**Implementation of ID3 Decision Tree Algorithm**

Project link: <https://colab.research.google.com/drive/1OuRzPT0dzxgwGZJy1hc9Ncowy3Sjs9Vn?usp=sharing>

Dataset:

<https://github.com/ZiqianTu/MachineLearningA2.git>

**Introduction**

Decision tree is a method of machine learning. Decision tree generation algorithms include ID3, C4.5 and C5.0, etc. A decision tree is a tree structure, in which each internal node represents a judgment on an attribute, each branch represents the output of a judgment result, and finally each leaf node represents a classification result. The decision tree in machine learning is a predictive model that represents a mapping between object attributes and object values. Each node in the tree represents the judgment condition of the object attribute, and its branches represent the objects that meet the node conditions. The leaf nodes of the tree represent the prediction results to which the object belongs.

This report is to present the implementation of ID3 decision tree algorithm using Python including parts of ID3 Algorithm, computing Entropy and Information Gain. Besides, this report will provide the Model Evaluation by showing the advantage and disadvantage of my ID3 decision tree algorithm and also, I will prepare a Data Set named “Airbnb price prediction” to test my algorithm in Experiment Design part. The final of this report will evaluate the test of this algorithm and then provide the result of output. This report will also show the code of my algorithm that how can I compute each step.

**ID3 Algorithm**

The algorithm is complete by computing Entropy and Information Gain. There are many of calculation to achieve so it is necessary to import Numpy and Pandas modules for some formula. In addition, this ID3 decision tree algorithm containing creating the Tree Node and structure.

A picture containing graphical user interface, text, application

Description automatically generated

**Entropy**

The meaning of entropy is that a measure of the uncertainty of random variables, and it is the probability contained in all possible values of all data categories in the database. If the entropy value in the data is larger, it means the uncertainty of the random variable is larger. In other words, the smaller the entropy value, the more data information it contains, which means that the predicted result is more accurate.

Let X be a random variable and set the values to be x1, x2, x3, … xn and there are the non-zero probability p1, p2, p3, … pn so that the entropy is:

A close up of a watch

Description automatically generated

According to the equation of entropy, *H(X)* represents a method to compute the uncertainty of X.

After we know the equation of entropy, we need to implement this algorithm in python. Therefore, we have to understand the logic so that it can be implemented in Python.

Text

Description automatically generated

Figure 1 Calculate Entropy

Line 1 – 6 is the code of calculating Entropy. Firstly, we need to confirm the length of the data set if the feature of data set less than 2, it will return 0 means the entropy is 0.

Regarding the *H(X),* the algorithm needs probability of each random variable in data set showing in line 4.

Line 5 is the implementation of the data set entropy shown as equation H(X) Then, return the equation for calEntropy(d).

**Information Gain**

Information gain represents measuring the ability of an attribute (x) to distinguish between samples (y). When a new attribute (x) is added, the change in entropy H(Y) is the information gain. The larger IG(Y|X), the more important variable x is.

*IG(Y,X) = H(Y)-H(Y|X)*

*Text, letter

Description automatically generated*



The Information Gain in ID3 algorithm is implemented in my Colab (Python) ml\_id3.ipynb, line 8 - 15 and 25 – 50, calcInformationGain and makeStructure is the code of implementation of Information Gain.

A close up of text on a screen

Description automatically generated

Line 23 and 39 - 42 shown the base dataset has been split into main group and subgroup so that it can implement the sub-entropy shown on line 12. “a” is assigned to target which is to be predicted.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Line 12 is the implementation of the subgroup entropy in the equation shown as *H(Y).*

Line 13 is the implementation of the sum of Y given an X in the equation *H(Y*|*X).*

Line 16 is returning the entropy - split entropy so that it should represent Information Gain in equation shown as *IG(Y,X).*

Line 26 – 35 represent the determination of attribute which is needed to implement in information Gain. The dataset feature should be unique in most situation.

**Model Evaluation**

The model is evaluated on “Airbnb price prediction” Dataset. The model is named “ml\_id3.ipynb”.

**Data Preparation**

The “Airbnb price prediction” Dataset” has been import to the notebook by using import pandas as pd. This dataset has 74,111 attribute and 25 features including ID. According to the meaning of the entropy, the value becomes smaller and the predicted result is more accurate. Thus, the dataset needed to be pre-processing. After I use some code to find the values which is null, I found that ‘host\_response\_rate’ and ‘review\_scores\_rating’ have the most missing values and those do not affect so much in dataset so I decided to remove both these features. ‘amenities’ is also removed due to so many attributes.

**Experiment Design and Evaluation**

After determining the dataset to implement my ID3 classifier, the original dataset has been split into 2 parts which are for training and another for testing. Obviously, the subgroup data is used for training and main group data is used for testing. The purpose of this classifier is to use a value in one of the features to determine(predict) another feature. The criteria of this classifier are

1. if the result comes up correctly and the code is matching to the equation of Entropy and Information Gain.
2. Besides, whether the result shows the condition of Airbnb room owner requires the cleaning fee.

**Evaluation Result**

After I entre the code ‘df[df['city'] == 'NYC']['cancellation\_policy']’to know the whole New York City Airbnb of cancellation policy, the result will show below, and it should be correct.

Graphical user interface, text, website

Description automatically generatedText

Description automatically generated

Also, I also code the condition of cancellation policy which is ‘flexible’, and the total row is match the result above. It means if I want to know the cancellation policy is flexible and it does not require to cleaning fee in New York, only need to compare the ID and we will know the result of cleaning fee.

Text

Description automatically generated

**Conclusion**

The core component of this classifier is the realization of entropy and Information Gain. This will directly affect the entire algorithm logic. If the two parts are calculated incorrectly or the coding is not logical, this will cause the entire algorithm or tree to fail to run. In addition, dividing the data set is also a very important part, because it will directly affect the calculation of information gain. Through the final test, it is found that although the decision tree is a good way to evaluate a set of data, there are still some shortcomings of the decision tree, which is that it does not perform well when dealing with data with relatively strong characteristics.