Assignment 1: Supervised Learning

Naziul Talukder
ntalukder6@gatech.edu

1 INTRODUCTION

In this paper, I will analyze two datasets from the UCI database. I will use different supervised learning algorithms for each dataset to solve a classification problem, analyze their efficiency, tune hyper-parameters appropriately to obtain the best model and the best algorithm to solve the problem.

One of the datasets is the heart disease dataset which contains 303 samples. Based on some patients attributes, the goal is to determine if the patient is likely to have a heart attack or not. The attributes contain personal information like age, sex; medical information like blood pressure, cholesterol, ECG results and condition of the heart. There are 165 examples where heart attack is likely and 138 examples where it is not likely. The dataset can be considered balanced.

The other dataset is the breast cancer dataset which contains 569 samples. The features describe characteristics of the cell nuclei present in the digitized image of a fine needle aspirate (FNA) of a breast mass. The dataset contains diagnosis: malignant or benign. The goal is to predict if a cell is malignant or benign looking at the characteristics of the cell. There are 30 features in the dataset describing the characteristics of the cell i.e. radius, texture, concavity, compactness, perimeter, symmetry etc. There are 212 malignant and 357 benign samples.

I sought a dataset with a greater number of features compared to the other dataset. The heart disease dataset contains 13 features and cancer dataset contains 30 features. I sought one dataset to have less examples than the other one. Heart dataset contains 303 samples whereas cancer dataset contains 569 samples. I did not want to choose an extremely large dataset to avoid the need for GPU to train the models. Both of my datasets hence contain less than 1000 samples. I chose datasets that are popular in ML world to avoid complex behavior in training the models. Because both datasets can be considered balanced, I used accuracy score as the metric of performance for both. Because they are both binary classification problem, accuracy score:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is

False Negative.

2 KNN

KNN is a non-parametric supervised learning model where the output is classified by the votes of its surrounding neighbors. For the heart disease dataset, I ran a simulation varying the nearest neighbors and the weights of the datapoints to observe the best accuracy score as shown in figure 1. With higher and higher neighbors the model's performance on the validation dataset did not improve. On the left side in the figure, the model overfits the data when neighbors chosen between 1 to 4. In this region, the training score increases but the validation score decreases. n = 9, provides the best validation score. Figure 1 left side shows the impact of uniform weights where all points in each neighborhood are weighted equally. On the right side weight of points depend on the distance: closer neighbors have greater influence. The later approach has higher accuracy on the training set but has similar trend on validation set as uniform weights. The validation scores do not improve compared to the uniform weights scenario. There are 6 instances of accuracy score less than 0.6 when uniform weights are considered. Whereas there are 12 instances of accuracy score less than 0.6 when weights depended on distances. From the hyperparameter tuning in KNN, n = 9 neighbors and uniform weights is likely to provide the best model.

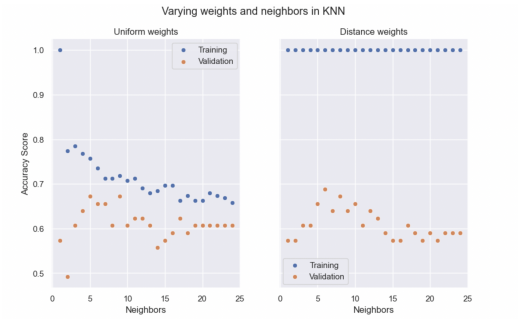

Figure 1—Neighbors and weights against accuracy score (heart disease data)

Figure 2 shows a similar comparison on the cancer dataset. Varying weights show similar trend in the accuracy score but distance based weights tend to have higher gap between training and validation scores. Neighbors in range (1-2) overfits the data using uniform weights. The validation score tends to be higher for neighbors in range (3-15), which indicates a disconnection between training

and validation dataset. There could be difference in temporal or spatial pattern in the training and validation set. Using 3 neighbors and uniform weights would provide the optimum KNN model for this dataset.

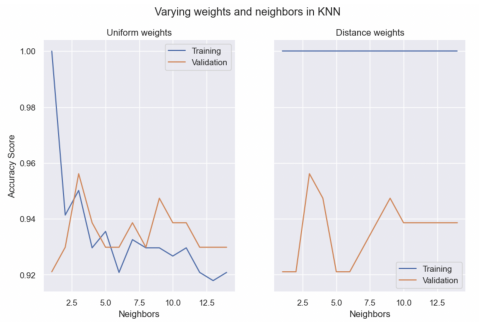

Figure 2—Neighbors and weights against accuracy score (cancer data)

3 SVM

Figure 3 shows how the accuracy score changes with implementation of various kernels and regularization parameters, C in heart disease dataset. In SVM, for higher C values, the optimization chooses a smaller-margin hyperplane ensuring that it classifies the examples correctly. Whereas for lower C values, the margin is larger ignoring the resultant misclassification of training examples. SVM with linear kernel tends to outperform every other kernel on both training and validation scores in figure 3. This implies that dataset has a linearly separable relationship. Figure 4 shows how the validation score decreases as higher order of polynomial is used as the kernel. Figure 3 shows the validation score is also lower for RBF and Sigmoid kernels. This strengthens the assumption that the dataset is linearly separable best using a linear kernel. 3 shows that linear Kernel with regularization parameter C = 1.4874 provides the best validation result.

Figure 5 shows a similar analysis on the cancer dataset. The figure only shows C in range of 0.001 to 2, because this is where the best performance is obtained. With higher C value there is little or no improvement on the accuracy score depending on the kernel. Linear kernel with C = 1.2 is the provides the most optimum model. But polynomial and rbf kernels also provide high validation score. It implies the cancer dataset has non-linear structure. If it were strictly linear the other kernels would perform poorly like figure 3 shows for the heart disease dataset. A comparison of figures 3 and 5 suggests that the cancer dataset has

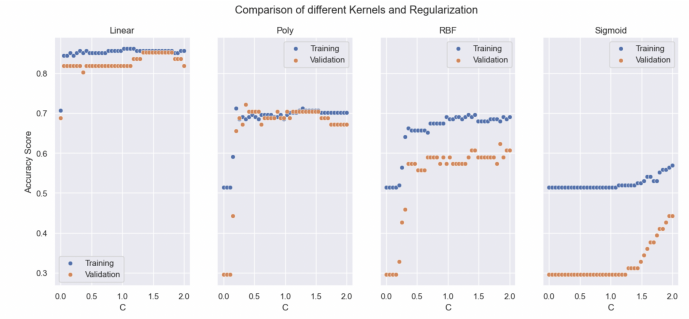

Figure 3—Kernels and regularization tuning (heart disease data)

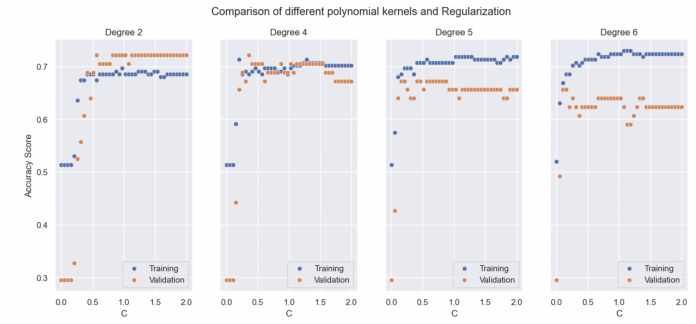

Figure 4—Accuracy Score with poly kernels and Regularization

more non-linear relationship compared to the heart disease dataset. Rbf and poly kernels are accurately capturing the non-linear relationship in the features for cancer dataset. But the dataset might have a linear decision boundary separating the classes which is why linear kernel outperforms others. The high validation score of linear kernel implies, the non-linear relationships in the features are not strong enough to use rbf or polynomial kernels.

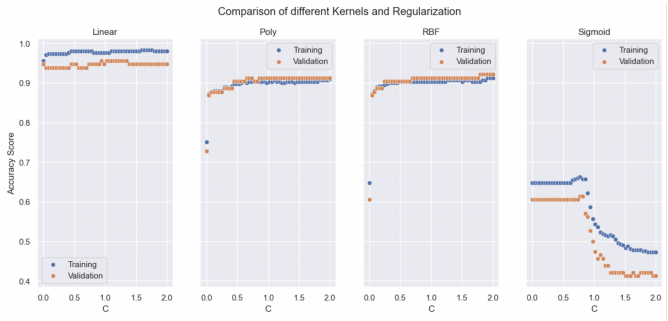

Figure 5—Kernels and regularization tuning (cancer data)

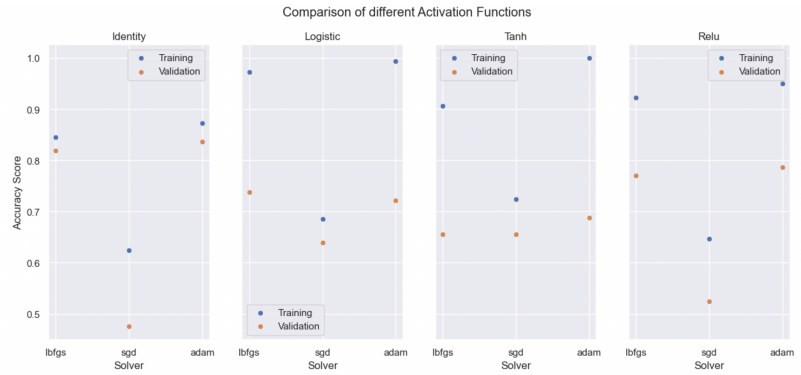

Figure 6—Accuracy score and activation functions relationship (heart disease data)

4 NN

I implemented a multi-layer perceptron algorithm that is trained using backpropagation. I also used multiple activation functions as well as solver methods to tune the classifier on both datasets. I considered identity ($f(x) = x$), logistic ($f(x) = \frac{1}{(1+exp(-x))}$), tanh ($f(x) = tanh(x)$) and relu ($f(x) = max(0, x)$) activation functions. **lbfgs** is an optimizer in the family of quasi-Newton methods, **sgd** refers to stochastic gradient descent, **adam** refers to a novel stochastic gradient-based optimizer. For the heart disease dataset, maximum iteration was 15000 and one hidden layer with 100 neurons were used. Figure 6 shows how accuracy score changes with different activation function and solvers. Using sgd solver the models perform poorly. In the SVM section the dataset demonstrated a behavior of being linearly separable and linear relationship. The activation functions that captures the linear relationship best is expected to perform the best. From the definition of the activation functions above, it's natural that the identity and relu functions tend to outperform others as seen in figure 6. Because this is a small dataset, lbfgs is expected to converge faster and perform better than adam. Figure 8 confirms this expectation. The accuracy score is lower for less than 50 iterations for lbfgs as it fails to converge. For higher iterations the accuracy score is close to one another. The lbfgs solver has more stable accuracy scores compared to adam for maximum iterations in range of 50 - 500. As lbfgs can converge with less iterations as opposed to adam, identity function with lbfgs would be the most optimum. Figure 7 shows that with more complex hidden layers the accuracy does not increase. A neural network with simpler hidden layer can be trained faster, can provide scores faster and less likely to overfit the

data. The most optimum neural network model can be obtained using 1 hidden layer with 100 nodes, identity as the activation function where lbfgs is used as the solver.

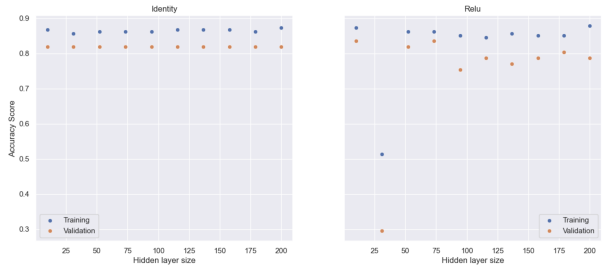

Figure 7—Accuracy score vs hidden layer size (heart disease data)

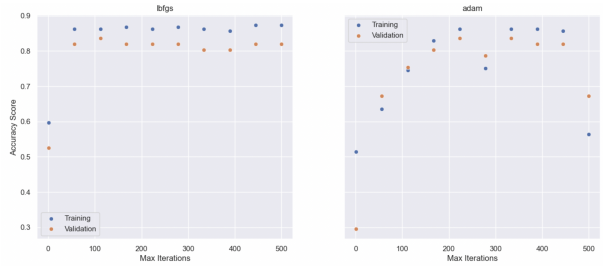

Figure 8—Loss curve of NN model varying activation (heart disease data)

On the cancer dataset I used similar simulations. The maximum iterations was 15000 and only 1 hidden layer with 100 neurons were used. Figure 9 shows how accuracy score changes with different activation function associated with a different solver method. The identity function using lbfgs solver fails to converge at global minima. Mostly, all other approaches provide high validation score. In figures 10 and 11 identity function with lbfgs provides better performance demonstrating that its performance degrades with too many iterations in figure 9. Similarly to SVM section, non-linear activation functions such as logisitc and tanh provides high validation score. Among all the combinations relu provides the best validation score with adam as solver method. Figure 10 shows that identity function outperforms relu for a simpler neural network. The identity function uses lbfgs solver and relu uses adam solver. A simpler neural network is faster to train, can provide scores faster and less likely to overfit. Figure 11 shows that identity activation function with lbfgs solver converges faster with smaller iterations than relu activation function with adam solver. The relu function uses 20 neurons in the hidden layer whereas identity uses only 1. For the cancer

dataset identity activation function with lbfgs solver and 1 neuron in the hidden layer provides the simplest model and best performance.

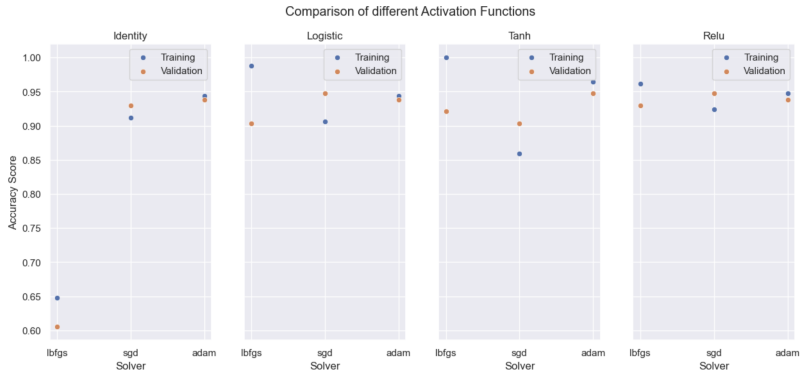

Figure 9—Accuracy score and activation functions relationship (cancer data)

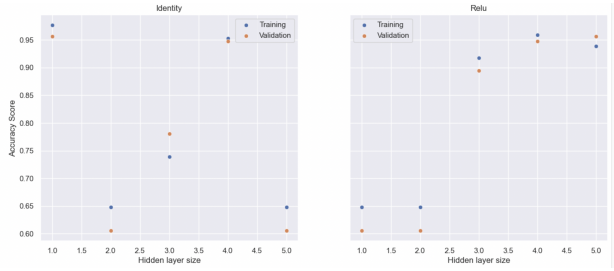

Figure 10—Accuracy score vs hidden layer size (cancer data)

5 DECISION TREE

I used gini impurity to decide the feature to split on in decision tree model. It can be calculated from

$$\mathrm{GiniIndex} = \sum_{i} p_i^2$$

where p_i is the probability of class i. The optimum split is chosen by the features with less Gini Index. Figure 12 shows how accuracy score changes with tree depth for the heart disease dataset. In figure 12 for higher tree depth, the model overfits the data. With pruning, maximum depth as 7 provides the best accuracy score observed in the figure 12. With extreme pruning, the model fails to generalize properly (tree depth 1 or 2) underfitting the data. The decision tree denotes number of major vessels (0-3) colored by fluoroscopy as the most important feature in the heart disease dataset.

Figure 13 shows a similar analysis on the cancer dataset. The highest accuracy

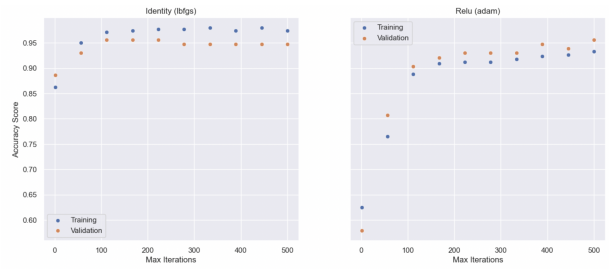

Figure 11—Loss curve of NN model varying activation (cancer data)

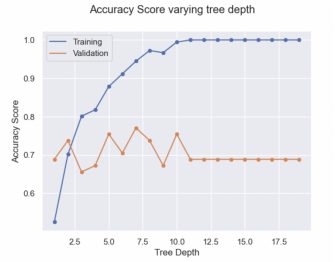

Figure 12—Decision tree varying tree depth (heart disease data)

score is obtained at 6 as maximum tree depth. With higher maximum depth the model becomes extremely complex overfitting the data. With shorter trees the model fails to generalize properly, resulting in lower validation score. The optimum decision tree model denotes concave points worst as the most important feature in the cancer dataset.

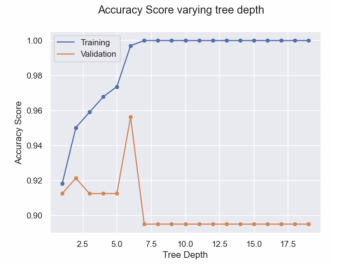

Figure 13—Decision tree varying tree depth (cancer data)

6 BOOSTING

In this implementation, the classifier is fit on the dataset and additional replicas of the classifier is also fit on the dataset adjusting the weights to ensure misclassified labels are prioritized over past fits. Figure 14 shows how accuracy score changes with using an adaptive boosted classifier and varying the depth of the

tree as well as the number of trees on the heart disease dataset. A weak learner is used as the classifier for the boosted trees because strong learners are prone to overfitting. Figure 14 shows that with higher depth the boosted tree tends to overfit for a high number of estimators. The best accuracy score is obtained using tree with depth 1 and 4 estimators in the heart disease dataset.

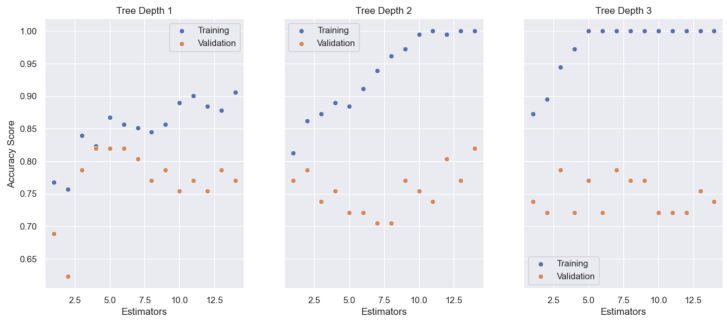

Figure 14—Accuracy score and number of trees in adaptive boosted trees with varying tree depth (heart disease data)

Figure 15 shows a similar implementation on the cancer dataset. The figure shows how trees with depth 1, 2 and 3 were used as weak learners and how accuracy score changes with more and more trees to classify the data accurately. A weak learner would need to have accuracy higher than 50% in this case, but the trees chosen have much higher accuracy as figure 13 shows. Due to having such high accuracy, in the figure the training score reaches 1 as number of estimators increases. Figure 15 shows the optimum boosted model has tree depth 3 and 17 estimators.

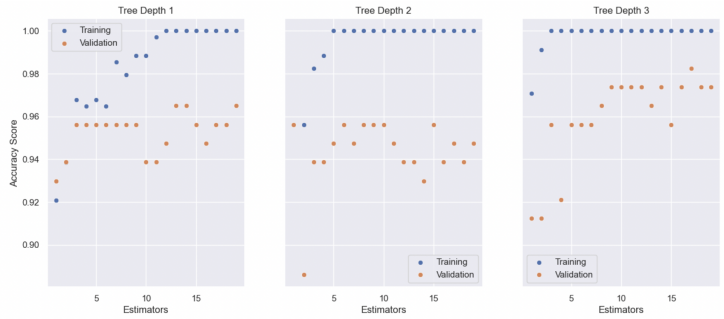

Figure 15—Accuracy score and number of trees in adaptive boosted trees with varying tree depth (cancer data)

7 LEARNING CURVE

Figure 16 shows the learning curve of algorithms we discussed: SVM, KNN, Decision Tree, Boosted Decision Tree, Neural Network with 5 fold CV on the heart disease dataset. Due to having a small dataset the neural network graph has high error, it failed to converge when we trained it on fewer data. The curve demonstrates that adaptive boosted tree performs better compared to simple decision tree showing less gap between training and validation score and higher accuracy score for validation curve. The figure suggests having more examples to train would improve the performance. Comparing the validation scores, SVM performed best even with our small dataset.

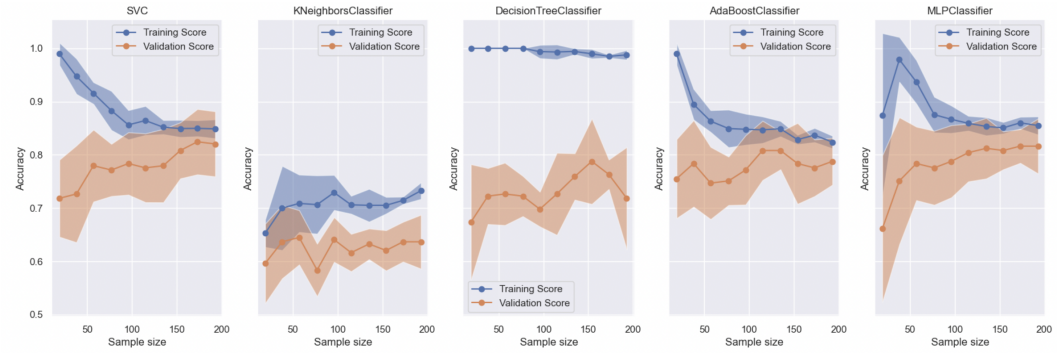

Figure 16—Learning curve of algorithms with 5 fold CV (heart disease data)

Figure 17 shows a similar learning curve analysis on cancer dataset. Mostly all the algorithms have high validation scores demonstrating that our dataset contains ample examples to learn from. Neural Network fails to perform as sample sizes shrinks lower than 100 samples. But with more data, the error tends to shrink. Decision Tree has the lowest accuracy score and Boosted Decision Tree demonstrates that boosting helps improving the score. Boosted decision trees shows the highest validation score among all the algorithms.

8 TIME COMPARISONS

Figure 18 shows the comparison between training and scoring time obtained on heart disease dataset through 5 fold CV. KNN, Decision Tree and Boosted Tree models are trained very quickly compared to SVM or Neural Network models. KNN and Boosted Tree models take longer than Decision Tree models in terms of scoring time. SVM takes longer to train the model but scoring is faster. Neural

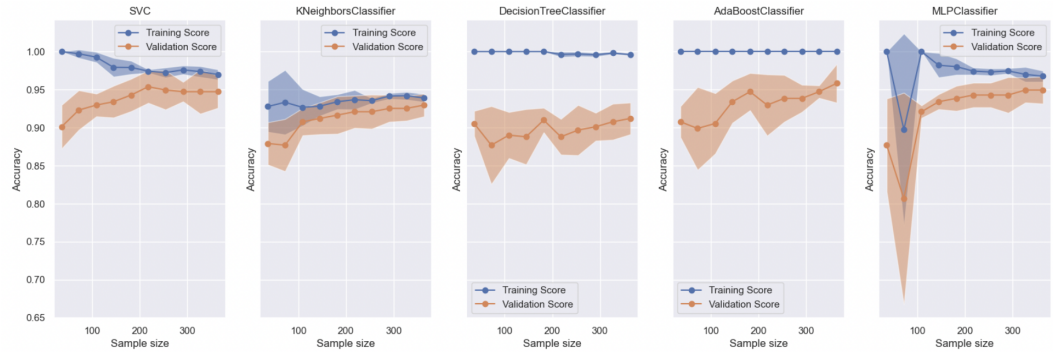

Figure 17—Learning curve of algorithms with 5 fold CV (cancer data)

Network takes the longest to score. In this small dataset, SVM's training time can be endurable and it is likely outperform other models.

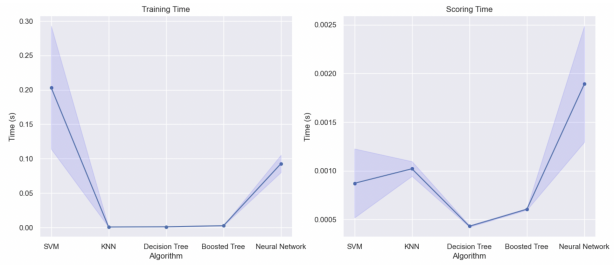

Figure 18—Time comparison of algorithms (heart disease data)

Figure 19 shows similar analysis on the cancer dataset. SVM takes the longest time to train the model and KNN takes the longest time to score. Boosted decision tree has the best validation score in figure 17. Only decision tree outperforms boosted decision trees in terms of scoring time. But simple decision tree has lower accuracy score than boosted decision tree as seen in figure 17.

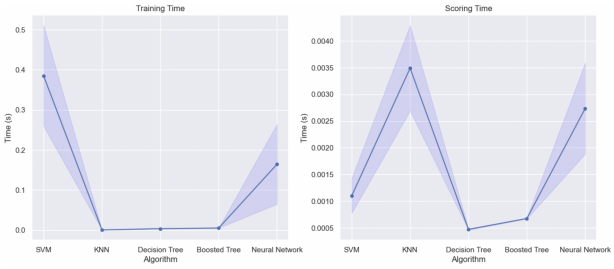

Figure 19—Time comparison of algorithms (cancer data)

9 TEST DATASET PERFORMANCE

Figure 20 shows how the most optimum model from each algorithm performs against the test dataset from heart disease dataset. SVM and Neural Network models tend to outperform all the other models. This is expected as the dataset demonstrated linear relationship and these models are tuned to capture it better than others. KNN performs the worst on test dataset. The figure also demonstrates how boosting remediates overfitting and improves performance of decision trees.

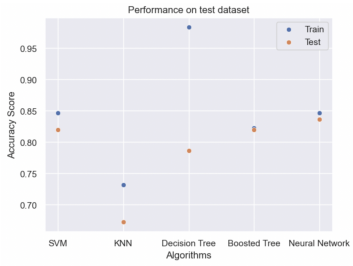

Figure 20—Performance of optimum algorithms on test data (heart disease data)

Figure 21 shows similar analysis on cancer dataset. As expected we observe the best accuracy score on test dataset for the boosted decision tree model. Even though KNN's validation score is lower compared to SVM in figure 17, in the test dataset it outperforms SVM. Neural Network model showed promise in figure 17 but in test dataset it performs poorly compared to other models. Decision Tree and Neural Network has the lowest accuracy score. Taking training time, scoring time, accuracy score on validation set and test set into account: adaptive boosted decision tree provides the most optimum model for the cancer dataset.

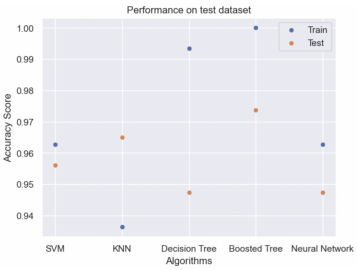

Figure 21—Performance of optimum algorithms on test data (cancer data)