# Assignment 5: An Application of Random Forest

Pradyumn K S A
*MSc. in Data Science and Artificial Intelligence*
*Indian Institute of Technology, Madras*
Chennai, India
ge23c012@smail.iitm.ac.in

***Abstract*—This report explores the application of Random Forests, a powerful ensemble learning technique, in the context of car evaluation. Car evaluation, a complex multi-class classification problem, involves the assessment of various attributes and features to make informed decisions about the quality, safety, and desirability of different vehicle models. The aim of this application is to leverage the strengths of Random Forests to develop an accurate and robust predictive model for categorizing cars based on attributes such as price, safety ratings, maintenance costs, and more. Random Forests mitigate overfitting by constructing an ensemble of decision trees, each trained on a random subset of the data, and by selecting random subsets of features for each split. This inherent diversity enhances the model's ability to handle high-dimensional and noisy data while improving predictive accuracy.**

***Index Terms*—Decision Tree, Ensemble Learning, Correlation Matrix, Confusion Matrix, F1 Score, Classification Report, Estimators.**

## I. Introduction

Random Forests is a powerful and versatile machine learning algorithm widely used for both classification and regression tasks. This ensemble learning method operates on the principle of constructing multiple decision trees and combining their outputs to make more accurate predictions. It's very popular due to its robustness, scalability, and ability to handle high-dimensional datasets. The fundamental idea behind Random Forests lies in their ability to reduce overfitting and enhance the model's generalization by aggregating the results from multiple decision trees, creating a forest of diversified learners.

Random Forests are composed of several key technical aspects. Each tree in the forest is constructed using a random subset of the training data through a process known as bootstrapping. This randomness in the data selection helps ensure diversity among the individual trees. Moreover, feature selection is randomized during each split, preventing any single feature from dominating the model. This combination of bootstrapping and random feature selection makes Random Forests resistant to overfitting and capable of handling noisy or high-dimensional data. The final prediction in a Random Forest is derived from a majority vote in classification problems or an average in regression problems, enhancing the model's robustness and predictive accuracy.

In the context of understanding and solving the car evaluation problem, Random Forests offer a promising approach. Car evaluation typically involves assessing various attributes and features of vehicles to determine their quality, safety, or desirability. By applying Random Forests to this problem, we can leverage their ability to handle multi-class classification tasks and make informed decisions about different car models based on a wide range of attributes such as price, safety, maintenance costs, and more. The versatility and robustness of Random Forests make them an ideal choice for this task, where the objective is to classify cars into different evaluation categories.

This report focuses on utilizing Random Forests to tackle the car evaluation problem with a specific dataset. The dataset likely contains information about numerous car models, including features like price, safety ratings, maintenance costs, and more. We aim to apply Random Forests to this dataset to develop a predictive model that can efficiently evaluate and categorize cars based on these features. It is essential to understand how the model's architecture, hyperparameters, and training methodology are adapted to the dataset's specific characteristics. This paper may contribute to the domain of car evaluation, providing insights into the performance and effectiveness of Random Forests in handling such a complex classification task.

## II. Random Forest

### A. *Working of a Random Forest*

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. You can find below a brief explanation of how it works.

- ***Decision Trees:*** Random Forest starts with the basic concept of decision trees. A decision tree is a tree-like structure used for classification or regression. It consists of nodes that represent features and branches that represent decision rules. The leaves of the tree represent the predicted outcomes.
- ***Bootstrapping:*** Random Forest introduces bootstrapping, which involves creating multiple random subsets of the training data. Given a dataset with N samples, you create B bootstrapped datasets, where each dataset is of size N and is constructed by randomly sampling N data points with replacement. This creates diversity among the trees. Mathematically, if D is the original dataset, a bootstrapped dataset Db can be represented as:
  $D_b = \{D_1, D_2, ..., D_B\}$

- ***Feature Randomness:*** In Random Forest, not all features are considered for each split in each tree. For each tree, a random subset of features, denoted as ‘m‘, is selected. This introduces feature randomness.
  Mathematically, if F is the set of all features, the random subset used for a tree can be represented as:
  $F_m \subseteq F, where|\mathrm{F}_m \; | = m$
- ***Aggregation:*** After constructing multiple decision trees using different bootstrapped datasets and feature subsets, the final prediction for a new data point is determined through aggregation. In classification problems, this is typically a majority vote, and in regression problems, it's the average of individual tree predictions.
  Mathematically, for classification problems, the final prediction can be represented as:
  $Y_f$ = Majority($Y_1$, $Y_2$, ..., $Y_B$)
  Where $Y_i$ is the predicted class label from the ith decision tree.

### B. *Key Principles Underlying Random Forest*

Random Forests is a powerful ensemble learning technique that combines multiple decision trees to make more accurate predictions. The key principles underlying Random Forests are as follows:

1) ***Bagging (Bootstrap Aggregating):*** Random Forests use bootstrapping to create multiple subsets of the training data. Bootstrapping involves random sampling of the data with replacement. This process generates diverse subsets of data, which are then used to train individual decision trees. Bagging helps reduce overfitting and enhances the model's robustness.
2) ***Random Feature Selection:*** In addition to bootstrapping, Random Forests introduce randomness in feature selection. For each split in each decision tree, only a random subset of features is considered. This prevents a single feature from dominating the decision-making process and promotes feature diversity. The choice of the number of features to consider (often denoted as ’m’) is a hyperparameter.
3) ***Ensemble of Decision Trees:*** Random Forests consist of multiple decision trees. These trees are constructed independently using different subsets of data and features. The individual decision trees are combined to make a final prediction.
4) ***Voting or Averaging:*** In classification tasks, Random Forests use majority voting to determine the final class label. Each decision tree ”votes” for a class, and the class with the most votes becomes the predicted class. In regression tasks, the final prediction is the average of the predictions from individual trees.
5) ***Reduced Overfitting:*** Random Forests are effective at reducing overfitting because of the combination of bootstrapping, feature randomness, and ensemble learning. Individual decision trees may overfit the data, but by averaging or taking majority votes, the ensemble produces more generalized predictions.
6) ***High Predictive Accuracy:*** Random Forests are known for their high predictive accuracy. They can handle a wide range of data types, including categorical and numerical features, and are robust to noisy data. They are less prone to overfitting compared to single decision trees.
7) ***Feature Importance:*** Random Forests can provide a measure of feature importance. By analyzing which features are more frequently used for splitting in the ensemble of trees, you can gain insights into the significance of each feature in making predictions.

## III. The Problem

We have been given the ”car evaluation” dataset which is a well-known and frequently used dataset in the field of data science and machine learning. It's designed to assess the suitability of different cars based on various attributes, making it a valuable resource for testing and demonstrating the capabilities of classification algorithms.

By analyzing this dataset with Decision Trees, you can create a model that can categorize cars into different evaluation classes based on the attributes like price, maintenance cost, the number of doors, passenger capacity, luggage capacity, and safety rating. Decision Trees offer transparency and interpretability, making it easier to understand the criteria used to make these evaluations, which can be useful for various applications, such as car purchasing decisions, quality assessment, or safety ratings and Random Forests improve the Accuracy Score of those evaluations.

### A. *Data Description*

The column names of the different columns in ”car evaluation” dataset are, ’buying’, ’maint’(maintainance), ’doors’, ’persons’, ’lug boot’(size of the luggage boot), ’safety’, ’class’(the variable that is to be predicted).

The target variable in the Car Evaluation Dataset is ”class.” It represents the overall evaluation of a car, with possible values like ”unacc” (unacceptable), ”acc” (acceptable), ”good,” or ”vgood” (very good). Decision Trees aim to predict this target variable based on the values of the other attributes.

In the context of Decision Trees, this dataset is used to demonstrate how Decision Trees can be applied to make decisions about car evaluations. The Decision Tree algorithm recursively splits the dataset into subsets based on the values of the attributes. It selects the attributes that provide the most discriminatory information to create an interpretable tree-like structure. The leaves of the tree represent the predicted car evaluation labels.

### B. *Data Preprocessing*

1) **Finding missing Values**: For the ”car evaluation” dataset the total number of samples are 1727 and there are 7 attribute columns, in which none of the attributes have missing values. Thus, there is no need of dealing with the missing values.

All the variables have different classes with equal count except the ”target” attribute with has uneven number of different classes.

From *Figures 1, 2, 3* we can observe the data distribution with respect to different attriutes in the ”car evaluation” dataset. *Figure 1 and 3* shows the count of different attributes of the ”car evaluation” dataset except the ”target” attribute. *Figure 2* shows the count of number of different ”target” classes.

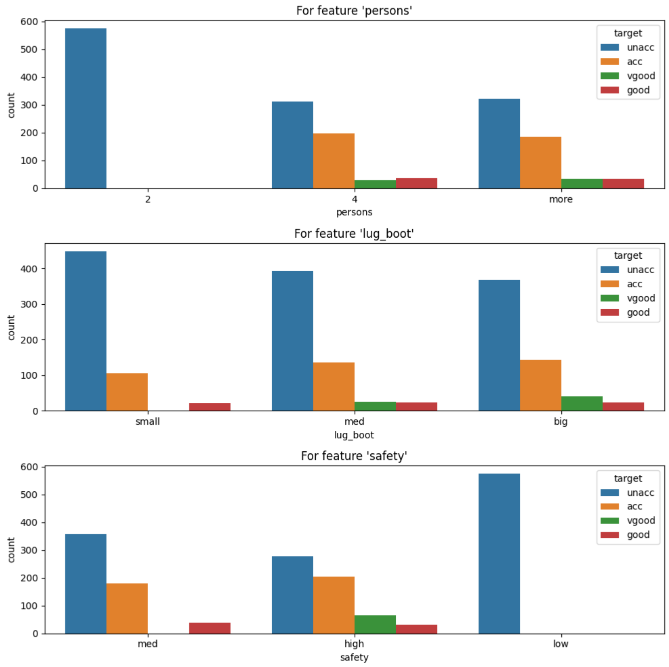


Fig. 1. Attributes vs Count (w.r.t. ”target”)

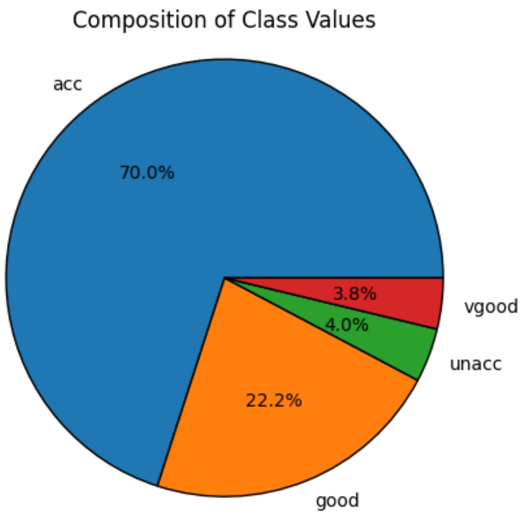


Fig. 2. Class Values Compositions

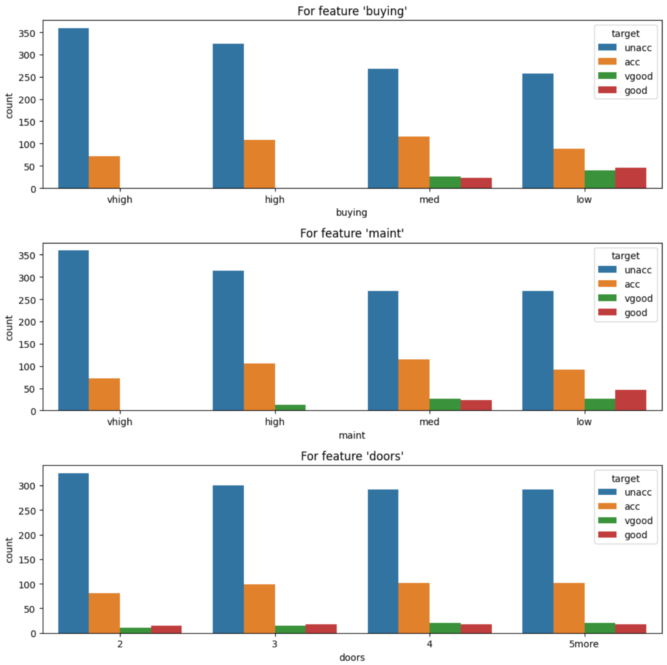
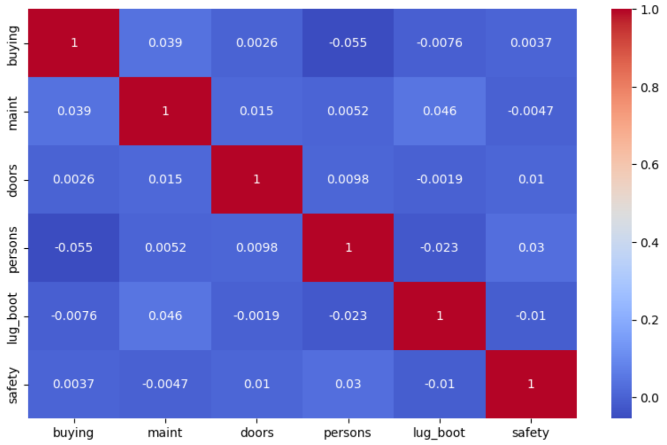


Fig. 3. Attributes vs Count (w.r.t. ”target”)

### C. *Exploratory Data Analysis(EDA) and Visualization*

In EDA we explore the distribution of categorical attributes, check for missing values, and investigate the balance of class labels. Visualization techniques, such as bar plots and histograms, help you understand attribute distributions. Correlation analysis and contingency tables reveal relationships between attributes and the target variable.



|  | buying | maint | doors | persons | lug_boot | safety |
|---|---|---|---|---|---|---|
| buying | 1 | 0.039 | 0.0026 | -0.055 | -0.0076 | 0.0037 |
| maint | 0.039 | 1 | 0.015 | 0.0052 | 0.046 | -0.0047 |
| doors | 0.0026 | 0.015 | 1 | 0.0098 | -0.0019 | 0.01 |
| persons | -0.055 | 0.0052 | 0.0098 | 1 | -0.023 | 0.03 |
| lug_boot | -0.0076 | 0.046 | -0.0019 | -0.023 | 1 | -0.01 |
| safety | 0.0037 | -0.0047 | 0.01 | 0.03 | -0.01 | 1 |

Fig. 4. Correlation Matrix

*Figure 4* shows us the correlation matrix between the different attributes of the ”car evaluation” dataset. EDA unveils how attributes like buying price, maintenance cost, or safety ratings affect car evaluations, for informed decision-making in later stages of analysis or modeling.

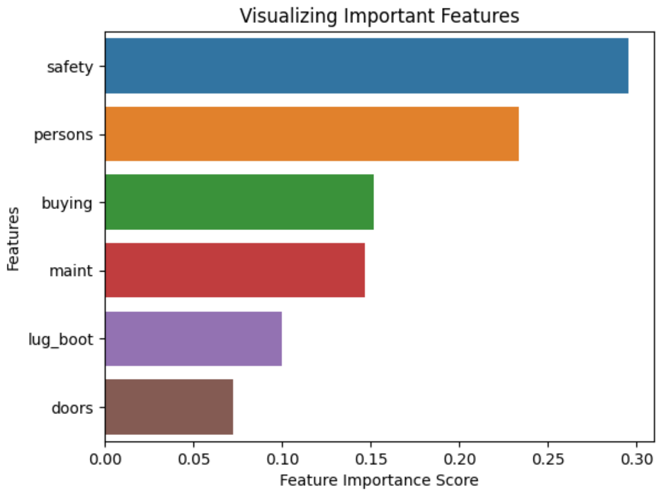


Fig. 5. Feature Importance Scores

### D. *Applying Random Forest Classifier*

After EDA, to apply a Random Forest Classifier to the Car Evaluation Dataset, you first load the dataset and perform data preprocessing by encoding categorical variables and splitting the data into features and the target variable.

Following this, a train-test split is executed to assess the model's performance on unseen data. Subsequently, you create a Random Forest Classifier and train it on the training set. After training, you make predictions on the test set and evaluate the model using metrics such as accuracy, a classification report, and a confusion matrix.

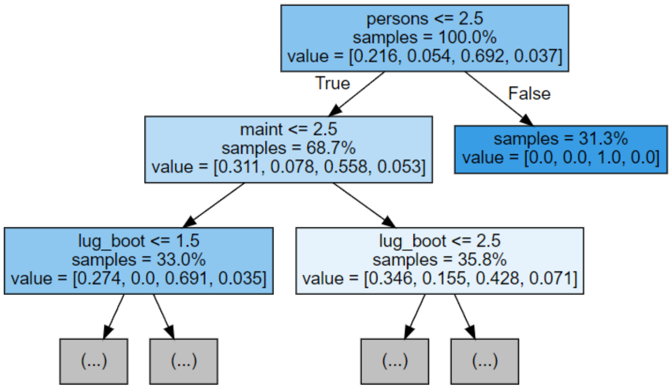

persons <= 2.5
samples = 100.0%
value = [0.216, 0.054, 0.692, 0.037]
True
False
maint <= 2.5
samples = 68.7%
value = [0.311, 0.078, 0.558, 0.053]
samples = 31.3%
value = [0.0, 0.0, 1.0, 0.0]
lug_boot <= 1.5
samples = 33.0%
value = [0.274, 0.0, 0.691, 0.035]
lug_boot <= 2.5
samples = 35.8%
value = [0.346, 0.155, 0.428, 0.071]
(...)
(...)
(...)
(...)

Fig. 6. Decision Tree in the Random Forest

We can visualize the trained Random Forest model to gain insights into the decision-making process, which is valuable for model interpretability. This comprehensive process enables you to categorize cars into various evaluation classes based on their features, providing a clear understanding of the criteria influencing the car evaluation process.

We can select what features to take into consideration in each iteration of the Random Forest Estimators depending on Feature Importance scores as seen in *Figure 5*. Depending on what accuracies we are getting by including different features we can deduce what is the perfect subset of features that should be used to train the model.

In *Figure 6* we can observe one of the iterations of the Random Forest Estimators, this is the one of the decision tree that will predict an outcome. Depending on the number of estimators we instantiate the variable "n_estimators" while running the Random Forest Classifier, we get those many number of decision trees with their respective prediction outcomes.

*Figure 7* shows us the confusion matrix of the data with respect to the target variable of test data's actual classes and predicted classes.

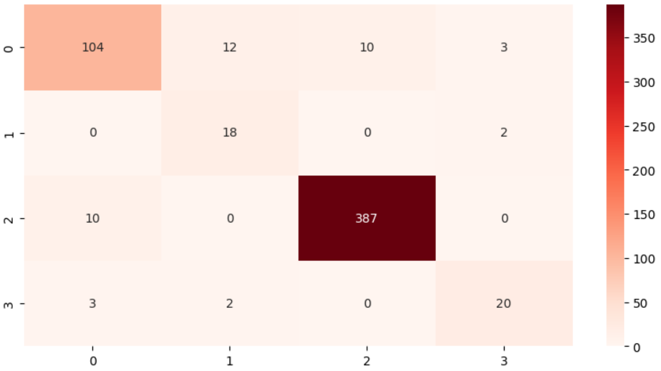


Fig. 7. Confusion Matrix

### E. *Insights and Observations*

The main goal of Random Forest Classifier in this analysis is to classify the sample into the four target values and give as much prediction accuracy as possible as we can observe in the Classification Report in *Figure 8*.

To verify whether the predictions are accurate or not, or the selected attributes are giving the optimal predictions, the model is evaluated by using train test split on the "train" dataset.

The test size is taken as 33% and the Random Forest Classifier Classifier is applied.

From ***TABLE I*** we can see the accuracy varies for different number of estimators for the Random Forest Classifier. All the accuracies are above 90%, we can look at the confusion matrix for better understanding. The following Classification Report is based on Random Forest Classification with selected features taken with respect to the feature importance scores.

The next step of evaluation is calculating the metric scores that is the precision, recall, F1 Score, etc. which is given in *Figure 8* (Classification Report).

Then we try to find the optimal number of estimators for the Random Forest classifier by applying the Random Forest Classifier to the Important Feature data and vary the number of estimators until they reach a constant value or keep decreasing.***TABLE I***. From ***TABLE I*** and *Figure 9* we can conclude that the maximum accuracy is obtained for ***number of estimators = 90 (or) 100***.

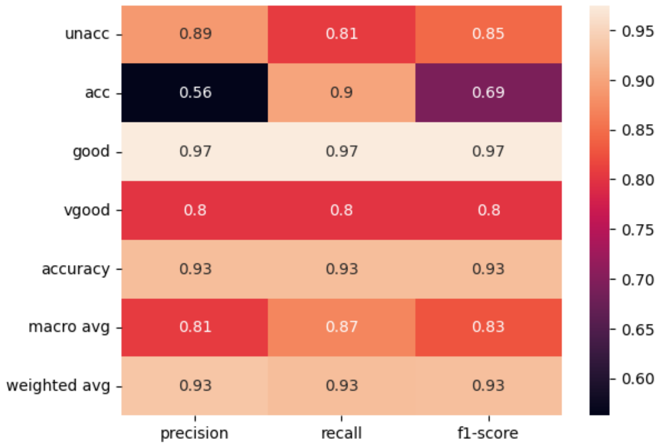


Fig. 8. Classification Report

TABLE I
RANDOM FOREST

| **Number of Estimators** | **Prediction Accuracy *Test Data*** |
|---|---|
| 10 | 0.924 |
| 20 | 0.936 |
| 30 | 0.935 |
| 40 | 0.940 |
| 50 | 0.942 |
| 70 | 0.938 |
| 80 | 0.942 |
| 90 | 0.945 |
| 100 | 0.945 |
| 200 | 0.940 |

From the classification Report in *Figure 9* we can see that accuracies are $>=$ 90% and the model is really good for the given test dataset in all of the cases.

The model is somewhat over fitted because all the accuracies for predictions of the train dataset are almost or equal to 100%.

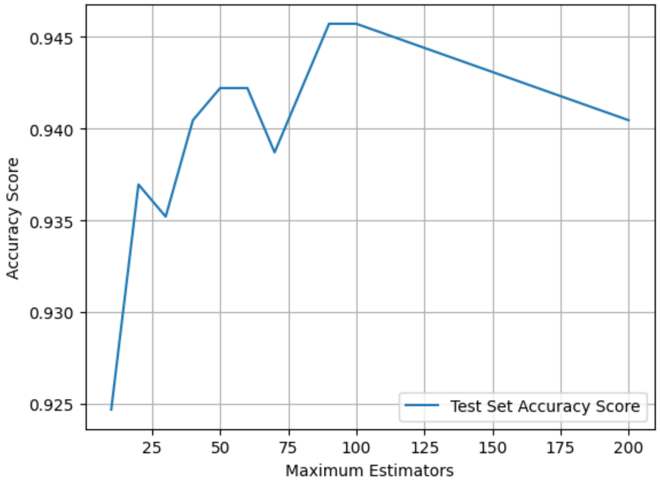


Fig. 9. Estimators vs Accuracy Score

## IV. CONCLUSION

Through the construction of an ensemble of decision trees, Random Forests have showcased their ability to accurately categorize cars based on attributes such as price, safety ratings, and maintenance costs. They also help in estimating the missing data and maintains accuracy when a large proportion of the data is missing. However, there are avenues for further improvement and expansion in the utilization of Random Forests for car evaluation:

- **Hyperparameter Tuning**: Fine-tuning the hyperparameters of the Random Forest algorithm, such as the number of trees, the depth of the trees, and the size of the feature subsets, can lead to improvements in model performance. Optimizing these parameters can make the model more efficient and accurate.
- **Feature Engineering**: A more comprehensive exploration of feature engineering could enhance the model's predictive power. The inclusion of additional features, such as historical car performance data or customer reviews, may lead to a more nuanced evaluation of cars.

## REFERENCES

[1] Bishop, Christopher M., Pattern Recognition and Machine Learning, 2006.

[2] Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition.