Machine Learning (CS60050) - Assignment 1 Report

NIKHIL SARASWAT - 20CS10039

AMIT KUMAR - 20CS30003

Tasks

1 Split Dataset A into 80%-20% to form training and testing sets, respectively. Build a Decision Tree Classifier using ID3 algorithm. Train the classifier using Information Gain (IG) measure (no packages to be used for Decision Tree Classifier).

2 Repeat (1) for 10 random splits. Print the best test accuracy and the depth of that tree.

3 Perform reduced error pruning operation over the tree obtained in (2). Plot a graph showing the variation in test accuracy with varying depths. Print the pruned tree obtained in hierarchical fashion with the attributes clearly shown at each level.

4 Prepare a report including all your results.

Dataset A

Customer segmentation is the practice of dividing a customer base into groups of individuals that are similar in specific ways relevant to marketing, such as age, gender, interests and spending habits. Companies employing customer segmentation operate under the fact that every customer is different and that their marketing efforts would be better served if they target specific, smaller groups with messages that those consumers would find relevant and lead them to buy something. Companies also hope to gain a deeper understanding of their customers' preferences and needs with the idea of discovering what each segment finds most valuable to more accurately tailor marketing materials toward that segment. This data set contains a total of 8068 customer details which are categorized into 4 segments (A, B, C, D).

The Decision Tree Algorithm Used

- We have used the ID3 algorithm for constructing the decision tree. However, the standard ID3 algorithm is restricted to attributes that take on a discrete set of values. Since in our Dataset there are some attributes which are varying very much like age, so we have divided those features in some blocks of values, like we have divided the age in block of 10 (i.e. 0-10 age will be given 1, 10-20 age will be given 2,……, 90-100 will be given 10,…., etc.)

- We are taking best attribute for growing the tree using Information gain. Then we are growing edges for all attribute values (we have discretised some attributes like – age). In this way we are taking an attribute only once at a depth.

- Here we are greedily choosing the best attribute and recursing over all of the values of this attribute which will give us the tree which can give better accuracy according to ID3 Algorithm.

Below is the pseudo-code of the procedure build_tree that explains the details of the algorithm.

```
build_tree(examples, depth):
    if size of examples == 1 or all examples have the same outcome value:
        return a leaf node with label same as the outcome value of the
        examples
    if depth == max depth:    # if we don't to grow the tree above a certain depth
        return a leaf node with label as the most probable value from the
    examples create node
    for attribute in list of attributes:
       find Information gain of all attributes and then sort them
        best attr = attribute with max gain
        node ← best att
    edge values = values of the best attributes
    node..append(build tree(left examples, depth + 1)
     return node
```

Some Important Terms and Definitions

Information Gain: For a collection S, $Entropy(S) = -\sum_i p(i)\, log_2\, p(i)$. The information gain is the reduction in entropy after choosing an attribute A. Mathematically,

$$InformationGain(S, A) = Entropy(S) \; - \sum_{\substack{v \in \\ values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Accuracy : $$Accuracy = \frac{No.\ of\ examples\ correctly\ classified}{Total\ no.\ of\ examples}$$

Procedure and Results

1 Determining Average Accuracy Over 10 Random Splits

Procedure :

- Divide the entire dataset into two parts - 80% training data, and 20% test data.
- Take some data for validation and keep it aside, as it will be used for pruning later.
- Repeat the above steps 10 times, and record the average test accuracy obtained
- Store the decision tree with the best test accuracy, as it will be used later for pruning.

Results :

Split No.	Test Accuracy (in %)
1	46.40644361833953
2	46.28252788104089
3	45.9727385377943
4	46.220570012391576
5	45.9727385377943
6	44.237918215613384
7	47.14993804213135
8	46.344485749690215
9	45.60099132589839
10	45.97278785377943

Average Test Accuracy: 46.020%
Best Test Accuracy: 47.150%

2 Variation of Depth and Number of Nodes

Procedure :

- Vary the maximum depth from to observe the variation of test accuracy with the maximum depth of the decision tree
- Create 5 decision trees for each depth using 10 random 80/20 splits, and use the average of their accuracies to determine the accuracy for that depth. Keep track of the test accuracy for each depth.

Depth wise accuracy is mentioned below: -

Depth: 1, Accuracy: 43.65551425030979

Depth: 2, Accuracy: 48.4634448574969

Depth: 3, Accuracy: 49.50433705080546

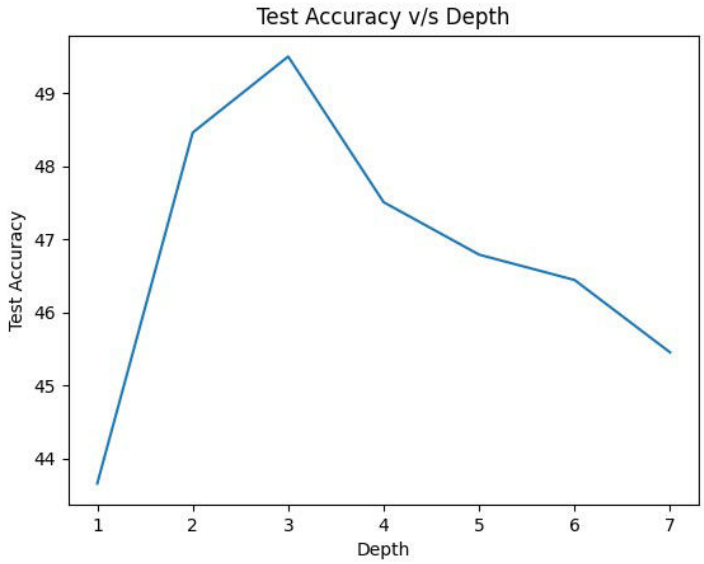
Depth: 4, Accuracy: 47.5092936802974

Depth: 5, Accuracy: 46.7905824039653

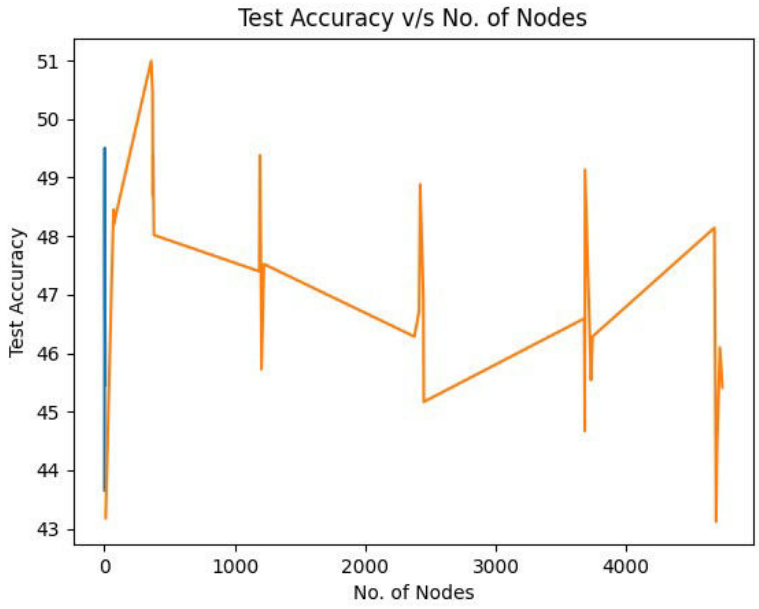
Depth: 6, Accuracy: 46.443618339529124

Depth: 7, Accuracy: 45.452292441140024

Results :



As can be seen from the data and plot, the best depth is 3. At depth 7, the training accuracy grows monotonically and approaches more than 90%. This is to be expected because the deeper the tree, the more training instances can be accommodated in the decision tree. We can also see that after depth 10, the accuracy drops dramatically. This is also consistent with the idea and occurs as a result of overfitting.



3 Pruning

Procedure:

- Take the decision tree with best test accuracy
- Because leaf nodes cannot be trimmed, begin with their parents. For each such node, prune the subtree below temporarily and check the accuracy on the validation set.
- If the accuracy improves, prune the tree below permanently and make the current node a leaf node by labelling it with the majority vote.
- Move up the tree and continue the same process.
- Continue till the accuracy on the validation set increases

We have the decision tree with best test accuracy (we have found it earlier steps),

We are presently doing the pruning procedure on this tree. We divided the 80% training data into two sets: 60% grow, which we utilised to train and build the decision tree, and 20% validation. We are now employing the validation set. Our method is known as reduced-error pruning. Pruning has the advantage of greatly minimising overfitting.

Results:
Best Accuracy Before Pruning : 45.081332300542215
Best Accuracy After Pruning : 48.7993803253292

Major Classes and Functions Used in the Code

We have used object Oriented Paradigm in our Model

There are two major classes – Node and Decision Tree.