TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM
HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:
XÂY DỰNG HỆ THỐNG RA QUYẾT ĐỊNH THUÊ TRỌ BẰNG CÂY QUYẾT ĐỊNH

Sinh viên thực hiện:

Trần Tuấn Anh	2023600188
Dương Anh Tuấn	2023600101
Bùi Hoàng Thanh	2017604382
Hoàng Đức Toản	2024601130

Lớp: 20242IT6094010
Khóa: K18
Nhóm: 01

Giảng viên hướng dẫn: Ths. Mai Thanh Hồng

Hà Nội, 2025

MỤC LỤC

LỜI CẢM ƠN

Để hoàn thành được đề tài **"Xây dựng hệ thống ra quyết định thuê trọ bằng cây quyết định"** nhóm 01 xin bày tỏ lòng biết ơn đến giảng viên bộ môn "Trí tuệ nhân tạo" – Ths.Mai Thanh Hồng. Cô là người đã tận tình dạy dỗ và truyền đạt những kiến thức quý báu cho chúng em trong suốt học kỳ qua. Trong thời gian tham dự lớp học của cô, chúng em đã được tiếp cận với nhiều kiến thức bổ ích và rất cần thiết cho quá trình học tập, làm việc sau này của chúng em. Nhóm cũng chân thành cảm ơn đến các bạn trong lớp, trong quá trình học đã giúp đỡ và tạo điều kiện thuận lợi để nhóm hoàn thành bài báo cáo. Với khoảng thời gian không quá dài, nhóm chúng em đã chứng minh cho cô và các bạn thấy chúng em đã nỗ lực, cố gắng để hoàn thành thật tốt đề tài. Chúng em xin kính chúc cô thật nhiều sức khỏe để cống hiến nhiều hơn trong sự nghiệp giảng dạy.

Chúng em xin chân thành cảm ơn!

MỞ ĐẦU

Tên đề tài

" Xây dựng hệ thống ra quyết định thuê trọ bằng cây quyết định"

1. Lý do chọn đề tài.

Trong những năm gần đây Trí tuệ nhân tạo không ngừng phát triển một cách mạnh mẽ và hiện đại. Sự ra đời của các phát minh, nghiên cứu làm phong phú, đời sống con người nâng cao rõ rệt, đóng góp to lớn trong sự phát triển nhân loại. Gắn với thực trạng khó khăn về vấn đề thuê trọ của sinh viên, chúng em quyết định chọn đề tài " xây dựng hệ thống ra quyết định thuê trọ bằng cây quyết định" nhằm hỗ trợ đưa ra gợi ý quyết định cho người dùng và đồng thời cũng tìm hiểu sâu hơn về cây quyết định và cách ứng dụng của nó vào bài toán cụ thể.

2. Mục tiêu đề tài.

- Nắm bắt được kiến thức về cây quyết định.
- Áp dụng được thuật toán vào bài toán đặt ra.
- Đánh giá hiệu quả của thuật toán.

3. Bố cục đề tài.

Chia thành 3 chương:

Chương 1: Giới thiệu về bài toán.

Chương 2: Tìm hiểu về một số thuật toán để giải quyết bài toán.

Chương 3: Ứng dụng cây quyết định trong bài toán xây dựng hệ thống ra quyết định thuê trọ.

4. Phương pháp.

- Thu thập tài liệu, dữ liệu vào, phân tích, tìm hiểu, hiểu được các kiến thức cơ bản về cây quyết định và những kiến thức liên quan.
- Sử dụng các kiến thức đã tìm hiểu được và dữ liệu vào để tiến hành áp dụng vào bài toán xây dựng hệ thống ra quyết định thuê trọ.

5. Đối tượng và phạm vi nghiên cứu.

- Đối tượng: sinh viên hoặc những người đang có nhu cầu thuê trọ.
- Phạm vi: cây quyết định và Naive Bayes.

CHƯƠNG 1: GIỚI THIỆU VỀ TRÍ TUỆ NHÂN TẠO VÀ BÀI TOÁN.

1.1. Giới thiệu chung về Trí Tuệ Nhân Tạo.

1.1.1. Khái niệm về trí tuệ nhân tạo

Theo như cha đẻ của trí tuệ nhân tạo, John McCarthy thì nó là "Khoa học và kỹ thuật của việc tạo ra những máy thông minh, đặc biệt là chương trình máy tính thông minh".

Trí tuệ nhân tạo là hướng đi của việc tạo ra máy tính, người máy điều khiển bằng máy tính hay là những phần mềm suy nghĩ thông minh hơn, tương tự như suy nghĩ thông minh của con người.

Trí tuệ nhân tạo được học như bộ não con người, như cách mà con người học, quyết định và làm việc khi giải quyết một vấn đề, và sau đó sử dụng kết quả của quá trình học đó như là nền tảng của việc phát triển phần mềm và hệ thống thông minh.

Ở thời điểm hiện tại, Thuật ngữ này thường dùng để nói đến các MÁY TÍNH có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo. Tức là mỗi loại trí tuệ nhân tạo hiện nay đang dừng lại ở mức độ những máy tính hoặc siêu máy tính dùng để xử lý một loại công việc nào đó như điều khiển một ngôi nhà, nghiên cứu nhận diện hình ảnh, xử lý dữ liệu của bệnh nhân để đưa ra phác đồ điều trị, xử lý dữ liệu để tự học hỏi, khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty,...

Hình 1. 1. Minh họa AI là một bộ phận của khoa học máy tính

Nói nôm na cho dễ hiểu: đó là trí tuệ của máy móc được tạo ra bởi con người. Trí tuệ này có thể tư duy, suy nghĩ, học hỏi,... như trí tuệ con người. Xử lý dữ liệu ở mức rộng lớn hơn, quy mô hơn, hệ thống, khoa học và nhanh hơn so với con người.

Rất nhiều hãng công nghệ nổi tiếng có tham vọng tạo ra được những AI (trí tuệ nhân tạo) vì giá trị của chúng là vô cùng lớn, giải quyết được rất nhiều vấn đề của con người mà loài người đang chưa giải quyết được.

Trí tuệ nhân tạo mang lại rất nhiều giá trị cho cuộc sống loài người, nhưng cũng tiềm ẩn những nguy cơ. Rất nhiều chuyên gia lo lắng rằng khi trí tuệ nhân tạo đạt tới 1 ngưỡng tiến hóa nào đó thì đó cũng là thời điểm loài người bị tận diệt. Rất nhiều các bộ phim đã khai thác đề tài này với nhiều góc nhìn, nhưng qua đó đều muốn cảnh báo loài người về mối nguy đặc biệt này.

Hinh 1. 2. Một cảnh trong bộ phim "I, Robot" nói về một AI đã tiến hóa

Trí tuệ nhân tạo là một ngành khoa học và công nghệ dựa trên nền tảng của Khoa học máy tính, Sinh học, Triết học, Ngôn ngữ học, Toán học và Kỹ thuật. Một chuyên ngành chính của Trí tuệ nhân tạo là phát triển chức năng của máy tính kết hợp với sự thông minh của con người, chẳng hạn như suy luận, học hỏi và giải quyết vấn đề.

Trong những lĩnh vực dưới đây, một hoặc nhiều lĩnh vực có thể góp thành để xây dựng hệ thống thông minh.

1.1.2. Vai trò của trí tuệ nhân tạo

Vai trò của AI là vô tận đối với cuộc sống của chúng ta. AI có thể tiếp cận với con người thông qua nhiều lĩnh vực, ngành nghề khác nhau. Ưu điểm của trí tuệ nhân tạo AI là khả năng xử lý dữ liệu khoa học hơn, nhanh hơn, hệ thống hơn so với con người. Việc phát triển và đưa các sản phẩm AI tới tay người dùng đúng cách sẽ thúc đẩy mạnh mẽ sự phát triển của toàn nhân loại. Mở ra một thế giới hoàn toàn mới cùng các giải pháp bù đắp cho những vấn đề mà con người không thể giải quyết.

Vai trò của trí tuệ nhân tạo trong y học

Hình 1. 3. Minh họa áp dụng AI trong y khoa

Công nghệ AI đã mở ra một trang mới cho nền y học thế giới, đặc biệt là nền y học nước nhà. Nó mang đến cho con người những giá trị đáng kinh ngạc trong việc bảo vệ sức khỏe và điều trị bệnh tật. Tại lĩnh vực này, trí tuệ nhân tạo có vai trò quan trọng trong việc hỗ trợ điều trị y tế như định lượng thuốc, các phương pháp điều trị khác nhau cho bệnh nhân và quy trình phẫu thuật trong phòng mổ. Chúng sử dụng những thuật toán phân tích để hỗ trợ bệnh nhân theo dõi kết quả điều trị 24/7.

Vai trò của AI trong tài chính

Hình 1. 4. Minh họa áp dụng AI trong lĩnh vực tài chính

Ngoài việc hỗ trợ con người chăm sóc sức khỏe, AI còn có vai trò quan trọng trong ngành tài chính ngân hàng. AI là công cụ giúp con người xử lý các hoạt động trong ngân hàng như xử lý giao dịch, theo dõi số dư, quản lý tài sản và các tài khoản tiền gửi lớn một cách nhanh chóng và chính xác nhất. Trí tuệ nhân tạo không những giúp các ngân hàng hợp lý hóa giao dịch mà còn có thể ước tính cung, cầu và định giá chứng khoán một cách dễ dàng hơn.

Vai trò của AI trong trò chơi và công nghệ

Hiện nay, những tập đoàn lớn đang ngày càng thúc đẩy việc sử dụng máy móc thông minh vào dây chuyền sản xuất. AI được sử dụng như các robot có thể thay thế một phần công việc của con người. Khối lượng công việc và thời gian hoàn thành sẽ nhanh chóng và nhẹ nhàng hơn dưới sự hoạt động của máy móc tích hợp trí tuệ nhân tạo. Tiêu biểu là với các sản phẩm như ô tô tự lái và trò chơi điện tử. Trong trò chơi điện tử, trí tuệ nhân tạo AI sẽ tự phân tích các hành vi và đưa ra những đáp án không kém cạnh với trí tuệ con người. Với ô tô tự lái, hệ thống AI tính toán tất cả các dữ liệu bên trong động cơ, tìm hiểu cách đi và ngăn chặn va chạm bởi chướng ngại vật

Sự kết hợp hoàn hảo của AI và robot hút bụi

Hinh 1. 5. Áp dụng AI để sản xuất robot hút bụi

Khi mọi người nghe đến trí tuệ nhân tạo, điều đầu tiên họ thường nghĩ đến là robot. Đối với lĩnh vực dọn dẹp tự động hóa gia đình, AI là điều không thể thiếu. Kết hợp các công nghệ tiên tiến cùng công nghệ AI siêu thông minh, các dòng máy robot hút bụi tự động liên tục được ra mắt trên thị trường. Tiêu biểu là dòng robot hút bụi Roomba của iRobot. Các sản phẩm tích hợp AI thường là những công cụ cao cấp nhất, đem lại hiệu quả cực lớn trong việc làm sạch sàn nhà của các hộ gia đình.

Với thời đại công nghệ 4.0 hiện nay, việc ứng dụng AI không còn xa lạ gì với cuộc sống của chúng ta. Trí tuệ nhân tạo có mặt trong mọi lĩnh vực đời sống từ giải trí cho đến y tế, xã hội. Đây chính là chìa khóa để mở ra một thế hệ mới đầy văn minh, thúc đẩy sự phát triển to lớn của loài người.

So sánh giữa lập trình không có TTNT và lập trình có TTNT

Lập trình không có TTNT	Lập trình có TTNT
Chương trình máy tính mà không có Trí tuệ nhân tạo thì chỉ có thể trả lời những câu hỏi xác định được quy định sẵn để giải quyết vấn đề.	Chương trình máy tính mà có Trí tuệ nhân tạo thì có thể trả lời những câu hỏi chung, cùng loại để giải quyết vấn đề.

Chỉnh sửa chương trình dẫn đến sự thay đổi trong cấu trúc của nó.	Chương trình Trí tuệ nhân tạo có thể tiếp thu sự ập nhật cái mới bằng cách đề cao tính độc lập của những thông tin với nhau. Vì vậy bạn có thể sửa đổi một phần thông tin trong chương trình mà không làm ảnh hưởng đến cấu trúc của nó.
Việc chỉnh sửa thường không nhanh và không dễ dàng. Nó dẫn đến việc ảnh hưởng chương trình của bạn	Chỉnh sửa chương trình nhanh và dễ dàng.

Những tác động của TTNT đến sản xuất trong nền công nghiệp 4.0 như sau:

● Chất lượng – Năng suất dự đoán : Vai trò của trí tuệ nhân tạo đầu tiên là giảm thiểu các hao tổn trong sản xuất và ngăn ngừa các quy trình sản xuất kém hiệu quả. Khi nhu cầu ngày càng tăng để đáp ứng sự cạnh tranh thì trí tuệ nhân tạo là điều vô cùng cần thiết.

● Bảo trì sự dự đoán : Một trong những lợi ích của trí tuệ nhân tạo nữa là bảo trì dự đoán. Thay vì việc bảo trì theo lịch trình định trước thì bảo trì dự đoán sẽ sử dụng thuật toán để dự đoán lỗi tiếp theo của một bộ phận/máy móc/hệ thống. Nhờ đó có thể cảnh báo nhân viên thực hiện các quy trình bảo trì tập trung để ngăn chặn sự cố. Bảo trì dự đoán có ưu điểm là giảm đáng kể chi phí trong khi loại bỏ nhu cầu về thời gian ngừng hoạt động theo kế hoạch trong nhiều trường hợp. Ngoài ra, nhờ nó mà Tuổi thọ hữu dụng còn lại của máy móc và thiết bị lâu hơn.

● Kết hợp giữa robot và con người

Hình 1.6. Minh họa Robot thay thế con người trong một số công việc

Tính đến năm 2020, ước tính có khoảng 1,64 triệu robot công nghiệp đang hoạt động trên toàn thế giới. Robot sản xuất được chấp thuận làm việc cùng với con người để tăng năng suất công việc.

Khi áp dụng robot ngày càng nhiều thì AI sẽ đóng một vai trò quan trọng trong việc đảm bảo an toàn cho con người. Đồng thời trao cho robot nhiều trách nhiệm hơn trong việc đưa ra các quyết định có thể tối ưu hóa các quy trình dựa trên dữ liệu thời gian thực được thu thập từ sàn sản xuất.

● Thiết kế sáng tạo : Nhà sản xuất có thể tận dụng trí tuệ nhân tạo vào giai đoạn thiết kế. Khi có bản tóm tắt thiết kế được xác định rõ rành làm đầu vào thì các nhà kỹ sư, thiết kế có thể sử dụng thuật toán al. Mục đích để khám phá tất cả các cấu hình có thể có của một giải pháp.

● Nhu cầu cung ứng thị trường : Vai trò của trí tuệ nhân tạo cuối cùng mà chúng tôi muốn nhắc đến là cung ứng thị trường. Hiện nay trí tuệ nhân tạo đang hiện hữu ở mọi nơi trong hệ sinh thái công nghiệp 4.0. Nhà sản xuất có thể sử dụng các thuật toán AI để tối ưu hóa chuỗi cung ứng của các hoạt động sản xuất. Đồng thời giúp họ phản ứng và dự đoán tốt hơn những thay đổi trên thị trường.

1.1.3. Kĩ thuật trong TTNT là gì và một số kĩ thuật cơ bản trong TTNT

Trong thế giới thực, Tri thức có một vài thuộc tính như sau:

- Dung lượng đồ sộ, phi thường.
- Tổ chức tốt, định dạng tốt.
- Luôn luôn cập nhật sự thay đổi.

Kỹ thuật Trí tuệ nhân tạo là một cách để tổ chức và sử dụng tri thức có hiệu quả trong những cách sau đây:

- Có thể nhận thức được người đã cung cấp cho nó.
- Có thể sửa đổi dễ dàng để sửa lỗi.
- Nó có thể hữu ích trong một số tình huống dù nó chưa hoàn thiện hoặc chưa chính xác lắm.

Kỹ thuật Trí tuệ nhân tạo nâng cao tốc độ thực thi của những chương trình phức tạp.

Một số kỹ thuật Trí tuệ nhân tạo cơ bản :

- Lý thuyết giải bài toán và suy diễn thông minh
- Lý thuyết tìm kiếm may rủi
- Các ngôn ngữ về TTNT
- Lý thuyết thể hiện tri thức và hệ chuyên gia
- Lý thuyết nhận dạng và xử lý tiếng nói
- Người máy
- Tâm lý học xử lý thông tin
- Xử lý danh sách, kỹ thuật đệ quy, kỹ thuật quay lui và xử lý cú pháp hình thức

1.1.4. Lịch sử phát triển của trí tuệ nhân tạo

Đây là lịch sử của Trí tuệ nhân tạo trong suốt thế kỷ XX.

Năm	Cột mốc/ Phát minh
1923	Vở kịch khoa học viễn tưởng của Karel Capek tên là "Rossum's Universal Robots" (RUR) diễn ra tại Luân Đôn (nước Anh). Lần đầu tiên sử dụng từ "robot" trong tiếng Anh.
1943	Nền tảng của mạng thần kinh được đặt nền móng.
1945	Isaac Asimov, một cựu sinh viên trường Đại học Columbia, đưa ra thuật ngữ "Robotics"
1950	Alan Turing giới thiệu Bài kiểm tra Turing để đánh giá sự thông minh và công bố Máy thông minh và Sự thông minh. Claude Shannon công bố "Phân tích chi tiết của việc chơi cờ".
1956	John McCarthy đưa ra thuật ngữ Trí tuệ nhân tạo. Biểu diễn chạy chương trình trí tuệ nhân tạo đầu tiên tại trường Đại học Carnegie Mellon.
1958	John McCarthy sáng tạo ra LISP, ngôn ngữ lập trình cho trí tuệ nhân tạo.
1964	Bài luận văn của Danny Bobrow tại MIT cho thấy máy tính có thể hiểu được ngôn ngữ tự nhiên của con người.
1965	Joseph Weizenbaum tại MIT đã xây dựng ELIZA, một vấn đề tương tác được mang trong đoạn đối thoại Tiếng Anh.
1969	Cá nhà khoa học tại Viện nghiên cứu Stanford đã phát triển Shakey, một robot, được trang bị sự vận động, nhận thức, và giải quyết vấn đề.
1973	Các nhóm hội về người máy tại Đại học Edinburgh đã xây dựng Freddy. Một người máy Scotland nổi tiếng, có khả năng sử dụng thị giác để định vị và lắp ráp mô hình.
1979	Xe tự quản được điều khiển bằng máy tính đầu tiên được xây dựng. Đó là

	Stanford Cart.
1985	Harold Cohen tạo và trình diễn chương trình đồ họa mang tên Aaron.
1985	Những chuyên đề nâng cao trong tất cả các lĩnh vực của Trí tuệ nhân tạo là: ● Có tính chất quan trọng trong "học máy". ● Suy luận theo tình huống ● Lên lịch trình ● Khai thác dữ liệu, thu thập web ● Hiểu và dịch ngôn ngữ tự nhiên của con người ● Thị giác và thực tế ảo ● Ứng dụng trong trò chơi
1997	Chương trình "Deep Blue Chess" đánh bại nhà vô địch cờ thế giới, Garry Kasparov.
2000	Những robot thú cưng có sự tương tác đã được thương mại hóa. MIT đã trình diễn Kismet - một robot có khuôn mặt có thể biểu lộ cảm xúc. robot Nomad khám phá những vùng xa xôi hẻo lánh của Nam Cực và xác định thiên thạch.

1.1.5. Các thành phần trong hệ thống của trí tuệ nhân tạo

Hệ thống trí tuệ nhân tạo bao gồm hai thành phần cơ bản đó là biểu diễn tri thức và tìm kiếm tri thức trong miền biểu diễn:

TTNT = Tri thức + Suy diễn

Tri thức của bài toán có thể được phân ra làm ba loại cơ bản đó là tri thức mô tả, tri thức thủ tục và tri thức điều khiển.

Để biểu diễn tri thức người ta sử dụng các phương pháp sau đây:

- Phương pháp biểu diễn nhờ luật
- Phương pháp biểu diễn nhờ mạng ngữ nghĩa
- Phương pháp biểu diễn nhờ bộ ba liên hợp OAV

● Phương pháp biểu diễn nhờ Frame

● Phương pháp biểu diễn nhờ logic vị tư

Sau khi tri thức của bài toán đã được biểu diễn, kỹ thuật trong lĩnh vực trí tuệ nhân tạo là các phương pháp tìm kiếm trong miền đặc trưng tri thức về bài toán đó. Với mỗi cách biểu diễn sẽ có các giải pháp tương ứng. Các vấn đề này sẽ được đề cập trong chương 3.

1.1.6. Các lĩnh vực nghiên cứu và ứng dụng cơ bản

Trí tuệ nhân tạo có những ảnh hưởng vượt trội trong nhiều lĩnh vực như:

● Game - Trí tuệ nhân tạo đóng vai trò cốt yếu trong những game chiến lược như cờ, đánh bài, tic-tac-toe (như cờ caro), ... nơi mà máy móc có thể suy nghĩ số lớn những trường hợp có khả năng xảy ra dựa trên tri thức.

● Xử lý ngôn ngữ tự nhiên - Nó có khả năng tương tác với máy tính, hiểu ngôn ngữ tự nhiên mà con người nói.

● Hệ thống chuyên môn hóa - Có một vài ứng dụng mà các máy móc thông minh, phần mềm và những thông tin đặc biệt để suy luận. Nó giải thích và đưa ra lời khuyên cho người dùng hệ thống đó.

● Hệ thống thị giác - Hệ thống có thể hiểu, phân tích và tiếp thu dữ liệu vào thuộc về thị giác ngay trên máy tính. Ví dụ như:

○ Những máy bay do thám chụp lại hình ảnh, sau đó sử dụng kỹ thuật này để mô hình hóa những thông tin không gian hay bản đồ của khu vực.

○ Bác sĩ sử dụng hệ thống buồng bệnh chuyên môn để chẩn đoán cho bệnh nhân.

○ Cảnh sát có thể sử dụng phần mềm máy tính để nhận diện khuôn mặt của tội phạm từ những hình chân dung được vẽ lại bởi những họa sĩ pháp y.

● Nhận diện lời nói - Một vài hệ thống thông minh có khả năng nghe và tiếp thu ngôn ngữ trong cấu trúc và nghĩa của câu trong khi con người nói. Nó có thể

nắm bắt được độ nhấn mạnh khác nhau, từ lóng, tiếng ồn phía sau, sự thay đổi trong âm thanh của con người do trời lạnh, ...

● Nhận diện chữ viết tay - Phần mềm nhận diện chữ viết tay đọc văn bản được viết trên giấy bằng bút hoặc viết trên màn hình bằng bút cảm ứng. Nó nhận dạng được hình dạng của chữ và chuyển nó thành văn bản có thể chỉnh sửa được.

● Người máy thông minh - Người máy có khả năng thực hiện nhiệm vụ mà con người giao cho. Nó có các cảm biến để nhận dạng các dữ liệu vật lý trong thế giới thực như ánh sáng, hơi nóng, nhiệt độ, sự di chuyển, âm thanh, sự va chạm và áp lực. Nó được trang bị bộ xử lý hiệu quả, đa cảm biến và bộ nhớ lớn để thể hiện sự thông minh. Hơn thế nữa, nó có khả năng học từ lỗi sai của nó và thích nghi với môi trường mới.

1.2. Giới thiệu về bài toán.

1.2.1. Giới thiệu.

- Xây dựng hệ thống ra quyết định thuê trọ là một giải pháp công nghệ nhằm hỗ trợ người tìm trọ trong quá trình tìm kiếm và đưa ra quyết định thuê. Hệ thống này sử dụng cây quyết định và mô hình quyết định để phân tích thông tin và đưa ra gợi ý, giúp người dùng chọn lựa trọ phù hợp với nhu cầu và yêu cầu của mình.

- Trong trường hợp này, mô hình cây quyết định sẽ xây dựng dựa trên dữ liệu thu thập được về các yếu tố quan trọng trong việc thuê trọ. Các yếu tố này có thể bao gồm: giá cả, vị trí, khoảng cách, diện tích, chất lượng, tiện nghi, an ninh, có chung chủ hay không và các yếu tố khác mà người tìm trọ quan tâm.

- Qua quá trình phân tích dữ liệu, cây quyết định sẽ xác định các quy tắc và tìm ra tiêu chí ưu tiên. Khi người dùng cung cấp thông tin về yêu cầu và tiêu chí của mình, hệ thống sẽ áp dụng mô hình câu quyết định để đưa ra các gợi ý và lựa chọn phù hợp.

- Để xây dựng một hệ thống ra quyết định thuê trọ hiệu quả, cần có dữ liệu đầu vào đáng tin cậy và đủ lớn để đảm bảo tính chính xác của quyết định đưa ra.

1.2.2. Bài toán.

Bài toán xây dựng hệ thống ra quyết định thuê trọ được giải quyết bằng thuật toán "Cây quyết định" và thuật toán " Naive Bayes". Bằng cách sử dụng hai thuật toán trên kết hợp với ngôn ngữ Python và Google Colab để giải quyết được bài toán. Thu thập dữ liệu đầu vào là một file csv gồm nhiều trường như khoảng cách, giá cả, dịch vụ, tiện nghi, giờ giấc, an ninh, … và cột khác nhau. Đầu ra là hai quyết định "có" hay "không".

1.2.3. Thu thập dữ liệu đầu vào.

Để giải quyết bài toán trên ta cần phải thu thập dữ liệu đầu vào là một bảng dữ liệu dạng file "csv" gồm các thuộc tính như sau:

- **Khoảng cách**

  Xác định khoảng cách thuê trọ, gần hay xa so với trường học, nơi làm việc, siêu thị, chợ, khu trung tâm,...

- **Giá cả**

  Thu thập thông tin về giá cả bao gồm: giá phòng, tiền điện, tiền nước, tiền internet, tiền dịch vụ và một số chi phí khác liên quan.

- **Diện tích**

  Diện tích của căn phòng trọ, bao gồm diện tích sử dụng và sử dụng chung nếu có.

- **Tiện nghi**

  Kiểm tra và ghi nhận các tiện nghi có sẵn trong căn phòng trọ như giường, tủ, bàn, ghế, quạt, máy lạnh, máy giặt, bếp, tủ lạnh…

- **Chất lượng**

  Đánh giá chất lượng căn phòng trọ, bao gồm sự sạch sẽ, tình trạng sửa chữa, vệ sinh, thông gió, ánh sáng tự nhiên…

- **Giờ giấc**

  Thu thập thông tin về giới hạn giờ giấc, có hạn chế hoặc không, quy định về việc ra vào căn phòng trọ trong thời gian nào…

- **An ninh**

  Đánh giá mức độ an ninh của khu vực và căn phòng trọ, bao gồm hệ thống an ninh, cửa khóa, camera giám sát, quy định về an ninh…

1.2.4. Dữ liệu ra.

Dữ liệu đầu ra của hệ thống ra quyết định thuê trọ là “Có” hay “Không”.

CHƯƠNG 2: TÌM HIỂU VỀ MỘT SỐ THUẬT TOÁN ĐỂ GIẢI QUYẾT BÀI TOÁN

2.1 Cây quyết định

Cây quyết định được dùng để đưa ra tập luật if – then nhằm mục đích dự báo, giúp con người nhận biết về tập dữ liệu. Cây quyết định cho phép phân loại đối tượng tùy thuộc vào các điều kiện tại các nút trong cây, bắt đầu từ gốc cây tới các nút sát lá-Nút xác định phân loại đối tượng. Mỗi nút trong của cây xác định điều kiện đối với thuộc tính mô tả của đối tượng. Mỗi nhánh tương ứng với điều kiện: Nút (thuộc tính) bằng giá trị nào đó. Đối tượng được phân loại nhờ tích hợp các điều kiện bắt đầu từ nút gốc của cây và các thuộc tính mô tả với giá trị của thuộc tính đối tượng.

2.1.1 Ưu và nhược điểm của cây quyết định

Ưu điểm

Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi bởi những lợi ích của nó:

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

Nhược điểm

Kèm với đó, cây quyết định cũng có những nhược điểm cụ thể:

- Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
- Cây quyết định hay gặp vấn đề overfitting.

2.1.2 Tạo cây quyết định

Xét bảng dữ liệu T = (A, D) trong đó A = {A_1, A_2,..., A_n } là tập thuộc tính dẫn xuất, D = {r_1, r_2, ..., r_n} là thuộc tính mục tiêu. Vấn đề đặt ra là trong tập thuộc tính A

ta phải chọn thuộc tính nào để phân hoạch? Một trong các phương pháp đó là dựa vào độ lợi thông tin. Hay còn gọi là thuật giải ID3.

Lựa chọn chủ yếu trong giải thuật ID3 là chọn thuộc tính nào để đưa vào mỗi nút trong cây. Ta sẽ chọn thuộc tính phân rã tập mẫu tốt nhất. Thước đo độ tốt của việc chọn lựa thuộc tính là gì? Ta cần xác định một độ đo thống kê, gọi là thông tin thu được, đánh giá từng thuộc tính được chọn tốt như thế nào còn phụ thuộc vào việc phân loại mục tiêu của tập mẫu. ID3 sử dụng thông tin thu được đánh giá để chọn ra thuộc tính cho mỗi bước giữa những thuộc tính ứng viên, trong quá trình phát triển cây.

$$Entropy(s) = \sum_{i=1}^{c} - p_i log_2 p_i$$

Để đánh giá chính xác thông tin thu được, dùng Entropy(S): Độ bất định (độ pha trộn/độ hỗn tạp) của S liên quan đến sự phân loại đang xét

Trong đó p_i là xác suất xuất hiện trạng thái i của hệ thống. Theo lý thuyết thông tin: mã có độ dài tối ưu là mã gán $- log_2 p$ bits cho thông điệp có xác suất là p. S là một tập huấn luyện.

Nếu gọi p là xác suất xuất hiện các ví dụ dương trong tập S, p là xác suất xuất hiện các ví dụ âm trong tập S. Entropy đo độ bất định của tập S sẽ là:

$$\text{Entropy (S)} = -p\ log_2 p\ - p\ log_2 p$$

Quy định 0.log 0 = 0

Chẳng hạn với tập S gồm 14 mẫu có chung một vài giá trị logic gồm 9 mẫu dương và 5 mẫu âm. Khi đó đại lượng Entropy của tập S liên quan đến sự phân loại logic này là:

Entropy([9+, 5-]) = - (9/14) log_2 (9/14) - (5/14) log_2 (5/14) = 0,940

<u>Chú ý :</u>

Đại lượng Entropy = 0 nếu tất cả thành viên của tập S cùng thuộc một lớp (vì nếu tất cả là dương (P+ = 1), do đó P- = 0, Entropy(S) = $-1log_2 1 - 0log_2 0 = 0$).Đại

lượng Entropy(S) = 1 khi tập S chứa tỷ lệ tập mẫu âm và mẫu dương là như nhau. Nếu tập S chứa tập mẫu âm và tập mẫu dương có tỉ lệ P+ khác P- thì Entropy(S) ∈ (0,1).Dựa trên sự xác định entropy, ta tính Gain(S, A) = Lượng giảm entropy mong đợi qua việc chia các ví dụ theo thuộc tính A

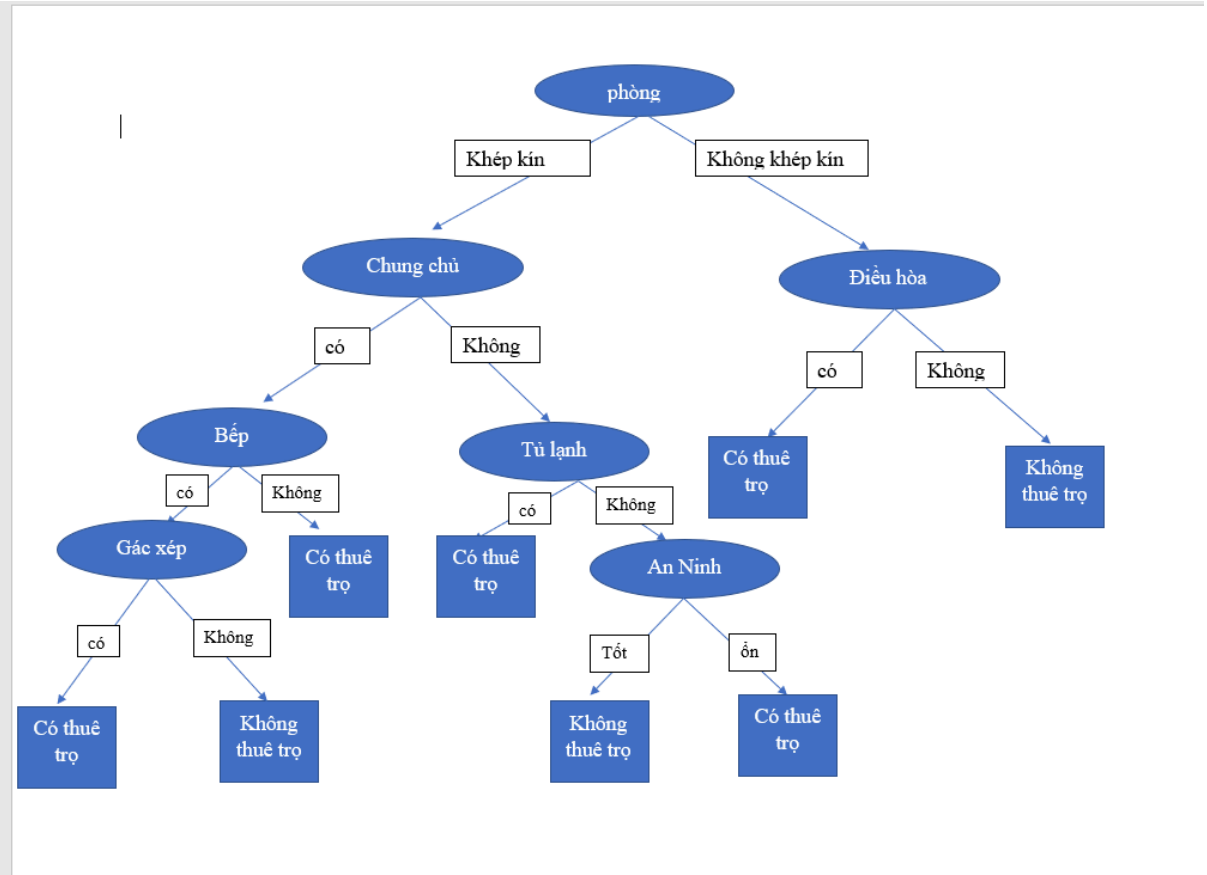
$$\text{Gain(S, A)} = \text{Entropy(S)} - \sum_{values(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

- Thuật toán cây quyết định với bài toán tìm nhà trọ:
  + Bảng huấn luyện:

D(1)	Phòng	chung chủ	điều hòa	Bếp	Tủ lạnh	Gác xép	an ninh	thuê trọ
D1(2)	Khép kín	có	có	có	có	có	tốt	có
D2(3)	Khép kín	có	không	không	không	có	ổn	có
D3(4)	không khép kín	có	có	không	không	không	ổn	có
D4(5)	Khép kín	không	không	không	không	không	tốt	không
D5(6)	không khép kín	không	có	không	không	không	ổn	có
D6(7)	Khép kín	có	không	có	không	không	tốt	có
D7(8)	Khép kín	không	có	có	có	không	tốt	có
D8(9)	Khép kín	không	có	không	không	không	ổn	không

D9	không khép kín	có	không	không	không	không	ổn	không

+ Cây quyết định:



2.2. Thuật toán phân lớp Naive bayes

Bộ phân lớp Bayes là một giải thuật thuộc lớp giải thuật thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Phân lớp Bayes được dựa trên định lý Bayes (định lý được đặt theo tên tác giả của nó là Thomas Bayes)

2.3.1. Định lý Bayes

- Gọi A, B là 2 biến cố:

Với P(B) > 0:

$$P(A|\ B) = P(AB)/P(B)$$

Suy ra:

$$P(AB) = P(A|\ B)\ P(B) = P(B|A)P(A)$$

Công thức Bayes:

$$P(B|A) = P(AB)/P(A) = (P(A|B)P(B)) / P(A)$$

$$= (P(A|B)P(B)) / (P(AB) + P(A\underline{B})$$

$$= (P(A|\ B)\ P(B)) / (P(A|\ B)\ P(B) + P(A|\underline{B})P(\underline{B}))$$

- Công thức Bayes tổng quát

Với P(A) > 0 và {B1, B2, …, Bn} là một hệ đầy đủ các biến cố:

+ Tổng xác suất của hệ bằng 1:

$$\sum_{k=1}^{n} p(Bk) = 1$$

+ Từng đôi một xung khắc:

$$P(Bi \cap Bj) = 0$$

+ Khi đó ta có:

$$P(Bk|A) = \frac{P(A|Bk)P(Bk)}{\sum_{I=1}^{n} P(A|Bi)P(Bi)}$$

Trong đó ta gọi A là một chứng cứ (evidence) (trong bài toán phân lớp A sẽ là một phần tử dữ liệu), B là một giả thiết nào để cho A thuộc về một lớp C nào đó. Trong bài toán phân lớp chúng ta muốn xác định giá trị P(B/A) là xác suất để giả thiết B là đúng với chứng cứ A thuộc vào lớp C với điều kiện ra đã biết các thông tin mô tả A. P(B|A) là một xác suất hậu nghiệm (posterior probability hay posteriori probability) của B với điều kiện A.

Giả sử tập dữ liệu liệu khách hàng của chúng ta được mô tả bởi các thuộc tính tuổi và thu nhập, và một khách hàng X có tuổi là 25 và thu nhập là 2000$. Giả sử H là giả thiết khách hàng đó sẽ mua máy tính, thì P(H|X) phản ánh xác suất người dùng X sẽ mua máy tính với điều kiện ta biết tuổi và thu nhập của người đó.

Ngược lại P(H) là xác suất tiền nghiệm (prior probability hay priori probability) của H. Trong ví dụ trên, nó là xác suất một khách hàng sẽ mua máy tính mà không cần biết các thông tin về tuổi hay thu nhập của họ. Hay nói cách khác, xác suất này không phụ thuộc vào yếu tố X. Tương tự, P(X|H) là xác suất của X với điều kiện H (likelihood), nó là một xác suất hậu nghiệm. VÍ dụ, nó là xác suất người dùng X (có tuổi là 25 và thu nhập là $200) sẽ mua máy tính với điều kiện ta đã biết người đó sẽ

mua máy tính. Cuối cùng P(X) là xác suất tiền nghiệm của X. Trong ví dụ trên, nó sẽ là xác suất một người trong tập dữ liệu sẽ có tuổi 25 và thu nhập $2000.

Posterior = Likelihood * Prior / Evidence

2.3.2. Phân loại Naive Bayes

Naive Bayes là một thuật toán phân loại cho các vấn đề phân loại nhị phân (hai lớp) và đa lớp. Kỹ thuật này dễ hiểu nhất khi được mô tả bằng các giá trị đầu vào nhị phân hoặc phân loại. Thuật toán Naive Bayes tính xác suất cho các yếu tố, sau đó chọn kết quả với xác suất cao nhất.

Tuy nhiên, ta cần lưu ý giả định của thuật toán Naive Bayes là các yếu tố đầu vào được cho là độc lập với nhau.

Thuật toán này là một thuật toán mạnh mẽ trong các bài toán:

- Dự đoán với thời gian thực
- Phân loại Text/ Lọc thư rác
- Hệ thống Recommendation

Về mặt toán học, ta có thể viết như sau:

Nếu ta có một Class E và các điểm dữ liệu x1,x2,x3, etc.

Đầu tiên ta sẽ phải tính xác suất P(x1| E) , P(x2 | E) … (xác suất của x1 thuộc class E xảy ra) và sau đó ta sẽ chọn class có xác suất xảy ra x1 cao nhất.

CHƯƠNG 3: ỨNG DỤNG CÂY QUYẾT ĐỊNH TRONG XÂY DỰNG HỆ THỐNG RA QUYẾT ĐỊNH THUÊ TRỌ

3.1. Lựa chọn, demo thuật toán

Từ chương 2 ta đã thấy rõ được ưu, nhược điểm của cây quyết định cũng như Naive Bayes. Tuy nhiên, với sự phổ biến, đơn giản và thuận tiện khi cài đặt chương trình, nhóm sẽ ứng dụng cây quyết định để giải quyết bài toán trên.

Để dựng được cây quyết định, nhóm mình dựa vào giải thuật ID3 tính toán độ bất định (Entropy) của tập thuộc tính và xác định giá trị thông tin thu được cho mỗi thuộc tính (Information Gain) với bảng dữ liệu như sau:

Bảng 3.1. Dữ liệu demo

Nguoi	Phong	Chung chu	Dieu hoa	Nong lanh	Tu lanh	Bep	Tu quan ao	Quyet dinh thue
N1	khep kin	co	co	co	co	co	tu go	co
N2	khep kin	co	khong	co	khong	khong	tu go	co
N3	khong khep kin	co	co	khong	khong	khong	tu nhua	khong
N4	khep kin	khong	khong	co	khong	khong	tu nhua	khong
N5	khong khep kin	khong	co	khong	khong	khong	tu vai	co
N6	khep kin	co	khong	co	khong	co	tu go	khong
N7	khep kin	khong	co	co	co	co	tu go	co
N8	khep kin	khong	co	co	khong	khong	khong	co
N9	khong khep kin	co	khong	khong	khong	khong	khong	khong
N10	khep kin	khong	co	co	khong	khong	tu nhua	co

Từ bảng dữ liệu trên, ta thấy tập S gồm 10 mẫu, trong đó có 6 mẫu dương và 4 mẫu âm, áp dụng công thức *Entropy(S)* = $- p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$, ta tính được đại lượng Entropy của tập S là:

Entropy([6+, 4-]) = - (6/10)$\log_2$(6/10) - (4/10)$\log_2$(4/10) = 0.971

Tiếp theo ta xác định giá trị thông tin thu được cho mỗi thuộc tính là:

+ Xét thuộc tính *Phong*:

Gain(S, Phong) = Entropy(S) - $\sum \frac{|Sv|}{|S|}$ Entropy(S_v)

= 0.971 - ($\frac{7}{10}$ Entropy($S_{khep\ kin}$) + $\frac{3}{10}$ Entropy($S_{khong\ khep\ kin}$))

= 0.971 - ($\frac{7}{10}$ (- $\frac{5}{7}$ $\log_2(\frac{5}{7})$ - $\frac{2}{7}$ $\log_2(\frac{2}{7})$) + $\frac{3}{10}$ (- $\frac{1}{3}$ $\log_2(\frac{1}{3})$ - $\frac{2}{3}$ $\log_2(\frac{2}{3})$))

= 0.0913

+ Xét thuộc tính *Chung chu*:

Gain(S, Chung chu) = 0.971 - ($\frac{5}{10}$ Entropy(S_{co}) + $\frac{5}{10}$ Entropy(S_{khong}))

= 0.971 - ($\frac{5}{10}$ (- $\frac{2}{5}$ $\log_2(\frac{2}{5})$ - $\frac{3}{5}$ $\log_2(\frac{3}{5})$) + $\frac{5}{10}$ (- $\frac{4}{5}$ $\log_2(\frac{4}{5})$ - $\frac{1}{5}$ $\log_2(\frac{1}{5})$))

= 0.125

+ Xét thuộc tính *Dieu hoa*:

Gain(S, Dieu hoa) = 0.971 - ($\frac{6}{10}$ Entropy(S_{co}) + $\frac{4}{10}$ Entropy(S_{khong}))

= 0.971 - ($\frac{6}{10}$ (- $\frac{5}{6}$ $\log_2(\frac{5}{6})$ - $\frac{1}{6}$ $\log_2(\frac{1}{6})$) + $\frac{4}{10}$ (- $\frac{1}{4}$ $\log_2(\frac{1}{4})$ - $\frac{3}{4}$ $\log_2(\frac{3}{4})$))

= 0.256

+ Xét thuộc tính *Nong lanh*:

Gain(S, Nong lanh) = 0.971 - ($\frac{7}{10}$ Entropy(S_{co}) + $\frac{3}{10}$ Entropy(S_{khong}))

= 0.971 - ($\frac{7}{10}$ (- $\frac{5}{7}$ $\log_2(\frac{5}{7})$ - $\frac{2}{7}$ $\log_2(\frac{2}{7})$) + $\frac{3}{10}$ (- $\frac{1}{3}$ $\log_2(\frac{1}{3})$ - $\frac{2}{3}$ $\log_2(\frac{2}{3})$))

= 0.09133

+ Xét thuộc tính *Tu lanh*:

Gain(S, Tu lanh) = 0.971 - ($\frac{2}{10}$ Entropy(S_{co}) + $\frac{8}{10}$ Entropy(S_{khong}))

= 0.971 - ($\frac{2}{10}$ 0 + $\frac{8}{10}$ (- $\frac{4}{8}$ $\log_2(\frac{4}{8})$ - $\frac{4}{8}$ $\log_2(\frac{4}{8})$))

= 0.171

+ Xét thuộc tính *Bep*:

Gain(S, Bep) = 0.971 - ($\frac{3}{10}$ Entropy(S_{co}) + $\frac{7}{10}$ Entropy(S_{khong}))

$= 0.971 - (\frac{3}{10} (- \frac{2}{3} \log_2(\frac{2}{3}) - \frac{1}{3} \log_2(\frac{1}{3})) + \frac{7}{10} (- \frac{4}{7} \log_2(\frac{4}{7}) - \frac{3}{7} \log_2(\frac{3}{7})))$

$= 0.00585$

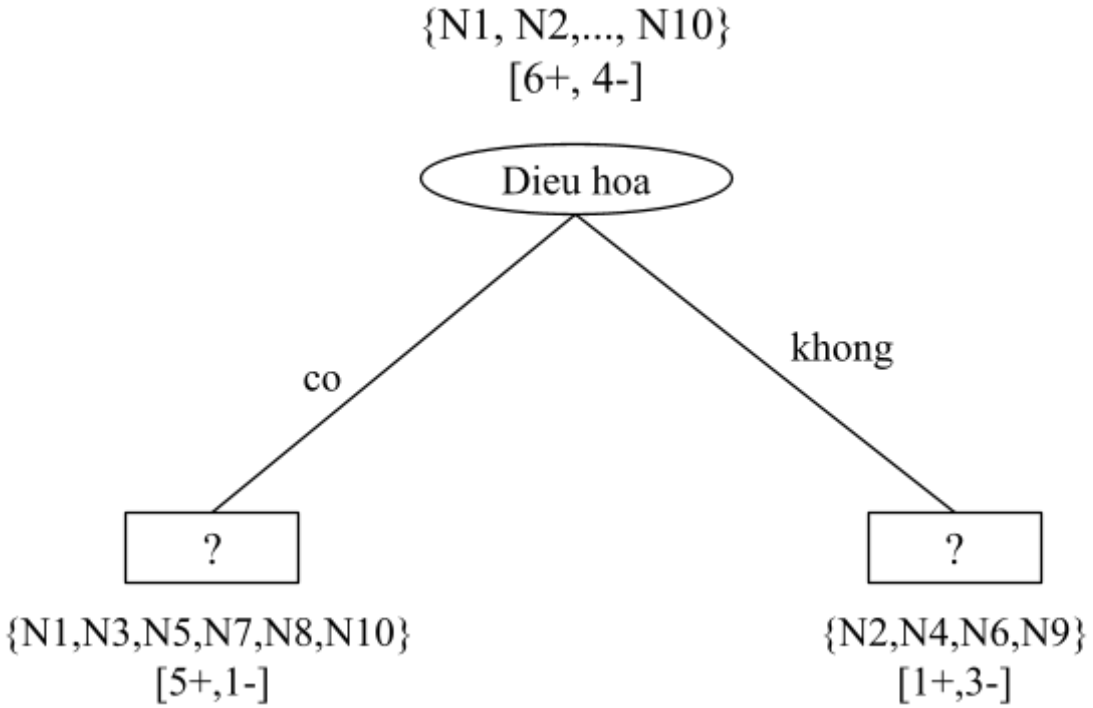
+ Xét thuộc tính *Tu quan ao*:

$\text{Gain(S, Tu quan ao)} = 0.971 - (\frac{4}{10} \text{Entropy}(S_{\text{tu go}}) + \frac{3}{10} \text{Entropy}(S_{\text{tu nhua}}) + \frac{1}{10} \text{Entropy}(S_{\text{tu vai}}) + \frac{2}{10} \text{Entropy}(S_{\text{khong}}))$

$= 0.971 - (\frac{4}{10} (- \frac{3}{4} \log_2(\frac{3}{4}) - \frac{1}{4} \log_2(\frac{1}{4})) + \frac{3}{10} (- \frac{1}{3} \log_2(\frac{1}{3}) - \frac{2}{3} \log_2(\frac{2}{3})) + \frac{1}{10} 0 + \frac{2}{10} (- \frac{1}{2} \log_2(\frac{1}{2}) - \frac{1}{2} \log_2(\frac{1}{2})))$

$= 0.171$

Theo đánh giá thông tin thu được, thuộc tính *Dieu hoa* cung cấp dự đoán tốt nhất về thuộc tính mục tiêu "Quyet dinh thue" trên tập mẫu. Do đó, thuộc tính *Dieu hoa* được chọn là thuộc tính quyết định cho nút gốc, nhánh được tạo ra dưới nút gốc tương ứng với mỗi giá trị của thuộc tính *Dieu hoa* cùng với tập mẫu sẽ thêm vào mỗi nút con mới. Với những nhánh con tương ứng với Dieu hoa = "co" và Dieu hoa = "khong" có giá trị Entropy $\neq 0$ nên quá trình học cây được tiếp tục với dữ liệu con tại nhánh đó.



Hình 3.1. Cây quyết định sau lần phân hạch đầu tiên

Trước hết, ta đi xây dựng nhánh Dieu hoa = "co"

Bảng 3.2. Tập dữ liệu con (S1) tại nhánh Dieu hoa = "co"

Nguoi	Phong	Chung chu	Nong lanh	Tu lanh	Bep	Tu quan ao	Quyet dinh thue
N1	khep kin	co	co	co	co	tu go	co
N3	khong khep kin	co	khong	khong	khong	tu nhua	khong
N5	khong khep kin	khong	khong	khong	khong	tu vai	co
N7	khep kin	khong	co	co	co	tu go	co
N8	khep kin	khong	co	khong	khong	khong	co
N10	khep kin	khong	co	khong	khong	tu nhua	co

Tương tự, ta tính độ bất định cho tập dữ liệu con (S1):

$$\text{Entropy}([5+, 1-]) = -\frac{5}{6}\log_2(\frac{5}{6}) - \frac{1}{6}\log_2(\frac{1}{6}) = 0.65$$

$$\text{Gain}(S1, \text{Phong}) = 0.65 - (\frac{4}{6}\text{Entropy}(S1_{\text{khep kin}}) + \frac{2}{6}\text{Entropy}(S1_{\text{khong khep kin}}))$$

$$= 0.65 - (\frac{4}{6}0 + \frac{2}{6}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})))$$

$$= 0.31667$$

$$\text{Gain}(S1, \text{Chung chu}) = 0.65 - (\frac{2}{6}\text{Entropy}(S1_{\text{co}}) + \frac{4}{6}\text{Entropy}(S1_{\text{khong}}))$$

$$= 0.65 - (\frac{2}{6}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) + \frac{4}{6}0)$$

$$= 0.31667$$

$$\text{Gain}(S1, \text{Nong lanh}) = 0.65 - (\frac{4}{6}\text{Entropy}(S1_{\text{co}}) + \frac{2}{6}\text{Entropy}(S1_{\text{khong}}))$$

$$= 0.65 - (\frac{4}{6}0 + \frac{2}{6}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})))$$

$$= 0.31667$$

$$\text{Gain}(S1, \text{Tu lanh}) = 0.65 - (\frac{2}{6}\text{Entropy}(S1_{\text{co}}) + \frac{4}{6}\text{Entropy}(S1_{\text{khong}}))$$

$$= 0.65 - (\frac{2}{6}0 + \frac{4}{6}(-\frac{3}{4}\log_2(\frac{3}{4}) - \frac{1}{4}\log_2(\frac{1}{4})))$$

$$= 0.10915$$

Gain(S1, Bep) = 0.65 - ($\frac{2}{6}$ Entropy($S1_{co}$) + $\frac{4}{6}$ Entropy($S1_{khong}$))

= 0.65 - ($\frac{2}{6}$ 0 + $\frac{4}{6}$ (- $\frac{3}{4}$ $\log_2(\frac{3}{4})$ - $\frac{1}{4}$ $\log_2(\frac{1}{4})$))

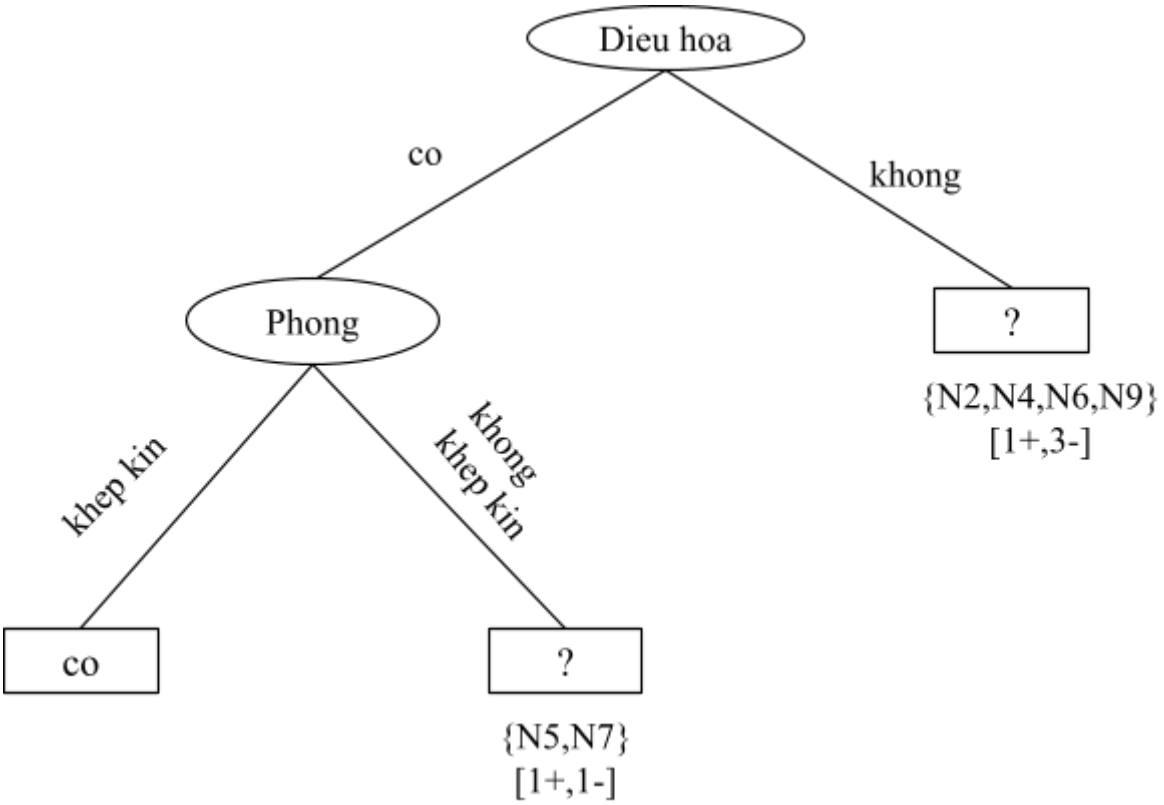
= 0.10915

Gain(S1, Tu quan ao) = 0.65 - ($\frac{2}{6}$ Entropy($S1_{tu\ go}$) + $\frac{2}{6}$ Entropy($S1_{tu\ nhua}$) + $\frac{1}{6}$ Entropy($S1_{tu\ vai}$) + $\frac{1}{6}$ Entropy($S1_{khong}$))

= 0.65 - ($\frac{2}{6}$ 0 + $\frac{2}{6}$ (- $\frac{1}{2}$ $\log_2(\frac{1}{2})$ - $\frac{1}{2}$ $\log_2(\frac{1}{2})$) + $\frac{1}{6}$ 0 + $\frac{1}{6}$ 0)

= 0.31667

Ta chọn thuộc tính *Phong* làm nút con. Mọi mẫu mà có *Phong* = "khep kin" thì là mẫu dương với thuộc tính *Quyet dinh thue*. Do vậy, nút này trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "co". Với nút con tương ứng với *Phong* = "khong khep kin" sẽ tiếp tục phát triển.



Bảng 3.3. Tập dữ liệu con (S1') tại nhánh Phong = "khong khep kin"

Nguoi	Chung chu	Nong lanh	Tu lanh	Bep	Tu quan ao	Quyet dinh thue
N3	co	khong	khong	khong	tu nhua	khong
N5	khong	khong	khong	khong	tu vai	co

Entropy(S1') = $-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1$

Gain(S1', Chung chu) = $1 - (\frac{1}{2}\text{Entropy}(S1'_{co}) + \frac{1}{2}\text{Entropy}(S1'_{khong}))$

$= 1 - (\frac{1}{2}0 + \frac{1}{2}0)$

$= 1$

Gain(S1', Nong lanh) = $1 - \text{Entropy}(S1'_{khong})$

$= 1 - (-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}))$

$= 0$

Gain(S1', Bep) = $1 - \text{Entropy}(S1'_{khong})$

$= 1 - (-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}))$

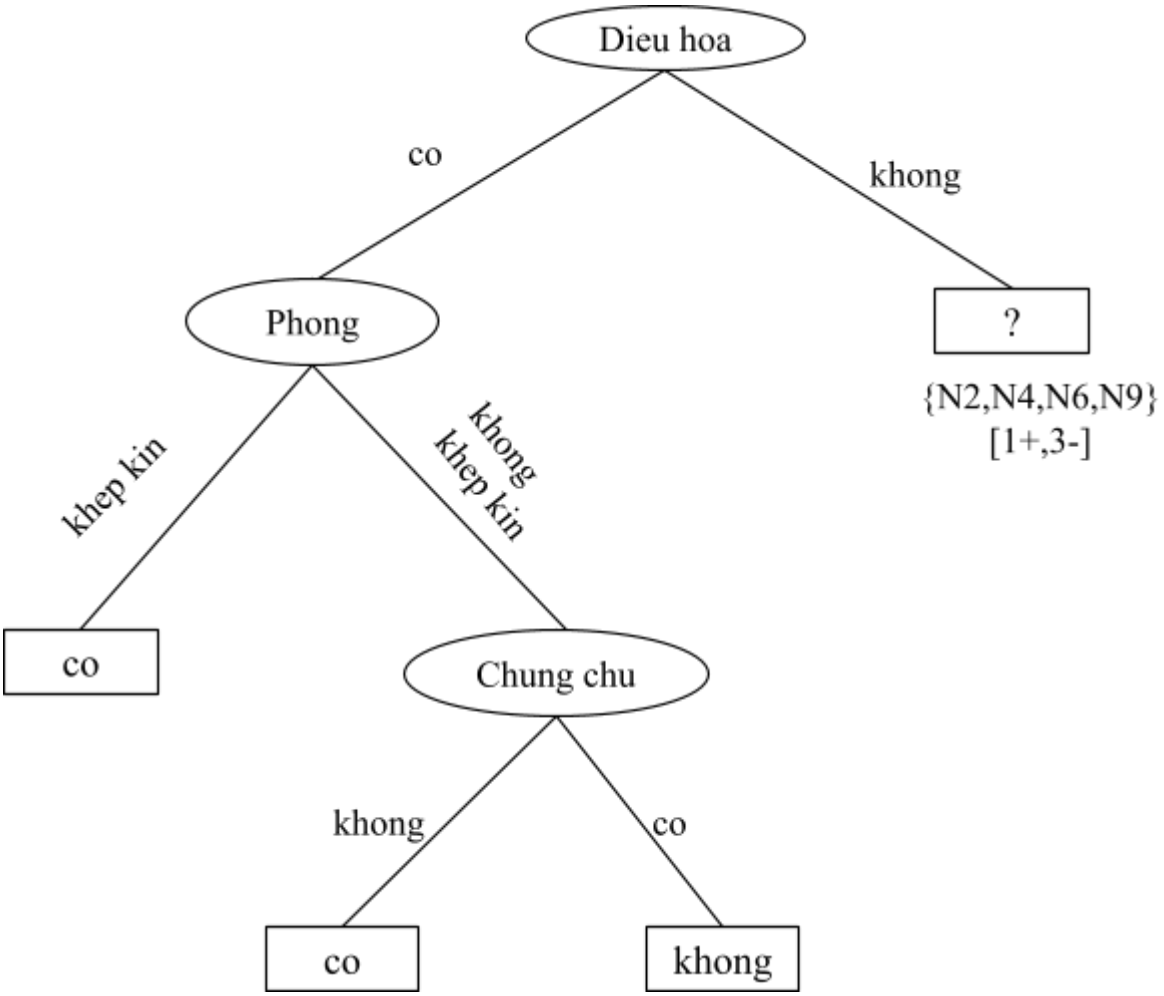
$= 0$

Gain(S1', Tu quan ao) = $1 - (\frac{1}{2}\text{Entropy}(S1'_{tu\ nhua}) + \frac{1}{2}\text{Entropy}(S1'_{tu\ vai}))$

$= 1 - (\frac{1}{2}0 + \frac{1}{2}0)$

$= 1$

Ta chọn thuộc tính *Chung chu* làm nút con tiếp theo. Mọi mẫu mà có *Chung chu* = "khong" thì là mẫu dương với thuộc tính *Quyet dinh thue*. Do vậy, nút này trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "co". Ngược lại, mẫu mà có *Chung chu* = "co" trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "khong".



Tương tự như cách dựng nhánh Dieu hoa = "co", ta đi xây dựng nhánh *Dieu hoa* = "khong" như sau:

Bảng 3.4. Tập dữ liệu con (S2) tại nhánh Dieu hoa = "khong"

Nguoi	Phong	Chung chu	Nong lanh	Tu lanh	Bep	Tu quan ao	Quyet dinh thue
N2	khep kin	co	co	khong	khong	tu go	co
N4	khep kin	khong	co	khong	khong	tu nhua	khong
N6	khep kin	co	co	khong	co	tu go	khong
N9	khong khep kin	co	khong	khong	khong	khong	khong

Entropy(S2) = $-\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4}) = 0.8113$

Gain(S2, Phong) = $0.8113 - (\frac{3}{4}\text{Entropy}(S2_{\text{khep kin}}) + \frac{1}{4}\text{Entropy}(S2_{\text{khong khep kin}}))$

$= 0.8113 - (\frac{3}{4}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) + \frac{1}{4}0))$

$= 0.1226$

Gain(S2, Chung chu) = $0.8113 - (\frac{3}{4}\text{Entropy}(S2_{\text{co}}) + \frac{1}{4}\text{Entropy}(S2_{\text{khong}}))$

$= 0.8113 - (\frac{3}{4}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) + \frac{1}{4}0))$

$= 0.1226$

Gain(S2, Nong lanh) = $0.8113 - (\frac{3}{4}\text{Entropy}(S2_{\text{co}}) + \frac{1}{4}\text{Entropy}(S2_{\text{khong}}))$

$= 0.8113 - (\frac{3}{4}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) + \frac{1}{4}0))$

$= 0.1226$

Gain(S2, Tu lanh) = $0.8113 - \text{Entropy}(S2_{\text{khong}})$

$= 0.8113 - (-\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4}))$

$= 0$

Gain(S2, Bep) = $0.8113 - (\frac{1}{4}\text{Entropy}(S2_{\text{co}}) + \frac{3}{4}\text{Entropy}(S2_{\text{khong}}))$

$= 0.8113 - (\frac{1}{4}0 + \frac{3}{4}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})))$

$= 0.1226$

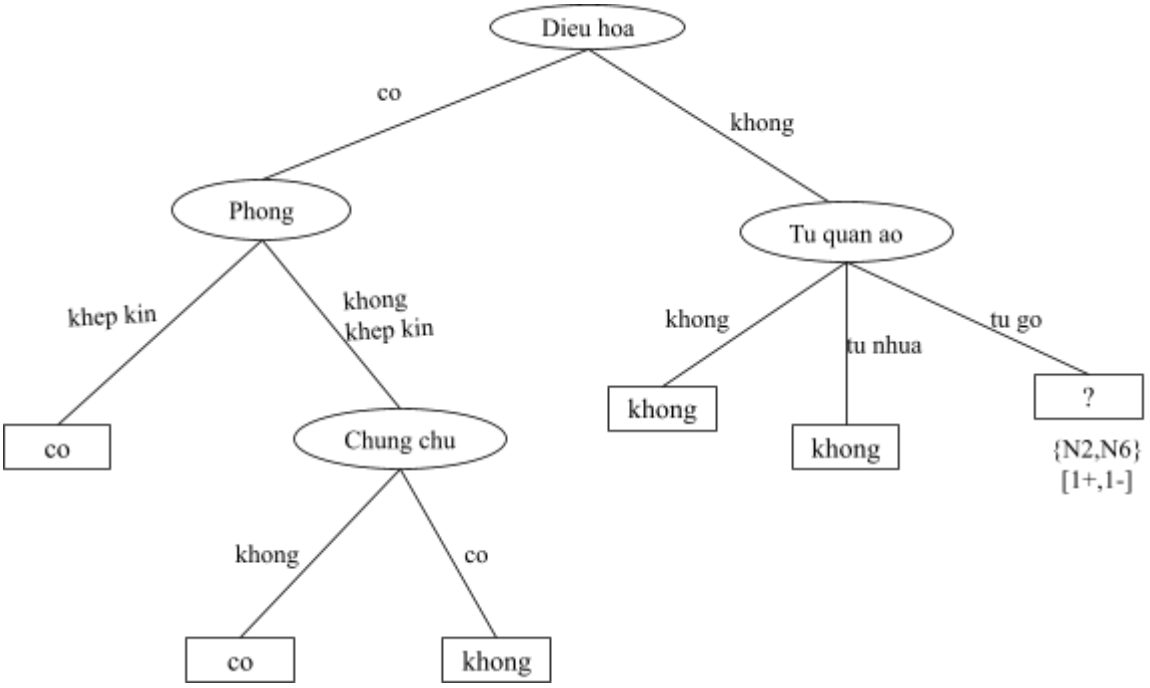
Gain(S2, Tu quan ao) = $0.8113 - (\frac{2}{4}\text{Entropy}(S1_{\text{tu go}}) + \frac{1}{4}\text{Entropy}(S2_{\text{tu nhua}})$

$+ \frac{1}{4}\text{Entropy}(S2_{\text{khong}}))$

$= 0.8113 - (\frac{2}{4}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) + \frac{1}{4}0 + \frac{1}{4}0)$

$= 0.3113$

Ta chọn thuộc tính *Tu quan ao* làm nút con. Mọi mẫu mà có *Tu quan ao* = "tu nhua" và *Tu quan ao* = "khong" thì là mẫu âm với thuộc tính *Quyet dinh thue*. Do vậy, các nút này trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "khong". Với nút con tương ứng với *Tu quan ao* = "tu go" sẽ tiếp tục phát triển.



Bảng 3.5. Dữ liệu con (S2') tại nhánh Tu quan ao = "tu go"

Nguoi	Phong	Chung chu	Nong lanh	Tu lanh	Bep	Quyet dinh thue
N2	khep kin	co	co	khong	khong	co
N6	khep kin	co	co	khong	co	khong

Entropy(S2') = $-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1$

Gain(S2', Phong) = 1 - Entropy($S2'_{\text{khep kin}}$)

= $1 - (-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}))$

= 0

Gain(S2', Chung chu) = 1 - Entropy($S2'_{\text{co}}$)

= $1 - (-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}))$

= 0

Gain(S2', Nong lanh) = 1 - Entropy($S2'_{co}$)

= 1 - (- $\frac{1}{2}$ $\log_2(\frac{1}{2})$ - $\frac{1}{2}$ $\log_2(\frac{1}{2})$)

= 0

Gain(S2', Tu lanh) = 1 - Entropy($S2'_{khong}$)

= 1 - (- $\frac{1}{2}$ $\log_2(\frac{1}{2})$ - $\frac{1}{2}$ $\log_2(\frac{1}{2})$)

= 0

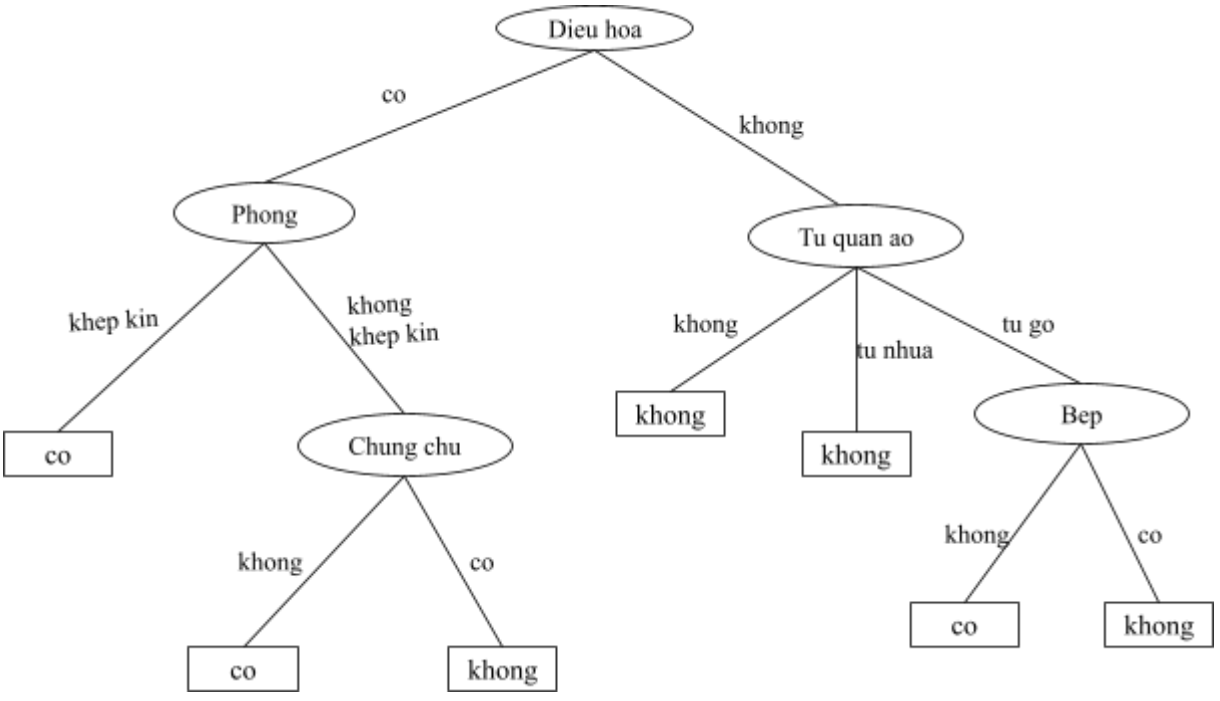
Gain(S2', Bep) = 1 - ($\frac{1}{2}$ Entropy($S2'_{co}$) + $\frac{1}{2}$ Entropy($S2'_{khong}$))

= 1 - ($\frac{1}{2}$ 0 + $\frac{1}{2}$ 0)

= 1

Ta chọn thuộc tính *Bep* làm nút con tiếp theo. Mọi mẫu mà có *Bep* = "khong" thì là mẫu dương với thuộc tính *Quyet dinh thue*. Do vậy, nút này trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "co". Ngược lại, mẫu mà có *Bep* = "co" trở thành nút lá với sự phân loại thuộc tính *Quyet dinh thue* = "khong".

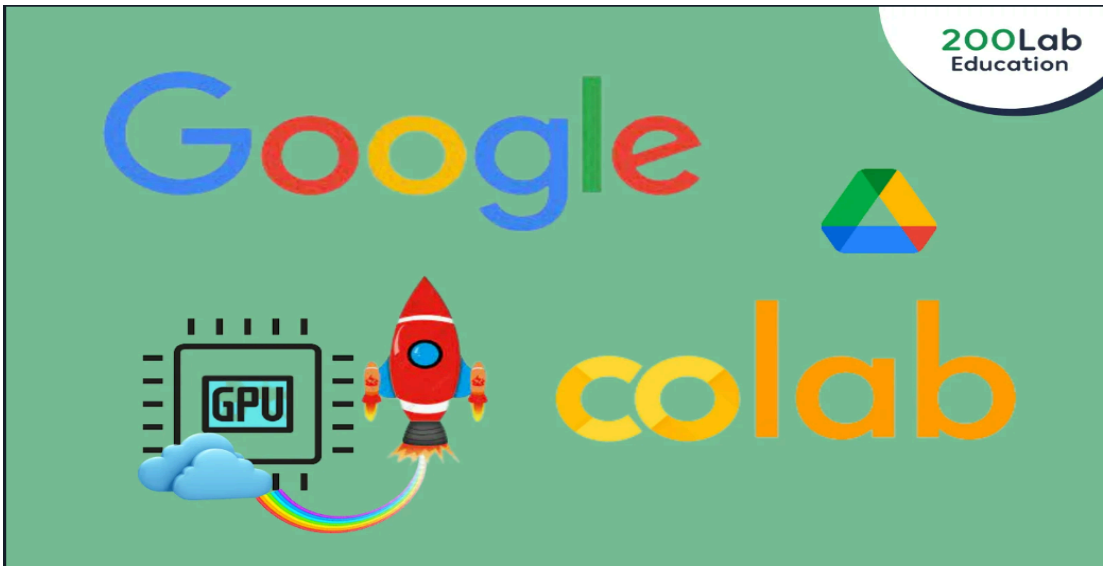
Như vậy, ta đã dựng xong cây quyết định:



Hình 3.2. Cây quyết định

3.2. Công cụ, phương tiện hỗ trợ

3.2.1. Google Colab



Colaboratory hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép thực thi Python trên nền tảng đám mây, đặc biệt phù hợp với Data analysis, machine learning và giáo dục.

Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, người dùng có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được đã được cung cấp.

Sử dụng Google Colab có những lợi ích ưu việt như: sẵn sàng chạy Python ở bất kỳ thiết bị nào có kết nối Internet mà không cần cài đặt, chia sẻ và làm việc nhóm dễ dàng, sử dụng miễn phí GPU cho các dự án về AI. Ngoài ra, Colab cũng cung cấp nhiều thư viện machine learning được cài đặt sẵn như Keras, Pytorch, Tensorflow. Mọi thứ sẽ được lưu trữ trong cục bộ máy khi bạn lựa chọn Jupyter Notebook làm môi trường làm việc. Nếu người dùng đề cao quyền riêng tư thì đây chắc chắn là một tính năng ưa thích.

3.2.1. File csv

File CSV là file giá trị được phân tách bằng dấu phẩy, nó chứa các tập dữ liệu văn bản thuần túy có thể chứa số, chữ cái và cấu trúc dữ liệu được phân tách bằng dấu phẩy. Với mỗi dòng trong file CSV sẽ đại diện cho một hàng cơ sở dữ liệu mới và mỗi hàng cơ sở dữ liệu sẽ bao gồm một hoặc nhiều trường được phân tách bằng dấu phẩy.

Các file được lưu bằng định dạng file CSV thường được sử dụng để trao đổi dữ liệu giữa các ứng dụng khác nhau. Ví dụ bạn có thể lưu thông tin liên hệ từ Microsoft Excel dưới dạng file CSV và nhập thông tin đó vào sổ địa chỉ trong Microsoft Outlook.

Các chương trình cơ sở dữ liệu, phần mềm phân tích và các ứng dụng khác lưu trữ lượng lớn thông tin (như danh bạ và dữ liệu khách hàng) thường sẽ hỗ trợ định dạng này.

CSV cũng là viết tắt của Computer Software Validation, Comma-Separated Variable, Circuit Switched Voice hoặc Colon-Separated Value. Nhưng cho dù mọi người có gọi file CSV là gì thì như cách gọi trên thì chúng đều nói về cùng một định dạng.

3.3. Cài đặt chương trình

3.3.1. Dữ liệu

- Đầu vào: Bộ dữ liệu khảo sát từ sinh viên về nhu cầu thuê trọ được lưu trong file csv. Các tiêu chí thuê như là kiểu phòng, các tiện nghi, an ninh, dịch vụ,... Cột cuối cùng thể hiện ý định thuê (có, không).
- Đầu ra: Đưa ra gợi ý quyết định có thuê hoặc không

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Phong	Chung chu	Dieu hoa	Nong lanh	Quat	Tu lanh	Bep	Tu quan a	Giuong	Gac xep	May giat	Khoang cach	An ninh	Dien tich	Tien dien	Tien nuoc	Tien rac/	Gia phong	Quyet dinh thue	
khep kin	co	co	co	quat tran	co	co	tu go	giuong doi	co	co	~1	tot	20m2	3k	20k	20k	3tr8	co	
khep kin	co	khong	co	quat tran	khong	khong	tu go	giuong don	co	khong	~1	on	15m2	4k	20k	20k	2tr5	co	
khong khep	co	co	khong	quat cay	khong	khong	tu nhua	giuong don	khong	khong	~0.5	on	8m2	3k	15k	15k	1tr	khong	
khep kin	khong	khong	co	quat treo tuong	khong	khong	tu nhua	giuong don	khong	khong	~1.5	tot	10m2	3.5k	12k	20k	1tr	khong	
khong khep	khong	co	khong	khong	khong	khong	tu vai	giuong don	khong	khong	~0.5	on	7m2	2.5k	15k	15k	700k	co	
khep kin	co	khong	co	quat cay	khong	co	tu go	giuong tang	khong	co	~3	tot	15m2	3k	25k	20k	2tr5	khong	
khep kin	khong	co	co	quat treo tuong	co	co	tu go	giuong doi	khong	co	~2	tot	25m2	3.5k	25k	20k	3tr5	co	
khep kin	khong	co	co	quat tran	khong	khong	khong	giuong don	khong	khong	~1	on	10m2	2.5k	15k	15k	1tr2	co	
khong khep	co	khong	khong	quat treo tuong	khong	khong	khong	giuong don	khong	khong	~1.5	on	8m2	2.5k	12k	12k	700k	khong	
khep kin	khong	co	co	khong	khong	khong	tu nhua	giuong don	co	co	~2.5	tot	15m2	3.5k	15k	20k	2tr	co	
khep kin	co	khong	co	khong	co	khong	tu nhua	giuong doi	khong	khong	~2	te	18m2	3k	15k	15k	1tr2	khong	
khep kin	khong	co	co	quat cay	co	co	khong	giuong don	co	khong	~1.5	te	20m2	2.5k	20k	15k	2tr8	co	
khong khep	co	co	khong	quat treo tuong	khong	khong	khong	giuong doi	khong	co	~2.5	tot	10m2	3k	12k	15k	1tr	khong	
khong khep	khong	co	co	quat tran	khong	co	tu nhua	giuong doi	khong	co	~1	on	18m2	3k	15k	20k	2tr	co	
khep kin	co	co	co	khong	co	co	tu go	giuong tang	khong	co	~0.5	te	20m2	3k	12k	12k	3tr5	khong	
khong khep	khong	co	co	quat cay	khong	khong	tu vai	giuong doi	khong	khong	~1	on	10m2	2.5k	15k	12k	800k	co	
khong khep	co	co	co	quat treo tuong	khong	co	tu go	giuong doi	khong	khong	~1.5	tot	25m2	3k	20k	15k	2tr8	khong	
khep kin	co	co	co	khong	co	co	tu nhua	giuong tang	co	khong	~4	tot	18m2	4k	20k	20k	2tr8	khong	
khep kin	khong	co	co	khong	co	co	tu go	giuong don	co	khong	~3	tot	18m2	3.5k	20k	10k	2tr5	co	
khep kin	khong	co	co	quat cay	co	co	tu go	giuong tang	co	co	~1	te	25m2	3k	25k	25k	4tr	khong	
khep kin	khong	co	co	quat treo tuong	khong	co	tu nhua	giuong doi	khong	co	~0.5	on	20m2	4k	20k	25k	3tr6	co	
khong khep	khong	co	khong	quat cay	khong	khong	tu go	giuong doi	khong	khong	~1	on	7m2	2.5k	15k	15k	1tr	co	
khong khep	khong	khong	co	quat tran	khong	khong	tu vai	giuong doi	khong	co	~0.5	tot	10m2	3.5k	15k	20k	1tr	khong	
khep kin	co	co	co	khong	khong	khong	tu go	giuong tang	khong	co	~5	tot	18m2	3k	20k	15k	1tr8	khong	

Hình 3.7. Bộ dữ liệu training

Phong	Chung chu	Dieu hoa	Nong lanh	Quat	Tu lanh	Bep	Tu quan a	Giuong	Gac xep	May giat	Khoang cach	An ninh	Dien tich	Tien dien	Tien nuoc	Tien rac/	Gia phong	Quyet dinh thue
khep kin	co	co	co	quat tran	khong	khong	tu go	giuong don	co	khong	~4	on	18m2	2.5k	15k	25k	2tr2	co
khong khep	khong	co	khong	khong	khong	khong	khong	giuong don	khong	khong	~2	te	7m2	2.5k	15k	12k	700k	khong
khong khep	khong	co	co	khong	khong	khong	khong	giuong don	khong	khong	~3	tot	10m2	4k	12k	15k	1tr5	khong
khep kin	khong	co	co	quat cay	co	co	tu nhua	giuong doi	co	khong	~3	on	30m2	3k	20k	20k	4tr5	khong
khep kin	khong	co	khong	quat tran	khong	co	tu vai	giuong doi	co	co	~1.5	on	20m2	3k	15k	12k	2tr	co
khong khep	co	co	co	quat treo tuong	khong	co	tu go	giuong doi	khong	co	~1	tot	12m2	3k	15k	15k	2tr	co
khep kin	khong	co	co	quat cay	co	co	tu go	giuong don	co	co	~2	on	25m2	2.5k	20k	15k	3tr2	co
khong khep	khong	khong	khong	khong	khong	khong	khong	giuong don	khong	khong	~0.5	on	7m2	3k	20k	15k	600k	co
khep kin	khong	co	co	khong	khong	co	tu nhua	giuong doi	co	co	~1.5	tot	20m2	2.5k	20k	20k	2tr8	co
khep kin	co	co	co	quat tran	khong	co	tu go	giuong tang	co	khong	~1	te	25m2	4k	20k	25k	3tr5	khong
khep kin	khong	co	co	quat cay	co	co	tu nhua	giuong tang	co	khong	~5	on	25m2	3.5k	20k	20k	3tr6	khong
khep kin	co	co	co	quat treo tuong	khong	co	tu nhua	giuong doi	khong	khong	~5	on	25m2	2.5k	15k	15k	3tr2	co
khep kin	khong	co	co	quat cay	khong	co	khong	giuong doi	co	khong	~1.5	tot	15m2	3k	15k	15k	1tr5	co
khep kin	khong	khong	co	quat cay	co	co	tu go	giuong doi	khong	khong	~1	on	18m2	3k	20k	15k	1tr5	co
khong khep	co	co	co	quat treo tuong	khong	khong	tu nhua	giuong doi	khong	khong	~0.5	tot	12m2	3k	12k	15k	1tr5	khong
khep kin	khong	khong	co	khong	co	co	tu go	giuong tang	khong	co	~2	on	15m2	3.5k	20k	25k	2tr2	khong
khep kin	khong	co	co	quat tran	co	co	tu go	giuong tang	co	co	~1.5	tot	30m2	3k	20k	20k	4tr5	co
khong khep	co	co	co	quat treo tuong	co	co	tu nhua	giuong doi	co	khong	~1	tot	15m2	3k	15k	15k	2tr	khong
khep kin	co	co	co	quat tran	co	co	tu go	giuong don	co	co	~3	on	25m2	3.5k	20k	20k	4tr	co
khep kin	khong	co	co	quat cay	khong	co	tu nhua	giuong doi	khong	co	~1.5	on	25m2	4k	20k	25k	4tr	khong
khong khep	khong	co	co	quat cay	khong	co	khong	giuong tang	khong	khong	~1	tot	10m2	3k	15k	12k	1tr2	khong
khong khep	khong	khong	co	quat treo tuong	khong	co	tu vai	giuong doi	khong	khong	~0.5	tot	12m2	2.5k	15k	15k	800k	khong

Hình 3.8. Bộ dữ liệu training (tiếp)

3.3.2. Chương trình

Bước 1: Nhập các thư viện cần thiết

```
[1]  from IPython import get_ipython
     from IPython.display import display
     from google.colab import files
     import pandas as pd
     import numpy as np
```

Hình 3.9. Mã nguồn nhập các thư viện cần thiết

- from IPython import get_ipython, from IPython.display import display: Các thư viện này thường được sử dụng trong môi trường notebook như Colab hoặc Jupyter để tương tác với môi trường hiển thị và kernel.
- from google.colab import files: Thư viện này dành riêng cho Google Colab, cho phép tải file lên từ máy cục bộ của người dùng.
- import pandas as pd: Thư viện phổ biến để làm việc với dữ liệu dạng bảng (DataFrame).
- import numpy as np: Thư viện cho các phép tính toán số học, đặc biệt là cho hàm logarit cần thiết trong tính toán entropy và information gain.

Bước 2: Tải dữ liệu training.

```
print("Vui lòng tải file:")
uploaded = files.upload()
```

```
Vui lòng tải file:
Choose Files DataTroNew.csv
• DataTroNew.csv(text/csv) - 4992 bytes, last modified: 6/9/2025 - 100% done
Saving DataTroNew.csv to DataTroNew.csv
```

Hình 3.10. Mã nguồn tải dữ liệu training.

- Đoạn mã này yêu cầu người dùng tải file từ máy tính của họ lên môi trường Colab. File được tải lên sẽ nằm trong biến uploaded.

Bước 3: Tạo hàm load_and_clean_data(file_path)

```python
def load_and_clean_data(file_path):
    """Tải và làm sạch dữ liệu từ file CSV."""
    df = pd.read_csv(file_path)
    # Chuyển Quyet dinh thue thành Yes/No
    # Kiểm tra cột 'Quyet dinh thue' có tồn tại trước khi mapping
    if 'Quyet dinh thue' in df.columns:
        df['Quyet dinh thue'] = df['Quyet dinh thue'].map({'co': 'Yes', 'khong': 'No'})
    else:
        print("Cảnh báo: Cột 'Quyet dinh thue' không tồn tại trong DataFrame.")
    # Loại bỏ giá trị thiếu (nếu có)
    df = df.dropna()
    return df
# Tải dữ liệu
# Đảm bảo 'DataTroNew.csv' là tên file bạn đã tải lên
df = load_and_clean_data('DataTroNew.csv')
print("Dữ liệu đã tải:")
print(df.head())
```

```
Dữ liệu đã tải:
            Phong Chung chu Dieu hoa Nong lanh            Quat Tu lanh  \
0        khep kin        co       co         co        quat tran      co
1        khep kin        co    khong         co        quat tran   khong
2  khong khep kin        co       co     khong         quat cay   khong
3        khep kin     khong    khong         co  quat treo tuong   khong
4  khong khep kin     khong       co     khong            khong   khong

     Bep Tu quan ao      Giuong Gac xep May giat Khoang cach(km) An ninh  \
0     co      tu go  giuong doi      co       co              ~1     tot
1  khong      tu go  giuong don      co    khong              ~1      on
2  khong     tu nhua  giuong don   khong    khong            ~0.5      on
3  khong     tu nhua  giuong don   khong    khong            ~1.5     tot
4  khong      tu vai  giuong don   khong    khong            ~0.5      on

  Dien tich Tien dien(nghin/kwh) Tien nuoc/m3 Tien rac/thang Gia phong  \
0      20m2                   3k          20k            20k      3tr8
1      15m2                   4k          20k            20k      2tr5
2       8m2                   3k          15k            15k       1tr
3      10m2                 3.5k          12k            20k       1tr
4       7m2                 2.5k          15k            15k      700k

  Quyet dinh thue
0             Yes
1             Yes
2              No
3              No
4             Yes
```

.Hình 3.11. Mã nguồn hàm load_and_clean_data(file_path)

- Hàm này nhận đường dẫn file CSV làm đầu vào.
- df = pd.read_csv(file_path): Đọc dữ liệu từ file CSV vào một DataFrame của pandas.
- if 'Quyet dinh thue' in df.columns:: Kiểm tra xem cột 'Quyet dinh thue' có tồn tại trong DataFrame hay không.

- df['Quyet dinh thue'] = df['Quyet dinh thue'].map({'co': 'Yes', 'khong': 'No'}): Nếu cột tồn tại, nó sẽ ánh xạ các giá trị 'co' thành 'Yes' và 'khong' thành 'No' trong cột 'Quyet dinh thue'.
- df = df.dropna(): Loại bỏ các hàng chứa bất kỳ giá trị thiếu nào.
- return df: Trả về DataFrame đã được tải và làm sạch.
- Đoạn mã sau đó gọi hàm này với tên file là 'DataTroNew.csv' và in ra vài dòng đầu của DataFrame đã tải.

Bước 4: Tại Hàm find_entropy(df, target_attribute).

```
def find_entropy(df, target_attribute):
    """Tính entropy của tập dữ liệu dựa trên thuộc tính mục tiêu."""
    target_values = df[target_attribute].unique()
    entropy = 0
    for value in target_values:
        prob = len(df[df[target_attribute] == value]) / len(df)
        if prob > 0:
            entropy -= prob * np.log2(prob)
    return entropy
```

.Hình 3.12. Mã nguồn hàm find_entropy(df, target_attribute)

- Tính entropy của thuộc tính mục tiêu trong DataFrame.
- Entropy là một thước đo mức độ "không thuần nhất" của dữ liệu. Entropy bằng 0 khi tất cả các mẫu có cùng một nhãn.
- Hàm lặp qua các giá trị duy nhất của thuộc tính mục tiêu, tính xác suất của từng giá trị và sử dụng công thức entropy: entropy = - Σ (p_i * log2(p_i)).

Bước 5: Tạo Hàm find_entropy_attribute(df, attribute, target_attribute)

```
def find_entropy_attribute(df, attribute, target_attribute):
    """Tính entropy trung bình của một thuộc tính."""
    attribute_values = df[attribute].unique()
    entropy = 0
    for value in attribute_values:
        subset = df[df[attribute] == value]
        prob = len(subset) / len(df)
        entropy += prob * find_entropy(subset, target_attribute)
    return entropy
```

.Hình 3.12. Mã nguồn hàmfind_entropy_attribute(df, attribute, target_attribute)

- Tính entropy trung bình có trọng số của một thuộc tính nhất định.
- Điều này đo lường entropy của thuộc tính mục tiêu sau khi chia dữ liệu dựa trên các giá trị của thuộc tính đầu vào.

- Hàm lặp qua các giá trị duy nhất của thuộc tính, tạo tập con dữ liệu cho mỗi giá trị, tính entropy của tập con đó và nhân với trọng số (tỷ lệ của tập con so với toàn bộ dữ liệu). Sau đó, tổng hợp các giá trị này.

Bước 6: Tạo hàm find_information_gain(df, attribute, target_attribute).

```
def find_information_gain(df, attribute, target_attribute):
    """Tính độ lợi thông tin của một thuộc tính."""
    total_entropy = find_entropy(df, target_attribute)
    attribute_entropy = find_entropy_attribute(df, attribute, target_attribute)
    return total_entropy - attribute_entropy
```

.Hình 3.13. Mã nguồn find_information_gain(df, attribute, target_attribute).

- Tính độ lợi thông tin (information gain) của một thuộc tính.
- Information gain là sự giảm entropy đạt được bằng cách chia dữ liệu dựa trên một thuộc tính. Nó được tính bằng công thức: Information Gain = Entropy(Target) - Weighted Average Entropy(Attribute).
- Thuộc tính có information gain cao nhất là thuộc tính tốt nhất để phân chia dữ liệu tại mỗi nút của cây quyết định.

Bước 7: Tạo hàm id3(df, features, target_attribute, parent_node_class=None)

```
def id3(df, features, target_attribute, parent_node_class=None):
    """Xây dựng cây quyết định ID3."""
    # Nếu tất cả mẫu có cùng nhãn, trả về nhãn đó
    if len(df[target_attribute].unique()) == 1:
        return df[target_attribute].iloc[0]

    # Nếu không còn thuộc tính, trả về nhãn phổ biến nhất
    if len(features) == 0:
        return df[target_attribute].mode()[0] if not df[target_attribute].mode().empty else parent_node_class

    # Tìm thuộc tính có information gain cao nhất
    gains = {attr: find_information_gain(df, attr, target_attribute) for attr in features}
    best_attribute = max(gains, key=gains.get)

    # Tạo node gốc với thuộc tính tốt nhất
    tree = {best_attribute: {}}
    remaining_features = [f for f in features if f != best_attribute]

    # Phân nhánh cho từng giá trị của thuộc tính
    for value in df[best_attribute].unique():
        subset = df[df[best_attribute] == value]
        if subset.empty:
            tree[best_attribute][value] = df[target_attribute].mode()[0] if not df[target_attribute].mode().empty else parent_node_class
        else:
            tree[best_attribute][value] = id3(subset, remaining_features, target_attribute, df[target_attribute].mode()[0])

    return tree
```

Hình 3.14. Mã nguồn id3(df, features, target_attribute, parent_node_class=None)

- Đây là hàm chính xây dựng cây quyết định sử dụng thuật toán ID3 một cách đệ quy.

Điều kiện dừng:

- Nếu tất cả các mẫu trong tập con hiện tại có cùng nhãn mục tiêu, trả về nhãn đó.

- Nếu không còn thuộc tính nào để chia, trả về nhãn phổ biến nhất trong tập con (hoặc nhãn của nút cha nếu tập con rỗng).
- Tìm thuộc tính tốt nhất: Tính information gain cho tất cả các thuộc tính còn lại và chọn thuộc tính có information gain cao nhất.

Tạo nút: Tạo một từ điển (hoặc cấu trúc tương tự) để biểu diễn nút cây, với thuộc tính tốt nhất là khóa.

Phân nhánh: Với mỗi giá trị duy nhất của thuộc tính tốt nhất, tạo một tập con dữ liệu tương ứng và gọi đệ quy hàm id3 trên tập con đó với các thuộc tính còn lại. Kết quả của mỗi lời gọi đệ quy sẽ là nhánh con của nút hiện tại.

Trả về cây: Hàm trả về cấu trúc cây quyết định đã được xây dựng.

Bước 8: Tạo hàm predict(tree, instance).

```
def predict(tree, instance):
    """Dự đoán nhãn dựa trên cây quyết định và một mẫu dữ liệu."""
    if not isinstance(tree, dict):
        return tree
    attribute = list(tree.keys())[0]
    value = instance.get(attribute)
    # Xử lý trường hợp giá trị thuộc tính không có trong cây
    if value not in tree[attribute]:
        print(f"Cảnh báo: Giá trị '{value}' cho thuộc tính '{attribute}' không có trong cây quyết định.")
        return None
    return predict(tree[attribute][value], instance)
```

Hình 3.15. Mã nguồn hàm predict(tree, instance)

- Hàm này nhận cây quyết định đã được xây dựng và một mẫu dữ liệu (một từ điển chứa các giá trị thuộc tính) làm đầu vào.
- Nó đi xuống cây quyết định, bắt đầu từ nút gốc, dựa trên các giá trị thuộc tính của mẫu dữ liệu.
- Nếu gặp một nút lá (không phải là từ điển), nó trả về nhãn dự đoán.
- Nếu gặp một giá trị thuộc tính trong mẫu dữ liệu không có trong cây, nó in cảnh báo và trả về None.

Bước 9: Tạo hàm get_valid_values(df, column).

```python
def get_valid_values(df, column):
    """Trả về danh sách giá trị hợp lệ và giá trị phổ biến nhất."""
    # Đảm bảo cột tồn tại trước khi truy cập
    if column not in df.columns:
        print(f"Cảnh báo: Cột '{column}' không tồn tại trong DataFrame.")
        return [], None # Trả về danh sách rỗng và giá trị mặc định là None

    values = df[column].dropna().unique().tolist()
    # Xử lý trường hợp không có giá trị nào sau khi dropna
    if not values:
        return [], None

    most_common = df[column].mode()[0] if not df[column].mode().empty else values[0]
    return values, most_common
```

Hình 3.16. Mã nguồn hàm get_valid_values(df, column)

- Hàm này nhận DataFrame và tên cột làm đầu vào.
- Nó trả về một danh sách các giá trị duy nhất (không bao gồm giá trị thiếu) trong cột đó.
- Nó cũng trả về giá trị phổ biến nhất trong cột (được sử dụng như giá trị mặc định trong giao diện tương tác).
- Bổ sung kiểm tra xem cột có tồn tại trong DataFrame hay không để tránh lỗi.

Bước 10: Tạo hàm interactive_predict(tree, df, features)

```python
def interactive_predict(tree, df, features):
    """Dự đoán quyết định thuê trọ với giao diện thân thiện."""
    print("=" * 50)
    print("HỆ THỐNG DỰ ĐOÁN QUYẾT ĐỊNH THUÊ TRỌ")
    print("=" * 50)
    instance = {}
    for feature in features:
        valid_values, default_value = get_valid_values(df, feature)
        if not valid_values and default_value is None:
            print(f"Bỏ qua thuộc tính '{feature}' vì không tìm thấy dữ liệu hợp lệ.")
            continue
        while True:
            user_input = input(f"Nhập số (1-{len(valid_values)}) hoặc nhấn Enter: ").strip()
            try:
                choice = int(user_input)
                if 1 <= choice <= len(valid_values):
                    instance[feature] = valid_values[choice - 1]
                    break
                else:
                    print(f"Vui lòng nhập số từ 1 đến {len(valid_values)}!")
            except ValueError:
                print("Vui lòng nhập số hợp lệ hoặc nhấn Enter!")
        print(f"Đã chọn: {instance[feature]}")
    print("\n" + "=" * 50)
    print("THÔNG TIN ĐÃ NHẬP:")
    for feature, value in instance.items():
        print(f"  {feature}: {value}")
    prediction = predict(decision_tree, instance) # Sử dụng biến decision_tree đã được tạo
    if prediction is None:
        print("\nKẾT QUẢ DỰ ĐOÁN:")
        print("Không thể dự đoán: Giá trị thuộc tính không hợp lệ hoặc không có trong cây quyết định.")
    else:
        result = 'Có' if prediction == 'Yes' else 'Không'
        print("\nKẾT QUẢ DỰ ĐOÁN:")
        print(f"Quyết định thuê trọ: {result}")

    print("=" * 50)
    return prediction
if 'Quyet dinh thue' in df.columns:
    features = [col for col in df.columns if col != 'Quyet dinh thue']
    target_attribute = 'Quyet dinh thue'
    if not df.empty:
        decision_tree = id3(df, features, target_attribute)
        print(decision_tree)
        interactive_predict(decision_tree, df, features)
    else:
        print("Không có dữ liệu sau khi làm sạch. Không thể xây dựng cây quyết định.")
else:
    print("Không tìm thấy cột 'Quyet dinh thue' trong dữ liệu. Không thể xây dựng cây quyết định.")
```

Hình 3.17. Mã nguồn hàm interactive_predict(tree, df, features)

- Hàm này tạo ra một giao diện dòng lệnh thân thiện để người dùng nhập thông tin về một mẫu dữ liệu mới để dự đoán.
- Nó lặp qua danh sách các thuộc tính (features) được sử dụng để xây dựng cây.
- Đối với mỗi thuộc tính, nó hiển thị các giá trị hợp lệ lấy từ DataFrame gốc và yêu cầu người dùng nhập số tương ứng hoặc nhấn Enter (để chọn giá trị mặc định).

Nó xây dựng một từ điển instance chứa các giá trị thuộc tính do người dùng nhập.

- Sau khi thu thập đủ thông tin, nó gọi hàm predict để dự đoán nhãn dựa trên instance và cây decision_tree.
- In ra thông tin đã nhập và kết quả dự đoán một cách rõ ràng.

Bước 11: Chạy chương trình và tìm ra điều kiện của bài toán.

- Kiểm tra xem cột 'Quyet dinh thue' có tồn tại trong DataFrame sau khi tải và làm sạch hay không.
- Nếu có, tạo danh sách features (tất cả các cột trừ 'Quyet dinh thue') và gán target_attribute là 'Quyet dinh thue'.
- Kiểm tra xem DataFrame có dữ liệu sau khi làm sạch hay không.
- Nếu có dữ liệu, gọi hàm id3 để xây dựng cây quyết định và lưu vào biến decision_tree.
- Sau đó, gọi hàm interactive_predict để bắt đầu quá trình dự đoán tương tác cho người dùng.
- Nếu cột mục tiêu không tồn tại hoặc không có dữ liệu sau khi làm sạch, in ra thông báo tương ứng.

Tóm lại, đoạn mã này thực hiện các bước sau: tải và làm sạch dữ liệu, tính toán entropy và information gain để tìm ra các thuộc tính tốt nhất, xây dựng cây quyết định ID3 một cách đệ quy, và cung cấp một giao diện tương tác để người dùng nhập dữ liệu và nhận dự đoán từ cây đã xây dựng.

3.3.3. Nhận xét chương trình

Ưu điểm:

- **Cấu trúc rõ ràng:** Mã được chia thành các hàm riêng biệt cho từng chức năng (tải dữ liệu, tính entropy, xây dựng cây, dự đoán, v.v.), giúp dễ đọc, hiểu và bảo trì.
- **Triển khai thuật toán ID3 cơ bản:** Mã tuân thủ các bước cơ bản của thuật toán ID3, bao gồm tính entropy, information gain, và xây dựng cây đệ quy.
- **Xử lý dữ liệu thiếu (đơn giản):** Hàm load_and_clean_data bao gồm bước dropna(), đây là một cách đơn giản để xử lý dữ liệu thiếu (loại bỏ các hàng chứa giá trị thiếu).
- **Giao diện tương tác:** Hàm interactive_predict cung cấp một cách thân thiện để người dùng nhập dữ liệu cho dự đoán, hiển thị các giá trị hợp lệ cho mỗi thuộc tính.
- **Kiểm tra cột mục tiêu:** Mã kiểm tra sự tồn tại của cột 'Quyet dinh thue' trước khi tiến hành, tránh lỗi nếu file dữ liệu không có cột này.
- **Xử lý trường hợp không có dữ liệu:** Mã kiểm tra xem DataFrame có rỗng sau khi làm sạch hay không trước khi xây dựng cây.

Nhược điểm và các điểm cần cải thiện:

- **Xử lý dữ liệu thiếu (còn hạn chế):** Việc chỉ đơn giản loại bỏ các hàng chứa giá trị thiếu (dropna()) có thể dẫn đến mất mát đáng kể dữ liệu, đặc biệt nếu dữ liệu thiếu nhiều. Các phương pháp xử lý dữ liệu thiếu khác như điền giá trị (mean, median, mode) hoặc sử dụng các kỹ thuật tiên tiến hơn có thể hiệu quả hơn.
- **Không xử lý dữ liệu liên tục:** Thuật toán ID3 gốc được thiết kế cho các thuộc tính phân loại. Mã hiện tại không có cơ chế xử lý các thuộc tính có giá trị liên tục (số thực). Đối với dữ liệu liên tục, cần phải thực hiện quá trình rời rạc hóa (discretization) trước khi áp dụng ID3.
- **Khả năng quá khớp (Overfitting):** Cây quyết định ID3 gốc có xu hướng quá khớp với dữ liệu huấn luyện, đặc biệt là khi cây rất sâu. Không có cơ chế cắt tỉa cây (pruning) được triển khai để giảm thiểu quá khớp.
- **Xử lý giá trị thuộc tính không thấy trong dự đoán:** Hàm predict chỉ in cảnh báo và trả về None nếu một giá trị thuộc tính trong mẫu dự đoán không tồn tại trong cây. Trong thực tế, có thể cần một chiến lược xử lý khác, ví dụ như đi theo nhánh phổ biến nhất hoặc sử dụng giá trị phổ biến nhất của thuộc tính đó từ dữ liệu huấn luyện.
- **Giao diện tương tác đơn giản:** Giao diện interactive_predict chỉ dựa trên nhập số thứ tự. Đối với các thuộc tính có nhiều giá trị, việc này có thể không tiện lợi lắm. Một giao diện đồ họa hoặc web sẽ thân thiện hơn.
- **Không có đánh giá mô hình:** Mã không bao gồm bất kỳ phương pháp đánh giá hiệu suất của cây quyết định (ví dụ: độ chính xác, ma trận nhầm lẫn) trên một tập dữ liệu kiểm tra riêng biệt. Điều này rất quan trọng để hiểu mô hình hoạt động tốt như thế nào trên dữ liệu mới.
- **Thiếu xử lý lỗi chi tiết:** Mặc dù có một số kiểm tra cơ bản (như tồn tại cột), mã có thể được cải thiện với việc xử lý lỗi chi tiết hơn, ví dụ như khi đọc file CSV gặp vấn đề.
- **Cảnh báo:** UserWarning: BoundsError: Make sure bounds is not empty: Lỗi này có thể xảy ra trong hàm df[attribute].mode()[0]. Nó xuất hiện khi không có giá trị phổ biến nhất (ví dụ: tất cả các giá trị đều xuất hiện chỉ một lần hoặc không có dữ liệu). Việc kiểm tra df[attribute].mode().empty và cung cấp giá trị mặc định (ví dụ: giá trị đầu tiên trong danh sách các giá trị hợp lệ) đã được thêm vào trong phiên bản mã bạn cung cấp, giúp khắc phục vấn đề này.
- **Tên biến và hàm:** Tên biến và hàm sử dụng tiếng Việt khá tốt và dễ hiểu trong ngữ cảnh của bài toán.

Kết luận:

Mã nguồn là một điểm khởi đầu tốt để hiểu và triển khai thuật toán ID3. Tuy nhiên, để sử dụng trong các ứng dụng thực tế, cần cải thiện đáng kể các khía cạnh như xử lý dữ liệu liên tục, chống quá khớp, xử lý dữ liệu thiếu hiệu quả hơn, và bổ sung các bước đánh giá mô hình. Giao diện tương tác cũng có thể được nâng cấp để thân thiện hơn.

KẾT LUẬN

Kết quả đạt được:

- Xây dựng được hệ thống ra quyết định thuê trọ ở mức cơ bản
- Hiểu sâu hơn về thuật toán, ứng dụng của cây quyết định và kiến thức cơ bản về một số thuật toán học máy khác.
- Có kiến thức, đánh giá chân thực hơn về trí tuệ nhân tạo trong cuộc sống
- Làm quen ngôn ngữ lập trình Python, Google Colab
- Áp dụng được kiến thức đã học vào bài toán thực tiễn.
- Các hoạt động tổ chức, xây dựng, lên kế hoạch làm việc nhóm.

Hạn chế:

- Thuật toán áp dụng có độ chính xác chưa được cao
- Hệ thống xây dựng cần hoàn thiện hơn
- Một số công việc còn chưa hoàn thành đúng tiến độ

TÀI LIỆU THAM KHẢO

[1] Nguyễn Phương Nga, Trần Hùng Cường : Giáo trình Trí Tuệ Nhân Tạo, Nhà Xuất Bản Thống Kê, 2021.

[2] Artificial Intelligence with Python, Packt, 2017.