Car Data Analysis by Using ID3 Algorithm

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

[1]ITSM, [2]FCST,UCS, Maubin, Myanmar

[1]am6244052@gmail.com,[2]htinlutt2016@gmail.com

Abstract

ID3 (Iterative Dichotomiser 3) is sub-subject of AI(Artificial Intelligent). Today's car buyers need to know the news of car. ID3 algorithm can analyze car data to show the buyers the prediction of news of car they specified so that they can decide buying or not. With 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 with entropy and information gain methods 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: AI, ID3 algorithm**,** car data prediction, 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 car to buy. 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 of data analysis, python code was used.

2. Background Algorithm [1]

- There are various decision tree algorithms, namely, ID3 which is Iterative Dichotomiser 3, C4.5, successor of ID, CART, Classification and Regression Tree, CHAID, Chi-square Automatic Interaction Detector, MARS. This article is about a classification decision tree with algorithm of ID3.
- One of the core algorithms for building decision trees is ID3 by *J. R. Quinlan*.The ID3 is used to generate a decision tree from a dataset commonly represented by a table. To construct a decision tree, The 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, the ID3 uses ***Entropy*** and ***Information Gain***.

2.1. Entropy & Information Gain[1]

Entropy (E)

The 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:

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

The 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), The attribute A is a selected attribute, and P(x) is a probability of an event x of an attribute A.

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

In this equation 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)

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 the 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 the entropy after a dataset is split on an attribute. The 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). The attribute will be the next decision node for the tree.

2.2. ID3 Algorithm Process[1]

Step1: ID3 Algorithm will perform following tasks recursively:

Step2: Create a root node for the tree

Step3: If all examples are positive, return leaf node ‘positive’.Else if all examples are negative, return leaf node ‘negative’

Step5: Calculate the entropy of current state *E(S)*

Step6: For each attribute, calculate the entropy with respect to the attribute ‘*A*’ denoted by *E(S, A)*

Step7: .Select the attribute which has the maximum value of *IG(S, A)* and split the current (parent) node on the selected attribute

Step8: Remove the attribute that offers highest IG from the set of attributes

Step9: Repeat until we run out of all attributes, or the decision tree has all leaf nodes.

3. Implementation

In car.data(Given Data) includes total records of 1729.
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,sa
fety,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-1car.data(Training Data)

Data ‘car.data’ can be downloaded from google.com.

3.1. Training

Python Code

Line1: import Chefboost as cb

Line2: import pandas as pd

Line3: df=pd.read_csv("dataset/car.data") print("ID3 for nominal features and target (large data set)")

Line4: config = {'algorithm': 'ID3'}

Line5:cb.fit(pd.read_csv("dataset/car.data",names="buying","maint","doors","persons","lug_boot","safety","
Decision"]), config)

Line6: output accuracy and run time

Line7: 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

0utput 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':
return 'acc'
elif obj[4] == 'med':
if obj[2] == '4':
return 'good'
elif obj[2] == '2':
return 'acc'
elif obj[2] == '5more':
return 'good'
elif obj[2] == '3':
return 'acc'
elif obj[4] == 'big':
return 'good'
elif obj[1] == 'vhigh':
if obj[4] == 'small':
return 'unacc'
elif obj[4] == 'med':
if obj[2] == '4':
return 'acc'
elif obj[2] == '2':
return 'unacc'
elif obj[2] == '5more':
return 'acc'
elif obj[2] == '3':
return 'unacc'
elif obj[4] == 'big':
return 'acc'
elif obj[1] == 'med':
if obj[4] == 'small':
return 'acc'
elif obj[4] == 'med':
```

Table-2 output rule

3.2 Testing

If features' values are vhigh, vhigh, 2, 2,small,lowTarget value is ?
Python code

- 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/