TRƯỜNG ĐẠI HỌC DUY TÂN

KHOA SAU ĐẠI HỌC

*Tiểu luận môn*

KHAI PHÁ DỮ LIỆU

ỨNG DỤNG MÔ HÌNH CÂY QUYẾT ĐỊNH ĐỂ

HỖ TRỢ ĐẶT HÀNG CHO SIÊU THỊ ĐIỆN THOẠI

Giảng viên : TS Nông Thị Hoa

Học viên : Phạm Minh Tuấn

Lớp : K22MCS (Khoa học máy tính)

Đà nẵng, 01/2021

MỤC LỤC

[DANH MỤC HÌNH VẼ 1](#_Toc62980233)

[LỜI MỞ ĐẦU 2](#_Toc62980234)

[Chương 1 3](#_Toc62980235)

[ĐẶT VẤN ĐỀ 3](#_Toc62980236)

[1. Bài toán đặt hàng 3](#_Toc62980237)

[1.1.1 Giới thiệu bài toán đặt hàng trong kinh doanh 3](#_Toc62980238)

[1.1.2 Quy trình đặt hàng trong kinh doanh 3](#_Toc62980239)

[2. Giới thiệu cây quyết định và thuật toán 4](#_Toc62980240)

[1.2.1 Mô hình cây quyết định 4](#_Toc62980241)

[1.2.2 Các loại cây quyết định 5](#_Toc62980242)

[1.2.3 Xây dựng cây quyết định 5](#_Toc62980243)

[1.2.4 Tiêu chí chọn thuộc tính phân lớp 6](#_Toc62980244)

[1.2.5 Thuật toán ID3 7](#_Toc62980245)

[3. Ví dụ về thuật toán ID3 8](#_Toc62980246)

[Chương 2 18](#_Toc62980247)

[MÔ HÌNH CÂY QUYẾT ĐỊNH VÀ CÔNG CỤ HỖ TRỢ WEKA 18](#_Toc62980248)

[2.1 Tập dữ liệu phân lớp 18](#_Toc62980249)

[2.2 Thao tác dữ liệu trên Weka 19](#_Toc62980250)

[2.2.1 Phân lớp bằng thuật toán Random Forest 20](#_Toc62980251)

[2.2.2 Phân lớp bằng thuật toán J48 21](#_Toc62980252)

[2.2.3 Đánh giá kết quả 22](#_Toc62980253)

[2.2.2 Vẽ cây và rút ra tập luật từ thuật toán J48 22](#_Toc62980254)

[Chương 3 26](#_Toc62980255)

[KẾT LUẬN 26](#_Toc62980256)

[TÀI LIỆU THAM KHẢO 27](#_Toc62980257)

# DANH MỤC HÌNH VẼ

|  |  |  |
| --- | --- | --- |
| **Ký hiệu** | **Nội dung** | **Trang** |
| Hình 1.1 | Cấu trúc cây quyết định | 4 |
| Hình 1.2 | Đồ thị của hàm entropy với n = 2 | 6 |
| Hình 1.3 | Cây quyết định vẽ bằng thuật toán ID3 | 17 |
| Hình 3.1 | Hình ảnh dữ liệu mẫu | 18 |
| Hình 3.2 | Thông tin dữ liệu mẫu trên Weka | 19 |
| Hình 3.3 | Thông tin các thuộc tính của dữ liệu mẫu | 20 |
| Hình 3.4 | Kết quả thuật toán Random Forest | 21 |
| Hình 3.6 | Kết quả thuật toán J48 | 22 |
| Hình 3.7 | Cây quyết định từ vẽ bằng thuật toán J48 | 23 |

# LỜI MỞ ĐẦU

1. **Lý do chọn đề tài**

Đối với hệ thống siêu thị điện thoại thì số lượng hàng hóa kinh doanh rất nhiều. Cùng với đó, các lần giao dịch mua bán luôn diễn ra liên tục. Nếu như trước đây khi chưa có hệ thống quản lý hoặc có nhưng chưa được can thiệp sâu vào khía cạnh hỗ trợ và tự động hóa thì việc lên kế hoạch kinh doanh cũng như đặt hàng chỉ dựa vào trực giác và kinh nghiệm. Việc đó vô tình tạo nên cho người kinh doanh áp lực về ghi nhớ số liệu và đưa ra những quyết định thiên về cảm tính vì không dựa vào phân tích và báo cáo dữ liệu mua bán. Vì vậy, việc đưa ra một giải pháp nhằm hỗ trợ cho việc phân tích số liệu và đặt hàng là vô cùng cần thiết để doanh nghiệp có thể giảm thiểu rủi ro và đưa ra những quyết định đúng đắn trong kinh doanh từ đó giúp doanh nghiệp phát triển ổn định.

1. **Mục tiêu, phương pháp nghiên cứu**

Mục tiêu của tiểu luận là đưa ra giải pháp hỗ trợ bộ phận kinh doanh của siêu thị điện thoại để đưa ra các quyết định đặt hàng đúng đắn dựa vào việc phân tích dữ liệu thực tế tại doanh nghiệp. Các giao dịch mua bán được sử dụng trong tiểu luận được kết xuất trong 6 tháng cuối năm 2020. Dữ liệu sẽ được làm sạch và tiền xử lý sau đó sử dụng công cụ Weka 3.9.5 để phân tích, đánh giá kết quả để đưa ra tập luật sử dụng cho giải pháp đặt hàng.

1. **Bố cục tiểu luận**

Nội dung của bài tiểu luận được trình bày với bố cục gồm 03 chương như sau:

Chương 1: Giới thiệu bài toán, nội dung mô hình cây quyết định và thuật toán ID3

Chương 2: Thu thập và xử lý dữ liệu thô, sử dụng công cụ Weka để phân tích và thực nghiệm dữ liệu sau đó đánh giá kết quả đạt được.

Chương 3: Kết luận: Trình bày kết quả đạt được của tiểu luận và tính thực tế cũng như định hướng phát triển trong tương lai.

# Chương 1

# ĐẶT VẤN ĐỀ

**1. Bài toán đặt hàng**

**1.1.1 Giới thiệu bài toán đặt hàng trong kinh doanh**

Trong các siêu thị điện thoại thì số lượng mặt hàng kinh doanh rất lớn và liên tục được sàng lọc. Chưa dừng lại ở đó, các giao dịch mua bán sẽ diễn ra liên tục trong ngày nên việc ghi nhớ dữ liệu mua bán cũng như đánh giá loại hàng nào kinh doanh hiệu quả rất là khó khăn nếu chỉ dựa vào trí nhớ của con người. Bên cạnh đó, nếu mô hình là một chuỗi siêu thị thì hầu như việc ghi nhớ là không thể. Vì vậy, yêu cầu đặt ra là làm thế nào để lưu lại và số hóa tất cả thông tin giao dịch mua bán để thay thế cách ghi nhớ bằng giấy thủ công. Trên cơ sở đó, khi đã có dữ liệu thì việc tìm ra giải pháp để hỗ trợ việc kinh doanh mà không cần dựa vào trí nhớ và trực quan của con người là hoàn toàn khả thi. Bài tiểu luận sẽ đề cập đến việc trích lọc dữ liệu và mô hình hóa dựa trên thuật toán của cây quyết định để tìm ra tập luật sau đó người dùng có thể xây dựng công cụ hoặc dựa vào các cách suy diễn đó để đặt hàng một cách hiệu quả.

**1.1.2 Quy trình đặt hàng trong kinh doanh**

Với một hệ thống lớn, mỗi nhân viên kinh doanh sẽ chịu trách nhiệm một nhóm mặt hàng cụ thể. Để đặt hàng thì bộ phận kinh doanh sẽ dùng phần mềm hỗ trợ để trích lọc các dữ liệu liên quan đến các loại hàng cần đặt sau đó phân tích và lên danh sách để tiến hành đặt hàng.

***Bước 1:*** Trích lọc dữ liệu theo loại hàng cần đặt hàng, bước này sẽ tùy theo thời điểm trong năm sẽ chọn khoảng thời gian phù hợp. Ví dụ mùa tết sẽ nhập nhiều nên ít quan tâm đến thống kê dữ liệu bán ra.

***Bước 2:*** Thống kê các loại hàng cần mua, rà soát số liệu bán ra theo loại hàng, màu sắc, đặc điểm, tính năng, sau đó lên danh sách các loại hàng cần mua. Bước này phải thống kê chi tiết vì ví dụ như màu sắc đẹp sẽ bán được nhiều hơn.

***Bước 3:*** Phân tích, so sánh số liệu với các nhãn hàng cùng phân khúc để quyết định số lượng nên đặt mua. Trong bước này, còn phải cân đối tài chính, các loại hàng đang hot hoặc mối quan hệ làm ăn…

***Bước 4:*** Rà soát lại lần cuối và tiến hành gửi danh sách đến nhà phân phối để tiến hành đặt hàng.

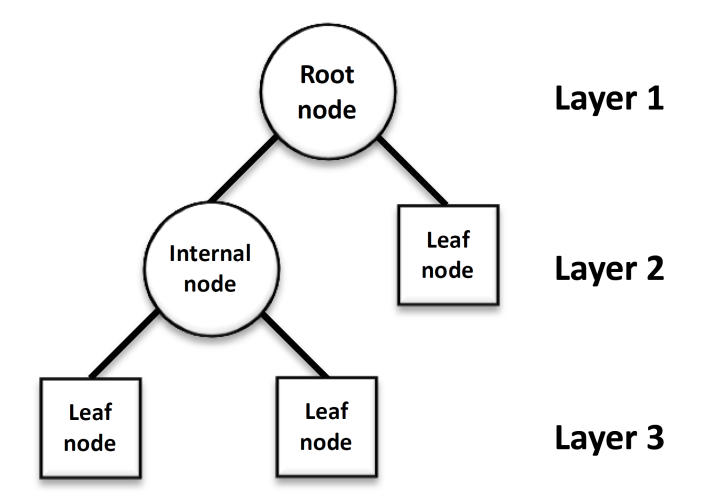
**2. Giới thiệu cây quyết định và thuật toán**

**1.2.1 Mô hình cây quyết định**

Mô hình cây quyết định là công cụ được dùng để phân lớp dữ liệu, mỗi cây quyết định tượng trưng cho một quyết định của lớp các dữ kiện nào đó. Mỗi nút trong cây là tên của một lớp hay một phép thử thuộc tính cụ thể nào đó, phép thử này phân chia không gian trạng thái các dữ kiện tại nút đó thành các kết quả có thể đạt được của phép thử. Mỗi tập con được phân chia của phép thử là không gian con của các sự kiện, nó tương ứng với một vấn đề con của sự phân lớp.

Một cách tổng thể, cây quyết định có cấu trúc và các tính chất sau:

* Mỗi nút trong (internal node) biểu diễn một thuộc tính cần kiểm tra giá trị (an attribute to be tested) đối với các các tập thuộc tính.
* Nút lá (leaf node) hay còn gọi là nút trả lời biểu thị cho một lớp các trường hợp mà nhãn của nó là tên của lớp, nó biểu diễn một lớp (a classififcation)
* Nút nhánh (branch) từ một nút sẽ tương ứng với một giá trị có thể của thuộc tính gắn với nút đó.
* Nhãn (lable) của nút này là tên của thuộc tính và có một nhánh nối nút này đến các cây con ứng với mỗi kết quả có thể có phép thử. Nhãn của nhánh này là các giá trị của thuộc tính đó. Nút trên cùng gọi là nút gốc.



**Hình 1.1: Cấu trúc cây quyết định**

**1.2.2 Các loại cây quyết định**

Cây quyết định có 2 loại cơ bản sau đây:

***Cây hồi quy (Regression tree):*** ước lượng các hàm giá có giá trị là số thực thay vì được sử dụng cho các nhiệm vụ phân loại. (ví dụ: ước tính giá một ngôi nhà hoặc khoảng thời gian một bệnh nhân nằm viện).

***Cây phân loại (Classification tree):*** nếu y là một biến phân loại như: giới tính (nam hay nữ), kết quả của một trận đấu (thắng hay thua).

**1.2.3 Xây dựng cây quyết định**

Có nhiều thuật toán khác nhau để xây dựng cây quyết định như: CLS, CART, ID3, C4.5, SLIQ, SPRINT, EC4.5, C5.0… Nhưng nói chung quá trình xây dựng cây quyết định đều được chia ra làm 3 giai đoạn cơ bản:

***Tạo cây:*** Cây quyết định được tạo thành bằng cách lần lượt chia (theo phương pháp đệ quy) một tập dữ liệu thành các tập dữ liệu con, mỗi tập con được tạo thành từ các phần tử của cùng một lớp. Các nút (không phải là nút lá) là các điểm phân nhánh của cây. Việc phân nhánh tại các nút có thể dựa trên việc kiểm tra một hay nhiều thuộc tính để xác định việc phân chia dữ liệu.

Quá trình xây dựng một cây quyết định cụ thể bắt đầu bằng một nút rỗng bao gồm toàn bộ các đối tượng huấn luyện và làm như sau:

* + - 1. Nếu tại nút hiện thời, tất cả các đối tượng huấn luyện đều thuộc vào một lớp nào đó thì nút này chính là nút lá có tên là nhãn lớp chung của các đối tượng.
      2. Trường hợp ngược lại, sử dụng một độ đo, chọn thuộc tính điều kiện phân chia tốt nhất tập mẫu huấn luyện có tại nút.
      3. Tạo một lượng nút con của nút hiện thời bằng số các giá trị khác nhau của thuộc tính được chọn. Gán cho mỗi nhánh từ nút cha đến nút con một giá trị của thuộc tính rồi phân chia các các đối tượng huấn luyện vào các nút con tương ứng.
      4. Nút con K được gọi là thuần nhất, trở thành lá, nếu tất cả các đối tượng mẫu tại đó đều thuộc vào cùng một lớp.
      5. Lặp lại các bước 1 - 3 đối với mỗi nút chưa thuần nhất.

***Tỉa cây:*** Sau giai đoạn tạo cây chúng ta có thể dùng phương pháp “Độ dài mô tả ngắn nhất” (Minimum Description Length) hay giá trị tối thiểu của IG để tỉa cây (chúng ta có thể chọn giá trị tối thiểu của IG trong giai đoạn tạo cây đủ nhỏ để cho cây phát triển tương đối sâu, sau đó lại nâng giá trị này lên để tỉa cây).

***Đánh giá cây:*** Dùng để đánh giá độ chính xác của cây kết quả. Tiêu chí đánh giá là tổng số mẫu được phân lớp chính xác trên tổng số mẫu đưa vào.

**1.2.4 Tiêu chí chọn thuộc tính phân lớp**

***Đại lượng Entropy***

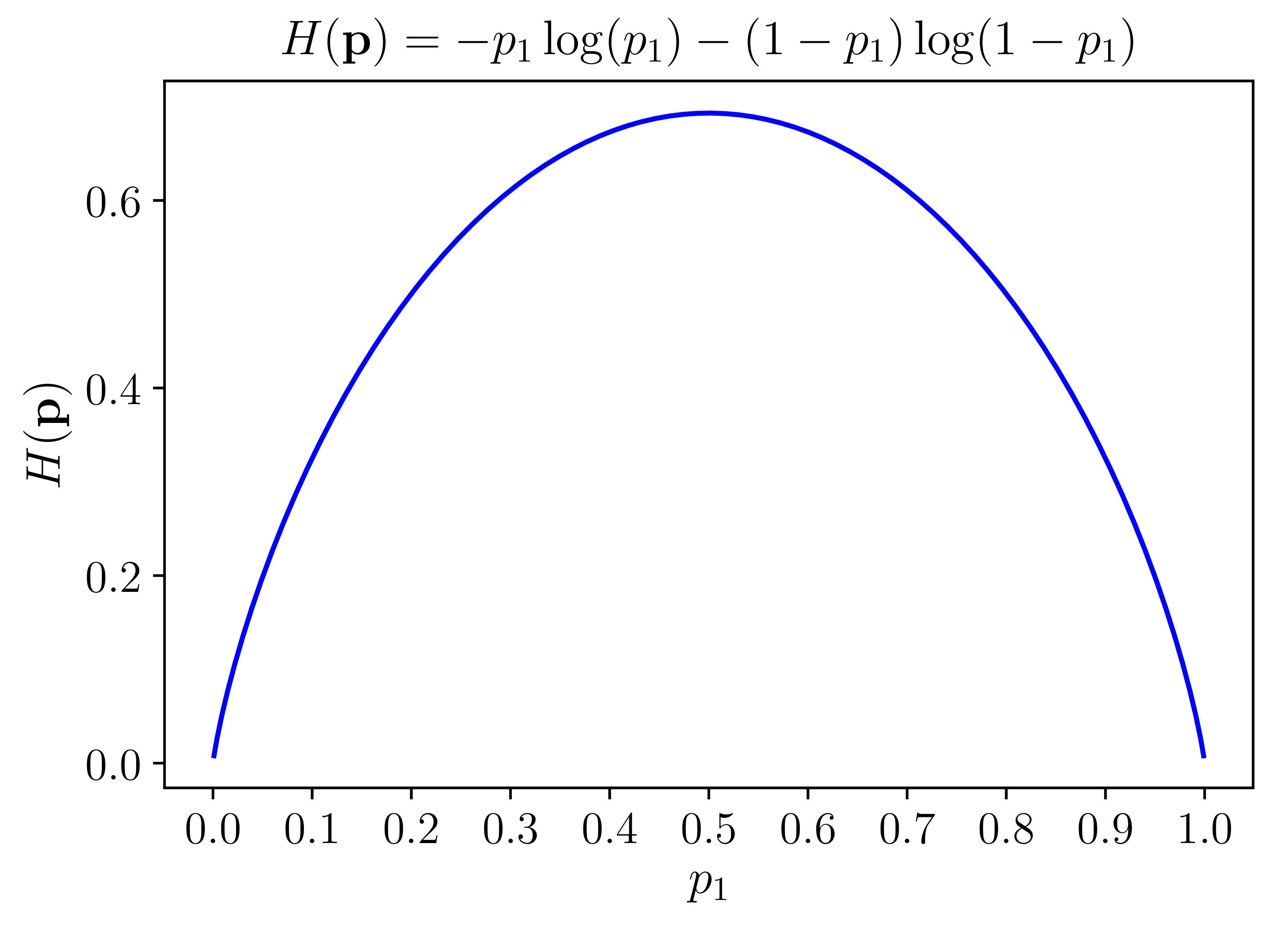
Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

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

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

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

Ví dụ như trường hợp tung một đồng xu, Entropy sẽ được tính như sau:



***Hình 2: Đồ thị của hàm entropy với n = 2***

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

P tinh khiết: pi = 0 hoặc pi = 1

P vẩn đục: pi = 0.5, khi đó hàm Entropy đạt đỉnh cao nhất

***Độ lợi thông tin Information Gain***

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

Để tính Infomation Gain tại mỗi nút ta thực hiện theo trình tự sau:

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

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

Bước 3: Chỉ số Gain Information được tính bằng:

**1.2.5 Thuật toán ID3**

Thuật toán ID3 được phát biểu bởi Quinlan (trường đại học Syney, Australia) và được công bố vào cuối thập niên 70 của thế kỷ 20. ID3 được xem như là một cải tiến của thuật toán CLS với khả năng lựa chọn thuộc tính tốt nhất để tiếp tục triển khai cây tại mỗi bước. ID3 xây dựng cây quyết định từ trên xuống (top -down).

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

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

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

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

Giải thuật ID3 xây dựng cây quyết định được trình bày như sau:

Lặp:

* + - 1. Chọn A <= thuộc tính quyết định “tốt nhất” cho nút kế tiếp
      2. Gán A là thuộc tính quyết định cho nút
      3. Với mỗi giá trị của A, tạo nhánh con mới của nút
      4. Phân loại các mẫu huấn luyện cho các nút lá
      5. Nếu các mẫu huấn luyện được phân loại hoàn toàn thì dừng, ngược lại thì lặp với các nút lá mới.

Thuộc tính tốt nhất ở đây là thuộc tính có Entropy trung bình thấp nhất theo thuộc tính kết quả với Entropy được tính như sau:

Gọi S là tập các mẫu huấn luyện, gọi p là tỷ lệ các mẫu dương trong S

Ta có H ≡ – p.log2p – (1 – p).log2(1 – p)

Entropy trung bình của một thuộc tính bằng trung bình theo tỉ lệ của Entropy các nhánh

**3. Ví dụ về thuật toán ID3**

Ta có tập dữ liệu như bảng bên dưới:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 2 | VSmart | 3000mAh | 16GB | 2GB | Yes |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 4 | VSmart | 4000mAh | 32GB | 4GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |
| 7 | VSmart | 3000mAh | 32GB | 4GB | Yes |
| 8 | VSmart | 4000mAh | 16GB | 4GB | Yes |

***1.3.1.1 Xét thuộc tính BRAND***

Thuộc tính này có thể nhận 1 trong 2 giá trị SamSung, VSmart, tương ứng với 2 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2​.

Sắp xếp lại theo thuộc tính Brand ta có 2 bảng nhỏ.

Brand VSmart (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 2 | VSmart | 3000mAh | 16GB | 2GB | Yes |
| 4 | VSmart | 4000mAh | 32GB | 4GB | Yes |
| 7 | VSmart | 3000mAh | 32GB | 4GB | Yes |
| 8 | VSmart | 4000mAh | 16GB | 4GB | Yes |

Brand SamSung (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

Child node ứng với Brand VSmart sẽ có Entropy = 0 do tất cả các giá trị trường ORDER đều là Yes. Vì vậy ta chỉ việc tính Entropy của Brand SamSung, sau đó tính Entropy trung bình.

Cụ thể như sau:

***1.3.1.2 Xét thuộc tính PIN***

Thuộc tính này có thể nhận 1 trong 3 giá trị 3000mAh, 4000mAh, 3500mAh tương ứng với 3 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2, S3​.

Sắp xếp lại theo thuộc tính PIN ta có 3 bảng nhỏ.

PIN 4000mAh (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 4 | VSmart | 4000mAh | 32GB | 4GB | Yes |
| 8 | VSmart | 4000mAh | 16GB | 4GB | Yes |

PIN 3000mAh (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 2 | VSmart | 3000mAh | 16GB | 2GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 7 | VSmart | 3000mAh | 32GB | 4GB | Yes |

PIN 3500mAh (S3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

Áp dụng công thức ta tính Entropy cho 3 loại PIN như sau:

***1.3.1.3 Xét thuộc tính ROM***

Thuộc tính này có thể nhận 1 trong 2 giá trị là 16GB, 32GB tương ứng với 2 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2​.

Sắp xếp lại theo thuộc tính ROM ta có 2 bảng nhỏ.

ROM 16GB (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 2 | VSmart | 3000mAh | 16GB | 2GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 8 | VSmart | 4000mAh | 16GB | 4GB | Yes |

ROM 32GB (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 4 | VSmart | 4000mAh | 32GB | 4GB | Yes |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |
| 7 | VSmart | 3000mAh | 32GB | 4GB | Yes |

Áp dụng công thức ta tính Entropy cho 2 loại ROM như sau:

***1.3.1.4 Xét thuộc tính RAM***

Thuộc tính này có thể nhận 1 trong 2 giá trị là 2GB, 4GB tương ứng với 2 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2​.

Sắp xếp lại theo thuộc tính RAM ta có 2 bảng nhỏ.

RAM 2GB (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 2 | VSmart | 3000mAh | 16GB | 2GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

RAM 4GB (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 4 | VSmart | 4000mAh | 32GB | 4GB | Yes |
| 7 | VSmart | 3000mAh | 32GB | 4GB | Yes |
| 8 | VSmart | 4000mAh | 16GB | 4GB | Yes |

Áp dụng công thức ta tính Entropy cho 2 loại RAM như sau:

***1.3.2 Chọn thuộc tính có giá trị Entropy nhỏ nhất***

* H(Brand,S) = 0.5
* H(Pin,S) ≈ 0.594
* H(Rom,S) ≈ 0.811
* H(Ram,S) ≈ 0.811

Thuộc tính Brand có giá trị Entropy nhỏ nhất nên ta chọn là node đánh giá đầu tiên.

Với Brand có giá trị là VSmart, tất cả các data đều có giá trị Yes, vì vậy ta thu được node là Yes ở nhánh VSmart.

Vì vậy, ta chỉ còn danh sách Brand còn lại là SamSung với tập data được đã được thu gọn như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

***1.3.3 Lặp lại tính và so sánh Entopy***

Lặp lại bước ***1.3.1 Tính Entropy của các thuộc tính*** và ***1.3.2 Chọn thuộc tính có giá trị Entropy nhỏ nhất*** cho bảng dữ liệu đã được thu gọn với các thuộc tính PIN, ROM, RAM.

***1.3.3.1 Xét thuộc tính PIN***

Thuộc tính này có thể nhận 1 trong 3 giá trị 4000mAh, 3500mAh, 3000mAh tương ứng với 3 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2, S3​.

Sắp xếp lại theo thuộc tính PIN ta có 3 bảng nhỏ.

PIN 4000mAh (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |

PIN 3500mAh (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

PIN 3000mAh (S3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |

Áp dụng công thức ta tính Entropy cho 3 loại PIN như sau:

***1.3.3.2 Xét thuộc tính ROM***

Thuộc tính này có thể nhận 1 trong 2 giá trị là 16GB, 32GB tương ứng với 2 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2​.

Sắp xếp lại theo thuộc tính ROM ta có 2 bảng nhỏ.

ROM 16GB (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |

ROM 32GB (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

Áp dụng công thức ta tính Entropy cho 2 loại ROM như sau:

***1.3.3.3 Xét thuộc tính RAM***

Thuộc tính này có thể nhận 1 trong 2 giá trị là 2GB, 4GB tương ứng với 2 child node.

Gọi tập hợp các điểm trong mỗi child node này lần lượt là S1, S2​.

Sắp xếp lại theo thuộc tính RAM ta có 2 bảng nhỏ.

RAM 2GB (S1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 1 | SamSung | 4000mAh | 16GB | 2GB | Yes |
| 5 | SamSung | 3000mAh | 16GB | 2GB | No |
| 6 | SamSung | 3500mAh | 32GB | 2GB | Yes |

RAM 4GB (S2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | BRAND | PIN | ROM | RAM | ORDER? |
| 3 | SamSung | 3500mAh | 32GB | 4GB | No |

Áp dụng công thức ta tính Entropy cho 2 loại RAM như sau:

***1.3.4 Chọn thuộc tính có giá trị Entropy nhỏ nhất***

* H(Pin,S) ≈ 0.5
* H(Rom,S) ≈ 1
* H(Ram,S) ≈ 0.688

Thuộc tính Pin có giá trị Entropy nhỏ nhất nên ta chọn là node đánh giá tiếp theo.

Với trường hợp Pin là 4000mAh hoặc 3000mAh, ta có ngay node lá vì chỉ có một kết quả.

Với trường hợp Pin là 3500mAh, do thuộc tính Rom là giống nhau với tất cả data, ta chọn node đánh giá tiếp theo là Ram.

Thuộc tính Ram có 2 giá trị là 2GB và 4GB và đều có 1 giá trị là Yes và No nên ta có ngay 2 nút lá của thuộc tính Ram. Thuật toán đến đây dừng lại

Brand?

VSmart

SamSung

4000mAh

3000mAh

Pin?

Ram?

3500mAh

2GB

4GB

***Hình 1.3: Cây quyết định bằng thuật toán ID3***

# Chương 2

# MÔ HÌNH CÂY QUYẾT ĐỊNH VÀ CÔNG CỤ HỖ TRỢ WEKA

**2.1 Tập dữ liệu phân lớp**

Dữ liệu phục vụ cho việc phân lớp được lấy từ giao dịch mua bán 6 tháng cuối năm 2020 của Công ty TNHH Quốc Hùng có địa chỉ tại 200 Nguyễn Văn Linh, Quận Thanh Khê, TP. Đà Nẵng.

Tập tin dữ liệu có tổng cộng 300 dòng và 6 trường: BRAND, PIN, RAM, ROM, COLOR, ORDER trong đó trường ORDER là trường quyết định (nút lá) của cây quyết định.



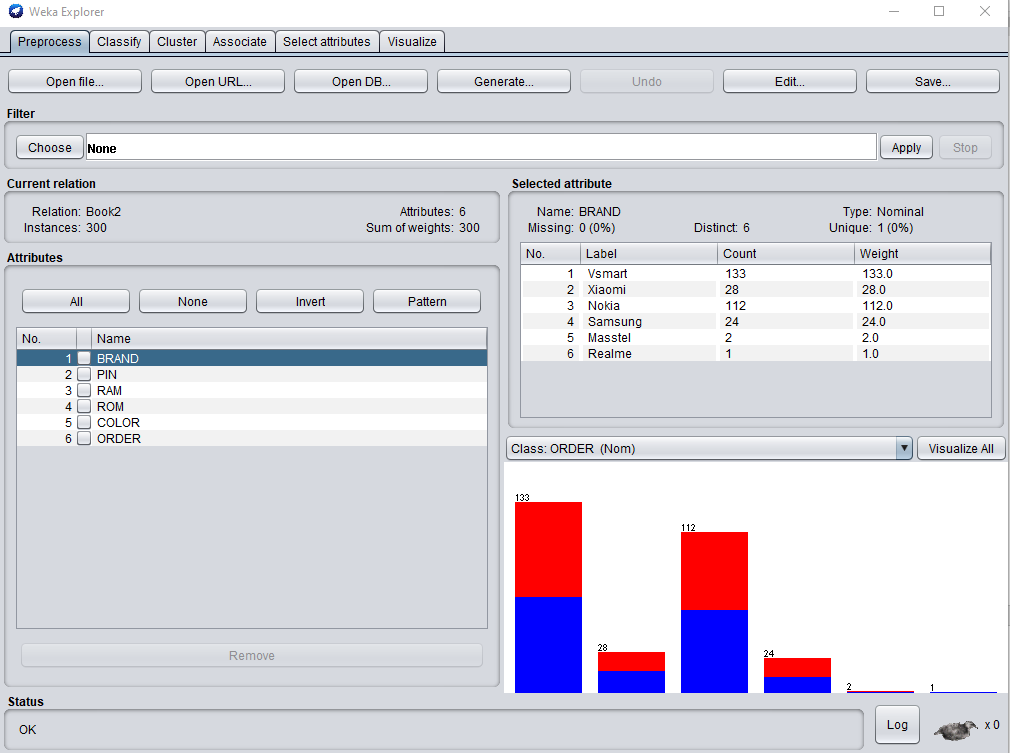
***Hình 3.1: Hình ảnh dữ liệu mẫu***

* BRAND: Nhãn hàng có gồm có 6 giá trị: Nokia, Xiaomi, Masstel, Realme, Vsmart, Samsung.
* PIN: Phân khúc Pin của điện thoại gồm 6 giá trị: 0000->1000mAh, 4000->5000mAh, 1000->2000mAh, 3000->4000mAh, 2000->3000mAh, 5000->9000mAh
* RAM: Bộ nhớ tạm của điện thoại gồm 5 giá trị: 1GB, 2GB, 4GB, 6GB, 8GB.
* ROM: Bộ nhớ trong của điện thoại gồm 5 giá trị: 16GB, 32GB, 64GB, 128GB, 512GB.
* COLOR: Màu sắc của điện thoại gồm 4 giá trị: Red, Gold, Black, Blue.
* ORDER: Quyết định đặt hàng hay không, YES (đặt hàng), NO(không đặt hàng).

**2.2 Thao tác dữ liệu trên Weka**

Dữ liệu sau khi truy vấn được lưu vào file DuLieu.csv với cấu trúc và số dòng tương ứng hình ảnh dữ liệu mẫu.

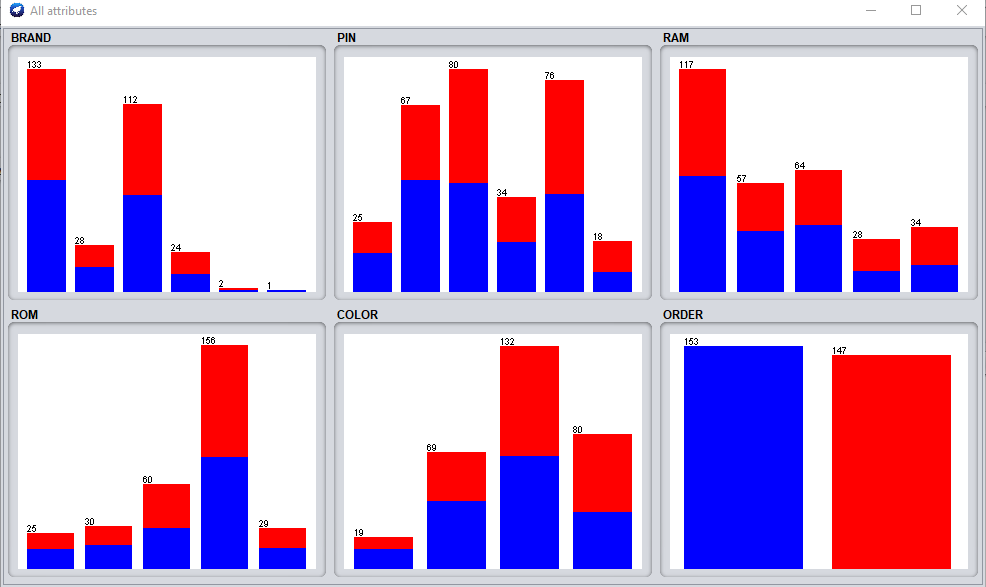
Sau khi mở tập tin DuLieu.csv bằng Weka thì sẽ hiển thị hình ảnh như bên dưới:



***Hình 3.2: Thông tin dữ liệu mẫu trên Weka***

Trong hình vẽ trên, ta thấy được tổng số Attributes (Số cột) là 6, Instances (Số dòng) là 300, kiểu dữ liệu là Nominal.

Sau khi kích vào Visualize All ta được hình ảnh hiển thị độ phân lớp dữ liệu tương ứng với từng trường trong tập tin dữ liệu như bên dưới:



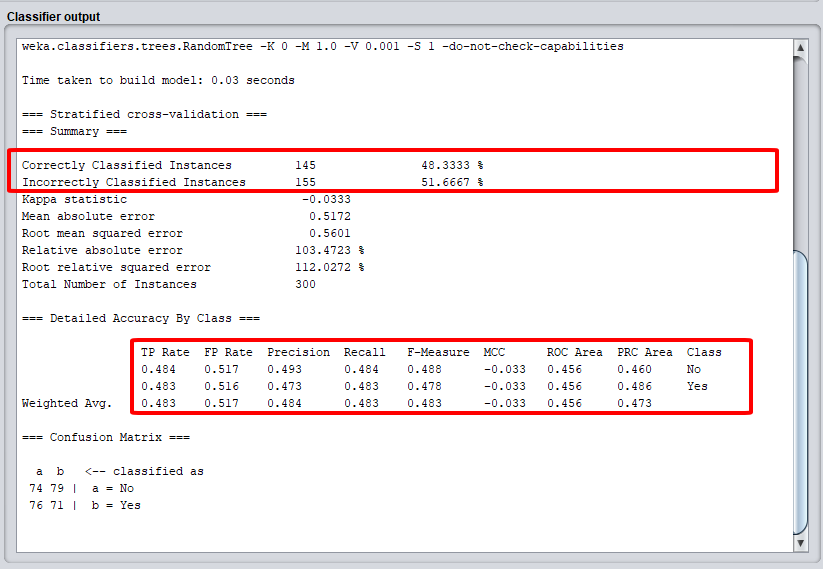
***Hình 3.3: Thông tin các thuộc tính của dữ liệu mẫu***

**2.2.1 Phân lớp bằng thuật toán Random Forest**

Mở tab Classify trong Weka, ở mục Classifier ta chọn thuật toán Random Forest.

Trong mục Test option, ta dùng Cross-validation Folds 10. Trong đó dữ liệu gốc sẽ được chia thành n phần bằng nhau (n-fold), và quá trình Train/Test Model thực hiện lặp lại n lần. Tại mỗi lần Train/Test Model, 1 phần dữ liệu dùng để Test và (n-1) phần còn lại dùng để Train.

Chọn thuộc tính phân lớp là (ORDER) type



***Hình 3.4: Kết quả thuật toán Random Forest***

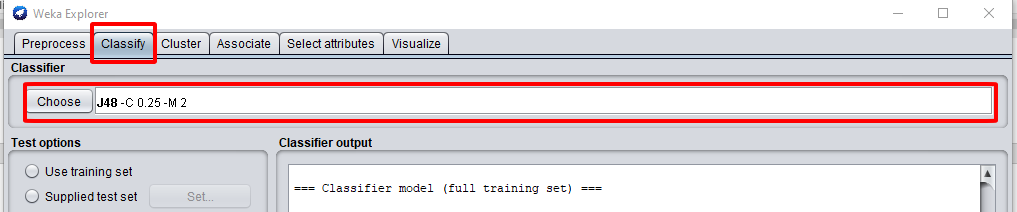
Phân ra 2 lớp của thuộc tính Order: No (không đặt hàng), Yes (đặt hàng)

Phân loại chính xác: 48% trường hợp

Phân loại không chính xác: 51 % trường hợp

**2.2.2 Phân lớp bằng thuật toán J48**

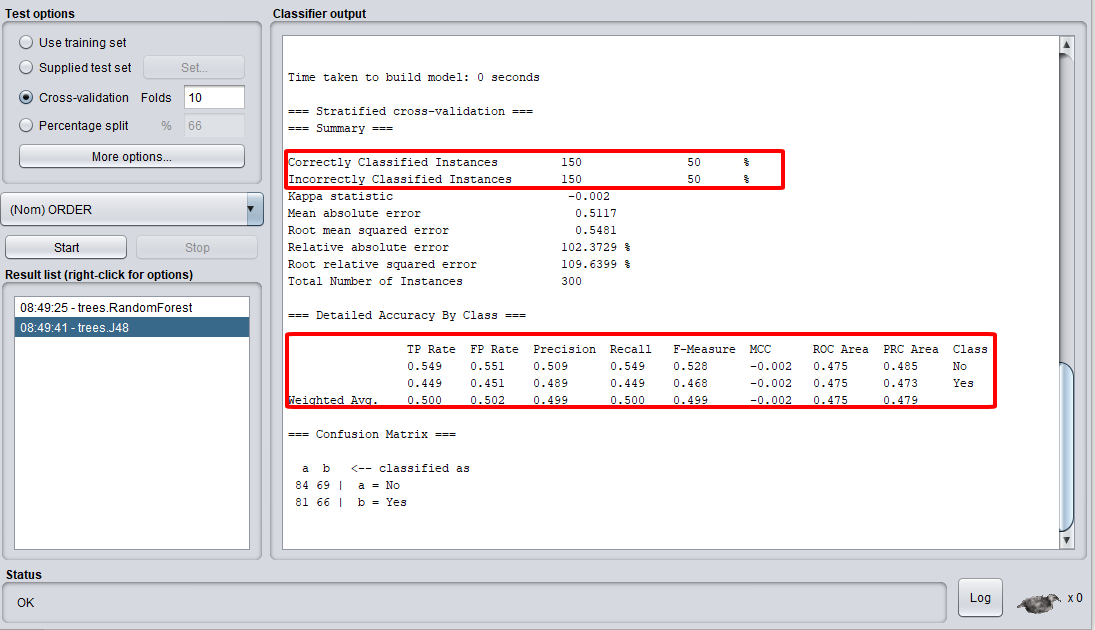
Mở tab Classify trong Weka, ở mục Classifier ta chọn thuật toán J48.



Trong mục Test option, ta dùng Cross-validation Folds 10. Đây là kỹ thuật chủ yếu được sử dụng trong xây dựng predictive Model. Trong đó dữ liệu gốc sẽ được chia thành n phần bằng nhau (n-fold), và quá trình Train/Test Model thực hiện lặp lại n lần. Tại mỗi lần Train/Test Model, 1 phần dữ liệu dùng để Test và (n-1) phần còn lại dùng để Train.

Chọn thuộc tính phân lớp là (ORDER) type

Sau khi bấm Start ta sẽ có được kết quả như hình vẽ dưới:



***Hình 3.5: Kết quả thuật toán J48***

Phân ra 2 lớp của thuộc tính Order: No (không đặt hàng), Yes (đặt hàng)

Phân loại chính xác: 50% trường hợp

Phân loại không chính xác: 50 % trường hợp

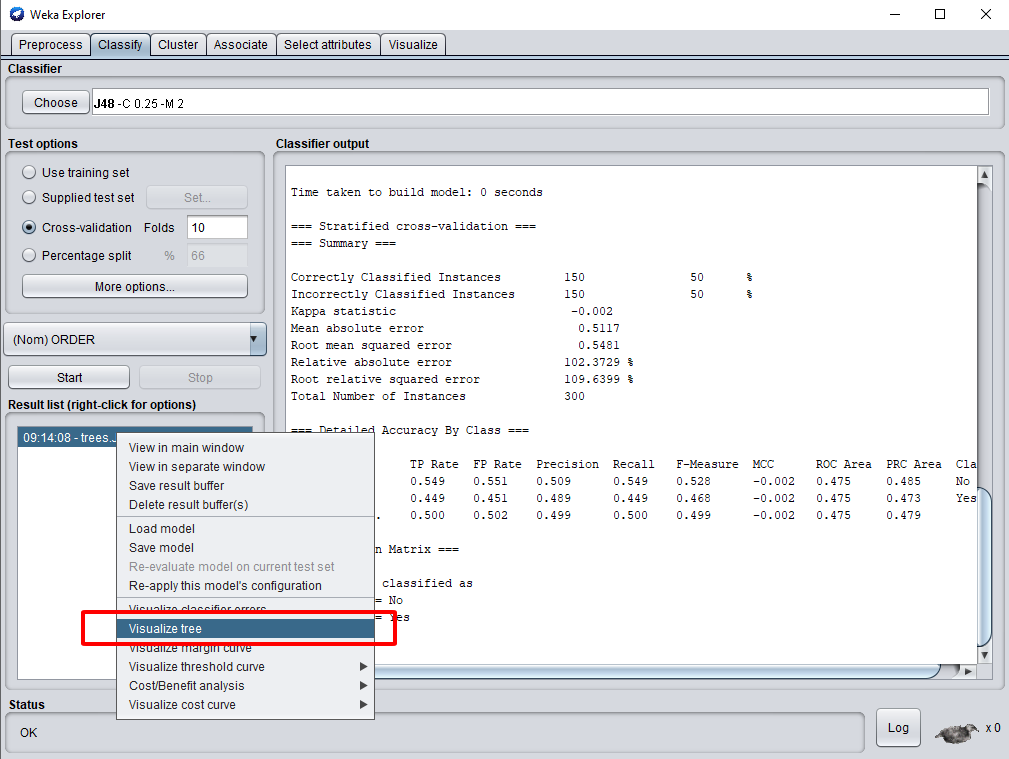
**2.2.3 Đánh giá kết quả**

Sau khi phân loại bằng 2 thuật toán là Random Forest và J48 ta thấy kết quả phân loại của thuật toán Random Forest có kết quả phân loại tốt hơn.

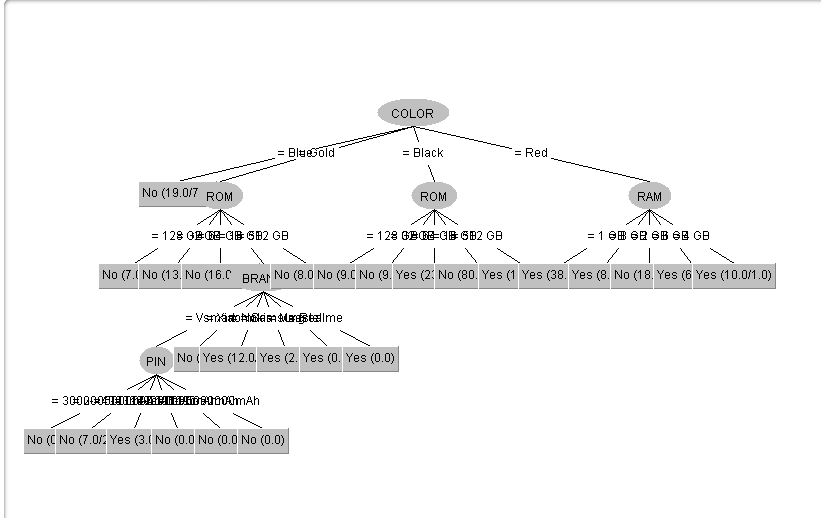
|  |  |
| --- | --- |
| Thuật toán | Confusion Matrix |
| Random Forest |  |
| J48 |  |

**2.2.2 Vẽ cây và rút ra tập luật từ thuật toán J48**

Sau khi chạy thuật toán J48, ở mục Result list bên trái chọn Visualize tree như hình vẽ dưới:



Sau khi Visualize tree ta có hình vẽ cây như bên dưới:



***Hình 3.6: Cây quyết định từ vẽ bằng thuật toán J48***

Dựa vào kết quả thuật toán J48 và hình ảnh tree ta đưa ra tập luật như sau:

* IF (COLOR = Blue) THEN No
* IF (COLOR = Gold AND ROM = 32 GB) THEN No
* IF (COLOR = Gold AND ROM = 64 GB) THEN No
* IF (COLOR = Gold AND ROM = 128 GB) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 3000->4000mAh) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 2000->3000mAh) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 5000->9000mAh) THEN Yes
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 0000->1000mAh) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 4000->5000mAh) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Vsmart AND PIN = 1000->2000mAh) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Xiaomi) THEN No
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Nokia) THEN Yes
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Samsung) THEN Yes
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Masstel) THEN Yes
* IF (COLOR = Gold AND ROM = 16 GB AND BRAND = Realme) THEN Yes
* IF (COLOR = Gold AND ROM = 512 GB) THEN No
* IF (COLOR = Black AND ROM = 128 GB) THEN No
* IF (COLOR = Black AND ROM = 32 GB) THEN No
* IF (COLOR = Black AND ROM = 64 GB) THEN Yes
* IF (COLOR = Black AND ROM = 16 GB) THEN No
* IF (COLOR = Black AND ROM = 512 GB) THEN Yes
* IF (COLOR = Red AND RAM = 1 GB) THEN Yes
* IF (COLOR = Red AND RAM = 8 GB) THEN Yes
* IF (COLOR = Red AND RAM = 6 GB) THEN Yes
* IF (COLOR = Red AND RAM = 4 GB) THEN Yes
* IF (COLOR = Red AND RAM = 2 GB) THEN No

# Chương 3

# KẾT LUẬN

Việc ứng dụng phân tích dữ liệu mua bán để hỗ trợ kinh doanh là bài toán thực tế và rất cần thiết cho những doanh nghiệp chưa có hệ thống quản lý hoặc hệ thống chưa hỗ trợ tính năng đặt hàng. Với những doanh nghiệp có nhiều mặt hàng và nhiều chi nhánh thì đó là khoảng đầu tư phát triển hợp lý. Bên cạnh đó, với sự phát triển không ngừng của Ngành công nghệ thông tin, các công cụ hỗ trợ phân tích dữ liệu ngày càng phong phú và hỗ trợ đắc lực con người trong công tác dự báo.

Thông qua quá trình nghiên cứu về mô hình cây quyết định và công cụ hỗ trợ WEKA, tiểu luận đã tiến hành giải quyết bài toán thực tế về công tác hỗ trợ ra quyết định. Cụ thể, tiểu luận đã đi sâu nghiên cứu và làm rõ những nội dung sau:

- Đưa ra cơ sở lý thuyết về mô hình cây quyết đinh ứng dụng trong việc phân tích dữ liệu để đặt hàng trong kinh doanh.

- Sử dụng công cụ hỗ trợ WEKA để giải quyết bài toán thực tế về phân tích dữ liệu bán hàng của Công ty THHH Quốc Hùng.

Tiểu luận đã cho thấy sự hữu ích của việc phân tích dữ liệu để áp dụng, giải quyết các bài toán thực tế. Tuy nhiên, do một số nguyên nhân khách quan và chủ quan, luận văn vẫn còn tồn tại một số hạn chế sau:

- Dữ liệu thu thập của duy nhất một đơn vị dẫn đến công tác dự bảo mới chỉ dừng lại ở phạm vi cục bộ.

- Chưa tìm hiểu hết tất cả các tính năng của công cụ hỗ trợ WEKA để giải quyết các bài toàn thực tế.

Để khắc phục những hạn chế nêu trên, trong thời gian tới, hướng nghiên cứu sẽ tiếp tục mở rộng phạm vi thu thập dữ liệu, tìm hiểu rõ công cụ hỗ trợ WEKA và các công cụ hỗ trợ khác để tiến hành dự báo có tính khái quát và chính xác hơn.

Tiểu luận chỉ ở mức tìm hiểu, nội dung phân tích chưa thật sự sâu sắc nên nếu có điều kiện về thời gian và nguồn tài liệu phong phú hơn thì đề tài sẽ được mở rộng và hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

1. Hiền, Đ. T. T., & Đức, N. H. (2019). Data mining: Bài giảng môn học Khai phá dữ liệu.
2. Nguyễn Thị Hạnh, T.S Hồ Cẩm Hà, “Khai phá dữ liệu bằng cây quyết định”, 2008 Nguyễn, T. T. M. (2010).
3. Nguyễn Thị Thùy Linh, “Nghiên cứu các thuật toán phân lớp dữ liệu dựa trên cây.