Assignment-4: A mathematical essay on Decision Trees

Vasudev Gupta
Interdisciplinary Dual Degree - Data Science
Indian Institute of Technology Madras
Chennai, India
me18b182@smail.iitm.ac.in

***Abstract*—In this work, we will present a mathematical overview of the Decision Trees. We will use the Decision Trees is to classify a car based on its safety. In order to provide the correct data to the model, we will also try exploratory data analysis and feature selection. Finally, we will examine if our model can properly generalise to previously unknown data.**

I. Introduction

Machine learning techniques are gaining popularity, helping many organisations solve challenging issues using data-driven approaches. Machine learning approaches try to estimate the function based on supplied training data by capturing the underlying distribution of the training data and then predicting unseen data using the learnt distribution. When creating the model, one wants the training data to be similar to previously unseen data and generalise well to future data.

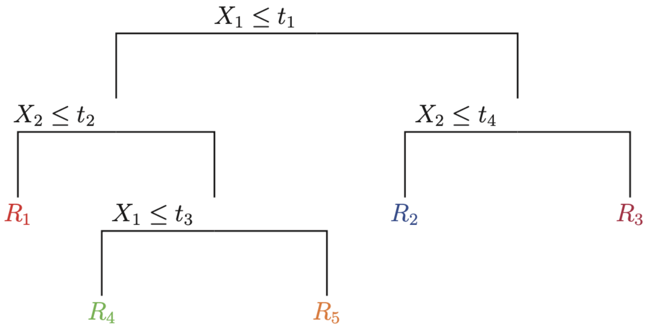
Classification and regression trees, also called decision trees, are defined by recursively partitioning the input space and defining a local model in each resulting region of input space. In the following section, we will go over a thorough mathematical analysis.

The Car Evaluation Database was derived from a simple hierarchical decision model. The prediction task is to classify a car based on its safety. We are given many qualities to base our predictions on, but we must evaluate if any of them may significantly influence the model's performance.

This article will infer a mathematical overview of the Decision Trees and use existing Python packages to fit the Decision Trees to the given data. We will also discuss the data insights we discovered and why we chose to alter or delete particular features.

II. Decision Trees

To explain the Decision trees, consider the tree in the figure below. The first node asks if x_1 is less than some threshold t_1. If yes, we then ask if x_2 is less than some other threshold t_2. If yes, we are in the bottom left quadrant of space, R_1. If no, we ask if x_1 is less than t_3. And so on.



We can write the model in the following form:

$$f(x) = E[y|x] = \sum_{m=1}^{M} w_m I(x \in R_m) = \sum_{m=1}^{M} w_m \phi(x; v_m)$$

where R_m is the m'th region, w_m is the mean response in this region, and v_m encodes the choice of variable to split on, and the threshold value, on the path from the root to the m'th leaf. This makes it clear that a decision tree is just a an adaptive basis-function model, where the basis functions define the regions, and the weights specify the response value in each region. We discuss how to find these basis functions below.

We can generalize this to the classification setting by storing the distribution over class labels in each leaf, instead of the mean response.

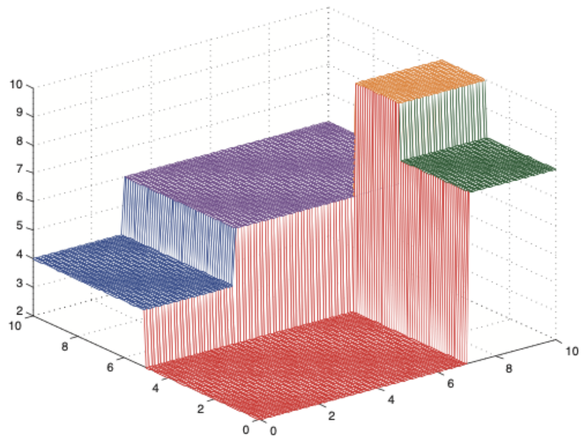
A. Growing a tree

Finding the optimal partitioning of the data is NP-complete, so it is common to use the greedy procedure shown in Algorithm 6 to compute a locally optimal MLE. The split function chooses the best feature, and the best value for that feature, as follows:

$$j^*, t^* = arg\ min\ min\ cost(xi, y_i : x_{ij} \le t) + cost(x_i, y_i : x_{ij} > t)$$

where the cost function for a given dataset will be defined below. For notational simplicity, we have assumed all inputs are real-valued or ordinal, so it makes sense to compare a feature x_{ij} to a numeric value t. The set of possible thresholds

T_j for feature j can be obtained by sorting the unique values of x_{ij}. Although we could allow for multi-way splits (resulting in non-binary trees), this would result in data fragmentation, meaning too little data might “fall” into each subtree, resulting in overfitting.



B. Classification cost

In the classification setting, there are several ways to measure the quality of a split. First, we fit a multinoulli model to the data in the leaf satisfying the test $X_j < t$ by estimating the class-conditional probabilities as follows:

$$\pi_c = \frac{1}{|D|}\sum_{i \in D} I(y_i = c)$$

where D is the data in the leaf. Given this, there are several common error measures for evaluating a proposed partition:

1. **Misclassification rate**: We define the most probable class label as $\hat{y_c} = arg\ max_c \hat{\pi_c}$. The corresponding error rate is then

$$\frac{1}{|D|}\sum_{i \in D} I(y_i \neq \hat{y})) = 1 - \hat{\pi}_{\hat{y}}$$

2. **Entropy, or deviance:**

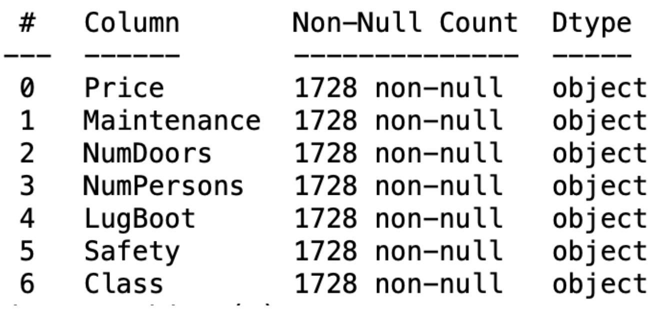
$$H(\hat{\pi}) = -\sum_{c=1}^{C} \hat{\pi}_c \log \hat{\pi_c}$$

III. The problem

The prediction task is to classify a car based on its safety. We will first analyse the data before attempting to fit the decision trees to it. The trained model will next be tested on an unknown dataset.

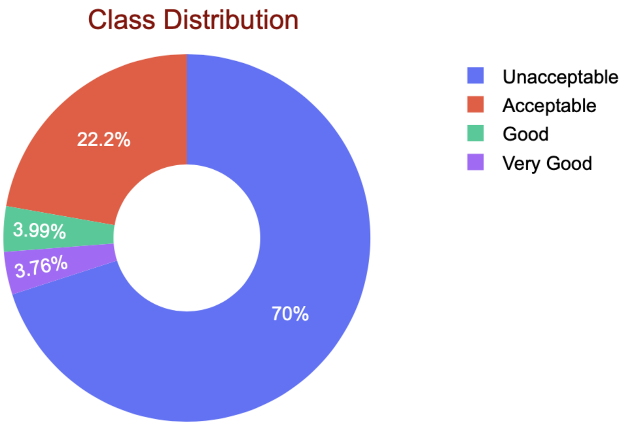
A. Data Analysis and Feature Engineering

Data was clean and did not have any `NaN` values, so we did not worry about them.



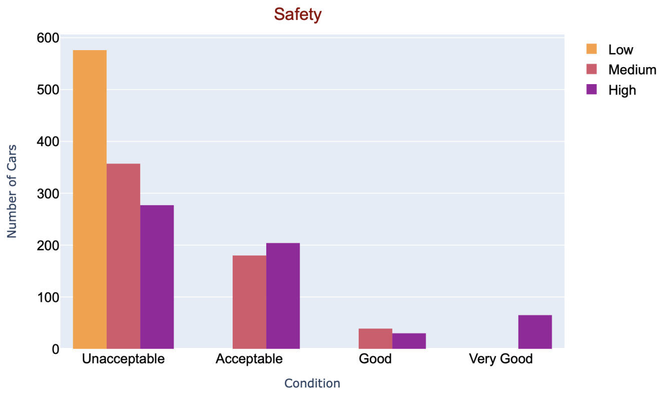
```
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Price        1728 non-null   object
 1   Maintenance  1728 non-null   object
 2   NumDoors     1728 non-null   object
 3   NumPersons   1728 non-null   object
 4   LugBoot      1728 non-null   object
 5   Safety       1728 non-null   object
 6   Class        1728 non-null   object
```

We tried to visualize the class distribution in the data and see if there was any imbalance in the target variable.

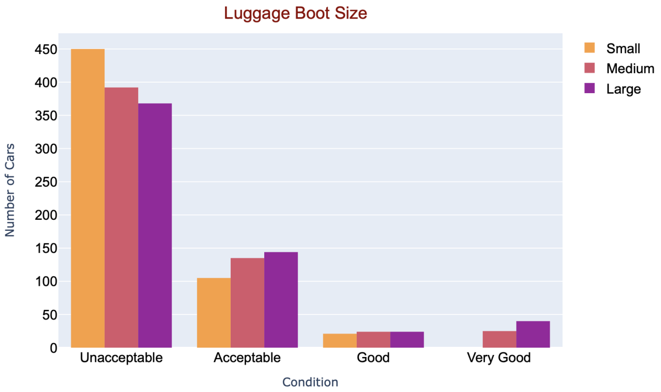


The above pie chart displays that data is quite imbalanced and most of the samples belongs to `Unacceptable` and `Acceptable` categories.

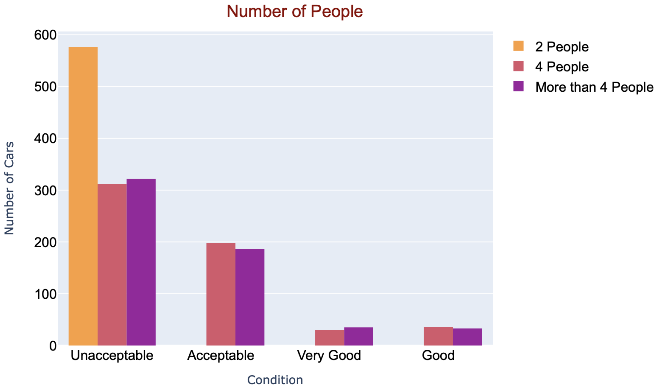
Further, before fitting the model to the data, we tried to understand the categorical features in the data using the bar plots.



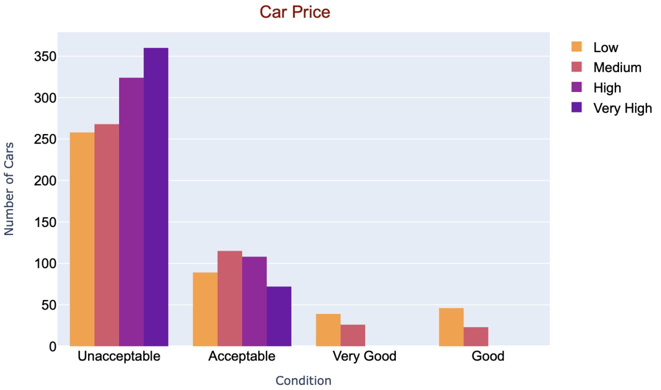
The above table depicts that safety in cars is directly proportional to the number of acceptable cars. Hence, the more unsafe the car is, the more likely it will not be acceptable.



The above figure depicts that more people prefer a car with either medium/high luggage capacity.



The above figure clearly says that more people tend to buy cars which can accommodate at least four people.



From the above plot, we can conclude that highly-priced cars are unacceptable and people like to get a car for less money.

B. Model Fitting

We fitted the Decision trees using the `sklearn` library. Since tree-based models are pretty prone to overfitting, we first tuned the hyper-parameters using `RandomizedSearchCV`. We also used stratified K-fold to find the best hyper-parameters of this dataset. Accuracy score and Mathews Correlation Coefficient are presented in the following table:

Mathews Correlation Coeff	84.5%
Accuracy	93%

In following table, we report the best hyper-parameters for this model and data.

min samples split	2
min samples leaf	1
max features	”sqrt”
max depth	12
criterion	”entropy”

IV. Conclusions

In this essay, we presented the mathematical intuition behind Decision Trees. We also applied Decision Trees to real-world data and inspected whether the fitted model can predict well. We conclude that decision trees can be used on complicated real-world datasets after proper tuning.

References

[1] Christopher M. Bishop, ”Pattern Recognition and Machine Learning”
[2] Kevin P. Murphy, ”Machine Learning, A Probabilistic Perspective”