CS7641 Assignment#1

Supervised Machine Learning Analysis

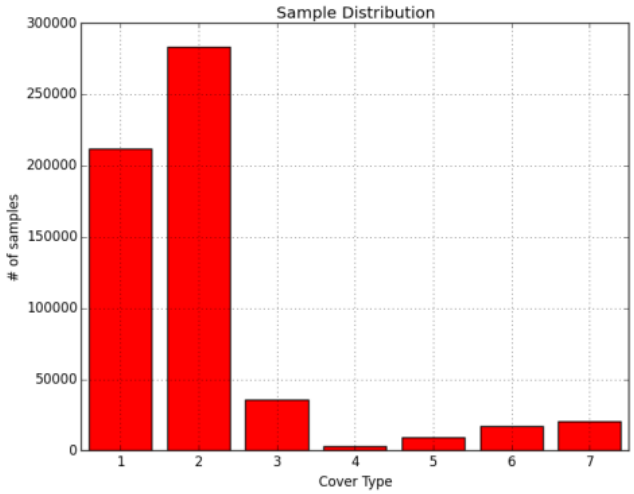
Yancheng Liu (yliu723@gatech.edu)

Datasets Chosen in This Study

The first dataset I chose to look at is the Breast Cancer Wisconsin Dataset. This dataset can be downloaded from UCI Machine Learning Repository. The data are real cancer patient data obtained from the University of Wisconsin Hospitals in 1990s. There are 699 samples in this dataset and 11 attributes, including cell size, cell shape, clump thickness etc, all represented by digits from 1-10 (except the ID numbers which will be excluded from fitting). The task is to predict whether a tumor from a certain patient is benign or malignant based on all the attributes.

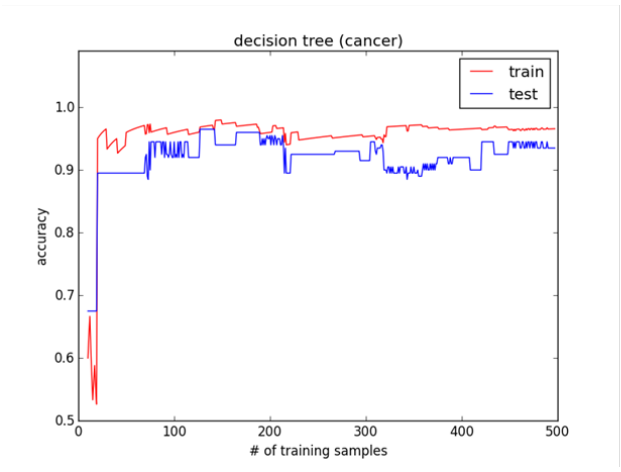
Breast cancer is the most common cancer among women and the second leading cause of cancer death. In 2014, an estimated 232,670 new cases of invasive breast cancer were expected to be diagnosed in women in the U.S. and about 1 in 8 women will develop invasive breast cancer over the course of her lifetime (CDC).Data for required by these attributes can be obtained from biopsy analysis. The accurate and efficient diagnosis of cancer progression is extremely critical for the following treatment choices.

The second dataset I examined is the Covertype Dataset. This dataset can also be downloaded from UCI Machine Learning Repository. The dataset include data about forest cover type in the Roosevelt National Forest of northern Colorado obtained from US Geological Survey. Forests from the area are with minimal human-caused disturbance. There are more than 500,000 samples in this dataset!!! The huge amount of data makes the fitting task very challenging. Thus I performed a random permutation, and used a subset of the data for my analysis. The dataset also contains 54 attributes, including elevation, azimuth, soil type etc. The task is to predict the cover forest type given information from all the attributes.

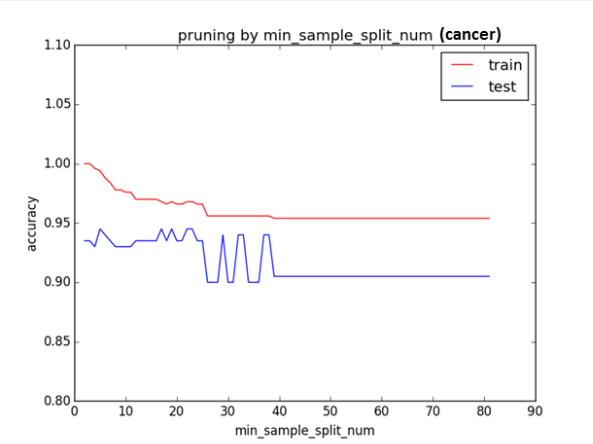


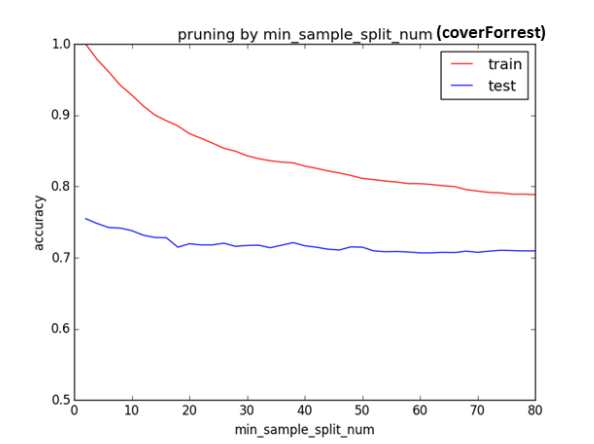
The forest cover dataset has important practical implication for ecological studies. Since cover type are with minimal human-caused disturbance, knowledge about how ecological factors can contribute to forest cover type can be learned by analyzing this dataset. The forest cover dataset is also very interesting in the aspects of machine learning. There are 7 cover types/classes in the dataset. However, not all the classes are well represented – type 1 and 2 are the most abundant and the rest types are not frequently trained. The training performance can be largely increased if more intensive training in type 3-7 were specifically designed. Moreover, the ranges of different attributes are quite different in this dataset (thousands versus 1), so normalization and feature scaling also helps with accuracy and running time.

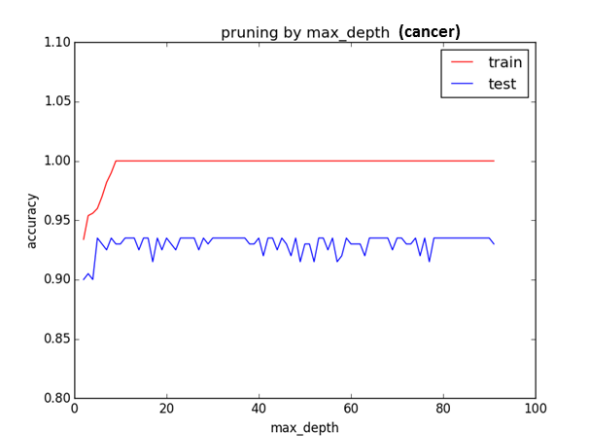
I. Decision Trees with Pruning

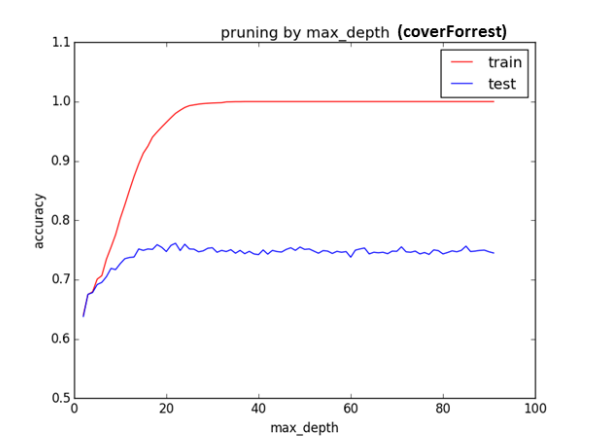


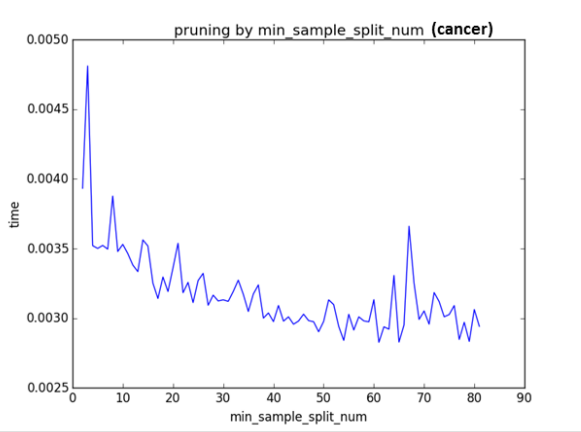
Compared to other machine learning algorithms which usually require data preprocessing such as normalization, an advantage of decision tree is that it needs minimal data preparation. As shown in the figure above, for both datasets, the increase of predication accuracy (and thus the decrease of cost/error) is logarithmic as the number of samples used to train the tree increases. Therefore the tree reached optimal accuracy with relatively low number of samples.

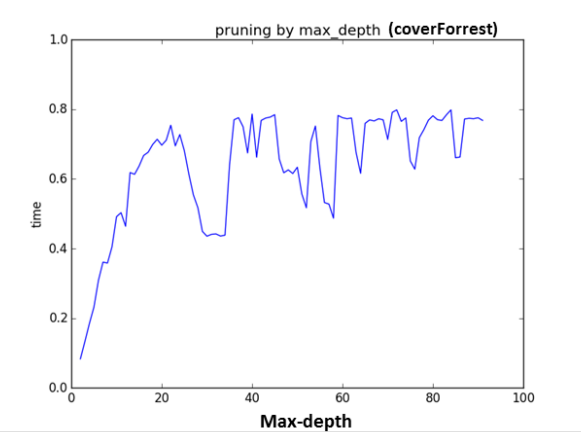






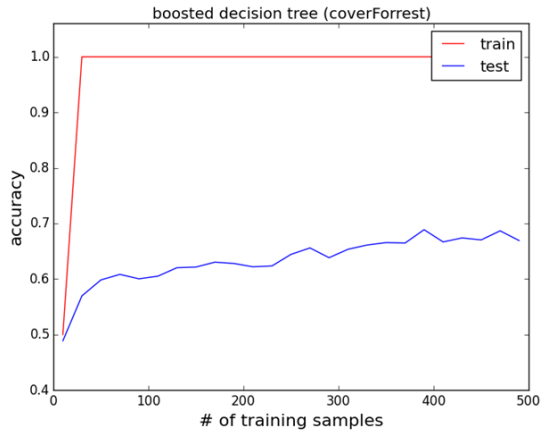




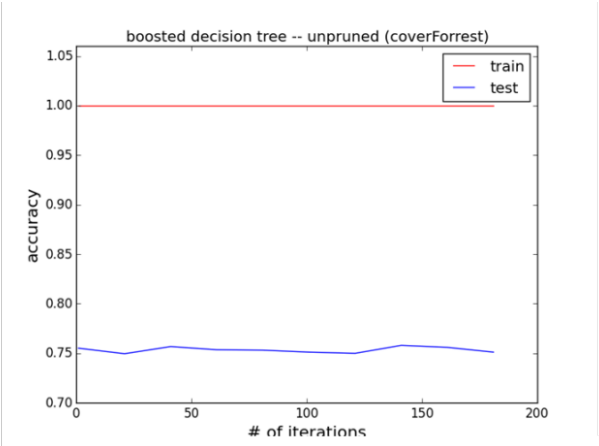


An important question when building a decision tree is that how complex the tree needs to be. Two related parameters ('min-samples-split' and 'max-depth') are pruned to test how accuracy is affected by the structure of the tree learner. The results are shown in the figure above. Over-simplified tree structure tends to not fit even the training data well. Prediction accuracy improved as min-samples-split decreased and max-depth increased. But creating an over-complex tree may cause the issue of overfitting. This issue was evident when building the tree for the cancer dataset: the learner obtained better accuracy when min-samples-split was set at 5 or 20 when compared to setting the parameter at smaller values such as 2 or 4. Finally, as an eager learner, the training takes much more time than testing, and the time increases when min-samples-split and max-depth are pruned towards building a more complex tree.

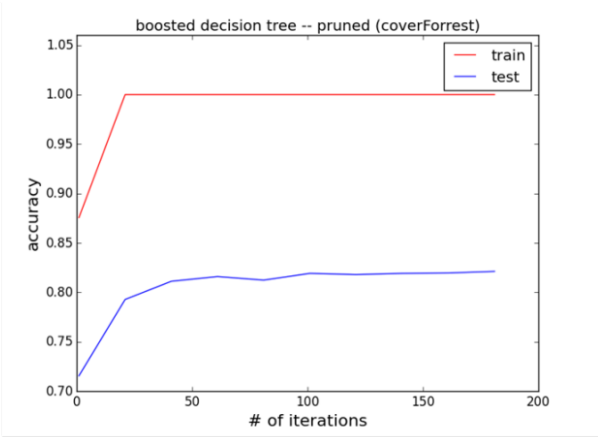
II. Boosting

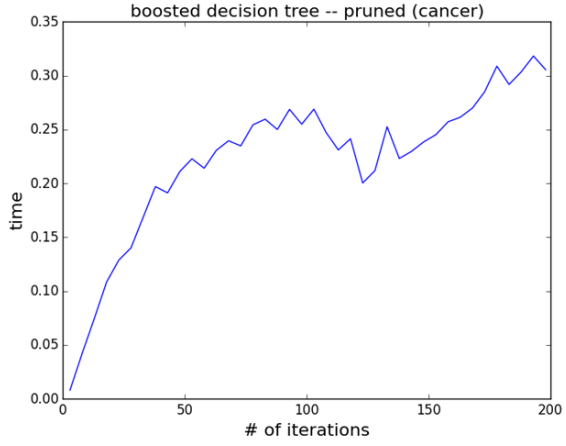


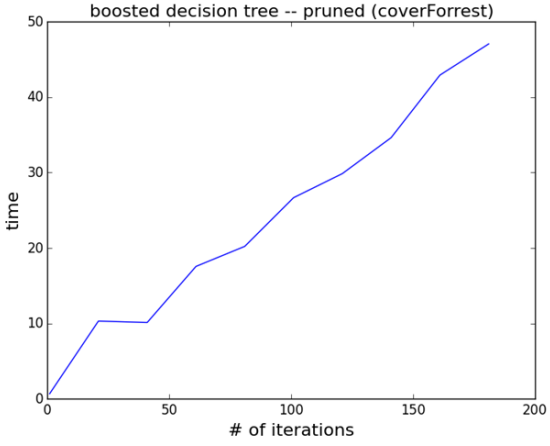

boosted decision tree -- unpruned (cancer)
train
test
accuracy
# of iterations

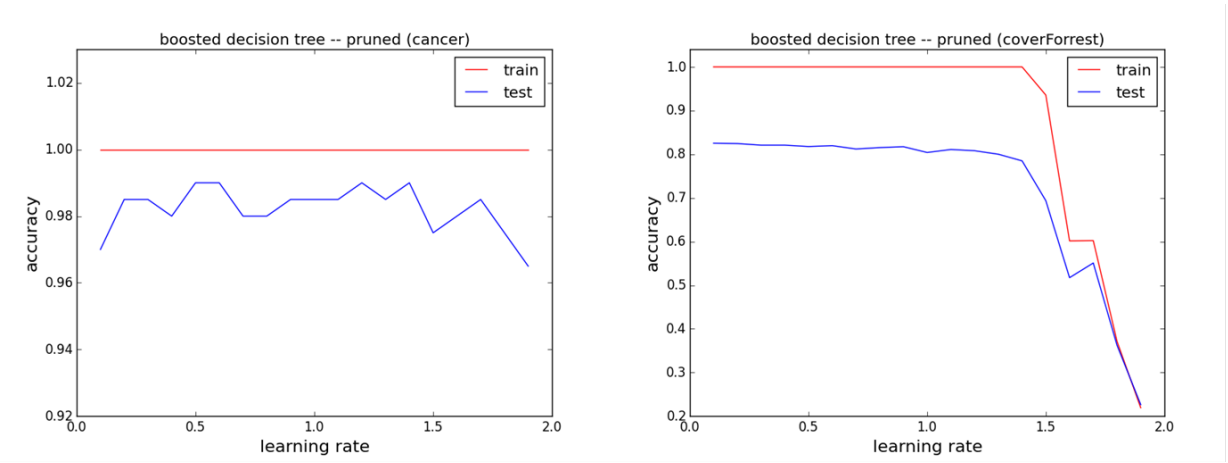

boosted decision tree -- unpruned (coverForrest)
train
test
accuracy
# of iterations

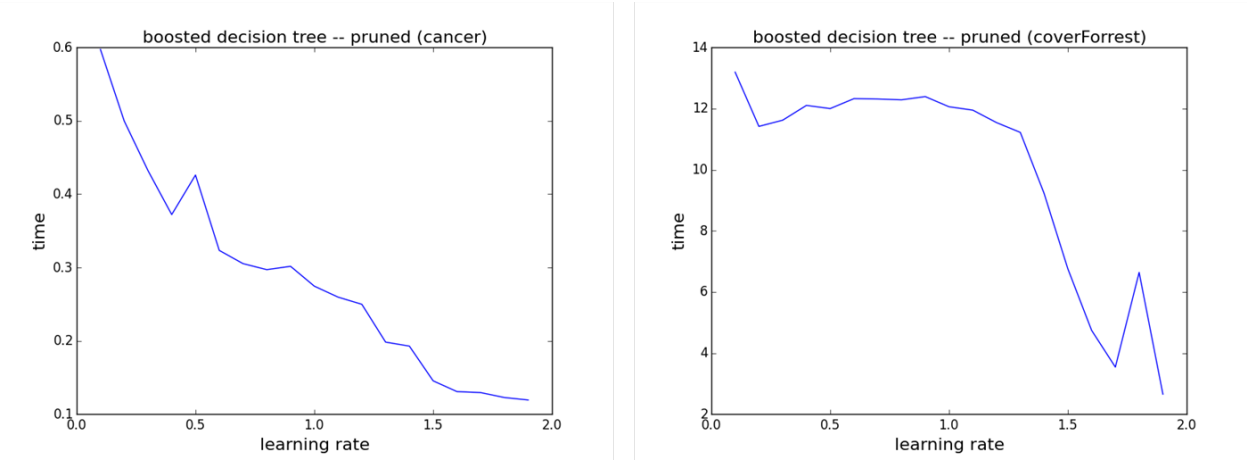

boosted decision tree -- pruned (cancer)
train
test
accuracy
# of iterations


boosted decision tree -- pruned (coverForrest)
train
test
accuracy
# of iterations


boosted decision tree -- pruned (cancer)
time
# of iterations


boosted decision tree -- pruned (coverForrest)
time
# of iterations

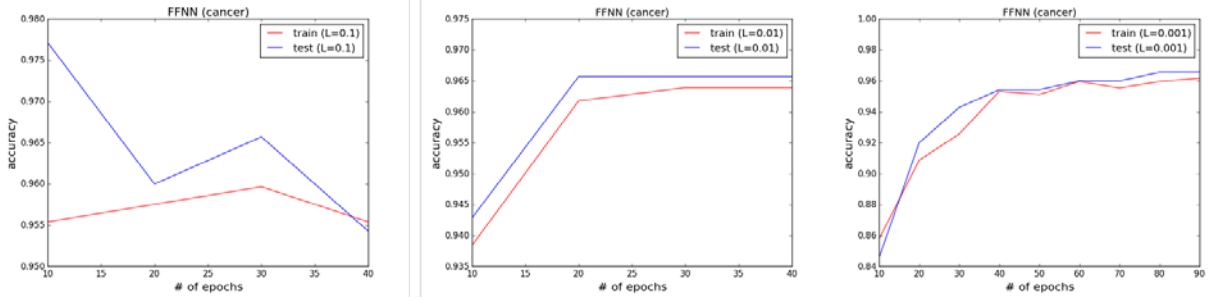




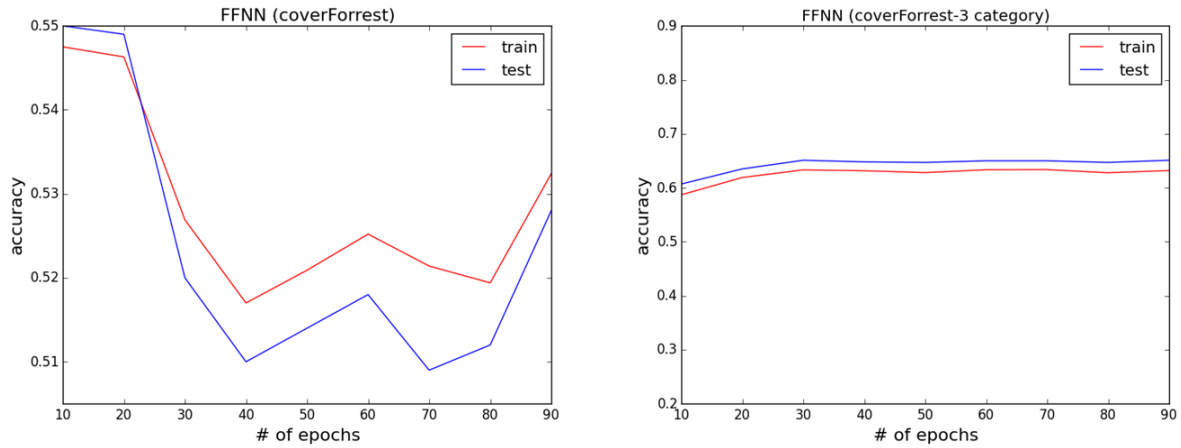
The AdaBoost fits a sequence of weaker learners and improves through iterations. It put increasing weights on those examples that are difficult to predict when the learner iterates. For this analysis, the boosted ensemble is built on the same decision tree used earlier. Just the tree learner, it runs fast. For smaller dataset like the cancer dataset, it reached optimal accuracy within 20 iterations, while for larger dataset like the coverForrest dataset, it took a lot more iterations to improve its prediction accuracy. As shown in the figure above, the time to train the learner is almost linearly proportional to the number of iterations used.

Change the base_estimator from an unpruned decision tree to a pruned one significantly improved the accuracy of the learner. The learner can be tuned by decreasing the learning_rate, which will reduce the weight of each classifier. With smaller learning_rate, it takes more time to train the learner, but it has a major effect on improving the prediction accuracy especially for the larger dataset coverForrest.

III. Neural Networks



A feed forward neural network (FFNN) with a backpropagation trainer was used for the analysis of the two datasets. This is a learner which showed very different performance when applied to the datasets. With the cancer dataset, it consistently produced good accuracy (>95%) with various learning rates. It is interesting to note that the accuracy usually increases as the number of epochs increases when learning rate is small (L=0.01 or 0.001), however, when learning rate is set to a larger value (L=0.1), the accuracy no longer correlates with the number of epochs. Learning rate determines how fast the link weights and node biases can be modified. The possible reason for the above observation is that if learning rate is large, the FFNN may learn more quickly, but if there is large variability in the training data, the learner has a better chance of being stuck around a local optimum and not being able to move towards the global optimum. Therefore, in practice it is better to set the learning rate at a small value and tweak it upward if the learner trains too slow.



Compared to the cancer dataset, the predication accuracy for the coverForrest dataset is low (approx 0.52). There are two possible explanations. The fact that there are 54 features and 7 classes in the coverForrest dataset makes the classification much harder than the cancer dataset which has a binary output. Another possible reason for this low accuracy is the screwed distribution of samples across the 7 classes in the coverForrest dataset. As mentioned earlier, type 1 and 2 are way over-represented, and type 3-7 are under-represented. In support of this possibility, when inspected the prediction results in detail, I found that the accuracy for predicting type 1 and 2 are much better compared to the accuracy for predicting type 3-7. To tackle this issue, I grouped type 3-7 together as a single class to even the distribution of samples across different classes. This tweak did improve the accuracy significantly (approx. 0.65).

IV. Supported Vector Machines

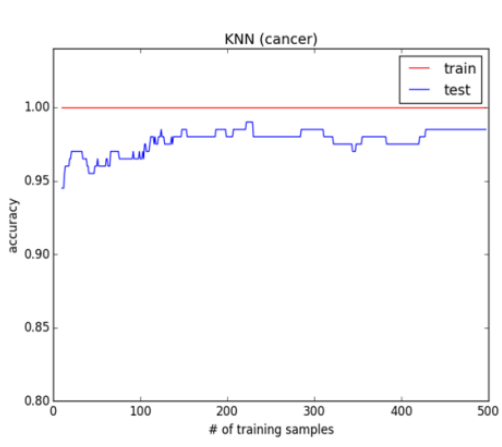
Dataset	Kernel	gamma	train_accuracy	test_accuracy	train_time	test_time
cancer	Linear	0	0.9699	0.98	0.5031	0.2494
cancer	Ploy (degree=2)	0	0.965	0.98	0.4671	0.1871
cancer	Ploy (degree=2)	0.01	0.994	0.94	1.8879	0.1284
cancer	Ploy (degree=2)	0.1	0.978	0.965	0.5151	0.1324
cancer	Ploy (degree=2)	0.5	0.9779	0.965	0.468	0.4991
cancer	Ploy (degree=2)	1	0.9699	0.9749	0.5605	0.1871
cancer	Ploy(degree=3)	0	1	0.935	0.406	0.187
cancer	Ploy(degree=3)	0.01	0.9739	0.96	0.5616	0.1872
cancer	Ploy(degree=3)	0.1	0.998	0.935	0.5149	0.1873
cancer	Ploy(degree=3)	0.5	1	0.94	0.593	0.1246
cancer	Ploy(degree=3)	1	1	0.94	0.4679	0.1247
cancer	rbf	0	0.996	0.95	0.6395	0.1248
cancer	rbf	0.01	0.9659	0.975	0.8733	0.1246
cancer	rbf	0.1	0.996	0.95	0.5138	0.1248
cancer	rbf	0.5	1	0.93	0.9508	0.1872
cancer	rbf	1	1	0.875	0.4639	0.1248
coverForrest	Linear	0	0.7	0.7355	1006.82	0.0281
coverForrest	Ploy(degree=2)	0	0.7565	0.65	7898.08	0.027

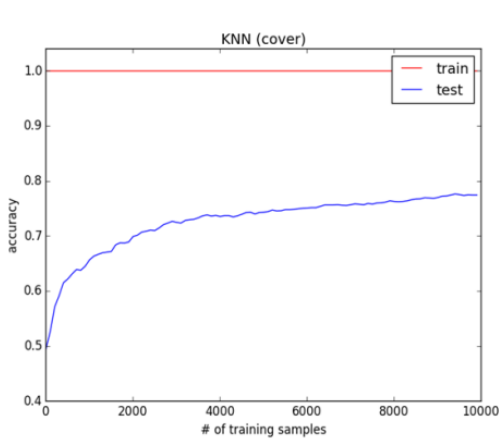
coverForrest	rbf	0	1	0.493	3.1375	0.1508
coverForrest	rbf	0.1	1	0.494	3.3411	0.1681
coverForrest	rbf	1	1	0.495	2.9685	0.145
coverForrest	sigmoid	0	0.4724	0.494	1.17	0.4202

An advantage of SVM learner is its flexibility of selecting different kernel functions. I tested three kernels for analyzing the cancer dataset: a linear kernel, a polynomial kernel with varying degrees and an rbf (Radial Basis Function) with varying gamma. Most of them worked very well in terms of prediction accuracy. Gamma is the parameter of a Gaussian Kernel. A small gamma can give low bias and high variance while a large gamma will lead to higher bias and low variance. This is consistent with the observation that the accuracy decreased from 0.975 to 0.875 when changing gamma from 0.01 to 1 with the rbf kernel. One disadvantage of SVM is that the time to train the learner can increase exponentially when there are a lot of features. Like in this analysis, it took less than 1 second to train the cancer data (10 features), but it took hours (1006.82 seconds with linear kernel and 7898.08 seconds with 2-degree poly kernel) when training the coverForrest data which has 54 features.

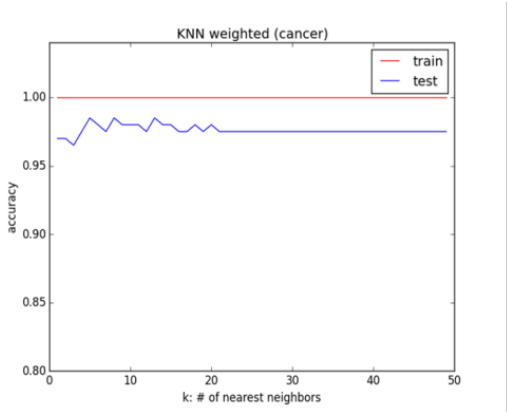
V. k-Nearest Neighbors

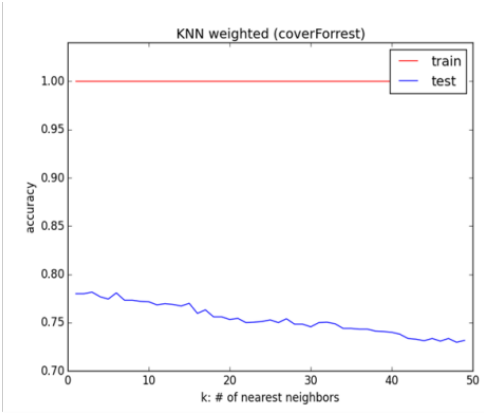
Generally, more samples were trained by k-NN algorithm, the more accuracy the prediction will be. After training with 150 samples in cancer dataset, the accuracy of the prediction reaches as high as 98% and remains that level when training samples increases. However, the accuracy of coverForrest prediction keeps increasing as the training samples increases. If more samples can be trained without the memory space limitation, the prediction accuracy should be better than 80%.

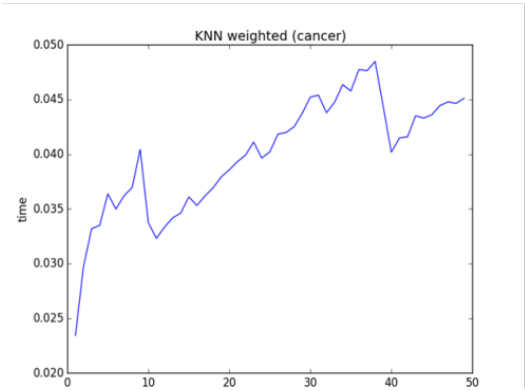


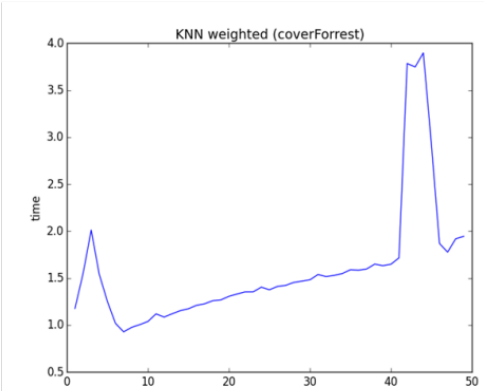


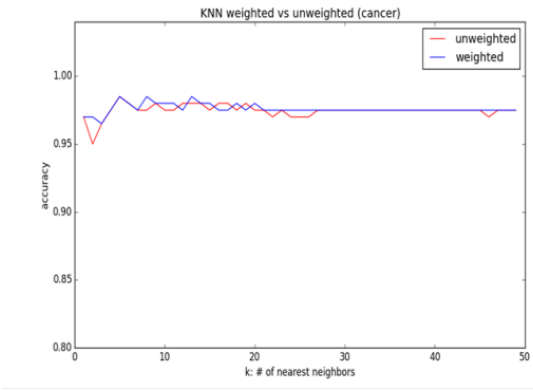
For the cancer dataset, k=5 gives the best prediction performance, but for the coverForrest dataset, better performance was observed with small k (k=1). Large k values are associated with longer training time. Weighted algorithms gives much better prediction in coverForrest dataset than the unweighted k-NN, but not much differences can be observed with the cancer dataset. This is because the class distribution in the coverForrest dataset is highly skewed. Weighting the contribution of each neighbor by their distance helps to overcome this problem.

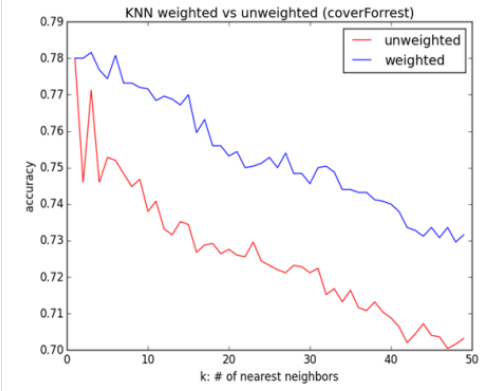

KNN weighted (cancer)
train
test
accuracy
k: # of nearest neighbors


KNN weighted (coverForrest)
train
test
accuracy
k: # of nearest neighbors


KNN weighted (cancer)
time


KNN weighted (coverForrest)
time


KNN weighted vs unweighted (cancer)
unweighted
weighted
accuracy
k: # of nearest neighbors


KNN weighted vs unweighted (coverForrest)
unweighted
weighted
accuracy
k: # of nearest neighbors