BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGUYỄN VŨ QUỲNH TRÂM

TRẦN THÀNH TRUNG

CÂY QUYẾT ĐỊNH CHO BÀI TOÁN PHÂN LỚP

(DECISION TREE FOR CLASSIFICATION)

ĐỒ ÁN MÔN HỌC

NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGUYỄN VŨ QUỲNH TRÂM

TRẦN THÀNH TRUNG

CÂY QUYẾT ĐỊNH CHO BÀI TOÁN PHÂN LỚP

(DECISION TREE FOR CLASSIFICATION)

Mã số sinh viên 1: 1751010166

Mã số sinh viên 2: 1751010172

ĐỒ ÁN MÔN HỌC

NGÀNH KHOA HỌC MÁY TÍNH

Giảng viên hướng dẫn: VÕ THỊ HỒNG TUYẾT

TP. HỒ CHÍ MINH, 2020

TRƯỜNG ĐẠI HỌC MỞ **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

THÀNH PHỐ HỒ CHÍ MINH **Độc lập – Tự do – Hạnh phúc**

KHOA CÔNG NGHỆ THÔNG TIN

———

GIẤY XÁC NHẬN

Tôi tên là: Nguyễn Vũ Quỳnh Trâm

Ngày sinh: 5/12/1999 Nơi sinh: Thành phố HCM

Chuyên ngành: Khoa học máy tính Mã sinh viên: 1751010166

Tôi đồng ý cung cấp toàn văn thông tin đồ án ngành hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án ngành vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên

(Ghi rõ họ và tên)

……

TRƯỜNG ĐẠI HỌC MỞ **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

THÀNH PHỐ HỒ CHÍ MINH **Độc lập – Tự do – Hạnh phúc**

KHOA CÔNG NGHỆ THÔNG TIN

———

GIẤY XÁC NHẬN

Tôi tên là: Trần Thành Trung

Ngày sinh: 19/05/1999 Nơi sinh: Thành phố Hồ Chí Minh

Chuyên ngành: Khoa học máy tính Mã sinh viên: 1751010172

Tôi đồng ý cung cấp toàn văn thông tin đồ án ngành hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án ngành vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên

(Ghi rõ họ và tên)

……

Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN NGÀNH

CỦA GIẢNG VIÊN HƯỚNG DẪN

Giảng viên hướng dẫn: Võ Thị Hồng Tuyết

Sinh viên thực hiện: Nguyễn Vũ Quỳnh Trâm

Lớp: DH17TH01

Ngày sinh: 5/12/1999 Nơi sinh: Thành phố Hồ Chí Minh

Tên đề tài:

CÂY QUYẾT ĐỊNH CHO BÀI TOÁN PHÂN LỚP

(DECISION TREE FOR CLASSIFICATION)

Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án trước Hội đồng:

.....

.....

.....

.....

.....

.....

.....

.....

Thành phố Hồ Chí Minh, ngày ... tháng ... năm ……

Người nhận xét

Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN NGÀNH

CỦA GIẢNG VIÊN HƯỚNG DẪN

Giảng viên hướng dẫn: Võ Thị Hồng Tuyết

Sinh viên thực hiện: Trần Thành Trung

Lớp: DH17TH01

Ngày sinh: 19/5/1999 **Nơi sinh: Thành phố Hồ Chí Minh**

Tên đề tài:

CÂY QUYẾT ĐỊNH CHO BÀI TOÁN PHÂN LỚP

(DECISION TREE FOR CLASSIFICATION)

Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án trước Hội đồng:

.....

.....

.....

.....

.....

.....

.....

.....

Thành phố Hồ Chí Minh, ngày ... tháng ... năm ……

Người nhận xét

……

LỜI CẢM ƠN

Trong suốt thời gian thực hiện đồ án, nhóm chúng em xin chân thành gửi lời cảm ơn tới Ths. Võ Thị Hồng Tuyết vì đã tận tâm hướng dẫn cũng như ủng hộ nhóm chúng em hoàn thành dự án.

Với trình độ kiến thức còn hạn hẹp và chưa có nhiều kinh nghiệm thì dự án của chúng em cũng không thể tránh khỏi những hạn chế và sai sót, qua đó chúng em cũng mong nhận được những nhận xét và góp ý để có thể tiếp tục xây dựng dự án một cách hoàn thiện nhất.

Chúng em cũng xin chân thành cảm ơn tới các giảng viên trường Đại học Mở thành phố Hồ Chí Minh vì đã truyền tải các kiến thức để chúng em có thể hoàn thành đồ án này.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

ML	Machine Learning
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
CART	Classification And Regression Tree
GPA	Grade Point Average
LMS	Learning Management System
MDD	Major depressive disorder
ODA	Official Development Assistanc
WDBC	Wisconsin Diagnosis Breast Cancer
ID3	Iterative Dichotomiser 3

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

Trong thời đại hiện nay, kể cả kinh doanh, sản xuất hay các ngành nghề khác đều cần tới dữ liệu nên song hành với nó là khối dữ liệu ngày càng khổng lồ, nhưng với việc công nghệ ngày càng hiện đại khiến cho việc quản lý dữ liệu trở nên dễ dàng hơn.

Với khối dữ liệu khổng lồ họ đã nghiên cứu ra nhiều thuật toán phân tích dữ liệu từ đó tạo ra nhiều tính năng tiên tiến để cung cấp cho các lĩnh vực khác. Và đó là lý do mà việc khai phá dữ liệu trở thành một lĩnh vực nghiên cứu giúp giải quyết các vấn đề liên quan tới dữ liệu, thu thập được những thông tin cần thiết từ việc phân tích khối dữ liệu khổng lồ.

Từ khi lĩnh vực này xuất hiện, phân tích dữ liệu trở nên phổ biến trong giới khoa học máy tính. Nó mang lại hiệu quả trong vấn đề giải quyết dữ liệu của nhiều lĩnh vực khác nhau.

Để khai phá dữ liệu có rất nhiều phương pháp như: Phân loại (Classification), Hồi quy (Regression), Phân cụm (Clustering), Tổng hợp (Summarization), Mô hình ràng buộc (Dependency Modeling) và Dò tìm biến đổi và độ lệch (Change and Deviation Detection). Và trong các phương pháp đó phương pháp được sử dụng nhiều nhất trong những công nghệ hiện đại ngày nay là phương pháp phân loại và trong phương pháp này thuật toán có thể nói là nổi tiếng nhất thường thấy trong lĩnh vực khoa học dữ liệu, phân tích dữ liệu đó là thuật toán cây quyết định (Decision Tree).

Và đó cũng sẽ là thuật toán mà chúng em sẽ dựa vào đó để phân tích kho dữ liệu động vật và phân loại chúng thành những loài khác nhau. Từ đó vẽ ra cây quyết định để dự đoán loài động vật nào dựa vào các đặc trưng của nó.

Chương 1. GIỚI THIỆU TỔNG QUAN

1.1. Giới thiệu đề tài

Với công nghệ ngày càng phát triển, thì số lượng dữ liệu trong ngành công nghiệp này đang ngày càng tăng lên. Vì vậy, việc phân tích khối dữ liệu lớn này để trở thành những thông tin hữu ích đang thực sự cần thiết. Bài toán phân lớp sử dụng cây quyết định là một trong những thuật toán phổ biến và đủ mạnh mẽ để có thể xử lý được khối thông tin này. Đề tài này tập trung nghiên cứu về một số thuật toán về Cây quyết định như ID3, C4.5, CART, … và kĩ thuật phân lớp và dự đoán dựa trên bộ dữ liệu.

1.2. Mục tiêu nghiên cứu

Mục tiêu nghiên cứu là tìm ra thuật toán tối ưu khi xây dựng cây quyết định để áp dụng vào bài toán phân lớp động vật. Từ đó dự đoán được tập dữ liệu loài động vật dủng để kiểm tra bằng mô hình cây quyết định.

1.3. Giới hạn/đối tượng nghiên cứu

Dataset chúng em sử dụng là bảng dữ liệu các loài động vật với 15 đặc trưng ở dạng có/không (1/0): Cho sữa (milk), có lông vũ (feathers), có lông (hair), sống trên không (airborne), sống dưới nước (aquatic), tập tính săn mồi (predator), cỏ răng (toothed), có xương sống (backbone), breathes, có nọc độc (venomous), có vây (fins), đẻ trứng (eggs), có đuôi (tail), vật nuôi (domestic), dấu chân (catsize) và một đặc trưng ở dạng số lượng là số chân (legs) với giá trị là 0, 2, 4, 5, 6, 8.

Các loài động vật này được chia làm 7 loài khác nhau (class type):

1. Động vật có vú (Mammal)
2. Chim (Bird)
3. Bò sát (Reptile)
4. Cá (Fish)
5. Lưỡng cư (Amphibian)
6. Côn trùng, bọ (Bug)
7. Động vật không xương sống (Invertebrate)

1.4. Phương pháp nghiên cứu

Tìm hiểu các thuật toán tạo ra nút đầu tiên trong cây quyết định bằng các phương pháp thống kê, phân tích và tổng hợp từ những thuật toán khác nhau từ đó chọn ra phương pháp đưa ra được kết quả có độ chính xác cao.

Sau khi có được phương pháp cần sử dụng tiếp tục phân tích các nút còn lại của cây bằng phương pháp phân tích, thống kê dữ liệu đầu vào, và bước cuối cùng khi xây dựng cây quyết định là áp dụng phương pháp so sánh để đối chiếu với cây quyết định tương tự từ thư viện có sẵn trong python.

1.5. Bố cục

Bài báo cáo gồm 4 chương:

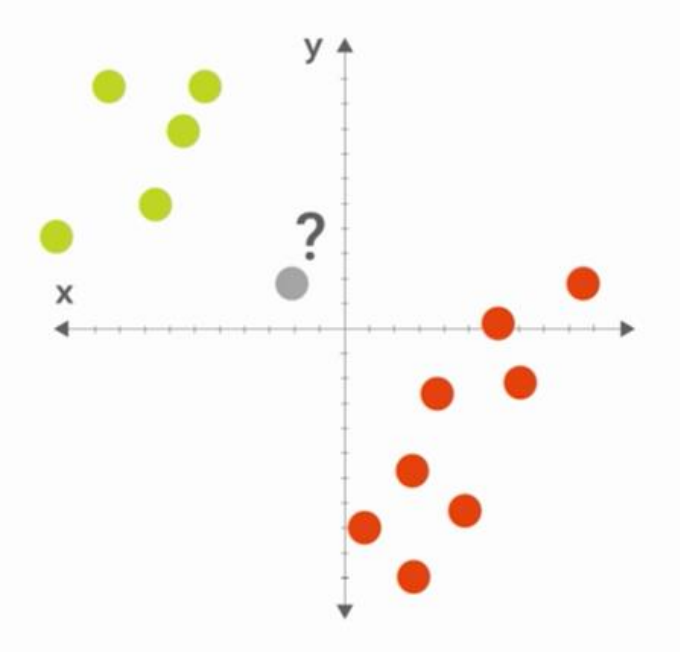
- Chương 1: Giới thiệu tổng quan về đề tài Cây quyết định cho bài toán phân lớp. Chương này giới thiệu mục tiêu nghiên cứu cũng như đối tượng, giới hạn và phương pháp nghiên cứu đề tài.
- Chương 2: Chương này giới thiệu về cơ sở lý thuyết của đề tài, tập trung giải thích các khái niệm, các thuật toán, thư viện và các công nghệ được sử dụng trong suốt quá trình nghiên cứu về cây quyết định và khảo sát các công trình nghiên cứu liên quan từ đó tạo ra tiền đề để áp dụng vào các bài toán nghiên cứu.
- Chương 3: Phương pháp đề xuất. Đây là chương trọng tâm của bài vì chương này sẽ mô tả chi tiết hơn về đề tài nghiên cứu dưới dạng bài toán thực tế, kết hợp với cơ sở lý thuyết ở chương trước đó.
- Chương 4: Kết quả nghiên cứu và kết luận. Là kết quả nghiên cứu cuối cùng sau khi so sánh với kết quả từ thư viện Scikit-Learn, từ đó rút ra được những ưu điểm cũng như khuyết điểm của phương pháp đề xuất nêu trên.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Khái niệm bài toán phân lớp

2.1.1. Khái niệm [2]

Bài toán phân lớp (classification) là bài toán phổ biến trong lĩnh vực Machine Learning (ML). Bài toán này là quá trình phân lớp hay phân loại một đối tượng dữ liệu vào một hay nhiều lớp đã cho trước nhờ một mô hình phân lớp được dựng lên (model). Mô hình này được tạo ra dựa trên một tập dữ liệu đầu vào có gán nhãn (label) hay còn gọi là tập huấn luyện. Quá trình phân lớp là quá trình gán nhãn có sẵn từ tập huấn luyện cho đối tượng cần phân lớp.



Hình 2.1 Bài toán phân loại bi [2]

Ở hình 2.1 là biểu đồ có các viên bi có 2 màu khác nhau (hay 2 loại khác nhau), trong đó có viên bi xám (hay dữ liệu đầu vào cần phân lớp) được thêm vào. Và nhiệm vụ của chúng ta là tìm ra viên đó thuộc màu gì.

Như vậy, bước đầu tiên là tìm ra mô hình phân lớp để khi có một dữ liệu đầu vào mới thì có thể xác định được dữ liệu đó thuộc phân lớp nào. Có nhiều bài toán về phân lớp với các loại phân lớp nhau như phân lớp nhị phân (binary), phân lớp đa trị hay phân lớp đa lớp (MultiClass).

Trong đó, bài toán phân lớp nhị phân là bài toán gắn nhãn cho đối tượng vào một trong hai lớp khác nhau dựa vào dữ liệu đó có hay không có đặc trưng (feature) của bộ phân lớp. Bài toán phân lớp đa lớp là quá trình phân lớp dữ liệu tương tự như phân lớp nhị phân nhưng với số lớp lớn hơn hai. Như vậy về thực chất bài toán phân lớp nhị phân là một bài toán nhỏ của bài toán phân lớp đa lớp. Bài toán này được ứng dụng rất rộng rãi trong thực tiễn, ví dụ như nhận diện giọng nói, nhận diện gương mặt, nhận diện vân tay, phát hiện những email rác hay nhận diện mống mắt…

Hình 2.2 Ví dụ về bài toán phân loại email rác

2.1.2. Quá trình phân lớp dữ liệu

Để dựng nên một mô hình phân lớp cần phải thực hiện các bước sau:

Bước 1: Chuẩn bị tập dữ liệu đầu vào để huấn luyện (train) và rút trích các đặc trưng.

Đây là bước khá quan trọng trong các bài toán phân lớp trong Machine Learning vì đây là dữ liệu dùng để train có ảnh hưởng đến kết quả phân lớp. Cho nên tập dữ liệu này cần phải lọc ra những đặc trưng quan trọng riêng biệt, loại bỏ những đặc trưng chung, mơ hồ, gây nhiễu (noise). Ước lược số chiều của dữ liệu là bao nhiêu cho hợp lý (chọn bao nhiêu đặc trưng). Nếu số chiều quá lớn có thể gây khó khăn trong quá trình tính toán thì có thể giảm số chiều lại nhưng vẫn giữ được độ chính xác (reduce dimension).

Ở bước này chúng ta cũng chuẩn bị tập dữ liệu để kiểm tra mô hình phân lớp. Thông thường sẽ sử dụng phương pháp kiểm tra chéo (cross-validation) để chia tập dữ liệu đầu vào thành hai tập dữ liệu, một tập để huấn luyện (training data) và một tập để kiểm tra như đã nêu trên (testing data). Trong cross-validation thì splitting và k-fold được sử dụng nhiều nhất.

Bước 2: Xây dựng mô hình phân lớp (classifier model)

Mô hình này mục đích là để tìm hàm f(x) và thông qua hàm f đó có thể gán label cho dữ liệu. Bước này là bước huấn luyện cho máy học (training).

$$f(x) = y$$

Trong đó:

+ x là các đặc trưng của dữ liệu.

+ y là nhãn gán cho dữ liệu đầu ra.

Các thuật toán thường được sử dụng để xây dựng mô hình phân lớp cho bài toán này là các thuật toán học giám sát (supervised learning) như K-nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes và Decision Tree.

Bước 3: Kiểm tra dữ liệu với mô hình (Prediction)

Sau khi xây dựng được mô hình phân lớp ở bước 2, thì bước này sẽ đưa tập dữ liệu kiểm tra để kiểm tra mô hình phân lớp.

Bước 4: Đánh giá mô hình và chọn ra mô hình tốt nhất

Bước cuối cùng để đánh giá mô hình là tính toán tỷ lệ chính xác khi đưa dữ liệu vào kiểm tra từ đó đánh giá được mức độ tối ưu của thuật toán sử dụng trong quá trình xây dựng mô hình. Và từ đó chọn ra mô hình tốt nhất để sử dụng vào bài toán phân lớp.

2.2. Các công trình nghiên cứu về bài toán phân lớp

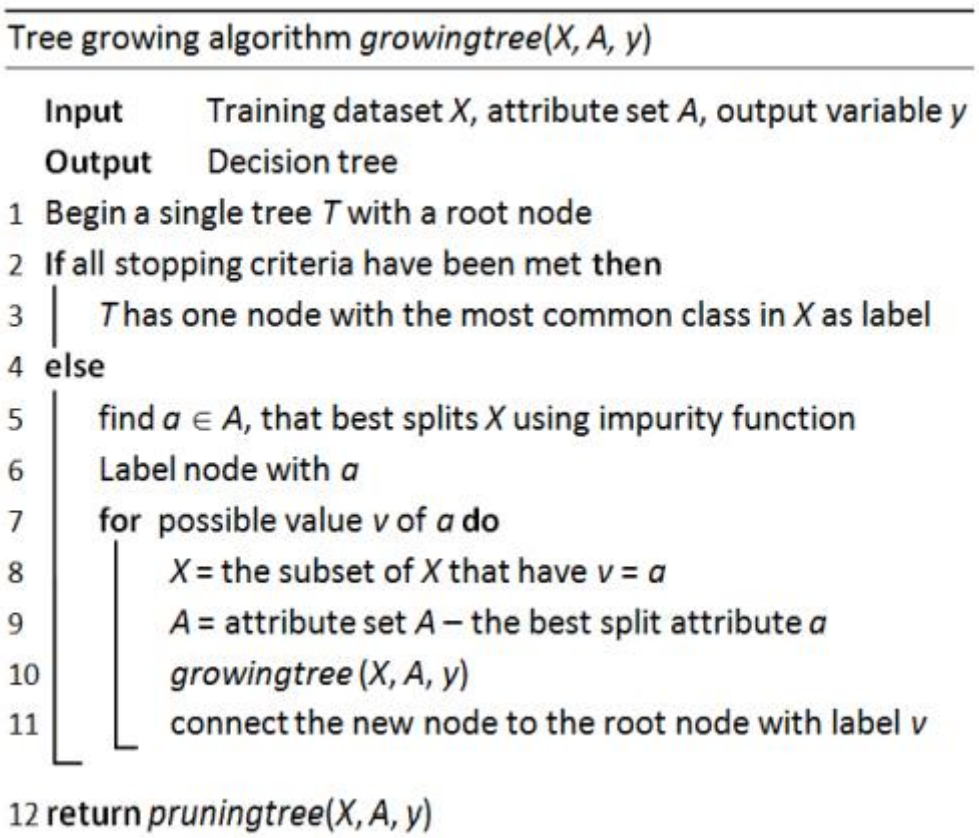
2.2.1. Báo cáo nghiên cứu về bài toán dự đoán kết quả học tập sử dụng thuật toán CART. [3]

Bài viết trình bày báo cáo về việc phân tích các dữ liệu được trích xuất trên các khóa học kết hợp trên nền tảng học trực tuyến Moodle. Bài viết này được đề cập trong hội nghị về Giáo dục hiện đại và Khoa học máy tính bởi tác giả Nick Z. Zacharis.

Nghiên cứu này sử dụng thuật toán CART để phân loại và dự đoán những sinh viên có nguy cơ bỏ học dựa trên 4 hoạt động chính của sinh viên như: trao đổi tin nhắn, tạo nội dung trên Wiki, mở các file trên khóa học, làm các bài kiểm tra trực tuyến.

Điểm bình quân (GPA), điểm môn học, các bài kiểm tra, các hoạt động trong phòng thí nghiệm, điểm chuyên cần cũng như các yếu tố về nhân khẩu học như tên, tuổi, giới tính, nền tảng gia đình và thói quen của sinh viên là những yếu tố quan trọng và thường được dùng để nghiên cứu về dự đoán hiệu suất của sinh viên.

Tệp dữ liệu được tác giả sử dụng bao gồm nhật kí tương tác của 134 sinh viên bao gồm các thông tin về khóa học, thời lượng tham gia các hoạt động và số lần đăng nhập vào LMS. Dùng 14 thuộc tính để tổng kết được các mối tương quan đặc trưng giữa điểm số được dùng như các biến độc lập trong qui trình hồi qui đa biến. Các thuộc tính được dùng để dự đoán về đầu ra của sinh viên là số lượng tin nhắn hay lượt xem tin nhắn, số lượng các câu hỏi đã trả lời, số lượng tập tin đã xem và số lần tham gia vào các bài tập nhóm.



Hình 2.3 Trình tự tạo cây bằng thuật toán CART

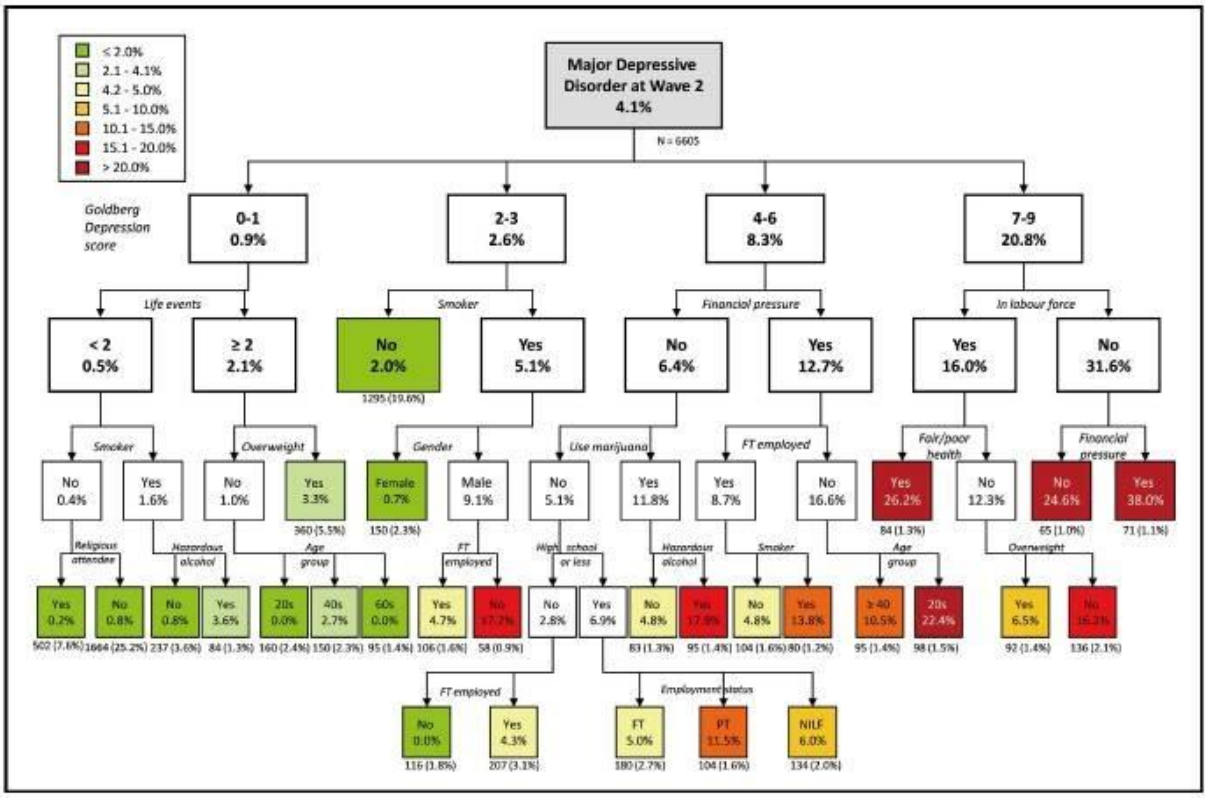
Nghiên cứu cho thấy mô hình đạt được độ chính xác lên đến 99.1%, điều này có thể chứng minh được rằng mô hình này có thể dự đoán những nhóm đối tượng sinh

viên có nguy cơ bỏ học nhằm đưa ra những biện pháp và hành động cụ thể để ngăn ngừa cũng như được các trường hợp sinh viên bỏ học.

2.2.2. Báo cáo nghiên cứu về hội chứng rối loạn trầm cảm nặng (MDD) [4]

Trong công trình nghiên cứu về hội chứng rối loạn trầm cảm ưu thế có sử dụng thuật toán cây quyết định để chỉ ra những yếu tố liên quan đến hội chứng MDD. Và thuật toán này được đề cập trong phần "Decision Tree methods: Applications for Classification and Prediction" của bài báo nghiên cứu.

Tác giả của bài nghiên cứu là Tiến sĩ Yan-yan Song, giảng viên Khoa sinh học tại Trường Y Đại học Giao thông Thượng Hải, hiện là học giả thỉnh giảng tại Phòng Thống kê Sinh học, Khoa Nghiên cứu và Chính sách về Y tế, Trường Đại học Y dược Stanford. Cô và cộng sự của cô, Ying LU, đã sử dụng thuật toán cây quyết định để phân tích các yếu tố nguy cơ liên quan đến rối loạn trầm cảm (MDD) trong một nghiên cứu dài hạn kéo dài 4 năm. Mục tiêu của cuộc nghiên cứu, phân tích này là xác định các yếu tố nguy cơ quan trọng nhất từ nhóm 17 yếu tố nguy cơ tiềm ẩn, bao gồm giới tính, tuổi tác, hút thuốc lá, tăng huyết áp, giáo dục, việc làm, các sự kiện trong đời, v.v. Mô hình cây quyết định được tạo từ tập dữ liệu được thể hiện trong hình dưới đây.



Hình 2.4 Mô hình cây quyết định dự đoán các yếu tố nguy cơ mắc chứng rối loạn trầm cảm [4]

Các cá thể sẽ được chia thành 28 phân nhóm từ nút gốc đến nút lá thông qua các nhánh khác nhau. Nguy cơ mắc chứng rối loạn trầm cảm chênh lệch từ 0 đến 38%. Ví dụ trong hình 2.3 ta sẽ thấy chỉ 2% những người không hút thuốc lá ở thời điểm ban đầu mắc MDD 4 năm sau đó, nhưng 17.2% là nam giới và cả hút thuốc là những người có điểm số 2 hoặc 3 trong thang điểm trầm cảm Goldberg và không hề có công việc chính thức hay toàn thời gian lúc ban đầu mắc MDD ở lần đánh giá theo dõi 4 năm.

Bằng cách sử dụng loại mô hình cây quyết định này, các nhà nghiên cứu có thể xác định sự kết hợp của các yếu tố nguy cơ tạo thành rủi ro cao nhất (hoặc thấp nhất) đối với một điều kiện nhất định, từ đó đưa ra những số liệu để dự đoán khả năng mắc hội chứng này.

2.2.3. Báo cáo nghiên cứu về chẩn đoán sớm bệnh ung thư vú [5]

Ung thư vú là loại ung thư nguy hiểm nhất trong số các loại ung thư dẫn đến tử vong ở phụ nữ. Nó là nguyên nhân tử vong thứ hai ở hầu hết phụ nữ đặc biệt là ở một số nước phát triển. Bệnh này không thể được chẩn đoán dễ dàng bằng các xét nghiệm

thông thường trong phòng thí nghiệm và rất khó để xác định được bệnh ở giai đoạn sớm. Ngoài ra khả năng tái phát của ung thư vú cao. Bài báo cụ thể này do tiến sĩ M Indra Devi hiện là giáo sư của trường Đại học Kỹ thuật Kamaraj, Virudhunagar, Ấn Độ và R Delshi Howsalya Devi, trợ lý giáo sư của trường Đại học Kỹ thuật K.L.N Sivagangai, Ấn Độ. Họ đã cùng nhau nghiên cứu ra phương pháp chẩn đoán căn bệnh ung thư vú bằng thuật toán cây quyết định.

Trong bài báo cáo, bước đầu tiên để nghiên cứu là tiến hành gom nhóm dữ liệu thành số bằng cách sử dụng thuật toán phân cụm xa nhất đầu tiên (Farthest First clustering algorithm). Do thu nhỏ được kích thước của tập dữ liệu nên thời gian tính toán giảm đáng kể. Bước thứ hai, xác định dữ liệu nhiễu từ tập dữ liệu ung thư vú bằng cách sử dụng Outlier Detection Algorithm (ODA). Trong bước thứ ba, xác định ung thư là lành tính hay ác tính từ tập dữ liệu được xử lý trước đó bằng thuật toán phân loại J48 của Decision Tree.

```
INPUT:     D  //Training data
OUTPUT:  T  //Decision tree
DTBUILD (*D){
T=φ;
T= Create root node and label with splitting attribute;
T= Add arc to root node for each split predicate and label;
For each arc do
D= Database created by applying splitting predicate to D;
If stopping point reached for this path, then
T'= create leaf node and label with appropriate class;
Else
T'= DTBUILD(D);
T= add T' to arc;
```

Hình 2.5 Trình tự xây dựng cây của thuật toán J48 [5]

Trong quá trình nghiên cứu, bộ dữ liệu ung thư vú Wisconsin (WBCD) và Chẩn đoán ung thư vú Wisconsin (WDBC) đã được sử dụng để kiểm tra hiệu quả của hệ thống đề xuất. Kết quả thử nghiệm chứng minh rằng hai bước được đề xuất là cách nghiên cứu tốt nhất với độ chính xác cao nhất là 99.9% đối với bộ dữ liệu WBCD và độ chính xác là 99,6% đối với bộ dữ liệu WDBC so với hiện tại nghiên cứu cho cùng một tập dữ liệu.

Nghiên cứu này sẽ giúp các bác sĩ chẩn đoán ung thư vú và từ đó giúp bệnh nhân đang mắc bệnh dần hồi phục.

2.3. Một số kiến thức liên quan

2.3.1. Một số thư viện trong Python

Ngôn ngữ lập trình Python là ngôn ngữ lập trình theo hướng diễn giải và là một trong những ngôn ngữ lập trình cấp cao, đa nhiệm, được chính thức ra mắt vào năm 1991 bởi Guido van Rossum. Python có thể hoạt động trên đa nền tảng như: lập trình Web, Phân tích số liệu, xử lý toán học, lập trình robotics, tự động hóa nhờ vào kho thư viện đa dạng và luôn được cập nhật và phát triển bởi cộng đồng người sử dụng ngôn ngữ Python.

- **Numpy [6]:** hỗ trợ lập trình trên các ma trận lớn, đa chiều và cung cấp một số các hàm toán học cao cấp để hỗ trợ các hoạt động trên nền tảng toán học.
  - **numpy.unique**: hàm trả về đại diện của phần tử trong bộ dữ liệu.

    vd:

    ```
    import numpy as np
    array = [1,2,3,1,2,2,3,5,6,4,4,6,5,3,2,8,9,1,5,2,8,4,6,9,7]
    print(np.unique(array))
    ```

    ```
    [1 2 3 4 5 6 7 8 9]
    ```

  - **numpy.argmax**: trả về chỉ mục của phần tử có giá trị lớn nhất trong mảng

    vd:

    ```
    import numpy as np
    array = [1,2,3,1,2,2,3,5,6,4,4,6,5,3,2,8,9,1,5,2,8,4,6,9,7]
    print('unique:',np.unique(array))
    print('max:',np.argmax(array))
    ```

    ```
    unique: [1 2 3 4 5 6 7 8 9]
    max: 16
    ```

  - và một số hàm tính toán học được dùng trong bài như sum, log2.
- **Pandas [7]:** thư viện hỗ trợ thao tác và phân tích dữ liệu, được xây dựng với mục đích có thể hoạt động mạnh mẽ nhất với các dạng cấu trúc dạng bảng, đa chiều và có các kiểu dữ liệu không đồng nhất.
  - **pandas.read_csv**: cho phép đọc dữ liệu từ file (*.csv) và tạo thành một bảng dữ liệu DataFrame

- **pandas.DataFrame**: kiểu dữ liệu 2 chiều, kích thước linh hoạt và kiểu dữ liệu trong bảng có thể là không đồng nhất.
- **pandas.DataFrame.iloc**: lấy dữ liệu tại vị trí cụ thể được nhận trong chỉ mục.
- **pandas.DataFrame.loc:** lấy dữ liệu với nhãn dán trong chỉ mục
- **pandas.DataFrame.dropna:** tự động xóa những dòng dữ liệu bị thiếu giá trị.
    - **pandas.DataFrame.columns:** lấy cột dữ liệu trong bảng dữ liệu.
- **Scikit-learn [8]:** thư viện hỗ trợ về mảng máy học, bao gồm các thuật toán về support vector machine. random forests, và k-neighbours,...cũng như các hàm hỗ trợ về số học và một số thư viện hỗ trợ nghiên cứu về khoa học như *NumPy* và *SciPy.*
    - sklearn.tree.DecisionTreeClassifier: thư viện hỗ trợ tạo mô hình phân lớp dưới dạng cây quyết định dựa trên *chỉ số gini* hoặc *chỉ số entropy*

2.3.2. Một số thuật toán

2.3.2.1. Thuật toán ID3 [9]

ID3 (Iterative Dichotomiser 3) được phát triển vào năm 1986 bởi Ross Quinlan. Sử dụng lượng thông tin ứng với biến số phân loại sau đó dùng kỹ thuật tham lam (greedy).

Trong ID3, chúng ta cần xác định thứ tự của thuộc tính cần xem xét tại mỗi bước. Với các bài toán có nhiều thuộc tính và mỗi thuộc tính có nhiều giá trị khác nhau, việc tìm được nghiệm tối ưu thường là không khả thi. Thay vào đó một phương pháp đơn giản thường được sử dụng là tại mỗi bước, một thuộc tính *tốt nhất* sẽ được chọn trên một tiêu chuẩn. Với mỗi thuộc tính được chọn, ta chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính đó rồi áp dụng phương pháp với mỗi nút con.

2.3.2.2. Thuật toán C4.5 [10]

Thuật toán C4.5 được Ross Quinlan mở rộng trên chính mô hình ID3 của mình để tạo mô hình cây quyết định. Thuật toán C4.5 sử dụng phương pháp đệ quy đến từng node của cây quyết định, chọn các nhánh có thể phân tách đến khi không thể phân tách được nữa. Một số đặc điểm của C4.5:

- Thuật toán C4.5 không bị ràng buộc bởi phân tách nhị phân có thể tạo ra một cây có nhiều nhánh trên một nút.
- Đối với các thuộc tính dạng phân loại, mặc định cây sẽ tách nhánh cho mỗi giá trị trong bảng. Nó có thể dẫn đến kết quả mô hình cây sẽ phức tạp hơn cần thiết bởi một số node có ít thông tin cần thiết và không liên kết đến các giá trị khác.
- Cách tính giá trị cho mỗi nút của thuật toán C4.5 là đồng nhất.

Thuật toán này dùng cách tính *Information Gain* và *Entropy Reduction* để tối ưu hóa cách phân nhánh.

2.3.2.3. Công thức đo lường [11]

Để chọn ra thuộc tính nào có thể tạo ra một bộ dữ liệu tinh khiết, ta cần một số công thức nhằm tính ra độ tinh khiết của bộ dữ liệu. Trong suốt quá trình tạo cây, ta cần phải tính những thông số này cho từng thuộc tính để tìm ra đâu là thuộc tính tối ưu nhất khi tạo các nhánh của cây.

- **Entropy:** trong lĩnh vực thông tin, entropy được xem là thước đo độ phức tạp cũng như độ tiư `nh khiết của tập dữ liệu

  Công thức tính Entropy:

$$entropy(p) = -\sum_{i=1}^{N} p_i \times \log_2 p_i \qquad \textbf{[1]}$$

- **Information Gain (IG):** chỉ số nhiễu thông tin(độ giảm entropy là thấp nhất)

$$IG(S,f) = E(S) - E(f,S) \quad \textbf{[2]}$$

Trong đó:

E(S) là tổng entropy của toàn tập S

E(f,S) là entropy được tính trên thuộc tính f

- **Gain Ratio:** chuẩn hóa Information Gain với giá trị thông tin cần phân tách (**Split Info**)

$$Split\ Info = -\sum_{i=1}^{N} D_i \times \log_2 D_i \quad \textbf{[3]}$$

Trong đó: D_i là xác suất phần tử thứ i so với tập dữ liệu

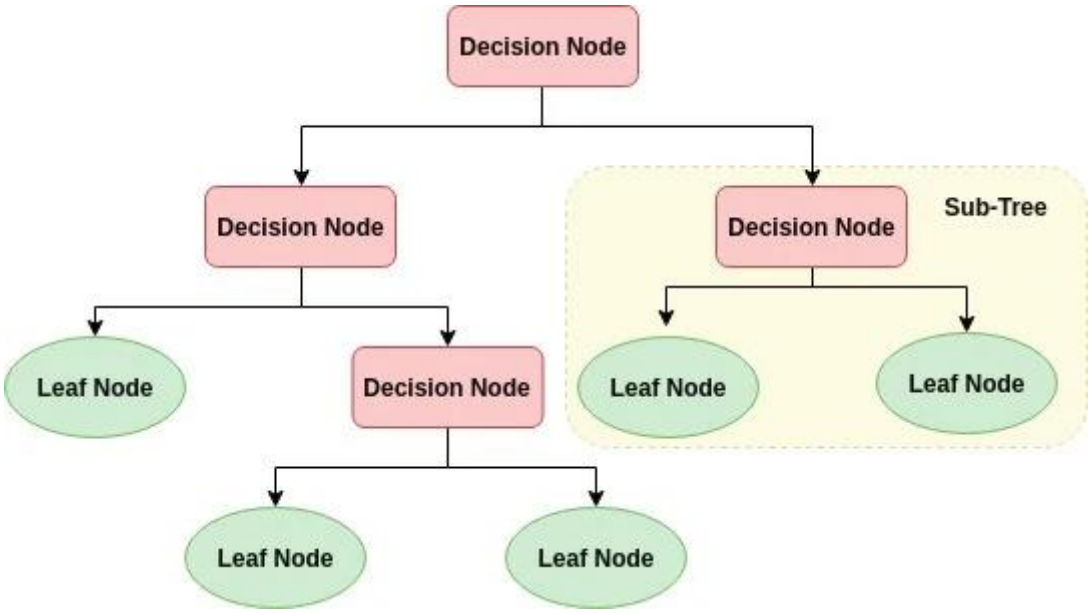
$$Gain\ Ratio = \frac{Information\ Gain}{Split\ Info} \quad \textbf{[4]}$$

2.3.3. Cây quyết định (Decision Tree) [11]

Cây quyết định được biết đến là một trong những công cụ mạnh và phổ biến nhất trong phân lớp máy học có giám sát, được sử dụng phổ biến trong Mô hình phân lớp(Classification Model) và Mô hình hồi quy (Regression Model). Được xây dựng thành cấu trúc có hình dạng tương tự một lưu đồ dạng cây với mỗi node trong cây được xem là một thuộc tính, mỗi nhánh đại diện cho dữ liệu của một thuộc tính, mỗi node lá là một phân lớp.Cấu trúc cây quyết định.

2.3.3.1. Cấu trúc cây quyết định

Cấu trúc của một cây quyết định bao gồm các nút (node) và các nhánh của nó. Nút dưới cùng được gọi là nút lá (leaf node) cũng chính là phân loại cuối cùng của cây dưới dạng yes/no. Các nút khác nút lá được gọi là các nút con hay còn gọi là nút quyết định (Decision node), các nút này đảm nhận vai trò phân nhánh các dữ liệu theo một thuộc tính (attribute) của tập dữ liệu. Mỗi một nhánh của cây xuất phát từ một nút p nào đó ứng với một phép so sánh dựa trên miền giá trị của nút đó.



Hình 2.6 Decision Tree [12]

Để xây dựng nên Decision Tree, thuật toán sẽ gồm các bước như sau:

- Chọn lựa thuộc tính của dữ liệu để phân nhánh dữ liệu bằng Attribute Selection Measures (ASM: Chỉ số đánh giá lựa chọn thuộc tính).
- Tạo các Decision node với các feature và điều kiện trên.
- Chọn ra một thuộc tính để làm nút gốc (node root).
- Phân nhánh dữ liệu tạo các node con và lặp lại tiến trình ở trên (node root chỉ tạo một lần) cho đến khi một trong các điều kiện sau thỏa mãn ta sẽ có được node lá:
    - Tất cả dữ liệu của node đều thỏa mãn điều kiện của node quyết định.
    - Không có thuộc tính với điều kiện nào có thể được chọn nữa.
    - Không còn dữ liệu nào thỏa mãn điều kiện của node quyết định.

2.3.3.2. Mục tiêu của cây quyết định

Cây quyết định được sử dụng để tạo ra các mô hình dữ liệu sẽ dự đoán các lớp hay các nhãn cho quá trình ra quyết định. Áp dụng cây quyết định giúp cho chúng ta dễ hình dung và dễ hiểu hơn dó đó nó là thuật toán phổ biến trong kỹ thuật khai thác dữ liệu.

2.3.3.3. Phương thức hoạt động của cây

Cây quyết định là một thuật toán có giám sát hoạt động cho cả biến rời rạc hay biến liên tục. Nó chia tập dữ liệu thành các tập con trên cơ sở thuộc tính quan trọng nhất trong tập dữ liệu. Việc cây quyết định xác định thuộc tính này như thế nào và việc phân tách này được thực hiện như thế nào là do các thuật toán quyết định.

Cây quyết định phân loại các lớp bằng cách sắp xếp các dữ liệu từ nút gốc đến nút lá. Cây phân loại một trường hợp bắt đầu từ nút gốc của cây. Sau đó, di chuyển xuống nhánh tương ứng với thuộc tính được chỉ định của nút quyết định. Quá trình này sẽ lặp lại liên tục cho các cây con cho đến khi phân lớp toàn bộ dữ liệu.

Để làm rõ hơn về phương thức hoạt động của cây quyết định, ta có ví dụ về quyết định có đi chơi hay không và mối quan hệ về thời tiết trong 14 ngày. (Bảng dữ liệu được tham khảo trong sách Data Mining: Practical Machine Learning Tools and Techniques.

outlook	**temperature**	**humidity**	**wind**	**play?**
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes

Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Bảng 2.1 Bảng dữ liệu thời tiết

Bảng có 4 thuộc tính:

- outlook có 3 giá trị: sunny, overcast, rainy
- temperature có 3 giá trị: hot, mild, cool
- humidity có 2 giá trị: high, normal
- wind có 2 giá trị: True, False

Và có 2 giá trị đầu ra: Yes, No

Bài toán yêu cầu đưa ra các dự đoán về việc có đi chơi hay không dựa vào các yếu tố thời tiết.

Đánh giá khách quan vào việc nhìn vào bảng dữ liệu, ta có thể thấy:

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

Sử dụng thuật toán ID3, để tạo cây, gọi bảng dữ liệu là tập S, ta có:

$$\textbf{\textit{entropy(S)}} = -\frac{9}{14} log_2 \frac{9}{\mathbf{14}} - \frac{5}{14} log_2 \frac{5}{14} \approx 0.94$$

$$\textbf{\textit{information gain(f) = E(S) - E(f)}}$$

trong đó: *S* là tập dữ liệu

f là thuộc tính

I_sunny= $-(\frac{2}{5} log_2 \left(\frac{2}{5}\right)) + (-(\frac{3}{5} log_2 \left(\frac{3}{5}\right)) = 0.97$
I_overcast= $-(1 log_2(1)) + (-(0 log_2(0)) = 0$
I_rainy= $-(\frac{3}{5} log_2 \left(\frac{3}{5}\right)) + (-(\frac{2}{5} log_2 \left(\frac{2}{5}\right)) = 0.97$

outlook	**yes**	**no**	**entropy**
sunny	2	3	0.97
overcast	4	0	0
rainy	3	2	0.97

Bảng 2.2 Chỉ số entropy của các giá trị trong outlook

E(outlook)= $\frac{5}{14} * 0.97 + \frac{4}{14} * 0 + \frac{5}{14} * 0.97 \approx 0.69$

IG (outlook) = 0.94-0.69 = 0.25

Tương tự với các thuộc tính còn lại ta có bảng sau:

f	**outlook**	**temperature**	**humidity**	**wind**
E (f, S)	0.69	0.89	0.79	0.81
IG (f, S)	0.25	0.05	0.15	0.13

Bảng 2.3 Chỉ số Information Gain của từng thuộc tính trong tập S (thời tiết)

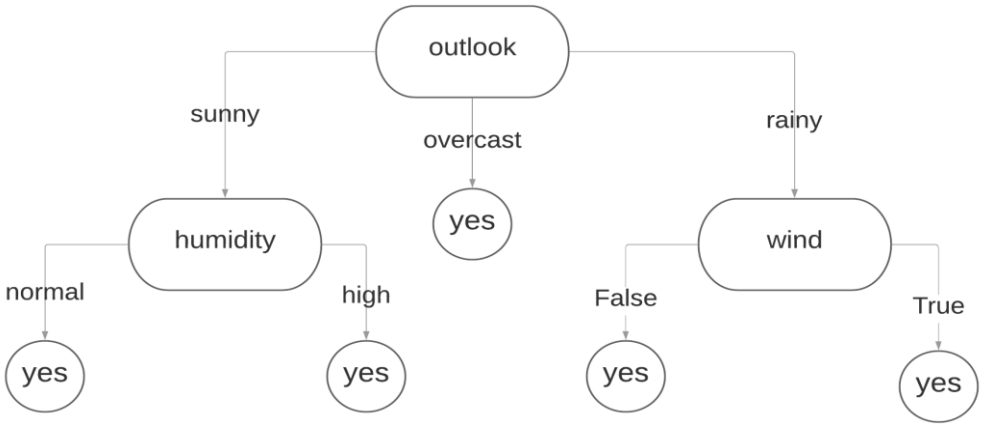
Dựa vào bảng trên, ta có thể thấy với chỉ số IG (outlook) là cao nhất, ta sẽ

chọn outlook để làm nút gốc (node root) với 3 nhánh là sunny, overcast và rainy. Lúc này ta lại tiếp tục, tìm nút có IG cao nhất với E(S) là tổng trọng số entropy của E(outlook=sunny) = 0.97, ta có bảng sau:

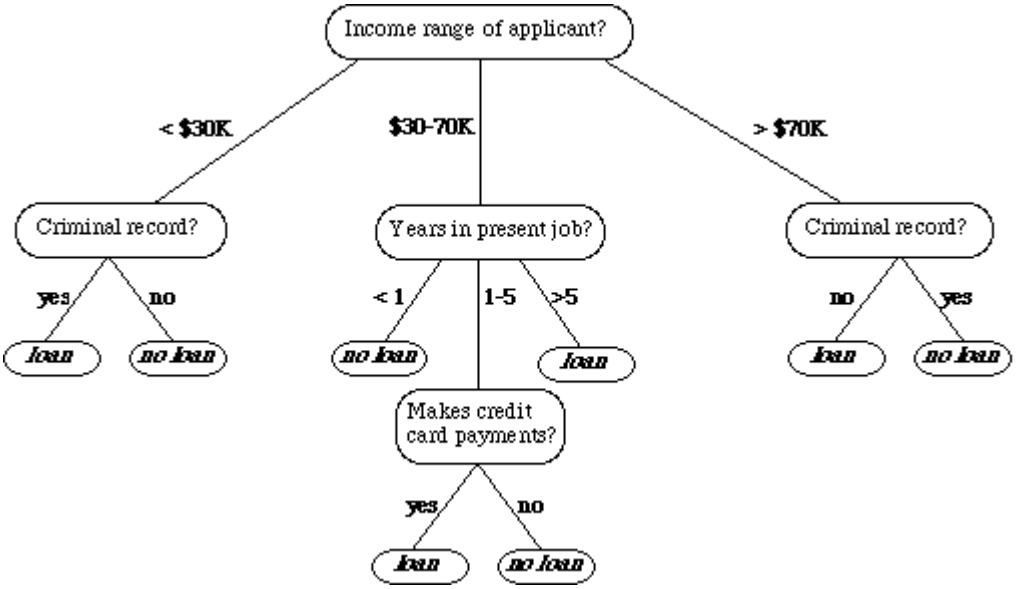
f	**temperature**	**humidity**	**wind**
E (f, outlook=sunny)	0.4	0	0.946
IG (f, outlook=sunny)	0.57	0.97	0.024

Bảng 2.4 Chỉ số IG của từng thuộc tính trong tập outlook = sunny

Do đó, với nhánh *outlook = sunny* ta tạo cây con với nút gốc là *humidity.* Và tương tự với các nhánh còn lại, ta tạo được cây quyết định như sau:



Hình 2.7 Cây quyết định của bài toán với tập dữ liệu S



Hình 2.8 Ví dụ về cây quyết định vay vốn ngân hàng [13]

Chương 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Tập dữ liệu nghiên cứu

Để quản lý hiệu quả các lớp thú trong sở thú cần một hệ thống xác định phân loài của từng loại động vật trong sở thú nhằm có các biện pháp chăm sóc phù hợp với từng loài.

Với bộ dữ liệu dưới dạng "csv" gồm 101 dòng với 16 thuộc tính bao gồm 7 phân lớp sẽ áp dụng thuật toán C4.5 để phân tích, huấn luyện, tạo cây quyết định và thử nghiệm độ chính xác của cây.

Bộ dữ liệu bao gồm:

- Phân lớp: gồm 7 phân lớp

1. Mammal (41 dòng dữ liệu)
2. Bird (20 dòng dữ liệu)
3. Reptile (5 dòng dữ liệu)
4. Fish (13 dòng dữ liệu)
5. Amphibian (4 dòng dữ liệu)
6. Bug (8 dòng dữ liệu)
7. Invertebrate (10 dòng dữ liệu)

- Thuộc tính: gồm 16 thuộc tính.

Hair, Feathers, Eggs, Milk, Airborne, Aquatic, Predator, Toothed, Backbone, Breathes, Venomous, Fins, Legs, Tail, Domestic, Catsize.

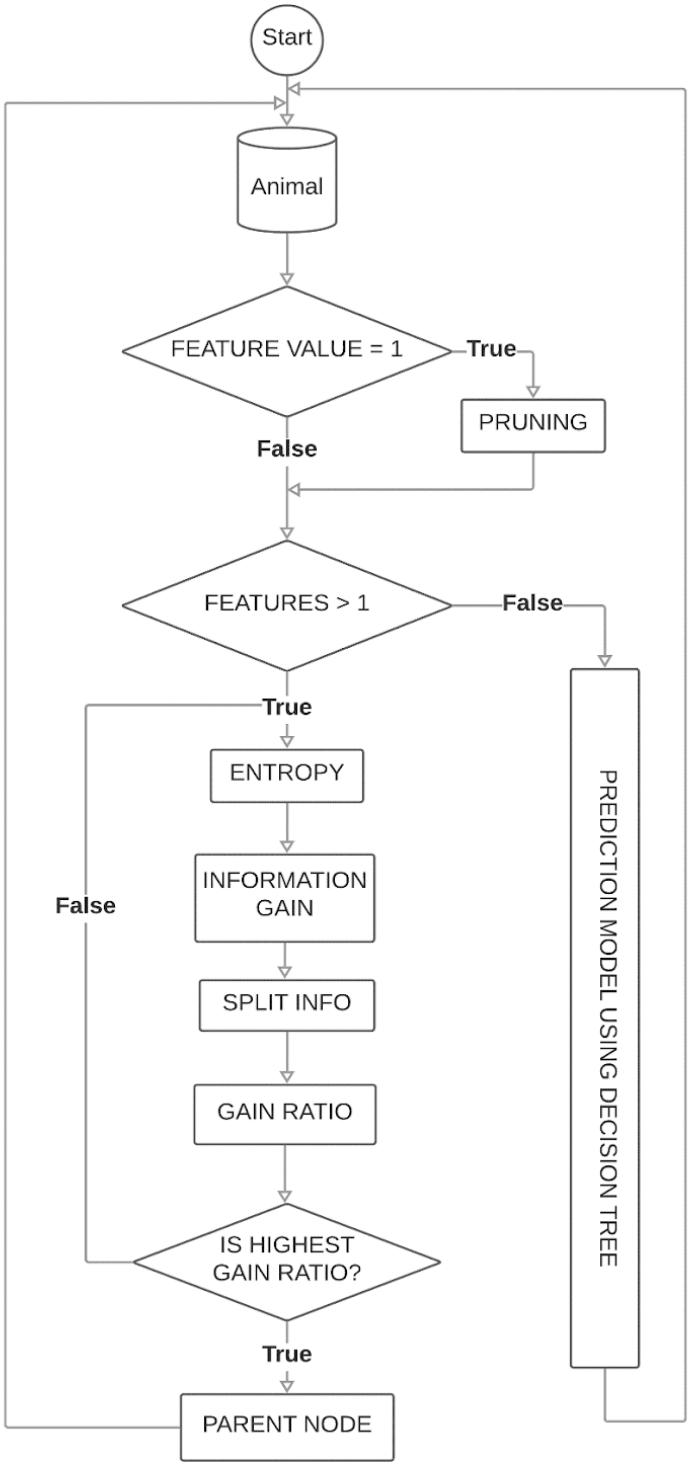
Hầu hết các thuộc tính trong bảng dữ liệu có 2 giá trị: 0(Không) và 1(Có). Riêng với thuộc tính Legs đại diện cho số chân của động vật sẽ có nhiều giá trị như 0,2,4,5,6,8.

Bộ dữ liệu này được lấy từ bộ dữ liệu trong sở thú của trang www.kaggle.com.

3.2. Phương pháp đề xuất

Để quản lý hiệu quả các lớp thú trong sở thú cần một hệ thống xác định phân loài của từng loại động vật trong sở thú nhằm có các biện pháp chăm sóc phù hợp với từng loài. Với bộ dữ liệu gồm 101 dòng với 16 thuộc tính bao gồm 7 phân lớp, nhóm em áp

dụng thuật toán C4.5 để phân tích, huấn luyện, tạo cây quyết định và thử nghiệm độ chính xác của cây.



Hình 3.1 Mô hình dự đoán sử dụng thuật toán cây quyết định

3.2.1. Xây dựng các hàm tính chỉ số

Entropy: để tính được entropy, ta cần tìm được số lượng từng phân loại tương ứng với 7 lớp

- Tạo mảng các giá trị trong cột class và đếm số lượng giá trị tương ứng với phân lớp đó.
- Tính tổng với công thức tính entropy [] với từng phần tử trong mảng giá trị trên.

S	1	2	3	4	5	6	7	E([], S)
	41	20	5	13	4	8	10	2.39

Bảng 3.1 Chỉ số Entropy của tập S

Feathers	1	2	3	4	5	6	7	E([], feathers)
[0]	41		5	13	4	8	10	2.08
[1]		20						0.0

Bảng 3.2 Chỉ số Entropy của thuộc tính Feathers

Milk	1	2	3	4	5	6	7	E([], milk)
[0]		20	5	13	4	8	10	2.39
[1]	41							0.0

Bảng 3.3 Chỉ số Entropy của thuộc tính Milk

Fins	1	2	3	4	5	6	7	E([], fins)
[0]	37	20	5		4	8	10	2.15
[1]	4			13				0.78

Bảng 3.4 Chỉ số Entropy của thuộc tính Fins

Information Gain: để tính được Information Gain ta cần tính được Entropy cho toàn tập dữ liệu và sử dụng để tính độ giảm entropy với từng thuộc tính.

- Cần tìm Weighted Entropy của thuộc tính là tổng entropy của từng giá trị trong thuộc tính.
- Tính Information Gain của từng thuộc tính là hiệu của entropy của toàn tập dữ liệu và entropy của từng thuộc tính.

- Ta tìm được bảng Information Gain như sau:

Information Gain							
Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Toothed
0.638	0.718	0.818	1.011	0.469	0.389	0.093	0.865

Information Gain							
Backbone	Breathes	Venomous	Fins	Legs	Tail	Domestic	Catsize
0.638	0.614	0.134	0.466	1.363	0.5	0.05	0.308

Bảng 3.5 Chỉ số Information Gain của 16 thuộc tính

- Dựa vào bảng trên, ta có thể thấy với thuộc tính Legs có chỉ số Information Gain cao nhất nên sẽ được chọn làm nút gốc trong thuật toán(sử dụng Information Gain làm thước đo).
- Do thuật toán ID3 chỉ tập trung vào các nút có nhiều giá trị phân tách nên sẽ ít xét đến các nút ít giá trị nhưng có độ bộ dữ liệu trong suốt(purity) hơn.
- Do đó, thuật toán C4.5 đã được xây dựng để cải thiện nhược điểm này để chọn ra thuộc tính tốt nhất ta dùng thước đo là hệ số Gain Ratio.

Split Info: hệ số trị thông tin phân tách với công thức đã được nêu ở chương 2 []

Gain Ratio: với các công thức đã được nêu ta tìm được bảng giá trị Gain Ratio

Gain Ratio							
Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Toothed
0.826	1.0	0.847	1.03	1.007	0.414	0.094	0.893

Gain Ratio							
Backbone	Breathes	Venomous	Fins	Legs	Tail	Domestic	Catsize
1.000..2	0.833	0.369	0.713	0.67	0.608	0.091	0.312

Bảng 3.6 Chỉ số Gain Ratio của 16 thuộc tính

Theo như bảng trên, với chỉ số Gain Ratio là lớn nhất so với các thuộc tính còn lại, ta chọn Milk làm nút gốc trong thuật toán C4.5.

3.2.2. Hàm xây dựng cây quyết định

Với các hàm tính chỉ số đã được xây dựng phía trên, ta áp dụng vào thuật toán xây dựng cây quyết định và sử dụng kỹ thuật đệ quy để tối ưu hoá bài làm.

Với mỗi lần đệ quy, ta lại chọn một nút gốc mới dựa trên bảng dữ liệu đã lọc và loại bỏ những nút đã được chọn làm nút gốc hoặc không mang đến giá trị để phân tách.

3.2.3. Hàm đánh giá kết quả

Với bộ dữ liệu test, ta áp dụng vào cây quyết định vừa xây dựng và so sánh với kết quả thực tế với kết quả sau khi áp dụng, từ đó thu được bảng so sánh như sau:

Số dòng	Thực tế	Dự đoán	Kết quả
81	3	3	TRUE
82	7	6	FALSE
83	4	4	TRUE
84	2	2	TRUE
85	1	1	TRUE
86	7	7	TRUE
87	4	4	TRUE
88	2	2	TRUE
89	6	6	TRUE
90	5	5	TRUE
91	3	5	FALSE
92	3	5	FALSE

93	4	4	TRUE
94	1	1	TRUE
95	1	1	TRUE
96	2	2	TRUE
97	1	1	TRUE
98	6	6	TRUE
99	1	1	TRUE
100	7	6	FALSE
101	2	2	TRUE
KẾT QUẢ: 17 TRUE/4 FALSE			

Bảng 3.7 Bảng dữ liệu dự đoán của dataset Animal

Độ chính xác của cây được tính dựa trên kết quả xác suất giữa 2 kết quả TRUE và FALSE, dựa theo bảng trên, độ chính xác của cây xấp sỉ 80,95% với công thức sau.

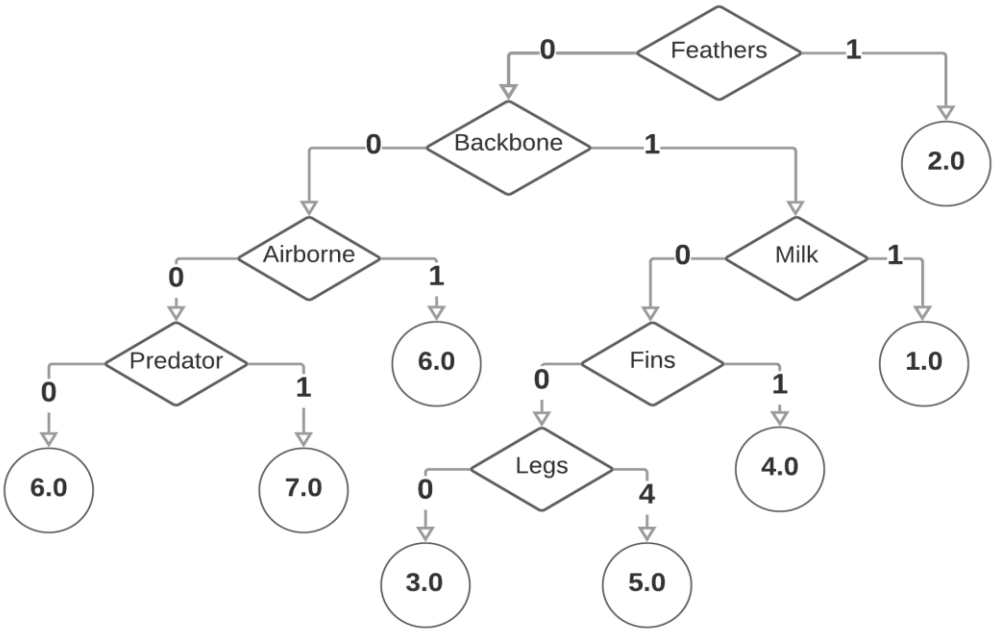
$$implement = \frac{\sum_{i=0}^{n}(predicted=true)}{\sum_{i=0}^{n} S}$$

Trong đó: *n* là số lượng dòng dữ liệu mà cột dự đoán mang giá trị True

S là tập dữ liệu

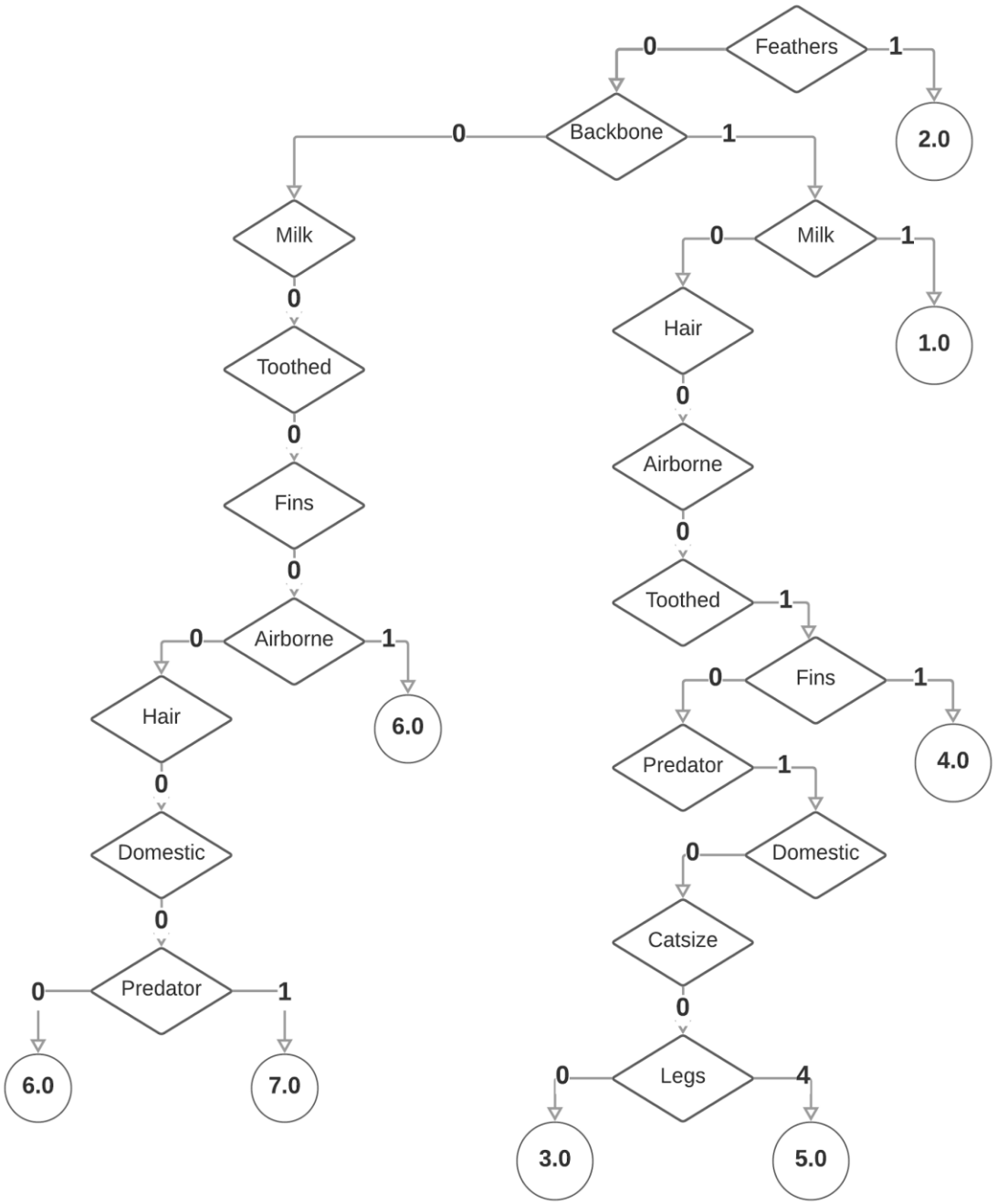
Pruning: Quá trình loại bỏ những nút không có giá trị phân tách thường được gọi là quá trình tỉa cây, quá trình này nhằm làm giảm độ phức tạp của cây tăng độ chính xác cho cây hơn.

- Cây đã được tỉa:



Hình 3.2 Decision Tree with pruning

- Cây chưa được tỉa:



Hình 3.3 Decision Tree without pruning

3.3. Phương pháp đánh giá

Đây là bước xây dựng cây quyết định bằng API với thư viện Scikit-Learn trong Python để xây dựng thêm một cây tương tự nhằm mục đích so sánh, đối chiếu kết quả đã nghiên cứu ở trên.

Bước 1: Nhập dữ liệu động vật

Các bước nhập (import) bảng dữ liệu, thư viện tương tự như các bước tính toán ở phương pháp C4.5.

Bước 2: Tách dữ liệu trong bảng

```
training_data = train_test_split(dataAni)[0]
testing_data = train_test_split(dataAni)[1]
```

Hình 3.4 Source Code tham khảo để tách cây bằng hàm có sẵn

Phương thức train_test_split là phương thức để tách dữ liệu làm 2 phần:

- Phần dữ liệu để huấn luyện (train) cho máy học (70% dữ liệu)
- Phần dữ liệu để kiểm tra (test) sau khi tạo cây (30% dữ liệu)

Bước 3: Tạo cây quyết định

Thư viện được sử dụng là *sklearn.tree.DecisionTreeClassifier* để tạo mô hình phân lớp dưới dạng cây quyết định dựa trên chỉ số entropy. Sau khi tách được dữ liệu thành 2 phần, ta tiến hành tách 2 phần dữ liệu đó thành 2 bảng là bảng features và bảng targets. Trong đó features là bảng gồm các số liệu của 16 đặc điểm của loài và targets là loài tương ứng với các đặc điểm đó. Sau đó tạo mô hình cây bằng phương thức tree.DecisionTreeClassifier với chiều sâu tối đa của cây (max_depth) là 5.

Tiếp theo áp bảng dữ liệu động vật và danh sách loài đối chiếu vào cây để chọn node là chia nhánh cho cây bằng hàm fit.

```
sklearn.tree.DecisionTreeClassifier(criterion='entropy'
train_features = dataAni.iloc[:80,:-1]
test_features = dataAni.iloc[80:,:-1]
train_targets = dataAni.iloc[:80,-1]
test_targets = dataAni.iloc[80:,-1]

clf = tree.DecisionTreeClassifier(max_depth=5)
clf = clf.fit(train_features, train_targets)
f_name = list(train_features.columns)
print(tree.plot_tree(clf, feature_names = f_name,
class_names = ["1:Mammal","2:Bird","3:Reptile",
               "4:Fish","5:Amphibian","6:Bug","7:Invertebrate"], filled = True))
```

Hình 3.5 Source code tham khảo tạo cây bằng scikit-learn

Cuối cùng dùng hàm tree.plot_tree để dựng cây.

Hình 3.6 Cây quyết định được dựng bằng thư viện matplotlib.pyplot

Tiếp theo sau đó dùng 30% dữ liệu test đã được tách ra trước đó để dự đoán loài động vật bằng hàm predicted.

Cuối cùng dùng công thức tính tỉnh lệ phần trăm:

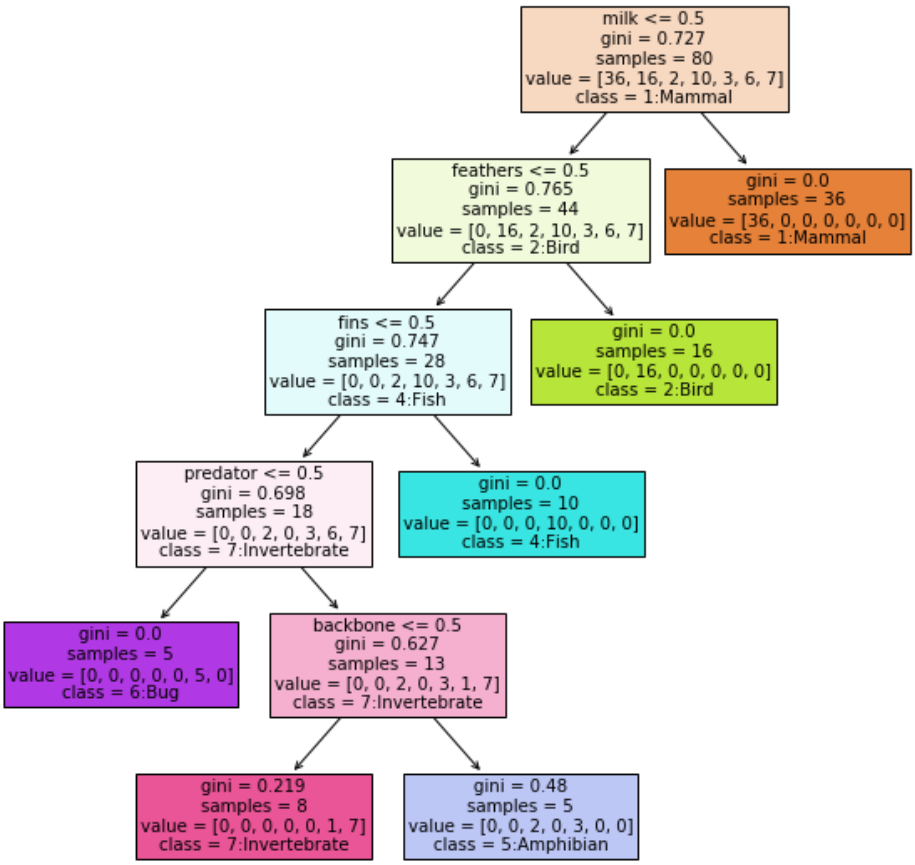
```
predicted = clf.predict(test_features)
print(predicted)

print ("Tỷ lệ dự đoán:",sum(predicted==test_targets)/float(len(test_targets))*100)
```

```
[5 7 4 2 1 7 4 2 7 5 7 5 4 1 1 2 1 6 1 7 2]
Tỷ lệ dự đoán: 80.95238095238095
```

Hình 3.7 Source code dự đoán

Dựa vào thư viện Scikit-Learn có sẵn để tạo cây thì tỷ lệ dự đoán chính xác là gần 81%, nhưng kể từ đợt huấn luyện thứ 2 trở đi thì tỷ lệ lại thay đổi với độ chính xác thấp hơn là 71%.



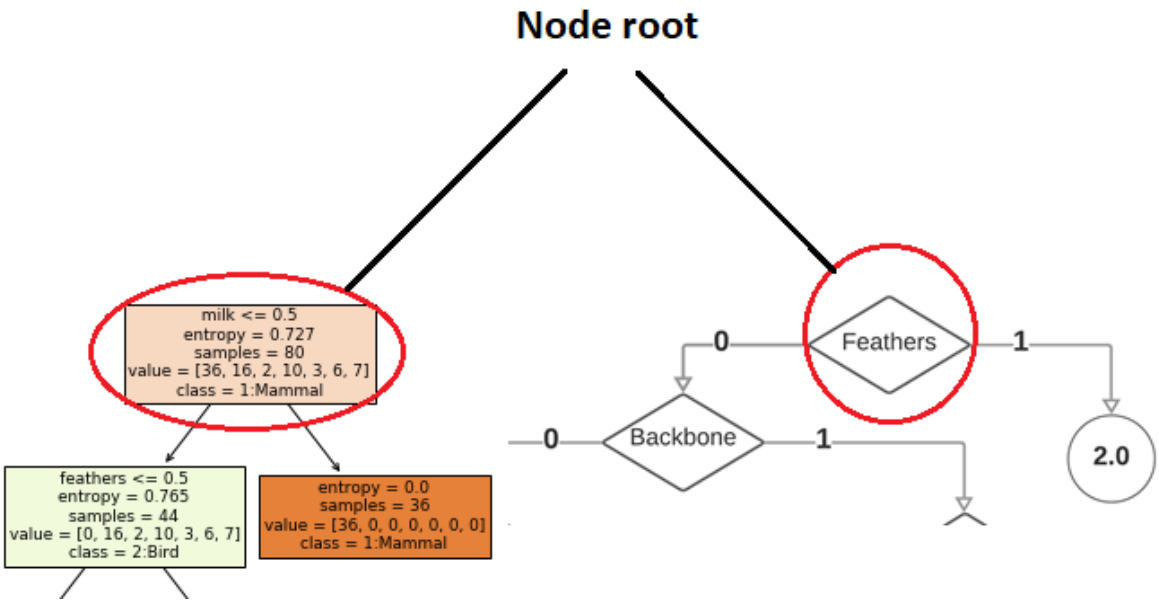
.

Hình 3.8 Cây quyết định sau khi được huấn luyện lần 2 bằng Scikit-Learn

Chương 4. KẾT QUẢ VÀ KẾT LUẬN

4.1. So sánh kết quả

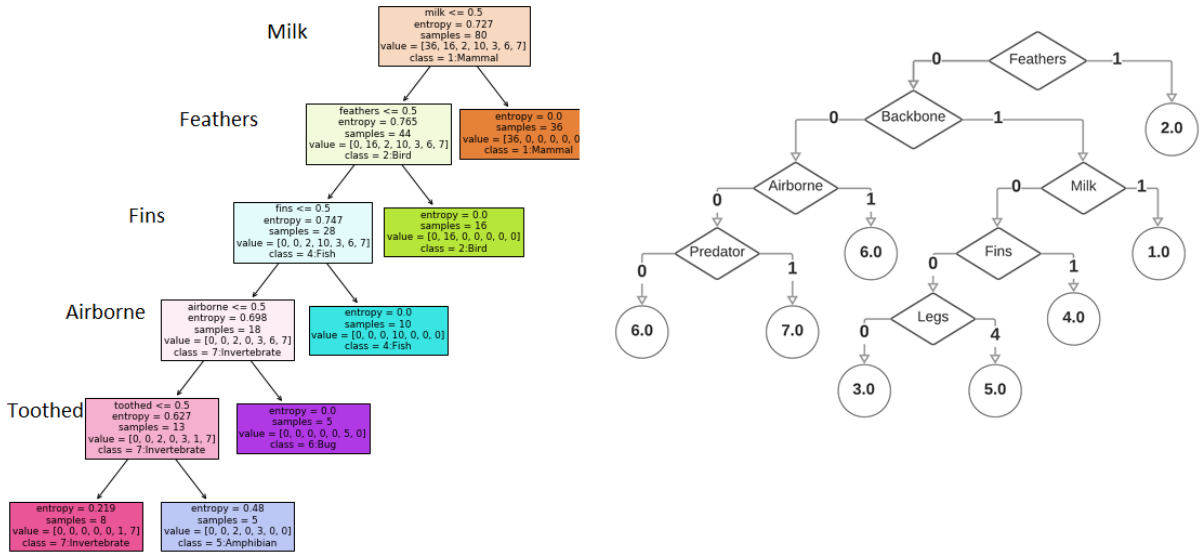
Như vậy, với phương pháp C4.5 đã xây dựng nên một mô hình cây quyết định hoàn chỉnh, từ đó so sánh với cây tương tự được xây dựng bằng Scikit-Learn. Tại đây ta sẽ so sánh giữa cây quyết định sau khi đã cắt tỉa (prunning) với cây của Scikit-Learn.



Hình 4.1 Sự khác biệt về nút gốc giữa 2 cây quyết định

Hình 4.1 cho thấy sự khác biệt khi chọn nút gốc. Ở cây quyết định được tạo bằng thuật toán C4.5 thì nút gốc là thuộc tính Feathers nhưng ở cây quyết định của thư viện Scikit-Learn thì nút được chọn làm gốc là Milk.

Vì nút gốc của cả hai cây khác nhau nên dẫn đến các nút con tiếp theo cũng có sự thay đổi. Ở thư viện Scikit-Learn thì cây quyết định có thứ tự các nút là Milk – Feathers – Fins – Airborne – Toothed nhưng ở thuật toán C4.5 thì thứ tự đã thay đổi như hình 4.2 dưới đây:



Hình 4.2 Sự khác biệt về thứ tự các nút của 2 cây quyết định

Tuy nhiên tỷ lệ chính xác ở cả 2 là như nhau, đều là ~81%. Nhưng khi chạy lại quá trình tạo cây từ lần thứ 2 của thư viện Scikit-Learn thì có sự chênh lệch. Ở lần thứ 2 thì Scikit-Learn tạo ra cây với tỷ lệ dự đoán là ~71% còn ở cây quyết định của phương pháp C4.5 thì vẫn giữ nguyên là 81%.

```
pprint(tree_C45)
print (" ")
print ("Tỷ lệ dự đoán của C4.5:"
test(testing_data,tree_C45)
```

```
Tỷ lệ dự đoán của C4.5:
implement:  80.95238095238095 %
```

Hình 4.3 Tỷ lệ dự đoán chính xác của thuật toán C4.5

4.2. Ưu điểm của phương pháp đề xuất

Thuật toán cây quyết định đã được hình thành và phát triển từ giữa thế kỷ 20, và suốt những năm sau đó, thuật toán này không ngừng được hoàn thiện và trở nên là một trong những thuật toán phổ biến nhất trong lĩnh vực máy học.

Một số ưu điểm của cây quyết định:

- Cây quyết định luôn được xây dựng dựa trên một quy luật nhất định.
- Mô hình cây quyết định có thể biểu diễn một cách rõ ràng cho người dùng mà không cần yêu cầu quá khắt khe về bộ máy hoạt động.
- Cây quyết định có khả năng xử lý được cả 2 kiểu dữ liệu: liên tục và phân loại.

- Cây quyết định cung cấp một dấu hiệu khá rõ ràng trong lĩnh vực phân lớp và dự đoán.

Đối với mô hình cây quyết định tự xây dựng: mô hình này cho thấy khả năng dự đoán đạt được độ chính xác khá cao và ổn định, có thể thấy được cây có thể hoạt động tốt với các phân lớp Mammal, Bird, Fish, những lớp có bộ dữ liệu sau khi lọc thì gần như tinh khiết.

4.3. Kết luận

Từ các kết quả trên, có thể kết luận được rằng, cây có thể hoạt động tốt với các phân lớp Mammal, Bird, Fish, tuy nhiên, với các phân lớp còn lại như Reptile, Amphibian, Bug và Invertebrate cây còn bị nhầm lẫn những loài động vật trong lớp Reptile và Amphibian có những đặc tính khá giống nhau và trong bảng dữ liệu xuất hiện một số dòng dữ liệu có đa phần các đặc tính giống nhau nhưng lại khác nhau về phân lớp và những dòng dữ liệu này có khả năng gây nhiễu về kết quả cuối cùng của cây.

TÀI LIỆU THAM KHẢO

[1] **Sunil Kumar & Himani Sharma, "A Survey on Decision Tree Algorithms of Classification in Data Mining", 2015**

[2] Nguyễn Nghĩa, "Bài toán phân lớp trong Machine Learning". https://eitguide.net/bai-toan-phan-lop-trong-machine-learning-classification-machine-learning/ (truy cập: 10/10/2020)

[3] Nick Z. Zacharis, "Classification and Regression Trees (CART) for Predictive Modeling in Blended Learning", International Journal of Intelligent Systems and Applications(IJISA), Vol.10, No.3, pp.1-9, 2018. DOI: 10.5815/ijisa.2018.03.01

[4] Yan-yan SONG & Ying LU, "Decision tree methods: applications for classification and prediction". https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/ (truy cập: 10/10/2020)

[5] R Delshi Howsalya Devi & Dr. M Indra Devi, "OUTLIER DETECTION ALGORITHM COMBINED WITH DECISION TREE CLASSIFIER FOR EARLY DIAGNOSIS OF BREAST CANCER", *Int J Adv Engg Tech/Vol. VII/Issue II/April-June.,2016/93-98*

[6] NumPy, "NumPy Documentations". https://numpy.org/doc/stable/ (truy cập: 25/10/2020)

[7] Pandas, "Pandas Documentations" https://pandas.pydata.org/docs/ (truy cập: 25/10/2020)

[8] Scikit-Learn, "Decision Trees". https://scikit-learn.org/stable/modules/tree.html (truy cập: 25/10/2020)

[9] Wikipedia, "ID3 algorithm". https://en.wikipedia.org/wiki/ID3_algorithm (truy cập: 28/10/2020)

[10] Wikipedia, "C4.5 algorithm". https://en.wikipedia.org/wiki/C4.5_algorithm (truy cập: 29/10/2020)

[11] Wikipedia, "Decision Tree".' https://en.wikipedia.org/wiki/Decision_tree (truy cập: 29/10/2020)

[12] Mamta Singhal, "Machine Learning: Decision Trees Example in Real Life". https://medium.com/datadriveninvestor/machine-learning-decision-trees-example-in-real-life-b78865015b6f (truy cập: 2/11/2020)

[13] Soham Save, "Personal Loan Prediction: Using Decision Tree". https://www.kaggle.com/sohamsave/personal-loan-prediction-using-decision-tree (truy cập: 2/11/2020)

PHỤ LỤC