CS 5830 - Project 8

Anny Chen,Suyash Mhetre

Introduction

Car evaluation is significant, useful and necessary when buying or selling new cars and used cars. In result, there are plenty of car evaluation websites, such as Kelley Blue Book, Edmunds, NADA and so on, trying to evaluate cars for easy car shopping. Since the necessity of car evaluation, there are plenty of car evaluation datasets in machine learning repositories. We chose one among these datasets to train a classifier which can predict cars' acceptability(unaccepted, accepted, good or very good). We chose decision trees and neural networks methods for car evaluation.

Slides can be found at:

https://docs.google.com/presentation/d/1G5WexSBy1c004Y3DvaWwLWqCVyrBtFiYJ6d8hRywJic/edit#slide=id.p

Dataset

We used a car evaluation dataset from Kaggle. It has 6 attributes viz. Buying, maintenance cost, number of person, number of doors, lug_boot and safety. The target parameter used was the decision column. It is categorical classification whether the car is acceptable, unacceptable, good or a very good situation. It has a 1728 number of rows.The data has ordinal categorical data-type. We need to convert it to numerical scale. Dataset had no empty or NULL values. Hence, a lot of data cleaning was not required.

1. Decision Tree:

Analysis Technique:

A Decision Tree is one of the most popular machine learning algorithms. It uses a tree like structure and their possible combinations to solve a particular problem. It belongs to the class of supervised learning algorithms where it can be used for both classification and regression purposes. But for this project we mainly focused only on classification analysis. Initially, we used all the data without splitting it into train and test data. We can see that figure 1 has depth of 5 and for figure 2 we have kept the depth of 7.

Result:

We observed that for when we increase the depth the accuracy of the decision tree increases. At root level (level 0) the nodes split based on the gini value. The attribute that has the greater gini value will split further. We also got some pure nodes with gini index 0. However at level 4, the nodes split on maintenance cost, lug boot and safety, since decision trees always choose the attribute which split data best at each node. When we increase the depth of decision trees, the bias of data becomes lower while the variance of data becomes higher. In other words, the bias-variance tradeoff depends on the depth of the tree.

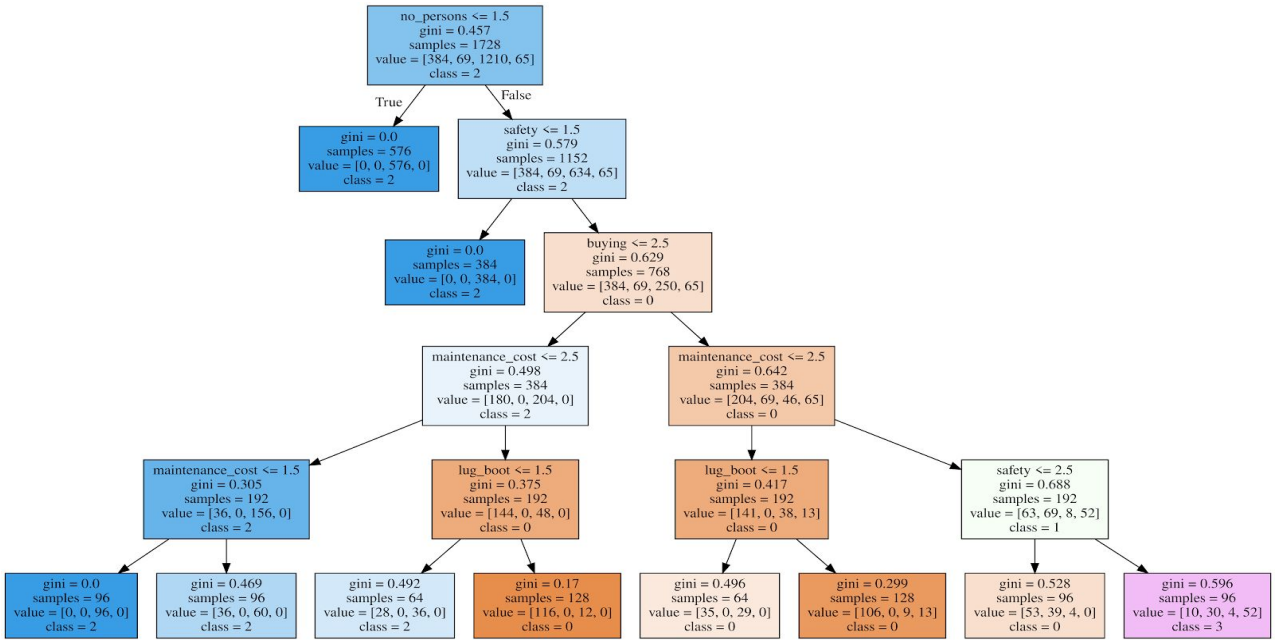


Figure 1

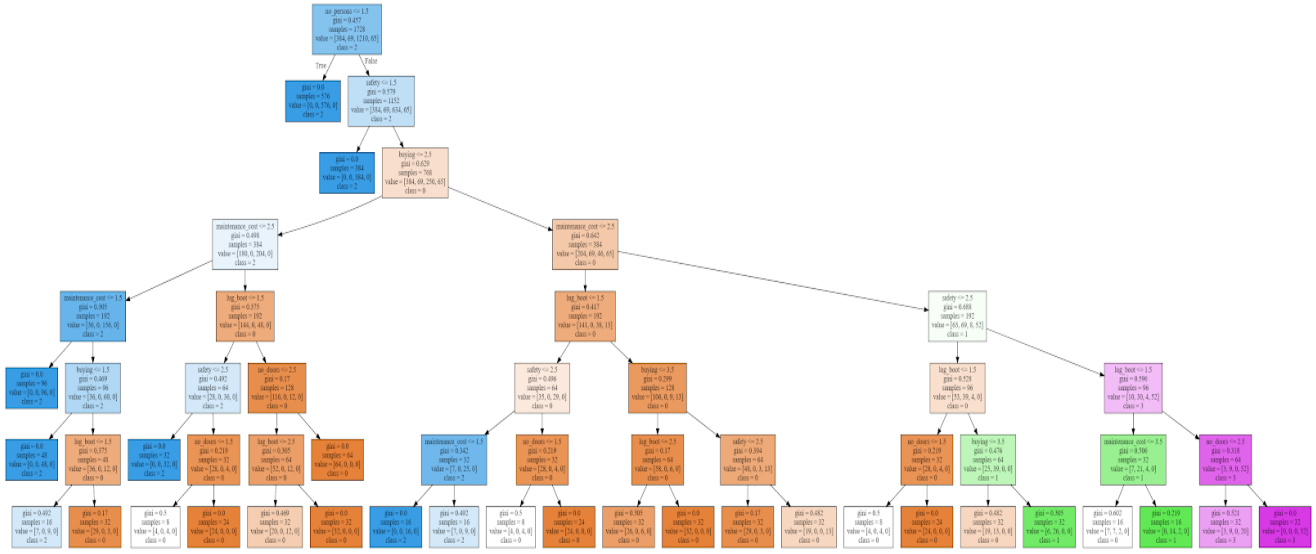


Figure 2

2. Neural Network:

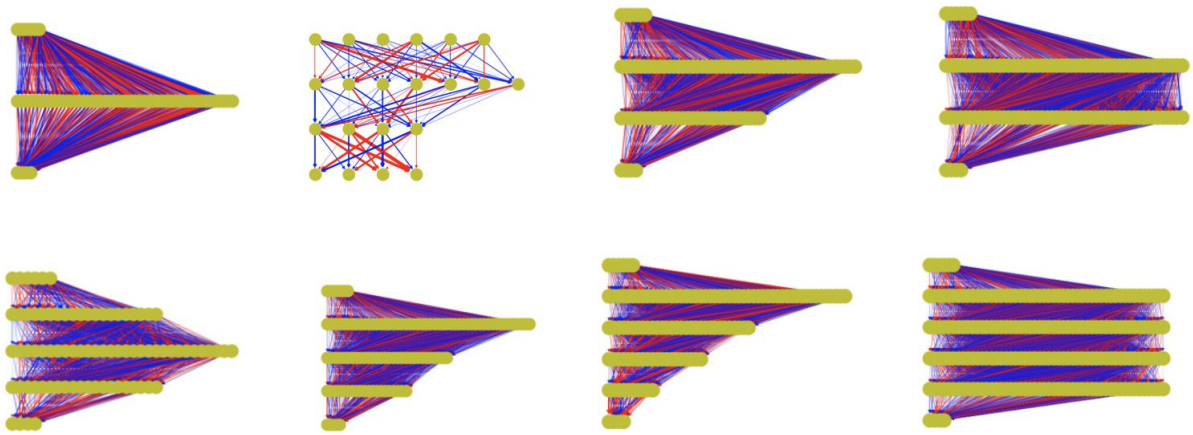
Analysis Technique:

For neural networks, we tried eight different structures with different layers and nodes. First structure only has one hidden layer with 50 nodes, second structure has two hidden layers with 7 and 4 nodes, third structure has two hidden layers with 50 and 30 nodes, fourth structure has two hidden layers with 50 and 50 nodes, fifth structure has three layers with 20, 30 and 20 nodes, sixth structure has three layers with 50, 30 and 20 nodes, seventh structure has four layers with 50, 30, 20 and 10 nodes, last structure has four layers with 50, 50, 50 and 50 nodes.

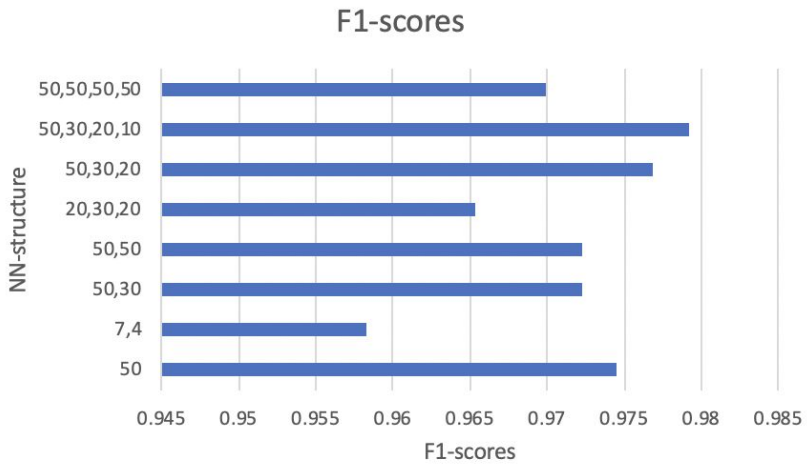
It will be interesting to see the relationship between F1-score and the structure of neural networks as it will tell us whether the depth or width of a neural network will influence the F1-score in positive correlation.

Result:

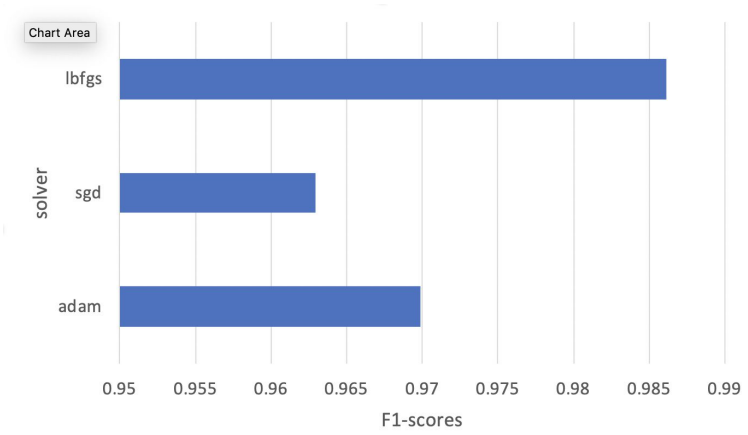
Here is the visualization of eight different neural networks we used for car evaluation.



And the F1-score of each structure:



From the plot we figure out that the width and depth of the network is not positively correlative with F1-scores. It is interesting that increasing the width and depth of the neural network may result in a worse F1-scores. The structure 50-30-20-10 comes out the best F1-scores, which indicates the wedge structure neural network may perform better.



We also change different solvers in neural networks. It turns out lbfgs solver works better on small dataset than adam.

Although the structure of decision trees and the neural networks looks similar, we don't consider there is any correlation. For each node in decision trees, we could point out which features separate these data. For neural networks, however, we could not understand the meanings of each weight or bias. In other words, neural networks are black boxes.