1a: Due to the error changes among those five proportion of training data , the learning curves can be observed. For most of the datasets, errors are reduced in both IBK and J48 algorithm by enhancing the proportion of data sets . It has been a sharp error reduction from 10% to 25% sample size percent, however, slightly diminishing errors are observed among 25%, 50%,75% and 100%. The cross validation error also decreases as data increases. With the increase of data, cross validation of sets are getting close to the training error rate. But there is a variation between datasets and algorithms. Let us look at the dataset auto, audiology and microarray. The cross validation remains high and the error reduces slightly. This means in the high validation environment, if the algorithm is suffering from high validation, more data could help to solve the problem. In dataset microarray , the error and cross validation both keep stable , and more data does not seem to help. Algorithm seems to suffer high bias problems . There are also differences between 2 algorithms, IBK in this problem only find the nearest element and this algorithm do not make any model. On the other hand, J48 build an decision tree to ignore the numbers of parameters ( eg. K in KNN )

1b: error = sum( (err0-erri)/err0 \* 100)/8

the 2x2 table:

Algorithm after 10% after 100%

IBK 35.2% 57.6%

J48 49.6% 71.9%

It is larger than I expected after seeing 10% of the training data, and it is smaller than I expected after seeing 100% of the data. First of all, due to the small amount of the data , I expected the learning curve would produce large errors and it is not easy to correct, also due to the amount of data, it is not easy to predict . When the data is large , i think machine could have more characteristic value to predict and 57.9% is smaller than I expected

The effect is more pronounced for J48, we calculate the average percentage reduction for all data

Algorithm after 10% after 25% after 50% after 75% after100

IBK 35.2% 44.6% 50.5% 54.6% 57.6%

J48 49.6% 59.9% 65.7% 65.9% 71.9%

Due to the calculation, we find that in any proportion of data J48 are more effective than IBK. J48 produce an decision tree to predict and IBK only use lazy studying so that J48 is better choice

2a: Yes, the tree learning has managed to avoid overfitting. Due to the result in q2.out,

when the noise is added to 20%, the error rate does not change a lot. When the noise is added to 50% with default parameters, the error rate grows lower than 50 %, so j48 algorithm uses subtree raising and subtree replacement strategy to avoid overfitting, by calculating the old and predicted expected classification error rate. If pruning leads to error increase , J48 will give up . After series of pruning , the decision tree with smallest classification error rate is retained. So J48 is managing to avoid overfitting durning the execution at low level .

2b: Yes. Compared with the default J48 and CVParameters J48 with 50% noise , we can observe that the error rate with C 2 30,5 ,which is 2 to 30 with 5 steps , is obviously lower than the default J48 so that it make efforts to the overfitting avoidance.

3a) : log : reduce cross-validation error

square: increase cross-validation error

cube : cross-validation error

3b) :I did the log ,square ,and cube transformation. Due to the result in Q3.out ,log operation narrow the mean absolute error as well as root mean square error. The log operation narrow the degree of dispersion of elements and lower the average value of real elements , so log operation reduce the cross-validation error.

Square and cube transformation greatly enlarge the actual value of the elements, and enlarge the average value of the real elemtn , it also expand the degree of dispersion of element so that it increase the cross-validation error. We can get the result from the Q3 that mean absolute error , root square error as well as relative absolute error has obviously increase.

4: First of all, there are 1000 attributes to be executed ,then each elements in the will obey the NaiveBayesMultinomial to classify , the machine calculate the probability of each element appeared in the different class and classify the element into the largest probability class, then count the correct rate of the classify instances . We use them as the basic standard of the result. Then we should compare the difference between J48 and NaiveByes from small attributes to large attributes, so we first start to use 10 attributes, then we selected decision trees to classify information. J48 algorithm uses information entropy and information gain rate as standard to classify the information, and uses subtree raising and subtree replacement strategy to build the tree. Then we back to NaiveBayes to deal with 10 attributes. We calculate the rate as follows

Attributes : 10 100 500

J48 23.6779 31.2823 32.1869

NB 28.7276 52.8728 62.8728

Among those three situations , we can observe that NaïveBayes classification are generally more effective than J48 tree method. and with the data amount increasing , the performance between two methods are different .The increasing rate of NaïveBayes are quicker than J48 algorithm. We can configure using the NaïveBayes is more suitable than J48 when each element are independent.