HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tiến Hiệp

XÂY DỰNG HỆ THỐNG HỖ TRỢ RA QUYẾT ĐỊNH HÒA GIẢI, ĐỐI THOẠI TRONG CÁC TRANH CHẤP HÔN NHÂN VÀ GIA ĐÌNH

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tiến Hiệp

XÂY DỰNG HỆ THỐNG HỖ TRỢ RA QUYẾT ĐỊNH HÒA GIẢI, ĐỐI THOẠI TRONG CÁC TRANH CHẤP HÔN NHÂN VÀ GIA ĐÌNH

Chuyên ngành : Hệ thống thông tin
Mã Số : 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS. ĐỖ TRUNG TUẤN

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi cam đoan luận văn "Xây dựng hệ thống hỗ trợ ra quyết định hòa giải, đối thoại trong các tranh chấp hôn nhân và gia đình" là công trình nghiên cứu của cá nhân tôi. Được thực hiện dưới sự hướng dẫn khoa học của PGS. TS Đỗ Trung Tuấn.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin hoàn toàn chịu trách nhiệm về lời cam đoan này.

Học viên

Nguyễn Tiến Hiệp

.

LỜI CẢM ƠN

Trước tiên, tôi xin gửi lời cảm ơn đến trường Học viện Công nghệ Bưu chính Viễn thông, đã tạo điều kiện và tổ chức khóa học này để tôi có thể có điều kiện tiếp thu những kiến thức mới, có thời gian học tập và hoàn thành luận văn cao học này.

Tôi xin chân thành cảm ơn các thầy cô khoa Công nghệ thông tin và các thầy cô khác đã truyền đạt cho chúng tôi những kiến thức quý báu trong quá trình học tập.

Đặc biệt, tôi bày tỏ lòng cảm ơn sâu sắc đến thầy PGS.TS. Đỗ Trung Tuấn, thầy đã tận tụy hướng dẫn tôi hoàn thành luận văn này.

Tôi xin chân thành cảm ơn Vụ Tổng hợp Tòa án nhân dân tối cao đã tạo mọi điều kiện thuận lợi cho tôi trong suốt quá trình học tập, nghiên cứu và hoàn thành luận văn này.

Tôi chân thành cảm ơn bạn bè cùng lớp đã giúp đỡ, động viên tôi trong quá trình học tập cũng như thực hiện luận văn.

Cuối cùng, tôi xin cảm ơn tới gia đình và người thân của tôi, những người đã hết lòng tạo điều kiện và động viên tôi để tôi có được kết quả ngày hôm nay.

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

Ký hiệu	Chú giải
C4.5	Thuật toán cây quyết định
CSDL	Cơ sở dữ liệu
DSS	Decision Support System – Hệ trợ giúp quyết định[1]
EIS	Hệ thống thông tin điều hành
HNGD	Hôn nhân gia đình
ICT	Công nghệ thông tin và truyền thông
ID3	Thuật toán cây quyết định Iterative Dichotomiser 3
ISDN	Trong lĩnh vực viễn thông, ISDN (Integrated Services Digital Network-Mạng số tích hợp đa dịch vụ) là công nghệ băng hẹp được sử dụng rộng rãi, cho phép truyền dữ liệu số hóa từ một hệ thống cuối (máy chủ) gia đình qua đường điện thoại ISDN tới một công ty điện thoại.
J48	Thuật toán phân loại, cài đặt theo thuật toán ID3
LAN	Local Area Network (tiếng Anh, viết tắt LAN), "mạng máy tính cục bộ") là một hệ thống mạng dùng để kết nối các máy tính trong một phạm vi nhỏ (nhà ở, phòng làm việc, trường học, …).
OLAP	Online Analystic Processing, xử lí phân tích trực tuyến
SQL SERVER	Hệ quản trị cơ sở dữ liệu của Microsoft
TAND	Tòa án nhân dân
WAN	Wide area network (viết tắt WAN), Mạng diện rộng WAN là mạng dữ liệu được thiết kế để kết nối giữa các mạng đô thị (mạng MAN) giữa các khu vực địa lý cách xa nhau.
What-if	Bài toán tính toán ngược (nếu... thì...) trong hệ thống trợ giúp quyết định DSS
Weka	Weka là một bộ phần mềm học máy tại Đại học Waikato, New Zealand, phát triển bằng Java.

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

1. Lý do chọn đề tài

Những năm gần đây, với nền kinh tế nhiều thành phần có độ mở cao, hội nhập quốc tế ngày càng sâu rộng đã mang lại những thành tựu nổi bật về phát triển kinh tế - xã hội của đất nước, nhưng cũng làm gia tăng các tranh chấp dân sự, hành chính, hôn nhân và gia đình, các tranh chấp dân sự, khiếu kiện hành chính vẫn không ngừng tăng lên tỷ lệ thuận với quy mô tăng dân số và tăng trưởng của nền kinh tế. Tính từ năm 2012 đến nay, số lượng các vụ án loại này đã tăng gấp hai lần với tính chất ngày càng phức tạp, đa dạng; nhiều vụ án dân sự, hành chính đã xét xử sơ thẩm, phúc thẩm nhưng vẫn tiếp tục có đơn đề nghị giám đốc thẩm, tái thẩm; làm cho số lượng các vụ việc mà Tòa án phải thụ lý, giải quyết tăng nhiều so với các năm trước, tính chất các vụ việc ngày càng phức tạp; số lượng đơn đề nghị giám đốc thẩm, tái thẩm ngày càng nhiều.

Tòa án luôn trong tình trạng quá tải; nhiều vụ án dân sự, hành chính phải xét xử qua nhiều cấp trong nhiều năm; bản án, quyết định của Tòa có hiệu lực pháp luật nhưng chậm được thi hành đã ảnh hưởng đến quyền, lợi ích hợp pháp của các tổ chức, cá nhân, ảnh hưởng đến niềm tin của người dân đối với Tòa án.

Nhận thức được vai trò và xu thế phát triển tất yếu của ứng dụng Công nghệ thống tin trong hoạt động, thực hiện nhiệm vụ cải cách tư pháp theo đúng quan điểm chỉ đạo của Đảng, Nhà nước và nhằm nâng cao hiệu quả trong công tác quản lý, điều hành, trong những năm qua, việc ứng dụng Công nghệ thông tin vào các hoạt động của Tòa án nhân dân là rất cần thiết đặc biệt là nâng cao hiệu lực, hiệu quả của công tác chỉ đạo, điều hành của lãnh đạo Tòa án nhân dân các cấp và hỗ trợ nghiệp vụ xét xử. Đây là những mục tiêu hướng tới xây dựng Tòa án điện tử trong tương lai.

Cùng với sự tăng cường các hoạt động ứng dụng công nghệ thông tin phục vụ cho các hoạt động của Tòa án nhân dân và người dân trong thời gian tới,... lượng người sử dụng và dữ liệu truy cập, xử lý trên hệ thống thông tin Trung tâm dữ liệu Tòa án nhân dân sẽ tăng lên nhanh chóng; lượng dữ liệu lớn bao gồm thông tin có

cấu trúc, thông tin không có cấu trúc vẫn đang liên tục tăng trưởng và được ghi nhận hàng ngày trên hệ thống thông tin.

Các xu hướng công nghệ thông tin được xác định sẽ ảnh hưởng đến phát triển hệ thống thông tin trong thời gian tới bao gồm:

- Xu hướng bùng nổ dữ liệu (Big Data);
- Xu hướng ảo hóa (Cloud computing);
- Xu hướng tăng cường tính di động (Mobility);
- Xu hướng định danh mọi thứ trên mạng vạn vật (Internet of things).

Ứng dụng rộng rãi công nghệ thông tin & truyền thông trong mọi lĩnh vực, khai thác có hiệu quả thông tin và tri thức trong tất cả các ngành nghề xã hội trong đó có công tác nghiệp vụ của Tòa án.

Trước những yêu cầu đặt ra về cải cách tư pháp trong tình hình mới và sự bùng nổ về phát triển công nghệ thông tin đặc biệt giai đoạn hiện nay về công nghệ 4.0, Tòa án cần có những nhìn nhận đánh giá tổng thể đề án phát triển công nghệ thông tin trong thời gian tới. Ngoài cơ sở hạng tầng cần phát triển để đáp ứng nền tảng hạ tầng thì giá trị cốt lõi của ngành Tòa án là cơ sở dữ liệu về các bản án, quyết định của Tòa án cần phải được quản lý, khai thác hiệu quả.

Vấn đề đặt ra trước mắt là hiện nay mỗi năm trung bình có khoảng 500,000 các vụ việc được Tòa xét xử đây là kho dữ liệu lớn có giá trị và ngày càng tăng. Mặc dù những năm gầy đây công nghệ thông tin tại Tòa án được đẩy mạnh và có nhiều bước phát triển mạnh mẽ để phục vụ cán bộ Tòa án và người dân phục vụ cải cách tư pháp tuy nhiên tiềm năng khai thác cơ sở dữ liệu về các bản án, quyết định của Tòa án là chưa nhiều. Cụ thể chưa áp dụng được phân tích khai phá dữ liệu từ các bản án, quyết định của Tòa án mà mới chỉ là thống kê, báo cáo đơn giản phục vụ công tác báo cáo Quốc hội và báo cáo ngành.

Vì vậy việc khai phá cơ sở dữ liệu về bản án, quyết định của Tòa án từ đó hỗ trợ các Hòa giải viên, Thẩm phán, lãnh đạo Tòa án có thể xem xét đánh giá các vụ việc sau khi thụ lý và trước khi xét xử, để từ đó có những định hướng hỗ trợ hòa giải, đối thoại có thể giúp các bên giải quyết mâu thuẫn bằng chính ý chí của mình

chứ không phải phán quyết của tòa án thông qua phiên tòa xét xử; qua đó, rút ngắn thời gian giải quyết vụ việc, tiết kiệm kinh phí của Nhà nước và các bên, hàn gắn những rạn nứt trong các quan hệ xã hội, góp phần xây dựng khối đoàn kết trong nhân dân; qua việc hòa giải, đối thoại, người tiến hành hòa giải, đối thoại còn có thể giải thích, nâng cao nhận thức pháp luật cho các bên, giúp việc thi hành thuận lợi. Xuất phát từ những nhu cầu thực tế trên và đó là những lý do học viên chọn đề tài "*Xây dựng hệ thống trợ giúp ra quyết định hòa giải, đối thoại trong các tranh chấp hôn nhân và gia đình*".

2. Tổng quan về vấn đề nghiên cứu

Qua tìm hiểu và nghiên cứu học viên được biết hiện nay có Tòa án tối cao Trung Quốc đã xây dựng Hệ thống hỗ trợ xử lý án thông minh, dự đoán kết quả tố tụng, từ đó đưa ra đề xuất kiến nghị phân tích hòa giải trước khi xét xử.

Hiện tại ở Việt Nam chưa có đề tài nào nghiên cứu khai khá dữ liệu ứng dụng trong hỗ trợ công tác xét xử tại Tòa án.

Xuất phát từ thực trạng cán bộ Tòa án luôn trong tình trạng quá tải; nhiều vụ án dân sự, hành chính, hôn nhân gia đình phải xét xử qua nhiều cấp trong nhiều năm qua. Vì vậy cần nghiên cứu hệ thống hỗ trợ ra quyết định trợ giúp công tác xét xử và hỗ trợ hòa giải cho cán bộ Tòa án. Để hoàn thành đề tài nghiên cứu học viên thực hiện các định hướng nghiên cứu bao gồm:

- Nghiên cứu các hệ trợ giúp ra quyết định;
- Phân tích và thu thập thông tin dữ liệu từ các bản án, quyết định của Tòa án;
- Thiết kế cơ sở dữ liệu phục vụ hệ thống hỗ trợ ra quyết định;
- Kiểm thử;
- Báo cáo đánh giá kết quả.

3. Mục đích nghiên cứu

Nghiên cứu tìm hiểu các hệ thống trợ giúp ra quyết định từ đó lựa chọn giải pháp xây dựng hệ thống trợ giúp ra quyết định áp dụng thực tế hỗ trợ cán bộ Tòa án trong các tranh chấp về hôn nhân và gia đình.

Cụ thể phân tích các thuộc tính đặc trưng của bản án, quyết định của Tòa án

về hôn nhân và gia đình như: Tên nguyên đơn, ngày tháng năm sinh, quê quán, nghề nghiệp, quan hệ pháp luật khi thụ lý, lý do ly hôn, số con chưa thành niên, tên bị đơn và người liên quan, ngày tháng năm sinh của bị đơn, nghề nghiệp,…Từ đó xây dựng kho dữ liệu trên các thuộc tính này và áp dụng thuật toán cây quyết định hỗ trợ ra quyết định với đơn ly hôn bao nhiêu khả năng ly hôn hoặc hòa giải, với đơn tranh chấp thì khả năng bao nhiêu phần trăm thắng kiện.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Nghiên cứu thông tin dữ liệu về các bản án, quyết định có hiệu lực của Tòa án về lĩnh vực hôn nhân và gia đình.

Phạm vi nghiên cứu: Hiện này Tòa án nhân dân Việt Nam chia các loại vụ việc xét xử ra làm 6 loại chính là:

- Dân sự;
- Hình sự;
- Hôn nhân và Gia đình;
- Hành chính;
- Kinh doanh thương mại;
- Lao động.

Trong phạm vi đề tài này nghiên cứu về các bản án, quyết định của Tòa án về lĩnh vực hôn nhân và gia đình.

5. Phương pháp nghiên cứu

- Nghiên cứu các hệ hỗ trợ ra quyết định, các kỹ thuật, thuật toán cây quyết định như ID3, C4.5 hỗ trợ ra quyết định để lựa chọn mô hình trợ giúp ra quyết định phù hợp;
- Phân tích dữ liệu các bản án, quyết định về hôn nhân gia đình, thiết kế cơ sở dữ liệu áp dụng cây quyết định xây dựng hê thống trợ giúp ra quyết định.
- Thiết kế giao diện truy xuất và xử lý dữ liệu để cung cấp thông tin cần thiết cho việc ra quyết định
- Đánh giá kết quả sử dụng cây quyết định

6. Cấu trúc của luận văn

Luận văn chia thành các chương.

- Chương 1 đề cập hệ thống trợ giúp quyết định, nhu cầu khai phá dữ liệu ra quyết định và các thuật toán cây phân loại ID3 và C4.5;
- Chương 2 thể hiện việc thực hiện phân loại nhờ cây quyết định, sử dụng thuật toán C4.5. Luận văn sử dụng cài đặt J48 trong phần mềm Weka;
- Chương 3 đề cập cơ sở dữ liệu về các án hôn nhân và sử dụng môi trường Visual C# để trợ giúp ra quyết định giải quyết vụ, việc hôn nhân gia đình. Hệ quản trị cơ sở dữ liệu là SQL SERVER.

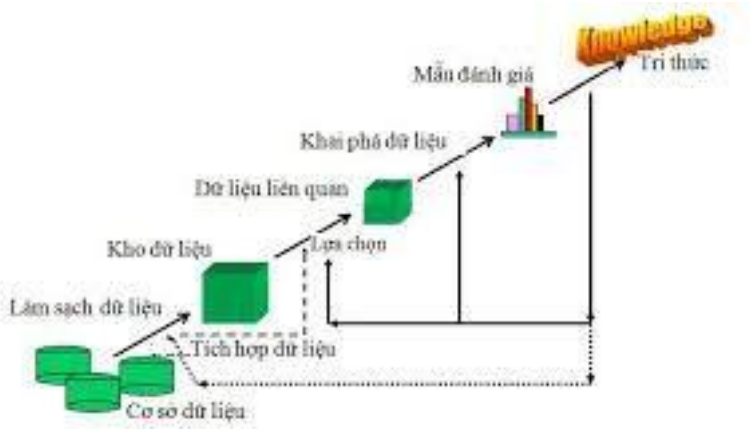
Cuối luận văn là phần kết luận, tự đánh giá về các kết quả đã đạt được và phương hướng nghiên cứu tiếp theo.

CHƯƠNG 1.
KHAI PHÁ DỮ LIỆU VÀ CÁC HỆ THỐNG RA QUYẾT ĐỊNH

1.1.Tổng quan về khai phá dữ liệu

1.1.1. Động cơ của việc khai phá dữ liệu

Trong một vài thập kỉ trở lại đây, khả năng tạo sinh và lưu trữ dữ liệu của con người đã tăng lên cực kì nhanh chóng. Lượng dữ liệu khổng lồ được lưu trữ đã dẫn đến việc đòi hỏi cấp bách những kĩ thuật mới, những công cụ tự động thông minh trợ giúp cho con người trong việc chuyển đổi một lượng lớn dữ liệu thành những thông tin hữu ích và tri thức.



Hình 1.1. Khai phá dữ liệu

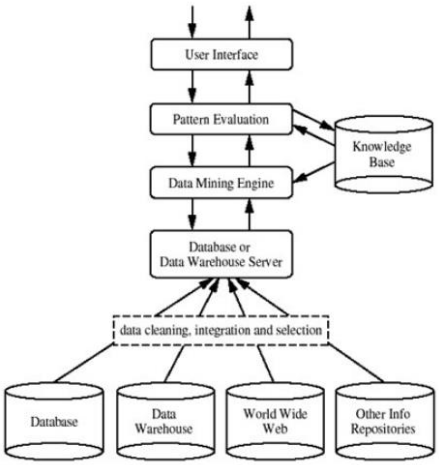
(*Nguồn: https://viblo.asia*)

Khai phá dữ liệu là công việc trích rút tri thức một cách tự động và hiệu quả từ một khối lượng dữ liệu rất lớn. Tri thức đó thường ở dạng các mẫu có tính chất không tầm thường, không tường minh, chưa được biết đến và có tiềm năng mang lại lợi ích. Có một số nhà nghiên cứu còn gọi khai phá dữ liệu là phát hiện tri thức trong cơ sở dữ liệu. Ở đây chúng ta sẽ xem khai phá dữ liệu là cốt lõi của quá trình phát hiện tri thức. Quá trình phát hiện tri thức bao gồm các bước:

1. Làm sạch dữ liệu: ở bước này các nhiễu và dữ liệu không nhất quán sẽ được loại bỏ.
2. Tích hợp dữ liệu: dữ liệu từ nhiều nguồn khác nhau có thể được tổ hợp lại.
3. Lựa chọn dữ liệu: những dữ liệu thích hợp với nhiệm vụ phân tích sẽ được

trích rút ra từ CSDL.

4. Chuyển đổi dữ liệu: dữ liệu sau khi được chọn lọc sẽ được chuyển đổi hay hợp nhất về dạng thích hợp cho việc khai phá.
5. Khai phá dữ liệu: đây là quá trình cốt lõi, tất yếu trong đó các phương pháp thông minh sẽ được áp dụng nhằm trích rút ra các mẫu dữ liệu.
6. Đánh giá mẫu: các nhà phân tích dữ liệu sẽ dựa trên một số độ đo nào đó để xác định lợi ích thực sự, độ quan trọng của các mẫu biểu diễn tri thức.
7. Biểu diễn tri thức: ở giai đoạn này các kĩ thuật biểu diễn và hiển thị tri thức sẽ được sử dụng để đưa tri thức đã lấy ra đến người dùng.



Hình 1.2. Kiến trúc khai phá dữ liệu

(Nguồn: https://viblo.asia)

Việc khai phá dữ liệu có thể được tiến hành trên một lượng lớn dữ liệu có trong các CSDL, các kho dữ liệu hoặc trong các loại lưu trữ thông tin khác.

1.1.2. Kiến trúc của hệ thống khai phá dữ liệu

Kiến trúc của một hệ thống khai phá dữ liệu điển hình như hình trên trong đó:

1. *CSDL, kho dữ liệu hoặc các thông tin lưu trữ khác*: đây là một hay một tập các CSDL, các kho dữ liệu, các trang tính hay các dạng khác của thông tin được lưu trữ. Các kĩ thuật làm sạch hoặc tích hợp dữ liệu có thể được thực hiện.

2. *Máy chủ CSDL hay máy chủ kho dữ liệu*: máy chủ này có nhiệm vụ lấy được những dữ liệu thích hợp dựa trên nhưng yêu cầu khai phá của người dùng.

3. *Cơ sở tri thức*: đây là miền tri thức được dùng để tìm kiếm hay đánh giá độ quan

trọng của các mầu kết quả. Tri thức này có thể bao gồm một sự phân cấp khái niệm dùng để tổ chức các thuộc tính hay các giá trị thuộc tính ở các mức trừu tượng khác nhau.

4. *Máy khai phá dữ liệu*: một hệ thống khai phá dữ liệu cần phải có một tập các module chức năng để có thể thực hiện được công việc, chẳng hạn như đặc trưng hóa, kết hợp, phân lớp, phân cụm, phân tích sự tiến hóa hoặc sự chệch hướng.

5. *Module đánh giá mẫu*: bộ phận này tương tác với các module khai phá dữ liệu để tập trung vào việc duyệt tìm các mẫu đáng tin cậy. Nó có thể dùng các ngưỡng về độ quan tâm để lọc các mẫu đã khám phá được.

6. *Giao diện người dùng*: bộ phận này cho phép người dùng giao tiếp với hệ thống khai phá dữ liệu. Thông qua giao diện này người dùng tương tác với hệ thống bằng cách đặc tả một yêu cầu khai phá hay một nhiệm vụ, cung cấp thông tin giúp cho việc tìm kiếm và thực hiện khai phá đánh giá trên các kết quả khai phá trung gian. Ngoài ra bộ phận này còn cho phép người dùng có thể xem được các lược đồ CSDL, lược đồ kho dữ liệu hay các cấu trúc dữ liệu, các đánh giá mẫu và hiển thị chúng trong các khuôn dạng mẫu khác nhau.

1.1.3 Các chức năng của khai phá dữ liệu

Nhìn chung các nhiệm vụ của một hệ khai phá dữ liệu có thể được phân chia thành hai loại: mô tả và dự đoán.

1. Công việc khai phá dữ liệu loại mô tả nhằm biểu thị các đặc điểm chung của dữ liệu có trong CSDL.

2. Công việc khai phá dữ liệu loại dự đoán nhằm thực hiện suy luận trên dữ liệu hiện tại để có thể đưa ra dự đoán.

1.1.4. Các phương pháp khai phá dữ liệu

Có nhiều phương pháp thực hiện việc khai phá dữ liệu theo [1] có các loại công cụ chính sau:

1. *Các phương pháp thống kê*: Các phương pháp gồm (i) hồi qui tuyến tính và phi tuyến; (ii) đánh giá điểm; (iii) phân bố xác xuất, định lý Bayes (iv) tương quan; (v) phân tích cụm;

2. *Cây quyết định*: Các cây quyết định được dùng trong các phương pháp phân lớp và phân cụm. Cây quyết định tách bài toán thành những tập con cụ thể dần dần, nhờ đi từ tổng quát hóa đến đặc biệt hóa trên thông tin. Cây quyết định được xác định theo nút gốc và các nút trong. Mỗi nút gắn với một câu hỏi. Các cung nối các nút bao trùm tất cả những khả năng hỏi dữ liệu. Mỗi câu trả lời biểu diễn một đầu ra có thể xảy ra;

3. *Lập luận theo trường hợp*: Sử dụng các trường hợp quá khứ, tiếp cận lập luận theo trường hợp cho phép ghi nhận các mẫu. Chẳng hạn các khách hàng của công ty Cognitive Systems dùng tiếp cận này để trợ giúp các ứng dụng văn phòng. Một khách hàng có thư viện với 50.000 câu hỏi theo trường hợp. Các trường hợp mới có thể khớp nhanh với 50.000 mẫu trong thư viện, để trả lời câu hỏi với chính xác 90%;

4. *Tính toán nơ ron*: Các mạng nơ ron dùng nhiều nút nối nhau, tương tự như khớp nối trong hệ thống nơ ron của con người. Tiếp cận này kiểm tra khối lượng lớn các dữ liệu lịch sử, để phát hiện các mẫu. Do vậy, người ta có thể duyệt cơ sở dữ liệu lớn, và phát hiện sự kiện mới, chẳng hạn các khách hàng tiềm năng đối với mặt hàng mới. Nhiều ứng dụng thuộc lĩnh vực tài chính và sản xuất;

5. *Các tác nhân thông minh*: Một trong những tiếp cận hứa hẹn nhất để tìm kiếm thống tin từ cơ sở dữ liệu, đặc biệt từ cơ sở dữ liệu ngoài, là dùng các tác nhân thông minh. Trước khả năng lớn nhanh của thông tin trên Internet, việc phát hiện đúng thông tin trở nên khó hơn. Các ứng dụng khai phá dữ liệu trên Web là các tác nhân phần mềm thông minh điển hình;

6. *Các thuật toán di truyền*: Các thuật toán di truyền làm việc trên nguyên tắc

mở rộng đầu ra. Khi cho số cố định các đầu ra, thuật toán di truyền tìm để xác định các giải pháp tốt nhất. Các thuật toán di truyền được dùng để phân cụm và phát hiện luật kết hợp;

7. *Các công cụ khác*: Người ta cũng dùng vài công cụ khai phá dữ liệu khác (i) suy diễn trên luật; (ii) hiển thị dữ liệu. Nhà cung cấp Web cũng cho phép phát triển các công cụ mới.

1.1.5. Đặc trưng hóa và phân biệt

Đặc trưng hóa là việc tổng kết các đặc điểm hay tính chất chung của một lớp dữ liệu đích. Dữ liệu đó tương đương với một lớp do người dùng đặc tả bằng một truy vấn CSDL.

Có một số phương pháp để tổng kết và biểu thị đặc trưng dữ liệu một cách hiệu quả. Chẳng hạn như thao tác ROLL-UP của hệ phân tích trực tuyến OLAP, dữ liệu dạng khối có thể được dùng để thực hiện tổng kết theo một chiều cụ thể dưới sự điều khiển của người dùng.

Dữ liệu trả về của quá trình đặc trưng hóa có thể được biểu diễn ở những khuôn dạng khác nhau. Ví dụ nó có thể là biểu đồ hình tròn, biểu đồ hình cột, khối dữ liệu đa chiều hay các bảng đa chiều bao gồm cả các bảng tham khảo chéo. Kết quả của quá trình khai phá mô tả cũng có thể được biểu diễn như các quan hệ tổng quát hay các luật.

1.1.6. Phân tích sự kết hợp

Phân tích sự kết hợp là việc khám phá ra các luật kết hợp trong một tập lớn dữ liệu. Các luật kết hợp thể hiện mối quan hệ giữa các giá trị thuộc tính mà ta nhận thấy được tự tần suất xuất hiện cùng với nhau. Các luật kết hợp được khám phá từ một tập lớn các bản ghi và những tập luật có ý nghĩa có thể giúp cho các nhà doanh nghiệp ra quyết định.

1.1.7. Phân lớp và dự đoán

Phân lớp là quá trình tìm một tập các mô hình (hoặc các chức năng) mô tả và phân biệt các lớp dữ liệu. Các mô hình này sẽ được sử dụng cho mục đích dự đoán về lớp của một đối tượng. Việc xây dựng mô hình dựa trên sự phân tích một tập các

dữ liệu huấn luyện. Một mô hình như vậy có thể được biểu diễn trong nhiều dạng, chẳng hạn các dạng luật phân lớp IF-THEN, cây quyết định, công thức toán hay mạng nơ-ron. Tuy sự phân lớp được sử dụng để dự đoán nhãn lớp cho các đối tượng dữ liệu, trong nhiều ứng dụng người dùng cũng có thể mong muốn dự đoán những giá trị dữ liệu khuyết thiếu nào đó. Thông thường đó là việc dự đoán các giá trị thuộc kiểu dữ liệu số. Sự dự đoán cũng bao gồm việc xác định khuynh hướng phân loại dựa trên những dữ liệu hiện có.

Để phân lớp và dự đoán, có thể cần trước một sự phân tích thích hợp. Sự phân tích đó nhằm xác định các thuộc tính không tham gia vào quá trình phân lớp và dự đoán, chúng sẽ bị loại trừu sau bước này.

1.1.8. Phân cụm

Không giống như phân lớp và dự đoán, sự phân cụm sẽ phân tích các đối tượng dữ liệu khi chưa biết nhãn của lớp, nghĩa là nhãn của lớp không tồn tại trong quá trình huấn luyện dữ liệu. Phân cụm có thể được sử dụng để đưa ra những nhãn của lớp.

Sự phân cụm có mục đích nhóm các đối tượng lại theo nguyên tắc:

1. Các đối tượng trong cùng một nhóm giống nhau ở mức cao nhất
2. Các đối tượng khác nhóm có mức giống nhau ít nhất

Điều này có nghĩa là các cụm sẽ được tạo ra sao cho các đối tượng trong mỗi cụm có độ tương tự cao khi so sánh với nhau và rất khác nhau khi so sanh với các đối tượng thuộc cụm khác. Mỗi cụm được tạo thành có thể được xem như một lớp đối tượng. Và các luật sẽ được trích rút ra từ đó. Việc phân cụm cũng đem lại một phương pháp để định dạng và phân loại.

1.1.9. Phân tích phần tử ngoài cuộc

Một cơ sở dữ liệu có thể chứa các đối tượng dữ liệu không tuân theo mô hình dữ liệu. Những đối tượng như vậy được gọi là phần tử ngoài cuộc. Hầu hết các phương pháp khai phá dữ liệu đều coi các phần tử ngoài cuộc là nhiễu và loại bỏ chúng.

Tuy nhiên thì trong một số ứng dụng nào đó các sự việc hiếm khi xảy ra lại

được quan tâm hơn là những gì thường xuyên gặp phải. Sự phân tích các phần tử ngoài cuộc được xem như là sự khai phá các phần tử ngoài cuộc. Có một số phương pháp thường được sử dụng để phát hiện các phần tử ngoài cuộc:

1. Dùng kiểm thử mang tính thống kê trên cơ sở một giả thuyết về phân phối dữ liệu hay một mô hình xác suất cho dữ liệu;
2. Dùng các độ đo khoảng cách, theo đó các đối tượng có một khoảng cách đáng kể đến cụm bất kì khác được xem là phần tử ngoài cuộc;
3. Dùng các phương pháp dựa trên độ lệch để kiểm tra sự khác nhau trong những đặc trưng chính của đối tượng trong một nhóm.

Phương pháp phân tích phần tử ngoài cuộc có thể khám phá ra những người sử dụng thẻ tín dụng ngân hàng một cách gian lận bằng việc phát hiện những việc mua sắm với một lượng tiền quá lớn trong tài khoản khi so sánh với những khoản chi phí thông thường được chi trả bằng chính tài khoản này. Những giá trị ngoài cuộc này cũng có thể được phát hiện với sự chú ý về địa điểm và loại mua sắm hoặc tần suất mua sắm.

1.2. Khái niệm về hệ thống hỗ trợ ra quyết định

1.2.1. Quyết định

1.2.1.1. Khái niệm về quyết định

Theo Simon (1960); Costello & Zalkind (1963); Churchman (1968) đó là một lựa chọn về "đường lối hành động", hay "chiến lược hành động" Fishburn (1964) dẫn đến một mục tiêu mong muốn" Churchman (1968).

"Một quá trình lựa chọn có ý thức giữa hai hay nhiều phương án để chọn ra một phương án tạo ra được kết quả mong muốn trong các điều kiện ràng buộc đã biết".

1.2.1.2. Hiểu rõ thêm về ra quyết định

Việc đưa ra quyết định đối với một vấn đề xuất hiện trong khắp các lĩnh vực, hoạt động của đời sống mà đôi khi chúng ta không nhận ra. Từ những việc đơn giản như chọn một bộ quần áo để đi dự tiệc cho đến các công việc lớn lao như phân bố ngân sách vào các chương trình của quốc gia đều là các công việc đưa ra quyết định.

Vậy đưa ra quyết định chính là chọn ra trong các giải pháp khả thi một giải pháp mà theo người đưa ra quyết định là phù hợp nhất.

1.2.2. Quá trình ra quyết định

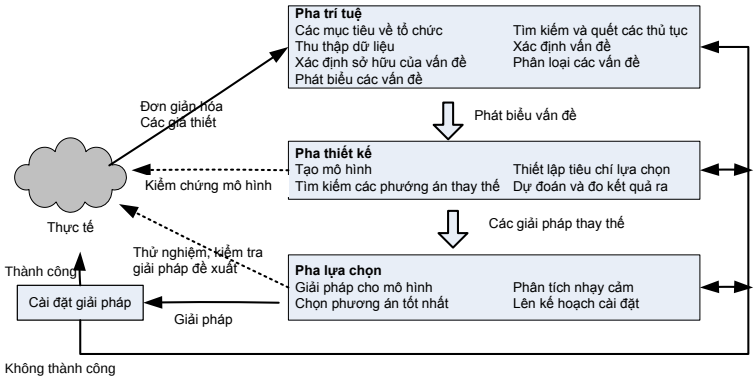
1.2.2.1. Phân loại quyết định

Có thể phân ra bốn loại quyết định như sau:

- *Quyết định có cấu trúc (Structured Decision)*: Các quyết định mà người ra quyết định biết chắc chắn đúng. Ví dụ: Bài toán quyết định thưởng, phạt Nhân viên.
- *Quyết định không có cấu trúc (NonStructured Decision)*: Các quyết định mà người ra quyết định biết là có nhiều câu trả lời gần đúng và không có cách nào để tìm ra câu trả lời chính xác nhất. Ví dụ: Bài toán quyết định chiến lược phát triển của Nhà trường.
- *Quyết định đệ quy (Recurring Decision)*: Các quyết định lặp đi lặp lại.
- *Quyết định không đệ quy (Nonrecurring Decision)*: Các quyết định không xảy ra thường xuyên.

1.2.2.2. Các giai đoạn của quá trình ra quyết định

Theo Simon, quá trình ra quyết định và quan hệ giữa chúng được giới thiệu ở hình dưới đây:



Hình 1.3. Các giai đoạn của quá trình ra quyết định

(Nguồn: "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016)

- Giai đoạn thứ nhất là nhận định (Intelligence): Tìm kiếm các tình huống dẫn đến việc phải ra quyết định, nhận dạng các vấn đề, nhu cầu, cơ hội, rủi ro.
- Giai đoạn thứ hai là thiết kế (Design): Phân tích các hướng tiếp cận để giải quyết vấn đề, đáp ứng các nhu cầu, tận dụng các cơ hội, hạn chế các rủi ro.

- Giai đoạn thứ ba là lựa chọn (Choice): Cân nhắc và đánh giá từng giải pháp và chọn giải pháp tối ưu.
- Cuối cùng là tiến hành ra quyết định (Implementation): Thực hiện giải pháp được chọn, theo dõi kết quả và điều chỉnh khi thấy cần thiết.

1.2.2.3. Tìm kiếm và đánh giá các lựa chọn một phần rất quan trọng trong hỗ trợ ra quyết định.

Giai đoạn lựa chọn (Choice Phase) là giai đoạn quan trọng nhất của quá trình ra quyết định. Giai đoạn này bao gồm ba bước chính sau đây:

- Tìm kiếm lựa chọn;
- Đánh giá lựa chọn;
- Giới thiệu lựa chọn.

Trong trường hợp này người ra quyết định muốn sử dụng mô hình quy chuẩn để tìm kiếm một lựa chọn tối ưu, thì Hệ hỗ trợ có thể sử dụng phương pháp vét cạn để duyệt hết các lựa chọn hay mô hình toán học để phân tích.

Đối với mô hình mô tả, ta có thể sử dụng phương pháp kinh nghiệm (Heuristic search) để duyệt các lựa chọn dựa trên các quy luật rút ra từ thử và sai hay kinh nghiệm.

Phương pháp đáng giá các lựa chọn được quy định khác nhau trong bài toán một mục tiêu và bài toán đa mục tiêu. Bài toán một mục tiêu có thể được mô hình hóa bằng bảng ra quyết định hay cây quyết định.

Một trong các phương pháp hiệu quả để giải quyết đa mục tiêu là đo lường trọng số của các ưu tiên ra quyết định (Analytical Hierarechy press of Expert choice). Một phương pháp khác là tối ưu hóa dựa trên các mô hình toán học tuyến tính (Microsoft Excel, Lingo…). Một phương pháp khác là lập trình kinh nghiệm sử dụng Heuristics như là tabu search, giải thuật di truyền.

1.2.3. Khái niệm hệ hỗ trợ quyết định

Trong thập niên 1970, Scott Norton đưa những khái niệm đầu tiên về hệ trợ giúp quyết định (Decision Support System, DSS). Ông định nghĩa "DSS là các hệ dựa trên máy tính, có tính tương tác, giúp các nhà ra quyết định dùng dữ liệu và mô

hình để giải các bài toán phi cấu trúc, những bài toán mờ, phức tạp với lời giải không hoàn chỉnh".

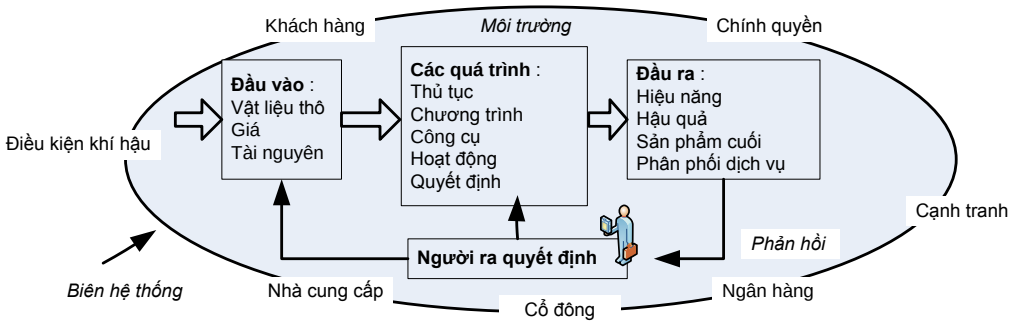
Cho đến nay chưa có một định nghĩa thống nhất về DSS. Tuy nhiên tất cả đều đồng ý mục đích cơ bản nhất của DSS là hỗ trợ và cải tiến việc ra quyết định.

1.3. Các thành phần của hệ thống ra quyết định

1.3.1. Các thành phần

Một hệ hỗ trợ quyết định gồm có ba thành phần chính:

- Quản lý dữ liệu;
- Quản lý mô hình;
- Quản lý giao diện người dùng.



Hình 1.4. Hệ thống ra quyết định và môi trường của nó

(Nguồn: "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016)

Quản lý dữ liệu (Data Management): Thực hiện công việc lưu trữ các thông tin của hệ và phục vụ cho việc lưu trữ, cập nhật, truy vấn thông tin.

Quản lý mô hình (Model Management) hay còn gọi là hệ quản trị cơ sở mô hình (MBMS, model base management system): bao gồm các mô hình ra quyết định (DSS models) và việc quản lý các mô hình này. Một số ví dụ của các mô hình này bao gồm: mô hình nếu thì, mô hình tối ưu, mô hình tìm kiếm mục đích, mô hình thống kê.

Quản lý giao diện người dùng giúp người sử dụng giao tiếp với và ra lệnh cho hệ thống. Các thành phần vừa kể trên tạo nên HHTQĐ, có thể kết nối với intranet/extranet của tổ chức hay kết nối trực tiếp với Internet.

1.3.2. Mô hình ra quyết định

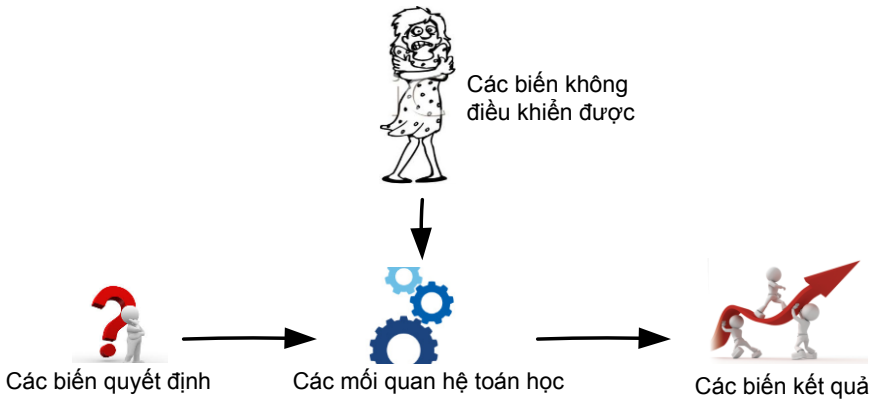
Một đặc trưng cơ bản của hệ hỗ trợ ra quyết định là phải có ít nhất một mô

hình hỗ trợ ra quyết định. Việc chọn lựa và xây dựng mô hình nằm trong giai đoạn thứ 2 (Design Phase) của quá trình ra quyết định.

Mô hình là một khái quát hóa hay trừu tượng hóa các vấn đề thực tế thành các mô hình định tính hay định lượng. Đó là một quy trình kết hợp cả khoa học (sự chính xác, logic) và nghệ thuật (sự sáng tạo).

Một mô hình gồm ba thành phần cơ bản:

- Decision Variables: Đây là các lựa chọn xác định bởi người ra quyết định. Chẳng hạn trong bài toán quyết định thưởng phạt nhân viên …



Hình 1.5. Cấu trúc chung của mô hình định lượng

(Nguồn: "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016)

- Uncontrollable Variables: Đây là các biến không nằm trong sự kiểm soát của người ra quyết định (bị tác động bởi các yếu tố bên ngoài). Chẳng hạn trong bài toán trên thì đây là…
- Result Variables: Đây là biến kết quả của mô hình. Chẳng hạn trong bài toán trên đây…

Khi lựa chọn quyết định cuối cùng, người ra quyết định có thể muốn có một quyết định tối ưu hay một quyết định thỏa đáng, phần tối ưu. Do vậy có thể chia ra hai loại mô hình hỗ trợ ra quyết định.

- Mô hình quy chuẩn (Normative Model): Mô hình này xem xét tất cả các phương án và chọn ra phương án tối ưu.
- Mô hình mô tả (Desscriptive Model): Mô hình xem xét một tập hợp các điều kiện theo ý người dùng và xem xét các phương án theo các điều kiện này và

đưa ra một kết quả thỏa đáng. Vì mô hình này không xem xét hết tất cả các phương án nên kết quả cuối cùng chỉ gần tối ưu.

Mô hình quy chuẩn thường được sử dụng trong bài toán tối ưu hóa một mục tiêu. Mô hình mô tả thường được sử dụng trong bài toán tối ưu hóa đa mục tiêu khi các mục tiêu này có thể mâu thuẫn nhau.

1.4. Phân loại các hệ thống ra quyết định

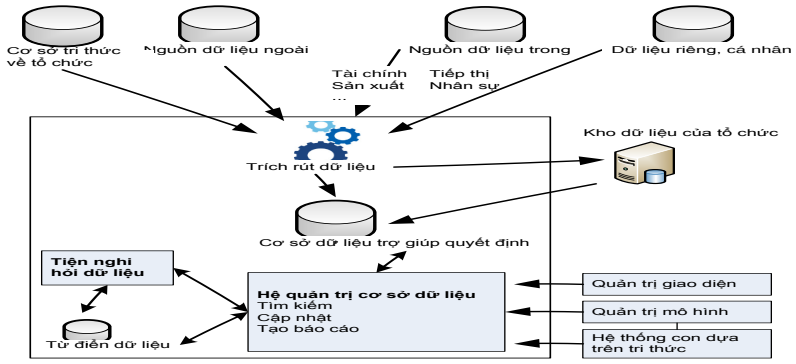
1.4.1. Các hệ thống ra quyết định

Hệ hỗ trợ ra quyết định được phân loại dựa trên nhiều tiêu chí. Hiện nay, vẫn chưa có cách phân loại thống nhất. Sau đây là hai cách phổ biến nhất.

Theo [4] có tất cả năm loại hệ hỗ trợ ra quyết định:

1. Hướng giao tiếp (Communication, Driven DSS);
2. Hướng dữ liệu (Data, Driven DSS);
3. Hướng tài liệu (Document, Driven DSS);
4. Hướng tri thức (Knowledge, Driven DSS);
5. Hướng mô hình (Model, Driven DSS).

Hướng giao tiếp, Hệ hỗ trợ ra quyết định sử dụng mạng và công nghệ viễn thông để liên lạc và cộng tác. Công nghệ viễn thông bao gồm mạng cục bộ (LAN), mạng diện rộng (WAN), Internet, ISDN, mạng riêng ảo…. là then chốt trong việc hỗ trợ ra quyết định. Các ứng dụng của Hệ hỗ trợ ra quyết định hướng giao tiếp là Phần mềm nhóm (Group ware), hội thảo từ xa (Videoconferencing), bản tin (Bulletin Boards) …



Hình 1.6. Mô hình khái niệm của DSS

(Nguồn: "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016)

Hướng dữ liệu, Hệ hỗ trợ ra quyết định dựa trên truy xuất và xử lý dữ liệu. Phiên bản đầu tiên được gọi là Hệ chỉ dành cho việc truy xuất dữ liệu (Retrieval, Only DSS). Kho dữ liệu (Data warehoure) là cơ sở dữ liệu tập trung chứa thông tin từ nhiều nguồn đồng thời sẵn sàng cung cấp thông tin cần thiết cho việc ra quyết định. OLAP có nhiều tính năng cao cấp. Ví dụ dữ liệu vật tư cần phải phân cấp theo nhiều chiều như theo trình độ học vấn, số năm công tác, theo người làm việc…

Hướng tài liệu, Hệ hỗ trợ ra quyết định dựa trên việc truy xuất và phân tích các văn bản, tài liệu…. Trong một đơn vị, có rất nhiều văn bản như các công văn đi, đến, nội bộ, giấy tờ… Internet cho phép truy xuất các kho tài liệu lớn như kho văn bản, hình ảnh, âm thanh.. Một công cụ tìm kiếm hiệu quả là phần quan trọng đối với Hệ hỗ trợ ra quyết định dạng này.

Hướng tri thức, Hệ hỗ trợ ra quyết định có thể đề nghị và đưa ra những tư vấn cho người ra quyết định. Những hệ này là các hệ chuyên gia với một kiến thức chuyên ngành cụ thể, nắm vững các vấn đề trong chuyên ngành đó và có kỹ năng để giải quyết những vấn đề này. Các công cụ khai mở dữ liệu có thể dùng để tạo ra các hệ dạng này.

Theo Holsapple và Whinston (1996) [1] phân ra 6 loại Hệ hỗ trợ ra quyết định.

- Hướng văn bản (Text, Oriented DSS);
- Hướng cơ sở dữ liệu (Database, Oriented DSS);
- Hướng bảng tính (Spreasheet, Oriented DSS);
- Hướng người giải quyết (Solver, Oriented DSS);
- Hướng luật (Rule, Oriented DSS);
- Hướng kết hợp (Compound DSS).

Hướng văn bản, Thông tin (bao gồm dữ liệu và kiến thức) được lưu trữ dưới dạng văn bản. Vì vậy hệ thống đòi hỏi lưu trữ và xử lý văn bản một cách hiệu quả. Các công nghệ mới như quản lý văn bản một cách hiệu quả. Các công nghệ mới như hệ quản lý văn bản dựa trên Web, Interlligent Agents có thể được sử dụng cùng với hệ này.

Hướng cơ sở dữ liệu, Cơ sở dữ liệu đóng vai trò chủ yếu trong hệ này. Thông tin trong cơ sở dữ liệu thường có cấu trúc chặt chẽ, các mô tả rõ ràng. Hệ này cho

phép người dùng truy vấn thông tin dễ dàng và rất mạnh về báo cáo.

Hướng bảng tính, Một bản tính là một mô hình để cho phép người dùng thực hiện việc phân tích trước khi ra quyết định. Bản tính bao gồm nhiều mô hình thống kê, lập trình tuyến tính… Bản tính phổ biến nhất Mircrosoft Excel. Hệ này được dùng rông rãi trong các hệ liên quan tới người dùng cuối.

Hướng người giải quyết, Một trợ giúp là một giải thuật hay chương trình để giải quyết một vấn đề cụ thể chẳng hạn như tính lượng hàng đặt tối ưu hay tính toán xu hướng bán hàng. Một số trợ giúp khác phức tạp như tối ưu hóa đa mục tiêu. Hệ này bao gồm nhiều trợ giúp như vậy.

Hướng luật, Kiến thức của hệ này được mô tả các quy luật thủ tục hay lý lẽ. Hệ này gọi là hệ chuyên gia. Các quy luật này có thể định tính hay định lượng. Các ví dụ của hệ này như là hướng dẫn không lưu, hướng dẫn giao thông trên biểu, trên bộ…

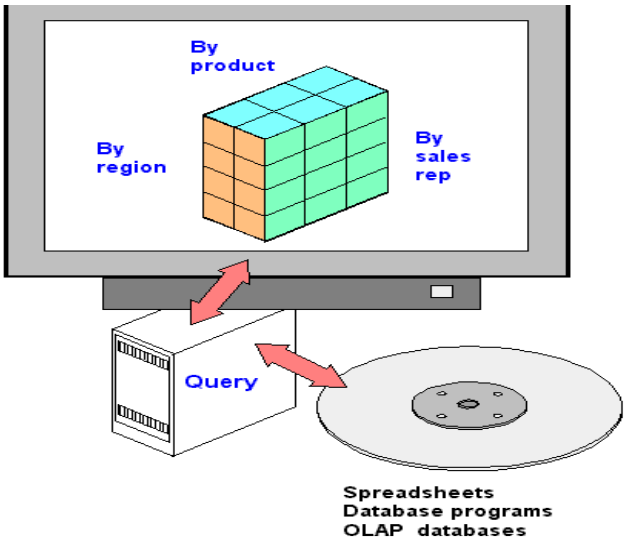
Hướng kết hợp, Một hệ tổng hợp có thể kết hợp hai hay nhiều trong số năm kể trên.

1.4.2. Năng lực của hệ hỗ trợ quyết định

Theo [1], năng lực của DSS, người ta thấy:

- Cung cấp trợ giúp cho người ra quyết định trong những tình huống không cấu trúc và nửa cấu trúc. Những tình huống này không thể giải quyết bằng các hệ thống tính toán khác.
- Sự trợ giúp được cung cấp cho các mức quản lý khác nhau từ người thực thi đến các nhà quản lý.
- Sự trợ giúp cho cá nhân và cho cả nhóm
- DSS trợ giúp cho các giai đoạn của quá trình ra quyết định: Giai đoạn trí tuệ, thiết kế, lựa chọn và cài đặt.
- DSS trợ giúp cho sự đa dạng của quá trình ra quyết định và các kiểu quyết định. Có sự phù hợp giữa DSS và tính cách của cá nhân người ra quyết định, như từ vựng và kiểu ra quyết định.
- DSS thích nghi và mềm dẻo. Do vậy người dùng có thêm xóa, kết hợp. thay đổi hoặc sắp đặt lại các phần tử cở bản để DSS có thể cung cấp sự trả lời nhanh chóng cho những tình huống không mong đợi.

- DSS dễ sử dụng, người dùng cảm thấy thoải mái đối với hệ thống do DSS thân thiện dùng, mềm, dẻo, những khả năng đồ họa mạnh và có ngôn ngữ giao diện người và máy thích hợp.
- DSS cố gắng nâng cao hiệu quả của quá trình ra quyết định, chẳng hạn như đúng đắn, chính xác, thời gian và chất lượng...
- Người ra quyết định điều khiển toàn bộ các bước của quá trình ra quyết định trong việc giải quyết các bài toán. DSS hướng vào sự trợ giúp chứ không thay thế những người ra quyết định. Người ra quyết định có thể bỏ qua lời khuyên của máy tính vào bất kỳ giai đoạn nào trong quá trình xử lý.
- DSS thường sử dụng các mô hình cho sự phân tích các tình huống ra quyết định. Khả năng mô hình hóa cho phép thí nghiệm với những chiến lược khác nhau và với những cấu hình khác nhau.
- DSS ở mức cao được trang bị thành phần trí thức, do vậy nó cho phép những giải pháp tiềm năng và hiệu quả để giải quyết những bài toán khó.



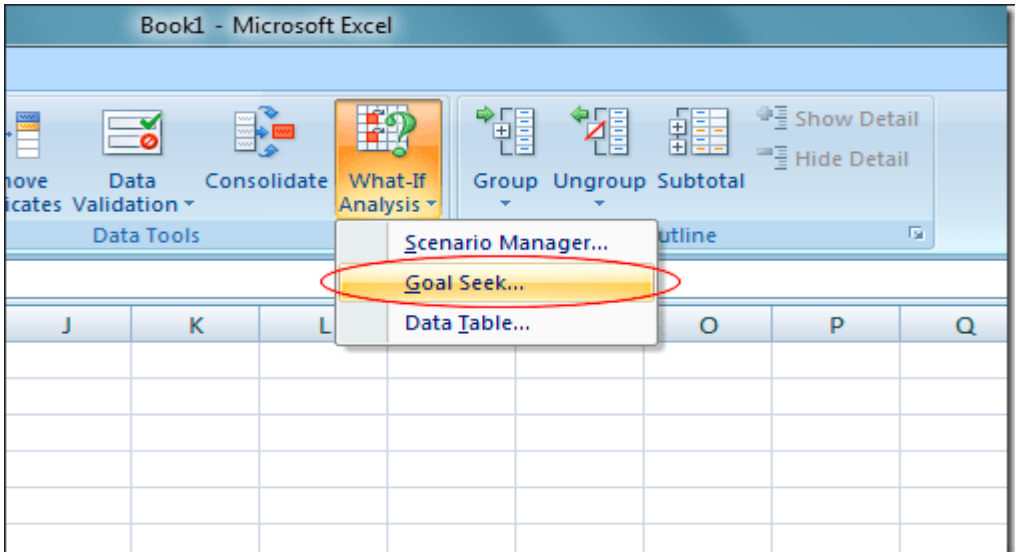
Hình 1.7. Thí dụ về DSS và EIS

(Nguồn: "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016)

1.4.3. Phân tích "What-if"

Một người làm mô hình tạo ra những dự đoán và những giả định để đánh giá dữ liệu vào. Công việc này nhiều khi để đánh giá tương lai không chắc chắn. Khi mô hình được giải quyết, các kết quả tất nhiên phụ thuộc vào những dữ liệu này.

Phân tích nhạy cảm cố gắng kiểm tra sự tác động của những sự thay đổi của dữ liệu vào trên những giải pháp được đề nghị (các biến kết quả). Kiểu của phân tích nhạy cảm được gọi là phân tích "What - if", bởi vì nó được cấu trúc như là "Điều gì xảy ra cho giải pháp nếu biến vào, giả thiết, hoặc giá trị của tham số được thay đổi"..



Hình 1.8. Thí dụ về chức năng what-if để phân tích dữ liệu

Nếu giao diện người sử dụng phù hợp thì các nhà quản lý dễ hỏi máy tính những câu hỏi kiểu như thế này. Hơn nữa họ có thể nhắc lại những câu hỏi và thay đổi tỷ lệ, hoặc thay đổi bất kỳ dữ liệu nào khác trong câu hỏiquả trong phần mềm Excel.

1.5. Cây quyết định

1.5.1. Khái niệm

Trong những năm qua, nhiều mô hình phân lớp dữ liệu đã được các nhà khoa học trong nhiều lĩnh vực khác nhau đề xuất như mạng notron, mô hình thống kê tuyến tính /bậc 2, cây quyết định, mô hình di truyền. Trong số những mô hình đó, cây quyết định với những ưu điểm của mình được đánh giá là một công cụ mạnh, phổ biến và đặc biệt thích hợp cho khai phá dữ liệu nói chung và phân lớp dữ liệu nói riêng [13].

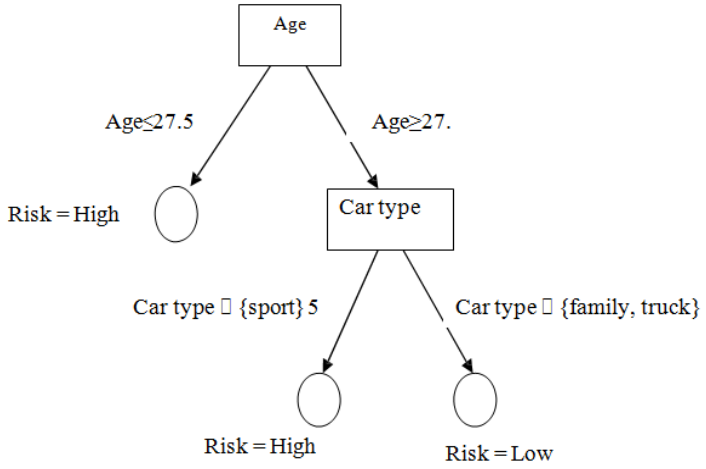
Cây quyết định là một cấu trúc ra quyết định có dạng cây (xem hình 1.9). Cây quyết định nhận đầu vào là một bộ giá trị thuộc tính mô tả một đối tượng hay một tình huống và trả về một giá trị rời rạc. Mỗi bộ thuộc tính đầu vào được gọi là một mẫu hay một ví dụ, đầu ra gọi là loại hay nhãn phân loại. Thuộc tính đầu vào còn

được gọi là đặc trưng và có thể nhận giá trị rời rạc hoặc liên tục. Để cho đơn giản, trước tiên ta sẽ xem xét thuộc tính rời rạc, sau đó sẽ mở rộng cho trường hợp thuộc tính nhận giá trị liên tục.

Cây quyết định được biểu diễn dưới dạng một cấu trúc cây (xem hình 1.9). Mỗi nút trung gian, tức là nút không phải nút lá, tương ứng với phép kiểm tra một thuộc tính.

Mỗi nhánh phía dưới của nút đó tương ứng với một giá trị của thuộc tính hay một kết quả của phép thử. Khác với nút trung gian, nút lá không chứa thuộc tính mà chứa nhãn phân loại.

Để xác định nhãn phân loại cho một ví dụ nào đó, ta cho ví dụ chuyển động từ gốc cây về phía nút lá. Tại mỗi nút, thuộc tính tương ứng với nút được kiểm tra, tùy theo giá trị của thuộc tính đó mà ví dụ được chuyển xuống nhánh tương ứng bên dưới. Quá trình này lặp lại cho đến khi ví dụ tới được nút lá và được nhận nhãn phân loại là nhãn của nút lá tương ứng.



Hình 1.9. Ví dụ về cây quyết định

(Nguồn: https://techblog.vn)

Trong cây quyết định:

- Gốc: là nút trên cùng của cây
- Nút trong: biểu diễn một kiểm tra trên một thuộc tính đơn (hình chữ nhật)
- Nhánh: biểu diễn các kết quả của kiểm tra trên nút trong (mũi tên)

- Nút lá: biểu diễn lớp hay sự phân phối lớp (hình tròn)

1.5.2. Các vấn đề khi sử dụng cây quyết định

Các vấn đề đặc thù trong khi học hay phân lớp dữ liệu bằng cây quyết định gồm: xác định độ sâu để phát triển cây quyết định, xử lý với những thuộc tính liên tục, chọn phép đo lựa chọn thuộc tính thích hợp, sử dụng tập dữ liệu đào tạo với những giá trị thuộc tính bị thiếu, sử dụng các thuộc tính với những chi phí khác nhau, và cải thiện hiệu năng tính toán. Sau đây sẽ đề cập đến những vấn đề chính đã được giải quyết trong các thuật toán phân lớp dựa trên cây quyết định.

1.5.2.1 Tránh "quá vừa" dữ liệu

Về khái niệm này, có thể hiểu đây là hiện tượng cây quyết định chứa một số đặc trưng riêng của tập dữ liệu đào tạo, nếu lấy chính tập dữ liệu huấn luyện để thử nghiệm lại mô hình phân lớp thì độ chính xác sẽ rất cao, trong khi đối với những dữ liệu tương lai khác nếu sử dụng cây đó lại không đạt được độ chính xác như vậy.

Quá vừa dữ liệu là một khó khăn đáng kể đối với học bằng cây quyết định và những phương pháp học khác. Đặc biệt khi số lượng ví dụ trong tập dữ liệu đào tạo quá ít.

Có hai phương pháp tránh "quá vừa" dữ liệu trong cây quyết định:

1. Dừng phát triển cây sớm hơn bình thường, trước khi đạt tới điểm phân lớp hoàn hảo tập dữ liệu đào tạo. Với phương pháp này, một thách thức đặt ra là phải ước lượng chính xác thời điểm dừng phát triển cây.
2. Cho phép cây có thể "quá vừa" dữ liệu, sau đó sẽ cắt, tỉa cây.

Mặc dù phương pháp thứ nhất có vẻ trực tiếp hơn, nhưng với phương pháp thứ hai thì cây quyết định được sinh ra được thực nghiệm chứng minh là thành công hơn trong thực tế. Hơn nữa việc cắt tỉa cây quyết định còn giúp tổng quát hóa, và cải thiện độ chính xác của mô hình phân lớp. Dù thực hiện phương pháp nào thì vấn đề mấu chốt ở đây là tiêu chuẩn nào được sử dụng để xác định kích thước hợp lý của cây cuối cùng.

1.5.2.2. Thao tác với thuộc tính liên tục

Việc thao tác với thuộc tính liên tục trên cây quyết định hoàn toàn không đơn giản như với thuộc tính rời rạc.

Thuộc tính rời rạc có tập giá trị (domain) xác định từ trước và là tập hợp các giá trị rời rạc. Ví dụ loại ô tô là một thuộc tính rời rạc với tập giá trị là: {xe tải, xe khách, xe con, taxi}.Việc phân chia dữ liệu dựa vào phép kiểm tra giá trị của thuộc tính rời rạc được chọn tại một ví dụ cụ thể có thuộc tập giá trị của thuộc tính đó hay không: value (A) ∈ X với X ⊂ domain (A). Đây là phép kiểm tra logic đơn giản, không tốn nhiều tài nguyên tính toán. Trong khi đó, với thuộc tính liên tục (thuộc tính dạng số) thì tập giá trị là không xác định trước. Chính vì vậy, trong quá trình phát triển cây, cần sử dụng kiểm tra dạng nhị phân: value (A) ≤ θ. Với θ là hằng số ngưỡng (threshold) được lần lượt xác định dựa trên từng giá trị riêng biệt hay từng cặp giá trị liền nhau (theo thứ tự đã sắp xếp) của thuộc tính liên tục đang xem xét trong tập dữ liệu đào tạo. Điều đó có nghĩa là nếu thuộc tính liên tục A trong tập dữ liệu đào tạo có d giá trị phân biệt thì cần thực hiện d-1 lần kiểm tra value (A) ≤ θi với i = 1..d-1 để tìm ra ngưỡng θ tốt nhất tương ứng với thuộc tính đó. Việc xác định giá trị của θ và tiêu chuẩn tìm θ tốt nhất tùy vào chiến lược của từng thuật toán. Trong thuật toán C4.5, θi được chọn là giá trị trung bình của hai giá trị liền kề nhau trong dãy giá trị đã sắp xếp.

Ngoài ra còn một số vấn đề liên quan đến sinh tập luật, xử lý với giá trị thiếu sẽ được trình bày cụ thể trong phần thuật toán C4.5.

1.5.3. Đánh giá cây quyết định trong lĩnh vực khai phá dữ liệu

1.5.3.1 Sức mạnh của cây quyết định

Cây quyết định có 5 sức mạnh chính sau [12]:

- *Khả năng sinh ra các quy tắc hiểu được*. Cây quyết định có khả năng sinh ra các quy tắc có thể chuyển đổi được sang dạng tiếng Anh, hoặc các câu lệnh SQL. Đây là ưu điểm nổi bật của kỹ thuật này. Thậm chí với những tập dữ liệu lớn khiến cho hình dáng cây quyết định lớn và phức tạp, việc đi theo bất cứ đường nào trên cây là dễ dàng theo nghĩa phổ biến và rõ ràng. Do vậy sự giải thích cho bất cứ một sự phân lớp hay dự đoán nào đều tương đối minh bạch.
- *Khả năng thực thi trong những lĩnh vực hướng quy tắc*. Điều này có nghe có vẻ hiển nhiên, nhưng quy tắc quy nạp nói chung và cây quyết định nói riêng

là lựa chọn hoàn hảo cho những lĩnh vực thực sự là các quy tắc. Rất nhiều lĩnh vực từ di truyền tới các quá trình công nghiệp thực sự chứa các quy tắc ẩn, không rõ ràng (underlying rules) do khá phức tạp và tối nghĩa bởi những dữ liệu lỗi (noisy). Cây quyết định là một sự lựa chọn tự nhiên khi chúng ta nghi ngờ sự tồn tại của các quy tắc ẩn, không rõ ràng.

- *Dễ dàng tính toán trong khi phân lớp*. Mặc dù như chúng ta đã biết, cây quyết định có thể chứa nhiều định dạng, nhưng trong thực tế, các thuật toán sử dụng để tạo ra cây quyết định thường tạo ra những cây với số phân nhánh thấp và các test đơn giản tại từng node. Những test điển hình là: so sánh số, xem xét phần tử của một tập hợp, và các phép nối đơn giản. Khi thực thi trên máy tính, những test này chuyển thành các toán hàm logic và số nguyên là những toán hạng thực thi nhanh và không đắt. Đây là một ưu điểm quan trọng bởi trong môi trường thương mại, các mô hình dự đoán thường được sử dụng để phân lớp hàng triệu thậm trí hàng tỉ bản ghi.
- *Khả năng xử lý với cả thuộc tính liên tục và thuộc tính rời rạc*. Cây quyết định xử lý "tốt" như nhau với thuộc tính liên tục và thuộc tính rời rạc. Tuy rằng với thuộc tính liên tục cần nhiều tài nguyên tính toán hơn. Những thuộc tính rời rạc đã từng gây ra những vấn đề với mạng neural và các kỹ thuật thống kê lại thực sự dễ dàng thao tác với các tiêu chuẩn phân chia (splitting criteria) trên cây quyết định: mỗi nhánh tương ứng với từng phân tách tập dữ liệu theo giá trị của thuộc tính được chọn để phát triển tại node đó. Các thuộc tính liên tục cũng dễ dàng phân chia bằng việc chọn ra một số gọi là ngưỡng trong tập các giá trị đã sắp xếp của thuộc tính đó. Sau khi chọn được ngưỡng tốt nhất, tập dữ liệu phân chia theo test nhị phân của ngưỡng đó.
- *Thể hiện rõ ràng những thuộc tính tốt nhất*. Các thuật toán xây dựng cây quyết định đưa ra thuộc tính mà phân chia tốt nhất tập dữ liệu đào tạo bắt đầu từ node gốc của cây. Từ đó có thể thấy những thuộc tính nào là quan trọng nhất cho việc dự đoán hay phân lớp.

1.5.3.2. Điểm yếu của cây quyết định

Dù có những sức mạnh nổi bật trên, cây quyết định vẫn không tránh khỏi có những điểm yếu. Đó là cây quyết định không thích hợp lắm với những bài toán với mục tiêu là dự đoán giá trị của thuộc tính liên tục như thu nhập, huyết áp hay lãi xuất ngân hàng, … Cây quyết định cũng khó giải quyết với những dữ liệu thời gian liên tục nếu không bỏ ra nhiều công sức cho việc đặt ra sự biểu diễn dữ liệu theo các mẫu liên tục.

- *Dễ xảy ra lỗi khi có quá nhiều lớp*. Một số cây quyết định chỉ thao tác với những lớp giá trị nhị phân dạng yes/no hay accept/reject. Số khác lại có thể chỉ định các bản ghi vào một số lớp bất kỳ, nhưng dễ xảy ra lỗi khi số ví dụ đào tạo ứng với một lớp là nhỏ. Điều này xẩy ra càng nhanh hơn với cây mà có nhiều tầng hay có nhiều nhánh trên một node.
- *Chi phí tính toán đắt để đào tạo*. Điều này nghe có vẻ mâu thuẫn với khẳng định ưu điểm của cây quyết định ở trên. Nhưng quá trình phát triển cây quyết định đắt về mặt tính toán. Vì cây quyết định có rất nhiều node trong trước khi đi đến lá cuối cùng. Tại từng node, cần tính một độ đo (hay tiêu chuẩn phân chia) trên từng thuộc tính, với thuộc tính liên tục phải thêm thao tác xắp xếp lại tập dữ liệu theo thứ tự giá trị của thuộc tính đó. Sau đó mới có thể chọn được một thuộc tính phát triển và tương ứng là một phân chia tốt nhất. Một vài thuật toán sử dụng tổ hợp các thuộc tính kết hợp với nhau có trọng số để phát triển cây quyết định. Quá trình cắt cụt cây cũng "đắt" vì nhiều cây con ứng cử phải được tạo ra và so sánh.

1.5.3.3 Xây dựng cây quyết định

Quá trình xây dựng cây quyết định gồm hai giai đoạn:

- *Giai đoạn thứ nhất phát triển cây quyết định:* Giai đoạn này phát triển bắt đầu từ gốc, đến từng nhánh và phát triển quy nạp theo cách thức chia để trị cho tới khi đạt được cây quyết định với tất cả các lá được gán nhãn lớp.
- *Giai đoạn thứ hai cắt, tỉa bớt các cành nhánh trên cây quyết định.* Giai đoạn này nhằm mục đích đơn giản hóa và khái quát hóa từ đó làm tăng độ chính

xác của cây quyết định bằng cách loại bỏ sự phụ thuộc vào mức độ lỗi (noise) của dữ liệu đào tạo mang tính chất thống kê, hay những sự biến đổi mà có thể là đặc tính riêng biệt của dữ liệu đào tạo. Giai đoạn này chỉ truy cập dữ liệu trên cây quyết định đã được phát triển trong giai đoạn trước và quá trình thực nghiệm cho thấy giai đoạn này không tốn nhiều tài nguyên tính toán, như với phần lớn các thuật toán, giai đoạn này chiếm khoảng dưới 1% tổng thời gian xây dựng mô hình phân lớp.

Do vậy, ở đây chỉ tập trung vào nghiên cứu giai đoạn phát triển cây quyết định. Dưới đây là khung công việc của giai đoạn này:

- *Chọn thuộc tính "tốt" nhất bằng một độ đo đã định trước*
- *Phát triển cây bằng việc thêm các nhánh tương ứng với từng giá trị của thuộc tính đã chọn*
- *Sắp xếp, phân chia tập dữ liệu đào tạo tới node con*
- *Nếu các ví dụ được phân lớp rõ ràng thì dừng. Ngược lại: lặp lại bước 1 tới bước 4 cho từng node con*

1.5.3.4. Giải thuật xây dựng cây quyết định

Trước khi sử dụng cây quyết định, ta cần xây dựng hay "học" cây quyết định từ dữ liệu huấn luyện. Có nhiều thuật toán khác nhau được đề xuất và sử dụng để học cây quyết định từ dữ liệu, trong đó đa số dựa trên nguyên tắc chung là xây dựng cây theo kiểu tìm kiếm tham lam, chia để trị, đệ qui từ cây đơn giản tới cây phức tạp hơn.

Phần lớn các thuật toán phân lớp dữ liệu dựa trên cây quyết định có cấu trúc như sau:

```
Make Tree (Training Data T)
{
   Partition (T)
}
Partition (Data S)
{
if (all points in S are in the same class) then return
for each attribute A do
   evaluate splits on attribute A;
   use best split found to partition S into S1, S2,..., Sk
   Partition (S1) Partition (S2)
...
   Partition (Sk)
 }
```

Mô tả: Xây dựng cây quyết định từ T là tập training data và các lớp được biểu diễn dưới dạng tập C = {C1, C2, …,Ck }

- *Trường hợp 1*: T chứa các case thuộc về một lớp đơn Cj, cây quyết định ứng với T là một lá tương ứng với lớp Cj
- *Trường hợp 2*: T chứa các case thuộc về nhiều lớp khác nhau trong tập C. Một kiểm tra được chọn trên một thuộc tính có nhiều giá trị {O1, O2, ….,On }. Tập T được chia thành các tập con T1, T2, …, Tn, với Ti chứa tất cả các case trong T mà có kết quả là Oi trong kiểm tra đã chọn. Cây quyết định ứng với T bao gồm một node biểu diễn kiểm tra được chọn, và mỗi nhánh tương ứng với mỗi kết quả có thể của kiểm tra đó. Cách thức xây dựng cây tương tự được áp dụng đệ quy cho từng tập con của tập training data.
- *Trường hợp 3*: T không chứa case nào. Cây quyết định ứng với T là một lá, nhưng lớp gắn với lá đó phải được xác định từ những thông tin khác ngoài T.

1.6. Các thuật toán cây quyết định

1.6.1. Thuật toán ID3

1.6.1.1. Mô tả

Trong ID3, chúng ta cần xác định thứ tự của thuộc tính cần được xem xét tại mỗi bước. Với các bài toán có nhiều thuộc tính và mỗi thuộc tính có nhiều giá trị khác nhau, việc tìm được nghiệm tối ưu thường là không khả thi. Thay vào đó, một phương pháp đơn giản thường được sử dụng là tại mỗi bước, một thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó. Với mỗi thuộc tính được chọn, ta chia dữ liệu vào các child node tương ứng với các giá trị của thuộc tính đó rồi tiếp tục áp dụng phương pháp này cho mỗi child node. Việc chọn ra thuộc tính tốt nhất ở mỗi bước như thế này được gọi là cách chọn greedy (tham lam). Cách chọn này có thể không phải là tối ưu, nhưng trực giác cho chúng ta thấy rằng cách làm này sẽ gần với cách làm tối ưu. Ngoài ra, cách làm này khiến cho bài toán cần giải quyết trở nên đơn giản hơn.

Sau mỗi câu hỏi, dữ liệu được phân chia vào từng child node tương ứng với các câu trả lời cho câu hỏi đó. Câu hỏi ở đây chính là một thuộc tính, câu trả lời

chính là giá trị của thuộc tính đó. Để đánh giá chất lượng của một cách phân chia, chúng ta cần đi tìm một phép đo.

Trước hết, thế nào là một phép phân chia tốt? Bằng trực giác, một phép phân chia là tốt nhất nếu dữ liệu trong mỗi child node hoàn toàn thuộc vào một class–khi đó child node này có thể được coi là một leaf node, tức ta không cần phân chia thêm nữa. Nếu dữ liệu trong các child node vẫn lẫn vào nhau theo tỉ lệ lớn, ta coi rằng phép phân chia đó chưa thực sự tốt. Từ nhận xét này, ta cần có một hàm số đo độ tinh khiết (purity), hoặc độ vẩn đục (impurity) của một phép phân chia. Hàm số này sẽ cho giá trị thấp nhất nếu dữ liệu trong mỗi child node nằm trong cùng một class (tinh khiết nhất), và cho giá trị cao nếu mỗi child node có chứa dữ liệu thuộc nhiều class khác nhau.

Một hàm số có các đặc điểm này và được dùng nhiều trong lý thuyết thông tin là hàm entropy.

1.6.1.2. Hàm số entropy

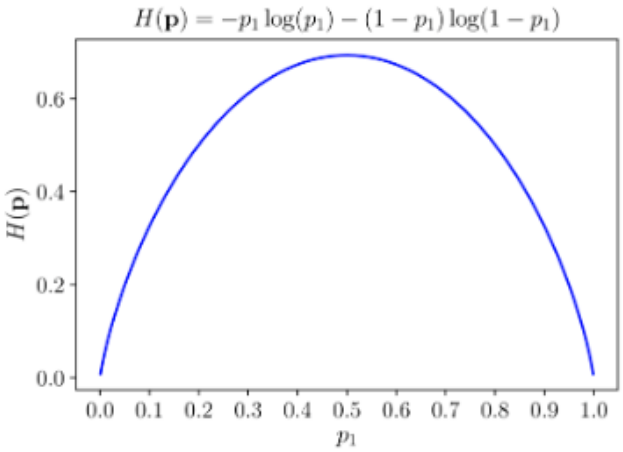
Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x_1, x_2,…, x_n x1,x2,…,xn. Giả sử rằng xác suất để x nhận giá trị này là $p_i=p(x=x_i)$ với $0 \leq p_i \leq 1, \sum_{i=1}^{n} p_i = 1$

Ký hiệu phân phối này là $p=(p_1,p_2,\ldots p_n)$ Entropy của phân phối này được định nghĩa là $$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log(p_i)$$

Xét một ví dụ với n=2 được cho trên hình bên dưới Trong trường hợp p là tinh khiết nhất, tức một trong hai giá trị p_i bằng 1, giá trị kia bằng 0, entropy của phân phối này là H(p)=0. Khi p là vẩn đục nhất, tức cả hai giá trị $p_i=0.5$ hàm entropy đạt giá trị cao nhất.

Hàm Entropy được biểu diễn dưới dạng đồ thị như trong hình. Tổng quát lên với n>2 hàm entropy đạt giá trị nhỏ nhất nếu có một giá trị $p_i=1$ đạt giá trị lớn nhất nếu tất cả các p_i bằng nhau ((việc này có thể được chứng minh bằng phương pháp nhân tử Lagrange). Trong ID3, tổng các trọng số của entropy tại các leaf-node sau khi xây dựng cây quyết định được coi là hàm mất mát của cây quyết định đó. Các

trọng số ở đây tỉ lệ với số điểm dữ liệu được phân vào mỗi node. Công việc của ID3 là tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt.



Hình 1.10. Hàm số entropy

(Nguồn: https://machinelearningcoban.com)

Như đã đề cập, việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất. Bài toán xây dựng một cây quyết định bằng ID3 có thể chia thành các bài toán nhỏ, trong mỗi bài toán, ta chỉ cần chọn ra thuộc tính giúp cho việc phân chia đạt kết quả tốt nhất. Mỗi bài toán nhỏ này tương ứng với việc phân chia dữ liệu trong một non-leaf node. Chúng ta sẽ xây dựng phương pháp tính toán dựa trên mỗi node này.

Xét một bài toán với C class khác nhau. Giả sử ta đang làm việc với một non-leaf node với các điểm dữ liệu tạo thành một tập S với số phần tử là |S|=N Giả sử thêm rằng trong số N điểm dữ liệu này N_c, c=1,2…,C điểm thuộc vào class C. Xác suất để mỗi điểm dữ liệu rơi vào một class C được xấp xỉ bằng N_c/N. Như vậy, entropy tại node này được tính bởi: $$H(\mathcal{S}) = -\sum_{c=1}^{C} \frac{N_c}{N} \log\left(\frac{N_c}{N}\right)$$.

Tiếp theo, giả sử thuộc tính được chọn là x. Dựa trên x, các điểm dữ liệu trong S được phân ra thành K child node S_1, S_2,…S_k với số điểm trong mỗi child

node lần lượt là $m_1, m_2, \ldots m_k$. Ta định nghĩa $H(x,\mathcal{S}) = \sum_{k=1}^{K} \frac{m_k}{N} H(\mathcal{S}_k)$ là tổng có trọng số entroy của mỗi child node. Việc lấy trọng số này là quan trọng vì các node thường có số lượng điểm khác nhau. Tiếp theo, ta định nghĩa information gain dựa trên thuộc tính x: $G(x,\mathcal{S}) = H(\mathcal{S}) - H(x,\mathcal{S})$. Trong ID3, tại mỗi node, thuộc tính được chọn được xác định dựa trên: $x^* = \arg\max_x G(x,\mathcal{S}) = \arg\min_x H(x,\mathcal{S})$, tức thuộc tính khiến cho information gain đạt giá trị lớn nhất.

1.6.1.3. Thuật toán ID3

Function Build_Tree(tap_du_lieu,tap_thuoc_tinh)

Begin

If mọi bộ dữ liệu trong tap_du_lieu đều nằm trong cùng lớp ***then***

return một nút lá được gán nhãn bởi lớp đó

else if tập_thuộc_tính là rỗng ***then***

return nút lá được gán nhãn bởi hợp của tất cả các lớp trong tap_du_lieu ***else***

begin

chọn một thuộc tính p, lấy nó làm gốc cho cây hiện tại;

xóa p ra khỏi tập thuộc tính;

với mỗi giá trị V của p

begin

tạo một nhánh của cây gán nhãn V;

Đặt vào phân vùng các ví dụ trong tộp ví dụ có giá trị V tại thuộc tỉnh P;

call Build_Tree(phân vùng V, tập_thuộc_tính), gắn kết quả vào nhánh V

end

end

end

1.6.1.3. Ví dụ

Xét ví dụ với dữ liệu huấn luyện được cho trong bảng dưới đây:

Bảng 1.1. Dữ liệu thí dụ cho thuật toán ID3

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

(Nguồn: https://machinelearningcoban.com)

Có bốn thuộc tính thời tiết:

- Outlook nhận một trong ba giá trị: sunny, overcast, rainy.
- Temperature nhận một trong ba giá trị: hot, cool, mild.
- Humidity nhận một trong hai giá trị: high, normal.
- Wind nhận một trong hai giá trị: weak, strong.

Đây có thể được coi là một bài toán dự đoán liệu đội bóng có chơi bóng không dựa trên các quan sát thời tiết. Cách dự đoán dưới đây tương đối đơn giản và khá chính xác, có thể không phải là cách ra quyết định tốt nhất:

- Nếu outlook = sunny và humidity = high thì play = no
- Nếu outlook = rainy và windy = true thì play = no
- Nếu outlook = overcast thì play = yes
- Ngoài ra, nếu humidity = normal thì play = yes.
- Ngoài ra, play = yes.

Chúng ta sẽ cùng tìm thứ tự các thuộc tính bằng thuật toán ID3. Trong 14 giá

trị đầu ra ở Bảng trên, có năm giá trị bằng no và chín giá trị bằng yes. Entroy tại

$$H(S) = -\frac{5}{14}\log\left(\frac{5}{14}\right) - \frac{9}{14}\log\left(\frac{9}{14}\right) \approx 0.65$$

root node của bài toán là:

Tiếp theo, chúng ta tính tổng có trọng số entropy của các child node nếu chọn một trong các thuộc tính outlook, temperature, humidity, wind, play để phân chia dữ liệu. Xét thuộc tính outlook. Thuộc tính này có thể nhận một trong ba giá trị sunny, overcast, rainy. Mỗi một giá trị sẽ tương ứng với một child node. Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_s, S_o, S_r với tương ứng m_s, m_o, m_r phần tử. Sắp xếp lại Bảng ban đầu theo thuộc tính outlook ta đạt được ba Bảng nhỏ sau đây.

Bảng 1.2. Ba bảng dữ liệu

id	**outlook**	**temperature**	**humidity**	**wind**	**play**
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
11	sunny	mild	normal	strong	yes

id	**outlook**	**temperature**	**humidity**	**wind**	**play**
3	overcast	hot	high	weak	yes
7	overcast	cool	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes

id	**outlook**	**temperature**	**humidity**	**wind**	**play**
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

(Nguồn: https://machinelearningcoban.com)

Quan sát nhanh ta thấy rằng child node ứng với outlook = overcast sẽ có

entropy bằng 0 vì tất cả m_o=4 output đều là yes. Hai node con còn lại với m_s=m_r=5 có entropy khá cao vì tần suất output bằng yes hoặc no là xấp xỉ nhau. Tuy nhiên, hai child node này có thể được phân chia tiếp dựa trên hai thuộc tính humidity và wind

$$H(\mathcal{S}_s) = -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right) \approx 0.673$$
$$H(\mathcal{S}_o) = 0$$
$$H(\mathcal{S}_r) = -\frac{3}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right) \approx 0.673$$
$$H(outlook, \mathcal{S}) = \frac{5}{14}H(\mathcal{S}_s) + \frac{4}{14}H(\mathcal{S}_o) + \frac{5}{14}H(\mathcal{S}_r) \approx 0.48$$

Xét thuộc tính temperature, ta có phân chia như các Bảng dưới đây.

Bảng 1.3. Bảng về thuộc tính nhiệt độ

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes

(Nguồn: https://machinelearningcoban.com)

Gọi S_h, S_m, S_c là ba tập con tương ứng với temperature bằng hot, mild, cool.

Bạn đọc có thể tính được

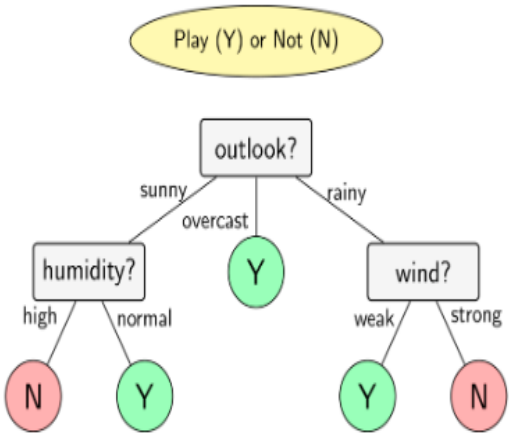
$$H(\mathcal{S}_h) = -\frac{2}{4}\log\left(\frac{2}{4}\right) - \frac{2}{4}\log\left(\frac{2}{4}\right) \approx 0.693$$
$$H(\mathcal{S}_m) = -\frac{4}{6}\log\left(\frac{4}{6}\right) - \frac{2}{6}\log\left(\frac{2}{6}\right) \approx 0.637$$
$$H(\mathcal{S}_c) = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) \approx 0.562$$
$$H(temperature, \mathcal{S}) = \frac{4}{14}H(\mathcal{S}_h) + \frac{6}{14}H(\mathcal{S}_m) + \frac{4}{14}H(\mathcal{S}_c) \approx 0.631$$

Việc tính toán với hai thuộc tính còn lại được dành cho bạn đọc. Nếu các kết quả là giống nhau, chúng sẽ bằng: H(humidity, S) ≈ 0.547, H(wind,S) ≈ 0.618.

Như vậy, thuộc tính cần chọn ở bước đầu tiên là outlook vì H(outlook,S) đạt giá trị nhỏ nhất (information gain là lớn nhất).

Sau bước phân chia đầu tiên này, ta nhận được ba child node với các phần tử như trong ba Bảng phân chia theo outlook. Child node thứ hai không cần phân chia tiếp vì nó đã tinh khiết. Với child node thứ nhất, ứng với outlook = sunny, kết quả tính được bằng ID3 sẽ cho chúng ta thuộc tính humidity vì tổng trọng số của entropy sau bước này sẽ bằng 0 với output bằng yes khi và chỉ khi humidity = normal. Tương tự, child node ứng với outlook = wind sẽ được tiếp tục phân chia bởi thuộc tính wind với output bằng yes khi và chỉ khi wind = weak.

Như vậy, cây quyết định cho bài toán này dựa trên ID3 sẽ có dạng như sau:



Hình 1.11. Đồ thị cây quyết định, sử dụng thuật toán ID3

(Nguồn: https://machinelearningcoban.com)

1.6.2. Thuật toán C4.5

1.6.2.1. Giới thiệu C4.5

Thuật toán C4.5 cũng được tác giả Quinlan phát triển và công bố vào năm 1996. Thuật toán này là một thuật toán được cải tiến từ thuật toán ID3 và giải quyết hầu hết các vấn đề mà ID3 chưa giải quyết như đã nêu trên. Nó thực hiện phân lớp tập mẫu dữ liệu theo chiến lược ưu tiên theo chiều sâu (Depth - First).

Thuật toán C4.5 là một thuật toán phân lớp dữ liệu hiệu quả và thông dụng được sử dụng trong bài toán cây quyết định. Là phiên bản cải tiến của thuật toán ID3. Loại bỏ điều kiện hạn chế của ID3: các đặc trưng phải là đặc trưng có thể phân loại bằng cách xác định một thuộc tính rời rạc (dựa trên các biến số) sao cho thuộc tính này có thể chia giá trị thuộc tính liên tục thành tập các khoảng rời rạc. Cây ở C4.5 được chuyển thành tập các lệnh if-then. Sau đó đánh giá độ chính xác của từng luật và sắp xếp thứ tự luật được áp dụng trong quá trình ra quyết định.

- Quá trình Pruning cũng khác so với ID3: C4.5 sẽ quyết định loại bỏ điều kiện tiên quyết của luật nếu như độ chính xác (accuracy) của luật này có thể tăng mà không phụ thuộc vào việc có sử dụng luật hay không. C4.5 có những đăc điểm khác với các thuật toán khác, đó là: cơ chế chọn thuộc tính để kiểm tra tại mỗi node. C4.5 dựa vào nghiên cứu tối ưu hóa, và sự lựa chọn cách phân chia mà có độ đo lựa chọn thuộc tính đạt giá trị cực đại. Độ đo được sử dụng trong C4.5 là Information Gain và Gain Ratio.

Để có thể dễ dàng tiếp cận với thuật toán C4.5, chúng ta sẽ phải nắm được khái niệm về “Information Gain” và “Entropy”.

1.6.2.2. Thuật toán C4.5

Function xaydungcay(T)

{

<Tính toán tần xuất các giá trị trong các lớp của T>;

If <Kiểm tra các mẫu, nếu thuộc cùng một lớp hoặc có rất ít mẫu khác lớp>Then <Trả về 1 nút lá>

Else <Tạo một nút quyết định N>;

For <với mỗi thuộc tính A> Do <Tính giá trị Gain(A)>;

<Tại nút N, thực hiện việc kiểm tra đế chọn ra thuộc tính có giá trị Gain tốt nhất (lớn nhất). Gọi N.test là thuộc tính có Gain lớn nhất>; lf<Nếu N.test là thuộc tính liên tục> Then <Tìm ngưỡng cho phép tách của N.test>;

For <với mỗi tập con T' được tách ra từ tập T> Do (T' được tách ra theo quy tắc:

- Nếu N.test là thuộc tính liên tục tách theo ngưỡng ở bước 5
- Nếu N.test là thuộc tính phân loại rời rạc tách theo các giá trị của thuộc tính này.

)

{ lf<Kiếm tra, nếu V rẫng>} Then

<Gán nút con này của nút N là nút lá>;

Else

<Gán nút con này là nút được trả về bằng cách gọi đệ qui lại đối với hàm xay dung cay(T'), với tập T'>;

<Tính toán các lỗi của nút N>;

<Trả về nút N>;

}

C4.5 có những đăc điểm khác với các thuật toán khác, đó là: cơ chế chọn thuộc tính để kiểm tra tại mỗi node, cơ chế xử lý với những giá trị thiếu, việc tránh "quá vừa" dữ liệu, ước lượng độ chính xác và cơ chế cắt tỉa cây.

1.6.2.3. Lượng thông tin Entropy

Trước khi định nghĩa entropy. Ta phải hiểu về dữ liệu không sạch, hỗn loạn hoặc không chắc chắn. Đây là tập dữ liệu mà khả năng nhận được thông tin chính xác là không cao.

Entropy là đại lượng quyết định việc điều chỉnh Cây quyết định sẽ phân nhánh như thế nào. Nó ảnh hưởng trực tiếp tới hình dáng của cây. Entropy có công thức $H(S) = -\sum_{i=1}^{n} p(X_i).\log_2 p(X_i)$; trong đó $p(X_i)$ là xác xuất xảy ra của biến cố X_i. n là số quyết định cuối cùng có thể xảy ra.

1.6.2.4. Lượng thông tin Information Gain

Lượng thông tin là đại lượng chỉ số lượng thông tin mà ta nhận được từ một sự kiện xảy ra. Information Gain là chìa khóa chính trong các thuật toán xây dựng

cây quyết định. Các thuật toán xây dựng cây sẽ luôn tìm cách để tìm giá trị lớn nhất của Information Gain. Một thuộc tính với Information Gain cao nhất sẽ được ưu tiên chọn làm nút trong cây. Information Gain có công thức $Gain(S,A) = H(S) - \sum_v \frac{|S_v|}{|S|}.H(S_v)$; trong đó, *v* là các giá trị có thể xảy ra của thuộc tính A, S là tập tất cả quan sát. S_v là tập con mà tại đó giá trị của A bằng v.

Nếu như trong các thuật toán xây dựng cây quyết định, Information Gain đóng vai trò quan trọng nhất thì trong C4.5 có một đại lượng khác đóng vai trò quyết định đó là Information Gain Ratio (Gain Ratio).

1.6.2.5. Information Gain Ratio

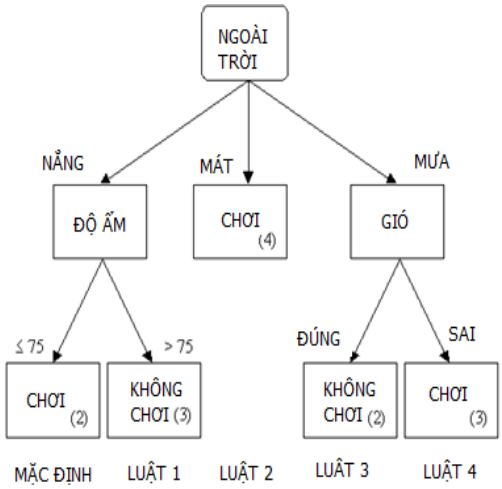
Giá trị phân chia thông tin (split information value) là giá trị biểu diễn thông tin tiềm ẩn khi ta chia tập dữ liệu *D* thành *n* phần (*n* là số các giá trị có thể đạt được của thuộc tính *A*).

$$SI_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|}.\log_2\left(\frac{|D_j|}{|D|}\right)$$

Gain ratio được định nghĩa như sau:

$$GainRatio(S,A) = \frac{Gain(S,A)}{SI(A)}$$

Thuộc tính nào có gain ratio lớn nhất sẽ được ưu tiên làm nút trước các thuộc tính có gain raio thấp hơn.



Hình 1.12. Ví dụ Cây quyết định tạo bởi thuật toán C4.5

1.6.2.6. Xén cây Pruning

Thuật toán C4.5 có một đặc điểm nổi bật đó là Pruning (cắt tỉa). Pruning là một kỹ thuật giúp xóa bỏ các nhánh trong cây mà các nhánh đó không ảnh hưởng đáng kể tới quyết định cuối cùng.

Quay trở lại ví dụ ban đầu, nếu ta *Giàu* thì ta sẽ đi *Mua sắm*. Giả sử có thêm một nút nữa là *Phương tiện*. Nếu ta có *Phương tiện* thì ta sẽ đi *Mua sắm* bằng *Ô tô*, nếu không thì *Đi bộ* đi *Mua sắm*. Tức là dù có *Phương tiện* hay không thì ta vẫn đi *Mua sắm*. Cho nên ta có thể bỏ nhánh *Phương tiện* đi mà không làm ảnh hưởng tới quyết định cuối. Việc cài đặt thuật toán C4.5 sẽ dành trong chương 2.

1.6.2.7. Chuyển đổi từ cây quyết định sang luật "if – then"

Việc chuyển đổi từ cây quyết định sang luật (rules) dạng if-then tạo ra những quy tắc phân lớp dễ hiểu, dễ áp dụng. Các mô hình phân lớp biểu diễn các khái niệm dưới dạng các luật sản xuất đã được chứng minh là hữu ích trong nhiều lĩnh vực khác nhau, với các đòi hỏi về cả độ chính xác và tính hiểu được của mô hình phân lớp. Dạng kết quả đầu ra tập luật sản xuất là sự lựa chọn "khôn ngoan". Tuy nhiên, tài nguyên tính toán dùng cho việc tạo ra tập luật từ tập dữ liệu đào tạo có kích thước lớn và nhiều giá trị sai là vô cùng lớn [12]. Khẳng định này sẽ được chứng minh qua kết quả thực nghiệm trên mô hình phân lớp C4.5

Giai đoạn chuyển đổi từ cây quyết định sang luật bao gồm 4 bước:

Bước 1 cắt tỉa: Luật khởi tạo ban đầu là đường đi từ gốc đến lá của cây quyết định. Một cây quyết định có l lá thì tương ứng tập luật sản xuất sẽ có l luật khởi tạo. Từng điều kiện trong luật được xem xét và loại bỏ nếu không ảnh hưởng tới độ chính xác của luật đó. Sau đó, các luật đã cắt tỉa được thêm vào tập luật trung gian nếu nó không trùng với những luật đã có.

*Bước 2 l*ựa chọn: Các luật đã cắt tỉa được nhóm lại theo giá trị phân lớp, tạo nên các tập con chứa các luật theo lớp. Sẽ có k tập luật con nếu tập training có k giá trị phân lớp. Từng tập con trên được xem xét để chọn ra một tập con các luật mà tối ưu hóa độ chính xác dự đoán của lớp gắn với tập luật đó.

*Bước 3 s*ắp xếp: Sắp xếp K tập luật đã tạo ra từ trên bước theo tần số lỗi. Lớp mặc định được tạo ra bằng cách xác định các trường hợp trong tập training không chứa trong các luật hiện tại và chọn lớp phổ biến nhất trong các trường hợp đó làm lớp mặc định.

Bước 4 ước lượng, đánh giá: Tập luật được đem ước lượng lại trên toàn bộ tập training, nhằm mục đích xác định xem liệu có luật nào làm giảm độ chính xác của sự phân lớp. Nếu có, luật đó bị loại bỏ và quá trình ước lượng được lặp cho đến khi không thể cải tiến thêm.

1.7. Kết luận

Chương 1 của luận văn đã trình bày hệ thống trợ giúp quyết định DSS và công việc khai phá dữ liệu.

Các hệ trợ giúp quyết định DSS là hệ thống cơ sở máy tính được thiết kế nâng cao hiệu quả của những người làm quyết định từng bài toán cụ thể. Nói cách khác, hệ trợ giúp quyết định DSS là hệ thống dựa trên tương tác máy tính, giúp người ra quyết định dùng dữ liệu và các mô hình để giải quyết bài toán nửa cấu trúc hoặc phi cấu trúc. Hệ trợ giúp quyết định DSS được sử dụng để trợ giúp công tác quản lý. Hệ trợ giúp quyết định DSS hoàn chỉnh có thể đưa ra những thông tin trợ giúp cho sự giải quyết các vấn đề.

Cây quyết định được sử dụng để xây dựng một kế hoạch nhằm đạt được mục tiêu mong muốn đó là hỗ trợ ra quyết định.

Cây quyết định có khả năng sinh ra các quy tắc có thể chuyển đổi được sang dạng tiếng Anh, hoặc các câu lệnh SQL. Đây là ưu điểm nổi bật của kỹ thuật này. Thậm chí với những tập dữ liệu lớn khiến cho hình dáng cây quyết định lớn và phức tạp, việc đi theo bất cứ đường nào trên cây là dễ dàng theo nghĩa phổ biến và rõ ràng. Do vậy sự giải thích cho bất cứ một sự phân lớp hay dự đoán nào đều tương đối minh bạch.

Theo [2],[5] C4.5 cơ chế sinh cây quyết định hiệu quả tối ưu hơn ID3. Các cơ chế xử lý với thiếu dữ liệu, thiếu và chống "quá vừa" dữ liệu của C4.5 cùng với cơ chế cắt tỉa cây đã tạo nên sức mạnh của C4.5, xử lý thuộc tính thực: rời rạc hóa hoặc rẽ nhánh.

Ngoài ra, mô hình phân lớp C4.5 còn có phần chuyển đổi từ cây quyết định sang luật dạng if-then, làm tăng độ chính xác và tính dễ hiểu của kết quả phân lớp. Đây là tiện ích rất có ý nghĩa đối với người sử dụng

Thuật toán phân loại dùng cây quyết định C4.5 được lựa chọn sử dụng trong luận văn. Việc thực hiện thuật toán trong bài toán hỗ trợ ra quyết định hòa giải hoặc xét xử trong các vụ việc hôn nhân sẽ được giới thiệu trong chương 2.

CHƯƠNG 2.
THỬ NGHIỆM HỆ THỐNG TRỢ GIÚP RA QUYẾT ĐỊNH HÒA GIẢI, XÉT XỬ

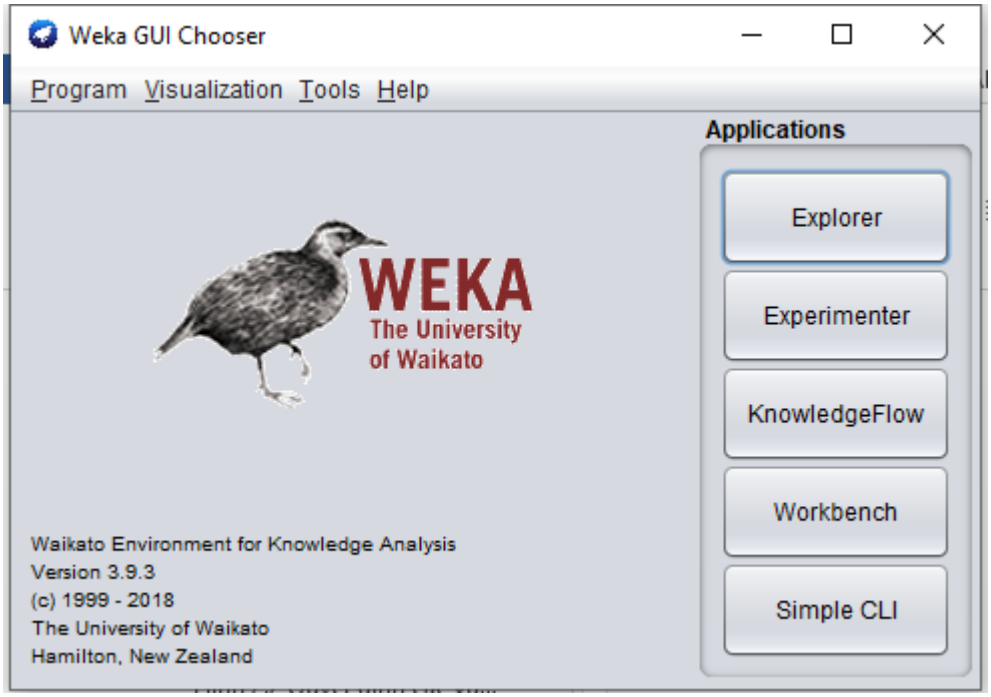
Chương này trình bày việc sử dụng phần mềm Weka để thực hiện thuật toán phân lớp dữ liệu án hôn nhân. Lựa chọn sử dụng thuật toán J48 vì thuật toán J48 được Weka cài đặt trên thuật toán C4.5 theo [6].

2.1. Phần mềm Weka

Weka (viết tắt của Waikato Environment for Knowledge Analysis) là một bộ phần mềm học máy được Đại học Waikato, New Zealand phát triển bằng Java. Weka là phần mềm tự do phát hành theo Giấy phép Công cộng GNU.

Weka chứa một tập hợp các công cụ và thuật toán trực quan để phân tích dữ liệu và mô hình dự đoán, cùng với giao diện người dùng đồ họa để dễ dàng truy cập vào các chức năng này. Phiên bản Weka gốc không phải là Java là thuật toán mô hình hóa Tcl / Tk (chủ yếu là bên thứ ba) được triển khai bằng các ngôn ngữ lập trình khác, cộng với các tiện ích tiền xử lý dữ liệu trong C và hệ thống dựa trên Makefile cho máy chạy học tập thí nghiệm. Phiên bản gốc này được thiết kế chủ yếu như một công cụ để phân tích dữ liệu từ các lĩnh vực nông nghiệp, nhưng gần đây là phiên bản đầy đủ bằng Java (Weka 3), bắt đầu phát triển vào năm 1997, hiện được sử dụng trong nhiều lĩnh vực ứng dụng khác nhau, đặc biệt cho mục đích giáo dục và nghiên cứu. Ưu điểm của Weka bao gồm:

- Sẵn có miễn phí theo Giấy phép Công cộng GNU.
- Tính di động, do nó được thực hiện đầy đủ trong ngôn ngữ lập trình Java và do đó chạy trên hầu hết mọi nền tảng điện toán hiện đại.
- Hỗ trợ tốt các thuật toán học máy (machine learning) và khai phá dữ liệu;
- Một bộ sưu tập toàn diện về kỹ thuật tiền xử lý và mô hình hóa.
- Trực quan hóa, dễ dàng xây dựng các ứng dụng thực nghiệm;
- Dễ sử dụng do giao diện người dùng đồ họa của nó.



Hình 2.1. Giao diện phần mềm Weka

Weka hỗ trợ một số tác vụ khai thác dữ liệu tiêu chuẩn, cụ thể hơn là tiền xử lý dữ liệu, phân cụm, phân loại, hồi quy, trực quan hóa và lựa chọn tính năng. Tất cả các kỹ thuật của Weka được xác định dựa trên giả định rằng dữ liệu có sẵn dưới dạng một tệp phẳng hoặc quan hệ, trong đó mỗi điểm dữ liệu được mô tả bởi một số thuộc tính cố định (thông thường, thuộc tính số hoặc danh nghĩa, nhưng một số loại thuộc tính khác cũng được hỗ trợ). Weka cung cấp quyền truy cập vào cơ sở dữ liệu SQL bằng cách sử dụng Kết nối cơ sở dữ liệu Java và có thể xử lý kết quả được trả về bởi truy vấn cơ sở dữ liệu. Weka cung cấp quyền truy cập vào deeplearning với Deeplearning4j. Nó không có khả năng khai thác dữ liệu đa quan hệ, nhưng có phần mềm riêng để chuyển đổi một tập hợp các bảng cơ sở dữ liệu được liên kết thành một bảng duy nhất phù hợp để xử lý bằng Weka. Một lĩnh vực quan trọng khác hiện không được bao phủ bởi các thuật toán có trong phân phối Weka là mô hình hóa trình tự.

2.2. Chuẩn bị dữ liệu

Dữ liệu đầu vào được cung cấp dưới dạng tệp excel với 265 bản án, quyết định đã có hiệu lực của Tòa án là dữ liệu mẫu của Tòa án nhân dân thành phố Hà Nội về sổ theo

dõi giải quyết các vụ, việc về hôn nhân gia đình. Bao gồm những thuộc tính:

+ Mã vụ việc;

+ Họ và tên (nguyên đơn);

+ Giới tính (nguyên đơn);

+ Năm sinh (nguyên đơn);

+ Tuổi (nguyên đơn);

+ Địa chỉ (nguyên đơn);

+ Con chung;

+ Họ và tên (bị đơn);

+ Giới tính (bị đơn);

+ Năm sinh (bị đơn);

+ Tuổi (bị đơn);

+ Địa chỉ (bị đơn);

+ Tên Thẩm phán;

+ Quyết định.

Đưa tập dữ liệu vào xử lý: Là bảng tính excel như sau:

Bảng thông tin vụ án

Mã vụ việc	Ngày thụ lý	Nguyên đơn (vợ/chồng)				Bị đơn (vợ/chồng)				Thông tin con chung	Quan hệ pháp luật	Thẩm phán	Quyết định
		Họ và tên	Năm sinh	Tuổi	*Địa chỉ*	*Họ và tên*	Năm sinh	Tuổi	*Địa chỉ*				
1	19/05/2016	Phạm Thị Cẩm Dung	1990	26	791/5/47/42 Bến Phú Định,	Phạm Thắng Lợi	1989	27	Phòng 4 C3 Tổ 21 Ngõ 124	có	(4) Do nghiện rượu, cờ bạc, ma túy (TN)	Bùi Thị Thanh Phương	(2) Xét xử (XX)
2	06/05/2016	Nguyễn Thanh Bình	1987	29	P512 C1 Thanh Xuân Bắc,	Nguyễn Duy Pháp	1987	29	P512 C1 Thanh Xuân Bắc,	có	(8) Bạo lực gia đình (BLGĐ)	Bùi Thị Thúc	(2) Xét xử (XX)
3	16/05/2016	Đỗ Thị Nhung	1992	24	Thôn Tân An, xã Thuỵ	Nguyễn Đình Sơn	1992	24	Tổ 20, phường phú Lương,	có	(1) Mâu thuẫn gia đình (MTGĐ)	CHU MINH SANG	(1)Hòa giải (HG)
4	23/05/2016	Lê Thị Uyên	1992	24	Tổ 8 khu giãn dân Mộ Lao, Hà	Nguyễn Tiến Văn	1989	27	Tổ 8 khu giãn dân Mộ Lao,	có	(1) Mâu thuẫn gia đình (MTGĐ)	DIỆP LÊ QUỲNH ANH	(2) Xét xử (XX)
5	16/03/2016	Thân Thị Thảo	1989	27	số 59 ngách 57/88 ngõ 57	Nguyễn Ngọc Tuấn	1984	32	số 59 ngách 57/88 ngõ 57	có	(3) Ngoại tình (NT)	Danh Đời	(1)Hòa giải (HG)
6	10/05/2016	Nguyễn Thị Hảo	1992	24	số 20 ngõ 111 đường Trung	Đào Tiến Long	1989	27	số 20 ngõ 111 đường Trung	có	(1) Mâu thuẫn gia đình (MTGĐ)	Dương Hồng Phương	(1)Hòa giải (HG)
7	20/09/2016	Phan Thị Hồng Nhung	1991	25	TDP Phú Đô, phường Phú Đô,	Hoàng Bá Tâm	1991	25	Nga Sơn, Cẩm Khê, Phú Thọ	có	(4) Do nghiện rượu, cờ bạc, ma túy (TN)	Dương Thị Huyền Trang	(2) Xét xử (XX)
8	02/03/2016	Đỗ Thị Hương	1973	43	xóm Chợ, phường Đại Mỗ,	Trần Đức Quyền	1959	57	Cao Sơn, Tiên Phương,	có	(1) Mâu thuẫn gia đình (MTGĐ)	Giản Thị Dung	(2) Xét xử (XX)
9	15/10/2016	Nguyễn Quỳnh Anh	1985	31	P304 nhà A 2 tập thể Bộ Thương	Phạm Ngọc Anh	1977	39	P304 nhà A 2 Tập thể Bộ	có	(7) Mâu thuẫn kinh tế (MTKT)	HOÀNG THỊ XUÂN	(2) Xét xử (XX)
10	07/06/2016	Phạm Anh Hồng	1981	35	P9K8 ngõ 43 đường Tương	Đặng Minh Quang	1977	39	Số 2 phường Dịch Vọng Hậu	có	(7) Mâu thuẫn kinh tế (MTKT)	HUỲNH THÚY NGÂN	(2) Xét xử (XX)
11	20/06/2016	Trịnh Thị Hải Duyên	1982	34	Tổ 10 Đại Từ	Nguyễn Khắc Tin	1981	35	Tổ 10 Đại Từ	có	(7) Mâu thuẫn kinh tế (MTKT)	Hà Hoa Thiên	(2) Xét xử (XX)
12	19/08/2016	Nguyễn Thị Minh Thuỷ	1989	27	P114 tập thể Thú y phương	Hoàng Quốc Lục	1985	31	Tổ 5C	có	(7) Mâu thuẫn kinh tế (MTKT)	Hà Quỳnh Nga	(1)Hòa giải (HG)
13	15/08/2016	Kim Bảo Giang	1985	31	P506 N21 khu đô thị Pháp	Lê Vũ Long	1978	38	Tổ 2 p. Mai Động	có	(1) Mâu thuẫn gia đình (MTGĐ)	HỒ NGỌC TRƯỜNG	(2) Xét xử (XX)
14	12/07/2016	Hoàng Thị Xuân Thu	1978	38	1AQ6 trương Định , Nguyễn	Phạm Hoàng Anh	1976	40	1A Q6 Trương Định , Nguyễn	có	(3) Ngoại tình (NT)	HỒ THỊ KIM OANH	(1)Hòa giải (HG)
15	28/07/2016	Trương Thị Thu Hiền	1982	34	Số 2 tổ 17 ngách 351/108/16	Trần Đình Hải	1977	39	Số 2 tổ 17 ngách 351/108/16	có	(1) Mâu thuẫn gia đình (MTGĐ)	HỒ THỊ KIỀU TRANG	(1)Hòa giải (HG)
16	12/08/2016	Tạ Phương Thảo	1989	27	P609N10 Khu Đồng Tàu	Đỗ Văn Lợi	1985	31	P609N10 Khu Đồng Tàu	có	(3) Ngoại tình (NT)	Hồ Quỳnh Trâm	(1)Hòa giải (HG)
17	10/05/2016	Hà Thị Hà	1993	23	Tổ 22 phường Hoàng Văn	Lê Xuân Đạt	1982	34	Tổ 22 phường Hoàng Văn	Không	(5) Bệnh tật, không có con (BT)	Hồ Thị Thu Thảo	(1)Hòa giải (HG)
18	31/08/2016	Nguyễn Thị Lý	1978	38	Số 89 ngõ 259 tổ 49	Dương Văn Việt	1964	52	Số 89 ngõ 259 tổ 49	có	(1) Mâu thuẫn gia đình (MTGĐ)	Hồ Thị Yến Khanh	(1)Hòa giải (HG)
19	29/10/2016	Lê Thị Tình	1975	41	P712 N3 Bán Đảo Linh Đàm	Nguyễn Văn Thành	1974	42	P712 N3 Bán Đảo Linh Đàm	có	(1) Mâu thuẫn gia đình (MTGĐ)	LÊ KHÁNH LINH	(2) Xét xử (XX)
20	03/11/2016	Phạm Thị Hà	1981	35	411 C10 Tân Mai	Nguyễn Trung Kiên	1979	37	411 C10 Tân Mai	có	(4) Do nghiện rượu, cờ bạc, ma túy (TN)	LÊ NGỌC TÚ QUYÊN	(2) Xét xử (XX)
21	28/09/2016	Mạnh Thị Lê Chinh	1976	40	164 Trương Định, P Trương	Đặng Hữu Nhuận	1968	48	số 9/69/255 Linh Nam	có	(1) Mâu thuẫn gia đình (MTGĐ)	LÊ THỊ HƯƠNG	(1)Hòa giải (HG)
22	08/10/2016	Hoàng Thị Hải Hà	1992	24	A9 tập thể xí nghiệp cung ứng	Nguyễn Công Hà	1988	28	A9 tập thể xí nghiệp cung ứng	có	(4) Do nghiện rượu, cờ bạc, ma túy (TN)	LÊ THỊ THANH VÂN	(1)Hòa giải (HG)
23	20/10/2016	Nguyễn Thị Chinh	1989	27	Số 16B tập thể Thổ Nhưỡng	Dương Quốc Việt	1985	31	P101B C29	có	(4) Do nghiện rượu, cờ bạc, ma túy (TN)	LÊ VĂN TUẤN	(2) Xét xử (XX)

Hình 2.2 Dữ liệu sổ theo dõi các vụ việc hôn nhân gia đình

(Nguồn: Tòa án nhân dân thành phố Hà Nội)

Trong qui trình khai phá dữ liệu, công việc xử lý dữ liệu trước khi đưa vào các mô hình là rất cần thiết, bước này làm cho dữ liệu có được ban đầu qua thu thập dữ liệu (gọi là dữ liệu gốc ordinal data) có thể áp dụng được (thích hợp) với các mô hình khai phá dữ liệu (data mining model) cụ thể. Các công việc cụ thể của tiền xử lý dữ liệu bao gồm những công việc như:

- Filtering Attributes: Chọn các thuộc tính phù hợp với mô hình;
- Filtering samples: Lọc các mẫu (instances, patterns) dữ liệu cho mô hình;
- Clean data: Làm sạch dữ liệu như xóa bỏ các dữ liệu bất thường (Outlier);
- Transformation: Chuyển đổi dữ liệu cho phù hợp với các mô hình như chuyển đổi dữ liệu từ numeric qua nomial hay ordinal;
- Discretization (rời rạc hóa dữ liệu): Nếu bạn có dữ liệu liên tục nhưng một vài mô hình chỉ áp dụng cho các dữ liệu rời rạc (như luật kết hợp chẳn hạn) thì bạn phải thực hiện việc rời rạc hóa dữ liệu.

Để thực hiện mô hình khai phá luật kết hợp ta cần hiệu chỉnh lại dữ liệu và loại bỏ các thuộc tính không cần thiết:

i. Loại bỏ các thuộc tính mà dữ liệu bị thiếu hoặc bị nhiễu quá nhiều;

ii. Loại bỏ thuộc tính "Họ tên", "Giới tính", "Địa chỉ" vì các thuộc tính này không dùng trong mô hình hay không phải là các thuộc tính đăc trưng, gọi là lọc thuộc tính;

iii. Biến số hóa: Đây là bước đưa các thuộc tính của bài toán thành các biến số để thuận tiện cho việc xử lý.

Hiện nay công tác thống kê tổng hợp tại Tòa án thực hiện hàng tháng với một số chỉ tiêu thống kê liên quan đến các công tác giải quyết các vụ, việc về hôn nhân gian đình, bao gồm những thông tin sau:

Thông tin về tuổi của nguyên đơn, năm sinh của nguyên đơn, năm sinh của bị đơn, có con chung, độ chênh lệch tuổi giữa nguyên đơn và bị đơn, quan hệ pháp luật và kết quả giải quyết. Những thống kê này chỉ dừng mức cung cấp số liệu.

Qua quá trình khảo sát tại thực tế tại Tòa án nhân dân thành phố Hà Nội và tham khảo ý kiến của các Thẩm phán, cán bộ hành chính tư pháp thì các thuộc tính

trên là các thuộc tính có độ tin tưởng nhất, đây cũng là các thuộc tính được sử dụng trong công tác báo cáo thống kê của Tòa án.

Dựa trên kết quả thực tế này ta chọn lọc và thực hiện phân lớp giải quyết các vụ, việc về hôn nhân gia đình trên các thuộc tính đó.

Ký hiệu các biến số như sau:

- Độ tuổi nguyên đơn : *tuoi_nguyen_don*;
- Độ tuổi bị đơn : *tuoi_bi_don*;
- Con chung: *co_con*;
- Độ lệch tuổi của nguyên đơn và bị đơn: *do_lech_tuoi*;
- Quan hệ pháp luật: *quan_he_phap_luat*;
- Quyết định của thẩm phán: *quyet_dinh*;

Điều chỉnh giá trị của các tham số về chung một dạng. Do ban đầu giá trị của các tham số có hai kiểu: Một kiểu là biểu thức logic như "Tuổi nhỏ hơn hoặc bằng 30","Tuổi lớn hơn 30" và kiểu còn lại là văn bản như "Mâu thuẫn", "Bạo lực",… Ta sẽ đưa các giá trị này về chung một kiểu giá trị là số tự nhiên.

iv. Rời rạc hóa dữ liệu.

Tất cả các giá trị cụ thể và gán nhãn được thể hiện trong bảng sau:

Bảng 2.1. Biến số hóa dữ liệu "độ tuổi"

STT	Giá trị thuộc tính	Gán nhãn
1	Độ tuổi của nguyên đơn nhỏ hơn hoặc bằng 30.	tuoi_nguyen_don: <=30
2	Độ tuổi của nguyên đơn lớn hơn 30.	tuoi_nguyen_don: >30
3	Độ tuổi của bị đơn nhỏ hơn hoặc bằng 30.	tuoi_bi_don: <=30
4	Độ tuổi của bị đơn lớn hơn 30.	tuoi_bi_don: >30

Bảng 2.2. Biến số hóa dữ liệu "con chung"

STT	Giá trị thuộc tính	Gán nhãn
1	Có con chung	co_chung: co
2	Không có con chung	co_chung: khong

Bảng 2.3. Biến số hóa dữ liệu "độ lệch tuổi"

STT	Giá trị thuộc tính	Gán nhãn
1	Chênh lệch tuổi của nguyên đơn và bị đơn trong khoảng nhỏ hơn hoặc bằng 5.	do_lech_tuoi: <=5
2	Chênh lệch tuổi của nguyên đơn và bị đơn trong khoảng lớn hơn hoặc 5 và nhỏ hoặc bằng 10	do_lech_tuoi: 5_10
3	Chênh lệch tuổi của nguyên đơn và bị đơn lớn hơn 10.	do_lech_tuoi: >10

Bảng 2.4. Biến số hóa dữ liệu "quan hệ pháp luật"

STT	Giá trị thuộc tính	Gán nhãn
1	Mâu thuẫn gia đình.	*quan_he_phap_luat: MTGD*
2	Yếu tố nước ngoài	*quan_he_phap_luat: NN*
3	Ngoại tình	*quan_he_phap_luat: NT*
4	Cờ bạc, rượu chè, ma túy	*quan_he_phap_luat: TNXH*
5	Bệnh tật, không có con	*quan_he_phap_luat: BT*
6	Một người mất tích	*quan_he_phap_luat: MT*
7	Mâu thuẫn kinh tế	*quan_he_phap_luat: MTKT*
8	Bạo lực gia đình	*quan_he_phap_luat: BLGD*

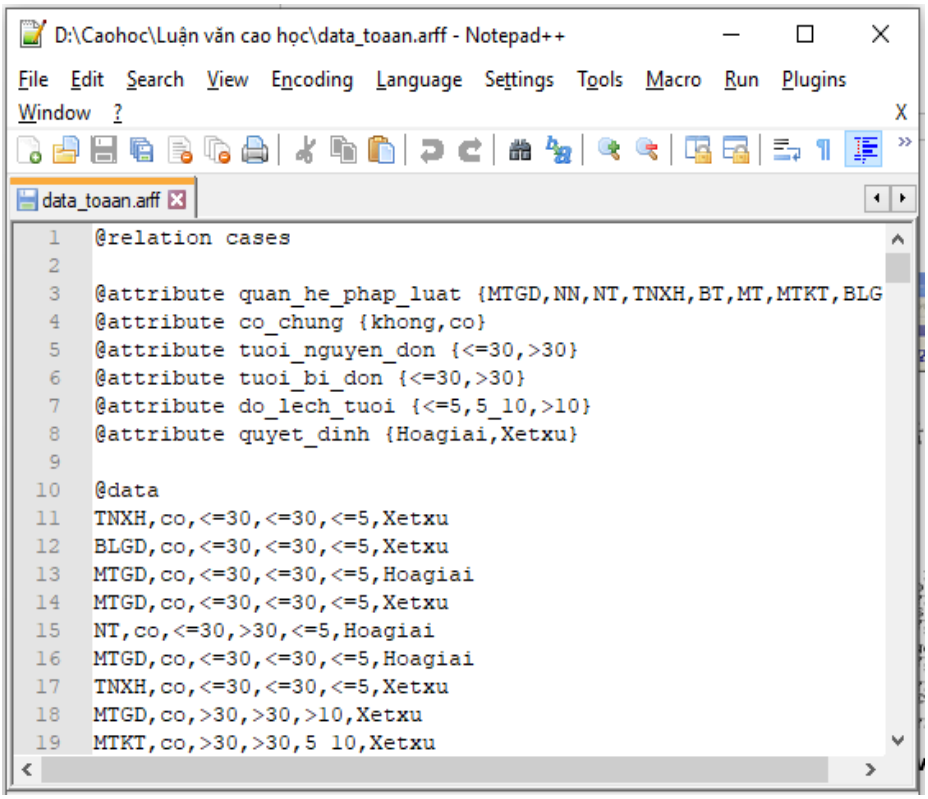
Bảng 2.5. Ý nghĩa biến "quyết định"

STT	Giá trị thuộc tính	Gán nhãn
1	Hòa giải	quyet_dinh: Hoagiai
2	Xét xử	quyet_dinh: Xetxu

Quyết định của thẩm phán được thể hiện ở hai lớp được gán nhãn là Hoagiai và Xetxu như trong bảng trên.

Tức là nếu đầu ra của *quyet_dinh* là *Hoagiai* thì quyết định của thẩm phán là đưa vụ việc ra hòa giải, là Xetxu thì Thẩm phán quyết định đưa vụ việc ra mở phiên tòa xét xử.

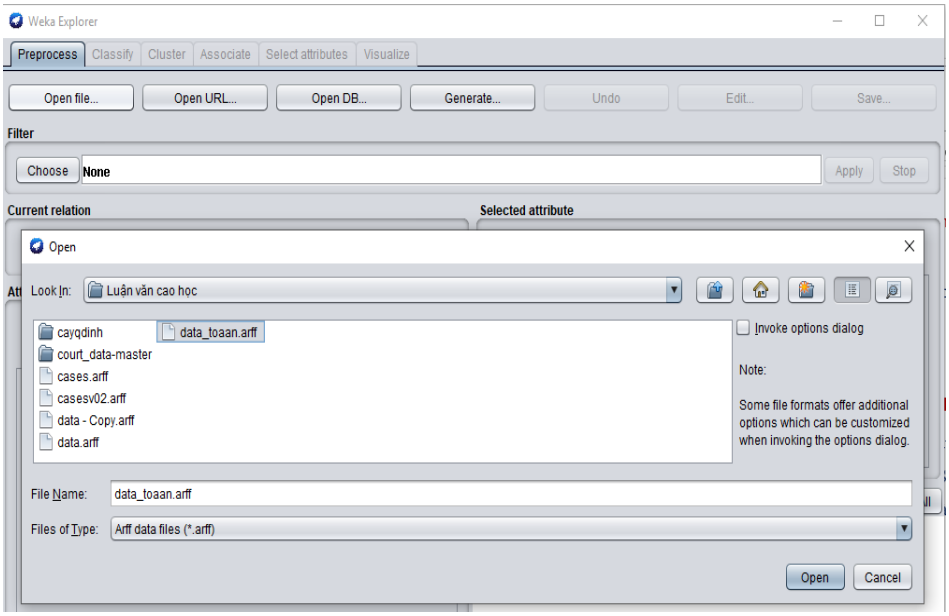
Sau khi được chuẩn hóa dựa trên các bảng mô tả trên có được dữ liệu đầu ra là tệp data_toaan.arff:



Hình 2.3 Dữ liệu sau chuẩn hóa

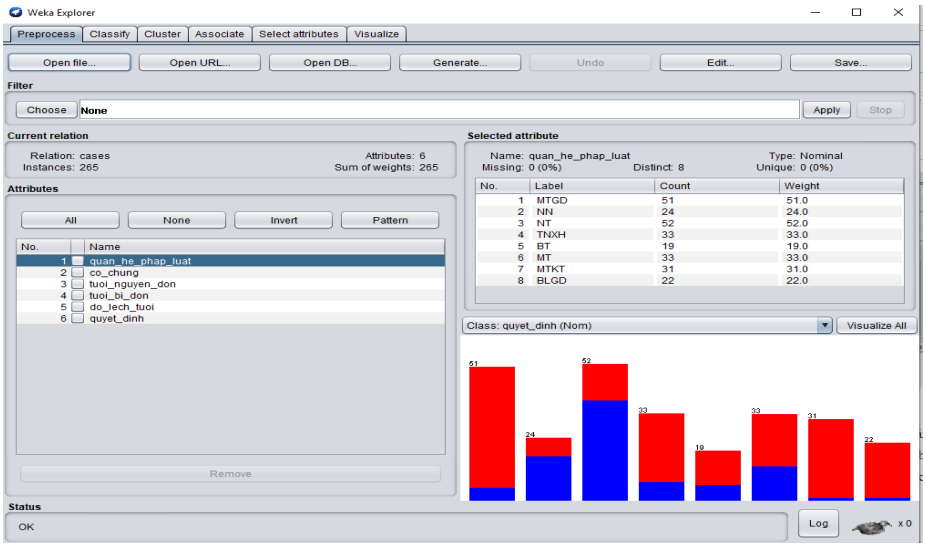
2.3. Thử nghiệm chương trình Weka với thuật toán J48

Khi khởi động, chương trình sẽ thực hiện đọc tệp mô tả cấu trúc của dữ liệu Toà án đã được chuẩn hóa tệp data_toaan.arff.



Hình 2.4. Chọn tệp dữ liệu data_toaan.arff

Trong các attribute (thuộc tính) thì ta sẽ sử dụng cây quyết định để dự đoán thuộc tính cuối cùng là quyết định của toà án. Sau khi đọc tệp data_toaan.arff thì chương trình phân tích số liệu và được trực quan hóa như sau:



Hình 2.5. Trực quan hóa dữ liệu data_toaan.arff

Phần mềm Weka hỗ trợ nhiều thuật toán phân lớp như ID3, J48, C4.5, CART, SVM… Việc chọn thuật toán nào để có hiệu quả phân lớp cao tuy thuộc vào rất nhiều yếu tố, trong đó cấu trúc dữ liệu ảnh hưởng rất lớn đến kết quả của các thuật toán.

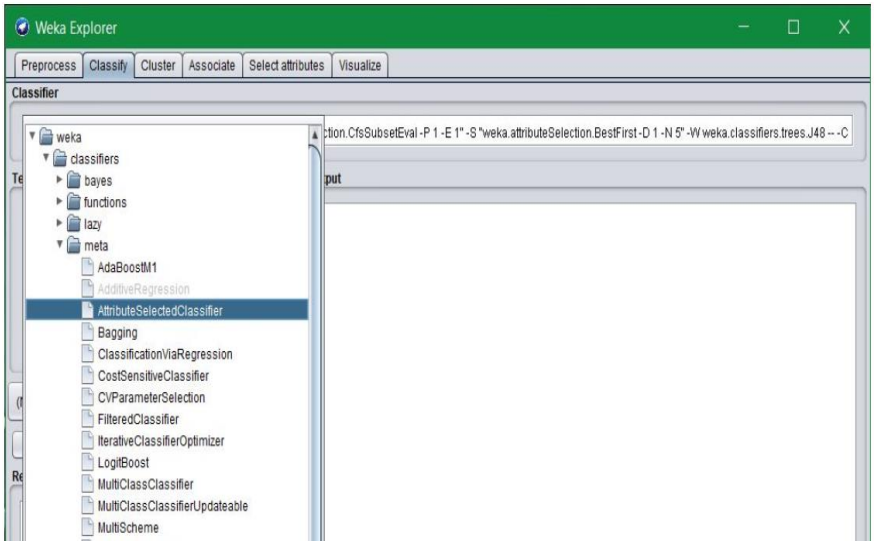
Với thuật toán ID3 và CART cho hiệu quả phân lớp rất cao đối với các trường dữ liệu số (quantitative value) trong khi đó các thuật toán như J48, C4.5 cho hiệu quả hơn đối với các dữ liệu Nominal. Từ kết quả đã chuẩn hóa dữ liệu từ tệp excel thì được tệp dữ liệu data_toaan.arff có kiểu Nominal, vì vậy ta sử dụng thuật toán J48 để đạt hiệu quả phân lớp cao.

a. Lựa chọn chỉ số ***Information Gain*** các bước thực hiện:

i. Chọn tab ***Classify;***

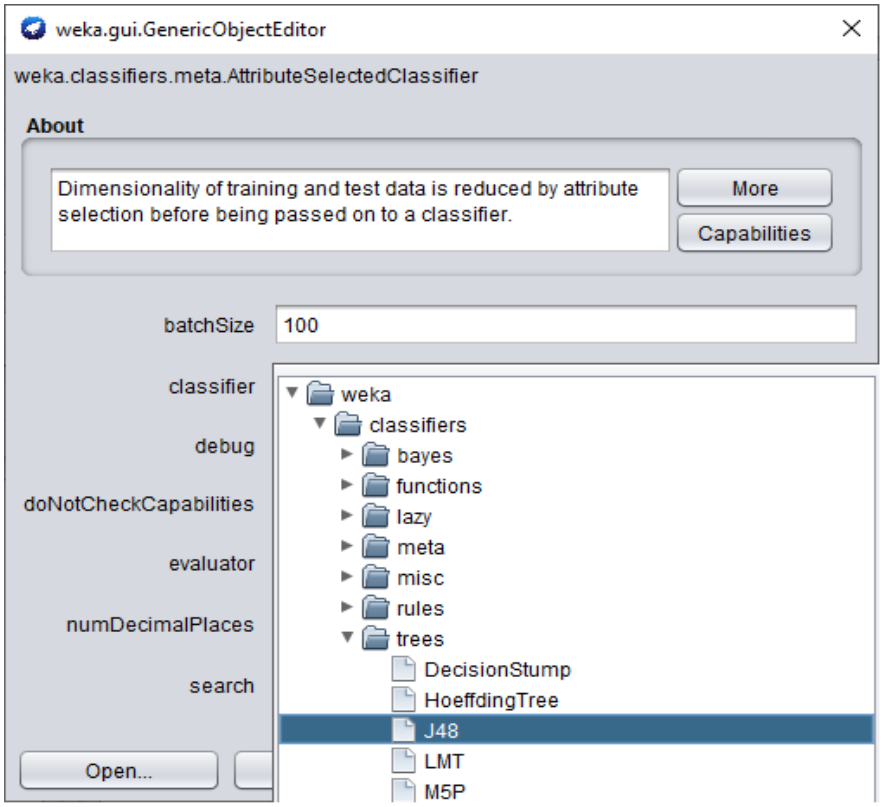
ii. Chọn ***meta***

iii. Chọn ***AttributeSelectedClassifier*** như hình vẽ



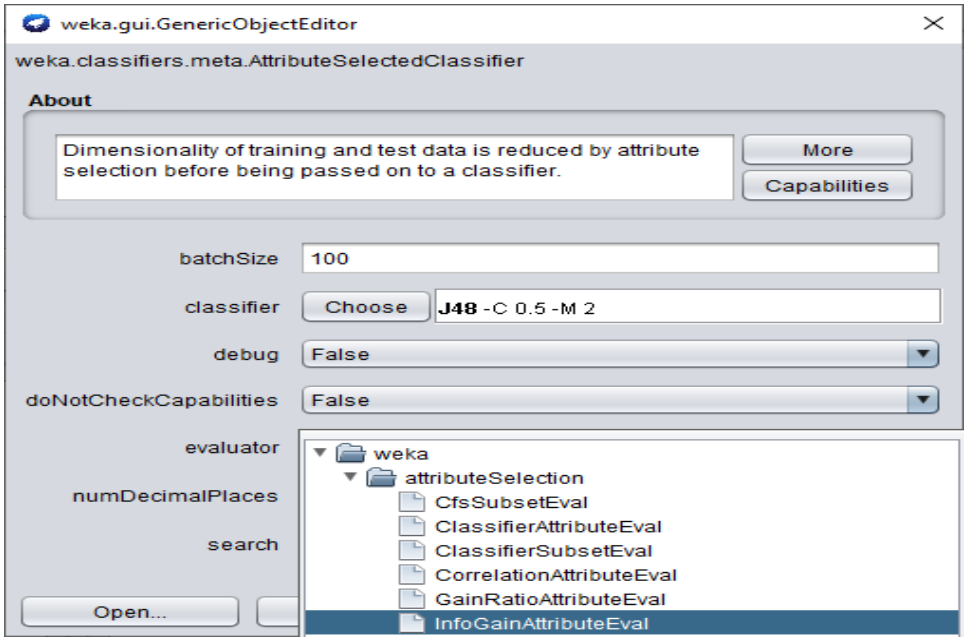
Hình 2.6. Chọn thuộc tính AttributeSelectedClassifier

iv. Chọn thuật toán ***j48***



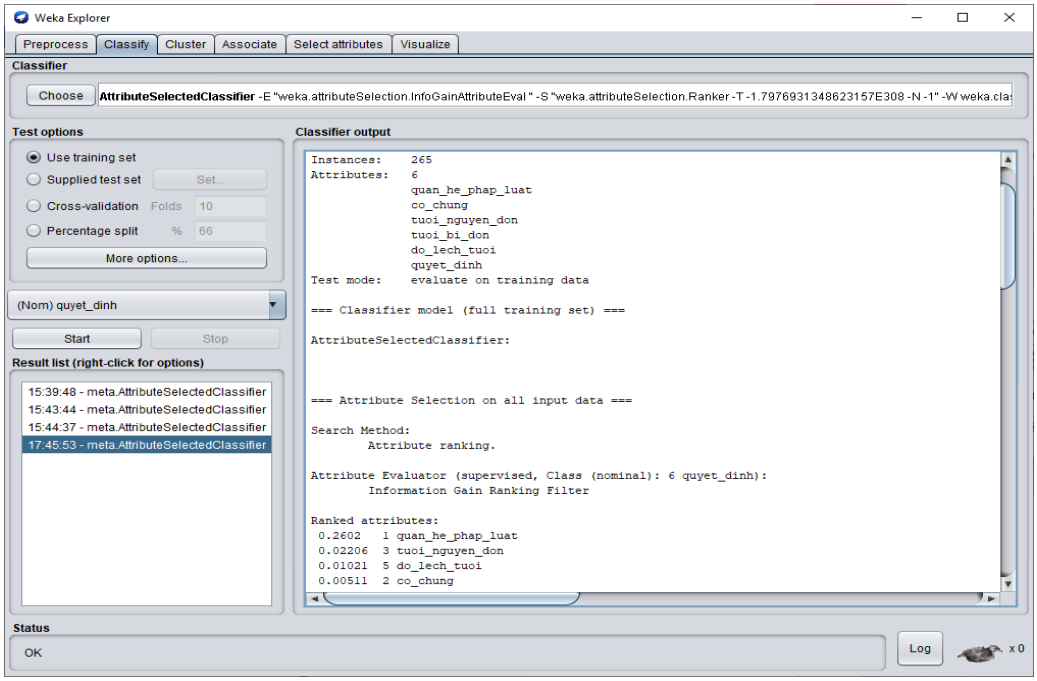
Hình 2.7. Chọn thuật toán j48

v. Lựa chọn thuộc tính ***Information Gain***



Hình 2.8. Chọn Information Gain

Sau khi chạy thuật toán này với các lựa chọn trên ta thu được kết quả như sau:



Hình 2.9. Kết quả thực hiện với lựa chọn Information Gain

Đánh giá kết quả phân lớp với mẫu dữ liệu số lá của cây là 14, kích thước là của cây 20 và thời gian thực hiện 0.01s.

Các chỉ số phân lớp với lựa chọn Information Gain:

- Độ phân lớp chính xác của dữ liệu: Độ phân lớp đạt kết quả chính xác rất cao của thuật toán với 221/265 tương đương 83.3962 %;
- Độ phân lớp không chính xác của dữ liệu: Độ phân lớp đạt kết quả không chính xác của thuật toán 44/265 tương đương 16.6038 %.

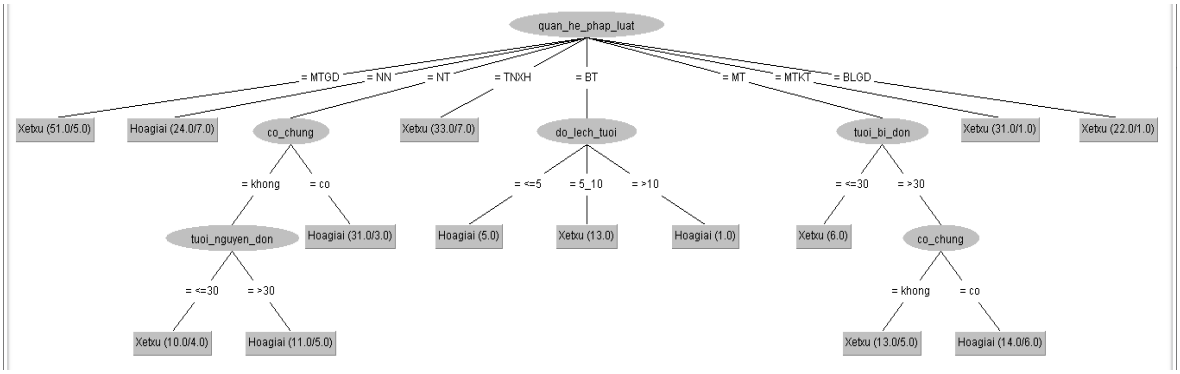
Đánh giá chỉ số Information Gain cho từng thuộc tính:

Bảng 2.6. Bảng xếp hạng chỉ số Information Gain

Xếp hạng	Thuộc tính đặc trưng	Chỉ số Information Gain
1	quan_he_phap_luat	0.26020
2	tuoi_nguyen_don	0.02206
3	do_lech_tuoi	0.01021
4	co_chung	0.00511
5	tuoi_bi_don	0.00137

Dựa trên việc xếp hạng các đặc trưng dựa trên chỉ số Information Gain tương ứng, ta có thể biết được mức độ thông tin chứa trong mỗi đặc trưng và mức quan trọng của đặc trưng đó, hay tầm ảnh hưởng của đặc trưng đó lên kết quả quyết định của thẩm phán. Ở đây thuộc tính quan_he_phap_luat được chọn làm gốc để phát triển cây quyết định.

Mô hình cây



Hình 2.10. Cây quyết định với lựa chọn Information Gain

b. Lựa chọn chỉ số Gain Ratio các bước thực hiện:

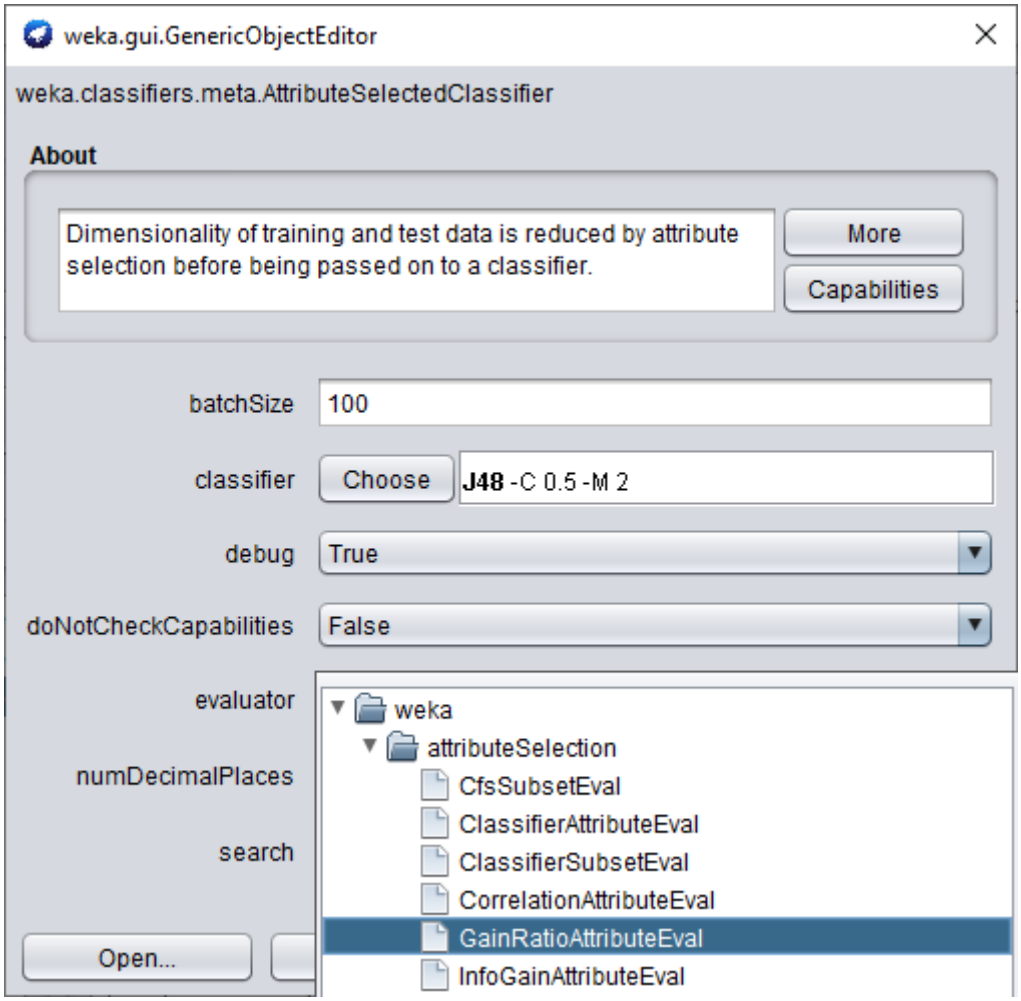
i. Chọn tab ***Classify;***

ii. Chọn ***meta***

iii. Chọn ***AttributeSelectedClassifier***

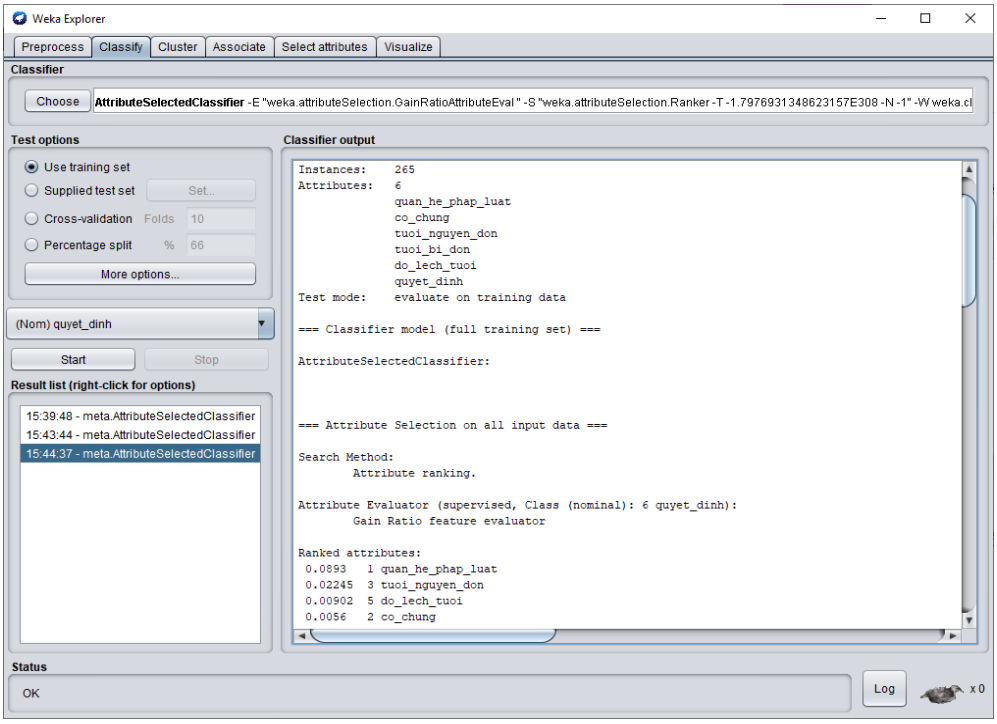
iv. Chọn thuật toán ***j48***

v. Lựa chọn thuộc tính ***Gain Ratio***



Hình 2.11. Chọn Gain Ratio

Sau khi chạy thuật toán này với các lựa chọn trên ta thu được kết quả như sau:



Hình 2.12. Kết quả thực hiện với lựa chọn Gain Ratio

Đánh giá kết quả phân lớp với mẫu dữ liệu số lá của cây là 14, kích thước là của cây 20 và thời gian thực hiện 0.01s.

Các chỉ số phân lớp với lựa chọn Gain Ratio:

- Độ phân lớp chính xác của dữ liệu: Độ phân lớp đạt kết quả chính xác rất cao của thuật toán với 221/265 tương đương 83.3962 %;
- Độ phân lớp không chính xác của dữ liệu: Độ phân lớp đạt kết quả không chính xác của thuật toán 44/265 tương đương 16.6038 %.

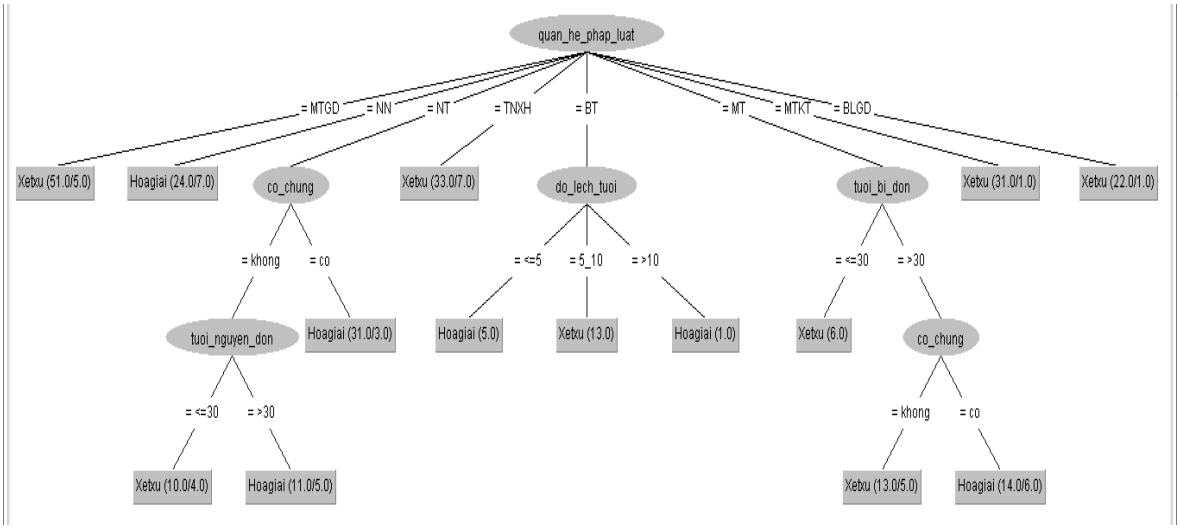
Đánh giá chỉ số Gain Ratio cho từng thuộc tính:

Bảng 2.7. Bảng xếp hạng chỉ số Gain Ratio

Xếp hạng	Thuộc tính đặc trưng	Chỉ số Information Gain
1	quan_he_phap_luat	0.0893
2	tuoi_nguyen_don	0.02245
3	do_lech_tuoi	0.00902
4	co_chung	0.0056
5	tuoi_bi_don	0.00164

Dựa trên việc xếp hạng các đặc trưng dựa trên chỉ số Gain Ratio tương ứng, ta có thể biết được mức độ thông tin chứa trong mỗi đặc trưng và mức quan trọng của đặc trưng đó, hay tầm ảnh hưởng của đặc trưng đó lên kết quả quyết định của thẩm phán. Ở đây thuộc tính quan_he_phap_luat được chọn làm gốc để phát triển cây quyết định. Tuy nhiên so với lựa chọn Information Gain thì thuộc tính quan_he_phap_luat với lựa chọn Gain Ratio có giá trị nhỏ hơn.

Mô hình cây



Hình 2.13. Cây quyết định với lựa chọn Gain Ratio

Cả hai lựa chọn trên đều cho sinh ra cùng một cây và bộ luật như nhau:

quan_he_phap_luat = MTGD: Xetxu (51.0/5.0)

quan_he_phap_luat = NN: Hoagiai (24.0/7.0)

quan_he_phap_luat = NT

| co_chung = khong

| | tuoi_nguyen_don = <=30: Xetxu (10.0/4.0)

| | tuoi_nguyen_don = >30: Hoagiai (11.0/5.0)

| co_chung = co: Hoagiai (31.0/3.0)

quan_he_phap_luat = TNXH: Xetxu (33.0/7.0)

quan_he_phap_luat = BT

| do_lech_tuoi = <=5: Hoagiai (5.0)

| do_lech_tuoi = 5_10: Xetxu (13.0)

| do_lech_tuoi = >10: Hoagiai (1.0)

quan_he_phap_luat = MT

| tuoi_bi_don = <=30: Xetxu (6.0)

| tuoi_bi_don = >30

| | co_chung = khong: Xetxu (13.0/5.0)

| | co_chung = co: Hoagiai (14.0/6.0)

quan_he_phap_luat = MTKT: Xetxu (31.0/1.0)

quan_he_phap_luat = BLGD: Xetxu (22.0/1.0)

Từ cây quyết định trên ta có bộ luật được chuẩn hóa dưới dạng if-then:

1. Rule 1: IF quan_he_phap_luat = "MTGD" THEN quyet_dinh="Xetxu";

2. Rule 2: IF quan_he_phap_luat = "NN" THEN quyet_dinh="Hoagiai";

3. Rule 3: IF quan_he_phap_luat = "NT" AND co_chung = "khong" AND tuoi_nguyen_don="<=30" THEN quyet_dinh="Xetxu";

4. Rule 4: IF quan_he_phap_luat = "NT" AND co_chung = "khong" AND tuoi_nguyen_don=">30" THEN quyet_dinh="Hoagiai";

5. Rule 5: IF quan_he_phap_luat = "NT" AND co_chung = "co" THEN quyet_dinh="Hoagiai";

6. Rule 6: IF quan_he_phap_luat = "TNXH" THEN quyet_dinh="Xetxu";

7. Rule 7: IF quan_he_phap_luat = "BT" AND do_lech_tuoi="<=5" THEN quyet_dinh="Hoagiai";

8. Rule 8: IF quan_he_phap_luat = "BT" AND do_lech_tuoi="5_10" THEN quyet_dinh="Xetxu";

9. Rule 9: IF quan_he_phap_luat = "BT" AND do_lech_tuoi=">10" THEN quyet_dinh="Hoagiai";

10. Rule 10: IF quan_he_phap_luat = "MT" AND tuoi_bi_don="<=30" THEN quyet_dinh="Xetxu";

11. Rule 11: IF quan_he_phap_luat = "MT" AND tuoi_bi_don=">30" AND co_chung="khong" THEN quyet_dinh="Xetxu";

12. Rule 12: IF quan_he_phap_luat = "MT" AND tuoi_bi_don=">30" AND

co_chung="co" THEN quyet_dinh="Hoagiai";

13. Rule 13: IF quan_he_phap_luat = "MTKT" THEN quyet_dinh="Xetxu";

14. Rule 14: IF quan_he_phap_luat = "BLGD" THEN quyet_dinh="Xetxu";

2.4. Kết luận

Weka giúp tạo ra cây quyết định và bộ luật từ tệp data_toaan.arff một cách dễ dàng và trực quan rất hiệu quả.

Khi áp dụng thư viện J48, ta lựa chọn phương thức định lượng (evaluator) là Information Gain, nghĩa là chúng ta muốn đánh giá mức ảnh hưởng của từng đặc trưng lên quyết định thông qua mức Information Gain của đặc trưng đó. Thêm vào đó, ta sử dụng phương pháp xếp hạng (Ranker) để tìm ra các đặc trưng có mức Information Gain cao nhất để tham gia vào xây dựng cây quyết định. Số lượng các đặc trưng được lựa chọn sau khi xếp hạng phụ thuộc vào ngưỡng được xác định sau khi ta lựa chọn các thông số cho quá tình cắt tỉa cây quyết định.

Từ dữ liệu đầu vào đánh giá độ hiệu quả phân lớp dữ liệu là rất cao với độ chính xác là 83.3962 %, không chính xác của thuật toán chỉ 16.6038 %.

Với việc có thể tự thay đổi quyết định và thêm quyết định thực tế vào cơ sở dữ liệu sẵn có, phần mềm sẽ không ngừng được phát triển và hoàn thiện nhằm đưa ra những quyết định chính xác, sát với thực tế hơn.

Chương trên đã thể hiện việc phân lớp các án hôn nhân với phần mềm Weka.

Với bộ luật được sinh là cơ sở để thực hiện xây dựng phần mềm hệ thống hỗ trợ ra quyết định về hòa giải, đối thoại được trình bày tại chương 3 của luận văn.

CHƯƠNG 3.
XÂY DỰNG HỆ THỐNG HỖ TRỢ RA QUYẾT ĐỊNH VỀ CÁC TRANH CHẤP HÔN NHÂN VÀ GIA ĐÌNH

Chương này trình bày hiện trạng xử lí dữ liệu xử án hôn nhân và phân tích, thiết kế, xây dựng cơ sở dữ liệu trên SQL SERVER, nhằm trợ giúp quyết định xử án hôn nhân.

3.1. Nhu cầu về cơ sở dữ liệu các bản án hôn nhân gia đình

3.1.1. Nhu cầu về xây dựng cơ sở dữ liệu về các bản án, quyết định của Tòa án

Cùng với sự phát triển của kinh tế xã hội trong những năm gần đây thể hiện qua các chỉ số và các báo cáo tích cực tại các kỳ họp Quốc hội thì bên cạnh đó cùng tồn tại những thách thức về các vấn đề chính sách an sinh xã hội, tranh chấp xã hội, các vụ việc liên quan đến lĩnh vực hình sự, dân sự, kinh doanh thương mại, lao động, hành chính, trẻ vị thành niên và hôn nhân gia đình ngày càng tăng, đó cũng là thách thức không nhỏ với ngành Tòa án nhân dân.

Cụ thể các vụ việc thụ lý hàng năm tăng cao trong khi nhân sự cán bộ Thẩm phán, Thư ký, Hành chính tư pháp,…không tăng. Được thể hiện qua các con số sau: Từ ngày 01/10/2018 đến ngày 30/9/2019, các Tòa án đã thụ lý 625.979 vụ việc[10], đã giải quyết được 500.361 vụ việc (đạt tỷ lệ 80%); so với năm 2018, số vụ việc đã thụ lý tăng 69.141 vụ (bằng 12,4%), đã giải quyết tăng 58.808 vụ (bằng13,3%).

Với nhân sự cán bộ Tòa án hiện nay số lượng Thẩm phán hiện nay vào khoảng 4000[12] người như vậy trung bình hàng năm mỗi Thẩm phán phải thụ lý và xem xét giải quyết khoảng 150 vụ, việc. Với một số các Tòa án như Tòa án nhân dân thành phố Hà Nội, thành phố Hồ Chí Minh, thành phố Hải Phòng là những đơn vị thụ lý cao nhất cả nước thì con số vụ, việc với cần xem xét thụ lý là rất cao. Đây là một áp lực lớn với cán bộ ngành Tòa án nói chung và với các Thẩm phán nói riêng đảm bảo công tác giải quyết các vụ việc phục vụ người dân được hiệu quả, đúng tiến độ. Đặc biệt với các vụ việc về hôn nhân gia đình chiếm tỷ lệ hơn 53%[11] trên tổng các loại vụ việc (Hình sự, Dân sự, Hôn nhân và gia đình, Kinh

doanh thương mại, Hành chính, Lao động).

Qua khảo sát khó khăn thực tế là khi tiếp nhận các đơn khởi kiện cụ thể với lĩnh vực hôn nhân gia đình Thẩm phán được phân công xem xét cần nghiên cứu kỹ nội dung đơn và những thông tin nguyên đơn, bị đơn để quyết định xem vụ việc này có thể hòa giải hay khó hòa giải mà phải mở phiên tòa xét xử. Việc quyết định này cần nhanh chóng không ảnh hưởng đến quyền lợi của người dân, tuy nhiên với lượng vụ việc ngày càng lớn thì đây là áp lực không nhỏ.

Do vậy vấn đề đặt ra cần phải có giải pháp hỗ trợ nghiệp vụ cho công tác xem xét thụ lý vụ việc là hướng đi rất cần thiết. Vậy hướng đi đó là gì?

Trong những năm gần đây công nghệ thông tin phát triển rất mạnh, đặc biệt về các công nghệ 4.0 như số lĩnh vực dữ liệu lớn, trí tuệ nhân tạo,…

Với ngành Tòa án có thể hướng tới tương lai phát triển Tòa án thông minh dựa trên việc khai phá kho dữ liệu về các bản án, quyết định của Tòa án hỗ trợ công tác xem xét giải quyết các vụ việc trong đó có vụ việc về hôn nhân gia đình.

Cùng với việc đẩy mạnh cải cách hành chính và hướng tới giải quyết các vụ việc được nhanh chóng, cần xây dựng một cơ sở dữ liệu và phần mềm áp dụng khai khá dữ liệu về các bản án hỗ trợ ra các quyết định nhanh chóng kịp thời.

3.1.2. Thủ tục giải quyết ly hôn tại Tòa án

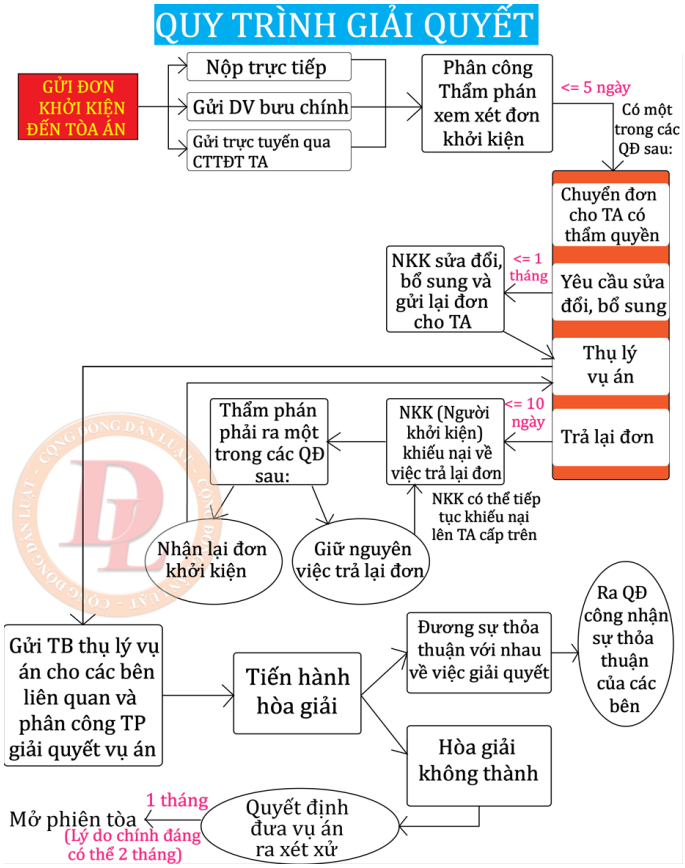
Thủ tục giải quyết ly hôn tại tòa án được quy định cụ thể về quyền của vợ/chồng trong việc yêu cầu tòa án giải quyết ly hôn và các nội dung khác liên quan như sau:

Quyền ly hôn và căn cứ cho ly hôn : Về nguyên tắc vợ, chồng hoặc cả hai người có quyền yêu cầu Toà án giải quyết việc ly hôn. Tuy nhiên trường hợp vợ đang mang thai hoặc đang nuôi con dưới mười hai tháng tuổi thì người chồng không có quyền yêu cầu xin ly hôn. Tòa án xem xét, quyết định cho ly hôn khi đáp ứng đủ các điều kiện sau đây:

1. Tình trạng của vợ chồng trầm trọng;
2. Đời sống chung không thể kéo dài;
3. Mục đích của hôn nhân không đạt.

Thủ tục thuận tình ly hôn (hai vợ chồng đồng thuận ly hôn) :

- Điều kiện tiến hành thủ tục thuận tình ly hôn (i) Hai bên thật sự tự nguyện ly hôn; (ii) Hai bên đã thoả thuận được với nhau về việc chia hoặc không chia tài sản, việc trông nom, nuôi dưỡng, chăm sóc, giáo dục con; (iii) Sự thoả thuận của hai bên về tài sản và con trong từng trường hợp cụ thể này là bảo đảm quyền lợi chính đáng của vợ và con.
- Thủ tục thuận tình ly hôn (i) Đầu tiên, người khởi kiện nộp hồ sơ khởi kiện về việc xin ly hôn tại TAND quận/huyện nơi cư trú, làm việc của vợ hoặc chồng; (ii) Bước 2 : Sau khi nhận đơn khởi kiện cùng hồ sơ hợp lệ Tòa án trong thời hạn 05 ngày làm việc Tòa án kiểm tra đơn và ra thông báo nộp tiền tạm ứng án phí cho người khởi kiện; (iii) Bước 3: người khởi kiện nộp tiền tạm ứng án phí dân sự sơ thẩm tại Chi cục thi hành án quận/huyện và nộp lại biên lai tiền tạm ứng án phí cho Tòa án; (iv) Bước 4: Trong thời hạn 15 ngày làm việc Tòa án tiến hành mở phiên hòa giải; (v) Bước 5: Trong thời hạn 07 ngày kể từ ngày hòa giải không thành (không thay đổi quyết định về việc ly hôn) nếu các bên không thay đổi ý kiến Tòa án ra quyết định công nhận thuận tình ly hôn.



Hình 3.1. Trình tự giải quyết

(Nguồn: https://danluat.thuvienphapluat.vn)

- Thủ tục đơn phương ly hôn (ly hôn theo yêu cầu một bên) như sau: Trình tự xin ly hôn (i) Bước 1: người khởi kiện nộp hồ sơ khởi kiện về việc xin ly hôn tại TAND quận/huyện nơi bị đơn (chồng hoặc vợ) đang cư trú, làm việc; (ii) Bước 2: Sau khi nhận đơn khởi kiện cùng hồ sơ hợp lệ Tòa án sẽ ra thông báo nộp tiền tạm ứng án phí cho người khởi kiện; (iii) Bước 3: người khởi kiện nộp tiền tạm ứng án phí dân sự sơ thẩm tại Chi cục thi hành án quận/huyện và nộp lại biên lai tiền tạm ứng án phí cho Tòa án; (iv) Bước 4: Tòa án thụ lý vụ án, tiến hành giải quyết vụ án theo thủ tục chung và ra Bản án hoặc quyết định giải quyết vụ án.

Thời gian giải quyết (i) thời hạn chuẩn bị xét xử: Từ 4 đến 6 tháng kể từ ngày thụ lý vụ án; (ii) thời hạn mở phiên tòa: Từ 1 đến 2 tháng kể từ ngày có quyết định đưa vụ án ra xét xử.

3.1.3. Hiện trạng dữ liệu về các bản án hôn nhân gia đình

Nội dung về các bản án hiện tại của ngành Tòa án được ghi các sổ theo dõi hoặc một số đơn vị ghi lại trên bảng tính, có nhiều đặc trưng. Tuy nhiên chưa có phần mềm quản lý lữu trữ tập trung cơ sở dữ liệu các bản án, quyết định của Tòa án. Nội dung này được gọi là sổ thụ lý và theo dõi kết quả giải quyết các vụ việc hôn nhân gia đình.

SỔ THỤ LÝ VÀ KẾT QỦA GIẢI QUYẾT CÁC VỤ VIỆC HÔN NHÂN GIA ĐÌNH SƠ THẨM

THỤ LÝ *Số, ngày tháng năm*		NGUYÊN ĐƠN HOẶC NGƯỜI YÊU CẦU *Họ tên, địa chỉ. Họ tên người đại diện, chức vụ, địa chỉ*	BỊ ĐƠN HOẶC NGƯỜI LIÊN QUAN TRONG VIỆC HÔN NHÂN GIA ĐÌNH *Họ tên, địa chỉ. Họ tên người đại diện, chức vụ, địa chỉ*	ĐƠN KHỞI KIỆN CỦA CƠ QUAN, TỔ CHỨC	NGƯỜI CÓ QUYỀN LỢI, NGHĨA VỤ LIÊN QUAN Họ tên, địa chỉ. Họ tên người đại diện, chức vụ, địa chỉ	HỌ TÊN NGƯỜI BẢO VỆ QUYỀN, LỢI ÍCH HỢP PHÁP CỦA ĐƯƠNG SỰ	QUAN HỆ PHÁP LUẬT KHI THỤ LÝ	ÁP DỤNG BIỆN PHÁP KHẨN CẤP TẠM THỜI Số, ngày tháng năm	CHUYỂN HỒ SƠ VỤ VIỆC Số, ngày tháng năm và nơi nhận	TẠM ĐÌNH CHỈ *Số, ngày tháng năm*
1		2	3	4	5	6	7	8	9	10
155	02.10 2017	Nguyễn Hoàng Hậu 1992 Đ/c: T4 - 22 -06 TimesCity, phường Minh Khai, quận Hai Bà Trưng, Hà Nội	Terry John Arthur Moody 1987 Đ/c: 59 AvonGrove, Bletchley,Milton Keynes, MK37BL, UK			TB109/27.9.17 BL09641/29.9 2017 ST: 300.000đ Tp Lại Vĩnh Trung	Y/c công nhận thuận tình ly hôn			
156	02.10 2017	Nguyễn Thị Bích Hảo 1958 Đ/c: Số 43, ngõ 70, phố Nguyễn An Ninh, tổ 20, quận Hoàng Mai, Hà Nội	Nguyễn Văn Hoà 1955 Đ/c: Đường Grusebxka - 39 - 1, phòng 6, nhà 2, Tp Odessa, CH Ukraina			TB122/27.9.17 BL09641/29.9 2017 ST: 300.000đ Tp Lại Vĩnh Trung	T/c hôn nhân và gia đình			

Hình 3.2. Bảng thông tin theo dõi kết quả giải quyết dạng tệp excel

(Nguồn: Tòa án nhân dân thành phố Hà Nội)

3.2. Phân tích bài toán về quản lý án hôn nhân

3.2.1. Thông tin nguyên đơn

Các thông tin cơ bản của bên Nguyên đơn:

- Họ và tên;
- Giới tính: Nam/Nữ;
- Năm sinh;
- Tuổi: Tuổi của nguyên đơn khi nộp đơn ra tòa;
- Địa chỉ;
- Con chung.

Mỗi một bản án sẽ có mã vụ việc riêng giúp cho việc quản lý dễ dàng hơn.

3.2.2. Thông tin bị đơn

Các thông tin cơ bản liên quan đến bên Bị đơn:

- Họ và tên;
- Giới tính: Nam/Nữ;
- Năm sinh;
- Tuổi: Tuổi của bị đơn khi nộp đơn ra tòa.
- Địa chỉ.

3.2.3. Thông tin quyết định

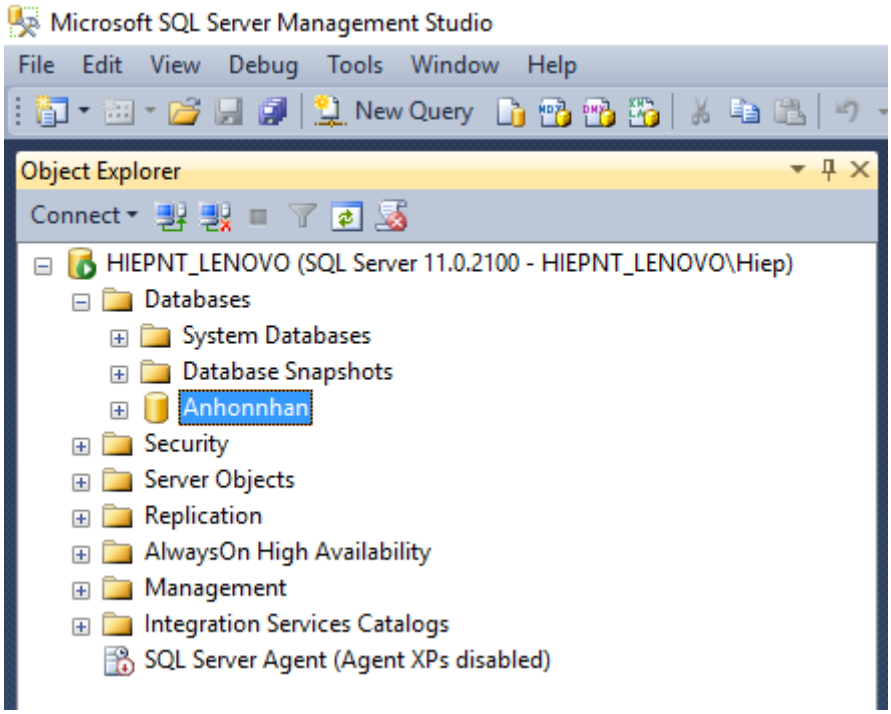
Với mỗi quyết định của toà án, sẽ bao gồm các thông tin về ngày thụ lý bản án, thẩm phán phụ trách, kết quả của vụ việc và quan hệ pháp luật của hai phía nguyên đơn và bị đơn.

- Ngày thụ lý bản án: Ngày toàn án tiếp nhận thụ lý bản án từ phía nguyên đơn.
- Thẩm phán: Họ và tên thẩm phán phụ trách vụ việc.
- Kết quả: Kết quả của vụ việc được tòa án tuyên bố (hòa giải hay xét xử).
- Quan hệ pháp luật: Các nguyên nhân dẫn đến việc đệ đơn (i) Mâu thuẫn gia đình; (ii) Bạo lực gia đình; (iii) Yếu tố nước ngoài; (iv) Một người mất tích; (v) Ngoại tình; (vi) Bệnh tật, không có con; (vii) Mâu thuẫn kinh tế; (viii) Nghiện rượu bia, cờ bạc, ma túy.

3.3. Thiết kế cơ sở dữ liệu án hôn nhân gia đình

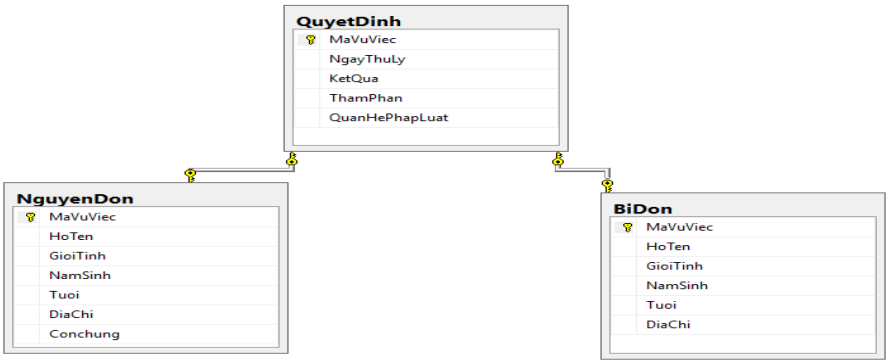
3.3.1. Cơ sở dữ liệu án hôn nhân gia đình

Cơ sở dữ liệu về các vụ, việc về hôn nhân gia đình trên hệ quản trị cơ sở dữ liệu SQL SERVER 2012 cụ thể là ***anhonnhan.***



Hình 3.3. Cơ sở dữ liệu về án hôn nhân

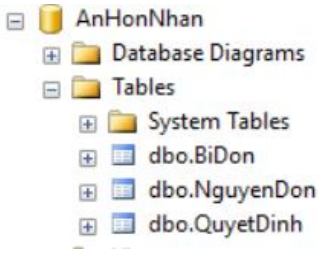
- Mô hình khái niệm gồm các thực thể và liên kết giữa các thực thể.



Hình 3.4. Sơ đồ thực thể quan hệ của bài toán

Các bảng dữ liệu được thiết kế theo mô hình khái niệm. Cơ sở dữ liệu cơ sở nhiều bảng quan hệ. Nó được thiết kế với 2 quan hệ:

- Bảng *NguyenDon*, cho biết các thông tin cơ bản của bên nguyên đơn.
- Bảng *BiDon*, cho biết các thông tin cơ bản của bên bị đơn tương ứng
- Bảng *QuyetDinh*, cho biết thông tin về kết quả của vụ việc, thẩm phán giải giải quyết với nguyên đơn và bị đơn tương ứng.



Hình 3.5. Các bảng quan hệ của cơ sở dữ liệu

3.3.2. Thiết kế chi tiết các bảng dữ liệu

Phần này giới thiệu các bảng dữ liệu theo thiết kế logic và thiết kế vật lí. Các thuộc tính được xác định (i) kiểu dữ liệu; (ii) độ rộng của thể hiện dữ liệu; (iii) ràng buộc dữ liệu. Mô tả ý nghĩa của từng thuộc tính có trong bảng chi tiết của hệ quản trị.

3.3.2.1. Bảng dữ liệu về nguyên đơn

Bảng *NguyenĐon* có 7 thuộc tính.

Column Name	Data Type	Allow Nulls
MaVuViec	int	☐
HoTen	nvarchar(100)	☐
GioiTinh	nvarchar(10)	☐
NamSinh	int	☐
Tuoi	int	☐
DiaChi	nvarchar(100)	☐
Conchung	nvarchar(50)	☐

Hình 3.6. Lược đồ bảng nguyên đơn

Bảng có khóa chính (Primary Key) là MaVuViec.

3.3.2.2. Bảng dữ liệu về bị đơn

Bảng *BiĐon* có 6 thuộc tính.

Bảng có khóa chính (Primary Key) là MaVuViec.

Column Name	Data Type	Allow Nulls
MaVuViec	int	☐
HoTen	nvarchar(100)	☐
GioiTinh	nvarchar(10)	☐
NamSinh	int	☐
Tuoi	int	☐
DiaChi	nvarchar(100)	☐

Hình 3.7. Lược đồ bảng bị đơn

3.3.2.3. Bảng dữ liệu về quyết định

- Bảng *QuyetDinh* có 5 thuộc tính.

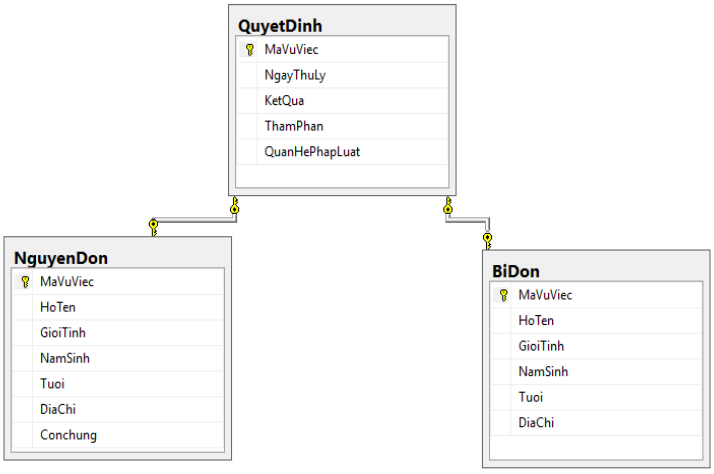
- Bảng có khóa chính (Primary Key) là MaVuViec.

	Column Name	Data Type	Allow Nulls
🔑	MaVuViec	int	☐
	NgayThuLy	date	☐
	KetQua	nvarchar(100)	☐
	ThamPhan	nvarchar(50)	☐
	QuanHePhapLuat	nvarchar(100)	☐

Hình 3.8. Lược đồ bảng quyết định

3.3.3. Quan hệ giữa các bảng dữ liệu

Bảng **QuyetDinh** có trường MaVuViec là khóa ngoại, tham chiếu đến trường MaVuViec của hai bảng **NguyenDon** và **BiDon** để lấy ra các thông tin của nguyên đơn và bị đơn.



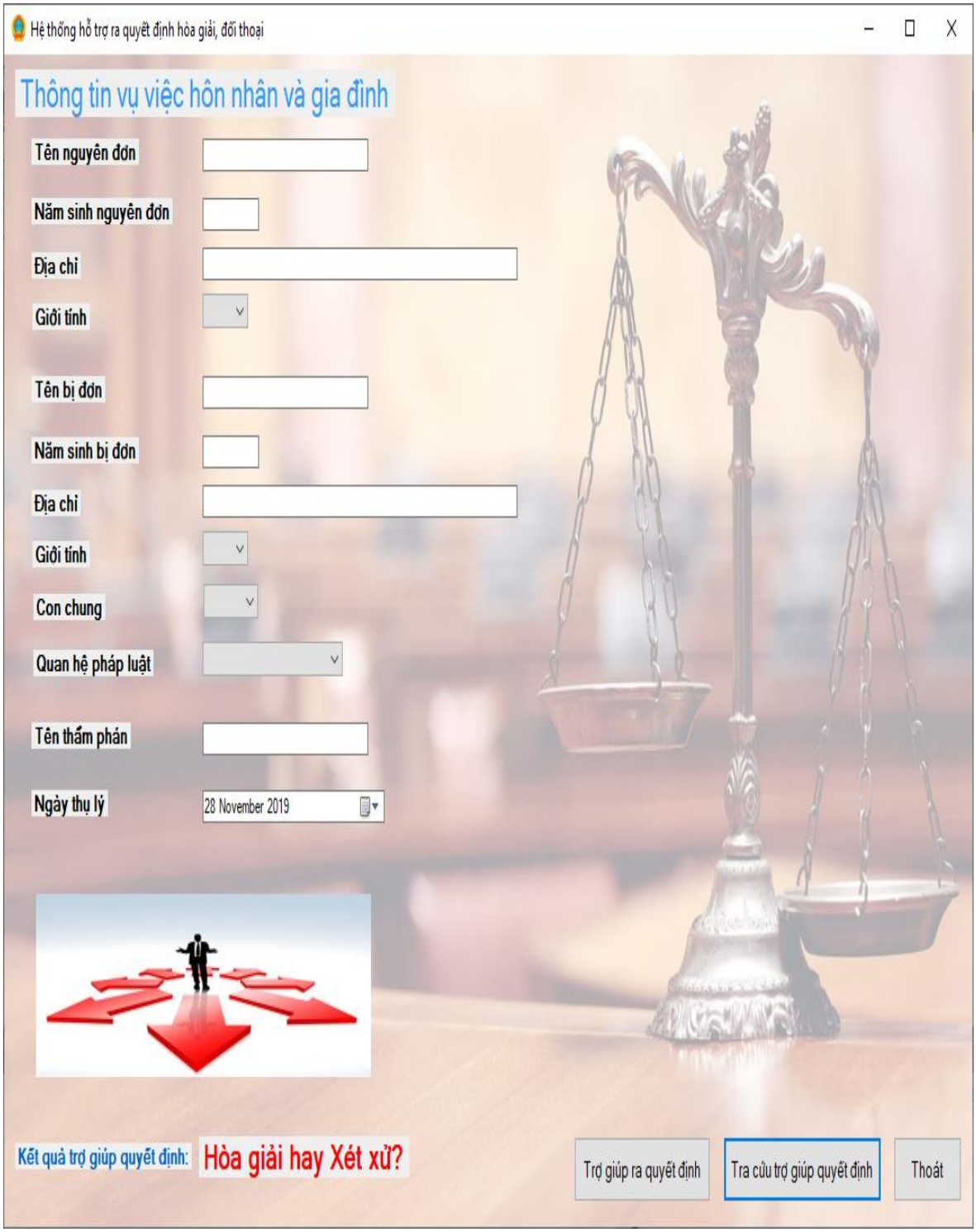
Hình 3.9. Lược đồ cơ sở dữ liệu

3.4. Xây dựng hệ thống trợ giúp quyết định trong môi trường C#

Visual Studio là một "công cụ hỗ trợ lập trình nổi tiếng" của Microsoft hiện nay mà chưa có phần mềm nào có thể thay thế được nó. Phần mềm Visual Studio được viết chính bằng hai ngôn ngữ đó là VB++ và C# đây là hai ngôn ngữ giúp người dùng có thể lập trình được hệ thống một cách dễ dàng nhất thông qua Visual Studio.

3.4.1. Chức năng Trợ giúp ra quyết định

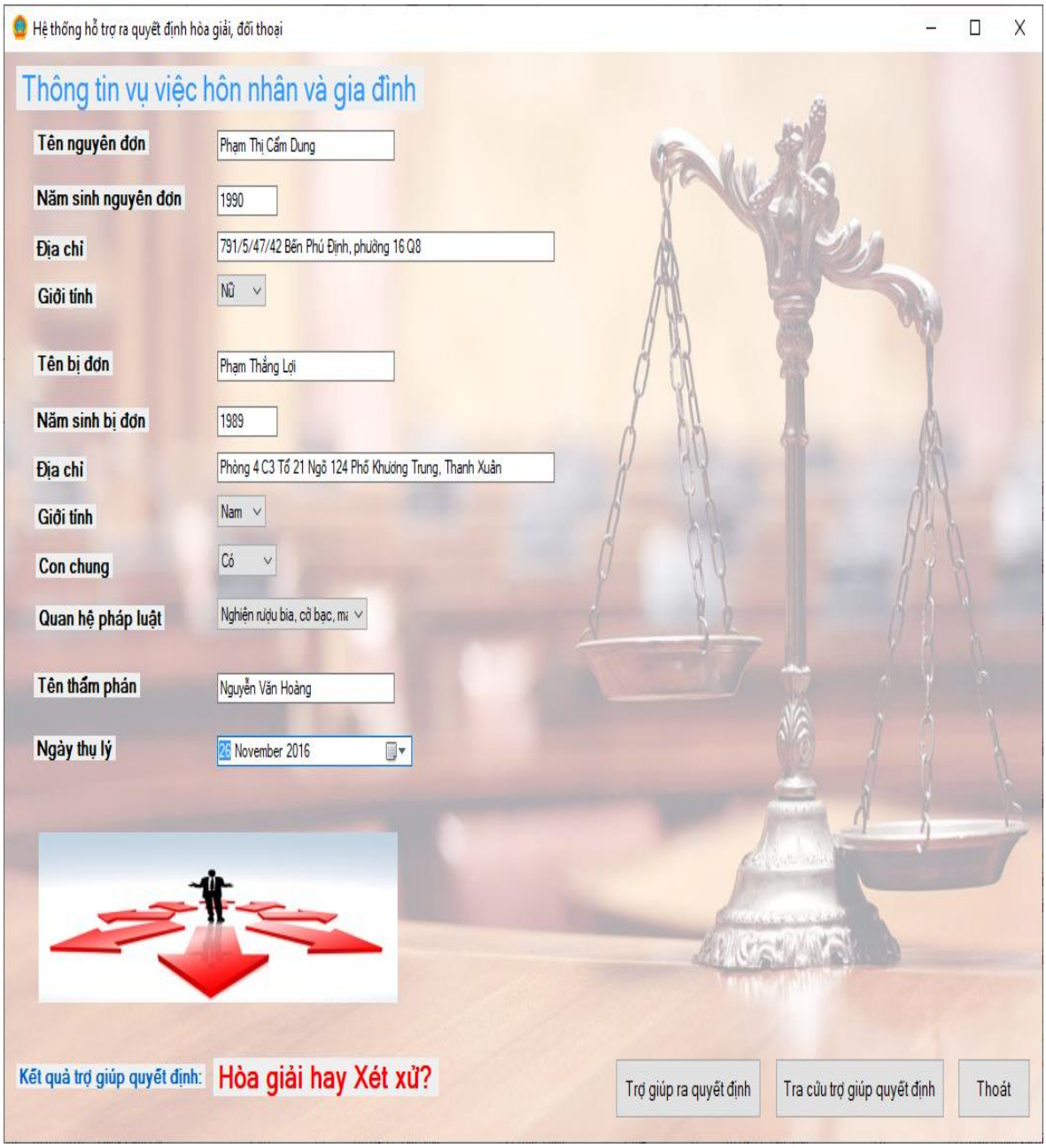
Chức năng chính của phần mềm là chức năng trợ giúp ra quyết định từ thông tin đầu vào của nguyên đơn và bị đơn được xây dựng trên bộ luật từ phần mềm Weka.



Hình 3.10. Giao diện chính

Khi Thẩm phán được phân công xem xét một đơn về vụ việc hôn nhân và giai đình bao, Thẩm phán đó sẽ xem xét nội dung đơn và đánh giá xem thuộc quan hệ pháp luật nào theo bộ luật tố tụng để lựa chọn.

Nhập thông tin đầu vào của đơn:

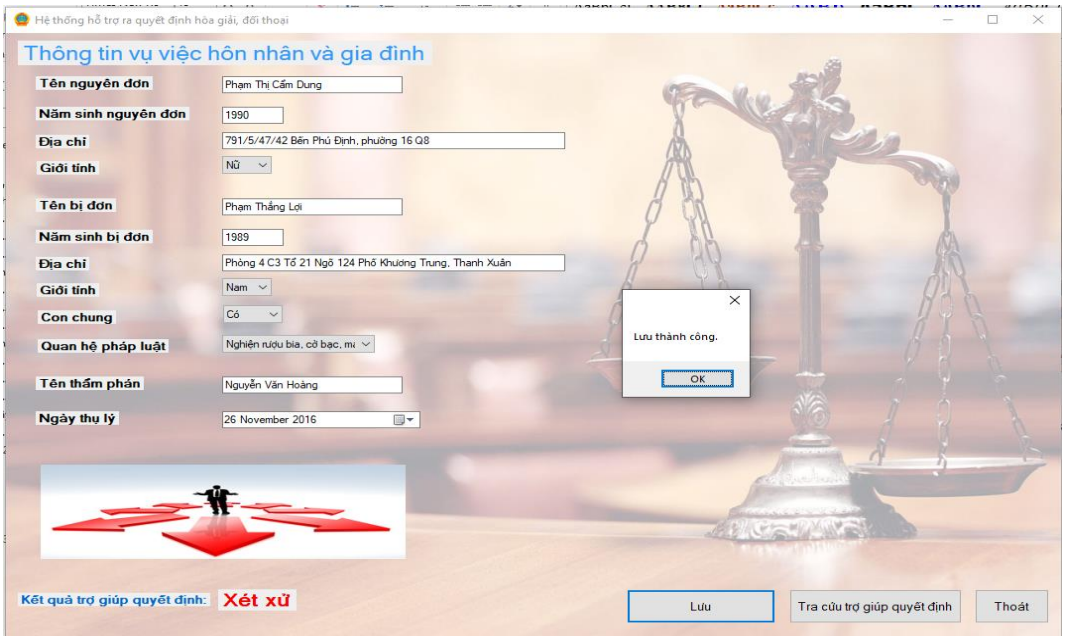


Hình 3.11. Nhập thông tin đơn ly hôn

Thực hiện chức năng "Trợ giúp ra quyết định": Khi Thẩm phán nhập đầy đủ thông tin như hình trên và thực hiện "Trợ giúp ra quyết định" hệ thống đưa ra kết quả trợ giúp có kết quả là "Xét xử" như sau:

Hình 3.12. Kết quả trợ giúp ra quyết định

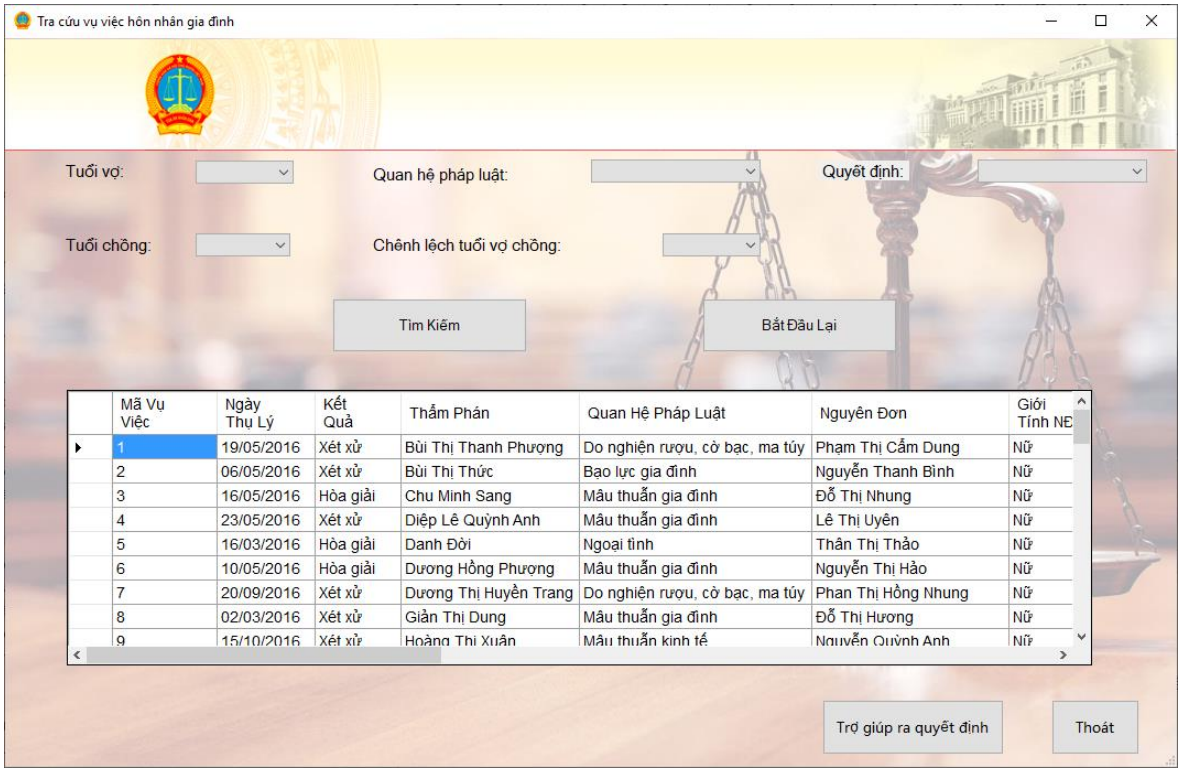
Thực hiện chức năng "Lưu": Thẩm phán xem kết quả trợ giúp và lưu kết quả trợ giúp và hệ thống phục vụ công tác lưu trữ cơ sở dữ liệu về các bản án và hỗ trợ tìm kiếm thống kê, giám sát thực hiện,…



Hình 3.13. Lưu kết quả trợ giúp ra quyết định

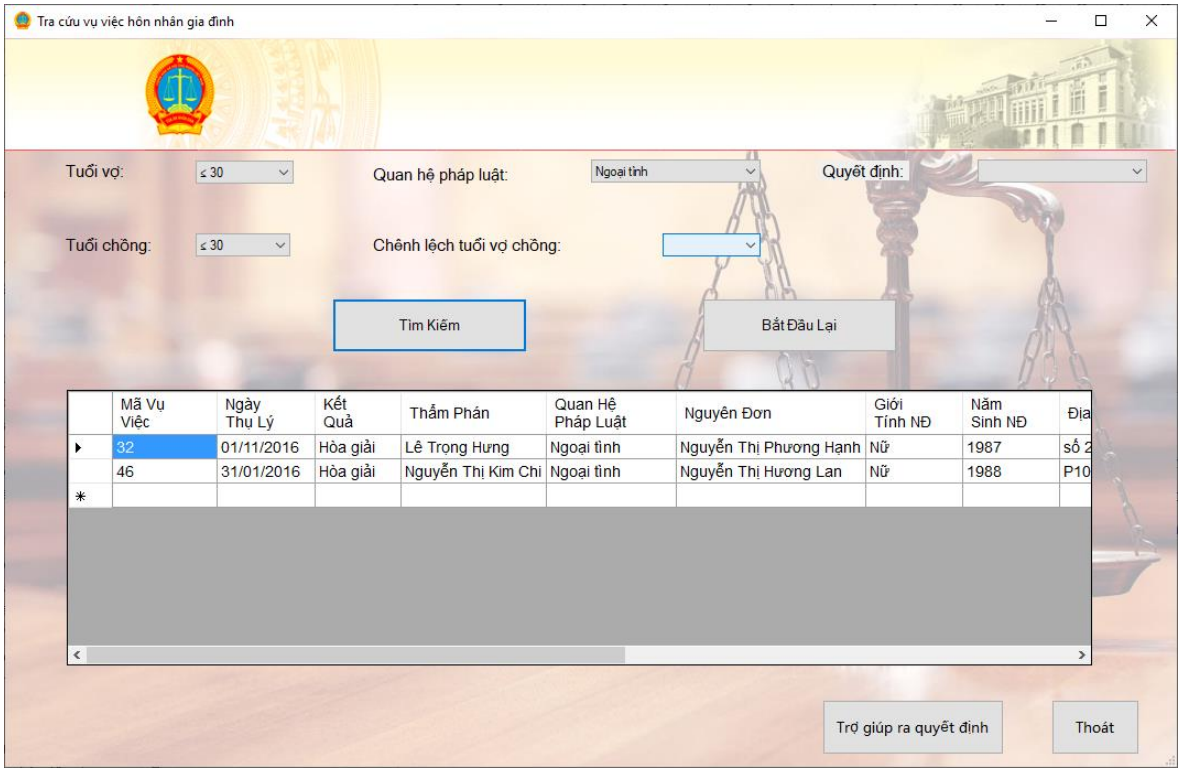
3.4.2. Chức năng tra cứu bản án, quyết định

Chức năng này phục vụ việc tra cứu thông tin về các vụ việc, bản án, quyết định của Tòa án theo các thuộc tính thông tin đầu vào



Hình 3.14. Màn hình tra cứu thông tin bản án, quyết định

Thực hiện tra cứu thông tin: Khi cán bộ được phân quyền, hoặc lãnh đạo muốn xem xét kết quả trợ giúp của hệ thống và tra cứu thông tin liên quan về nguyên đơn, bị đơn,…Cán bộ Tòa án nhập vào các thông tin cần tra cứu và thực hiện "Tìm kiếm" hệ thống sẽ hiển thị danh sách và bản án, quyết định cần tra cứu:



Hình 3.15. Kết quả tra cứu thông tin bản án, quyết định

3.3. Kết luận

Chương trên đã trình bày hệ thống truy cập dữ liệu và trợ giúp ra quyết định về án hôn nhân. Hệ thống sử dụng (i) hệ quản trị cơ sở dữ liệu SQL SERVER; (ii) môi trường Visual C#.

Ứng dụng với giao diện phù hợp, thuận tiện đơn giản cho người sử dụng, các chức năng đáp ứng yêu cầu trong đó có chức năng "Trợ giúp ra quyết định" được thực hiện với đầu vào là các thông tin về nguyên đơn, bị đơn, quan hệ pháp luật xử lý hỗ trợ ra quyết định được xây dựng trên bộ luật được tạo ra từ cây quyết định tại chương 2.

Thông tin từ cơ sở dữ liệu là hữu ích đối với việc tra cứu các án hôn nhân, cũng như đưa ra phương án dự thảo án mới, trước tình huống cần trợ giúp, tư vấn.

KẾT LUẬN

Những kết quả đạt được

Luận văn đã trình bày nhu cầu xây dựng và sử dụng hệ thống trợ giúp quyết định trong việc thụ lý, xem xét đưa ra quyết định về vụ, việc trong lĩnh vực hôn nhân và gia đình. Bản án, quyết định về hôn nhân gia đình được đề cập như đối tượng để thực hiện việc ra quyết định với vụ, việc về hôn nhân gia đình dựa trên khai phá dữ liệu về các bản án, quyết định của Tòa án.

Trong quá trình thực hiện và hoàn thiện luận văn với sự hướng dẫn của giáo viên hướng dẫn tôi đã thực hiện, hoàn thiện và đạt được một số kết quả như đã nêu trong mục tiêu của đề cương cụ thể như sau:

Chương đầu đã trình bày và hiểu được nội dung tổng quan, kiến trúc, các chức năng, phân tích lớp và dự đoán, phân cụm khai phá dữ liệu. Khái niệm về các hệ thống ra quyết định, phân loại các hệ thống ra quyết định, cây quyết định trong đó đã thực hiện nghiên cứu, phân tích đánh giá chung các thuật toán phân lớp dữ liệu là thuật toán ID3 và thuật toán C4.5. Qua đó đánh giá được thuật toán C4.5 là thuật toán xử lý đầy đủ các vấn đề phân lớp dữ liệu và phổ biến nhất, đây là cơ sở lựa chọn áp dụng thuật toán này vào việc xây dựng hệ thống hỗ trợ ra quyết định về hòa giải trong các tranh chấp về hôn nhân giai đình.

Chương thứ hai áp dụng được việc thử nghiệm phương pháp phân lớp bằng cây quyết định sử dụng thuật toán J48 trong bộ công cụ phần mềm WEKA, kết quả đưa ra được là cây quyết định và bộ luật phân lớp áp dụng xây dựng phần mềm hỗ trợ ra quyết định.

Chương cuối đã xây dựng cơ sở dữ liệu và phần mềm hỗ trợ ra quyết định mà luận văn đề xuất sử dụng trong công việc của Tòa án. Cơ sở dữ liệu sử dụng hệ quản trị cơ sở dữ liệu SQL SERVER. Việc đưa công nghệ thông tin và truyền thông vào công tác hỗ trợ giải quyết vụ việc trước hết cần hệ thống thông tin; mà cốt lõi của nó là cơ sở dữ liệu về án và cài đặt thuật toán phân lớp dữ liệu. Đối với luận văn, đó là cơ sở dữ liệu về các án hôn nhân gia đình và thuật toán phân lớp dữ liệu

dựa trên cây quyết định. Những kết quả này có ý nghĩa khoa học thực tiễn và có hướng gợi mở cho việc nghiên cứu tiếp theo cho các lĩnh vực tố tụng khác của Tòa án như hình sự, dân sự, kinh doanh thương mại, lao động, hành chính. Là yếu tố hướng tới xây dựng Tòa án thông minh.

Hướng nghiên cứu phát triển của luận văn

Trong giới hạn của luận văn hệ thống mới xây dựng thử nghiệm trên bộ dữ liệu thực tế còn hạn chế từ sổ thụ lý và kết qủa giải quyết các vụ việc hôn nhân gia đình sơ thẩm của Tòa án nhân dân thành phố Hà Nội. Do luận văn đề xuất (i) hệ thống cơ sở dữ liệu; (ii) phân loại tự động các bản án hôn nhân gia đình, nên hướng tiếp theo sau luận văn của học viên sẽ là :

- Hoàn thiện và phát triển hệ thống cơ sở dữ liệu đã tạo ra trong quá trình làm luận văn tốt nghiệp;
- Hoàn thiện các chức năng của phần mềm hướng tới áp dụng thực tế;
- Bổ sung dữ liệu quá khứ về các án hôn hôn nhân gia đình, dữ liệu của các đơn vị Tòa án khác trong ngành Tòa án;
- Nghiên cứu thêm các thuật toán khác như C5.0, SPRINT;
- Nghiên cứu xây dựng hệ thống mở rộng áp dụng cho các lĩnh vực tố tụng khác như hình sự, dân sự, hành chính, lao động, kinh doanh thương mại.

Thực tế trong công tác thụ lý giải quyết vụ, việc và xét xử còn nhiều yếu tố và cần phải nghiên cứu đòi hỏi tính chặt chẽ căn cứ theo các luật, bộ luật tố tụng tại Việt Nam. Hệ thống phần mềm được xây dựng trong luận văn mang tính chất "hỗ trợ" ra quyết định trợ giúp cho công tác giải quyết vụ, việc ngày càng nhiều được nâng cao và hiệu quả hơn trong cải cách tư pháp tại Tòa án nhân dân.

TÀI LIỆU THAM KHẢO

[1.] Đỗ Trung Tuấn, "Hệ trợ giúp quyết định", nxb. Đại học Quốc gia Hà Nội, 2016;

[2.] Hoàng Xuân Huấn, "Hệ thống trợ giúp quyết định, bài giảng trường Đại học Công nghệ, Đại học Quốc gia Hà Nội", 2009;

[3.] Hồ Thuần, Hồ Cẩm Hà, "Các hệ cơ sở dữ liệu, lí thuyết và thực hành", tập II, Nxb. Giáo dục, 2010;

[4.] Hồ Tú Bảo (bao@jaist.ac.jp), Khoa học Dữ liệu và Cách mạng Công nghiệp lần thứ Tư;

[5.] Badr HSSINA, Abdelkarim MERBOUHA, Hanane ZZIKOURI, Mohammed ERRITALI, "A comparative study of decision tree ID3 and C4.5";

[6.] Ian H. Witten, Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques, Second Edition", 2005;

[7.] Delic, K.A., Douillet,L. and Dayal, U., "Towards an architecture for real-time decision support systems:challenges and solutions", 2001;

[8.] Power, D. J. Web-based and model-driven decision support systems: concepts and issues. in proceedings of the Americas Conference on Information Systems, Long Beach, California, 2000;

[9.] Sprague et als., Decision Support Systems, Ed. Prentice Hall, 2010;

[10.] Tòa án nhân dân tối cao, "Báo cáo công tác năm 2019 của Chánh án Tòa án nhân dân tối cao trước Quốc hội";

[11.] Tòa án nhân dân tối cao, "Hệ thống Công bố bản án, quyết định của Tòa án https://congbobanan.toaan.gov.vn";

[12.] Tòa án nhân dân tối cao , "Hệ thống quản lý cán bộ công chức ngành Tòa án nhân dân, https://qlcb.toaan.gov.vn";

[13.] Decision tree learning, https://en. wikipedia.org/ wiki/ Decision_tree_learning;

[14.] Introduction to Decision Trees, https://medium.com/greyatom/ decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb;

PHỤ LỤC

1. Các đoạn mã sử dụng chuẩn hóa dữ liệu

Chuẩn hóa dữ liệu từ tệp excel sang tệp data_toaan.arff.

1.1.Đọc dữ liệu

```
from pandas import ExcelFile
from case import Case

pd.options.mode.chained_assignment = None

# MARK:- get discrete data

df = pd.read_excel('source.xlsx')

ids = df['id']
dates = df['date']
legal_relas = df['legal_rela']
plaintiff_ages = df['plaintiff_age']
defendant_ages = df['defendant_age']
decisions = df['decision']

length = len(legal_relas) - 1

distances = []

for i in range(0, length):
    relation = legal_relas[i]
    decision = decisions[i]

    corr_legal = re.search(r'\d', relation)
    corr_decis = re.search(r'\d', decision)

    legal_relas[i] = int(corr_legal.group())
    decisions[i] = int(corr_decis.group())

# MARK:- create list objects

cases = []

for i in range(0, length):
    case = Case(ids[i], legal_relas[i], plaintiff_ages[i], defendant_ages[i], decisions[i])
    cases.append(case)

# MARK:- write to file
file = open("data_toaan.arff", "w")

for case in cases:
    file.write(case.toString())

file.close()
```

1.2.Gán nhãn cho các thuộc tính

```
class Case:
    """
    represent a case in a court
    including:
     id,
     legal_rela,
     age_dist,
     has_child,
     husb_age,
     wife_age,
     decision
    """

    def __init__(self, id, legal_rela, plaintiff_age, defendant_age, decision):
        self.id = id
        self.legal_rela = legal_rela
        self.getPlaintiffAge(plaintiff_age)
        self.getDefendantAge(defendant_age)
        self.decision = decision
        self.getAgeDistance(plaintiff_age, defendant_age)
        self.getHasChild(legal_rela)

    def getDefendantAge(self, defendant_age):
        if defendant_age <= 30:
            self.defendant_age = 1
        else:
            self.defendant_age = 2

    def getPlaintiffAge(self, plaintiff_age):
        if plaintiff_age <= 30:
            self.plaintiff_age = 1
        else:
            self.plaintiff_age = 2

    def getAgeDistance(self, plaintiff_age, defendant_age):
        age_dist = abs(plaintiff_age - defendant_age)

        if age_dist <= 5:
            self.age_dist = 1
        elif age_dist <= 10:
            self.age_dist = 2
        else:
            self.age_dist = 3

    def getHasChild(self, legal_rela):
```

```
46          if legal_rela == 5:
47              self.has_child = 0
48          else:
49              self.has_child = 1
50
51      def rela_convert(self, legal_rela):
52          switcher = {
53              1: "MTGD",
54              2: "NN",
55              3: "NT",
56              4: "TNXH",
57              5: "BT",
58              6: "MT",
59              7: "MTKT",
60              8: "BLGD"
61          }
62          return switcher.get(legal_rela, "MTGD")
63
64      def child_convert(self, has_child):
65          switcher = {
66              0: "khong",
67              1: "co"
68
69          }
70          return switcher.get(has_child, "khong")
71
72      def age_convert(self, age):
73          switcher = {
74              1: "<=30",
75              2: ">30"
76          }
77          return switcher.get(age, "<=30")
78
79      def dist_convert(self, age_dist):
80          switcher = {
81              1: "<=5",
82              2: "5_10",
83              3: ">10"
84          }
85          return switcher.get(age_dist, "<=5")
```

```
86
87      def decision_convert(self, decision):
88          switcher = {
89              1: "Hoagiai",
90              2: "Xetxu"
91          }
92          return switcher.get(decision, "Hoagiai")
93
94      def toString(self):
95          output = []
96          output.append(self.rela_convert(self.legal_rela))
97          output.append(self.child_convert(self.has_child))
98          output.append(self.age_convert(self.plaintiff_age))
99          output.append(self.age_convert(self.defendant_age))
100         output.append(self.dist_convert(self.age_dist))
101         output.append(self.decision_convert(self.decision))
102
103         output = ','.join(output)
104         output += '\n'
105         return output
106
```

1.3.Ghi ra tệp data_toaan.arff

```
# MARK:- libs
import pandas as pd
import re
from pandas import ExcelWriter
from pandas import ExcelFile
from random import seed
from random import randint
from random import choice
from case import Case


# MARK:- support functions
def new_case_one():
    """
        output.append(str(self.legal_rela))
        output.append(str(self.has_child))
        output.append(str(self.plaintiff_age))
        output.append(str(self.defendant_age))
        output.append(str(self.age_dist))
        output.append(str(self.decision))
    """

    legal_rela = choice([4, 7, 8])
    age_dist = choice([1, 2])
    decision = 2
    plaintiff_age = randint(23, 53)
    defendant_age = plaintiff_age + randint(1, 5)

    case = Case(1000, legal_rela, plaintiff_age, defendant_age, decision)

    return case


def new_case_two():
    legal_rela = choice([1, 5])
    age_dist = 2
    decision = 2
    plaintiff_age = randint(18, 53)
    defendant_age = plaintiff_age + randint(6, 10)

    case = Case(1000, legal_rela, plaintiff_age, defendant_age, decision)
    case.has_child = 0
    return case
```

```
44  def new_case_three():
45      legal_rela = choice([2, 3, 6])
46      age_dist = choice([1, 2])
47      decision = choice([1, 2])
48      plaintiff_age = randint(18, 53)
49      defendant_age = plaintiff_age + randint(6, 10)
50
51      case = Case(1000, legal_rela, plaintiff_age, defendant_age, decision)
52      case.has_child = choice([0,1])
53      return case
54
55
56  # MARK:- write to file
57  file = open("data_toaan.arff", "a")
58  seed(1, version=1)
59  # generate some random numbers
60  for _ in range(10):
61      case = new_case_one()
62      file.write(case.toString())
63
64  seed(1, version=2)
65  for _ in range(30):
66      case = new_case_two()
67      file.write(case.toString())
68
69  seed(1, version=3)
70  for _ in range(30):
71      case = new_case_three()
72      file.write(case.toString())
73  seed(1, version=4)
74  for _ in range(30):
75      case = new_case_one()
76      file.write(case.toString())
77
78  seed(1, version=5)
79  for _ in range(10):
80      case = new_case_two()
81      file.write(case.toString())
82
83  seed(1, version=6)
84  for _ in range(50):
85      case = new_case_three()
86      file.write(case.toString())
87
88  file.close()
```

2. Cơ sở dữ liệu

- Bảng QUYETDINH

```
USE [Anhonnhan]
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[QuyetDinh](
    [MaVuViec] [int] IDENTITY(1,1) NOT NULL,
    [NgayThuLy] [date] NOT NULL,
    [KetQua] [nvarchar](100) NOT NULL,
    [ThamPhan] [nvarchar](50) NOT NULL,
    [QuanHePhapLuat] [nvarchar](100) NOT NULL,
 CONSTRAINT [PK_QuyetDinh_MaVuViec] PRIMARY KEY CLUSTERED
(
    [MaVuViec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [
) ON [PRIMARY]

GO
```

- Bảng NGUYENDON

```
USE [Anhonnhan]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[NguyenDon](
    [MaVuViec] [int] NOT NULL,
    [HoTen] [nvarchar](100) NOT NULL,
    [GioiTinh] [nvarchar](10) NOT NULL,
    [NamSinh] [int] NOT NULL,
    [Tuoi] [int] NOT NULL,
    [DiaChi] [nvarchar](100) NOT NULL,
    [Conchung] [nvarchar](50) NULL,
 CONSTRAINT [PK_NguyenDon_MaVuViec] PRIMARY KEY CLUSTERED
(
    [MaVuViec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[NguyenDon]  WITH CHECK ADD FOREIGN KEY([MaVuViec])
REFERENCES [dbo].[QuyetDinh] ([MaVuViec])
GO
```

- Bảng BIDON

```
USE [Anhonnhan]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[BiDon](
    [MaVuViec] [int] IDENTITY(1,1) NOT NULL,
    [HoTen] [nvarchar](100) NOT NULL,
    [GioiTinh] [nvarchar](10) NOT NULL,
    [NamSinh] [int] NOT NULL,
    [Tuoi] [int] NOT NULL,
    [DiaChi] [nvarchar](100) NOT NULL,
 CONSTRAINT [PK_BiDon_MaVuViec] PRIMARY KEY CLUSTERED
(
    [MaVuViec] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [P
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[BiDon]  WITH CHECK ADD FOREIGN KEY([MaVuViec])
REFERENCES [dbo].[QuyetDinh] ([MaVuViec])
GO
```

- Thủ tục SP_INSERT

```
USE [Anhonnhan]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

Create PROCEDURE [dbo].[SP_INSERT]
(
                @NgayThuLy [datetime],
                @KetQua [nvarchar](100),
                @ThamPhan [nvarchar](50),
                @quanhephapluat [nvarchar](100),
                @HoTenND [nvarchar](100),
                @GioiTinhND [nvarchar](10),
                @NamSinhND [int],
                @tuoiND [int],
                @DiaChiND [nvarchar](100),
                @Conchung [nvarchar](50),
                @HoTenBD [nvarchar](100),
                @GioiTinhBD [nvarchar](10),
                @NamSinhBD [int],
                @tuoiBD [int],
                @DiaChiBD [nvarchar](100)
)
 AS
BEGIN TRANSACTION AT
    BEGIN
        INSERT INTO [dbo].[QuyetDinh]
        (
            NgayThuLy,
            KetQua,
            ThamPhan,
            QuanHePhapLuat
        )
        VALUES
        (
            @NgayThuLy,
            @KetQua,
            @ThamPhan,
            @quanhephapluat
        );
```

- Các constraint

```
ALTER TABLE [dbo].[BiDon]  WITH CHECK ADD FOREIGN KEY([MaVuViec])
REFERENCES [dbo].[QuyetDinh] ([MaVuViec])
GO
ALTER TABLE [dbo].[NguyenDon]  WITH CHECK ADD FOREIGN KEY([MaVuViec])
REFERENCES [dbo].[QuyetDinh] ([MaVuViec])
GO
```

3. Mã code chương trình c#

Một số đoạn code cơ bản trong chương trình

- Form trợ giúp ra quyết định

```
public partial class Trogiupraquyetdinh : Form
{
    public static SqlConnection Connection()
    {
        SqlConnection conn = new SqlConnection("Data Source = localhost; Initial Catalog = AnHonNhan; Integrated Security = True");
        conn.Open();
        return conn;
    }
    public bool Insert_Update()
    {
        SqlConnection conn = Connection();
        SqlCommand comm = new SqlCommand("SP_INSERT", conn);
        comm.CommandType = CommandType.StoredProcedure;
        SqlCommandBuilder.DeriveParameters(comm);
        SqlTransaction tran = conn.BeginTransaction();
        comm.Transaction = tran;
        comm.Parameters["@NgayThuLy"].Value = dateThuly.Value;
        comm.Parameters["@KetQua"].Value = ketquaQD.Text;
        comm.Parameters["@ThamPhan"].Value = txtTenThamphan.Text;
        comm.Parameters["@quanhephapluat"].Value = cbQHPL.Text;
        comm.Parameters["@HoTenND"].Value = txtTenND.Text;
        comm.Parameters["@GioiTinhND"].Value = cbGioitinhND.Text;
        comm.Parameters["@NamSinhND"].Value = txtNamSinhND.Text;
        comm.Parameters["@tuoiND"].Value = int.Parse(dateThuly.Value.ToString("yyyy")) - int.Parse(txtNamSinhND.Text);
        comm.Parameters["@DiaChiND"].Value = txtDiachiND.Text;
        comm.Parameters["@Conchung"].Value = cbConchung.Text;

        comm.Parameters["@HoTenBD"].Value = txtTenBD.Text;
        comm.Parameters["@GioiTinhBD"].Value = cbGioitinhBD.Text;
        comm.Parameters["@NamSinhBD"].Value = txtNamsinhBD.Text;
        comm.Parameters["@tuoiBD"].Value = int.Parse(dateThuly.Value.ToString("yyyy")) - int.Parse(txtNamsinhBD.Text);
        comm.Parameters["@DiaChiBD"].Value = txtDiachiBD.Text;
        try
        {
            comm.ExecuteNonQuery();
            tran.Commit();
            return true;
        }
        catch
        {
            tran.Rollback();
            return false;
        }
        finally
        {
            conn.Close();
        }
    }
```

```
62          public Trogiupraquyetdinh()
63          {
64              InitializeComponent();
65              load_drop();
66          }
67          private void label2_Click(object sender, EventArgs e)
68          {
69
70          }
71
72          public void load_drop()
73          {
74              dateThuly.Value = DateTime.Now;
75
76              txtTenND.Text="";
77              txtNamSinhND.Text = "";
78              txtDiachiND.Text = "";
79              txtTenBD.Text = "";
80              txtNamsinhBD.Text = "";
81              txtDiachiBD.Text = "";
82              txtTenThamphan.Text = "";
83
84              this.cbQHPL.Items.Clear();
85              this.cbQHPL.Items.Add("Mâu thuẫn gia đình");
86              this.cbQHPL.Items.Add("Yếu tố nước ngoài");
87              this.cbQHPL.Items.Add("Ngoại tình");
88              this.cbQHPL.Items.Add("Nghiện rượu bia, cờ bạc, ma túy");
89              this.cbQHPL.Items.Add("Bệnh tật, không có con");
90              this.cbQHPL.Items.Add("Một người mất tích");
91              this.cbQHPL.Items.Add("Mâu thuẫn kinh tế");
92              this.cbQHPL.Items.Add("Bạo lực gia đình");
93
94
95              this.cbGioitinhND.Items.Clear();
96              this.cbGioitinhND.Items.Add("Nữ");
97              this.cbGioitinhND.Items.Add("Nam");
98
99              this.cbGioitinhBD.Items.Clear();
100             this.cbGioitinhBD.Items.Add("Nữ");
101             this.cbGioitinhBD.Items.Add("Nam");
102
103             this.cbConchung.Items.Clear();
104             this.cbConchung.Items.Add("Có");
105             this.cbConchung.Items.Add("Không");
106
107             trogiupquyetdinh.Text = "Trợ giúp ra quyết định";
108             this.ketquaQD.Text = "Hòa giải hay Xét xử?";
```

```
109
110         }
111         private void button1_Click(object sender, EventArgs e)
112         {
113             this.Visible = false;
114             Form1 f1 = new Form1();
115             f1.ShowDialog();
116         }
117
118         private void label7_Click(object sender, EventArgs e)
119         {
120
121         }
122
123         private void label6_Click(object sender, EventArgs e)
124         {
125
126         }
127
128         private void textBox4_TextChanged(object sender, EventArgs e)
129         {
130
131         }
132
133         private void textBox5_TextChanged(object sender, EventArgs e)
134         {
135
136         }
137
138         private void cbQHPL_SelectedIndexChanged(object sender, EventArgs e)
139         {
140
141
142         }
143
144         private void Form2_Load(object sender, EventArgs e)
145         {
146
147         }
148
149         private void btTrogiupraquyetdinh_Click(object sender, EventArgs e)
150         {
151             if (trogiupquyetdinh.Text == "Luu")
152             {
153                 if (Insert_Update() == true)
154                 {
155                     MessageBox.Show("Luu thành công.");
156                     load_drop();
```

```
157                 }
158                 else
159                 {
160                     MessageBox.Show("Có lỗi xảy ra, bạn kiểm tra lại." );
161                 }
162             }
163             else
164             {
165                 int tuoind, tuoibd, do_lech_tuoi;
166                 tuoind = int.Parse(dateThuly.Value.ToString("yyyy")) - int.Parse(txtNamSinhND.Text);
167                 tuoibd = int.Parse(dateThuly.Value.ToString("yyyy")) - int.Parse(txtNamsinhBD.Text);
168                 do_lech_tuoi = Math.Abs(tuoind - tuoibd);
169                 //Rule 1:
170 
171                 if (cbQHPL.Text == "Mâu thuẫn gia đình")
172                 {
173                     ketquaQD.Text = "Xét xử";
174                 }
175 
176                 //Rule 2:
177                 if (cbQHPL.Text == "Yếu tố nước ngoài")
178                 {
179                     ketquaQD.Text = "Hòa giải";
180                 }
181 
182                 //Rule 3:
183                 if (cbQHPL.Text == "Ngoại tình" && cbConchung.Text == "Không" && tuoind <= 30)
184                 {
185                     ketquaQD.Text = "Xét xử";
186                 }
187 
188                 //Rule 4:
189                 if (cbQHPL.Text == "Ngoại tình" && cbConchung.Text == "Không" && tuoind > 30)
190                 {
191                     ketquaQD.Text = "Hòa giải";
192                 }
193 
194                 //Rule 5:
195                 if (cbQHPL.Text == "Ngoại tình" && cbConchung.Text == "Có")
196                 {
197                     ketquaQD.Text = "Hòa giải";
198                 }
199 
200                 //Rule 6:
201                 if (cbQHPL.Text == "Nghiện rượu bia, cờ bạc, ma túy")
202                 {
203                     ketquaQD.Text = "Xét xử";
204                 }
```

```
205
206             //Rule 7:
207             if (cbQHPL.Text == "Bệnh tật, không có con" && do_lech_tuoi <= 5)
208             {
209                 ketquaQD.Text = "Hòa giải";
210             }
211
212             //Rule 8:
213             if (cbQHPL.Text == "Bệnh tật, không có con" && 5 < do_lech_tuoi && do_lech_tuoi <= 10)
214             {
215                 ketquaQD.Text = "Xét xử";
216             }
217
218             //Rule 9:
219             if (cbQHPL.Text == "Bệnh tật, không có con" && do_lech_tuoi > 10)
220             {
221                 ketquaQD.Text = "Hòa giải";
222             }
223
224             //Rule 10:
225             if (cbQHPL.Text == "Một người mất tích" && tuoibd <= 30)
226             {
227                 ketquaQD.Text = "Xét xử";
228             }
229
230             //Rule 11:
231             if (cbQHPL.Text == "Một người mất tích" && tuoibd > 30 && cbConchung.Text == "Không")
232             {
233                 ketquaQD.Text = "Xét xử";
234             }
235
236             //Rule 12:
237             if (cbQHPL.Text == "Một người mất tích" && tuoibd > 30 && cbConchung.Text == "Có")
238             {
239                 ketquaQD.Text = "Hòa giải";
240             }
241
242             //Rule 13:
243             if (cbQHPL.Text == "Mâu thuẫn kinh tế")
244             {
245                 ketquaQD.Text = "Xét xử";
246             }
247
248             //Rule 14:
249             if (cbQHPL.Text == "Bạo lực gia đình")
250             {
251                 ketquaQD.Text = "Xét xử";
252             }
```

```
236                 //Rule 12:
237                 if (cbQHPL.Text == "Một người mất tích" && tuoibd > 30 && cbConchung.Text == "Có")
238                 {
239                     ketquaQD.Text = "Hòa giải";
240                 }
241
242                 //Rule 13:
243                 if (cbQHPL.Text == "Mâu thuẫn kinh tế")
244                 {
245                     ketquaQD.Text = "Xét xử";
246                 }
247
248                 //Rule 14:
249                 if (cbQHPL.Text == "Bạo lực gia đình")
250                 {
251                     ketquaQD.Text = "Xét xử";
252                 }
253
254                 //-----------------------------------
255                 if (ketquaQD.Text == "Hòa giải hay Xét xử?")
256                 {
257                     trogiupquyetdinh.Text = "Trợ giúp ra quyết định";
258
259                 }
260                 else
261                 {
262                     trogiupquyetdinh.Text = "Luu";
263                 }
264             }
265         }
266
267         private void btThoat_Click(object sender, EventArgs e)
268         {
269             Application.Exit();
270         }
271
272         private void button2_Click_1(object sender, EventArgs e)
273         {
274
275         }
276
277         private void label15_Click(object sender, EventArgs e)
278         {
279
280         }
281     }
282 }
283
```

- Form Tra cứu

```
14  public partial class Tracuu : Form
15  {
16      SqlConnection connection;
17      SqlCommand command;
18      string cn = @"Data Source=localhost;Initial Catalog=AnHonNhan;Integrated Security=True";
19      SqlDataAdapter adapter = new SqlDataAdapter();
20      DataTable table = new DataTable();
21
22
23      void loaddata(string str)
24      {
25          command = connection.CreateCommand();
26          command.CommandText = str;
27          adapter.SelectCommand = command;
28          table.Clear();
29          adapter.Fill(table);
30          //dgv.DataSource = "";
31          dgv.DataSource = table;
32
33      }
34
35      private void Tracuu_Load(object sender, EventArgs e)
36      {
37          connection = new SqlConnection(cn);
38          connection.Open();
39          loaddata("select QuyetDinh.MaVuViec as 'Mã Vụ Việc', QuyetDinh.NgayThuLy as 'Ngày Thụ Lý', QuyetDinh.KetQua as 'Kết Quả', QuyetDinh.ThamPhan as 'Thẩm Phán', QuyetDinh.Qu
40      }
41
42      public Tracuu()
43      {
44          InitializeComponent();
45          // this.label1.BackColor = System.Drawing.Color.Transparent;
46          this.label3.BackColor = System.Drawing.Color.Transparent;
47          this.label4.BackColor = System.Drawing.Color.Transparent;
48          this.label5.BackColor = System.Drawing.Color.Transparent;
49          this.label6.BackColor = System.Drawing.Color.Transparent;
50
51          this.cbTuoiVo.Items.Add("≤ 30");
52          this.cbTuoiVo.Items.Add("> 30");
53          this.cbTuoiVo.Items.Add("");
54
55          this.cbChenhLech.Items.Add("tuổi ≤ 5");
56          this.cbChenhLech.Items.Add("5 < tuổi ≤ 10");
57          this.cbChenhLech.Items.Add("tuổi > 10");
58          this.cbChenhLech.Items.Add("");
```

```
59
60              this.cbTuoiChong.Items.Add("≤ 30");
61              this.cbTuoiChong.Items.Add("> 30");
62              this.cbTuoiChong.Items.Add("");
63
64
65              this.cbQHPL.Items.Add("Mâu thuẫn gia đình");
66              this.cbQHPL.Items.Add("Yếu tố nước ngoài");
67              this.cbQHPL.Items.Add("Ngoại tình");
68              this.cbQHPL.Items.Add("Nghiện rượu bia, cờ bạc, ma túy");
69              this.cbQHPL.Items.Add("Bệnh tật, không có con");
70              this.cbQHPL.Items.Add("Một người mất tích");
71              this.cbQHPL.Items.Add("Mâu thuẫn kinh tế");
72              this.cbQHPL.Items.Add("Bạo lực gia đình");
73
74              this.cbKetQua.Items.Add("Xét xử");
75              this.cbKetQua.Items.Add("Hòa giải");
76              this.cbKetQua.Items.Add("");
77
78              this.dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
79              this.dgv.Font = new Font("Arial", 10);
80
81          }
82
83
84
85
86          private void button1_Click(object sender, EventArgs e)
87          {
88              string str_tuoivochong, str_qhpl, str_chenhlech, str_ketqua;
89              //Tuoi vo chong
90              if (cbTuoiVo.Text == "" && cbTuoiChong.Text == "")
91                  str_tuoivochong = "";
92              else if (cbTuoiVo.Text != "" && cbTuoiChong.Text == "")
93                  str_tuoivochong = " AND (" +timTuoiVo(cbTuoiVo.Text) +")";
94              else if (cbTuoiVo.Text == "" && cbTuoiChong.Text != "")
95                  str_tuoivochong = " AND (" + timTuoiChong(cbTuoiChong.Text) + ")";
96              else
97                  str_tuoivochong = " AND (" + timTuoiVC(cbTuoiVo.Text, cbTuoiChong.Text) + ")";
98
99              // Quan He PL
100             if (cbQHPL.Text == "")
101                 str_qhpl = "";
102             else
103                 str_qhpl = " and " + timQHPL(cbQHPL.Text);
```

```
104
105         //Chenh lech tuoi vo chong
106         if (cbChenhLech.Text == "")
107             str_chenhlech = "";
108         else
109             str_chenhlech = " and " + timChenhLech(cbChenhLech.Text);
110
111         //ket qua
112         if (cbKetQua.Text == "")
113             str_ketqua = "";
114         else
115             str_ketqua = " and " + timKetQua(cbKetQua.Text);
116
117         loaddata("select QuyetDinh.MaVuViec as 'Mã Vụ Việc', QuyetDinh.NgayThuLy as 'Ngày Thụ Lý', QuyetDinh.KetQua as 'Kết Quả', QuyetDinh.ThamPhan as 'Thẩm Phán', QuyetDinh.Qua
118
119     }
120
121
122     private string timQHPL(string p)
123     {
124         return "QuanHePhapLuat = N'"+p+ "'";
125     }
126
127     private string timChenhLech(string p)
128     {
129         if (p == "tuổi ≤ 5")
130             return "ABS(NguyenDon.Tuoi - BiDon.Tuoi) <= 5";
131         if (p == "5 < tuổi ≤ 10")
132             return "ABS(NguyenDon.Tuoi - BiDon.Tuoi) > 5 AND ABS(NguyenDon.Tuoi - BiDon.Tuoi) <= 10";
133         return "ABS(NguyenDon.Tuoi - BiDon.Tuoi) > 10";
134     }
135
136     private string timTuoiVo(string p)
137     {
138         if (p == "≤ 30")
139             return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi <= 30) OR (BiDon.Tuoi <= 30 AND BiDon.GioiTinh = N'Nữ')";
140         return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi > 30) OR (BiDon.Tuoi > 30 AND BiDon.GioiTinh = N'Nữ')";
141     }
142
143     private string timTuoiChong(string p)
144     {
145         if (p == "≤ 30")
146             return "(BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi <= 30) OR (NguyenDon.Tuoi <= 30 AND NguyenDon.GioiTinh = N'Nam')";
147         return "(BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi > 30) OR (NguyenDon.Tuoi > 30 AND NguyenDon.GioiTinh = N'Nam')";
148     }
```

```
149
150         private string timTuoiVC(string v, string c)
151         {
152             if (v == "≤ 30" && c == "≤ 30")
153                 return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi <= 30) AND (BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi <= 30)";
154             else if (v == "≤ 30" && c == "> 30")
155                 return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi <= 30) AND (BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi > 30)";
156             else if (v == "> 30" && c == "≤ 30")
157                 return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi > 30) AND (BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi <= 30)";
158             return "(NguyenDon.GioiTinh = N'Nữ' AND NguyenDon.Tuoi > 30) AND (BiDon.GioiTinh = N'Nam' AND BiDon.Tuoi > 30)";
159         }
160
161         private string timKetQua(string p)
162         {
163             if (p == "Xét xử")
164                 return "QuyetDinh.KetQua = N'Xét xử'";
165             return "QuyetDinh.KetQua = N'Hòa giải'";
166
167
168         }
169
170         private void btnReset_Click(object sender, EventArgs e)
171         {
172             cbTuoiVo.ResetText();
173             cbTuoiChong.ResetText();
174             cbQHPL.ResetText();
175             cbKetQua.ResetText();
176             cbChenhLech.ResetText();
177             loaddata("select QuyetDinh.MaVuViec as 'Mã Vụ Việc', QuyetDinh.NgayThuLy as 'Ngày Thụ Lý', QuyetDinh.KetQua as 'Kết Quả', QuyetDinh.ThamPhan as 'Thẩm Phán', QuyetDinh.Qu
178
179         }
180
181         private void button1_Click_1(object sender, EventArgs e)
182         {
183             Application.Exit();
184         }
185
186         private void button2_Click(object sender, EventArgs e)
187         {
188             this.Visible = false;
189             Trogiupraquyetdinh f2 = new Trogiupraquyetdinh();
190             f2.ShowDialog();
191         }
192
```