BỘ GIÁO DỤC VÀ ĐÀO TẠO **BỘ NÔNG NGHIỆP VÀ PTNT**

TRƯỜNG ĐẠI HỌC THỦY LỢI

----------***----------

PHẠM MINH TIẾN

DỰ ĐOÁN CƠ HỘI VIỆC LÀM DÀNH CHO SINH VIÊN NĂM CUỐI

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, THÁNG 1 NĂM 2023

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

PHẠM MINH TIẾN

DỰ ĐOÁN CƠ HỘI VIỆC LÀM DÀNH CHO SINH VIÊN NĂM CUỐI

Ngành : Công nghệ thông tin
Mã số: 7480201

NGƯỜI HƯỚNG DẪN :
ThS. Vũ Anh Dũng

HÀ NỘI, NĂM 2023

GÁY BÌA ĐỒ ÁN TỐT NGHIỆP, KHÓA LUẬN TỐT NGHIỆP

HỌ VÀ TÊN PHẠM MINH TIẾN

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2023

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

---------★---------

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Phạm Minh Tiến

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 dành cho sinh viên năm cuối
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	15%
Chương 2: Học máy và một số thuật toán trong học máy	20%
Chương 3: Ứng dụng thuật toán xây dựng mô hình	35%
Chương 4: Kết quả và đánh giá mô hình	30%

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

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

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày 22 tháng 9 năm 2022

Trưởng Bộ môn	**Giáo viên hướng dẫn chính**
(Ký và ghi rõ Họ tên)	*(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

Tiến

Phạm Minh Tiến

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

Chữ ký

Phạm Minh Tiến

LỜI NÓI ĐẦU

Big Data, Internet of Things, AI, Data mining … là những thuật ngữ đã quá quen thuộc trong thời đại công nghệ 4.0 hiện nay. Một thời đại của công nghệ, thời đại tập trung phát triển những gì liên quan đến hệ thống vật lý không gian mạng Internet. Mang đến cho con người nhiều lợi thế trong việc xây dựng một xã hội văn minh, giúp các doanh nghiệp dễ dàng hợp tác và chia sẻ dữ liệu với khách hàng, nhà sản xuất, nhà cung cấp và các bên khác trong chuỗi cung ứng. Năng suất được cải thiện và khả năng cạnh tranh cũng được nâng cao, giúp cá nhân hoặc doanh nghiệp tiến đến với công nghệ kỹ thuật từ đó có cơ hội để đạt được công việc mong muốn. Từ đó cải thiện vấn đề việc làm nâng cao chất lượng cuộc sống và nâng cao mức thu nhập cho con người đặc biệt là các sinh viên Đại học năm cuối hoặc vừa ra trường mong muốn bản thân có việc làm ổn định. Việc ứng dụng công nghệ vào quá trình học tập và làm việc giúp cho các sinh viên có lựa chọn đúng đắn dựa vào thành tích học tập cũng như ngành học của mình

Song song với phần lớn người có việc làm ổn định ở những người có kinh nghiệm, một bộ phận không nhỏ vẫn đang băn khoăn lo lắng về vấn đề việc làm đặc biệt là những sinh viên năm cuối sắp tốt nghiệp ra trường. Sự lo lắng về vấn đề việc làm của các sinh viên không chỉ dừng lại ở chính bản thân họ mà cả người thân hay các công ty doanh nghiệp cũng mong muốn mình có nguồn nhân sự đầu vào tốt. Nhu cầu về nhân lực và chất lượng của các công ty doanh nghiệp cũng ngày một tăng lên đòi hỏi chất lượng nhân sự cũng phải đáp ứng thể hiện thông qua kết quả của quá trình học tập là điều rõ ràng nhất. Điều này không chỉ là vấn đề xuất hiện tại Việt Nam mà tất cả các nước trên thế giới đều có.

Hàng năm tỷ lệ sinh viên năm cuối tốt nghiệp chuẩn bị ra trường ngày một nhiều. Tỷ lệ nhu cầu đào thải của các công ty doanh nghiệp cũng không hề nhỏ hay đơn giản là sự cạnh tranh giữa các ứng viên cũng không hề nhỏ. Các lĩnh vực, ngành luôn yêu cầu ứng viên cũng phải có năng lực kết quả học tập trên trường tốt. Đặc biệt, là các ngành nghề hot và nổi bật là nơi yêu cầu đối với chất lượng đầu vào gắt gao nhất. Kết quả học tập và những yếu tố bên lề sẽ là điều quan trọng để ứng viên có niềm tin về việc bản thân sẽ được nhận vào công ty doanh nghiệp làm việc

Do vậy, ứng dụng công nghệ thông tin vào việc dự đoán cơ hội việc làm dành cho các sinh viên năm cuối là điều thiết thực sẽ giúp các ứng viên tự tin và phấn đấu hơn với kết quả của bản thân trong quá trình học tập

Theo hướng nghiên cứu thuật toán ID3(Iterative Dichotomiser 3) trong học máy để xử lý dữ liệu từ đó để đưa ra một mô hình dự đoán khả năng "có" hoặc "không" về bài toán dự đoán cơ hội việc làm của sinh viên năm cuối, cụ thể ở báo cáo này hướng tới đề tài "***Dự đoán cơ hội việc làm cho các sinh viên năm cuối***"

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

- Chương 1: Tổng quan
  - Trình bày tổng quan về bài toán, đưa ra vấn đề của bài toán
  - Trình bày đối tượng nghiên cứu trong bài toán và hướng phát triển của bài toán
- Chương 2: Học máy và một số thuật toán trong học máy
  - Trình bày lý thuyết chung về học máy, phân loại các thuật toán trong học máy
  - Trình bày lý thuyết của các thuật toán được sử dụng trong bài toán: Thuật toán cây quyết định ID3(Iterative Dichotomiser 3)
  - Trình bày chi tiết các công cụ được sử dụng:
    - Ngôn ngữ lập trình: Python
    - Xây dựng giao diện hiển thị kết quả dự đoán: Dựa trên công cụ Qt Designer
    - Môi trường chạy: PyCharm
    - Một số thư viện được sử dụng trong bài toán: Scikit-learning, numpy, pandas, matplotlib…
- Chương 3: Ứng dụng thuật toán xây dựng mô hình
  - Trình bày chi tiết bài toán, quy trình thực hiện bài toán
  - Dữ liệu đưa vào xây dựng mô hình đưa ra kết quả dự đoán

- Xây dựng giao diện dự đoán và hiển thị kết quả với Qt designer

- Chương 4: Kết quả và đánh giá mô hình
  - Kết quả thực nghiệm và đưa ra đánh giá mô hình

LỜI CẢM ƠN

Lời đầu tiên em xin được bày tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Thủy Lợi và Ban chủ nhiệm khoa Công nghệ Thông tin đã tận tình giúp đỡ em trong suốt thời gian học tại trường.

Trong suốt thời gian bốn năm học tập và rèn luyện tại Trường Đại học Thủy Lợi cho đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của thầy cô và bạn bè. Với lòng biết ơn sâu sắc và chân thành nhất, em xin gửi đến thầy cô ở Khoa Công nghệ Thông tin – Trường Đại học Thủy Lợi đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Qua bài báo cáo với đề tài ***"Dự đoán cơ hội việc làm dành cho sinh viên năm cuối"*** này, em xin chân thành cảm ơn ThS Vũ Anh Dũng đã tận tâm hướng dẫn em qua từng buổi nói chuyện, thảo luận về lĩnh vực học máy.

Em cũng xin bày tỏ lòng biết ơn đến ban lãnh đạo của Trường Đại học Thủy Lợi và các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

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)

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 dang tài nguyên thống nhất (Uniform Resource Locator)

CHƯƠNG 1 TỔNG QUAN

1.1 Đặt vấn đề

Nền kinh tế của một đất nước muốn phát triển thì bên cạnh những mặt thuận lợi thì cũng có không ít nhưng khó khăn, tiêu cực phát sinh có thể kể đến đó là tình trạng thất nghiệp của sinh viên sau khi ra trường ngày càng tăng trong cơ chế thị trường hiện nay.

Đất nước càng phát triển bao nhiêu ngoài những công nghệ hiện đại phục vụ cho kinh doanh sản xuất thì một trong những yếu tố quyết định sự phát triển của đất nước là lực lượng lao động, trong nền kinh tế thị trường ngày nay lực lượng lao động là những sinh viên được đào tạo trong các trường đại học, cao đẳng…, là lực lượng trẻ của đất nước rất năng động và có năng lực trong công việc. Chính vì vậy, sinh viên là nguồn nhân lực rất quan trọng chúng ta cần biết cách sử dụng một cách hợp lý và hiệu quả nhất. Nhưng tình trạng thất nghiệp của sinh viên sau khi ra trường hiện nay đã ảnh hưởng rất nhiều đến sự phát triển kinh tế, xã hội của đất nước. Vậy câu hỏi đặt ra của những nhà quản lý, của đất nước là nguyên nhân vì sao dẫn tới tình trạng đó? Phải chăng là do quá trình đào tạo của các trường đại học còn nhiều mặt chưa được; hay do những chính sách của Nhà nước chưa hợp lý trong việc sử dụng lao động.

Qua tìm hiểu thì thấy nổi lên một số nguyên nhân đó là, tình hình hoạt động sản xuất, kinh doanh của các doanh nghiệp có nhiều khó khăn, nhu cầu tuyển dụng lao động bị thu hẹp; Đối với các cơ quan, tổ chức nhà nước, nhu cầu tuyển dụng công chức, viên chức ngày càng có yêu cầu cao hơn về chất lượng; Bên cạnh đó Việc có thêm nhiều cơ sở đào tạo (trường cao đẳng, đại học) ra đời dẫn đến số lượng sinh viên được đào tạo ở cùng các ngành, chuyên ngành ngày càng nhiều, cung vượt cầu. Thực tế cho thấy, số lượng ứng viên đăng ký dự tuyển hàng năm để tìm việc làm khá đông, song kết quả số người đáp ứng được yêu cầu của nhà tuyển dụng còn rất hạn chế. Trước những thách thức nêu trên, đòi hỏi các cơ sở đào tạo và bản thân người học (sinh viên) phải có cách nhìn nhận mới về vấn đề việc làm sau khi tốt nghiệp ra trường.

Nhận thức sâu sắc về việc đào tạo nguồn nhân lực đáp ứng với yêu cầu trong tình hình mới, trong những năm qua, Các trường học ở Việt Nam nói riêng và trên thế giới nói chung đã thực hiện nhiều giải pháp nhằm nâng cao chất lượng đào tạo, như: Cập nhật chương trình, giáo trình đào tạo theo hướng hiện đại, tích cực đổi mới phương pháp dạy học, đẩy mạnh ứng dụng công nghệ thông tin, chú trọng giáo dục cho sinh viên những kỹ năng mềm về giao tiếp, xử lý tình huống, ngoại ngữ, tin học, khuyến khích sinh viên tham gia nghiên cứu khoa học,... nâng cao chất lượng kết quả

học tập nhằm cho sinh viên năm cuối sau khi ra trường sẽ có một công việc tốt, ổn định và có thể dự đoán được dựa trên các tiêu chí ngay từ khi còn đang học trên trường

Vì vậy, đồ án tốt nghiệp của em thực hiện "Dự đoán cơ hội việc làm cho các sinh viên năm cuối" bằng việc dùng ngôn ngữ Python kết hợp với thuật toán cây quyết định ID3(Iterative Dichotomiser 3) trong học máy để đưa ra kết quả dự đoán khả năng về việc làm dành cho các sinh viên năm cuối và còn có thể nhân rộng ra toàn bộ đối với các sinh viên vẫn còn đang trong quá trình học tập

1.2 Mục tiêu đề tài

Mục tiêu của đề tài là nghiên cứu thuật toán ID3(Iterative Dichotomiser 3) trong học máy ứng dụng cho bài toán dự đoán cơ hội viêc làm dành cho các sinh viên . Đặc biệt, nghiên cứu tập trung vào hai mục tiêu chính

- Bằng việc sử dụng ngôn ngữ lập trình Python và ứng dụng thuật toán ID3(Iterative Dichotomiser 3), xây dựng mô hình cây quyết định từ đó đưa ra dự đoán về khả năng "có" hoặc "không" về cơ hội việc làm cho sinh viên
- Bên cạnh đó, xây dựng giao diện để có thể nhập thông tin từ đó dự đoán cơ hội việc làm dành cho sinh viên

1.3 Đối tượng nghiên cứu

Dựa trên công bố của một trường đại học với tệp dữ liệu trên trang web kaggle (nền tảng trực tuyến cho cộng đồng Machine Learning (ML) và Khoa học dữ liệu. Sử dụng dữ liệu này để dự đoán và phân tích xem liệu một sinh viên có được nhận vào làm tại một công ty, doanh nghiệp nào đó hay không với cơ sở là dựa trên lý lịch của sinh viên đó.

Vì vậy, đề tài "Dự đoán cơ hội việc làm cho các sinh viên năm cuối" được thực hiện trong khuôn khổ đồ án tốt nghiệp để ứng dụng thuật toán ID3(Iterative Dichotomiser 3), xây dựng mô hình cây quyết định từ đó đưa ra dự đoán về khả năng "có" hoặc "không" về cơ hội việc làm cho sinh viên và từ đó có thể sử dụng chính mô hình xây dựng sau đó dùng dữ liệu của bất sinh viên nào để dự đoán về cơ hội việc làm của sinh viên đó

1.4 Phạm vi nghiên cứu đề tài

Việc nghiên cứu và triển khai bài toán dự đoán cơ hội việc làm dành cho sinh viên năm cuối được thực hiện trong khuôn khổ đồ án tốt nghiệp đưa ra nghiên cứu của các vấn đề cơ bản:

- Sử dụng một số thư viện mở có sẵn.

- Sử dụng mô hình thuật toán cây quyết định ID3(Iterative Dichotomiser 3) trong học máy
- Ứng dụng thuật toán cây quyết định từ đó đưa ra dự đoán “có” hoặc “không” về cơ hội nhận được việc làm của các sinh viên năm cuối
- Sử dụng một số tham số đánh giá mô hình để đưa ra kết quả dự đoán có độ chính xác lớn nhất
- Sử dụng Qt Designer để hiển thị kết quả dự đoán khi người dùng tương tác với giao diện

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

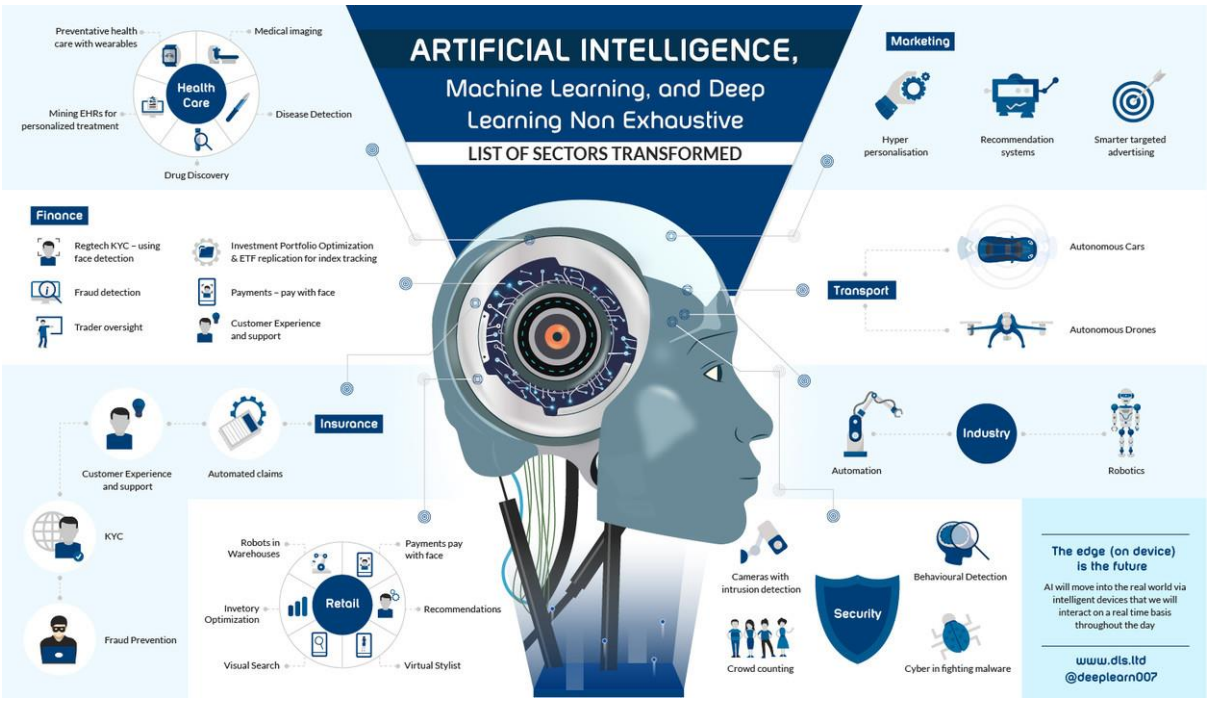
2.1 Học máy

2.1.1 Khái niệm liên quan học máy (Machine learning)

Trong những năm vừa qua, cùng với sự bùng nổ của cuộc cách mạng công nghiệp 4.0, các thuật ngữ như trí tuệ nhân tạo (AI), học máy (machine learning) và học sâu (deep learning) đang dần trở nên phổ biến và trở thành những khái niệm mà đã quá đỗi quen thuộc với chúng ta trong thời đại 4.0 này

2.1.1.1 Machine learning là gì?

Machine learning - một công nghệ phát triển từ lĩnh vực AI



Hình 2.1 Meachine learning

Học máy (meachine learning) là một nhánh nhỏ của trí tuệ nhân tạo 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. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được huấn luyện,

nghiên cứu cho phép máy tính dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học) để đưa ra phân loại hoặc dự đoán.

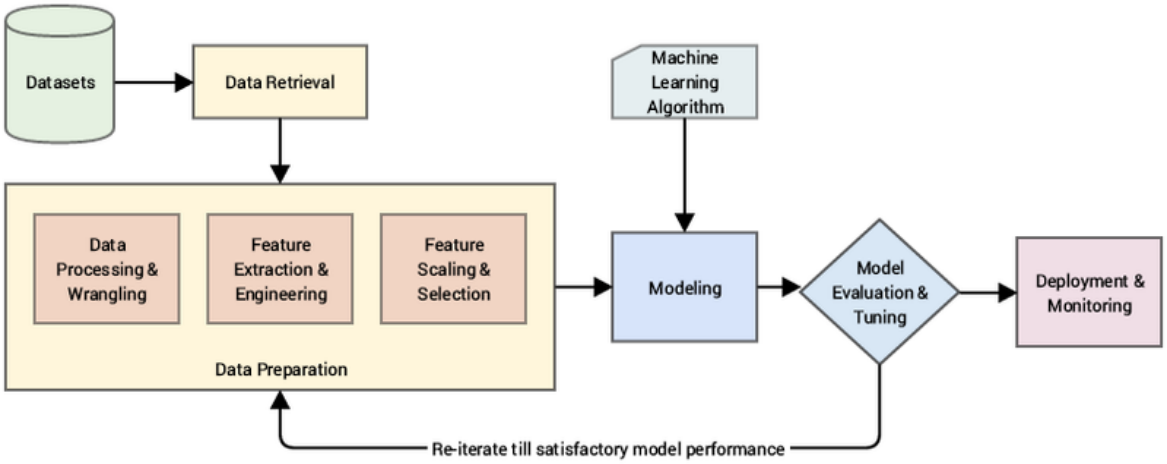
Thuật ngữ Học máy ngày càng được phổ biến trên toàn thế giới. Dữ liệu lớn (Big Data) đang ngày tăng trưởng mạnh mẽ kết hợp cùng các thuật toán Machine Learning đã cải thiện độ chính xác của những mô hình dự đoán tương lai. Từ đó, có thể tạo ra các mô hình có thể phân tích những dữ liệu lớn có tính phức tạp và đưa ra kết quả nhanh hơn, chính xác hơn ngay cả trên quy mô rất lớn. Bằng việc xây dựng các mô hình chính xác, tổ chức công ty hay cá nhân sẽ có cơ hội tốt hơn trong việc xác định các cơ hội sinh lời hoặc tránh những rủi ro chưa biết.

Hầu hết các ngành công nghiệp làm việc với lượng lớn dữ liệu đã cần ứng dụng công nghệ Máy Học một cách nhanh chóng. Bằng cách thu thập thông tin chi tiết từ dữ liệu thường là trong thời gian thực, các tổ chức có thể làm việc hiệu quả hơn hoặc giành được lợi thế so với các đối thủ cạnh tranh.

2.1.1.2 Machine learning hoạt động như thế nào?

- ❖ *Quy trình hoạt động của machine learning*

Các thuật toán Machine Learning được hướng dẫn để sử dụng một bộ dữ liệu đào tạo, từ đó tạo ra một mô hình nguyên mẫu. Khi thuật toán này tiếp nhận dữ liệu mới, nó sẽ đưa ra những dự đoán phân tích dựa trên nguyên mẫu căn bản.



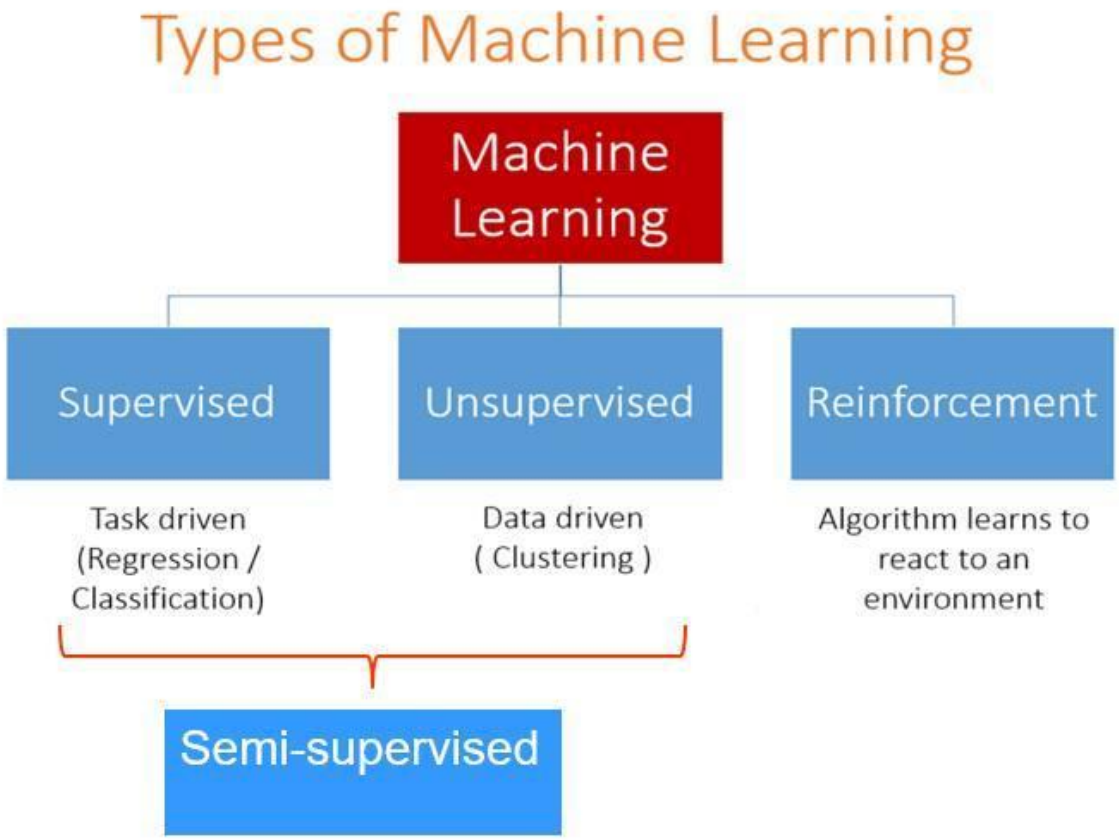
Hình 2.2 Một quy trình học máy chung

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

1. Thu thập dữ liệu: Dựa vào đặc trưng của bài toán, ta thu thập dữ liệu (hình ảnh, số liệu) từ những nguồn dữ liệu chính thống để mô hình có thể đưa ra kết qua đúng và đạt hiệu quả cao.
2. Chuẩn bị dữ liệu: Sau khi thu thập được dữ liệu, ta thực hiện một số thao tác để đưa dữ liệu ban đầu vào mô hình: chuẩn hóa dữ liệu, loại bỏ các thuộc tính không cần thiết, gán nhã, mã hóa, trích xuất đặc trưng, rút gọn dữ liệu.
3. Huấn luyện mô hình: Tùy theo đặc trưng bài toán, ta cần lựa chọn thuật toán sao cho phù hợp với yêu cầu bài toán và dữ liệu ta thu thập được. Dữ liệu sau khi được xử lý sẽ được chia ra làm tập huấn luyện (Training set) và tập test rồi đưa vào mô hình để mô hình học trên dữ liệu đã thu thập được (dữ liệu từ quá khứ), từ đó đưa ra kết quả dự đoán
4. Đánh giá mô hình: sau khi mô hình được huấn luyện và đưa ra kết quả đoán, ta cần đánh giá mô hình học máy. Chúng ta cần dùng các độ đo để đánh giá mô hình, tùy vào từng tham số đánh giá khác nhau mà cho ra các số liệu khác nhau. Độ chính xác của mô hình đặt trên 80% được cho là tốt.
5. Đào tạo lại mô hình: Trong trường hợp, các tham số đo độ chính xác của mô hình cho kết quả không mấy khả thi, ta có thể huấn luyện lặp lại để tăng độ chính xác cho mô hình
6. Áp dụng: Sau khi thu được kết quả dự đoán, ta có thể sử dụng nó để ứng dụng vào bài toán thực tiễn.

2.1.1.3 Một số phương pháp học máy phổ biến

Hai trong số các phương pháp học máy được áp dụng rộng rãi nhất là ***Học có giám sát*** (*Supervised learning*) và ***Học không giám sát*** (*Unsupervised learning*). Ngoài ra, machine learning còn có thể phân làm các loại như: Học bán giám sát *(Semi-supervised learning),* Học củng cố/tăng cường *(Reinforce learning).*



Hình 2.3 Minh họa phân loại học máy

- **Học có giám sát** (*Supervised learning*):

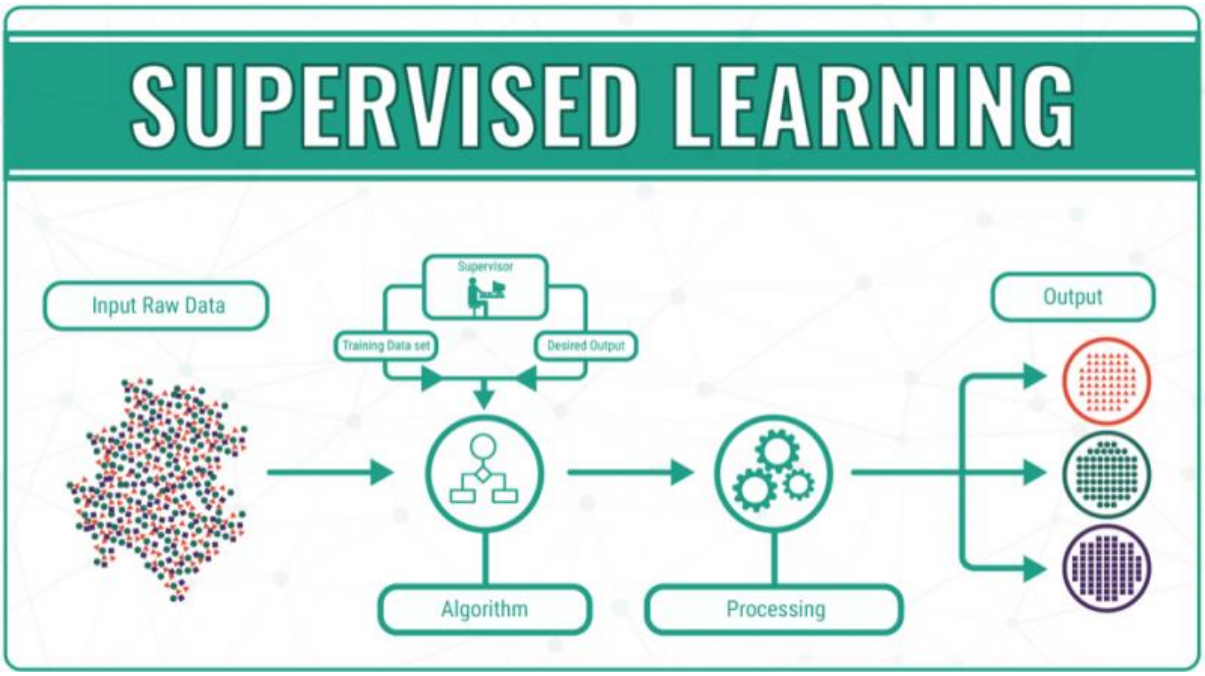
Học có giám sát là nhóm phổ biến nhất trong các thuật toán Machine Learning. Thuật toán được đào tạo dựa trên dữ liệu đầu vào đã được gán nhãn cho đầu ra cụ thể để suy luận ra quan hệ giữa đầu vào và đầu ra. Mô hình trải qua quá trình đào tạo và có thể đưa ra dự đoán cho các bộ dữ liệu mới.

Một số thuật toán phổ biến trong Học có giám sát bao gồm Máy vectơ hỗ trợ (SVM), Hồi quy logistic, Naive Bayes, Mạng noron, K – láng giềng gần nhất (KNN) và Rừng ngẫu nhiên.

Học máy có giám sát thường có tính ứng dụng cao trong một số bài toán như:

- Phân tích dự đoán (giá nhà, giá giao dịch chứng khoán, v.v.)
- Nhận dạng văn bản
- Phát hiện thư rác

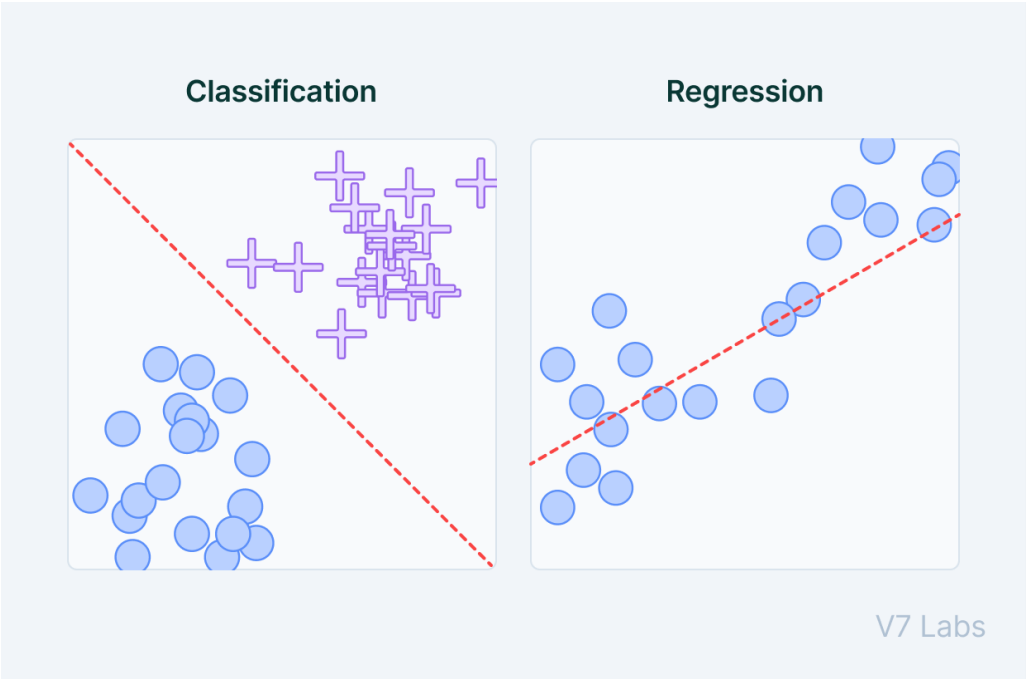
- Phát hiện đối tượng (khuôn mặt, chữ viết…)



Hình 2.4 Học có giám sát (Supervised learning) (Nguồn: V7 laps)

Học có giám sát còn tiếp tục được chia nhỏ thành hai loại:

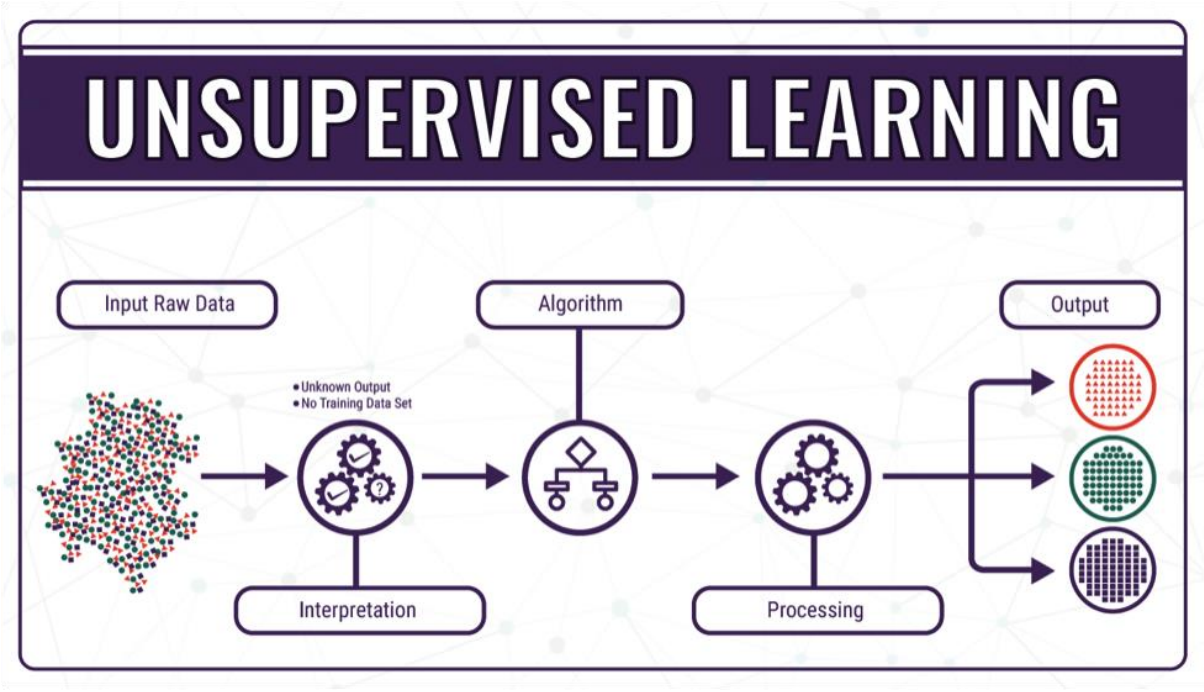
- Classification (Phân loại): Bài toán chia tập dữ liệu đầu vào thành hai hoặc nhiều phân loại
- Regression (Hồi quy): Bài toán sử dụng dữ liệu đầu vào có gán nhãn dự báo ra một giá trị thực cụ thể



Hình 2.5 Phân loại học có giám sát (Nguồn: V7Laps)

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

Học không giám sát sử dụng những dữ liệu chưa được gán nhãn sẵn. Thuật toán sẽ dựa vào cấu trúc của dữ liệu để suy luận và tìm cách thực hiện công việc nào đó ví dụ như phân nhóm hoặc giảm số chiều của dữ liệu.



Hình 2.6 Minh họa phương pháp học không giám sát

Học không giám sát cũng được chia nhỏ thành hai loại chính:

- Clustering (phân nhóm): Một bài toán phân nhóm toàn bộ dữ liệu thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm
- Association: Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước

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

Học bán giám sát được sử dụng cho các ứng dụng tương tự như Học có giám sát. Nhưng nó sử dụng cả dữ liệu được gắn nhãn và không được gắn nhãn để huấn luyện. Học bán giám sát rất hữu ích khi chi phí liên quan đến việc gắn nhãn quá cao để cho phép một quá trình huấn luyện được gắn nhãn đầy đủ.

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: Dạy thuật toán dịch ngôn ngữ dựa trên ít hơn một từ điển từ đầy đủ.
- Phát hiện gian lận: Xác định các trường hợp gian lận khi bạn chỉ có một vài ví dụ tích cực.
- Dán nhãn dữ liệu: Các thuật toán được đào tạo trên tập dữ liệu nhỏ có thể học cách áp dụng nhãn dữ liệu cho các tập lớn hơn một cách tự động.

- **Học tăng cường** (reinforcement learning)

Học tăng cường là một lĩnh vực của ML. Đó là việc thực hiện hành động phù hợp để tối đa hóa phần thưởng trong một tình huống cụ thể. Nó được sử dụng bởi các phần mềm và ML khác nhau để tìm ra hành vi hoặc đường dẫn tốt nhất có thể mà nó nên thực hiện trong một tình huống cụ thể. Học tăng cường khác với học có giám sát ở chỗ trong học có giám sát, dữ liệu huấn luyện có khóa trả lời với nó, do đó mô hình được huấn luyện với câu trả lời chính xác trong khi trong học củng cố, không có câu trả lời nhưng tác nhân củng cố quyết định phải làm gì để thực hiện nhiệm vụ đã cho. Trong trường hợp không có tập dữ liệu đào tạo, nó nhất định phải học hỏi kinh nghiệm của nó.

- Ưu điểm của việc học tăng cường là:
  + Tối đa hóa hiệu suất
  + Duy trì thay đổi trong một thời gian dài
- Nhược điểm của học tăng cường:
  + Quá nhiều gia cố có thể dẫn đến quá tải các trạng thái có thể làm giảm kết quả

2.1.1.4 Ứng dụng học máy vào thực tiễn

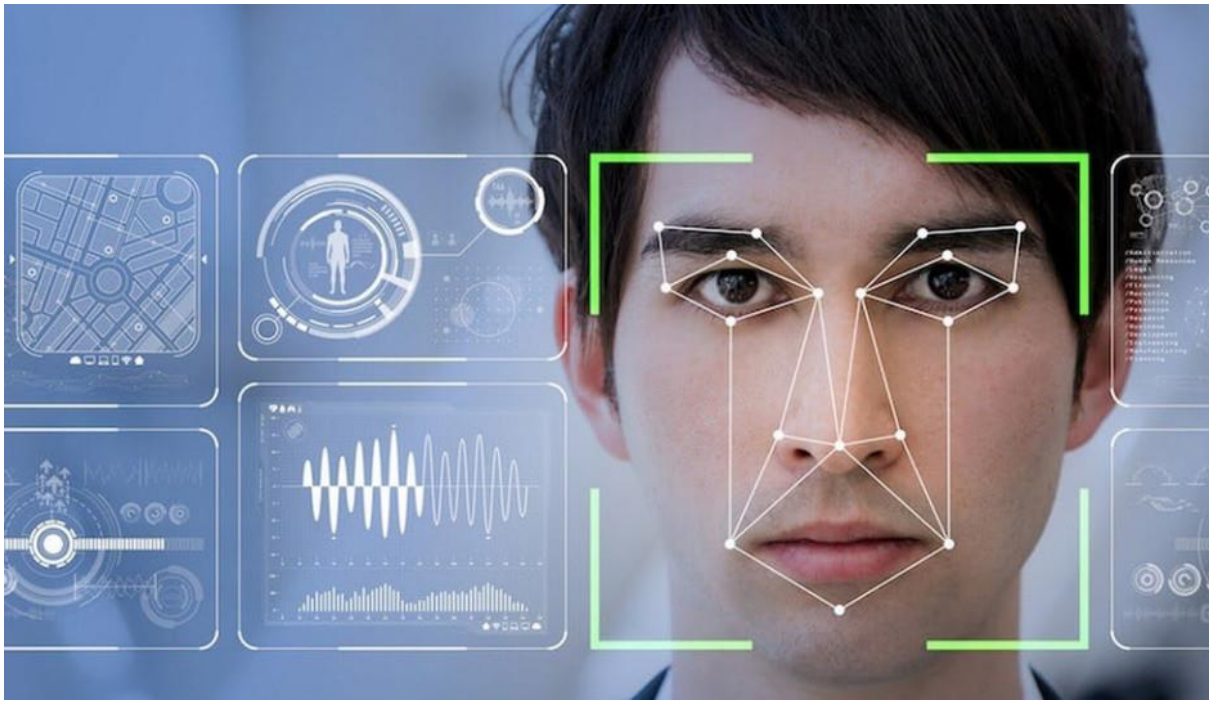
Học máy đang ngày càng được ứng dụng rộng rãi, đa lĩnh vực trong thực tiễn cuộc sống con người ngày nay và được sử dụng với mục đích phân tích dữ liệu lớn để có thể đưa ra những dự đoán xu hướng trong tương lai đồng thời cũng được áp dụng trong công nghệ nhận diện hình ảnh. Ví dụ như dự đoán kết quả bầu cử chính trị, dự đoán biến động của thị trường chứng khoán, phát hiện và nhận diện khuôn mặt …

❖ *Machine learning được ứng dụng cực kỳ nhiều trong đời sống hiện nay trong mọi lĩnh vực:*

- Tài chính – ngân hàng
- Sinh học
- Nông nghiệp
- Tìm kiếm, trích xuất thông tin
- Tự động hóa
- Robotics
- Hóa học
- Mạng máy tính
- Khoa học vũ trụ
- Xử lý ngôn ngữ tự nhiên
- Thị giác máy tính

➢ *Một số ứng dụng cụ thể của học máy*

- Phát hiện và nhận diện hình ảnh: nhận diện hình ảnh là một trong những ứng dụng của học máy và trí tuệ nhân tạo phổ biến nhất. Về cơ bản, nó là một cách tiếp cận để xác định và phát hiện các đặc trưng của một đối tượng trong hình ảnh kỹ thuật số. Bên cạnh đó, kỹ thuật này có thể được sử dụng để phân tích sâu hơn, chẳng hạn như nhận dạng mẫu, nhận diện hình khuôn, nhận dạng khuôn mặt, nhận dạng ký tự quang học và nhiều hơn nữa, ...



Hình 2.7 Minh họa nhận diện khuôn mặt trong ảnh (Ảnh: Internet)

- Lọc thư rác, phân loại văn bản: dựa trên nội dung thư điện tử, chia thư thành loại "thư rác" hay "thư bình thường"; hoặc phân chia tin tức thành các thể loại khác nhau như "xã hội", "kinh tế", "thể thao". v.v.
- Dịch tự động: dựa trên dữ liệu huấn luyện dưới dạng các văn bản song ngữ, hệ thống dịch tự động học cách dịch từ ngôn ngữ này sang ngôn ngữ khác.
- Chẩn đoán y tế: mô hình học máy học cách dự đoán xem người bệnh có mắc hay không mắc một số bệnh nào đó dựa trên triệu chứng quan sát được. Ví dụ như chuẩn đoán người bệnh có bị sâu răng hay không dựa trên tập huấn luyện là một tập hình ảnh chụp X-quang răng
- Phân loại khách hàng và dự đoán sở thích: dựa vào các yếu tố cơ bản về màu sắc, kích thước … để sắp xếp khách hàng vào một số loại, từ đây dự đoán sở thích tiêu dùng của khách hàng.

Hình 2.8 Ví dụ minh họa phân loại khách hàng (Ảnh: Internet)

- Dự đoán chỉ số thị trường: căn cứ giá trị một số tham số hiện thời và trong lịch sử, đưa ra dự đoán, chẳng hạn dự đoán giá chứng khoán, giá vàng...
- Các hệ khuyến nghị, hay hệ tư vấn lựa chọn: đề xuất một danh sách ngắn các loại hàng hóa, phim, video, tin tức v.v. mà người dùng nhiều khả năng quan tâm. Ví dụ ứng dụng loại này là phần khuyến nghị trên Youtube hay trên các trang thương mại điện tư như Amazon, Shopee, Lazada
- Trợ lý cá nhân ảo (Virtual Personal Assistants): trợ lý các nhân ảo hỗ trợ tìm kiếm thông tin thông qua văn bản, giọng nói hoặc hình ảnh ví dụ như Google Assitant, Siri …

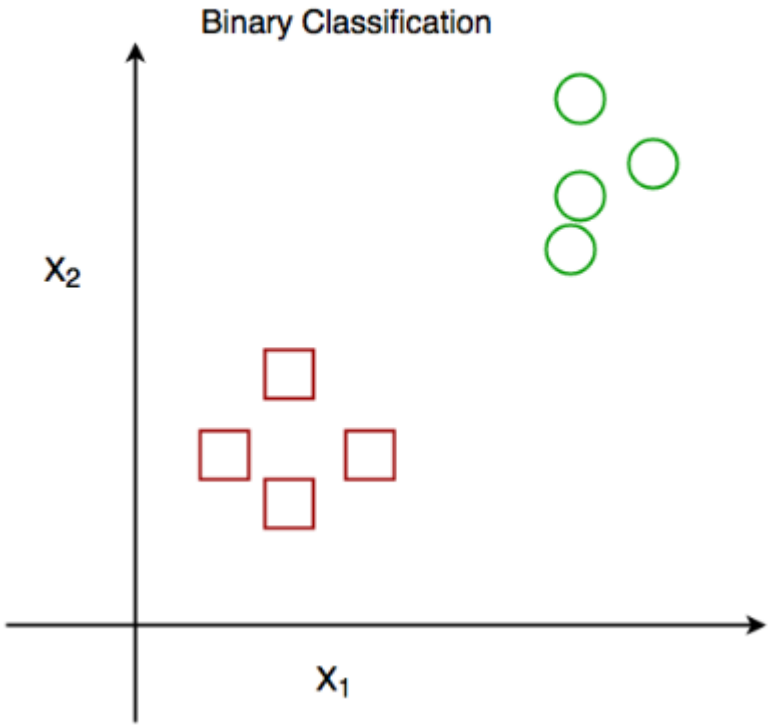
Hình 2.9 Ví dụ minh họa trợ lý ảo Google Assistant

2.1.2 Một số bài toán trong học máy

Học máy là ngành khoa học đang được phát triển mạnh, nó giúp máy tính dự đoán ra những dữ liệu chưa biết dựa trên các dữ liệu đã biết thông qua các thuật toán. Chính vì vậy nó được ứng dụng trên nhiều mảng khác nhau. Mặc dù vậy, người ta quy về một số bài toán phổ biến như:

- Phân loại nhị phân - Binary Classification

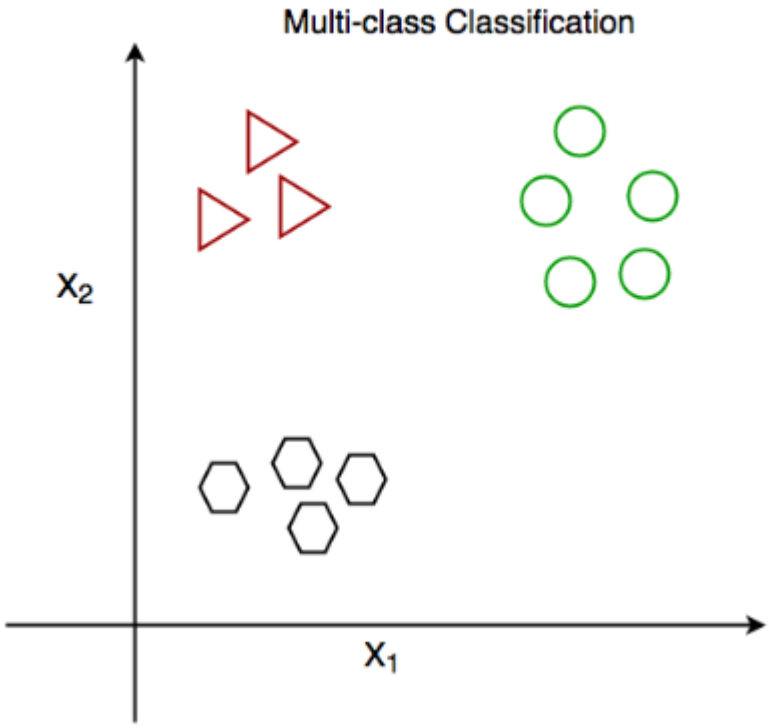
Phân loại nhị phân là một trong những bài toán phân loại phổ biến nhất trong Machine Learning. Ta xem xét một vector dữ liệu trong tập dữ liệu được gán nhãn với class với nhận 2 giá trị 0 hoặc 1 tức là chỉ có 2 trạng thái, 2 class



Hình 2.10 Minh họa bài toán phân loại nhị phân

- Phân loại đa lớp - Multiclass Classification

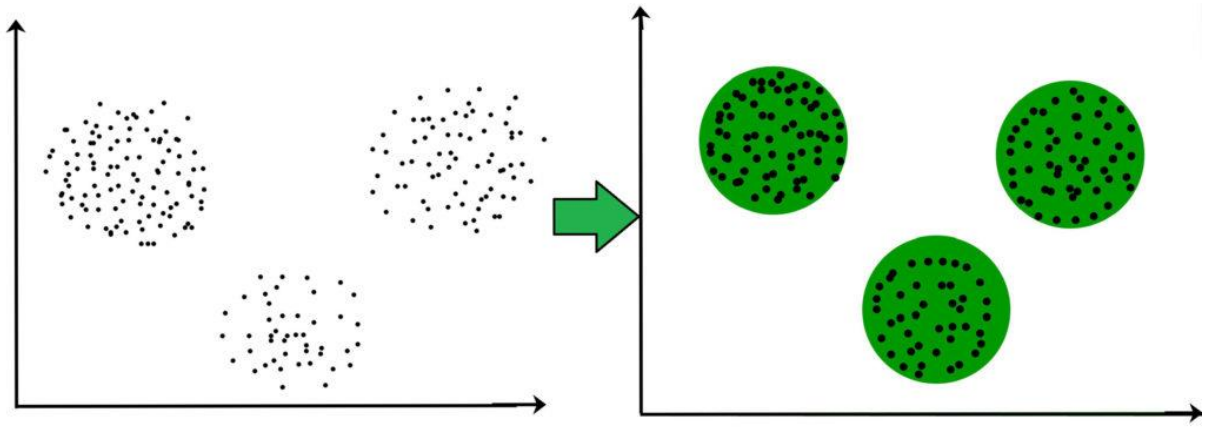
Tương tự với logic của phân loại nhị phân ở phần trên, tuy nhiên số class trong trường hợp này không còn là 2 nữa. Điểm khác biệt lớn nhất có thể thấy đó chính là độ chính xác hay chính là cost error của bài toán. Bài toán phân loại đa lớp được áp dụng phổ biến trong một số vấn đề thực tế như nhận dạng khuôn mặt, nhận dạng chữ viết tay, nhận dạng ô tô….



Hình 2.11 Minh họa phân loại đa lớp

- Phân cụm - Clustering

Nhiệm vụ của phân cụm là phân chia nhóm các điểm dữ liệu thành một số nhóm sao cho các điểm dữ liệu trong cùng một nhóm là giống nhau và khác với các điểm dữ liệu trong các nhóm khác. Phân cụm là một bài toán mô tả hướng tới việc nhận biết một tập hữu hạn các cụm hoặc các lớp để mô tả dữ liệu [3]. Về cơ bản, nó là một tập hợp các đối tượng trên cơ sở giống nhau và không giống nhau giữa chúng.



Hình 2.12 Ví dụ minh họa bài toán phân cụm

- Hồi quy - Regression

Đây là bài toán mà từ một tập các dữ liệu quan sát được, ta sẽ tìm cách để xây dựng các công thức biểu diễn để mô tả và tổng quát hóa tự nhiên. Về mặt bản chất hồi quy là bài toán gán nhãn cho dữ liệu thực, biểu diễn và dự đoán đầu ra dựa trên tổng quát hóa các dữ liệu từ đầu vào để tìm ra một hàm dự đoán. Chẳng hạn xây dựng hàm dự đoán giá nhà, giá cổ phiếu theo thời gian hoặc các biến đầu vào khác. Ta đánh giá một mô hình hồi quy thông qua độ lệch dự đoán trung bình của đầu ra. Điểm khó khăn nhất với hồi quy là xác định hàm hồi quy, và vấn đề gây rắc rối nhất là các dữ liệu nhiễu hoặc quan sát dữ liệu sai. Tất nhiên, trên thực tế người ta đã sử dụng mô hình hồi quy để ứng dụng vào các bài toán như:

- Dự đoán giá cả của sản phẩm
- Dự đoán biến động chứng khoán
- Dự đoán thời tiết
- …

Bài toán hồi quy là một dạng bài toán được đa phần các nhà khoa học thực hiện để mô tả thế giới bằng việc 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 Trees)

2.2.1.1 Khái niệm

Cây quyết định (Decision Trees) là một thuật toán học tập có giám sát không tham số, được sử dụng cho cả nhiệm vụ phân loại và hồi quy. Nó có cấu trúc dạng cây, phân cấp, bao gồm nút gốc (root node), các nhánh, các nút bên trong (internal node) và các nút lá (leaf nodes).

Cây quyết định bắt đầu bằng một nút gốc, không có bất kỳ nhánh nào đến. Các nhánh đi từ nút gốc sau đó đưa vào các nút bên trong, còn được gọi là nút quyết định. Dựa trên các đặc điểm sẵn có, cả hai loại nút đều tiến hành đánh giá để tạo thành các tập

con đồng nhất, được ký hiệu bằng các nút lá, hoặc các nút đầu cuối. Các nút lá đại diện cho tất cả các kết quả có thể có trong tập dữ liệu.

Cây quyết định xây dựng cây bằng cách đặt một loạt câu hỏi vào dữ liệu để đi đến quyết định. Do đó người ta nói rằng Cây Quyết định bắt chước quá trình 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 và nhược điểm của cây quyết định

- Ưu Điểm

So với các phương pháp khai phá dữ liệu khác, cây quyết định là phương pháp có một số ưu điểm:

+ Cây quyết định dễ hiểu. Người ta có thể hiểu mô hình cây quyết định sau khi được giải thích ngắn.

+ Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết. Các kỹ thuật khác thường đòi hỏi chuẩn hóa dữ liệu, cần tạo các biến phụ (dummy variable) và loại bỏ các giá trị rỗng.

+ Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến. Chẳng hạn, các luật quan hệ chỉ có thể dùng cho các biến tên, trong khi mạng nơ-ron chỉ có thể dùng cho các biến có giá trị bằng số.

+ Cây quyết định là một mô hình hộp trắng. Nếu có thể quan sát một tình huống cho trước trong một mô hình, thì có thể dễ dàng giải thích điều kiện đó bằng logic Boolean. Mạng nơ-ron là một ví dụ về mô hình hộp đen, do lời giải thích cho kết quả quá phức tạp để có thể hiểu được.

+ Có thể thẩm định một mô hình bằng các kiểm tra thống kê. Điều này làm cho ta có thể tin tưởng vào mô hình.

+ Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn. Có thể dùng máy tính cá nhân để phân tích các lượng dữ liệu lớn trong một thời gian đủ ngắn để cho phép các nhà chiến lược đưa ra quyết định dựa trên phân tích của cây quyết định.

• Nhược điểm

+ Khó giải quyết được những vấn đề có dữ liệu phụ thuộc thời gian liên tục - dễ xảy ra lỗi khi có quá nhiều lớp chi phí tính toán để xây dựng mô hình cây quyết định.

+ Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.

2.2.1.3 Một vài thuật ngữ trong cây quyết định

Nút gốc (Root node)

Nút gốc là nơi bắt đầu cây quyết định. Nó đại diện cho toàn bộ tập dữ liệu, được chia thành hai hoặc nhiều tập đồng nhất.

Nút lá (Leaf node)

Các nút lá là nút đầu ra cuối cùng và cây không thể được phân tách thêm sau khi nhận được nút lá.

Tách (Splitting)

Tách là quá trình phân chia nút quyết định / nút gốc thành các nút con theo các điều kiện cho trước.

Cành / Cây phụ (Branch/Sub Tree)

Cây được hình thành bằng cách tách cây.

Tỉa cành (Pruning)

Tỉa cành là quá trình loại bỏ những cành không mong muốn khỏi cây.

Nút cha / nút con (Parent/Child node)

Nút gốc của cây được gọi là nút cha, và các nút khác được gọi là nút con.

2.2.1.4 Cơ chế hoạt động của cây quyết định

- Cây quyết định sử dụng chiến lược phân chia và chinh phục bằng cách thực hiện một tìm kiếm tham lam để xác định các điểm phân tách tối ưu trong một cây. Quá trình tách này sau đó được lặp lại theo cách thức đệ quy từ trên xuống cho đến khi tất cả hoặc phần lớn các bản ghi đã được phân loại theo các nhãn lớp cụ thể.

- Tất cả các điểm dữ liệu có được phân loại thành các tập đồng nhất hay không phần lớn phụ thuộc vào độ phức tạp của cây quyết định. Các cây nhỏ hơn có thể dễ dàng đạt được các nút lá thuần túy — tức là các điểm dữ liệu trong một lớp duy nhất.

Tuy nhiên, khi một cây phát triển về kích thước, việc duy trì độ tinh khiết này ngày càng trở nên khó khăn và nó thường dẫn đến quá ít dữ liệu nằm trong một cây con nhất định.

- Khi điều này xảy ra, nó được gọi là phân mảnh dữ liệu và nó thường có thể dẫn đến trang bị quá mức. Kết quả là, cây quyết định ưu tiên những cây nhỏ, điều này phù hợp với nguyên tắc của parsimony trong Occam's Razor; nghĩa là, "các thực thể không nên được nhân lên quá mức cần thiết." Nói cách khác, cây quyết định chỉ nên tăng thêm độ phức tạp khi cần thiết, vì lời giải thích đơn giản nhất thường là tốt nhất.

- Để giảm bớt sự phức tạp và ngăn ngừa việc trang bị quá nhiều, việc cắt tỉa thường được sử dụng; đây là một quá trình, loại bỏ các nhánh phân chia trên các đối tượng địa lý có tầm quan trọng thấp. Sau đó, sự phù hợp của mô hình có thể được đánh giá thông qua quá trình xác nhận chéo.

- Một cách khác mà cây quyết định có thể duy trì độ chính xác của chúng là bằng cách tạo thành một tập hợp thông qua một thuật toán rừng ngẫu nhiên, trình phân loại này dự đoán kết quả chính xác hơn, đặc biệt khi các cây riêng lẻ không tương quan với nhau.
- Trong cây quyết định, để dự đoán lớp của tập dữ liệu đã cho, thuật toán bắt đầu từ nút gốc của cây. Thuật toán này so sánh các giá trị của thuộc tính gốc với thuộc tính bản ghi (tập dữ liệu thực) và dựa trên sự so sánh, đi theo nhánh và nhảy đến nút tiếp theo.
- Đối với nút tiếp theo, thuật toán lại so sánh giá trị thuộc tính với các nút con khác và di chuyển xa hơn. Nó tiếp tục quá trình cho đến khi nó đạt đến nút lá của cây. Quy trình hoàn chỉnh có thể được hiểu rõ hơn bằng cách sử dụng thuật toán dưới đây:

+ Bước 1: Bắt đầu cây với nút gốc (Đặt tên: S), nút này chứa tập dữ liệu hoàn chỉnh.

+ Bước 2: Tìm thuộc tính tốt nhất trong tập dữ liệu bằng cách sử dụng Phép đo lựa chọn thuộc tính (ASM).

+ Bước 3: Chia S thành các tập con chứa các giá trị có thể có cho các thuộc tính tốt nhất.

+ Bước 4: Tạo nút cây quyết định 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. Tiếp tục quá trình này cho đến khi đạt đến một giai đoạn mà bạn không thể phân loại thêm các nút và được gọi là nút cuối cùng là nút lá.

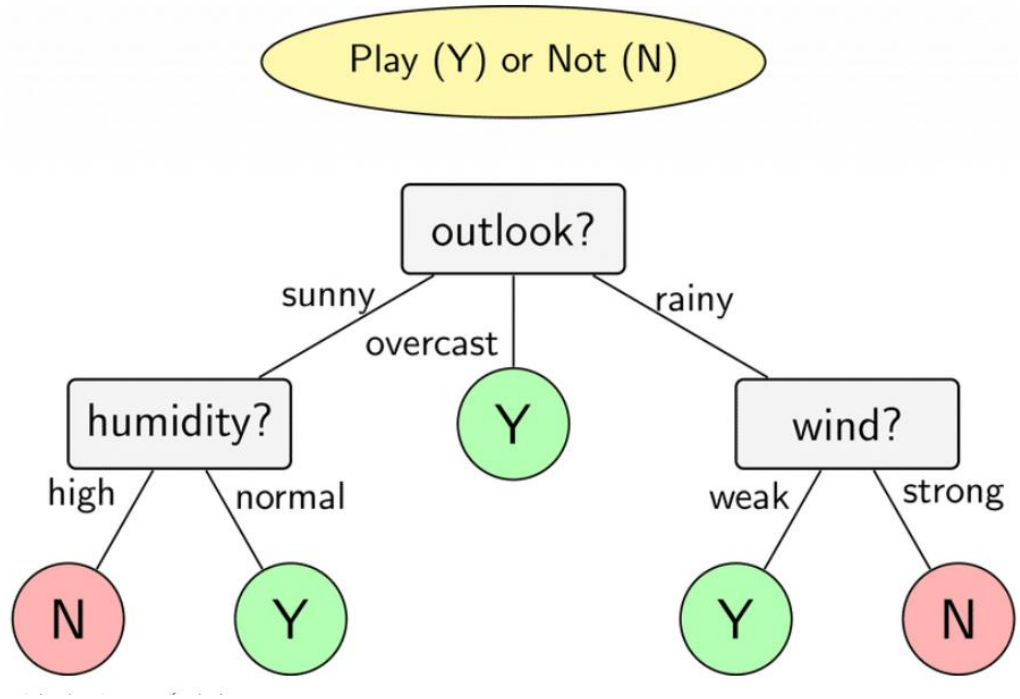
2.2.1.5 Ví dụ

Ta hãy xét một ví dụ khá phổ biến và kinh điển về cây quyết định. Giả sử dựa theo thời tiết mà quyết định đi đá bóng hay không?

Những đặc điểm ban đầu là:

+ Thời tiết(outlook)
+ Độ ẩm(humidity)
+ Gió(wind)

Dựa vào những thông tin trên, bạn có thể xây dựng được mô hình như sau:



Hình 2.13 Ví dụ cơ trong thuật toán cây quyết định

Dựa theo mô hình trên, ta thấy:

Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

2.2.1.6 Thuật toán mở rộng cây quyết định

Thuật toán của Hunt, được phát triển vào những năm 1960 để mô hình hóa việc học tập của con người trong Tâm lý học, tạo thành nền tảng của nhiều thuật toán cây quyết định phổ biến, chẳng hạn như sau:

– **ID3**: Ross Quinlan được ghi nhận trong quá trình phát triển ID3, viết tắt của "Iterative Dichotomiser 3." Thuật toán này tận dụng entropy và thu thập thông tin làm số liệu để đánh giá sự phân chia ứng viên.

– **C4.5**: Thuật toán này được coi là sự lặp lại sau này của ID3, thuật toán này cũng được phát triển bởi Quinlan. Nó có thể sử dụng tỷ lệ thu được hoặc thu được thông tin để đánh giá các điểm phân tách trong cây quyết định.

– Ngoài ID3, C4.5, ta còn một số thuật toán khác như:

+ Thuật toán CHAID: tạo cây quyết định bằng cách sử dụng thống kê Chi-square để xác định các phân tách tối ưu. Các biến mục tiêu đầu vào có thể là số (liên tục) hoặc phân loại.

+ Thuật toán C&R: sử dụng phân vùng đệ quy để chia cây. Tham biến mục tiêu có thể dạng số hoặc phân loại.

+ MARS

+ Conditional Inference Trees.

2.2.2 Một số thuật toán mở rộng cây quyết định

2.2.2.1 Thuật toán ID3

Iterative Dichotomiser 3 (ID3) là thuật toán nổi tiếng để xây dựng Decision Tree, áp dụng cho bài toán Phân loại (Classification) mà tất các các thuộc tính để ở dạng category

a) Ý tưởng

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

- Sau mỗi *câu hỏi*, dữ liệu được phân chia vào từng *child node* tương ứng với các câu trả lời cho câu hỏi đó. *Câu hỏi* ở đây chính là một thuộc tính, câu trả lời chính là giá trị của thuộc tính đó. Để đánh giá *chất lượng* của một cách phân chia, chúng ta cần đi tìm một phép đo.

- Trước hết, thế nào là một phép phân chia tốt? Bằng trực giác, một phép phân chia là tốt nhất nếu dữ liệu trong mỗi *child node* hoàn toàn thuộc vào một class–khi đó *child node* này có thể được coi là một *leaf node*, tức ta không cần phân chia thêm

nữa. Nếu dữ liệu trong các *child node* vẫn lẫn vào nhau theo tỉ lệ lớn, ta coi rằng phép phân chia đó chưa thực sự tốt. Từ nhận xét này, ta cần có một hàm số đo *độ tinh khiết* (*purity*), hoặc *độ vẩn đục* (*impurity*) của một phép phân chia. Hàm số này sẽ cho giá trị thấp nhất nếu dữ liệu trong mỗi *child node* nằm trong cùng một class (tinh khiết nhất), và cho giá trị cao nếu mỗi *child node* có chứa dữ liệu thuộc nhiều class khác nhau.

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

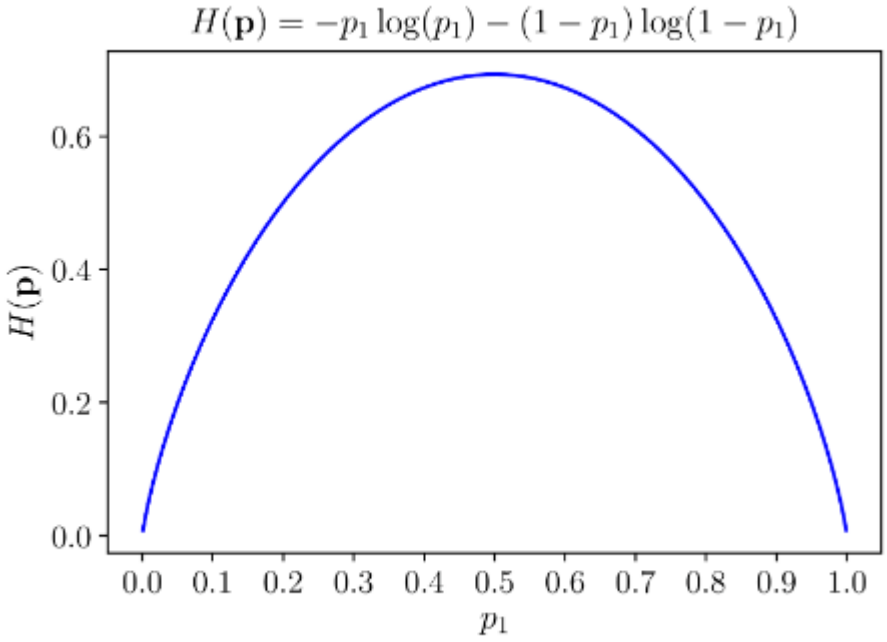
b) Hàm số entropy

Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2, \ldots, x_n$. Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x = x_i)$ với $0 \le p_i \le 1, \sum_{i=1}^{n} p_i = 1$. Ký hiệu phân phối này là $\mathbf{p} = (p_1, p_2, \ldots, p_n)$. Entropy của phân phối này được định nghĩa là

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

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

Xét một ví dụ với $n = 2$ được cho trên Hình 3. Trong trường hợp $\mathbf{p}$ là *tinh khiết* nhất, tức một trong hai giá trị p_i bằng 1, giá trị kia bằng 0, entropy của phân phối này là $H(\mathbf{p}) = 0$. Khi $\mathbf{p}$ là *vẩn đục* nhất, tức cả hai giá trị $p_i = 0.5$, hàm entropy đạt giá trị cao nhất.



Hình 2.14 Đồ thị của hàm entropy với n=2

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

Những tính chất này của hàm entropy khiến nó được sử dụng trong việc đo độ vẩn đục của một phép phân chia của ID3. Vì lý do này, ID3 còn được gọi là entropy-based decision tree.

c) Thuật toán ID3

Trong ID3, tổng có trọng số của entropy tại các leaf-node sau khi xây dựng decision tree được coi là hàm mất mát của decision tree đó. Các trọng số ở đây tỉ lệ với số điểm dữ liệu được phân vào mỗi node. Công việc của ID3 là tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt. Như đã đề cập, việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất. Bài toán xây dựng một decision tree bằng ID3 có thể chia thành các bài toán nhỏ, trong mỗi bài toán, ta chỉ cần chọn ra thuộc tính giúp cho việc phân chia đạt kết quả tốt nhất. Mỗi bài toán nhỏ này tương ứng với việc phân chia dữ liệu trong một non-leaf node. Chúng ta sẽ xây dựng phương pháp tính toán dựa trên mỗi node này.

Xét một bài toán với C class khác nhau. Giả sử ta đang làm việc với một non-leaf node với các điểm dữ liệu tạo thành một tập $\mathcal{S}$ với số phần tử là $|\mathcal{S}| = N$. Giả sử thêm rằng trong số N điểm dữ liệu này, $N_c, c = 1, 2, \ldots, C$ 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, entropy tại node này được tính bởi:

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

Tiếp theo, giả sử thuộc tính được chọn là x. Dựa trên x, các điểm dữ liệu trong $\mathcal{S}$ được phân ra thành K child node $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K$ với số điểm trong mỗi child node lần lượt là $m_1, m_2, \ldots, m_K$. Ta định nghĩa

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

là tổng có trọng số entroy của mỗi child node–được tính tương tự như (2). Việc lấy trọng số này là quan trọng vì các node thường có số lượng điểm khác nhau.

Tiếp theo, ta định nghĩa information gain dựa trên thuộc tính x:

$$G(x, \mathcal{S}) = H(\mathcal{S}) - H(x, \mathcal{S})$$

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

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

tức thuộc tính khiến cho information gain đạt giá trị lớn nhất.

d) Ví dụ

Để mọi thứ được rõ ràng hơn, chúng ta cùng xem ví dụ với dữ liệu huấn luyện được cho trong Bảng dưới đây. Bảng dữ liệu này được lấy từ cuốn sách Data Mining: Practical Machine Learning Tools and Techniques, trang 11. Đây là một bảng dữ liệu được sử dụng rất nhiều trong các bài giảng về decision tree. Bảng dữ liệu này mô tả mối quan hệ giữa thời tiết trong 14 ngày (bốn cột đầu, không tính cột id) và việc một đội bóng có chơi bóng hay không (cột cuối cùng). Nói cách khác, ta phải dự đoán giá trị ở cột cuối cùng nếu biết giá trị của bốn cột còn lại.

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 2.15 Bảng giá trị thời tiết

Có bốn thuộc tính thời tiết:

Outlook nhận một trong ba giá trị: sunny, overcast, rainy.

Temperature nhận một trong ba giá trị: hot, cool, mild.

Humidity nhận một trong hai giá trị: high, normal.

Wind nhận một trong hai giá trị: weak, strong.

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

Nếu outlook = sunny và humidity = high thì play = no.

Nếu outlook = rainy và windy = true thì play = no.

Nếu outlook = overcast thì play = yes.

Ngoài ra, nếu humidity = normal thì play = yes.

Ngoài ra, play = yes.

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

Trong 14 giá trị đầu ra ở Bảng trên, có năm giá trị bằng no và chín giá trị bằng yes. Entroy tại 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 tính tổng có trọng số entropy của các child node nếu chọn một trong các thuộc tính outlook, temperature, humidity, wind, play để phân chia dữ liệu.

Xét thuộc tính outlook. Thuộc tính này có thể nhận một trong ba giá trị sunny, overcast, rainy. Mỗi một giá trị sẽ tương ứng với một child node. Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_s, S_o, S_r với tương ứng m_s, m_o, m_r phần tử. Sắp xếp lại Bảng ban đầu theo thuộc tính outlook ta đạt được ba Bảng nhỏ sau đây.

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 2.16 Bảng giá trị theo thời tiết là sunny

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 2.17 Bảng giá trị theo thời tiết là overcast

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 2.18 Bảng giá trị theo thời tiết là rainy

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

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

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 2.19 Bảng giá trị theo nhiệt độ là hot

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 2.20 Bảng giá trị theo nhiệt độ là mild

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 2.21 Bảng giá trị theo nhiệt độ là cool

Gọi $\mathcal{S}_h, \mathcal{S}_m, \mathcal{S}_c$ là ba tập con tương ứng với temperature bằng hot, mild, cool.

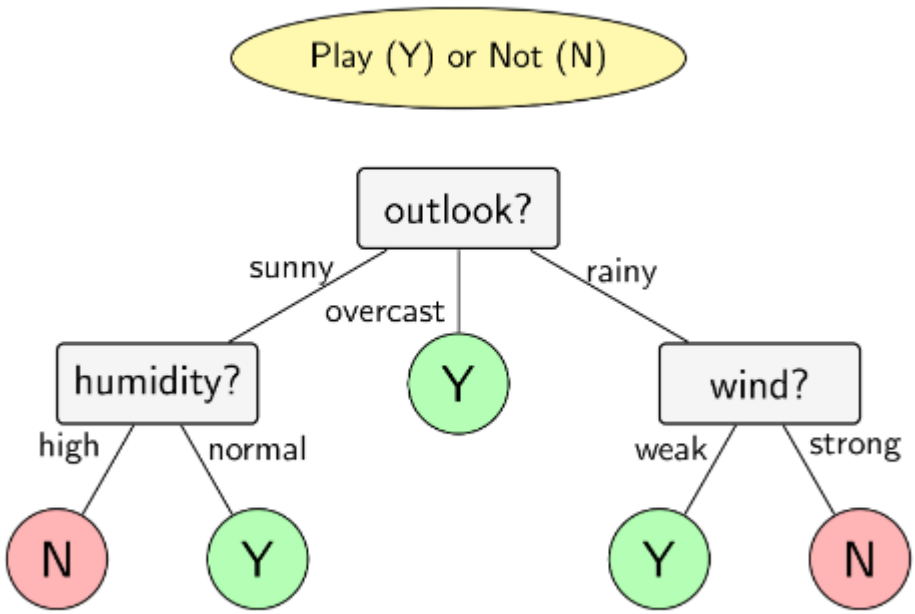
Việc tính toán với hai thuộc tính còn lại được dành cho bạn đọc. Nếu các kết quả là giống nhau, chúng sẽ bằng:

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

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

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

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



Hình 2.22 Cây quyết định ID3

e) Điều kiện dừng

Trong các thuật toán decision tree nói chung và ID3 nói riêng, nếu ta tiếp tục phân chia các node chưa tinh khiết, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, tree có thể sẽ rất phức tạp (nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng overfitting sẽ xảy ra.

Để tránh overfitting, một trong số các phương pháp sau có thể được sử dụng. Tại một node, nếu một trong số các điều kiện sau đây xảy ra, ta không tiếp tục phân chia node đó và coi nó là một leaf node:

nếu node đó có entropy bằng 0, tức mọi điểm trong node đều thuộc một class.

nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting. Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.

nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh overfitting.

nếu tổng số leaf node vượt quá một ngưỡng nào đó.

nếu việc phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

Ngoài các phương pháp trên, một phương pháp phổ biến khác được sử dụng để tránh overfitting là pruning, tạm dịch là cắt tỉa.

2.2.2.2 Thuật toán C4.5

a. Khái niệm

Thuật toán C4.5 là thuật toán cải tiến của ID3.

Trong thuật toán ID3, Information Gain được sử dụng làm độ đo. Tuy nhiên, phương pháp này lại ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Do vậy, để khắc phục nhược điểm trên, ta sử dụng độ đo Gain Ratio (trong thuật toán C4.5) như sau:

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

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

Trong đó: Split Info được tính như sau:

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

b. Điều kiện dừng

Trong các thuật toán Decision tree, với phương pháp chia trên, ta sẽ chia mãi các node nếu nó chưa tinh khiết. Như vậy, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp

(nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng overfitting sẽ xảy ra.

Để tránh trường hợp này, ta có thể dừng cây theo một số phương pháp sau đây:

- nếu node đó có entropy bằng 0, tức mọi điểm trong node đều thuộc một class.
- nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting. Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.
- nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh overfitting.
- nếu tổng số leaf node vượt quá một ngưỡng nào đó.
- nếu việc phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

Ngoài ra, ta còn có phương pháp cắt tỉa cây.

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 các ứng dụng phần mềm, khoa học dữ liệu, học máy.

Python là ngôn ngữ có cấu trúc rõ ràng, dễ học và thuận tiện cho người mới học lập trình. Vì thế nó được sử dụng rộng rãi.

Python là ngôn ngữ đa nền tảng hỗ trợ nhiều mẫu đa lập trình khác nhau như: mệnh lệnh, lập trình hướng đối tượng, lập trình hàm… được dùng đa lĩnh vực: web, 3D CAD, Hiện đang được sử dung hầu hết bởi các công ty công nghệ lớn như Google, Amazon, Facebook, Instagram, Dropbox, Uber, v.v.

Hình 2.23 Minh họa ngôn ngữ lập trình Python

2.3.1.2 Một số ứng dụng của Python

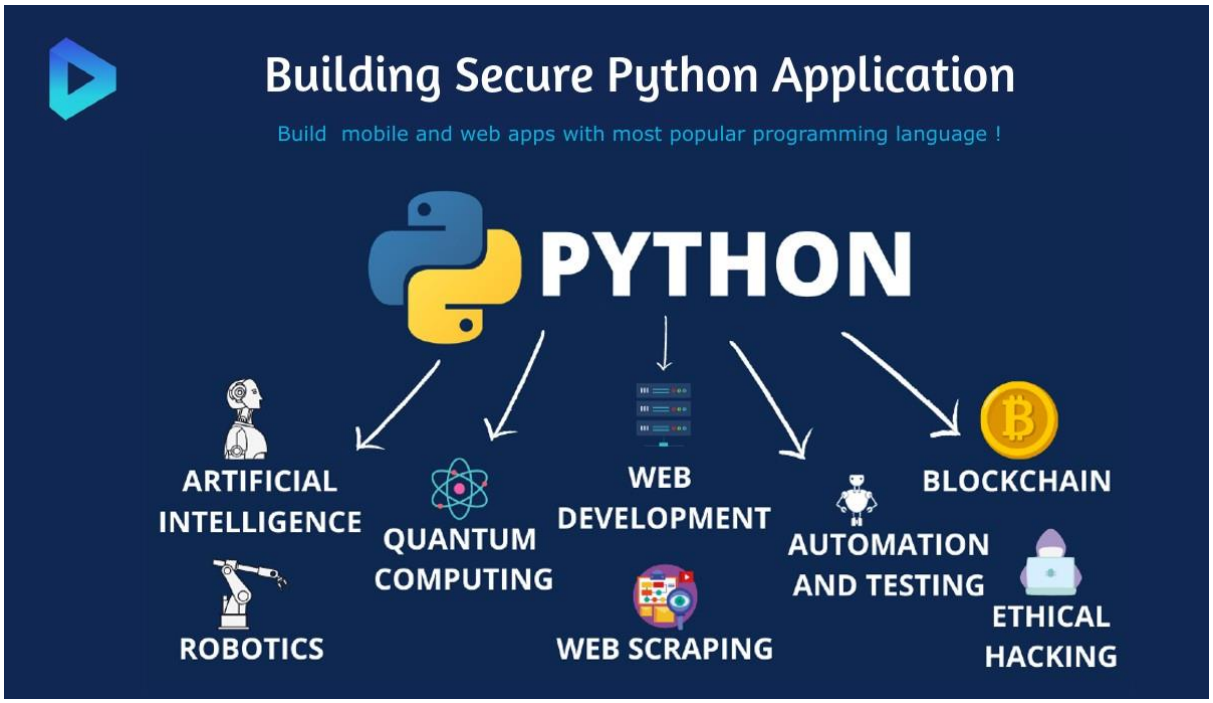
Ngôn ngữ Python đang được sự dụng rộng rãi trong nhiều lĩnh vực khác nhau. Một số lĩnh vực ứng dụng ngôn ngữ Python:

- Học máy và trí tuệ nhân tạo
- Phân tích dữ liệu
- Các ứng dụng GUI (như Kivy , Tkinter, PyQt, Qt Designer v.v.)
- Các framework web như Django (được sử dụng bởi YouTube, Instagram, Dropbox)
- Xử lý hình ảnh (như OpenCV)
- Quét web (như Scrapy, BeautifulSoup, Selenium)
- Xử lý văn bản và nhiều hơn nữa …

2.3.1.3 Một số tính năng chính của Python

- Phát triển trang web (phía máy chủ).

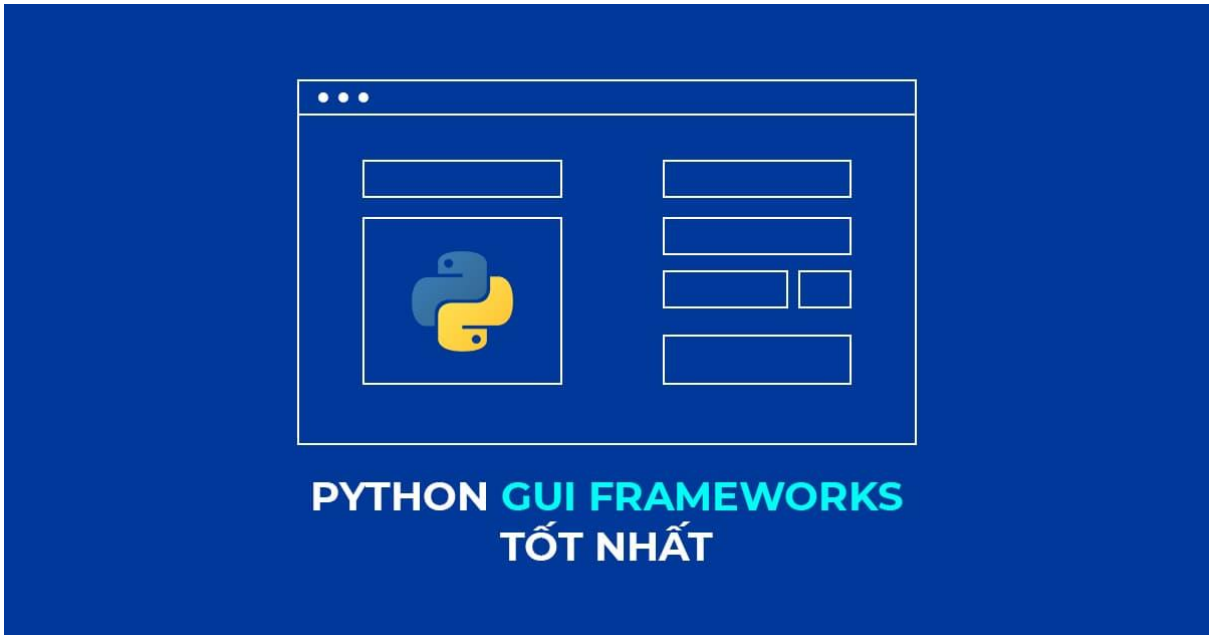
- Phát triển phần mềm
- Tự động hóa kiểm thử phần mềm
- Python có thể kết nối với các hệ thống cơ sở dữ liệu.
- Xử lý dữ liệu lớn và thực hiện các phép toán phức tạp.



Hình 2.24 Python và các ứng dụng trong thực tế

2.3.2 Các Python GUI Frameworks tốt nhất

Python là một ngôn ngữ dễ học và nổi bật nhất trong thời điểm hiện tại và việc sử dụng Python để viết các chương trình GUI (Giao diện người dùng đồ họa) cũng không hề khó.



Hình 2.25 Python GUI Frameworks

Nhất là khi Python có rất nhiều GUI Frameworks tốt hỗ trợ việc tạo GUI đơn giản, mạnh mẽ hơn. Từ các GUI Frarmework đa nền tảng (Cross-Platform) đến các GUI Framwork cho nền tảng cụ thể (Platform-Specific), có thể liệt kê ra 6 Python GUI framework tốt như:

2.3.2.1 Python GUI Framworks #1: Kivy

- Kivy là một Framework tăng tốc OpenGL ES 2 để tạo giao diện người dùng mới. Kivy chạy trên Linux, Windows, OS X, Android, iOS và Raspberry Pi. Bạn có thể chạy cùng một mã nguồn trên tất cả các nền tảng được hỗ trợ.

- Nó có thể sử dụng nguyên bản hầu hết các đầu vào, giao thức và thiết bị bao gồm WM_Touch, WM_Pen, Mac OS X Trackpad và Magic Mouse, Mtdev, Linux Kernel HID, TUIO. Một mô phỏng chuột cảm ứng đa điểm.

- Kivy được sử dụng miễn phí 100%, theo giấy phép MIT (bắt đầu từ 1.7.2) và LGPL 3 cho các phiên bản trước. Bộ công cụ được phát triển, hỗ trợ chuyên nghiệp. Có thể sử dụng nó trong một sản phẩm thương mại.

- Framework ổn định và có tài liệu API tốt, cùng với hướng dẫn lập trình kèm theo để giúp chúng ta bắt đầu nhanh chóng.

2.3.2.2 Python GUI Framworks #2: PyQT

- PyQT là một trong những ràng buộc Python đa nền tảng được ưa chuộng triển khai thư viện Qt cho Framework phát triển ứng dụng Qt (thuộc sở hữu của Nokia)
- Hiện tại, PyQT có sẵn cho Unix / Linux, Windows, Mac OS X và Sharp Zaurus. Nó kết hợp những gì tốt nhất của Python và Qt và tùy thuộc vào từng lập trình viên để quyết định tạo một chương trình bằng cách viết code hay sử dụng Qt Designer để tạo các hộp thoại trực quan.

- PyQT có sẵn trong cả giấy phép thương mại cũng như GPL. Mặc dù một số tính năng có thể không có trong phiên bản miễn phí, nhưng nếu ứng dụng là mã nguồn mở thì có thể sử dụng theo giấy phép miễn phí.

2.3.2.3 Python GUI Framworks #3: Tkinter

- Tkinter thường được đóng gói với Python, và nó là GUI Framework tiêu chuẩn của Python. Nó phổ biến vì sự đơn giản và giao diện người dùng đồ họa, mã nguồn mở và có sẵn theo Python License.

- Một trong những lợi thế của việc chọn Tkinter là vì nó được cung cấp theo mặc định, nên có rất nhiều tài nguyên, cả code và sách tham khảo.

- Ngoài ra, với cộng đồng lâu đời và năng động, có nhiều người có thể sẵn sàng giúp bạn trong trường hợp bạn mới bắt đầu học, rất nhiều lỗi bạn có thể tìm ra cách sửa chữa ngay lập tức.

2.3.2.4 Python GUI Framworks #4: WxPython

- WxPython là một trình bao bọc mã nguồn mở cho thư viện GUI đa nền tảng WxWidgets (trước đó được gọi là WxWindows) và được triển khai như một mô-đun mở rộng Python.

- Với WxPython, bạn có thể tạo các ứng dụng gốc cho Windows, Mac OS và Unix.

2.3.2.5 Python GUI Framworks #5: PyGUI

- PyGUI là một GUI Frarmewowrk đa nền tảng ứng cho Unix, Macintosh và Windows. So với một số GUI Framework khác, cho đến nay, PyGUI là đơn giản nhất và nhẹ nhất, vì API hoàn toàn đồng bộ với Python.

- PyGUI chèn rất ít code giữa nền tảng GUI và ứng dụng Python, do đó giao diện của ứng dụng thường hiển thị GUI tự nhiên của nền tảng.

2.3.2.6 Python GUI Framworks #6: PySide

- PySide là một dự án phần mềm mã nguồn mở cung cấp các ràng buộc Python cho Qt Framework. Qt là một ứng dụng đa nền tảng và GUI Framework, cho phép các lập trình viên viết ứng dụng một lần và triển khai chúng trên nhiều hệ điều hành mà không cần viết lại mã nguồn, trong khi Python là một ngôn ngữ lập trình hiện đại, năng động với một cộng đồng lập trình viên khổng lồ.

- Kết hợp sức mạnh của Qt và Python, PySide cung cấp Qt Framework phong phú cho các lập trình viên sử dụng Python phát triển ứng dụng GUI nhanh chóng trên tất cả các hệ điều hành chính.

2.3.3 Xây dựng giao diện đồ hoạ với Py QT5, Qt Designer

Để hỗ trợ người dùng trong quá trình tìm kiếm thông tin, dự đoán hay xếp hạng, gợi ý cho người dùng các thông tin liên quan. Việc xây dựng mô hình dự đoán dựa trên các thuật toán học máy (Machine Learning) để thực hiện tính toán và đưa ra các dự đoán phù hợp nhất cho người dùng là việc xử lý phía bên dưới của kết quả mà người dùng có thể hiểu.

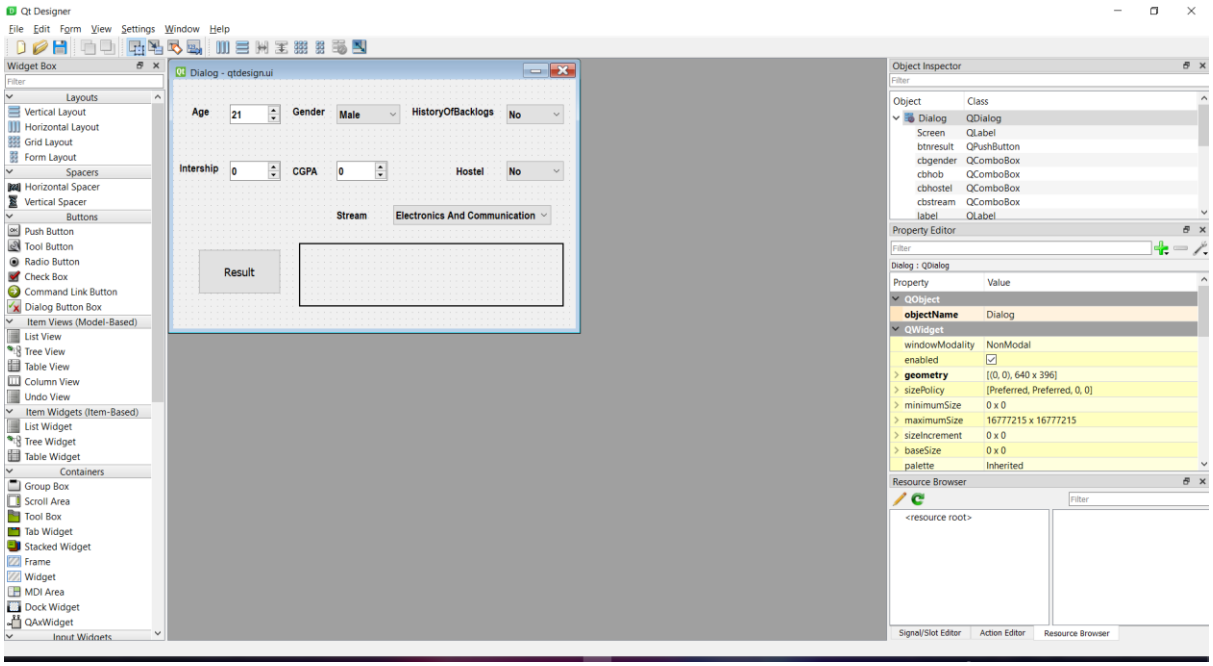
Trong quá trình học lập trình với ngôn ngữ Python, rất nhiều người quan tâm tới việc tạo các ứng dụng có giao diện như Windows Form. Trong python sẽ có rất nhiều thứ để tạo ra giao diện đồ hoạ mà ở trong đây chúng ta sẽ dùng tới PyQt5 và Qt Designer

Hình 2.26 Giao diện đồ hoạ với qt designer

Bằng việc sử dụng ngôn ngữ python xây dựng mô hình dự đoán, em lựa chọn sử dụng PyQt5 và Qt Designer làm công cụ để xây dựng giao diện hiển thị được kết quả sinh viên có thể tương tác với giao diện từ đó đưa ra kết quả dự đoán về cơ hội việc làm của sinh viên đó mà không phải tìm hiểu code ở màn hình đen xì như những gì người phát triển nhìn thấy.

2.3.3.1 Qt Designer là gì?

Qt Designer là một công cụ để nhanh chóng xây dựng giao diện người dùng đồ họa với các widget từ khung Qt GUI. Nó cung cấp cho bạn một giao diện kéo và thả đơn giản để bố trí các thành phần như nút, trường văn bản, hộp tổ hợp và hơn thế nữa. Đây là ảnh chụp màn hình của Qt Designer trên Windows:



Hình 2.27 Giao diện của Qt Designer

Qt Designer tạo ra .ui các tệp. Đây là một định dạng dựa trên XML đặc biệt để lưu trữ các widget của bạn dưới dạng cây. Bạn có thể tải các tệp này trong thời gian chạy hoặc dịch chúng sang ngôn ngữ lập trình như C ++ hoặc Python.

2.3.3.2 PyQt5 là gì

– Qt là một Application framework đa nền tảng viết trên ngôn ngữ C++ , được dùng để phát triển các ứng dụng trên desktop, hệ thống nhúng và mobile. Hỗ trợ cho các platform bao gồm : Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số platform khác. PyQt là Python interface của Qt, kết hợp của ngôn ngữ lập trình Python và thư viện Qt, là một thư viện bao gồm các thành phần giao diện điều khiển (widgets , graphical control elements).

– PyQt API bao gồm các module bao gồm số lượng lớn với các classes và functions hỗ trợ cho việc thiết kế ra các giao diện giao tiếp với người dùng của các phần mềm chức năng. Hỗ trợ với Python 2.x và 3.x.

– PyQt được phát triển bởi Riverbank Computing Limited

– Các class của PyQt5 được chia thành các module, bao gồm:

+ QtCore : là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.

+ QtGui : bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.

+ QtWidgets : bao gồm các class cho widget, ví dụ : button, hộp thoại, … được sử dụng để tạo nên giao diện người dùng cơ bản nhất.

+ QtMultimedia : thư viện cho việc sử dụng âm thanh, hình ảnh, camera,…

+ QtBluetooth : bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.

+ QtNetwork : bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client , server hỗ trợ việc lập trình mạng.

+ QtPositioning : bao gồm các class giúp việc hỗ trợ xác định vị.

+ Enginio : module giúp các client truy cập các Cloud Services của Qt.

+ QtWebSockets : cung cấp các công cụ cho WebSocket protocol.

+ QtWebKit : cung cấp các class dùng cho làm việc với các trình duyệt Web , dựa trên thư viện WebKit2.

+ QtWebKitWidgets : các widget cho WebKit.

+ QtXml : các class dùng cho làm việc với XML file.

+ QtSvg : dùng cho hiển thị các thành phần của SVG file.

+ QtSql : cung cấp các class dùng cho việc làm việc với dữ liệu.

+ QtTest : cung cấp các công cụ cho phép test các đơn vị của ứng dụng với 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()
```

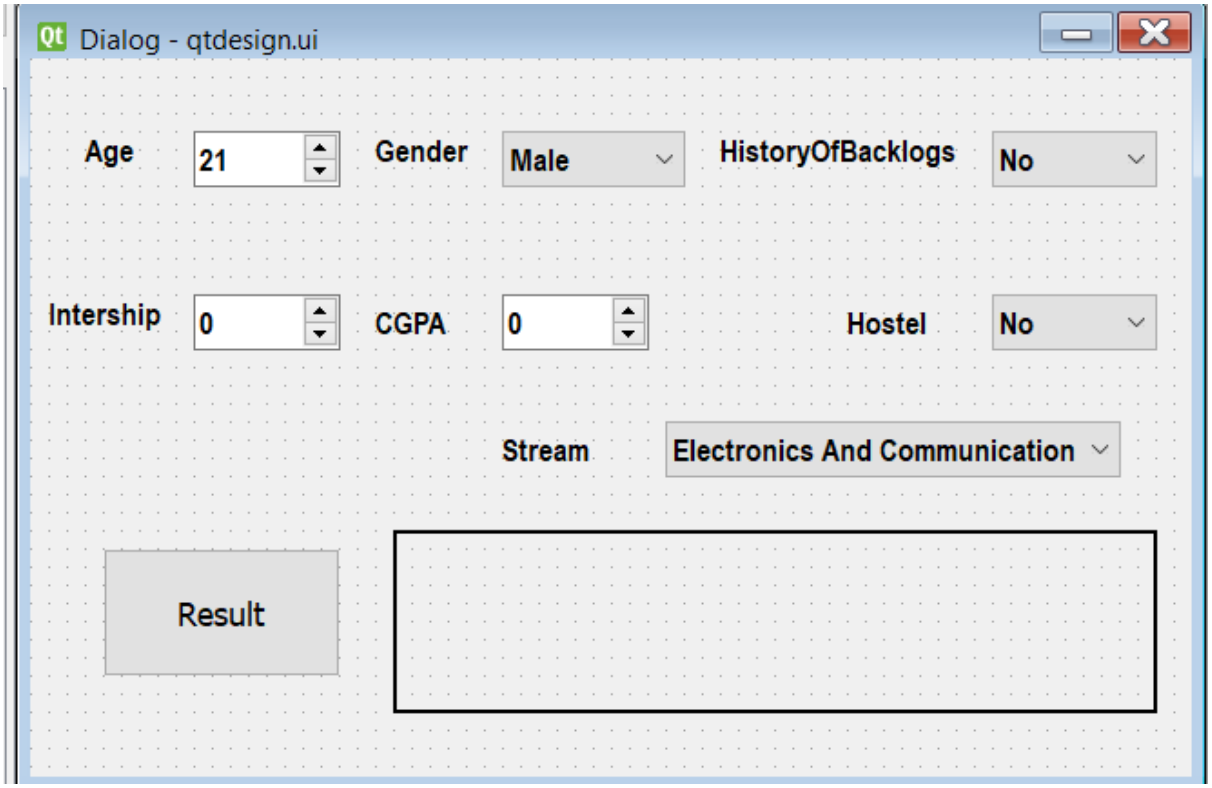
Cách 2:

```
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 2.28 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:

```
# To convert .ui file to .py using pyqt5 module
        >> pyuic5 -x main.ui -o main.py # pyQt5 version
        >> pyuic4 -x main.ui -o main.py # pyQt4 version
```

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 chạy trên tất cả các nền tảng máy tính để bàn lớn và hầu hết các nền tảng di động hoặc nhúng. Hầu hết các chương trình GUI được tạo bằng Qt đều có giao diện tự nhiên, trong trường hợp này Qt được phân loại là widget toolkit. Ngoài ra các chương trình không phải GUI cũng có thể được phát triển, chẳng hạn như các công cụ dòng lệnh và consoles cho server. Một ví dụ về một chương trình không phải GUI sử dụng Qt là khung công tác web Cutelyst.
- Qt hỗ trợ các trình biên dịch khác nhau, bao gồm trình biên dịch GCC C++ và bộ Visual Studio và có hỗ trợ quốc tế hóa rộng rãi. Qt cũng cung cấp Qt Quick, bao gồm một ngôn ngữ kịch bản lệnh được gọi là QML cho phép sử dụng JavaScript để cung cấp logic. Với Qt Quick, việc phát triển ứng dụng nhanh chóng cho các thiết bị di động trở nên khả thi, trong khi logic vẫn có thể được viết bằng mã gốc để đạt được hiệu suất tốt nhất có thể.
- Các tính năng khác bao gồm truy cập cơ sở dữ liệu SQL, phân tích cú pháp XML, phân tích cú pháp JSON, quản lý luồng và hỗ trợ mạng.

2.3.3.4 Các ứn dụng được xây dựng bằng Qt

Hiện nay có nhiều phần mềm tự do được phát triển dựa trên Qt, chẳng hạn như:

- LyX: phần mềm soạn thảo văn bản LaTeX
- Quantum GIS: phần mềm hệ thống thông tin địa lý
- QCad: phần mềm vẽ kĩ thuật
- Scribus: phần mềm xuất bản điện tử
- Skype: phần mềm giao tiếp qua mạng internet.

Một thống kê đầy đủ cho thấy Qt không chỉ xuất hiện trong máy tính mà còn 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 cho Python. Nó thường được sử dụng để phát triển ứng dụng Python. Một số tổ chức kỳ lân như Twitter, Facebook, Amazon và Pinterest cũng sử dụng PyCharm làm IDE Python của họ!

Hình 2.29 Biểu tượng của PyCharm

Chúng ta có thể chạy PyCharm trên Windows, Linux hoặc Mac OS. Ngoài ra, nó chứa các module và các package giúp các lập trình viên phát triển phần mềm bằng Python tiết kiệm thời gian và công sức. Hơn nữa, nó cũng có thể được tùy chỉnh theo yêu cầu của các nhà phát triển.

- Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như C/C++, C#, F#, JavaScript, JSON, Visual Basic, HTML, CSS, ...
- Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.
- Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug), …
- Hỗ trợ đa nền tảng: Linux, Mac, Windows, ...
- Ít dung lượng
- Tính năng mạnh mẽ
- Kiến trúc mạnh mẽ và người dùng có thể khai thác mở rộng

2.3.4.1 Các tính năng của PyCharm:

1. Trình sửa code thông minh

- Giúp chúng ta viết mã chất lượng cao hơn
- Nó bao gồm các lược đồ màu (color schemes) cho từ khóa, lớp và hàm. Điều này giúp tăng khả năng đọc và hiểu mã.
- Giúp xác định lỗi một cách dễ dàng.
- Cung cấp tính năng tự động hoàn thành và hướng dẫn để hoàn thành mã.

2. Điều hướng mã
   - Nó giúp các nhà phát triển chỉnh sửa và cải thiện mã với ít nỗ lực và thời gian hơn.
   - Với điều hướng mã, lập trình viên có thể dễ dàng điều hướng đến một hàm, lớp hoặc tệp.
   - Một lập trình viên có thể xác định vị trí một phần tử, một ký hiệu hoặc một biến trong mã nguồn trong thời gian ngắn.
   - Hơn nữa, bằng cách sử dụng chế độ thấu kính, nhà phát triển có thể kiểm tra và vá lỗi toàn bộ mã nguồn một cách kỹ lưỡng.
3. Tái cấu trúc
   - Nó có lợi thế là thực hiện các thay đổi hiệu quả và nhanh chóng đối với cả biến cục bộ (local variables) và biến toàn cục (global variables).
   - Tái cấu trúc trong PyCharm cho phép các nhà phát triển cải thiện cấu trúc bên trong mà không thay đổi hiệu suất bên ngoài của code.
   - Pycharm cũng giúp phân chia các lớp và chức năng mở rộng tốt hơn với sự trợ giúp của phương pháp trích xuất.
4. Hỗ trợ cho nhiều công nghệ web khác
   - Nó giúp các nhà phát triển tạo các ứng dụng web bằng Python.
   - Nó hỗ trợ các công nghệ web phổ biến như HTML, CSS và JavaScript.
   - Các nhà phát triển có lựa chọn chỉnh sửa trực tuyến với IDE này. Đồng thời, họ có thể xem trước trang web đã cập nhật/đã tạo.
   - Các nhà phát triển có thể theo dõi các thay đổi trên trình duyệt web trực tiếp.
   - PyCharm cũng hỗ trợ AngularJS và NodeJS để phát triển các ứng dụng web.
5. Hỗ trợ cho các web framework Python phổ biến
   - PyCharm hỗ trợ các web framework như Django.
   - Cung cấp tính năng tự động điền và gợi ý cho các thông số của Django.
   - Giúp vá lỗi các code của Django.
   - Hỗ trợ các web framework thông dụng như web2py và Pyramid
6. Hỗ trợ cho Thư viện Khoa học Python
   - PyCharm hỗ trợ các thư viện khoa học của Python như Matplotlib, NumPy và Anaconda.

- Các thư viện khoa học này giúp xây dựng các dự án về Khoa học Dữ liệu và Học máy.
- Hỗ trợ các biểu đồ tương tác giúp các nhà phát triển hiểu dữ liệu tốt hơn.
- Nó có khả năng tích hợp với những công cụ khác nhau như IPython, Django và Pytest. Sự tích hợp này giúp thúc đẩy các giải pháp độc đáo.

7. Ưu và nhược điểm của việc sử dụng PyCharm
   - Ưu điểm
     - Cài đặt PyCharm rất dễ dàng.
     - PyCharm là một IDE dễ sử dụng.
     - Có rất nhiều plugin hữu ích và phím tắt hữu ích trong PyCharm.
     - PyCharm tích hợp các tính năng của thư viện và IDE như tự động hoàn thành và tô màu.
     - Nó cho phép xem mã nguồn trong một cú nhấp chuột.
     - Tiết kiệm thời gian phát triển phần mềm
     - Tính năng đánh dấu lỗi trong code giúp nâng cao hơn nữa quá trình phát triển.
     - Cộng đồng các nhà phát triển Python vô cùng lớn và chúng ta có thể giải quyết các thắc mắc/ nghi ngờ của mình một cách dễ dàng.
   - Nhược điểm
     - PyCharm không miễn phí và phiên bản Professional của nó khá đắt.
     - Tính năng tự điền (auto-complete) sẽ không tốt cho các lập trình viên newbie
     - Nó có thể gây ra sự cố trong khi sửa chữa các công cụ như venv.

2.3.5 Một số thư viện được sử dụng

2.3.5.1 Thư viện Scikit-learning

Skikit-learn (Sklearn) là một thư viện mạnh mẽ nhất trong những thư viện phổ biến trong các thuật toán học máy cổ điển được viết trên ngôn ngữ Python. Thư viện Scikit-learning cung cấp các công cụ giúp xử lý các bài toán machine learning và statistical. Scikit-learning hỗ trợ hầu hết các thuật toán học có giám sát và không giám sát. Scikit-learning cũng có thể được sử dụng để khai thác dữ liệu và phân tích dữ liệu, đồng thời nó cũng chạy được trên nhiều nền tảng.

Để cài đặt scikit-learn trước tiên phải cài thư viện SciPy (Scientific Python). Những thành phần gồm:

- **Numpy**: Thư viện xử lý dãy số và ma trận nhiều chiều

- **category_encoders**: Thư viện giúp đưa dữ liệu về dạng để máy có thể hiểu và làm việc
- **Cross Validation**: Kiểm thử chéo, đánh giá độ hiệu quả của thuật toán học giám sát sử dụng dữ liệu kiểm thử (validation data) trong quá trình huấn luyện mô hình(trong bài dùng K-Fold)
- IPython: Notebook dùng để tương tác trực quan với Python
- SymPy: Thư viện các kí tự toán học
- **Pandas**: Thư viện xử lý, phân tích dữ liệu dưới dạng bảng

Về cơ bản, các thư viện Numpy, Pandas là hai trong những thư viện cần thiết hơn cả.

Hình 2.30 Minh họa thư viện Scikit-learn

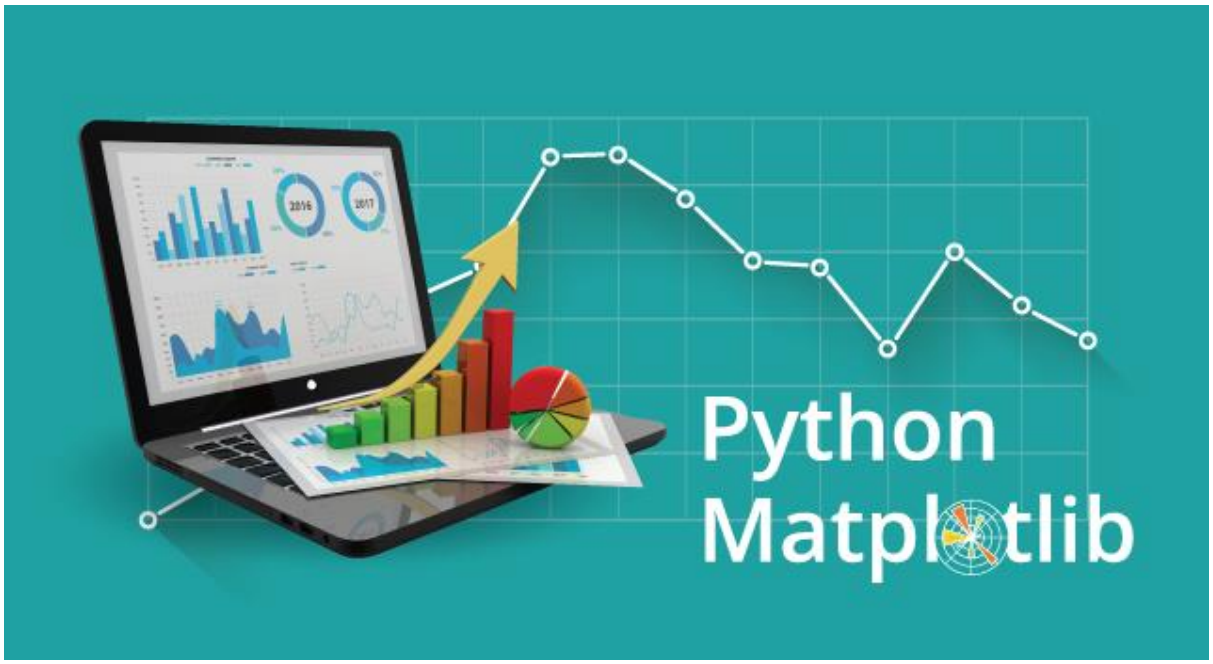
2.3.5.2 Thư viện Numpy

NumPy là một thư viện python rất phổ biến để xử lý mảng và ma trận lớn đa chiều, với sự trợ giúp của một bộ sưu tập lớn các hàm toán học cấp cao. Numpy rất hữu ích cho các tính toán khoa học cơ bản trong học máy. Nó cũng là một thư viện giúp cho các lập trình tạo và quản lý nhóm, thao tác với các hình dạng logic và đặc biệt thực hiện phép toán đại số tuyến tính, biến đổi Fourier và số ngẫu nhiên.

Hình 2.31 Minh họa thư viện numpy

2.3.5.3 Thư viện Matplotlib

Matpoltlib là một thư viện Python rất phổ biến để trực quan hóa dữ liệu. Các lập trình viên sử dụng nó để hiển thị dữ liệu dưới dạng đồ thị. Nó đặc biệt hữu ích khi một lập trình viên muốn hình dung các mẫu trong dữ liệu. Matpoltlib có thể hiển thị dữ liệu dưới dạng các biểu đồ khác nhau ví dụ như biểu đồ đường, biểu đồ cột.

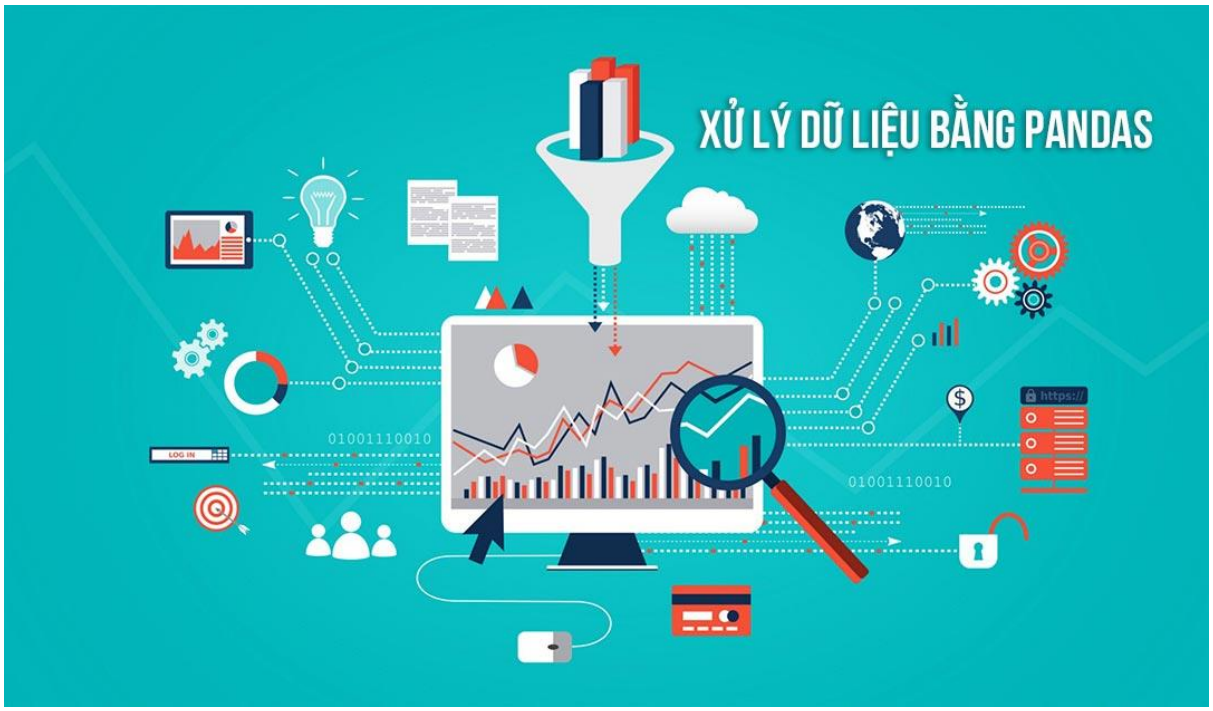


Hình 2.32 Ảnh minh họa thư viện Matplotlib

2.3.5.4 Thư viện Pandas

Pandas là một thư viện Python phổ biến để phân tích dữ liệu. Pandas cung cấp cấu trúc dữ liệu cấp cao được tối ưu hóa và nhiều công cụ đa dạng để phân tích dữ liệu

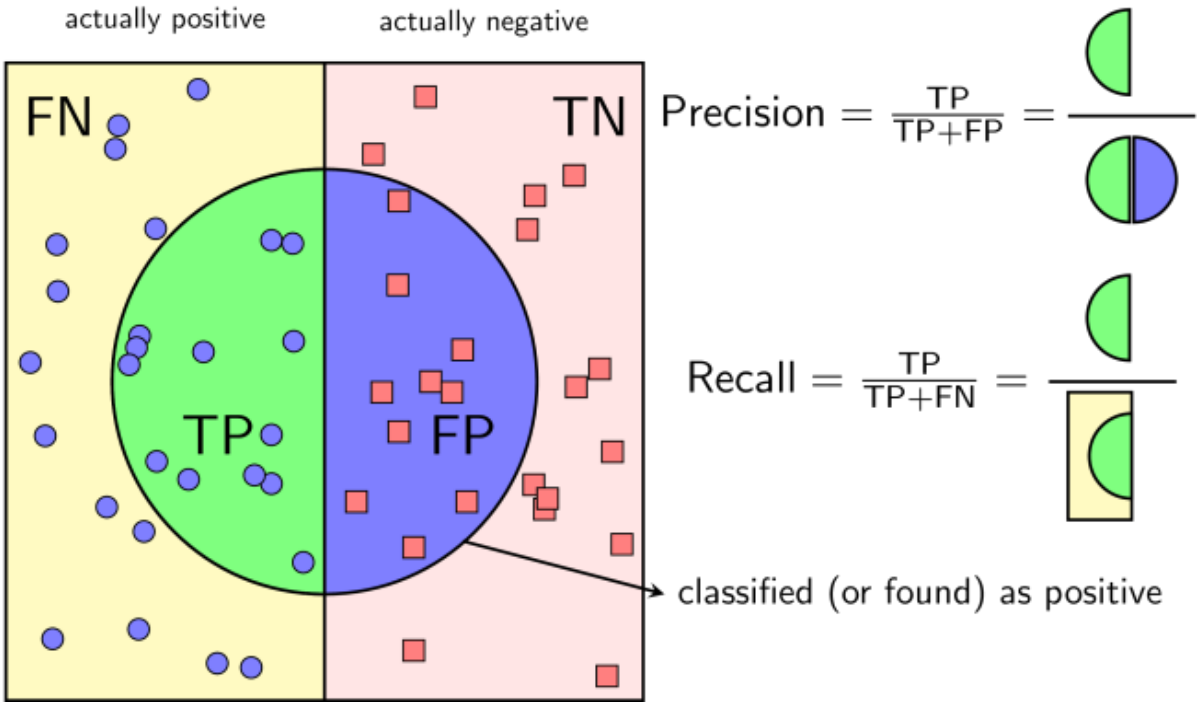
chuỗi thời gian và dữ liệu có cấu trúc. Thư viện này được sử dụng nhiều trong khoa học dữ liệu, phân tích dữ liệu và học máy. Bên cạnh đó, nó cung cấp nhiều phương pháp có sẵn để dò tìm, kết hợp và lọc dữ liệu. Ví dụ, ta có thể sử dụng Pandas để đọc, ghi, lọc và nhóm các dữ liệu.



Hình 2.33 Ảnh 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 điểm mô hình dự đoán là positive đều là positive càng nhiều. Precision = 1, tức là tất cả số điểm mô hình dự doán là Positive đều đúng, hay không có điểm nào có nhãn là Negative mà mô hình dự đoán nhầm là Positive.

- Recall càng cao, tức là số điểm là positive bị bỏ sót càng ít. Recall = 1, tức là tất cả số điểm có nhãn là Positive đều được mô hình nhận ra.

- Để hiểu rõ hơn về hai chỉ số này, ta có thể tưởng tượng một ví dụ như sau. Khi một người nghĩ là mình đang mắc bệnh gì đó, họ thường đi đến bệnh viện để làm các xét nghiệm để bác sĩ chẩn đoán xem kết quả là dương tính hay là âm tính. Ta có hai trường hợp về tình trạng bệnh là mắc bệnh hoặc không mắc bệnh. Ta có hai trường hợp về kết quả chẩn đoán là dương tính và âm tính.

- Khi đó, precision là tỉ lệ người được chẩn đoán là dương tính thật sự mắc bệnh trên tổng số người được chẩn đoán là dương tính. Nếu precision = 0.9, thì cứ 100 người được chẩn đoán là dương tính thì sẽ thật sự có 90 người mắc bệnh. Precision càng cao thì xác suất người được chẩn đoán là dương tính có khả năng mắc bệnh càng cao.

- Recall là tỉ lệ người được chẩn đoán là dương tính thật sự mắc bệnh trên tổng số người thật sự mắc bệnh. Nếu recall = 0.9, thì cứ 100 người mắc bệnh thì sẽ chẩn đoán 90 người dương tính. Recall càng cao thì xác suất người mắc bệnh được chẩn đoán là dương tính càng cao.

- F1-score

+ Tuy nhiên, chỉ có Precision hay chỉ có Recall thì không đánh giá được chất lượng mô hình.

+ Chỉ dùng Precision, mô hình chỉ đưa ra dự đoán cho một điểm mà nó chắc chắn nhất. Khi đó Precision = 1, tuy nhiên ta không thể nói là mô hình này tốt.

+ Chỉ dùng Recall, nếu mô hình dự đoán tất cả các điểm đều là positive. Khi đó Recall = 1, tuy nhiên ta cũng không thể nói đây là mô hình tốt.

Khi đó F1-score được sử dụng. F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0)

- Chi tiết hơn thì Precision được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm mô hình dự đoán là Positive. Recall được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu).

- Hệ số xác định R^2 là ma trận cho thấy mức độ hiệu quả của mô hình dự đoán kết quả. Thống kê này cho thấy tỷ lệ phương sai trong kết quả mà mô hình có thể dự đoán dựa trên các đặc điểm của nó.

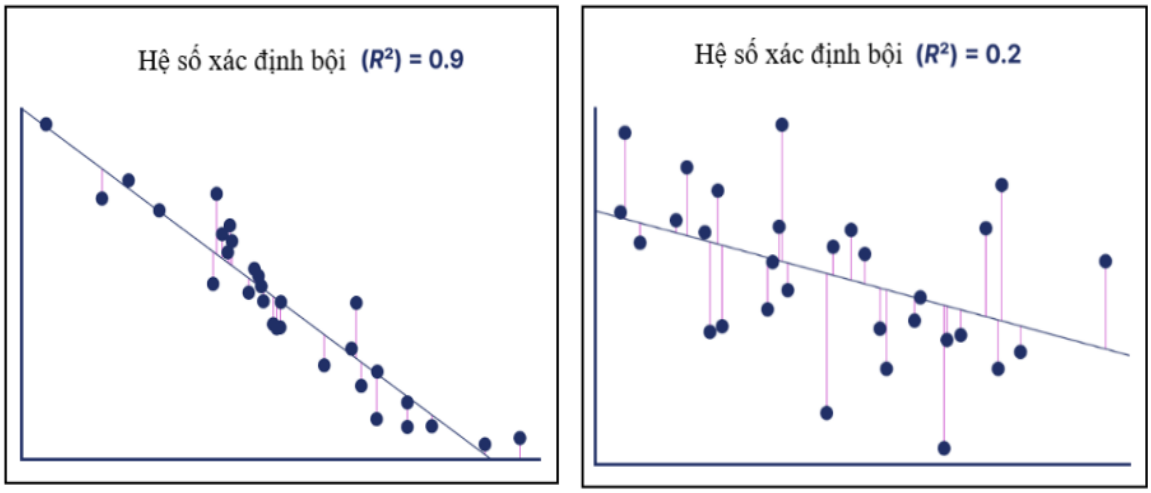
Ta có công thức:

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

Trong đó $\hat{y}_i$ là giá trị dự đoán, y_i là giá trị thực. $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ và $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$

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

- Nếu R^2 càng tiến về 0(từ 0.5 đến 0), mô hình dự đoán chưa 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 tiến đến 1, mô hình dự đoán là một mô hình tốt, R^2 càng gần 1 thì khả năng dự đoán chính xác càng cao
- Nếu R^2 bằng 1, điều này là không thể bởi luôn luôn xuất hiện phần dư trong mô hình



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)

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

(2.18)

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

Trong đó: y_{test} là giá trị thực tế, y_{pre} là 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 giá trị tuyệt đối sai số giữa giá trị thực tế và giá trị dự đoán

(2.19)

$$\mathbf{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_{test} - y_{pre}\right|$$

Trong đó: y_{test} là giá trị thực tế, y_{pre} là giá trị dự đoán

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

RMSE là độ lệch chuẩn của phần dư (lỗi dự đoán). RMSE là thước đo mức độ hiệu quả của mô hình bằng cách đo sự khác biệt giữa các 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} là giá trị thực tế, y_{pre} là giá trị dự đoán

- Ở trong bài toán này em sẽ chọn 3 phương pháp tính độ đo đó là độ đo Precision, Recall, F1-score. Qua các lần máy học sẽ chọn ra một mô hình mà ở đó cả 3 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 vấn đề mà mọi sinh viên đều trăn trở lo nghĩ đặc biệt là với những sinh viên năm cuối sắp tốt nghiệp ra trường

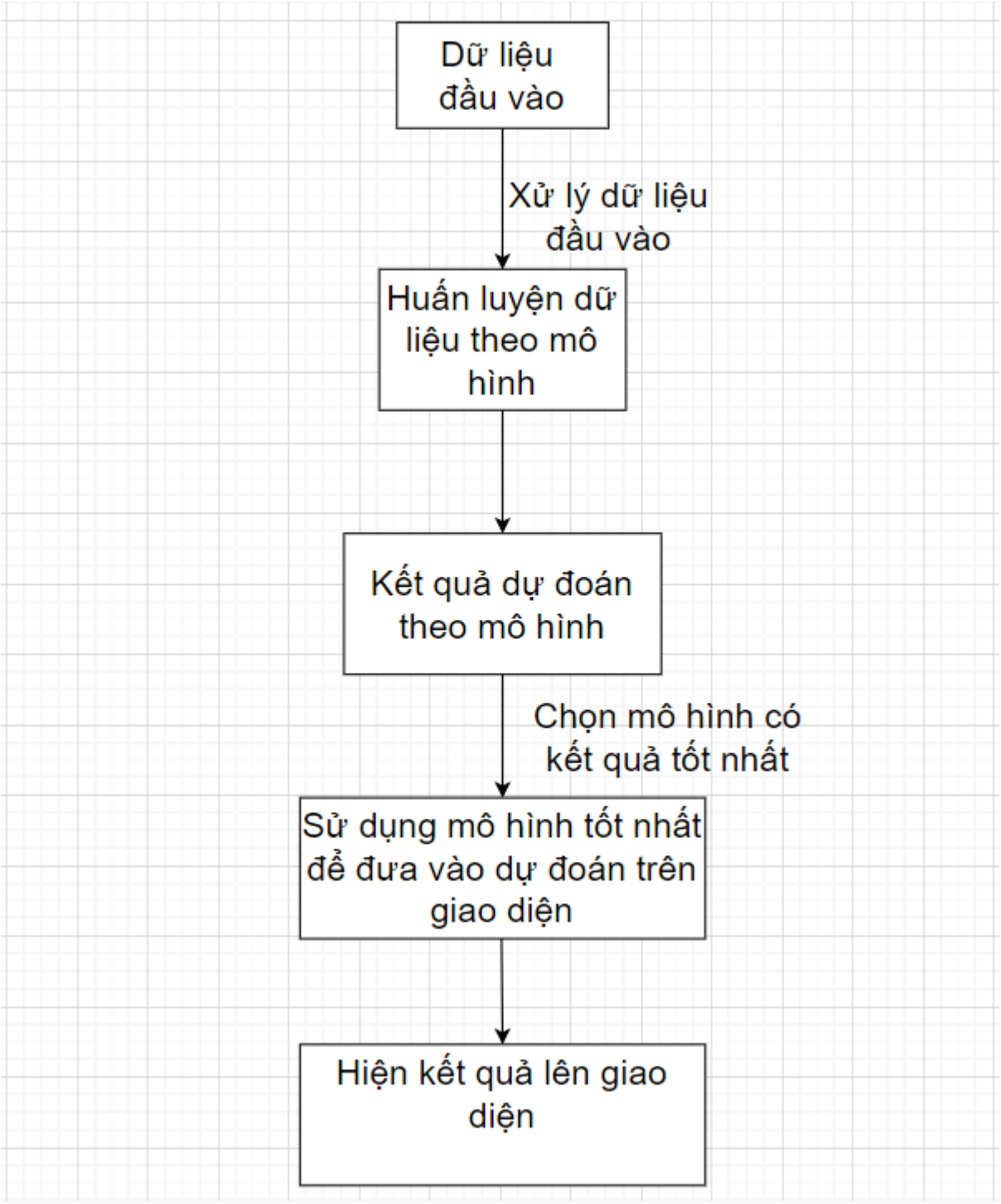
Hiện nay, các công ty doanh nghiệp đều có yêu cầu nhất định đối với nguồn nhân lực mới, trẻ và các sinh viên đều lo lắng không biết mình có cơ hội để được nhận vào làm không. Sau khi nghiên cứu, phân tích một số nền tảng, công cụ về học máy bài báo cáo này quyết định sử dụng thuật toán Cây quyết định Iterative Dichotomiser 3 (ID3) để xây dựng mô hình dự đoán cơ hội việc làm cho sinh viên năm cuối. Bên cạnh việc xây dựng mô hình dự đoán, báo cáo này còn định hướng đưa kết quả dự đoán lên giao diện người dùng có thể tương tác.

Như vậy, yêu cầu chung của bài toán:

- Giao diện có đủ tính năng của một hệ thống về dự đoán:
    - Người dùng có thể nhập, chọn trực tiếp số liệu trên giao diện
    - Hiện thị kết quả dự đoán chính xác nhất của mô hình trên giao diện
- Giao diện đảm bảo dễ sử dụng, dễ hiểu
- Đưa ra kết quả dự đoán cho độ chính xác cao hơn

3.1.2 Quy trình thực hiện

Với các yêu cầu như trên, chương trình được thực hiện như sơ đồ dưới đây:



Hình 3.1 Sơ đồ quy trình thực hiện bài toán

3.2 Xây dựng mô hình học máy

Để đi tới kết quả dự đoán, em thực hiện tiền xử lý dữ liệu và xây dựng mô hình dự đoán với thuật toán cây quyết định ID3(Iterative Dichotomiser 3).

Kiểm tra độ chính xác của máy học dựa trên cross validation K-Fold. Đây là phương pháp nâng cấp của hold-out. Toàn bộ dữ liệu được chia thành K tập con. Quá trình học của máy có K lần. Trong mỗi lần, một tập con được dùng để kiểm tra và K-1 tập còn lại dùng để dạy.

Các thực nghiệm được đánh giá trên cùng tập dữ liệu huấn luyện, kiểm thử và một số phương pháp đánh giá độ tin cậy: Precision, Recall và F1_score

Với các tham số thực nghiệm được lựa chọn thông qua các thực nghiệm thay đổi tham số, kết quả được đưa ra dưới đây là các tham số đạt kết quả tốt nhất.

Sau khi lựa chọn được thuật toán học máy có kết quả tốt nhất cho tập dữ liệu, tiến hành xây dựng giao diện trên mô hình đã lựa chọn từ đó có thể giúp người dùng thao tác với giao diện.

3.2.1 Môi trường thực nghiệm

- Máy HP RAM 8G; ổ cứng SSD, HDD; CPU core i- 6820HQ 2.70GHz.
- Hệ điều hành Windows 10 pro 64bit
- Python phiên bản 3.10.2
- Framework PyQt

3.2.2 Dữ liệu đầu vào

Dữ liệu đầu vào được thu thập tại trang web cộng đồng dành cho dân trí tuệ nhân tạo và khoa học dữ liệu trên toàn cầu kaggle (data) bản ghi đo được vào năm 2013 và 2014 tổng gồm 2000 trường dữ liệu .

Sau khi có được số liệu ban đầu, tiến hành chuẩn hoá dữ liệu với category_encoders để đưa toàn bộ dữ liệu từ dạng chữ về dạng số. Sau đó chia K-Fold với các lần train và test lần lượt là K-1 train và 1 test từ đó tìm ra quá trình máy học đạt kết quả tốt nhất

Như vậy, trong đồ án này em sử dụng tập dữ liệu đầu vào được chia ra để train và test, từ đó dựa vào các tham số đánh giá mô hình lựa chọn ra tập dữ liệu tốt nhất cho bài toán và mô hình tốt nhất cho tập dữ liệu đó.

3.2.3 Xây dựng mô hình dự đoán

Xây dựng mô hình dự đoán cơ hội việc làm dành cho sinh viên năm cuối với các tham số:

- Mô hình thuật toán cây quyết định Iterative Dichotomiser 3 sử dụng DecisionTreeClassifier () với các tham số đầu vào mặc định

➢ Với tập dữ liệu đưa vào mô hình dự đoán (mô hình dự đoán dựa trên thuật toán cây quyết định Iterative Dichotomiser 3)

Tập dữ liệu đưa vào gồm 7 cột X tương ứng dữ liệu về:

+Age

+ Gender

+ Stream

+ Internships

+ CGPA

+ Hostel

+ HistoryOfBacklogs

Cột Y tương ứng với kết quả "có" hoặc "không" về khả năng nhận được cơ hội việc làm

Dữ liệu được chia thành hai tập: train/test theo tỷ lệ K-Fold tuỳ người dùng nhập (tức là trong quá trình học của máy sẽ có K lần với mỗi lần tỷ lệ sẽ là 1 tập để test và K-1 tập còn lại dùng để train) nhằm kiểm tra độ chính xác của mô hình và chọn ra mô hình tốt nhất

3.3 Xây dựng giao diện hiển thị kết quả

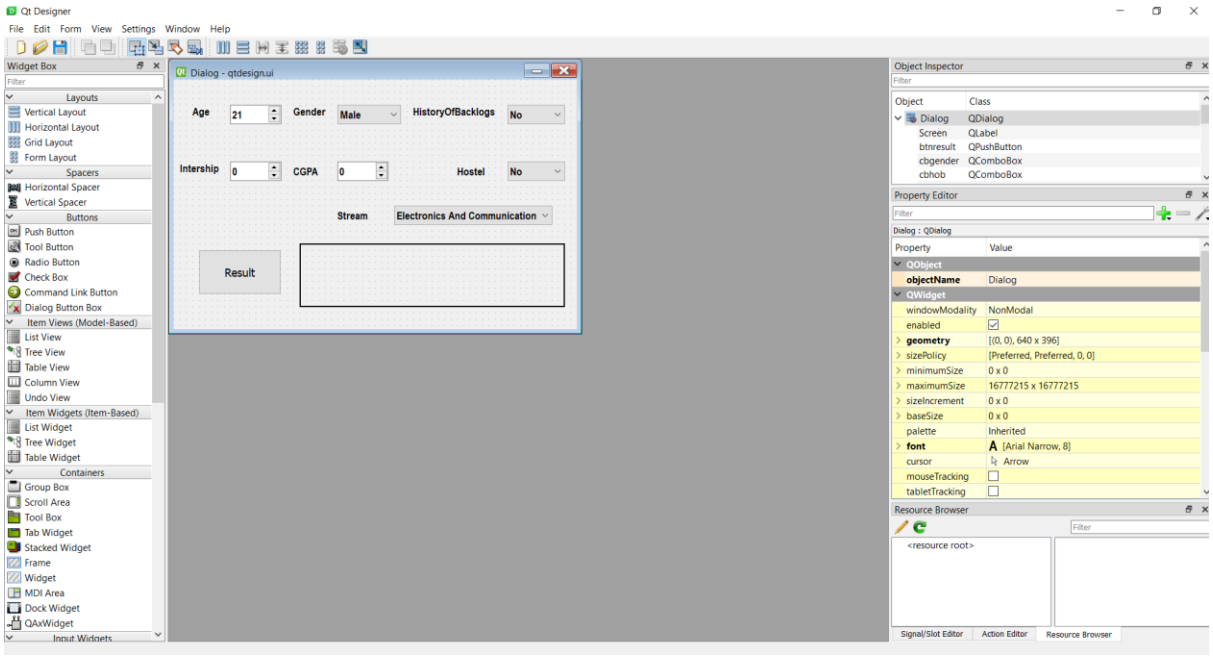
- Với mô hình thuật toán cây quyết định Iterative Dichotomiser 3, em dùng toàn bộ tập dữ liệu ở trường hợp mà có lần máy học có tỷ lệ dự đoán tốt nhất là tập huấn luyện cho mô hình. Mô hình được huấn luyện có kết quả tốt nhất sẽ được

lưu lại và sử dụng mô hình đó làm mô hình để đưa ra kết quả dự đoán khi người dùng tương tác trên giao diện

- Yêu cầu cơ bản đối với giao diện hiển thị kết quả:
    - Có các trường dữ liệu để người dùng có thể nhập dữ liệu để đưa ra dự đoán
    - Sử dụng kết quả dự đoán có độ chính xác cao nhất
    - Giao diện thân thiện, dễ sử dụng

Giao diện bao gồm hai phần chính:

- Phần cho người dùng nhập, chọn các giá trị tương ứng với từng dữ liệu của bản thân để đưa vào mô hình đưa ra dự đoán
- Phần đưa ra kết quả dự đoán cho bài toán



Hình 3.2 Thiết kế giao diện với Qt Designer

Xây dựng giao diện sẽ gồm 9 thành phần chính. Trong đó có 7 mục là dữ liệu đầu vào có:

- 3 Spin Box
    + Age
    + Internships
    + CGPA
- 4 Combo Box
    + Gender
    + Stream
    + Hostel
    + HistoryOfBacklogs

Có 1 nút bấm để xử lý kết quả dự đoán và một ô text để hiện ra kết quả dự đoán về khả năng sinh viên đó có nhận được cơ hội việc làm hay không

CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

4.1 Kết quả và đánh giá mô hình

4.1.1 Kết quả mô hình

- Kết quả đánh giá mô hình ở trường hợp K = 7. Quá trình học của máy sẽ có 7 lần với mỗi lần tỷ lệ sẽ là 1 tập để test và 6 tập còn lại dùng để train

1. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ nhất

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8074270291883247	79.72 %
Recall	0.8015196078431372	
F1-score	0.7967156862745097	

Bảng 1 Lần máy học thứ 1

Dựa vào bảng cho thấy, ở trường hợp này, độ đo Precision và độ đo Recall chênh nhau không đáng kể trong khi độ đo F1-score thấp hơn còn tỷ lệ dự đoán chính xác đạt 79.72%.

2. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ hai

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8406945463438493	83.92 %
Recall	0.8563097856667861	
F1-score	0.8376043055349824	

Bảng 2 Lần máy học thứ 2

Dựa vào bảng cho thấy, ở lần máy học thứ 2 độ đo Precision, độ đo Recall và F1-score đều khác nhau với độ đo cao nhất là Recall còn độ đo thấp nhất là F1-score còn tỷ lệ dự đoán chính xác đạt 83.92 %

3. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ ba

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8460677749360614	83.57 %
Recall	0.8227329192546584	
F1-score	0.8283598288961247	

Bảng 3 Lần máy học thứ 3

Dựa vào bảng cho thấy, ở lần máy học thứ 3 độ đo Recall và F1-score đạt 0.82 thấp hơn độ đo Precision cao nhất là 0.84 còn tỷ lệ dự đoán chính xác đạt 83.57 %

4. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ tư

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8329032258064516	80.07 %
Recall	0.802861335289802	
F1-score	0.7965175060850029	

Bảng 4 Lần máy học thứ 4

Dựa vào bảng cho thấy, ở lần máy học thứ 4 độ đo Precision, độ đo Recall và F1-score đều thấp hơn so với lần máy học thứ 3 với độ đo cao nhất là Precision còn độ đo thấp nhất là F1-score còn tỷ lệ dự đoán chính xác đạt 80.07 %

5. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ 5

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8942462230793957	90.88 %
Recall	0.9055886243386244	
F1-score	0.8993971977842945	

Bảng 5 Lần máy học thứ 5

Dựa vào bảng cho thấy, ở lần máy học thứ 5 độ đo Precision, độ đo Recall và F1-score đều cao. Độ đo nhất là Recall còn độ đo Precision và F1-score thấp hơn còn tỷ lệ dự đoán chính xác đạt 90.88 %

6. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ 6

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8036294691224268	80.35 %
Recall	0.8049812030075187	
F1-score	0.803312629399586	

Bảng 6 Lần máy học thứ 6

Dựa vào bảng cho thấy, ở lần máy học thứ 6 độ đo Precision, độ đo Recall và F1-score đều khá đồng đều nhưng thấp hơn hẳn so với độ đo ở lần học thứ 5 còn tỷ lệ dự đoán chính xác đạt 80.35 %

7. Đánh giá mô hình học máy qua 3 độ đo: Precision, Recall, F1-score cùng với tỷ lệ dự đoán chính xác của mô hình học máy ở lần học thứ 7

Độ đo	Kết quả	Tỷ lệ dự đoán chính xác
Precision	0.8467908902691512	84.56 %
Recall	0.8474074074074074	
F1-score	0.845596926713948	

Bảng 7 Lần máy học thứ 7

Dựa vào bảng cho thấy, ở lần đo cuối cđộ đo Precision, độ đo Recall và F1-score đều gần như bằng nhau với độ đo cao nhất là Recall còn độ đo thấp nhất là F1-score còn tỷ lệ dự đoán chính xác đạt 84.56 %

- Như vậy, cả 7 lần máy học thì tỷ lệ dự đoán tương chính xác đều tương đối cao. Độ đo Precision thấp nhất là ở lần máy học thứ 6 còn cao nhất là ở lần máy học thứ 5. Recall thấp nhất ở lần máy học thứ 1 cao nhất là lần máy học thứ 5, còn độ đo F1-Score thấp nhất ở lần máy học thứ 1 cao nhất vẫn là lần máy học thứ 5. Từ đó ta thấy kết quả máy học ở lần học thứ 5 là mô hình tốt nhất. Tỷ lệ dự đoán chính xác ở lần máy học thứ 5 đạt tới 90.88 %

4.1.2 Đánh giá và lựa chọn mô hình

Sau khi đã thu được kết quả dự đoán, ta tổng hợp kết quả thu được và lấy tiêu chí ở tỷ lệ dự đoán chính xác, từ đó thấy được tổng quan việc dự đoán của mô hình:

	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 %

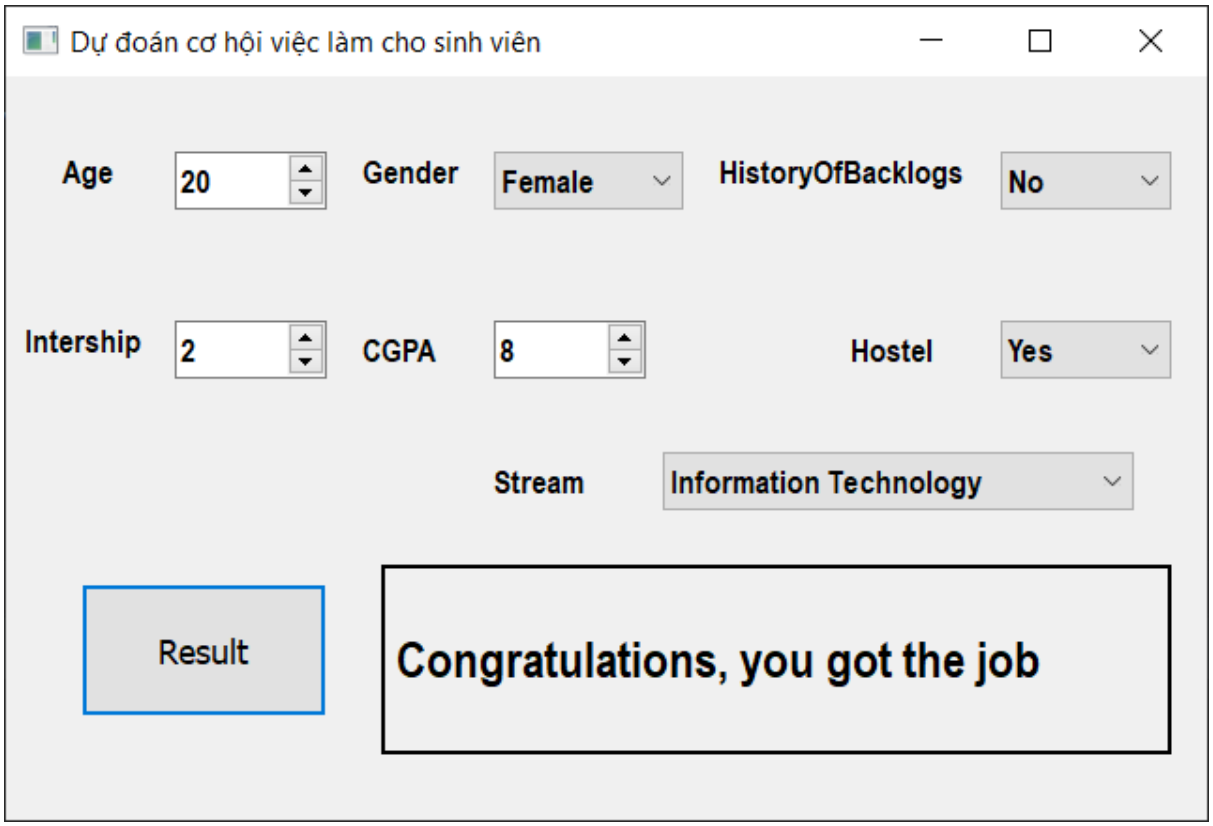
Lần 3	0.8460	0.8227	0.8283	83.57 %
Lần 4	0.8329	0.8028	0.7965	80.07 %
Lần 5	0.8942	0.9055	0.8993	90.88 %
Lần 6	0.8036	0.8049	0.8033	80.35 %
Lần 7	0.8467	0.8474	0.8455	84.56 %

Bảng 8 Tổng hợp kết quả độ đo và dự đoán

Từ bảng tổng hợp ta thấy, ở lần máy học thứ 5 thì tỷ lệ dự đoán chính xác và các độ đo Precision, Recall và F1-score là cao nhất

Do đó, với bài toán dự đoán cơ hội việc làm cho sinh viên năm cuối, em quyết định sử dụng mô hình thuật toán của lần máy học thứ 5 từ đó làm mô hình để dự đoán cho giao diện hiển thị

4.2 Demo giao diện hiển thị



Hình 4.1 Demo giao diện hiển thị

KẾT LUẬN

❖ Nội dung đã đạt được

- ✓ Nghiên cứu tìm hiểu bài toán đoán cơ hội việc làm cho sinh viên năm cuối, ứng dụng học máy để xây dựng mô hình dự đoán
- ✓ Phân tích tìm hiểu về mô hình cây quyết định
- ✓ Tìm hiểu và ứng dụng thuật toán cây quyết định ID3 trong học máy vào bài toán để tìm ra mô hình cho ra kết quả tốt nhất
- ✓ Xây dựng một giao diện tương tác hiển thị kết quả dự đoán của mô hình trên nền tảng GUI Qt Designer

❖ Hướng phát triển

- ✓ Nghiên cứu thêm một số kiến thức về các thuật toán khác trong học máy
- ✓ Xử lý dữ liệu và đào tạo mô hình để đưa ra dự đoán không chỉ dành cho các sinh viên năm cuối mà dành cho mọi sinh viên
- ✓ Cải thiện thêm giao diện hiển thị kết quả, có thể tương tác thêm nhiều chức năng khác

❖ Hạn chế

- ✓ Do kinh nghiệm và kiến thức còn hạn hẹp, ĐATN này chỉ dừng lại ở việc nghiên cứu, ứng dụng thuật toán vào bài toán thực tế. Mặc dù, kết quả dự đoán thu được tương đối khả quan nhưng vẫn còn nhiều thiếu sót từ khâu chuẩn bị dữ liệu đến chia tập dự liệu và đánh giá mô hình. Do vậy ĐATN này còn nhiều thiếu sót, mong nhận được sự góp ý và sửa đổi của thầy cô.

DANH MỤC TÀI LIỆU THAM KHẢO

1. https://machinelearningcoban.com/
2. Slide bài giảng về học máy https://sites.google.com/a/wru.vn/cse445fall2016/lecture-materials
3. Slide bài giảng khai phá dữ liệu https://sites.google.com/site/tlucse404/
4. https://www.kaggle.com/tejashvi14/engineering-placements-prediction
5. https://1upnote.me/post/2018/10/ds-ml-decision-tree-id3/
6. https://stackoverflow.com/
7. Sử dụng QT Designer https://doc.qt.io/qt-6/qtdesigner-manual.html
8. Phương pháp đánh giá độ tin cậy của mô hình học máy https://rabiloo.com/vi/blog/cac-phuong-phap-danh-gia-mo-hinh-machine-learning-va-deep-learning