CS 4641 Supervised Learning Assignment

Jiahao Luo GT#903103970

1 Introduction

In this report, it mainly talks about two interesting dataset analyzed by implementing five basic supervised learning algorithms. Through the entire analyses for both datasets, it will cover on different interesting topics. Besides, the exploration is based on a popular suite of machine learning software, Weka.

2 Dataset Details

At first, the two datasets that I selected are downloaded from the Auto-Weka website. (http://www.cs.ubc.ca/labs/beta/Projects/autoweka/datasets/) They are the "Car Evaluation" and "Wine Quality" datasets. Here are some details about them.

Table 1. Dataset details of Car Evaluation

Title	Car Evaluation		
Number of instances	1728 (Nominal)	Missing	0
Number of attributes	6	Buying	Maint
		Doors	Persons
		Lug_boot	Safety

Table 2. Dataset details of Wine Quality

Title	Wine Quality		
Number of instances	6497 (Numeric)	Missing	0
Number of attributes	12	Fixed acidity	Volatile acidity
		Citric acid	Residual sugar
		Chlorides	Free sulfur dioxide
		Total sulfur dioxide	Density
		pH	Sulphates
		Alcohol	Quality

3 Why are these interesting datasets?

Before I choose these two datasets, I am curious about how to determine if a commodity is valuable or not, especially in the field that may require more experiences to make a decision, such as car and wine quality evaluations. When you are a layman in the car or wine field. How can you estimate the value of them? Which attribute of them should be regard as very important? Can machine learning help us to make a better decision even if we are not familiar with the knowledge about them? If we can figure out this, it not only provide a better decision for consumers (or auto dealers and wine collectors), but also help the producers to better produce the products.

With this question in my mind, I will carry out several experiments to explore the inner significance behind the data.

4 Experiment methodology

The entire experiments will include two parts. In the first part, it will focus on selecting the best algorithm model for specific datasets. Therefor the first thing I do is to optimize proper parameters which represent the model complexity for related algorithm models, such as confidence interval for decision tree, k value for kNN algorithm, kernel types for SVM algorithm and so on. Then by using the parameters selected from the first step, we can compare different algorithm on the same learning curve graph to judge which one performs better for specific dataset.

Before the experiments start, the datasets will be split up to two parts, one is the training set, the other one is the testing set. (Roughly, training set will occupy 70% of the entire data and testing set will be 30 %.) In order to get a more accurate measurements, I will use 10 folds cross-validation to train and test my training data only while remaining the test set unseen by the classifier.

4.1 Decision Tree

In the Weka software, I will use the J48 algorithm to perform a decision tree function. Here are the results.

Table 3. Parameter optimization of dataset using J48 algorithm

Car Evaluation

Prune state	Confidence	# of leaves	Tree size	Training error rate %
pruned	0.125	67	93	12.23
pruned	0.25	84	117	10.41
pruned	<u>0.5</u>	119	164	9.17
unpruned	/	141	194	7.85

Wine Quality

Prune state	Confidence	# of leaves	Tree size	Training error rate %
pruned	0.125	413	825	44.27
pruned	0.25	520	1039	43.45
pruned	<u>0.5</u>	547	1093	43.45
unpruned	/	576	1151	43.39

According to the tables show above, we can easily tell from the table show above, the proper confidence interval for Car Evaluation and Wine Quality are both 0.5 (underscored) considering the training error rate and. Then we can use these parameter to the algorithm to train the data and test their performance.

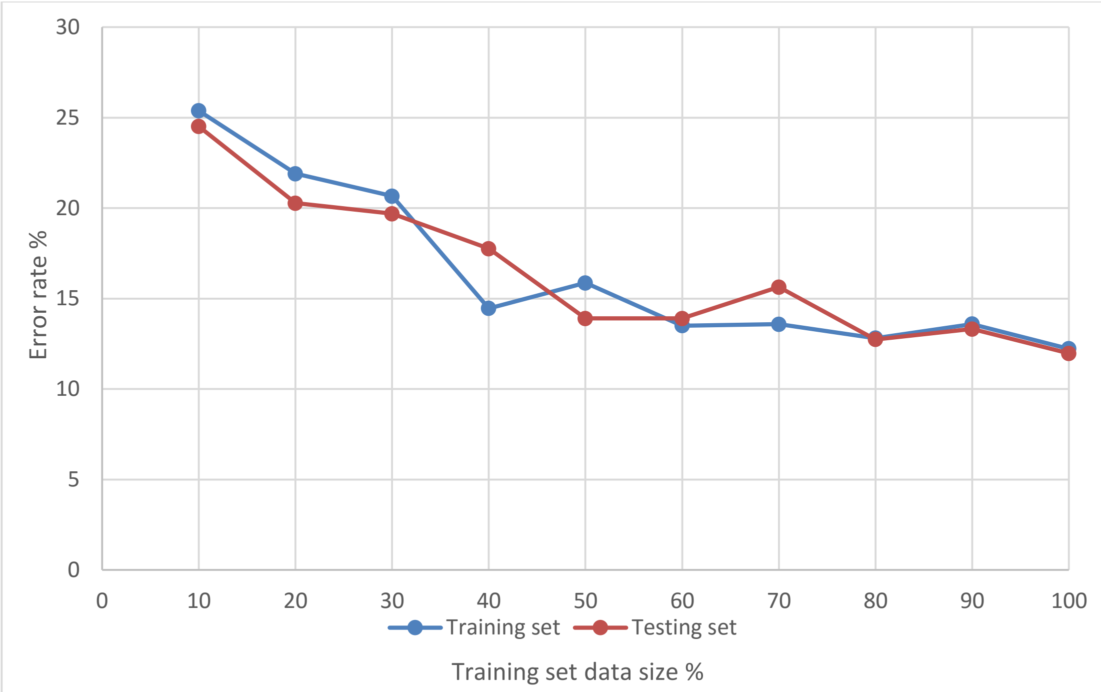

Figure 1. Learning curve for Car Evaluation dataset using J48 algorithm

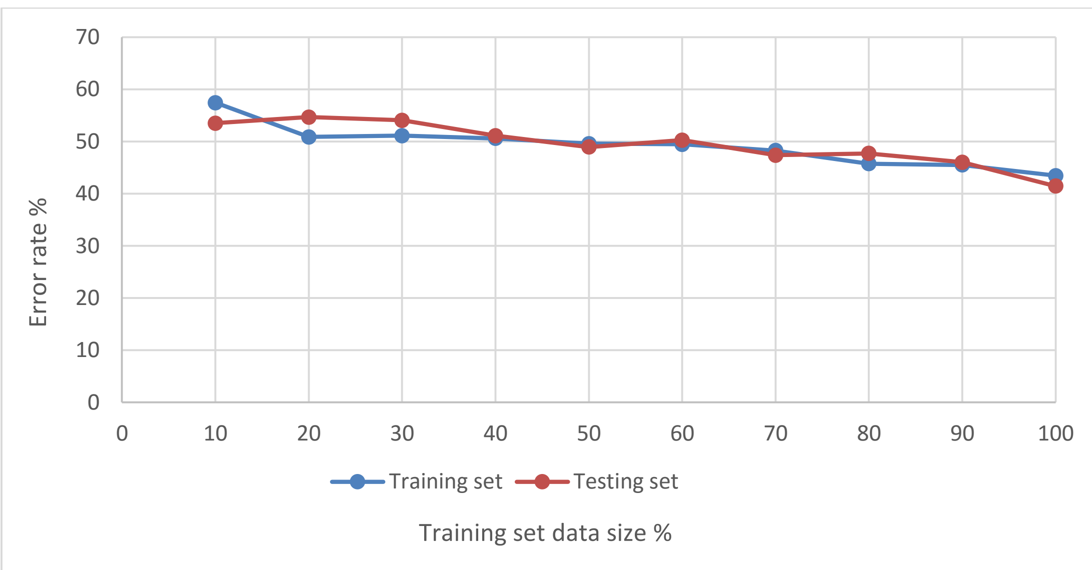

Figure 2. Learning curve for Wine Quality dataset using J48 algorithm

Analysis:

There are several features of note that are evident in the results for decision trees. First, the pruning method helps to decrease the training error about 1~5% for the Car Evaluation dataset, however, it doesn't help a lot to the Wine Quality dataset whose training rate just slightly decreases. Pruning can stop the growth of the decision tree so that it won't be overfitting.

Second, as we can see decision trees perform better in the Car Evaluation case the Wine Quality. In my opinion, it may result from the number of class. Since there are 4 classes for the first one dataset, and 11 classes. Actually, we can try to group the 11 classes to 5 or less classes so that it would be easier for the classifier to perform a better result. Besides, the number of instances of Wine Quality is much bigger than the other one which can also lead to the prediction difference.

4.2 Neural Networks

In the Weka software, I will use the MultilayerPerceptron algorithm to perform a neural networks.

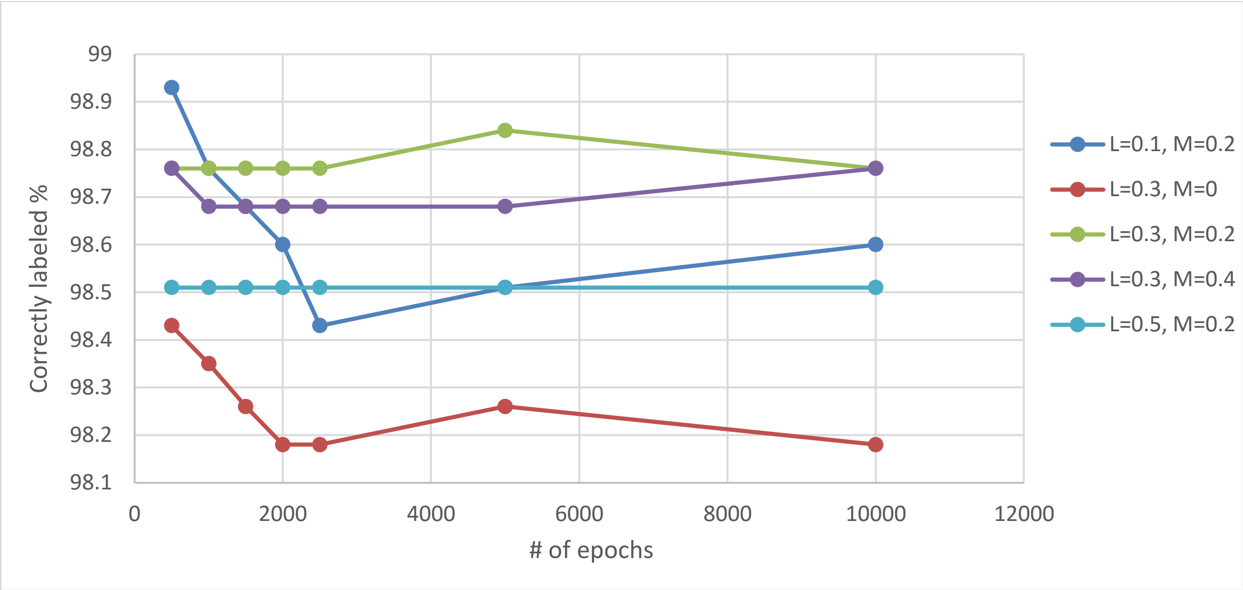

Figure 3. MultilayerPerceptron parameter optimization (Car Evaluation)

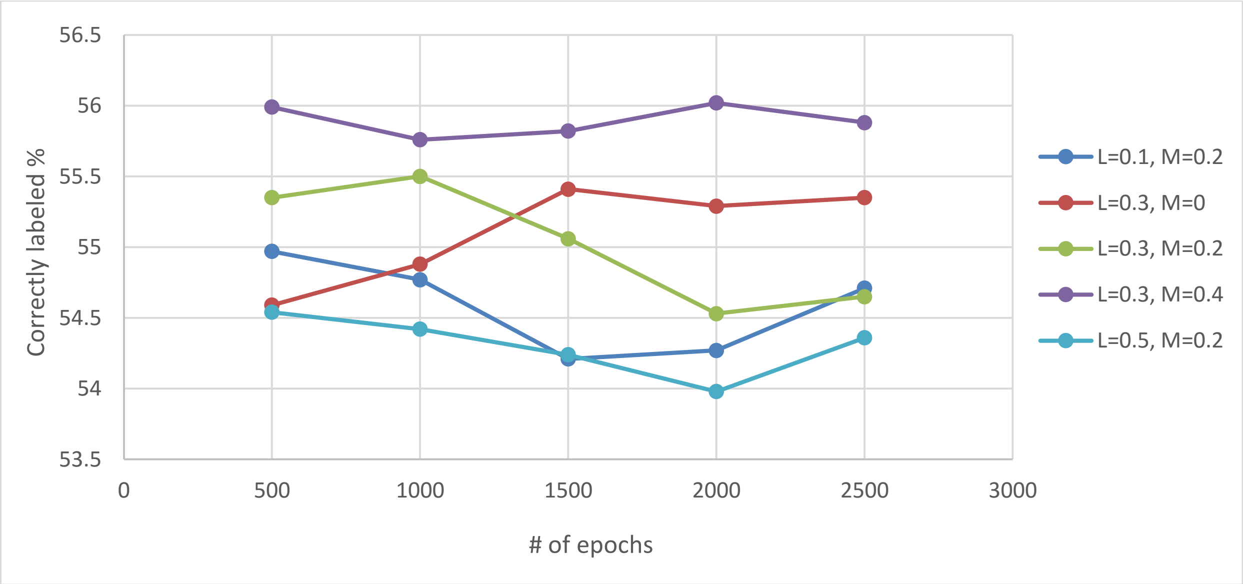

Figure 4. MultilayerPerceptron parameter optimization (Wine Quality)

According to the figures, through comparing different learning rate and momentum combination, we can get the different correctly labeled percentage. By looking at the plot above, when L = 0.3, M=0.2 for Car Evaluation and when L=0.3, W=0.4 for Wine Quality, the MultilayerPerceptron algorithm can perform better. Therefore, I will use these parameters for the later parts.

Figure 5. Learning curve for Car Evaluation dataset using MultilayerPerceptron algorithm

Figure 6. Learning curve for Wine Quality dataset using MultilayerPerceptron algorithm

Analysis:

For the neural network algorithm, there are several interesting parts from the result we should notice. At first, neural network can help decrease the error rate rapidly for both cases along with the increment of data size. However, neural network algorithm is the most time-consuming one among those algorithms. When I was optimizing the parameter for neural network algorithm, it took me several hours to perform the result, especially when number of epochs reached to 3000. So when the goal we are chasing is sensitive to the time or unaffordable to time consuming, then we need to think about it if we really want to use this algorithm to predict.

4.3 Boosting

In the Weka software, I will use the AdaBoostM1 algorithm to perform a decision tree J48 function. Find an optimal value for number of iterations and confidence interval from below, then record these parameter for the latter part to acquire the testing set results.

Table 4. Parameter optimization of dataset using AdaBoostM1 algorithm

Car Evaluation

Prune state	boost iteration	confidence	training accurate rate
pruned	10	0.125	94.46
pruned	10	0.25	94.63
pruned	10	0.5	94.38
unpruned	10	/	93.88
pruned	20	0.125	95.7
pruned	20	0.25	95.12
pruned	20	0.5	94.88
unpruned	20	/	94.88
pruned	40	0.125	96.2
pruned	40	0.25	95.37
pruned	40	0.5	94.79
unpruned	40	/	95.04

Wine Quality

Prune state	Boost iteration	Confidence	Training accurate rate
pruned	10	0.125	61.91
pruned	10	0.25	63.49
pruned	10	0.5	61.39
unpruned	10	/	63.98
pruned	20	0.125	63.49
pruned	20	0.25	63.93
pruned	20	0.5	63.11
unpruned	20	/	64.48
pruned	40	0.125	64.95

pruned	40	0.25	64.68
pruned	40	0.5	64.33
unpruned	40	/	65.03

From the table above, I choose the prune state as “pruned”, the boost iteration as 40 and the confidence as 0.125 for both dataset.

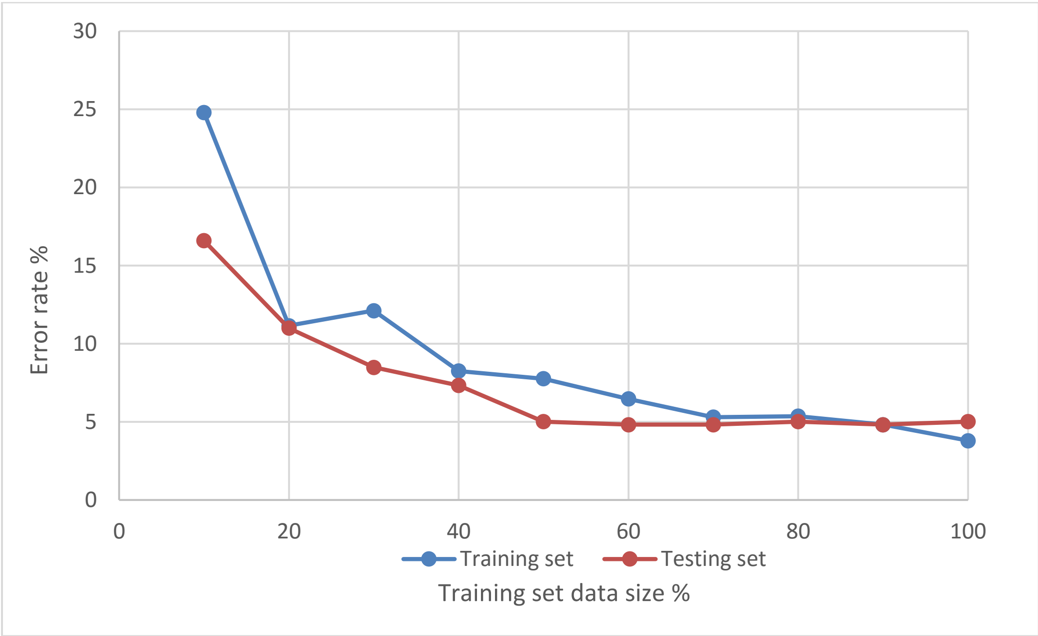

Figure 7. Learning curve for Car Evaluation dataset using AdaBoostM1 algorithm

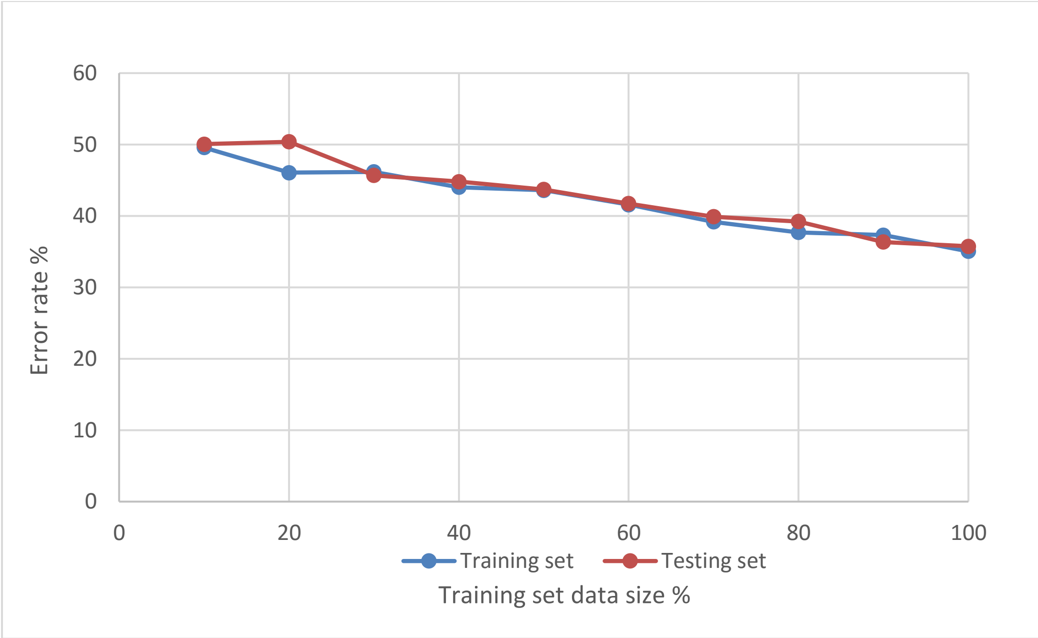

Figure 8. Learning curve for Wine Quality dataset using AdaBoostM1 algorithm

Analysis:

For the boosting algorithm, there are several interesting parts from the result we should notice. At first, boosting can help decrease the error rate rapidly for both cases. Also it gives us a similar result of decision tree which is high error rate for Wine Quality and relatively low error rate for Car Evaluation. I think the reason is the same as stated in the decision tree part. Because the large number of classes and the highly clustered data may cause it hard to split on the Wine Quality case. Second, for the car evaluation part, the error rate is higher than the testing data set initially, that means the algorithm has a high bias between the function and dataset.

4.4 k-NN

In the Weka software, I will use the IBK algorithm to perform a k near neighbor function.

Car Evaluation error rate

k	Manhattan distance	Euclidean distance	Chebyshev distance
1	9.4	9.4	30.33
3	9.4	9.4	30.33
5	9.45	9.45	30.33
7	10.11	10.11	30.33
9	12.76	12.76	30.33
11	17.91	17.91	30.33
13	22.4	22.4	30.33
15	23.55	23.55	30.33
17	23.58	23.58	30.33
19	23.58	23.58	30.33

Wine Quality error rate

k	Manhattan distance	Euclidean distance	Chebyshev distance
1	38.61	38.9	40.16
3	46.22	46.19	47.3
5	45.7	46.78	47.36
7	46.05	46.02	46.46
9	46.28	45.9	46.81
11	45.49	45.82	46.72
13	45.47	45.2	46.43
15	45.23	44.91	46.34
17	45.38	45.2	46.54
19	45.38	45.09	46.19

From the tables above, I choose parameter k as 1 and Manhattan distance for both cases.

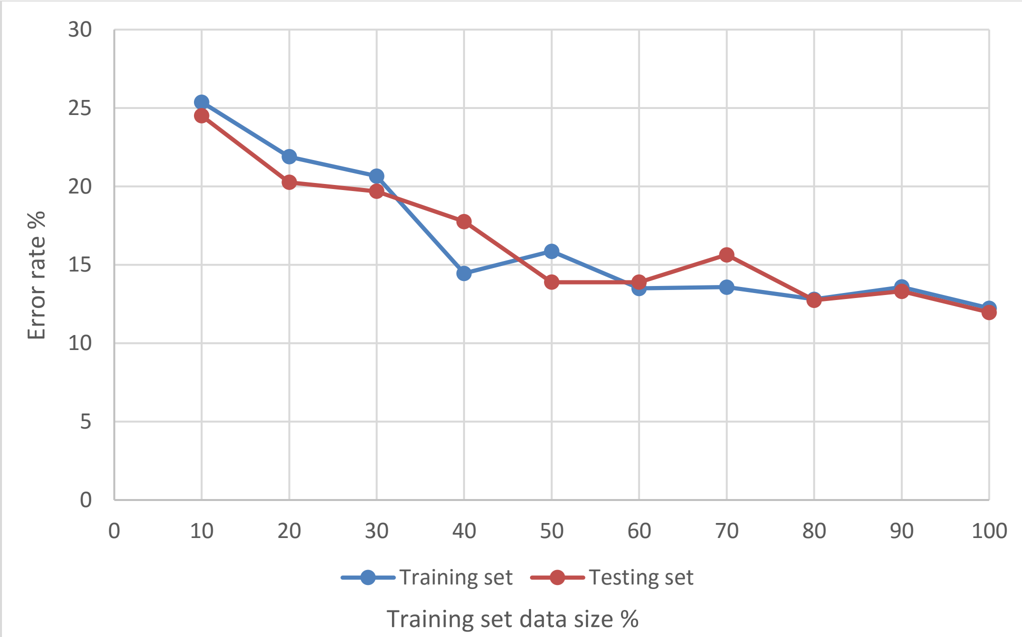

Figure 9. Learning curve for Car Evaluation dataset using IBK algorithm

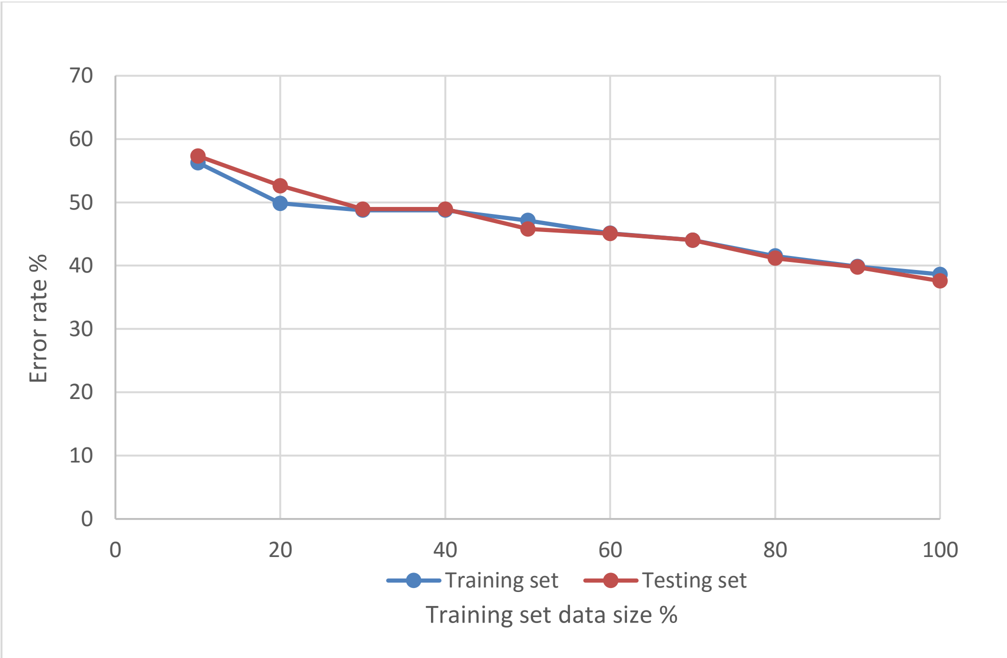

Figure 10. Learning curve for Wine Quality dataset using IBK algorithm

Analysis:

As to the k-NN algorithm, it is obvious that the error rate for training and testing set are decreasing as the training set data size increasing. However, in both of dataset, the error rate of training set and testing set are very close. From the tables above, we can tell Manhattan distance and k = 1 would be better to fit the dataset.

4.5 Support Vector Machines

In the Weka software, I will use the SMO algorithm to perform a support vector machines function.

Kernel	Car Evaluation		Wine Quality	
	Parameter	Training accurate rate	Parameter	Training accurate rate
Poly	1	91.98	<u>1</u>	<u>52.87</u>
	2	99.09	2	52.61
	<u>3</u>	<u>99.75</u>	3	52.46
RBF	0.01	77.93	0.01	45.12
	0.5	96.86	0.5	47.19
	1	87.93	1	52.81

From the tables above, I select the poly kernel and 3 as its exponent for Car Evaluation, and poly kernel and 1 as its exponent for Wine Quality since those configuration can perform better.

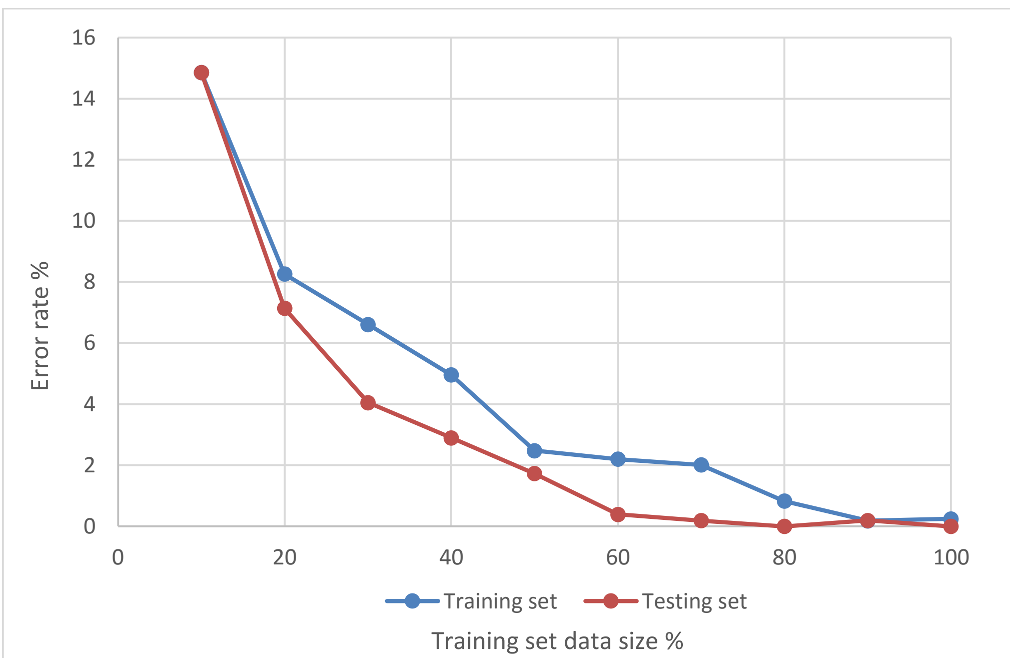

Figure 11. Learning curve for Car Evaluation dataset using IBK algorithm

Figure 12. Learning curve for Wine Quality dataset using IBK algorithm

Analysis:

As to the SVM algorithm, it is obvious that the error rate for training and testing set are decreasing as the training set data size increasing in the Car Evaluation case. However, in the Wine Quality case, the error rate of testing set go up at first and then remain a relatively flat platform while the training set keep going down. There is high bias when the Wine Quality dataset size is small and variance between the training set and testing set is increasing when dataset size is bigger which is pretty interesting here. Although I am not quite sure about this, but I guess maybe the parameters selection is not good enough, I should expand the range for proper parameter.

5 Evaluation of classifier on different datasets

Table 1. Evaluation of classifier on Car Evaluation dataset with Cross-validation.

Classifier	**Time taken to build model**	**Error rate**	**Correctly classified instances**	**Incorrectly classified instances**	**Mean absolute error**	**Root mean squared error**
Trees - J48	0	11.97	456	62	0.0735	0.2132
Functions - MultilayerPerceptron	3.44	0.77	514	4	0.0093	0.0547
AdaBoost	0.11	5.02	492	26	0.0243	0.1528
Functions - SMO	0.79	0	518	0	0.2503	0.3119

Lazy - IBK	0	7.34	480	38	0.1084	0.1936

Table 2. Evaluation of classifier on Wine Quality dataset with Cross-validation.

Classifier	**Time taken to build model**	**Error rate**	**Correctly classified instances**	**Incorrectly classified instances**	**Mean absolute error**	**Root mean squared error**
Trees - J48	0.15	41.46	860	609	0.0808	0.2551
Functions - MultilayerPerceptron	11.1	46.56	785	684	0.1026	0.236
AdaBoost	12.6	35.74	944	525	0.0647	0.2493
Functions - SMO	0.39	49.9	736	733	0.1474	0.2656
Lazy - IBK	0	37.58	917	552	0.0686	0.261

6 Appendix

A1. Error rate of training and testing set using J48 algorithm

	Car Evaluation dataset		Wine Quality	
Training dataset split size (%)	Training set error rate (%)	Testing set error rate (%)	Training set error rate (%)	Testing set error rate (%)
10	25.38	24.51	57.43	53.51
20	21.9	20.27	50.87	54.66
30	20.66	19.69	51.12	54.05
40	14.46	17.76	50.58	51.12
50	15.87	13.9	49.59	48.94
60	13.5	13.9	49.44	50.24
70	13.58	15.64	48.25	47.38
80	12.81	12.74	45.75	47.72
90	13.59	13.32	45.5	46.02
100	12.23	11.97	43.45	41.46

A2. Parameter optimization of dataset using MultilayerPerceptron algorithm (Car Evaluation)

Learning rate	Momentum	500	1000	1500	2000	2500	5000	10000
0.1	0.2	98.93	98.76	98.68	98.6	98.43	98.51	98.6
0.3	0	98.43	98.35	98.26	98.18	98.18	98.26	98.18
0.3	0.2	98.76	98.76	98.76	98.76	98.76	98.84	98.76
0.3	0.4	98.76	98.68	98.68	98.68	98.68	98.68	98.76
0.5	0.2	98.51	98.51	98.51	98.51	98.51	98.51	98.51

A3. Parameter optimization of dataset using MultilayerPerceptron algorithm (Wine Quality)

Learning rate	Momentum	500	1000	1500	2000	2500	5000	10000
0.1	0.2	54.97	54.77	54.21	54.27	54.71	55.5	55.96
0.3	0	54.59	54.88	55.41	55.29	55.35	-	-
0.3	0.2	55.35	55.5	55.06	54.53	54.65	-	-
<u>0.3</u>	<u>0.4</u>	<u>55.99</u>	55.76	55.82	56.02	55.88	-	-
0.5	0.2	54.54	54.42	54.24	53.98	54.36	-	-