**Explain decision tree learning algorithm. Apply this algorithm on the following dataset to generate classification rules. Loan application dataset.**

# Loan Application Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Age | Has-job | Own-house | Credit-rating | Class |
| 1 | Young | False | False | Fair | No |
| 2 | Young | False | False | Good | No |
| 3 | Young | True | False | Good | Yes |
| 4 | Young | True | True | Fair | Yes |
| 5 | Young | False | False | Fair | No |
| 6 | Middle | False | False | Fair | No |
| 7 | Middle | False | False | Good | Yes |
| 8 | Middle | True | True | Good | Yes |
| 9 | Middle | False | True | Excellent | Yes |
| 10 | Middle | True | True | Excellent | Yes |
| 11 | Old | False | True | Excellent | Yes |
| 12 | Old | False | False | Good | Yes |
| 13 | Old | True | False | Excellent | Yes |
| 14 | Old | True | False | Good | Yes |
| 15 | Old | False | False | Fair | No |

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node.

During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). This work expanded on earlier work on concept learning systems, described by E. B. Hunt, J. Marin, and P. T. Stone. Quinlan later presented C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared. In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C. Stone) published the book Classification and Regression Trees (CART), which described the generation of binary decision trees. ID3 and CART were invented independently of one another at around the same time, yet follow a similar approach for learning decision trees from training tuples. These two cornerstone algorithms spawned a flurry of work on decision tree induction.

Algorithm: Generate\_decision\_tree.

Generate a decision tree from the training tuples of data partition, D.

\*\*Input:\*\*

- Data partition, D, which is a set of training tuples and their associated class labels;

- attribute\_list, the set of candidate attributes;

- Attribute\_selection\_method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a splitting\_attribute and, possibly, either a split-point or splitting subset.

\*\*Output:\*\* A decision tree.

\*\*Method:\*\*

1. Create a node N;

2. If tuples in D are all of the same class, C, then

- Return N as a leaf node labeled with the class C;

3. If attribute\_list is empty then

- Return N as a leaf node labeled with the majority class in D; // majority voting

4. Apply Attribute\_selection\_method(D, attribute\_list) to find the "best" splitting\_criterion;

5. Label node N with splitting\_criterion;

6. If splitting\_attribute is discrete-valued and multiway splits are allowed (not restricted to binary trees), then

- attribute\_list ← attribute\_list − splitting\_attribute; // remove splitting\_attribute

7. For each outcome j of splitting\_criterion:

- Let Dj be the set of data tuples in D satisfying outcome j; // a partition

- If Dj is empty then

- Attach a leaf labeled with the majority class in D to node N;

- Else attach the node returned by Generate\_decision\_tree(Dj, attribute\_list) to node N;

8. End for;

9. Return N;

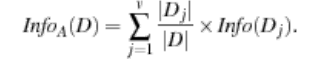
**Information Gain**

ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or “information content” of messages. Let node N represent or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or “impurity” in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found. The expected information needed to classify a tuple in D is given by

****

where pi is the nonzero probability that an arbitrary tuple in D belongs to class Ci and is estimated by . A log function to the base 2 is used, because the information is encoded in bits. Info (D) is just the average amount of information needed to identify the class label of a tuple in D. Note that, at this point, the information we have is based solely on the proportions of tuples of each class. Info (D) is also known as the entropy of D. Now, suppose we were to partition the tuples in D on some attribute A having v distinct values, { }, as observed from the training data. If A is discrete valued, these values correspond directly to the v outcomes of a test on A. Attribute A can be used to split D into v partitions or subsets, { }, where Dj contains those tuples in D that have outcome aj of A. These partitions would correspond to the branches grown from node N. Ideally, we would like this partitioning to produce an exact classification of the tuples. That is, we would like for each partition to be pure. However, it is quite likely that the partitions will be impure (e.g., where a partition may contain a collection of tuples from different classes rather than from a single class).

How much more information would we still need (after the partitioning) to arrive at an exact classification? This amount is measured by



Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A). That is,



**Solution**

**IG OF AGE**

**Entropy of entire dataset S{+9,-6}**

**(9 Yes and 6 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.97

**Entropy of all attributes**

**Entropy of Young{+2,-3} =** 22 = 0.97 let’s denote it with (Y)

**Entropy of Middle{+3, -2} =** 22 = 0.97 let’s denote it with (M)

**Entropy of Old{+4, -1} =** 22 = 0.72 let’s denote it with (O)

**Information Gain of Age: - Entropy(S)-Entropy(Y)-Entropy(M)-Entropy(O)**

**=** 0.97-0.97-0.97-0.72

**Information Gain of Age** = -1.69

**IG OF HAS-JOB**

**Entropy of entire dataset S{+9,-6}**

**(9 Yes and 6 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.97

**Entropy of all attributes**

**Entropy of False{+4,-6} =** 22 = 0.97 let’s denote it with (F)

**Entropy of True{+5, 0} =** 22 = 0 let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex:{+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2,-2} or {+1, -1} the answer would be 1.

**Information Gain of Has-Job: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.97-0.97-0

**Information Gain of Age** = 0

**IG OF OWN-HOUSE**

**Entropy of entire dataset S{+9,-6}**

**(9 Yes and 6 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.97

**Entropy of all attributes**

**Entropy of False{+3,-6} =** 22 = 0.91 let’s denote it with (F)

**Entropy of True{+6, 0} =** 22 = 0 let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex:{+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2,-2} or {+1, -1} the answer would be 1.

**Information Gain of Own-House: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.97-0.91-0

**Information Gain of Own-House** = 0.06

**IG OF CREDIT-RATING**

**Entropy of entire dataset S{+9,-6}**

**(9 Yes and 6 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.97

**Entropy of all attributes**

**Entropy of Fair{+1,-4} =** 22 = 0.72 let’s denote it with (F)

**Entropy of Good{+4, -2} =** 22 = 0.91 let’s denote it with (G)

**Entropy of Excellent{+4, 0} =**  0 let’s denote it with (E)

\*\*Pro-tip\*\*: if 0 consists in the count ex:{+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2,-2} or {+1, -1} the answer would be 1.

**Information Gain of Credit-Rating: - Entropy(S)-Entropy(F)-Entropy(G)-Entropy(E)**

**=** 0.97-0.72-0.91-0

**Information Gain of Credit-Rating** = -0.66

* Gain(S, Age) = -1.69
* Gain(S, Has-Job) = 0
* Gain(S, Own-House) = 0.06
* Gain(S, Credit-Rating) = 0.66

Select the highest value as the root node (Credit-Rating).

Credit-Rating

yes

Fair

Good

Excellent

Notice that Excellent always leads to yes according to table, no matter the attributes of Age, Has-job and Own-house. If the credit-rating is excellent then we must provide the loan.

Now repeat the same process but only considering ‘Fair’ and ‘Good’ values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Credit-rating | Age | Has-job | Own-house | Class |
| Fair | Young | False | False | No |
| Fair | Young | True | True | Yes |
| Fair | Young | False | False | No |
| Fair | Middle | False | False | No |
| Fair | Old | False | False | No |

The entropy of the entire data changes as we are only considering the ‘Fair’ values.

**IG OF AGE**

**Entropy of entire dataset Sfair {+1,-4}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.72

**Entropy of all attributes**

**Entropy of Young {+1, -2} =** 22 = 0.72 let’s denote it with (Y)

**Entropy of Middle {0, -1} =**  let’s denote it with (M)

**Entropy of Old {0, -1} =** 0 let’s denote it with (O)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Age: - Entropy(S)-Entropy(Y)-Entropy(M)-Entropy(O)**

**=** 0.72-0.91-0-0

**Information Gain of Age** = -0.19

The entropy of the entire data changes as we are only considering the ‘Fair’ values.

**IG OF HAS-JOB**

**Entropy of entire dataset Sfair {+1,-4}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.72

**Entropy of all attributes**

**Entropy of False {0, -4} =** 0 let’s denote it with (F)

**Entropy of True {1, 0} =**  let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Has-Job: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.72-0-0

**Information Gain of Has-Job** = 0.72

The entropy of the entire data changes as we are only considering the ‘Fair’ values.

**IG OF OWN-HOUSE**

**Entropy of entire dataset Sfair {+1,-4}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.72

**Entropy of all attributes**

**Entropy of False {0, -4} =** let’s denote it with (F)

**Entropy of True {1, 0} =**  let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Own-house: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.72-0-0

**Information Gain of Own-house** = 0.72

* Gain (SFair, Age) = -0.19
* Gain (SFair, Has-Job) = 0.72



* Gain (SFair, Own-House) = 0.72

Select the highest among these, now choose any one from the two, (For ex:- Has-job)

Credit-Rating

yes

Fair

Good

Excellent

Has-Job

False

True

yes

no

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Credit-rating | Age | Has-job | Own-house | Class |
| Good | Young | False | False | No |
| Good | Young | True | False | Yes |
| Good | Middle | False | False | No |
| Good | Middle | True | True | Yes |
| Good | Old | False | True | Yes |
| Good | Old | True | False | Yes |

The entropy of the entire data changes as we are only considering the ‘Good’ values.

**IG OF AGE**

**Entropy of entire dataset Sgood {+4,-2}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.91

**Entropy of all attributes**

**Entropy of Young {+1, -1} =** 22 = 1 let’s denote it with (Y)

**Entropy of Middle {+1, -1} =**  let’s denote it with (M)

**Entropy of Old {2, 0} =** 0 let’s denote it with (O)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Age: - Entropy(S)-Entropy(Y)-Entropy(M)-Entropy(O)**

**=** 0.91-1-1-0

**Information Gain of Age** = -1.09

**IG OF HAS-JOB**

**Entropy of entire dataset Sgood {+4,-2}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

= 22 = 0.91

**Entropy of all attributes**

**Entropy of False {+2, -2} =** 1 let’s denote it with (F)

**Entropy of True {+2, 0} =**  let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Has-Job: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.91-1-0

**Information Gain of Has-Job** = -0.09

**IG OF OWN-HOUSE**

**Entropy of entire dataset Sgood {+4,-2}**

**(1 Yes and 4 No in the entire dataset, +,- are just for indication not actual operations)**

**= 22 = 0.91**

**Entropy of all attributes**

**Entropy of False {+2, -2} =** let’s denote it with (F)

**Entropy of True {+2, 0} =**  let’s denote it with (T)

\*\*Pro-tip\*\*: if 0 consists in the count ex: {+5, 0} then answer would be zero, similarly if both yes and no are same ex: {+2, -2} or {+1, -1} the answer would be 1.

**Information Gain of Own-house: - Entropy(S)-Entropy(F)-Entropy(T)**

**=** 0.91-1-0

**Information Gain of Own-house** = 0.09

* Gain (SGood, Age) = 0.91
* Gain (SGood, Has-Job) = -0.09
* Gain (SGood, Own-House) = -0.09

The highest value is of Age

As we can see if age is old then it results in yes no matter the values of Age and Has-Job

Credit-Rating

Fair

Good

Excellent

yes

Age

Has-Job

Young

Middle

Old

True

False

yes

no

yes

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Has-Job | Own-House | Class |
| Young | False | False | No |
| Young | True | False | Yes |

The entropy of the entire data changes as we are only considering the ‘Young’ values

**IG OF HAS-JOB**

**Entropy of the entire dataset Syoung{+1, -1} = 1**

**Entropy of all attributes**

**Entropy of False{0, -1} =** 0 let’s denote it with (F)

**Entropy of True{+1, 0} =** 0 let’s denote it with (T)

**Information Gain = Entropy(S)-Entropy(F)-Entropy(T)**

**=** 1-0-0

**Information Gain of Has-Job** = 1

**IG OF OWN-HOUSE**

**Entropy of the entire dataset Syoung{+1, -1} = 1**

**Entropy of all attributes**

**Entropy of False{+1, -1} =** 1 let’s denote it with (F)

**Information Gain = Entropy(S)-Entropy(F)**

**=** 1-1

**Information Gain of Own-House** = 0

* Gain (Syoung, Has-Job) = 1
* Gain (Syoung, Own-House) = 0

The highest value is of Has-Job

Credit-Rating

Fair

Good

Excellent

yes

Age

Has-Job

False

Young

Middle

Old

True

Has-Job

no

yes

yes

False

True

no

yes

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Has-Job | Own-House | Class |
| Middle | False | False | No |
| Middle | True | True | Yes |

The entropy of the entire data changes as we are only considering the ‘Middle’ values

**IG OF HAS-JOB**

**Entropy of the entire dataset Smiddle{+1, -1} = 1**

**Entropy of all attributes**

**Entropy of False{0, -1} =** 0 let’s denote it with (F)

**Entropy of True{+1, 0} =** 0 let’s denote it with (T)

**Information Gain = Entropy(S)-Entropy(F)-Entropy(T)**

**=** 1-0-0

**Information Gain of Has-Job** = 1

**IG OF OWN-HOUSE**

**Entropy of the entire dataset Smiddle{+1, -1} = 1**

**Entropy of all attributes**

**Entropy of False{0, -1} =** 0 let’s denote it with (F)

**Entropy of True{+1, 0} =** 0 let’s denote it with (T)

**Information Gain = Entropy(S)-Entropy(F)**

**=** 1-0-0

**Information Gain of Own-House** = 1

* Gain (Syoung, Has-Job) = 1
* Gain (Syoung, Own-House) = 1

Both are same, so choose any one (Ex:- Has-Job)

Credit-Rating

Fair

Good

Excellent

yes

Age

Has-Job

False

Young

Middle

Old

True

Has-Job

no

yes

yes

False

True

no

yes