IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

From Information Theory To Predictive Modeling: Statistical Insights Into Decision Tree Algorithms

1Koyye Joshitha, 2Tailam Venkata gayatri, 3Darla Divya lakshmi prasanna

1Assistant professor, 2Assistant professor, 3Assistant professor

1Godavari Global University,

2Godavari Global University,

3Godavari Global University

Abstract:

Decision trees are a non-parametric supervised learning method used for classification and regression tasks in statistics, data mining, and machine learning. This approach constructs a tree-like structure to identify patterns in data, providing results that are both accurate and easily interpretable. Decision trees are capable of handling missing values, processing both numerical and categorical data, and are scalable to large datasets. They are widely applied in diverse domains such as radar signal classification, medical diagnosis, and speech recognition.

This paper provides an overview of decision tree methodology, including classification and regression trees, with a particular focus on the underlying probabilistic and statistical concepts such as entropy, information gain, gain ratio, and Gini index. These measures play a crucial role in evaluating node purity and guiding the tree-building process through recursive partitioning. The C5.0 algorithm, an industry-standard advancement over earlier decision tree models, is highlighted for its robustness, efficiency, and interpretability. An example using a student admission dataset is presented to illustrate the practical application of decision trees in predictive modeling and decision-making under uncertainty.

Introduction

A decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. It is a non-parametric supervised learning method used for both classification and regression tasks. They are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. The structure of a decision tree includes a root node, branches, and leaf nodes. Each internal node de- notes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

Decision tree methodology is more commonly known as learning decision tree from data. The algorithms that come under decision tree may further be classified as classification trees and regression trees. A decision tree with numeric output is called regression tree. By a classification tree, it is understood that the output variable is categorical. Classification trees are very popular because they are used in a wide variety of areas, they generate rules that are easy to interpret, and are easy to use. Classification trees are powerful because they can handle a variety of data types, scalable, can handle missing values, and, when used in ensembles of trees, they provide excellent accuracy. Classification trees have been successfully used for classification in diverse areas such as radar signal classification, character recognition, remote sensing, medical diagnosis, and expert systems and speech recognition. They require little data preparation in terms of parameter settings, and are well suited to exploratory knowledge discovery. They can handle both numerical and categorical data. It is possible to validate a model classification tree with statistical tests and for a human expert to interpret it. Consequently, the reliability of the model can be expressed and accounted for. Classification trees can handle datasets that may have errors or missing values. The most important characteristics of decision tree classifiers are their efficacy to divide a complex decision-making process into a number of simpler decisions, and thus provide a solution which is easily interpretable.

Decision Trees - An Overview

A classification tree divides the data into smaller and smaller portions to identify patterns that can be used for prediction. The knowledge so gained is presented in the form of a logical structure that can be easily understood without any statistical knowledge. This aspect of the model assists managerial personnel for developing business strategies and process improvement.

In a classification tree the output variable is a categorical variable. A decision tree model allows us to make prediction on an output variable, based on a series of rules arranged in a tree-like structure. Each rule in a decision tree corresponds to a series of split points. These split points are referred to as **nodes**. A node may or may not have a sub-node and this is based on the class composition of the data items currently available at that node. This class composition is usually expressed in terms of purity of the node. If every data item at a node belongs to a single class, then that node is called a **pure node**. A pure node becomes **leaf node**, meaning that no further split of the data points is possible. Every node is labelled with class name and this is decided based on the highest proportion of observations of a particular class available at that node. Depending on the error rate that we may commit or the **bucket size** prescribed, a decision is made whether to split or not to split the node.

3. Decision Tree Algorithms

ID3 (Iterative Dichotomiser 3)

- Uses **Information Gain** to split nodes.
- Tends to favor features with more levels (bias issue).

C4.5

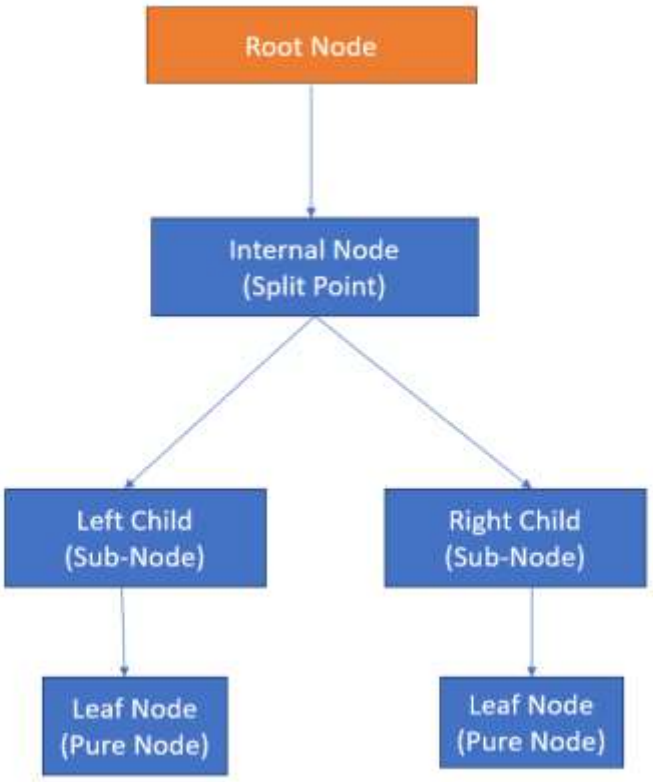
- An improvement over ID3.
- Uses **Gain Ratio** to address bias.
- Handles both categorical and continuous features.
- Manages missing values.

C5.0

- An industry-standard upgrade to C4.5.
- Faster and more memory-efficient.
- Better accuracy.
- Available in R (C5.0 function from the C50 package).

CART (Classification and Regression Trees)

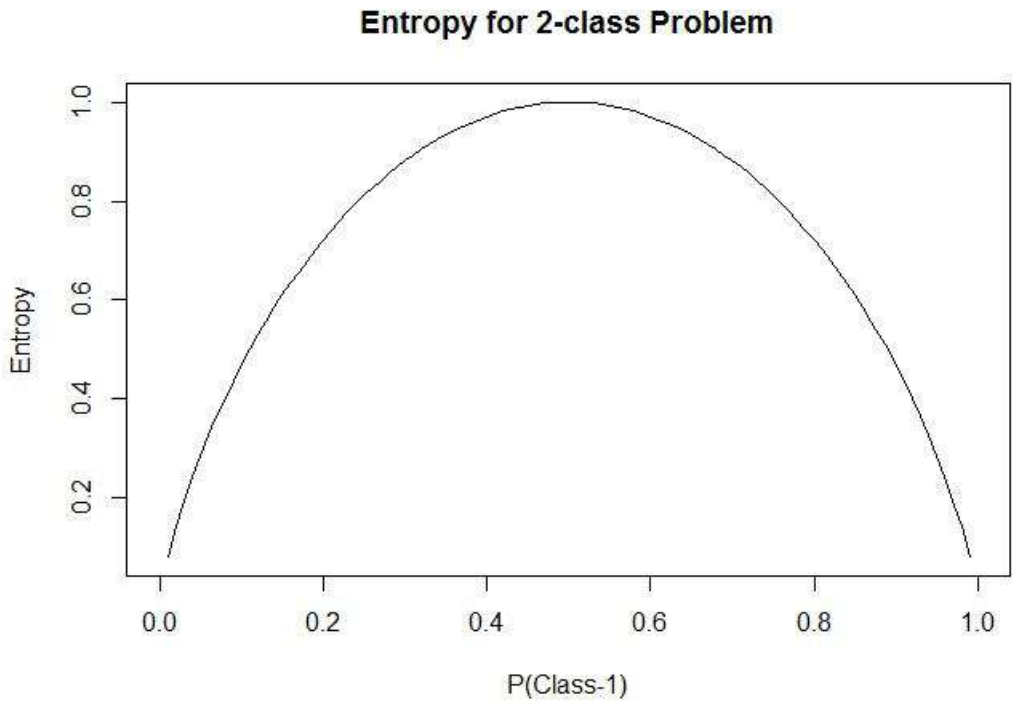
- Uses **Gini Index** for classification and **variance reduction** for regression.
- Produces only binary splits (two-way splits).



Entropy :

Entropy is defined as the average number of binary digits needed to communicate information via a message as a function of the probabilities of the different symbols used. When the symbols or components of a system have equal probabilities, there will be high degree of uncertainty, but the entropy will be lower when one symbol is far likelier than the others. This has led the entropy as measure of node purity. In the binary class problem, entropy is defined by

$$Entropy = -p \log_2 p + (1 - p) \log_2 p$$



Entropy for Two-class Problem

The definition of entropy may be extended to the case where there are c classes as,

$Entropy = -\sum_{k=1}^{c} p_k \log_2 p_k$

Information gain

The ID3 algorithm uses the weighted entropy reduction, which is also known as the information gain, as the splitting criterion. It is given by:

$Information\ Gain = Entropy_{initial} - \sum_{i=1}^{p} \frac{n_i}{n} = Entropy$

This is not a good splitting criterion, since it suffers from selection bias. This algorithm tends to favour categorical variables with large number of values to continuous variable with linear range of values. To combat this, *information gain ratio* is used in an improved version of ID3 algorithm, called *C4.5*. Information gain ratio is a normalized version of information gain, normalization is with respect to a quantity known as the *spilt information value*. This in turn represents the potential increase in information that we can get just by the size of the partitions themselves. A high spilt information value occurs when we have evenly sized partitions and a low value occurs when most of the data values are concentrated in small number of the partitions. In summary, we have:

Information Gain Ratio = $\frac{Information\ Gain}{Split\ Information\ value}$

Split information value = $\sum_{i=1}^{p} \frac{n_i}{n} \log_2\left(\frac{n_i}{n}\right)$

Gini Index

The aim of a classification tree is to partition the data to get child nodes that are as homogeneous as possible. This partition of the data items is done based on the values of a single feature that results in groups with minimum misclassification error. As there are several features available, we are faced with the problem of choosing a feature that result in minimum misclassification error. Gini index is one measure used to decide upon the purity of a node. For a two-class problem, it is defined as

$$p_1 (1 - p_1) + p_2 (1 - p_2)$$

where$p_1$and p_2 are class probabilities for Class-1 and Class-2, respectively. As there are only two classes by assumption, $p_1 + p_2 = 1$, and hence, the above index is equivalent to $2p_1p_2$. As $p1$ increases, p_2 decreases and vice versa, the Gini index is minimized when one of the class probabilities approaches zero. In other words, as one of the class probabilities tends to zero, the node becomes pure with respect to one of the classes. The Gini index gets maximized when $p_1 = p_2$. In this case, the node is least pure.

Need for Decision Making

As a part of their day to day work, different organizations such as business firms, banks, hospitals, research institutes and so on collect huge amounts of data. These organizations require analysis of the data so collected in a timely manner either for making important decisions, or to discover important patterns in the data unseen or unknown earlier. The important decisions involved in a particular situation may be as follows: in the case of a bank, the important decision may be, based on the parameter as collected by the bank, whether to sanction a loan to a prospective customer or to offer a credit card, or to issue an insurance policy and so on; in the case of a medical situation, a doctor has to decide whether the patient has a particular disease on the basis of the physical symptoms observed and the reports of the medical tests conducted on the patient; on the basis of the data collected on a patient or doctor may also be faced with making several decisions such as: whether to treat the patient as in-patient or out-patient, whether to conduct an operation or to prescribe medicines only and the list goes on. In a business situation, a marketing manager, having the huge number of customer records describing large number of customer records describing large number of features of each customer, is faced with how to segment the customer database for targeting a particular marketing campaign. There are quite a large number of situations, one is required analyse large amounts of data quickly and correctly, as these both factors may have an influence on their business.

6. Sources of Solution Methods

Decision trees offer simplicity, interpretability, and efficiency. Key algorithm families include:

- **ID3, C4.5, C5.0** – developed by Ross Quinlan.
- **CART** – developed by Breiman et al.

To further improve performance, **ensemble methods** like Bagging, Boosting, and Random Forests build multiple trees and combine their predictions for improved accuracy and robustness.

Here's an example of a decision tree using a dataset:

Dataset:

GPA	Test scores	Extracurricular	Admission
25	50000	700	Yes
30	60000	800	Yes
35	70000	900	Yes
20	40000	600	No
40	80000	950	Yes
28	55000	750	Yes
32	65000	850	Yes
22	45000	650	No
38	75000	920	Yes
42	85000	980	Yes

- Entropy of the **target variable** (Admission)
- Information Gain for **GPA** and **Test Scores** (after binning/simplifying)
- Identify best feature for root split

Step 2: Target Variable Entropy (Admission)

There are **10 samples**:

- **Yes = 8**
- **No = 2**Using entropy formula: entropy(s) =-$p_1 log_2(p_1)$-$p_{2log_2(p_2)}$

=-($\frac{8}{10}$)log_2($\frac{8}{10}$)- ($\frac{2}{10}$)log_2($\frac{2}{10}$)

=-0.8log_2(0.8)-0.2log2(0.2)

=-0.8(-0.3219)-0.2(-2.3219)

= 0.2575+0.4644

=0.7219

Step 3: Binning GPA into Categories

To simplify calculation, bin GPA into:

- **Low**: GPA ≤ 30
- **Medium**: 30 < GPA ≤ 35
- **High**: GPA > 35

GPA	GPA Bin	Admission
25	Low	Yes
30	Low	Yes
35	Medium	Yes
20	Low	No
40	High	Yes
28	Low	Yes
32	Medium	Yes
22	Low	No
38	High	Yes
42	High	Yes

Counts:

- **Low (5)**: Yes = 3, No = 2
- **Medium (2)**: Yes = 2, No = 0
- **High (3)**: Yes = 3, No = 0

Step 4: Entropy of GPA Bins

For Low (3 Yes, 2 No):

Entropy= -($\frac{3}{5}$) $log_2 \frac{3}{5}$ -($\frac{2}{5}$) $log_2 \frac{2}{5} \approx -0.6(-0.7369) - 0.4(-1.3219)$

=0.4421+0.5288=0.9709

For Medium (2 Yes):

Entropy = 0 (pure node)

For High (3 Yes):

Entropy = 0 (pure node)

Step 5: Weighted Average Entropy After GPA Split

InfoGPA=$\frac{5}{10}$0.9709+$\frac{2}{10}$0+$\frac{3}{10}$ 0 = 0.5·0.9709 = 0.4854

Step 6: Information Gain from GPA Split

Gain(GPA)=Entropy(before)−Entropy(after)=0.7219−0.4854=0.2365

Step 7: Split by Test Scores (Simplified)

Bin **Test Scores** into:

- **Low**: < 750
- **High**: ≥ 750

Test Score	Category	Admission
700	Low	Yes
800	High	Yes
900	High	Yes
600	Low	No
950	High	Yes
750	High	Yes
850	High	Yes
650	Low	No
920	High	Yes
980	High	Yes

Counts:

- **Low (3)**: Yes = 1, No = 2 → Entropy = ?
- **High (7)**: Yes = 7, No = 0 → Entropy = 0

Entropy (Low): Entropy= $-(\frac{1}{3})\ log_2 \frac{1}{3}$ $-(\frac{2}{3})\ log_2 \frac{2}{3}$

≈ −0.333·(−1.585)−0.666·(−0.585)=0.5283+0.389

= 0.9173

Step 8: Weighted Entropy (Test Score Split):

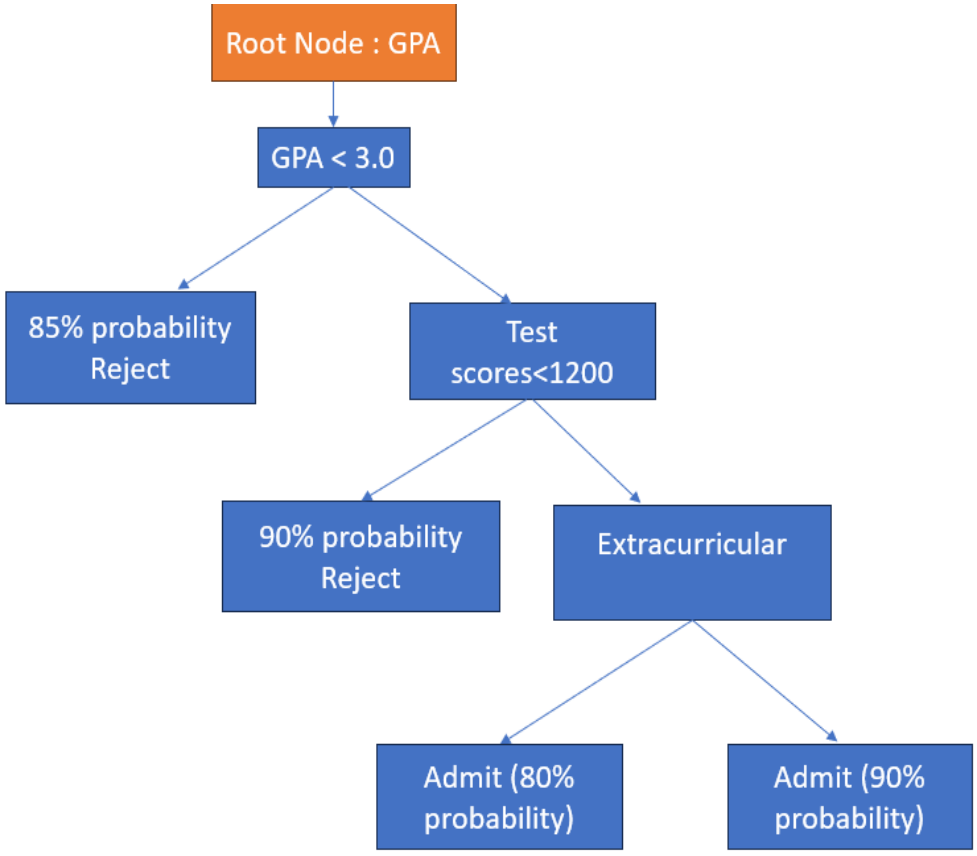
InfoTest=$\frac{3}{10}$(0.9173)+ $\frac{7}{10}$ 0 = 0.2752

Step 9: Information Gain (Test Score):

Gain(Test Score)=0.7219−0.2752 = 0.4467

Final Comparison

Feature	Info Gain
GPA	0.2365
Test Scores	0.4467



1. Root Node: GPA

- If GPA < 3.0, reject (85% probability)

- If GPA ≥ 3.0, proceed to Node: Test Scores

2. Node: Test Scores

- If Test Scores < 1200, reject (90% probability)

- If Test Scores ≥ 1200, proceed to Node: Extracurricular

3. Node: Extracurricular

- If Extracurricular, admit (80% probability)

- If No Extracurricular, admit (95% probability)

This decision tree considers four factors:

1. GPA (academic performance)

2. Test Scores (standardized test performance)

3. Extracurricular activities (leadership and community involvement)

The leaf nodes represent the predicted outcomes:

- Reject (85% or 90% probability)

- Admit (80% or 95% probability)

This example illustrates how decision trees work and provide insights into the relationships between variables.

C5.0 decision tree

There are many implementations of decision trees. One of the most popular implementation is C5.0 algorithm. This algorithm was due to Quinlan, which is an improved version of his earlier algorithm for decision tree called C4.5. The C4.5 algorithm itself is an improvement over the ID3 algorithm which is also due to Quinlan. In R this algorithm is available through the package C5.0.

This algorithm has become industry standard for developing decision trees. It can handle both small and large datasets. The learning process is highly automatic and handles both numeric and nominal variables. It can handle missing data efficiently. The results of C5.0 algorithm are easy to interpret.

Before we actually use the C5.0 ( ) function of C50 package we must install the package using the command

Conclusion:

Decision trees are powerful and intuitive tools for classification and regression tasks in machine learning and data analysis. Rooted in concepts from probability and statistics—such as entropy, information gain, and the Gini index—decision trees offer a structured way to model decisions under uncertainty. Their ability to handle both categorical and numerical data, manage missing values, and produce interpretable results makes them suitable for a wide range of applications, from medical diagnosis to business decision-making.

Algorithms such as ID3, C4.5, and C5.0 have advanced the development of decision trees by improving efficiency, reducing bias, and enhancing accuracy. Among these, the C5.0 algorithm stands out for its speed, scalability, and ease of use, making it an industry standard in predictive modeling. Despite some limitations, such as potential overfitting and sensitivity to complex data types, decision trees remain a foundational technique in data science. When combined with ensemble methods like Bagging, Boosting, and Random Forests, their performance can be significantly enhanced.

In summary, decision trees provide a statistically sound, interpretable, and effective approach to data-driven decision-making, making them indispensable in both academic and industry settings.

References:

1. Graham Williams (2011). Data Mining with Rattle and R-The Art of Exacting Data for Knowledge Discovery Springer
2. Daniel T. Larose Chantal D. Larose Data Mining and Predictive Analytics-Wiley Series on Methods and Application in Data Mining WILEY
3.Breiman L., Friedman, J. H., Olshen R. A., & Stone, C. J. (1984). Classification and Regression Trees
4. Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.
5. "Classification and Regression Trees" by Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen (1984) - Considered a foundational book on decision trees.
6. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei (2012) - Covers decision trees in the context of data mining.
7. "Pattern Recognition and Machine Learning" by Christopher M. Bishop (2006) - Includes a comprehensive chapter on decision trees.

8. "Machine Learning" by Tom M. Mitchell (1997) - Chapter 3: Decision Tree Learning (ID3 algorithm)
9. "Introduction to Machine Learning with Python" by Andreas C. Muller and Sarah Guido (2016) - Chapter 4: Decision Trees (ID3 and C4.5)
10. "Data Analysis and Decision Making" by S. Christian Albright, Wayne L. Winston, and Christopher Zappe (2017)
11. "Business Analytics: Data Analysis and Decision Making" by Robert Stine (2016)
12. "Data-Driven Decision Making" by Peter R. Schuh (2014)