Peter Lee
Professor Byron Boots
CS4641: Project # 1 - Supervised Learning
Username: plee99
September 25, 2018

Introduction:

Overview: In this document, I will analyze a few supervised learning algorithms as they relate to two data sets that I have chosen. I hope that the visualizations of training performance aid in the analysis of these methods. I have learned a lot and have become more familiar with a few of these algorithms as a result of this project.

Selection and Description of Datasets:

The two datasets that I chose for this project was the Breast Cancer Wisconsin (Diagnostic) Data Set and the Sloan Digital Sky survey data set. Both of these problems are classification problems with multiple variables. The datasets were both sufficiently large enough (both over 1k samples) to be split into a training set and testing set. The testing set was never used to train the model and was used as a metric to estimate the overall effectiveness of the learning algorithms. The hope of splitting the data was such that the information contained in the training set was enough such that the model was able to generalize the information to correctly classify and predict the labels of the test set, which the machine learning model has never seen. I split the data such that the training and testing set were consistent between all of the various machine learning models. The datasets were the following:

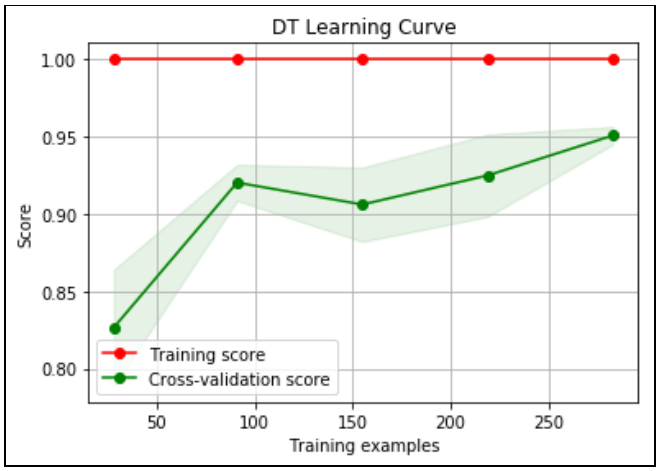
Breast Cancer Wisconsin (Diagnostic) Data Set - Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.They describe characteristics of the cell nuclei present in the image. [1] The 32 features were associated with a label that indicates whether the breast contained a benign or malignant tumor. The features describes various statistical measures associated with the image of the breast mass such as the symmetry, concavity, and area of the region. The data was preprocessed from images and the dataset was a compilation of various statistics from the images. I scaled the data before implementing it in the code.

Sloan Digital Sky survey data set - The dataset consists of 10,000 observations of space taken by the Sloan Digital Sky Survey. Each observation is described by 17 feature columns and 1 target column which identifies the observation to be either a star, a galaxy or a quasar. [2] The dataset has multiple variables as well as 3 possible labels. The data was already preprocessed from images prior to use.

Training and Testing:

Decision Trees (DT) for the Breast Cancer dataset

Decision trees are a simple and highly intuitive supervised learning algorithm. The basic idea is that data is entered into a tree such that at each node, there is a test of the attributes which determines which branch the data is then passed into, at the end of the tree, a classification or 'decision' is made about what label the data represents.



The following is the classification report of the Decision tree.

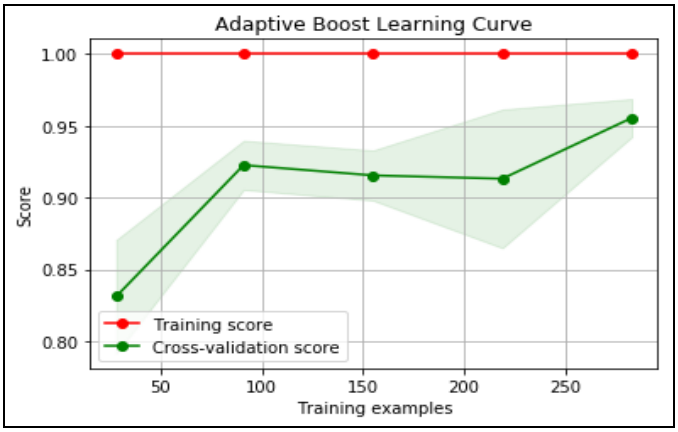
	precision	recall	f1-score	support
0	0.97	0.89	0.92	97
1	0.80	0.93	0.86	46
avg / total	0.91	0.90	0.90	143

Adaptive Boosting with Decision Trees (Boost) for the Breast Cancer dataset

Adaboost is able to improve the performance of decision trees by improving the model to better adapt and fit for the misclassified data.

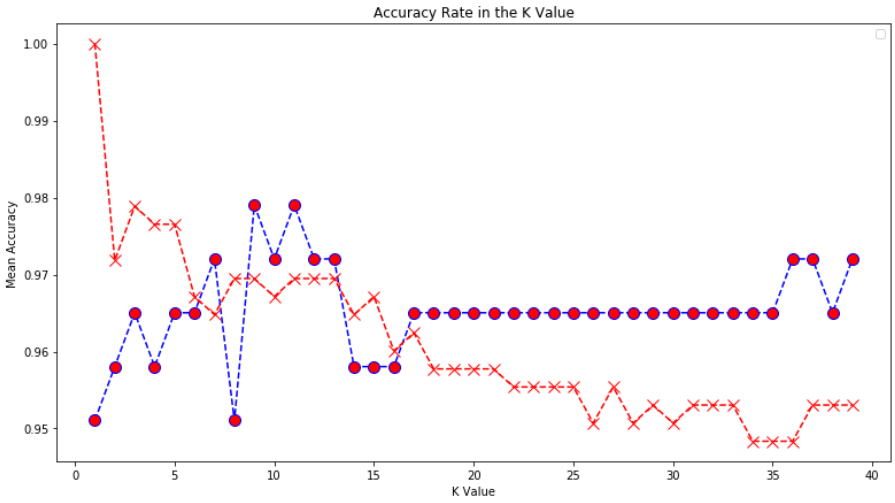
	precision	recall	f1-score	support
0	0.98	0.98	0.98	97
1	0.96	0.96	0.96	46
avg / total	0.97	0.97	0.97	143

Here is a classification report of the Adaboost decision tree. As you can see the average precision, recall and f1-score of the Adaboost decision tree is much better than the decision tree. The adaboost and decision tree had a precision of 97% and 91, a recall of 90% and 97%, and an f1-score of 97% and 90% respectively. Overall, Adaboost did much better than a traditional decision tree within this dataset.

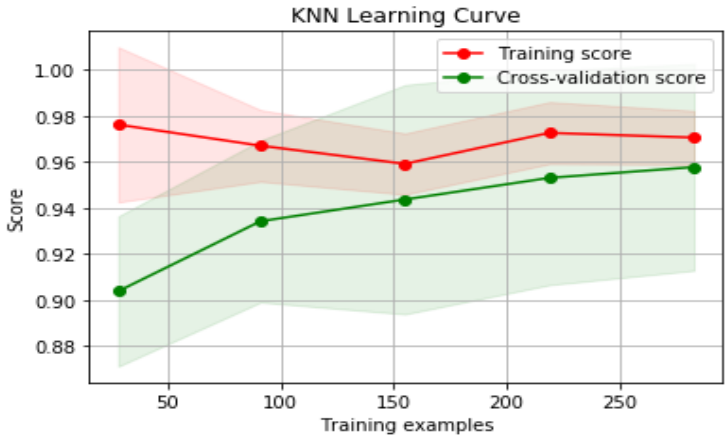


K - Nearest Neighbors (KNN) for the Breast Cancer dataset

K nearest neighbor is a simple and powerful learning algorithm. The intuition is that data in multiple dimensions tends to cluster and therefore by looking at a number of the nearest neighbors with respect to features, the data's label should also be similar to its neighbors. In order to 'pick' the k value of the algorithm, I plotted the test accuracy as a function of k.



According the test set accuracy, the optimal k value would be either 9 or 11. I chose 9 as it was taught in class "the simpler the better." Interestingly, as the k value increased, the test The following is the learning curve of the K-nearest neighbor with respect to the number of training examples that it was exposed to. An interesting thing that is observed is that unlike the other algorithms, the K-nearest neighbor never becomes almost perfect at classifying the training data. This makes sense if some of data of different labels are also near the other data.

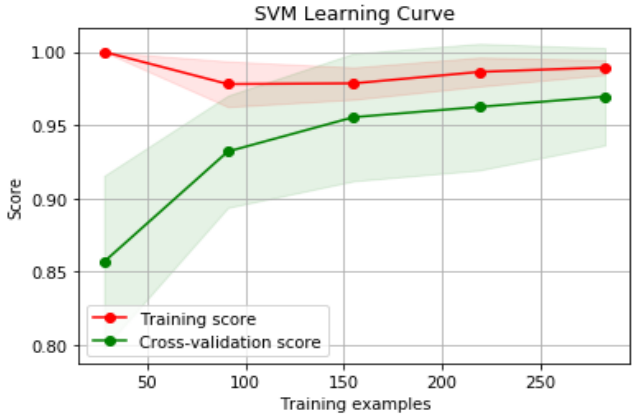


The following is a classification report, based on all the metrics, the KNN has outperformed all the other algorithms in its precision, recall, and f1-score.

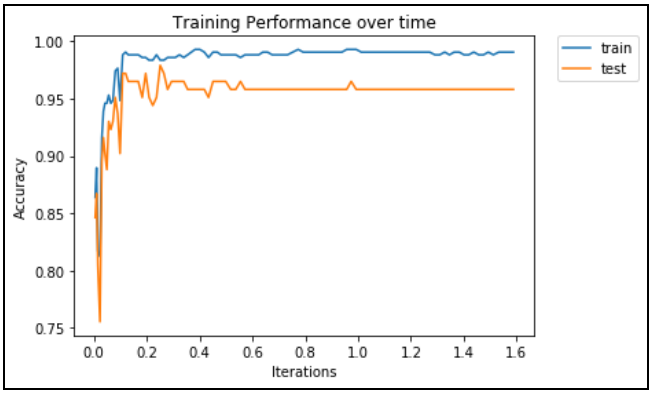
	precision	recall	f1-score	support
0	0.98	0.99	0.98	97
1	0.98	0.96	0.97	46
avg / total	0.98	0.98	0.98	143

Support Vector Machines (SVM for the Breast Cancer dataset):

The underlying intuition of an SVM is that if we imagine a map of hyperparameters, we can create lines within hyperspace to separate and classify the data. The lines are weighted in such a way that the nearest data points 'support' the line by almost trying to balance the line in the boundary. The support vector machine for the Breast Cancer Dataset did very well, we can see that as SVM is fed more training examples that the overall generalization becomes much better.



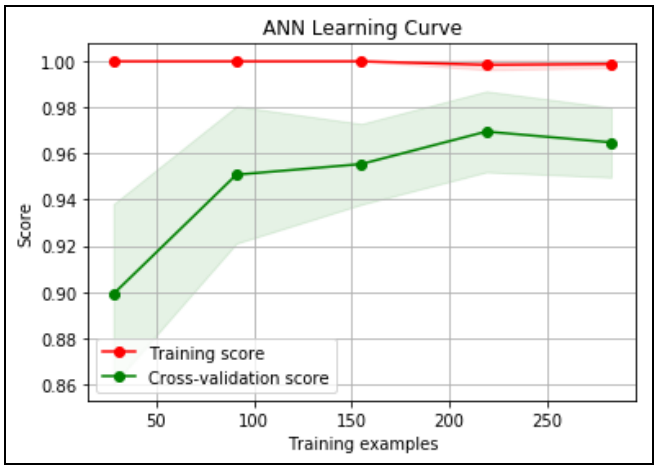
Below is the training performance over time. It can also be observed in the training performance over time (measured in seconds) that the model stops improving at 0.2 seconds and 'flat-lines' and fails to improve as it has optimized itself to the maximum extent. It even decreases as time goes on slightly, suggesting that the model has possibly overfit.



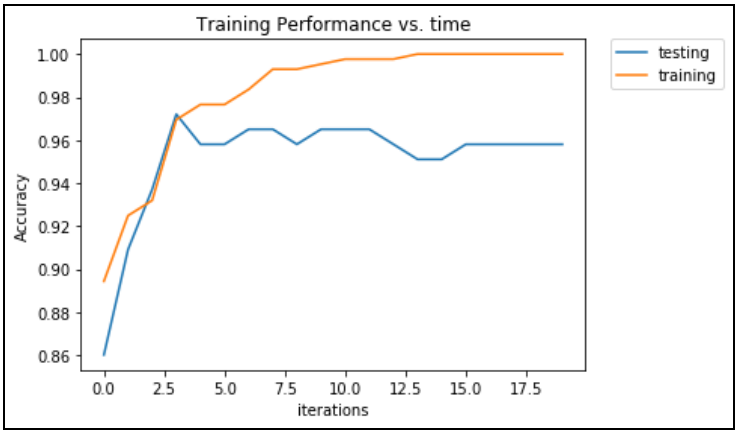
	precision	recall	f1-score	support
0	0.98	0.96	0.97	97
1	0.92	0.96	0.94	46
avg / total	0.96	0.96	0.96	143

Artificial Neural Networks (ANN) for the Breast Cancer dataset

The Artificial neural network is a very power machine learning algorithm that attempts to replicate how the neurons of our brains work. Each layer has individual nodes where the inputs of one layer undergo a multiplication by weights and an addition of a bias term through an activation function. The neural network improves itself through a process known as backpropagation. Here in the following chart, you can see how a neural net improves its performance by increasing the training examples.



Below is the training over time. As you can see, the testing accuracy peakes around the 3rd iteration. This is a drawback of training over too many iterations, this phenomenon is known as over-fitting where the model becomes so good at evaluating only the training data that it is no longer as good at generalizing broad datasets.



Finally, I created a classification report of the ANN and found the results to be ok relative to the other training algorithms.

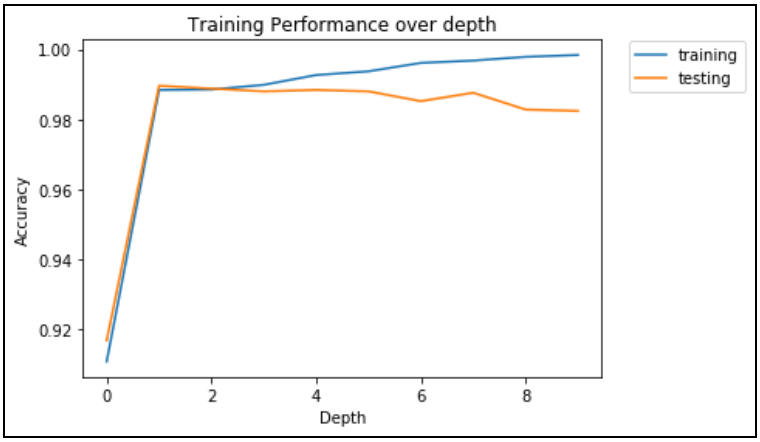
	precision	recall	f1-score	support
0	0.97	0.97	0.97	97
1	0.93	0.93	0.93	46
avg / total	0.96	0.96	0.96	143

Summary of the Breast Cancer dataset:

Overall, the Adaptive Boosted Decision Trees did the best with the data set in terms of the precision, recall, and f1-score. I think It did very well because of an underlying clustering structure in the hyperspace which would enable decision trees good accuracy. This would make sense as people with similar severities of tumours should also have similar attributes such as discolouration and size and a Adaptive Boosted Decision Tree would capture that information.

Decision Trees (DT) for the Sloan Digital Sky dataset

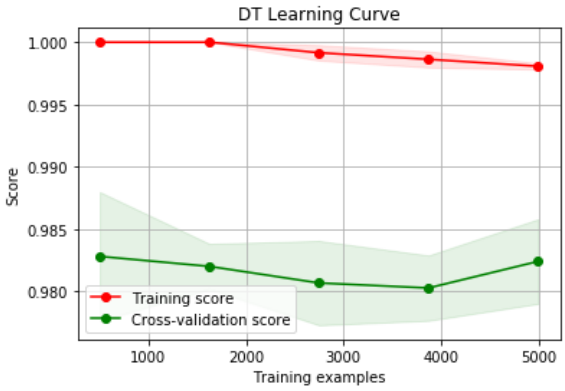
Overall the Decision Trees performance was very good for the Sloan Digital dataset for classifying between the different stellar formations. Below is a plot relating accuracy to the depth of the decision tree. We can clearly see that as the decision tree is ‘deeper’ as in there are more layers of decisions that could be made, the performance decreases. This is most likely due to overfitting where the training accuracy increases at the expense of the test accuracy. The decision tree becomes very adept at analyzing the training set and fits to the nuances of that training set such that it is unable to generalize the classification to the testing set, which the model has never been tested on.



Using the optimal depth of about 3 layers, I created a Decision Tree and evaluated its performance on the test set. The overall results were fantastic. It did not perform so well with QSO or Quasars. Quasars account for only about 9% of the dataset so that may play a role in its poorer performance.

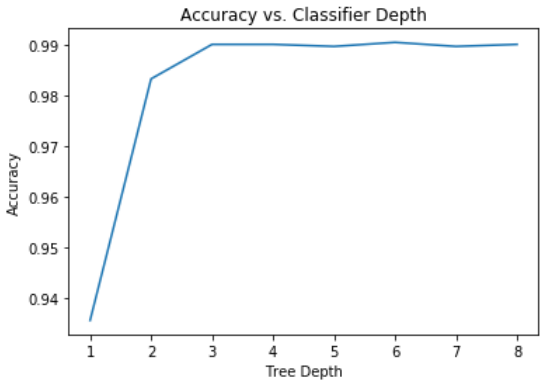
	precision	recall	f1-score	support
GALAXY	0.99	0.99	0.99	1256
QSO	0.97	0.93	0.95	201
STAR	1.00	1.00	1.00	1043
avg / total	0.99	0.99	0.99	2500

Below is a graph of the performance of a decision tree in relation to the amount of data it is trained on. Interestingly, the cross-validation score, a metric to observe its generalizing ability, decreases as the number of training examples that is being fed in. This is opposite of intuition as it should improve as there is more data to train on.



Adaptive Boosting with Decision Trees (Boost) for the Sloan Digital Sky dataset

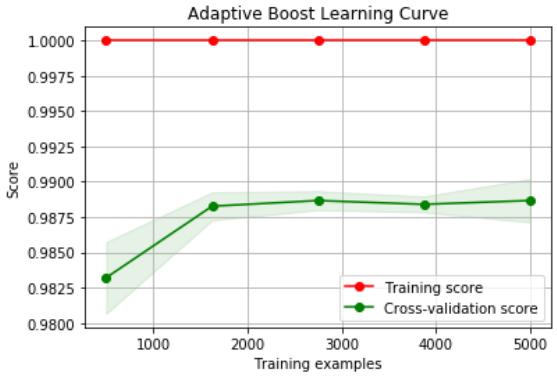
The first thing that I tested was how the accuracy of the boosted tree was affected by the depth of the tree. As it is clearly shown, after about a tree depth of 3, the accuracy ‘flat-lined’ so the I chose a classifier depth of 3 since the simpler the tree, the better chance of generalizability.



Below is a classification report. It lookes identical to the Decision tree classification report however the recall of the QSO label is 0.94 instead of 0.93 so it is slightly better.

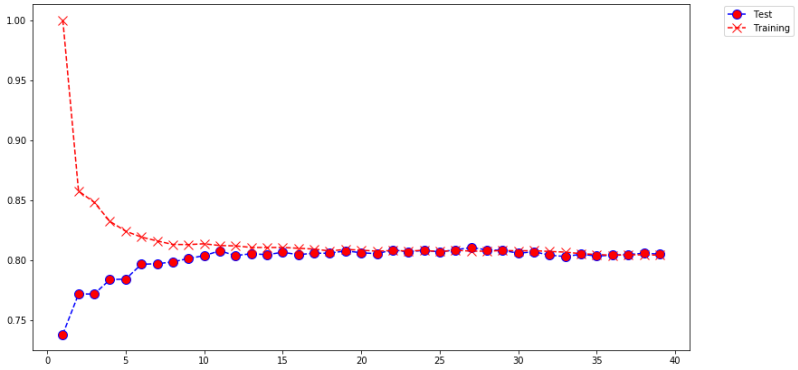
	precision	recall	f1-score	support
GALAXY	0.99	0.99	0.99	1256
QSO	0.97	0.94	0.95	201
STAR	1.00	1.00	1.00	1043
avg / total	0.99	0.99	0.99	2500

Finally, I plotted how the performance of the Boosted classifier changes as the number of training examples increased. As expected, the performance of the Adaptive Boosted Decision Tree increased as the training example size increased. The Adaptive Boosted Decision Tree performed better than the normal Decision Tree as expected.

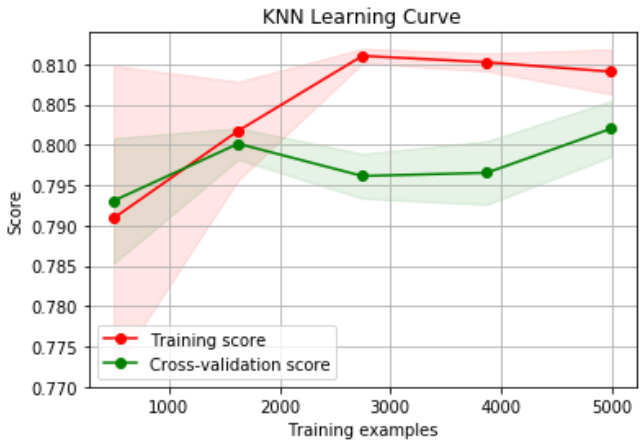


K - Nearest Neighbors (KNN) for the Sloan Digital Sky Survey dataset

The first thing I did was I tried to find the optimal number of neighbors, so I used the default settings and plotted the accuracy as a function of the number of neighbors or the k value. As we can see, as the K value increased, so did the test accuracy (red dots) until the accuracy began to plateau where K was about 12.



Next, we plotted the accuracy vs. training examples and we see that the cross-validation score increases as the number of training examples increases, which means that the algorithm is getting better with more data.



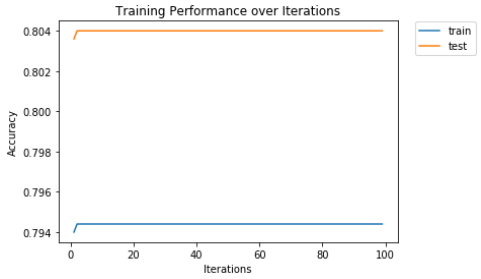
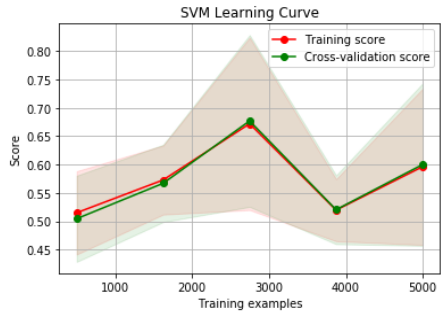
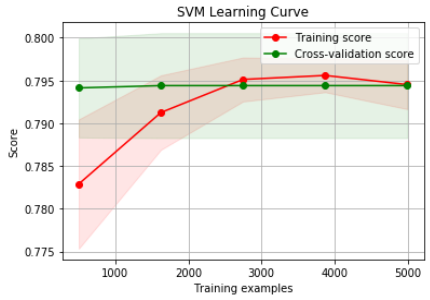
Using the optimal 15 nearest neighbors, I created a classification report for the dataset. The KNN did not do as well as the Adaptively Boosted Decision Tree.

	precision	recall	f1-score	support
GALAXY	0.75	0.96	0.84	1256
QSO	0.32	0.04	0.07	201
STAR	0.91	0.77	0.84	1043
avg / total	0.79	0.81	0.78	2500

Support Vector Machines (SVM) for the Sloan Digital Sky Survey dataset

I created a Linear Support Vector Classification and it used a one vs. rest classification method in order to support multiclass labelling since the dataset included 3 possible labels. For the sake of simplicity, i used the default settings available in Scikit-learn and plotted training performance over iterations. What's odd is that as we train the model more and more, the accuracy plummets. We should expect accuracy to increase however it doesn't. I investigated further and found that there was a parameter known as dual which has to do something with the optimization of the linear Support Vector Machine. Scikit-learn by default has dual = true. I have little idea what a Lagrangian Duality is however I read about in this machine learning notes paper in Stanford. [3] Honestly the theory behind much of this was too much to dive into for this

project however I enjoyed hunting for this obscure parameter that dramatically made my fits better.

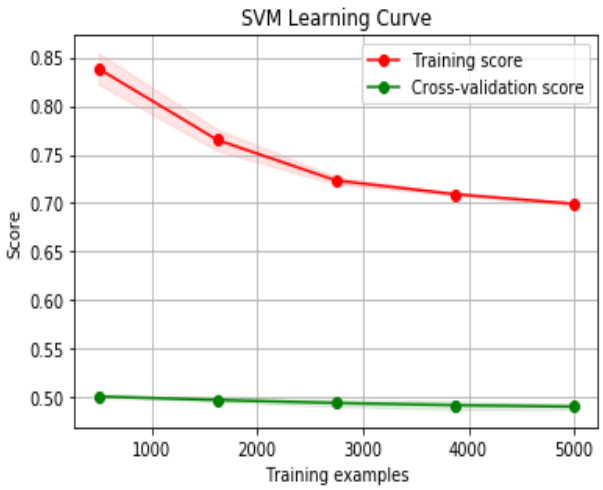
Dual Optimization	Primal Optimization
	
	

As we can see, the primal optimization problem did a lot better than the Dual optimization. If we look at the SVM learning curve we can see that the training score and cross-validation score almost become equal. I learned in class that an SVM is quite resistant to overfitting and the similarities between the cross-validation and training score at the end of training was surprisingly similar unlike all of the other architectures.

After creating the classification report I found that I was improperly calling the linear classifier however I still wanted to talk about what I learned through this failure.

	precision	recall	f1-score	support
GALAXY	0.74	0.98	0.84	1256
QSO	0.00	0.00	0.00	201
STAR	0.93	0.75	0.83	1043
avg / total	0.76	0.80	0.77	2500

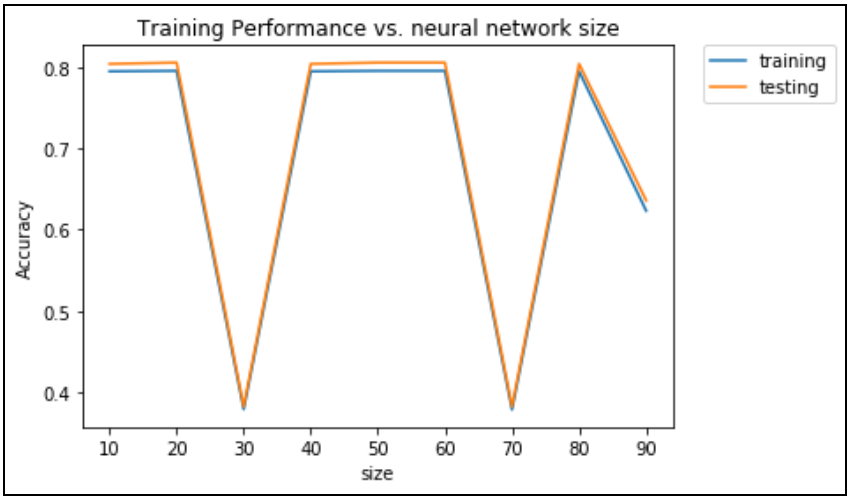
I quickly corrected my error an instead called on SVC in scikit-learn. I obtained these training results as well as a classification report. Both were pretty poor in their performance as all the metrics were very bad.



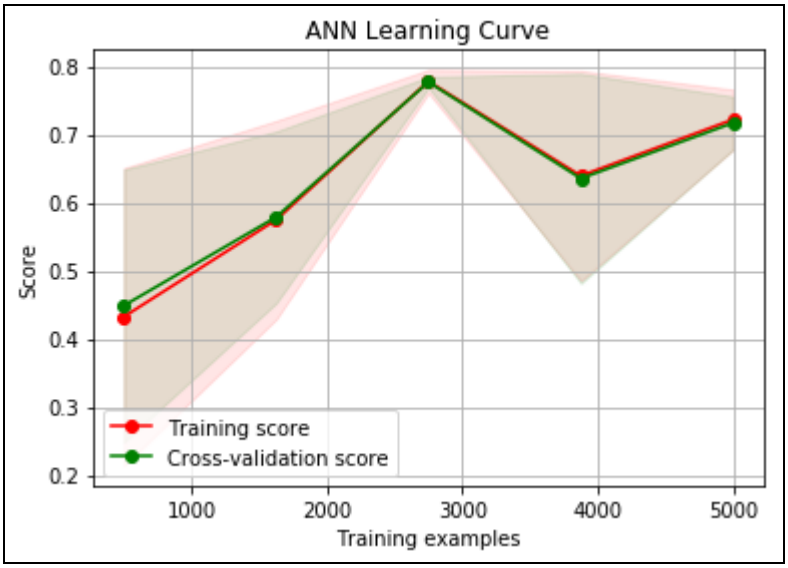
	precision	recall	f1-score	support
GALAXY	0.50	0.94	0.65	1256
QSO	0.57	0.02	0.04	201
STAR	0.32	0.04	0.07	1043
avg / total	0.43	0.49	0.36	2500

Artificial Neural Networks (ANN) for the Sloan Digital Sky Survey dataset

The Artificial Neural Network defied my expectations in the training over multiple iterations, the performance did not improve as I increased my iteration size. I'm not sure If I didn't implement it correctly or If the performance was not very good for such a classification problem.



Afterwards, I plotted the ANN's accuracy as a function of training examples. With 3 hidden layers, each with a size of 100 nodes. As expected the ANN generally increased in its accuracy in the cross-validation set as training increased.



Finally, I looked into the classification report. Scikit learn stated that the Multi-layer perceptron that I used for this problem supports multi-class labeling using the softmax activation function however It failed to properly label the “QSO” or quasar label. I believe this was a major source of error according to the classification report. Softmax is able to “squash” the output and assign probabilities for each label.

	precision	recall	f1-score	support
GALAXY	0.78	0.65	0.71	1256
QSO	0.00	0.00	0.00	201
STAR	0.63	0.89	0.74	1043
avg / total	0.66	0.69	0.66	2500

Summary of Sloan Digital Sky Survey Dataset

Overall, the Adaptive Boosted Decision Tree did the best within the dataset. Overall, the ANN, KNN, and SVM failed to perform as adequately as the decision trees. I think in part, SVM may n

Summary of Project:
Overall, I really enjoyed this project and learned a lot about the various statistical measures for machine learning as well as some important algorithms. Implementing many of these algorithms were quite simple with the libraries available such as scikit-learn. I learned a lot through a lot of my failed attempts and learned a bit of theory as well.

references:

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[2] https://www.kaggle.com/lucidlenn/sloan-digital-sky-survey
[3] http://cs229.stanford.edu/notes/cs229-notes3.pdf