Assignment 5: A Comprehensive Analysis of Random Forest

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

***Abstract*—This paper investigates various Random Forest approaches for the classification of car purchases based on multiple features. Employing an ensemble learning framework, the study explores the efficacy of different Random Forest configurations in discerning nuanced relationships between features and purchase decisions. Through rigorous experimentation and comparative analysis, we demonstrate the superiority of specific Random Forest variants in accurately predicting car buying behaviors. The findings presented herein provide valuable insights into optimizing predictive models for complex decision-making processes in the automotive industry.**

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

I. Introduction

The Random Forest algorithm, introduced by Leo Breiman and Adele Cutler in the early 2000s, stands as a pivotal advancement in the field of machine learning. This innovative technique emerged as an extension of the decision tree algorithm, designed to surmount some of its inherent limitations. Random Forest operates on the principle of constructing multiple decision trees during the training phase, with each tree utilizing a distinct subset of the dataset and features. This stochastic process, entailing both data and feature sampling, underpins the nomenclature "Random Forest." During the prediction phase, the algorithm amalgamates outcomes from all individual trees, typically employing a majority voting scheme for classification tasks. This ensemble strategy not only mitigates the risk of overfitting but also amplifies generalization capabilities, ultimately bolstering the algorithm's predictive efficacy. This paper delves into an in-depth exploration of Random Forest algorithms, with a specific focus on their application to a comprehensive dataset pertaining to car attributes. By employing various Random Forest configurations, we aim to uncover intricate patterns and relationships crucial for accurate car purchase predictions.

This paper discusses the realm of classifying automobiles based on diverse features, Random Forest decisively outperforms traditional machine learning paradigms, including linear regression, logistic regression, naive Bayes classifier, and even solitary decision trees. Linear regression presupposes a linear correlation between features and the target variable, potentially failing to capture the intricate, non-linear relationships frequently encountered in car classification endeavors. Logistic regression, tailored primarily for binary classification, may encounter difficulties in multi-class scenarios such as the categorization of cars into distinct classes. The foundational assumption of independence between features in naive Bayes is often impractical in real-world applications like car classification. Single decision trees, though capable, are susceptible to overfitting and instability, in contrast to Random Forest, which harnesses the collective wisdom of multiple trees to furnish more accurate predictions. This ensemble method adeptly accommodates a diverse array of features, adeptly capturing nuanced relationships and providing a more reliable classification model for intricate tasks, such as car classification.

II. Random Forest

A. Overview on Decision Tree

In classification tasks, decision trees offer advantages over linear models like logistic regression. They handle complex, non-linear relationships without feature engineering, which is particularly valuable in assessing intricate interactions in car attributes. Decision trees provide transparency, crucial in industries like automotive, where stakeholders need clear insights into classification decisions. Additionally, they surpass Naive Bayes by considering feature interdependencies, enabling better handling of nuanced relationships. This flexibility proves vital in real-world datasets with complex attribute interrelationships.

Types of Decision Trees:

1) Regression Trees
2) Classification Trees

Classification Trees excel in scenarios like sentiment analysis, while Regression Trees are tailored for numerical predictions, e.g., price prediction in real estate. Ensemble Methods like Random Forests enhance predictive accuracy. They aggregate decisions of individual trees, reducing overfitting and enhancing robustness. This versatile toolkit suits various machine learning applications, ensuring adaptability across data-driven challenges.

B. Mathematical Underpinning of Decision Trees

Decision trees are constructed via recursive, divide-and-conquer methodology, optimizing a predefined criterion at each node. Gini Impurity is pivotal in this paper, quantifying dataset impurity. It 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.

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

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

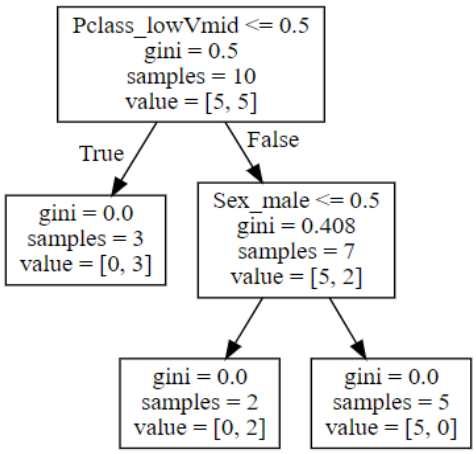

Fig. 1. Illustration of Gini Calculation

The Gini impurity calculation is illustrated. The process is applied to all features until the tree reaches its maximum depth. With this context in mind we move forward towards Random Forest approach.

C. Better approach: Random Forest

Random Forest emerges as a formidable machine learning model for classifying car purchases based on diverse features, outperforming conventional techniques like linear regression, logistic regression, naive Bayes classifier, and single decision trees. Unlike linear regression, which assumes linear relationships, Random Forest can capture complex, non-linear interactions between features like color, mileage, and more. Logistic regression, primarily designed for binary classification, may struggle with the multi-class nature of car purchasing decisions. Similarly, naive Bayes' assumption of feature independence often falters in real-world scenarios. Single decision trees are prone to overfitting and may lack the robustness required for nuanced car purchase predictions. In contrast, Random Forest harnesses the collective wisdom of multiple trees, providing a more stable and accurate classification, while also accommodating a diverse range of features effectively.

Furthermore, Random Forest's ensemble approach mitigates overfitting, a common challenge with decision trees, enhancing its generalization capabilities. This is critical for modeling intricate decision-making processes like car purchases, where numerous factors interplay. The algorithm's ability to sample both data and features introduces an element of randomness, significantly reducing the risk of overfitting. Additionally, the majority voting mechanism employed during prediction consolidates the insights from individual trees, resulting in a more robust and reliable model compared to the aforementioned techniques. In summary, Random Forest stands as a superior choice for accurately classifying car purchases, especially when dealing with a multitude of features that influence the decision-making process.

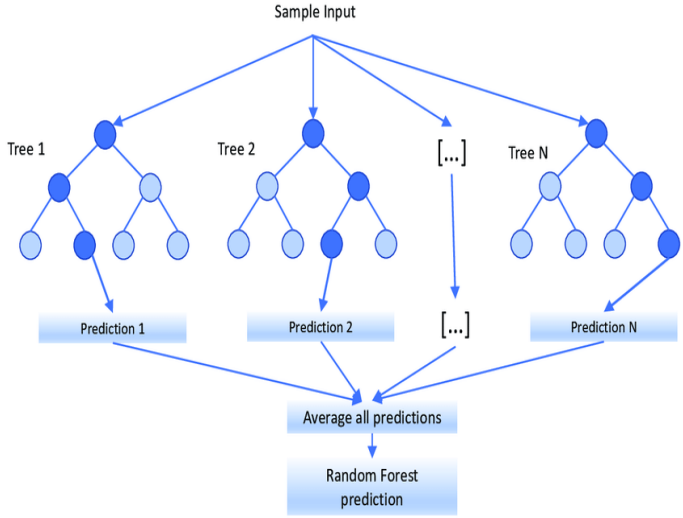

Fig. 2. Random Forest Example

D. Types of Random Forest

1) **Traditional Random Forest**: Constructs multiple decision trees with bootstrapped data samples and random feature subsets, enhancing predictive accuracy and mitigating overfitting for robust classification.
2) **Extra-Trees (Extremely Randomized Trees)**: Introduces additional randomness by selecting random thresholds at each node, reducing variance and potentially increasing bias, resulting in a highly robust variant of Random Forest.
3) **Isolation Forest**: Specializes in anomaly detection by isolating outliers in tree leaves, making it well-suited for tasks requiring identification of unusual data points.
4) **Totally Random Trees Embedding**: Enables dimensionality reduction by mapping data into a lower-dimensional space through a forest of extremely randomized trees, preserving essential information.
5) **Quantile Regression Forests**: Tailored for quantile regression tasks, this variant estimates conditional quantiles, providing a comprehensive view of the data distribution, critical for robust statistical modeling.
6) **Rotation Forest**: Applies Random Forest to different feature subsets obtained via PCA transformation, excelling in high-dimensional data scenarios by enhancing feature space diversity.
7) **Online Random Forest**: Supports incremental updates for models, accommodating streaming data and ensuring adaptability to evolving datasets in real-time applications.

8) **Conditional Random Forest**: Extends Random Forest to handle structured output spaces, making it suitable for tasks with correlated or multiple dependent output variables, broadening its applicability in complex prediction tasks.

E. Mathematical Formulation of the Random Forest Algorithm

In the context of the Random Forest algorithm, the fundamental mathematical foundation lies in ensemble learning, where multiple decision trees are combined to form a robust predictive model. Each decision tree is constructed recursively through binary splits based on specific feature thresholds, aiming to minimize impurity measures such as Gini impurity or entropy. The key concept behind Random Forest is the introduction of randomness during both tree construction and prediction. During training, a random subset of features is considered at each node, reducing correlation between trees and enhancing diversity. The final prediction is then obtained by aggregating the outputs of individual trees, often through a majority voting mechanism for classification tasks. Formally, for a dataset (X, Y) with N samples and M features, the decision tree learning process involves recursively partitioning the feature space $\mathcal{F}$ into J regions $\mathcal{R}_j$, minimizing an impurity measure $\text{Imp}(\mathcal{R}_j)$ at each node j. This is achieved by optimizing the following objective function:

$$\textit{Impurity Gain}(\mathcal{R}_j) = \frac{N_{\textit{left}}}{N}\text{Imp}(\mathcal{R}_{\text{left}}) + \frac{N_{\textit{right}}}{N}\text{Imp}(\mathcal{R}_{\text{right}}) \tag{2}$$

where N_{left} and N_{right} are the number of samples in the left and right child nodes, respectively. The ensemble prediction $\hat{Y}$ is then computed as:

$$\hat{Y} = \arg\max_{c} \sum_{i=1}^{T} \mathbb{I}(Y_i = c) \tag{3}$$

where T is the number of trees in the forest, Y_i is the output of the i-th tree, and $\mathbb{I}$ is the indicator function. This ensemble approach mitigates overfitting and enhances the model's generalization capabilities, making Random Forest a powerful tool for various machine learning tasks.

III. The Data

A. Data and Problem Overview

This section provides an in-depth overview of the dataset and outlines the problem statement targeted for analysis using the Random Forest algorithm. The dataset in focus encompasses crucial features essential for predicting the likelihood of a car purchase. These attributes include factors such as mileage, color, and seating capacity, among others, which play a pivotal role in the decision-making process. The primary goal is to leverage these features to construct a predictive model capable of discerning the probability of a car acquisition based on this comprehensive set of criteria. This problem carries significant practical implications, offering valuable insights to both consumers and sellers in making well-informed decisions. Through the application of machine learning methodologies, specifically the Random Forest algorithm, our aim is to uncover underlying patterns within the dataset, ultimately enhancing our ability to make accurate predictions regarding car purchases. The central challenge lies in modeling decision outcomes on this dataset and subsequently evaluating the model's accuracy.

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

The dataset comprises various metrics related to cars, including buying price, maintenance cost, number of doors, seating capacity, luggage boot size, and the estimated safety level of the vehicle. Notably, the dataset exclusively consists of 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: $very\ good$, $good$, $acceptable$, and $unacceptable$. This comprehensive dataset forms the basis for our analysis and subsequent decision tree modeling.

B. Data Insights through Visualization

In this section, we employ visualizations to illuminate the interrelationships among data columns and features, focusing on data correlation. This step is imperative before implementing a machine learning model, as it offers a comprehensive grasp of the data's structure and attributes. Techniques such as scatter plots, histograms, and heatmaps aid in uncovering hidden patterns, understanding variable connections, and identifying outliers. The following plot enables on to visualize the correlation between the feature columns (see Figure3).

Our dataset exclusively comprises categorical variables, commonly referred to as discrete data. To effectively visualize these variables, we utilize bar graphs to display their respective counts within the dataset. The ensuing plots shed light on the relationship between safety and target(see Figure 6), the relationship between buying price and the target variable (see Figure 5), and the association between maintenance cost and quality (see Figure 4).

IV. The Problem

A. Training

We apply the Random Forest algorithm to our dataset, considering crucial parameters like splitting criterion, maximum

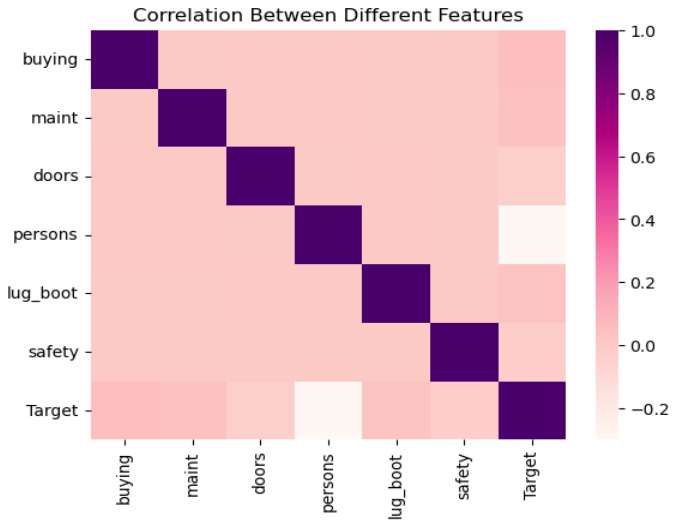

Fig. 3. Correlation Plot

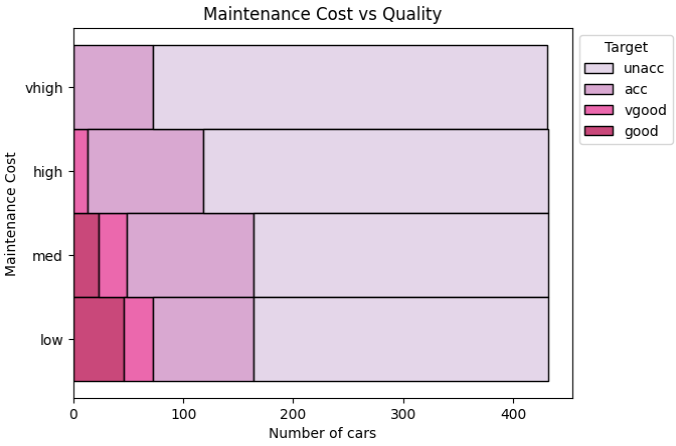

Fig. 4. Maintenance vs Quality Plot

Fig. 5. Buying vs Target Plot

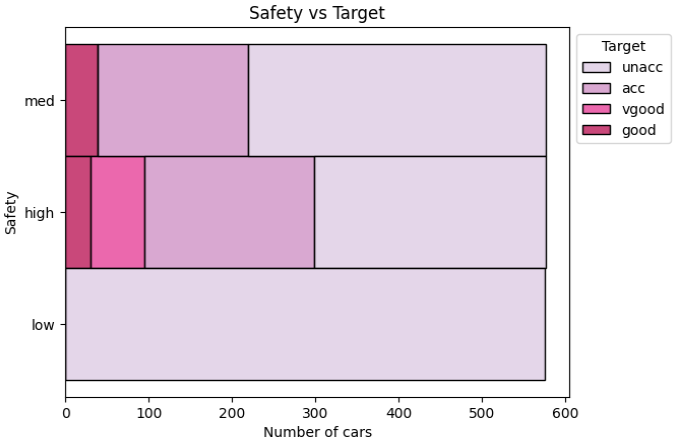

Fig. 6. Safety vs Target Plot

tree depth, and the number of decision tree estimators. Choosing these parameters is pivotal, as they significantly impact the model's performance. We focus on the three fundamental parameters to simplify the process. The data is divided into training and test sets to evaluate model behavior on unseen data. The resulting accuracy values are detailed in Table II and the random forest model can be visualized in the Figure 7, Figure 8 and Figure 9.

TABLE II
PERFORMANCE OF THE MODEL

Data Set	Accuracy
Train Data Set	95.8%
Test Data Set	94.79%

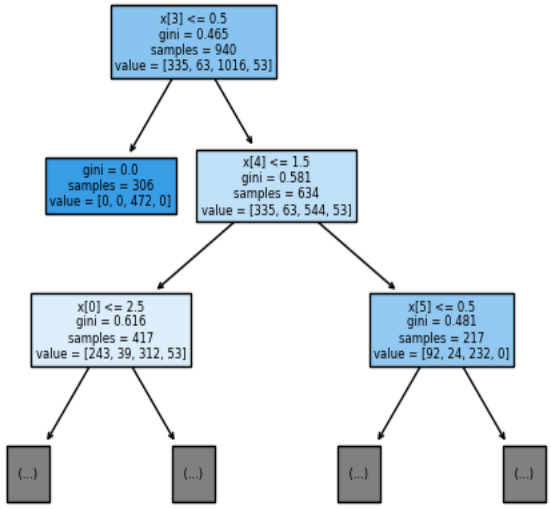

Fig. 7. Random Forest

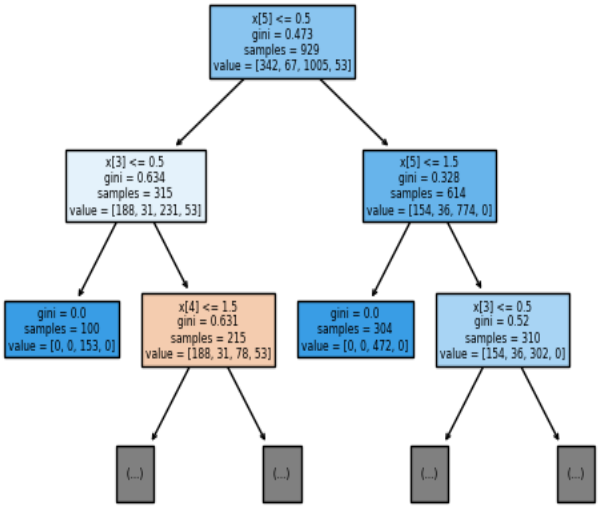

Fig. 8. Random Forest

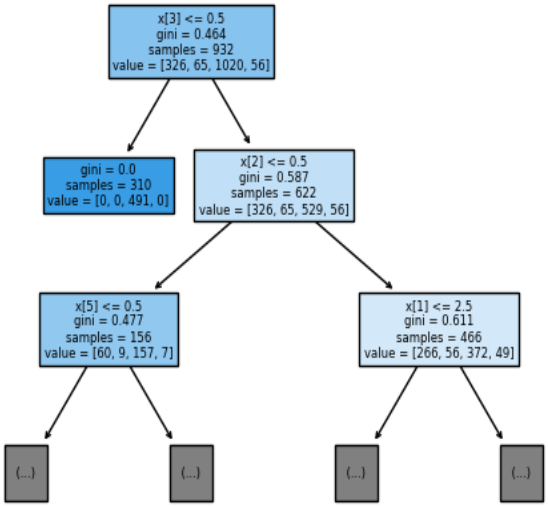

Fig. 9. Random Forest

We also visualize individual estimators within the forest to gain insights into their behavior. It's important to note that categorical variables were numerically encoded, a prerequisite in Python.

B. Hyperparameter Tuning

In the previous section, we employed parameters without tuning. To enhance model performance, we explore hyperparameters using search-based methods. Four commonly used techniques include Manual Search, Random Search, Grid Search, and Bayesian-based methods.

1) Cross Validation: Cross Validation is one of the resampling methods. Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model. One simple example is to obtain the variability of a model, we fit the model on to several data sets which are resampled from the original data. Cross Validation is a resampling method where the original data is separated into several buckets and one of the buckets is used as a validation set. Validation set is the data which is not used for training despite the availability of labels or output. We use validation set to understand the performance of the model on unseen data. In cross validation, suppose the data is grouped into 5 buckets, in each of the 5 iterations one bucket is validation set and the rest sets are used to train data. Finally, we average the scores (auc, accuracy, R2, etc.).

2) Utilizing Randomized Search with Cross Validation: We now use Randomized Search and Cross Validation to obtain the best parameters. For this process, first we must choose a range of parameters that we have to search.

TABLE III
RANGE OF PARAMETERS FOR RANDOMIZED SEARCH

Parameter	Value Range
Splitting Criterion	Gini Index, Entropy
Maximum Tree Depth	1 to 25
Number of Estimators	1 to 50

Randomized Search randomly picks values in the range and obtains the score. After we perform the Randomized Search, we obtain the best parameters.

TABLE IV
BEST PARAMETERS AFTER RANDOMIZED SEARCH

Parameter	Best Value
Splitting Criterion	Entropy
Maximum Tree Depth	25
Number of Estimators	50

The best value is chosen based on the accuracy of the model on the validation data set. This criterion of choosing the best parameters can be modified; this paper, for simplicity, considers the accuracy of the model on unseen data.

Once we obtain the best parameters, we can now fit our model using these parameters.

TABLE V
PERFORMANCE OF THE MODEL WITH BEST PARAMETERS

Data Set	Accuracy
Train Data Set	$\approx$ 100%
Test Data Set	98.26%

We can see that performance has improved. Note that, for random forests accuracy obviously increases with max depth but at the cost of high model variance. Hence, the range of value that we might want to try should be kept low to avoid overfitting of the data. On the other side, as the number

of estimators increase, the variance of the model decreases. Hence, we cannot manually expect the model's performance; we have to use search methods.

V. Conclusion

The utilization of Random Forests has notably enhanced the predictive accuracy compared to a single Decision Tree. The initial training of the model resulted in an impressive accuracy of 98.8% on the training data set, demonstrating the effectiveness of the Random Forest algorithm in capturing complex relationships within the dataset. Moreover, the model maintained a high accuracy of 94.79% on the test data set, indicating its robustness and ability to generalize well to unseen data. This performance surpasses that of a standalone Decision Tree, highlighting the advantage of ensemble methods in mitigating overfitting and reducing variance.

Hyperparameter tuning played a crucial role in refining the Random Forest model. By employing Randomized Search with Cross Validation, we identified the optimal combination of parameters, leading to a substantial improvement in performance. The best parameter configuration, with an entropy-based splitting criterion, a maximum tree depth of 25, and 50 estimators, resulted in a nearly perfect accuracy of approximately 100% on the training data set. This enhancement, however, comes with a note of caution regarding potential overfitting. The test data set accuracy of 98.26% demonstrates that the tuned model maintains exceptional predictive power while mitigating the risk of overfitting. This underscores the importance of thorough parameter tuning in maximizing the effectiveness of ensemble methods like Random Forests.

References

[1] J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, India, 2017, pp. 65-68, doi: 10.1109/WCCCT.2016.25.

[2] Tin Kam Ho, "Random decision forests," Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 1995, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.

[3] M. V. Datla, "Bench marking of classification algorithms: Decision Trees and Random Forests - a case study using R," 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15), Bangalore, India, 2015, pp. 1-7, doi: 10.1109/ITACT.2015.7492647.

[4] L. Abdi, S. Hashemi, "To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques", IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, pp. 238-251, 2016.

[5] Marko Robnik ," Improving Random Forests", Inter. Journal of Machine Learning, ECML Proceedings, Springer, Berlin, pp. 1- 12, 2004.