DAL 2023 Assignment 5: A Mathematical Essay on Random Forests

Arun Palaniappan
Department of Mechanical Engineering
Indian Institute of Technology Madras
Chennai, India
me20b036@smail.iit.ac.in

***Abstract*—This paper provides an overview of the mathematical formulations and applications of Random Forests. First, the fundamentals of Random Forests are explored, as well as the specifics necessary to comprehend the research. The problem statement and data-set are introduced in a detailed way. The data-set under consideration is one of the evaluation of cars, to explore which feature of the car is more contributing to its acceptability. To accomplish this efficiently, the data has been thoroughly evaluated, and both visual and quantitative insights have been provided in this paper. Finally, the results of the Random Forest model are reported to better understand which features are essential when associating maintenance cost, safety and buying price with the acceptability of the car. We have improved the "Random Forests" section by introducing the boosting algorithm.**

***Index Terms*—Random Forests, F_1-score, precision, recall, car, acceptability, safety, visualization**

I. Introduction

Random Forest is a statistical and non-parametric ensemble technique employed to model and predict the association between various factors or variables. It falls under the category of supervised learning algorithms, used for both classification and regression tasks, such as anomaly detection. The objective of Random Forests is to fit several Decision Trees(set of conditional control statements) to represent the training data.

This is accomplished by an intelligent divide and conquer algorithm for each tree in the Random Forest. We split the data-set for each tree using bootstrapping to reduce variance. The purity of a sub-set of the data-set is defined mathematically, by using different functions like *GINI-Index* and *Entropy*. Each tree in the Random Forest uses only a subset of the features so as to not overfit the data-set. To test how our model will perform on unforeseen data-sets in the future, we build a train-validation split, then train the model (minimize the impurity) on the train split, and then check the prediction accuracy on the validation split.

The "Car Evaluation" data-set contains information about various attributes of cars, such as their maintenance costs, the number of doors, the number of passengers that can be seated, luggage capacity, safety rating, and the overall acceptability of the car. This data-set is useful, particularly in the context of car quality assessment.

In this paper, a comprehensive understanding of Random Forest fundamentals are provided. The paper also undertakes a detailed exploration of the targeted problem and elucidates how the resulting insights are supported by a combination of visual representations and quantitative evaluation.

II. Random Forests

In this section, we will describe the mathematical details of Random Forests. Random Forests is a supervised learning algorithm used for modeling the relationship between two types of variables, the target(dependent) and features(independent). The following is an overview of the jargons and mathematics utilised in the Random Forests approach:

- *Bootstrapping:* Bootstrapping in Random Forest involves randomly drawing N samples with replacement from a data-set of size N, leading to potential repetition of some samples and omission of others. This ensures each tree captures diverse data nuances. By training trees on these varied samples, Random Forest decreases prediction variance, giving more robustness compared to a singular decision tree.
- *Random feature subspace:* In addition to bootstrapping samples, Random Forests introduce another layer of randomness by using a random subset of features for each decision tree's split. This randomness ensures that the individual trees in the forest are not only diverse due to different data samples but also because they consider different features for making decisions. Thus the model does not learn the noise of a particular feature.
- *Aggregating:* Aggregating in Random Forests is the process of combining the predictions of its individual trees to derive a final output. For classification, each tree "votes" for a class label, and the most voted label becomes the ensemble's prediction. In regression, the ensemble averages the continuous predictions from all trees.

A. Assumptions in Random Forests

Random Forest is a powerful statistical technique, but its validity and reliability rests upon certain assumptions that must be met for accurate and meaningful results. Firstly, every tree in the ensemble is given equal weightage, but it may not be the case. Secondly, the features are assumed to have little to no correlation. A statistical method is used to decide which attributes will be the leaf or internal nodes of each tree. This strategy aims to choose attributes that increase information gain.

B. Types of purity metrics used in each tree

- *GINI-Index:* It is a measure of impurity due to a split in the intermediate nodes. It is calculated as,

$$\text{GINI-Index} = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{1}$$

  A weighted sum of all GINI-indices are calculated for all the features, then the feature with the lowest GINI-impurity is chosen to make the split at that node.
- *Entropy information gain:* It denotes randomness due to a split at the intermediate nodes. It is calculated as,

$$\text{Entropy} = -\sum_{i=1}^{n} p_i * log(p_i) \tag{2}$$

  Information Gain is calculated as the difference of the entropy of the parent node and the weighted average of the entropies of the child nodes:

$$\text{Gain} = Entropy(P) - (Entropy(C))_{avg} \tag{3}$$

  The information gain for all the features are calculated and the feature with the highest gain is chosen to make the split at that node.

C. Algorithm

We will now explain the steps involved in modeling a Random Forest:

- We begin with a bootstrapped data-set in the root node of the first tree with a random subset of the features.
- We use one of the two purity measures defined above to find the best feature to split the data-set at the current node.
- Generate the child nodes from the parent node using the above criterion.
- We repeat step-2 and step-3 to continue splitting the nodes until a stopping criterion like the max depth of the tree, minimum number of samples per leaf is met or there is no further significant improvement in the information gain.
- Now we have created one tree. We now repeat step-1 to step-4 to create several such trees.
- Now,to classify a new data point, we take a vote from each of the trees that we have created and the most voted class label becomes the ensemble's prediction.

D. Boosted Trees Algorithm

Boosting is an ensemble learning method that builds a strong predictive model by aggregating the predictions of several weak sequential learners, which are usually simple models. Decision stumps are used as weak learners in XG-Boost, which is one of the well-known boosting algorithms.

Here is an explanation of how boosting works:

- Boosting creates a sequence of weak learners. Every weak learner receives instruction on how to fix the mistakes made by the ones before them.
- A weight is given to every data point during the training process. Every data point has the same weight at first. However, the weights are changed to give misclassified data points more weight after training the first weak learner.
- A weight is given to each weak learner after training, based on how they perform. In the final ensemble, weak learners which perform well are assigned higher weights and weak learners which do not perform well are assigned lower weights.
- The weighted sum of the predictions made by the weak learners constitutes the final robust model. Each learner's weight is determined by how much they have contributed to reducing the total error.

Let $\hat{f}$ be the final model, and the individual trees be represented as $\hat{f}^1, \ldots, \hat{f}^B$. Let r_i represent the residual corresponding to each data point. The tune-able parameters of boosted trees are:

- *B:* It is the number of sequential trees in the boosted model.
- *λ:* It is the learning rate of the boosting algorithm.
- *d:* It is the depth of split of each weak learner.

We will now list the steps involved in modelling an ensemble of boosted trees.

- We will set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set initially.
- For $b = 1, 2, \ldots, B$, iteratively repeat there three steps:
  - Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes in each weak learner) to the training data (X, r).
  - Update $\hat{f}$ by adding a version of the new tree multiplied by the learning rate:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x) \tag{4}$$

  - Update the residuals for the next tree:

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i) \tag{5}$$

- The final model, which is the weighted sum of the predictions made by the weak learners, can be represented as:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x) \tag{6}$$

E. Evaluation Metrics

We now define two quantities which can be used to evaluate the model.

- *Weighted average of F_1-scores ($F_{1,avg}$):* It is defined as the weighted average of F_1 scores of all the target classes, where the weights are the support (S_i) of each class.

$$F_{1,i} = \frac{2 * precision_i * recall_i}{precision_i + recall_i} \tag{7}$$

$$F_{1,avg} = \frac{\sum_{i=1}^{i=n} S_i * F_{1,i}}{\sum_{i=1}^{i=n} S_i} \tag{8}$$

- *Accuracy:* It is the total number of correct predictions divided by total number of validation points,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

F. Advantages of Random Forests over Decision Trees

- *Reduced overfitting:* Random Forests, by combining the results from multiple trees, reduces the overfitting problem that can affect individual decision trees.
- *Effective noise detection:* Random Forests can handle noisy data better than individual decision trees as they are less sensitive to outliers.
- *Higher Accuracy:* Random Forests often achieve better accuracy than single decision trees because they aggregate the results of several trees.

G. Important hyper-parameters of Random Forests

- *n_estimators:* The number of trees in the forest.
- *criterion:* The function to measure the purity of a data subset. It supports 'GINI-Index(*gini*)' and 'Information Gain(*entropy*)' criteria.
- *max_depth:* The maximum depth of each tree in the forest.
- *bootstrap:* Enables the use of bootstrapped samples when building trees.
- *min_samples_split:* The minimum sample size requirement to split an internal node of each tree.
- *min_samples_leaf:* The minimum sample size required for a node to be a leaf node.

III. The Data

A data-set with information about various attributes of cars, such as their maintenance costs, the number of doors, the number of passengers that can be seated, luggage capacity, safety rating, and the overall acceptability of the car are provided to us. The ultimate aim of the study is to identify key relationships between a car's acceptability and its attributes.

A. Data description

In this section, we describe the structure of the data-set. The data-set has 1728 entries. It has been split into train(80%)-validation(20%) data-set. The columns from the data-set are:

- *buying:* The buying price of the car {vhigh, high, med, low}.
- *maintenance:* The cost spent in maintenance of the car {vhigh, high, med, low}.
- *doors:* The number of doors in the car {2, 3, 4, 5, more}.
- *persons:* The capacity of the car in terms of the number of people that can be seated {2, 4, more}.
- *lug_boot:* The size of the luggage boot of the car {small, med, big}.
- *safety:* This contains the safety rating information of the car {low, med, high}.

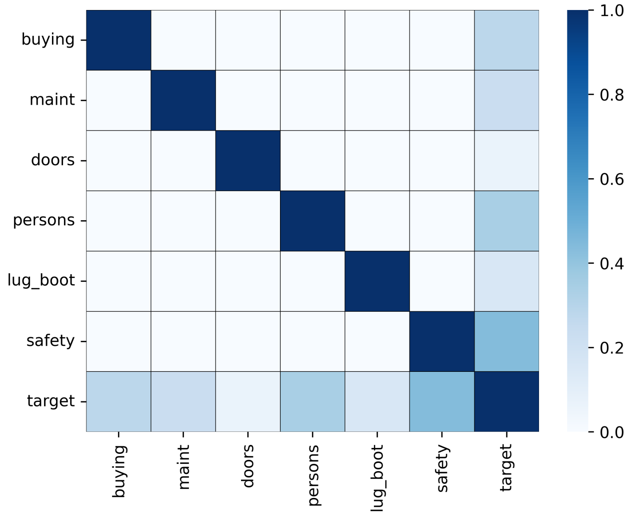


Fig. 1. Correlation heat-map within the variables

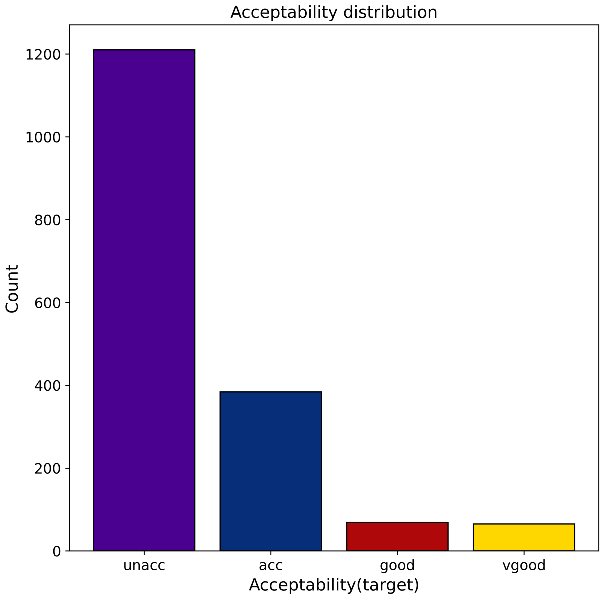


Fig. 2. Acceptability distribution

- *target:* This is the target variable that we have to predict which is the acceptability of the car among people {unacc, acc, good, vgood}.

B. Feature engineering

Categorical classes were encoded by integers using a manual ordinal encoding to maintain the importance of feature order. For instance, safety column was encoded using ['low':0, 'med':1, 'high':2].

C. Data Visualization

1) Correlation plot: Fig. 1 shows the heat map of the correlation values between variables in the data-set.

From the correlation plot we can conclude that:

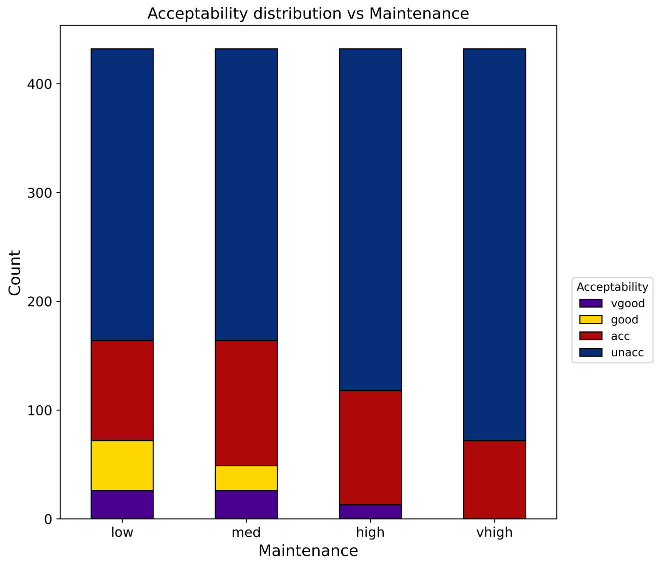


Fig. 3. Maintenance cost distribution vs Acceptability

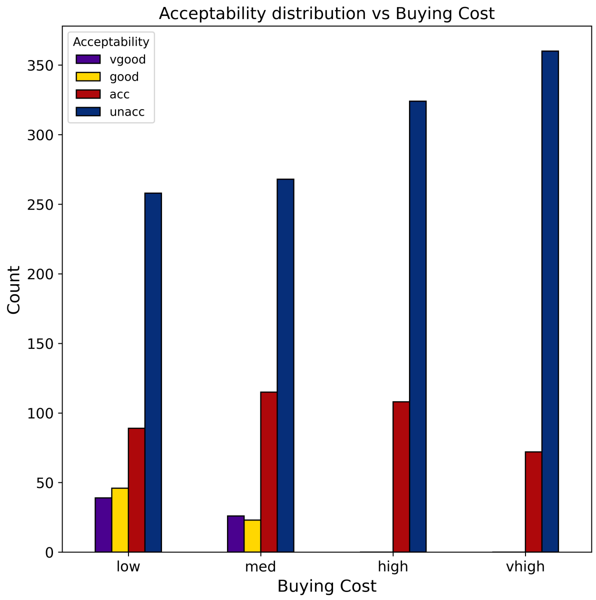


Fig. 4. Buying cost distribution vs Acceptability

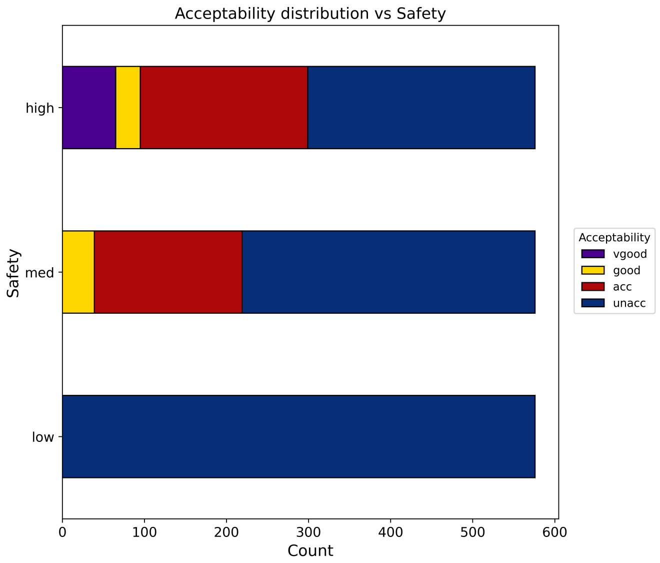


Fig. 5. Safety distribution vs Acceptability

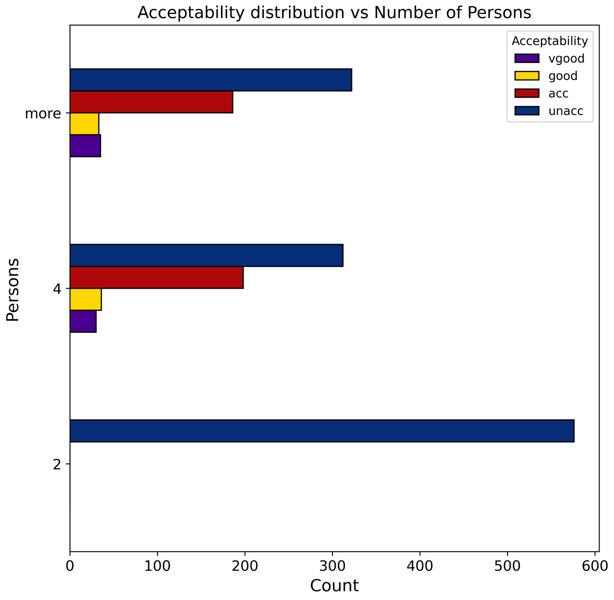


Fig. 6. Number of Persons vs Acceptability distribution

- Two obvious correlations are the *target* with the *number of passengers* and *target* with the *safety*. Higher the capacity and safety, higher is the acceptability of the car.
- There are no other noticeable correlations among the variables.

2) Acceptability distribution: From fig. 2, we can observe significant class imbalance because around 70% of the cars in the data are unacceptable, around 20% are acceptable and the other 10% are either good or very good.

3) Maintenance cost distribution and Acceptability: We can observe from fig. 3 that the majority of cars that are deemed as good or very good have low to medium maintenance costs. Hence, vehicles with high maintenance expenses are often viewed as unacceptable.

4) Buying cost distribution and Acceptability: Fig. 4 shows that all vehicles that are rated as good or very good have low or medium purchase costs. Thus, expensive cars are typically considered unacceptable.

5) Safety distribution and Acceptability: We can observe from fig. 5 that most good or very good cars have medium to high safety. Hence, vehicles with low safety are often viewed as unacceptable.

6) Number of Persons vs Acceptability distribution: Fig. 6 shows that all vehicles that are rated as good or very good have a capacity of 4 or more. Thus, 2-seater cars are typically considered unacceptable.

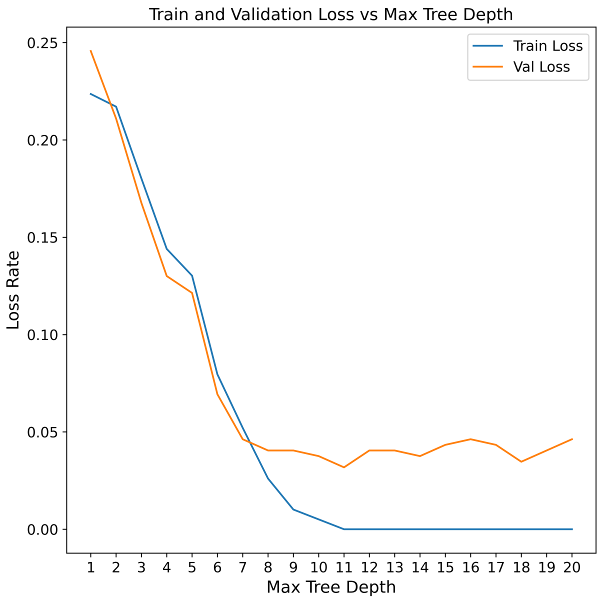
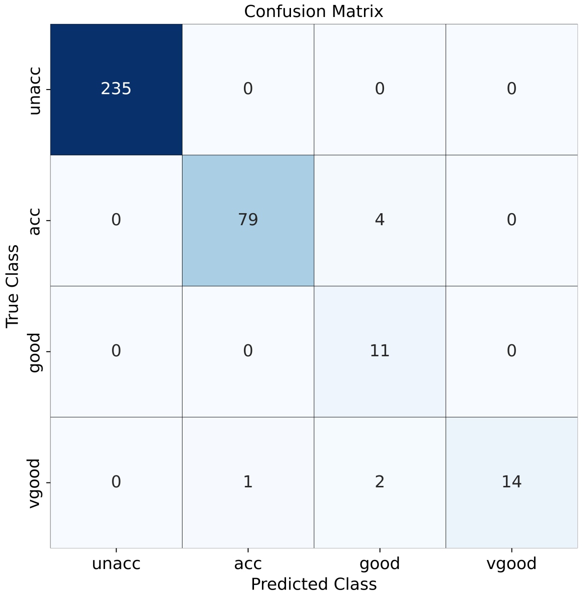


Fig. 7. Train and Validation Loss vs Max Tree Depth



	unacc	acc	good	vgood
unacc	235	0	0	0
acc	0	79	4	0
good	0	0	11	0
vgood	0	1	2	14

Fig. 8. Confusion Matrix

IV. The Problem

A. Outline

We have done the analysing, cleaning and visualizing the data. A train-set and a validation-set are created from the data-set. The validation-set is used to make predictions after the Random Forest model has been trained using the train-set. The validation-set predictions are used to calculate two different metrics to assess model performance.

B. Random Forest modelling

Before fitting the Random Forest to the training data by using *scikit-learn* package in python, we need to specify the *max_depth* hyper-parameter of the tree, which denotes the maximum depth to which the tree branches. In order to accomplish this, we plot the train loss and validation loss against *max_depth* as a variable. From fig. 7, we can observe an elbow point at *max_depth=11*, achieving a very low loss and having no further improvement in accuracy with the increase in *max_depth*.

We now instantiate a Random Forest model, with hyper-parameter (*max_depth=11*), for predicting the acceptability. The predictor variables that we use are *'buying', 'maint', 'doors', 'persons', 'lug_boot' & 'safety'*. We use weighted average of F_1-scores ($F_{1,avg}$) and *Accuracy* score as our evaluation criteria on the validation data-set. After testing on the Validation Split, the model obtains a $F_{1,avg}$-score of 0.97 and also an *Accuracy* score of 0.97.

In the process of improving the paper, we have used Boosted Sequential Trees to do the same task, and it has performed better than Random Forest by obtaining an Accuracy score of 0.98 and is more robust than Random Forest model.

C. Confusion matrix

Fig. 8 represents the confusion matrix generated and gives the count to all the possible outcomes in the validation set(20% of the data).

V. Conclusion

The conclusions that can be drawn from all the above analysis are:

- We can observe that the data-set is structured in such a way that both Decision Tree model and Random Forest model produced almost the same accuracy on it.
- Irrespective of which purity parameter that is used (GINI-Index or Entropy), we almost had the same accuracy in both the cases.
- The acceptability of a car is greatly influenced by its safety rating. All vehicles categorized as very good have a high safety rating, whereas all vehicles with low safety ratings are unacceptable.
- Two seater cars are generally considered unacceptable because the majority of the population have families.
- The number of doors does not affect the acceptability of cars in general.

Future research could use *Random-Forrest* along with pruning, so as to not overfit the data-set. The boosting technique can also be applied to develop robust models. We could also explore on using regularised techniques such as adding a penalty for exceeding the the max tree depth limit, because deeper trees tend to overfit the data-set. We can also use *One-Hot Encoding* for categorical variables, that is creating dummy columns for denoting the presence of a class to avoid any unnecessary bias. We have observed that, there is class imbalance in the acceptability column. This can create inherent bias in the model, hence we can adopt class-balancing techniques to overcome this bias.

References

[1] ”Random forest.” Wikipedia, Wikimedia Foundation, 28 Sept. 2023, *https://en.wikipedia.org/wiki/Random _forest* (accessed 22 Oct. 2023)

[2] sklearn.ensemble.RandomForestClassifier. (n.d.). Scikit-learn. *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble. RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier* (accessed 22 Oct. 2023)

[3] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning : with Applications in R*. New York :Springer, 2013.