Machine Learning
Report on Assignment – 1
Decision Tree

By: Group 32
Krinal Patel (21CS60R39)
Sarvesh Gupta (21CS60R53)

Dataset: Indian Liver Patient Dataset

Details of Dataset:
This Dataset of "Indian Liver Patient Dataset" is taken from Kaggle. This data set contains 416 liver patient records and 167 non liver patient records. The data set was collected from test samples in North East of Andhra Pradesh, India. 'is_patient' is a class label used to divide into groups (liver patient or not). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

Link of Dataset: https://www.kaggle.com/jeevannagaraj/indian-liver-patient-dataset

Procedure:

Data Pre-Processing:

- We are handling the missing values, by replacing it with the median of that column. We came to a conclusion of going for median, rather than choosing mean, or any other value, is based on final computation. As by selecting median, we are getting better accuracy at the end. (This difference is very less, as we have only 4 missing values, but difference is there).
- We have implemented train_test_split function to split dataset into 2 parts, training and testing. Ratio of

  training dataset : testing dataset == 80 : 20

Building the Tree:

The Decision Tree is built recursively, and each node of the tree have 2 child nodes. As mentioned in assignment, we have built the tree using 2 impurity measure:

- Gini Impurity
- Information Gain

Gini Impurity is one of the methods used to decide the optimal split for all the node. Lesser the Gini Impurity, more optimal is the split.

$$Gini\ Impurity = 1 - Gini$$

$$Gini = (p_1^2 + p_2^2 + p_3^2 + \ldots\ldots\ldots + p_i^2)$$

(p_i = Probability of particular category of nodes among all the categories)

$$Weighted\ Gini\ Impurity = (GI_1 * (N_1 / N)) + (GI_2 * (N_2 / N)) + \ldots\ldots\ldots\ldots + (GI_j * (N_j / N))$$

(GI_j = Gini Impurity of j^{th} child of Parent Node)
(N_j = Number of elements in j^{th} child)
(N = Number of elements in parent node)

Average accuracy of Tree (Gini Gain): 0.65

Information Gain determines the best attributes or features, that gives maximum information about the class. Higher the Information Gain, better is the split.

$$Information\ Gain = 1 - Entropy$$

$$Entropy\ (P) = -\ [(p_1\ log_2(p_1)) + (p_2\ log_2(p_2)) + \ldots\ldots\ldots\ldots + (p_i\ log_2(p_i))]$$

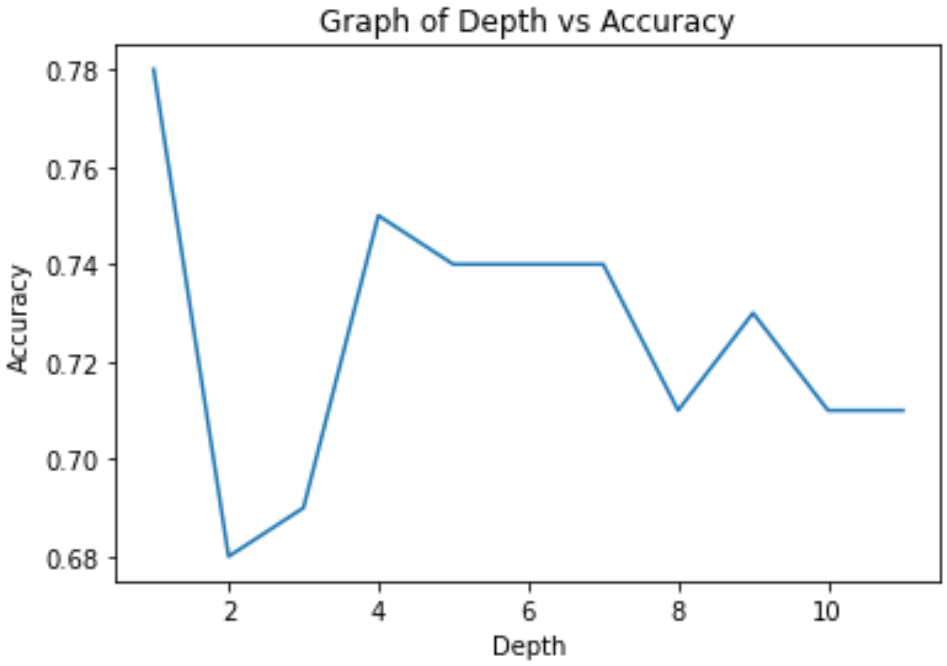
p_i = Probability of i^{th} category in the class

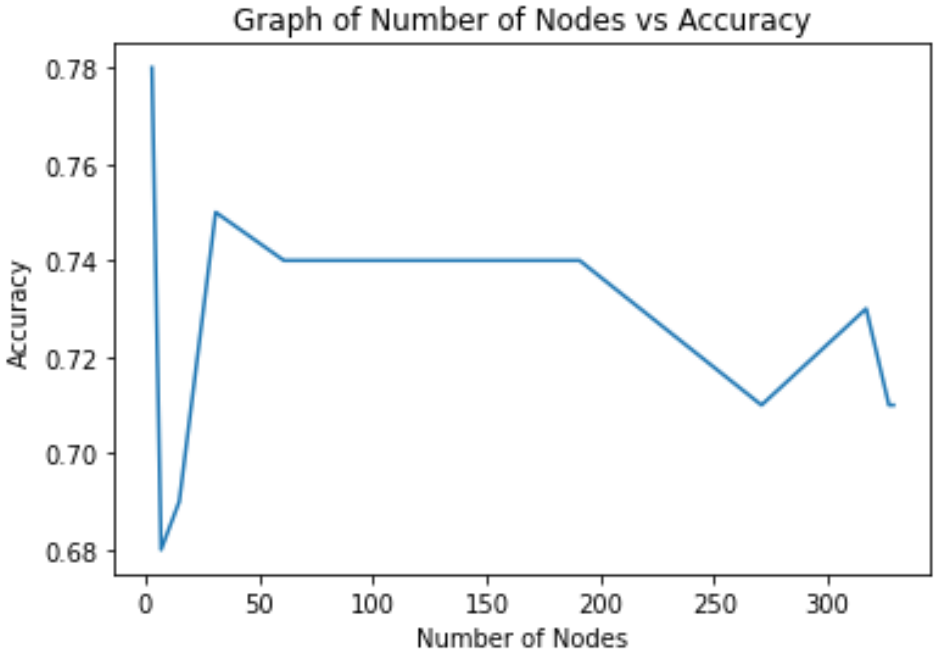
And averaging the accuracy obtained by running both algorithm 10 times, we found that we are getting better results while using Information Gain. So, we built our dataset using information gain.

Average accuracy of Tree (Info Gain): 0.66

Best Possible Depth Limit to be used for our dataset is:
Maximum depth of a best tree 11
Depth: 1 Nodes: 3 Accuracy: 0.78
True Positive: 91 True Negative: 0 False Positive: 26 False Negative: 0
Depth: 2 Nodes: 7 Accuracy: 0.68
True Positive: 74 True Negative: 6 False Positive: 20 False Negative: 17
Depth: 3 Nodes: 15 Accuracy: 0.69
True Positive: 75 True Negative: 6 False Positive: 20 False Negative: 16
Depth: 4 Nodes: 31 Accuracy: 0.75
True Positive: 80 True Negative: 8 False Positive: 18 False Negative: 11
Depth: 5 Nodes: 61 Accuracy: 0.74
True Positive: 78 True Negative: 8 False Positive: 18 False Negative: 13
Depth: 6 Nodes: 115 Accuracy: 0.74
True Positive: 77 True Negative: 9 False Positive: 17 False Negative: 14

Depth: 7 Nodes: 191 Accuracy: 0.74
True Positive: 76 True Negative: 11 False Positive: 15 False Negative: 15
Depth: 8 Nodes: 271 Accuracy: 0.71
True Positive: 72 True Negative: 11 False Positive: 15 False Negative: 19
Depth: 9 Nodes: 317 Accuracy: 0.73
True Positive: 73 True Negative: 12 False Positive: 14 False Negative: 18
Depth: 10 Nodes: 327 Accuracy: 0.71
True Positive: 71 True Negative: 12 False Positive: 14 False Negative: 20
Depth: 11 Nodes: 329 Accuracy: 0.71
True Positive: 71 True Negative: 12 False Positive: 14 False Negative: 20





As we can see graphically, as well as by our accuracy results, that depth = 1 is producing maximum accuracy of 0.78, but we cannot limit our depth to 1. As after closer inspection we observed that True Positive: 91, True Negative: 0, False Positive: 26, False Negative: 0 which means it always classify it as a liver patient. And it means it is not learning anything. So, we can't consider it as valid depth and opting for second maximum accuracy, that is for **depth = 4.**

Pruning:

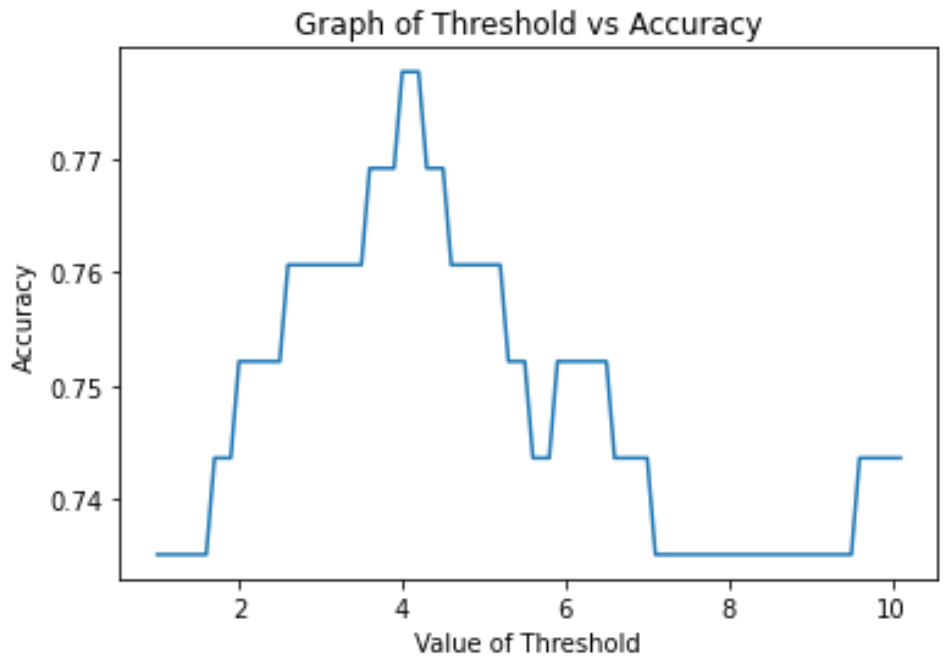
To avoid overfitting pruning is required. We have used statistical test chi square method to prune the tree

$$K = \sum_{\substack{\text{all classes } i \\ \text{children } j}} \frac{(N_{ij} - N'_{ij})^2}{N'_{ij}}$$

K is the chi-square value.
Nij= Number of points from class i in child j
N'ij= Number of points from class i in child j assuming a random selection
N'ij= Ni x Pj

Here it computes chi square at each node. Higher chi square means higher significance and vice versa. Pruning function recursively start from leaf and prune the node whose chi square is less than threshold value.

To Compute Threshold, we have tested multiple values, and obtained the maximum accuracy, for this we have plotted the graph of Threshold vs Accuracy:



Obtained value of threshold is 4, as it has maximum accuracy. So for Threshold = 4, we had performed pruning.

	Before Pruning	After Pruning
Accuracy for Complete Tree	0.71	0.78
Depth	11	8
Number of Nodes	329	137

Conclusion:

Average Accuracy over 10 Times:
Accuracy of Tree build using Gini Gain = 0.65
Accuracy of Tree build using Information Gain = 0.66

So, we finalized the tree with Information Gain as impurity measure to get the optimal split.
Best Accuracy of Tree Over Testing Dataset: **0.71**
Depth of the Tree: **11**
Number of Nodes in the Tree: **329**

Best Possible Depth Limit: **4**

After Pruning:
Best Accuracy of Tree Over Testing Dataset: **0.78**
Depth of the Tree: **8**
Number of Nodes in the Tree: **137**