

CS7641 Fall 2018 Assignment1 Supervised learning

Xiaoxi Wang (xwang738)

email: xwang738@gatech.edu

Summary

Datasets

Why they are interesting

Nontrivial, allows tuning and comparing of different algorithms

	Task	#Instances	#Attributes						
HTRU2	Binary Classification								
Breast Cancer Wisconsin	Binary Classification	683 ^[1]	9 ^[2]	Real					

Table 1.

[2] Removed 16 instances with missing data

[3] ID number attribute was dropped

Scaling

Splitting into training (70%) and testing (30%)

Results and Discussion

1. Decision Tree

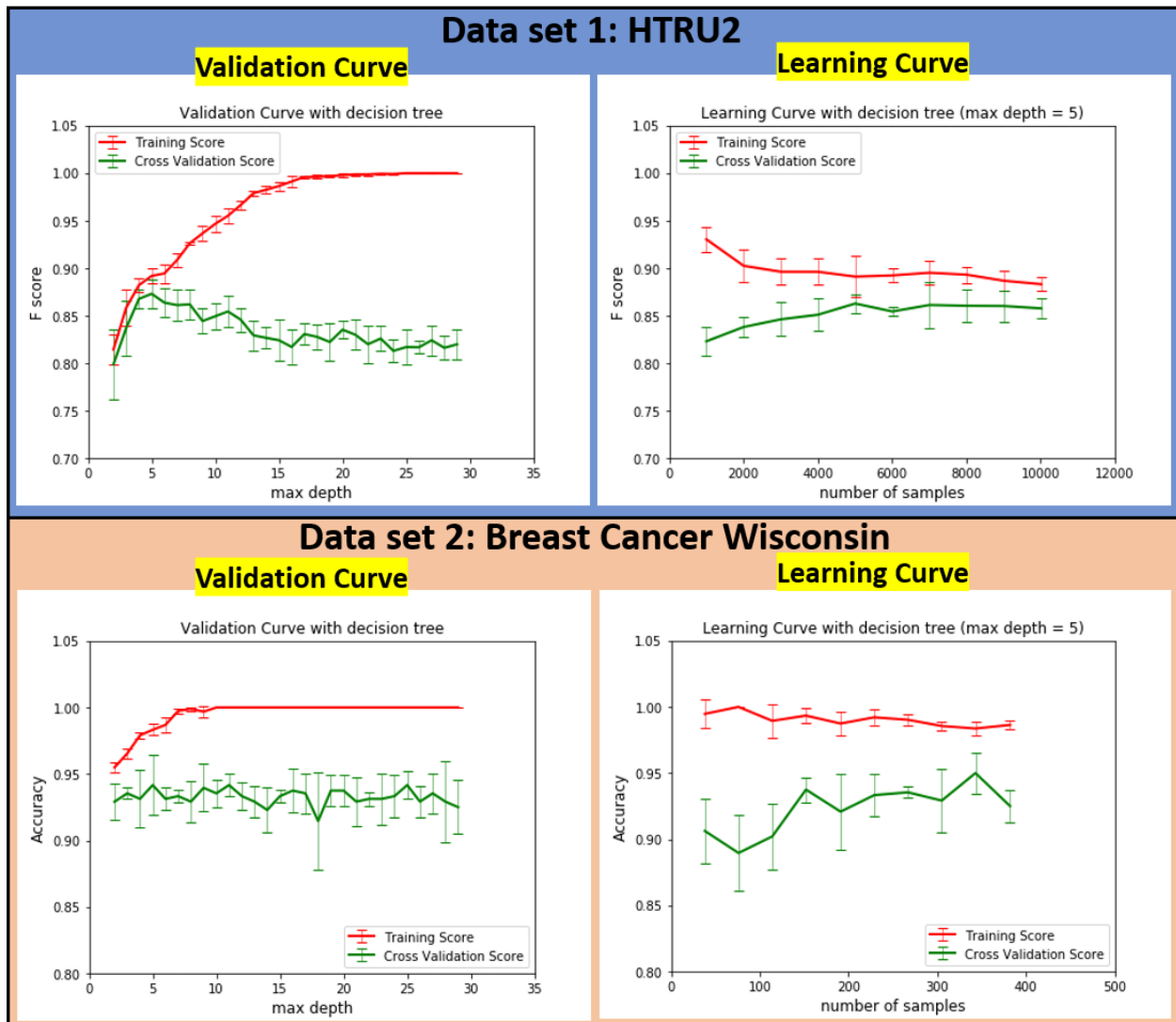


Figure 1. Decision Tree: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

2. k -Nearest Neighbors

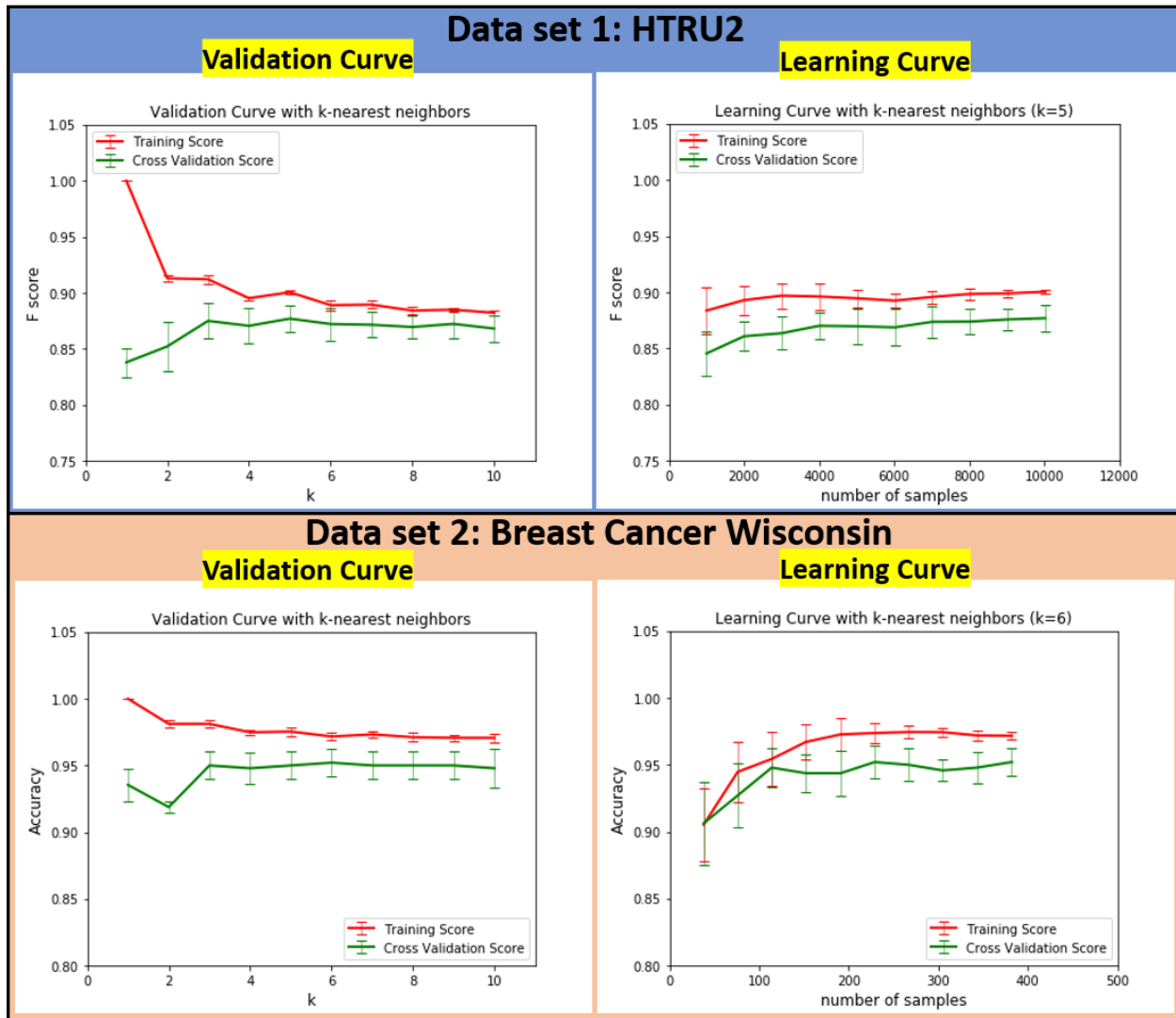


Figure 2. k -Nearest Neighbors: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

The effects of different values of k
The HTRU2 dataset

3. Support Vector Machine

3.1 linear kernel

The HTRU2 dataset

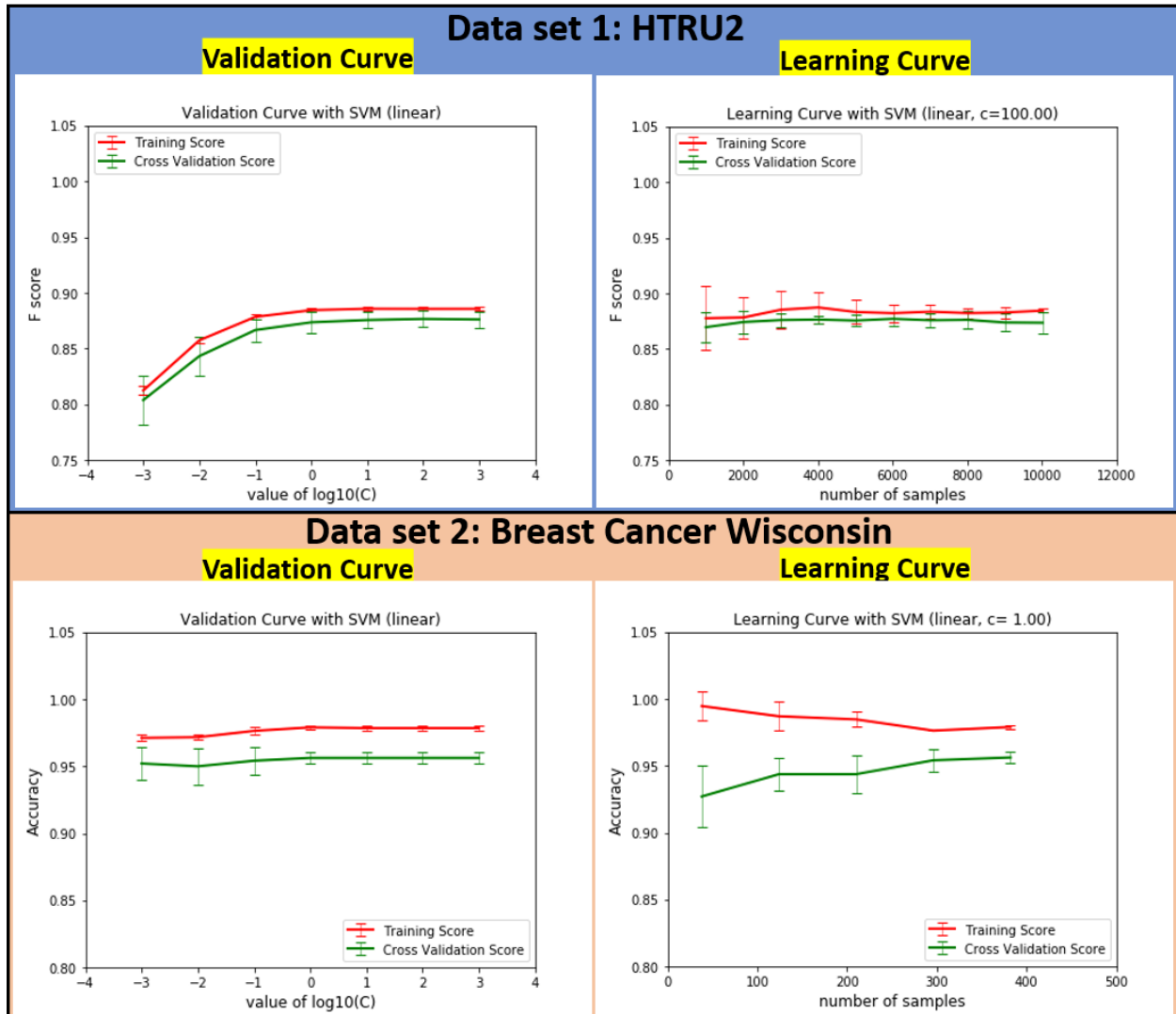


Figure 3.1. SVM with linear kernel: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

3.2 RBF kernel

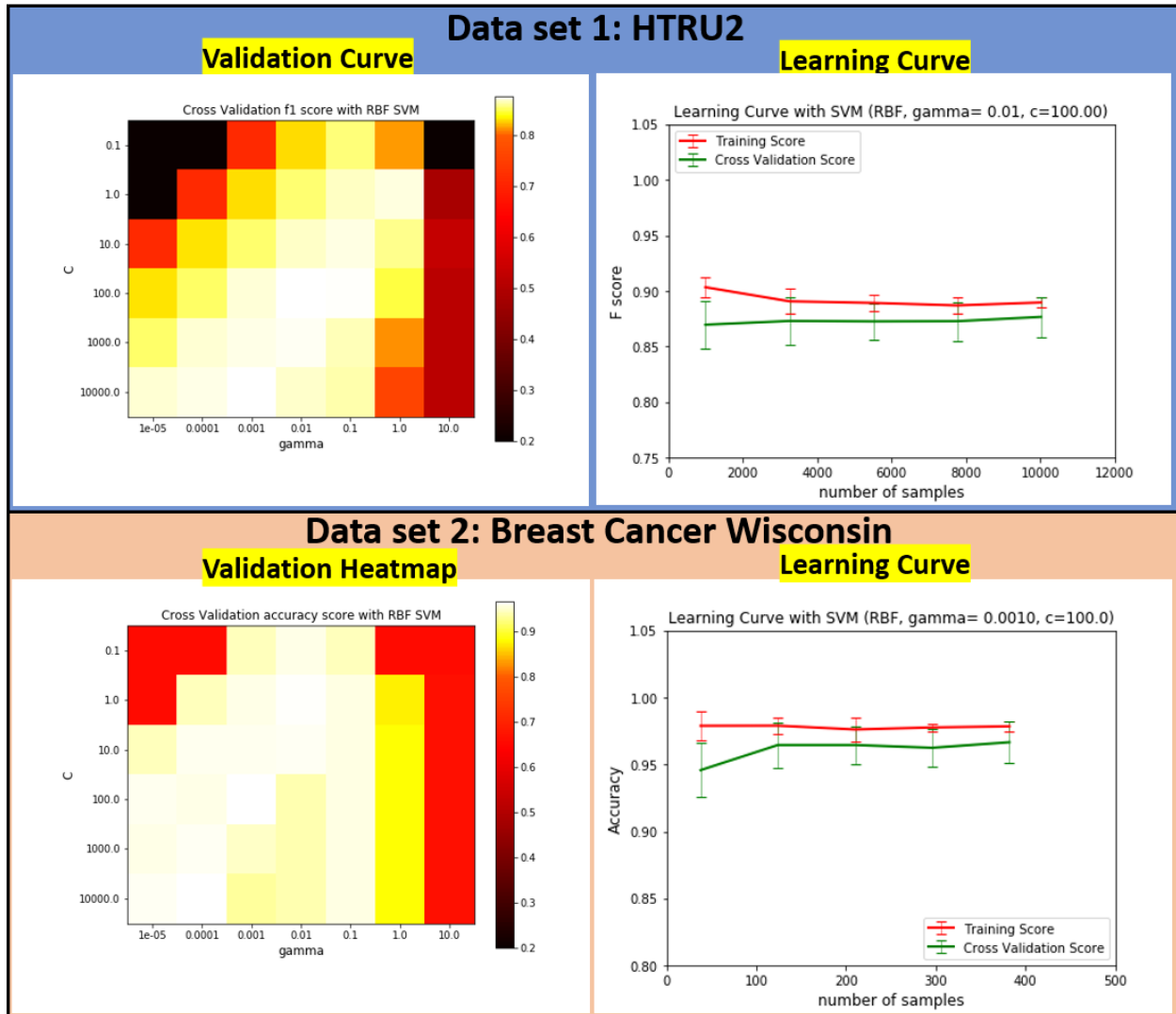


Figure 3.2. SVM with RBF kernel: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

4. Boosting

Base classifier: decision tree with more aggressive pruning (max depth 3 for HTRU2 and 1 for breast cancer data set)

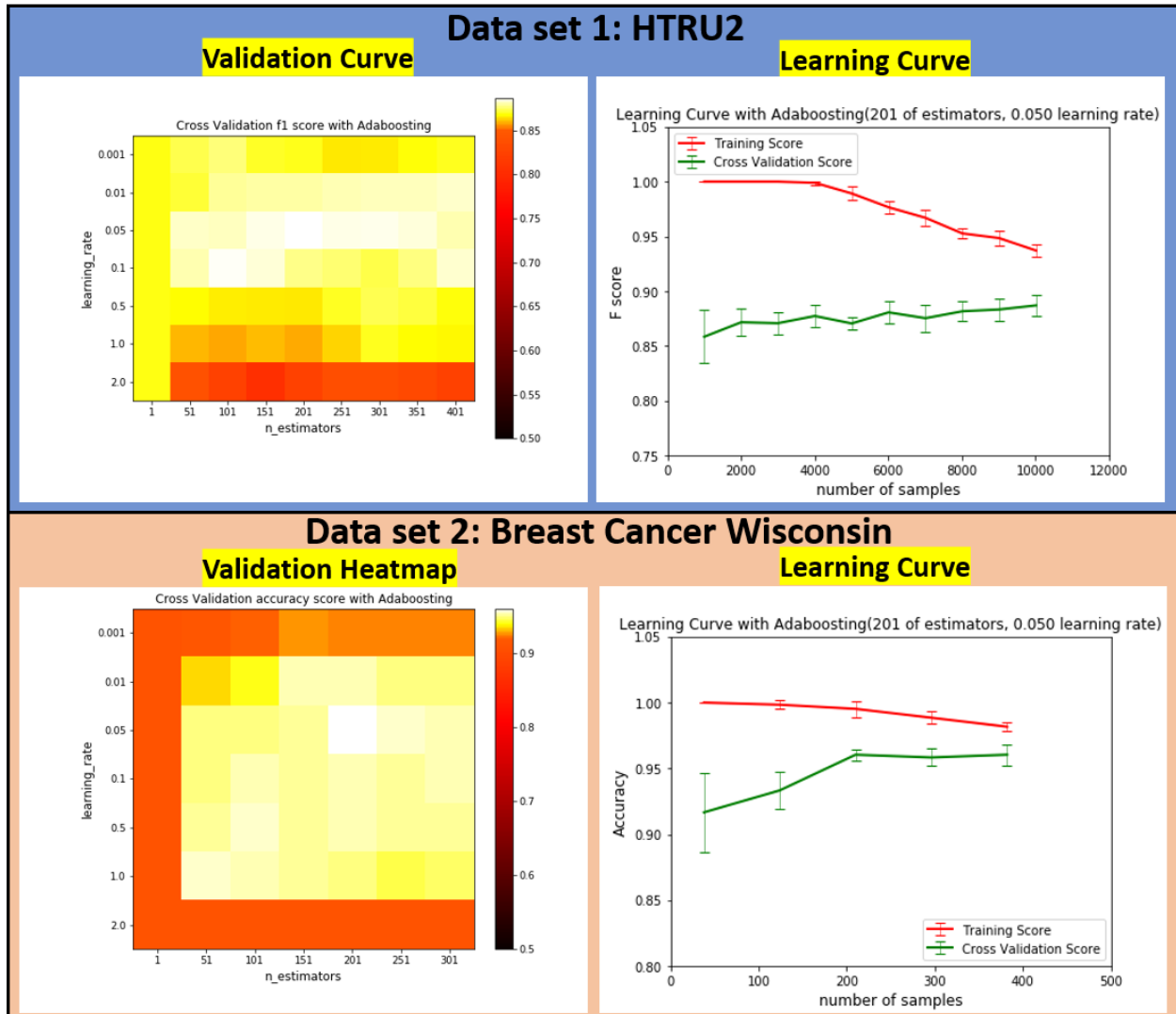


Figure 4. Adaboosting: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

5. Neural Networks

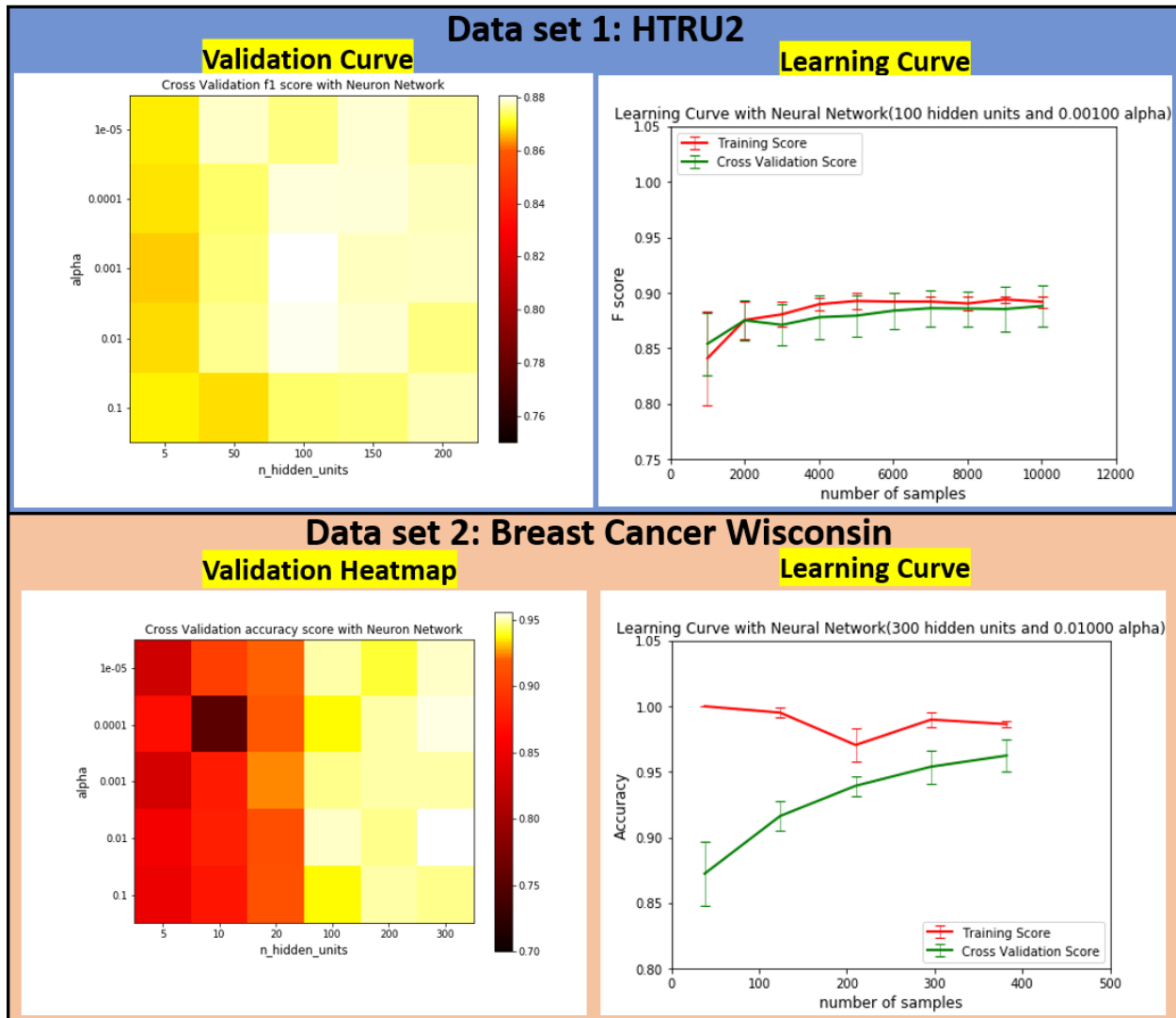


Figure 5. Neural Networks: Validation curve and learning curve
(5 folds cross validation. Error bars represent standard deviation)

1 hidden layer # of hidden units the value of alpha, activation function

6. Compare and contrast the different algorithms.

Learning Algorithms	Test AUC Score	Test Accuracy	Test F Score	Training Time	Test Time	Hyperparameters
HTRU2 Data Set						
Decision Tree	0.89898	0.97821	0.85921	0.0132626	0.00346114	Max depth = 5
k-Nearest Neighbor	0.90994	0.97952	0.86967	0.0091312	0.126501	k=5
SVM (linear kernel)	0.90952	0.98063	0.87560	3.70600	0.0311281	C=100
SVM (RBF kernel)	0.91270	0.98082	0.87753	0.454213	0.0916295	C=100 gamma=0.01
Boosting	0.91762	0.98045	0.87691	6.54262	0.102999	Decision tree with max_depth = 3 200 estimators Learning rate 0.05
Neural Networks	0.91987	0.98082	0.87953	4.57695	0.00571682	alpha=0.001 1 hidden layer with 100 nodes
Breast Cancer Wisconsin Data Set						
Decision Tree	0.94077	0.94634	0.92617	0.0006914	0.00035675	Max depth = 5
k-Nearest Neighbor	0.95128	0.95610	0.93960	0.0007125	0.00180639	k=6
SVM (linear kernel)	0.94744	0.95122	0.93333	0.0027717	0.00044157	C=1
SVM (RBF kernel)	0.96744	0.96585	0.95425	0.0023225	0.00058448	C=100 gamma=0.001
Boosting	0.93795	0.94634	0.92517	0.309154	0.0149498	Decision tree with max_depth = 1 200 estimators Learning rate 0.05
Neural Networks	0.93795	0.94634	0.92517	1.75296	0.00088109	alpha=0.01 1 hidden layer with 300 nodes

Table 2. Comparison of different learning algorithms

Which algorithm performed best? How do you define best?

What sort of changes might you make to each of those algorithms to improve performance?

How fast were they in terms of wall clock time? Iterations?

How much performance was due to the problems you chose? How about the values you chose for learning rates, stopping criteria, pruning methods, and so forth (and why doesn't your analysis show results for the different values you chose?)?

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.

Reference

W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

[Web Link]

O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.

[Web Link]