REPORT

Methods used to implement Decision Tree:

Entropy

Entropy = sum of (p(i)*log2(1/p(i)) where i denotes different attribute values for the target attribute and p(i) denotes probability of selecting ith value from the sample.
This function takes a list of indices denoting indices of training samples which are to be considered to calculate entropy at given node and computes and returns corresponding entropy.

GainRatio

Gain ratio = information gain/split information
= (entropy of parent node-weighted sum of entropies of child nodes) / (split info)
weighted entropy of child = p(i)*Entropy(data with attribute value as ith value)
Split info = sum of (p(i)*log2(1/p(i)) where i denotes different attribute values for given attribute of parent node and p(i) is probability of selecting them from current sample
This function takes an attribute and a subset of indices of training dataset and returns gain ratio corresponding to splitting the given subset of training set with respect to input attribute.

MaxGainAttribute

This function takes a subset of indices of training dataset and a boolean list indicating the attributes considered to split tree to reach current node from root node and returns the best attribute that was not chosen to split till now that has maximum gain ratio by iteratively computing gain ratio for all unselected attributes in boolean list over the input subset of training set.

printTree

Takes a node of decision tree and level of the node as input and recursively prints the tree by printing all nodes on the path from current node to each leaf in subtree rooted at the current node in a DFS pattern.Printing a node includes details like level,node type-leaf or internal,node label-used to reach current node from its parent,node attribute-attribute used to split the tree at current node if it is an internal node or value of no. of deaths(target attribute) if it is a leaf.

deathValue

Takes a node as input and uses the list of indices of training data stored in the node and finds the most frequently occurring label of target attribute in the corresponding subset of training data and returns this value of target attribute.

createTree

Takes a tree node,depth of the current node and a boolean list marking attributes if they are used in reaching the current node from the root of the tree as input.The node sent as parameter contains an attribute to be used to split the current node and list of indices storing a subset training dataset to be used for current node.createTree splits the current subset of training data into further subsets each having all samples with same value of node attribute and computes maxGainAttribute for each child node formed for distinct attribute

labels using above defined functions and stores list of indices of training data and this attribute corresponding to max gain into the new child node and recursively builds tree with the child node as root.So this function also works in a similar way as DFS.If at any point depth of node to be used to build tree(sent as input parameter) equals max possible depth or if all possible attributes are already used to split nodes from root to current node then that node will be marked as leaf node.

predictDeath

Takes the root of decision tree and a instance from dataset and predicts the death value using the attribute values in the given instance and the root.At each node,if there is a child node with label same as the attribute value of the instance corresponding to attribute at that node then function recursively computes on child node.If there is no proper child with required label,then deathValue of parent(current node) is returned.

accuracy

Takes the root of decision tree and a list of samples from dataset as input and returns the accuracy of that subset of dataset in floating point format.This method utilizes predictDeath method defined above and predicts death value(target attribute) for all data instances and if predicted value is same as given death value in that instance,that instance is considered in computing accuracy.

Accuracy =(No. of samples correctly predicted)/(Total samples)

pruneTree

Takes a node of decision tree from which pruning is to be performed,a validation dataset and the root of the current tree that is being pruned as input and performs pruning recursively on subtree rooted at input node in a bottom up process and finally making the tree root given in input parameters as the root of pruned tree.At each node,the function computes accuracy of validation dataset by making the current node as leaf node.If this accuracy is better than the accuracy of pruned tree till now then this node will be marked as leaf node or else function does no changes on current node and recursion continues to prune for parent nodes.

printAvgAccuracy

This method is used to print accuracies needed in Part 1 of the assignment.It takes a maxDepth and a boolean printEach as inputs and performs 10 random splits of dataset into 60:20:20 -training,validation and test datasets and for each split builds tree using methods defined above with maxDepth as given input parameter value.It prints test and train sets accuracy for each of these splits and average train and test set accuracies over all 10 splits if the input parameter printEach is true or else skips printing this part.It returns average test accuracy value and the seed used to split the dataset(by random library) to get the maximum test accuracy.

bestAccuracyTree

This method is used to obtain best depth tree for Part 2 of the assignment.It calls the above defined printAvgAccuracy function for all possible depths of trees as maxDepth

value(0-5) and considers the depth for which average test accuracy is maximum.It returns that depth value and the corresponding seed value used to split dataset to model that tree.

pruneAndPrint

This method is used to print accuracies before and after pruning as part of Part 3 in the assignment.It takes as input a maxDepth value, and seed value corresponding to the tree with best test set accuracy obtained from part 2 and build a decision tree without pruning and prints accuracies on train ,test and validation sets.Now it performs pruning using validation data and returns root of pruned tree.

Data Structure - Node:

Attribute name		Description
leaf	-	a boolean,true if current node is leaf node,false by default
attribute	-	attribute name used to split decision tree at current node(if node is internal) or death value(target attribute value - if node is leaf)
children	-	a list of Node objects denoting children of current node
label	-	attribute value of parent's attribute type used to split data from parent to this child node,it is None for root
listOfIndices	-	a list of indices corresponding to indices in training data that are used to build tree from the current node
depth	-	Level of current node in the tree with root starting from 0

A decision tree is represented by its root node.To traverse through the tree ,children attribute in each node is used,i.e. To go from level i to level i+1 in tree,all the nodes in children attribute of all nodes at level i are to be utilised.

Implementation

The given dataset is loaded and split into 3 parts of 60% used for training the decision tree ,20% as validation data used only while pruning the tree and the last 20% as test data used to check accuracy after building the tree wherever necessary.As date and time are continuous and we found no effect of them on the target data,we did not consider it as an attribute to train the model.As death values lied in range 0-5, we modelled it as a classification problem to predict one of the values in 0-5.State/Union Territory is considered as discrete valued attribute with each name representing one discrete value.For the other attributes,we did a frequency based split of intervals to keep the number of samples in input dataset belonging to each interval as close as possible.
Interval splits used:
ConfirmedIndianNationals: [0-1], [2-3], [4-30], [31-42], [43-55], [56-100] ,[101-177]
ConfirmedForeignNationals: [0-0] , [1-1], [2-2], [3-4], [5-9], [10-13], [14-14]
Total Confirmed:[0-1], [2-3], [4-30], [31-42],[43-55], [56-100], [101-180]
Cured: [0-0], [1-2] , [3-3] , [4-10] , [11-25]
So our model has input attributes as States, ConfirmedIndianNationals, ConfirmedForeignNationals, TotalConfirmed, Cured and target attribute as Deaths.
As a statistical measure to find the best attribute to be used to split the tree at any internal node,we used the value of gain ratio as described in function implementation above.

For pruning, accuracy improvement is the measure used to decide whether to mark an internal node as leaf node.Maximum possible depth of tree constructed by this method is 5 as the number of attributes used is 5.So if maxDepth for part 1 is greater than 5 the results(accuracy,tree) will be same as that of when maxDepth is 5.i.e. Plot of accuracy vs depth saturates and remains constant for depth >= 5.

Part -1:

Program takes an integer as input and builds a tree using printAvgAccuracy and buildTree functions described above for 10 different splits of dataset and prints train and test set accuracy for each split as well as the average accuracies over 10 splits.

Part-2:

Program uses the bestAccuracyTree and printAvgAccuracy and buildTree methods described above and selects the depth that has the best average test data accuracy among all different depths and considers the seed of tree that has the highest test accuracy among all 10 splits and trees built using that depth as max depth. It then prints the best depth as well as the seed from the function obtained as the output.

Part-3:

Program uses the seed and depth values generated from previous part and calls the prueAndPrint method described above to perform pruning on tree constructed with these depth and seed values and prints the accuracy details before and after pruning.

Part-4:

Program uses the printTree method and prints the pruned tree generated from the previous part following the proper hierarchy of nodes and describing each node using suitable parameters like type of node(leaf or internal),level of node,attribute values etc as defined in the printTree method above.

Results

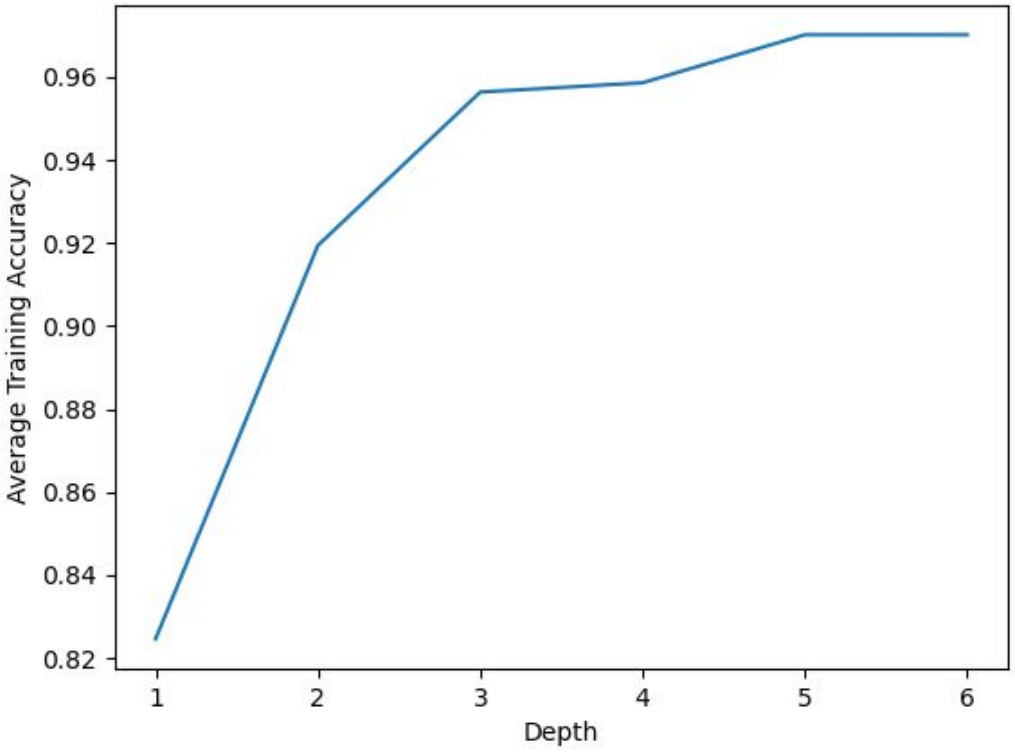
The output test accuracy increases as the depth increases. But as the depth reaches maximum possible depth, test accuracy becomes constant. The best average training accuracy is for the tree with depth 5 and its value is equal to 97.01%. The best average test accuracy is for the tree with depth 5 and its value is equal to 87.75%.

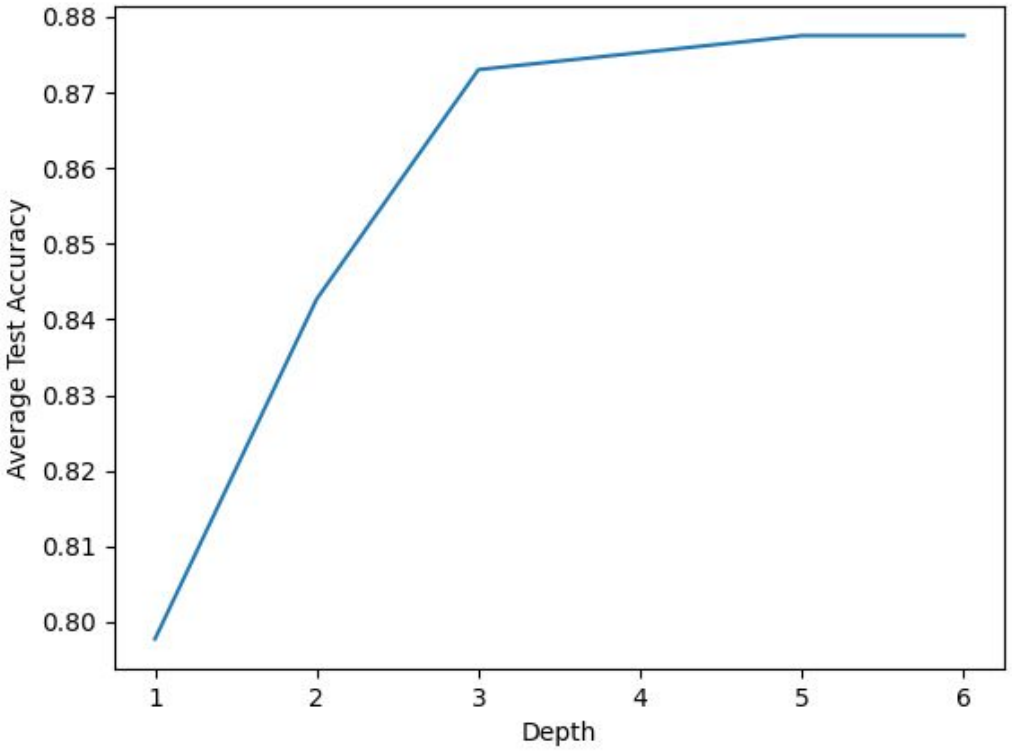
After analyzing the test accuracies out of the randomly chosen 10 splits, the best test accuracy for the tree of depth 5 is 89.88% and for the purpose of pruning the accuracy of the validation set is 84.27%. The above discussed results are for the tree before pruning. Post pruning the results are

Validation set accuracy after pruning: 88.76%

Test accuracy after pruning: 91.01%

Explaining Part - 2 Results(Plot for Test Accuracy vs Depth)





Depth	Avg Training Accuracy(%)	Avg Test Accuracy(%)
1	82.46	79.77
2	91.94	84.27
3	95.63	87.30
4	95.85	87.53
5	97.01	87.75
6	97.01	87.75

Based on the above table the plots for Average Training Accuracy vs Depth and Average Test Accuracy vs Depth have been taken using matplotlib python library. As we can see from the plots the average training accuracy increases at a big rate as we go deeper this is because of the fact that the tree is getting more and more specific. In the case of average test accuracy the rate of increase is very less as the tree grows deeper and it becomes constant after depth 4, so we have chosen the best depth as 5 for parts 3 and 4 ie., to prune the tree and print the final tree.