Analysis of various classifier implementation

HKUST 19/20 Spring term (Comp4331)

Prepared by: Yu, Kijun (20319196)

Decision Tree Classifier

Decision tree algorithm is often used to predict the classification of given test data set. Before testing the data, the tree is initially studied with other data set with given classification and the tree is built according to the probability of each attribute which is later used to produce information of a node (or aspects of attribute: either discrete or continuous) and calculate the gain. There are various types of decision tree algorithm, in which this report will discuss about ID3 algorithm and C4.5 algorithm with example and tested program. The tests in this assignment were carried out with spyder3 in Windows 10, the CPU used was Intel® Core™ i7-8700 @ 3.20GHz.

I. ID3 Decision Tree

The attribute with the highest gain is selected for ID3 decision tree's initial node. Each node will be classified with a single attribute and the dataset split in the node will contain same criteria for given attribute. In this assignment, there were 8 attribute which was provided to classify the nursery value.

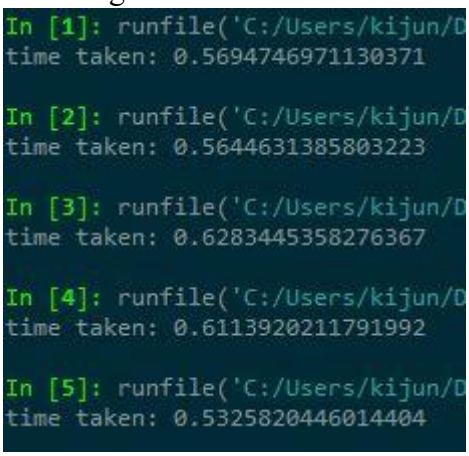
The table of given attributes and their values are provided as table below.

Parents	**Has_nurs**	**Form**	**Children**	**Housing**	**Finance**	**Social**	**Health**
Usual	Proper	Complete	1	Convenient	Convenient	Nonprob	Recommended
Pretentious	Less_proper	Completed	2	Less_conv	Inconv	Slightly_prob	Priority
Great_pret	Improper	Incomplete	3	critical		problematic	Not_recom
	Critical	Foster	more				
	Very_critical						

The nursery class had 5 classifications: not_recom, recommended, very_recom, priority, spec_priority.

Given data set had various combinations of each attributes and classes, the data set contained 12950 entries which were processed to calculate the gain and decide to build decision tree. There are several conditions for decision tree node split to stop, which are at condition where there are no more attributes to provide decision, at when there are no more entries to be split, or if all the entries in given node belongs to same class.

The image below shows the runtime of building tree and testing 10 samples with ID3 algorithm.



II. C4.5

C4.5 algorithm uses similar logic to ID3 algorithm the only difference between them is calculation of gain. The gain calculation in ID3 algorithm are further divided with split node info, the entropy of given attributes distribution. And with this info the tree is built in same manner with ID3 algorithm.

To build C4.5 algorithm decision tree, the algorithm was adapted from ID3 algorithm with small modification and similarly the test was again carried out with same test data set. The result time of tree building and testing 10 sample data are shown below.

```
In [9]: runfile('C:/Users/kijun/De
time taken: 0.5565071105957031

In [10]: runfile('C:/Users/kijun/D
time taken: 0.5326023101806641

In [11]: runfile('C:/Users/kijun/D
time taken: 0.5535473823547363

In [12]: runfile('C:/Users/kijun/D
time taken: 0.6303129196166992

In [13]: runfile('C:/Users/kijun/D
time taken: 0.592414140701294
```

As it is visible in these two results, the time taken for both algorithms are with very minor difference which are neglectable. This is because two algorithms adapt with same mechanism and only difference is the final info calculation. Otherwise all other steps are same for these two methods. The result of both algorithms was also the same which is viewed below.

```
result of ID3
test 01 result not_recom
test 02 result spec_prior
test 03 result priority
test 04 result priority
test 05 result priority
test 06 result priority
test 07 result not_recom
test 08 result spec_prior
test 09 result priority
test 10 result priority
```

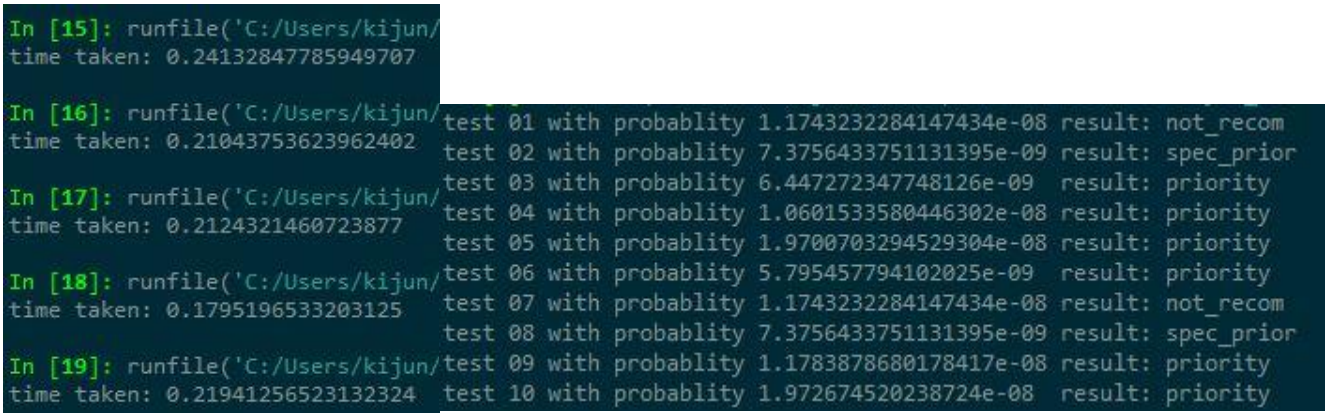
```
result of C4.5
test 01 result not_recom
test 02 result spec_prior
test 03 result priority
test 04 result priority
test 05 result priority
test 06 result priority
test 07 result not_recom
test 08 result spec_prior
test 09 result priority
test 10 result priority
```

Naïve Bayes Classifier

Naïve Bayes classifier is a type of classifier that uses logic adapted from Bayes classifier. Bayesian classifier performs probabilistic prediction. Naïve Bayesian classifier, compared to decision tree algorithm, is expected to have comparable performance. Each probability of attribute labels is calculated according to there class label and frequency in entire data set. The initial probability solely depends on single attribute. i.e.) for this assignment, attribute Parents, would have probability of 1 if all aspects are added together. And would be divided into several classes of nursery label. After processing each of probability, the probability is multiplied together to find out class label of unknown test data. To carry out, many data set are needed to be studied in advance to provide class, hence the computational cost for this mechanism would be significant.

To avoid 0 probability, each aspect is added with small number to decide probability of attributes-class pair with 0 existence. This is called Laplacian correction.

The time taken for naïve Bayesian classifier is given below along with the expected output of 10 test data set, which are same to the test set used above.



The time taken has been reduced for this classifier as the data set is only studied once to provide probability, however for the classifiers above, there is need of repeated checking of attribute aspects after splitting a node according to the label of attribute. i.e.) parents, has_nurs etc. for this assignment.

However, the results are same to the previous classifiers. Hence it can be concluded Bayesian classifier is both almost accurate as decision classifiers and fast compared to the decision tree algorithm. Yet, it could be less accurate in smaller data set as it is built by assumption and each probability is independent from other attribute's probability.

Hence, according to the data set and need of accuracy, the programmer needs to decide which algorithm to adapt, to process and test given data set.