A Mathematical Essay on Decision Tree

Vignesh Kumar S
Department of Chemical Engineering
Indian Institute of Technology, Madras
Email address: ch18b118@smail.iitm.ac.in

***Abstract*—The purpose of this article is to understand the Decision Trees and the mathematics behind it. In this work, we demonstrate the application of the Decision Tree classifier illustrated through Car Evaluation Database. Due to the complex nature and the abstractness involved in classifying a car given its features, it is often difficult to obtain a good solution through manual approaches. Adopting a machine learning technique can help in obtaining insights and analysing different features in a large amount of data comprehensively. We try to establish a nexus between the nature of the car and different features involving its features.**

I. Introduction

In this paper, we will study Decision Trees, a technique predominantly based on tree-like models of decisions that only contains control statements. It is used to analyze and model either a dichotomous or multiple outcomes in classification setting and could also be extended as a regressor in regression setting. We will use this technique to classify the nature of a car using factors such as safety, capacity, costs etc.. This analysis would aid in obtaining insights among a large dataset and understanding the trend behind the nature of a car and various other factors.

The classification problem attempts to establish a relationship between a categorical target variable with one or more explanatory variable(s). Decision Trees are flow-chart like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. Visually, Decision trees segments the predictor space into a number of simple regions using a rule based approach. In order to make a prediction for a given observation, we typically use the mean or mode of the training observations in the region to which it belongs. They are the fundamental units behind more powerful algorithms such as Random Forest and Adaptive boosting and form a crucial aspect in Machine learning.

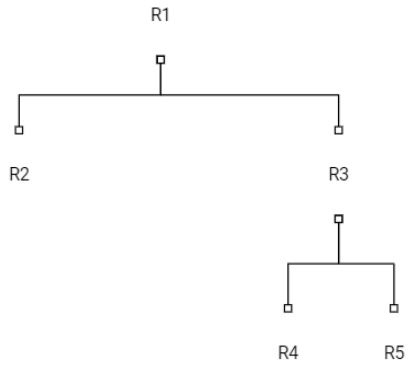
The dataset used for this problem comprises safety, luggage space, capacity, doors, maintenance and buying costs.. We use Decision Tree classifier to learn and predict whether the nature of the car (unaccountable, accountable, good, very good) given these input features.

This work represents the concepts behind Decision Trees and evaluation metrics involved. We then use this technique to establish the relationship to predict the class of of a car it may belong to.

II. Decision Tree

Decision Trees can be applied to both classification and regression problems. In this paper, we'll primarily look at it in a classification point of view but the following analysis could easily be extended into regression setting just by predicting the mean among the leaf nodes and using a squared loss function. Decision trees are easier to interpret and are often compared as a white-box model due to the ability we could compare its prediction. Later on, we'll see that combining many trees can result in powerful algorithms with improvements but with the penalty of loss in interpretation.

The goal in classification problem is to take an input feature $\boldsymbol{x}$ and assign it to one of the K classes. A Decision tree learns to segment the predictor space into simple regions with training dataset and when a new observation comes, it assesses where the new datapoint lies and make the predictions accordingly. Since there are no restrictions to the number of regions that could be made, a tree could grow long and have the potential to overfit every data point. Hence, Decision Trees are known to be low bias and high variance models.



R_2, R_4, R_5 are called Terminal nodes or leaves of the tree. The point along the tree where the prediction nodes are split is referred to as internal nodes.

A. Prediction via stratification of feature space

As we could see, decision trees are easier to visualize.

1) We divide the predictor space- that is the set of possible values for $X_1, X_2, ..X_p$ into J distinct and non overlapping regions $R_1, R_2, ..R_J$.
2) For every observation that falls into the region R_J, we make the same prediction, which is the most frequent observation for a classification problem while mean in regression problem in R_J.

Decision tree aims to find or learn the Regions $R_1, R_2, ..R_J$ that minimizes the cost function. Before considering the choices for cost function, it is important to understand that it is often computationally infeasible to find the best set of Regions or the global optimum. Decision Trees take a top-down, greedy approach with recursive binary splitting.

Generally for a classification setting, accuracy is often deemed as a standard metric that could be used as a loss function. However, for a decision tree, accuracy is not sufficiently sensitive and in practice two other measures are more preferred.

1) Gini Index/Impurity:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

This is a measure of total variance across the K classes. On close observation, this resembles to the variance of a Bernoulli distribution. This above can be equivalently written as:

$$G = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$$

where $\hat{p}_{mk}$ is the proportion of training observation in the m^{th} region that are from the k^{th} class.

Notice that the gini index takes on a small value if all of the $\hat{p}_{mk}$'s are close to zero or one. So, small value implies that a node predominantly contains one class. And a zero value implies that the node is pure. Thus Gini index could also be viewed as a measure of impurity.

2) Entropy:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log\, \hat{p}_{mk}$$

Notice the close resemblance with the thermodynamics entropy formulation. Likewise, here too, entropy takes on a small value if the m^{th} node is pure or less random and large values if it is impure.

B. Learning

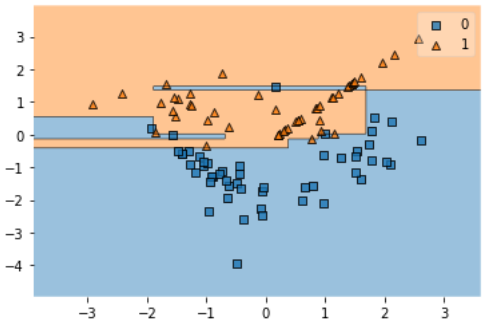
When building a Decision Tree, either Gini-index or entropy is typically used to evaluate the quality of a split. However, a split produces 2 nodes and in order to calculate the measure, a weighted average of leaf node is taken.

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

where m_{left} and m_{right} are the number of samples in left and right node and m is the sum of both. The split that results in the most reduction in the measure (Gini index/ Entropy) is chosen as the best split.

$J(k, t_k)$ could be viewed as the CART (Classification and Regression Trees) cost function. The algorithm first splits the training set into 2 subsets using a single feature k and a threshold t_k. A pair (t_k, k) is searched that minimizes the cost function the best. This procedure is repeated, looking for the best predictor and the best cutpoint in order to split the data further so as to minimise the cost function. This process continues till an appropriate stopping criterion is reached or when every training dataset is perfectly classified.

It is to be noted that without the intervention of appropriate stopping criterion or other means, the tree could grow deep, leading to overfitting. Decision Trees are also unstable, meaning trees learnt from different samples of a dataset are likely to differ in their structure compared to approaches such as Logistic Regression or Naive Bayes.



'The above plot represents the decision boundary of a Decision Tree classifier obtained on a synthetic dataset of two features. The decision tree learned is clearly overfit and doesn't seem to generalise well. In the coming sections, we'll look at ways to reduce the overfitting and improve the predictions.

C. Tree Pruning

A smaller tree with fewer splits might lead to lower variance and better interpretation at the cost of little bias. Let's consider a strategy to grow a very large tree and then prune it.

1) Cost Complexity Pruning:: Rather than considering every possible subtree, we consider a sequence of trees indexed by non-negative tuning parameter α. For each α, these corresponds to a subtree T $\subset T_0$ such that

$$J(k, t_k) + \alpha|T|$$

is as small as possible. Here $|T|$ indicates the number of terminal nodes of a tree. Notice that if $\alpha = 0$, the subtree T is the original tree T_0. We can select a value for α using a validation set or cross-validation.

D. Bagging

Decision Trees suffer from high variance. Bagging is a procedure specifically useful for high capacity classifiers for reducing their variance. Bootstrapping is a technique to obtain datapoints using random sampling with replacement.

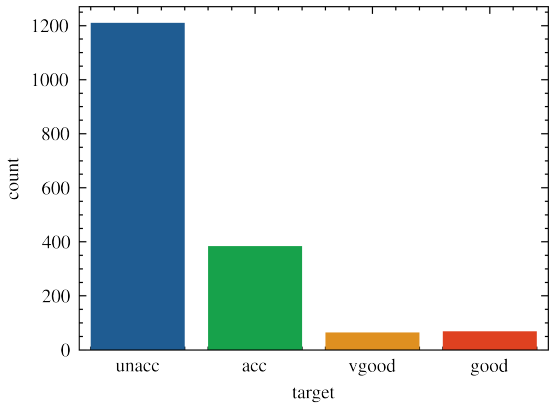
To apply bagging to trees, we simply construct B trees using B bootstrapped training sets and average the resulting predictions. The constructed trees however are grown deep and not pruned, hence suffer from high variance. But averaging reduces variance and results in a better model. This infact is a precursor to Random forest which differs as in addition to this, it deploys random features during splitting whose capability are not explored in this paper. Bagging improves prediction by lowering the variance. But the expense of loss of interpretability should be well noted.

III. The Problem

In this section, We will analyse the Car Evaluation database and try to predict the nature of a car.

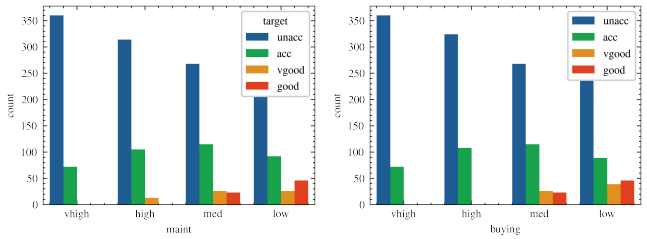
A. Imbalanced Dataset

The dataset predominantly contains class of type 'unacc'. Number of points containing classes 'vgood' and 'good' are considerably less. This is handled by giving appropriate weights to the decision tree learned from the data.



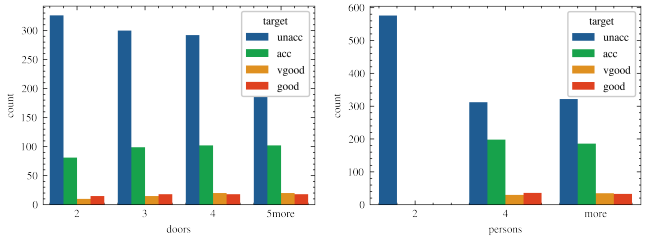
B. Attributes

From plot (a) and (b), We could infer that Most cars falls under unaccountable category. Cars that have high and vhigh buying and maintenance prices are generally unaccountable while the lower priced cars consists of all 4 categories. No good and vgood cars are present with vhigh buying or maintenance.



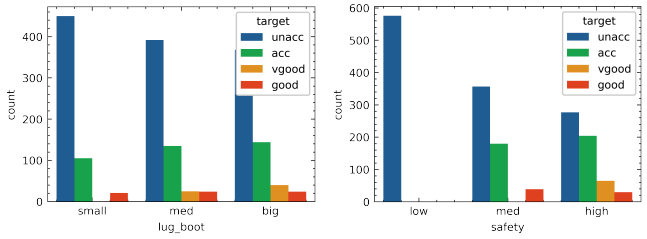
(a) (b)

From plot (c), Almost all categories have similar distributions. Number of doors doesn't seem to be a crucial criteria that could distinguish classes. From plot (d), Cars that have only 2 person capacity are unaccountable. 4 and 4 plus have similar distributions.



(c) (d)

From plot (e), Cars which have small lug boot are never 'vgood'. There has been mixed reviews across cars which have medium to big luggage. However, a trend of bigger space implying higher ratings are seen. From plot (f), Safety seems to play a major role. Every car which have a estimated low safety are unaccountable. No cars are classified as 'vgood' which have medium or low safety. A mixed reviews are seen across cars which were classified as high safety.



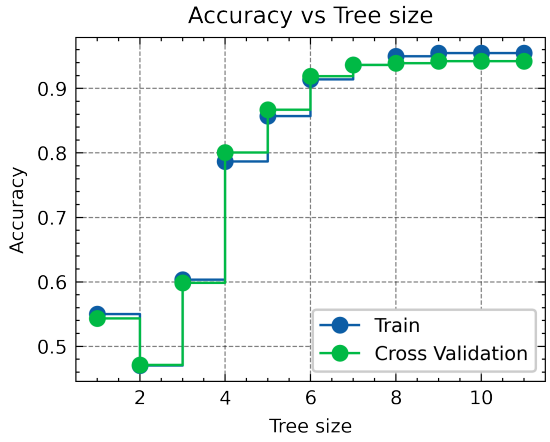
(e) (f)

The dataset after checking for missing values is Label Encoded as the features found were categorical of type ordinal. There exists some order in ordinal features which makes label encoding a better choice compared to one-hot encoding.

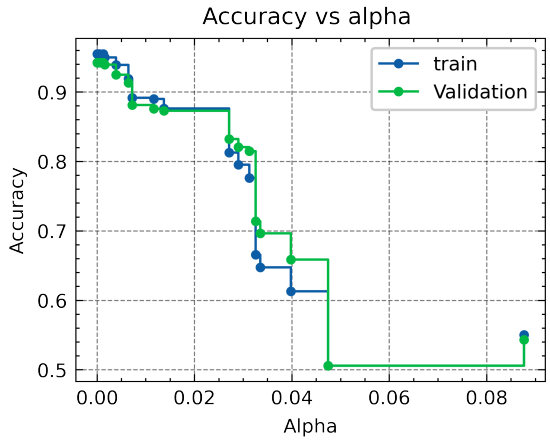
C. Model

The following models all assume a minimum number of samples required to split to be 20. This is kept so that model doesn't overfit too much and predictions could be obtained with confidence. However, this parameter could be experimented for different results.

1) CLF0: Decision Trees are constructed for varying depths. Significant improvements over accuracy is not seen beyond depth 7 with the cross validation set. We'll consider this classifier as the optimum clf0 with depth 7.



2) CLF1: Decision trees are trained with cost complexity pruning. A best α is chosen through validation set. The optimal α obtained was 0.00129.



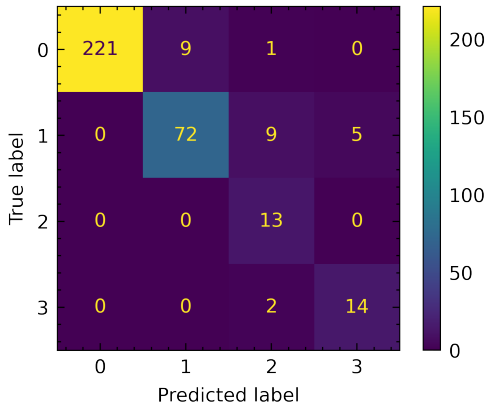
3) CLF2: The third classifier was constructed based on bagging 200 decision trees. Minimum samples required for a split was still considered to be 20 but there were no other restrictions used.

Decision Tree Classifier			
Metric	CLF0 Score	CLF1 Score	CLF2 Score
Accuracy	0.9364	0.9421	0.9104

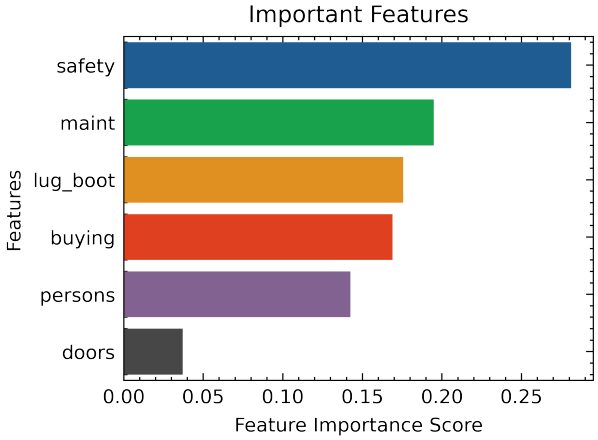
In a classification setting, accuracy may not be the best score to consider especially in a skewed class situation. The training data in our case is imbalanced. Precision is the ratio of correct positive predictions to the total positive predictions of our model. The recall is the ratio of positive instances that are correctly detected by our classifier. The F1 score is simply a harmonic average of these two. The precision, recall and f1 scores were also calculated for each classifier. However since they are of dimensions 4 (equal to the number of classes), they are not reported here. The classifier that performs the best comparing all the criteria in the cross validation set is the CLF1, the pruned classifier.

CLF1 is retrained, but this time on training and cross validation together and the model assessments on the train set are reported below.

4) Best Classifier: The best classifier CLF1 reports the following confusion matrix on the training dataset.



The classifier performs well on the test set. However, some difficulty is seen in getting the class label 2 and 3 accurately (class 'good' and 'vgood' respectively). This could be due to the fact that the dataset was skewed at the first place and hence the errors.



The above suggests that safety is the main feature followed by maintenance to predict the nature of the cars. The least important feature is found to be number of doors. Note that this agrees with the intuition that we obtained through visualization of the dataset.

IV. Conclusions

Based on our analysis, the key takeaways we had from this exercise are the following:

1) Decision Trees are prone to overfitting. Without intervention, they tend not to generalise well and hence appropriate means should be employed to counter this.

2) Decision trees are easier to interpret. However, they are prone to instability and could vary with few differences in data. To address this, we discussed bagging, a method to reduce variance but at the expense of interpretability.
3) Safety is the most important feature that is considered for classifying the nature of a car. We also saw how just by visualizing, how a large amounts of insights were obtained.

Possible avenues of research could include trying out dimensionality reduction before applying decision trees, visualizing decision stumps and using boosting methods to improve the results.

REFERENCES

[1] An Introduction to Statistical Learning, Gareth James et al. pp.303-323
[2] Aurelian Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems, pp. 215–228
[3] Christopher M. Bishop, Pattern Recognition and Machine Learning