International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 27-31

Car Data Analysis by Using ID3 Algorithm

Aung Cho[1], Aung Si Thu[2]

[1,2]Leturer, ITSM,University of Computer Studies (Maubin), UCS(Maubin), Maubin, Myanmar

Email: Am6244052@gmail.com[1], Htinlutt2016@gmail.com[2]

Abstract— *Todays car buyers need to know the correct news of car before buying. ID3 algorithm can analyze car data to show the buyers the correct prediction of news of car they specified so that they can decide good buying or not. For ID3 algorithm, 'car.data' file was used and it's features are buying, maint, doors, persons, lug_boot, safety and it's target is Decision. The values of target are 'unacc', 'acc', 'good' and 'vgood'. As first step, the given data is trained to produce rule and in second step use the rule to test the sample data for prediction of output target. The 'car.data' file was downloaded from google.com and python code was used for implementation of data analysis.*

Keywords— ID3; Training; Testing; Entropy & Information Gain

1.Introduction

The important data of car such as buying, maint, doors, persons, lug_boot, safety should be known by car buyers to optimize for their using car. The needs of them can be helped by ID3 algorithm. ID3 algorithm can classify and predict data for many fields. This paper used car.data file which includes 1729 records and it's features are buying, maint, doors, persons, lug_boot, safety and target is Decision. In data analysis, first step is to train data for producing rule and second is to test data for output target by using the rule. In this paper includes two tables such as car.data table and output rule table. For implementation data analysis, python code was used.

2. ID3 Algorithm[1]

There are various decision tree algorithms, namely, ID3 (Iterative Dichotomiser 3), C4.5 (successor of ID3), CART (Classification and Regression Tree), CHAID (Chi-square Automatic Interaction Detector), MARS. This article is about a classification decision tree with ID3 algorithm.

One of the core algorithms for building decision trees is ID3 by *J. R. Quinlan*. ID3 is used to generate a decision tree from a dataset commonly represented by a table. To construct a decision tree, ID3 uses a top-down, greedy search through the given columns, where each column (further called ***attribute***) at every tree node is tested, and selects the attribute that is best for classification of a given set. To decide what attribute is best to select to construct a decision tree, ID3 uses ***Entropy*** and ***Information Gain***.

2.1Entropy & Information Gain[1]

Entropy *(E)*

Entropy is the measure of the ***amount of uncertainty*** or ***randomness*** in data. Intuitively, it shows predictability of a certain event. If an outcome of an event has a probability of 100%, the entropy is zero (no randomness exists), and if an outcome is 50%, the entropy takes the maximum value (i.e. equals to 1 since it is the log base 2) as it projects perfect randomness. For example, consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. The entropy here is the highest possible value (i.e., equals 1), since there's no chance to precisely determine the outcome. Alternatively, consider a coin which has heads on both the sides, the outcome of such an event can be predicted perfectly since we know beforehand that it will always be heads. In other words, this event has ***no randomness***, hence its entropy is zero. ***ID3 follows the rule: a branch with an entropy of 0 is a leaf node (endpoint). A branch with an entropy more than 0 needs further splitting.*** In case it is not possible to achieve zero entropy in the leaf nodes, the decision is made by the method of a **simple majority**.

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

1. Entropy *E(S)* using the frequency table of one attribute, where *S* is a current state (existing outcomes) and *P(x)* is a probability of an event *x* of that state *S*:

$$E(S) = \sum_{x \in X} -P(x) \log_2 P(x) \qquad (1)$$

2. Entropy *E(S, A)* using the frequency table of two attributes - *S* and *A*, where *S* is a current state with an attribute *A* (existing outcomes with an attribute *A*), *A* is a selected attribute, and *P(x)* is a probability of an event *x* of an attribute *A*.

$$E(S,A) = \sum_{x \in X}[P(x) * E(S)] \qquad (2)$$

E(S) is the Entropy of the entire set, while the second term *E(S, A) relates* to an Entropy of an attribute A.

Information Gain *(IG)*[1]

Information gain (also called as Kullback-Leibler divergence) denoted by *IG(S, A)* for a state *S* is the ***effective change in entropy*** after deciding on a particular attribute *A*. It measures the relative change (decrease) in entropy with respect to the independent variables, as follows:

$$IG = E(S) - E(S,A) \qquad (3)$$

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about selecting each attribute (*A*) to calculate Information Gain and finding such an attribute that returns the highest IG (i.e., the most homogeneous branches). This attribute will be the next decision node for the tree.

ID3 Algorithm will perform following tasks recursively:[1]

1.Create a root node for the tree

2.If all examples are positive, return leaf node 'positive'

3.Else if all examples are negative, return leaf node 'negative'

4.Calculate the entropy of current state *E(S)*

5.For each attribute, calculate the entropy with respect to the attribute '*A*' denoted by *E(S, A)*

6.Select the attribute which has the maximum value of *IG(S, A)* and split the current (parent) node on the selected attribute

7.Remove the attribute that offers highest IG from the set of attributes

8.Repeat until we run out of all attributes, or the decision tree has all leaf nodes.

3.Implementation

car.data(Given Data)

Total records : 1729 in the following textbox.

This textbox size can be extended.

Features are buying,maint,doors,persons,lug_boot,safety.

Target is Decision.

Target values are unacc,acc,good,vgood.

As first step, training the given data to produce rule.

Second step, use the rule and test sample data and predict target value.

buying,maint,doors,persons,lug_boot ,safety,Decision
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc

Table-1 car.data(Given Data)

3.1.Training:

Python Code

```
import Chefboost as cb
import pandas as pd
df=pd.read_csv("dataset/car.data")
print("ID3 for nominal features and target (large data set)")
config = {'algorithm': 'ID3'}
cb.fit(pd.read_csv("dataset/car.data",names=["buying
","maint","doors","persons","lug_boot","safety","Decision"]), config)
```

output accuracy and run time:

```
ID3 for nominal features and target (large data set)
ID3  tree is going to be built...
Accuracy:  100.0 % on  1729  instances
finished in  17.365615367889404  seconds
```

output rule in the following textbox which can be extended and looked.

```
def findDecision(obj): #obj[0]:
buying, obj[1]: maint, obj[2]:
doors, obj[3]: persons, obj[4]:
lug_boot, obj[5]: safety

if obj[5] == 'med':

if obj[3] == '2':

return 'unacc'

elif obj[3] == '4':

if obj[0] == 'low':

if obj[1] == 'low':

if obj[4] == 'small':
```

Table-2 output rule

3.2.Testing:

If features' values are vhigh, vhigh, 2, 2,small,low

Target value is ?

test_instance = ['vhigh', 'vhigh', '2', '2','small','low']

prediction = cb.predict(model, test_instance)

prediction

'unacc'

Next tests can calculated as the above code.
For examples:
If features' values are low,low,5more,4,small,med
Target value is acc.
If features' values are low,low,5more,4,med,med
Target value is good.
If features' values are low,low,5more,4,med,high
Target value is vgood.

4. Conclusion

Today in the world, cars are buying in high rate. But car buyers sometime face with difficulty to use the car they bought because they do not know news of car in detail. This case can be solved by data analyzer called ID3 algorithm so that they can decide to buy the car with correct news of the car before buying. Python code much provides to implementation of ID3 algorithm.

References

1. https://www.thelearningmachine.ai/tree-id3
2. https://github.com/tofti/python-id3-trees

3. https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/
4. https://sefiks.com/2019/08/31/a-begineers-guide-to-decision-trees-in-python/

Author Profile

	**Aung Cho** received the B.A.(Eco) degree from Yangon University in 1987 and M.I.Sc. (Information Science) degree from University of Computer Studies, Yangon in 2001. After got Master degree, I served as a teacher at the software, information science and application departments of the computer universities. I am now with University of Computer Studies, Maubin, Myanmar.
	**Aung Si Thu** received the B.Sc.(Hons)(Chemistry) degree from Magwe University in 2003 and M.I.Sc.(Information Science) degree from University of Computer Studies, Yangon in 2009. After got Master degree, I served as a teacher at the software, information science and hardware departments of the computer universities. I am now with University of Computer Studies, Maubin, Myanmar.