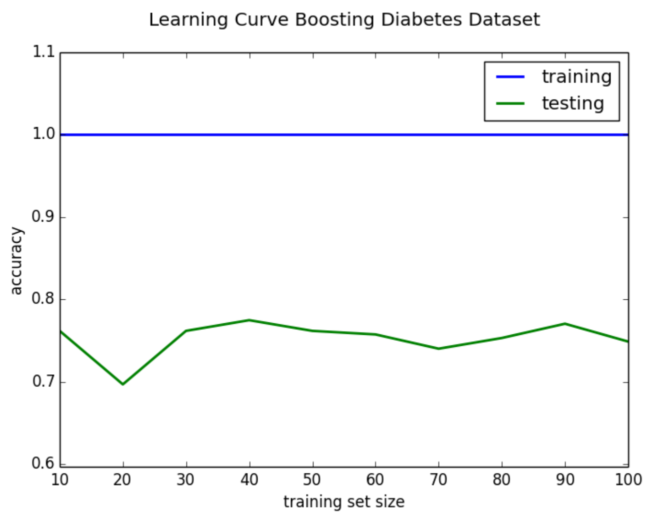


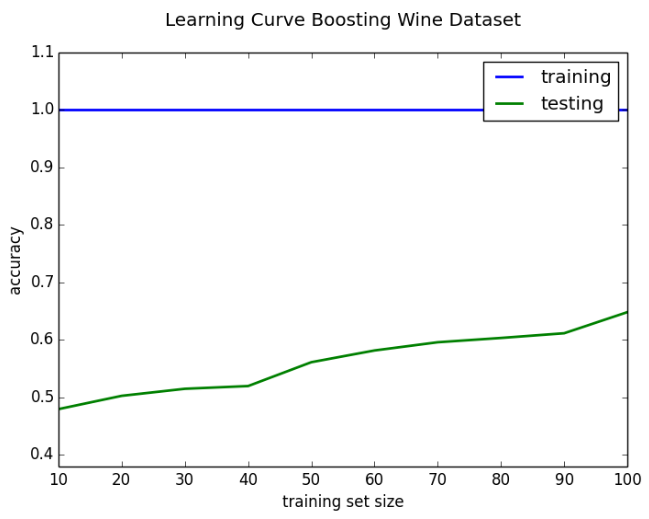


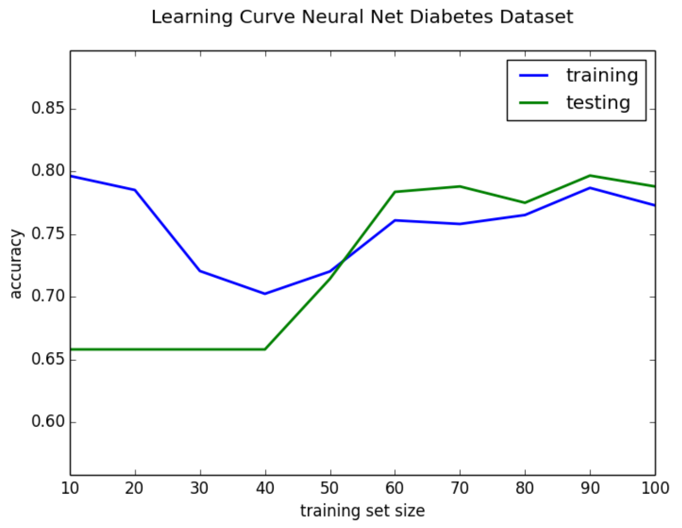


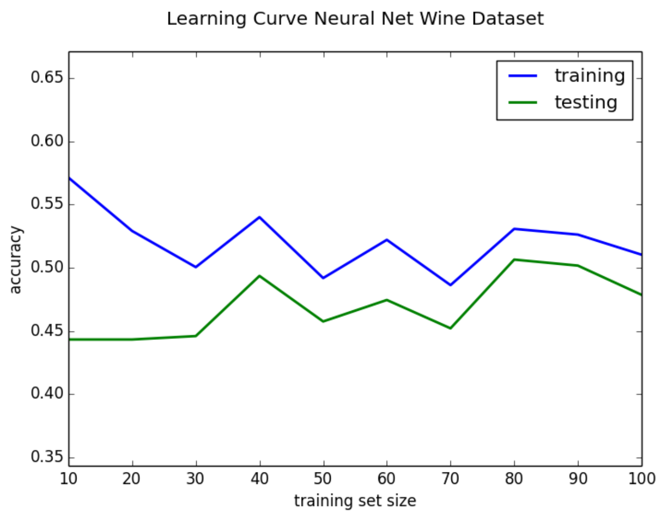

Learning Curve SVM Diabetes Dataset
training
testing
accuracy
training set size
0.60
0.65
0.70
0.75
0.80
0.85
10
20
30
40
50
60
70
80
90
100


Learning Curve SVM Wine Dataset
training
testing
accuracy
training set size
0.40
0.45
0.50
0.55
0.60
0.65
0.70
10
20
30
40
50
60
70
80
90
100


Learning Curve Boosting Diabetes Dataset
training
testing
accuracy
training set size
0.6
0.7
0.8
0.9
1.0
1.1
10
20
30
40
50
60
70
80
90
100


Learning Curve Boosting Wine Dataset
training
testing
accuracy
training set size
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
10
20
30
40
50
60
70
80
90
100


Learning Curve Neural Net Diabetes Dataset
training
testing
accuracy
training set size
0.60
0.65
0.70
0.75
0.80
0.85
10
20
30
40
50
60
70
80
90
100


Learning Curve Neural Net Wine Dataset
training
testing
accuracy
training set size
0.35
0.40
0.45
0.50
0.55
0.60
0.65
10
20
30
40
50
60
70
80
90
100