

**BỘ GIÁO DỤC VÀ ĐÀO TẠO    BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**

-----\*\*\*-----



**TRẦN TUẤN ĐẠT**

**DỰ ĐOÁN CƠ HỘI VIỆC LÀM CHO SINH VIÊN SAU TỐT  
NGHIỆP**

**ĐỒ ÁN TỐT NGHIỆP**

**HÀ NỘI, NĂM 2025**

BỘ GIÁO DỤC VÀ ĐÀO TẠO    BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**

-----\*\*\*-----



**DỰ ĐOÁN CƠ HỘI VIỆC LÀM CHO SINH VIÊN SAU TỐT  
NGHIỆP**

Ngành:      Công nghệ thông tin

Mã số:

NGƯỜI HƯỚNG DẪN : THS. HOÀNG QUỐC DŨNG

HÀ NỘI, NĂM 2025



Họ tên sinh viên: Trần Tuấn Đạt

Hệ đào tạo: Đại học chính quy

Lớp: 60TH1

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1. TÊN ĐỀ TÀI: Dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp
2. CÁC TÀI LIỆU CƠ BẢN: <https://sites.google.com/site/tlucse404/>
3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung cần thuyết minh	Tỷ lệ %
Chương 1: Tổng quan	5%
Chương 2: Học máy và một số thuật toán trong học máy	40%
Chương 3: Ứng dụng thuật toán xây dựng mô hình	30%
Chương 4: Kết quả và đánh giá mô hình	25%

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Phần	Họ tên giáo viên hướng dẫn
Chương 1: Tổng quan	ThS. Hoàng Quốc Dũng
Chương 2: Học máy và một số thuật toán trong học máy	ThS. Hoàng Quốc Dũng
Chương 3: Ứng dụng thuật toán xây dựng mô hình	ThS. Hoàng Quốc Dũng
Chương 4: Kết quả và đánh giá mô hình	ThS. Hoàng Quốc Dũng

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày 30 tháng 3 năm 2025

**Trưởng Bộ môn**

(Ký và ghi rõ Họ tên)

**Giáo viên hướng dẫn chính**

(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . . tháng. . . . năm 20....

**Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày... tháng...  
năm 20...

**Sinh viên làm Đồ án tốt nghiệp**

Trần Tuấn Đạt



TRƯỜNG ĐẠI HỌC THỦY LỢI  
KHOA CÔNG NGHỆ THÔNG TIN

**BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP**

**Tên đề tài: Dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp**

*Sinh viên thực hiện:* Trần Tuấn Đạt

*Lớp:* 60TH1

*Mã sinh viên:* 1851061821

*Số điện thoại:* 0334565719

*Email:* trantuandat10b12k@gmail.com

*Giáo viên hướng dẫn:* ThS. Hoàng Quốc Dũng

## **TÓM TẮT ĐỀ TÀI**

Hàng năm có hơn 400.000 sinh viên tốt nghiệp tuy nhiên có đến một nửa trong số đó ra trường không có việc làm hoặc làm việc trái với chuyên ngành. Trong bối cảnh ấy, có được việc làm khi ra trường là vấn đề mà bất kỳ sinh viên nào cũng cảm thấy quan tâm. Chính vì vậy, xây dựng mô hình dự đoán cơ hội việc làm cho sinh viên là một giải pháp thiết thực giúp sinh viên có cái nhìn trực quan về việc bản thân có thể nhận được một công việc hay không.

Do vậy, em quyết định thực hiện đồ án tốt nghiệp “Dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp” bằng ngôn ngữ lập trình Python và áp dụng thuật toán ID3 (Iterative Dichotomiser 3) phân loại trong học máy để đưa ra một mô hình dự đoán được việc sinh viên có hay không có việc làm khi ra trường.

## **CÁC MỤC TIÊU CHÍNH**

- Tìm hiểu, phân tích các dữ liệu liên quan đến bài toán.
- Tìm hiểu về thuật toán ID3 và mô hình cây quyết định trong học máy.
- Sử dụng ngôn ngữ lập trình Python và thuật toán ID3 để xây dựng mô hình cây quyết định dự đoán cơ hội việc làm cho sinh viên.
- Phân tích và đánh giá mô hình dự đoán.

## KẾT QUẢ DỰ KIẾN

- Tài liệu phân tích mô hình cây quyết định, lập trình implement dùng thuật toán ID3 để xây dựng mô hình dự đoán việc làm cho sinh viên sau tốt nghiệp.

## **LỜI CAM ĐOAN**

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực từ trong quá trình nghiên cứu, giám sát và tiến hành thực hiện. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN/KLTN

Trần Tuấn Đạt

## LỜI NÓI ĐẦU

Trong thời buổi hiện đại ngày nay, công nghệ thông tin cũng như những ứng dụng của nó không ngừng phát triển, các thuật ngữ như Big Data, Internet of Things, AI, Data mining ... đã trở nên quá quen thuộc. Sự phát triển vượt bậc của công nghệ không chỉ mang đến nhiều cơ hội đổi mới cho doanh nghiệp mà còn mở ra những hướng đi mới trong việc nâng cao chất lượng cuộc sống và cải thiện năng suất lao động. Đặc biệt, việc ứng dụng công nghệ vào giáo dục và định hướng nghề nghiệp đang trở thành một trong những giải pháp hiệu quả để hỗ trợ sinh viên trong việc lựa chọn và chuẩn bị cho con đường sự nghiệp sau khi ra trường.

Tuy nhiên, thực tế cho thấy, bên cạnh một bộ phận sinh viên có định hướng nghề nghiệp rõ ràng và tìm được việc làm phù hợp, vẫn còn nhiều sinh viên năm cuối hoặc vừa tốt nghiệp phải đối mặt với nỗi lo về cơ hội việc làm. Áp lực cạnh tranh, yêu cầu ngày càng cao từ phía nhà tuyển dụng, cũng như sự thiếu hụt trong việc kết nối giữa năng lực cá nhân và nhu cầu thị trường lao động đã và đang là thách thức lớn. Không chỉ bản thân sinh viên mà cả gia đình và doanh nghiệp cũng mong muốn có được một nguồn nhân lực chất lượng, phù hợp và đáp ứng yêu cầu thực tế.

Trong bối cảnh đó, việc ứng dụng các thuật toán học máy vào phân tích và dự đoán khả năng việc làm cho sinh viên năm cuối là một hướng tiếp cận mang tính thực tiễn cao. Đề tài “**Dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp bằng thuật toán ID3**” được thực hiện với mục tiêu xây dựng một mô hình dự đoán dựa trên các yếu tố như kết quả học tập, ngành học và các thông tin liên quan, từ đó phân tích khả năng “có” hoặc “không” có cơ hội việc làm cho từng đối tượng sinh viên. Với việc sử dụng thuật toán **ID3 (Iterative Dichotomiser 3)** – một trong những thuật toán xây dựng cây quyết định phổ biến trong học máy, đề tài không chỉ nhằm cung cấp một công cụ hỗ trợ sinh viên định hướng nghề nghiệp mà còn góp phần thúc đẩy việc ứng dụng công nghệ vào giải quyết các vấn đề xã hội thiết thực.



Đề tài hy vọng sẽ là một đóng góp nhỏ trong việc kết nối giữa tri thức học thuật và nhu cầu thực tiễn, đồng thời mở ra hướng nghiên cứu mới trong việc tận dụng tiềm năng của dữ liệu để hỗ trợ quá trình ra quyết định trong lĩnh vực giáo dục và nhân sự.

Báo cáo gồm 4 chương chính với nội dung như sau:

### **Chương 1: Tổng quan**

- Trình bày khái quát về đề tài và nêu rõ vấn đề cần giải quyết.
- Xác định đối tượng nghiên cứu và đề xuất định hướng phát triển của bài toán.

### **Chương 2: Học máy và các thuật toán liên quan**

- Trình bày cơ sở lý thuyết về học máy, bao gồm phân loại các thuật toán học máy.
- Trình bày chi tiết lý thuyết về thuật toán được áp dụng trong đề tài: Thuật toán cây quyết định ID3 (Iterative Dichotomiser 3).
- Giới thiệu các công cụ và môi trường phát triển được sử dụng, bao gồm:
  - Ngôn ngữ lập trình: Python
  - Công cụ xây dựng giao diện người dùng: Qt Designer
  - Môi trường phát triển: PyCharm
  - Các thư viện hỗ trợ: Scikit-learn, NumPy, Pandas, Matplotlib,...

### **Chương 3: Xây dựng mô hình dự đoán bằng thuật toán ID3**

- Mô tả cụ thể bài toán và quy trình triển khai mô hình dự đoán.
- Trình bày tập dữ liệu được sử dụng và cách thức xử lý dữ liệu đầu vào để phục vụ cho việc huấn luyện mô hình.
- Thiết kế và xây dựng giao diện hiển thị kết quả dự đoán sử dụng Qt Designer.

### **Chương 4: Kết quả thực nghiệm và đánh giá mô hình**

- Trình bày kết quả thu được từ quá trình thực nghiệm.
- Đánh giá hiệu quả của mô hình dựa trên các tiêu chí phù hợp và đề xuất hướng cải tiến.

## LỜI CẢM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc tới Ban Giám hiệu Trường Đại học Thủy Lợi cùng Ban Chủ nhiệm Khoa Công nghệ Thông tin đã luôn quan tâm, tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và rèn luyện tại trường.

Trong suốt bốn năm học tập tại Trường Đại học Thủy Lợi, em đã nhận được rất nhiều sự hỗ trợ, chỉ dẫn tận tình từ các thầy cô giáo và sự giúp đỡ quý báu từ bạn bè. Em xin gửi lời cảm ơn chân thành và sâu sắc nhất tới tập thể giảng viên Khoa Công nghệ Thông tin – Trường Đại học Thủy Lợi, những người đã truyền đạt cho chúng em những kiến thức quý báu, là nền tảng để em hoàn thành đồ án này.

Đặc biệt, em xin trân trọng cảm ơn **ThS. Hoàng Quốc Dũng** – người đã tận tình hướng dẫn, định hướng và đồng hành cùng em trong suốt quá trình thực hiện đề tài "**Dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp**". Những buổi trao đổi và góp ý chuyên môn từ thầy đã giúp em hiểu rõ hơn về lĩnh vực học máy và ứng dụng thực tiễn của nó.

Em cũng xin gửi lời cảm ơn đến lãnh đạo nhà trường cùng các phòng ban chức năng đã trực tiếp và gián tiếp hỗ trợ em trong quá trình học tập và thực hiện đề tài.

Do hạn chế về thời gian và kinh nghiệm thực tiễn, bài báo cáo không tránh khỏi những thiếu sót. Em kính mong nhận được sự chỉ bảo, góp ý quý báu từ các thầy cô để em có thể hoàn thiện hơn về kiến thức cũng như kỹ năng, phục vụ tốt cho công việc sau này.

Em xin chân thành cảm ơn!

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu đề tài.....	2
1.3 Đối tượng nghiên cứu.....	3
1.4 Phạm vi nghiên cứu.....	3
CHƯƠNG 2: HỌC MÁY VÀ MỘT SỐ THUẬT TOÁN TRONG HỌC MÁY .....	4
2.1 Học máy .....	4
2.1.1 Khái niệm liên quan học máy ( Machine Learning) .....	4
2.1.1.1 Machine learning là gì .....	4
2.1.1.2 Machine learning hoạt động như thế nào .....	5
2.1.1.3 Một số phương pháp học máy phổ biến .....	6
2.1.1.4 Ứng dụng học máy vào thực tiễn.....	11
2.1.2 Một số bài toán trong học máy .....	14
2.2 Thuật toán cây quyết định và mở rộng.....	17
2.2.1 Thuật toán cây quyết định (Decision Tree) .....	17
2.2.1.1 Khái niệm .....	17
2.2.1.2 Ưu điểm và nhược điểm của cây quyết định .....	18
2.2.1.3 Một vài thuật ngữ trong cây quyết định .....	19
2.2.1.4 Cơ chế hoạt động của cây quyết định.....	20
2.2.1.5 Ví dụ .....	21
2.2.1.6 Thuật toán mở rộng cây quyết định .....	23
2.2.2 Một số thuật toán mở rộng cây quyết định .....	24
2.2.2.1 Thuật toán ID3.....	24
2.2.2.2 Thuật toán C4.5 .....	34
2.3 Công cụ sử dụng xây dựng bài toán .....	35
2.3.1 Ngôn ngữ lập trình Python .....	35
2.3.1.1 Python là gì ? .....	35
2.3.1.2 Một số ứng dụng của Python.....	36
2.3.1.3 Một số tính năng chính của Python .....	37

2.3.2 Các Python GUI Frameworks tốt nhất .....	38
2.3.2.1 Python GUI Frameworks #1: Kivy.....	38
2.3.2.2 Python GUI Frameworks #2: PyQt .....	39
2.3.2.3 Python GUI Frameworks #3: Tkinter.....	39
2.3.2.4 Python GUI Frameworks #4: WxPython .....	40
2.3.2.5 Python GUI Frameworks #5: PyGUI .....	40
2.3.2.6 Python GUI Frameworks #6: PySide .....	40
2.2.1 Xây dựng giao diện đồ hoạ với Py QT5, Qt Designer .....	41
2.3.3.1 Qt Designer là gì ? .....	42
2.3.3.2 PyQt5 là gì ? .....	42
2.3.3.3 Mục đích và khả năng của Qt .....	46
2.3.3.4 Các ứng dụng được xây dựng bằng Qt.....	46
2.3.4 Trình soạn thảo Pycharm.....	47
2.3.4.1 Các tính năng của PyCharm .....	48
2.3.5 Một số thư viện được sử dụng .....	50
2.3.5.1 Thư viện Scikit-learn .....	50
2.3.5.2 Thư viện NumPy .....	51
2.3.5.3 Thư viện Matplotlib.....	51
2.3.5.4 Thư viện Pandas .....	52
2.4 Các phương pháp đánh giá độ tin cậy của mô hình .....	53
CHƯƠNG 3 ỨNG DỤNG THUẬT TOÁN XÂY DỰNG MÔ HÌNH .....	57
3.1 Mô tả bài toán.....	57
3.1.1 Phân tích chi tiết bài toán.....	57
3.2.1 Môi trường thực hiện .....	58
3.2.2 Dữ liệu đầu vào .....	58
3.2.3 Xây dựng mô hình dự đoán .....	58
3.3 Xây dựng giao diện hiển thị kết quả.....	59
CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH .....	62
4.1 Kết quả đánh giá mô hình .....	62
4.1.1 Kết quả mô hình.....	62
4.1.2 Đánh giá và lựa chọn mô hình .....	65

4.2 Demo giao diện hiển thị .....	66
KẾT LUẬN .....	67
DANH MỤC TÀI LIỆU THAM KHẢO .....	68

## DANH MỤC HÌNH ẢNH

Hình 2.1 Machine learning ( nguồn : internet) .....	4
Hình 2.2 Một quy trình học máy chung( nguồn : internet) .....	5
(Hình 2.3 – Minh họa phân loại học máy, nguồn: Internet) .....	7
(Hình 2.4 – Mô hình học có giám sát, nguồn: V7 Labs) .....	8
(Hình 2.5 – Phân loại học có giám sát, nguồn: V7 Labs) .....	9
(Hình 2.6 – Minh họa học không giám sát, nguồn: Internet) .....	10
Hình 2.7 Minh họa nhận diện khuôn mặt trong ảnh (Ảnh: Internet) .....	12
Hình 2.8 Ví dụ minh họa phân loại khách hàng (Ảnh: Internet) .....	13
Hình 2.9 Ví dụ minh họa trợ lý ảo ( nguồn : internet) .....	14
Hình 2.10 Minh họa bài toán phân loại nhị phân( nguồn : internet) .....	15
Hình 2.11 Minh họa phân loại đa lớp( nguồn : internet) .....	16
Hình 2.12 - Ví dụ minh họa bài toán phân cụm (Nguồn: Internet) .....	16
Hình 2.13 Ví dụ cơ trong thuật toán cây quyết định ( nguồn : internet) .....	22
Hình 2.14 Đồ thị của hàm entropy với $n=2$ ( nguồn : internet) .....	26
Hình 2.15 Bảng giá trị thời tiết( nguồn : internet) .....	29
Hình 2.16 Bảng giá trị theo outlook là sunny( nguồn : internet) .....	30
Hình 2.17 Bảng giá trị theo outlook là overcast( nguồn : internet) .....	31
Hình 2.18 Bảng giá trị theo outlook là rainy( nguồn : internet) .....	31
Hình 2.19 Bảng giá trị theo temperature là hot( nguồn : internet) .....	31
Hình 2.20 Bảng giá trị theo temperature là mild( nguồn : internet) .....	32
Hình 2.21 Bảng giá trị theo temperature là cool( nguồn : internet) .....	32
Hình 2.22 Cây quyết định ID3( nguồn : internet) .....	33
Hình 2.23 Minh họa ngôn ngữ lập trình Python( nguồn : internet) .....	36
Hình 2.24 Python và các ứng dụng trong thực tế( nguồn : internet) .....	37
Hình 2.25 Python GUI Frameworks( nguồn : internet) .....	38
Hình 2.26 – Giao diện đồ họa với Qt Designer (Nguồn: Internet) .....	41
Hình 2.27 – Giao diện của Qt Designer (chạy trên Windows) .....	42
Hình 2.28 Giao diện tương tác trong qt designer .....	45
Hình 2.29 Biểu tượng của PyCharm .....	47

Hình 2.30 Minh họa thư viện Scikit-learn.....	50
Hình 2.31 Minh họa thư viện NumPy .....	51
Hình 2.32 Minh họa thư viện Matplotlib.....	52
Hình 2.33 Minh họa thư viện Pandas (Nguồn: Koodibar) .....	53
Hình 2.34 Độ đo tin cậy Precision và Recall.....	53
Hình 2.35 Minh họa phân bố dữ liệu khi $R^2$ gần phía 1 (bên trái) và $R^2$ gần phía 0 (bên phải) .....	55
Hình 3.2 Thiết kế giao diện với Qt Designer .....	61
Hình 4.1 Demo giao diện hiển thị .....	66

## DANH MỤC BẢNG BIỂU

Lần máy học thứ 1 .....	62
Lần máy học thứ 2 .....	62
Lần máy học thứ 3 .....	63
Lần máy học thứ 4 .....	63
Lần máy học thứ 5 .....	63
Lần máy học thứ 6 .....	64
Lần máy học thứ 7 .....	64
Bảng tổng hợp các độ đo và tỷ lệ dự đoán chính xác qua từng lần huấn luyện .....	65



## **DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ**

**AI** Trí tuệ nhân tạo (Artificial Intelligence)

**DFS** Một tổ chức phi lợi nhuận độc lập (Django Software Foundation)

**ĐATN** Đồ án tốt nghiệp

**HTML** tệp văn bản chưa bố cục trang web (HyperText Markup Language)

**PYTHON** là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML).

**ID3** là một thuật toán do Ross Quinlan phát minh được sử dụng để tạo cây quyết định từ một tập dữ liệu

**C4.5** là một thuật toán được sử dụng để tạo cây quyết định do Ross Quinlan phát triển và là một phần mở rộng của thuật toán ID3.

**ML** Học máy (Machine learning)

**MSE** lỗi bình phương trung bình (Mean Square Error)

**RMSE** Lỗi trung bình bình phương gốc (Root Mean Square Error)

**ID3** Cây Quyết Định (Iterative Dichotomiser 3)

**GUI** Giao diện đồ họa người dùng (Graphical User Interface)

**URL** một loại mã nhận dạng tài nguyên thống nhất (Uniform Resource Locator)

**DATA** dữ liệu

**DATA MINING** là hành động tự động tìm kiếm các kho thông tin lớn để tìm ra các xu hướng và các mẫu vượt ra ngoài các quy trình phân tích đơn giản

**WEB** còn gọi là trang website hoặc trang mạng, và nội dung liên quan được xác định bằng một tên miền chung và được xuất bản trên ít nhất một máy chủ web

## **CHƯƠNG 1: TỔNG QUAN**

### **1.1 Đặt vấn đề**

Sự phát triển kinh tế của một quốc gia không chỉ dựa vào các yếu tố thuận lợi mà còn phải đối mặt với nhiều thách thức, trong đó đáng chú ý là tình trạng thất nghiệp ngày càng gia tăng của sinh viên sau khi ra trường trong cơ chế thị trường hiện nay.

Trong bối cảnh đất nước không ngừng phát triển, bên cạnh việc ứng dụng các công nghệ hiện đại vào sản xuất kinh doanh, yếu tố then chốt quyết định sự phát triển bền vững chính là nguồn nhân lực. Lực lượng lao động trẻ, được đào tạo từ các trường đại học, cao đẳng... là những người năng động, có trình độ, và đóng vai trò quan trọng trong sự phát triển xã hội. Tuy nhiên, tình trạng thất nghiệp của sinh viên sau khi tốt nghiệp đã và đang tác động tiêu cực đến nền kinh tế - xã hội. Vấn đề đặt ra là: Nguyên nhân của tình trạng này là gì? Có phải chương trình đào tạo của các trường đại học chưa phù hợp, hay chính sách sử dụng lao động của Nhà nước còn nhiều bất cập?

Qua nghiên cứu, có thể chỉ ra một số nguyên nhân chính như: hoạt động sản xuất, kinh doanh của nhiều doanh nghiệp gặp khó khăn dẫn đến việc cắt giảm nhu cầu tuyển dụng; các cơ quan nhà nước ngày càng có yêu cầu cao hơn đối với chất lượng tuyển dụng công chức, viên chức; sự gia tăng nhanh chóng số lượng cơ sở đào tạo khiến nguồn cung sinh viên vượt xa nhu cầu của thị trường lao động. Trên thực tế, mỗi năm có rất nhiều sinh viên tốt nghiệp nhưng tỷ lệ đáp ứng được yêu cầu của nhà tuyển dụng vẫn còn thấp.

Trước những thách thức đó, các cơ sở đào tạo và bản thân sinh viên cần có cái nhìn mới, chủ động hơn trong việc chuẩn bị cho cơ hội việc làm sau tốt nghiệp. Nhận thức được vai trò then chốt của việc đào tạo nguồn nhân lực phù hợp với yêu cầu thực tiễn, trong những năm gần đây, các trường đại học tại Việt Nam và trên thế giới đã thực hiện nhiều giải pháp nhằm nâng cao chất lượng đào tạo như: cập nhật chương trình, giáo trình theo hướng hiện đại, đổi mới phương pháp giảng dạy, tăng cường ứng dụng công

nghe thông tin, chú trọng đào tạo kỹ năng mềm, ngoại ngữ, tin học và khuyến khích sinh viên tham gia nghiên cứu khoa học...

Xuất phát từ thực tiễn đó, đề án tốt nghiệp của em lựa chọn đề tài “Dự đoán cơ hội việc làm cho sinh viên năm cuối” bằng cách ứng dụng ngôn ngữ lập trình Python kết hợp với thuật toán cây quyết định ID3 (Iterative Dichotomiser 3) trong học máy nhằm đưa ra mô hình dự đoán khả năng tìm được việc làm của sinh viên năm cuối. Mô hình này có thể được mở rộng để áp dụng cho tất cả sinh viên đang trong quá trình học tập.

## **1.2 Mục tiêu đề tài**

Mục tiêu của đề tài là nghiên cứu và ứng dụng thuật toán cây quyết định ID3 trong học máy để xây dựng mô hình dự đoán cơ hội việc làm cho sinh viên. Các bước thực hiện gồm:

- Tiền xử lý dữ liệu và xây dựng mô hình dự đoán bằng thuật toán ID3.
- Kiểm tra độ chính xác của mô hình bằng phương pháp K-Fold Cross Validation – một phiên bản nâng cao của phương pháp hold-out, chia dữ liệu thành K tập, luân phiên sử dụng một tập để kiểm tra và K-1 tập còn lại để huấn luyện.
- Đánh giá hiệu quả mô hình dựa trên các chỉ số như Precision, Recall và F1-score. Các tham số của mô hình được điều chỉnh qua nhiều lần thử nghiệm để chọn ra cấu hình tối ưu nhất.

Sau khi xác định được mô hình có độ chính xác cao nhất, đề tài tiếp tục triển khai giao diện tương tác để người dùng có thể dễ dàng nhập thông tin và nhận kết quả dự đoán. Như vậy, đề tài tập trung vào hai mục tiêu chính:

- Xây dựng mô hình cây quyết định dựa trên thuật toán ID3 bằng Python để dự đoán khả năng “có” hoặc “không” tìm được việc làm của sinh viên.
- Xây dựng giao diện tương tác để người dùng có thể nhập thông tin đầu vào và nhận kết quả dự đoán.

### **1.3 Đối tượng nghiên cứu**

Đối tượng nghiên cứu là tập dữ liệu được công bố bởi một trường đại học, lấy từ nền tảng Kaggle – nơi cung cấp các bộ dữ liệu phục vụ cho cộng đồng học máy và khoa học dữ liệu. Dữ liệu bao gồm thông tin lý lịch của sinh viên và kết quả có được tuyển dụng hay không.

Từ đó, đề tài “Dự đoán cơ hội việc làm cho sinh viên năm cuối” được thực hiện nhằm ứng dụng thuật toán ID3 để xây dựng mô hình cây quyết định, từ đó dự đoán khả năng sinh viên có được tuyển dụng hay không. Mô hình có thể được sử dụng để dự đoán cơ hội việc làm cho bất kỳ sinh viên nào dựa trên thông tin cá nhân và học tập.

### **1.4 Phạm vi nghiên cứu**

Đề tài được triển khai trong khuôn khổ đồ án tốt nghiệp, tập trung vào các nội dung chính sau:

- Sử dụng ngôn ngữ lập trình Python.
- Tận dụng các thư viện mã nguồn mở có sẵn.
- Nghiên cứu và áp dụng thuật toán cây quyết định ID3 trong học máy.
- Xây dựng mô hình dự đoán khả năng “có” hoặc “không” tìm được việc làm của sinh viên năm cuối.
- Đánh giá mô hình bằng các chỉ số độ chính xác (Accuracy), Precision, Recall, F1-score.
- Lựa chọn mô hình có hiệu suất cao nhất để tích hợp vào giao diện người dùng.
- Thiết kế giao diện tương tác sử dụng công cụ Qt Designer trong Python.
- Hiển thị kết quả dự đoán ngay khi sinh viên nhập thông tin đầu vào.

## CHƯƠNG 2: HỌC MÁY VÀ MỘT SỐ THUẬT TOÁN TRONG HỌC MÁY

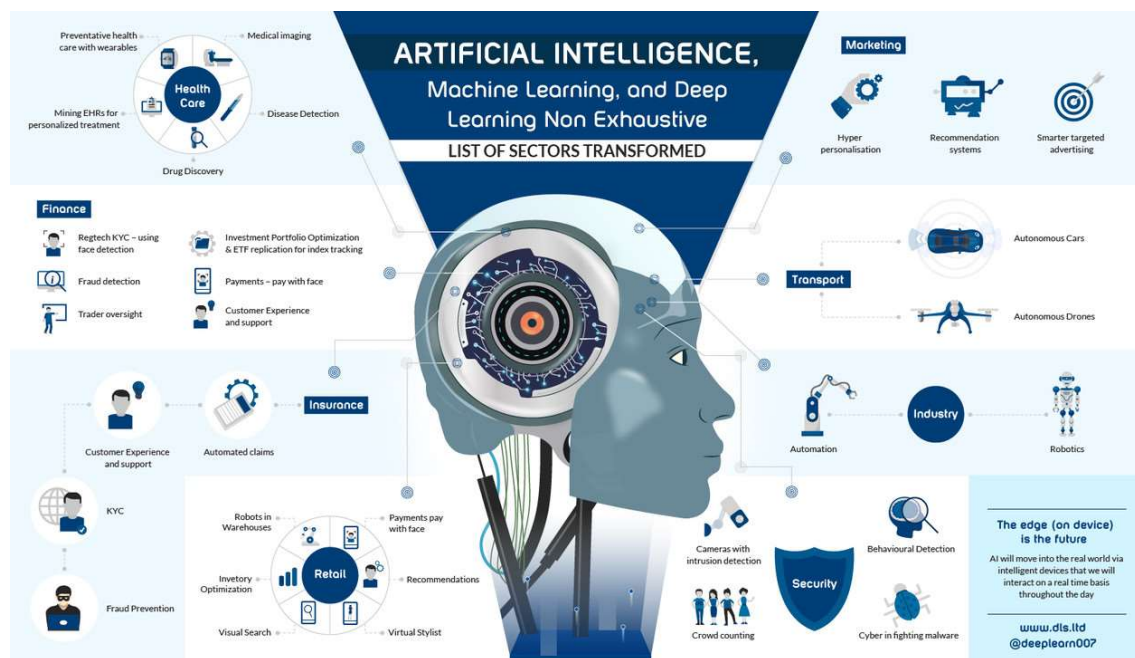
### 2.1 Học máy

#### 2.1.1 Khái niệm liên quan học máy (Machine Learning)

Trong những năm gần đây, cùng với sự bùng nổ của cuộc Cách mạng Công nghiệp 4.0, các thuật ngữ như *trí tuệ nhân tạo (Artificial Intelligence - AI)*, *học máy (Machine Learning)* và *học sâu (Deep Learning)* đã dần trở nên phổ biến, trở thành những khái niệm quen thuộc trong thời đại công nghệ hiện nay.

##### 2.1.1.1 Machine learning là gì

Machine learning là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo (AI)



Hình Error! No text of specified style in document..1 Machine learning ( nguồn : internet)

Học máy là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính (Computer Science) - phương pháp phân tích dữ liệu để tự động hóa việc xây dựng mô hình phân tích, từ đó bắt chước cách con người học, dần dần cải thiện độ chính xác của nó mà không cần sự can thiệp hay trợ giúp của con người.

Học máy là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Học máy sử dụng các phương pháp thống kê và kỹ thuật xử lý dữ liệu để huấn luyện

các mô hình phân tích, giúp máy tính có thể đưa ra dự đoán hoặc phân loại dựa trên dữ liệu mẫu (training data) hoặc từ kinh nghiệm (những gì đã học được).

Ngày nay, học máy đóng vai trò then chốt trong lĩnh vực khoa học dữ liệu (*Data Science*). Khi kết hợp với *Dữ liệu lớn (Big Data)*, các thuật toán học máy giúp cải thiện đáng kể độ chính xác của các mô hình dự đoán, từ đó hỗ trợ phân tích dữ liệu phức tạp một cách nhanh chóng và hiệu quả, ngay cả với quy mô rất lớn.

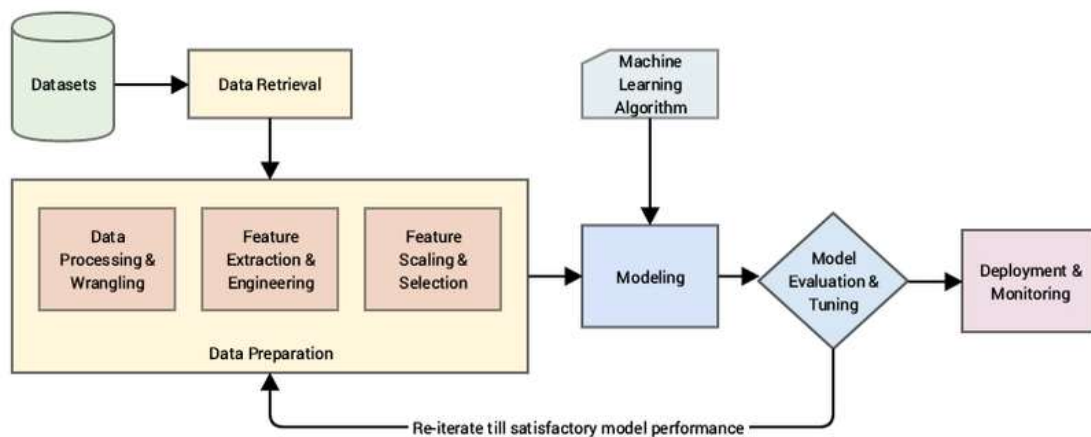
Việc xây dựng các mô hình học máy chính xác không chỉ giúp các tổ chức, doanh nghiệp khai thác dữ liệu hiệu quả, mà còn tạo lợi thế trong việc nhận diện cơ hội sinh lời, tối ưu hóa hoạt động, cũng như hạn chế rủi ro tiềm ẩn.

Hiện nay, hầu hết các ngành công nghiệp hoạt động với khối lượng dữ liệu lớn đều đang ứng dụng học máy như một công cụ quan trọng. Việc khai thác thông tin giá trị từ dữ liệu theo thời gian thực giúp các tổ chức nâng cao hiệu quả hoạt động và tăng cường năng lực cạnh tranh trên thị trường.

#### 2.1.1.2 Machine learning hoạt động như thế nào

- Quy trình hoạt động của Machine learning

Các thuật toán học máy (Machine Learning) được huấn luyện trên một tập dữ liệu đầu vào gọi là *dữ liệu huấn luyện* (training data), nhằm xây dựng một mô hình có khả năng phân tích và dự đoán. Khi tiếp nhận dữ liệu mới, mô hình này sẽ áp dụng những gì đã học được từ dữ liệu quá khứ để đưa ra kết quả dự đoán hoặc phân loại.



Hình Error! No text of specified style in document..2 Một quy trình học máy chung( nguồn : internet)

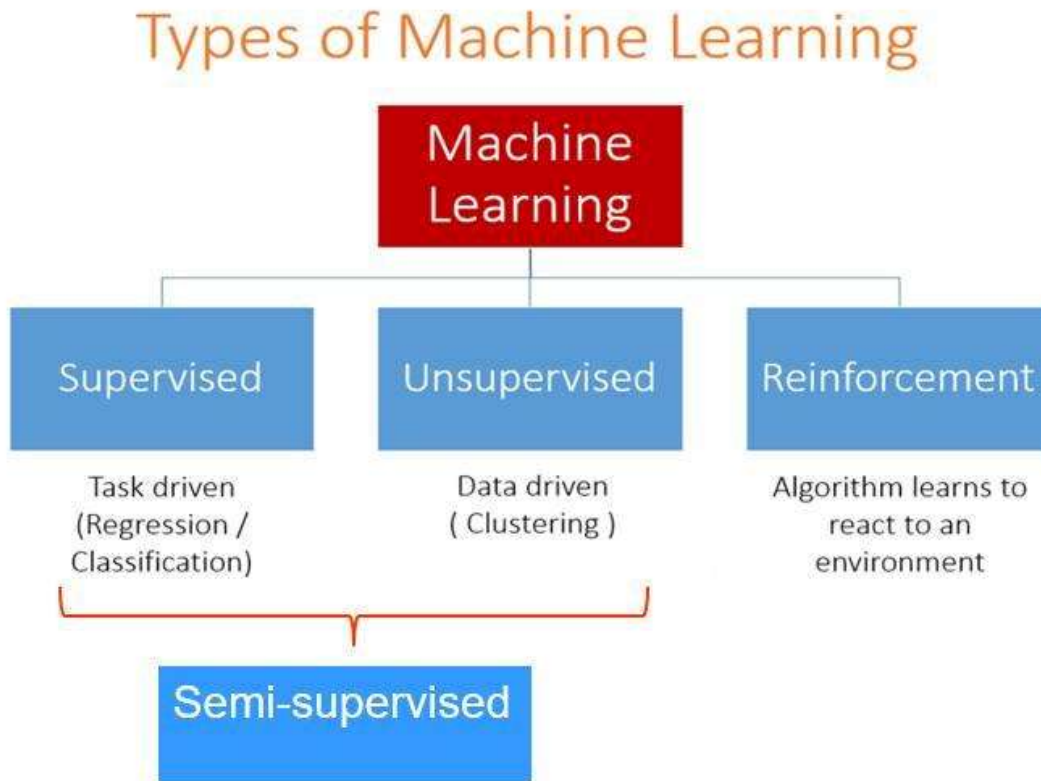
Nhìn chung 1 mô hình học máy sẽ có quy trình chung như sau:

1. *Thu thập dữ liệu*: Dựa trên đặc điểm của bài toán cụ thể, ta tiến hành thu thập dữ liệu (có thể là hình ảnh, văn bản, số liệu, v.v.) từ các nguồn tin cậy. Việc thu thập dữ liệu đầy đủ và chất lượng cao là nền tảng để mô hình đạt được hiệu quả và độ chính xác cao.
2. *Chuẩn bị dữ liệu*: Sau khi thu thập dữ liệu, cần tiến hành các bước xử lý như: làm sạch dữ liệu, loại bỏ các thuộc tính dư thừa, mã hóa dữ liệu dạng chuỗi thành số, gán nhãn, trích xuất đặc trưng, chuẩn hóa / rút gọn dữ liệu để đảm bảo dữ liệu đầu vào phù hợp với yêu cầu của mô hình.
3. *Huấn luyện mô hình*: Tùy vào loại bài toán và đặc điểm của dữ liệu, ta lựa chọn thuật toán học máy phù hợp. Dữ liệu sau xử lý sẽ được chia thành hai phần: tập huấn luyện (training set) và tập kiểm tra (test set). Mô hình sẽ được huấn luyện trên tập huấn luyện, từ đó có khả năng đưa ra dự đoán cho dữ liệu mới.
4. *Đánh giá mô hình*: Sau khi huấn luyện, mô hình được kiểm tra bằng tập dữ liệu kiểm tra nhằm đánh giá hiệu quả. Các chỉ số đánh giá phổ biến gồm: Accuracy (độ chính xác), Precision, Recall, F1-score,... Tùy thuộc vào mục tiêu bài toán, có thể lựa chọn độ đo phù hợp. Một mô hình có độ chính xác trên 80% thường được xem là đạt yêu cầu.
5. *Đào tạo lại mô hình*: Nếu kết quả đánh giá chưa đạt yêu cầu, ta có thể điều chỉnh mô hình: thay đổi thuật toán, điều chỉnh siêu tham số, cải thiện chất lượng dữ liệu, hoặc thu thập thêm dữ liệu. Sau đó, tiến hành huấn luyện lại để cải thiện hiệu suất.
6. *Áp dụng*: Khi mô hình đã đạt kết quả mong muốn, ta triển khai mô hình để áp dụng vào bài toán thực tiễn.

#### 2.1.1.3 Một số phương pháp học máy phổ biến

Trong học máy (Machine Learning), các phương pháp được phân loại dựa trên cách mà mô hình học hỏi từ dữ liệu. Hai nhóm phương pháp phổ biến nhất là **Học có giám sát** (*Supervised Learning*) và **Học không giám sát** (*Unsupervised Learning*). Ngoài ra, còn

có Học bán giám sát (*Semi-supervised Learning*) và Học củng cố tăng cường (*Reinforce Learning*), phù hợp với các đặc thù bài toán và môi trường ứng dụng khác nhau.



(Hình 2.3 – Minh họa phân loại học máy, nguồn: Internet)

- **Học có giám sát (Supervised Learning)**

Học có giám sát là nhóm thuật toán phổ biến nhất trong các thuật toán Machine learning. Trong học có giám sát, mô hình được huấn luyện trên tập dữ liệu đã gán nhãn, nghĩa là mỗi dữ liệu đầu vào đi kèm với nhãn cho đầu ra cụ thể để suy luận ra quan hệ giữa đầu vào và đầu ra. Trải qua quá trình đào tạo từ đó mô hình đưa ra dự đoán cho bộ dữ liệu mới.

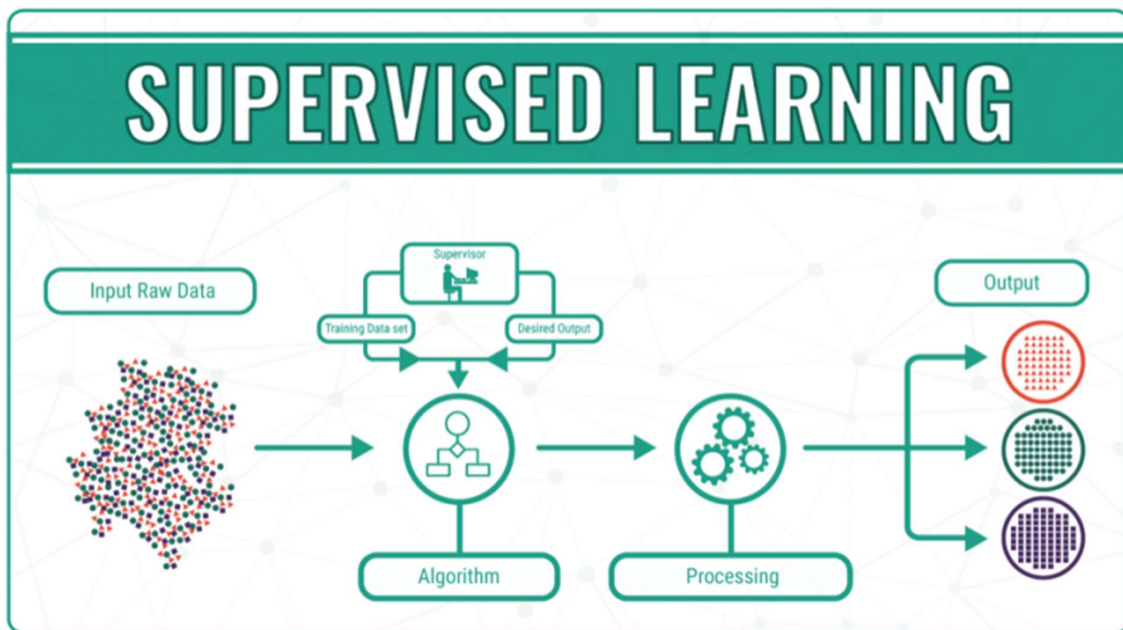
Một số thuật toán tiêu biểu trong Học có giám sát bao gồm:

- Hồi quy logistic (Logistic Regression)
- Máy vector hỗ trợ (SVM)
- Naive Bayes
- Mạng nơ-ron (Neural Networks)
- K-láng giềng gần nhất (KNN)
- Rừng ngẫu nhiên (Random Forest)



Một số bài toán thực tế có tính ứng dụng cao của mô hình học máy có giám sát như:

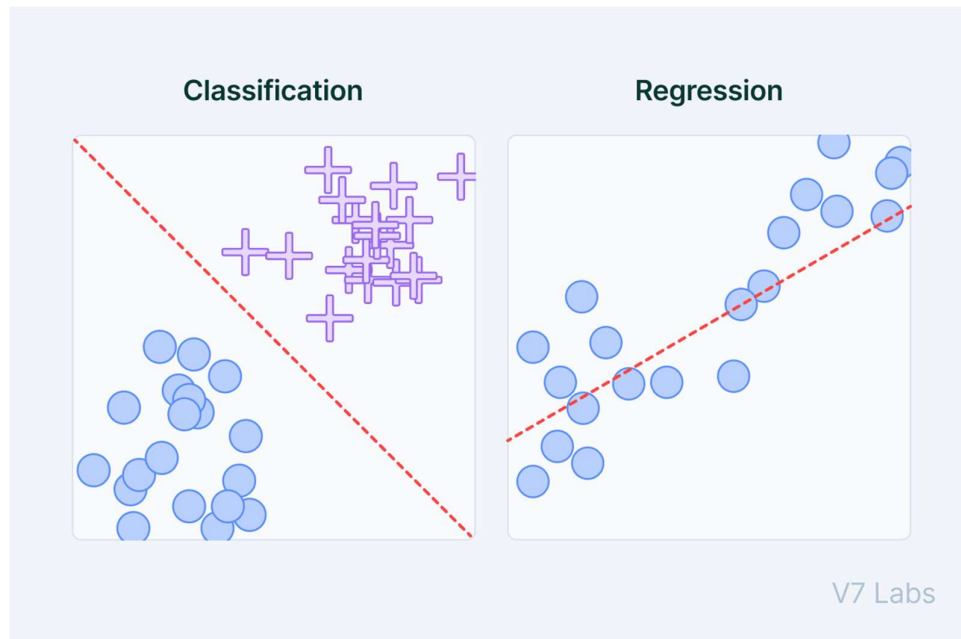
- Dự đoán phân tích giá cả (nhà, cổ phiếu, ...)
- Nhận dạng văn bản và giọng nói
- Phát hiện thư rác
- Nhận diện đối tượng (con người, chữ viết, ...)



(Hình 2.4 – Mô hình học có giám sát, nguồn: V7 Labs)

Học có giám sát con được chia thành hai nhóm nhỏ:

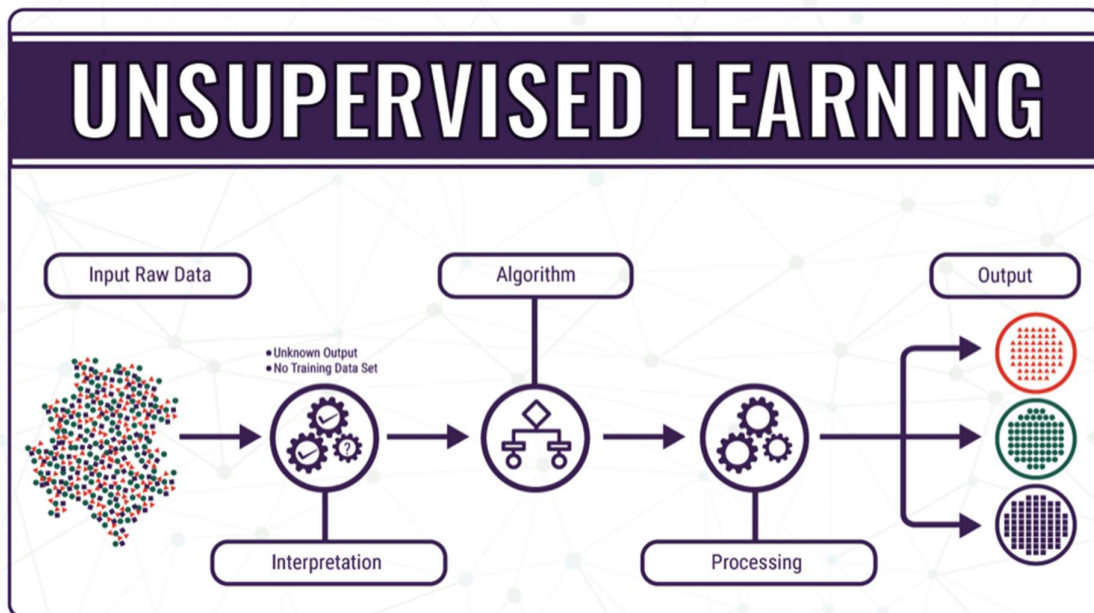
- **Phân loại (Classification):** là một thủ tục trong đó một mô hình hoặc hàm phân tách dữ liệu thành các giá trị rời rạc, tức là nhiều lớp tập dữ liệu sử dụng các tính năng độc lập (VD: phân loại email là "spam" hoặc "bình thường").
- **Hồi quy (Regression):** Dự đoán giá trị liên tục từ đầu vào được cung cấp (VD: dự đoán giá nhà)



(Hình 2.5 – Phân loại học có giám sát, nguồn: V7 Labs)

- **Học không giám sát (Unsupervised Learning)**

Khác với học có giám sát, phương pháp này sử dụng dữ liệu chưa được gán nhãn. Mục tiêu là phát hiện các cấu trúc tiềm ẩn trong dữ liệu như phân nhóm, trích xuất đặc trưng, hay giảm chiều dữ liệu.



(Hình 2.6 – Minh họa học không giám sát, nguồn: Internet)

Học không giám sát được chia làm hai loại chính:

- Phân nhóm (Clustering): Gom các điểm dữ liệu vào các nhóm có tính tương đồng cao
- Luật kết hợp (Association): Tìm ra các quy luật liên hệ giữa các yếu tố dựa trên dữ liệu cho trước (VD: khách mua sữa thường mua bánh mì)

- **Học bán giám sát (Semi-supervised Learning)**

Đây là phương pháp kết hợp giữa học có giám sát và không giám sát, có ứng dụng tương tự như Học có giám sát. Mô hình học từ một phần dữ liệu có nhãn và phần còn lại không có nhãn. Học bán giám sát rất hữu ích khi việc gán nhãn dữ liệu tốn kém thời gian hoặc đòi hỏi chi phí cao.

Học bán giám sát đặt nền tảng trung gian giữa hiệu suất của học có giám sát và hiệu quả của học không giám sát. Một số lĩnh vực sử dụng phương pháp học bán giám sát bao gồm:

- **Dịch máy:** Học dịch ngôn ngữ từ tập song ngữ nhỏ
- **Phát hiện gian lận:** Khi có rất ít dữ liệu gán nhãn gian lận
- **Tự động gán nhãn dữ liệu:** Mô hình học từ tập nhỏ có thể mở rộng gán nhãn cho dữ liệu lớn hơn

---

- **Học tăng cường (Reinforcement Learning)**

Học tăng cường (RL) là một kỹ thuật học máy đào tạo phần mềm đưa ra quyết định nhằm thu về kết quả tối ưu nhất. Kỹ thuật này bắt chước quy trình học thử và sai mà con người sử dụng để đạt được mục tiêu đã đặt ra. Học tăng cường giúp phần mềm tăng cường các hành động hướng tới mục tiêu, đồng thời bỏ qua các hành động làm xáo lãng mục tiêu.

Thuật toán Học tăng cường sử dụng mô hình khen thưởng và trừng phạt trong quy trình xử lý dữ liệu. Các thuật toán này tiếp thu ý kiến phản hồi của từng hành động và tự khám phá ra con đường xử lý tốt nhất để thu về kết quả cuối cùng. Thuật toán RL còn có khả năng trì hoãn khen thưởng. Chiến lược tổng thể tốt nhất có thể đòi hỏi phải đánh đổi một vài lợi ích trước mắt, vì vậy cách tiếp cận tốt nhất mà RL khám phá ra có thể bao gồm một số trừng phạt hoặc giai đoạn quay lui. RL là phương thức hiệu quả giúp hệ thống trí tuệ nhân tạo đạt kết quả tối ưu trong môi trường chưa biết.

- Ưu điểm:
  - ✓ Khả năng tối ưu hành vi lâu dài
  - ✓ Học được trong môi trường động
- Nhược điểm:
  - ✓ Dễ rơi vào trạng thái quá tải (state explosion)
  - ✓ Cần nhiều thời gian và tài nguyên để huấn luyện

#### *2.1.1.4. Ứng dụng học máy vào thực tiễn*

Học máy đang được ứng dụng ngày càng rộng rãi trong nhiều lĩnh vực của đời sống, từ phân tích dữ liệu lớn, dự đoán xu hướng tương lai, đến công nghệ thị giác máy tính, xử lý ngôn ngữ tự nhiên và robot.

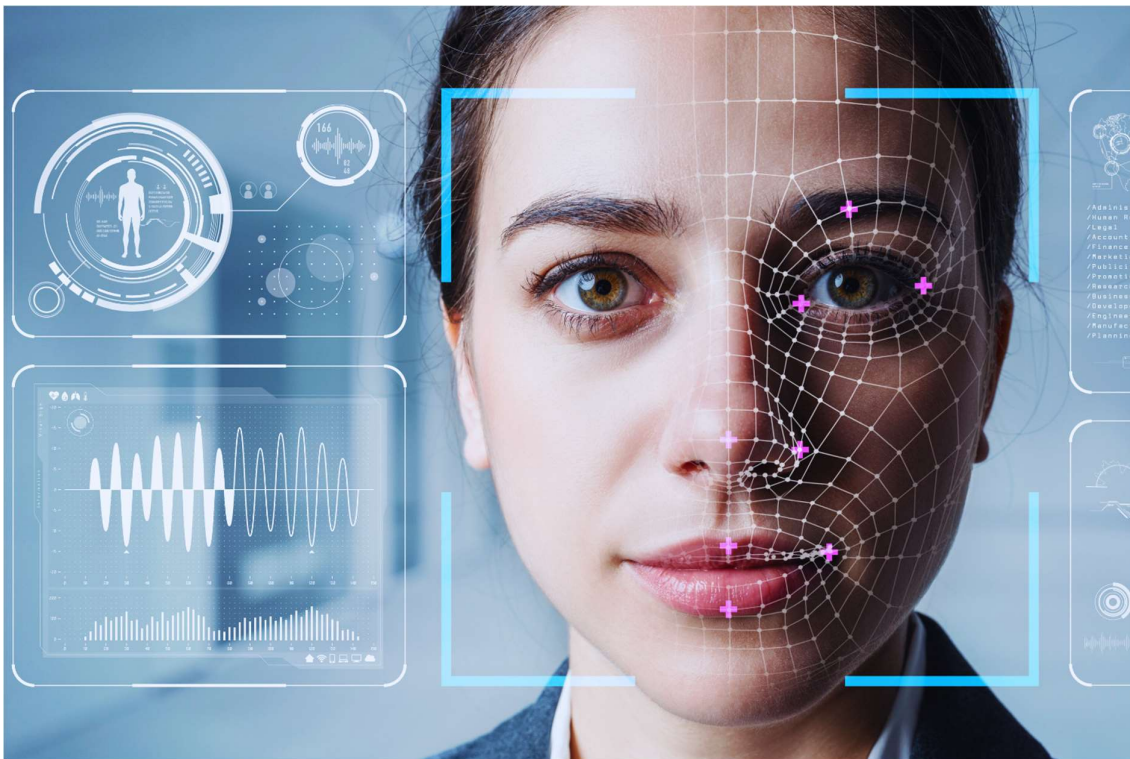
Các lĩnh vực ứng dụng học máy:

- Chatbot
- Tài chính – Ngân hàng
- Y sinh học – Chẩn đoán bệnh
- Nông nghiệp thông minh
- Hóa học – Vật liệu
- Tự động hóa, Robotics
- Tìm kiếm và trích xuất thông tin
- Khoa học vũ trụ

- Mạng máy tính – An ninh mạng
- Thị giác máy tính (Computer Vision)
- Xử lý ngôn ngữ tự nhiên (NLP)

Một số ứng dụng cụ thể:

- Phát hiện và nhận diện hình ảnh: Đây là một trong những ứng dụng phổ biến nhất của học máy và trí tuệ nhân tạo. Kỹ thuật này cho phép hệ thống nhận diện và xác định các đặc trưng của đối tượng trong hình ảnh số. Ngoài ra, nó còn được mở rộng để thực hiện các tác vụ phức tạp hơn như nhận dạng mẫu, nhận diện khuôn mặt, nhận dạng ký tự quang học (OCR), phân tích hình thái học trong ảnh y khoa,...



Hình **Error! No text of specified style in document.**3 Minh họa nhận diện khuôn mặt trong ảnh (Ảnh: Internet)

- Lọc thư rác và phân loại văn bản: Học máy được áp dụng để phân tích nội dung thư điện tử nhằm phân loại thành "thư rác" hoặc "thư hợp lệ". Bên cạnh đó, nó

cũng có thể phân loại tin tức, tài liệu hoặc bài viết theo các chủ đề như xã hội, kinh tế, thể thao, giải trí, v.v.

- Dịch tự động: Các hệ thống dịch sử dụng học máy được huấn luyện trên các cặp văn bản song ngữ, từ đó học cách dịch chính xác từ một ngôn ngữ này sang ngôn ngữ khác, ví dụ như Google Translate, DeepL.
- Chẩn đoán y tế: Học máy hỗ trợ bác sĩ trong việc chẩn đoán bệnh, bằng cách phân tích dữ liệu từ triệu chứng, xét nghiệm, hoặc hình ảnh y tế như X-quang, MRI. Ví dụ: mô hình có thể dự đoán bệnh sâu răng từ hình ảnh X-quang răng của bệnh nhân.
- Phân loại khách hàng và dự đoán sở thích: Dựa trên các yếu tố như độ tuổi, giới tính, sở thích, hành vi tiêu dùng hay màu sắc, kích thước sản phẩm ưa thích,... học máy có thể phân nhóm khách hàng và dự đoán nhu cầu tiêu dùng để tối ưu chiến lược tiếp thị.



Hình *Error! No text of specified style in document.* 4 Ví dụ minh họa phân loại khách hàng  
(Ảnh: Internet)

- Dự đoán chỉ số thị trường: Học máy được ứng dụng để phân tích dữ liệu tài chính, từ đó đưa ra dự đoán về xu hướng biến động của giá cổ phiếu, giá vàng, tỷ giá hối đoái,... dựa trên dữ liệu lịch sử và các chỉ số kinh tế hiện tại.
- Hệ thống khuyến nghị (Recommendation Systems): Các hệ thống này đề xuất sản phẩm, phim, video hoặc tin tức mà người dùng có thể quan tâm, dựa trên lịch sử tương tác và hành vi trước đó. Ứng dụng điển hình như mục "gợi ý cho bạn" trên YouTube, Amazon, Shopee hay Netflix.
- Trợ lý ảo cá nhân (Virtual Personal Assistants): Trợ lý ảo sử dụng học máy để hiểu và phản hồi các yêu cầu của người dùng thông qua văn bản, giọng nói hoặc hình ảnh. Ví dụ: Siri, Google Assistant, Amazon Alexa,...



*Hình Error! No text of specified style in document..5 Ví dụ minh họa trợ lý ảo (nguồn : internet)*

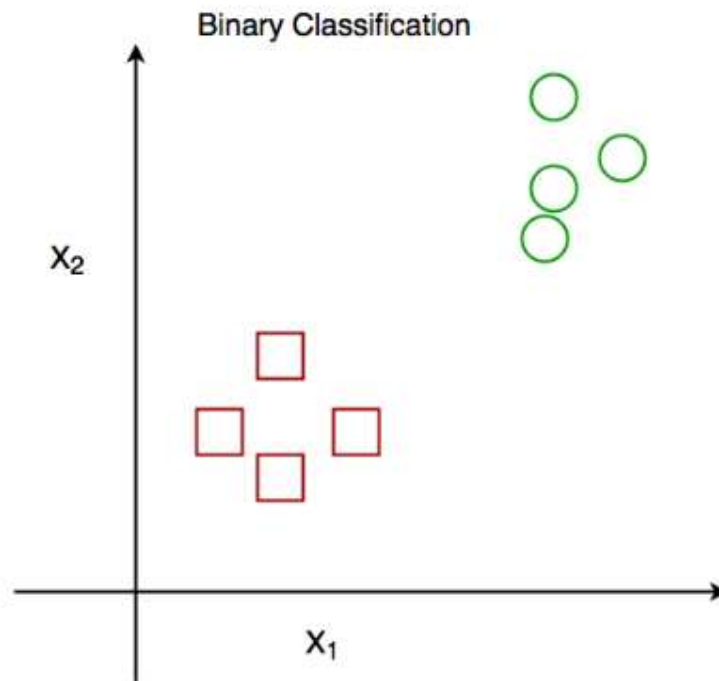
### **2.1.2 Một số bài toán trong học máy**

Học máy (Machine Learning) là một ngành khoa học đang phát triển mạnh mẽ, cho phép máy tính dự đoán các dữ liệu chưa biết dựa trên những dữ liệu đã biết thông qua việc học các mô hình từ dữ liệu. Nhờ khả năng này, học máy được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống. Tuy nhiên, các ứng dụng của học máy thường được quy về một số dạng bài toán phổ biến như sau:



- Phân loại nhị phân (Binary Classification)

Phân loại nhị phân là một trong những bài toán cơ bản và phổ biến nhất trong học máy. Trong bài toán này, mỗi mẫu dữ liệu trong tập huấn luyện được gán nhãn thuộc một trong hai lớp (class), thường ký hiệu là 0 và 1. Mục tiêu của mô hình là học được cách phân biệt hai lớp này, ví dụ như phân loại email là "thư rác" hoặc "không phải thư rác", hoặc phân biệt giữa "có bệnh" và "không có bệnh".

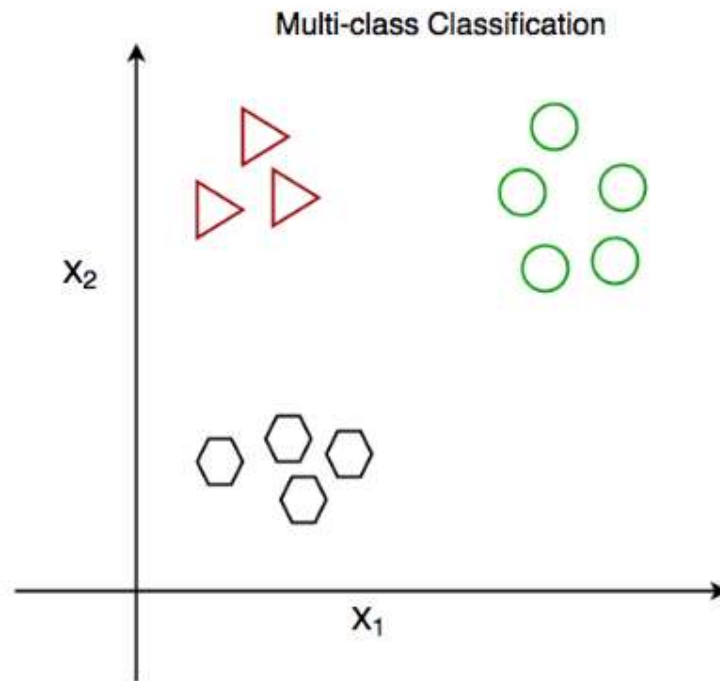


*Hình Error! No text of specified style in document..6 Minh họa bài toán phân loại nhị phân( nguồn : internet)*

- Phân loại đa lớp (Multiclass Classification)

Phân loại đa lớp mở rộng từ phân loại nhị phân, trong đó mỗi mẫu dữ liệu có thể thuộc vào một trong nhiều lớp (nhiều hơn hai). Đây là một dạng bài toán quan trọng, thường gặp trong các ứng dụng như nhận dạng khuôn mặt, nhận dạng chữ viết tay, nhận dạng đối tượng (xe, người, động vật...) trong ảnh. So với phân loại nhị phân, bài toán đa lớp thường phức tạp hơn và yêu cầu mô hình có độ chính xác cao hơn để xử lý nhiều khả năng phân loại.

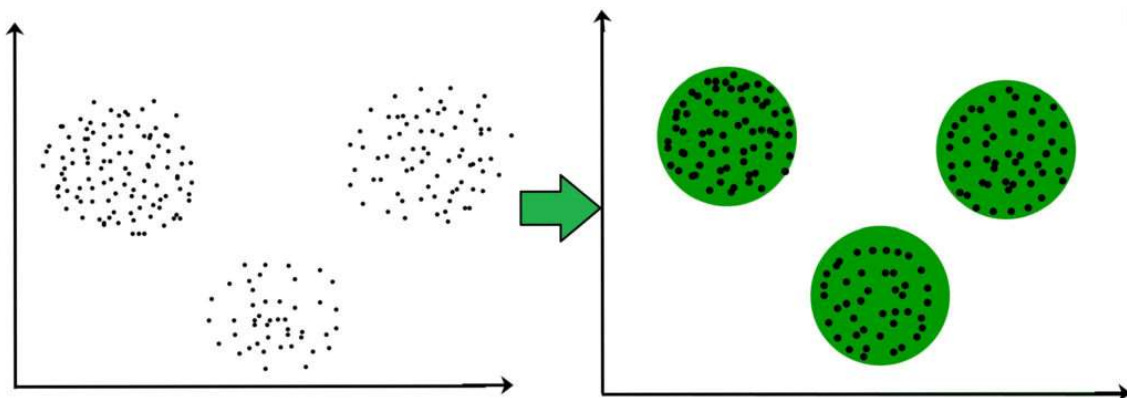




Hình *Error! No text of specified style in document.* Minh họa phân loại đa lớp( nguồn : internet)

- Phân cụm (Clustering)

Phân cụm là một bài toán học không giám sát (unsupervised learning), trong đó mục tiêu là chia tập dữ liệu thành các nhóm (cụm) sao cho các điểm dữ liệu trong cùng một nhóm có đặc điểm tương đồng với nhau và khác biệt so với các điểm dữ liệu ở nhóm khác. Phân cụm thường được sử dụng trong khai phá dữ liệu, phân tích khách hàng, phát hiện bất thường, và nhiều lĩnh vực khác.



Hình 2.12 - Ví dụ minh họa bài toán phân cụm (Nguồn: Internet)

- Hồi quy (Regression)

Bài toán hồi quy nhằm dự đoán giá trị liên tục dựa trên dữ liệu đầu vào. Từ tập dữ liệu quan sát được, mô hình hồi quy tìm ra một hàm toán học mô tả mối quan hệ giữa các biến đầu vào và đầu ra. Hồi quy được sử dụng trong nhiều tình huống thực tế như dự đoán giá nhà, giá cổ phiếu, dự báo thời tiết, v.v.

Đặc điểm quan trọng của bài toán hồi quy là yêu cầu mô hình phải tổng quát hóa tốt từ dữ liệu để đưa ra dự đoán chính xác, đồng thời cần xử lý tốt dữ liệu nhiễu hoặc sai lệch. Độ chính xác của mô hình thường được đánh giá bằng sai số dự đoán trung bình (mean prediction error).

Một số ứng dụng điển hình của hồi quy:

- Dự đoán giá sản phẩm
- Dự đoán biến động thị trường chứng khoán
- Dự báo thời tiết
- Dự đoán lượng tiêu thụ sản phẩm theo thời gian

Hồi quy là một trong những công cụ chính được các nhà khoa học sử dụng để mô hình hóa và xây dựng các quy tắc tổng quát.

## **2.2 Thuật toán cây quyết định và mở rộng**

### **2.2.1 Thuật toán cây quyết định (Decision Tree)**

#### *2.2.1.1 Khái niệm*

Cây quyết định (Decision Tree) là một thuật toán học có giám sát, không tham số, được sử dụng cho cả bài toán phân loại và hồi quy. Mô hình này có cấu trúc dạng cây phân cấp, bao gồm nút gốc (root node), các nhánh (branches), nút bên trong (internal/decision nodes) và nút lá (leaf nodes).

Thuật toán bắt đầu từ nút gốc – nơi không có bất kỳ nhánh đi vào. Từ nút gốc, cây mở rộng ra các nút quyết định, nơi dữ liệu được phân tách dựa trên các đặc trưng đầu vào. Quá trình phân chia tiếp tục cho đến khi các nút lá được hình thành. Mỗi nút lá đại diện cho một kết quả đầu ra hoặc một dự đoán cụ thể của mô hình.

Cây quyết định xây dựng mô hình bằng cách lần lượt đặt ra các câu hỏi về dữ liệu, nhằm chia tập dữ liệu thành các nhóm con ngày càng đồng nhất. Chính vì vậy, người ta cho rằng cây quyết định mô phỏng quá trình ra quyết định của con người. Trong quá trình xây dựng cây, nó chia toàn bộ dữ liệu thành các tập dữ liệu con cho đến khi đưa ra quyết định.

#### 2.2.1.2 Ưu điểm và nhược điểm của cây quyết định

- Ưu điểm

So với nhiều phương pháp khai phá dữ liệu khác, cây quyết định sở hữu một số ưu điểm nổi bật như sau:

- Dễ hiểu và dễ giải thích: Cấu trúc dạng cây giúp người dùng dễ dàng theo dõi và giải thích các quyết định mà mô hình đưa ra chỉ sau một phần giải thích ngắn gọn.
- Yêu cầu tối thiểu về xử lý dữ liệu đầu vào: Cây quyết định không đòi hỏi quá trình chuẩn bị dữ liệu phức tạp. Không cần chuẩn hóa dữ liệu, tạo biến giả (dummy variables), hay loại bỏ giá trị thiếu như nhiều mô hình khác.
- Xử lý linh hoạt cả dữ liệu định lượng và định danh: Cây quyết định có khả năng hoạt động hiệu quả với cả dữ liệu số và dữ liệu phân loại. Trong khi đó, nhiều phương pháp khác chỉ phù hợp với một kiểu dữ liệu cụ thể (ví dụ: mạng nơ-ron chủ yếu với dữ liệu số, luật kết hợp với dữ liệu định danh).
- Mô hình "hộp trắng" (white-box): Người dùng có thể hiểu và giải thích quyết định của mô hình bằng các điều kiện logic (logic Boolean). Điều này trái ngược với các mô hình "hộp đen" như mạng nơ-ron, vốn rất khó diễn giải kết quả đầu ra.
- Có thể đánh giá bằng kiểm định thống kê: Việc kiểm tra độ tin cậy và chất lượng mô hình có thể thực hiện thông qua các phương pháp thống kê, giúp củng cố niềm tin vào kết quả dự báo.
- Xử lý hiệu quả với dữ liệu lớn: Cây quyết định có thể được huấn luyện nhanh chóng trên tập dữ liệu lớn mà không cần đến các hệ thống tính toán phức tạp, giúp hỗ trợ ra quyết định kịp thời trong thực tế.

- Nhược điểm
  - Không phù hợp với dữ liệu tuần tự hoặc phụ thuộc theo thời gian: Cây quyết định không thể hiện hiệu quả cao đối với các bài toán liên quan đến dữ liệu chuỗi thời gian hoặc khi tính liên tục là yếu tố quan trọng.
  - Dễ bị thay đổi cấu trúc bởi dữ liệu đầu vào: Mô hình rất nhạy cảm với thay đổi nhỏ trong dữ liệu huấn luyện. Một thay đổi nhỏ có thể dẫn đến việc xây dựng một cấu trúc cây hoàn toàn khác, làm giảm độ ổn định của mô hình.
  - Rủi ro quá khớp (overfitting): Nếu không được cắt tỉa (pruned) hoặc kiểm soát độ sâu hợp lý, sẽ dẫn đến việc mô hình học rất tốt trên dữ liệu huấn luyện nhưng lại không hoạt động tốt trên dữ liệu mới.

### 2.2.1.3 Một vài thuật ngữ trong cây quyết định

#### **Nút gốc (Root Node):**

Nút gốc là điểm bắt đầu của cây quyết định. Nó đại diện cho toàn bộ tập dữ liệu và được phân chia thành hai hoặc nhiều tập con dựa trên các đặc trưng của dữ liệu.

#### **Nút lá (Leaf Node):**

Nút lá là nút kết thúc của cây, nơi không còn sự phân tách nào nữa. Mỗi nút lá tương ứng với một kết quả hoặc nhãn đầu ra cuối cùng trong bài toán phân loại hoặc hồi quy.

#### **Tách (Splitting):**

Là quá trình chia nhỏ một nút (nút gốc hoặc nút quyết định) thành các nút con, dựa trên điều kiện phân tách xác định theo đặc trưng của dữ liệu.

#### **Cành / Cây con (Branch / Sub-tree):**

Mỗi nhánh của cây đại diện cho một kết quả của quá trình phân tách tại một nút. Tập hợp các nhánh từ một nút tạo thành một cây con (sub-tree).

#### **Tỉa cành (Pruning):**

Tỉa cành là kỹ thuật được sử dụng để loại bỏ các nhánh không cần thiết hoặc dư thừa khỏi cây nhằm giảm thiểu hiện tượng quá khớp (overfitting) và cải thiện khả năng tổng quát của mô hình.

### **Nút cha / Nút con (Parent / Child Node):**

Trong cấu trúc cây, bất kỳ nút nào được phân tách ra đều là nút cha, còn các nút được tạo ra từ nó được gọi là các nút con.

#### *2.2.1.4 Cơ chế hoạt động của cây quyết định*

- Cây quyết định hoạt động dựa trên chiến lược “chia để trị” (divide and conquer), thực hiện tìm kiếm tham lam (greedy search) để xác định các điểm phân tách tối ưu trong một cây. Quá trình này diễn ra theo hướng đệ quy từ trên xuống, tiếp tục phân chia dữ liệu cho đến khi tất cả hoặc phần lớn các bản ghi được phân loại vào các nhãn lớp cụ thể.
- Tất cả các điểm dữ liệu của các tập con sau phân chia có được phân loại thành các tập đồng nhất hay không phụ thuộc đáng kể vào độ phức tạp của cây. Các cây có kích thước nhỏ thường dễ đạt được các nút lá thuần túy – tức là tất cả các điểm dữ liệu trong một nút lá thuộc về cùng một lớp. Tuy nhiên, khi cây phát triển quá lớn, việc duy trì độ thuần nhất trở nên khó khăn hơn và dẫn đến tình trạng một số cây con chứa quá ít dữ liệu.
- Hiện tượng này được gọi là phân mảnh dữ liệu (data fragmentation) và thường dẫn đến quá khớp (overfitting). Do đó, cây quyết định thường ưu tiên các cây đơn giản hơn – điều này phù hợp với nguyên lý của parsimony trong Occam’s Razor: “Các thực thể không nên được nhân lên quá mức cần thiết.” Nói cách khác, mô hình càng đơn giản càng tốt, miễn là vẫn đảm bảo được độ chính xác.
- Để kiểm soát độ phức tạp và hạn chế quá khớp, kỹ thuật tỉa cành (pruning) được áp dụng. Đây là quá trình loại bỏ các nhánh không cần thiết, thường là những phân chia không mang lại lợi ích đáng kể cho độ chính xác của mô hình. Sau khi tỉa, mô hình có thể được đánh giá lại thông qua quá trình xác nhận chéo (cross-validation) nhằm đảm bảo hiệu quả tổng quát.
- Ngoài ra, để nâng cao độ chính xác và giảm sai số tổng thể, cây quyết định còn có thể được sử dụng như một thành phần trong các mô hình tổ hợp, tiêu biểu là rừng ngẫu nhiên (Random Forest). Phương pháp này xây dựng nhiều cây quyết định độc lập và kết hợp kết quả dự đoán từ các cây để cho ra dự báo cuối cùng,

giúp cải thiện hiệu suất đặc biệt khi các cây riêng lẻ không có mối tương quan cao.

- Quy trình dự đoán trong cây quyết định diễn ra như sau: bắt đầu từ nút gốc, thuật toán so sánh giá trị thuộc tính của dữ liệu đầu vào với điều kiện tại nút hiện tại. Dựa trên kết quả so sánh, thuật toán di chuyển theo nhánh tương ứng đến nút tiếp theo và tiếp tục quá trình này cho đến khi đi đến nút lá, nơi đưa ra quyết định phân loại cuối cùng. Quy trình hoàn chỉnh có thể được hiểu rõ hơn bằng thuật toán sau:
  - Bước 1: Bắt đầu với nút gốc (S), chứa toàn bộ tập dữ liệu huấn luyện.
  - Bước 2: Xác định thuộc tính tốt nhất để phân chia dữ liệu bằng cách sử dụng Phép đo lựa chọn thuộc tính (Attribute Selection Measure - ASM).
  - Bước 3: Dựa trên thuộc tính đã chọn, chia S thành các tập con tương ứng với các giá trị có thể của thuộc tính đó.
  - Bước 4: Tạo nút cây quyết định mới chứa thuộc tính tốt nhất.
  - Bước 5: Tạo một cách đệ quy cây quyết định mới bằng cách sử dụng các tập con của tập dữ liệu đã tạo ở bước 3. Quá trình dừng lại khi không còn khả năng phân chia thêm, hoặc khi một tiêu chí dừng (như số mẫu tối thiểu, độ sâu cây) được thỏa mãn. Khi đó, nút tương ứng được gán là nút lá và chứa kết quả dự đoán.

#### 2.2.1.5 Ví dụ

Một trong những ví dụ kinh điển để minh họa cho cơ chế hoạt động của cây quyết định là bài toán: “Có nên đi chơi bóng hay không?” dựa trên các đặc điểm thời tiết.

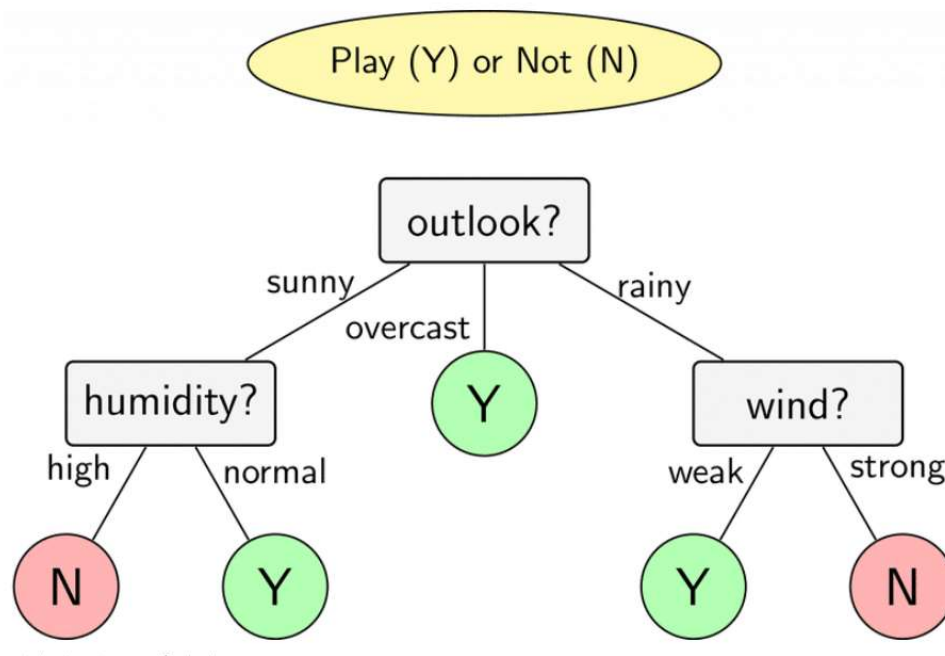
Các thuộc tính đầu vào:

- Thời tiết (Outlook): Nắng (Sunny), Âm u (Overcast), Mưa (Rain)
- Độ ẩm (Humidity): Cao (High), Bình thường (Normal)
- Gió (Wind): Mạnh (Strong), Nhẹ (Weak)

Dựa vào tập dữ liệu huấn luyện và quá trình chọn thuộc tính tốt nhất ở từng bước, thuật toán sẽ xây dựng cây quyết định để phân loại hành động “Chơi bóng” hay “Không chơi bóng”.

Cây quyết định có thể được xây dựng như sau:

- Nếu thời tiết là "Âm u" → Chơi bóng
- Nếu thời tiết là "Nắng":
  - Nếu độ ẩm là "Cao" → Không chơi bóng
  - Nếu độ ẩm là "Bình thường" → Chơi bóng
- Nếu thời tiết là "Mưa":
  - Nếu gió là "Mạnh" → Không chơi bóng
  - Nếu gió là "Nhẹ" → Chơi bóng



Hình **Error! No text of specified style in document..8** Ví dụ cơ trong thuật toán cây quyết định ( nguồn : internet)

Theo mô hình trên, ta thấy:

Khi trời nắng, yếu tố độ ẩm trở thành tiêu chí quyết định: độ ẩm bình thường thì đi chơi, còn độ ẩm cao thì không.

Khi trời mưa, quyết định phụ thuộc vào gió: gió nhẹ thì đi chơi, còn gió mạnh thì không.

Trời âm u luôn tạo điều kiện thuận lợi → chơi bóng trong mọi trường hợp.

#### 2.2.1.6 Thuật toán mở rộng cây quyết định

Thuật toán của Hunt, được phát triển vào những năm 1960, ban đầu nhằm mô hình hóa quá trình học tập của con người trong lĩnh vực Tâm lý học. Thuật toán này đặt nền móng cho nhiều phương pháp xây dựng cây quyết định hiện đại và vẫn giữ vai trò quan trọng trong lĩnh vực học máy ngày nay. Từ cơ sở này, nhiều thuật toán cây quyết định đã được phát triển, tiêu biểu như:

- **ID3** (Iterative Dichotomiser 3)
  - Được phát triển bởi Ross Quinlan, ID3 là một trong những thuật toán xây dựng cây quyết định đầu tiên và phổ biến nhất.
  - Thuật toán sử dụng Entropy và Information Gain để lựa chọn thuộc tính phân tách tại mỗi nút. Mỗi bước đều chọn thuộc tính làm giảm độ bất định (entropy) của tập dữ liệu nhiều nhất.
- **C4.5**
  - Là sự kế thừa và mở rộng từ ID3, cũng do Quinlan phát triển.
  - C4.5 cải thiện ID3 bằng cách hỗ trợ:
    - Dữ liệu thiếu
    - Các giá trị liên tục (qua việc tìm điểm phân cắt tối ưu)
    - Cắt tỉa cây tự động (automatic pruning)
  - C4.5 sử dụng Information Gain Ratio thay vì chỉ Information Gain, để tránh ưu tiên thuộc tính có nhiều giá trị.
- Ngoài ra còn một số thuật toán cây quyết định khác như:



- CHAID(Chi-squared Automatic Interaction Detector):  
Dựa trên kiểm định thống kê Chi-square để xác định điểm phân tách tối ưu. Có thể xử lý biến đầu vào phân loại hoặc liên tục. Thường được sử dụng trong nghiên cứu thị trường và khoa học xã hội.
- C&R Tree (Classification and Regression Tree - CART):  
Áp dụng phân vùng đệ quy (recursive partitioning) để xây dựng cây. Hỗ trợ cả bài toán phân loại và hồi quy. CART thường sử dụng Gini Index cho phân loại và Mean Squared Error cho hồi quy.
- MARS (Multivariate Adaptive Regression Splines):  
Không hoàn toàn là cây quyết định truyền thống nhưng sử dụng phương pháp phân vùng và hồi quy spline để mô hình hóa mối quan hệ phi tuyến giữa biến đầu vào và đầu ra. Phù hợp cho các bài toán hồi quy phi tuyến.
- Conditional Inference Trees:  
Xây dựng cây dựa trên các kiểm định thống kê có điều kiện. Loại bỏ các thiên lệch trong việc lựa chọn thuộc tính bằng cách sử dụng kiểm định giả thuyết thay vì các chỉ số như Gain hay Gini. Phù hợp với mô hình có biến đầu vào hỗn hợp và cấu trúc dữ liệu phức tạp.

## **2.2.2 Một số thuật toán mở rộng cây quyết định**

### **2.2.2.1 Thuật toán ID3**

ID3 (Iterative Dichotomiser 3) là một trong những thuật toán cây quyết định đầu tiên, được phát triển bởi Ross Quinlan, chuyên dùng cho bài toán phân loại (classification), trong đó tất cả các thuộc tính đầu vào đều là thuộc tính rời rạc (categorical).

#### **a) Ý tưởng**

- Trong quá trình xây dựng cây quyết định bằng thuật toán ID3, một bước quan trọng là xác định thứ tự lựa chọn các thuộc tính tại mỗi nút của cây. Trong các bài toán có nhiều thuộc tính và mỗi thuộc tính lại có nhiều giá trị, việc tìm kiếm nghiệm tối ưu toàn cục cho toàn bộ cây là điều không khả thi về mặt tính toán. Do đó, ID3 sử dụng một chiến lược tham lam (greedy strategy) để đơn giản hóa vấn đề.

- Cụ thể, tại mỗi bước, thuật toán lựa chọn một thuộc tính tốt nhất để phân chia dữ liệu, dựa trên một tiêu chí đánh giá cụ thể. Sau khi chọn được thuộc tính, dữ liệu sẽ được chia thành các nút con (child nodes) tương ứng với các giá trị khác nhau của thuộc tính đó. Quá trình này được lặp lại đệ quy trên từng nút con cho đến khi đạt điều kiện dừng (chẳng hạn như nút đã thuần nhất hoặc không còn thuộc tính để phân chia).
- Chiến lược tham lam này không đảm bảo tìm được cây tốt nhất toàn cục, tuy nhiên về mặt trực giác, việc lựa chọn thuộc tính tốt nhất tại từng bước vẫn cho ra một cây có hiệu suất tương đối cao. Hơn nữa, cách tiếp cận này giúp giảm đáng kể độ phức tạp tính toán, giúp việc xây dựng cây khả thi và hiệu quả trên thực tế.
- Tại mỗi nút, để chọn thuộc tính tối ưu, cần có một hàm đo chất lượng của phép phân chia. Trong ngữ cảnh này, có thể hiểu một phép phân chia là tốt nếu dữ liệu tại mỗi nút con sau phân chia thuần nhất – tức là tất cả các mẫu dữ liệu trong cùng một nút con thuộc về cùng một lớp (class). Khi điều này xảy ra, nút con có thể được xem là nút lá (leaf node), không cần phân chia thêm nữa.
- Ngược lại, nếu các nút con vẫn chứa dữ liệu pha trộn giữa nhiều lớp, phép phân chia đó được xem là kém chất lượng. Do đó, cần một hàm số có thể đo mức độ "tinh khiết" (purity) hoặc "vẩn đục" (impurity) của dữ liệu sau phân chia. Hàm số lý tưởng phải cho giá trị thấp nhất khi dữ liệu thuộc một lớp duy nhất, và cao nhất khi dữ liệu được chia đều giữa các lớp.
- Một trong những hàm được sử dụng phổ biến nhất trong lý thuyết thông tin để đo mức độ hỗn tạp là hàm entropy.

## Hàm 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 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  $x_1, x_2, \dots, x_n$ .

Giả sử rằng xác suất để  $x$  nhận các giá trị này là  $p_i = p(x=x_i)$ .

Ký hiệu phân phối này là  $p = (p_1, p_2, \dots, p_n)$ . Entropy của phân phối này được định nghĩa là:

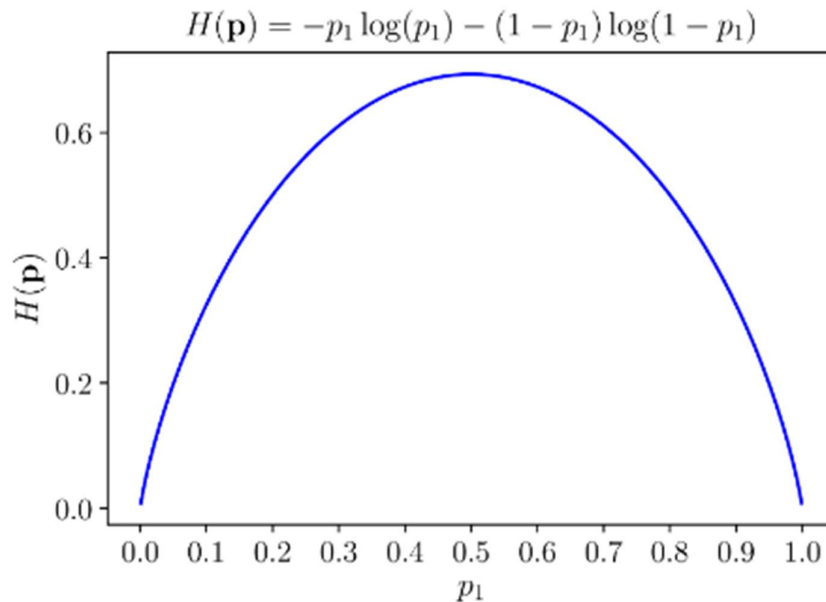
$$H(p) = - \sum_{i=1}^n p_i \log(p_i)$$

trong đó  $\log$  là logarit tự nhiên (Một số tài liệu dùng logarit cơ số 2, nhưng giá trị của  $H(p)$  chỉ khác đi bằng cách nhân với một hằng số.) và quy ước  $0 \log(0) = 0$

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

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

. Xét một ví dụ với  $n=2$  được cho trên Hình 2.14. Trong trường hợp  $p$  là *tinh khiết* nhất, tức một trong hai giá trị  $p_i$  bằng 1, giá trị còn lại bằng 0, entropy của phân phối này là  $H(p) = 0$ . Khi  $p$  là *vẫn được* nhất, tức cả hai giá trị  $p_i = 0.5$ , hàm entropy đạt giá trị cao nhất.



Hình Error! No text of specified style in document..9 Đồ thị của hàm entropy với  $n=2$ (  
nguồn : internet)

Tổng quát lên với  $n > 2$ , khi có một giá trị  $p_i = 1$  hàm entropy đạt giá trị nhỏ nhất, ngược lại hàm đạt giá trị lớn nhất nếu tất cả các  $p_i$  bằng nhau (việc này có thể được chứng minh bằng phương pháp nhân tử Lagrange).

ID3 còn được gọi là entropy-based decision tree vì nó sử dụng hàm entropy trong việc đo độ vẩn đục của một phép phân chia.

### Thuật toán ID3

Trong thuật toán ID3, quá trình xây dựng cây quyết định có thể được nhìn nhận như một quá trình tối ưu hóa hàm mất mát, trong đó hàm mất mát chính là tổng entropy có trọng số tại các node lá của cây sau khi hoàn tất. Mỗi node lá chứa một số lượng điểm dữ liệu nhất định, và do đó trọng số tương ứng chính là tỷ lệ (hoặc số lượng tuyệt đối) của dữ liệu thuộc về node đó.

Mục tiêu của ID3 là xây dựng cây quyết định sao cho tổng entropy tại các node lá này nhỏ nhất có thể. Điều này tương đương với việc tạo ra một cây phân loại mà dữ liệu tại mỗi nhánh là càng “thuần nhất” càng tốt, tức có xác suất thuộc về một lớp gần 1 (entropy thấp). Để đạt được điều đó, thuật toán ID3 lặp đi lặp lại việc tìm kiếm thuộc tính tối ưu nhất để phân chia dữ liệu tại mỗi bước. Cụ thể, tại mỗi nút chưa phải lá (non-leaf node):

- ID3 tính lượng thông tin thu được (Information Gain) nếu chia dữ liệu tại node đó theo từng thuộc tính có thể.
- Chọn ra thuộc tính có Information Gain lớn nhất, tương đương với việc giảm được nhiều entropy nhất sau khi phân chia.
- Tiến hành phân chia dữ liệu thành các nhánh con ứng với từng giá trị của thuộc tính đã chọn.

Quá trình này tiếp tục đệ quy trên từng nhánh con, cho đến khi dữ liệu tại node là hoàn toàn đồng nhất (entropy = 0) hoặc không còn thuộc tính nào để chia.

Như vậy, bài toán lớn – xây dựng toàn bộ cây – được chia nhỏ thành các bài toán con, mỗi bài toán là việc chọn thuộc tính tối ưu để phân chia dữ liệu tại một node cụ thể. Chúng ta sẽ xây dựng phương pháp tính toán dựa trên mỗi node này.

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

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

Tiếp theo, giả sử thuộc tính được chọn là  $x$ . Dựa trên  $x$ , các điểm dữ liệu trong  $S$  được phân ra thành  $K$  child node  $S_1, S_2, \dots, S_K$  với số điểm trong mỗi child node lần lượt là  $m_1, m_2, \dots, m_K$ . Ta định nghĩa

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

là tổng có trọng số entropy của mỗi child node. Các node thường có số lượng điểm khác nhau nên việc lấy trọng số này là cần thiết.

Tiếp theo, ta tính chỉ số Gain information dựa trên thuộc tính  $x$ :

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

Trong ID3, tại mỗi node, thuộc tính được chọn được xác định dựa trên:

$$x^* = \arg \max_x G(x, S) = \arg \min_x H(x, S)$$

tức thuộc tính khiến cho chỉ số Gain information đạt giá trị lớn nhất.

## b) Ví dụ

Để làm rõ hơn cách hoạt động của thuật toán, chúng ta sẽ cùng xem xét một ví dụ sử dụng tập dữ liệu huấn luyện được trình bày trong bảng dưới đây. Tập dữ liệu này được trích từ cuốn *Data Mining: Practical Machine Learning Tools and Techniques* (trang 11) và là một ví dụ kinh điển thường được sử dụng trong các bài giảng về cây quyết định.

Tập dữ liệu mô tả mối liên hệ giữa các yếu tố thời tiết trong vòng 14 ngày (được thể hiện qua bốn thuộc tính đầu tiên, không bao gồm cột ID) và quyết định của một đội bóng về việc có ra sân thi đấu hay không (thể hiện ở cột cuối cùng). Nói cách khác, bài toán đặt ra là: dựa vào giá trị của bốn thuộc tính thời tiết, hãy dự đoán giá trị của thuộc tính đầu ra — liệu đội bóng có chơi bóng hay không.

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

*Hình Error! No text of specified style in document..10 Bảng giá trị thời tiết( nguồn : internet)*

Tập dữ liệu này bao gồm bốn thuộc tính thời tiết như sau:

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

Bài toán này có thể được xem là một bài toán dự đoán liệu đội bóng có ra sân thi đấu hay không, dựa trên các yếu tố thời tiết quan sát được. Tất cả các thuộc tính trong tập dữ liệu đều mang tính phân loại (categorical). Một số quy tắc ra quyết định đơn giản nhưng khá chính xác dù chưa hẳn là tối ưu có thể được nêu như sau:

- Nếu outlook = sunny và humidity = high, thì play = no
- Nếu outlook = rainy và wind = strong, thì play = no
- Nếu outlook = overcast, thì play = yes
- Ngoài ra, nếu humidity = normal, thì play = yes
- Trong các trường hợp còn lại, play = yes

Ta thấy, trong 14 giá trị đầu ra ở Bảng trên, có năm giá trị là no và chín giá trị là yes. Entropy tại root node của bài toán là:

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

Tiếp theo, chúng ta sẽ tính tổng entropy có trọng số tại các node con (child node) nếu phân chia dữ liệu theo từng thuộc tính: Outlook, Temperature, Humidity, Wind hoặc Play.

Xét thuộc tính Outlook: thuộc tính này có thể nhận một trong ba giá trị là sunny, overcast và rainy. Mỗi giá trị tương ứng với một node con trong cây quyết định. Gọi tập các điểm dữ liệu rơi vào mỗi node con này lần lượt là  $S_s, S_o, S_r$  với số lượng phần tử tương ứng là  $m_s, m_o, m_r$  phần tử. Khi sắp xếp lại bảng dữ liệu ban đầu theo giá trị của thuộc tính Outlook, ta thu được ba bảng nhỏ tương ứng với ba giá trị sunny, overcast và rainy.

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

*Hình Error! No text of specified style in document..11 Bảng giá trị theo outlook là sunny( nguồn : internet)*

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

*Hình Error! No text of specified style in document..12 Bảng giá trị theo outlook là overcast( nguồn : internet)*

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

*Hình Error! No text of specified style in document..13 Bảng giá trị theo outlook là rainy( nguồn : internet)*

Quan sát sơ bộ cho thấy rằng node con tương ứng với giá trị outlook = overcast có entropy bằng 0, vì tất cả các mẫu  $m_o = 4$  trong nhóm này đều có kết quả đầu ra là yes. Trong khi đó, hai node còn lại  $m_s = m_r = 5$  (ứng với sunny và rainy) có entropy tương đối cao, do tỉ lệ giữa các mẫu có kết quả yes và no gần như ngang nhau. Tuy nhiên, hai node này vẫn có thể tiếp tục được phân chia, dựa trên các thuộc tính còn lại là humidity và wind. Tiếp theo, ta xét đến thuộc tính Temperature. Khi phân chia dữ liệu theo thuộc tính này, ta thu được các bảng nhỏ như sau:

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

*Hình Error! No text of specified style in document..14 Bảng giá trị theo temperature là hot( nguồn : internet)*



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

*Hình Error! No text of specified style in document..15 Bảng giá trị theo temperature là mild( nguồn : internet)*

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

*Hình Error! No text of specified style in document..16 Bảng giá trị theo temperature là cool( nguồn : internet)*

Gọi  $S_h, S_m, S_c$  là ba tập con tương ứng với các giá trị của thuộc tính temperature lần lượt là: hot, mild và cool.

Việc tính toán chi tiết cho hai thuộc tính còn lại (Humidity và Wind) được để lại cho người đọc. Nếu các kết quả tính toán này là tương đương nhau, ta sẽ có:

$$H(\text{humidity}, S) \approx 0.547, \quad H(\text{wind}, S) \approx 0.618$$

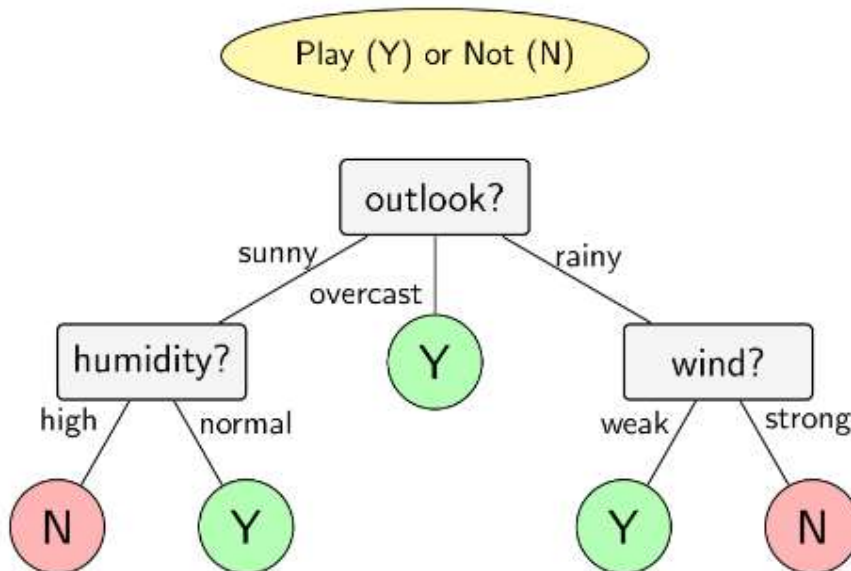
⇒ Thuộc tính cần được chọn ở bước đầu tiên là Outlook, vì việc phân chia theo thuộc tính này cho tổng entropy  $H(\text{outlook}, S)$  có giá trị thấp nhất (tức là Gain information lớn nhất).

Sau bước phân chia đầu tiên, ta thu được ba child node tương ứng với ba giá trị của Outlook. Trong đó, node con thứ hai Outlook = overcast là node tinh khiết vì toàn bộ output là yes, nên không cần phân chia tiếp.

Với node con thứ nhất Outlook = sunny, khi áp dụng thuật toán ID3, ta chọn thuộc tính Humidity để phân chia, vì tổng entropy có trọng số sau bước này sẽ bằng 0 với output sẽ là yes khi và chỉ khi humidity = normal.

Tương tự, node con tương ứng với Outlook = rainy sẽ tiếp tục được phân chia theo thuộc tính Wind, với kết quả là play = yes khi và chỉ khi wind = weak.

Như vậy, cây quyết định cho bài toán này được xây dựng theo thuật toán ID3 sẽ có cấu trúc như được minh họa trong hình dưới đây.



Hình Error! No text of specified style in document..17 Cây quyết định ID3( nguồn : internet)

### c) Điều kiện dừng

Trong các thuật toán cây quyết định nói chung, và ID3 nói riêng, nếu ta tiếp tục phân chia tất cả các node chưa thuần nhất đến cùng, ta có thể tạo ra một cây mà mọi điểm trong tập huấn luyện đều được phân loại chính xác (giả định không có hai mẫu đầu vào giống nhau nhưng cho ra kết quả khác nhau). Tuy nhiên, điều này dẫn đến tree có cấu trúc rất phức tạp (nhiều node), với nhiều leaf node chỉ chứa một số lượng rất ít dữ liệu. Hệ quả là mô hình dễ bị quá khớp (overfitting) với dữ liệu huấn luyện.

Để hạn chế hiện tượng overfitting, có thể áp dụng một số điều kiện dừng, tức là tại một node, nếu thỏa mãn một trong các điều kiện sau thì ta không tiếp tục phân chia nữa và coi đó là node lá:

- Entropy tại node bằng 0, tức toàn bộ dữ liệu tại node thuộc cùng một lớp.

- Số lượng phần tử trong node nhỏ hơn một ngưỡng nhất định. Trong trường hợp này, dù node chưa tinh khiết, ta chấp nhận cho một số điểm bị phân loại sai để tránh làm mô hình quá chi tiết. Class của leaf node có thể lấy theo class chiếm đa số trong node.
- Chiều sâu của node tính từ root node vượt quá một giá trị cho trước. Việc giới hạn độ sâu giúp giảm độ phức tạp của tree và góp phần chống overfitting.
- Tổng số leaf node trong toàn bộ tree vượt quá một ngưỡng định sẵn.
- Gain information tại node nhỏ hơn một giá trị ngưỡng. Khi việc phân chia không làm giảm entropy đáng kể, việc tiếp tục chia là không hiệu quả.

Ngoài ra, một kỹ thuật thường được sử dụng để giảm overfitting là pruning (tạm dịch là cắt tỉa), trong đó một số nhánh của cây sẽ được loại bỏ sau khi cây đã được xây dựng xong nhằm tăng khả năng tổng quát hóa của mô hình.

#### 2.2.2.2 Thuật toán C4.5

##### a) Khái niệm

Thuật toán C4.5 là một phiên bản cải tiến của thuật toán ID3.

Trong ID3, việc lựa chọn thuộc tính để phân chia dữ liệu dựa trên tiêu chí Information Gain. Tuy nhiên, cách tiếp cận này có xu hướng ưu tiên các thuộc tính có nhiều giá trị, dẫn đến việc lựa chọn không tối ưu trong một số trường hợp, đặc biệt khi thuộc tính có nhiều mức phân chia nhưng không thật sự mang nhiều ý nghĩa phân loại.

Để khắc phục nhược điểm đó, thuật toán C4.5 sử dụng một độ đo mới gọi là Gain Ratio, được tính như sau:

Đầu tiên, ta chuẩn hoá information gain với trị thông tin phân tách (split information):

$$\text{Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Info}}$$

Trong đó: Split Info được tính như sau:

$$-\sum_{i=1}^n D_i \log_2 D_i$$

## b) Điều kiện dừng

Trong các thuật toán cây quyết định Decision Tree, nếu áp dụng phương pháp phân chia liên tục tại các node chưa tinh khiết, ta sẽ tiếp tục chia đến khi mọi điểm trong tập huấn luyện được dự đoán chính xác (giả sử không tồn tại hai mẫu đầu vào giống nhau cho kết quả khác nhau). Khi đó, mô hình sẽ trở nên quá phức tạp, với nhiều node và các leaf node chỉ chứa rất ít điểm dữ liệu. Điều này dẫn đến khả năng overfitting cao, tức là mô hình hoạt động tốt trên tập huấn luyện nhưng kém hiệu quả trên dữ liệu mới.

Để tránh tình trạng trên, có thể dừng quá trình xây dựng cây dựa trên một số tiêu chí sau:

- Node có entropy bằng 0, tức toàn bộ mẫu tại node thuộc cùng một class.
- Số lượng phần tử trong node nhỏ hơn một ngưỡng xác định. Khi đó, chấp nhận có thể xảy ra sai lệch nhỏ để tránh overfitting. Nhãn cho leaf node có thể được chọn theo class chiếm đa số.
- Chiều sâu từ node đến gốc đạt một giá trị giới hạn, nhằm kiểm soát độ phức tạp của cây.
- Tổng số node lá vượt quá một ngưỡng nhất định, giúp kiểm soát kích thước toàn bộ cây.
- Information Gain sau khi chia nhỏ hơn một giá trị ngưỡng, nghĩa là việc chia không còn mang lại hiệu quả rõ rệt.

Ngoài ra, còn có thể áp dụng kỹ thuật cắt tỉa cây (pruning) sau khi cây đã được xây dựng hoàn chỉnh, nhằm giảm độ phức tạp và cải thiện khả năng tổng quát hóa của mô hình.

## 2.3 Công cụ sử dụng xây dựng bài toán

### 2.3.1 Ngôn ngữ lập trình Python

#### 2.3.1.1 Python là gì ?

Python là một ngôn ngữ lập trình mã nguồn mở được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển phần mềm, khoa học dữ liệu, và học máy.

Với cấu trúc cú pháp rõ ràng, dễ đọc, dễ viết, Python đặc biệt phù hợp với người mới bắt đầu học lập trình. Chính sự đơn giản và linh hoạt này đã góp phần làm nên sự phổ biến rộng rãi của Python trên toàn thế giới.

Python là một ngôn ngữ đa nền tảng, hỗ trợ nhiều mô hình lập trình khác nhau như: Lập trình mệnh lệnh (imperative programming), Lập trình hướng đối tượng (object-oriented programming), Lập trình hàm (functional programming),...

Python còn được ứng dụng trong nhiều lĩnh vực như: Phát triển web, phân tích và trực quan hóa dữ liệu, thiết kế 3D CAD, tự động hóa tác vụ,...

Hiện nay, Python được sử dụng rộng rãi tại các công ty công nghệ hàng đầu như Google, Amazon, Facebook, Instagram, Dropbox, Uber, và nhiều doanh nghiệp khác.



*Hình Error! No text of specified style in document..18 Minh họa ngôn ngữ lập trình Python( nguồn : internet)*

### 2.3.1.2 Một số ứng dụng của Python

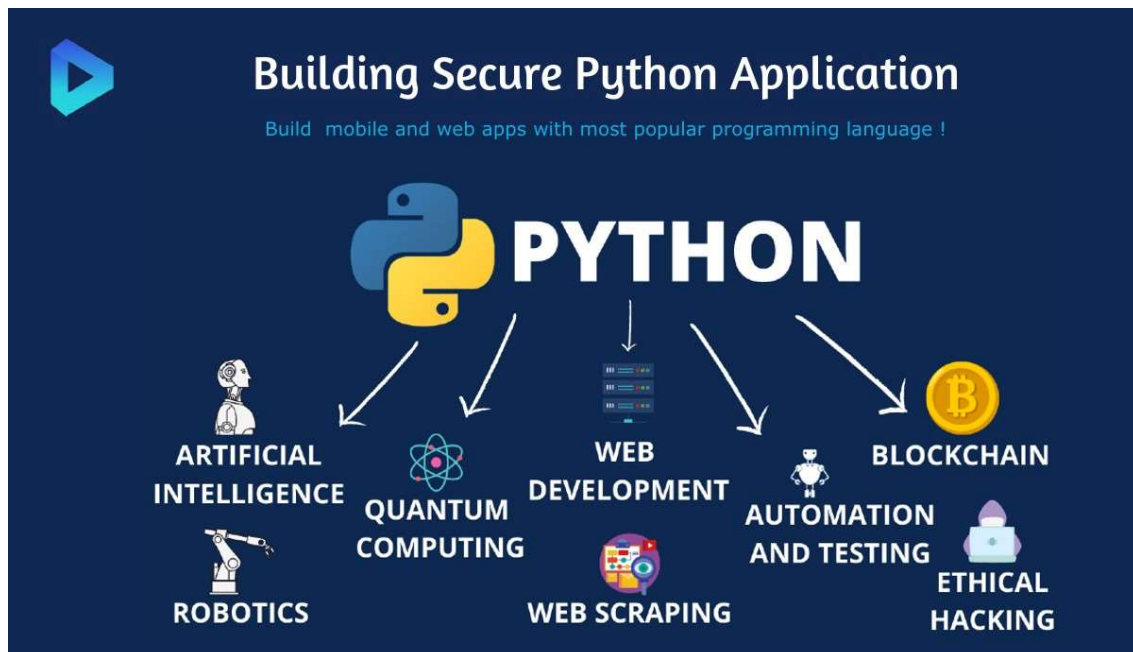
Ngôn ngữ Python hiện đang được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau nhờ vào tính linh hoạt và khả năng mở rộng mạnh mẽ. Một số lĩnh vực ứng dụng tiêu biểu của Python bao gồm:

- Học máy (Machine Learning) và trí tuệ nhân tạo (AI)

- Phân tích và trực quan hóa dữ liệu
- Phát triển ứng dụng giao diện người dùng (GUI) với các thư viện như: Kivy, Tkinter, PyQt, Qt Designer,...
- Phát triển web với các framework nổi bật như: Django, Flask (Django đang được sử dụng bởi các nền tảng lớn như YouTube, Instagram, Dropbox)
- Xử lý hình ảnh với thư viện OpenCV
- Web scraping (quét dữ liệu từ web) bằng Scrapy, BeautifulSoup, Selenium
- Xử lý văn bản và các tác vụ tự động hóa khác

### 2.3.1.3 Một số tính năng chính của Python

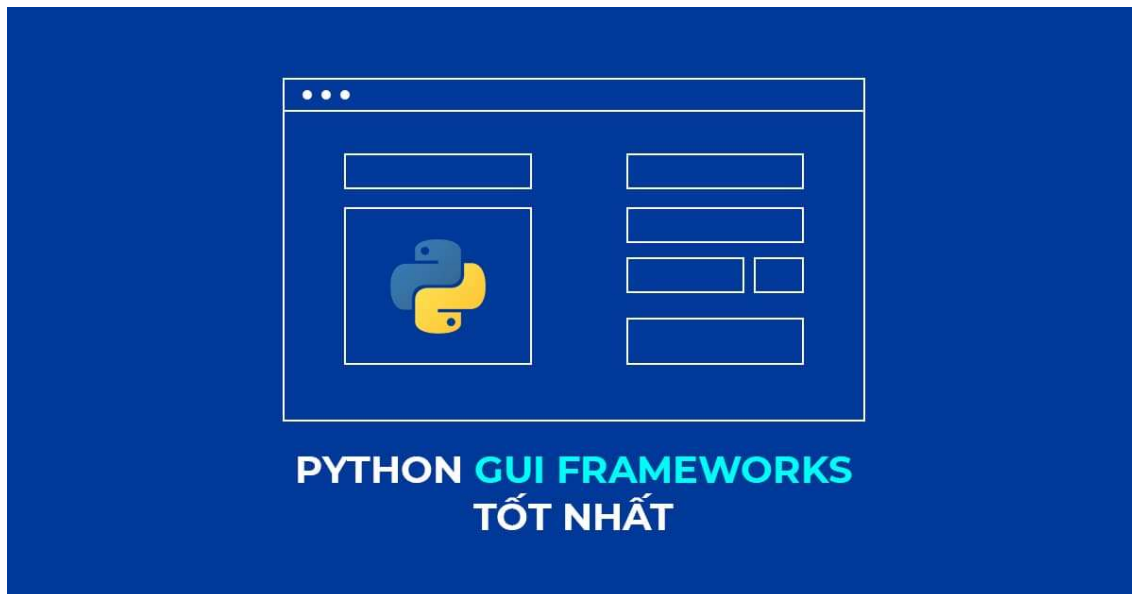
- Phát triển website phía máy chủ
- Phát triển các ứng dụng phần mềm đa nền tảng
- Tự động hóa kiểm thử phần mềm
- Hỗ trợ kết nối và tương tác với nhiều hệ quản trị cơ sở dữ liệu
- Thích hợp cho xử lý dữ liệu lớn (big data) và thực hiện các phép toán phức tạp



Hình Error! No text of specified style in document..19 Python và các ứng dụng trong thực tế( nguồn : internet)

### 2.3.2 Các Python GUI Frameworks tốt nhất

Python là một ngôn ngữ dễ học và được xem là một trong những ngôn ngữ lập trình nổi bật nhất hiện nay. Đặc biệt, việc sử dụng Python để xây dựng các ứng dụng có giao diện người dùng đồ họa (GUI – Graphical User Interface) cũng tương đối dễ dàng và thuận tiện.



Hình Error! No text of specified style in document..20 Python GUI Frameworks( nguồn : internet)

Python cung cấp rất nhiều framework GUI mạnh mẽ giúp việc thiết kế giao diện trở nên đơn giản và hiệu quả. Các framework này bao gồm cả đa nền tảng (cross-platform) lẫn dành riêng cho từng nền tảng (platform-specific). Dưới đây là 6 framework GUI phổ biến và được đánh giá cao trong cộng đồng Python.

#### 2.3.2.1 Python GUI Frameworks #1: Kivy

- Kivy là một framework mã nguồn mở sử dụng OpenGL ES 2 để tăng tốc giao diện người dùng hiện đại. Kivy có khả năng chạy trên nhiều hệ điều hành như Linux, Windows, macOS, Android, iOS và cả Raspberry Pi. Đặc biệt, cùng một đoạn mã có thể chạy đồng nhất trên tất cả các nền tảng này.

- Kivy hỗ trợ nguyên bản nhiều dạng đầu vào và thiết bị như: WM\_Touch, WM\_Pen, Mac OS X Trackpad, Magic Mouse, Mtdev, Linux Kernel HID, TUIO, và cả mô phỏng cảm ứng đa điểm. Điều này giúp ứng dụng có thể xử lý linh hoạt tương tác người dùng từ nhiều nguồn khác nhau.
- Framework này hoàn toàn miễn phí 100%, được phát hành theo giấy phép MIT (từ phiên bản 1.7.2 trở đi) và LGPL 3 cho các phiên bản cũ hơn. Kivy phù hợp để sử dụng cả trong dự án cá nhân lẫn sản phẩm thương mại.
- Ngoài ra, Kivy là một framework ổn định, có tài liệu API đầy đủ, cùng với hướng dẫn lập trình chi tiết, giúp người dùng có thể bắt đầu nhanh chóng và dễ dàng.

#### 2.3.2.2 Python GUI Frameworks #2: PyQT

- PyQt là một trong những framework GUI đa nền tảng phổ biến nhất, được xây dựng dưới dạng ràng buộc Python với thư viện Qt — một bộ công cụ phát triển giao diện người dùng mạnh mẽ (trước đây thuộc sở hữu của Nokia).
- Hiện nay, PyQt hỗ trợ nhiều hệ điều hành, bao gồm: Unix/Linux, Windows, macOS, và cả Sharp Zaurus. Framework này kết hợp những ưu điểm của Python và Qt, cho phép lập trình viên viết giao diện bằng code thủ công hoặc thiết kế trực quan bằng Qt Designer.
- PyQt được phát hành dưới hai loại giấy phép: thương mại và GPL. Trong trường hợp phát triển phần mềm mã nguồn mở, lập trình viên có thể sử dụng phiên bản miễn phí theo giấy phép GPL. Tuy nhiên, một số tính năng nâng cao chỉ có trong phiên bản thương mại.

#### 2.3.2.3 Python GUI Frameworks #3: Tkinter

- Tkinter là framework GUI tiêu chuẩn đi kèm với Python, tức là nó thường được cài sẵn theo mặc định. Nhờ vào tính đơn giản, mã nguồn mở và dễ học, Tkinter là lựa chọn phổ biến cho những người mới bắt đầu lập trình GUI với Python.
- Một lợi thế lớn của Tkinter là sự phổ biến và hỗ trợ rộng rãi từ cộng đồng. Có rất nhiều tài nguyên học tập, ví dụ, tài liệu và sách tham khảo. Với một cộng



đồng lâu đời và tích cực, lập trình viên dễ dàng tìm thấy giải pháp cho các lỗi thường gặp hoặc trao đổi khi mới bắt đầu học.

#### *2.3.2.4 Python GUI Frameworks #4: WxPython*

- WxPython là một thư viện GUI mã nguồn mở, là trình bao bọc của wxWidgets (trước đây gọi là wxWindows) – một bộ công cụ GUI đa nền tảng.
- Với WxPython, lập trình viên có thể tạo ra các ứng dụng giao diện gốc (native GUI) hoạt động ổn định trên Windows, macOS và Unix/Linux. Framework này cho phép xây dựng giao diện có cảm giác và trải nghiệm giống như phần mềm bản địa trên từng nền tảng.

#### *2.3.2.5 Python GUI Frameworks #5: PyGUI*

- PyGUI là một framework GUI đa nền tảng dành cho các hệ điều hành như Unix, macOS và Windows. Điểm mạnh nổi bật của PyGUI là sự đơn giản và nhẹ, nhờ vào việc đồng bộ hóa API chặt chẽ với Python.
- PyGUI chèn rất ít lớp trung gian giữa mã Python và hệ thống GUI gốc, do đó giao diện người dùng thường hiển thị một cách tự nhiên, gần giống hoàn toàn với giao diện mặc định của hệ điều hành.

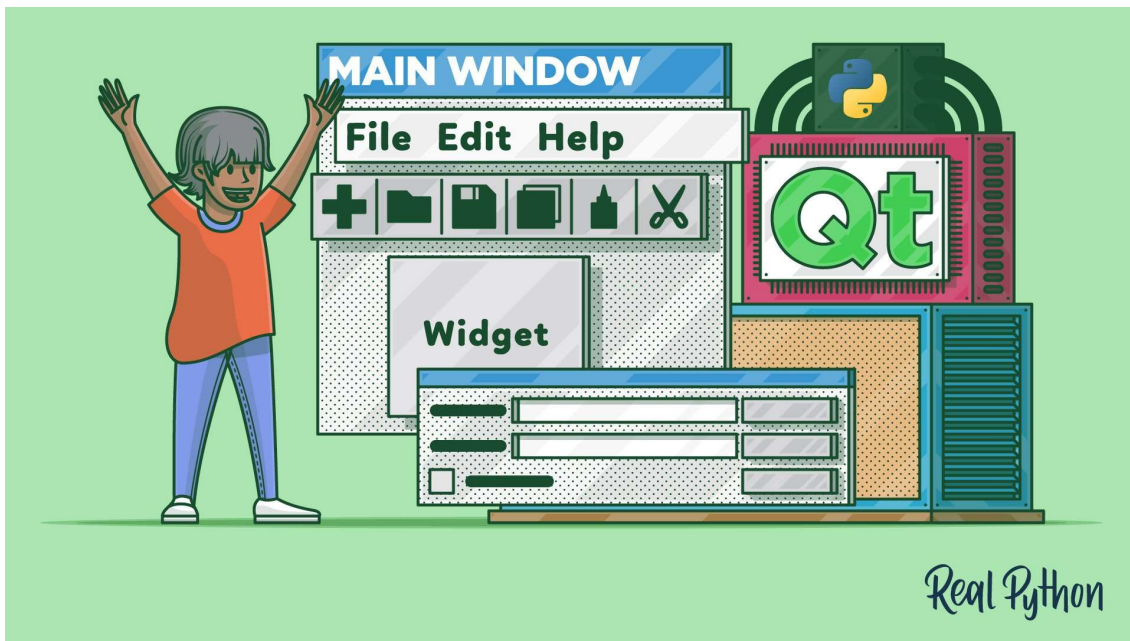
#### *2.3.2.6 Python GUI Frameworks #6: PySide*

- PySide là một dự án mã nguồn mở cung cấp ràng buộc Python cho Qt Framework, tương tự như PyQt. Qt là một bộ công cụ GUI đa nền tảng cho phép viết ứng dụng một lần và chạy trên nhiều hệ điều hành mà không cần chỉnh sửa lại mã nguồn.
- Kết hợp sức mạnh của Qt và sự linh hoạt của Python, PySide cho phép các lập trình viên phát triển ứng dụng GUI nhanh chóng và hiệu quả trên tất cả các nền tảng chính như Windows, Linux và macOS. Đây là lựa chọn tuyệt vời cho những ai muốn tận dụng hệ sinh thái Qt trong khi vẫn sử dụng Python.

### 2.2.1 Xây dựng giao diện đồ họa với Py QT5, Qt Designer

Để hỗ trợ người dùng trong quá trình tìm kiếm thông tin, đưa ra dự đoán, xếp hạng hoặc gợi ý, việc xây dựng các mô hình dự đoán dựa trên các thuật toán học máy (Machine Learning) là cần thiết. Các thuật toán này giúp xử lý và phân tích dữ liệu ở tầng bên dưới, nhằm tạo ra kết quả dễ hiểu và phù hợp với nhu cầu của người dùng.

Trong quá trình học lập trình với ngôn ngữ Python, rất nhiều người quan tâm đến việc xây dựng các ứng dụng có giao diện người dùng đồ họa (GUI) tương tự như Windows Form. Trong Python, có nhiều công cụ hỗ trợ thiết kế GUI, và trong phạm vi đề tài này, em lựa chọn sử dụng PyQt5 kết hợp với Qt Designer để xây dựng giao diện.

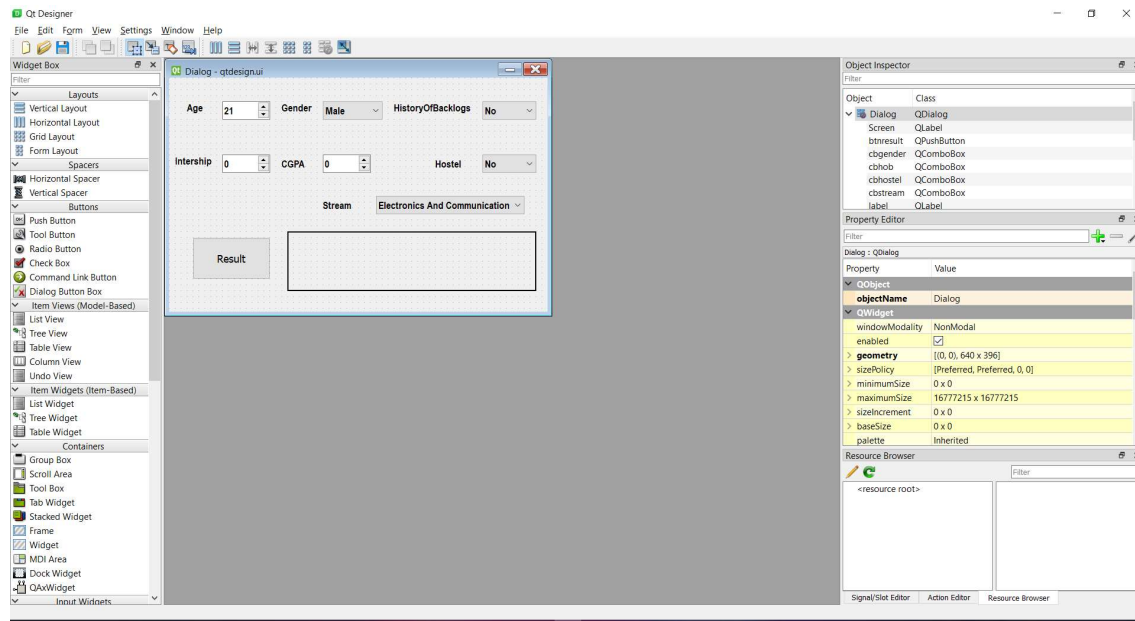


Hình 2.26 – Giao diện đồ họa với Qt Designer (Nguồn: Internet)

Bằng cách sử dụng Python để xây dựng mô hình dự đoán, em đã lựa chọn PyQt5 và Qt Designer làm công cụ phát triển giao diện, nhằm giúp người dùng – cụ thể là sinh viên – tương tác trực tiếp với hệ thống thông qua giao diện trực quan, thay vì phải làm việc với màn hình dòng lệnh (console) vốn chỉ phù hợp với lập trình viên. Qua đó, hệ thống có thể hiển thị kết quả dự đoán về cơ hội việc làm một cách thân thiện, rõ ràng và dễ sử dụng.

### 2.3.3.1 Qt Designer là gì ?

Qt Designer là một công cụ hỗ trợ thiết kế giao diện người dùng đồ họa (GUI) một cách nhanh chóng và trực quan, sử dụng các widget có sẵn từ bộ công cụ Qt GUI. Công cụ này cung cấp một giao diện kéo và thả thân thiện, giúp người dùng dễ dàng bố trí các thành phần giao diện như nút bấm (button), ô nhập văn bản (text field), hộp chọn (combo box) và nhiều thành phần khác.



Hình 2.27 – Giao diện của Qt Designer (chạy trên Windows)

Qt Designer tạo ra các tệp có phần mở rộng .ui, là các tệp định dạng XML đặc biệt dùng để lưu trữ thông tin về cấu trúc giao diện người dùng dưới dạng cây phân cấp các widget.

Các tệp .ui này có thể:

- Tải trực tiếp trong thời gian chạy (runtime) thông qua các thư viện hỗ trợ của Python như PyQt5.uic.
- Hoặc được biên dịch sang mã nguồn bằng các công cụ chuyển đổi, giúp tích hợp vào các ngôn ngữ lập trình như C++ hoặc Python để sử dụng trong chương trình chính.

### 2.3.3.2 PyQt5 là gì ?

Qt là một application framework đa nền tảng, được viết bằng ngôn ngữ C++, dùng để phát triển các ứng dụng trên desktop, hệ thống nhúng và mobile. Nó hỗ trợ nhiều nền

tảng bao gồm: Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số nền tảng khác.

PyQt là giao diện Python của Qt, là sự kết hợp giữa ngôn ngữ lập trình Python và thư viện Qt. Đây là một thư viện bao gồm các thành phần giao diện điều khiển (widgets – graphical control elements).

PyQt API bao gồm nhiều module với số lượng lớn các lớp (classes) và hàm (functions), hỗ trợ việc thiết kế giao diện người dùng cho các phần mềm chức năng. PyQt hỗ trợ cả Python 2.x và 3.x.

PyQt được phát triển bởi Riverbank Computing Limited.

Các lớp của PyQt5 được chia thành nhiều module, bao gồm:

Các module chính của PyQt5:

- QtCore: Gồm các thành phần cốt lõi không thuộc GUI, ví dụ: làm việc với thời gian, tệp và thư mục, kiểu dữ liệu, streams, URLs, MIME types, luồng (threads) và tiến trình (processes).
- QtGui: Bao gồm các lớp dùng trong lập trình giao diện như tích hợp hệ thống cửa sổ, xử lý sự kiện, đồ họa 2D, xử lý ảnh cơ bản, phông chữ và văn bản.
- QtWidgets: Gồm các lớp về widget như button, hộp thoại,... dùng để xây dựng giao diện người dùng cơ bản.
- QtMultimedia: Thư viện hỗ trợ sử dụng âm thanh, hình ảnh, camera,...
- QtBluetooth: Gồm các lớp giúp tìm kiếm và kết nối với thiết bị thông qua Bluetooth.
- QtNetwork: Hỗ trợ lập trình mạng, bao gồm client/server TCP/IP và UDP.
- QtPositioning: Hỗ trợ các tính năng định vị.
- Enginio: Module cho phép các client truy cập dịch vụ đám mây của Qt.
- QtWebSockets: Cung cấp công cụ để làm việc với giao thức WebSocket.
- QtWebKit: Cung cấp các lớp để xử lý trình duyệt web, dựa trên thư viện WebKit2.

- QtWebKitWidgets: Các widget hỗ trợ cho WebKit.
- QtXml: Làm việc với tệp XML.
- QtSvg: Hiển thị các thành phần từ tệp SVG.
- QSql: Làm việc với cơ sở dữ liệu.
- QTest: Cung cấp công cụ để kiểm thử đơn vị (unit test) trong ứng dụng PyQt5.

Giả sử bạn đã lưu tệp của mình từ Qt Designer dưới dạng dialog.ui. Sau đó, bạn có thể tạo một tệp khác, chẳng hạn như main.py, với nội dung sau:

- Cách 1:

```
from PyQt5 import uic

from PyQt5.QtWidgets import QApplication

Form, Window = uic.loadUiType("dialog.ui")

app = QApplication([])

window = Window()

form = Form()

form.setupUi(window)

window.show()

app.exec()
```

- Cách 2:

```
from PyQt5 import QtWidgets, uic

import sys

class Ui(QtWidgets.QMainWindow):
```

```

def __init__(self):
    super(Ui, self).__init__()
    uic.loadUi('dialog.ui', self)
    self.show()

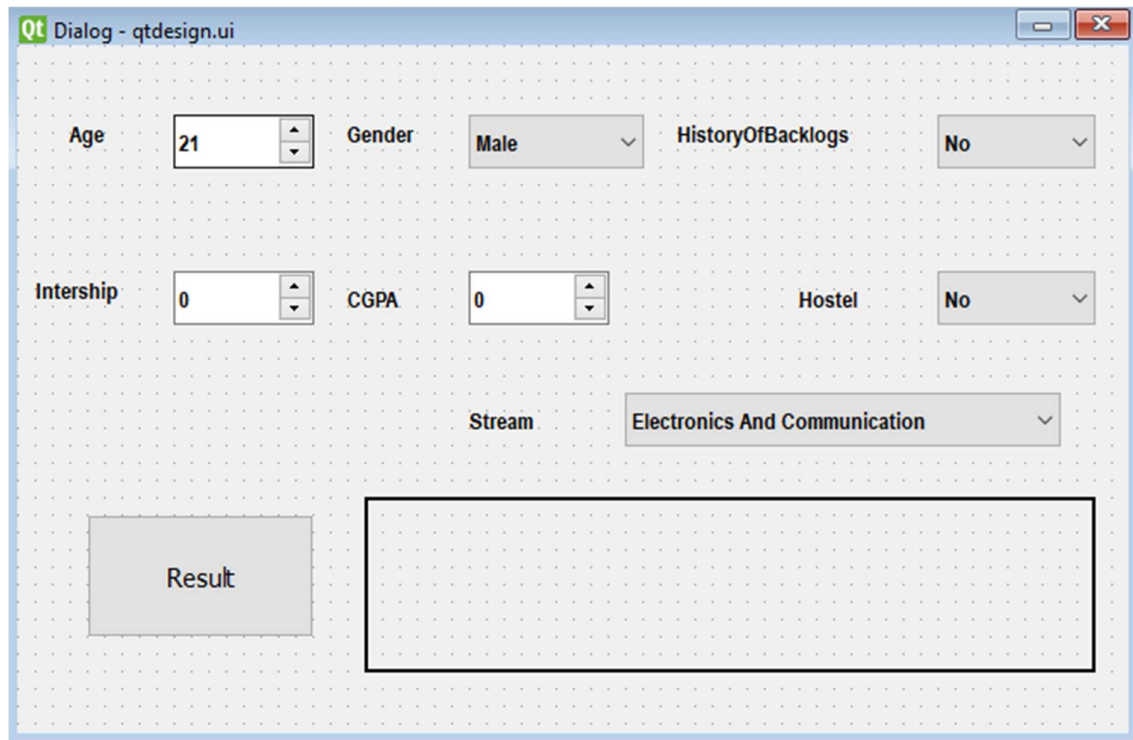
app = QtWidgets.QApplication(sys.argv)

window = Ui()

app.exec_()

```

Sau đó, khi bạn gọi python main.py trên dòng lệnh, hộp thoại của bạn sẽ mở ra:



*Hình Error! No text of specified style in document..21 Giao diện tương tác trong qt designer*

Ngoài ra, các bạn cũng có thể chuyển đổi từ tệp .ui (tệp giao diện người dùng) sang tệp .py (tệp Python) bằng câu lệnh trên **Terminal** như sau:

# Chuyển đổi tệp .ui sang .py bằng module PyQt5

- `pyuic5 -x main.ui -o main.py` # Dành cho phiên bản PyQt5
- `pyuic4 -x main.ui -o main.py` # Dành cho phiên bản PyQt4

### 2.3.3.3 Mục đích và khả năng của Qt

- Qt được sử dụng để phát triển giao diện người dùng đồ họa (GUI) và các ứng dụng đa nền tảng, có thể chạy trên tất cả các nền tảng máy tính để bàn chính và hầu hết các nền tảng di động hoặc nhúng. Hầu hết các chương trình GUI được xây dựng bằng Qt đều có giao diện tự nhiên, vì vậy Qt được phân loại như một widget toolkit. Ngoài ra, Qt cũng hỗ trợ phát triển các ứng dụng phi-GUI, chẳng hạn như công cụ dòng lệnh hoặc console dành cho máy chủ. Một ví dụ về ứng dụng không có giao diện đồ họa sử dụng Qt là Cutelyst, một framework web.
- Qt hỗ trợ nhiều trình biên dịch khác nhau, bao gồm GCC C++, Visual Studio, và có hỗ trợ quốc tế hóa toàn diện. Qt cũng cung cấp Qt Quick, bao gồm một ngôn ngữ kịch bản gọi là QML, cho phép sử dụng JavaScript để xử lý logic. Với Qt Quick, việc phát triển ứng dụng nhanh chóng cho thiết bị di động trở nên dễ dàng hơn, trong khi phần logic có thể được viết bằng mã gốc để đảm bảo hiệu năng tối ưu.
- Các tính năng nổi bật khác của Qt bao gồm:
  - Truy cập cơ sở dữ liệu SQL
  - Phân tích cú pháp XML và JSON
  - Quản lý luồng xử lý (multithreading)
  - Hỗ trợ lập trình mạng (networking)

### 2.3.3.4 Các ứng dụng được xây dựng bằng Qt

Hiện nay có nhiều phần mềm mã nguồn mở và thương mại được phát triển dựa trên Qt, ví dụ:

- LyX: Phần mềm soạn thảo văn bản LaTeX
- Quantum GIS (QGIS): Phần mềm hệ thống thông tin địa lý
- QCad: Phần mềm vẽ kỹ thuật CAD
- Scribus: Phần mềm xuất bản điện tử (desktop publishing)

- Skype: Phần mềm giao tiếp qua Internet

Một số thống kê cho thấy Qt không chỉ được sử dụng trong các ứng dụng máy tính, mà còn xuất hiện rộng rãi trong các thiết bị nhúng và đồ điện gia dụng.

#### 2.3.4 Trình soạn thảo Pycharm

PyCharm là một nền tảng hybrid được JetBrains phát triển như một IDE dành cho Python. Nó thường được sử dụng để phát triển các ứng dụng Python. Một số công ty lớn như Twitter, Facebook, Amazon và Pinterest cũng sử dụng PyCharm làm môi trường phát triển chính cho Python của họ.



Hình 2.29 Biểu tượng của PyCharm

PyCharm có thể chạy trên Windows, Linux hoặc Mac OS. Ngoài ra, nó tích hợp các module và package hỗ trợ giúp lập trình viên tiết kiệm thời gian và công sức khi phát triển phần mềm bằng Python. Hơn nữa, PyCharm còn có thể được tùy chỉnh theo nhu cầu của từng nhà phát triển.

Một số đặc điểm nổi bật:

- Hỗ trợ đa dạng ngôn ngữ lập trình như C/C++, C#, F#, JavaScript, JSON, Visual Basic, HTML, CSS,...



- Ngôn ngữ và giao diện tối giản, thân thiện, giúp lập trình viên dễ dàng tập trung và định hình nội dung.
- Tích hợp các tính năng quan trọng như bảo mật (Git), tăng tốc xử lý vòng lặp (Debug),...
- Hỗ trợ đa nền tảng: Linux, Mac, Windows,...
- Gọn nhẹ về dung lượng
- Tính năng mạnh mẽ
- Kiến trúc mở rộng, dễ khai thác theo nhu cầu người dùng

#### 2.3.4.1 Các tính năng của PyCharm

##### 1. Trình sửa mã thông minh

- Giúp viết mã chất lượng cao hơn
- Bao gồm các lược đồ màu cho từ khóa, lớp và hàm, tăng khả năng đọc hiểu mã
- Xác định lỗi dễ dàng
- Tự động hoàn thành và hướng dẫn hoàn tất mã

##### 2. Điều hướng mã

- Hỗ trợ chỉnh sửa và cải tiến mã nhanh chóng, tiết kiệm thời gian
- Cho phép điều hướng đến các hàm, lớp hoặc tệp dễ dàng
- Xác định nhanh vị trí phân tử, biến, hoặc ký hiệu trong mã nguồn
- Sử dụng chế độ lens mode để kiểm tra và debug toàn bộ mã hiệu quả

##### 3. Tái cấu trúc

- Hỗ trợ thay đổi nhanh với biến cục bộ và toàn cục
- Cải thiện cấu trúc nội bộ mà không làm thay đổi hiệu suất bên ngoài
- Hỗ trợ chia nhỏ lớp và hàm với phương pháp extract method

##### 4. Hỗ trợ công nghệ web

- Cho phép phát triển ứng dụng web bằng Python
- Hỗ trợ HTML, CSS, JavaScript, AngularJS, NodeJS
- Cho phép chỉnh sửa và xem trước web trực tiếp từ IDE
- Theo dõi thay đổi trực tiếp trên trình duyệt

#### 5. Hỗ trợ framework web Python

- Tích hợp Django, web2py, Pyramid
- Tự động điền và gợi ý tham số Django
- Hỗ trợ debug mã Django hiệu quả

#### 6. Hỗ trợ thư viện khoa học Python

- Tích hợp Matplotlib, NumPy, Anaconda
- Hỗ trợ xây dựng các dự án về Data Science và Machine Learning
- Hỗ trợ biểu đồ tương tác và tích hợp với IPython, Django, Pytest,...

#### 7. Ưu và nhược điểm của PyCharm

##### Ưu điểm:

- Cài đặt đơn giản
- Giao diện thân thiện, dễ sử dụng
- Nhiều plugin và phím tắt hữu ích
- Tích hợp tốt với thư viện và IDE
- Hỗ trợ xem mã nguồn chỉ với một cú nhấp chuột
- Tiết kiệm thời gian phát triển
- Tính năng đánh dấu lỗi nâng cao chất lượng code
- Cộng đồng người dùng Python lớn, dễ tìm tài nguyên và hỗ trợ

##### Nhược điểm:

- Không miễn phí (phiên bản Professional có giá khá cao)
- Tự động hoàn thành có thể gây khó cho người mới học

- Có thể gặp sự cố khi xử lý môi trường ảo (venv)

### ***2.3.5 Một số thư viện được sử dụng***

#### ***2.3.5.1 Thư viện Scikit-learn***

Scikit-learn (sklearn) là một trong những thư viện mạnh mẽ và phổ biến nhất cho các thuật toán học máy cổ điển, được viết bằng Python. Thư viện này cung cấp công cụ xử lý các bài toán machine learning và thống kê. Nó hỗ trợ hầu hết các thuật toán học có giám sát và không giám sát, và có thể chạy được trên nhiều nền tảng khác nhau.

Để sử dụng scikit-learn, trước tiên cần cài đặt thư viện SciPy. Một số thành phần chính gồm:

- NumPy: Xử lý mảng số và ma trận đa chiều
- category\_encoders: Chuyển đổi dữ liệu về dạng máy có thể hiểu
- Cross Validation: Kiểm thử chéo (ví dụ: K-Fold)
- IPython: Công cụ tương tác trực quan với Python
- SymPy: Hỗ trợ biểu thức toán học
- Pandas: Phân tích và xử lý dữ liệu dạng bảng

Trong đó, NumPy và Pandas là hai thư viện quan trọng và thường được sử dụng nhất.



*Hình 2.30 Minh họa thư viện Scikit-learn*

#### 2.3.5.2 Thư viện NumPy

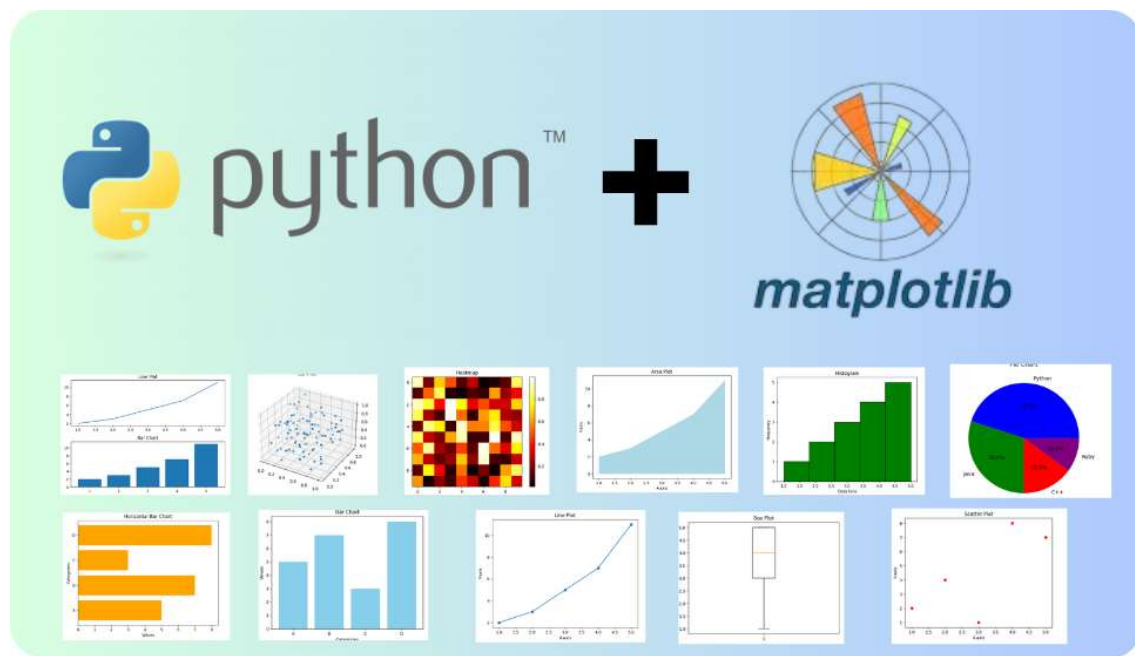
NumPy là thư viện Python nổi bật để xử lý các mảng và ma trận lớn đa chiều. Nó cung cấp một tập hợp các hàm toán học cấp cao hữu ích cho tính toán khoa học, đại số tuyến tính, biến đổi Fourier, và xử lý số ngẫu nhiên. Đây là một nền tảng quan trọng trong học máy.



*Hình 2.31 Minh họa thư viện NumPy*

#### 2.3.5.3 Thư viện Matplotlib

Matplotlib là thư viện phổ biến để trực quan hóa dữ liệu trong Python. Nó được sử dụng để hiển thị dữ liệu dưới dạng các loại biểu đồ như biểu đồ đường, biểu đồ cột,... và đặc biệt hữu ích khi cần phân tích trực quan các mẫu dữ liệu.



Hình 2.32 Minh họa thư viện Matplotlib

#### 2.3.5.4 Thư viện Pandas

Pandas là thư viện Python phổ biến hỗ trợ phân tích dữ liệu. Nó cung cấp cấu trúc dữ liệu tối ưu và công cụ mạnh mẽ để xử lý dữ liệu chuỗi thời gian và dữ liệu có cấu trúc. Pandas thường được sử dụng trong khoa học dữ liệu, học máy và phân tích dữ liệu.

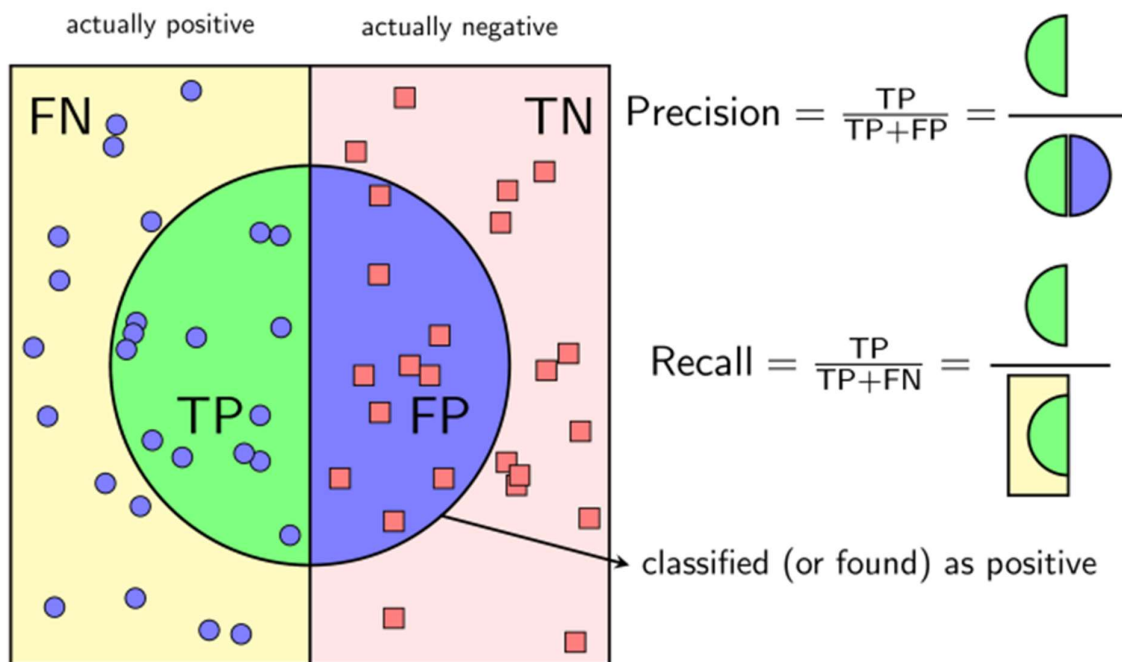
Các tính năng nổi bật bao gồm: đọc/ghi dữ liệu, lọc, nhóm, nối,... dữ liệu.



Hình 2.33 Minh họa thư viện Pandas (Nguồn: Koodibar)

## 2.4 Các phương pháp đánh giá độ tin cậy của mô hình

Precision và Recall



Hình 2.34 Độ đo tin cậy Precision và Recall

- Precision càng cao, tức là số lượng dự đoán "Positive" đúng càng nhiều. Nếu Precision = 1, thì tất cả các dự đoán là Positive đều đúng, không có trường hợp nào bị nhầm lẫn với Negative.
- Recall càng cao, tức là mô hình bỏ sót ít các điểm có nhãn Positive. Recall = 1 có nghĩa là mô hình đã nhận diện được toàn bộ các điểm thật sự Positive.
- VD minh họa: Khi một người nghi ngờ mắc bệnh và đi kiểm tra, ta có 2 tình huống: mắc bệnh hoặc không. Kết quả xét nghiệm cũng có thể dương tính hoặc âm tính.
  - Precision: Tỷ lệ người thật sự mắc bệnh trên tổng số người được chẩn đoán là dương tính. Ví dụ, nếu precision = 0.9, thì cứ 100 người được chẩn đoán là dương tính sẽ có 90 người thật sự mắc bệnh.
  - Recall: Tỷ lệ người được chẩn đoán là mắc bệnh trên tổng số người thật sự mắc bệnh. Nếu recall = 0.9, thì trong 100 người thật sự mắc bệnh, mô hình sẽ chẩn đoán đúng 90 người.

#### F1-score

- Chỉ dựa vào Precision hoặc Recall riêng lẻ thì chưa đủ đánh giá độ tốt của mô hình.
- F1-score là trung bình điều hòa của Precision và Recall. Nó giúp đánh giá cân bằng giữa hai chỉ số này, đặc biệt trong các bài toán mất cân bằng dữ liệu.
- Precision = số điểm Positive dự đoán đúng / tổng số điểm dự đoán là Positive
- Recall = số điểm Positive dự đoán đúng / tổng số điểm thực sự là Positive
- Hệ số xác định  $R^2$  (R-squared) là chỉ số đánh giá mức độ mô hình giải thích được sự biến thiên của dữ liệu. Nó cho biết phần trăm phương sai trong dữ liệu đầu ra có thể được giải thích bằng mô hình.

Ta có công thức:

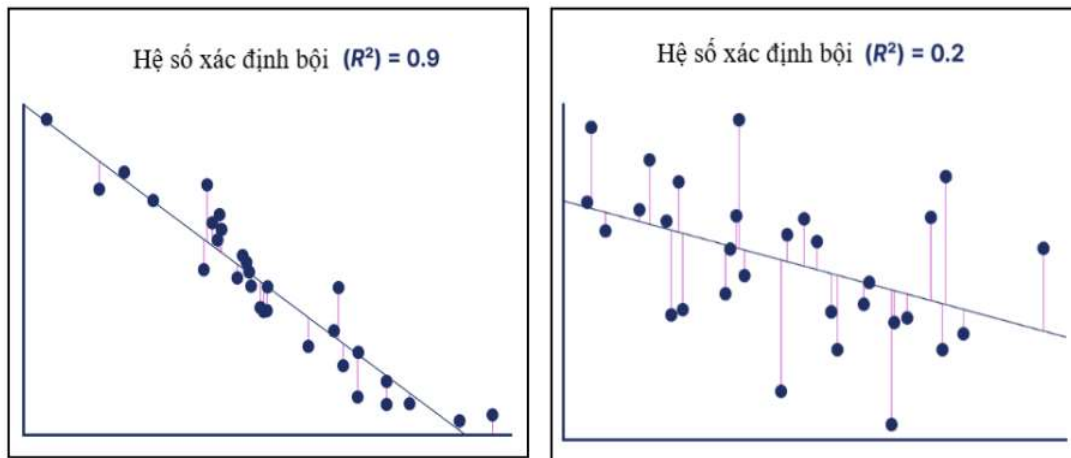
$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Trong đó  $\hat{y}_i$  là giá trị dự đoán,  $y_i$  là giá trị thực.  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$

$$\text{và } \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \epsilon_i^2$$

Giá trị của  $R^2$  dao động trong khoảng từ 0 đến 1:

- Nếu  $R^2$  càng tiến về 0 (từ 0,5 đến 0), mô hình dự đoán chưa phải là mô hình tốt, khả năng dự đoán chính xác của mô hình tương đối thấp.
- Nếu  $R^2$  nằm trong khoảng từ 0,5 đến gần 1, mô hình được xem là một mô hình dự đoán tốt;  $R^2$  càng gần 1 thì khả năng dự đoán chính xác càng cao.
- $R^2$  bằng 1 là điều gần như không thể xảy ra trong thực tế, vì luôn tồn tại sai số (phần dư) trong mô hình.



Hình *Error! No text of specified style in document.* Minh họa phân bố dữ liệu khi  $R^2$  gần phía 1 (bên trái) và  $R^2$  gần phía 0 (bên phải)

**Lỗi bình phương trung bình (Mean Square Error - MSE)** là một trong những metric phổ biến nhất trong các bài toán hồi quy. MSE tính trung bình của bình phương sai số giữa giá trị thực tế và giá trị dự đoán:

$$\mathbf{MSE} = \frac{1}{N} \sum_{i=1}^N (y_{test} - y_{pre})^2 \quad (2.18)$$

Trong đó:

- $y_{test}$  : giá trị thực tế
- $y_{pre}$  : giá trị dự đoán



**Sai số tuyệt đối trung bình (Mean Absolute Error - MAE)** là một metric đánh giá mô hình bằng cách tính trung bình của giá trị tuyệt đối sai số giữa giá trị thực tế và giá trị dự đoán:

$$\mathbf{MAE} = \frac{1}{N} \sum_{i=1}^N |y_{test} - y_{pre}| \quad (2.19)$$

Trong đó:

- $y_{test}$  : giá trị thực tế
- $y_{pre}$  : giá trị dự đoán

**Lỗi trung bình bình phương gốc (Root Mean Square Error - RMSE)** là độ lệch chuẩn của phần dư (lỗi dự đoán). RMSE đánh giá mức độ hiệu quả của mô hình bằng cách đo sự khác biệt giữa giá trị dự đoán và giá trị thực tế. RMSE càng nhỏ thì độ tin cậy của mô hình càng cao:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_{test} - y_{pre})^2}{n}} \quad (2.20)$$

Trong đó:

- $y_{test}$  : giá trị thực tế
- $y_{pre}$  : giá trị dự đoán

Trong bài toán này, em sẽ sử dụng ba phương pháp đánh giá mô hình là Precision, Recall và F1-score. Qua nhiều lần huấn luyện, thuật toán sẽ lựa chọn ra mô hình mà cả ba độ đo đều đạt kết quả tốt nhất.

## CHƯƠNG 3 ỨNG DỤNG THUẬT TOÁN XÂY DỰNG MÔ HÌNH

### 3.1 Mô tả bài toán

#### 3.1.1 Phân tích chi tiết bài toán

Việc làm là một vấn đề khiến mọi sinh viên đều trăn trở và lo lắng, đặc biệt là đối với sinh viên năm cuối sắp tốt nghiệp ra trường.

Hiện nay, các công ty và doanh nghiệp đều có những yêu cầu nhất định đối với nguồn nhân lực trẻ, điều này khiến nhiều sinh viên băn khoăn không biết liệu mình có đủ khả năng để được tuyển dụng hay không.

Sau quá trình nghiên cứu và phân tích một số nền tảng, công cụ về học máy, bài báo cáo này quyết định sử dụng thuật toán Cây quyết định Iterative Dichotomiser 3 (ID3) để xây dựng mô hình dự đoán cơ hội việc làm cho sinh viên sau tốt nghiệp.

Bên cạnh việc xây dựng mô hình dự đoán, báo cáo còn hướng tới việc triển khai kết quả dự đoán trên một giao diện người dùng có thể tương tác.

Từ đó, yêu cầu chung của bài toán được xác định như sau:

- Giao diện phải có đầy đủ tính năng của một hệ thống dự đoán cơ hội việc làm
- Người dùng có thể nhập hoặc chọn trực tiếp dữ liệu đầu vào trên giao diện

- Kết quả dự đoán của mô hình được hiển thị trực tiếp trên giao diện với độ chính xác cao nhất
- Giao diện đảm bảo tính thân thiện, dễ sử dụng và dễ hiểu
- Hệ thống đưa ra kết quả dự đoán với độ chính xác cao

### **3.2.1 Môi trường thực hiện**

- Hệ điều hành Windows 10 pro 64bit
- Python phiên bản 3.10.11
- Framework PyQt

### **3.2.2 Dữ liệu đầu vào**

Dữ liệu đầu vào được thu thập từ Kaggle — một trang web cộng đồng toàn cầu dành cho những người quan tâm đến trí tuệ nhân tạo và khoa học dữ liệu. Bộ dữ liệu được ghi nhận trong các năm 2013 và 2014, với tổng cộng 2.000 bản ghi( [data](#)).

Sau khi thu thập dữ liệu, tiến hành chuẩn hoá dữ liệu bằng thư viện `category_encoders` nhằm chuyển toàn bộ dữ liệu từ dạng chữ (categorical) sang dạng số. Tiếp theo, dữ liệu được chia theo phương pháp K-Fold, với mỗi lần lặp gồm K-1 phần để huấn luyện (train) và 1 phần để kiểm tra (test), nhằm tìm ra cấu hình mô hình học máy cho kết quả tốt nhất.

Như vậy, trong đề án này, em sử dụng tập dữ liệu đầu vào được chia để huấn luyện và kiểm tra mô hình. Dựa trên các tham số đánh giá mô hình, em lựa chọn ra tập dữ liệu tối ưu nhất và mô hình phù hợp nhất cho bài toán dự đoán cơ hội việc làm.

### **3.2.3 Xây dựng mô hình dự đoán**

Xây dựng mô hình dự đoán cơ hội việc làm dành cho sinh viên sau tốt nghiệp với các thông số sau:

- Mô hình sử dụng thuật toán Cây quyết định Iterative Dichotomiser 3 (ID3), được triển khai thông qua thư viện `DecisionTreeClassifier()` với các tham số đầu vào mặc định.

- Tập dữ liệu đầu vào được đưa vào mô hình dự đoán sử dụng thuật toán cây quyết định ID3.

Cấu trúc dữ liệu đầu vào:

Tập dữ liệu bao gồm 7 đặc trưng (cột X) đại diện cho thông tin của sinh viên, cụ thể:

- Age: Độ tuổi
- Gender: Giới tính
- Stream: Chuyên ngành học
- Internships: Số lượng kỳ thực tập đã tham gia
- CGPA: Điểm trung bình tích lũy
- Hostel: Có ở ký túc xá hay không
- HistoryOfBacklogs: Lịch sử nợ môn

Biến mục tiêu (cột Y) gồm hai giá trị: “YES” hoặc “NO”, tương ứng với việc có hoặc không có cơ hội nhận được việc làm.

Phương pháp đánh giá mô hình:

- Dữ liệu được chia thành tập huấn luyện và kiểm tra (train/test) theo tỷ lệ K-Fold, với K là số lần phân chia do người dùng nhập.
- Ở mỗi lần lặp, 1 phần dữ liệu sẽ được sử dụng để kiểm tra (test), trong khi K-1 phần còn lại được sử dụng để huấn luyện (train).
- Quá trình này nhằm đánh giá độ chính xác của mô hình qua từng lần huấn luyện và từ đó lựa chọn ra mô hình có hiệu suất tốt nhất

### 3.3 Xây dựng giao diện hiển thị kết quả

Với mô hình sử dụng thuật toán cây quyết định Iterative Dichotomiser 3 (ID3), em sử dụng toàn bộ tập dữ liệu tương ứng với lần huấn luyện có tỷ lệ dự đoán chính xác cao nhất làm tập huấn luyện cuối cùng cho mô hình.

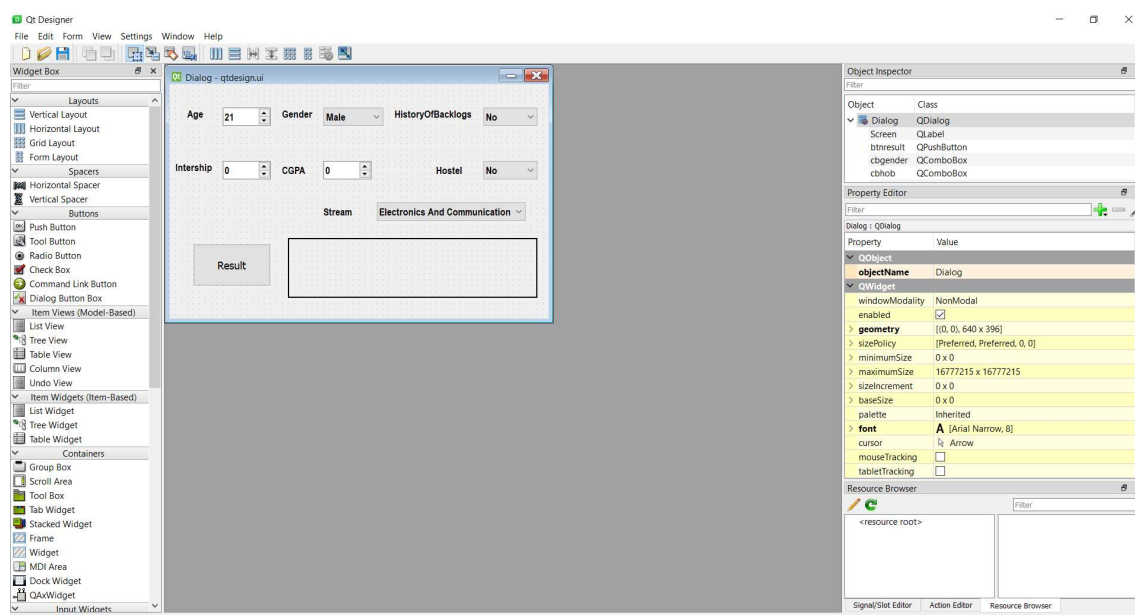
Mô hình sau khi được huấn luyện trong trường hợp này và cho kết quả tốt nhất sẽ được lưu lại, và được sử dụng làm mô hình chính để đưa ra dự đoán khi người dùng tương tác với giao diện.

Yêu cầu cơ bản đối với giao diện hiển thị kết quả dự đoán:

- Có các trường nhập liệu cho phép người dùng nhập hoặc chọn thông tin dữ liệu đầu vào.
- Sử dụng mô hình đã được huấn luyện với độ chính xác cao nhất để đưa ra kết quả dự đoán.
- Giao diện phải đảm bảo thân thiện, trực quan và dễ sử dụng cho người dùng.

Cấu trúc giao diện bao gồm hai phần chính:

- Phần nhập dữ liệu
  - Cho phép người dùng nhập hoặc lựa chọn các giá trị tương ứng với từng đặc trưng (Age, Gender, Stream, Internships, CGPA, Hostel, HistoryOfBacklogs).
  - Dữ liệu này sẽ được truyền vào mô hình để thực hiện dự đoán.
- Phần hiển thị kết quả:
  - Hiển thị kết quả dự đoán về cơ hội việc làm (có hoặc không) dựa trên dữ liệu mà người dùng đã cung cấp.



*Hình Error! No text of specified style in document..23 Thiết kế giao diện với Qt Designer*

Giao diện người dùng được xây dựng bao gồm 9 thành phần chính, trong đó có 7 thành phần dùng để nhập dữ liệu đầu vào, cụ thể như sau:

- Ba ô nhập liệu dạng Spin Box:
  - Age – Nhập độ tuổi của sinh viên
  - Internships – Nhập số kỳ thực tập mà sinh viên đã tham gia
  - CGPA – Nhập điểm trung bình tích lũy của sinh viên (theo thang 10)
- Bốn ô chọn dữ liệu dạng Combo Box:
  - Gender – Chọn giới tính của sinh viên (Male, Female)
  - Stream – Chọn chuyên ngành học
  - Hostel – Chọn thông tin sinh viên có ở ký túc xá hay không (Yes/No)
  - HistoryOfBacklogs – Chọn thông tin về lịch sử nợ môn (Yes/No)
- Ngoài ra, giao diện còn bao gồm:
  - Nút bấm "Dự đoán" (Predict Button): Khi người dùng nhập xong dữ liệu và nhấn nút này, hệ thống sẽ xử lý và sử dụng mô hình đã huấn luyện để đưa ra kết quả dự đoán.
  - Ô hiển thị kết quả dự đoán (Text Display): Kết quả sẽ hiển thị thông tin sinh viên có hoặc không có khả năng nhận được cơ hội việc làm, dựa trên dữ liệu đầu vào và mô hình đã được huấn luyện.

## CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

### 4.1 Kết quả đánh giá mô hình

#### 4.1.1 Kết quả mô hình

Đánh giá mô hình học máy với  $K = 7$  (K-Fold Cross Validation)

Ở trường hợp  $K = 7$ , quá trình học máy sẽ được lặp lại 7 lần, mỗi lần sử dụng 1 tập làm dữ liệu kiểm tra (test) và 6 tập còn lại làm dữ liệu huấn luyện (train). Mô hình được đánh giá thông qua ba độ đo phổ biến: Precision, Recall, F1-score, cùng với tỷ lệ dự đoán chính xác trong từng lần huấn luyện.

Lần máy học thứ 1

Độ đo	Kết quả
Precision	0.8074
Recall	0.8015
F1-score	0.7967
Độ chính xác	79.72%

=> Precision và Recall có chênh lệch nhỏ, F1-score thấp hơn đôi chút. Tỷ lệ chính xác đạt 79.72%.

Lần máy học thứ 2

Độ đo	Kết quả
Precision	0.8407
Recall	0.8563
F1-score	0.8376
Độ chính xác	83.92%

=> Recall cao nhất, F1-score thấp nhất trong ba độ đo. Tỷ lệ chính xác khá cao.

Lần máy học thứ 3

Độ đo	Kết quả
Precision	0.8461
Recall	0.8227
F1-score	0.8284
Độ chính xác	83.57%

=> Precision cao hơn rõ rệt so với Recall và F1-score. Tỷ lệ chính xác ổn định.

Lần máy học thứ 4

Độ đo	Kết quả
Precision	0.8329
Recall	0.8029
F1-score	0.7965
Độ chính xác	80.07%

=> Các độ đo giảm nhẹ so với lần thứ 3, F1-score tiếp tục là giá trị thấp nhất.

Lần máy học thứ 5

Độ đo	Kết quả
Precision	0.8942
Recall	0.9056
F1-score	0.8994
Độ chính xác	90.88%

=> Đây là lần huấn luyện có kết quả tốt nhất: cả ba độ đo đều cao, tỷ lệ chính xác gần 91%.



#### Lần máy học thứ 6

Độ đo	Kết quả
Precision	0.8036
Recall	0.8050
F1-score	0.8033
Độ chính xác	80.35%

=> Các giá trị khá đồng đều nhưng thấp hơn rõ rệt so với lần thứ 5.

#### Lần máy học thứ 7

Độ đo	Kết quả
Precision	0.8468
Recall	0.8474
F1-score	0.8456
Độ chính xác	84.56%

=> Cả ba độ đo đều gần tương đương nhau. Tỷ lệ chính xác ở mức cao và ổn định.

#### Tổng kết

Qua 7 lần huấn luyện, mô hình đều cho kết quả tương đối tốt, với tỷ lệ chính xác dao động từ 79.72% đến 90.88%.

- Precision thấp nhất: lần 6 (0.8036), cao nhất: lần 5 (0.8942)
- Recall thấp nhất: lần 1 (0.8015), cao nhất: lần 5 (0.9056)
- F1-score thấp nhất: lần 1 (0.7967), cao nhất: lần 5 (0.8994)

**Kết luận:** Mô hình đạt hiệu suất tốt nhất ở lần máy học thứ 5, với tỷ lệ dự đoán chính xác 90.88%. Đây là mô hình được lựa chọn để lưu lại và sử dụng cho hệ thống dự đoán chính thức.

#### 4.1.2 Đánh giá và lựa chọn mô hình

Sau khi thực hiện quá trình huấn luyện và đánh giá mô hình qua 7 lần ( $K = 7$ ), kết quả các độ đo được tổng hợp trong bảng sau:

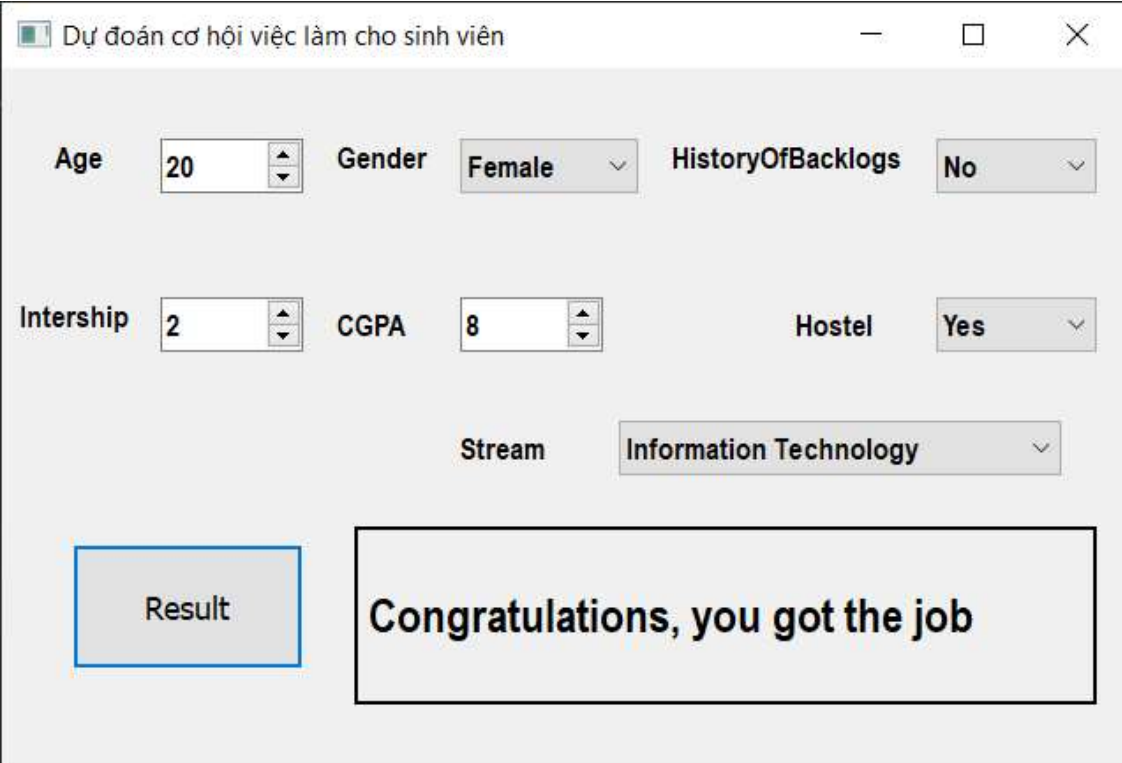
Lần học	Precision	Recall	F1-score	Tỷ lệ dự đoán chính xác
Lần 1	0.8074	0.8015	0.7967	79.72%
Lần 2	0.8406	0.8563	0.8376	83.92%
Lần 3	0.8460	0.8227	0.8283	83.57%
Lần 4	0.8329	0.8028	0.7965	80.07%
Lần 5	0.8942	0.9055	0.8993	90.88%
Lần 6	0.8036	0.8049	0.8033	80.35%
Lần 7	0.8467	0.8474	0.8455	84.56%

*Bảng tổng hợp các độ đo và tỷ lệ dự đoán chính xác qua từng lần huấn luyện*

Từ bảng tổng hợp cho thấy, ở lần máy học thứ 5, cả ba độ đo Precision, Recall và F1-score đều đạt giá trị cao nhất, cùng với tỷ lệ dự đoán chính xác đạt 90.88% – cao nhất trong tất cả các lần.

Do đó, trong bài toán dự đoán cơ hội việc làm cho sinh viên năm cuối, em quyết định lựa chọn mô hình huấn luyện ở lần thứ 5 làm mô hình chính để sử dụng trong quá trình dự đoán trên giao diện người dùng.

## 4.2 Demo giao diện hiển thị



The screenshot shows a Windows application window titled "Dự đoán cơ hội việc làm cho sinh viên". The interface contains several input fields and a result display area. The inputs are: Age (20), Gender (Female), HistoryOfBacklogs (No), Internship (2), CGPA (8), Hostel (Yes), and Stream (Information Technology). A "Result" button is located on the left, and a large box on the right displays the message "Congratulations, you got the job".

Field	Value
Age	20
Gender	Female
HistoryOfBacklogs	No
Internship	2
CGPA	8
Hostel	Yes
Stream	Information Technology

**Result**

**Congratulations, you got the job**

Hình *Error! No text of specified style in document.*24 Demo giao diện hiển thị

## KẾT LUẬN

Sau quá trình thực hiện đồ án, em đã tích lũy thêm nhiều kiến thức mới mà trước đây còn thiếu sót, đồng thời học hỏi được nhiều kinh nghiệm thực tiễn trong quá trình nghiên cứu và xây dựng sản phẩm. Dù còn hạn chế về thời gian và kiến thức, em đã nỗ lực hoàn thành các mục tiêu đề ra và tối ưu hóa sản phẩm trong khả năng của bản thân.

### ❖ Nội dung đã đạt được

- ✓ Nghiên cứu và tìm hiểu bài toán dự đoán cơ hội việc làm cho sinh viên năm cuối, ứng dụng học máy để xây dựng mô hình dự đoán.
- ✓ Phân tích và làm rõ nguyên lý hoạt động của mô hình Cây quyết định.
- ✓ Tìm hiểu và ứng dụng thuật toán ID3 vào bài toán, lựa chọn mô hình cho kết quả tối ưu.
- ✓ Xây dựng giao diện tương tác người dùng sử dụng nền tảng Qt Designer, hiển thị kết quả dự đoán trực tiếp từ mô hình.

### ❖ Hướng phát triển

- Nghiên cứu thêm các thuật toán khác trong học máy như: Random Forest, SVM, KNN,... để mở rộng và so sánh hiệu quả mô hình.
- Mở rộng khả năng dự đoán không chỉ dành cho sinh viên sau tốt nghiệp mà cho tất cả sinh viên đang theo học.
- Cải thiện giao diện người dùng, bổ sung thêm tính năng như lưu kết quả, thống kê lịch sử dự đoán hoặc đề xuất cải thiện hồ sơ cá nhân dựa trên kết quả đầu ra.

### ❖ Hạn chế

Do hạn chế về kinh nghiệm và thời gian, đồ án mới chỉ dừng lại ở mức nghiên cứu và ứng dụng cơ bản của một thuật toán học máy vào bài toán thực tế.

Việc chuẩn bị dữ liệu, tiền xử lý và đánh giá mô hình vẫn còn nhiều điểm cần cải thiện để nâng cao độ tin cậy và khả năng tổng quát của mô hình.

Em rất mong nhận được những góp ý, phản hồi từ thầy cô để hoàn thiện hơn trong các dự án tiếp theo.

## DANH MỤC TÀI LIỆU THAM KHẢO

<https://machinelearningcoban.com/>

Slide bài giảng học máy – <https://sites.google.com/a/wru.vn/cse445fall2016/lecture-materials>

Slide bài giảng khai phá dữ liệu – <https://sites.google.com/site/tlucse404/>

Engineering Placements Prediction Dataset –

<https://www.kaggle.com/tejashvi14/engineering-placements-prediction>

Giới thiệu thuật toán cây quyết định ID3 – <https://1upnote.me/post/2018/10/ds-ml-decision-tree-id3/>

Stack Overflow – <https://stackoverflow.com/>

Tài liệu sử dụng Qt Designer – <https://doc.qt.io/qt-6/qtdesigner-manual.html>

Phương pháp đánh giá độ tin cậy mô hình học máy – <https://rabiloo.com/vi/blog/cac-phuong-phap-danh-gia-mo-hinh-machine-learning-va-deep-learning>

