CSCI5523 Project3 Report

Daihui DOU dou00005@umn.edu 5514178

1 Implementation

The implementation of decision tree classifier is similar to the algorithm in the text book:

Algorithm 4.1 A skeleton decision tree induction algorithm.

```
TreeGrowth (E, F)
 1: if stopping_cond(E,F) = true then
 2:    leaf = createNode().
 3:    leaf.label = Classify(E).
 4:    return leaf.
 5: else
 6:    root = createNode().
 7:    root.test_cond = find_best_split(E, F).
 8:    let V = {v|v is a possible outcome of root.test_cond }.
 9:    for each v ∈ V do
10:       E_v = {e | root.test_cond(e) = v and e ∈ E}.
11:       child = TreeGrowth(E_v, F).
12:       add child as descendent of root and label the edge (root → child) as v.
13:    end for
14: end if
15: return root.
```

1. The createNode() function extends the decision tree by creating a new node.
A non-leaf node has four attributes: left, right, attribute, value. 'left' and 'right' are its left child and right child. 'attribute' is the attribute index to split. 'value' is the split value.
A leaf node has one attribute: label, which is the class label that data points of this leaf belong to.

2. The find_best_split() function determines which attribute and split value should be selected as the test condition for splitting the training records. Data points less than the split value goes to left child node, and data points no less than the split value goes to right child node.

3. The Classify() function determines the class label to be assigned to a leaf node. For each leaf node t, it is the label that has the most number of data points.

4. The stopping_cond() function is used to terminate the tree-growing process by testing whether all the records have either the same class label, or whether all attributes are used, or whether the number of records is below minimum frequency.

2 Running

To run the programs, please see README for details.
Training rep1 data takes about 2 hours.
Training rep2 data takes about 6 minutes.
I attach model files in submission for convenience to test.

3 Result

3.1 rep1 data

1. minfreq = 1

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep1_1_pred.csv
Total number:  10000
The confusion matrix:
[[ 902    0    8    5    5   18   15    4   10   13]
 [   2 1078   10    9    2    5    5    5   18    1]
 [  20    9  866   35    9   10   13   28   31   11]
 [  10    2   28  858   12   33    3    8   35   21]
 [   3    2   13    9  849   10   14   11   23   48]
 [  15    8    2   29   11  756   13   10   26   22]
 [  16    7   12    9   23   15  846    2   20    8]
 [   3    7   15   17    5   10    2  933    7   29]
 [   9    3   20   35   24   27   17   10  792   37]
 [  17    4   11   11   34   11    5   15   26  875]]
Accuracy:  0.8755
```

2. minfreq = 5

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep1_5_pred.csv
Total number:  10000
The confusion matrix:
[[ 912    1    8    7    6   13   12    5    8    8]
 [   0 1089   11    8    3    5    5    3   10    1]
 [  17   20  881   32    7    9   12   22   24    8]
 [   8    7   31  868   10   31    9    6   24   16]
 [   6    3   10   10  852    9   15   11   23   43]
 [  16    9    4   43    6  749   18    8   21   18]
 [  15    8   13    9   25   13  845    2   22    6]
 [   4   13   24   20    5    4    2  919   11   26]
 [  18    5   23   43   22   27   17   10  786   23]
 [  17    5   10   17   40   13    4   20   25  858]]
Accuracy:  0.8759
```

3. minfreq = 10

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep1_10_pred.csv
Total number:  10000
The confusion matrix:
[[ 914    0   10    4    6   14   11    5    8    8]
 [   3 1092    8    9    2    4    7    2    7    1]
 [  12   17  885   33   10   10   12   21   26    6]
 [   8    6   33  874    6   30    6    6   25   16]
 [   7    6    8    8  858   13   13    9   23   37]
 [  16    7    5   43    7  749   21    7   21   16]
 [  15    7   12    9   24   19  846    1   18    7]
 [   5    7   27   18    5    3    4  927   12   20]
 [  14    4   22   41   20   30   18   10  791   24]
 [  18    7   12   16   39   13    6   18   28  852]]
Accuracy:  0.8788
```

4. minfreq = 20

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep1_20_pred.csv
Total number:  10000
The confusion matrix:
[[ 913    1   12    5    5   10   12    8    8    6]
 [   3 1090    9    8    3    7    5    2    7    1]
 [  21   14  867   34   12   13   12   29   21    9]
 [   7    6   28  871    8   39    3   10   22   16]
 [   7    5    9    7  861   12   12   12   24   33]
 [  17    8    4   46    7  742   20    8   22   18]
 [  16    5   14   11   22   25  835    1   22    7]
 [   5    8   28   20    5    5    4  926    8   19]
 [  15    6   24   49   24   32   14    7  785   18]
 [  16    7   10   16   47   12    5   16   25  855]]
Accuracy:  0.8745
```

3.2 rep1 data

1. minfreq = 1

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep2_1_pred.csv
Total number:  10000
The confusion matrix:
[[ 834    2   24   18    9   30   26    6   27    4]
 [   2 1093    5    4    1    5    8    2   15    0]
 [  20    4  835   42    9   15   19   16   63    9]
 [  21    4   29  832    4   44    5   12   49   10]
 [   8    7   12    8  781    9   20   18   20   99]
 [  33    5   17   53   13  639   24    5   81   22]
 [  22    5   19    3   20   28  842    4    9    6]
 [   9   14   25   14   26    7    5  845   29   54]
 [  22    5   28   44   24   66   14   17  725   29]
 [   7   11    8   14   72   22    6   45   22  802]]
Accuracy:  0.8228
```

2. minfreq = 5

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep2_5_pred.csv
Total number:  10000
The confusion matrix:
[[ 845    2   24   18    6   24   25    6   26    4]
 [   4 1094    4    4    1    4    7    2   15    0]
 [  22    5  846   39    7   17   17   12   61    6]
 [  24    6   34  829    3   41    6   11   48    8]
 [   5    7   14    7  783    9   20   19   20   98]
 [  28    6   19   60   16  641   26    5   74   17]
 [  21    5   20    3   21   29  845    4    8    2]
 [  12   14   22   17   26    7    5  849   27   49]
 [  28    7   29   51   22   70   11   17  709   30]
 [   7   10   10   14   78   21    7   42   20  800]]
Accuracy:  0.8241
```

3. minfreq = 10

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep2_10_pred.csv
Total number:  10000
The confusion matrix:
[[ 841    4   24   19    6   25   25    5   27    4]
 [   4 1093    6    4    1    5    7    2   13    0]
 [  19    4  844   38   12   14   19   15   63    4]
 [  26    6   35  828    2   41    6   10   47    9]
 [   3    7   13    5  782   11   21   16   18  106]
 [  33    7   16   56   16  645   24    4   75   16]
 [  21    7   22    3   21   27  841    3    8    5]
 [   9   15   25   12   29    7    3  851   26   51]
 [  25    7   35   48   23   67   12   13  715   29]
 [   6    9    8   12   89   23    7   54   22  779]]
Accuracy:  0.8219
```

4. minfreq = 20

```
ddou@ddou-LENOVO-Y50-70:~/Desktop/CSCI5523/hw/CSCI5523_project3$ python3 showcon
fmatrix.py rep2_20_pred.csv
Total number:  10000
The confusion matrix:
[[ 833    5   26   26    6   28   22    7   26    1]
 [   4 1088    9    3    3    7    6    3   11    1]
 [  26    2  846   40   12    8   21   15   56    6]
 [  34    6   29  817    3   49    9   11   49    3]
 [   7    9   14    7  786   11   17   16   18   97]
 [  39    8   23   54   19  635   20    8   76   10]
 [  18    6   22    4   21   26  848    3    7    3]
 [  10   12   25   12   30   10    2  845   25   57]
 [  26    7   39   48   25   68   14   10  714   23]
 [   6    7    8   14   91   30    8   51   18  776]]
Accuracy:  0.8188
```