DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

RUPNAGAR-140001, INDIA

DLPS REPORT

For

CAR EVALUATION CLASSIFICATION TASK AND EXPLORING DECISION TREE ALGORITHM

Submitted by

Sachin (2018MEB1258)
Tamandeep (2018MEB1271)
Vishal (2018MEB1274)

Supervised by
Dr. Manish Agarwal

Table of Contents

Abstract

Cars are essentially part of our regular day to day life. There are various kinds of cars produced by different manufacturers, consequently the buyers have a decision to make.

When an individual considers buying a car, there are numerous aspects that could influence his/her choice on which kind of car he/she is keen on. The choice buyer or drivers have generally relied upon the price, safety, and how spacious the car is.

Car evaluation database is significant structure information that everyone should take a look at for the car features and is useful in decision making. This dataset is labeled according to the specification of PRICE, COMFORT and SAFETY.

The objective of this report is especially to determine the decision making, identifying the car variables like car price value with other various variables to decide between a good acceptable car from the unaccepted values from the target value.

INTRODUCTION

Generally we need a car as a method for transportation however as we include fun into it and we tend to forget that we shouldn't underestimate i.e. understanding the idea in making a decision on a choice in getting a car..

In present times it is continuously the car sales representative who encourages us to purchase this car or not. We often depend upon the conclusions of our family and companions who had past experience with vehicle inconveniences. We may or probably won't know it consciously however we are basically ignoring the factors that would help us financially, comfortably, and safety in the long run.

In this assignment we process the data, exploring the variables relationship between the attributes and we model the data from different classification models, Decision trees in terms of their best set of parameters for each case and performance on car evaluation data set.

Dataset Attributes

The Car Evaluation dataset contains following concept structure:

CAR- car acceptability.

buying -buying price

maint -price of the maintenance

doors -number of doors

persons- capacity in terms of persons to carry

lug_boot- the size of luggage boot

safety- estimated safety of the car

Attributes

The car directly relies on six attributes :
'buying','maint','doors','persons','lug_boot','safety','classes'.

The dataset contains 1727 instance and possible values each attribute are below

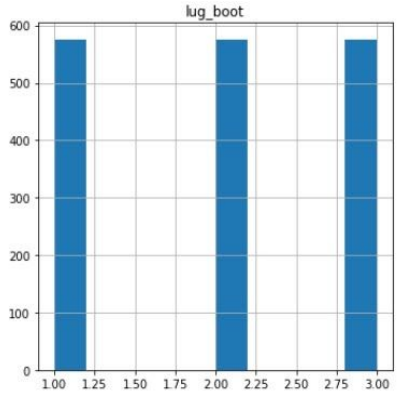
INPUT Attributes

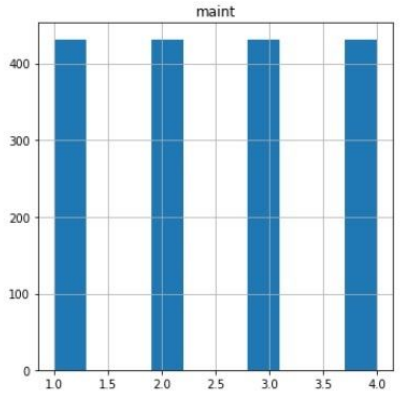
buying	vhigh, high, med,low
maint	Vhigh, high, med, low
doors	2, 3, 4, 5more
persons	2, 4, more
lug_boot	Small, med, big
safety	Low, med, high

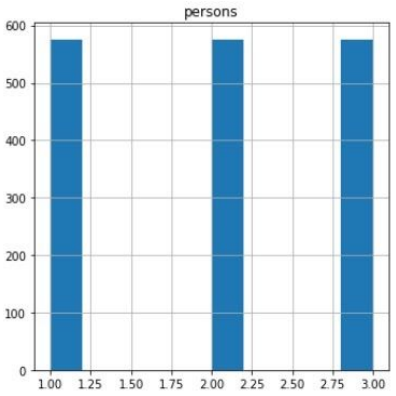
The data analysis is done on this dataset to identify some patterns and also attributes range with their Percentages(frequency).

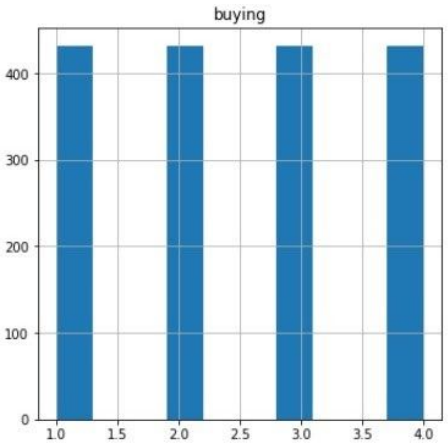
classes	Number of observation per class	Percentage
unacc	1209	70.023
acc	384	22.222%
good	69	3.993%
vgood	65	3.762%

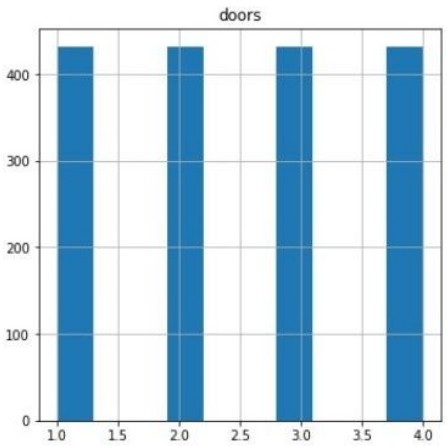
Data Observation

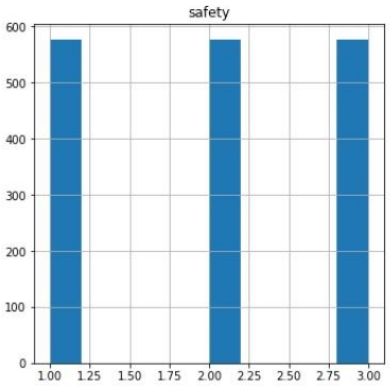












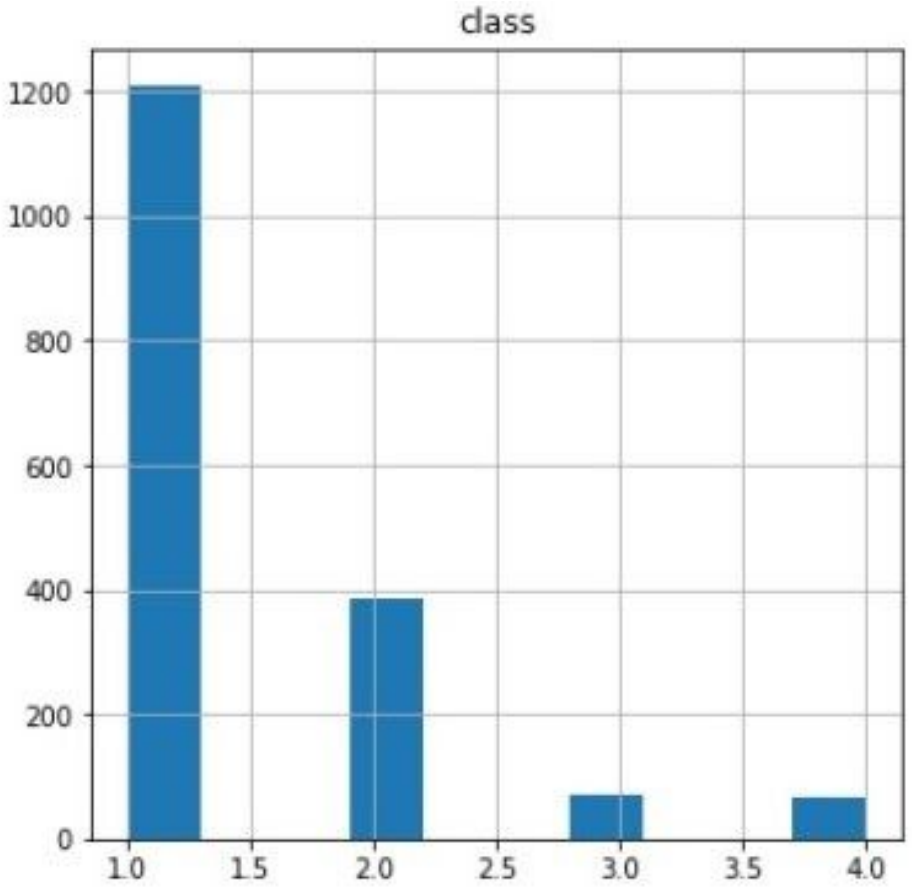
	buying	maint	doors	persons	lug_boot	safety	class
count	1728	1728	1728	1728	1728	1728	1728
unique	4	4	4	3	3	3	4
top	med	med	5more	2	med	med	unacc
freq	432	432	432	576	576	576	1210

```
df.head(5)
```

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Above graph which give the number of count (unique values in the column) vs the classes

From the given graph result almost 70% of cars are in classes unacceptable(unacc), which means it skewed left distribution.



In the above graph, out of the total 1727 instances of car in the dataset 1209(70%) were unacceptable, 384(22%) were acceptable, 69(3.9%) were in good condition and 65(3.7%) were in very good condition. From the graph we can come to the conclusion that more than half of the cars evaluated were not in acceptable.

Methodology

Encoding Data

Encoding data means to convert categorical features (e.g. unacc, acc, good, vgood) into numerical i.e. (1,2,3,4).

Decision trees

Decision tree is a module that uses a tree-like-graph or module of condition of decisions and their possible consequences. It is one approach to display an algorithm that contains only conditional control statements.

It follows a flowchart like structure in each internal node that is condition on each attribute, each branch represents the outcome of the condition, and each leaf node represents a class table. The top down approach from the root to the leaf represents classification rules.

Root Node:

This Node represents the total population (instances) and further breakdown into branch class sub-nodes based on the conditions.

Decision node

When a sub node gets divided into further sub nodes then its called decision node

Leaf node

When node cannot split further into sub nodes

Accuracy Test

Accuracy: The measurement of correct classifications / the total amount of classifications.

Train accuracy: The accuracy of a model on samples it was constructed on.

Test accuracy: The accuracy of a model on samples it hasn't seen.

Code Explanation

Initially, we read the data from the car data file, converting it into proper format. Then we uploaded it to 'Pandas'. As our data contains a lot of categorical features that too in string, so we converted it into the numerical form. After summarizing the data, we plotted histogram for all the different features. Using correlation matrix, we then found out the correlation in the data.

In order to create a tree, we first created a function which is recursive in nature. This function takes three inputs i.e. data, done (whichever inputs are done), head (pointer of the current node). Then this function call 'fun' function in order to take the remaining inputs i.e. maximum information gain for the remaining inputs and the probabilities in order to check

that if the probability of any output is not more than 0.95. If it encounters that the probability is more than 0.95 then it will create the leaf node, and if not then it will find out the input with maximum information gain and of that input, it will create a node and consequently four children which will be called recursively.
It has a few exceptional cases-

1. If the input is used then whichever output has highest probability, it will create its leaf node.
2. Some outputs have three children but in order to make it generic we have taken four children for all of the nodes.Hence, there are some nodes which are not possible, so for that error value is stored and then the node is converted into leaf node.
3. There is another function called the dtree function which takes data(input), head(pointer). It will check all the input nodes and will send it to any of its children in the tree. We have taken the leaf node's value to be negative because whenever it will encounter a negative value it will get to know that it's a leaf node and then it will multiply it with -1 and then return the value.

The fun function that we have described earlier is used to return the maximum information gain and the probabilities. This function first calculates the probability and then the entropy. Using this entropy data is split on the basis of different features it has; if a particular column has three different features then the data will split into three and if the column has four different features then the data will split into four. Then the entropy is calculated and consequently information gain. Comparing the information gain that we have calculated the function returns the maximum of those values which has the maximum information gain.

At last, we calculate the accuracy test for the code described above. Accuracy test contains training accuracy as well as test accuracy.

Result

29.237346501126616-test error

29.39853004111403- training error

Remarks

Due to lack of resources we have taken the probability to be greater than 0.75, then it took a lot of time to compile and hence we got test error and training error almost the same. But if we take the probability to be greater than 0.95 instead of 0.75, the accuracy of our code will increase and hence we will get desired results accurately.

References

1. https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
2. https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293