Gen Mark Tanno

CS 4641 Assignment 1: Supervised Learning

Introduction

The task for this assignment was to analyze the performance of five algorithms: decision trees, neural networks, boosting, support vector machines (SVMs), and k-nearest neighbors (k-NN). These algorithms were tested on two data sets obtained from Kaggle. The first data set contains data on edible or poisonous mushrooms, and the second set contains data on pulsar stars.

Edible or Poisonous Mushrooms

A common question that pops up whenever people spot a mushroom is, "Is this mushroom edible?" In fact, it is a question that I myself always sought to find the answer to. It is possible to classify a mushroom's edibility by figuring out the biological features of it. However, one might not have the knowledge or tools to study these features and only have limited observations of a mushroom. The mushroom data set contains a total of 8124 data points and 22 attributes which include color, shape, and size of different parts of the mushroom. The data set contains only qualitative data which means that there are no numerical data.

Pulsar Stars

The second dataset I chose was on pulsar stars, as I am very enthusiastic about things outside Earth. A pulsar star is a neutron star or white dwarf that rotates and emit electromagnetic radiation. The classification problem that was set up was to determine whether a star was pulsar or not given mathematical data such as mean, standard deviation, and skewness of signal-to-noise ratio readings. Note that unlike the first dataset, this dataset contains numerical data. There are only total of 8 attributes which is less that that for the mushroom dataset, but there is a total of 17898 data points which is more than that of mushroom.

Discussion

We now look into how the different algorithms performed in the two datasets. For this project, I used scikit-learn which is Python library which features the algorithms necessary here. I also learned how to use Python's matplotlib library for creating the graphs. I also included cross validations in applicable graphs. For each of the algorithms, I set the training set to be 70 % and testing set to be 30%.

Decision Tree

The first algorithm used on the data set was the decision tree. I chose to use pre-pruning in which I changed the values of the max depth before running the main algorithm. I also chose to use a small step size of 1 because the decision tree algorithm only takes about a minute to run. Thus, I was able to retrieve an accurate graph that shows more detailed changes in each time step.

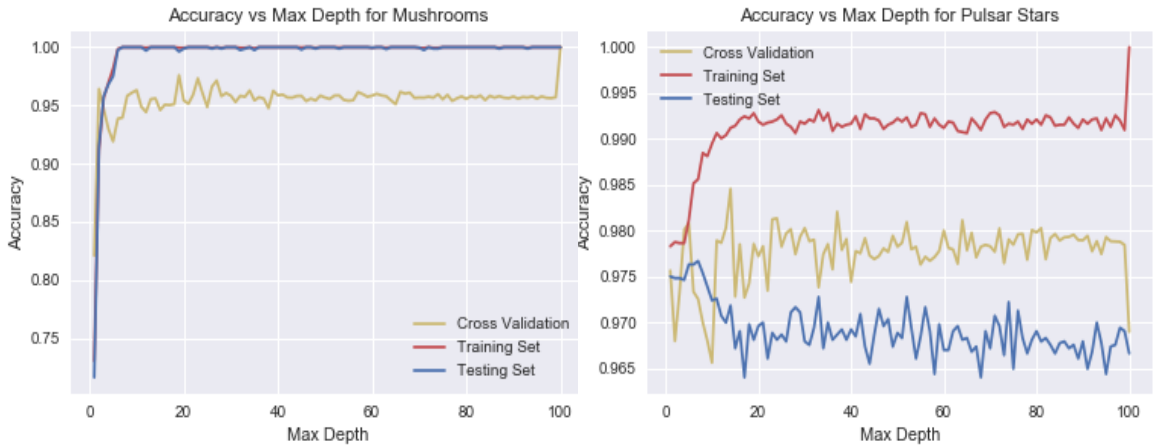

Figure 1: *Graphs showing the effect of pruning on the decision tree performance for mushrooms and pulsar stars datasets*

Figure 1 shows the relationship between the max depth and the accuracy. In the mushroom dataset the accuracy is significantly raised in the first few max depth increments. This is different from the result for the pulsar data set where the accuracy is already above 0.95. A max depth of 15 seems to be sufficient especially for the mushroom data set that achieved an accuracy of 1.

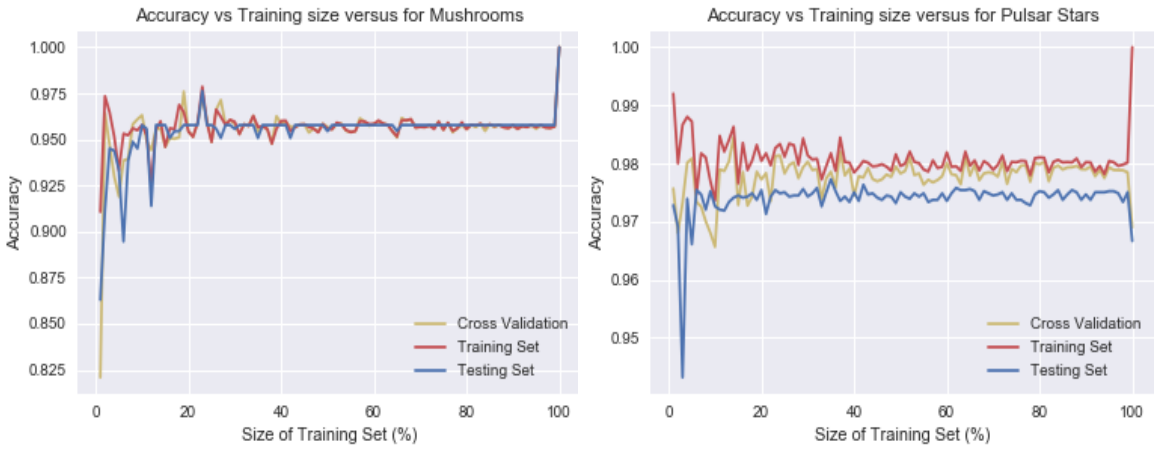

Figure 2: *Graphs showing the effect of increasing the training set size on the decision tree performance for mushrooms and pulsar stars datasets*

Figure 2 shows the relationship between the training set size and the accuracy. Increasing the training set size had a more positive effect on the accuracy for the mushroom dataset, as there is an increase in mushroom testing set's accuracy from 0.85 to 0.96. Both datasets had a nearly constant accuracy from a training set size of 20% which means that having more data points beyond some point does not make the decision tree to perform better.

Neural Networks

The next algorithm that I used on the datasets is neural networks. I decided to find the performance based on changes in the number of hidden layers, number of neurons in a hidden layer, and the size of the training set. The running time for neural networks was about 10 minutes.

Figure 3: *Graphs showing the effect of increasing the number of hidden layers on the neural networks performance for mushrooms and pulsar stars datasets*

Figure 3 shows the relationship between the number of hidden layers and the accuracy. It can be inferred that 1 hidden layer is optimal for the mushroom dataset as the model with the least number of hidden layers that results to the highest accuracy is preferred. On the other hand, 2 or 12 hidden layers seems to be the optimal for the pulsar star dataset. 2 hidden layers would be a better choice for a simpler model, and 12 layers might be better for a higher accuracy.

Figure 4: *Graphs showing the effect of increasing the number of neurons on the neural networks performance for mushrooms and pulsar stars datasets*

Figure 4 shows the relationship between the number of neurons and the accuracy. For the mushroom dataset, it can be inferred that 8 neurons or more than 10 neurons is optimal. There is a decrease in performance when there are 9 to 10 neurons, but an accuracy above 0.97 is still very high. For the pulsar star dataset, the graph does not provide a clear picture on the effect of increasing the neuron number. The accuracy oscillates as the neurons are increased and an accuracy of 1.0 is not achieved unlike the mushroom dataset. The frequency of the oscillation is high, and so it is difficult to find the optimal number of neurons for the pulsar star data set.

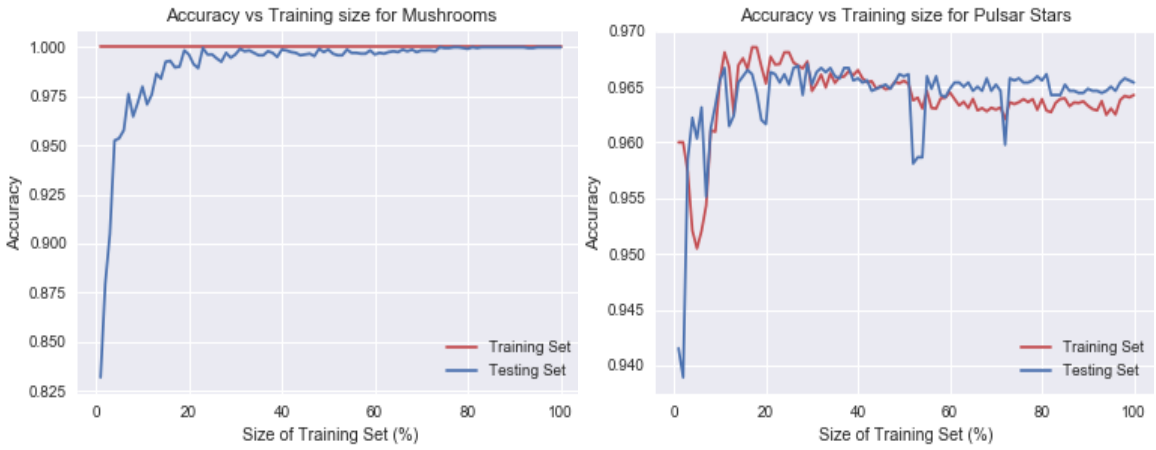

Figure 5: *Graphs showing the effect of increasing the training set size on the neural networks performance for mushrooms and pulsar stars datasets*

Figure 5 shows the relationship between the size of the training set and the accuracy. Increasing the training set size had a positive effect on both the mushroom and pulsar star datasets. It is important to choose a sufficiently large training data set to avoid possible underfitting. From the graphs, it is also evident that the accuracies of training and testing data sets are converging which is a sign of the model performing better.

Boosting

Boosting algorithms are used in several learning problems as they can reduce bias and variance. I chose to use the gradient tree boosting algorithm for this project. Note that a time step of 10 was used for the training size to keep the running time to about 10 to 15 minutes.

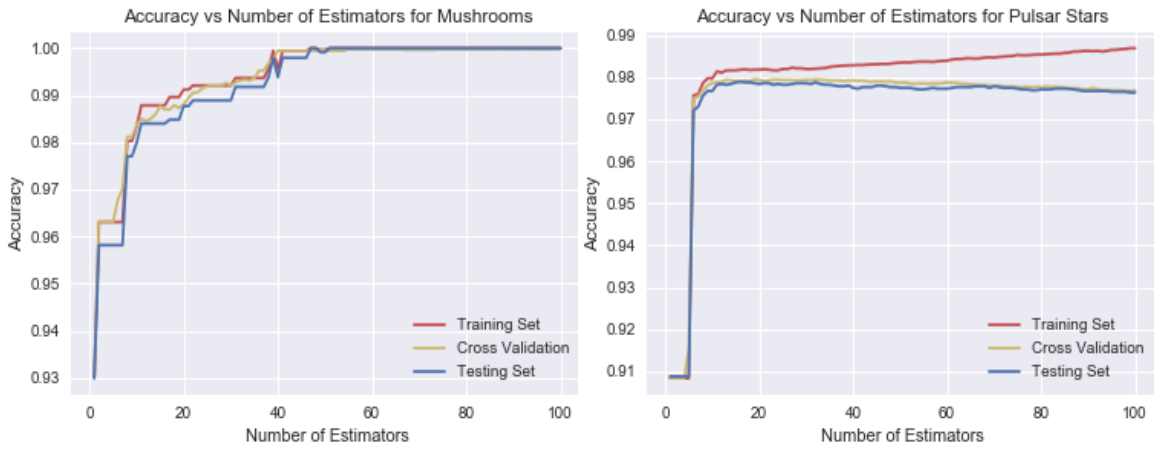

Figure 6: *Graphs showing the effect of increasing the number of estimators on the gradient tree boosting performance for mushrooms and pulsar stars datasets*

Figure 6 shows the relationship between the number of estimators and the accuracy. It is evident in the graph for mushrooms that increasing the number of estimators increases the accuracy. The optimal number of estimators is about 50 for the mushroom dataset. On the other hand, there is a huge increase in accuracy when the number of estimators is about 5 in the pulsar star dataset. Observing the testing set, one can see that the optimal number of estimators would be about 10 as the accuracy decreases a little beyond that point. The testing and training set accuracies are not converging which is another consideration for not increasing the number of estimators above 10.

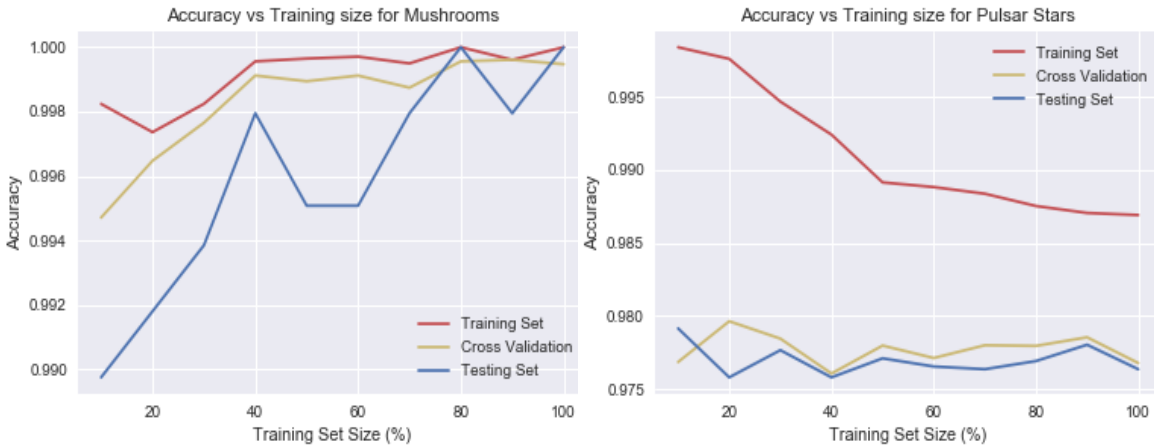

Figure 7: *Graphs showing the effect of increasing training set size on the gradient tree boosting performance for mushrooms and pulsar stars datasets*

Figure 7 shows the relationship between the size of the training set and the accuracy. For the mushroom dataset, increasing the training set size showed a positive effect on the accuracy. Furthermore, the training set and testing set are converging which is a sign that the performance is better for larger training set. For the pulsar star dataset, increasing the training set size does not increase the accuracy. Moreover, there is no clear sign whether the lines will converge unlike the mushroom dataset.

The gradient tree boosting algorithm was effective in both datasets because of a result of high accuracy. However, the algorithm seemed to have worked more effectively on the mushroom dataset than the pulsar star dataset.

Support Vector Machines (SVMs)

For the support vector machine algorithm, I chose the linear kernel to run on the datasets. For the sake of the project, a 10% step size for training data size was used to keep the program running time to less than 20 minutes.

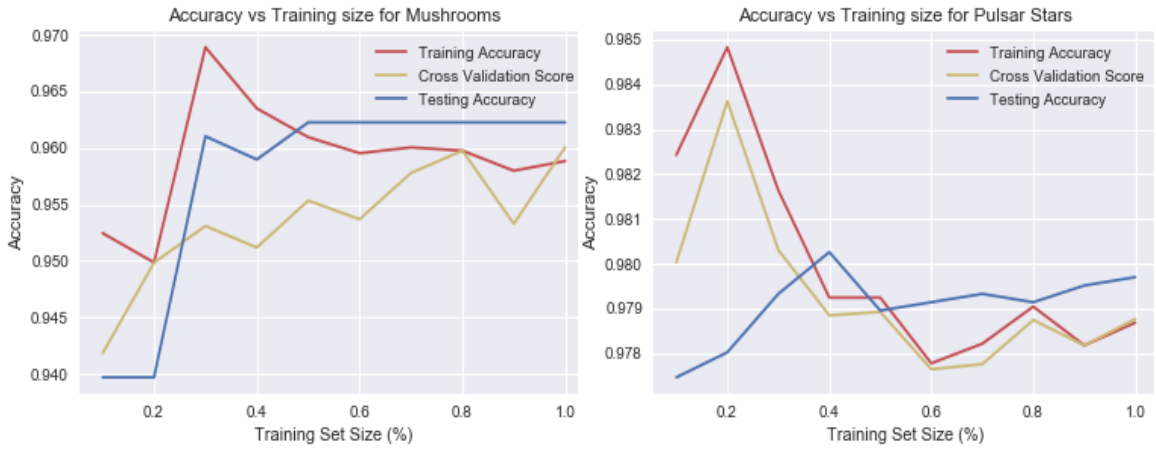

Figure 8: *Graphs showing the effect of increasing training set size on the SVM performance for mushrooms and pulsar stars datasets*

Figure 8 shows the relationship between the size of the training set and the accuracy. For the mushroom dataset, increasing the training set size showed a positive effect until an optimal training size of 30%. It is at this point where both training and testing set reach its global maximum. The accuracy value stays stable after this point. For the pulsar star data set, the optimal solution is also between a training set size of 20% and 30%. Unlike the mushroom dataset however, the pulsar star dataset's accuracy decreases after the optimal point. This means that increasing the training set size for the pulsar star dataset is not efficient when running SVM. From both graphs, it can be inferred that SVM is not highly dependent on the training set size.

K-Nearest Neighbors (k-NN)

Finally, the k-nearest neighbors algorithm was used on the datasets. The program first ran the algorithm to find the optimal k value. Then, the k value obtained was used to run iterations in finding the effect of changing the training set size on the performance. The running time of the algorithm is about a minute.

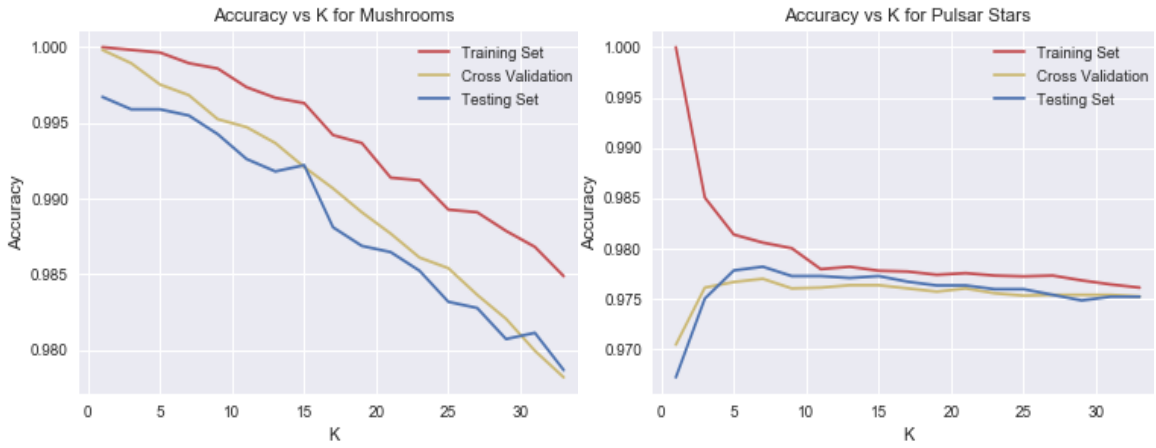

Figure 9: *Graphs showing the effect of k nearest neighbors on the k-NN performance for mushrooms and pulsar stars datasets*

Figure 9 shows the relationship between the size of the training set and the accuracy. The algorithm is more efficient in lower values of k for the mushroom data set. For the pulsar star dataset, the training set decreases and testing set increases as they converge at higher values of k. For both graphs, it can be seen that the training set reaches an accuracy value of 1 at starting k values. Also, the graphs are examples overfitting data because the training set is always at higher accuracies than the testing set. The difference is that the overfitting rate decreases in the pulsar star dataset as the training and testing sets converge.

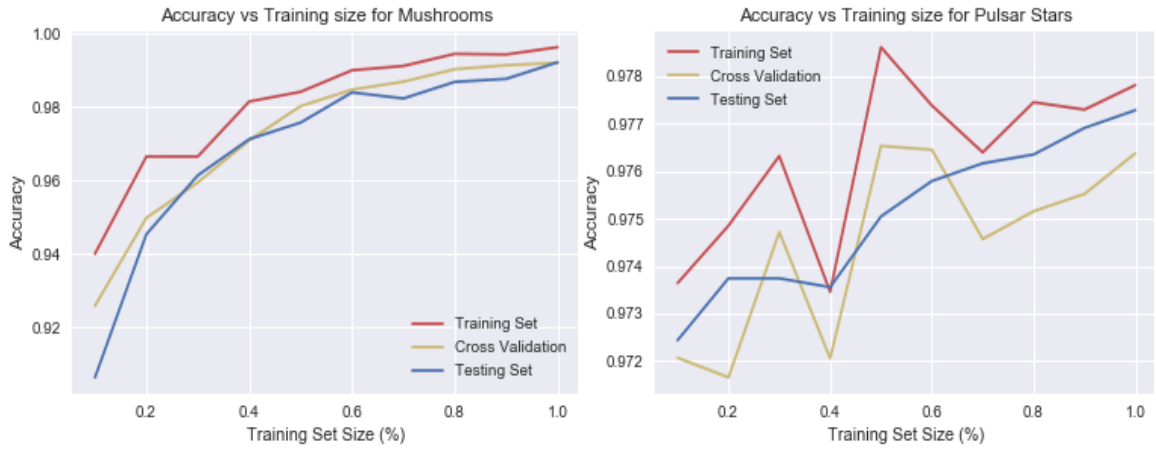

Figure 10: *Graphs showing the effect of increasing the training set size on the k-NN performance for mushrooms and pulsar stars datasets*

Figure 10 shows the relationship between the size of the training set and the accuracy. From both graphs, it can be inferred that increasing the training set size results in a positive effect for this algorithm. Again, the training set is always above the testing set which is an evidence that the algorithm performs better for seen data than unseen data.

Conclusion

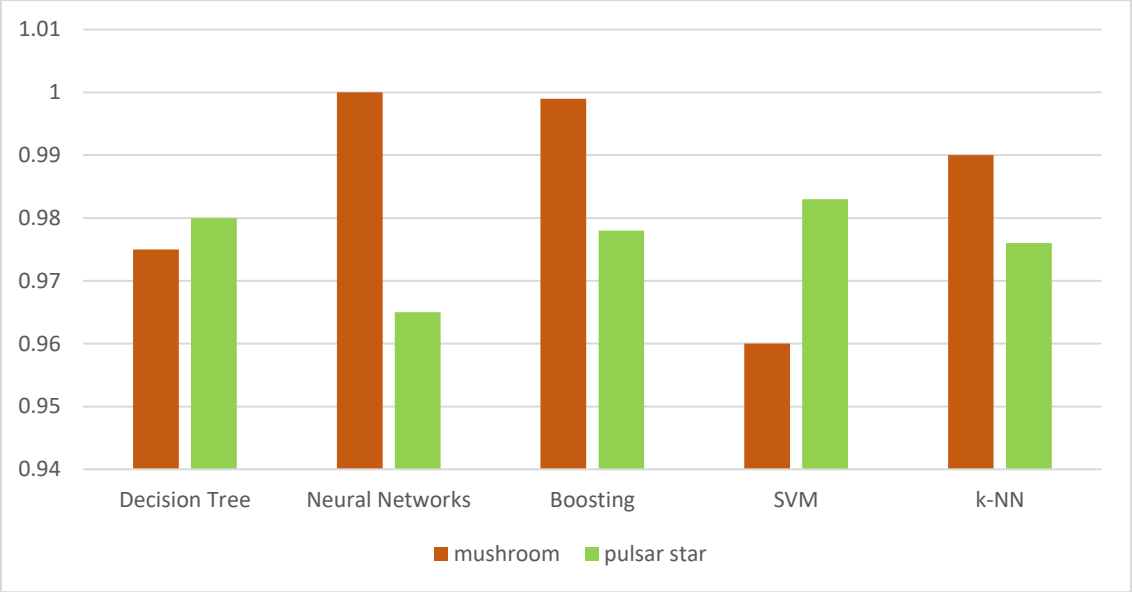

Figure 11: *Graph showing a comparison of the accuracies of the five algorithms on the mushroom and pulsar star datasets*

Figure 11 summarizes the performance of the different algorithms on the mushroom and puslar star dataset. It is evident that all the five algorithms performed very in classifying whether a mushroom is edible and whether a star is a pulsar star. It could be seen that neural networks had the highest accuracy for mushroom dataset and SVM had the highest accuracy for pulsar star dataset. However, boosting almost had the best performance in both datasets. Running the algorithm in other datasets is required to conclude whether one algorithm is more efficient than another.