Assignment-4: A Mathematical essay on decision trees.

Ahmed Shmels Muhe
M.Tech in Data Science
Indian Institute of Technology Madras, IIT Madras
Chennai, India
ge22m009@smail.iitm.ac.in

***Abstract*—In this revised (v. 2) work on the decision tree classifier, we will give a mathematical study of decision trees and the mathematics behind them. In this work, we demonstrate the application of the decision tree classifier illustrated through the Car Evaluation Database, which is applied to a data set containing different car attributes. The critical task is to classify them as unacceptable, acceptable, exemplary, or very good based on their purchase price, cost of maintenance, number of doors, capacity in terms of people to carry, size of the luggage boot, and estimated safety of the car, and determine whether some are acceptable. In this revised version, I include the mathematics behind the decision tree classifier in Section 2. In Sections 2.1 and 2.2, I explained the decision tree classifier entropy and Gini index/impurity. In Section 3, I evaluated the data sets and processed them. In Section 4, I worked on model improvement and updated the writing and some of the outputs after some modifications in the code.**

***Index Terms*—Decision trees, Database.**

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 analyse and model either dichotomous or multiple outcomes in a classification setting and could also be extended as a regressor in a regression setting. We will use this technique to classify the nature of a car using factors such as safety, capacity, cost, etc. This analysis would aid in obtaining insights from a large data set 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 and one or more explanatory variables. Decision Trees are flow-chart structures 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 segment 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 of Machine Learning.

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

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

II. DECISION TREES

Decision trees can be applied to both classification and regression problems. In this paper, we'll primarily look at it from a classification point of view, but the following analysis could easily be extended into a 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 to a white-box model due to the ability to compare their predictions. 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 a classification problem is to take an input feature x and assign it to one of the K classes. A decision tree learns to segment the predictor space into simple regions with a training data set, and when a new observation comes, it assesses where the new data point lies and makes the predictions accordingly. Since there are no restrictions to the number of regions that could be made, a tree could grow and have the potential to over-fit every data point. Hence, decision trees are known to be low-bias and high-variance models.

To explain the Decision trees, consider the tree in the figure below. The first node asks if x1 is less than some threshold t1. If yes, we then ask if x2 is less than some other threshold t2. If yes, we are in the bottom left quadrant of space, R1. If no, we ask if x1 is less than t3. And so on.

We can write the model in the following form:

$$f(x) = E[y|x] = \sum_{m=1}^{m} WmI(X \in Rm) = \sum_{m=1}^{m} Wm\theta(x; Vm) \tag{1}$$

where Rm is the m'th region, Wm is the mean response in this region, and VM 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 an adaptive basis-function model, where the basis functions define the

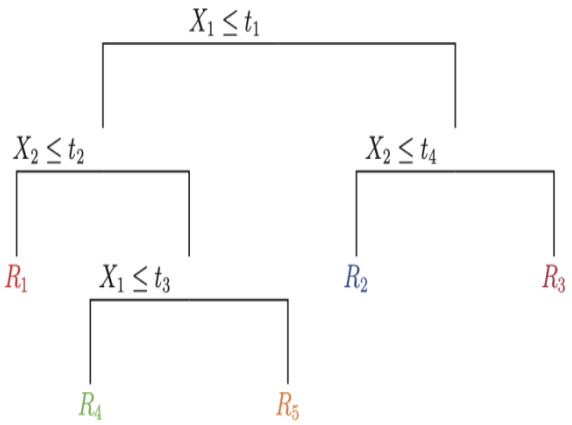

Fig. 1. Visualize decision-trees how node arranged.

regions, and the weights specify the response value in each region. We discuss how to find these basic functions below. We can generalize this to the classification set by storing the distribution of class labels in each leaf, instead of the mean response

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 preferred. which are called attribute selection measures.

1) Entropy: Entropy measures the impurity in the given data set. In Physics and Mathematics, entropy is referred to as the randomness or uncertainty of a random variable X. In information theory, it refers to the impurity in a group of examples. Information gain is the decrease in entropy. Information gain computes the difference between entropy before the split and average entropy after the split of the data set based on given attribute values.

Entropy is represented by the following formula:-

$$D = -\sum_{K=1}^{c} Pmk \log_{(2} Pmk). \quad (2)$$

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

2) Gini Index/Impurity:

$$D = 1 - \sum_{K=1}^{c} P^2 mk. \quad (3)$$

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

Steps to Calculate Gini for a split:

- Calculate Gini for sub-nodes, using the formula sum of the square of probability for success and failure $p^2 + q^2$.
- Calculate the Gini for the split using the weighted Gini score of each node of that split.

In the case of a discrete-valued attribute, the subset that gives the minimum Gini index for that chosen is selected as a splitting attribute. In the case of continuous-valued attributes, the strategy is to select each pair of adjacent values as a possible split point and a point with a smaller Gini index chosen as the splitting point. The attribute with a minimum Gini index is chosen as the splitting attribute.

III. THE PROBLEM

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

A. Imbalanced Data set

The data set predominantly contains a 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. Data Analysis and Feature Engineering

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

```
car_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   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
dtypes: object(7)
memory usage: 94.6+ KB
```

Fig. 2. We tried to visualize the class distribution in the data and see if there was any imbalance in the target variable.

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

C. 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 Randomized Search CV. We also used stratified K-fold to find the best hyper-parameters of this data set. in the following table:

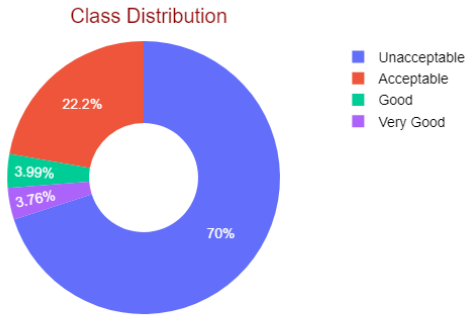

Fig. 3. The above pie chart displays that data is quite imbalanced and most of the samples belong to Unacceptable and Acceptable categories.

Fig. 4. Fig 4 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.

Fig. 5. The above figure depicts that more people prefer a car with either medium/high luggage capacity.

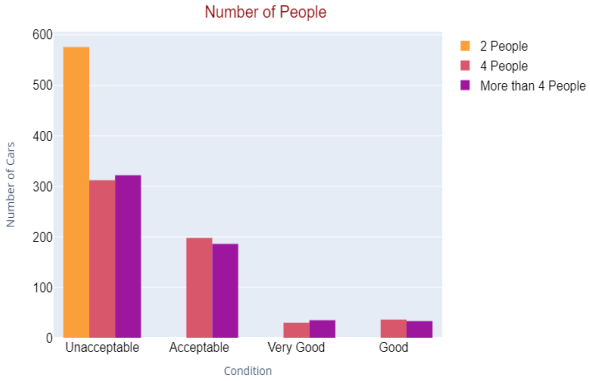

Fig. 6. The above figure clearly says that more people tend to buy cars which can accommodate at least four people.

Fig. 7. From the above plot, we can conclude that highly-priced cars are unacceptable and people like to get a car for less money.

Fig. 8. From the above plot, we can conclude that people are not preferred to buy cars that cost much for maintenance

IV. MODEL IMPROVEMENTS

We selected a subset of features using the Recursive feature elimination technique. Recursive feature elimination, in short

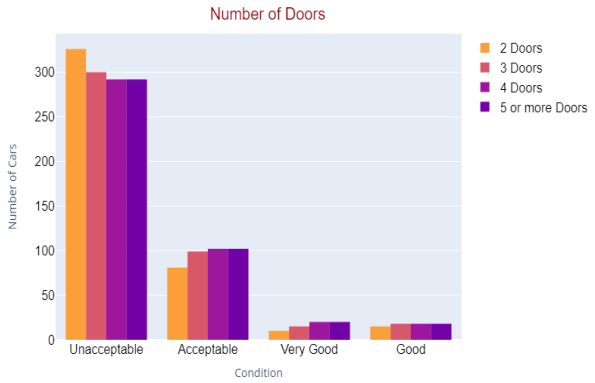

Fig. 9. From the above plot, we can conclude that people prefer to buy cars that have 2 doors mostly

Mathews Correlation Coeff	84.5%
Accuracy	93%

Fig. 10. The accuracy score and Mathews Correlation Coefficient are presented

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

Fig. 11. From the above table, we can say the best hyper-parameters for this model and data.

RFE, is a greedy optimization algorithm that aims to find the best-performing feature subset. Each iteration trains a classification model (in this case, a Decision Tree Classifier). It eliminates the worst-performing feature until all the components are exhausted or satisfy stopping criteria. We observed that RFE helped increase the model's performance significantly. With RFE, we were able to reduce the number of features by 20 %. Our new results are summarized in the following table:

Mathews Correlation Coeff	88.4%
Accuracy	94.3%

We obtained the above results using the Decision Tree as a base estimator in RFE. In the base estimator, the following hyperparameters are used:

min samples split	2
max features	”auto”
max depth	12
criterion	”entropy”

V. CONCLUSIONS

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

- Decision Trees are prone to overfitting. Without intervention, they tend not to generalise well and hence appropriate means should be employed to counter this.
- Decision trees are easier to interpret. However, they are prone to instability and could vary with a few differences in data. To address this, we discussed bagging, a method to reduce variance but at the expense of interpretability.
- Safety is the most important feature that is considered for classifying the nature of a car. We also saw how just by visualizing, how 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] Kingsford, C., Salzberg, S. L. (2008). What are decision trees?. Nature biotechnology, 26(9), 1011-1013.
[2] Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261-283.
[3] Rokach, L., Maimon, O. (2005). Decision trees. In Data mining and knowledge discovery handbook (pp. 165-192). Springer, Boston, MA.
[4] Quinlan, J. R. (1987). Simplifying decision trees. International journal of man-machine studies, 27(3), 221-234.
[5] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.
[6] Brodley, C. E., Utgoff, P. E. (1995). Multivariate decision trees. Machine learning, 19(1), 45-77.
[7] Bouckaert, R. R. (2004, December). Naive bayes classifiers that perform well with continuous variables. In Australasian joint conference on artificial intelligence (pp. 1089-1094).