Assignment 4: Classification Using Decision Trees: A Comprehensive Analysis and Evaluation

Boneshwar V K
Department of Biotechnology
Indian Institute of Technology, Madras
bs20b012@smail.iitm.ac.in

***Abstract*—This paper provides a comprehensive analysis and evaluation of vehicle classification utilizing decision trees. The study focuses on the classification of cars into distinct purchase categories based on a diverse set of influencing factors. We delve into the mathematical foundations of the decision tree model and elucidate the intricacies of its construction process. Furthermore, the research scrutinizes the model's key parameters and assesses its performance using a dedicated dataset. The results offer valuable insights into the effectiveness of decision trees in classifying vehicles, providing a practical framework for decision-making in the automotive industry.**

***Index Terms*—decision trees, probabilistic modeling**

I. Introduction

Classification is a fundamental task in machine learning, involving the categorization of data points based on their intrinsic characteristics. This process enables algorithms to discern patterns and relationships within the data, facilitating accurate predictions or decisions for new, unseen data. It finds applications in diverse fields.

Within computer programming, trees serve as a fundamental data structure, and this paradigm extends to the realm of data science. Decision trees, a prevalent application of this structure, are particularly adept at handling categorization tasks. Operating within a supervised learning framework, decision trees split each node based on specific predictor variables, providing an output variable as we progress down the tree. This paper focuses on comprehending decision trees through their application in addressing classification challenges, specifically in evaluating automobiles. This evaluation process takes into account various metrics such as pricing, door count, and safety features, leveraging decision trees to categorize these vehicles into distinct classes.

This paper aims to provide a comprehensive analysis and evaluation of the vehicle classification process using decision trees. It explores the mathematical foundations and construction principles of decision trees, emphasizing their utility in solving real-world problems. The study delves into the complexities of classifying cars into purchase categories based on a diverse set of influencing factors. Furthermore, the research investigates critical model parameters and assesses its performance using a dedicated dataset. The results offer valuable insights into the effectiveness of decision trees in classifying vehicles, providing a practical framework for decision-making in the automotive industry.

II. Decision Trees

A. Why Decision Trees?

When confronted with classification tasks, the decision tree algorithm exhibits distinct advantages over its counterparts. Unlike linear logistic regression, which assumes linear relationships between features, decision trees can handle complex, non-linear relationships in the data without necessitating feature engineering. This is particularly pertinent in the assessment of cars, where intricate interactions between attributes may not adhere to linear assumptions. Additionally, decision trees provide inherent interpretability, crucial in practical applications like the automotive industry, where stakeholders require transparent insights into the factors driving classification decisions. The following figure 1 illustrates an simple example for purchasing a car using the decision tree algorithm.

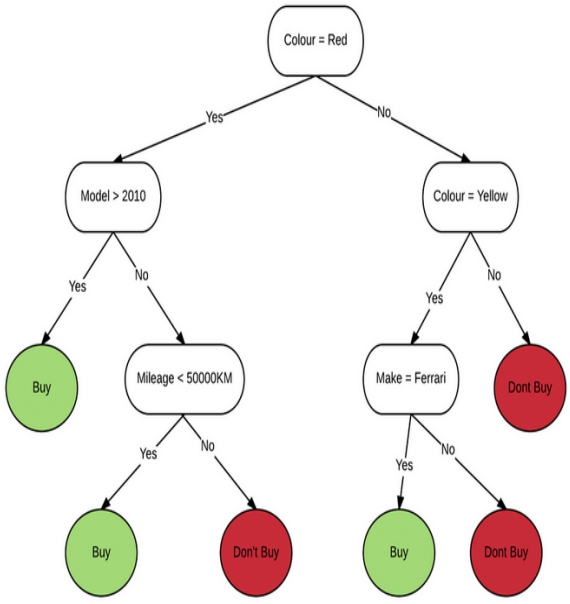

Fig. 1. Example of decision tree on car purchase

Furthermore, in comparison to the Naive Bayes classifier, decision trees do not impose strict independence assumptions among features. While Naive Bayes simplifies by assuming feature independence, decision trees consider interdependencies among attributes, making them more adept at capturing nuanced relationships in the data. In the context of vehicle

classification, where attributes like safety features, pricing, and fuel efficiency may interact in subtle ways, this flexibility allows decision trees to offer more accurate and nuanced predictions. This enhanced modeling capability is particularly advantageous when dealing with real-world datasets that often exhibit complex interrelationships between attributes.

B. Types of Decision Trees

Decision trees come in various types, each tailored for specific use cases.
There are 2 types of Decision trees:
1) Regression Trees
2) Classification Trees

The most common types include Classification Trees and Regression Trees. Classification Trees are adept at handling categorical target variables, making them invaluable in scenarios like sentiment analysis, where the goal is to classify data into distinct categories. On the other hand, Regression Trees are tailored for predicting numerical values, making them well-suited for tasks such as price prediction in the context of real estate or financial markets. Additionally, Ensemble Methods like Random Forests, which consist of multiple decision trees, excel in tasks requiring high predictive accuracy, like medical diagnosis. They aggregate the decisions of individual trees, reducing overfitting and enhancing robustness. These different types of decision trees offer a versatile toolkit for various machine learning applications, ensuring adaptability across a wide range of data-driven challenges.

C. Understanding of Decision Trees: Mathematical approach

This section deals with the mathematical understanding of Decision Trees and throws some light on how the algorithm outperforms linear regression, logistic regression and other machine learning algorithms especially classifying our dataset.

The mathematical underpinning of a decision tree algorithm revolves around the delineation of data splits and decision-making at each node. Typically, decision trees are constructed through a recursive, divide-and-conquer methodology. At every node, a pivotal step involves the discernment of the optimal feature (referred to as F) and its corresponding threshold or value (referred to as V) to partition the data into distinct subsets. This process aims to maximize a predefined criterion.

In the context of classification, various criteria like Information Gain, Gini Impurity, or Cross-Entropy are commonly employed. This paper particularly delves into the utilization of Gini Impurity as the pivotal criterion for splitting. Gini Impurity serves as a metric for the disorder or impurity within a dataset. Within the framework of decision trees, it quantifies the efficacy of a potential split in classifying the data into distinct categories. Formally, the Gini impurity for a dataset D is defined as follows:

$$G = 1 - \sum_{i=1}^{k} p_i^2 \tag{1}$$

Here p represents the probabilities or ratio of occurrence of class i among k classes in the dataset D.

The Gini impurity scales from 0 to 0.5, with 0 meaning that the dataset is completely pure and contains only elements that belong to one class. With examples uniformly distributed across all classes, a dataset with a Gini impurity of 0.5 is said to be perfectly impure.

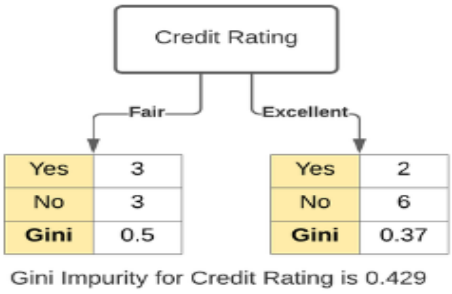

Fig. 2. Illustration of Gini Calculation

The figure2 illustrates how Gini calculations are done explicitly. One node of a decision tree is shown in the figure above. Fair is the credit rating on the left leaf, while Excellent is the credit rating on the right leaf. The Gini impurity can be written as follows on the left leaf $(Credit\ rating = Fair)$:

$$G = 1 - (p_{\text{yes}})^2 - (p_{\text{no}})^2 \tag{2}$$

$$G = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5 \tag{3}$$

The Gini impurity for the right leaf node was computed as 0.37. To determine the impurity of the Credit Rating node, we calculate a weighted average of the two impurities based on the sample sizes. The Gini impurity for Credit Rating is computed as

$$G_{\textit{Credit Rating}} = (0.5 \times \frac{6}{14} + 0.37 \times \frac{8}{14} = 0.429) \tag{4}$$

This process is applied to all features in the dataset. Subsequently, the feature node with the lowest Gini impurity is considered. This iterative process continues until the maximum depth of the tree is reached.

III. The Data

A. Data and Problem Overview

This section comprises the overview of the problem and our dataset, and the reason why we chose decision tree. The dataset encompasses a variety of cars, each characterized by specific attributes intended for safety categorization. The following table illustrates how the data is distributed.

These attributes encompass details like buying price, maintenance cost, number of doors, seating capacity, luggage boot size, and the estimated safety level of the vehicle. Notably, the dataset exclusively comprises categorical columns, wherein attributes related to purchase price, maintenance cost, and estimated safety level are classified into distinct categories: $very\ high$, $high$, $medium$, and low. The target condition of the cars is further categorized into four specific classes:

TABLE I
KEY INSIGHT OF THE DATA FEATURES

Feature	Key	Variable Type
Buying Price	vhigh, high, med, low	Predictor
Cost of Maintenance	vhigh, high, med, low	Predictor
Number of Doors	2, 3, 4, 5, more	Predictor
Capacity (Persons)	2, 4, more	Predictor
Size of Luggage Boot	small, med, big	Predictor
Estimated Safety	low, med, high	Predictor
Evaluation	unacc, acc, good, vgood	Output

very good, *good*, *acceptable*, and *unacceptable*. This comprehensive dataset forms the basis for our analysis and subsequent decision tree modeling.

B. Visualizations

Under this section we visualize the key insights of the data, how the data columns and features are related to each other, *i.e.* data correlation. Further the correlated data are being visualized. It is crucial to visualise a dataset before using a machine learning model. It is crucial to visualise a dataset before using a machine learning model. A comprehensive understanding of the structure and properties of the data is provided by data visualisation techniques including scatter plots, histograms, and heatmaps. It helps in locating hidden patterns, comprehending the connections between variables, and spotting outliers.

Only categorical variables, often known as discrete data, are present in our data. Looking at bar graphs, which show the counts of each variable in the data, is the best approach to visualise categorical variables. The following plots throws some light on the *correlation of the features* 3, relation between *buying price and target*4, and finally, *maintenance cost and quality*5.

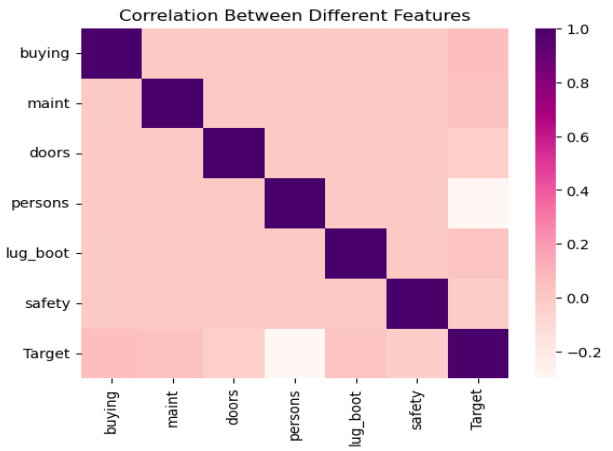

Fig. 3. Correlation Of the Data

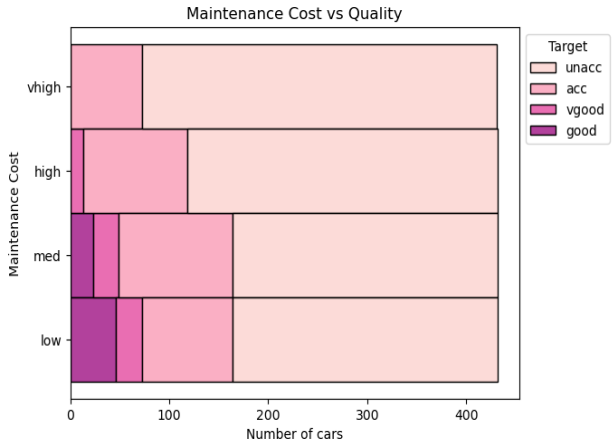

Fig. 4. Relation between Maintainence and Quality

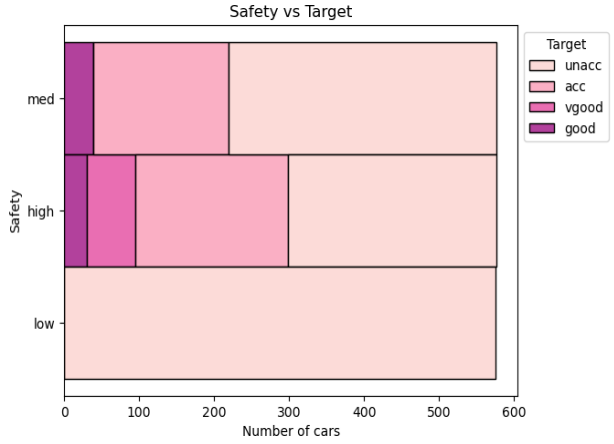

Fig. 5. Relation between Safety and Target

IV. THE PROBLEM

A. Training and Hyperparameter Tuning

In the context of decision tree models, hyperparameter tuning is a critical step in optimizing their predictive performance. Parameters such as maximum depth, minimum samples per leaf, and splitting criteria profoundly influence the tree's structure and, consequently, its predictive accuracy. Through systematic experimentation and validation techniques like k-fold cross-validation, practitioners aim to identify the most effective combination of hyperparameters. This process ensures that the decision tree model generalizes well to new data and avoids overfitting. Fine-tuning these hyperparameters is imperative for deploying accurate and reliable decision tree models across diverse applications and datasets.

The decision tree's maximum depth, a crucial hyperparameter, was manually set in the previous section. This section employs cross-validation and grid search to fine-tune it.

1) Cross Validation: Cross-validation partitions the dataset into k subsets for systematic model training and evaluation. The common k-fold method involves iterative training and

testing, with performance metrics computed for each iteration. The average of these scores provides an overall assessment.

For the decision tree, we initially considered a range of depths from 1 to 25. After Grid Search, optimal parameters were obtained. This hyperparameter tuning significantly enhanced model accuracy, from 95.76% to 97.31%. Notably, increasing the maximum depth improves accuracy, but excessive depth can lead to overfitting and reduced real-world performance.The decision tree 6 has been visualized below.

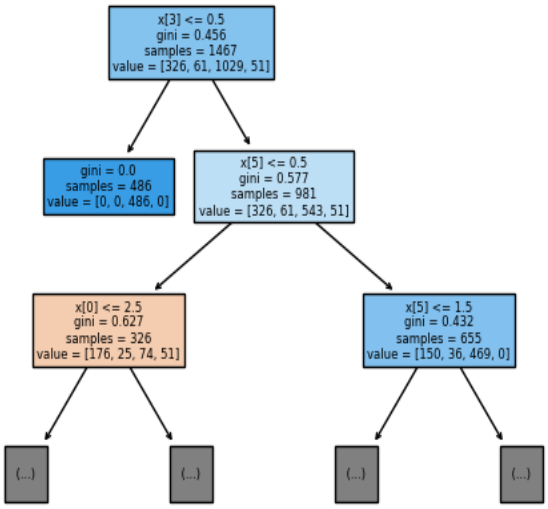

Fig. 6. Decision Tree

B. Key Insights and Observations: Unraveling the Decision Tree's Performance

A comprehensive accuracy table is presented, encapsulating the model's performance metrics across different configurations. This table serves as a valuable reference for model selection and hyperparameter tuning. Furthermore, detailed parameter tables provide insights into the specific settings chosen for the final decision tree model. This rigorous analysis ensures that our model is finely calibrated to deliver accurate and reliable results.

TABLE II
BEST PARAMETERS AFTER GRID SEARCH

Parameter	Best Score
Splitting Criterion	Gini Index
Maximum tree Depth	25
Splitter	Best

Once we obtain the best parameters, we can now fit our model using these parameters. The below table **??** summarizes the accuracies of the model.

TABLE III
ACCURACY OF MODEL

Data Set	Accuracy
Train Data Set	100.000%
Test Data Set	97.307%

V. CONCLUSION

The fundamental importance of decision trees in the field of machine learning has been examined in this research, in conclusion. Decision trees provide a clear and understandable method for performing classification and regression tasks, making them an important resource for deciphering complex data and resolving practical issues. We've spoken about how they're made, how hyperparameters work, and how they're used in a variety of fields.

Different ensemble approaches, including Random Forests, Gradient Boosted Decision Trees, AdaBoost, etc., were developed as a result of decision trees. XGBoost, also known as extreme gradient boosting The ultimate goal of gradient boosting has been realised in a scalable and effective manner. We have achieved an accuracy around 97.3% which outperforms linear regression, logistic regression and naive bayes theorem.

REFERENCES

[1] Z. Wang, M. Agung, R. Egawa, R. Suda and H. Takizawa, "Automatic Hyperparameter Tuning of Machine Learning Models under Time Constraints," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4967-4973, doi: 10.1109/BigData.2018.8622384.

[2] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860040.

[3] S. Pathak, I. Mishra and A. Swetapadma, "An Assessment of Decision Tree based Classification and Regression Algorithms," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2018, pp. 92-95, doi: 10.1109/ICICT43934.2018.9034296.