Wenjun Wu
CS-4641
5 February 2017

Analysis of Supervised Learning Techniques

Introduction

This report explores the performance and characteristics of supervised learning techniques, which include: Decision Trees, Neural Networks, Boosting, Support Vector Machines and k-Nearest Neighbors. In order to determine the behavior of each algorithms, experiments were performed with various parameters of each algorithm.

Dataset Description

1. Wilt:

   This dataset is extracted from Quickbird imagery that detects diseased trees. The dataset consists of image segments, which were generated by segmenting the images using the "multiresolution Segmentation" algorithm [1]. The segments contain spectral information including standard deviation, grey-level co-occurrence matrix (GLCM) texture and mean spectral values for G, R and NIR bands. The task is to detect diseased and non-diseased trees from those multispectral image attributes. This dataset has a total of 4889 instances and 6 attributes. The dataset is split into 3387 training cases and 1452 testing cases (70% training and 30% testing).

   This dataset in very interesting. Successful detection of the conditions of trees and surrounding areas through remote sensors could significantly reduce the cost to maintain forests and prevent spreading of disease and harmful insects. It is also interesting because all the attributes are continuous were derived from sensors. This may set the foundation for many other advanced task in computer vision, such as pattern recognition, disease detection from histopathological images etc. Since the data is derived from real-world images, it should contain certain level of noise.

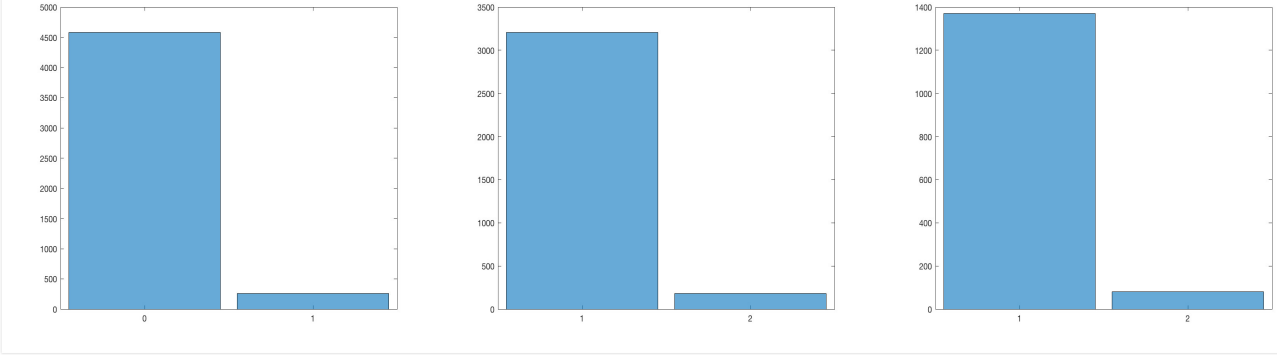


Figure 1 Histogram of Class Distribution of Wilt dataset. Left: Original Dataset; Middle: Training Dataset; Right: Test Dataset.

2. Car Evaluation

   This dataset was derived from a simple hierarchical decision model for decision making. The model evaluates cars according to a concept structure considering 6 attributes: the overall price, maintenance price, number of doors, capacity, the size of luggage boot and estimated safety of the car. All attributes are enum attributes. This dataset has a total of 1728 instances and is split into 1210 training cases and 518 testing cases (70% training and 30% testing).

   In contrast to the Wilt dataset, The Car Evaluation dataset only has 1728 instances and all attributes are categorical and have enum values. Comparison between experiments performed on Car Evaluation dataset and Wilt dataset can demonstrate how algorithms behave differently on different types of dataset. Also, the difference in size can also demonstrate how size can influence the performance of classifiers.

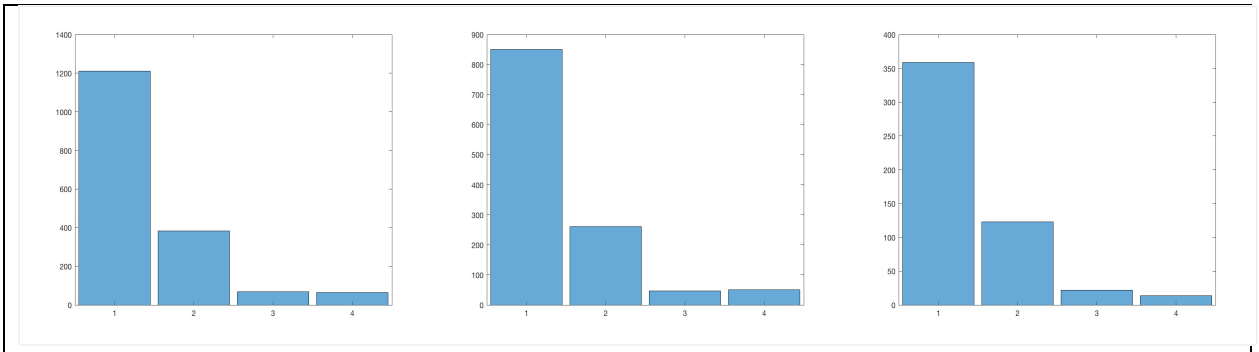
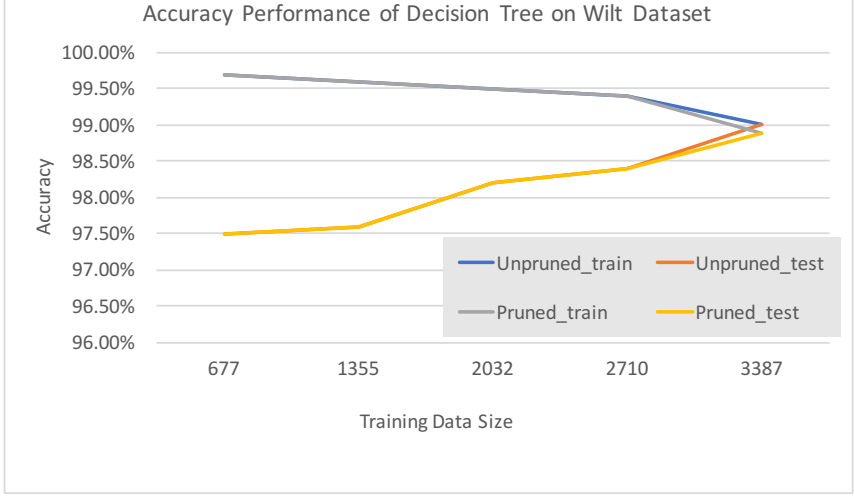


Figure 2 Histogram of Class Distribution of Car Evaluation dataset. Left: Original Dataset; Middle: Training Dataset; Right: Test Dataset.

Decision Tree

For decision tree, the J48 implementation of the C4.5 Decision tree is chosen. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits the rest of dataset into subsets that is abundant in one class. The splitting criteria used by this algorithm is the information gain (difference in entropy). The confidence factor is 0.25. The experiments were performed twice with shuffled training set. The mean of the results is reported in Figure 3.



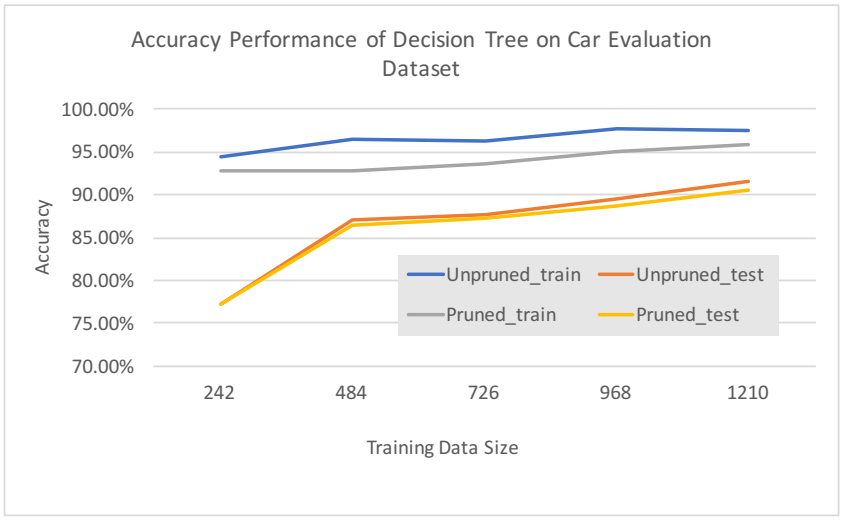


Figure 3 Performance of Pruned and UnPruned Decision Tree

As can be seen from Figure 3, in both datasets, there are minimal level of overfitting in pruned decision tree. From Table 1 and Table 2, we can see that after pruning, the size of the tree and the number of leaves decreased as expected, the accuracy of the pruned decision tree on test data does not change much. Another thing to note is that from the trend shown in Figure 3, we can show that in both datasets, while the size of training data increases, the accuracy of classifier on testing dataset increases. However, the increase in Wilt dataset, while the increase is

relatively small comparing with that in Car Evaluation dataset. Also, in Car Evaluation dataset, the accuracy of decision tree (both pruned and unpruned) on testing dataset increases in a linear relationship as the training data size increases. This implies that the sample in training data of Wilt dataset covers the domain of testing data. In other words, increasing dataset size in Wilt dataset should not increase the performance of decision tree. However, in Car Evaluation dataset, increasing dataset size will likely result in higher performance.

	Wilt				
IsPruned	Cross Validation Accuracy	Variance	Leaves	Tree Size	Training Time(s)
Yes	97.60%	1.116E-6	10	19	0.01
No	99.00%	2.512E-5	12	23	0.01

Table 1. Pruned and Unpruned Decision Trees for Wilt Dataset

	Car Evaluation				
IsPruned	Cross Validation Accuracy	Variance	Leaves	Tree Size	Training Time(s)
Yes	89.90%	2.424E-5	105	144	0.01
No	92.80%	2.351E-5	141	192	0.01

Table 2. Pruned and Unpruned Decision Trees for Car Evaluation Dataset

Boosting

AdaBoostM1 algorithm is used to boost the pruned and unpruned J48 Decision tree. Different number of iterations was tested using 10-fold cross validation and summarized in the following table. Since with boosting, more aggressive pruning can be afforded, I used confidence factor of 0.1 for the following experiment. As shown by Table 3, the optimized number of iteration for Wilt dataset is 15 and for Car Evaluation dataset is 10. The optimized number of iterations for each dataset is used to generate Figure 4.

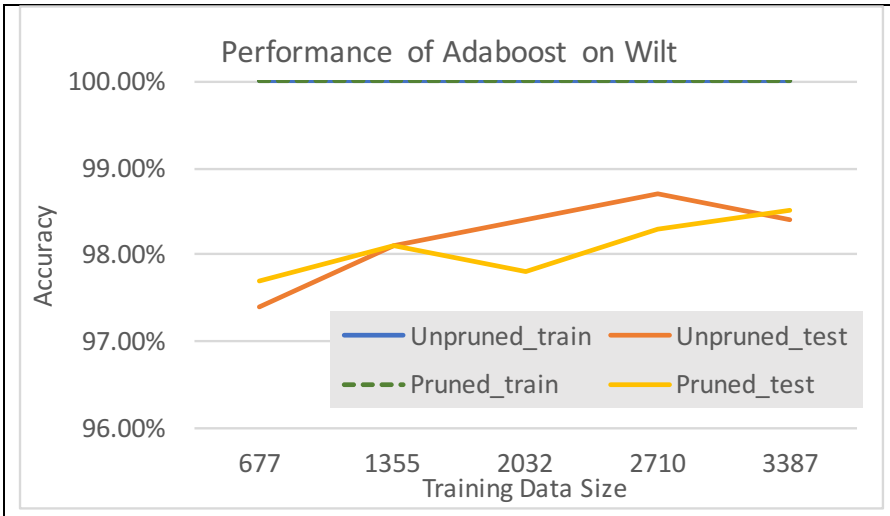
	Wilt		
Iterations	Pruned Tree CV Accuracy	Unpruned Tree CV Accuracy	Time Cost (s)
5	94%	93.70%	0.2
10	95.50%	95.00%	0.4
15	96.50%	95.40%	1.0
20	96%	95.30%	1.1
25	96.20%	95.40%	1.9
30	96.40%	95.50%	2.8

Table 3. Boosting of Pruned and Unpruned Tree with Different No. Iterations for Wilt Dataset

	Car Evaluation		
Iterations	Pruned Tree CV Accuracy	Unpruned Tree CV Accuracy	Time Cost (s)
5	97.90%	97.40%	1.2

10	98.03%	97.50%	1.7
15	98.01%	97.70%	2.1
20	97.90%	97.60%	3.4
25	97.90%	97.60%	4.9
30	97.90%	97.70%	5.6

Table 4. Boosting of Pruned and Unpruned Tree with Different No. Iterations for Car Evaluation Dataset



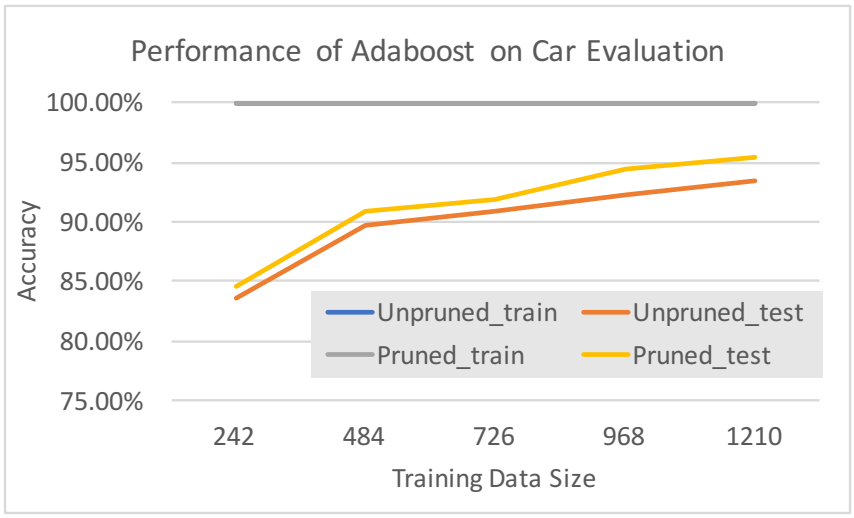


Figure 4 Performance of AdaboostM1 using prune and unpruned trees on Wilt and Car Evaluation Dataset

It is obvious from the Table 3 and 4 that when number of iteration increases, the precision increases first but then stabilized. From Figure 4, we can see that Adaboost generates high classification accuracy in both datasets. In general, Adaboost with pruned trees has better performance than Adaboost with unpruned trees. Also, Adaboost performs better than the best result of pruned trees. This result is reasonable since unpruned trees have some overfitting and

this is improved by pruning and Adaboost. However, the difference between pruned and unpruned trees is not quite noticeable. This also confirms with our previous conjecture that both datasets are separable in the domain of decision tress. Another thing to note is that there is more fluctuation in the performance curve on the Wilt dataset, which may be due to the nature of the data which is noisier than the Car Evaluation Dataset. This could also suggest that there are still overfitting in the model. Also, looking at the shape of the two curves shown in Figure 4, increase in training data size for Wilt could still have the potential to increase accuracy, but should not have a large effect on Car Evaluation since the slope of the curve is already smooth when using 100% training data size. In order to further improve the result, thorough investigation of parameters such as grid search of confidence factors and minimum samples split should be performed.

Neural Networks

For neural networks, I choose the feed forward Neural Networks. The parameters that are considered in the experiment are: activation function, number of hidden layers and number of neurons. The latter two determined the complexity of the neural network. Thus, if our dataset is very complex, then higher number of neurons and hidden layers should be used. However, if high number of neurons and hidden layers are used on relatively simple problem, this could also cause overfitting and bad performance. Activation function determines the behavior of the neurons in the model, which could also determine the performance based on the nature of the dataset. In total of 9 training functions were tested: Levenberg-Marquardt backpropagation (trainlm), BFGS quasi-Newton backpropagation (trainbfg), Resilient backpropagation (trainrp), Scaled conjugate gradient backpropagation (trainscg), Powell -Beale conjugate gradient backpropagation (traincgb), Fletcher-Powell conjugate gradient backpropagation (traincgf), Polak-Ribiere conjugate gradient backpropagation (traincgp), One step secant backpropagation (trainoss), and Gradient descent w/momentum & adaptive lr backpropagation (traindx). First, I compared the effect of different activation functions, change in number of hidden layers and neurons for both datasets using 10-fold cross validation of the two training datasets. Then both Wilt and Car Evaluation are run using the optimized parameters. Due to the large volume of the data, only results from the top two training functions are shown in Figure 5.

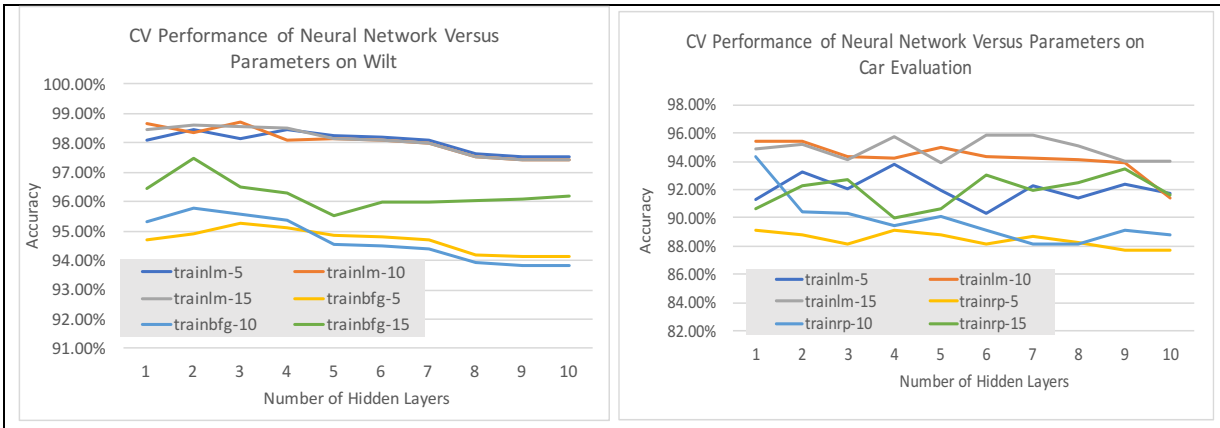


Figure 5 Cross-Validation Performance of Neural Network Versus Number of Hidden Layers, Training functions and Number of Neuron. The number after the name of training function is the number of neurons (i.e. trainlm-5 means that the number of neurons is 5 and training function is Levenberg-Marquardt backpropagation.

As can be seen from Figure 5, for both datasets, the best performing training function is trainlm. When using the same function, the cross-validation accuracy does not always increase as the number of neurons increases. In Figure 5-left, trainlm-5, trainlm-10 and trainlm15 almost have the same cross-validation accuracy. This means that the complexity that achieves by 5 neurons is enough for the model. The best performing parameters are summarized in Table 5 below.

	Number of Hidden Layers	Number of Neurons	Training function	Training Time
Wilt	3	10	trainlm	5.2s
Car Evaluation	7	15	trainlm	8.6s

Table 4. Best Performing parameters for Neural Networks

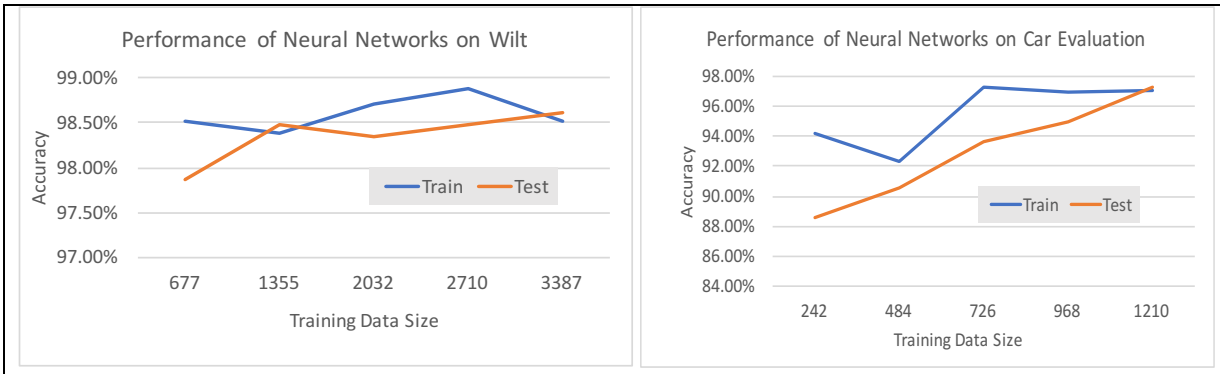


Figure 6 Performance of Neural Network Versus Training Data Size. Train – Accuracy tested on training dataset; Test – Accuracy tested on testing dataset.

The parameters in Table 4 was used to generate the result shown in Figure 6. As can be seen from Figure 6, the performance of Neural Networks on both datasets is very high. The training

curve of Wilt dataset is quite stable comparing to that of Car Evaluation, suggesting the model built on Car Evaluation dataset have larger variances than the model built on Wilt dataset. Although the performance on the training set is very similar to that on the testing set, the fluctuation in in the testing curve on Wilt dataset suggests that there is still some level of overfitting in the model, which complies with the behavior of Adaboost on Wilt dataset.

Support Vector Machine

Support Vector Machine (SVM) is a discriminative classifier defined by a separating hyperplane. Given the labeled training data, SVM outputs an optimal hyperplane that categorizes new examples. For SVM, we use the SMO function provided by Weka. Different kernel functions are used to investigate the behavior SVM.

Wilt		
Kernel	**CV Accuracy**	**Training Time(s)**
Poly Kernel	92.1%	1.9
RBF Kernel	92.1%	3.8
Puk	92.1%	2.77

Table 5. Performance of different kernels for Wilt Dataset

Car Evaluation		
Kernel	**CV Accuracy**	**Training Time(s)**
Poly Kernel	**92.8%**	**1.3**
RBF Kernel	**73.5%**	**1.4**
Puk	**87.6%**	**20.6**

Table 6. Performance of different kernels for Car Evaluation Dataset

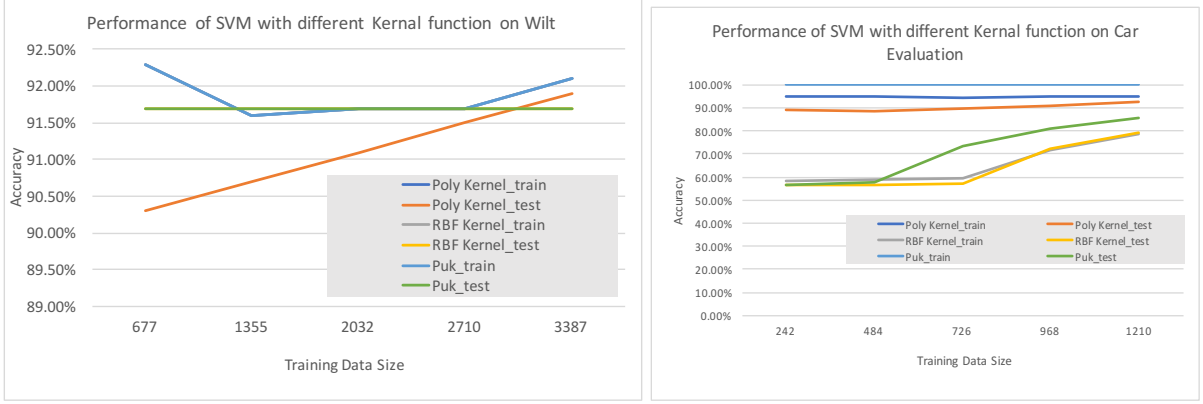


Figure 7 Performance of SVM versus Data Size

On the Wilt dataset, the accuracy of all kernel functions on training data are decent. The best performing kernel function on test data of Wilt dataset is the Poly kernel function, suggesting there exists a polynomial hyperplane that can separate the data space relatively well. However, the fluctuation in the training data suggests that there might be some level of overfitting. Other functions also have relatively similar performance, suggesting that Wilt is can be separate by different non-linear hyperplane. On the Car Evaluation dataset, the Poly Kernel function also performs the best, suggesting that there exists a polynomial hyperplane that can separate the data space relatively well. RBF and Puk were both performing terribly at small training size, but Puk improves relatively quick while RBF remains unoptimistic. The fact that RBF performs so terrible suggests that this dataset is not normally distributed since RBF is based on Gaussian distribution.

To further improve the performance, a grid search of optimized complexity parameter and gamma (kernel coefficient) should be performed to determine the best set of parameters for each kernel function. Other function such as linear kernel and sigmoid kernel should also be tested.

K-Nearest Neighbors

K-Nearest Neighbors (kNN) is a non-parametric method used for classification. kNN is also instance-based learning where the function is only approximated locally and all computation is deferred until classification. For each row (observation) in the testing dataset, the distance between its position and each row in training dataset is calculated.

For kNN, I used the IBK implementation from Weka. The parameters that investigated in the experiment is the k value. Figure 8 shows the performance curve as k increases.

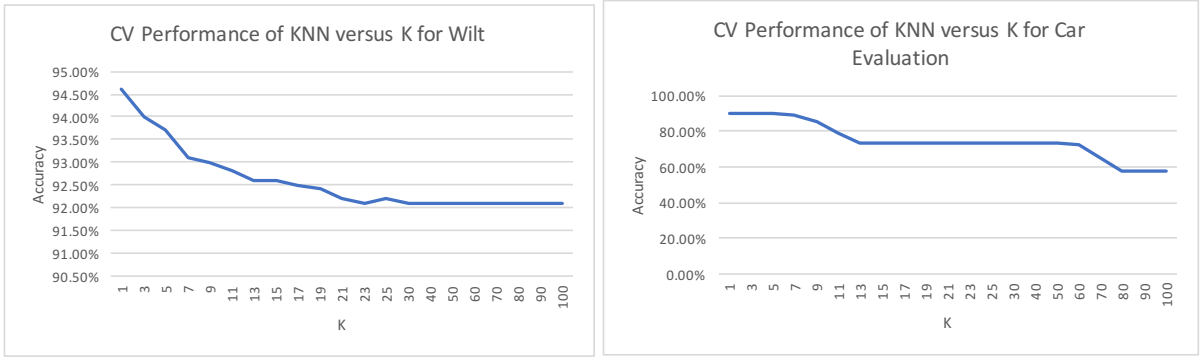


Figure 8 Performance of K-Nearest Neighbors versus K

As can be seen from Figure 8 above, the cross-validation performance decreases as k increases in both datasets. KNN is proficient in dealing with noise in the dataset as the number of neighbors, k increases. This is might be because there isn't a lot of noise in both dataset. When comparing the two curves above, we can see that the curve on the right is more much smoother than the curve on the right, suggesting that Car Evaluation dataset has much less noise than the Wilt dataset. When the k increases, more instances are included and introduce more confusion to the classifier. This observation complies with my conjecture at the beginning: since Wilt dataset is derived from live sensor, it should contain more noise than the dataset with enum values, which is the Car Evaluation dataset.

Thus, when the dataset is not corrupted by noise, smaller k values are better. For both datasets, I used k = 1 to generate Figure 9. However, when there is a lot of noise in the dataset, as k value increases, the accuracy will increase first, but will then decrease as k continues to increase.

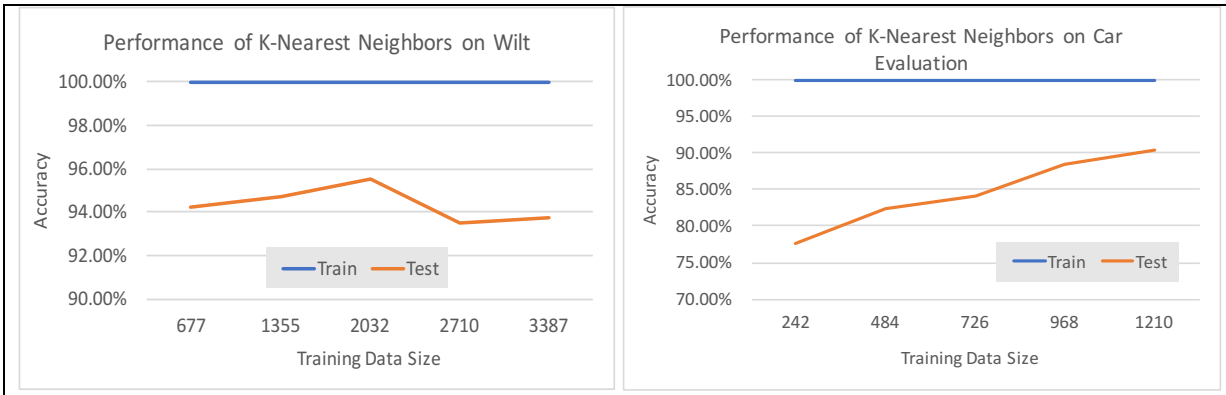


Figure 9 Performance of K-Nearest Neighbors versus Training Data Size

As shown by Figure 9, for Wilt dataset, the performance on training data is always perfect, but as the training data size increases, the testing set performance fluctuates, suggesting that there is

overfitting in the model. For Car Evaluation dataset, the performance on the training data is also always perfect. In contrast to the result from Wilt dataset, as training data size increases, the testing set performance also increases in a linear fashion, suggesting there is no overfitting in the model. To further improve the result, more data point is needed for Car Evaluation dataset.

Conclusion

From the analysis above, we can see that the performance of classifier is determined by the selection of hyper-parameters and the nature of dataset. In Wilt dataset, there is always some level of fluctuation in the performance curve that suggests overfitting while in Car Evaluation dataset, the performance curves are much smoother. This could also attribute to the nature of the datasets: Wilt contains more noise since the data is derived from sensors, while in Car Evaluation, since all the attributes are enums, there may exist some contracting samples that confuses the algorithms.

	Decision Tree		Adaboost		SVM		Neural Network		kNN	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
Wilt	98.90%	0.01	98.50%	0.1	91.90%	0.23	98.62%	5.2	93.70%	0.01
Car Evaluation	91.60%	0.01	95.40%	0.21	92.60%	0.02	97.30%	8.6	90.30%	0.01

Table 7. Summary of Performance of 5 supervised algorithms

As shown by Table 7 above, for Wilt dataset, the highest accuracy was achieved by Adaboost, which is 98.5%. Adaboost also has high efficiency since the training time is only 0.1 second for the Wilt dataset. For Car Evaluation dataset, the highest accuracy was achieved by Neural Network, which is 97.3%. However, the training time of Neural Network is much higher than all the other algorithms. Adaboost was able to achieve 95.4% accuracy with only 0.21 second of training time for Car Evaluation dataset. Thus, if in the future, the car selling company wants to apply the system to larger dataset, Adaboost may be a better option. If accuracy is more important than the time cost, then Neural Network is a better option.

Reference

[1] Johnson, B., Tateishi, R., Hoan, N., 2013. A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. International Journal of Remote Sensing, 34 (20), 6969-6982.

[2] M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988.

[3] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.