Good Cars’ Criterias

This project is about determining what makes a good car purely based on the dataset and using a tree as a classifier. The dataset contains cars that have specific attributes such as buying price, maintenance cost, number of doors etc… The complete description can be found here

<http://archive.ics.uci.edu/ml/machine-learning-databases/car>

Select car.names to download it. Then to view it, right click on car.names and edit it with notepad.

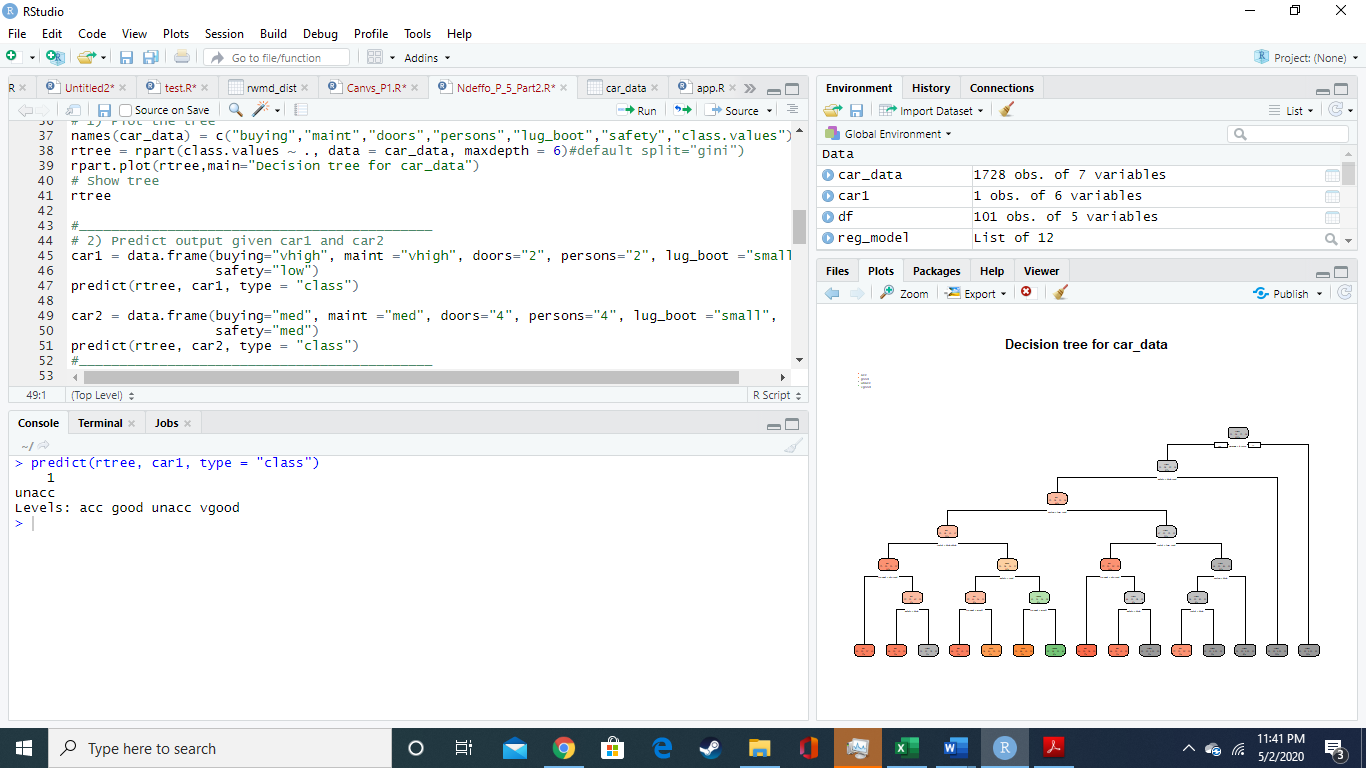
1. **Describing the training tree model and showing the plot**

The tree model assigns for a set of cars in the dataset a class label that rates cars from unacceptable to very good. That class label tells if a given set of attributes of a car match together. For example a car that has buying = vhigh, maint = vhigh, doors = 2, persons = 2, safety = low should have a class label of acceptable. After classification, the plot of the tree can be shown as the following

The maxdepth of the tree has been to set to 4 because higher maxdepth caused smaller plot details of the tree components.

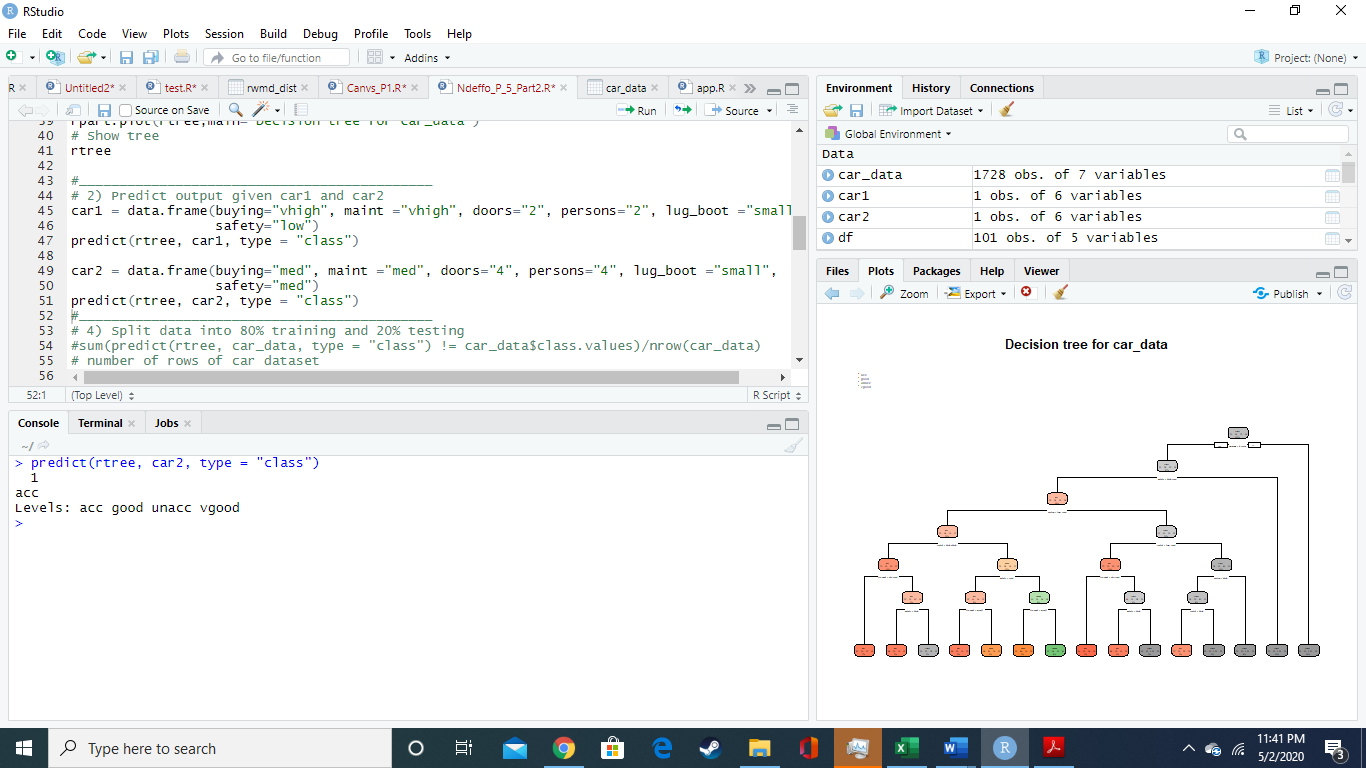
1. **Predicting two cars with the following attributes**
2. performance car: buying="vhigh", maint="vhigh", doors="2", persons=“2", lug\_boot="small", safety="low"

The predicted class label is **unacc** which means **unacceptable** as shown in the picture below



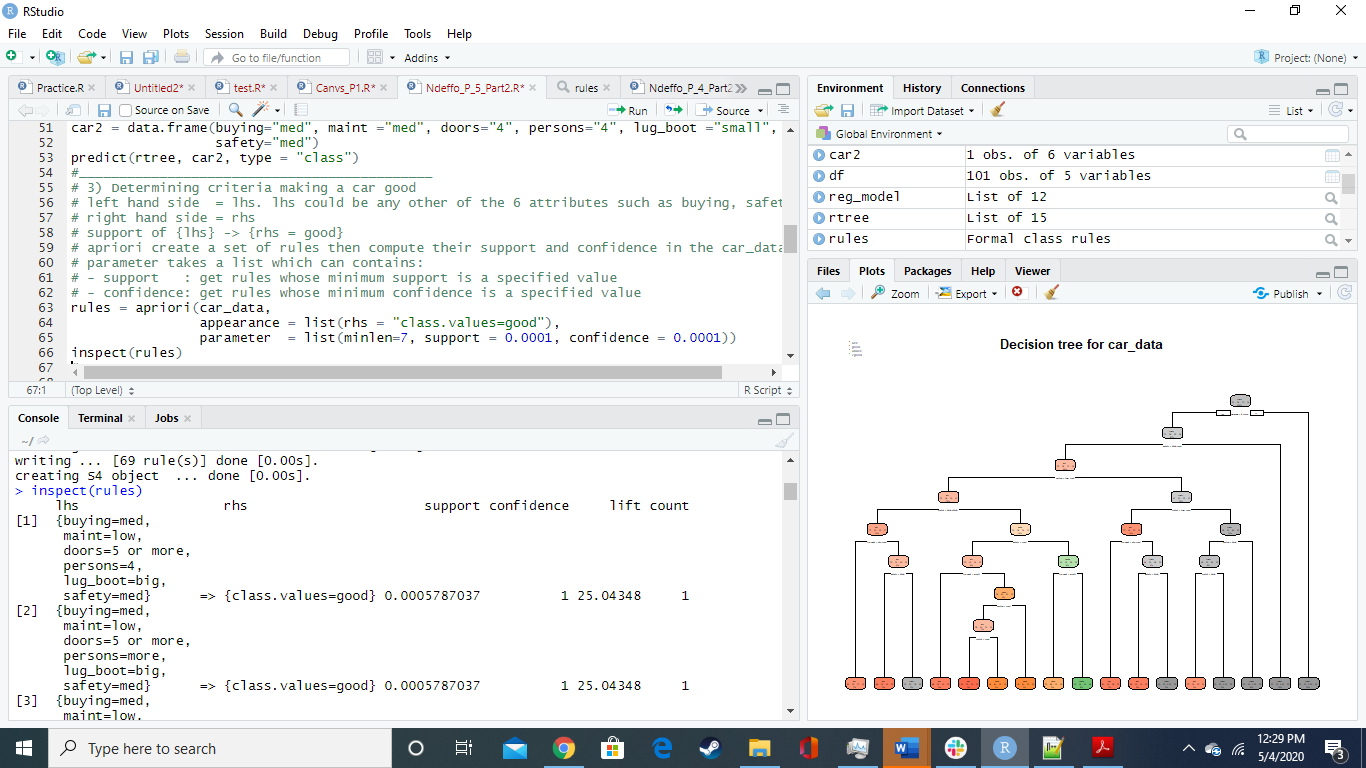
1. compact SUV: buying=“med", maint=“med", doors=“4", persons=“4", lug\_boot=“small", safety=“med"

The predicted class label for this set of attributes is **good**.



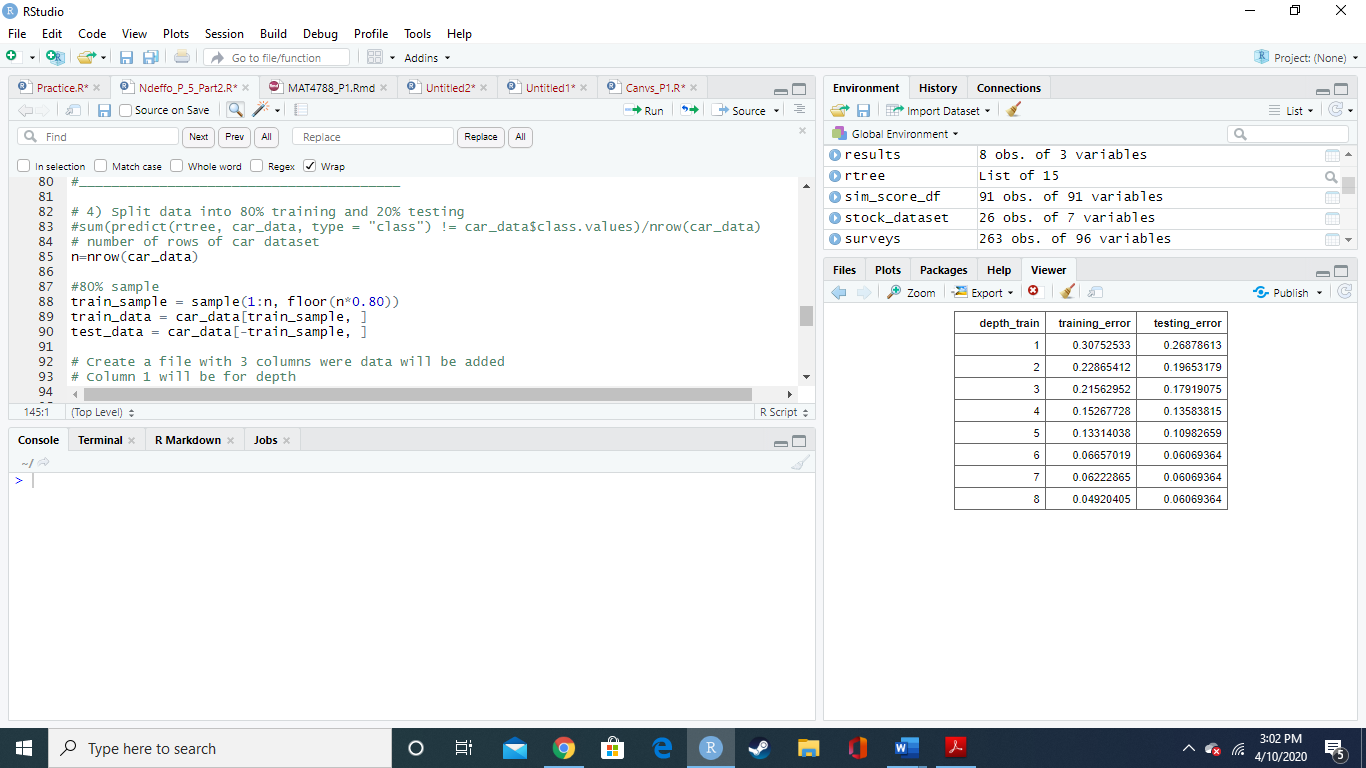
1. **Criteria for a good car based on the model**

A good car should have seats for **4 persons or more, safety = high or medium, buying price = low** or **medium, lug\_boot = small, medium or high** and **maintenance = high** or **very high**. This can be verified by looking for the most recurrent attributes whenever a good car appear in the class.values. We can compute the support and confidence using the following commands

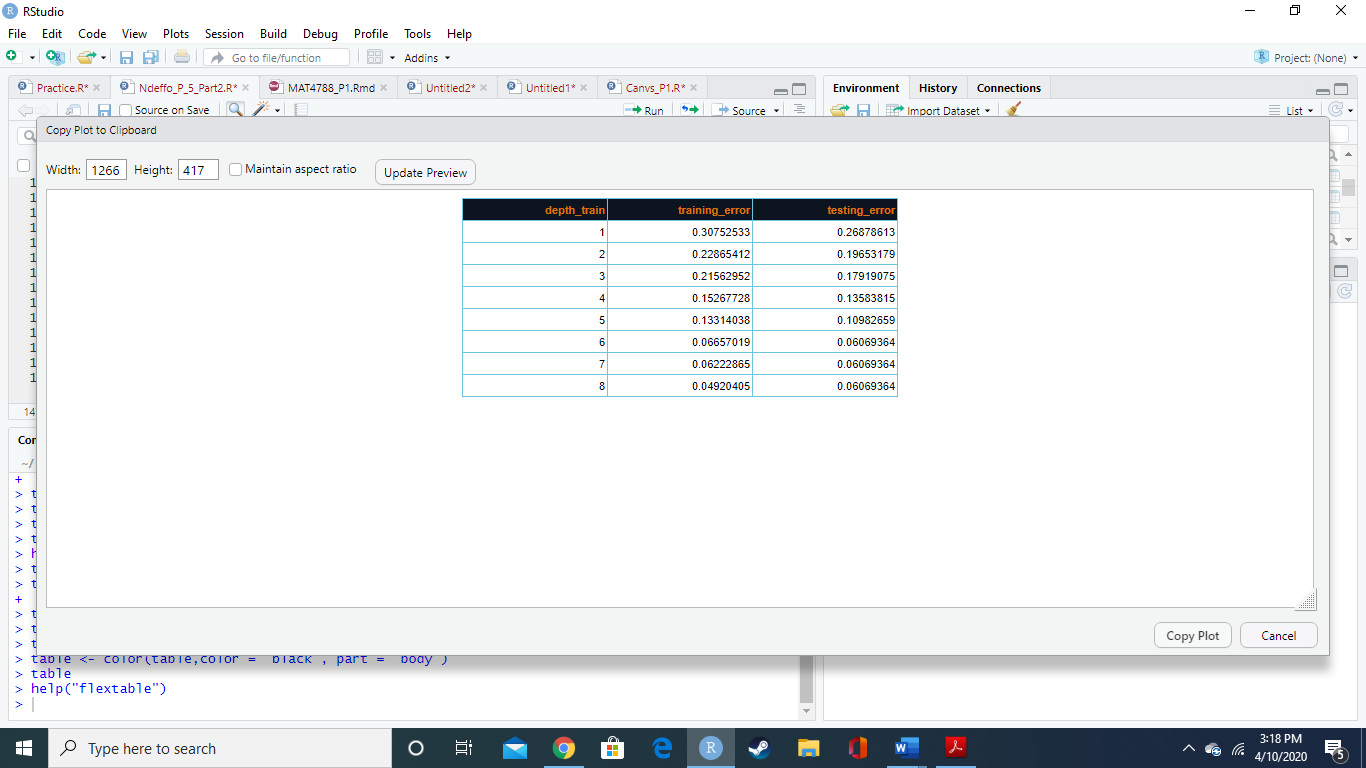


1. **Splitting the dataset into 80% training, 20% testing then constructing a table using different maxdepth to compute training error and testing error**

Splitting the data is done using the following commands



The generated table look like this



1. **Interpreting the results from question 4)**

As it is showed in the table above, as the **maxdepth of tree increases, the training and testing error decreases**.

1. **Finding the 3 main factor in the car evaluation**

The 3 main factors are the number of seats, the safety and the buying price. Those 3 factors are the ones that have the highest splitting errors. For that reason, they are at the top of the tree.