Assignment 4: Decision Tree

Rohith D
Electrical Engg.
IIT Madras
Chennai, India
ee18b148@smail.iitm.ac.in

***Abstract*—This is a study on decision tree classifiers. We use an exhaustive data-set containing size, capacity, and other information about cars. Using a Decision Tree Classifier, we predict a target variable based on safety.**

***Index Terms*—decision tree, classification, car safety**

I. Introduction

The Car Evaluation Data-set has been derived from a simple hierarchical decision model. It contains 1728 samples and is exhaustive, i.e., it contains all possible combinations of the categorical data. It is used to classify a car based on safety. Since the data-set is exhaustive, the decision tree can perfectly classify any new sample that is presented to it.

We do an exploratory analysis of the given data using some charts to get a preliminary idea of the distribution of the data. The model establishes a Decision Tree Classifier between the input and output variables. A decision tree is a predictive model that uses a tree-like structure with nodes that make decisions based on the input features. It is widely used because of it's intelligibility and simplicity.

The key task is to train a decision tree from the given data. This decision tree can accurately classify any new data that is input to it. We can study how the decision tree accuracy changes with hyper parameters.

We start with an overview of Decision Trees. Then, we do an exploratory analysis based on some plots. We present our observations based on the data visually. We then analyze the fitted Decision Tree classifier. From these plots we arrive at a conclusion.

II. Decision Tree

Decision Tree Learning is a supervised machine learning algorithm which uses a decision tree as a predictive model.

A decision tree has a tree-like structure where each non-leaf node makes a decision based on a feature value. The branches represent the decisions and the leaves (end nodes) represent the predictions made by the model. The inputs can be categorical, continuous, or both (mixed). Since a decision tree is represented visually by a flow chart, it is easy to understand and interpret. It doesn't require much data preparation/ normalization and can work well with large data-sets. However, decision trees can also lead to non-robust models and over-fitting.

A decision tree has three important hyperparameters: maximum depth, splitting criterion, and splitter.

Splitter refers to the how the splitting feature at a node is selected. 'Best' selects the best split and 'Random' selects the best split out of a subset of features. The criterion used to measure the quality of a split is either Gini impurity or entropy/ information gain.

Gini impurity is a measure of how often an element chosen randomly from the set would be labeled incorrectly if it was randomly labeled according to distribution of labels in the subset. It is at a minimum when all cases in the node fall into a single category. If p_i be fraction of items from label i in a set with J classes, then Gini impurity is

$$I_g(p) = \sum_{i=1}^{J}(p_i \sum_{k \neq i} p_k) = \sum_{i=1}^{J}(p_i(1-p_i))$$

$$= \sum_{i=1}^{J}(p_i - p_i^2) = 1 - \sum_{i=1}^{J} p_i^2$$

Information gain is derived from the concept of entropy in information theory, which is defined as

$$H(T) = -\sum_{i=1}^{J}(p_i \log_2 p_i)$$

The disagreement between Gini impurity and Information Gain is usually very low, so either can be used in most cases. Information gain is slower to compute since it requires a logarithmic computation.

Similar to other classifiers, we compare different Decision Tree Classifiers using accuracy=$\frac{Predictions_{correct}}{Predictions_{total}}$ computed on test data-sets.

III. The Problem

There is a variety of features present in the given data. We look at their distribution and see how they affect then target variable.

We first clean the data before working with it: We convert the categorical features which are ordinal in nature, using ordinal encoding into numbers. We use encode buying price, maintenance price and safety using one scale and boot capacity using another scale. The number of doors and capacity features are already numerical in nature

We see that the data is exhaustive by looking at the data frame, which has all possible feature combinations.

To start, we try to draw some preliminary relations from the data using graphs. We check for the relation between safety and target variable.

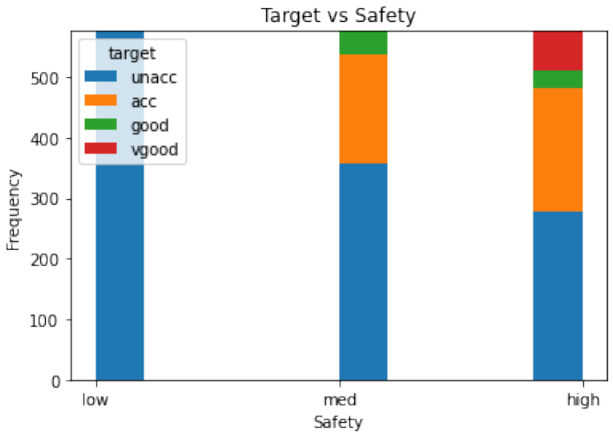

Fig. 1. Target according to Safety

We see that all low safety cars are unacceptable, while most medium and high safety cars are at least acceptable. The cars rated very good have to be highly safe.

Next, we check how the target varies with buying and maintenance prices

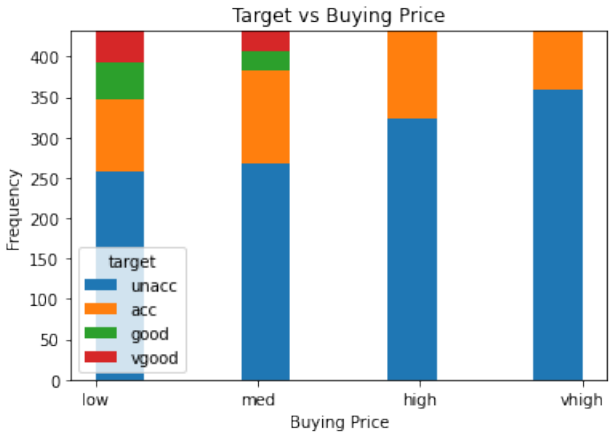

Fig. 2. Target according to Buying Price

We see the rating of a car goes up as the buying price goes down, with highly and very highly expensive cars to buy not being rated more than acceptable. It is similar for maintenance price, though some highly expensive cars are rated very good.

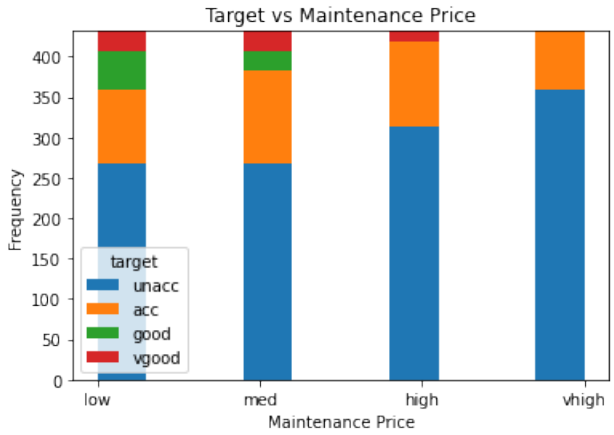

Fig. 3. Target according to Maintenance Price

Next. we analyze the capacity's effect on rating using a violin plot.

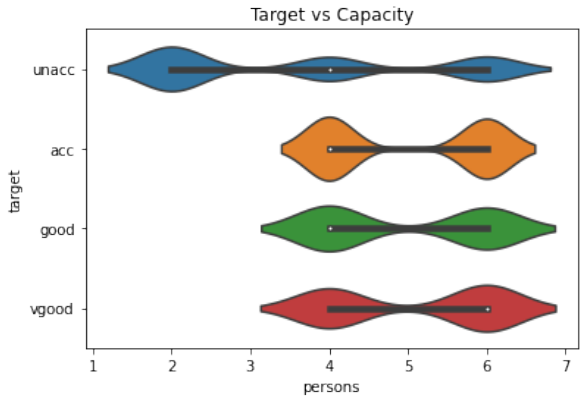

Fig. 4. Rating according to Capacity

We see that all cars with a 2 person capacity are rated unacceptable, and rating is higher if capacity is more.

We look at effect of boot size and number of doors on rating.

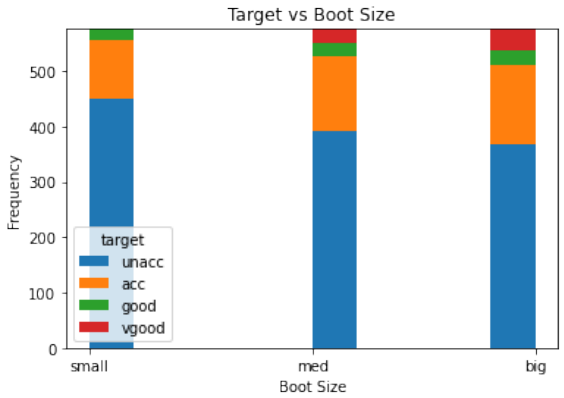

Fig. 5. Rating according to Boot Size

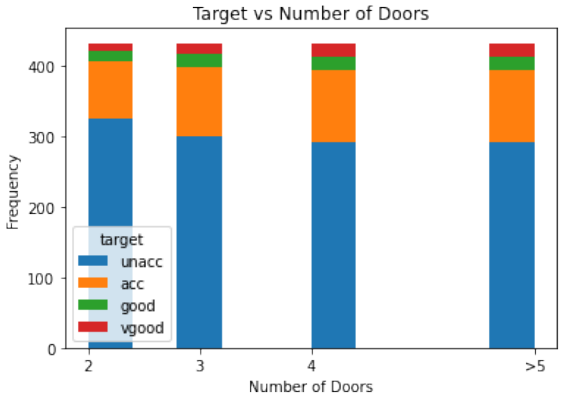

Fig. 6. Rating according to Number of Doors

We see that the higher the boot size the better the rating. Similarly, Cars with more doors get better ratings, but this relation is almost negligible.

As the data-set contains all possible combinations of features, we use the complete data-set to train a decision tree model, since using a train test split would lose us important information.

Since the data-set is exhaustive, any new sample will already be present in the data-set, so we do not do a train-test split before training, knowing that we will get a perfect accuracy of 100% for classification.

We run the decision tree classifier and find that we get a perfect classification rate for a depth of 12. Since the decision tree is quite large, we plot the it to a depth of 4 to analyze it.

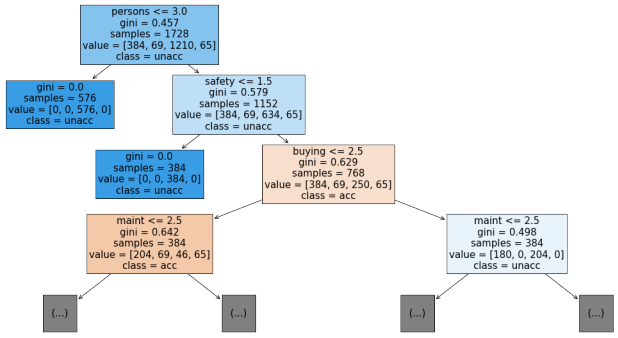

Fig. 7. Decision Tree

We see that the decision trees makes similar conclusions to what we got from the graphs, i.e, all cars with less that 3 person capacity are unacceptable, all cars with a low safety are unacceptable and so on. The Gini index generally keeps decreasing as we go down the tree until it reaches a minimum of zero at the leaves.

Thus, we have a perfect accuracy decision tree built using the exhaustive data-set. We check to see the accuracy score for some lower maximum depths. We see that we get a score of 0.982, 0.993 and 0.997 for a depth of 9, 10, and 11 respectively. So we see that we need at least a depth of 12 to reach 100% accuracy.

IV. Conclusions

We have seen which features affected the target rating of a car and how it is affected. We see that cheaper cars, both to buy and to maintain, are more likely to have higher ratings. Cars with low safety and a capacity of 2 are all rated as unacceptable. The rating goes higher as boot size increases, but the effect of number of doors is negligible. We have made a decision tree model that can give us a 100% accuracy. Studying it visually has shown us that it makes many of the same conclusions based on the features that we have seen. For example, the first split it makes is that all cars with capacity of 2 are unacceptable. Thus, we could use the trained decision tree to make conclusions about the data instead of charts of the input features.

Further study is possible on tuning hyper-parameters to get an accuracy close to 1 for a lower depth, so that high accuracy classification can be done with even lower computational cost.

References

[1] https://en.wikipedia.org/wiki/Decision_tree
[2] https://www.geeksforgeeks.org/decision-tree/
[3] https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052