

Question 1)

a) 1) (58)

2) (58) (85)

3) (58) (85) (93)

4) (58) (24) (85) (93)

5) (58) (24) (85) (19) (44) (93)

6) (58) (24) (85) (19) (44) (93)

7) (58) (24) (85) (19) (44) (93)

8) (58) (24) (85) (19) (44) (93)

9) (58) (24) (85) (19) (44) (93)

10) (58) (24) (85) (19) (44) (93)

b) Preorder [58, 24, 19, 13, 44, 37, 57, 85, 93, 94]
Inorder [13, 19, 24, 37, 44, 57, 58, 85, 93, 94]
Postorder [13, 19, 37, 57, 44, 24, 94, 93, 85, 58]

1) (58) (24) (85) (19) (44) (93)

2) (58) (24) (85) (19) (44) (93)

3) (58) (24) (85) (19) (44) (93)

4) (58) (13) (85) (37) (93) (57)

5) (58) (13) (85) (37) (93) (57)

In my program, first of all i write my calculateEntropy that looks class counts to calculate entropy. Then, i calculated information gain after splition by looking children node's entropy and parent's entropy. The difference according to formula that is given in the homework pdf gives us the information gain after split. Then, i tried to create my decision tree structure according to given data and labels. First of all, i decided to give each node a classId (int) , featureId (int) , isLeaf (bool), rightN (pointer to right node children), leftN(pointer to its left node children), usedSamples(bool array after splition which items goes for right Node) and usedSamplesL(bool array after splition which items goes for left Node). Because after splition some of them will be true for left, others will be true for right so i decided to hold both of the array in node to give to their children correct usedSamples array. First of all, in train method i tried to find my best feature Id to give it to root node. I wrote my best Id finder method that returns best feature index integer by calculating each feature's information gain according to usedsamples. After giving this integer value to featureId of root, train method changes usedSamples accorfding to this best id feature. Some of datas go to left children (falses), some of them goes for right children (trues). I recursively recall (for each left and right node pointers) the my recursion train method that does samething (go for left finding best feature id changing usedSamples for right and left and recalling it for right and left node pointer) . It's base case is checking if a node reached purity. If it reached purity, then it gives isLeaf (bool) true and returns. However, i encountered a problem in my train method that i cannot solved. It doesn't create a tree that is desired as in homework. Because i understand how a tree should be, i tried to do my predict method. It takes a new data bool array and starts from beginning of tree, until it reaches a leaf node it goes left or right childrens. After reaching leaf node it returns it's classId. Then i tried to implement test method, that takes another data two dimensional dynamic bool array and label. I apply predict function for each row of two dimensional data array and check their results with label that is given as an input in the function. I check whether my prediction true or false and according to their true and false counts i return accuracy. In my print method, i tried to go for each nodes in preorder and print their feature id and class id (if it is leaf). At the end, because of i couldn't solve the problem in my train method, i couldn't test them but according to what i understand i tried to do them.

Time Complexity Analyses

calculateEntropy : $O(n)$	calculateInformationGain : $O(n^2)$	train: $O(n \log n)$
predict: $O(n)$	test: $O(n^2)$	print: $O(n \log n)$