Brian Holliday

Professor Li

Intro to Data Mining

2 May 2020

Project 5b

1.

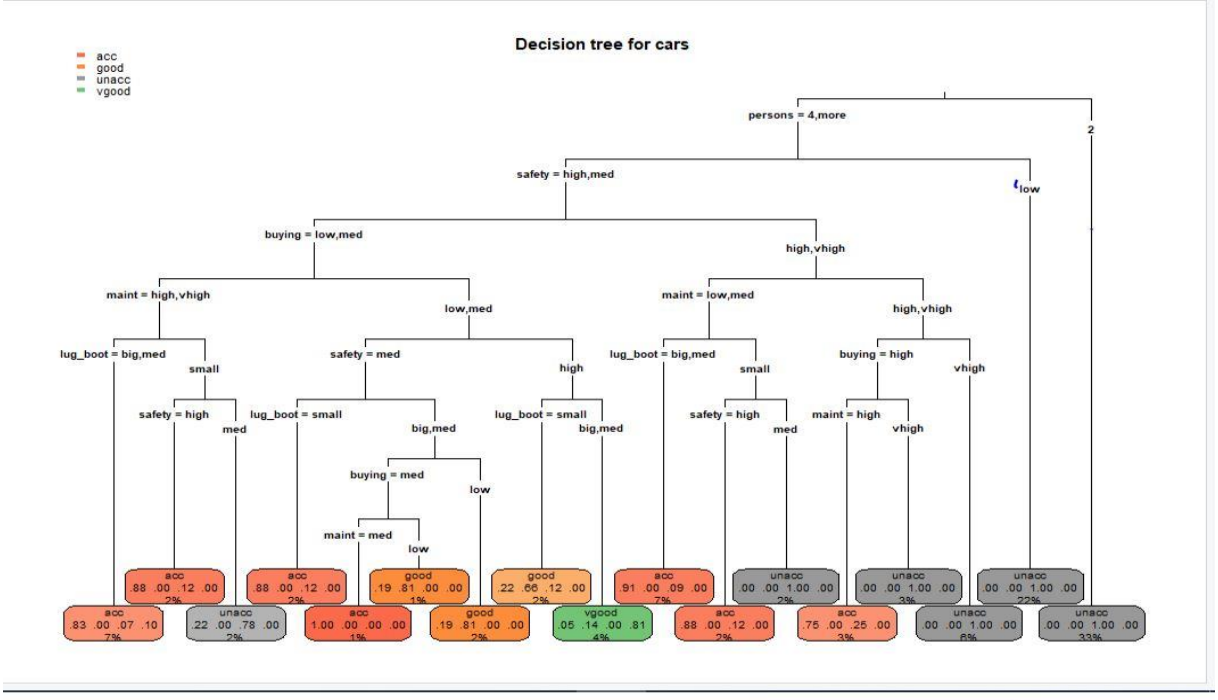
Describe the trained tree model and plot the tree.

Figure 1: Car Tree Code

```
#Project 5B

#Number 1

library(rpart)
library(rpart.plot)

cars = read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data",
                  sep = ",", header = FALSE)

names(cars) = c('buying', 'maint', 'doors',
                'persons', 'lug_boot', 'safety', 'car_evaluation')

ctree = rpart(car_evaluation ~ ., data = cars, main = 'Decision tree for Car Evaluation')  #de
ctree

#Plot the tree
rpart.plot(ctree, main="Decision tree for cars",type = 3)
```

This trained decision tree is for a dataset about evaluating cars on a criterion of six categories. Two price categories, the buying price and the maintenance price. The four technical categories being number of doors, number of persons, lug-boot size and estimated safety. If we use the tree starting with the person category, we can evaluate different types of cars to get to a car evaluation. There are four different grades for the car evaluation; unacceptable, acceptable, good, and very good.

Figure 2: Cars Decision Tree



2.

Predict the evaluation for the following two types of cars:

a) performance car: buying="vhigh", maint="vhigh", doors="2", persons="2", lug_boot="small", safety="low"

b) compact SUV: buying="med", maint="med", doors="4", persons="4", lug_boot="small", safety="med"

The performance car has buying="vhigh", maint="vhigh", doors="2", persons="2", lug_boot="small" and safety="low". According to the decision tree in Figure 2, the evaluation for the performance car is unacceptable. The tree suggests that any car that only seats two people is going to get a grade of unacceptable, with 33 percent of the data points seating only two people. Therefore, the performance car is unacceptable because it has two persons.

The compact SUV has buying ="med", maint="med", doors="4", persons="4", lug_boot="small" and safety="med". According to the decision tree in Figure 2, the evaluation for the compact SUV is acceptable. This type of car is rare in this dataset only accounting for 2 percent of the data rows.

3.

Based on the trained model, what is the criteria for being a very good car?

According the tree there is only one way to get a very good car; buying = low/med, maint = low/med, persons = 4/more, lug_boot = big/med and safety = high. This type of car only accounts for 4 percent of the data. This criteria also describes a large suv/mini-van type of vehicle. This vehicle must be of low to medium price, with good storage space,

4.

Split the data into training data (80%) and testing data (20%) . On the same sample data set, compute the training and testing errors for the tree models with depth = 1, 2, 3, …, 7,8. Construct a table to exhibit your result.

Figure 3: Data Split Code

```
testing <- rep(1:8)
training <- rep(1:8)

#maxdepth = 1
train_sample = sample(1:nrow(cars), floor(nrow(cars)*0.80))
train_data = cars[train_sample, ]
test_data = cars[-train_sample, ]

#get tree for train and test
fit_tree_train_1 = rpart(car_evaluation ~ ., data = train_data, maxdepth = 1)
fit_tree_test_1 = rpart(car_evaluation ~., data = test_data, maxdepth = 1)

#predict the training and testing data
train_predict_1 = predict(fit_tree_train_1, train_data, type = "class")
test_predict_1 = predict(fit_tree_test_1, test_data, type = 'class')

#Get the error for both the training and test
train_error_1 = sum(train_predict_1 != train_data$car_evaluation)/nrow(train_data)
test_error_1 = sum(test_predict_1 != test_data$car_evaluation)/nrow(test_data)
............

#maxdepth = 8

#get tree for train and test
fit_tree_train_8 = rpart(car_evaluation ~ ., data = train_data, maxdepth = 8)
fit_tree_test_8 = rpart(car_evaluation ~., data = test_data, maxdepth = 8)

#predict the training and testing data
train_predict_8 = predict(fit_tree_train_8, train_data, type = "class")
test_predict_8 = predict(fit_tree_test_8, test_data, type = 'class')

#Get the error for both the training and test
train_error_8 = sum(train_predict_8 != train_data$car_evaluation)/nrow(train_data)
test_error_8 = sum(test_predict_8 != test_data$car_evaluation)/nrow(test_data)
```

After entering the testing and training data for each level of the tree, we get a chart that looks like this:

Figure 4: Test and Train Data

	depth	training	testing
1	1	0.30463097	0.28034682
2	2	0.22648336	0.20520231
3	3	0.20911722	0.19942197
4	4	0.15412446	0.12138728
5	5	0.12590449	0.11560694
6	6	0.06657019	0.09826590
7	7	0.05933430	0.08959538
8	8	0.05571635	0.08959538

5.

From the results of the of calculating the error for both the training and the testing data we see that most of the time the training error is slightly higher than the testing error. The outlier in this i s the maxdepth = 2 row. We see that depth 1 through 5 the testing and training error are very clos e, but the calculation for the testing error goes unchanged after maxdepth is equal to 5. This tells us that the depth 5 tree is good enough for the tree because not much important data is missing fr om the tree after depth five.

6.

According to our tree in Figure 2, the biggest factor in the tree is persons or the number of passengers the car can fit. We see that if your car can only seat two people, it is unacceptable and that accounts for 33 percent of the data. Next would be the safety rating. If your car can seat four or more people and the safety is low this accounts for 22 percent of the data. The third most important factor is the buying price which accounts for the remaining 45 percent of the data, if your first two choices are 4 persons and safety high to medium.