CS 7641 Supervised Learning

Jun Wang
jwang3316@gatech.edu

Abstract—In this assignment, I implemented 5 different learning algorithms: Decision Trees, Neural Networks, Boosting, SVM, k-Nearest Neighbors, on two interesting classification problems. Then, I discussed their the comparison of these different algorithms' performance and how to improve them.

1 DATASET

1.1 Selection

Glass Identification Data Set[1] - The purpose of this data set is to distinguish the type (headlamp, tableware, vehicle windows, etc.) and chemical composition (Silicon, Na, etc.) of the glass, which belongs to the multi classification problem.

Breast Cancer Data Set[2] - This dataset is also from UCI machine learning repository. The purpose of this data set is to judge whether a patient has breast cancer by learning the characteristics of the sample.

1.2 Why interesting

For the first data set, the Glass dataset, its interesting thing is that it is a multi classification problem, and the distribution of glass data is uneven, which may lead to insufficient machine learning training. But it also makes the problem more realistic and more likely to solve practical problems, such as industrial production, environmental protection, criminal detection, etc.

For the second data set, the Breast Cancer dataset, Its interesting thing is that, unlike the first dataset, it is a binary classification problem, but at the same time, its feature training involves 32 different dimensions. This enables the model to

[1] Glass dataset from UCI machine learning repository.

[2] Breast Cancer dataset from UCI machine learning repository.

help doctors judge the condition, improve efficiency and reduce the probability of misdiagnosis in the actual medical diagnosis process.

1.3 Data Processing

Because both datasets are very classic datasets (frequently used) and have been processed in advance, we do not need to do additional preprocessing. I divide the data set into 80% training set and 20% test set. As the experiment progresses, the training set may be further divided into test sets, which will reach 3:1 in the later stage. The process is shown below as Figure 1.

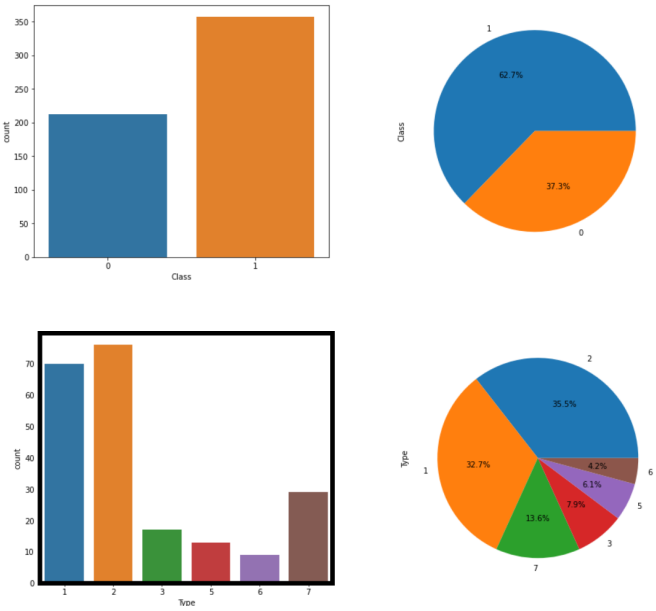

Figure 1—Data processing for two dataset. Upper two figures are Breast Cancer's classification, while the bottom two are Glass's

2 ANALYSIS

In this section, I discussed the performance of the 5 learning algorithms on the two problems defined above and how to improve them.

2.1 Decision Trees

Split Attributes - Information gain is used.

Pruning - During the process of tunning hyperparameters, experiments on control variables are conducted. Each parameter is tested separately with keep others the same. This is to find the optimal value of each super parameter. Like Figure 2 shows, I tunned 5 hyperparameters for both two problems: Max depth, Min samples leaf, criterion (information gain), splitter, min_samples_split.

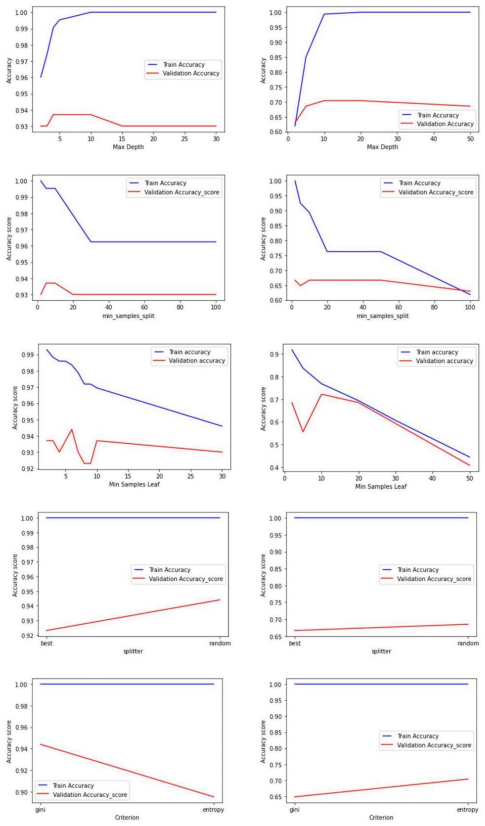

Figure 2—left row is Breast Cancer, right row is Glass. X-axis is different hyperparameter, Y-axis is the accuracy of DT model. Red line is validation accuracy, blue line is train accuracy.

It can be seen from the figure that the maximum depth has a great impact on the accuracy score. With the increase of the depth of the Decision Tree, the accuracy increases first and then decreases. The first increase is because the relationship between attributes and object value is more and more accurate by the splitter. The later decrease is because, with the increase of training, the hierarchy of the Decision Tree becomes more and more deep, making the model more and more

complex. This leads to overfitting. Therefore, I perform pruning operation on the decision tree to control the hyperparameters of the decision tree, such as max depth and min sample leaf, and reduce overfitting.

After pruning, the performance of Decision Tree on both two problems improve a lot. As Figure 3 shows, the learning curves, the accuracy score keep increasing or keep the level with the increase of training times. In addition, there is underfitting problem in Glass dataset after pruning, which needs further adjustments.

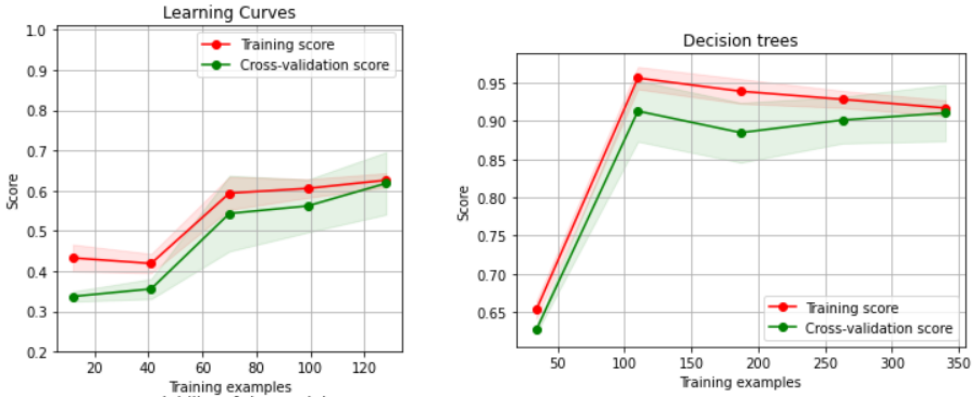

Figure 3—After pruning. The performance and traing size. Left is Glass, right is Breast Cancer.

2.2 Boosting on two problems

Algorithm - AdaBoost

Adjustments - To construct an AdaBoost model, the first step is to determine the number of trees in the forest. Therefore, the hyperparameters that needs optimized is n_estimators and learning rate.

In general, the larger the n_estimators, the better, but the memory and the training and prediction time will also increase correspondingly, and the marginal benefit is decreasing, so we should select the largest n_estimators within the affordable memory and time. As shown in Figure 4, the accuracy curve gradually tends to be flat, which means the hyperparameters reach the optimal area. Find the optimal value of the hyperparameter.

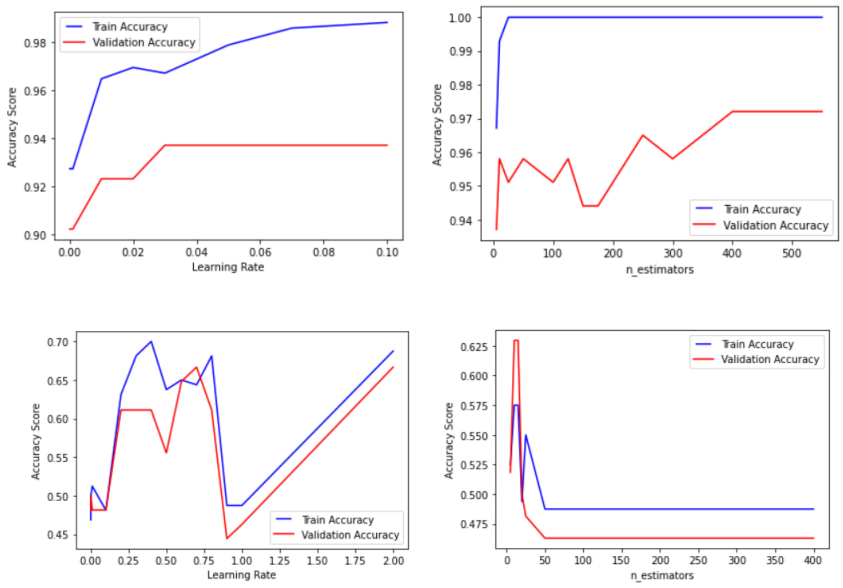

Figure 4—Hyperparameter and accuracy. Left row is learning rate, right row is n_estimators. Top is Breast Cancer, bottom is Glass.

As for training times, after hyperparameter tunning, we can test it. For Breast Cancer dataset, as shown in Figure 5 left row, the learning curves and the scalability of the model, the curve generally flats out without over-fitting as the traing times increase. For Glass dataset, the accuracy decreases with training times increase, due to AdaBoost itself. As I defined above, the Glass dataset is a multi-classification problem, which the AdaBoost model can't work well on. When implement AdaBoost on multi-classification, it will classify one class as one, and classify the others as the second one. So, to improve it, I need to change the algorithm.

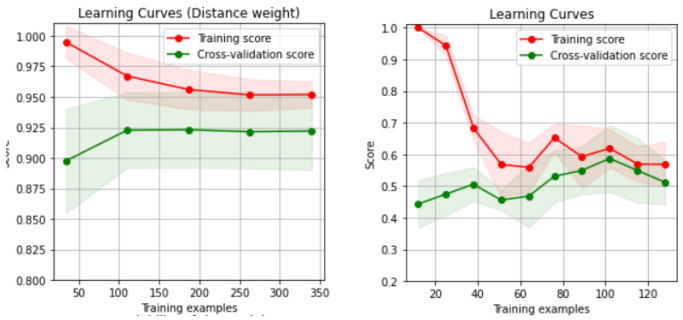

Figure 5.a—performance & training size. Left Cancer, right Glass

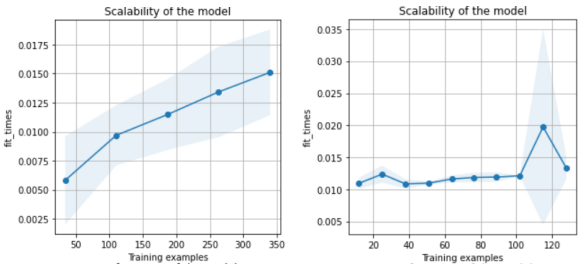

Figure 5.a—scalability & training size. Left Cancer, right Glass

2.3 Neural Network

NN - Multi-Layer Perceptron

Hyperparameter Optimization - As shown in Figure 6, as epoch increases (epoch means iterative times), the accuracy increases first and then decreases. That's because more iterations improve the convergency of the Nerual Network, but if the epoch is too much, there will be an overfitting of the NN. So, we need to adjust the hyperparameters of the NN. First one is learning rate, which impacts learning time. Second one is the number of neurons in the Nerual Network, which contributes to the complexity of the Network.

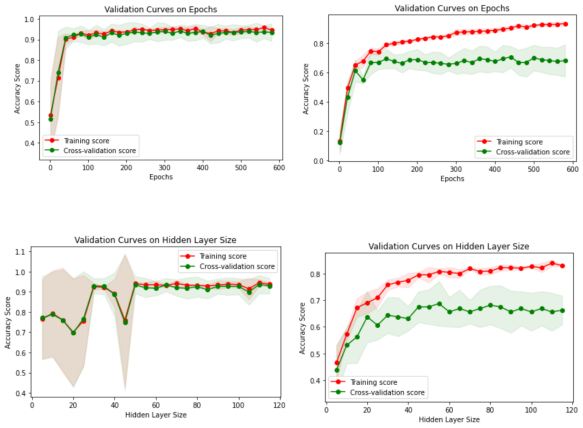

Figure 6—First line is epoch, second is number of neurons. Left is Cancer, right is Glass

Training Size - As Figure 7 shows, after the hyperparameters' optimization, single- layer Nerual Networks' and multi-layer Nerual Networks' complexity increase as training times increase without overfitting.

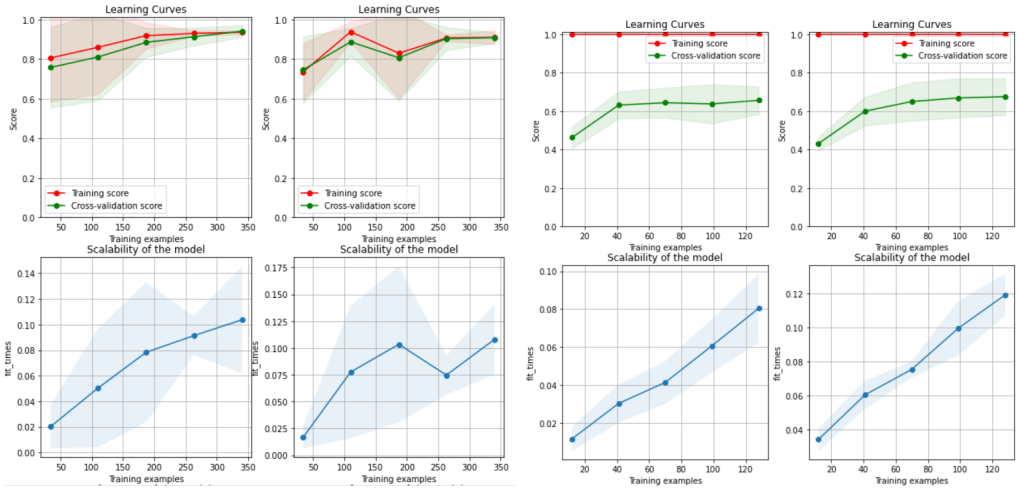

Figure 7—training size & accuracy or scalability. Left 1 is Cancer 1-layer NN, left 2 is Cancer 3-layer NN,right 2 is Glass 1-layer NN, right 1 is 3-layer NN.

2.4 SVM

Parameter Optimization - C and the kernel function's relevant parameter (gamma, degree, etc.). With the increase of the C value, the decision boundary shrinks, and the model will fit the training data more closely. However, the increase of the C value does not increase linearly with the effect of the model. As shown in Figure 8, C has impact on the correct classification and maximum marginal decision.

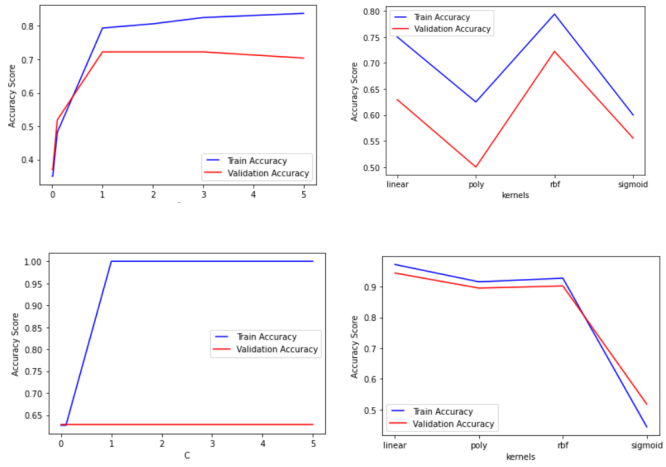

Figure 8—C and kernel function. Left is C, right is kernel. Top is Cancer, right is Glass

As we can see from the figures, the increase of C possibly result in overfitting, and C could reach the optimal when the curve keep the level. And from the right, different kernels have different impact on the accuracy.

Training - From Figure 9, it indicates that SVM works better in the problem with larger number of feature attributes (better in Breast Cancer problem). To improve performance, SVM can be combined with Boosting, acting as a weak learner.

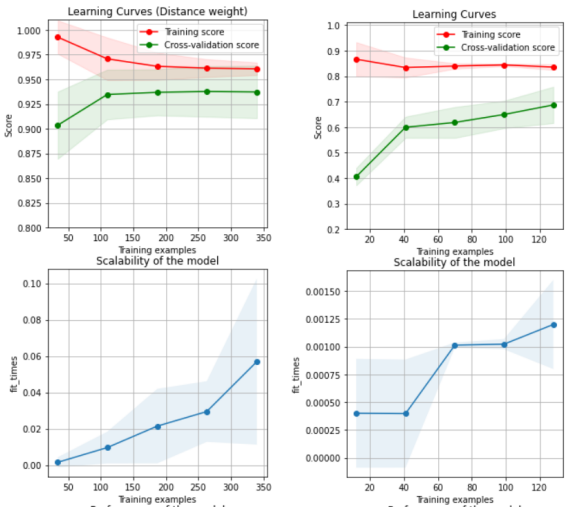

Figure 9—Training size & learning or scalability. Left is Cancer, right is Glass

2.5 k-Nearest Neighbors

Hyperparameter Tuning - n and the weights.(n means number of neighbors)

From Figure 10, it indicates that the accuracy increases first and then decreases as n increases. The reason is that the increase of n is better for estimation at the first phase, but more and more points become less relevant (too far away) when n becomes too large, which will hurt the accuracy. At the same time, the weights of the neighbors also contribute to accuracy. There are two weights relationship, between the current point and its neighbors, Uniform and Distance. From the figure, the Uniform improves the overfitting condition, the Distance can bring better accuracy.

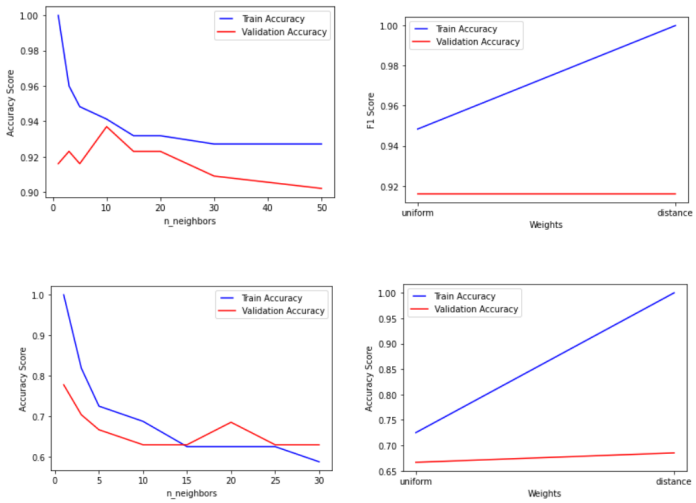

Figure 10—n and weights impact accuracy. Left Cancer, right Glass

Training - As shown in Figure 11, training size increase result in accuracy's increase. And the curve in Breast Cancer becomes flatten gradually. But the performance on Glass dataset seems to be under-fitting. I plan to add more data to test, and achieve k-NN model's convergence.

3 COMPARISON & CONCLUSION

Table 1—Accuracy of the 5 algorithms on two datasets.

Dataset	**score_type**	**DT**	**Boosting**	**SVM**	**NN**	**k-NN**
Breast Cancer	*train_score*	0.895	0.923	0.937	0.909	0.937
Breast Cancer	*cv_score*	0.925	0.911	0.944	0.934	0.932
Glass	*train_score*	0.667	0.667	0.704	0.722	0.630
Glass	cv_score	0.619	0.513	0.688	0.650	0.588

From Table 1, it's clear that all the algorithms work better on Breast Cancer problem, which is a binary-classification. I though it might result from the lack of samples. Therefore, I plan to implement k-fold cross validation to see if it would help.

4 REFERENCES

1. B. German, "Glass Identification Data Set", UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/glass+identification
2. William H. Wolberg,W. Nick Street,Olvi L. Mangasarian, "Breast Cancer Wisconsin (Diagnostic) Data Set", UCI Machine Learning Repository.https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)