Assignment 1 – Supervised Learning

Sowmya Yellapragada
(syellapragada3 - 903351714)

Abstract

This paper reports the classification analysis of supervised learning algorithms on two UCI datasets – "Statlog (Landsat Satellite)" and "Spam email database". The following algorithms were used to conduct the classification: Decision Trees, Boosting, K Nearest Neighbors, Support Vector Machines and Artificial Neural Networks.

Data Sets

1. **Statlog (Landsat Satellite):**
   Soil classification is of particular importance in large and developing countries, like India, where huge percentage of the population still relies on agriculture. Broadly classifying the different soil types in the country, will help the government or organizations identify which crops would be more suitable to grow in which areas.

   Here we analyze the classification of soil into – red soil (1), cotton crop(2), grey soil(3), damp grey soil(4), soil with vegetation stubble(5), mixture class(6) and very damp grey soil(7). The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighborhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighborhood and a number indicating the classification label of the central pixel. Thus, there are 36 attributes, each of which has integer values in the range of 0 and 255. The dataset was classified into testing and training data sets, containing 4435 and 2000 examples respectively.

2. **Spam E-mail Database:**
   Spam email is a major concern for most people using an email service. Important emails are often lost among the huge dump of spam emails. Simple classification based on what the user had marked spam is no longer sufficient. Machine learning algorithms have become common place for identifying trends among data that at the first sight looks random or unsuspicious. In this report we explore classification of email into spam (1) or not spam (0). Most of the 58 attributes in the data set indicate whether a particular word or character was frequently occurring in the e-mail. The dataset containing 4601 instances was classified into testing and training split containing 25%, 75% of the instances respectively.

The algorithms were implemented using the python scikit-learn library, which provides simple and efficient tools for data analysis and machine learning algorithms. First the algorithms were implemented multiple times on cross validated train data, by modifying the hyper parameters, to identify the parameters that predict results with highest accuracy. Then these hyper parameters

were used to train the algorithm using train data and further test it against test data. Here we discuss the cross-validation accuracies of different algorithms and then provide a comparative study of the prediction accuracies of all the algorithms on the test data.

Algorithm Implementation and Analysis

I. Decision Trees (with pruning)

Decision trees are classifiers which utilize a tree structure to model the relationships among the features and the output. Decision tree were implemented using scikit's decision tree classifier. First the variation in accuracy was observed by varying the minimum samples split. Using the split that proves to generate results with highest accuracy, we further vary the maximum depth.

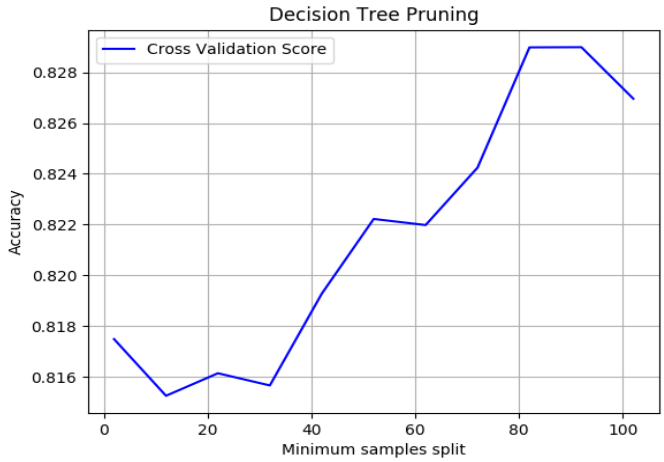

Figure 1: Varying minimum samples split hyper parameter(Statlog)

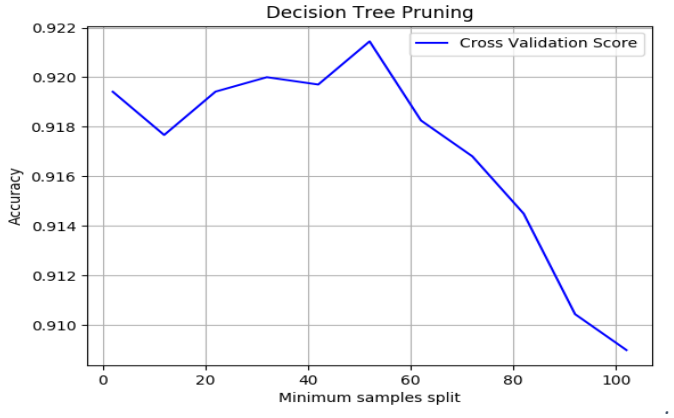

Figure 2 Varying minimum sample split hyper parameter (Spam)

As can be clearly noted from the graph above, the maximum accuracy was attained at the minimum sample split size of 80 (approximately) for the statlog data set and is around 50 for the spam data set. Low accuracies at low values of the split size, indicates that there is a

significant variation and noise in the data. Low split size will cause an overfitting of the data, as it allows further classification to accommodate oddities. Hence for the statlog data set, a high split value was more suitable. However, in comparison, the accuracies at lower split values is comparable to the highest accuracy in the case of spam data

We further pruned the decision tree by varying the maximum depth parameter. It can be observed from the graph below. As can be noted, the accuracy of the decision tree for Statlog data peaks at a maximum depth of 50 and gradually decreases thereafter. By setting a maximum depth to the decision tree we are restricting to the number of steps the decision tree can continue to do until every node has less the minimum split size of nodes. As discussed when there is a significant noise in the data there will be a high chance of overfitting if the decision tree is allowed to build up on till it reaches pure leaves.

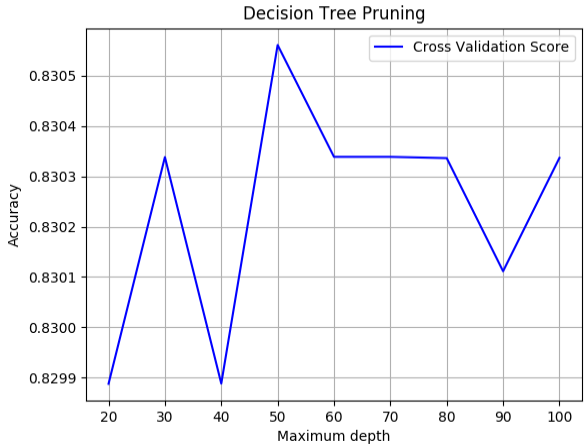

Figure 3: Maximum depth variation (Statlog)

On the contrary, for the spam data we can observe that for the spam data high accuracy was obtained at a low maximum depth. Comparable accuracies were achieved at the maximum depths of 50 and 70 as well, but we know that lower is a better parameter in case of the same validation accuracy, because increasing depth may lead to over fitting the data. Hence, we always prefer small depth in final model building

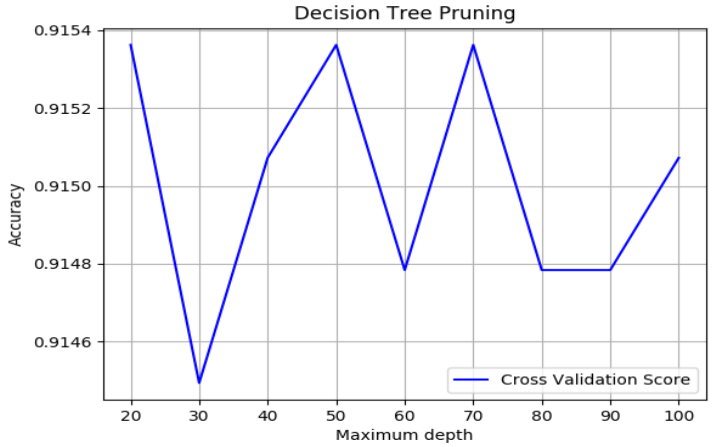

Figure 4 Maximum depth variation (Spam)

II. K- Nearest Neighbors

The K-Nearest Neighbors model is the one of the simplest algorithm and a lazy learner. It uses the information about its neighbors to classify the unknown label. For the purpose of this analysis we use the Euclidean distance metric and varied the number of nearest neighbors used to make a prediction

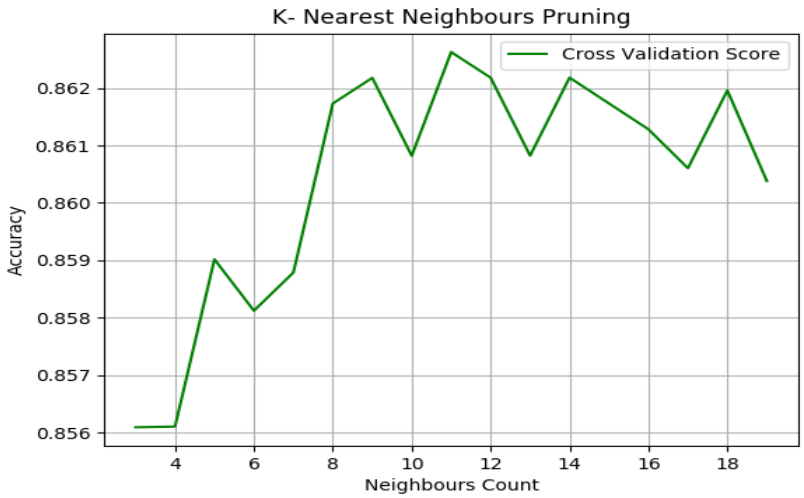

Figure 5 Neighbor count variation (Statlog)

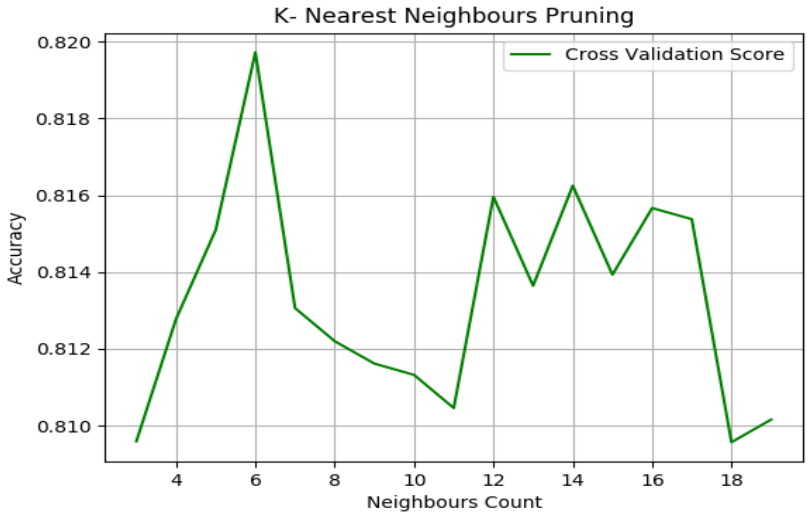

Figure 6 Neighbor count variation (spam)

Although for the Statlog data, the accuracies are comparable for k values between 8 and 12, the maximum accuracy seems to be attained at 11. As you increase the value of k, smoothing takes place and eventually we reach a stage where the data is under fit rather than over fitting. Choosing a larger value with high accuracy, allows us to avoid overfitting by generating a smoothed prediction. Hence, for the Statlog data, 11 would be an ideal value for K. For Spam data however, a peak is observed at a relatively lower value and there is a steep drop in accuracy, indicating that an increase in K further causes under fitting.

III. Artificial Neural Networks (ANN)

Neural networks are a learning structure designed to mimic the function of a neuron. For this analysis, the MLPClassifier of the scikit library was used. Different solver, activations and learning rates were experimented with cross validation, to identify the most suitable hyper parameters for our problem.

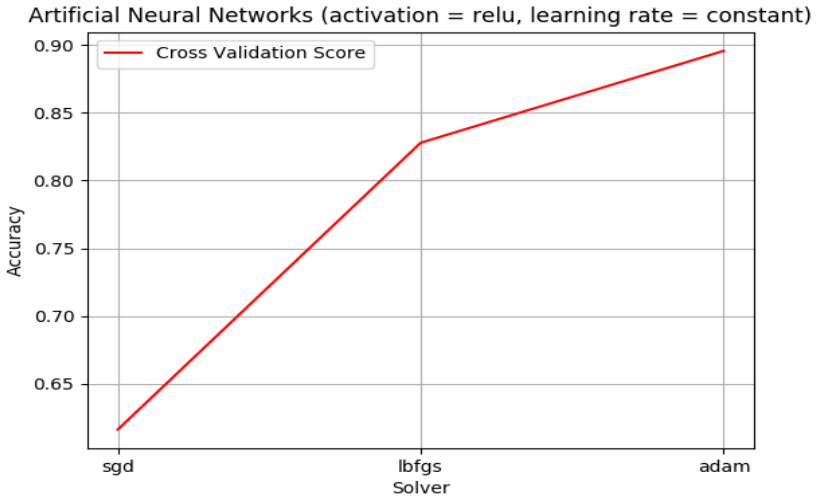

Figure 7 Varying solver for ANN (Spam)

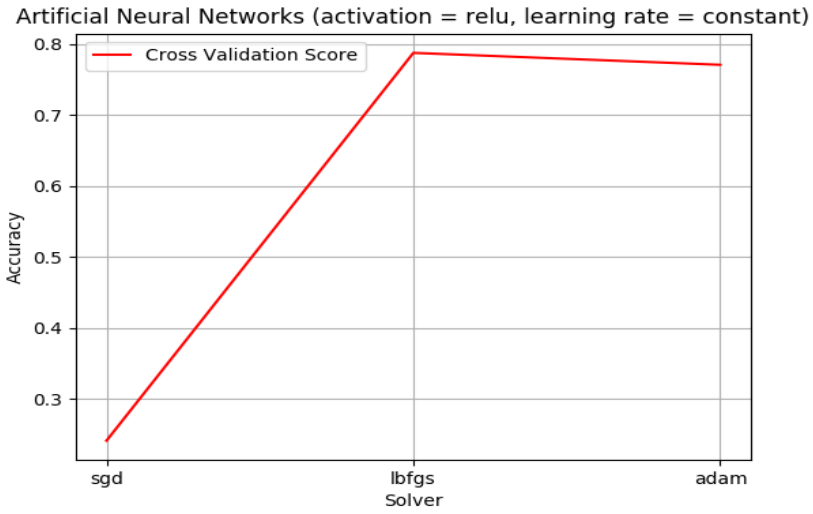

Figure 8 Varying solvers for ANN (Statlog)

As can be observed from the graph, both "lbfgs" and "adam" provide comparable accuracies for both the data sets. As both data sets are quite large (over 4000 instances), adam has been observed to be more suitable in terms of training time and accuracy. Further, the learning rate schedule for weight updates was varied as "constant", "invscaling" and "adaptive".

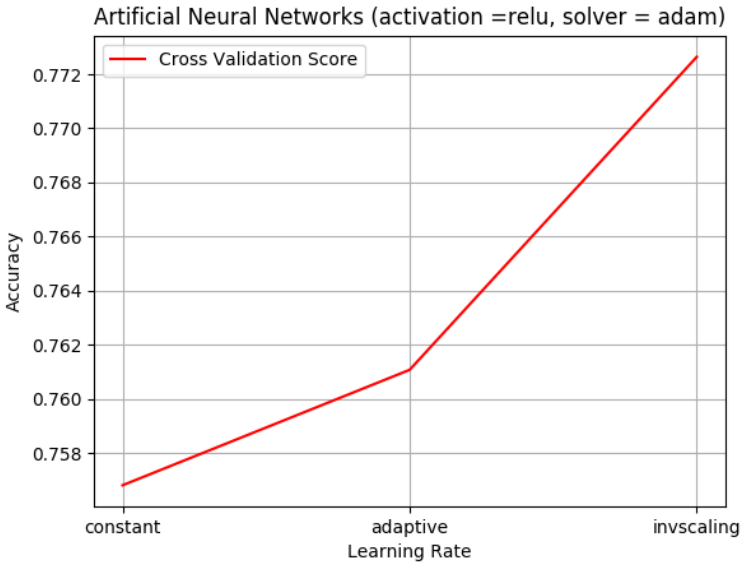

Figure 9 Varying the learning rate(Statlog)

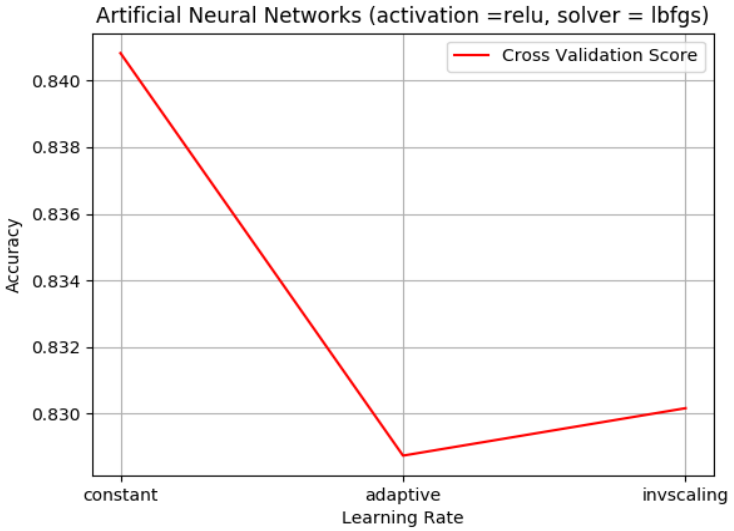

Figure 10 Varying the learning rate(Spam)

It can be observed that constant was more suitable in the case of spam data set, this can be attributed to the categorical nature of its output. Invscaling was noted to be more suitable to the statlog data.

IV. Support Vector Machines (SVM)

Support Vector Machines creates a boundary called a hyperplane, which divides the space to create a homogeneous partition on either side. The optimization in this model is performed by maximizing the distance the two decision boundaries. In order to separate non-linear models, it uses a kernel trick. Here we implement SVM using scikit learn's svc. Here we analyse the performance of three kinds of kernels: linear, rbf (radial basis function) and sigmoid. We further choose the best suitable one, and vary the penalty parameter to observe changes in accuracy.

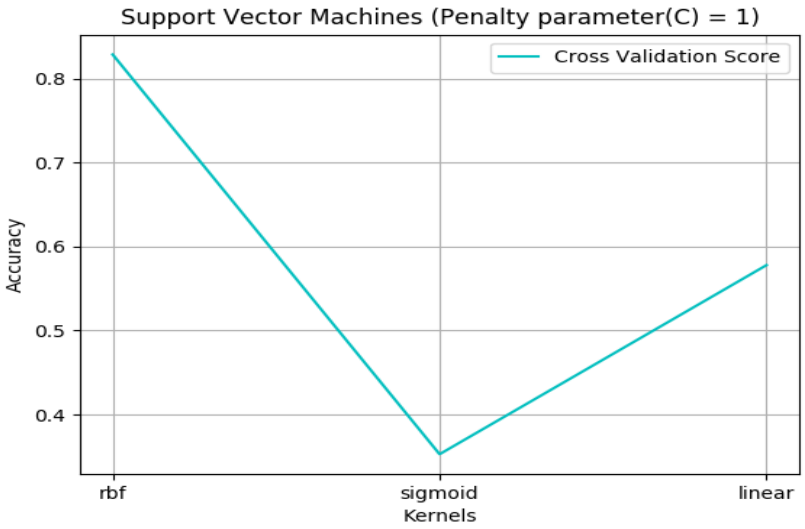

Figure 11 Varying Kernels(Spam)

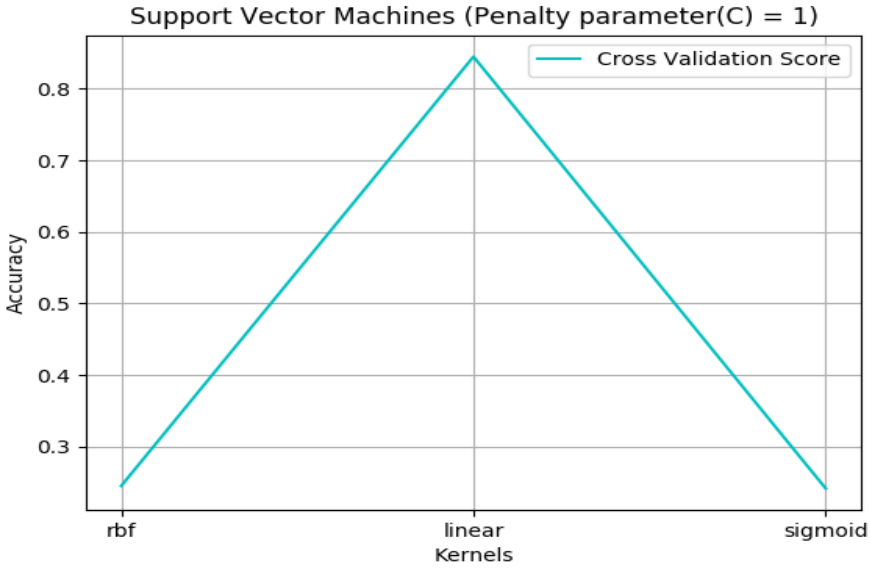

Figure 12 Varying Kernels (Statlog)

As can be observed from the graphs, rbf performs best for spam data and linear kernel works well for the stalog data. Further choosing the best kernel fit, we vary the penalty parameter. However, no meaningful change has been observed by changing this parameter.

V. Boosting

Boosting was performed using Adaboost package. Decision tree classifier was chosen as the weak learner on which boosting was applied. The following trends were observed.

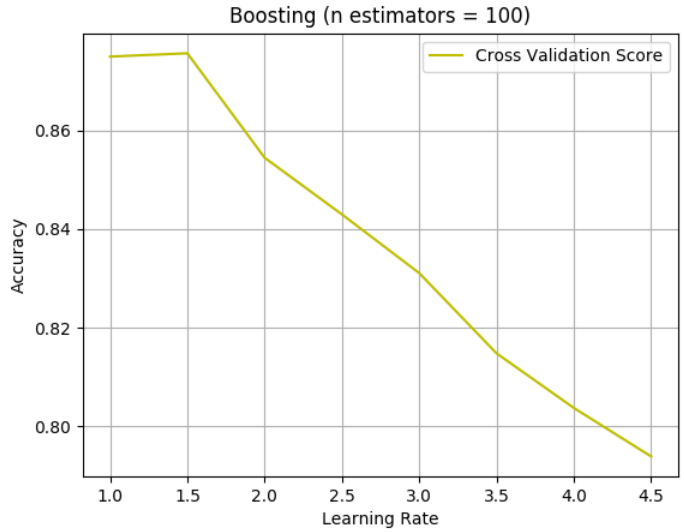

Figure 13 Stat log data (Learning rate)

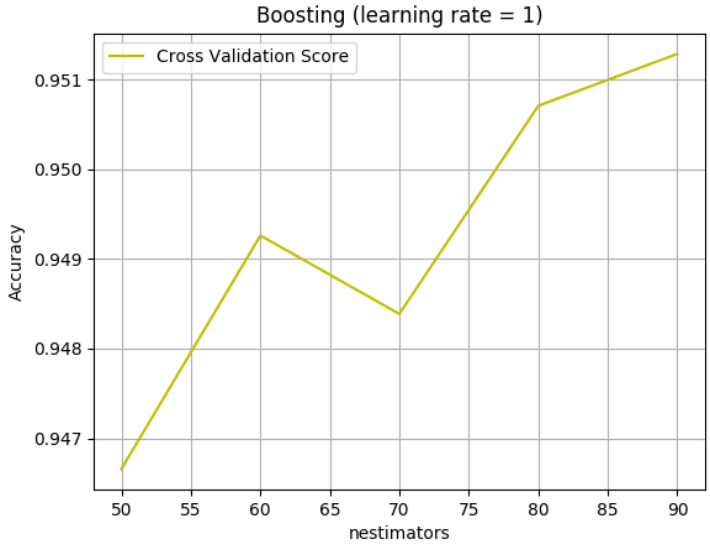

Figure 14 n estimators

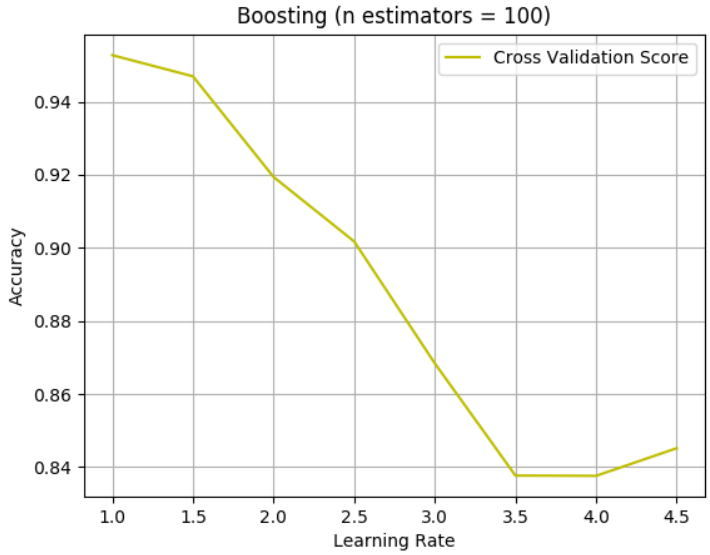

Figure 15 SPam data

A low learning rate has been observed to produce the most accurate results.

VI Comparative Study

By choosing the best parameter fit for each of the algorithms a comparative study was conducted to see their performances.

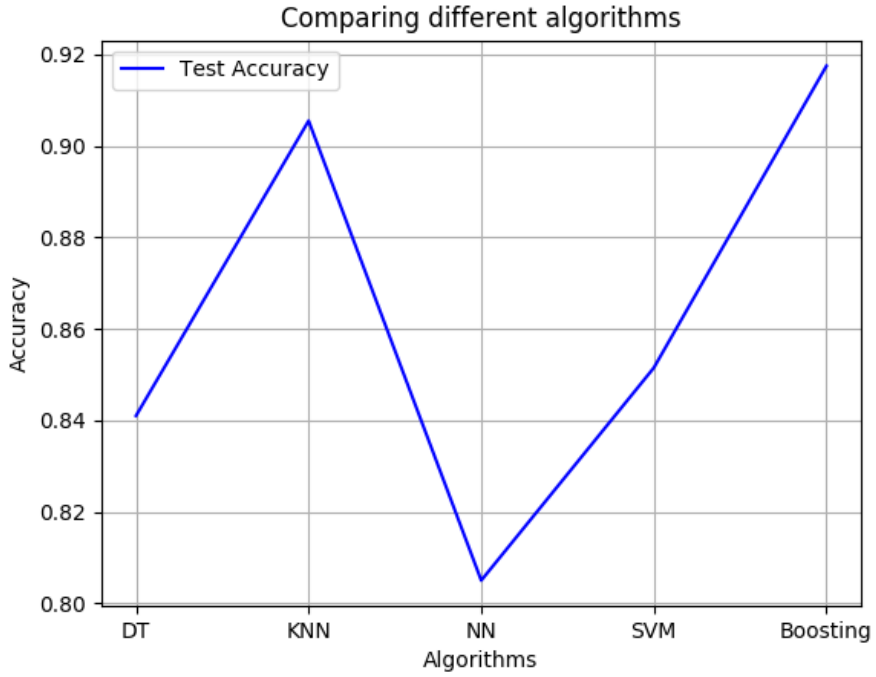

Figure 16 Statlog

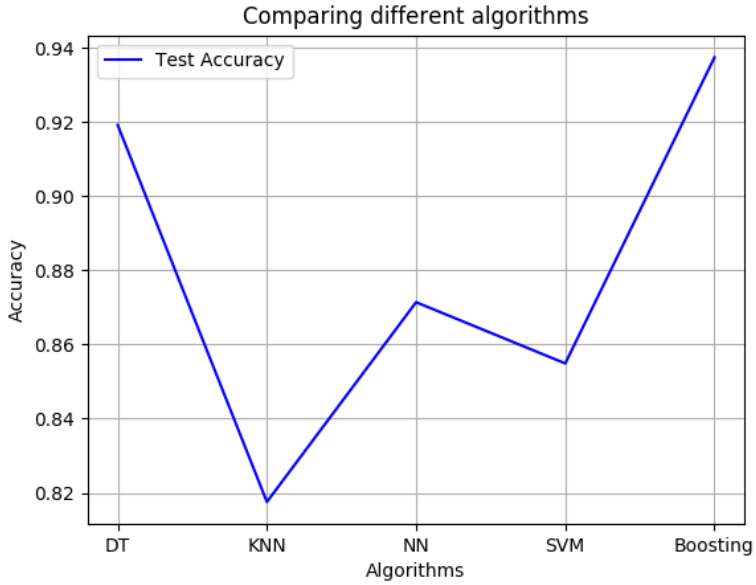

Figure 17 Spam data

Boosting applied on decision tree clearly works well for both the data sets. KNN seems to work better for statlog data, indicating that nearest neighbor are good indicator of a data points value. KNN might not be suitable to spam data, as the final output is categorical. SVM although perform moderately well in both the cases, this is because of the restriction put on the number of iterations

to limit its running time. Boosting, which is an improvement over the decision tree classifier seems to be the best choice for the classification of these data sets