Car Evaluation Decision Tree

Devcharan K
Indian Institute of Technology, Madras

***Abstract*—Decision tree classifier is one of the predictive modelling approaches used in statistics and machine learning. It uses a decision tree to go from observations about an item (represented in the branches) to conclusions about the item's target value. In this paper, data about car safety evaluation was analyzed using the Decision Tree classifier. The features were selected in two different methods- Gini index and Entropy criterion. The classifier is used to predict the safety of a given car based on its features.**

I. Introduction

Cars are essentially part of our regular day to day life. Car safety is extremely important because the roads which we drive on are much more dangerous than we think. When an individual considers buying a car, there are numerous aspects that could influence their choice on which kind of car they are keen on. The aim of this paper is to be able to predict the safety level of the car with the data provided.

A Decision Tree algorithm is one of the most popular machine learning algorithms. It uses a tree-like structure and their possible combinations to solve a particular problem. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes 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.

In this paper, the data about cars such as price, maintenance cost, number of doors, seating capacity, storage space is considered for evaluation. The aim is to classify the car based on its safety.

In Section II, Decision Tree and its terminology and techniques used are explained. In Section III, the problem and the method to solve it is shown. In Section IV, the concepts used to prepare the model is evaluated. Finally the inferences of the analysis are presented in Section V.

II. Decision Tree

In a Decision Tree algorithm, there is a tree-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from the root node to leaf node represent classification rules.

Terminology:

- Root node: It represents the entire population or sample. This further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision node: When a sub-node splits into further sub-nodes, then it is called a decision node.
- Leaf or Terminal node: Nodes that do not split are called Leaf or Terminal nodes.
- Pruning: When sub-nodes of a decision node are removed, the process is called pruning. It is the opposite process of splitting.
- Branch or Subtree: subsection of an entire tree is called a branch or sub-tree.
- Parent and Child node: A node, which is divided into sub-nodes is called the parent node of sub-nodes where sub-nodes are the children of a parent node.

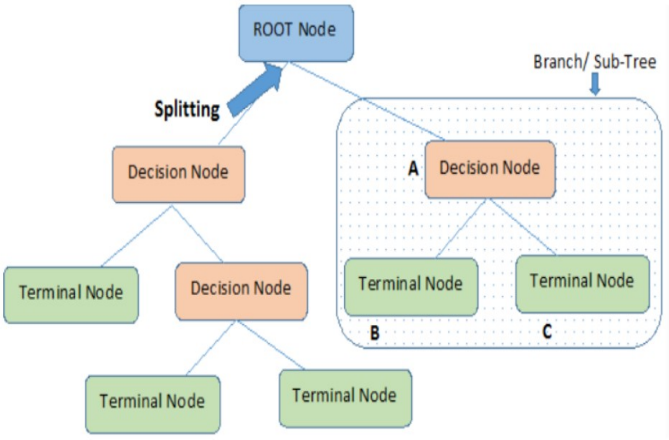

Fig. 1. Decision Tree Terminology

The Decision-Tree algorithm is one of the most frequently and widely used **supervised machine learning algorithms** that can be used for both classification and regression tasks. The intuition behind the Decision-Tree algorithm is as follows:

- For each attribute in the dataset, the Decision-Tree algorithm forms a node. The most important attribute is placed at the root node.
- For evaluating the task in hand, we start at the root node and we work our way down the tree by following the corresponding node that meets our condition or decision.

- This process continues until a leaf node is reached. It contains the prediction or the outcome of the Decision Tree.

A. Attribute Selection Measures

The primary challenge in the Decision Tree implementation is to identify the attributes which are to be considered as the root node and each level. This process is known as the **Attribute selection**. There are different attribute selection measures to identify the attribute which can be considered as the root node at each level. Two famous attribute selection measures are

- Information gain
- Gini Index

1) Information Gain: By using information gain as a criterion, the aim is to estimate the information contained by each attribute. To understand the concept of Information Gain, another concept called Entropy is necessary.

Entropy:

Entropy measures the impurity in the given dataset. In Physics and Mathematics, entropy is referred to as the randomness or uncertainty of a random variable X. In information theory, it refers to the impurity in a group of examples. **Information gain** is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.

$$Entropy = \sum_{i=1}^{c} -p_i * log_2(p_i) \tag{1}$$

Here, c is the number of classes and p_i is the probability associated with the i^{th} class.

The ID3 (Iterative Dichotomiser) Decision Tree algorithm uses entropy to calculate information gain. So, by calculating decrease in entropy measure of each attribute we can calculate their information gain. The attribute with the highest information gain is chosen as the splitting attribute at the node.

2) Gini Index: Another attribute selection measure that **CART (Categorical and Regression Trees)** uses is the Gini index. It uses the Gini method to create split points.

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{2}$$

Here, again c is the number of classes and p_i is the probability associated with the i^{th} class.

Gini index says, if one randomly selects two items from a population, they must be of the same class and probability for this is 1 if the population is pure.

III. The Problem

The car details such as price, maintenance cost, seating capacity, luggage space etc is considered. The aim of this exercise is to classify the cars based on their safety level which can be analyzed with the help of the other data provided. This model can also be used to predict the safety level of other cars provided the appropriate data.

A. Data Cleaning and Exploration

The dataset has 7 columns but no column names. The columns are named as follows: *'buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class'*. There are no missing values in the 1728 data samples. *'Class'* is the target variable and the remaining variables belong to the feature vector. All the variables are categorical in nature

IV. Model Evaluation

The model to be created is a Decision Tree classifier. The above mentioned techniques are used to create the model.

A. Feature Engineering

Feature Engineering is the process of transforming raw data into useful features that help us to understand our model better and increase its predictive power. In this case that is done by encoding the categorical variables using the package *'category_encoders'* which is used to convert categorical variables to numeric for ease of usage in modeling.

B. Test Train Split

The dataset is first split into two parts called training data and testing data. The training data is used to train the model and then the model is applied on the testing data to see its accuracy.

C. Decision Tree with criterion Gini Index

The model accuracy score with the training set is 0.7952. The model accuracy score with testing set is 0.7933.

The two values are quite comparable, there is no sign of overfitting.

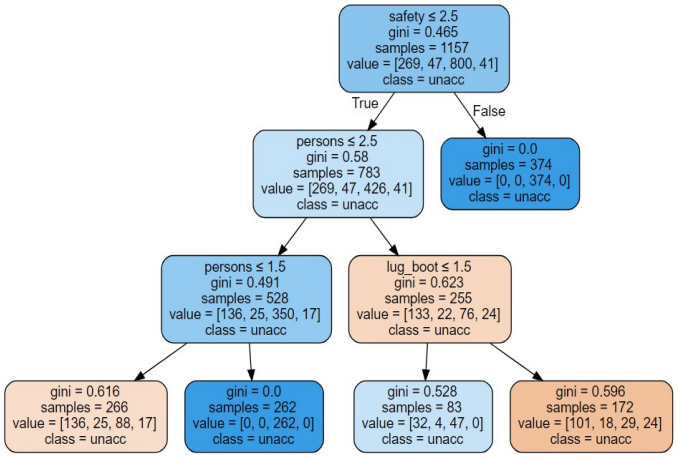

Fig. 2. Decision Tree with criterion gini index

D. Decision Tree with criterion Entropy

The model accuracy score with the training set is 0.7416. The model accuracy score with the testing set is 0.7758.

The two values are quite comparable, there is no sign of overfitting.

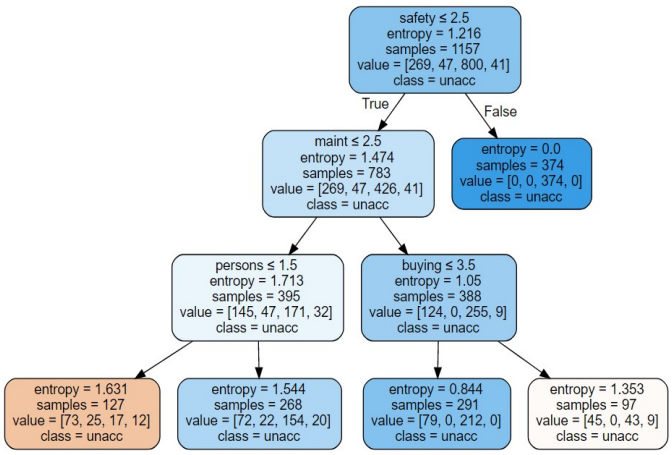

Fig. 3. Decision Tree with criterion Entropy

E. Confusion Matrix

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give a clear picture of classification model performance and the types of errors produced by the model. It gives a summary of correct and incorrect predictions broken down by each category.

The confusion matrix with gini index is

$$\begin{bmatrix} 102 & 0 & 13 & 0 \\ 17 & 0 & 5 & 0 \\ 59 & 0 & 351 & 0 \\ 24 & 0 & 0 & 0 \end{bmatrix}$$

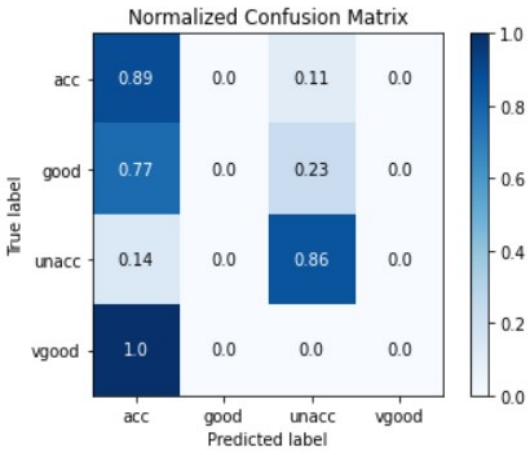

Fig. 4. Confusion matrix for Decision tree with Gini Index criterion

The confusion matrix with entropy criterion is

$$\begin{bmatrix} 59 & 0 & 56 & 0 \\ 11 & 0 & 11 & 0 \\ 26 & 0 & 384 & 0 \\ 16 & 0 & 8 & 0 \end{bmatrix}$$

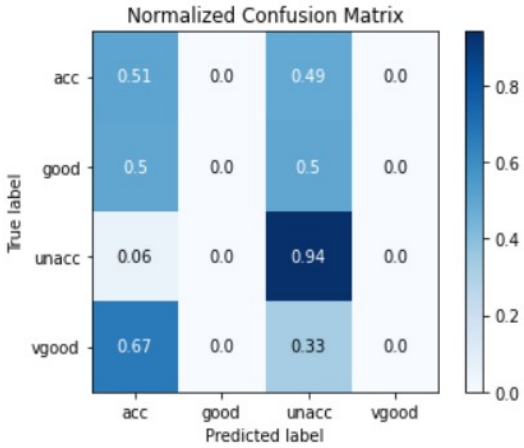

Fig. 5. Confusion matrix for Decision tree with Entropy criterion

V. Inferences

The decision tree classifier to predict the safety of a car yields a very good performance as indicated by the model accuracy score of 79%. There is no sign of over-fitting in both the cases of decision trees implemented. The confusion matrix yields a very good performance.

VI. Conclusion

The Decision Tree Classifier model is a very powerful classifier as indicated by the accuracy scores above. The model with 'gini index criterion' had an accuracy score of 79.52% with training data and 79.33% with testing data.
Similarly, the model with 'entropy criterion' had an accuracy score of 74.16% with training data and 77.58% with testing data.

References

[1] A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, 2011, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826.

[2] P. Tu and J. Chung, "A new decision-tree classification algorithm for machine learning," Proceedings Fourth International Conference on Tools with Artificial Intelligence TAI '92, 1992, pp. 370-377, doi: 10.1109/TAI.1992.246431.

[3] Aurelien Geron,"Hands on Machine Learning with Scikit-Learn and Tensorflow", 2017.

[4] "Decision tree" Wikipedia [Accessed on 30 October] Available: https://en.wikipedia.org/wiki/Decision_tree

[5] "Entropy (Information Theory)" Wikipedia [Accessed on 30 October] Available: https://en.wikipedia.org/wiki/Entropy_(information_theory)

[6] Decision Tree Classification in Python Available: https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[7] Evaluating Machine Learning Models using Hyperparameter Tuning Available: https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/

[8] Decision Trees in Python with Scikit-Learn Available: https://stackabuse.com/decision-trees-in-python-with-scikit-learn/