CS 5830 Project 7

Peter Fowles and Jake Thurgood

Repository | Slides

Introduction

Every year hundreds of thousands of cars are bought and sold. Because of the free market nature of that exchange the rate of exchange is not predefined. Thus there is intuition about finding the best value in the field. By looking at several different factors, a subjective value proposition can be determined. Thus our stakeholders are primarily consumers looking to purchase a car, while secondary stakeholders are those looking to sell a car as they can reflect the price of the car based on the results of other factors.

Dataset

Our dataset comes from the year 1990 and looks at 6 variables in an attempt to determine if a car should be classified as an unacceptable, acceptable, good or very good deal. With 6 categorical classifiers we look at the price, maintenance history, number of occupants, doors, trunkSize and safety features of the car we are able to train our neural network and decision tree. To support the decision making tree we had to one hot encode the categorical values, for better understanding of the encoding refer to the second set of python code in our notebook. Thankfully all values were ordered which made this significantly easier on our model. To support the neural network, we took these encoded values and normalized them with a standard scaler.

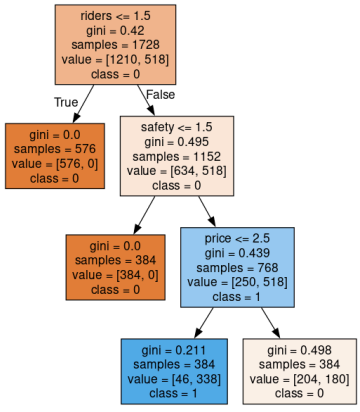
Analysis

On the decision trees, we began by simply putting our encoded dataset into a decision tree with a depth of 3. We then tried excluding the safety and number of potential riders. After that, we used all feature columns again but with a depth of 5, then a depth of 2, then a depth of 4.

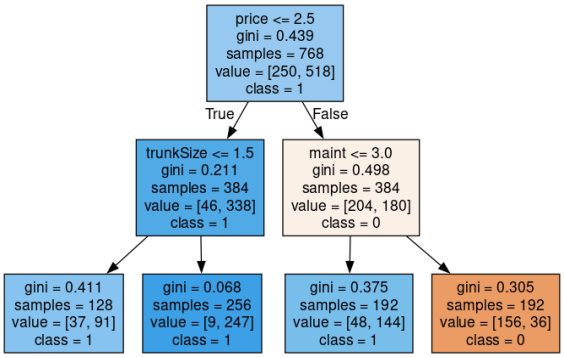
For the neural network, we began with a simple structure with an input layer consisting of our 6 features, a single hidden layer of 4 nodes, and a binary output layer that would determine whether a car is good or bad. We then tried neural networks with varied amounts of hidden layers and nodes per hidden layer, looking at the precision, recall, f-scores, and accuracies of each network's test results.

Results

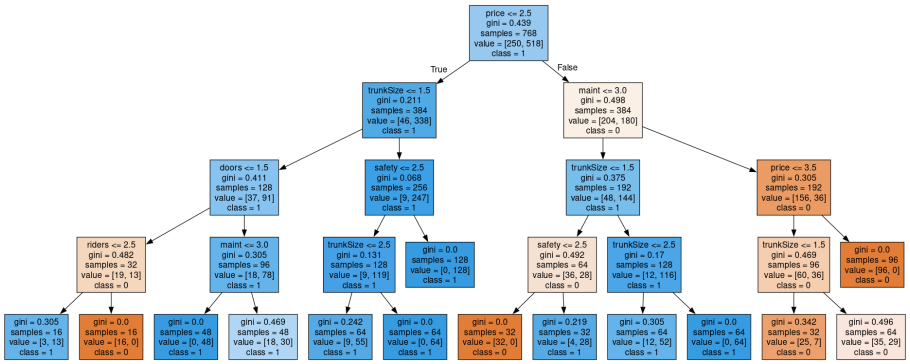
When we started working with the dataset to develop the decision tree we quickly realized that the creators of the dataset have made some very subjective calls about creating the classification. Any car that didn't have the ability to carry more than 2 riders or had a safety rating of bad was automatically considered a bad value. Thus for the ease of exploration we accepted that decision of the dataset creators and removed all entries where the vehicle was considered unsafe or only had the ability to carry 2 passengers. Meaning the resulting trees' decision points are technically 2 deeper then visually demonstrated.

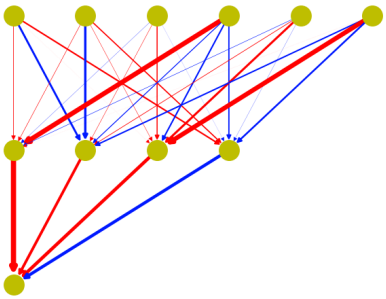


When staying at depth 2 we are able to keep all final decisions below .42 in gini score. However the overall accuracy of the tree is still averaging somewhere in the .2 on the gini scale. We let the decision tree go deeper in the decision making process and started to get some perfect scores resulting from the decision trees. Which makes this an interesting conundrum, we are technically 6 decision deep to reach this conclusion however our samples are still a decent size for representation of the overall entries thus we do not feel that going down 6 decision is not an overfitting of the model onto the decision tree because we are able to keep the gini scores within an acceptable range given the overall average of the tree.

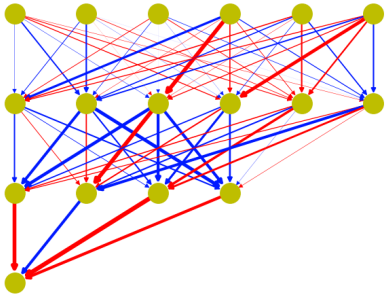


Going across the decision points it was interesting that in the 6 deep the fourth level was completely unique qualifiers for decision making. Which is interesting because it suggests that there is a lot of variance going on and that there is no next best option. Thus it feels the model is looking at the data given to it and sorting that individually allowing for more variability between levels.



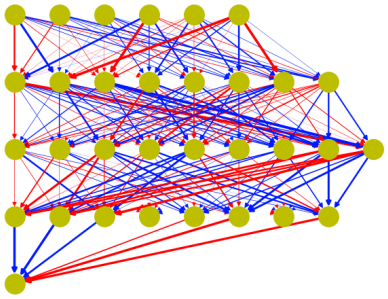


The first neural network we tested had 6 input nodes, 4 nodes in a single hidden layer, and one binary output node. When given test data, the network showed very good results with high precision and recall and an accuracy of 0.93. The confusion matrix also shows mostly true positives and negatives, with very few false ones. Because of the simplicity of this structure, it is apparent how significantly each feature impacts both the positive and negative decision weights when given a car.



The second neural network had the same 6 input nodes, a hidden layer of 6 nodes, a hidden layer of 4 nodes, and a single binary output layer. While it still had high precision and recall, the accuracy was a single hundredth worse than that of the first structure (0.96). While the positive and

negative weights are still fairly traceable, they are much more confusing to figure out.



The third neural network had our 6 input nodes, a hidden layer of 8 nodes, then a hidden layer of 9, then another hidden layer of 8, then the output layer. Interestingly, there were fewer false positives and negatives, but the accuracy was still 0.96 while the first structure gave an accuracy of 0.97. The positive and negative weights are near impossible to trace.

Technical

The dataset was initially downloaded from Kaggle. No cleanup was necessary for the data itself except for the one hot encoding we did to map the categorical data to integers for easier processing. The decision trees were trained and constructed from SckiKitLearn's "tree" module, and visualized with graphviz. The neural networks were constructed from SciKitLearn's neural network MLPClassifier library and visualized with Dr.Edwards' code that used the networkx and colorsys libraries.