TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

----------o0o----------

Báo cáo môn học: Khai phá dữ liệu

Nhóm 20

ĐỀ TÀI: Ứng dụng thuật toán ID3 xử lý dữ liệu trò chơi Tic Tac Toe

Sinh viên thực hiện:

Họ và tên	Mã sinh viên	Lớp
Nguyễn Quang Phúc	191200803	CNTT3 – K60
Vũ Trung Hiếu	191202485	CNTT3 – K60
Phạm Anh Tuấn	191201089	CNTT3 – K60
Nguyễn Trọng Cường	191204096	CNTT3 – K60

Giảng viên hướng dẫn: ***Nguyễn Quốc Tuấn***

MỤC LỤC

I. Phân lớp dữ liệu

1. Khái niệm

Ngày nay phân lớp dữ liệu (classification) là một trong những hướng nghiên cứu chính của khai phá dữ liệu. Thực tế đặt ra nhu cầu là từ một cơ sở dữ liệu với nhiều thông tin ẩn con người có thể trích rút ra các quyết định nghiệp vụ thông minh. Phân lớp và dự đoán là hai dạng của phân tích dữ liệu nhằm trích rút ra một mô hình mô tả các lớp dữ liệu quan trọng hay dự đoán xu hướng dữ liệu tương lai. Phân lớp dự đoán giá trị của những nhãn xác định (categorical label) hay những giá trị rời rạc (discrete value), có nghĩa là phân lớp thao tác với những đối tượng dữ liệu mà có bộ giá trị là biết trước. Trong khi đó, dự đoán lại xây dựng mô hình với các hàm nhận giá trị liên tục. Ví dụ mô hình phân lớp dự báo thời tiết có thể cho biết thời tiết ngày mai là mưa, hay nắng dựa vào những thông số về độ ẩm, sức gió, nhiệt độ, … của ngày hôm nay và các ngày trước đó.

2. Quá trình phân lớp dữ liệu

Kĩ thuật phân lớp được tiến hành bao gồm 2 bước: Xây dựng mô hình và sử dụng mô hình.

- Xây dựng mô hình: là mô tả một tập những lớp được định nghĩa trước trong đó: mỗi bộ hoặc mẫu được gán thuộc về một lớp được định nghĩa trước như là được xác định bởi thuộc tính nhãn lớp, tập hợp của những bộ được sử dụng trong việc sử dụng mô hình được gọi là tập huấn luyện. Mô hình được biểu diễn là những luật phân lớp, cây quyết định và những công thức toán học.
- Sử dụng mô hình: Việc sử dụng mô hình phục vụ cho mục đích phân lớp dữ liệu trong tương lai hoặc phân lớp cho những đối tượng chưa biết đến. Trước khi sử dụng mô hình người ta thường phải đánh giá tính chính xác của mô hình trong đó: nhãn được biết của mẫu kiểm tra được so sánh với kết quả phân lớp của mô hình, độ chính xác là phần trăm của tập hợp mẫu kiểm tra mà phân loại đúng bởi mô hình, tập kiểm tra là độc lập với tập huấn luyện.

3. Các thuật toán phân lớp

- Phân lớp với cây quyết định (decision tree)
- Phân lớp với Naïve Bayesian
- Phân lớp với k phần tử gần nhất (k-nearest neighbor)
- Phân lớp với máy vector hỗ trợ (SVM)
- Phân lớp với mạng neural (neural network)
- Phân lớp dựa trên tiến hoá gen (genetic algorithms)

- Phân lớp với lý thuyết tập thô, tập mờ (rough sets)
- Phân lớp với lý thuyết tập mờ (fuzzy sets)

II. Phân lớp dữ liệu bằng cây quyết định

1. Khái niệm

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như nhị phân (binary) , định danh (nominal), thứ tự (ordinal), số lượng (quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là binary hoặc ordinal. Cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

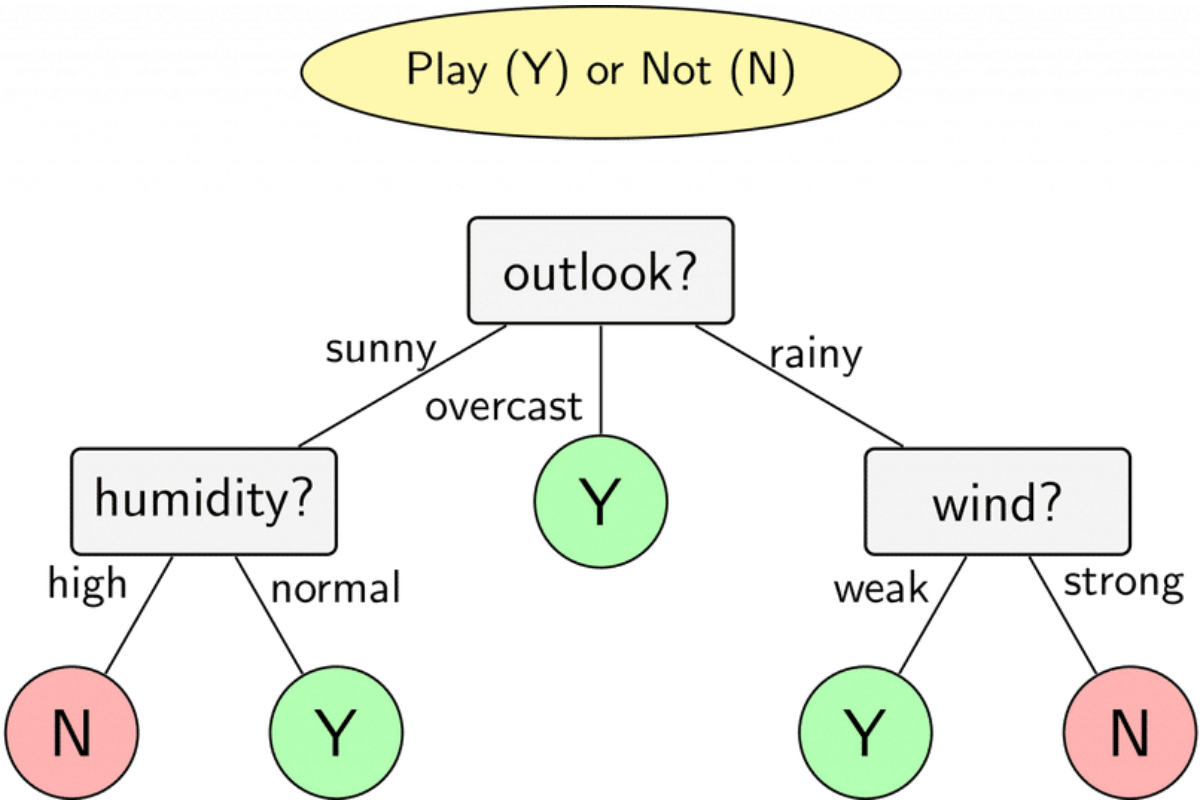
2. Ví dụ

Xét một ví dụ về cây quyết định. Giả sử dựa theo thời tiết mà các bạn nam sẽ quyết định đi đá bóng hay không?

Những đặc điểm ban đầu là:

- Thời tiết
- Độ ẩm
- Gió

Dựa vào những thông tin trên, ta có thể xây dựng được mô hình như sau:



Mô hình cây quyết định

Dựa theo mô hình trên, ta thấy: Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

3. Ưu điểm và nhược điểm

3.1. Ưu điểm

- Cây quyết định là thuật toán đơn giản và phổ biến.
- Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
- Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.
- Có thể làm việc với cả dữ liệu số và dữ liệu phân loại.
- Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
- Có khả năng làm việc với dữ liệu lớn.

3.2. Nhược điểm

- Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
- Cây quyết định hay gặp vấn đề overfitting (Là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến).

III. Thuật toán ID3 (Iterative Dichotomiser 3)

1. Khái niệm thuật toán ID3

Giải thuật ID3 (thường được gọi tắt là ID3) được phát triển bởi J. R. Quinlan trong AI và Breiman, Friedman, Olsen và Stone trong thống kê. ID3 sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. Đây là một giải thuật học khá đơn giản nhưng lại đem lại rất nhiều thành công trong các lĩnh vực khác nhau.

ID3 là một giải thuật hay ho, bởi vì những lý do:

- Cách biểu diễn tri thức học được của nó.

- Tiếp cận trong việc quản lý tính phức tạp.

- Heuristic dùng cho việc chọn lựa các khái niệm ứng viên.

- Tiềm năng đối với việc xử lý dữ liệu nhiễu.

ID3 biểu diễn các khái niệm (concept) ở dạng các cây quyết định (decision tree). Biểu diễn này cho phép xác định phân loại một đối tượng bằng cách kiểm tra các giá trị của nó trên một số thuộc tính nào đó.

Như vậy, nhiệm vụ của giải thuật ID3 là học cây quyết định từ một tập các ví dụ rèn luyện (training example) hay còn gọi là dữ liệu rèn luyện (training data).

- Input: Một tập hợp các ví dụ. Mỗi ví dụ bao gồm các thuộc tính mô tả một tình huống, hay một đối tượng nào đó, và một giá trị phân loại của nó.

- Output: Cây quyết định có khả năng phân loại đúng đắn các ví dụ trong tập dữ liệu rèn luyện, và hy vọng là phân loại đúng cho cả các ví dụ chưa gặp trong tương lai.

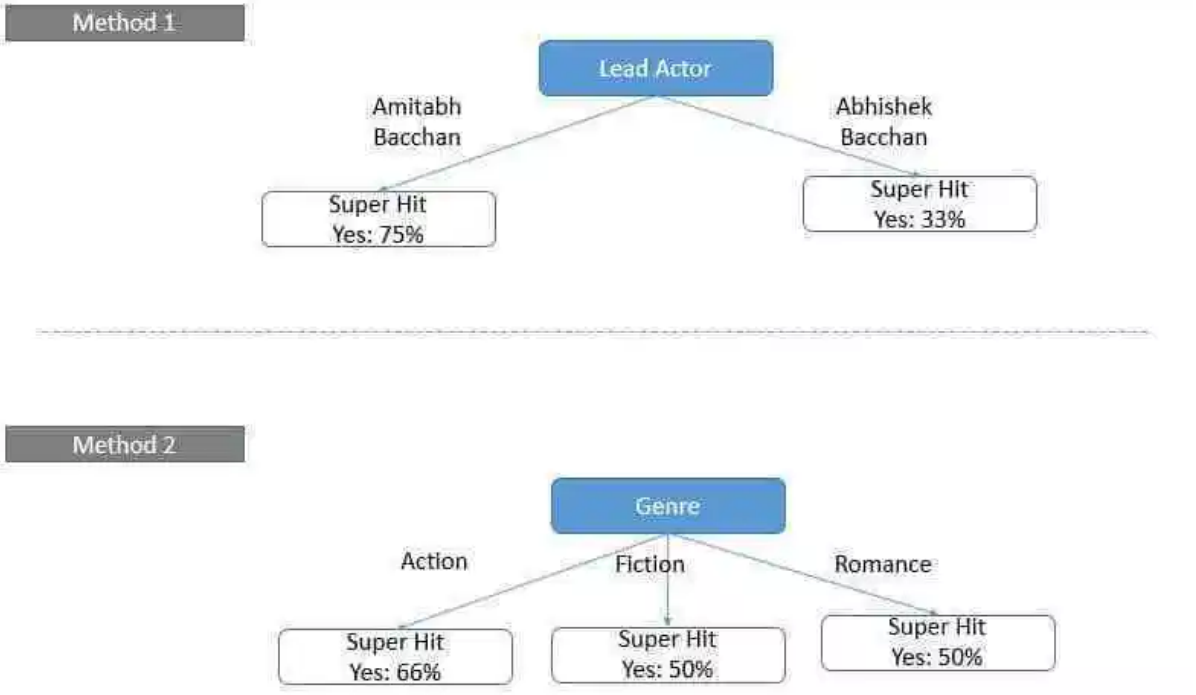
ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

Ta xét ví dụ 1:

Bạn muốn xem xét sự thành công của một bộ phim thông qua hai yếu tố: diễn viên chính của phim và thể loại phim:

Lead Actor	**Genre**	**Hit (Y/N)**
Amitabh Bacchan	Action	Yes
Amitabh Bacchan	Fiction	Yes
Amitabh Bacchan	Romance	No
Amitabh Bacchan	Action	Yes
Abhishek Bacchan	*Action*	*No*
Abhishek Bacchan	*Fiction*	*No*
Abhishek Bacchan	*Romance*	*Yes*

Giả sử, bạn muốn xác định độ thành công của bộ phim chỉ trên một yếu tố, bạn sẽ có hai cách thực hiện sau: qua diễn viên chính của phim và qua thể loại phim.



Qua sơ đồ, ta có thể thấy rõ ràng ràng, với phương pháp thứ nhất, ta phân loại được rõ ràng, trong khi phương pháp thứ hai, ta có một kết quả lộn xộn hơn. Và tương tự, cây quyết định sẽ thực hiện như trên khi thực hiện việc chọn các biến.

Có rất nhiều hệ số khác nhau mà phương pháp cây quyết định sử dụng để phân chia. Dưới đây, tôi sẽ đưa ra hai hệ số phổ biến là ***Information Gain*** và **Gain Ratio** (ngoài ra còn hệ số Gini).

2. Hàm Entropy trong cây quyết định

Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x1, x2, …, xn.

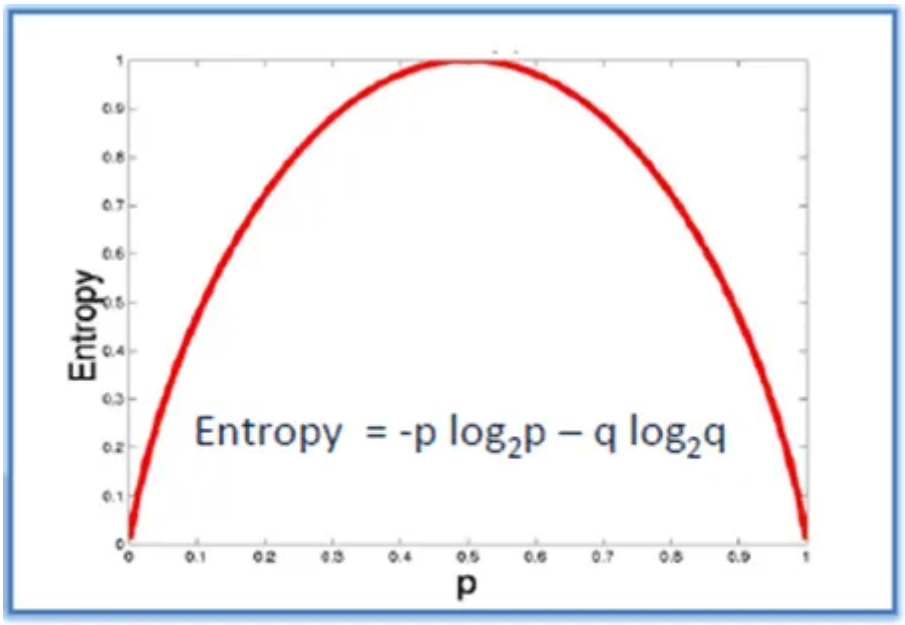
Giả sử rằng xác suất để x nhận các giá trị này là pi = p (x = xi).

Ký hiệu phân phối này là p = (p1, p2, …, pn). Entropy của phân phối này được định nghĩa là:

$$H(p) = -\sum_{n=1}^{n} p_i \log(p_i)$$

Giả sử bạn tung một đồng xu, entropy sẽ được tính như sau:

$$H = -[0.5 \ln(0.5) + 0.5 \ln(0.5)]$$



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Hình vẽ trên biểu diễn sự thay đổi của hàm entropy. Ta có thể thấy rằng, entropy đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

- P tinh khiết: pi = 0 hoặc pi = 1
- P vẩn đục: pi = 0.5, khi đó hàm Entropy đạt đỉnh cao nhất

3. Information Gain trong cây quyết định

Information Gain dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Information gain cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Information Gain tại mỗi nút theo trình tự sau:

- **Bước 1**: Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:

$$H(S) = -\sum_{c=1}^{C} (N_c/N) \log(N_c/N)$$

- **Bước 2**: Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S1, S2, …, SK với số điểm trong mỗi child node lần lượt là m1, m2 ,…, mK , ta có:

$$H(x, S) = \sum_{k=1}^{K} (m_k / N) * H(S_k)$$

- **Bước 3**: Chỉ số Gain Information được tính bằng:

$$G(x, S) = H(S) - H(x,S)$$

Với ví dụ 1, ta tính được hệ số Entropy như sau:

*EntropyParent = -(0.57*ln(0.57) + 0.43*ln(0.43))* = 0.68

Hệ số Entropy theo phương pháp chia thứ nhất:

Entropy*left = -(.75*ln(0.75) + 0.25*ln(0.25)) = 0.56*

*Entropyright = -(.33*ln(0.33) + 0.67*ln(0.67)) = 0.63*

Ta có thể tính hệ số ***Information Gain*** như sau:

*Information Gain = 0.68 – (4*0.56 + 3*0.63)/7 = 0.09*

Hệ số Entropy với phương pháp chia thứ hai như sau:

Entropy*left = -(.67*ln(0.67) + 0.33*ln(0.33)) = 0.63*

*Entropymiddle = -(.5*ln(0.5) + 0.5*ln(0.5)) = 0.69*

*Entropyright = -(.5*ln(0.5) + 0.5*ln(0.5)) = 0.69*

Hệ số ***Information Gain***:

*Information Gain = 0.68 – (3*0.63 + 2*0.69 + 2*0.69)/7 = 0.02*

So sánh kết quả, ta thấy nếu chia theo phương pháp 1 thì ta được giá trị hệ số Information Gain lớn hơn gấp 4 lần so với phương pháp 2. Như vậy, giá trị thông tin ta thu được theo phương pháp 1 cũng nhiều hơn phương pháp 2.

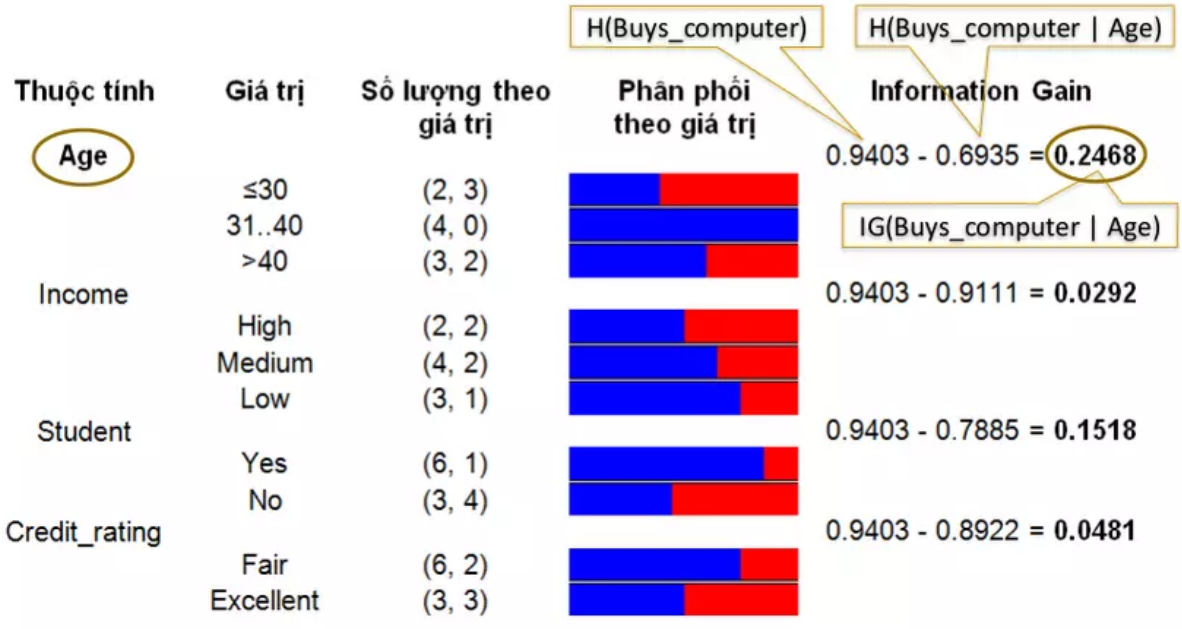
4. Ví dụ

Các thuộc tính điều kiện (dùng để phân loại)

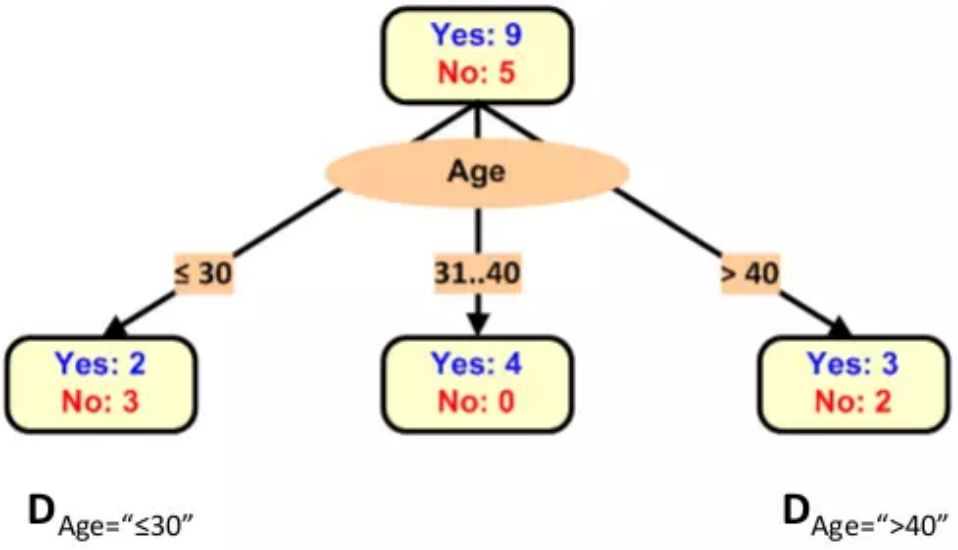
Thuộc tính cần phân loại (target attribute)

ID	Age	Income	Student	Credit_rating	Buys_computer
1	≤30	High	No	Fair	No
2	≤30	High	No	Excellent	No
3	31..40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31..40	Low	Yes	Excellent	Yes
8	≤30	Medium	No	Fair	No
9	≤30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	≤30	Medium	Yes	Excellent	Yes
12	31..40	Medium	No	Excellent	Yes
13	31..40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

Information Gain theo từng thuộc tính



Dựng cây với Node là thuộc tính Age



Tiếp tục tính Information Gain cho những thuộc tính còn lại

Thuộc tính	Giá trị	Phân phối	Information Gain
Income	High	(0, 2)	0.971 - 0.4 = 0.571
	Medium	(1, 1)	
	Low	(1, 0)	
Student	Yes	(2, 0)	0.971 - 0 = **0.971**
	No	(0, 3)	
Credit_rating	Fair	(1, 2)	0.971 - 0.951 = 0.02
	Excellent	(1, 1)	

Tiếp tục dựng cây

Thuộc tính	Giá trị	Phân phối	Information Gain
Income	High		0.971 - 0.951 = 0.02
	Medium	(2, 1)	
	Low	(1, 1)	
Student	Yes	(2, 1)	0.971 - 0.951 = 0.02
	No	(1, 1)	
Credit_rating	Fair	(3, 0)	0.971 - 0 = **0.971**
	Excellent	(0, 2)	

Kết quả

IV. Ứng dụng vào bài toán thực tế

1. Phần mềm sử dụng

Weka

2. Bài toán

- Tên bài toán: Ứng dụng thuật toán ID3 xử lý dữ liệu trò chơi Tic Tac Toe
- Mô tả bài toán: Bài toán có dữ liệu đầu vào là một file csv chứa kết quả của các ván Tic Tac Toe khác nhau. Ứng dụng thuật toán phân lớp bằng cây quyết định, ta có thể phân lớp người chiến thắng ván đấu dựa vào các nước đã đi trong ván đấu đó.
- Giải thích các cột trong file csv:
    + 9 cột đầu tiên là 9 ô trong bàn cờ Tic Tac Toe (TL, TM, TR, ML, MM, MR, BL, BM, BR), x là ô đó được người chơi x đi, o là ô đó được người chơi O đi, b là ô đó chưa có ai đi mặc dù ván cờ đã kết thúc. X luôn là người đi trước.
    + Cột class thể hiện X có thắng ván đấu không. TRUE là X thắng ván đấu còn FALSE là X thua ván đấu.

1	TL	TM	TR	ML	MM	MR	BL	BM	BR	class
2	x	x	x	x	o	o	x	o	o	TRUE
3	x	x	x	x	o	o	o	x	o	TRUE
4	x	x	x	x	o	o	o	o	x	TRUE
5	x	x	x	x	o	o	o	b	b	TRUE
6	x	x	x	x	o	o	b	o	b	TRUE
7	x	x	x	x	o	o	b	b	o	TRUE
8	x	x	x	x	o	b	o	o	b	TRUE
9	x	x	x	x	o	b	o	b	o	TRUE
10	x	x	x	x	o	b	b	o	o	TRUE

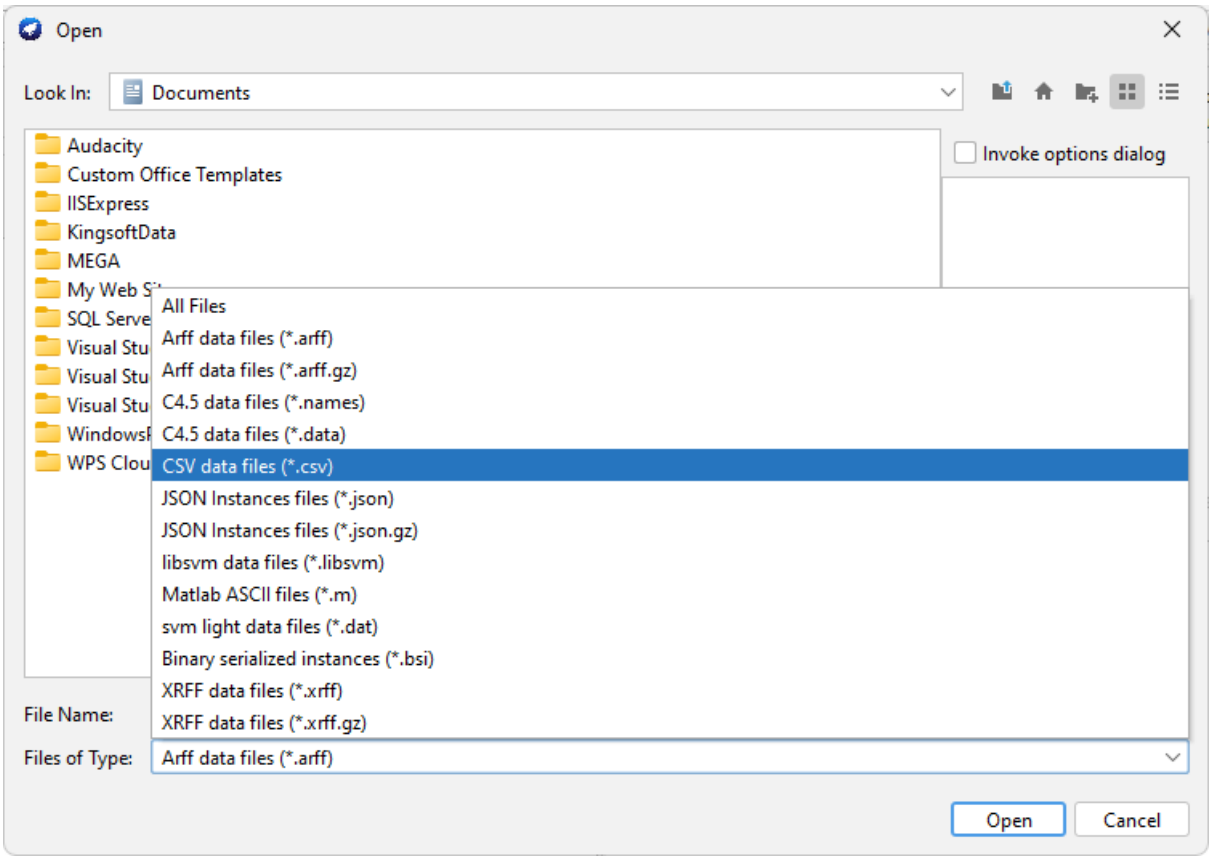
File csv chứa dữ liệu bài toán

	TL	TM	TR
	ML	MM	MR
	BL	BM	BR

Các vị trí trên bàn cờ Tic Tac Toe

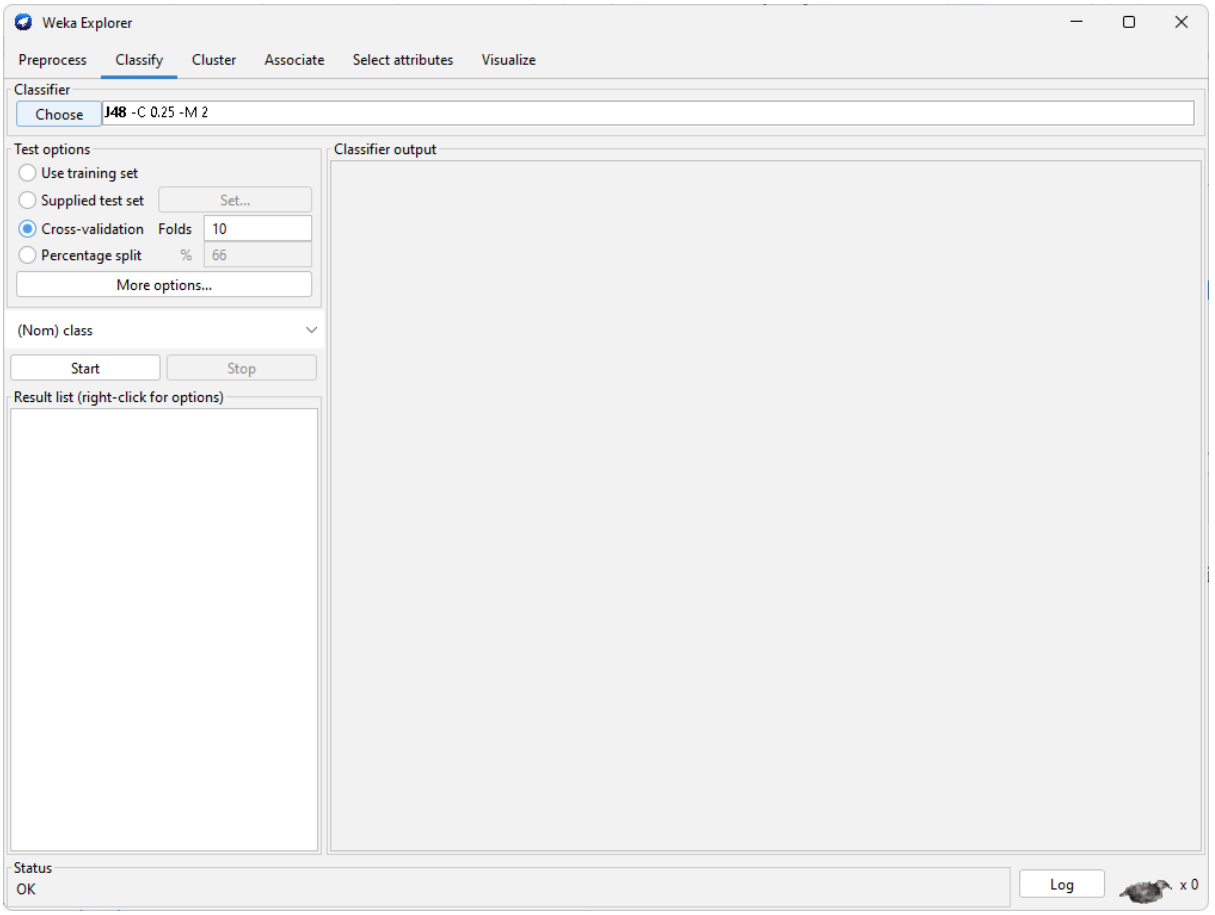
3. Thực hiện bài toán

- Mở Weka, chọn Explorer.
- Ở tab Preprocessor chọn Open file…
- Ở mục Files of Type chọn csv.



- Tìm đến file dữ liệu csv.

- Sang tab Classify chọn Choose -> weka -> classifiers -> trees -> J48



4. Kết quả

- Ấn Start, cửa sổ bên phải sẽ hiện kết quả chạy:

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: tic-tac-toe

Instances: 958

Attributes: 10

TL

TM

TR

ML

MM

MR

BL

BM

BR

class

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

MM = o

| TL = x

| | TR = x

| | | TM = x: True (38.0)

| | | TM = o

| | | | BM = o: False (12.0)

| | | | BM = x: False (7.0/2.0)

| | | | BM = b: True (5.0/1.0)

| | | TM = b

| | | | ML = x: True (4.0/1.0)

| | | | ML = o

| | | | | MR = o: False (6.0)

| | | | | MR = b: False (0.0)

| | | | | MR = x: True (2.0)

| | | | ML = b: True (1.0)

| | TR = o

| | | BL = x

| | | | ML = x: True (17.0)

| | | | ML = o: False (6.0/2.0)

| | | | ML = b: True (4.0/2.0)

| | | BL = o: False (30.0)

| | | BL = b: False (3.0)

| | TR = b

| | | BL = x

| | | | ML = x: True (14.0)

| | | | ML = o: False (7.0/1.0)

| | | | ML = b: False (3.0/1.0)

| | | BL = o: False (3.0)

| | | BL = b: False (6.0)

| TL = o

| | BR = o: False (50.0)

| | BR = x

| | | TM = x

| | | | BL = x

| | | | | BM = o: False (3.0/1.0)

| | | | | BM = x: True (5.0)

| | | | | BM = b: False (1.0)

| | | | BL = o: False (8.0/2.0)

| | | | BL = b: True (3.0/1.0)

| | | TM = o

| | | | TR = x: True (8.0/1.0)

| | | | TR = o: False (2.0)

| | | | TR = b: True (3.0/1.0)

| | | TM = b: True (15.0/1.0)

| | BR = b: False (5.0)

| TL = b

| | TM = x: False (26.0/5.0)

| | TM = o

| | | BM = o: False (12.0)

| | | BM = x: True (10.0/1.0)

| | | BM = b: True (3.0)

| | TM = b: True (18.0/7.0)

MM = b

| TL = x

| | BR = o

| | | TR = x: True (20.0/3.0)

| | | TR = o

| | | | MR = o: False (7.0)

| | | | MR = b: True (3.0)

| | | | MR = x: True (3.0/1.0)

| | | TR = b

| | | | BL = x: True (6.0)

| | | | BL = o: False (3.0)

| | | | BL = b: True (0.0)

| | BR = x: True (16.0)

| | BR = b: True (20.0)

| TL = o

| | BR = o: False (4.0)

| | BR = x

| | | TR = x: True (20.0/3.0)

| | | TR = o

| | | | TM = x: True (3.0/1.0)

| | | | TM = o: False (7.0)

| | | | TM = b: True (3.0)

| | | TR = b

| | | | BL = x: True (6.0)

| | | | BL = o: False (3.0)

| | | | BL = b: True (0.0)

| | BR = b: False (8.0)

| TL = b

| | BR = o: False (8.0)

| | BR = x: True (20.0)

| | BR = b: True (0.0)

MM = x

| TL = x

| | BR = o

| | | TR = x

| | | | BL = x: True (10.0)

| | | | BL = o

| | | | | TM = x: True (3.0)

| | | | | TM = o: False (4.0)

| | | | | TM = b: False (3.0)

| | | | BL = b: True (3.0)

| | | TR = o

| | | | MR = o: False (12.0)

| | | | MR = b: True (5.0/1.0)

| | | | MR = x

| | | | | ML = x: True (5.0)

| | | | | ML = o: False (3.0/1.0)

| | | | | ML = b: False (1.0)

| | | TR = b

| | | | BL = x: True (3.0)

| | | | BL = o

| | | | | BM = o: False (5.0)

| | | | | BM = x: True (2.0)

| | | | | BM = b: True (1.0)

| | | | BL = b: True (2.0)

| | BR = x: True (90.0)

| | BR = b: True (20.0)

| TL = o

| | TR = x

| | | BL = x: True (45.0)

| | | BL = o: False (26.0/10.0)

| | | BL = b: True (9.0)

| | TR = o

| | | TM = x: True (26.0/4.0)

| | | TM = o: False (21.0)

| | | TM = b: True (11.0/2.0)

| | TR = b

| | | ML = x: True (19.0)

| | | ML = o

| | | | BL = x: True (4.0)

| | | | BL = o: False (7.0)

| | | | BL = b: True (3.0)

| | | ML = b: True (7.0)

| TL = b

| | BR = o

| | | MR = o

| | | | TR = x: True (10.0/1.0)

| | | | TR = o: False (7.0)

| | | | TR = b: True (3.0)

| | | MR = b

| | | | BL = x: True (7.0)

| | | | BL = o

| | | | | BM = o: False (3.0)

| | | | | BM = x: True (3.0)

| | | | | BM = b: True (0.0)

| | | | BL = b: True (2.0)

| | | MR = x: True (23.0/3.0)

| | BR = x: True (20.0)

| | BR = b: True (30.0)

Number of Leaves : 95

Size of the tree : 142

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	815	85.0731 %
Incorrectly Classified Instances	143	14.9269 %

Kappa statistic 0.6669

Mean absolute error 0.1696

Root mean squared error 0.3459

Relative absolute error 37.4444 %

Root relative squared error 72.6852 %

Total Number of Instances 958

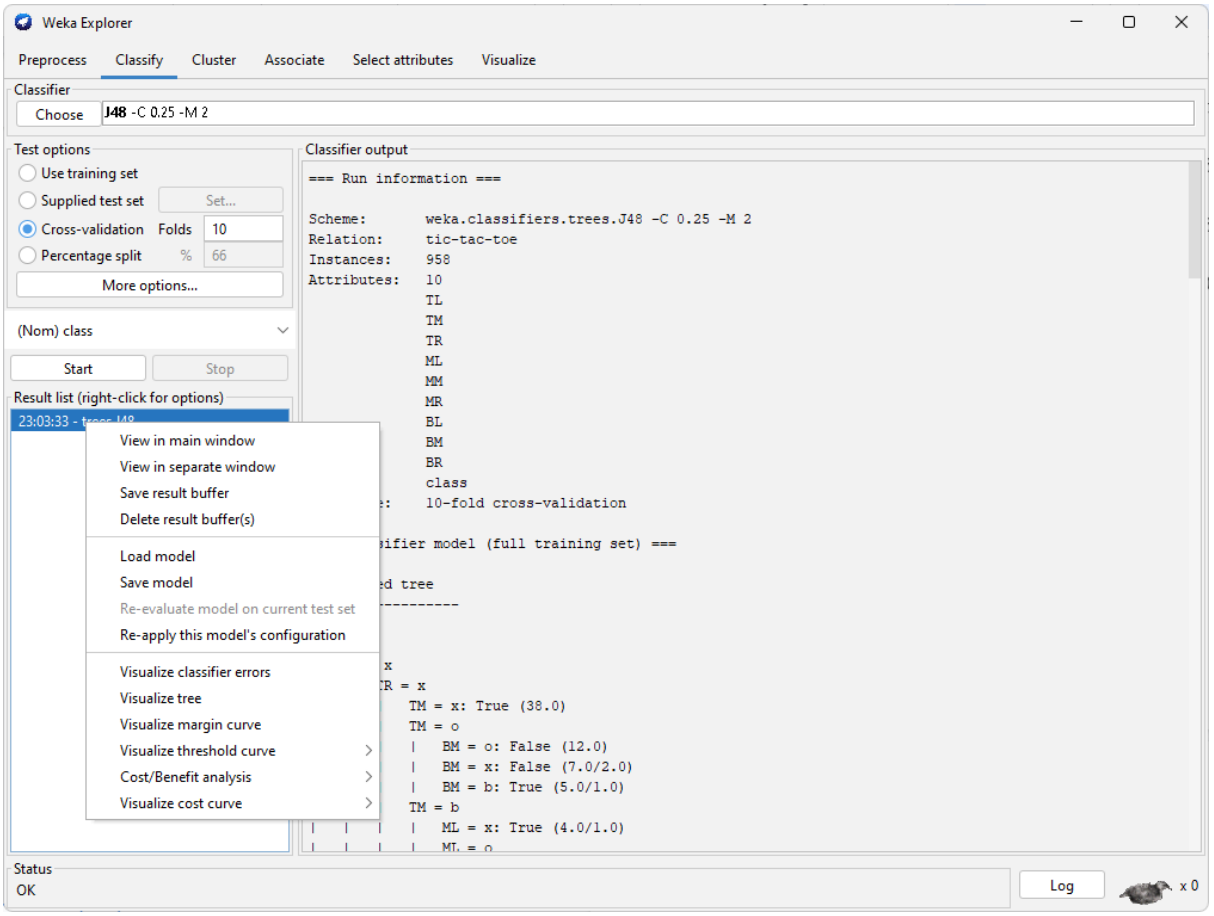
=== Detailed Accuracy By Class ===

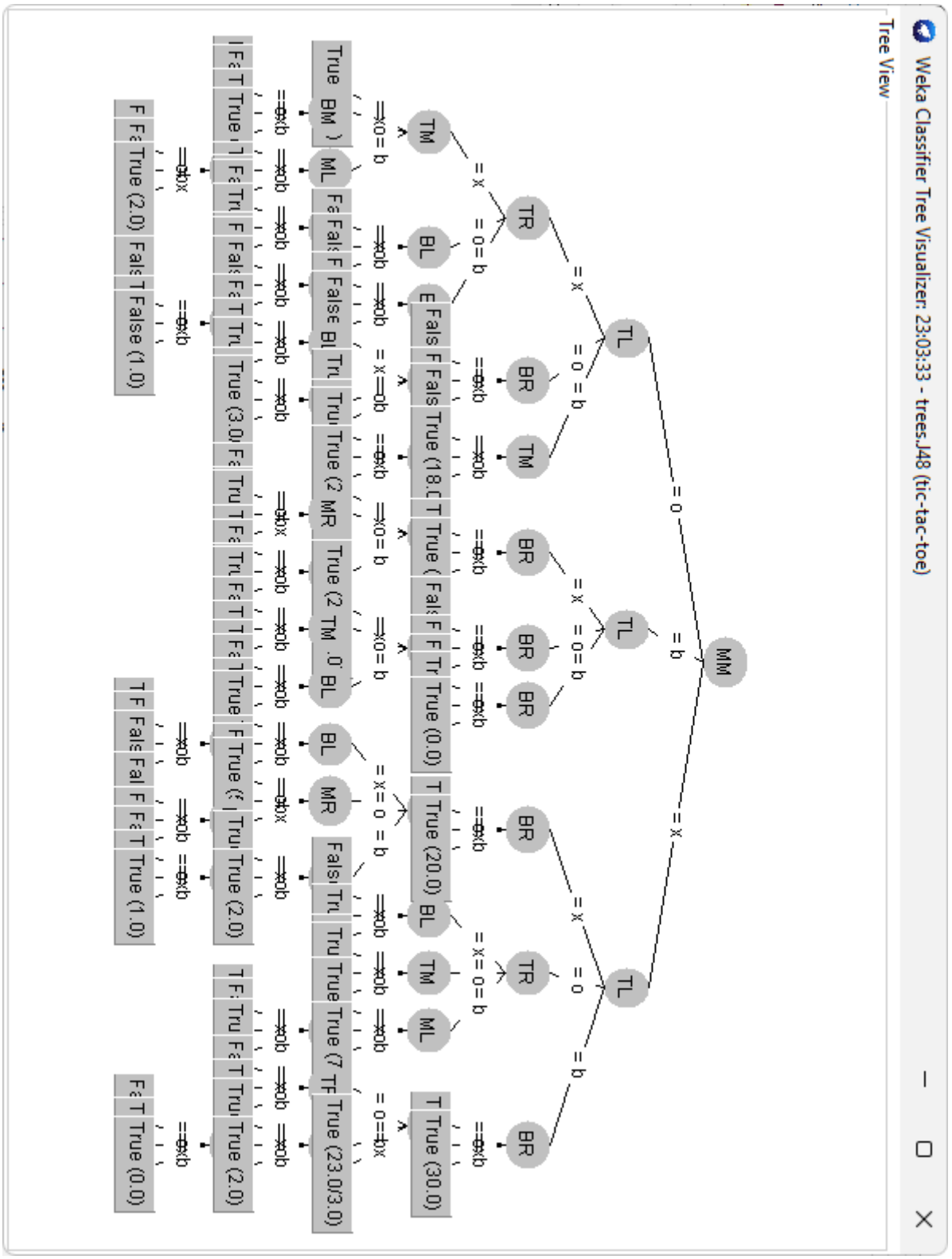
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.898	0.238	0.877	0.898	0.887	0.667	0.897	0.923	True
	0.762	0.102	0.798	0.762	0.780	0.667	0.897	0.835	False
Weighted Avg.	0.851	0.191	0.849	0.851	0.850	0.667	0.897	0.893	

=== Confusion Matrix ===

a	b	<-- classified as
562	64	a = True
79	253	b = False

- Tại phần Result List bên trái ta click chuột phải vào trees.J48 -> Visualize tree để phần mềm hiện cây quyết định.





Cây quyết định