EECS394 Machine Learning

Problem Set 2

Student Name: Haikun
NetID: hlg483

Question 1.
Data reprocess for training set and validation set:

For each item with unknown attribute in data_set
    If item[attribute] is nominal
        item[attribute] = mode (all item[attribute] in data_set, item[attribute] is known)
    If item[attribute] is numerical
        item[attribute]= average (all item[attribute] in data_set, item[attribute] is known)

Classification:

If current node is nominal
    If instance[decision_attribute] is unknown or not any key value of node.children
        Randomly assign a branch (child), and continue classification
Else if current node is numerical
    If instance[decision_attribute] is unknown
        Randomly assign a branch (child), and continue classification

Question 2.
Note: because of the size and dimensionality of training set, the training set is randomly sampled into a 1000-record subset. Step size of gain_ratio_numeric() is 1, splits on numerical is 5 and tree depth is limited to 20. The first 16 leaf nodes are printed as below:

```
(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND
numinjured < 0.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND
numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent <
0.25883265915 AND winpercent < 0.610572681992 --> 1) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND
numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent <
0.25883265915 AND winpercent >= 0.610572681992 --> 0) OR
```

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent >= 0.25883265915 AND opprundifferential < 82.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent >= 0.25883265915 AND opprundifferential >= 82.0 AND winpercent < 0.657877376153 --> 1) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent >= 0.25883265915 AND opprundifferential >= 82.0 AND winpercent >= 0.657877376153 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential >= 94.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent >= 0.802881935831 --> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential < 9.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent < 0.0110579633989 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential < 0.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential < 18.0 AND rundifferential < 35.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential < 18.0 AND rundifferential >= 35.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND

```
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured < 1.0 AND opprundifferential < 29.0
AND temperature < 61.7945566892 --> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured < 1.0 AND opprundifferential < 29.0
AND temperature >= 61.7945566892 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured < 1.0 AND opprundifferential >=
29.0 AND oppwinningpercent < 0.233146724089 AND winpercent < 0.503474156774 -->
1) OR
```

Question 3.

```
(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential < 9.0 --> 1)
```

For a given record, if its numinjured is less than 2.0, oppnuminjured is no less than 1.0 and opprundifferential is less than 9.0, the classification result will be 1 (win the game)

Question 4.

1). Calculate the original_accuracy of tree
2). Make a new copy of the tree as new_tree
2). Use Breadth-First-Search to collect and arrange nodes of the new_tree into a list
3). For each node in the list, prune the node (make the node into a leaf) and calculate the accuracy of the pruned new_tree
4). Find the pruned node with maximal_accuracy
5). If maximal_accuracy > original_accuracy, prune that node from the tree and goto the 1st step; Else, return the tree as pruned tree

Question 5.

The first 16 leaf nodes are printed as below:

```
(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND
numinjured < 0.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND
```

numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent < 0.25883265915 AND winpercent < 0.610572681992 --> 1) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent < 0.25883265915 AND winpercent >= 0.610572681992 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential < 94.0 AND oppwinningpercent >= 0.25883265915 --> 1) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent < 0.802881935831 AND numinjured >= 0.0 AND opprundifferential >= 94.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured < 1.0 AND winpercent >= 0.802881935831 --> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential < 9.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent < 0.0110579633989 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential < 0.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential < 18.0 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >= 18.0 AND oppnuminjured < 2.0 AND numinjured < 1.0 --> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >= 18.0 AND oppnuminjured < 2.0 AND numinjured >= 1.0 AND oppwinningpercent < 0.339279429888 AND winpercent < 0.323583406808 --> 1) OR

```
(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured >= 1.0 AND oppwinningpercent <
0.339279429888 AND winpercent >= 0.323583406808 AND opprundifferential < 24.0 -
-> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured >= 1.0 AND oppwinningpercent <
0.339279429888 AND winpercent >= 0.323583406808 AND opprundifferential >= 24.0
--> 0) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured < 2.0 AND numinjured >= 1.0 AND oppwinningpercent >=
0.339279429888 --> 1) OR

(numinjured < 2.0 AND oppnuminjured >= 1.0 AND opprundifferential >= 9.0 AND
oppwinningpercent >= 0.0110579633989 AND rundifferential >= 0.0 AND
oppdayssincegame < 6.0 AND rundifferential < 106.0 AND opprundifferential >=
18.0 AND oppnuminjured >= 2.0 AND opprundifferential < 24.0 AND rundifferential
< 50.0 --> 0) OR
```

Question 6.

The number of leaves on unpruned tree = 98
The number of leaves on pruned tree = 39

Question 7.

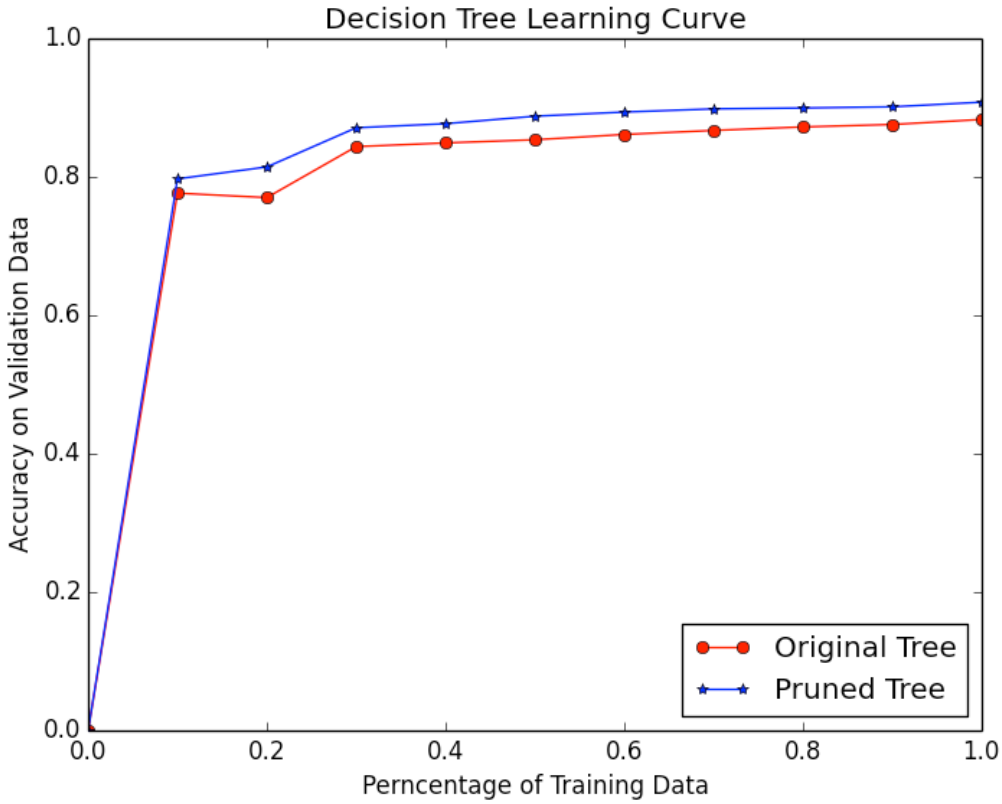
Accuracy on validation set: 0.882983193277
Accuracy on validation set (after pruning): 0.907983193277

The accuracy of pruned tree on validation_set is higher than unpruned tree. That is because:

1) the reduce-error-pruning is applied to relieve the over-fitting of the unpruned tree, which is built on training_set;

2) the objective of reduce-error-pruning is to improve the accuracy on validation_set

Question 8.



The overall accuracy of pruned tree with respect to different sample sizes is higher than unpruned tree, as shown in the plot above, with the learning curve of pruned tree is higher than unpruned tree.

Question 9.

Pruned tree. The reduce-error-pruning relieves the over-fitting of original tree, so that the pruned tree could be generalized to give more accurate predictions.

Question 10.

Haikun Liu(hlg493): ID3.py, pruning.py, preprocess.py
Wenting Zhou(wzg249): node.py, graph.py, predictions.py

Question 11.

Note: Only the first 1000 records of the training_set are used. The values in the table are accuracy measurements on validation_set

Step size	1	2	5	10	20	100	200
Information Gain Ratio	0.866	0.872	0.865	0.865	0.854	0.811	0.815

Information Gain	0.870	0.867	0.861	0.852	0.857	0.809	0.775

As shown in the table above, using Information Gain Ratio instead of Information Gain could boost accuracy; and smaller step size for numerical attributes could improve accuracy, however, it takes longer to train the decision tree than larger step size.