Decision Tree - ClickStream Analysis using ID3 Algorithm

Pavan Manjunath - SBU ID – 109916057

Santosh Kumar Ghosh – SBU ID – 109770622

Goal

Given a set of page views, will the visitor view another page on the site or will he leave?

Algorithm

We have employed **ID3 Decision Tree algorithm** to solve this problem. As the attribute (feature) values are continuous, we construct a **Binary Tree**, whose left sub-tree will contain all entries lesser than the **median** of the attribute's values and the right sub-tree will contain all entries greater than or equal to the median of the attribute's values. The **binary tree is constructed recursively** until the stopping condition is reached. The **stopping condition** being when **all attributes are expanded** or when we arrive at a **pure set.**

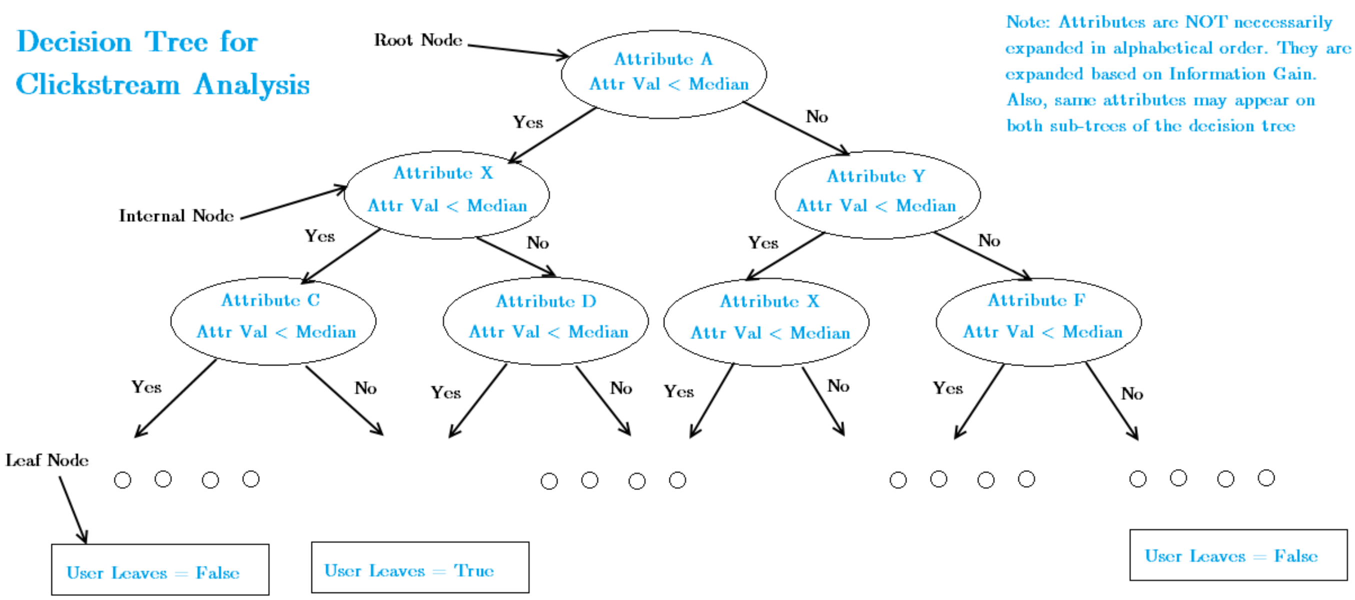


Figure 1: Decision Tree for ClickStream Analysis

Information Gain - Choosing the Best Attribute

The best attribute is one whose information gain is the maximum. Information Gain is calculated based on the following formula-

$$Gain(S,A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} H(S_V)$$

V ... possible values of A
S ... set of examples {X}
S_V ... subset where $X_A = V$

Where

- Entropy: $H(S) = - p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$ bits
  - S ... subset of training examples
  - $p_{(+)}$ / $p_{(-)}$ ... % of positive / negative examples in S

Pruning based on Chi-Square Test

The input data set has 274 attributes. As such, the decision tree will grow enormously and will result in over-fitting. Hence we need to **stop growing the tree whenever we encounter irrelevant attributes.** For this, we first calculate S, based on the formula-

$$S = \sum_{i=1}^{m} \left(\frac{(p_i^{'} - p_i)^2}{p_i^{'}} + \frac{(n_i^{'} - n_i)^2}{n_i^{'}} \right)$$

Where

$$p_i^{'} = p \frac{|T_i|}{N}$$

$$n_i^{'} = n \frac{|T_i|}{N}$$

where p_i, n_i are the positives and negatives in partition T_i.

Note that we have built a binary classifier. Hence the number of degrees of freedom (m) is 1. And for the given thresholds of 0.01, 0.05 and 1, we get Chi-Square Score of 6.65, 3.85 and 0. If the calculated value of S is less than the Chi-square score, then we treat that particular node as irrelevant and do not expand it.

Results

1. Threshold = 0.01

Correct Predictions	= 18,435
Total Predictions	= 25,000
Accuracy	= 73.74%
Number of Nodes in Tree	= 133
Total Run Time	= 89.536 seconds.

2. Threshold = 0.05

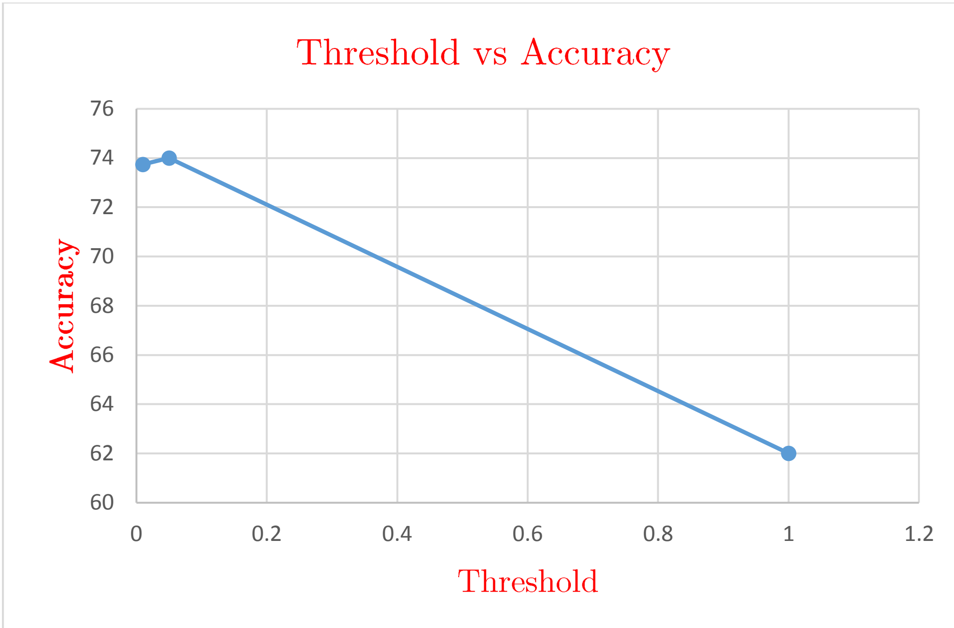
Correct Predictions	= 18,501
Total Predictions	= 25,000
Accuracy	= 74.004%
Number of Nodes in Tree	= 319
Total Run Time	= 115.665 seconds.

3. Threshold = 1 (Full Tree)

Correct Predictions	= 15,712
Total Predictions	= 25,000
Accuracy	= 62.848%
Number of Nodes in Tree	= 322,407
Total Run Time	= 855.542 seconds.

Observations and Improvements

As you see, when we grow the full tree, accuracy drops to approximately 62%. This is because of **Overfitting** – the three is tightly fit to the input data and is unable to generalize to new unseen data. As we prune the tree using Chi-Squared Test, the accuracy rises up to 74%.



Another observation is the **input is highly biased towards zero class**. Hence the Decision tree will be biased towards the zero class and will output more zeroes and less ones. **In order to correct this bias, we have applied "Under-sampling" and 'Oversampling" technique**. In under-sampling, we randomly drop a percentage of training samples that are labeled 0. In oversampling, we repeat all training samples that are labeled 1.

But unfortunately, we did not see significant improvement in accuracy.

Conclusion and Future Work

As seen from results, decision tree works better with pruning. In order to improve accuracy, we can attempt multi-way split instead of a two split. This may result in better accuracy and also reduce the number of nodes needed in the tree.