CS 6375 – Machine Learning
Homework#1

Name: Revanth Segu

The program expected seven arguments as mentioned in home work(<L> <K> < training_set> < validation_set> < test_set> <to-print> <Hueristic_no>).

L : integer (used in post pruning)
K : integer (used in post pruning)
training_set : absolute path of Training set file(in csv format)
validation_set : absolute path of Validation set file(in csv format)
test_set : absolute path of Test set file(in csv format)
to-print :{yes,no} yes – prints tree (both normal and post pruned)
Hueristic_no :{1,2,3} heuristic for which tree need to be printed

If training set file is present in the same directory then '.' Can be given instead of absolute path.

Hurestic No
1-Information Gain
2-One step look ahead
3-Variance imputiry

If an one the provided arguments are incompatible program exits and shown error message. The programs expects any string in any of the fields(XB,XC,.. XU, Class) '0' is assigned instead.

The program can be executed by following the three steps given below:

1.javac DecisionTreeNode.java
2.javac BuildDecisionTree.java
3.java BuildDecisionTree 1000 20 . . . no 1

In the above case taining_set, validation_set, test_set are present in same directory so '.' Is given instead of absolute path.

There are also methods for testing all the three heuristic trees (and their post pruned version).

1-Information Gain
Uncomment line - //testIGTree (); (to test tree accuracy with taining_set)
Uncomment line - // testTreeValid(igRoot); (to test accuracy with validation_set)
Uncomment line - // testTreeTrain(igRootPP); (to test post pruned tree accuracy with taining_set)
Uncomment line - // testTreeValid(igRootPP); (to test post pruned accuracy with validation_set)

2-One step look ahead
Uncomment line - // testOlaTree (); (to test tree accuracy with taining_set)
Uncomment line - // testTreeValid(olaRoot); (to test accuracy with validation_set)
Uncomment line - // testTreeTrain(olaRootPP); (to test post pruned tree accuracy with taining_set)
Uncomment line - // testTreeValid(olaRootPP); (to test post pruned accuracy with validation_set)

3-Variance impurity
Uncomment line - // testVITree (); (to test tree accuracy with taining_set)
Uncomment line - // testTreeValid(viRoot); (to test accuracy with validation_set)
Uncomment line - // testTreeTrain(viRootPP); (to test post pruned tree accuracy with taining_set)

Uncomment line - // testTreeValid(viRootPP); (to test post pruned accuracy with validation_set)

Uncomment line //printTrainData(); to print the training set

Accuracies of decision trees(before post pruning):

Default Data sets
1-Information Gain : 94.67%
2-One step look ahead : 95.67%
3-Variance impurity : 94.67%

data _sets-1
1-Information Gain : 74.85%
2-One step look ahead : 78.05%
3-Variance impurity : 75.3%

data_sets-2
1-Information Gain : 67.17%
2-One step look ahead : 73.67%
3-Variance impurity : 67.67%

Accuracies of Post Pruned decision trees:
For L- 100000 K-20 Program takes approximately 1 to 2 minutes to execute
For L- 1000 K-20 Program takes approximately 2 to 5 seconds to execute

Default Data sets
1-Information Gain:

L(down) K(Right→)	20	50	100
100	94.67%	94.67%	94.67%
500	94.67%	94.0%	94.0%
1000	94.0%	94.17%	94.0%

2-One step look

L(down) K(Right→)	20	50	100
100	96.0%	95.5%	95.67%
500	96.0%	96.0%	95.83%
1000	95.83%	94.0%	96.0%

3-Variance impurity

L(down) K(Right→)	20	50	100
100	94.67%	94.0%	94.67%
500	94.67%	93.67%	94.67%
1000	94.0%	93.83%	94.67%

Data_ sets_1

1-Information Gain

L(down) K(Right→)	20	50	100
20000	77.0%	78.55%	77.75%
50000	78.25%	77.9%	78.75%
100000	78.9%	77.7%	77.35%

2-One step look

L(down) K(Right→)	20	50	100
20000	83.2%	81.4%	81.9%
50000	82.1%	82.05%	81.6%
100000	80.8%	81.55%	82.45%

3-Variance impurity

L(down) K(Right→)	20	50	100
20000	78.05%	77.9%	77.95%
50000	78.35%	78.65%	77.8%
100000	78.45	78.9%	77.55%

Data_sets_2

1-Information Gain

L(down) K(Right→)	20	50	100
20000	71.67%	70.5%	68.83%
50000	71.17%	71.67%	69.83%
100000	72.0%	71.175	70.55%

2-One step look

L(down) K(Right→)	20	50	100
20000	76.0%	74.33%	73.83%
50000	74.17%	74.33%	74.5%
100000	74.67%	74.33%	74.67%

3-Variance impurity

L(down) K(Right→)	20	50	100
20000	71.33%	70.33%	72.55
50000	71.67%	69.83%	71.5%
100000	73.0%	72.17%	71.33%