DECISION TREE LEARNING

Tree Based learning algorithms are most popular algorithms used for supervised machine learning technique. Tree based methods have high accuracy and good interpretation. These methods have good stability.
Methods like decision trees, random forests are existing tree based methodologies.

3.1 Decision Tree

Decision tree is one of the supervised machine learning algorithm. It is mostly used for classification and prediction purpose. It is useful for both categorical and continuous variables. In this technique whole sample set is divided into two or more homogeneous sets. Decision Trees are divided into two types:

1. Categorical Variable Decision Trees
   Decision Tree in which target variable is categorical variable such tree is called as Categorical Variable Decision Trees. For Example: For Customer database, Customer will buy product or not i.e ‘Yes’ or ‘No’.

2. Continuous Variable Decision Trees
   Decision Tree in which target variable is continuous variable such tree is called as Continuous Variable Decision Trees. For Example: For Customer database, Predicting Customer Income/Salary based on its buying preferences.

3.2 Terminologies Used in Decision Trees

1. Root Node- It is a representative of whole sample or population. Based on the values of root node whole population is divided into two or more homogeneous sets.

2. Splitting- Splitting is a process in which based on some criteria whole sample set if divided into partitions.

3. Split Node-When Sub node split into further nodes then that node is called as split node.

4. Leaf/Terminal Node-The Node which do not split is called as leaf node.
3.3 Deciding Where to Split

Decision tree accuracy depends on the split criteria used. Decision tree uses different criteria for deciding split. This criteria is decided based type of target variables. Following split methods are used in decision trees.

1) **Gini Index**-

   It is used mainly for binary split. It is useful for categorical values like ‘Yes’ or ‘No’. Higher is the value of Gini index, higher is the homogeneity. CART(Classification and Regression Tree) uses Gini index as split criteria. Gini Index for subnode is calculated by taking summation of squares of success and failure probabilities. Gini Index of Split node is calculated by using weighted Gini score of each node of split.
## Table 3.1: Sample Student Dataset

<table>
<thead>
<tr>
<th>Student Roll No</th>
<th>Gender</th>
<th>Class</th>
<th>Play cricket</th>
<th>Student Roll No</th>
<th>Gender</th>
<th>Class</th>
<th>Play cricket</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Male</td>
<td>Class IX</td>
<td>No</td>
<td>2.</td>
<td>Female</td>
<td>Class IX</td>
<td>No</td>
</tr>
<tr>
<td>3.</td>
<td>Male</td>
<td>Class IX</td>
<td>No</td>
<td>4.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
</tr>
<tr>
<td>5.</td>
<td>Male</td>
<td>Class IX</td>
<td>No</td>
<td>6.</td>
<td>Female</td>
<td>Class IX</td>
<td>No</td>
</tr>
<tr>
<td>7.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
<td>8.</td>
<td>Female</td>
<td>Class IX</td>
<td>No</td>
</tr>
<tr>
<td>9.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
<td>10.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
</tr>
<tr>
<td>11.</td>
<td>Female</td>
<td>Class IX</td>
<td>No</td>
<td>12.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
</tr>
<tr>
<td>13.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
<td>14.</td>
<td>Male</td>
<td>Class IX</td>
<td>Yes</td>
</tr>
<tr>
<td>15.</td>
<td>Male</td>
<td>Class X</td>
<td>Yes</td>
<td>16.</td>
<td>Female</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>17.</td>
<td>Female</td>
<td>Class X</td>
<td>No</td>
<td>18.</td>
<td>Female</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>19.</td>
<td>Male</td>
<td>Class X</td>
<td>Yes</td>
<td>20.</td>
<td>Male</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>21.</td>
<td>Female</td>
<td>Class X</td>
<td>No</td>
<td>22.</td>
<td>Male</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>23.</td>
<td>Male</td>
<td>Class X</td>
<td>No</td>
<td>24.</td>
<td>Male</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>25.</td>
<td>Male</td>
<td>Class X</td>
<td>No</td>
<td>26.</td>
<td>Male</td>
<td>Class X</td>
<td>No</td>
</tr>
<tr>
<td>27.</td>
<td>Male</td>
<td>Class X</td>
<td>No</td>
<td>28.</td>
<td>Female</td>
<td>Class X</td>
<td>Yes</td>
</tr>
<tr>
<td>29.</td>
<td>Male</td>
<td>Class X</td>
<td>No</td>
<td>30.</td>
<td>Female</td>
<td>Class X</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Number of students in class=30

Number of Students playing Cricket=15

Number of Girls=10

Number of Boys=20

Number Boys Play Cricket=13
Number of Girls play Cricket=02
Number of students in Class IX=14
Number of students in Class X=16

**Split on Gender:**

Calculate, Gini for sub-node Female = $(0.2)*(0.2)+(0.8)*(0.8)=0.68$

Gini for sub-node Male = $(0.65)*(0.65)+(0.35)*(0.35)=0.55$

Calculate weighted Gini for Split Gender = $(10/30)*0.68+(20/30)*0.55 = 0.59$

**Similar for Split on Class:**

Gini for sub-node Class IX = $(0.43)*(0.43)+(0.57)*(0.57)=0.51$

Gini for sub-node Class X = $(0.56)*(0.56)+(0.44)*(0.44)=0.51$

Calculate weighted Gini for Split Class = $(14/30)*0.51+(16/30)*0.51 = 0.51$

As we can see that Gini score for Split on Gender is higher than Split on Class, hence, the node split will take place on Gender.

2) **Chi-Square**-

It is useful for categorical target variable like success or failure. It can perform two or more splits. Higher is the value of chi-Square, higher is the statistical difference between sub nodes and parent nodes. It is calculated by finding summation of squares of differences between expected and actual. Chi-Square for each node is calculated by using formula:

$$\text{Chi-Square} = \frac{(\text{Actual–Expected})^2}{\text{Expected}^{1/2}}$$ (3.1)

For Above problem, Actual value for node “Play Cricket is 2” and “Not Play Cricket” is 8.

As parent node probability is 50% (15 students out of 30 play cricket).Expected value would be 5 for both “Play Cricket” and “Not play Cricket”. Chi-Square can be calculated by applying above formula. Chi-Square for Gender split is greater than class split.
3) **Information Gain**-

Information gain is based on entropy. Entropy is used to measure homogeneity of population. If sample is completely homogeneous then entropy is zero. If sample is equally divided into 50%-50%, then it has entropy 1.

\[
Ent\text{ropy} = -p \log p - q \log q \quad (3.2)
\]

\[
Information_{gain} = (1 - Ent\text{ropy}) \quad (3.3)
\]

Here, p and q are probabilities of success and failure. Lesser the entropy better is the attribute.

First Entropy of parent node is calculated. Then entropy for each split node is calculated and weighted average of all sub nodes is calculated based on split.

For Above Example,

Entropy for parent node (Play Cricket) = \((-15/30 \log (15/30) - (15/30) \log (15/30)) = 1\)

Entropy for female node = \((-2/10 \log (2/10) - 8/10 \log (8/10) = 0.72\)

Entropy for Male node = \((-13/20 \log (13/20) - 7/10 \log (7/20) = 0.93\)

Entropy for Split on Gender = \((10/30) * 0.72 + (20/30) * 0.93 = 0.86\)

Entropy for Class IX node = \(- (6/14) \log_2 (6/14) - (8/14) \log_2 (8/14) = 0.99\)

Entropy for Class X node, \(- (9/16) \log_2 (9/16) - (7/16) \log_2 (7/16) = 0.99\).

Entropy for split Class = \((14/30)*0.99 + (16/30)*0.99 = 0.99\)

We can observe here entropy of gender split is less. So, it more suitable for splitting.

### 3.4 Advantages of Decision Tree Modelling [75]

**Easy to Understand**: Decision tree output is very easy to understand. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.

**Useful in Data exploration**: Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict
target variable. It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.

**Less data cleaning required:** It requires less data cleaning compared to some other modelling techniques. It is not influenced by outliers and missing values to a fair degree.

**Data type is not a constraint:** It can handle both numerical and categorical variables.

**Non Parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

### 3.5 Dis-advantages of Decision Tree Modelling [75]

**Over fitting:** Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning

**Not fit for continuous variables:** While working with continuous numerical variables, decision tree looses information when it categorizes variables in different categories.

### 3.6 Decision Tree Algorithm

Decision tree is very useful classification and regression technique. Decision Trees are very flexible, easy to understand, and easy to debug. They will work with classification problems and regression problems. So if you are trying to predict a categorical value like (red, green, up, down) or if you are trying to predict a continuous value like 2.9, 3.4 etc., decision Trees will handle both problems. One of the good things about Decision Trees is they only need a table of data and they will build a classifier directly from that data without needing any up front design work to take place. Decision tree algorithm as a whole is explained below.

- **Decision node:** specifies a test on a single attribute
- **Leaf node:** indicates the value of the target attribute
- **Arc/edge:** split of one attribute
- **Path:** a disjunction of test to make the final decision

Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

The tree can grow huge. These trees are hard to understand.
Larger trees are typically less accurate than smaller trees.

Following measures are used for selection of an attribute to test at each node for classifying examples.

Information Gain-It measures how well a given attribute separates the training examples according to their target classification.

This measure is used to select among the candidate attributes at each step while growing the tree.

Entropy-A measure of homogeneity of the set of examples. Given a set $S$ of positive and negative examples of some target concept (a 2-class problem), the entropy of set $S$ relative to this binary classification is

\[ E(S) = - p(P) \log_2 p(P) - p(N) \log_2 p(N) \]  

(3.4)

The entropy is 0 if the outcome is `certain`.

The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).

Knowing the `when` attribute values provides larger information gain than `where`.

Therefore the `when` attribute should be chosen for testing prior to the `where` attribute.

Similarly, we can compute the information gain for other attributes.

At each node, choose the attribute with the largest information gain.

Stopping rule

Every attribute has already been included along this path through the tree, or

The training examples associated with this leaf node all have the same target attribute value (i.e., their entropy is zero).

### 3.7 Hoeffding Trees

Decision tree algorithms outperforms when the data is in large volume or continuous stream because these algorithms store data on memory. So, Hoeffding trees are used to handle this data. Hoeffding trees are decision trees based on hoeffding bound concept.

Hoeffding bound concept is used to delete the leaf nodes which are not active. So, the memory can be freed and used for other nodes.

The Hoeffding tree algorithm is given below.

Algorithm HoeffdingTreeInduction ($E$, HT)
Input: E is a training instance
Input: HT is the current state of the decision tree
1: Use HT to sort E into a leaf l
2: Update sufficient statistic in l
3: Increment the number of instances seen at l (which is nl)
4: if nl mod nmin = 0 and not all instances seen at l belong to the same class then
5: For each attribute, compute Gl(Xi)
6: Find Xa, which is the attribute with highest Gl
7: Find Xb, which is the attribute with second highest Gl
8: Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}}$
9: if Xa $\neq$ Xb and Gl(Xa)- Gl(Xb) $> \epsilon$ or $\epsilon < \zeta$ then
10: Replace l with a split-node on Xa
11: for all branches of the split do
12: Add a new leaf with derived sufficient statistic from the split node
13: end for
14: end if
15: end if

3.8 Summary

Decision tree algorithms are useful for analysis of massive data. A decision tree provides visual representation of data. Decision trees handle missing data easily. Accuracy of decision tree is better. We can use different splitting criteria’s with decision trees like Gini index, information gain or chi-square. Hoeffding trees are decision trees which apply real time processing of data. It is useful for big data analysis.