Rohith Krishnan

Supervised Learning Analysis

Datasets

Student-Scores-20. This data set was created from a survey of secondary school students in Math and Portuguese language courses and contains information on a student's gender, family background, weekend and daily alcohol consumption, and grades at different points during the semester. The classification problem was to predict the final scores of the students (noted as 'G3' in the dataset) given the various social statistics of the students. The scores in the dataset were on a numeric scale from 0 to 20 which means there are twenty different classes for each classifier to predict. For an additional comparison on accuracy I modified the dataset to have only four classes (called Student-Scores-4) for the final score where each class represented a range of scores on the original grading scale (i.e. grades in the range from 0 to 4 were mapped to a '1', ranges from 5 to 10 were mapped to a '2' etc.). This dataset contained 32 features and 1044 instances. A distribution of the different classes is shown below. This distribution shows that many samples have scores between 10 and 15 and is skewed slightly to the left.

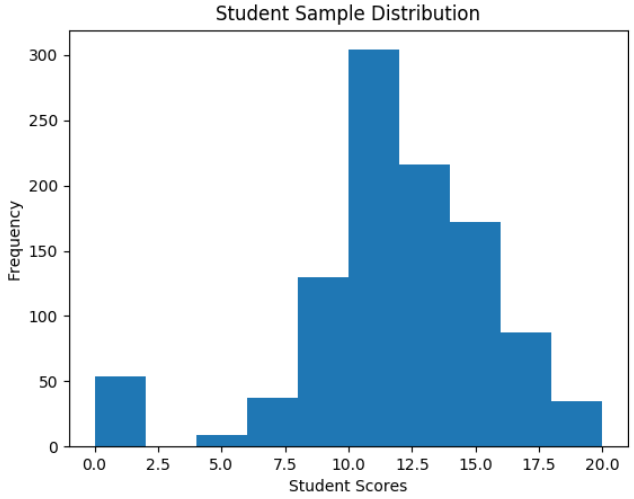


Figure 1. Distribution of Student Scores

Wine Quality. This dataset contains physiochemical data about different types of Portuguese white wine and contains a quality score of the wine sample on a scale from 0 to 10. The classification problem here was to predict the quality score of the wine given the different chemical data such as acidity, pH levels. This problem contained only ten different classes for each supervised learning model. The dataset contains 12 features and 4899 instances. The distribution of the quality scores is not well balanced as there are more instances of scores between 5 to 7 and very few scores on the lower or higher end of the spectrum.

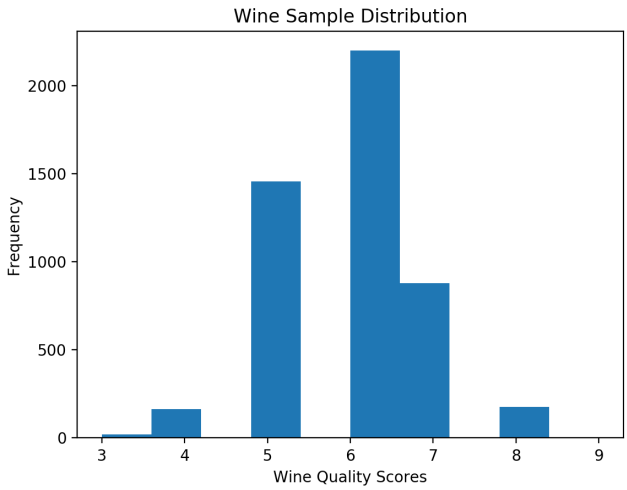


Figure 2: Distribution of Wine Quality Scores

Why These Datasets are Interesting

Before applying the different algorithms, I found the Student Scores data set interesting for the social implications that come from it. This data can help universities determine how different students will perform given their background and alcohol habits and to predict how likely a student is to fail a certain course. This will give universities a better metric of what external factors contribute to student performance and what resources need to be made available to students to overcome these factors. The wine dataset is of interest to wine enthusiasts and collectors of high end wine. Having a good supervised learning model that can predict the quality of a wine based on its chemical makeup can help consumers determine if they are paying a fair price for a certain quality of wine.

In terms of supervised learning, I found the Student dataset interesting because it did not achieve an accuracy score above fifty percent for any of the supervised learning techniques used on it. These relatively low accuracy scores could be caused by a lack of examples compared to the number of features in the dataset. As there are only 32 features and 1044 examples, the different supervised learning algorithms struggled to generalize from the training set and were affected by the curse of dimensionality. Additionally, the large number of classes in the student dataset contributed to the low accuracy from the different models (accuracy significantly improved when I reduced the number of classes down to four). The wine dataset was interesting because it is very unbalanced in terms of the distribution of target values but it does have many examples relative to the number of features.

Decision Trees

For the decision tree experiments, I used sci-kit learn's Decision Tree Classifier which uses the CART (Classification and Regression Trees) algorithm. This algorithm is like ID3 except that it can dynamically define discrete attributes from continuous numerical values by splitting the attribute into sets of discrete intervals. The algorithm converts the trained trees into sets of if-then rules. A form of post-pruning is performed on the set by seeing if removing a rule's precondition will improve the accuracy of the rule. This form of pruning should improve the test accuracy across the two datasets as it reduces the number of rules used to split nodes in the tree and should reduce the amount of overfitting

that a decision tree would have on the training set. For both datasets, I ran five-fold cross-validation to find the best maximum depth for the decision tree model. After determining the depth that gave the highest cross validation score, I ran the model on different sizes of the training set with that maximum depth and plotted training and testing scores for the different sizes.

Student Dataset Results (4 classes):
Test Accuracy: 0.8439490445859873

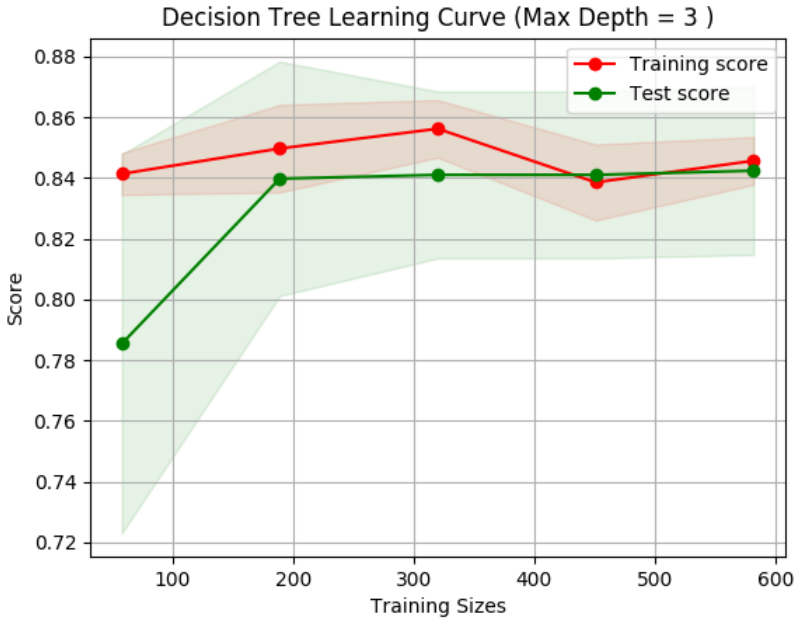


Figure 3: Student Dataset Learning Curve (4 classes)

Student-Scores-20
Test Accuracy: 0.4299363057324841

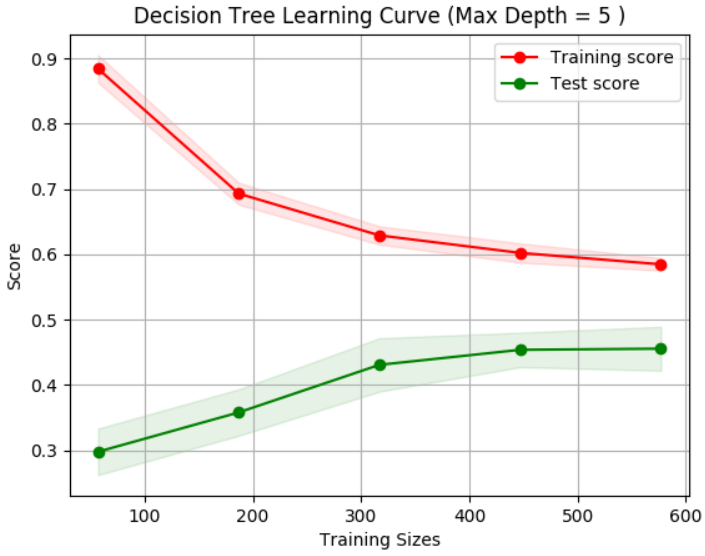


Figure 4: Student-Scores-20 Decision Tree Learning Curve

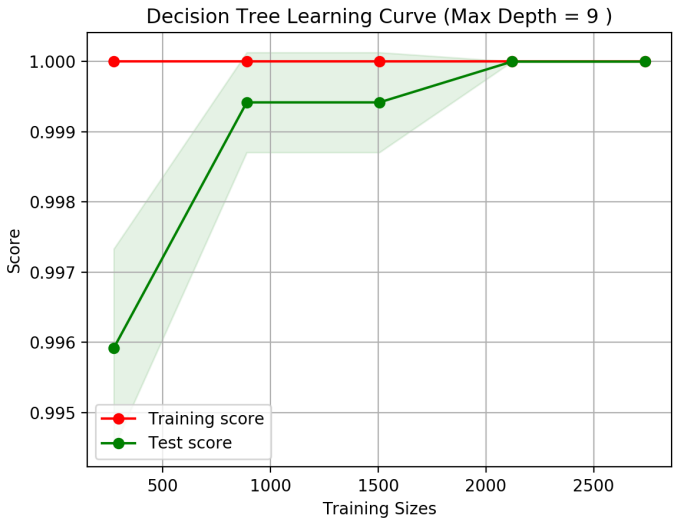


Figure 5: Wine Dataset Learning Curve.

In the learning curve above for Student-Scores-20, we can see that the training scores and the testing scores are converging to a relatively low score as the training set size increases. This trend indicates that the decision tree learner with a maximum depth of five is not benefitting from increasing the training set size and is suffering from high bias and is under fitting. Another important note comes from the precision and recall scores (can be found in dtreeresultsstudent.txt under /graphs/student) which show that the decision tree is more likely to predict final scores between that are either zero or between 10 and 12, matching the distribution of scores from the dataset. This means that the model is unable to generalize well and performs poorly on unseen values from the testing set. On the other hand, the wine dataset's training scores remained unchanged as the training sizes increased which indicates that the model is severely overfitting to the training set.

The results for the Student-Scores-20 could be caused by the fact that there are twenty classes which makes the task of identifying which score a student should get given his or her features more difficult. The significant improvement in accuracy on the test set with fewer number of classes shows how the large granularity of the classes impacted the performance of the decision tree model. Another probable cause for these results could be the distribution of scores in the dataset. Since most the scores were between 10 and 15 as shown in Figure 1, the decision tree model would have a preference bias to predict scores within that range and would have less accuracy predicting scores that fell on extreme lows or extreme highs of the grading scale. Changing the classes to cover different ranges of student scores helps overcome this bias as it can encapsulate the extreme or rare scores in the dataset.

For the wine dataset, the high accuracy can be attributed to the dimensionality of the dataset and the distribution of the target values. Since there are only twelve attributes to split on and nearly 5000 examples, the decision tree has enough examples to generalize as training set size increases. The overfitting to the training set could be the result of the distribution of the dataset. Since most of the quality scores are between 5 and 7, the model will tend to predict scores within that range and would struggle to predict very low quality wines or very high quality lines. One way to overcome this overfitting would be to have a sampling that covers a higher range of quality scores.

Neural Networks

For the neural network experiments, I used sci-kit learn's Multi-Layered Perceptron classifier. For both datasets, I ran five-fold cross validation while varying the maximum number of iterations. After finding the iteration number with the highest cross validation score, I ran the network on different sizes of the training set. For both datasets, the neural networks from sci-kit learn had 100 perceptrons and only one hidden layer by default and a logistic activation function.

Student-Scores-20

Test Accuracy: 0.3057324840764331

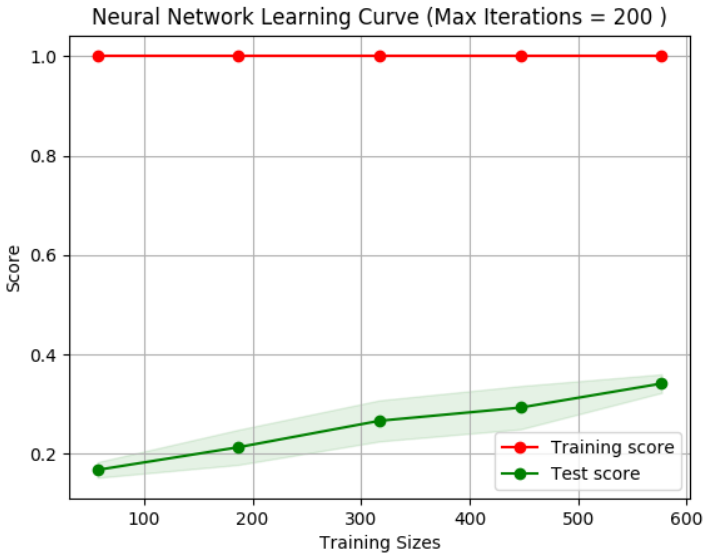


Figure 6: Student-Scores-20 Neural Network Learning Curve

Student Dataset (4 classes)
Test Accuracy: 0.7929936305732485

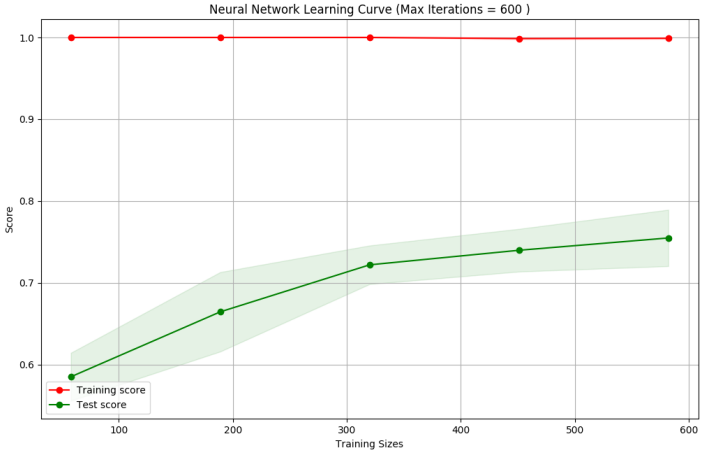


Figure 7: Student-Scores-4 Neural Network Learning Curve

Wine Dataset
Test Accuracy: 0.995238095238

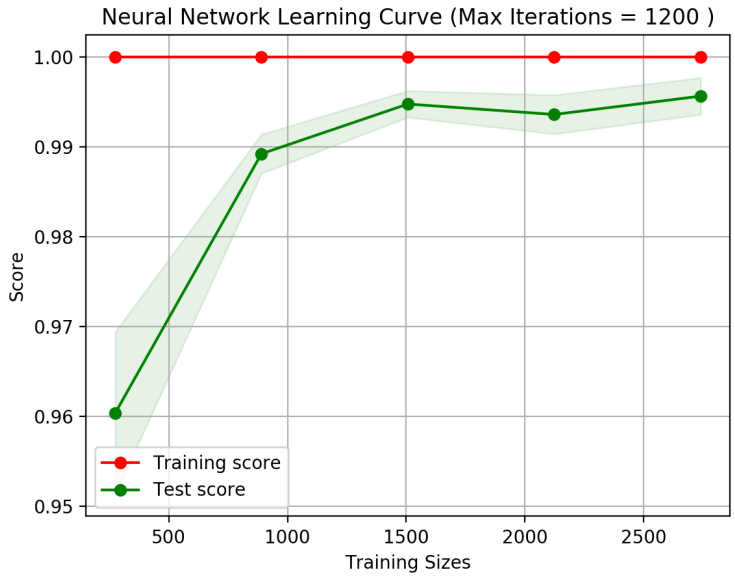


Figure 8: Wine Dataset Neural Network Learning Curve

Neural Networks performed considerably worse on the student dataset and were slightly less accurate on the wine dataset when compared to the decision tree models. For both versions of the student dataset, the neural network models suffered from high variance as there was a large difference between the training and testing scores for both datasets. Also, the training scores for the two datasets remains the same regardless of the number of examples, which is a strong indication of overfitting to the training set. The testing scores for the two versions did increase as training set size increased indicating that more data would improve accuracy, but it does seem that Student-20 would need very large amounts of data given the slow increase of its testing scores.

One possible reason that a neural net had lower accuracy than decision trees was that there were certain attributes in the student dataset that were non-numeric and were encoded into numbers using label encoding. Certain attributes such as whether a student had internet access were encoded from yes-no values to integer values such as 1 and 2. Encoding these attributes to integers could confuse a neural network as it will interpret these values without the context of their real-life values. Therefore, it will weigh attributes with larger encoded values with higher weight and try to predict target values that correspond to higher values of the label encoding. This will result in the model failing to generalize to the real-world representations of the encoded data. One way to overcome this confusion would be to use a One-Hot encoding scheme instead of label encoding where different values of a feature get a 1 or 0 depending on if the sample feature has that value or not. (ex: 1 or 0 if a student's father is a teacher, 1 or 0 if the father is in service etc.) This would allow the neural network to better distinguish between different values of non-numeric features.

Boosting

Student-Scores-4
Test Accuracy: 0.7356687898089171

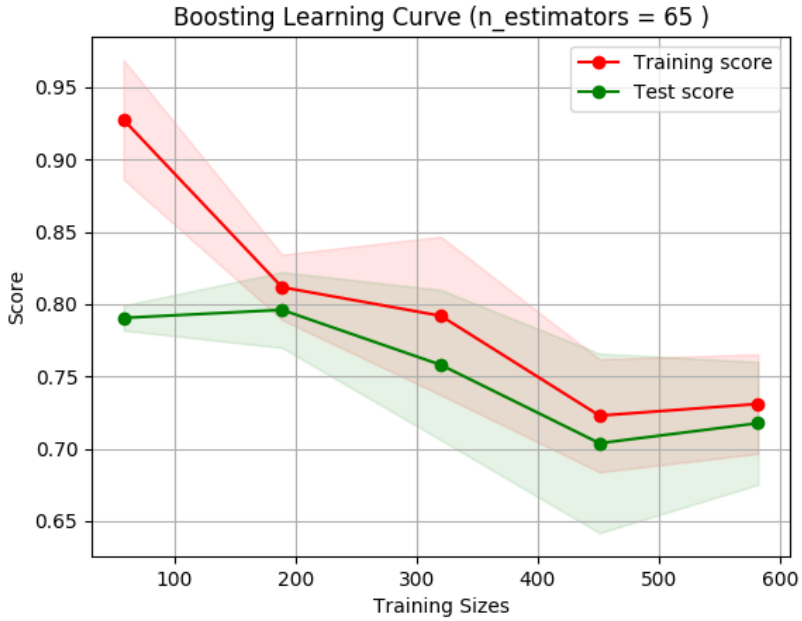


Figure 9: Student-Scores-4 Boosting Learning Curve.

Student-Scores-20
Test Accuracy: 0.3471337579617834

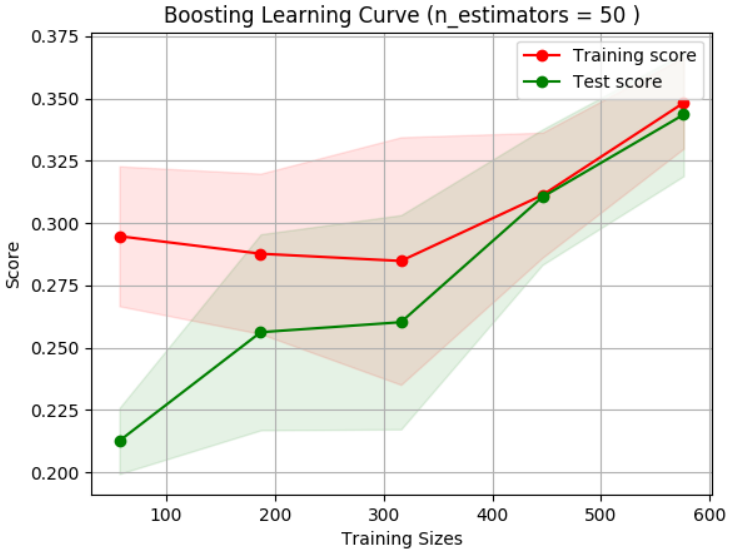


Figure 10: Student-Scores-20 Boosting Learning Curve

Wine Dataset
Test Accuracy: 0.761224489796

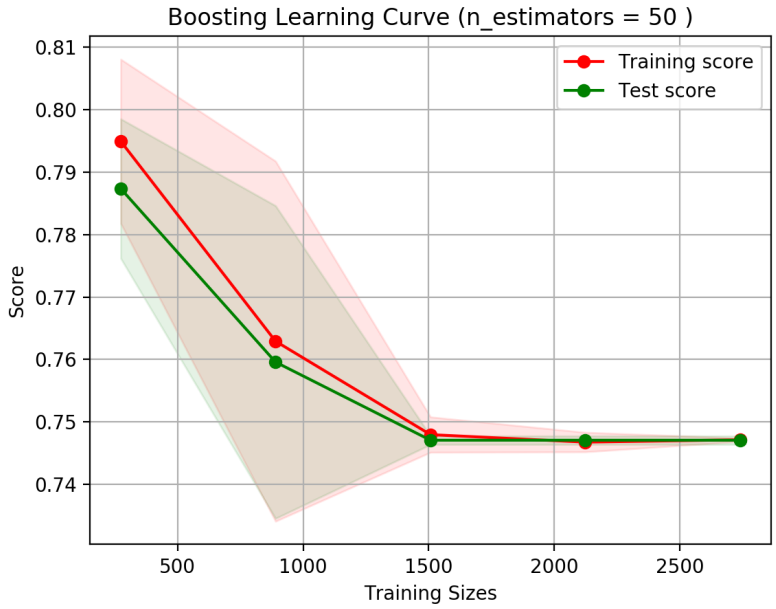


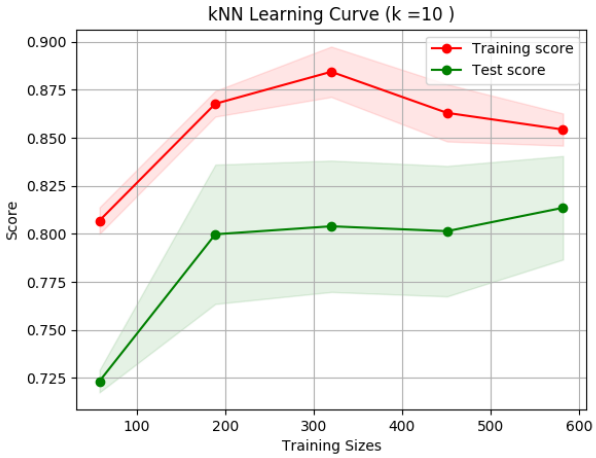
Figure 11: Wine Dataset Boosting Learning Curve

I used sci-kit learn's AdaBoostClassifier which uses Decision Trees as the base estimator that it builds the boosted ensemble on. For both datasets, the boosted decision trees were less accurate than compared to using one decision tree classifier and compared to using a neural net. The Wine Quality dataset and Student-Scores-4 both saw the training and testing scores decrease with larger portions of the dataset which shows that the boosted trees were underfitting and were unable to learn or generalize from the testing data. It seems that the boosted trees suffered from high bias across both datasets which can be attributed to the high clustering of samples around a small range of values as shown in Figure 1 and Figure 2.

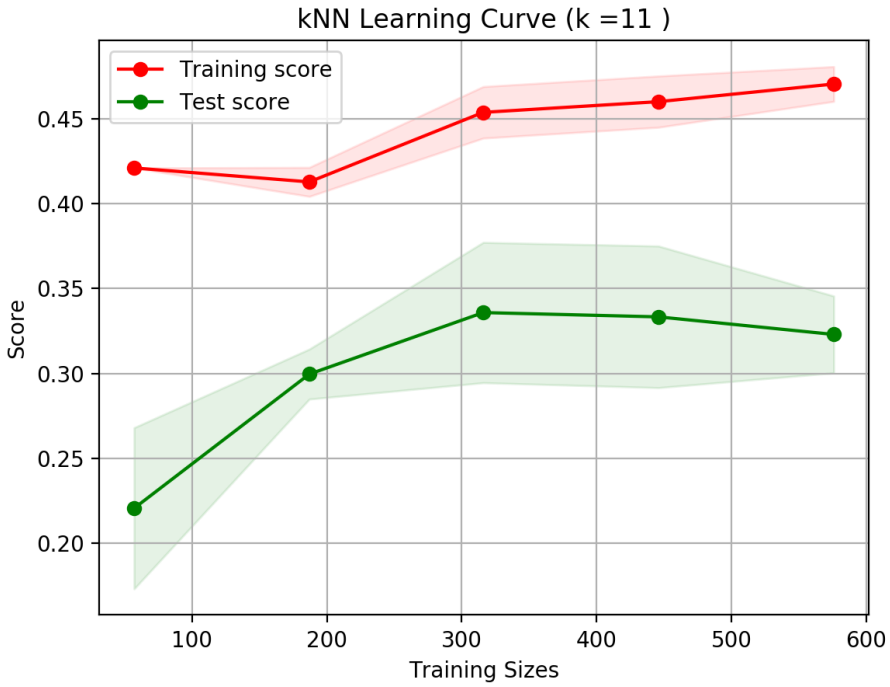
kNN

For both datasets, cross-validation was performed to find the best value for k. Once the best k value was found, the classifier was run on different sizes of the training set

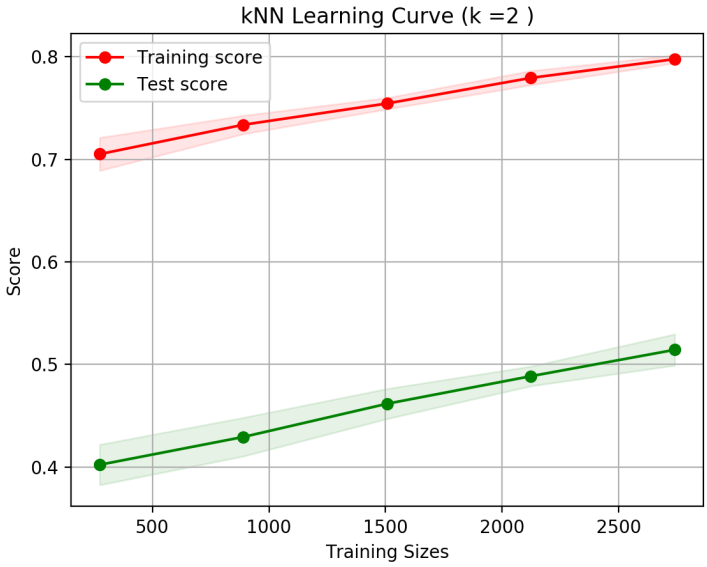
Student-Scores-4:
Test Accuracy: 0.7993630573248408



Student-Scores-20:
Test Accuracy: 0.305732484076



Wine Dataset
Test Accuracy: 0.545578231293

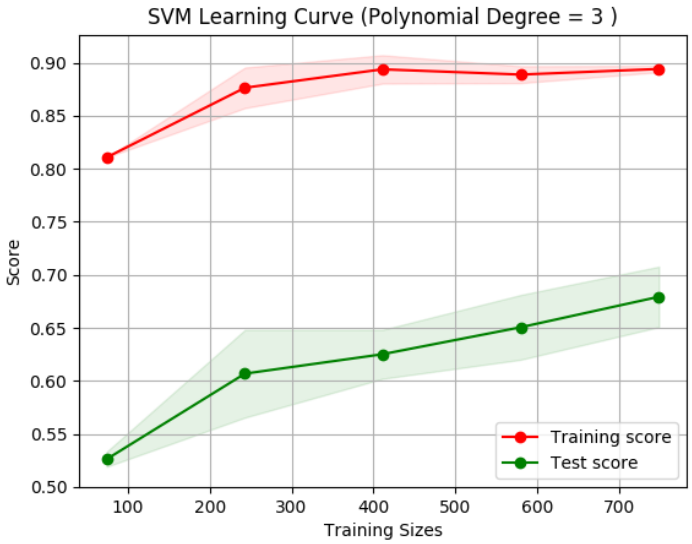


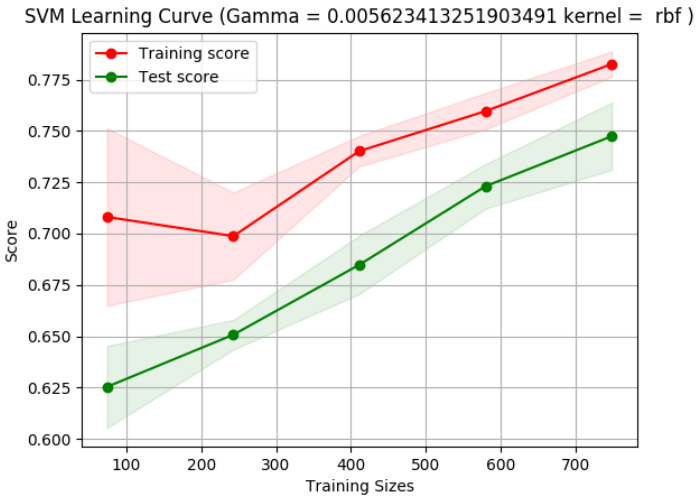
For the wine dataset, I expected kNN to perform as well as the decision tree and neural network models given that the dataset has many examples relative to the number of features, however kNN had the lowest test accuracy across the five different supervised learning models. The learning curve for the data set shows a large amount of variance as there is a large difference between the training and testing curves. The poor performance of kNN on this dataset could be due to the high clustering of data between scores of 5 and 7. The testing scores did increase with larger sizes of the training data which does follow the general behavior of kNN performing better with larger dataset sizes. Another way that kNN can improve on this data set is to have a more random sampling of different wine quality scores so it can account for larger ranges in quality.

I expected kNN to not perform as well on the student dataset compared to the previous techniques since there were many features and a relatively limited number of examples. Because of the large number of features of this dataset, the kNN algorithm could have been fooled by attributes that are not as relevant to the target value and would be affected by the curse of dimensionality. Another cause for error could also be the choice to use Label Encoding for the non-numeric features. Since label encoding converts the non-numeric features to whole number integers, it is hard to determine if the "distances" calculated between two classes accurately represent real world distances (i.e. how can you determine distance between different occupations of a student's parents?). Using a One-Hot encoding scheme as described earlier could help eliminate some of this confusion.

Support Vector Machines

I used sci-kit learn's Support Vector Classifier with a Radial-Basis kernel function (RBF) and a polynomial kernel function. Cross validation was used to find the best kernel coefficient or gamma value for the RBF kernel and the best degree for the polynomial kernel.

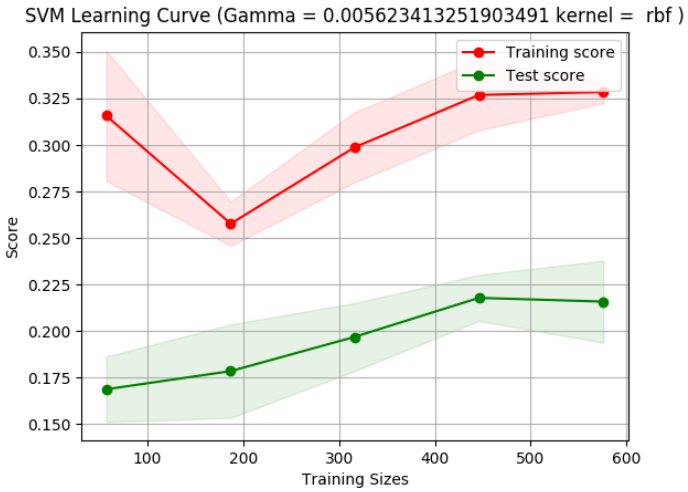


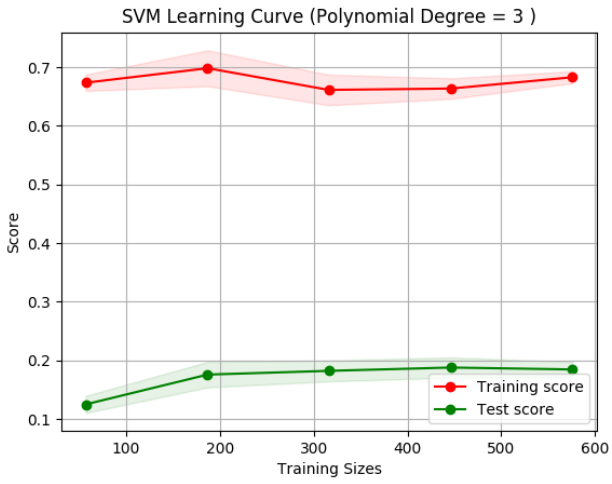


Student-Scores-4 SVM Learning curves with Polynomial and RBF kernel functions
SVM-RBF Test Accuracy: 0.8380952380952381
SVM-Polynomial Test Accuracy: 0.6952380952380952

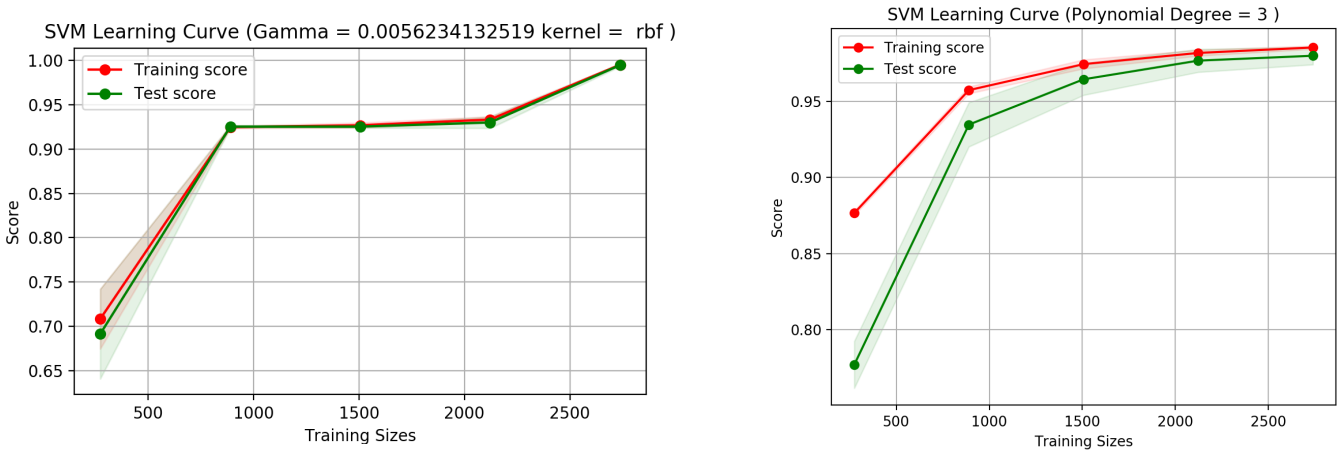
The SVM with a polynomial kernel function experienced high variance as shown by the large difference between the training and testing curves and was far less accurate on the test set when compared to the SVM with a radial-basis kernel function. It seems that the polynomial kernel function was less helpful in transforming the features into a dimension where the data can be linearly separated, which could be caused by the large number of features in the dataset. The RBF kernel could handle the large number of features by transforming each feature into a higher dimensional feature space and could capture the complexity of the data a lot more accurately compared to the polynomial kernel.

Student-Scores-20
SVM-Polynomial Test Accuracy: 0.15605095541401273
SVM-RBF Test Accuracy: 0.19745222929936307





Wine Dataset
SVM – RBF Test Accuracy: 0.991836734694
SVM - Polynomial Kernel Test Accuracy: 0.978911564626



SVM Learning Curves for the Wine Dataset.

SVM's performed the worst on the original student dataset when compared to the previous four techniques. The low accuracy and high variance of both SVM's could be caused by the high clustering of student scores which would make it harder to transform the scores into a linearly separable space. However, the SVM models had higher test accuracy on the wine dataset when compared to the two versions of the student dataset, even though the wine dataset also has a high amount of clustering around a small range of values.