Classifying car price ranges

Sivert M. Skarning

Mai 2019

1 Introduction

1.1 Problem set

Today everything is moving online. Advertising your car by using services online is very common. So how much is your car worth? How do you categorize the price range of your car? This project focuses on how to predict the value of cars, based the car features.

The data set chosen for this project is the car evaluation data set created by Marko Bohanec. This data set contains information about 1728 cars.

- buying: vhigh, high, med, low.
- maint: vhigh, high, med, low.
- doors: 2, 3, 4, 5more.
- persons: 2, 4, more.
- lug-boot: small, med, big.
- safety: low, med, high.
- Acceptability: unnac, acc, good, v-good

Based on maintenance costs, number of doors, how many persons that fits in the car, boot size, acceptability and safety rating we will try to make a model that predicts the buying price of the car. The model will use 6 predictors to classify which class the car is in. The predictors are maintenance, number of doors, number of persons that can fit in the car, luggage size, acceptability and safety.

1.2 Data set

Usage The car evaluation has been referenced in many scientific papers. Amongst the most noteworthy are:

- MML Inference of Decision Graphs with Multi-way Joins and Dynamic Attributes [1]
- Stopping Criterion for Boosting-Based Data Reduction Techniques: from Binary to Multi class Problem [2]
- Impact of learning set quality and size on decision tree performance [3]

The data set was used as an example for displaying multi attribute decision-making [5].

Generation According to Bohanec the Car evaluation data set was created from a hierarchical decision model. This model was created for the demonstration of a decision making software called DEX [4].

2 Classification

2.1 C5.0

I decided to use C5.0 for the classification. It is free and easy use. C5.0 is an algorithm to produce decision trees for classification purposes. For configuring the data and running the algorithm I used R. In R you can use the C5.0 package to generate basic tree-models and rule-based models. In this chapter I will present my findings when using this algorithm on the car evaluation data set. To help me get started with R and the C5.0 algorithm I used this guide [6].

2.2 Findings

When running the C5.0 algorithm on the car evaluation dateset we found that the accuracy was not very high. We got a 55 percent error rate when using all the predictors. The summary of the first run can bee seen in figure 1 and the decision tree generated can be seen in figure 2.

```
Evaluation on training data (1000 cases):

          Decision Tree
        ----------------
        Size      Errors

          22  580(58.0%)   <<


         (a)   (b)   (c)   (d)    <-classified as
        ----  ----  ----  ----
          77    76    28    59    (a): class high
          25   194    20    28    (b): class low
          41   126    35    39    (c): class med
          54    61    23   114    (d): class vhigh


       Attribute usage:

       100.00% unacc
        83.00% safety
        77.70% maint
        37.00% persons
        30.60% lug_boot
```

Figure 1: tree-summary

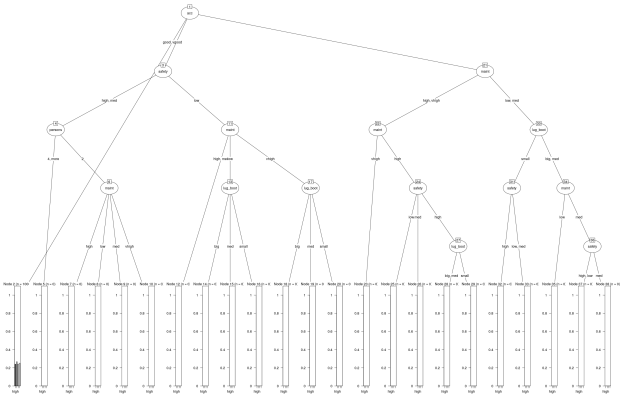

Figure 2: Decision-tree

2.3 Trees and rules

When comparing trees in figure 1 and rules in figure 3 we see that the error rate is similar. The tree has a slightly better error rate. This is expected because rules are a simplified tree. The rules consists of if-then rules. The rules makes it easier to read, however it might yield a slightly worse result.

```
Evaluation on training data (1000 cases):

            Rules
      ----------------
        No      Errors

        11  615(61.5%)   <<


       (a)   (b)   (c)   (d)    <-classified as
      ----  ----  ----  ----
        67     6     9   158    (a): class high
        22   103         142    (b): class low
        35    63    11   132    (c): class med
        48               204    (d): class vhigh


     Attribute usage:

     100.00% acc
      27.80% safety
      25.60% maint
      13.80% persons
      12.80% lug_boot
```

Figure 3: Rules

Rules When generating rules from the tree model we get 11 rules or 11 paths through the tree. Some rule are not very sensible like rule number 10

which states that if the car acceptability is unacceptable then the price range is very high. Some rules are more logical like rule 5 that states that if it has medium safety, a small boot size and its in acceptable condition it will be be priced in a low price range. Rule 9 also states that if the maintenance costs are medium, it has a big boot and its in acceptable condition it will be priced very high which is very true when it comes to big station wagons or SUV's. This means that some of the rules have logic that someone with domain knowledge can recognize.

2.4 Testing the model

When I started the building of the model i split the data set. A thousand rows of data was reserved for training the data and 728 rows was reserved for testing. When using the predict function C5.0 on the test data set we get the results displayed in figure 4.

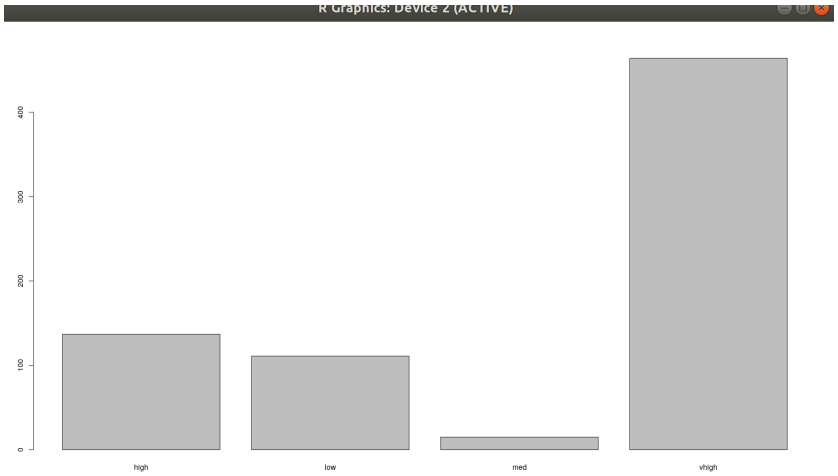

Figure 4: Prediction on training set with the rule model

As you can see in figure 4 the number of cars classified in the v-high price range, way exceed the number of cars that exceed the medium price range. This is expected when you look at the rules of the model. Three ways thorough the tree gives a classification of v-high, but only one classifies it as medium.

Figure 5 shows the actual distribution.

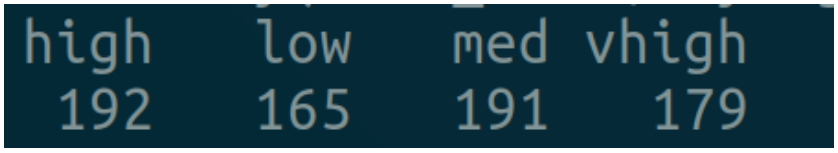

Figure 5: Actual distribution of test data

2.5 Winnowing, boosting and pruning

Boosting Boosting is an algorithm where multiple model are created. When classifying, all of the models vote for which classification the row will be. I tried to enable boosting for my data set, but it showed up with an error that said that the last classifier was not accurate enough.

Winnowing Winnowing is an algorithm that exclude attributes that does not provide any value able information. When i applied the C5.0 algorithm with winnowing enabled it winnowed three attribute, the number of doors, the number of persons that can fit in the car and the boot size. This had a negative effect on the model and the error rate increased with 4.3 percent.

3 Result

In this project i hopes to achieve much greater accuracy than what I did. I have tried the C5.0 algorithm with several configurations and additional algorithms. The best result I was able to achieve was 58 percent error rate. 58 percent in this domain is not satisfactory for the use cases of this model. I have worked systematically Troy try to improve it without no real breakthrough. In order to improve my model I will try other algorithms like XgBoost. I will also try other data set about cars in order to get more predictors to work with.

References

[1] peter J Tan, David L Towe, 2003 *MML Inference of Decision Graphs with Multi-way Joins and Dynamic Attributes* .

[2] Marc Sebban, Richard Nock, Stéphane Lallich, 2002 *Stopping Criterion for Boosting-Based Data Reduction Techniques: from Binary to Multi-class Problems.*

[3] M. Sebban, R. Nock2, J.H. Chauchat and R. Rakotomalala *Impact of learning set quality and size on decision tree performance*

[4] Marko Bohanec *http://archive.ics.uci.edu/ml/datasets/Car+Evaluation*

[5] Marko Bohanec, Vladislav Rajkovic *Knowledge acquisition and explanation for multi-attribute decision making*

[6] Rulequest research *C5.0: An Informal Tutorial*