Iterative Dichotomiser 3 Algorithm to predict the Mode of Transport

FARSHANA FATHIMA-COE17B001 # DIKSHANYA LASHMI R-CED17I008

INTRODUCTION

This article will talk about the prediction of Mode of Transport used by a person based on various elements that influence the decision making process using the ID3(Iterative Dichotomiser 3)-Decision Tree Algorithm.

DECISION TREE

Decision tree learning algorithm has been successfully used in expert systems in capturing knowledge. The main task performed in these systems is using inductive methods to the given values of attributes of an unknown object to determine appropriate classification according to decision tree rules.It is one of the most effective forms to represent and evaluate the performance of algorithms, due to its various eye catching features: simplicity, comprehensibility, no parameters, and being able to handle mixed-type data. There are many decision tree algorithm available named ID3, C4.5, CART, CHAID, QUEST, GUIDE, CRUISE, and CTREE

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision .Decision tree are commonly used for gaining information for the purpose of decision - making. Decision tree starts with a root node on which it is for users to take actions. From this node, users split each node recursively according to decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome .ID3 are basically most common decision tree algorithms in data mining

A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called “root” that has no

incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attributes values. Each leaf is assigned to one class representing the most appropriate targetvalue. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path.

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision .Decision tree are commonly used for gaining information for the purpose of decision -making. Decision tree starts with a root node on which it is for users to take actions. From this node, users split each node recursively according to decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome.

ID3

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy

H (S) or the information gain I G (S) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split or partitioned by the selected attribute to produce subsets of the data

---

FORMULAS

Entropy

Entropy H (S) is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$H(S) = \Sigma_{x \in X}(-p(x).log_2 p(x))$$

Where,

- S – The current dataset for which entropy is being calculated
  - This changes at each step of the ID3 algorithm, either to a subset of the previous set in the case of splitting on an attribute or to a "sibling" partition of the parent in case the recursion terminated previously.
- X – The set of classes in S
- p (x) – The proportion of the number of elements in class x to the number of elements in set S.

In ID3, entropy is calculated for each remaining attribute. The attribute with the **smallest** entropy is used to split the set S on this iteration. Entropy in information theory measures how much information is expected to be gained upon measuring a random variable; as such, it can also be used to quantify the amount to which the distribution of the quantity's values is unknown.

A constant quantity has zero entropy, as its distribution is perfectly known. In contrast, a uniformly distributed random variable (discretely or continuously uniform) maximizes entropy. Therefore, the greater the entropy at a node, the less information is known about the classification of data at this stage of the tree; and therefore, the greater the potential to improve the classification here.

As such, ID3 is a greedy heuristic performing a best-first search for locally optimal entropy values. Its accuracy can be improved by preprocessing the data.

Information gain

Information gain I G (A) is the measure of the difference in entropy from before to after the set S is split on an attribute A . In other words, how much uncertainty in S was reduced after splitting set S on attribute A .

$$IG(S,A) = H(S) - \Sigma_{t \in T} p(t).H(t)$$
$$= H(S) - H(S/A)$$

Where,

- H (S) – Entropy of set S
- T – The subsets created from splitting set S by attribute A such that S = ∪ t ∈ T t
- p (t) The proportion of the number of elements in t to the number of elements in set S
- H (t) – Entropy of subset t

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the **largest** information gain is used to split the set S on this iteration.

DATASET

gender	car ownership	travel cost	income level	FINAL
male	0	cheap	low	bus
male	1	cheap	medium	bus
female	1	cheap	medium	train
female	0	cheap	low	bus
male	1	cheap	medium	bus
male	0	standard	medium	train
female	1	standard	medium	train
female	1	expensive	high	car
male	2	expensive	medium	car
female	2	expensive	high	car

ALGORITHM

PSEUDOCODE OF ID3

ID3 (Examples, Target_Attribute, Attributes)

Create a root node for the tree

If all examples are positive, Return the single-node tree Root, with label = +.

if all examples are negative, Return the single-node tree Root, with label = -.

If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.

Otherwise Begin

A ← The Attribute that best classifies examples.

Decision Tree attribute for Root = A.

For each possible value, v_i, of A,

Add a new tree branch below Root, corresponding to the test A = v_i.

Let Examples(v_i) be the subset of examples that have the value v_i for A

If Examples(v_i) is empty

Then below this new branch add a leaf node with label = most common target value in the examples

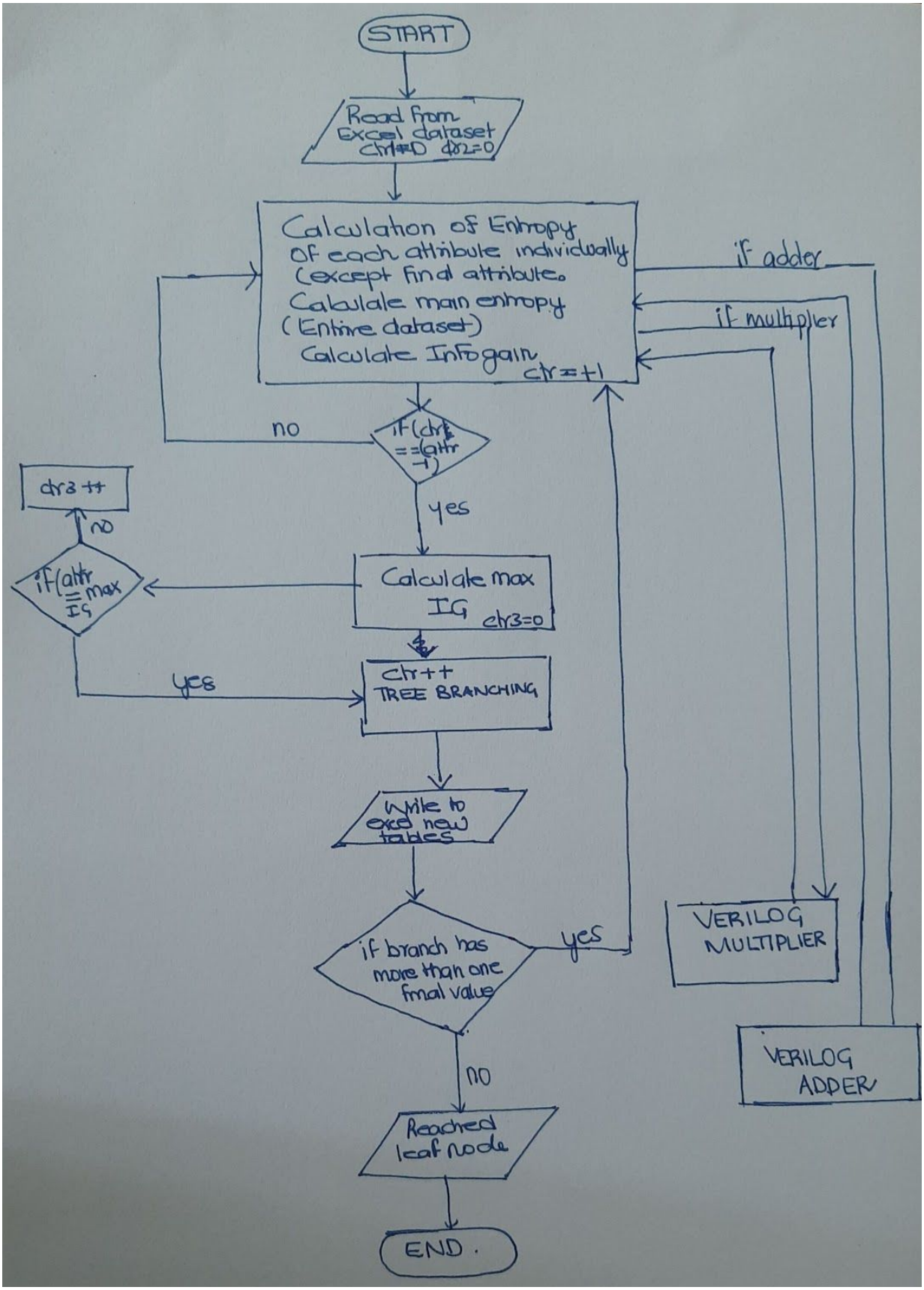
Else

below this new branch add the subtree ID3 (Examples(v_i), Target_Attribute, Attributes – {A})

End

Return Root

FLOW CHART OF THE CODE ALGO



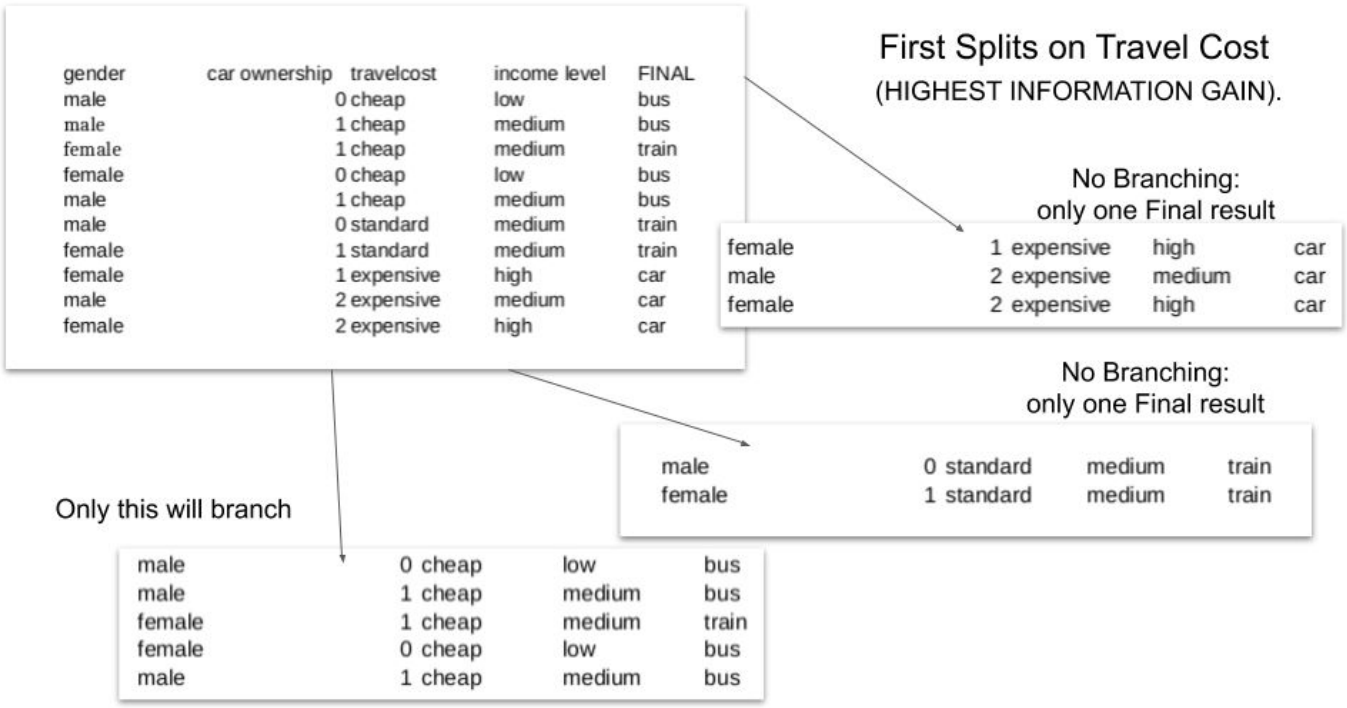
BRANCHING

Decision:

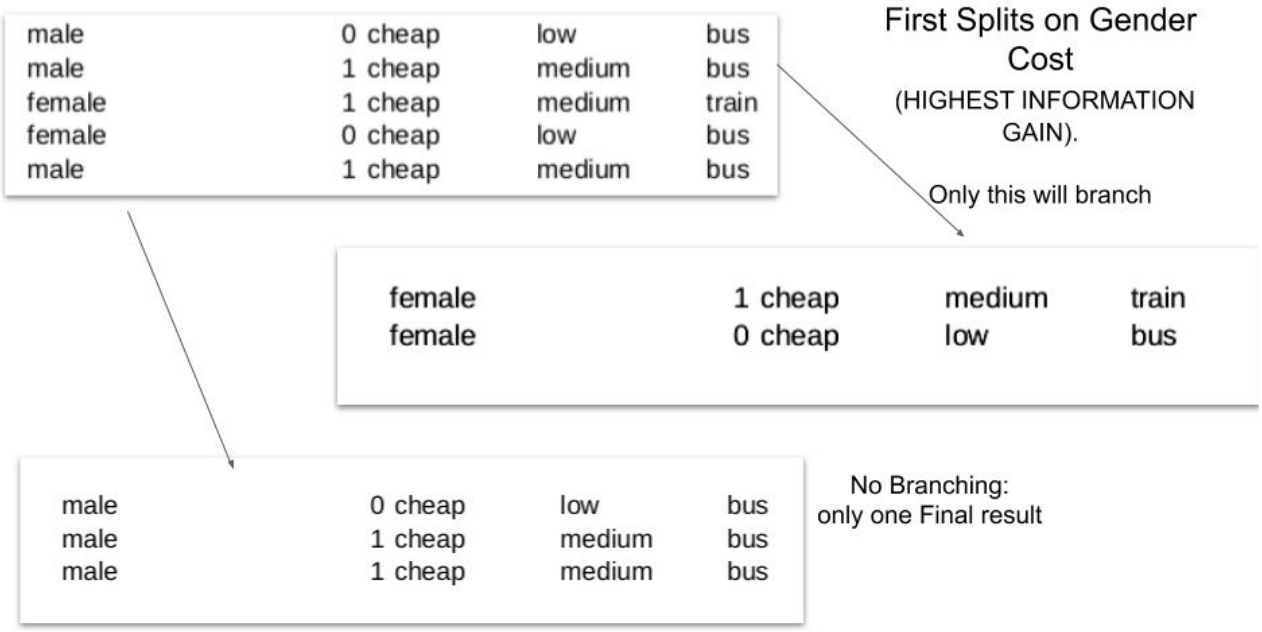
The attribute having the highest information gain or lowest entropy(the attribute that gives least impurity in terms of data set) will branch based on its label

Tree Branching of the given Dataset:

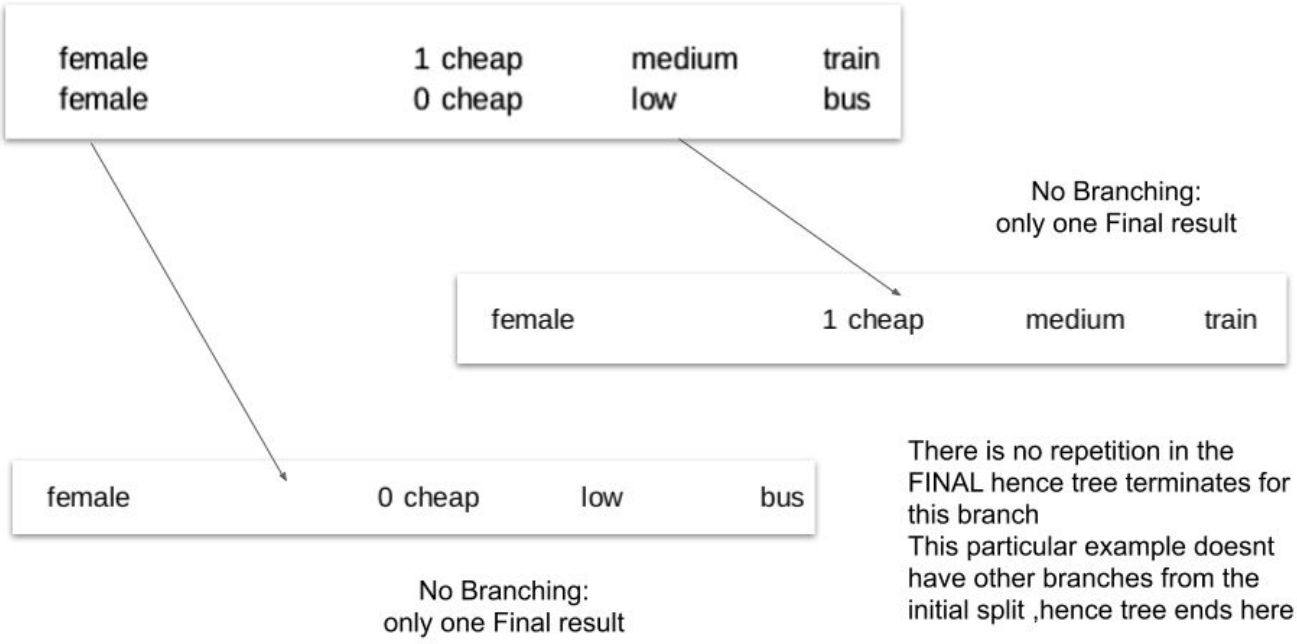
ITERATION ONE:



ITERATION TWO:



ITERATION THREE:



EXIT:

No more branching hence exit

VERILOG MODULES:

ADDER:

```
def adder(a,b):
    k=convert(a)
    l=convert(b)
    aq="a="+str(k)
    bq=" +b="+str(l)
    x="./fulladder32_tb +"+aq+" +"+bq+" +c=0"
    s=subprocess.check_output(x,shell=True)
    y=str(s).strip("b\'res=    ")
    y=y.strip("\\n")
    y=reconverta(y)
    return float(y)
```

MULTIPLIER:

```
def multi(a,b):
    k=convert(a)
    l=convert(b)
    aq="a="+str(k)
    bq=" +b="+str(l)
    x="./wallace32b_tb +"+aq+" +"+bq
    s=subprocess.check_output(x,shell=True)
    y=str(s).strip("b\'res=    ")
    y=y.strip("\\n")
    y=reconvertm(y)
    return float(y)
```

USAGE of adder and multiplier:

PER ITERATION:

If n_i is the no of labels in an attribute and t is the number of attributes in the iteration

Then,

No of adders: Entropy: $n_1 + n_2 + \ldots\ldots + n_t$

$$= \Sigma_{i \in t} n_i$$

Information Gain: $n_1 + n_2 + \ldots\ldots + n_t$

$$= \Sigma_{i \in t} n_i$$

Main Entropy: 1

Total: Adder: $= 2.(\Sigma_{i \in t} n_i) + 1$

No of Multipliers: Information Gain: $n_1 + n_2 + \ldots\ldots + n_t$

$$= \Sigma_{i \in t} n_i$$

Total: Multiplier: $= \Sigma_{i \in t} n_i$

LIMITATIONS OF ID3:

1. Result of the algorithm may not be true for all cases, but it is shown that ID3 shows 95% times the correct result.
2. Can be prejudiced based on the given dataset
3. does not guarantee an optimal solution
4. When all the attributes are used and if there is still some randomness in the data ,the algorithm finds the most probable of the outputs and displays as the outputs

Hardware Limitations:

Implementation of division has proven to be difficult ,has code requires high degree of precision

BIBLIOGRAPHY:

1. https://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

2. https://en.m.wikipedia.org/wiki/ID3_algorithm

3. https://www.slideshare.net/HARDIKSINGH7/id3-algorithm-56632554