Decision Trees

Author: Rodriguez Noh Santiago Miguel
Professor: Ph.D. Anabel Martin Gonzalez
Link to code: https://github.com/Santiagomrn/Decision_Trees.git

I. Introduction

A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A. Theoretical framework

Entropy:
Entropy basically tells us how impure a collection of data is.

$$Entropy(S) = -(P(yes)log_2P(yes) + P(no)log_2P(no)) \tag{1}$$

Information Gain:
The measure we will use called information gain, is simply the expected reduction in entropy caused by partitioning the data set according to this attribute. The information gain (Gain(S,A) of an attribute A relative to a collection of data set S, is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v\epsilon Values(A)} \frac{|Sv|}{S} Entropy(Sv) \tag{2}$$

II. Decision Trees

A. Using the training data, construct a decision tree for the binary classification of customers ofthe restaurant "Mama's Pasta" into 'Satisfied' or 'Unsatisfied'. Use the Information Gain (IG) as the decision criterion to select which attribute to split on. Show your calculations for the IG for all possible attributes for every split.

	OVERCOOKED_PASTA	WAITING_TIME	RUDE_WAITER	SATISFIED
0	yes	long	no	yes
1	no	short	yes	yes
2	yes	long	yes	no
3	no	long	yes	yes
4	yes	short	yes	no

Fig. 1. Training set.

Selection of the root:

```
  OVERCOOKED_PASTA WAITING_TIME RUDE_WAITER SATISFIED
0              yes         long          no       yes
1               no        short         yes       yes
2              yes         long         yes        no
3               no         long         yes       yes
4              yes        short         yes        no
OVERCOOKED_PASTA  : 0.4199730940219749
WAITING_TIME  : 0.01997309402197489
RUDE_WAITER  : 0.17095059445466854
best gain:  0.4199730940219749 best feature : OVERCOOKED_PASTA
```

Fig. 2. Compare gains.

The algorithm determined that the best characteristic for the root is OVERCOOKED_PASTA.
Now looking at the table and relating OVERCOOKED_PASTA = yes

```
  WAITING_TIME RUDE_WAITER SATISFIED
0         long          no       yes
2         long         yes        no
4        short         yes        no
WAITING_TIME  : 0.2516291673878229
RUDE_WAITER  : 0.9182958340544896
best gain:  0.9182958340544896 best feature : RUDE_WAITER
```

Fig. 3. Compare gains.

The second and last division occurs with the RUDE_WAITER characteristic.
As a result I get the following decision tree.

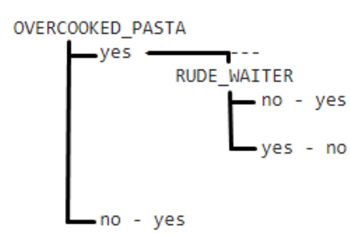

Fig. 4. Decision tree.

After observing the decision tree we notice that the WAITING_TIME variable is not part of the tree, and this is because it does not provide enough information to make a prediction, in this way the algorithm discarded it.

B. Now use the decision tree you have created to predict whether each of the test users will be satisfied or not after their visit to "Mama's Pasta".

Person ID	Overcooked pasta?	Waiting time	Rude waiter?
6	No	Short	No
7	Yes	Long	Yes
8	Yes	Short	No

Fig. 5. Test data.

```
{'OVERCOOKED_PASTA': 'no', 'WAITING_TIME': 'short', 'RUDE_WAITER': 'no'}
SATISFIED  : yes

{'OVERCOOKED_PASTA': 'yes', 'WAITING_TIME': 'long', 'RUDE_WAITER': 'yes'}
SATISFIED  : no

{'OVERCOOKED_PASTA': 'yes', 'WAITING_TIME': 'short', 'RUDE_WAITER': 'no'}
SATISFIED  : yes
```

Fig. 6. Predictions.

III. Final Comments

A. other decision tree

In order to verify the correct operation of the algorithm, I also generated the decision tree of the material seen in class and I obtained the following.

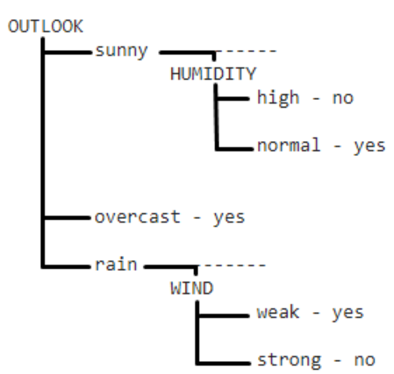

Fig. 7. Decision tree.

For the development of this practice use recursive functions, I think it is something that should not be overlooked because recursion is one of the most complicated issues when learning to program, something that caught my attention about decision trees, is the possibility that some variables of your data set are not found in the final decision tree, that is, they are not considered when making a prediction.

References

[1] Pranto, B. (2020, 4 marzo). *Entropy Calculation, Information Gain Decision Tree Learning.*, https://medium.com/analytics-vidhya/entropy-calculation-information-gain-decision-tree-learning-771325d16f.